This article is intended for submission as an Article in the Tree of Life issue (CFP-Tree) of *MBE*.

A coarse-graining, ultrametric approach to resolve the phylogeny of prokaryotic strains with frequent recombination

Tin Yau Pang

Institute for Computer Science, Heinrich Heine University, Düsseldorf, 40225, Germany

To whom correspondence should be addressed. Tel: +49-211-81-11651; Fax: +49-211-81-15767; Email: pang@hhu.de

keywords: phylogenetic algorithm, homologous recombination, ultrametric tree

Abstract

Introduction

Homologous recombination happens when a foreign DNA segment replaces a similar segment on the genome of a prokaryotic cell. For a genome pair, recombination affects their phylogenetic reconstruction in multiple ways: (i) a genome can recombine with a DNA segment that is similar to the other genome of the pair, thereby reducing their pairwise sequence divergence; (ii) a genome can also recombine with a segment from an outgroup-genome and increase the pairwise divergence. Most phylogenetic algorithms cannot account for recombination; while some do, they cannot account for all effects of recombination.

Results

We develop a fast algorithm that takes recombination into account and reconstructs ultrametric-trees. Instead of considering individual positions of genome sequences, we use a coarse-graining approach, which divides a genome sequence into short segments. For each genome pair considered, our coarse-graining phylogenetic (CGP) algorithm enumerates the pairwise single-site-polymorphisms (SSPs) on each segment to obtain the pairwise SSP-distribution; we then fit each empirical SSP-distribution to a theoretical SSP-distribution. We test the performance of our algorithm against other state-of-the-art algorithms on simulated and real genomes. For genomes with a substantial level of recombination, such as *E. coli*, we show that the age of internal nodes calculated by CGP is more accurate than those predicted by other algorithms, while the reconstructed tree topology is at least as accurate.

Conclusion

We develop a phylogenetic algorithm that accounts for recombination. It predicts ultrametric-trees more accurately than alternative algorithms, and is also substantially faster than the current state-of-the-art algorithms in recombination-aware phylogenetic reconstruction.

Introduction

Horizontal transfer of DNA segments between prokaryotes—termed horizontal gene transfer (HGT) or lateral gene transfer (LGT)—is a major driver of prokaryotic evolution (Pál et al. 2005). It is caused by a variety of different mechanisms, including transformation, transduction, conjugation, and gene transfer agents (Ochman et al. 2000; Lang et al. 2012). Many prokaryotic genomes encode defense systems against foreign DNA, such as the restriction modification system (Wilson and Murray 1991). A foreign DNA segment that enters the prokaryotic cell and survives these host defenses may be incorporated into the host genome. If the incoming DNA segment is highly similar to a segment on the host genome, then homologous recombination may occur, where the incoming segment homologously recombines with the host segment and overwrites it (Dixit et al. 2015). Apart from recombination, the incoming segment may also be inserted directly into the host genome through non-homologous recombination.

Horizontal gene transfer allows the fast spread of beneficial genes, allowing prokaryotes to adapt to changes in the environment; for example, HGT is responsible to the spread of antibiotic resistance genes in the pathogenic bacteria (Huddleston 2014). Moreover, recombination is crucial for the long-term maintenance of prokaryotic populations, as it can help to repair DNA damaged by deleterious mutations to avoid the mutational meltdown of Muller's ratchet (Takeuchi et al. 2014); computational modelling also suggests that recombination may help prokaryotes to purge selfish mobile genetic elements (Croucher et al. 2016).

Recombination and HGT can severely disturb phylogeny reconstructions. If we apply a phylogenetic algorithm that does not account for recombination to genomes that recombine frequently, branch lengths will deviate systematically from the true branch lengths. For example, (i) when a segment of genome X recombines with a DNA segment from genome Y, it will erase some of the single site polymorphisms (SSNs) that previously differentiated X and Y, shortening the apparent distance between the genomes; here, an SSN refers either to single nucleotide polymorphism (SNP) or to single amino acid polymorphism (SAP). Conversely, (ii) when X recombines with a DNA segment of an outgroup genome (a genome that diverged before the split of the X and Y lineages), then it introduces SSPs into X, increasing the apparent X-Y distance.

Multilocus sequence typing (MLST) can extract sequences of housekeeping genes from prokaryotic genomes, which can then be applied for phylogenetic reconstruction to resolve evolutionary relationships (Spratt 1999). However, MLST genes may also experience frequent recombination, and phylogenetic reconstruction without accounting for recombination can compromise the resulting trees (Vos and Didelot 2009). In fact, the frequency for recombination to cover a gene can be of the same order of magnitude as the mutation rate of a gene (Dixit et al. 2015). For this reason, application of conventional phylogenetic algorithms without accounting for recombination can lead to a severe underestimation of the age of the common ancestors (Schierup and Hein 2000). When there are more than two strains, recombination can disturb not only the relative divergence times between strains, but may also affect the reliability of the tree topology. Currently, only one published phylogenetic algorithm explicitly accounts for recombination: ClonalFrame (Didelot and Falush 2007), which comes with its sister algorithm ClonalOrigin (Didelot et al. 2010); however, ClonalFrame only takes type (ii) but not type (i) recombination into account.

To correctly and efficiently account for past homologous recombination events in tree topology and in particular in divergence time inferences, we developed a coarse-graining phylogenetic (CGP) algorithm to reconstruct ultrametric phylogenetic trees. While most conventional phylogenetic algorithms consider every variable positions of the core genome, the CGP algorithm divides the core genome into equally sized segments. For an aligned pair of genomes, CGP enumerates the mutual SSPs on every segment of the pair, and thus obtains the pairwise SSP distribution. CGP fits the empirical SSP distributions of all genome pairs to theoretical distributions generated by ultrametric trees to estimate the true phyology. We tested the accuracy of CGP tree topology and branch length predictions with those of other state-of-the-art algorithms, including RAxML (Stamatakis 2014), BEAST (Drummond et al. 2012)**,** and ClonalFrame (Didelot and Falush 2007), using both simulated genomes and real *E. coli* genomes.

Results

A coarse-graining approach to phylogenetic reconstruction

We developed a coarse-graining phylogenetic (CGP) algorithm, which explicitly accounts for recombination while reconstructing ultrametric phylogenetic trees. CGP is based on a mathematical model (Dixit et al. 2015; Pang and Lercher 2016) that quantitatively describes the evolution of genomic sequence divergence in a neutral coalescent framework (Materials and Methods). Instead of considering individual positions of nucleotide or amino acid sequence alignments, CGP considers genomic segments—a fixed number of consecutive nucleotides or amino acids positions—where an alignment is represented by a chain of non-overlapping segments. Given the alignment of a pair of genome sequences (or a set of orthologous genes), the CGP algorithm divides it into L_{seg} segments, where each segment has l_s nucleotide or amino acid sites; CGP then enumerates positions with single site polymorphisms (SSPs) within each segment to obtain the SSP distribution of the pair of genomes. The pairwise SSP distributions of all considered genomes are then used as input for the phylogenetic reconstruction. A segment with l_s sites can have either 0, 1, …, l_s SSPs; thus a segment has l_s+1 states, and a SSP distribution can be represented by a vector of l_s+1 elements. To save computational resources, our algorithm uses a vector of l_s^{cutoff}+1 elements to represent a SSP distribution, assigning segments with l_s^{cutoff} or more SSPs to be in the same state.

The CGP algorithm infers the coalescent time of two genomes by comparing their empirical SSP distribution with theoretical distributions. The details of this algorithm are given in Materials and Methods. The remainder of this subsection, which summarizes the idea of the CGP algorithm; readers who are more interested in the application than in the technical details of CGP can skip this part. For readers who want to try using the algorithm, a link to the source code is provided in Supplementary File S3.

When a prokaryotic lineage splits into two new lineages X and Y, the initial SSP distribution consists of a single peak at zero SSP. As time proceeds, mutations and homologous recombination bring in new SSPs, reshaping the SSP distribution (see Materials and Methods for the detailed model). There are five parameters in the model of CGP that determine the theoretical SSP distribution of a genome pair: (i) mutation rate μ per segment, (ii) homologous recombination rate ρ per segment (*i.e.*, the probability that a recombination event somewhere on the chromosome covers a given segment), (iii) average sequence divergence θ per segment between a random genome pair in the population, (iv) transfer efficiency δ_{TE} , which relates the success rate of recombination with the sequence divergence between the incoming and the host segment, and (v) coalescent time t_{XY} between the segment pair. The unit of divergence θ and efficiency δ_{TE} can be (a) the number of SSPs per segment or (b) the corresponding percentage.

