OrthoSelect
User Guide

Fabian Schreiber

October 22, 2008

Disclaimer The program OrthoSelect is a beta-test version which is still under development. The author is not aware of bugs that would cause the program to obtain incorrect results, but they could exist. Even though the author tries to make this program as reliable as possible, it could be that parts of the program do not work as intended. Please report any crashes, bugs, or problems you have with this program (fschrei@gwdg.de). This program is distributed in the hope that it will be useful, but without any warranty.

Contents

1 Introduction

DNA and protein sequences provide a wealth of information which is routinely used in phylogenetic studies. Traditionally, single genes or small groups of genes have been used to infer the phylogeny of a group of species under study. It has been shown, however, that molecular phylogenies based on single genes often lead to apparently conflicting tree hypotheses (Delsuc *et al.*, 2005). The combination of a large number of genes and species in genome-scale approaches to reconstruct phylogenies can be useful to overcome these difficulties (Gee, 2003). This approach has been termed *phylogenomics* (Eisen, 1998).

Since complete genome sequences are available only for a limited number of species, many phylogenomic studies rely on EST sequences. EST sequences are short (200 - 800 bases), unedited, randomly selected single-pass reads from cDNA libraries that sample the diversity of genes expressed by an organism or tissue at a particular time under particular conditions. The relatively low cost and rapid generation of EST sequences can deliver insights into transcribed genes from a large number of taxa. Moreover, EST sequences contain a wealth of phylogenetic information. Several recent phylogenomic studies used EST sequences to generate large data matrices (Bourlat *et al.*, 2006; Delsuc *et al.*, 2006; Dunn *et al.*, 2008). Such studies start with the generation of EST libraries for a set of species. ESTs are then assembled, and ortholog genes are identified as a basis for phylogenetic reconstruction. Phylogenetically related sequences are called ortholog if they were separated by a speciation event, as opposed to paralog sequences, which were separated by a duplication event within the same species (Fitch, 1970). Orthologs are usually functionally conserved whereas paralogs tend to have different functions (Koonin, 2005) and are less useful in phylogenetic studies.

A typical protocol for detecting orthologs in phylogenomic studies should include (1) a similarity search using tools like BLAST (Altschul *et al.*, 1997), (2) a strategy to select a subset of hits returned by this search, (3) a criterion to identify sequences as potential orthologs, (4) a strategy for eliminating potential paralogs - in case several potential orthologs from the same species have been assigned to the same OG.

Orthology assignment is a crucial prerequisite for phylogeny reconstruction as faulty assumptions about orthology – e.g. the inclusion of paralogs – can lead to an incorrect tree hypothesis (Zmasek and Eddy, 2002). Errors can result from similarity searches against non-specialized databases, e.g. NCBI's *nr* database, or from best-hit selection strategies such as *best reciprocal hit* (Mushegian *et al.*, 1998) or *best triangular hit* that may lead to false positive orthology predictions. The similarity between a query and a database sequence stemming from a similarity search – expressed for example as a bit-score or expectation value – is usually taken as a criterion to predict an ortholog relationship. Since the results of these methods depend on the choice of a database and on the strategy to select sequences from similarity search hits, a more reliable protocol for ortholog predictions is needed.

Several databases and computational methods for predicting orthologs have

been implemented. Multi-species ortholog databases have been developed based on different sources of ortholog information. They include information about ortholog relationships between sequences. The OrthoMCL-DB database (Chen *et al.*, 2006) has been constructed on the basis of whole genome comparisons, HomoloGene (Zhang *et al.*, 2000) on the basis of synteny and HOVERGEN (Duret *et al.*, 1994) was constructed using the information from phylogenetic trees. Two of these databases, OrthoMCL-DB and KOG (Tatusov *et al.*, 2003), explicitly define ortholog groups (OG) which can be used as a basis for orthology assignment of unknown sequences using similarity searches. Ortholog groups in these databases have been identified by analyzing complete genomes.

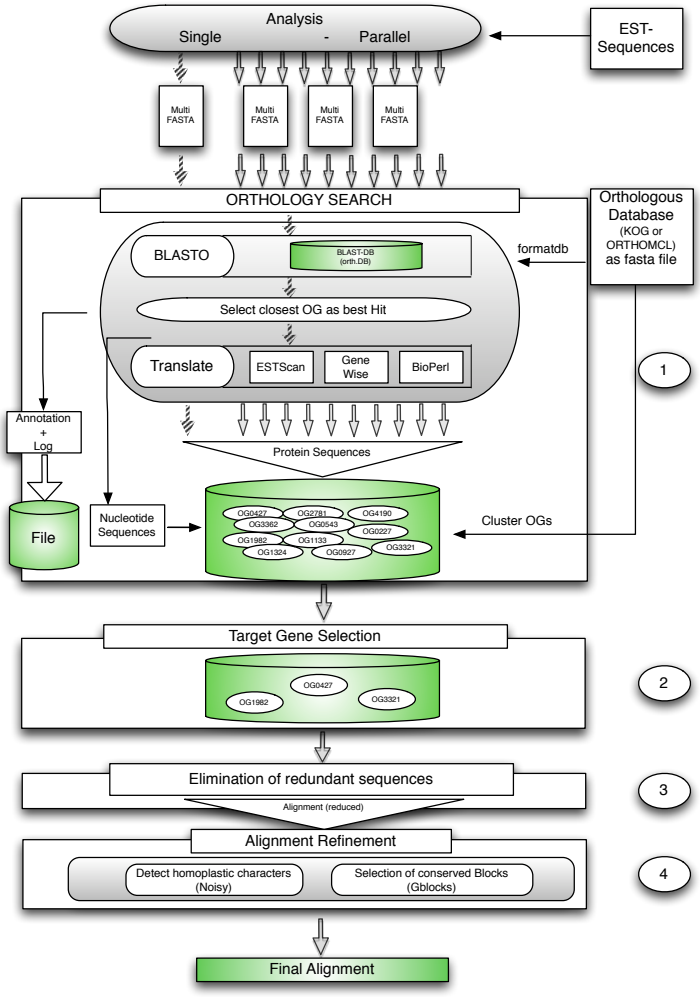
Most computational methods to identify orthologs are based either on a phylogenetic analysis or on *all-against-all* BLAST searches (Dolinski and Botstein, 2007). The former approach is computationally expensive and usually requires manual intervention. *All-against-all* approaches use every sequence from the input data set as a query for BLAST searches against the sequences from the respective other species. This approach generates OGs based on some similarity measure, e.g. using all best reciprocal hits. These OGs can be further processed to merge, delete, or seperate overlapping groups using a clustering algorithm as has been done, e.g. for OrthoMCL (Li *et al.*, 2003) or Inparanoid (O'Brien *et al.*, 2005). Zhou and Landweber (2007) implemented a computational method of ortholog prediction by including information from an ortholog database.

Other important aspects in data set construction for phylogenetic analysis on a large scale are (1) correct identification of open reading frames in ESTs and their translation, (2) careful selection of target genes to maximize the phylogenetic information, (3) elimination of redundant sequences, and (4) a final refinement step to select conserved blocks and remove homoplasy from multiple sequence alignments.

Nowadays, data sets in phylogenomic studies can easily contain dozens of taxa and hundreds of genes (Dunn *et al.*, 2008). The construction of data sets of that size for phylogenomic studies is time-consuming and can hardly be done manually.

