12-13-2021

Algorithms for Analysis of Heterogeneous Cancer and Viral Populations Using High-Throughput Sequencing Data

Viachaslau Tsyvina

ALGORITHMS FOR ANALYSIS OF HETEROGENEOUS CANCER AND VIRAL POPULATIONS USING HIGH-THROUGHPUT SEQUENCING DATA

by

Viachaslau Tsyvina

Under the Direction of Pavel Skums, PhD

ABSTRACT

Next-generation sequencing (NGS) technologies experienced giant leaps in recent years. Short read samples reach millions of reads, and the number of samples has been growing enormously in the wake of the COVID-19 pandemic. This data can expose essential aspects of disease transmission and development and reveal the key to its treatment. At the same time, single-cell sequencing saw the progress of getting from dozens to tens of thousands of cells per sample. These technological advances bring new challenges for computational biology and require the development of scalable, robust methods to deal with a wide range of problems varying from epidemiology to cancer studies.

The first part of this work is focused on processing virus NGS data. It proposes algorithms that can facilitate the initial data analysis steps by filtering genetically related sequencing and the tool investigating intra-host virus diversity vital for biomedical research and epidemiology.

The second part addresses single-cell data in cancer studies. It develops evolutionary cancer models involving new quantitative parameters of cancer subclones to understand the underlying processes of cancer development better.

INDEX WORDS: quasispecies, next-generation sequencing, haplotype calling, single-cell sequencing, cancer subclones, intra-tumor heterogeneity, phylodynamics

ALGORITHMS FOR ANALYSIS OF HETEROGENEOUS CANCER AND VIRAL POPULATIONS USING HIGH-THROUGHPUT SEQUENCING DATA

by

Viachaslau Tsyvina

A Dissertation Submitted in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

in the College of Arts and Sciences

Georgia State University

2021

ALGORITHMS FOR ANALYSIS OF HETEROGENEOUS CANCER AND VIRAL POPULATIONS USING HIGH-THROUGHPUT SEQUENCING DATA

by

Viachaslau Tsyvina

Committee Chair:	Pavel Skums
Committee:	Alex Zelikovsky
	Robert Harrison
	Ion Mandoiu

Electronic Version Approved:

Office of Graduate Studies

College of Arts and Sciences

Georgia State University

December 2021

DEDICATION

To all my friends

ACKNOWLEDGMENTS

I want to thank my advisor Dr. Pavel Skums for the chance to join his lab and constant support through my studies and research. Most parts of this work were possible only through his continuous guidance and great wisdom. I want to thank Dr. Alex Zelikovsky for the assistance, mentorship, and collaborative work that, in the end, became a part of this thesis.

It was great to work with all the peers in our lab: Dr. Igor Mandric, Dr. Sergey Knyazev, Dr. Pelic Icer, Dr. Andrew Melnik, Dr. Kiril Kuzmin, Fil Rondel, Alina Nemira, and many others.

I want to express appreciation to all my friends that I met in the Computer Science department and with whom we completed this long path together.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Next-generation sequencing (NGS) in viral research

The class of viruses that use RNA to carry genetic information is called RNA viruses[170]. RNA viruses cause diseases such as the common cold, influenza, HIV, COVID-19, hepatitis, Ebola, hepatitis, polio, and measles.

RNA viruses are famous for high mutation rates as high as 10^{-3} substitutions per nucleotide per replication cycle due to error-prone replication. Generally, such mutations are well tolerated, and viruses start to form quasispecies - a viral population represented by a cloud of diverse variants that are genetically linked. The genetic heterogeneity of RNA viruses plays a crucial role in biological implications such as the efficiency of viral transmission, disease progression, evolving resistance to vaccines and antiviral drugs.

With the advent of next-generation sequencing (NGS) technologies, molecular epidemiology and virology are undergoing a fundamental transformation that is already changing our approach to epidemiological data analysis, disease prevention, and treatment[27,32,62,127]. We see that for SARS-CoV-2, the databases contain data for more than a million patients and many countries rapidly scaled up the sequencing of samples[115,110]. This data allows identifying viral populations at great depth and provides new opportunities for dealing with problems like the inference of relatedness between viral samples, identification of quasispecies composition, and outbreak investigation. Most tools, however, deal only with the consensus sequences, ignoring minor haplotypes that can shed light on the development of the disease, its severity, and even enhance the transmission network reconstruction.

In this work, I address two important problems in viral data analysis:

- The search of genetically similar sequences across samples or within a sample. This problem appears in the initial stage of the biological study of viral transmissions. It identifies the set of related sequences from sampled datasets from infected individuals and can help to build the network of intra-host viral variants or antibody clonotypes.
- Reconstruction of quasispecies composition within the sample by building the links between SNPs supported by a statistically significant number of reads and the estimation of haplotypes abundance.

1.2 Next-generation sequencing (NGS) in cancer research

Cancer is a major public health threat responsible for more than 600 000 deaths in the USA annually. It is a disease driven by an uncontrollable growth of cancer cells caused by an extremely complex set of genome mutations and rearrangements varying from a single nucleotide polymorphism to chromothipsis[163]. Clonal heterogeneity plays a vital role in tumor progression and has important implications for diagnostics and therapy, since rare drug-resistant variants could become dominant and lead to relapse in the patient.

Historically most cancer data comes from bulk sequencing. Recently, the most promising technological breakthrough was the advent of single-cell sequencing(scSeq), which allows access to cancer clone populations at the finest possible resolution. This technology provides an opportunity to make significant steps in understanding the evolutionary mechanisms of cancer. It brings single-cell data to the scale of bulk data but keeps more information of cell haplotypes instead of giving a mixture of them.

There were many methods to tackle the problem of evolutionary history reconstruction that gave birth to tools like SCITE[69], infSCITE[83], or SiFit[187]. Most of the tools rely on infinite site assumption and have difficulty with non-perfect phylogeny on two levels:

- Navigate through enormous space of possible topologies with various lost and repeated mutations
- Develop an objective function that can prefer one topology over another because most models try to maximize only the correlation of inferred mutation profiles with observed data and, usually, many alternative topologies give the same likelihood

But besides just topology reconstruction, there is an interest in estimating quantitative features of cancer subclones. It is a pretty recent direction of studies, and there is still no consensus on the rules guiding cancer evolution[34,162,177,119]. The open questions include the laws of evolution (neutral, linear, branching, or punctuated), ways of interaction between clonal variants (competition or cooperation) and the role of epistasis (non-linear interaction of SNVs or genes). In this work, the two evolutionary parameters that try to explore those questions are described: fitness landscape and mutation rate landscape.

As a result, two tools, SCIFIL and MULAN, were developed. Furthermore, they can help to choose alternative topology in conjunction with infSCITE, and MULAN confirms the previous finding that the $SESN2$ gene may lead to genetic instability[68].

1.3 Contributions

The dissertation describes the following contributions:

- Introduces the tool SignatureSJ for fast search of similar genetic sequences in massive databases using Hamming and Edit distances and analyses the possible future improvements. It can be beneficial for the study of HIV and HCV because the amount of collected data increased drastically in recent years.

- Designing a novel haplotype assembly algorithm CliqueSNV[79] based on the representation of haplotype assembly as a clique enumeration problem. The algorithm also estimates frequencies of haplotypes by Expectation-Maximization methods. CliqueSNV is more accurate than other methods that were proven on a series of real and simulated sequencing benchmarks for different viruses.

- Developed evolutionary models for cancer to address highly high intra-tumor heterogeneity. These models introduce two new quantitative features as fitness landscape and mutation rate landscape. As a result, we developed two tools - SCIFIL and MULAN - to calculate those landscapes given cancer mutation tree topology. One of the main advantages of the approach is that these two models agree on the most probable alternative mutation tree topologies allowing us two choose between different evolutionary paths with lost and back-wards mutations.

1.4 Roadmap

The rest of the dissertation is organized as follows. Chapter 2 presents the SignatureSJ algorithm for fast search in genetic databases. Chapter 3 describes CliqueSNV – a tool for viral quasispecies reconstruction and its comparison with state-of-the-art tools. Chapter 4 and Chapter 5 are concentrated on inference of cancer quantitative features for fitness landscape and mutation rate landscape and include two developed tools SCIFIL and MULAN.

1.5 Products

1.5.1 Peer reviewed journals

1. **Tsyvina, V.**, Campo, D. S., Sims, S., Zelikovsky, A., Khudyakov, Y., & Skums, P. (2018). Fast estimation of genetic relatedness between members of heterogeneous populations of closely related genomic variants. BMC bioinformatics, 19(11), 1-10.

2. S. Knyazev*, **V. Tsyvina***, A. Shankar, A. Melnyk, A. Artyomenko, T. Malygina, Y. Porozov, E. Campbell, S. Mangul, W. Switzer, P. Skums, and A. Zelikovsky (under revision) Accurate Assembly of Minority Viral Haplotypes from Next-Generation Sequencing through Efficient Noise Reduction. Nucleic Acids Research *-equal contribution

3. Skums, P., **Tsyvina, V.**, & Zelikovsky, A. (2019). Inference of clonal selection in cancer populations using single-cell sequencing data. Bioinformatics, 35(14), i398-i407.

4. **Tsyvina, V.**, Zelikovsky, A., Snir, S., & Skums, P. (2020). Inference of mutability landscapes of tumors from single cell sequencing data. PLOS Computational Biology, 16(11),

e1008454.

5. Rogovskyy, A. S., Caoili, S. E. C., Ionov, Y., Piontkivska, H., Skums, P., **Tsyvina, V.**, ... & Waghela, S. D. (2019). Delineating surface epitopes of Lyme disease pathogen targeted by highly protective antibodies of New Zealand White rabbits. Infection and immunity, 87(8).

1.5.2 Abstracts and Presentations

1. **Tsyvina, V.**, Zelikovsky, A., Skums, P. Inference of intra-host SARS-CoV-2 heterogeneity from noisy NGS data(ICCABS 2020)

2. **Tsyvina, V.**, Zelikovsky, A., Snir, S., Skums, P. Inference of mutability landscapes of tumors from single cell sequencing data (RECOMB-CCB 2020)

3. Skums, P.,**Tsyvina, V.**, Zelikovsky, A. Inference of clonal selection in cancer populations using single-cell sequencing data at ISMB 2019

4. Skums, P.,**Tsyvina, V.**, Zelikovsky, A. Joint inference of evolutionary inference and fitness landscape of a tumor from bulk and single-cell sequencing data" at ICCABS 2019

5. Sergey Knyazev, **Viachaslau Tsyvina**, Andrew Melnyk, Alexander Artyomenko, Tatiana Malygina, Yuri B Porozov, Ellsworth Campbell, William M Switzer, Pavel Skums, and Alex Zelikovsky (2018) CliqueSNV: Scalable Reconstruction of Intra-Host Viral Populations from NGS Reads. The 8th RECOMB Satellite Workshop on Massively Parallel Sequencing (RECOMB-Seq)

1.5.3 Software Packages

1. SignatureSJ. Tool for retrieving related sequenced grom large genomic databases. `https://github.com/vtsyvina/signature-sj`

2. CliqueSNV. Tool for restoring virus variants from NGS sequencing data, SNPs variant calling. `https://github.com/vtsyvina/CliqueSNV`

3. SCIFIL. Estimation of mutations fitness landscape. `https://github.com/compbel/SCIFIL`

4. MULAN. The tool to infer mutation rated from single-cell DNA data. `https://github.com/compbel/MULAN`

CHAPTER 2

FAST ESTIMATION OF GENETIC RELATEDNESS BETWEEN MEMBERS OF HETEROGENEOUS POPULATIONS OF CLOSELY RELATED GENOMIC VARIANTS

List of abbreviations

- NGS: Next-generation Sequencing
- HVR1: Hypervariable Region 1
- HCV: Hepatitis C Virus

2.1 Background

We Consider two sets T_1 and T_2 each containing N DNA or RNA sequences of length L. The *similarity join problem* consists in locating the set P of all pairs of sequences, with one sequence from T_1 and the other from T_2, within an edit distance or Hamming distance defined by the specified threshold t. In molecular epidemiology, this computational problem needs to be solved for detection of viral transmissions from sequences of intra-host viral variants sampled from infected individuals[25,140]. Viral populations, for which the minimal inter-sample distance does not exceed the threshold, are considered to be potentially linked by transmission[25], while the number of pairs in P may suggest the time since a transmission event[56]. The related *genetic network construction* problem aims to build a graph with vertices corresponding to sequences from a given dataset T and edges corresponding to all pairs of sequences with an edit or Hamming distance less than the threshold t. This problem arises in studying and analysis of viral populations[156] or antibody repertoires[149]. Similar problems also emerged under different names in various areas of computer science[131,55,88,108,118].

The edit distance between a pair of sequences can be calculated in time $O(L^2)$ using dynamic programming[171]. If only distances below a desired threshold t which is small relative to L are desired. The distance calculation can be carried out with a small subset of diagonals neighboring the main diagonal of the dynamic programming matrix, leading to $O(tL)$ time algorithm[61]. In this case a naïve algorithm for the similarity join problem requiring pairwise comparison of all sequences has an asymptotic running time $O(tLN^2)$, which is still impractical for more than several thousand sequences.

Several *filtering-based approaches* have been put forward to improve the efficiency of the similarity join-type problems by reducing the number of pairs to be compared. Note that while fast heuristic and approximate methods exist such as Shingling[21], LSH[55], or BLAST[4], this paper focuses on the problem of exact distance calculation.

The common filtering approach is based on on the fundamental idea that related sequences should share long *k-mers* (substrings of length k)[93]. Several existing methods rely on signature schemes to quickly locate feasibly linked pairs[131] by assuming that pairs with an edit or Hamming distance which does not exceed a threshold t will share at least a certain number of k-mer-based signature keys. However, straightforward application of this technique to viral sequencing data is not sufficiently efficient, since mutations are not distributed uniformly along viral genomes, but tend to concentrate in short hypervariable regions[33]. As a result, many viral sequences share k-mers, thus significantly reducing the efficiency of filtering. The same effect has been observed for immunosequencing data[149], where all antibodies originating from the same V gene often share a k-mer from that gene.

In this paper, we describe a tool which uses k-mer-based signature filtering scheme optimized for viral data to solve the following problems:

- *Sample pair filtering*: given two NGS sequence samples T_1 and T_2, quickly determine whether the distances between all inter-sample pairs of sequences are greater than the threshold t.
- *Inter-sample sequence retrieval* (similarity join): given two NGS sequence samples T_1 and T_2, find all inter-sample pairs of sequences at edit distance or hamming distance below the threshold t.
- *Intra-sample sequence retrieval* (or genetic network construction): given an NGS sequence sample T_1, find all pairs of sequences at edit distance or hamming distance below the threshold t.

The tool was validated using Hepatitis C Virus (HCV) data in the settings used for detection of viral transmissions and outbreaks[25,140].

2.2 Methods

2.2.1 Notation

In the methods description, we assume that input sequence samples T_1 and T_2 both contain N sequences of length L, which cover the same genomic region. From here onwards we will use k

as a fixed predefined parameter.

Further we will use the following notation:

- $S = s_1 s_2 \dots s_L$ - sequence over the alphabet $\{A, C, T, G\}$.
- $S[i : j] = s_i s_{i+1} \dots s_j$ - subsequence of S starting at position i and ending at position j.
- *k-mer* - any subsequence of length k
- *k-segment* - k-mer that starts at a position $1 + ik$, $i = 0, 1, 2, ...$(i.e. first k-segment starts at first position in sequence, second starts right after first).
- $K(S)$ - the set of all k-mers of the sequence S.
- $R(S)$ - the list of all k-segments of the sequence S (possibly with repetitions).
- $h(S, Q)$ - Hamming distance between two sequences S and Q
- $l(S, Q)$ - edit distance (Levenshtein distance) between two sequences S and Q
- $led(S, Q) = \begin{cases} l(S, Q), & \text{if } l(S, Q) \leq t \\ -1, & \text{otherwise} \end{cases}$ - limited edit distance, as mentioned above, could be calculated using dynamic programming[61]

2.2.2 Main Data Structure

Our signature-based filtering scheme is based on the following simple observation:

Proposition 1. [131] *If $l(S, Q) \leq t$, then $|K(Q) \cap R(S)| \geq m - t$, where $m = \lfloor \frac{L}{k} \rfloor$.*

Proof. If S and Q differ by an edit distance of t, then by the pigeon hole principal at most t k-segments differ between the sequences S and Q. So at least $m - t$ k-segments must be the same. □

Thus we need a fast way to calculate the number of common k-segments and k-mers for a given pair of sequences. To do it we introduce a hash function:

$$hash(S[i:j]) = \sum_{l=i}^{j} f^{j-l}(s_l), \tag{2.1}$$

where $f : \{A, C, G, T\} \rightarrow \{0, 1, 2, 3\}$ is an arbitrary bijection. For k-mers with $k < 32$, this hash function allows us to store them as 64-bit integers and can be quickly recursively calculated as follows:

$$hash(S[i+1 : j+1]) = hash(S[i:j]) - 4^{n-1}f(s_i) + f(s_{j+1}) \tag{2.2}$$

In addition, the hash can be inverted and so only the hash values of $k - mers$ need to be stored.

In the proposed framework, each sample T is stored using a data structure further referred to as a T-dictionary and denoted by $dict(T)$, which consists of the following fields:

- $dict(T).HM$ - an inverted index of T^{191}, i.e. a hash table, where each key is a k-mer hash and its value is a set of all sequences from T that contain this k-mer.
- $dict(T).KM$ - A set of all possible k-mer hashes in T
- $dict(T).KS$ - hash table, where keys are sequences and values are lists of their k-segments (represented by their hash values) from 1 to m

- $dict(T).SC$ - A list of L sets $SC_1, ..., SC_m$, where SC_i is a set of all k-segments in a position $1 + ik$ (represented by their hash values).

2.1.3 Algorithm Description

We will first describe the approach for the sample pair filtering problem. Building a simple and fast filter for unrelated samples T_1 and T_2 is easy by applying Proposition 1 to whole samples as follows. Recall that T_1 and T_2 are considered to be genetically related, if the minimal edit distance between their sequences does not exceed the threshold t. Given two dictionaries $dict(T_1)$ and $dict(T_2)$, the necessary condition for their genetic relatedness is an existence of at least $m - t$ positions $\{i_1, i_2, \ldots, i_{m-t}\}$ such that $dict(T_1).SC_{i_j} \cap dict(T_2).KM \neq \emptyset$ for every $j = 1, ..., m-t$. The sample pair filter pseudocode is presented at Algorithm 1.

Algorithm 1 Simple filter for unrelated samples

```
function CALCULATECOINCIDENCES(dictT1, dictT2, m, t)
    coincidences = 0
    for lSegmentHashes $\in$ dict1.SC do
        for hash $\in$ lSegmentHashes do
            if hash $\in$ dict2.KM then
                coincidences $\leftarrow$ coincidences $+ 1$
                break
            end if
        end for
    end for
    return coincidences $\geq m - t$
end function

```

Assuming that membership verification for a hash set $dict(T).KM$ can be performed in time $O(1)$, the worst-case running time of the filter is $O(NL)$. In real settings, samples with genetically related sequences produce significantly smaller maps $dict(T).SC$, thus leading to a lower average running time than in the worst case.

The algorithms for inter-sample sequence retrieval and intra-sample sequence retrieval problems are very similar, so we will describe the approach for the former problem. As before, let T_1 and T_2 be two samples. The algorithm first constructs the set of *candidate neighbors* $CN_S \subseteq T_2$ for every sequence $S \in T_1$. This procedure (*the filtering*), is followed by the *verification* procedure, which calculates actual neighbors of all sequences $S \in T_1$ by calculating distances between S and all sequences $S' \in CN_S$. The pseudocode for inter-sample sequence retrieval algorithm is presented in Algorithm 2.

The basic filtering strategy utilizes Proposition 1, with the following features aiming at improvement of the running time. For each $S \in T_1$, the set CN_S can be implemented as a hash table, with keys being sequences $S' \in T_2$ and values $CN_S(S')$ being numbers of matches between k-segments in S and k-mers of S'. Let L_S be the number of k-segments of S that occur as k-mers in T_2, and $I = (i_1, i_2, \ldots .i_{L_S})$ be the list of starting positions of these k-segments. To calculate the number of matches between k-segments in S and k-mers of S' we may iterate over the list I and increment the current value of $CN_S(S')$, when necessary. If after j iterations the inequality

$$m - t \leq L_S - j + CN_S(S') \tag{2.3}$$

does not hold, then S' cannot accumulate the required number of matches with the remaining iterations, and therefore the sequence S' can be filtered out right away. These considerations imply that the order in which starting positions of k-segments are examined is important in determining the algorithm's running time.

The order of k-segment starting positions is determined heuristically as follows. For each position i let $k_S(i) = |dict(T_2).HM(S[i : i + k - 1])|$ be the number of sequences from T_2 that contain the i-th k-segment from S. If we sort positions by ascending order of the numbers $k_S(i)$ it usually leads to faster pruning of sequence pairs as this order minimizes the size of the candidate set that must be examined at each iteration.

Another simple adjustment could be implemented using the fact that the hamming distance is an upper bound for an edit distance, while the calculation of the former is significantly faster. Therefore if $h(S, Q) \leq t$, then Q can be added to the list of neighbors of S without the edit distance calculation.

2.2 Hamming distance adjustment

The filtering strategy described above can be further improved, if the input sequences are aligned to a reference. In this case the samples can be compared using Hamming distance instead of an edit distance. For Hamming distance, Proposition 1 can be applied to k-segments of both comparable sequences thus simplifying the filling and filtering steps.

Furthermore, genomic heterogeneity is distributed highly irregularly along the genomes of species of interest. For example, Fig. 2.1 illustrates the distribution of nucleotide entropy for a particular intra-host population along the 264bp-long genomic HCV region at the junction of envelope glycoproteins E1 and E2, which is often used in epidemiological and immunological studies[124,10,25]. It should be noticed that k-segments from conserved regions are significantly less useful for the filtering as we want to maximize detectable differences between tested sequences. The non-uniformity in genomic heterogeneity can be taken into account by switching to the framework with k-segments of unequal size. By selecting k-segment boundaries that con-

Algorithm 2 Signature-based filter to find sequence pairs closer than threshold

```
1:  function SIGNATUREFILTER(T1, T2, dictT1, dictT2, t)
2:      for s ∈ T1 do    ▷ lines 4 through 12 calculate candidates list $CN_s$ for each sequence $s$
3:          $CN_s$ ←hash map (sequence→hit count)
4:          order ← indexes of segments sorted in ascending order by their frequencies in $T_2$
5:          $L_s$ ← number of $k$-segments from s that appear in T2
6:          for i $= 0 \to L_s$ do
7:              if i $\leq L_s - (m - t)$ then
8:                  FILL($CN_s$, s, dictT1, dictT2, order, i)    ▷ Fill candidates list
9:              else
10:                 $CN_s$ ← FILTER($CN_s$, s, dictT1, dictT2, order, i)    ▷ Filter sequences from candidates list that do not share $m - k$ $k$-mers
11:             end if
12:         end for
13:         for s′ ∈ keys of $CN_s$ do
14:             if h(s, s′) $\leq t$ or led(s, s′) $\neq -1$ then
15:                 s and s′ are related
16:             end if
17:         end for
18:     end for
19: end function
20:                 ▷ function FILL adds all sequences from $T2$ that share the same $k$-mer with $s$
21: function FILL($CN_s$, s, dictT1, dictT2, order, i)
22:     segmentHash ← dictT1.KS[s][order[i]]
23:     for s′ ∈ dict2.HM[segmentHash] do
24:         add {s′,1} to $CN_s$ or increment current value for key s′
25:     end for
26: end function
27:                 ▷ function filters candidate sequences if they do not share enough $k$-mers with $s$
28: function FILTER($CN_s$, s, dictT1, dictT2, order, i)
29:     segmentHash ← dictT1.KS[s][order[i]]
30:     filteredCandidates ← hash map (sequence→hit count)
31:     for s′ ∈ $CN_s$ do
32:         isInDict ← s′ ∈ dictT2.HM[segmentHash]
33:         if isInDict or m − t $\leq L_s$ − i + candidates[s′] then
34:             addVar ← 1 if isInDict is true, 0 otherwise
35:             add {s′, $CN_s$[s′] + addVar} to filteredCandidates
36:         end if
37:     end for
38:     return filteredCandidates
39: end function
```

tain roughly equal amounts of average information entropy over the dataset, the filtering speed and quality could be significantly improved. Figure 2.2 provides an example, when entropy-based k-segments length allows more accurate filtering than uniform k-segments length. Formally, let $H_i = -\sum_{j=1}^{4} P(x_j^i) log_2(P(x_j^i))$ be the sample nucleotide entropy at position i, where $P(x_j^i)$ is a frequency of nucleotide x_j^i on i-th position of the alignment. The segments are selected in such a way that for every segment $[i, j]$ we have $\sum_{l=i}^{j} H_l \approx \frac{H}{m}$, where $H = \sum_{i=1}^{L} H_i$ and m is the number of segments. Different numbers of segments were examined empirically and the best performance was obtained with $m = t + 7$.

Figure 2.1 Distribution of nucleotide entropy along the E1/E2 region of HCV for a population of 469 unrelated genotype 1a sequences obtained from NCBI.

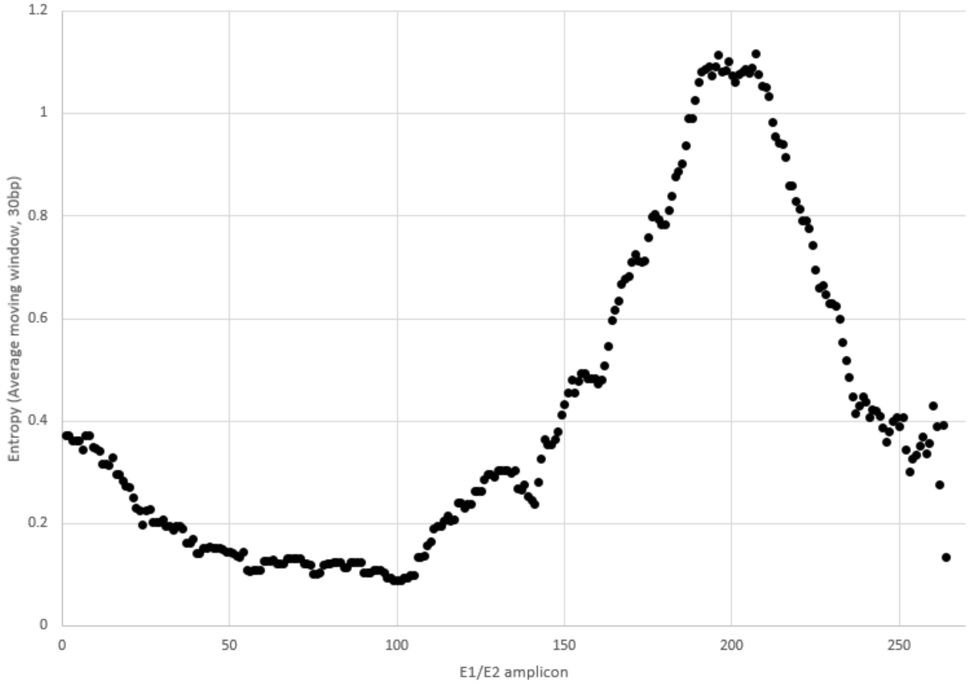
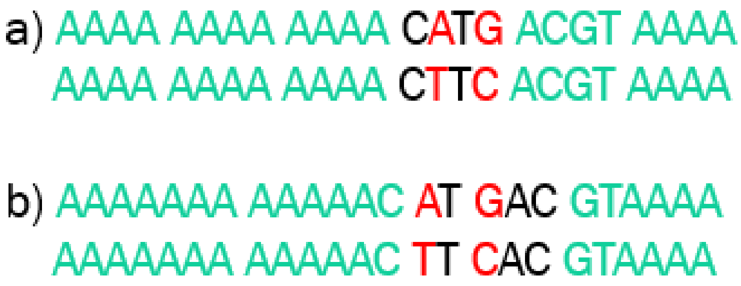


Figure 2.2 Example of two exact pairs of strings, but with equal ($k = 4$) (a) and entropy-based (b) segments size and $t = 1$. In case (a) the pair passes the filter, in case (b) it doesn't pass the filter.


a) AAAA AAAA AAAA CATG ACGT AAAA
AAAA AAAA AAAA CTTC ACGT AAAA
b) AAAAAAA AAAAAC AT GAC GTAAAA
AAAAAAA AAAAAC TT CAC GTAAAA

2.3 Results

2.3.1 Validation Data

The developed tool was validated using NGS datasets of intra-host HCV populations sampled from infected individuals. Each dataset contains the E1/E2 junction of the HCV genome of length 264nt, which contains the Hyper Variable Region 1 (HVR1) region. Each sample was processed by error correction and haplotyping tools, and as a result we receive as an input datasets consisting of unique HCV haplotypes.

We used a set of 413 samples from[140] with 501.5 haplotypes per sample in average produced by NGS; 8 datatsets $d_1, ..., d_8$ with 1000, 2000, . . . , 128 000 sequences produced by random sampling from NGS dataset with sequences sampled from chronically infected individuals and one additional NGS dataset m_1 consisting of 10 467 sequences. The data are available in tool's repository.

In all tests, the threshold $t = 3.77\% \equiv 10$nt was used. This value is derived in[25] as empirically validated recommended threshold for separation between epidemiologically related and unrelated intra-host HCV HVR1 populations.

All tests were run on server with 128 Intel Xenon E7-4850 2.1GHz cores and 1.4Tb RAM. For Inter-sample sequence retrieval desktop PC was used with 4 Intel(R) Core(TM) i7-5500 2.4GHz cores and 8Gb RAM. All code is written on Java to provide a threaded, platform independent solution.

2.3.2 Sample pair filtering and Inter-Sample Sequence Retrieval validation

For Sample pair filtering and Inter-Sample Sequence Retrieval problems, we validated the tool using HCV datasets from[140]. The proposed approach has been compared with the Filter Composition pipeline proposed in[140]. Both methods were run on a desktop computer, as in the original paper[140]. The results are reported in Table 2.1. Here we show the result of comparison of all pairs of samples and all inter-sample pairs of sequences.

Table 2.1 Results of Filter Composition pipeline and k-mer based signature scheme filtering for Sample pair filtering and Inter-Sample Sequence Retrieval problems

Method	Filter Composition	Signature Scheme
Percent of filtered sample pairs	85.1%	92%
Percent of filtered sequence pairs	91.5%	99.996%
Total Time	$\sim$ 5 min	$\sim$ 15 sec

The proposed sample pair filtration algorithm removed 92% of all possible samples pair comparisons, and sequence pair filtering algorithm managed to filter out 99.996% of all possible sequence pairs. The latter means that only 888,914 out of 22,037,502,011 sequence pairs passed from filtering to verification stage of the algorithm. As a result, the proposed approach significantly outperforms the Filter Composition Pipeline in filtering quality and in running time.

We studied how the filtering quality is affected by different optimization subroutines (Table 2.2). Disabling sample pair filtering increases the running time for comparison of all samples by

42%, while the impact of sorting of k-segment starting positions is even higher, with disabling of this step slowing down the comparison by 254%.

Table 2.2 Algorithm run time without optimization subroutines

Feature	Time
No sample pair filter	$\sim 21.3s$
No sorting of k-segment starting positions	$\sim 38.1s$

Preprocessing and dictionary building can take up a significant portion of the total running time of a signature-based filtering algorithm, when samples are distant and few distance calculations are required. For the given collection of 413 samples, preprocessing of all samples takes $\sim$ 4840ms, which constitutes approximately 1/3 of the total running time of the algorithm. Note that in the case when significant number of closely related sequence pairs is present, the situation is different (see the next section).

The algorithm performance depends on the size of the k-mers and k-segments. Small k leads to larger number of matches between k-segments and k-mers of distant sequences, which can cause extra sequences to be added to the candidate lists thus leading to decrease in filtering quality. Larger k leads to fewer false matches but unfortunately also a larger k-mer dictionaries. We examined different k-mer sizes to determine the optimal size for our datasets and found that $k = 11$ gives the best performance.

2.3.3 Intra-sample Sequence Retrieval Validation

For Intra-sample Sequence Retrieval Problem, we validated the proposed approach using datasets d1,...,d8,m1. First, it was compared with a single-thread, brute force method with the worst-case complexity $O(N^2Lt)$, which performs pairwise comparison of all sequences and calculates limited edit distance using dynamic programming as described in[61]. The results are presented in Table 2.3.

Table 2.3 Intra-sample Sequence Retrieval Running Time

Dataset	Pairs in output	Brute force time, s	Signature method time, s
d1	60 421	6.6	0.2
d2	370 262	25.9	0.3
d3	1 800 945	102	1.8
d4	5 848 556	413	2.8
d5	18 570 536	1 624	4
d6	38 835 302	6 499	7.8
d7	155 373 208	26 400	23
d8	621 556 832	105 555	83
m1	51 453 578	883	17

The running time of the proposed tool was also compared with the running time of a recently published method from[149], which was originally designed for the analogous problem for immunosequencing data. Fig. 2.3 illustrates that signature-based filtering approach demonstrates the significant advantage.

