This article is intended for submission as an Article of in the tree of life issue 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 stretch replaces a similar stretch 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 stretch that is similar to the other genome of the pair, thereby reducing their pairwise sequence divergence; (ii) a genome can also recombine with a stretch 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 reconstructs ultrametric-trees while explicitly accounting for recombination. Instead of considering individual positions of genome sequences, we use a coarse-graining approach, which divides a genome sequence into short segments to account for local density of nucleotide-substitution. 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 fit each empirical SSP-distribution to a theoretical SSP-distribution. We test the accuracy 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 prediction of internal nodes by CGP is more accurate than other algorithms, while the tree topology is at least as accurate.

Conclusion

The CGP algorithm is more accurate and faster than alternative recombination-aware methods for ultrametric phylogenetic reconstructions.

Introduction

Horizontal transfer of DNA stretches 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 stretch that enters the prokaryotic cell and survives these host defenses may be incorporated into the host

genome. If the incoming DNA stretch is highly similar to a stretch on the host genome, then homologous recombination may occur, where the incoming DNA stretch homologously recombines with the host stretch and overwrites it (Dixit et al. 2015). Apart from recombination, the incoming stretch may also be inserted directly into the host genome through non-homologous recombination.

Horizontal gene transfer allows the fast spread of beneficial genes, facilitating prokaryotes to adapt to changes in the environment; for example, HGT is responsible for 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 can severely disturb phylogeny reconstructions; its effect on genome divergence is complex, as it can first speed up the divergence of a genome pair and then slow it down (Dixit et al. 2016). 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 stretch of genome X recombines with a DNA stretch from genome Y, it will erase some of the single site polymorphisms (SSPs) that previously differentiated X and Y, shortening the apparent distance between the genomes; here, an SSP refers either to single nucleotide polymorphism (SNP) or to single amino acid polymorphism (SAP). Conversely, (ii) when X recombines with a DNA stretch 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, there are several recombination-aware algorithms, including ClonalFrame (with sister algorithms

ClonalOrigin, ClonalFrameML, and Bacter package in BEAST2 that implements ClonalOrigin), Gubbins (Didelot and Falush 2007; Didelot et al. 2010; Croucher et al. 2015; Didelot and Wilson 2015; Vaughan et al. 2016); there are also non-phylogenetic algorithms detecting recombinations, such as BratNextGen and fastGEAR (Marttinen et al. 2012; Croucher et al. 2016). While these algorithms can ascribe genomic stretches with high number of substitutions to recombination with distant strains and account for type (ii) recombination, they do not take 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 all variable nucleotide / amino acid 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 their true phylogeny. We tested the accuracy of CGP tree topology and branch length predictions with other state-of-the-art algorithms that reconstructs ultrametric trees, on both simulated and real *E. coli* genomes. These algorithms include BEAST and ClonalFrame (Didelot and Falush 2007; Drummond et al. 2012); additionally, we also tested RAxML (Stamatakis 2014), a popular and fast algorithm, but not ultrametric and not recombination-aware. The only other ultrametric and recombination-aware algorithm is Bacter in BEAST2; but as we tested Bacter on *E. coli* genomes following the same procedure that we did on the other algorithms, we found that the test-runs cannot finish within one week; therefore we considered Bacter to be not suitable for routine genome scale phylogenetic reconstruction.

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; Dixit et al. 2016) that quantitatively describes the evolution of genomic sequence divergence in a neutral coalescent framework (Materials and Methods). Recombination can introduce DNA stretches characterized by high density of substitutions; for this reason, CGP considers genomic segments, where a segment contains a number of sites. A genome is represented by a chain of non-overlapping

segments, as density of substitutions can be defined on segments. The size of a segment should be reasonably small, so that a recombination stretch can cover multiple segments, but a segment will not overlap with multiple recombination stretches. 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; it then enumerates positions with SSPs within each segment to obtain the SSP distribution of the genome pair. 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 an 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 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. The source code of CGP is available at https://github.com/TinPang/coarse-graining-phylogenetics. Notice that the algorithm takes the alignments of each core genes as input, instead of a signal concatenated alignment like most other phylogenetic algorithm.

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 genome pair. The unit of divergence θ and efficiency δ_{TE} can be either (a) the number of SSPs per segment or (b) the corresponding density in percentage.

