BratNextGen-Manual

Detecting recombination events in bacterial genomes from large population samples

Pekka Marttinen

4/18/2011

BratNextGen was developed as a joint project of Pekka Marttinen (1), William P. Hanage (2), Nicholas J. Croucher (3), Thomas R. Connor(3), Simon R. Harris(3), Stephen Bentley(3) and Jukka Corander(4,5). (1) Aalto university, (2) Harvard, (3) Sanger Institute, Cambridge, (4) Åbo Akademi University, (5) University of Helsinki

Table of contents

Introduction

BratNextGen is a software tool designed for the detection of recombination events within large bacterial data sets. BratNextGen can be used for analysing aligned whole genome data sets. The output of BratNextGen consists of detected recombinogenic segments.

A detailed description of the models and algorithms implemented in BratNextGen can be found in:

> **Marttinen P, Hanage WP, Croucher NJ, Connor TR, Harris SR, Bentley S, Corander J. (2011). Detection of recombination events in bacterial genomes from large population samples.** ***Submitted.***

BratNextGen can be downloaded from http://web.abo.fi/fak/mnf/mate/jc/software/bratNextGen.html and used freely for academic purposes. If you use BratNextGen, please cite the paper specified above.

> **BratNextGen comes with no warranties whatsoever. The user alone is responsible for results obtained with BratNextGen.**

We note that BratNextGen has been extensively tested with synthetic data sets and using several real whole genome alignments in order to verify the basic correctness of the algorithm. However, if you observe suspicious behaviour or unexpected crashes, please notify us by sending e-mail to the address given below. For guidance about how to interpret the results, see the discussion and examples in Marttinen et al. (2011).

Questions and feedback can be sent to:
pekka.marttinen@aalto.fi

Installation

BratNextGen is created by Matlab, and compiled with Matlab compiler. Due to the limitations in the properties of the compiler (software must be compiled inside the same operating system in which it is run), the compiled version is currently available only for Windows XP, Windows 7 and Mac OS X. To run the compiled version of BratNextGen, **Matlab Compiler Runtime (MCR) must be installed**. MCR is free of charge and is distributed with BratNextGen.

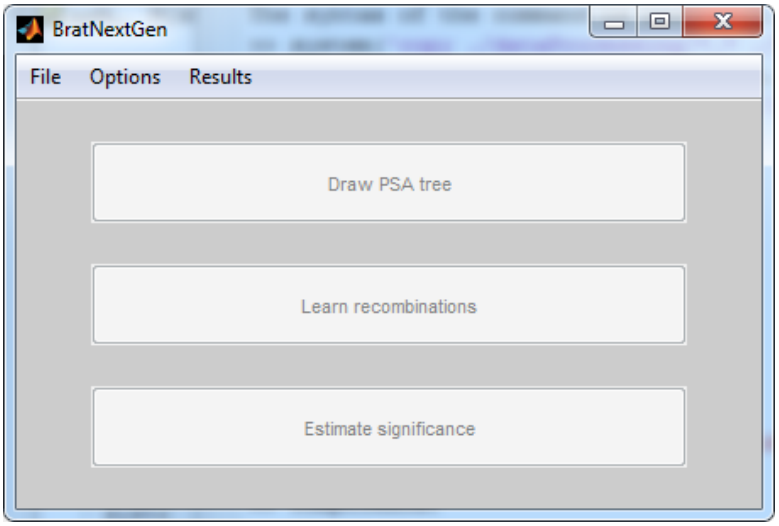
MCR is installed by running MCRInstaller.exe application. The MCR can be installed anywhere in the operating system, EXCEPT under the Matlab path, if Matlab is installed on your computer. After installation, system paths must be updated (on Windows, the paths are set automatically). See readme.txt file for further information.

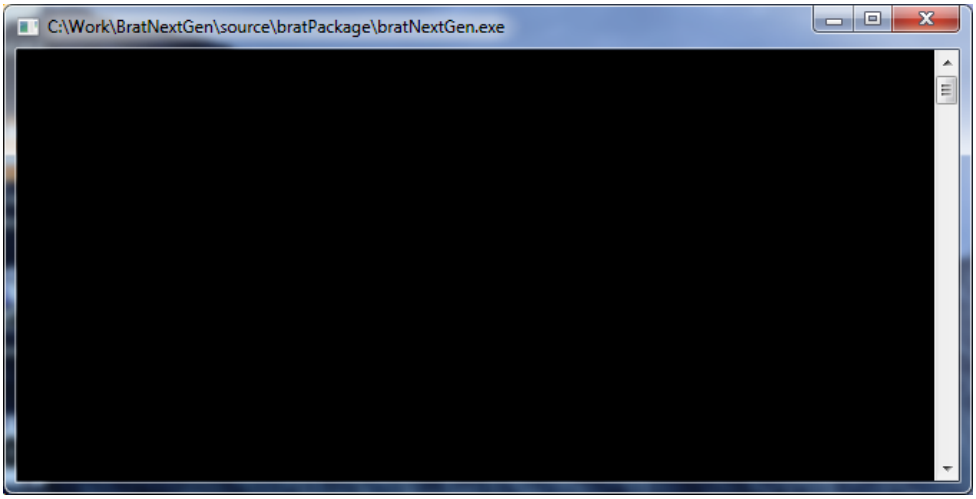
After the installation of the MCR, unzip the BratNextGen package in any folder, and the program is ready for use.

