Ismail and Tang *BMC Genomics* 2019, **20**(Suppl 12):1002
https://doi.org/10.1186/s12864-019-6328-3

BMC Genomics

RESEARCH Open Access

Clonal reconstruction from time course genomic sequencing data

Wazim Mohammed Ismail* and Haixu Tang

From The International Conference on Intelligent Biology and Medicine (ICIBM) 2019
Columbus, OH, USA. 9-11 June 2019

Abstract

Background: Bacterial cells during many replication cycles accumulate spontaneous mutations, which result in the birth of novel clones. As a result of this *clonal expansion*, an evolving bacterial population has different clonal composition over time, as revealed in the long-term evolution experiments (LTEEs). Accurately inferring the *haplotypes* of novel clones as well as the clonal frequencies and the clonal evolutionary history in a bacterial population is useful for the characterization of the evolutionary pressure on multiple correlated mutations instead of that on individual mutations.

Results: In this paper, we study the computational problem of reconstructing the haplotypes of bacterial clones from the *variant allele frequencies* observed from an evolving bacterial population at multiple time points. We formalize the problem using a maximum likelihood function, which is defined under the assumption that mutations occur spontaneously, and thus the likelihood of a mutation occurring in a specific clone is proportional to the frequency of the clone in the population when the mutation occurs. We develop a series of heuristic algorithms to address the maximum likelihood inference, and show through simulation experiments that the algorithms are fast and achieve near optimal accuracy that is practically plausible under the maximum likelihood framework. We also validate our method using experimental data obtained from a recent study on long-term evolution of Escherichia coli.

Conclusion: We developed efficient algorithms to reconstruct the clonal evolution history from time course genomic sequencing data. Our algorithm can also incorporate clonal sequencing data to improve the reconstruction results when they are available. Based on the evaluation on both simulated and experimental sequencing data, our algorithms can achieve satisfactory results on the genome sequencing data from long-term evolution experiments.

Availability: The program (ClonalTREE) is available as open-source software on GitHub at https://github.com/COL-IU/ClonalTREE.

Keywords: Clonal reconstruction, Time course, Maximum likelihood, Long-term evolution experiment

Background

Long-term evolution experiment (LTEE) has long been adopted to study how genetic variations are generated and maintained in a period of time and how novel variations are associated with the adaptation of the species to novel environmental conditions [1]. Due to their high genetic diversity and rapid evolution, unicellular microbes, predominantly *E. coli*, are used in LTEEs [2–4], although LTEE was also conducted on multi-cellular model animals such as Drosophila [5]. The *E. coli* long-term evolution experiment conducted by Lenski and colleagues is the longest on-going LTEE, in which twelve initially identical *E. coli* strains (i.e., the *founder* clones) were grown in parallel, each under a daily serial passage for 30 years [3, 6, 7]. A variety of phenotypic changes were observed in the bacterial population during the experiment, including increased fitness to specific growth conditions [8] and elevated mutation rates [9].

*Correspondence: wazimoha@iu.edu
School of Informatics, Computing and Engineering, Indiana University, Bloomington, IN, USA

In recent years, LTEE were combined with *metagenome sequencing* (i.e., sequencing the whole genomes in the population, also referred to as the Pool-Seq, or the sequencing of pooled individual genomes) to characterize genetic variations introduced during the course of experiment, and the allele frequencies of these variations in a population [3, 10]. Some of these novel variations were revealed to be associated with observed phenotypic changes, e.g., the defective mutations in the DNA repair pathways causing elevated mutation rates [9], and the novel genetic traits selected for citrate use [10]. Furthermore, population-wide metagenome sequencing can be conducted on the evolving population at multiple time points to monitor the dynamic changes of genetic variations in complex and heterogeneous growth environments. The main objective of these studies is to identify clones adapted to specific environmental niche over the time course. However, due to the nature of metagenome sequencing, it is not straightforward to determine the *haplotypes* of the clones arising in the experiment. Instead, selections are often detected on the novel variations by applying statistical tests [11, 12] to the time series allele frequencies derived from the sequencing data. Because a novel variation, e.g., a single nucleotide variation (SNV), may be shared by multiple clones in the population (i.e. subsequent mutations may occur in a clone already containing mutations instead of the founder clone), the tests on variation may be less sensitive than the tests directly on the frequency profiles of haplotypes, and thus may miss the selection on some clones, especially when the population is dominated by a few clones containing many variations.

To address this issue, in a recent study, metagenome sequencing coupled with clonal sequencing was adopted for the study of populations of wild-type (WT) and repair-deficient *E. coli* evolving over three years [4]. To characterize the haplotypes in the populations, whole genome sequencing was carried out on randomly selected clones at the end of the experiments. In addition, the haplotype frequencies of the major clones were derived from the metagneome sequencing data. The dynamic changes of these major clones during the course of experiment showed a clear picture of the subpopulation structure (e.g., using a Muller plot; see Fig 1c), in which the major clones evolved with different genotypes associated with nutrition metabolism. Despite the demonstrated success here, the clonal sequencing has two disadvantages in practice. First, because the sequenced clones are randomly selected, minor clones with low abundances in the population may not be characterized (while major clones are sequenced repetitively), and thus their frequency profiles during the time course will not be considered in the subsequent analyses. More importantly, the clones are usually chosen at the end of the experiment; as a result, the clones with high abundance in the middle but becoming less abundant towards the end of the experiment are less likely to be characterized, which will not only miss some clones under selection during the time course, but also miscalculate the allele frequencies of characterized clones in the middle of time course. Therefore, unless the clonal sequencing covers a large number of clones (that may contain many duplicated clones) compared to the complexity of the population, it is desirable to develop computational methods to reconstruct the haplotypes of clones from time series metagenome sequencing data.

