Detection of recombination events in bacterial genomes from large population samples, supplementary material

Pekka Marttinen, William P. Hanage, Nicholas J. Croucher, Thomas R. Connor, Simon R. Harris, Stephen D. Bentley, Jukka Corander

October 5, 2011

Contents

1 Supplementary materials and methods

1.1 Details of model learning

In Step 2 of the model learning algorithm (see the main text), we use hidden Markov models (HMMs) in order to simulate from the conditional posterior distribution $p(M_{i\cdot}|M_{-i}, \rho_0, \rho, a, Y)$, see e.g. [1] for a detailed discussion of the characteristics of HMMs. The hidden states are in this case the variables M_{ij}, $j = 1, \ldots, L$, where L is the total number of SNPs, and the corresponding observations are Y_{ij}, $j = 1, \ldots, L$. Each hidden state M_{ij} has K possible values corresponding to the K possible clusters $S_j = \{s_{jh}\}_{h=1,\ldots,K}$ that are detectable at site j. The conditional HMM structure then follows, since

$$p(Y_{ij}|M_{i\cdot}, M_{-i}, Y_{-i}, \rho_0, \rho, a) = p(Y_{ij}|M_{ij}, M_{-i}, Y_{-i}),$$

i.e. conditionally on M_{ij}, Y_{ij} is independent on rest of the $M_{i\cdot}$ (given all other conditioning variables). In other words, to specify the probability of observation Y_{ij}, we only need to know the cluster from which the observation was emitted, when the clustering for the remainder of the data is fixed at site j. Formally, these emission probabilities are given by

$$P(Y_{ij} = y|M_{ij} = h, M_{-i}, Y_{-i}) = \frac{\alpha/4 + n_{hjy}}{\sum_{k=1}^{4} \alpha/4 + n_{hjk}}, \tag{1}$$

where n_{hjk} denotes the observed frequency of nucleotide k in cluster s_{jh} at site j ($k = 1, \ldots, 4$, $h = 1, \ldots, K$, $j = 1, \ldots, L$), while excluding the observations of the taxon under investigation. The formula (1) follows by first updating the prior (Eq 2 in the main text) of the base frequencies in cluster s_{jh} to

$$(p_{hj1}, \ldots, p_{hj4}) \sim Dirichlet(\alpha/4 + n_{hj1}, \ldots, \alpha/4 + n_{hj4}), \tag{2}$$

and integrating these frequency parameters out analytically, which is possible because the distribution (2) is conjugate to the multinomial likelihood (Eq 1 in the main text) (for derivation see e.g. [2]). We further have the probabilities

$$p(M_{ij}|M_{i,1:j-1}, M_{-i}, Y_{-i}, \rho_0, \rho, a) = p(M_{ij}|M_{i,j-1}, \rho_0, \rho, a),$$

i.e. the hidden states form a Markov chain. These transition probabilities are determined by the transition matrix (Eq 3 in the main text).

In Step 2, we update parameters ρ_0, ρ, a by maximizing $p(\rho_0, \rho, a|M, Y)$. To do this we notice that given segmentations of the genomes to different origins (determined by M), these parameters are independent of the observations Y.

After some straightforward algebra, we obtain the expression:

$$\begin{aligned}\log p(\rho_0, \rho, a|M) &= const + \log(M|\rho_0, \rho, a) + \log p(\rho_0, \rho, a) \\ &= const + A_1 \log \rho_0 + A_2 \log(1-\rho_0) \\ &+ A_3 \log \rho + A_4 \log(1-\rho) \\ &+ A_5 \log\left[(1-\rho)a\right] + A_6 \log\left[(1-\rho)(1-a)\right] \\ &+ (L^* - 2)\log \rho_0 - 2\log(1-\rho) \\ &+ (\alpha_a - 1)\log(a) + (\beta_a - 1)\log(1-a).\end{aligned} \tag{3}$$

The constants $A_1, \ldots, A_6$ depend on the numbers and lengths of the segments assigned to have their origin either in cluster 1 (the non-recombining case) or in some other cluster. For example,

$$A_1 = \left(\sum_{i=1}^{m} l_i\right) - m,$$

where m is the number of segments among all taxa assigned to cluster 1 and $l_1, \ldots, l_m$ are the lengths of these segments. The last two rows in (3) follow from the prior assumptions for ρ_0, ρ, a (Eqs 4, 5 and 7 in the main text). Notice that each term in the sum (3) depends on one of the parameters ρ_0, ρ, a only. Therefore, each parameter can be straightforwardly maximized separately by finding the point in which the respective partial derivative equals zero.

