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

The Harvard community has made this article openly available. Please share how this access benefits you. Your story matters

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Published Version</td>
<td>doi:10.1093/nar/gkt550</td>
</tr>
<tr>
<td>Citable link</td>
<td><a href="http://nrs.harvard.edu/urn-3:HUL.InstRepos:11855880">http://nrs.harvard.edu/urn-3:HUL.InstRepos:11855880</a></td>
</tr>
<tr>
<td>Terms of Use</td>
<td>This article was downloaded from Harvard University’s DASH repository, and is made available under the terms and conditions applicable to Other Posted Material, as set forth at <a href="http://nrs.harvard.edu/urn-3:HUL.InstRepos:dash.current.terms-of-use#LAA">http://nrs.harvard.edu/urn-3:HUL.InstRepos:dash.current.terms-of-use#LAA</a></td>
</tr>
</tbody>
</table>
SECISearch3 and Seblastian: new tools for prediction of SECIS elements and selenoproteins

Marco Mariotti¹,², Alexei V. Lobanov¹, Roderic Guigo²,* and Vadim N. Gladyshev¹,*

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://seblastian.crg.es. SECISearch3 replaces its predecessor SECISearch as a tool for prediction of eukaryotic SECIS elements. Seblastian is a new method for selenoprotein gene detection that uses SECISearch3 and then predicts selenoprotein sequences encoded upstream of SECIS elements. Seblastian 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. Seblastian 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://seblastian.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 UGAG/KGAW. Additionally, a stretch of
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 ~5200nt 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 SECISe 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, begin-
ning from the core. Like Infernal, Covels predictions
include a bit-score that can be used for filtering. The rec-
commended threshold value is 15. However, it should be
taken into account that for SECIS elements not conform-
ing 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.

We computed an F-score (20) of the methods,
diction methods when applied to the full genomes of these
organisms. We computed an F-score (20) of the methods,
diction methods when applied to the full genomes of these
organisms. We computed an F-score (20) of the methods,
diction methods when applied to the full genomes of these
organisms. We computed an F-score (20) of the methods,
diction methods when applied to the full genomes of these
organisms. We computed an F-score (20) of the methods,
diction methods when applied to the full genomes of these
organisms. We computed an F-score (20) of the methods,
diction methods when applied to the full genomes of these
organisms. We computed an F-score (20) of the methods,
diction methods when applied to the full genomes of these
organisms. We computed an F-score (20) of the methods,
diction methods when applied to the full genomes of these
organisms. We computed an F-score (20) of the methods,
diction methods when applied to the full genomes of these
organisms. We computed an F-score (20) of the methods,
diction methods when applied to the full genomes of these
organisms. We computed an F-score (20) of the methods,
diction methods when applied to the full genomes of these
organisms. We computed an F-score (20) of the methods,
diction methods when applied to the full genomes of these
organisms. We computed an F-score (20) of the methods,
diction methods when applied to the full genomes of these
organisms. We computed an F-score (20) of the methods,
diction methods when applied to the full genomes of these
organisms. We computed an F-score (20) of the methods,
diction methods when applied to the full genomes of these
organisms. We computed an F-score (20) of the methods,
diction methods when applied to the full genomes of these
organisms. We computed an F-score (20) of the methods,
diction methods when applied to the full genomes of these
organisms. We computed an F-score (20) of the methods,
diction methods when applied to the full genomes of these
organisms. We computed an F-score (20) of the methods,
diction methods when applied to the full genomes of these
organisms. We computed an F-score (20) of the methods,
diction methods when applied to the full genomes of these
organisms. We computed an F-score (20) of the methods,
diction methods when applied to the full genomes of these
organisms. We computed an F-score (20) of the methods,
diction methods when applied to the full genomes of these
organisms. We computed an F-score (20) of the methods,
diction methods when applied to the full genomes of these
organisms. We computed an F-score (20) of the methods,
diction methods when applied to the full genomes of these
organisms. We computed an F-score (20) of the methods,
diction methods when applied to the full genomes of these
organisms. We computed an F-score (20) of the methods,
diction methods when applied to the full genomes of these
organisms. We computed an F-score (20) of the methods,
diction methods when applied to the full genomes of these
organisms. We computed an F-score (20) of the methods,
diction methods when applied to the full genomes of these
organisms. We computed an F-score (20) of the methods,
diction methods when applied to the full genomes of these
organisms. We computed an F-score (20) of the methods,
diction methods when applied to the full genomes of these
organisms. We computed an F-score (20) of the methods,
diction methods when applied to the full genomes of these
organisms. We computed an F-score (20) of the methods,
diction methods when applied to the full genomes of these
organisms. We computed an F-score (20) of the methods,
diction methods when applied to the full genomes of these
organisms. We computed an F-score (20) of the methods,
diction methods when applied to the full genomes of these
organisms. We computed an F-score (20) of the methods,
diction methods when applied to the full genomes of these
organisms. We computed an F-score (20) of the methods,
diction methods when applied to the full genomes of these
organisms. We computed an F-score (20) of the methods,
diction methods when applied to the full genomes of these
organisms. We computed an F-score (20) of the methods,
diction methods when applied to the full genomes of these
organisms. We computed an F-score (20) of the methods,
diction methods when applied to the full genomes of these
organisms. We computed an F-score (20) of the methods,
diction methods when applied to the full genomes of these
organisms. We computed an F-score (20) of the methods,
diction methods when applied to the full genomes of these
organisms. We computed an F-score (20) of the methods,
diction methods when applied to the full genomes of these
organisms. We computed an F-score (20) of the methods,
diction methods when applied to the full genomes of these
organisms. We computed an F-score (20) of the methods,
diction methods when applied to the full genomes of these
organisms. We computed an F-score (20) of the methods,
diction methods when applied to the full genomes of these
organisms. We computed an F-score (20) of the methods,
diction methods when applied to the full genomes of these
organisms. We computed an F-score (20) of the methods,
diction methods when applied to the full genomes of these
organisms. We computed an F-score (20) of the methods,
diction methods when applied to the full genomes of these
organisms. We computed an F-score (20) of the methods,
diction methods when applied to the full genomes of these
organisms. We computed an F-score (20) of the methods,
diction methods when applied to the full genomes of these
organisms. We computed an F-score (20) of the methods,
diction methods when applied to the full genomes of these
organisms. We computed an F-score (20) of the methods,
diction methods when applied to the full genomes of these
organisms. We computed an F-score (20) of the methods,
diction methods when applied to the full genomes of these
organisms. We computed an F-score (20) of the methods,
diction methods when applied to the full genomes of these
organisms. We computed an F-score (20) of the methods,
diction methods when applied to the full genomes of these
organisms. We computed an F-score (20) of the methods,
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 1. Workflow of the SECISearch3 program.
SECIS elements in relatively small databases. As default settings, we recommend to use the Infernal model with a score threshold of 10, prioritizing sensitivity.

