ProtTrace Manual

Arpit Jain
jain@bio.uni-frankfurt.de

August 25, 2017

1 Introduction.

protTrace is a simulation based workflow to estimate for each protein, its evolutionary traceability as a function of time. Evolutionary traceability defines for each protein the evolutionary distance up to, which a protein ortholog can be detected, based on sequence similarity, if the ortholog is conditioned to be present. protTrace is first such method to provide information when and when not increasing the ortholog search sensitivity benefit for identification of sequentially diverged orthologs.

2 Getting started.

2.1 Downloading protTrace from GitHub.

ProtTrace is available as a GitHub repository: https://github.com/BIONF/protTrace.git. For usage, do either of the following:

2.1.1 Cloning a git repository.

If 'git' is already installed on your computer, then clone a copy of protTrace using the following command:

```
git clone https://github.com/BIONF/protTrace.git
```

2.1.2 Downloading protTrace as zipped file.

Alternatively, download protTrace as a compressed zipped file. Unpack the downloaded file using the following command:

```
unzip protTrace-master.zip
```

2.2 Installing program dependencies.

ProtTrace requires installation of a number of softwares/libraries before a successful run can be done. Here, we list the program dependencies along with their installation procedures. Once installation is complete, the paths for these programs have to be provided in program configuration file (`prog.config`).

2.2.1 Python v2.7.13

Python 2.7.13 version can be downloaded from https://www.python.org/downloads/release/python-2713/ and the installation procedure is provided at https://docs.python.org/2/using/index.html.
Please note that the current version of protTrace has been written in Python v2.7 and thus, will not be executable with Python v3.0 or higher.
In addition, install Dendropy module for Python available at https://pythonhosted.org/DendroPy/downloading.html.

2.2.2 Perl v5 or higher

Perl v5 or higher can be downloaded and installed using the manual provided at https://www.perl.org/get.html.

2.2.3 JAVA v1.7 or higher

JAVA can be downloaded and installed using the instructions provided at https://java.com/en/download/.

2.2.4 R v3 or higher

R is available for download and installation at https://www.r-project.org/.

2.2.5 MAFFT v6 or higher

protTrace uses MAFFT tool for multiple sequence alignment. MAFFT is available for download at http://mafft.cbrc.jp/alignment/software/.

2.2.6 RAxML v8 or higher

protTrace uses RAxML tool for reconstructing maximum likelihood phylogenetic trees. RAxML is available for download at: https://sco.h-its.org/exelixis/web/software/raxml/index.html.

2.2.7 HMMER tools – *hmmscan* and *hmmfetch*

HMMER tools can be downloaded and installed from http://hmmer.org/.

2.2.8 ClustalW

ClustalW is used in protTrace to provide mutliple sequence alignment format conversions (.aln to .phy). ClustalW is available at http://www.clustal.org/clustal2/.

2.2.9 blastall / blastp / formatdb / makeblastdb

Blast executables can be downloaded from ftp://ftp.ncbi.nlm.nih.gov/blast/executables/blast+/LATEST/. Install all the programs listed above.

2.2.10 HaMStR / HaMStR-OneSeq

protTrace implements HaMStR and HaMStR-OneSeq for targeted orthologs identification. HaMStR and HaMStR-OneSeq and can be downloaded from https://github.com/BIONF/HaMStR.git.

2.2.11 TreePuzzle

In protTrace, TreePuzzle is used for likelihood mapping on the multiple sequence alignment. TreePuzzle is available to download at http://www.tree-puzzle.de/.

2.2.12 FigTree

FigTree is used for tree visualization and generating pdf files as output. FigTree is available for download at http://tree.bio.ed.ac.uk/software/figtree/.

3 Understanding protTrace directory structure.

Once protTrace is downloaded or cloned, a directory will be created with following directories – `toy_example`, `used_files` and `bin`. The details for each directory are provided below.

3.1 `toy_example.`

This is the main directory from where protTrace should be run. A program configuration file (`prog.config`) is present in this directory where paths to software dependencies and necessary files have to be set. In addition, users can configure the execution of all (some) steps in protTrace workflow. This is discussed in detail in section 4. Two test files (`test.id` and `test.fasta`) are provided to check if the protTrace run is successful.

3.2 `used_files.`

This directory contains all the supporting files for running protTrace. The content of each file is explained below.

3.2.1 `REvolver.jar`

REvolver is a protein sequence evolution simulation tool that maintains proteins domain constraints during simulation.

