MPACT

Multimetric Pairwise Comparison Tool

Usage Manual and Tutorial

2025

1 Introduction

MPACT, the **M**ultimetric **PA**irwise **C**omparison **T**ool, is an integrated toolbox for pairwise all-against-all comparison of primary nucleotide or amino acid sequences, and 3D protein structures. For nucleotide sequences, the program utilizes identity percentage and maximum likelihood (ML) distance as metrics. In the case of proteins, MPACT sequence identity and similarity percentages and maximum likelihood distance of amino acid sequences, and structural similarity (TM-scores) and 3Di character similarity of 3D-structures. For each metric, the program performs data clustering and generates heatmaps, frequency distribution plots, and dendrograms. Finally, MPACT can also partition the dataset according to user-defined criteria for each metric. MPACT is written in Python 3 (https://www.python.org/) and utilizes the following third-party programs: Needle (from the EMBOSS package version 6.6.0 [23]), MAFFT [14], IQ-TREE 2 [15], Foldseek [22] and TM-align [24]. The program executable, usage manual, tutorial, and a Docker container image for easy execution are available on GitHub (https://github.com/gruberlab/mpact).

2 Main features

- All-against-all pairwise comparison of sequences and protein structures using five possible metrics: identity, amino acid similarity, maximum likelihood distance, TM-scores, and 3Di character similarity.
- Generation of comparison results in the form of customizable heatmaps.
- Sample clustering in heatmaps based on pairwise comparison values, employing various methods, including neighbor-joining, UPGMA, and others.
- Construction of dendrograms representing relationships obtained through clustering processes.
- Creation of distribution graphs for pairwise comparison values.
- Access to all intermediate files generated by third-party programs, primarily including alignments and phylogenetic reconstructions.
- Generation of all-against-all pairwise comparison matrices for the five available metrics.

- Production of tables displaying intra- and intergroup maximum and minimum values for labeled subsets.
- Data partitioning of samples (sequences or structures) in FASTA format files according to a user-defined range of values of one of the available metrics.

3 Workflow of the program

The MPACT program's workflow is depicted in Figure 1. The program utilizes two types of input data: biological sequences, protein or nucleotide, in FASTA format, and 3D protein structure files in PDB format. Biological sequences are submitted to all-against-all pairwise alignments using the Needleman-Wunsch algorithm implemented in the Needle program (EMBOSS package). MPACT then extracts global identity and similarity percentage values from these alignments and stores the results as matrices. The input sequences are also submitted to a multiple sequence alignment (MSA) with MAFFT. The resulting MSA is then used ML phylogenetic analysis using IQ-TREE 2, and the resulting ML phylogenetic tree and distance matrix are stored. Three-dimensional protein structures are submitted to all-against-all pairwise structural comparisons using TM-align. All resulting pairwise scores, normalized by the average length of the compared structures, are used to create a TM-score matrix. PDB files are also converted to 3Di-character sequences by the Foldseek program, and these sequences are aligned using Needle with a 3Di substitution matrix [22].

All matrices are subjected to data clustering using the clustermap function of the Seaborn data visualization library. Clustering results are stored as trees (Newick format), and heatmap images in both raster (JPG) and vector (SVG) formats. MPACT also converts the matrix data into range-tabulated values and generates frequency distribution plots using the Seaborn's lineplot function.

The final section of the workflow is devoted to data partitioning. In a first step, all matrices are converted to distance matrices. The data is then submitted to Neighbor-joining (NJ) hierarchical clustering using the Biopython's Bio.Phylo package. The generated NJ tree is rooted at the midpoint, and its nodes are ordered in increasing order of branch length. This order is used for data partitioning to generate groups of sequences selected within a range of user-defined upper and lower values for each chosen metric. Finally, MPACT stores the resulting tree, heatmap, and sequence files.