A complete example

Beginning

Start BratNextGen by double clicking bratNextGen.exe. Two windows are opened, one for operating BratNextGen, the other (the DOS window) for displaying information about program status:

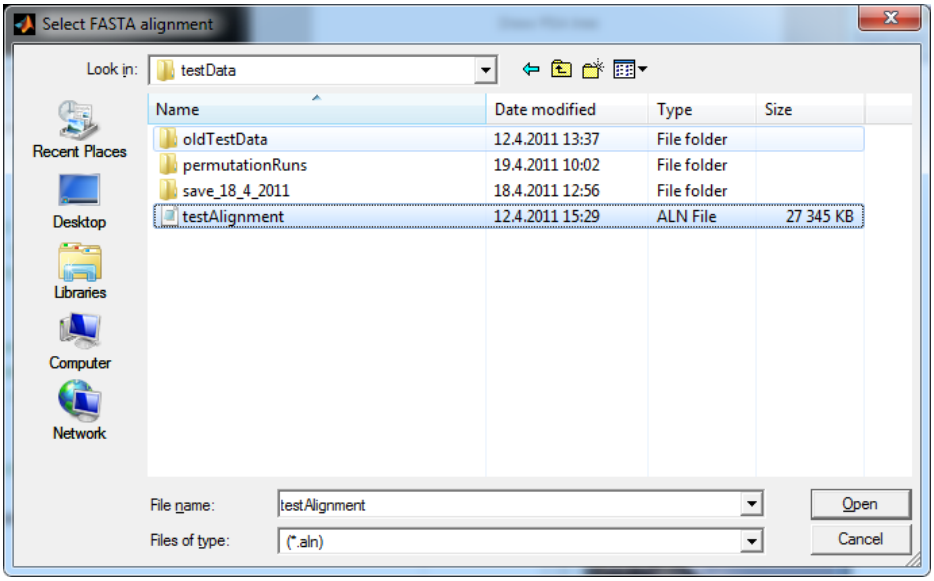




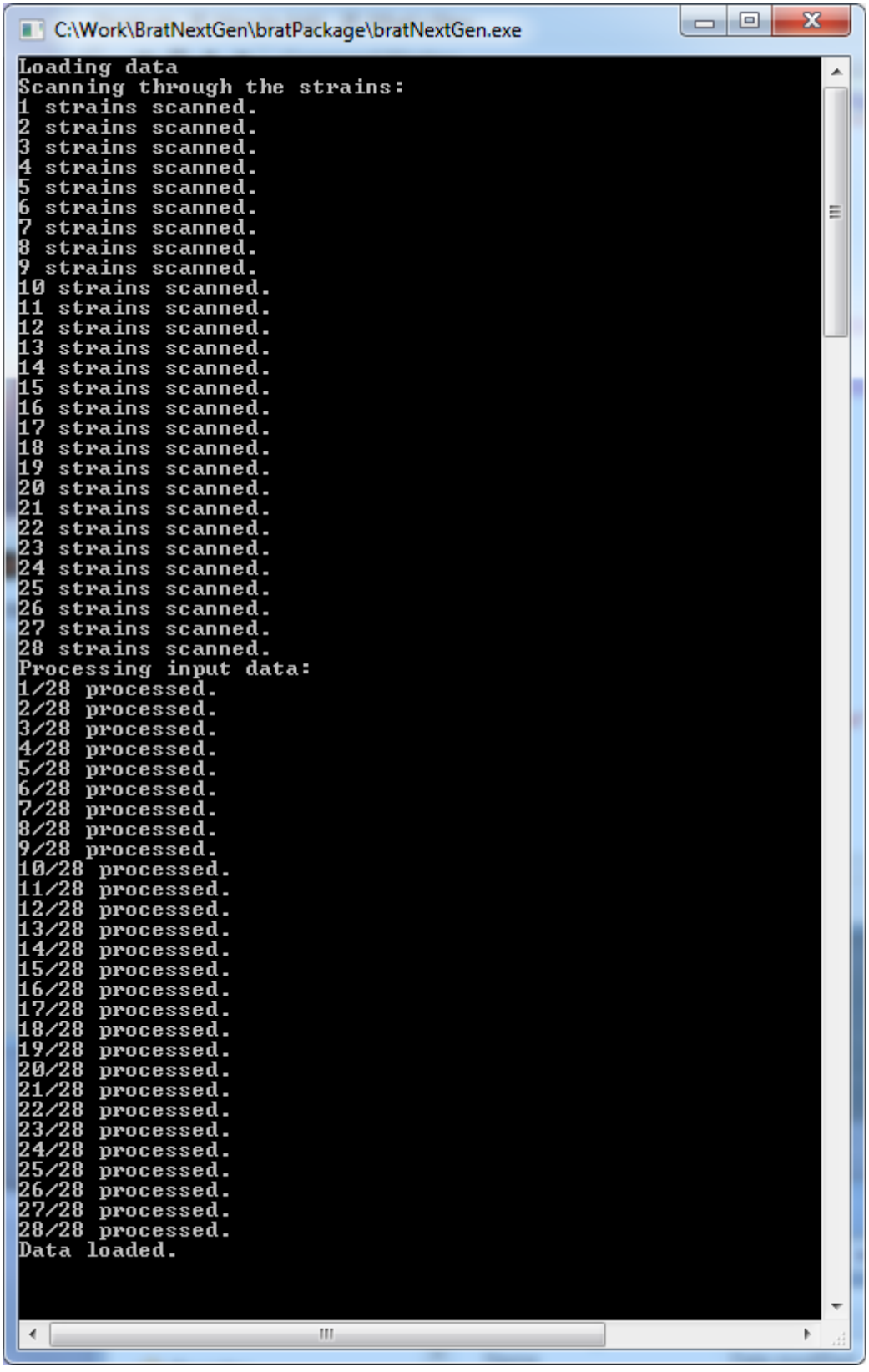
Loading data

The example data set is a FASTA alignment, and it can be found in file .\testData\testAlignment.aln.

