# DendroUPGMA: A dendrogram construction utility

http://genomes.urv.es/UPGMA/

# Tutorial

Santi Garcia-Vallvé
Cheminformatics and Nutrition Research Group
http://www.cheminformatics-nutrition.recerca.urv.cat/
Biochemistry and Biotechnology Department
Rovira i Virgili University (URV)
Tarragona, Catalonia, Spain

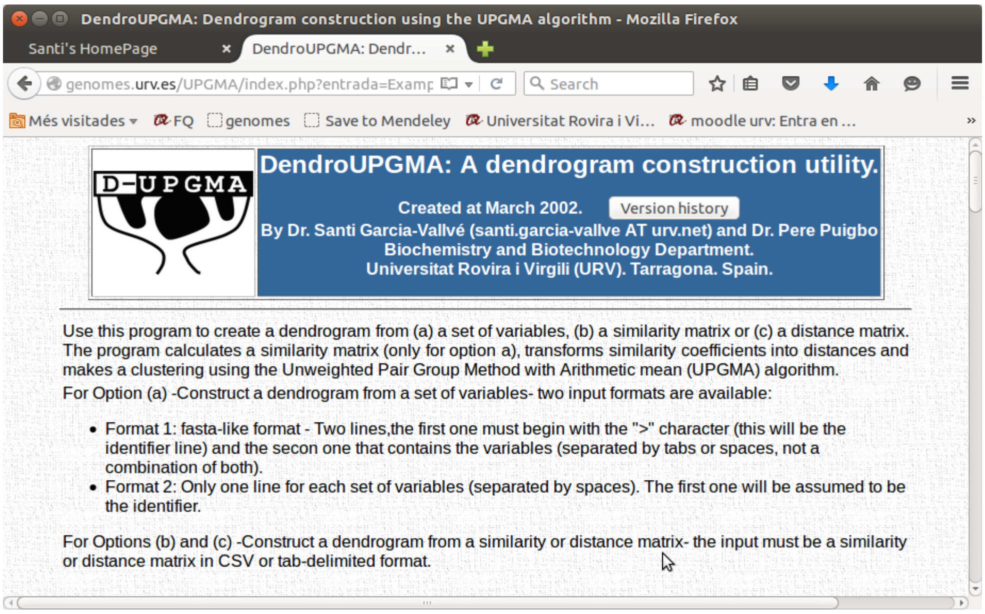
Pere Puigbò
Turku Collegium for Science and Medicine
Department of Biology
University of Turku
Turku, Finland

## Table of contents

# Introduction

## What is DendroUPGMA?

**DendroUPGMA** is a web server, freely available at http://genomes.urv.es/UPGMA/, that allows the construction of dendrograms, using the UPGMA (Unweighted Pair Group Method with Arithmetic Mean) or WPGMA (Weighted Pair Group Method with Arithmetic Mean) algorithm.


DendroUPGMA: Dendrogram construction using the UPGMA algorithm - Mozilla Firefox
Santi's HomePage
DendroUPGMA: Dendr...
genomes.urv.es/UPGMA/index.php?entrada=Examp
Search
Més visitades
FQ
genomes
Save to Mendeley
Universitat Rovira i Vi...
moodle urv: Entra en ...
D-UPGMA
Version history

DendroUPGMA: A dendrogram construction utility.

Created at March 2002.

By Dr. Santi Garcia-Vallvé (santi.garcia-vallve AT urv.net) and Dr. Pere Puigbo
Biochemistry and Biotechnology Department.
Universitat Rovira i Virgili (URV). Tarragona. Spain.

Use this program to create a dendrogram from (a) a set of variables, (b) a similarity matrix or (c) a distance matrix. The program calculates a similarity matrix (only for option a), transforms similarity coefficients into distances and makes a clustering using the Unweighted Pair Group Method with Arithmetic mean (UPGMA) algorithm.

