

Computational genomics of selenoproteins

Marco Mariotti

TESI DOCTORAL UPF / ANY 2013

DIRECTOR DE LA TESI

Roderic Guigó - Departament Bioinformatics and Genomics, Centre de Regulació Genòmica (CRG)



Abstract

Selenoproteins are a diverse class of proteins containing selenocysteine, the 21st amino acid. Selenocysteine is inserted co-translationally, recoding very specific UGA codons through a dedicated machinery. Standard gene prediction programs consider UGA only as translational stop, and for this reason selenoprotein genes are typically misannotated. In the past years, we developed computational tools to predict selenoproteins at genomics scale. With these, we characterized the set of selenoproteins across many sequenced genomes, and we inferred their phylogenetic history. We dedicated particular attention to selenophosphate synthetase, a selenoprotein family required for selenocysteine biosynthesis, that can be used as marker of the selenocysteine coding trait. We show that selenoproteins went through a very diverse evolution in different lineages. While very conserved in vertebrates, selenoproteins were lost independently in many other organisms. Using genome sequencing, we traced with precision the path of genomic events that lead to recent selenoprotein extinctions in certain fruit flies.

Resum

Les selenoproteïnes s'agrupen en una classe heterogènia de proteïnes les quals contenen selenocisteïna, l'aminoàcid 21. La selenocisteïna és insertada durant el procés de traducció, recodificant codons UGA molt específics, mitjançant una maquinària dedicada. Els programes estàndard de predicció de gens interpreten el codó UGA només com a senyal d'*stop* de la traducció, i per aquesta raó els gens de selenoproteïnes solen estar mal anotats. En els darrers anys, hem desenvolupat eines computacionals per a predir selenoproteïnes a escala genòmica. Amb aquestes, hem caracteritzat el conjunt de selenoproteïnes en aquells genomes que han estat seqüenciats, inferint la seva història filogenètica. Hem dedicat especial atenció a la família *selenophosphate synthetase*, selenoproteïna necessària per a la síntesi de selenocisteïna, i que per tant pot ser utilitzada com a marcador de codificació de selenocisteïna. Mostrem que les selenoproteïnes han patit una evolució molt diversa en diferents llinatges. Tot i que es troben molt conservades en vertebrats, les selenoproteïnes van ser perdudes de manera independent en molts altres organismes. Gràcies a la seqüenciació de genomes, vam traçar amb precisió els esdeveniments que van portar a l'extinció de selenoproteïnes a diverses espècies de drosòfila.

Prologue: why study selenoproteins?

Selenoproteins are proteins that contain the amino acid selenocysteine (Sec), the 21st amino acid. This is inserted during translation, by recoding an in-frame UGA (normally a stop codon). Sec is synthesized on its own tRNA, and a set of factors is dedicated to both its production and insertion. There are very few selenoprotein genes in genomes. Human has 25, mouse 24, common fruit fly has 3 and *C.elegans* has just 1. Plants, molds, some insects and a lot of others lineages has none. Then why should we study selenoproteins? It is a small percentage of the total proteome. We could think they have little effect, and little importance. Some fruit flies have no selenoproteins, and they are doing fine. Many other insects lost naturally their selenoproteins: ants, bees, beetles, butterflies. By selenoprotein loss, we mean that selenoprotein genes either disappeared from the genome, or were converted to cysteine homologues, mutating the Sec-UGA codon into UGU or UGC. When there are no more selenoproteins in a genome, the translation machinery degenerates, and the species loses its ability to code selenocysteine. It looks like selenocysteine is of little importance to insects. But it was to their ancestors. The selenoproteome (set of selenoproteins) size of the last common ancestor of insects is estimated 4-5. Going up on the phylogenetic tree (thus back on time) we find arthropods, in which we have 15-20. In this thesis, we will show how this set of selenoproteins was reduced on the road to fruit flies, in steps that can be mapped to common ancestors with other species. Thus, the fruit flies can live without selenoproteins because they dropped them gradually, transferring their functions to different genes. For human it is a different story. Our selenoproteome consists of 25 genes, most of them well conserved in all vertebrates. Although our ancestors certainly went through a lot of genome transformations too, their selenoproteome was mostly kept intact since the metazoan radiation. In vertebrates, selenoproteins constitute an arsenal of redox enzymes active mainly in the anti-oxidant defense, but also in many other processes: thyroid hormone maturation, selenium transport, folding control in the endoplasmic reticulum. Many human selenoproteins are still functionally uncharacterized. Selenocysteine is important to human, since selenoproteins are playing essential roles. And this is a sufficient reason to study them. But it is when we zoom out from our species, that selenocysteine gets really interesting: it can be a prop to understand how evolution works at the gene function level. Proteins are functionally linked in very complex ways. Each one depends on a lot of other proteins in the genome: if one of those gets compromised, the protein cannot perform its function. Evolution shapes this complex network during time. Some links are broken, some new are created, as selection acts on the functions of the proteins. The story of the selenocysteine and selenoproteins, with its roots in the last universal common ancestor, is a insightful snapshot of this phenomena in act. In this thesis, I tried to reconstruct the history of selenoproteins along the tree of life, with particular attention to vertebrates, to insects, and to the major points of radiation of the tree of life. Mostly, my work consisted in developing bioinformatics tools to find the selenoprotein genes and reconstruct their phylogenetic relationships. Thanks to the growing number of public genomes, and also to the sequencing effort of my lab, we were able to follow selenoproteins both on a small and large evolutionary scale.

Contents

1	INTRODUCTION	1
1.1	Selenocysteine, the 21st amino acid	1
1.1.1	What are selenoproteins?	1
1.1.2	(Un)related to selenoproteins	2
1.2	Sec machinery	3
1.2.1	Eukaryotic Sec synthesis on its own tRNA	3
1.2.2	Eukaryotic Sec insertion: tweaking translation	4
1.2.3	The bacterial Sec machinery	5
1.2.4	The archaeal Sec machinery	5
1.2.5	SECIS elements	7
1.3	Selenoprotein genomics	10
1.3.1	Selenoproteinless organisms and cysteine homologues	10
1.3.2	Selenocysteine vs cysteine	10
1.3.3	Known selenoprotein families	12
1.3.4	Selenoproteins in vertebrates	16
1.3.5	Hierarchical regulation of selenium supply	18
1.3.6	Sec extinctions in insects	18
1.3.7	Nematodes: a minimal selenoproteome	19
1.3.8	Selenoproteins in non-animal eukaryotes	21
1.3.9	Prokaryotic vs eukaryotic selenoproteome	23
1.4	Bioinformatics of selenoproteins	24
1.4.1	Selenoproteins are misannotated	24
1.4.2	Novel selenoprotein identification in eukaryotes	25
1.4.3	Novel selenoprotein identification in prokaryotes	27
1.4.4	Annotation of known selenoproteins	28
2	METHODS	31
2.1	Selenoprofiles	31
2.1.1	Automatic selenoprotein annotation	32
2.1.2	Selenoprofiles paper	32
2.2	SECISearch3 and Seblastian	46
2.2.1	Computational identification of SECIS elements	47
2.2.2	SECISearch3 and Seblastian paper	48

3	RESULTS	69
3.1	Consortium projects	69
3.1.1	Selenoproteins in the gencode reference annotation	70
3.1.2	A novel selenoprotein extinction in the genome of pea aphid	72
3.1.3	Centipede genome annotation	72
3.2	SelenoDB 2.0	76
3.3	The vertebrate selenoproteome	78
3.3.1	Phylogeny of selenoproteins in vertebrates and mammals .	79
3.3.2	Vertebrate selenoproteome paper	79
3.4	Selenoprotein extinctions in insects	98
3.4.1	The known Sec extinction in <i>D.willistoni</i>	98
3.4.2	Sampling selenoproteins in drosophila by degenerate PCR	98
3.4.3	Genome sequencing of 8 drosophila from the Saltans group	99
3.4.4	Building a phylogenetic tree of all 29 sequenced drosophila	101
3.4.5	Novel Sec extinctions in the Saltans group	103
3.4.6	Full annotation of drosophila genomes	108
3.4.7	GC content and codon usage shift in Willistoni/Saltans . .	113
3.4.8	Widening the picture: other arthropods	116
3.4.9	A functional model for selenocysteine in drosophila . . .	119
3.4.10	Why Willistoni/Saltans?	121
3.4.11	Conclusions	123
3.5	The SelenoPhosphate Synthetase family (SPS)	125
3.5.1	Abstract	126
3.5.2	Introduction	126
3.5.3	Results and Discussion	127
3.5.4	Conclusions	144
3.5.5	Methods	145
3.5.6	Supplementary Material	145
4	DISCUSSION	181
4.1	Are selenoproteins essential?	181
4.2	Selenoproteins as test case	182
4.3	Before and after	183
4.4	... and next	184
4.4.1	Selenoprofiles as genome annotation tool	184
4.4.2	Future research on Willistoni/Saltans	184
4.4.3	The SPS story, and the unknown amino acid	185
5	CONCLUSIONS	187
	Bibliography	189
	Appendix: Selenoprofiles 3.0 manual	205

Chapter 1

INTRODUCTION

1.1 Selenocysteine, the 21st amino acid

1.1.1 What are selenoproteins?

Selenoproteins are a diverse group of proteins containing selenocysteine residue (Sec). Selenocysteine is a non-standard amino acid analog to cysteine, with selenium replacing sulfur. Like the 20 standard amino acids, it is inserted co-translationally and has its own tRNA. Nonetheless, Sec is not found in all genomes. Also, it does not have a fully dedicated codon. Instead, it is inserted in correspondence of a UGA codon, which normally signals for translation termination. In selenoproteins transcripts, a complex molecular mechanism takes place to “recode” UGA to Sec. A set of trans-factors is required to produce and insert selenocysteine, which we collectively call Sec machinery. Also, a specific stem-loop structure is required on the selenoprotein transcripts, called the SECIS element. For these reasons, selenocysteine is considered an extension of the genetic code, and is often referred to as the 21st amino acid [Böck et al., 1991].

Selenoproteins have been discovered in the seventies, first in Bacteria [Turner and Stadtman, 1973; Andreesen and Ljungdahl, 1973] then also in mammals [Flohe et al., 1973], through the identification of selenium in the purified protein. Soon, selenocysteine was identified as the actual selenium carrier [Cone et al., 1976]. However, at the time DNA sequencing technologies were very expensive and not commonly used, and the presence of the in-frame UGA in the gene sequence was noted only 10 years later [Chambers et al., 1986]. This prompted years of research to characterize the production of Sec and its insertion of into selenoproteins. Today, these mechanisms and their players are quite well characterized in Bacteria [Yoshizawa and Böck, 2009; Kryukov and Gladyshev, 2004] and Eukarya [Squires and Berry, 2008], and more poorly also in Archaea [Rother et al., 2001]. In the next pages, we will review these processes.

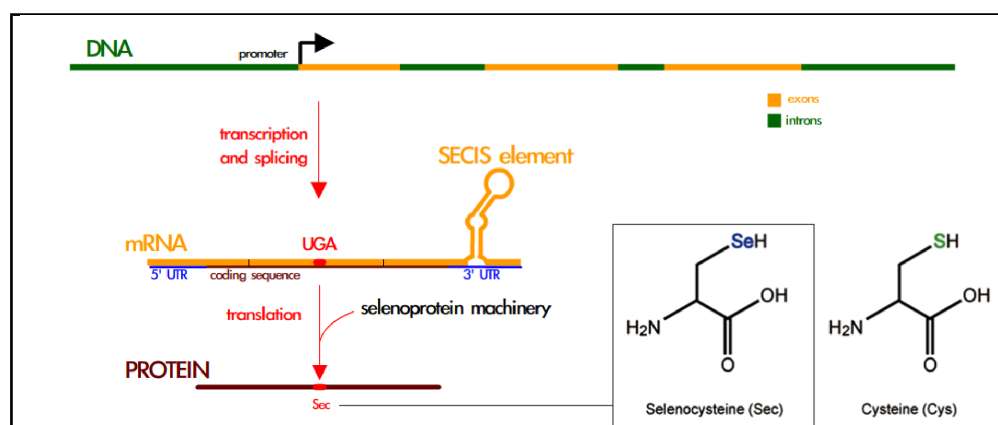


Figure 1.1: General schema of an eukaryotic selenoprotein gene. The structure of selenocysteine is shown in comparison with cysteine.

1.1.2 (Un)related to selenoproteins

In selenoproteins, selenium is contained in one (or sometimes few) selenocysteine residues, inserted during translation in specific positions of specific transcripts. Other selenocompounds are also present in cells, and some are found in proteins (e.g. selenomethionine, Se-methylselenocysteine [Whanger, 2002]). A marked difference with selenoproteins is that, in all these cases, Se-containing amino acids are inserted non-specifically, scattered through the proteome, and depending on the concentration of selenium. This phenomenon can be explained by the fact that selenium enters specifically sulfur pathways. In plants, non-specific selenocysteine is produced through cysteine pathways and inserted in proteins, and is thought to be a factor mediating selenium toxicity [Van Hoewyk, 2013]. Another important selenocompound is selenouridine (SeU), a Se-containing nucleotide used in the wobble position of specific tRNAs in some prokaryotes [Wittwer and Ching, 1989], altering codon specificity.

Also, selenocysteine is often associated in literature with pyrrolysine, known as the 22nd amino acid. Pyrrolysine insertion also requires the recoding of a stop codon (UAG). It was identified in a very narrow number of proteins encoded in archaeal and bacterial genomes. In contrast to selenocysteine, to date no recoding signals have been identified for pyrrolysine, and the characterization of its pathway is still poor [Yuan et al., 2010].

This thesis is centered only on the genomics and functions of selenocysteine, as inserted in selenoproteins *sensu strictu* (excluding non-specific selenocompounds). Selenouridine will also make some appearances later, for it shares common pathways with Sec.

1.2 Sec machinery

A set of specific factors is required in a genome in order to express selenoproteins, for the production and insertion of Sec. First, we will review the known pathways in eukaryotic organisms, the main target of this work. Then, we will describe the differences in the bacterial and archaeal systems.

1.2.1 Eukaryotic Sec synthesis on its own tRNA

Unlike all other amino acids, Sec is synthesized on its own tRNA, in a process that resembles tRNA-dependent synthesis of cysteine, glutamine and asparagine in prokaryotes [Sheppard et al., 2008]. tRNA^{Sec} has a very peculiar structure, with a long extra arm (figure 1.2). Similar, but shorter arms are found in certain tRNAs for serine, leucine, tyrosine [Itoh et al., 2009].

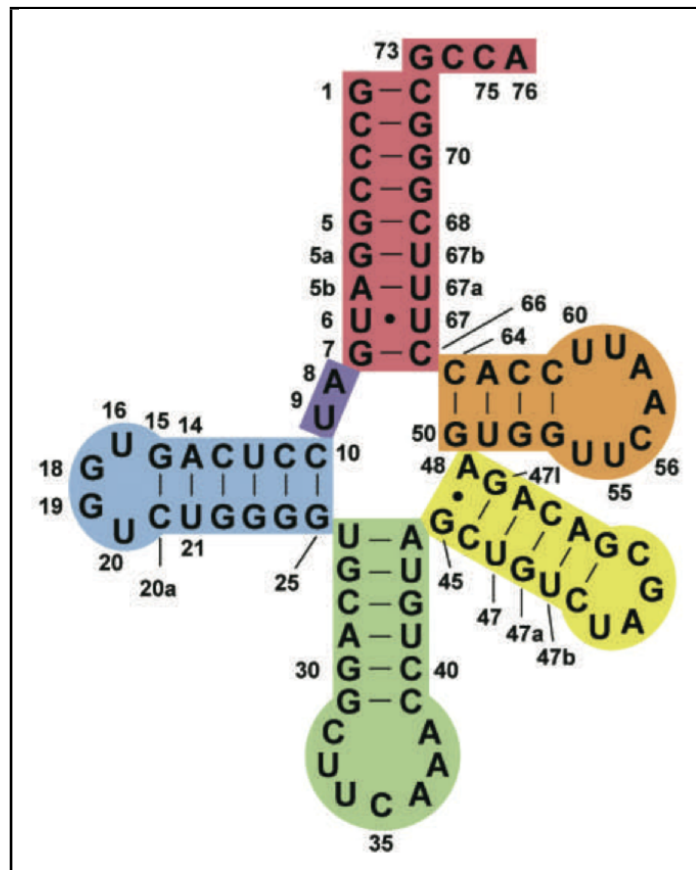


Figure 1.2: Human tRNA^{Sec} in cloverleaf model, adapted from [Itoh et al., 2009]

tRNA^{Sec} (also called SelC in prokaryotes) is recognized by the standard seryl-tRNA synthetase (SerRS), and it is initially charged with serine. Then, serine is converted to selenocysteine in two steps (see figure 1.3). First, it is activated

through phosphorylation by PSTK (PhosphoSeryl-tRNA[Ser]Sec Kinase) [Carlson et al., 2004]. Then, it is processed by protein Selenocysteine Synthase (SecS or SepSecS or SLA/LP, called SelA in prokaryotes). SecS catalyzes the conversion of the phosphoseryl moiety into selenocysteiny group, using selenophosphate as selenium donor [Palioura et al., 2009]. Selenophosphate is produced from selenide by Selenophosphate Synthetase enzymes (SPS, called SelD in prokaryotes). This family has the unique characteristic of being part of the Sec machinery, and a selenoprotein family itself. In fact, the gene responsible for the production of selenophosphate in human and fruit fly (SPS2) contains selenocysteine on a N-terminal domain, believed to bind the selenide and deliver it to the catalytic site. A second gene belonging to this family was identified in both human and drosophila (SPS1). This does not carry selenocysteine, and has been proposed to have a molecular function distinct from selenophosphate synthesis. A large section of this thesis is dedicated to the phylogeny of the SPS family, and covers also the origin and possible functions of SPS1 proteins.

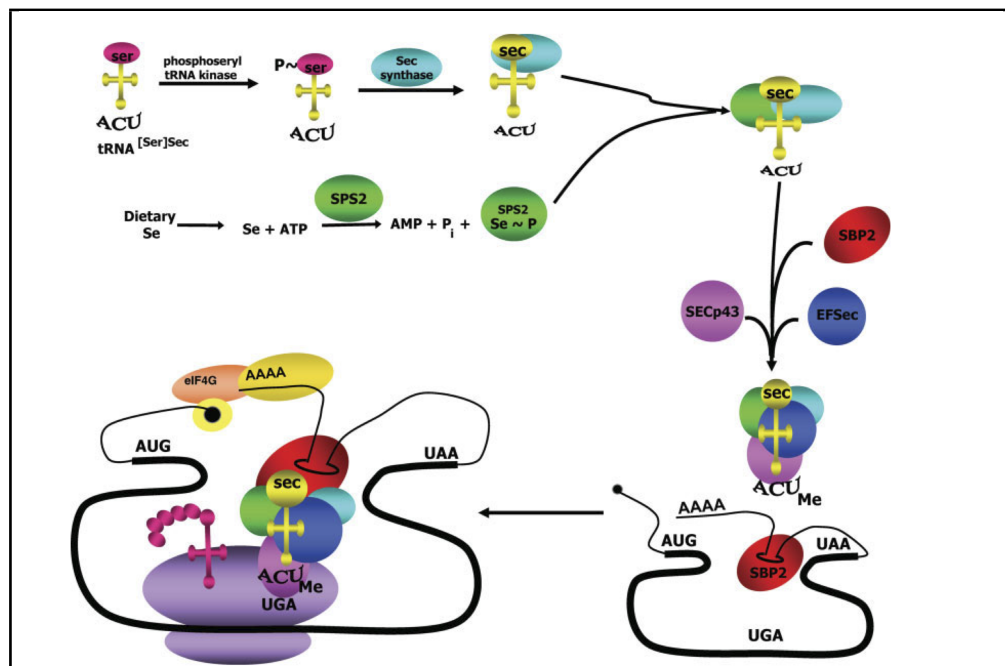


Figure 1.3: The mechanism of selenocysteine synthesis and insertion in eukaryotes. From [Squires and Berry, 2008].

1.2.2 Eukaryotic Sec insertion: tweaking translation

Insertion of Sec in selenoproteins occurs co-translationally, and it is mostly identical to a normal elongation step during translation. The recoding of UGA (selenocysteine insertion instead of translation termination) is obtained through a few Sec

specific factors. The protein eEFsec (eukaryotic Elongation Factor for selenocysteine, called SelB in prokaryotes) is the key to tweak the ribosomal machinery. On its N-terminal domain, eEFsec carries a domain very similar to EF-tu, a ubiquitous elongation factor involved in the deliver of charged tRNAs to the site A of the ribosome, allowing peptide bond formation and thus elongation. eEFsec performs a function analog to EF-tu, but it is used uniquely when a Sec UGA is read by the ribosome. In fact, the SECIS element in the 3'UTR of selenoprotein transcripts is recognized by protein SBP2 (Selenocysteine Binding Protein 2). SBP2 then recruits tRNA^{Sec} in complex with eEFsec [Tujebajeva et al., 2000]. Protein SECp43 is also proposed to form part of the complex, as it has been shown to bind tRNA^{Sec} [Ding and Grabowski, 1999] and also eEFsec [Small-Howard et al., 2006]. Importantly, SECp43 was also shown to be required for tRNA^{Sec} methylation in the wobble position [Xu et al., 2005]. Other RNA binding proteins are sometimes listed as Sec machinery, although their function appears to be not limited to the Sec pathway. Ribosomal protein L30 has been proposed to bind SECIS element in competition with SBP2, possibly to disassociate the complex and allow the completion of Sec decoding [Chavatte et al., 2005]. Protein nucleolin was also identified to bind SECIS elements [Wu et al., 2000], although its functional role is still unclear.

1.2.3 The bacterial Sec machinery

The bacterial system for Sec synthesis is essentially analog to the eukaryotic one (see figure 1.4). The major difference is that the tRNA^{Sec} charged with serine is read by SelA (selenocysteine synthase) without prior activation by phosphorylation. Thus, there is no PSTK protein. Eukaryotic and bacterial selenocysteine synthase (SecS and SelA) are both type-I pyridoxal 5-phosphate (PLP)-dependent enzymes. Despite catalyzing extremely similar reactions, their sequence and also their structure are very different [Itoh et al., 2013], casting doubts on them being phylogenetically related.

The Sec insertion process exhibits more differences with eukaryotes. The structure of the SECIS element is radically different, as we will see later. Also, its position is different: in the 3'UTR in eukaryotes, within the coding sequence just downstream of the Sec UGA in bacteria. The Sec-specific elongation factor SelB performs the functions of both eukaryotic eEFsec and SBP2. In fact, its C-terminal domain recognizes bacterial SECIS elements, while the N-terminal again works as elongation factor. There is no SBP2 protein, and also no SECp43 in bacteria.

1.2.4 The archaeal Sec machinery

The Sec production system appears very similar in archaea and eukaryotes (see figure 1.5). The serine charged on tRNA^{Sec} is phosphorylated by a protein homologous to eukaryotic PSTK. The selenocysteine synthase (here named SepSecS) is more similar to SecS than to SelA [Stock and Rother, 2009]. Instead, the Sec in-

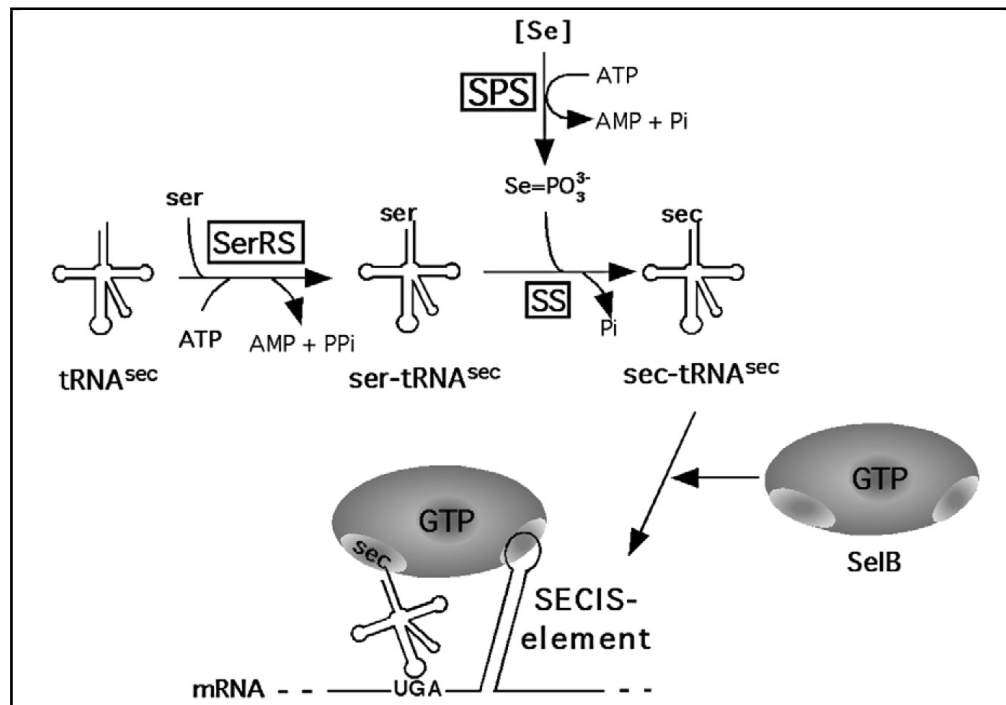


Figure 1.4: Sec synthesis and insertion in bacteria (*E.coli*). From [Stock and Rother, 2009].

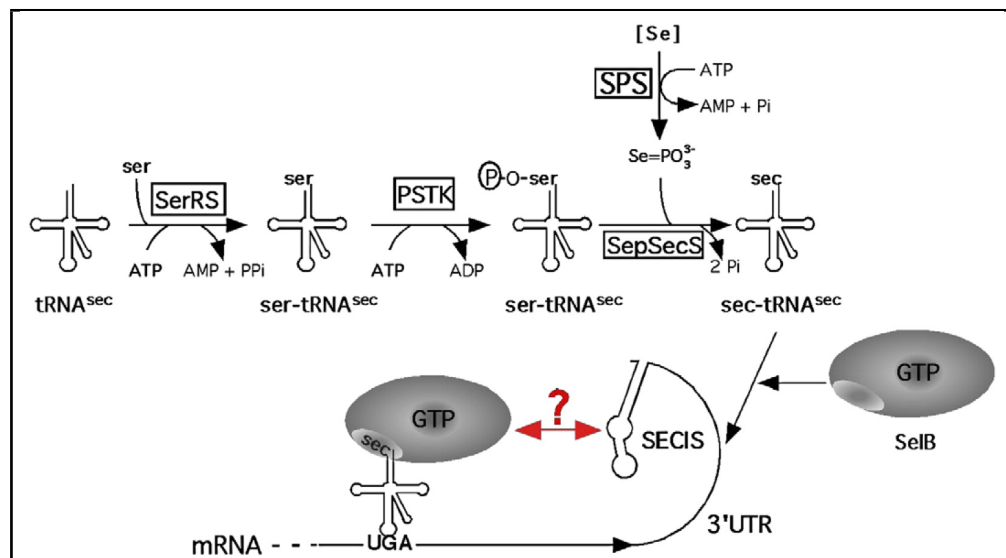


Figure 1.5: Sec synthesis and insertion in archaea. From [Stock and Rother, 2009].

sersion system has again peculiar features. Archaeal SECIS elements are different in sequence and structure from their counterpart in both other kingdoms, as we will see in the next section. Like eukaryotic SECIS, they generally reside in the 3'UTR of selenoprotein transcripts, although for one gene (*fdhA* of *M.jannaschii*) it was found in the 5'UTR instead [Wilting et al., 1997a]. An archaeal Sec-specific elongation factor (SelB) was identified and characterized. Unlike bacteria, the archaeal SelB appears not to be binding SECIS elements [Rother et al., 2000]. Nonetheless, no SBP2 homologue could be found in archaeal genomes, and no other protein dedicated to SECIS binding has been identified so far. Thus, the question of how the SECIS and the site of translation (SelB/ribosome) communicate remains open [Stock and Rother, 2009].

1.2.5 SECIS elements

SECIS elements are the principal signal for UGA to Sec recoding. SECIS stands for selenocysteine insertion sequences. In this work, we use the term SECIS alone to designate eukaryotic SECIS elements, in contrast to terms bSECIS and aSECIS for bacterial and archaea respectively. SECIS elements are stem loop structures containing two non-Watson-Crick AG pairs at their core (quartet), forming a peculiar RNA motif known as Kink-turn [Latrèche et al., 2009]. Two types of SECIS have been described [Grundner-Culemann et al., 1999] (see figure 1.6), with form II possessing an extra short stem on top (helix 3). Apart from this, they have the same topology, with two helices separated by a loop, and the quartet found at the base of helix 2. Although type II SECIS is more abundant [Krol, 2002], the two forms appear to be functionally equivalent. There has been a conspicuous effort to identify the SECIS nucleotides important for SBP2 binding, both experimentally and computationally (see a review in [Krol, 2002], and the most recent works [Latrèche et al., 2009] and [Chapple et al., 2009]). It resulted that the RNA structure itself, rather than the sequence, is important for function.

In fact, the conserved features are mostly base pairings, with specific length constraints on the helices formed. Only a few parts show conservation at the primary sequence level. The most conserved region is the quartet, with the invariant non-canonical AG pairs, and also the surrounding bases showing a strong composition bias. This is consistent with SBP2 footprinting experiments, mapping its binding to this region [Fletcher et al., 2001]. Additional conserved unpaired nucleotides are found at the 5' end of the second loop (apical loop in type I, internal loop 2 in type II). Higher eukaryotes possess almost invariably a stretch of 2 or 3 adenines here. The SelM and SelO SECIS constitute a notable exception, carrying cytosines instead. SECIS elements are found in the 3'UTR of selenoprotein transcripts. The distance between the Sec-UGA and the SECIS element varies substantially, with the reported maximum in mammals being ~5200 nt (DI2). The minimal distance to allow Sec insertion was tested in human embryonic kidney line 293 cells for the DI1 gene [Martin et al., 1996], and it was found to be ~50/110 nt.

Bacterial SECIS elements (bSECIS, figure 1.7) are instead located within the

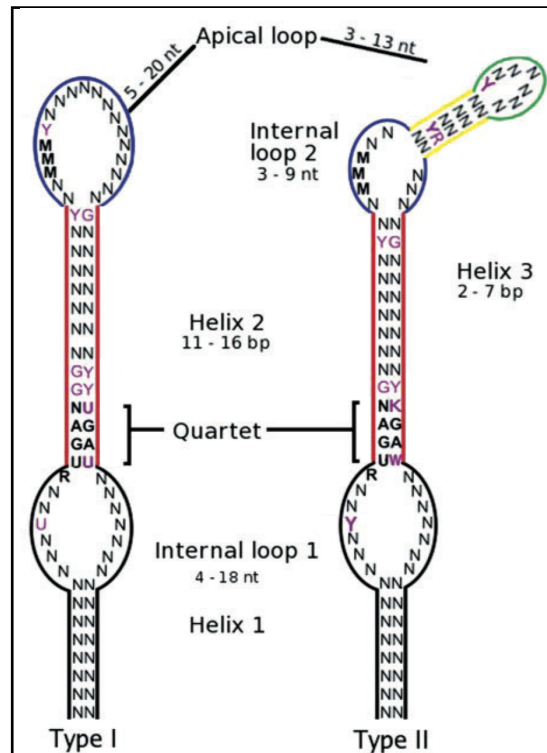


Figure 1.6: Model structures for eukaryotic SECIS elements, adapted from [Chapple et al., 2009]. Letters different from N (any nucleotide) indicate conserved positions. Ambiguous nucleotide codes are used: Y=U/C, K=G/U, W=A/U, R=A/G, M=A/C. Nucleotides in magenta were identified for the first time in [Chapple et al., 2009].

coding sequence, and are characterized by a large stem that includes the Sec-UGA. Mostly, they have been characterized in *E.coli*. Here, the bSECIS has been partitioned in two putative domains: the first includes the Sec-UGA, and serves to prevent the binding of termination factor 2. The second domain includes the apical loop, shown to be recognized by SelB [Krol, 2002]. Looking across species, bacterial SECIS elements exhibit very poor sequence identity, and also a high amount of structural variation [Zhang and Gladyshev, 2005]. It is plausible that such lineage-specific characteristics make certain bSECIS not transferable between bacterial species, for they co-evolved with the Sec machinery.

Archaeal SECIS elements (aSECIS, figure 1.8) were characterized mostly in *M.jannaschii*. Here, six out of seven aSECIS were located in the 3'UTR, while the aSECIS of *fdhA* was found in the 5'UTR. Despite displaying high variation in sequence and also stem length, all these aSECIS elements possess a very conserved motif, containing a purine-only loop followed by three consecutive CG pairs.

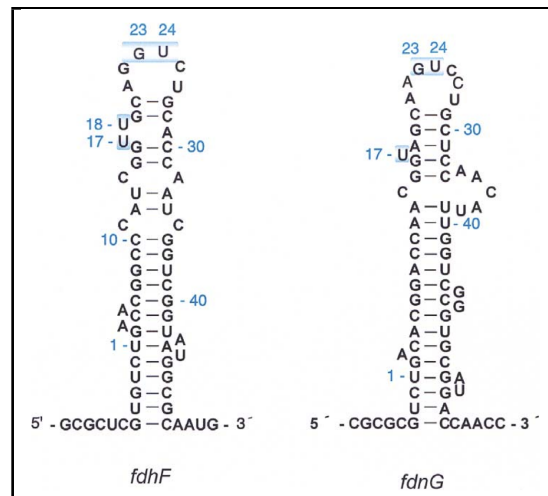


Figure 1.7: Model structures for bSECIS elements (bacterial) found in two formate dehydrogenase genes of *E.coli*. The loop nucleotides interacting with SelB are highlighted in blue. The numbering starts from the Sec UGA. Figure from [Krol, 2002], after [Hüttenhofer et al., 1996].

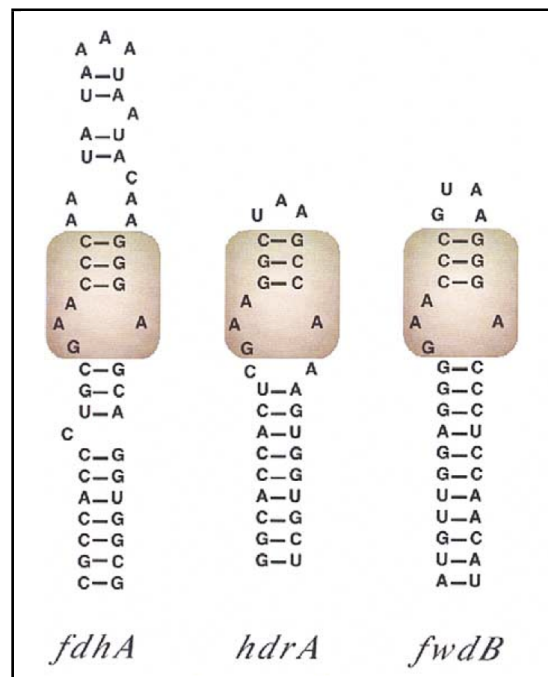


Figure 1.8: Model for aSECIS elements (archaeal) found in the *M.jannaschii* genes for formate dehydrogenase (*fdhA*), heterodisulfide reductase (*hdrA*) and formyl-methanofurane-dehydrogenase (*fwdB*). The conserved motifs are boxed. From [Krol, 2002], after [Wilting et al., 1997b].

1.3 Selenoprotein genomics

1.3.1 Selenoproteinless organisms and cysteine homologues

Although selenoproteins are spread across all kingdoms of life, they are not present in all organisms. Actually, the selenocysteine coding trait was found only in a minority (~14/20%) of investigated prokaryotes [Kryukov and Gladyshev, 2004]. Many eukaryotic lineages are also devoid of selenocysteine, most notably Fungi (including yeast, *S.cerevisiae*) and plants (except green algae) [Lobanov et al., 2009]. As we will see in detail later, many insects are also selenoproteinless. All these genomes not only lack selenoprotein genes, but also Sec machinery is missing or incomplete. Still, there are regular genes that have a selenoprotein orthologue in other species. Typically, non-Sec homologues of selenoproteins carry a cysteine (Cys) aligned to the Sec position, which reflects the functional similarity of the two amino acids. Cysteine homologues are known for the great majority of selenoprotein families [Fomenko et al., 2007; Fomenko and Gladyshev, 2012]. Naturally, Cys homologues are present also in species that are able to code selenocysteine, both as paralogues, or orthologues to selenoprotein genes in other organisms. We use the term selenoprotein family to indicate a group of homologous proteins, presumably with the same structural fold, that include selenoproteins and cysteine homologues. If we partition virtually the comprehensive proteome of all living organisms in protein families, only a very small fraction contain selenoproteins. In other words, selenocysteine is advantageous in very few of all possible cysteine sites. Interestingly, for many selenoprotein families the Sec forms exhibit a scattered distribution in the species tree (see for example SelU in figure 1.9). This has been taken as an indication of a dynamic process acting on selenocysteines, with many known events of selenocysteine to cysteine conversion. The two cysteine codons in the standard genetic code (UGU, UGC) are just one point mutation from the Sec UGA. Cysteine to selenocysteine conversions has been also theorized in bacteria [Zhang et al., 2006].

1.3.2 Selenocysteine vs cysteine

Given the high sequence and structural similarity, there is no doubt that in the great majority of cases the overall molecular function of a cysteine homologue is the same of its selenoprotein counterpart. This opens the question of exchangeability of selenocysteine and cysteine. If the same molecular function can be obtained with a cysteine, why use a selenocysteine? To justify the use of such a complex system like the Sec machinery, and the conservation of the Sec UGA codons against Cys conversion drift, there has to be some advantage in Sec over Cys. Selenocysteine is found generally in a single residue per protein, in a catalytic site. Most selenoproteins families are thiol oxidoreductases acting as anti-oxidants [Fomenko and Gladyshev, 2012], where selenocysteine replaces one of the cysteines in their redox domains (often CxxC, known as redox box). The canonical form of Cys

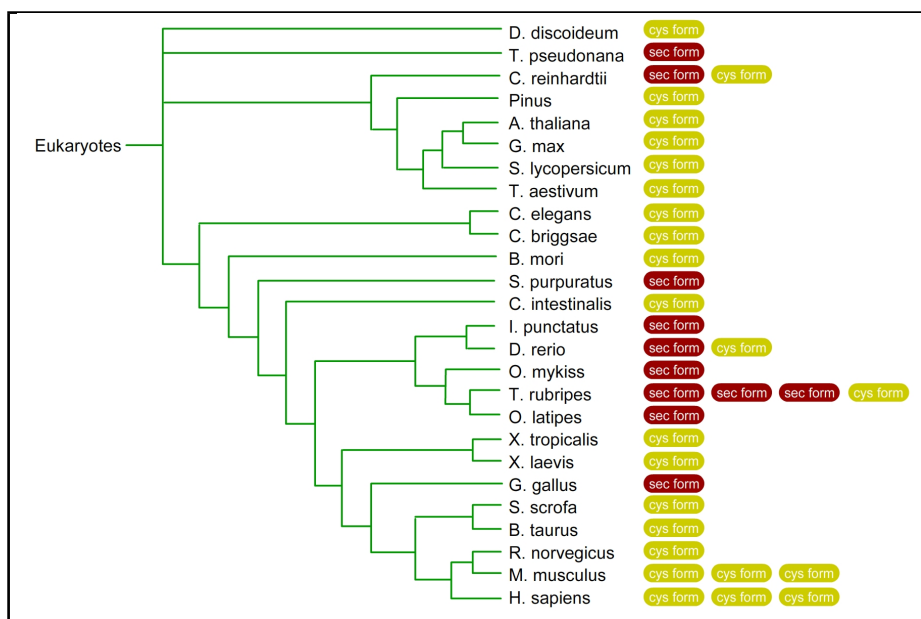


Figure 1.9: Scattered distribution of Sec forms in the SelU family. Data from [Castellano et al., 2004]

(namely its reduced form) exposes a thiol group as side chain. When oxidized, the thiol groups of two cysteines form disulfide bonds, which can be intra- or inter-molecular. In most selenoproteins, Sec acts analogously, reacting with a cysteine to form a selenenyl-sulfide bond. Often, the target is another oxidoreductase, producing a cascade of electrons downward the redox potential. The cell utilizes these processes to defend itself from oxidative damage: when strong oxidants (such as reactive oxygen species, ROS) are present, they would react with pretty much anything in the cell, altering proteins and nucleic acids. Anti-oxidant proteins are strong reductants that intercept and channel the flow of electrons, thus avoiding uncontrolled oxidation. The redox function of selenoproteins is well understandable when considering the chemical properties of Sec and Cys. In fact selenocysteine has both a lower pKa and a higher reduction potential than cysteine, which makes it very suitable for redox functions, and particularly for anti-oxidant activity. Nonetheless, the higher reactivity of Sec makes it also potentially dangerous for cells. This may explain why selenocysteine is not constitutive in the genetic code, and also is not present in free form in the cell, in contrast to cysteine and to all other standard amino acids.

Some researchers sought insights on the exchangeability of Sec and Cys using artificial mutants. When the mammalian selenoenzyme thioredoxin reductase has its selenocysteine replaced by a cysteine, its catalytic activity is dramatically reduced, and its optimum pH increases [Zhong and Holmgren, 2000]. The drosophila thioredoxin reductase is a natural cysteine homologue. Several artificially variants

of this enzyme were generated, including Cys to Sec mutants. Other amino acid changes near the catalytic site were also explored [Gromer et al., 2003]. In that study, Cys to Sec mutants exhibited higher or comparable catalytic activities. It must be stressed that artificial Sec/Cys conversions do not account for changes that would be accommodated in time by natural selection after the conversion. Thus, they reflect the fitness of mutants just one step away from the observed wild-type enzymes. It is not surprising, then, that the extant thioredoxin reductases encoded by mammals (with Sec) and drosophila (Cys) carry activities of the same order of magnitude, although Sec to Cys conversion of the mammalian enzyme showed a much more dramatic reduction [Zhong and Holmgren, 2000]. In contrast, a Cys to Sec conversion in the drosophila enzyme increased the catalytic activity even without any other accommodating mutation. This can be taken as indication of essential superiority of Sec over Cys in this redox site. The advantage of Sec over Cys is obviously restricted to a very limited number of proteins. Also, we must remember the complexity of natural selection: catalytic efficiency is not always a major determinant of the fitness of a protein. Other factors may be more important in many cases, such as substrate specificity or promiscuity, or just regulation. Even reduced activity may be advantageous in certain situations. As we will see in the next paragraphs, and then extensively in the rest of this thesis, the Sec/Cys exchangeability varies a lot in different organisms, and also for different selenoproteins. Thus, this subject should be approached keeping in mind that every gene may have its own story in regard. It is then useful to give an overview of the known selenoprotein families.

1.3.3 Known selenoprotein families

Most selenoprotein families are thiol oxidoreductases in which Sec replaces a catalytic cysteine. Many of these enzymes belong to a single superfamily, thioredoxin-like, and operate mostly in the **anti-oxidant defense**. Thioredoxins (Trx) are small oxidoreductase proteins found in all living organisms. They are characterized by a core of four-stranded, antiparallel beta sheets, located between three alpha helices. Their catalytic site carries an active redox box (CXXC), which switches between two redox states (thiols/disulfide). Several Sec-containing thioredoxins have been identified in prokaryotes [Zhang and Gladyshev, 2008] and eukaryotes [Lobanov et al., 2009]. As reductants, thioredoxins constitute a defense against oxidative stress, but also serve as electron donors to other redox reactions. Glutathione (GSH) is a tripeptide with analogous (but distinct) activities. In cells, dedicated pathways operate to maintain a reduced pool of thioredoxin and glutathione. These pathways, together with enzymes using these molecules as donors, constitute the thioredoxin and glutathione systems. Many of the proteins in both the Trx and GSH systems are selenoprotein families. This is the case of two among the largest and most studied selenoenzymes: TR and GPx, both characterized by a thioredoxin-like fold. Thioredoxin reductases (TR or TrxR) are large flavoproteins, the only responsible for the reduction of thioredoxins in cells, at the expenses of

NADPH. They are present in most living organisms. In vertebrates and in several other eukaryotes, TR are essential selenoproteins carrying Sec at their C-terminus. Glutathione peroxidases (GPx) are another large selenoprotein family with many paralogues in human. GPx catalyze the reduction of dangerous peroxides (such as H_2O_2) using glutathione as electron donor. In mammals, glutathione is reduced by enzyme glutathione reductase (GR or GSR), never observed as selenoprotein. Glutaredoxins (Grx) are peculiar thioredoxin-like oxidoreductases, active in the anti-oxidant defense and important also for many other functions (e.g. deoxyribonucleotides synthesis). Grx are reduced non-enzymatically by glutathione, in contrast to thioredoxins which are dependent on TR. This family has been found as selenoprotein in a limited number of prokaryotes [Kryukov and Gladyshev, 2004]. Also, glutaredoxin domains are found in several other selenoenzymes. Peroxiredoxins (Prx) are another class of oxidoreductases. They catalyze the reduction of peroxides at expenses of thioredoxin, thus they are also known as thioredoxin peroxidases. Several Prx-like selenoprotein families were found in prokaryotes [Zhang et al., 2006; Zhang and Gladyshev, 2008] and also in green algae [Palenik et al., 2007]. Recently, the peroxiredoxin family of alkyl hydroperoxide reductases (AhpC or TSA, for thiol-specific antioxidant) was found also in a sponge genome [Jiang et al., 2012]. Methionine sulfoxide reductases (Msr) are other enzymes active in the anti-oxidant defense. When ROS are present in the cell, they tend to oxidize proteins unspecifically, and their targets are typically the two amino acids most susceptible to oxidation, cysteine and methionine. The oxidation of methionine lead to methionine sulfoxide, in a racemic mixture. Two different classes of Msr are specialized for the two stereoisomers: MsrA reduces methionine-R-sulfoxide residues in proteins, while MsrB (also called SelR or SelX) reduces the S form, both as free amino acid and inserted in proteins [Lee et al., 2009a]. MsrA and MsrB are radically different in sequence and structure, and probably are phylogenetically unrelated. MsrA was found as selenoprotein in bacteria and in several eukaryotes, including green algae, cnidaria, sea urchins, and even arthropods [Kim et al., 2006]. MsrB was found as selenoprotein in mammals, but not in prokaryotes or insects [Kryukov et al., 2002]. Human has three copies of MsrB, one of which is a selenoprotein (MsrB1). Recently, a very unusual MsrB selenoprotein containing 4 Sec residues and two SECIS has been characterized [Lee et al., 2011a].

The role of selenoproteins and of the thioredoxin-like fold is not limited to redox protection and homeostasis. In prokaryotes, many selenoproteins work in **electron transfer / energy metabolism** pathways. The alpha subunit of formate dehydrogenase (FDH, FdhA) is one of the most common selenoprotein in prokaryotes [Zhang et al., 2006]. FDH catalyzes the reversible oxidation of formate to CO_2 , normally using $NADP^+$ as electron acceptor (although others have been observed, such as ferredoxin). FDH is involved in a high number of processes, including acetogenesis and methanogenesis [Stock and Rother, 2009]. A Sec-containing Split-Soret cytochrome C protein was characterized in anaerobic bacteria [Kim et al., 2009]. Other selenoproteins belonging to the cytochrome C1 family were also observed. Then, the bacterial operon Rnf encodes for a membrane bound

complex dedicated to electron transport to nitrogenase. Subunits RnfB and RnfC are NADH:ubiquinone oxidoreductases that have been observed as prokaryotic selenoproteins. The protein DsrE (named after the dissimilatory sulphite reductase bacterial operon) is a sulfurtransferase possibly involved in electron transport, identified as selenoprotein in a few bacterial species [Zhang et al., 2006]. Other, uncharacterized sulfurtransferases were also observed [Zhang and Gladyshev, 2008, 2005]. Also, a not better characterized NADH oxidase has also been identified [Zhang et al., 2006]. Many of the proteins involved in electron transport contain Fe-S clusters (e.g. cytochrome C, but also RnfB and RnfC). Some belong to the family of ferredoxins, small proteins that act as electron capacitors, using a redox switch made with iron-sulfur clusters. Ferredoxins are typically used in electron transport, including respiration and photosynthesis. Several selenoproteins were observed in the ferredoxin pathways. HesB protein for example is involved in the biosynthesis of Fe-S clusters, and was detected as selenoprotein in some bacterial and archaeal species [Zhang and Gladyshev, 2008; Stock and Rother, 2009]. A few Fe-S oxidoreductase selenoenzymes (e.g. GlpC [Zhang and Gladyshev, 2008]) were identified in prokaryotes, including a radical SAM domain selenoprotein [Zhang and Gladyshev, 2005]. The ferredoxin thioredoxin reductase (FTR, Frx, Ftrb) is an interesting enzyme that links photosynthesis to the thioredoxin system. It catalyzes the reduction of Trx proteins using light generated electrons. In [Zhang and Gladyshev, 2008], FTR selenoproteins were identified in oceanic samples; selenocysteine is located in the enzymatic site that reduces Trx. Ferredoxins play an important role also in methanogenesis, where many enzymes have been observed also as selenoprotein: besides aforementioned formate dehydrogenase, we have formyl-methanofuran dehydrogenase (FMD), F₄₂₀-reducing hydrogenase (alpha subunit FruA or FrhA; delta subunit: FruD or FrhD), F₄₂₀-non-reducing hydrogenase (VhuD, VhuU), heterodisulfide reductase (HdrA). Other selenoenzymes are reductases utilized in acetogenesis pathways (see [Stock and Rother, 2009]). This include glycine reductase (GrdA, GrdB), proline reductase (PrdB), sarcosine reductase (GrdH), betaine reductase (GrdF). Then, the bacterial gene UshA codes for an hydrolase that converts UDP-glucose to glucose-1-phosphate, which can enter several pathways (e.g. glycogenesis). Usha-like selenoproteins have been found in prokaryotes [Zhang and Gladyshev, 2008]. The protein inosine monophosphate dehydrogenase (IMPD) instead converts Inosine 5'-phosphate to xanthosine 5'-phosphate, using NAD⁺ as electron acceptor. This is an important step in the *de novo* synthesis of guanine nucleotides, and plays a role in cell growth. IMPD selenoproteins were found in prokaryotes [Zhang and Gladyshev, 2010].

A conspicuous number of selenoproteins are then found in the **oxidative protein folding** pathways. Many are protein disulfide isomerases (PDI), catalyzing the formation and breakage of disulfide bonds in substrate proteins, typically for correct protein folding. This includes the prokaryotic selenoproteins DsbA and DsbG [Zhang et al., 2006]. PDI selenoenzymes were also identified in eukaryotes: in green algae [Lobanov et al., 2007], in coccolithophores [Obata and Shiraiwa, 2005] and also in some chordates [Jiang et al., 2010]. The eukaryotic selenopro-

tein superfamily including Sel15, Fep15, and SelM localize to the endoplasmatic reticulum (ER), and are proposed to have a role in the control of the correct folding of proteins [Gromer et al., 2005]. Selenoproteins SelK (SelG in flies) and SelS are likely to work in the related pathway of ER-associated degradation (ERAD), which targets misfolded proteins and signals them to the proteasome for disposal [Shchedrina et al., 2011].

Detoxification and transport are other cellular processes in which we find many selenoproteins. In particular, some prokaryotic selenoenzymes were found in the arsenic detoxification pathway: arsenate reductase (ArsC) and arsenite S-adenosylmethyltransferase (ArsS) [Zhang and Gladyshev, 2008], the latter with a Sec-homologue also in green algae and diatoms [Lobanov et al., 2007]. Some putative mercuric transport selenoproteins have also been described in prokaryotes (e.g. MerP). Glutathione S-transferases (GST) are another class of enzymes involved in protein detoxification and transport. They utilize reduced glutathione, which they conjugate to a variety of compounds to make them more soluble, and thus easier to dispose. GST selenoproteins have been observed in prokaryotic oceanic samples [Zhang and Gladyshev, 2008]. Some prokaryotic rhodanese-related (rhor) selenoproteins have also been observed [Zhang and Gladyshev, 2005]. Rhodanese is a sulfurtransferase that detoxifies cyanide by converting to thiocyanate. Nonetheless, no clear function has been assigned to these similar prokaryotic selenoenzymes. Vertebrate selenoprotein P (SelP or SePP1) is a very singular selenoprotein, for it contains multiple Sec residues (10 in human). Two SECIS at the 3'UTR direct their incorporation. SelP is a secreted glycoprotein abundant in plasma, whose main function is believed to be the transport and storage of selenium in form of inserted selenocysteines [Gromer et al., 2005]. Its N-terminal possesses a thioredoxin-like domain that includes its first Sec, and for this reason it is also proposed to have an anti-oxidant role in plasma.

Many selenoproteins (including thioredoxin-fold selenoenzymes and other oxidoreductases) are involved in **other functions** not yet mentioned, or are functionally uncharacterized. Iodothyronine deiodinases (DI or DIO) for example are thioredoxin-fold selenoenzymes responsible in vertebrates for the activation and deactivation of the thyroid hormones, important metabolism regulators. Although the thyroid gland is present only in vertebrates, DI-like selenoproteins have been identified also in prokaryotes [Zhang and Gladyshev, 2008], where their function remains unknown. The selenoprotein family of selenophosphate synthetases (SPS, SelD) was already mentioned, as it is part of the selenocysteine machinery. Sec-containing SPS were found in bacteria [Kryukov and Gladyshev, 2004], archaea [Stock and Rother, 2009] and many animals, including drosophila and human [Lobanov et al., 2009]. Selenoprotein J (SelJ) is a very peculiar selenoprotein, detected in a few animal genomes [Castellano et al., 2005]. For its similarity with jellyfish J1-crystallins and preferential expression in the eye lens, it is proposed to have a structural role, which would make it unique among characterized selenoproteins. Selenoprotein I (SelI) is among the few selenoproteins with no known cysteine homologues. SelI contains a CDP-alcohol phosphatidyltransferase domain.

The activity of human SelI was characterized as ethanolamine phosphotransferase [Horibata and Hirabayashi, 2007], although we believe that the experiments had flaws potentially mining their conclusions, as explained later in results [Mariotti et al., 2012]. Selenoprotein N (SelN) is a vertebrate glycoprotein localized to the ER. Its molecular function is unknown, but from its expression pattern it is predicted to be involved in early development, and in proliferation and regeneration in striated muscles. Mutations in the human SelN gene lead to rare myopathies, most notably rigid spine muscular dystrophy [Gromer et al., 2005]. Selenoproteins H, T, W and V (SelH, SelT, SelW, SelV) constitute a thioredoxin-like superfamily with Sec containing genes in mammals. They all contain a conserved redox box and thus are assumed (or were shown) to act as oxidoreductases [Dikiy et al., 2007], although their precise molecular function is yet unclear. Probably the most ancestral member of the family is SelW, since similar selenoproteins were observed in prokaryotes [Zhang et al., 2006]. Selenoprotein L (SelL) is another Sec-containing oxidoreductase, with unknown molecular function and distribution apparently limited to some animals. Its peculiarity is the presence of two Sec residues in a Sec-only redox box (UXXU), forming a rare diselenide bond [Shchedrina et al., 2007]. Selenoprotein O (SelO) is a large human selenoprotein with cysteine homologues in prokaryotes. Although its function remains experimentally uncharacterized, it was recently proposed as non-canonical protein kinase, for its similarity with this class of proteins [Dudkiewicz et al., 2012]. An uncharacterized membrane-bound selenoprotein (MSP) was identified in some eukaryotes, including green algae and slime molds [Lobanov et al., 2007]. Additionally, selenoproteins with a restricted phylogenetic distribution were predicted in prokaryotes [Zhang and Gladyshev, 2008] and in unicellular eukaryotes including green algae [Lobanov et al., 2007], *Plasmodium* [Lobanov et al., 2006a], *Leishmania* [Cassago et al., 2006], *Trypanosoma* [Lobanov et al., 2006b] and *Toxoplasma* [Novoselov et al., 2007]. These selenoproteins have no annotated homologue, and their functions are unknown.

1.3.4 Selenoproteins in vertebrates

The full human selenoproteome consists of 25 Sec containing proteins (figure 1.10), and it was presented for the first time in 2003 [Kryukov et al., 2003], when the genes known from previous studies were flanked by 7 novel genes verified experimentally. Three families together constitute almost a half of human selenoproteins: iodothyronine deiodinases (DI or DIO), glutathione peroxidases (GPx), thioredoxin reductases (TR or TrxR). All human selenoproteins were found also in the mouse and rat genomes, with the only exception of GPx6, converted to cysteine homologue in rodents. Since 2003, no other mammalian selenoprotein have been discovered. Along with other hints, this suggests that the mammalian selenoproteome identified so far is complete. Four novel vertebrate selenoprotein families have instead been discovered in bony-fishes, basal vertebrates with rich selenoproteomes: SelU [Castellano et al., 2004], SelJ [Castellano et al., 2005], Fep15

Selenoprotein	Chromosomal location (number of exons)	Sec location in protein (length of protein)	Selenoprotein structure
15kDa	1p22.3 (5)	93 (162)	
DI1	1p32.3 (4)	126 (249)	
DI2	14q31.1 (2)	133 (265)	
DI3	14q32	144 (278)	
GPx1	3p21.31 (2)	47 (201)	
GPx2	14q23.3 (2)	40 (190)	
GPx3	5q33.1 (5)	73 (226)	
GPx4	19p13.3 (7)	73 (197)	
GPx6	6p22.1 (5)	73 (221)	
H	11q12.1 (4)	44 (122)	
I	2p23.3 (10)	387 (397)	
K	3p21.31 (5)	92 (94)	
M	22q12.2 (5)	48 (145)	
N	1p36.11 (12)	428 (556)	
O	22q13.33 (9)	667 (669)	
P	5p12 (4)	59, 300, 318, 330, 345, 352, 367, 369, 376, 378 (381)	
R	16p13.3 (4)	95 (116)	
S	15q26.3 (6)	188 (189)	
SPS2	-	60 (448)	
T	3q24 (6)	36 (182)	
TR1	12q23.3 (15)	498 (499)	
TR2	3q21.2 (16)	655 (656)	
TR3	22q11.21 (18)	522 (523)	
V	19q13.13 (6)	273 (346)	
W	19q13.32 (6)	13 (87)	

Figure 1.10: Human selenoprotein genes, from [Kryukov et al., 2003]. Selenoproteins newly identified in that study are highlighted. The protein structure is shown on the right, with red lines indicating Sec positions and blue areas indicating alpha helices downstream of Sec residues.

[Novoselov et al., 2006], SelL [Shchedrina et al., 2007].

In [Castellano et al., 2009], the question of exchangeability of Sec and Cys was addressed for vertebrates, using for the first time tools from evolutionary and population genetics theory. Selenoproteins were roughly predicted in the available genomes, and the ancestral vertebrate selenoproteome was deduced. Then, the number of observed Sec to Cys conversions was compared with estimates based on the assumptions of complete (neutral) or partial exchangeability. All simulations lead to the conclusion that there is a deficit of conversions in vertebrates, consistent with strong purifying selection on Sec sites against Cys mutations. Additionally, authors computed the correlation of inferred ancestral selenoproteome sizes with the estimated levels of molecular oxygen in the atmosphere in the corresponding geological periods. The correlation was not statistically significant, mining the hypothesis that Sec was depleted in time because more sensitive to oxidation by the increasing O₂ [Leinfelder et al., 1988; Jukes, 1990]. Finally, the genomic regions corresponding to selenoprotein genes were searched in public databases of variation in human populations. Interestingly, no variant at all was detected at Sec UGA sites. The emerging picture suggests that selenocysteine is very important to vertebrates. Our selenoprotein genes are subject to strong purifying selection against conversion to cysteine.

1.3.5 Hierarchical regulation of selenium supply

In mammals, useful insights into the regulation of selenoproteins came from experiments testing different levels of selenium supplementation in diets (see a review in [Schomburg and Schweizer, 2009]). It is known that when selenium is scarce, it is preferentially retained in certain tissues, mostly in testes, adrenals and brain. The existence of such hierarchical regulation is also evident when considering the expression of different selenoproteins, even within the same family: thus for example, GPx4 and GPx2 are expressed even in condition of low selenium, while this is not true for GPx1 and GPx3. Interestingly, the diverse response of different selenoprotein genes has been ascribed to differences in their SECIS elements [Schomburg and Schweizer, 2009; Bermanno et al., 1996]. Also, the hierarchy of expression approximately follows the physiological importance of each selenoprotein, measured by how drastic is the phenotypic effect of the KO. For example, GPx4 is essential for life, while GPx1 deficient mice are viable and healthy [Schomburg and Schweizer, 2009].

1.3.6 Sec extinctions in insects

The selenoproteome of *Drosophila melanogaster* is very different than human. There are only three selenoproteins in this genome: SPS2 (also part of the Sec machinery), SelH and SelK. When the first 12 drosophila genomes were sequenced [Drosophila-Consortium, 2007], it was noted that in species *D. willistoni* the SelH and SelK genes were cysteine homologues, and SPS2 was missing. Consistently,

other parts of the Sec machinery (for example tRNA^{sec}) were also missing. All other drosophila carried a complete machinery, and also possessed the same selenoproteins found in *D.melanogaster* (with two possible exceptions in *D.grimshawi* and *D.persimilis*, differing by one selenoprotein more and less respectively). Evidently, *D.willistoni* has lost the selenoproteins after the split with the rest of drosophila. This organism was the first selenoproteinless animal being discovered, and this fact changed the paradigm that selenocysteine is essential to metazoan life. After this, our group proceeded to the characterization of selenoproteins in all other sequenced insects [Chapple and Guigó, 2008] (figure 1.11). This resulted in the identification of other insects that lost selenocysteine: the red flour beetle *Tribolium castaneum* (Coleoptera), the silkworm *Bombyx mori* (Lepidoptera), parasitic wasp *Nasonia vitripennis* and honey bee *Apis mellifera* (both Hymenoptera). Basically, among holometabolic insects (Endopterygota) selenoproteins were identified only in Diptera (lineage including flies and mosquitoes). Thus, here the lack of selenoproteins actually resembles more a rule than an exception.

From the phylogenetic structure of selenoproteinless species, it is evident that their last common ancestor still possessed Sec, and multiple selenoprotein extinctions occurred independently in these lineages. All selenoproteinless insects appeared to lack a complete Sec machinery: tRNA^{sec} and eEF^{sec} were always lost. The rest of genes were retained in a scattered fashion, presumably because some had acquired also another function, unrelated to selenocysteine. Among animals, selenoprotein extinctions have been observed only in insects. The rest of investigated arthropods possess a richer selenoproteome, as already observed in [Chapple and Guigó, 2008] even without available genomes (only EST sequences). Considering these facts, it was hypothesized that some important change occurred at the root of insects to make selenoprotein more dispensable for this class of organisms. The nature of such “relaxation of selective constraints” remains speculative, but it may be related to the peculiarities of insect anti-oxidant systems. In fact, many differences with vertebrates exist in these pathways. The protein glutathione reductase (GR) is missing in *D.melanogaster*, although glutathione is evidently utilized. Glutathione reduction appears to be carried out by the thioredoxin system instead [Kanzok et al., 2001]. Two genes resembling the GPx family are present, with Cys aligned to the Sec position of the vertebrate GPx selenoenzymes. Nonetheless, it was shown experimentally that at least one of the two (Gtpx-1) use thioredoxin, rather than glutathione, as electron donor [Missirlis et al., 2003], and thus should be named thioredoxin peroxidase (peroxiredoxin) instead. All these changes (see [Corona and Robinson, 2006] for an overview) indicate that the redox systems mutated radically in insects. It is reasonable to think that this somehow reduced the importance of selenocysteine, allowing (or maybe favoring) the subsequent losses.

1.3.7 Nematodes: a minimal selenoproteome

The case of selenoproteins in nematodes is puzzling. A single selenoprotein was identified in the *C.elegans* genome, despite a plethora of genomic approaches were

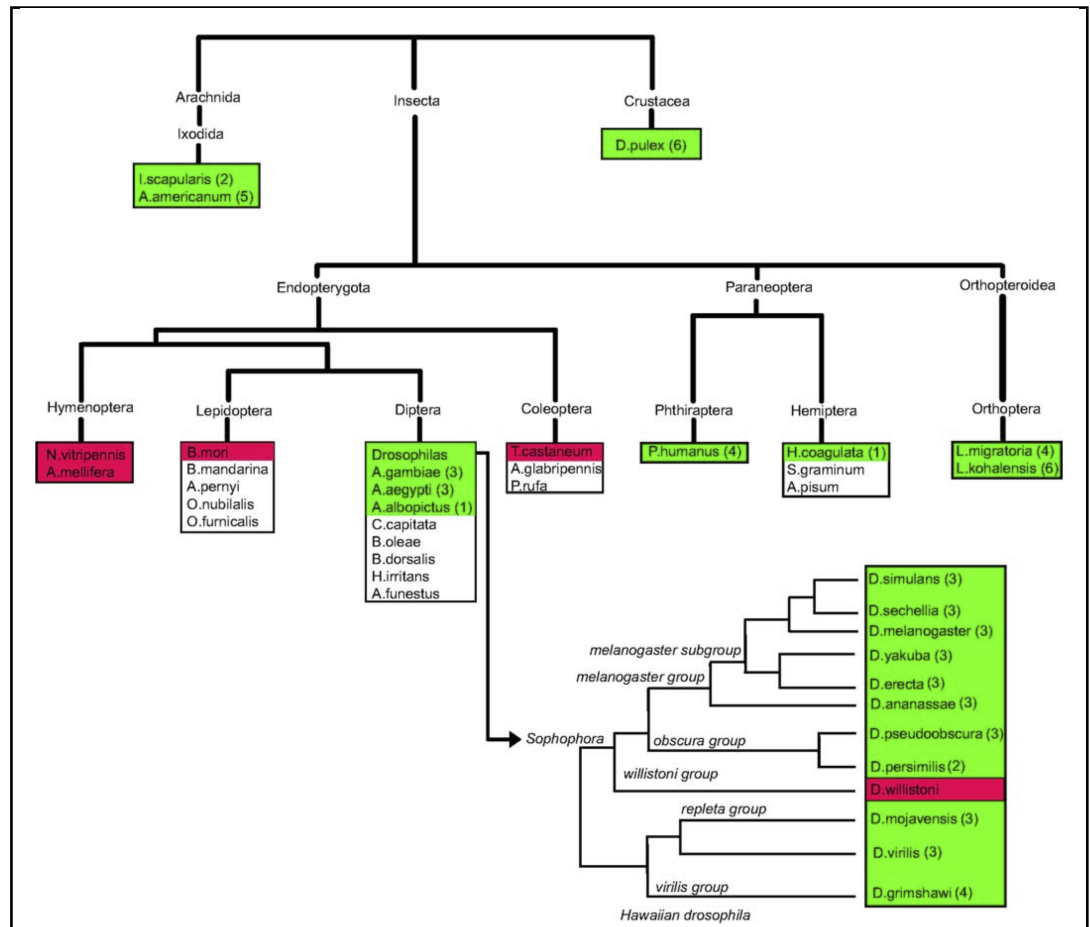


Figure 1.11: Selenoprotein extinctions identified in insects, from [Chapple and Guigó, 2008]. Species with selenoproteins are in green, the others are in red. Next to each species, the number of selenoprotein genes identified is indicated in parenthesis.

applied [Taskov et al., 2005]. The Sec machinery is conserved here to insert only one selenocysteine, in thioredoxin reductase protein TrxR1. Recently, functional characterization of TrxR1 showed that this gene is dispensable for growth, development and molting [Stenvall et al., 2011]. Also, it seems not involved in antioxidant defense, since knockout mutants do not show increased sensitivity to oxidative stress. Instead, experiments supported that the main function of *C. elegans* TrxR1 is in the removal of the old cuticle during molting, a process that involves the reduction of disulfide groups in cuticle components. TrxR1 function appears to be overlapping with the single glutathione reductase of *C. elegans* (GSR-1), since only the knockout of both genes shows phenotypic effects [Stenvall et al., 2011]. In [Taskov et al., 2005], authors searched other nematodes for selenoproteins, exploiting available EST sequences. Additional selenoprotein families were found

with Sec forms, particularly in the most basal nematodes like *Trichinella spiralis* (figure 1.12).

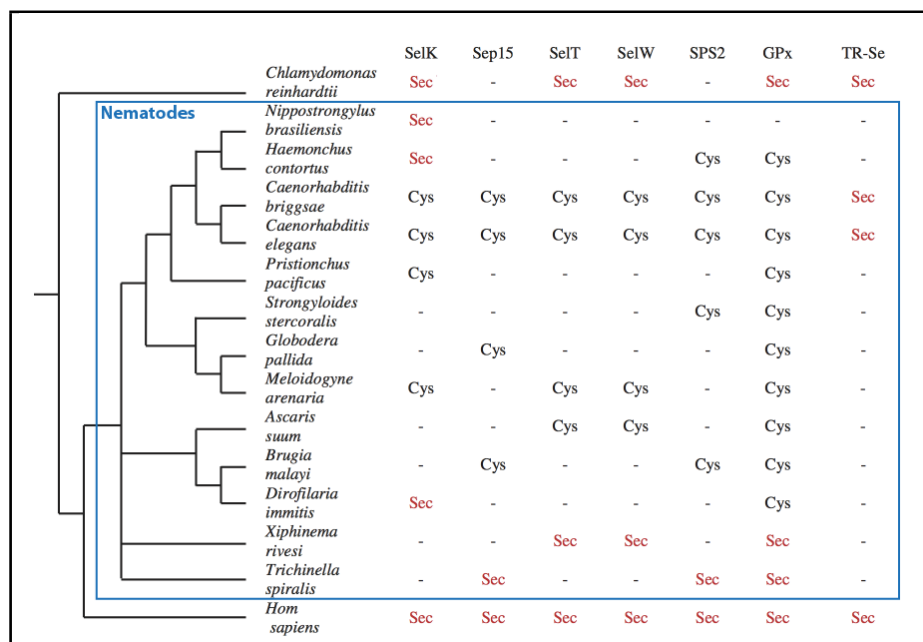


Figure 1.12: Map of selenoprotein genes found in nematodes, adapted from [Taskov et al., 2005].

These selenoproteins are found as Cys homologues in *Caenorhabditis*, suggesting they were converted. Overall the picture is quite similar to insects, with a progressive selenoproteome reduction in this lineage. It may seem surprising that, despite the fact that *C.elegans* is closer to a complete Sec extinction than *D.melanogaster* (1 dispensable selenoprotein versus 3), there are no documented cases of selenoproteinless nematodes. However, we expect some to be present in nature, and may be revealed through genome sequencing in the next years.

Last minute addition: at the 2013 Selenium conference in Berlin, Gustavo Salinas (Universidad de la Republica, Montevideo, Uruguay) showed that indeed he found some plant nematode parasites that lost selenoproteins and Sec machinery.

1.3.8 Selenoproteins in non-animal eukaryotes

Although selenoproteins were certainly best studied in animal model organisms, a number of other eukaryotes were also analyzed. Green algae were the subject of several studies: *Chlamydomonas reinhardtii* [Novoselov et al., 2002], *Ostreococcus* species *lucimarinus* and *tauri* [Palenik et al., 2007; Lobanov et al., 2007]. All these genomes were found rich in selenoproteins, in contrast to land plants which

have none. Selenoproteins were identified also in the diatom *Thalassiosira pseudonana* and in the amoeba *Dictyostelium discoideum* [Lobanov et al., 2007] (figure 1.13).

Selenoprotein family	<i>O. tauri</i>	<i>O. lucimarinus</i>	<i>T. pseudonana</i>	<i>D. discoideum</i>	<i>D. pseudoobscura</i>
SelK	+	+		+	+
SelH	+	+			+
SPS2			+	+	+
DI				+	
Sep15	+	+		+	
MSP	+	+		+	
Gpx	+++++	+++++	++		
SelT	+	+	+		
TR	+	+	+		
SelM	+	+	++		
SelU	+	+	++		
MsrA	+	+	+		
PDI	+++	+++	++		
Methyltransferase	+	+	+		
Peroxisredoxin	+	+++	++		
Thioredoxin-fold protein	+	+	+		
SelO	+	+			
SelW	+	++			
SelS	+	+			
Hypothetical protein 1	+	+			
Hypothetical protein 2	+	+			
Hypothetical protein 3	+	+			
Total	26	29	16	5	3

Each '+' corresponds to one selenoprotein gene.

Figure 1.13: Selenoproteins identified in some non-animal eukaryotes, from [Lobanov et al., 2007]. The last column reports the selenoproteins found in *Drosophila pseudoobscura*, for comparison.

Notably, the selenoproteome of all these organisms is largely overlapping with mammals (compare with figure 1.10), despite the huge phylogenetic distance. This, together with the scarce overlap of bacterial and eukaryotic selenoproteins [Driscoll and Chavatte, 2004], has been taken as indication that most eukaryotic selenoproteins were generated at the base of the eukaryotic radiation, and then lost independently in many clades [Lobanov et al., 2007]. Some selenoproteins do not follow this rule, and were generated specifically in some lineages. The case of the red algae *Cyanidioschyzon merolae* is intriguing in this regard: although Sec machinery was found, no known selenoproteins could be detected in this genome [Lobanov et al., 2007]. Its unknown selenoproteome may be then composed entirely of novel selenoproteins. This appears to be the case for the *Plasmodium* genera, for which several species have been sequenced. Only 4 selenoproteins were detected in *Plasmodia*, all of which have no homology with any annotated protein [Lobanov et al., 2006a]. Kinetoplastida (parasites including *Leishmania* and *Trypanosoma*) possess some selenoproteins orthologues to mammals (SelK, SelT, SPS2), and some lineage specific selenoproteins: SelTryp [Lobanov et al., 2006b] and Lmsel1 [Cassago et al., 2006] (only *Leishmania*). An analog situation was observed for the apicomplexan *Toxoplasma gondii*, with a similar core of ancestral

selenoprotein genes, plus the novel SelQ [Novoselov et al., 2007]. Remarkably, SelQ was not found in the close species *Neospora caninum*, in contrast to the rest of selenoproteins. Other apicomplexans seem instead to have lost all selenoproteins [Lobanov et al., 2007]. Recently, the harmful pelagophyte *Aureococcus anophagefferens* was described for its rich selenoproteome [Gobler et al., 2011, 2013]. This species was found to possess at least 59 Sec-containing genes, including the great majority of ancestral eukaryotic selenoproteins and a lot of novel ones. Most of *A. anophagefferens* selenoproteins contain a thioredoxin-fold and are predicted to possess oxidoreductase functions.

Summarizing, selenoproteins in non-metazoa exhibit a very dynamic evolution. Many species lost completely the Sec trait (including Fungi, land plants and many protozoa). While most selenoproteins were generated presumably in a short burst at the root of eukaryotes, several additions occurred in specific lineages, ending up with a completely renewed selenoproteome in a few cases (e.g. *Plasmodium*, maybe red algae).

1.3.9 Prokaryotic vs eukaryotic selenoproteome

The selenocysteine trait is found only in a minority of prokaryotic species [Kryukov and Gladyshev, 2004], and it is scattered through their phylogenetic tree. In archaea, the Sec trait was found uniquely in two genera of methanogens, *Methanococcus* and *Methanopyrus*. Here, all selenoproteins except SPS are involved in hydrogenotrophic methanogenesis [Stock and Rother, 2009]. Instead, bacterial selenoproteins carry out a range of very diverse functions, including redox homeostasis, electron transport / energy metabolism, compound detoxification and transport, oxidative protein folding. The eukaryotic selenoproteome shows little overlap with prokaryotes [Driscoll and Chavatte, 2004]. Using very loose criteria for comparison, the known shared families to date are SPS, GPx, MsrA, DI-like, PDI-like, SelW-like, Prx-like, Trx-like, methyltransferase. Also, some of these shared families have a different function in bacteria and, for example, vertebrates (this is evident for DI proteins). The most notable novelty in the eukaryotic selenoproteome is probably TR, which became essential for the redox metabolism of most eukaryotic organisms. Many other redox related selenoproteins were also originated, mostly with thioredoxin fold. Attempting a generalization, we can say that eukaryotes expanded selenoproteins families for redox defense, reduced those for compound detoxification and transport, and lost those for electron transport / energy metabolism. The oxidative protein folding pathways, already populated with PDI-like selenoproteins in some prokaryotes, also underwent notable expansions in eukaryotes, with novel superfamilies SelK/SelS and Sel15/Fep15/SelM.

1.4 Bioinformatics of selenoproteins

1.4.1 Selenoproteins are misannotated

The peculiar role of UGA in selenoprotein genes makes problematic their computational identification. In the coding sequences of any organism (with a standard genetic code), the almost totality of UGA codons are interpreted as translation termination signals. Standard gene prediction programs consider only this role for UGA, and thus fail with selenoproteins. In most genome sequencing projects, gene annotation is carried out mostly by such automated methods, and as a result the majority of selenoproteins are missing or wrongly annotated in public databases. Typically, three types of “misannotation” are observed (figure 1.14). First, the coding sequence is truncated at the three prime, for the Sec-UGA is interpreted as stop. This happens mostly for selenoproteins carrying Sec is at their C-terminus (e.g. TR, SelO, SelK). Second, the annotated coding sequence starts only downstream of the Sec UGA. Analogously, this happens more often for selenoprotein with Sec close to their N-terminus (e.g. SPS2, SelT, SelW). Third, the Sec UGA is skipped in the annotation, although this contains accurate coding regions both upstream and downstream of it. This happens because prediction programs try to avoid in-frame stop codons, penalizing them in their internal scoring schemes. The exon containing the Sec-UGA may be then completely skipped, or its splice sites shifted so that the Sec-UGA is considered intronic sequence.

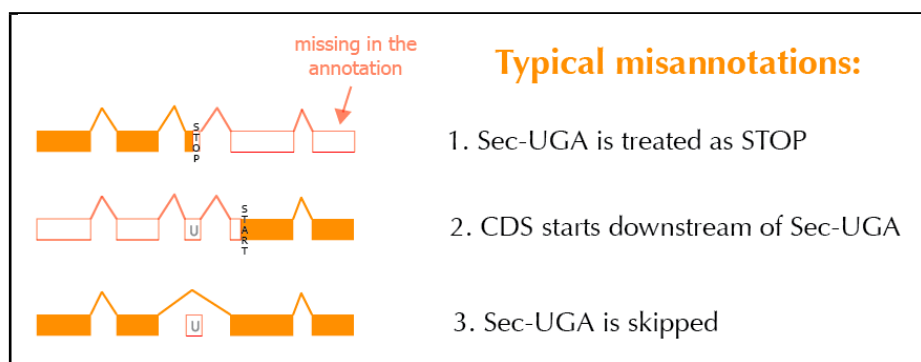


Figure 1.14: Typical selenoprotein misannotations.

Selenoproteins are well annotated only for a few model organisms, thanks to the manual annotation by few experts in the field. The database selenoDB was started in 2008 [Castellano et al., 2008] as an effort to amend the chaos of mis-annotated selenoproteins, inevitably increasing as more and more genomes were sequenced. SelenoDB 1.0 provided manually curated annotations for 9 eukaryotic species. Although only human and fruit fly were fully annotated, it provided for the first time a reference selenoprotein set, useful to predict selenoproteins by homology in other close species. More recently, the selenoprotein section of dbTeu [Zhang and Gladyshev, 2010] also provided such a reliable sequence set, remark-

ably extended to organisms from the whole tree of life.

1.4.2 Novel selenoprotein identification in eukaryotes

Finding the selenoproteins encoded in a genome can be divided in two conceptual problems: the annotation of known selenoprotein families, and the identification of novel selenoproteins. Naturally, the latter has found the interest of many groups in the last twenty years, when even the full human selenoproteome was yet to discover, and sequence databases were just starting to grow. Despite different for authors, implementation and performances, all methods relied mostly on three main concepts to support a selenoprotein candidate: 1. identification of a SECIS element properly located, 2. identification of annotated selenoprotein homologues (Sec/Sec alignment, for known selenoproteins), 3. identification of annotated cysteine homologues (Sec/Cys alignment).

In 1999, Vadim Gladyshev and collaborators [Kryukov et al., 1999] presented the first computational tool to search for eukaryotic SECIS elements: SECISearch. In this program, nucleotide sequences are scanned with sequence patterns that model the SECIS structure and conserved regions. Then, the thermodynamic stability of candidate structures are evaluated using RNAfold [Hofacker et al., 1994], and those too unstable are filtered out. In [Kryukov et al., 1999], the program was used to provide a list of candidate SECIS elements, and UGA-containing ORFs (possible selenoprotein genes) were searched upstream. Candidates were then screened experimentally, labeling with radioactive selenium. In this way, human MsrB1 (named SelR in the paper) and SelT were discovered. In the same year, an analogous method developed in the group of Alain Krol led to the discovery of a novel selenoprotein, SelN [Lescure et al., 1999], and also identified MsrB1 (here named SelX). SECIS prediction was carried out using a descriptor for the pattern-based program RNAMOT [Laferrière et al., 1994]. SECIS candidates were then analyzed and filtered using a variety of criteria, including thermodynamic stability and sequence similarity clustering. Experimental verification of novel candidate selenoproteins was again carried out using radioactive selenium incorporation. Two years later, an alternative approach was developed in our group and applied to the newly published *D.melanogaster* genome [Castellano et al., 2001]. The program geneid [Guigó et al., 1992] is a *de novo* gene predictor, that initially searches and scores potential genomic features such as starts, stops and splice sites in nucleotide sequences, using position weighted arrays. It then assembles these elements in potential gene structures, according to an underlying gene syntax (simplest example: start, many coding codons, a stop codon). One important signal used by geneid is the coding potential, i.e. the composition bias of coding sequences in comparison to those non-coding. In [Castellano et al., 2001], the program geneid was modified to allow prediction of Sec-containing proteins: a selenoprotein gene syntax was created, allowing in practice to detect genes with good scoring potential, a single in-frame UGA, and a potential SECIS element not too far downstream (predicted by SECISearch). This version of geneid (named here se-

lenogeneid) provided the first tool able to detect selenoproteins with no homology to any known sequences. It succeeded to predict the full *D.melanogaster* selenoproteome, consisting in known SPS2 and novel SelH (dselM or BthD) and SelG (G-rich), later noted as homologue to SelK, despite high sequence dissimilarity. Shortly after, Gladyshev and collaborators replicated these results using their own method [Martin-Romero et al., 2001]. The pattern-based SECISearch (webserver at <http://genome.unl.edu/SECISearch.html>) was the most successful tool used for SECIS prediction for many years, despite a profile-based method was also proposed: in [Lambert et al., 2002], authors showed that the program ERPIN [Gautheret and Lambert, 2001] could be trained with SECIS sequences and used for their identification, with remarkable specificity.

The full human selenoproteome was presented for the first time in 2003 [Kryukov et al., 2003]. Seven novel selenoproteins (GPx6, SelI, SelO, SelS, SelV and also SelH, SelK already identified in drosophila) were discovered by a combination of computational procedures again followed by experimental verification. Selenogeneid and a new version of SECISearch (including a covariance score computed with program covels) were both applied to the human genome. A new strategy was also devised to exploit the availability of the mouse and rat genomes, which we name SECIS orthology. Each human SECIS candidate was run with blastn against the candidates in mouse and rat, producing a list of putative orthologous SECIS elements. This allowed to obviate to the high number of false positives deriving from the application of SECISearch to vertebrate genomes, reducing the candidates for the manual downstream analysis. Nonetheless, this method alone would have missed human GPx6, since this selenoprotein is a Cys homologue in rodents (thus, SECIS-lacking). Instead, this protein was detected for its obvious identity with other Sec-containing GPx. The first SECIS-independent strategy for novel selenoprotein finding was used in [Castellano et al., 2004] and led to the identification of a new selenoprotein family in fish (SelU). In this work, the Sec/Cys alignment criteria was extensively used. Selenogeneid and standard geneid were used on the human and fugu genomes. Then, blastp was used to find matches in the predicted proteomes, building putative orthologous pairs. The blastp alignments were then filtered to focus on those with Sec/Sec pairs (putative selenoprotein in both species) or Sec/Cys pairs (putative novel selenoprotein in one species, Cys homologue in the other). The same procedure was used for a intra-species comparison, looking for Sec or Cys paralogues. The putative Sec sites were then scanned using conservation criteria. SECISearch was used only as last step, to characterize the SECIS element of the new candidate Sec-containing family. Later, the same strategy was also applied on the *Tetraodon nigroviridis* genome, and resulted in the identification of another new selenoprotein in fishes, SelJ [Castellano et al., 2005]. All mentioned strategies (SECIS orthology, Sec/Sec alignment, Sec/Cys alignment) were applied on the *C.elegans* and *C.briggsae* genomes [Taskov et al., 2005], here only to confirm that these nematodes encode a single selenoprotein, TrxR1. SECISearch and its applications were then fundamental for the identification of the last vertebrate selenoproteins discovered (so far): Fep15 [Novoselov et al., 2006]

and SelL [Shchedrina et al., 2007]. Recently, selenogeneid was rebuilt and improved by a group located in Shenzhen (China). The algorithm to assemble gene structures was redesigned specifically for selenoproteins (SelGenAmic), increasing its sensitivity. The method was applied for the identification of selenoproteins in chordate genomes [Jiang et al., 2010], and then in other invertebrates [Jiang et al., 2012], revealing two eukaryotic selenoprotein families that were thought limited to prokaryotes: DsbA and AhpC.

As we will see in the results section, our work in past years has contributed significantly to selenoprotein computational identification. We improved the program SECISearch, combining it with the RNA-search programs covels and Infernal [Nawrocki et al., 2009] to generate SECISearch3 [Mariotti et al., 2013]. This pipeline exploits a structural model for Infernal built from over a thousand SECIS sequences, and outperformed its predecessors. SECISearch3 itself was combined with blastx to create a new tool for selenoprotein gene finding in nucleotide sequences: Sebastian [Mariotti et al., 2013]. This program looks for potential selenoprotein genes upstream of each putative SECIS predicted by SECISearch3, using a reference protein database to search for homologues. In practice, Sebastian is a SECIS-dependent method that applies the Sec/Sec and Sec/Cys alignment strategies, and thus is able to predict both known selenoproteins, and novel selenoproteins with annotated cysteine homologues. SECISearch3 and Sebastian are publicly accessible through webserver at <http://sebastian.crg.es/> or

<http://gladyshevlab.org/SelenoproteinPredictionServer/>.

1.4.3 Novel selenoprotein identification in prokaryotes

The same key concepts used for eukaryotic selenoprotein finding were also applied to prokaryotes, correcting for the different structure and location of SECIS elements. In 2004, a computational tool to predict archeal SECIS elements was created [Kryukov and Gladyshev, 2004], named here aSECISearch. This program was built following the same structure of the first SECISearch: fixed patterns (built inspecting the known examples of aSECIS) are used to scan a target nucleotide sequence, matches are then filtered evaluating their thermodynamic stability, and a final filter checks additional structural criteria. aSECISearch, together with a SECIS-independent method based on Sec/Sec and Sec/Cys alignment criteria, was used to characterize for the first time the prokaryotic selenoproteome [Kryukov and Gladyshev, 2004]. An analogous program for bacterial sequences (bSECISearch) was presented one year later [Zhang and Gladyshev, 2005]. This program had a more complex structure, consisting of three modules. The first (bSECIScan) considers each potential UGA-containing ORF in the target sequence, slices a window just downstream and tries to predict its structure using RNAfold. Then, it filters candidates comparing them with a bSECIS consensus structural model. The second and third modules perform additional filtering procedures: bSECISProfile scores the candidates using position specific scoring matrices (PSSM) derived from an aligned set of 60 bona-fide bSECIS elements, while bSECISFilter uses tblastn

and blastx to apply protein conservation criteria (including Sec/Sec and Sec/Cys alignment). bSECISearch was fundamental for the genome wide characterization of bacterial selenoproteomes [Zhang and Gladyshev, 2005; Zhang et al., 2006], also in oceanic metagenomic samples [Zhang et al., 2008]. It is available through a webserver at <http://genomics.unl.edu/bSECISearch/>. In 2009, a new method for archaeal selenoprotein identification was devised (Asec-Prediction) [Li et al., 2009], following the same concepts described before: all UGA-containing ORF are considered, aSECIS-like hairpins are predicted in their proximity, and Sec or Cys homologues are searched in a reference protein database.

Finally, the characterization of selenoproteins and the discovery of pyrrolysine (the 22nd amino acid, inserted by recoding a TAG [Yuan et al., 2010]) has prompted researchers to design generic methods to detect rarely encoded amino acids in prokaryotes [Chaudhuri and Yeates, 2005; Fujita et al., 2007]. The question of whether other such non-standard amino acids existed was addressed in two studies. In [Lobanov et al., 2006c], a program to detect unusual tRNAs (including those for Sec and Pyr) was presented. In [Fujita et al., 2007], authors chose instead a protein-level approach, searching for conserved stop codons aligned to known proteins in other species. The conclusions of both studies were that no other non-standard amino acids exists, or at least not as widespread as Sec.

1.4.4 Annotation of known selenoproteins

The prediction of the known selenoproteins (Sec-containing genes with a known selenoprotein homologue) encoded in a genome have been generally carried out with standard gene prediction tools. Typically, a set of bona-fide selenoproteins is run with tblastn [Altschul et al., 1997] against a new genome. Results are then manually inspected, or previously parsed with scripts to detect Sec/UGA alignments. This procedure (or a similar one) was carried out whenever a reliable selenoprotein annotation was necessary (e.g. [Chapple and Guigó, 2008; Castellano et al., 2009]). This process could be very accurate, but also very time consuming. Also, it suffered from the drawbacks of standard gene prediction programs: unless specific options are used, stop codons are heavily penalized in coding sequences. This may be negligible if the UGA is embedded in high scoring blocks, but it has a deep effect for selenoproteins in which the Sec residue is very close to the C-terminus, or in very small exons. In these cases, the alignment output by tblastn is often incomplete, and researchers have to notice this and manually correct it.

In 2010, we presented a method addressing the annotation of known selenoprotein in genomes: Selenoprofiles [Mariotti and Guigó, 2010], described in detail in the next section. This program is a generic pipeline for profile-based gene finding: given an alignment of a protein family (profile), it scans target genomes for putative homologous regions, using a PSSM derived from the profile. Candidate gene structures are built joining different exons, and then refined with two widely used gene prediction programs, exonerate [Slater and Birney, 2005] and genewise [Birney et al., 2004]. A set of flexible filters (definable for each differ-

ent protein family) are then applied, finally outputting all the genes in the target genome that belong to the protein family in input. Selenoprofiles incorporates expedients to allow the correct prediction at Sec sites. First, all programs run internally (blast, exonerate, genewise) are used with modified scoring schemes to favor Sec/UGA alignments. Second, all profile sequences are modified to carry Sec in certain columns of the alignment, all those in which at least one Sec is present in the profile. In this way, Selenoprofiles scores positively UGA codons only when aligned to a known Sec position, and allows at the same time to exploit all cysteine homologues, extending the Sec/Sec alignment criteria at the level of protein families. We created and maintain profiles for all known eukaryotic and prokaryotic selenoprotein families, as well as for all Sec machinery proteins (the pipeline is not limited to selenoprotein families), allowing for out-of-the-box computational characterization of selenoproteomes. Selenoprofiles (now at its version 3.0, <http://big.crg.cat/services/selenoprofiles>) is public, and can be installed on any unix machine.

Chapter 2

METHODS

2.1 Selenoprofiles

When I started my PhD, selenoprotein search was carried out “manually”, sitting in front of the computer to run tblastn using known selenoprotein as queries. The blast hits were inspected, or filtered through parser programs, typically focusing on the alignment of a (seleno)cysteine with a stop codon in the target. The process was accurate, but very time consuming. Today, Selenoprofiles allows to characterize the selenoproteome content of a newly sequenced species within minutes or hours, with remarkable efficiency even without human intervention. In time, Selenoprofiles has become a genomics scale tool for profile-based search of any protein family, useful also for the comprehensive annotation of genomes.

Publication:

Mariotti M, Guigó R.

Selenoprofiles: profile-based scanning of eukaryotic genome sequences for selenoprotein genes. *Bioinformatics*. 2010 Nov 1;26(21):2656-63.

2.1.1 Automatic selenoprotein annotation

From the beginning of my PhD, my main objective was to build a tool to automatically annotate selenoproteins. This was necessary in first place to produce accurate annotations of genomes, whose number greatly increased in these years of great sequencing effort and advance. But also, such tool was needed to collect large sets of selenoprotein genes, the necessary data for any computational study. We chose to implement an homology-based tool, currently the best performing strategy for gene predictors. In these tools, a query protein sequence is aligned to a target nucleotide sequence, predicting in the target a gene homologous to the query. In general, these tools use protein coding sequences as input, so they actually predict only the protein coding portion of genes. One of the first and most widely used such tool is *tblastn* [Altschul et al., 1997], for its speed. For more accurate gene prediction and splice sites modeling, *exonerate* [Slater and Birney, 2005] and *genewise* [Birney et al., 2004] are commonly used. With Selenoprofiles, we created a pipeline to integrate and process the predictions of these and others programs, to finally annotate a protein coding gene structure using a profile alignment as input. Although we created Selenoprofiles for selenoprotein search, we intended to make it a useful genomic tool in general, for the prediction of any protein family. In the last years this pipeline has grown a lot and it is now a complex and flexible tool with an increasing number of users. Its main use is for comparative genomic studies, to search custom families across many genomes, and display results graphically using trees. But it can even be used to fully annotate genomes, using a comprehensive set of profiles of homologous sequences, as we did for the drosophila genomes.

In our publication in *Bioinformatics* [Mariotti and Guigó, 2010], here next, we validated Selenoprofiles (version 1.0) for selenoprotein annotation. To illustrate the many novelties of Selenoprofiles version 3.0, and show its value as a flexible annotation tool, we provide its latest manual in the appendix. In results section 3.4, you will see how Selenoprofiles was used for full annotation of drosophila genomes.

2.1.2 Selenoprofiles paper

We include here the paper describing Selenoprofiles published in *Bioinformatics* in 2010. Supplementary material sections are also included (except the gene prediction files).

Mariotti M, Guigó R. [Selenoprofiles: profile-based scanning of eukaryotic genome sequences for selenoprotein genes](#). *Bioinformatics*. 2010 Nov 1; 26(21): 2656-63.
DOI: 10.1093/bioinformatics/btq516

Selenoprofiles: profile-based scanning of eukaryotic genome sequences for selenoprotein genes

M. Mariotti^{1*} and R. Guigó¹

¹ Center for Genomic Regulation, Universitat Pompeu Fabra, Barcelona, Catalonia, Spain

Associate Editor: Prof. Martin Bishop

ABSTRACT

Motivation: Selenoproteins are a group of proteins that contain selenocysteine (Sec), a rare amino acid inserted co-translationally into the protein chain. The Sec codon is UGA, which is normally a stop codon. In selenoproteins UGA is recoded to Sec in presence of specific features on selenoprotein gene transcripts. Due to the dual role of the UGA codon, selenoprotein prediction and annotation are difficult tasks, and even known selenoproteins are often misannotated in genome databases.

Results: We present an homology-based *in silico* method to scan genomes for members of the known eukaryotic selenoprotein families: selenoprofiles. The core of the method is a set of manually curated highly reliable multiple sequence alignments of selenoprotein families, which are used as queries to scan genomic sequences. Results of the scan are processed through a number of steps, to produce highly accurate predictions of selenoprotein genes with little or no human intervention. Selenoprofiles is a valuable tool for bioinformatic characterization of eukaryotic selenoproteomes, and can complement genome annotation pipelines.