Unlike in a fully Bayesian analysis, which would use MCMC to explore the posterior distribution over the whole model space (at least in theory), we use the MCMC in Step 2 as a search algorithm for refining our approximation of the posterior mode of M. In practice the iterative operators in Step 2 tend to converge quite rapidly, such that after approximately 5 iterations the model structure M remains almost unchanged. When running *in silico* experiments with our method, we executed the algorithm with 1000 iterations for the *S. pneumoniae* data. When the results were compared with those obtained using only five iterations, the differences were found to be negligible. All results presented in this manuscript are obtained by stopping the iterative algorithm after 10 iterations.

In the Step 3 our algorithm obtains the required probabilities $p(M_{ij} = 1|M_{-i}, \rho_0, \rho, a, Y)$ immediately using the standard forward-backward algorithm. Notice that the probabilities are only reported conditional on the optimal model. The reason for this is that the contents of the clusters may change between iterations, when taxa are moved from one cluster to another. Therefore, averaging the probabilities over the iterations in Step 3 would lack any reasonable interpretation.

In the Step 4, for each segment detected in a particular re-analysis of a permuted data set we calculate the Bayes factor [3] against having the non-recombining state. Then, an empirical p-value for each segment detected in the original data can be obtained as a proportion of permuted data sets in which some segment for the same sequence has a higher value of the Bayes factor than

the original segment. In all analyses we carried out 100 permutation runs and used $p = 0.05$ as the threshold for reporting a finding as statistically significant.

1.2 Blockwise clustering for initialization

To initialize the algorithm presented in the previous section, we use the following clustering approach. Firstly, the total genome is evenly split into contiguous segments of length 5kb and the single-nucleotide polymorphisms (SNPs) within each segment are identified. In cases where a segment contains less SNPs than a given threshold, say <20, the segment is combined with the next neighboring segment in the alignment. This merging procedure is continued until all considered segments contain at least as many SNPs as specified by the threshold. Assume now that the data pre-processing step yields m segments. Then, we process the data from each segment c into a separate matrix B_c where columns map uniquely to the SNPs of the segment and the elements are the bases at the corresponding sites. The m data block matrices $B_1, \ldots, B_m$ are then clustered independently of each other using the basic model (with hyperparameter α set as described in the main text) for clustering of individual samples in BAPS software [4]. Notice that these analyses can be effectively performed in a parallel environment by running m independent copies of the estimation algorithm if needed. Furthermore, as demonstrated earlier in a highly complex bacterial population setting [5], the stochastic optimization algorithm used in BAPS can efficiently infer even several dozens of underlying groups hidden in a data set.

The posterior mode estimates of the m clusterings are then used to create a hierarchical representation of the taxa, defined as the proportion of shared ancestry (PSA) tree. Let $P_1, \ldots, P_m$ denote the estimated optimal clusterings for the segments, such that each P_c is a partition of the N taxa into an arbitrary number of non-overlapping subsets. The information contained in the m partitions is summarized in terms of an ultrametric tree as follows:

1. Calculate distances d^*_{ij} between every pair (i, j) of taxa. The distances are calculated as the fraction of all 5kb segments in which the taxa i and j are in different clusters in the respective clusterings S_c.

2. Calculate a dendrogram, i.e. an indexed binary tree with the N taxa as leaf nodes. The dendrogram is calculated using the standard complete linkage distances, see e.g. [6].