To reconstruct the phylogenetic inheritance of n genomes, the CGP algorithm starts from the $n(n\text{-}1)/2$ 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 $n(n\text{-}1)/2$ 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 $n(n\text{-}1)/2$ empirical SSP distribution $g(x)$, where x is the number of SSPs on a segment, has a corresponding theoretical distribution $f(x)$; these $n(n\text{-}1)/2$ theoretical distributions depend on μ, ρ, θ, δ_{TE}, and T. We used the (negative) cross entropy (Rubinstein and Kroese 2004; Boer et al. 2005) as a score to measure the similarity between an empirical distribution $g(x)$ and its corresponding theoretical distribution $f(x)$, because the logarithm of the posterior probability of a point in the solution space—described by μ, ρ, θ, δ_{TE} and T—multiplied by -1, is the summation of the $n(n\text{-}1)/2$ 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. The algorithm terminates when the maximum of the chain of 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 accuracy of CGP with that of other algorithms, we performed forward-in-time simulation on populations of haploid genomes following the neutral coalescent model with recombination (Fraser et al. 2007), and selected genomes from these populations to test the phylogenetic algorithms. We also generated genomes using SimBac—the only coalescent-based simulation algorithm that has feature of recombination—to generate genome sequences that mimic those from a large population (Brown et al. 2016)**.**

In the forward-simulations, each genome in a population is made of 100 stretches; each stretch has 1000 binary sites, and each recombination transfers one stretch. We used 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, 5%, 0.8%)); note that the unit of μ and ρ here is per stretch per time step. These parameter sets correspond to species with low, intermediate, and high levels of recombination. 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 close to previous reports of 2.2% (Fraser et al. 2007) and 0.8% (Dixit et al. 2015) for *E. coli*. The ratio of the contributions to divergence by recombination and by mutation events (r/m) under these three

model settings are estimated to be smaller than 1.6, 40, and 80 (these values are like upper bound; see Materials and Methods). The r/m values of the forwardly-simulated population largely overlap with the r/m values found in nature, which range from 0.02 to 63.6 (Vos and Didelot 2009). For each of the three parameter sets, we collected around one hundred groups of closely-related genomes from the simulated populations. We imposed constraints when collecting genomes into test groups, so that they are (i) closely-related and (ii) their most recent common ancestor (MRCA) is much younger than the MRCA of the population that they are sampled. Hence the genomes in a test group come from a small local clade in the population; they exchange DNA stretches with themselves and closely related genomes in their local clade, and also with genomes in the large "outgroup" of the population.

Apart from forward-simulation, we also used the coalescent-based simulation algorithm SimBac (Brown et al. 2016) to generate 100 test-groups of genomes. Each group has 10 genomes, and each genome has a nucleotide sequence with length 100kb (Materials and Methods).

We reconstructed the phylogeny of the genome test-groups generated in the forward-simulation and also in the SimBac simulation. To reconstruct the phylogeny of genomes in each test-group, we applied CGP with two different segment sizes (l_s=20, l_s^{cutoff}=20 and l_s=100, l_s^{cutoff}=100), denoted as CGP20 and CGP100, along with three other state-of-the-art algorithms: RAxML, BEAST and ClonalFrame. For genomes generated from the forward-simulation, we shifted the frame of the segments so that some of them reside on the boundary of two recombination stretches. We compared these reconstructed trees with the actual authentic trees to evaluate the accuracy of different algorithms.

All tested algorithms generate a series of posterior trees; we compared the tree-series of each algorithm with the true phylogenetic tree. To evaluate the accuracy of tree topology of an algorithm, 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. s_{topo} is also denoted as 'efficiency' (Didelot and Falush 2007), and the higher is s_{topo}, the more accurate is the topology prediction. To evaluate the accuracy of branch length estimates, we measured the average deviation between the normalized age of internal nodes in the real tree and the normalized age of the corresponding nodes in the posterior trees, and called it as the node age deviation d_{age}. d_{age} is essentially a standard error, and so the lower is d_{age}, the more accurate is the node age of the reconstructed tree. Moreover, to help compare s_{topo} and d_{age} of different algorithms, we converted the s_{topo} (and also d_{age}) of different algorithms in each test-groups into z-scores (Materials and Methods).

Figure 1 shows the boxplot of z-score(s_{topo}), and Supplementary Figure S1 shows the boxplot of s_{topo} of the phylogenetic trees reconstructed by the 5 algorithms. For genomes

from forward-simulation with intermediate level of recombination, the trees reconstructed by CGP (l_s=100) have topology prediction as accurate as those by ClonalFrame; at high level of recombination, CGP (l_s=20 and l_s=100) are at least as accurate as ClonalFrame. For genomes generated by SimBac, CGP at l_s=20 is accurate as RAxML and BEAST, and more accurate than ClonalFrame; CGP at l_s=100 is more accurate than RAxML and ClonalFrame, and as accurate as BEAST. All these findings are supported by Wilcoxon signed rank tests on the s_{topo} distributions at significance level 0.05 (data in Supplementary File S1).

Figure 2 shows the boxplot of z-score(d_{age}) and Supplementary Figure S2 shows the boxplot of d_{age}. For genomes from forward simulation at all levels of recombination, the trees reconstructed by CGP (both l_s=20 and l_s=100) have node age more accurately predicted than those by BEAST and ClonalFrame. For genomes from SimBac, trees reconstructed by CGP at l_s=20 have node age more accurately predicted than those by ClonalFrame; CGP at l_s=100 is more accurate than ClonalFrame and as accurate as BEAST. These findings are supported by Wilcoxon signed rank tests on d_{age} at significance level 0.05 (data in Supplementary File S1).