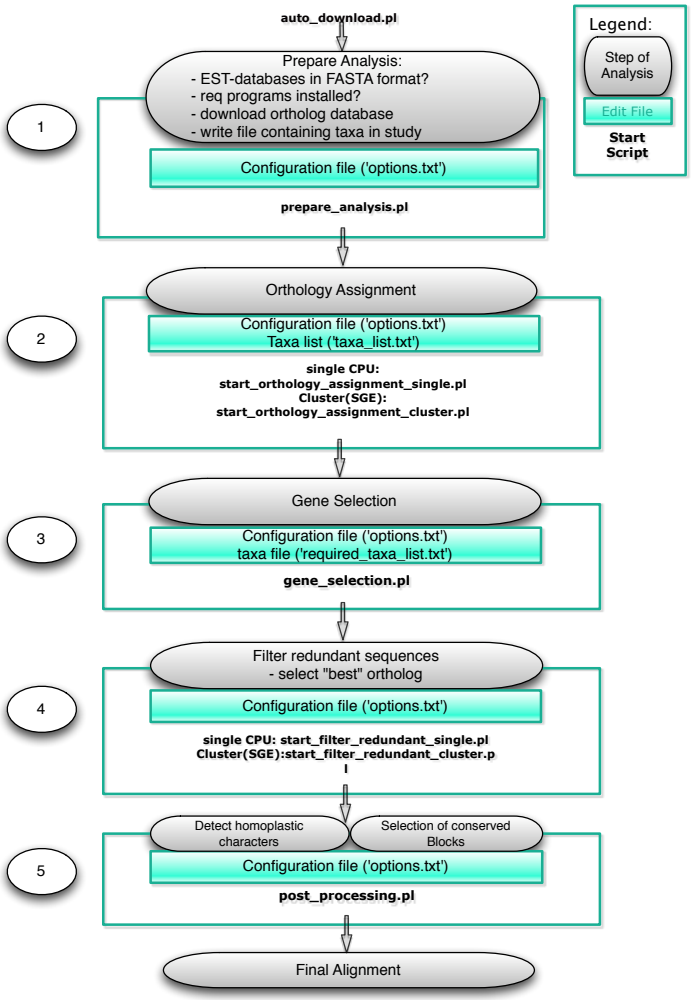
2 Program Overview

The main workflow of the software pipeline to detect ortholog sequences in phylogenomic studies. Input are EST libraries and an ortholog database (either KOG or OrthoMCL-DB) as multi-fasta files. The analysis comprises four parts. (1) The orthology detection - which can be performed on a single computer or a computer cluster using a batch system (e.g. Sun Grid Engine)- blasts each EST against the ortholog database, selects the closest ortholog group as the best hit and translates it and stored together with the nucleotide sequences in the corresponding OG. (2) Target genes can be selected. (3) The sequence most likely being an ortholog is selected by eliminating potential paralogs. (4) Alignments are refined to increase phylogenetic signal.



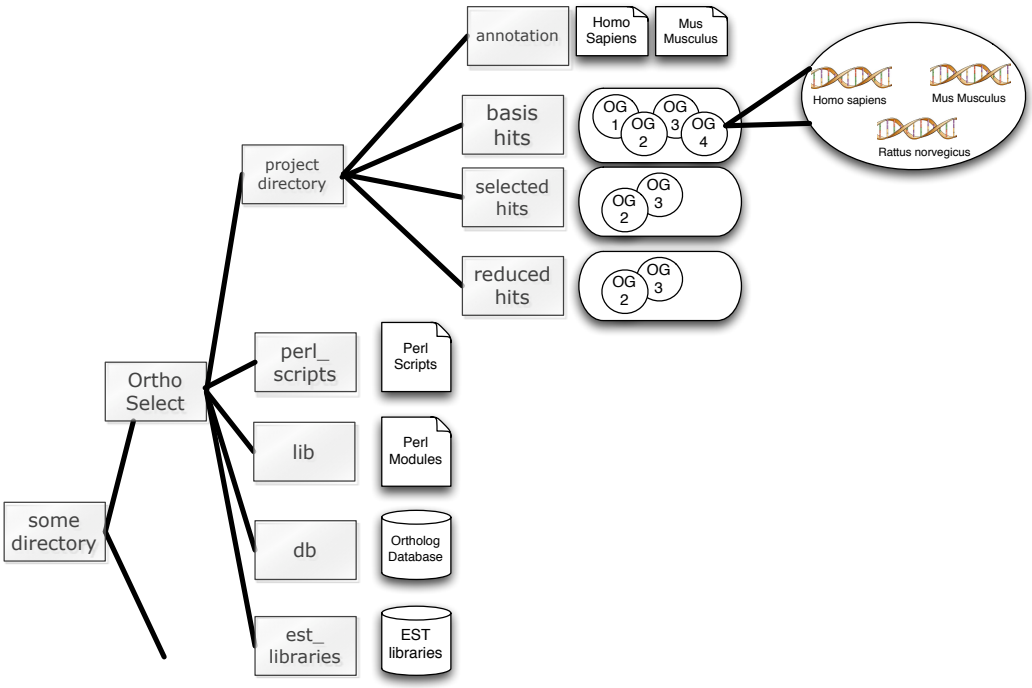
3 Script Overview

This Figure shows the workflow of the software pipeline. This time, the name of the perl script (in red) as well as the name of the file(s) to be adapted (in green) are mentioned for each step of the analysis (in gray). The first step ('auto_download.pl) is optional and will automatically download and install all required programs.



4 Data Overview

This Figure shows an overview of the directory structure that is created by the software pipeline throughout the analysis. The 'root directory' is the directory where the software pipeline is installed. The 'lib' directory contains required Perl modules, the 'db' will contain the automatically downloaded ortholog database and additional files. The EST libraries in fasta format will be stored in a directory (in this case 'test data') and the 'test data' directory contains the EST libraries in fasta format. All results will be stored in the 'project directory'. Results from the orthology assignment will be stored in 'basis hits' and annotations in 'annotation'. Further results, such as the selected target genes (in this case 'selected genes'), the genes where redundant sequences have been eliminated and the post-processed (in this case 'reduced genes'), will be stored in different folders.



5 How to use this manual

The main purpose of this manual is to make the user familiar with OrthoSelect and provide the user with every information to use OrthoSelect. For this, we splitted this manual into two parts. The first part in section 6 is a long version of how to use OrthoSelect. It includes detailed information for each step of the analysis. Taxon names used in that section are used as examples only. They should be replaced by the real taxa names in the study. Contrary to that, section 8 is a short and quick guide to start the analysis. This part can also be regarded as a tutorial.

6 Long Version - Preliminary Work

Contrary to the previous section, this section gives a more theoretical and detailed overview of all available options and parameters of OrthoSelect. Taxa names in this section are used as examples only. They should be replaced by the real taxa names in the study.

6.1 Preliminary steps

The section describes some preliminary steps to prepare your system and the data for the analysis. It covers the download and installation of external programs as well as the selection of an ortholog database and a final check if everything is correctly set up.

6.2 Required external Programs

The following external programs need to be installed on your system. Download and install them. Make sure that all programs are accessible from the command line[1]. The tested versions of the program are in brackets.

- BioPerl (Version 1.5.1)
- BLAST (Version 2.2.18)
- ESTScan (Version 2-2.1)
- Wise (Version 2.2.0)
- Clustalw (Version 2.0.8)
- T-Coffee (Version 5.72) or Muscle (Version 3.7)
- Gblocks (Version 0.91b)
- Noisy (Version 1.5.7)[2]

6.3 Automatical download of required programs

Alternatively, you can use a script that automatically and installs all required programs. The script has been tested with Macosx 10.5.2, Ubuntu (32-bit), and the bash shell (which is the default shell).