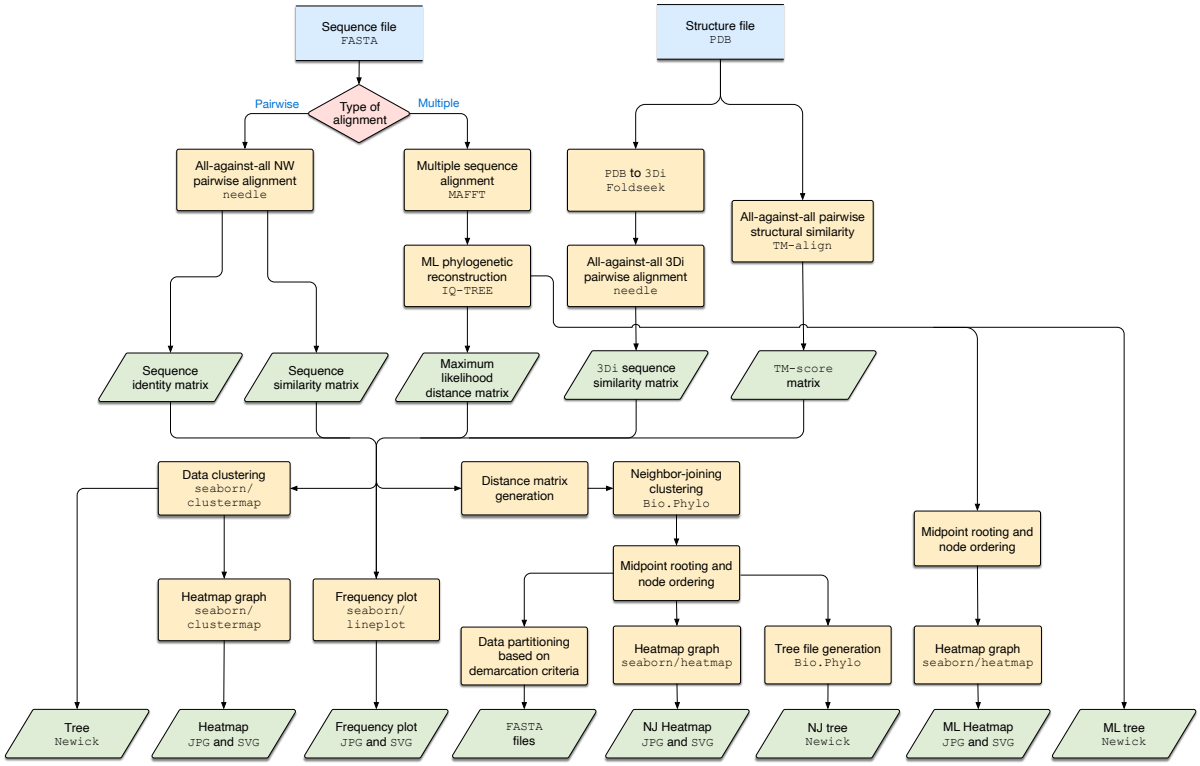

Figure 1. Workflow of the MPACT program. The program utilizes two types of input data: biological sequences in FASTA format and 3D protein structures in PDB format. Sequences are submitted to all-against-all pairwise alignments using the needle program (EMBOSS package), and to a multiple sequence alignment (MSA) using the MAFFT program. The MSA is submitted to the IQ-TREE program for phylogenetic analysis, generating a maximum likelihood (ML) tree and a heatmap. Three-dimensional protein structures are subjected to all-against-all alignments using the TM-align program, and are also converted to 3Di characters by Foldseek, and aligned with needle. Alignment results are clustered and used to generate heatmaps and frequency distribution plots. Also, distance matrices are produced and the data clustered by the Neighbor-joining algorithm, and the resulting tree is rooted at the midpoint and sorted in ascending order of the nodes. This order is used for data partitioning to generate groups of sequences selected within a range of user-defined values.

4 Before using MPACT

4.1 System requirements

MPACT was developed in Python3 and can be used in any POSIX-compliant operating system, such as UNIX and Linux distributions with an installed Python interpreter (https://www.python.org/).

4.2 Manual installation of the program

MPACT requires the following third-party programs:

- `needle` (https://emboss.sourceforge.net/apps/release/6.0/emboss/apps/needle.html)
- `MAFFT` (https://mafft.cbrc.jp/alignment/software/)
- `IQ-Tree` (https://github.com/iqtree/)
- `Foldseek` (https://github.com/steineggerlab/foldseek)
- `TM-align` (https://zhanggroup.org/TM-align/)

If the user intends to run the Python program directly, it is essential to install these programs and ensure they are included in the operating system's PATH.

The MPACT script can be downloaded from the program's GitHub (https://github.com/gruberlab/mpact).

4.3 Installing MPACT with a Docker container image

Two MPACT docker images are available on the Docker repository. These images come pre-installed with the third-party programs mentioned in the previous section, eliminating the need for manual installation. To use them, the user only needs to install Docker (https://docs.docker.com/) and download the image using the command provided in Section 6.2.

The first image, named *mpact*, includes all third-party programs and was created on a system with an Intel® Core™ i7-5500U 2.40 GHz processor running Linux (Ubuntu 24.04.1 LTS). Due to Foldseek limitations, this image is only compatible with machines equipped with an Intel® Core™ i7-5500U 2.40 GHz processor or higher. The second image, *mpact_no_fs*, does not include the Foldseek program and, therefore, is not subject to the processor restrictions mentioned above.

4.4 How to cite

If you use MPACT for your publication, please cite it as following:

- MPACT program (developed by Igor Custódio dos Santos and Arthur Gruber, University of São Paulo, Brazil, manuscript in preparation)

5 Understanding MPACT parameters

5.1 Mandatory parameters:

MPACT requires two mandatory parameters that must be specified by the user either in the command line or in the configuration file:

`-i | input <file name>:` Input fasta file containing biological sequences. The program verifies that the file exists, is not empty, and begins with ">". If any of these conditions are not met, an error is reported. Additionally, the program determines whether the file contains nucleotide or protein sequences.

`-s | structure <directory>`: Directory containing 3D protein structure files in PDB format.

5.2 Optional parameters:

`-c | color <parameter>`: Color palette for graphs (default = RdBu). Accepted palettes are any that are available in the matplotlib library (https://br.matplotlib.net/stable/tutorials/colors/colormaps.html#google_vignette).

