fastGEAR – post-processing example

fastGEAR was developed by Rafal Mostowy, Nicholas J. Croucher, Cheryl P. Andam, Jukka Corander, William P. Hanage, and Pekka Marttinen*

NOTE: **Python versions** of the postprocessing scripts described in this document are available at https://github.com/shimbalama/post_fastGEAR, and they have been written by Liam McIntyre, University of Melbourne.

Contents

Introduction

fastGEAR is a software for analysing sequence alignments, and it has been described in a paper:

> Mostowy, R., Croucher, N. J., Andam, C. P., Corander, J., Hanage, W. P., & Marttinen, P. (2017). Efficient inference of recent and ancestral recombination within bacterial populations. *Molecular biology and evolution*, *34*(5), 1167-1182. https://doi.org/10.1093/molbev/msx066

The purpose of this document is to demonstrate creating some of the possible summaries from the *fastGEAR* output. See *fastGEAR_manual.pdf* for a general introduction on using *fastGEAR*. As our example we use an analysis of alignments of 54 genes, published in:

> David, S., Sanchez-Buso, L., Harris, S.R., Marttinen, P., Rusniok, C., Buchrieser, C., Harrison, T.G., Parkhill, J. (2017). Dynamics and impact of homologous recombination on the evolution of *Legionella pneumophila, PLOS Genetics*, in press. http://journals.plos.org/plosgenetics/article?id=10.1371/journal.pgen.1006855.

Note: **Python versions** of the postprocessing scripts described in this document are available at https://github.com/shimbalama/post_fastGEAR, and they have been written by Liam McIntyre, University of Melbourne.

Questions and feedback can be sent to:

pekka.marttinen@aalto.fi

Setup

The folder */m/triton/scratch/cs/mi/pemartti/Brat4/wholeGenomeAnalyses/legionella_analysis* is called in the subsequent examples as the `rootToResults`. In this folder, there are (at least) two folders *legionellaResults* and *legionellaSummaries* (Figure 1). Both of these two start with 'legionella', which is called the `datasetName` in the example. The scripts assume that these two two folders always can be found from the root folder, i.e., folders named as **Results* and **Summaries* where the asterisk * is replaced by `datasetName`.

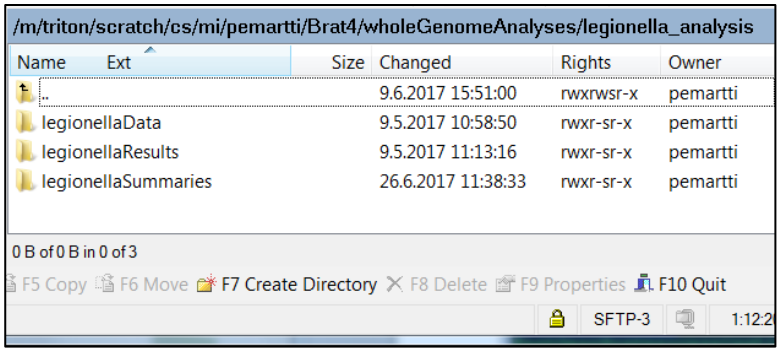
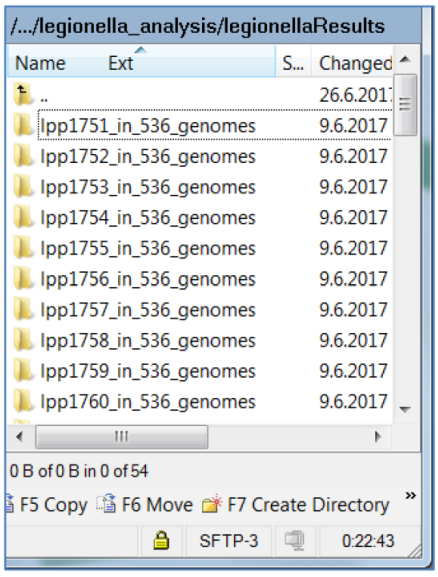