3.2.2 `stepWiseTree.nw`

This is a custom made tree with step-wise increasing branch lengths in 0.1 substitutions per site reaching a maximum of 7.4 substitutions per site.

3.2.3 `iqtree-24`

This tool is used to calculate insertions and deletions rates for protein.

3.2.4 `paramsMaxLikelihoodMapping.txt`

This file contains the parameters to be used with TreePuzzle for maximum likelihood mapping.

3.2.5 `r_nonlinear_leastsquare.R`

This R script is used to fit non-linear least square curve to detection probability distribution for protein.

3.2.6 `plotPdf.jar`

Javascript to plot pdfs using FigTree.

3.2.7 `speciesTreeMapping.txt`

This is a tab separated cross reference mapping file for the reference genomes of interest. The file format is the following:

<hamstr_id><tab><species_name><tab><taxa_ncbi_id><tab><species_oma_id>

For *hamstr_id*, simply take the first letter from genus name followed with the species name. Add '@' followed with taxa ncbi id, '@' and the genome version of the respective taxa used for

HaMStR. Do not use underscores ('_') in *hamstr_id* as this may result in parsing errors in protTrace. The *species_name* should be joined with an underscore. Please avoid using very large names and try to limit the species name consisting only a single underscore. *taxa_ncbi_id* is obtained for each taxa from NCBI database. *species_oma_id* for every taxa is obtained preferentially from OMA database. In cases where the taxa of interest is not present in OMA database (http://omabrowser.org/oma/home/), simply create a 5-lettered identifier using the first three letters from genus and first two letters from species name.

For example, *Homo sapiens* can have the following cross-reference format: Hsapiens@9606@1<tab>Homo_sapiens<tab>9606<tab>HUMAN. And *Chara vulgaris,* not present in OMA database, can have the following cross-reference: Cvulgaris@55564@1<tab>Chara_vulgaris<tab>55564<tab>CHAVU. Before assigning, please check that the custom *species_oma_id* (for example, CHAVU in this case) does not exists in OMA database.

Here, Hsapiens@9606@1 and Cvulgaris@55564@1 are *hamstr_id* for the two species respectively. In cases where the genome version is not known, simply write '1' or any value of your choice.

3.2.8 `speciesTree.nw`

This is a reference phylogenetic tree file for representing traceability results. This tree can be obtained using NCBI common tree tool available at https://www.ncbi.nlm.nih.gov/Taxonomy/CommonTree/wwwcmt.cgi.

Box 1: Creating species tree for your reference taxa:

1. Prepare a file containing ncbi taxa ids of your reference taxa.
2. Open web server and go to 'https://www.ncbi.nlm.nih.gov/Taxonomy/CommonTree/wwwcmt.cgi'
3. Upload the taxa ids containing file on the common tree tool and click "Add from file".
4. Save the tree as "phylip tree".
5. Open the downloaded phylip tree in FigTree.
6. Save the tree in newick format.

Please make sure that the taxa names are consistent with *species_name* in the cross-reference file discussed in section 3.2.7. Any discrepancy in the taxa names will lead to unwanted termination of the program.

3.2.9 `speciesLikelihoodMatrix.txt`

This file contains all vs all species maximum likelihood distances. The first row and first column corresponds to the *species_oma_id* present in the cross-reference file (*see* section 3.2.7). For your reference taxa of interest, the all vs all maximum likelihood distances can be calculated as explained in Box 2.

3.3 `bin`

This folder contains the main python scripts to run protTrace. There exists a file '`protTrace.py`' which is the main script to perform traceability calculations.

3.4 Others (cache, output)

These folders are generated once the protTrace is run. `cache` directory stores run files that can be re-used later for faster protTrace execution. `output` directory stores the traceability output for a protein.

NOTE: - The user can also select different name for cache and output directory by specifying it in the program configuration file. For the sake of this manual, we'll use `cache` and `output` terminology.

Box 2: Creating species maximum likelihood distance matrix for your reference taxa:

1. If your reference taxa are present in OMA database, download "Pairwise orthologs" and "Protein sequences" file from http://omabrowser.org/oma/current/.
2. If your reference taxa are not present in OMA database, generate pairwise orthologs using your favorite ortholog prediction tools. We would recommend using OMA standalone tool.
3. For every orthologous pairs between 2 taxa, calculate maximum likelihood distance. This can be done in following steps:
   - 3.1 Align the ortholog sequences.
   - 3.2 Duplicate the aligned sequences and give them separate identifier names. This way you should have an alignment file with four sequences.
   - 3.2 Convert the alignment into phylip format.
   - 3.3 Perform likelihood mapping using TreePuzzle and obtain maximum likelihood distance between the orthologous pair.
4. From all the maximum likelihood distances, choose the median as a representative for maximum likelihood distance between a set of taxa.