Fig. 2.4 demonstrates that for aligned sequences in most cases the adjustment utilizing entropy-based variable-size k-segments allows to achieve a significant speedup with respect to a constant-size k-segment.

The speedup described above is achieved by the combination of the several features. The first feature is the quality of filtering, which is analyzed in Table 2.4 and Table 2.5. On average, only $\sim 10\%$ of sequence pairs that pass filtering step (”false positives”) are not genetically related. As expected, most of the false positive pairs were very close to the threshold (Figure 2.5). With the threshold set at $t = 10$, pairs with an edit distance of $l(S, Q) = 11, 12, 13$ represent up to 75% of all false positives. Pairs that are so close to the threshold are difficult to filter out.

Another important feature is the fact that as the input increases in size the runtime of the al-

Figure 2.3 Running times of method from[149] (blue) and the proposed method (red) on datasets d1-d8

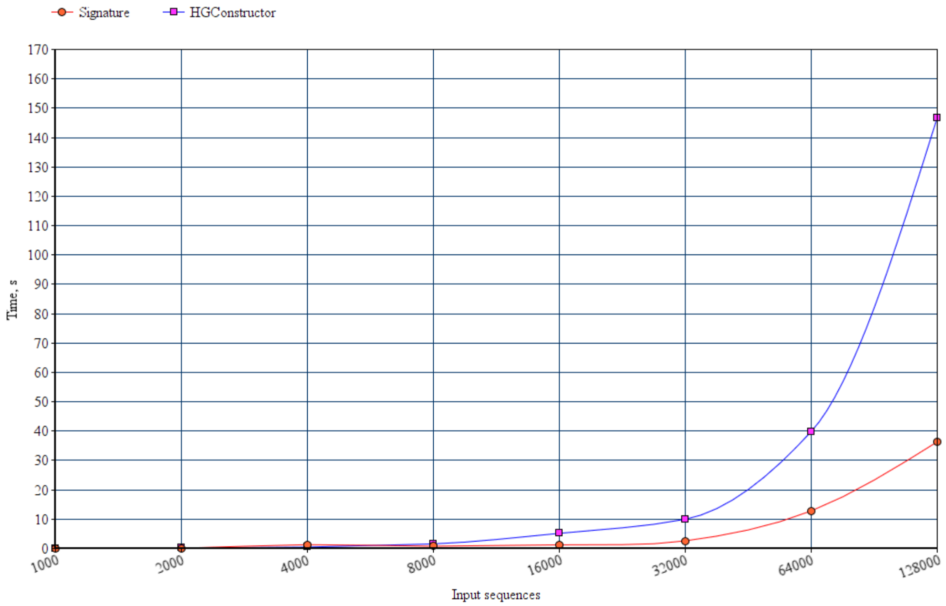


Figure 2.4 Comparison of running times of equal segment size and entropy-based approaches for single sample problem

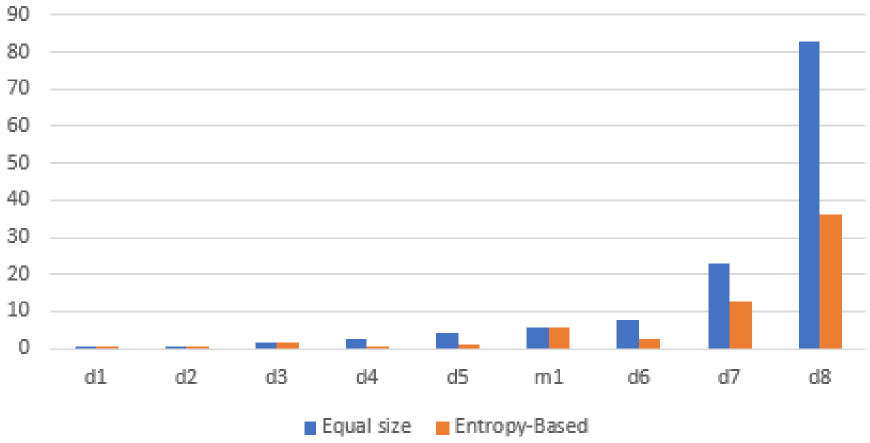
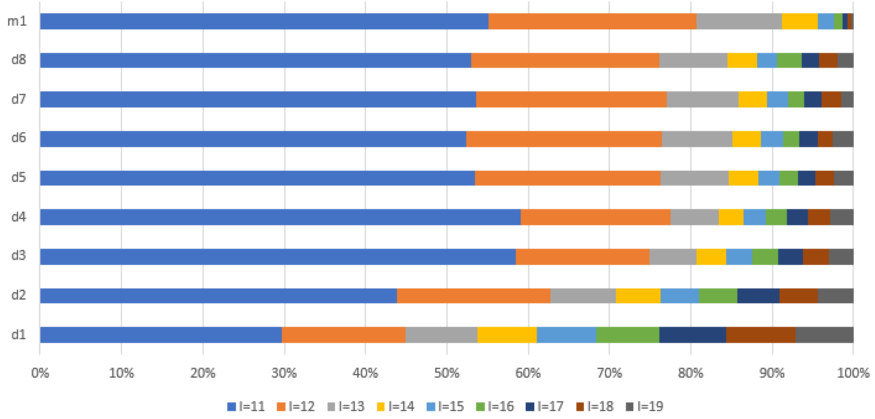
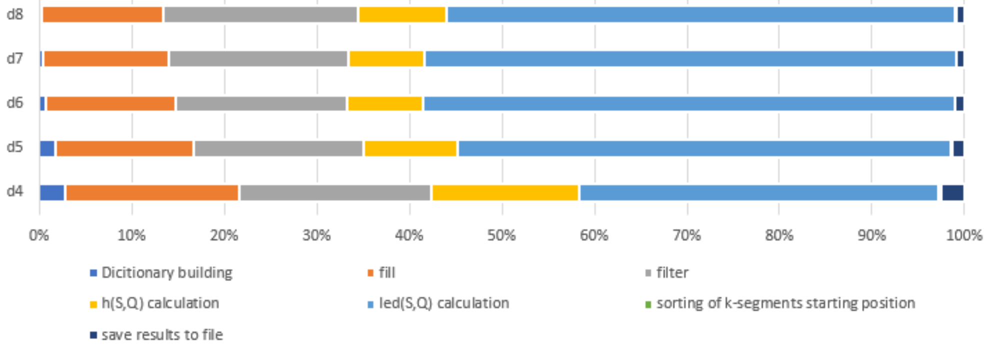


Figure 2.5 False positive sequence pairs($l(S, Q) > t$) at different edit distances l



gorithm is dominated by the edit distance calculations (Figure 2.6). However, the filtering and the Hamming distance shortcut reduces the number of edit distance calculations that must be performed. As a result, the actual edit distance is only calculated on small portion of the total pairs from the dataset (Table 2.4).

Figure 2.6 Contribution of algorithm subroutines to its total running time, unaligned sequences



We attempted to improve the filtering performance using other methods such as k-mer similarity[125], true matches[131], Hamming radius filter[140]. However, the overhead of these methods was greater than any runtime savings.

Table 2.4 Filtering quality (unaligned sequences)

Test	Pairs in output	Pairs that passed filter	Filtering PPV	$\# \, led(S,Q)$ calculations	$\frac{led(S,Q)}{allpairs}$
d1	60 421	65 937	0.9163	5 517	1.1%
d2	370 262	397 987	0.9303	18 754	0.93%
d3	1 800 945	1 873 268	0.9614	72 820	0.91%
d4	5 848 556	6 256 934	0.9347	411 660	1.28%
d5	18 570 536	21 028 890	0.8831	2 477 531	1.94%
d6	38 835 302	46 744 915	0.8308	7 952 495	1.55%
d7	155 373 208	187 011 650	0.8308	31 809 970	1.55%
d8	621 556 832	748 119 580	0.8308	127 239 860	1.55%
m1	51 453 578	54 640 978	0.9417	7 303 118	14.2%

Table 2.5 Filtering quality (aligned sequences)

Test	Pairs in output	Pairs that passed filter	Filtering PPV
d1	60 420	64 573	0.9357
d2	379 233	385 646	0.9834
d3	1 800 448	1 862 914	0.9665
d4	5 845 274	6 204 049	0.9422
d5	18 551 359	20 706 813	0.8959
d6	38 792 420	44 939 957	0.8632
d7	155 201 680	179 791 828	0.8632
d8	620 870 720	719 231 312	0.8632
m1	47 101 270	48 888 011	0.9635

2.4 Discussion

In this paper we presented an efficient signature-based tool to solve problems of edit or Hamming distance sequence retrieval for NGS data obtained from heterogeneous viral populations. It outperforms other approaches to this problem by including several data-specific steps and filters. The proposed approach was designed having problems of computational molecular epidemiology in mind. Until recent years, genomic analyses of viral transmissions and epidemic spread used a single viral sequence per infected individual. The advent of sequencing technologies now allows to analyze thousands of viral haplotypes per patient. Furthermore, just in the United States, from 2.7

million to 3.9 million people are infected with HCV[172], while ~ 1.1 million people are infected with HIV[48]. These numbers put an immense computational burden on real-time advanced molecular surveillance systems, such as Global Hepatitis Outbreak Surveillance Technology (GHOST)[89], which is currently being deployed by Centers for Disease Control and Prevention. When deployed, such system should have computational capacity to identify, whether a query set of viral samples is genetically related with any sample from a database consisting of hundreds of thousands of samples each consisting of thousands of sequences. The proposed approach aim to allow to process such queries efficiently. It builds on the general idea proposed in[131], which is heavily optimized by utilization of efficient data structures, such as inverted indexes and hash maps, and introduction of running time-improving procedures, such as efficient hash values calculation and determination of optimal order of k-mers processing. The proposed optimization steps allows for more than 2.5-fold running time decrease in comparison with the non-optimized filtering (Table 2.2). Furthermore, the proposed method takes into account uneven distribution of heterogeneous position along viral genomes by using variable entropy-based k-mers. It allows to improve both filtering quality (Fig. 2.2) and speed (Fig. 2.4). In general, for viral samples comparison the proposed filtering approach allows to eliminate the overwhelming majority of sequence comparisons and achieve a substantial running time decrease (Tables 2.1- 2.5).

2.5 Conclusion

The proposed tool allows for efficient detection of genetic relatedness between genomic samples produced by deep sequencing of heterogeneous populations. The tool is freely available for down-load at `https://github.com/vyacheslav-tsivina/signature-sj`. It should be especially useful for analysis of relatedness of genomes of viruses with unevenly distributed variable genomic regions, such as HIV and HCV. For the future we envision, that besides applications in molecular epidemiology the tool can also be adapted to immunosequencing and metagenomics data.

CHAPTER 3

CLIQUESNV: SCALABLE RECONSTRUCTION OF INTRA-HOST VIRAL POPULATIONS FROM NGS READS

3.1 Introduction

Rapidly evolving RNA viruses such as human immunodeficiency virus (HIV), hepatitis C virus (HCV), influenza A virus (IAV), SARS, and SARS-CoV-2 form populations of closely related genomic variants inside infected hosts[74,63,90,42,105,161,37,135,182,148]. The intra-host viral populations include minority viral variants that are frequently responsible for drug resistance, immune escape, and disease transmission[13,39,49,66,134,24,153,25,56,155,181,109,17]. Therefore, accurately predicting minority viral populations from extremely large and noisy viral genomic data is important for biomedical research, epidemiology, and clinical applications. Although this problem has recently attracted significant interest from the biomedical research community[38,9,53], numerous obstacles still delay NGS integration into the viral studies. The last decade witnessed numerous attempts to employ NGS and bioinformatics methods for reconstructing intra-host viral populations. These methods are not accurate enough for clinical and epidemiological applications since they cannot reliably identify haplotypes accounting for a substantial portion of the population. Existing methods are ill-equipped to assemble closely related haplotypes and have elevated false-positive rates. Additionally, there is only one in vitro viral sequencing benchmark for validation of haplotyping tools[53], and to convincingly demonstrate that such tools are ready for clinical and epidemiological applications, new comprehensive sequencing benchmarks are urgently required[78].

Next-generation sequencing (NGS) technologies now provide versatile opportunities to study viral populations. In particular, the popular Illumina MiSeq/HiSeq platforms produce 25-320 million reads, which allow multiple coverage of highly variable viral genomic regions. This high coverage is essential for capturing rare variants. Ability of NGS technologies to efficiently identify minority variants have recently gained FDA approval[122]. However, *haplotyping* of heterogeneous viral populations (i.e., assembly of full-length genomic variants and estimation of their frequencies) is extremely complicated due to the vast number of sequencing reads, the need to assemble an unknown number of closely related viral sequences and to identify and preserve low-frequency variants. Single-molecule sequencing technologies, such as PacBio, provide an alternative to short-read sequencing by allowing full-length viral variants to be sequenced in a single pass. However, the high level of sequence noise due to background or platform-specific sequencing errors produced by all currently available platforms makes inference of low-frequency genetically close variants especially challenging, since it is required to distinguish between real and artificial genetic heterogeneity produced by sequencing errors.

Recently, a number of computational tools for inference of viral quasispecies populations from NGS reads have been proposed[78], including Savage[9], PredictHaplo[128], aBayesQR[2], QuasiRecomb[166], HaploClique[167], VGA[103], VirA[152,102], SHORAH[188], ViSpA[7], QURE[129] and others[189,151,154,11,174]. Even though these algorithms proved useful in many applications, accurate and scalable viral haplotyping remains a challenge. In particular, inference of low-frequency viral variants is still problematic, while many computational tools designed for the previous generation of sequencing platforms have severe scalability problems when applied to datasets produced by state-of-the-art technologies.

Previously, several tools such as V-phaser[95], V-phaser2[184] and CoVaMa[139] exploited linkage of mutations for single nucleotide variant (SNV) calling rather than haplotype assembly, but they do not accommodate sequencing errors when deciding whether two variants are linked. These tools are also unable to detect the frequency of mutations above sequencing error rates[169]. The 2SNV algorithm[5] accommodates errors in links and was the first such tool to be able to correctly detect haplotypes with a frequency below the sequencing error rate.

We propose a novel method that can accurately identify minority haplotypes from NGS reads consisting of three steps. First, we extract pairs of statistically linked mutations. Second, we find maximal sets of pairwise linked mutations (cliques) where each clique corresponds to a set of mutations in a minority haplotype. Finally, we assign each read to the closest clique, and for each clique, we form a haplotype as a consensus of reads assigned to it.

All haplotyping tools require solid and convincing validation benchmarks[104,112]. The true viral variants and their distribution are only known for simulated data[46], but sequencing errors, variation of coverage depth, PCR bias, and systematic noise are difficult to simulate (see e.g.,[60]). Therefore experimental sequencing benchmarks that provide an adequate evaluation of haplotyping tools are necessary.

By now, there are only two experimental sequencing benchmarks – (i) Illumina sequencing reads consisting of a mixture of five HIV-1 strains (HIV5exp, see Table 1)[54] and (ii) PacBio sequencing reads from a sample consisting of ten IAV viral variants (IAV10exp, see Table 1)[5]. In the HIV5exp, five different HIV-1 strains each having 20% frequency were prepared to mimic an intra-host viral population. Unfortunately, this benchmark is not realistic enough since the observed intra-host viral populations consist of variants that are much closer to each other than different strains and contain both frequent and rare variants[190]. The IAV10exp benchmark significantly better mimics the intra-host viral population since its variants are very similar to each other and the variant frequencies are realistically non-uniform. Thus, similar to the IAV10exp benchmark, it would be beneficial to develop Illumina benchmarks which adequately imitate intra-host viral populations containing closely related minority variants.

To validate our method's performance, we have introduced two novel in vitro sequencing HIV-1 benchmarks, which consist of Illumina MiSeq experiments on haplotype mixtures based on the mutation profile from an existing patient.

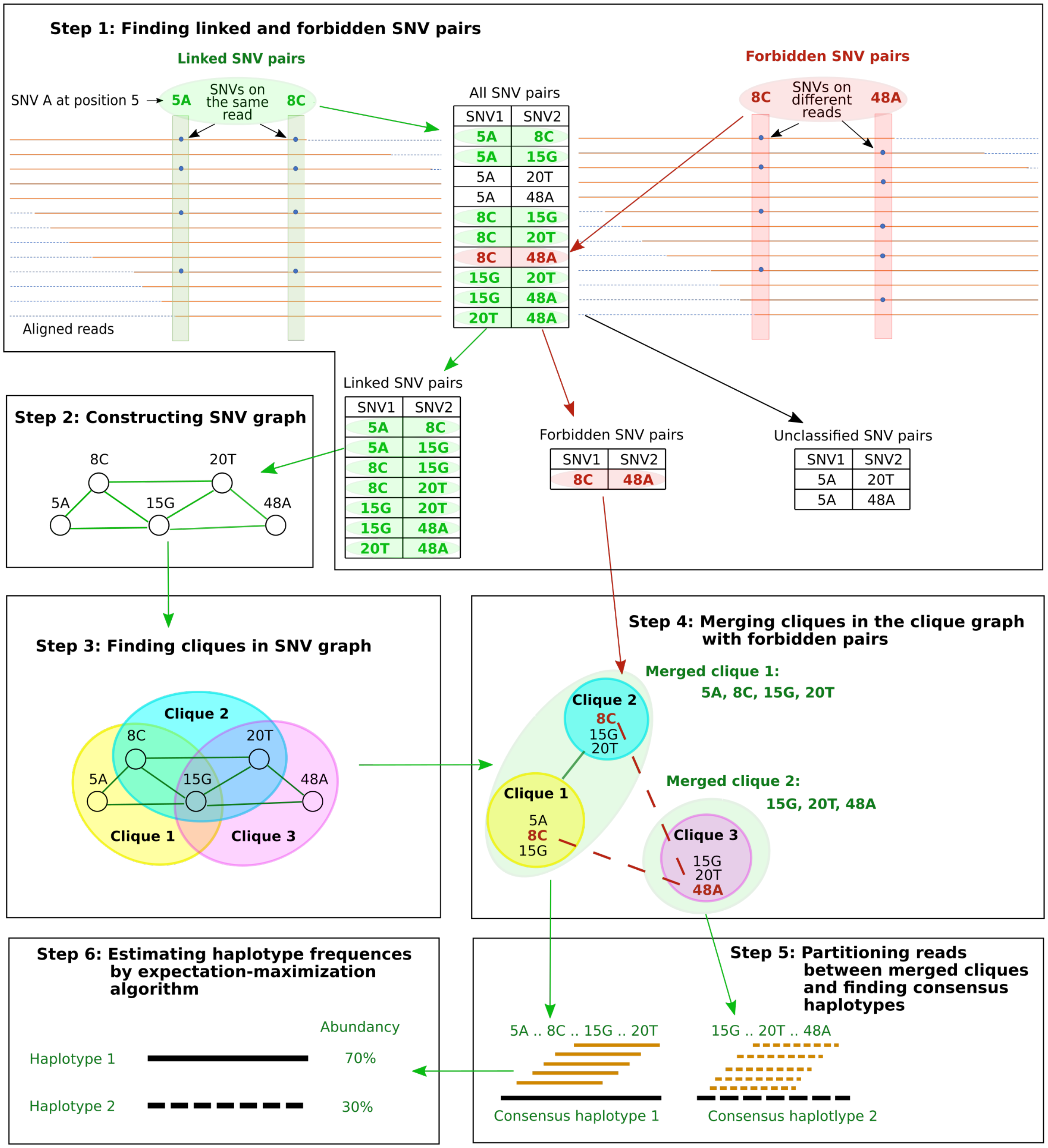
Finally, there is a essential gap in existing quality measures of intra-host viral population assembly. Up-to-date, instead of *populations* (i.e. haplotypes with their frequencies), only *sets* of reconstructed and the ground truth haplotypes are compared[128]. Here we propose to measure differences between haplotype populations using Matching Error and the Earth Mover's Distance which account for both the distances between haplotypes and their frequencies.

3.2 Materials and Methods

3.2.1 CliqueSNV algorithm idea

A schematic diagram of the CliqueSNV algorithm is shown in Figure 3.1. The algorithm takes aligned reads as input and infers haplotype sequences with their frequencies as output. The method consists of six steps:

- Step 1 uses aligned reads to build the consensus sequence and identifies all SNVs. Then



All SNV pairs

SNV1	SNV2
5A	8C
5A	15G
5A	20T
5A	48A
8C	15G
8C	20T
8C	48A
15G	20T
15G	48A
20T	48A

Linked SNV pairs

SNV1	SNV2
5A	8C
5A	15G
8C	15G
8C	20T
15G	20T
15G	48A
20T	48A

Forbidden SNV pairs

SNV1	SNV2
8C	48A

Unclassified SNV pairs

SNV1	SNV2
5A	20T
5A	48A

Figure 3.1 Schematic representation of the CliqueSNV algorithm. Where SNV is single nucleotide variation.

all pairs of SNVs are tested for dependency and are then divided into three groups: *linked*, *forbidden*, or *unclassified*. Each SNV is represented as a pair (p, n) of its position p and nucleotide value n in the aligned reads. If there are enough reads that have two SNVs (p, n) and (p', n') simultaneously, then they are tested for dependency. If the dependency test is positive and statistically significant (see CliqueSNV algorithm details for more information), then the algorithm classifies these two SNVs as *linked*. Otherwise, these two SNVs are tested for independency. If the independency test is positive and statistically significant (see Detailed description for details), then these two SNVs are classified as a *forbidden* pair.

- In Step 2, we build a graph $G = (V, E)$ with a set of nodes V representing SNVs, and a set of edges E connecting linked SNV pairs.

- Ideally, SNVs of each true minority haplotype form a clique in G. A maximal clique $C \subseteq V$ is a set of nodes such that $(u, v) \in E$ for any $u, v \in C$ and for any $x \notin C$ there is $u \in C$ such that $(x, u) \notin E$. Step 3 finds all *maximal cliques* in G.

- For real sequencing data, the linkage between some SNV pairs may be undetected due to sequencing noise, uneven coverage, or the shortness of the NGS reads. As a result, a single clique corresponding to a haplotype will be split into several overlapping cliques. Step 4 merges such overlapping cliques. In order to avoid merging distinct haplotypes, two cliques are not merged if they contain a forbidden SNV pair.

- Step 5 assigns each read to a merged clique with which it shares the largest number of SNVs. Then CliqueSNV builds a consensus haplotype from all reads assigned to a single merged clique.

- Finally, haplotype frequencies are estimated via an expectation-maximization algorithm in Step 6.

3.2.2 Intra-host viral population sequencing benchmarks

Name	Type	Virus	#haplotypes	Haplotype frequencies	Hamming distance
HIV9exp	experimental	HIV-1	9	0.2-50%	0.22-2.1%
HIV2exp	experimental	HIV-1	2	50-50%	1.2%
HIV5exp	experimental	HIV-1	5	20-20%	2-3.5%
IAV10exp	experimental	IAV	10	0.1-50%	0.1-1.1%
HIV7sim	simulated	HIV-1	7	14.3-14.3%	0.6-3%
IAV10sim	simulated	IAV	10	0.1-50%	0.1-1.1%

Table 3.1 Four experimental and two simulated sequencing datasets of human immunodeficiency virus type 1 (HIV-1) and influenza A virus (IAV). The datasets contain MiSeq and PacBio reads from intra-host viral populations consisting of two to ten variants each with frequencies in the range of 0.1-50%, and Hamming distances between variants in the range of 0.1-3.5%.

We tested the ability of CliqueSNV to assemble haplotype sequences and estimate their frequencies from PacBio and MiSeq reads using four real (experimental) and two simulated datasets from HIV and IAV samples (Table 3.1). Each dataset contains between two to ten haplotypes with frequencies of 0.1 to 50%. The Hamming distances between pairs of variants for each dataset are shown in Appendix A Figure 7.

Experimental datasets:

1–2. *HIV-1 subtype B plasmid mixtures and MiSeq reads (HIV2exp and HIV9exp).* We designed nine *in silico* plasmid constructs comprising a 950-bp region of the HIV-1 subtype B polymerase *(pol)* gene that were then synthesized and cloned into pUCIDT-Amp (Integrated

DNA Technologies, Skokie, IL). Each clone was confirmed by Sanger sequencing. This 950-bp region at the beginning of *pol* contains known protease and reverse transcriptase genes that are monitored for drug-resistant mutations and is monitored with sequence analysis for patient care. Each of these plasmids contains a specific set of point mutations chosen using mutation profiles of patient p7 from a real clinical study[190] to create nine unique synthetic HIV-1 *pol* haplotypes. Different proportions of these plasmids were mixed and then sequenced using an Illumina MiSeq protocol to obtain 2x300-bp reads (see Supplementary Methods). HIV2exp and HIV9exp are mixtures of two and nine variants, respectively.

3. *HIV-1 subtype B mixture and MiSeq reads (HIV5exp).* This dataset consists of Illumina MiSeq 2×250-bp reads with an average read coverage of ˜20,000× obtained from a mixture of five HIV-1 isolates: 89.6, HXB2, JRCSF, NL43, and YU2 available at[54]. Isolates have pairwise Hamming distances in the range from 2-3.5%(27 to 46-bp differences). The original HIV-1 sequence length was 9.3kb, but was reduced to the beginning of *pol* with a length of 1.3kb.

4. *Influenza A mixture and PacBio reads (IAV10exp).* This benchmark contains ten IAV virus clones that were mixed at a frequency of 0.1-50%. The Hamming distances between clones ranged from 0.1-1.1% (2-22–bp differences)[5]. The 2kb-amplicon was sequenced using the PacBio platform yielding a total of 33,558 reads with an average length of 1973 nucleotides.

Simulated datasets:

1. *HIV-1 subtype B mixture and MiSeq reads (HIV7sim).* This benchmark contains simulated

Illumina MiSeq reads with a 10k-coverage of 1-kb *pol* sequences. The reads were simulated from seven equally distributed HIV-1 variants chosen from the NCBI database: AY835778, AY835770, AY835771, AY835777, AY835763, AY835762, and AY835757. The Hamming distances between clones are in the range from 0.6-3.0% (6 to 30-bp differences). We used SimSeq[15] for generating reads.

2. *Influenza A mixture and MiSeq reads (IAV10sim).* This benchmark contains simulated IAV Illumina MiSeq reads with the same IAV haplotypes and their frequencies as for the IAV10exp benchmark. The sequencing of a 2kb-amplicon with 40k coverage with paired Illumina MiSeq reads was simulated by SimSeq[15] with the default sequencing error profile in SimSeq.

3.2.3 Validation metrics for viral population inference

3.2.3.1 Precision and recall

Inference quality is typically measured by precision and recall.

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

where TP is the number of true predicted haplotypes, FP is the number of false predicted haplotypes, and FN is the number of undiscovered haplotypes.

Initially we measured precision and recall strictly by treating a predicted haplotype with a single mismatch as an FP. Additionally, like in[128] we introduced an acceptance threshold, which is the number of mismatches permitted for a predicted haplotype to count as a TP.

3.2.3.2 Matching errors between populations

However, precision and recall do not take into account (i) distances between true and inferred viral variants as well as (ii) the frequencies of the true and inferred viral variants. Instead, we chose to use analogues of precision and recall defined for populations as follows.

Let $T = \{(t, f_t)\}$, be the true haplotype population, where f_t is the frequency of the true haplotype t, $\sum_{t\in T} f_t = 1$. Similarly, let $P = \{(p, f_p)\}$, be the reconstructed haplotype population, where f_p is the frequency of the reconstructed haplotype p, $\sum_{p\in P} f_p = 1$. Let d_{pt} be the distance between haplotypes p and t. Thus, instead of precision, we used the *matching error* $E_{T\to P}$ which measures how well each reconstructed haplotype $p \in P$ weighted by its frequency is matched by the closest true haplotype.

$$E_{T\to P} = \sum_{p\in P} f_p \min_{t\in T} d_{pt}$$

Indeed, precision increases while $E_{T\to P}$ decreases and reaches 100% when $E_{T\to P} = 0$. Similarly, instead of recall, we propose to use the *matching error* $E_{T\leftarrow P}$ which measures how well each true haplotype $t \in T$ weighted by its frequency is matched by the closest reconstructed haplotype.[52]

$$E_{T\leftarrow P} = \sum_{t\in T} f_t \min_{p\in P} d_{pt}$$

Note that recall increases while $E_{T\leftarrow P}$ decreases and reaches 100% when $E_{T\leftarrow P} = 0$.

3.2.3.3 Earth mover's distance (EMD) between populations

The matching errors described above match haplotypes of true and reconstructed populations but do not match their frequencies. In order to simultaneously match haplotype sequences and their frequencies, we allowed for a fractional matching when portions of a single haplotype p of population P are matched to portions of possibly several haplotypes of T and *vice versa*. Thus, we separated f_p into f_{pt}'s each denoting portion of p matched to t such that $f_p = \sum_{t\in T} f_{pt}$, $f_{pt} \geq 0$. Symmetrically, f_t's are also separated into f_{pt}'s, i.e, $\sum_{p\in P} f_{pt} = f_t$. Finally, we chose f_{pt}'s minimizing the total error of matching T to P which is also known as Wasserstein metric or the EMD between T and P[87,101].

$$EMD(T,P) = \min_{f_{pt}>0} \sum_{t\in T} \sum_{p\in P} f_{pt} d_{pt}$$

$$\text{s.t.} \sum_{t\in T} f_{pt} = f_p, \text{ and} \sum_{p\in P} f_{pt} = f_t$$

EMD is efficiently computed as an instance of the transportation problem using network flows.

EMDs can vary a lot over different benchmarks since they may have different complexities, which depends on the number of true variants, the frequency distribution, the similarity between haplotypes, sequencing depth, sequencing error rate, and many other parameters. Hence, we measured the complexity of a benchmark as the EMD between the true population and a population consisting of a single consensus haplotype[183].

3.2.4 CliqueSNV algorithm details

Data input for CliqueSNV consists of PacBio or Illumina reads from an intra-host viral population aligned to a reference genome. Output is the set of inferred viral variant RNA sequences with their frequencies. The formal high-level pseudocode of the CliqueSNV algorithm is described in the supplementary materials. Below we describe in detail the six major steps of CliqueSNV that are schematically presented in Figure 3.1.

Step 1: Finding linked and forbidden SNV pairs. At a given genomic position I, the most frequent nucleotide is referred to as a *major variant* and is denoted 1. Let us fix one of the less frequent nucleotide (referred to as a *minor variant*) and denote it 2. A pair of variants at two distinct genomic positions I and J is referred to as a 2-haplotype. There are four 2-haplotypes with major and minor variants at I and J: (11),(12),(21), and (22). Let $O_{11}, O_{12}, O_{21}, O_{22}$ be the observed counts of 2-haplotypes in the reads covering I and J. In this step, CliqueSNV tries to decide whether the O_{22} reads are sequencing errors or they are produced by an existing haplotype containing the 2-haplotype (22).

The pairs of minor variants (referred to as SNV pairs) are classified into three categories: linked, forbidden, and unclassified. An SNV pair is *linked* if it is highly probable that there exists a haplotype containing both minor variants. On the contrary, an SNV pair is *forbidden* if it is extremely unlikely that the corresponding minor variants belong to the same haplotype. All other SNV pairs are referred to as *unclassified*.

Assuming that errors are random, it has been proven in[5] that if the 2-haplotype (22) does not exist, then the expected number of reads E_{22} containing the 2-haplotype (22) should not exceed

$$E_{22} \leq \frac{E_{21} \cdot E_{12}}{E_{11}} \tag{3.1}$$

where E_{21}, E_{12}, and E_{11} are the expected numbers of reads containing the 2-haplotypes (21), (12) and (11), respectively. To determine if a pair of SNVs (the minor variants in positions I and J) are linked, we need to estimate the probability that the observed counts of 2-haplotypes $O_{11}, O_{12}, O_{21}, O_{22}$ are produced by 2-haplotype counts satisfying equation 3.1.

Let n be the total number of reads covering both positions I and J. Then

$$p = \frac{O_{21} \cdot O_{12}}{O_{11} \cdot n} \tag{3.2}$$

is the probability of observing O_{22} reads with the both minor variants given that the variant (22) does not exist.

The 2-haplotype (22) exists with high probability $1 - P$ and the corresponding pair of SNVs is linked if the value of p satisfies the following inequality[5]

$$1 - \sum_{i=0}^{O_{22}-1} \binom{n}{i} p^i (1-p)^{n-i} \leq \frac{P}{\binom{L}{2}} \tag{3.3}$$

where P is the user-defined P-value (by default $P = 0.01$) and dividing by $\binom{L}{2}$ is the Bonferroni correction for multiple testing.

Pairs of SNVs passing this linkage test are classified as a *linked* SNV pairs. For every other pair of SNVs, we check whether they can be classified as a *forbidden* SNV pair, i.e., whether the probability of observing at most 0_{22} reads is low enough (< 0.05) given that the variant (22) has frequency $T_{22} \geq t$ (by default $t = 0.001$).

$$P(x \leq O_{22} | T_{22} \geq t) \leq \sum_{i=0}^{O_{22}} \binom{n}{i} t^i (1-t)^{n-i} \tag{3.4}$$

Step 2: Constructing the SNV graph. The SNV graph $G = (V, E)$ consists of vertices corresponding to minor variants and edges corresponding to linked pairs of minor variants from different positions. If the intra-host population consists of very similar haplotypes, then in the graph G, the number of non-isolated vertices is very small which makes the number of edges very small as well. Indeed, the PacBio dataset for IAV encompassing $2,500$ positions is split into 10,000 vertices, while the SNV graph contains only 700 edges, and, similarly, the simulated Illumina read dataset for the same haplotypes contains only 368 edges.