[1] You can check this manually by entering the program name in the command line. If the program starts, it is installed and accessible, if not, then an error message with inform you

[2] Due to difficulties installing this software we do not offer to automatically install these software. Nevertheless, Noisy - manually installed - can be used by OrthoSelect.

6.3.1 Start the automated download

You can now start the script that automatically downloads and installs the missing required programs by typing[3]

```
perl perl_scripts/auto_download.pl -o OS -p t
```

where OS can be replaced by 'macosx' or 'linux' if you want to install the programs on Macosx or Linux. Set the parameter "-p t", if you also wish to install bioperl. You can check whether you have bioperl installed or not by typing:

```
perl -e 'use Bio::Perl'
```

If this command results in an error message, then bioperl is not yet installed on your system. The script downloads and installs all required programs in the folder "programs". If you experience any problems, try to install the program by using 'fink'[4] or 'darwinports'[5] or ask your local administrator.

All programs should be now installed on your computer. In order for OrthoSelect to access all required programs, the profile file needs to be re-read by the shell. To do this, simply type:

```
source ~/.profile
```

6.4 User Input - EST libraries

6.4.1 Where to put the EST libraries

The EST libraries should be in fasta format. Copy all EST libraries you want to analyse in the subfolder "est_libraries" of the folder where OrthoSelect has been installed (see Fig. 4).

6.4.2 Adapting the fasta header of the EST libraries

To guarantee a smoothly flow of the analysis the fasta file are required to meet the following two criteria:

Fasta header It is important that each EST sequence in a fasta file is distinguishable by a unique identifier (e.g. an accession number). Fasta headers are required to match the following format.

```
>Accession_number | [Taxon]
```

where the Taxon name is optional. E.g.

```
>id203335 | Aspergillus_niger
```

[3] All perl scripts will be in the directory "perl_scripts" and need to be entered as mentioned in the example

[4] `http://www.finkproject.org/`

[5] `http://www.macports.org/`

or

```
>id203335|Aspergillus_niger
```

or

```
>id203335|
```

would be a correct format.
This format will be important when the program needs to distinguish between sequences from the same species (see Section 6.4.2). The script in Section 6.4.2 will try to do this in an automatic way for you[6].

Name of fasta files Fasta files should be named according to the taxon they belong to. E.g. ESTs from *Aspergillus niger* should be saved in a file

```
Aspergillus_niger.fa
```

Naming of files is important, since OrthoSelect will automatically generate a file containing all taxa in the study (See Section 6.7). Fasta files can automatically adapted using the following script.

Fasta Script The following script works for sequence data downloaded from either the JGI website[7] or TBestDB[8]. Assuming that a fasta file containing transcripts is downloaded from JGI and is stored in the appropriate directory, that is 'est_libraries', type the following to change the fasta headers of the file 'ests_from_aspergillus_niger.fa' and name the file 'Aspergillus_niger.fa':

```
perl perl_scripts/fasta_header_converter.pl -i est_libraries/ests_from_apergillus_niger.fa
 -t Aspergillus_niger -s t
```

For sequences downloaded from TBestDB, please type "-s d", for EST sequences downloaded from JGI, please type "-s e" and for sequences downloaded from dbEST, please type "-s n"[9].

Fasta Script - batch mode Adapting the fasta header of several est libraries can also be done in batch mode. For this, simply enter the following information in a file and save it under "taxa_conversion.txt". This file should include the name of the est libraries, the name of the taxons and the source of the est libraries as described above. For an example see Appendix D

The conversion can then be started typing:

```
perl perl_scripts/fasta_header_converter.pl
```

Alternatively to using the script, you can use a stream-editor such as *sed*[10] or perl's one-liners[11].

[6] Note that since the fasta header does not require a clear syntax, there is no guarantee that the script will work perfectly.

[7] see `http://genome.jgi-psf.org/euk_home.html`

[8] see `http://amoebidia.bcm.umontreal.ca/pepdb/searches/login.php`

[9] For more information see Appendix C

[10] visit e.g. `http://www.student.northpark.edu/pemente/sed/sed1line.txt`

[11] visit e.g. `http://sial.org/howto/perl/one-liner/`

6.5 The configuration file

The configuration file contains all important parameters and settings. This is the main file that has to be adapted by the user. All perl scripts use the information contained in the configuration file to perform the BLAST searches, calls of external programs, input-/output actions, etc.
By default the name of the configuration file is 'options.txt' (See Appendix F for an example of a full configuration file.)
The file consists of simple key-value pairs[12].
E.g.:

```
project_name = "Example"
```

This allows the user to set a project name.
Note that every value entered by the user needs to be within quotes.
E.g.:

```
project_name = "Example"
```

would be correct, but

```
project_name = Example
```

not.
The paths for files or folder need to end with an "/". E.g.:

```
fasta_directory = "/user/home/pipeline/fasta_files/"
```

would be correct, but

```
fasta_directory = "/user/home/pipeline/fasta_files"
```

not.

6.6 Orthologous Databases

Multi-species ortholog databases have been developed on the basis of whole genome comparisons, synteny, and phylogenetic trees to include ortholog information. Two of these databases explicitly define ortholog groups (OrthoMCL-DB and NCBIs KOG) which can be used as a basis for orthology assignment of unknown sequences using similarity searches.

To select an ortholog database, simply edit the following line in the configuration file to choose KOG as the ortholog database:

```
orthology_database_type = "k"
```

To select the OrthoMCL-DB, enter "o".

The download and configuration of the ortholog database will be done automatically (see Section 6.7).

[12]Lines starting with a hash key are comments

6.7 Final Check

So far, a lot of preliminary work has been done to prepare OrthoSelect on the system. The following script will perfom the following tasks to make sure everything is set up for the main analysis (see Section 7) to start.

- Check if the EST multi-fasta files are in correct fasta format
- Check if all required programs are accessible and correctly installed
- Download the ortholog database specified by the user
- Turn the ortholog database in a blastable database
- Test the ortholog database by performing a test BLAST search
- Write a file containing all taxa in the study ('taxa_list.txt')

Options You need to tell the script the location, where you installed OrthoSelect. You can also give your analysis a project name. To do this, set the following options in the configuration file ('options.txt' by default).

```
project_name = "Example"
root_directory = "/user/home/OrthoSelect/"
```

The 'root_directory' is the directory where OrthoSelect is installed. The project folder will be created in the 'root_directory' as can be seen in Figure 4. All results and data will be saved in the project folder. The ortholog database will be installed in the 'db'-directory[13] (see Fig. 4)

Perform Analysis The script can be called as follows:

```
perl perl_scripts/prepare_analysis.pl
```

Taxa file An overview of the taxa present in the study will be written in the file 'taxa_list.txt'. The file will also contain a recommended shortcut for each taxon name. This is because some alignment viewer or formats (e.g. the phylip format) can restrict the length of taxa names to 10 characters. The shortcuts will be later used in the fasta headers. Make sure that each shortcut is a unique identifier of a taxon. E.g.: If you have the two blood-flukes *Schistosoma mansoni* and *Schistosoma malayensis* present in your study. The recommended shortcuts will be

```
"Schistosoma_mansoni" "Schisto_ma"
"Schistosoma_malayensis" "Schisto_ma"
```

[13]Note that it can take some time to download the ortholog databases depending on the connection speed and the size of the database (KOG = 50 Mb, OrthoMCL-DB-DB = 400 Mb)

Since the program will only deal with the shortcut form, you have to change the appropriate entry to e.g.

```
"Schistosoma_mansoni" "Schisto_ma"
"Schistosoma_malayensis" "Schisto_my"
```

to avoid the program treating both taxa as the same.
On the other hand, if you have several data source for one taxon (e.g. transcripts and est-sequences) then choosing the same shortcut for each data source will let the program select the best sequence from all data sources.
Note: If you want to analyze protein sequence rather than EST sequences, you simply have to add a "p" after each taxon. Edit the taxa file as follows:

```
"Schistosoma_mansoni" "Schisto_ma" p
"Schistosoma_malayensis" "Schisto_my" p
```

With this, the program assignes protein sequences to OGs using blastp instead of blastx as with EST sequences.