`-conf <file>:` Configuration file in which all parameters can be specified. In the file, each line must start with the parameter letter, followed by a equal sign and the parameter value. The format of the configuration file should be as shown below:

```
i= seq_teste.fasta
p= 0,1,2
o= saida_teste_conf
l= yes
c= Spectral
f= --maxiterate 1000 --thread 20
```

`-f | MAFFT` <parameters>: specifies the parameters for multiple sequence alignment with the MAFFT program. The user can change `--maxitarate` (the number of iterative refinement cycles – default: 1000) and `--threads` (the number of processing cores – default: half of the machine's processing cores). The argument for this parameter must be a string (use quotes on the command line). For example: `-f "--maxiterate 1000 --thread 20".`

`-g| subGroups <parameter>:` specifies metric, lower limit, and upper limit, in that order, separated by commas. Grouping is done hierarchically based on the order of samples presented in the neighbor-joining heatmap. For example: `-g`

`1,65,90` (sequences with a similarity value between 65% and 90% are grouped together).

`-h|help`: program's help page.

`-k| Clustering method <parameter>`: specifies the clustering method that determines the position of the samples on the heatmap graph. It is possible to choose between "nj" (default, from Bio.Phylo package and always executed), "average", "single", "complete", "weighted", "centroid", "median" or "ward" (scipy.cluster.hierarchy.linkage methods).

`-l (lines) <yes|no>:` specifies whether lines will be displayed between cells in the final heatmaps. Lines are white and are not recommended for heatmaps with many sequences. (default = no).

`-n | names <id|org|all>`: specifies how sequences in the FASTA file should be labeled in the final heatmap.

```
id: YP_003289293.1
org: Drosophila totivirus 1
all:YP_003289293.1_RNA-dependent
RNA_polymerase_Drosophila_totivirus_1
```

This parameter will work only if the sequence headers have the NCBI standard (e.g.: ">YP_003289293.1 RNA-dependent RNA polymerase [Drosophila totivirus 1]). Otherwise, it will be available `id` and `all` options.

`-o | output <name>`: Output directory to store files.

`-p protocol <integer> (default=all methods):` specifies which protocols to run when MPACT is executed. By default, the program runs all protocols (0, 1, 2, 3, 4). The format for this parameter is a comma-separated list of protocol numbers (e.g., "0,2,3").

```
Metric options: 0 - pairwise aa sequence identity
                1 - pairwise aa sequence similarity
                2 - maximum likelihood (ML)
                3 - pairwise structural alignment
               4 - pairwise 3Di-character alignment
Examples:
          ML 7.5-9,0: -s 2,7.5,9.0
          Similarity 70-85%: -s 2,70,85
```

`TM-score 0.75-1.0: -s 3,0.75,1`

`-q | IQ-TREE <parameters>`: IQTREE parameters. The user can change "`-m`", "`-bb`" e "`-nt`" parameters, which refer to the evolutionary model, number of bootstrap replicates (default: 1000), and the number of processing cores to be used (default: half of the processing cores available on the machine), respectively. The evolutionary model has no default; IQ-TREE establishes the best one.

e.g.: `-iqtree "-m Q.pfam+F+R6 -bb 1000 -nt 20"`

`-t tags <prefixes>`: Prefixes of already known groups to obtain their intra- and inter-group value ranges. Separate the prefixes by commas.

e.g. `-t "Group1,Group2,Group3"`

`-v | version`: displays the program version.

5.3 About third-party programs

This section describes the parameters used in each third-party program.

5.3.1 needle

For nucleotide and amino acid sequences, the gap open and gap extension parameters used in MPACT were set to their default values (respectively 1.0 and 0.5). The EBLOSUM62 matrix was used for amino acid sequences, while the EDNAFULL matrix was applied for nucleotide sequences.

When 3Di character sequences were used, the gap opening penalty and gap extension parameters applied were set to 8.0 and 2.0, respectively. For the substitution matrix, a specific matrix for 3Di characters, available on the Foldseek program's GitHub page (https://github.com/steineggerlab/foldseek/blob/master/data/mat3di.out#L1C1-L25C86) was used.

5.3.2 MAFFT

To run MAFFT, the parameters for the maximum number of interactions (`--maxiterate`) and number of processing cores (`--thread`) can be configured. By default, MPACT uses `--maxiterate` to 1000 and `--thread` to half of of the available processor cores.

5.3.3 IQ-TREE

The following parameters are used to run the IQ-TREE program in MPACT:

`-bb`: specifies the number of bootstrap replicates (1000: default);

`-nt:` specifies the number of processing cores. (default: half of the available processor cores).

`-m:` defines the evolutive model (default: `Q.pfam+F+R6`).

The user can modify the evolutionary model by setting the **-q** parameter in MPACT..

5.3.4 Foldseek

The Foldseek program is run by MPACT using the `structureto3didescriptor` mode. The following parameters are also defined:

-v 0: runs Foldseek in silent mode;

--threads: specifies the number of processing cores (default: half of the available processor cores).

`--chain-name-mode:` determines whether the sequence naming mode includes the entire PDB file name (value: 1).