We observed that CGP is more accurate when l_s is larger. For genomes obtained in forward simulation, CGP can always more accurately predict the node age than other tested algorithms; and for genomes generated by SimBac, CGP is less accurate in node age prediction but more accurate in topology prediction. A possible reason to this deviation is that, the dynamics of recombination in SimBac simulation is different from that in the forward-simulation and the real genomes, which results in very different SSP distributions. Supplementary Figure S3 shows the SSP distributions of genome pairs from forward-simulation (blue), SimBac (red), and also from a pair of real *E. coli* genomes (green); all these genome-pairs have sequence divergence ~2.4%. There, the SSP distribution of pairs from forward-simulation (blue) and real genome pairs (green) show a characteristic exponential tail, but this tail is not clear in the SSP distribution of SimBac (red). The origin of this exponential tail is the exponential dependence of recombination success rate on sequence divergence between the incoming DNA stretch and the host stretch. This dynamics of recombination is not available in SimBac, which is likely the cause of this deviation.

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. Since μ is fixed, the meaningful parameters are ρ/μ, θ, and δ_{TE}. We calculated the mean of ρ/μ, θ, and δ_{TE} of each genome test-group by taking the average of the posterior parameters of the last 200,000 MCMC steps (data in Supplementary File S1). Supplementary Figure S4 shows the boxplots of ρ/μ, θ, and δ_{TE} at different levels of recombination. We can see that a larger l_s leads to a more accurate prediction of ρ/μ and

δ_{TE}. But this trend in not observed in θ. A probable explanation is that, CGP is measuring θ based on genomes in a very small clade of the population; this leads to a lack of recombination with very divergent DNA stretches, which makes the measured θ to be smaller than what used in the simulation.

CGP predicts the ultrametric phylogenetic tree of real genomes more accurately 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 a 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 on BEAST and ClonalFrame, 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. This treatment will not save the computational time of CGP, because CGP works entirely on SSP distribution and independent of sequence length; the data structure and memory for storing the SSP distribution of a pair of genomes of 1kb length is the same as that for a pair of 1Mb. Each of the 10 strains in a test-group is represented by a nucleotide sequence and an amino acid sequence, with the sequences of all 10 strains forming an alignment. We used segment size (l_s=30, l_s^{cutoff}=30) and (l_s=300, l_s^{cutoff}=100) for nucleotide sequences, and called them CGP30n and CGP300n; we used segment size (l_s=10, l_s^{cutoff}=10) and (l_s=100, l_s^{cutoff}=100) for amino acid sequences, and called them CGP10a and CGP100a. We applied seven different phylogenetic reconstruction methods on each genome group, including CGP30n, CGP300n, BEASTn, ClonalFrame, CGP10a, CGP100a and BEASTa. Here, BEASTn (BEASTa) refers to BEAST applied on nucleotide

(amino-acid) sequences (see Materials and Methods, and also Supplementary File S2 for table of the strains and genes used in different test groups).

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. 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 is the representative tree, 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} values of individual test-groups. To help visualize the discrepancy of accuracy of the algorithms, we converted the GPL and N_{HGT} of different algorithms applied on the same test-group into z-score. Figure 3 shows the boxplot of the z-score(GPL), and Figure 4 shows the boxplot of the z-score(N_{HGT}). Both Figure 3 and 4 show that CGP10a is more accurate than other algorithms, for genome sequences with low or high divergence. If we only consider algorithms applied to nucleotide sequences, then CGP30n is as accurate as ClonalFrame on all tested conditions except on low divergence strains (test 1) using GPL as indicator. These findings are supported by Wilcoxon signed rank test at significance level 0.05.

Figure 3 and 4 also show that CGP algorithm with a smaller l_s gives better prediction than a larger l_s, which contradicts the finding in the simulated genomes and also Figure 1 and 2. A larger l_s increases the risk for a segment to cover multiple recombination stretches. Figures 2 of *Mostowy et al.* shows the length distribution of recombination stretches in

several prokaryotic genomes, where a significant portion of the recombination stretches have a length below 300bp (Mostowy et al. 2014). Hence a 300bp nucleotide segment may cover three or more recombination stretches, which confuses the algorithm and reduces the accuracy of the phylogenetic reconstruction.

The CGP algorithm also records the posterior parameters ρ/μ, θ, and δ_{TE} (data in Supplementary File S2). For each genome test group, 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. Supplementary Figure S5 shows the boxplot of these posterior parameters measured at different conditions. While the distributions of ρ/μ measured by CGP at different l_s on both test 1 and 2 are roughly consistent with each other, this is not the case for θ, and δ_{TE}. The value of δ_{TE} is known to be around 1% ~ 3% (Fraser et al. 2007), but only CGP300n applied on high divergence genomes gives a similar range. This shows that the algorithm requires more fine-tuning in order to get better measurement of θ, and δ_{TE}.