To reconstruct the phylogenetic inheritance of *n* genomes, the CGP algorithm starts from the $n(n$-1)/2 empirical SSP distributions. The vertical phylogenetic inheritance of these *n* genomes is represented by an ultrametric phylogenetic tree. An ultrametric tree *T* with *n* leaves can have no more than *n*-1 internal nodes, and thus the CGP algorithm infers *n*-1 coalescent times from the $n(n$-1)/2 SSP distributions. The solution space of the CGP algorithm includes the model parameters μ, ρ, θ, δ_{TE}—assumed to be constant across segments and across lineages—as well as the ultrametric tree *T*, whose branch lengths directly correspond to time. Each of the $n(n$-1)/2 empirical SSP distribution $g(x)$, where *x* is the number of SSPs on a segment, has a corresponding theoretical distribution $f(x)$; these $n(n$-1)/2 theoretical distributions depend on μ, ρ, θ, δ_{TE}, and *T*. We used the cross entropy (Boer et al. 2005) to measure the similarity between an empirical distribution $g(x)$ and its corresponding theoretical distribution $f(x)$, and the logarithm of the posterior probability of a point in the solution space—described by μ, ρ, θ, δ_{TE} and *T*—is the summation of the $n(n$-1)/2 cross entropies (Materials and Methods).

The CGP algorithm starts with an ultrametric tree constructed from single linkage clustering based on the SSP matrix of the genome pairs, and then performs Markov chain Monte Carlo (MCMC). In each step, it mutates the model parameters μ, ρ, θ, δ_{TE} or the tree *T*, and accepts the move according to its posterior probability. The algorithm proceeds and records the posterior parameters and the posterior tree every 1000 steps; it terminates when the maximum posterior score has not increased by more than 1 for 200,000 steps (Materials and Methods).

CGP accurately predicts coalescent times of simulated genomes

To compare the performance of CGP with that of alternative algorithms, we simulated populations of haploid genomes following the neutral coalescent model with recombination (Fraser et al. 2007), using three different parameter sets ((μ, ρ, θ, δ_{TE})=(0.05, 0.01, 10%, 0.8%), (0.05, 0.25, 10%, 0.8%), and (0.025, 0.25, 10%, 0.8%)). These parameter sets correspond to species with low, intermediate, and high levels of recombination. We evolved genomes with L_{seg}=100 segments, where each segment has l_s=1000 binary sites. Population size is maintained constant throughout the simulation (N_e=1000), and the phylogenetic history of the entire population is recorded (Materials and Methods). We set the transfer efficiency δ_{TE}=0.8% in all simulations, which is consistent with previous reports of 0.8% (Dixit et al. 2015) and 2.2% (Fraser et al. 2007) for *E. coli*. The ratio of the contributions to divergence by recombination and by mutation events (r/m) was found to be around 1.6, 40, and 80, respectively, for the three sets of model parameters (Materials and Methods). The r/m values of the simulated population largely overlap with the r/m values found in nature, which range from 0.02 to 63.6 (Vos and Didelot 2009), and thus are appropriate to represent the broad range of recombination levels observed in real genomes.

For each of the three parameter sets (representing low, intermediate, and high recombination levels), we collected around one hundred groups of closely related genomes from the simulated populations; we then reconstructed the phylogeny of each of these groups; we set the maximal recorded segment difference l_s^{cutoff}=100 in CGP to save computational resources. We compared the estimates of CGP with RAxML (Stamatakis 2014), BEAST (Drummond et al. 2012)**,** and ClonalFrame (CF) (Didelot and Falush 2007). All tested algorithms involve MCMC and generate a series of trees during the process of phylogenetic reconstruction; we compared the series of posterior trees of each algorithm with the true phylogenetic tree of the simulated genome group. To evaluate the accuracy of an algorithm with respect to tree topology, we measured the similarity between the posterior trees and the true tree using the topology similarity score s_{topo}, which is the ratio of correctly inferred clusters over all clusters present, and is also denoted as ‘efficiency’ (Didelot and Falush 2007); to evaluate the accuracy of branch length estimates, we measured the average deviation between the age of internal nodes in the real tree and the age of the corresponding nodes in the posterior trees (the node age deviation d_{age}) (Materials and Methods).

Table 1 summarizes the average topology similarity score and age deviation for different algorithms; Supplementary Tables S2 and S3 show pairwise comparisons among algorithms for topology prediction accuracy and node age accuracy, respectively (Supplementary File S1 shows a table of the raw outputs). We first examined the topologies of the reconstructed trees. CGP was designed to deal with problems in phylogenetic reconstruction that arise through recombination; accordingly, it performed worst among the four tested algorithms for genomes with very low recombination levels. However, CGP performed as well as BEAST and significantly better than both RAxML and ClonalFrame at intermediate and high levels of recombination (Table 1; see Supplementary Table S2 for the statistical significance of pairwise method comparisons). For node age prediction, CGP is significantly better than BEAST and ClonalFrame (Table 1; see Supplementary Table S3 for the statistical significance of pairwise method comparisons); note that RaxML does not provide node age estimates.

Along with the series of posterior trees *T*, CGP also records the posterior parameters (μ, ρ, θ, δ_{TE}); these parameters can provide a general overview of the evolutionary dynamics of the population, in particular the level of recombination and the sequence divergence. μ is fixed throughout the MCMC simulation of CGP, and thus the meaningful parameters are ρ/μ, θ, and δ_{TE}. We calculated the mean of ρ/μ, θ, and δ_{TE} of each genome group by taking the average of the posterior parameters of the last 200,000 MCMC steps. Supplementary Figure S1 shows the histograms of ρ/μ, θ, and δ_{TE} of different genome groups at different levels of recombination. Table 2 compares the average value and standard deviation of these distributions of posterior parameters with the true values of the simulated populations, which shows that the predictions of the posterior parameters are correct within an order of magnitude.

CGP predicts the ultrametric phylogenetic tree of real genomes better than alternative algorithms

To test the accuracy of different algorithms on real data, we collected the nucleotide and amino acid sequences of the core genome of 55 *E. coli* and *Shigella* strains, aligned the alleles of each orthologous gene family, and also prepared the pairwise SSP distributions of the genome pairs (Materials and Methods; see Supplementary Table S1 for the strains included). We will use the umbrella term *E. coli* to refer to all 55 strains, as *Shigella* is sometimes considered as belonging to the *E. coli* species. *Dixit et. el.* pointed out that when the nucleotide sequence divergence between a pair of genomes reaches a boundary of 1.3%, there is virtually no segment of the pair left untouched by recombination (Dixit et al. 2015). As recombination erases phylogenetic signals of a genome, the CGP algorithm might perform differently below and above this cut-off. Hence, we performed two different tests, assigning different constraints to the nucleotide sequence divergence of the strains. In test 1, we imposed the constraint that no strain pair within a test group can exceed an average nucleotide sequence divergence of 1.3%; in test 2, we did not impose any constraint on sequence divergence. Each of test 1 and test 2 contains 100 groups; each group has 10 strains, randomly picked from the 55 strains, following the criteria imposed on the tests. To save computational resources, we did not concatenate all universal genes into a 'super-gene' for each strain, but instead randomly selected 100 genes, concatenating their nucleotide sequences and separately their amino acid sequences. Thus, each of the 10 strains in a group is represented by a nucleotide sequence and an amino acid sequence, with the sequences of all 10 strains forming an alignment. We used a segment size l_s=30 for nucleotide sequences and l_s=10 for amino acid sequences, and set l_s^{cutoff} to its maximum possible value, i.e., l_s^{cutoff}=l_s, to simplify the numerical calculations (see Supplementary File S2 for table of the strains and genes used in different test groups). We performed five different phylogenetic reconstructions for each sequence set, using (i) CGP, (ii) BEAST, and (iii) ClonalFrame on nucleotide sequences, as well as (iv) CGP and (v) BEAST on amino acid sequences (Materials and Methods).