Interestingly, the clonal reconstruction has been extensively studied in the field of cancer genomics for tracking the evolution of cancer cells by bulk tumor genome sequencing [13, 14], in an attempt to characterize the intra-tumor heterogeneity (i.e., clonal tree and composition) and in the mean time to identify the clones carrying *driver mutations* that occur in the early stage of cancer and drive the cancer progression [15]. Computationally, the *clonal reconstruction* (also referred to as the *clonality inference*) takes as input the allele frequencies of a set of genetic variants in multiple samples (e.g., dissected from the same tumor tissue), and aims to reconstruct a set of clones, each carrying a subset of the variants, and simultaneously infer the fraction of these clones in each sample [16]. Many algorithms addressed the clonal reconstruction problem [16–21] by inferring the evolutionary history of reconstructed clones and the generation of variants (assuming that each variant is generated only once, i.e., the *infinite sites assumption* [22]), from which the likelihood of a variant being the driver can be prioritized [23, 24]. It is worth noting that here, the clonal evolution was not inferred from time series sequencing data (which are difficult to obtain in cancer genomics), but the inherent constraints among variant frequencies due to the infinite sites assumption, (e.g., no clone can carry two variants unless the frequencies of one variant is always greater than the other; for details see [16]). Finally, similar to the clonal sequencing in LTEE, single cell sequencing data offers complementary information to clonal reconstruction in cancer genomics [25], and algorithms became available to infer tumor heterogeneity from low coverage single cell sequencing data [26, 27].

In this paper, we formalize the problem of clonal reconstruction from time course genomic sequencing data in a maximum likelihood framework, and devise a series of heuristic algorithms to address it. We further extend the algorithms to incorporate clonal sequencing data, aiming at reconstructing additional clones that are not sequenced. We simulated the bacterial population in long-term evolution experiments, and use the simulated genomic data to test our algorithms. The results show that the heuristic algorithms could accurately reconstruct as many clones as reconstructed by the brute-force

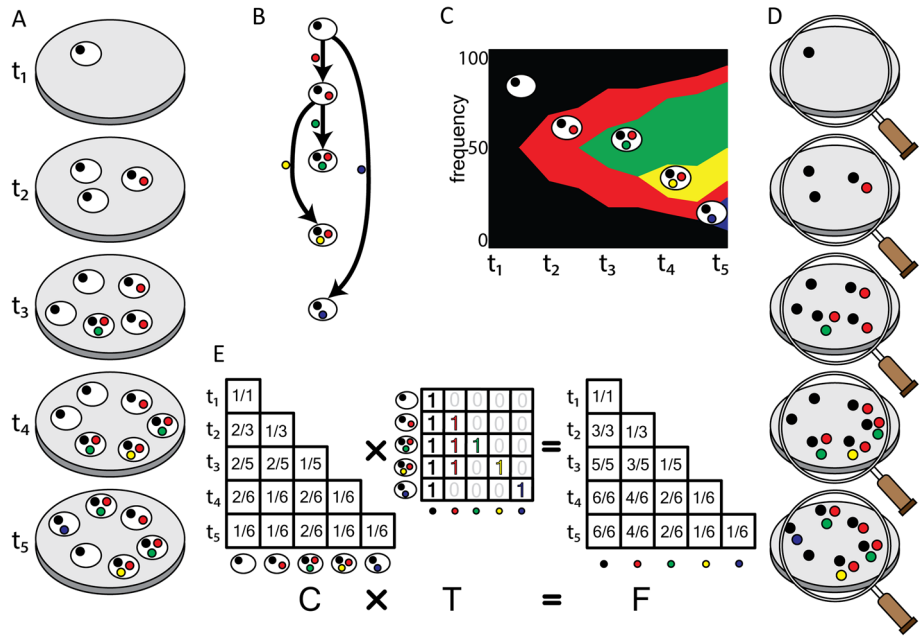

Fig. 1 A schematic illustration of the clonal structure in an evolving bacterial population and the time course clonal reconstruction problem. **a** Starting from a single founder clone (shown in black) at time t_1, four mutations (shown in red, green, yellow and blue, respectively) occur at time point t_2 to t_5, respectively, resulting in four novel clones (denoted by their unique variants). **b** The clonal tree represents the evolutionary history of these clones, in which each node represents a clone including the founder clone as the root, and each edge represents the mutations that occur at specific time points. **c** The Muller plot shows the evolutionary dynamics with the novel clones along with their frequencies at each time point. **d** Metagenome sequencing conducted at different time point, from which the variant allele frequencies (VAF) matrix can be derived. **e** The VAF matrix can be viewed as the product of the clonal tree (*T*) and the clone frequencies (*C*), similar to the formulation in cancer genomics [16]. The goal of this work is to reconstruct the clonal tree (*T*) and the clone frequencies (*C*) from the observed VAF matrix

algorithm or even better on average, while improving significantly on speed. We also discuss the effect of varying the number of clones in the population and the number of time points. Finally, we test our algorithms on a real LTEE dataset [4] from an *E. coli* population. Our algorithms successfully reconstruct clonal haplotypes that are not characterized by clonal sequencing, and reveal the evolutionary dynamics of the clones during the LTEE.

Methods

Modeling clonal evolution of bacteria

We model an evolving bacterial population using the clonal theory [28, 29], similar to the one used in cancer genomics [30]. We assume that all bacterial cells in an evolving population are descendants of a single *founding clone*. During the course of the evolution experiment, bacterial cells accumulate novel *mutations* forming new *clones*. In this study, we focus only on single nucleotide variations (SNVs); but the other types of variations (e.g., indels, structural variations and copy number variations) can be modelled in the same way. We further assume that the occurrences of mutations follow the *infinite sites assumption*, i.e., a mutation occurs at a single locus at most once during the period of evolution experiment.

The ancestral relationships between the clones in the evolving population can be represented as a *directed tree T*, referred to as the *clonal tree* in which the root represents the founder clone, every other node represents a clone introduced by one or more novel mutations, and each edge represents the direct ancestral relationships between the clones (Fig 1b). Each edge is labeled by the mutation(s) that distinguishes the *child* from its *parent*. When more than one mutation occurs during the evolution from the parent to the child, they can be clustered together and considered as a single mutation group. As a result, the *haplotype* of a clone (i.e., the variants contained in the clone) is represented by the path from the root to the node representing the clone.