For Option (a) -Construct a dendrogram from a set of variables- two input formats are available:

- Format 1: fasta-like format - Two lines,the first one must begin with the ">" character (this will be the identifier line) and the secon one that contains the variables (separated by tabs or spaces, not a combination of both).
- Format 2: Only one line for each set of variables (separated by spaces). The first one will be assumed to be the identifier.

For Options (b) and (c) -Construct a dendrogram from a similarity or distance matrix- the input must be a similarity or distance matrix in CSV or tab-delimited format.

A **dendrogram** (from the Greek *dendron* “tree” and *gramma* “drawing”) is a diagram frequently used to illustrate the arrangement of different clusters produced by hierarchical clustering. It consists of U-shaped lines that connect data points in a hierarchical tree. The length of lines represents the distance between two data points that are connected. A dendrogram is not, however, an phylogenetic tree because it does not show evolutionary information.

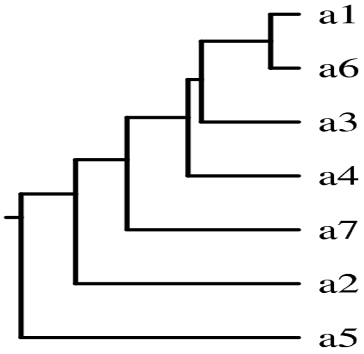


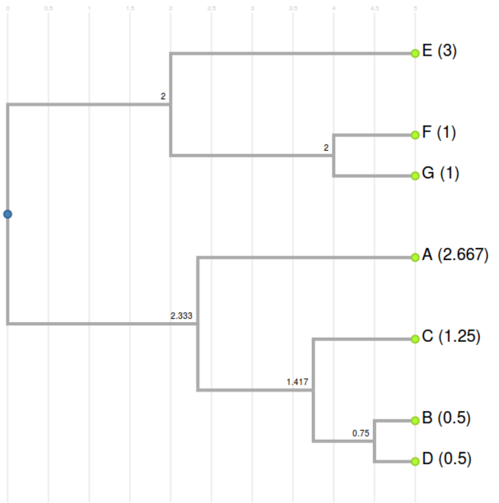
Figure 1. Example of a dendrogram.

The **UPGMA** (**U**nweighted **P**air **G**roup **M**ethod with **A**rithmetic Mean) method (Sneath and Sokal, 1973) is a simple agglomerative hierarchical clustering method to produce a dendrogram from a distance matrix. The UPGMA method employs a sequential clustering algorithm, in which local topological relationships are inferred in order of decreasing similarity and a dendrogram is built in a stepwise manner. That is, first the two closest data points are identified and grouped in the dendrogram. After the first clustering, the two closest data points are treated as a single data point (composite) and new distances are computed using the average of the distances between a simple data point and the constituents of the composite data point. Then the next closest data points are added to the dendrogram until all data points are included. The WPGMA (Weighted Pair Group Method with Arithmetic Mean) algorithm is similar to its unweighted variant, the UPGMA algorithm. At the WPGMA algorithm, the distance between clusters is calculated as a simple average. At the UPGMA algorithm the averages are weighted by the number of taxa in each cluster at each step. Note that the unweighted term indicates that all distances contribute equally to each average that is computed and does not refer to the math by which it is achieved. Thus the simple averaging in WPGMA produces a weighted result and the proportional averaging in UPGMA produces an unweighted result.

Consider the following distance matrix:

| | A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|---|
| A | 0 | 4 | 7 | 5 | 13 | 8 | 6 |
| B | | 0 | 3 | 1 | 9 | 12 | 10 |
| C | | | 0 | 2 | 8 | 13 | 11 |
| D | | | | 0 | 6 | 11 | 13 |
| E | | | | | 0 | 5 | 7 |
| F | | | | | | 0 | 2 |
| G | | | | | | | 0 |

Application of the UPGMA method gives the following dendrogram representation:



Note that the sum of the paths between B and D (0.5 + 0.5) corresponds to the B-D distance (1) in the matrix. Adapted from Example 5.7 from Lesk 2014.