As we do not know the true vertical phylogeny of the *E. coli* genomes, we evaluated the accuracy of each reconstructed phylogeny by comparing its posterior trees with the phylogenetic signals inferred from absence and presence of genes across different genomes (Cohen et al. 2010). We summarized a series of posterior trees (of CGP or BEAST) using the "treeannotator" program, which is part of the BEAST package and calculates the maximum clade credibility tree from a posterior tree series, to create a representative tree for a phylogenetic reconstruction; for ClonalFrame, its output file already contains a consensus tree, and we used it as its representative tree. Treating each internal node of the representative tree as an ancestral strain, we applied GLOOME (Cohen et al. 2010), a maximum likelihood algorithm, to reconstruct the presence and absence of genes in the ancestral strains, and also the genes transferred horizontally into the ancestral genomes, based on the representative tree and the presence and absence of genes across different extant strains (Materials and Methods). We used the GLOOME posterior likelihood (GPL) of the ancestral genome reconstruction to serves as an indicator to quantify the accuracy of each representative tree: the more accurate the representative tree is, the better it should match the phylogenetic signal inferred from absence and presence of genes in different genomes, and hence the higher its GPL. Further, we reconstructed the HGT events and the genes transferred in each event; we used the number of reconstructed HGT events (N_{HGT}) as another indicator to evaluate the accuracy of a representative tree (Materials and Methods)—we expect that the more the representative tree deviates from the true phylogeny, the more erroneous HGT events are inferred; thus, lower values of N_{HGT} indicate more reliable representative trees.

Supplementary File S2 lists the GPL and N_{HGT} of individual test groups, and Table 3 summarizes the accuracy of the different algorithms as assessed by the average GPL and N_{HGT}. Supplementary Table S4 and Supplementary Table S5 further compare the accuracy of different algorithms by testing the GPL and N_{HGT} of their reconstructions using Wilcoxon signed rank tests. From these tables, we can infer that the accuracy of the algorithms ranked by GPL is largely consistent with that ranked by N_{HGT}. The only exception is that according to GPL, ClonalFrame appears to result in more accurate trees than BEAST applied to nucleotide sequences, while N_{HGT} indicates the reverse to be true. The reason behind this inconsistency appears complicated; one possible explanation is that one of the two indicators is more sensitive to the topology of the tree, while the other is more sensitive to the branch lengths. Nonetheless, both GPL and N_{HGT} support that, in general, CGP applied to amino acid sequences more accurately reconstructs phylogenetic trees than all other tested algorithms.

The CGP algorithm records the posterior tree as well as other model parameters, including ρ/μ, θ, and δ_{TE}; these posterior parameters can reflect the level of recombination and the sequence divergence within the population. For each genome group for which we applied the CGP algorithm, we took the average of the parameters of the last 200,000 MCMC steps to get the representative values of ρ/μ, θ, and δ_{TE} for the group. Table 4 summarizes the mean and standard deviation of the distributions of the posterior ρ/μ, θ, and δ_{TE}. The value of δ_{TE} is known to be around 1% ~ 3% (Fraser et al. 2007), but is measured to be from 6.1% to 13% by CGP (geometric mean in Table 4). A possible cause of this large deviation is the small number of sites per segment l_s that we used in the calculation (l_s=30 for nucleotide sequences and l_s=10 for amino acid sequences). While a small l_s (and small l_s^{cutoff}) can speed up the CGP algorithm, it also introduces uncertainties, as each additional SSP on a segment increases the segment divergence by 3.3% when l_s=30, and 10% when l_s=10. The divergence introduced by an SSP, denoted as Δ_{SSP}, is comparable to δ_{TE}, which makes the estimations deviate from the expectation. When we used larger segment sizes, l_s=300 for nucleotide sequences and l_s=100 for amino acid sequences, with l_s^{cutoff}=80 to save computational time, we obtain a smaller Δ_{SSP}; in this case, CGP estimates δ_{TE} to be around 1.4% - 3.3% (Table 5, geometric average), consistent with the literature values.

We also compared the computational cost of different algorithms. For each 10-genomes test group of both test 1 and 2, we performed phylogenetic reconstruction using CGP, BEAST, ClonalFrame on the nucleotide sequences as well as CGP and BEAST on the amino acid sequences, and measured the CPU time of each run (see Supplementary File S2 for results of individual test groups and Table 6 for average CPU times of each algorithm).

The ranking of running times of different algorithms is BEAST_n<CGP_a<CGP_n<BEAST_a<ClonalFrame; the significance of this ranking is confirmed by a comparison of CPU times using Wilcoxon signed rank tests at significance level of 0.05. Table 6 shows that, while BEAST performed on nucleotide sequences has the shortest run time, the running times of the CGP algorithm on nucleotide sequences as well as on amino acid sequences are of the same order of magnitude, and an order of magnitude shorter than those of ClonalFrame.

Discussion

In this work, we developed a coarse-graining phylogenetic (CGP) reconstruction algorithm. The model behind CGP can directly account for homologous recombination in prokaryotic genomes, which is a feature missing in many other phylogenetic reconstruction algorithms. We have conducted extensive analyses to compare the accuracy of CGP with other state-of-the-art algorithms, reconstructing ultrametric phylogenies for simulated as well as for real *E. coli* genomes. On simulated genomes, CGP performs better than other algorithms in predicting branch lengths for sets of genomes of all examined levels of recombination; CGP is also at least as good as other algorithms for topology prediction of the phylogenetic tree at intermediate or higher levels of recombination. On real *E. coli* genomes, we examined the consistency between the reconstructed phylogenetic trees and the phylogenetic signal inferred from the absence and presence of genes in the genomes; we showed that the phylogenetic tree reconstructed by CGP based on amino acid sequences is significantly more accurate than those generated by the other algorithms.

In constructing the ultrametric phylogeny, the CGP algorithm also estimates the level of recombination and sequence divergence in the population. Table 2 compares the true parameters used to simulate the genome populations with the posterior values estimated by CGP. It shows that the CGP measurements of ρ/μ, θ, and δ_{TE} is accurate within an order of magnitude. Table 4 and Table 5 shows the posterior values of ρ/μ, θ, and δ_{TE} estimated for *E. coli* genomes, one using a smaller segment size l_s and the other one using a larger l_s. It shows that while parameter estimates using a smaller l_s and a larger l_s are consistent with each other for ρ/μ, they do not agree with each other for θ and δ_{TE}. The estimate of δ_{TE} based on a larger l_s is consistent with published values (Table 5), while the same is not true for the estimate based on a smaller l_s (Table 4). A smaller l_s leads to a lower resolution of the δ_{TE} measurement and thereby reduces its accuracy. We need to compare the CGP algorithm with other state-of-the-art programs, such as ClonalOrigin (Didelot et al. 2010), in order to understand how good are the measurements of ρ/μ, θ and δ_{TE}.

The CGP algorithm is fast; its speed is independent of the sequence length of individual genomes except for calculation of the pairwise distance distributions. Its speed is comparable to that of the fastest tested algorithm (BEAST on nucleotide sequences; Table 6), and CPG is an order of magnitude faster than ClonalFrame—the only other published algorithm that accounts for homologous recombination. CGP infers the phylogeny from the pairwise SSP distributions, $f(x,t)$, of the genomes considered; $f(x,t)$ is represented by an array with l_s^{cutoff} elements in the program, which uses the same amount of memory regardless of whether a genome sequence is made up of 100 genes or 10,000 genes. The segment size l_s and the cutoff number for SSPs on a segment, l_s^{cutoff}, affect the computational cost and the accuracy of the reconstruction. The reconstructed phylogeny is more accurate if we set a high l_s^{cutoff}, such as $l_s^{cutoff}=l_s$; but the reconstruction process is faster if we use a smaller l_s. When l_s is too small, this will reduce the accuracy of the posterior parameters (ρ/μ, θ, δ_{TE}). Thus, we need to adjust the parameters in the CGP algorithm to the problem at hand in order to balance speed and accuracy, and we leave this for future work.