In our approach the PSA tree is used to initialize the search, but it may also be of interest in itself for describing evolutionary relationships among recombinogenic samples. We investigated the behaviour of PSA trees using simulated data. We analyzed two different data sets corresponding to tree heights $d = 0.01$ and $d = 0.03$ (see the main text), each with and without recombinations introduced. The true recombinations in either case were those shown in Fig. 3 of the main text. When no recombinations were present, the estimated PSA trees were completely flat, accurately reflecting the absence of recombinations. Supplementary Fig. S1 shows the estimated PSA trees when recombinations were

added to the data sets. When compared to the PSA tree with clustering results perfectly reflecting the recombinations (the 'true' PSA tree), the tree estimated with $d = 0.03$ is closer to the optimal result. With both $d = 0.01$ and $d = 0.03$ the height of the estimated tree is underestimated compared to the truth, which is caused by the insensitivity of the clustering model to detect all recombinations. However, with both data sets the branches in the estimated PSA trees consist of sequences which have common recombinations, thus providing a suitable basis for initializing our algorithm as described in the following. Notice how sequences close to each other in a PSA tree are closely related also in the original tree, due to the fact that a recombination event always affects a branch of the phylogeny comprising a group of closely related sequences.

The PSA tree is used to create a *summary clustering* of taxa in a data set. The summary clustering can be created in a standard manner by cutting the tree at an appropriate level, the branches below the cutoff level then forming the clusters. This decision is guided by subjective consideration but standard guidelines for analyzing dendrogram appearance can be used. Nevertheless, since the clusters are used only for initializing the search algorithm, different cutoff levels can be used to initialize the algorithm on different runs, if desired. Finally, the initial state is obtained by concatenating the clusterings $P_1, \ldots, P_m$ on separate 5 kb intervals, and selecting the labels for the clusters on each interval using the following procedure:

1. For all intervals $c = 1, \ldots, m$, label all clusters in P_c starting from the largest cluster. When labeling a cluster, select a label unused in that interval such that the *cluster in the summary clustering* with that label is most prevalent in the cluster.

2. Finally, when all clusters for all intervals are assigned labels, we iterate over the intervals to further refine the initial state. This is achieved by finding a permutation of the labels of clusters in P_c that minimizes the average entropy of label distributions over the sequences. These iterations are continued until no changes occur (typically this would happen after a minor number of iterations). The outcome of this procedure is that the labelings of the sequences change as few times as possible over the genome.

The process of creating the initial state is illustrated by showing the intermediate states for *S. pneumoniae* data: i) clusterings shown side by side without ordering the cluster labels (colors), ii) clusterings shown side by side and labels ordered according to step 1, and iii) clusterings shown side by side and labels ordered using both steps 1 and 2. These intermediate results are shown in supplementary Figs. S2-S4, respectively.

1.3 Simulated test data sets

The simulated data sets were created as described below.

- We created a random tree (or genealogy) using *Recodon* [7], which is a software for generating genealogies and sequences using a coalescent.

The tree was created for 90 sequences which had been sampled from 3 populations each of size 30. We used the following command to create the tree:

- *recodon1.6.0_win -n1 -s90 -l1 -e1000 -q3 30 30 30 0.0001 -r0.00 -jtrees*

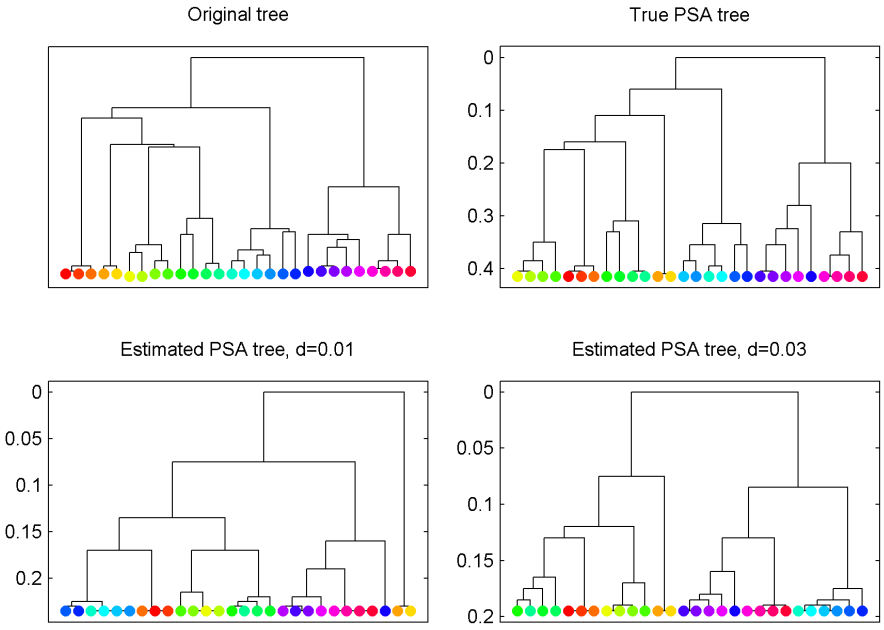
The tree is shown in Fig. 2 in the main text.

