Ten easy steps to annotate genes, proteins and SECIS elements in `SelenoDB`

Sergi Castellano

— November 1, 2005 —

Abstract

Easy-to-follow protocol to annotate genes, proteins and SECIS elements in `SelenoDB`. Please, follow these steps and submit annotations to `scaste@imim.es`.

Cell and Molecular Biology
University of Hawaii at Manoa

Current address:
Molecular Biology and Genetics
Cornell University

General Rules

1. These instructions are only valid for genes with only one transcript and one SECIS. They should be valid for all prokaryotic and most eukaryotic selenoproteins. More complex cases will be adressed soon.
2. Use "NULL" as indicated or when data is Not Available (=unknown). Eg. the program used to predict an exon is not known.
3. Use "N/A" when the information required is Not Applicable to that field. Eg. no middlename for an author.
4. Separate fields (columns) with tabs. Be aware that copy&paste may turn tabs into simple blanks.
5. SelenoDB initial version is 1.0 (release.revision).
6. All features annotated for the first time have a version number of 1.0 (release.revision).
7. All features reannotated increase their revision number in steps of 1. Eg. 1.1, 1.2,...,1.n
8. Release number only changes when SelenoDB version changes (major modifications are needed for this to happen).
9. The source genomic sequence as obtained from the reference database is considered to be in forward.
10. Features are annotated in forward or reverse depending on the orientation of the feature in the source genomic sequence.
11. All coordinates are given in relation to the forward source genomic sequence.
12. Upstream features (in relation to the forward strand) are annotated first than the downstream features.
13. In general, first letter of first word in each annotation field should be uppercase.

Step 0: Set up the annotation directory

Name one directory after the species and, within this, name another directory after the gene subfamily (if any) or family you want to annotate:

```
Species/(sub)family
```

Example:

```
H.sapiens/GPx1, D.melanogaster/SelH, E.coli/SPS2
```

Note: There is not a unique way to name subfamilies. In general, I suggest to use numbers (eg. GPx1, GPx2...) when functional differences have been described and letters when functional differences are unclear or unknown (eg. SelJa, SelJb).

Note: Usually, Cys- and Sec-containing proteins of the same family and subfamily (if any) will share the same name.

Step 1: Author layer

In the gene directory above, create an `author.data` file with the following data:

```
firstname   middlename   lastname   email   NULL   NULL
```

Example:

```
Robert   J.  Stillwell   r.j.stillwell@gmail.com   NULL   NULL

Sergi   N/A   Castellano   scaste@imim.es   NULL   NULL
```

Note: We keep track of the features annotated by each curator to help fix annotation errors.

Step 2: Species layer

In the gene directory above, create an `species.data` file with the following data:

```
scientific_name_gen   scientific_name_sp   common_name   taxonomy_link   NULL   NULL
```

Example:

```
Homo   sapiens   Human   http://www.ncbi.nlm.nih.gov/Taxonomy/Browser/wwwtax.cgi?id=9606   NULL   NULL
```

Note: Use scientific and common name (if any) from NCBI taxbrowser. If no common name exists use N/A.
Note: All taxonomy links are http://www.ncbi.nlm.nih.gov/Taxonomy/Browser/wwwtax.cgi?id= plus the taxon ID.

Step 3: Family layer

3.1

In the gene directory above, create a `family.data` file with the following data:

```
family_symbol   family_name   description   NULL   NULL
```

Example:

```
GPx   Glutathione Peroxidase   Reduction and detoxification of different types of peroxides...   NULL   NULL
```

Note: If symbol and/or name does not exist, we will assign a new and unique symbol/name to each annotated gene subfamily (symbols and names cannot be repeated). Symbols and names do not vary among species.
Note: This file is only needed if a gene belonging to this subfamily is annotated for the first time.

3.2

In the gene directory above, create a `subfamily.data` file with the following data:

```
subfamily_symbol   family_symbol   subfamily_name   description   NULL   NULL
```

Example:

```
GPx1   GPx   Glutathione Peroxidase 1   Cytosolic Glutathione Peroxidase   NULL   NULL
```

Note: If symbol and/or name does not exist, we will assign a new and unique symbol/name to each annotated gene subfamily (symbols and names cannot be repeated). Symbols and names do not vary among species.
Note: This file is only needed if a exist and a gene belonging to this subfamily is annotated for the first time.

3.3

In the gene directory above, create a `family_symbol_syn.data` file with the following data:

```
family_symbol_syn   family_symbol   NULL   NULL
```

Example:

```
SEPN   SelN   NULL   NULL

SelX   SelR   NULL   NULL

MsrB1   SelR   NULL   NULL
```

Note: This file is only needed if synonimous symbol(s) exist and a gene belonging to this family is annotated for the first time.