4 protTrace program configuration file (prog.config).

Using the *prog.config* file, you can modularly run / skip steps from protTrace workflow.

4.1 General settings

4.1.1 species

Give the name of the species for the seed protein. This id needs to be the '*species_oma_id*' format as described in section 3.2.7. For example, if the seed protein is from *Homo sapiens*, then species_oma_id can be HUMAN, the same name as defined in the cross-reference file `speciesTreeMapping.txt`.

4.1.2 nr_of_processors

The number of processors to be used by protTrace.

4.1.3 delete_temporary_files

This option cleans all the temporary files produced by protTrace. It is highly recommended to use this option.

4.1.4 reuse_cache

If set to YES, protTrace will use pre-existing information present in the cache and working directory.

4.2 preprocessing

This is the first step in the protTrace workflow. A series of events occur in preprocessing step – (i) Preparation of BLAST directory for subsequent traceability calculations. (ii) Compilation of orthologs. (iii) Calculation of FAS scores. (iv) Multiple sequence alignment for orthologs. (v) Orthologs tree reconstruction. (vi) Scaling factor (k) calculation. (vii) Insertion / deletion (indel) rates calculation. (viii) Preparing input parameter XML file for REvolver simulations. You can turn ON and turn OFF certain steps depending on your need.

4.2.1 orthologs_prediction

This flag informs protTrace whether the orthologs for the seed protein needs to be computed or is the user providing a custom set of orthologs for usage. There are 4 alternatives that protTrace currently provides for orthologs prediction.

4.2.1.1 Only OMA orthologs

If you wish to use only the pre-compiled orthologs set provided by OMA database, then simply set run_hamstr, run_hamstrOneSeq and include_paralogs flag to NO in the program configuration file and set search_oma_database flag to YES.

Box 3: Preparing files for OMA database search:

When working with OMA database, 2 files have to be downloaded from OMA database – OMA groups (http://omabrowser.org/All/oma-groups.txt.gz) and OMA Protein Sequences (http://omabrowser.org/All/oma-seqs.fa.gz).
Always check for the latest versions at http://omabrowser.org/oma/current/. Once downloaded, do the following:

```
(1)   cd path/to/download/directory
(2)   gunzip oma-groups.txt.gz           #Unzip file
(3)   gunzip oma-seqs.fa.gz              #Unzip file
(4)   awk '/^>/ {printf("\n%s\n",$0);next; } { printf("%s",$0);}  END
      {printf("\n");}' < oma-seqs.fa > oma-seqs.fa  #Convert Multi-line
      fasta to single-line fasta
(5)   sed -i -e ‘s/ >/>/g’ oma-seqs.fa    #Remove space in the fasta identifiers
      (> Identifier → >Identifier)
```

Now, the OMA files are ready to be used with protTrace. Provide the path of these files in the program configuration file under tabs path_oma_group and path_oma_seqs respectively.

4.2.1.2 OMA + HaMStR Extension orthologs

If you have reference taxa that are not present in OMA database, you can extend the orthologs search in the respective taxa using HaMStR. In this strategy, the initial core-orthologs set would be retrieved from OMA database and subsequently a profile hidden markov model based targeted

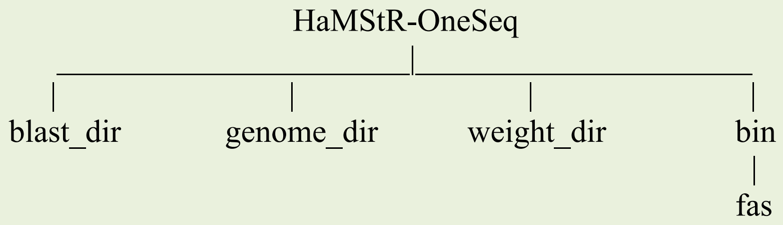
ortholog search will be done by HaMStR. To facilitate, set search_oma_database and run_hamstr flags to YES in the program configuration file.
protTrace also provides an option to include in-paralogs during ortholog search with HaMStR and HaMStR-OneSeq. If you are interested in in-paralogs to be included in the initial orthologs set, set include_paralogs flag to YES. Refer Box 4 to setup directories for HaMStR / HaMStR-OneSeq.