Note that the isolated minor variants correspond to genotyping errors unless they have a significant frequency. This fact allows us to estimate the number of errors per read, assuming that all isolated SNVs are errors. As expected, the distribution of the PacBio reads has a heavy tail (see Appendix A Figure 10), which implies that most reads are (almost) error free, while a small number of heavy-tail reads accumulate most of the errors. Our analysis allows the identification of such reads, which can then be filtered out. By default, we filter out $\approx 10\%$ of PacBio reads, but we do not filter out any Illumina reads. The SNV graph is then constructed for the reduced set of reads. Such filtering allows the reduction of systematic errors and refines the SNV graph significantly.

Step 3: Finding cliques in the SNV graph G. Although the MAX CLIQUE is a well-known NP-complete problem and there may be an exponential number of maximal cliques in G, a standard

Bron-Kerbosch algorithm requires little computational time since G is very sparse[23].

Step 4: Merging cliques in the clique graph C_G. The clique graph $C_G = (C, F, L)$ consists of vertices corresponding to cliques in the SNV graph G and two sets of edges F and L. A *forbidding edge* $(p, q) \in F$ connects two cliques p and q with at least one forbidden pair of minor variants from p and q respectively. A *linking edge* $(p, q) \in L$ connects two cliques p and q, $(p, q) \notin F$, with at least one linked pair of minor variants from p and q respectively. Any true haplotype corresponds to a maximal $(L \setminus F)$-connected subgraph H of C_G which is connected with edges from L and does not contain any edge from F (see Fig. 3.1 (4)).

Unfortunately, even deciding whether there is a L-path between p and q avoiding forbidding edges is known to be NP-hard[81]. We find all subgraphs H as follows (see Appendix A Figure 11): (i) connect all pairs of vertices except connected with forbidding edges, (ii) find all maximal super-cliques in the resulted graph $C'_G = (C, C^{(2)} - F)$ using[23], (iii) split each super-clique into L-connected components, and (iv) output maximal L-connected components.

Step 5: Partitioning reads between merged cliques and finding consensus haplotypes. Let S be the set of all positions containing at least one minor variant in V. Let q_S be an *major clique* corresponding to a haplotype with all major variants in S. The distance between a read r and a clique q equals the number of variants in q that are different from the corresponding nucleotides in r. Each read r is assigned to the closest clique q (which can possibly be q_S). In case of a tie, we assign r to all closest cliques. In that case the read r will contribute only $1/n$ frequency in consensus calculation, where n is the number of closets cliques. In most cases the number of assigned cliques is 1, although in a case of of IAV, when most clique share the same position a significant portion of reads go to many cliques (see Appendix A Figures 12-14).

Finally, for each clique q, CliqueSNV finds the consensus $v(q)$ of all reads assigned to q. Then $v(q)$ is extended from S to a full-length haplotype by setting all non-S positions to major SNVs.

Step 6: Estimating haplotype frequencies by using the expectation-maximization (EM) algorithm. CliqueSNV estimates the frequencies of the assembled intra-host haplotypes via an expectation-maximization algorithm similar to the one used in IsoEM[116]. Let K be the number of assembled viral variants, and let α be the probability of sequencing error. EM algorithm works as follows:

1. Initialize frequencies of viral variants $f_j^{(0)} \leftarrow \frac{1}{K}$,

   Compute the probability of l_i-long read r_i $i = \overline{1, N}$, being emitted by viral variant $j = \overline{1, K}$,

   $h_{ji} = \prod_{l=1}^{l_i}((1-\alpha)M_{ji,l} + \frac{\alpha}{3}(1 - M_{ji,l}))$,

   where $M_{ji,l}$ - indicator if i-th read coincides with j-th viral variant in the position l

2. (Expectation) Update the amount of read r_i emitted by the jth viral variant $p_{ij} \leftarrow \frac{f_j^{(n-1)}h_{ji}}{\sum_{u=1}^{k} f_u^{(n-1)}h_{ui}}$

3. (Maximization) Update the frequency of the jth viral variant $f_j^{(n)} \leftarrow \frac{\sum_{i=1}^{N} p_{ij}}{\sum_{u=1}^{k}\sum_{i=1}^{N} p_{iu}}$

4. if $||f_j^{(n-1)} - f_j^{(n)}|| > \varepsilon$, then $n \leftarrow n + 1$ and go to step 2

5. Output estimated frequencies $f_j^{(n)}$

3.3 Results

3.3.1 Performance of haplotyping methods

We compared CliqueSNV to the 2SNV, PredictHaplo, and aBayesQR haplotyping methods. Since CliqueSNV, PredictHaplo and aBayesQR use Illumina reads, we compared them using the HIV9exp, HIV2exp, HIV5exp, HIV7sim, and IAV10sim datasets. Since CliqueSNV, 2SNV, and PredictHaplo can also use PacBio reads, we compared them using the IAV10exp dataset. We also used consensus sequences in the comparisons[183] because of its simplicity and to evaluate sequences most similar to those generated by the Sanger sequencing method[77].

Benchmark	CliqueSNV		aBayesQR		PredictHaplo	
	Precision	Recall	Precision	Recall	Precision	Recall
HIV9exp	**0.60**	**0.33**	0.00	0.00	0.00	0.00
HIV2exp	**0.66**	**1.00**	0.11	0.50	0.50	0.50
HIV5exp	0.18	**0.40**	0.00	0.00	**0.33**	0.20
HIV7sim	**1.00**	**0.71**	**1.00**	0.42	0.45	**0.71**
IAV10sim	**0.75**	**0.30**	0.11	0.10	0.33	0.10

(a)

Benchmark	CliqueSNV		2SNV		PredictHaplo	
	Precision	Recall	Precision	Recall	Precision	Recall
IAV10exp	**1.00**	**1.00**	0.82	0.90	0.70	0.70

(b)

Table 3.2 Prediction statistics of haplotype reconstruction methods using experimental and simulated (a) MiSeq and (b) PacBio datasets. The precision and recall was evaluated stringently such that if a predicted haplotype has at least one mismatch to its closest answer, then that haplotype is scored as a false positive.

The precision and recall of haplotype discovery for each method is provided in Table 3.2. CliqueSNV had the best precision and recall for five of the six datasets. For the HIV5exp dataset, PredictHaplo was more conservative and predicted less false positive variants (better precision) than CliqueSNV.

Following study[128], we also showed how precision and recall grew with the reduction of restriction on mismatches (Fig. 3.2). The number of true predicted haplotypes for CliqueSNV was always greater than that of the other methods on real experimental sequencing benchmarks indicating that CliqueSNV more accurately identified the true haplotypes. The number of falsely

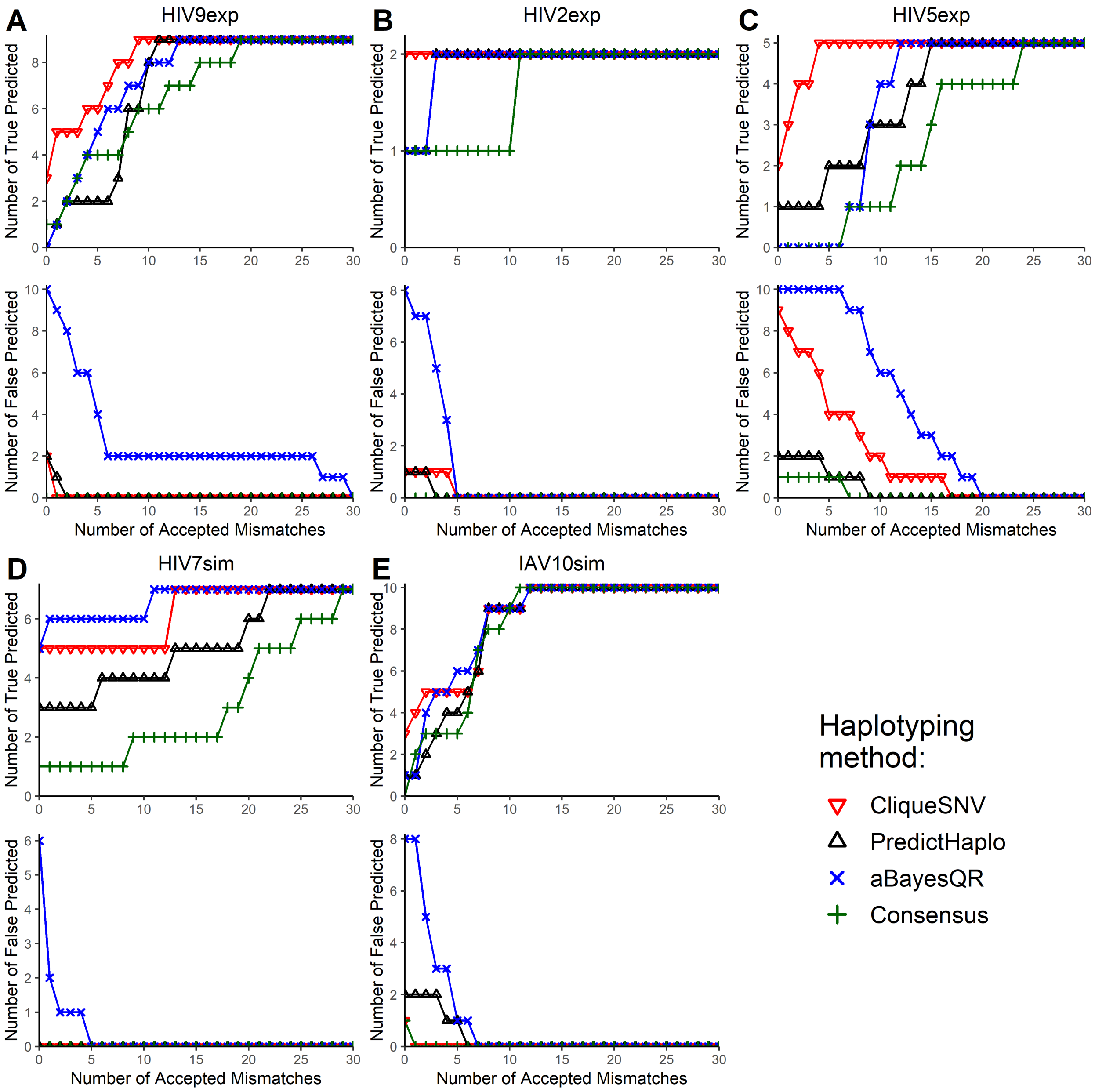


Figure 3.2 The number of true and false predicted haplotypes depending on the number of accepted mismatches for five benchmarks: (A) HIV9exp; (B) HIV2exp; (C) HIV5exp; (D) HIV7sim; (E) IAV10sim. Two haplotypes are regarded identical if the Hamming distance between them is at most the number of accepted mismatches.

predicted haplotypes for CliqueSNV was always lower than those for aBayesQR, but similar to those predicted by PredictHaplo on four out of five datasets indicating that both CliqueSNV and PredictHaplo had the best precision with MiSeq datasets.

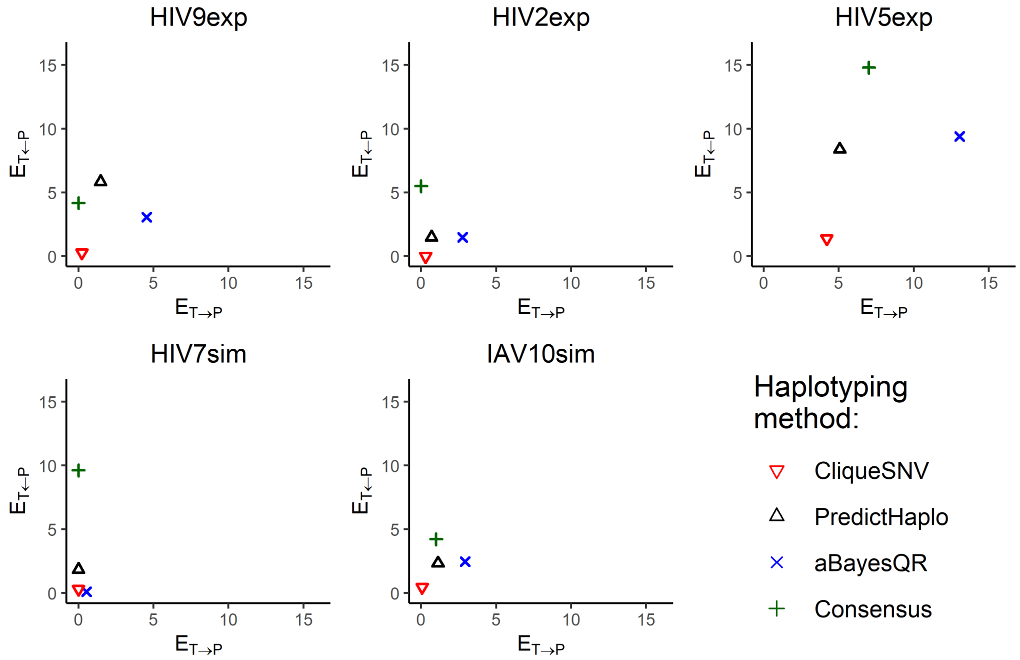


Figure 3.3 Matching distances $E_{T\leftarrow P}$ and $E_{T\rightarrow P}$ between the true haplotype population T and the reconstructed haplotype population P for five benchmarks.

Matching distance analysis showed that matching distances $E_{T\leftarrow P}$ and $E_{T\rightarrow P}$ are better for CliqueSNV than for both PredictHaplo and aBayesQR on four out of five MiSeq datasets (Fig. 3.3). For HIV7sim, $E_{T\leftarrow P}$ for aBayesQR was slightly better than for CliqueSNV. Using HIV9exp, HIV2exp, HIV7sim, and IAV10sim datasets, the $E_{T\leftarrow P}$ and $E_{T\rightarrow P}$ for CliqueSNV were very close to zero indicating that the predictions were almost perfect. Since $E_{T\leftarrow P}$ and $E_{T\rightarrow P}$ correlate with precision and recall, matching distance analysis indicates that CliqueSNV had a better precision, and significantly outperformed both PredictHaplo and aBayesQR. Since aBayesQR had a higher $E_{T \to P}$ on MiSeq datasets, it is more likely to make more false predictions. Notably, on the HIV7sim dataset, aBayesQR outperformed both CliqueSNV and PredictHaplo by $E_{T \leftarrow P}$.

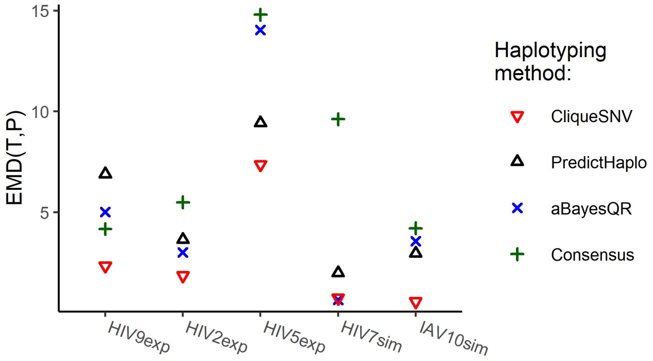


Figure 3.4 Earth Movers' Distance (EMD) between true and reconstructed haplotype populations for five benchmarks.

Benchmark	Consensus	CliqueSNV		aBayesQR		PredictHaplo	
	EMD	EMD	Impr.	EMD	Impr.	EMD	Impr.
HIV9exp	4.18	**2.35**	**1.78**	5.02	0.83	6.90	0.61
HIV2exp	5.50	**1.87**	**2.94**	3.02	1.82	3.65	1.51
HIV5exp	14.80	**7.37**	**2.01**	14.05	1.05	9.43	1.57
HIV7sim	9.63	0.76	12.72	**0.67**	**14.4**	2.00	4.80
IAV10sim	4.22	**0.59**	**7.2**	3.57	1.18	2.97	1.42

(a)

Benchmark	Consensus	CliqueSNV		2SNV		PredictHaplo	
	EMD	EMD	Impr.	EMD	Impr.	EMD	Impr.
IAV10exp	4.22	**0.22**	**19.18**	0.23	18.35	0.38	11.12

(b)

Table 3.3 Earth Movers' Distance from predicted haplotypes to the true haplotype population and haplotyping method improvement. Four haplotyping methods(aBayesQR, CliqueSNV, Consensus, PredictHaplo) are benchmarked using five MiSeq (a) and one PacBio datasets (b). The column Impr. (improvement) shows how much better is prediction of haplotyping method over inferred consensus, and it is calculated as $\frac{EMD_m}{EMD_c}$, where EMD_c is an EMD for consensus, and EMD_m is an EMD for method.

The EMD between the predicted and true haplotype populations for all five MiSeq datasets are shown in Figure 3.4. The exact EMD values are provided in Table 3.3. CliqueSNV provided the lowest (the best) EMD across all tools on four out of five MiSeq benchmarks. For the simulated and PacBio datasets, CliqueSNV had almost a zero EMD indicating a low error in predictions. PredictHaplo had a lower EMD than aBayesQR on four out of five MiSeq datasets. aBayesQR has almost a zero EMD with the HIV7sim dataset and outperformed CliqueSNV, while using the HIV5exp dataset, aBayesQR performed poorer than other methods.

Next, CliqueSNV, 2SNV, and PredictHaplo were compared using the IAV10exp benchmark dataset (see Appendix A Table 2). CliqueSNV correctly recovered all ten true variants, including the haplotype with frequencies significantly below the sequencing error rate. 2SNV recovered nine true variants but found one false positive. PredictHaplo recovered only seven true variants and falsely predicted three variants. To further explore the precision of these three methods with the IAV10exp data, we simulated low-coverage datasets by randomly subsampling $n = 16K, 8K, 4K$ reads from the original data. For each dataset, CliqueSNV found at least one true variant more than both 2SNV and PredictHaplo.

3.3.2 Runtime comparison

To compare the computational run time of each method, we used the same PC (Intel(R) Xeon(R) CPU X5550 2.67GHz x2 8 cores per CPU, DIMM DDR3 1,333 MHz RAM 4Gb x12) with the CentOS 6.4 operating system. The runtime of CliqueSNV is sublinear with respect to the number of reads while the runtime of PredictHaplo and 2SNV exhibit super-linear growth. For the 33k IAV10sim reads the CliqueSNV analysis took 21 seconds, while PredictHaplo and 2SNV took around 30 minutes. The runtime of CliqueSNV is quadratic with respect to the number of SNVs rather than by the length of the sequencing region (Appendix A Fig. 8).

We also generated five HIV-1 variants within 1% Hamming distance from each other, which is the estimated genetic distance between related HIV variants from the same person[173]. Then we simulated 1M Illumina reads for sequence regions of length 566, 1132, 2263 and 9181 nucleotides for which CliqueSNV required 37, 144, 227, and 614 seconds, respectively, for analyzing these datasets (Appendix A Fig. 9). For the HIV2exp benchmark, aBayesQR, PredictHaplo, and CliqueSNV required over ten hours, 24 minutes, and only 79 seconds, respectively.

3.4 Discussion

Assembly of haplotype populations from noisy NGS data is one of the most challenging problems of computational genomics. High-throughput sequencing technologies, such as Illumina MiSeq and HiSeq, provide deep sequence coverage that allows discovery of rare, clinically relevant haplotypes. However, the short reads generated by the Illumina technology require assembly that is complicated by sequencing errors, an unknown number of haplotypes in a sample, and the genetic similarity of haplotypes within a sample. Furthermore, the frequency of sequencing errors in Illumina reads is comparable to the frequencies of true minor mutations[154]. The recent development of single-molecule sequencing platforms such as PacBio produce reads that are sufficiently long to span entire genes or small viral genomes. Nonetheless, the error rate of single-molecule sequencing is exceptionally high reaching $13 - 14\%$[132], which hampers PacBio sequencing to detect and assemble rare viral variants.

We developed CliqueSNV, a new reference-based assembly method for reconstruction of rare genetically-related viral variants such as those observed during infection with rapidly evolving RNA viruses like HIV, HCV and IAV. We demonstrated that CliqueSNV infers accurate haplotyping in the presence of high sequencing error rates and is also suitable for both single-molecule and short-read sequencing. In contrast to other haplotyping methods, CliqueSNV infers viral haplotypes by detection of clusters of statistically linked SNVs rather than through assembly of overlapping reads used with methods such as Savage[9].

Applied to the novel in vitro sequencing HIV-1 benchmark, CliqueSNV correctly reconstructed 87% of the intra-host haplotype population. At the same time, other state-of-the-art tools were not able to recover even a single haplotype without errors. Additionally, we have used the only previously known and commonly used in vitro benchmark[53] and simulated datasets to evaluate the accuracy of existing haplotyping methods. In contrast to the existing methods, CliqueSNV was able to detect minority haplotypes at a low 0.1% frequency and distinguish minority haplotypes differently in only two base pairs.

Although very accurate and fast, CliqueSNV has some limitations. Unlike Savage[9], CliqueSNV is not a *de novo* assembly tool and requires a reference viral genome. This obstacle could easily be addressed by using Vicuna[183] or other analogous tools to first assemble a consensus sequence from the NGS reads, which can then be used as a reference. Another limitation is for variants that differ only by isolated SNVs separated by long conserved genomic regions longer than the read length which may not be accurately inferred by CliqueSNV. While such situations usually do not occur for viruses, where mutations are typically densely concentrated in different genomic regions, we plan to address this limitation in the next version of CliqueSNV.

The ability to accurately infer the structure of intra-host viral populations makes CliqueSNV applicable for studying viral evolution, transmission and examining the genomic compositions of RNA viruses. In addition, we envision that the application of our method could be extended to other highly heterogeneous genomic populations, such as metagenomes, immune repertoires, and cancer cell genes.

CHAPTER 4

INFERENCE OF CLONAL SELECTION IN CANCER POPULATIONS USING SINGLE-CELL SEQUENCING DATA

4.1 Introduction

Cancer is responsible for more than $600,000$ deaths in the USA annually[150]. It is a disease driven by the uncontrolled growth of cancer cells having series of somatic mutations acquired during the tumor evolution. Cancer clones form heterogeneous populations, which include multiple subpopulations constantly evolving to compete for resources, metastasize, escape immune system and therapy[40,82,59,186]. Clonal heterogeneity plays key role in tumor progression[111], and has important implications for diagnostics and therapy, since rare drug resistant variants could become dominant and lead to relapse in the patient[40,85]. Therefore cancer is now viewed as a dynamic evolutionary process defined by complex interactions between clonal variants, which include both competition and cooperation[59,186,16].

Recent advances in sequencing technologies promise to have a profound effect on oncological research. Study of genomic data for different tumors produced by next-generation sequencing (NGS) led to progress in understanding evolutionary mechanisms of cancer[186,59,82]. Most of cancer data have been obtained using bulk sequencing, which produces admixed populations of cells. Recently, the most promising technological breakthrough was the advent of *single cell sequencing* (scSeq), which allows to access cancer clone populations at the finest possible resolution. scSeq protocols combined with NGS allow to analyze genomes of individual cells, thus providing deeper insight into biological mechanisms of tumor progression.

The cornerstone of such analysis is an estimation of parameters defining the evolution of heterogeneous clonal populations. Currently there is no scientific consensus about the rules guiding the evolution of cancer cells[34,162,177,119], with multiple competing theories being advanced by different researchers. The open questions include the rules of evolution (neutral, linear, branching or punctuated), ways of interaction between clonal variants (competition or cooperation) and the role of epistasis (non-linear interaction of SNVs or genes). These questions could be addressed by estimation of evolutionary parameters for cancer lineages from NGS data[177,162].

One of the most important evolutionary parameters is the collection of replicative fitnesses of individual genomic variants, commonly termed *fitness landscape* in evolutionary biology[50]. Several computational tools have been proposed for *in vitro* estimation of fitness landscapes[145,94,65,47]. However, *in vitro* studies are cost- and labor-intensive, consider organisms removed from their natural environments and does not allow to capture all population genetic diversity[146]. One of the possible ways to infer fitness landscape *in vivo* is to analyze follow-up samples taken from a patient at multiple time points and compute fitnesses directly by measuring changes of frequencies of genomic variants over time. However, follow-up samples are very scarce, and the overwhelming majority of data represent individual samples.

Quantification of clonal selection from individual samples is computationally challenging, but extremely important for understanding mechanisms of cancer progression[177,162]. In particular, recent findings on structures of fitness landscapes of cancer from bulk sequencing data[176] initiated a lively scientific discussion published in several papers[162,119,177]. It can be anticipated that single cell sequencing data will be able to shed light into this important problem. It is known that relative abundances of genomic variants alone are not indicative of variant fitnesses[146]. Existing methods for inference of fitnesses from single samples utilize more sophisticated approaches, but have various limitations including reliance on the assumption that the population is in equilibrium state, or disregard of population heterogeneity and variability of fitness landscapes, or customization to bulk sequencing data[146,35,177].

Contributions. We propose a computational method SCIFIL (Single Cell Inference of FItness Landscape) for *in vivo* inference of clonal selection and estimate of fitness landscapes of heterogeneous cancer clone populations from single cell sequencing data. SCIFIL estimates fitnesses of clonal variants rather than alleles, and does not assume allele independence which allows to take into account the effects of epistasis. Instead of assuming that sampled populations are in the equilibrium state, our method estimates fitnesses of individual clone types using a maximum likelihood approach. We demonstrate that the proposed method allows for accurate inference of fitness landscapes and quantification of clonal selection. We conclude by applying SCIFIL to real tumor data.

4.2 Methods

We propose a maximum likelihood approach, which estimates fitnesses of individual clonal variants by fitting into the tumor phylogeny an evolutionary model with the parameters explaining the observed data with the highest probability. We first establish the ordinary differential equations (ODE) model for the tumor evolutionary dynamics, and define the likelihood of the observed data given the model parameters. We conclude with finding fitnesses maximizing the likelihood by

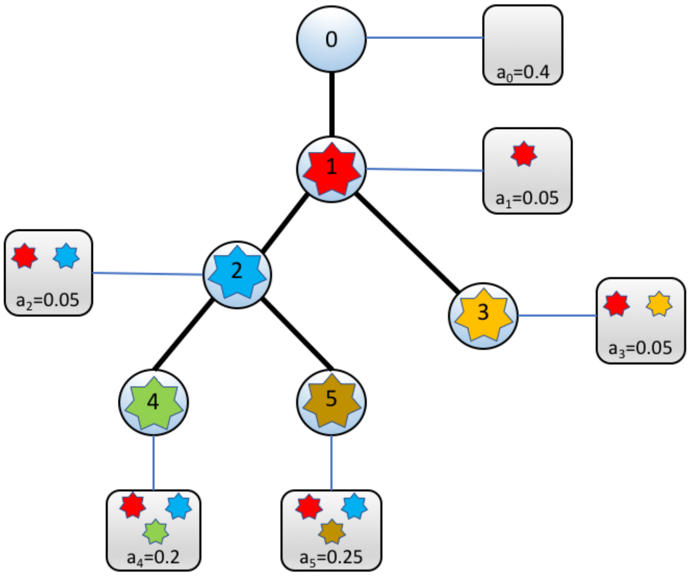


Figure 4.1 Mutation tree

reducing the problem to finding the most likely mutation order and applying branch-and-bound search to solve that problem.

Traditionally, evolutionary histories are represented using binary phylogenetic trees. Following[69], we use an alternative representation of an evolutionary history of a tumor using a *mutation tree*. The internal nodes of a mutation tree represent mutations, leafs represent single cells, internal nodes are connected according to their order of appearance during the tumor evolution and the mutation profile of each cell equals the set of mutations on its path to the root (Fig. 4.1). In addition we accumulate all leafs attached to the same internal node into a single leaf with an abundance representing a particular clone. For simplicity we assume that there is a leaf attached to every internal node, with some leafs having an abundance 0 (or rather a small number $\delta << 1$). Generally we do not need to employ the infinite site assumption, i.e. repeats of mutations are allowed provided that mutation profiles of all clones in a tree are unique. It agrees with recent findings[83]. A mutation tree can be constructed using currently available tools, such as SCITE[69], infSCITE[83] or SiFit[187].

Formally, we consider the following algorithmic problem. Given are:

- mutation tree T with $n+1$ leafs corresponding to clonal variants. We assume that internal nodes of T are labeled $0, 1, ..., n$ and the ith clone is attached to the node i. The root of T correspond to the mutation 0, which represent absence of somatic mutations or healthy tissue.
- observed relative abundances $\mathcal{A} = (a_0, ..., a_n)$ of clones.
- Mean cancer cells mutation rate θ. This is a well-studied parameter with estimations provided by prior studies[64].

The goal is to find fitnesses $\mathcal{F} = (f_0, ..., f_n)$ maximizing the likelihood

$$p(\mathcal{A}|T, \mathcal{F}, \theta) \tag{4.1}$$

This section is organized as follows. First we introduce our evolutionary model of choice and the definition of the probability (4.1). Next, we describe how the likelihood is modified to transform the maximum likelihood problem (4.1) into a discrete optimization problem. Finally, we describe the method of estimation of fitnesses $\mathcal{F}$ maximizing (4.1).

Evolutionary model. We consider tumor evolution as a branching process described by the mutation tree T. Let $V(T)$, $V_I(T)$ and $E(T)$ be the node set, the internal node set and an the arc set of

T, respectively. Let also p_i denote the parent of a node $i \in V_I(T)$. We assume that nodes $V_I(T)$ represent mutation events, with jth event occurring at rate θ_j. The mutation event corresponding to a node i happens at time t_i; at the event the clonal variant corresponding to the parent node p_i gives birth to a variant i. The dynamics of the cancer clone population is described by the *piecewise continuous* function $x = (x_0, ..., x_n)$, where $x_i = x_i(t)$ is the relative abundance of the ith clonal variant. The discontinuity points of x correspond to mutation events. Let r, i, j be 3 consecutive mutation events with times $t_r < t_i < t_j$, and $x_k^{(i)}$ be the restriction of x_i to the interval $[t_i, t_j]$. Between mutation events i and j clonal frequencies $x_k^{(i)}$ follow the system of ODEs[121]:

$$\frac{d}{dt}x_k^{(i)} = f_k x_k^{(i)} - x_k^{(i)} \sum_{l=0}^{n} f_l x_l^{(i)}, \quad k = 0, ..., n \tag{4.2}$$

with initial conditions

$$x_k^{(i)}(t_i) = \begin{cases} \varepsilon x_{p_i}^{(r)}(t_i), \text{ if } k = i \\ (1-\varepsilon) x_k^{(r)}(t_i), \text{ if } k = p_i \\ x_k^{(r)}(t_i), \text{ otherwise.} \end{cases} \tag{4.3}$$

Subtraction of the term $x_k^{(i)} \sum_{l=1}^{n} f_l x_l^{(i)}$ ensures that relative abundances of variants sum up to 1. Initial conditions (4.3) link clone abundances before and after the mutation event i and indicate that at time t_i the clone i is generated by the clone p_i. The parameter $\varepsilon << 1$ is a small number. At time 0, the root clonal variant (healty tissue) gives birth to the first mutation, with the corresponding clones having relative abundances $1 - \varepsilon$ and ε. The model (4.2) is a branching-type variant of the quasispecies model, which is applicable to cancer evolution[178] and agrees or extends several

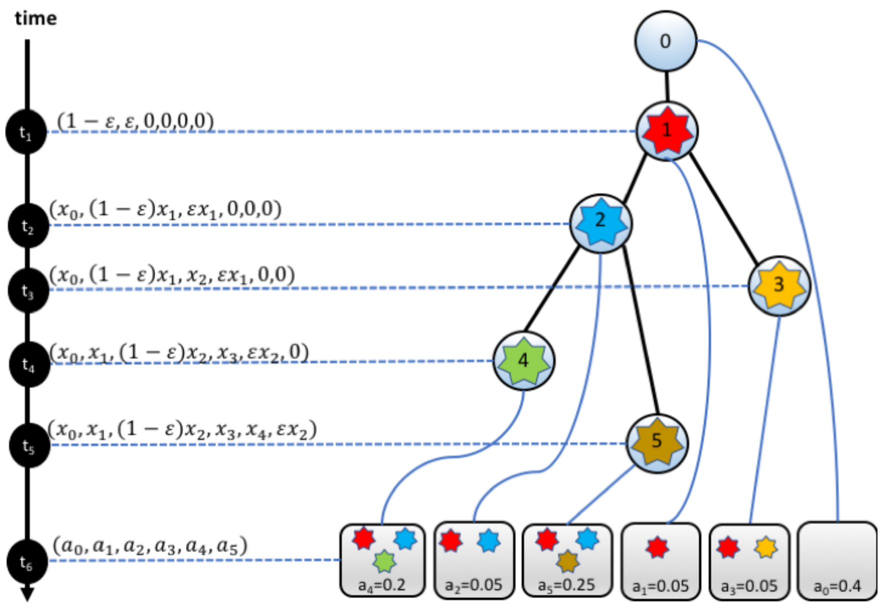

time
t_1
$(1-\varepsilon, \varepsilon, 0,0,0,0)$
t_2
$(x_0, (1-\varepsilon)x_1, \varepsilon x_1, 0,0,0)$
t_3
$(x_0, (1-\varepsilon)x_1, x_2, \varepsilon x_1, 0,0)$
t_4
$(x_0, x_1, (1-\varepsilon)x_2, x_3, \varepsilon x_2, 0)$
t_5
$(x_0, x_1, (1-\varepsilon)x_2, x_3, x_4, \varepsilon x_2)$
t_6
$(a_0, a_1, a_2, a_3, a_4, a_5)$
0
1
2
3
4
5
a_4=0.2
a_2=0.05
a_5=0.25
a_1=0.05
a_3=0.05
a_0=0.4

Figure 4.2 Depiction of the evolutionary model. Tree nodes represent mutation events whose times are marked on the time axis. Leafs represent the sampling event. For each node the distribution of clone abundances after the corresponding event is shown.

classical population genetics concepts[175], including those describing genetic systems governed by mutation and selection[76,113]. It does not include specific assumptions about clonal competition or cooperation.