# Input of the program

The input of the DendroUPGMA server may be:

- sets of variables in a variety of formats. The DendroUPGMA server calls the number of sets of variables as rows. The number of variables are the number of columns. The data could be in fasta-like format, *i.e.* a first line that begins with the “>” character and contains the identifier field, and a second line that contains a set of variables separated by tabs or spaces (but not a combination of both). In addition, only one line for each set of variables (separated by spaces) can be used. In this case, the first variable is assumed to be the identifier field.

- a similarity or a distance matrix in CSV or tab-delimited format.

Examples of input formats recognized by the DendroUPGMA server:

Fasta-like format with 4 rows, each of them with 8 variables:

```
>D45211¶
0.87    1.48    0.78    0.87    0.50    1.50    0.93    1.07¶
>U12334¶
0.64    2.02    0.97    0.37    0.50    1.50    1.17    0.83¶
>D00834¶
0.53    2.19    0.53    0.75 0.50       1.50    0.39    1.61¶
>D23668¶
0.00    1.35    2.46    0.18    0.00    2.00    0.00    2.00¶
```

One line for each row:

```
a1 1 1 1 1 1 1 1 1 1 1¶
a2 0 0 0 0 0 0 1 1 0 0¶
a3 1 1 1 0 1 1 0 1 0 1¶
a4 1 1 1 0 1 1 0 1 1 1¶
```

The first variable is assumed to be the identifier field.

Similarity matrix in tab-delimited format:

```
  →    a1→   a2→   a3→   a4→   a5→   a6→   a7→   a8→   a9→   a10¶
 a1→   1→    0.20→ 0.75→ 0.70→ 0.11→ 0.90→ 0.50→ 0.71→ 0.64→ 0.70¶
 a2→   →     1→    0.12→ 0.28→ 0.30→ 0.22→ 0.16→ 0.28→ 0.14→ 0.12¶
 a3→   →     →     1→    0.55→ 0.50→ 0.87→ 0.34→ 0.40→ 0.44→ 0.55¶
 a4→   →     →     →     1→    0.10→ 0.60→ 0.33→ 0.55→ 0.85→ 0.55¶
 a5→   →     →     →     →     1→    0.20→ 0.40→ 0.10→ 0.44→ 0.58¶
 a6→   →     →     →     →     →     1→    0.55→ 0.77→ 0.50→ 0.60¶
 a7→   →     →     →     →     →     →     1→    0.71→ 0.37→ 0.71¶
 a8→   →     →     →     →     →     →     →     1→    0.44→ 0.55¶
 a9→   →     →     →     →     →     →     →     →     1→    0.62¶
a10→   →     →     →     →     →     →     →     →     →     1¶
```

Distance matrix in tab-delimited format:

```
 →     a1→   a2→    a3→    a4→    a5→    a6→    a7→    a8→    a9→    a10¶
a1→    0→    0.80→  0.30→  0.38→  1.00→  0.10→  0.50→  0.30→  0.40→  0.30¶
a2→    →     0→     0.87→  0.71→  1.00→  0.77→  0.83→  0.71→  0.86→  0.87¶
a3→    →     →      0→     0.44→  1.00→  0.40→  0.66→  0.60→  0.56→  0.44¶
a4→    →     →      →      0→     1.00→  0.40→  0.66→  0.44→  0.14→  0.44¶
a5→    →     →      →      →      0→     1.00→  1.00→  1.00→  1.00→  1.00¶
a6→    →     →      →      →      →      0→     0.44→  0.22→  0.50→  0.40¶
a7→    →     →      →      →      →      →      0→     0.29→  0.62→  0.29¶
a8→    →     →      →      →      →      →      →      0→     0.56→  0.44¶
a9→    →     →      →      →      →      →      →      →      0→     0.37¶
a10→   →     →      →      →      →      →      →      →      →      0
```

After reading the input, a series of checkups are done. This includes the checkup that variables are separated by tabs or spaces (not a mix of both), that all the sets of variables have the same number of variables (and it is higher than 3) and that the identifier names are not repeated. As a reference, use only alphanumeric characters (alphabetic and numerical characters) as identifier names.