Seblastian

Based on SECISearch3, we build a new method for selenoprotein gene prediction and analysis: Seblastian. 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), Seblastian 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.

Seblastian 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 Seblastian 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, Seblastian procedures and filters were designed to accept also candidates with multiple Sec residues, such as selenoprotein P.

Testing Seblastian

We benchmarked Seblastian 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 Seblastian 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
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.
prediction owing to spurious similarity with a portion of the selected query (SPS2 from *Saccoglossus kowalevskii*).

The method was also able to find new selenoproteins. Across all eukaryotes, we estimate that Sebastian alone would have identified at least 25% of all known selenoproteins. We believe that this is a remarkable result, given the difficulty of *de novo* prediction of selenoproteins. Indeed, for known selenoproteins, a Blastx alignment between an annotated Sec and a UGA is unlikely to happen by chance, and thus it is a sufficient argument to call a selenoprotein gene. For new selenoproteins, any cysteine of any query is a candidate Sec position. Thus, many false positives arise. Possible false candidates are real genes with sequencing errors occurring in cysteine positions, pseudogenes with a single in-frame UGA in a cysteine position, or non-coding repetitive stretches of sequence matching our criteria just by chance. Therefore, we need to apply the filters described earlier in the text to maintain false positives to a manageable level, even though this procedure would miss some true candidates.

### New selenoprotein candidates

We ran Sebastian on a number of genomes of non-metazoan eukaryotes, which normally represent the most challenging cases. In addition, we expected that some selenoproteins remain undiscovered in some of these lineages, based on the previous searches with other eukaryotic genomes (27,28,33). Sebastian yielded a ranked set of 186 selenoprotein predictions in 25 species. Although we expect a portion of them to be false positives, we also believe that the set includes *bona fide* novel selenoproteins. We implemented a procedure to assign a score to the predicted selenoproteins. The score takes into account the SEICIS-coding sequence distance, the Covels score and the grade of the SEICIS element, the Blastx E-value, the presence of a redox box motif including the candidate Sec, the similarity with other Sebastian candidates, and the matches with EST and protein databases.

The new selenoprotein candidates, the species list and a more detailed explanation of the scoring procedure can be found in Supplementary Material S5.

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 new selenoprotein candidates in our Sebastian predictions were found in *M.brevicollis* and *E.huxleyi*, here framed in orange. The figure shows them aligned with other AhpC selenoproteins predicted using Selenoprofiles in eukaryotes (top) and prokaryotes (bottom). Some metazoan cysteine homologues are also shown on the top. The Sec is found in the highlighted redox box UXXC, present also in vertebrates as CXXC. For the full alignment and further details regarding the search for AhpC proteins, see Supplementary Material S6.

---

**Figure 4.** AhpC selenoproteins. Two selenoprotein candidates in our Sebastian predictions were found in *M.brevicollis* and *E.huxleyi*, here framed in orange. The figure shows them aligned with other AhpC selenoproteins predicted using Selenoprofiles in eukaryotes (top) and prokaryotes (bottom). Some metazoan cysteine homologues are also shown on the top. The Sec is found in the highlighted redox box UXXC, present also in vertebrates as CXXC. For the full alignment and further details regarding the search for AhpC proteins, see Supplementary Material S6.
metazoan cysteine homologues. Among the Selenoprofiles AhpC predictions, we also found our second best scoring Seblastian candidate, in the *Emiliania 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 Seblastian, 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 Seblastian 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 webservice for SECISearch3 and Seblastian**

We created a web server to allow users world-wide to upload any nucleotide sequence and run SECISearch3 and/or Seblastian (Figure 5). It is hosted both at http://gladyshevlab.org/SelenoproteinPredictionServer and at http://seblastian.crg.es. The user can choose to run Seblastian 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 Seblastian 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 Seblastian. 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. Seblastian 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 Seblastian 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 Seblastian
are public and can be run on a dedicated web server at http://gladyshevlab.org/SelenoproteinPredictionServer or http://seblastian.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**