4.2.1.3 Only HaMStR-OneSeq orthologs

If you want to compile orthologs only using HaMStR-OneSeq, then set search_oma_database and run_hamstr flags to NO, and set run_hamstrOneSeq flag to YES in the program configuration file.
NOTE: - If all the three flags - search_oma_database, run_hamstr and run_hamstOneSeq are set to YES at the same time, then first the core-orthologs set will be retrieved from OMA database. If at least one ortholog is obtained from OMA, a targeted ortholog search will be done with HaMStR. Otherwise, if no ortholog is obtained from OMA database, a targeted ortholog search will be done with HaMStR-OneSeq.

Box 4: Preparing files / directories for HaMStR / HaMStR-OneSeq:

Once HaMStR-OneSeq has been successfully configured, you can see the following directory layout:



blast_dir :- This is the directory where blast databases for your reference taxa are to be stored. For every reference taxa, create a directory using *hamstr_id* and copy the gene set file inside this directory with same name (*see* section 3.2.7). For example, if your reference taxa is human, a gene set can look like - `/path/to/HaMStR-OneSeq/blast_dir/Hsapiens@9606@1/Hsapiens@9606@1.fa`. As a next step, create a blast database using formatdb: `formatdb –i /path/to/HaMStR-OneSeq/blast_dir/Hsapiens@9606@1/Hsapiens@9606@1.fa`

genome_dir :- This directory has the same structure as the blast directory except that the blast database for the taxa gene sets need not be created.

weight_dir :- This directory contains FAS annotations for every taxa. See Box 5 for more details.

bin :- This directory contains the scripts for HaMStR and HaMStR-OneSeq run.

fas :- This directory contains scripts required for FAS scores calculations.

4.2.1.4 Using Custom orthologs

If you wish to use your pre-computed orthologs for your seed protein, then set orthologs_prediction to NO in the program configuration file. To inform protTrace about your pre-computed orthologs, copy the orthologs file in the cache directory and rename the file to '`ogSeqs_$seedProteinName.fa`'. Make sure the orthologs file is in a single-line fasta format (refer to Box 3, step 4 to convert multi-line fasta file into single-line fasta format). The *$seedProteinName* should be the same fasta identifier as provided in the input fasta file for protTrace run. For example, if your seed protein input file is as shown below, then the *$seedProteinName* should be 'Human_OR4F5'.

```
(/path/to/protTrace/toy_example/seed_protein.fa)
>Human_OR4F5
MVTEFIFLGLSDSQELQTFLFMLFFVFYGGIVFGNLLIVITVVSDSHLHSPMYFLLANLSLIDLSL
SSVTAPKMITDFFSQRKVISFKGCLVQIFLLHFFGGSEMVILIAMGFDRYIAICKPLHYTTIMCGN
ACVGIMAVTWGIGFLHSVSQLAFAVHLLFCGPNEVDSFYCDLPRVIKLACTDTYRLDIMVIANSGV
LTVCSFVLLIISYTIILMTIQHRPLDKSSKALSTLTAHITVVLLFFGPCVFIYAWPFPIKSLDKFL
AVFYSVITPLLNPIIYTLRNKDMKTAIRQLRKWDAHSSVKF
```

As a second step, the fasta identifiers inside the orthologs file have to be changed to the *species_oma_id*, described in section 3.2.7. A simple example file is shown below.

```
(/path/to/protTrace/cache/ogSeqs_Human_OR4F5.fa)
>HUMAN
MVTEFIFLGLSDSQELQTFLFMLFFVFYGG…
>NEMVE
MNDSSSIACSSHKLEVGILLAVNCVSAIAT…
>CAEEL
MFTSTLAPMVLALLENDTSIIATTQSSMSP…
```