Availability and Implementation: Selenoprofiles is a python-built pipeline that internally runs psitblastn, exonerate, genewise, SECISearch, and a number of custom made scripts and programs. The program is available at

<http://big.crg.cat/services/selenoprofiles>.

The predictions presented in this paper are available through DAS at http://genome.crg.cat:9000/das/Selenoprofiles_ensembl.

Contact: marco.mariotti@crg.es

Supplementary Information: Supplementary data are available at Bioinformatics online

1 INTRODUCTION

Selenoproteins are a rare class of proteins containing selenocysteine (Sec), an unusual amino acid which is a cysteine analog with selenium replacing sulfur. Specific machinery is needed for the recoding of the UGA codon (usually a stop codon) to Sec (Allmang *et al.*, 2009; Xu *et al.*, 2007; Hatfield *et al.*, 2006). The main signal for UGA recoding is a RNA secondary structure element called SECIS (from SElenoCysteine Insertion Sequence) present in the 3' UTR of eukaryotic selenoprotein gene transcripts (Grundner-Culemann *et al.*, 1999; Copeland *et al.*, 2001). Selenoprotein homologues (not containing Sec) have been found

both as orthologues and paralogues. In most of them a cysteine residue is aligned to Sec. There are currently 21 known families of selenoproteins in higher eukaryotes: Glutathione Peroxidases (GPx), Iodothyronine Deiodinase (DI), Selenoprotein 15 (Sel15 or 15kDa), Fish selenoprotein 15 (Fep15), SelM, SelH, SelI, SelJ, SelK, SelL, SelN, SelO, SelP, SelR, SelS, SelT, SelU, SelV, SelW, Thioredoxin Reductases (TR), SelenoPhosphate Synthetase (SPS). Some of these families may contain more than one member in a given genome (e.g. *Homo sapiens* contains 25 selenoproteins belonging to 17 families). All known selenoproteins contain just one Sec, with a few exceptions: SelP, SelN, some DI isoforms (Gromer *et al.*, 2005), SelL (Shchedrina *et al.*, 2007). In protists selenoproteomes are variable, and recently some selenoprotein families limited to protist specific lineages were identified (Cassago *et al.*, 2006; Obata and Shiraiwa, 2005; Novoselov *et al.*, 2007; Lobanov *et al.*, 2006a,b). Some lineage specific selenoprotein families have been identified in algae as well (Lobanov *et al.*, 2009; Novoselov *et al.*, 2002; Palenik *et al.*, 2007). Selenoproteins' function is wide-ranging, and still unknown for many families (see Gromer *et al.* 2005 and Lobanov *et al.* 2009).

During the last decade, several computational methods have been developed and used to identify novel selenoproteins (see Driscoll and Chavatte 2004 for a review; Zhang and Gladyshev 2005; Li *et al.* 2009; Jiang *et al.* 2010). Most of these methods rely on the prediction of SECIS elements. A limitation of methods based on predicted SECISes is that they cannot identify selenoproteins with non-canonical SECIS elements, and they can be applied only to the species or taxonomic groups for which they were developed, since bacterial, archaeal and eukaryotic SECISes differ in their structure and also in their localization within the transcript (Krol, 2002). Also, SECIS prediction is problematic: while there is conservation of the secondary structure, the sequence is poorly conserved. Thus, genomic search for potential SECISes often lead to a large number of false positives (as well as, occasionally, some false negatives). Other strategies, not based on SECIS prediction, scan the target nucleotide sequence searching for ORFs with a conserved in frame UGA (Castellano *et al.*, 2004; Jiang *et al.*, 2010). These strategies also produces a large number of selenoprotein candidates in eukaryotic genomes. Like those based in SECIS searches, they require substantial manual curation. As a result, selenoprotein prediction is usually ignored in the standard genome annotation pipelines and selenoprotein genes are generally mispredicted, either by truncation of 3' end of the gene (the UGA codon assumed to be the stop codon of the coding region), or by truncation of

*to whom correspondence should be addressed

the 5' end (the coding region assumed to start at the first AUG downstream of the UGA Sec codon), or by exclusion of the exon or the region containing the UGA/Sec codon. As the number of genome sequences available grows exponentially, automatic tools that produce high quality genome annotations with minimal human intervention are essential. Here we present a computational pipeline, which we name selenoprofiles, capable of producing reliable gene predictions for known eukaryotic selenoprotein families. Selenoprofiles can be used in conjunction with automatic gene annotation methods to predict otherwise misannotated selenoprotein genes in eukaryotic genomes. Importantly, selenoprofiles does not rely on the prediction of SECIS elements. Also, selenoprofiles does not rely on individual selenoprotein sequences to be used as initial queries, but on sequence profiles characteristic of each eukaryotic selenoprotein family. For each eukaryotic selenoprotein family, we have thus built an high quality, manually curated multiple amino acid sequence alignment including all known orthologous and paralogous members of the family, and we derived a Position Specific Scoring Matrix (PSSM) from it. Profiles derived from multiple sequence alignments implemented as PSSM, Markov models or other structures, capture more precisely the intrinsic variation within a protein family, and often lead to searches that are both more sensitive (thus allowing for the identification of distant relatives) and more specific (easing the identification of spurious hits) (Altschul *et al.*, 1997). We show that selenoprofiles can be used with little or no human intervention to accurately identify known selenoproteins in eukaryotic genomes. Application of selenoprofiles to the publicly available reference annotation of metazoan genomes reveals hundreds of misannotated selenoprotein genes.

2 METHODS

Algorithm: the selenoprofiles pipeline

Selenoprofiles is a computational pipeline that, provided an alignment for a protein family, identifies all members of said family encoded in a target genome sequence. Selenoprofiles includes curated amino acid sequence alignments of all known eukaryotic selenoprotein families and selenoprotein factors. However, it can actually be used with alignments from any protein family. Technically, therefore, the pipeline is a general homology-based gene finder program with specific features that make it particularly suitable for selenoprotein identification. In selenoprofiles, the program *psitblastn* is used to identify matches in the target genome to the selenoprotein sequence alignments (see Figure 1-a). These matches are then used, through two different splice alignment programs, *exonerate* (Slater and Birney, 2005) (see Figure 1-b) and *genewise* (Birney *et al.*, 2004) (see Figure 1-c/d) to deduce the exonic structure of the candidate selenoprotein genes. The predictions of these two programs are analyzed to produce a final one (see Figure 1-e). Finally, the program *SECISearch* (Kryukov *et al.*, 1999) is used to identify suitable SECIS elements downstream of the coding region of the candidate selenoprotein genes (see Figure 1-f). Through the entire pipeline a number of steps are performed (detailed below) to filter out likely false positives and to keep the number of potential candidates under manageable levels. Next, we detail first the building of the selenoprotein profiles and then the different steps in the pipeline.

Multiple sequence alignments of protein families

Selenoprofiles includes amino acid sequence profiles for all known eukaryotic selenoproteins, as well as for all known selenoprotein-specific factors, that is, proteins involved in the synthesis of selenoproteins: SECIS binding protein 2 (SBP2), selenocysteine specific elongation

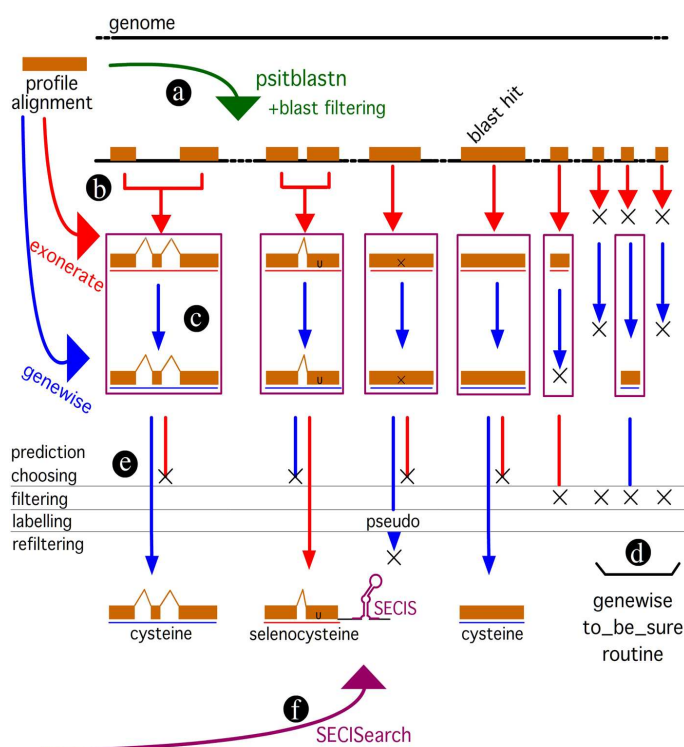


Fig. 1. Schema of the selenoprofiles pipeline. Initially, a *psitblastn* search is run using a PSSM built from the profile alignment (a). The resulting genomic intervals are merged into “superexon” intervals, and cyclic *exonerate* is run on each of them (b). Then, *genewise* is run both to refine *exonerate* predictions (c) and when *exonerate* failed recovering blast alignments (d - *genewise* “to be sure” routine). The *exonerate* or the *genewise* prediction is chosen (e), and then results are filtered, labelled, and then refiltered with family specific filters. Lastly, *SECISearch* is used to detect potential SECIS elements downstream of the genes (f).

factor (eEFsec), O-phosphoserine-tRNA^{sec} kinase (PSTK), O-phosphoserine-tRNA^{sec}:selenocysteine synthase (SepSecS or just SecS), selenocysteine tRNA associated protein 43 (secp43), and selenophosphate synthetases (SPS1/SPS2). Searching for selenoprotein factors, in addition to the search for selenoproteins, is important because some of these factors appear to be good markers of selenoprotein encoding (Chapple and Guigó, 2008). While all selenoprotein factors (apart SPS2) are not selenoproteins themselves, and therefore their annotation does not suffer from the intrinsic limitations of selenoproteins, still the usage of selenoprofiles may result in a more accurate annotation than that produced by standard automatic annotation methods.

The seed sequences (one per family) to build the selenoprotein profiles were taken from SelenoDB (Castellano *et al.*, 2008), a database of selenoproteins and selenoprotein factors. The human protein sequence was used when available. One exception was the SelK family, for which two distinct profiles were built, one utilizing the human sequence as seed and another utilizing the drosophila sequence. This was necessary because this protein family is very divergent in insects. Also, the two selenoprotein families SelV and SelW were merged into a single profile alignment, since they share high sequence similarity (even though SelV possesses an additional N-terminal domain). Representative sequences from families not yet included in SelenoDB: SelJ, SelL, Fep15, were taken from the genomes where they were identified (see respectively Castellano *et al.* 2005; Shchedrina *et al.* 2007; Novoselov *et al.* 2006). For all families, the

sequences used to build the profile were selected running the seed protein with either psiblast or blastp (Altschul *et al.*, 1997) against nr (Sayers *et al.*, 2010), with a very loose e-value filtering (max e-value=1). The resulting sequences were aligned with the seed with t_coffee ver. 5.65 (Notredame *et al.*, 2000). The alignment was then trimmed for redundancy with the t_coffee trim subroutine. Each alignment was then manually inspected and modified to remove spurious sequences or to add sequences that were missed during this process.

Finding matches to the selenoprotein profiles in the target genomes

In selenoprofiles, the multiple sequence alignments in input are compared to the sequence of the target genome using psitblastn, a member of the psiblast family of programs. This program is an extension of tblastn, that uses a protein PSSM to search nucleotide sequences translated in all 6 frames. While the psiblast programs are generally used to search iteratively a database and build an increasingly accurate profile, in this pipeline the profile is given as input, so a single search is performed against the target genome. Selenoprofiles utilizes psitblastn from the ncbi blastall package, version 2.2.22. The results of the search are filtered using the program alignthingie.pl (Charles E. Chapple, personal communication). Three types of blast hits pass the filter: those in which the Sec position is aligned to a UGA codon, those hits in which it is aligned to a cysteine coding codon, and all other hits whose e-value is below a certain threshold.

Inferring the exonic structure of the selenoprotein candidate genes

For each selenoprotein alignment, the output of the step above is a set of hits in the genomic sequence (genomic intervals), roughly corresponding to the exons of candidate selenoprotein genes (see Figure 1-a). Each such genomic interval is used to initiate an iterative exonerate alignment that would ideally recover the entire exonic structure of the candidate selenoprotein gene. This initial structure may be subsequently refined through the usage of genewise, another splice alignment tool. Before running exonerate, the genomic intervals likely to correspond to exons of the same gene are merged in "superexon" genomic intervals, to minimize subsequent computation (see Figure 1-b). For two hits to be merged, one must align a region of the profile that is downstream of the region aligned in the other one, and also be located downstream along the genome sequence within a given distance.

Cyclic_exonerate. Exonerate is a generic tool for pairwise sequence comparison. Selenoprofiles utilizes exonerate version 2.2.0, in protein-to-genome mode, that aligns a single protein sequence (the query) to a nucleotide sequence (the target or subject), incorporating prediction of splice sites. Selenoprofiles runs exonerate in a peculiar way, hereafter described as the cyclic_exonerate routine (see Figure 2). We use this procedure to ensure that the whole gene structure of a candidate is found, without the need to use as subject the whole target chromosome and neither making *a priori* assumptions on the gene width. This method initially runs exonerate using as target the genomic interval defined in a blast hit (or in a "superexon"). It then runs exonerate again on the same interval extended at both ends, and compares the two alignments produced. In case that the second run of exonerate extends the coding sequence with respect to the first, then additional runs will be performed, as long as extending the genomic interval results in an extended gene structure prediction. If the extension parameter is chosen larger than the largest expected intron, the whole gene structure of the target should be detected.

Exonerate can only take as protein query a single sequence – and not an entire alignment or a profile. At each run of exonerate, cyclic_exonerate thus maps the current query-target alignment into the profile alignment, and selects as a query the sequence in the profile which is the most similar to the predicted protein sequence. In selenoprofiles, only a subset of the sequences of the profile are allowed to be chosen as exonerate/genewise queries, since the profile may contain also incomplete

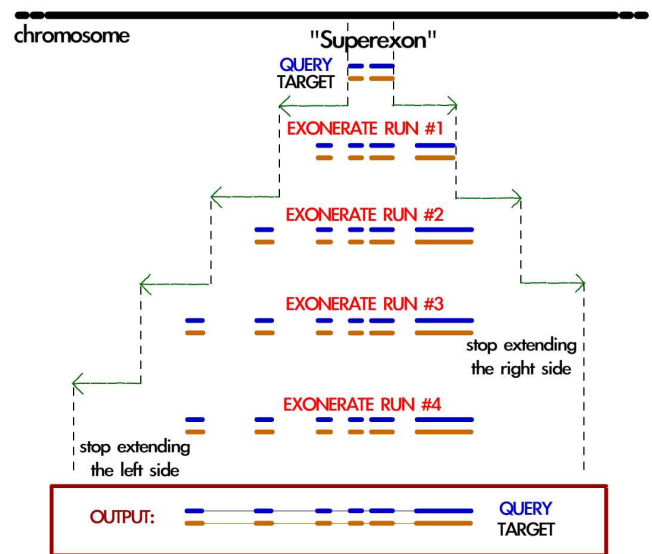


Fig. 2. Schema illustrating the cyclic exonerate routine. The program is run on a genomic interval initially defined by a blast hit (or a set of merged blast hits - "superexon"), which is extended at each cycle. After each exonerate run (except the first one), the resulting prediction is compared with the previous one and the program decides whether to perform another run or not. Just before running exonerate (not displayed), the current alignment is mapped to the profile alignment and the query protein which is most similar to the target sequence is chosen. Although in the shown example exonerate is run 4 times, cyclic exonerate runs on average 3.03 cycles (on well assembled genomes such as the ones used for testing).

sequences. Cyclic_exonerate launches exonerate with a custom scoring matrix (derived from BLOSUM62) which is favoring the extension of the alignment over Sec-encoding UGA codons: in the query protein selected from the profile, the position(s) aligned to Sec are replaced with a flag character (*). The custom scoring matrix contains positive values corresponding to the alignments of this character with * (any stop codon, score 8), and with cysteine (score 4), as well as with arginine (score 2) and threonine (score 1), since these amino acids have been found aligned to Sec in some eukaryotic selenoprotein families. The alignment of * with any other amino acid is scored with -4.

When multiple predictions are present in an exonerate output, only the main prediction (defined as the highest scoring among those overlapping the original input "superexon") is reported by selenoprofiles. Often, however, exonerate fails to join predictions which actually belong to the same gene, because no canonical splice sites are found or because a region of the query sequence that would bridge the predictions is not found in the target sequence. Therefore selenoprofiles uses secondary exonerate predictions to extend the main one: such predictions must align a region of the profile that is downstream (upstream) of the region aligned in the main one, and they should also be located downstream (upstream) in the genome sequence. That is, co-linearity needs to be maintained.

It is possible that more than one exon per gene initiates the exonerate cycle. In most of these cases the procedure just described converges, leading to the choice of the same query protein and therefore to identical gene structure predictions. In a few cases, the procedure does not converge and slightly different gene structures are predicted. Exonerate predictions are processed to produce a unique gene structure per genomic loci: identical predictions are considered just once, and predictions which are completely included within the boundaries of another are discarded. Rarely, partially

overlapping predictions, not including each other, are produced by this procedure. These will be output separately at the end of the pipeline. Note that there may be multiple non-overlapping exonerate predictions for a given selenoprotein profile, which could correspond to different members of the selenoprotein family. Selenoprofiles attempts next to refine the exonerate predictions through genewise.

Genewise. Genewise is a program belonging to the Wise2 package that performs a protein to DNA splice alignment (analogously to exonerate). Selenoprofiles utilizes genewise from Wise version 2.2.3. Generally, genewise is used to refine the gene boundaries of predictions already produced by exonerate (see Figure 1-c). Sometimes, however, the exonerate routine seeded by a psitblastn hit (or by a “superexon”) produces no output. We also use genewise in these cases to produce a prediction on the genomic interval outlined by blast and extended by 10.000 bp on each side (genewise “to be sure” - see Figure 1-d). As with exonerate, the query sequence to be used is chosen from the profile alignment maximizing the sequence similarity to the predicted peptide sequence in the target. Genewise is run just once, so the query sequence in the standard routine is always the last one used by exonerate. When no exonerate output is available, the query sequence is chosen maximizing the similarity to the target peptide sequence predicted by psitblastn. Genewise can accept as query also a profile (not only a single sequence), in the form of a Hidden Markov Model (HMM). Nonetheless, selenoprofiles implements the use of genewise only with single protein queries, to keep the time of computation acceptable in genome wide searches. As with exonerate, genewise is run with a custom scoring matrix favoring the alignment of Sec with UGA codons, with cysteine codons, or, with a lower score, with arginine or threonine codons. The query sequence chosen from the profile is replaced with a flag character (in this case U) in the positions that are aligned to Sec in the profile. In the case of genewise, though, it is possible to customize the program behavior to favor only the alignment of the U with UGA codons (not with other stop codons): this is accomplished by providing a different codon table to genewise, in which UGA codes for U.

Final prediction. At this point, selenoprofiles compares the genewise prediction with the prediction by exonerate, and chooses only one of them (see Figure 1-e). In our experience, using the two programs instead of just one of them improves both the performance and the stability of the pipeline (see section S3). Since the scores of the two programs are not comparable, selenoprofiles chooses the prediction with the longest protein sequence, unless it is likely to correspond to a pseudogene (that is, it contains frameshifts or non-Sec coding stop codons), or it does not include a residue aligned to the Sec position(s) of the profile. In this case, the shorter prediction is chosen provided that it does not verify these two conditions. In our analysis, the genewise and exonerate predictions are identical in 27% of the cases. When they are different, selenoprofiles chooses the genewise prediction over the original prediction by exonerate in 68% of the cases. The final predictions are then filtered (see next section).

Filtering, labelling and outputting

Gene predictions are filtered so that only predictions spanning at least a given fraction of the profile alignment (40%) or longer than a given threshold (60 amino acids) are reported. All gene predictions that pass this filtering step are output, producing sequence files (in fasta format) and gene coordinate files (in General Feature Format - GFF - see <http://www.sanger.ac.uk/resources/software/gff/spec.html>). Each gene prediction is labelled according to the codon that aligns to Sec in the profile. If a UGA codon is occurring at this position, the gene is labelled as “selenocysteine”. If another codon is occurring, the label takes the name of the correspondent amino acid (which is cysteine most of the times). There are some other possible labels, detailed in the caption of Figure 3.

SECISearch

Finally, selenoprofiles utilizes SECISearch version 2.0 (Kryukov *et al.*, 1999), as adapted in Chapple *et al.* 2009, to search for potential SECIS elements in the genomic region downstream from the predicted selenoprotein genes (see Figure 1-f). By default, a region of 3000 bp is scanned. Initially, selenoprofiles attempts to find SECIS element matching the standard pattern, which fits both forms of eukaryotic SECISes (see Krol 2002). If no SECISes are found matching this pattern, SECISearch is run with two increasingly degenerate SECIS patterns (all patterns are reported as supplementary material, section S1). It is possible that more than one SECIS is found in this way. It is also possible that no SECIS is found at all. Nevertheless, selenoprofiles does not drop a prediction for lacking a SECIS prediction. We believe that in most cases the occurrence of a UGA aligned to a Sec position of a known selenoprotein family is a very strong evidence for selenoprotein function. The lack of a detectable SECIS in the genomic region downstream of a real selenoprotein gene can be due to unusual features of the SECIS, but also to poor quality in the genome assembly, or to the presence of long and/or many introns in the 3' UTR of the candidate.

Refiltering

Some profiles report false hits, either because the protein alignment for the family features poor sequence information (causing spurious hits along the genome), or because the family shares a certain degree of similarity with members of some other non selenoprotein families (causing the profile to identify these genes). Through our experience with specific protein families we have learnt to recognize these cases, and we have thus implemented a number of filters to identify, label and remove them. Filters are specific of each selenoprotein family. As an example, the refiltering for the SelV family is as follows. This family is characterized by a long, unstructured N-terminal domain showing very poor conservation, and a conserved C-terminal region. The N-terminal region sometimes causes this protein profile to produce many spurious hits in the genome. Through the refiltering, we ignore the hits that align only in this unstructured N-terminal region.

```
SelV: result.obj.label!='pseudo' and
      result.obj.boundaries.inprofile()[1]>=300
```

Implementation

Selenoprofiles has been implemented in python. Selenoprofiles contains a number of profile alignments and scripts, including a program for graphical output: `selenoprofiles.drawer.pl` (see Figure 3). A Perl program (`get.annotation.pl`) is used when searching genomes with annotations in Ensembl. This program interrogates online the Ensembl database utilizing the Perl Core API, and retrieve the most similar annotation in Ensembl to each selenoprofile prediction. The database releases for all species considered in this article are reported in Table S2. The code and manual of selenoprofiles is available at <http://big.crg.cat/services/selenoprofiles>. Selenoprofiles scanned the human genome for all the 27 implemented families in 1, 100 minutes (about 18 hours) in a computer equipped with 2 double-core Intel(R) Xeon(TM) processors (2.80 Hz) and 4 Gb of RAM. About 46% of the time was spent on the SelV family alone.

3 RESULTS

Evaluation of selenoprofiles

We have tested the performance of selenoprofiles on the genomes of *Homo sapiens* (25 selenoproteins and 5 selenoprotein factors), *Drosophila melanogaster* (3 selenoproteins and 5 selenoprotein factors) and *Saccharomyces cerevisiae* (no selenoproteins and no selenoprotein factors), since these genomes are well annotated in Ensembl, and they have all entries in SelenoDB. We ran selenoprofiles removing preemptively all sequences belonging to the tested species from the profiles alignments. In addition to the families already mentioned, we included the Methionine

sulfoxide reductase A (MsrA) family as well, since this family is included in SelenoDB (although it was found as selenoprotein only in *Chlamydomonas reinhardtii*, Novoselov *et al.* 2002). Overall, selenoprofiles found 27 out of the 28 selenoprotein genes, 10 out of 10 selenoprotein factor genes, and 26 out of 28 annotated selenoprotein homologues. The three genes missed by selenoprofiles are drosophila SelK2, and human SelW1 and SelW2.

SelK2 is a cysteine homologue of SelK, and is located adjacent to it on the fly genome, confounding selenoprofiles. The human SelW proteins (the selenoprotein SelW1 and the cysteine homologue SelW2) have an exon structure made of very short exons which produces, in the psitblastn search, e-values that are higher than the threshold. The sequences are correctly predicted when searching the ncbi human ESTs database with selenoprofiles (data not shown).

For selenoproteins (meaning in this case all predictions labelled “selenocysteine”), selenoprofiles produced no false positives in the yeast and drosophila genomes (see Table S3). In the human genome, five selenoprotein genes that were not present among Ensembl or SelenoDB annotations were predicted – these are very likely to be false positives (see section S5 in the supplementary material). Regarding the selenoprotein machinery, four false positives in total were predicted by selenoprofiles in the three genomes (see section S5). For non-Sec homologues of selenoproteins, more false positives were predicted (see Table S3). Their number depends mostly on the protein family considered (that is, on the effectiveness of the refiltering steps specific to that family).

In addition to assessing whether selenoprofiles was able to identify the selenoprotein and machinery genes in complete genomic sequences, we also evaluated the quality of the exonic structure inferred by selenoprofiles for these genes. Predicted and annotated gene structures were compared and the usual measures of sensitivity and specificity at gene, exon and nucleotide level (Burset and Guigó, 1996) were computed using the script evaluation.pl (Eduardo Eyras, personal communication). The details of the procedure and the results appear in Table S3. Overall, accuracy values are comparable (or even higher) to those obtained through the most accurate automatic gene annotation pipelines: for selenoproteins, both the average sensitivity and the average specificity at the nucleotide level are above 90%.

Using selenoprofiles to identify selenoproteins in eukaryotic genomes

To further assess both selenoprofiles and the current status of selenoprotein annotation in eukaryotic genomes, we ran selenoprofiles on all 46 currently available Ensembl genomes (all eukaryotes). 837 selenoprotein genes, 925 non-Sec homologues, and 236 selenoprotein factors were found. A summary of the results is given in Figure 3. The figure, produced by the program selenoprofiles_drawer, lists the selenoprotein families found in the analyzed genomes and the number of genes in each family, indicating whether these are selenoproteins, cysteine homologues or contain other amino acids at the Sec position. Consistent with our assessment in the human, fly and yeast genomes, results indicate that, while selenoprofiles finds most of the known selenoprotein genes, it also misses some of them. This is due in part to limitations of the profiles, but mostly to the quality of the genome sequence.

For example, the mosquitoes *Aedes aegypti* and *Anopheles gambiae* are known to possess the selenoprotein SelK (Chapple

and Guigó, 2008), but their protein sequence is quite distant from both drosophila SelK (used to seed the SelK.insect profile) and vertebrate SelK (with human SelK used to seed the SelK profile). Consequently, the annotated SelK is missed in these two genomes by both SelK profiles searches. Other genes are missed in the psitblastn search because of the e-value of the alignment is above the threshold. In other cases, selenoproteins are not found because of incompleteness in the genome sequence. Thus, no SPS2 is predicted by selenoprofiles in *Gallus gallus* genome, but this gene can be easily found searching the EST data available at ncbi for this organism (data not shown). Other cases of genes that we expect to be present, but are missed by selenoprofiles correspond to predictions labelled as pseudogenes, because of frameshift(s) or in-frame stop codons. This happens with selenoprotein families as well with machinery proteins (e.g. SecS in *Microcebus murinus* and PSTK in *Rattus norvegicus*). Since all Ensembl species (apart from *Saccharomyces cerevisiae*) possess selenoproteins and therefore must have the necessary machinery, we believe this suggests the occurrence of sequencing errors in the genomes. Many genomes included in Ensembl are characterized by low coverage, and this is known to heavily affect the inferences on gene presence in such species (Milinkovitch *et al.*, 2010). Out of the 837 selenoproteins predicted by selenoprofiles, 658 of them contain a putative SECIS elements. We find a correspondent gene annotation in Ensembl for 604 of them. In 66 cases, the gene was correctly annotated as a selenoprotein. Given the low false positive rate of selenoprofiles, most of the 771 remaining cases are likely to correspond to misannotations. For the 233 cases in which no correspondent Ensembl annotation was found, we believe that the in-frame UGA confounded the Ensembl annotation pipeline to the point that no annotation at all was produced. Among the 538 remaining cases, we observed a few recurrent patterns of misannotation: in 154 cases (28.6%) the annotated coding region in Ensembl ends exactly at the Sec-UGA site (mostly for families with a C-terminal Sec), while in 100 cases (18.6%) starts downstream of it (for families with a N-terminal/central Sec). In 231 (42.9%) cases, there is a deletion in the annotated coding region compared to the selenoprofiles prediction that includes the Sec-UGA codon. Often the deletion eliminates only this codon through the annotation of a 3 bp intron. The 53 (9.9%) remaining cases do not fall in any of the previous categories. A list of the misannotated genes for each category is provided as supplementary data. Selenoprofiles predictions on all Ensembl genomes can be accessed through DAS at http://genome.crg.cat:9000/das/Selenoprofiles_ensembl.

4 DISCUSSION

In spite of significant advances, gene annotation of newly sequenced genomes remains a challenging task. While manual curation is still essential to produce high quality gene and transcript annotations (Guigó *et al.*, 2006), automatic genome annotation pipelines produce increasingly accurate gene sets (Harrow *et al.*, 2009), in particular for well characterized protein coding families and when other well annotated evolutionary close genomes exist. Due to their peculiar recoding of the standard genetic code, selenoproteins constitute the most notable exception; even for well annotated genomes, they are often mispredicted. Indeed, as we have shown through the analysis described here, most eukaryotic selenoproteins

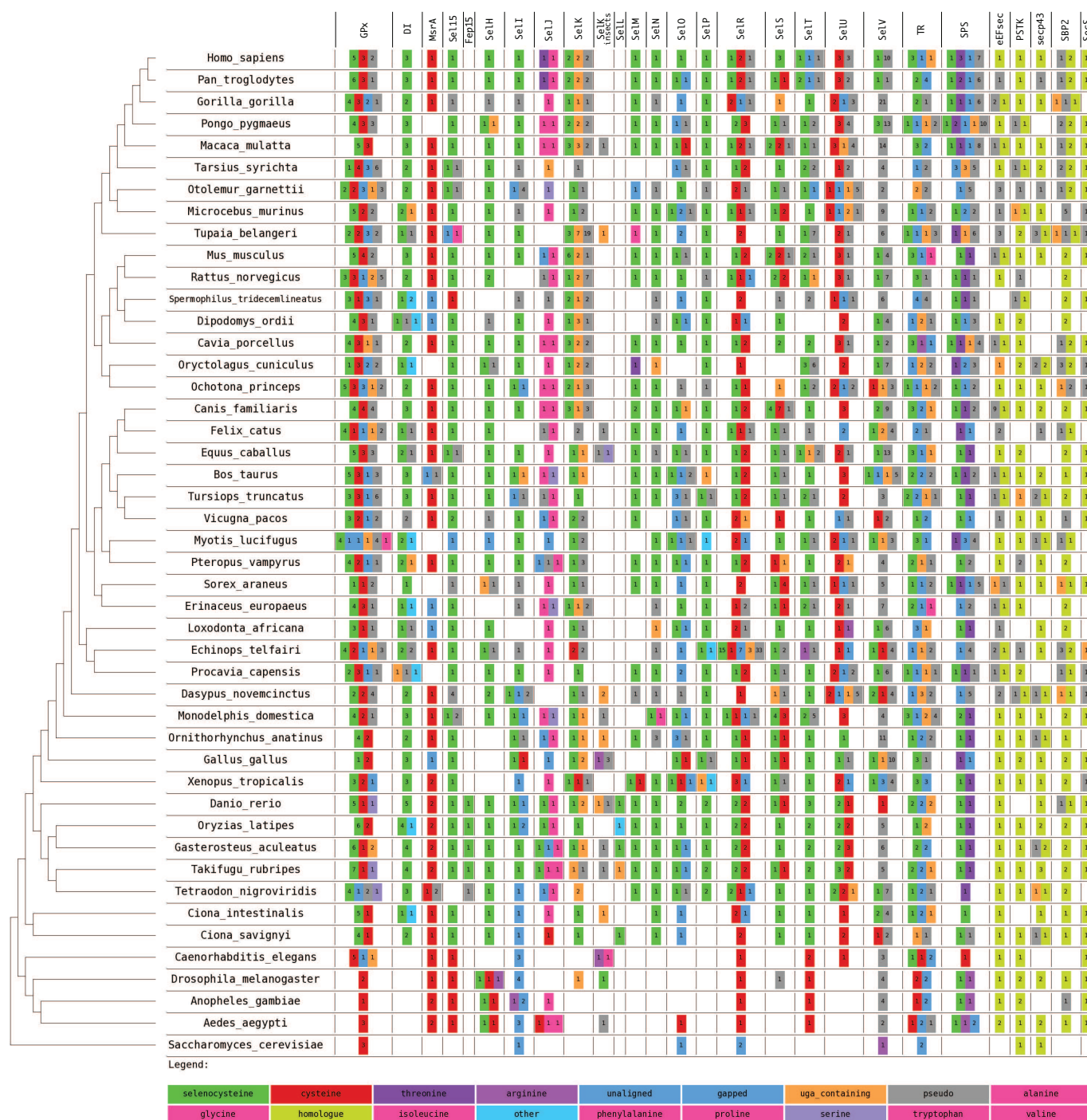


Fig. 3. Graphical summary of non redundant selenoprofiles predictions on all Ensembl genomes. The summary have been obtained with the program selenoprofiles_drawer. For each species, the numbers in the colored boxes indicate how many hits were found for each protein family (column) and label (box color). A color-to-label legend is located at the bottom: selenoproteins are in green, cysteine homologues in red and so on. Rare labels (such as “glutamine”, “tryptophan”, “glutamic acid”) are all indicated with the pink color and cannot be differentiated in the picture. Hits labelled as “pseudo” contain frameshifts or stop codons other than UGA (these were included in this figure although they are filtered out by selenoprofiles). The label “uga_containing” is used when the only in frame stop codon(s) are UGAs (not aligned to any Sec). This is useful since the scoring scheme rarely allows the alignment over a non-Sec encoding UGA. When no profile Sec position is aligned, the hit is labelled as “gapped” in case the prediction aligns regions in the profile both upstream and downstream of the Sec position, “unaligned” otherwise. The label “other” is only for selenoprotein families with more than one Sec, when none of them are aligned to a UGA. The selenoprotein machinery families (not containing Sec) are on the right of the figure. The non-pseudo, non-uga_containing predictions for these families are labelled as “homologue”. A phylogenetic tree serves to indicate the evolutionary position of the investigated species (Toni Gabaldón, personal communication). In the tree, three unresolved nodes were given an arbitrary topology for visualization purposes. This image can be downloaded at http://genome.crg.es/datasets/selenoprofiles2010/results_ensembl52.png

are misannotated in the available reference gene sets. Since misannotation invariably involves the deletion of the region in the protein sequence including the Sec-UGA-key to proper family assignation-misprediction in the case of selenoproteins have the additional negative effect, beyond simply protein truncation, of impairing proper functional characterization.

Proper annotation of selenoprotein genes—even those belonging to well characterized protein families—requires substantial human intervention. Indeed, due to the degeneration of the sequence of the SECIS element, and to the complex evolutionary history of selenoprotein genes, with frequent gene duplications and family expansions, pseudogenizations, and the yet not completely understood evolutionary dynamics of Cys to Sec inter-conversion (Castellano *et al.*, 2009), detection of sequence homology is, in general, not sufficient for correct selenoprotein identification. In fact, the correct annotation of the two dozen (at the most) selenoprotein genes corresponding to known selenoprotein families which may be encoded in a newly sequenced eukaryotic genome takes, in our experience, two to three weeks of full time work of an experienced scientist. He/she has to browse through a maze of multiple sequence alignments and SECIS predictions, making often ad-hoc decisions, which generally involve running additional, more sophisticated alignment programs and post-processing their output. In selenoprofiles we have attempted to encapsulate the experience that we have accumulated during the years in manual identification of selenoproteins. Selenoprofiles includes standard sequence similarity search and sequence alignment programs together with custom made post-processing scripts and a number of rules that direct the overall flow of the process. The core of selenoprofiles is a set of very high quality multiple sequence alignments for the different selenoprotein families and subfamilies. Given that we know *a priori* which positions in a profile alignment are allowed to bear a selenocysteine, selenoprofiles favors the alignment to UGA codons only if these are aligned to one such position. Therefore an important feature of each profile alignment is the position or positions that contain Sec, and one of the major determinants of the efficiency of the selenoprofiles pipeline are the species and the subfamilies represented in the profile. Selenoprofiles automatically selects the best sequence to be used as query from the profile. Consequently if the profile contains at least one sequence that is very similar to the protein coded by the gene that is predicting, the prediction will be accurate. But if the most similar sequence in the profile differs from the real protein encoded in the investigated genome in the presence or absence of some domains, or if there is poor conservation between the two sequences at some regions (often at one or both ends), then the prediction may be inaccurate. Input profile alignments for selenoprofiles should, therefore, be as consistent, complete and representative as possible. In this regard, as new genomes are being analyzed, we keep updating selenoprofiles, and we are working in a procedure to substantially automate this updating.

While selenoprofiles does not completely eliminate the need for manual intervention, it dramatically reduces it. We estimate that, after running selenoprofiles on a (newly sequenced) genome, an experienced scientist will need, in general, only a few hours to produce a high quality annotation of the selenoprotein genes corresponding to known families in the genome. But, given its low false positive rate, even the default output of selenoprofiles will generally be a much superior annotation of selenoprotein genes than

that produced by automatic annotation pipelines—including the most sophisticated ones. In this regard, we believe that selenoprofiles would be a useful complement of such pipelines, and we are working on a method to automatically correct the misannotated selenoproteins taking into account the selenoprofiles output. Using directly this output may not be an option, since sophisticated annotation pipelines rely on transcript information (such as ESTs and cDNA sequences), as well as genomic sequence conservation across species, and the overall gene structure delineated using this information is likely to be superior to the one delineated by selenoprofiles, with the exception of the region including the Sec-UGA. Therefore, a better strategy will be to conciliate the selenoprofiles prediction with the annotated gene, giving predominance to the selenoprofiles prediction in the region (exon) containing the Sec-UGA, but to the annotated prediction in the rest of the gene/transcript.

One limitation of selenoprofiles is that it predicts, with a few exceptions only one transcript per gene. Nonetheless, if alternative splicing forms (Sec/non-Sec) exist for a gene, the pipeline is likely to pick the Sec containing transcript, or one of them, due to the scoring scheme used. If selenoprofiles is used on transcribed sequences (such as ESTs, cDNAs, or RNA sequences) instead of genomic sequences, it could potentially produce predictions for multiple splicing isoforms of selenoprotein genes. While we have developed and tested selenoprofiles to annotate eukaryotic selenoproteomes, the strategy that we have employed can be easily ported to prokaryotic genomes as well. This requires the building and curation of the corresponding profiles, the usage of the bacterial and archaeal SECIS patterns, and the modification of some of the selenoprofiles rules.

5 CONCLUSION

Selenoprofiles is an homology-based method to produce accurate predictions of known selenoprotein families, and can be used in conjunction with automatic annotation pipelines. Running selenoprofiles on all available eukaryotic genomes reveals hundreds of misannotated selenoprotein genes. Selenoprofiles predictions constitute the largest available collection of eukaryotic selenoproteins, and are in this regard, an invaluable resource for selenoprotein research.

ACKNOWLEDGEMENT

We thanks Toni Gabaldón for providing a phylogenetic tree of all species present in Ensembl. Thanks also to Eduardo Eyraes for the script evaluation.pl to test the performances of selenoprofiles. Finally, a special thanks to Charles E. Chapple for his script alignthingie.pl and for endless support.

Funding: This work was supported by grant BIO2006-03380 from the Spanish Ministerio de Educacion y Ciencia, and by grant 1U54HG004555 from the NIH/NHGRI.

REFERENCES

- Allmang, C., Wurth, L., and Krol, A. (2009). The selenium to selenoprotein pathway in eukaryotes: more molecular partners than anticipated. *Biochimica et Biophysica*

- Acta (BBA)-General Subjects*, **1790**(11), 1415–1423.
- Altschul, S., Madden, T., Schaffer, A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D. (1997). Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic acids research*, **25**(17), 3389.
- Birney, E., Clamp, M., and Durbin, R. (2004). GeneWise and Genomewise. *Genome research*, **14**(5), 988–995.
- Burset, M. and Guigó, R. (1996). Evaluation of gene structure prediction programs. *Genomics*, **34**(3), 353–67.
- Cassago, A., Rodrigues, E. M., Prieto, E. L., Gaston, K. W., Alfonso, J. D., Iribar, M. P., Berry, M. J., Cruz, A. K., and Thiemann, O. H. (2006). Identification of Leishmania selenoproteins and SECIS element. *Molecular and biochemical parasitology*, **149**(2), 128–34.
- Castellano, S., Novoselov, S. V., Kryukov, G. V., Lescure, A., Blanco, E., Krol, A., Gladyshev, V. N., and Guigó, R. (2004). Reconsidering the evolution of eukaryotic selenoproteins: a novel nonmammalian family with scattered phylogenetic distribution. *EMBO reports*, **5**(1), 71–7.
- Castellano, S., Lobanov, A., Chapple, C., Novoselov, S., Albrecht, M., Hua, D., Lescure, A., Lengauer, T., Krol, A., Gladyshev, V., and Others (2005). Diversity and functional plasticity of eukaryotic selenoproteins: identification and characterization of the SelJ family. *Proceedings of the National Academy of Sciences of the United States of America*, **102**(45), 16188.
- Castellano, S., Gladyshev, V., Guigo, R., and Berry, M. (2008). SelenoDB 1.0: a database of selenoprotein genes, proteins and SECIS elements. *Nucleic acids research*, **36**(Database issue), D332–8.
- Castellano, S., Andres, A., Bosch, E., Bayes, M., Guigo, R., and Clark, A. (2009). Low exchangeability of selenocysteine, the 21st amino acid, in vertebrate proteins. *Molecular biology and evolution*, **26**(9), 2031.
- Chapple, C. E. and Guigó, R. (2008). Relaxation of selective constraints causes independent selenoprotein extinction in insect genomes. *PLoS ONE*, **3**.
- Chapple, C. E., Guigó, R., and Krol, A. (2009). SECISal, a web-based tool for the creation of structure-based alignments of eukaryotic SECIS elements. *Bioinformatics (Oxford, England)*, **25**(5), 674–5.
- Copeland, P., Stepanik, V., and Driscoll, D. (2001). Insight into mammalian selenocysteine insertion: domain structure and ribosome binding properties of Sec insertion sequence binding protein 2. *Molecular and cellular biology*, **21**(5), 1491.
- Driscoll, D. M. and Chavatte, L. (2004). Finding needles in a haystack. In silico identification of eukaryotic selenoprotein genes. *EMBO reports*, **5**(2), 140–1.
- Gromer, S., Eubel, J. K., Lee, B. L., and Jacob, J. (2005). Human selenoproteins at a glance. *Cellular and molecular life sciences : CMLS*, **62**(21), 2414–37.
- Grundner-Culemann, E., Martin, G. W., Harney, J. W., and Berry, M. J. (1999). Two distinct SECIS structures capable of directing selenocysteine incorporation in eukaryotes. *RNA (New York, N.Y.)*, **5**(5), 625–35.
- Guigó, R., Flicek, P., Abril, J., Reymond, A., and J (2006). EGASP: the human ENCODE genome annotation assessment project. *Genome biology*, **7** Suppl 1, S2.1—S231.
- Harrow, J., Nagy, A., Reymond, A., Alioto, T., Patthy, L., Antonarakis, S., and Guigó, R. (2009). Identifying protein-coding genes in genomic sequences. *Genome biology*, **10**(1), 201.
- Hatfield, D., Carlson, B., Xu, X., Mix, H., and Gladyshev, V. (2006). Selenocysteine incorporation machinery and the role of selenoproteins in development and health. *Progress in nucleic acid research and molecular biology*, **81**, 97–142.
- Jiang, L., Liu, Q., and Ni, J. (2010). In silico identification of the sea squirt selenoproteome. *BMC genomics*, **11**(1), 289.
- Krol, A. (2002). Evolutionarily different RNA motifs and RNA-protein complexes to achieve selenoprotein synthesis. *Biochimie*, **84**(8), 765–774.
- Kryukov, G., Kryukov, V., and Gladyshev, V. (1999). New mammalian selenocysteine-containing proteins identified with an algorithm that searches for selenocysteine insertion sequence elements. *Journal of Biological Chemistry*, **274**(48), 33888.
- Li, M., Huang, Y., and Xiao, Y. (2009). A Method for Identification of Selenoprotein Genes in Archaeal Genomes. *Genomics, Proteomics & Bioinformatics*, **7**(1-2), 62–70.
- Lobanov, A., Gromer, S., Salinas, G., and Gladyshev, V. (2006a). Selenium metabolism in Trypanosoma: characterization of selenoproteomes and identification of a Kinetoplastida-specific selenoprotein. *Nucleic acids research*, **34**(14), 4012.
- Lobanov, A., Delgado, C., Rahlfs, S., Novoselov, S., Kryukov, G., Gromer, S., Hatfield, D., Becker, K., and Gladyshev, V. (2006b). The plasmodium selenoproteome. *Nucleic acids research*, **34**(2), 496.
- Lobanov, A. V., Hatfield, D. L., and Gladyshev, V. N. (2009). Eukaryotic selenoproteins and selenoproteomes. *Biochimica et biophysica acta*, **1790**(11), 1424–8.
- Milinkovitch, M. C., Helaers, R., Depiereux, E., Tzika, A. C., and Gabaldón, T. (2010). 2x genomes - depth does matter. *Genome biology*, **11**(2), R16.
- Notredame, C., Higgins, D. G., and Heringa, J. (2000). T-Coffee: A novel method for fast and accurate multiple sequence alignment. *Journal of molecular biology*, **302**(1), 205–17.
- Novoselov, S., Rao, M., Onoshko, N., Zhi, H., Kryukov, G., Xiang, Y., Weeks, D., Hatfield, D., and Gladyshev, V. (2002). Selenoproteins and selenocysteine insertion system in the model plant cell system, Chlamydomonas reinhardtii. *The EMBO Journal*, **21**(14), 3681.
- Novoselov, S., Hua, D., Lobanov, A., and Gladyshev, V. (2006). Identification and characterization of Fep15, a new selenocysteine-containing member of the Sep15 protein family. *Biochemical Journal*, **394**(Pt 3), 575.
- Novoselov, S., Lobanov, A., Hua, D., Kasaikina, M., Hatfield, D., and Gladyshev, V. (2007). A highly efficient form of the selenocysteine insertion sequence element in protozoan parasites and its use in mammalian cells. *Proceedings of the National Academy of Sciences of the United States of America*, **104**(19), 7857–62.
- Obata, T. and Shiraiwa, Y. (2005). A novel eukaryotic selenoprotein in the haptophyte alga Emiliania huxleyi. *Journal of Biological Chemistry*, **280**(18), 18462.
- Palenik, B., Grimwood, J., Aerts, A., Rouzé, P., Salamov, A., Putnam, N., Dupont, C., Jorgensen, R., Derelle, E., Rombauts, S., Zhou, K., Otillar, R., Merchant, S. S., Podell, S., Gaasterland, T., Napoli, C., Gendler, K., Manuell, A., Tai, V., Vallon, O., Piganeau, G., Jancek, S., Heijde, M., Jabbari, K., Bowler, C., Lohr, M., Robbins, S., Werner, G., Dubchak, I., Pazour, G. J., Ren, Q., Paulsen, I., Delwiche, C., Schmutz, J., Rokhsar, D., Van De Peer, Y., Moreau, H., and Grigoriev, I. V. (2007). The tiny eukaryote Ostreococcus provides genomic insights into the paradox of plankton speciation. *Proceedings of the National Academy of Sciences of the United States of America*, **104**(18), 7705–10.
- Sayers, E. W., Barrett, T., Benson, D. A., Bolton, E., Bryant, S. H., Canese, K., Chetvernin, V., Church, D. M., Dicuccio, M., Federhen, S., Feolo, M., Geer, L. Y., Helmberg, W., Kapustin, Y., Landsman, D., Lipman, D. J., Lu, Z., Madden, T. L., Madej, T., Maglott, D. R., Marchler-Bauer, A., Miller, V., Mizrahi, I., Ostell, J., Panchenko, A., Pruitt, K. D., Schuler, G. D., Sequeira, E., Sherry, S. T., Shumway, M., Sirotkin, K., Slotta, D., Souvorov, A., Starchenko, G., Tatusova, T. A., Wagner, L., Wang, Y., John Wilbur, W., Yaschenko, E., and Ye, J. (2010). Database resources of the National Center for Biotechnology Information. *Nucleic acids research*, **38**(Database issue), D5–16.
- Shchedrina, V., Novoselov, S., Malinouski, M., and Gladyshev, V. (2007). Identification and characterization of a selenoprotein family containing a diselenide bond in a redox motif. *Proceedings of the National Academy of Sciences*, **104**(35), 13919.
- Slater, G. S. C. and Birney, E. (2005). Automated generation of heuristics for biological sequence comparison. *BMC Bioinformatics*, **6**, 31.
- Xu, X., Carlson, B., Irons, R., Mix, H., Zhong, N., Gladyshev, V., and Hatfield, D. (2007). Selenophosphate synthetase 2 is essential for selenoprotein biosynthesis. *Biochemical Journal*, **404**(Pt 1), 115.
- Zhang, Y. and Gladyshev, V. N. (2005). An algorithm for identification of bacterial selenocysteine insertion sequence elements and selenoprotein genes. *Bioinformatics (Oxford, England)*, **21**(11), 2580–9.

Supplementary Data to:

Selenoprofiles: profile-based scanning of eukaryotic genome sequences for selenoprotein genes

Marco Mariotti and Roderic Guigó

Section S1: patterns used with SECISearch

We report here the patterns used with SECISearch in the current implementation of selenoprofiles. The syntax is the one used by PatScan, which is run under the hood by SECISearch. We are currently working to improve the patterns in terms of both specificity and sensitivity, so these may change soon.

Standard:

$r1=\{au,ua,gc,gc,gu,ug\}$ NNNNNNNNNN $p1=7...7$ 3...13 ATGAN $p2=10...13$ AA (4...12 | 0...3 $p3=3...6$ 3...6 $r1\sim p3$ 0...3) ($r1\sim p2[2,1,1]$ NGAN | $r1\sim p2[2,1,0]$ NNGAN) 3...10 $r1\sim p1[1,1,1]$ NNNNNNNNNN

Non-Standard:

$r1=\{au,ua,gc,gc,gu,ug\}$ NNNNNNNNNN $p1=7...7$ 3...13 NNGAN $p2=10...13$ NN (4...13 | 0...2 $p3=3...4$ 3...4 $r1\sim p3$ 0...2) ($r1\sim p2[1,1,1]$ NGAN | $r1\sim p2[1,1,0]$ NNGAN) 3...10 $r1\sim p1[1,1,1]$ NNNNNNNNNN

Twilight:

$r1=\{au,ua,gc,gc,gu,ug\}$ NNNNNNNNNN $p1=7...7$ 3...13 NTGAN $p2=10...13$ (AR | CC) (4...12 | 0...3 $p3=3...6$ 3...6 $r1\sim p3$ 0...3) ($r1\sim p2[2,1,1]$ NGAN | $r1\sim p2[2,1,0]$ NNGAN) 3...10 $r1\sim p1[1,1,1]$ NNNNNNNNNN

Table S2: List of releases of the Ensembl core databases used in this work. The genome release is 52 for all species except *Vicugna Pacos* for which is 51.

Species name	Ensembl core database release
<i>Aedes aegypti</i>	aedes_aegypti_core_52_1d
<i>Anopheles gambiae</i>	anopheles_gambiae_core_52_3k
<i>Bos taurus</i>	bos_taurus_core_52_4b
<i>Caenorhabditis elegans</i>	caenorhabditis_elegans_core_52_190
<i>Canis familiaris</i>	canis_familiaris_core_52_2j
<i>Cavia porcellus</i>	cavia_porcellus_core_52_3a
<i>Ciona intestinalis</i>	ciona_intestinalis_core_52_2l
<i>Ciona savignyi</i>	ciona_savignyi_core_52_2h
<i>Danio rerio</i>	danio_rerio_core_52_7e
<i>Dasypus novemcinctus</i>	dasypus_novemcinctus_core_52_1h
<i>Dipodomys ordii</i>	dipodomys_ordii_core_52_1a
<i>Drosophila melanogaster</i>	drosophila_melanogaster_core_52_54a
<i>Echinops telfairi</i>	echinops_telfairi_core_52_1g
<i>Equus caballus</i>	equus_caballus_core_52_2b
<i>Erinaceus europaeus</i>	erinaceus_europaeus_core_52_1e
<i>Felis catus</i>	felis_catus_core_52_1f
<i>Gallus gallus</i>	gallus_gallus_core_52_2j
<i>Gasterosteus aculeatus</i>	gasterosteus_aculeatus_core_52_1i
<i>Gorilla gorilla</i>	gorilla_gorilla_core_52_1
<i>Homo sapiens</i>	homo_sapiens_core_52_36n
<i>Loxodonta africana</i>	loxodonta_africana_core_52_1g
<i>Macaca mulatta</i>	macaca_mulatta_core_52_10j
<i>Microcebus murinus</i>	microcebus_murinus_core_52_1b
<i>Monodelphis domestica</i>	monodelphis_domestica_core_52_5g
<i>Mus musculus</i>	mus_musculus_core_52_37e
<i>Myotis lucifugus</i>	myotis_lucifugus_core_52_1g
<i>Ochotona princeps</i>	ochotona_princeps_core_52_1c
<i>Ornithorhynchus anatinus</i>	ornithorhynchus_anatinus_core_52_1i
<i>Oryctolagus cuniculus</i>	oryctolagus_cuniculus_core_52_1h
<i>Oryzias latipes</i>	oryzias_latipes_core_52_1h
<i>Otolemur garnettii</i>	otolemur_garnettii_core_52_1e
<i>Pan troglodytes</i>	pan_troglodytes_core_52_21j
<i>Pongo pygmaeus</i>	pongo_pygmaeus_core_52_1c
<i>Procapra capensis</i>	procavia_capensis_core_52_1a
<i>Pteropus vampyrus</i>	pteropus_vampyrus_core_52_1a
<i>Rattus norvegicus</i>	rattus_norvegicus_core_52_34u
<i>Saccharomyces cerevisiae</i>	saccharomyces_cerevisiae_core_52_1i
<i>Sorex araneus</i>	sorex_araneus_core_52_1e
<i>Spermophilus tridecemlineatus</i>	spermophilus_tridecemlineatus_core_52_1g
<i>Takifugu rubripes</i>	takifugu_rubripes_core_52_4k
<i>Tarsius syrichta</i>	tarsius_syrichta_core_52_1a
<i>Tetraodon nigroviridis</i>	tetraodon_nigroviridis_core_52_8b
<i>Tupaia belangeri</i>	tupaia_belangeri_core_52_1f
<i>Tursiops truncatus</i>	tursiops_truncatus_core_52_1a
<i>Vicugna pacos</i>	vicugna_pacos_core_51_1
<i>Xenopus tropicalis</i>	xenopus_tropicalis_core_52_41l

Table S3: Performances indices of selenoprofiles testing on human, drosophila and yeast genome. All families cited in the main article plus MsrA were considered. As reference, we considered the exonic structures annotated in Ensembl Core database, fetching the most similar to each selenoprofiles prediction. All annotations fetched in this way were then checked manually and compared with SelenoDB to make sure that both the selenoproteins were correctly annotated and that all genes were considered. In a some cases (drosophila SelK, SelH, SPS2 and human SelK, SelH, SelS, SelT, SelV, SelW1, TR1, TR2 and TR3) the fetched annotation was not carrying the selenocysteine residue, therefore it was modified to respect the annotation in SelenoDB. For machinery proteins not included in SelenoDB (SecS, PSTK, secp43), the annotations were selected among the selenoprofiles candidates analyzing the gene description in Ensembl. For some drosophila genes no description was available and the gene was selected after a manual sequence analysis. The annotations are split in three sets: selenoproteins, non-Sec homologues and machinery proteins. The selenoprotein set was compared with all selenoprofiles predictions with label "selenocysteine", while the homologues set was compared with the predictions with any other label. The machinery set was compared with all selenoprofiles predictions for machinery protein families.

Sensitivity (SN) and specificity (SP) were computed at the gene, exon, and nucleotide level. At the gene level, the number of false positives (FP) is reported instead of specificity. The exon level indexes are computed considering only the genes that were correctly paired between the predictions and the annotations, while the nucleotide indexes are computed considering everything. The average indexes at the end of the table are computed pulling together all genes for each set.

gene	level	exon	level	nucleotide	level	family, class, gene numbers
SN	FP	SN	SP	SN	SP	
						Homo sapiens
1	0	0	0	1	1	sps-selenoproteins: 1 gene
1	0	0.57	0.75	0.89	1	GPx-selenoproteins: 5 genes
1	0	0.63	0.71	0.98	0.97	DI-selenoproteins: 3 genes
1	0	1	1	1	1	15-kDa-selenoproteins: 1 gene
1	0	1	1	1	1	SelM-selenoproteins: 1 gene
1	0	1	1	1	1	SelH-selenoproteins: 1 gene
1	0	0.9	0.9	1	0.97	SelI-selenoproteins: 1 gene
1	1	0.6	0.75	1	0.5	SelK-selenoproteins: 1 gene
1	0	0.83	0.91	0.89	1	SelN-selenoproteins: 1 gene
1	0	1	1	1	1	SelO-selenoproteins: 1 gene
1	0	1	1	1	1	SelP-selenoproteins: 1 gene
1	0	1	1	1	1	SelR-selenoproteins: 1 gene
1	2	1	1	1	0.46	SelS-selenoproteins: 1 gene
1	1	0.8	0.8	0.96	0.53	SelT-selenoproteins: 1 gene
0.5	1	0.8	0.67	0.74	0.79	SelV-selenoproteins: 2 genes
1	0	0.91	0.89	0.99	0.92	TR-selenoproteins: 3 genes
1	2	0.88	0.88	0.96	0.4	sps-homologues: 1 gene
1	0	0.45	0.56	0.72	0.99	GPx-homologues: 3 genes
1	0	1	1	1	1	MsrA-homologues: 1 gene
/	2	/	/	/	/	SelJ-homologues: 0 genes
/	2	/	/	/	/	SelK-homologues: 0 genes
1	0	0.82	0.9	0.86	1	SelR-homologues: 2 genes
/	1	/	/	/	/	SelT-homologues: 0 genes
1	0	0.78	0.78	0.99	0.95	SelU-homologues: 3 genes
0	0	0	0	0	0	SelV-homologues: 1 gene
/	2	/	/	/	/	TR-homologues: 0 genes
1	1	0.76	0.81	0.99	0.43	sbp2-machinery: 1 gene
1	0	0.5	0.5	0.79	0.81	pstk-machinery: 1 gene
1	0	0.22	0.5	0.32	0.93	secp43-machinery: 1 gene
1	0	1	1	1	1	SecS-machinery: 1 gene
1	0	1	1	1	1	eEFsec-machinery: 1 gene

Drosophila melanogaster						
1	0	0.25	0.25	0.91	1	sps-selenoproteins: 1 gene
1	0	0	0	0.58	0.89	SelH-selenoproteins: 1 gene
1	0	1	1	1	1	SelK_insect-selenoproteins: 1 gene
1	0	0	0	0.99	1	sps-homologues: 1 gene
1	1	0.33	0.5	0.68	0.51	GPx-homologues: 1 gene
1	0	0	0	0.3	0.95	MsrA-homologues: 1 gene
1	0	0.33	0.5	0.92	1	15-kDa-homologues: 1 gene
/	1	/	/	/	/	SelM-homologues: 0 genes
1	0	0	0	0.92	0.88	SelH-homologues: 2 genes
1	3	0.5	0.4	0.88	0.33	SelI-homologues: 1 gene
/	1	/	/	/	/	SelK-homologues: 0 genes
0	0	0	0	0	0	SelK_insect-homologues: 1 gene
1	0	0.75	0.75	1	0.95	SelR-homologues: 1 gene
1	0	1	1	1	1	SelT-homologues: 1 gene
/	1	/	/	/	/	SelV-homologues: 0 genes
1	2	0.6	0.6	0.92	0.71	TR-homologues: 2 genes
1	0	1	1	1	1	sbp2-machinery: 1 gene
1	1	0	0	1	0.54	pstk-machinery: 1 gene
1	1	0.5	0.33	0.94	0.54	secp43-machinery: 1 gene
1	0	0.5	0.5	1	0.95	SecS-machinery: 1 gene
1	0	1	1	1	1	eEFsec-machinery: 1 gene
Saccharomyces cerevisiae						
1	0	0	0	0.97	1	GPx-homologues: 3 genes
1	0	0	0	0.61	1	MsrA-homologues: 1 gene
1	0	0	0	0.26	1	SelO-homologues: 1 gene
1	1	0	0	0.62	0.39	SelR-homologues: 1 gene
/	3	/	/	/	/	TR-homologues: 0 genes
/	1	/	/	/	/	pstk-machinery: 0 genes
Average (FP column refers to the total number)						
0.96	5	0.81	0.85	0.94	0.91	selenoproteins
0.97	22	0.57	0.6	0.8	0.58	homologues
1	4	0.71	0.77	0.93	0.68	machinery

Section S4: Exonerate vs genewise

In the following table, we report the global performance indices when we force the pipeline to choose always the exonerate or always the genewise prediction. When the standard routine of selenoprofiles is used (one of the two predictions is chosen according to the criteria detailed in the text) the indices improve or are the same.

gene	level	exon	level	nucleotide	level	class
SN	FP	SN	SP	SN	SP	
Average (FP column refers to the total number) choosing EXONERATE						
0.89	3	0.78	0.83	0.86	0.93	selenoproteins
0.9	14	0.6	0.63	0.73	0.65	homologues
0.9	4	0.74	0.72	0.91	0.68	machinery
Average (FP column refers to the total number) choosing GENEWISE						
0.96	5	0.8	0.85	0.94	0.91	selenoproteins
0.93	20	0.5	0.56	0.76	0.59	homologues
0.9	4	0.67	0.76	0.82	0.67	machinery

We observe that genewise is generally performing better than exonerate. Nonetheless, genewise is much slower than exonerate (it would not be feasible to use the cyclic procedure for genewise), so we believe that the best way to combine them is to use exonerate to outline the gene boundaries and genewise to refine the prediction. Anyway, since genewise appears to be more sensitive than exonerate, we created the `genewise_to_be_sure` routine (see text in the main manuscript) to ensure that we do not lose any potential candidates that would be missed by exonerate but caught by genewise. Also, in our experience genewise crashes systematically for some predictions (although it never crashed for the predictions in the testing set). We believe this is due to the fact that it was never tested with our particular scoring scheme, which may confound its computation. When this happens, selenoprofiles uses exonerate prediction instead, and this is another advantage of having two predictions available.

Section S5: Discussion of false positives

1. Selenocysteine labelled

In the human genome, 5 genes for which no annotation was found were predicted and labelled as “selenocysteine”. One belongs to the SelT family. This is characterized by a single-exon structure, and no potential SECIS was identified downstream. An additional analysis revealed that the conservation of the coding sequence extends in the 5' side for an additional portion respect to selenoprofiles prediction. This extension contains a frameshift. All these facts make us believe that this is a recent retro-transcribed pseudogene.

Two selenocysteine containing SelS genes were predicted. In both cases a poor scoring SECIS element was found downstream of the predicted coding sequence. The SelS family is characterized by domains of repetitive sequences, rich in lysine, glutamic acid and glycine. These domains causes the profile to hit the genome in a lot of locations. In both predicted genes, the conservation with the profile is too poor to conclude that these are real genes: excluding the regions of repetitive sequence, we found no significant similarity with any other known protein. It is very likely that these predictions have said selenoprotein features just by chance.

Then, a selenocysteine containing SelK gene was predicted. This gene is characterized by a single-exon structure, and two poor scoring SECIS elements were found downstream. No annotation corresponding to this gene was found in Ensembl. Nonetheless, a search with blast found an human hypothetical protein (gi code: 169213282), matching with 100% identity the selenoprofiles prediction but stopping at the UGA position. A blast search in ncbi human EST dataset resulted in no perfect matches, suggesting that this genomic region is not transcribed. The single exon structure and the absence of transcription suggest the occurrence of a retro-transcribed pseudogene.

Lastly, a selenocysteine containing SelV gene was predicted, consisting of two exons with two poor scoring SECIS elements downstream. This corresponds to the Ensembl pseudogene ENSG00000215900. Searching ncbi human ESTs, we found no evidence of transcription. We think that this is most likely a pseudogene, too.

2. Selenocysteine machinery proteins

For these proteins, 4 false positives were predicted in total in the human, fly and yeast genome by selenoprofiles. Two false PSTKs were predicted, one in drosophila and one in yeast. The PSTK proteins share a domain with high similarity with another protein family, KTI12, and this causes selenoprofiles to find also KTI12 proteins when searching the PSTK profile in genomes.

One false SECP43 protein was predicted in drosophila. This is actually a portion of the protein Rox8 (or RE71384p), since it shares a nucleotide binding domain with SECP43.

Lastly, the human protein SBP2-like is found using the SBP2 profile. These two proteins diverged recently, during vertebrate evolution (see Donovan et al, “Evolutionary history of selenocysteine incorporation from the perspective of SECIS binding proteins”, BMC evolutionary biology, 2009). They share high sequence similarity and, possibly, they are also functionally linked.

2.2 SECISearch3 and Seblastian

Selenoprofiles revealed to be powerful, for it can exploit profiles from a range of species to find homologous protein genes. Similar tools exist also for RNA motif finding. Typically they model positional correlation in sequences, and/or use an underlying structural model. SECISearch3 is the product of testing and combining these programs to search for our favourite RNA motif: eukaryotic SECIS elements. The program Seblastian combines SECISearch3 in a selenoprotein gene finding pipeline, using homology information of selenocysteine or cysteine homologues in a large protein database. I completed this project during my stay at Vadim Gladyshev's lab, in Boston. Thus, I had the luck of having expert eyes guiding me through SECIS predictions, those of the brilliant Alexei Lobanov.

Publication:

Mariotti M, Lobanov AV, Guigó R, Gladyshev VN

SECISearch3 and Seblastian: new tools for prediction of SECIS elements and selenoproteins. Nucleic Acids Research. 2013 Aug 1;41(15):e149. doi: 10.1093/nar/gkt550.

2.2.1 Computational identification of SECIS elements

The prediction of SECIS elements is of key importance for selenoprotein genomics. Its presence, properly positioned in respect to a candidate in-frame UGA, is generally a sufficient argument to be convinced that a selenocysteine is inserted. When I started the PhD, the only program available for eukaryotic SECIS element prediction was SECISearch [Kryukov et al., 2003]. This program is in a way an “ab initio” predictor: target sequences are scanned with fixed patterns, then hits are fold into secondary structures and evaluated thermodynamically. Although extremely useful, the program has some limitations, mainly the lack of a score assigned to candidates, and its dependence on manually written patterns.

In collaboration with Alexei Lobanov and Vadim Gladyshev we created SECISearch3, a new program to predict eukaryotic SECIS elements that outperforms its predecessor. SECISearch3 can combine the pattern based predictions of the original SECISearch with homology based predictions, using a large SECIS alignment as model for covariance and secondary structure search. The programs Infernal [Nawrocki et al., 2009] and covels (<http://selab.janelia.org/software.html>) are used. The set of SECIS to build the model was obtained with Selenoprofiles, run on a large collection of genomes. Knowing the positions of selenoprotein genes, we searched downstream for SECIS elements with the original SECISearch, and with the new methods in development. The final alignment constituting the Infernal model includes 1122 eukaryotic SECIS elements, widely spread over sequenced lineages. The advantage of SECISearch3 over the original SECISearch is not only in sensitivity and specificity, but also in that it provides scores for the matches. This allows to easily adjust filtering to achieve the desired trade-off of sensitivity and specificity. Also, as it is mainly based on two alignment models (for Infernal and covels), it is relatively easy to update and improve when more SECIS sequences are available.

After creating this new tool of SECIS prediction, and having developed a library of functions for parsing and manipulating gene structure predictions, I decided to write a straightforward new program to predict selenoprotein genes: Seblastian. This pipeline runs SECISearch3 as first step to predict potential SECIS in the target, then it searches for selenoprotein coding sequences upstream of each one. Seblastian is homology-based: the target sequence is run as query with blastx against a protein database. Blastx alignments with an in-frame UGA are processed and filtered, then exonerate is used to attempt refining gene structures. When the protein database used is the collection of known proteins with Sec, Seblastian predicts selenoprotein genes homologues to any known selenoprotein family (Sec-to-Sec). When the database is a comprehensive set of proteins (such as ncbi nr), Seblastian can predict also novel selenoproteins, homologous to some known non-Sec protein family, with a Sec-to-Cys alignment. After validating our new methods, we used them to predict new selenoproteins in basal eukaryotic genomes, where we expected selenoproteins to be discovered yet. In our analysis of the best candidates we describe the phylogeny of the selenoprotein family AhpC, shared by bacteria

and several eukaryotic lineages, and present as Cys homologue in human.

Finally, we built a webserver to allow public access to run our programs. We hope this will allow the discovery of new selenoprotein sequences by users worldwide, even without bioinformatics expertise. It is hosted both at

<http://gladyshevlab.org/SelenoproteinPredictionServer> and at

<http://big.crg.cat/services/sebastian>

2.2.2 SECISearch3 and Sebastian paper

We include here the paper published in Nucleic Acids Research in 2013. The paper contains the following supplementary material sections, some of which are also included in this thesis:

1. SECISearch1 patterns (not included here)
2. Building an Infernal model for eukaryotic SECIS
3. Details of testing SECIS prediction methods (not included here)
4. Python procedures for filtering and scoring
5. Details on selenoprotein candidates (not included here)
6. Analysis of the AhpC selenoprotein family

The full supplementary sections can be accessed online at:

<http://nar.oxfordjournals.org/content/41/15/e149/suppl/DC1>

Mariotti M, Lobanov AV, Guigo R, Gladyshev VN. [SECISearch3 and Sebastian: new tools for prediction of SECIS elements and selenoproteins](#). Nucleic Acids Res. 2013; 41(15): e149. DOI: 10.1093/nar/gkt550

SECISearch3 and Sebastian: new tools for prediction of SECIS elements and selenoproteins

Marco Mariotti^{1,2}, Alexei V. Lobanov¹, Roderic Guigo^{2,*} and Vadim N. Gladyshev^{1,*}

¹Division of Genetics, Department of Medicine, Brigham and Womens Hospital and Harvard Medical School, 77 Avenue Louis Pasteur, 02115, Boston, MA, USA and ²Bioinformatics and Genomics Programme, Centre for Genomic Regulation (CRG), Dr. Aiguader 88, 08003 Barcelona, Spain and Universitat Pompeu Fabra (UPF), 08003, Barcelona, Spain

Received April 17, 2013; Revised May 22, 2013; Accepted May 25, 2013

ABSTRACT

Selenoproteins are proteins containing an uncommon amino acid selenocysteine (Sec). Sec is inserted by a specific translational machinery that recognizes a stem-loop structure, the SECIS element, at the 3' UTR of selenoprotein genes and recodes a UGA codon within the coding sequence. As UGA is normally a translational stop signal, selenoproteins are generally misannotated and designated tools have to be developed for this class of proteins. Here, we present two new computational methods for selenoprotein identification and analysis, which we provide publicly through the web servers at <http://gladyshevlab.org/SelenoproteinPredictionServer> or <http://sebastian.crg.es>. SECISearch3 replaces its predecessor SECISearch as a tool for prediction of eukaryotic SECIS elements. Sebastian is a new method for selenoprotein gene detection that uses SECISearch3 and then predicts selenoprotein sequences encoded upstream of SECIS elements. Sebastian is able to both identify known selenoproteins and predict new selenoproteins. By applying these tools to diverse eukaryotic genomes, we provide a ranked list of newly predicted selenoproteins together with their annotated cysteine-containing homologues. An analysis of a representative candidate belonging to the AhpC family shows how the use of Sec in this protein evolved in bacterial and eukaryotic lineages.

INTRODUCTION

Selenoproteins are a class of proteins that contain the amino acid selenocysteine (Sec), known as the 21st amino acid in the genetic code. Sec is inserted

co-translationally by recoding a UGA codon, which normally serves as a stop signal (1–4). Owing to this dual function of the UGA codon, selenoproteins are generally missed or mispredicted in genome projects, and their annotation has to be carried out with *ad hoc* developed tools. Since the beginning of the genomic era, a considerable effort has been placed at developing computational methods for selenoprotein prediction, including the detection and analysis of eukaryotic, archaeal and prokaryotic SECIS elements, and the identification of selenoproteins in genomes *ab initio* or by homology (5–17).

In this study, we present two new computational methods for selenoprotein prediction and analysis. SECISearch3 is a pipeline for predicting SECIS elements that significantly outperforms its predecessor SECISearch. Sebastian is a new method for the identification of selenoprotein genes in sequence databases that uses SECISearch3 and then identifies selenoprotein sequences upstream of the detected SECIS elements. Both services can be freely run through web servers at <http://gladyshevlab.org/SelenoproteinPredictionServer> and <http://sebastian.crg.es>.

Eukaryotic SECIS elements

SECIS elements are stem-loop structures that specify recoding of a UGA codon from its canonical translation termination function to a non-canonical one, Sec insertion. SECIS elements are completely different in eukaryotes, bacteria and archaea and may also be located in different regions of selenoprotein genes (18). Here, we focus on eukaryotic SECIS elements. These structures can be classified into two classes, type I and type II, differing in the presence of an additional helix in type 2 SECIS elements (19). The highest sequence conservation in SECIS elements is found in the core (or quartet), which forms a kink-turn motif through the non-canonical pairing of AG-GA. The core bears the conserved sequence UGAN/KGAW. Additionally, a stretch of

*To whom correspondence should be addressed. Tel: +1 617 5255122; Fax: +1 617 5255147; Email: vgladyshev@rics.bwh.harvard.edu
Correspondence may also be address to Roderic Guigo. Tel: +34 93 3160110; Fax: +34 93 3969983; Email: roderic.guigo@crg.cat

conserved nucleotides are found in the apical loop, typically adenines (or cytosines in a few cases). The structural parts of SECIS elements are also found to be within specific length constraints [see (13) for a summary], although the precise definition of these boundaries has changed during the years, particularly with the analysis of these structures in newly sequenced eukaryotes. The distance between the Sec-UGA and the SECIS element varies substantially, e.g. from ~ 200 to ~ 5200 nt in mammalian selenoproteins. The minimum functional distance was tested in human embryonic kidney line 293 cells for deiodinase 1 (20), and it was found to be between 51 and 111 nt.

The original SECISearch

The most widely used method for SECIS prediction has been SECISearch (9). This method relies on sequence patterns (searched with PatScan <http://blog.theseed.org/servers/2010/07/scan-for-matches.html>) to identify initial hits in the query sequence, which are then processed and filtered. Several SECIS patterns were developed and optimized in the past 10 years. All patterns model a partition of the SECIS in helix1, core, loop1, helix2, conserved apical nucleotides, loop2 and optionally helix3 (only in type II SECIS elements). Thus, these criteria require the hits to have specific nucleotides in the core and in the apical nucleotides and to have stretches of nucleotides of a defined length that can pair to form the stems. The various patterns differ in the required conserved nucleotides and in the length and pairing rules allowed in stems. Currently, the patterns used by SECISearch are the following: strict, default, loose and loosest (loose+) (see Supplementary Material S1). The hits by PatScan are fed into RNAfold from the ViennaRNA package (21,22), which predicts their secondary structure and thermodynamic stability. This is used to filter out unstable structures. Finally, another filter analyzes the predicted secondary structure and the pattern-based partition of the candidate and filters out unlikely candidates with certain structural characteristics (e.g. Y-shaped or O-shaped). Although SECISearch has been extremely useful to selenoprotein research, it has some limitations. The main one is its dependence on sequence patterns. The patterns have been manually built to accommodate SECIS elements. As a result, whenever a species from a newly sequenced distant lineage is analyzed, the patterns had to be modified to optimize the searches. The current routine identifies a first set of *bona fide* selenoproteins by running SECISearch with the existing patterns or by homology to known selenoproteins with the tools such as Tblastn [or lately, with the more sophisticated Selenoprofiles (15)]. Then, a new pattern is developed that includes the *bona fide* selenoproteins while keeping the number of predictions under a manageable level, and the genome search is then done with this pattern. Another limitation of the original SECISearch is that it lacks the assignment of a score to the candidates.

MATERIALS AND METHODS

New SECIS prediction methods

In the past several years, several programs have emerged for family-based prediction of RNA structures. To build a better tool for SECIS prediction, we tested three available methods: Infernal, Covels and Erpin. In most cases, we built our own SECIS models.

The program Infernal (Inference of RNA alignments) (23) 'is an implementation of a special case of profile stochastic context-free grammars called covariance models (CMs). A CM is like a sequence profile, but it scores a combination of sequence consensus and RNA secondary structure consensus'. Infernal can be used to build a CM model from a secondary structure alignment and then search the model in nucleotide databases. To obtain a large set of SECIS elements for the alignment, we exploited an extensive collection of *bona fide* selenoprotein sequences predicted with Selenoprofiles (15). Initially, SECISearch was run on sequences downstream of all selenoprotein coding sequences. This set was used to build a first, very rough alignment, forcing the structural parts to be aligned (stem1, core, loop1, etc.) as shown in Supplementary Material S2. A consensus secondary structure was manually assigned to this alignment, based on the known pairings (part1 of helix1 with part2, and so on). The resulting secondary structure alignment was inspected with RALEE, a RNA alignment editor (24), to identify and extract the sequences satisfying the consensus secondary structure assigned, i.e. to obtain a subset of well-aligned sequences. This subset was used to build an Infernal model, and the Infernal program *cmalign* was used to align additional SECIS elements to the model. As this was a template-based alignment, the resulting quality was much superior. This procedure was used iteratively, inspecting manually the alignment to add or remove sequences, until we obtained our final model: a secondary structure alignment of 1122 SECIS elements from diverse eukaryotic lineages. We use this model with the Infernal program *cmsearch* as a new method to predict SECIS elements. Infernal computes two types of scores for each candidate: a bit-score, expressing how well it fits in the model, and an *E*-value, expressing how many alignments with the same or better bit-score are expected by chance searching the current target. We decided to use a bit-score-based filtering, for this is not dependent on the target size.

The program Covels (<http://selab.janelia.org/software.html>) is also based on variance models, but it does not model secondary structure explicitly. We built a Covels model as described in the program manual. For this purpose, 300 SECIS elements were manually aligned to produce the best results. Sequences were extracted from RefSeq NCBI database. Because our goal was to generate a 'consensus model', we did not consider here SECIS elements from organisms (such as *Ostreococcus* or *Toxoplasma* species) in which these structures have lineage-specific characteristics. In our study, we found that regions flanking the core lack the consensus (data not shown), therefore, including them in the model would lower the sensitivity. Thus, we included only the

most functionally relevant part of their structure, beginning from the core. Like Infernal, Covels predictions include a bit-score that can be used for filtering. The recommended threshold value is 15. However, it should be taken into account that for SECIS elements not conforming to this model the score could be significantly lower.

The program Erpin (25) is another RNA motif search program. Given a secondary structure-based alignment, it infers a structural profile, which is then searched in the target database using a dynamic programming algorithm. Erpin also provides scores for the matches. In the case of Erpin, we found a SECIS model provided by the authors; therefore, we proceeded to the testing phase with this model. We noticed early on that a limitation of this program is that gaps are not allowed in the alignment model nor in the matches with the profile, thus any motif with insertions or deletions in respect to the model is missed.

RESULTS AND DISCUSSION

Testing SECIS prediction methods

To test the performance of the three methods and relate them to SECISearch, we first built a set of reliable SECIS elements from as diverse lineages as possible. The set contained 116 SECIS elements: 1 from *Caenorhabditis elegans* (11), 8 from *Chlamydomonas reinhardtii* (26), 5 from *Toxoplasma gondii* (27), 4 from *Plasmodium falciparum* (28), 4 from *Dictyostelium purpureum*, 3 from *Drosophila melanogaster* (8), 26 from *Homo sapiens* (9), 25 from *Mus musculus* and 40 from *Danio rerio* (see Supplementary Material S3 for details). We then evaluated all SECIS prediction methods when applied to the full genomes of these organisms. We computed an F-score (20) of the methods, which combines sensitivity and precision into a single

measure, giving 20 times more importance to sensitivity (the desired trade-off in most SECISearch applications). Results are given in Table 1. When comparing the methods, Infernal with the score threshold of 20 was the best performer. Covels also performed well, with better sensitivity but additional false positives. SECISearch ranked third owing to the low values for sensitivity, and Erpin was the worst performer, owing to its low sensitivity. For all methods, SECIS elements from the non-metazoan eukaryotes were the hardest to predict (see Supplementary Material S3). We also tested the speed of the various methods. SECISearch was the quickest, although the time varied significantly depending on the pattern chosen. Erpin was the slowest, followed by Covels. It should be mentioned that Infernal can reduce its running time depending on the score threshold specified, owing to heuristics it adopts. In this case, a loose threshold was used (score ≥ 5); therefore, its speed was somewhat underestimated.

SECISearch3

Given these results, we built a pipeline that combined the predictions of Infernal, Covels and the original SECISearch. We call the new program SECISearch3 (see Figure 1). The Infernal model is central to the program. It is used not only as a prediction method but also to derive the secondary structure of the predictions by Covels and SECISearch, ensuring consistency. The redundant predictions are then removed, and a procedure of structural refinement is executed. This process compensates for structural inconsistencies owing to the template-based structure assignment of Infernal, particularly improving the pairing near insertions and in the boundaries of helices and loops. After refinement, the thermodynamic stability of the structure is predicted with RNAeval from

Table 1. Testing SECIS prediction methods

	TP	FP	Sn (%)	Pr (%)	FP/Mb	F-score(20)	Speed (min/Mb)	TP after filtering	FP after filtering
Covels.5	114	1 747 455	98.3	0.007	224.54	0.026	33.51	107	*201482
Covels.10	108	188 466	93.1	0.057	24.22	0.184		101	35945
Covels.15	104	16 691	89.7	0.619	2.15	0.660		97	4152
Infernal.10	106	166 085	91.4	0.064	21.34	0.200	6.92	105	50814
Infernal.15	98	9383	84.5	1.034	1.21	0.703		97	5697
Infernal.20	86	485	74.1	15.061	0.06	0.734		85	393
Secisearch.strict	65	20 694	56.0	0.313	2.66	0.388	0.14	60	10557
Secisearch.def	86	110 532	74.1	0.078	14.20	0.220	0.18	76	42719
Secisearch.loose	79	262 710	68.1	0.030	33.76	0.102	3.18	64	*54775
Secisearch.looser	84	2 689 478	72.4	0.003	345.59	0.012	2.62	66	*542199
Erpin.25	70	225 801	60.3	0.031	29.01	0.103	75.37		
Erpin.35	58	3754	50.0	1.522	0.48	0.463			
Erpin.45	43	48	37.1	47.253	0.01	0.371			

The test set consisted of 116 SECIS elements from nine species (see Supplementary Material S3). For Covels, Infernal and Erpin, various score thresholds were considered; different patterns were considered for SECISearch. The two last columns show the effect of the SECISearch3 filter (see text). Erpin is not shown, as it is not included in SECISearch3.

For the methods indicated with a star (asterisk), the number of false positives after filtering was estimated by running the filter only on a subset of the total predictions, to save computational time. TP, number of true positives; FP, number of false positives; Sn, sensitivity (recall); Pr, precision; FP/Mb, average number of false positives per Mb of input sequence; F-score(20), F-score computed with $\beta = 20$; Speed, total run time divided by the total input sequence length (~8 Gb); TP after filtering, true positives passing the SECIS filter; FP after filtering, false positives passing the SECIS filter.

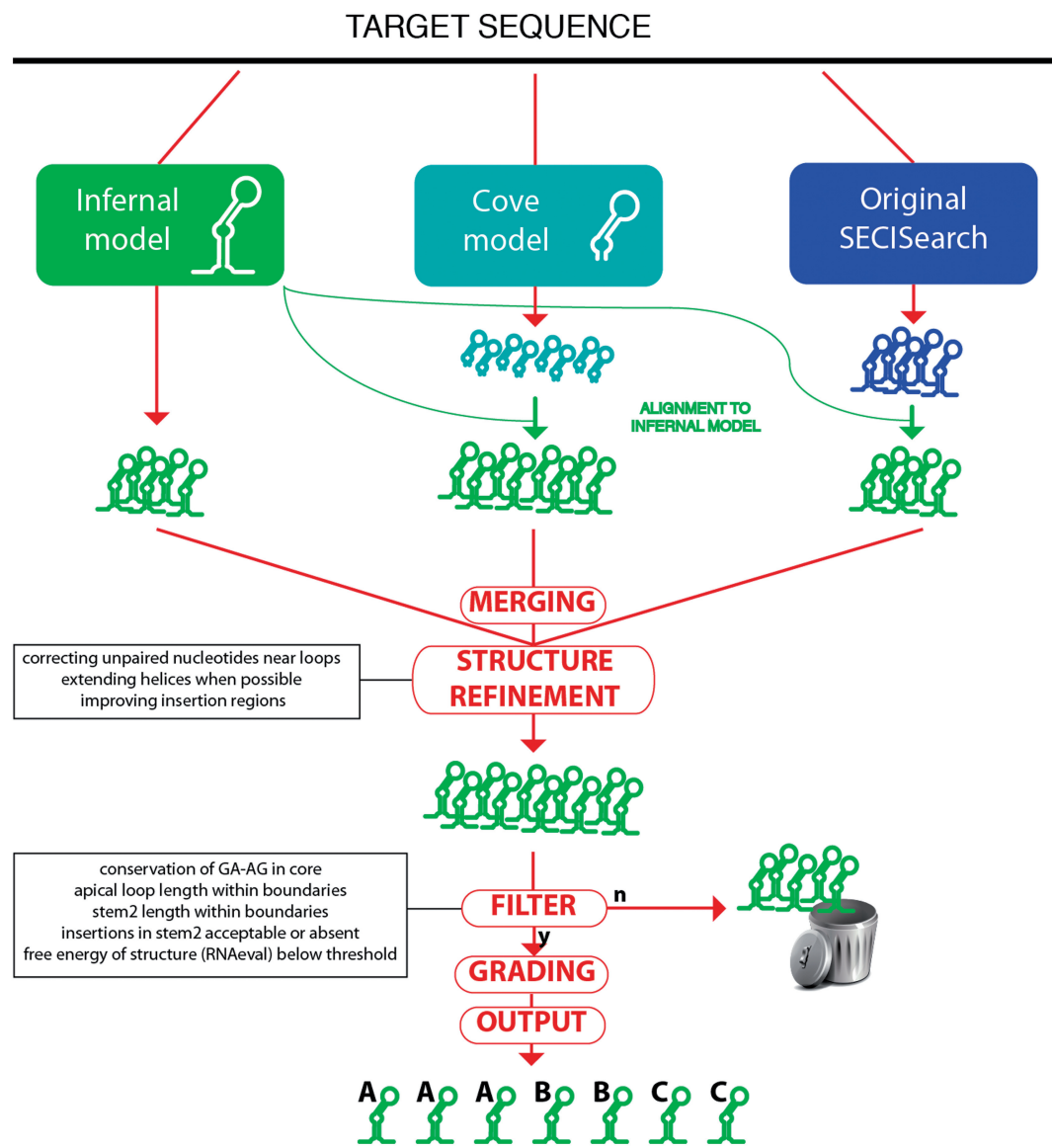


Figure 1. Workflow of the SECISearch3 program.

the Vienna package (21,22). At this point, all predictions are also assigned a score by the Covels model.

Next, a filtering procedure is applied to the candidate SECIS elements. The candidates are discarded if they have any of following features (see the SECISearch filtering section in Supplementary Material S4): core is not included in the prediction, no GA-AG in the core, apical loop is too short or too long, helix2 is too short or too long, too much bending (computed as the difference in number of insertions on the two sides of helix2) and the free energy is too high. The effect of this filter is shown in Table 1 (right column): although true positives remain stable, the number of false positives significantly decreases following the filtering.

Lastly, the remaining candidates are assigned a grade (A, B or C). We included this procedure after inspecting and grading manually hundreds of SECIS elements trying

to incorporate our extensive experience with these structures. The grade depends on several characteristics: the presence of conserved unpaired nucleotides in the apical loop, the bending coefficient for helix2, the Covels score, the presence of mismatches or insertion in key positions (just before or just after the core, or in any two consecutive positions along helix2). For details, see the SECIS grading section in Supplementary Material S4. SECISearch3 may generate graphical output of publication quality: the program RNAplot from the RNAfold package is used with custom settings to highlight the key SECIS features (see Figure 2). We designed SECISearch3 to be as flexible as possible. Any combination of the prediction methods (or any single method) can be run. This allows balancing the trade-off between sensitivity and speed. For example, Covels should be avoided for large databases but may be used to find unusual candidate

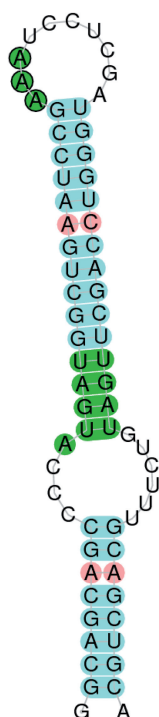


Figure 2. Example of SECISearch3 generated image: SECIS type I of human SelN. The core and the unpaired conserved nucleotides of the SECIS element are highlighted in green, and mismatches in red. SECISearch3 uses internally RNAplot.

SECIS elements in relatively small databases. As default settings, we recommend to use the Infernal model with a score threshold of 10, prioritizing sensitivity.

Sebastian

Based on SECISearch3, we build a new method for selenoprotein gene prediction and analysis: Sebastian. This pipeline automatizes a process that we used to carry out to predict selenoproteins in newly sequenced species (Figure 3). First, all potential SECIS elements are predicted in a target sequence (a genome, for instance), and then the sequences upstream of each SECIS candidate are examined for selenoprotein coding potential. To search for selenoprotein-coding sequences, we use homology information: the sequence upstream of each SECIS is run with Blastx (29) against a comprehensive protein database (Genbank NCBI nr). As Blastx is used to make a gene prediction on the nucleotide sequence, we refer to the proteins annotated in the database as queries and to the nucleotide sequence as the target. The Blastx output is parsed, and, mostly, two types of blast alignments are considered: (i) those in which a Sec in a query protein is aligned with a UGA in the target sequence and (ii) those in which a cysteine in a query is aligned with a UGA in the target. This procedure yields two conceptually different classes of output candidates: known selenoproteins and new selenoprotein homologues of known proteins. The second category includes the candidate selenoproteins for which sequence

homologues exist, but none of them is a selenoprotein (i.e. known protein family, undiscovered selenoprotein family). As the absolute majority of known selenoproteins possess cysteine homologues (30,31), Sebastian is effectively able to predict new selenoproteins. In practice, other types of blast alignments are also kept to ensure maximum sensitivity: for example, all blast hits in which the query has a Sec in its sequence are kept, even if it is not aligned to a UGA in the target sequence. Blast alignments are then filtered, and those with the same query and likely to belong to the same gene are joined. Here, the concept of colinearity is used: if blast hit A is found in the target downstream of blast hit B, and also the portion of the query aligned in blast hit A is downstream of that in blast hit B, they will be joined. A set of joined blast hits constitutes a possibly multiexonic gene prediction.

Sebastian then attempts to improve the gene structure predictions by running the program Exonerate (32). As query, the full sequence of the nr protein in the blast alignment is used. As target, we use the region in the same blast alignment, properly extended: to ensure an optimal choice of the target boundaries, we use the cyclic Exonerate routine (15). At this point, the Exonerate and Blastx predictions for each candidate are compared, and only the best one is kept.

Finally, all candidates must pass a filter (see Sebastian filtering section in Supplementary Material S4). This requires the gene predictions to have the SECIS element properly positioned (downstream from the coding sequence) and not possess pseudogene-like features such as frameshifts or in-frame stop codons apart from the candidate Sec-UGA. Also, candidates are required to possess a convincing pattern of conservation on both sides of the Sec-UGA. Although the vast majority of selenoproteins contain a single Sec, Sebastian procedures and filters were designed to accept also candidates with multiple Sec residues, such as selenoprotein P.

Testing Sebastian

We benchmarked Sebastian using the same data set used for testing SECIS prediction methods. For SECISearch3, we chose Infernal with the score threshold of 15. Our test set was thus limited to the SECIS elements that this method is able to predict. Two separate benchmarks were executed for known selenoproteins and for new selenoproteins.

For known selenoproteins, we ran Sebastian using a modified version of the nr protein database, containing only the protein sequences with at least 1 Sec. This database was also depleted of all sequences coming from any of the tested species, to simulate a run on a newly sequenced species. For new selenoproteins, we used again nr but removed all selenoproteins, thus simulating the situation as if all selenoprotein families were undiscovered (Table 2). The search for known selenoproteins worked well, with sensitivity of ~80% and specificity >90%. We analyzed in detail the false positives for known selenoproteins, as none were expected, as these predictions must feature a good alignment between a candidate and a known selenoprotein, with a Sec to UGA

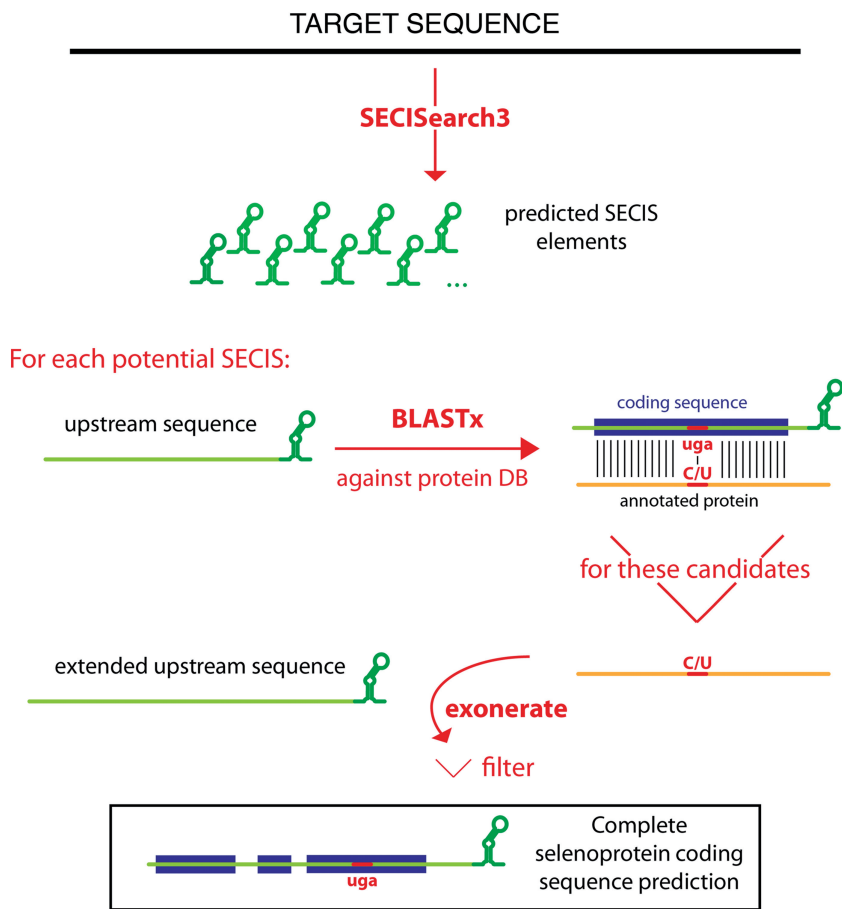


Figure 3. Workflow of the Sebastian program.