Figure 1

In the folder *legionellaResults* there are 54 folders in total, one for each gene analysed, named by the name of the gene, containing *fastGEAR* results for the gene. Example contents of this result folder are shown in Figure 2A. The first gene analysed was named as *lpp1751_in_536_genomes*, and so on. The contents of the folder for the first gene are shown in Figure 2B. It contains the *fastGEAR* output file for the gene, named as *lpp1751_in_536_genomes_res.mat*, and other output files for the gene (obtained by running fastGEAR using the alignment for the gene as the input file, and *lpp1751_in_536_genomes_res.mat* as the output file). The subsequent scripts assume results are arranged in this way, i.e., the *<datasetName>Results* contain folders for each gene analysed, named by the name of the gene, and the folder for the gene contains a result file, which named as *<geneName>_res.mat*, and other outputs for the gene.

A)



B)

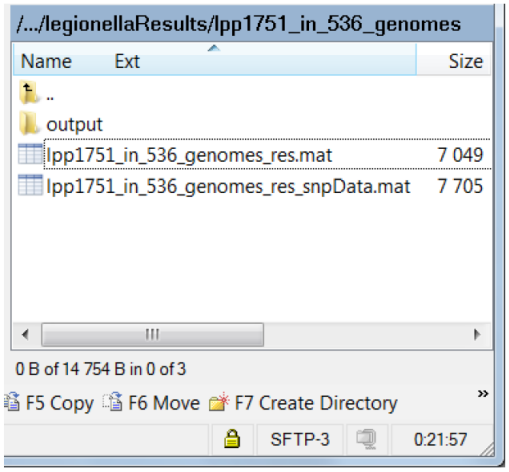


Figure 2

In *legionellaSummaries* there are files *allNamesFromTop.txt*, *subtreeNamesFromTop.txt, and* *ST1_subtree.txt* (Figure 3). The first one contains names of all 536 strains analysed, one row per strain. The second contains names in a subtree of the whole tree, a branch includuding isolates for ST1 containing 81 strains. The subsequent plotting scripts will plot the strains in the order given in these files. The third file, *ST1_subtree.tre*, contains a Newick formatted phylogeny for a subset of strains, ST1, corresponding to one branch of the tree. The scripts below assume that isolate names in any individual alignment analysed are a subset of strain names that appear in the *allNamesFromTop.txt*, i.e., if there is a strain with a certain name in some individual alignment, then that name must appear in *allNamesFromTop.txt*.

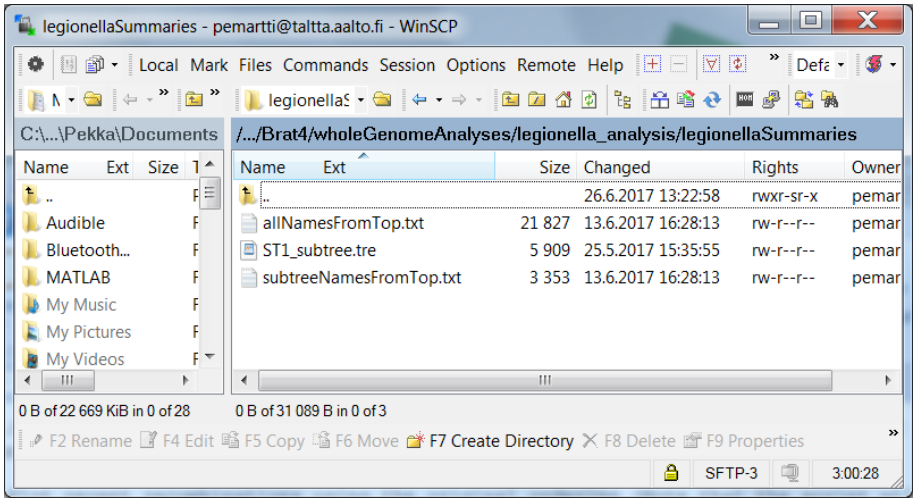


Figure 3

Reordering the cluster labels across genes

The purpose of this is to reorder cluster labels across the 54 genes such that each strains is colored with as few colors as possible, by minimising the average entropy of the color frequency distribution (see the Supplementary material of Mostowy et al. 2017 for details). This improves the visualisations considerably.