1) Select Load data from File menu.
2) Navigate to the folder containing testAlignment.aln, and select the file.
3) Click ‘Open’.

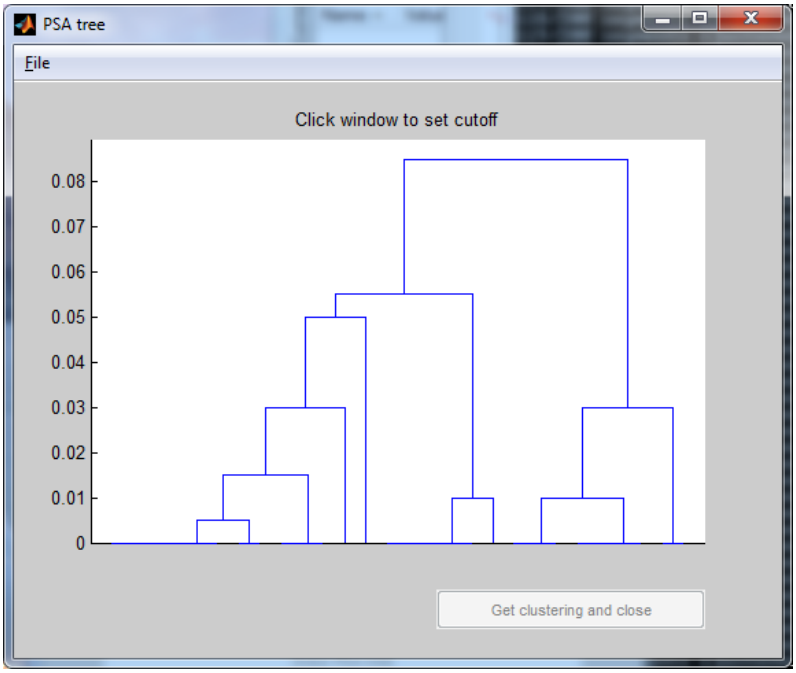


The software scans through the sequences in testAlignment.aln and processes the sequences (this takes about 20 seconds on 3.0 GHz PC). The following output can be seen in the status window:

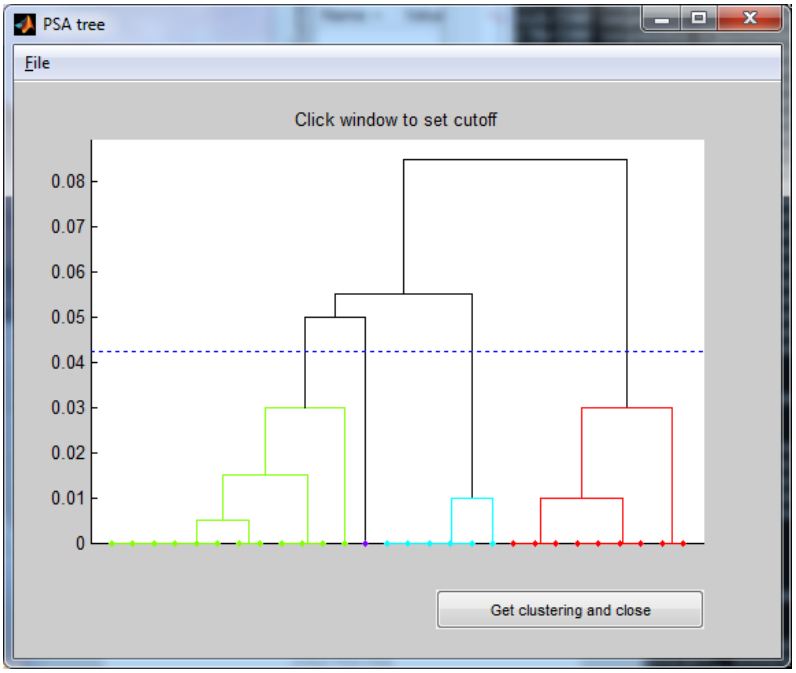


Drawing the proportion of shared ancestry (PSA) tree

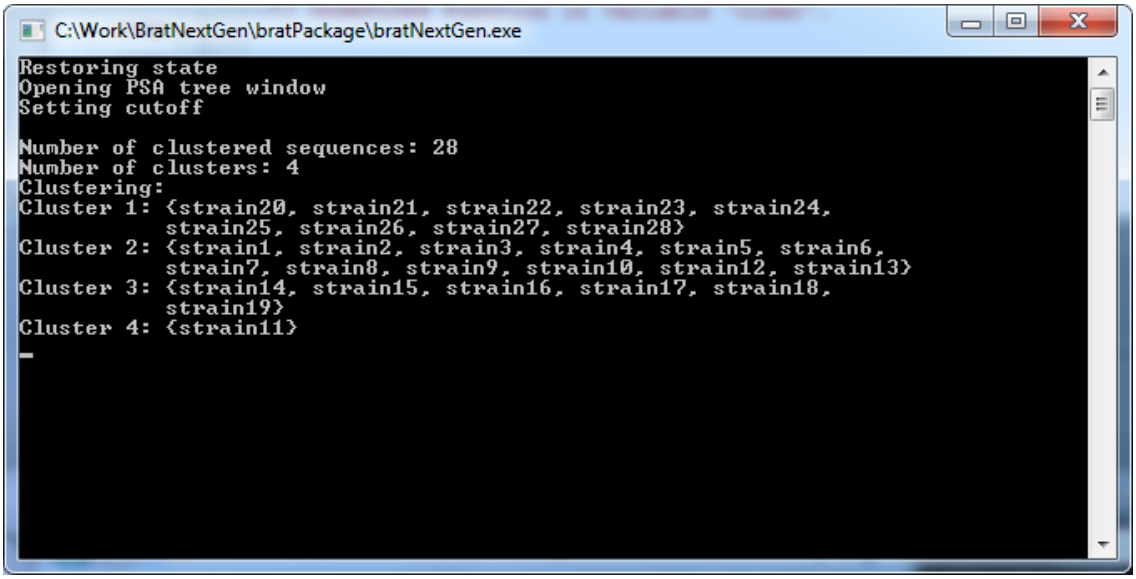
1) Click ‘Draw PSA tree’. After learning the hyperparameter ‘alpha’ and performing clustering of the samples on separate 5 kb intervals, the following window is opened:



2) Click the window to set an appropriate cutoff level.



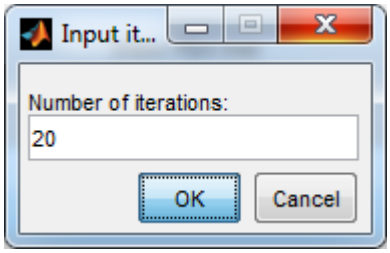
3) Branches below the cutoff level form clusters, which are printed on the status screen:



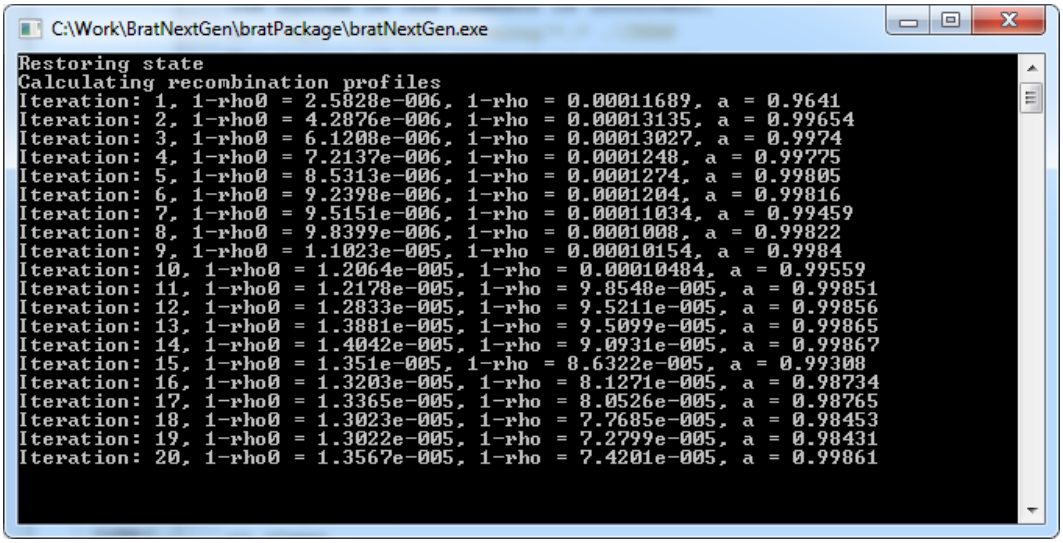
4) Click 'Get clustering and close' button. This saves the clustering created using the current cutoff level. These clusters are used when initializing recombination model learning algorithm.

Learning recombinations

1) Click ‘Learn recombinations’.
2) A dialog box opens. Select the number of iterations and click OK:



3) The following output is printed on the status screen. When the values of the parameters have stabilized, the convergence is approximately reached. If needed, you can run more iterations by clicking ‘Learn recombinations’ again after the algorithm has finished.

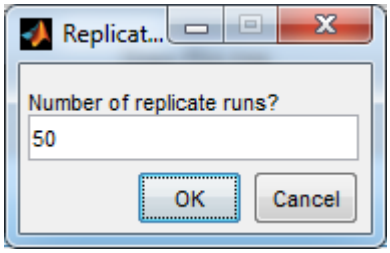


Estimating significance (single processor)

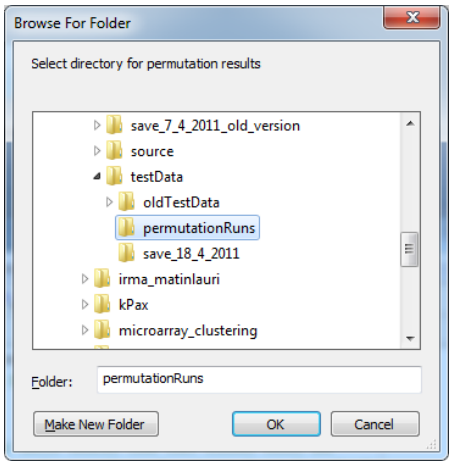
The significance is estimated by running the same analysis a given number of times, such that the columns of the data matrix are permuted. By comparing the results from these permutation resampling runs with the original results, empirical p-values for the detected segments can be inferred. The permutation runs can be done either in parallel if necessary facilities are available, or locally as a batch in a single computer. The recommended number of permutation runs is 100. The minimum number of runs to achieve 0.05 resolution in the empirical p-values is 20.

This section explains how to run the permutations on a single computer.