Materials and Methods

Coalescent framework to model a neutral population of genomes with mutation and homologous recombination

The coarse-graining phylogenetic (CGP) algorithm assumes genomes to follow dynamics described in the framework of neutral coalescent model (Kingman 2000) with homologous recombination (following the one in *Fraser et al.* (Fraser et al. 2007)). This neutral coalescent model considers a constant population of N_e nodes with non-overlapping generations, each node in the population is haploid and contains a genome. A node in one generation randomly picks another node in the previous generation as parent and inherit its genome, thereafter mutation and homologous recombination can occur on the node's genome. A genome has L_{seg} segments that represent genes, and each segment has l_s binary sites that represent nucleotide or amino acid. Mutation occurs at a rate μ per segment per generation, which mutates a random position on a segment. Recombination occurs at a rate ρ per segment per generation, where it tries to recombine with an "allele" of the segment from a random genome in the population. Recombination has a success rate of exp($-x/\delta_{\mathrm{TE}}$), where x is the divergence between the incoming foreign segment and the host segment; the unit of segment divergence, as well as δ_{TE}, can be number of single site polymorphisms (SSPs), or simply %. If the recombination succeeds, then the foreign segment will replace the host segment; otherwise, the recombination fails and the host segment will not be replaced. The average segment divergence in the population, i.e., divergence of a segment between a pair of genomes averaged over all genome pairs in the population, is denoted as $\theta=2\mu N_e$.

A model to describe the evolution of divergence distribution

We applied the theoretical model introduced in *Dixit et al.* (Dixit et al. 2015; Dixit et al. 2016), which describes the evolution of SSP distribution between a pair of genomes X and Y. Let us divide the genome of X and Y into L_{seg} consecutive segments with l_p positions, and let $f(x,t)$ be the distribution of divergence on the segments, where $x\geq0$ is segment divergence, $t\geq0$ is the coalescent time between X and Y, and $f(x,t)$ is normalized to unity.

At t=0, i.e., the time when the most recent common ancestor (MRCA) splits into the X and Y lineages, both genomes are identical, and thus $f(x,0)=1$ when $x=0$, and $f(x,t)=0$ otherwise. After the MRCA splits into two lineages, mutation and recombination occur and affect their SSP distribution, and we assumed each recombination covers only one segment. The evolution of $f(x,t)$ is described by the following equation:

$$\frac{df(x,t)}{dt} = 2\mu(f(x-\Delta_{SSP},t)-f(x,t)) + 2\rho(\int_0^{inf} dyP(x|y,\theta,\delta_{TE})f(y,t)-f(x,t)) \quad (1)$$

Here, the first term accounts for mutation, and the second term accounts for recombination. Δ_{SSP} is the increase in divergence on a segment when a mutation occurs; if we take the number of SSPs as the unit of divergence, then $\Delta_{SSP}=1$. μ is the segment mutation rate, ρ is the rate for recombination to occur on a segment. The factor 2 in both terms accounts for the fact that, mutation or recombination occurring on either X or Y will affect $f(x,t)$. In reality, recombination can cover multiple segments, and so we have $\rho=\rho_{ini}L$, where ρ_{ini} is the rate for a recombination to initiate at a segment, and L is the average length of a recombination stretch (in unit of segment). Moreover, μ and ρ change if we use a different segment size l_s, but the ratio ρ/μ remains the same.

$P(x|y,\theta,\delta_{TE})$ of Eq. (1) is the recombination kernel, which is the probability for a recombination to change the number of SSPs on a segment from y to x. θ here is the average segment divergence in the population, and δ_{TE} is the transfer efficiency with unit divergence. $P(x|y,\theta,\delta_{TE})$ is divided into three terms,

$$P(x|y,\theta,\delta_{TE}) = \Theta(y-x)A_1(x|y,\theta,\delta_{TE}) + \Theta(x-y)A_2(x|y,\theta,\delta_{TE}) + \delta(x)A_3(y,\theta,\delta_{TE})$$

Here, $\Theta(x)$ is step function, which is 1 when $x>0$ and 0 when $x\leq0$; $\delta(x)$ is Dirac delta function, where $\delta(x)=0$ when $x\neq0$, and its integral around $x=0$ gives 1. The three terms of $P(x|y,\theta,\delta_{TE})$ represent three different possible scenarios of recombination:

1. when $y>x$, recombination reduces divergence, and

$$A_1(x|y,\theta,\delta_{TE}) = \frac{1}{\theta}exp(-\frac{y}{\delta_{TE}})exp(-\frac{2x}{\theta})$$

2. when *y*<*x*, recombination increases divergence, and

$$A_2(x|y,\theta,\delta_{TE}) = \frac{1}{\theta}exp(-\frac{x}{\delta_{TE}})exp(-\frac{y}{\theta})exp(-\frac{x}{\theta})$$

3. when *y*=*x*, the transferred DNA segment fails to recombine, or the recombination does not change the divergence of the pair,

$$A_3(y,\theta,\delta_{TE}) = 1 - \int_0^y dxA_1(x|y,\theta,\delta_{TE}) - \int_y^\infty dxA_2(x|y,\theta,\delta_{TE})$$

P($x|y,\theta,\delta_{TE}$) satisfies the normalization condition

$$\int_0^\infty dxP(x|y,\theta,\delta_{TE}) = 1$$

Supplementary Figure S2 shows an example SSP distributions of the model at different coalescent time *t*, with model parameters (μ, ρ, θ, δ_{TE}) equal (0.01, 0.01, 2%, 1%).

A caveat of the recombination process modelled in this work is that, while the recombination kernel in the model in *Dixit et al.* (Dixit et al. 2015) assumes that a DNA segment transferred into another host will always result in a successful recombination, the kernel defined here also considers the case where a transferred DNA fails to recombine with the host segment. We deemed the current treatment more appropriate, as it separates the process of DNA segment transfer and the process of recombination, and assumes the rate of a DNA transfer to be constant; while in *Dixit et al.* (Dixit et al. 2015), it assumes that the rate of successful recombination is constant.

Estimating r/m—ratio of contributions to divergence by homologous recombination and by mutation

The r/m measures the ratio between the contributions to sequence divergence by homologous recombination and by mutation. A higher r/m means that recombination contributes more SSPs, while a lower r/m means mutation contributes more. We can estimate the r/m value from the model parameters (Dixit et al. 2016):

$$\frac{r}{m} \leq \frac{\rho}{\mu}min(\theta,\delta_{TE}) \qquad (2)$$

The average number of SSPs brought by mutations on a segment within a unit time is μ. The average number of recombination that covers a segment within a unit time is ρ. If $\theta \ll \delta_{TE}$, then most recombination will be successful, and the average number of SSPs introduced to a segment by a recombination is approximately θ—the average segment divergence between random genome pairs. If $\theta \gg \delta_{TE}$, then on average a successful recombination will introduce δ_{TE} SSPs to a segment.

The number of SSPs introduced by recombination is around min(θ,δ_{TE})ρ, which gives (ρ/μ)min(θ,δ_{TE}) after divided by μ. A caveat here is that, what this expression estimates is an upper bound to the true r/m; because the rate of successful recombination is lower than ρ when $\theta \gg \delta_{\mathrm{TE}}$.

Computational simulation of the neutral coalescent model

We simulated genome populations following the framework of a neutral coalescent model that has recombination; this model has been applied to understand the effect of recombination on sequence divergence (Fraser et al. 2007). We set the population size N_e=1000, each genome in the population has L_{seg}=100 segments, and each segment has l_s=1000 binary sites. Throughout a simulation, we recorded the history of node inheritance, so that we have the exact phylogenetic tree of all the nodes in the population. We allowed the simulation to last for at least 10,000 generations. Starting from the 10,000th generation, we traced for the most recent common ancestor (MRCA) of all nodes in the population; the MRCA of all nodes may not exists if the simulation does not last long enough; if the MRCA has emerged, then we stop the simulation and recorded the binary genome of all the nodes.

We performed simulation on three sets of parameters, with ρ/μ = 0.2, 5, 10 to represent prokaryotic species with low, intermediate and high level of recombination. The parameters (μ, ρ, θ, δ_{TE}) of the simulations include:

1. ρ/μ=0.2: (0.05, 0.01, 10%, 0.8%), r/m≈1.6
2. ρ/μ=5: (0.05, 0.25, 10%, 0.8%), r/m≈40
3. ρ/μ=10: (0.025, 0.25, 5%, 0.8%), r/m≈80

We repeated the simulation ten times for each parameter set, generating ten different populations to test the performance of our coarse-graining algorithm.