5.3.5 TM-align

Regarding the execution of `TM-align`, the only parameter specified by MPACT is -a `"T"`, which instructs the program to calculate the TM-score normalized by the average lengths of the protein chains in the compared structures.

6 Running MPACT

6.1 Running the MPACT using the script

To run MPACT using the Python script, the user should download the script from GitHub (https://github.com/gruberlab/mpact) and run the program using the following command line:

```
python3 MPACT.py -i <fasta_file> -s <protein_structure_dir> -o <output_dir>
```

6.2 Running the MPACT using docker images

To run the program using a docker image, the user should download the image file from the Docker repository to a directory of your choice. If your computer has a x86_64 processor, please download the image that contains the Foldseek using the following command:

```
docker pull lilianesantana/mpact:1.0
```

Unless specified otherwise, please download the image without the Foldseek using the command below:

```
docker pull lilianesantana/mpact_no_fs:1.0
```

To run Docker images, the user must use the docker program with the *run* parameter, as demonstrated in the next section. This section illustrates the execution of the program using both Docker files and the Python script, with examples from some viral groups.

6.3 MPACT tutorials

To install the data for the tutorials, download the `mpact_tutorial.tar.gz` file from GitHub to a directory of your choice. Decompress the file using the following command:

```
tar xzvf mpact_tutorial.tar.gz
```

This command will create the `mpact_tutorial` directory. This directory contains the following items:

- `Microviridae_119_VP1.fasta`: it contains 119 protein sequences of VP1 (major capsid protein) of Microviridae phages.
- `119_VP1_Microviridae`: files containing the 3D protein structures for the 119 VP1 protein sequences.
- `Microviridae.conf`: a configuration file for running MPACT on the 119 Microviridae VP1 protein dataset. This file should be used when executing the Python program.

- `Microviridae_docker.conf`: configuration file for running MPACT on the 119 Microviridae VP1 protein dataset. This file should be used when running one of the docker images.
- `Orthobunyavirus_55_segment_L_prefix.fasta`: contains 55 nucleotide sequences of L segment from Orthobunyavirus genus.
- `Orthobunyavirus.conf`: configuration file to run MPACT on the dataset of 107 *Orthornavirae* RDRP proteins. This file should be used when executing the Python program.
- `Orthobunyavirus_docker.conf`: same file described in the previous item to be used when running one of the docker images.
- `Orthobunyavirus2.conf`: configuration file to run data partitioning based on identity values on the dataset of 107 *Orthornavirae* RDRP proteins. This file should be used when executing the Python program.
- `Orthobunyavirus2_docker.conf`: same file described in the previous item to be used when running one of the docker images.
- `Orthobunyavirus3.conf`: configuration file to run data partitioning based on maximum-likelihood values on the dataset of 107 *Orthornavirae* RDRP proteins. This file should be used when executing the Python program.
- `Orthobunyavirus3_docker.conf`: same file described in the previous item to be used when running one of the docker images.
- `Orthornavirae_107_RDRP.fasta`: contains 107 protein sequences of RDRP (RNA-dependent RNA polymerase) of Orthornavirae viruses.
- `107_RDRP_Orthornavirae`: files containing the 3D protein structures for the 107 RDRP protein sequences.
- `Orthornavirae.conf`: configuration file to run MPACT on the dataset of 107 Orthornavirae RDRP proteins. This file should be used when running the Python program.
- `Orthornavirae_docker.conf`: same file described in the previous item to be used when running one of the docker images.

6.3.1 Example 1 – Using 119 protein sequences of VP1 (major capsid protein) of Microviridae phages

To run the program using the configuration file and the Python script, the following command can be used:

```
python3 mpact.py -conf Microviridae.conf
```

If you prefer to run the program using one of the Docker images, you must create a virtual directory in the execution line where the files will be stored, which will be associated with a real directory (parameter **-v**). In the command below, we pass the full path of the `mpact_tutorial` folder, and the virtual directory `/mnt` is generated:

```
docker run -v <PATH>/tutorial_dir:/mnt mpact(no_fs) -conf
/mnt/ Microviridae.conf
```

The `Microviridae.conf` configuration file includes the following content:

```
i=Microviridae_119_VP1.fasta
s=119_VP1_Microviridae
o=Output_Microviridae
```

Alternatively, we can execute MPACT with exactly the same parameters and options using the line command. Bellow, you can find the commands for both the Python script and the Docker image.

Python script:

```
python3 mpact.py -i Microviridae_119_VP1.fasta -s
119_VP1_Microviridae -o Output_Microviridae
```

Docker image:

```
docker run -v <files_directory>:/mnt mpact(no_fs) -I /mnt/
Microviridae_119_VP1.fasta -s /mnt/119_VP1_Microviridae -o
/mnt/Output_Microviridae
```

6.3.2 Understanding the parameters

Both previous command line run MPACT with all five available metrics (default: -p 0,1,2,3,4): identity (0), similarity (1) and maximum-likelihood distance (2) of the sequences from the Microviridae_119_VP1.fasta file (-i) and TM-scores (3) and 3Di characters similarity (4) of the structures from 119_VP1_Microviridae directory (-s). The output directory (-o) is named `Output_Microviridae`.