- We used the tree created in the previous step as an input to *Seq-gen* software [8] to create the actual sequences of length 1 Mb. We used the HKY mutation model [9] with transition-transversion ratio set to 2.0 and the stationary distribution equal to (0.1, 0.4, 0.4, 0.1). The tree height in units of substitutions per site was either 0.01 or 0.03. The cyan-colored taxa in Fig. 2 in the main text were considered as the samples. We simulated a given number of recombination events into the branches of the subtree containing the cyan-colored taxa. The origin of each recombination event was randomly selected to be one of the colored clusters shown in Fig. 2. The sequences affected by recombination were assigned a segment from randomly selected taxa from the donor cluster. The resulting numbers of polymorphic sites in the alignments before and after the recombinations were introduced are shown in Table 2 in the main text. If a recombination affected a branch which contains the majority of all taxa, the complement group of these taxa was considered as recombinant when evaluating the results. The reason for this is the nonidentifiability issue which is discussed in more detail in the *Discussion* in the main text.

2 Supplementary figures

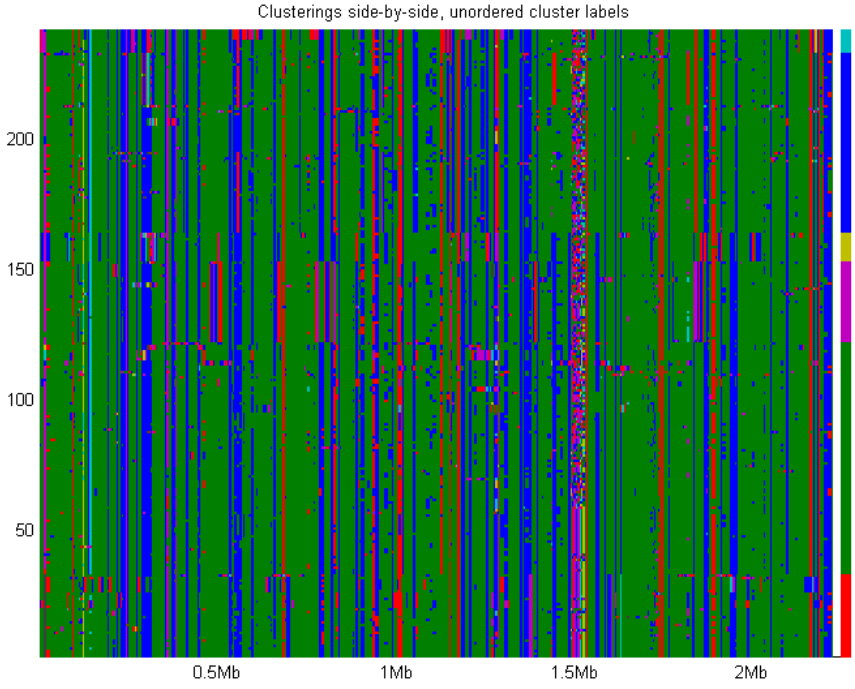
2.1 Supplementary figure S1: Estimated PSA trees

The figure shows estimated PSA trees for simulated data. The top left panel shows the underlying phylogenetic tree, according to which data were generated. The top right panel shows the PSA tree corresponding to the situation that all recombinations were detected by the initial clustering analysis. The panels at the bottom show estimated PSA trees when the generating tree height was 0.01 and 0.03, respectively, and recombinations were introduced to the data sets.