7 The Main Analysis

7.1 Orthology Assignment

The first step of the software pipeline comprises the detection of potential orthologs in EST libraries (see Figure 1)

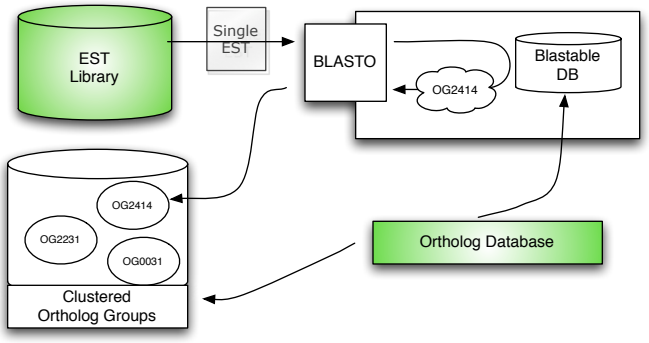


Figure 1: Workflow of orthology detection in detail. The two databases colored in green are user supplied. The ortholog database will be converted in a BLAST database as well as clustered in ortholog groups. Each EST sequences (upper left) from the EST library is assigned to that OG (lower left) returned by a BLASTO search against the ortholog database (upper right).

Options Set the following options in the configuration file ('options.txt' by default)

```
taxa_list = "taxa_list.txt"
e_value = "1e-10"
minimal_length_of_hit = "10"
no_threads = "2000"
```

Simply enter the name of the file containing the taxa whose EST libraries you want to analyse[14] (by default, all taxa will be analysed) and the expectation value (E-value). 'no_threads' is only available with an analysis using a computer cluster. It will do the following: E.g. if a EST library contains 40,000 sequences, the analysis will be split into 4 parts of 10,000 ('no_threads') sequences each and will be parallel analysed to increase the speed of the orthology search. All hits with less than 10 AA positions will be discarded.

Perform Analysis Two options are available.

For the analysis on a single computer (e.g. Desktop) type:

[14]The file 'taxa_list.txt' will be automatically created in Section 6.7

```
perl perl_scripts/start_orthology_assignment_single.pl
```

For the analysis on a computer cluster with a sun grid engine type:

```
perl perl_scripts/start_orthology_assignment_cluster.pl
```

Output By default, the results will be stored in the subfolder 'basis_hits' of the project folder (see Fig. 4 for an overview). Each OG contains the est sequence (files with "_nucl") and its translation (files with "_prot") with the name of the file corresponding to the taxon the sequence belong to. The annotations can be found in the subfolder 'annotations' of the project folder (see Fig. 4 for an overview). Annotation files contain the following information: Identifier in fasta file, Taxon name, Assigned OG, one-letter functional annotation, Annotation, E-value of best hit, Identifier of best hit of assigned OG, Method used for translation, E-value for translation with GeneWise, standard 6-frame-translation, and ESTScan by comparing the translated sequence with the best database hit using *bl2seq* E.g.

```
Aspergi_ni|ACC69732    Aspergillus_niger   KOG4663 C
Cytochrome b    3e-75   HsMi013 GeneWise        [7e-83] 2e-82   3e-27
```

Here, the EST sequence with the accession number ACC69732 from the organism *Aspergillus_niger* was assigned to the OG KOG4663, a cytochrome b, with an E-value of 3e-75. The best hit was with the sequence HsMi013 and the sequence was translated using GeneWise, since it produced the most significant translation (E-value 7e-83). The table containing the one-letter functional classification used in the KOG database can either be found in file "fun.txt" in the database-directory (see Fig. 4 for an overview) or in the appendix E.

7.2 Statistics

View results from orthology search As mentioned before, the OGs with the est sequences assigned to it are in the subdirectory 'basis_hits' of the project-directory (see Fig. 4 for an overview).

Create statistic file After the orthology assignment, the results folder will contain a lot of hits. Given a list of taxa in the study (e.g. taxa_list.txt) one want to know the distribution of hits for the different taxa and OGs. A presence/absence tab-delimited text file[15] will be created for all taxa from 'taxa_list.txt' and OGs in the study. The statistic can also be generated to a later point of the analysis (For this, you have to change the statistic_directory in the configuration file). This text file can then be easily imported into spreadsheet applications (e.g. Excel).

[15] presence of a species is coded as "1", absence as "0"

Options Set the following options in the configuration file ('options.txt' by default)

```
statistic_directory = "basis_hits"
```

The 'statistic_directory' is the directory containing the OGs to be analysed.

Perform Analysis Start the analysis by typing:

```
perl perl_scripts/stats.pl > text.file
```

where 'text.file' will then contain the statistics. 'text.file' can be replaced by any name to avoid to overwrite existing statistical files.

Output By default, the statistics will be stored in the file 'text.file'.

7.3 Gene Selection - Optional

With assembled EST sequences assigned to predefined ortholog groups (OG) and translated into proteins, the next step is the proper selection of OGs suitable for phylogenetic analysis. Since EST libraries represent snapshots of expressed genes, not every OG will contain EST sequences from all species under study; some OGs may contain only a few sequences. Such OGs do not contain sufficient information and are therefore not suitable for further consideration. On the other hand, we do not require every OG to contain all sequences of interest. So far, there is no consensus about the influence of missing genes on the resulting phylogeny exists (Wiens (2006)), so there is no reliable criterion which OGs should be used for phylogenetic inference. Our software offers the following three options:

1. The user selects a group of species. In this case, those OGs will be selected that contain ESTs from all of the user-selected species
2. The user defines *groups* of species. Our tool will then select those OGs that contain at least one EST sequence for each of the specified groups.
3. The selection of OGs based on a user-defined percentage of missing data

The selection of genes according to the existence of user-selected taxa can be useful to reduce the number of OGs to a manageable set of OGs. In order to skip the step of the analysis, one can simply set go to Section 7.4 and enter 'basis_hits' as the 'distance_calculation_input'. Remember that 'basis_hits' is the directory containing all OGs after the orthology search (see Sec. 7).

Options Set the following options in the configuration file ('options.txt' by default)

```
gene_selection_option = "m"
```

The selection of "s" corresponds to strategy 1, "m" to strategy 2.

```
gene_selection_directory = "selected_genes"
required_taxa_file = "required_taxa_list.txt"
```

The directory 'gene_selection_directory' will contain only those OGs selected by one of the search strategies. The file 'required_taxa_file' will contain the taxa upon which a selection of OGs will be made.

Depending on the selection criterion used, the file should look as follows:

Strategy 1 The selection of those OGs according to the existence in at least one member of a pre-defined monophyletic group. The syntax of the file should be:

```
Name_for_monophylum = "Species1","Species2",...,"SpeciesX"
```

e.g.:

```
Tetraconata = "Drosophila_melanogaster","Daphnia_magna","Carcinus_maenas"
Mammalia = "Homo_sapiens","Mus_musculus","Rattus_norvegicus"
```

This will define the three tetraconata 'Drosophila melanogaste','Daphnia magna','Carcinus maenas' as one monophylum and the three mammalia 'Homo sapiens','Mus musculus','Rattus norvegicus' as another one. OGs will be selected for which at least one out of the three species is present.