3.4

In the gene directory above, create a `family_name_syn.data` file with the following data:

```
family_name_syn   family_name   NULL   NULL
```

Example:

```
Methionine-R-sulfoxide reductase 1    Selenoprotein R   NULL   NULL

Selenoprotein X   Selenoprotein R   NULL   NULL
```

Note: This file is only needed if synonimous name(s) exist and a gene belonging to this family is annotated for the first time.

3.5

In the gene directory above, create a `subfamily_symbol_syn.data` file with the following data:

```
subfamily_symbol_syn   subfamily_symbol   NULL   NULL
```

Example:

```
GPx1   cGPx   NULL   NULL
```

Note: This file is only needed if synonimous symbol(s) exist and a gene belonging to this subfamily is annotated for the first time.

3.6

In the gene directory above, create a `subfamily_name_syn.data` file with the following data:

```
subfamily\_name_syn   subfamily_name   NULL   NULL
```

Example:

```
Cellular glutathione peroxidase   Glutathione peroxidase 1   NULL   NULL
```

Note: This file is only needed if synonimous name(s) exist and a gene belonging to this subfamily is annotated for the first time.

Step 4: Analysis layer

In the gene directory above, create a `analysis.data` file with the following data:

```
program   program_version   program_type   description   NULL   NULL
```

Where program type is one of the following:

```
Ab initio gene prediction
Comparative gene prediction
Protein homology
Transcript homology
```

Example:

```
genewise   1.5   Protein homology   Human protein as template   NULL   NULL
```

Note: Description is free text but consistent for the same program.

Step 5: Sequence layer

In the gene directory above, create a `sequence.data` file with the following data:

```
NULL   scientific_name_gen   scientific_name_sp   sequence   NULL   NULL
```

Where the sequence has been extracted in the following way:

1. Genomic in prokaryotes: the sequence encompassing the gene (which includes the SECIS) plus 1000 nt upstream (promoter region at the start of operons) and 1000 downstream. The annotated gene should start at position 1001. Longest possible genomic sequence stored containing a gene is 16Mb (contact me if you come across a selenoprotein gene whose length exceeds this limit).
2. Genomic in eukaryotes: the sequence encompassing the gene (including SECIS) plus 1000 nt upstream (promoter region) and 1000 downstream. The annotated gene should start at position 1001. Longest possible genomic sequence stored containing a gene is 16Mb (contact me if you come across a selenoprotein gene whose length exceeds this limit).
3. Transcript: the most complete transcript.
4. Protein: the most complete protein

Example:

```
NULL   Drosophila   melanogaster   AAAAAGGGTTTTTTCCCCTTTTTTTTT   NULL   NULL
```

Step 6: Feature version annotation layer

6.1

In the directory above, create a `exon_version.data` file with the following data:

```
NULL   NULL   NULL   program   program_version   seq_start   seq_end   version   NULL   NULL
```

Example:

```
NULL   NULL   NULL   genewise   1.5   1234   1345   1.0   NULL   NULL
```

6.2

In the gene directory above, create a `transcript_version.data` file with the following data:

```
NULL   NULL   NULL   version   NULL   NULL
```

Example:

```
NULL   NULL   NULL   1.0   NULL   NULL
```

6.3

In the gene directory above, create a `gene_version.data` file with the following data:

```
NULL   NULL   subfamily_name   family_name   version   strand   NULL    NULL
```

Example:

```
NULL  NULL  Glutathione Peroxidase 1  Glutathione Peroxidase  1.0  Forward  NULL  NULL
```

Note: Strand is Forward or Reverse.

6.4

In the gene directory above, create a `translation_version.data` file with the following data:

```
NULL NULL program program_version seq_start start_exon_version_id seq_end end_exon_version_id TGA_seq_start start_TGA_exon_version_id end_TGA_exon_version_id version NULL NULL
```

Example:

```
NULL   NULL   Conceptual translation   Standard Code with Sec   8947   NULL   18397   NULL   13056   NULL   1.0   NULL   NULL
```

Note: start_exon_version_id, end_exon_version_id, start_TGA_exon_version_id and end_TGA_exon_version_id will be updated automatically.

6.5

In the gene directory above, create a `secis_version.data` file with the following data:

```
NULL  NULL  program  program_version  seq_start  start_exon_version_id  seq_end  end_exon_version_id  version  NULL  NULL
```

Example:

```
NULL   NULL   secisearch   2.19   3456   NULL   3521   NULL   1.0   NULL   NULL
```

Note: `start_exon_version_id` and `end_exon_version_id` will be updated automatically.