We also made a rough comparison on the computational cost of different algorithms. For each 10-genomes test group of both test 1 and 2, we performed phylogenetic reconstruction using the seven algorithms, and measured the CPU time of each run (see Supplementary Figure S6 for the box plot of computational time and their z-score; raw data in Supplementary File S2). We found that CGP10a is the fastest algorithm (supported by Wilcoxon signed rank tests at significance level of 0.05), followed by BEASTn and CGP30n that are almost as fast. While we are using 100-genes-genomes to test the algorithms, in reality we may use as many as 4000 genes to present a genome. This will substantially increase the computational cost of BEAST and ClonalFrame; and for CGP, a longer sequence means less noise in the empirical SSP distributions and better phylogeny prediction, but not computational cost. Further, one should also note that, different algorithms make use of different criteria for MCMC termination. A typical CGP run lasts for around 300,000 steps, while a typical BEAST run lasts for 10,000,000 steps, and the algorithm run-times depend on these criteria.

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 is more accurate 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 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 with small segment size l_s 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. Supplementary Figure S4 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 and δ_{TE} at large l_s is more accurate, while at small l_s the prediction is worse. Supplementary Figure S5 shows the boxplot of the the posterior values of ρ/μ, θ, and δ_{TE} estimated for *E. coli* genomes; while the estimation of ρ/μ using a different l_s under different conditions are fairly consistent with each other, the consistency is worse for θ and δ_{TE}. This is probably because, different combinations of θ and δ_{TE} may lead to similar shape of SSP distribution; a remedy to this is to limit the search space of θ and δ_{TE}, i.e., as we have measured δ_{TE} to be around 1% to 3%, we can limit δ_{TE} to be <5% in CGP, which may reduce ambiguity in the parameter space and improve the parameter prediction of CGP. We leave all these fine tuning of the algorithm to future work.

The CGP algorithm is the fastest of the algorithms tested (Supplementary Figure S6). Its speed is independent of the sequence length of individual genomes except for calculation of the pairwise distance distributions. CGP infers the phylogeny from the pairwise SSP distributions, represented by the symbol $f(x|t)$, of the genomes considered. Mathematically, $f(x|t)$ is the probability for a segment of a genome-pair that has coalescent time t to have divergence x. Fixing the time t, $f(x|t)$ 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. Instead, the speed of CGP is sensitive to n and l_s^{cutoff}. In a simulation step, if any of the the model parameters is updated, it will need to recalculate $f(x|t)$, and then sum up the cross entropies of the $n(n\text{-}1/)2$ strain pairs to obtain the new posterior probability. Hence at large n, the computational time scales as $O(n^2)$. The segment size l_s and the cutoff number for SSPs on a segment (l_s^{cutoff}) also affect the computational cost and the accuracy of the reconstruction. As the calculation of $f(x|t)$ involves multiplication of $(1+l_s^{cutoff})\times(1+l_s^{cutoff})$ matrices, at large l_s^{cutoff} the computational cost scales like $O((l_s^{cutoff})^2)$. A large l_s improves the accuracy of tree and parameters prediction, as it is shown in the analysis on simulated genomes (Figure 1 and 2, and Supplementary Figure S4). However, this also increases the computational time, and also the risks for a segment to

cover multiple recombination stretches that consequently reduces the accuracy of the reconstruction, which is the case in the analysis on real genomes (Figure 3 and 4). Thus, we need to carefully adjust the segment size in order to optimize CGP.

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 in a population 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 framework 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. CGP considers segments instead of individual nucleotide / amino-acid sites as the basic unit of a genome, because local SSP density can be defined on segments. A genome has L_{seg} segments, 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 site of a segment. The rate for a segment to be covered by a recombination stretch is ρ per segment per generation. In fact, a recombination stretch can cover multiple segments, and thus $\rho=\rho_{ini}L$, where ρ_{ini} is the rate for a recombination stretch to start at a segment, and L is the average length of a recombination stretch (in unit of segment); for simplicity, in our model neighbouring segments are considered to have independent recombination events. Recombination has a success rate of $\exp(-x/\delta_{TE})$, where x is the divergence between the incoming foreign segment and the host segment, and the constant δ_{TE} is also called transfer efficiency (Dixit et al. 2015). The unit of segment divergence, as well as δ_{TE}, can be expressed as number of 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 density distribution between a pair of genomes X and Y. Let us divide the alignment of genome X and Y into L_{seg} consecutive segments with l_s sites, and let $f(x|t)$ be the distribution of divergence on the segments, where $x \geq 0$ is discrete and represents segment divergence, $t \geq 0$ is continuous and represents the coalescent time between X and Y. Let us use the number of SSPs on a segment, instead of percentage, to represent divergence; this makes x an integer with range 0, 1, …, l_s, and we further assumed that a segment can have no more than l_s^{cutoff} SSP to save computational resource. Thus this makes $x \in (0, 1, \ldots, l_s^{cutoff})$, and $f(x|t)$ is normalized to unity when summed over x:

$$\sum_{x=0}^{l_s^{cutoff}} f(x|t) = 1$$

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 $f(x|0)=\delta_{x,0}$ (Kronecker delta), where $f(x|0)=1$ when $x=0$, and $f(x|0)=0$ when $x \neq 0$. After the MRCA splits into two lineages, mutation and recombination events occur and affect their SSP distribution. The evolution of $f(x|t)$ is described by the following equation:

$$\frac{df(x|t)}{dt} = 2\mu\left(\sum_{y=0}^{l_s^{cutoff}} M(x|y)f(y|t) - f(x|t)\right) + 2\rho\left(\sum_{y=0}^{l_s^{cutoff}} P(x|y,\theta,\delta_{TE})f(y|t) - f(x|t)\right) \quad (1)$$