USAGE: `./run_reorderMultipleGenes.sh <path to matlab/mcr> <rootToResults> <datasetName> <fileWithOrderedStrains> <recombinationType (recent, ancestral, both)>`

EXAMPLE: `./run_reorderMultipleGenes.sh /opt/matlab2016a /m/cs/scratch/mi/pemartti/Brat4/wholeGenomeAnalyses/legionella_analysis/legionella allNamesFromTop.txt both`

Running this produces several files in the summary folder *./legionella_analysis*/legionellaSummaries, which are needed when plotting the results in the following sections.

Plotting multiple genes side by side

USAGE: `./run_plotCombinedGenes.sh <path to matlab/mcr> <rootToResults> <datasetName> <fileWithNamesOrdered>`

EXAMPLE: `./run_plotCombinedGenes.sh /opt/matlab2016a /m/cs/scratch/mi/pemartti/Brat4/wholeGenomeAnalyses/legionella_analysis/legionella allNamesFromTop.txt`

Running this opens a window showing *fastGEAR* results for the different genes side-by-side, where strains have been ordered according to the ordering given as input (Figure 4).

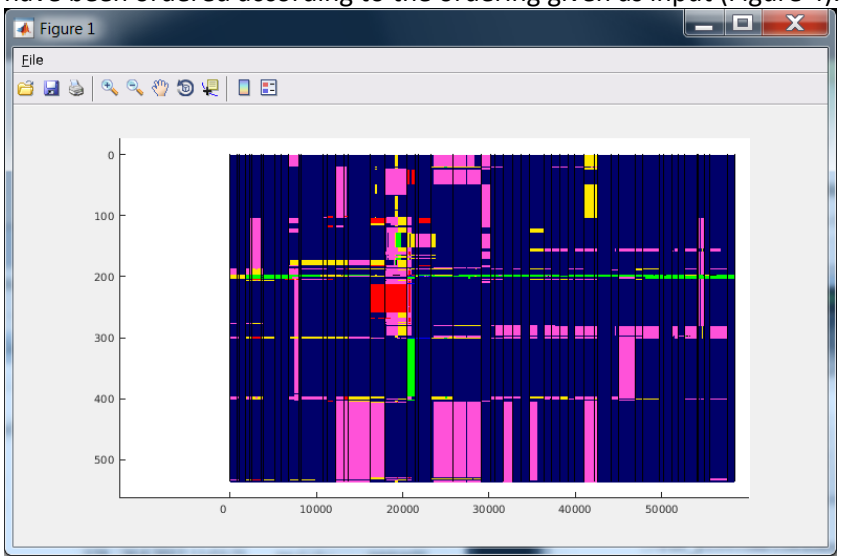


Figure 4

Inspecting colors

We want to inspect which colors in Figure 4 correspond to which lineage labels. To do this, use the '`plotColors`' function provided in the original fastGEAR package.

USAGE: `./run_plotColors.sh <path to matlab/mcr> <number of lineages, use - for the number detected)> <outputFile>`

EXAMPLE: `./run_plotColors.sh /opt/matlab2016a -/m/cs/scratch/mi/pemartti/Brat4/wholeGenomeAnalyses/legionella_analysis/legionellaSummaries/combinedPopStructure.mat`

The output is shown in Figure 5 below.