Strategy 2 Selection of those OGs according to the existence in at least one sequence for each selected species. You can simply copy the file containing all taxa in the study ('taxa_list.txt') and remove all unwanted. The file could look than as follows:

```
"Aspergillus_niger" "Aspergi_ni"
"Ciona_intestinalis" "Ciona_in"
"Lottia_gigantea" "Lottia_gi"
```

Perform Analysis Start the analysis by typing:

```
perl perl_scripts/gene_selection.pl
```

Output By default, the results will be stored in the subfolder 'selected_hits' of the project folder.

7.4 Eliminating Redundancies

Multiple divergent copies of the same gene and different levels of stringency during EST assembly can lead to the situation where OGs contain more than one sequence per species. The same is true for the ortholog groups contained in KOG, where many groups contain both orthologs and paralogs (Dessimoz *et al.*, 2006). In these cases, a fast and reliable method is needed to select the best sequence per species. Assuming that orthologs between organisms are more similar to each other than to paralogs, all sequences belonging to the same OG are aligned and two types of distance matrices can be used to decide which sequence is to be kept for further analysis. These two matrix types are:

1. An initial distance matrix as computed by alignment methods like Clustal W (Chenna *et al.*, 2003)
2. A specialised distance matrix selecting those sequences that have the highest number of matching positions in pairwise comparisons using Muscle (Edgar, 2004) .

Options Set the following options in the configuration file ('options.txt' by default)

```
distance_calculation_input = "selected_genes"
distance_calculation_output = "reduced_genes"
```

for the input and output directories for this step of the analysis.

```
alignment_method = "m"
```

Selection of the alignment method used to align the sequences in an ortholog group. Use "m" to align the sequences using Muscle and "t" to align the sequences using T-coffee

```
distance_matrix_type = "g"
```

For matrix type 1 select 'a' and for matrix type 2 select 'g'.

Other Alignment Methods The user can use any alignment method available by providing the command line call of the alignment method. E.g. if the user wants to use the alignment method "align", then he needs to provide the system call of the program "align" and replace the input and output file name by $fasta_file_reduced and $final_alignment_file respectively. The appropriate value in the configuration file should look then as follows:

```
different_alignment_method = "align -in $fasta_file_reduced -out $final_alignment_file"
```

Perform Analysis Again, two options are available.

For the analysis on a single computer (e.g. Desktop) type [16]:

```
perl perl_scripts/start_filter_redundant_single.pl
```

For the analysis on a computer cluster with a sun grid engine type:

```
perl perl_scripts/start_filter_redundant_cluster.pl
```

Output By default, the results will be stored in the subfolder 'reduced_genes' of the project folder. The alignments will have the postfix 'final.fasta', unaligned protein the postfix 'prot_hits.fasta' and unaligned nucleotide sequences the postfix 'nucl_hits.fasta'.

7.5 Alignment Curation

The final part comprises the use of different algorithms to refine the alignment and improve the accuracy of the following phylogenetic reconstruction. Since not all parts of a gene evolve at the same rate, alignments can be composed of highly conserved and less conserved regions. Useful regions for phylogenetic analysis are those that are conserved to a certain degree, because either regions full of the identical characters or regions too divergent to be correctly aligned do not contain useful phylogenetic signal.

Note that the user is encouraged to manually check all alignments. Since OrthoSelect selects that sequence from an organism most likely being an ortholog in Section 7.4, it can select paralogs in case orthologs are missing in the data or organism (due to gene loss). In case the alignment contains sequences that obviously do not fit in the alignment, the user is encouraged to check the annotations as well as the nucleotide sequence of that sequences.

The following two programs try to select parts of the alignment suitable for phylogenetic analysis (Gblocks) and to eliminate potentially homoplastic sites (Noisy).

7.5.1 Gblocks

Gblocks (Castresana, 2000) is a tool that automatically select conserved blocks from multiple sequences for their use in phylogenetic analysis.
For more information about the programm see the user manual of Gblocks.

Options Set the following options in the configuration file ('options.txt' by default)

```
post_method = "g"
```

The post-processing method: 'g' stands the post-processing of alignments using Gblocks, 'n' stands for post-processing using 'Noisy'.

[16] Note that this step can take a while depending on the size of the OGs.

```
post_process_directory = "reduced_genes"
```

This is the directory where the post-processed files will be saved. The Gblocks parameters can be adjusted by changing the following values in the configuration file.

```
##### GBLOCKS ###############
gblocks_b1 = ""
gblocks_b2 = ""
gblocks_b3 = ""
gblocks_b4 = ""
gblocks_b5 = ""
```

By default (Parameters left blank) OrthoSelect uses Gblocks with standard settings[17].

Filter out sequences with a user-defined percentage of missing characters Based on the alignment processed using Gblocks, the user can select to filter out sequences with a percentage of characters below a user-defined threshold. To activate this option the following parameter have to be set in the configuration file:

```
post_alignment_filter = "t"
post_alignment_threshold = "50"
```

where 'post_alignment_filter = "t" ' turns the filter on (default is post_alignment_filter = "t" and the filtering option turned off) and 'post_alignment_threshold = "50"' means that sequences with less than 50% of characters in the alignment are discarded.

Perform Analysis Start the analysis by typing:

```
perl perl_scripts/post_processing.pl
```

7.5.2 Noisy

Noisy[18] is a program that tries to eliminate potentially homoplastic sites in multiple sequence alignments.
For more information about the programm see (Dress *et al.*, 2008)

Options Set the following options in the configuration file ('options.txt' by default)

```
post_method = "n"
```

[17] for a detailed description see `molevol.ibmb.csic.es/Gblocks.html`.

[18] Due to difficulties installing this software we do not offer to automatically install these software. Nevertheless, Noisy can be used by OrthoSelect if it is manually installed.

The post-processing method: 'g' stands the post-processing of alignments using Gblocks, 'n' stands for post-processing using 'Noisy'.

```
post_process_directory = "reduced_genes"
```

This is the directory where the post-processed files will be saved.

Perform Analysis Start the analysis by typing:

```
perl perl_scripts/post_processing.pl
```

Output By default, the results will be stored in the subfolder 'reduced_hits' of the project folder. The geblocked alignments will have the postfix 'gblocked.fasta', the alignments processed with noisy will have the postfix 'out.fas'. Nosiy additionally creates the files '-sta.gr' and '-typ.eps'. Geblocked alignments with sequences filtered out have the postfix 'gblocked_filtered.fasta'. The name and the percentage of character content of the sequences which have been filtered out are in the file 'LOG_rejected_sequences.txt'.

8 Short Version - An Example Analysis

This section is a quick guide to start the analysis. For this purpose, a test dataset will be included in the OrthoSelect download. The dataset consists of orginal sequences downloaded from JGI. The original dataset has been reduced to include only a few sequences from each taxon that will be assigned to the same ortholog groups. For details and background, please see the long version in section 6.

8.1 Input Data

8.1.1 Install external programs

After extracting OrthoSelect, all missing required external programs as well as BioPerl will be downloaded and installed under linux[19] typing

```
perl perl_scripts/auto_download.pl -o linux -p t
```

We re-read the shell profile so that OrthoSelect can find all required programs and variables by typing:

```
source ~/.profile
```

8.1.2 Fasta files

Our test data set consists of EST sequences and transcripts (shown in the table below) downloaded from JGI[20].