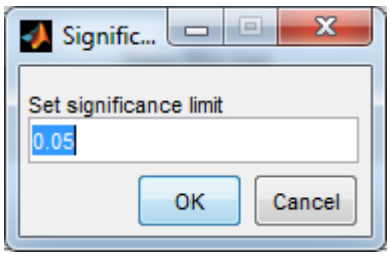
1) Click 'Estimate significance' and select 'On this computer'
2) Specify the number of replicate runs, e.g. 50. The total time is 50 times that of the original run. The analysis of a single replicate of the example data set takes about 2 minutes on a single 3.0 GHz PC such that all 50 replicates take about 2*50 minutes i.e. approximately two hours.



3) Select a directory to which to save results from the permutation runs (e.g. `C:\Work\BratNextGen\testData\permutationRuns`):



4) After the runs have completed, select significance threshold, e.g. 0.05:

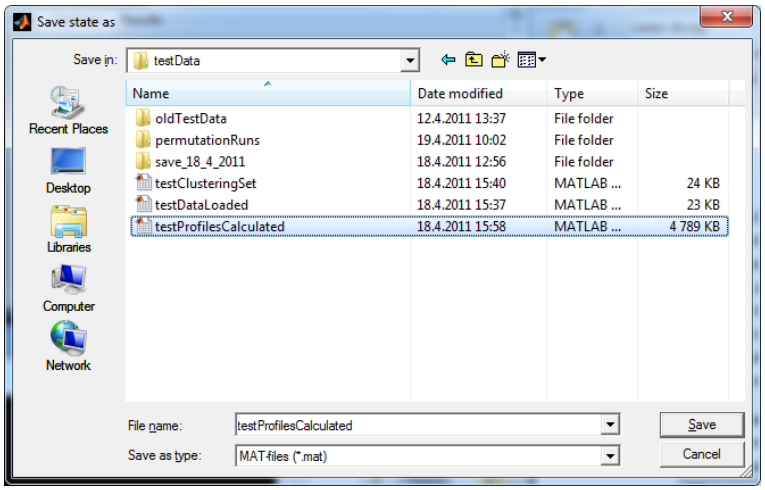


5) The software calculates empirical p-values for all detected segments. Now you can proceed to Result summaries section to view the significantly detected segments.

Estimating significance (parallel computation)

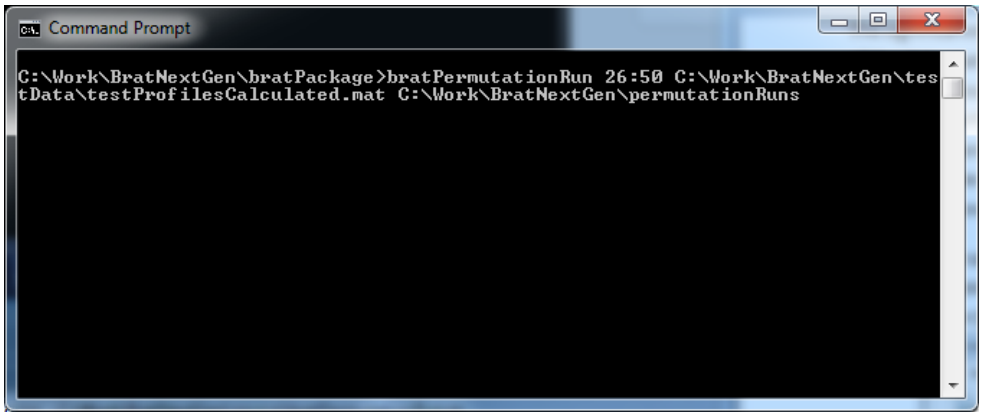
This section describes how to run the permutation runs in parallel in Windows environment.

1) After calculating the recombination profiles, save the state of the program by selecting File->Save analysis. Save the analysis for example to a file named 'testProfilesCalculated.mat' in the 'testData' folder (this is the same folder in which the original alignment resided).

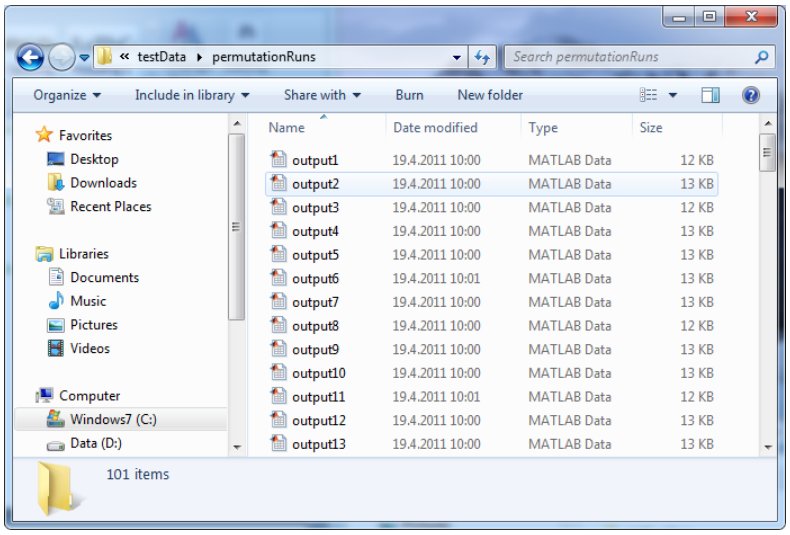


2) Open "Command prompt", and navigate to the Brat folder (see the screen shot below). Run the command: *bratPermutationRun <iterationNumbers> <inputFile> <outputFolder>* , where *iterationNumbers* specifies the iterations to run, *inputFile* is the file saved in step 1) and *outputFolder* is the folder to which to save the results from the permutation re-sampling runs. For example, you can do 100 permutations in four parallel batches by running the following commands on 4 different processors:

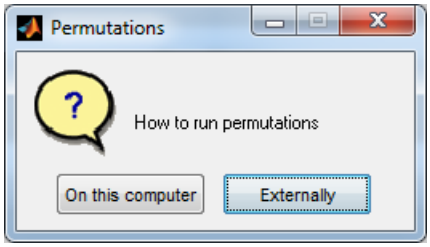
```
bratPermutationRun 1:25
C:\Work\BratNextGen\testData\testProfilesCalculated.mat
C:\Work\BratNextGen\testData\permutationRuns

bratPermutationRun 26:50
C:\Work\BratNextGen\testData\testProfilesCalculated.mat
C:\Work\BratNextGen\testData\permutationRuns

bratPermutationRun 51:75
C:\Work\BratNextGen\testData\testProfilesCalculated.mat
C:\Work\BratNextGen\testData\permutationRuns

bratPermutationRun 76:100
C:\Work\BratNextGen\testData\testProfilesCalculated.mat
C:\Work\BratNextGen\testData\permutationRuns
```



3) After the runs have completed, you should have 100 files in the folder *<outputFolder>* (in the example this folder is `C:\Work\BratNextGen\testData\permutationRuns`)

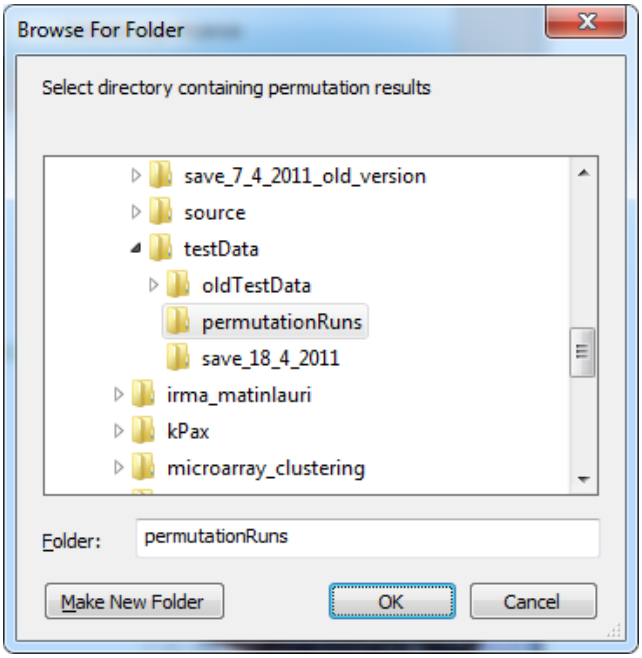


4) Return to the BratNextGen software, and click 'Estimate significance'. A question box appears:



5) Select 'Externally'.

6) Browse for folder containing the results from the permutation runs, and click 'ok'.



7) Select significance threshold, e.g. 0.05.

8) The software calculates empirical p-values for all detected segments. Now you can proceed to Result summaries section to view the detected significant segments.

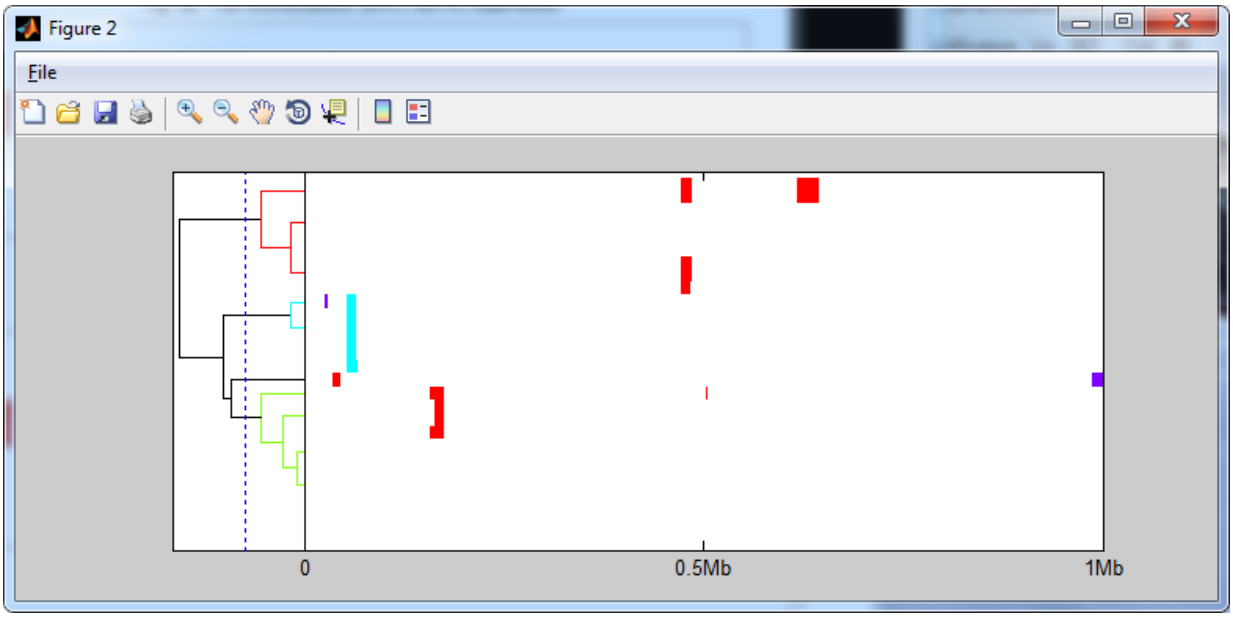
Result summaries

Results->Draw segments

By selecting 'Draw segments' from the results menu, the following figure appears. On the left, the PSA tree is shown. On the right, the segments detected in each sample are shown as colored stripes. The colors should be interpreted as follows:

- The same color at the same column means that the segments in the respective samples are from the same origin.
- A continuous stretch colored with a single color represents a single recombinogenic segment.