6.3.3 Inspecting the output files

Once MPACT finishes processing, an output directory will be created. Within this directory, you will find several files and sub-directories created by MPACT:

- `119_VP1_Microviridae_3Di.fasta:` FASTA format file containing 3Di character sequences obtained with the `Foldseek` program from the 3D protein structures. This file is used to run `needle` with the 3Di substitution matrix to obtain the 3Di similarity values.
- `foldseek_dir:` Directory comprising the output files generated by `Foldseek`. The file `Output_Microviridae_foldseek_output.txt` is the output file from the `Foldseek` program, containing a table with the structure name, amino acid sequence, 3Di character sequence, and various e-values generated by the program. `error1` and `error2` files are error redirect files from the `Foldseek` run. These can be inspected in the case of an error in the `Foldseek` run, but, ideally, these files are empty.
- `graphics_dir_1:` Directory of graphics generated by `MPACT`. The names of heatmaps contain details about themselves, such as metric used (e.g.: "`mldist`" or "`tmscores`"), clustering method (ends with "nj" or others) and type of file (extensions "`svg`" or "`jpg`"). The frequency distribution plots end with "`_freqplot`" and are generated for every metric. See below all the files that must be in your directory:
  - `Output_Microviridae_3Di_freqplot.jpg`
  - `Output_Microviridae_3Di_freqplot.svg`
  - `Output_Microviridae_3Di_nj.jpg`
  - `Output_Microviridae_3Di_nj.svg`

- Output_Microviridae_ident_freqplot.jpg
- Output_Microviridae_ident_freqplot.svg
- Output_Microviridae_ident_nj.jpg
- Output_Microviridae_ident_nj.svg
- Output_Microviridae_mldist_freqplot.jpg
- Output_Microviridae_mldist_freqplot.svg
- Output_Microviridae_mldist_nj.jpg
- Output_Microviridae_mldist_nj.svg
- Output_Microviridae_mldist_phylogeny.jpg
- Output_Microviridae_mldist_phylogeny.svg
- Output_Microviridae_simil_freqplot.jpg
- Output_Microviridae_simil_freqplot.svg
- Output_Microviridae_simil_nj.jpg
- Output_Microviridae_simil_nj.svg
- Output_Microviridae_tmscores_freqplot.jpg
- Output_Microviridae_tmscores_freqplot.svg
- Output_Microviridae_tmscores_nj.jpg
- Output_Microviridae_tmscores_nj.svg

If any clustering method other than *nj* is specified, in addition to the heatmap files, the dendrograms resulting from these clustering processes will also be stored in this directory.

- iqtree_dir: Directory of output files of the IQ-TREE phylogenetic reconstruction. The most important file is the one of extension ".mldist", which is the file that contains the maximum likelihood distance matrix. Also important is the resulting phylogenetic tree, contained in the file with the extension ".treefile". The error files can be inspected in the case of an error on IQ-TREE run. The error1 presents the content of a IQ-TREE logfile and error2 is ideally empty. See below all the files that must be in your directory:
  - error1_Output_Microviridae_iqtree
  - error2_Output_Microviridae_iqtree
  - Output_Microviridae
  - Output_Microviridae.bionj
  - Output_Microviridae.ckp.gz
  - Output_Microviridae.contree

  - Output_Microviridae.iqtree
  - Output_Microviridae.log
  - Output_Microviridae.mldist
  - Output_Microviridae.model.gz
  - Output_Microviridae_ordered.treefile
  - Output_Microviridae.splits.nex
  - Output_Microviridae.treefile

- mafft_dir: Directory of output files of the MAFFT multiple sequence alignment. Output_Microviridae.align is the main file in this directory; it is the multiple sequence alignments of the FASTA sequence given to the program. The other file in this directory is error_Output_Microviridae_mafft, which contains information about the MAFFT run.
- needle3Di_dir: The directory of output files for the 3Di character sequences' similarity (with specific substitution matrix for 3Di character sequences) is determined with the needle program. The file of extension ".needle" is the main output of the needle program, which concatenates all the results of pairwise alignments between sequences of the input dataset. The pattern of result in .needle file is:

```
#=======================================
#
# Aligned_sequences: 2
# 1: Alpavirinae_CAAFKQ010000288_1
# 2: Alpavirinae_CAAGFK010000198_1
# Matrix: /home/geninfo/isantos/programação/3A-DGT/3Di_matrix.txt
# Gap_penalty: 8.0
# Extend_penalty: 2.0
#
# Length: 698
# Identity:     269/698 (38.5%)
# Similarity:   411/698 (58.9%)
# Gaps:         163/698 (23.4%)
# Score: 1152.0
#
#
#=======================================

Alpavirinae_C      1 DPPPPPPPP-DDDDFDWDKDWPWDWDWAFFFAFFKAFQDKDWDAAFKKKK     49
                     ||||||..| ..|..|:||||:||:.|:|||:|:||||.||||:|::|||
Alpavirinae_C      1 DPPPPPVCPLLFDLPDKDKDWDWDKFWDFFFFFQKAFQEKDWDFAQWKKK     50

Alpavirinae_C     50 KWKWKKKWFFWWQFLFPWWKKKKKFKWWDFVCLAAVCVVVQVVVDQDDPD     99
                     |:|:|||.|:.:.|::|:|::|||||:|||.||:||||..:||.:.||
Alpavirinae_C     51 KKKKKKKAFADFPFQAPFWWWKKKFKFWDFPCLQAVCVLQCVVQHWDD--     98

Alpavirinae_C    100 PPPFNRDRPDDAQWEF--PVVLLVQLL----------------------L    125
                      :|:..||.:||:|||  ..|||.|||                     .
Alpavirinae_C     99 -DPADIDRQPDAAWEFLLVLVLLLQLLWDKWKWAQDPVRDTDTDQPPPVV    147
```

```
Alpavirinae_C    126 LVVDDQPAPLCADPQGFGQS------QQLQRRCVSLPV-DGDPV------    162
                     |||.:||:.|:..:..::::      .......|:.|. |:|||
Alpavirinae_C    148 LVVQPQPDQLVPRDRPARCNVRHDPDDDDPQDPVNAPDWDDDPVRRMIMG    197
```