Taxon	File name	Type of sequence
Aspergillus niger	Aspni1_FilteredModels1.na.fasta	Transcripts
Ciona intestinalis	Cintestinals_EST_clusters.fasta	ESTs
Daphnia pulex	FrozenGeneCatalog_2007_07_03.na.fasta	Transcripts
Lottia gigantea	Lotgi1_EstClusters_na....fasta	ESTs
Monosiga brevicollis	Monbr1_ESTclusters.fasta	ESTs

The files are in the folder "test_data". Currently, the fasta headers of these files look like this:

```
File: test_data/Aspni1_FilteredModels1.na.fasta
Header: >jgi|Aspni|202811|estExt_fgenesh1_pg.C_20516
File: test_data/Cintestinals_EST_clusters.fasta
Header: >847937:2
File: test_data/FrozenGeneCatalog_2007_07_03.na.fasta
Header: >jgi|Dappu1|48454|gw1.58.76.1
```

[19] see section 6.3.1 for details about other operating systems

[20] The full file name of *Lottia gigantea* is Lotgi1_EstClusters_naClusterCr705LottiaJgi20060727.fasta, but has been reduced to save space.

```
File: test_data/Lotgi1_EstClusters_naClusterCr705LottiaJgi20060727.fasta
Header: >4243977:1
File: test_data/Monbr1_ESTclusters.fasta
Header: >3716647:1
```

Since a correct fasta header is a precondition for a successful analysis, we make sure that the fasta headers are all in correct format. To automatically rename the fasta headers, we write the following information in a file called "taxa_conversion.txt":

```
"test_data/Aspni1_FilteredModels1.na.fasta"="Aspergillus_niger"="t"
"test_data/Cintestinals_EST_clusters.fasta"="Ciona_intestinalis"="e"
"test_data/FrozenGeneCatalog_2007_07_03.na.fasta"="Daphnia_pulex"="t"
"test_data/Lotgi1_EstClusters_naClusterCr705LottiaJgi20060727.fasta"="Lottia_gigantea"="e"
"test_data/Monbr1_ESTclusters.fasta"="Monosiga_brevicollis"="e"
```

We convert the fasta headers in the correct format by typing:

```
perl perl_scripts/fasta_header_converter.pl
```

The sequences with correct fasta headers will now be in the diretory "est_libraries", that is the directory OrthoSelect looks for EST libraries. The correct fasta headers will then look like this:

```
File: est_libraries/Aspergillus_niger.fa
Header: >202811|Aspergillus_niger
File: est_libraries/Ciona_intestinalis.fa
Header: >847937_2|Ciona_intestinalis
File: est_libraries/Daphnia_pulex.fa
Header: >48454|Daphnia_pulex
File: est_libraries/Lottia_gigantea.fa
Header: >4243977_1|Lottia_gigantea
File: est_libraries/Monosiga_brevicollis.fa
Header: >3716647_1|Monosiga_brevicollis
```

8.2 Prepare Analysis

8.2.1 Configuration File

Now we have to tell the program the location where we installed OrthoSelect. Furthermore, we tell the program the name of our project. All results will be saved in that project folder. We adapt the configuration file as follows.

```
project_name = "TEST_ANALYSIS"
root_directory = "/home/user/OrthoSelect/"
```

The path "/home/user/orthoselect/" needs to be replaced by the path OrthoSelect is installed in. This path will be known as *root_path* from now on.

8.2.2 Selection of ortholog database

We decide to use the KOG database. Therefore, we mark the corresponding option in the configuration file

```
orthology_database_type = "k"
```

8.2.3 Final preparation

To automatically download, format and test the ortholog database as well as to perform some final tests, we call the perl script "prepare_analysis.pl" as follows:

```
perl perl_scripts/prepare_analysis.pl
```

8.3 Orthology Assignment

Now we want to assign orthology to our EST sequences in folder

```
root_path/est_libraries/
```

Using default settings in the configuration file, we start the analysis on our single computer by typing:

```
perl perl_scripts/start_orthology_assignment_single.pl
```

After the analysis has finished, the project folder ("*root_path*/TEST_ANALYSIS/") will contain the following directories:

```
annotations basis_hits
```

The results have been saved to the folder

```
root_path/TEST_ANALYSIS/basis_hits/
```

and contain the following OGs:

```
KOG0003  KOG0019  KOG0020  KOG0027  KOG0179  KOG0213
```

The EST sequence as well as its translation will be saved in the corresponding OG subfolder.
E.g. an EST sequence from Daphina_pulex assigned to 'KOG0213' will be saved as

```
KOG0213_Daphnia_pulex_prot_hits.fasta
KOG0213_Daphnia_pulex_nucl_hits.fasta
```

in "*root_path*/TEST_ANALYSIS/basis_hits/KOG0213/". Annotations are saved in

```
root_path/TEST_ANALYSIS/annotations/
```

This folder will now contain the annotation for all our taxa in the study. The folder comprises:

```
Aspergillus_niger.txt
Ciona_intestinalis.txt
Daphnia_pulex.txt
Lottia_gigantea.txt
Monosiga_brevicollis.txt
```

8.4 Gene Selection

Now we may want to reduce the dataset to include only OGs with our species of interest present. The selection of OGs will be according to the existence of at least one sequence for each of the following species: *Aspergillus niger* and *Daphnia pulex*.
We simply copy the file containing all taxa in the study ('taxa_list.txt' by default), delete all unwanted lines and save the file under 'required_taxa_list.txt'. The file then looks like this:

```
"Aspergillus_niger" "Aspergi_ni"
"Daphnia_pulex" "Daphnia_pu"
```

Since we are using default settings, we do not need to change the configuration file. We start this part of the analysis by typing:

```
perl perl_scripts/gene_selection.pl
```

The folder "*root_path*/TEST_ANALYSIS/selected_genes" will contain then the following OGs:

```
KOG0019  KOG0020  KOG0027  KOG0179  KOG0213
```

8.5 Eliminating Redundancies

So far, our OGs contain homologous sequences, that are orthologs and paralogs, but we want to keep the orthologs only. The selected target genes are in the directory 'selected_genes' and we want the results of this step of the analysis to be saved in the new folder 'reduced_genes'. The distance matrix that is used to select the sequences most likely to be an ortholog is calculated from the alignment. The sequence will be aligned using muscle. So make sure that the configuration file has the following entries.

```
distance_calculation_input = "selected_genes"
distance_calculation_output = "reduced_genes"
alignment_method = "m"
distance_matrix_type = "g"
```

We start the analysis by typing:

```
perl perl_scripts/start_filter_redundant_single.pl
```

After the analysis has finished, the ortholog group e.g. KOG0213 will then contain the following files:

```
KOG0213_final.fasta
KOG0213_nucl_hits.fasta
KOG0213_prot_hits.fasta
```

with the unaligned protein and nucleotid sequences in "KOG0213_prot_hits.fasta" and "KOG0213_nucl_hits.fasta", respectively. The alignment containing only the best sequence for each taxon is "KOG0213_final.fasta".

8.6 Alignment Curation

The final step is the automatic curation of alignments with the goal to select potential conserved region or remove homoplastic sites. In this case, we want to select potential conserved regions of our alignments using 'Gblocks'. We will use Gblocks with default settings here and our alignments are in folder 'reduced_genes'. We also want to get rid of too short sequences. Therefore we turn the post_alignment_filter on in the configuration file and set the threshold to 50% to allow only sequences that have at least half the length of all other sequences. We make sure that the configuration file has the following entries.

```
post_method = "g"
post_process_directory = "reduced_genes"
post_alignment_filter = "t"
post_alignment_threshold = "50"
```

We start the analysis by typing:

```
perl perl_scripts/post_processing.pl
```

The blocked alignments as well as the filtered alignments will be put in the same folder. So, the folder containing the OG KOG0213 will contain the following files:

```
KOG0213_final.fasta
KOG0213_final_gblocked.fasta
KOG0213_final_gblocked_filtered.fasta
KOG0213_nucl_hits.fasta
KOG0213_prot_hits.fasta
LOG_rejected_sequences.txt
```

You can now view the blocked alignment *KOG0213_final_gblocked.fasta* and compare it to the filtered alignment *KOG0213_final_gblocked_filtered.fasta* to see how the filter option works.