The frequency of each clone at each specific time point is represented as a matrix $C = [c_{ij}]$, referred to as the *clonal frequency matrix (CFM)*, in which $c_{i,j}$ indicates the frequency of clone j at the time point i. Our model assumes that the mutation occurs spontaneously; as a result, at any given time, the likelihood of a candidate clone to acquire a new mutation hence spawn a new clone

Algorithm 1 Exhaustive tree search algorithm (ET)

procedure EXHAUSTIVE-TREE-SEARCH(F) ▷ F is a square lower diagonal matrix of VAFs
  Let G be a graph such that an edge links vertices j and k, if $F_{i,j} \geq F_{i,k}, \forall i$
  for each spanning tree T in G **do**
    likelihood ← LIKELIHOOD(T, F) ▷ Compute the likelihood based on Eq. 1
    keep *max_tree* with the maximum value of *likelihood*.
  end for
  return *max_tree*
end procedure

is proportional to the frequency of the clone in the population at the time. The clonal tree T and the CFM C together can be depicted in a *Muller plot* [31] (Fig 1c), which is commonly used to visualize the evolutionary dynamics in a population [32].

Time course clonal reconstruction problem

In order to monitor the evolutionary process in a bacterial population, metagenome sequencing can be conducted at a series of N time points, from which a *variant allele frequencies (VAF)* for all variation sites are obtained at each specific time point and represented as a *VAF matrix* [16], $F = [f_{ij}]$, where $f_{i,j}$ indicates the allele frequency of the variant j at the time point i. Notably, each variant is first introduced by a mutation (or multiple mutations) at the time point t_j, generating a novel clone (denoted by the specific mutation j) from its parent. Apparently, t_j is defined as the earliest time point t, such that $f_{t,j} > 0$, and for $\forall i < t, f_{ij} = 0$.

Given a VAF matrix F, our goal is to reconstruct the haplotype of each clone (i.e, the novel variants it contains) arising during the evolution experiment, or equivalently, to infer a clonal tree containing all observed mutations. Based on the clonal evolution model, we formally define the *time course clonal reconstruction problem* using a maximum likelihood formulation: given the input of matrix $F = [f_{i,j}]$ where $1 \leq i,j \leq N$ over N mutations (or novel clones) *sorted* over N time points (i.e., each mutation occurring at a known distinct time point), we want to find a directed tree $T^* = (pr(i), i)$, $i = 1, 2, ..., N$ on N nodes (where $pr(i)$ is the *only* parent node of node i) that maximizes the following likelihood function,

$$L(T) = \prod_{i=2}^{N} C_{(i-1),pr(i)} = \prod_{i=2}^{N} \left(F_{(i-1),pr(i)} - \sum_{\substack{j \in ch(pr(i)), \\ 1 \leq j < i}} F_{(i-1),j} \right) \quad (1)$$

where $C_{i,j}$ represents the (unknown) frequency of the clone j at the time point i, and $ch(i)$ represents the set of all children of the node i. The likelihood function is computed by multiplying the likelihood of generating each clone in the clonal tree. As described above, the likelihood of generating the clone i or the probability of introducing the mutation i in clone $pr(i)$ within the time segment between the points $i-1$ and i is approximated by the frequency of the clone $pr(i)$ at the time point $i-1$, which can be computed as the frequency of the variant $pr(i)$ subtracting the frequencies of all children of $pr(i)$ born before the time point i.

We search for the optimal solution of a clonal tree T in the search space containing a total of $(N-1)!$ trees because at any given time i there are $i-1$ putative parents to choose from. While some trees can be identified as invalid solutions when

$$\exists (1 \leq i \leq N) \quad \text{s.t.} \sum_{\substack{j \in \text{ch}(\text{pr}(i)), \\ 1 \leq j < i}} F_{i,j} > F_{i,\text{pr}(i)}, \quad (2)$$

a brute force approach to search the ML solution in the entire tree space, referred to as the *exhaustive tree search algorithm* (ET; Algorithm 1) is still computationally expensive. Once the clonal tree is constructed, the haplotype of each clone (corresponding to a node in the tree) can be derived from the path from the root to the node. Note that a variant of this problem called the *variant allele frequency factorization problem (VAFFP)*, where the order (time) of appearance of each mutation is unknown and the likelihood assumption is not applicable, is proven to be NP-complete [16].

Greedy tree search algorithm (GT)

To reduce the computational complexity of the *exhaustive tree search* (ET), we propose an algorithm using a greedy approach as follows (see Algorithm 2 for details). Start growing the directed tree from the root node (founder) such that at each iteration $i > 1$,

$$pr(i) \leftarrow \underset{1 \leq k < i}{\arg\max}\, C_{(i-1),k} = \underset{1 \leq k < i}{\arg\max} \left(F_{(i-1),k} - \sum_{\substack{j \in ch(k), \\ 1 \leq j < i}} F_{(i-1),j} \right) \quad (3)$$

provided $pr(i)$ does not lead to an invalid solution (according to Eq. 2). At any iteration i, if the assignment of $pr(i)$ leads to an invalid solution, choose the next optimal

choice at iteration $i-1$, and continue the search. At any iteration i, if no more assignment leads to a valid solution, choose the next optimal choice at iteration $i-1$ and continue the search until we find a valid *greedy solution* or until we run out of all choices (thus output no valid solution is found). Note that the worst case running time of this algorithm is still in $O(N!)$ time, although in best case it runs in $O\left(N^2\right)$ time.

Algorithm 2 Greedy tree search algorithm (GT)

procedure GREEDY-TREE-SEARCH(F) ▷ F is a square lower diagonal matrix of VAFs
  $n \leftarrow size(F)$
  $T \leftarrow [-1]$ ▷ T is a tree represented as a list of parent of each index
  Initialize a new stack *choices*
  choices.push(T)
  while *choices* is not empty **do**
    $T \leftarrow$ *choices*.pop()
    Continue loop if T is invalid (Eq. 2)
    if size(T)$= n$ **then**
      return T
    end if
    c_row $\leftarrow$ Calculate clone frequencies corresponding to T and $F[\text{size}(T)]$ ▷ Eq. 1
    putative_parents $\leftarrow$ argsort(*c_row*)
    for each x in *putative_parents* **do**
      if $x \leq$ size(T) and *c_row*$[x] > 0$ **then**
        new_T $\leftarrow T$
        new_T.append(x)
        choices.push(*new_T*)
      end if
    end for
  end while
  return "No valid solution found!"
end procedure