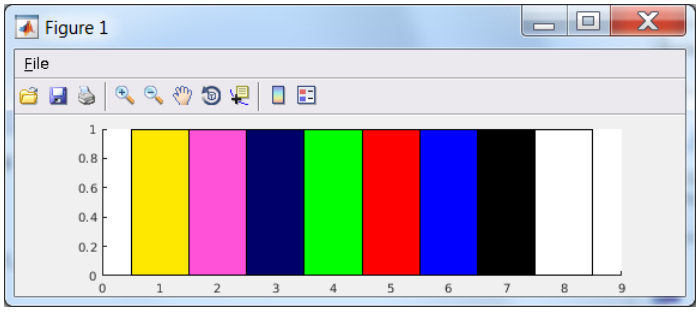


Figure 5

Plotting genes with colors in a different order

We see from Figure 5 that the most common color in Figure 4, the dark blue, corresponds to lineage label 3. We want to swap the colors of lineage labels 1 and 3, so that the most common color in Figure 4 would be yellow (which makes the figure less 'heavy'). This can be done by providing two additional parameters to the `plotCombinedGenes`

function. The first additional parameter specifies the space occupied by the alignments relative to the white space on the left side of the figure (this is needed because the white space is used in the Matlab scripts to plot a tree on the left side of the alignments. However, the tree-plotting functionality is not implemented in the compiled scripts, due to the limitations of the Matlab compiler). The second additional parameter specifies how the lineage colors should be ordered. In total there are 6 non-empty lineages (see Figure 5), and we wanted to change the colors of lineages 1 and three with each other; therefore, this parameter is specified as 3,2,1,4,5,6.

USAGE: `./run_plotCombinedGenes.sh <path to matlab/mcr> <rootToResults> <datasetName> <fileWithNamesOrdered> <alignmentWidth> <colorOrdering>`

EXAMPLE: `./run_plotCombinedGenes.sh /opt/matlab2016a /m/cs/scratch/mi/pemartti/Brat4/wholeGenomeAnalyses/legionella_analysis/legionella allNamesFromTop.txt 5 3,2,1,4,5,6`

The output is shown in Figure 6. This is exactly the same as Figure 4, except that blue and yellow colors have been swapped.

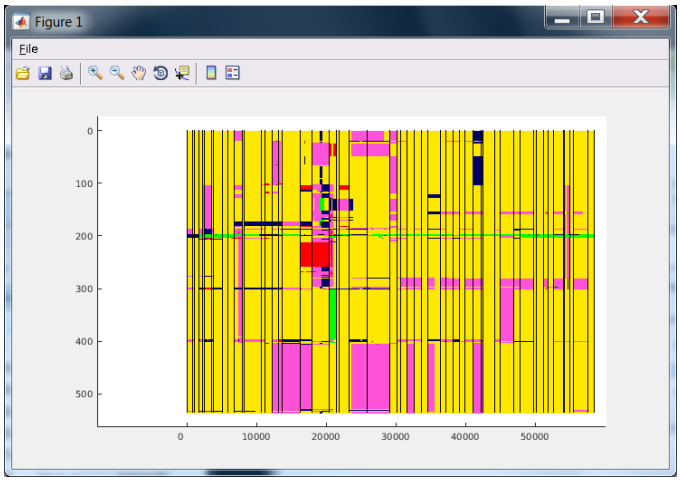


Figure 6

Plotting a single branch in the tree

This allows plotting the *fastGEAR* results for only a subset of isolates, in a given order. The function call is exactly same as with the whole set of strains, except that now the `fileWithNamesOrdered` contains names for the single branch in the tree which is to be plotted.

USAGE: `./run_plotCombinedGenes.sh <path to matlab/mcr> <rootToResults> <datasetName> <fileWithNamesOrdered> <alignmentWidth> <colorOrdering>`

EXAMPLE: `./run_plotCombinedGenes.sh /opt/matlab2016a /m/cs/scratch/mi/pemartti/Brat4/wholeGenomeAnalyses/legionella_analysis/legionella subtreeNamesFromTop.txt 5 3,2,1,4,5,6`

The output is shown in Figure 7.