Here, HUMAN, NEMVE and CAEEL are *species_oma_id* for *Homo sapiens*, *Nematostella vectensis* and *Caenorhabditis elegans* respectively. In case, if in-paralogs are also present in the orthologs file, simply add a number before every identifier. For example, the following file below shows orthologs file with in-paralogs.

```
(/path/to/protTrace/cache/ogSeqs_Human_OR4F5.fa)
>HUMAN
MVTEFIFLGLSDSQELQTFLFMLFFVFYGG…
>1_NEMVE
MNDSSSIACSSHKLEVGILLAVNCVSAIAT…
>2_NEMVE
MQSTFNGTHDNTCFFLRLDTRAVHEVYASF…
>3_NEMVE
MKKSCLFYTYDSANFTEKSSLIVLIILNSV…
>1_CAEEL
MFTSTLAPMVLALLENDTSIIATTQSSMSP…
>2_CAEEL
MDSNVKYFMYEIFIPSIIILCCVAAFLNFM…
```

4.2.2 FAS Scores Calculation

FAS-S is a tool that compares feature architecture (eg – Pfam or SMART domains, low complexity regions etc.) between two proteins, and returns a Feature Architecture Similarity (FAS) score ranging between 0 and 1. By default, *FAS-S* is provided with HaMStR-OneSeq installation (for

installing HaMStR-OneSeq, *see* section 2.2.10). Once HaMStR-OneSeq is installed, you can find *FAS-S* scripts in folder `/path/to/HaMStR-OneSeq/bin/fas/`. Mainly, there are two scripts – `annotation.pl` and `greedyFAS.py`.
Before using the FAS calculations, you need to prepare FAS annotations files for your reference taxa gene sets (*see* Box 5).

Box 5: Preparing FAS annotation files for reference taxa gene sets

Lets assume that gene set for your reference taxa, *H. sapiens* is placed in HaMStR-OneSeq blast directory (`/path/to/HaMStR-OneSeq/blast_dir/Hsapiens@9606@1/Hsapiens@9606@1.fa`) and the output directory for FAS annotations is `/path/to/HaMStR-OneSeq/weight_dir/`.

```
(1)   cd path/to/HaMStR-OneSeq/bin/fas/
(2)   perl annotation.pl –fasta=/path/to/HaMStR-OneSeq/blast_dir/Hsapiens@9606@1/Hsapiens@9606@1.fa -path=/path/to/HaMStR-OneSeq/weight_dir/ -name=Hsapiens@9606@1 -force
```

This will create FAS annotations for *H. sapiens* in the `weight_dir`. Repeat the same for all reference taxa.

4.2.3 Orthologs Tree Reconstruction

protTrace reconstructs a maximum likelihood tree using the orthologs sequences. Orthologs tree is a primary requirement for the indel rate calculation. If you wish to use your pre-computed tree, then simply set orthologs_tree_reconstruction to NO and copy your tree file in the working directory. Rename the file as `RAxML_bestTree.$seedProteinName` (*see* section 4.2.1.4).The working directory is found inside the `output` directory with the same name as *$seedProteinName*. In case you are running protTrace for the first time, you may need to first create both `output` and work directory.

4.2.4 Scaling Factor Calculation

Scaling factor, κ is calculated as the median of the ratio of the pairwise maximum likelihood distance between sequences found in orthologs tree to the pairwise maximum likelihood distance between respective species in the all vs all maximum likelihood distance matrix file. For details on species all vs all maximum likelihood distance matrix file refer section 3.2.9. By default, protTrace uses a default scaling factor of 1.57 however, you are free to change the default value as per your need.

4.2.5 Multiple Sequence Alignment

This step is important for ortholog tree reconstruction, indel rates calculation and scaling factor calculation. protTrace uses first MAFFT (`--linsi`) to create an alignment file (`.aln` format) from the ortholog sequences and subsequently converts it into phylip (`.phy` format) using ClustalW.

4.2.6 Calculate Indel rates

This step requires ortholog sequences alignment file (`.phy` format) and ortholog sequences tree file as input. By default, protTrace uses indel rates of 0.08 and indel lengths distribution of 0.25 however, you can change it in the program configuration file as per your need.

4.3 Traceability Calculation

The status of this flag determines whether traceabilities calculations will be done or not. REvolver by default provides the following amino acid substitution models – WAG, JTT, JTT_dcmut, Dayhoff, Dayhoff_dcmut, mtMAM, mtART, mtREV, rtREV, cpREV, Vt, Blosum62, LG, HIVb, and HIVw. You can also use a custom substitution model by providing the path to the custom model. Secondly, you can also alter the number of simulation steps for traceability calculations.

4.4 Writing / Representing traceabilities to file / reference species tree

If set to YES, protTrace will write traceability estimates to a text file (`~/output/$seedProteinName/trace_results_$seedProteinName.txt`) and also represent traceabilities over reference species tree (`~/output/$seedProteinName/nexus_$seedProteinName_edit.nexus.pdf`).