In this equation, the first term accounts for mutation, and the second term accounts for recombination. μ is the segment-wise mutation rate; $M(x|y)=\delta_{x,y+1}+(\delta_{x,lscutoff}\times\delta_{y,lscutoff})$ is the mutation matrix, which accounts for the fact that a segment with divergence x jumps to x+1 when a mutation occurs. ρ is the rate for a segment to be covered by a recombination stretch. There is a factor 2 in both terms because mutation or recombination occurring on either X or Y will affect $f(x|t)$. In reality, a recombination stretch 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 is invariant. $P(x|y,\theta,\delta_{TE})$ in Eq. (1) is the recombination matrix, which is the probability for a segment to change its state from y to x during a recombination. θ is the average segment divergence in the population, and δ_{TE} is the transfer efficiency that governs the success rate of recombination, as the

model allows a recombination event to fail; both θ and δ_{TE} have unit of divergence. $P(x|y,\theta,\delta_{TE})$ is like

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

Here $\Theta(x)$ is step function, which is 1 when $x>0$ and 0 when $x\leq 0$; δ_{xy} is Kronecker delta. 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 recombination either event failed, or succeeded but did not change the divergence, and $A_3(y,\theta,\delta_{TE}) = 1 - \sum_{x=0}^{y-1} A_1(x|y,\theta,\delta_{TE}) - \sum_{x=y+1}^{l_s^{cutoff}} A_2(x|y,\theta,\delta_{TE})$;

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

$$\sum_{x=0}^{l_s^{cutoff}} P(x|y,\theta,\delta_{TE}) = 1$$

Starting from Equation (1) with the boundary condition $f(x|0)=\delta_{x0}$, we can calculate $f(x|t)$ at different t. Supplementary Figure S7 shows an example of 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 matrix 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 recombination matrix defined here also includes the case where a transferred DNA fails to recombine with the host segment.

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):

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

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 $\min(\theta,\delta_{TE})\rho$, which gives $(\rho/\mu)\min(\theta,\delta_{TE})$ after divided by μ. A caveat here is that, what this expression estimates is an upper bound to the true r/m. This is because the rate of successful recombination is lower than ρ when $\theta \gg \delta_{TE}$.

Computational forward-simulation of the neutral coalescent model

We performed forward-simulation of genome populations following the framework of a neutral coalescent model that has recombination; this simulation framework has been applied to understand the effect of recombination on sequence divergence (Fraser et al. 2007). In each simulation, we set the population size N_e=1000, each genome in the population has 100 stretches that are basic units of recombination, and each stretch has 1000 binary sites. Recombination is a random Poisson process that happens at a rate ρ per stretch per time-step, which transfers one stretch when it occurs. 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 exist if the simulation does not last long enough, and we let the simulation continue; if the MRCA has emerged, then we stopped 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, with unit of μ and ρ being per stretch per time step, 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 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 accuracy 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 test-group of genomes that come from a small local clade of the population, so that a genome in the test-group recombines with other within-clade-genomes, as well as with that of the extra-clade-genomes. Therefore, the random genomes are picked with the constraint that, the age of their MRCA should be small, i.e., $t_{root} \ll 1000$.

We forwardly-simulated ten populations for each of the three levels of recombination. In each simulated population, we picked ten test-groups of genomes, each test-group has it own constraints on the age of their MRCA: t_{root} ~ 10, 20, …, 100. Each test-group has around 10 genomes. In this way, this generates 100 groups of genomes for each of the three ρ/μ values.

Coalescent-based simulation of the neutral model

Coalescent-based simulation is an alternative framework to forward-simulation of the neutral model, which is fast and can generate genomes that mimic those evolved in a large population. We used SimBac (Brown et al. 2016), a recently published coalescent-based simulation algorithm that can implement recombination, to generate 100 groups of genomes. Each group contains 10 genomes, and each genomes is represented by 100kb nucleotide sequence. SimBac emulates an 'internal' population of genomes recombining with themselves and with an 'external' population. We set the external recombination rate to be 0.001, a tenth of internal recombination rate, and also set the upper bound of divergence in the regions recombined with external population to be 3%; we used default settings for the remaining parameters, including 500bp transfer stretch length. After feeding in the parameters, SimBac outputs a fasta file that contains the genomes and also their true phylogenetic tree.