The r/m values of our simulations estimated from Eq. (2) are 1.6, 40, 80. The r/m values reported from a previous study for a wide scope of prokaryotic species ranges from 0.02 to 63.6 (Vos and Didelot 2009); hence the parameters we picked coincides can represent the levels of recombination occurred in nature, and is suitable for testing the performance of different phylogenetic reconstruction algorithms.

We picked genomes from the simulated populations, and used them to test the performance of different phylogenetic algorithms. If we randomly pick the genomes from a population, then their root nodes is likely to have an age t_{root}~1000 because N_e=1000. However, we need a set of genomes such that one frequently recombines with each other's DNA segments; therefore, the random genomes are picked with the constraint that, the age of their MRCA should be small, i.e., $t_{\mathrm{root}} \ll 1000$; this constraint allows us to pick the genomes that are likely to have exchanged their DNA segments.

We simulated ten populations for each of the three levels of of recombination. In each simulated population, we picked ten sets of genomes, each set with different constraints on the age of their MRCA: t_{root} ~ 10, 20, …, 100. We allowed each set to have number of genomes lied in a small range 10≥n≥4, as it is not always possible to find a lot of genomes with root age lied within a particular range. This generates 100 groups of genomes for each ρ/μ value.

Evaluating the fit of a theoretical SSP distribution to an empirical SSP distribution

We need to fit an empirical SSP distribution with a theoretical distribution in order to infer the coalescent time of a genome pair. Let us consider a pair of genomes X and Y that are divided into L_{seg} segments. Let $g_{XY}(x)$ to be number of segments with divergence x, and so $g(x)$ is normalized to L_{seg}. Further, let us denote the theoretical distribution as $f_{\mu,\rho,\theta,\delta TE}(x,t)$, which is normalized to unity when it integrates over x. The probability to observe the empirical distribution $g_{XY}(x)$ given the theoretical distribution $f_{\mu,\rho,\theta,\delta TE}(x,t)$ is

$$\prod_{x}[f_{\mu,\rho,\theta,\delta_{TE}}(x,t)]^{g_{XY}(x)}$$

which, if we take the logarithm on this term, becomes the cross entropy (Boer et al. 2005).

Suppose that we have n genomes, denoted as X_1, X_2, …, X_n, and suppose that we describe their phylogeny of inheritance with an ultrametric tree T; we assumed that the $n(n-1)/2$ pairwise SSP distributions evolve according to the the neutral coalescent model with parameters μ, ρ, θ, δ_{TE}. Let us also denote the coalescent time of genome pair X_a and X_b inferred from the tree T to be $t_T(X_a,X_b)$. The logarithm of the posterior probability to observe the $n(n-1)/2$ empirical pairwise SSP distributions given the model with parameters (μ, ρ, θ, δ_{TE}) and the ultrametric tree T, denoted as $S(X_1,X_2, \ldots,X_n|\mu, \rho, \theta, \delta_{TE},T)$, is the summation of the $n(n-1)/2$ cross entropy:

$$S(X_1,\ldots X_n|\mu,\rho,\theta,\delta_{TE},T)=\sum_{all\ (X_a,X_b)\ pairs} log\{\prod_{x}[f_{\mu,\rho,\theta,\delta_{TE}}(x,t_T(X_a,X_b))]^{g_{X_aX_b}(x)}\} \quad (3)$$

Markov chain Monte Carlo simulation to search for the best fit ultrametric tree and model parameters

The CGP algorithm performs Markov chain Monte Carlo (MCMC) simulation to reconstruct the ultrametric phylogenetic tree of genomes. Given n genomes, the CGP algorithm started by calculating the distance matrix of the n genomes based on their pairwise sequence divergence, and inferred the ultrametric tree from the distance matrix using single linkage clustering; this tree is used at the start of the MCMC simulation. CGP algorithm calculated the theoretical SSP distribution at different discrete time steps, and fitted these theoretical distributions to the empirical distributions; thus the age of an internal node inferred by CGP is always an integer. At the start, we set the numerical value of the mutation rate to be μ=min(0.02, $<x_{farthest}>$/200) (unit: number of mutation per segment per time step); here, $<x_{farthest}>$ is the average number of SSPs per segment between the most divergent pair among the n genomes.

To calculate the numerical value of the SSP distribution $f_{\mu,\rho,\theta,\delta TE}(x,t)$, we represented the SSP distribution as a vector in the program. Each segment can have possible l_s+1 states, corresponding to 0, 1, 2, …, l_s SSPs in the segment; thus a vector of l_s+1 elements is appropriate to represent $f_{\mu,\rho,\theta,\delta TE}(x,t)$. The processes of mutation and homologous recombination can be represented as an (l_s+1)×(l_s+1) matrix, which transforms the SSP distribution at time t to time t+1. However, to save computational resources, we set a cutoff $l_s^{cutoff} \leq l_s$, where segments with number of SSPs greater than l_s^{cutoff} are considered as having l_s^{cutoff} SSPs. This artificially reduces computational time, at the expense of accuracy of the phylogenetic reconstruction.

After the initial tree T and the initial parameters (μ, ρ, θ, δ_{TE}) are determined, the MCMC proceeds. In each step, one of the following moves is considered:

1. There is a $n^{-2}/2$ chance to mutate one of the parameters (ρ, θ, δ_{TE}). The algorithm considers six different parameter sets ($\rho(1+\varepsilon)$, θ, δ_{TE}), ($\rho(1-\varepsilon)$, θ, δ_{TE}), (ρ, $\theta(1+\varepsilon)$, δ_{TE}), (ρ, $\theta(1-\varepsilon)$, δ_{TE}), (ρ, θ, $\delta_{TE}(1+\varepsilon)$), ($\rho$, θ, $\delta_{TE}(1-\varepsilon)$), with random variable ε, $1 \gg \varepsilon > 0$. Absolute upper limits are imposed for some of the parameters: $\rho<1$,$\theta<100\%$ and $\delta_{TE}<100\%$. These new parameter sets, along with the original one, is selected according to their relative posterior probabilities.
2. Else, there is a $n^{-2}/2$ chance for a random branch of the tree to be cut and grafted to a different part of the tree. In this move, a branch with the younger internal node Y and older internal node O is picked randomly, with their ages denoted as t_Y and t_O. This branch is then cut at the height t_O, and the entire sub-clade is then grafted to all other branches that are present at height t_O to generate new ultrametric trees. All these new trees, along with the original one, is selected according to their relative posterior probabilities.
3. Else, an internal node is picked randomly and moved one time step upwards or downwards to generate a new tree. If the picked node cannot move in the chosen direction, as it is blocked by another node, then the algorithm will swap the branches to produce new trees (see Supplementary Figure S3 for illustration); one of these new trees, along with the original tree, is selected according to their relative posterior probability.

The simulation continues, and stops when the maximal score (logarithm of the posterior probability) of the chain of MCMC steps has not increased by more than 1 for the last

200,000 steps. (see Supplementary File S3 for source code).

Constructing the SSP distribution of real genomes

In this study, we used 55 *E. coli* and *Shigella* genomes to test our model algorithm (see Supplementary Table S1 for the list of 55 genomes). Let us simply call all these strains *E. coli*, as *Shigella* strains are sometimes considered as a subclade of *E. coli*. We performed the following procedure to prepare the sequence of each strain for phylogenetic reconstruction:

1. We created a file in FASTA format that contains the amino acid sequence of the genes (CDS features in the Genbank files) for each strain; the corresponding nucleotide sequences of the genes of the same strain is stored in another FASTA file; thus a strain has two FASTA file, both have the same number of genes.
2. We identified the orthologous gene families by performing the Proteinortho program (Lechner et al. 2011) on the amino acid FASTA file of the 55 strains.
3. For each orthologous gene family that is universal to 55 strains and has only one allele on each strain, we aligned the nucleotide sequence of its alleles using MAFFT (Katoh and Standley 2013) with options “--maxiterate 1000” and “--localpair”; we performed another alignment on the amino acid sequences of the orthologous gene family using the same MAFFT settings.
4. For each alignment (nucleotide acid or amino acid), we removed the positions with a dash, so as to make the alleles of the orthologous gene family to have equal length; moreover, positions with ‘J’ on amino acid alignments are also removed, as ‘J’ represents an ambiguous amino acid and may trigger an error in the BEAST program.