From the sets of variables the DendroUPGMA server calculates a similarity matrix and then a distance matrix (calculated as 1 - similarity matrix). This distance matrix is the input of the UPGMA algorithm. The calculation of the similarity matrix from the input consisting of sets of variables can be done with several coefficients:

- The **Pearson coefficient**. It is a measure of the linear correlation between two sets of variables. It gives a value between +1 and -1, inclusive, where 1 corresponds to a complete positive correlation.

- **Jaccard** similarity coefficient, also known as Tanimoto coefficient. It measures the similarity between two sets of binary data. It is defined as the size of the intersection divided by the size of the union of the sample sets. It gives a value between 0 and 1.

$$\text{Jaccard similarity coefficient: } S_{AB} = \frac{c}{a+b-c}$$

a is defined as the number of bits set to “1” in A, b as the numbers of bits set to “1” in B and c as the numbers of bits that are “1” in both A and B

- **Dice** coefficient. It measures the similarity between two sets of binary data and it ranges from 0 to 1. It is similar to the Jaccard coefficient, but gives twice the weight to agreements.

$$\text{Dice coefficient: } S_{AB}=\frac{2c}{a+b}$$

a is defined as the number of bits set to "1" in A, b as the numbers of bits set to "1" in B and c as the numbers of bits that are "1" in both A and B

In addition, four distances can be used to construct a distance matrix directly. In this case a similarity matrix is not calculated.

- **Euclidean** distance. It is the "ordinary" distance between two points in Euclidean space, *i.e.* it is the length of the line segment connecting them.

  For a matrix of m objects with n variables each object:

| | | variables | | | | |
|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | i | n |
| objects | 1 | $x_{11}$ | $x_{12}$ | $x_{13}$ | $x_{1i}$ | $x_{1n}$ |
| | 2 | $x_{21}$ | $x_{22}$ | $x_{23}$ | $x_{2i}$ | $x_{2n}$ |
| | 3 | $x_{31}$ | $x_{32}$ | $x_{33}$ | $x_{3i}$ | $x_{3n}$ |
| | j | $x_{j1}$ | $x_{j2}$ | $x_{j3}$ | $x_{ji}$ | $x_{jn}$ |
| | k | $x_{k1}$ | $x_{k2}$ | $x_{k3}$ | $x_{ki}$ | $x_{kn}$ |
| | m | $x_{m1}$ | $x_{m2}$ | $x_{m3}$ | $x_{mi}$ | $x_{mn}$ |
| | | $\overline{X}_1$ | $\overline{X}_2$ | $\overline{X}_3$ | $\overline{X}_i$ | $\overline{X}_n$ |
| | | $s_1$ | $s_2$ | $s_3$ | $s_i$ | $s_n$ |

  The Euclidean distance between j and k objects is:

$$Euclidean\,distance_{jk}=\sqrt{\sum_{i=1}^{n}(x_{ji}-x_{ki})^2}$$

- **Manhattan** metric, also known as city block or taxicab distance. It is the sum of absolute differences. The names of this metric reflect the fact that in a perfectly regular street system, we cannot always get from one point to another along a straight line and one has to walk round the blocks.

  For the above matrix, the Manhattan distance between the j and k objects is:

$$Manhattan\,distance_{jk}=\sum_{i=1}^{n}|x_{ji}-x_{ki}|$$

- **Root Mean Square Deviation** (RMSD). It represents the sample standard deviation of differences between two set of variables. It is frequently used in Cheminformatics to compare the distance between two conformers.

  For the above matrix, the RMSD value between the j and k objects is:

$$RMSD_{jk}=\sqrt{\frac{\sum_{i=1}^{n}(x_{ji}-x_{ki})^2}{n}}$$

- **Mean Square Deviation** (MSD). It is analogous to RMSD, but the square root is not applied.