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 $g(x)$ is normalized to L_{seg} when summed over x. Further, let us denote the theoretical

distribution as $f_{\mu,\rho,\theta,\delta TE}(x|t)$, which is normalized to unity when summed 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

$$\prod_x [f_{\mu,\rho,\theta,\delta_{TE}}(x|t)]^{g_{XY}(x)} \quad (3)$$

which, if we take the logarithm on this term, becomes the (negative) cross entropy between $g(x)$ and $f_{\mu,\rho,\theta,\delta TE}(x|t)$ (Rubinstein and Kroese 2004; 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 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 $n(n-1)/2$ individual terms:

$$S(X_1,\ldots X_n|\mu,\rho,\theta,\delta_{TE},T) = \sum_{all\ (X_a,X_b)\ pairs} log\{\prod_x [f_{\mu,\rho,\theta,\delta_{TE}}(x|t_T(X_a,X_b))]^{g_{X_aX_b}(x)}\} \quad (4)$$

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 ultrametric phylogenetic trees; for instance, let the number of genomes be n. Given the initial parameters (μ, ρ, θ, δ_{TE}), CGP first rescales μ and ρ by the same ratio, with the rescaled mutation rate μ=min(0.02, $<x_{farthest}>$/200) (unit: number of mutation per segment per time step), and ρ is rescaled to be a tenth of μ; here $<x_{farthest}>$ is the average segment divergence between the most divergent genome pair. There are two considerations behind this rule: (i) μ (and also ρ) should be $\ll$1 to reduce numerical error, but the smaller is μ, the larger is the magnitude of the branch lengths, which demands a higher computational cost to solve Equation (1) to calculate the cross-entropies; (ii) thus the branch lengths should also be numerically small to reduce the computational time. The current rescaling rule makes a balance between these criteria.

Next, CGP calculated the theoretical SSP distribution $f_0(x|t)$ given the initial parameters, and constructed the initial distance matrix of genome pairs based on $f_0(x|t)$: for a genome pair X and Y with empirical SSP distribution $g_{XY}(x)$, their initial distance t_{XY} is chosen such that it maximizes the similarity between $f_0(x|t)$ and $g_{XY}(x)$ measured by their cross-

entropy. CGP then converted this initial distance matrix into the initial ultrametric tree by single linkage clustering.

CGP represents the space of an n-leaves ultrametric tree by an $n \times n$ distance matrix (tree matrix), which maps to its ultrametric tree through single linkage clustering. While this tree matrix has n^2 entries, the actual number of freedom is around n-1, since there are no more than n-1 internal nodes in the tree. Moving in the direction of a degree of freedom corresponds to moving an internal node of the tree up or down. For a node with two children, a local search move corresponds to moving it upwards or downwards by one time-step, which can lead to the joining of two nodes into one and change the tree topology. For an internal node that has number of children c>2, apart from moving the entire node upwards or downwards, a possible move involves breaking it into two nodes, one with two children and the other with c-1 children (see Supplementary Figure S8 for an example).

After the initial tree T and the initial parameters (μ, ρ, θ, δ_{TE}) are determined, the MCMC proceeds to search for the optimal parameters and trees. 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 one of 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: ρ<1,θ<100% and δ_{TE}<100%. The new parameter set is selected if it leads to a higher posterior probability; otherwise, it is selected according to its relative probability to the original parameter set.
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 another random branch on the tree that is present at height t_O to generate a new ultrametric tree. This new tree is selected if it leads to a higher posterior probability; otherwise, it is selected according to its relative probability to the original tree.
3. Else, an internal node is picked randomly. One of the possible moves of the node described above is selected randomly to generate a new tree. If this new tree leads to higher probability, then it is selected; otherwise, it is selected according to its probability relative to the original tree.

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 Table S2 for the link to the 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 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 use all 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 CGP with l_s=20, l_s^{cutoff}=20 and l_s=100, l_s^{cutoff}=100 (denoted as CGP20 and CGP100), RAxML (Stamatakis 2014), BEAST (Drummond et al. 2012) and ClonalFrame (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 CGP20 and CGP100 to infer their ultrametric tree; the CGP algorithms last for at least 200,000 steps, and it records the posterior parameters and posterior tree every 1,000 steps. It terminates when the logarithm of the posterior probability does not increase by more than 1 for 200,000 MCMC steps, and the posterior trees / parameters of the last 200,000 steps are collected for analysis.