While we can single out the orthologous gene families that are universal to all 55 strains and do not have paralogs on the genomes, and concatenate their alleles to make a ‘super-gene’ to represent each strain, we avoided this approach because subsequent phylogenetic reconstruction using ClonalFrame can last for more than a week. Instead, we randomly selected 100 orthologous gene families, concatenated their alleles to generate ‘super-gene’ that represent different strains.

Since the CGP algorithm takes the distribution of pairwise SSPs on the genome segments as input, we divided the sequence of the chosen 100 genes in segments; we used segment size l_s=30, 300 for nucleotide and l_s=10, 100 for amino acid sequences, discarding the last segment of each gene if it has fewer than l_s sites, and calculated the SSP distributions based on those segments.

Testing algorithms of phylogenetic reconstruction with simulated genomes

We simulated the neutral coalescent model to generate genome data, and used them to test the performance of different phylogenetic algorithms, including our CGP algorithm, RAxML (Stamatakis 2014), BEAST (Drummond et al. 2012) and ClonalFrame (CF) (Didelot and Falush 2007). Given n binary genomes generated in the simulation, we calculated the $n(n\text{-}1)/2$ pairwise SSP distributions and applied CGP to infer their ultrametric tree; the CGP algorithm lasts for at least 200,000 steps, and it records the posterior parameters and posterior tree every 1,000 steps. The CGP algorithm terminates when the logarithm of the posterior probability does not increase by more than 1 for 200,000 MCMC steps, and we collected the data generated in the last 200,000 steps for analysis.

For RAxML, we applied the substitution model BINGAMMA, which is suitable for binary data; we also used the rapid bootstrap options '-x' and '-N 200' in RAxML, which carried out 200 ML searches on 200 randomized stepwise addition parsimony trees; this generated a series of 200 posterior trees for further analysis.

For BEAST, we converted the 0 and 1 in the binary genomes into A and T, and applied the default nucleotide substitution model HKY (Hasegawa et al. 1985), strict clock, constant size coalescence (Kingman 1982; Drummond et al. 2002), along with other default settings, to perform the phylogenetic reconstruction and record the posterior tree. We discarded the first 25% of the posterior trees in the series generated by BEAST, as they might not have reached equilibrium, and collected the remaining 75% of the trees for analysis.

For ClonalFrame, we converted the binary genomes into sequences of A and T, and fed them into the ClonalFrame program with default setting. We manually extracted the posterior tree series from the ClonalFrame output file using a sister program in the ClonalFrame package, and used the entire tree series for analysis.

We considered three sets of parameters in our simulation, which correspond to populations of prokaryotes with low, intermediate and high level of recombination (ρ/μ = 0.2, 0.5, 10, see above sessions). We prepared 100 groups of genomes for each parameter set, and applied CGP, RAxML, BEAST and ClonalFrame to reconstruct their phylogenies. Each of the four algorithms outputs a series of trees, and we compared the topology and branch length of the reconstructed trees in the series with authentic tree of the genomes.

To appraise the accuracy of the topology predictions of different algorithms, we defined the topology similarity score s_{topo}, also denoted as 'efficiency' (Didelot and Falush 2007), which is the probability for an internal node of the authentic tree, excluding the root node, to find its corresponding internal node on a posterior tree that groups the leaves into ingroup and outgroup the same way as it does. Suppose the true phylogeny of five genomes

A, B, C, D, E is described by tree T_0, and an internal node m in T_0 has A and B as ingroup leaves; if T_i is a tree in the posterior tree series generated by a phylogenetic reconstruction algorithm to represent the phylogeny of the five genomes, and there exists a node in T_i with exactly A and B as ingroup leaves, then node m in T_0 has a corresponding node in T_i that divides the leaves into two groups as m does. Moreover, if a reconstructed tree T_j has an internal node with ingroup leaves C, D and E, then m also has its corresponding node in T_j because T_j groups A and B together in the outgroup; this allows the topology score compares not only rooted ultrametric trees, but also unrooted tree such as those reconstructed by RAxML. The topology similarity score is bounded, $0 \le s_{topo} \le 1$, and the higher the score the more accurate are the reconstructed posterior trees.

To evaluate the deviation of the node ages and branch lengths between the authentic tree and the posterior trees of different algorithms, we defined the node age deviation d_{age}, which is the error between the age (normalized by the total branched length of the tree) of an internal node in the authentic tree and that of its corresponding node in a posterior tree. Let m be an internal node in the authentic tree T_0, and let τ_m be the normalized age of the authentic tree, i.e., age of m divided by the total branch length of the tree. Also, let m'_T be the corresponding node on a reconstructed tree T that divides the leaves in the same way that T_0 does; an internal node in T_0 that does not have a corresponding node in T is not considered. The node age deviation is defined by the following expression:

$$d_{age} = \sqrt{\frac{\sum_T \sum_m (\tau_m - \tau_{m'_T})^2}{\sum_T \sum_m 1}}$$

Since the node age deviation is like a standard error, it is bounded below by zero, $d_{age} \ge 0$, and the smaller the deviation the more accurate is the node age prediction.

Testing phylogenetic reconstruction algorithms with real *E. coli* genomes

We also tested the accuracy of CGP, BEAST and ClonalFrame on real genomic sequences, using different combinations of genomes chosen from the 55 *E. coli* strains. We represent each strain its concatenated nucleotide sequence and also amino acid sequence of the core genes. Moreover, as pointed out in *Dixit et al.* (Dixit et al. 2015), when the nucleotide sequence divergence between a pair of *E. coli* genomes goes beyond 1.3%, all their segments have been recombined after their separation from the MRCA. Since recombination erases clonal signal, we expected difference in the accuracy of phylogenetic reconstruction using CGP. Hence, we separately tested the algorithms using genomes with lower and higher divergence. We conducted two tests, each with its own constraint to select the strains in the test groups:

1. low divergence strains: the pairwise nucleotide sequence divergence between all pairs in a group is ≤1.3%;
2. high divergence strains: no constraint on sequences divergence;

We generated 100 genome groups in each test, in each group ten strains are randomly chosen from the 55 *E. coli* strains, following the criterion of the test; further, instead of concatenating sequences of all orthologous gene families that are universally present in the 55 strains to make a ‘super-gene’ for each strain, we randomly chose 100 universal orthologous gene families to represent the strains in the group to save computational resources. We concatenated the nucleotide sequences of the 100 chosen orthologous gene families to make a concatenated nucleotide sequence for each of the ten strains; we also concatenated the amino acid sequences of the 100 chosen orthologous gene families to make a concatenated amino acid sequences for each of the ten strains (see Supplementary File S2 for the strains and orthologous gene families chosen in each test group). There are 45 pairs in a ten strain group, we calculated the 45 SSP distributions based on the segments of nucleotide sequences of 100 orthologous gene families, and another 45 SSP distributions based on the segments of the amino acid sequences of the 100 orthologous gene families, to prepare for the CGP algorithm (see above sections). We then performed phylogenetic reconstruction to infer the ultrametric tree of each 10-strain-group in five different ways:

1. CGP on nucleotide sequences (l_s=30, l_s^{cutoff}=30, and also l_s=300, l_s^{cutoff}=80);
2. BEAST (Drummond et al. 2012) on nucleotide sequences with HKY substitution model (Hasegawa et al. 1985), strict clock, constant size coalescence (Kingman 1982; Drummond et al. 2002) and other default settings;
3. ClonalFrame (Didelot and Falush 2007) with default setting on the nucleotide sequences;
4. CGP on amino acid sequences (l_s=10, l_s^{cutoff}=10, and also l_s=100, l_s^{cutoff}=80);
5. BEAST (Drummond et al. 2012) with Blosum2 substitution model (Henikoff and Henikoff 1992), strict clock, constant size coalescent (Kingman 1982) and other default settings.

We then summarized the results of each phylogenetic reconstruction using a representative tree. For CGP, we collected the posterior trees generated in the last 200,000 steps, applied ‘treeannotator’—a program that is bundled with the BEAST package (Drummond et al. 2012)—to calculate the maximum clade credibility tree and use it to be the representative tree of the phylogenetic reconstruction; for BEAST, we discard the first 25% of the posterior trees and summarised the remaining 75% using its default program ‘treeannotator’ to generate its representative tree; for ClonalFrame, its output file already contains one consensus tree, and we used it as the representative tree.