Furthermore, `Output_Microviridae_3Di_simil_matrix.csv` is the 3Di character similarity values matrix in CSV format. `Output_Microviridae_stdout.txt` is a redirect file, which usually remains empty.

- `needle_dir`: Directory of output files of the protein identity and similarity (with `EBLOSUM62` substitution matrix) determination with `needle`. The file of extension `".needle"` is the main output of the `needle` program, which concatenates all the results of pairwise alignments between sequences of the input dataset. The pattern of result in `.needle` file is:

```
#=======================================
#
# Aligned_sequences: 2
# 1: Alpavirinae_CAAFKQ010000288
# 2: Alpavirinae_CAAGFK010000198
# Matrix: EBLOSUM62
# Gap_penalty: 10.0
# Extend_penalty: 0.5
#
# Length: 750
# Identity:     132/750 (17.6%)
# Similarity:   234/750 (31.2%)
# Gaps:         267/750 (35.6%)
# Score: 171.5
#
#
#=======================================

Alpavirinae_C      1 -MNSIFAK----QKSKANLQRNAFDLSRSDIFSASAGMLLPCFVEEVNPN     45
                       :.:||.|    .|:|.......||.:.:..|    |::.|.|.:.:|.|
Alpavirinae_C      1 IIMAIFDKISVGVKTKKYTHHIPFDNNTTMQF----GVIQPLFHQFLNAN     46

Alpavirinae_C     46 EHFEITPSSFLRTMPLNTAAYTR--LKQNVEFY-----------FVPYRL     82
                     :.........:|..||....:.|  |:.||:|.           |:.::.
Alpavirinae_C     47 DRLSSDVRQLVRLSPLPVPTFGRMHLENNVKFVKMSDIYPAYEAFLSHKF     96

Alpavirinae_C     83 LLRQFPQFIVG--------------TEYSISSVGQLNT---YKAPLPSLD    115
                     :.....|:|..              .:||:.:..:.||   |.......|
Alpavirinae_C     97 VNTSHSQYIPVQLPFVTPALLCLWLLKYSVYATYKRNTDGNYALTTLPAD    146

Alpavirinae_C    116 FQKTLLALV-------------------GAASASPS---------SEHSL    137
                     .| ||.|.:                    ..:.:.|.         .||.:
Alpavirinae_C    147 VQ-TLSATLFDPHTGKTRYNLKNFFTPQAISNVTPDGADYVVYSPDEHYI    195
```

Furthermore, `Output_Microviridae_simil_matrix.csv` and `Output_Microviridae_simil_matrix.csv` are the amino acid identity and similarity values matrix in csv format, respectively.

`Output_Microviridae_stdout.txt` is a redirect file, which remains usually empty.

- `neighbor-joining_dir`: Directory for files related to neighbor-joining clustering executions. The files that end with `njdistmat.csv` are distance matrices generated from similarity and dissimilarity values of all the metrics of `MPACT`. Additionally, the `.tree` extension files are the neighbor-joining trees that translate the relationships obtained in the clustering.
- `Output_Microviridae_1.log`: `MPACT` program log file. The numbering increases each time you run `MPACT` again with the same output directory. The log file contains information about all parameters provided by the user at runtime, date and time information, all third-party program execution commands, file save paths, and all metrics matrices.
- `TMalign_dir`: Directory of output directories of the `TM-align` pairwise comparisons between the 3D protein structures (`PDB` files). In each directory, the file "`.log`" contains the main output of `TM-align` program for each pairwise comparison. Furthermore, `Output_Microviridae_tmscores_matrix.csv` is the TM-score values matrix in csv format.