Likelihood definition. In addition to n mutation events, we consider the $(n+1)$th event representing cell sampling. Suppose that times of mutation events $\Omega = (t_i)_{i=1}^{n+1}$ and mutation rates between events $\Theta = (\theta_i)_{i=1}^{n}$ are given. Let $\sigma = (\sigma_1, ..., \sigma_{n+1})$ be the permutation of events in order of their appearance, i.e. $0 = t_{\sigma_1} < t_{\sigma_2} < ... < t_{\sigma_n} < t_{\sigma_{n+1}}$. The probability of observing abundances $\mathcal{A}$ given T,$\mathcal{F}$, Ω, Θ and θ is defined as the product of probabilities of mutation events and probabilities of observed clone abundances.

The mutation event in the vertex σ_j occurs if 2 conditions are met:

(a) no mutation events have been observed over the time interval $(t_{\sigma_{j-1}}, t_{\sigma_j})$;

(b) at time t_{σ_j} the mutation happened in the clone p_{σ_j} rather than in other clones which exist at that time.

Appearance of mutation is a classical rare event, and therefore we assume that the time intervals between consecutive mutation events i and j follow a Poisson distribution with the mean $\frac{1}{\theta_i}$. Mutation rates are distributed normally with the mean θ and the standard deviation ν. Assuming that mutations are random, the probability of (b) is equal to the frequency $x_{p_{\sigma_j}}(t_{\sigma_j})$ of the clone p_{σ_j} at time t_{σ_j} according to the system (4.2). Finally, we assume that the probability of seeing observed frequencies given model-based frequencies at the sampling time follows a multinomial distribution $\mathcal{M}(a_0, ..., a_n | x_0(t_{n+1}), ..., x_n(t_{n+1}))$. After putting all probabilities together, we have

$$p(\mathcal{A}|T, \mathcal{F}, \Omega, \theta) = \prod_{j=2}^{n+1} Pois(t_{\sigma_j} - t_{\sigma_{j-1}}, \frac{1}{\theta_{j-1}}) \cdot \prod_{j=1}^{n} \mathcal{N}(\theta_j, \theta, \nu) \cdot \\ \cdot \prod_{j=1}^{n} x_{p_j}(t_j) \cdot \mathcal{M}(a_0, ..., a_n | x_0(t_{n+1}), ..., x_n(t_{n+1})) \quad (4.4)$$

Our goal is to find best fitting fitnesses $\mathcal{F}_{ML}$, rates Θ_{ML} and times Ω_{ML} by solving the following maximum likelihood problem:

$$(\mathcal{F}_{ML}, \Theta_{ML}, \Omega_{ML}) = \arg\max_{\mathcal{F}, \Theta, \Omega} p(\mathcal{A}|T, \mathcal{F}, \Omega, \theta) \quad (4.5)$$

The probabilities $\prod_{j=2}^{n+1} Pois(t_{\sigma_j} - t_{\sigma_{j-1}}, \frac{1}{\theta_{j-1}})$, $\prod_{j=1}^{n} \mathcal{N}(\theta_j, \theta, \nu)$,

$\prod_{j=1}^{n} x_{p_{\sigma_j}}(t_{\sigma_j})$ and $\mathcal{M}(a_0, ..., a_n | x_0(t_{n+1}), ..., x_n(t_{n+1}))$ are further referred as *time likelihood*, *rate likelihood*, *mutation likelihood* and *abundance likelihood*, respectively. For the tree shown on Fig. 4.2 it is equally feasible that the mutation 2 appeared before the mutation 3 or vise versa. However, clone 2 later produces mutations 4 and 5, and therefore the mutation likelihood suggests that at that mutation events it had high abundance. This situation is probable if either 2nd mutation appeared earlier or it appeared later but has a high fitness. Time, rate and abundance likelihoods allow to choose between these two alternatives.

Reduction to discrete optimization. The standard way to solve the maximum likelihood problem (4.5) is to optimize $\mathcal{F}$, Θ and Ω jointly using Markov Chain Monte Carlo (MCMC) sampling. However, our experiments have shown that the function (4.1) has too many local optima which makes MCMC search over the continuous space of possible solutions inefficient. Therefore we suggest an alternative heuristic approach, which transforms the problem (4.5) into a discrete optimization problem akin to a scheduling problem. This problem is then solved using a specifically designed combinatorial heuristic search.

Firstly, we assume that all fitnesses are relative with respect to a fitness of a clone 0 which is set to be $f_0 = 1$. By default, this clone corresponds to the normal tissue. For the problem of inference of clonal selection such assumption does not restrict the predictive power. Next, we observe that any assignment of event times Ω defines the order of appearance μ_i for each node $i \in V(T)$ (e.g. on Figure 4.2 $\mu_i = i$ for $i = 1, ..., 5$). This order agrees with the natural vertex order induced by T, i.e. $\mu_i < \mu_j$ whenever i is an ancestor of j. It turned out that conversely any order μ defines times Ω^{μ}, rates Θ^{μ} and fitnesses $\mathcal{F}^{\mu}$ which maximize the partial likelihood

$$\prod_{j=2}^{n+1} Pois(t_{\sigma_j} - t_{\sigma_{j-1}}, \frac{1}{\theta_{j-1}}) \cdot \prod_{j=1}^{n} \mathcal{N}(\theta_j, \theta, \nu) \cdot$$

$$\cdot \mathcal{M}(a_0, ..., a_n | x_0(t_{n+1}), ..., x_n(t_{n+1})) \quad (4.6)$$

More precisely, the following proposition holds.

Proposition 2. *For a given order vector μ, times Ω^μ, rates Θ^μ and fitnesses $\mathcal{F}^\mu$ maximizing (4.6) can be estimated as follows:*

$$\theta_i = \theta, \quad t_i = \frac{\mu_i - 1}{\theta}, i = 1, ..., n, \;\; t_{n+1} = \frac{n}{\theta} \quad (4.7)$$

$$f_i = 1 - \theta \sum_{j \in A_i \setminus \{0\}} \frac{1}{n - \mu_j + 1} \log(\frac{\varepsilon}{1 - \varepsilon} \frac{a_{p_j}}{a_j}), i = 1, ..., n. \quad (4.8)$$

Here A_i is the set of ancestors of a node i (including itself).

Proof. Poisson and Gaussian probabilities achieve maximums at their means, i.e. the rate and time likelihoods are maximal, when for consecutive events i, j we have $\theta_i = \theta$, $t_j - t_i = \frac{1}{\theta}$. This yields the solution (4.7). The multinomial probability $\mathcal{M}(a_0, ..., a_n | x_0(t_{n+1}), ..., x_n(t_{n+1}))$ is maximal when $x_i(t_{n+1}) = a_i$ for all $i \in [n]$. This can be rewritten as

$$\frac{x_i(t_{n+1})}{x_i(t_{n+1}) + x_{p_i}(t_{n+1})} = \frac{a_i}{a_i + a_{p_i}} \text{ for all } i = 1, ..., n. \quad (4.9)$$

Our goal is to find fitnesses $\mathcal{F}$ such that (4.9) holds. We find an approximate solution to this problem by disregarding the discontinuity of the abundances $x = (x_i(t))_{i=0}^{n+1}$. We use the observation that the system (4.2) is invariant with respect to the transition to relative abundances of any pair of clones. Namely, for each clone pair $i, j = 0, ..., n$ dynamics of their relative abundances with respect to each other $y_i = \frac{x_i}{x_i + x_j}$ and $y_j = \frac{x_j}{x_i + x_j}$ is described by the system of ODEs of the same form as (4.2):

$$\begin{aligned} \dot{y_i} &= f_i y_i - y_i(f_i y_i + f_j y_j), \\ \dot{y_j} &= f_j y_j - y_j(f_i y_i + f_j y_j), \end{aligned} \tag{4.10}$$

On the interval $[t_i, t_{n+1}]$ relative abundance $y_i = \frac{x_i}{x_i + x_{p_i}}$ satisfy the system (4.10) with the initial condition $y_i(t_i) = \varepsilon$. After shifting time interval to $[0, t_{n+1} - t_i]$ this system can be linearized and solved in closed form, producing a solution

$$y_i(t) = \frac{\varepsilon e^{f_i t}}{(1 - \varepsilon) e^{f_{p_i} t} + \varepsilon e^{f_i t}} \tag{4.11}$$

After putting the expressions (4.11) into the equations (4.9) with $t = t_{n+1} - t_i$ we get the following system of equations to find fitnesses $\mathcal{F}$:

$$f_{p_i} - f_i = \frac{1}{t_{n+1} - t_i} \log(\frac{\varepsilon}{1 - \varepsilon} \frac{a_{p_i}}{a_i}), i = 1, ..., n; \quad f_0 = 1. \tag{4.12}$$

Solving it with t_i described by (4.7) yields the solution (4.8). □ □

Using Proposition 2, we replace the maximum likelihood problem (4.5) with the following discrete problem: find the ordering μ maximizing the mutation log-likelihood

$$L_\mu = \log(p(\mu)) = \sum_{j=1}^{n} \log(x_{p_j}(t_j)) \tag{4.13}$$

with times Ω^μ and fitnesses $\mathcal{F}^\mu$ described by (4.7),(4.8) subject to the constraint that μ agrees with with the ancestral-descendant order of T.

Finding optimal ordering. The problem (4.13) could be considered as a variant of scheduling problem with precedent constraints and with non-linear cumulative cost function[36]. Here mutations play roles of jobs, ordering of mutations corresponds to scheduling of jobs on a single processor, mutation tree represent job precedence constraints, and the objective (4.13) indicates that the cost of job processing depends on the previously processed jobs. Such problems are usually NP-hard[36]. For small number of mutations, it can be solved by a branch-and-bound search in the space of feasible orderings via backtracking over the mutation tree. In general, we solve it by a heuristic approach combined with the search in the space of feasible sub-orderings of nodes of the mutation tree T. The proposed scheme is described by Algorithm 3. The algorithm starts with the initial tree $T' = T$ and iteratively transforms it into a total order as follows. We call two simple paths of T' *sibling paths*, if they share the starting vertex. We traverse the nodes of T' in a bottom-up direction and merge sibling paths into one path representing optimal sub-order of their nodes with respect to the objective (4.13). The algorithm stops when all nodes form a single path.

Merging of sibling paths P_1 and P_2 is performed by Algorithm 4. We note that feasible orders of paths' nodes bijectively correspond to k-subsets of the set $[k + l]$: for a given k-subset X, a

Algorithm 3 Algorithm for node ordering

```
Let $U$ be the list of nodes of $T$ sorted in inverse order of their discovery by Breadth First Search from the root; $T' = T$;
for $u \in U$ do
    while $u$ has more than 1 child do
        Choose sibling paths $P_1$ and $P_2$ with the start node $u$
        Join $P_1$ and $P_2$ into a single path $P$ using Algorithm 4
        Modify $T'$ by replacing $P_1$ and $P_2$ by $P$
    end while
end for
```

Algorithm 4 Algorithm for path joining

```
Require: Sibling paths $P_1$ and $P_2$
Ensure: is calculated by calling MergePaths($\emptyset$,1)
    MergePaths($Y$,$i$)
                    ▷ $Y$ is the current $k$-subset, $i$ is the next element to be added to it
                    ▷ $\mu_{opt}$ and $opt$ are the current optimal order and its likelihood
if $|Y| = k$ or $i > k + l$ then
    return
end if
$Y_{new} = Y \cup \{i\}$, $\mu' = \mu_{Y_{new}}$
while $\mu'$ is not a total order do
    $w_1 = P_1^Y(1)$, $w_2 = P_2^Y(1)$,$j = |\mu'| + 1$
    $t = \frac{j-1}{\theta}$, $f_{w_1} = f_{p_{w_1}} + \frac{1}{t_{n+1}-t}\log(\frac{\varepsilon}{1-\varepsilon}\frac{a_{p_{w_1}}}{a_{w_1}})$,
    $f_{w_2} = f_{p_{w_2}} + \frac{1}{t_{n+1}-t}\log(\frac{\varepsilon}{1-\varepsilon}\frac{a_{p_{w_2}}}{a_{w_2}})$
    if $f_{w_1} \leq f_{w_2}$ then
        $\mu' = \mu' \cup \{w_1\}$, $P_1^Y = P_1^Y \setminus \{w_1\}$
    else
        $\mu' = \mu' \cup \{w_2\}$, $P_2^Y = P_2^Y \setminus \{w_2\}$
    end if
end while
MergePaths($Y$,$i + 1$)
if $L_{\mu_Y} > opt$ then
    $opt = L_{\mu'}$, $\mu_{opt} = \mu'$
    MergePaths($Y_{new}$,$i + 1$)
end if
```

feasible order μ_X is obtained by placing nodes from $P_1 \setminus \{u\}$ (resp., $P_2 \setminus \{u\}$) at positions from X (resp., $[k+l] \setminus X$) in order of their appearance in P_1 (resp., P_2); inverse is also true. Algorithm 4 recursively generates k-subsets via branching and prune branches, if the corresponding orders are likely to be sub-optimal.

The k-subsets are generated recursively[117] using the property that every k-subset X of $[k+l]$ is either k-subset of the set $[2 : k+l]$ or has the form $X = \{1\} \cup Y$, where Y is a $k-1$-subset of $[2 : k+l]$. Suppose that at a given iteration a partial k'-subset Y, $k' \leq k$, and the corresponding pre-order μ_Y has been constructed. For all nodes v covered by μ_Y we calculate their appearance times t_v and fitnesses f_v using (4.7),(4.8), and abundance distributions $x^v = (x_0(t_v), ..., x_n(t_v))$ from the system (4.2)-(4.3) (in fact, it is not necessary to recalculate all values since some of them has been already calculated at previous iterations). Next, we heuristically extend μ_Y to a total order as described below. If the likelihood of the constructed solution is below the current optimum, then the recursion tree branch of the partial solution Y is pruned. Otherwise, the current optimum is updated and the recursion continues.

Finally, we describe how an order μ_Y is extended (lines 5-14 of Algorithm 4). We consider the subpaths P_1^Y and P_2^Y formed by the nodes of P^1 and P^2 that are not covered by μ_Y. For the first nodes of these subpaths, we calculate their provisional fitnesses under the assumption that each node is added to μ_Y as the next element. The node with the smaller provisional fitness is added to μ_Y. This procedure is repeated until μ_Y covers all nodes. The logic behind this approach is based on the observation that according to (4.2) the frequency of a clone grows while its fitness is larger than the average fitness of the population, and declines otherwise. For a given iteration, adding

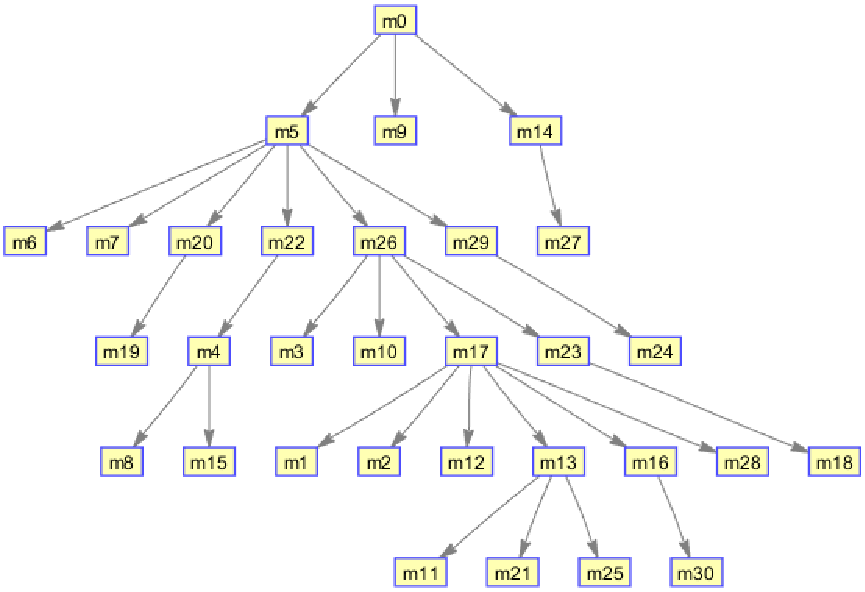


Figure 4.3 Example of simulated mutation tree

clone with a smaller fitness slows down the average fitness growth. As a result, for preceding clones probabilities of appearances of their children in the future may become higher.

4.3 Results

4.3.1 Simulated data

We simulated 100 test examples with the numbers of mutations ranging from $m = 30$ to $m = 120$, which correspond to numbers of mutations for real single cell sequencing data analyzed in previous studies[69,82,86]. For each test example, clonal evolution was simulated as follows. (a) Mutations $1, ..., m$ are generated randomly. For the time interval between mutation events i and $i + 1$ the current mutation rate θ_i is sampled from the normal distribution with the mean $\theta = 0.01$ and standard deviation $\sigma \in \{0.1 \cdot \theta, 0.5 \cdot \theta, 0.9 \cdot \theta\}$. At each moment of time of that interval a mutation event happens with the probability θ_i; at the event a random clone p selected with the probability equal to its current relative abundance gives birth to a new clone j with the random fitness f_j by acquiring a random mutation $i+1$. In our primary fitness sampling scheme, new fitness is sampled uniformly from the interval $[\phi, f_{max}]$, where ϕ is an average fitness of the population at the time of mutation event. This scheme accounts for the fact that according to the evolutionary model (4.2) the clone with the fitness below ϕ is not viable and will not be observed at sampling time. In additional set of experiments, the secondary sampling scheme has been employed, when new fitness is sampled uniformly from the interval $[f_{min}, f_{max}]$ (by default $f_{min} = 1$, $f_{max} = 1.2$). When there is no mutation event, abundances of existing clones are updated according to (4.2). After the end of the simulation, final abundances were randomly perturbed by 10% to incorporate the possible noise in their estimation. The simulated mutation tree and clone abundances were used as an input for SCIFIL.

It should be noted that the construction of the proposed algorithm implies that its performance would be higher on mutation trees with monoclonal structure, both in terms of speed and accuracy. However, our simulation scheme predominantly produces trees with polyclonal structures (see Fig. 4.3), thus providing no a priori advantage to SCIFIL.

We quantified the performance of SCIFIL using two measures:

1) Mean relative accuracy $MRA = 1 - \frac{1}{n}\sum_{i=1}^{n}\frac{|f_i^* - f_i|}{f_i^*}$, where f_i^* and f_i are true and inferred fitnesses, respectively.

2) Spearman correlation SC between true and inferred fitnesses.

MRA and SC highlight different aspects of the problem. MRA measure the accuracy of fitness value estimation, while SC measures how well we are able to qualitatively detect selective advantage of particular clones over other clones. Fitness ranking can be used in evolutionary studies even when actual fitness values are missing or inaccurate[31].

The results of SCIFIL evaluation on simulated data are shown on Figs. 4.4-4.5. The algorithm demonstrated high accuracy as measured by both parameters. The number of mutations (Fig. 4.4) does not have a great impact on the Spearman correlation, which averages 97.35% (standard deviation 1.2%) over all analyzed test cases. MRA decreases when the number of mutations grows, but remains above 88% for all datasets. Increase in variation of mutation rate (Fig. 4.5) does not significantly affect SC, and results in slight decrease of average MRA. Relative robustness of SCIFIL to the variation of mutation rates (which also introduce variation in mutation times) indirectly suggests, that the proposed algorithm is able to well approximate the original maximum likelihood problem (4.4). In the case of near-neutral selection ($f_{max} = 1.01$), MRA does not significantly change and SC declines to 87.54%.

Additionally, we have compared SCIFIL output with the topology of input mutation trees to evaluate the contribution of the tree-based prior information to the algorithm's accuracy. Specifically, the clones have been ranked by their estimated fitnesses and by their tree heights, and Spearman correlation SC^T between fitnesses and tree ranks have been calculated (combined with the permutation test to account for the presense of clones of the same rank). The experiment has been repeated two times using the primary and secondary fitness sampling schemes, with the latter being a completely random uniform sampling from the constant interval. For the first

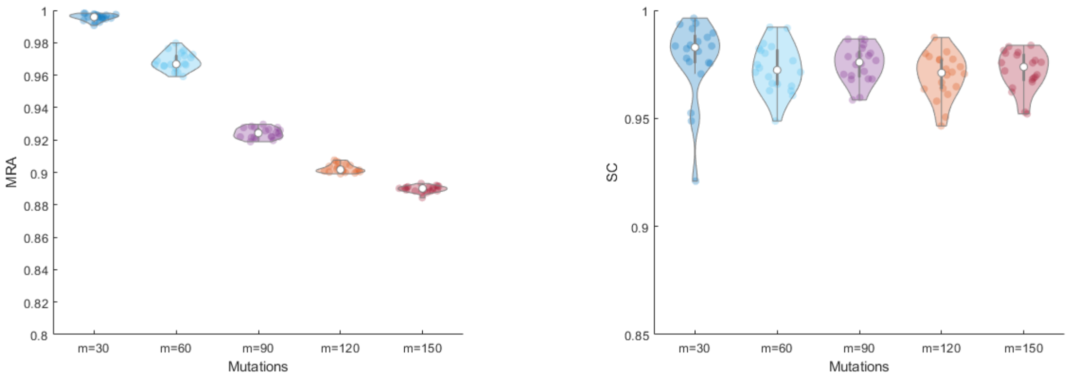


Figure 4.4 Performance of SCIFIL on simulated data with m mutations and fixed standard deviation of mutation rate. Left: mean relative accuracy of fitness estimation. Right: Spearman correlation between true and inferred fitness vectors

sampling scheme, the average correlation between fitness and tree ranks was $SC_1^T = 0.698$ (with $SC = SC_1 = 0.969$). For the second sampling scheme, SC^T drops to $SC_2^T = 0.314$, while the correlation between real and estimated fitnesses decreases to $SC_2 = 0.871$. The value $\tau = 100 \cdot \frac{SC_1 - SC_2}{SC_1^T - SC_2^T}$ (decrease in accuracy per one percent decrease in tree/fitness correlation) may serve as a measure of contribution of a tree topology to the SCIFIL quality. In our case, this value is equal to 25.7%. Transition to near-neutral selection ($f_{max} = 1.01$) has the similar effect, with the correlations being $SC = 0.875$ and $SC^T = 0.379$.

ScSeq data are prone to errors. To evaluate SCIFIL's robustness to trees inferred from noisy data, random errors were introduced to clone mutation profiles at false negative rates $\alpha \in \{0.1, 0.2\}$ and the false positive rate $\beta = 10^{-5}$, and mutation trees were reconstructed from these profiles using the state-of-the-art tool SCITE[69]. The simulated/reconstructed mutation trees were used as an input for SCIFIL. It turned out that in $\sim 8\%$ of cases SCIFIL was not able to produce a feasible solution. This issue could be resolved by performing several additional steps of the local

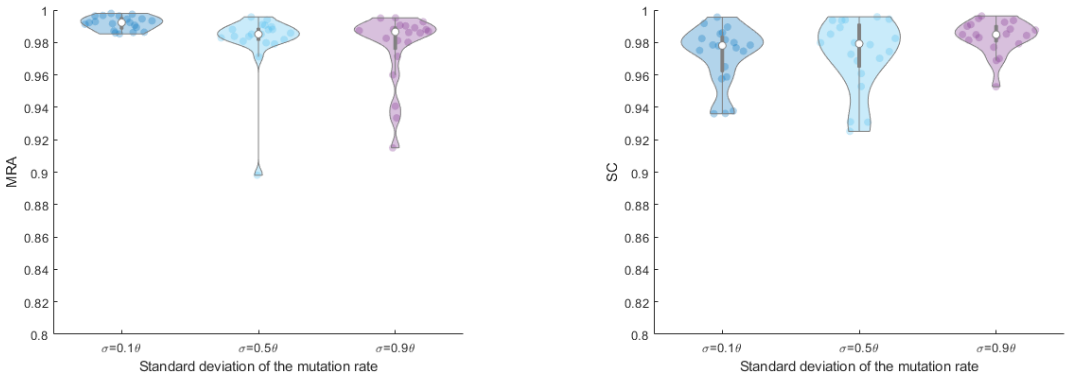


Figure 4.5 Performance of SCIFIL on simulated data with $m = 50$ mutations and different standard deviations of mutations rates. Left: mean relative accuracy of fitness estimation. Right: Spearman correlation between true and inferred fitness vectors

search with the same tree modification operations as SCITE and with the objective (4.4). With this modification, SCIFIL reconstructs fitnesses accurately, although, as expected, the accuracy decreases with the error rate's growth (Fig. 4.6). To check the influence of undersampling, we assumed that $\gamma = 10\%$ of clones with lowest frequencies were not observed at the sampling time. For such clones, the auxiliary frequency $\delta << \varepsilon$ has been assigned before running SCITE. For $m = 50$, the average MRA decreased from 0.99 to 0.96 in comparison to the complete data, but SC remained stable (0.972 and 0.968, respectively).

Finally, we compared our approach with the previously published tool QuasiFit[146]. Although originally designed for viruses, QuasiFit is based on quasispecies model, which is applicable to both intra-host viral populations and cancer clone populations[178] and is essentially a fully continuous version of the model used by SCIFIL. Both QuasiFit and SCIFIL reconstruct replicative fitnesses of individual clones (rather than alleles). In addition to genomic data, both algorithms utilize other information: SCIFIL uses a mutation tree, while QuasiFit assumes that the population

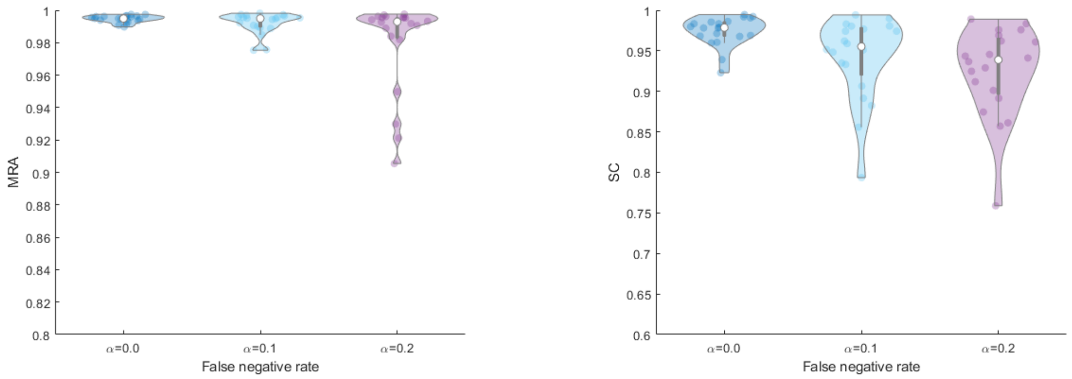


Figure 4.6 Performance of SCIFIL on simulated data with different false negative error rates α and with mutation trees reconstructed by SCITE[69]. Left: mean relative accuracy of fitness estimation. Right: Spearman correlation between true and inferred fitness vectors

is in equilibrium state of the quasispecies model. Thus, SCIFIL has access to information about partial clones order encoded by the mutation tree, while equilibrium site assumption allows QuasiFit to eliminate from consideration the temporal component. Furthermore, SCIFIL is a discrete optimization approach, while QuasiFit implements Markov Chain Monte Carlo sampling.

QuasiFit was run with the per-cell mutation rate $\mu = \varepsilon\theta$ (which is a fully continuous analogue of the parameters used by SCIFIL) and fitnesses were estimated after a burn-in of 10^5 iterations. As QuasiFit uses a different fitness vector normalization, following[146] we used only the parameter SC for the comparison. The results are shown on Fig. 4.7 (left). On our simulated data, SCIFIL outperforms QuasiFit indicating that in certain settings the proposed model could be more accurate for the inference of clonal selection than the equilibrium state assumption.

Computational experiments suggest that the algorithm's running time scales quadratically with the number of mutations (Fig. 4.7, correlation = 0.981). It allows SCIFIL to finish in a few seconds for all analyzed data sets when run on a simple desktop computer.

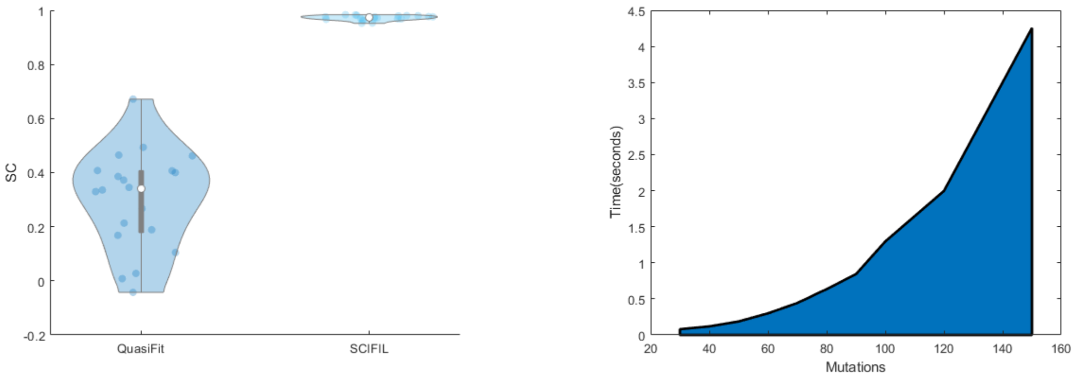


Figure 4.7 Left: Spearman correlation between true and inferred fitness vectors for QuasiFit and SCIFIL. Right: running time of SCIFIL

4.3.2 Experimental data

Fitness landscapes. We used SCIFIL to infer fitness landscapes for 2 recently published experimental cancer datasets. The first dataset is single-cell sequencing data from a JAK2-negative myeloproliferative neoplasm (essential thrombocythemia)[68], the second one represents metastatic colon cancer[86]. The latter dataset includes SNVs sampled from the main tumor and two metastases. We confined our analysis only to the primary tumor, since it is biologically meaningful to compare fitnesses of clones sampled from the same environment. For both datasets, their mutation trees were reconstructed using SCITE[69], and fitnesses and mutation appearance times were inferred by SCIFIL with the cell-wise mutation rate 10^{-6}. It is important to note that varying SCIFIL parameters may change absolute values of inferred fitnesses, but preserve relations between them. The relations are the most informative factors for evolutionary analysis.

We visualized inferred fitness landscapes as follows. We calculated pairwise distances between clones defined as the sum of their hamming distance and the absolute difference of their orders of appearance. The distances were used to map clones to the plain $\mathbb{R}^2$ using multidimensional scaling. Fitness values of the points corresponding to clones were interpolated using biharmonic splines, and the resulting surface was visualized as a contour plot (Fig. 4.8), where colors represent fitness values, and distance from each tree node to the root reflects its appearance time.

For myeloproliferative neoplasm (Fig. 4.8, left) we observe linear accumulation of mutations with slight selective advantages at the beginning of tumor evolution, followed by the subclone expansion of two lineages with significantly faster fitness growth. The rate of fitness growth after the branching event is ~ 3 times higher than before it. Thus, answering the question posed in[69], we may predict that recent subclones will replace ancestral clones. However, based on the available information it is hard to decide whether one of the subclone lineages will out-compete the other one, or they will continue to coexist.

Evolution of the colon tumor (Fig. 4.8, right) follows different scenario, with 3 independent lineages co-existing at the beginning without a clear selective advantage enjoyed by any of them. This stage is followed by the fast expansion of one of the lineages, which climbs a fitness peak and acquires selective advantage over other lineages. Exactly at this stage the advantageous lineage seeded the metastatic tumor at two seeding events (highlighted in black on Fig. 4.8).

Experimental data also allow to emphasize how SCIFIL estimations extend predictions implied by the underlying evolutionary model. Although the model suggests positive selection with fitness growth along each path of the mutation tree as the most probable scenario, it does not imply any restrictions on the comparative fitnesses of different lineages. In particular, fitness advantages of clones are not defined only by their distances from the root, as emphasized by the fitness landscape

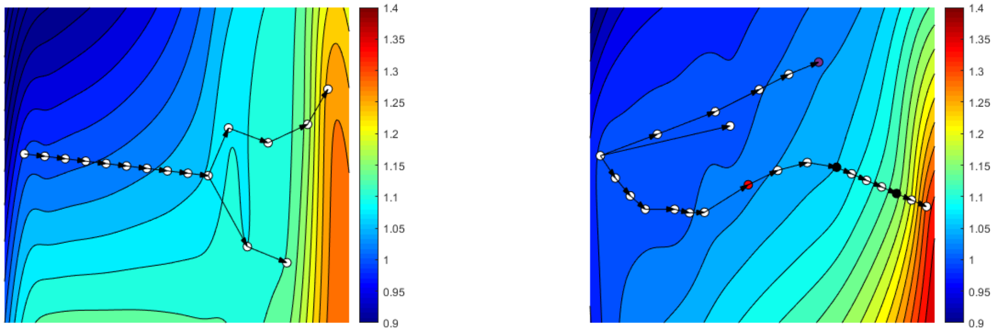


Figure 4.8 Fitness landscape and mutation tree for JAK2-negative myeloproliferative neoplasm[68] (left) and colorectal cancer (right)[86] inferred by SCIFIL. Colors represent fitness values, and distance from each tree node to the root is approximately proportional to its time of appearance.

of the colon tumor, where, for instance, the node highlighted in purple has higher fitness than the node highlighted in red. The reason is that clone abundances contribute to the estimation of fitness values as much as the evolutionary model and the topology of mutation tree.

