# Algorithms for Genome Rearrangements

Pedro Feijão

Lecture 11 - Tools for Real Data Analysis using Genome Rearrangements

Summer 2015

`pfeijao@cebitec.uni-bielefeld.de`

# Introduction

How do we apply models of rearrangement in real data?

1. Transform real DNA into *gene families* to allow the application of rearrangement models.
2. With a suitable representation of all the genomes (permutations, for instance), apply rearrangement models.
   1. Pairwise distances and distance-based phylogeny (z.b., Neighbour Joining)
   2. Multiple Genome Rearrangement – Find trees that minimize the number of rearrangement events

# Real Dataset

We will use 7 *Yersinia pestis* genomes. There are more than 200 strains of YP that have been sequenced:
http://www.ncbi.nlm.nih.gov/genome/153

- Yersinia pestis Antiqua

- Yersinia pestis CO92

- Yersinia pestis KIM 10

- Yersinia pestis Nepal516

- Yersinia pestis Pestoides F

- Yersinia pestis Z176003

- Yersinia pestis biovar Microtus str. 91001

Each strand has a link to the corresponding NCBI page, where the GenBank file can be downloaded.

# Alignment and detection of syntenic regions

To find the conserved regions on each genome, we will use Mauve
http://gel.ahabs.wisc.edu/mauve/
You can use the graphical interface to run the progressiveMauve
algorithm, and then export a permutation file (look at the menus, you
will find all options there). Alternatively, you can run in command-line:

- Preprocess and generate .sslist files for each .gb (GenBank) input
  file:
  progressiveMauve --mums --output=mauve-output.xmfa *gb

- Align all genomes and generate a *permutation file*:
  mauveAligner --output=all.mauve
  --permutation-matrix-output=permutation.txt
  {Antiqua,CO92,KIM10,Nepal516,PestoidesF,Z176003,
  biovar_Microtus_str_91001}.{gb,gb.sslist}

The file permutation.txt contains one line for each genome, with a
permutation from 0 to the number of blocks (79 in this case).

# Rearrangement Scenarios and Distance Phylogeny

To calculate the pairwise distance between all genomes we will use
UniMoG:
http://bibiserv.techfak.uni-bielefeld.de/dcj/welcome.html

To execute it: java -jar UniMoG.jar &

- Copy file with permutation and edit it, adjusting to UniMoG format (name of genomes, spacing, etc.)
- Run a DCJ scenario in UniMoG.
- Copy distance matrix in a file

The output contains example DCJ scenarios between all the genomes and also a distance matrix, in phylip format.

# Distance Phylogeny

We will use SplitsTree http://www.splitstree.org/.

First, we need to convert the format from

```
7
Antiqua
CO92      25
KIM_10    24 17
Nepal     22 16 11
Pestoides 31 23 22 23
Z176003   24 9 18 17 24
Biovar    36 27 28 27 32 28
```

to

```
7
Antiqua 0 25 24 22 31 24 36
CO92 25 0 17 16 23 9 27
KIM_10 24 17 0 11 22 18 28
Nepal 22 16 11 0 23 17 27
Pestoides 31 23 22 23 0 24 32
Z176003 24 9 18 17 24 0 28
Biovar 36 27 28 27 32 28 0
```

# Converting Distance matrix

There are of course several ways of doing this. I did a R-script:

```
mat <- read.table("dcj-dist.phy",fill=TRUE,col.names=paste("V",1:8),skip=1,row.names=1)
mat[is.na(mat)] <- 0
mat <- mat + t(mat)
write.table(mat, file="dcj.phy",col.names=FALSE, quotes=FALSE)
```

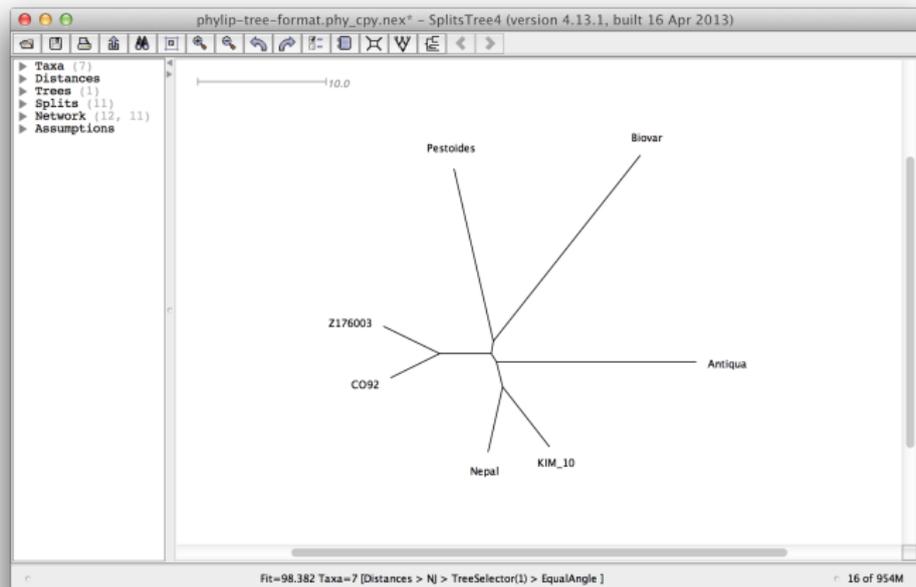Then add a first line with the number of genomes, in this example, 7.