6.3.4 Example 2 – Using 55 nucleotide sequences of L segment of Orthobunyavirus viruses and data partitioning

6.3.4.1 Generating identity and maximum-likelihood distance graphics

To run the program using the configuration file and the Python script, the following command can be used:

```
python3 MPACT.py -conf Orthobunyavirus.conf
```

To run using a Docker image, you must also create a virtual directory in the execution line where the files will be stored, which will be associated with a real directory (parameter **-v**) :

```
docker run -v <PATH>/tutorial_dir:/mnt mpact(no_fs) -conf
/mnt/Orthobunyavirus_docker.conf
```

The `Orthobunyavirus.conf` configuration file includes the following content:

```
i= Orthobunyavirus_55_segment_L_prefix.fasta
p=0,2
o=Output_Orthobunyavirus
```

Alternatively, we can execute MPACT with exactly the same parameters and options using the line command. Bellow, you can find the commands for both Python script and the docker image:

Python script:

```
python3 MPACT.py -i Orthobunyavirus_55_segment_L_prefix.fasta
-p 0,2 -o Output_Orthobunyavirus
```

Docker image:

```
docker run -v <files_directory> :/mnt mpact(no_fs) -I /mnt/
Orthobunyavirus_55_segment_L_prefix.fasta -s 0,2 -o /mnt/
Output_Orthobunyavirus
```

6.3.4.2 Data partitioning based on identity

Python script with configuration file:

```
python3 MPACT.py -conf Orthobunyavirus2.conf
```

Docker image with configuration file:

```
docker run -v <PATH>/tutorial_dir:/mnt mpact(no_fs) -conf
/mnt/Orthobunyavirus2.conf
```

The `Orthobunyavirus2.conf` configuration file includes the following content:

```
i= Orthobunyavirus_55_segment_L_prefix.fasta
p=0
o=Output_Orthobunyavirus
g=0,65,100
```

Python script in command line:

```
python3 MPACT.py -i Orthobunyavirus_55_segment_L_prefix.fasta
-p 0  -o Output_Orthobunyavirus -g 0,65,100
```

Docker image in command line:

```
docker run -v <files_directory> :/mnt mpact(no_fs) -I /mnt/
Orthobunyavirus_55_segment_L_prefix.fasta -s 0 -o /mnt/
Output_Orthobunyavirus -g 0,65,100
```

6.3.4.2 Data partitioning based on maximum-likelihood distance

Python script with configuration file:

```
python3 mpact.py -conf Orthobunyavirus3.conf
```

Docker image with configuration file:

```
docker run -v <PATH>/tutorial_dir:/mnt mpact(no_fs) -conf /mnt/
Orthobunyavirus3.conf
```

The `Orthobunyavirus3.conf` configuration file includes the following content:

```
i= Orthobunyavirus_55_segment_L_prefix.fasta
p=2
o=Output_Orthobunyavirus
g=2,0,1.5
```

Python script in command line:

```
python3 mpact.py -i Orthobunyavirus_55_segment_L_prefix.fasta
-p 2  -o Output_Orthobunyavirus -g 2,0,1.5
```

Docker image in command line:

```
docker run -v <files_directory> :/mnt mpact(no_fs) -I /mnt/
Orthobunyavirus_55_segment_L_prefix.fasta -s 2 -o /mnt/
Output_Orthobunyavirus -g 2,0,1.5
```

6.3.5 Understanding the parameters

On the first run, only identity (0) and maximum-likelihood distance (2) metrics are specified, as these two metrics are available for the nucleotide sequences from Orthobunyavirus_55_segment_L_prefix.fasta file (-i). The main output directory is named Output_Orthobunyavirus.

On the second run, data partitioning is performed. The -g parameter 0,65,100 specifies that sequences with 65% to 100% of identity (0) are clustered together. The metric specified in the -g parameter is also specified in the -p parameter.

Finally, on the third run, another data partitioning is performed. The -g parameter 2,0,1.5 specifies that sequences with a maximum-likelihood distance of 0 to 1.5 (0) are clustered together.

6.3.6 Inspecting the output files

On the first run, the files and directories in the output directory are the same as those related to Microviridae sequences explained in item 6.3.3. These include:

- iqtree_dir/
- graphics_dir_1/
- mafft_dir/
- needle_dir/
- neighbor-joining_dir/

- Output_Orthobunyavirus_1.log

On the second run, the program will create only the clstr_ident_65_100 directory and the log file. In the clstr_ident_65_100 directory, the FASTA files (one for each cluster) have standardized names, with [No. cluster]cl[No. sequences]s.fasta. There should be 18 files (18 clusters) contained in this directory. They are:

- 10cl1s.fasta
- 11cl9s.fasta
- 12cl2s.fasta
- 13cl1s.fasta
- 14cl1s.fasta
- 15cl4s.fasta
- 16cl13s.fasta
- 17cl1s.fasta
- 18cl1s.fasta
- 1cl8s.fasta
- 2cl1s.fasta
- 3cl3s.fasta
- 4cl1s.fasta
- 5cl4s.fasta
- 6cl1s.fasta
- 7cl1s.fasta
- 8cl2s.fasta
- 9cl1s.fasta

Similarly to the second run, the third run will generate only the clstr_mldist_0_1.5 directory and the log file. The content of the clstr_mldist_0_1.5 directory, with 13 clusters, must be:

- 10cl5s.fasta
- 11cl1s.fasta

- 12cl2s.fasta
- 13cl2s.fasta
- 1cl9s.fasta
- 2cl4s.fasta
- 3cl3s.fasta
- 4cl1s.fasta
- 5cl13s.fasta
- 6cl1s.fasta
- 7cl1s.fasta
- 8cl9s.fasta
- 9cl4s.fasta