Table 2. Testing Sebastian

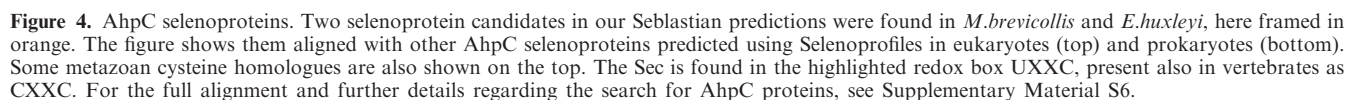
Species	Selenoproteins ^a	Known selenoproteins		New selenoproteins	
		Sn (%)	Pr (%)	Sn (%)	Pr (%)
<i>Caenorhabditis elegans</i>	1	100.00	100.00	0.00	0.00
<i>Chlamydomonas reinhardtii</i>	3	33.33	100.00	0.00	0.00
<i>Danio rerio</i>	32	65.63	100.00	9.38	27.27
<i>Drosophila melanogaster</i>	3	33.33	100.00	66.67	66.67
<i>Homo sapiens</i>	25	96.00	100.00	40.00	21.28
<i>Mus musculus</i>	24	91.67	81.48	33.33	7.84
<i>Toxoplasma gondii</i>	3	66.67	100.00	33.33	100.00
<i>Dictyostelium purpureum</i>	1	100.00	100.00	0.00	0.00
<i>Plasmodium falciparum</i>	2	100.00	100.00	0.00	0.00
Global	94	79.79	93.75	25.53	14.63

The testing was split for known and new selenoproteins, as described in the text.
^aTo test Sebastian independently of SECISearch3, we considered here only the selenoproteins whose SECIS elements were correctly predicted by Infernal with the score threshold of 15. Thus, the number of selenoproteins reported here do not necessarily represent the complete selenoproteome of the species (see Supplementary Material S3 for full sets).
Sn, sensitivity (recall); Pr, precision.

alignment. There were five false positives, all in mouse. All were similar in sequence to one of two known selenoproteins in the same species, either SelK or GPx4, but they all were intronless and with no evidence of transcription. These are recently retrotransposed pseudogenes, so similar to real selenoproteins that it is actually desirable that our method finds them. There were 19 false negatives, caused by a variety of reasons. For example, *Drosophila*

SelK was missed because all other SelK proteins annotated in nr were too distant to give good Blastx alignments. This small selenoprotein is known to show poor homology even among closely related organisms. *Drosophila* SPS2 was processed as a candidate, but it was discarded during filtering owing to the presence of in-frame stop codons. These in reality reside in an intron of the gene, but they were included in the coding sequence

The best scoring candidate was found in the choanoflagellate *Monosiga brevicollis* and showed homology to AhpC. This is a thioredoxin-like protein family (like many known selenoproteins), and its distant homolog was previously detected as a selenoprotein in Bacteria. Recently, an AhpC-like selenoprotein was also predicted in some sponges (17), but it was thought to be limited to this lineage. Using Selenoprofiles, we built a profile alignment with the AhpC selenoproteins in Bacteria, choanoflagellates and Porifera, including also a number of metazoan cysteine homologues. We used our new profile to scan a collection of eukaryotic and prokaryotic genomes and detected AhpC selenoproteins in a wide range of lineages, including protists and basal metazoans. In Figure 4, we present an alignment of the Sec-containing domain of AhpC selenoproteins, along with some



metazoan cysteine homologues. Among the Selenoprofiles AhpC predictions, we also found our second best scoring Sebastian candidate, in the *Emiliana huxleyi* genome.

Given the conservation of these genes, the thioredoxin fold, the cysteine homology and the presence of SECIS elements in most of eukaryotic candidates, the finding leaves no doubt that this is a true selenoprotein. For details and data on the analysis on AhpC, see Supplementary Material S6. It may seem controversial that our best new selenoprotein candidate was already described in literature as a eukaryotic selenoprotein, in Porifera. However, this eukaryotic selenoprotein family was novel to Sebastian, as no Porifera AhpC selenoprotein was yet annotated in the nr database. Bacterial homologues were annotated, but their phylogenetic distance exceeds the detection power of our method. The example of AhpC supports the quality of Sebastian predictions. Further use of this tool should be instrumental in finding new selenoproteins, both in our current ranked set and in future runs, as more and more species are sequenced.

A webserver for SECISearch3 and Sebastian

We created a web server to allow users world-wide to upload any nucleotide sequence and run SECISearch3 and/or Sebastian (Figure 5). It is hosted both at <http://gladyshvlab.org/SelenoproteinPredictionServer> and at <http://sebastian.crg.es>. The user can choose to run Sebastian or just SECISearch3 and can also control the main options of the programs. For example, the SECIS prediction methods can be chosen and their stringency can

be set, the SECIS filter can be toggled and so forth. An important option for Sebastian is whether the search is done for known selenoproteins or new ones. In the first case, Blastx is run only against a reduced version of nr containing only selenoproteins, which greatly reduces the computational time. Once ready, results can be inspected directly on a web page or downloaded as fasta or gff files. Until today, selenoprotein prediction was a task typically carried out by only a few experts in the field. This web server allows for the first time any user, even with little expertise in bioinformatics, to perform reliable selenoprotein predictions on any nucleotide sequence of interest.

CONCLUSION

We describe two new computational methods for selenoprotein prediction and analysis: SECISearch3 and Sebastian. The former is a major improvement of SECISearch and is currently the best method to predict eukaryotic SECIS elements. The latter is a new method to predict selenoproteins in nucleotide sequences, which is built based on SECIS prediction. Sebastian is able to predict known selenoproteins as well as new selenoprotein homologues of known proteins, provided that they have at least one cysteine homologue. We ran Sebastian on the available protist genomes, where we expect a number of selenoproteins to be still undiscovered, and we provided a list of ranked selenoprotein candidates. An analysis of a representative candidate selenoprotein AhpC is used to illustrate the predictions and evolution of new selenoprotein families. Both SECISearch3 and Sebastian

Selenoprotein prediction server

Welcome to the SECISearch3/Sebastian server. Mouse over the different fields to display information about them in this box.

☐ SECIS prediction
SECISearch3

☒ search also complementary strand
☒ filter improbable structures
☒ generate SECIS images (dpi: 150)

SECISearch3 method:

☒ Infernal
score threshold: 10
☐ Covels

☐ Original SECISearch

Upload your sequence file:
Browse...
or paste it here:

Submit

Selenoprotein id: 1 **Category: known selenoprotein**

Protein prediction
Predicted by: exonerate **Blastx value: 2e-66**
Query protein: gil61230152|gblAAX40994.1| glutathione peroxidase 2 [synthetic construct]
Positions on query: 1-190 Query length: 191

Target name: SPT00000005_1.0
Positions on target: 3352-3573,338-685 **Strand: -**

Query MAFIAKSPYDLISAISLDGEIVDFNTFGRVLIHVASLGTTTDFDTQLNELQCFRRLVVLGPPCNO
|||||< 266bp > |||||
Target MAFIAKSPYDLISAISLDGEIVDFNTFGRVLIHVASLGTTTDFDTQLNELQCFRRLVVLGPPCNO
agttagatttgcagacggaggttaacgagggcagaggtctgaacgtaccagcctcaccggcgtctac
ttcttaactaatgctgagaataactggcttaactcctggccgcatcataaataggctgtttgtcgaa
gtctgcctctcccggtggagctgcgcggcgtgtgtgcacacgcggcgagcctgcggcctccca

Query FGHQ <---Intron---> ENCQNEIILNSLYVRPGGYOPTTLVQCEVINGQNEHPVFAILEDKLP
|||||< 266bp > |||||
Target FGHQ ENCQNEIILNSLYVRPGGYOPTTLVQCEVINGQNEHPVFAILEDKLP
tgcc gatcaggacaacatgccgggtccatagcagcgagcgcgctgcagacc
agaaaaatagttaagtcgggaacctcttaagatagaaaaactcataaatc
gctgtggcgtcgtctgttgacgccctcaatggcgtgtcccgccgc

Query YPYDDPFSIMTDPPELLINSVPRSDVAMNFEFLIGPEGPFRRYSRTFTPTINIEPDKRLKVAI
|||||
Target YPYDDPFSIMTDPPELLINSVPRSDVAMNFEFLIGPEGPFRRYSRTFTPTINIEPDKRLKVAI
cttggttcaagacaataacgcctgggtatgatcagcgggtctctacataaaagcgaaccaggga
acaacactctcaatttggctgcgatcgaatttgagagctggaggtctctacacatagttatct
ctttcatcgcctgcgtgctgcacgcgtggcaggaggaacacccacacctgtcgcgcatca

SECIS prediction
Predicted by: Infernal (score 35.57)
Covels score: 29.54
Free Energy of structure: -23.3
Target name: SPT00000005_1.0
Positions on target: 52-116 Strand: -
Distance between candidate CDS and SECIS: 221
Distance between Sec-UGA and SECIS: 671
SECIS grade: A

SEQ GSCCUUCACAGAAUGAUGGCAACUCCUAAACCCUACUGGUGUGUCUGAGAGGCGAAGGCG
SS ..(((((((.....((((((((((((((.....)))))))))))))111).....))))))..

Figure 5. Two snapshots of the SECISearch3/Sebastian web server. On the left, the input form. On the right, the output page displayed when submitting the human GPx2 sequence.

are public and can be run on a dedicated web server at <http://gladyshevlab.org/SelenoproteinPredictionServer> or <http://sebastian.crg.es>.

SUPPLEMENTARY DATA

Supplementary Data are available at NAR Online: Supplementary Materials 1–6.

ACKNOWLEDGEMENTS

The authors thank Sean Eddy for his assistance over building an Infernal model for eukaryotic SECIS.

FUNDING

Funding for open access charge: NIH [GM061603].

Conflict of interest statement. None declared.

REFERENCES

- Hoffmann, P.R. and Berry, M.J. (2005) Selenoprotein synthesis: a unique translational mechanism used by a diverse family of proteins. *Thyroid*, **15**, 769–775.
- Allmang, C., Wurth, L. and Krol, A. (2009) The selenium to selenoprotein pathway in eukaryotes: more molecular partners than anticipated. *Biochim. Biophys. Acta*, **1790**, 1415–1423.
- Hatfield, D., Carlson, B., Xu, X., Mix, H. and Gladyshev, V. (2006) Selenocysteine incorporation machinery and the role of selenoproteins in development and health. *Prog. Nucleic Acid Res. Mol. Biol.*, **81**, 97–142.
- Squires, J. and Berry, M. (2008) Eukaryotic selenoprotein synthesis: mechanistic insight incorporating new factors and new functions for old factors. *IUBMB Life*, **60**, 232–235.
- Driscoll, D.M. and Chavatte, L. (2004) Finding needles in a haystack. In *silico* identification of eukaryotic selenoprotein genes. *EMBO Rep.*, **5**, 140–141.
- Kryukov, G.V., Kryukov, V.M. and Gladyshev, V.N. (1999) New mammalian selenocysteine-containing proteins identified with an algorithm that searches for selenocysteine insertion sequence elements. *J. Biol. Chem.*, **274**, 33888–33897.
- Lescure, A., Gautheret, D., Carbon, P. and Krol, A. (1999) Novel selenoproteins identified in *silico* and in *Vivo* by using a conserved RNA structural motif. *J. Biol. Chem.*, **274**, 38147.
- Castellano, S., Morozova, N., Morey, M., Berry, M., Serras, F., Corominas, M. and Guigó, R. (2001) In *silico* identification of novel selenoproteins in the *Drosophila melanogaster* genome. *EMBO Rep.*, **2**, 697.
- Kryukov, G., Castellano, S., Novoselov, S., Lobanov, A., Zehab, O., Guigo, R. and Gladyshev, V. (2003) Characterization of mammalian selenoproteomes. *Science*, **300**, 1439.
- Kryukov, G. and Gladyshev, V. (2004) The prokaryotic selenoproteome. *EMBO Rep.*, **5**, 538.
- Taskov, K., Chapple, C., Kryukov, G.V., Castellano, S., Lobanov, A.V., Korotkov, K.V., Guigó, R. and Gladyshev, V.N. (2005) Nematode selenoproteome: the use of the selenocysteine insertion system to decode one codon in an animal genome? *Nucleic Acids Res.*, **33**, 2227–2238.
- Li, M., Huang, Y. and Xiao, Y. (2009) A method for identification of selenoprotein genes in archaeal genomes. *Genomics Proteomics Bioinformatics*, **7**, 62–70.
- Chapple, C.E., Guigó, R. and Krol, A. (2009) SECISaln, a web-based tool for the creation of structure-based alignments of eukaryotic SECIS elements. *Bioinformatics*, **25**, 674–675.
- Jiang, L., Liu, Q. and Ni, J. (2010) In *silico* identification of the sea squirt selenoproteome. *BMC Genomics*, **11**, 289.
- Mariotti, M. and Guigó, R. (2010) Selenoprofiles: profile-based scanning of eukaryotic genome sequences for selenoprotein genes. *Bioinformatics*, **26**, 2656–2663.
- Gobler, C.J., Berry, D.L., Dyhrman, S.T., Wilhelm, S.W., Salamov, A., Lobanov, A.V., Zhang, Y., Collier, J.L., Wurch, L.L., Kustka, A.B. *et al.* (2011) Niche of harmful alga *Aureococcus anophagefferens* revealed through ecogenomics. *Proc. Natl Acad. Sci. USA*, **108**, 4352–4357.
- Jiang, L., Ni, J. and Liu, Q. (2012) Evolution of selenoproteins in the metazoan. *BMC Genomics*, **13**, 446.
- Krol, A. (2002) Evolutionarily different RNA motifs and RNA-protein complexes to achieve selenoprotein synthesis. *Biochimie*, **84**, 765–774.
- Grundner-Culemann, E., Martin, G.W., Harney, J.W. and Berry, M.J. (1999) Two distinct SECIS structures capable of directing selenocysteine incorporation in eukaryotes. *RNA*, **5**, 625–635.
- Martin, G.W., Harney, J.W. and Berry, M.J. (1996) Selenocysteine incorporation in eukaryotes: insights into mechanism and efficiency from sequence, structure, and spacing proximity studies of the type 1 deiodinase SECIS element. *RNA*, **2**, 171–182.
- Hofacker, I., Fontana, W., Stadler, P., Bonhoeffer, S., Tacker, M. and Schuster, P. (1994) Fast folding and comparison of RNA secondary structures. *Monatshfte für Chemie*, **125**, 167–188.
- Hofacker, I.L. (2003) Vienna RNA secondary structure server. *Nucleic Acids Res.*, **31**, 3429–3431.
- Nawrocki, E.P., Kolbe, D.L. and Eddy, S.R. (2009) Infernal 1.0: inference of RNA alignments. *Bioinformatics*, **25**, 1335–1337.
- Griffiths-Jones, S. (2005) RALEE-RNA ALIGNMENT editor in Emacs. *Bioinformatics*, **21**, 257–259.
- Gautheret, D. and Lambert, A. (2001) Direct RNA motif definition and identification from multiple sequence alignments using secondary structure profiles. *J. Mol. Biol.*, **313**, 1003–1011.
- Novoselov, S., Rao, M., Onoshko, N., Zhi, H., Kryukov, G., Xiang, Y., Weeks, D., Hatfield, D. and Gladyshev, V. (2002) Selenoproteins and selenocysteine insertion system in the model plant cell system, *Chlamydomonas reinhardtii*. *EMBO J.*, **21**, 3681.
- Novoselov, S., Lobanov, A., Hua, D., Kasaikina, M., Hatfield, D. and Gladyshev, V. (2007) A highly efficient form of the selenocysteine insertion sequence element in protozoan parasites and its use in mammalian cells. *Proc. Natl Acad. Sci. USA*, **104**, 7857–7862.
- Lobanov, A., Delgado, C., Rahlfs, S., Novoselov, S., Kryukov, G., Gromer, S., Hatfield, D., Becker, K. and Gladyshev, V. (2006) The plasmodium selenoproteome. *Nucleic Acids Res.*, **34**, 496.
- Altschul, S., Madden, T., Schaffer, A., Zhang, J., Zhang, Z., Miller, W. and Lipman, D. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.*, **25**, 3389.
- Fomenko, D.E., Xing, W., Adair, B.M., Thomas, D.J. and Gladyshev, V.N. (2007) High-throughput identification of catalytic redox-active cysteine residues. *Science*, **315**, 387–389.
- Fomenko, D.E. and Gladyshev, V.N. (2012) Comparative genomics of thiol oxidoreductases reveals widespread and essential functions of thiol-based redox control of cellular processes. *Antioxidants Redox Signal.*, **16**, 193–201.
- Slater, G.S. and Birney, E. (2005) Automated generation of heuristics for biological sequence comparison. *BMC Bioinformatics*, **6**, 31.
- Gobler, C.J., Lobanov, A.V., Tang, Y.Z., Turanov, A.A., Zhang, Y., Doblin, M., Taylor, G.T., SañudoWilhelmy, S.A., Grigoriev, I.V. and Gladyshev, V.N. (2013) The central role of selenium in the biochemistry and ecology of the harmful pelagophyte, *Aureococcus anophagefferens*. *ISME J.*, **7**, 1333–1343.

Supplementary Material S2: Building an infernal model for eukaryotic SECIS

The structural partition by SECISearch1 (coming directly from the PatScan patterns) was used to align them: the subsequences belonging to each partition were aligned independently, and all resulting alignments were then concatenated. The result is a rough, structure-based alignment. A secondary structure was assigned to each column of this alignment by computing a consensus of the secondary structures output by SECISearch1 (RNAfold) for each SECIS, and then refining manually the consensus.

[illegible]

This is a small sample of the actual alignment, which is too large to be visualized here. RALEE highlights the nucleotides that are paired according to the consensus secondary structure below, and also respect the standard pairing rules. As you can see, the vast majority of sequences are not aligned well. We inspected manually this alignment, to identify and extract a subalignment of well alignment sequences. This resulted in a small bona-fide alignment

displayed below. This alignment was used to build a “seed” Infernal model using the program cmbuild.

[illegible]

This seed alignment was then expanded by aligning more potential SECISes from the initial set, in an iterative procedure as suggested in the Infernal manual. The set of Selenoprofiles-predicted three prime UTRs of selenoproteins was then scanned again with the new model, and we replaced some SECISearch1 false positives in our set, poorly aligned, with a new, more convincing Infernal prediction in the same UTR. At last, we added manually inspected SECIS from certain poorly represented lineages, such as basal eukaryotes. During this procedure, we ran jackknife tests (data not shown) to assess the quality of the model, to test different score thresholds when adding sequences to the growing seed model, as well as testing some parameters of cmbuild, and the effect of a sequence-identity based redundancy filter, which we implemented to avoid overfitting of the model. Our tests showed that the filter did not improve the model, indicating that the Infernal models already takes well into account the potential overfitting. Our final secondary structure alignment contained 1122 SECIS sequences. Find below a RALEE snapshot. Again, a small subseq of sequences were randomly

[illegible]

Supplementary Material S4: Python procedures for filtering and scoring

We report here the actual python code used for filtering SECISearch3 and Sebastian predictions, and for scoring the SECIS elements. It contains useful parameters, such as the accepted stem length and the free energy threshold. The comments in green should allow even those non initiated to programming to follow.

1. SECISearch3 filtering

The python procedure used internally by SECISearch3 to filter unlikely structures is shown below. The procedure consists in a series of checks on specific characteristics of the “self” object – the SECIS prediction being evaluated. If a check fails, a `False` value is returned (thus the predicted is filtered out) also with a reason why, here colored in red. See the comments indicating what characteristic is checked, here colored in green. If the prediction passes all checkes, it arrives to the last line where a `True` value is returned.

```
def secis_filter(self):
    #checking that core is aligned
    if not self.core or len(self.core[0])<2:                return False, "no core aligned"

    #checking that core contains invariant GA-GA
    core_seq_5 = join( [self.sequence()[ i-1 ] for i in self.core[0] ] , '' )
    core_seq_3 = join( [self.sequence()[ j-1 ] for j in self.core[1] ] , '' )
    if not 'GA' in core_seq_5 or not 'GA' in core_seq_3:    return False, "no GA-GA in core"

    a_length= self.apical_loop_length() #checking apical loop length
    ### IMPORTANT: the apical loop is here defined as anything included between the two
    aligned parts of stem2, so it includes the length of stem3 when present; this is why the
    upper boundary is so high
    if a_length<8:                return False, "apical loop too short"
    if a_length>30:               return False, "apical loop too long "

    s2_length=self.stem2_length() #checking stem2 length
    if s2_length<7:               return False, "stem2 too short"
    if s2_length>15:              return False, "stem2 too long"

    ins_s2=self.insertion_stem2() #checking stem2 bending (n of insertions 1 side vs other)
    if max(ins_s2)-min(ins_s2) > 2:    return False, "too much bending"

    #checking energy previously computed by RNAeval; ### opt['secis_energy'] = -4
    if self.energy > opt['secis_energy']:    return False, "free energy too high"
    return True, "ok"
```

2. SECIS grading

The python procedure used for grading SECIS (labelling them with a qualitative score) is reported here. This procedure was heuristically tuned to give the same results of the eye of a bioinformatician expert in selenoproteins. A custom score is computed throughout the procedure, taking into account various features (read the comments in green). At the end, this score is transformed into a grade: A or B or C.

```
def grade(self):
    score=0.0
    ##### FAVORING CERTAIN FEATURES: increasing score
    # checking the difference in number of insertions on the two sides of stem2 (bending
    # coefficient). Scoring positively when bending coefficient < 2
    i=self.insertion_stem2()
    if max(i)-min(i)<=1: score+=1

    # checking stem2 for bad pairs (not allowed in canonical or wooble watson-crick
    # pairing rules). Scoring positively when there are less than 3
    pairs_stem2=self.stem2_pairs()
    if len([1 for obj in pairs_stem2 if obj[2]=='o']) < 3:        score+=1

    # checking the conserved unpaired nucleotides at the apical loop. Scoring positively
    # those with AA, and, although with lower score, those with CC
    if self.unpaired_nts:
        nts=join([ self.sequence()[p-1] for p in self.unpaired_nts ], '')
        if 'AA' in nts:        score+=1        # AA or AAA in unpaired nts
        elif 'CC' in nts:        score+=0.4        # CC in unpaired nts

    # scoring positively those with a good covels score (this is always computed or each
    # candidate, even if cove was not run on the whole target)
    if self.get_cove_score() >= 15:        score+=1

    ##### PENALIZING OTHER FEATURES: decreasing score
    # checking here the positions just before and after the core.
    # Penalizing if a mismatch, or bad pair, or insertion, is present just after it.
    # Penalizing if a match is present just before it
    if self.core:
        last_core_pos_x=max(self.core[0]);    last_core_pos_y=min(self.core[1])
        after_pos_x_is_paired=False; after_pos_y_is_paired=False;
        for x, y, category in pairs_stem2:
            if x == last_core_pos_x+1:
                after_pos_x_is_paired=True
                if y == last_core_pos_y-1: after_pos_y_is_paired=True
                if category=='o':        score-=0.6 # bad pair after core
                break
            if not after_pos_x_is_paired or not after_pos_y_is_paired:
                #insertion after core on either sides
                score-=0.6
        first_core_pos_x=min(self.core[0]);    first_core_pos_y=max(self.core[1]);
        if first_core_pos_x>1 and len(self.sequence())>first_core_pos_y:
            nt_precore_x= self.sequence()[first_core_pos_x-2]
            nt_precore_y= self.sequence()[first_core_pos_y]
            try: category=category_per_pair.get(nt_precore_x, nt_precore_y)
            except KeyError: category='o'
            if category in 'acw':        score-=0.4 #match just before core

    # Penalizing if consecutive bad pairs (mismatches) are predicted in stem2
    last_pair_category=''
    for x, y, category in pairs_stem2:
        if last_pair_category and last_pair_category=='o' and category=='o':    score-=1
        last_pair_category=category

    ##### TRANSFORMING score into a category A, B or C. Category A is allowed only for
    # predictions with a good covels score
    if score<1.5:    out='C'
    elif score<2.5:    out='B'
    elif score>=2.5 and self.get_cove_score() >= 15:        out='A'
    else: out='B'
    return out
```


3. Sebastian filtering

The python procedure used for filtering Sebastian candidates is reported below. Like in the SECISearch3 filtering procedure, a series of checks is performed on the candidate (called obj in the code); if False is returned, the prediction is discarded. Mostly, the filter checks the distance of the predicted coding sequence with its SECIS, and the conservation at the two sides of the predicted selenocysteine.

```
def default_filter(obj, min_conserved_per_side=3):
    # checking the distance of the SECIS from the coding sequence
    ### opt['max_secis_distance'] = 3000 (nt)
    if obj.distance_from_secis()>opt['max_secis_distance']: return False

    # the next line deals with very specific cases: those in which the original blast
    # alignment features an alignment between a X in the nr annotated protein and a UGA.
    # Those were kept as sometimes selenocysteine are annotated as X). We filter out here
    # those cases if the nr annotated protein has lots of X. This part of the filter was
    # implemented specifically to filter out spurious hits coming from a set of plant
    # proteins annotated in NR with lots of X in their sequence
    if obj.category=='ALI_X' and obj.query_full_seq.count('X')>1: return False

    #keeping all cases in which more than a selenocysteine is predicted in the candidate
    if obj.protein().count('U')>1: return True

    ##### CONSERVATION FILTER #####
    # this is the most important part of the filter. The gene prediction is an alignment
    # between a nr annotated protein (query) and a translated genomic region (target).
    # In this procedure, we parse all alignment positions at the left (upstream) and at the
    # right of the predicted selenocysteine, and we count the number of conserved position.
    # we call conserved an alignment between two identical or similar aminoacids (those
    # with a positive score in blosum62). The filter requires at least
    # min_conserved_per_side = 3 residues for each side. If the conservation at the right
    # side is not there, a prediction could still pass the filter, but only if the
    # selenocysteine is the last or penultimate residue of the prediction, and there is a
    # non-UGA stop codon just downstream of the predicted coding sequence. This allows TR
    # and TRlike protein predictions to pass the filter

    pos_u_in_ali=obj.alignment.seq_of('t').index('U')
    conserved_left=0; conserved_right=0
    for i in range(pos_u_in_ali):
        aa_query= obj.alignment.seq_of('q')[i]; aa_target= obj.alignment.seq_of('t')[i]
        if not '-' in aa_query+aa_target and ( aa_query==aa_target or similar_aas(aa_query,
aa_target) ): conserved_left+=1
    for i in range(pos_u_in_ali+1, obj.alignment.length()):
        aa_query= obj.alignment.seq_of('q')[i]; aa_target= obj.alignment.seq_of('t')[i]
        if not '-' in aa_query+aa_target and ( aa_query==aa_target or similar_aas(aa_query,
aa_target) ): conserved_right+=1
    if conserved_left< min_conserved_per_side: return False
    if conserved_right >= min_conserved_per_side: return True
    if obj.query_prot_length - obj.query.boundaries()[1] <2 :
        try:
            codon_downstream= upper(obj.downstream(0, 3).fasta_sequence()[1])
            if codon_downstream in ['TAG', 'TAA']: return True
        except: pass
    return False
```

Supplementary Material S6: Analysis of the AhpC selenoprotein family

This document contains a summary of the analysis of the top Sebastian candidate, belonging to the AhpC selenoprotein family. Note that this supplementary material section includes also two alignment files.

Building a AhpC profile alignment

Our top scoring new selenoprotein candidate by Sebastian was in choanoflagellate *Monosiga brevicollis* (*Monosiga_brevicollis*.SeB.10).

Using this sequence with blastp against the nr database, we identified several homologues across eukaryotes, including protein AhpC/TSA antioxidant enzyme in *Homo sapiens*.

AhpC was already described as selenoprotein in some bacteria, and very recently in some sponges (Jiang et al, BMC genomics, 2012).

The top blastp hit was a hypothetical protein in choanoflagellate *Salpingoeca* sp. ATCC 50818 (gi|326427370|gb|EGD72940.1). This alignment showed a very suspicious gap in the *Salpingoeca* sequence corresponding to the predicted Sec region in *Monosiga*. Scanning the genome of *Salpingoeca* sp. ATCC 50818, we could identify the Sec region as conserved, and we could also detect a SECIS downstream.

We start collecting AhpC sequences in order to build a profile alignment to be used with Selenoprofiles, a pipeline for profile-based gene prediction pipeline able to correctly predict selenoproteins (Mariotti and Guigo, Bioinformatics, 2010). We included the *Monosiga* sequence, the corrected *Salpingoeca* sequence, and their most similar proteins annotated in nr (best blastp hits).

Additionally, we used the sequences shown in Figure 1 of the paper Jiang et al, BMC genomics, 2012. This included the 5 AhpC-like selenoproteins from sponges *Amphimedon queenslandica* and *Oscarella carmela* (we excluded a partial sequence from *Suberites domuncula*), and a set of bacterial Sec containing AhpC identified by blastp with the *Amphimedon* sequence (various *Geobacter* species, *Dehalogenimonas lykanthroporepellens*, plus some other sequences not reported in that figure, such as one from *Desulfovibrio salexigens*).

The resulting profile alignment is included in this suppl. material: see file *AhpC_profile_alignment.fa* (online).

Searching AhpC across genomes

Then, we used Selenoprofiles version 3.0 (download at big.crg.cat/services/selenoprofiles) (Mariotti and Guigó, 2010).

We ran the AhpC profile with quite strict filtering (`aws_i_z_score > -2` -- see selenoprofiles manual) on a large collection of genomes, eukaryotic and prokaryotic.

At a manual inspection, results looked very good: despite the high sequence diversity of the input alignment, all predictions appeared to "fit" into the profile.

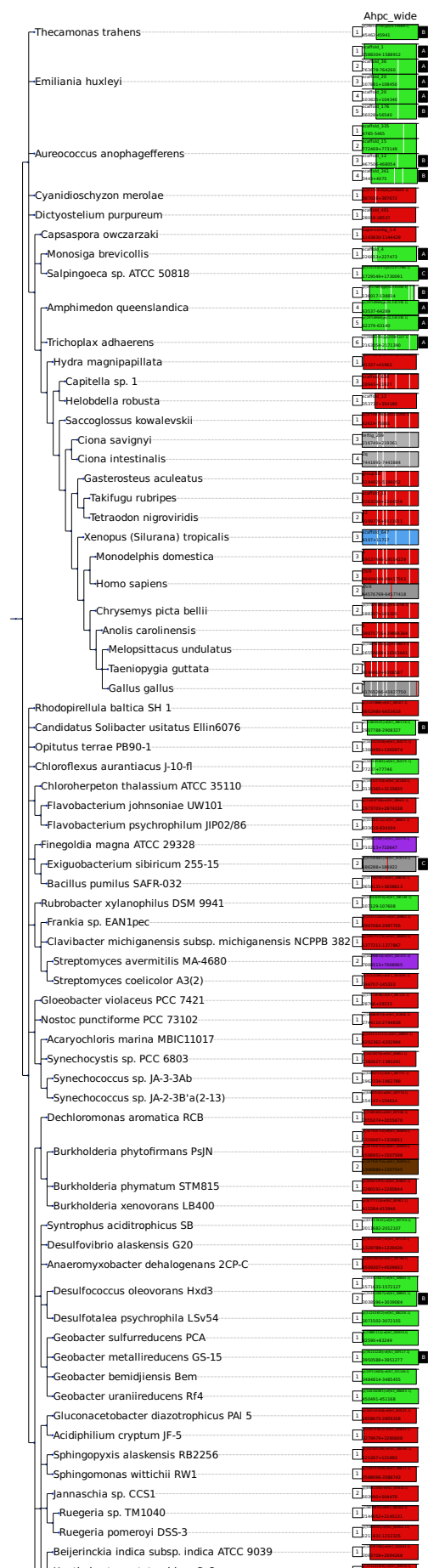
A total of 108 genes were predicted: 28 with selenocysteine, 72 with cysteine, 8 with something else aligned to Sec position.

In the file `AhpC_results.aligned_with_profile.fa`, you have all these results aligned with the profile sequences on top. In each fasta header, the target species and the genomic coordinates of the predictions can be found.

An overview of the results can be seen in the figure `AhpC_results.tree_drawer.pdf`, produced by program `selenoprofiles_tree_drawer` (next page, or available online). On the left, a phylogenetic tree of all species with at least a AhpC prediction is shown. The tree is derived from the ncbi taxonomy tree, which does not resolve well certain nodes. Eukaryotes are on the top, bacteria on the bottom. On the right, there is a colored box for each AhpC gene found in that species. The color of the box depends on the amino acid found in the Sec position: green means selenocysteine, red means cysteine, dark grey means some pseudogene features are found (either frameshifts or inframe stop codons); the other colors are for the rare cases, e.g. purple for threonine, brown for serine. The width and position of each colored box indicate the coverage of the prediction in respect to the profile, meaning which portion of the profile is aligned with this gene prediction. Inside the box, vertical white lines indicate the position of introns, projected against the protein alignment with the profile. Inside the boxes, the scaffold name and the positions are shown. On the left side of the colored box, the selenoprofiles prediction id is shown, which allows to identify univocally any sequence in the alignment of results. On the right side, there is a black box for each gene for which SECISearch3 detected a SECIS downstream (maximum distance with CDS: 5kb); inside the box, the grade for the SECIS is reported.

As you can see, there are predictions across a wide range of eukaryotic and prokaryotic lineages. We found at least one Sec-containing AhpC in *Thecamonas trahens*, *Emiliana huxleyi*, *Aureococcus anophagefferens*, *Tricoplax adhaerens*, and in the already described lineages of sponges and choanoflagellates. All these predictions except two have a SECIS downstream.

We found Sec-AhpC also in 10 bacterial species, mostly from Deltaproteobacteria. Some have a false (?) SECISearch3 prediction.



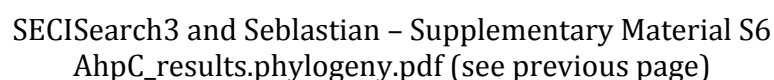
Note: this figure was reduced to allow visualization in this document, leaving out some prokaryotic cysteine forms. Refer to online supplementary material for the full figure.

Phylogenetic analysis of results

Inspecting by eye the results alignment, we noticed that the sequences did not appear to follow the phylogenetic relationships of the investigated species. Thus, we ran a phylogenetic reconstruction pipeline (described in the methods of paper Mariotti et al, PLoS One, 2012) on all AhpC results mentioned above. You have an overview of results in AhpC_results.phylogeny.pdf (next page) . On the left, the predicted phylogeny of proteins and their distance is displayed. Each leaf of the tree represent a protein, as a ball colored with the same color schema as indicated above, followed by its selenoprofiles numerical id, and its species and taxonomy.

As anticipated, there are inconsistencies with species phylogeny. The two Sec-AhpC in *Desulfococcus oleovorans* Hxd3 cluster with eukaryotic sequences rather than bacterial. Choanoflagellate Sec-AhpC sequences are embedded within deuterostome sequences rather than at its root. Then, most strikingly, *Amphimedon* Sec-AhpC (sponges) cluster with bacteria.

Thus, although the predicted AhpC selenoprotein genes are arguably real selenoproteins, their real phylogeny is difficult to resolve. Although this is only at the level of speculation, we think that the current data suggest that horizontal transfer occurred at least twice, giving origin to a bacterial-like AhpC in basal eukaryotes, and then independently in Porifera.



Chapter 3

RESULTS

3.1 Consortium projects

This section covers a few international projects dedicated to sequencing/annotating the genome of organisms of interest. This type of collaborative project is generally undertaken by researches from diverse nations and groups, united in a “consortium”. Typically researchers interact only (or mostly) through the internet (conference calls, emails). In the cases I present, my contribution consisted in the accurate annotation of the selenoproteins in the genome in question.

Publications:

ENCODE Project Consortium.

A user’s guide to the encyclopedia of DNA elements (ENCODE). PLoS Biol. 2011 Apr;9(4):e1001046. doi: 10.1371/journal.pbio.1001046.

ENCODE Project Consortium.

An integrated encyclopedia of DNA elements in the human genome. Nature. 2012 Sep 6;489(7414):57-74. doi: 10.1038/nature11247.

International Aphid Genomics Consortium.

Genome sequence of the pea aphid *Acyrthosiphon pisum*. PLoS Biol. 2010 Feb 23;8(2):e1000313. doi: 10.1371/journal.pbio.1000313.

3.1.1 Selenoproteins in the gencode reference annotation

The Encyclopedia of DNA elements (ENCODE) is an ambitious project launched in 2003 by the US National Human Genome Research Institute. Its goal was to find and characterize all functional elements in the human genome (transcripts/genes, regulatory regions). Its pilot phase focused only on 1% of the genome, to develop and evaluate different experimental and computational techniques. From 2007 ENCODE entered its production phase, and finally released results in a set of 30 papers published simultaneously in 2012 (see <http://www.nature.com/encode/>). A set of wisely selected human cell lines were subject to many and diverse experimental procedures. New sequencing technologies were massively applied using different sample preparations (e.g. enriching in short or long RNA, with or without poly-A tail), to have the most accurate description of the human transcriptional landscape. Many other genomic features were abundantly sampled, as for example the DNase 1 hypersensitive sites, and the binding sites of a number of transcription factors, through ChIP-Seq. One of the most striking results of the project was an unexpected high proportion of genome being transcribed, and the identification of novel classes of non-coding RNA.

ENCODE required the effort of hundreds of scientists world-wide, and it is very likely the biggest collaborative project ever undertook in biology. Research groups from all over the world participated for different tasks, from experimental data collection to storing, distribution and analysis. ENCODE was used to improve the annotation of the human genome: algorithms were designed to exploit the abundant available data and infer the presence and genomic coordinates of genes. Ultimately, a reference annotation called gencode was produced, and then updated as techniques and data evolved. A large portion of gencode comes from the work of annotators, manually inspecting genomic regions to detect and correct mistakes, or add new genes. My contribution was mainly as selenoprotein annotator: I checked that all genes had their selenocysteines right, correcting the mistakes found. Because the automated pipelines for gene annotation changed with the early gencode versions, I had to repeat this process a few times, and I observed how sometimes a correctly annotated gene disappeared in a newer version. Gencode is today at its version 18: <http://www.encodegenes.org/>. This work was worth my inclusion as author in two papers by the ENCODE consortium [ENCODE-Consortium, 2011, 2012]. For their scarce relevance with the rest of the thesis, we include here only their abstract (figures 3.1 and 3.2).

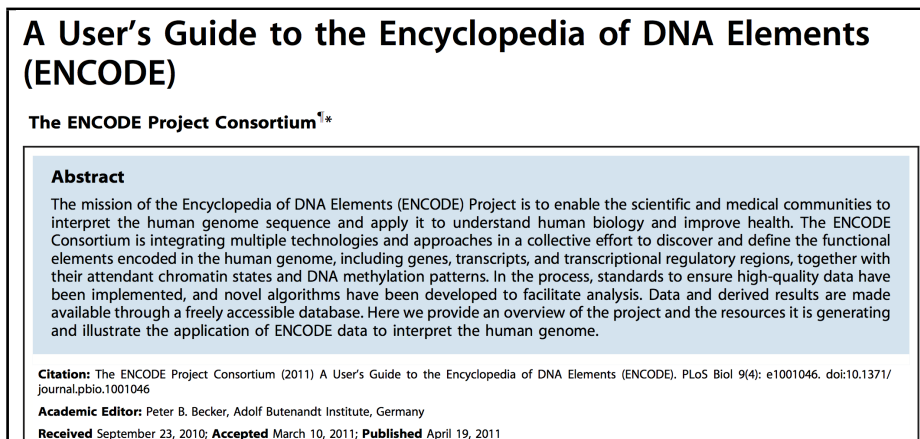


Figure 3.1: Snapshot of ENCODE paper [ENCODE-Consortium, 2011].

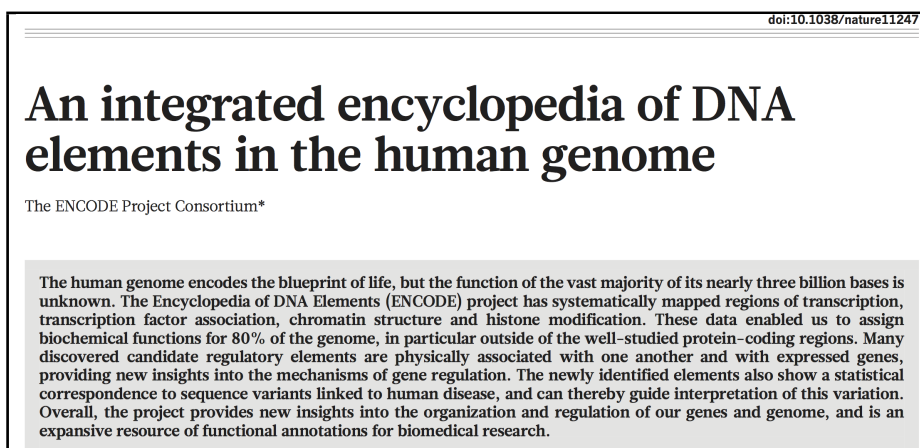


Figure 3.2: Snapshot of ENCODE paper [ENCODE-Consortium, 2012].

3.1.2 A novel selenoprotein extinction in the genome of pea aphid

After selenoproteinless insects were discovered [Drosophila-Consortium, 2007; Chapple and Guigó, 2008], we were eager to analyze novel insect sequences to complete the picture. Our involvement in the pea aphid genome project [Aphid-Consortium, 2010] (see abstract in figure 3.3) gave us the opportunity to have a look at this peculiar species before anyone else. Aphids are a class of sap-sucking insects, considered among the most destructive pests on cultivated plants. Pea aphid (*Acyrtosiphon pisum*) is its most studied species. This organism has an amazingly complex life cycle, including both sexual and asexual forms, both winged and unwinged. Aphids belong to the order Hemiptera, superorder Para-neoptera, whose species undergo partial metamorphosis (hemimetabola). Phylogenetically, they are placed basal to most well studied insects, including Diptera (mosquitoes, flies), Lepidoptera (butterflies, moths), Coleoptera (beetles) and Hymenoptera (ants, wasps and bees), which all together form the superorder of Endopterygota, the holometabolic insects. In [Chapple and Guigó, 2008] (see figure 1.11), some selenoproteins were identified in EST sequences of the paraneopteran *Pediculus humanus* (louse) and *Homalodisca coagulata* (a leafhopper also known as *H. vitripennis*). To our surprise, we found none in the pea aphid genome. No cysteine homologues for the drosophila selenoproteins were found neither. Consistently, many Sec machinery genes were also absent: tRNA^{sec}, SBP2, eEF^{sec}, secp43, pstk, SPS2. The genes for SecS and SPS1 were present, and they are probably carrying out functions unrelated to selenocysteine. SPS1 is retained in all other known selenoproteinless insects too [Chapple and Guigó, 2008; Lobanov et al., 2008].

The analysis of the pea aphid genome lead to the discovery of a novel selenoprotein extinction, which, to date, includes only this species. This was the first Sec extinction documented outside Endopterygota, and reinforced our idea that important steps to lose selenoproteins had been already completed at the root of insects.

3.1.3 Centipede genome annotation

Our involvement in the *Strigamia maritima* genome project also stems from our interest in insect selenoproteins. This centipede (Myriapoda) is in fact a non-insect arthropod, a useful outgroup to study the massive selenoproteome reduction at the root of insects. At the 4th Annual Arthropod Genomics Symposium, held in Kansas City (USA) in 2010, Michael Akam from the Darwin College in Cambridge presented his project of sequencing the centipede genome. In that occasion, we asked him to join the consortium to characterize its selenoproteome.

The annotation of this genome, coordinated by Stephen Richards, consisted in the application of a variety of computational methods, which can be divided in three classes: 1. protein-to-genome aligners (e.g. tblastn, exonerate, genewise) search matches of protein queries in the genome translated in all possible frames; 2. RNA-to-genome mappers (e.g. tophat, gem) align RNAseq data to find their

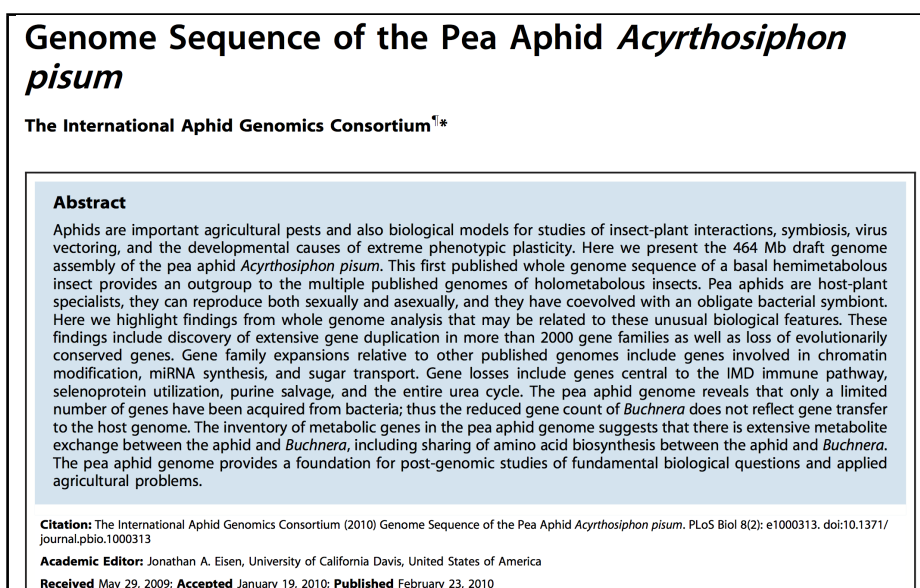


Figure 3.3: Snapshot of the paper by the International Aphid Genomics Consortium [Aphid-Consortium, 2010].

original location in the genome, optionally assembling them first in transcripts; 3. *de novo* predictors (e.g. geneid) scan nucleotides with an general gene model, scoring features such as coding potential and splice sites. The output of these programs were condensed in gene structure models using the program Maker [Cantarel et al., 2008; Holt and Yandell, 2011], resulting in a genome annotation made of non-overlapping genes (alternative isoforms were not considered). At this point, manual annotators (including myself) intervened to check the predictions corresponding to their families of expertise. The Apollo genome annotation and curation tool [Lee et al., 2009b] was chosen as gateway. Apollo is a genome viewer that allows to visualize and edit genomic features like genes, transcripts, coding sequences. Manual annotators received and analyzed the genome assembly. Then, they loaded in Apollo the automated annotations by Maker, focusing only on the genomic regions of their gene of interest (see figure 3.4). The genomic features previously input to Maker were available to display, and additional custom annotations could be loaded using gff files.

The problems encountered in the annotation were discussed in conference calls, so even the automated annotation was improved in next releases. For example, an early problem can be seen in figure 3.5. A single gene (Smar.temp_007506) occupies the entire region on the plus strand, mostly by its impossibly large UTRs. The selenoproteins prediction corresponding to gene SelW2a (here with id SelW.1.-selenocysteine-RA) is located approximately at 0.62 Mb, and is included within its 3'UTR. When RNA-to-genome matches are given high weight in annotation pipelines, regions with such a high density of genes are sometimes problematic.

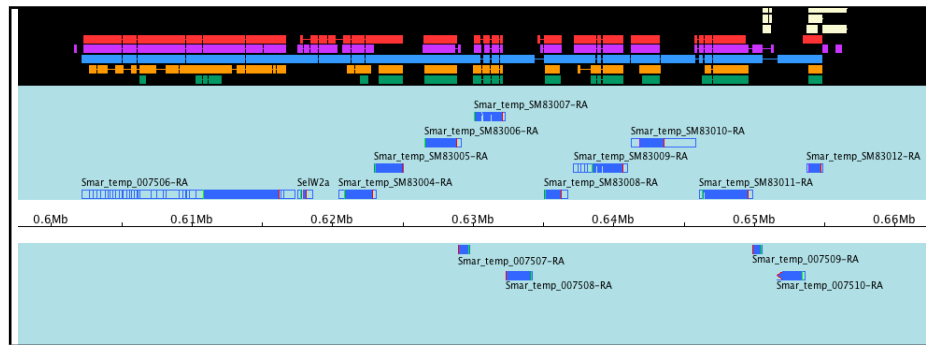


Figure 3.4: The Apollo genome viewer with the final annotation for the region of the SelW2 gene. The central white band indicate the numeric position in this scaffold. Above and below, the annotations on the plus and negative strand respectively are shown as blue rectangles. Coding sequence is indicated in darker blue. On top, the features mapped to the plus strand in this genomic region are shown with different colors (e.g. orange for tblastn matches).

Transcripts coming from consecutive (or partially overlapping) genes are joined, resulting in fused gene predictions. For this region, I searched the sequence of transcript Smar_temp.007506 for the most likely breakpoints, annotating one by one the proteins indicated with ids Smar_temp.SM83004-12, besides SelW2a.

The centipede genome revealed to be rich in selenoproteins: we found 20 (see table 3.1), along with a complete Sec machinery (tRNA^{sec}, SecS, SBP2, eEF-sec, pstk, secp43, SPS2). The centipede selenoproteome is extremely similar to the vertebrate one, with the notable exception of MsrA. Extending our search, we found Sec-containing MsrA also in other non-insect arthropods (*Daphnia pulex*, *Ixodes scapularis*) and in early chordates (*Branchiostoma floridae*). This suggests

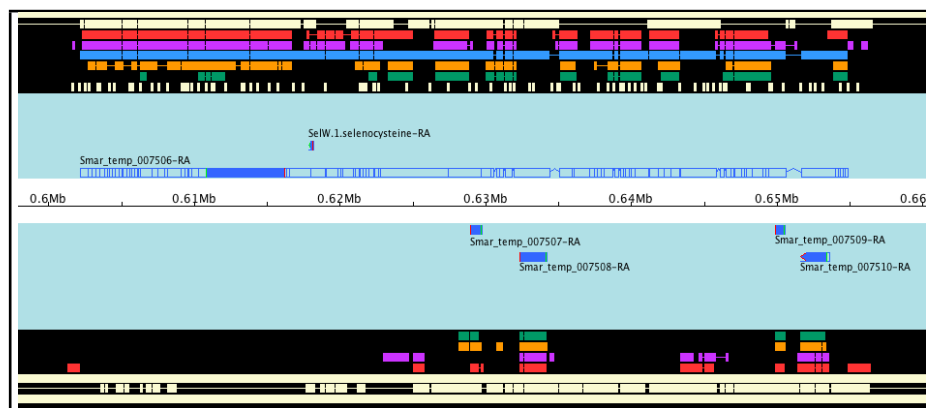


Figure 3.5: The Apollo genome viewer with the same region of figure 3.4, before manual curation.

that Sec-MsrA was present in their last common ancestor and it was later lost or converted in several lineages, both in protostomes and deuterostomes. The rich centipede selenoproteome supports again that selenoprotein extinctions are limited to insects, and can be attributed to changes in at their root.

Symbol	Gene Name
SPS2	Selenophosphate Synthetase 2
GPx1	Glutathione Peroxidase 1
GPx3	Plasma Glutathione Peroxidase 3
GPx4	Phospholipid Glutathione Peroxidase 4
TrxR1	Thioredoxin Reductase 1
TrxR2	Thioredoxin Reductase 2
MsrA	Methionine-S-Sulfoxide Reductase A
Sel15	Selenoprotein 15
SelM	Selenoprotein M
SelR	Selenoprotein R – Methionine-R-Sulfoxide Reductase B
SelT	Selenoprotein T
SelT2	Selenoprotein T2
SelU	Selenoprotein U
SelW2A	Selenoprotein W2-A
SelW2B	Selenoprotein W2-B
SelP	Selenoprotein P
SelK	Selenoprotein K
SelS	Selenoprotein S
SelO1	Selenoprotein O-1
SelO2	Selenoprotein O-2

Table 3.1: Selenoproteins identified in the genome of centipede *Strigamia maritima*.

3.2 SelenoDB 2.0

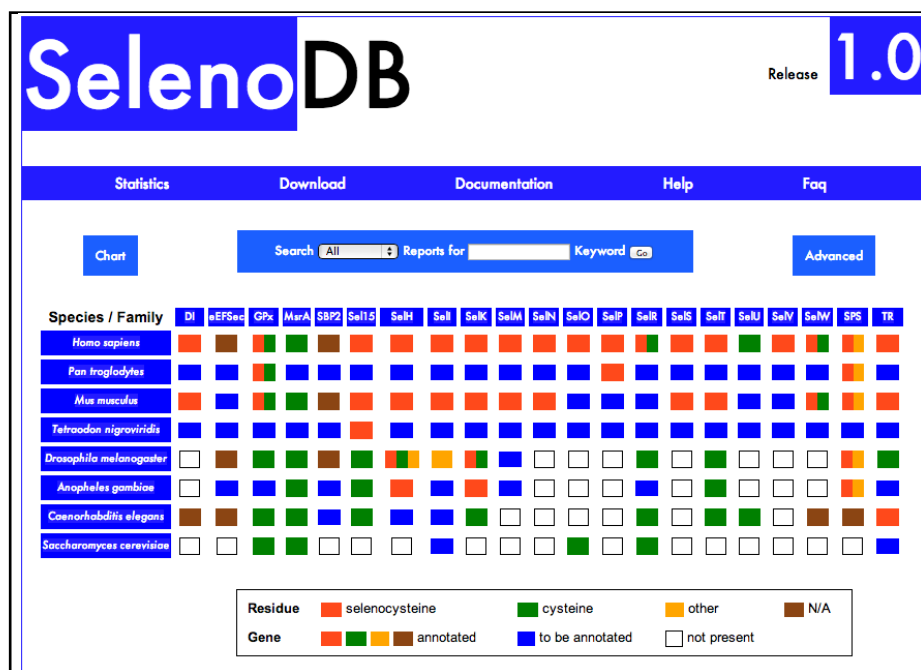


Figure 3.6: SelenoDB release 1.0. Note that the color scheme is reversed in comparison with the rest of the thesis (here it is green for cysteine, red for selenocysteine)

SelenoDB (<http://www.selenodb.org/>) was started in 2008 to provide correct selenoprotein annotations to selenium researchers [Castellano et al., 2008]. Although useful to a number of researchers (see <http://www.selenodb.org/leipzig/cite/>), the release 1.0 contained a very limited number of species, and actually only human and *D.melanogaster* were fully annotated (see figure 3.6). In 2013, we started a collaboration with Sergi Castellano, at the Max Planck Institute for Evolutionary Anthropology in Leipzig, to enrich SelenoDB with automated annotations. We provided selenoprofiles annotations for selenoproteins, cysteine homologues and Sec machinery proteins on the full set of Ensembl genomes (release 68), consisting of 57 metazoan species. We also predicted the SECIS elements of selenoproteins genes using SECISearch3. The use of automatization resulted in huge increase of annotated genes (from 81 to 2800). Among the species considered, human alone was manually annotated by Didac Santesmasses, who inspected and polished the gencode annotation. Human is also the only species for which alternative transcripts were annotated. Additionally, this new release of SelenoDB (2.0) contains variation data (single nucleotide polymorphisms, SNP) for human, bonobo and chimp. This data was obtained through exome capture and sequencing of all selenoprotein genes in the 928 human samples in the reference panel CEPH HGP

[Cann et al., 2002], including 53 different populations.

A manuscript describing SelenoDB 2.0 was recently accepted for the database issue of Nucleic Acid Research. The new version of the database will be made public very soon (it is probably active at the time you read this).

Publication: (not included in this thesis)

Romagné F, Santesmasses D, White L, Sarangi GK, Mariotti M, Hubler R, Weihmann A, Parra G, Gladyshev VN, Guigó R, Castellano S

SelenoDB 2.0: annotation of selenoprotein genes in Eukaryotes and their genetic diversity in humans. Nucleic Acids Research, Database Issue (manuscript accepted).

3.3 The vertebrate selenoproteome

Vertebrate selenoproteins have been discovered mostly in the last 15 years, combining computational and experimental techniques. Since 2007, no novel Sec genes have been discovered in this lineage, suggesting that our view of the vertebrate selenoproteome is already complete. Thus, we thought that times were mature for a comprehensive computational analysis of selenoprotein genes in vertebrate genomes. I was particularly interested in the characterization of their phylogenetic history, tracing all relevant genomic events (gene duplication, gene losses, Sec-to-Cys conversions). This work was carried out during my stay at Vadim Gladyshev's group in Boston, and drew from earlier research by members of his lab.

Publication:

Mariotti M, Ridge PG, Zhang Y, Lobanov AV, Pringle TH, Guigó R, Hatfield DL, Gladyshev VN

Composition and Evolution of the Vertebrate and Mammalian Selenoproteomes. PLoS ONE 2012 7(3): e33066. doi:10.1371/journal.pone.0033066

3.3.1 Phylogeny of selenoproteins in vertebrates and mammals

In this study we investigated the evolution of the selenoproteome of vertebrates, with particular focus on mammals. Among eukaryotic lineages, vertebrates have a quite rich, and particularly conserved selenoproteome. Many selenoproteins play essential roles. In particular, three important selenoprotein families constitute alone between a third and a half of vertebrate selenoproteomes: Glutathione Peroxidases (GPx), Thioredoxin Reductases (TR), Deiodinases (DI).

Here, we traced the evolution of all selenoproteins in sequenced vertebrates, mapping to the species tree all events of duplication, loss and conversion to cysteine of selenoprotein genes. The GPx family exhibited a particularly dynamic history: duplications occurred in bony fishes and in placentals, and many conversions to cysteine were also observed. The case of GPx6 was most interesting: this human selenoprotein is a cysteine homologue in rabbit, some rodents and also in the primate marmoset, implying independent conversions. Selenoprotein SelW showed gene losses and gains in vertebrates too, generating also selenoprotein SelV by duplication and addition of a large, possibly unstructured N-terminal domain. A few more selenoproteins originated within vertebrates, always by duplication of an existing selenoprotein gene. For SPS2, we observed an interesting gene replacement by a retrotransposed copy (SPS2b), finally resulting in intron loss in placental mammals. Marsupials still carry both the parental and the retrotransposed copy, although it is unclear whether they are both functional.

This study provided a phylogenetic atlas for researchers studying any vertebrate selenoprotein. It also includes useful data for studying the mechanisms of cysteine conversion in general, since it lists many such events in different lineages and protein families.

3.3.2 Vertebrate selenoproteome paper

Find here the manuscript as published in PLoS One in 2012. This paper contains an extensive supplementary section (42 figures), which is too large to be included here. You can access it online at:

<http://www.plosone.org/article/info%3Adoi%2F10.1371%2Fjournal.pone.0033066>.

Mariotti M, Ridge PG, Zhang Y, Lobanov AV, Pringle TH, Guigo R, Hatfield DL, Gladyshev VN. [Composition and evolution of the vertebrate and mammalian selenoproteomes.](#) PLoS One. 2012; 7(3): e33066. DOI: 10.1371/journal.pone.0033066

Composition and Evolution of the Vertebrate and Mammalian Selenoproteomes

Marco Mariotti^{1,2,3}, Perry G. Ridge^{3,9}, Yan Zhang^{1,4,9}, Alexei V. Lobanov¹, Thomas H. Pringle⁵, Roderic Guigo², Dolph L. Hatfield⁶, Vadim N. Gladyshev^{1*}

1 Brigham and Women's Hospital and Harvard Medical School, Boston, Massachusetts, United States of America, **2** Center for Genomic Regulation and Universitat Pompeu Fabra, Barcelona, Spain, **3** Department of Biochemistry and Redox Biology Center, University of Nebraska, Lincoln, Nebraska, United States of America, **4** Key Laboratory of Systems Biology, Shanghai Institutes for Biological Sciences, Chinese Academy of Sciences, Shanghai, China, **5** Spering Foundation, Eugene, Oregon, United States of America, **6** Laboratory of Cancer Prevention, National Cancer Institute, National Institutes of Health, Bethesda, Maryland, United States of America

Abstract

Background: Selenium is an essential trace element in mammals due to its presence in proteins in the form of selenocysteine (Sec). Human genome codes for 25 Sec-containing protein genes, and mouse and rat genomes for 24.

Methodology/Principal Findings: We characterized the selenoproteomes of 44 sequenced vertebrates by applying gene prediction and phylogenetic reconstruction methods, supplemented with the analyses of gene structures, alternative splicing isoforms, untranslated regions, SECIS elements, and pseudogenes. In total, we detected 45 selenoprotein subfamilies. 28 of them were found in mammals, and 41 in bony fishes. We define the ancestral vertebrate (28 proteins) and mammalian (25 proteins) selenoproteomes, and describe how they evolved along lineages through gene duplication (20 events), gene loss (10 events) and replacement of Sec with cysteine (12 events). We show that an intronless selenophosphate synthetase 2 gene evolved in early mammals and replaced functionally the original multiexon gene in placental mammals, whereas both genes remain in marsupials. Mammalian thioredoxin reductase 1 and thioredoxin-glutathione reductase evolved from an ancestral glutaredoxin-domain containing enzyme, still present in fish. Selenoprotein V and GPx6 evolved specifically in placental mammals from duplications of SelW and GPx3, respectively, and GPx6 lost Sec several times independently. Bony fishes were characterized by duplications of several selenoprotein families (GPx1, GPx3, GPx4, Dio3, MsrB1, SelJ, SelO, SelT, SelU1, and SelW2). Finally, we report identification of new isoforms for several selenoproteins and describe unusually conserved selenoprotein pseudogenes.

Conclusions/Significance: This analysis represents the first comprehensive survey of the vertebrate and mammal selenoproteomes, and depicts their evolution along lineages. It also provides a wealth of information on these selenoproteins and their forms.

Citation: Mariotti M, Ridge PG, Zhang Y, Lobanov AV, Pringle TH, et al. (2012) Composition and Evolution of the Vertebrate and Mammalian Selenoproteomes. PLoS ONE 7(3): e33066. doi:10.1371/journal.pone.0033066

Editor: Vincent Laudet, Ecole Normale Supérieure de Lyon, France

Received: December 26, 2011; **Accepted:** February 3, 2012; **Published:** March 30, 2012

Copyright: © 2012 Mariotti et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Funding: Supported by NIH GM061603 and GM065204. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing Interests: The authors have declared that no competing interests exist.

* E-mail: vgladyshev@rics.bwh.harvard.edu

These authors contributed equally to this work.

Introduction

Selenocysteine (Sec)-containing proteins (selenoproteins) have been identified in all domains of life [1–3]. In these proteins Sec is encoded by UGA, a codon typically used for termination of protein synthesis. Sec insertion is possible when a stem-loop structure, the Sec insertion sequence (SECIS) element, is located in the 3'-untranslated regions (UTRs) of selenoprotein genes in eukaryotes and archaea, and immediately downstream of Sec-encoding UGA codon in the coding regions of bacterial selenoprotein genes [4–8]. A set of selenoproteins in an organism is known as the selenoproteome. The human selenoproteome is encoded in 25 selenoprotein genes, whereas 24 selenoprotein genes were found in mouse [9].

The largest and the best studied selenoprotein families are glutathione peroxidase (GPx), thioredoxin reductase (TR) and iodothyronine deiodinase (Dio) families, with 5, 3, and 3 Sec-containing genes in the human genome, respectively. The function of approximately half of mammalian selenoproteins is not known. Among the functionally characterized selenoproteins, many have a role in redox regulation. In mice, at least three selenoproteins, cytosolic/nuclear TR (TR1, Txnrd1), mitochondrial TR (TR3, Txnrd2) and glutathione peroxidase 4 (GPx4, Phgpx), are essential [10–12] and several others, when knocked out, resulted in reduced fitness or disease [13–16]. Additionally, selenoproteins have been implicated in cancer prevention, modulation of the aging process, male reproduction, and immune response [17–21]. The mammalian selenoproteins can be broadly classified into two classes: housekeeping and stress-related [22]. Housekeeping selenoproteins

are less affected by dietary selenium (Se) status and often serve functions critical to cell survival, whereas stress-related selenoproteins are not essential for survival and often show decreased expression in Se-deficient conditions.

Previous analyses of the selenoproteome in various model organisms have revealed widely different selenoprotein sets. For example, some green algae and vertebrates have more than 20 selenoproteins, whereas red algae, insects and nematodes less than 5, and higher plants and yeast do not have any [23]. Recent studies also showed that aquatic organisms generally have larger selenoproteomes than terrestrial organisms, and that mammalian selenoproteomes show a trend toward reduced use of selenoproteins [24,25]. However, whereas a variety of organisms have been analyzed for selenoprotein occurrence [24–32], a comprehensive survey of the vertebrate or the mammalian selenoproteomes has not been done.

The aim of this work was to address questions regarding Se utilization and evolution of selenoproteins in vertebrates, focusing on mammals. We used both genomic sequences and other diverse datasets to analyze the composition, evolution, and properties of mammalian and other vertebrate selenoproteomes. We characterized the origin and loss of each selenoprotein from fish to mammals and report a comprehensive analysis of each of these proteins that revealed novel insights into the use of Sec in these organisms.

Results

Identification and comparative analysis of vertebrate selenoproteomes

We characterized vertebrate selenoproteomes by searching for all known selenoproteins in Trace Archive, non-redundant, expressed sequence tag (EST), and genomic databases of 44 vertebrates (including 34 mammals) (Figure 1 and Supplementary Table S1). The search was supplemented with the analysis of SECIS elements via SECISearch [9], and with the subsequent phylogenetic analysis of proteins belonging to the same superfamily. Overall, the searches yielded 45 selenoproteins (selenoprotein subfamilies) in sequenced vertebrates, 28 of which were found in mammals (Table 1). However, none of the mammals analyzed contained all these proteins: at most, 25 selenoproteins were detected. The largest selenoproteomes were found in bony fishes, with a maximum of 38 selenoproteins in zebrafish. The smallest selenoproteome (24 selenoprotein genes) was predicted in frog and in some mammals (Figure 1). 21 selenoproteins were found in all vertebrates: GPx1-4, TR1, TR3, Dio1, Dio2, Dio3, SelH, SelI, SelK, SelM, SelN, SelO, SelP, MsrB1 (methionine-R-sulfoxide reductase 1), SelS, SelT1, SelW1, Sep15. The other selenoproteins were found only in certain lineages, highlighting a dynamic process by which new selenoprotein genes were generated by duplication, while others were lost or replaced their Sec with cysteine (Cys). The predicted ancestral vertebrate selenoproteome is indicated in Figure 1, along with the details of its transformations across vertebrates. We found 28 proteins in the ancestral vertebrate selenoproteome and 25 in the ancestral mammalian selenoproteome.

Several selenoproteins genes were found duplicated in all bony fishes investigated, probably owing to the whole genome duplication in the early evolution of ray-finned fishes [33]. This event generated selenoproteins GPx1b, GPx3b, GPx4b, Dio3b, SelT2, MsrB1b and SelU1c. Additionally, some gene duplications were observed only in specific lineages of bony fishes. In zebrafish only, we found additional copies of SelO, SelT1 and SelW2, named respectively SelO2, SelT1b, and SelW2b. In medaka and

stickleback (*Smegmamorpha*), we identified a selenoprotein generated by a duplication of SelJ, which we named SelJ2. In *Percomorpha* (which include all bony fishes in this study apart from zebrafish), we observed a duplication of selenoprotein gene SelU1 generating SelU1b. In medaka, this gene was missing, while in stickleback Sec was replaced by Cys. Also in *Percomorpha*, we traced another duplication of SelW2, generating a selenoprotein gene that we named SelW2c. This protein lost Sec in pufferfish.

After the split with fishes, several selenoproteins were generated also in the lineage to mammals. These events are mentioned here, and their analysis will be detailed in the next section. Thioredoxin/glutathione reductase (TGR) evolved prior to the split of tetrapods through a duplication of an ancestral TR1 protein containing a glutaredoxin domain. SPS2b arose initially by a retrotransposition before the split of marsupials, while SelV and GPx6 appeared at the root of placental mammals by duplications of SelW and GPx3, respectively.

Several selenoproteins were lost across vertebrates after the terrestrial environment was colonized. This is consistent with the idea that mammals reduced their utilization of Sec compared with fishes [25]. Selenoproteins SelL and SelJ are today found only in fishes, among vertebrates. Fep15 (fish 15 kDa selenoprotein) was previously identified only in bony fishes [34]. We now identified this selenoprotein in the cartilaginous fish elephant shark and also found it as a Cys homolog in frog. These facts imply that Fep15 was a part of the ancestral vertebrate selenoproteome and was lost prior to the split of reptiles. Selenoprotein SelW2 was also lost approximately at the same point, as we find it today only in fish and frog. Finally, before the split of placental mammals selenoproteins SPS2a and SelPb were lost. We observed a few selenoprotein losses also in bony fishes: SelW1 was lost in *Percomorpha*, and selenoproteins SelU1b and GPx1b were lost in medaka.

One process contributing to the reduction of selenoproteome is the conversion of Sec to Cys. This process is specific to selenoproteins and can be accomplished by a single point mutation can transform a Sec UGA into a Cys codon. However, it has to be noted that Sec and Cys are not functionally equivalent, and Cys conversions are not neutral, although the reasons are still unclear [32]. We observed 12 conversions to Cys along vertebrates, 8 of which happened after the split of mammals (Figure 1). Some were found common to many organisms and were mapped back to their common ancestor (e.g. SelU1 in mammals), while others were found in relatively narrow lineages, sometimes even in single species (e.g., GPx6 in marmoset).

Comparative analyses of selenoprotein families

We built multiple sequence alignment for all vertebrate selenoproteins (Supplementary Figures S1, S2, S3, S4, S5, S6, S7, S8, S9, S10, S11, S12, S13, S14, S15, S16, S17, S18, S19, S20, S21, S22, S23, S24, S25, S26, S27, S28, S29, S30) and analyzed their phylogenetic relationships and sequence features. The most conserved selenoprotein was SelT, with an impressive identity across all mammals even at the nucleotide sequence level (Supplementary Figure S31). Below, we report our analysis for the selenoprotein families with most interesting findings.

Selenophosphate synthetase 2. The function of selenophosphate synthetase 2 (SPS2) is to generate the Se donor compound (selenophosphate) necessary for Sec biosynthesis, and interestingly it is itself a selenoprotein. Although SPS2 was found as a selenoprotein in all vertebrates, we observed that a gene replacement took place. In mammals, the SPS2 gene appeared initially as a multiple exon gene (SPS2a), but was then replaced by a single exon copy (SPS2b). In monotremes and non-mammalian

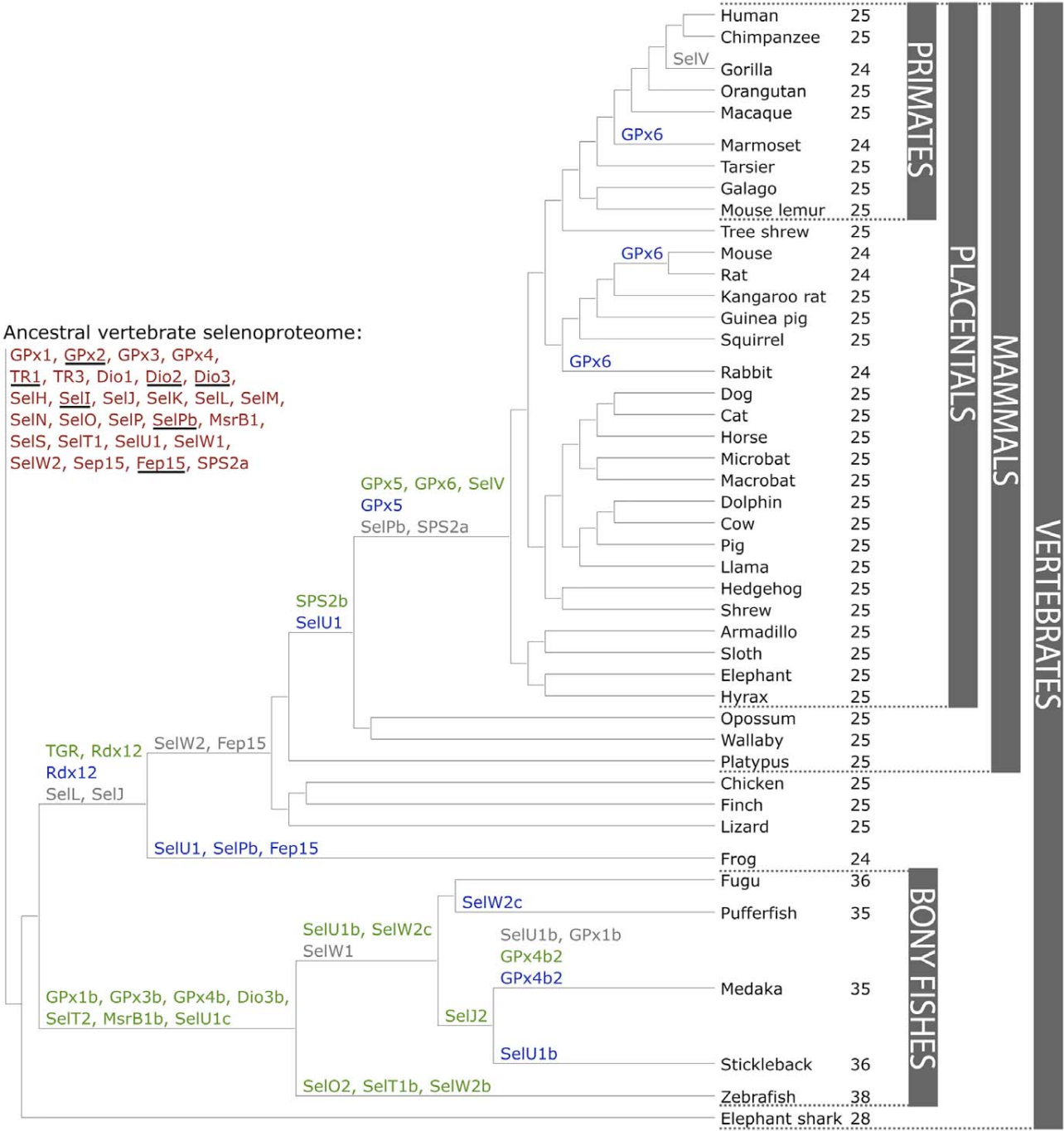


Figure 1. Evolution of the vertebrate selenoproteome. The ancestral vertebrate selenoproteome is indicated in red, and its changes across the investigated vertebrates are depicted along their phylogenetic tree. The ancestral selenoproteins found uniquely in vertebrates are underlined. The creation of a new selenoprotein (here always by duplication of an existing one) is indicated by its name in green. Loss is indicated in grey. Replacement of Sec with Cys is indicated in blue (apart from SelW2c in pufferfish, which is with arginine). Events of conversion of Cys to Sec were not found. On the right, the number of selenoproteins predicted in each species is shown.
doi:10.1371/journal.pone.0033066.g001

vertebrates, only SPS2a is present, in placental mammals only SPS2b is present, whereas marsupials still possess both genes (Figure 2). The protein alignment of SPS2a/b is provided in Supplementary Figure S22. In opossum, both SPS2a and SPS2b have strong SECIS elements (Supplementary Figure S32), and the distance between the stop codons and the SECIS element is

comparable in the two versions (596 nucleotides in the single exon version and 555 nucleotides in the multi-exon version). UGA-to-SECIS distances are also comparable (1805 versus 1542 nucleotides, respectively). Due to lack of transcription data, we cannot be sure that both versions are active. In wallaby (another marsupial), we also detected both SPS2a and SPS2b genes. In this

Table 1. Vertebrate Selenoproteins.

Selenoproteins	Commonly used abbreviations	Fish	Frog	Birds	Mammals		
					Platypus	Marsupials	Placentals
15 kDa selenoprotein	Sep15, Sel15	+	+	+	+	+	+
Fish 15 kDa selenoprotein-like	Fep15	+					
Glutathione peroxidase 1a	GPx1, GSHPx1, GPx, cGPx	+	+	+	+	+	+
Glutathione peroxidase 1b	GPx1b	+					
Glutathione peroxidase 2	GPx2, GSHPx-GI, GPRP, GI-GPx, GSGPx-2	+	+	+	+	+	+
Glutathione peroxidase 3	GPx3, pGPx, GPx-P, GSHPx-3, GSHPx-P, EGPx+	+	+	+	+	+	+
Glutathione peroxidase 3b	GPx3b	+					
Glutathione peroxidase 4a	GPx4, PHGPx, MCSP, snGPx, snPHGPx, mtPHGPx	+	+	+	+	+	+
Glutathione peroxidase 4b	GPx4b	+					
Glutathione peroxidase 6	GPx6, OMP						+
Iodothyronine Deiodinase 1	Dio1, DI1, 5DI, TXDI1, ITDI1	+	+	+	+	+	+
Iodothyronine Deiodinase 2	Dio2, DI2, D2, 5DII, TXDI2, SelY	+	+	+	+	+	+
Iodothyronine Deiodinase 3a	Dio3, DI3, 5DIII, TXDI3	+	+	+	+	+	+
Iodothyronine Deiodinase 3b	Dio3b, DI3b	+					
Methionine-R-Sulfoxide Reductase 1a	MsrB1, SelR, SelX, SepR,	+	+	+	+	+	+
Methionine-R-Sulfoxide Reductase 1b	MsrB1b	+					
Selenophosphate Synthetase 2a	SPS2a, SEPHS2, Ysg3	+	+	+	+	+	
Selenophosphate Synthetase 2b	SPS2b					+	+
Selenoprotein H	SelH, SepH	+	+	+	+	+	+
Selenoprotein I	SelI, SepI	+	+	+	+	+	+
Selenoprotein J	SelJ	+					
Selenoprotein J2	SelJ2	+					
Selenoprotein K	SelK, SelG, SepK	+	+	+	+	+	+
Selenoprotein L	SelL	+					
Selenoprotein M	SelM, SepM	+	+	+	+	+	+
Selenoprotein N	SelN, SepN1, RSS, MDRS1, RSMD1	+	+	+	+	+	+
Selenoprotein O	SelO, SepO	+	+	+	+	+	+
Selenoprotein O2	SelO2	+					
Selenoprotein P	SelP, SeP, SepP1, Se-P, SelPa	+	+	+	+	+	+
Selenoprotein Pb	SelPb	+		+	+	+	
Selenoprotein S	SelS, VIMP, ADO15, SBB18, SepS1, AD-015	+	+	+	+	+	+
Selenoprotein T1a	SelT1a, SepT	+	+	+	+	+	+
Selenoprotein T1b	SelT1b	+					
Selenoprotein T2	SelT2	+					
Selenoprotein U1a	SelU1, SepU1	+		+	+		
Selenoprotein U1b	SelU1b, SepU1b	+					
Selenoprotein U1c	SelU1c, SepU1c	+					
Selenoprotein V	SelV, SepV						+
Selenoprotein W1	SelW1, SeW, SepW1	+	+	+	+	+	+
Selenoprotein W2a	SelW2a	+	+				
Selenoprotein W2b	SelW2b	+					
Selenoprotein W2c	SelW2c	+					
Thioredoxin reductase 1	TR1, TxnRd1, TxnR, TrxR1, GRIM-12	+	+	+	+	+	+
Thioredoxin reductase 3	TR3, TR2, TxnRd2, SelZ, TrxR2, TR-Beta	+	+	+	+	+	+
Thioredoxin/glutathione reductase	TGR, TR2, TR3, TxnRd3, TrxR3		+	+	+	+	+

Selenoproteins detected by genomic searches in vertebrate genomes are shown. The groups for which a given selenoprotein was found in at least one organism are marked.

doi:10.1371/journal.pone.0033066.t001

case though, we could not reconstruct the entire genes due to incomplete genome assembly. In addition, SPS2a sequence appears to contain a 2 bp insertion in the penultimate exon, which would result in a frameshift. Nonetheless, given the very high conservation of the gene also downstream of the insertion and the poor coverage of sequence data, we think that this is sequencing/assembly artifact and that the gene is intact and functional.

Overall, our results suggest that SPS2b arose by reverse transcription following the monotreme/marsupial split and eventually replaced SPS2a in placental mammals. Interestingly, opossum SPS2a is located on the X chromosome. Although it must be said that the number of available genomes assembled in chromosomes is quite limited, this is the only case in which we found an SPS2 gene on a mammalian sexual chromosome. This is almost unique also when considering all mammalian selenoprotein genes: the only exceptions are platypus GPx6 residing on chromosome X1 (though the sex chromosome system of monotremes is radically different from other mammals and is still poorly understood [35]) and a pseudogene of GPx1, described later, which is localized on chromosome X. Selenoproteins and Se pathways are linked to sex-specific traits [36]. It is known that the X chromosome is overrepresented with sex-specific genes, and is a preferred site for retrotranspositions both on and off [37]. It could

be speculated that the retrotransposition generating SPS2b and its subsequent functionalization may have been a response to a previous chromosome rearrangement that brought the SPS2a gene on the chromosome X at the root of marsupials.

SelV and SelW. SelV was the least conserved mammalian selenoprotein (Supplementary Figure S19) that likely arose from a duplication of SelW in the placental stem. The functions of SelV and SelW are not known, but SelV is expressed exclusively in testes [9], whereas SelW is expressed in a variety of organs. SelW and SelV exhibited the same gene structure; each contained 6 exons with intron locations and phases conserved. Coding regions were within exons 1–5. Exon 6 contained only the last portion of the 3'-UTR, including the SECIS element. Significant variation between SelW and SelV was found only in exon 1. Translated protein length of this exon has an average length of 261 residues (ranged from 228 amino acids in cat to 334 in dog), in contrast to SelW that had only 9 residues derived from exon 1 in most mammals. Only the last four residues of SelW and SelV in exon 1, which were located immediately upstream of the CxxU motif, were conserved; in contrast, their homology was high in exons 2–5 (Figure 3) as well as in the SECIS element in exon 6 (Supplementary Figure S28), suggesting that evolution of SelV by SelW gene duplication might have followed up by the addition of N-terminal sequences. Additional changes were observed in the last exon (exon 6, Supplementary Figure S33). First, a shift in the 5' splicing site of SelV exon 6 was identified, with the effect of shortening the sequence preceding the SECIS element in this exon. SelW exon 6 had an average of 38 nucleotides from the beginning to the SECIS core, in contrast to SelV which had an average of 13 nucleotides. Second, compared to SelW, a substantial portion of the 3'-UTR downstream of the SECIS element was lost in SelV. Both changes resulted in a much shorter 3'-UTR in SelV (152 nucleotides on average) than SelW (358 nucleotides).

In a recent paper, it was reported that SelV was lost by deletion specifically in gorilla [38]. Our results confirm this finding. Indeed, we did not find SelV in any available sequences from this organism. Also, we could identify the region in the gorilla genome syntenic to the human SelV gene: consistent with a gene-specific deletion, the neighboring genes were present and conserved, while SelV was missing.

Several SelW homologs were observed across non-mammalian vertebrates. Phylogenetic analysis revealed a distinct group of proteins, SelW2. We found SelW2 as a selenoprotein in bony fishes, but also in frog and in elephant shark, which suggests that it was part of the ancestral vertebrate selenoproteome. In mammals, only a remote homolog of SelW2 is present: Rdx12 [39], which is not a selenoprotein and aligns a Cys to the Sec residue of SelW2. Frog is the only species in which we found both selenoprotein SelW2 and Rdx12. In all other tetrapods, we found only Rdx12. Thus we hypothesize that before the split of amphibians SelW2 duplicated and was immediately converted to a Cys form generating Rdx12, and then SelW2 was lost prior to the split of reptiles.

In bony fishes, we observed multiple copies of SelW2, whose phylogenetic relationships are very hard to entangle. Zebrafish had two copies of SelW2 (SelW2a, SelW2b), both selenoproteins, located in tandem on chromosome 3. The rest of bony fishes (Percomorpha) had a SelW2 protein similar to both SelW2a and SelW2b, plus a second protein located on a different chromosome (or scaffold), which we named SelW2c. In contrast, they all appear to have lost SelW1. Phylogenetic analysis shows that SelW2c proteins do not cluster with SelW2b, with Rdx12 or with SelW1 (Supplementary Figure S34). We think that most likely the

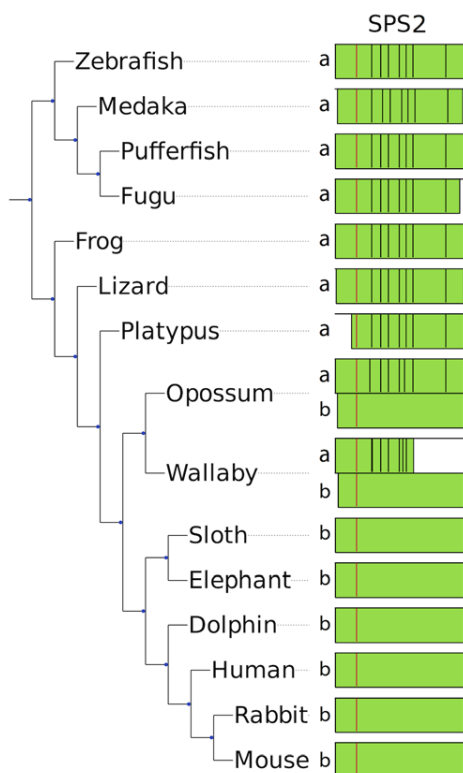


Figure 2. Replacement of a multiexon SPS2a by an intronless SPS2b. In the figure, the SPS2 genes found in some representative species are shown. The positions of introns along the protein sequence are displayed with black lines, and the Sec residue is displayed in red. In a few cases, the predicted genes were incomplete because of poor sequence data (e.g., the N-terminal region in platypus). Placental mammals (bottom) possess a single intronless gene, SPS2b. Non-mammalian vertebrates (top) and platypus possess a single multiexon gene, SPS2a. Marsupials (opossum and wallaby) possess both. doi:10.1371/journal.pone.0033066.g002

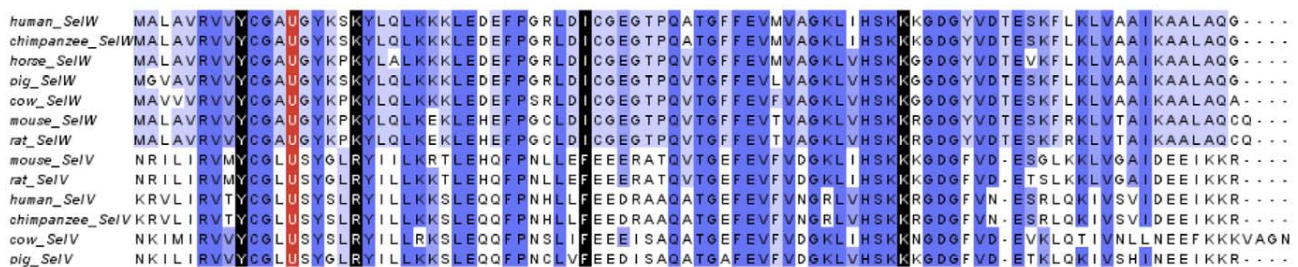


Figure 3. Multiple sequence alignment of SelV and SelW. The last 9 residues of SelV exon 1 and exons 2–5 are shown aligned to complete SelW sequences. The last residue of each exon is marked in black and the Sec in red.
doi:10.1371/journal.pone.0033066.g003

SelW2a/b tandem duplication was specific to zebrafish, and that SelW2c was generated by another duplication of SelW2 at the root of Percomorpha, more or less concomitant with the SelW1 loss. Nonetheless, we cannot exclude a possibility that SelW2c is actually one of the two genes SelW2a/b or SelW1, which would have increased abruptly the sequence divergence rate in Percomorpha, confounding the phylogenetic reconstruction. Interestingly, SelW2c in pufferfish is not a selenoprotein: the Sec codon was mutated to an arginine codon (CGA) and SECIS element was lost or degenerated. Therefore, the CxxU domain that is present in all SelW, SelW2 and SelV proteins is CxxR in this protein. We found evidence of the expression of this gene in ESTs.

Glutathione peroxidases. Glutathione peroxidases are the largest selenoprotein family in vertebrates. Mammals have 8 GPx homologs, 5 of which are selenoproteins: GPx1–4, GPx6. We present here an unambiguous phylogeny of the GPx tree wherein three evolutionary groups were observed: GPx1/GPx2, GPx3/GPx5/GPx6, and GPx4/GPx7/GPx8 (Figure 4). Our findings are consistent with another study that examined GPx evolution [31]. It appeared that Cys-containing GPx7 and GPx8 evolved from a GPx4-like selenoprotein ancestor, but this happened prior to separation of mammals and fishes. GPx5 and GPx6 are the most recently evolved GPxs, which appeared to be the result of a tandem duplication of GPx3 at the root of placental mammals. Interestingly, no Sec-containing GPx5 form could be identified. As phylogeny indicates that this protein evolved from a duplication of selenoprotein GPx3, the Sec to Cys displacement must have happened very early in the evolution of GPx5.

For GPx6, we observed several independent Cys conversions: in the primate marmoset, in rat and mouse, and in rabbit (Figure 1). We suggest that the Cys-containing GPx6 was not present in the last ancestor of rabbit and rodents because the Sec-containing GPx6 was observed in other rodents, such as squirrel, guinea pig, kangaroo rat. In bony fishes, we observed three GPx duplications, generating GPx1b, GPx3b and GPx4b. All investigated species of this branch were found to have these three genes, with the exception of medaka, which apparently lost GPx1b. In this same species, we found an additional Cys copy of GPx4b, that we named GPx4b2.

Each of the mammalian Sec-containing GPx genes was highly conserved. Four of the five had better than 80% nucleotide sequence identity, while GPx1 had ~70% sequence identity within mammalian sequences. GPx4 was one of the most conserved selenoproteins with better than 90% nucleotide sequence identity. Furthermore, considering full length selenoprotein sequences (i.e., including signal peptides), GPx4 had the highest level of conservation of any selenoprotein.

Thioredoxin reductases. TRs control the redox state of thioredoxins, key proteins involved in redox regulation of cellular processes. Mammals have three TR isozymes: cytosolic TR1, _itochondrial TR3, and TGR. Only two of these, TR1 and TR3, were detected in fish genomes. We thus investigated the phylogenesis of TGR. Previous studies have revealed various transcript (splicing forms) and/or protein (isoforms) variants in each mammalian TR in mammals [40–46] (see reviews [47,48]). All TR1 alternative splicing was upstream of the first coding exon (exon 1) of the major form of TR1. Upstream exons were given these letter designations, 5' to 3': U1, A, U2, B, C, D1, D2, E, F, G, and H. Among the many splicing forms of TR1, one coded for an N-terminal Grx domain (Grx-TR1) [40,42]. This TR1 form was derived from alternative exons A, B, C, and E (followed by common exons), with translation beginning in exon A. We found that in fish the Grx domain is present in the major form of TR1. In mammals, the major form of TR1 lacked this domain, but this occurred in TGR and in the TR1 alternative isoform mentioned above. Notably, the Grx-TR1 isoform was absent in rodents (but its fossil sequences could be identified [40]).

Sequence-based phylogenetic analyses suggested that mammalian TGR and TR1 evolved by duplication of the protein that corresponds to fish TR1. TR1 and TGR first appeared together in amphibians. Comparing mammalian TGR and the Grx-TR1 form with fish TR1, we found significant homology among the three proteins (Supplementary Figure S35). In addition, exon and intron boundaries were the same in all three genes. Interestingly, zebrafish TR1 had higher homology to mammalian TGR than to mammalian Grx-TR1. On the other hand, synteny placed zebrafish TR1 together with mammalian TR1 based on conservation of the downstream gene (upstream genes were different in all three TR genes). Overall, the data suggests that mammalian TR1 and TGR evolved by gene duplication from the ancestral protein that is similar to fish TR1, and this happened prior to the appearance of amphibians. Some time after the duplication, the Grx domain was retained in TR1 only in an alternative isoform, which was lost in rodents.

Sequence analysis highlighted also an important change in the predicted active site of Grx domains of mammalian TGR. In Grx-TR1s, fish TR1s and amphibian, reptile, and bird TGRs, we find a conserved CxxC motif. In mammalian TGRs, the second Cys in the Grx domain of TGR was mutated to serine (CxxS motif). Additionally, we found an interesting form of Grx-TR1 in cow where the motif was CRC. In the CxxC motif, the two Cys residues may form a catalytic disulfide bond, similar to fish and mammals.

Another interesting isoform of TR1 is one identified in a previous study, containing a thioredoxin-fold domain [40]. In this isoform, alternative exons B and H, or just H, were included upstream of exon 1 with translation beginning in exon H. While

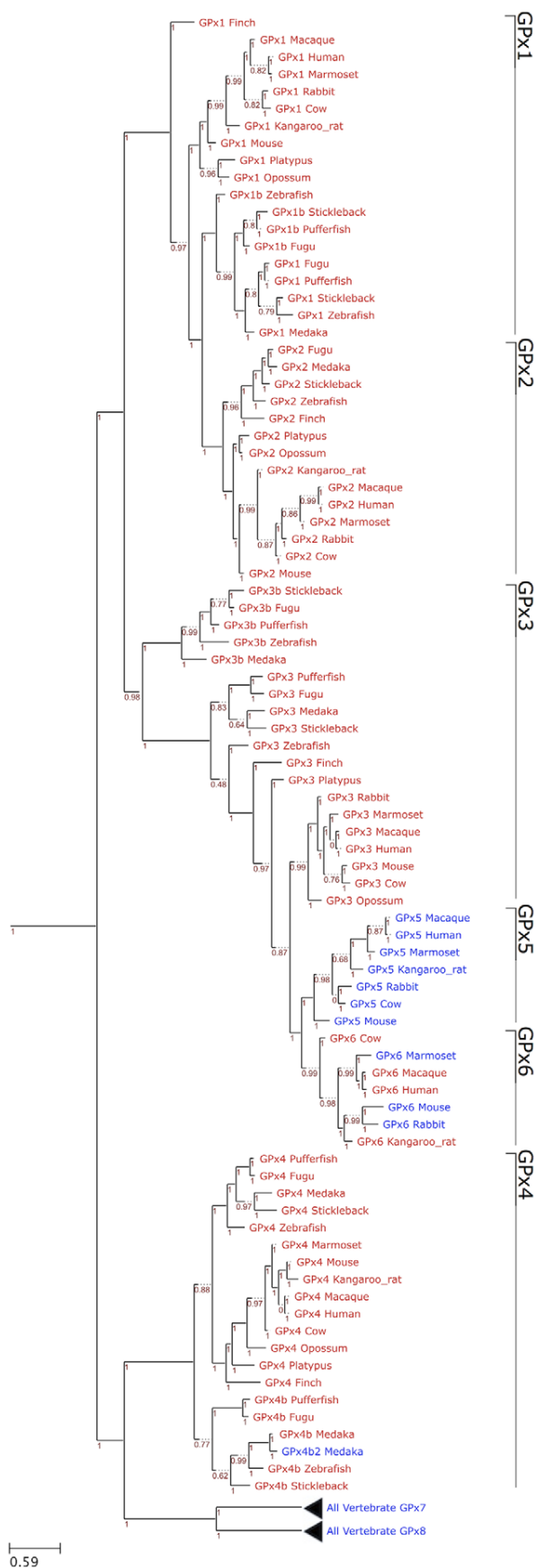


Figure 4. Phylogenetic tree of GPx family in eukaryotes. The figure shows a ML tree computed using the JTT substitution model. In the phylogram, Sec-containing proteins are shown in red and Cys-containing homologs are shown in blue. The GPx families are indicated on the right. The distance scale in substitutions per position is indicated at the bottom left. The branch support is shown in red. doi:10.1371/journal.pone.0033066.g004

EST data were found for this version only in rodents, there was overwhelming sequence similarity of exon H that suggests its importance even if there is a lack of transcriptome data. Exon H, and by extension this isoform, was identified in all placental mammals, but was absent in early mammals and in the rest of vertebrates. One last isoform is worth mentioning: isoform 4 [40], consisting of exons D1, D2, and E (with translation beginning in exon D2), was found to already occur in chicken, and was easily identified by sequence similarity in many inspected mammals (horse, opossum, and all rodents). Furthermore, EST data from humans, cows, and chickens confirmed the widespread expression of this isoform in a variety of tissues.