For RAxML, we applied the substitution model BINGAMMA (GTRGAMMA) for genomes from forward-simulation (SimBac coalescent-based simulation), which is suitable for binary sequence (nucleotide sequence); 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, the genomes from forward-simulation are binary and we converted the 0 and 1 into A and T; the genomes from SimBac coalescent simulation does not require special treatment. We 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, the genomes from forward-simulation are binary and we converted the 0 and 1 into A and T; the genomes from SimBac coalescent simulation does not require special treatment. We fed the genomes into the ClonalFrame program with default setting, and then 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 the forward-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), and prepared around 100 genome groups for each parameter set; we also prepared 100 genome groups using coalescent-based simulation algorithm SimBac. We applied five algorithms to reconstruct their phylogenies (CGP20, CGP100, RAxML, BEAST, ClonalFrame). Each of the five 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 ultrametric tree, excluding the root node, to find its corresponding internal node on a posterior tree that clusters the leaf-nodes in the same way; this posterior tree does not have to be ultrametric. The topology similarity score is bounded, $0 \leq s_{topo} \leq 1$, 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 clusters the leaves in the same way as 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 satisfies $d_{age} \geq 0$, and the smaller the deviation the more accurate is the node age prediction.

Furthermore, we converted s_{topo} and d_{age} into z-score, because it helps visualization. Each genome test group is reconstructed by multiple algorithms, and each algorithm get its s_{topo} and d_{age}, these s_{topo} (and d_{age}) are converted into z-scores together on a group-by-group basis.

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 by 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:

1. CGP on nucleotide sequences (l_s=30, l_s^{cutoff}=30, and also l_s=300, l_s^{cutoff}=100, denoted as CGP30n and CGP300n);
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 (BEASTn);
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}=100, denoted as CGP10a and CGP100a);
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 (BEASTa).

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—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.

Further, 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 2017). 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.

Furthermore, we converted GPL and N_{HGT} into z-score to help data visualization. Each genome test group is reconstructed by multiple algorithms, and each algorithm get its GPL and N_{HGT}, these GPL (and N_{HGT}) are converted into z-scores together in a group-by-group basis.

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 the seven algorithms: CGP30n, CGP300n, BEASTn, ClonalFrame on the nucleotide sequences, and CGP10a, CGP100a, BEASTa 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). We converted the computational time of each algorithm on every genome test group into z-score on a group-by-group basis.

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.

Figure legends

Figure 1. Boxplot for the distribution of the z-score of s_{topo} (the higher the s_{topo}, the more accurate is the tree topology) of the phylogenetic trees reconstructed by five different algorithms. The first three panels are results from forwardly-simulated populations at various levels of recombination, and the last panel is from coalescent-based simulation (SimBac).

Figure 2. Boxplot for the distribution of the z-score of d_{age} (the lower the d_{age}, the more accurate is the node age prediction) of the phylogenetic trees reconstructed by five different algorithms; note that the *y*-axes of the plots are upside down. The first three panels are results from forwardly-simulated populations at various levels of recombination, and the last panel is from coalescent-based simulation (SimBac).

Figure 3. Boxplot showing the distribution of z-score of GPL of trees reconstructed by seven different algorithms. The higher the GPL (and its z-score), the more accurate is the tree.

Figure 4. Boxplot showing the distribution of z-score of N_{HGT} of trees reconstructed by seven different algorithms. The lower the N_{HGT} (and its z-score), the more accurate is the tree. Note that the *y*-axes of the plots are upside down.

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.

Brown T, Didelot X, Wilson DJ, Maio ND. 2016. SimBac: simulation of whole bacterial genomes with homologous recombination. Microb. Genomics [Internet] 2. Available from: http://mgen.microbiologyresearch.org/content/journal/mgen/10.1099/mgen.0.000044

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.

Croucher NJ, Page AJ, Connor TR, Delaney AJ, Keane JA, Bentley SD, Parkhill J, Harris SR. 2015. Rapid phylogenetic analysis of large samples of recombinant bacterial whole genome sequences using Gubbins. Nucleic Acids Res. 43:e15–e15.

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.

Didelot X, Wilson DJ. 2015. ClonalFrameML: Efficient Inference of Recombination in Whole Bacterial Genomes. PLOS Comput Biol 11:e1004041.

Dixit PD, Pang TY, Maslov S. 2016. Recombination-driven genome evolution and stability of bacterial species. bioRxiv:67942.

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.

Marttinen P, Hanage WP, Croucher NJ, Connor TR, Harris SR, Bentley SD, Corander J. 2012. Detection of recombination events in bacterial genomes from large population samples. Nucleic Acids Res. 40:e6.

Mostowy R, Croucher NJ, Hanage WP, Harris SR, Bentley S, Fraser C. 2014. Heterogeneity in the Frequency and Characteristics of Homologous Recombination in Pneumococcal Evolution. PLOS Genet. 10:e1004300.

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 MJ. 2017. Supra-operonic clusters of functionally related genes (SOCs) are a source of horizontal gene co-transfers. Sci. Rep. 7:40294.

Rubinstein RY, Kroese DP. 2004. The Cross-Entropy Method: A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation and Machine Learning. Springer Science & Business Media

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.

Vaughan TG, Welch D, Drummond AJ, Biggs PJ, George T, French NP. 2016. Bayesian inference of ancestral recombination graphs for bacterial populations. bioRxiv:59105.

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 times of the phylogenetic reconstructions.

