Gene Order Analysis \2017

Practical 5 - Comp Gen 2017
Assistants: Daniel Morgan and Mateusz Kaduk

PURPOSE **1**

MATERIALS NEEDED **1**

ACTIVITY **1**

Dataset 2

Dotter 2

Reconstructing phylogeny from gene order 2

PURPOSE

Studies of gene order conservation are important to understand the evolution of genomes. In this practical you will learn how to perform basic gene-order analysis.

The report should be formatted in one PDF file and sent in by the end of the week. It should cover all points listed below. Missing or failed items will result in a reduced grade for this practical.

MATERIALS NEEDED

1. Dataset (your bacterial proteomes)
2. Dotter
3. GRIMM

ACTIVITY

Perform the following steps in this order

Ortholog groups

For this exercise you will need all ortholog groups found last week in exercise 4.

Alternatively, you can infer ortholog groups by running InParanoid for all your 4 bacterial genomes, and re-using (modifying) provided script.

1. Copy InParanoid

```
cp -R /common/courses/comparative_genomics/InParanoid/ ./
```

2. Run InParanoid one by one (not in parallel)

```
perl ./inparanoid.pl reference_genome.fa target_genome1.fa
perl ./inparanoid.pl reference_genome.fa target_genome2.fa
```

3. Adopt /common/courses/comparative_genomics/getClustersInParanoid.py to get clusters of orthologs.

Dataset

1. For the bacteria you have (the rest are probably too far distant to have retained gene order), use the information you have on which genes are homologous to create gene order lists for each genome.
2. There is a script **getGeneOrder.py** which reads in a proteome multifasta file and a cluster file from Lab 4, then outputs the gene order list for the corresponding genome.
3. If you use this script, please answer the following questions:
    a. Can there be ambiguities in clustering, so that one gene appears in several clusters in the cluster file? If so, why is this? What does this script do in that case?
    b. Can this script handle forward and reverse strandedness in gene order lists? If not, how should this be done?

Dotter

Comparing gene order between proteomes to detect change in location

4. Generate a numbered list (dictionary) of random 20aa sequences, at least as long as the highest number of lines in any of the ortholog groups files from exercise 4. (use: **rndseq.py**)
5. For each gene order file from **getGeneOrder** script, assign random sequence with corresponding number.
6. Combine all 20aa sequence for each proteome into long single sequence.
    a. Result should be one long sequence assembled from all 20aa random sequences per proteome.
7. Combine all sequences into one multi-fasta file and use it as dotter input against itself into dotter.

    a. Another script called **makeIconicGenome.py** reads in a list of random sequences (there must be more of these than there are cluster IDs) and a gene order file, and prints out the resulting pseudogenome. If you use this script explain what is it doing as short pseudo-code in your report.
    b. Using these "iconic genomes", look at pairs of genomes using the program dotter. Can you see how blocks of genes have been shuffled around at once?

Reconstructing phylogeny from gene order

8. Use a tool (such as GRIMM) to determine the genomic rearrangement distances between the species. GRIMM wants as input lists of numbers corresponding to genes, so in that case, translate your gene order lists to lists of corresponding numbers. This should give you a distance matrix.
9. it turns out that you need to change the format of the gene order lists from the previous part to get GRIMM/MGR to accept it. It only accepts input data where both sequences have exactly the same set of orthologous genes, and they must be listed from 1 upwards.

To convert your gene lists to this format, use

```
python getGeneOrderGrimm.py <max number of genes in output> <input gene
order file 1> <input gene order file 2> ... <input gene order file N>
```

for all your prokaryotic gene order lists at the same time. This will give you the output for all of them, consisting only of the genes present exactly once in each of them, and renumbered the way GRIMM/MGR wants it.

```
grimm -f grim_genomes.txt -o distance.grim -C -m
```

Use the resulting distance matrix to reconstruct the species tree in Belvu.

```
./belvu_Ubuntu_12.04.3_64bit -T R distance.txt
```

Distance matrix must be in tab delimited format, where first column are the genome names,

10. Compare the species tree to the bacterial part of your trees from practical 4. How do they differ and why?