  For the above matrix, the MSD value between the j and k objects is:

$$MSD_{jk}=\frac{\sum_{i=1}^{n}(x_{ji}-x_{ki})^2}{n}$$

Prior to the calculation of any of the above four distances, the input data can be normalized by adjusting values measured on different scales to a common scale. The **normalization** is achieved by replacing each data point with the Student's t-statistic, calculated by subtracting the mean and dividing by the standard deviation of each variable (column). This statistic measures the sigma distance of actual data from the average (*i.e.* the number of standard deviations by which an observation or data is above or below the mean).

$$Student's\,t\text{-}statistic=\frac{(x_{ji}-\overline{X}_i)}{s_i}$$

When the input consists of sets of variables, it is possible to calculate 100 **bootstrap replicates**. Bootstrap is a statistical test that relies on random sampling with replacement that is widely used in phylogenetic tree construction. When this option is selected, the DendroUPGMA server samples the input variables randomly (and allowing replacement) to create a pseudoreplicate of the input. Each pseudoreplicate contains the same number of elements as the input, but some original variables are represented more than once and some not at all. This process is repeated 100 times, and the number of pseudoreplicates that contain each cluster is counted and represented in the final dendrogram. The bootstrap values give therefore an estimation of whether a particular node is supported by the input data. If you get a low bootstrap value for a node, this suggests that only a few variables of the input data support that node, as removing variables at random leads to different clusters. Bootstrap, therefore, only makes sense if there is a sufficient number of variables in the input data.

Original dataset:

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| A | 0.8 | 1.4 | 0.7 | 0.8 | 0.5 | 1.5 | 0.9 | 1.0¶ |
| B | 0.6 | 2.0 | 0.9 | 0.3 | 0.5 | 1.5 | 1.1 | 0.8¶ |
| C | 0.5 | 2.1 | 0.5 | 0.7 | 0.5 | 1.5 | 0.3 | 1.6¶ |
| D | 0.0 | 1.3 | 2.4 | 0.1 | 0.0 | 2.0 | 0.0 | 2.0¶ |

Bootstrap Pseudoreplicates:

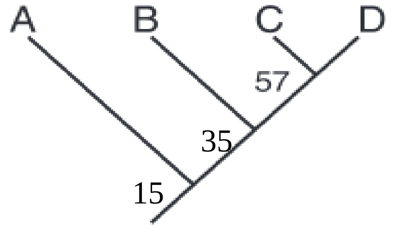
Bootstrap Pseudoreplicate 1:

| | 7 | 4 | 3 | 7 | 5 | 1 | 7 | 2 |
|---|---|---|---|---|---|---|---|---|
| A | 0.9 | 0.8 | 0.7 | 0.9 | 0.5 | 0.8 | 0.9 | 1.4¶ |
| B | 1.1 | 0.3 | 0.9 | 1.1 | 0.5 | 0.6 | 1.1 | 2.0¶ |
| C | 0.3 | 0.7 | 0.5 | 0.3 | 0.5 | 0.5 | 0.3 | 2.1¶ |
| D | 0.0 | 0.1 | 2.4 | 0.0 | 0.0 | 0.0 | 0.0 | 1.3¶ |

Bootstrap Pseudoreplicate 2:

| | 6 | 1 | 8 | 4 | 2 | 1 | 7 | 7 |
|---|---|---|---|---|---|---|---|---|
| A | 1.5 | 0.8 | 1.0 | 0.8 | 1.4 | 0.8 | 0.9 | 0.9¶ |
| B | 1.5 | 0.6 | 0.8 | 0.3 | 2.0 | 0.6 | 1.1 | 1.1¶ |
| C | 1.5 | 0.5 | 1.6 | 0.7 | 2.1 | 0.5 | 0.3 | 0.3¶ |
| D | 2.0 | 0.0 | 2.0 | 0.1 | 1.3 | 0.0 | 0.0 | 0.0¶ |

Dendrogram with bootstrap values constructed from 100 pseudoreplicates:



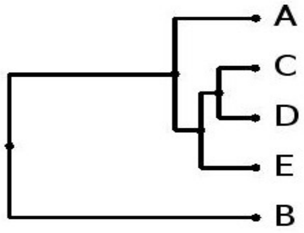
To simplify the output dendrogram, the DendroUPGMA server includes an option to remove the rows, all but one, that are identical or have a similarity of “1” in the similarity matrix.