Figures

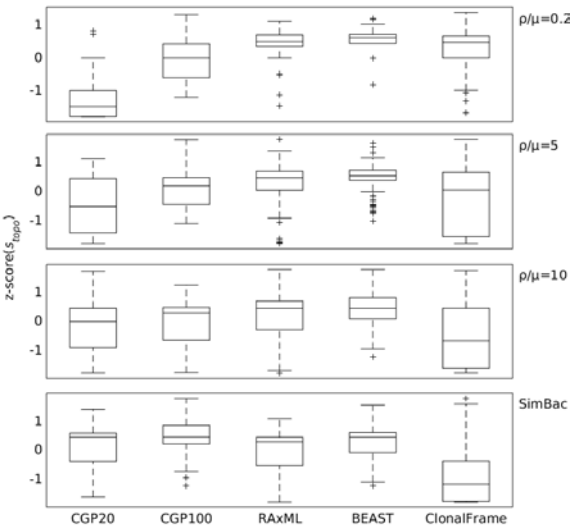

Figure 1.

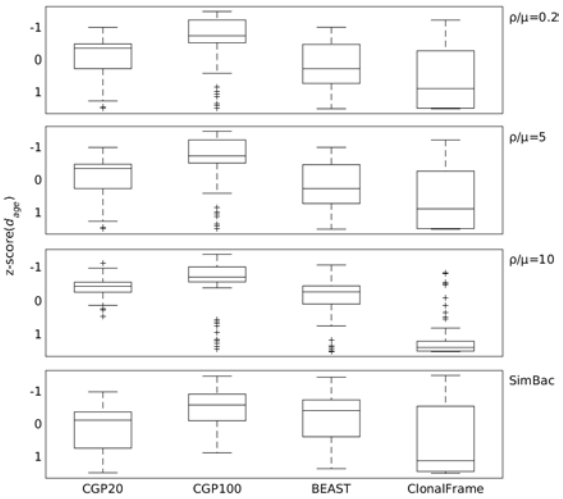

Figure 2.

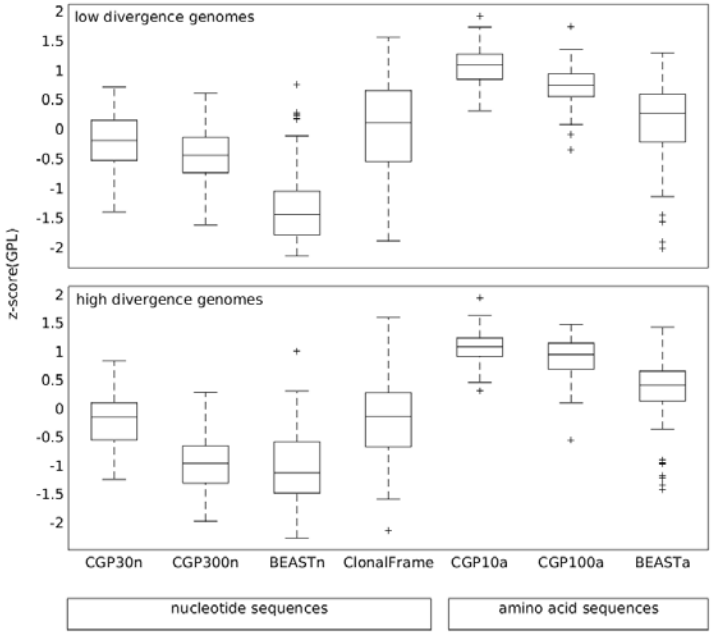

Figure 3.

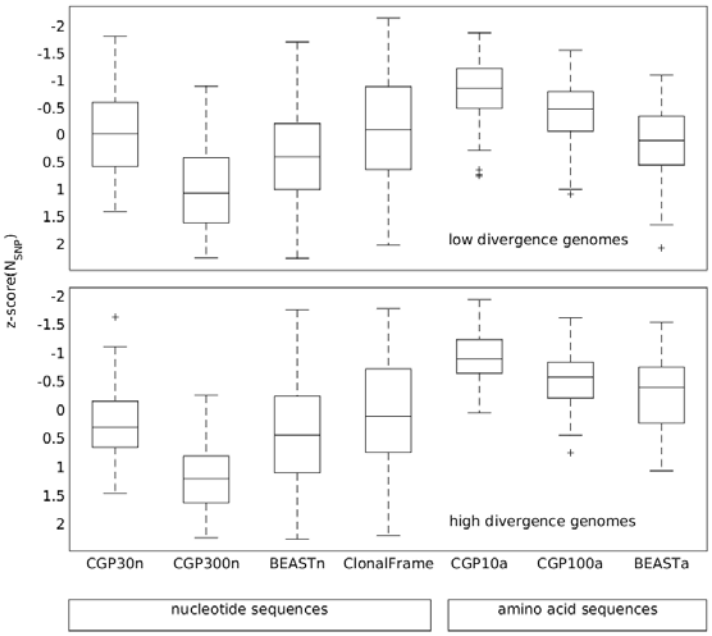

Figure 4.