Step 7: External reference layer

In general, we will provide only one external reference and only for some features. If possible, links should be made to the following databases in the order shonwn:

1. Ensembl (It already annotates properly some selenoproteins): for genes, transcripts and proteins
2. Refseq: for transcripts
3. Uniprot: for proteins
4. Genbank: for genes, transcripts and proteins
5. Well established species-specific databases
6. Well established protein-family-specific databases

7.1

In the gene directory above, create a `sequence_ext_ref.data` file with the following data:

```
NULL sequence_type sequence_ext_start sequence_ext_start sequence_ext_id db db_link sequence_ext_link NULL NULL
```

Where `sequence_type` is one of the following:

```
Genome -assembly-
Genome -traces-
Transcript -mRNA/cDNA-
Transcript -EST-
Protein -complete-
Protein -fragment-
```

Example:

```
NULL  Genome -assembly-  345678  355789  AL022328.21.1.177241  Ensembl  http://www.ensembl.org  http://www.ensembl.org/Homo_sapiens/contigview?region=AL022328.21.1.177241  NULL  NULL
```

Note: `sequence_ext_start` and `sequence_ext_start` refer to the positions of the extracted sequence (up to 16Mb) in the source sequence (any size).

7.2

In the gene directory above, create a `transcript_version_ext_ref.data` file with the following data:

```
NULL  transcript_version_ext_id  db  db_link  transcript_version_ext_link  NULL  NULL
```

Example:

```
NULL  ENST00000248845  Ensembl  http:///www.ensembl.org  http://www.ensembl.org/Homo_sapiens/transview?transcript=ENST00000248845  NULL  NULL
```

Note: This file is only needed if the transcript is known to be annotated somewhere.

7.3

In the gene directory above, create a `gene_version_ext_ref.data` file with the following data:

```
NULL  gene_version_ext_id  db  db_link  gene_version_ext_link  NULL  NULL
```

Example:

```
NULL  ENSG00000073169  Ensembl  http:///www.ensembl.org  http://www.ensembl.org/Homo_sapiens/geneview?gene=ENSG00000073169  NULL  NULL
```

Note: This file is only needed if the gene is known to be annotated somewhere.

7.4

In the gene directory above, create a translation_version_ext_ref.data file with the following data:

```
NULL  protein_version_ext_id  db  db_link  protein_version_ext_link  NULL  NULL
```

Example:

```
NULL  ENSP00000248845  Ensembl  http:///www.ensembl.org  http://www.ensembl.org/Homo_sapiens/protview?peptide=ENSP00000248845  NULL  NULL
```

Note: This file is only needed if the protein is known to be annotated somewhere.

Step 8: Feature version description layer

This annotation layer is intended to provide more biological and functional annotation of genes, proteins and SECIS. This layer will be expanded in the future as we discuss which type of information is most relevant to selenoproteins. In the future, this layer should be made accessible to researchers (one per gene or gene family) for functional annotation through a web annotation interface. Relevant literature references could be included in this layer.

8.1

In the directory above, create a `gene_version_description.data` file with the following data:

```
gene_version_id  description  bio_function  NULL  NULL
```

Example:

```
NULL  blablablabla  blablablabla  NULL  NULL
```

Note: description is a free text "physical" description of the gene.

8.2

In the directory above, create a `transcript_version_description.data` file with the following data:

`transcript_version_id  description  bio_function  NULL  NULL`

Example:

`NULL  blablablabla  blablablabla  NULL  NULL`

Note: description is a free text "physical" description of the transcript.

8.3

In the directory above, create a `translation_version_description.data` file with the following data:

`translation_version_id  description  bio_function  NULL  NULL`

Example:

`NULL  blablablabla  blablablabla  NULL  NULL`

Note: description is a free text "physical" description of the protein.

8.4

In the directory above, create a`secis_version_description.data` file with the following data:

`secis_version_id  description  bio_function  type  NULL  NULL`

Example:

`NULL  blablablabla  blablablabla  1  NULL  NULL`

Note: type is 1 or 2.
Note: `description` is a free text "physical" description of the SECIS.

Step 9: Recheck, pack and compress all files

Take some time to revise the annotation of all the biological features obtained with your gene annotation pipeline. All the files you just created will be automatically imported into SelenoDB and used to generate many additional linking tables. Please, make sure they conform to this guidelines and to the table structure depicted in the SelenoDB schema.

```
tar cvf Species.tar Species/ or

gzip -9 Species.tar
```

Example:

```
tar cvf H.sapiens.tar H.sapiens/

gzip -9 H.sapiens.tar
```

or

```
tar cvf Species_(sub)family.tar Species/(sub)family

gzip -9 Species_(sub)family.tar
```

Example:

```
tar cvf H.sapiens_GPx1.tar H.sapiens/GPx1

gzip -9 H.sapiens_GPx1.tar
```

Step 10: Submit files

Send

`Species.tar.gz`

or

`Species\_(sub)family.tar.gz`

to

`scaste@imim.es`