Addressing sparse time course sequencing data

In practice, because of the often scattered genomic sequencing conducted in a time course, we may observe many mutations at the same time point. If the VAFs of some of these mutations are very similar across the time course, they are likely from the same clone, and thus can be grouped together and represented as a single mutation (group) as described above. If multiple mutations remain not grouped, but are all first observed at the same time point t, multiple clones should have emerged between the time points $t-1$ and t. If the occurrence order of these mutations is determined, we may assume that the VAFs of all variants remain approximately constant between the time points $i-1$ and i. Hence, we can simply extend the VAF matrix into a square lower diagonal matrix by introducing new rows between $t-1$ and t for each mutation that is first observed at t while keeping the remaining VAFs constant. Then we can apply the greedy tree search (GT) algorithm to identify the ML clonal tree. In practice, as we do not know the occurrence order of these mutations, we have to also search for their optimal occurrence order among all possible permutations. An *exhaustive permutation search* (EP; Algorithm 3) would include $\prod_{i=1}^{m} n_i!$ candidate permutations when there are m sets of unordered mutations with cardinalities $n_1, n_2, ..., n_m$, such that $\sum_{i=1}^{m} n_i \leq N$. This is again computationally very expensive.

Greedy permutation search (GP)

To reduce the computational complexity of the *exhaustive permutation search* (EP) algorithm, we propose a heuristic algorithm (for details see Algorithm 4) using a greedy approach as follows. For each set of unordered mutations $group_t$ that first occur at the same time point t, extend the F matrix into a square lower diagonal matrix (up to time t) using each permutation of $group_t$. Then, using the *greedy tree search* (GT; Algorithm 2) find the *ML tree* and the *ML score* for all permutations. The *ML permutation* at time t is determined as the one with the maximum ML score. At any iteration i, if the assignment of a permutation leads only to invalid trees, choose the next optimal choice at iteration $i-1$, and continue the search until we find a valid tree. In the *greedy permutation search* algorithm, we only search $\sum_{i=1}^{m} n_i!$ candidate matrices in the best case, but $\prod_{i=1}^{m} n_i!$ candidates in the worst case.

Constrained search using sequenced clones

Sometimes we have additional information from the experiment when some randomly selected clones are sequenced during or at the end of the experiment. The haplotypes of these clones can be used to improve the search algorithm by enforcing that the clonal tree is consistent with the sequenced clones. There are two constraints that can be checked during the tree search to ensure the consistency. First, let the haplotypes of each sequenced clone (or the variants present in each clone) be represented as sets $A_1, A_2, ...A_m$. Then for each variant v in any sequenced clone, $pr(v) \in \bigcup_{j=1}^{m}(A_j | v \in A_j) - \{v\}$. That is when we know the haplotype of at least one clone that contains variant v, its parent can only be one of the other variants in all the sequenced clones that has v. The second constraint is that for each pair of clones A_i and A_j and any variant $v \in (A_i - A_j) \cup (A_j - A_i)$, all variants $u \in A_i \cap A_j$ must appear before v in the path from the root node to v in the clonal tree. This constraint will make sure that the variants in the symmetric difference

Algorithm 3 Exhaustive permutation search algorithm (EP)

procedure EXHAUSTIVE-PERMUTATION-SEARCH(F)
  groups ← Find groups of mutations whose first occurrence time are equal
  for each *product* in product of permutations of *groups* **do**
    sq_F ← SQUARIFY(F, *product*) ▷ SQUARIFY extends F into a square matrix.
    T ← GREEDY-TREE-SEARCH(*sq_F*) ▷ or EXHAUSTIVE-TREE-SEARCH(*sq_F*)
    likelihood ← LIKELIHOOD(T, *sq_F*) ▷ Compute the likelihood based on Eq. 1
    keep *max_tree* with the maximum value of *likelihood*.
  end for
  return *max_tree*
end procedure

Algorithm 4 Greedy permutation search algorithm (GP)

procedure GREEDY-PERMUTATION-SEARCH(F)
  n ← Number of columns (mutations) in F
  groups ← Initialize queue with groups of mutations whose first occurrence time are equal
  (T, P) ← $([-1], [0])$ ▷ T is a tree. P is a list of indices (permutation).
  Initialize a new stack *choices*
  choices.push((T, P))
  while *choices* is not empty **do**
    (T, P) ← *choices*.pop()
    if size(T)= n **then**
      return T
    end if
    group ← *groups*.next()
    for each p in permutations of *group* **do**
      partial_F ← SQUARIFY($F, P + p$) ▷ SQUARIFY extends F into a square matrix.
      partial_T ← GREEDY-TREE-SEARCH(*partial_F*)
    end for
    Push all valid $(partial\_T, P + p)$ into *choices* in ascending order of their likelihood
  end while
  return "No valid solution found!"
end procedure

of two clones branch off after the common ancestral path is formed by the variants in the intersection of the two clones; otherwise, the infinite sites assumption will be violated.

Metagenome sequencing data from an evolving *E. coli* population

We used data from the LTEE study on an *E. coli* population [4] to validate our methods. We used the paired-end Illumina sequencing reads data for the population 125, on which the metagenome sequencing was performed at six months interval during the course of three years, and eight clones were isolated and sequenced separately at the end of the experiment. We used Trimmomatic version 0.33 [33] to remove adapters and low quality bases and then mapped the reads to *E. coli* K12 MG1655 reference sequence (NC_000913.3) [34] with bwa-mem version 0.7.12 [35]. We removed reads supporting bases with forward/reverse read balance less than 0.25. Then we called variant sites where all of the following conditions satisfied: the VAF (approximated by the ratio of the number of reads supporting the variant allele to the sum of number of reads supporting the reference and variant alleles) was above 0.05, the sum of the number of reads supporting the variant and reference allele was above 10 and the number of variant reads was above 6. Then we removed inconsistent sites comparing the calls from different time points when the VAF at a site becomes zero and then non-zero again at a later time point. The VAFs were input to our algorithms to predict the clonal tree and the clonal frequencies, which were then visualized using Muller plot created using the R library ggmuller [36].

