## Introduction

MfcCluster.exe is a sample application able to load a sequence fasta file and cluster and visualize its content using different algorithms and settings.

## Implementation

The fMLC software is implemented in C++ and supports multi-threaded parallelism. fMLC takes DNA sequences in a FASTA file and a list of sequence similarity thresholds to perform clustering at multiple levels using e.g. BLAST-based sequence comparisons (Altschul *et al.* 1997). At each level, if the number of sequences is smaller than a given number *M*, the sequences are clustered using the connected component-based clustering (CCBC) by Bolten *et al.* (2001). Otherwise, the sequences are first divided into blocks and then clustered per block using the greedy clustering (GC) by Edgar (2010). Next, the representative sequences of the groups are re-clustered with CCBC and, finally, each sequence is assigned to the group of its representative sequence (Vu *et al.*, 2014). For the Windows version of fMLC, we used our own implementation of BLAST as it has been optimized for an in-house developed software managing all the databases at the Westerdijk institute.

Further, fMLC is integrated with an interactive web-based tool called DiVE to visualize the resulting DNA sequences-based “embeddings” in 2D or 3D. Specifically, DiVE is written in Javascript and makes use of freely available libraries (*graphosaurus.js* and *three.js*) to ensure portability across web browsers. In the visualization, data points (which denote DNA sequences) can be colored according to the biological properties (separated by the character “|”) in the FASTA sequence headers. Moreover, the data points can be further inspected for metadata and/or filtered using advanced (boolean) search functionality. Data visualization using DiVE involves the following steps: i) cluster data (e.g. DNA sequences), ii) save the matrices (CSM or SSM) obtained through sequence comparisons in the first step, iii) compute 2D/3D coordinates for each data point from the matrix using LargeVis and iv) visualize the data embeddings in 2D/3D. A SSM is obtained by comparing only the representative sequences of the groups at each level and the sequences of each group at the final level of fMLC. To improve the accuracy of LargeVis, the sequences of a group whose similarity score is less than the clustering threshold, are recompared. The maximum number of these sequences is , where *K* is the *K*-neighbor number specified for LargeVis and *n* is the number of the sequences.

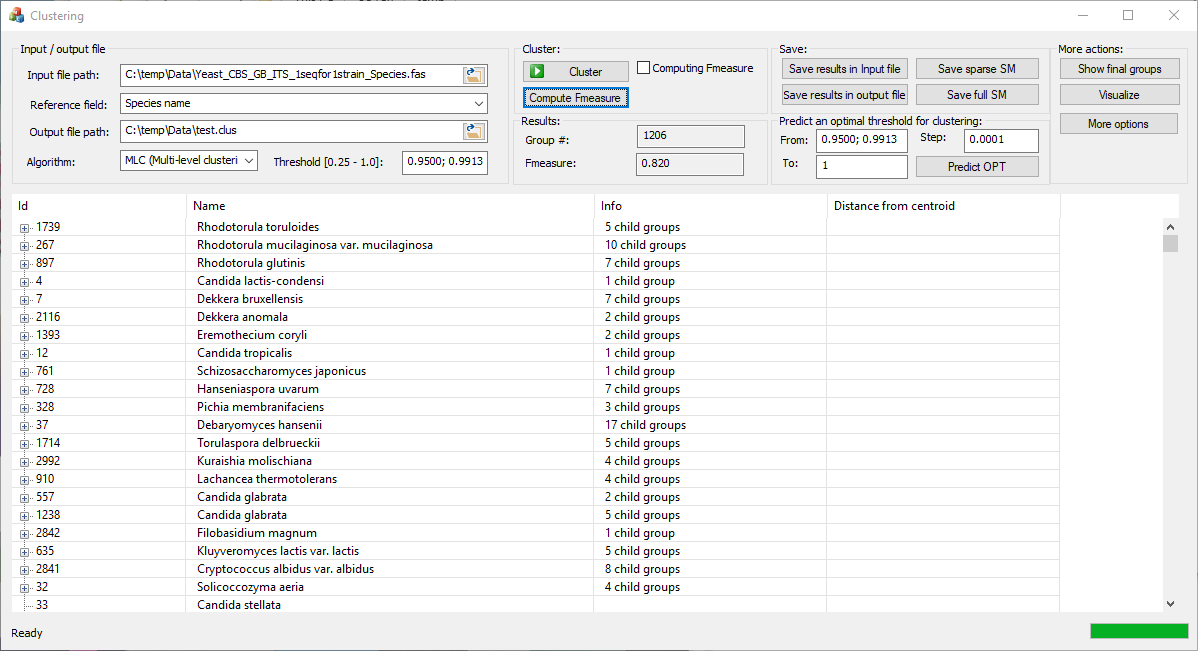
## Compiling

Dependencies:

* Boot 1.60.0 (http://www.boost.org/users/history/version\_1\_60\_0.html)
* Eigen323 (http://eigen.tuxfamily.org/dox-3.2/)
* DiVE
* LargeVis/compute\_coordinates.exe (https://github.com/lferry007/LargeVis)
* BioScience.x64.dll

The folders LargeVis and DIVE and the file BioScience.x64.dll should be put in the same folder where the application file MfcCluster.exe is.