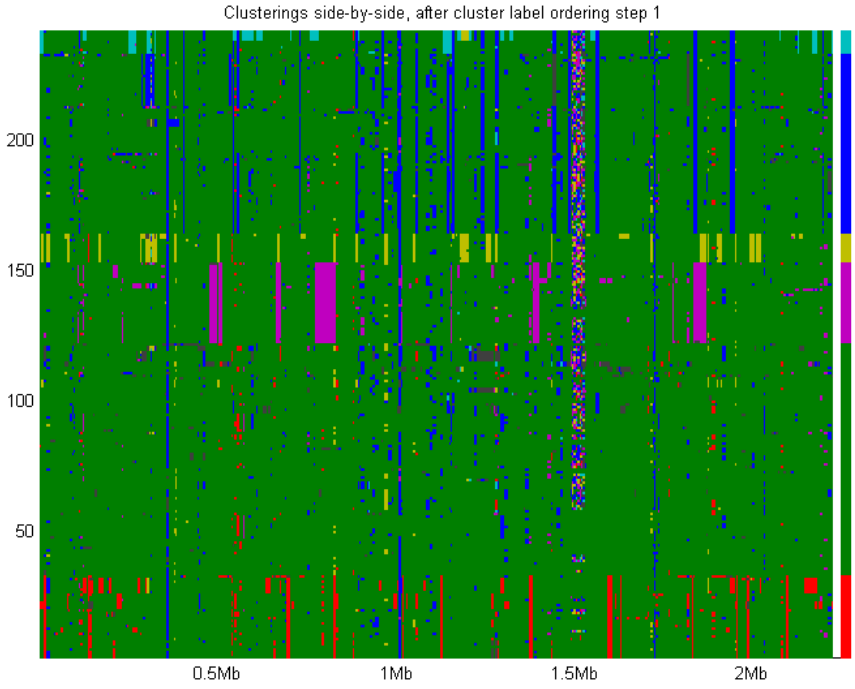
2.2 Supplementary figure S2: Initialization, unordered cluster labels

The figure shows the first intermediate state of the initialization of the search algorithm which consists of concatenated clusterings at 5 kb intervals before ordering of the cluster labels. The vertical axis comprises the 241 *S.pneumonia* samples, and the horizontal axis shows the position along the genome. The 5 kb columns are colored with different colors corresponding to different cluster labels. The color bar on the right shows the same *global* clusters as in Fig. 4 of the main text.



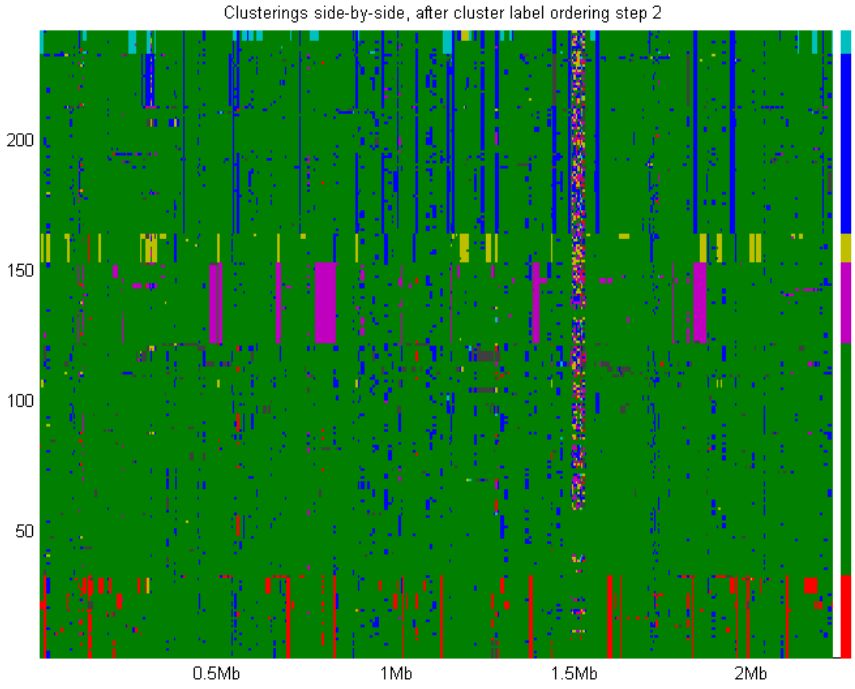
2.3 Supplementary figure S3: Initialization, after the first cluster label ordering step

The figure shows the second intermediate state of the initialization of the search algorithm after the cluster label ordering step 1.