Iodothyronine deiodinases. The iodothyronine deiodinases (Dio) regulate activation and inactivation of thyroid hormones [49]. There are three Dio enzymes known in mammals, all of which contain Sec: Dio1, Dio2, Dio3. The deiodinases possess a thioredoxin-fold and show significant intrafamily homology. As mentioned above, we found the protein Dio3 duplicated in all bony fishes (Dio3b). Dio3 irreversibly inactivates the thyroid hormone by deiodination of the inner tyrosyl ring [50]. Interestingly, all detected Dio3 genes (including Dio3b) are intronless. All other genes in vertebrates, apart from SPS2b, were found to consist of multiple exons.

Dio2 is an ER-resident protein which activates the thyroid hormone by deiodination of the outer tyrosyl ring [50]. An interesting feature in Dio2 is that its mRNA has a second in-frame UGA codon. It was previously found that, in a cell culture system, the second UGA could insert Sec when the first UGA codon was mutated [51]. We extended translation to the next stop codon (after the second UGA), which was located an additional 9 (all mammals with the exception of primates) to 21 (in primates only) nucleotides downstream, but the additional amino acids were not conserved (Supplementary Figure S36). Thus, it appears that the primary function of the second UGA is to serve as stop codon.

Selenoprotein I. Selenoprotein I (SelI) is one of the least studied selenoproteins. It contains a highly conserved CDP-alcohol phosphatidyltransferase domain. This domain is typically encountered in choline phosphotransferases (CHPT1) and choline/ethanolamine phosphotransferases (CEPT1). CHPT1 catalyzes the transfer of choline to diacylglycerol from CDP-choline [52]. CEPT1 catalyzes an analogous reaction but accepts both choline and ethanolamine. SelI has seven predicted transmembrane domains, which correspond to the predicted topologies of CHPT1 and CEPT1. The most critical portion of this structure is located between the first and second transmembrane domains, and there are three aspartic acids, which are critical for function. Figure 5 shows an alignment of the active site region of SelI and its closest sequence homologs. The full alignment is shown in Supplementary Figure S37, and a phylogenetic tree based on that alignment is shown in Supplementary Figure S38. Not only are the three aspartic acids conserved in all SelI proteins, but the entire active region is highly similar between SelI and its homologs. The most prominent difference between SelI and its homologs is a C-terminal extension in SelI, which contains Sec. The function of this extension is unknown. We were unable to find Cys forms with homology to the SelI C-terminal extension.

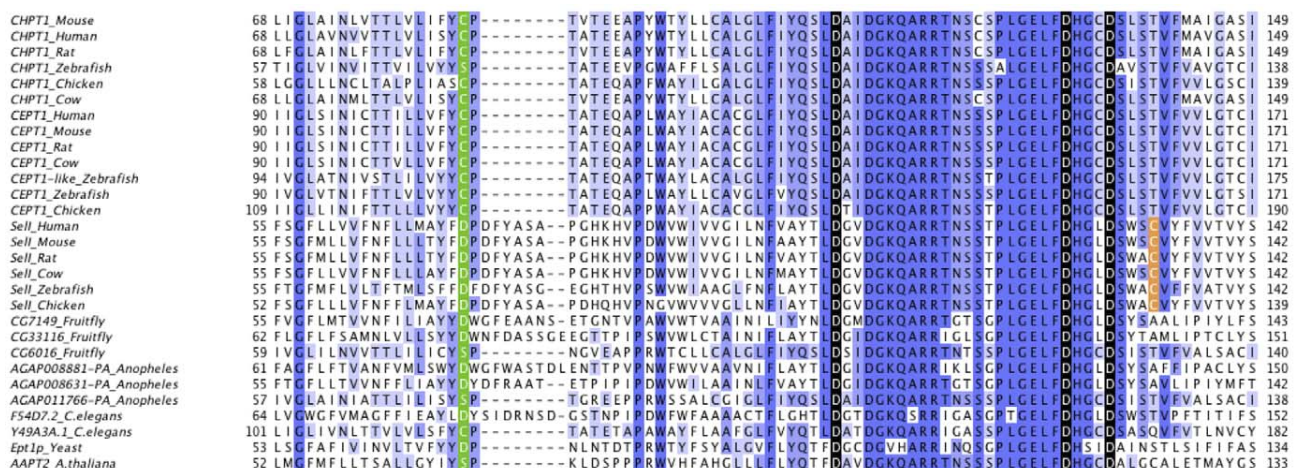


Figure 5. Multiple sequence alignment of Sell and its homologs. The multiple sequence alignment of the active site and preceding regions of CHPT1, CEPT1, and Sell is shown. Homologs are labeled with the annotated name. Proteins in the bottom section comprise a large group of diverse proteins containing the same domain. The most critical residues are marked in red. The residue in green marks the end of the first transmembrane domain. The cysteine residue near the active site emerged specifically in Sell proteins is marked in orange. The full length alignment is provided in Supplementary Figure S37 and the corresponding phylogenetic tree in Supplementary Figure S38. doi:10.1371/journal.pone.0033066.g005

Sec residues are often involved in selenenylsulfide bonds with cysteines. Thus, we searched for cysteines emerged specifically in Sell proteins. We selected the cysteines completely conserved in Sell sequences and missing in all other sequence homologs. There were three such cysteines, at positions 133, 229 and 310 of human Sell (Supplementary Figure S37). The cysteine at position 133 (Figure 5) is the best candidate: it is predicted to reside on the same membrane side (internal) as the Sec, and it is also extremely close to the conserved aspartic acids.

In a recent work [53], human Sell protein was tested for CHPT1/CEPT1 enzymatic activities, reporting a specific ethanolamine phosphotransferase (EPT) activity. However, the authors used a bacterial expression system for purification of human Sell. Since eukaryotic SECIS elements are not recognized in bacteria, a truncated form of Sell was expressed, lacking the Sec residue and the rest of the C-terminus. Therefore, the function of intact Sell may be different, especially since the Sec residue of selenoproteins is known to be essential for function. Truncated forms of some selenoproteins, such as TR [54], show activity towards non-primary substrates. As truncated forms of selenoproteins are normally not observed in vivo, most of such activities are probably not biologically relevant. For these reasons, we believe that the real molecular function of Sell has yet to be discovered. One plausible possibility is that the EPT activity is just the first step in Sell function, with phosphatidylethanolamine further processed in a Sec-dependent step. Another possibility is that the Sec extension provides completely different substrate specificity to Sell.

Vertebrate-specific selenoproteins

We were interested to know what fraction of the vertebrate selenoproteome is found uniquely in vertebrates. Therefore, we searched all vertebrate selenoproteins in the sequenced basal chordates (amphioxus, tunicates), and, as a control, in any other sequenced eukaryotes as well. Among the ancestral 28 selenoproteins, 6 were detected uniquely in vertebrates: Fep15, GPx2, Dio2, Dio3, Sell, SelPb. Most of them (Fep15, GPx2, Sell, SelPb) showed at least partial conservation of intron structure with their closest homologs (Sep15, GPx1, CDP-alcohol phosphatidyltransferases, SelP, respectively). This may suggest that they were

generated during the whole genome duplication occurred at the root of vertebrates [55]. These 6 selenoproteins, together with the 17 selenoproteins generated through duplication within vertebrates (GPx1b, GPx3b, GPx4b, GPx6, Dio3b, SelT1b, SelT2, MsrB1b, SelU1b, SelU1c, SelW2b, SelW2c, SelJ2, SelO2, TGR, SPS2b, SelV), constitute the set of vertebrate-specific selenoproteins.

Analysis of UTRs, SECIS elements and UGA locations of mammalian selenoprotein genes

The untranslated regions (UTRs) of mRNAs are important sites where regulatory elements are typically found. 3'-UTRs are especially important for selenoprotein mRNAs as this is the location of SECIS elements in eukaryotes and archaea. We analyzed the lengths of 5' and 3'-UTRs of mammalian selenoproteins (Supplementary Figures S39, S40). On average, the length of 5'-UTRs was 127 nucleotides, whereas that of 3'-UTRs was 1027 nucleotides. These observations fit the general characteristic of vertebrate mRNAs [56]. Dio2 had both the longest average 5'- and 3'-UTRs of all selenoprotein genes (409 and 5174 nucleotides, respectively). The shortest average 5'-UTR was observed in SelO, 61 nucleotides. SelV, despite having its 3'-UTR split into two exons, had the shortest 3'-UTRs with an average length of 152 nucleotides. We also examined selenoprotein lengths versus the UTR size, but did not observe significant correlation between them.

The SECIS element is present in all eukaryotic selenoprotein genes and is the fundamental signal for Sec insertion. While the overall stem-loop structure of the SECIS element is critical to its function, several especially important regions (and bases) have been identified. First, the base of the main stem has non-Watson-Crick interacting bases, known as the Quartet, or the core, including the invariant GA/GA pairs [5,57,58]. Next, in the apical loop, two unpaired bases are important for function, although their exact role is not known. In most cases, these two bases are AA. A comprehensive analysis of vertebrate SECIS elements showed that, as expected, almost all examined SECIS elements have the GA/GA quartet and AA in the apical loop. The exception included two selenoproteins, SelM and SelO, in which

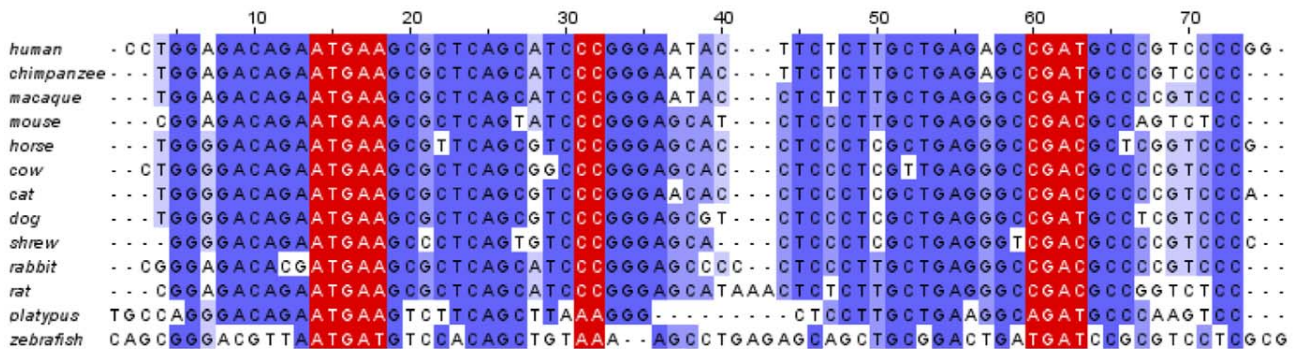
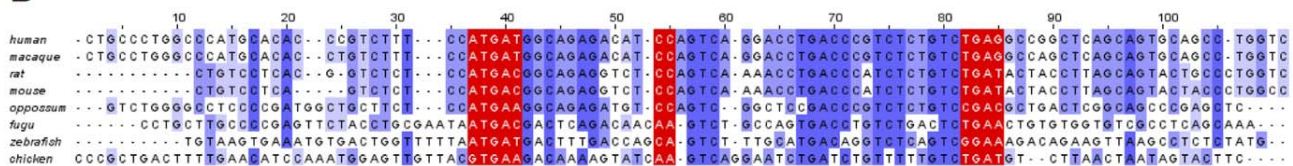
A**B**

Figure 6. SECIS elements of SelM and SelO. Multiple sequence alignment of SelM (A) and SelO (B) SECIS elements. Critical regions are marked in red.
doi:10.1371/journal.pone.0033066.g006

we found CC in the apical loop (Figure 6). The CC in SelM SECIS elements was only found in placental mammals, while all other vertebrates had AA. In SelO SECIS elements, the CC sequence was found in all mammals and in no other species. Thus, it appears that the CC forms of SECIS elements evolved specifically in mammals. Further analysis did not show any significant features that correlate with the presence of CC pattern.

The SECIS elements of most selenoprotein genes were wholly contained within the exon containing the stop codon. However, in several selenoprotein genes (SelH, SelT, SelV, SelW, and TR3), the 3'-UTR was split between two exons and the SECIS element was entirely located in the last exon. In one selenoprotein, SelK, the exon which contained the stop codon had a splice site immediately adjacent to the stop codon, and thus, the entire 3'-UTR was located in the next (last) exon. Finally, the two selenoproteins SelL and SelP have multiple Sec residues. The two Sec residues in SelL are only two residues apart and are inserted with the help of a single SECIS element [59]. SelP has a varying number of Sec residues and is unique in that it contains two SECIS elements. These two SECIS elements were separated by an average of 334 nucleotides and were always located in the same exon in the 3'-UTR in examined vertebrates.

To better understand general features of Sec insertion, we examined the distance between Sec-encoding UGA codons and SECIS elements (UGA-to-SECIS). Previous studies have attempted to define a minimal distance between these cis-elements in the mRNA. In one study performed on Dio1, the minimum spacing was defined as 51–111 nucleotides [60]. Other studies have shown that the location of the UGA can be varied within the gene and still maintain efficient UGA decoding, and that a SECIS element can be added to a non-selenoprotein 3'-UTR and an in-frame UGA be decoded as Sec [57,61,62]. We observed a wide range of UGA-to-SECIS distances (from 207 to 5207 nucleotides) for mammalian selenoproteins, all greater than the 51–111 base minimum. The average distance for all mammalian selenoproteins

was 872 nucleotides. Dio2 and TR3 had the average longest and shortest UGA-to-SECIS distances, respectively.

Identification of pseudogenes

Over the years, pseudogenes have been described for various selenoproteins, such as GPx1 [63], SelW [64], GPx4 [65], GPx2 [66], and Sep15 [67]. In our study, a total of 11 selenoprotein genes were found to be represented by additional pseudogenes in mammals (Table 2). Most of these pseudogenes had frameshifts or other mutations compromising their functionality. We observed a tendency for shorter selenoproteins to have more pseudogenes. The average length of selenoproteins with pseudogenes was 182 amino acids (10 kb genes), whereas selenoproteins which had no pseudogenes had an average length of 386 amino acids (24 kb genes).

Among the 11 selenoproteins with pseudogenes, SelK had more than any other selenoprotein (27 pseudogenes in 11 organisms), and rodents had the highest number. For example, mouse and rat had 5 and 4 SelK pseudogenes, respectively. SelW was another selenoprotein, which had many pseudogenes (19 in 13 organisms). An interesting GPx1 pseudogene was identified in humans and chimpanzees. The active site (surrounding the Sec) was conserved in both the functional and pseudogene versions of GPx1 and the overall conservation was quite high (Figure 7A). Three codon positions were particularly interesting (positions 6, 114, and 123). At each of these positions the residues translated from the pseudogenes matched, but were different than the residues in the corresponding position in GPx1. Therefore, it appeared the GPx1 pseudogene had been maintained since the human/chimpanzee split with few differences between the human and chimpanzee copies of the pseudogene. Furthermore, SECIS elements were also intact in these pseudogenes. However, a single base mutation at amino acid 161 in the human pseudogene sequence (TGG->TAG) resulted in a premature stop codon downstream of the active site. Due to this mutation and no supporting EST data, it is

Table 2. Mammalian selenoprotein pseudogenes.

Selenoprotein	# Pseudogenes	Organisms (# pseudogenes)
SelT	9	Human (2), Chimpanzee (2), Mouse (2), Rabbit (2), Horse
GPx1	3	Human (1), Squirrel (1), Rabbit (1)
GPx2	1	Human (1)
GPx4	4	Mouse (2), Rat (1), Microbat (1)
MsrB1	3	Human (1), Chimpanzee (1), Macaque (1)
SelH	5	Rat (1), Rabbit (1), Shrew (1), Hedgehog (1), Armadillo (1)
SelK	>27	Human (3), Chimpanzee (3), Macaque (3), Galago (2), Mouse (5), Rat (4), Guinea Pig (1), Squirrel (2), Dog (1), Cat (1), Microbat (2)
SelS	1	Hedgehog (1)
SelW	19	Human (2), Chimpanzee (2), Orangutan (1), Gibbon (1), Macaque (2), Rat (1), Cow (2), Dog (1), Cat (2), Microbat (1), Hedgehog (1), Elephant (1), Armadillo (2)
Sep15	5	Galago (1), Dog (1), Armadillo (2), Opossum (1)
SPS2b	4	Human (2), Macaque (1), Guinea Pig (1)

The selenoproteins with pseudogenes, the number of total pseudogenes identified in all mammals, and their occurrence in individual organisms are given.
doi:10.1371/journal.pone.0033066.t002

unlikely that this pseudogene is expressed. The Ka/Ks ratio (used as an indicator of the selective pressure) was 1.58 for this gene, which suggested the possibility of positive selection.

A similar case was observed with SelW. This pseudogene arose sometime after the split between marmoset and macaque, but before macaque split with subsequent primates. Consequently, this pseudogene was identified in macaque, gibbon (first exon of the pseudogene only), orangutan, chimpanzee and human. Several

features of this pseudogene are peculiar. First, while the potential protein sequences of pseudogenes were highly homologous to SelW, the gene structure was different. The pseudogene consisted of two coding exons whereas SelW had five coding exons. The first exon of the pseudogene covered most of the first three coding exons of SelW and the second exon the remainder of SelW. Further analysis suggested that this gene was subject to positive selection. In Figure 7B, highlighted in green, are residues

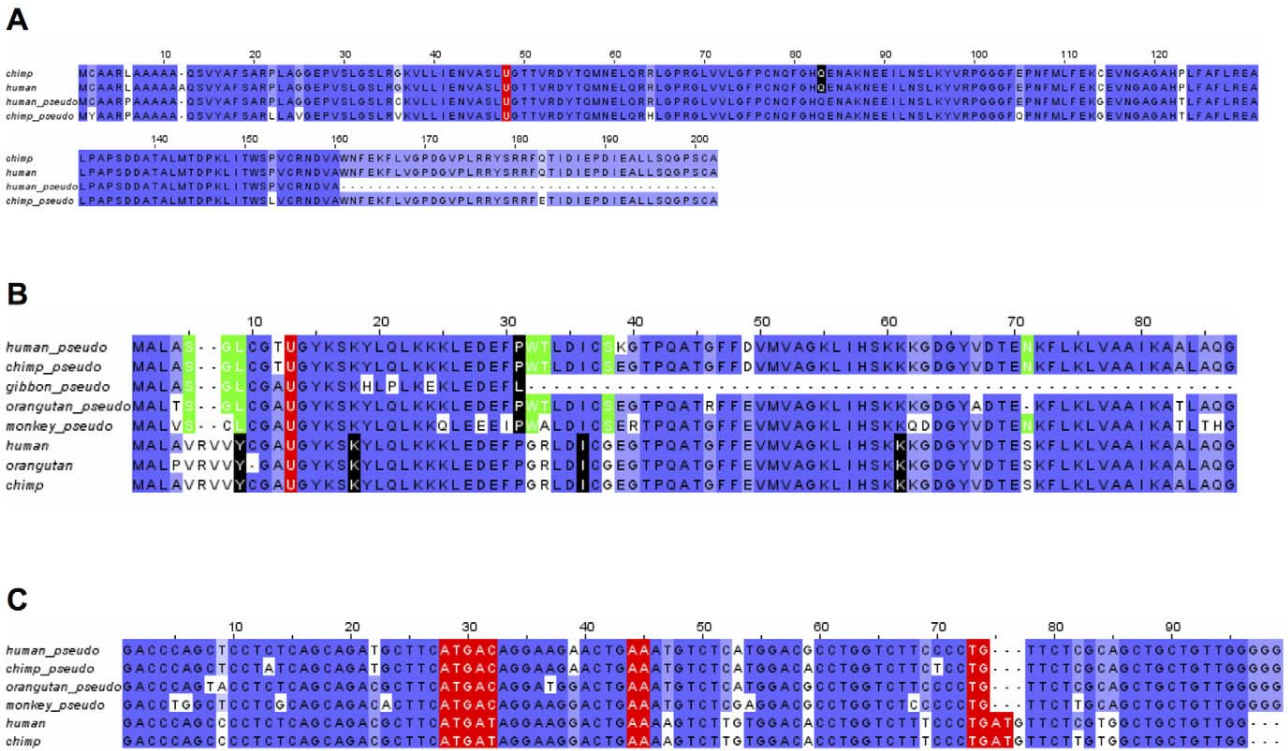


Figure 7. Multiple sequence alignment of selenoprotein genes and pseudogenes. A. GPx1. Multiple sequence alignment of human and chimpanzee GPx1 pseudogenes. B. SelW. The last residue of each exon is marked in black and Sec in red. Residues marked in green are described in the text. C. SECIS elements of SelW and SelW pseudogene.
doi:10.1371/journal.pone.0033066.g007

conserved in the pseudogenes but different in the corresponding positions in SelW. This situation occurred in 7 different positions in the pseudogenes. Similar to the GPx1 case above, this was indicative of constraint on the evolution of this gene. Furthermore, where nucleotide deletions occurred they happened in triplets thus preserving the reading frame. At positions 6 and 7 in the multiple sequence alignment, six bases were removed from the gene in all the pseudogenes. At position 71 in the alignment, the orangutan had a deleted amino acid, maintaining the reading frame. Additionally, the Ka/Ks ratio for human/maquette (the two most distant organisms in the dataset) was 0.59. Together, these data suggest that this pseudogene may have been under purifying selection for millions of years since it first appeared following the macaque/marmoset split.

However, further analysis suggested that this gene is not presently functional. First, the lack of ESTs in any of the representative organisms was inconsistent with this gene being expressed (or it only expresses in a very narrow niche). In addition, its SECIS element had a critical single base mutation. As discussed above, one of the salient features of SECIS elements is the GA/GA base-pairing in the stem loop necessary for binding of the proteins involved in Sec insertion [58]. An alignment of SECIS elements found in SelW and the pseudogenes (Figure 7C) showed that the pseudogene SECIS elements were missing the necessary GA sequence towards the end of the SECIS element. The SECIS elements from the pseudogenes still formed an appropriately shaped stem loop structure, but current evidence suggests that the missing GA should prevent Sec insertion. Together, this data suggests that while selection on this pseudogene has occurred, it is unlikely to be a currently active protein coding gene and that the in-frame UGA, if the gene was expressed, would result in early termination of translation.

Identification of alternative splicing forms

Alternative splicing has previously been reported for several selenoproteins, including TR1 [40–42,45–48], TR3 [43], Dio2 [68], Sep15 [67], and GPx4 [69,70]. We examined ESTs for all mammalian selenoproteins to characterize alternative splicing forms of these proteins. A challenge to identify splicing variants is the dependence on the quantity of EST data available. Only six mammals (human, mouse, rat, macaque, dog, and cow) had a sufficient number of ESTs to provide useful information. We found an association between the number of ESTs available for an organism and the number of identified variants (Supplementary Figure S41), suggesting that more variants may still be discovered as new sequences become available. In human, we found 17 selenoproteins to have alternative splicing isoforms. Supplementary Figure S42 shows the number of splicing forms identified for each of them. TR1 alternative splicing is discussed in detail above and is the most abundant of all mammalian selenoproteins, with at least 10 splicing forms. Three selenoproteins (SelT, Dio1, and TR3) had each 4 identified splicing isoforms.

Of note was a splice variant in the Sep15 gene. In this isoform, the entire 4th exon was removed during processing of the pre-mRNA (Figure 8), resulting in a frameshift in the next exon and premature stop codon. However, there was evidence that this isoform may be expressed, primarily from the high number of ESTs supporting this variant. In humans, 41 ESTs from 26 libraries, representing a variety of tissues, supported this variant. In total, there were only 99 ESTs for this portion of Sep15, so ~41% of ESTs for this region represented this variant. The variant was conserved in the mouse, where a single EST supported it. We expect that upcoming additional EST data for other mammals will confirm the presence of this isoform also in other species.

SelT is another selenoprotein for which we observed a previously unreported alternative isoform. This isoform contained an extra exon in the first intron. Multiple early stop codons and a frameshift were introduced by this new exon. We examined the new form for occurrence of an alternative translation start site, but no good candidates were found in this exon. Any transcript including this exon would thus code for a short protein which would be inactive. However, this form was supported by 13 human ESTs from several libraries and tissues. 15% (13 of 88) of the ESTs from these libraries supported this alternative form.

Two additional isoforms were identified in SelO, both of which were conserved in mice, rats, and humans. In the first, the penultimate exon was included in the transcript. This version was supported by 1 EST in humans, 10 ESTs in rats, and more than 30 ESTs in mice. The first full codon in the intron region was a stop codon in all three organisms, so there was high conservation along the entire protein, and they all terminated at the same location. The second newly identified isoform in SelO was similar to the previous, except that in this case the last intron was included in the mature transcript. Again, it was conserved in humans (1 EST), mice (16 ESTs), and rats (6 ESTs). This variant resulted in a frameshift in humans and rats, but not mice. Termination occurred in a different location in each of the three organisms with only mice still predicted to code for Sec.

We also identified a new isoform in GPx4. This isoform was conserved in mice and cows with 4 ESTs and 29 ESTs, respectively. In this isoform, the last intron was included in the mature transcript. No frameshift occurred in either of the animals; however, in mice a premature stop codon was introduced while the cow sequence was predicted to terminate as usual. An interesting point to consider was that termination in mice, although premature, was not far from the stop codon of the major form and was far from the Sec, so perhaps this isoform could be functional.

MsrB1 is another selenoprotein with alternative splicing forms. In most mammals, MsrB1 had 4 exons. In mice two different splicing forms were identified: a 4 exon version and a 5 exon version. The 5 exon version contained an extra intron in the 3'-UTR, so the protein sequence was unchanged. EST data suggested that the forms are equally expressed. Rats appeared to have only the 5 exon version, whereas other mammals appeared to have only the 4 exon version. We recently experimentally verified the occurrence of these forms, both of which result in the expression of the active MsrB1 [71].

Lastly, an interesting transcript variant was identified in SelS in humans. The major form of SelS contained 6 exons, whereas the alternative version had 7. Similar to MsrB1, the alternative splicing modified only the 3'-UTR. In the major form, the 3'-UTR and the last coding portion of the gene were in exon 6. In the alternative version, most of the 3'-UTR of the major form was spliced out and an entirely different 3'-UTR further downstream was included in the transcript. The unique feature of this splicing variant is that the alternative form did not contain a SECIS element. This would result in a non-functional truncated protein as Sec cannot be inserted. However, the detection of 13 human ESTs from numerous libraries and tissues suggested that this variant does in fact exist. A similar case was found for GPx3, although with less EST support. A variant in the 3'-UTR of GPx3 was identified in humans, featuring an extra intron in the 3'-UTR which corresponds almost exactly to the SECIS element.

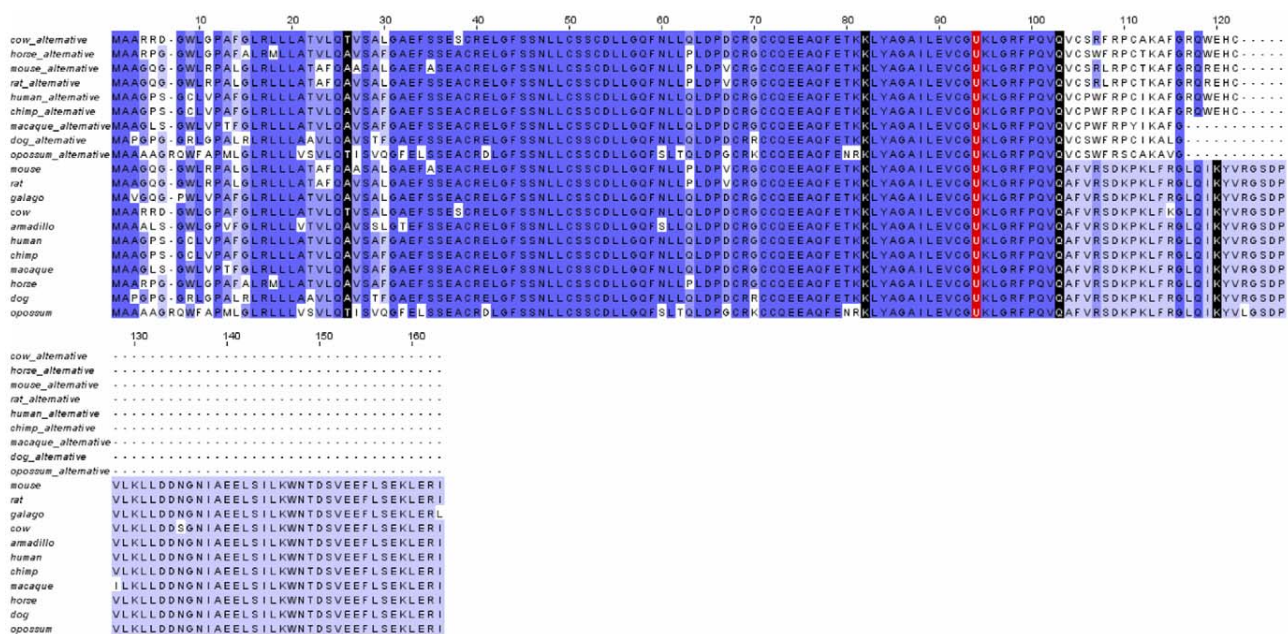


Figure 8. Multiple sequence alignment of Sep15 and a Sep15 alternative isoform. The last residue of each exon is marked in black and Sec in red. For human and mouse, ESTs support the presence of the isoform. For the other species shown, the protein sequences were predicted simulating skipping the 4th exon.

doi:10.1371/journal.pone.0033066.g008

Discussion

Although much effort has been devoted to identifying selenoprotein genes and characterizing Sec insertion machinery, evolution of the vertebrate selenoproteome is incompletely understood. Important insights concerning the vertebrate selenoproteomes and individual selenoproteins have previously been provided based on the analyses of a limited number of sequenced genomes [25,32]. In the present study, we scrutinized Sec- and Cys-containing homologs of known eukaryotic selenoprotein families in 44 vertebrate genomes, including 34 mammals. The number of organisms examined in this study should be considered sufficiently deep to identify the main themes in selenoprotein evolution. Although ongoing vertebrate genome projects will undoubtedly uncover various clade-specific features and allow refinements, the general features of the utilization and evolution of Se should not change. Across all vertebrates, a set of 45 selenoproteins was identified, with at most 38 represented in a single organism (zebrafish). 27 selenoproteins were found to be unique to vertebrates. 20 of them were generated through duplication of an existing selenoprotein in some vertebrate lineage, while 6 of them were part of the predicted ancestral selenoproteome. This implies that these latter 6 proteins (GPx2, Dio2, Dio3, SelI, SelPb, Fep15) were generated at the root of vertebrates. Individual mammalian selenoproteomes consist of 24/25 selenoproteins, from a set of 28. Our results reinforce the idea that the mammalian selenoproteome has remained relatively stable. However, a number of evolutionary events that changed its composition were observed (Figure 1): GPx6 and SelV were originated, SelPb was lost, SPS2b appeared and replaced SPS2a, SelV was lost in gorilla, and selenoproteins SelU1, Dio3 and GPx6 were converted to their Cys-containing forms in major or minor mammalian lineages.

The ancestral vertebrate selenoproteome was uncertain, as fish had many selenoproteins resulting of genome duplication and gene

duplication within bony fishes [25]. Previously, it has been suggested that the ancestral vertebrate selenoproteome consists of 31 selenoproteins: Dio1-3, GPx1-4, SelH, SelI, SelJ, SelK, SelL, SelM, SelN, SelO, SelP, SelPb, MsrB1, SelS, SelT1, SelU1-3, SelV, SelW1, SelW2a, Sep15, SPS2, TR1, TR3 and TGR [32]. In this study, we examined the occurrence of these selenoproteins in additional mammals and newly sequenced organisms which are important outgroups for understanding the evolution of different vertebrate clades (such as platypus and opossum). Particularly, we used both genomic and Trace databases for reconstruction of the selenoproteome of the phylogenetically oldest group of living jawed vertebrates, the elephant sharks. As a result, a number of new aspects were uncovered: (i) Fep15, which was previously thought to evolve in bony fish, was detected as a selenoprotein in elephant shark and as a Cys homolog in frog, and therefore should be viewed as part of the ancestral selenoproteome; (ii) TGR was found exclusively in tetrapods; (iii) SelV was found exclusively in placental mammals; (iv) phylogenetic analysis of Sec- and Cys-containing forms of the SelU family suggested that all Sec-containing SelU sequences belong to the SelU1 group (Figure 9). Mammals contain three Cys-containing SelU proteins (SelU1-3), whereas some fish (such as fugu and pufferfish) have three Sec-containing SelU proteins. It was previously thought that the three Cys-containing SelU proteins in mammals evolved from the three Sec-containing SelU sequences in fish. In this study, we could not find evidence that supports an early Sec-to-Cys conversion event for SelU2 and SelU3 proteins. Thus, the revised ancestral selenoproteome consists of the following 28 selenoproteins: GPx1-4, TR1, TR3, Dio1-3, SelH, SelI, SelJ, SelK, SelL, SelM, SelN, SelO, SelP, SelPb, MsrB1, SelS, SelT1, SelU1, SelW1, SelW2, Sep15, Fep15 and SPS2a (Figure 1).

Our analysis also uncovered the changes in the ancestral selenoproteome across vertebrates. Bony fishes were confirmed to be a lineage featuring several duplications. We predicted 14 in total: Dio3b, GPx1b, GPx3b, GPx4b, GPx4b2, MsrB1b, SelJ2,

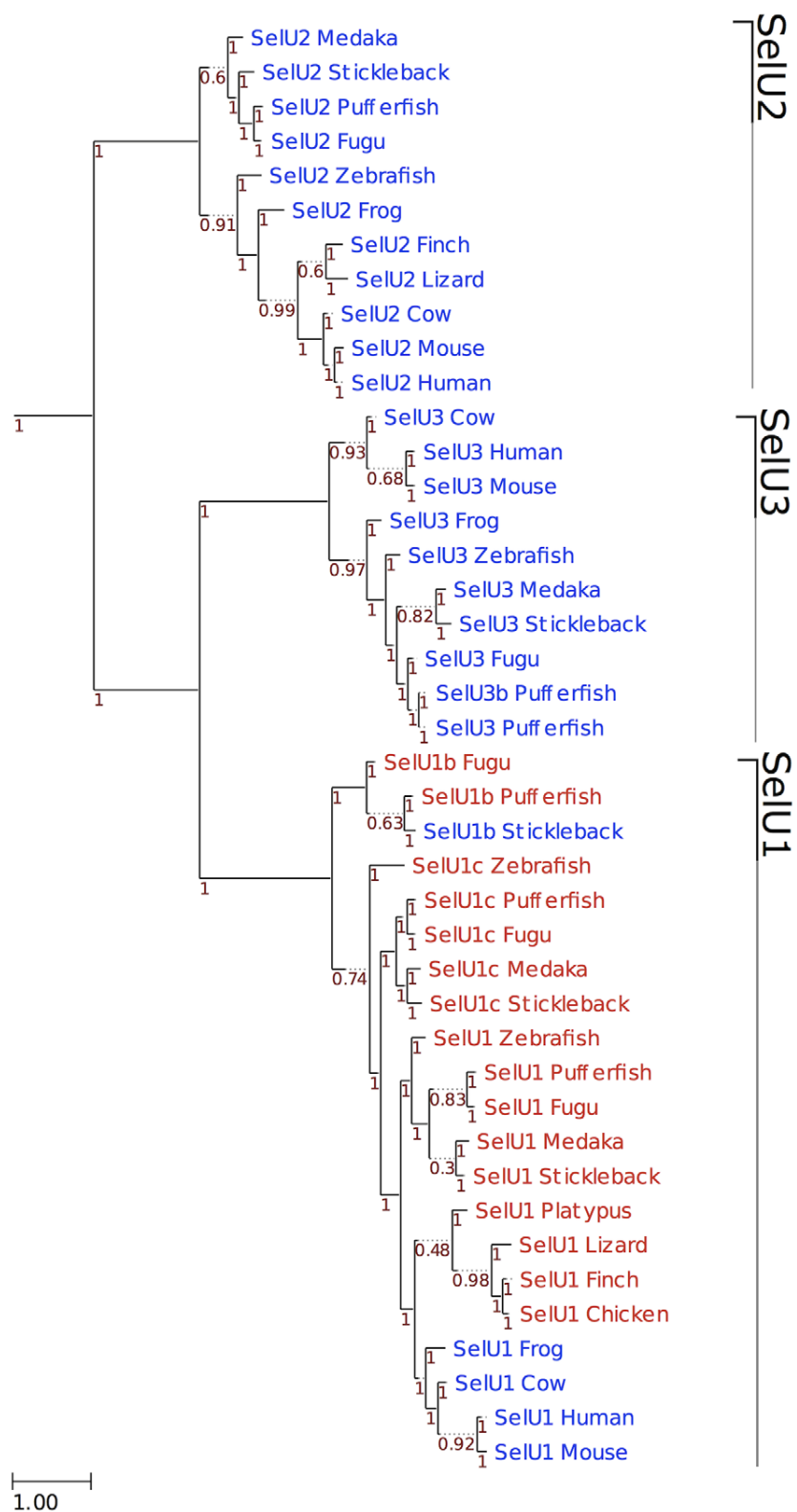


Figure 9. Phylogeny of SelU family in vertebrates. ML tree computed using the JTT substitution model. Sec-containing proteins are shown in red, whereas the Cys-containing homologs are shown in blue. At the bottom left, the distance scale in substitutions per position is shown. Branch support is shown along the tree in red.
doi:10.1371/journal.pone.0033066.g009

SeIO2, SeIT2, SeIT1b, SeIU1b, SeIU1c, SeIW2b and SeIW2c. Interestingly, we found 3 selenoprotein duplications specifically in zebrafish (SeIO2, SeIT1b, SeIW2b). As more fish sequences become available, further analysis will tell how common recent and lineage specific these duplications are. We also predicted all Sec to Cys conversions along the vertebrate tree, finding 12 such events. Particularly interesting was the case of GPx6, which was converted to the Cys form in at least 3 mammalian lineages independently. One of these events occurred in marmoset, a unique case among all 9 primates investigated. Notably, we observed proteins that do not bear Sec in any organism, but were generated through duplication of selenoprotein genes. In these cases (GPx5, GPx4b2, Rdx12), the conversion of the Sec TGA to a Cys codon must have happened early after the duplication, probably before the duplicated gene haplotype became fixed.

Comparative analyses of nucleotide and protein sequences of vertebrate selenoproteins revealed complex evolutionary histories in several families. First, SeIV most likely arose from duplication of SeIW in the ancestor of placental mammals, followed by addition of N-terminal sequences whose function is unclear as well as a deletion of a substantial portion of the 3'-UTR. Second, our analysis of GPx1-8 families highlighted three evolutionarily related groups: GPx1/GPx2, GPx3/GPx5/GPx6 and GPx4/GPx7/GPx8 (Figure 4). GPx4 appeared to be the most ancient GPx, whereas GPx5 and GPx6 were the most recently evolved GPx forms. Third, phylogenetic analyses of TR1 and TGR showed that these proteins evolved by gene duplication from an ancestral TR protein that is similar to a fish Grx-containing TR1. TR1 then suffered the loss of the Grx domain, except in some organisms (such as humans), which still retain it as an alternative isoform, whereas TGR acquired a new function (related to spermatogenesis) during evolution.

One of the most important features of selenoprotein genes is the SECIS element, which is located in the 3'-UTR. The most critical parts of the SECIS element are the SECIS core (located within the stem) and the two conserved nucleotides (of unknown function) in the apical loop. Within every examined SECIS element the GA/GA, paired non-canonically, were essential and conserved. Additionally, the two unpaired nucleotides within the apical loop are typically adenines; however, SECIS elements of SeIM and SeIO evolved cytosines in these positions specifically within mammals.

We also examined additional features of mammalian selenoprotein genes. First, we identified interesting pseudogenes of GPx1 and SeIW. These genes showed patterns of high conservation, including Ka/Ks values that may suggest active selective pressure. However, other characteristics indicate that they cannot code for functional proteins. Lack of EST data suggests that they are not (at least widely) expressed. It has been reported that quite few pseudogenes can go through the process of transcription in a tissue-specific manner [72]. They may play a role in regulation and expression of homologous genes or other genes [73–77]. Thus, it is possible that these pseudogenes may still be expressed in narrow niches to regulate the mRNA stability of SeIW or for other functions.

Second, we identified a number of alternative splicing forms for the majority of mammalian selenoproteins in different organisms that had not been previously reported. This data may provide new insights into the post-transcriptional regulation of selenoprotein genes in mammals. Many of the alternative transcripts reported here also possess features that suggest they cannot code a functional protein, particularly due to the presence of frameshifts. The evidence of transcription and the conservation in multiple species suggests nevertheless some biological role. The alternative

splicing forms that appeared to be conserved in multiple species (such as Sep15, SeIT, SeIO, GPx4 and MsrB1) represent top candidates for further experimental investigation.

Concluding, in this work we carried out comprehensive analyses of selenoproteomes in sequenced vertebrates to better define the roles of selenium and selenoproteins in these organisms. Our data provide a wealth of information on the composition and evolution of vertebrate and mammalian selenoproteomes. We revised the ancestral vertebrate selenoproteome and traced its evolution across all sequenced vertebrate lineages. This provided new insights into the evolution of selenoprotein families, in particular of glutathione peroxidases and thioredoxin reductases. Furthermore, we performed comparative analyses of gene structures and SECIS elements in mammalian selenoproteins, identified novel alternative splicing forms, and reported unusually conserved selenoprotein pseudogenes.

Materials and Methods

Genomic sequences and resources

All vertebrate genomes with significant sequence coverage from the current Entrez Genome Project at NCBI were used in this study (a total of 44 organisms). Additional databases that are related to each organism, such as Trace Archive database, EST database, non-redundant protein and nucleotide databases, were also retrieved from NCBI.

Identification and analyses of Sec/Cys-containing homologs, UTRs, SECIS elements, alternative splicing forms

We used several representative sequences of all known eukaryotic selenoproteins that were reported in previous studies as queries to search for Cys- and Sec-containing homologs in mammals and other vertebrates via BLAST with default parameters [78,79]. The automated predictions by program Selenoprofiles 2 [80] were also examined. For selenoprotein superfamilies (those including many genes sharing high homology, such as GPx and TR), the subfamilies were assigned based on the phylogenetic analysis. Gene losses were trusted only when observed in multiple species, or when both a high coverage genome assembly and abundant ESTs were available. The set of vertebrate-specific selenoproteins were determined by searching all ancestral selenoprotein sequences in the genomes of two chordate outgroups: amphioxus and sea squirt. For proteins not found in these species, additional searches were also performed in all non-vertebrate animal sequences. For selenoprotein superfamilies, a phylogenetic analysis of the non-vertebrate candidate sequences along with the vertebrate members was performed to assign subfamilies. UTRs were determined using EST data, and multiple sequence alignments were used to predict UTRs in animals with inadequate EST data. SECIS elements were predicted using SECISearch program [9]. Instances of alternative splicing were identified using BLAST search against EST data.

Multiple sequence alignment and phylogenetic analysis

Multiple sequence alignments were performed using ClustalW [81] and Mafft [82]. Phylogenetic reconstruction was performed as follows. ML trees were reconstructed using the best-fitting evolutionary model (BestML). To select the evolutionary model best fitting each protein family, a phylogenetic tree was reconstructed using a Neighbour Joining (NJ) approach as implemented in BioNJ [83]. The likelihood of this topology was computed, allowing branch-length optimization, using seven different models (JTT, LG, WAG, Blosum62, MtREV, VT and

Dayhoff), as implemented in PhyML version 3.0 [84]. The two evolutionary models best fitting the data were determined by comparing the likelihood of the used models according to the AIC criterion [85]. Then, ML trees were derived using these two models with the default tree topology search method NNI (Nearest Neighbor Interchange). A similar approach based on NJ topologies to select the best-fitting model for a subsequent ML analysis has been shown previously to be highly accurate [86]. Branch support was computed using an aLRT (approximate likelihood ratio test) parametric test based on a chi-square distribution, as implemented in PhyML. Finally, multiple sequence alignments were visualized with Jalview [87], and phylogenies with Ete2 [88].

Supporting Information

Figure S1 Multiple sequence alignment of Dio1. The approximate positions of introns are marked in black and the Sec is shown in red.
(TIFF)

Figure S2 Multiple sequence alignment of Dio2. Residues are marked as in Supplementary Figure S1.
(TIFF)

Figure S3 Multiple sequence alignment of Dio3. Residues are marked as in Supplementary Figure S1.
(TIFF)

Figure S4 Multiple sequence alignment of GPx1. Residues are marked as in Supplementary Figure S1.
(TIFF)

Figure S5 Multiple sequence alignment of GPx2. Residues are marked as in Supplementary Figure S1.
(TIFF)

Figure S6 Multiple sequence alignment of GPx3. Residues are marked as in Supplementary Figure S1.
(TIFF)

Figure S7 Multiple sequence alignment of GPx4. Residues are marked as in Supplementary Figure S1.
(TIFF)

Figure S8 Multiple sequence alignment of GPx6. Residues are marked as in Supplementary Figure S1.
(TIFF)

Figure S9 Multiple sequence alignment of MsrB1. Residues are marked as in Supplementary Figure S1.
(TIFF)

Figure S10 Multiple sequence alignment of SelH. Residues are marked as in Supplementary Figure S1.
(TIFF)

Figure S11 Multiple sequence alignment of SelI. Residues are marked as in Supplementary Figure S1.
(TIFF)

Figure S12 Multiple sequence alignment of SelK. Residues are marked as in Supplementary Figure S1.
(TIFF)

Figure S13 Multiple sequence alignment of SelM. Residues are marked as in Supplementary Figure S1.
(TIFF)

Figure S14 Multiple sequence alignment of SelN. Residues are marked as in Supplementary Figure S1.
(TIFF)

Figure S15 Multiple sequence alignment of SelO. Residues are marked as in Supplementary Figure S1.
(TIFF)

Figure S16 Multiple sequence alignment of SelP. Residues are marked as in Supplementary Figure S1. Note that there are multiple Sec in each protein.
(TIFF)

Figure S17 Multiple sequence alignment of SelPb. Residues are marked as in Supplementary Figure S1.
(TIFF)

Figure S18 Multiple sequence alignment of SelS. Residues are marked as in Supplementary Figure S1.
(TIFF)

Figure S19 Multiple sequence alignment of SelT. Residues are marked as in Supplementary Figure S1.
(TIFF)

Figure S20 Multiple sequence alignment of SelV. Residues are marked as in Supplementary Figure S1.
(TIFF)

Figure S21 Multiple sequence alignment of SelW and SelW2 proteins. Residues are marked as in Supplementary Figure S1.
(TIFF)

Figure S22 Multiple sequence alignment of Sep15. Residues are marked as in Supplementary Figure S1.
(TIFF)

Figure S23 Multiple sequence alignment of SPS2. Residues are marked as in Supplementary Figure S1. Note that in more ancient mammals and vertebrates the SPS2 gene is a multi-exon gene.
(TIFF)

Figure S24 Multiple sequence Alignment of TGR. Residues are marked as in Supplementary Figure S1.
(TIFF)

Figure S25 Multiple sequence alignment of TR1. Residues are marked as in Supplementary Figure S1.
(TIFF)

Figure S26 Multiple sequence alignment of TR3. Residues are marked as in Supplementary Figure S1.
(TIFF)

Figure S27 Multiple sequence alignment of SelL. The Sec is shown in red.
(TIFF)

Figure S28 Multiples sequence alignment of Fep15. Residues are marked as in Supplementary Figure S1.
(TIFF)

Figure S29 Multiple sequence alignment of SelJ. Residues are marked as in Supplementary Figure S1.
(TIFF)

Figure S30 Multiple sequence alignment of SelU1. Residues are marked as in Supplementary Figure S1.
(TIFF)

Figure S31 Multiple sequence alignment of mammalian SelT1 coding sequences. The last residue of each exon is marked in black, and the codon corresponding to the Sec is shown in red.
(TIFF)

Figure S32 SECIS in SPS2a and SPSb. Multiple sequence alignment of opossum SPS2a and SPS2b, platypus SPS2a, and human SPS2b SECIS elements. Critical portions are marked in red.
(TIFF)

Figure S33 Multiple sequence alignment of SelW and SelV 3'-UTRs. Critical portions of the SECIS elements are marked in red. The last nucleotide of exon 5 is marked in black.
(TIFF)

Figure S34 Phylogenetic tree of SelW1, SelW2 and SelV proteins. ML phylogenetic tree of SelW1, SelW2 and SelV protein sequences computed using the WAG substitution model. The branch support for each node (computed as described in the methods) is shown in red. The bar at the bottom left shows the scale in substitutions per position. The Rdx12 gene was found in all tetrapodes but only frog, mouse and human were included in the phylogenetic tree. In contrast all SelW2 detected were included: this gene is missing in all tetrapodes apart from frog. SelW1 is missing from bony fishes apart from zebrafish. SelV was detected in all placentals except gorilla but only rat, cow and human were included. Note that while the SelV-SelW1 duplication is clear and well supported, the rest of the tree is more confused. Nonetheless SelW2c, SelW2b and Rdx12 appear to have been generated by independent duplications.
(TIFF)

Figure S35 Multiple sequence alignment of TGR, zebrafish TR1, and GRx-containing TR1. Residues are marked as in Figure S1. Note positions where zebrafish TR1 and TGR match, but are different than GRx-containing TR1 (i.e., positions 43, 142, 143, 149, 150, 324, etc.).
(TIFF)

Figure S36 Multiple sequence alignment of extended Dio2 sequences. The last residue of each exon is marked in black and the Sec residues in red. The second Sec, residue 269, is the stop codon or potential second Sec.
(TIFF)

Figure S37 Multiple sequence alignment of SelI and its sequence homologs. Homologs are labeled with the annotated name. Some sequences not annotated as CHPT1 or CEPT1 were also included, as they contain the same domain. Important residues in the active sites are marked in red. The last residue of each side of all predicted transmembrane regions are marked in green. Selenocysteines are marked in red. The cysteines emerged

specifically in SelI proteins are also marked: the best candidate (near the catalytic side, on the same side of membrane of Sec) is marked in orange, while the other 2 cysteines are marked in yellow.
(TIFF)

Figure S38 Phylogenetic tree of SelI and its sequence homologs. ML tree computed using WAG model and the alignment shown in supplementary figure S37. The bar at the bottom left shows the scale in substitutions per position, while the branch support for each node is shown in red.
(TIFF)

Figure S39 5'-UTR lengths. 5'-UTR lengths are shown for various mammals. No length means there was insufficient EST data to define the 5'-UTR.
(TIFF)

Figure S40 3'-UTR lengths. 3'-UTR lengths are reported for various mammals. No length means there was insufficient EST data to define the 3'-UTR.
(TIFF)

Figure S41 Number of ESTs versus number of splicing forms. Left axis corresponds to the number of available ESTs (millions) and the right axis to the number of identified splicing forms identified in the listed animals.
(TIFF)

Figure S42 Splicing forms per selenoprotein. The numbers of splicing forms for each of 25 selenoproteins in mammals are shown.
(TIFF)

Table S1 Scientific names of species investigated in this study.
(DOCX)

Acknowledgments

We thank Salvador Capella-Gutiérrez and Toni Gabaldón for their phylogeny reconstruction pipeline.

Author Contributions

Conceived and designed the experiments: MM PGR YZ VNG. Performed the experiments: MM PGR YZ AVL. Analyzed the data: MM PGR YZ AVL THP RG DLH VNG. Wrote the paper: MM PGR YZ VNG.

References

- Böck A, Forchhammer K, Heider J, Leinfelder W, Sawers G, et al. (1991) Selenocysteine: the 21st amino acid. *Mol Microbiol* 5: 515–520.
- Stadtman TC (1996) Selenocysteine. *Annu Rev Biochem* 65: 83–100.
- Hatfield DL, Gladyshev VN (2002) How selenium has altered our understanding of the genetic code. *Mol Cell Biol* 22: 3565–3576.
- Berry MJ, Banu L, Chen YY, Mandel SJ, Kieffer JD, et al. (1991) Recognition of UGA as a selenocysteine codon in type I deiodinase requires sequences in the 3' untranslated region. *Nature* 353: 273–276.
- Low SC, Berry MJ (1996) Knowing when to stop: selenocysteine incorporation in eukaryotes. *Trends Biochem Sci* 21: 203–208.
- Rother M, Resch A, Gardner WL, Whitman WB, Böck A (2001) Heterologous expression of archaeal selenoprotein genes directed by the SECIS element located in the 3' non-translated region. *Mol Microbiol* 40: 900–908.
- Suppmann S, Persson BC, Böck A (1999) Dynamics and efficiency in vivo of UGA-directed selenocysteine insertion at the ribosome. *EMBO J* 18: 2284–2293.
- Tujebajeva RM, Copeland PR, Xu XM, Carlson BA, Harney JW, et al. (2000) Decoding apparatus for eukaryotic selenocysteine insertion. *EMBO Rep* 1: 158–163.
- Kryukov GV, Castellano S, Novoselov SV, Lobanov AV, Zehrab O, et al. (2003) Characterization of mammalian selenoproteomes. *Science* 300: 1439–1443.
- Yant LJ, Ran Q, Rao L, Van Remmen H, Shibata T, et al. (2003) The selenoprotein GPX4 is essential for mouse development and protects from radiation and oxidative damage insults. *Free Radic Biol Med* 34: 496–502.
- Conrad M, Jakupoglu C, Moreno SG, Lippl S, Banjac A, et al. (2004) Essential role for mitochondrial thioredoxin reductase in hematopoiesis, heart development, and heart function. *Mol Cell Biol* 24: 9414–9423.
- Jakupoglu C, Przemeck GK, Schneider M, Moreno SG, Mayr N, et al. (2005) Cytoplasmic thioredoxin reductase is essential for embryogenesis but dispensable for cardiac development. *Mol Cell Biol* 25: 1980–1988.
- Matsui M, Oshima H, Oshima H, Takaku K, Maruyama T, et al. (1996) Early embryonic lethality caused by targeted disruption of the mouse thioredoxin gene. *Dev Biol* 178: 179–185.
- Olson GE, Winfrey VP, Nagdas SK, Hill KE, Burk RF (2005) Selenoprotein P is required for mouse sperm development. *Biol Reprod* 73: 201–211.
- Peters MM, Hill KE, Burk RF, Weeber EJ (2006) Altered hippocampus synaptic function in selenoprotein P deficient mice. *Mol Neurodegener* 1: 12.
- Fomenko DE, Novoselov SV, Natarajan SK, Lee BC, Koc A, et al. (2009) MsrB1 (methionine-R-sulfoxide reductase 1) knock-out mice: roles of MsrB1 in redox regulation and identification of a novel selenoprotein form. *J Biol Chem* 284: 5986–5993.

17. Arnér ES, Holmgren A (2006) The thioredoxin system in cancer. *Semin Cancer Biol* 16: 420–426.
18. Hatfield DL, Carlson BA, Xu XM, Mix H, Gladyshev VN (2006) Selenocysteine incorporation machinery and the role of selenoproteins in development and health. *Prog Nucleic Acid Res Mol Biol* 81: 97–142.
19. Papp LV, Lu J, Holmgren A, Khanna KK (2007) From selenium to selenoproteins: synthesis, identity, and their role in human health. *Antioxid Redox Signal* 9: 775–806.
20. Koc A, Gladyshev VN (2007) Methionine sulfoxide reduction and the aging process. *Ann N Y Acad Sci* 1100: 383–386.
21. Brigelius-Flohé R (2008) Selenium compounds and selenoproteins in cancer. *Chem Biodivers* 5: 389–395.
22. Carlson BA, Xu XM, Gladyshev VN, Hatfield DL (2005) Selective rescue of selenoprotein expression in mice lacking a highly specialized methyl group in selenocysteine tRNA. *J Biol Chem* 280: 5542–5548.
23. Lobanov AV, Hatfield DL, Gladyshev VN (2009) Eukaryotic selenoproteins and selenoproteomes. *Biochim Biophys Acta* 1790: 1424–8.
24. Lobanov AV, Fomenko DE, Zhang Y, Sengupta A, Hatfield DL, et al. (2007) Evolutionary dynamics of eukaryotic selenoproteomes: large selenoproteomes may associate with aquatic life and small with terrestrial life. *Genome Biol* 8: R198.
25. Lobanov AV, Hatfield DL, Gladyshev VN (2008) Reduced reliance on the trace element selenium during evolution of mammals. *Genome Biol* 9: R62.
26. Gladyshev VN, Kryukov GV (2001) Evolution of selenocysteine-containing proteins: significance of identification and functional characterization of selenoproteins. *Biofactors* 14: 87–92.
27. Novoselov SV, Gladyshev VN (2003) Non-animal origin of animal thioredoxin reductases: implications for selenocysteine evolution and evolution of protein function through carboxy-terminal extensions. *Protein Sci* 12: 372–378.
28. Castellano S, Novoselov SV, Kryukov GV, Lescure A, Bianco E, et al. (2004) Reconsidering the evolution of eukaryotic selenoproteins: a novel nonmammalian family with scattered phylogenetic distribution. *EMBO Rep* 5: 71–77.
29. Stillwell RJ, Berry MJ (2005) Expanding the repertoire of the eukaryotic selenoproteome. *Proc Natl Acad Sci U S A* 102: 16123–16124.
30. Zhang Y, Romero H, Salinas G, Gladyshev VN (2006) Dynamic evolution of selenocysteine utilization in bacteria: a balance between selenoprotein loss and evolution of selenocysteine from redox active cysteine residues. *Genome Biol* 7: R94.
31. Toppo S, Vanin S, Bosello V, Tosatto SC (2008) Evolutionary and structural insights into the multifaceted glutathione peroxidase (gpx) superfamily. *Antioxid Redox Signal* 10: 1501–1514.
32. Castellano S, Andrés AM, Bosch E, Bayes M, Guigó R, et al. (2009) Low exchangeability of selenocysteine, the 21st amino acid, in vertebrate proteins. *Mol Biol Evol* 26: 2031–2040.
33. Taylor JS, Braasch I, Frickey T, Meyer A, Van de Peer Y (2003) Genome duplication, a trait shared by 22000 species of ray-finned fish. *Genome Res* 13: 382–390.
34. Novoselov SV, Hua D, Lobanov AV, Gladyshev VN (2006) Identification and characterization of Fep15, a new selenocysteine-containing member of the Sep15 protein family. *Biochem J* 394: 575–9.
35. Ferguson-Smith MA, Rens W (2010) The unique sex chromosome system in platypus and echidna. *Genetika* 46: 1314–9.
36. Schomburg L, Schweizer U (2009) Hierarchical regulation of selenoprotein expression and sex-specific effects of selenium. *Biochim Biophys Acta* 1790: 1453–62.
37. Khil PP, Oliver B, Camerini-Otero RD (2005) X for intersection: retrotransposition both on and off the X chromosome is more frequent. *Trends Genet* 21: 3–7.
38. Ventura M, Catacchio CR, Alkan C, Marques-Bonet T, Sajjadian S, et al. (2011) Gorilla genome structural variation reveals evolutionary parallelisms with chimpanzee. *Genome Res* 21: 1640–9.
39. Dikiy A, Novoselov SV, Fomenko DE, Sengupta A, Carlson BA, et al. (2007) SelT, SelW, SelH, and Rdx12: genomics and molecular insights into the functions of selenoproteins of a novel thioredoxin-like family. *Biochemistry* 46: 6871–6882.
40. Su D, Gladyshev VN (2004) Alternative splicing involving the thioredoxin reductase module in mammals: a glutaredoxin-containing thioredoxin reductase 1. *Biochemistry* 43: 12177–12188.
41. Rundlöf AK, Carlsten M, Giacobini MM, Arnér ES (J) Prominent expression of the selenoprotein thioredoxin reductase in the medullary rays of the rat kidney and thioredoxin reductases mRNA variants differing at the 5' untranslated region. *Biochem* 2000, 347: 661–668.
42. Rundlöf AK, Janard M, Miranda-Vizuete A, Arnér ES (2004) Evidence for intriguingly complex transcription of human thioredoxin reductase 1. *Free Radic Biol Med* 36: 641–656.
43. Turanov AA, Su D, Gladyshev VN (2006) Characterization of alternative cytosolic forms and cellular targets of mouse mitochondrial thioredoxin reductase. *J Biol Chem* 281: 22953–22963.
44. Gerashchenko MV, Su D, Gladyshev VN (2009) CUG start codon generates thioredoxin/glutathione reductase isoforms in mouse testes. *J Biol Chem* 285: 4595–4602.
45. Damdimopoulou PE, Miranda-Vizuete A, Arnér ES, Gustafsson JA, Damdimopoulos AE (2009) The human thioredoxin reductase-1 splice variant TXNRD1_v3 is an atypical inducer of cytoplasmic filaments and cell membrane filopodia. *Biochim Biophys Acta* 1793: 1588–1596.
46. Dammeyer P, Damdimopoulos AE, Nordman T, Jiménez A, Miranda-Vizuete A, et al. (2008) Induction of cell membrane protrusions by the N-terminal glutaredoxin domain of a rare splice variant of human thioredoxin reductase 1. *J Biol Chem* 283: 2814–2821.
47. Arnér ES (2009) Focus on mammalian thioredoxin reductases—important selenoproteins with versatile functions. *Biochim Biophys Acta* 1790: 495–526.
48. Selenius M, Rundlöf AK, Olm E, Fernande AP, Björnstedt M (2010) Selenium and the selenoprotein thioredoxin reductase in the prevention, treatment and diagnostics of cancer. *Antioxid Redox Signal* 12: 867–880.
49. Bianco AC, Larsen PR (2006) Selenium, deiodinases and endocrine function. In: Hatfield DL, Berry MJ, Gladyshev VN, eds. *Selenium: Its molecular biology and role in human health* Springer. pp 207–219.
50. Bianco AC, Salvatore D, Gereben B, Berry MJ, Larsen PR (2002) Biochemistry, cellular and molecular biology, and physiological roles of the iodothyronine selenodeiodinases. *Endocr Rev* 23: 38–89.
51. Salvatore D, Harney JW, Larsen PR (1999) Mutation of the Secys residue 266 in human type 2 selenodeiodinase alters 75Se incorporation without affecting its biochemical properties. *Biochimie* 81: 535–538.
52. Henneberry AL, McMaster CR (1999) Cloning and expression of a human choline/ethanolaminephosphotransferase: synthesis of phosphatidylcholine and phosphatidylethanolamine. *Biochem J* 339: 291–298.
53. Horibata Y, Hirabayashi Y (2007) Identification and characterization of human ethanolaminephosphotransferase 1. *J Lipid Res* 48: 503–8.
54. Lothrop AP, Ruggles EL, Hondal RJ (2009) No selenium required: reactions catalyzed by mammalian thioredoxin reductase that are independent of a selenocysteine residue. *Biochemistry* 48: 6213–23.
55. Panopoulou G, Hennig S, Groth D, Krause A, Poustka AJ, et al. (2003) New evidence for genome-wide duplications at the origin of vertebrates using an amphioxus gene set and completed animal genomes. *Genome Res* 13: 1056–66.
56. Mignone F, Gissi C, Liuni S, Pesole G (2002) Untranslated regions of mRNAs. *Genome Biol* 3: REVIEWS0004.
57. Berry MJ, Banu L, Harney JW, Larsen PR (1993) Functional characterization of the eukaryotic SECIS elements which direct selenocysteine insertion at UGA codons. *EMBO J* 12: 3315–3322.
58. Krol A (2002) Evolutionarily different RNA motifs and RNA-protein complexes to achieve selenoprotein synthesis. *Biochimie* 84: 765–774.
59. Shchedrina VA, Novoselov SV, Malinouski MY, Gladyshev VN (2007) Identification and characterization of a selenoproteins family containing a diselenide bond in a redox motif. *Proc Natl Acad Sci U S A* 104: 13919–13924.
60. Martin GW, 3rd, Harney JW, Berry MJ (1996) Selenocysteine incorporation in eukaryotes: insights into mechanism and efficiency from sequence, structure, and spacing proximity studies of the type 1 deiodinase SECIS element. *RNA* 2: 171–182.
61. Kollmus H, Flohé L, McCarthy JE (1996) Analysis of eukaryotic mRNA structures directing cotranslational incorporation of selenocysteine. *Nucleic Acids Res* 24: 1195–1201.
62. Shen Q, Chu FF, Newburger PE (1993) Sequences in the 3'-untranslated region of the human cellular glutathione peroxidase gene are necessary and sufficient for selenocysteine incorporation at the UGA codon. *J Biol Chem* 268: 11463–11469.
63. Diamond AM, Cruz R, Bencsics C, Hatfield D (1992) A pseudogene for human glutathione peroxidase. *Gene* 122: 377–380.
64. Bellingham J, Gregory-Evans K, Fox MF, Gregory-Evans CY (2003) Gene structure and tissue expression of human selenoprotein W, SEPWL, and identification of a retroprocessed pseudogene, SEPWLIP. *Biochim Biophys Acta* 1627: 140–146.
65. Boschan C, Borchert A, Ufer C, Thiele BJ, Kuhn H (2002) Discovery of a functional retrotransposon of the murine phospholipid hydroperoxide glutathione peroxidase: chromosomal localization and tissue-specific expression pattern. *Genomics* 79: 383–394.
66. Winkler K, Brigelius-Flohé R (1999) Gastrointestinal glutathione peroxidase. *Biofactors* 10: 245–249.
67. Kumaraswamy E, Malykh A, Korotkov KV, Kozyavkin S, Hu Y, et al. (2000) Structure-expression relationships of the 15-kDa selenoprotein gene. Possible role of the protein in cancer etiology. *J Biol Chem* 275: 35540–35547.
68. Ohba K, Yoshioka T, Muraki T (2001) Identification of two novel splicing variants of human type II iodothyronine deiodinase mRNA. *Mol Cell Endocrinol* 172: 168–175.
69. Maiorino M, Scapin M, Ursini F, Biasolo M, Bosello V, et al. (2003) Distinct promoters determine alternative transcription of gpx-4 into phospholipid-hydroperoxide glutathione peroxidase variants. *J Biol Chem* 278: 34286–34290.
70. Moreno SG, Laux G, Brielmeyer M, Bornkamm GW, Conrad M (2003) Testis-specific expression of the nuclear form of phospholipid hydroperoxide glutathione peroxidase (PHGPx). *Biol Chem* 384: 635–643.
71. Liang X, Fomenko DE, Hua D, Kaya A, Gladyshev VN (2010) Diversity of protein and mRNA forms of mammalian methionine sulfoxide reductase B1 due to intronization and protein processing. *PLoS One* 5: e11497.
72. Zheng D, Frankish A, Baertsch R, Kapranov P, Reymond A, et al. (2007) Pseudogenes in the ENCODE regions: consensus annotation, analysis of transcription, and evolution. *Genome Res* 17: 839–851.

73. Hirotsune S, Yoshida N, Chen A, Garrett L, Sugiyama F, et al. (2003) An expressed pseudogene regulates the messenger-RNA stability of its homologous coding gene. *Nature* 423: 91–96.
74. Balakirev ES, Ayala FJ (2003) Pseudogenes: are they “junk” or functional DNA. *Annu Rev Genet* 37: 123–151.
75. Svensson O, Arvestad L, Lagergren J (2006) Genome-wide survey for biologically functional pseudogenes. *PLoS Comput Biol* 2: e46.
76. Tam OH, Aravin AA, Stein P, Girard A, Murchison EP, et al. (2008) Pseudogene-derived small interfering RNAs regulate gene expression in mouse oocytes. *Nature* 453: 534–538.
77. Polisenio L, Salmena L, Zhang J, Carver B, Haveman WJ, et al. (2010) A coding-independent function of gene and pseudogene mRNAs regulates tumour biology. *Nature* 465: 1033–1038.
78. Altschul SF, Mapped TL, Schaffer AA, Zhang J, Zhang Z, et al. (1997) (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res* 25: 3389–3402.
79. Zhang Z, Schwartz S, Wagner L, Miller W (2000) A greedy algorithm for aligning DNA sequences. *J Comput Biol* 7: 203–214.
80. Mariotti M, Guigó R (2010) Selenoprofiles: profile-based scanning of eukaryotic genome sequences for selenoprotein genes. *Bioinformatics* 26(21): 2656–63.
81. Thompson JD, Higgins DG, Gibson TJ (1994) CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res* 22: 4673–4680.
82. Katoh K, Toh H (2008) Recent developments in the MAFFT multiple sequence alignment program. *Brief Bioinform* 9: 286–98.
83. Gascuel O (1997) BIONJ: an improved version of the NJ algorithm based on a simple model of sequence data. *Mol Biol Evol* 14: 685–95.
84. Guindon S, Dufayard JF, Lefort V, Anisimova M, Hordijk W, et al. (2010) New algorithms and methods to estimate maximum-likelihood phylogenies: assessing the performance of PhyML 3.0. *Syst Biol* 59: 307–21.
85. Akaike H (1973) Information theory and extension of the maximum likelihood principle. *Proceedings of the 2nd international symposium on information theory, Budapest, Hungary*. pp 267–281.
86. Huerta-Cepas J, Capella-Gutierrez S, Pryszcz LP, Denisov I, Kormes D, et al. (2011) PhylomeDB v3.0: an expanding repository of genome-wide collections of trees, alignments and phylogeny-based orthology and paralogy predictions. *Nucleic Acids Res* 39: D556–60.
87. Clamp M, Cuff J, Searle SM, Barton GJ (2004) The Jalview Java alignment editor. *Bioinformatics* 20: 426–427.
88. Huerta-Cepas J, Dopazo J, Gabaldón T (2010) ETE: a python Environment for Tree Exploration. *BMC Bioinformatics* 11: 24.

3.4 Selenoprotein extinctions in insects

3.4.1 The known Sec extinction in *D. willistoni*

In the work of [Chapple and Guigó, 2008], our group described how several insects lost selenoproteins along with the ability to make selenocysteine. Sec extinctions happened in parallel in insect lineages, and occurred at different times. In the past years we investigated the mechanisms of this process, trying to answer questions such as, in what order events take place? is the first event a mutation in a specific Sec machinery gene, or do all selenoproteins have to be converted to Cys-homologues first? We chose to investigate the most recent (thus hopefully most insightful) Sec extinction known: that of *Drosophila willistoni*. This species is estimated to have diverged from the rest of sequenced drosophila about 35 million years ago. A few features set it apart from the other drosophila: a lower genomic GC content, a lower codon bias in coding sequences (favoring AT nucleotides) [Powell et al., 2003; Drosophila-Consortium, 2007], and a peculiar genomic fusion of Muller Elements E and F (in *D. melanogaster*, chromosomes 3R and 4). The F element (also known as dot chromosome), is very small in *D. melanogaster* (4.2 Mb) and possesses a very peculiar chromatin structure, showing characteristics of both euchromatin and heterochromatin [Riddle et al., 2009]. Interestingly, the F element exhibits very low level of recombination, and also GC content and codon usage more prone to AT than the rest of the genome [Vicario et al., 2007].

3.4.2 Sampling selenoproteins in drosophila by degenerate PCR

In collaboration with the group of Montserrat Corominas at the Universitat de Barcelona (particularly Andrea Mateo), we attempted to widen the spotlight around *D. willistoni*, trying to map more precisely its Sec loss. We exploited the availability of large stock centers (for example <https://stockcenter.ucsd.edu/info/welcome.php>) from which researchers can order a variety of drosophila species, and have them sent to their lab. We ordered many, focusing on those phylogenetically closest to *D. willistoni*. We then designed degenerate PCR primers for the three drosophila selenoprotein genes, considering the variation observed in the 12 public genomes.

We sampled 23 species in this way (see table 3.2), although with many gaps due to experimental uncertainties. Investigated species belonged to three groups, with unresolved phylogenetic topology: Willistoni (including *D. willistoni*), Obscura and Saltans. From PCR results, all species from the Willistoni group appeared to lack SPS2, and have cysteine homologues for SelH and SelK. Thus we can map their Sec extinction before their split. On the contrary, all Obscura species appeared to have the same selenoproteome of *D. melanogaster*. The Saltans group was the most interesting, for we found a cysteine conversion in SelH of *D. neocordata*. Also, SPS2 and SelK were not detected in any species, although we detected some intact SelH selenoproteins. We thought that this group may contain both species with and without selenoproteins. To follow this up, we decided to apply the recent

group	species	SelH	SelK	SPS2
Willistoni	willistoni	TGT	TGT	X
	capricornis	?	?	X
	tropicalis	TGT	TGT	X
	equinoxalis	?	TGT	X
	nebulosa	TGT	X	X
	fumipennis	?	?	X
	sucina	?	?	X
Obscura	ambigua	TGA	TGA	TGA
	bifasciata	TGA	TGA	TGA
	guanche	?	TGA	?
	subobscura	?	TGA	TGA
	azteca	TGA	TGA	TGA
	affinis	TGA	?	?
	algorquin	TGA	TGA	TGA
	miranda	?	TGA	TGA
Saltans	austrosaltans	TGA	?	X
	lusaltans	?	?	?
	prosaltans	TGA	?	X
	saltans	TGA	?	?
	milleri	?	?	X
	sturtevanti	?	?	X
	emarginata	?	?	X
	neocordata	TGC	?	X

Table 3.2: Selenoprotein genes sampled in drosophila species with degenerate primers (by Andrea Mateo). The codon found at the Sec position is shown, colored after the amino acid coded (green for selenocysteine, red for cysteine). A red cross means the gene was called absent. A question mark means no call.

advances in sequencing technologies, and get the full genome of the 8 available species in the Saltans group.

3.4.3 Genome sequencing of 8 drosophila from the Saltans group

Genome sequencing was performed at the Ultrasequencing facility in our institute (<http://seq.crg.es/Home/WebHome>). A few rounds of sequencing using different technologies and strategies were applied, finally resulting in the data presented in table 3.3.

To produce genome assemblies from these sets of reads, the program SOAP-denovo was used [Luo et al., 2012], using various options and inspecting results. This work was carried out by Manuela Hummel at the CRG Ultrasequencing facility. We analyzed the resulting assemblies comparing them with the other available drosophila genomes (see table 3.4).

	GA IIX, paired-end	HiSeq, paired-end	HiSeq, mate-pair	Total n. reads
D.austrosaltans	2 lanes	1 lane	1 lane	449,201,768
D.emarginata	1 lane	2 lanes	1 lane	577,676,568
D.lusaltans	1 lane	1 lane	1 lane	313,710,006
D.milleri	1 lane	1 lane	1 lane	289,392,052
D.neocordata	1 lane	2.5 lanes*	1 lane	368,712,938
D.prosaltans	1 lane	1 lane	1 lane	334,461,414
D.saltans	6 lanes		2 lanes	467,718,816
D.sturtevantii	1 lane	1 lane	1 lane	260,774,246

Table 3.3: Summary of genome sequencing resources for 8 drosophila from Saltans group. Three technologies were used. GAIIX paired-end= Illumina Genome Analyzer IIX, including 2 paired reads of 76 bp per lane, with insert size of 350/400 bp. HiSeq, paired-end = Illumina HiSeq2000, including 2 paired reads of 100 bp per lane, with insert size of 350/400 bp. HiSeq, mate-pair = Illumina HiSeq2000, including 2 paired reads of 50 bp per lane, with insert size of 3000 bp for *D.milleri*, *D.sturtevantii*, *D.neocordata*, and of 4500-5000 bp for the rest of species. In one lane of *D.neocordata* HiSeq paired-end run (*), the procedure on one of two paired reads failed, halving the amount of output reads in that lane.

set	Species	N. scaffolds	N50	after filtering contigs	
				N. scaffolds	N50
Saltans	D.austrosaltans	184,967	103,546	22,301	117,679
	D.emarginata	497,369	7,555	39,637	14,080
	D.lusaltans	505,447	4,104	63,835	6,401
	D.milleri	162,202	204,707	6,968	241,683
	D.neocordata	414,856	100,293	10,581	158,467
	D.prosaltans	118,896	166,451	16,527	188,386
	D.saltans	87,945	254,311	17,171	287,761
	D.sturtevantii	296,041	12,844	31,352	23,841
reference 12	D.persimilis	12,838	1,869,541		
	D.pseudoobscura	4,896	12,523,060		
	D.mojavensis	6,841	24,764,193		
Baylor's	D.rhopaloea	23,004	44,904		
	D.ficusphila	5,785	1,050,437		
	D.biarmipes	5,640	3,129,048		

Table 3.4: Statistics for Saltans genome assemblies. N50 = length for which the collection of all scaffolds of that length or longer contains at least half of the total of the lengths of the scaffolds. The worst and best Saltans assemblies for the N50 statistic are highlighted in green and red respectively. For comparison, we included also the worst, average and best genomes among the 12 reference drosophila [Drosophila-Consortium, 2007], and among those recently sequenced at Baylor's. We noticed that the vast majority of scaffolds in our genomes were very short. Thus we produced better versions of our genomes by filtering out all of those shorter than 500 bp, unless they carried some annotation. For this, we checked our full annotation of the genomes, presented later.

Our genomes have worse quality than the others available for drosophila (much worse than the reference 12), and we will need to take this into account for any analysis. The difference in quality is due mainly to the higher coverage of the other genomes, but also derives from genetic variation: for most of the other genomes, species were inbred for several generations to reduce heterozygosity before sequencing. This resulted in much better assemblies (Stephen Richards, personal communication). Despite our genomes may seem poor when compared to other drosophila, their quality was good enough for our purposes, as we will see in the next sections. Lately, we also produced RNAseq for *D.willistoni* and for 4 species belonging to the Saltans group: *D.sturtevanti*, *D.milleri*, *D.neocordata* and *D.saltans*. This data is used only marginally in this thesis, since we are still in the process of fully analyzing it.

3.4.4 Building a phylogenetic tree of all 29 sequenced drosophila

In order to make sense of our new genomes, it was necessary to place them phylogenetically in respect to each other, and to the rest of drosophila. This project was carried out in collaboration with the group of Toni Gabaldón at CRG, expert in phylogenetic reconstruction methods, and in particular with Salvador Capella-Gutiérrez. To infer the phylogenetic relationships among a set of species, normally researchers utilize sequence-based methods on protein coding genes, or sometimes on tRNAs. However, every gene has its own history, that may not reflect perfectly the species phylogeny. Moreover, the phylogenetic signal in sequences is inherently noisy, leading to different topologies depending on the method/model considered. When whole genomes are available, normally large sets of genes are used for phylogenetic reconstruction, concatenating sequences as if it there was a single, enormous protein. In this way, the noise across different genes is reduced, as the average gene history better approaches the species history. We decided to predict the phylogeny of all 29 drosophila with an available genome, including the first 12 reference species [Drosophila-Consortium, 2007], plus other 8 public genomes recently sequenced by Baylor’s College, plus *D.santomea* (sequenced at Princeton), plus our 8 species from Saltans group. Among all these, only the 12 reference and *D.santomea* have a clear phylogenetic tree annotated in flybase. The choice of genes used for reconstruction is known to have a deep effect on the resulting topology. We decided to predict ourselves a set of “core” proteins with clean phylogenetic signal in all genomes. We considered a starting set of 566 proteins, which in another study resulted to have clear 1-to-1 orthology in all 12 reference drosophila (see <http://www.phylomedb.org/> and [Huerta-Cepas et al., 2011]). We built as single sequence profile for each of them, and we searched them with selenoprofiles in all 29 genomes. We then applied the “best-bidirectional hit” criteria: for each *D.melanogaster* query protein, we asked what is the protein candidate best matching this sequence in (for example) the *D.saltans* genome. Then, we took this candidate and ran it with blastp against the full proteome of *D.melanogaster*. The best-bidirectional hit criteria is satisfied if the best blast hit is the same *D.melanogaster*

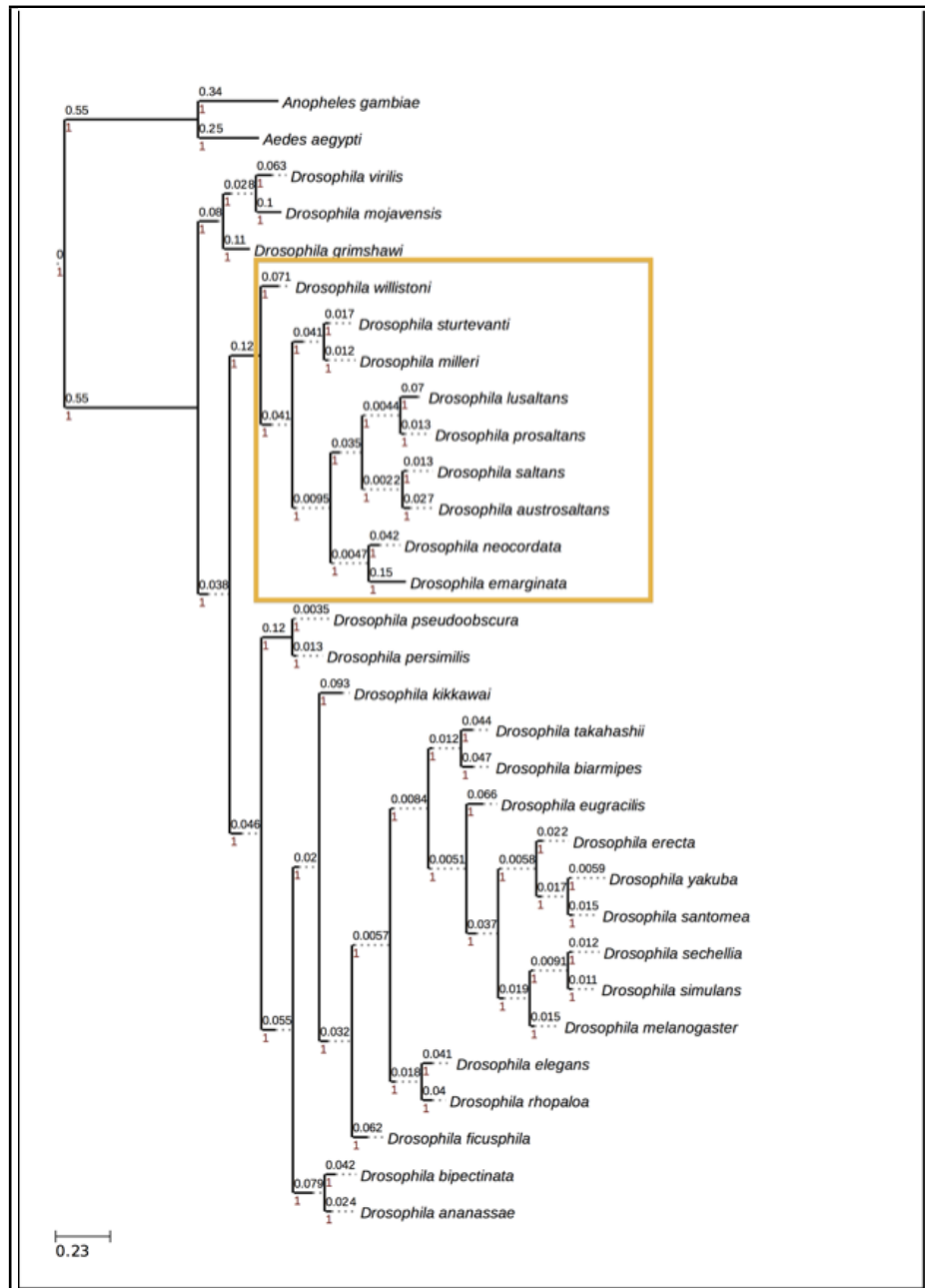


Figure 3.7: Phylogenetic tree of all 29 sequenced drosophila. The Willis-toni/Saltans group is framed in orange. The branch lengths (also reported in black numbers) reflect the distance between species, in number of substitutions per amino acid site in the concatenated alignment. The support for each node reported with red numbers, and was computed with aLRT (approximate likelihood ratio test) parametric test based on a chi-square distribution, as implemented in PhyML

query used to begin with. We filtered our protein set to keep only those for which this criteria was satisfied in at least 14 species, among the 17 with unknown phylogeny. We also applied additional filtering criteria to exclude recent pseudogenes to bias the analysis. This resulted in 455 protein groups that were concatenated and used for phylogenetic reconstruction, using the same procedure reported in [Mariotti et al., 2012], after [Huerta-Cepas et al., 2011]. The final topology is shown in Figure 3.7. This is completely consistent with the accepted phylogeny of the 12 reference drosophila, and roughly consistent with the few phylogenetic studies on species from the Saltans group [Rodríguez-Trelles et al., 1999; Singh et al., 2006]. Other methods, which we applied later using different, and larger sets of proteins, also gave the same topology. Note that the Saltans group is sister to *D.willistoni*.

3.4.5 Novel Sec extinctions in the Saltans group

Using selenoprofiles, we searched selenoproteins and Sec machinery in our new genomes, as well as in the rest of available drosophila. A summary of the raw results is presented in figure 3.8. Although generated in full automation¹, these predictions replicate well the results in [Chapple and Guigó, 2008]. The same selenoproteome and machinery of *D.melanogaster* were found in all 12 reference drosophila, with the two exceptions of *D.grimshawi*, carrying an additional Sec copy of SelH1, and *D.persimilis*, where SelG sequence exhibits several insertions near the 3' of the coding sequences, resulting in frameshifts and a premature in-frame stop codon. In the same species we noticed that eEFsec also carries 2 frameshifts. Considering that the other selenoprotein genes (SPS2, SelH1) are well conserved, we must believe that the eEFsec gene is actually functional, and that the frameshifts are assembly artifact. Thus, we must consider the possibility that also the SelG insertions are not real, and that the gene is still a selenoprotein in this species. As in [Chapple and Guigó, 2008], we searched SECIS element downstream of SelG, and thanks to the new SECISearch3, we found a better candidate than the one previously reported. Figure 3.9 shows its sequence aligned to other drosophila SelG SECIS elements.

Although with good overall conservation, *D.persimilis* SECIS shows again an insertion, right before the 5' GA forming the core of the kink-turn. If real, we expect this insertion to impair SECIS function. Considering all observations together, we think that both scenarios for *D.persimilis* SelG are plausible: either SelG is not a selenoprotein anymore, for at least one of its insertions are real, or SelG is still a selenoprotein, and the gene is just prone to artifactual insertions (like eEFsec). We tried to search other nucleotide sources from *D.persimilis* (ESTs), but this did not help to clarify the picture. We hope that future data will solve this enigma.

In the public drosophila genomes previously not analyzed (Baylor's, and *D.santomea*), we found the same Sec genes of *D.melanogaster*. It must be said that

¹The only "manual" intervention to the data displayed in figure 3.8 was the exclusion of gene candidates evidently coming from a bacterial contamination, in our Saltans genomes and also in a few from Baylor's.

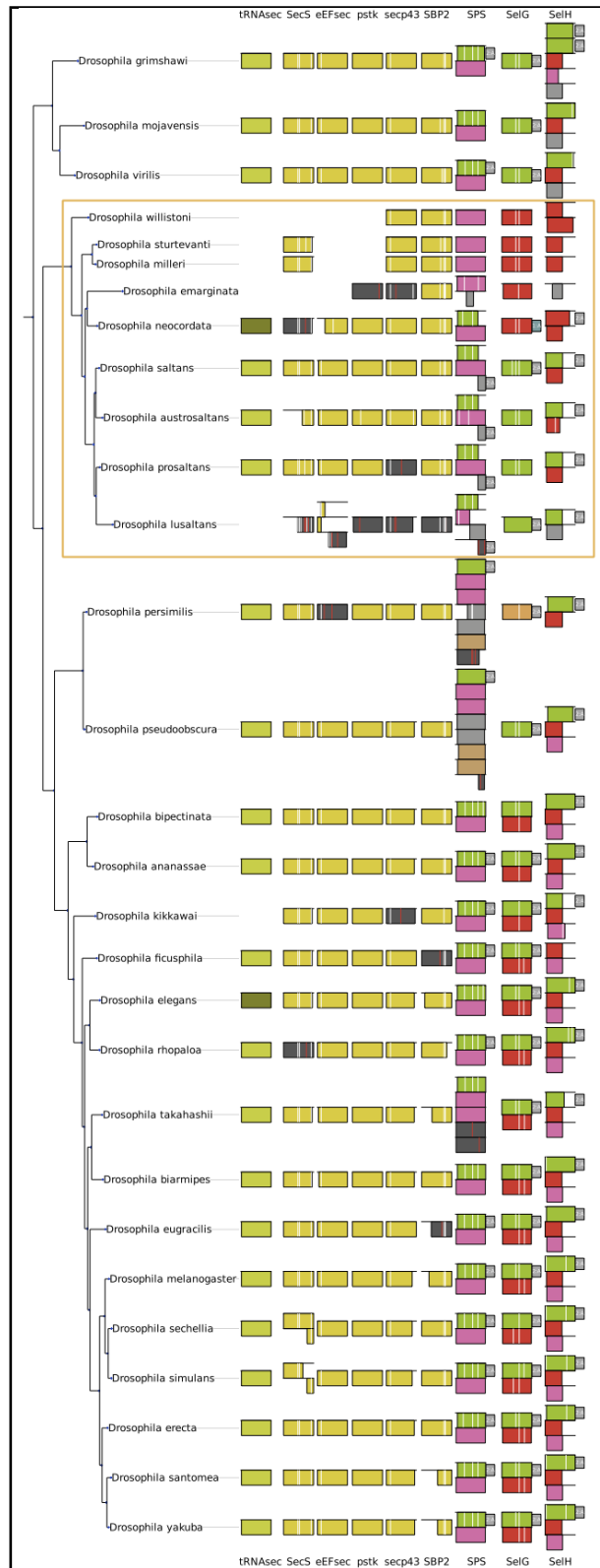


Figure 3.8: (Caption next page)

Figure 3.8: (Previous page.) Summary of selenoprofiles predictions in the 29 sequenced drosophila, drawn with selenoprofiles.tree_drawer. Each gene is indicated as a colored rectangle, whose size and position reflect how the prediction spans the profile alignment. The color indicates the selenoprofiles label, with the same color scheme used before in this thesis (green selenocysteine, red cysteine, pink arginine, yellow machinery protein, dark grey pseudogenes). White lines indicate the position of introns, and red lines indicate insertions causing frameshifts. SECIS elements identified with SECISearch3 are shown as grey boxes on the right side of selenoproteins

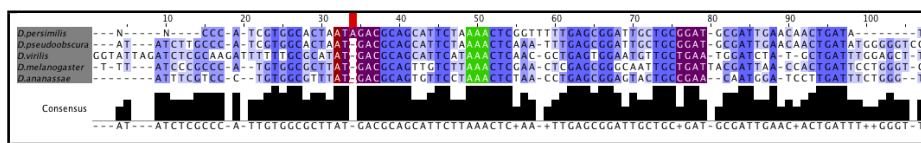


Figure 3.9: SECIS candidate in *D.persimilis* SelG compared with its orthologues in some reference drosophila. The nucleotide pairs that form the SECIS core are highlighted in purple, and the apical conserved adenosines are in green. Note the insertion near the core in *D.persimilis* (column highlighted in red). Picture drawn with the alignment visualizer Jalview [Clamp et al., 2004].

few Sec machinery genes have been predicted with pseudogenes features (in-frame stop codons or frameshifts), and thus were labelled as “pseudo” by selenoprofiles. Nonetheless, considering the imperfect quality of genome assemblies, again we must assume that these are actually intact in the real genome. Indeed, we found such cases mostly in the genomes with worst quality, as approximated by their N50 statistic. This reinforces the idea that *D.persimilis* (the poorest assembly among the 12 reference drosophila) may have nothing special about selenoproteins when compared to *D.melanogaster*.

Inspecting figure 3.8, we noticed an expansion in the SPS family in *D.persimilis* and *D.pseudoobscura* (probably predating their divergence) and also in *D.takahashii*. After phylogenetic analysis, we concluded that they derive from duplications of SPS1. None of them appears to be a selenoprotein, and almost all of them are intronless. It is unlikely that they all are functional genes (again, the scarce RNA data for these species did not help to address this question). We think that they are just gene fragments, generated by a retrotransposition mechanism which for some reason was enhanced for this gene and lineages.

Finally, the Saltans group revealed to be very interesting for selenoproteins, as expected from the PCR results (although some were contradicted). SelH1 was detected with Sec in four species, with Cys in *D.neocordata*, and not found in the *D.sturtevantii*, *D.milleri* and *D.emarginata* genomes. SelG was detected in all species, but only as a Cys homologue in *D.sturtevantii*, *D.milleri*, *D.neocordata* and *D.emarginata*. SPS2 was not found in *D.sturtevantii*, *D.milleri*, and *D.emarginata*.

Already from the selenoprotein content, it was evident that at least *D.sturtevantii*, *D.milleri* and *D.emarginata* lost selenoproteins, either by conversion (SelG, SelH1) or by actual gene loss (SelH1, SPS2). As expected, Sec machinery is incomplete in these species: eEFsec was missing in all these genomes; PSTK is missing in *D.sturtevantii* and *D.milleri* genomes; SecS was missing from *D.emarginata*; tRNAsec (predicted with tRNAscan [Lowe and Eddy, 1997] and then inspected by eye) was missing in *D.milleri*, *D.sturtevantii* and *D.emarginata*, and only a low scoring candidate was predicted for *D.neocordata*. Proteins SBP2, secp43 and SPS1 instead have been found in all Saltans species. For protein SPS2 specifically, we noticed a problem in our assemblies. For many Saltans species (see figure 3.8), we found its last coding exon in a separated contig, presumably because the assembly program could not find an overlap to join it to the rest of the gene. For all those species for which we had RNAseq data, we checked and found the full length transcripts, as expected, pointing to an assembly artifact rather than something biologically relevant. After analyzing all selenoprotein and Sec machinery genes in our species set, we inferred their phylogenetic history, in terms of gene losses or conversions. Figure 3.10 displays a summary of the extant genes and events in the Willistoni/Saltans group.