Conclusion This was a short example to show how easy OrthoSelect can be used. Since paralogs can be selected in case orthologs are missing - either in the study or in the species - all temporary results and especially the final alignments should be checked with much care. At this point, the user should be more or less familiar with the way OrthoSelect works. The user can now read the next chapter to perform the analysis using real data.

9 Troubleshooting

9.1 Error Messages

Error

```
No CDS matrix found for 29.6666666666667 \% GC.
at sw/estscan/BTLib-2.0b/ESTScan/ESTScan line 160, line 6.
```

Problem ESTScan cannot find a CDS matrix. The environmental variable *$ESTSCANDIR* is not properly set. A reason could be that a previous ESTScan installation has been deleted (without deleting the corresponing environmental variable). The variable has to point to the ESTScan directory containing the matrix file ('Hs.smat' by default).

Solution Type

```
echo "export ESTSCANDIR=$dir" >> ~/.profile;
```

where $dir is the installation directory of ESTScan. By default, this is a subdirectory in the OrthoSelect directory.

Error

```
Fatal Error
Could not build objects!
```

Problem GENEWISE cannot find several matrices for translation. The environmental variable *$WISECONFIGDIR* is not properly set. The variable has to point to the WISECONFIGDIR directory containing several matrix files (e.g. 'blosum62.bla').

Solution Type

```
echo "export WISECONFIGDIR=$dir"/wisecfg/ >> ~/.profile;
```

where $dir is the installation directory of WISECONFIGDIR. By default, this is a subdirectory in the OrthoSelect directory.

Error

```
I used the perl script 'auto_download.pl' to download
all required programs, but they do not seem to be installed
```

Problem It is possible that your shell does not know yet where to search for the installed programs (This information has been added to your .profile file during the execution of the perl script). Your shell needs to re-read the .profile file.

Solution Type

```
source $HOME/.profie
```

where $HOME is your home-directory.

Problem MSG: cannot find path to blastall

Solution OrthoSelect cannot find blastall. Either you havent installed blastall or you are using a different shell and this shell does not know where to find blastall. Either type

```
source $HOME/.profie
```

where $HOME is your home-directory to re-read the .profile file or

```
perl auto_download.pl
```

to install the missing blastall package.

Problem Can't locate LWP/Simple.pm in @INC

Solution Your system cannot find the Perl module Simple.pm from the package LWP.
Install it by typing

```
perl -MCPAN -e 'install LWP::Simple'
```

A Format 'required_taxa_list.txt'

Selection strategy based on defined monophyla:

```
"MonophylumX" = "Species_name", "Species_name"
```

```
Monophylum1 = "Monosiga_ovata","Monosiga_brevicollis"
Monophylum2 = "Homo_sapiens","Mus_musculus"
```

B Format 'taxa_list.txt'

```
"Species_name" "Species_name_shortcut"
```

e.g.

```
"Acropora_millepora" "Acropor_mi"
"Allomyces_macrogynus" "Allomyc_ma"
"Amphimedon_queenslandica" "Amphime_qu"
```

C Fasta Conversions

Possible conversions of fasta headers using the script "fasta_header_converter.pl"

C.1 dbEST (Option "-s n")

dbEST Format

```
>gi|166077299|gb|FD528199.1|FD528199 RUS94C02w HZ Hordeum vulgare subsp. vulgare cDNA clone
```

will be converted to

```
>166077299|Taxon_name
```

C.2 TBestDB (Option "-s d")

TBestDB Format

```
>Cluster Id : ACL00003079 AutoFACT Annotation : 14-3-3-like regulatory protein
```

will be converted to

```
>ACL00003079|Taxon_name
```

C.3 JGI - ESTs (Option "-s e")

JGI EST Format

```
>3666157:1
```

will be converted to

```
>3666157_1|Taxon_name
```

C.4 JGI - transcripts (Option "-s t")

JGI Transcript Format

```
>jgi|Triad1|53994|fgeneshTA2_pg.C_scaffold_3000001
```

will be converted to

```
>53994|Taxon_name
```

D Fasta Conversions - Batch mode

Example file for converting fasta headers in the correct format. The syntax is:

```
"FOLDER/NAME_OF_SEQUENCE_FILE"="TAXON_NAME"="DATABASE_SOURCE"
```

An example file could be:

```
"downloaded_sequences/Oxytricha_trifallax_clusters"="Oxytricha_trifallax"="d"
"downloaded_sequences/Taphrina_deformans_clusters"="Taphrina_deformans"="d"
"downloaded_sequences/Triad1_best_transcripts.fasta"="Trioplax_adhaerens"="t"
```

E Overview - One-letter functional classification used in the KOG database

```
INFORMATION STORAGE AND PROCESSING
 [J] Translation, ribosomal structure and biogenesis
 [A] RNA processing and modification
 [K] Transcription
 [L] Replication, recombination and repair
 [B] Chromatin structure and dynamics

CELLULAR PROCESSES AND SIGNALING
 [D] Cell cycle control, cell division, chromosome partitioning
 [Y] Nuclear structure
 [V] Defense mechanisms
 [T] Signal transduction mechanisms
 [M] Cell wall/membrane/envelope biogenesis
 [N] Cell motility
 [Z] Cytoskeleton
 [W] Extracellular structures
 [U] Intracellular trafficking, secretion, and vesicular transport
 [O] Posttranslational modification, protein turnover, chaperones

METABOLISM
 [C] Energy production and conversion
 [G] Carbohydrate transport and metabolism
```

```
 [E] Amino acid transport and metabolism
 [F] Nucleotide transport and metabolism
 [H] Coenzyme transport and metabolism
 [I] Lipid transport and metabolism
 [P] Inorganic ion transport and metabolism
 [Q] Secondary metabolites biosynthesis, transport and catabolism

POORLY CHARACTERIZED
 [R] General function prediction only
 [S] Function unknown
```