Algorithm 5 Simulation algorithm

procedure SIMULATE(n) ▷ n is number of clones
  T ←$[0] * n$
  C ←$[[0] * n]$
  $C[0][0]$ ← 1
  likelihood ← 1
  for i ← $1, n-1$ **do**
    parent ← random sample from distribution $C[i-1]$
    $T[i]$ ← *parent*
    likelihood ← *likelihood* $* C[i-1][parent]$
    rand ← choose $i+1$ random numbers
    norm_rand ← *rand*/sum(*rand*)
    new_row ← *norm_rand* $+ ([0] * (n-i-1))$
    C.append(*new_row*)
  end for
  return C, T, *likelihood*
end procedure

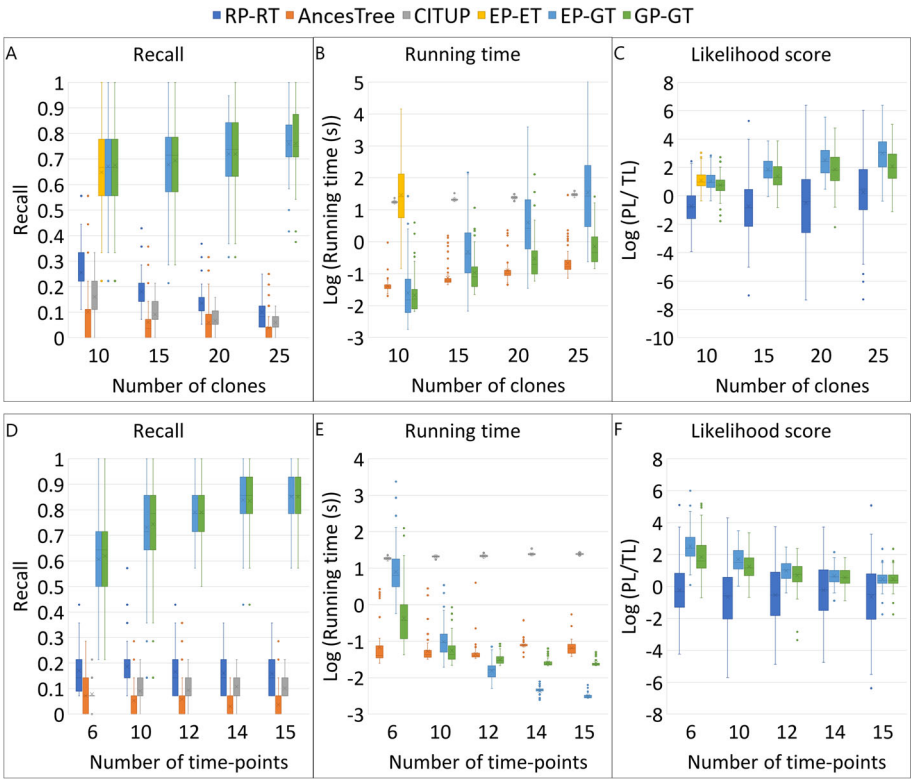

Fig. 2 Comparison of recall, running time and likelihood scores of algorithms: RP-RT (Random Permutation search + Random Tree search), AncesTree, CITUP, EP-ET (Exhaustive Permutation search + Exhaustive Tree search), EP-GT (Exhaustive Permutation search + Greedy Tree search), GP-GT (Greedy Permutation search + Greedy Tree search). PL represents the likelihood score of the predicted clonal tree, and TL stands for the likelihood score of the true tree. Positive $Log(PL/TL)$ indicates the algorithm predicted a clonal tree with greater likelihood than the real tree

Results

We compare the prediction performance of combinations of the two tree search algorithms (*ET* and *GT*) and the two permutation search algorithms (*EP* and *GP*), with two algorithms designed for tumor clonal reconstruction, AncesTree [16] and CITUP [21] on simulated data. *EP-ET* is the slowest algorithm. Hence, this algorithm is applied only to the case where the number of clones is very small. The other two combinations compared are *EP-GT* and *GP-GT*. The combination *GP-ET* does not differ much in performance with *GP-GT*. Therefore, we do not include the results from this combination here. We also compare the results with a baseline (or random) algorithm (*RP-RT*) where at each time-point t the parent is chosen at random from all the clones that have appeared before t. For non-square F matrices, a random permutation is chosen for each group of mutations that appear at the same time. The simulation procedure follows the clonal model described in methods. It starts with a founder clone and then at each new time point a new clone is introduced whose parent is chosen by random sampling from existing clones based on their frequencies in the population. The clonal frequencies are modified following a stochastic process between two consecutive time points (Algorithm 5).

Effect of number of clones

We generated 100 simulations for different number of clones −10, 15, 20, 25, using the simulation algorithm. The number of time points at which the VAFs were observed is sampled from binomial distribution $B(n, 0.6)$ where n is the number of clones. Figure 2 shows the distribution of recall (the proportion of clones correctly reconstructed by the algorithm), the running time in log scale and the log likelihood ratio of the predicted likelihood score over the true likelihood score. Since AncesTree and CITUP can output more than one solution, we calculated recall for each solution and used the maximum recall for comparison. The *GP-GT* algorithm correctly reconstructs almost as many or even more clones on average than *EP-ET* and *EP-GT* algorithms, while having a considerable (2-3 magnitudes) speed advantage (Fig. 2a and 2b). The likelihood score returned by the *EP-ET* algorithm is the upper bound of any ML algorithm because this algorithm traverses the entire space of valid solutions and returns the tree that gives the maximum likelihood score. But

the real clonal tree (following the simulation) may not be the same as the ML tree (Fig. 2a and 2c). As the number of clones increases the likelihood scores returned by *EP-GT* deviates much further from the likelihood score of the true tree compared to the deviation of *GP-GT*, implying that the greedy heuristic not only helps in improving the speed but also reduces error in clonal reconstruction as the number of sequenced clones increases. Notably, AncesTree and CITUP do not take into consideration the sequential order of pooled sequencing data, and thus none of their reported trees are similar to the real clonal trees.