## The main windows



### Input/output group*:*

### *Input file path*: The path of the fasta file of sequences to be clustered. In this fasta file, a sequence is represented by two lines:

* The title line starting with character ">" containing multiple information fields separated by the pipe character "|". The first information field is the index of the sequence starting at 1.
* the second line contains the sequence.

Another input file with the same name of the fasta file in the format .title describing the information fields can be given optionally. The two input files Yeast\_CBS\_GB\_ITS\_1seqfor1strain\_Species.fas and Yeast\_CBS\_GB\_ITS\_1seqfor1strain\_Species.title in the Data folder can be seen as given examples.

*Reference field***:** When the input file is given, the drop down list on second position will display all possible fields found between the pipe characters. If the .title file is given, then the drop down list will display the information given in this file, otherwise, it displays the information given in the first sequence of the fasta file.

### *Output file path*: The path to save the clustering result, or to save the prediction of the optimal threshold to cluster the given dataset.

### *Algorithms*: The algorithm used to cluster. There are three clustering algorithms to be selected for clustering: MLC (multilevel clustering, Vu *et al*. 2014), CCBC (connected components based cluster, Bolten *et al*. 2001) and GC (the greedy clustering, Edgar 2010) using multi threads or single threads.:

### *Thresholds:* For CCBC and GC, this is a threshold to cluster the dataset. Its value is in between 0 and 1. For MLC, this is a list of increasing thresholds between 0 and 1. The final threshold of the list is the actual threshold that we want to cluster the dataset with. For example, the thresholds to cluster a dataset can be 0.95;0.98. MLC will first cluster the dataset with the threshold of 0.95, and then the obtained groups will be clustered with the threshold of 0.98.

### The Cluster group

### *Cluster*: Cluster the given dataset with the selected algorithm and thresholds.

### *Computing Fmeasure checkbox*: Compute Fmeasure immediately after clustering.

### *Compute Fmeasure*: Compute Fmeasure based on the clustering result.

### The Results group

### *Group #*:The group number obtained after clustering.

### *Fmeasure*:The Fmeasure (Paccanaro *et al.* 2006) obtained by comparing the clustering result with the classification of the given dataset based on the selected field.

### *The bottom grid*:The grid at the bottom of the window displays the obtained clusters after clustering.

### The Save group

### *Save result in input file*:The titles of the sequences in the input file will be extended with the centrality indexes of the groups that the sequences belong to at each level.

### *Save result in output file*:The clustering result are saved in the output file in tab delimited format.

### *Save sparse SM*:Save a sparse similarity matrix based on the clustering result.

### *Save full SM*:Save a complete similarity matrix.

### Predict an optimal threshold for clustering

### *From*: The lower boundary threshold for the prediction.

### *To*:The upper boundary threshold for the prediction.

### *Step*:The incremental step of the thresholds in the prediction.

### *Predict OPT*: To find an optimal threshold between the lower and upper boundaries that produces the best Fmeasure for clustering.

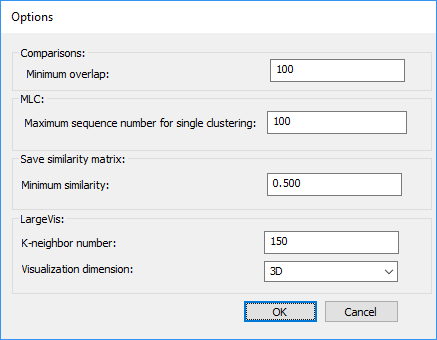
### The More actions group

### *Show final groups*:Display only the grouping at the final level of MLC.

### *Visualize*:Visualize the dataset using DiVE. For this action, a sparse/full similarity matrix is required.

### *More options*:Open an option windows to modify parameters used in clustering, saving similarity matrix and visualizing.

## The options windows



### *Minimum overlap*:This parameter is used to recompute the similarity score between two sequences if the overlap obtained by BLAST when aligning them is shorter than this value (see Vu *et al*. 2014).

### *Minimum sequence number for MLC*:The minimum sequence number for that MLC can be applied.

### *Minimum similarity*:The minimum similarity score to be saved for a sparse or full similarity matrix.

### *K-neighbor number*: This K-neighbor number parameter set up for LargeVis. It’s default value is 150. The remaining parameters of LargeVis are set as default.

### *Visualization dimension*: 2D or 3D.

## References

Bolten, E., Schliep, A., Schneckener, S., Schomburg D. & Schrader, R (2001). Clustering protein sequences- structure prediction by transitive homology. Bioinformatics 17, 935-941.

Edgar, R.C (2010). Search and clustering orders of magnitude faster than BLAST. Bioinformatics 26, 2460-2461.

Paccanaro, P., Casbon, J.A. & Saqi, M.A (2006). Spectral clustering of proteins sequences. Nucleic Acids Res 34, 1571.

Vu D. et al. (2014). Massive fungal biodiversity data re-annotation with multi-level clustering. Scientific Reports 4: 6837.