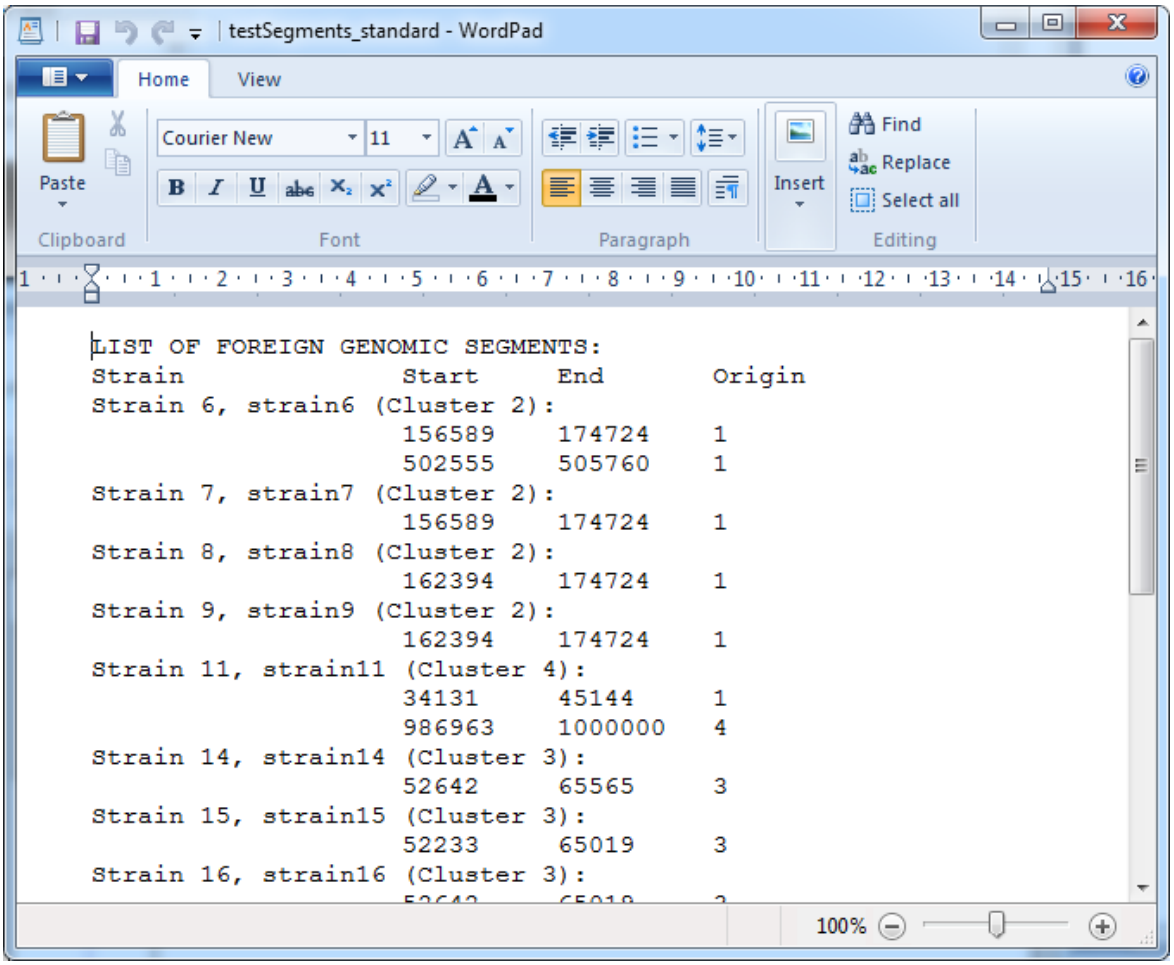
On the other hand, red color at different non-overlapping columns should NOT be interpreted to represent the same ancestral origin for the respective segments. Notice that it depends on the resolution of your monitor whether shortest segments will be visible in practice. However, you can use zoom in to inspect more closely any particular area.