Effect of sparse time course data

To study the effect of sparse time course sequencing data for a fixed number of clones, we generated 100 simulations with fixed number of clones (15), but varied the number of time points, where the VAFs were observed at 6, 10, 12, 14 and 15 time points, respectively. The results are shown in Figs. 2d, 2e and 2f. As the number of time points increases, the number of unordered mutation groups reduces, which in turn reduces the size of the search space for the permutation search. As a result, the recall increases, reaching a maximum of about 0.85 on average when the number of time points is equal to the number of clones, for which no permutation search is needed.

Constrained search

To test the effectiveness of constrained search given a set of sequenced clones we used the simulations generated earlier with 20 clones and compared the performance of the constrained search version of *GP-GT* algorithm by giving different number of true clones as input. We see that as the number of sequenced clones increases, the average number of misconstructed clones decreases, and so does the standard deviation (Fig. 3).

Analysis of metagenome sequencing data from the *E. coli* population

We used the metagenome sequencing data obtained from an LTEE study of an *E. coli* population [4], to validate our methods. It is to be noted that since the proportion of read support is used as an approximation for variant allele frequencies, these values are very noisy. When we did not allow any negative values in the clonal frequency matrix C, our algorithm did not return any valid solution. Thus, we relaxed this criterion to allow cells in C to have negative values greater than -0.4, which are then considered to have the frequency of 0. The input matrix F provided to the algorithm had VAF for 14 mutations at 6 time points. We use four haplotypes from eight sequenced clones (the other four being redundant clones with respect to the variants observed) in the constrained search using *GP-GT*[P]. The resulting clonal tree is shown in Fig. 4a, which is consistent with sequenced clones (highlighted in gray). Figure 4b shows the haplotype of each clone and Fig. 4c shows the Muller plot showing the change in clonal frequencies over time. Note that the negative values in C were set to zero and then the rows were normalized to one. Figure 4d shows the clonal tree obtained when the known clones were not given as constraints. As shown in the simulation experiments, the accuracy of clonal reconstruction can be improved by including more time points, or by sequencing more clones.

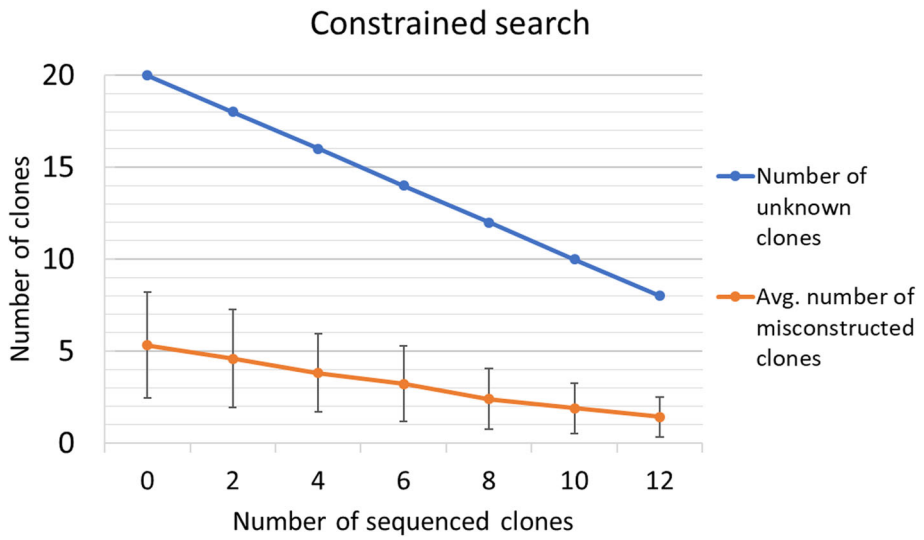

Fig. 3 Results of constrained search on 100 simulations of 20 clones each

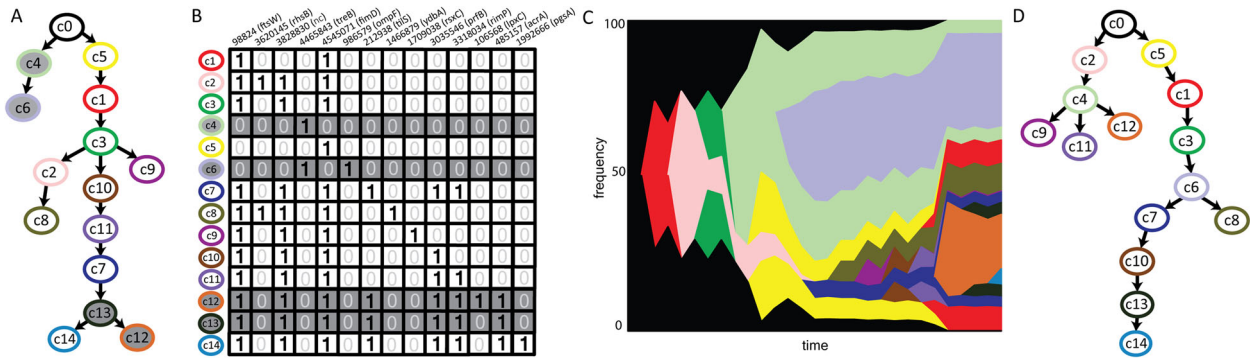

Fig. 4 Clonal reconstruction on the *E. coli* LTEE data. **a** Clonal tree obtained by constrained search with *GP-GT* on the metagenome sequencing data. The four sequenced clones are shown in gray. **b** Reconstructed haplotypes depicted in a table where each row represents a clone and each column represents a variant. The variant locus and the gene where it is located are shown in the column header. Non coding region is marked as nc. When a clone has a specific variant, the corresponding cell is marked as 1, otherwise 0. Sequenced clones are highlighted in gray. **c** The Muller plot showing the predicted clonal frequency changes over time. The colors correspond to those in the clonal tree. **d** Clonal tree obtained without constrained search

Discussion

In this paper, we presented a maximum likelihood framework and a series of greedy-based heuristic algorithms to reconstruct the clonal haplotypes in a bacterial population from metagenome sequencing data obtained in a time course. All these algorithms can tolerate sparse time points sampling (thus multiple clones may arise in the same time period) while the constrained search algorithm can also incorporate clonal sequencing data as additional constraints for reconstructing un-sequenced clonal haplotypes. The results based on simulation experiment showed that, although the clones reconstructed by our algorithms are not identical to the real ones used in the simulation, they are highly similar, and more importantly, the likelihood computed on the reconstructed clones is comparable with (often higher than) the likelihood of the real ones, which implies that our algorithms achieved practically plausible optimal solutions under the maximum likelihood framework. Furthermore, our results demonstrated that the accuracy of clonal reconstruction can be improved by increasing the number of time points for metagenome sequencing or by increasing the number of sequenced clones. In particular, by sequencing more clones, not only the haplotypes of more clones can be directly derived, these derived haplotypes can impose additional constraints on the unknown (minor) haplotypes and thus improve the clonal reconstruction.