We evaluated the accuracy of the representative tree by comparing the representative tree with the phylogenetic signals encoded in the absence and presence of genes across different genomes. We considered each internal node of the representative tree to be an ancestral strain, and used the maximum likelihood algorithm GLOOME (Cohen et al. 2010), along with the default parameters of the online version of GLOOME (Evolutionary model: fixed gain/loss ratio, rate distribution Gamma), to reconstruct the presence and absence of different orthologous gene families in the ancestral strains. GLOOME reconstructs the ancestral genomes based on the representative tree and the gene profile—the presence and absence of different orthologous gene families across the strains considered. We used Proteinortho (Lechner et al. 2011) to map the orthologous gene families in the 55 genomes, and an orthologous gene family is present in an extant strain if there is one or more alleles there, and absent otherwise. GLOOME reports the probability for different orthologous gene families to be present in the ancestral genomes, and also the GLOOME posterior likelihood (GPL) for the ancestral genome reconstruction. We used GPL to quantify the accuracy of the representative tree, because the higher the GPL, the more consistent is the representative tree with the phylogeny inferred from the absence and presence of genes across different genomes.

Furthermore, we evaluated the accuracy of the representative tree by analyzing the horizontal gene transfer events that it infers. Using the output data of GLOOME, we considered orthologous gene families with present probability $P \geq 0.5$ in an ancestral genome to be present, and $P<0.5$ to be absent. Orthologous gene families that are not presents in the ancestor of a branch but present in the descendent of a branch are transferred into the branch horizontally. Since multiple orthologous gene families can be added in a single horizontal gene transfer (HGT) event, we used a greedy algorithm to group the genes transferred into the same branch into HGT events. Assuming that genes transferred together will not get separated into different clusters on the genome, we put two transferred genes into the same HGT event if they are located on any 30kb segments in any of the 55 extant genomes. We used 30kb as the capacity of the HGT agent, because the length distribution of DNA segments acquired horizontally cuts off at a distance of 30 kb (Bobay et al. 2014; Pang and Lercher 2016). Let us denote a set **S** to contain all the orthologous gene families transferred into a branch; the procedure of the greedy algorithm to group transferred genes into HGT event includes:

1. identify the start positions of every gene in **S** in all 55 genomes;
2. pick a random gene g_A in **S** and put it in a new set **P**;
3. for each gene g not included in **P** (represented by $g \notin \mathbf{S} \backslash \mathbf{P}$ in set theory convention), enumerate the number of extant genomes that accommodate it along with other genes included in **P** within a 30 kb segment;

4. pick the one gene outside **P** supported by the highest segment count, and add it to **P**; if there are multiple genes that can be chosen, pick one randomly;
5. repeat step 3 - 4, test each remaining gene in **S** outside **P** by enumerating the genomes that support its grouping with other genes in **P**; the one gene with the highest support is then added to **P**;
6. when no more genes can be added to **P**, the genes in **P** are then grouped into an HGT event; these genes are removed from **S**, and **P** is emptied; step 2 - 5 is repeated to reconstruct another HGT event, until every gene is assigned to an HGT event.

In this way, we grouped all the genes transferred into the branches of the representative tree into different HGT events, and the number of HGT events is denoted as N_{HGT}. We used N_{HGT} to quantify the accuracy of the representative tree, because the more the representative tree deviates from the authentic phylogeny, the more likely for GLOOME to assign co-transferred genes into different branches, and the higher N_{HGT} gets.

Measuring the cpu-time of phylogenetic reconstruction of different algorithms

We have also measured the computational cost of different algorithms. For each 10-genomes test group of both test 1 and 2, we performed phylogenetic reconstruction using CGP, BEAST, ClonalFrame on the nucleotide sequences, and CGP, BEAST on the amino acid sequences. As every algorithm that we tested is single-threaded, we assigned each run a cpu-core of ‘Intel(R) Xeon(R) CPU E5-2670 0 @ 2.60GHz’ with operating system ‘Scientific Linux release 6.5 (Carbon)’, and used the ‘Benchmark’ package in perl to measure its wall-clock time of each run (see Supplementary File S2 for the table of computational costs).

Acknowledgement

This work was supported by the German Research Foundation (DFG grant CRC 680 to Martin Lercher). We would like to thank Martin Lercher for helpful comments and advice.

Tables

Table 1. A summary of average topology similarity score s_{topo} and average node age deviation d_{age} for different algorithms. Cells with dark background correspond to the best algorithm for topology and node age prediction at different recombination level.

HR[1] level	ρ/μ	topology similarity score s_{topo}				node age deviation d_{age}			
		CGP	RAxML	BEAST	CF[2]	CGP	RAxML	BEAST	CF[2]

low	0.2	0.9725	0.9946	0.9971	0.9847	0.0073	-	0.0154	0.0214
intermediate	5	0.9426	0.9167	0.9244	0.8708	0.0107	-	0.0252	0.0412
high	10	0.9297	0.9024	0.9118	0.8134	0.0214	-	0.0312	0.0456

[1]HR: homologous recombination

[2]CF: ClonalFrame

Table 2. Authentic parameters including ρ/μ, θ, δ_{TE} and their posterior counterparts measured by CGP averaged over different genome test groups. Extreme values are removed from the calculation of θ and δ_{TE} (θ>90% or δ_{TE}>90%).

recombination level	Parameters used in simulation (ρ/μ, θ, δ_{TE})	posterior (ρ/μ, θ^*, δ_{TE}^*)	
		arithmetic avg / SD	geometric avg / SD
low	(0.2, 10^*, 0.8^*)	(2.6 ± 8.4, $13\pm24^*$, $7\pm21^*$)	($0.19\times e^{0\pm2.4}$, $2.7\times e^{0\pm1.8*}$, $0.48\times e^{0\pm2.1*}$)
intermediate	(5, 10^*, 0.8^*)	(3.4 ± 1.5, $49\pm28^*$, $1.3\pm3.1^*$)	($3.1\times e^{0\pm0.46}$, $34\times e^{0\pm1.1*}$, $1.3\times e^{0\pm0.18*}$)
high	(10, 5^*, 0.8^*)	(6.4 ± 4.4, $13\pm19^*$, $2.5\pm6.3^*$)	($5.4\times e^{0\pm0.58}$, $5.8\times e^{0\pm1.2*}$, $1.7\times e^{0\pm0.55*}$)

*unit: %

Table 3. Average accuracy of CGP, BEAST and ClonalFrame on nucleotide and amino acid sequences, tested with sequences of real *E. coli* genomes, measured with GPL (the higher the more accurate is the tree) and N_{HGT} (the lower the more accurate is the tree) as indicators. Test 1 examined groups of real genomes with lower divergence, and Test 2 examined groups of real genomes with higher divergence. The best algorithm in a test is highlighted with a dark background.

		nucleotide			amino acid	
		CGP_n	$BEAST_n$	CF^2	CGP_a	$BEAST_a$
Test 1	<GPL>*	-26614	-27573	-26458	-25777	-26435
	<N_{HGT}>*	931.9	940.9	1145.6	900.5	934.2
Test 2	<GPL>*	-31483	-32336	-31451	-30590	-31112
	<N_{HGT}>*	1132.1	1145.7	1367.5	1094.1	1103.5

*<> represents averaging over different test groups

$_n$ performed on nucleotide sequences

$_a$ performed on amino acid sequences

[2]CF: ClonalFrame

Table 4. The arithmetic and geometric average and standard deviation of the posterior ρ/μ, θ and δ_{TE} in Test 1 (groups of real genomes with lower divergence) and Test 2 (groups with

higher divergence), measured by CGP in different tests, using l_s=30 for nucleotide sequences, l_s=10 for amino acid sequences, and l_s^{cutoff}=l_s. Each test involves the phylogenetic reconstruction of 100 genome groups. Extreme data points (θ>90%, δ_{TE}>90%) are removed from calculation of mean and standard deviation.