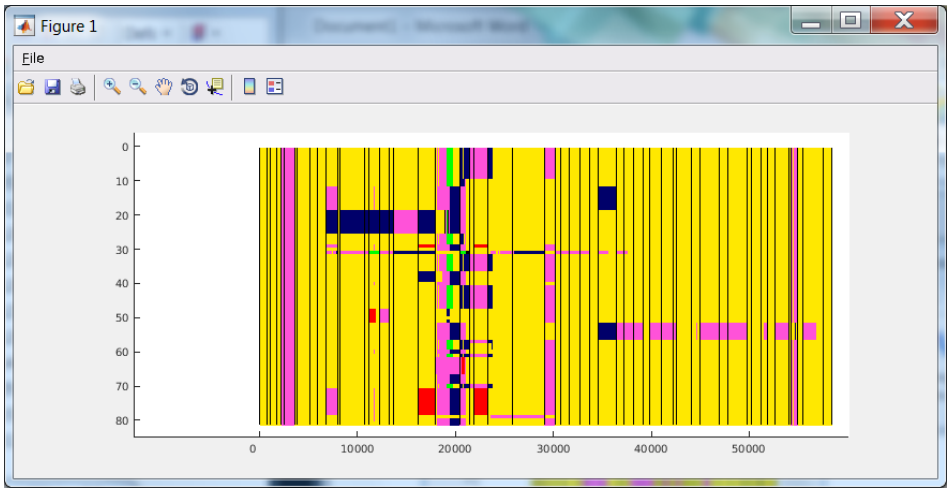


Figure 7

Reconstructing recombinations to a given tree

In David et al. (2017) *PLOS Genetics*, the numbers of recombination events were computed in two ways: first, the recent and ancestral recombinations as obtained from the fastGEAR output, and, second, by reconstructing the changes in the population structure, as estimated by fastGEAR, on a given tree, and reporting the number of times the population structure changed along the branches of the tree. The latter was done to make the results comparable with those from Gubbins (see David et al.), which identifies recombinations that have occurred in branches of a phylogenetic tree. In detail, if, for example, two kinds of sequences are observed in some gene alignment that differ from each other across the whole length of the sequence, then fastGEAR will detect this as two lineages within that alignment, but will not detect recent or ancestral recombinations. If, however, we have an externally estimated tree, we can compare the lineages with the tree, and interpret the same lineage seen in different branches of the tree as having arised through recombination.

USAGE: `./run_startAncestryReconstruction.sh <path to matlab/mcr> <rootToResults> <datasetName> <treeFileName>`

`The treeFileName` is the name of a file that is located in the summary folder, and which contains the tree on which the changes in population structure are to be reconstructed in Newick file format.

EXAMPLE: `./run_startAncestryReconstruction.sh /opt/matlab2016a /m/cs/scratch/mi/pemartti/Brat4/wholeGenomeAnalyses/legionella_analysis/ legionella ST1_subtree.tre`

The output is written in the summary folder in a file named as *legionella_reconstructed_num_rec.txt*.

Collecting recombination statistics

This collects the numbers of recent and ancestral recombinations in different genes

USAGE: `./run_collectRecombinationStatistics.sh <path to matlab/mcr> <rootToResults> <datasetName>`

EXAMPLE: `./run_collectRecombinationStatistics.sh /opt/matlab2016a /m/cs/scratch/mi/pemartti/Brat4/wholeGenomeAnalyses/legionella_analysis/legionella`

The output is written in the summary folder in a file named as *legionella_recSummaries.txt*. The column names in this file are self-descriptive – the last two report the numbers of ancestral and recent recombinations in each gene.

Plotting the proportion of shared ancestry

This plots the proportion of shared ancestry (PSA) matrix for a given set of strains (see Mostowy et al. 2017).

USAGE: `./run_plotPSA.sh <path to matlab/mcr> <rootToResults> <datasetName> <fileWithNamesOrdered>`

EXAMPLE: `./run_plotPSA.sh /opt/matlab2016a /m/cs/scratch/mi/pemartti/Brat4/wholeGenomeAnalyses/legionella_analysis/legionella subtreeNamesFromTop.txt`