We note that the algorithms presented here report not only the haplotypes of clones, but also their frequencies over the time course. The next step after the clonal reconstruction is to identify the clones under selection during the course of evolution based on their frequencies. In this paper, we started to evaluate our algorithms on a relatively simple wild-type *E. coli* population. We plan to apply our algorithms to analyzing the time course genomic sequencing data from DNA repair deficient *E. coli* strains, in which hundreds of mutations occurred [4], to characterize the evolutionary dynamics of the complex population.

Although our algorithms were designed to analyze the sequencing data acquired from LTEE of bacterial populations (e.g., *E. coli*), it may have other applications such as in cancer genomics as described in the introduction. In addition, the metagenome sequencing approach was commonly adopted to study microbial communities containing hundreds of bacterial species, e.g., the human microbiome [37, 38] and the microbiome from natural habitats [39]. Recently, sequencing data acquired from the same microbial community at multiple time points become available [40, 41]. The current analyses of these data focus on the investigation of species and functional diversity in these communities. The computational approaches presented here can also be applied to these data, which will enable haplotype reconstruction of bacterial genomes and may reveal concerted evolution among bacterial species in the community. Interestingly, in the applications to both the cancer genomics and microbiome studies, clonal sequencing can be obtained through single cell sequencing, where our algorithm incorporating the clonal sequencing data can be directly applied.

Conclusions

The main contribution of this paper is to develop a maximum likelihood framework to infer clonal evolutionary history from time course pooled sequencing data. The testing results on the simulation data show that our approach works better than the existing methods that do not take into consideration the sequential order of pooled sequencing data. The algorithms presented here is

ready to be used for the analyses of sequencing data from large-scale LTEEs.

Abbreviations
LTEE: Long-term evolution experiment; SNV: Single nucleotide variation; WT: Wild-type; CFM: Clonal frequency matrix; VAF: Variant allele frequency; ET: Exhaustive tree search; GT: Greedy tree search; RT: Random tree search; EP: Exhausitve permutation search; GP: Greedy permutation search; RP: Random permutation search; ML: Maximum likelihood

Acknowledgements
We thank Drs. Megan Behringer and Michael Lynch for very inspiring discussions.

Authors' contributions
HT and WMI designed the study. WMI contributed tools for the analysis. WMI and HT analyzed the data, and wrote the paper. All authors read and approved the final manuscript.

About this supplement
This article has been published as part of *BMC Genomics Volume 20 Supplement 12, 2019: The International Conference on Intelligent Biology and Medicine (ICIBM) 2019: Bioinformatics methods and applications for human diseases: genomics*. The full contents of the supplement are available online at https://bmcgenomics.biomedcentral.com/articles/supplements/volume-20-supplement-12.

Funding
This research including the publication costs was partially supported by a Multidisciplinary University Research Initiative Award W911NF-09-1-0444 from the US Army Research Office, the National Institute of Health grant 1R01AI108888 and Indiana University (IU) Precision Health Initiative (PHI).

Availability of data and materials
The program (ClonalTREE) is available as open-source software on GitHub at https://github.com/COL-IU/ClonalTREE.

Ethics approval and consent to participate
Not applicable.

Consent to publication
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Published: 30 December 2019