		arithmetic avg / SD			geometric avg / SD		
		ρ/μ	θ^*	δ_{TE}^*	ρ/μ	θ^*	δ_{TE}^*
Test 1	nucleotide l_p=30	12±5.0	12±9.0	6.2±0.88	$10\times e^{0\pm0.47}$	$11\times e^{0\pm0.43}$	$6.1\times e^{0\pm0.13}$
	amino acid l_p=10	7.5±8.4	34±18	17±15	$3.3\times e^{0\pm1.5}$	$30\times e^{0\pm0.51}$	$13\times e^{0\pm0.68}$
Test 2	nucleotide l_p=30	7.5±3.2	14±6.9	6.1±0.68	$6.8\times e^{0\pm0.43}$	$13\times e^{0\pm0.35}$	$6.1\times e^{0\pm0.11}$
	amino acid l_p=10	7.4±11	36±15	17±13	$3.1\times e^{0\pm1.4}$	$32\times e^{0\pm0.49}$	$14\times e^{0\pm0.58}$

*unit: %

Table 5. The arithmetic and geometric average and standard deviation of the posterior ρ/μ, θ and δ_{TE} in Test 1 (groups of real genomes with lower divergence) and Test 2 (groups with higher divergence), measured by CGP in different tests, using l_s=300 for nucleotide sequences, l_s=100 for amino acid sequences, and l_s^{cutoff}=80. Each test involves the phylogenetic reconstruction of 100 genome groups. Extreme data points (θ>90%, δ_{TE}>90%) are removed from calculation of mean and standard deviation.

		arithmetic avg / SD			geometric avg / SD		
		ρ/μ	θ^*	δ_{TE}^*	ρ/μ	θ^*	δ_{TE}^*
Test 1	nucleotide l_p=300	14±9.9	7.5±9.0	10±17	$10\times e^{0\pm0.78}$	$4.5\times e^{0\pm0.95}$	$3.3\times e^{0\pm1.3}$
	amino acid l_p=100	10±13	13±15	3.4±6.7	$5.2\times e^{0\pm1.4}$	$6.9\times e^{0\pm1.1}$	$2.2\times e^{0\pm0.67}$
Test 2	nucleotide l_p=300	8.6±5.9	42±13	1.6±2.4	$7.3\times e^{0\pm0.57}$	$37\times e^{0\pm0.62}$	$1.4\times e^{0\pm0.33}$
	amino acid l_p=100	7.9±5.6	15±14	2.3±1.3	$5.9\times e^{0\pm0.89}$	$9.3\times e^{0\pm0.94}$	$2.2\times e^{0\pm0.34}$

*unit: %

Table 6. Average cpu-time of different phylogenetic reconstruction algorithms. The fastest algorithm, BEAST on amino acid sequence, is highlighted with a grey background. For CGP on nucleotide sequences, l_p=30; For CGP on amino acid sequences, l_p=10.

	nucleotide			amino acid	
	CGP$_n$	BEAST$_n$	CF^2	CGP$_a$	BEAST$_a$
<cpu-time> (second)	1570	706	30423	810	79239

*<> represents averaging over different groups

$_n$ performed on nucleotide sequences

$_a$ performed on amino acid sequences

^{2}CF: ClonalFrame

References

Bobay L-M, Touchon M, Rocha EPC. 2014. Pervasive domestication of defective prophages by bacteria. Proc. Natl. Acad. Sci. U. S. A. 111:12127–12132.

Boer P-T de, Kroese DP, Mannor S, Rubinstein RY. 2005. A Tutorial on the Cross-Entropy Method. Ann. Oper. Res. 134:19–67.

Cohen O, Ashkenazy H, Belinky F, Huchon D, Pupko T. 2010. GLOOME: gain loss mapping engine. Bioinformatics 26:2914–2915.

Croucher NJ, Mostowy R, Wymant C, Turner P, Bentley SD, Fraser C. 2016. Horizontal DNA Transfer Mechanisms of Bacteria as Weapons of Intragenomic Conflict. PLOS Biol. 14:e1002394.

Didelot X, Falush D. 2007. Inference of Bacterial Microevolution Using Multilocus Sequence Data. Genetics 175:1251–1266.

Didelot X, Lawson D, Darling A, Falush D. 2010. Inference of Homologous Recombination in Bacteria Using Whole-Genome Sequences. Genetics 186:1435–1449.

Dixit PD, Pang TY, Maslov S. 2016. Recombination-driven genome evolution and stability of bacterial species. bioRxiv:067942.

Dixit PD, Pang TY, Studier FW, Maslov S. 2015. Recombinant transfer in the basic genome of Escherichia coli. Proc. Natl. Acad. Sci. U. S. A. 112:9070–9075.

Drummond AJ, Nicholls GK, Rodrigo AG, Solomon W. 2002. Estimating mutation parameters, population history and genealogy simultaneously from temporally spaced sequence data. Genetics 161:1307–1320.

Drummond AJ, Suchard MA, Xie D, Rambaut A. 2012. Bayesian phylogenetics with BEAUti and the BEAST 1.7. Mol. Biol. Evol.:mss075.

Fraser C, Hanage WP, Spratt BG. 2007. Recombination and the nature of bacterial speciation. Science 315:476–480.

Hasegawa M, Kishino H, Yano T. 1985. Dating of the human-ape splitting by a molecular clock of mitochondrial DNA. J. Mol. Evol. 22:160–174.

Henikoff S, Henikoff JG. 1992. Amino acid substitution matrices from protein blocks. Proc. Natl. Acad. Sci. U. S. A. 89:10915–10919.

Huddleston JR. 2014. Horizontal gene transfer in the human gastrointestinal tract: potential spread of antibiotic resistance genes. Infect. Drug Resist. 7:167–176.

Katoh K, Standley DM. 2013. MAFFT multiple sequence alignment software version 7: improvements in performance and usability. Mol. Biol. Evol. 30:772–780.

Kingman JFC. 1982. The coalescent. Stoch. Process. Their Appl. 13:235–248.

Kingman JFC. 2000. Origins of the Coalescent: 1974-1982. Genetics 156:1461–1463.

Lang AS, Zhaxybayeva O, Beatty JT. 2012. Gene transfer agents: phage-like elements of genetic exchange. Nat. Rev. Microbiol. 10:472–482.

Lechner M, Findeiß S, Steiner L, Marz M, Stadler PF, Prohaska SJ. 2011. Proteinortho: Detection of (Co-)orthologs in large-scale analysis. BMC Bioinformatics 12:1–9.

Ochman H, Lawrence JG, Groisman EA. 2000. Lateral gene transfer and the nature of bacterial innovation. Nature 405:299–304.

Pál C, Papp B, Lercher MJ. 2005. Horizontal gene transfer depends on gene content of the host. Bioinformatics 21:ii222-ii223.

Pang TY, Lercher M. 2016. Supra-operonic clusters of functionally related genes (SOCs) are a source of horizontal gene co-transfers. ArXiv160207266 Q-Bio [Internet]. Available from: http://arxiv.org/abs/1602.07266

Schierup MH, Hein J. 2000. Consequences of recombination on traditional phylogenetic analysis. Genetics 156:879–891.

Spratt BG. 1999. Multilocus sequence typing: molecular typing of bacterial pathogens in an era of rapid DNA sequencing and the internet. Curr. Opin. Microbiol. 2:312–316.

Stamatakis A. 2014. RAxML Version 8: A tool for Phylogenetic Analysis and Post-Analysis of Large Phylogenies. Bioinformatics:btu033.

Takeuchi N, Kaneko K, Koonin E. 2014. Horizontal Gene Transfer Can Rescue Prokaryotes from Muller’s Ratchet: Benefit of DNA from Dead Cells and Population Subdivision. G3 GenesGenomesGenetics 4:325–339.

Vos M, Didelot X. 2009. A comparison of homologous recombination rates in bacteria and archaea. ISME J. 3:199–208.

Wilson GG, Murray NE. 1991. Restriction and Modification Systems. Annu. Rev. Genet. 25:585–627.

Supplementary Tables

Supplementary Table S1. List of 55 genomes of *E. coli* and *Shigella* analysed in this study for model testing.

Supplementary Files

Supplementary File S1. Scores of different algorithms to reconstruct the phylogeny of simulated genomes.

Supplementary File S2. *E. coli* strains and genes used in different test groups, as well as their GPL, N_{HGT} and computational costs of the phylogenetic reconstructions.

Supplementary File S3. Source code to perform the CGP algorithm.