## Outputs

The outputs of the DendroUPGMA server include:

- A summary of the number of rows and variables read by the server.
- The similarity matrix and the method used to calculate it. For the RMSD and MSD methods, and if the input is already a distance matrix, this step is omitted.
- The distance matrix used to create the dendrogram.
- The dendrogram in Newick format. The Newick format is a way of representing dendrograms and phylogenetic trees as a text, using parentheses and commas. This format is recognized by all the tree-visualization programs.

  Example: The following dendrogram in Newick format
  ((A:0.138,((C:0.062,D:0.062):0.031,E:0.094):0.044):0.282,B:0.420);
  corresponds to the following dendrogram:

- In addition to the dendrogram in Newick format, the DendroUPGMA server draws the dendrogram using a javascript code that uses the d3 (http://d3js.org/) and newick.js (https://github.com/jasondavies/newick.js) modules.
- The Cophenetic Correlation Coefficient (CP). It is a measure of how faithfully a dendrogram preserves the pairwise distances between the original unmodeled data points. It gives a value between 0 and 1, where 1 represents a perfect match. This coefficient is only calculated when the number of rows is lower than a specific value (see the version history link of the server at http://genomes.urv.es/UPGMA/versions.html for changes in this value).

# Examples of use

The DendroUPGMA server can be used with different types of data and with different purposes. Below we summarize some of its uses:

**1. To compare the codon usage of genes and detect horizontally transferred genes, specially for genes from bacteria and archaea.**

This was the reason why the DendroUPGMA server was developed. In this case each row represent the codon usage of a gene and the variables are the frequency (absolute or relative to the synonymous codon usage) of use of each codon. Genes from one species of bacteria or archaea often share similarities in codon frequency. Thus, genes with a different codon usage would be horizontally acquired genes (Garcia-Vallve et al. 1999; 2000).