Recurrent mutations. Until recently, most studies of tumor evolution utilized *infinite sites assumption*, which states that every genomic position mutates at most once over the evolutionary history. However, recently it has been demonstrated using ScSeq data, that the infinite site assumption could be violated, with the same genomic positions mutationally affected multiple times over the tumor evolution[83]. Without infinite site assumption, the number of possible alternative evolutionary histories accurately explaining the observed ScSeq data increases, and it becomes challenging to choose the most appropriate one.

We utilized SCIFIL for the analysis possible evolutionary histories with recurrent mutations for a JAK2-negative myeloproliferative neoplasm[68]. We used infSCITE[83] to generate the perfect phylogeny and 18 mutation trees T_{m_i} under the assumption that one of 18 mutations m_i has a recur-

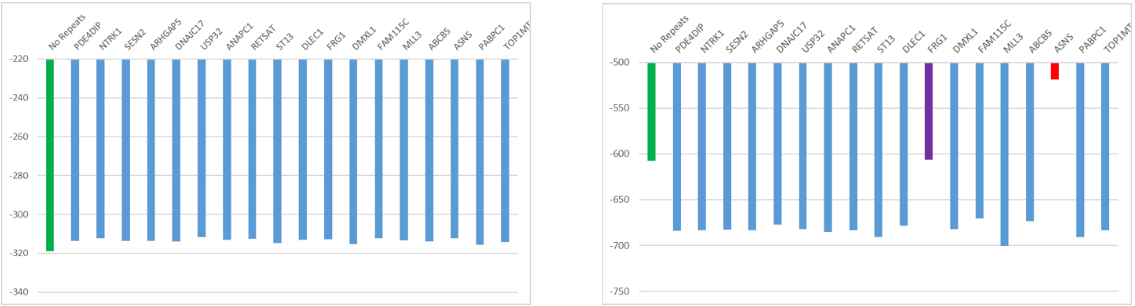


Figure 4.9 Log-likelihoods of trees with and without recurrent mutations. Left: log-likelihoods produced by infSCITE. Right: evolutionary likelihoods produced by SCIFIL. Likelihoods of perfect phylogeny are shown in green. Purple and red: trees with the evolutionary likelihoods higher than for the perfect phylogeny.

rence (*recurrence trees*). Just as reported in[83], the results strongly support recurrent mutations: the average log-likelihood for recurrence trees produced by infSCITE in our experiments was -313.45 (standard deviation 1.065), while the log-likelihood of the perfect phylogeny was equal to -319.08 (Fig. 4.9). However, differences between log-likelihoods of recurrence trees were small in comparison to their difference with the one of the perfect phylogeny, thus impeding the reliable selection of the single most likely recurrence tree. To choose such tree, we utilized evolutionary likelihood estimated by SCIFIL. Among 18 trees, only 2 have evolutionary likelihoods higher than for the perfect phylogeny (Fig. 4.9). Notably, the log-likelihood of the tree T_{ASNS} is significantly higher than for other recurrence trees (-518.62 vs -674.696 in average (standard deviation 25.62)), thus providing the strong support for that particular evolutionary history with respect to other possible histories. These results indicate that SCIFIL's can be efficiently used in conjunction with infSCITE or other similar tool for detection of the most probable evolutionary scenarios.

4.4 Discussion

Intra-tumor heterogeneity is one of the major factors influencing cancer progression and treatment outcome. Cancer clones form complex populations of genomic variants constantly evolving to compete for resources, proliferate, metastasize and escape immune system and therapy. Quantification of clonal selection for tumors may provide valuable information for understanding mechanisms of disease progression and for design of personalized treatment. Single cell sequencing provides an unprecedented insight into intra-tumor heterogeneity allowing to study fitness landscapes at finest possible resolution and quantify selective advantages on the level of individual clones.

In this paper, we presented SCIFIL, a likelihood-based method for inference of fitnesses of clonal variants. Unlike other available methods for related problems, SCIFIL takes full advantage of the information about structure and evolutionary history of clonal population provided by single cell sequencing. It uses individual cells as evolutionary units, in contrast to the tools based on bulk sequencing which perform their analysis on the level of subpopulations or lineages. Furthermore, SCIFIL can also handle bulk sequencing data as long as clones are reconstructed and mutation tree is constructed using available tools such as AncesTree[44], PhyloSub[70], CITUP[97].

In contrast to previous approaches, SCIFIL employs dynamic evolutionary model rather than assumption that the population achieved the equilibrium state. We have demonstrated that our approach allows for accurate inference of fitness landscapes and can be used for analysis of evolutionary history and clonal selection for real tumors. We envision that SCIFIL can be also used to infer epistasic interactions and to identify combinations of mutations driving the tumor growth.

In addition, it can be applied to other highly mutable heterogeneous populations, such as viral quasispecies or bacterial communities.

The proposed approach has limitations which should be addressed in the future work. Fitness is not defined by the genetic composition alone, and depends on the environment. Thus SCIFIL quantitative predictions are more reliable when the analyzed clones are sampled from the same tumor. Fitness inference relies on the observed clone abundances, and therefore significant inaccuracies in abundance estimation may affect accuracy of fitness reconstruction. For single cell data it is particularly important owing to its susceptibility to allelic dropouts and PCR bias. However, this problem can be addressed by using a combination of bulk and single cell sequencing data. There exist a plethora of tools which can estimate clone abundances from composite bulk and single cell sequencing data (see, e.g.[12,114]). In addition, such composite data can be employed to increase an accuracy of mutation trees reconstruction[98]. We expect SCIFIL reliability to increase when it will be combined with these tools.

Another set of limitations arise from the selected evolutionary model (4.2). It was selected due to its generality[178] and suitability for fitness landscape inference[120]. However, it has certain underlying assumptions: the mutation rates are supposed to be normally distributed, while the dynamical system (4.2) implies positive selection with the gradual growth of average population fitness. It should be noted that in many cases such assumptions are sufficiently realistic, and have been used in several studies to obtain valuable insights into the dynamics of tumor evolution[20,71]. In particular, other studies demonstrated that even a normal mutation rate is sufficient to produce significant intra-tumor heterogeneity and emphasized the relative importance of selection over both the size of the cell population and the mutation rate[14]. Although equations (4.12) suggest that in most cases fitness growths along each path of the mutation tree, the model does not imply any restrictions on the comparative fitnesses of different lineages. Furthermore, observed relative abundances of clones are independent of the model, and their contribution to the estimated fitness values is paramount. Nevertheless, we expect that our approach can be extended by incorporating other models capturing different evolutionary scenarios, such as gradual mutation rate growth over the course of tumor evolution, and clonal clonal competition/cooperation, as well as spatial tumor heterogeneity. It should be noted, though, that currently there is no universal evolutionary model for tumor progression. Alternative models will inevitably introduce other limitations and can be less practical for fitness estimation.

On algorithmic side, the optimization problem behind our approach can be viewed as the type of scheduling problem with precedent constraints and with non-linear objective[36]. Such problems are generally NP-hard, although the complexity of our problem is unknown. It is known that for certain simple objectives and well-structured precedence constraints (e.g. defined by series-parallel graphs) the corresponding scheduling problems are polynomially solvable[36]. For our problem precedence constraints have the form of a tree. It gives a certain hope of existence of exact polynomial or a good approximation algorithm, although the complex objective function may keep our problem NP-hard. This question requires additional study.

CHAPTER 5

INFERENCE OF MUTABILITY LANDSCAPES OF TUMORS FROM SINGLE CELL SEQUENCING DATA

5.1 Introduction

Cancer is a dynamical evolutionary process in the heterogeneous population of subclones[59,186,16], with clonal heterogeneity playing the paramount role in disease progression and therapy outcome[111,40,85]. Intra-tumor *genomic heterogeneity* originated from a variety of somatic events (e.g. SNVs, gains/losses of chromosomes) provides an evolutionary environment that facilitates the emergence of *phenotypic heterogeneity* that manifests itself in the extremely high diversity of phenotypic features within the tumor cell population[40,82,59,186]. The genotype-phenotype mapping is often highly non-linear. It means that the effect of a combination of genes or SNVs is different from the joint effect of these genes or SNVs taken separately[96,91,6].In cancer genomics, examples of such non-linear behaviour include synthetic lethality[96,123], epistasis[106,168] or genetic interactions[18,180]. When phenotypic effects are associated with the reproductive success, they are often summarized within the concept of *fitness landscape*[50,67,157,160]. Within this concept, each genotype is assigned a quantitative measure of its replicative success (*fitness or height of the landscape*).

One of the hallmarks of cancer is the extremely high mutability and genetic instability of tumor cells, with intra-tumor rates of mutation, gain/loss/translocation of chromosomal regions and aneusomy (changes in numbers of chromosomes) often being several orders of magnitude higher than the normal rate[165,58,57,29]. Instability rates of subclones are just as heterogeneous as other phenotypic features. They are also subject to epistatic effects or genetic interactions[136]. As a result, it is reasonable to argue that the mutation or instability rates of a clonal population form a *mutability landscape*, whose structure is shaped by selection and genetic interactions.

Recent advances in sequencing technologies profoundly impacted cancer studies. Until recent years the most prevalent sequencing technology has been bulk sequencing, which produces admixed populations of cells. However, the most promising recent technological breakthrough was the advent of single-cell sequencing (scSeq). In the context of the current study, one of the most important advantages of scSeq is its ability to reliably and accurately distinguish exact cancer clones rather than just SNVs. It allows to study composition and evolution of intra-tumor clone populations at the finest possible resolution and take into account complex topological properties of tumor fitness and mutability landscapes, including those associated with non-linear effects.

A rich arsenal of available phylogenetic models and tools has been applied to scSeq data for solving the first important goal of reconstructing the phylogeny of cancer subclones assuming first infinite site model and then exploring more realistic but challenging models allowing recurrent or backward mutations[69,82,30,43,1]. These advances give an opportunity to address the next important challenge: use reconstructed phylogenies to infer quantitative evolutionary parameters for cancer lineages, which can give cancer researchers a statistically and computationally sound evaluation of the effects of particular mutations or their combinations[84,160,80]. This problem is of paramount importance, especially for the design of efficient treatment strategies in the context of personalized medicine[96,133,141,107,92,84]. However, in contrast to the phylogenetic inference, very few computational tools for assessment of cancer evolutionary parameters are currently available[84,160,80]. In particular, several studies recently addressed the problem of inference of cancer fitness landscapes[157,177]. In this paper, we expand the cancer evolutionary analysis toolkit by proposing a computational method for *inference of mutability landscapes and quantification of genetic instability* within clonal cancer populations.

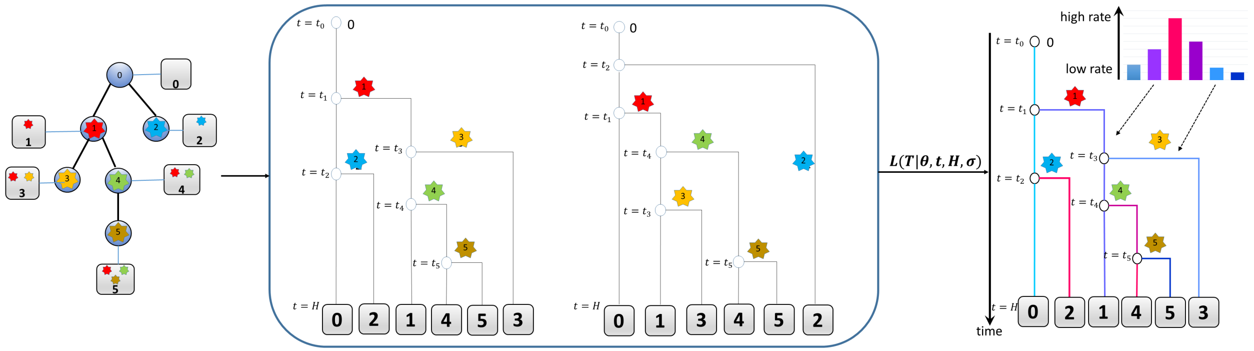
Standard strict molecular clock-based models[22], that assume constant mutation rates, do not accurately reflect the inherent heterogeneity of cancer clone populations. Relaxation of rate constancy in the form of so-called relaxed molecular clock[130,41] or genomic universal pacemaker[158,179] was already introduced in other evolutionary settings such as evolution of species[158,41] or epigenetic aging[159]. However, intrinsic heterogeneity of tumor clonal populations pose additional challenges for rate inference that should be addressed by the methods specifically tailored to cancer settings. The major challenges could be summarized as follows.

First, many currently available methods assume that closely related organisms have similar evolutionary rates[130,142,164] (autocorrelation property) or that rates of different genes are synchronized (genomic universal pacemaker model). In contrast, the genomic stability of individual cells is controlled by multiple molecular mechanisms for DNA damage surveillance, detection, and repair. Disruption or dysregulation of any of these mechanisms could result in different degrees of genomic instability[185]. Thus, it could be expected that mutability landscapes of intra-tumor populations are significantly more rugged than those of species or individual organisms.

Second, reconstruction of mutation rate heterogeneity via phylogenetic inference is more challenging for cancer populations than for species or organisms. Indeed, the estimation of mutation rates requires estimation of times of mutation events. The standard model for such timing is a binary phylogenetic tree, whose internal nodes represent these events and leafs correspond to sampled subclones. The timing is complicated by *polytomies* (ambiguities in order of bifurcations) that should be resolved for the inference. In cases when the expected number of mutations between a parent and its offspring is comparatively large, polytomies are relatively rare, and evolutionary distances between species provide prior information about the order of bifurcations. For the cancer subclonal populations, multiple subclones are usually at the same distance from their common parent (Fig 5.1), thus making polytomies extremely wide-spread. In addition, most existing approaches for single-cell cancer phylogenetics[69,82,30,43,1,99,44,70,97,137,100] use character-based *mutation trees* rather than binary phylogenetic trees (Fig 5.1). The internal nodes of a mutation tree represent mutations, leafs represent subclones, and each subclone have mutations on its path to the root. For such trees, resolution of polytomies is equivalent to finding the orders of sibling nodes, and it is crucial for the mutation rate estimation.

Finally, in established models, changes in genetic instability rates are usually associated with individual mutations. In contrast, a more accurate model would associate them with subclones, which allow capturing the effects of epistasis, including pairwise synthetic lethality, which explains cancer driver genes' tissue specificity[96]. In general, a combined effect of several mutations cannot be explained by a linear regression model, so it is necessary to take into account the entire subclone for estimation of the mutation rate.

Here we propose MULAN (MUtability LANdscape inference) - a likelihood-based method for inference of mutability landscapes of cancer subclonal populations from single-cell sequencing data. It utilizes the partial information about the orders of mutation events provided by cancer mutation trees reconstructed from scSeq data and extends it by inferring full evolutionary history



(a) Mutation Tree T

(b) Binary phylogenies $B_1(T)$ and $B_2(T)$

(c) ML phylogeny with mutation rates

Figure 5.1 **Algorithm for the Maximum Likelihood inference of mutability landscape.** (a) Mutation tree T. (b) Two binary phylogenies $B_1(T)$ and $B_2(T)$ corresponding to two different orders of events $t_0 < t_1 < t_3 < t_2 < t_4 < t_5 < H$ and $t_0 < t_2 < t_1 < t_4 < t_3 < t_5 < H$. Each internal vertex is labeled with its time stamp, thus resulting in the same mutation tree T. Each branch (t_i, t_j) is labeled by the leaf-subclone on the vertical line through its endpoint t_j. All leaves have the sampling time stamp $t = H$. (c) Maximum Likelihood phylogeny and mutability landscape. Mutation rates along the branches corresponding to different subclones are highlighted in different colors.

and mutability landscape of a tumor. To the best of our knowledge, it is one of the first methods specifically tailored to the cancer clone populations and scSeq data and aimed at addressing the aforementioned challenges. In particular, previously published tool SiFit[187] performs a phylogenetic inference, which includes an estimation of deletion and loss of heterogeneity rates, but these rates are assumed to be the same for all subclones. It should be noted that our method infers mutation rates of subclones rather than individual genes, thus making it possible to use the obtained results to detect and quantify genomic interactions and epistasis.

5.2 Materials and methods

5.2.1 Model

Time-aware phylogenetic model. scSeq data are usually represented as a 0-1 matrix in which rows correspond to sequenced cells, and columns correspond to cancer mutations. The set of ones of each row represents a *mutation profile* of a cell. Following most existing approaches for cancer phylogenetics[69,82,30,43,1,99,44,70,97,137,100], our basic cancer cell evolutionary model will be a *mutation tree* $T = (V_T, E_T)$ with the vertex $0 \in V_T$ being the root, the internal nodes of a mutation tree representing mutations connected according to their order of appearance during the tumor evolution, the leaves correspond to the sampled subclones and the mutation profile of each cell being defined by the set of mutations on its path to the root (Fig 5.1A). In what follows, we assume that the ith subclone is attached to the internal node i and does not consider the leaves explicitly. The mutation tree T reconstructed using one of the existing methods from scSeq data constitutes and input of our algorithm. Note that T does not have to be a perfect phylogeny, and can contain both repeated mutations and mutation losses.

Next, we extend the phylogenetic model by accounting for times of mutation events. The mutation tree T provides a *partial* information about these times, as it establishes the order of mutation appearances along each path, but does not do it for sibling mutations. Therefore we need to consider a *binary phylogenetic tree* $B(T)$ corresponding to the mutation tree T. The tree $B(T)$ is defined as follows (see Fig 5.1):

(a) The root represents a subclone at the beginning of cancer lineage evolution.

(b) Each internal node is labeled by timestamp $t = t_i$ representing the birth event of the offspring subclone i,

(c) Each leaf $i = 0, \ldots, n$ represents the sampling event of the subclone i. The tree $B(T)$ is usually assumed to be ultrametric, i.e., all leaves are sampled simultaneously (although the model is generalizable to the non-ultrametric case, as discussed below). H will further denote the sampling time. Note that this value is relative, as the birth time of the root is assumed to be 0.

(d) Each edge (t_i, t_j) is labeled by the parent subclone of the corresponding mutation event (on Fig 5.1 it is the leaf k on the vertical through the endpoint t_j).

(e) The orders of birth events in $B(T)$ and mutation events in T agree with each other

The topology of a binary phylogeny $B(T)$ is uniquely determined by the orderings $\sigma_i = (\sigma_{i,0}, \sigma_{i,1}, ...\sigma_{i,d_i})$ of the offsprings of each node $i = 0, 1, \ldots, n$ in the mutation tree T, where d_i is the degree of the i-th node in T. As a result, for a given mutation tree there are usually several corresponding binary phylogenies. An example of a mutation tree T and the corresponding binary phylogenies $B_1(T)$ and $B_2(T)$ is shown in Fig 5.1. The trees $B_1(T)$ and $B_2(T)$ correspond to two different plausible orders of mutation events.

Mutability landscape likelihood model. Next, we bring in variable mutation rates and introduce the likelihood function. We consider the **mutability landscape evolutionary model** describing subclone evolution with the underlying time-aware model similar to the model described in[138]. In this model, the appearance of mutations in each subclone is a Poisson process and time intervals between consecutive events follow the Erlang distribution. Specifically,

(a) each subclone k has a mutation rate θ_k,

(b) the probability of each edge between internal nodes $e = (t_i, t_j)$ labeled by k in the binary evolutionary tree is calculated as $p(e) = \theta_k^2 (t_j - t_i) e^{-\theta_k (t_j - t_i)}$,

(c) the probability of each edge between an internal node and a leaf $e = (t_i, t_j)$ labeled by k in the binary evolutionary tree is exponential and is calculated as $p(e) = \theta_k e^{-\theta_k(H-t_i)}$.

The total probability of the tree $B(T)$ equals $p(B(T)|\theta, t) = \prod_{e \in E(B(T))} p(e)$.

The described model is used to find mutability landscapes jointly with the most likely binary phylogeny $B(T)$. We first consider the following optimization problem:

Given: A mutation tree $T = (V_T, E_T)$ with mutations $\{0, ..., n\} \in V_T$ and vertex outdegrees $d_0, ..., d_n$.

Find: Mutation rates $\theta = (\theta_i)_{i=1}^n$, times of occurrence $t = (t_i)_{i=1}^n$ of each mutation $i = 1, \ldots, n$ and the sampling time H that maximize the probability $p(T|\theta, t, H, \sigma)$ of the tree T given the model parameters.

As noted above, setting the phylogeny $B(T)$ is equivalent to setting the family of offspring orderings $\sigma = (\sigma_1, ..., \sigma_n)$. For a given ordering family σ we have

$$p(T|\theta, t, H, \sigma) = \prod_{i=0}^{n}(\prod_{j=1}^{d_i} \theta_i^2 (t_{\sigma_{i,j}} - t_{\sigma_{i,j-1}}) e^{-\theta_i(t_{\sigma_{i,j}} - t_{\sigma_{i,j-1}})}) \theta_i e^{-\theta_i(H - t_{\sigma_{i,d_i}})} \tag{5.1}$$

After the straightforward simplifications, the log-likelihood $L(T|\theta, t, H, \sigma)$ can be written as follows:

$$L(T|\theta, t, H, \sigma) = \sum_{i=0}^{n} \theta_i t_i + \sum_{i=0}^{n} \sum_{j=1}^{d_i} \log(t_{\sigma_{i,j}} - t_{\sigma_{i,j-1}}) - (\sum_{i=0}^{n} \theta_i) H + \sum_{i=0}^{n} (2d_i + 1) \log(\theta_i), \tag{5.2}$$

where $t_0 = 0$, $0 \leq t_i \leq H$, $i = 1, ..., n$.

Our goal is to find an optimal ordering σ^*, times t^*, sampling time H^*, and mutation rates θ^* by solving the following maximum likelihood problem:

$$(\theta^*, t^*, H^*, \sigma^*) = \text{argmax}_{(\theta,t,h,\sigma)} L(T|\theta, t, H, \sigma) \tag{5.3}$$

Note that we usually assume that the rate θ_0 is fixed (for example, to the value corresponding to the normal tissue).

The likelihood function (5.2) is non-linear and all nodes effectively contribute to it. This makes straightforward utilization of standard methods based on dynamic programming to solve the problem (5.3) is challenging. Indeed, the model implies that there exists a certain dependency between birth times of sibling subclones since they belong to the same time interval. Suppose that a subclone i mutated twice during the time between its birth and sampling. Although the two acquired mutations are independent and distributed uniformly at random between $t = t_i$ and $t = H$, the expected birth times of two corresponding offsprings are $t_i + (H - t_i)/3$ and $t_i + 2(H - t_i)/3$ rather than $t_i + (H - t_i)/2$. The effect of such non-linear properties of the model could be illustrated using an example on Fig 5.1. Intuitively, clone 1 produced two offsprings, while clone 2 produces zero offsprings. This imbalance can be explained in two ways: either (i) the clone 2 has a higher mutation rate, or (ii) clone 1 was born early and had time to accumulate mutations while clone 2 was born late and didn't have time to accumulate mutations. When assessing these two alternatives, other clones also come into play. For example, the alternative (ii) means (a) the longer interval between the birth of clone 1 and birth of clone 2 – the likelihood of such interval depends on the mutation rate of the parent clone 0; (b) the longer interval between the birth of clone 1 and the sampling – the likelihood of such interval depends on the mutation rates of the descendants of 1. Maximum likelihood inference allows us to choose between these alternatives.

In many real settings the realistic mutation rates are subject to constraints. We account for these considerations by adding to the model a prior probability $p(\theta)$. In this case, we utilize lasso regression-type approach, i.e. we solve the problem (5.3) under the constraint $l(\theta) = \log(p(\theta)) \geq l_0$. The simplest prior assumes that the rates are distributed uniformly on the segment $[\theta_{\min}, \theta_{\max}]$. Assuming that genetic instability increase events are not frequent, we are also particularly interested in the models with the limited number of such events. In *s-model*, we assume that the rate changes in at most s vertices of the mutation tree. When $s > 0$, we assume that one of these rates is the normal rate and, therefore, is fixed.

Finally, we note that it is straightforward to generalize the model to the case when the tumor cells are sampled at different time points. It can be done by allowing different model-based sampling times H_i and setting the differences between them equal to the differences between actual sampling times.

5.2.2 Algorithms

To describe the algorithms and derive the associated mathematical claims, we will use the following notations: T^k is the subtree of T with the root k; d_k is the degree of the node k in T; $n_k = |V(T^k)|$; θ^k is the collection of mutation rates of the vertices in T^k and $\Theta_k = \sum_{j \in V(T^k)} \theta_j$.

A. The case without a prior $p(\theta)$. In this case, we propose to solve the problem using an expectation-maximization approach described by Algorithm 1. This algorithm takes as an input the mutation tree T, feasible rates segment $[\theta_{\min}, \theta_{\max}]$ and initial mutation rates $\theta = \theta^0$, and produce as an output the mutation rates θ^*, times t^*, sampling time H^* and orderings σ^* that are supposed to maximize $L(T|\theta, t, H, \sigma)$. The algorithm is described as follows:

Algorithm 1. EM algorithm for mutability landscape inference

Repeat the following steps until convergence:

M step: for given θ, find t, H and σ maximizing $L_{T,\theta} = L(T|\theta, t, H, \sigma)$ using Algorithm 2.

E step: for times t and H, find the expected rates:

$$\theta_i = \frac{d_i}{H - t_i} \tag{5.4}$$

Next, we describe how M step is carried out. In what follows, we formulate several claims forming the foundation of our approach, and provide their proofs in the Subsection 5.2.3. For the fixed orderings σ and rates θ, (5.3) is a convex optimization problem with linear constraints, and thus it can be efficiently solved using standard techniques[19]. However, orderings σ introduce discontinuity to the objective and discretize the problem, thus making it computationally hard. The number of possible orderings σ is equal to $\prod_{i=0}^{n} d_i!$, which makes an exhaustive search over the space of all orderings infeasible. Therefore our goal is to optimize the search. Specifically, we employ the following dynamic programming approach:

Algorithm 2. Algorithm to find optimal orderings and times, when rates θ are fixed

Input: mutation tree T with the root 0 and its children $1, ..., d$, mutation rates θ

Output: times t^*, sampling time H^* and orderings σ^* maximizing $L_{T,\theta}$

1. Recursively find optimal orderings σ_k^* for the subtrees T^k, $k = 1, ..., d$.

2. Perform an exhaustive search over the set of permutations of $(1, ..., d)$. For each generated permutation σ_0, we solve the problem (5.2) with the orderings $\sigma = \{\sigma_0\} \bigcup_{k=1}^{d} \sigma_k^*$ subject to the constraints $\frac{d_i}{\theta_{\max}} \leq H - t_i \leq \frac{d_i}{\theta_{\min}}$ as a convex optimization problem, and update the current best solution, if necessary. The constraints ensure that the rates calculated at each iteration of EM belong to the feasible interval.

The worst-case running time of Algorithm 2 is $O(\sum_{i=0}^{n} T(n_i) \cdot d_i!)$, where $T(n_i)$ is the running time of a numerical convex optimization algorithm with n_i variables. It makes the algorithm scalable for the majority of real cases when vertex degrees are not high. However, the optimality of solutions produced by Algorithm 2 is not immediately clear, and its analysis requires deeper understanding of the properties of the optimization problem (5.3). Such properties are established by Lemma 1 and Theorem 1. Consider the restricted version of the problem (5.3) with the fixed rates θ and the sampling time H:

$$L_{T,\theta}(H) = \max_{\sigma,t} L(T|\theta, t, H, \sigma). \tag{5.5}$$

Suppose that $1, ..., d$ are the children of the root 0 of T. Then the following recurrent relation holds:

Lemma 1.

$$L_{T,\theta}(H) \approx \max_{\sigma_0} \max_{t_1,...,t_d} \left(H \sum_{k=1}^{d} \Theta_k t_k + \sum_{k=1}^{d} \log(t_k - t_{k-1}) + \sum_{k=1}^{d} n_k \log(1 - t_k) + \right.$$
$$\left. + \sum_{k=1}^{d} L_{T^k,\theta^k}((1 - t_k)H) \right) - \Theta_0 H + n \log(H) + \sum_{i=0}^{n} (2d_i + 1) \log(\theta_i), \tag{5.6}$$

where the maximum is taken over permutations σ_0 of $1, ..., d$ and over $t_1, ..., t_d \in \mathbb{R}$ such that

$0 \leq t_i \leq 1$.

The relation (5.6) can serve as a basis for dynamic programming algorithm. However, it is not guaranteed yet that such algorithm will be efficient. Indeed, it is theoretically possible that the values of the functions L_{T^k,θ^k} are achieved on different orderings for different arguments, thus forcing the algorithm to store an exponential number of subproblem solutions. However, the following Theorem 1 guarantees that Algorithm 2 is exact, when H is large enough.

Theorem 1. *For all large enough H, the optimal ordering σ^* that maximizes (5.5) is the same. It has the form $\sigma^* = \{\sigma_0^*\} \bigcup_{k=1}^{d} \sigma_k^*$, where σ_k^* are optimal orderings of subtrees T^k and σ_0^* is the permutation of $1, ..., d$ that maximizes (5.6).*

B. The case with a prior $p(\theta)$. The simplest prior assumes that the rates are distributed uniformly on the segment $[\theta_{\min}, \theta_{\max}]$. For this model, initial numerical experiments suggest that the selection of the initial solution in the feasible segment ensures convergence of the EM algorithm to the feasible solution. For more complex priors, we utilize specially enhanced Markov Chain Monte Carlo (MCMC) sampling from the rates distribution that will allow for more efficient traversing of the solution space than the default approach. In particular, for s-model, each feasible solution could be represented by the subset $X \subseteq V(T)$ of s internal vertices corresponding to rate change events together with the collection of $s + 1$ rates corresponding to the connected components of $T \setminus X$. Then MCMC draws the new rate from the normal distribution centered on the current rate, while new subset X' is drawn from the 1-flip neighborhood of the current subset X[73] (i.e. $X' = (X \setminus \{u\}) \cup \{v\}$ for some $u \in X$, $v \in V(T) \setminus X$).

5.2.3 Mathematical foundations of the algorithms

In this subsection we prove Lemma 1 and Theorem 1. Due to the space limit, we present the general outline of the proofs and omit some particularly technical details. Let $D[k] = V(T^k)$ and $D(k) = V(T^k) \setminus \{k\}$ be the closed set of descendants and set of descendants of k, respectively.

Proof of Lemma 1. After variable substitution $t_i := t_i/H$, maximization of (5.2) is equivalent to the maximization of

$$L'(T|\theta, t, H, \sigma) = H\sum_{i=0}^{n}\theta_i t_i + \sum_{i=0}^{n}\sum_{j=1}^{d_i}\log(t_{\sigma_{i,j}} - t_{\sigma_{i,j-1}}) - \Theta_0 H + n\log(H) + \sum_{i=0}^{n}(2d_i+1)\log(\theta_i), \tag{5.7}$$

subject to the constraints $t_1 = 0, 0 \leq t_i \leq 1, i = 2, ..., m$.