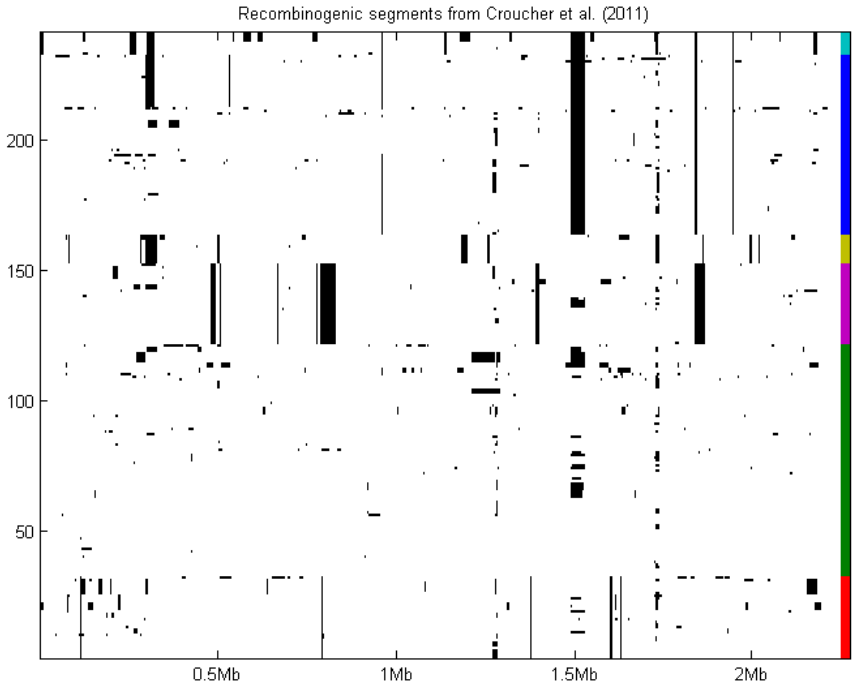
2.4 Supplementary figure S4: Initialization, after the second cluster label ordering step

The figure shows the third intermediate state of the initialization of the search algorithm after the cluster label ordering step 2.



2.5 Supplementary figure S5: Recombinogenic segments for *S. pneumoniae* data from Croucher et al. (2011)

The figure shows the recombination events for the *S.pneumoniae* data, detected in [10]. The samples are ordered in the same order as in Fig. 4 of the main text, and the colored bars on the right are used to label the same clusters as in Fig. 4 of the main text.



References

[1] C.M. Bishop. *Pattern recognition and machine learning*, volume 4. Springer New York, 2006.

[2] R.E. Neapolitan. *Learning bayesian networks*. Pearson Prentice Hall Upper Saddle River, NJ, 2004.

[3] R. Kass and A. E. Raftery. Bayes factors. *Journal of American Statistical Association*, 90:773–795, 1995.

[4] J. Corander, P. Marttinen, J. Sirén, and J. Tang. Enhanced Bayesian modelling in BAPS software for learning genetic structures of populations. *BMC bioinformatics*, 9(1):539, 2008.

[5] J. Tang, W.P. Hanage, C. Fraser, and J. Corander. Identifying currents in the gene pool for bacterial populations using an integrative approach. *PLoS Comput Biol*, 5(8):e1000455, 2009.

[6] K.V. Mardia, J.T. Kent, and J.M. Bibby. *Multivariate analysis*. Academic press, 1980.

[7] M. Arenas and D. Posada. Recodon: coalescent simulation of coding DNA sequences with recombination, migration and demography. *BMC bioinformatics*, 8(1):458, 2007.

[8] A. Rambaut and N.C. Grass. Seq-Gen: an application for the Monte Carlo simulation of DNA sequence evolution along phylogenetic trees. *Computer applications in the biosciences: CABIOS*, 13(3):235–238, 1997.

[9] M. Hasegawa, H. Kishino, and T. Yano. Dating of the human-ape splitting by a molecular clock of mitochondrial DNA. *Journal of molecular evolution*, 22(2):160–174, 1985.

[10] N.J. Croucher, S.R. Harris, C. Fraser, M.A. Quail, J. Burton, M. van der Linden, L. McGee, A. von Gottberg, J.H. Song, K.S. Ko, et al. Rapid pneumococcal evolution in response to clinical interventions. *Science*, 331(6016):430–434, 2011.