Results->Write segments

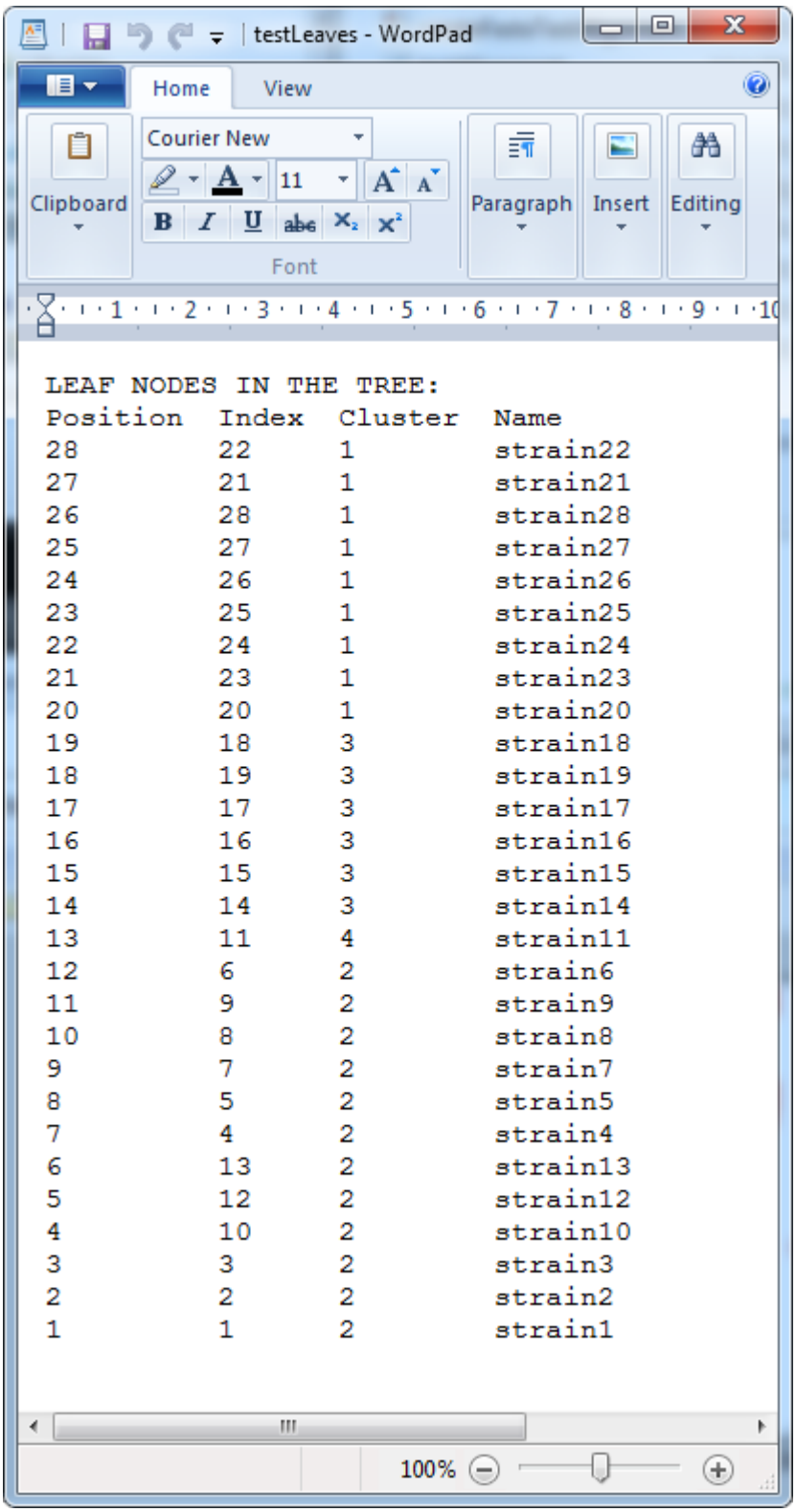
This option can be used to write the detected recombinogenic segments to a file. The segments are written to a file specified by the user. Two files are written in different formats: one in 'standard' format, as shown below, and the other in a 'tabular' format making it easier to read in by other programs.

In this example, a segment 156589-174724 in 'strain6' was detected as recombinogenic, for example, with origin in cluster 1. Again, the cluster labels for the recombinogenic segments do not have any other interpretation except that the same origin at overlapping segments in different samples means that the segments are from the same ancestral origin.



Results->Write tree leaf names

The figure obtained by 'Draw segments' does not contain sample labels. The order of the samples in the tree can be obtained using this option. The output is written to a file specified by the user. The example shows the output for the 'testAlignment.aln' data set. Positions are ordered such that the largest value is the sequence on top of the output figure from 'Draw segments'. 'Index' specifies the index of the sequence in the original alignment.



LEAF NODES IN THE TREE:

Position	Index	Cluster	Name
28	22	1	strain22
27	21	1	strain21
26	28	1	strain28
25	27	1	strain27
24	26	1	strain26
23	25	1	strain25
22	24	1	strain24
21	23	1	strain23
20	20	1	strain20
19	18	3	strain18
18	19	3	strain19
17	17	3	strain17
16	16	3	strain16
15	15	3	strain15
14	14	3	strain14
13	11	4	strain11
12	6	2	strain6
11	9	2	strain9
10	8	2	strain8
9	7	2	strain7
8	5	2	strain5
7	4	2	strain4
6	13	2	strain13
5	12	2	strain12
4	10	2	strain10
3	3	2	strain3
2	2	2	strain2
1	1	2	strain1

Options and other information

Options

If you wish to change some parameters, it is highly recommended that you do that right after loading in data and, after that, keep the selected values. If you change some options after some calculations have already been done, the outcome may be unpredictable.

- **Set hyperparameter**: enables user to specify the parameter 'alpha', see Marttinen et al. (2011). You can either specify a fixed value, or select the option 'Learn alpha', which uses the strategy described in Marttinen et al. (2011) for learning alpha.
- **Select model**:allows user to select among different models (this feature is unavailable for the moment).
- **Probability threshold**: enables user to set a threshold probability value for detecting some segment as recombinogenic. Sequence positions in which the non-recombinogenic cluster gets a probability lower than the given value are detected as putatively recombinogenic when 'Learn recombinations' action has been selected.

Saving and restoring analysis

The analysis can be saved at any stage using **File->Save analysis**, and restored using **File->Restore analysis**. Saving the analysis after each step is recommended such that you do not have to repeat everything from the beginning if something goes wrong. The saving is required if you wish to run the permutation re-sampling runs in parallel outside the program, as described in section *Estimating significance (parallel computation)*.