Suppose that the rates θ, the sampling time H and the family of orderings $\sigma = (\sigma_0, \sigma^1, ..., \sigma^d)$ are fixed. Consider the partial likelihood $M(T|\theta, t, H, \sigma) = H\sum_{i=0}^{n}\theta_i t_i + \sum_{i=0}^{n}\sum_{j=1}^{d_i}\log(t_{\sigma_{i,j}} - t_{\sigma_{i,j-1}})$, which constitutes the part of the total likelihood (5.7) that depends on t and σ. Using simple arithmetic transformations, we get

$$\begin{aligned} M(T|\theta, t, H, \sigma) = H\sum_{k=1}^{d}\Theta_k t_k + \sum_{k=1}^{d}\log(t_k - t_{k-1}) + \sum_{k=1}^{d} n_k\log(1-t_k) + \\ + \sum_{k=1}^{d}\left((1-t_k)H\sum_{i\in D(k)}\theta_i\frac{t_i - t_k}{1-t_k} + \sum_{i\in D[k]}\sum_{j=1}^{d_i}\log\left(\frac{t_{\sigma_{i,j}} - t_k}{1-t_k} - \frac{t_{\sigma_{i,j-1}} - t_k}{1-t_k}\right)\right) \end{aligned} \tag{5.8}$$

Change of variables $t_i := \frac{t_i - t_k}{1-t_k}$, $i \in D[k]$ yields

$$M_{T,\sigma}(H) \approx \max_{t_1,\dots,t_d} \left(H \sum_{k=1}^{d} \Theta_k t_k + \sum_{k=1}^{d} \log(t_k - t_{k-1}) + \sum_{k=1}^{d} n_k \log(1-t_k) + \right. \\ \left. + \sum_{k=1}^{d} M_{T_k,\sigma^k}((1-t_k)H) \right) \quad (5.9)$$

Thus, the relation (5.6) follows. □

Now, let $M_{T,\sigma}(H) = \max_t M(T|\theta, t, H, \sigma)$ and $M_T(H) = \max_\sigma M_{T,\sigma}(H)$. Theorem 1 directly follows from the following lemma:

Lemma 2. *$M_{T,\sigma}(H) \approx a_T H - b_T \log(H) + c_{T,\sigma}$, where a_T and b_T are constants depending only on T, and $c_{T,\sigma}$ is a constant depending on both T and σ.*

Proof. We will prove the lemma by induction. Suppose without loss of generality that d is the outdegree of the root 0 of T, $1,...,d$ are its children and the ordering σ_0 has the form $\sigma_0 = (0, 1, ..., d)$).

a) Suppose that T is a star (i.e. it has 1 internal node and d leafs). Then we have $\sigma = (\sigma_0)$, $n_k = a_{T_k} = 0$ and $\Theta_k = \theta_k$ for all $k = 1, ..., d$. For the objective we have $M(T|\theta, t, H, \sigma) = H \sum_{k=1}^{d} \theta_k t_k + \sum_{k=1}^{d} \log(t_k - t_{k-1})$, where $t_0 = 0$. Karush-Kuhn-Tucker (KKT) optimality conditions for t have the following form:

$$\begin{aligned} H\theta_k + \frac{1}{t_k - t_{k-1}} - \frac{1}{t_{k+1} - t_k} &= 0, \quad k = 1,..,d-1, \\ H\theta_d + \frac{1}{t_d - t_{d-1}} - \mu_d &= 0, \quad t_d = 1, \end{aligned} \quad (5.10)$$

where μ_d is the dual variable corresponding to the constraint $t_d \leq 1$. After multiplying the kth equation by t_k and summing the obtained equations we get $H \sum_{k=1}^{d} \theta_i t_i = \mu_d - d$. Furthermore, (5.10) yield that $t_k - t_{k-1} = 1/(\mu_d - H\sum_{i=k}^{d}\theta_i)$. These identities imply the following formula for $M_{T,\sigma}(H)$:

$$M_{T,\sigma}(H) = \mu_d - d - \sum_{k=1}^{d}\log(\mu_d - H\sum_{i=k}^{d}\theta_i), \tag{5.11}$$

where $\mu_d \geq H\sum_{i=1}^{d}\theta_i$ and μ_d satisfies the equation $\sum_{k=1}^{d}\frac{1}{\mu_d - H\sum_{i=k}^{d}\theta_i} = 1$. We will seek for the approximation of μ_d of the form $\mu_d = H\sum_{i=1}^{d}\theta_i + \varepsilon$, where $\varepsilon > 0$. Then from the equation for μ_d we have $\frac{1}{\varepsilon} + \sum_{k=2}^{d}\frac{1}{H\sum_{i=1}^{k-1}\theta_i + \varepsilon} = 1$. For large H, we have $\frac{1}{\varepsilon} + o(1) = 1$, thus implying that the good approximation is achieved when $\varepsilon = 1$. By substitution the expression for μ_d to (5.11) we get

$$M_{T,\sigma}(H) = H\sum_{i=1}^{d}\theta_i + 1 - d - d\log(H) - \sum_{k=1}^{d}\log(\sum_{i=1}^{k-1}\theta_i + o(1)) \approx a_T H - b_T\log(H) + c_T, \tag{5.12}$$

where $a_T = \sum_{i=1}^{d}\theta_i$, $b_T = d$ and $c_T = -\sum_{k=1}^{d}\log(\sum_{i=1}^{k-1}\theta_i) - d + 1$. The only term depending on the order σ here is the term $\sum_{k=1}^{d}\log(\sum_{i=1}^{k-1}\theta_i)$, which achieves the minimal value (thus maximizing $M_T(H)$), when $\theta_1 \leq \theta_2 \leq ... \leq \theta_d$. Thus, the base case for the induction is proved.

b) Now suppose that T is not a star. By the induction hypothesis, for every subtree T_i the same ordering σ^k maximizes $M_{T_k}(H)$ for all H. These ordering also define the corresponding optimal binary phylogenies B_k. We claim that it is possible to approximately estimate the optimal times $t_1, ..., t_d$ and ordering σ_0 recursively, if the solutions for the subtrees T_k are known. The following arguments slightly differ technically for the cases when d is a leaf or an internal vertex. We will demonstrate the scheme of the proof for the former case (the latter case could be handled similarly).

Consider the relation (5.6). After applying the induction hypothesis to M_{T_k,σ^k} we get the expression

$$M_{T,\sigma}(H) \approx \max_{t_1,\dots,t_d} \left(H \sum_{k=1}^{d} \Theta_k t_k + \sum_{k=1}^{d} \log(t_k - t_{k-1}) + \sum_{k=1}^{d} n_k \log(1-t_k) + \right.$$
$$\left. + \sum_{k=1}^{d} (a_k H(1-t_k) - b_k \log(H(1-t_k)) + c_k) \right), \quad (5.13)$$

where $a_k = a_{T_k}$, $b_k = b_{T_k}$ and $c_k = c_{T_k,\sigma_k}$. Using the approximation $\log(1-t_k) \approx -t_k$, we rewrite it as

$$M_{T,\sigma}(H) \approx \max_{t_1,\dots,t_d} \left(\sum_{k=1}^{d} (H(\Theta_k - a_k) + b_k - n_k) t_k + \sum_{k=1}^{d} \log(t_k - t_{k-1}) \right) +$$
$$+ H \left(\sum_{k=1}^{d} a_k \right) - \log(H) \left(\sum_{k=1}^{d} b_k \right) + \sum_{i=1}^{k} c_k, \quad (5.14)$$

Let $\lambda_k = H(\Theta_k - a_k) + b_k - n_k = H(\Theta_k - a_k) + o(H)$, $k = 1, ..., d$. As in a), we will use KKT optimality conditions for $t_1, ..., t_d$, which in this case have the following form:

$$\lambda_k + \frac{1}{t_k - t_{k-1}} - \frac{1}{t_{k+1} - t_k} = 0, \quad k = 1, .., d-1,$$
$$\lambda_d + \frac{1}{t_d - t_{d-1}} - \mu_d = 0, \quad t_d = 1 \quad (5.15)$$

where μ_d is the dual variable corresponding to the constraint $t_d \leq 1$. Similarly to a), after multiplying the kth equation by t_k and summing the obtained equations we get $\sum_{k=1}^{d} \lambda_i t_i = \mu_d t_d - d$

and $t_k - t_{k-1} = 1/(\mu_d - \sum_{i=k}^{d} \lambda_i)$. These identities imply that

$$M_{T,\sigma}(H) \approx \mu_d - d - \sum_{k=1}^{d} \log(\mu_d - \sum_{i=k}^{d} \lambda_i) + H\left(\sum_{k=1}^{d} a_k\right) - \log(H)\left(\sum_{k=1}^{d} b_k\right) + \sum_{i=1}^{k} c_k. \quad (5.16)$$

As above, we can use the approximation $\mu_d \approx \sum_{k=1}^{d} \lambda_k + 1$. It implies that

$$M_{T,\sigma}(H) \approx H\left(\sum_{k=1}^{d} \Theta_k\right) - \log(H)\left(d + \sum_{k=1}^{d} b_k\right) - \sum_{k=2}^{d} \log(\sum_{i=1}^{k-1}(\Theta_i - a_i)) + \sum_{i=1}^{k}(c_k + b_k - n_k) - d + 1. \quad (5.17)$$

In this formula, only the constant term depends on the order of vertices. Theorem is proved. □

5.2.4 Quantification of rate estimation uncertainty.

MULAN implements a maximum likelihood approach that uses the combination of discrete optimization and continuous optimization techniques to infer the solution that explains the observed data in the best possible way. In this, it follows the same paradigm as other recently published scSeq analysis tools[45,100,143]. However, given the uncertainty of the mutation tree estimation, it could be beneficial to provide errors or confidence intervals for the inferred rates. One possible way to do it is to combine MULAN with any tree topology sampling scheme by calculating mutation rates for the trees sampled from the particular posterior distribution given the scSec data (after burn-in). This procedure generates the posterior distribution of inferred mutation rates that can be used to calculate standard errors and/or confidence intervals. Here, we implemented this approach by combining MULAN with the tree sampling procedure utilized by SCITE[69].

5.3 Results

5.3.1 Simulated data

In this subsection, we report the results of validation of the proposed algorithm using simulated datasets. We simulated test examples with the numbers of mutations ranging from $m = 70$ to $m = 150$, which correspond to numbers of mutations for real single-cell sequencing data analyzed in previous studies[69,82,86]. For each test example, the simulation starts with the single clone without mutations and with the random mutation rate θ_0. At subsequent iterations, existing clones i produce offspring at rates θ_i; at each such event an existing clone i gives birth to a new clone j with the mutation rate θ_j uniformly sampled from the interval $[\theta_{min}, \theta_{max}]$ (by default $\theta_{min} = 0.005$, $\theta_{max} = 0.01$) by acquiring a random mutation from the set $\{1, ..., m\}$. The simulation ends when the desired number of clones is produced.

We validated the ability of MULAN to infer all three families of parameters of the model (5.3), i.e., the transmission rates, the times of mutation events, and the binary tree topology (or, equivalently, orderings of offspring of the mutation tree nodes). For the primary experiments, Algorithm 1 was executed with the initial mutation rates $\theta_i^0 = \frac{1}{2}(\theta_{\min} + \theta_{\max}), i = 1, ..., m.$ The following accuracy measures were used:

- Rate and time inferences were quantified by the mean absolute percentage accuracy $MAPA = 1 - MAPE$, where $MAPE$ is the mean absolute percentage error.
- Ordering inference was quantified by the mean Kendall tau distance between true and inferred offspring orders for the nodes with outdegrees $d_i \geq 2$.

The mutation rates of leafs were not considered, since they do not have offsprings required for reliable rate estimation.

The results of MULAN evaluation on simulated trees are shown in Fig 5.2. The mean accuracies of rate, time and order inference were 0.86 ($std = 0.02$), 0.92 ($std = 0.11$) and 0.98 ($std = 0.01$), respectively. The ability of MULAN to accurately reconstruct tree topologies is particularly important, as it validates the application of MULAN to the analysis of evolutionary histories described in Subsection 5.3.2. The number of mutations does not have a great impact on the algorithm accuracy, possibly because the algorithm is likely to produce the optimal solution with respect to the objective (5.2) owing to the optimized search over the space of possible mutation orderings and the accuracy of the estimations suggested by Theorem 1. Indeed, the crucial assumption of our approach is based on Theorem 1, which establishes the hierarchy of mutation orderings that is valid for all sampling times. Although Theorem 1 operates with approximations, the experimental validation suggests that this hierarchy is always valid (Fig 5.3, right). Changing initial conditions to the random values uniformly sampled from the interval $[\theta_{\min}, \theta_{\max}]$ does not significantly affect the results, with the mean rate, time and order inference accuracy changing to 0.83, 0.92 and 0.96, respectively.

In another evaluation experiment, we compared MULAN with an MCMC-based method, which samples from the space of tree edge lengths using the method proposed in[187], calculates birth times and orderings from these lengths and estimates mutation rates using (5.4). The mean accuracies of rate, time and order inference of this method were 0.72 ($std = 0.03$), 0.40 ($std = 0.11$) and 0.18 ($std = 0.16$), respectively (Fig 5.3, left). We also verified MULAN's robustness to the sequencing noise and to the choice of the tumor phylogeny inference method. In that case, random errors were introduced to clone mutation profiles with $n = 70$ mutations and with 3 copies of each clone at false-negative rates $\alpha = 0.1$ and the false positive rate $\beta = 10^{-5}$, the mutation trees were reconstructed from these profiles using the state-of-the-art tool SCITE[69] and the recently released tool PhISCS-BnB[100,8]. The accuracy of rate inference was affected insignificantly (Fig 5.3) indicating the robustness of MULAN results to the sequencing noise provided the properly selected phylogeny inference algorithm.

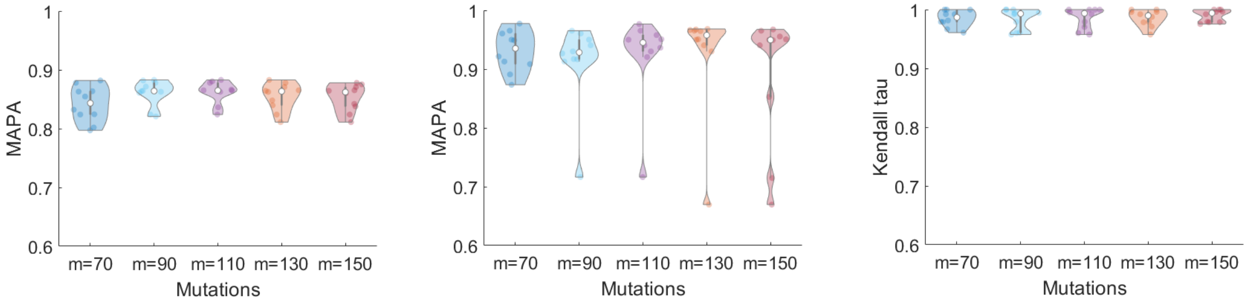


Figure 5.2 Performance of MULAN on simulated data with $n = 70, ..., 150$ mutations. Left: accuracy of rate estimation. Center: accuracy of times estimation. Right: accuracy of orderings estimation.

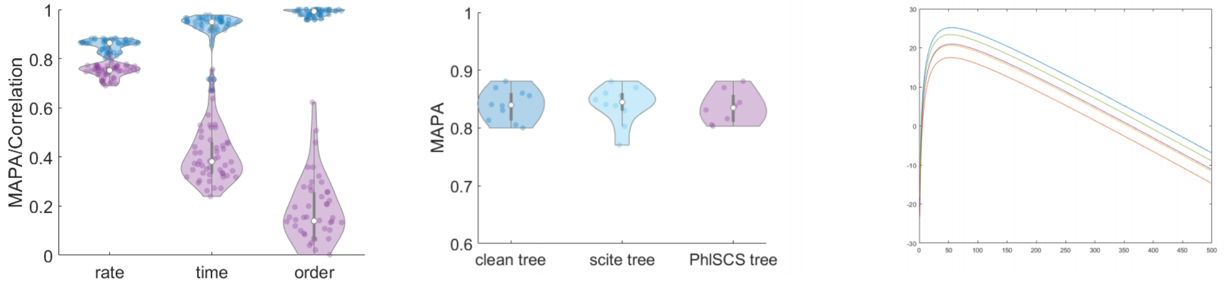


Figure 5.3 Left: accuracies of rate, time and order estimation for MULAN (blue) and MCMC algorithm (red). Center: accuracy of rate estimation ($n = 70$) for the clean data and the trees inferred by SCITE and PhISCS-BnB from noisy data. Right: likelihoods $L_{T,\sigma}(H)$ for different orderings σ. The graph demonstrates the hierarchy of orderings based on the corresponding likelihoods that remain the same for all sampling times H.

The algorithm scales polynomially with the problem size and produces the results within minutes (Fig 5.4, left). In the overwhelming majority of cases, EM converges within 10 iterations.

Finally, Fig 5.4, center and right, demonstrates the posterior distributions and relative standard errors (i.e. the standard error divided by the mean) of inferred mutation rates for several test datasets, as estimated using the method described in Subsection 5.2.4.

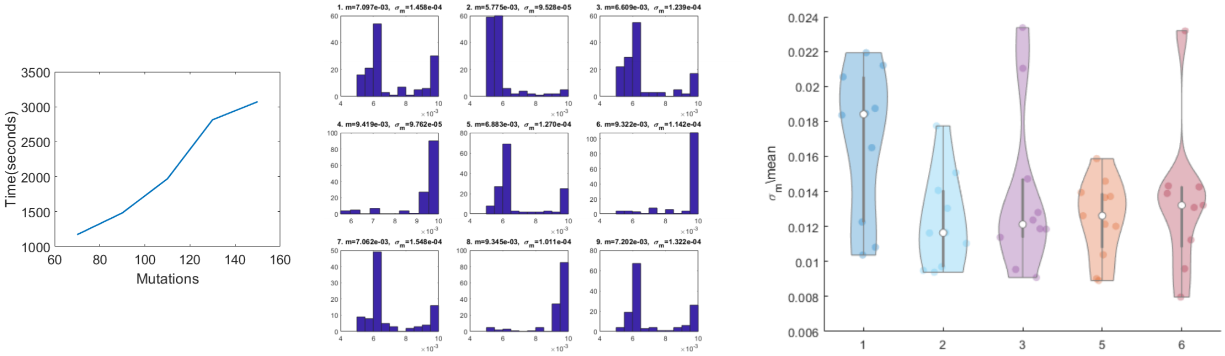


Figure 5.4 Left: algorithms' running time. Center: the posterior distributions of inferred mutation rates for 9 selected subclones in one of the test datasets. Each small plot shows the rate distribution for the particular subclone together with the mean value m and the standard error σ_m. Right: distributions of relative standard errors of rate distributions for five test datasets.

5.3.2 *Experimental data*

In this subsection, we used MULAN to analyze scSeq data from $JAK2$-negative myeloproliferative neoplasm[68] and from lymphoblastic leukemia[51]. The datasets contain 18, 20, 16, 10 mutations and 58, 111, 115 and 146 cells, respectively, and were analyzed as is without any modifications.

Analysis of evolutionary histories. Here we used the MULAN model to assess the likelihoods of alternative tumor evolutionary histories. The datasets under consideration were used in[83] to demonstrate the violation of the infinite site assumption. For a dataset with m mutations, the authors of[83] used the tool infSCITE to infer the perfect phylogeny and m mutation trees T_i with one of m mutations i having a recurrence (*recurrence trees*). According to the error-based likelihood model used in[83], the recurrent trees have much higher likelihoods than the perfect phylogeny (Fig 5.5), thus strongly pointing to the presence of recurrent mutations. However, differences between the likelihoods of recurrence trees are of much smaller magnitude than their difference with the perfect phylogeny. It suggests that without the infinite site assumption, the number of possible alternative evolutionary histories accurately explaining the observed ScSeq data increases, and it becomes challenging to choose between by taking into account only sequencing errors. In what follows we demonstrate that evolutionary-based likelihood estimated using MULAN allows to significantly reduce the set of plausible evolutionary histories.

For each tree constructed by infSCITE, we estimated the following:

(a) the evolutionary likelihood of the most probable fitness landscape, as calculated by our recently published tool SCIFIL[157]. Roughly speaking, this likelihood measures the probability to observe given subclone frequencies when the clonal population evolutionary trajectory over the most likely inferred fitness landscape is described by the tree T.

(b) the likelihoods of mutation instability landscapes with three mutation rates, one of which correspond to the normal rate.

It turned out that for the analyzed dataset, mutability likelihoods and evolutionary likelihood provided an additional strong signal that allows to resolve the ambiguities present in the error-based model. It is especially visible for the $JAK2$-negative myeloproliferative neoplasm (Fig 5.5). There, both likelihoods point to the same two mutations $FRG1$ and $ASNS$ as most probable recurrent mutations and trees T_{FRG} and T_{ASNS} as most probable trees. Only these two trees had higher likelihoods than the perfect phylogeny (even despite the fact that they define more transmission events), and their mean mutability log-likelihoods were higher than for other recurrence trees: -70.46 ($std = 1.53$) vs -78.75 ($std = 7.79$).

Independent acquisitions of mutations with confirmed cancer effects in parallel lineages potentially indicate the convergent evolution and may be suggestive of their evolutionary advantage. In this context, it should be noted that both $FRG1$ and $ASNS$ have been identified in[68] as belonging to the shorter list of selected mutations having the highest likelihood of being involved in essential thrombocythemia initiation and/or progression. Furthermore, 5 out of 7 most likely repeated mutations identified by MULAN belong to that list.

For the lymphoblastic leukemia datasets, the signal was not so strong, possibly because introductions of repeated mutations did not significantly alter the topologies of the recurrence trees (see[83]), thus resulting in many of them having close mutability likelihoods. Nevertheless, even then, the correlations between evolutionary and mutability likelihoods of the trees of the 5 analyzed datasets were 0.85, 0.31, 0.96, 0.91, and 0.69, respectively, with both models agreeing on the most probable recurrence trees. The fact that the same signal was produced by two independent models can be considered as an indicator of their validity. It also suggests that the reliable inference of tumor phylogenies under the finite site assumption requires the utilization of advanced likelihood models that take into account the dynamics of cancer evolution in addition to the simpler models regulating the number and type of mutation events.

Analysis of mutability models. In this set of experiments, our purpose was to test the assumption

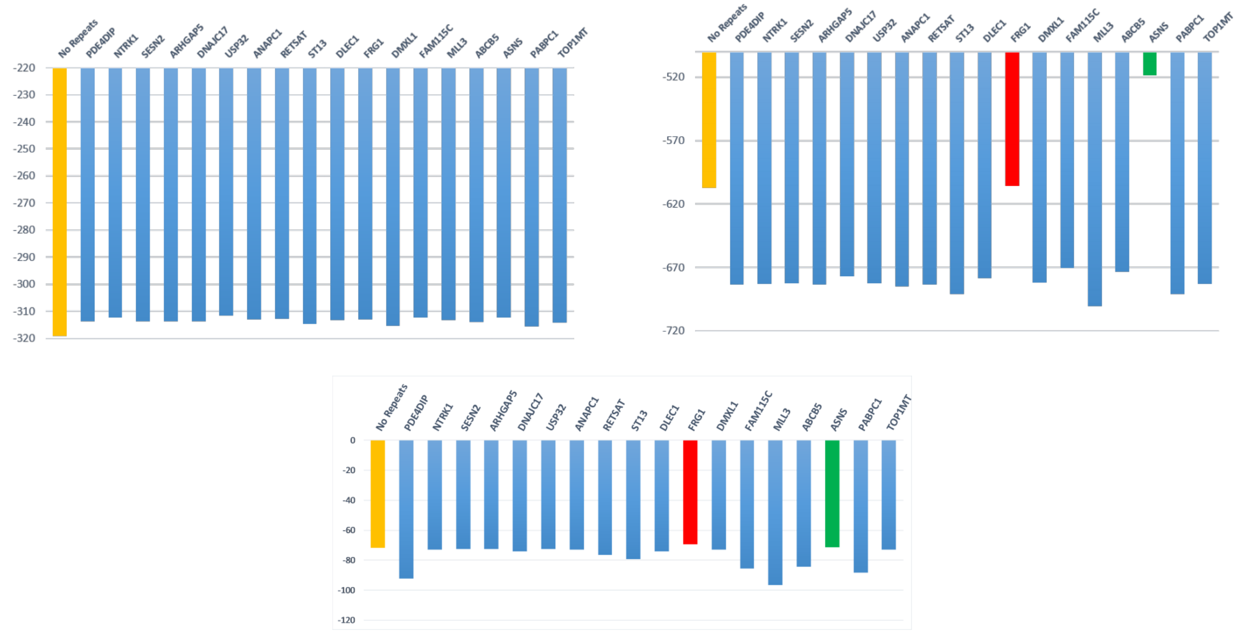


Figure 5.5 Log-likelihoods of trees with and without recurrent mutations for $JAK2$-negative myeloproliferative neoplasm. Upper left: log-likelihoods produced by infSCITE. Upper right: log-likelihoods produced by SCIFIL. Lower middle: log-likelihoods produced by MULAN.

that mutation rates change over the course of tumor evolution. For this purpose, we compared the single-rate model with the simplest model non-flat mutability landscape model that assumes two mutation rates. Following[83] and[158], the moldels were compared using Bayes factor BF[72], Akaike Information Criterion difference ΔAIC[3] and Bayesian Information Criterion difference ΔBIC[144]. In our case, these parameters are estimated as

$$BF = exp(L_2 - L_1), \quad \Delta AIC = 2(k_1 - k_2) + 2(L_2 - L_1), \quad \Delta BIC = (k_1 - k_2)\log(n) + 2(L_2 - L_1), \tag{5.18}$$

where n is the number of vertices of the tree T, L_1 and L_2 are maximum log-likelihoods of one-mutation and two-mutation models, and $k_1 = 1$ and $k_2 = 3$ are the numbers of parameters estimated by these models (the mutation rate in the former case and the two mutation rates and one rate change event in the latter case). Larger positive values of parameters indicate the preference of the two-rate model over the one-rate model. The models were compared for the perfect phylogeny T_{PF} and the two most probable recurrence trees T_{FRG} and T_{ASNS} for the $JAK2$-negative myeloproliferative neoplasm[68], as well as for the trees produced by SCITE[69] for lymphoblastic leukemia datasets[51]. For 3 out of 6 trees, the evidence for the variable mutation rate is considered as very strong (according to[72]), for 2 trees - as strong, and for one tree (T_{FRG}) the evidence for any of the models was not conclusive (Table 5.1).

Tree	T_{PF}[68]	T_{FRG}[68]	T_{ASNS}[68]	T_1[51]	T_2[51]	T_3[51]
BF	$5.010 \cdot 10^5$	$1.448 \cdot 10^1$	$2.587 \cdot 10^5$	$5.037 \cdot 10^3$	$3.882 \cdot 10^2$	$9.199 \cdot 10^1$
ΔAIC	26.249	5.3456	24.925	13.049	7.923	5.043
ΔBIC	20.358	-0.543	19.036	11.058	6.378	4.438

Table 5.1 Comparison of one-rate and two-rate models for experimental data

Mutability landscape of $JAK2$-negative myeloproliferative neoplasm. For two most likely recurrent trees T_{FRG} and T_{ASNS} identified above, more detailed analysis of their mutability landscapes using the general MULAN model demonstrated that in both cases the increase in the inferred mutation rates is likely associated with the emergence of mutation in the gene $SESN2$ (Fig 5.6). $SESN2$ is an antioxidant activated by p53, and it is indeed known that mutations in this gene may lead to genetic instability[68]. The structures of inferred mutability landscapes for these two trees also suggests that under the maximum parsimony criterion the first tree could be considered as more plausible than the second tree, where clones revert from higher to lower rates in one of its branches.

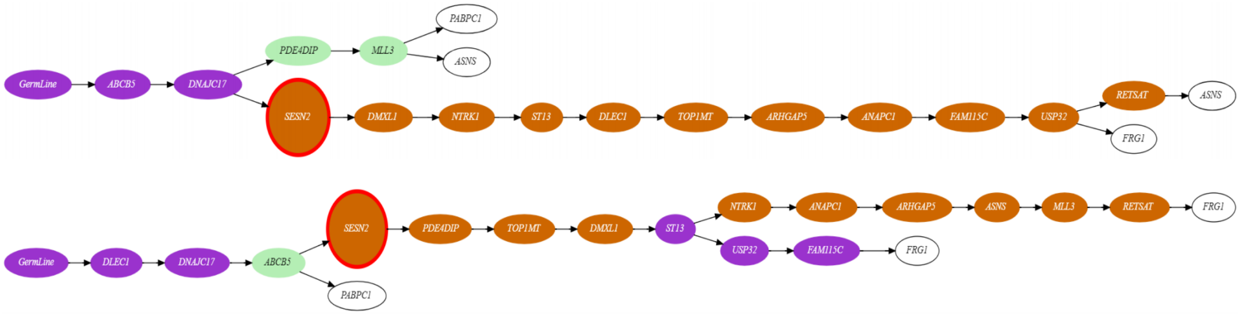


Figure 5.6 Two alternative mutation trees with the repeated mutations in $ASNS$ gene (top) and $FRG1$ gene (bottom), respectively. The different mutation rates are color-coded from green (low rate) to orange (high rate). The node corresponding to the mutation in $SESN2$ gene is highlighted. Leafs (not taken into account) are highlighted in white.

5.4 Discussion

Genomic instability is a typical characteristic of cancer cells, which may significantly contribute to tumor progression. Another paramount feature of cancer is an extremely high intra-tumor heterogeneity, with the genomic instability being one of the traits that may significantly differ between subclones. Thus, quantification of differential mutability and genomic instability for tumors may provide valuable information for understanding mechanisms of cancer progression and the design of personalized treatment strategies. The phenomenon of heterogeneous genomic instability could be geometrically represented by a concept of *mutability landscape*, which is the analog of the classical concept of the fitness landscape. Single-cell sequencing provides an unprecedented insight into intra-tumor heterogeneity and allows us to assess and study mutability landscapes of tumors on the finest possible level of individual subclones. In this paper, we presented likelihood-based methods for the inference of mutability landscapes of cancer subclonal populations from single-cell sequencing data. Most available methods for inference of differential mutation rates are tailored to the populations consisting of relatively distant genomes. In contrast, our method is specifically tailored to the specifics of cancer clone populations that consist of highly similar but distinct genomes and takes full advantage of the information about the structure and evolutionary history of the clonal population provided by single-cell sequencing. It infers mutation rates of subclones rather than individual genes, thus making it possible to use the obtained results to detect and quantify genomic interactions and epistasis. Instead, then considering all possible cancer phylogenies, MULAN uses as a starting point, a character-based mutation tree produced by other tools. This tree represents partial information about the order of the appearance of the clones. MULAN enriches this information by reconstructing orders of the appearance of sibling clones in the tree and uses it to infer mutation rates and clone appearance times. Thus, our methods can be used jointly with available tools for cancer tree inference from scSec data, such as SCITE[69], SiFit[187], SPhyR[43] and SCARLET[143], as well as from a combination of bulk and scSec data such as B-SCITE[99] and PhISCS[100]. The latter approach could be especially useful in the context of mutation clusters resolution. Indeed, MULAN assumes by default that every mutation results in a new subclone. However, scSec-based methods sometimes infer branches of mutations whose linear ordering cannot be resolved and group them into mutation clusters. Bulk data provides information about variant allele frequencies that allows inferring the temporal order of such mutations[99]. If such data is unavailable, ambiguities in clusters could be resolved arbitrarily, but the set of inferred mutation rates of clustered nodes should be interpreted as representing the whole subpopulation rather than individual subclones.

Our experiments demonstrated that the proposed approach allows for accurate inference of mutability landscapes and can be used for the analysis of the evolutionary history for real tumors. In particular, MULAN was able to detect a mutability increase event during the evolution of $JAK2$-Negative Myeloproliferative Neoplasm, that could be linked to the mutation in the gene with known associations with genetic instability. In addition, for several analyzed tumors the evolutionary signal produced by our mutability landscape model agreed with the signal produced by an independent fitness landscape model. This fact could be considered as an indication of the validity of both models.

There are several directions for the possible expansion of the proposed computational framework. Since mutation rates are the most important parameters for the inference, it could be beneficial to marginalize the likelihood over the remaining parameters. It may require the derivation of analytical expressions and/or accurate approximations for the marginalized likelihood that allows reducing its maximization to convex programming. Another direction is the development of the joint model for the inference of mutation and replication rates of cancer subclones. In this paper, we follow the common assumption of the standard molecular clock-based methods that do not consider population sizes. This assumption is usually justified, for example, using the neutral theory of molecular evolution[75,28], which is also applicable to cancer[26,176]. To take into account a wider range of evolutionary scenarios, a comprehensive framework incorporating replication rate and mutation rate diversity should be developed. One of advantages of such approach is its ability to utilize the observed frequencies of sequenced clones for the inference (for example, of mutation orders). Such utilization is not straightforward[157,147]: high frequency of a particular clone can be indicative of its earlier birth time or of its higher replication rate. To distinguish between these alternatives, an incorporation of a separate maximum likelihood framework is necessary. It potentially could be achieved, for example, by integrating MULAN with our previously published framework SCIFIL for the inference of cancer fitness landscapes[157]. Finally, MULAN was developed with targeted single-cell sequencing experiments in mind and it scales well for datasets typical for such settings. It is still scalable for whole-genome sequencing, if the mutation tree has not too many branching events. However, for more branching trees with thousands of vertices the scalability could become an issue. In that case, faster strategy for search in the space of mutation orderings should be considered.

REFERENCES

1. N. Aguse, Y. Qi, and M. El-Kebir. Summarizing the solution space in tumor phylogeny inference by multiple consensus trees. *Bioinformatics*, 35(14):i408–i416, 2019.

2. S. Ahn and H. Vikalo. abayesqr: A bayesian method for reconstruction of viral populations characterized by low diversity. In *International Conference on Research in Computational Molecular Biology*, pages 353–369. Springer, 2017.

3. H. Akaike. A new look at the statistical model identification. In *Selected Papers of Hirotugu Akaike*, pages 215–222. Springer, 1974.

4. S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. Basic local alignment search tool. *Journal of molecular biology*, 215(3):403–410, 1990.

5. A. Artyomenko, N. C. Wu, S. Mangul, E. Eskin, R. Sun, and A. Zelikovsky. Long single-molecule reads can resolve the complexity of the influenza virus composed of rare, closely related mutant variants. *Journal of Computational Biology*, 24(6):558–570, June 2017. doi: 10.1089/cmb.2016.0146. URL `https://doi.org/10.1089/cmb.2016.0146`.

6. A. Ashworth, C. J. Lord, and J. S. Reis-Filho. Genetic interactions in cancer progression and treatment. *Cell*, 145(1):30–38, 2011.