F Format 'options_.txt'

```
#############################################################################
# Script name  :    options.txt
#
# Date created :    August 2008
#
# Author       :    Fabian Schreiber <fschrei@gwdg.de>
#
# This is the configuration file for OrthoSelect
# See the User manual for detailed descriptions
#
# NOTE: Paths must end with "/"
#############################################################################

#####################################
####### PROJECT OPTIONS #############
#####################################
## PROJECT NAME
project_name = "TEST"
## ROOT DIRECTORY FOR ANALYSIS (absolute pathname required)
root_directory = "/Users/home/OrthoSelect/"
#####################################
#####################################

#####################################
####### ORTHOLOG DATABASE ###########
#####################################
# Database to blast against
# KOG = "k"
# OrthoMCL = "o"
#####################################
orthology_database_type = "k"
#####################################



#####################################
####### ORTHOLOGY SEARCH ############
#####################################
##### Taxa List to analyse###########
### Name of file in (root-directory)
taxa_list = "taxa_list.txt"
##### BLAST-Options #################
e_value = "1e-10"
#####################################
```

```
##### Minimum length of Hit (AA)#####
minimum_length_of_hit = "10"
#####################################
##### FOR PARTITIONED ANALYSIS ######
no_threads = "2000"
#####################################




#####################################
##### Statistics ####################
#####################################
#####################################
### LOCATION OF FOLDER CONTAINING TAXA OF INTEREST
### Name of file in (project-directory)
statistic_directory = "basis_hits"
### LOCATION OF ANNOTATION FILE
### Name of file in (root-directory)
annotation_file = "db/kog_list"
#####################################
#####################################



#####################################
##### GENE SELECTION (OPTIONAL) #####
#####################################
# Options
# Single Taxa = "s"
# Monophylum = "m"
#####################################
gene_selection_option = "m"
### LOCATION OF FOLDER CONTAINING TAXA OF INTEREST
### Name of directory in (project-directory)
gene_selection_directory = "selected_genes"
### LOCATION OF FILE CONTAINING TAXA OF INTEREST
### Name of file in (root-directory)
required_taxa_file = "required_taxa_list.txt"
#####################################
#####################################



#####################################
##### ELIMINATING REDUNDANCIES ######
```

```
#####################################
### INPUT FOLDER FOR CALCULATION
### Name of directory in (project-directory)
distance_calculation_input = "basis_hits"
### OUTPUT FOLDER FOR CALCULATION
### Name of directory in (project-directory)
distance_calculation_output = "reduced_genes"
### ALIGNMENT METHOD
# Options
# Muscle = m
# T-COFFEE = t
#####################################
alignment_method = "m"
#####################################
## OTHER ALIGNMENT METHODS
# System call of different alignment method
# Use "$fasta_file_reduced" as Input and
# "$final_alignment_file" as Output
# e.g. for program xyz:
# xyz -in $fasta_file_reduced -out $final_alignment_file
#
#####################################
different_alignment_method = ""
#####################################
### DISTANCE MATRIX TYPE
# Options
# From Alignment (matrix type 1) = "a"
# Custom Distance Matrix (matrix type 2) = "g"
#####################################
distance_matrix_type = "g"
#####################################
#####################################



#####################################
##### POST PROCESSING ###############
#####################################
# Select Method to be used
# Gblocks = "g"
# Noisy = "n"
#####################################
post_method = "g"
### INPUT FOLDER
### Name of directory in (project-directory)
post_process_directory = "reduced_genes"
```

```
##### GBLOCKS ###############
# PARAMETERS FOR GBLOCKS (leave blank for defaults)
gblocks_b1 = ""
gblocks_b2 = ""
gblocks_b3 = ""
gblocks_b4 = ""
gblocks_b5 = ""
#####################################
# Filter sequencing with less than X % character
post_alignment_filter = "t"
post_alignment_threshold = "50"
#####################################
```

References

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. *Nucl. Acids Res.*, **25**(17), 3389–3402.

Bourlat, S. J., Juliusdottir, T., Lowe, C. J., Freeman, R., Aronowicz, J., Kirschner, M., Lander, E. S., Thorndyke, M., Nakano, H., and Kohn, A. B. (2006). Deuterostome phylogeny reveals monophyletic chordates and the new phylum xenoturbellida. *Nature*, **444**(7115), 85–88.

Castresana, J. (2000). Selection of conserved blocks from multiple alignments for their use in phylogenetic analysis. *Mol Biol Evol*, **17**(4), 540–552.

Chen, F., Mackey, A. J., Stoeckert, Christian J., J., and Roos, D. S. (2006). OrthoMCL-DB: querying a comprehensive multi-species collection of ortholog groups. *Nucl. Acids Res.*, **34**(Database Issue), D363–368.

Chenna, R., Sugawara, H., Koike, T., Lopez, R., Gibson, T. J., Higgins, D. G., and Thompson, J. D. (2003). Multiple sequence alignment with the Clustal series of programs. *Nucl. Acids Res.*, **31**(13), 3497–3500.

Delsuc, F., Brinkmann, H., and Philippe, H. (2005). Phylogenomics and the reconstruction of the tree of life. *Nature Reviews Genetics*, **6**(5), 361–375.

Delsuc, F., Brinkmann, H., Chourrout, D., and Philippe, H. (2006). Tunicates and not cephalochordates are the closest living relatives of vertebrates. *Nature*, **439**(7079), 965–968.

Dessimoz, C., Boeckmann, B., Roth, A. C. J., and Gonnet, G. H. (2006). Detecting non-orthology in the COGs database and other approaches grouping orthologs using genome-specific best hits. *Nucl. Acids Res.*, **34**(11), 3309–3316.

Dolinski, K. and Botstein, D. (2007). Orthology and functional conservation in eukaryotes. *Annual Review of Genetics*, **41**, 465–507.

Dress, A., Flamm, C., Fritzsch, G., Grunewald, S., Kruspe, M., Prohaska, S., and Stadler, P. (2008). Noisy: Identification of problematic columns in multiple sequence alignments. *Algorithms for Molecular Biology*, **3**, 7.

Dunn, C. W., Hejnol, A., Matus, D. Q., Pang, K., Browne, W. E., Smith, S. A., Seaver, E., Rouse, G. W., Obst, M., and Edgecombe, G. D. (2008). Broad phylogenomic sampling improves resolution of the animal tree of life. *Nature*, **452**(7188), 745–749.

Duret, L., Mouchiroud, D., and Gouy, M. (1994). HOVERGEN: a database of homologous vertebrate genes. *Nucl. Acids Res.*, **22**(12), 2360–2365.

Edgar, R. C. (2004). MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucl. Acids Res.*, **32**(5), 1792–1797.

Eisen, J. A. (1998). Phylogenomics: Improving functional predictions for uncharacterized genes by evolutionary analysis. *Genome Res.*, **8**(3), 163–167.

Fitch, W. M. (1970). Distinguishing homologous from analogous proteins. *Syst. Zool*, **19**(2), 99–113.

Gee, H. (2003). Evolution: ending incongruence. *Nature*, **425**, 798–804.

Koonin, E. V. (2005). Orthologs, paralogs, and evolutionary genomics. *Annual Review of Genetics*, **39**, 309–338.

Li, L., Stoeckert, Christian J., J., and Roos, D. S. (2003). OrthoMCL: Identification of ortholog groups for eukaryotic genomes. *Genome Res.*, **13**(9), 2178–2189.

Mushegian, A. R., Garey, J. R., Martin, J., and Liu, L. X. (1998). Large-scale taxonomic profiling of eukaryotic model organisms: a comparison of orthologous proteins encoded by the human, fly, nematode, and yeast genomes. *Genome Res.*, **8**(6), 590–598.

O'Brien, K. P., Remm, M., and Sonnhammer, E. L. L. (2005). Inparanoid: a comprehensive database of eukaryotic orthologs. *Nucl. Acids Res.*, **33**(Database Issue), D476–480.

Tatusov, R., Fedorova, N., Jackson, J., Jacobs, A., Kiryutin, B., Koonin, E., Krylov, D., Mazumder, R., Mekhedov, S., Nikolskaya, A., Rao, B. S., Smirnov, S., Sverdlov, A., Vasudevan, S., Wolf, Y., Yin, J., and Natale, D. (2003). The COG database: an updated version includes eukaryotes. *BMC Bioinformatics*, **4**, 41.

Wiens, J. (2006). Missing data and the design of phylogenetic analyses. *Journal of Biomedical Informatics*, **39**, 34–42.

Zhang, Z., Schwartz, S., Wagner, L., and Miller, W. (2000). A greedy algorithm for aligning DNA sequences. *Journal of Computational Biology*, **7**(1-2), 203–214.

Zhou, Y. and Landweber, L. F. (2007). BLASTO: a tool for searching orthologous groups. *Nucl. Acids Res.*, **35**(Web Server Issue), W678–682.

Zmasek, C. and Eddy, S. (2002). RIO: Analyzing proteomes by automated phylogenomics using resampled inference of orthologs. *BMC Bioinformatics*, **3**, 14.