Manual:

This a manual for the PANNZER program.

Installation:

The PANNZER requirements are:

1. Linux OS
2. Python 2.x (https://www.python.org)
3. MySQL database (http://www.mysql.com)
4. NCBI blastall version 2.2.21 (ftp://ftp.ncbi.nlm.nih.gov/blast/executables/release/2.2.21/)

Setup a MySQL user account for the PANNZER program with reading privileges. Follow MySQL instructions: http://dev.mysql.com/doc/#manual

Download PANNZER installation package from here:
http://ekhidna.biocenter.helsinki.fi/pannzer/Download.html

Extract the files:

```
tar –xzvf pannzer.tar.gz
```

Set Python path to the folder where you extracted the PANNZER package. e.g. in Bash:

```
export PYTHONPATH="path/to/your/folder/”
```

Download databases from here. (7GB)

How to install databases:

1. Uncompress the downloaded file:

```
# tar -xvjf pannzer.tar.bz2
```

2. Load up the MySQL database:

```
# mysql -u root -p[root_password] [database_name] < Pannzer_dump.sql
```

3. Format BLAST database:

```
# formatdb -i UNIPROT -p T
```

4. Set the correct name for the BLAST database in the `.conf` -file:

```
DB=UNIPROTClean.tab
```

5. Set the correct path for the data files in the `.conf` -file:

```
DATA_FOLDER=/path/to/the/folder/ [the folder where you uncompressed
these data files]
```

PLEASE NOTICE!! If running PANNZER for the first time or the databases has been updated, set `GENERATE_IDF=True` **in** `.conf (See: Preparing and settings)` **file at the first run. This will create Python pickles and other additional tables needed for PANNZER to work. After the first run, set** `GENERATE_IDF=False.`

Preparing and settings:

First check that your input fasta file has a unique identifier for every sequence in a file. Unique identifier needs to be located before first white space in header line. Correct headers if needed.

The PANNZER analysis starts with the BLAST (or in near future the SANS alignment). Align your query sequences against the UniProtKB database using the BLAST or the SANS program. Use default output from the SANS and XML -format (-m 7) from the BLAST.

Copy and rename the `blank.conf` -file.

Edit the renamed .conf file as follows:

`[GENERAL SETTINGS]`

`INPUT_TYPE=` #Select `BLASTXML` or `SANS` depending on which sequence retrieval tool was used in previous step. (SANS not yet supported!)

`INPUT_FILE=` #Type in the name of the BLAST or SANS output file.

`XML=` #`True` if `BLAST` was used, `False` if `SANS`. (SANS doesn't support XML format yet!)

`DATA_FOLDER=` #Type in the full path to the folder where the databases were downloaded (See Installation step 3).

`DB=` # Type in the name of the database you prefer to use. From `DATA_FOLDER` select a file that has `.tab` suffix and use that, not the UniProtKB fasta file.

`RESULT_FOLDER=` #Type in the full path to the folder where you want the result files to be written.

`RESULT_BASE_NAME=` # Type in the prefix that is used in result file names.

`INPUT_BASE_NAME=` # If you are annotating multi species datasets, type in the prefix of the `.desc` -file (See: Multiple species). Leave blank if every sequence is from single species.

`INPUT_FOLDER=` #Type in the full path to the folder where the input files (BLAST or SANS output etc.) are located.

QUERY_TAXON= #Fill in the NCBI taxonomy number of the query organism if multiple species is NOT used. If taxonomy number is not available, select a closely related species or use taxonomy number of the parent class in NCBI taxonomy lineage.

GET_TAXON= #Set False if QUERY_TAXON is used.

GENERATE_IDF= True **if PANNZER is used for the first time or the databases has been updated**, else False.

MULTIPLE_SPECIES= #True if multiple species in fasta file (**requires .desc file!! See:** Using multiple species), with single species False (Requires QUERY_TAXON).

[TRESHOLD VALUES]

Fill in the threshold values for the sequence filtering. Default values can be found from blank.conf.

[MYSQL]

Fill in the MySQL user name, password etc. **Nb! SQL_DB has to be corresponding to the one that was used in INPUT_FILE.**

[TAXONOMY]

DB= #Taxonomy lineage file (taxonomy-all.tab)

CALCULATE= # True if you want to use taxonomic distance between species, else False.

NODE_SELECTOR= # The level in NCBI taxonomy tree where the hits are reported in output file. E.g. 1 will report counts of Bacteria, Archaea, Eukaryota, etc hits. **Doesn't affect on results!**

ONLY_ONE_HIT_PER_SPECIE= #True if only best hit from each species are included in results, else False.

[GO]

WRITE_GO= #True if GO prediction is used, else False.

OBO= #Fill in the Gene Ontology OBO file name (gene_ontology_ext.obo).

ID_MAPPING= #Specify the "idmapping" file (idmapping_selected.tab)

ENZYME= #Specify the Enzyme database file (enzyme.dat)

[LEVEL_OF_PRINTING]

`CLUSTER=` #If you want to print all clusters `True`, else `False`.

`CLUSTER_MEMBERS=` #If you want to print all cluster members `True`, else `False`.

`ALL=` #If you want to print all hits `True`, else `False`.

`ERROR=` #If you want to print possible error's messages `True`, else `False`.

`DEBUG=` #If you want to print debug level messages `True`, else `False` .

`INFO=` #If you want to print info level messages `True`, else `False` .

`[EVALUATION]`

Modify evaluation parameters only if you know what you are doing!!!

Running PANNZER

Run PANNZER program with command: `python run.py file.conf`
where `file.conf` is the configure file you modified in previous step.

Using multiple species

If you have a fasta file that contains multiple different species, you need a .desc file to identify these species.

File format for the .desc file is following:

1. Unique id for the sequence (located before first white space in header line)
2. Empty / `NA` (reserved for debugging and evaluation only)
3. NCBI taxonomy number

Fields are separated with tabs.

Example:

```
Q8GBW6    NA   1752
P0A5B8    NA   233413
A6X5T5    NA   439375
Q84I68    NA   33990
A5EJ46    NA   288000
...
```

Name the file and fill the prefix of the file in .conf file where `INPUT_BASE_NAME` is required. Set `MULTIPLE_SPECIES= True` in .conf file.