7. I. Astrovskaya, B. Tork, S. Mangul, K. Westbrooks, I. Măndoiu, P. Balfe, and A. Zelikovsky. Inferring viral quasispecies spectra from 454 pyrosequencing reads. *BMC bioinformatics*, 12 (Suppl 6):S1, 2011.

8. E. S. Azer, F. R. Mehrabadi, X. C. Li, S. Malikić, A. A. Schäffer, E. M. Gertz, C.-P. Day,

E. Pérez-Guijarro, K. Marie, M. P. Lee, et al. Phiscs-bnb: A fast branch and bound algorithm for the perfect tumor phylogeny reconstruction problem. *bioRxiv*, 2020.

9. J. A. Baaijens, A. Z. El Aabidine, E. Rivals, and A. Schönhuth. De novo assembly of viral quasispecies using overlap graphs. *Genome Research*, 27(5):835–848, 2017.
10. D. Bankwitz, E. Steinmann, J. Bitzegeio, S. Ciesek, M. Friesland, E. Herrmann, M. B. Zeisel, T. F. Baumert, Z.-y. Keck, S. K. Foung, et al. Hepatitis c virus hypervariable region 1 modulates receptor interactions, conceals the cd81 binding site, and protects conserved neutralizing epitopes. *Journal of virology*, 84(11):5751–5763, 2010.
11. S. Barik, S. Das, and H. Vikalo. Viral quasispecies reconstruction via correlation clustering. *bioRxiv*, page 096768, 2016.
12. M. Baron, A. Veres, S. L. Wolock, A. L. Faust, R. Gaujoux, A. Vetere, J. H. Ryu, B. K. Wagner, S. S. Shen-Orr, A. M. Klein, et al. A single-cell transcriptomic map of the human and mouse pancreas reveals inter-and intra-cell population structure. *Cell systems*, 3(4):346–360, 2016.
13. N. Beerenwinkel, T. Sing, T. Lengauer, J. Rahnenfuehrer, and K. Roomp. Computational methods for the design of effective therapies against drug resistant HIV strains. *Bioinformatics*, 21:3943–3950, 2005.
14. N. Beerenwinkel, T. Antal, D. Dingli, A. Traulsen, K. W. Kinzler, V. E. Velculescu, B. Vogelstein, and M. A. Nowak. Genetic progression and the waiting time to cancer. *PLoS computational biology*, 3(11):e225, 2007.
15. S. Benidt and D. Nettleton. Simseq: a nonparametric approach to simulation of rna-sequence

datasets. *Bioinformatics*, 31(13):2131–2140, 2015. doi: 10.1093/bioinformatics/btv124. URL `+http://dx.doi.org/10.1093/bioinformatics/btv124`.

16. R. Bonavia, W. K. Cavenee, F. B. Furnari, et al. Heterogeneity maintenance in glioblastoma: a social network. *Cancer research*, 71(12):4055–4060, 2011.

17. V. Boskova and T. Stadler. PIQMEE: Bayesian phylodynamic method for analysis of large datasets with duplicate sequences. *Molecular Biology and Evolution*, June 2020. doi: 10.1093/molbev/msaa136. URL `https://doi.org/10.1093/molbev/msaa136`.

18. B. Boucher and S. Jenna. Genetic interaction networks: better understand to better predict. *Frontiers in genetics*, 4:290, 2013.

19. S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

20. I. Bozic, T. Antal, H. Ohtsuki, H. Carter, D. Kim, S. Chen, R. Karchin, K. W. Kinzler, B. Vogelstein, and M. A. Nowak. Accumulation of driver and passenger mutations during tumor progression. *Proceedings of the National Academy of Sciences*, 107(43):18545–18550, 2010.

21. A. Z. Broder, S. C. Glassman, M. S. Manasse, and G. Zweig. Syntactic clustering of the web. *Computer Networks and ISDN Systems*, 29(8-13):1157–1166, 1997.

22. L. Bromham and D. Penny. The modern molecular clock. *Nature Reviews Genetics*, 4(3): 216, 2003.

23. C. Bron and J. Kerbosch. Algorithm 457: Finding all cliques of an undirected graph. *Commun. ACM*, 16(9):575–577, Sept. 1973. ISSN 0001-0782. doi: 10.1145/362342.362367. URL `http://doi.acm.org/10.1145/362342.362367`.

24. D. S. Campo, P. Skums, Z. Dimitrova, G. Vaughan, J. C. Forbi, C.-G. Teo, Y. Khudyakov, and D. T. Lau. Drug resistance of a viral population and its individual intrahost variants during the first 48 hours of therapy. *Clinical Pharmacology & Therapeutics*, 95(6):627–635, 2014.

25. D. S. Campo, G.-L. Xia, Z. Dimitrova, Y. Lin, J. C. Forbi, L. Ganova-Raeva, L. Punkova, S. Ramachandran, H. Thai, P. Skums, et al. Accurate genetic detection of hepatitis c virus transmissions in outbreak settings. *Journal of Infectious Diseases*, 213(6):957–965, 2016.

26. V. L. Cannataro and J. P. Townsend. Neutral theory and the somatic evolution of cancer. *Molecular biology and evolution*, 35(6):1308–1315, 2018.

27. M. R. Capobianchi, E. Giombini, and G. Rozera. Next-generation sequencing technology in clinical virology, 2013.

28. L. Chao and D. E. Carr. The molecular clock and the relationship between population size and generation time. *Evolution*, 47(2):688–690, 1993.

29. G. S. Charames and B. Bapat. Genomic instability and cancer. *Current molecular medicine*, 3(7):589–596, 2003.

30. S. Ciccolella, M. S. Gomez, M. Patterson, G. Della Vedova, I. Hajirasouliha, and P. Bonizzoni. Inferring cancer progression from single cell sequencing while allowing loss of mutations. *bioRxiv*, page 268243, 2018.

31. K. Crona, A. Gavryushkin, D. Greene, and N. Beerenwinkel. Inferring genetic interactions from comparative fitness data. *eLife*, 6:e28629, 2017.

32. M. Cruz-Rivera, J. C. Forbi, L. H. T. Yamasaki, C. A. Vazquez-Chacon, A. Martinez-Guarneros, J. C. Carpio-Pedroza, A. Escobar-Gutiérrez, K. Ruiz-Tovar, S. Fonseca-

Coronado, and G. Vaughan. Molecular epidemiology of viral diseases in the era of next generation sequencing. *J. Clin. Virol.*, 57(4):378–380, Aug. 2013.

33. L. Cuypers, G. Li, P. Libin, S. Piampongsant, A.-M. Vandamme, and K. Theys. Genetic diversity and selective pressure in hepatitis c virus genotypes 1–6: significance for direct-acting antiviral treatment and drug resistance. *Viruses*, 7(9):5018–5039, 2015.

34. A. Davis, R. Gao, and N. Navin. Tumor evolution: Linear, branching, neutral or punctuated? *Biochimica et Biophysica Acta (BBA)-Reviews on Cancer*, 1867(2):151–161, 2017.

35. K. Deforche, R. Camacho, K. Van Laethem, P. Lemey, A. Rambaut, Y. Moreau, and A.-M. Vandamme. Estimation of an in vivo fitness landscape experienced by hiv-1 under drug selective pressure useful for prediction of drug resistance evolution during treatment. *Bioinformatics*, 24(1):34–41, 2007.

36. A. Dolgui, V. Gordon, and V. Strusevich. Single machine scheduling with precedence constraints and positionally dependent processing times. *Computers & Operations Research*, 39 (6):1218–1224, 2012.

37. E. Domingo, J. Sheldon, and C. Perales. Viral quasispecies evolution. *Microbiology and Molecular Biology Reviews*, 76(2):159–216, 2012.

38. M. Döring, J. Büch, G. Friedrich, A. Pironti, P. Kalaghatgi, E. Knops, E. Heger, M. Obermeier, M. Däumer, A. Thielen, R. Kaiser, T. Lengauer, and N. Pfeifer. geno2pheno[ngs-freq]: a genotypic interpretation system for identifying viral drug resistance using next-generation sequencing data. *Nucleic Acids Research*, 46(W1):W271–W277, May 2018. doi: 10.1093/nar/gky349. URL `https://doi.org/10.1093/nar/gky349`.

39. N. G. Douek DC, Kwong PD. The rational design of an AIDS vaccine. *Cell*, 124:677–681, 2006.

40. M. A. Doyle, J. Li, K. Doig, A. Fellowes, and S. Q. Wong. Studying cancer genomics through next-generation dna sequencing and bioinformatics. *Clinical Bioinformatics*, pages 83–98, 2014.

41. A. J. Drummond, S. Y. Ho, M. J. Phillips, and A. Rambaut. Relaxed phylogenetics and dating with confidence. *PLoS biology*, 4(5):e88, 2006.

42. M. Eigen, J. McCaskill, and P. Schuster. The molecular quasi-species. *Advances in chemical physics*, 75:149–263, 1989.

43. M. El-Kebir. Sphyr: tumor phylogeny estimation from single-cell sequencing data under loss and error. *Bioinformatics*, 34(17):i671–i679, 2018.

44. M. El-Kebir, L. Oesper, H. Acheson-Field, and B. J. Raphael. Reconstruction of clonal trees and tumor composition from multi-sample sequencing data. *Bioinformatics*, 31(12):i62–i70, 2015.

45. M. El-Kebir, G. Satas, and B. J. Raphael. Inferring parsimonious migration histories for metastatic cancers. *Nature Genetics*, 2:5, 2018.

46. A. Eliseev, K. M. Gibson, P. Avdeyev, D. Novik, M. L. Bendall, M. Pérez-Losada, N. Alexeev, and K. A. Crandall. Evaluation of haplotype callers for next-generation sequencing of viruses. *Infection, Genetics and Evolution*, 82:104277, Aug. 2020. doi: 10.1016/j.meegid.2020.104277. URL `https://doi.org/10.1016/j.meegid.2020.104277`.

47. A. L. Ferguson, J. K. Mann, S. Omarjee, T. Ndung'u, B. D. Walker, and A. K. Chakraborty.

Translating hiv sequences into quantitative fitness landscapes predicts viral vulnerabilities for rational immunogen design. *Immunity*, 38(3):606–617, 2013.

48. C. for Disease Control, Prevention, et al. Diagnoses of hiv infection in the united states and dependent areas, 2015. *HIV Surveillance Report*, 27:1–114, 2016.

49. B. Gaschen, J. Taylor, K. Yusim, B. Foley, and F. Gao. Diversity considerations in HIV-1 vaccine selection. *Science*, 296:2354–2360, 2002.

50. S. Gavrilets. *Fitness landscapes and the origin of species (MPB-41)*, volume 41. Princeton University Press, 2004.

51. C. Gawad, W. Koh, and S. R. Quake. Dissecting the clonal origins of childhood acute lymphoblastic leukemia by single-cell genomics. *Proceedings of the National Academy of Sciences*, 111(50):17947–17952, 2014.

52. E. Gerasimov. *Analysis of NGS Data from Immune Response and Viral Samples*. PhD thesis, Georgia State University, 2017.

53. F. D. Giallonardo, A. Töpfer, M. Rey, S. Prabhakaran, Y. Duport, C. Leemann, S. Schmutz, N. K. Campbell, B. Joos, M. R. Lecca, A. Patrignani, M. Däumer, C. Beisel, P. Rusert, A. Trkola, H. F. Günthard, V. Roth, N. Beerenwinkel, and K. J. Metzner. Full-length haplotype reconstruction to infer the structure of heterogeneous virus populations. *Nucleic Acids Research*, 42(14):e115–e115, June 2014. doi: 10.1093/nar/gku537. URL `https://doi.org/10.1093/nar/gku537`.

54. F. D. Giallonardo, A. Töpfer, M. Rey, S. Prabhakaran, Y. Duport, C. Leemann, S. Schmutz, N. K. Campbell, B. Joos, M. R. Lecca, A. Patrignani, M. Däumer, C. Beisel, P. Rusert,

A. Trkola, H. F. Günthard, V. Roth, N. Beerenwinkel, and K. J. Metzner. Full-length haplotype reconstruction to infer the structure of heterogeneous virus populations. *Nucleic Acids Research*, 42(14):e115, 2014. doi: 10.1093/nar/gku537. URL +http://dx.doi.org/10.1093/nar/gku537.

55. A. Gionis, P. Indyk, R. Motwani, et al. Similarity search in high dimensions via hashing. In *VLDB*, volume 99, pages 518–529, 1999.

56. O. Glebova, S. Knyazev, A. Melnick, A. Artyomenko, Y. Khudyakov, A. Zelikovsky, and P. Skums. Computational inference of transmission characteristics between viral populations. *BMC Bioinformatics*, accepted.

57. W. M. Grady. Genomic instability and colon cancer. *Cancer and metastasis reviews*, 23(1-2): 11–27, 2004.

58. M. Greaves. Nothing in cancer makes sense except.... *BMC biology*, 16(1):1–8, 2018.

59. M. Greaves and C. C. Maley. Clonal evolution in cancer. *Nature*, 481(7381):306, 2012.

60. T. Griebel, B. Zacher, P. Ribeca, E. Raineri, V. Lacroix, R. Guigó, and M. Sammeth. Modelling and simulating generic RNA-seq experiments with the flux simulator. *Nucleic Acids Research*, 40(20):10073–10083, Sept. 2012. doi: 10.1093/nar/gks666. URL https://doi.org/10.1093/nar/gks666.

61. D. Gusfield. *Algorithms on strings, trees and sequences: computer science and computational biology*. Cambridge university press, New York, NY, USA, 1997.

62. M. Gwinn, D. MacCannell, and G. L. Armstrong. Next-Generation sequencing of infectious pathogens, 2019.

63. B. Hajarizadeh, J. Grebely, and G. J. Dore. Epidemiology and natural history of hcv infection. *Nature Reviews Gastroenterology and Hepatology*, 10(9):553–562, 2013.

64. D. Hao, L. Wang, and L.-j. Di. Distinct mutation accumulation rates among tissues determine the variation in cancer risk. *Scientific Reports*, 6:19458, 2016.

65. T. Hinkley, J. Martins, C. Chappey, M. Haddad, E. Stawiski, J. M. Whitcomb, C. J. Petropoulos, and S. Bonhoeffer. A systems analysis of mutational effects in hiv-1 protease and reverse transcriptase. *Nature genetics*, 43(5):487, 2011.

66. J. Holland, J. De La Torre, and D. Steinhauer. RNA virus populations as quasispecies. *Curr Top Microbiol Immunol*, 176:1–20, 1992.

67. S.-R. Hosseini, R. Diaz-Uriarte, F. Markowetz, and N. Beerenwinkel. Estimating the predictability of cancer evolution. *Bioinformatics*, 35(14):i389–i397, 07 2019. ISSN 1367-4803. doi: 10.1093/bioinformatics/btz332. URL `https://doi.org/10.1093/bioinformatics/btz332`.

68. Y. Hou, L. Song, P. Zhu, B. Zhang, Y. Tao, X. Xu, F. Li, K. Wu, J. Liang, D. Shao, et al. Single-cell exome sequencing and monoclonal evolution of a jak2-negative myeloproliferative neoplasm. *Cell*, 148(5):873–885, 2012.

69. K. Jahn, J. Kuipers, and N. Beerenwinkel. Tree inference for single-cell data. *Genome biology*, 17(1):86, 2016.

70. W. Jiao, S. Vembu, A. G. Deshwar, L. Stein, and Q. Morris. Inferring clonal evolution of tumors from single nucleotide somatic mutations. *BMC bioinformatics*, 15(1):35, 2014.

71. S. Jones, W.-d. Chen, G. Parmigiani, F. Diehl, N. Beerenwinkel, T. Antal, A. Traulsen, M. A.

Nowak, C. Siegel, V. E. Velculescu, et al. Comparative lesion sequencing provides insights into tumor evolution. *Proceedings of the National Academy of Sciences*, 105(11):4283–4288, 2008.

72. R. E. Kass and A. E. Raftery. Bayes factors. *Journal of the american statistical association*, 90(430):773–795, 1995.

73. B. W. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. *Bell system technical journal*, 49(2):291–307, 1970.

74. P. H. Kilmarx. Global epidemiology of hiv. *Current Opinion in HIV and AIDS*, 4(4):240–246, 2009.

75. M. Kimura. *The neutral theory of molecular evolution*. Cambridge University Press, 1983.

76. M. Kimura and T. Maruyama. The mutational load with epistatic gene interactions in fitness. *Genetics*, 54(6):1337, 1966.

77. D. E. Kireev, A. E. Lopatukhin, A. V. Murzakova, E. V. Pimkina, A. S. Speranskaya, A. D. Neverov, G. G. Fedonin, Y. S. Fantin, and G. A. Shipulin. Evaluating the accuracy and sensitivity of detecting minority HIV-1 populations by Illumina next-generation sequencing. *J. Virol. Methods*, 261:40–45, 11 2018.

78. S. Knyazev, L. Hughes, P. Skums, and A. Zelikovsky. Epidemiological data analysis of viral quasispecies in the next-generation sequencing era. *Briefings in Bioinformatics*, June 2020. doi: 10.1093/bib/bbaa101. URL `https://doi.org/10.1093/bib/bbaa101`.

79. S. Knyazev, V. Tsyvina, A. Shankar, A. Melnyk, A. Artyomenko, T. Malygina, Y. B. Porozov, E. M. Campbell, W. M. Switzer, P. Skums, et al. Accurate assembly of minority viral

haplotypes from next-generation sequencing through efficient noise reduction. *Nucleic acids research*, 49(17):e102–e102, 2021.

80. V. Körber and T. Höfer. Inferring growth and genetic evolution of tumors from genome sequences. *Current Opinion in Systems Biology*, 2019.
81. J. Kováč. Complexity of the path avoiding forbidden pairs problem revisited. *Discrete Appl. Math.*, 161(10-11):1506–1512, July 2013. ISSN 0166-218X. doi: 10.1016/j.dam.2012.12.022. URL `http://dx.doi.org/10.1016/j.dam.2012.12.022`.
82. J. Kuipers, K. Jahn, and N. Beerenwinkel. Advances in understanding tumour evolution through single-cell sequencing. *Biochimica et Biophysica Acta (BBA)-Reviews on Cancer*, 1867(2):127–138, 2017.
83. J. Kuipers, K. Jahn, B. J. Raphael, and N. Beerenwinkel. Single-cell sequencing data reveal widespread recurrence and loss of mutational hits in the life histories of tumors. *Genome research*, 2017.
84. D. Laehnemann, J. Köster, E. Szcureck, D. McCarthy, S. C. Hicks, M. D. Robinson, C. A. Vallejos, N. Beerenwinkel, K. R. Campbell, A. Mahfouz, et al. 12 grand challenges in single-cell data science. Technical report, PeerJ Preprints, 2019.
85. D. A. Landau, S. L. Carter, P. Stojanov, A. McKenna, K. Stevenson, M. S. Lawrence, C. Sougnez, C. Stewart, A. Sivachenko, L. Wang, et al. Evolution and impact of subclonal mutations in chronic lymphocytic leukemia. *Cell*, 152(4):714–726, 2013.
86. M. L. Leung, A. Davis, R. Gao, A. Casasent, Y. Wang, E. Sei, E. Sanchez, D. Maru, S. Kopetz, and N. E. Navin. Single cell dna sequencing reveals a late-dissemination model

in metastatic colorectal cancer. *Genome research*, pages gr–209973, 2017.

87. E. Levina and P. Bickel. The earthmover's distance is the mallows distance: Some insights from statistics. *Proceedings of ICCV 2001*, pages 251–256, 2001.

88. C. Li, B. Wang, and X. Yang. Vgram: Improving performance of approximate queries on string collections using variable-length grams. In *Proceedings of the 33rd international conference on Very large data bases*, pages 303–314. VLDB Endowment, 2007.

89. A. Longmire, S. Sims, I. Rytsareva, D. Campo Rendon, Z. Dimitrova, et al. Ghost: Global health outbreak and surveillance technology. *BMC Bioinformatics*, accepted.

90. R. Lozano, M. Naghavi, K. Foreman, S. Lim, K. Shibuya, V. Aboyans, J. Abraham, T. Adair, R. Aggarwal, S. Y. Ahn, et al. Global and regional mortality from 235 causes of death for 20 age groups in 1990 and 2010: a systematic analysis for the global burden of disease study 2010. *The lancet*, 380(9859):2095–2128, 2012.

91. X. Lu, P. R. Kensche, M. A. Huynen, and R. A. Notebaart. Genome evolution predicts genetic interactions in protein complexes and reveals cancer drug targets. *Nature communications*, 4:2124, 2013.

92. J. Luo, N. L. Solimini, and S. J. Elledge. Principles of cancer therapy: oncogene and non-oncogene addiction. *Cell*, 136(5):823–837, 2009.

93. B. Ma, J. Tromp, and M. Li. Patternhunter: faster and more sensitive homology search. *Bioinformatics*, 18(3):440–445, 2002.

94. J. Ma, C. Dykes, T. Wu, Y. Huang, L. Demeter, and H. Wu. vfitness: a web-based computing tool for improving estimation of in vitro hiv-1 fitness experiments. *BMC bioinformatics*, 11

(1):261, 2010.

95. A. R. Macalalad, M. C. Zody, P. Charlebois, N. J. Lennon, R. M. Newman, C. M. Malboeuf, E. M. Ryan, C. L. Boutwell, K. A. Power, D. E. Brackney, et al. Highly sensitive and specific detection of rare variants in mixed viral populations from massively parallel sequence data. *PLoS computational biology*, 8(3):e1002417, 2012.

96. A. Magen, A. D. Sahu, J. S. Lee, M. Sharmin, A. Lugo, J. S. Gutkind, A. A. Schäffer, E. Ruppin, and S. Hannenhalli. Beyond synthetic lethality: Charting the landscape of pairwise gene expression states associated with survival in cancer. *Cell reports*, 28(4):938–948, 2019.

97. S. Malikic, A. W. McPherson, N. Donmez, and C. S. Sahinalp. Clonality inference in multiple tumor samples using phylogeny. *Bioinformatics*, 31(9):1349–1356, 2015.

98. S. Malikic, K. Jahn, J. Kuipers, C. Sahinalp, and N. Beerenwinkel. Integrative inference of subclonal tumour evolution from single-cell and bulk sequencing data. *bioRxiv*, page 234914, 2017.

99. S. Malikic, K. Jahn, J. Kuipers, S. C. Sahinalp, and N. Beerenwinkel. Integrative inference of subclonal tumour evolution from single-cell and bulk sequencing data. *Nature communications*, 10(1):2750, 2019.

100. S. Malikic, F. R. Mehrabadi, S. Ciccolella, M. K. Rahman, C. Ricketts, E. Haghshenas, D. Seidman, F. Hach, I. Hajirasouliha, and S. C. Sahinalp. Phiscs: a combinatorial approach for subperfect tumor phylogeny reconstruction via integrative use of single-cell and bulk sequencing data. *Genome Research*, 29(11):1860–1877, 2019.

101. C. L. Mallows. A note on asymptotic joint normality. *Annals of Mathematical Statistics*, 43

(2):508–515, 1972.

102. N. Mancuso, B. Tork, P. Skums, L. Ganova-Raeva, I. Măndoiu, and A. Zelikovsky. Reconstructing viral quasispecies from ngs amplicon reads. *In silico biology*, 11(5):237–249, 2011.

103. S. Mangul, N. C. Wu, N. Mancuso, A. Zelikovsky, R. Sun, and E. Eskin. Accurate viral population assembly from ultra-deep sequencing data. *Bioinformatics*, 30(12):i329–i337, 2014.

104. S. Mangul, L. S. Martin, B. L. Hill, A. K.-M. Lam, M. G. Distler, A. Zelikovsky, E. Eskin, and J. Flint. Systematic benchmarking of omics computational tools. *Nature Communications*, 10(1), Mar. 2019. doi: 10.1038/s41467-019-09406-4. URL `https://doi.org/10.1038/s41467-019-09406-4`.

105. M. Martell, J. Esteban, J. Quer, J. Genesca, A. Weiner, R. Esteban, J. Guardia, and J. Gomez. Hepatitis c virus (hcv) circulates as a population of different but closely related genomes: quasispecies nature of hcv genome distribution. *Journal of Virology, 66*, pages 3225–3229, 1992.

106. D. Matlak and E. Szczurek. Epistasis in genomic and survival data of cancer patients. *PLoS computational biology*, 13(7):e1005626, 2017.

107. D. P. McLornan, A. List, and G. J. Mufti. Applying synthetic lethality for the selective targeting of cancer. *New England Journal of Medicine*, 371(18):1725–1735, 2014.

108. P. Medvedev, E. Scott, B. Kakaradov, and P. Pevzner. Error correction of high-throughput sequencing datasets with non-uniform coverage. *Bioinformatics*, 27(13):i137–i141, 2011.

109. A. Melnyk, S. Knyazev, F. Vannberg, L. Bunimovich, P. Skums, and A. Zelikovsky. Using earth mover's distance for viral outbreak investigations. *bioRxiv*, May 2019. doi: 10.1101/628859. URL `https://doi.org/10.1101/628859`.

110. L. W. Meredith, W. L. Hamilton, B. Warne, C. J. Houldcroft, M. Hosmillo, A. S. Jahun, M. D. Curran, S. Parmar, L. G. Caller, S. L. Caddy, et al. Rapid implementation of sars-cov-2 sequencing to investigate cases of health-care associated covid-19: a prospective genomic surveillance study. *The Lancet infectious diseases*, 20(11):1263–1272, 2020.

111. L. M. Merlo, N. A. Shah, X. Li, P. L. Blount, T. L. Vaughan, B. J. Reid, and C. C. Maley. A comprehensive survey of clonal diversity measures in barrett's esophagus as biomarkers of progression to esophageal adenocarcinoma. *Cancer prevention research*, 3(11):1388–1397, 2010.

112. K. Mitchell, J. J. Brito, I. Mandric, Q. Wu, S. Knyazev, S. Chang, L. S. Martin, A. Karlsberg, E. Gerasimov, R. Littman, B. L. Hill, N. C. Wu, H. T. Yang, K. Hsieh, L. Chen, E. Littman, T. Shabani, G. Enik, D. Yao, R. Sun, J. Schroeder, E. Eskin, A. Zelikovsky, P. Skums, M. Pop, and S. Mangul. Benchmarking of computational error-correction methods for next-generation sequencing data. *Genome Biology*, 21(1), Mar. 2020. doi: 10.1186/s13059-020-01988-3. URL `https://doi.org/10.1186/s13059-020-01988-3`.

113. P. A. Moran. Global stability of genetic systems governed by mutation and selection. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 80, pages 331–336. Cambridge University Press, 1976.

114. S. Mukherjee, Y. Zhang, J. Fan, G. Seelig, and S. Kannan. Scalable preprocessing

for sparse scrna-seq data exploiting prior knowledge. *Bioinformatics*, 34(13):i124–i132, 2018. doi: 10.1093/bioinformatics/bty293. URL http://dx.doi.org/10.1093/bioinformatics/bty293.

115. B. B. O. Munnink, D. F. Nieuwenhuijse, M. Stein, Á. O'Toole, M. Haverkate, M. Mollers, S. K. Kamga, C. Schapendonk, M. Pronk, P. Lexmond, et al. Rapid sars-cov-2 whole-genome sequencing and analysis for informed public health decision-making in the netherlands. *Nature medicine*, 26(9):1405–1410, 2020.

116. M. Nicolae, S. Mangul, I. Mandoiu, and A. Zelikovsky. Estimation of alternative splicing isoform frequencies from rna-seq data. *Algorithms for Molecular Biology*, 6:9, 2011. URL http://www.almob.org/content/6/1/9.

117. A. Nijenhuis and H. S. Wilf. *Combinatorial algorithms: for computers and calculators*. Elsevier, 2014.

118. S. I. Nikolenko, A. I. Korobeynikov, and M. A. Alekseyev. Bayeshammer: Bayesian clustering for error correction in single-cell sequencing. *BMC genomics*, 14(1):S7, 2013.

119. J. Noorbakhsh and J. H. Chuang. Uncertainties in tumor allele frequencies limit power to infer evolutionary pressures. *Nature genetics*, 49(9):1288, 2017.

120. M. A. Nowak. *Evolutionary dynamics*. Harvard University Press, 2006.

121. M. A. Nowak and R. M. May. Virus dynamics, 2000.

122. Office of the Commissioner. FDA authorizes marketing of first next-generation sequencing test for detecting HIV-1 drug resistance mutations. https://www.fda.gov/news-events/press-announcements/

fda-authorizes-marketing-first-next-generation-sequencing-test-detectin May 2019. Accessed: 2019-12-28.

123. N. J. O'Neil, M. L. Bailey, and P. Hieter. Synthetic lethality and cancer. *Nature Reviews Genetics*, 18(10):613–623, 2017.

124. J.-M. Pawlotsky, M. Pellerin, M. Bouvier, F. Roudot-Thoraval, C.-J. Soussy, and D. Dhumeaux. Genetic complexity of the hypervariable region 1 (hvr1) of hepatitis c virus (hcv): Influence on the. *Journal of medical virology*, 54:256–264, 1998.

125. P. Peterlongo, G. A. T. Sacomoto, A. P. do Lago, N. Pisanti, and M.-F. Sagot. Lossless filter for multiple repeats with bounded edit distance. *Algorithms for Molecular Biology*, 4(1):3, Jan 2009. ISSN 1748-7188. doi: 10.1186/1748-7188-4-3. URL https://doi.org/10.1186/1748-7188-4-3.

126. P. J. Peters, P. Pontones, K. W. Hoover, M. R. Patel, R. R. Galang, J. Shields, S. J. Blosser, M. W. Spiller, B. Combs, W. M. Switzer, et al. Hiv infection linked to injection use of oxymorphone in indiana, 2014–2015. *New England Journal of Medicine*, 375(3):229–239, 2016.

127. J. A. Polonsky, A. Baidjoe, Z. N. Kamvar, A. Cori, K. Durski, W. J. Edmunds, R. M. Eggo, S. Funk, L. Kaiser, P. Keating, O. l. P. de Waroux, M. Marks, P. Moraga, O. Morgan, P. Nouvellet, R. Ratnayake, C. H. Roberts, J. Whitworth, and T. Jombart. Outbreak analytics: a developing data science for informing the response to emerging pathogens. *Philos. Trans. R. Soc. Lond. B Biol. Sci.*, 374(1776):20180276, July 2019.

128. S. Prabhakaran, M. Rey, O. Zagordi, N. Beerenwinkel, and V. Roth. HIV haplotype in-

ference using a propagating Dirichlet process mixture model. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 11(1):182–191, 2014.

129. M. C. Prosperi and M. Salemi. Qure: software for viral quasispecies reconstruction from next-generation sequencing data. *Bioinformatics*, 28(1):132–133, 2012.

130. O. G. Pybus. Model selection and the molecular clock. *PLoS Biology*, 4(5):e151, 2006.

131. J. Qin, W. Wang, Y. Lu, C. Xiao, and X. Lin. Efficient exact edit similarity query processing with the asymmetric signature scheme. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*, pages 1033–1044. ACM, 2011.

132. M. A. Quail, M. Smith, P. Coupland, T. D. Otto, S. R. Harris, T. R. Connor, A. Bertoni, H. P. Swerdlow, and Y. Gu. A tale of three next generation sequencing platforms: comparison of ion torrent, pacific biosciences and illumina miseq sequencers. *BMC genomics*, 13(1):341, 2012.

133. P. Rathert, M. Roth, T. Neumann, F. Muerdter, J.-S. Roe, M. Muhar, S. Deswal, S. Cerny-Reiterer, B. Peter, J. Jude, et al. Transcriptional plasticity promotes primary and acquired resistance to bet inhibition. *Nature*, 525(7570):543, 2015.

134. S.-Y. Rhee, T. Liu, S. Holmes, and R. Shafer. HIV-1 subtype B protease and reverse transcriptase amino acid covariation. *PLoS Comput Biol*, 3:e87, 2007.

135. F. Rodriguez-Frias, M. Buti, D. Tabernero, and M. Homs. Quasispecies structure, cornerstone of hepatitis b virus infection: mass sequencing approach. *World J Gastroenterol*, 19(41): 6995–7023, 2013.

136. I. B. Rogozin, Y. I. Pavlov, A. Goncearenco, S. De, A. G. Lada, E. Poliakov, A. R. Panchenko,

and D. N. Cooper. Mutational signatures and mutable motifs in cancer genomes. *Briefings in bioinformatics*, 19(6):1085–1101, 2017.

137. E. M. Ross and F. Markowetz. Onconem: inferring tumor evolution from single-cell sequencing data. *Genome biology*, 17(1):69, 2016.

138. S. Rosset. Efficient inference on known phylogenetic trees using poisson regression. *Bioinformatics*, 23(2):e142–e147, 2007.

139. A. Routh, M. W. Chang, J. F. Okulicz, J. E. Johnson, and B. E. Torbett. Covama: Co-variation mapper for disequilibrium analysis of mutant loci in viral populations using next-generation sequence data. *Methods*, 91:40–47, 2015.

140. I. Rytsareva, D. S. Campo, Y. Zheng, S. Sims, S. V. Thankachan, C. Tetik, J. Chirag, S. P. Chockalingam, A. Sue, S. Aluru, et al. Efficient detection of viral transmissions with next-generation sequencing data. *BMC genomics*, 18(4):372, 2017.