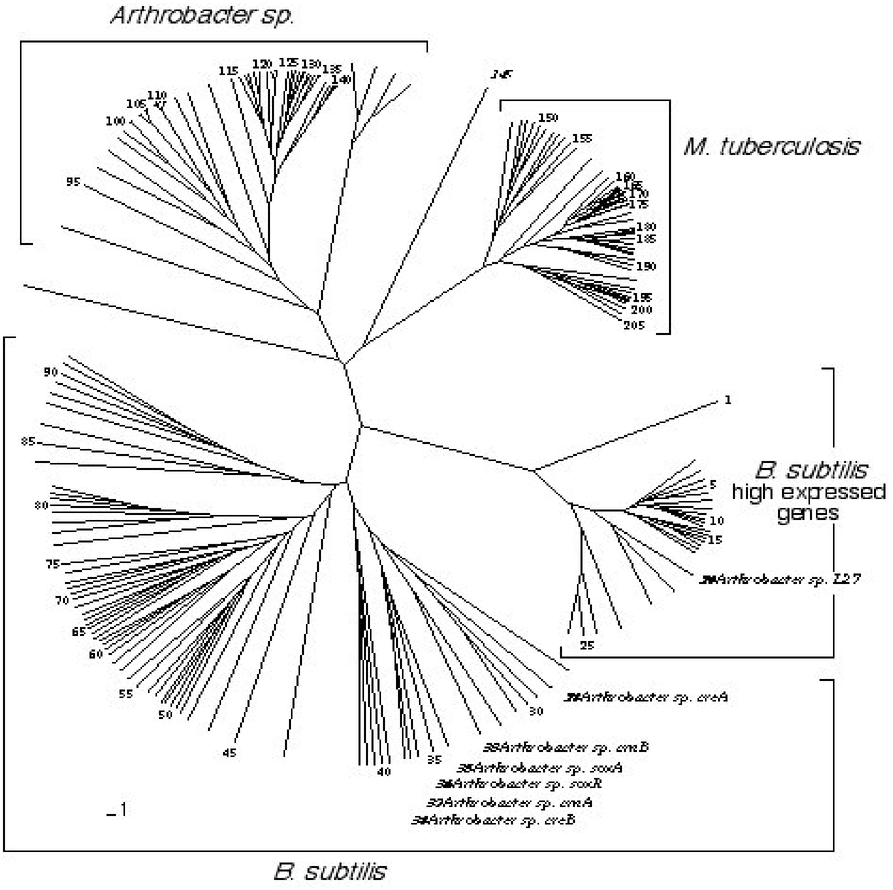


Figure 2. Dendrogram showing that a group of genes from *Arthrobacter sp.* has a codon usage similar to genes from *Bacillus subtilis* and different from other *Arthrobacter sp.* genes (Garcia-Vallvé et al. 2002).

### 2. To compare the restriction fragment length polymorphism (RFLP) or any other genomic markers for typifying or classifying strains of microorganisms.

The presence or absence of several genomic markers can be used to compare and classify different species or strains of the same species. If binary data is used, the jaccard or dice similarity coefficients are usually used.

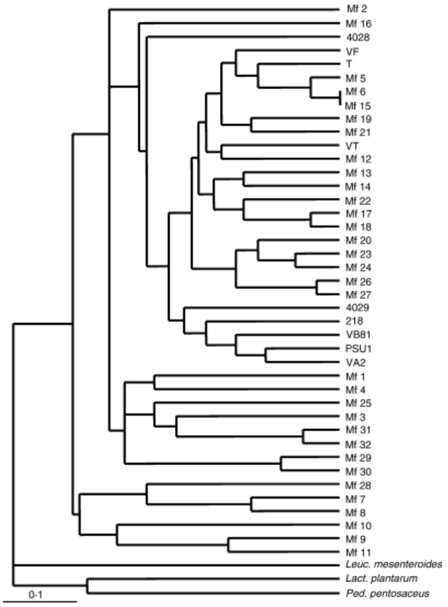


Figure 3. UPGMA dendrogram comparing the genomic profiles of different *Oenococcus oeni* strains, using the Dice similarity coefficient (figure extracted from Reguant and Bordons 2003).

## 3. To compare the chemical similarity between small molecules.

The chemical similarity between small molecules can be compared using molecular fingerprints. This involves turning the molecule into a sequence of bits, depending on the presence in the compound of certain substructures or features from a given list of structural keys or by analyzing all the fragments of the molecule (Cereto-Massague et al. 2015). The sets of bits can then be compared using the Jaccard (also known as Tanimoto coefficient) or Dice index. Similar molecules are grouped in the same cluster.