4.5 Configuring paths for protTrace dependencies

In this section, provide paths to protTrace dependencies. We assume that python, perl and JAVA are installed and accessible globally. If not, make sure these programs are executable at a global level.

4.6 Used files / directories

In this section, provide paths for some files or directories that are prerequisite for protTrace.

4.6.1 Files provided in the `used_files` directory by default

Some files, programs are provided once you download / clone protTrace from github. The Xref_mapping_file, reference_species_tree and species_MaxLikMatrix files need to be updated as per your requirement. By default, we provide these ready-to-use files for 232 representative species from three domains of life. You can always limit the files to represent only your taxa of interest.

4.6.2 path_oma_seqs and path_oma_group

These files can be downloaded from OMA database. See Box 3 for further processing of these files.

4.6.3 pfam_database

Latest release of the Pfam database (Pfam database v31) can be downloaded from ftp://ftp.ebi.ac.uk/pub/databases/Pfam/releases/Pfam31.0/Pfam-A.hmm.gz. As a subsequent step, you need to prepare a hmm database by using *hmmpress* provided by HMMER tools. Alternatively, Pfam hmm file can be found in HaMStR-OneSeq directory (`path/to/HaMStR-OneSeq/bin/fas/Pfam/Pfam-hmms/Pfam-A.hmm`).

4.6.4 fas_annotations

Provide here the path where FAS annotations for your reference taxa are stored (*see* Box 4).

4.6.5 hamstr_environment

By default, HaMStR and HaMStR-OneSeq operate on `/path/to/HaMStR-OneSeq/blast_dir`, `/path/to/HaMStR-OneSeq/genome_dir` and `/path/to/HaMStR-OneSeq/weight_dir`. If you have stored your reference gene set in other directories like `/path/to/HaMStR-OneSeq/blast_dir_yourName and so on, then the extension has to be provided as a hamstr environment (hamstr_environment:yourName). If using default directories, simply write hamstr_environment:default in program configuration file.`

4.6.6 Path Configuration

By default, protTrace creates a `/path/to/protTrace/output` and `/path/to/protTrace/cache` directories to store the results and cache / intermediate data respectively. You can specify a different location by providing path in the program configuration file.

5 Running protTrace

Once all the dependencies are installed and all the file paths are provided in program configuration file, you are ready for the first run of protTrace. For test run, two files have been provided – `test.fasta` and `test.id`, the former containing a protein fasta sequence from *H. sapiens* (*species_oma_id*: HUMAN) and the latter containing only the OMA identifier from *S. cerevisiae* (*species_oma_id*: YEAST).

protTrace can be run in two ways:

5.1 protTrace – Input protein fasta file

If the input file is a fasta sequence (like in `test.fasta`), then simply run:

```
python /path/to/protTrace/bin/protTrace.py –f test.fasta –c prog.config
```

Here, `prog.config` is the program configuration file for protTrace.

5.2 protTrace – Input OMA identifiers

If the input file consists of OMA identifier (like in `test.id`), then simply run:

```
 python /path/to/protTrace/bin/protTrace.py –i test.id –c prog.config
```

6 protTrace Output

protTrace output files are stored in /path/to/protTrace/output/$seedProteinName/. A number of files are provided in the output directory. I list here only the most important ones:

6.1 trace_results_$seedProteinName.txt

This file contains traceabilities for every reference taxon listed in your Xref_mapping_file (`species_tree_maping.txt`). The third column in the file gives the traceabilities values.

6.2 nexus_$seedProteinName_edit.nexus.pdf

This pdf file represents traceabilities in a color-coded manner on the reference species tree (`speciesTree.nw`). The traceabilities follow a gradient from green, representing high traceability, to yellow, representing intermediate traceability, and finally to red, representing low traceability.

6.3 decay_summary_$seedProteinName.txt.pdf

This pdf file shows the detection probability decay pattern for the seed protein.

6.4 $seedProteinName_phyloMatrix.txt

This file provides ready-to-use phylogenetic profile matrix for PhyloProfile, a tool for dynamic visualization of multi-layered phylogenetic profiles. PhyloProfile is available at https://github.com/trvinh/phyloprofile.git.

6.5 ogSeqs_$seedProteinName.domains

This file stores the feature architecture information for every ortholog sequences. This file is generated only when FAS scores are calculated during protTrace run.