We consider *D.neocordata* the most interesting species in our set. Here, selenoproteins SelG and SelH1 have been converted to cysteine. A full length SPS2 was detected, but after manual inspection we concluded it is a pseudogene: a single base insertion is present around 50 bp downstream of the Sec-TGA, causing a frameshift that results in an premature stop codon shortly after. In contrast to all other cases, our RNAseq data confirmed the insertion. Other Sec machinery genes showed similar characteristics: SecS and eEFsec were also found with insertions or deletions causing frameshifts, confirmed by RNAseq. A tRNAsec candidate was detected, but evidently degenerated in comparison to the drosophila species with selenoproteins. Given their dissimilarity, we cannot even be sure that our candidate is the real orthologue to tRNAsec of other drosophila (the same is valid for the tRNAsec found in [Chapple and Guigó, 2008] for *D.willistoni*, which is even more dissimilar). The genes SBP2, secp43, pstk, SPS1 were instead found intact in the genome, and also expressed. Taken altogether, these observations indicate that *D.neocordata* underwent a selenoprotein extinction very recently. All Sec machinery genes (including SPS2) are still recognizable, and even transcribed in the cell. Nonetheless, the genes SPS2, eEFsec, SecS and tRNAsec are supposed not functional. We expect these genes to be subject to neutral drift, accumulating mutations that in time will make them not transcribed anymore, and then not even recognizable. *D.emarginata* (sister with *D.neocordata*) has also lost selenoproteins. SelH1 is absent, and so are some Sec machinery genes. SelG is a cysteine homologue. Parsimoniously, we mapped the SelG conversion before the split of *D.emarginata* and *D.neocordata*, although the codons are different in the two species (TGT and TGC respectively), and we consider almost equally likely that 2 parallel conversions actually happened. Although both species are seleno-

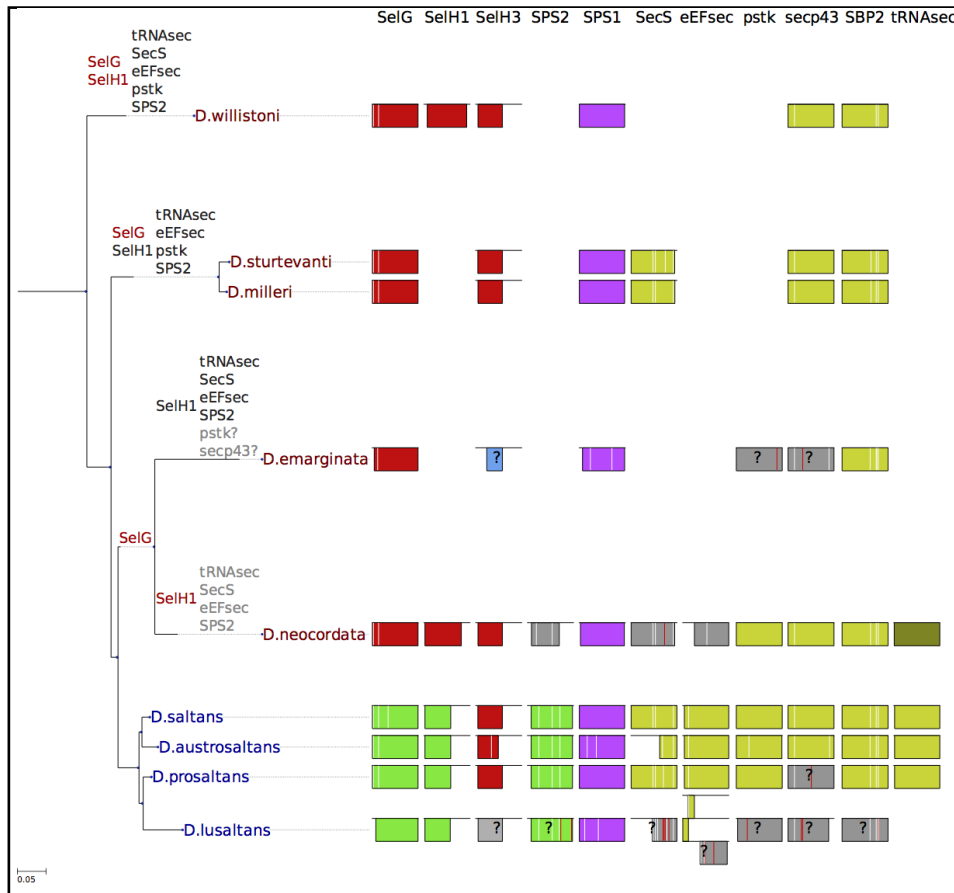


Figure 3.10: Tree of the Willistoni/Saltans group. The colored rectangles represent the selenoprotein and Sec machinery genes found, here split in columns by orthology (figure 3.8 is instead by superfamily). On the branches, gene names indicate the genomic events occurred: red for Sec-to-Cys conversion, black for gene loss, grey for pseudogenization. The names of species that lost selenoproteins are in red, the others in blue.

proteinless, we predict that their common ancestor had selenoproteins. In fact, the extinction of *D. neocordata* is undoubtedly very recent. In *D. emarginata* we see no traces of the Sec machinery genes which are pseudogenized in *D. neocordata*. If the Sec extinction happened before their split, it would have the same age in the two species, and you would expect those genes to have accumulated roughly the same number of mutations (unless something really drastic happened in the rate of neutral evolution in one species only, which sounds unlikely). The two sister species *D. sturtevantii* and *milleri* underwent another, independent Sec extinction, before their split. Here SelH1 was lost, rather than converted. Interestingly, SecS was found intact in both these genomes, suggesting that it may have acquired another function.

Summarizing, we found 3 more Sec extinction events in the Saltans group, one of which is so recent that all Sec machinery genes are still recognizable (*D.neocordata*). Including *D.willistoni*, we have now 4 events of Sec extinctions that happened in parallel drosophila lineages. Considering that the Saltans and Willistoni groups are phylogenetically sisters, we can say that all such events (although independent) happened in a single lineage of drosophila. This prompted us to think that a physiological change occurred at the root of this lineage, favoring later Sec extinctions. This hypothesis is analogous to the one proposed in [Chapple and Guigó, 2008] for the root of insects, and must be seen complementary to it. We then aimed to find specific features in the Willistoni/Saltans group that we could relate to the selenoprotein extinction. For this and other reasons, we decided to fully annotate all drosophila genomes.

3.4.6 Full annotation of drosophila genomes

The full annotation of the protein coding genes in a genome is generally carried out with two conceptually different types of gene prediction methods: homology based, and *de novo*. The first approach is typically much better performing, but unable to predict genes with no homology to any annotated protein. Given the advances of selenoprofiles in the last years, and our familiarity with it, we decided to try and use this homology based tool as the main method for the annotation of drosophila genomes. We used a prototype of selenoprofiles 3 (version 2.3g), which in method is almost identical to version 3.0, whose manual is included in the appendix of this thesis. Before, selenoprofiles had been only used for the finely tuned prediction of few protein families in genomes. Nonetheless, it contained already the steps to resolve overlaps of predictions from different profiles, and so it was suitable for runs with large sets of protein families. Shortly before we undertook this project, we had designed and tested the AWSI scoring method, and integrated it in selenoprofiles (see manual). This proved to be a good method to profile the variance in a protein family alignment, and use it to judge whether gene candidates fit such profile. Before this, the filtering procedures had to be manually tuned for each protein family to get decent results. Now, we believed that the pipeline had become efficient even in the “blind” (i.e., completely automatized) prediction of protein families, given representative profile alignments. To predict the full set of proteins coded in drosophila genomes, we then just needed a comprehensive set of profile alignments, so that almost all proteins coded in the target genomes have some representative homologue among the profiles. We chose to use the Flybase database (<http://flybase.org/>, [Marygold et al., 2013]) as source of good quality annotations. We used the April 2012 release (Dmel_r5.46). Flybase provides full annotations for the 12 reference drosophila, and also includes orthology information linking proteins from different species. This is extremely *D.melanogaster* centric: all groups contain a protein in *D.melanogaster*. We built an alignment for each orthology group in flybase, using the software t-coffee [Notredame et al., 2000]. The almost totality (98%) of the orthologous groups contain only 1-to-1 gene relation-

ships to other drosophila. Thus, very similar protein families with paralogues in *D.melanogaster* are split in different profiles, in contrast to the superfamily-based approach that we used to manually build the selenoproteins and Sec machinery profiles. In other words, the flybase-profiles that we built are orthology-based, rather than homology based, with mostly single copy of any gene per species. We decided to keep it this way, both for simplicity, and to have a rough orthology annotation of results coming from the profile-assignment routine of the pipeline. In selenoprofiles, when gene predictions from different profiles overlap, only one prediction is kept, the one from the profile which is most similar to the candidate. By construction, this process should approximately respect the same orthology assignment that would result from a more complete analysis of results, based on the branching topology of an inferred phylogeny of protein sequences.²

We detected a very few inconsistencies in the data, in particular for the presence of genes with annotated multiple in-frame stop codons, and poor sequence identity with its annotated orthologues. After analyzing manually a few cases, we noticed examples of genes with (putative) biologically relevant events of frameshifts or readthrough, or artifactual frameshifts (insertions in the genome assemblies in respect to the real genome), both poorly managed in Flybase, so that the annotated coding sequences were not correct. Because they were just a very few cases, we simply removed all sequences annotated with multiple stop codons. Since the orthology relationship in Flybase are gene based, we had to decide what to do with alternative protein isoforms, coming from the translation of different transcripts. In practice this was an issue only for *D.melanogaster* (the only species in Flybase with a complete annotation of multiple forms per gene), but we formulated a procedure to solve this problem in general. For each gene, the average length of protein isoforms annotated in each species was computed, and those values were then averaged among all drosophila. For each species with multiple isoforms, the one with length closer to this “consensus length” was selected. Lately, the group of Mar Alba addressed well the problem of protein isoform choice, and also evaluated the impact of different methods on downstream gene analysis [Villanueva-Cañas et al., 2013]. We were delighted to find out that the strategy they developed and proposed as best performing (PALO, <http://evolutionarygenomics.imim.es/palo>) is essentially the same procedure that we had applied for drosophila profiles.

Finally, we ended up with a set of 12170 drosophila profile alignments, with variable number of sequences and different overall sequence similarities (see figure 3.11 and figure 3.12).

We noticed that some profiles were extremely conserved, up to 100% identity across all 12 reference drosophila. This might cause selenoprofiles to be too strict

²Lately Didac Santesmasses tested this, by manually analyzing a single superfamily (thioredoxins). He reports that, while profile-based orthology assignment worked reasonably well, there are advantages in using a comprehensive phylogenetic reconstruction of results of similar families, also because *de novo* predictions can also be included, and thus assigned an orthology. For these reasons, in the next months we will reassign orthology of the full set of predictions, using sequence identity-based clustering and phylogenetic reconstruction.

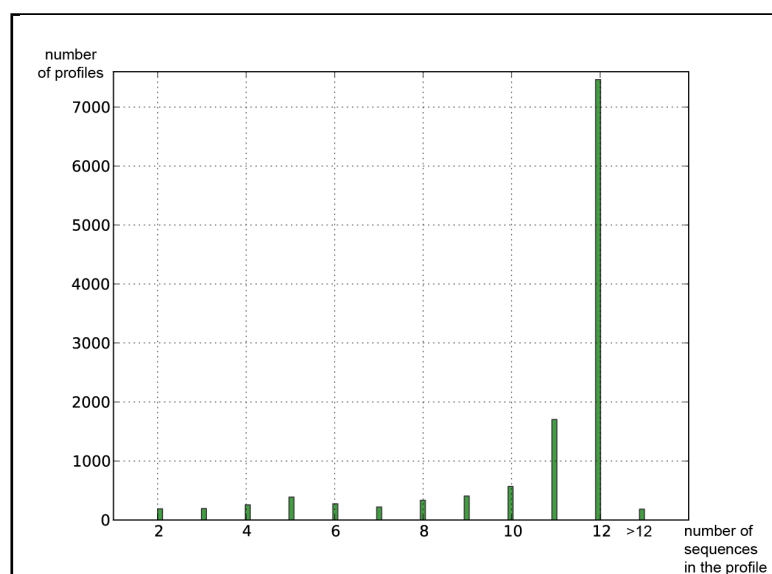


Figure 3.11: Number of sequences in drosophila profiles derived from Flybase.

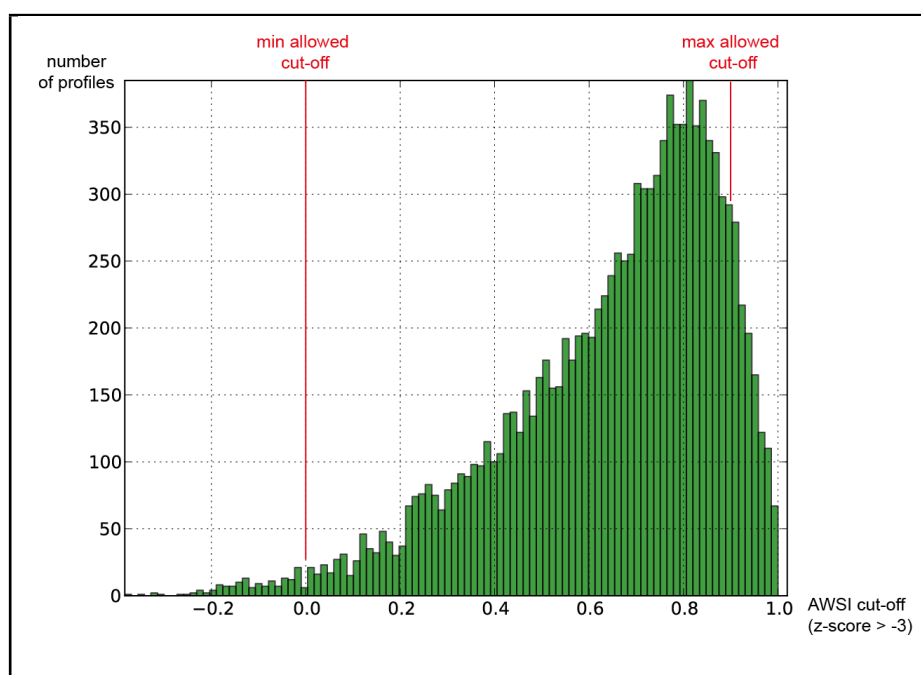


Figure 3.12: AWSI cut-off for the drosophila profiles derived from Flybase, computed as the average AWSI of profiles sequences minus 3 standard deviations. A maximum cut-off value of 0.9 was set, otherwise very conserved profiles would be too strict.

when considering gene candidates, because even little imperfections in the gene prediction will make the identity score lower than the threshold. Thus, we set a maximum possible cut-off for profiles of 0.9, which is still very conservative. This strategy was then implemented as default in selenoprofiles v3.0.

We ran the full set of profiles against all the 29 available drosophila genomes, for many reasons. First, the species already annotated could serve as controls. Second, we wanted to keep the annotations on different species as homogenous as possible in method, to minimize the bias on our genomes. To this aim, we used the annotations that we generated also for the species with an available Flybase annotation, presumably of better quality. We backed up the selenoprofiles predictions with those by the *de novo* tool geneid [Parra et al., 2000], to be able to predict also those proteins for which we lacked a profile. The program was run against all 29 genomes, using the parameter configuration previously optimized on *D.melanogaster* (see <http://genome.crg.es/software/geneid/index.html>). We benchmarked selenoprofiles and geneid predictions against the flybase coding sequence annotations for the 12 reference drosophila. We used the principles illustrated in [Burset and Guigó, 1996], using the same script by Eduardo Eyras already used for the selenoprofiles paper. The sensitivity and specificity at the gene, exon and nucleotide level were computed, allowing to explore the pro and cons of the two methods (see table 3.5).

Selenoprofiles predictions appeared to be very specific (i.e., very few false positives), and more accurate in exon boundaries. Geneid instead was more sensitive, capturing a better proportion of all annotated genes. We thus decided to combine the predictions from the two programs, and we used a simple hierarchical concept: whenever predictions from these two methods overlapped, we removed the geneid prediction from the annotation, keeping the one from selenoprofiles. Anyway, since this would have drastically lowered the reliability (specificity) of our annotations, we filtered geneid predictions using the score assigned by this program. We considered different score thresholds, combined the resulting filtered set with the set of selenoprofiles predictions, and benchmarked using the statistics already mentioned. Table 3.6 shows a summary of results in three representative species.

As expected, our annotations were improved combining geneid and flybase-selenoprofiles. Now we got correct about 90% of the coding sequences in drosophila genomes, with 90% specificity. Although the annotations could still be greatly improved by better exploring the parameters (and in particular improving the profiles), we decided that this was good enough to start looking at our genomes. Unfortunately, *D.willistoni* is the species where we perform worst. This is most likely due to the different GC content of this species, which complicates the job of geneid. This could be avoided using parameters trained for *D.willistoni*, but we preferred to use the same settings for all genomes. We also wanted our full annotation set to include an accurate prediction of the selenoproteins and Sec machinery, independent of the annotation state of these genes in Flybase. Thus, we also included the predictions from our Sec profiles (those shown in figures 3.8 and 3.10), as-

SNg	SPg	SNet	SPet	SNnt	SPnt	
0.87	0.77	0.71	0.64	0.88	0.76	D.ananassae.geneid
0.8	0.98	0.88	0.9	0.85	0.97	D.ananassae.selenoprofiles
0.87	0.99	0.72	0.67	0.89	0.89	D.erecta.geneid
0.83	1	0.91	0.92	0.89	0.97	D.erecta.selenoprofiles
0.87	1	0.68	0.62	0.87	0.89	D.grimshawi.geneid
0.79	0.98	0.86	0.88	0.83	0.97	D.grimshawi.selenoprofiles
0.89	0.84	0.72	0.68	0.88	0.84	D.melanogaster.geneid
0.9	0.97	0.87	0.91	0.89	0.98	D.melanogaster.selenoprofiles
0.86	0.95	0.67	0.6	0.86	0.85	D.mojavensis.geneid
0.8	1	0.87	0.88	0.83	0.97	D.mojavensis.selenoprofiles
0.85	0.81	0.64	0.58	0.86	0.74	D.persimilis.geneid
0.75	1	0.79	0.74	0.8	0.93	D.persimilis.selenoprofiles
0.87	0.98	0.7	0.64	0.89	0.89	D.pseudoobscura.geneid
0.77	1	0.86	0.88	0.81	0.98	D.pseudoobscura.selenoprofiles
0.85	0.82	0.69	0.63	0.87	0.79	D.sechellia.geneid
0.8	1	0.85	0.82	0.86	0.94	D.sechellia.selenoprofiles
0.85	0.95	0.68	0.61	0.86	0.84	D.simulans.geneid
0.8	1	0.83	0.8	0.84	0.93	D.simulans.selenoprofiles
0.87	0.94	0.68	0.61	0.86	0.85	D.virilis.geneid
0.81	1	0.87	0.89	0.83	0.97	D.virilis.selenoprofiles
0.78	1	0.6	0.5	0.73	0.86	D.willistoni.geneid
0.75	0.95	0.85	0.88	0.8	0.96	D.willistoni.selenoprofiles
0.86	0.94	0.71	0.65	0.88	0.87	D.yakuba.geneid
0.81	0.99	0.9	0.91	0.87	0.97	D.yakuba.selenoprofiles
Averages:						
0.858	0.918	0.683	0.619	0.861	0.839	geneid
0.801	0.996	0.862	0.868	0.842	0.962	selenoprofiles

Table 3.5: Benchmarking geneid and selenoprofiles with drosophila profiles on the Flybase annotation for the 12 reference species. SN: sensitivity (true positives / all annotated genes). SP: specificity (true positives / all predicted genes). SN and SP were computed at the gene level (SNg, SPg), exon level (SNet, SPet) and nucleotide level (SNnt, SPnt). For the exon level, only the genes correctly paired at the gene level were considered, while the nucleotide level contains everything.

signing them the highest place in hierarchy when combining them with geneid and selenoprofiles-flybase predictions.

Figure 3.13 shows the number of predictions in the final annotation set for all drosophila. Our genomes harbor fewer predictions than the rest. Presumably this is due mostly to the assembly quality, although other factors may also play a role. The assemblies of *D.emarginata* and *D.lusaltans* were particularly poor (consistent with their N50, see table 3.4), followed by *D.sturtevanti* and *D.milleri*.

For many genome-wide analysis, it is appropriate to use set of genes having a 1-to-1 orthologue in considered species. We defined such sets of 1-to-1 orthologues initially using the profile-approximation, taking all the profiles which had exactly one prediction per species. Depending on the species chosen, this number will vary a lot. Considering all 29 species, this results in 1160 orthologous group of proteins (profiles). Considering a reduced set of 13 species with best quality

species	threshold	SNnt	SPnt
D.virilis	0	0.91	0.88
	5	0.9	0.9
	10	0.9	0.93
	15	0.89	0.94
	20	0.89	0.95
	25	0.88	0.96
	30	0.88	0.96
	35	0.88	0.96
	40	0.88	0.96
	45	0.87	0.97
	50	0.87	0.97
D.willistoni	0	0.87	0.88
	5	0.86	0.91
	10	0.85	0.93
	15	0.84	0.94
	20	0.84	0.95
	25	0.83	0.95
	30	0.83	0.95
	35	0.83	0.96
	40	0.83	0.96
	45	0.82	0.96
	50	0.82	0.96
D.melanogaster	0	0.93	0.86
	5	0.93	0.88
	10	0.93	0.9
	15	0.92	0.92
	20	0.92	0.94
	25	0.92	0.95
	30	0.92	0.95
	35	0.92	0.96
	40	0.92	0.97
	45	0.91	0.97
	50	0.91	0.97

Table 3.6: Benchmarking the annotation sets of combined geneid and selenoprofiles, using different thresholds for filtering geneid predictions. SNnt= sensitivity at the nucleotide level. SPnt= specificity at the nucleotide level. The indexes for three representative species are shown. We chose 10 as optimal score threshold.

genomes, we have 6080 groups instead. Lately, we increased this number using the phylogenetic signal: we computed a protein tree for each profile, including the predictions in all species. Within each tree, we then searched for clusters of genes all belonging to different species, indicating their orthology. This extends the concept previously described. All profiles with exactly one result per species are included, but not only: if for example two sets of results are present for a profile, forming two completely separated clusters in the protein tree, two 1-to-1 orthologous groups will result within the same profile. Considering all 29 species, this now results in 1196 orthologous groups, a very modest increase. This again is due to the fact that the profiles were already built based on orthologous groups, so we do not expect duplications within the results of each single profile.

3.4.7 GC content and codon usage shift in Willistoni/Saltans

Having decent annotations for most drosophila species, we analyzed the genomic features previously mentioned. From literature [Powell et al., 2003], we expected Saltans and Willistoni to be homogenous for GC content and codon bias. Figure

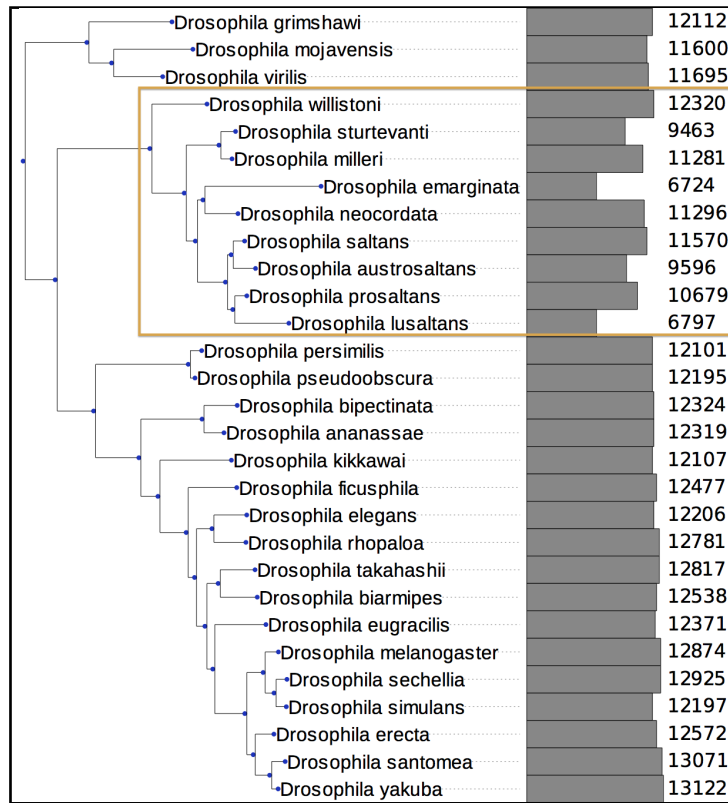


Figure 3.13: Number of gene predictions per drosophila genome in our final annotation set.

3.14 shows the GC content in the genome and 4-fold degenerate boxes of 1-to-1 orthologous coding sequences, across all drosophila. Indeed, the genomic GC content of all species belonging to the Willistoni/Saltans group is lower than any other drosophila, although not by much. The GC content in coding sequences is also lower and exhibits a much bigger difference, almost 2-fold. In a previous study [Singh et al., 2006], authors quantified the background substitutional patterns in *D.saltans* and *D.willistoni*, comparing with those of *D.melanogaster*. They found that indeed the naturally occurring mutations changed in Willistoni/Saltans, shifting the theorized GC equilibrium point towards more AT rich. Nonetheless, according to their math, this shift can account for differences observed at the whole genome level (or in introns), but not for the bigger difference in the coding sequences. Thus, authors hypothesized that a major shift in codon preference also occurred.

When codon bias is considered, the Willistoni/Saltans group again appears homogenous, and different from the rest of drosophila. We considered two measures to quantify codon bias. The effective number of codons (ENC) ranges from 20 to 61, and quantifies how far the codon usage of a gene departs from equal usage

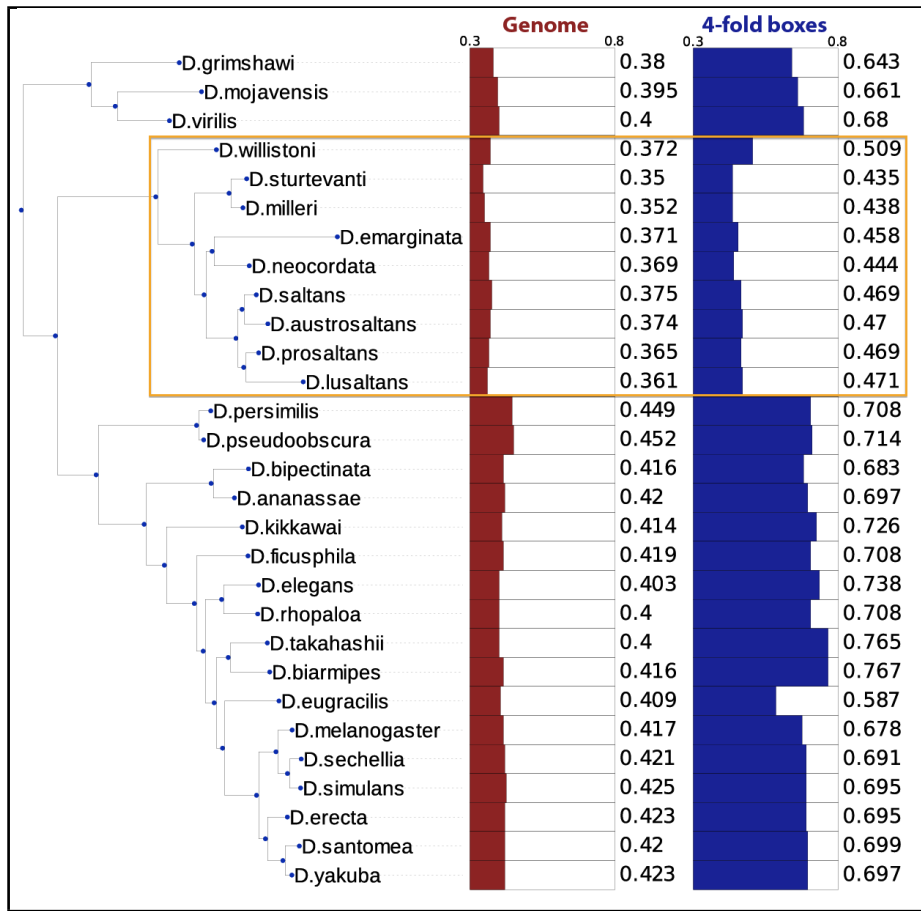


Figure 3.14: GC content across drosophila computed on the whole genome and on the 4-fold degenerate boxes in the coding sequences of 1160 1-to-1 orthologues.

of synonymous codons [Wright, 1990]. The lower the ENC, the more biased is the codon usage: the extreme value of 20 implies that for each amino acid, a single codon is always used. We computed ENC as described in [Sun et al., 2013]. The relative synonymous codon usage (RSCU) is a measure for each codon, and similarly it quantifies how much this codon is overrepresented comparing to neutral expectations (all synonymous codons with equal frequency). We used ENC to have a global idea of how much biased coding sequences are in a genome, and RSCU to pinpoint the differences in usage for each codon.

Figure 3.15 and figure 3.16 show a summary of the results on 1-to-1 orthologous genes in all drosophila. As expected, the Willistoni/Saltans group exhibits a lower overall bias (higher ENC). The preferred codons for many amino acids changed in this lineage favoring A or T ending codons. Interestingly we identified another, unreported case of codon usage shift in *D. eugracilis*. In contrast with Willistoni/Saltans, the genomic GC content in this species does not seem affected.

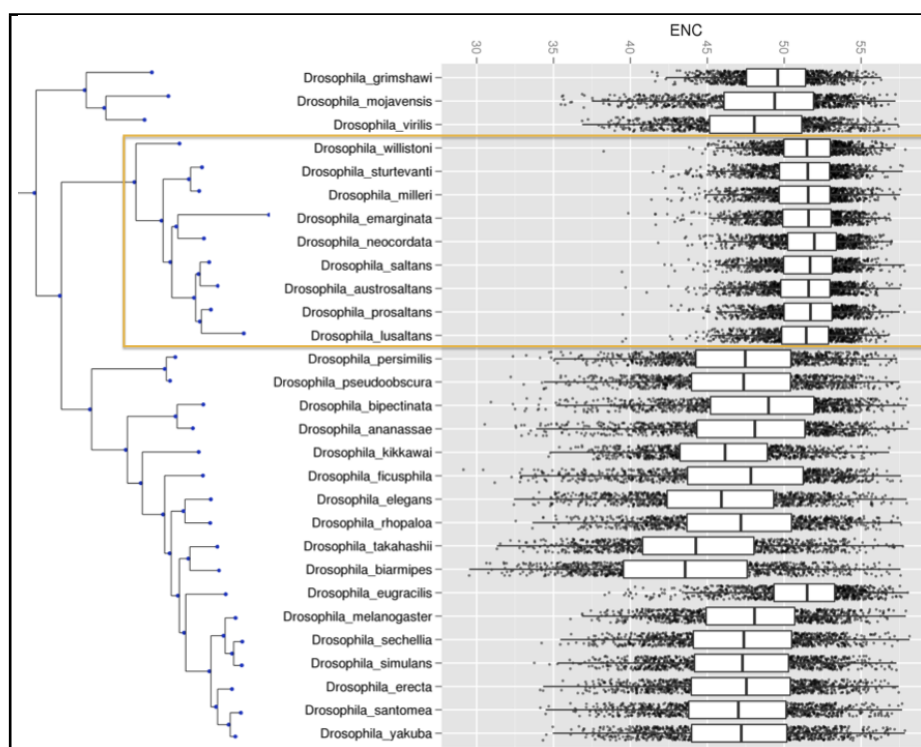


Figure 3.15: ENC (effective number of codons) computed on 1160 1-to-1 orthologous genes across all sequenced drosophila. Each dot represent a single gene. The bar includes 50% of values. Lower ENC values mean more codon bias. The Willistoni/Saltans group is less biased overall, similarly to (but independently of) *D.eugracilis*. Plot provided by Didac Santesmasses.

This could be explained by a different nature of the shift (not by change in background mutational patterns), or by a more recent age of the shift, so that this is still not observable at the whole genome level. We computed the same plots also considering fewer species (13), allowing to exclude the genomes with worst quality and at the same time to increase greatly the number of analyzed 1-to-1 genes (6080). This did not change the patterns observed, so we included here only the most complete plots.

3.4.8 Widening the picture: other arthropods

To put the drosophila Sec extinctions in context, we investigated as many insects and other arthropods as possible. All available genomes were downloaded from NCBI, and scanned with selenoprofiles. Figure 3.17 shows a summary of results to date.

In accordance with our previous results on fewer species, all organisms belonging to Hymenoptera, Coleoptera and Lepidoptera showed no intact selenoprotein

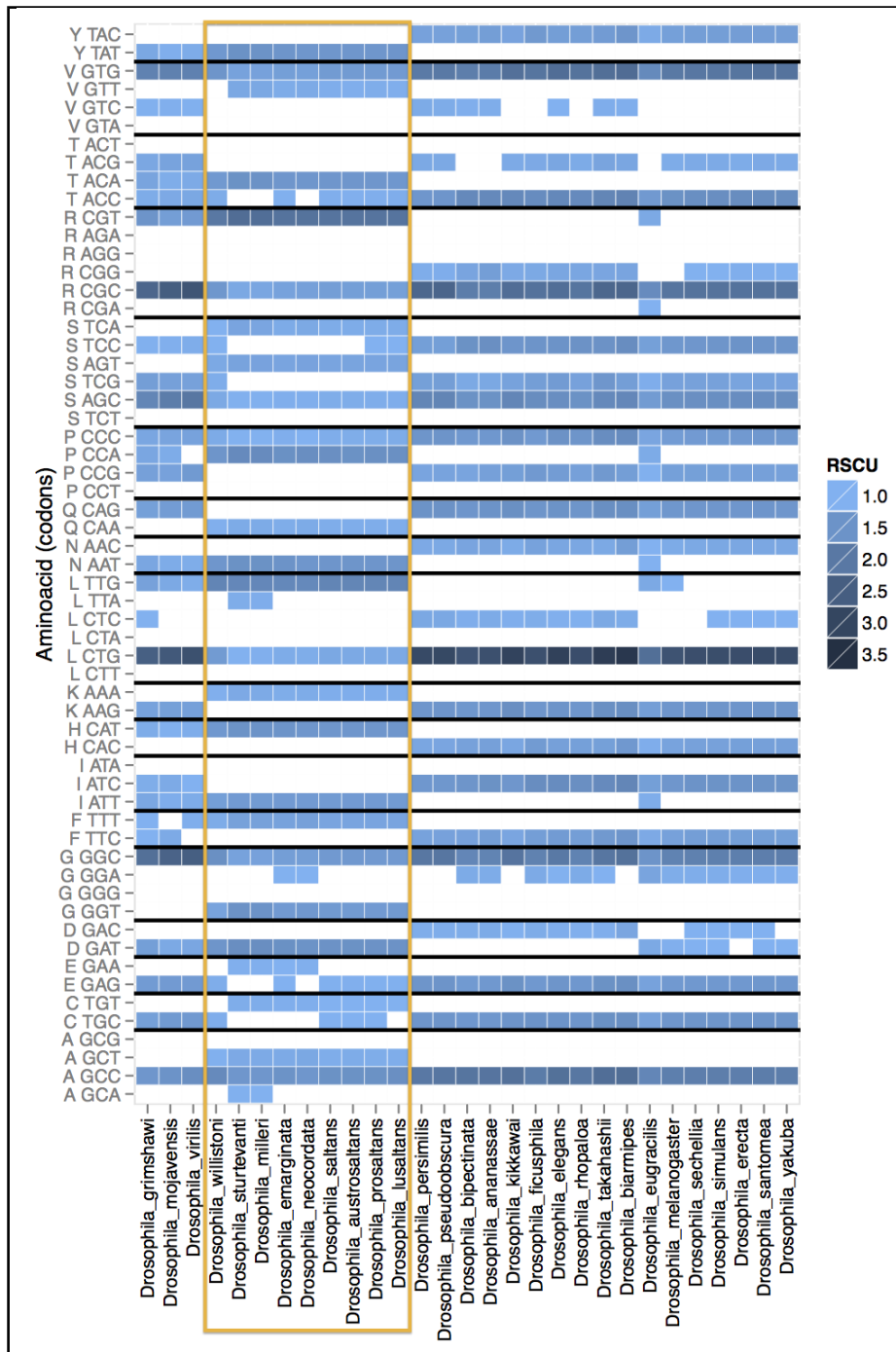


Figure 3.16: RSCU (relative synonymous codon usage) computed on 1160 1-to-1 orthologous genes across all sequenced drosophila. Higher RSCU values (darker color) means higher preference for that codon. Note that Willistoni/Saltans shifted the optimal codons towards AT rich. *D.eugracilis* exhibits a similar trend. Plot provided by Didac Santesmasses.

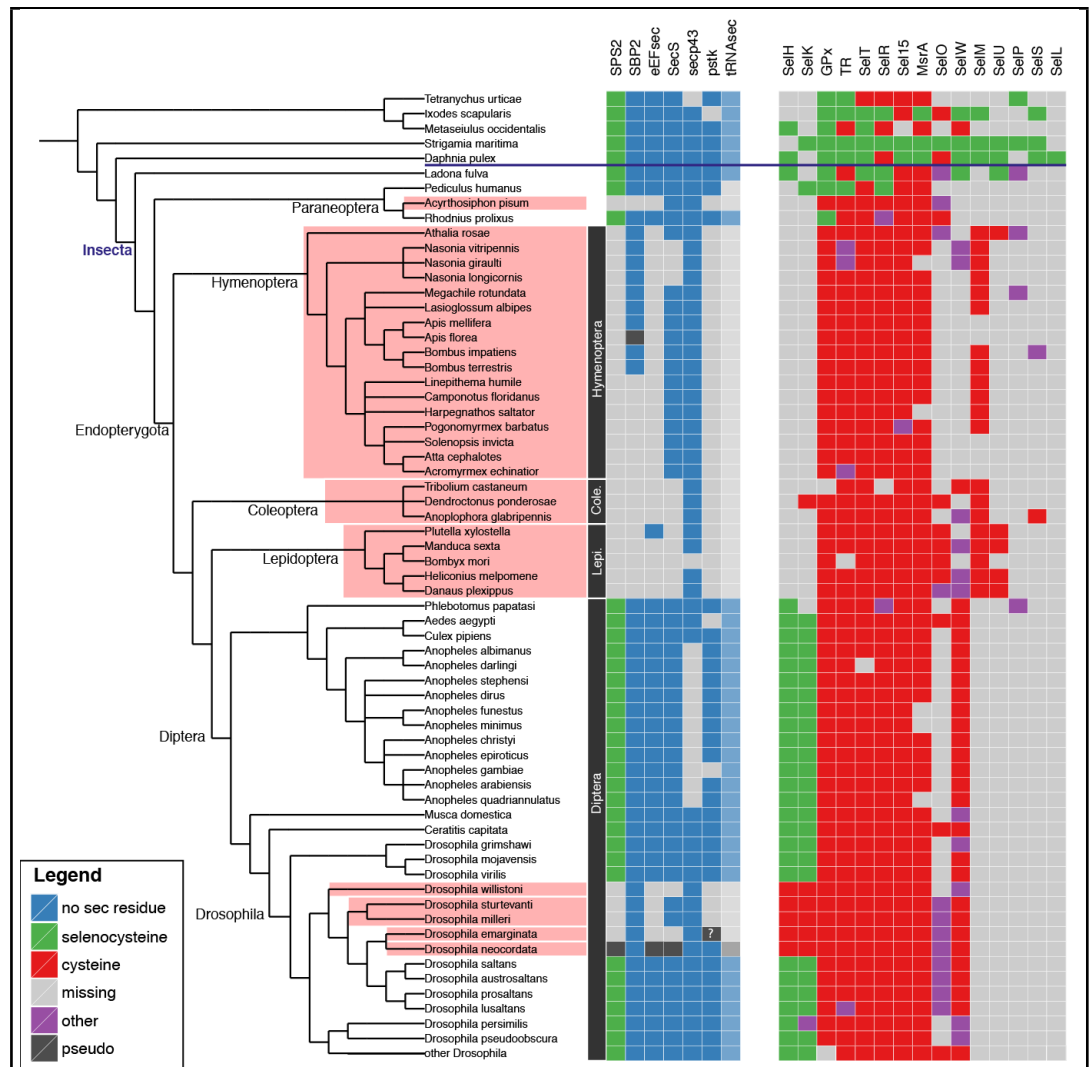


Figure 3.17: Sec machinery and selenoprotein genes in arthropods. The colored tables show the presence of Sec machinery genes (middle) and of selenoprotein families (right). Multiple genes may be present for a selenoprotein family, but only one label is displayed: this is chosen by hierarchy (selenocysteine over cysteine, over other, over missing). Selenoproteinless species are highlighted in red background. The thick blue line (top) separates insects from the rest of arthropods. Adapted from poster "Selenoprotein extinction in insects" by Didac Santesmasses, presented at Selenium meeting 2013.

genes, and also lacked a complete machinery. Given their phylogenetic topology, these three insect orders appear to have lost selenocysteine in independent events, probably at their root. Looking at Sec machinery, we identified as best markers of Sec coding ability the genes eEFsec³, tRNA^{sec} and SPS2 (although the presence of SPS1 genes may complicate its use as Sec marker - see next results section). Pstk also works reasonably well, except for the fact that we could not detect it in a few species with selenoproteins (although this could be due to genome incompleteness). SBP2 instead was conserved in all selenoproteinless drosophila, as well as in number of Hymenoptera. SecS was conserved in *D.sturtevantii* and *milleri*, and also in all Hymenoptera except the parasitic wasp genera of *Nasonia*. Finally, Secp43 was conserved in almost all selenoproteinless species, and apparently lost in all *Anopheles* mosquitoes (that possess selenoproteins). Among Diptera, no selenoprotein other than those observed in *D.melanogaster* could be found. Instead, other eukaryotic families were found as selenoproteins in Paraneoptera. In this insect order, we have 3 species with an available genome: *Pediculus humanus* (human louse), *Acyrtosiphum pisum* (pea aphid) and *Rhodnius prolixus* (known as kissing bug). Notably, among them only pea aphid lacks selenoproteins, suggesting a more recent Sec extinction in comparison to those detected in the other insect orders (excluding drosophila). *P.humanus* possess a rich selenoproteome, including three important anti-oxidant selenoprotein families: glutathione peroxidase (GPx), thioredoxin reductase (TR) and methionine-S-sulfoxide reductase (SelR). If we keep walking away from drosophila, we then find the most basal insect in our set: *Ladona fulva* (dragonfly). This species possesses the richest selenoproteome found among insects: it has Sec forms for families SelH, GPx, SelT, SelR, SelW and SelU. The same families, plus others, were found in non-insect arthropods. For example *Strigamia maritima* (centipede) and *Daphnia pulex* (water flea) possess a very rich selenoproteome, quite similar to the vertebrate one. In one or both these genomes, we found Sec forms for 16 selenoprotein families, all those included in figure 3.17.

These results fit very well with earlier work [Chapple and Guigó, 2008], and traces a path of progressive Sec loss in insects, culminating in complete Sec extinction in the selenoproteinless organisms like *D.willistoni*. We predict two main points of Sec loss: one at the root of insects, when many eukaryotic selenoproteins were lost or converted (Sel15, MsrA, SelM, SelO, SelU, SelP, SelS, SelL); and the other one at the root of Endopterygota (GPx, TR and SelR), possibly corresponding to a major change in the anti-oxidant systems of insects.

3.4.9 A functional model for selenocysteine in drosophila

Let's zoom back in to drosophila. Only three selenoproteins were present in their ancestral genome, with one that works only to produce selenocysteine (SPS2).

³Searching eEFsec, we found in several assemblies some genes extremely similar to SelB of common bacterial contaminants (e.g. Enterobacter). If one uses eEFsec as Sec marker, this must be taken into account.

SelG is the putative orthologue of human SelK (despite very poor sequence identity), which appears to perform a redox reaction possibly related to endoplasmic reticulum associated degradation process (ERAD). The targeted knock-down of *D.melanogaster* SelG gene by RNAi [Morozova et al., 2003] decreased viability (25% of embryos hatched) and caused morphology defects. A cysteine paralogue (SelG2) is present in the Melanogaster group only (visible in figure 3.8), for which we have no phenotypic data.

SelH (BthD) is believed to have a redox related role. Its knockdown in *D.melanogaster* reduced drastically viability (7% of embryos hatched) and decreased the antioxidant capacity of cells [Morozova et al., 2003]. Two paralogues are found in *D.melanogaster*, one with cysteine (SelH3) and one with arginine (SelH2) aligned to Sec. Interestingly, from phylogenetic reconstruction SelH2 appears to have originated before the split of drosophila, and was lost at the root of the Willistoni/Saltans group (and also in *D.persimilis* independently). This could be somehow related to the subsequent selenoprotein loss: possibly, it testifies the decreased importance of a redox related process for the fitness of this lineage. The severe SelH and SelG knockdown phenotypes are in apparent contradiction with the viable SPS2 knockout by transposable elements (Flybase, original reference: [Bellen et al., 2004]), since SPS2 is required for selenocysteine production. In vertebrates, there are evidences of sulfur entering selenium pathways when selenium supply is low, creating a Sec to Cys backup system. The same system in drosophila, if present, may explain the absence of evident SPS2 phenotypes.

The patterns of selenoprotein conversion and loss observed in the Willistoni/Saltans group prompted us to attempt the formalization of a model for Sec extinctions (see figure 3.18). Although some concepts can be generalized to insects, this model is thought for drosophila only.

While SelG was found conserved in all drosophila (as a cysteine homologue in selenoproteinless species), SelH was either lost or converted (see figure 3.10). This suggests that SelG function is more important, or at least more difficult to replace, than SelH. We believe that SPS2 is always the last selenoprotein to be lost. In fact, as part of the Sec machinery, its function is required as long as any other selenoprotein has a useful function. We never observed SPS2 converted to cysteine in insects. This may be related to its increased evolution rate specifically in this lineage (see next results section on SPS). Because of this, we expect this enzyme to possess a lower catalytic efficiency than, for example, its human orthologue, and if this is the case, its cysteine conversion would be less acceptable, for it would decrease even more the catalytic efficiency. The first key events to Sec extinction are then on SelG and SelH. These selenoproteins have to be converted to cysteine homologue, or become “useless enough” so that they can be lost without affecting fitness. When this happens, selenocysteine remains with no selection acting on it, for it is not anymore linked to any function. In this transient state, which we can see in extant *D.neocordata*, the genes forming the Sec machinery start to accumulate mutations that rapidly inactivate them, and finally make them disappear from the genome (diverge beyond recognition power). We observed that not all Sec ma-

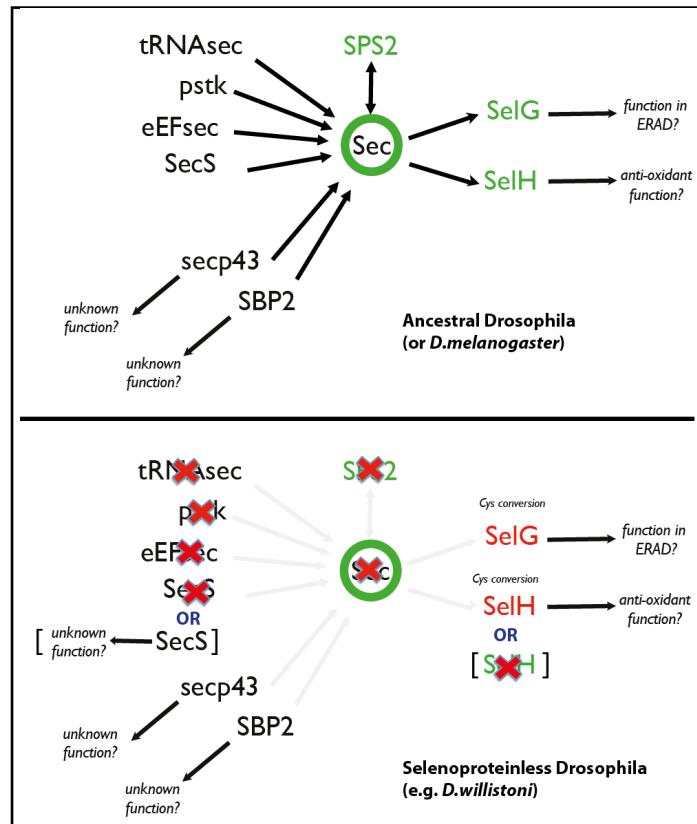


Figure 3.18: Functional model of selenocysteine in drosophila. Arrows can be read as "is required for". The upper panel show the situation in most extant drosophila, and is predicted to apply to the ancestral drosophila. The lower panel summarizes the observations in selenoproteinless species of the Willistoni/Saltans group.

chinery genes are lost: secp43 and SBP2 are always conserved in drosophila. This means that purifying selection is still active, implying that these genes are linked to some other function. For SBP2, a possible explanation is its involvement in the GAPsec-mediated readthrough system [Hirosawa-Takamori et al., 2009]: although still poorly characterized, this system appears in fact to depend on SECIS elements. SecS is generally lost when selenocysteine disappear. A notable exception is its conservation in *D.milleri* and *D.sturtevantii*. We believe that this indicates that it has been adopted to some other function specifically in this lineage.

3.4.10 Why Willistoni/Saltans?

As said, we observe selenoproteinless drosophila only within this group, despite a reasonable number of genomes are available. It is logical then to search for a link between the peculiar genomic characteristic of this group and its propensity to lose selenoproteins. Despite our efforts, we could not find any clear relation.

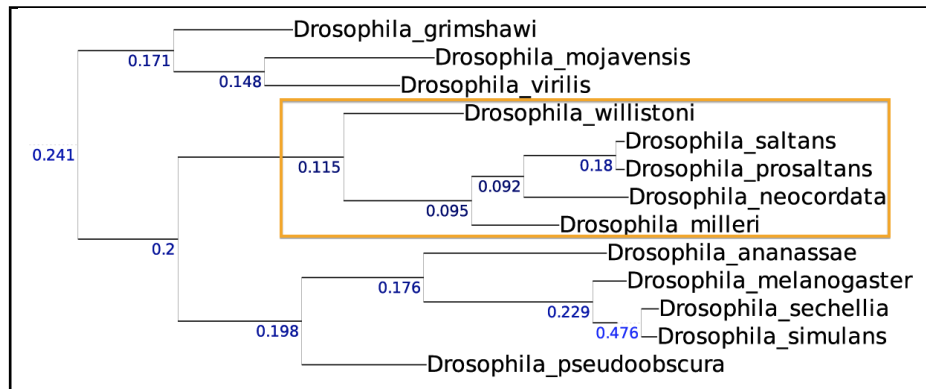


Figure 3.19: Overall non-synonymous per synonymous change rates in drosophila, computed with pycodeml on 6080 orthologous protein coding genes.

It was previously claimed that this group has an increased rate of amino acid evolution, possibly related to its codon usage and GC content shift [Rodríguez-Trelles et al., 1999]. If this is true, selenoproteins may have been lost as a consequence of a generic reduction of selection efficiency. To test this, we selected a set of 13 drosophila species. We focused on the best quality genomes, and also we tried to balance the tree to have an approximately symmetric topology between the Saltans and Melanogaster groups (see figure 3.19). We then extracted from our annotations the proteins with 1-to-1 orthology in these 13 species (6080 groups). We aligned them at the protein level with t-coffee [Notredame et al., 2000], and the inferred coding sequence alignments were concatenated. All columns with any gap were then removed to minimize noise. This resulted in a large alignment of 2,570,641 codons. We used the program pycodeml (Mariotti M, unpublished) to infer the sequences at ancestral nodes using the sankoff algorithm, and to compute a lineage-dN/dS value for each node (numbers below branches in figure 3.19). The lineage-dN/dS constitutes a better approximation for the rate of non-synonymous to synonymous changes than the classical Ka/Ks value, which simply counts the observed changes. The lineage-dN/dS is computed as the proportion of possible non-synonymous mutations of an ancestral sequence that are observed in at least one extant species under the tested node, divided by the same proportion for synonymous changes. As you can see in figure 3.19, this value is lowest in the Willistoni/Saltans group. Other analogous tests (not shown) also gave the same results. In contrast to our expectations, the overall rate of non-synonymous per synonymous change seems lower in Willistoni/Saltans. Note that the worse quality of our genomes would lead to an overestimation of the dN/dS, because there are more possible non-synonymous changes than synonymous. For these reasons, we find unlikely that selenoproteins are being lost for a generic decrease of selection efficiency across the entire genome.

Instead, we think that the causes of Sec extinctions in drosophila have to be

searched around the functions of SelH and SelG. The mentioned loss of SelH2 (arginine homologue) in Willistoni/Saltans may be an indication of some important change in the anti-oxidant system. In this view, we analyzed the content in anti-oxidant families in our annotations. Interestingly, we found a duplication of the thioredoxin deadhead (dhd) mapping precisely at the root of Willistoni/Saltans group. Although very speculative, it is plausible that this novel thioredoxin made more dispensable the function of the SelH family, allowing the loss of SelH2, and the cys-conversion or loss of SelH1. More likely, these genes are just few players among many that contributed to a change of the redox biology of the species. In this scenario, we can speculate that the GC and codon usage shift have played a role as a driving force of change. Genes which are particularly GC rich should have a general reduced fitness under the new codon usage pressure. The loss of such a redox related gene could have triggered a change in these systems. Curiously, we see that indeed the GC content of SelH2 in the basal *D.mojavensis* is exceptionally high in 4-fold boxes (0.86), although this does not hold for *D.virilis* (0.67) and just partially for *D.grimshawi* (0.72).

3.4.11 Conclusions

In this study we followed selenocysteine extinctions in insect genomes, and particularly in drosophila Willistoni/Saltans. We sequenced and annotated 8 genomes from this group, a useful resource to investigate also their peculiar GC content and codon usage. We found 4 independent Sec extinctions in this lineage, and 4 more in other insects (Hymenoptera, Lepidoptera, Coleoptera, pea aphid). Analyzing also other arthropods, we traced a precise path of selenoprotein conversions or losses, which started at the root of insects and then culminated in parallel complete Sec extinctions. Within Willistoni/Saltans, we found the most recent Sec extinction in *D.neocordata*. This species appears to be in a transient state, with several Sec machinery genes not functional, yet still recognizable as pseudogenes. Observing the pattern of gene loss of Sec machinery in selenoproteinless lineages, we hypothesize some of them to be working in pathways unrelated to selenocysteine (at least in some species): SPS1, SBP2, secp43, SecS. We condensed our observations in a functional schema for selenocysteine in drosophila. According to our model, the last selenoprotein standing is always SPS2, because it cannot be lost without compromising other selenoproteins, and because its conversion to cysteine does not seem feasible for function in this lineage.

3.5 The SelenoPhosphate Synthetase family (SPS)

SPS is both a selenoprotein and part of the selenocysteine machinery, and for this reason I have always considered it the most interesting selenoprotein. I wanted to describe its phylogeny as accurately as possible, thinking that its history would reflect the history of selenocysteine itself, to which it is tightly linked functionally. After years of research, we show how the SPS genes possessed by living species passed through an incredible journey of genomic events, such as gene duplications, gene losses, conversions from cysteine to selenocysteine and vice versa, gain of function and subfunctionalization events. The history of SPS proteins gives us an example of how gene functions can be lost or duplicated, and shows a few possible outcomes of function duplication. It is entirely plausible that all extant protein families went through a similar complexity of events, shaping their function and their sequences.

This part is in form of a paper draft, almost ready for submission. Here is its provisional title and authors:

Mariotti M, Santesmasses D, Mateo A, Capella-Gutiérrez S, Arnan C, Johnson R, Yim SH, Gladyshev VN, Gabaldón T, Corominas M, Guigó R.

Neo/subfunctionalization in the phylogeny of SelenoPhosphate Synthetases, marker of selenocysteine utilization.

3.5.1 Abstract

SelenoPhosphate Synthetase (SelD/SPS) is an enzyme necessary for the production of selenocysteine (Sec), the 21st amino acid inserted specifically in selenoproteins. SPS is a selenoprotein itself in many organisms, and is present in all species encoding selenocysteine. In this work, we predicted and reconstructed the phylogeny of all SelD/SPS proteins, providing a map of selenium utilization traits across the sequenced tree of life. Moreover, SPS in metazoa revealed an insightful snapshot of protein function evolution. Supported by KO-rescue experiments in *Drosophila*, we claim that the ancestral metazoan Sec-containing SPS (SPS2) acquired a secondary function, probably exerted by an alternative protein isoform produced by non-Sec readthrough. In time, this led to an impressive variety of genomic events occurring independently in various metazoa lineages, all transferring the secondary function to a new selenocysteine-less protein (SPS1): alternative transcripts originated in ascidians, and gene duplication by retrotranspositions or other means occurred in ascidians, insects, annelida, and vertebrates.

3.5.2 Introduction

SelenoPhosphate synthetase (SPS, also called SelD or selenide water dikinase) catalyzes the synthesis of selenophosphate from selenide, ATP and water, producing AMP and inorganic phosphate as products. Selenophosphate (SeP) is the selenium donor for the production of the non-standard amino acid selenocysteine (Sec or U), taking place of its own tRNA [Palioura et al., 2009; Xu et al., 2007b]. Selenocysteine is inserted co-translationally into a number of proteins (selenoproteins) in response of a UGA. This stop codon is recoded by the presence of a stem loop structure on gene transcripts, the SECIS element, in a mechanism which was elucidated in Bacteria [Yoshizawa and Böck, 2009; Kryukov and Gladyshev, 2004], Eukaryotes [Squires and Berry, 2008], and Archaea [Rother et al., 2001]. In order for an organism to express selenoproteins, it is required a set of factors which we will collectively call the selenocysteine machinery. These include proteins necessary both for the production and insertion of Sec. SPS serves for the former, and interestingly is often found as a selenoprotein itself - being the only Sec factor with this characteristic.

SPS proteins are conserved from Bacteria to human with about 30% identity, and are found in all known species encoding selenoproteins. In prokaryotes, SPS is found also in species where selenophosphate is used to produce selenouridine in tRNAs [Romero et al., 2005]. The two traits (Se-tRNA and Sec) overlap but not completely, with species identified to possess one, the other, both or none [Romero et al., 2005]. In eukaryotes SPS is generally found as selenoprotein (SPS2), while in prokaryotes homologues with cysteine aligned to the Sec position are also common. Conversion to cysteine of selenoproteins is a common process and it was observed extensively within insects [Chapple and Guigó, 2008], and also in vertebrates [Mariotti et al., 2012]. Cysteine-homologues of selenoproteins have their

same expected molecular function, although catalytic efficiency or substrate specificity may change: substitution of Sec to Cys decreased (but did not abolish) SPS2 activity in mouse [Kim et al., 1997]. Both in vertebrates and in insects two SPS genes are known, one being a selenoprotein - SPS2 - and one being not - SPS1, carrying a threonine in vertebrates and an arginine in insects [Tamura et al., 2004]. Conversion of Sec to something different than a cysteine is a much more rare event to observe. In contrast to cysteine conversion, here the molecular function appears to have changed. While SPS2 has been shown to produce SeP, SPS1 seems not to: murine SPS1 was shown not to produce SeP in vitro, and neither consuming ATP in a selenium dependent way [Xu et al., 2007a]. *Drosophila* SPS1 too was shown not to catalyze selenide dependent ATP hydrolysis and not to complement a SPS lesion in *E. coli* [Persson et al., 1997]. Knockout by RNAi of SPS1 in mouse cell lines has been shown not to affect selenoprotein synthesis [Xu et al., 2007b]. In insects, SPS1 is conserved in species which underwent selenoprotein extinctions events, that were lost along with machinery proteins [Chapple and Guigó, 2008]. This suggests that SPS1 functions in a pathway unrelated to selenoprotein biosynthesis [Lobanov et al., 2008], although this is still debated: human SPS1 has been found to interact with selenocysteine synthase (SecS, also named SLA/LP) [Small-Howard et al., 2006]. Also, human SPS1 has been proposed to recycle selenocysteine, since a *E. coli* SelD mutation can be rescued by SPS1 but only when fed L-selenocysteine [Tamura et al., 2004]. The structure of a bacterial SPS and of human SPS1 have been recently solved [Itoh et al., 2009; Wang et al., 2009], and the mechanism of reaction is debated. SPS acts as a dimer. The Sec residue is positioned on a N-terminal domain which appear to be mobile in the various structural configurations. The proposed role for Sec in this protein is the delivery of the selenide (tied by a perselenide bond) to the catalytic site, where an ATP is split subsequently into ADP then AMP [Ogasawara et al., 2001; Wang et al., 2009]. Attempting to untangle the functional relationship of SPS proteins, we tried to solve their phylogeny. The results revealed an unpredictable complexity. We first discovered that human and *drosophila* SPS1 were generated independently along the two lineages. We then found other SPS gene variants, one of which was particularly puzzling: in hymenopteran, we found a single SPS2 gene, with a tightly conserved in-frame TGA. Since it lacks a SECIS, and more importantly Sec was lost in this lineage [Chapple and Guigó, 2008], the TGA cannot be readthrough as Sec. But why should it be conserved anyway? Since selenocysteine was lost in Hymenoptera, the gene's most presumable function (production of SeP) is supposed useless. To solve the enigma, we attempted to reconstruct the full history of SPS genes, starting from its roots in prokaryotes.

3.5.3 Results and Discussion

We used a profile-based gene prediction tool [Mariotti and Guigó, 2010] to search for SelD/SPS genes in all sequenced eukaryotic and prokaryotic lineages. We then reconstructed their phylogenetic history, by a combination of approaches. Analyses

are fully discussed in Supplementary Material S1-S5. Results are summarized hereafter.

3.5.3.1 SelD in prokaryotes as marker for Se utilization traits

Figure 1 shows the presence of SelD genes in a set of reference prokaryotic genomes, along with the presence of other selenium utilization gene markers (see caption). SelD was found in prokaryotic lineages as Sec (20%) or Cys (80%) forms, with a rather scattered distribution. SelD genes were found only in 27-35% of the investigated prokaryotes species (see Supplementary Material S1). Its presence fits well with the rest of the machinery for selenocysteine (SelA, tRNA^{sec}) and selenouridine (ybbB), and also with selenoprotein presence. The selenocysteine trait (SelD, SelA, tRNA^{sec}, selenoproteins) was found more abundant than selenouridine (SelD, ybbB), although the two traits had good overlap. Supplementary Material S1 contains a description of the genes found in each major lineage investigated. Numbered points in Figure 1 attempt to provide an overview through snapshots of certain lineages of interest. Among Archaea (1), SelD was found uniquely in Methanococcales and Methanopyri genomes, which are rather rich in selenoproteins (see also [Rother et al., 2003]). In Methanococcales only, the selenouridine trait was also found, although with a peculiarity: ybbB is split in two adjacent genes [Su et al., 2012]. Clostridia (2) exhibit a large diversity, including species with and without selenocysteine and selenouridine, and many examples of Sec to Cys conversions of the SelD gene. In Pasteurellales (3) we identified instead a bona-fide Cys to Sec conversion, the first one ever documented. In fact, most of Gammaproteobacteria appear to possess a Cys-SelD (or none), and Sec forms are found almost only in Pasteurellales. Phylogenetic sequence signal supports codon conversion rather than horizontal transfer as most likely explanation (Supplementary Material S1). During our analysis, we found also many examples of horizontal gene transferred SPS genes, involving diverse lineages. The selenocysteine and selenouridine traits were found conserved in all *Escherichia* (4). In general though, their presence appeared to be rather scattered across the prokaryotic tree, testifying a dynamic process of gene loss and gain. It is then quite common to observe a very limited number of species in a lineage possessing a Se utilization trait, as for example selenouridine in *Bacillus coagulans* (5) and *Paenibacillus mucilaginosus*. Notably, increasing the number of analyzed species (and thus the resolution) reveals a more complex pattern, and one can see for example that some Bacilli possess selenocysteine (see Figure SM1.1, in contrast to what figure 1 would suggest). In almost every species with a SelD gene, a SelA and/or ybbB gene was identified, indicating the utilization of SeP for selenocysteine and/or selenouridine. A notable exception was in the *Enterococcus* genus, where many species including *E. faecalis* (6) possessed SelD but no other marker. This had already been reported as indicator of a 3rd selenium utilization trait [Romero et al., 2005; Zhang et al., 2008]. Selenium is in fact used here as cofactor to molybdenum hydroxylases [Haft and Self, 2008; Srivastava et al., 2011].

Figure 1: (Previous page.) Phylogenetic map of SPS and selenium utilization traits in prokaryotes. The sunburst tree shows the phylogenetic structure of investigated species, and the presence of SelD genes and other markers of selenium utilization. The section for Archaea is zoomed as a guide to interpret the plot (1). Every circular section represent a taxonomic rank in NCBI taxonomy (superkingdom, phylum, class, order, family, genus). The last two outermost sections are for species, and display observations for each of the 223 reference prokaryotes. A black bar outside represents the number of selenoproteins detected. The outermost circle is color coded for the presence of ybbB and SelA, with a black dot inside for tRNA^{Sec}. The sunburst tips are labelled for SelD presence and type: Sec-SelD, Cys-SelD, no gene found. The color is propagated to the lower ranks by hierarchy. Transparency is used to display how many species under a lineage have its same label. Assuming no Cys to Sec conversion and no horizontal transfer, colors reflect the predicted SelD presence at ancestral nodes. This allows to detect by eye the Sec to Cys conversions, for example in Clostridia (2). The hierarchical color assignment was violated only for Gammaproteobacteria, altered to be red (Cys-SelD). In fact its only sublineage with Sec-SelD is Pasteurellales (see 3), for which our analysis points to a Cys to Sec conversion instead. The plot can be used the map the selenocysteine (SeC) and selenouridine (SeU) traits (see top-right panel). For example *Escherichia coli* (4) has both, Pasteurellales only SeC (3), and *Bacillus coagulans* only SeU (5). *Enterococcus faecalis* has a Cys-SelD gene but no other Se utilization marker (6) [Romero et al., 2005; Zhang et al., 2008]. Expanded versions of the plot (up to 8286 species) are available in Supplementary Material S1. Note that gene fusions and extensions are not considered here.

3.5.3.2 SPS2 in eukaryotes as marker for selenocysteine

Figure 2 shows the SPS genes and predicted selenoproteins found in a representative set of eukaryotic genomes. The presence of SPS2 genes (defined as those with Sec, or Cys instead) correlates perfectly with the presence of selenoproteins. The search for ybbB (selenouridine synthase) lead to only a few candidates, mostly in protozoa, which are not reported here (analysis is ongoing). SPS2 and selenoproteins were found always together, but with a rather scattered phylogenetic distribution in protozoa, testifying a dynamic evolution similar to bacteria. In fact selenocysteine is not found in several species of Stramenopiles, Alveolata, Amoebozoa, presumably due to multiple independent event of selenoprotein extinction. The highest number of selenoproteins was predicted among stramenopiles, in pelagophyte algae *Aureococcus anophagefferens* already described for its rich selenoproteome [Gobler et al., 2011, 2013]. Most other stramenopiles species were also predicted with selenoproteins: brown algae *Ectocarpus siliculosus*, diatoms like *Phaeodactylum tricornutum*, several Oomycetes including *Phytophthora*, and the parasite *Blastocystis hominis*. Among red algae, the species *Chondrus crispus* was predicted devoid, while *Cyanidioschyzon merolae* was predicted to possess

Figure 2: (Previous page.) Phylogenetic map of SPS genes and approximate selenoproteome size of eukaryotes. The plot summarizes the results on 505 genomes analyzed, compressed to 213 displayed here. The tree was partitioned in lineages and highlighted in grey tones for the only purpose of visualization. Near the tips, the presence of SPS proteins is displayed as colored rectangles. Selenocysteine (green) and cysteine (red) forms are called SPS2, with the other homologues are called SPS1 (top legend). The SPS gene extensions found for some Cys-SPS2 are indicated with a letter inside its rectangle (see bottom left legend). The number of selenoproteins predicted in each genome is indicated with a black bar.

SPS and two selenoproteins. Consistently with literature [Lobanov et al., 2009], no bona fide SPS and selenoproteins were found in Fungi or land plants (Embryophyta), despite the high number of genomes searched (284 and 41 respectively, compressed for Figure 2). In contrast, green algae genomes were found abundant in selenoproteins, as expected [Novoselov et al., 2002; Palenik et al., 2007], with *Ostreococcus lucimarinus* reaching the peak of 28 predicted selenoproteins. SPS2 and selenoproteins were found also in all investigated Kinetoplastida (Euglenozoa), including *Trypanosomas* and *Leishmanias*. Selenoproteins with no homology to any known domain have been previously described in these species [Lobanov et al., 2006b; Cassago et al., 2006]. Other lineage specific selenoproteins have also been reported (and detected in this work) for alveolates: in Sarcocystidae (including *Neospora* and *Toxoplasma* [Novoselov et al., 2007]), and in *Plasmodium* species [Lobanov et al., 2006a]. Additionally, we identified the selenocysteine trait in other alveolates genomes: the apicomplexan *Eimeria tenella*, all investigated Ciliates, and species *Perkinsus marinus*. Selenoproteins and SPS were detected also in all investigated Amoebozoa (including Mycetozoa like *Dictyostelium*), with the only exception of the Archamoebal *Entamoeba*. A rich variety of SPS genes were found in metazoa, described in detail in the next paragraphs. But before moving our attention to the animal genomes, we describe how several SPS proteins in lower eukaryotes were found to possess extensions with additional domains.

3.5.3.3 Gene fusions and extensions in primitive eukaryotes, shared with prokaryotes

Some SPS genes were previously reported to be fused with other genes: with a NADH-dehydrogenase domain in certain bacteria and lower eukaryotes, with a Cys sulfinate desulfinate / NifS protein in *Geobacter sp. FRC-32* [Zhang et al., 2008]. Recently, heterolobosean species *Naegleria gruberi* [da Silva et al., 2013] was reported to possess a SPS gene fused with a methyltransferase protein. We ran a computational procedure to identify both annotated and undiscovered gene fusions or extensions in our prediction datasets (see Supplementary Material S2). We report here the extensions supported by strong conservation. In all cases detected, the extension was at the N-terminal side of SPS, and SPS carried Cys (not

Sec), with the only exception of the NifS/Sec-SPS fusion in *Geobacter sp. FRC-32*. Fusions with NADH dehydrogenases were by far the most common, being found in a wide range of Bacteria (including Cyanobacteria, Alphaproteobacteria, Betaproteobacteria, Gammaproteobacteria - see figure SM2.1) and also in many lower eukaryotes, including both green and brown algae and other protists (see Figure 2). On the contrary the methyltransferase-SPS fusion was detected only in *N. gruberi*. In the same species, we identified a second SPS protein not previously reported. This is also product of a gene fusion, with a NifS-like protein. We detected similar NifS-SPS fusions in two bacterial species, *Geobacter sp. FRC-32* and the obscure *Caldithrix abyssi* [Miroschnichenko et al., 2003], and also in the amoeba *Acanthamoeba castellanii*. Lastly, all *Plasmodium* species were found to possess a large extension at the N-terminal (>500 amino acids). This extension shows no homology with any known protein, and its function remains unknown. In almost all cases of extended SPS, we expect the gene to have acquired an additional function, retaining the original SeP production activity. In fact, we found selenoproteins and other Sec machinery genes in all these species. A possible exception is the NifS-SPS fusion, since we observe a second SPS gene in every species possessing it (all except *A. castellanii*, for which nonetheless we found a gene fragment possibly indicating the presence of an additional Sec-SPS gene). It has been long known that gene fusions are very common in prokaryotes and also in primitive eukaryotes, and are important tools by which protein functional networks evolve. In metazoans we observed a very different functional scenario, with frequent gene duplications.

3.5.3.4 Independent duplications of SPS2 generates SPS1 proteins in metazoans

In eukaryotes, SPS cysteine forms were found common only among lineages basal to metazoans, with the only exception of nematodes (see figure 2). Therefore we argue that the last metazoan ancestor possessed a single SPS gene with selenocysteine (SPS2). In many metazoan lineages we detected additional SPS genes, generated by independent duplications of SPS2. Although not monophyletic, we will refer to all these genes as SPS1, for reasons that will be clear later. SPS1 proteins can be distinguished from SPS2 in that they are neither selenoproteins, nor cysteine homologues. Human SPS1 carries a threonine aligned to selenocysteine of SPS2. We mapped its origin within the documented whole genome duplication at the root of vertebrates (see Supplementary Material S3). Besides support by sequence-based phylogenetic reconstruction, you can see in Figure 2 that all investigated non-vertebrate deuterostomes possess a single SPS2 gene (although tunicates deserve a special mention in the next section). Also, the conservation of intron structure is consistent with a whole gene duplication. As reported in [Mariotti et al., 2012], then in mammals the SPS2 gene duplicated again, this time by retrotransposition, generating a second SPS2 gene almost identical to the parental, except for the lack of introns. In placentals, the intronless SPS2 then replaced

functionally the parental gene, while non-placentals mammals still retain the two copies (see for example *Monodelphis domestica* in Figure 2). We identified another SPS2 duplication at the root of the Clitellata lineage (Annelida), generating a SPS1 protein carrying leucine (Leu) aligned to the Sec position (see Supplementary Material S3). We mapped the origin of insect SPS1 proteins after the split with other arthropods, as described later. Researchers working on eukaryotic SPS always assumed that human (vertebrate) and drosophila (insect) SPS1 were orthologous, monophyletic genes. Our phylogenetic analysis demonstrates that they were instead generated independently along the two lineages, forcing to reconsider previous experiments at the light of this finding.

3.5.3.5 Alternative transcript isoforms in ascidians split by gene duplication in Styelidae and Pyuridae

Tunicates are the closest outgroup to vertebrates [Delsuc et al., 2006], with ascidians (sea squirts) constituting its best studied and most sequenced lineage. In the ascidian *Ciona* we identified a single SPS gene, that appears to be the direct descendant of the ancestral metazoan SPS2. Nonetheless, here the gene produces two different protein isoforms, deriving from alternative exon structures at the 5' (see Supplementary Material S4). One isoform carries selenocysteine (SPSsec - SPS2), while the other one, previously unreported, has a glycine aligned instead (SPSgly - SPS1). Extending the analysis to all tunicates, the picture became even more interesting (see summary within Figure 3). We mapped the origin of the SPSgly isoform to the root of ascidians, since it was not found in the non-ascidian tunicate *Oikopleura dioica*. In contrast, an homologous gene producing SPSgly and SPSsec isoforms was identified in ascidian *Molgula tectiformis*. The most interesting case was found in species *Botryllus schlosseri* and *Halocynthia roretzi*, belonging to the sister lineages of Pyuridae and Styelidae. Here the coding sequences of SPSgly and SPSsec were found, but they reside in distinct genomic loci: a distinct gene for each isoform (see Supplementary Material S4). The SPSsec gene is intronless, and possess a SECIS downstream as expected. The SPSgly gene possesses instead the ancestral intron structure (very similar to *O.dioica* SPS2), and has no SECIS. Therefore, we concluded that the SPSsec-specific transcript retrotransposed to the genome at the root of Pyuridae and Styelidae. This generated a copy that soon replaced functionally the SPSsec isoform of the parental gene, which as a result specialized only in the SPSgly isoform, as both the selenocysteine coding exon and the SECIS element degenerated.

3.5.3.6 SPS1 in Hymenoptera and the conserved SRE element: non-Sec readthrough

Hymenoptera is an order of insects that includes ants, bees and wasps, and that has been target of an intense sequencing effort in the last years. Just like several other insect lineages, hymenopterans have lost their ability to use selenocysteine: in fact they lack a complete Sec machinery, including the Sec-tRNA, eEFsec and

pstk [Chapple and Guigó, 2008]. In these genomes we did not find any intact selenoprotein gene, as they were converted to cysteine or lost. With one possible, very puzzling, exception. In all investigated hymenopterans we found a single, extremely conserved SPS gene. It possess a in-frame UGA just like SPS2, but no SECIS element can be found downstream. Because a complete Sec machinery is missing in these organisms, the gene cannot be a selenoprotein. But its striking conservation pattern strongly argues for it is indeed translated. As we already mention in [Chapple and Guigó, 2008], we believe that SPS in hymenoptera is translated through a readthrough mechanism which does not involve Sec insertion. We have now several points to support this (see Supplementary Material S5). First, there is evidence for abundant stop codon readthrough in insects, with TGA being the most frequently observed readthrough codon observed in drosophila [Jungreis et al., 2011]. Second, we noticed a conserved hexanucleotide subsequent to the UGA in *Hymenoptera*: GGG-TG[C/T]. This was found highly overrepresented subsequent to known viral “leaky” stop codons (although in those cases the stop codon was UAG, [Harrell et al., 2002]). Third, the gene contains a very conserved secondary structure just downstream of the UGA. It was first described in [Howard et al., 2005]: in this work, similar stem-loop structures (called SRE, from selenocysteine redefinition elements) were identified in many selenoprotein genes, including SPS2. The SRE element of human SelN was analyzed in particular detail, and was shown to promote readthrough of reporter genes. We expect SRE of SPS2 to possess an analog readthrough-promoting activity, and this should be valid even more for SPS1 hymenoptera, for its peculiar hexanucleotide. We ran the program RNAz [Gruber et al., 2010] to characterize the secondary structures embedded in the coding sequence of all SPS genes (see Supplementary Material S5). In prokaryotes, this yielded the bacterial SECIS of the Sec containing SelD genes (figure SM5.1). In eukaryotes, we obtained stable stem loops in the same region of all UGA-containing SPS genes. The largest and most stable structures were in Hymenoptera, where we predicted a 3 stems clover-like structure with the UGA on the apex of the middle stem. Overall similar structures were predicted in all metazoan SPS2 genes (see figure SM5.2). The readthrough-enhancing hexanucleotide can be seen in the consensus structures of SPS1-rt hymenoptera, but also in SPS2 genes of other bilateria and metazoa that are basal to insects and vertebrates. Therefore we analyzed the codons found in this region of all SPS genes (figure SM5.3). Figure 5 contains a summary of results in all UGA containing SPS genes. The hexanucleotide is found in SPS1-rt genes in hymenoptera and paraneoptera (described in the next paragraph), and also in SPS2 genes of metazoa that are basal to insects (other protosomes), to vertebrates (other deuterostomes) or to both (non-bilaterian metazoa).

3.5.3.7 SPS phylogeny in insects

Insects provide a unique phylogenetic framework to study selenoproteins. Many insect lineages underwent a complete selenoprotein extinction, in which seleno-

protein genes were converted to cysteine homologues or lost, and the selenocysteine machinery degenerated. This process occurred in several lineages in parallel: Hymenoptera, Lepidoptera, Coleoptera (or at least *Tribolium castaneum*), a single sequenced species of drosophila (*D. willistoni*) [Chapple and Guigó, 2008], and paraneopteran pea aphid (*Acyrtosiphon pisum*, [Aphid-Consortium, 2010]). Consistently with its expected function, the SPS2 gene was found in every insect genome with selenoproteins, and missing in all others (see Figure 2). As said, *D. melanogaster* possess a second gene called SPS1, with arginine aligned to the Sec positions of SPS2. Similar SPS1-Arg genes were identified in all Endopterygota except the basal Hymenoptera (Lepidoptera, Coleoptera, Diptera) and also in pea aphid. In our phylogenetic reconstruction (Supplementary Material S3, figure SM3.5) the hymenopteran SPS1-rt gene clusters with these SPS1-Arg. Two interesting paraneopteran genes are found in the same cluster, belonging to *Rhodnius prolixus* and *Pediculus humanus*. These genes possess the in-frame UGA but no SECIS, just like SPS1-rt in hymenoptera. They also exhibit the same hexanucleotide just downstream, GGGTGT (see Figure 5). We believe them to be readthrough, like hymenopteran SPS. A second gene was found in both these genomes, with a UGA and a SECIS downstream: SPS2. Other selenoproteins were detected in these two paraneopteran genomes (Figure 2). We concluded (follow Figure 3) that all insect SPS1 genes derive from a SPS2 duplication at the root of insects, initially generating a UGA-containing, SECIS lacking gene (SPS1-rt). The original SPS2 gene then started to diverge more rapidly, and was finally lost in all lineages where selenocysteine disappeared. Meanwhile, the new gene switched the UGA codon to arginine generating SPS1-Arg proteins, both in the pea aphid and in the last ancestor of Coleoptera, Lepidoptera, Diptera. In hymenoptera and the rest of paraneoptera, the gene is still conserved with UGA and no SECIS, namely as SPS1-rt.

3.5.3.8 Functional hypothesis: parallel subfunctionalization generates SPS1 proteins

So far, we observed how the ancestral metazoan SPS2 selenoprotein duplicated independently through various genomic events (see Figure 3 and 4): whole genome duplication (vertebrates), gene duplication (annelids, insects), alternative exon usage (ascidians). Besides, we noted the readthrough SPS2 in Hymenoptera and Paraneoptera. In all these cases, there is a new protein overall similar to the original selenoprotein SPS2, but with selenocysteine replaced by some amino acid different than a cysteine. These non-Cys, non-Sec SPS homologues (which we collectively call SPS1) have molecular function which is different from SPS2 – or at least it is true for both insect and vertebrate SPS1. We formulated an hypothesis to explain the duplications of the SPS family in eukaryotes. We think that the ancestral SPS2 protein had not only its known catalytic activity (synthesis of selenophosphate from selenide - f1), but also an additional function, which we name f2. Eventually, several eukaryotic lineages have split these two func-

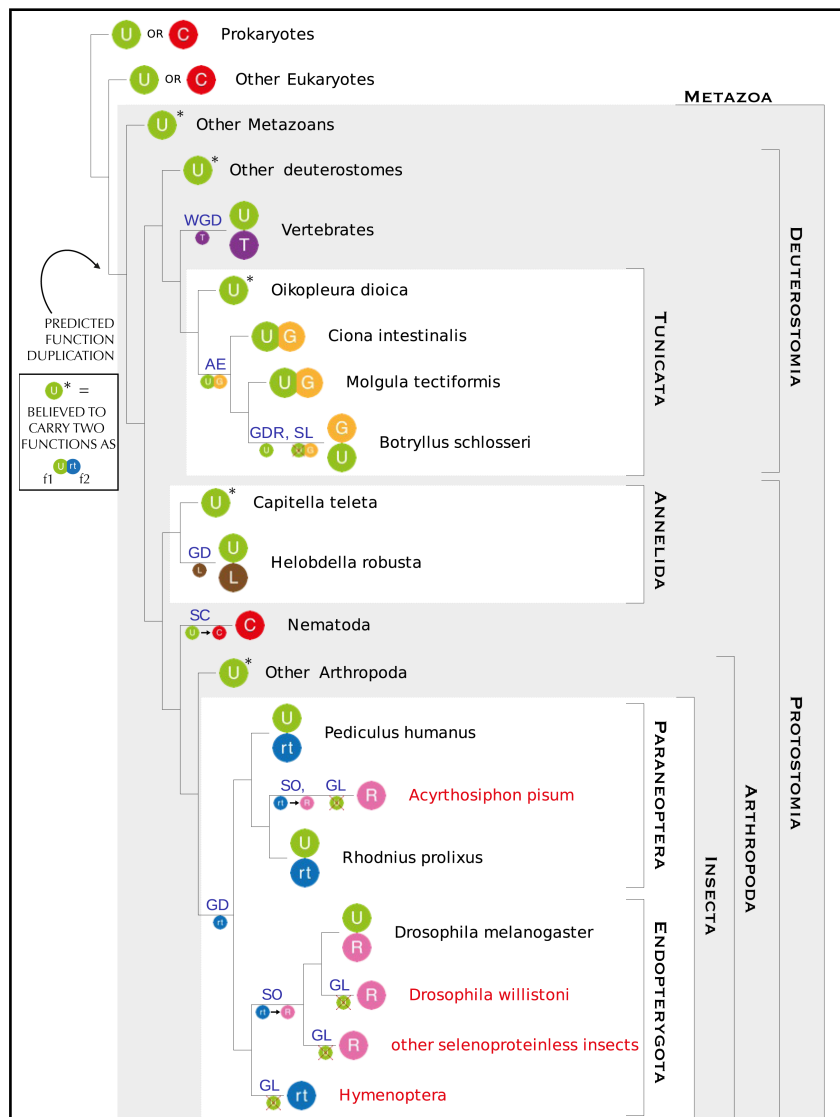


Figure 3: Phylogeny of SPS proteins. The colored balls represent SPS genes, indicating the residue found at the Sec position (U for selenocysteine, C for cysteine, T for threonine, G for glycine, L for leucine, R for arginine, rt for readthrough - unknown residue). The structure of the genes is expanded in Figure 4. Insects lacking selenoproteins are in red font. The main genomic events shaping SPS genes are indicated on the branches: WGD gene copy retained after whole genome duplication, AE origin of an alternative exon, GDR gene duplication by retrotransposition, SL selenocysteine loss, GD gene duplication, SC conversion of selenocysteine to cysteine, SO conversion of selenocysteine to something other than cysteine, GL gene loss. In our subfunctionalization hypothesis (see text), we map parsimoniously the duplication of function at the root of Metazoa.





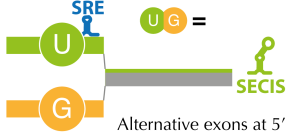


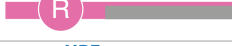

SPS forms	Name	Found in	Function
	Sec-SelD	Bacteria	f1
	Cys-SelD Cys-SPS	Bacteria, Basal eukaryotes, Nematodes	f1
	SPS2	Eukaryotes	f1 (+ f2)*
	SPS1-Thr	Vertebrates	<u>f2</u>
	SPS-ae	Ascidians (e.g. Ciona)	f1 + <u>f2</u>
	SPS1-Gly	Styelidae & Pyuridae (e.g. Botryllus)	f2
	SPS1-Leu	Clitellata (e.g. Helobdella)	f2
	SPS1-Arg	Insects	<u>f2</u>
	SPS1-rt	Hymenoptera, Paraneoptera	<u>f2</u>

Figure 4: Summary of SPS forms identified. Proteins are classified for the residue found at the Sec position (see also figure 3). The presence of peculiar secondary structures is also indicated: bSECIS for bacterial SECIS element, SRE for selenocysteine recoding element [Howard et al., 2005], SECIS for eukaryotic SECIS element, HRE for hymenopteran readthrough element. In the rightmost column, the functions predicted for the various protein forms is indicated. f1 is the production of SeP. f2 is defined as the uncharacterized molecular function of *Drosophila* SPS1 (double underlined), and was confirmed for other proteins with our KO-rescue system in *drosophila* (underlined). *: for eukaryotic SPS2, f2 in parentheses indicates that some such genes are predicted to possess both functions, those indicated also with a star (*) in figure 3 (basically all metazoans with no SPS1 protein). Note that gene fusions and extensions are not considered in this figure.

tions, with a new, duplicated protein taking over f2. If this hypothesis is true, then the molecular functions of all SPS1 proteins (although paraphyletic) should be the same: f2. Also, the species which never experienced a duplication of SPS2, but descends from the same common ancestor of species that did, are expected to possess a SPS2 gene carrying both f1 and f2. This allowed us to design an experiment to test indirectly our hypothesis, through a KO-rescue experiment. Homozygous loss of function mutations in SPS1 results in lethality at third instar larvae and flies present very reduced and abnormal imaginal discs ([Alsina et al., 1998], see figure 6E). Defining f2 the molecular function of drosophila SPS1, we can test whether a certain protein possess f2 by expressing this protein in the mutant background. We tested 3 different proteins: human SPS1, SPSgly isoform of the *Ciona* SPS-ae, and SPS1-rt from *Atta cephalotes* (ant - hymenopteran) using arm-Gal4 as a driver (M&M and Supplementary material S6). We observed that size and morphology of imaginal discs were considerably recovered in the case of *Ciona* (Fig. 6B; 90% of the cases showed this phenotype). A very slight recovery of size was detected for human and *Atta*, but only in around 5% of the cases (Fig 6C,D). Although species-specific signals and/or partners may impair the appropriate task of the transgenic constructs, we believe that our experiments indicate that all these SPS1 proteins have the molecular function f2, which is distinct from selenophosphate synthesis.

3.5.3.9 Readthrough as a tool of function duplication

In the hypothesis of parallel subfunctionalization, we can now explain the observed evolutionary path of metazoan SPS, even in hymenoptera. A key point is the presence of SRE elements in some selenoprotein genes (including SPS2), which has important consequences. While the standard selenoprotein transcript possesses a 3'UTR with a SECIS element, we can easily imagine that truncated forms are also produced, either by mistake or by design (unefficient transcription, alternative poly-adenylation sites, etc.). If no SECIS elements are present in the transcript at the time of translation, no Sec insertion will take place, but the SRE may still promote a Sec-independent readthrough [Howard et al., 2005]. Thus, protein isoforms with another amino acid instead of Sec (or no amino acid at all in this position) are expected to be produced in the cell. When eukaryotic SPS2 was first described [Guimarães et al., 1996], authors showed that the 3'UTR (where the SECIS resides) was necessary for the production of good yields of full length mouse SPS2 in COS-7 cells, and it was essential for incorporation of selenium. Anyway, now the same data (figure 4 in [Guimarães et al., 1996]) can be seen as an indication that Sec-lacking full length forms of SPS2 are produced from constructs lacking the 3'UTR, although with much less efficiency. Such alternative isoforms could be raw material to selection: if one acquires an useful function, its production will be increased and conserved. A single gene then carries two functions, similarly to what happens when a new splicing isoforms is created. In presence of a double function, duplication and subfunctionalization is a possible outcome. We believe that this is what happened in metazoa (see Figure 3). Initially, the ancestral SPS2

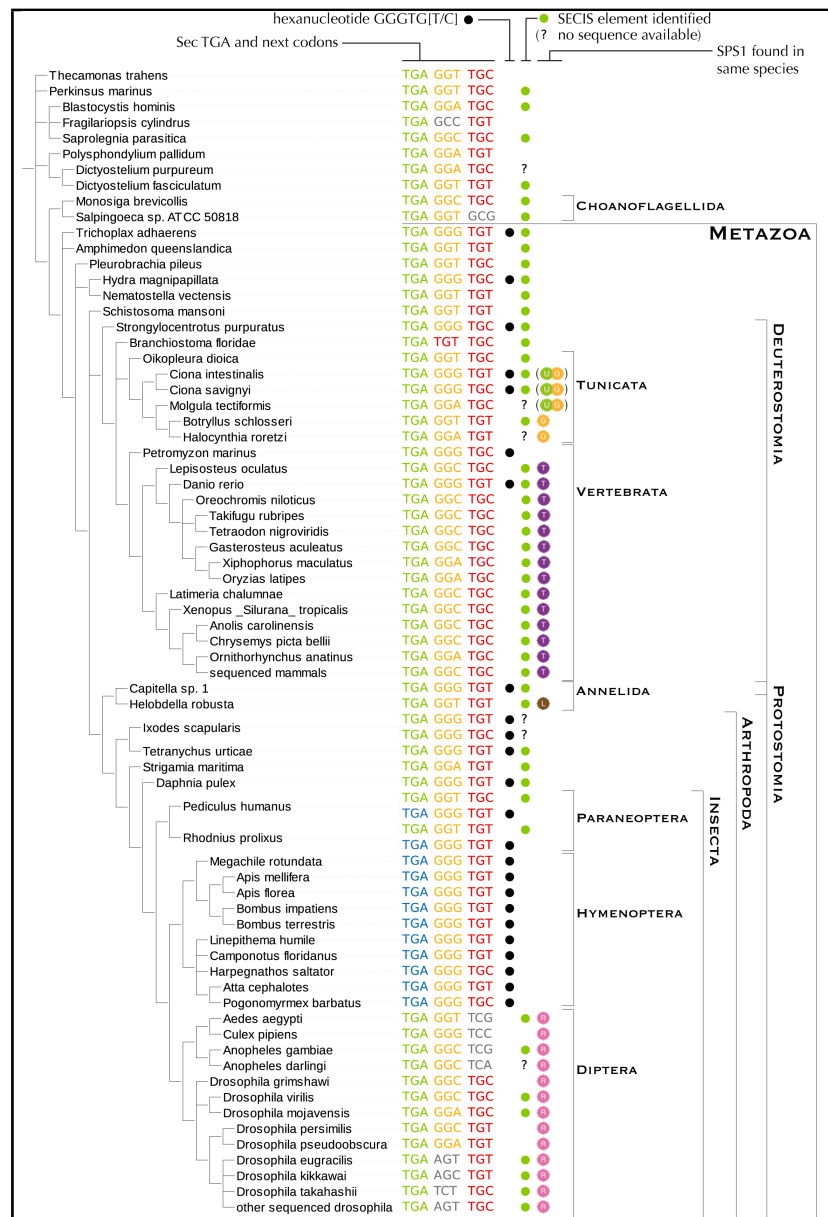


Figure 5: Readthrough enhancing hexanucleotide in SPS genes. This phylogenetic tree of investigated species (on the left) show the nucleotide alignment at the UGA site in SPS sequences. Only SPS2 and SPS1-rt genes are shown here (see full plot in figure SM5.3). The codons are colored according to their translation, following the same color schema used for figure 2 and 4 (grey for other amino acids). The presence of the hexanucleotide described in [Harrell et al., 2002] is highlighted with a black dot. Green dots mark the genes for which a bona-fide SECIS element was identified. The last column indicates the presence of SPS1 proteins in the same genome.

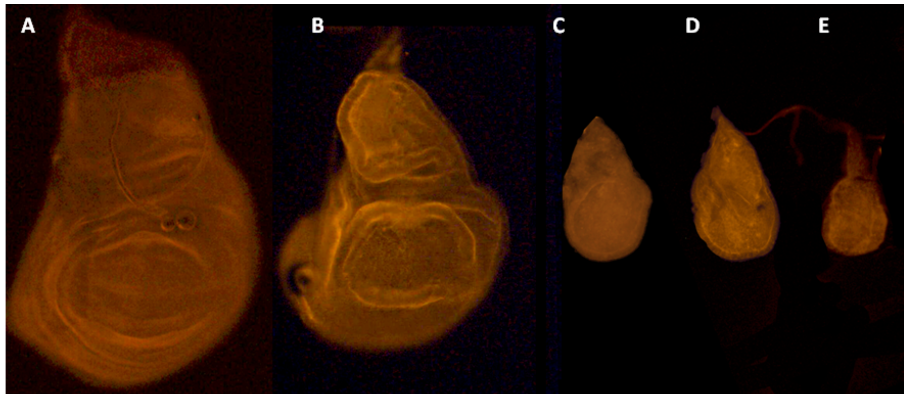


Figure 6: Rescue of drosophila SPS1 knock out (ptuf) by heterologous SPS1 proteins. Transgenic flies were obtained as described in Supplementary Material S6. A. ptuf/CyO arm Gal4 wing imaginal disc. B ptuf /ptuf ; arm Gal4/ UAS ciona Sps1 wing imaginal disc. C ptuf /ptuf ; arm Gal4/ UAS atta Sps1 wing imaginal disc. D ptuf /ptuf ; arm Gal4/ UAS human Sps1 wing imaginal disc. E ptuf /ptuf ; arm Gal4 wing imaginal disc.

assumed an additional function f2, carried out by a Sec lacking protein produced by UGA readthrough during translation. This was enhanced by the appearance of the hexanucleotide, and possibly of a third stem upstream of the existing SRE structure. The gene was then duplicated in insects and a copy lost the SECIS element, specializing in the non-Sec readthrough to perform f2 (becoming SPS1-rt). Hymenoptera then lost SPS2 as the selenocysteine trait disappeared from the genome, but still conserve SPS1-rt. Paraneoptera with selenoproteins (*Rhodnius prolixus* and *Pediculus humanus*) still maintain the two genes. In the pea aphid and in non-hymenopteran Endopterygota, the SPS1-rt gene mutated the UGA to an arginine codon, becoming the "standard" gene that we know today as drosophila SPS1.

3.5.3.10 Stem-loops structures evolution

Collectively, SPS genes bear all known secondary structures peculiar of selenoprotein transcripts: bSECIS (bacterial), SECIS (eukaryotic) and SRE - see Figure SM5.1 and SM5.2. SECIS elements are the main signals for UGA-to-Sec decoding. The bacterial Sec insertion system is different from its eukaryotic one counterpart for the SECIS structure, and also for its position: in the 3'UTR in eukaryotes, just downstream of the Sec-UGA (within the coding sequence) in bacteria. bSECIS elements are read by SelB, a Sec-specific elongation factor with a specialized N-terminal domain. In eukaryotes SECIS elements are bound by protein SBP2, that recognizes specific structural features mainly around its kink-turn core (see [Krol, 2002]). In all examined cases (data not shown), SECIS elements degenerate beyond recognition power when a selenoprotein gene is converted to cysteine, or it is copied to a non-Sec homologue. In contrast to SECIS elements, SRE are found

only in some selenoprotein genes, [Howard et al., 2005]. Due to their nature of stem-loop RNA structures and to their position, we believe that at least some of the SRE descend from bSECIS elements. This is evident for SelD/SPS2, conserved from bacteria to eukaryotes most likely through a Sec containing ancestry. It is thought that the role of the SRE elements is to facilitate Sec insertion. These secondary structures halt or slow down termination, probably hampering the access of termination factors to the translating ribosome. This is something that bacterial SECISes also have to do. Tracing a path to the present day honey bee from its last common ancestor with prokaryotes, we can follow the SPS history from the perspective of its ancestral bSECIS. When the archaeal and eukaryotic Sec insertion system took over, its function was "downgraded" to helper for Sec insertion. Then in ancestral metazoans, it was kept under selection to allow both Sec-insertion and readthrough, when a non-Sec isoform (given by its intrinsic readthrough ability) acquired a useful function f2. Finally, after a gene duplication at the root of insects, the structure in one gene copy (SPS1-rt) specialized only for non-Sec readthrough, becoming what here we named hymenopteran readthrough element (HRE).