# Phylogenetic Trees with SplitsTree

Opening the file `dcj.phy` in SplitsTree automatically generates a SplitTree representation:
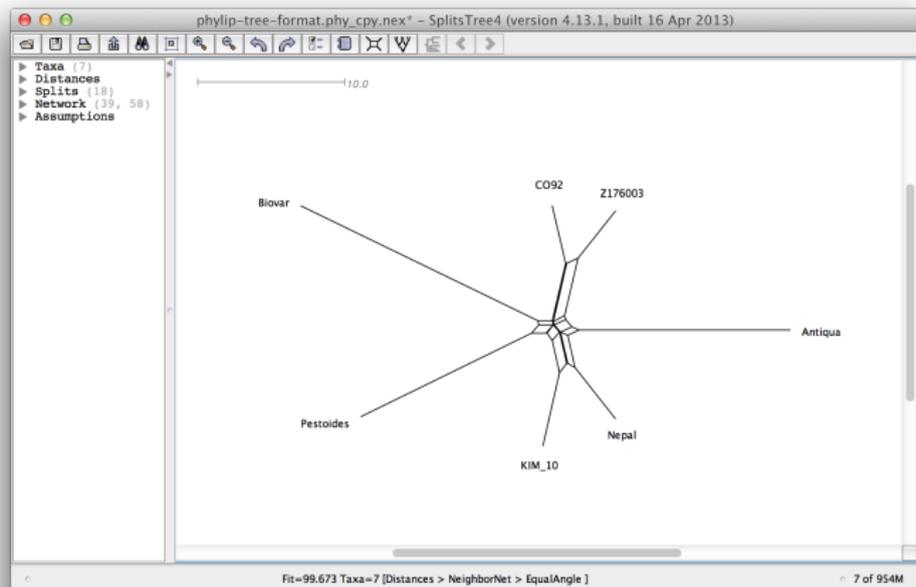
# Phylogenetic Trees with SplitsTree

In the menu `Trees/NJ` it is possible to find the Neighbour Joining tree:
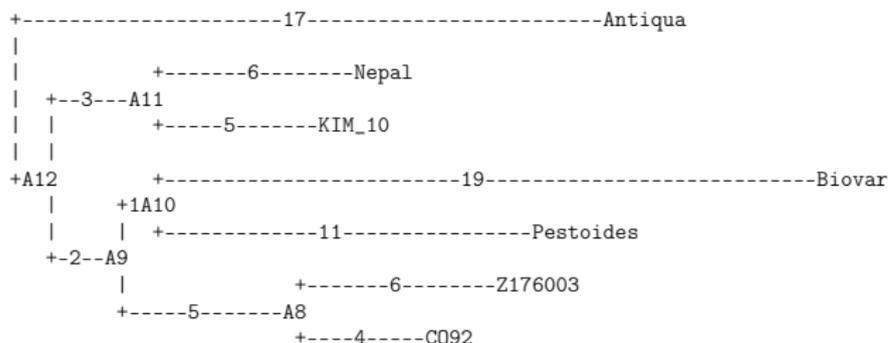
# Multiple Genome Rearrangement

- Another way to find a phylogenetic tree with rearrangement data is to solve the *Multiple Genome Rearrangement* problem.
- Since it is a NP-Hard problem, we can only do it with a small dataset. In our case, 7 genomes with +- 50 genes is small enough.
- We will use a software called MGR (yes, not very creative) http://grimm.ucsd.edu/MGR/
- The input is the same as UniMoG, but the 0's have to be removed.

```
MGR -f permutations.txt -C -H 1
```

The -C flag tells that the genomes are unichromosomal circular, and -H 1 applies an heuristic for the Median problem, solving it much faster.

# MGR Output

The output is the best tree found (not necessarily the optimal), with branch lenghts corresponding to the number of rearrangement events:

```
+----------------------17------------------------Antiqua
|
|              +-------6--------Nepal
|  +--3---A11
|  |       +-----5-------KIM_10
|  |
+A12          +-------------------------19---------------------------Biovar
   |      +1A10
   |      |  +-------------11---------------Pestoides
   +-2--A9
        |                   +-------6-------Z176003
        +-----5-------A8
                         +----4-----CO92
```

The output also contains the permutations of the internal nodes (A8, A9, ..., A12), giving a possibility for the gene arrangement for ancestral genomes.