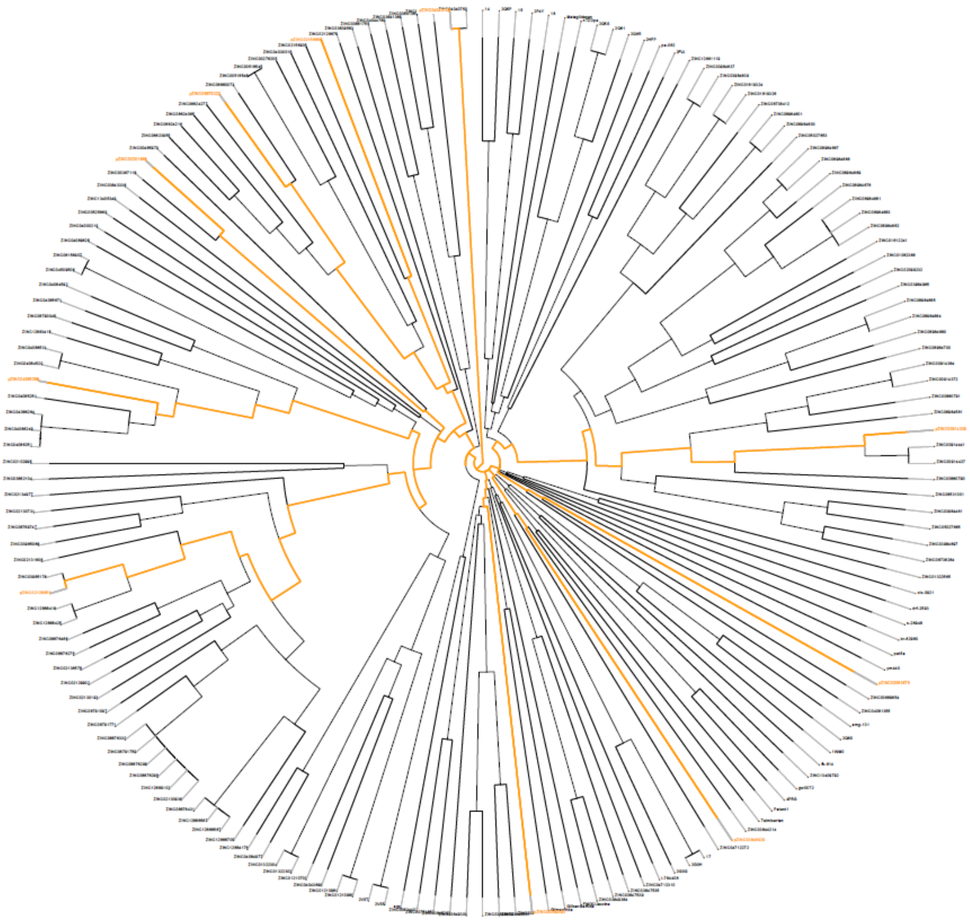


Figure 4. UPGMA dendrogram comparing the fingerprints between molecules, showing the chemical similarities between them. Brown lines shown the result molecules of a virtual screening and they are compared to known molecules with the same bioactivity (example from Guasch et al. 2012).

# References

Cereto-Massagué A, Ojeda MJ, Valls C, Mulero M, Garcia-Vallvé S, Pujadas G. 2015. Molecular fingerprint similarity search in virtual screening. Methods 71:58-63.

Garcia-Vallve, S., Palau, J. and Romeu, R. 1999. Horizontal gene transfer in glycosyl hydrolases inferred from codon usage in *Escherichia coli* and *Bacillus subtilis*. Molecular Biology and Evololution 16(9):1125-1134.

Garcia-Vallve, S., Romeu, R. and Palau, J. 2000. Horizontal gene transfer of glycosyl hydrolases of the rumen fungi. Molecular Biology and Evolution 17(3):352-361.

Garcia-Vallve S, Simo FX, Montero MA, Arola L, Romeu A. 2002. Simultaneous horizontal gene transfer of a gene coding for ribosomal protein l27 and operational genes in *Arthrobacter sp.* Journal of Molecular Evolution 55:632-637.

Guasch L, Sala E, Castell-Auví A, Cedó L, Liedl KR, Wolber G, Muehlbacher M, Mulero M, Pinent M, Ardévol A, Valls C, Pujadas G, Garcia-Vallvé S. 2012. Identification of PPARgamma partial agonists of natural origin (I): development of a virtual screening procedure and in vitro validation. PLoS One 7(11):e50816.

Lesk, A.M. 2014. Introduction to Bioinformatics. $4^{Th}$ Edition. Oxford University Press.

Reguant C. and Bordons A. 2003. Typification of Oenococcus oeni strains by multiplex RAPD-PCR and study of population dynamics during malolactic fermentation. Journal of Applied Microbiology 95(2):344-353.

Sneath, P.H. and Sokal, R.R. 1973. Numerical Taxonomy. W.H. Freeman and Company, San Francisco.