References
1. Elena SF, Lenski RE. Microbial genetics: evolution experiments with microorganisms: the dynamics and genetic bases of adaptation. Nature Rev Genet. 2003;4(6):457.
2. Rainey PB, Rainey K. Evolution of cooperation and conflict in experimental bacterial populations. Nature. 2003;425(6953):72.
3. Barrick JE, Yu DS, Yoon SH, Jeong H, Oh TK, Schneider D, Lenski RE, Kim JF. Genome evolution and adaptation in a long-term experiment with escherichia coli. Nature. 2009;461(7268):1243.
4. Behringer MG, Choi BI, Miller SF, Doak TG, Karty JA, Guo W, Lynch M. Escherichia coli cultures maintain stable subpopulation structure during long-term evolution. Proc Nat Acad Sci. 2018.
5. Burke MK, Dunham JP, Shahrestani P, Thornton KR, Rose MR, Long AD. Genome-wide analysis of a long-term evolution experiment with drosophila. Nature. 2010;467(7315):587.
6. Lenski RE, Rose MR, Simpson SC, Tadler SC. Long-term experimental evolution in escherichia coli. i. adaptation and divergence during 2,000 generations. Am Natural. 1991;138(6):1315–1341.
7. Sniegowski PD, Gerrish PJ, Lenski RE. Evolution of high mutation rates in experimental populations of e. coli. Nature. 1997;387(6634):703.
8. Vasi F, Travisano M, Lenski RE. Long-term experimental evolution in escherichia coli. ii. changes in life-history traits during adaptation to a seasonal environment. Am natural. 1994;144(3):432–56.
9. Wielgoss S, Barrick JE, Tenaillon O, Cruveiller S, Chane-Woon-Ming B, Médigue C, Lenski RE, Schneider D. Mutation rate inferred from synonymous substitutions in a long-term evolution experiment with escherichia coli. G3: Genes, Genomes, Genetics. 2011;1(3):183–6.
10. Blount ZD, Barrick JE, Davidson CJ, Lenski RE. Genomic analysis of a key innovation in an experimental escherichia coli population. Nature. 2012;489(7417):513.
11. Jewett EM, Steinrücken M, Song YS. The effects of population size histories on estimates of selection coefficients from time-series genetic data. Mole Biol Evol. 2016;33(11):3002–27.
12. Taus T, Futschik A, Schlötterer C. Quantifying selection with pool-seq time series data. Mole Biol Evolution. 2017;34(11):23–334.
13. Perdigoto C. Cancer genomics: Tracking cancer evolution. Nature Rev Genet. 2017;18(7):391.
14. McGranahan N, Swanton C. Clonal heterogeneity and tumor evolution: past, present, and the future. Cell. 2017;168(4):613–28.
15. Pon JR, Marra MA. Driver and passenger mutations in cancer. Ann Rev Pathol: Mech Disease. 2015;10:25–50.
16. El-Kebir M A-FHRBOesperL. Reconstruction of clonal trees and tumor composition from multi-sample sequencing data. Bioinformatics. 2015;31(12):.
17. Hajirasouliha I, Mahmoody A, Raphael BJ. A combinatorial approach for analyzing intra-tumor heterogeneity from high-throughput sequencing data. Bioinformatics. 2014;30(12):78–86.
18. Deshwar AG, Vembu S, Yung CK, Jang GH, Stein L, Morris Q. Phylowgs: reconstructing subclonal composition and evolution from whole-genome sequencing of tumors. Genome Biol. 2015;16(1):35.
19. Donmez N, Malikic S, Wyatt AW, Gleave ME, Collins CC, Sahinalp SC. Clonality inference from single tumor samples using low coverage sequence data. Journal of computational biology. 2017;24(6):515–523.
20. McPherson AW, Roth A, Ha G, Chauve C, Steif A, de Souza CP, Eirew P, Bouchard-Côté A, Aparicio S, Sahinalp SC, et al. Remixt: clone-specific genomic structure estimation in cancer. Genome Biol. 2017;18(1):140.
21. McPherson AW, Sahinalp CS, Donmez N, Malikic S. Clonality inference in multiple tumor samples using phylogeny. Bioinformatics. 2015;31(9):1349–56.
22. El-Kebir M, Satas G, Oesper L, Raphael BJ. Inferring the mutational history of a tumor using multi-state perfect phylogeny mixtures. Cell Syst. 2016;3(1):43–53.
23. Qiao Y, Quinlan AR, Jazaeri AA, Verhaak RG, Wheeler DA, Marth GT. Subcloneseeker: a computational framework for reconstructing tumor clone structure for cancer variant interpretation and prioritization. Genome Biol. 2014;15(8):443.
24. Deveau P, Colmet Daage L, Oldridge D, Bernard V, Bellini A, Chicard M, Clement N, Lapouble E, Combaret V, Boland A, et al. Quantumclone: clonal assessment of functional mutations in cancer based on a genotype-aware method for clonal reconstruction. Bioinformatics. 2018;34(11):1808–16.
25. Roerink SF, Sasaki N, Lee-Six H, Young MD, Alexandrov LB, Behjati S, Mitchell TJ, Grossmann S, Lightfoot H, Egan DA, et al. Intra-tumour diversification in colorectal cancer at the single-cell level. Nature. 2018;556(7702):457.
26. Navin NE. Delineating cancer evolution with single cell sequencing. Sci Trans Med. 2015;7(296):296–29.
27. Ross EM, Markowetz F. Onconem: inferring tumor evolution from single-cell sequencing data. Genome Biology. 2016;17(1):69.
28. Tibayrenc M, Kjellberg F, Ayala FJ. A clonal theory of parasitic protozoa: the population structures of entamoeba, giardia, leishmania, naegleria, plasmodium, trichomonas, and trypanosoma and their medical and taxonomical consequences. Proc Nat Acad Sci. 1990;87(7):2414–8.
29. Shapiro BJ. How clonal are bacteria over time?. Curr Opin Microbiol. 2016;31:116–23.
30. Nowell PC. The clonal evolution of tumor cell populations. Science. 1976;194(4260):23–28.
31. Muller HJ. Some genetic aspects of sex. Am Natural. 1932;66(703):118–38.
32. Herron MD, Doebeli M. Parallel evolutionary dynamics of adaptive diversification in escherichia coli. PLoS Biology. 2013;11(2):1001490.
33. Bolger AM, Lohse M, Usadel B. Trimmomatic: a flexible trimmer for illumina sequence data. Bioinformatics. 2014;30(15):2114–20.
34. Blattner FR, Plunkett G, Bloch CA, Perna NT, Burland V, Riley M, Collado-Vides J, Glasner JD, Rode CK, Mayhew GF, Gregor J, Davis NW, Kirkpatrick

HA, Goeden MA, Rose DJ, Mau B, Shao Y. The complete genome sequence of escherichia coli k-12. Science. 1997;277(5331):1453–62.
35. Li H, Durbin R. Fast and accurate long-read alignment with burrows–wheeler transform. Bioinformatics. 2010;26(5):589–95. https://doi.org/10.1093/bioinformatics/btp698.
36. Noble R. R package: ggmuller. https://cran.r-project.org/package=ggmuller. Accessed: 2018-11-04.
37. Huttenhower C, Gevers D, Knight R, Abubucker S, Badger JH, Chinwalla AT, Creasy HH, Earl AM, FitzGerald MG, Fulton RS, et al. Structure, function and diversity of the healthy human microbiome. Nature. 2012;486(7402):207.
38. Methé BA, Nelson KE, Pop M, Creasy HH, Giglio MG, Huttenhower C, Gevers D, Petrosino JF, Abubucker S, Badger JH, et al. A framework for human microbiome research. Nature. 2012;486(7402):215.
39. Gilbert JA, Jansson JK, Knight R. The earth microbiome project: successes and aspirations. BMC Biol. 2014;12(1):69.
40. Narayanasamy S, Muller EE, Sheik AR, Wilmes P. Integrated omics for the identification of key functionalities in biological wastewater treatment microbial communities. Microbial Biotechnology. 2015;8(3):363–368.
41. Halfvarson J, Brislawn CJ, Lamendella R, Vázquez-Baeza Y, Walters WA, Bramer LM, D'Amato M, Bonfiglio F, McDonald D, Gonzalez A, et al. Dynamics of the human gut microbiome in inflammatory bowel disease. Nature Microbiol. 2017;2(5):17004.

Publisher's Note