3.5.3.11 Neo and subfunctionalization, alternative splicing and gene duplication

The phylogeny of metazoan SPS provides a snapshot of protein evolution. It can be seen as the history of the novel function f2 from its birth to its propagation, resulting in most cases in its relocation from its native gene to a new one. Remarkably, we could follow the history of f2 despite the fact that it was, and remains, uncharacterized at the molecular level. This was possible for the peculiar features of selenocysteine, found almost uniquely in catalytic sites, and functionally replaceable only (and only partially [Castellano et al., 2009]) with cysteine. Sequence homologues that do not carry cysteine aligned to Sec are extremely rare, and even in absence of other indications, prompt for distinct molecular functions. Besides this, our functional deductions owe to the abundant availability of insects genomes, and to the previous work characterizing their selenoprotein extinction process [Chapple and Guigó, 2008]. Also, SPS2 is the only Sec machinery factor that is a selenoprotein itself. All these factors make SPS peculiar, in that they provide a solid functional model underlying the hypotheses and conclusions here presented. Among the many transformations of the SPS gene in animals, the path in ascidians is probably the clearest. We observed how the SPS2 gene acquired a secondary transcript isoform to perform f2, by alternative exon usage at the 5'. Then one of the two isoforms was "detached" from the native gene, when a novel gene was generated by retrotransposition specifically in Stelidae and Pyuridae. Alternative splicing and gene duplication are considered the main contributors to protein diversity, and exhibit inverse correlation at the genomic scale [Talavera et al., 2007]. This and other data has been taken as an indication of the essential equivalence of the two processes. Considering the phenomena of retrotransposition, it is obvious that alternative splicing isoforms can be a base for gene duplication.

Nonetheless, to date there are just a very few cases in which there is a clear correspondence between alternative splicing forms in one species and gene copies in another species: eukaryotic splicing factor U2AF35 in vertebrates [Pacheco et al., 2004], Pax6 in drosophila [Dominguez et al., 2004] and mitf in fish [Altschmied et al., 2002]. This work places ascidian SPS1 among them.

3.5.3.12 Thoughts and speculations on f2

The tridimensional structure of SPS1 and SPS2 have been solved. The two proteins possess a very similar fold, and they share most domains. Although SPS1 function is unlikely to be directly related to selenium, the molecular mechanism of reaction should be very similar to that of SPS2. It is very likely that ATP is still consumed to AMP, and a substrate is phosphorylated. In the scenario in which f2 first arose in a readthrough isoform, this substrate was probably already processed by SPS2, and with an efficiency necessarily higher for the readthrough, Sec-lacking isoform than for the standard, Sec-containing SPS2 protein. In this paper, we always used the term function to refer to a molecular reaction catalyzed by a protein, selected by evolution (molecular function). Nonetheless, the same molecular function can be used for very different biological processes: for example the same reaction can generate signalling cascades with totally different outcomes in different cell types. Therefore it is plausible (although unlikely) that despite catalyzing the same reaction, some SPS1 proteins have a different global biological function. The function f2 appears to be very important in insects: knocking out SPS1 in *Drosophila* causes larval lethality [Alsina et al., 1998], while the SPS2 KO has little or no effect in laboratory conditions (Flybase phenotypic data [Marygold et al., 2013] from [Bellen et al., 2004]). This is also reflected in the tight conservation of the SPS1 sequence within all insects that possess it, while SPS2 shows a high degree of divergence (Supplementary Data S3). Other support to the fact that f2 is essential in insects is that we do not observe any insect that lost SPS2 (and selenoproteins) without generating SPS1 first, that is to say, transferring f2 to a non-selenoprotein gene. In *D.melanogaster*, flies lacking SPS1 arrest development during marginal disc formation, with cells accumulating ROS and entering apoptosis [Alsina et al., 1998, 1999]. Heterozygotes for a SPS1 knockout mutation are hypersensitive to oxidative stress [Morey et al., 2003b]. In genetic mosaics (that allow to pass the critical disc formation phase), we can see that the lack of SPS1 causes also aberrant eyes [Morey et al., 2003a]. The effects are mediated by the caspase-dependent p53/reaper apoptotic pathway, since they can be rescued by DIAP1 overexpression [Morey et al., 2003a]. Recently, drosophila SPS1 was suggested to regulate vitamin B6 synthesis [Lee et al., 2011b], since its knockdown decreases intracellular pyridoxal phosphate (its active form) and causes a transcriptional shift specifically in genes involved in this pathway.

Some of the experiments on vertebrate SPS1 revealed common themes. Human SPS1 overexpression was associated with an enhanced expression of certain redox enzymes and a decrease of reactive oxygen species (ROS), and also with an

enhancement of radiosensitivity mediated by p53 [Chung et al., 2006]. In another work [Kim et al., 2010], alternative splice variants of human SPS1 were characterized and quantified in synchronized cells. Each alternative form was regulated during cell cycle, and the expression level of the major type gradually increased until G2/M phase and then decreased. Summarizing, SPS1 appears to be linked to oxidative stress, apoptosis, cell cycle, and vitamin B6 metabolism. Since its molecular function remains unknown, we cannot predict in which of these processes SPS1 is primarily involved, and which instead are affected only indirectly.

3.5.4 Conclusions

In this study we traced the genomic evolution of SelD/SPS, ancestral selenoprotein shared by prokaryotes and eukaryotes. As this selenoprotein is itself part of the Sec machinery, its history is tightly entangled with that of selenocysteine, and thus of all other selenoproteins. SPS was found in 27-35% of sequenced prokaryotes, either as selenoprotein or (in 80% of cases) as cysteine homologue. Frequent Sec-to-Cys conversions were observed, and the well supported Cys-to-Sec conversion was identified in *Pasteurellaceae* (Gammaproteobacteria). In general, SelD/SPS2 makes a good marker for selenocysteine coding ability. Exceptions are found in prokaryotes, where SeP is used also for selenouridine in tRNAs and as cofactor to molybdenum-containing hydroxylases [Romero et al., 2005; Zhang et al., 2008; Haft and Self, 2008; Srivastava et al., 2011]. Both in lower eukaryotes and in animals, SPS2 was found only in genomes in which selenoproteins have been predicted. Insects were particularly informative, as we could observe SPS2 lost specifically the lineages going through a complete selenoprotein extinction. For these reasons, the phylogeny of SPS provides a phylogenetic map of selenium utilization across the sequenced tree of life, unprecedented for completeness and resolution (see figure 1, figure 2, supplementary figure SM1.1). In metazoa, the SPS phylogeny also provides a nice snapshot of protein function evolution. We argue that ancestral metazoan SPS2 acquired an additional function f2, presumably exercised by a non-Sec readthrough isoform. In time, this lead to an impressive variety of genomic events all leading to protein duplication across parallel lineages, driven by the subfunctionalization of the ancestral gene. Gene duplications occurred in vertebrates, insects, Clitellata (annelid). In ascidians a new Gly isoform emerged on the same gene, by alternative exon usage at the 5'. Then, in the Styelidae and Pyuridae (ascidians including *Botryllus*), the Sec form retrotransposed to the genome, originating a new gene. The parental gene lost its SECIS element and TGA containing exon, specializing in the Gly form only. The stem-loop structure embedded in the coding sequence of SPS genes played a key role in this process. By diversificating the translation products of this gene, this secondary structure allowed the birth of a novel protein function, then propagated to regular genes. This underlines the importance of readthrough as tool of neofunctionalization.

3.5.5 Methods

A large collection of prokaryotic and eukaryotic genomes were searched for the SPS family using Selenoprofiles ver 3.0 [Mariotti and Guigó, 2010]. The same program was used with a wide collection of selenoprotein families to probe the number of selenoprotein per lineage, as those displayed in Figure 1 and 2. Ciliates were manually curated, for their different genetic code. In addition to genomes, the ncbi EST database was also used to investigate certain eukaryotic lineages of interest (see Supplementary Material S3 and S4). tRNAscan [Lowe and Eddy, 1997] and Aragorn [Laslett and Canback, 2004] were used to search for tRNAsec search. For prokaryotes, a subset of species was selected to build a reference set of predictions, which were inspected and filtered to exclude duplicates, pseudogenes and contaminations of the genome assemblies. The plots on the full sets of species are available in the Supplementary Material sections. Alignments were computed using t-coffee [Notredame et al., 2000] and mafft [Katoh et al., 2005]. To deduce the phylogenetic history of SPS we used a variety of approaches: maximum likelihood reconstruction of protein phylogeny (as explained in [Mariotti et al., 2012] after [Huerta-Cepas et al., 2011]), mapping to a species tree, intron structure analysis. Figure 1 and 2 were generated with the script sunburst (DS, personal communication). All other tree-based plots were generated using ete2 [Huerta-Cepas et al., 2010]. The approximate phylogenetic tree of investigated species was derived from the ncbi taxonomy database [Sayers et al., 2009]. The history presented is the product of reasoning the data mainly using parsimony as main principle. Supplementary Material S1-S5 contain a detailed description of the process. Supplementary Material S6 details the rescue experiments in *Drosophila*.

3.5.6 Supplementary Material

Find next the following supplementary sections:

- S1: Seld in prokaryotes
- S2: Gene fusions and extensions
- S3: Phylogeny of eukaryotic SPS proteins
- S4: Alternative isoforms split by gene duplication in ascidians
- S5: Secondary structures within coding sequences of SPS genes
- S6: Rescue experiments in *Drosophila*

Supplementary Material S1

SelD in prokaryotes

SelD gene finding in sequenced prokaryotes

We downloaded a total of 8286 prokaryotic genomes from NCBI, including 54 archaeal genomes. We scanned them with the program Selenoprofiles (Mariotti 2010, <http://big.crg.cat/services/selenoprofiles>) using two SPS-family profiles, one prokaryotic (*seld*) and one mixed eukaryotic-prokaryotic (*SPS*). Selenoprofiles removes overlapping predictions from different profiles, keeping only the prediction from the profile that seems closer to the candidate sequence. As expected, the great majority of output predictions in prokaryotic genomes were from the *seld* profile. We will refer to the prokaryotic SPS/SelD genes as SelD, following the most common nomenclature in literature.

To be able to inspect results by hand, and also to focus on good-quality genomes, we considered a reduced set of species. We took the `prok_reference_genomes.txt` list from ftp://ftp.ncbi.nlm.nih.gov/genomes/GENOME_REPORTS/, which NCBI claims to be a "small curated subset of really good and scientifically important prokaryotic genomes". We named this the prokaryotic reference set (223 species - see Supplementary Material S7). We manually curated most of the analysis in this set, while we kept automatized the analysis on the full set.

We detected SelD proteins in 60 genomes (27%) in the prokaryotic reference set, which become 2908 (35%) when considering the prokaryotic full set. The difference in proportion between the two sets is due largely to the presence of genomes of very close strains in the full set, which we consider redundant. The *Escherichia* genera in particular constitutes alone more than 7% of the genomes in the full set, and since SelD and selenoproteins are present in this genera, this inflates the proportion of species with SelD.

Generally, a single SelD protein (or none) was detected in each genome, with only a few exceptions of multiple genes (just 2 in the reference set). Only a minority of detected SelD contained selenocysteine (20% in the reference set), with the rest carrying a cysteine instead.

No homologues with a different amino acid in this position were detected in the prokaryotic reference set, and there were no predictions unaligned in the Sec position.

Searching other markers for selenium utilization traits

Figure 1 in the main paper shows the presence of Sec and Cys SelD proteins in the reference set of species, as a phylogenetic sunburst. As you can see, SelD presence is mostly scattered, highlighting a very dynamic process acting on selenium utilization traits. To link SelD to its functional network, we searched our collection of genomes also for other selenium trait markers: **tRNA^{sec}** and **SelA** (SelenoCysteine synthase/L-seryl-tRNA(Sec) selenium transferase) and for the selenocysteine trait, and **ybbB** (tRNA 2-selenouridine synthase), for the selenouridine trait. We also predicted the selenoproteins encoded in each genome, to have an estimation of the selenoproteome size. All these searches were carried out using the program Selenoprofiles, building profile alignments on purpose when necessary. The only exception was tRNA^{sec}: for this, we ran the programs tRNAscan-SE (Lowe 1997) and Aragorn (Laslett 2004). Both programs are thought for predictions of all

tRNAs. Considering only tRNA^{sec}, Aragorn appeared to be more sensitive, but less specific than tRNA^{scan}-SE, but really none of the programs gave satisfactory results, mainly for the presence of false positives in many lineages. To have a reliable set of tRNA^{sec} annotation we thus restricted our search to the reference prokaryotic set, and we manually inspected and filtered the predictions. We simply excluded all tRNA^{sec} candidates lacking the characteristic extra arm (Palioura 2009). We believe most of these false positives constitute real tRNAs with a UCA anticodon that can read UGA, but which do not load selenocysteine. Such tRNA predictions were present for example in all Mycoplasmas, which are known to use UGA for tryptophan. With this filtering, 43 out of 45 species with a SelA prediction possessed tRNA^{sec}. 2 species were predicted to possess tRNA^{sec} but not SelA: *Rhodospirillum rubrum* ATCC 11170 (Alphaproteobacteria, Rhodospirillales) and *Cupriavidus necator* N-1 (Betaproteobacteria, Burkholderiales).

As said, the majority of investigated prokaryotic species do not possess SelD, and thus are expected unable to produce selenoproteins and selenouridine containing tRNAs.

Considering the continuity of the Se traits as the basic scenario (although punctual horizontal transfers have certainly occurred), this means that multiple losses of Se-traits happened along the prokaryotic tree. The selenocysteine and selenouridine trait were found to have a good overlap, with 26 species in the reference set possessing both ybbB and SelA. Selenocysteine appears to be more common than selenouridine: 20 species were found to possess SelA but not ybbB, while only 12 species possessed ybbB but not SelA. With few exceptions that can probably be ascribed to genome assembly uncertainty, or limitations of search methods, selenoproteins were predicted only in species with SelA and tRNA^{sec}, regardless of ybbB presence, as expected (see Figure 1).

We detected at least one selenium utilization trait in all reference species with a SelD gene, with a single exception: *Enterococcus faecalis* (see below).

Se utilization in Archaea

We had 54 archaeal genomes in our full dataset (6 in the prokaryotic reference set -- see point number 1 in Figure1). SelD, SelA and selenoproteins were found only in the two lineages: Methanococcales (*Methanocaldococcus jannaschii* DSM 2661, *Methanococcus aeolicus* Nankai-3, *Methanococcus maripaludis* strains: C5, C7, S2, *Methanococcus vanniellii* SB) and Methanopyri (*Methanopyrus kandleri* AV19). All archaeal SelD forms detected were with selenocysteine. These genomes are quite rich in selenoproteins (Rother 2003). In *M. maripaludis* we identified 7 selenoproteins, 4 of which belonging to the formate dehydrogenase family (fdha), one coenzyme F420-reducing hydrogenase large subunit (frha or fruA), one HesB-like selenoprotein, plus the Sec-containing SelD. Since for our selenoproteome size estimation we prioritized specificity rather than sensitivity, additional selenoproteins missing in our annotation are expected in Archaea, as well as in other prokaryotes. This is the case for example for selenoprotein VhuD (Rother 2001).

The archaean ybbB gene is split in two genes in comparison to bacteria, one with the rhodanese domain delivering the selenium (N-terminal in bacteria), and one with a P-loop WalkerA motif (C-terminal). The genes are located adjacent, on the same strand, but with inverted positions (the C-terminal domain gene is upstream). While gene similar to the rhodanese-ybbB were found in other archaeal genomes lacking SelD, WalkerA-ybbB was found only in Methanococcales. Interestingly, it is missing in Methanopyri, which then appear to have lost selenouridine but kept selenocysteine.

Sec / Cys conversions of SelD

Sec to Cys conversions are a process peculiar to selenoprotein genes. Cysteine codons are just one point mutation away from TGA, and cysteine and selenocysteine are similar for many properties. For most selenoproteins, cysteine homologues (orthologues or paralogues) are known (Fomenko 2012). Thus, most molecular functions performed by selenoproteins can be achieved by their cysteine counterparts as well. Selenocysteines are tightly conserved in some lineages (vertebrates: Castellano 2009), therefore there must be selective constraints to keep selenocysteine rather than converting to cysteine in these lineages. Although the exchangeability of selenocysteine and cysteine is still a open debate (see Arner 2010), it is clear that differences in catalytic efficiency, substrate specificity or translation regulation may be important.

Despite the fact that Sec to Cys conversions have been widely observed (see for example Mariotti 2012), no Cys to Sec conversion is described in literature.

Nonetheless, considering that selenoprotein families of prokaryotes and eukaryotes have little overlap (Driscoll 2004), and that some eukaryotic selenoprotein families have homologues without Sec in prokaryotes, it is natural to assume that Cys to Sec conversions have indeed occurred, generating new selenoprotein families from existing protein families.

In order to identify Sec / Cys conversions, we ran our phylogenetic reconstruction pipeline (as explained in Mariotti 2012, after Huerta-Cepas 2011) obtaining phylogenetic trees of all SelD proteins predicted in prokaryotes. This data, together with the species tree annotated with predictions, allowed us to reliably trace some of these conversions events.

Dynamic evolution of Se traits in Clostridia

Clostridia are a very diverse lineage when we consider selenium utilization traits. You can see this in Figure 1 in the main paper (point number 2), or in Figure SM1.1.

Some organisms (such as *Desulfitobacterium hafniense*) possess both the selenocysteine and selenouridine trait, others (such as *Clostridium botulinum* A str. ATCC 3502) possess only the selenocysteine trait, and others again (such as *Clostridium thermocellum*) have none. Intermediate states are also sometimes found.

Within this lineage we noticed many Sec-to-Cys conversions. To investigate them in detail, we have extracted all *Clostridium* predictions from our prokaryotic full set, removing redundancy at the species level.

Figure SM1.3 and figure SM1.4 show respectively their predicted protein tree, and the species tree annotated with the predictions.

We hypothesize that the last common ancestor of this lineage had a Sec-SelD gene, and this was converted to a cysteine homologue many times independently in various lineages. Interestingly some of these conversions must be very recent, as for example some strains of Lachnospiraceae were found with Sec-SelD, and others with Cys-SelD.

We believe that the scattered presence of SelD proteins across all sequenced prokaryotes is the product of the same process we observe in Clostridia, with frequent Sec to Cys conversions from an ancestral Sec-SelD form, and also frequent gene losses (concomitant with the loss of Se traits).

Selenocysteine losses in Bacilli

Bacilli constitutes a well studied bacterial lineage (including among others *Staphylococcus*, *Streptococcus* and *Enterococcus*) that together with Clostridia forms the phylum of Firmicutes. Most Bacilli appear to lack SelD, and thus the selenium utilization traits. In fact, if we consider just the prokaryotic reference set (Figure 1), there are only three species

with SelD: *Bacillus coagulans* and *Paenibacillus mucilaginosus*, both possessing the selenouridine trait, and *Enterococcus faecalis*, that do not possess neither SelA, ybbB, or the Sec-tRNA. The presence of an “orphan” SelD gene in this species has been previously noted (Zhang2008, Haft2008), and may be explained by the use of Se as cofactor to molybdenum hydroxylases (Srivastava2011).

When we increase the number of considered species, thus increasing the resolution (see Figure SM1.1), we notice that not all Bacilli lost selenocysteine. There are several species, phylogenetically scattered, that possess either the selenocysteine trait, the selenouridine trait, or even both. The genus *Bacillus*, *Paenibacillus* and *Lactobacillus* exhibit such diversity, roughly analogous to the situation described for Clostridia.

Using the full set of 8286 prokaryotic species, only a few families of Bacilli show no presence of SelD at all: Leuconostocaceae, Listeriaceae, Staphylococcaceae.

The case of Streptococcaceae is bizarre, and interesting. In our full prokaryotic set we have 873 genomes belonging to this lineage, and we found SelD in a single species: *Streptococcus sobrinus* TCI-157. This is also the only Streptococcaceae species with any other Se marker: a bona fide ybbB gene was identified.

This suggests that this species really possesses and utilizes SelD to produce selenouridine containing tRNAs, and that this feature is extremely rare (if not unique) in this family. There are two possible explanations: either selenouridine (SelD + ybbB) was lost independently in the lineages coming out from the Streptococcaceae radiation, and was kept only in this one (extremely unlikely), or most probably it was lost at the root of this family, and re-acquired just in this species by horizontal transfer.

Running blastp using SelD and ybbB from *S. sobrinus* TCI-157, we see that the most similar proteins annotated are from the genus of *Paenibacillus* or *Bacillus*, which thus are the most likely sources of horizontal transfer.

Se traits in Proteobacteria

Proteobacteria are a major group of Bacteria that contains many lineages of interest, as for example *Escherichia*, *Salmonella*, *Burkholderia* and *Campylobacter*. Proteobacteria constitutes the most represented phylum in our datasets, constituting 44-47% of the total number of species. The sequenced species belong to the five major classes of alpha, beta, gamma, delta and epsilon proteobacteria. One zetaproteobacteria species was also present in our full dataset (*Mariprofundus ferrooxydans* PV-1); it appears to lack SelD as well as selenoproteins and any other Se marker.

Alphaproteobacteria

As for other cases already mentioned, increasing the resolution reveals a more complex pattern in Alphaproteobacteria: compare Figure 1, generated using the reference set, with Figure SM1.1, generated using the full species set. Selenium utilization remains quite uncommon, but scattered through most Alphaproteobacteria sublineages.

The order of Rhodobacterales shows the highest diversity, with species having the SeC trait, SeU trait, both or none. In the rest of the phylum, selenium traits are much less common. Selenouridine was found only scattered through Caulobacterales, and selenocysteine only in the orders of Rhizobiales and Rhodospirillales.

Betaproteobacteria

SeC and SeU are more common in Betaproteobacteria, although still exhibiting a diversified pattern that testifies the dynamic process acting on these traits. Most Burkholderiales sublineages possess SelD and at least one complete Se trait. The genus of *Burkholderia* itself shows a recent (if not present) dynamic evolution, with closely related species that differ for the presence of Se utilization traits.

Within the order of Neisseriales, SelD and Se traits (both SeC and SeU) are found only in few species in our dataset (*Laribacter hongkongensis* HLHK9, *Chromobacterium* sp. C-61, *Pseudogulbenkiania ferrooxidans* and *Pseudogulbenkiania* sp. NH8B).

Gammaproteobacteria: a Cys to Sec conversion in Pasteurellales

Gammaproteobacteria are a class of bacteria that contains many important human pathogens, including among others the genus *Escherichia*, *Salmonella* and *Pseudomonas*. This class is well represented in our sequence datasets, with 49 species in our reference set, 2545 in our full set (best represented proteobacteria order). SelD proteins were detected in the majority of Gammaproteobacteria (57% of species in reference set, 65% in full set).

The SeC trait was identified in the vast majority of Enterobacteriales (including *Escherichia*, *Yersinia*, *Salmonella*, *Shigella*, *Enterobacter*). SeU is also found in the same lineages, with the only notable exception of the *Yersinia* genus, that apparently lost SeU but kept SeC.

The majority of species in the family of Pseudomonadaceae (including *Pseudomonas*) possess SelD, with either both SeC and SeU, or just SeU, apparently important for this lineage. In contrast, its sister family Moraxellaceae exhibits no SelD, no ybbB, no SelA and no selenoprotein prediction, indicating a complete loss of known Se utilization pathways. SelD is quite uncommon in the order of Xanthomonadales, where it was found only among *Stenotrophomonas*, and also in the species *Wohlfahrtiimonas chitiniclastica* SH041 and *Dyella japonica* A8.

Intermediate states were found in the orders of Alteromonadales and Oceanospirillales, both exhibiting a diversified, scattered pattern with species possessing mostly SeU, both SeU and SeC, or none.

We were surprised to see a very low number of Sec containing SelD proteins in Gammaproteobacteria (7% of total). Most of them were found in the family of *Pasteurellales*, where the majority of SelD are with Sec, although some Cys-SelD were also identified (e.g. *Gallibacterium anatis* UMN179). Then, the rest of Gammaproteobacteria Sec-SelD were found only in very narrow lineages: in some *Photobacteria* (Vibrionales), in species *Allochromatium vinosum* (Chromatiales), and in species *Wohlfahrtiimonas chitiniclastica* (Xanthomonadales).

Given the rich sampled diversity with the Gammaproteobacteria genomes, and the extremely low number of Sec-SelD forms, it is natural to think that their last common ancestor contained a single Cys-SelD gene.

Thus, Sec-SelD proteins may have arisen in the lineages mentioned above by one of two possible mechanisms: horizontal gene transfer (HGT) of a Sec-SelD, or conversion of Cys-SelD to selenocysteine.

To investigate this, we extracted all Gammaproteobacteria SelD genes from our full set of predictions. We then reduced the set by removing sequence redundancy, that is to say, keeping only one representative for each cluster of almost identical (>95%) protein sequences. In this process, we took care that no Sec protein was dropped in favor of a Cys containing representative. We then ran our phylogenetic reconstruction pipeline on this protein dataset. Figure SM1.5 shows the predicted protein tree topology.

Additionally, to control for HGT, we have ran blastp for each Gammaproteobacteria Sec-SelD, to search for the closest related sequences outside its taxonomic order. So for example we have run the Sec-SelD of *Photobacterium profundum* SS9 against the whole set of annotated proteins (nr), excluding those belonging to any Vibrionales. Below, we report our conclusions.

The Sec-SelD proteins found in some *Photobacteria* (*P. profundum* SS9, *P. profundum* 3TCK, *P. sp.* AK15) appear to be product of horizontal transfer. In fact, the most similar proteins annotated in nr belong to very distant species (Firmicutes, or Chloroflexi). Most notably, these Sec-SelD do not cluster with the rest of Vibrionales sequences (see figure SM1.5), falling very far from the (Cys-containing) SelD found in other *Photobacteria*. *Allochromatium vinosum* Sec-SelD most probably comes from another horizontal transfer. The most similar sequences returned by blastp all belong to the lineages of Firmicutes. Actually, all Chromatiales SelD sequences do not cluster together, but rather form small genus specific clusters, which suggests that even the Cys-forms may have been acquired by multiple horizontal transfers.

Wohlfahrtiimonas chitiniclastica also appears to have acquired Sec-SelD by horizontal transfer. While the rest of (Cys-containing) SelD sequences nicely cluster together in the protein phylogeny (figure SM1.5), *W. chitiniclastica* Sec-SelD clusters instead with Pseudomonadales sequences. Blastp also returns proteins from those lineages as the most similar to this query. There is an apparent paradox with this: we could not find any Sec-containing SelD in Pseudomonadales, only Cys forms, despite a good representativity in our dataset. This means that either 1. the source of the horizontal transfer is a species from a unknown, Sec-SelD containing bacterial lineage which is not sequenced yet, whose closest relative in our datasets is Pseudomonadales, or 2. the original SelD gene transferred was with cysteine, and was converted to Sec during, or shortly after, the transfer.

Finally, we think that Pasteurellales acquired Sec-SelD by a cysteine to selenocysteine conversion. In fact, all their SelD protein sequences (both Sec and Cys containing, found in different species) form a unique similarity cluster (see figure SM1.5). The most similar sequences found in other taxonomic orders (both by blastp and in our protein tree) are from Enterobacteriales, the closest related order to Pasteurellales (Gao2009). Thus, the most likely scenario involves a Cys to Sec conversion in the SelD gene in the last common ancestor of Pasteurellales. Then, the codon switched back to cysteine in several lineages independently (e.g. *Haemophilus parasuis*).

Concluding, we found in Pasteurellales the first well supported Cys to Sec conversion ever documented. In one such event, it is of key importance that a functional bacterial SECIS element is established at the time of the mutation that originates the TGA. In this case, this was probably favored by the biased sequence composition of this gene region, for it had already contained a bacterial SECIS once (parsimoniously, we assume the presence of a Sec-SelD gene in the last universal common ancestor).

Deltaproteobacteria are selenoprotein rich

The majority of Deltaproteobacteria were predicted to possess a Sec-SelD gene, a complete SeC machinery, and plenty of selenoproteins. Species *Desulfobacterium autotrophicum* HRM2 exhibited the largest predicted selenoproteome among prokaryotes: we found 31 selenoprotein genes, belonging to 18 distinct protein families.

Some Deltaproteobacteria appeared to possess both the SeC and SeU traits (e.g. *Geobacter*). Only a few possessed SeU but not SeC (e.g. *Bdellovibrio*).

Epsilonbacteria

Sequenced Epsilonbacteria belong mainly to two families: Campylobacteraceae and Helicobacteraceae. The former appear to possess both SeC and SeU, and several selenoproteins were predicted in their genome. In contrast, we found two distinct situations for Helicobacteraceae. Certain species possess a complete SeC machinery, with also a few selenoproteins predicted in their genome, and can either have also SeU (e.g. *Helicobacter pullorum*) or not (e.g. *Helicobacter hepaticus*).

The rest of species, which actually form the majority of Helicobacteraceae (including *Helicobacter pylori*), are predicted to possess no selenoprotein and no SelD. Surprisingly, for most of them we predicted a SelA gene in the genome. Given the absence of SelD and of predicted selenoproteins in these genomes, we think that this protein may have been readapted to a different function.

Actinobacteria

This class of bacteria also exhibited a highly scattered pattern of Se traits, testifying a very dynamic evolution. The gene ybbB was not found in any species in this lineage, and therefore we expect SeU not to be utilized. SelD was found only in ~19% of species in our full dataset, scattered through sublineages (see figure SM1.1); 86% of these species possessed SelA, and 94% had at least one selenoprotein predicted in the genome. So, it appears that this pattern is the product of a real process of SeC loss acting on parallel lineages. The genus *Mycobacterium* showed a remarkable diversity in this, with only ~20% of these species possessing SelD and selenoproteins.

On a total of 140 SelD proteins predicted in Actinobacteria, only 11 carried selenocysteine. All Sec-SelD were found within the order of Coriobacteriales, with the exception of species *Kineosphaera limosa* NBRC 100340 and *Rubrobacter xylanophilus* DSM 9941.

Other bacterial lineages

We here provide a short report on the rest of prokaryotic taxonomic classes present in our dataset.

Cyanobacteria appear to have lost SeC: SelA was not found in any of these genomes, and a very limited number of selenoprotein genes were predicted. At manual inspection, most of them appeared to be false positives. Nonetheless, ~39% of Cyanobacteria were predicted to possess SelD (always with Cys). ybbB was identified in 92% of the SelD containing species, indicating that some Cyanobacteria retained SelD to produce SeU-containing tRNAs.

Bacteroidetes exhibit a similar pattern, with few species conserving SelD as part of the SeU trait. SelA is not found in any genome, with the only exception of *Chryseobacterium taeanense*, which carries a gene almost identical to SelA as found in the Betaproteobacteria genus *Delftia*. Interestingly a Sec containing formate-dehydrogenase was found in the same genome. This potentially supports a second acquisition of the SeC trait in *C. taeanense* by horizontal transfer; nonetheless, given that we observe this in a single genome, we cannot exclude that the genes are actually from a contamination introduced in the sequencing process.

Spirochaetes show a scarce presence of Se traits. Using the reference set (figure 1) this lineage appeared to completely lack SelD, but with more resolution (figure SM1.1) we can notice this is not the case. SelD was found in a limited number of species (e.g. *Brachyspira pilosicoli*) apparently to produce selenocysteine. Sec-SelD genes were also detected, uniquely in the genus *Treponema*.

Lastly, Chlamydiae were found devoid of SelD, SelA and ybbB, indicating a complete loss of Se utilization. Tenericutes (including Mycoplasmas) are also predicted to lack all Se traits.

Figures in Supplementary Material S1:

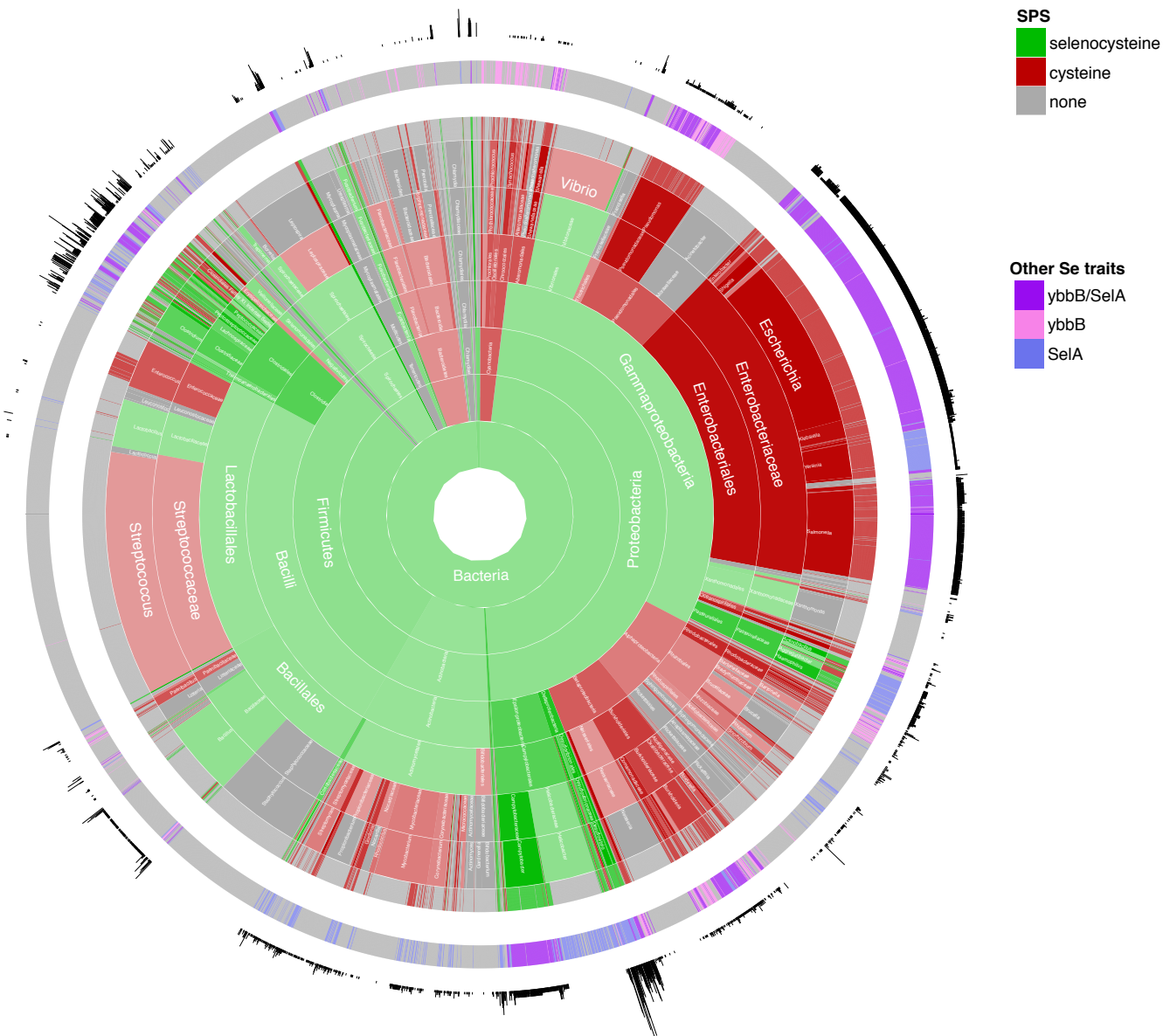


Figure SM1.1:

Sunburst tree of SelD and other Se-trait markers in the full set of prokaryotes. See caption of Figure 1 in the main paper (reference set) for explanations.

0.69

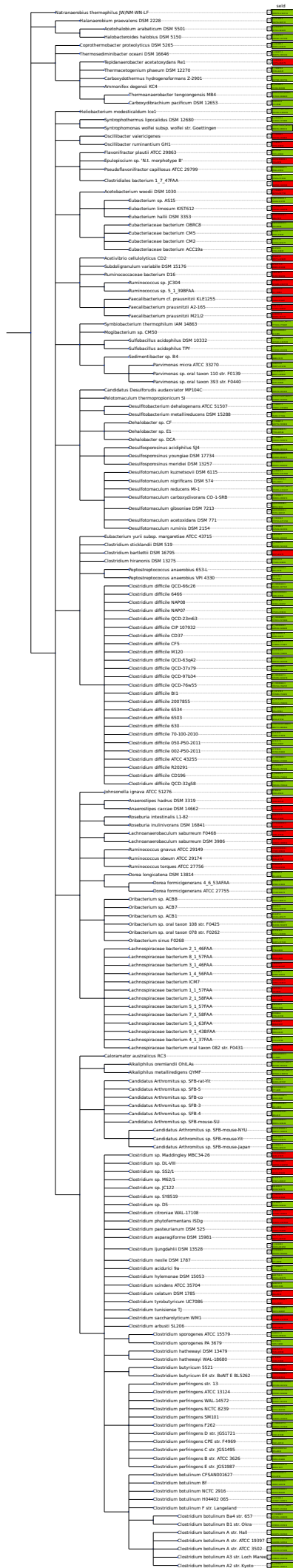


Figure SM1.2:
Reconstructed protein phylogeny of all SelD/SPS proteins in the reference prokaryotic set.



etases (SPS)

Figure SM1.3:
Reconstructed protein
phylogeny of all SeID/SPS
proteins predicted in Clostridia.

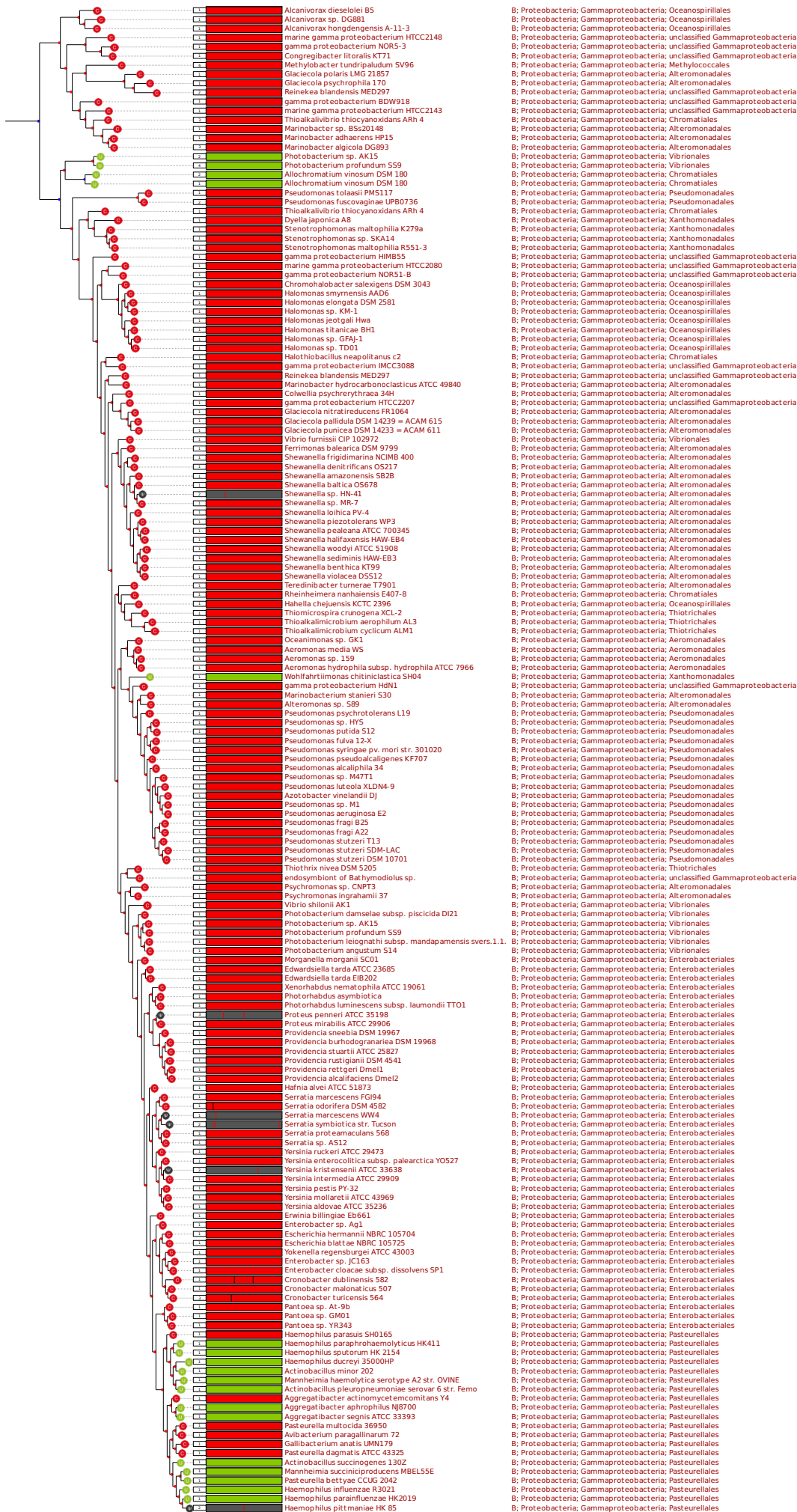


Phylogeny of Selenophosphate synthetases (SPS)

Figure SM1.4:
Clostridia species tree annotated with SeID/SPS proteins identified in each genome. Genomes with no results are not displayed here.

Figure SM1.5: (next page)
Reconstructed protein phylogeny of all SeID/SPS proteins predicted in Gammaproteobacteria. See notes in the text about Sec containing SeID genes.

Phylogeny of Selenophosphate synthetases (SPS)



(Figure
SM1.5)

Supplementary Material S2:

Gene fusions and extensions

After scanning for SPS proteins our wide collections of prokaryotic and eukaryotic genomes, we searched for possible protein extensions or fusions to other genes. Some such gene fusions were already reported for SPS: with a NADH-dehydrogenase domain in bacteria and some lower eukaryotes (Zhang 2008), with a Cys sulfinatase / NifS protein in *Geobacter sp. FRC-32* (Zhang 2008). Recently, species *Naegleria gruberi* (Da Silva 2013) was reported to possess an gene product of the fusion of a SPS protein with a methyltransferase protein. Through genetic experiments, authors show that the fused protein still performs the canonical SPS function (SeP production). The N-terminal probably possess an additional function. It is possible that this is related to a detoxification process, as authors find that this domain conferred partial resistance to selenium toxicity (see Da Silva 2013).

As said in the paper, we used the program Selenoprofiles (Mariotti 2010, <http://big.crg.cat/services/selenoprofiles>) to predict SPS genes in genomes. To detect possible extensions or gene fusions, we used two different strategies.

1) We used the two selenoprofiles methods "complete_at_three_prime" and "complete_at_five_prime" (see selenoprofiles manual) to detect long stretches of potentially coding sequence (i.e. without in-frame stop codons) at the 5' or 3' to the gene structures predicted by homology. The candidate extensions were then clustered by similarity, and run with blastp against the ncbi nr protein database. Finally, they were manually inspected.

2) To detect annotated gene fusions, we run the whole set of SPS selenoprofiles predictions with blastp (loose filters) against the ncbi nr protein database. The resulting set of matches constitutes then a good approximation of all annotated proteins with a SPS domain. We then parsed this set, to get all the blast hits with start and end indices suggesting the possible presence of additional domains in the same annotated protein. In particular, we searched for blast alignments in which the target (protein annotated in ncbi nr) contains large unaligned portions, at the 5' or at the 3'. All potentially interesting blast alignments were then manually inspected.

The candidate extensions from the two strategies were merged and manually analyzed. For the most interesting cases, a new alignment profile was built including the SPS domain and the fused domain, and used to scan again our collection of genomes. We report below a summary of results, grouped by the identity of the gene extensions.

A unique methyltransferase-SPS fusion in *Naegleria gruberi*

The *Naegleria gruberi* SPS gene fusion described in literature was detected by our procedure. The N-terminal showed homology to proteins arsenite methyltransferase, UbiE/COQ5 methyltransferase, methyltransferase type 11. The C-terminus appears to be a complete SPS gene, with a cysteine aligned to the usual Sec position.

We were surprised to find this fusion only in species *Naegleria gruberi*. Nonetheless, it must be noted that this taxonomic group (heterolobosea) is scarcely sequenced to date.

Interestingly, we noted that arsenite methyltransferases includes some selenoproteins in bacteria (see Zhang 2008), suggesting a functional link between the two domains. From experiments in (Da Silva 2013), it is most likely that this fused protein possesses an additional, rather than an alternative, function. In fact the fused protein (or even only its SPS domain) is able to complement a SelD deficiency in *Escherichia coli*. This is consistent with the identification of other selenocysteine machinery proteins in *N. gruberi* by the same authors, which advocates for the ability of *N. gruberi* to code selenocysteine. Nonetheless, a single selenoprotein was identified in (Da Silva 2013): a thioredoxin reductase with homology to mammalian TR3.

Using the selenoprotein prediction tools that we developed in the last years (Mariotti 2010, Mariotti 2013), we could predict two additional selenoproteins in this genome: a second thioredoxin reductase, and a deiodinase-like protein. Also, we found a second cys-containing SPS gene in this genome, unreported in (Da Silva 2013). To our increasing surprise, we noted that this gene is also the product of a fusion: the C-terminal has homology to SPS, while the N-terminal is homologous to NifS proteins.

NifS-SPS fusions

The Cys sulfinate desulfinate (NifS) proteins process indiscriminately cysteine or selenocysteine, producing alanine and elemental sulfur or selenium respectively (Mihara 1997). They are thus directly involved in selenium metabolism (as well as in sulfur's), and they are proposed to be a possible selenium donor for SPS proteins (Mihara 2002). Bacterial NifS proteins exhibit sequence homology to metazoan protein selenocysteine lyase, which nonetheless appear to be specifically acting only on selenocysteine. The fusion of NifS and SPS proteins was already observed in (Zhang 2008), uniquely in species *Geobacter sp. FRC-32*, a deltaproteobacteria classified among Desulfuromonales. Our procedure recovered the fused protein in *Geobacter* species. Interestingly, the SPS domain in this known fused protein contains a selenocysteine, in the usual site -- all other SPS fused are with cysteine instead. Notably, we identified an additional SPS protein in *Geobacter*, also selenocysteine containing, but without extensions.

Caldithrix abyssi is a bacterial species that seems to represent a novel lineage of its own (see Miroshnichenko 2003). In this species we found a NifS-SPS fusion, in which SPS is with cysteine. Additionally, we identified another SPS gene with selenocysteine in the same genome. This gene appears to be normal (not fused), although we couldn't find a starting Methionine. Several other selenoproteins, and Sec machinery proteins were identified in the genome, supporting the fact that this species utilizes selenocysteine. But NifS-SPS proteins are not limited to prokaryotes, since we found also in the genomes of two lower eukaryotes: the heterolobosean *Naegleria gruberi* (see above) and the amoeba *Acanthamoeba castellanii*. Both genes include a number of (small) introns, and have a cysteine aligned to the Sec position of SPS2 genes.

As previously said, we identified an additional SPS gene in the *N. gruberi* genome fused with a methyltransferase, and a few selenoproteins. In the genome of *A. castellanii* we found 5 selenoproteins, similar to the *Naegleria* set: a thioredoxin reductase, two deiodinase-like proteins, a glutathione peroxidase, and selenoprotein O. Additionally a partial N-terminal SPS sequence was found in this organism, carrying an in-frame TGA in the expected place. Due to the limited availability of sequences from this organism, it is unclear whether this represents the only sequenced fragment of a real additional SPS gene in this organism, or if it is a relic of a gene that was lost, or even if this comes from a genomic contamination.

Considering that all species with a NifS-SPS fusion possess also another copy of SPS (possibly with the exception of *A. castellanii*), it is possible that the fused protein cannot full

perform the original SPS function, for example we could expect that it can phosphorylate selenium only when NifS is the donor (when recycling selenocysteine).

NADH dehydrogenase-SPS fusions

This fusion was already observed in (Zhang 2008), and appeared to be very common within prokaryotes. Consistently we detected such fused genes in a wide range of Bacteria, including Cyanobacteria (Prochlorales, Oscillatoriothrixaceae, Stigonematales), Alphaproteobacteria (Rhodobacterales, Rhodospirillales), Gammaproteobacteria (Alteromonadales, Oceanospirillales, Methylococcales, Chromatiales), Betaproteobacteria (Burkholderiales, Nitrosomonadales).

Interestingly, this fusion was detected as well in several eukaryotic species, belonging to a number of diverse basal lineages (see Figure 2): *Ostreococcus tauri*, *Ostreococcus lucimarinus*, *Chlorella variabilis*, *Coccomyxa subellipsoidea* (all of which are green algae), *Aureococcus anophagefferens* (pelagophyte), *Phaeodactylum tricornutum* (diatom), *Ectocarpus siliculosus* (brown algae), *Emiliania huxleyi* (haptophyte), *Toxoplasma gondii* (Apicomplexan), and even the metazoan *Hydra magnipapillata* (cnidaria - hydrozoan).

Figure SM2.1 shows the sequence-based predicted phylogeny of the identified bacterial and eukaryotic NADH-dehydrogenase / SPS fusions. Most of eukaryotic NADH-SPS proteins cluster together, with two exceptions: *Toxoplasma* and *Hydra* sequences are clustering within bacterial sequences, quite far one from another and also far from the other eukaryotic fusions. This supports the idea that NADH-SPS fusions emerged more than once during evolution. Actually, the *Hydra* fused protein resembles so much the bacterial homologues that it is entirely possible that this is a genomic contamination, and the gene is actually from a bacteria. The lack of introns would support this. In *Toxoplasma* and most other eukaryotic species, the gene has introns so we can be confident that it is really integrated in the genome, and functional.

Interestingly, the phylogenetic cluster of eukaryotic sequences contain two *Rhodospirillales* (Alphaproteobacteria) sequences. This may suggest that horizontal transfer occurred.

Other gene extensions

Several other extensions of SPS genes were predicted in prokaryotes and basal eukaryotes. Typically these are found in a narrow lineages. In general, due to the low number of available sequences, this makes their call much less reliable.

In *Plasmodium* species, we detected a 5' extension of a cys-SPS which we believe to be very reliable, since we observe it conserved in all 7 investigated genomes in this lineage. This extension is about 500/550 amino acids long, and shows no homology with any known protein domain. *Plasmodiums* have a very lineage-specific selenoproteome (Lobanov2006), with at least 4 conserved selenoproteins with no homology to any other selenoprotein. The function of the extension remains totally unknown.

Figures in Supplementary Material S2:

Figure SM2.1: (next page)

Reconstructed protein phylogeny of all NADH dehydrogenase - SPS fused proteins identified in the tree of life.

(Figure SM2.1)



Supplementary Material S3:

Phylogeny of eukaryotic SPS proteins

We used selenoprofiles version 3 (Mariotti 2010, <http://big.crg.cat/services/selenoprofiles>) to scan our collection of eukaryotic and prokaryotic genomes. We used two profile alignments, *SPS* (containing sequences from all lineages) and *seld* (only bacterial SPS), to optimize sensitivity. Specificity of the profiles was ensured by manually calibrating filters until satisfactory results were obtained.

The final filter for both profiles requires the predictions to "fit" the profile in terms of average sequence identity with the profile sequences (awsi score -- see selenoprofiles manual). Also, every candidate is used as query with blastp against the nr database, and the titles of nr sequences matching a candidate are parsed; only if these titles match the expectations, defined by certain tags (such "selenophosphate synthetase" or "Selenide,water dikinase"), the candidate pass the filter (tag score -- see selenoprofiles manual).

Predictions by SPS and seld profiles were pulled together for all analyses.

The phylogenetic reconstruction procedure (see methods) was run on all the sets of all predictions in eukaryotes, all predictions in prokaryotes, predictions in the prokaryotic reference species, and prokaryotic reference+all eukaryotic predictions.

After manually inspecting results, we manually filtered out lots of eukaryotic predictions for a variety of reasons, producing a reference eukaryotic set. Some predictions were obvious bacterial contaminations, which we filtered out. Then, a number of duplicated predictions were removed, which are caused by the presence of the same stretch of DNA in two locations of the genome assembly (these are common just in certain species, presumably for a poor assembly strategy).

Lastly, pseudogenes were excluded; vertebrate genomes in particular were found very rich in SPS1 retrotransposed copies, recognizable for their lack of introns, and of active transcription.

Figure SM3.1 shows a tree of prokaryotic and eukaryotic SPS proteins pulled together. The predicted tree follows approximately the phylogenetic relationship of the lineages. Exceptions are found in some basal eukaryotic lineages, that possess a bacterial-like SPS when compared with metazoans. This includes all green algae (Chlorophyta), alveolates, but also some amoebozoa and heterolobosea. All fused SPS proteins found in eukaryotes are found here, clustering with bacterial sequences. This suggests again that they were horizontally transferred from bacteria to lower eukaryotes. Metazoans and their closest outgroups (choanoflagellida and other opisthokonts) instead show no sign of horizontal transfer, as they form an homogenous cluster. The unique possible exception is the NADH/SPS fusion found in *Hydra magnipapillata*, for which nonetheless we cannot be sure that the gene actually comes from a bacterial contamination (see also supplementary material S2).

SPS forms with no Sec nor Cys

Figure SM3.2 shows the predicted protein phylogeny of the eukaryotic reference set.

Mostly, non-metazoan eukaryotes possess a single SPS gene with either selenocysteine or cysteine, like Bacteria. Other forms, with none of these two amino acids, are found only in metazoans (see also Figure 2).

The great majority of vertebrates were predicted to possess two SPS genes, one with selenocysteine (SPS2) and one with threonine (SPS1-Thr). Among the exceptions, non-placental mammals (such as marsupials) possess two copies of SPS2, one of which is intronless. As we already described in (Mariotti 2012), at the root of placentals SPS2 was functionally replaced by one of its retrotranscribed copies. Non-placental mammals still retain both copies, although it is unclear whether they are both functional.

In the bird genome assemblies (genus *Melopsittacus*, *Taeniopygia*, *Gallus*, *Meleagris*), only SPS1 was found. Nonetheless, we could identify SPS2 in some EST sequences from *Gallus gallus*, which can not be mapped back to the genome. Thus we believe birds actually possess both SPS1 and SPS2 in their genome, but the latter is missing from the assemblies, presumably because of characteristics of their genomic location that make sequencing difficult.

Thus, with the only exception of non-placental mammals, we predict all other vertebrates to possess the two genes SPS1-Thr and SPS2, and we ascribe their absence in few species in our prediction set to the imperfect quality of genomes.

Non-vertebrate deuterostomes (such as *Strongylocentrotus* and *Branchiostoma*) possess a single gene with selenocysteine (SPS2). Along with the conservation of intron positions between the two genes (see figure SM3.2), and with the strong phylogenetic signal, this supports the fact that the vertebrate SPS1 gene was generated by duplication of SPS2 at the root of vertebrates, which is likely to have happened in the context of the notorious whole genome duplication.

In the tunicate *Ciona intestinalis*, we initially identified a single SPS gene with glycine aligned to Sec position. Later, we found that this gene produces two alternative forms, one with selenocysteine and one with glycine (the analysis on tunicates is expanded in Supplementary Material S4).

SPS1-Leu in Clitellata (Annelida)

In the genome of Annelida species *Helobdella robusta*, we identified two SPS genes, one with selenocysteine (SPS2) and another one carrying Leucine aligned to the Sec position (SPS1-Leu). The only other annelidan genome in our datasets (*Capitella teleta*) appears to possess a single SPS2 gene instead. Thus, we downloaded all EST available at NCBI from the lineage of Annelida, and we scanned them with Selenoprofiles to detect SPS genes. We found two distinct situations in the two main annelidan lineages, Polychaeta and Clitellata.

In the lineage Polychaeta, we have sublineages Sipuncula (represented by EST sequences of species *Sipunculus nudus*) and Scolecida (represented by the genome sequence of *Capitella teleta* and by EST sequences of *Capitella teleta*, *Malacoceros fuliginosus* and *Alvinella pompejana*). In all these cases, we found a single Sec-containing SPS gene (SPS2).

The lineage Clitellata is represented in our datasets by the genome sequence of *Helobdella robusta*, and by the EST sequences of *Helobdella robusta*, *Hirudo medicinalis* and *Tubifex tubifex*.

For all these species we found both SPS2 and SPS1-Leu. In the genome of *H. robusta*, we can see that these two genes possess a nearly identical intron structure. Both genes have EST support in *H. robusta* and *H. medicinalis*, with SPS1-Leu much more abundantly transcribed. In *T. tubifex* (for which we have relatively few EST, and no genome) we could

not observe SPS2, although we think that this is due only to low sequence coverage. In contrast, we observed two very similar SPS1-Leu proteins in the EST of this species.

In figure SM3.3, we compiled a collection of all SPS sequences found in Annelida EST and genome sequences. Altogether, data indicates that the ancestral annelidan SPS2 gene was duplicated in Clitellata, generating the SPS1-Leu gene, that conserved the intron structure of its parental gene. Then, this new gene was duplicated again in the lineage of *Tubifex tubifex*, after the split of Oligochaeta (containing *Tubifex*) with Hirudinida (containing *Helobdella* and *Hirudo*).

Testing duplication topologies for insect and vertebrate SPS1

The precise topology of SPS gene duplication and losses proved to be very hard to resolve in insects. This is due mostly to the high rate of sequence evolution of SPS2 in Dipteran insects. As you can see in figure SM3.2, the SPS2 protein of *Drosophila* or other Diptera is placed basal to both vertebrate SPS1 and SPS2. Interpreting literally this tree, this would imply that an ancestral duplication occurred, with a gene loss in vertebrates. We think this is just an effect of high sequence diversity in Diptera, a phylogenetic artifact known as long branch attraction. In fact, all other data point support instead two independent duplications in the separate lineages leading to vertebrate and insects: the species mapping at the root of vertebrates (non-vertebrate deuterostomes) and at the root of insects (non-insect protostomes) generally possess a single SPS gene (see Figure 2). The same thing is valid for non-bilaterian metazoans, which share the same last common ancestor with insects and vertebrates.

We built two “artificial” phylogenetic trees, representing the two possible duplication topologies for insects and vertebrates (see figure SM3.4). We then run the branch length optimization by phyml (Guindon 2003), and we computed the likelihood of the resulting trees. Applying a likelihood ratio test, we saw that the two values are not statistically different. The ancestral duplication topology has better score, and thus it is the one consistently reported by phyml. But the other topology (independent duplications) is much better supported by other observations. Thus, we can reliably say that SPS1 of insects and vertebrates were generated independently in the two lineages, after their split.

SPS phylogeny in arthropods and insects

We predicted the last common ancestor of all arthropoda to possess a single SPS2 gene (with Sec). To resolve the gene phylogeny within arthropoda, we created an alignment of all SPS genes found in arthropoda, and we run our phylogenetic reconstruction pipeline. The resulting tree can be inspected in figure SM3.5. Non-insect arthropods appear to possess only a single SPS2 gene. *Ixodes scapularis* only has a second copy of this gene, also with TGA. The protein tree indicates that this is a species-specific duplication. As the genome assembly available is quite fragmented, we cannot know whether the two genes possess a SECIS element, but we expect at least one gene to have one.

Among insects, different gene sets were identified in different lineages, including other SPS1 proteins.

In all Diptera, Lepidoptera and Coleoptera we identified an Arg-SPS gene (SPS1), and these genes clearly cluster together by sequence similarity. Then, in Diptera we also observe the Sec-containing SPS2 genes, which also cluster together, although with a higher degree of diversification. In *Drosophila willistoni* alone - the only known drosophila that lost selenoproteins - SPS2 was not found, consistent with its selenocysteine related function. Similarly, Lepidoptera and Coleoptera (that lost selenoproteins, Chapple 2008) lack a SPS2 gene.

Hymenoptera, as previously said, were found to possess a single TGA containing SPS gene, without identifiable SECIS in all genomes, which we believe to be readthrough. Additional gene fragments similar to SPS1 were found in some hymenopteran genomes, and also in the fly genomes of *D.persimilis* and *D.pseudoobscura* (see figure SM3.5). However none of those were supported by EST (in contrast with the readthrough gene, abundantly confirmed). Although we cannot rule out that these genes are true SPS family expansions in these lineages, we think that most likely they are just non-functional retrotransposed copies, and thus we excluded them of all subsequent analysis. Lastly, we found a very interesting situation in the basal insect group of Paraneoptera, with 3 genomes available: *Pediculus humanus* (Phthiraptera), *Rhodnius prolixus* (Hemiptera), *Acyrtosiphon pisum* (Hemiptera). In both *P.humanus* and *R.prolixus* (for which selenoproteins were identified), we found two TGA containing SPS genes. One had a clear SECIS, and clustered with Dipteran SPS2. The other had no SECIS, and clustered with known insect SPS1 proteins and hymenopteran SPS, and with arthropod SPS2 as outgroup (see figure SM3.5). These two genes in *R.prolixus* share a very similar intron structure, while the SPS2 gene in *P.humanus* has no introns at all. In contrast, *A.pisum* (that lost selenoproteins, Aphid Consortium 2010) contained a single SPS gene with arginine, also clustering with other SPS1 proteins.

All together, we think that data supports the following phylogenetic history (you may follow Figure 2 in the main paper). At the root of insects, the SPS2 gene was duplicated conserving its intron structure.

One copy retained the SECIS element, and presumably kept the SeP production function (SPS2). This gene started to evolve faster just after the duplication, as we see it accelerated in all insects.

The other copy did not retain the SECIS, and we believe that it exerts its function through a Sec-independent readthrough. This gene can be seen in this state in extant hymenopterans, as well as in paraneopteran *R.prolixus* and *P.humanus*. Then, both at the root of Diptera/Lepidoptera/Coleoptera and in the lineage of *A.pisum* (after the split with the other paraneoptera in our set) this gene switched the TGA codon to an arginine codon, becoming what we know as SPS1. Thus, we will refer to all the progeny of this SECIS-lacking, TGA-containing SPS as (insect) SPS1, using the suffix *rt* (readthrough) to denote the genes in which the TGA is still present, and readthrough (e.g. SPS1-rt Hymenoptera). As we discovered later (see Supplementary Material S6), this phylogenetic history is also well supported by analysis of the secondary structures and motifs found near the TGA site.

Figures in Supplementary Material S3:

Phylogeny of Selenophosphate synthetases (SPS)

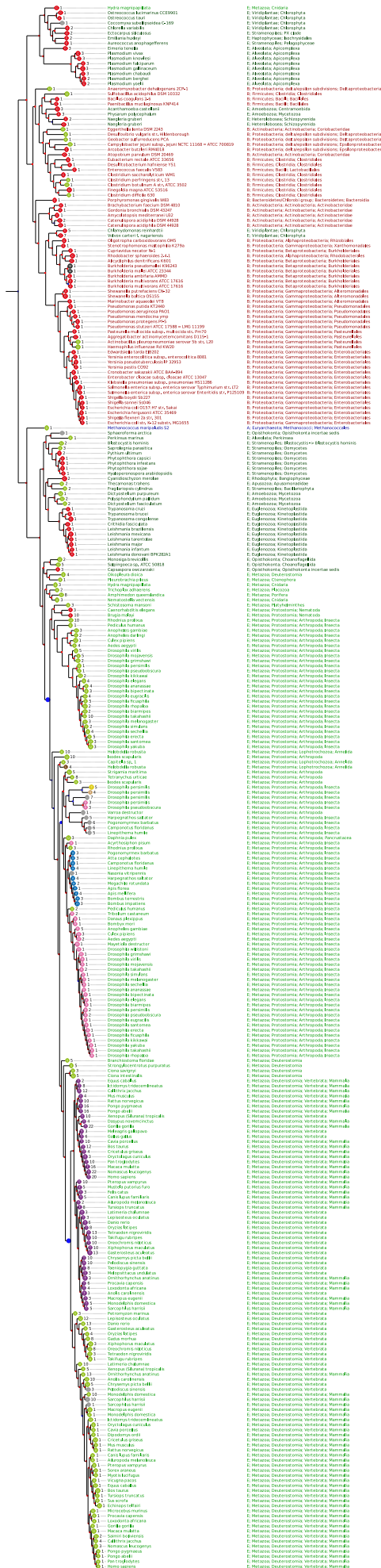


Figure SM3.1:

Reconstructed protein phylogeny of SelD/SPS proteins in the prokaryotic and eukaryotic reference set. On the left, the tree shows the predicted phylogenetic topology, with speciation and duplication events displayed as red and blue branch points, respectively. Colored balls are used to indicate the type of SPS (residue found at Sec position), as illustrated in Figure 4 of main paper. In addition to the protein types in Figure 4, here there are also some predictions in which the residue in Sec position is unknown, because not aligned; those are indicated as grey balls containing “-”. A few predictions contain pseudogene features (frameshifts or stop codons), and are indicated as dark grey balls containing “Ψ”. Next to each colored ball, the numeric id assigned by selenoprofiles is reported, allowing to identify uniquely this gene in the sequence set in Supplementary Material S7. Then, two columns report the species to which the gene belongs to, and a summary of their ncbi taxonomy. Both species and taxonomy are colored according to their kingdom: bacteria are in red, archaea are in blue, and eukaryotes in green (darker green for non-metazoan eukaryotes).

[illegible]

Figure SM3.2: Reconstructed protein phylogeny of the eukaryotic reference set of SPS proteins. See caption of figure SM3.1 for plot explanation. In respect to SM3.1, an additional column is present, displaying each gene as a colored rectangle. The width and position of the rectangle represents how the prediction spans the protein profile; black lines are used to indicate the intron positions, as projected in the protein alignment.

Phylogeny of Selenophosphate synthetases (SPS)

Figure SM3.3:

Alignment of SPS forms found in genomes and EST of species of Annelida. The Sec position is indicated in purple.

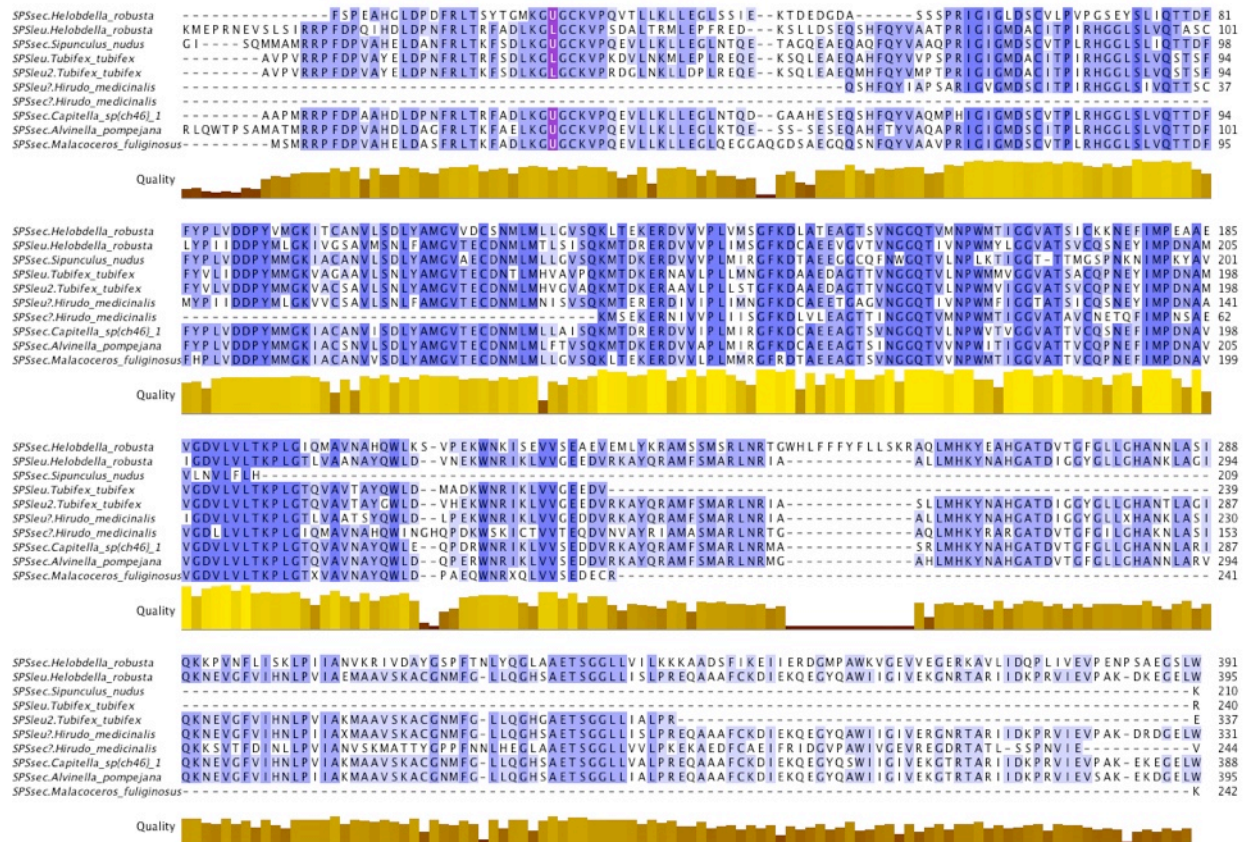
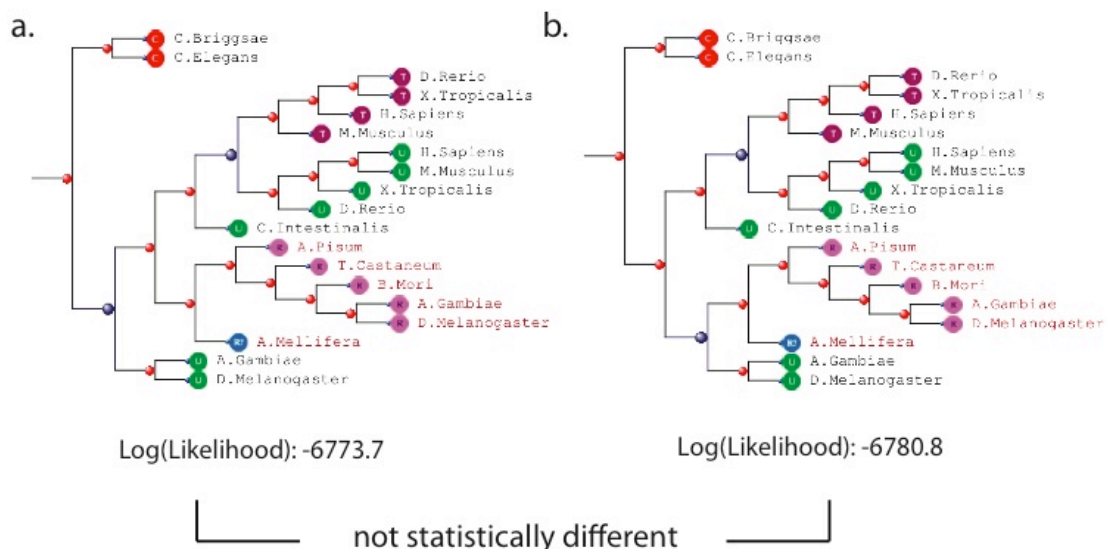


Figure SM3.4:

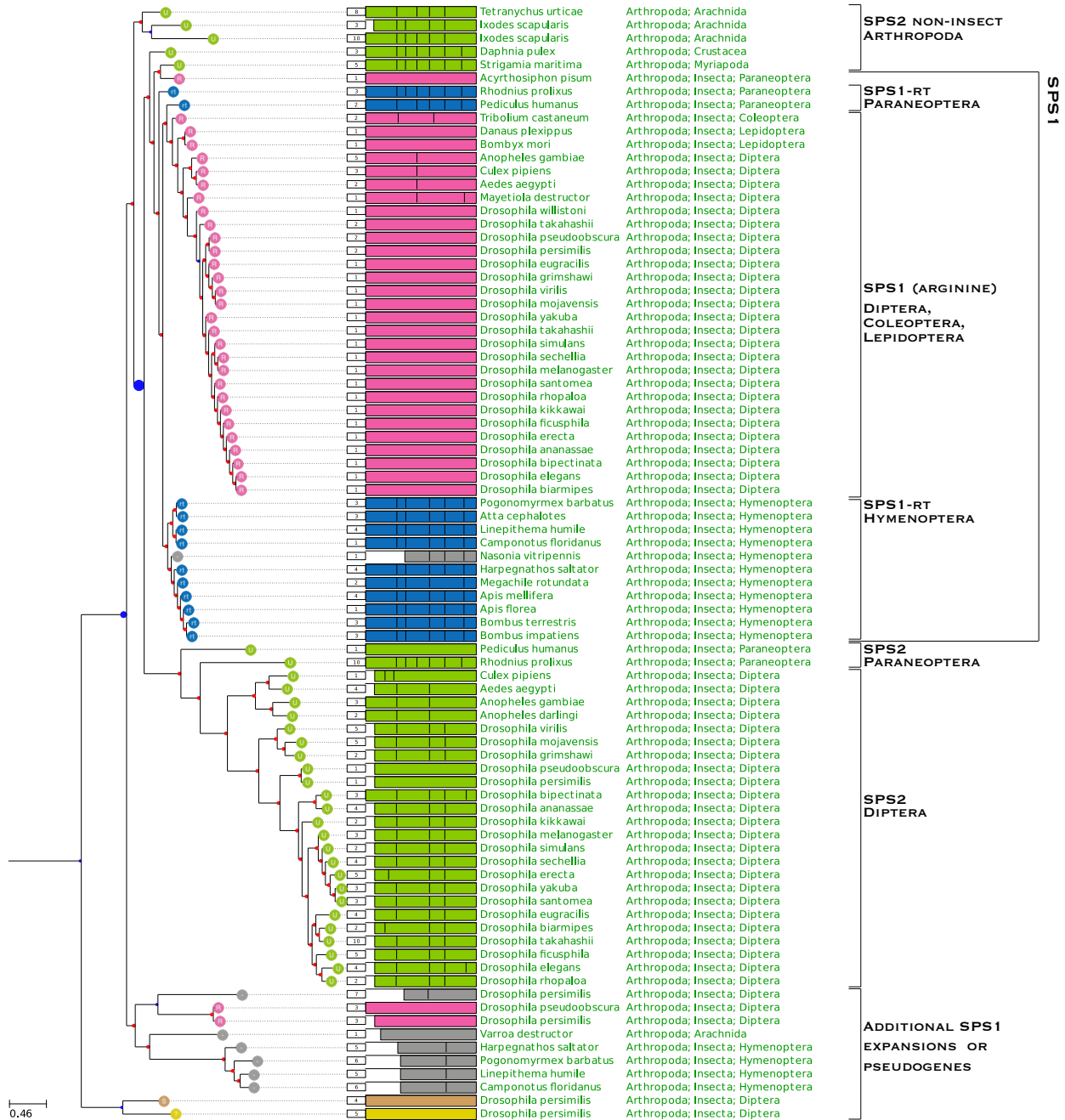
The two “backbone” phylogenetic trees representing the two possible topologies for vertebrate and insect SPS duplications: ancestral duplication (a) or lineage specific duplications (b). The first topology (the result of ML reconstructions) has better score. Nonetheless, its score is not statistically different than the one for the second topology, which is largely supported by other observations.



Phylogeny of Selenophosphate synthetases (SPS)

Figure SM3.5:

Reconstructed protein phylogeny of arthropoda SPS genes. See caption of figure SM3.1 and SM3.2 for plot explanation. UGA containing genes were classified as selenocysteine coding or readthrough based on phylogenetic clustering and presence of SECIS.



In our annotation of metazoan SPS genes, we ran into an interesting case in tunicates, that constitute the most closely related outgroup to vertebrates (Delsuc 2006). In the genome of *Ciona intestinalis*, we initially identified a single SPS gene, with glycine (Gly) aligned to Sec position. Later, we found that this gene actually produces two alternative forms, one with selenocysteine and one with glycine (see Figure SM4.1). The two forms differ only by the two first exons, and they are both supported by EST data. Both contain the same SECIS element downstream. Both alternative forms were found conserved in *Ciona savignyi*. In this genome assembly though, we see a single base insertion in the coding sequence, near the N-terminal, which would produce a frameshift, but we believe this to be artifactual, given the conservation in the rest of the region. Here we show the N-term portion of the alignment between SPSgly form of *C.intestinalis* (Query) and the corresponding genomic region in *C.savignyi* (Target):

Below, the N-term portion of the alignment between the other form (SPSsec) of *C.intestinalis* (Query) and the same genomic region in *C.savignyi* (Target) is shown. Notice that the two last exons shown (as well as the rest of the gene structure downstream, not shown here) are the same in the alignments above and below.

To gain some phylogenetic context, we searched SPS genes also in all other sequenced tunicates species. Figure SM4.2 shows an alignment of results found in all non-*Ciona*

Phylogeny of Selenophosphate synthetases (SPS)

tunicate ESTs. In *Oikopleura dioica*'s genome and ESTs, a single SPS gene was identified, with only a Sec form produced. In *Molgula tectiformis* EST data, the two forms with Sec and Gly were found, with a prevalence of Gly forms (in contrast to all other tunicates with 2 forms). The two forms differed only at the 5' end, as in *Ciona*. In *Halocynthia roretzi* and *Botryllus schlosseri* EST data, we found two distinct forms in each species. In these cases though, we see that the two forms differed in several positions, with divergence not limited to the N-terminal region. The differences spread across all protein length imply that two forms are produced by two distinct genes in these species. This was confirmed by the analysis of a preliminar genome assembly of *B.schlosseri*, not yet public (Ayelet Voskoboynik, personal communication). The two SPS genes as predicted from EST were both found in the genome assembly, in different scaffolds. Interestingly, while *B.schlosseri* SPSgly was found to possess approximately the same intron structure as the *Ciona* gene, *B.schlosseri* SPSsec gene has no introns. A SECIS element was found downstream of *Botryllus* SPSsec, but not downstream of SPSgly.

Showing here the alignment between the SPSgly isoform of *C.intestinalis* (Query) and *B.schlosseri* SPSgly gene (Target):

Query	DKKFRLTKYTGLHGGGCKVPNDVLVKLLQELGANPYHDEQYMGGMIMPRLG	<---Intron---	IGLDCCVIPLRFGGLSLLQTDDFFYPLIDDPYM
	/ / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / /	< 624bp	
Target	NNKSKSYTISGLHGGGCKVPREVLQKLLLEDGQSQYKEEHFMGGGIMPRLG		IGMDTCVILPLRFGGLSLLQTDDFFYPLVDDPYM
	aaatattaatgcccgggtagcaggccatcggtgcactaggctaggaacccg		agagatgactctggcctttcaagttctctgggcta
	caacacactctgtagggatcgatataattatgcagaaaaattggttctgt	gt ag	gtgtacgtttctgtggtcttaccattacattaacat
	ctatagtttattattccggcaacagaaggtctgcagcaaatcggcgcgata		ctcgcgcgaagaatctctgggggtttctcacctttg

Query	M	<---Intron---	GK ACANVLSDLYAIGVTECDNMLMLLVSSKFTEKERDVIPLMIHGFK	<---Intron---	DSAEAGTSGINGGQTVL
		<		>	
			54bp		143bp
Target	M		GK ACANVLSDMYAMGVIECDNMLMLLVSSKLETERDVVPLMIRGFK		DCADVAGTSITGGQTVL
	a		gaagtgagtagatgaggagtgaaacatcggatactgagcggggctaactga		gtgggggataaggcagc
	t	gt	gatcgcatgtgatactgttagaatttttgcatacacagattcttttgta	gt	agcagtcgctcgacct
	g	ag	aatccctggctgcgggaacgtcggggcttcgaataaagtatggccttcg	ag	ctctggtgacagagtgs

Query	NPWCLIGGVATTVCQQNEFIM	<---Intron---	PDNAVPGDVLVLTKPLGTQVACNSHQWLEQRNDKWNRIKLVVSEDEVEKAYHDAMFNMARLNR
		< 645bp >	
Target	NPWCLIGGVGTSVCQPNFIM		PDNAVPGDVLVLTKPLGTQVACVACWQLDQAADKWNRIKLVVSEDEVEKAYHDAMFMSRLNR
	actttaggggatgtccagtaa		cgaggcggggcgtaacccgacgttgccctcgcgggatacaacaagggtgattcgggataatacaa
	acggttggtgtcgtcacaatt	gt ag	caactctcagtttaccgtgatctcgaagtaacccaagattatttaaaagacaacaacttttcgttag
	tgggccttcgcggacgacacc		ggcctctaccctacgggttggaatctcgttagctgcgcacagttgcggccgaacaggttcctgcgcgaacc

[illegible][illegible]

Below, the alignment between SPSsec form of *C.intestinalis* (Query) and *B.schlosseri* SPSsec gene (Target):

Query	WDPVVHELSEEFRLTNFTGLKGUGCKVPQKVLKLLGLEALSNGFQN - GQLQPTPTVGIGLDCCVPLRFGGLSLLQTTDFFPYPLIDDPYMMGKIACAN
Target	WDPEEHGLEKFFRLTDYTGKKGUGCKVPQKVLKLLGLTNDGQPSQREGFPQTLPPTIGLDSCVIPLRFGGLSLLQTTDFFPYLVNDPYAMGKIACAN
	tgcggcgcaaatccagtagcagtgtagccagccaccggcaaggcctccgggtccaccaagagcggtgaccatggcttccaagtttccgagctgagaagtga gacaaaataaatgtcaacgtagggatcaatttattagtcaagaccagagtagctacctgtgactgtctgtgctttaccattacttaacagtactgcga gtcaacgaaggccgcgcacaaataactaactgattgtgtcccgactatcaccagacatcatataataaattgtttatatcttcatcccgctgtactcc

Phylogeny of Selenophosphate synthetases (SPS)

```

Query  VLSDLYAIGVTECDNMLMLLGVSSEKTEKERDVTIPLMIHGFKDSAEAGTSINGGQTVLNPWCLIGGVATTVCQNEFIMPDNAVPGDVLVLTPLGTQ
Target VLSDLYAMGVTECDNMLMLLGLSSKFTEERDVVPMIQQGFRDLAVEAGTNVTGGQTVINPWCLIGGVATSVCCQNEFIMPDQAVVGDVLVLTPLGTQ
gcagctgaggagtgaataccgtatatagggcggggcaaacgtagcgggggaaggaggcagaacttcaggggatgtccagtaacgcggggcgcaactgac
ttgatactgtcagaatttttggcatcaaagatttcttttagtgatctacgcacgcgacttacgggtgggtccctgaaaatttcaacttgattttcactgca
gtcttctgctgactcgggatagtagtgaaaccttagtggaacacctaatactgacgtacgttctgtgtatgggtataaacactgttaacctatggtataaga

Query  VACNSHQWLEQRNDKWNRIKLVSSEDEVEKAYHDAMFNMARLNRTAAQLMHTFNHSGATDVTGFGILGHAANLAKQQRSEVNFVIHNLPCIAKMAAIIKA
Target PAVNAFQWMNQKNQHNRIKHVISAEDTIKAYSDAILHMSRLNRHAARLMHVQAHAAATDVTGFGILGHAENLAKQQRNEVTFAIHNLPVISKMAAVSRA
cggagtctaacaaacctaaaacgatgggaaagtaggatcatacaccggacacgtcgcggaggagtgatgcggatgaccaggatgacatcgataagggacg
cctactagtaaaaaagagtaattccaactacagacttatcgtagaccgttattacacccatcgtgttgacaatcaaagaatctctaattcttcctcctggc
gtccgtaggtagtatgtatgcttggatacagccctcatggaactcggaggcataccgtctaataccagactgtagagatcagtcctctgggagaggcgcat

Query  CGNMFGLLQGTSAETSGLLICLPREQAQKFAEIKKVEGNQAWIIGIVEAGNRTARIIEKPRVIEV
Target SIVNFGLLKGTSAETSGLLIVLSREQATKYCQMIATEGHQAWIIGVVEKGDRSARIIGRPRIIEV
aagatgtcagatggatggccagctcggaattccaagaggccgttaagggaggatgcaagacaaagg
gttatgttagccaccgggttttctgaaccaagaattccagaacgttggttaagagccgttgcggttat
ccctcaggaatccgatctgacaccagatcgccgagcccaacgggtcgctaaacgtcatccggataac

```

In (Turon 2004), a phylogeny of Ascidians is reported. *Halocynthia roretzi* and *Botryllus schlosseri* are in two sister lineages (Styelidae and Pyuridae). *Molgula tectiformis* is not among the specimen analyzed in this paper. This species is classified under the Molgulidae lineage, which together with Styelidae and Pyuridae constitute the order of Stolidobranchiata (NCBI taxonomy). *Ciona* is basal to all others mentioned Ascidia, within the Enterogona order. Finally, *Oikopleura dioica* is a tunicate, but not ascidian, and thus constitutes our outgroup.

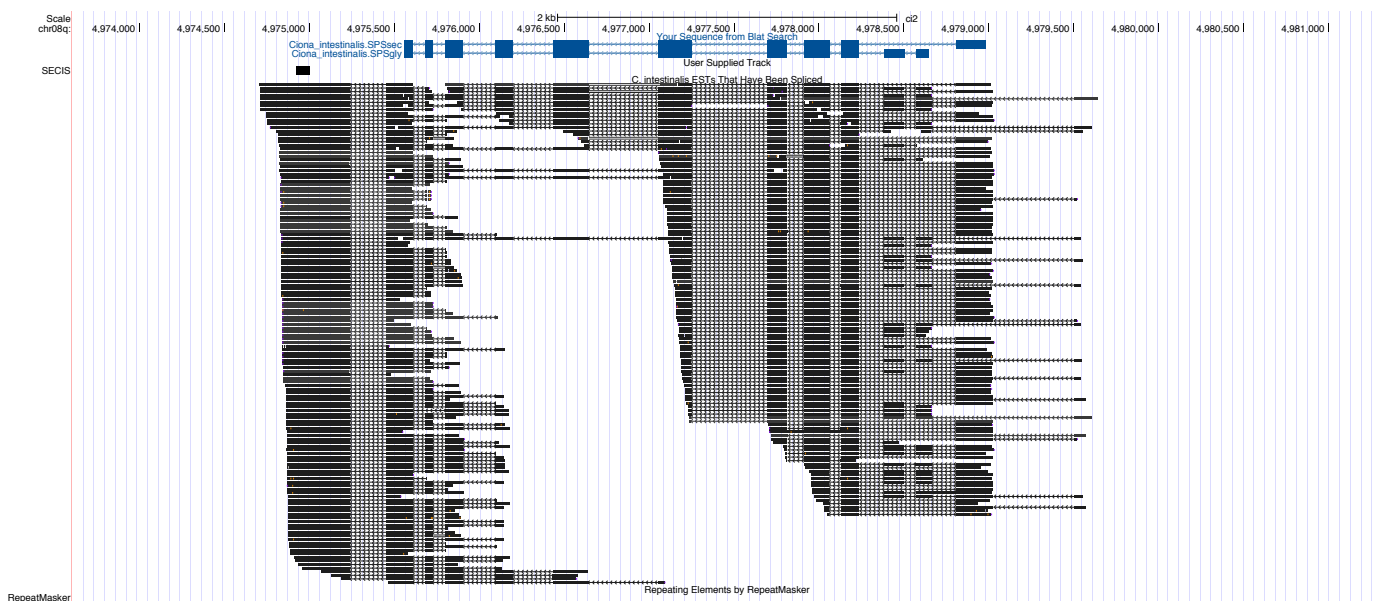
Altogether, we think the data clearly strongly support the following gene history. At the root of tunicates, a single SPS gene with selenocysteine was present (SPS2) -- as we found in *O.dioica*. Presumably at the root of Ascidians, the same gene originated a secondary isoform with glycine aligned to selenocysteine, as we found today in *Ciona* and in *M.tectiformis* (SPS-ae gene). At the root of Styelidae and Pyuridae, the selenocysteine isoform retrotransposed to the genome, generating a functional, intronless copy of SPSsec.

The parental gene then quickly lost its SPSsec isoform, thus specializing only in the SPSgly isoform. As result, the SECIS downstream of the parental gene (SPSgly) degenerated, while it was kept in the new SPSsec gene. This gene duplication is observed in species *H.roretzi* and *B.schlosseri*.

Figures in Supplementary Material S4:

Figure SM4.1:

Snapshot of the UCSC genome browser on the *Ciona intestinalis* genome at the SPS gene locus. The exonic structure of the coding sequence of the two isoforms is shown on top, in blue. The gene is on the negative strand, and the two forms differ only for the first exons (top right). Just below, the localization of the SECIS element is shown as a black rectangle. Below, the aligned EST sequences available at the genome browser are shown. ESTs support the two isoforms, and show that both share the same 3' UTR and thus the SECIS, although this is expected useless for the Gly form.



Phylogeny of Selenophosphate synthetases (SPS)

Figure SM4.2:

Alignment of SPS genes predicted with Selenoprofiles on tunicate ESTs downloaded from NCBI, excluding the *Ciona* genus. The column with selenocysteine is framed in red. On the left, the protein ids assigned by Selenoprofiles allow to identify the target species. The id also contains a label after the amino acid found at the Sec column. In some cases, the label is instead “pseudo”, when stop codons or frameshifts are predicted. Given the high level of gene conservation, those are probably caused just by low quality ESTs.