6.3.7 Example 3 – Using 107 protein sequences of RDRP of Orthornavirae viruses

In the last example, we will use **Orthornavirae.conf** in our analysis. To run it using the Python script, the command line should be executed as follows:

```
python3 mpact.py -conf Orthornavirae.conf
```

If the user chooses to use a Docker image, the command used should be the one below:

```
docker run -v <PATH>/tutorial_dir:/mnt mpact(no_fs) -conf /mnt/
Orthornavirae.conf
```

The `Orthornavirae.conf` configuration file includes the following content:

```
i=Orthornavirae_107_RDRP.fasta
s=107_RDRP_Orthornavirae
o=Output_Orthornavirae
t=Totiviridae,Amalgaviridae,Partitiviridae,Leviviridae,Mi
toviridae,Botourmiaviridae,Narnaviridae
```

Alternatively, we can execute MPACT with exactly the same parameters and options using the line command. Bellow you can find the commands for both python script and docker image:

Python script:

```
python3 mpact.py -i Orthornavirae_107_RDRP.fasta -s
107_RDRP_Orthornavirae -o Output_Orthornavirae -t
Totiviridae,Amalgaviridae,Partitiviridae,Leviviridae,Mitovirid
ae,Botourmiaviridae,Narnaviridae
```

Docker image:

```
docker run -v <files_directory> :/mnt mpact(no_fs) -I /mnt/
Orthornavirae_107_RDRP.fasta -s /mnt/107_RDRP_Orthornavirae -o
/mnt/ Output_Orthornavirae -t
Totiviridae,Amalgaviridae,Partitiviridae,Leviviridae,Mitovirid
ae,Botourmiaviridae,Narnaviridae
```

6.3.8 Understanding the parameters

Similar to the Microviridae analysis, in this analysis all the five available metrics are performed: identity (0), similarity (1) and maximum-likelihood distance (2) of the sequences from Orthornavirae_107_RDRP.fasta file (-i) and TM-scores (3) and 3Di characters similarity (4) of the structures from 107_RDRP_Orthornavirae directory (-s). The main output directory is named `Output_Orthornavirae`. The **-t** parameter specifies the prefixes for protein sequences and 3D protein structures, which will have defined maximum and minimum values, as well as intra and intergroup values within their group.

6.3.9 Inspecting the output files

The files and directories in the output directory (Output_Orthornavirae) are the same as those explained in item 6.3.3. In addition, based on the prefixes provided in that run by the **-t**

parameter, the file metrics_ranges_table.csv is also created. This file contains a table of the maximum and minimum, intra and intergroup values for the groups defined by the prefixes of the **-t** parameter, in CSV format.

Additionally, in the last line of the table, the mean and standard deviation for all metrics are displayed. These statistics are also shown in the logfile. The content of this file is

:

```
Prefix;AA identity %;AA similarity %;ML distance;TM-score;3Di similarity %
Totiviridae;;;;;
intragroup;1.7-95.9;3.8-97.4;0.0094086-4.8496564;0.13676-0.97336;2.7-89.3
intergroup;0.1-19.4;0.3-33.2;3.2949151-7.9910929;0.13901-0.60534;1.2-49.5
Amalgaviridae;;;;;
intragroup;10.4-71.7;17.0-81.9;0.3988024-4.5451336;0.29309-0.94959;26.7-91.7
intergroup;0.1-20.0;0.3-34.2;2.8784341-7.7397575;0.15456-0.78105;1.2-58.5
Partitiviridae;;;;;
intragroup;12.6-84.4;23.1-91.7;0.1809728-4.0757964;0.57348-0.99037;41.3-95.3
intergroup;0.0-20.0;0.1-34.2;2.8784341-7.6867443;0.15187-0.78105;4.9-58.5
Leviviridae;;;;;
intragroup;49.9-49.9;64.0-64.0;0.8029433-0.8029433;0.934-0.934;83.6-83.6
intergroup;1.1-18.5;2.3-33.1;3.2039968-6.2764277;0.15244-0.60926;4.0-47.6
Mitoviridae;;;;;
intragroup;16.7-95.5;27.0-98.7;1e-06-3.269462;0.42156-0.97813;40.7-91.2
intergroup;0.1-20.6;0.5-33.6;3.2671393-7.9910929;0.14921-0.62616;17.8-51.7
Botourmiaviridae;;;;;
intragroup;13.0-47.6;21.6-60.7;0.9411408-4.1648931;0.47349-0.8686;28.4-77.9
intergroup;0.3-21.4;0.3-33.6;3.0208689-6.9254761;0.17212-0.70271;4.1-53.4
Narnaviridae;;;;;
intragroup;10.2-96.8;14.6-98.0;0.0189144-3.361849;0.32626-0.90106;19.3-86.3
intergroup;0.0-21.4;0.1-33.4;3.0208689-7.5811115;0.13901-0.70271;2.8-53.4
Mean+-STD;13.5325+-7.5387;22.0808+-10.117;22.0808+-10.117;4.6345+-1.173;38.7478+-10.4613
```