141. A. D. Sahu, J. S. Lee, Z. Wang, G. Zhang, R. Iglesias-Bartolome, T. Tian, Z. Wei, B. Miao, N. U. Nair, O. Ponomarova, et al. Genome-wide prediction of synthetic rescue mediators of resistance to targeted and immunotherapy. *Molecular systems biology*, 15(3), 2019.

142. M. J. Sanderson. A nonparametric approach to estimating divergence times in the absence of rate constancy. *Molecular biology and evolution*, 14(12):1218–1231, 1997.

143. G. Satas, S. Zaccaria, G. Mon, and B. J. Raphael. Scarlet: Single-cell tumor phylogeny inference with copy-number constrained mutation losses. *Cell Systems*, 10(4):323–332, 2020.

144. G. Schwarz et al. Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464, 1978.

145. M. R. Segal, J. D. Barbour, and R. M. Grant. Relating hiv-1 sequence variation to replication capacity via trees and forests. *Statistical applications in genetics and molecular biology*, 3 (1):1–18, 2004.

146. D. Seifert, F. Di Giallonardo, K. J. Metzner, H. F. Günthard, and N. Beerenwinkel. A framework for inferring fitness landscapes of patient-derived viruses using quasispecies theory. *Genetics*, pages genetics–114, 2014.

147. D. Seifert, F. Di Giallonardo, K. J. Metzner, H. F. Günthard, and N. Beerenwinkel. A framework for inferring fitness landscapes of patient-derived viruses using quasispecies theory. *Genetics*, 199(1):191–203, 2015.

148. Z. Shen, Y. Xiao, L. Kang, W. Ma, L. Shi, L. Zhang, Z. Zhou, J. Yang, J. Zhong, D. Yang, L. Guo, G. Zhang, H. Li, Y. Xu, M. Chen, Z. Gao, J. Wang, L. Ren, and M. Li. Genomic diversity of severe acute respiratory syndrome–coronavirus 2 in patients with coronavirus disease 2019. *Clinical Infectious Diseases*, Mar. 2020. doi: 10.1093/cid/ciaa203. URL `https://doi.org/10.1093/cid/ciaa203`.

149. A. Shlemov, S. Bankevich, A. Bzikadze, M. A. Turchaninova, Y. Safonova, and P. A. Pevzner. Reconstructing antibody repertoires from error-prone immunosequencing datasets. In *Research in Computational Molecular Biology*, page 396. Springer, 2017.

150. R. L. Siegel, D. Miller Kimberly, and A. Jemal. Cancer statistics, 2018. *CA: a cancer journal for clinicians*, 68(1):7–30, 2018.

151. P. Skums, Z. Dimitrova, D. S. Campo, G. Vaughan, L. Rossi, J. C. Forbi, J. Yokosawa, A. Zelikovsky, and Y. Khudyakov. Efficient error correction for next-generation sequencing

of viral amplicons. *BMC Bioinformatics*, 13(S-10):S6, 2012.

152. P. Skums, N. Mancuso, A. Artyomenko, B. Tork, I. Mandoiu, Y. Khudyakov, and A. Zelikovsky. Reconstruction of viral population structure from next-generation sequencing data using multicommodity flows. *BMC bioinformatics*, 14(Suppl 9):S2, 2013.

153. P. Skums, L. Bunimovich, and Y. Khudyakov. Antigenic cooperation among intrahost hcv variants organized into a complex network of cross-immunoreactivity. *Proceedings of the National Academy of Sciences*, 112(21):6653–6658, 2015.

154. P. Skums, A. Artyomenko, O. Glebova, D. S. Campo, Z. Dimitrova, A. Zelikovsky, and Y. Khudyakov. Error correction of ngs reads from viral populations. *Computational Methods for Next Generation Sequencing Data Analysis*, 2016.

155. P. Skums, A. Zelikovsky, R. Singh, W. Gussler, Z. Dimitrova, S. Knyazev, I. Mandric, S. Ramachandran, D. Campo, D. Jha, L. Bunimovich, E. Costenbader, C. Sexton, S. O'Connor, G.-L. Xia, and Y. Khudyakov. QUENTIN: reconstruction of disease transmissions from viral quasispecies genomic data. *Bioinformatics*, 34(1):163–170, June 2017. doi: 10.1093/bioinformatics/btx402. URL `https://doi.org/10.1093/bioinformatics/btx402`.

156. P. Skums, A. Zelikovsky, R. Singh, W. Gussler, Z. Dimitrova, S. Knyazev, I. Mandric, S. Ramachandran, D. Campo, D. Jha, et al. Quentin: reconstruction of disease transmissions from viral quasispecies genomic data. *Bioinformatics*, 34(1):163–170, 2018.

157. P. Skums, V. Tsyvina, and A. Zelikovsky. Inference of clonal selection in cancer populations using single-cell sequencing data. *Bioinformatics*, 35(14):i398–i407,

2019. doi: 10.1093/bioinformatics/btz392. URL https://doi.org/10.1093/bioinformatics/btz392.

158. S. Snir, Y. I. Wolf, and E. V. Koonin. Universal pacemaker of genome evolution. *PLOS Computational Biology*, 8(11):e1002785–, 11 2012. URL https://doi.org/10.1371/journal.pcbi.1002785.

159. S. Snir, B. M. vonHoldt, and M. Pellegrini. A statistical framework to identify deviation from time linearity in epigenetic aging. *PLOS Computational Biology*, 12(11):1–15, 11 2016. doi: 10.1371/journal.pcbi.1005183. URL https://doi.org/10.1371/journal.pcbi.1005183.

160. J. A. Somarelli, H. Gardner, V. L. Cannataro, E. F. Gunady, A. M. Boddy, N. A. Johnson, J. N. Fisk, S. G. Gaffney, J. H. Chuang, S. Li, et al. Molecular biology and evolution of cancer: from discovery to action. *Molecular biology and evolution*, 2019.

161. D. Steinhauer and J. Holland. Rapid evolution of rna viruses. *Annual Review of Microbiology, 41*, pages 409–433, 1987.

162. M. Tarabichi, I. Martincorena, M. Gerstung, A. M. Leroi, F. Markowetz, P. T. Spellman, Q. D. Morris, O. C. Lingjaerde, D. C. Wedge, and P. Van Loo. Neutral tumor evolution? *Nature Genetics*, page 1, 2018.

163. I. The, T. P.-C. A. of Whole, G. Consortium, et al. Pan-cancer analysis of whole genomes. *Nature*, 578(7793):82, 2020.

164. J. Thorn, H. Kishino, and I. Painter. Estimating the rate of evolution of the rate of evolution. *Mol Biol Evol*, 15:1647–1657, 1998.

165. I. P. Tomlinson, M. Novelli, and W. Bodmer. The mutation rate and cancer. *Proceedings of the National Academy of Sciences*, 93(25):14800–14803, 1996.

166. A. Töpfer, O. Zagordi, S. Prabhakaran, V. Roth, E. Halperin, and N. Beerenwinkel. Probabilistic inference of viral quasispecies subject to recombination. *Journal of Computational Biology*, 20(2):113–123, 2013.

167. A. Töpfer, T. Marschall, R. A. Bull, F. Luciani, A. Schönhuth, and N. Beerenwinkel. Viral quasispecies assembly via maximal clique enumeration. *PLoS Computational Biology*, 10 (3), 2014. doi: 10.1371/journal.pcbi.1003515. URL `http://dx.doi.org/10.1371/journal.pcbi.1003515`.

168. J. van de Haar, S. Canisius, K. Y. Michael, E. E. Voest, L. F. Wessels, and T. Ideker. Identifying epistasis in cancer genomes: a delicate affair. *Cell*, 177(6):1375–1383, 2019.

169. B. M. Verbist, K. Thys, J. Reumers, Y. Wetzels, K. Van der Borght, W. Talloen, J. Aerssens, L. Clement, and O. Thas. Virvarseq: a low-frequency virus variant detection pipeline for illumina sequencing using adaptive base-calling accuracy filtering. *Bioinformatics*, 31(1): 94–101, 2014.

170. E. K. Wagner, M. J. Hewlett, D. C. Bloom, and D. Camerini. *Basic virology*, volume 3. Blackwell Science Malden, MA, 1999.

171. R. A. Wagner and M. J. Fischer. The string-to-string correction problem. *Journal of the ACM (JACM)*, 21(1):168–173, 1974.

172. J. W. Ward. The hidden epidemic of hepatitis c virus infection in the united states: occult transmission and burden of disease. *Topics in antiviral medicine*, 21(1):15–19, 2013.

173. J. O. Wertheim, A. J. Leigh Brown, N. L. Hepler, S. R. Mehta, D. D. Richman, D. M. Smith, and S. L. Kosakovsky Pond. The global transmission network of hiv-1. *The Journal of Infectious Diseases*, 209(2):304–313, 2014. doi: 10.1093/infdis/jit524. URL `+http://dx.doi.org/10.1093/infdis/jit524`.

174. K. Westbrooks, I. Astrovskaya, D. Campo, Y. Khudyakov, P. Berman, and A. Zelikovsky. Hcv quasispecies assembly using network flows. *Bioinformatics Research and Applications*, pages 159–170, 2008.

175. C. O. Wilke. Quasispecies theory in the context of population genetics. *BMC evolutionary biology*, 5(1):1, 2005.

176. M. J. Williams, B. Werner, C. P. Barnes, T. A. Graham, and A. Sottoriva. Identification of neutral tumor evolution across cancer types. *Nature genetics*, 48(3):238, 2016.

177. M. J. Williams, B. Werner, T. Heide, C. Curtis, C. P. Barnes, A. Sottoriva, and T. A. Graham. Quantification of subclonal selection in cancer from bulk sequencing data. *Nature genetics*, page 1, 2018.

178. D. Wodarz and N. Komarova. *Computational biology of cancer: lecture notes and mathematical modeling*. World Scientific, 2005.

179. Y. I. Wolf, S. Snir, and E. V. Koonin. Stability along with extreme variability in core genome evolution. *Genome biology and evolution*, 5(7):1393–1402, 2013. doi: 10.1093/gbe/evt098. URL `https://www.ncbi.nlm.nih.gov/pubmed/23821522`.

180. S. L. Wong, L. V. Zhang, A. H. Tong, Z. Li, D. S. Goldberg, O. D. King, G. Lesage, M. Vidal, B. Andrews, H. Bussey, et al. Combining biological networks to predict genetic interactions.

Proceedings of the National Academy of Sciences, 101(44):15682–15687, 2004.

181. C. Wymant, M. Hall, O. Ratmann, D. Bonsall, T. Golubchik, M. de Cesare, A. Gall, M. Cornelissen, and C. F. and. PHYLOSCANNER: Inferring transmission from within- and between-host pathogen genetic diversity. *Molecular Biology and Evolution*, 35(3):719–733, Nov. 2017. doi: 10.1093/molbev/msx304. URL `https://doi.org/10.1093/molbev/msx304`.

182. D. Xu, Z. Zhang, and F.-S. Wang. SARS-associated coronavirus quasispecies in individual patients. *New England Journal of Medicine*, 350(13):1366–1367, Mar. 2004. doi: 10.1056/nejmc032421. URL `https://doi.org/10.1056/nejmc032421`.

183. X. Yang, P. Charlebois, S. Gnerre, M. G. Coole, N. J. Lennon, J. Z. Levin, J. Qu, E. M. Ryan, M. C. Zody, and M. R. Henn. De novo assembly of highly diverse viral populations. *BMC genomics*, 13(1):475, 2012.

184. X. Yang, P. Charlebois, A. Macalalad, M. R. Henn, and M. C. Zody. V-phaser 2: variant inference for viral populations. *BMC genomics*, 14(1):674, 2013.

185. Y. Yao and W. Dai. Genomic instability and cancer. *Journal of carcinogenesis & mutagenesis*, 5, 2014.

186. L. R. Yates and P. J. Campbell. Evolution of the cancer genome. *Nature Reviews Genetics*, 13(11):795, 2012.

187. H. Zafar, A. Tzen, N. Navin, K. Chen, and L. Nakhleh. Sifit: inferring tumor trees from single-cell sequencing data under finite-sites models. *Genome biology*, 18(1):178, 2017.

188. O. Zagordi, A. Bhattacharya, N. Eriksson, and N. Beerenwinkel. Shorah: estimating the

genetic diversity of a mixed sample from next-generation sequencing data. *BMC bioinformatics*, 12(1):119, 2011.

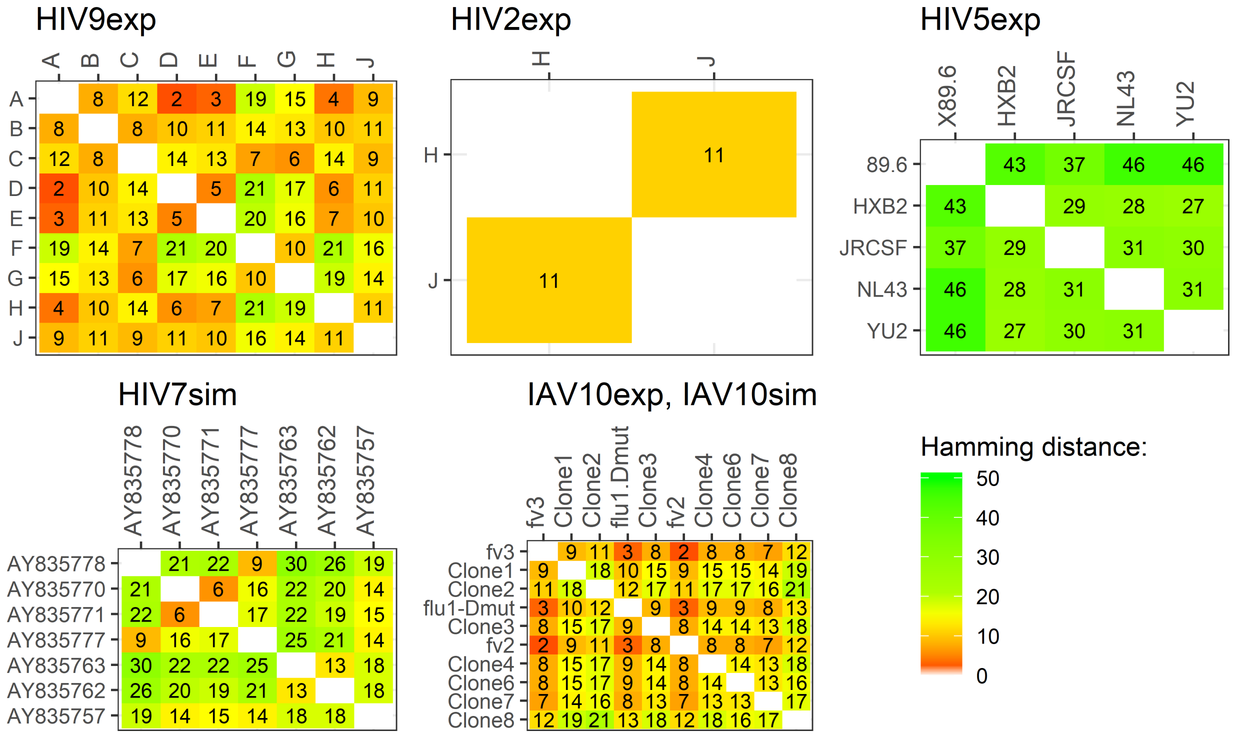
189. O. Zagordi, A. Töpfer, S. Prabhakaran, V. Roth, E. Halperin, and N. Beerenwinkel. Probabilistic inference of viral quasispecies subject to recombination. In *Proceedings of the 16th Annual international conference on Research in Computational Molecular Biology*, RECOMB'12, pages 342–354, Berlin, Heidelberg, 2012. Springer-Verlag. ISBN 978-3-642-29626-0.

190. F. Zanini, J. Brodin, L. Thebo, C. Lanz, G. Bratt, J. Albert, and R. A. Neher. Population genomics of intrapatient hiv-1 evolution. *eLife*, Dec 2015. URL `https://elifesciences.org/articles/11282`.

191. J. Zobel, A. Moffat, and K. Ramamohanarao. Inverted files versus signature files for text indexing. *ACM Transactions on Database Systems (TODS)*, 23(4):453–490, 1998.

A CliqueSNV materials

Here additional materials and figures for Chapter 3 are present.



HIV9exp

	A	B	C	D	E	F	G	H	J
A		8	12	2	3	19	15	4	9
B	8		8	10	11	14	13	10	11
C	12	8		14	13	7	6	14	9
D	2	10	14		5	21	17	6	11
E	3	11	13	5		20	16	7	10
F	19	14	7	21	20		10	21	16
G	15	13	6	17	16	10		19	14
H	4	10	14	6	7	21	19		11
J	9	11	9	11	10	16	14	11	

HIV2exp

	H	J
H		11
J	11	

HIV5exp

	X89.6	HXB2	JRCSF	NL43	YU2
89.6		43	37	46	46
HXB2	43		29	28	27
JRCSF	37	29		31	30
NL43	46	28	31		31
YU2	46	27	30	31	

HIV7sim

	AY835778	AY835770	AY835771	AY835777	AY835763	AY835762	AY835757
AY835778		21	22	9	30	26	19
AY835770	21		6	16	22	20	14
AY835771	22	6		17	22	19	15
AY835777	9	16	17		25	21	14
AY835763	30	22	22	25		13	18
AY835762	26	20	19	21	13		18
AY835757	19	14	15	14	18	18	

IAV10exp, IAV10sim

	fv3	Clone1	Clone2	flu1.Dmut	Clone3	fv2	Clone4	Clone6	Clone7	Clone8
fv3		9	11	3	8	2	8	8	7	12
Clone1	9		18	10	15	9	15	15	14	19
Clone2	11	18		12	17	11	17	17	16	21
flu1-Dmut	3	10	12		9	3	9	9	8	13
Clone3	8	15	17	9		8	14	14	13	18
fv2	2	9	11	3	8		8	8	7	12
Clone4	8	15	17	9	14	8		14	13	18
Clone6	8	15	17	9	14	8	14		13	16
Clone7	7	14	16	8	13	7	13	13		17
Clone8	12	19	21	13	18	12	18	16	17	

Figure 7 Pairwise Hamming distances between variants in the experimental (exp) and simulated (sim) datasets HIV9exp, HIV2exp, HIV5exp, HIV7sim, IAV10sim, and IAV10exp.

PacBio # of Reads	Method	Variant	fv3	Clone1	Clone2	flu1-Dmut	Clone3	fv2	Clone4	Clone5	Clone6	Clone7	FP
		True Freq.,%	50	25	12.5	6.25	3.125	1.56	0.78	0.39	0.19	0.097	
33.5K (all)	CliqueSNV	Match	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	0
		Freq., %	52.6	23.7	12.6	6.4	2.3	1.17	0.7	0.35	0.12	0.051	0
	2SNV	Match	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	1
		Freq., %	51.8	23.7	12.5	6.4	2.3	1.2	0.7	0.3	0.1	0	1.0
	PredictHaplo	Match	✓	✓	✓	×	✓	×	✓	✓	×	×	0
		Freq.,%	56.7	23.8	13.7	0	3.1	0	1.5	1.2	0	0	0
Subsampling													
16K	CliqueSNV	Match,%	100	100	100	100	100	90	100	100	100	20	0.1
		Freq., %	52.9	23.7	12.5	6.4	2.3	1.19	0.71	0.32	0.12	0.69	1.15
	2SNV	Match,%	100	100	100	100	100	100	100	100	0	0	0.2
		Freq., %	52.4	23.7	12.5	6.4	2.3	1.1	0.7	0.3	0	0	0.6
	PredictHaplo	Match	100	100	100	70	100	0	100	40	0	0	0.3
		Freq.,%	54.2	23.5	13.1	6.0	2.9	0	1.4	1.0	0	0	0.5
8K	CliqueSNV	Match,%	100	100	100	100	100	90	100	100	30	0	0
		Freq., %	52.8	23.6	12.5	6.5	2.3	1.2	0.7	0.35	0.16	0	0
	2SNV	Match,%	100	100	100	100	100	100	100	0	0	0	0
		Freq., %	53.1	23.7	12.5	6.5	2.3	1.25	0.7	0	0	0	0
	PredictHaplo	Match,%	100	100	100	0	100	0	100	20	0	0	0.2
		Freq.,%	58.1	24.0	12.7	0	3.1	0	1.6	1.3	0	0	0.5
4K	CliqueSNV	Match,%	100	100	100	100	100	80	100	40	0	0	0
		Freq., %	53.3	23.7	12.3	6.4	2.4	1.19	0.7	0.39	0	0	0
	2SNV	Match,%	100	100	100	100	100	100	20	0	0	0	0
		Freq., %	53.7	23.7	12.3	6.5	2.4	1.2	0.9	0	0	0	0
	PredictHaplo	Match,%	100	100	100	0	70	0	10	0	0	0	0.3
		Freq.,%	60.1	23.9	12.8	0	3.5	0	2.5	0	0	0	0.5

Table 2 Comparison of CliqueSNV, 2SNV and PredictHaplo on full and sub-sampled data ***(PacBio, experimental)***. For all 33.5K reads, the sign "✓" (respectively, "×") denotes fully matched (respectively, unmatched) true variant and the column FP reports the number of incorrectly predicted variants (false positives) and their total frequency. For each sub-sample size (16K,...,4K), the table reports the percent of runs when a variant is completely matched and its average frequency over runs when the variant was detected. Similarly, the column FP reports the average number of false positive variants and their average total frequency. Colors indicate the percent of matched variants: green - high percent, red - low percent.

Benchmark preparation

We used 50,000 total copies of plasmid DNA from these nine constructs as input for a nested PCR reaction to amplify the polymerase region using the following primary and nested primers

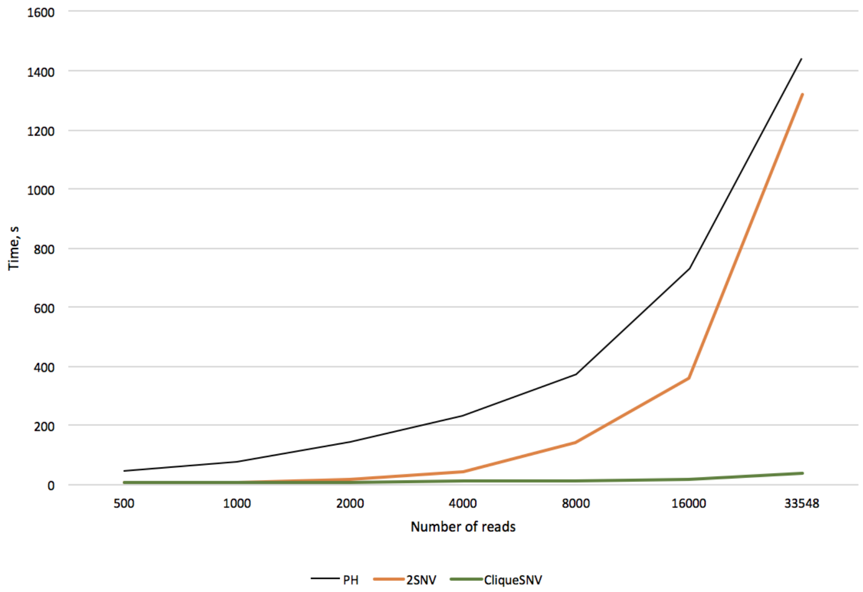


Figure 8 Runtimes of PredictHaplo (PH), 2SNV and CliqueSNV on datasets with different read sizes.

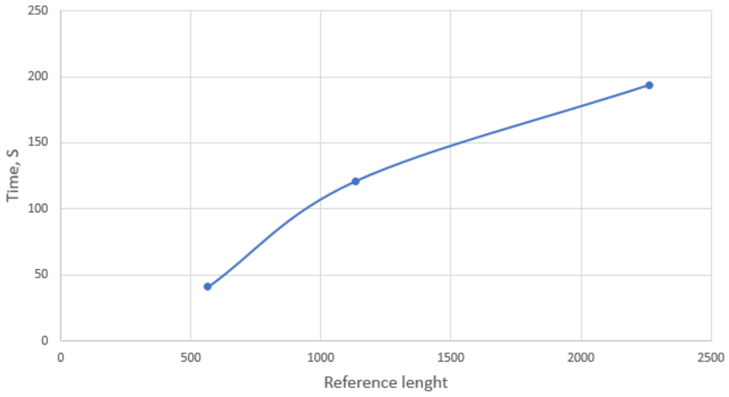


Figure 9 CliqueSNV runtime on datasets with different reference length and same coverage (about 1M reads in total).

respectively: HIV-B PRO-OUT.3F, 5’ CCT CAG ATC ACT CTT TGG CAA CG 3’ and HIV-RT 215/219.3R, 5’ CTT CTG TAT GTC ATT GAC AGT CC 3’ Nested PCR: HIV-B PR/RT.2F, 5’ CTT TGG CAA CGA CCC CTY GTC CA 3’ and HIV-RT 181-190.1.4R, 5’ ATC AGG ATG GAG TTC ATA ACC CA 3’.

The primary and nested PCRs were done using 94°C for four minutes, followed by 40 cycles of 94°C for one minute, 50°C for 30 seconds, and 72°C for two minutes and a final extension at 72°C for five minutes[126].

We created two plasmid mixtures to generate artificial mixtures simulating clinical specimens containing many variants at different virus levels. The mixtures comprised nine and two plasmids with varying copy numbers of each plasmid.

PCR reactions were generated and purified using the QIAquick PCR purification kit. (Qiagen, Valencia CA) The purified amplicons (10 ng) were subsequently used for NGS library construction using the Nextera XT DNA Library Prep kit (Illumina Inc., San Diego, CA). Libraries were pooled, and enriched for 900-1,000-bp fragments using magnetic bead based size selection (AMPure XP, Beckman Coulter, Brea, CA) and sequenced on a MiSeq v3 (600-cycle) flow cell on the Miseq system (Illumina Inc., San Diego CA).

Pseudocode of the CliqueSNV algorithm

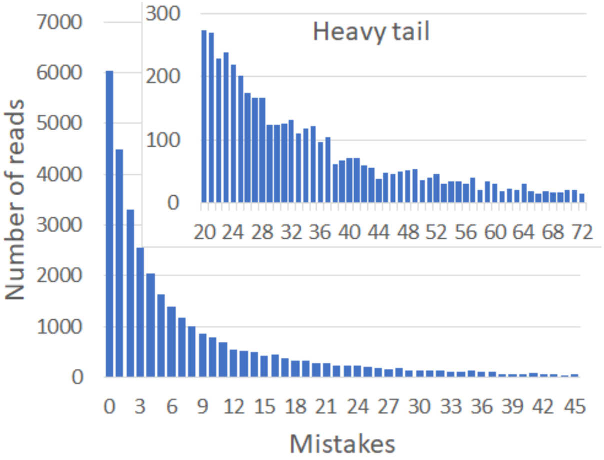


Figure 10 A typical distribution of errors in PacBio reads. The heavy tail indicates that a significant portion of errors is accumulated by a relatively small number of reads.

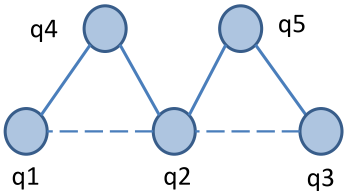


Figure 11 The clique graph C_G with 5 vertice corresponding to cliques in G, 4 edges and two forbidden pairs (q_1, q_2) and (q_2, q_3). There 3 maximal connected subgraphs avoiding forbidden pairs: $\{q_1, q_4\}$ $\{q_4, q_2, q_5\}$ $\{q_5, q_3\}$

Algorithm 5 CliqueSNV Algorithm

```
**Step 1: finding linked and forbidden SNV pairs**
    Split the read alignment $M_{L\times N}$ into binary matrix $4M$
    Construct a compact representation of the binary matrix $4M$
    For each $I, J \in \{1, \ldots, 4L\}$ find $O^{IJ}$ and $O^{IJ}_{22}$, where
        $O^{IJ}$ = # of reads covering both $I$ and $J$
        $O^{IJ}_{22}$ = # of reads with both minor SNVs
        If $O^{IJ}_{22} > \epsilon O^{IJ}$ compute $p$-value (default $\epsilon = 0.0003$)
    Find all linked SNV pairs with the adjusted p-value $< 1\%$
**end Step**
**Step 2: constructing the SNV graph**
    Filter out $10\%$ of the most erroneous PacBio reads
    Construct the SNV graph $G = (V, E)$, where
        $V = \{1, \ldots, 4L\}$, and $E$ are links between minor SNVs
**end Step**
**Step 3: finding maximal cliques in the SNV graph using Bron-Kerbosch algorithm[23]**
**end Step**
**Step 4: merging cliques in the clique graph with forbidden pairs**
    Find the clique graph $C_G$ with pairs.
    Find all maximal connected subgraphs in $C_G$.
    Merge all cliques inside each maximal connected subgraph.
**end Step**
**Step 5: partitioning reads between merged cliques and finding consensus haplotypes**
    Find the set $S$ of all positions that belong to at least one clique.
    Make an empty clique on $S$.
    Assign each read to the closest clique.
    Find the consensus $v(q)$ of all assigned reads for each $q$.
**end Step**
**Step 6: estimating haplotype frequencies by expectation-maximization algorithm**
**end Step**
```

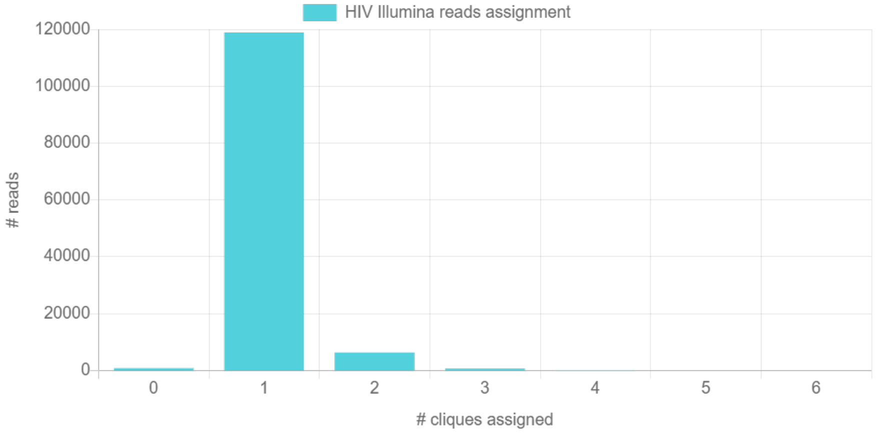


Figure 12 The number of reads assigned to different number of cliques in HIV Illumina dataset.

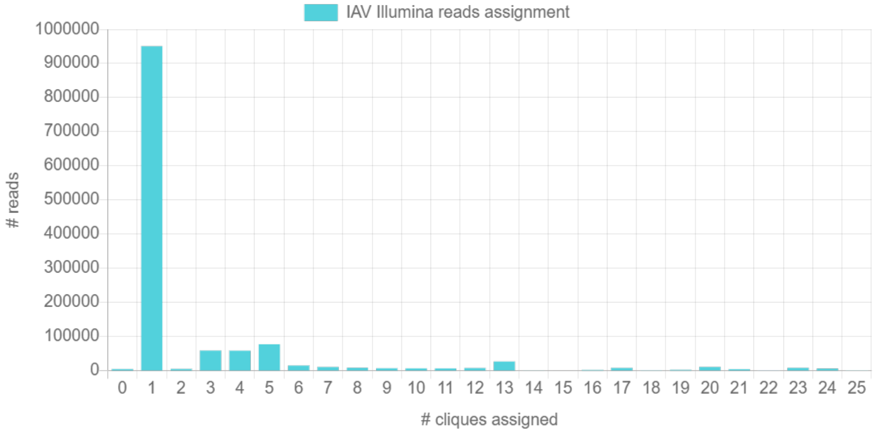


Figure 13 The number of reads assigned to different number of cliques in IAV Illumina dataset.

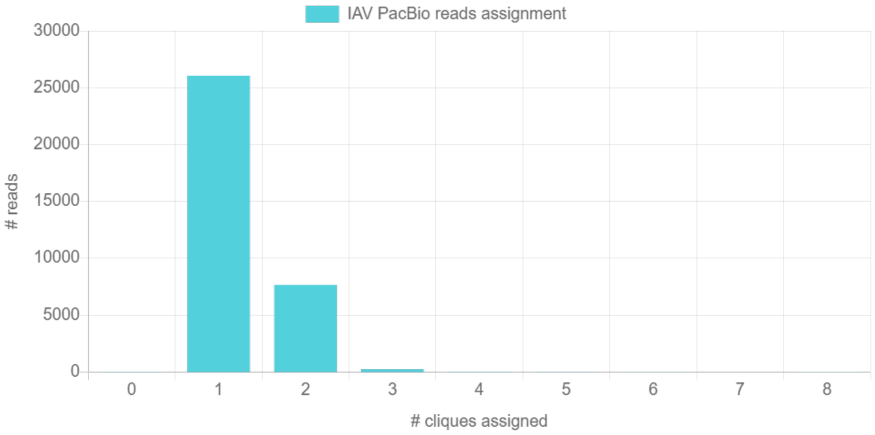


Figure 14 The number of reads assigned to different number of cliques in IAV PacBio dataset.