Ciona intestinalis SPgSgly 1 MALRPFKDPQSHNDRKFLRTKYLGHGGCKVQNDVLKLEGGAGA-NPTH-----DEQYMGGMIMPLRLGIGLSCVPLRFRGGLSLQTTDFFFPLVNDPYMMKG101
Ciona intestinalis SPgSrec 1 -----KWDVVHVEEELRFTLFTFGGCKVQNDVLKLEGGALSNGNQ-----QPTPTFEEGFMGMIMPLRLGIGLSCVPLRFRGGLSLQTTDFFFPLVNDPYMMKG102
Botryllus schlosseri SPs 6 unaligned 1 -----FMGMGMIMPLRLGIGLSCVPLRFRGGLSLQTTDFFFPLVNDPYMMKG103
Botryllus schlosseri SPs 83 glycine 1 MALRSKFDPEELQKSNRFLRTDYLGLKGCKVQKQVLLKLEGTN-SGQY-----EFGGGMIMPLRLGIGLSCVPLRFRGGLSLQTTDFFFPLVNDPYMMKG104
Botryllus schlosseri SPs 103 selenocysteine 1 MASEKLDWPEEHGLEKFRFLRTDYLGLKGCKVQKQVLLKLEGTN-DGQP-----SQRGEGFQTLPTTIGIGLSCVPLRFRGGLSLQTTDFFFPLVNDPYMMKG105
Botryllus schlosseri SPs 107 selenocysteine 1 MASEKLDWPEEHGLEKFRFLRTDYLGLKGCKVQKQVLLKLEGTN-DGQP-----SQRGEGFQTLPTTIGIGLSCVPLRFRGGLSLQTTDFFFPLVNDPYMMKG106
Botryllus schlosseri SPs 60 selenocysteine 1 MASEKLDWPEEHGLEKFRFLRTDYLGLKGCKVQKQVLLKLEGTN-DGQP-----SQRGEGFQTLPTTIGIGLSCVPLRFRGGLSLQTTDFFFPLVNDPYMMKG107
Botryllus schlosseri SPs 63 selenocysteine 1 MASEKLDWPEEHGLEKFRFLRTDYLGLKGCKVQKQVLLKLEGTN-DGQP-----SQRGEGFQTLPTTIGIGLSCVPLRFRGGLSLQTTDFFFPLVNDPYMMKG108
Botryllus schlosseri SPs 77 selenocysteine 1 MASEKLDWPEEHGLEKFRFLRTDYLGLKGCKVQKQVLLKLEGTN-DGQP-----SQRGEGFQTLPTTIGIGLSCVPLRFRGGLSLQTTDFFFPLVNDPYMMKG109
Botryllus schlosseri SPs 78 selenocysteine 1 MASEKLDWPEEHGLEKFRFLRTDYLGLKGCKVQKQVLLKLEGTN-DGQP-----SQRGEGFQTLPTTIGIGLSCVPLRFRGGLSLQTTDFFFPLVNDPYMMKG110
Botryllus schlosseri SPs 79 selenocysteine 1 MASEKLDWPEEHGLEKFRFLRTDYLGLKGCKVQKQVLLKLEGTN-DGQP-----SQRGEGFQTLPTTIGIGLSCVPLRFRGGLSLQTTDFFFPLVNDPYMMKG111
Botryllus schlosseri SPs 80 selenocysteine 1 MASEKLDWPEEHGLEKFRFLRTDYLGLKGCKVQKQVLLKLEGTN-DGQP-----SQRGEGFQTLPTTIGIGLSCVPLRFRGGLSLQTTDFFFPLVNDPYMMKG112
Botryllus schlosseri SPs 81 selenocysteine 1 MASEKLDWPEEHGLEKFRFLRTDYLGLKGCKVQKQVLLKLEGTN-DGQP-----SQRGEGFQTLPTTIGIGLSCVPLRFRGGLSLQTTDFFFPLVNDPYMMKG113
Botryllus schlosseri SPs 94 pseudo 1 MASEKLDWPEEHGLEKFRFLRTDYLGLKGCKVQKQVLLKLEGTN-DGQP-----SQRGEGFQTLPTTIGIGLSCVPLRFRGGLSLQTTDFFFPLVNDPYMMKG114
Botryllus schlosseri SPs 132 pseudo 1 MASEKLDWPEEHGLEKFRFLRTDYLGLKGCKVQKQVLLKLEGTN-DGQP-----SQRGEGFQTLPTTIGIGLSCVPLRFRGGLSLQTTDFFFPLVNDPYMMKG115
Botryllus schlosseri SPs 147 pseudo 1 MASEKLDWPEEHGLEKFRFLRTDYLGLKGCKVQKQVLLKLEGTN-DGQP-----SQRGEGFQTLPTTIGIGLSCVPLRFRGGLSLQTTDFFFPLVNDPYMMKG116
Botryllus schlosseri SPs 150 pseudo 1 MASEKLDWPEEHGLEKFRFLRTDYLGLKGCKVQKQVLLKLEGTN-DGQP-----SQRGEGFQTLPTTIGIGLSCVPLRFRGGLSLQTTDFFFPLVNDPYMMKG117
Botryllus schlosseri SPs 166 pseudo 1 MASEKLDWPEEHGLEKFRFLRTDYLGLKGCKVQKQVLLKLEGTN-DGQP-----SQRGEGFQTLPTTIGIGLSCVPLRFRGGLSLQTTDFFFPLVNDPYMMKG118
Botryllus schlosseri SPs 169 glycine 1 MASEKLDWPEEHGLEKFRFLRTDYLGLKGCKVQKQVLLKLEGTN-DGQP-----SQRGEGFQTLPTTIGIGLSCVPLRFRGGLSLQTTDFFFPLVNDPYMMKG119
Botryllus schlosseri SPs 171 pseudo 1 MASEKLDWPEEHGLEKFRFLRTDYLGLKGCKVQKQVLLKLEGTN-DGQP-----SQRGEGFQTLPTTIGIGLSCVPLRFRGGLSLQTTDFFFPLVNDPYMMKG120
Botryllus schlosseri SPs 391 pseudo 1 MASEKLDWPEEHGLEKFRFLRTDYLGLKGCKVQKQVLLKLEGTN-DGQP-----SQRGEGFQTLPTTIGIGLSCVPLRFRGGLSLQTTDFFFPLVNDPYMMKG121
Okipeleura dioica SPs 1 unaligned 1 -----PEEHGLEKFRFLNFTGLKGCKVQKQVLLKLEGTN-NSA-PAHEVPKSNQNCNPKHNAIKSDEIGLGCQVPIPHSDFIQTTDFFFPLVNDPYMMKG122
Okipeleura dioica SPs 9 selenocysteine 1 -----SVRDWLWPEEHGLEKFRFLNFTGLKGCKVQKQVLLKLEGTN-NS-PAHEVPKSNQNCNPKHNAIKSDEIGLGCQVPIPHSDFIQTTDFFFPLVNDPYMMKG123
Okipeleura dioica SPs 14 selenocysteine 1 -----SVKDVWVPEEHGLEKFRFLNFTGLKGCKVQKQVLLKLEGTN-NSA-PAHEVSKSNQNCNPKHNAIKSDEIGLGCQVPIPHSDFIQTTDFFFPLVNDPYMMKG124
Okipeleura dioica SPs 15 unaligned 1 -----EIGLGCQVPIPHSDFIQTTDFFFPLVNDPYMMKG125
Okipeleura dioica SPs 19 unaligned 1 -----KEFLNFTGLKGCKVQKQVLLKLEGTN-NSA-PAHEVPKSNQNCNPKHNAIKSDEIGLGCQVPIPHSDFIQTTDFFFPLVNDPYMMKG126
Okipeleura dioica SPs 20 selenocysteine 1 -----SVKDVWVPEEHGLEKFRFLNFTGLKGCKVQKQVLLKLEGTN-NSA-PAHEVPKSNQNCNPKHNAIKSDEIGLGCQVPIPHSDFIQTTDFFFPLVNDPYMMKG127
Okipeleura dioica SPs 44 selenocysteine 1 -----SVKDVWVPEEHGLEKFRFLNFTGLKGCKVQKQVLLKLEGTN-NSA-PAHEVPKSNQNCNPKHNAIKSDEIGLGCQVPIPHSDFIQTTDFFFPLVNDPYMMKG128
Okipeleura dioica SPs 56 pseudo 1 -----SVKDVWVPEEHGLEKFRFLNFTGLKGCKVQKQVLLKLEGTN-NSA-PAHEVPKSNQNCNPKHNAIKSDEIGLGCQVPIPHSDFIQTTDFFFPLVNDPYMMKG129
Okipeleura dioica SPs 108 selenocysteine 1 -----SVKDVWVPEEHGLEKFRFLNFTGLKGCKVQKQVLLKLEGTN-NSA-PAHEVPKSNQNCNPKHNAIKSDEIGLGCQVPIPHSDFIQTTDFFFPLVNDPYMMKG130
Okipeleura dioica SPs 110 selenocysteine 1 -----SVKDVWVPEEHGLEKFRFLNFTGLKGCKVQKQVLLKLEGTN-NSA-PAHEVPKSNQNCNPKHNAIKSDEIGLGCQVPIPHSDFIQTTDFFFPLVNDPYMMKG131
Okipeleura dioica SPs 117 pseudo 1 -----PEEHGLEKFRFLNFTGLKGCKVQKQVLLKLEGTN-NSA-PAHEVPKSNQNCNPKHNAIKSDEIGLGCQVPIPHSDFIQTTDFFFPLVNDPYMMKG132
Okipeleura dioica SPs 123 selenocysteine 1 -----SVKDVWVPEEHGLEKFRFLNFTGLKGCKVQKQVLLKLEGTN-NSA-PAHEVPKSNQNCNPKHNAIKSDEIGLGCQVPIPHSDFIQTTDFFFPLVNDPYMMKG133
Okipeleura dioica SPs 129 selenocysteine 1 -----SVKDVWVPEEHGLEKFRFLNFTGLKGCKVQKQVLLKLEGTN-NSA-PAHEVPKSNQNCNPKHNAIKSDEIGLGCQVPIPHSDFIQTTDFFFPLVNDPYMMKG134
Okipeleura dioica SPs 130 selenocysteine 1 -----SVKDVWVPEEHGLEKFRFLNFTGLKGCKVQKQVLLKLEGTN-NSA-PAHEVPKSNQNCNPKHNAIKSDEIGLGCQVPIPHSDFIQTTDFFFPLVNDPYMMKG135
Okipeleura dioica SPs 485 selenocysteine 1 -----PEEHGLEKFRFLNFTGLKGCKVQKQVLLKLEGTN-NSA-PAHEVSKSNQNCNPKHNAIKSDEIGLGCQVPIPHSDFIQTTDFFFPLVNDPYMMKG136
Okipeleura dioica SPs 491 selenocysteine 1 -----PEEHGLEKFRFLNFTGLKGCKVQKQVLLKLEGTN-NSA-PAHEVPKSNQNCNPKHNAIKSDEIGLGCQVPIPHSDFIQTTDFFFPLVNDPYMMKG137
Okipeleura dioica SPs 498 selenocysteine 1 -----SVRDWLWPEEHGLEKFRFLNFTGLKGCKVQKQVLLKLEGTN-NS-PAHEVPKSNQNCNPKHNAIKSDEIGLGCQVPIPHSDFIQTTDFFFPLVNDPYMMKG138
Okipeleura dioica SPs 499 selenocysteine 1 -----PEEHGLEKFRFLNFTGLKGCKVQKQVLLKLEGTN-NSA-PAHEVPKSNQNCNPKHNAIKSDEIGLGCQVPIPHSDFIQTTDFFFPLVNDPYMMKG139
Halocynthia roretzi SPs 169 glycine 1 MALRPFKDPQSHNDRKFLRTKYLGHGGCKVQNDVLKLEGGAGA-TQYQ-----DEQYLGGMIMPLRLGIGLSCVPLRFRGGLSLQTTDFFFPLVNDPYMMKG140
Halocynthia roretzi SPs 124 glycine 1 MALRPFKDPQSHNDRKFLRTKYLGHGGCKVQNDVLKLEGGAGA-TQYQ-----DEQYLGGMIMPLRLGIGLSCVPLRFRGGLSLQTTDFFFPLVNDPYMMKG141
Halocynthia roretzi SPs 8 unaligned 1 -----GNXTDKRFLRTDYLGHGGCKVQNDVLKLEGTN-TPNP-----TDNYQTNVLLSVLTGMLRFRGGLSLQTTDFFFPLVNDPYMMKG142
Halocynthia roretzi SPs 70 selenocysteine 1 -----MAMLLKWDPEEHGCKFRFLRTDYLGHGGCKVQNDVLKLEGTN-TPNP-----TDNYQTNVLLSVLTGMLRFRGGLSLQTTDFFFPLVNDPYMMKG143
Halocynthia roretzi SPs 125 unaligned 1 -----KFSPEALGDKNFRFLRTDYLGHGGCKVQNDVLKLEGGSD-D-----EDSKDNKLGPIVIGLSCVPLRFRGGLSLQTTDFFFPLVNDPYMMKG144
Halocynthia roretzi SPs 169 glycine 1 -----PEELKPNRFLRTKFSGLHGGCKVPIDNLKLEGG-SNYH-----EQYIGGMAPRLGIGLSCVPLRFRGGLSLQTTDFFFPLVNDPYMMKG145
Halocynthia roretzi SPs 13 unaligned 1 -----MAYRPFKDPPELCKPNRFLRTKFSGLHGGCKVPIDNLKLEGG-SNYH-----EQYIGGMAPRLGIGLSCVPLRFRGGLSLQTTDFFFPLVNDPYMMKG146
Halocynthia roretzi SPs 16 unaligned 1 -----MAYRPFKDPPELCKPNRFLRTKFSGLHGGCKVPIDNLKLEGG-SNYH-----EQYIGGMAPRLGIGLSCVPLRFRGGLSLQTTDFFFPL

Supplementary Material S5:

Secondary structures within coding sequences of SPS genes

The program RNAz (Gruber 2010) predicts structured RNA motives in nucleotide alignments of homologous regions in different species. The conservation of base pairings (with particular attention to compensatory mutations), is used to infer the presence of a functional secondary structure in a set of species.

We have run RNAz 2.1 on SPS genes coding sequences, to characterize the known secondary structures overlapping or just subsequent to the TGA.

We built various subset alignments of SPS sequences along the tree of life, based on the residue found at Sec position and on the species phylogenetic branching. Different resolutions (i.e. lineage depths) were tested, thus certain subsets are contained in other more general subsets. The final list of lineages for which subsets were created is the following:

- prokaryotes, archaea, bacteria, clostridiales, proteobacteria, campylobacter (epsilonproteobacteria), deltaproteobacteria, pasteurilla;
- non-metazoan eukaryotes, metazoa, non-bilaterian metazoa, basal bilateria (non-insecta, non-chordata), vertebrata, insecta, diptera, drosophila, hymenoptera, non-hymenopteran insecta

For each lineage, up to 4 subsets were created, depending on the type of SPS found: selenocysteine, cysteine, other residue, all together. The coding sequences of all genes in the filtered set of eukaryotic and prokaryotic SelD/SPS predictions were considered. Sequences were aligned using their peptide translation.

Then, for each lineage subset, we have trimmed off the alignment columns with more than 70% gaps. Additionally, some sequences were removed from subsets after manual inspection, for carrying large gapped regions.

Full length coding sequences alignments were input to the RNAz utility *rnazWindow.pl*, that partitioned each alignment with a sliding window, 80 or 120 bp wide, with a step size of 20. This program also reduces the number of sequences to six, selecting representatives for each window. For some large sequence subsets, we decided to try also another method to select representatives: we ran trimal (Capella-Gutiérrez 2009) on our full length alignments to select 10, 14, 18, 22, 26 or 30 representatives, and then we fed the resulting alignments to *rnazWindow.pl*. In some cases, this improved the predicted stability and probability call of RNAz hits.

The RNAz outputs on all combinations of lineage, SPS type, and trimming procedure were parsed and inspected, to produce a reliable set of secondary structures. A few secondary structures were predicted far from the Sec TGA but still within the coding sequence, but only in certain lineages and with narrow combination of parameters. We considered those to be false positives, justified by the huge number of alignments tested. The only region that was consistently predicted to contain secondary structures was where the TGA resides, in selenocysteine containing genes and in hymenopteran and paraneopteran SPS1. For each candidate region, we carefully inspected all relevant RNAz outputs to choose the most likely structure, trying to minimize the computed fold energy. Finally,

images were produced as indicated in the RNAz manual, that is to say, using tools from the Vienna RNA package (Lorenz 2011).

Prokaryotes: bSECIS elements

A general structure from the set of all selenocysteine containing SelD proteins in prokaryotes, bacteria or archaea could not be obtained, presumably for that exceeds in diversity the detection power of RNAz. Nonetheless, we could get models for several bacterial sublineages, shown in Figure SM5.1.

In proteobacteria, all three structures (B, C, D) feature two stem-loops (stem1, stem2) downstream of the Sec TGA, separated just by a small, variable bulge. The apex of the second stem is minimal: only 3 or 4 bases are predicted to form this unpaired loop.

An additional stem, which just precedes or includes the Sec TGA, is often predicted (A, C, D). The bSECIS predicted for Clostridiales is somehow different, in that stem2 appears to be located downstream of stem1, in contrast to the rest of predictions in which stem2 falls within the two arms of stem1. No structure was predicted in archaeal Sec-SPS coding sequences.

Eukaryotes: SRE and HRE, and the readthrough enhancing hexanucleotide

Although rather different in sequence, the eukaryotic consensus structures are similar to the bacterial counterpart, in that they all contain stable stems starting about 2-10 bp downstream of the TGA (see Figure SM5.2). The most stable and large stems were predicted in hymenopteran sequences. As said, hymenoptera lost the ability to produce selenocysteine, and no SECIS is found downstream. We believe this gene to be readthrough in a Sec independent mechanism, supported by its conservation in all hymenoptera genomes. In respect to a bacterial SECIS, this hymenopteran readthrough element (HRE) contains an additional large upstream stem, forming a 3 stems clover structure with the TGA on the apex of the middle stem. A similar structure is predicted in basal metazoans, although stem lengths are quite different.

In all hymenoptera, we noticed a peculiar readthrough enhancing hexanucleotide (Harrell 2002) extremely conserved, right next to TGA: GGGTG[T/C].

This hexanucleotide can be seen also in the consensus structure for basal metazoa and basal bilateria (figure SM5.2). We thus searched it in all our aligned sequence set.

Besides hymenoptera and paraneopteran SPS1-rt, the hexanucleotide was found in a number of some metazoan SPS2, phylogenetically located basal to insects, vertebrates or to all bilateria. We noticed an inverse correlation between the presence of the hexanucleotide in a TGA containing SPS gene and the presence of a SPS1-like paralogue in the same species (see Figure 5 in the main paper).

Examining results in view of our functional hypotheses

If we inspect this data in the view of our subfunctionalization hypotheses (see paper), we see that it gives it support. In fact, we predict that a function duplication occurred in the ancestral SPS2, before the split of bilateria, and we think that the secondary function was carried out by a non-Sec readthrough isoform.

Thus, for those species that possess uniquely the descendant of that gene (no gene duplications, losses, or conversions to Cys), we expect the production of a non-SECIS, non-Sec dependent readthrough isoform to be important.

Excluding a few vertebrates and *Ciona*, we observe the presence of the hexanucleotide precisely in these species. Among vertebrates, we see this only in a few, basal species. *Ciona* themselves are basal to vertebrates. Thus, the presence of the hexanucleotide in these species suggest that it was present in their last common ancestor, and it was then lost in most vertebrates, presumably as consequence of the appearance of SPS1-Thr: as the secondary function was transferred to another gene, the function of SRE went back to

be only a support for Sec insertion, while before it had to maintain also an acceptable level of non-Sec readthrough.

Among ascidians, the hexanucleotide is found only in the most basal *Ciona* lineage, while it is mutated in the rest of species, including *Botryllus* and *Halocynthia*, that possess a retrotransposed copy of the Gly-SPS isoform (see Supplementary Material S4).

Interestingly, in Annelida we see the hexanucleotide in *Capitella sp. 1* but not in *Helobdella robusta*, as in the latter (but not in the former) a duplication presumably transferred the second function to a Leu homologue.

Concluding, we think that the hexanucleotide can be seen as an approximate marker for a conserved non-Sec readthrough, that together with a SECIS element, is an indicator of a double function. Nonetheless, note that for we expect the readthrough to be happening and important for a few species without the hexanucleotide (e.g. *Schistosoma*, *Oikopleura*), and viceversa in some species we think it is just a relic and it will degenerate in time (e.g. *Ciona*, *Danio*).

Figures in Supplementary Material S5:

Figure SM5.1:

bSECIS elements in prokaryotic SPS genes. The structures obtained with sequences of Clostridiales (A), Campylobacter (B), Deltaproteobacteria (C) and Pasteurellales (D) are shown. Red base pairs are conserved in all representatives sequences. Yellow and green base pairs are supported by 2 or 3 different pairs (compensatory mutations). Pale colors indicate only partial sequence support. The Sec TGA is circled in purple.

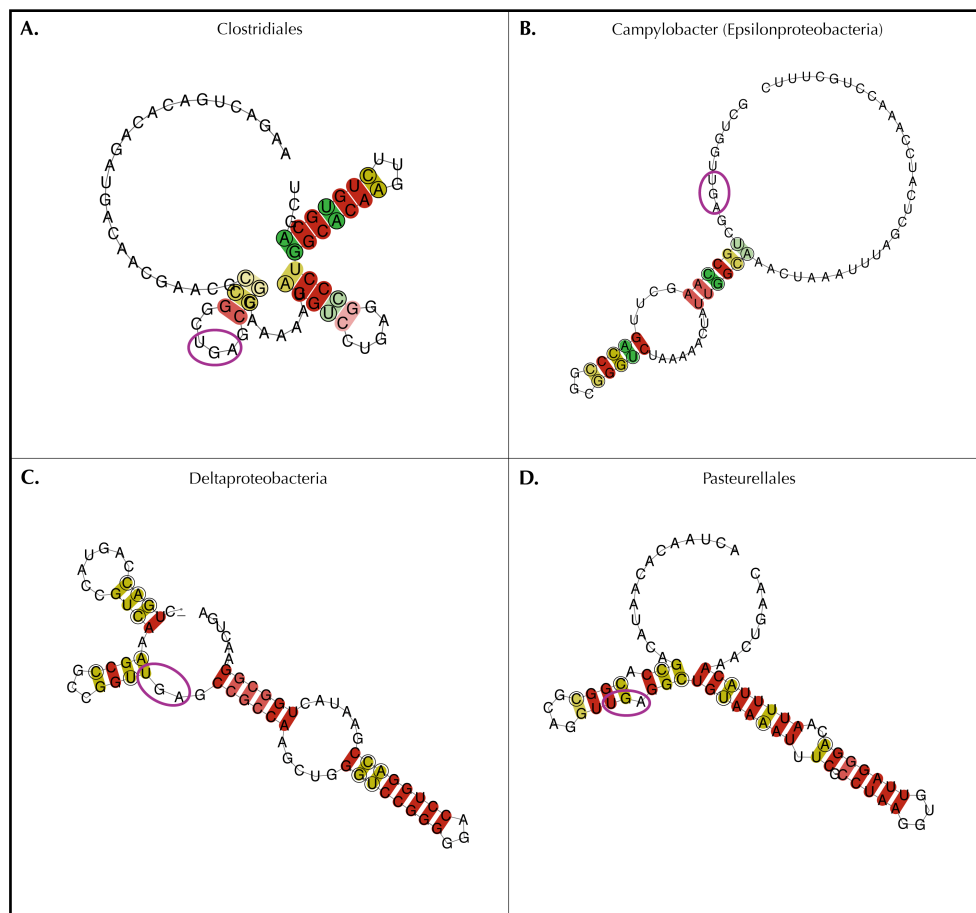
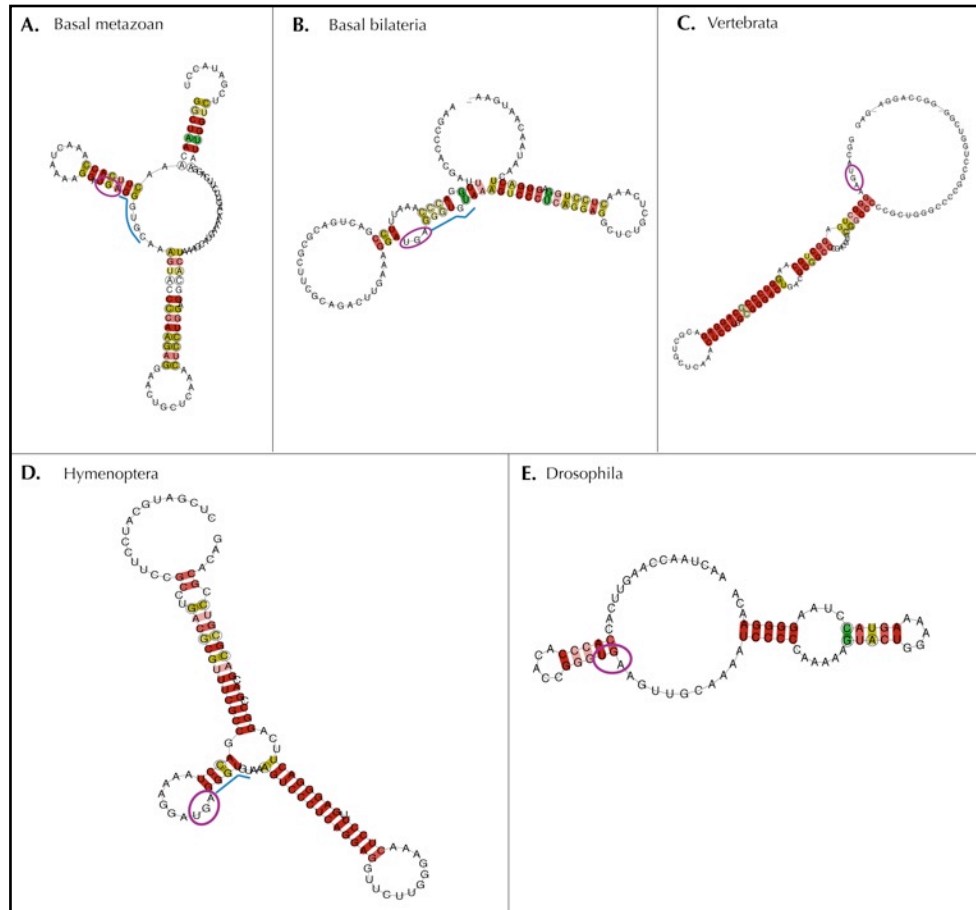


Figure SM5.2:

Recoding elements in eukaryotic SPS genes. The structures obtained with sequences of non-bilaterian metazoa (A), non-vertebrate, non-insect bilateria (B), Vertebrata (C), Hymenoptera (D) and *Drosophila* (E) are shown. See caption of SM5.1 for explanation of base coloring. The Sec TGA is circled in purple. The presence of readthrough enhancing hexanucleotides GGGTG[C/T] is indicated with a blue line.



Supplementary Material S6:

Rescue experiments in *Drosophila*

We obtained cDNA for human SPS1 from the Harvard resource core (<http://plasmid.med.harvard.edu/PLASMID/>). For *Ciona intestinalis* (ascidian), we obtained the cDNA corresponding to the Gly isoform of SPS-ae by performing targeted PCR on extract provided by Salvatore D'Aniello, currently at Stazione Zoologica Anton Dohrn (Napoli). We obtained cDNA for SPS1-rt from *Atta cephalotes* (leafcutter ant) by performing targeted PCR on extracts provided by James F.A. Traniello at Boston University.

We transformed DH5 α cells and performed midi preps of the three DNA samples to get enough amount of DNA to inject embryos (at least 10 ml with the minimal concentration of 300ng/ml).

sample	ID	Date	ng/ul
269	human Sps1	10/04/2012	850.96
270	ciona Sps1	10/04/2012	775.72
265	atta Sps1	10/04/2012	683.71

To obtain the transgenic flies, we used the method described by Bischof et al. (PNAS February 27, 2007; 3312-3317)

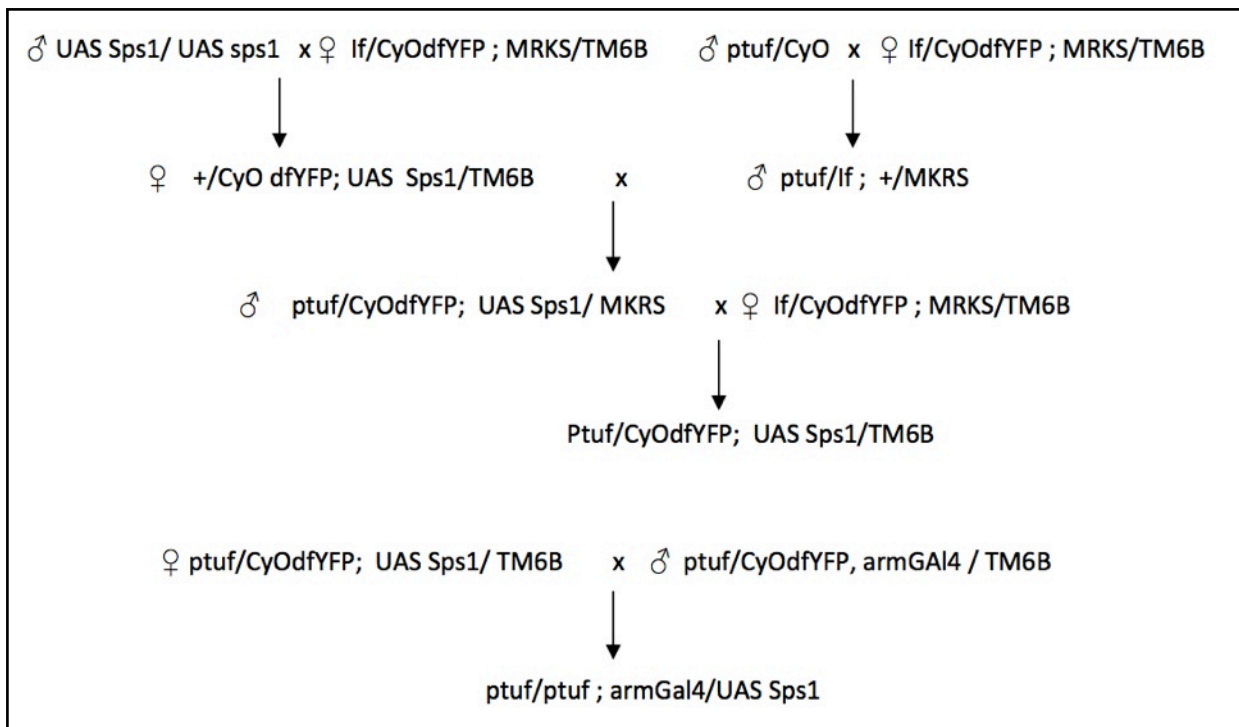
We used the line ywM{eGFP.vas-int.DM}ZH-2A; + ; 3: M{RFP.attP}ZH-86Fb to direct the insertion to the 3R chromosome (86F).

We designed the crosses illustrated in figure SM6.1 to get the expression of each transgenic SPS1 in a ptuf (*drosophila* SPS1) mutant background. We then examined the imaginal disc in these organisms (see figure 6 in main paper).

Figures in Supplementary Material S6:

Figure SM6.1:

Schema of crosses to obtain a transgenic fly expressing an heterologous SPS1 in a ptuf mutant background (ptuf = drosophila SPS1).



Chapter 4

DISCUSSION

Since the majority of this thesis is in form of scientific papers, most of the discussion has been already presented in the relevant sections. Here, I include thoughts arising from a general view of our work and experience with selenoproteins.

4.1 Are selenoproteins essential?

This question is often asked in the field of selenoproteins, in one form or another. The real question is probably whether selenocysteine is essential. From the selenoproteinless organisms we can say that it is definitely not essential for life, meaning not for all living organisms. But for some of them, it is: vertebrates possess a very rich selenoproteome, with indispensable selenoenzymes such as GPx and TR. The selection acting on these important genes propagates to the selenocysteine trait, translating in tight conservation of Sec machinery. Insects, on the other side, have gradually reduced the selenoproteome size in various steps from the split with other arthropods. The most prominent selenoprotein families have been lost or converted sequence and function. In *D.melanogaster*, the only two selenoproteins left (not counting SPS2) are still required for growth, as their knock-down phenotype suggests. But this species is not so far away from selenocysteine dispensability. A few changes in the anti-redox or some other system, and these two genes are allowed to be converted or lost. When this happens, the genes forming the Sec trait start to drift, like in extant *D.neocordata*. Given enough time, the only memory of Sec in this genome will be its ex-selenoproteins, now cysteine homologues, and the Sec machinery genes that were conserved for non-Sec related functions, as we see in extant *D.willistoni* and in lots of other selenoproteinless insects.

Thus, questions about the fitness of selenocysteine must be addressed separately for different lineages: selenocysteine is just as necessary as the selenoprotein genes that requires its incorporation. In other words, the importance of selenocysteine for an organism depends critically on the phylogenetic and functional history of its selenoproteins. And the amount and nature of the selenoproteins in any species changed drastically during the history of life, shaped by natural selection,

and also certainly by pure chance. During my PhD, I developed computational tools to predict and trace the history of selenoproteins across genomes, delineating a phylogenetic context that hopefully allows a better understanding of the selective forces acting on selenocysteine.

4.2 Selenoproteins as test case

Selenoproteins are just a tiny fraction of the total proteome known in living organisms. Despite their peculiarities, we expect that most aspects regarding their evolution are analogous for most, if not all proteins. The key events shaping the evolution of protein-coding genes are universal: gene duplication, gene loss, amino acid replacements, alternative isoforms, regulation control. We think that selenoproteins provide advantages for functional genomics studies. The presence of selenocysteine alone indicates a potential role in redox reactions. The observed replacement of selenocysteine with cysteine in homologous genes supports that, and analogously the replacement with something else is often an argument for a change of molecular function in the homologue. Our understanding of selenocysteine biology can be exploited to gain insights on the evolution and function of the rest of proteins. For example, in [Fomenko and Gladyshev, 2012], authors characterized thiol-oxidoreductases on a genome-wide scale, taking advantage of Sec homologues of candidate proteins. In a way, this approach resembles the strategy for selenoprotein identification that searches for Cys homologues, but reversed. Similarly, we believe that the contribution of our studies will not be limited to the selenoprotein field. One clear example is the program selenoprofiles, initially developed for selenoproteins, but now useful for general annotation purposes. Then, we believe that our discoveries in *D.neocordata* could be useful to the field of evolutionary genomics. This branch of biology tries to understand how allele variants are generated, how their frequency changes over time in populations, and how speciation events come in the picture. Measures of spontaneous mutation and neutral evolution rate are essential parameters in any evolutionary model, and yet many problems arise from currently used models. The genome in the cell is a complex macromolecule with dynamically evolving compaction states, reflecting the regulation of processes such as transcription or duplication. Mutations do not occur with equal probability on a naked or condensed genome. In *D.neocordata*, the very recent Sec machinery pseudogenes offer the opportunity to observe how mutations accumulate in real genes just after they lose their function, allowing the development of better models for neutral evolution and pseudogenization.

In a completely different perspective, selenocysteine could be used as a generic prototype of function used by living organisms. As depicted in our model for drosophila, Sec can be seen in a functional network located upstream of selenoproteins, with Sec machinery upstream of Sec itself. In our studies, we observed how Sec machinery degenerates when selenoproteins are lost from a genome. Not all genes are lost though: some are conserved, presumably because they had acquired

additional functions in certain lineages. I believe that the process that we described may exemplify well the evolution of any trait or function. Life is an extremely complex functional network in evolution, with new links being created or broken, and new players appearing, while others disappear. The study of selenocysteine in insects provide a small, very partial snapshot of this process, yet clear enough to be understood. I am personally very interested in how we can exploit the large and increasing availability of sequenced genomes to understand how pathways evolve, and infer functional links in extant species. Current techniques for phylogenetic profiling are a very simple, yet successful way to do this. Shortly, the idea is to search for genes whose presence in a large number of genomes strongly correlate, which points to a functional link between them. I fancy the idea that, with more and more data becoming available, we will be able to build models for functional evolution at genomics scale, and arrive to predict whole networks just based on the observed pattern of gene loss and duplication. If I will ever develop in this direction, selenoproteins and Sec machinery will certainly be among the positive controls, for we already know both their history and their functional links.

4.3 Before and after ...

The most solid contribution of my PhD work is the development of computational tools for prediction of selenoprotein genes. The selenoprotein prediction server (<http://sebastian.crg.es/>) allows for the first time the selenoprotein research community (typically not very skilled in bioinformatics) to predict selenoprotein genes in custom input sequences.

Then, the program selenoprofiles basically solved the long standing program of selenoprotein annotation. Before I started, selenoprotein prediction had to be carried out manually, inspecting results one by one. Today, selenoprofiles can predict automatically the selenoproteome and Sec machinery content in any newly sequence genome, with good approximation. Also, this program constitutes a powerful framework for refining predictions, since its flexibility allow to easily tune parameters and get the desired genes in output. Automized analysis of nucleotide sequences has evidently many drawbacks when compared to manual curation, since important peculiarities may go unnoticed. Nonetheless, we think automatization is very necessary in this era, when sequence databases grow almost exponentially with time. Although small mistakes can arise in isolated cases, we gain power from the increased magnitude of results, and if the program is good enough, the trade-off is definitively worth.

The computational tools I developed were already the basis for a few comparative genomics projects, aiming at describing the phylogenetic history of genes that use or produce selenocysteine in lineages of interest. In particular, our research clarified the evolution of the selenoproteome in vertebrates, dissected the mechanism of Sec extinction in insects, and untangled the intricate history of SPS across the tree of life.

4.4 ... and next

In fields of research like biology, there is typically no clear end to projects, since what is yet to discover or understand inevitably surpasses the knowledge we hardly gained. In this section, I review the (possible) future developments of some of the projects described here.

4.4.1 Selenoprofiles as genome annotation tool

In this thesis, we have shown how this program has evolved from a relatively simple pipeline for selenoproteins, to a generic prediction tool for protein-coding genes at genomics scale. Notably, it is the only profile-based tool for genome annotation available. We expect this class of programs to become more and more common in the next years, because they generally exhibit both better performance, and better scalability than single sequence approaches. In fact, we can see the profiles as a conceptualization, or generalization, of certain biological elements (e.g. protein families). When more and more instances for that element are observed and included, the profile improves its recognition power. After so many examples, the learning curve typically approaches a plateau. Thus, maintaining profiles seems a reasonable strategy for computational tools acting on rapidly growing sequences, also for they can be used in an iterative fashion on new databases. In the next months, we plan to continue the process to make selenoprofiles a tool to fully annotate proteins in genomes. The prediction method itself is quite well established, also because it uses gene prediction tools which are standards in the field. What is left to develop is a collection of profiles allowing to predict all proteins in any genome, from any lineage. Rather than a fixed set of alignments, it is convenient to design a strategy to efficiently partition a comprehensive protein database (such as NCBI nr) in as many alignment as necessary. To this purpose, several aspects require careful consideration; for example, what criteria should be used to cluster sequences, how to align them, how to deal with fused proteins or common domains. Since we already had experience with them, the drosophila genomes will be a good test case. The quality of our annotations of Saltans and other drosophila genomes will also benefit from this process.

4.4.2 Future research on Willistoni/Saltans

The genomes that we sequenced from the Saltans group provide a useful resource for many research directions, which we have explored just very partially. The causes of the GC and codon usage shift in particular remain totally obscure. We plan to follow up on this, focusing on the factors that appear most intimately related: tRNAs, and tRNA modification enzymes. But also, we are designing methods to detect any peculiarities in these genomes when compared to other drosophila. The plan is to search for gene families significantly depleted, or expanded, or showing important change in expression, or with particular evolutionary

patterns such as marks for positive selection. We hope this may reveal characteristics potentially explaining their many peculiarities. In the future, I would like to develop generic methods to detect scan for genome features like those mentioned so that, given a new genome, we will have a way to automatically place it in its phylogenetic context, and highlight its innovations. Again, Saltans genomes will be like a pilot phase for this project.

4.4.3 The SPS story, and the unknown amino acid

The phylogenetic history of the SPS family in metazoans has revealed an interesting snapshot of protein evolution, and showed how readthrough can contribute to create new functions. Although the story we delineated make sense, there are still some aspects left to explain. The most intriguing one is the conservation of in-frame UGA in SPS1-rt genes of all Hymenoptera, a lineage that diverged some 250 Mya. We do not know what amino acid is inserted in this protein. Since this same gene was converted to arginine homologue at the root of all other Endopterygota, and also independently in pea aphid, the simplest explanation is that arginine is inserted. Nonetheless, it seems counterintuitive that a UGA codon is so tightly conserved in Hymenoptera just to insert a standard amino acid. Two possible arginine codons are just one point mutation away (CGA, AGA). Even without doing the math, just looking at the drop of conservation in introns it is evident that these organisms had plenty of chances to fix a mutation at this site. We must assume that the purifying selection here testifies another type of constraint. A possibility is that the readthrough mechanism is important for the regulation of the protein. If this is a limiting step during translation, the cell may be using it to express this protein only when needed. In this scenario, a mutation abolishing the stop codon could have a deep impact on fitness despite not changing the final protein product. By the way, I believe that this effect is extremely relevant for selenoprotein genes, for which it is known that translation at the Sec-UGA is a limiting step. Conservation of regulation may then be an important factor determining the purifying selection at the Sec positions of vertebrate selenoprotein genes, which thus cannot be ascribed only to low exchangeability of Sec and Cys.

It is plausible that the conservation of Hymenopteran SPS1-rt has another meaning. The UGA here may code for a non-standard amino acid, by either co-translational insertion, or by modification somehow triggered by the readthrough mechanism. The only way to know would be determining experimentally what amino is inserted there – which is something I'm personally considering.

Chapter 5

CONCLUSIONS

During my PhD, I developed computational tools for the characterization of selenoprotein genes in nucleotide sequences.

- I improved the program SECISearch, and created with Sebastian the first webserver for selenoprotein gene prediction;
- I wrote selenoprofiles, solving the problem of the annotation of selenoproteins at genomic scale. This assumes importance when viewed in the temporal context, a starting era of massive genome sequencing. The substantial expansion of SelenoDB with selenoprofiles predictions is a concrete example of its usefulness. Selenoprofiles is now a generic protein family annotation tool, that can be used also to accurately annotate proteomes in genomes.

Using these tools, I also actively participated to selenoprotein research in a few projects.

- I contributed to the annotation of the selenoproteins and Sec machinery genes in the human (gencode) and centipede genome;
- I characterized the content and evolution of the vertebrate selenoproteome, providing a phylogenetic context which has been already useful to many selenoprotein researchers;
- I followed the Sec extinctions in insects, first finding a novel one in pea aphid, and then analyzing in detail the Willistoni/Saltans group. From these observations, we derived a model of Sec extinction in drosophila;
- lastly, I traced the phylogenetic history of the SPS family across sequenced genomes, which revealed to contain an insightful snapshot of function evolution. Since SPS serves as a marker for selenium utilization (selenocysteine in proteins, selenouridine in tRNAs), this work also provided a phylogenetic map of these traits across the tree of life.

Bibliography

- Alsina, B., Corominas, M., Berry, M. J., Baguña, J., and Serras, F. (1999). Disruption of selenoprotein biosynthesis affects cell proliferation in the imaginal discs and brain of *Drosophila melanogaster*. *J Cell Sci*, 112 (Pt 1:2875–2884.
- Alsina, B., Serras, F., Baguna, J., and Corominas, M. (1998). patufet, the gene encoding the *Drosophila melanogaster* homologue of selenophosphate synthetase, is involved in imaginal disc morphogenesis. *Molecular and General Genetics MGG*, 257(2):113–123.
- Altschmied, J., Delfgaauw, J., Wilde, B., Duschl, J., Bouneau, L., Volff, J.-N., and Scharl, M. (2002). Subfunctionalization of duplicate mitf genes associated with differential degeneration of alternative exons in fish. *Genetics*, 161(1):259–67.
- Altschul, S., Madden, T., Schaffer, A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D. (1997). Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic acids research*, 25(17):3389.
- Andreesen, J. R. and Ljungdahl, L. G. (1973). Formate dehydrogenase of *Clostridium thermoaceticum*: incorporation of selenium-75, and the effects of selenite, molybdate, and tungstate on the enzyme. *Journal of bacteriology*, 116(2):867–73.
- Aphid-Consortium, T. I. G. (2010). Genome sequence of the pea aphid *Acyrtosiphon pisum*. *PLoS biology*, 8(2):e1000313.
- Bellen, H. J., Levis, R. W., Liao, G., He, Y., Carlson, J. W., Tsang, G., Evans-Holm, M., Hiesinger, P. R., Schulze, K. L., Rubin, G. M., Hoskins, R. A., and Spradling, A. C. (2004). The BDGP gene disruption project: single transposon insertions associated with 40% of *Drosophila* genes. *Genetics*, 167(2):761–81.
- Bermano, G., Arthur, J. R., and Hesketh, J. E. (1996). Role of the 3' untranslated region in the regulation of cytosolic glutathione peroxidase and phospholipid-hydroperoxide glutathione peroxidase gene expression by selenium supply. *The Biochemical journal*, 320 (Pt 3(Pt 3):891–5.
- Birney, E., Clamp, M., and Durbin, R. (2004). GeneWise and Genomewise. *Genome research*, 14(5):988–995.

- Böck, A., Forchhammer, K., Heider, J., Leinfelder, W., Sawers, G., Veprek, B., and Zinoni, F. (1991). Selenocysteine: the 21st amino acid. *Molecular microbiology*, 5(3):515–20.
- Burset, M. and Guigó, R. (1996). Evaluation of gene structure prediction programs. *Genomics*, 34(3):353–67.
- Cann, H. M., de Toma, C., Cazes, L., Legrand, M.-F., Morel, V., Piouffre, L., Bodmer, J., Bodmer, W. F., Bonne-Tamir, B., Cambon-Thomsen, A., Chen, Z., Chu, J., Carcassi, C., Contu, L., Du, R., Excoffier, L., Ferrara, G. B., Friedlaender, J. S., Groot, H., Gurwitz, D., Jenkins, T., Herrera, R. J., Huang, X., Kidd, J., Kidd, K. K., Langaney, A., Lin, A. A., Mehdi, S. Q., Parham, P., Piazza, A., Pistillo, M. P., Qian, Y., Shu, Q., Xu, J., Zhu, S., Weber, J. L., Greely, H. T., Feldman, M. W., Thomas, G., Dausset, J., and Cavalli-Sforza, L. L. (2002). A human genome diversity cell line panel. *Science (New York, N.Y.)*, 296(5566):261–2.
- Cantarel, B. L., Korf, I., Robb, S. M. C., Parra, G., Ross, E., Moore, B., Holt, C., Sánchez Alvarado, A., and Yandell, M. (2008). MAKER: an easy-to-use annotation pipeline designed for emerging model organism genomes. *Genome research*, 18(1):188–96.
- Carlson, B. A., Xu, X.-M., Kryukov, G. V., Rao, M., Berry, M. J., Gladyshev, V. N., and Hatfield, D. L. (2004). Identification and characterization of phosphoseryl-tRNA[Ser]^{Sec} kinase. *Proceedings of the National Academy of Sciences of the United States of America*, 101(35):12848–53.
- Cassago, A., Rodrigues, E. M., Prieto, E. L., Gaston, K. W., Alfonzo, J. D., Iribar, M. P., Berry, M. J., Cruz, A. K., and Thiemann, O. H. (2006). Identification of Leishmania selenoproteins and SECIS element. *Molecular and biochemical parasitology*, 149(2):128–34.
- Castellano, S., Andres, A., Bosch, E., Bayes, M., Guigo, R., and Clark, A. (2009). Low exchangeability of selenocysteine, the 21st amino acid, in vertebrate proteins. *Molecular biology and evolution*, 26(9):2031.
- Castellano, S., Gladyshev, V., Guigo, R., and Berry, M. (2008). SelenoDB 1.0: a database of selenoprotein genes, proteins and SECIS elements. *Nucleic acids research*, 36(Database issue):D332–8.
- Castellano, S., Lobanov, A., Chapple, C., Novoselov, S., Albrecht, M., Hua, D., Lescure, A., Lengauer, T., Krol, A., Gladyshev, V., and Others (2005). Diversity and functional plasticity of eukaryotic selenoproteins: identification and characterization of the SelJ family. *Proceedings of the National Academy of Sciences of the United States of America*, 102(45):16188.
- Castellano, S., Morozova, N., Morey, M., Berry, M., Serras, F., Corominas, M., and Guigó, R. (2001). In silico identification of novel selenoproteins in the *Drosophila melanogaster* genome. *EMBO reports*, 2(8):697.

- Castellano, S., Novoselov, S. V., Kryukov, G. V., Lescure, A., Blanco, E., Krol, A., Gladyshev, V. N., and Guigó, R. (2004). Reconsidering the evolution of eukaryotic selenoproteins: a novel nonmammalian family with scattered phylogenetic distribution. *EMBO reports*, 5(1):71–7.
- Chambers, I., Frampton, J., Goldfarb, P., Affara, N., McBain, W., and Harrison, P. (1986). The structure of the mouse glutathione peroxidase gene: the selenocysteine in the active site is encoded by the 'termination' codon, TGA. *The EMBO Journal*, 5(6):1221.
- Chapple, C. E. and Guigó, R. (2008). Relaxation of selective constraints causes independent selenoprotein extinction in insect genomes. *PLoS ONE*, 3.
- Chapple, C. E., Guigó, R., and Krol, A. (2009). SECISaln, a web-based tool for the creation of structure-based alignments of eukaryotic SECIS elements. *Bioinformatics (Oxford, England)*, 25(5):674–5.
- Chaudhuri, B. N. and Yeates, T. O. (2005). A computational method to predict genetically encoded rare amino acids in proteins. *Genome biology*, 6(9):R79.
- Chavatte, L., Brown, B., and Driscoll, D. (2005). Ribosomal protein L30 is a component of the UGA-selenocysteine recoding machinery in eukaryotes. *Nature structural & molecular biology*, 12(5):408–416.
- Chung, H., Yoon, S., Shin, S., Koh, Y., Lee, S., Lee, Y., and Bae, S. (2006). p53-mediated enhancement of radiosensitivity by selenophosphate synthetase 1 overexpression. *Journal of cellular physiology*, 209(1):131.
- Clamp, M., Cuff, J., Searle, S., and Barton, G. (2004). The jalview java alignment editor. *Bioinformatics*, 20(3):426.
- Cone, J. E., Del Río, R. M., Davis, J. N., and Stadtman, T. C. (1976). Chemical characterization of the selenoprotein component of clostridial glycine reductase: identification of selenocysteine as the organoselenium moiety. *Proceedings of the National Academy of Sciences of the United States of America*, 73(8):2659–63.
- Corona, M. and Robinson, G. E. (2006). Genes of the antioxidant system of the honey bee: annotation and phylogeny. *Insect Mol Biol*, 15(5):687–701.
- da Silva, M. T. A., Caldas, V. E. A., Costa, F. C., Silvestre, D. A. M. M., and Thiemann, O. H. (2013). Selenocysteine biosynthesis and insertion machinery in *Naegleria gruberi*. *Molecular and biochemical parasitology*, 188(2):87–90.
- Delsuc, F., Brinkmann, H., Chourrout, D., and Philippe, H. (2006). Tunicates and not cephalochordates are the closest living relatives of vertebrates. *Nature*, 439(7079):965–8.

- Dikiy, A., Novoselov, S., Fomenko, D., Sengupta, A., Carlson, B., Cerny, R., Ginalska, K., Grishin, N., Hatfield, D., and Gladyshev, V. (2007). SelT, SelW, SelH, and Rdx12: genomics and molecular insights into the functions of selenoproteins of a novel thioredoxin-like family. *Biochemistry*, 46(23):6871–6882.
- Ding, F. and Grabowski, P. J. (1999). Identification of a protein component of a mammalian tRNA(Sec) complex implicated in the decoding of UGA as selenocysteine. *RNA (New York, N.Y.)*, 5(12):1561–9.
- Dominguez, M., Ferres-Marco, D., Gutierrez-Aviño, F. J., Speicher, S. A., and Beneyto, M. (2004). Growth and specification of the eye are controlled independently by Eyegone and Eyeless in *Drosophila melanogaster*. *Nature genetics*, 36(1):31–9.
- Driscoll, D. M. and Chavatte, L. (2004). Finding needles in a haystack. In silico identification of eukaryotic selenoprotein genes. *EMBO reports*, 5(2):140–1.
- Drosophila-Consortium, T. (2007). Evolution of genes and genomes on the *Drosophila* phylogeny. *Nature*, 450(7167):203–18.
- Dudkiewicz, M., Szczepinska, T., Grynberg, M., and Pawlowski, K. (2012). A novel protein kinase-like domain in a selenoprotein, widespread in the tree of life. *PloS one*, 7(2):e32138.
- ENCODE-Consortium, T. (2011). A user’s guide to the encyclopedia of DNA elements (ENCODE). *PLoS biology*, 9(4):e1001046.
- ENCODE-Consortium, T. (2012). An integrated encyclopedia of DNA elements in the human genome. *Nature*, 489(7414):57–74.
- Fletcher, J. E., Copeland, P. R., Driscoll, D. M., and Krol, A. (2001). The selenocysteine incorporation machinery: interactions between the SECIS RNA and the SECIS-binding protein SBP2. *RNA (New York, N.Y.)*, 7(10):1442–53.
- Flohe, L., Gunzler, W., and Schock, H. (1973). Glutathione peroxidase: a selenoenzyme. *FEBS letters*, 32(1):132.
- Fomenko, D. E. and Gladyshev, V. N. (2012). Comparative genomics of thiol oxidoreductases reveals widespread and essential functions of thiol-based redox control of cellular processes. *Antioxidants & redox signaling*, 16(3):193–201.
- Fomenko, D. E., Xing, W., Adair, B. M., Thomas, D. J., and Gladyshev, V. N. (2007). High-throughput identification of catalytic redox-active cysteine residues. *Science (New York, N.Y.)*, 315(5810):387–9.
- Fujita, M., Mihara, H., Goto, S., Esaki, N., and Kanehisa, M. (2007). Mining prokaryotic genomes for unknown amino acids: a stop-codon-based approach. *BMC bioinformatics*, 8:225.

- Gautheret, D. and Lambert, A. (2001). Direct RNA motif definition and identification from multiple sequence alignments using secondary structure profiles. *Journal of molecular biology*, 313(5):1003–11.
- Gobler, C. J., Berry, D. L., Dyhrman, S. T., Wilhelm, S. W., Salamov, A., Lobanov, A. V., Zhang, Y., Collier, J. L., Wurch, L. L., Kustka, A. B., Dill, B. D., Shah, M., VerBerkmoes, N. C., Kuo, A., Terry, A., Pangilinan, J., Lindquist, E. A., Lucas, S., Paulsen, I. T., Hattenrath-Lehmann, T. K., Talmage, S. C., Walker, E. A., Koch, F., Burson, A. M., Marcoval, M. A., Tang, Y.-Z., Lecleir, G. R., Coyne, K. J., Berg, G. M., Bertrand, E. M., Saito, M. A., Gladyshev, V. N., and Grigoriev, I. V. (2011). Niche of harmful alga *Aureococcus anophagefferens* revealed through ecogenomics. *Proceedings of the National Academy of Sciences of the United States of America*, 108(11):4352–7.
- Gobler, C. J., Lobanov, A. V., Tang, Y.-Z., Turanov, A. A., Zhang, Y., Doblin, M., Taylor, G. T., Sañudo Wilhelmy, S. A., Grigoriev, I. V., and Gladyshev, V. N. (2013). The central role of selenium in the biochemistry and ecology of the harmful pelagophyte, *Aureococcus anophagefferens*. *The ISME journal*, doi:10.103.
- Gromer, S., Eubel, J. K., Lee, B. L., and Jacob, J. (2005). Human selenoproteins at a glance. *Cell Mol Life Sci*, 62(21):2414–2437.
- Gromer, S., Johansson, L., Bauer, H., Arscott, L., Rauch, S., Ballou, D., Williams, C., Schirmer, R., and Arnér, E. (2003). Active sites of thioredoxin reductases: Why selenoproteins? *Proceedings of the National Academy of Sciences of the United States of America*, 100(22):12618.
- Gruber, A. R., Findeiß, S., Washietl, S., Hofacker, I. L., and Stadler, P. F. (2010). RNAz 2.0: improved noncoding RNA detection. *Pacific Symposium on Biocomputing*. *Pacific Symposium on Biocomputing*, pages 69–79.
- Grundner-Culemann, E., Martin, G. W., Harney, J. W., and Berry, M. J. (1999). Two distinct SECIS structures capable of directing selenocysteine incorporation in eukaryotes. *RNA (New York, N.Y.)*, 5(5):625–35.
- Guigó, R., Knudsen, S., Drake, N., and Smith, T. (1992). Prediction of gene structure. *J Mol Biol*, 226(1):141–157.
- Guimarães, M. J., Peterson, D., Vicari, a., Cocks, B. G., Copeland, N. G., Gilbert, D. J., Jenkins, N. a., Ferrick, D. a., Kastelein, R. a., Bazan, J. F., and Zlotnik, a. (1996). Identification of a novel selD homolog from eukaryotes, bacteria, and archaea: is there an autoregulatory mechanism in selenocysteine metabolism? *Proceedings of the National Academy of Sciences of the United States of America*, 93(26):15086–91.
- Haft, D. H. and Self, W. T. (2008). Orphan SelD proteins and selenium-dependent molybdenum hydroxylases. *Biology direct*, 3:4.

- Harrell, L., Melcher, U., and Atkins, J. F. (2002). Predominance of six different hexanucleotide recoding signals 3' of read-through stop codons. *Nucleic Acids Res*, 30(9):2011–2017.
- Hirosawa-Takamori, M., Ossipov, D., Novoselov, S. V., Turanov, A. A., Zhang, Y., Gladyshev, V. N., Krol, A., Vorbrüggen, G., and Jäckle, H. (2009). A novel stem loop control element-dependent UGA read-through system without translational selenocysteine incorporation in *Drosophila*. *FASEB J*, 23(1):107–113.
- Hofacker, I., Fontana, W., Stadler, P., Bonhoeffer, S., Tacker, M., and P, S. (1994). Fast Folding and Comparison of RNA Secondary Structures. *Monatshefte f. Chemie*, 125:167–188.
- Holt, C. and Yandell, M. (2011). MAKER2: an annotation pipeline and genome-database management tool for second-generation genome projects. *BMC bioinformatics*, 12:491.
- Horibata, Y. and Hirabayashi, Y. (2007). Identification and characterization of human ethanolaminephosphotransferase1. *Journal of lipid research*, 48(3):503–8.
- Howard, M. T., Aggarwal, G., Anderson, C. B., Khatri, S., Flanigan, K. M., and Atkins, J. F. (2005). Recoding elements located adjacent to a subset of eukaryal selenocysteine-specifying UGA codons. *The EMBO journal*, 24(8):1596–607.
- Huerta-Cepas, J., Capella-Gutierrez, S., Pryszcz, L. P., Denisov, I., Kormes, D., Marcet-Houben, M., and Gabaldón, T. (2011). PhylomeDB v3.0: an expanding repository of genome-wide collections of trees, alignments and phylogeny-based orthology and paralogy predictions. *Nucleic acids research*, 39(Database issue):D556–60.
- Huerta-Cepas, J., Dopazo, J., and Gabaldón, T. (2010). ETE: a python Environment for Tree Exploration. *BMC bioinformatics*, 11:24.
- Hüttenhofer, A., Westhof, E., and Böck, A. (1996). Solution structure of mRNA hairpins promoting selenocysteine incorporation in *Escherichia coli* and their base-specific interaction with special elongation factor SELB. *RNA (New York, N.Y.)*, 2(4):354–66.
- Itoh, Y., Bröcker, M. J., Sekine, S.-i., Hammond, G., Suetsugu, S., Söll, D., and Yokoyama, S. (2013). Decameric Sela-tRNA(Sec) ring structure reveals mechanism of bacterial selenocysteine formation. *Science (New York, N.Y.)*, 340(6128):75–8.
- Itoh, Y., Sekine, S.-i., Matsumoto, E., Akasaka, R., Takemoto, C., Shirouzu, M., and Yokoyama, S. (2009). Structure of Selenophosphate Synthetase Essential for Selenium Incorporation into Proteins and RNAs. *Journal of molecular biology*, 385(5):1456–1469.

- Jiang, L., Liu, Q., and Ni, J. (2010). In silico identification of the sea squirt selenoproteome. *BMC genomics*, 11(1):289.
- Jiang, L., Ni, J., and Liu, Q. (2012). Evolution of selenoproteins in the metazoan. *BMC genomics*, 13:446.
- Jukes, T. H. (1990). Genetic code 1990. Outlook. *Experientia*, 46(11-12):1149–1157.
- Jungreis, I., Lin, M. F., Spokony, R., Chan, C. S., Negre, N., Victorsen, A., White, K. P., and Kellis, M. (2011). Evidence of abundant stop codon readthrough in *Drosophila* and other metazoa. *Genome research*, 21(12):2096–113.
- Kanzok, S. M., Fechner, A., Bauer, H., Ulschmid, J. K., Müller, H. M., Botella-Munoz, J., Schneuwly, S., Schirmer, R., and Becker, K. (2001). Substitution of the thioredoxin system for glutathione reductase in *Drosophila melanogaster*. *Science*, 291(5504):643–646.
- Katoh, K., Kuma, K.-i., Toh, H., and Miyata, T. (2005). MAFFT version 5: improvement in accuracy of multiple sequence alignment. *Nucleic acids research*, 33(2):511–8.
- Kim, H.-Y., Fomenko, D. E., Yoon, Y.-E., and Gladyshev, V. N. (2006). Catalytic advantages provided by selenocysteine in methionine-S-sulfoxide reductases. *Biochemistry*, 45(46):13697–704.
- Kim, H.-Y., Zhang, Y., Lee, B. C., Kim, J.-R., and Gladyshev, V. N. (2009). The selenoproteome of *Clostridium* sp. OhILAs: characterization of anaerobic bacterial selenoprotein methionine sulfoxide reductase A. *Proteins*, 74(4):1008–17.
- Kim, I. Y., Guimarães, M. J., Zlotnik, A., Bazan, J. F., and Stadtman, T. C. (1997). Fetal mouse selenophosphate synthetase 2 (SPS2): characterization of the cysteine mutant form overproduced in a baculovirus-insect cell system. *Proceedings of the National Academy of Sciences of the United States of America*, 94(2):418–21.
- Kim, J. Y., Lee, K. H., Shim, M. S., Shin, H., Xu, X.-M., Carlson, B. A., Hatfield, D. L., and Lee, B. J. (2010). Human selenophosphate synthetase 1 has five splice variants with unique interactions, subcellular localizations and expression patterns. *Biochemical and biophysical research communications*, 397(1):53–8.
- Krol, A. (2002). Evolutionarily different RNA motifs and RNA-protein complexes to achieve selenoprotein synthesis. *Biochimie*, 84(8):765–774.
- Kryukov, G., Castellano, S., Novoselov, S., Lobanov, A., Zehtab, O., Guigo, R., and Gladyshev, V. (2003). Characterization of mammalian selenoproteomes. *Science*, 300(5624):1439.

- Kryukov, G. and Gladyshev, V. (2004). The prokaryotic selenoproteome. *EMBO reports*, 5(5):538.
- Kryukov, G., Kumar, R., Koc, A., Sun, Z., and Gladyshev, V. (2002). Selenoprotein R is a zinc-containing stereo-specific methionine sulfoxide reductase. *Proceedings of the National Academy of Sciences of the United States of America*, 99(7):4245.
- Kryukov, G. V., Kryukov, V. M., and Gladyshev, V. N. (1999). New mammalian selenocysteine-containing proteins identified with an algorithm that searches for selenocysteine insertion sequence elements. *The Journal of Biological Chemistry*, 274(48):33888–97.
- Laferrière, A., Gautheret, D., and Cedergren, R. (1994). An RNA pattern matching program with enhanced performance and portability. *Computer applications in the biosciences : CABIOS*, 10(2):211–2.
- Lambert, A., Lescure, A., and Gautheret, D. (2002). A survey of metazoan selenocysteine insertion sequences. *Biochimie*, 84(9):953–9.
- Laslett, D. and Canback, B. (2004). ARAGORN, a program to detect tRNA genes and tmRNA genes in nucleotide sequences. *Nucleic acids research*, 32(1):11–6.
- Latrèche, L., Jean-Jean, O., Driscoll, D. M., and Chavatte, L. (2009). Novel structural determinants in human SECIS elements modulate the translational recoding of UGA as selenocysteine. *Nucleic acids research*, 37(17):5868–80.
- Lee, B. C., Dikiy, A., Kim, H.-Y., and Gladyshev, V. N. (2009a). Functions and evolution of selenoprotein methionine sulfoxide reductases. *Biochimica et biophysica acta*, 1790(11):1471–7.
- Lee, B. C., Lobanov, A. V., Marino, S. M., Kaya, A., Seravalli, J., Hatfield, D. L., and Gladyshev, V. N. (2011a). A four selenocysteine, two SECIS element methionine sulfoxide reductase from *Metridium senile* reveals a non-catalytic function of selenocysteines. *The Journal of biological chemistry*, 286(21):18747–18755.
- Lee, E., Harris, N., Gibson, M., Chetty, R., and Lewis, S. (2009b). Apollo: a community resource for genome annotation editing. *Bioinformatics (Oxford, England)*, 25(14):1836–7.
- Lee, K. H., Shim, M. S., Kim, J. Y., Jung, H. K., Lee, E., Carlson, B. A., Xu, X.-M., Park, J. M., Hatfield, D. L., Park, T., and Lee, B. J. (2011b). *Drosophila* selenophosphate synthetase 1 regulates vitamin B6 metabolism: prediction and confirmation. *BMC genomics*, 12:426.
- Leinfelder, W., Zehelein, E., MandrandBerthelot, M., and Bock, A. (1988). Gene for a novel tRNA species that accepts L-serine and cotranslationally inserts selenocysteine. *Nature*, 331:723–725.

- Lescure, A., Gautheret, D., Carbon, P., and Krol, A. (1999). Novel Selenoproteins Identified in Silico and in Vivo by Using a Conserved RNA Structural Motif. *Journal of Biological Chemistry*, 274(53):38147.
- Li, M., Huang, Y., and Xiao, Y. (2009). A Method for Identification of Selenoprotein Genes in Archaeal Genomes. *Genomics, Proteomics & Bioinformatics*, 7(1-2):62–70.
- Lobanov, A., Delgado, C., Rahlfs, S., Novoselov, S., Kryukov, G., Gromer, S., Hatfield, D., Becker, K., and Gladyshev, V. (2006a). The plasmodium selenoproteome. *Nucleic acids research*, 34(2):496.
- Lobanov, A., Gromer, S., Salinas, G., and Gladyshev, V. (2006b). Selenium metabolism in Trypanosoma: characterization of selenoproteomes and identification of a Kinetoplastida-specific selenoprotein. *Nucleic acids research*, 34(14):4012.
- Lobanov, A., Hatfield, D., and Gladyshev, V. (2008). Selenoproteinless animals: selenophosphate synthetase SPS1 functions in a pathway unrelated to selenocysteine biosynthesis. *Protein Science: A Publication of the Protein Society*, 17(1):176.
- Lobanov, A. V., Fomenko, D. E., Zhang, Y., Sengupta, A., Hatfield, D. L., and Gladyshev, V. N. (2007). Evolutionary dynamics of eukaryotic selenoproteomes: large selenoproteomes may associate with aquatic life and small with terrestrial life. *Genome biology*, 8(9):R198.
- Lobanov, A. V., Hatfield, D. L., and Gladyshev, V. N. (2009). Eukaryotic selenoproteins and selenoproteomes. *Biochimica et biophysica acta*, 1790(11):1424–8.
- Lobanov, A. V., Kryukov, G. V., Hatfield, D. L., and Gladyshev, V. N. (2006c). Is there a twenty third amino acid in the genetic code? *Trends Genet*, 22(7):357–360.
- Lowe, T. and Eddy, S. (1997). tRNAscan-SE: A program for improved detection of transfer RNA genes in genomic sequence. *Nucleic acids research*, 25(5):955.
- Luo, R., Liu, B., Xie, Y., Li, Z., Huang, W., Yuan, J., He, G., Chen, Y., Pan, Q., Liu, Y., Tang, J., Wu, G., Zhang, H., Shi, Y., Liu, Y., Yu, C., Wang, B., Lu, Y., Han, C., Cheung, D. W., Yiu, S.-M., Peng, S., Xiaoqian, Z., Liu, G., Liao, X., Li, Y., Yang, H., Wang, J., Lam, T.-W., and Wang, J. (2012). SOAPdenovo2: an empirically improved memory-efficient short-read de novo assembler. *GigaScience*, 1(1):18.
- Mariotti, M. and Guigó, R. (2010). Selenoprofiles: profile-based scanning of eukaryotic genome sequences for selenoprotein genes. *Bioinformatics (Oxford, England)*, 26(21):2656–63.

- Mariotti, M., Lobanov, A. V., Guigo, R., and Gladyshev, V. N. (2013). SE-CISearch3 and Seblastian: new tools for prediction of SECIS elements and selenoproteins. *Nucleic acids research*, 41(15):e149.
- Mariotti, M., Ridge, P. G., Zhang, Y., Lobanov, A. V., Pringle, T. H., Guigo, R., Hatfield, D. L., and Gladyshev, V. N. (2012). Composition and evolution of the vertebrate and mammalian selenoproteomes. *PloS one*, 7(3):e33066.
- Martin, G. W., Harney, J. W., and Berry, M. J. (1996). Selenocysteine incorporation in eukaryotes: insights into mechanism and efficiency from sequence, structure, and spacing proximity studies of the type 1 deiodinase SECIS element. *RNA (New York, N.Y.)*, 2(2):171–82.
- Martin-Romero, F., Kryukov, G., Lobanov, A., Carlson, B., Lee, B., Gladyshev, V., and Hatfield, D. (2001). Selenium metabolism in *Drosophila*. *Journal of Biological Chemistry*, 276(32):29798.
- Marygold, S. J., Leyland, P. C., Seal, R. L., Goodman, J. L., Thurmond, J., Strelets, V. B., and Wilson, R. J. (2013). FlyBase: improvements to the bibliography. *Nucleic acids research*, 41(Database issue):D751–7.
- Miroshnichenko, M. L., Kostrikina, N. A., Chernyh, N. A., Pimenov, N. V., Tourova, T. P., Antipov, A. N., Spring, S., Stackebrandt, E., and Bonch-Osmolovskaya, E. A. (2003). *Caldithrix abyssi* gen. nov., sp. nov., a nitrate-reducing, thermophilic, anaerobic bacterium isolated from a Mid-Atlantic Ridge hydrothermal vent, represents a novel bacterial lineage. *International journal of systematic and evolutionary microbiology*, 53(Pt 1):323–9.
- Missirlis, F., Rahlfs, S., Dimopoulos, N., Bauer, H., Becker, K., Hilliker, A., Phillips, J. P., and Jäcke, H. (2003). A putative glutathione peroxidase of *Drosophila* encodes a thioredoxin peroxidase that provides resistance against oxidative stress but fails to complement a lack of catalase activity. *Biological chemistry*, 384(3):463–72.
- Morey, M., Corominas, M., and Serras, F. (2003a). DIAP1 suppresses ROS-induced apoptosis caused by impairment of the selD/sps1 homolog in *Drosophila*. *Journal of cell science*, 116(Pt 22):4597–604.
- Morey, M., Serras, F., and Corominas, M. (2003b). Halving the selenophosphate synthetase gene dose confers hypersensitivity to oxidative stress in *Drosophila melanogaster*. *FEBS letters*, 534(1-3):111–114.
- Morozova, N., Forry, E. P., Shahid, E., Zavacki, A. M., Harney, J. W., Kraytsberg, Y., and Berry, M. J. (2003). Antioxidant function of a novel selenoprotein in *Drosophila melanogaster*. *Genes Cells*, 8(12):963–971.
- Nawrocki, E. P., Kolbe, D. L., and Eddy, S. R. (2009). Infernal 1.0: inference of RNA alignments. *Bioinformatics (Oxford, England)*, 25(10):1335–7.

- Notredame, C., Higgins, D. G., and Heringa, J. (2000). T-Coffee: A novel method for fast and accurate multiple sequence alignment. *Journal of molecular biology*, 302(1):205–17.
- Novoselov, S., Hua, D., Lobanov, A., and Gladyshev, V. (2006). Identification and characterization of Fep15, a new selenocysteine-containing member of the Sep15 protein family. *Biochemical Journal*, 394(Pt 3):575.
- Novoselov, S., Lobanov, A., Hua, D., Kasaikina, M., Hatfield, D., and Gladyshev, V. (2007). A highly efficient form of the selenocysteine insertion sequence element in protozoan parasites and its use in mammalian cells. *Proceedings of the National Academy of Sciences of the United States of America*, 104(19):7857–62.
- Novoselov, S., Rao, M., Onoshko, N., Zhi, H., Kryukov, G., Xiang, Y., Weeks, D., Hatfield, D., and Gladyshev, V. (2002). Selenoproteins and selenocysteine insertion system in the model plant cell system, *Chlamydomonas reinhardtii*. *The EMBO Journal*, 21(14):3681.
- Obata, T. and Shiraiwa, Y. (2005). A novel eukaryotic selenoprotein in the haptophyte alga *Emiliania huxleyi*. *Journal of Biological Chemistry*, 280(18):18462.
- Ogasawara, Y., Lacourciere, G., and Stadtman, T. C. (2001). Formation of a selenium-substituted rhodanese by reaction with selenite and glutathione: possible role of a protein perselenide in a selenium delivery system. *Proceedings of the National Academy of Sciences of the United States of America*, 98(17):9494–8.
- Pacheco, T. R., Gomes, A. Q., Barbosa-Morais, N. L., Benes, V., Ansorge, W., Wollerton, M., Smith, C. W., Valcárcel, J., and Carmo-Fonseca, M. (2004). Diversity of vertebrate splicing factor U2AF35: identification of alternatively spliced U2AF1 mRNAs. *The Journal of biological chemistry*, 279(26):27039–49.
- Palenik, B., Grimwood, J., Aerts, A., Rouzé, P., Salamov, A., Putnam, N., Dupont, C., Jorgensen, R., Derelle, E., Rombauts, S., Zhou, K., Otiillar, R., Merchant, S. S., Podell, S., Gaasterland, T., Napoli, C., Gendler, K., Manuell, A., Tai, V., Vallon, O., Piganeau, G., Jancek, S., Heijde, M., Jabbari, K., Bowler, C., Lohr, M., Robbens, S., Werner, G., Dubchak, I., Pazour, G. J., Ren, Q., Paulsen, I., Delwiche, C., Schmutz, J., Rokhsar, D., Van De Peer, Y., Moreau, H., and Grigoriev, I. V. (2007). The tiny eukaryote *Ostreococcus* provides genomic insights into the paradox of plankton speciation. *Proceedings of the National Academy of Sciences of the United States of America*, 104(18):7705–10.
- Palioura, S., Sherrer, R. L., Steitz, T. A., Söll, D., and Simonovic, M. (2009). The human SepSecS-tRNA^{Sec} complex reveals the mechanism of selenocysteine formation. *Science*, 325(5938):321–325.

- Parra, G., Blanco, E., and Guigó, R. (2000). GeneID in Drosophila. *Genome Res*, 10(4):511–515.
- Persson, B. C., Böck, A., Jäckle, H., and Vorbrüggen, G. (1997). SelD homolog from Drosophila lacking selenide-dependent monoselenophosphate synthetase activity. *Journal of molecular biology*, 274(2):174–80.
- Powell, J. R., Sezzi, E., Moriyama, E. N., Gleason, J. M., and Caccone, A. (2003). Analysis of a shift in codon usage in Drosophila. *Journal of molecular evolution*, 57 Suppl 1:S214–25.
- Riddle, N. C., Shaffer, C. D., and Elgin, S. C. R. (2009). A lot about a little dot - lessons learned from Drosophila melanogaster chromosome 4. *Biochemistry and cell biology = Biochimie et biologie cellulaire*, 87(1):229–41.
- Rodríguez-Trelles, F., Tarrío, R., and Ayala, F. J. (1999). Switch in codon bias and increased rates of amino acid substitution in the Drosophila saltans species group. *Genetics*, 153(1):339–50.
- Romero, H., Zhang, Y., Gladyshev, V. N., and Salinas, G. (2005). Evolution of selenium utilization traits. *Genome biology*, 6(8):R66.
- Rother, M., Mathes, I., Lottspeich, F., and Böck, A. (2003). Inactivation of the selB gene in Methanococcus maripaludis: effect on synthesis of selenoproteins and their sulfur-containing homologs. *Journal of bacteriology*, 185(1):107–14.
- Rother, M., Resch, A., Wilting, R., and Böck, A. (2001). Selenoprotein synthesis in archaea. *BioFactors (Oxford, England)*, 14(1-4):75–83.
- Rother, M., Wilting, R., Commans, S., and Böck, A. (2000). Identification and characterisation of the selenocysteine-specific translation factor SelB from the archaeon Methanococcus jannaschii. *Journal of molecular biology*, 299(2):351–8.
- Sayers, E. W., Barrett, T., Benson, D. A., Bolton, E., Bryant, S. H., Canese, K., Chetvernin, V., Church, D. M., Dicuccio, M., Federhen, S., Feolo, M., Geer, L. Y., Helmberg, W., Kapustin, Y., Landsman, D., Lipman, D. J., Lu, Z., Madden, T. L., Madej, T., Maglott, D. R., Marchler-Bauer, A., Miller, V., Mizrachi, I., Ostell, J., Panchenko, A., Pruitt, K. D., Schuler, G. D., Sequeira, E., Sherry, S. T., Shumway, M., Sirotkin, K., Slotta, D., Souvorov, A., Starchenko, G., Tatusova, T. A., Wagner, L., Wang, Y., Wilbur, W. J., Yaschenko, E., and Ye, J. (2009). Database resources of the National Center for Biotechnology Information. *Nucleic Acids Research*.
- Schomburg, L. and Schweizer, U. (2009). Hierarchical regulation of selenoprotein expression and sex-specific effects of selenium. *BBA - General Subjects*.

- Shchedrina, V., Novoselov, S., Malinouski, M., and Gladyshev, V. (2007). Identification and characterization of a selenoprotein family containing a diselenide bond in a redox motif. *Proceedings of the National Academy of Sciences*, 104(35):13919.
- Shchedrina, V. a., Everley, R. a., Zhang, Y., Gygi, S. P., Hatfield, D. L., and Gladyshev, V. N. (2011). Selenoprotein K binds multiprotein complexes and is involved in the regulation of endoplasmic reticulum homeostasis. *The Journal of biological chemistry*, 286(50):42937–48.
- Sheppard, K., Yuan, J., Hohn, M. J., Jester, B., Devine, K. M., and Söll, D. (2008). From one amino acid to another: tRNA-dependent amino acid biosynthesis. *Nucleic acids research*, 36(6):1813–25.
- Singh, N. D., Arndt, P. F., and Petrov, D. A. (2006). Minor shift in background substitutional patterns in the *Drosophila saltans* and *willistoni* lineages is insufficient to explain GC content of coding sequences. *BMC biology*, 4:37.
- Slater, G. S. C. and Birney, E. (2005). Automated generation of heuristics for biological sequence comparison. *BMC Bioinformatics*, 6:31.
- Small-Howard, A., Morozova, N., Stoytcheva, Z., Forry, E. P., Mansell, J. B., Harney, J. W., Carlson, B. A., Xu, X.-M., Hatfield, D. L., and Berry, M. J. (2006). Supramolecular complexes mediate selenocysteine incorporation in vivo. *Molecular and cellular biology*, 26(6):2337–46.
- Squires, J. and Berry, M. (2008). Eukaryotic Selenoprotein Synthesis: Mechanistic Insight Incorporating New Factors and New Functions for Old Factors. *IUBMB life*, 60(4):232–235.
- Srivastava, M., Mallard, C., Barke, T., Hancock, L. E., and Self, W. T. (2011). A selenium-dependent xanthine dehydrogenase triggers biofilm proliferation in *Enterococcus faecalis* through oxidant production. *Journal of bacteriology*, 193(7):1643–52.
- Stenvall, J., Fierro-González, J. C., Swoboda, P., Saamarthy, K., Cheng, Q., Cachovaladez, B., Arnér, E. S. J., Persson, O. P., Miranda-Vizueté, A., and Tuck, S. (2011). Selenoprotein TRXR-1 and GSR-1 are essential for removal of old cuticle during molting in *Caenorhabditis elegans*. *Proceedings of the National Academy of Sciences of the United States of America*, 108(3):1064–9.
- Stock, T. and Rother, M. (2009). Selenoproteins in Archaea and Gram-positive bacteria. *Biochimica et Biophysica Acta (BBA)-General Subjects*, 1790(11):1520–1532.
- Su, D., Ojo, T. T., Söll, D., and Hohn, M. J. (2012). Selenomodification of tRNA in archaea requires a bipartite rhodanese enzyme. *FEBS letters*, 586(6):717–21.

- Sun, X., Yang, Q., and Xia, X. (2013). An improved implementation of effective number of codons (nc). *Molecular biology and evolution*, 30(1):191–6.
- Talavera, D., Vogel, C., Orozco, M., Teichmann, S. A., and de la Cruz, X. (2007). The (in)dependence of alternative splicing and gene duplication. *PLoS computational biology*, 3(3):e33.
- Tamura, T., Yamamoto, S., Takahata, M., Sakaguchi, H., Tanaka, H., Stadtman, T. C., and Inagaki, K. (2004). Selenophosphate synthetase genes from lung adenocarcinoma cells: Sps1 for recycling L-selenocysteine and Sps2 for selenite assimilation. *Proceedings of the National Academy of Sciences of the United States of America*, 101(46):16162–7.
- Taskov, K., Chapple, C., Kryukov, G. V., Castellano, S., Lobanov, A. V., Korotkov, K. V., Guigó, R., and Gladyshev, V. N. (2005). Nematode selenoproteome: the use of the selenocysteine insertion system to decode one codon in an animal genome? *Nucleic acids research*, 33(7):2227–38.
- Tujebajeva, R., Copeland, P., Xu, X., and Carlson, B. (2000). Decoding apparatus for eukaryotic selenocysteine insertion. *EMBO reports*, 1(2):158.
- Turner, D. C. and Stadtman, T. C. (1973). Purification of protein components of the clostridial glycine reductase system and characterization of protein A as a selenoprotein. *Archives of biochemistry and biophysics*, 154(1):366–81.
- Van Hoewyk, D. (2013). A tale of two toxicities: malformed selenoproteins and oxidative stress both contribute to selenium stress in plants. *Annals of botany*.
- Vicario, S., Moriyama, E. N., and Powell, J. R. (2007). Codon usage in twelve species of *Drosophila*. *BMC evolutionary biology*, 7:226.
- Villanueva-Cañas, J. L., Laurie, S., and Albà, M. M. (2013). Improving genome-wide scans of positive selection by using protein isoforms of similar length. *Genome biology and evolution*, 5(2):457–67.
- Wang, K., Wang, J., Li, L., and Su, X. (2009). Crystal Structures of Catalytic Intermediates of Human Selenophosphate Synthetase 1. *Journal of molecular biology*, 390(4):747–759.
- Whanger, P. D. (2002). Selenocompounds in plants and animals and their biological significance. *Journal of the American College of Nutrition*, 21(3):223–32.
- Wilting, R., Schorling, S., Persson, B., and Böck, A. (1997a). Selenoprotein synthesis in archaea: identification of an mRNA element of *Methanococcus jannaschii* probably directing selenocysteine insertion1. *Journal of molecular biology*, 266(4):637–641.

- Wilting, R., Schorling, S., Persson, B. C., and Böck, A. (1997b). Selenoprotein synthesis in archaea: identification of an mRNA element of *Methanococcus jannaschii* probably directing selenocysteine insertion. *Journal of molecular biology*, 266(4):637–41.
- Wittwer, A. J. and Ching, W. M. (1989). Selenium-containing tRNA(Glu) and tRNA(Lys) from *Escherichia coli*: purification, codon specificity and translational activity. *BioFactors (Oxford, England)*, 2(1):27–34.
- Wright, F. (1990). The 'effective number of codons' used in a gene. *Gene*, 87(1):23–29.
- Wu, R., Shen, Q., and Newburger, P. E. (2000). Recognition and binding of the human selenocysteine insertion sequence by nucleolin. *Journal of cellular biochemistry*, 77(3):507–16.
- Xu, X., Carlson, B., Irons, R., Mix, H., Zhong, N., Gladyshev, V., and Hatfield, D. (2007a). Selenophosphate synthetase 2 is essential for selenoprotein biosynthesis. *Biochemical Journal*, 404(Pt 1):115.
- Xu, X., Carlson, B., Mix, H., Zhang, Y., Saira, K., Glass, R., Berry, M., Gladyshev, V., and Hatfield, D. (2007b). Biosynthesis of selenocysteine on its tRNA in eukaryotes. *PLoS Biol*, 5(1):e4.
- Xu, X. M., Mix, H., Carlson, B. A., Grabowski, P. J., Gladyshev, V. N., and Berry, M. J. (2005). Evidence for direct roles of two additional factors, SECp43 and soluble liver antigen, in the selenoprotein synthesis machinery. *Biol. Chem.*, 280:41568–41575.
- Yoshizawa, S. and Böck, A. (2009). The many levels of control on bacterial selenoprotein synthesis. *Biochimica et Biophysica Acta (BBA)-General Subjects*, 1790(11):1404–1414.
- Yuan, J., O'Donoghue, P., Ambrogelly, A., Gundllapalli, S., Sherrer, R. L., Palioura, S., Simonović, M., and Söll, D. (2010). Distinct genetic code expansion strategies for selenocysteine and pyrrolysine are reflected in different aminoacyl-tRNA formation systems. *FEBS letters*, 584(2):342–9.
- Zhang, Y. and Gladyshev, V. N. (2005). An algorithm for identification of bacterial selenocysteine insertion sequence elements and selenoprotein genes. *Bioinformatics (Oxford, England)*, 21(11):2580–9.
- Zhang, Y. and Gladyshev, V. N. (2008). Trends in selenium utilization in marine microbial world revealed through the analysis of the global ocean sampling (GOS) project. *PLoS genetics*, 4(6):e1000095.
- Zhang, Y. and Gladyshev, V. N. (2010). dbTEU: a protein database of trace element utilization. *Bioinformatics (Oxford, England)*, 26(5):700–2.

- Zhang, Y., Romero, H., Salinas, G., and Gladyshev, V. (2006). Dynamic evolution of selenocysteine utilization in bacteria: a balance between selenoprotein loss and evolution of selenocysteine from redox active cysteine residues. *Genome Biology*, 7(10):R94.
- Zhang, Y., Turanov, A., Hatfield, D., and Gladyshev, V. (2008). In silico identification of genes involved in selenium metabolism: evidence for a third selenium utilization trait. *BMC genomics*, 9(1):251.
- Zhong, L. and Holmgren, A. (2000). Essential role of selenium in the catalytic activities of mammalian thioredoxin reductase revealed by characterization of recombinant enzymes with selenocysteine mutations. *The Journal of biological chemistry*, 275(24):18121–8.

Selenoprofiles 3 manual

Pipeline for profile-based protein finding in genomes



Last update: September 24th 2013
selenoprofiles version 3.0c

contact: marco.mariotti@crg.eu

Table of contents

Introduction	4
Installation	5
Getting started	6
<i>The pipeline in summary</i>	<i>7</i>
<i>Building a profile</i>	<i>8</i>
<i>Configuration file vs command line</i>	<i>10</i>
<i>The results folder</i>	<i>12</i>
<i>Output options</i>	<i>13</i>
<i>The results database</i>	<i>13</i>
<i>Inspecting results: .p2g format</i>	<i>14</i>
<i>Searching multiple targets</i>	<i>15</i>
The Selenoprofiles pipeline	18
<i>Psitblastn</i>	<i>18</i>
<i>Exonerate</i>	<i>19</i>
<i>Genewise</i>	<i>21</i>
<i>Improving predictions</i>	<i>22</i>
<i>Prediction program choice</i>	<i>23</i>
<i>Labeling</i>	<i>24</i>
<i>Final filtering</i>	<i>24</i>
<i>Removing inter-family redundancy</i>	<i>25</i>
<i>Running selenoprofiles in parallel</i>	<i>25</i>
Advanced usage	27
<i>The p2ghit class</i>	<i>27</i>
<i>Custom output: option -fasta_add</i>	<i>29</i>
<i>Actions</i>	<i>29</i>
<i>Blast filtering</i>	<i>30</i>

<i>AWSI Z-score based filtering</i>	31
<i>Other filtering functions</i>	33
<i>Tag blast filtering</i>	33
<i>GO score filtering</i>	34
<i>Integrate your own code: option -add</i>	34
<i>Custom prediction features</i>	37
Appendix 1: guide to profile building	40
Appendix 2: full list of operations	42
Appendix 3: links and references	43
Appendix 4: troubleshooting	44
<i>Blast error</i>	44
<i>Genewise errors</i>	44

If this program is useful to your research, please cite:

Mariotti M, Guigó R (2010)

Selenoprofiles: profile-based scanning of eukaryotic genome sequences for selenoprotein genes. Bioinformatics. 2010 Nov 1;26(21):2656-63. Epub 2010 Sep 21

Introduction

Selenoprofiles is a pipeline for profile-based protein prediction in genomes. The program takes two inputs per run:

- one or more profile alignments, representing the protein families to search for,
- a genome (or any other nucleotide database), the target you want to scan.

Selenoprofiles runs internally a number of "slave" programs, whose predictions are analyzed and combined. The main programs used are: blast (psitblastn flavor, from blastall ncbi package), exonerate (utilized in protein-to-genome mode) and genewise. All these programs, although different in the algorithm and in speed, are based on the same principle: the target (nucleotide) is translated in all possible frames, and the query (protein) is aligned to such translated sequences, searching for high-scoring matches. The procedures of exonerate and genewise include also the prediction of splice sites, to bridge the matches into more complete, multi-exonic gene predictions.

Selenoprofiles use blast as first step, and attempts to refine its predictions with exonerate and genewise. It then processes the candidate gene structures, finally producing non-overlapping gene predictions for all input profiles.

The main purpose of selenoprofiles is the accurate search of a set protein families in a wide range of sequenced species. Nonetheless, it has been used also for the complete annotation of genomes. In this case a comprehensive, large set of input profiles has to be provided. A virtue of selenoprofiles is flexibility: its workflow can be substantially modified using options and configuration files, allowing in particular a finely tuned filtering of results. Also, the user can also easily plug-in its own code for specific annotations and analysis. Finally, the selenoprofiles package includes a few additional programs to collect and visualize the results of searches along the phylogenetic tree of target species.

Selenoprofiles can be used with any input protein family, but we initially developed it for selenoproteins. These peculiar proteins contain a selenocysteine, the 21st amino acid. Selenocysteine (Sec, or U) is inserted in correspondence to specific UGA codons, which normally signal translation termination. In selenoprotein transcripts we find specific secondary structures (SECIS elements), which targets a specific UGA to be read as Sec instead that as a stop. Since selenoproteins possess this peculiar feature (recoding of specific stop codons), normal gene prediction programs fail to predict them. Selenoprofiles in contrast is able to correctly include selenocysteine positions, by using technical expedients detailed in this manual. Selenoprofiles includes built-in profiles for selenoproteins and other proteins related to selenocysteine, allowing out-of-the-box prediction of these families.

This manual describes the selenoprofiles pipeline starting from the simplest usage, moving then to most complex customization methods. It covers almost the totality of selenoprofiles options. The full list can be inspected running *Selenoprofiles --help full*.

The pipeline is also described in a paper in Bioinformatics (see [references on Appendix 3](#)), in which we also detail how we validated the method. Note that the paper refers to the version 1, while here we describe version 3, with several major improvements.

Installation

Selenoprofiles can be installed on any unix system with python 2.6 or newer. A python command line installer (*install_selenoprofiles.py*) is provided inside the installation package that you can find at <http://big.crg.cat/services/selenoprofiles>. The user needs to take care of the installation of all slave programs: ncbi blast package 2.2.18¹, exonerate version 2.0.0 or newer, genewise from the Wise2 package, and also mafft. These programs have to be available in the bash environment for the installer to work. Find useful links for their installation in [Appendix 3](#). If you experience any problem with their installation, visit [Appendix 4, troubleshooting](#). Selenoprofiles needs also the ncbi taxonomy database, to assign species names. The installer will attempt to fetch it if it is not provided.

Selenoprofiles provides a wide range of filtering functions, some of which scan a protein database (ncbi nr) to search the candidate sequences with blastp, and parse results to infer the goodness of the prediction. Since some of the built-in profiles for selenoproteins and Sec machinery feature this kind of filtering, the database is needed for their use. The blast nr database is large (>3 Gb) and it may take a long time to download it.

If you plan to scan for your custom families, and you do not need to use the built-in profiles, you may want to skip this step, and perform a minimal installation (*python install_selenoprofiles.py -min*). The installation script skips the download of a GO annotation of nr sequences, and skips the system search for the program SECISearch3, an external program for the prediction of SECIS elements, secondary structures peculiar to selenoproteins.

If instead you plan to use selenoprofiles to scan for selenoproteins and Sec machinery, you have to perform a full installation. If you already have ncbi nr on your system, you can link it using installer option *-nrdb* (see *install_selenoprofiles.py --help*).

After installation, you can test it using script *test_selenoprofiles.py*, located inside the installation directory. This script runs the pipeline on a few test sequences and checks that the output is as expected. You can also run anytime *selenoprofiles -test* to perform a presence check of all slave programs and modules used either by selenoprofiles, or by the additional programs included for visualization.

In particular, *selenoprofiles_build_profile.py* requires Pylab (<http://www.scipy.org/PyLab>) to plot the sequence identity characteristics of profiles, and *selenoprofiles_tree_drawer.py* requires ete2 (<http://ete.cgenomics.org/>) for tree-based visualization of results across species. Although none of this two modules is compulsory, we strongly suggest to install ete2 for projects aimed at searching certain protein families in a wide range of species, to conveniently visualize results as an annotated species tree.

¹ All 2.2.x versions are expected to work. The newer versions, called blast+, will not work

Getting started

This chapter will cover the basic use of selenoprofiles. To begin, we will use a profile alignment included in selenoprofiles package. Let's get practical. Let's say that we want to scan the genome of the species *Macaca mulatta*, contained in the file */db/genome.fasta*, for the built-in AhpC profile.

Here's a basic command line:

```
Selenoprofiles results_folder -t /db/genome.fasta -s "Macaca_mulatta" -p AhpC
```

The first argument of selenoprofiles is the folder where all results will be stored. If not existing, it will be created. It will be called **results folder** from now on.

The second argument, provided with option *-t*, is the **target file**. A multi-fasta file must be provided. This is formatted with formatdb and fastaindex to be used by the slave programs. The file name, without the extension, is used for naming in selenoprofiles and will be referenced as the target name (in the example, *genome*). Each short title (defined as the first word in a fasta header) must be unique, and no empty sequence should be present. The option **species** (or *-s*) allows to specify to which organism the genome belongs to. The species name provided will be searched into the ncbi taxonomy database, from where a taxid will be derived. The definition of the species is highly recommended but not compulsory: if none is specified or it is not found in ncbi, the species will be set to *unidentified*. Note that the combination of species name and target name must be unique in a given results folder.

The other key argument to the program is the **profile**, or the profiles, that will be searched in the genome. If none is specified, the list of profiles is read from the configuration file, which defaults to the selenoproteins and Sec machinery families. The option *-profile* (or *-p* or *-P*) can accept multiple arguments, that must be comma separated with no space within. Each such argument can be the name of profile (which is searched into the profiles folder), the path to a profile fasta alignment, or a keyword indicating a list of families defined in the main configuration file. When a family alignment is provided for the first time to selenoprofiles, its profile is built on the fly (see [building a profile](#)).

For example, to scan the same genome with two custom profiles alignments you can use:

```
Selenoprofiles results_folder -t /db/genome.fasta -s "Macaca_mulatta" \
-p /somewhere/profiles/family1.fa,/somewhere/profiles/family2.fa
```

Or alternatively, defining the profiles folder in the command line:

```
Selenoprofiles results_folder -t /db/genome.fasta -s "Macaca_mulatta"
-profiles_folder /somewhere/profiles/ -p family1,family2
```

By default, selenoprofiles executes the full pipeline. The final output files will be found inside the results folder, inside the target subfolder, in a folder called output. For the example above, this folder would be:

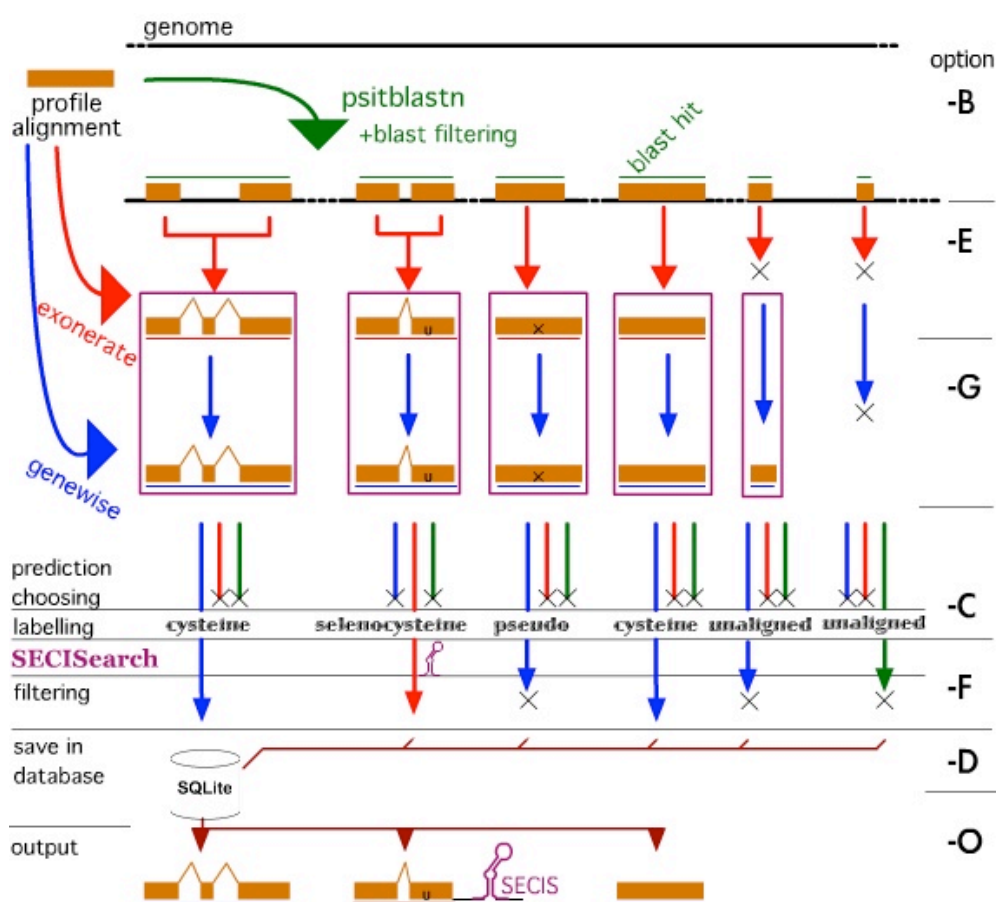
```
results_folder/Macaca_mulatta.genome/output/
```

The pipeline in summary

The pipeline workflow is detailed in the [next section](#), and it is here summarized (see also figure below). The program `psitblastn` is used with a PSSM derived from the profile alignment to identify matches in the target genome. These matches are then used, through the two splice alignment programs `exonerate` and `genewise`, to deduce the exonic structure of the candidate genes. The predictions of these three programs are analyzed to choose one, which is then labelled through a dedicated procedure.

Through the entire pipeline a number of steps are performed to filter out likely false positives and to keep the number of potential candidates under manageable levels. There are three layers of filtering: at the top the blast filtering, which controls how many gene candidates will be processed. Then the (p2g) filtering and (p2g) refiltering, both of which are at the end of the pipeline. All filtering steps are user definable, which can create filters adapted to his/her protein family of interest. We provide a sensible default filtering for user input families: each alignment is examined and, based on its sequence conservation, a similarity threshold is chosen. This means that a very conserved profile will output only very similar sequences. Also, when multiple profiles are searched, overlapping matches are assigned to one or the other family based on sequence similarity.

For selenoprotein families, the program `SECISearch3` (if installed) is also used to identify suitable SECIS elements downstream of the coding region of the candidate selenoprotein genes. The workflow of selenoprofiles can be easily customized to perform similar operations: running custom code for specific gene candidates, then storing and outputting genomic annotations (see [custom features](#) in the advanced usage section).



Graphical summary of the selenoprofiles pipeline.

Selenoprofiles normally performs the full pipeline, taking care of skipping the steps executed previously. The steps of selenoprofiles are: blast, exonerate, genewise, prediction choice, prediction filtering, output, denoted respectively by the step-options *-B -E -G -C -F -D -O* (see figure). After the filtering step, results are stored in a SQLite database. When selenoprofiles is run, it checks first if the results database contain already the results, and if it does, it passes directly to the output step. If the user specify any step-option, the execution of the corresponding step and of all next ones is forced. This is necessary if you changed parameters or profile specific procedures. If for example you changed some parameter relative to the filtering phase, you can force filtering and output with *-F*. **Important:** even when output is forced, selenoprofiles will overwrite previous output files, but it will never delete any. This may lead to overlapping predictions in the output, thus we recommend to always delete the output files before any second run on a certain genome. For the full chronological list of operations performed by Selenoprofiles, see [Appendix 2](#).

Building a profile

A profile alignment is a set of aligned sequences which allows to find and predict genes that *fit* in it. This source of information is used in different forms by the slave programs to find regions of homology and model the genes found in the target.

Building a profile alignment means formatting it to be used with selenoprofiles. You just need a sequence alignment named after your family, with only alphanumeric characters or underscores. The only format accepted is fasta (aligned, with gaps as “-”). The title names must have a unique starting word.

When you provide a fasta file as profile argument, selenoprofiles will attempt to build it with default options. Optionally, you can use the script *selenoprofiles_build_profile.py* (located inside the installation directory) to build the profile before running selenoprofiles. This script allow to control profile-specific parameters and procedures, using the library of functions described in this manual. It also provides other utilities, such as a tool to trim redundant sequences. For fast runs, alignments should be trimmed to less than 100 sequences. Small profiles are also discouraged, since the variation in profile sequence similarity is an important determinant for filtering. A minimum of 10 sequences is suggested. For a guide to build a good profile, see [Appendix 1](#).

When a profile is built, its sequences are reordered (overwriting the input file) and two files are produced: a *.profile_data* file, containing data derived from its sequences for lazy computing, and a *.config* file, with all the non-sequence information associated to this profile. The sequences are ordered based on “completeness” respect to the whole profile. The *.config* file can be inspected and edited with any text editor to modify the profile attributes. Its content can vary a lot, since all the attributes that are not found in there are taken from the selenoprofiles main configuration file.

The only options in the *.config* file that the user typically wants to check are the filtering procedures. By default, a loose blast filtering is used (evalue < 0.01). For p2g filtering, only predictions spanning at least 40% of the profile length (or longer than 60 aminoacids) are kept. In the last layer of filtering (p2g refiltering), the AWSI measure is evaluated. As explained later (see [AWSI score](#)), this method computes a score of average similarity of the candidate with all profile sequences, and compares it with the average similarity within the profile itself. In this way, very conserved profile alignments will output only very conserved genes. The user can modify the filtering procedures by adding (or editing) lines in the profile *.config* files. It is also possible to edit the default values in the main configuration file, affecting all profiles with no procedures defined in their *.config* file.

For example, to tighten up the blast filtering for a certain family, you can include this line in its *.config* file:

```
blast_filtering = x.evalue < 1e-8
```

To modify the default p2g_filtering for all profiles, find and edit the line corresponding to this in the pipeline main configuration file (*selenoprofiles.config*):

```
p2g_filtering.DEFAULT = x.coverage()>0.5 and x.label != 'pseudo'
```

This will require the predictions to span at least half of the profile width, and to possess a label different than pseudo. A single label is assigned to each result during the pipeline workflow. The labeling procedure can also be customized (see [option -add](#) section). By default, there are only two possible labels: *pseudo* (assigned to all results with in-frame stop codons, or with insertions or deletions creating frameshifts), and *homologue* (assigned to all others).

Other elements in the profile configuration file

Let's inspect an example of a built-in profile: AhpC. Its *.config* file contains:

```
name = AhpC
queries = all
blast_options = SELENO
exonerate_options = SELENO
genewise_options = SELENO
```

- *name*: the name of the family. Taken from the input file name.
- *queries*: the queries in a profile are those eligible to be used with exonerate and genewise. In a well curated, clean alignment, all sequences are queries. The value of the *queries* attribute can accept various formats (see *selenoprofiles_build_profile.py --help*), but normally you won't need to change it from its default value, *all*. Just for selenoproteins, it is important to take particular care on the alignment of the position(s) with selenocysteine. Thus, by default a sequence is excluded from the queries if it has no residue aligned to the position of selenocysteine in the alignment, or to any of them if there are many such positions.

All other elements may or not be present in the file. In the case they are not, they are set to the defaults specified in the selenoprofiles main configuration file. All these options can be controlled by keywords. Keywords are defined in the main configuration file, in the form:

```
option_name.KEYWORD1 = value
```

This sets the keyword *KEYWORD1* for the option called *option_name*. This will allow you to refer to this keyword in any profile configuration file when defining that specific option. For example, in the main configuration file you have this line:

```
blast_options.SELENO = -b 5000 -F F
```

which allows the profile configuration files to bear this:

```
blast_options = SELENO
```

This tells the program that it must refer to the keyword *SELENO* for the *blast_options* of this profile, which is translated to the value: *-b 5000 -F F*

This and some other elements in the profile configuration files are **program options**. These can be recognized by their suffix `_options`. These are basically strings which will be concatenated to the command line when the corresponding program is run: `blast` (`psitblastn`), `exonerate`, `genewise` or `tag_blast` (when a [tag_score](#) or [GO_score](#) method is called). *SELENO* is set as the value of all program options when at least a selenocysteine (U) is detected in the alignment. This allows to use specific scoring schemes for these columns.

We have already seen examples of another type of profile configuration element, the **filtering procedures**. These can be recognized by their suffix `_filtering`. All filtering procedures inside selenoprofiles are written in python code and use the variable `x` to indicate the prediction to which the filtering procedure is applied. For advanced filtering, you should see the [advanced usage](#) section to understand and be able to use its syntax. There are three types of filtering: *blast_filtering* (applied to all blast hits to decide which ones will be considered), *p2g_filtering* and *p2g_refiltering* (both applied as a final filter to decide which predictions will be output).

Filters represent the most important non-sequence information layer of a profile. As a rule of thumb, when you use a new profile you may leave the filters as defaults and run selenoprofiles a first time. Then, inspect the results and change them to calibrate your profiles, then rerun selenoprofiles (removing output file and using step option `-F`). You will learn how to create filters suitable to your protein family in subsequent sections.

There are more elements that can appear in a profile configuration file. These will be treated later during this manual as their use is explained: *max_blast_hits_number*, *clustering_seq_id*, *max_column_gaps_for_blast_query*, *tag_db*, *gi2go_db*, *tags*, *go_terms*, *neutral_tags*.

Configuration file vs command line

The configuration file contains all the settings of selenoprofiles, and it can be used for a deep customization of its behavior. In selenoprofiles, all options can be specified in the configuration file or in the command line, with the latter overriding the former default values.

Options in the configuration file have the form

```
option_name = value
```

while in the command line they have the usual form

```
-option_name value2
```

These are the system settings options in the configuration file:

- *temp* = folder

This will be used for the temporary files produced during the workflow. Actually, a subfolder with a random name is used, and deleted at the end of the computation. You should choose a temporary folder with free space at least of the size of the target file.

- *save_chromosomes* = 1 / 0

When active, subfolders are created in the temp folder to unpack the multifasta target files into single fasta files. Only the necessary chromosomes (or contigs) are extracted. Following principles of lazy computation, these files are saved and reused when selenoprofiles is run again on the same target. If you turn this option off, the single fasta files will be instead written in the random name subfolder and deleted at the end.

² To catch option values of multiple words in command line, use double-quotes to delimit them:
`-blast_options " -a 4 "`

- *profile* = profile_name/set_keyword/file

The keyword *profile* in main configuration file denotes the default set of profiles searched, defined as described [here](#). The default value is *eukaryotic*, which is a keyword for all eukaryotic built-in profiles.

- *profiles_folder* = folder

As said, you can provide the profiles list to be searched using directly paths to alignment files, keywords for set of families, or family names. When you use family names, this is the folder where the alignment files named after them are searched for. If you want to use a set of custom profiles, you should create a folder for them and set this option to point it.

The main configuration file is the place where keywords are defined. Keywords can be used for the categories presented in the last chapter, for profile specific parameters and procedures. There's an additional element that use a keyword logic: the set of families.

```
families_set.machinery = sps,sbp2,pstk,secp43,SecS,eEFsec
```

This line in the configuration file allows to use the word *machinery* as a *-profile* option. This will be unpacked into the list of families on runtime. For a very large set of input profiles, we recommend to use option *-fam_list* that overrides *-p* (or *-profile*) option.

Other options found in the configuration file are:

```
three_prime_length=3000
```

This is the length of the sequence cut when the method *three_prime* is called. For selenoprotein families, this is the width of the region downstream the prediction where the SECIS is searched for. The option *five_prime_length* is not present in the default configuration file, but it can be set by the user on runtime or written in the configuration file. This is necessary only if the output five prime is active.

```
blast_opt      = -a 7
exonerate_opt  =
genewise_opt   =
```

The *_opt* program options are concatenated to the command line when using slave programs are run, exactly as *_options* program options in the profile configuration. The difference between them is that the former are always used, while the latter can be set for every profile. In the example, the option *-a* for blast allows to specify the maximum number of CPUs to be used for computation. This will be used for all psitblastn searches.

```
exonerate_extension = 200000
genewise_extension  = 100
genewise_tbs_extension = 10000
```

These parameters are used for when extending the seed alignment provided to the exonerate or genewise routines, described in the [next section](#).

```
species_library = /somepath/names.dmp
GO_obo_file     = /somepath/gene_ontology_ext.obo
```

These two options tell the system where the reference file for the species names and the GO annotation file is located. The first is compulsory present on your system, the second is not.

Some lines in the configuration file start with *ACTION*:

```
ACTION.pre_choose._improve1 = if x.prediction_program()=='blast': x.remove_internal_introns()
```

This defines an action. Actions are operations that are run on every prediction. They may serve different functions. Actions are performed at a certain point during the workflow, defined by their category (in this case *pre_choose*). Some actions are active by default to

improve the predictions and are covered in the [improving prediction](#) chapter of the next section. You will learn more on actions (including how to write them) in a [later chapter](#). There are many more options, some of which will be mentioned later. The full list of options can be obtained by running selenoprofiles with `--help full`

The results folder

The results folder contains all files produced by selenoprofiles. A single folder can store the output data for multiple targets. For each one, a subfolder for target is created concatenating with a dot the species and target names (e.g. *Homo_sapiens.genome*). Think to the results folder as a working environment for a project that include searching multiple profiles in several species, or also in several targets for the same species (for example, genome and transcriptome).

The content of each target folder will vary depending not only on the results of the search, but also on the options specified by the user.

In its most complete form, the target folder will contain the file:

- `results.sqlite` database storing all filtered results on this target

and the folders:

- `output` contains the output files of selenoprofiles
- `blast` contains the psitblastn output files
- `exonerate` contains the exonerate output files
- `genewise` contains the genewise output files
- `prediction_choice` contains the output files for the prediction choice/labelling step
- `filtering` contains the output files for the filtering step
- `tag_blast` contains the output files of the tag blast, if used (see [tag blast](#))

Inside these folders, files are named with a prefix for the profile name. Exonerate and genewise each produce a file for each blast hit satisfying the filtering conditions. Here, the file names are composed adding to the profile name a index linked to a blast hit (example: *fam.1.exonerate*). Additionally, these files are contained in subfolders of the exonerate folder named as each profile, to avoid having too many files in single folders when tons of hits are found by loose profiles. In the output folder, files names contain also the label assigned to each result, followed by the file format (example: *fam.1.selenocysteine.gff*)

Example: files produced searching *SelM* (profile name) in the *genome* (target name) of *Macaca_mulatta* (species name).

```
results_folder/info_target.txt
results_folder/Macaca_mulatta.genome/blast/SelM/SelM.psitblastn.1
results_folder/Macaca_mulatta.genome/exonerate/SelM/SelM.1.exonerate
results_folder/Macaca_mulatta.genome/genewise/SelM/SelM.1.genewise
results_folder/Macaca_mulatta.genome/prediction_choice/SelM.tab
results_folder/Macaca_mulatta.genome/filtering/SelM.tab
results_folder/Macaca_mulatta.genome/output/SelM.ali
results_folder/Macaca_mulatta.genome/output/SelM.1.selenocysteine.p2g
```

If you plan to run selenoprofiles massively, you may want to delete the intermediate files that it produces to avoid an excessive use of disk space. All subfolders listed above can be deleted; as long as results have already been stored in the results database, selenoprofiles will be able to retrieve the desired predictions and produce output files. When run with option `-clean`, selenoprofiles will delete all such subfolders apart from `output/` at the end of the computation.

Output options

As you see in the above example list, an alignment file (`SelM.ali`) is produced as output. This fasta formatted alignment contains the sequences of all results found in this target along with all the profile sequences. This is useful to inspect all results found a certain target, and compare their conservation and spanning respect to the profile. The alignment is computed by mapping each pairwise alignment constituting a prediction (protein-to-genome, or p2g) into the profile alignment. The program `mafft` is used to realign only certain columns of the alignment which deteriorate when adding many predictions in this way.

In the file, the fasta headers of the results start with the “output id” of the prediction (“family.index.label”, for example *SelM.1.selenocysteine*) and contain also other essential information.

As said, the rest of the output files are named after the output id of the prediction plus the format. The available output formats are:

- `p2g` default output format (explained later in the [visualizing results](#) section)
- `fasta` protein sequence
- `gff` genomic coordinates in GFF
- `gtf` genomic coordinates in GTF
- `cds` coding sequence in fasta
- `dna` the full gene sequence, including introns, in fasta
- `three_prime` the sequence downstream of the prediction
- `five_prime` the sequence upstream of the prediction (must specify *-five_prime_length*)
- `introns` the sequence of all introns split in a multi-fasta file

The desired output formats are read from the options in the command line or the configuration file starting with *output_*: for example if option *-output_fasta* is active, the fasta files of all results will be produced, and so on. For all these formats, it is possible alternatively to produce a single file containing all results, by adding *_file* to the option and providing an argument. If for example you want to produce a single GTF with all predictions, use

```
Selenoprofiles [...] -output_gtf_file all_results.gtf
```

In the main configuration file you can see what file formats are produced by default. Out-of-the-box, the only active output options are *output_ali* (for the alignment of results along with the profile) and *output_p2g*. Sometimes, you may also want to use a different output folder: this can be chosen with *-outfolder*.

You can define your own output format by writing a method in python, and add it to selenoprofiles using the *-add* option (see later [option -add](#)).

The results database

At the end of the pipeline, before outputting, results are stored in SQLite database called *results.sqlite*, placed inside the subfolder for this target in the results folder. It is possible to browse through results opening the database files with an SQLite browser, although normally you will not need to. The script *selenoprofiles_database.py* can be used to query or modify the database.

Inspecting results: .p2g format

Selenoprofiles native output format is the following: .p2g

FILE: /results_folder/Gallus_gallus.genome/output/Ahpc_1.4.pseudo.p2g

```
--
Output_id:  AhpC.3.pseudo
-----
-Species      Gallus gallus                      -Taxid 9031
-Target       /users/rg/mmariotti/Genomes/Gallus_gallus/genome.fa
-Chromosome (-) Z
-Program      exonerate
-Query name   Anolis_carolinensis
-Query range  34-226      length:226   coverage: 0.85
-Profile range 58-289      length:303   coverage: 0.77   sec_position: [99]
-ASI:         0.2521      (ignoring gaps: 0.2708)
-AWSIc:       0.4486      Z-score: 1.06
-AWSIw:       0.4561      Z-score: 1.145
-State        kept

----- alignment -----
Query  AAQCPLLDAAAGEKTPFGTLFRDRKAIVVFVR <---Intron---> HFLUYTCKEYVEDLAKIPKKYLE <---Intron---> DANVRLVVIGQSSP
      ||| /||| | / ||| | / ||||| | < 435nt > /|| ||||| ||||| || / ||| < 1167nt > /||| | / |||||
Target  AAYCLVVDADGSRIPFGALYRRQKAIVVFVR      NFLCYTCKEYVEDLAKVPRSYLQ      EANVRLIVIGQSSY
      ggtttgggggaaactggttaccagaggtgc      attttatagtgggcgagcaattc      ggagacagagcttt
      ccagtttacaggtctgtctaggaactttttg      gt      ag      attgacgaaataatcatcggata      gt      ag      acatgttttgacca
      cccgggcgcggtgcccgcgcggggcccgggtgg      tcgtcctggtaacgaaccgttaa      aatggtattagatt
      *

Query  DHIK <---Intron---> PFCHLTGYSHEIYVDPGREIYKILGMKNGETADTPV <---Intron---> QSPHVKSSFLSGHIKSIWRAVFSPAADF
      ||| < 409nt > ||| ||||| /| / ||||| ||||| ||| | | < 197nt > ||||| ||| | /| / ||||| / |||||
Target  HHIK      PFCSLTGYTHEMYVDPQREIYKMLGMKRGEENDVSV      QSPHVKSSMLLGSIRSMWRAMTSPAADF
      ccaa      cttatagtagtgatggccagataacgaaaggaggtg      caccgataactgaaaaatagaaacgtgt
      aata      gt      ag      ctggtcgacaatatagataattgtaggagaatc      gt      ag      tagcatagctttggtggtggtcgcctat
      ttcg      ctctatgtatagtagataagattagtcgaatttcaa      ggcttaaacgcgcttatggaagtcattcc

Query  QGDPTQGGGALILGPG <---Intron---> NQVHFVHLDNRLDHVPINTVLQLA ! FRAME ! GVQTVNFTQRSQIIDV
      |||| |||| ||||| < 553nt > | / ||||| /| / ||||| ||||| ||||| ! SHIFT ! || |||| / |||||
Target  QGDPAQGGGTLILGPG      NEVHFLHHDNRNLDHVPINSVLQLA      1nt      GVNPNVFTNKPQIIDV
      cggcgccggatatgcg      aggcttcggaatgcgcaatgtccg      c      ggacgataaaccaagg
      agaccaaggctttgc      gt      ag      gaattattaaagagtaattctacttatc      c      gtactatcaacattat
      aacttagaatgcaca      ttatttgtttacagtttcttatggga      atcaatcacacgttta

----- positions -----
Exon 1      41768514      41768606
Exon 2      41768010      41768078
Exon 3      41766789      41766842
Exon 4      41766274      41766379
Exon 5      41765945      41766076
Exon 6      41765315      41765391
Exon 7      41765266      41765313

----- features -----
None
----- 3' seq -----
Total sequence length available downstream >= 6000
Sequence until first stop codon:
TGA
*
```


The header of the file contains the basic information about this gene prediction, and is pretty self-explanatory. Some numbers are reported: the ASI is the average of the sequence identities computed comparing the candidate sequence with each one of the profile sequences, and gives an idea of how much it *fits* in the profile. AWSIc and AWSIw are analog similarity scores, detailed later (see [AWSI score](#)). Their linked Z-score is obtained by comparing the score of this candidate sequence with the distribution of scores of the sequences in the profiles, comparing each one to all others. The default refiltering requires the AWSIc Z-score to be greater than -3.

Next in the output file, there is a line indicating the attribute *State*. This is always *kept*, unless the *-state* option ([as explained here](#)) is active.

Then, the query-target pairwise alignment constituting the gene structure prediction is shown. Between the amino acids, bars are used to show the identity | or the similarity / of the aligned residues. Predicted in-frame stop codons (absent in the example) and selenocysteine columns in the input alignment are marked below with X and * respectively. An insertion in the target producing a frameshift is present near the end of the prediction. When analyzing low-quality genomes, frameshifts and stop codons should be not trusted, and checked with sequence data from the same organism by a different source, if available. In this example, the gene structure looks well conserved except for the insertion. The presence of introns and good splice sites also suggest that this is not a pseudogene. Thus, this result should be considered a valid gene despite its label *pseudo*. This is the reason why by default selenoprofiles does not filter out potential pseudogenes. When working with high quality target sequences, one can decide to filter out results with this label, as shown [here](#).

Next in the file, the genomic positions of the exons are reported. The first nucleotide of a chromosome or scaffold is indexed as 1. The frameshift is considered as a short intron, dividing the real exon in two.

In the next section, all features found belonging to this predictions are shown. Features are objects linked to a p2g result, which the user can manipulate to add layers of analysis to the pipeline, and get custom output here in the *.p2g* file (as explained [later](#)).

Finally, the sequence at the three prime of the gene structure prediction is reported, until the first stop codon. In this example a TGA is found right downstream, indicating that the coding sequence prediction is complete at the 3'.

Searching multiple targets

Selenoprofiles is meant to search for one or more protein families of interest in many species and compare results. We suggest to use a certain structure for the file paths in this case. The genome sequences of all investigated species should be in subfolders named after the species, with spaces replaced by underscores. The file name of the genome fasta sequence file (or a link to it) should be *genome.fa*. Example:

```
/home/genome_links/Drosophila_melanogaster/genome.fa
/home/genome_links/Homo_sapiens/genome.fa
/home/genome_links/Mus_musculus/genome.fa
/home/genome_links/Pan_troglodytes/genome.fa
```

When selenoprofiles is run on a target, it will format the sequence database file creating files such as *genome.index*, *genome.lengths* in the same species subfolder. Also, an advantage of this structure is that selenoprofiles will detect the species name from the target path, thus option *-s* is not strictly needed.

After the pipeline has been run, the results of a profile in many targets should be inspected all together. The program *selenoprofiles_join_alignments.py* searches for the *.ali* alignments in the results folder and joins those of the same family into new alignments,

which will contain the results in all targets along with the profile sequences. In the new alignment, the title identifiers corresponding to the predictions look like this:

```
>family.id.label.species_name.target_name
```

They are different from those in the previous *.ali* files, in that they contain the species and the target name as part of the first word, to make each title identifier unique. For more information on *selenoprofiles_join_alignments.py*, run it with option *--help*.

Every prediction consists of a pairwise alignment between a profile protein query and a nucleotide target. The new, joined alignments are produced by mapping all pairwise alignments to the profile. A procedure is used to detect columns that are misaligned by the process (for example when an insertion is present in many targets, but absent from all queries), and mafft is used to realign them.

Such procedure of alignment mapping is used to ensure the consistency of the alignment between the profile sequences, no matter how many predictions are present in the same alignment. Anyway, you may want to realign your results using a more sophisticated tool, such as T-coffee (<http://www.tcoffee.org/>).

The resulting alignment of your results can be inspected using a number of programs (http://en.wikipedia.org/wiki/List_of_alignment_visualization_software).

The joined alignments are also the input to the program *selenoprofiles_tree_drawer.py*, for visualizing the results of (potentially) multiple profiles in (potentially) multiple species with known phylogenetic relationship. The program requires the installation of the ete2 tree python environment (see <http://ete.cgenomics.org/>), and loads a tree of the investigated species in newick or phylip format: round parenthesis such as "(" and ")" are used to group lineages that cluster together. With few species, one can manually write such a file. For example the tree for human, chimp, mouse in simple newick would be:

```
((Homo sapiens,Pan troglodytes),Mus musculus);
```

If we add rat and fruit fly, we have:

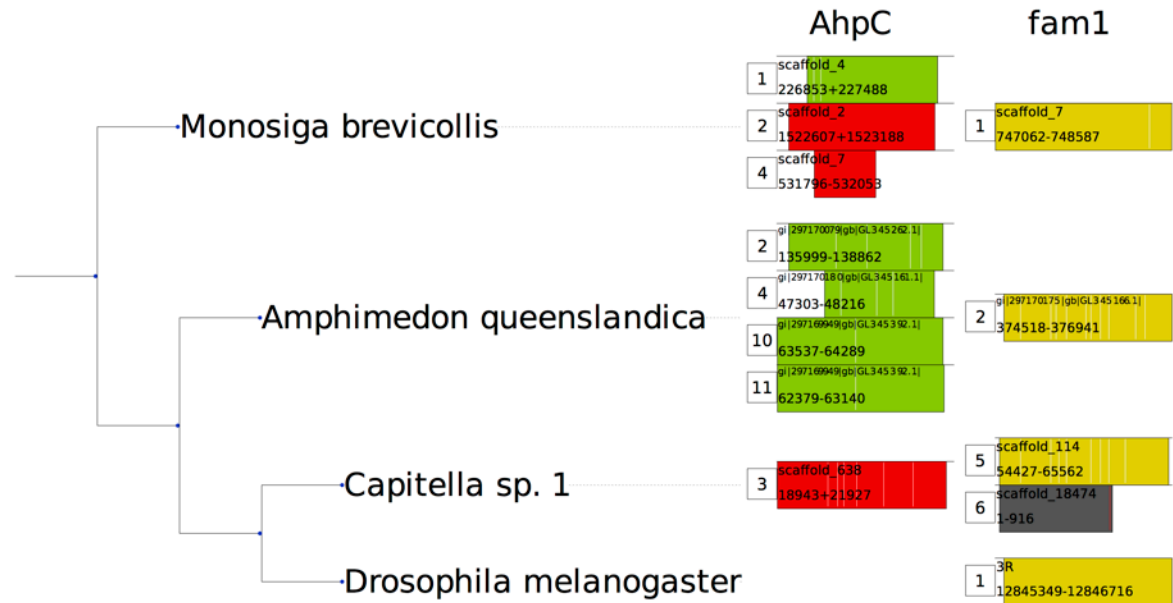
```
((Homo sapiens,Pan troglodytes),(Mus musculus,Rattus norvegicus)), Drosophila melanogaster);
```

For searches on wide range of species, it may be useful to derive their rough tree from the ncbi taxonomy database. This can be done directly at its portal at <http://www.ncbi.nlm.nih.gov/Taxonomy/CommonTree/wwwcmt.cgi>, or with more automated tools such as http://github.com/jhcepas/ncbi_taxonomy.

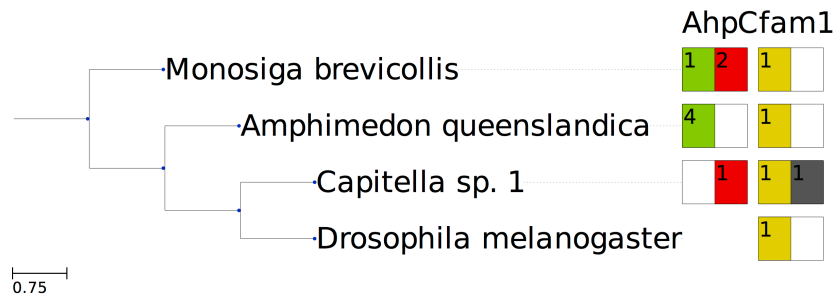
Once you have your joined alignments of results, for example for profiles AhpC and fam1) and a species tree containing (at least) your species of interest, you can run:

```
selenoprofiles_tree_drawer.py AhpC.ali fam1.ali -t species_tree.nw
```


This will open the ete2 graphical environment, showing something like this:



The species tree is indicated on the left. It contains only the species with at least one prediction. The results for different profiles are shown as different columns, on the right. Multiple results for a profile in a species are shown as adjacent rows. Each result is shown as a colored rectangle. A numeric tag at its left indicates its selenoprofiles numeric id. The color depends on its label, with an hard-coded dictionary for selenoprotein families: green for selenoproteins, red for cysteine homologues, (...). For standard, non-selenocysteine containing families (such as fam1 in the example) the only labels are *homologue* (yellow) and *pseudo* (dark grey). The dictionary of colors can be edited by the user directly inside the script *selenoprofiles_tree_drawer.py* (see *label_to_color* declaration). The rectangle width and position indicates the prediction coverage and horizontal span when mapped in the profile alignment. You will find some additional information printed inside each rectangle: the id of the chromosome (or contig), and the genomic coordinate boundaries, separated with “+” for results on the plus strand, and “-” for results on the minus strand. Finally, the intron positions as relative to the protein alignment are shown as vertical white lines. When frameshifts are present, they are shown as vertical red lines. *Selenoprofiles_tree_drawer* can be used to produce images or pdf files summarizing even large sets of results, and has many options for customization (see *selenoprofiles_tree_drawer.py --help*). When a very high number of results have to be visualized, certain options can be used to reduce the amount of information per result shown. The option *-a* in particular allow to compress the number of results by label:



The Selenoprofiles pipeline

Psitblastn

Selenoprofiles uses psitblastn from the ncbi blastall package. This program can be considered an extension of tblastn, which can use not only a single sequence as query, but also a Position Specific Scoring Matrix (PSSM). This allows to utilize the additional information of the relative proportions of the allowed residues at each position. Normally, its more famous relative psiblast (extension of blastp) is used iteratively against a sequence database, building a PSSM with the matches it finds. In our use of psitblastn, no iteration at all is performed, since the profile alignment is already provided as input and the PSSM can readily be derived.

- Pre-clustering

We experienced that when a profile is broad (i.e., contains sequences quite dissimilar to each other), the psitblastn search is not very sensitive. For this reason, selenoprofiles implements a procedure that analyzes the input profile alignment in terms of its variability, and clusters its sequences based on their sequence identity. If the profile has a high variability, then this procedure will produce more than one cluster.

Then, a psitblastn search for each cluster is performed: one PSSM is built from the sequences of each cluster. Consequently, often there are overlapping blast hits coming from the searches of different clusters. Those are merged, keeping only the best one for each overlapping set. The sequence identity threshold can be defined for each profile (*clustering_seqid* parameter), or goes to the default defined in the main configuration file.

- Consensus blast query

Psitblastn build a PSSM along the positions of a certain sequence of the profile, elected as the blast query. In our experience, the choice of the blast query has a big effect on the results of the search. The blast query for each search is not a sequence already present in the profile, but instead a consensus sequence computed on purpose. Its sequence is given by the most present amino acid at each position of the alignment (or of the cluster, if more than one is present). There are two exceptions to this. In the positions where at least a Sec is detected, the blast query always bears a U. The positions featuring a lot of gaps in the alignment are skipped. The maximum percentage of gaps for a column depends on the *max_column_gaps_for_blast_query* option, either specified in the profile configuration or set to the default in the main configuration file.

For technical reasons, all blast hits loaded in selenoprofiles are transformed so that their alignments are between the target and a unique query sequence, named the master blast query. This allows to have a more homogenous kind of data for subsequent computation: otherwise, blast hits coming from different clusters searches would have different sequences as query.

- Merging exons by co-linearity

After the overlapping hits from the various cluster searches are removed, blast hits are once again analyzed, and those likely to be exons of the same gene are joined: they are merged by co-linearity. This means that if a blast hit is downstream of another one, and also the correspondent portions of the aligned query sequences are one downstream of the other in the same direction, the blast hits will be merged into a single object (if they are not too far away). This procedure is done to minimize redundant computation.

- **Blast filtering**

Blast hits are filtered according to criteria that may be specified for each profile. In our experience, different protein families need very distinct criteria. Some families typically match a lot of spurious hits, while some others need loose filters to find all results. All filtering procedures in selenoprofiles are written in python and can be customized by the user, utilizing a set of methods that are already provided or can be created by the user. Filtering is detailed in a later [section](#). Blast filtering is performed actually before removing redundancy across cluster searches, and also before merging by co-linearity. This is because merging blast hits requires loading them all into memory, sorting them and parsing them -- which sometimes would take very long if all blast hits in a output file are considered.

If for some reason you want to inspect manually the blast hits passing the filter, you can use option `-filtered_blast_file` and provide an file as argument, which will be created. The blast hits inside have not been subject to inter-cluster and co-linearity merging.

- **Maximum number of blast hits**

In selenoprofiles, the computation is largely dependent on the number of blast hits passing filtering. For this reason, there is a fixed maximum number of blast hits which can be considered. The default value is very loose: 2500. When the limit is passed for a family, a warning is printed on screen and the workflow follows keeping only the blast hits found so far. Blast hits are read in the order they are in the blast output file. Blast sorts the hits according to the chromosomes (or contigs) they are located on, ordering the chromosomes according to the e-value of the best HSP found on them. This way of sorting is not strictly best-to-worse but it is similar, therefore most likely you won't lose any bona-fide gene because you reached the maximum limit of blast hits.

Also, the blast outputs produced searching the different clusters are read in order, with the cluster containing the highest number of sequences being first. Therefore, the first blast output read should be the most representative.

In an older version of selenoprofiles, the computation would simply stop if the max number of blast hits is reached. This behavior can be restored by setting off the relevant option, with `-blast_filtering_warning 0`.

Exonerate

Each alignment coming from the blast phase is used as a seed to run exonerate in the corresponding genomic region.

- **Reading and joining exonerate predictions**

Given that exonerate is run on a region where a blast hit was found, typically it will give only a prediction in output. Nonetheless, this is not always the case. For this reason selenoprofiles considers only the exonerate prediction which, among those in its output file, overlaps with the blast hit used as seed. If more than one overlapping prediction is present (very rarely), the best scoring is taken.

Also, exonerate generally joins the exons belonging the same gene, including the prediction of splice sites. Nonetheless, often no good scoring splice sites are found and such predictions may be found separated. Thus, selenoprofiles attempts to merge the "main" exonerate prediction with the others in the same file, using the co-linearity concept previously mentioned for blast hits.

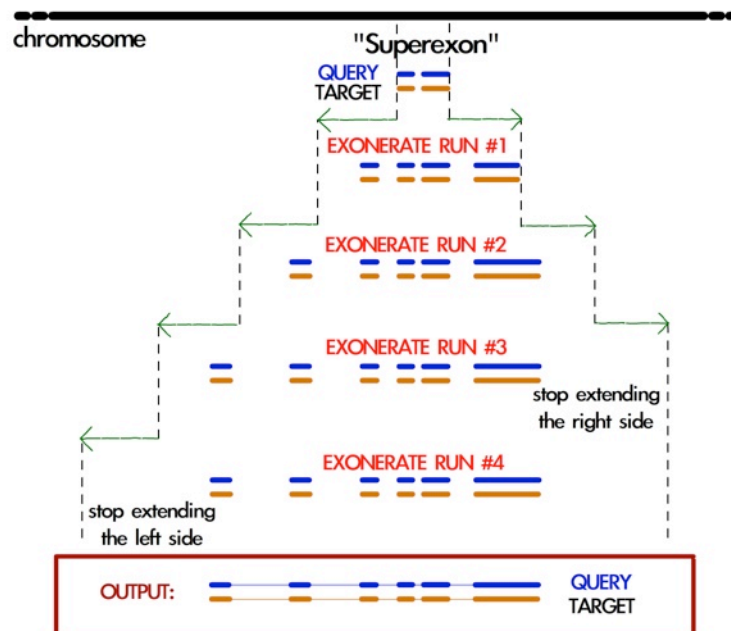
- **Cyclic exonerate**

Exonerate is run through a peculiar routine called cyclic exonerate (see figure below; see also selenoprofiles paper). This procedure comes in response to the following problem: if we want to run exonerate on a certain genomic region where a blast alignment gave us

the hint of an homology match, we need to decide the boundaries of the region searched by exonerate. Of course the region provided by blast needs to be extended, but by how much? Genes sizes are incredibly variable. Taking the biggest size ever observed would result in a huge amount of useless computation, while on the other side taking an average would obviously be inappropriate for a fraction of cases.

This routine solves this problem by running exonerate more than once, increasing progressively the genomic space searched on both sides by a fixed parameter. The cycle stops when a run predicts the same coding sequence of the previous one. If the extension parameter is chosen bigger than the biggest expect intron, the procedure ensures that the widest prediction possible is achieved.

The cyclic routine runs exonerate on average less than 3 times. Given the high speed of exonerate, this is more than acceptable also considering that this step is not the most computationally intensive in selenoprofiles. Also, if the chromosome (contig) is comparable in size to the extension parameter, the cyclic routine is not performed and the whole chromosome is used as target. The default *exonerate_extension* is 200.000 bases.



Schema of the cyclic exonerate routine, from selenoprofiles paper (see [references](#)). A "superexon" represents either a blast hit or more than one merged by co-linearity.

- Choosing the best query from the profile

Exonerate accepts a single sequence query, but in the pipeline the information of a whole profile of sequences is available. Thus, selenoprofiles chooses the best query sequence in the profile for each candidate gene, by searching the query which is most similar to the sequence predicted in the target. To do so, the current predicted sequence is mapped to the profile alignment exploiting the query, which is in common between the prediction alignment and the profile alignment. This is done at every cycle, before running exonerate. At the first run the predicted sequence in the target is given by the blast prediction, and for each subsequent run is given by the previous exonerate prediction. Before closing the cyclic routine, it is checked that the best query is still the one that was lastly chosen, otherwise one more cycle is run.

- Modifying exonerate behavior for selenocysteine sites

Selenoprofiles was created to predict genes belonging to selenoprotein families. It is able to do so by using special scoring schemes with exonerate and genewise (blast is used with a “neutral” schema at these sites).

When dealing with Sec families a particular scoring matrix derived from BLOSUM62 is used, in which the alignment of a “*” character to a stop codon in the target is scored positively. When the query is chosen from the alignment, its sequence is modified before it is used by exonerate: all the positions which contains at least one Sec in the profile are changed to “*”, favoring *de facto* the alignment of Sec positions to UGA codons³.

- Removing redundant exonerate hits

Often, blast hits representing exons of the same genes pass through the co-linearity merge procedure previously described, mainly because this is kept with loose parameters to avoid joining accidentally similar, close genes. When this happens, such blast hits are used to seed cyclic exonerate runs which end up in identical gene structure predictions.

After all exonerate runs are computed, their predictions are analyzed and the redundant ones are dropped, to save computational time in the genewise phase.

Genewise

Generally, genewise represents the most computationally expensive step in selenoprofiles, together with blast. Genewise performs basically the same task of exonerate, which is a tblastn-like alignment including also prediction of splice sites. Nonetheless, this program does not use heuristics and its running time is considerably higher. When you need to maximize speed, you can skip the genewise step using option *-dont_genewise* (the option *-dont_exonerate* is also available, but has to be coupled with *-dont_genewise*).

Genewise is generally run on genomic regions defined by an exonerate prediction, attempting to refine them. Such genomic regions are extended by a parameter, *genewise_extension*, which is only 100 bases by default, and unlike exonerate the program is run only once.

- Genewise “to be sure” routine

In many cases exonerate does not produce any prediction in output. This happens particularly for very low scoring blast hits, which cannot be reproduced by exonerate. In these cases, selenoprofiles performs a genewise routine called “to be sure”, in which a blast hit (instead of an exonerate prediction) is used as seed of a genewise run. In our experience this rescues many predictions, but it is very computationally expensive. The extension of genomic region in the blast hit is defined by the *genewise_tbs_extension* parameter, which is 10.000 bases by default. One can avoid running this routine using option *-genewise_to_be_sure 0*.

- The query in genewise

As for exonerate, a single query sequence needs to be chosen to be run with genewise. In a standard run, the same query used by exonerate is chosen, as this is already the most similar to the target sequence. When a blast hit is used in the genewise “to be sure” routine, the best sequence is chosen from the profile by maximizing identity with the target, in the same way it is done in the first cycle of an exonerate routine.

- Modifying genewise behavior for selenocysteine sites

For genewise, a trick similar to the one described for exonerate is used when searching for selenoprotein families. Each query used is modified to bear a selenocysteine (“U”)

³ The alignment of Sec positions to other stop codons is also favored. This is collateral, as no way was found for exonerate to favor the alignment only to UGA codons. Predictions in which a non-UGA stop codon is present in-frame would be labelled as pseudogenes.

corresponding to every column of the alignment which possesses at least one. Then, the translation table normally used by genewise is changed, using one in which UGA is translated as "U". The scoring matrix given to genewise is then a modified BLOSUM62, in which a "U" in the target is score positively only to a "U" in the query.

Improving predictions

In selenoprofiles a few steps are dedicated to the processing of the predicted gene structures, in order to correct them. All of them are implemented as methods of the superclass *p2ghit*, which comprises the classes for blast, exonerate or genewise predictions (see later [p2ghit class](#)). These methods are run through actions (see later [actions](#)) specified in the main configuration file. You can turn off the improvements methods by removing or commenting (with #) the corresponding lines in the main configuration file.

The first improvement is called *remove_internal_introns* and is performed only on blast hits. This method is useful because often blast joins in a single HSP two or more exons, when the exons are on the same frame and the resulting stretch of unaligned amino acids in the target is acceptable in terms of scoring. A typical blast hit containing an evident intron is shown here:

```
Score = 100 bits (249), Expect = 4e-20
Identities = 49/93 (52%), Positives = 59/93 (63%), Gaps = 26/93 (27%)
Frame = +2

Query: 12      LEPYMDENFITRAFAKMGENVSVKLIRNKMTG-----E 45
              LEPYMDENFI+RAFA MGE  +SVK+IRN++TG
Sbjct: 103916 LEPYMDENFISRAFATMGELVLSVKIIRNRLTGYV*SLFVFYHIPNFGVHLHTLFSLSRI 104095

Query: 46      PAGYCFVEFADEASAERAMHKLNGKPIPGANPP 78
              PAGYCFVEFAD A+AE+ +HK+NGKP+PGA P
Sbjct: 104096 PAGYCFVEFADLATAEKCLHKINGKPLPGATPV 104194
```

The portion YV*SLFVFYHIPNFGVHLHTLFSLSRI is the translation of an intron. It has no correspondence in the query, and it also contains a stop codon (it is normal as introns have no coding constraint). The *remove_internal_introns* method detects these cases by searching the sequence in the target for stretches of at least 18 bp (6 amino acids) not aligned to the query, and removes them from the prediction.

The second improvement is performed by function *clean_inframe_stop_codons*. This is applied to predictions by all programs, and comes from the observation that often these programs include stop codons that should be avoided. This would cause these predictions to be mislabelled as pseudogenes. This method is simple in principle: it checks for the presence of stop codons close to exon boundaries (default maximum: 10 codons). If it finds any, it removes the stop codons and also the portion which links it to the closest exon boundary.

The third improvement is *exclude_large_introns*. This is particularly useful on exonerate predictions, which sometimes possess extremely large introns, due only to spurious similarity with far away regions, and to the presence of decent splice sites just by random. This function detects each such large intron (default ≥ 140000 nt), and removes all exons (typically just one) at one side of that intron, the side with the smallest coding sequence.

While all described methods are applied before prediction choice, the fourth and fifth improvements are performed after filtering, and only on predictions passing the filter.

The functions *complete_at_five_prime* and *complete_at_three_prime* are attempts to complete the coding sequence predictions looking for an upstream ATG and a downstream stop codons. Let's see the corresponding lines in the *selenoprofiles.config* file (expanded for readability):

```

ACTION.post_filtering._improve4=
\\ if x.filtered=='kept':
\\     x.complete_at_three_prime(max_extension=10, max_query_unaligned=30)

ACTION.post_filtering._improve5=
\\ if x.filtered=='kept':
\\     x.complete_at_five_prime(max_extension=15, max_query_unaligned=30, full=False)

```

The completion at 5' is performed only if a ATG is found before a stop codon, and if at most 15 codons would be added. Also, two other conditions must be met: no non-standard characters must be found in the 5' extension, and the profile query of this prediction must have an unaligned portion at N-terminal not bigger than 30 amino acids. This is to avoid completing partial hits, whose upstream ATG are not likely to be the real starts, as other large portion of coding sequence are expected upstream.

Also, normally the function stops when the first methionine is found upstream -- if the first codon is already a AUG, no extension is performed. When *full=True* is provided, it attempts instead to extend to the furthest possible methionine, when coupled with high values of *max_extension*.

The completion at the 3' is performed only if the profile query has an unaligned portion at C-terminal not bigger than 30 amino acids, if the extension is at most 10 codons, and if no strange characters are found in the candidate extension.

The behavior of these functions can be easily altered by the user with the main configuration file. When searching bacteria in particular, one may want to increment the extension parameters, as the absence of introns makes extensions more reliable.

Selenoprofiles can be customized to perform additional improvements. The user has to write a function accepting a *p2ghit* as input, and modify the main configuration file to run the function at the right step, using actions.

Prediction program choice

After the genewise step, three predictions are available for every candidate: one by blast, one by exonerate, and one by genewise. The predictions are analyzed and only one is taken to represent this candidate gene to the filtering phase, and possibly to output. The function *choose_prediction* is used to decide among any number of candidates. This same function is used during all steps in which genes are merged to remove redundancy, to decide which one to keep. The following conditions are checked in order: if at any point only one of the predictions shows to be better than all others for a criteria, the function stops and that prediction is returned.

The first condition checked is the presence of frameshifts. If a prediction possesses frameshifts while another doesn't, the latter is taken⁴.

Then, if the predictions come from a selenoprotein family, the number of aligned Sec positions is considered: if one possesses more than the others, it is chosen.

The number of in-frame stop codons (others than SecTGAs) is then checked: if one possesses less than the others (for example one has none, while the others have), it is chosen.

After, the length of the predicted coding sequence is determinant: the prediction featuring the longest sequence is chosen.

⁴ Nonetheless, blast predictions are automatically discarded if any other prediction contains frameshifts. This is necessary because blast does not predict frameshifts. Thus, when a real pseudogene with frameshifts is analyzed, the prediction choice routine would inevitably go to the blast prediction since the others have frameshifts and blast does not.

If at this point the choice has not been made yet, the prediction whose program has highest priority is chosen, given these priorities in descending order: genewise, exonerate, blast.

Option *-no_blast* forces Selenoprofiles to choose the exonerate or genewise prediction. This is useful only if an accurate splice sites prediction is important for you. It comes at the cost that, when only the blast prediction is available (for example because exonerate produced an empty output, and genewise an invalid alignment), the candidate is always discarded.

Labeling

After a single prediction per candidate is chosen, this is analyzed and labelled.

For standard families, there are only two possible labels: *homologue* (a regular prediction) and *pseudo* (with any in-frame stop codon or frameshift). It is possible for the user to define its own labeling procedure: this is shortly described in the [option -add chapter](#).

For selenoprotein families, labeling is used to characterize the amino acid aligned to the Sec position. Generally there's a single Sec in selenoproteins. If there's more than one, the label assigned by selenoprofiles depends on the most-left aligned Sec position. The possible labels are *selenocysteine*, *cysteine* or any other amino acid (only rarely found at these positions though). If the prediction does not span any Sec position, it is labelled as *unaligned*. If it contains frameshifts or in-frame stop codons (apart from Sec-TGA), then it is labeled as *pseudo*. An additional label, *uga_containing*, is assigned to those predictions whose only pseudogene feature is one or more in frame UGAs (of course not aligned to Sec positions). This label is useful because very rarely the scoring schemes used for selenoprotein families allow the alignment over a non-Sec UGA, and we don't want to filter those out as if it were pseudos. Also, the label may be useful to discover new Sec positions in known selenoprotein families.

Final filtering

After labeling, predictions are evaluated through the final filter before output. This filter, exactly as the blast filter, can be specific for each family and be written using the methods provided in selenoprofiles classes. The filter outcome is summed up in a filtering label, hereafter called "filtering state" (or just state) to differentiate it from the label assigned in the previous step. The final filter actually consists of two separate filters, called *p2g_filtering* and *p2g_refiltering* in the configuration files. A prediction excluded by the first one will be assigned a state of *filtered*. A prediction excluded by the second one will be assigned a state of *refiltered*.

Just before the predictions enter the final filter, there is an additional redundancy check: the predictions overlapping each other are compared and only the best one is kept. Predictions discarded this way are assigned a state of *redundant*.

Those predictions which passed all the redundancy check and the two steps of the final filter without being discarded are assigned a state of *kept* and represent the normal output of selenoprofiles.

Nonetheless, the user may decide to output the predictions with a different state, using the *-state* option, optionally with multiple arguments, comma separated with no space within. If for example you want to output all *filtered* and *refiltered* predicted, add to your command line:

```
-state filtered,refiltered
```

The *-state* option can accept the following arguments: *kept*, *filtered*, *refiltered*, *redundant* or *overlapping* (see below). There is a way to have even more control on what prediction are

output: the `-output_filter` option. This accepts a procedure with the same syntax of filters and actions, which is evaluated for every prediction: those for which this evaluates to *True* will be output. If for example you want to output only predictions on the positive strand, you can use:

```
-output_filter "x.strand=='+'
```

To do this, you need to know a bit about the classes used in selenoprofiles, described in the [advanced usage](#) section. After filtering, results are stored in the sqlite database, ready for the [output phase](#).

Removing inter-family redundancy

Selenoprofiles scans for multiple profiles in a single run. The output is produced only when all families have been searched. This is because results from different profiles may overlap, especially when some of them share a certain degree of sequence similarity. So after all results are stored in the database, this is parsed and every prediction is compared with all others on the same chromosome (or contig). When two such predictions overlap, the function *choose_among_overlapping_p2gs_interfamily* is used to decide which one to keep. The other is assigned a state of *overlapping*. These predictions will not be output by default. Note that this operation is performed directly on the database: the intermediate text files written in the filtering phase will display the state previously assigned.

Another important note: the inter-family redundancy check is performed every time an output phase is run, and depends on the results present in the database at that moment. For this reason, searching several profiles in distinct selenoprofiles runs will lead to more (or the same number of) output files than searching all of them in a single run. The results database at the end will be identical, but as when every profile reached its output phase, the predictions of all other profiles were not available, the inter-family redundancy cannot be checked properly.

If you searched different profiles on separates runs, the best thing to do is just delete all output files and rerun selenoprofiles with all these profiles using `-D` flag to re-run database storage. No heavy computation will be repeated, and only the output files for the non-overlapping predictions will be produced.

Running selenoprofiles in parallel

Selenoprofiles can be easily parallelized to be run on a large number of targets. Since the computation is independent for each target, such selenoprofiles jobs (optionally scanning for multiple profiles) can be freely split and submitted to different nodes of a computer cluster. But selenoprofiles allows also to split the computation on a single target, which is necessary if you are using it to completely annotate a genome with a comprehensive collection of protein profiles. In this case, the potential overlap of results by different profiles is a hurdle to parallelization. Thus, the strategy is not to proceed to output until results from all profiles are available. This can be accomplished by option `-stop`. With this option, the program will stop after having filtered and stored the results in the sqlite database. So, you can parallelize the search for each profile, using `-stop` in each such command line. Following the first example shown in this manual:

```
Selenoprofiles results_folder -t /db/genome.fasta -s "Macaca_mulatta" -p family1 -stop
Selenoprofiles results_folder -t /db/genome.fasta -s "Macaca_mulatta" -p family2 -stop
Selenoprofiles results_folder -t /db/genome.fasta -s "Macaca_mulatta" -p family3 -stop
Selenoprofiles results_folder -t /db/genome.fasta -s "Macaca_mulatta" -p family4 -stop
...
Selenoprofiles results_folder -t /db/genome.fasta -s "Macaca_mulatta" -p familyN -stop
```

Each of the commands above can be sent to a different node in a computer cluster. When all of them are finished, you can then run:

```
Selenoprofiles results_folder -t /db/genome.fasta -s "Macaca_mulatta" -p fam_all -merge
```

Assuming that the keyword *fam_all* is defined in the main configuration file as the list of all profiles, this will make selenoprofiles load all results previously computed from the database, remove inter-family overlaps, and proceed to output for all profiles.

This strategy works only if all selenoprofiles instances in the parallelized phase work until completion. If for any reason any job crashes, this may leave the sqlite database in a state that compromises the other jobs as well. If you experience database errors, you may need to cleanse the *results.sqlite* file using script *selenoprofiles_database.py*, and rerun. In the worst case, you can delete the sqlite file. As all intermediates files by slave programs are kept (unless you activated option *-clean*), the great majority of computation is never repeated anyway.

Option *-no_db* provides a more robust alternative to *-stop*. When *-no_db* is active, the sqlite database is not used at all by selenoprofiles, and execution is stopped after the final filtering step. Therefore, you can parallelize the jobs as before:

```
Selenoprofiles results_folder -t /db/genome.fasta -s "Macaca_mulatta" -p family1 -no_db
Selenoprofiles results_folder -t /db/genome.fasta -s "Macaca_mulatta" -p family2 -no_db
...
Selenoprofiles results_folder -t /db/genome.fasta -s "Macaca_mulatta" -p familyN -no_db
```

and finally compute overlaps and output with this:

```
Selenoprofiles results_folder -t /db/genome.fasta -s "Macaca_mulatta" -p fam_all -merge
```

In this case, the computational time required for the last run is significantly increased, since all intermediate files need to be parsed again, and all actions have to be rerun to populate the database. Normally though, this is acceptable time-wise.

Advanced usage

Selenoprofiles was designed to be as customizable as possible. It offers to the user the possibility of writing python code which will be integrated and run. The code can be provided mainly through the configuration file of each profile, and through the main selenoprofiles configuration file. Additionally, custom modules can be loaded using option *-add*, as we will see later.

In the simplest use of custom code, the user can set profile specific procedures, exploiting the built-in methods for filtering:

```
### fam1.fa.config
blast_filtering = x.evalue < 1e-15
p2g_filtering = x.aws_i_filter (aws_i=0.3)
p2g_refiltering = x.coverage() > 0.5
```

With more experience, it is possible to add custom information to output, or even annotate motifs or secondary structures in the predictions:

```
### selenoprofiles config
(...)
ACTION.pre_output.see_cys= write(x.output_id()+ " Cys:" +( join([str(i) for i, aa in
enumerate(x.protein()) if aa== "C"] or "None" ), 1)
```

```
### output
fam1,1.homologue Cys:14,17,64,189,192
fam1,5.homologue Cys:18,21,194,197
fam1,11.pseudo Cys:60,63
fam1,19.pseudo Cys:None
```

The *p2ghit* class

To learn how to use custom code, you need to be familiar with some variables and classes in selenoprofiles, as these are the objects that your code will be manipulating. To do this, you should have already some experience with python code and classes. The *p2ghit* class is the key of user customization. It represents a prediction of selenoprofiles, coming from any source among blast, exonerate or genewise. It contains the alignment of a query against a target, and the genomic coordinates of such alignments. Let's see its mostly used attributes and methods (for a full list, read script *selenoprofiles.py* at *class p2ghit*):

***p2ghit* class**

Attribute or method	Description
id	The numeric id of the prediction (string). It is unique for that family and target
chromosome	The first word of the fasta header of the chromosome or scaffold where this prediction resides
strand	The strand of the prediction (+ or -)
label	The label assigned to this prediction in the labeling phase
filtered	The filtered state assigned by the filtering phase (<code>kept</code> , <code>filtered</code> , <code>refiltered</code> , <code>redundant</code>). After inter-family overlaps are computed, the state <code>overlapping</code> is also possible
output_id()	The prediction name displayed in output (profile name.index id.label). Example: <code>Se1K.1.pseudo</code>
prediction_program()	The program that generated this prediction (<code>blast</code> , <code>exonerate</code> or <code>genewise</code>)
query_full_name()	The full name of the query, as it appears in the profile alignment
coverage()	A float value, indicating how much profile is spanned by the prediction (max is 1.0)
protein()	Protein sequence, with * for stop codons, U for Sec
cds()	Nucleotide coding sequence, as ATGC characters
positions_summary()	A string with the positions of all exons. Examples: 24-40,70-100 (+ strand) 400-450,340-354 (- strand)
exons	A list (array) containing the exons. Each exon is a list of 2 elements (integers), the position of start and the position of end of the coding sequence, both 1-based and included. Each prediction has at least one exon.
header()	A string used as default fasta header. Contains lots of non-sequence information. Example: SBP2.1.homologue chromosome:scaffold1 strand:+ positions:869-881,1163-1417 species:Polysphondylium_pallidum_PN500 target:genomes/P.pallidum/genome.fa prediction_program:exonerate
dna()	Full nucleotide gene sequence, including introns and frameshifts if present.
splice_site_sequences()	A list of 4 letter strings, with the first two and last two nucleotides of each intron in the prediction.
subsequence(self, start, length)	Generic function to return any nucleotide subsequence of a prediction, using lazy computing. It can be used with negative start or large length to get the sequence around the genomic interval. Normally the indexes are relative to the predicted coding sequence, but you can use <code>include_introns=True</code> to count any nucleotide in the gene prediction.
alignment	Pairwise protein alignment between a profile query and the target, as an instance of the <i>alignment</i> class in <i>MMlib</i> (if interested, check its code in the installation directory).

There are plenty more of methods. Many are actually inherited from the *p2ghit* parent class, called *gene*, defined in the library *MMlib.py*.

Custom output: option *-fasta_add*

The *-fasta_add* option represents an elegant and fast way to add information to output. A python written procedure with the same style of actions and filters must be provided as argument. The procedure is evaluated to a string which is inserted in the fasta headers of the files in output. All the fasta files in output will contain the add-on, as they all call the same function to determine the fasta header. Files with extension *fasta*, *cds*, *dna*, *three_prime*, *five_prime* and also *ali* will have it. Let's see an example. Normally the fasta headers contain the following information:

```
>GPx.6.selenocysteine chromosome:chr3 strand:- positions:
49395460-49395711,49394824-49395180 species:"Homo sapiens" target:/Genomes/
Homo_sapiens/genome.fa prediction_program:genewise
```

Let's say that you want to add the length of the protein to the header. You could add this to your command line:

```
-fasta_add '"seq_length:" +str( len(x.protein()) )'
```

Now if you run selenoprofiles with this (forcing the replacement of the old output with *-O* or specifying another output folder), you will have:

```
>GPx.6.selenocysteine chromosome:chr3 strand:- positions:
49395460-49395711,49394824-49395180 species:"Homo sapiens" target:/Genomes/
Homo_sapiens/genome.fa prediction_program:genewise seq_length:203
```

Actions

The actions are performed during the workflow on each prediction coming from the prediction choice/labelling step. The action is provided as python code that is directly executed in the selenoprofiles environment. In a classical *for* loop, the variable *x* in the code is replaced by each *p2ghit* instance and executed. The keyword *ACTION* in the main configuration file denotes the active actions. Actions can be specified also in the command line. From now on, we will display the examples with the configuration file syntax:

```
ACTION.pre_filtering.echo = print 'hello world', x.id, x.label
```

Separating the left side with dots, the first field is the keyword *ACTION*, the second field is the category of the action and the third is the name of the action. The category determines the time point of the actions, while the name is used only to order the actions in the same category. In this example, the user will just see something like this appearing in the output of selenoprofiles:

```
...
CH00SE: choosing among available predictions, assigning label --> selenoprofiles_results/
Polysphondylium_pallidum_PN500.genome/prediction_choice/SelI.tab (just loading file)
SelI.1      : exonerate  longest CDS predicted      unaligned
SelI.3      : blast      longest CDS predicted      unaligned
SelI.4      : exonerate  longest CDS predicted      unaligned
SelI.7      : blast      SectGA aligned             pseudo
hello world 1 unaligned
hello world 3 unaligned
hello world 4 unaligned
hello world 7 pseudo
...
```

Each action is performed on all available prediction at a certain step of the pipeline, determined by his category. There are many possible categories of actions:

post_blast_filter, *post_blast*, *post_blast_merge*, *pre_choose*, *pre_filtering*, *post_filtering*, *pre_output*.

The categories names are pretty self-explanatory, but see [Appendix 2](#) for their precise temporal mapping. The actions *post_blast* and *post_blast_merge* are performed on blast hits, while the others are performed on blast hits or exonerate/genewise predictions.

You will have to choose the category of your actions depending on what operation you want to perform. Actions executed during *pre_filtering* can be used to improve the predictions, but remember that their attribute *.filtered* is not set yet. *post_filtering* actions can access the *.filtered* attribute and are performed before storing results on the database. *pre_output* actions can add useful information to the log output.

Let's see an example which uses an if statement to execute operations only on a certain subset of the available predictions. Typically, the attributes that you want to check are the *.label* and the *.filtered* attributes. Let's say for example that we want to check the chromosomes and strands where the prediction with label "unaligned" rely:

```
ACTION.post_filtering.test = "if x.label=='unaligned': print x.output_id(), ' CHROMOSOME', x.chromosome, x.strand "
```

This adds something like this in the standard output of selenoprofiles:

```
...
SelI.1.unaligned CHROMOSOME gi|284795330|gb|GL290990.1| +
SelI.3.unaligned CHROMOSOME gi|284795323|gb|GL290997.1| +
SelI.4.unaligned CHROMOSOME gi|284795338|gb|GL290984.1| -
...
```

The next action is for giving a quick look to the protein sequence of all discarded predictions. Below is the output added.

```
ACTION.post_filtering.check_aln = "if x.filtered != 'kept': print x.output_id(), x.protein()"
```

```
...
SelI.4.unaligned ITLVGLFCNIAMYLIYVFQCPGLTEPAPRWCFILIAFLIFAYQTLDNLDGKQARRTKSSSPLGELFDHCCDA
SelI.7.pseudo VTATGFVCNFIALFLMSSYMRPVNDGQEPV
...
```

After the *post_filtering* actions are performed, the results are stored in the selenoprofiles database. Remember that if selenoprofiles finds the results in the database, it does not perform the steps up to filtering. Therefore beware that if you specify actions of category pre or post filtering (or any of the categories before them) on a second run of selenoprofiles, it won't perform them unless you force the proper routine, for example with option -F to force the filtering routine. *pre_output* actions, on the contrary, are performed both if in the current run results are produced or loaded from the database, but only on the results which are output (determined by the *-state* option).

Later, we will see how actions can be used to correct gene structures, or to add custom genomic features to the predictions.

Blast filtering

There are 3 layers of filtering in selenoprofiles, all regulated by procedures defined in the profile. We have already seen them: blast filtering, p2g filtering and refiltering. The same grammar applies to all of them. For blast filtering, the most common attribute checked is the *eval*, an attribute specific of blast hits. The blast hit is a subclass of *p2ghit* and has

the same methods. Let’s see a simple blast filtering procedure as written in a profile configuration file; this accepts only the blast hits with *evaluate* minor (better) than 1e-5:

```
blast_filtering = x.evalue < 1e-5
```

Selenoprofiles offers also more sophisticated tools, which map the prediction back to profile alignment to use what we know from the profile alignment. For example many families possess N-terminal regions of disordered or repetitive sequence, which hits spuriously many regions in the genome. The resulting blast hits span only the initial portion of the profile.

You may want to exclude those, using function *is_contained_in_profile_range*:

```
blast_filtering = x.evalue < 1e-5 and not x.is_contained_in_profile_range(1, 35)
```

The similar function *spans_profile_range* asks whether the predictions spans certain columns of the alignment, useful when you want only proteins with a certain conserved domain.

```
blast_filtering = x.evalue < 1e-5 and x.spans_profile_range(50, 60)
```

The function *show_conservation_in_profile_range* is useful when dealing with blast filtering of profiles with regions of low information. It checks the number of pairwise similarities (defined as positive scores in the BLOSUM62 matrix) between the amino acids in the query and in the target in the prediction along a certain profile range. In the example below, predictions are required to have 3 conserved amino acids in the region from positions 1 to 50.

```
blast_filtering = x.show_conservation_in_profile_range(1, 50, 3)
```

AWSI Z-score based filtering

We developed various method to score how much a sequence “fits” in a protein profile. We called the best performing one Average Weighted Sequence Identity (AWSI).

It is based on the Weighted Sequence Identity (WSI), a scoring method for comparison of two sequences, with one of the two belonging to a profile alignment.

The WSI score is computed as an average of sequence identities with different weights on the different columns of the profiles. In the pairwise comparison between the profile sequence and the candidate sequence, the weight is given by the representation of the amino acid in this profile sequence and column across all the profile. More conserved columns are given more weight thus more importance. This weight is also multiplied by the column coverage, that is to say, the total number of characters which are not gaps divided by the total number of profile sequences. In this way, the alignment regions present only in a small subset of sequences have less importance.

When the term AWSI is used in this manual, we refer to the variant AWSI_c, computed as just explained. There is another variant (AWSI_w), which is computed in the same way, but the weight is not multiplied by the column coverage.

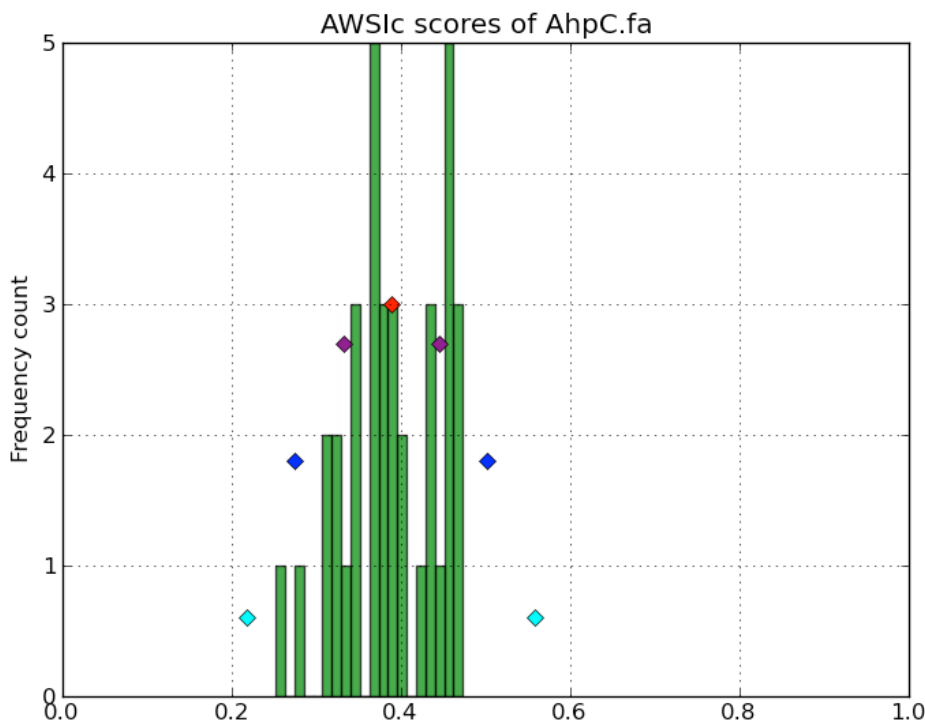
When comparing a candidate sequence against a profile, a WSI for each profile sequence is calculated. Each one ranges from 0 to 1, as it is normalized to the sum of weights in that WSI. Now the AWSI of the candidate sequence is just the average of all computed WSI.

Although the range of AWSI is also between 0 and 1, the maximum value it can assume is constrained by the profile characteristics. In a profile with very dissimilar sequences, no candidate sequence can reach high scores (as if it matches a sequence of the profile, it cannot match the different ones at the same time). Thus, it is useful to adjust the AWSI threshold for each profile.

For this purpose, profile alignments are analyzed when used for the first time, and AWSI values for all sequences are computed. For each profile sequence, we compute its AWSI as explained above, considering this sequence as a candidate, and the rest of sequences as the profile to compare against.

The distribution of these AWSI scores is used to decide the similarity threshold when fitting a sequence into this particular profile. The AWSI score of the target sequence is fit in a Gaussian distribution with the profile average and standard deviation, and a Z-score is computed. In the default p2g refiltering procedure (*aws_i_filter*), the Z-score must be greater than -3.

The script *selenoprofiles_build_profile.py* can be used to display the distribution of the AWSI scores with option *-d*, as shown here above (pylab must be installed). The frequencies of the computed AWSI values are shown as green columns, while colored dots are used to display the approximated gaussian distribution: the red dot is the average, while the purple, blue and cyan dots correspond to the average minus 1, 2, 3 standard deviation respectively. The default cut-off point is thus indicated by the leftmost cyan dot.



The methods of the *p2ghit* class relevant to AWSI scores are:

- *aws_i()* with no arguments, it returns the AWSIc value for this candidate. Used as *aws_i(with_coverage=False)*, returns AWSIw instead
- *aws_i_z_score()* returns the z-score compute comparing the AWSI of this candidate with the profile distribution. This function also accepts the *with_coverage=False* switch to return AWSIw instead.
- *aws_i_filter()* returns *True* if the prediction would pass the default AWSI-based filtering, *False* otherwise. This function also accepts the *with_coverage=False* switch to return AWSIw instead. This is normally computed just as *aws_i_z_score()>-3*, with two possible exceptions. For extremely conserved profiles, the cut-off threshold would be generally too strict. Thus, if the candidate has an extremely high AWSI (>0.9), it is accepted

regardless of the profile characteristics. The second exception is for profiles with few sequences (<3). In these case, the computed AWSI standard deviation is always zero or extremely close to it, and this would also result in filtering too strict. Thus, for these profiles the filter just checks that *aws_i()* ≥ 0.3

One can easily alter the filter behavior using any of these arguments to the *aws_i_filter* function: *z_score*, *aws_i*, *few_sequences_aws_i*. For example *aws_i_filter(aws_i=0.5)* accepts any candidate scoring a AWSI with the profile of 0.5 or greater (or a *z_score* >3).

Other filtering functions

Here's some other methods useful for blast or p2g filtering of specific families.

The function *seq_in_profile_pos* provides the amino acid predicted in the target at a certain position of the profile alignment (may be - for unaligned). It can be used to check that certain domains are complete (e.g. redox boxes CXXC).

```
p2g_refiltering = x.seq_in_profile_pos(31) == 'C' and x.seq_in_profile_pos(34) == 'C'
```

The function *sequence_identity_with_profile* computes a quantitative measure of how much the prediction fits in the profile: it computes the sequence identity of the prediction with every profile sequence, and average them. It is a simplification of the AWSI score. With no arguments, internal (but not terminal) gaps are counted as mismatches. The choice of the threshold in this case depends largely on the profile.

```
p2g_refiltering = x.label != 'pseudo' and x.sequence_identity_with_profile() >= 0.25
```

The more useful function *sequence_identity_in_range* is analogous the previous one, but computes the average sequence identity only on a certain range of the profile. Predictions not spanning this region are given 0.

```
p2g_refiltering = x.label != 'pseudo' and x.sequence_identity_in_range(40, 80) >= 0.35
```

For a full list of the methods of the *p2ghit* class, run *selenoprofiles_3.py* with *-help full* or inspect the script inside your installation directory.

Tag blast filtering

Tag blast is an implemented form of filtering. This consists in searching the protein sequence predicted in the target against a comprehensive protein database (typically nr - non redundant proteins at ncbi). The output generally provides a good annotation of the protein in question. Note that your profile may match sequences in the genome that are real genes, but do not belong to your family and are hit because of their sequence similarity. These predictions usually have blast hits against proteins in nr which are not in your protein family. Tag blast utilizes a set of profile-defined tags to scan the titles in the blast output and assign a score to the prediction. A predicted sequence that resembles proteins not belonging to the family are likely to be spurious, and will be assigned a negative tag score. To use tag blast, you must first set the list of tags for your profile in its configuration file. Tags are strings which are interpreted as perl regular expressions. In the configuration file of the profile, the tags are written as a python list of strings:

```
tags = ['SecS ', '(Sec|selenocysteine|tRNA).* selenium transferase']
```

Tags should be carefully designed in order to recognize all sequences of the profile and those with similar names. For each blast hit appearing in the blast file, the tags are tested and a score is assigned to the title. Its absolute value is the negative logarithm of the

evaluate: a blast hit with *evaluate* 1e-5 gets 5 points. The final tag score assigned to prediction is the sum of all the titles. If the title matches any profile tag, its score will be positive. If it matches any neutral tag, its score will be zero. If a title does not match any profile or neutral tag, its score will be negative. The neutral tags are used to skip all the blast hits with uninformative titles and those based only on computational prediction. The neutral tags are defined in your main configuration file, with a decent default value. For filtering, we check whether the final tag score assigned to predictions is positive:

```
p2g_refiltering = x.label!='pseudo' and x.tag_score() > 0
```

If you want to use the tag score in a filter, we suggest you to inspect manually the results and check their tag score first. For example with this action (paste it in the main configuration file):

```
ACTION.post_filtering.check_score = print "Tag score of", x.output_id()+" filtered: "+x.filtered+"\n"+str(x.tag_score(verbose=1))5
```

The verbose mode will allow you to check the titles of all proteins present in the blast output and the score assigned to them. This will allow you to build and improve useful tags for your family.

When the method *tag_score* is run for the first time on a *p2ghit*, blastp is run against the database defined in the profile or in the main configuration file (under the keyword *tag_db*). The output file is kept in the *tag_blast* subfolder inside the folder dedicated to this target. A tag blast run takes a few minutes, so take care of avoid doing it on a lot of hits. If you put the *tag_score* evaluation on the right side of an *and* construct, the tag blast will not be performed unless all conditions to his left are true:

```
p2g_refiltering = x.coverage()>0.4 and x.label!='pseudo' and x.tag_score()>0
```

GO score filtering

Similarly to the tag score, the GO score utilizes the same blast search against nr, but in this case it is the GO terms associated to the proteins found which are evaluated. A list of the positive GO terms is to be provided in the profile configuration file:

```
go_terms = ["GO:08028", "GO:08030"]
```

A score is assigned to each blast hit depending on the *evaluate*, as in the tag score. The GO terms are searched considering their hierarchy: if for a certain title in the blast output, a GO term is found which is a child of a GO term defined in the profile configuration, this will count as positive. Blast hit with no annotated GO are scored neutral. Only molecular functions GO terms are checked.

```
p2g_refiltering = x.label!='pseudo' and x.go_score()>0
```

Integrate your own code: option *-add*

With the *-add* option, you can provide a python add-on file that will be loaded in selenoprofiles. This will allow you to define functions can then be used in any procedure, for example for filtering or output. The code inside the file provided is read line by line and executed in selenoprofiles when all variables are already loaded and everything is ready to run.

⁵ the `str()` function is necessary to convert the integer returned by `tag_score` into a string that can be concatenated and printed

The label is then typically used for filtering:

```
p2g_refiltering = x.label.startswith("long")
```

There are a few global functions in selenoprofiles that user may be interested in altering. In various steps of the workflow, the program must decide which gene structure prediction is best among 2 or more candidates. The first such function is named *choose_prediction*. This is used in the prediction choice step, when a single prediction among blast, exonerate and genewise is chosen. It accepts a list of *p2ghit*, with variable length (1-3). It returns a tuple like (*p*, *s*) where *p* is the chosen *p2ghit* and *s* is a string with a reason why (it will be printed and stored in a file). The native function is the quite complex, and takes into account the presence of frameshifts, presence of stop codons, aligned Sec position (for selenoprotein families), length of coding sequence (you can inspect the code at *def choose_prediction* in *selenoprofiles.py*). Let's see an example in which this function is replaced by a simple hierarchal function, choosing predictions by genewise over those by exonerate, over those by blast (note that it is still possible that even blast is chosen in this way, if for a given hit the exonerate and genewise predictions are empty or non-valid). Put this into your *extension.py* file provided to option *-add*:

```
global choose_prediction
def choose_prediction(candidates):
    for c in candidates:
        if c.prediction_program()=='genewise': return ( c, 'genewise is available')
    for c in candidates:
        if c.prediction_program()=='exonerate': return ( c, 'exonerate is 2nd best')
    return (candidates[0], 'only blast available')
```

When writing a new *choose_prediction* function, you may still want to call internally the old function, which you can refer to as *choose_prediction_selenoprofiles*. In this example, the new function keeps the behavior of the old one, except for blast predictions which are forced to be never chosen. This is accomplished by returning an *empty_p2g()* object when only blast is available.

```
global choose_prediction
def choose_prediction(candidates):
    if all( [ c.prediction_program()=='blast' for c in candidates ] ):
        return empty_p2g(), 'excluding blast'
    else:
        return choose_prediction_selenoprofiles(candidates)
```

The second such function is named *choose_among_overlapping_p2gs_intrafamily* and is used when removing intrafamily redundancy. This accepts two *p2ghit* that were found overlapping and returns the best one, which is kept. The default function calls internally *choose_prediction*. In its code, this is named *choose_prediction_selenoprofiles*, so if you override the *choose_prediction*, the *choose_among_overlapping_p2gs_intrafamily* function will still run the original, built-in procedure.

If you want to remove intrafamily redundancy using an overridden *choose_prediction* function, it is necessary to override *choose_among_overlapping_p2gs_intrafamily* too. You can search its code in *selenoprofiles_3.py* as a template.

The third and last function is named *choose_among_overlapping_p2gs_interfamily* and is used when removing redundancy between gene predictions by various profiles. This also accepts two *p2ghit* and returns one. The default function considers the AWSI score of the candidate with the 2 profiles, and their filtered attribute (a prediction kept by a profile is never masked by an overlapping prediction filtered by another profile). Let's see how to

replace it with a function which always keeps the prediction with longer protein sequence. Create an *extension.py* file like this:

```
global choose_among_overlapping_p2gs_rem_red
def choose_among_overlapping_p2gs_rem_red(p2g_hit_A, p2g_hit_B):
    if len(p2g_hit_A.protein()) > len(p2g_hit_B.protein()): return p2g_hit_A
    elif len(p2g_hit_A.protein()) < len(p2g_hit_B.protein()): return p2g_hit_B
    else: return p2g_hit_A
```

If you believe that your own function may be useful to other users, or if you need help building your own function, feel free to contact me (see email on the cover page).

Custom prediction features

Selenoprofiles offers the possibility to annotate and manipulate custom features linked to gene predictions. Such annotations (*p2g_features*) can be used for example for protein motifs or domains, or signal sequences, or secondary structures, present in all or some gene predictions. Within selenoprofiles, SECIS elements are implemented as *p2g_features*. Technically, *p2g_feature* is a python class, thought to be generic so the user can create a child-class (subclass) to adapt it to his specific purpose.

Selenoprofiles includes a built-in example to show the capabilities of *p2g_features*: the class *protein_motif*. This is thought to annotate a short motif within the protein sequence, the redox box, expressed as the perl-like regular expression *C..C* (*C* stands for cysteine, and *.* means any character). The class *protein_motif* allows to detect these motifs and easily integrate them in the p2g or gff output.

For any custom *p2g_feature*, the user has to define at least the following procedures: how to search and assign these features, how to dump them in the sqlite database, how to load them back. Then, optionally one can define how to output them to the gff and/or p2g file, and also how to reload the features if gene structure predictions are modified. The *protein_motif* includes examples of all these procedures.

All the code relevant to the *protein_motif* is here below, copied from *selenoprofiles_3.py*.

```
def annotate_protein_motif(p, silent=False):
    """p is a p2ghit. This is an example of method to annotate the p2g_feature protein_motif. To use,
    add this to the main configuration file:
    ACTION.post_filtering.annotate_motif = "if x.filtered == 'kept': annotate_protein_motif(x)"
    """
    s = protein_motif.motif.search( p.protein() )    ##using search method of re.RegexObject --
    protein_motif.motif is such an object
    while s:
        protein_motif_instance = protein_motif()
        protein_motif_instance.start = s.start()+1    #making 1 based
        protein_motif_instance.end = s.end()          #making 1 based and included, so it'd be +1-1
        protein_motif_instance.sequence = \
            p.protein() [ protein_motif_instance.start-1 : protein_motif_instance.end ]
        p.features.append(protein_motif_instance)    ## adding feature to p2g object
        if not silent: printerr('annotate_protein_motif found a motif: ' \
            +protein_motif_instance.output()+ ' in prediction: '+p.output_id(), 1)
        s = protein_motif.motif.search( p.protein(), pos= s.start()+1 ) ## searching again, starting from
        just right of the previously found position

class protein_motif(p2g_feature):
    """ protein_motif is an example of a p2g_feature, to annotate the positions of a certain motif
    defined as a perl-style regexp. The motif is defined in the line following this, as a class
    attribute. In the example, the redox box (CXXC) is the motif.
    Attributes:
    - start      start of the protein motif in the protein sequence (1-based, included)
    - end        end of protein motif in the protein sequence (1-based, included)
    - sequence   motif sequence
    """
    motif = re.compile( 'C..C' )
    included_in_output = True
    included_in_gff = True

def dump_text(self):
    """ Returns a string with all the information for this feature. This string is stored in the
    sqlite database. """
    return str(self.start)+':'+str(self.end)+':'+self.sequence

def load_dumped_text(self, txt):
    """ Reverse the dump_text method: gets a string as input, and loads the self object with the
    information found in that string. """
    start, end, sequence = txt.split(':')
    self.start = int(start);    self.end = int(end);    self.sequence = sequence

def output(self):
    """ Returns a string. This will be added to the p2g output of the prediction to which this
    feature is linked -- if class attribute included_in_output is True"""
    return 'Motif: '+self.sequence+' Start: '+str(self.start)+' End: '+str(self.end)

def gff(self, **keyargs):
    """This must return a gff-like tab-separated string. In this case, we are exploiting and
    overriding the gff method of the gene class, which is a parent class for p2g_feature"""
    ## getting a gene object with the genomic coordinates of the protein motif. we use the gene
    method subseq, which returns a subsequence of the parent gene. Indexes are adjusted for protein-
    nucleotide conversion
    motif_gene_object = self.parent.subseq( start_subseq = (self.start-1)*3 +1, \
        length_subseq = (self.end-self.start+1)*3, minimal=True )
    #now motif_gene_object has a .exons attributes with the genomic coordinates of the protein
    motif. now we can use the native gff method of the obtained gene object
    return gene.gff(motif_gene_object, **keyargs)

def reset(self):
    """ This method is called when the linked prediction is modified, to allow to recompute some or
    all attributes of the feature. In this case, we are removing all features of this class, and
    annotating them again with the same method used to add them in first place:
    annotate_protein_motif"""
    ##removing instances of this class
    for index_to_remove in \
        [i for i, f in enumerate(self.parent.features) if f.__class__ == protein_motif ] [::-1]: \
        self.parent.features.pop(index_to_remove)
    #reannotating
    annotate_protein_motif( self.parent, silent=True )
```

The code contains the definition of a class (*protein_motif*, including 5 methods), and the function *annotate_protein_motif*. This function takes as input a *p2ghit* instance, analyzes it, and if any protein motif is found, it populates its *.features* attribute with one *protein_motif* instance for each motif found.

If this function is never run, the *protein_motif* class is unused. As mentioned within the code, to activate it you should add this line to the main configuration file:

```
ACTION.post_filtering.annotate_motif = if x.filtered == 'kept': annotate_protein_motif(x)
```

In this way, the *annotate_protein_motif* will be run on every prediction that passed filtering. The protein motif *C..C* is defined as the class attribute *motif*, which is of type *RegexObject* from the pattern matching module *re*. Inside the *annotate_protein_motif* function, it is searched in the predicted protein sequence its dedicated method *search*. For each motif found, a *protein_motif* instance is created, and the start and end positions of the match are stored within this object; the protein sequence of the motif is also derived and stored. Once the *protein_motif* instance is ready, it is appended to the *.features* list attribute of the input *p2ghit*. Shortly after, this *p2ghit* reaches the database step, and its information is stored as a sqlite entry. All the features associated to it are also stored in the database. For this reason, the method *dump_text* is called on every feature instance. This method must return a string containing all the information sufficient to then load it back. The method *load_dumped_text* is its reverse, and is used during the output phase to load the dumped information from the database into an empty *protein_motif* instance. An annotating function (in this case *annotation_protein_motif*), and the *p2g_feature* class methods *dump_text* and *load_dumped_text* are the minimal set of definitions to make a functional feature. Other attributes and methods can be used to output the features. To output features to the native selenoprofiles format (*.p2g*, [previously illustrated](#)), the class attribute *included_in_output* must be *True*, and the *output* method has to be defined. Features can be used for gff output too, if the class attribute *included_in_gff* is set to *True*. In this case, it makes sense to take advantage of the *gene* class, the parent of both classes *p2ghit* and *p2g_feature*. The *gene* object is designed to represent a genomic interval, optionally composed by multiple exons, on a certain chromosome (or scaffold) of a target file. It provides plenty of methods such as for fasta fetching, cutting subsequences, computing overlaps, merging gene structures and so on. Its native *gff* method returns one line for each exon in the object, reporting its coordinates and optionally other attributes. In the example above, the *protein_motif* class is not really used as a *gene* object, but just as a data container for the attributes *start*, *end*, *sequence*: its attributes *chromosome*, *strand*, *exons* are not used. Instead, the correct genomic coordinates of the protein motif are derived dynamically, and added to output by overriding the native *gff* method of the class *gene*. For each motif instance, its start and end positions relative to the full protein sequence are available. Thus, the *gene* method *subseq* is used to derive the global genomic coordinates of the motif. This function accepts as input a *gene* (self) object, a start position and a region length, and returns another *gene* object, which contains a subset of the genomic intervals in the self object. If the desired region spans any exon boundary, the returned object contains multiple exons. In the code, the indexes are adjusted for converting protein-based to nucleotide-based positions. Once the appropriate gene object containing the global genomic coordinates for the motif is ready (*motif_gene_object*), the native *gene* class *gff* method can be called.

Lastly, the method *reset* can be defined for custom features that have to be recomputed when the predictions are modified, by actions such as those explained in [improving predictions](#). In the example, the *protein_motif* instances are searched and expelled from the *features* list of the *p2ghit* object for which the *reset* function is run. Then, the annotating function *annotate_protein_motif* is run again.

Appendix 1: guide to profile building

Building good profiles is of key importance for the accuracy of predictions. Their sensitivity and specificity mostly depends on their sequence variation (many representatives for a family are better than few), and on the filters used. The best way to build good profiles is to progressively tune them by inspecting results. If you plan to search a large number of genomes, it is a good routine to begin with just a few of them to get the profile right. First thing on the checklist is the number of processed blast hits. If there are thousands, you should tighten up the blast filtering procedure. Then, ideally the genes in output should be inspected, to see if they fit your expectations.

You can parse log files for OK tags, indicating an output gene, or *DROPPED* tags, that denotes predictions discarded by the filter, as well as for *WARNING* or *ERROR* tags to see if everything went fine. Then, the programs *selenoprofiles_join_alignments* and *selenoprofiles_tree_drawer* constitute useful tools to collect and visualize results.

If there are too many genes in output, or too few, try and change the filtering procedures. By default, the stringency of a profile depends on the distribution of the AWSI scores of its sequences, which measure how similar its sequences are among themselves. For each candidate result, a AWSI score is computed and compared with the profile distribution, computing a Z-score which must be greater than -3 to pass the filter. A simple way to control the stringency of a profile is to alter the minimum Z-score of its filtering procedure:

```
p2g_refiltering = x.aws_i_filter(z_score=-5)
```

Using the AWSI Z-score, profiles with very similar sequences accept only results which are also very similar, while broader profiles are more loosely filtered. Thus, a good profile should possess an amount of sequence variation which is not too low, nor too high. As a rule of thumb, profiles should contain more than ten sequences, but no more than a few hundreds. The script *selenoprofiles_build_profiles* can be used with option *-r* to remove redundancy in an input alignment, in order to trim large profiles to an acceptable number of sequences. The same script can be used with option *-d* to inspect the AWSI distribution of a profile. Generally the profiles with AWSI cut-offs between 0.2 and 0.6 work reasonably well. If the cut-off is higher, it means that the profile is extremely conserved, and thus will output only extremely similar candidates. In this case stringency can be lowered by setting manually a AWSI cut-off independent of the Z-score. The same *aws_i_filter* function can be used, as it accepts also a AWSI threshold: a candidate is accepted if either the AWSI or the Z-score are higher than the respective thresholds.

```
p2g_refiltering = x.aws_i_filter(aws_i=0.5)
```

If the default AWSI cut-off is very low, it means that the profile is too broad, containing sequences too dissimilar to each other. If large, the best strategy is generally to split the input alignment into two or more profile alignments. Alternatively, one can try to keep the profile as it is, and set an efficient filter using the tools explained in this manual.

A useful filtering tool is the coverage: the prediction is mapped into the profile, and the distance between its projected boundaries, divided by the profile alignment length gives the coverage. A strict coverage filter excludes partial protein predictions:

```
p2g_refiltering = x.coverage()>0.75
```


When you are searching for protein families containing of common domains, you may want to exclude the hits limited to these protein regions, using again the positions of the prediction mapped to the profile:

```
p2g_refiltering = not x.is_contained_in_profile_range(1, 60) and not
                  \\\ x.is_contained_in_profile_range(100, 160)
```

The tag and GO score are powerful tools to allow to discriminate even between similar protein families. Both tag and GO score procedures require a run of blastp against nr, and thus are quite computationally expensive. For this reason, they should be used only for the most difficult profiles, for which the AWSI score is not enough to differentiate bona-fide genes and spurious hits. Even then, it is worth to additionally limit the number of results for which this is run, for example checking AWSI. In this example, all results with AWSI greater than 0.6 automatically pass the filter, while for those with AWSI between 0.2 and 0.6 the *go_score* is evaluated.

```
p2g_refiltering = x.awsi()>0.6 or ( x.awsi()>0.2 and x.go_score()>0 )
```

The tags should be written by searching the results with blastp against nr and looking at protein titles. For GO scoring, the script *selenoprofiles_build_profiles* provides a utility to find suitable terms, if the input profile sequences contain gi codes from ncbi nr. The GO annotations for all profile sequences are fetched, and their number is compared with the total number of proteins for each GO term.

Appendix 2: full list of operations

Load variables and functions:

- Read configuration file
- Read command line
- Check presence of target file and profiles
- Check/convert species name
- Initialize results database if necessary
- Read active actions
- Read parameters

Load file provided with -add option

Load/compute length of all chromosomes in the target file

For each input profile:

- Load/compute clusters of profile alignment
- Check if results are already in database. If so, skipping all these steps:
- For each cluster:

- Run/load psitblastn
- For each blast hit in the blast output for this cluster:
 - Transform it to have it relative to the master blast query
 - Replace "*" with U in the target sequence if a UGA is aligned to a Sec position
 - Perform pre_blast_filter actions
 - Evaluate if blast hit passes blast filtering. If it doesn't, discard it
 - Perform post_blast actions

If more than one cluster: merge overlapping blast hits from the different cluster searches

Merge blast hits by colinearity

- For each blast hit: Perform post_blast_merge actions
- For each blast hit: Run/load exonerate using blast hit as seed
- Discard duplicated exonerate hits and the blast hits associated to them
- For each blast hit:

- If an exonerate hit is available: run/load genewise using it as seed
- Else: run/load genewise using the blast hit as seed (genewise_to_be_sure routine)

For each blast hit:

- For each non-empty prediction among blast, exonerate, genewise:
- Perform pre_choose actions

Check if the choose prediction output file is already present. If not:

- Choose a prediction among the available ones: blast, exonerate, genewise
- Assign label to the chosen prediction

Write choose prediction output file

For each prediction: Perform pre_filtering actions

Check if the filtering predictions output file is already present. If not:

Determining the overlap between predictions

For each prediction:

- If the prediction overlaps an identical or smaller prediction, filter it as "redundant"
- Else, evaluating p2g_filtering. If it doesn't pass, filter prediction as "filtered"
- Else, evaluating p2g_refiltering. If it doesn't pass, filter prediction as "refiltered"
- Else: filter prediction as "kept"

Writing filtering predictions output file

For each prediction: Perform post_filtering actions

Write predictions (including their filtered state) in the database

Checking if results from different families overlap each other. Filtering those as "overlapping"

For each input profile:

Computing list of predictions to be output (based on output states / output filter)

For each prediction to be output:

Perform pre_output actions

For each active output format:

If the output file is not already present: write output file

Write alignment output (with all predictions to be output along with profile sequences)

Appendix 3: links and references

Selenoprofiles:

Mariotti M, Guigó R. Selenoprofiles: profile-based scanning of eukaryotic genome sequences for selenoprotein genes. *Bioinformatics*. 2010 Nov 1;26(21):2656-63

website: <http://big.crg.cat/services/selenoprofiles>

Blast:

Altschul SF, Madden TL, Schäffer AA, Zhang J, Zhang Z, Miller W, Lipman DJ. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res*. 1997 Sep 1;25(17):3389-402. Review.

installation: <ftp://ftp.ncbi.nlm.nih.gov/blast/executables/release/LATEST/>

Exonerate:

Slater GS, Birney E. Automated generation of heuristics for biological sequence comparison. *BMC Bioinformatics*. 2005 Feb 15;6:31.

website: <http://www.ebi.ac.uk/~guy/exonerate/>

Genewise:

Birney E, Clamp M, Durbin R. GeneWise and Genomewise. *Genome Res*. 2004 May;14(5):988-95.

website: <http://www.ebi.ac.uk/Tools/Wise2/>

installation: <ftp://ftp.ebi.ac.uk/pub/software/unix/wise2/wise2.2.0.tar.gz>

NCBI protein databases:

search: <http://www.ncbi.nlm.nih.gov/sites/entrez?db=protein&itool=toolbar>

download: <ftp://ftp.ncbi.nih.gov/blast/db/FASTA/nr.gz>

Gene ontology:

website: <http://www.geneontology.org/>

The python code to query the gene ontology used in selenoprofiles is partially from:

<http://gitorious.org/annotation/annotation/trees/master>.

which is an adaptation by François Serra of the code by Nepusz Tamás (**thanks** to both!)

<https://github.com/ntamas/biopython>

MAFFT alignment program:

Katoh K, Asimenos G, Toh H. Multiple alignment of DNA sequences with MAFFT.

Methods Mol Biol. 2009;537:39-64.

website: <http://mafft.cbrc.jp/alignment/software/>

ETE2 tree visualization:

Huerta-Cepas J, Dopazo J, Gabaldón T. ETE: a python Environment for Tree Exploration. *BMC Bioinformatics* 2010, 11:24.

website: <http://ete.cgenomics.org/>;

PyLab graph visualization:

website: <http://www.scipy.org/PyLab>

SECISearch3:

Mariotti M, Lobanov AV, Guigó R, Gladyshev VN. SECISearch3 and Seblastian: new tools for prediction of SECIS elements and selenoproteins. *Nucleic Acids Res*. 2013; manuscript in publication.

website: <http://seblastian.crg.es/> or <http://gladyshevlab.org/SelenoproteinPredictionServer/>

Appendix 4: troubleshooting

Here's some errors that I experienced often installing selenoprofiles and the required slave programs in different systems. If you have selenoprofiles errors which are not reported here, contact me (see email address in the cover page).

Blast error

Selenoprofiles runs the *blastpgp* binary (to build a PSSM for each profile) through symbolic links in its installation directory. In some systems this may cause this error:

```
[blastpgp] WARNING: Unable to open BLOSUM62
[blastpgp] WARNING: BlastScoreBkMatFill returned non-zero status
[blastpgp] WARNING: SetUpBlastSearch failed.
```

Blast cannot find the BLOSUM62 matrix, that is to say, its installation data folder. To fix the problem, edit (or create) the file `~/.ncbirc` and add something like this to its content:

```
[NCBI]
data=/path_to_blast_installation/blast-2.2.2x/data
```

To know what is your blast installation folder, use the *which* command in bash (e.g. *which blastpgp*) and follow possible symbolic links until you have something like:

```
/path_to_blast_installation/ncbi_blast-2.2.2x/bin/blastpgp
```

The data folder to insert in `~/.ncbirc` is then the one shown above.

Genewise errors

Genewise is part of the wise2 package that can be found here (newer versions may exist): <ftp://ftp.ebi.ac.uk/pub/software/unix/wise2/wise2.2.0.tar.gz>

In some systems, an error appears as you build the program with *make*:

```
sqio.c:232: error: conflicting types for 'getline'
/usr/include/stdio.h:653: note: previous declaration of 'getline' was here
make[1]: *** [sqio.o] Error 1
make[1]: Leaving directory `/PATH/src/HMMer2'
make: *** [realall] Error 2
```

The problem is in a function declaration (*getline*) in the file `HMMer2/sqio.c`, since this function is already declared in many compilers. To solve it, type:

```
cd wise2.2.0/src/HMMer2/
sed 's/getline/getline_new/' sqio.c > a && mv a sqio.c
```

Now get back to `wise2.2.0/src/` and type *make all*. Take care of the final message it shows: you need to set the environmental variable *WISECONFIGDIR* to point to right place for genewise to work. If you do not, you may have the following error:

```
Warning Error    Could not open human.gf as a genefrequency file
Warning Error    Could not read a GeneFrequency file in human.gf
Fatal Error      Could not build objects!
```

To take care of this, add to your bash configuration file `~/.bashrc` something like this:

```
export WISECONFIGDIR=/path_to_installation/wise2.2.0/wisecfg/
```

so this will be executed for every bash instance you will run from now on.