ETH zürich

ETH Library

Inferring Markov Chains to Describe Convergent Tumor Evolution with CIMICE

Journal Article

Author(s):
Rossi, Nicolò; Gigante, Nicola; Vitacolonna, Nicola; Piazza, Carla

Publication date:
2024-01

Permanent link:
https://doi.org/10.3929/ethz-b-000661070

Rights / license:
Creative Commons Attribution 4.0 International

Originally published in:
IEEE/ACM Transactions on Computational Biology and Bioinformatics 21(1), https://doi.org/10.1109/TCBB.2023.3337258

This page was generated automatically upon download from the ETH Zurich Research Collection.
For more information, please consult the Terms of use.

Inferring Markov Chains to Describe Convergent Tumor Evolution With CIMICE

Nicolò Rossi, Nicola Gigante, Nicola Vitacolonna, and Carla Piazza

***Abstract*—The field of tumor phylogenetics focuses on studying the differences within cancer cell populations. Many efforts are done within the scientific community to build cancer progression models trying to understand the heterogeneity of such diseases. These models are highly dependent on the kind of data used for their construction, therefore, as the experimental technologies evolve, it is of major importance to exploit their peculiarities. In this work we describe a cancer progression model based on Single Cell DNA Sequencing data. When constructing the model, we focus on tailoring the formalism on the specificity of the data. We operate by defining a minimal set of assumptions needed to reconstruct a flexible DAG structured model, capable of identifying progression beyond the limitation of the infinite site assumption. Our proposal is conservative in the sense that we aim to neither discard nor infer knowledge which is not represented in the data. We provide simulations and analytical results to show the features of our model, test it on real data, show how it can be integrated with other approaches to cope with input noise. Moreover, our framework can be exploited to produce simulated data that follows our theoretical assumptions. Finally, we provide an open source R implementation of our approach, called CIMICE, that is publicly available on BioConductor.**

***Index Terms*—Cancer progression, Markov processes, modeling, theory and models.**

I. Introduction

CANCER, one of the primary causes of death in developed countries, is a genetic disease where mutations change the behavior of some body cells inducing an out-of-control proliferation, with effects on the host comparable to those of a parasite. In addition, tumors are a complex class of diseases varying both at the macroscopic level (e.g., tumor location and size) and at the microscopic level (e.g., genetic asset and gene expression). Understanding the mechanisms behind the development of such diseases is an interdisciplinary challenge far from being solved. Current models represent such genetic drift as an evolutionary process [1], albeit with its own peculiarities. According to such a view, a tumor originates from a single cell and progresses by acquiring genetic variability, and therefore giving rise to several genetically distinct and relatively unstable cell populations called *clones,* competing (or cooperating) for the limited resources in their micro-environment. Several tumor evolution models have been proposed to explain such intra-tumor heterogeneity [2], and they aim to become powerful tools for understanding cancer progression and helping design effective treatments. Having mathematical descriptions of tumor evolution would provide a solid basis for the development of cancer research and bringing this to a single patient level would be a breakthrough for personalized medicine.

Exploiting this evolutionary perspective, several *tumor phylogenetic* techniques and methods have been developed over the years. These operate either by adapting computational approaches used in biology to reconstruct species evolution or by creating newer models, specifically crafted for this context [3], [4]. Independently of the chosen strategy, it is possible to define three main categories of cancer data [3]:

- *cross-sectional* methods, that combine samples from different tumors of different patients;
- *regional bulk* methods, where samples from different tumor sites of a single patient are collected; and
- more recently, *single-cell* methods, which analyze genomic data sequenced from single cells originating from a single tumor site [5].

Cross-sectional methods have shown how tumors tend to be quite diverse when considering primary site classification (see, e.g., [3] for an overview). To investigate this heterogeneity, *Single-Cell Sequencing (SCS)* samples are potentially the most useful as they allow the direct observation of instances of clonal subpopulations, but they are also the hardest to collect, and the scarcest among the currently available datasets. Nevertheless, as genomic sequencing becomes cheaper and more accurate, SCS datasets will become easier to produce and collect, and suitable computational methods are needed for their exploitation. In addition DNA data, with respect to RNA and protein one, is expected to have a more stable behavior and is directly inherited from

Manuscript received 11 June 2022; revised 16 November 2023; accepted 25 November 2023. Date of publication 28 November 2023; date of current version 5 February 2024. This work was supported in part by PRIN project NiRvAna CUP under Grant G23C22000400005 and in part by the National Recovery and Resilience Plan (NRRP), Mission 4 Component 2 Investment 1.4 - Call for tender No. 3138 of 16 December 2021, rectified by Decree No. 3175 of 18 December 2021 of the Italian Ministry of University and Research, funded by the European Union – NextGenerationEU; Project code CN_00000033, Concession Decree No. 1034 of 17 June 2022, adopted by the Italian Ministry of University and Research, CUP under Grant G23C22001110007, Project title "National Biodiversity Future Center - NBF." *(Corresponding author: Carla Piazza.)*

Nicolò Rossi is with the Department of Biosystems Science and Engineering, ETH Zürich, 4056 Basel, Switzerland, and also with the Life Science Zürich Graduate School of ETH Zürich and University of Zürich, Systems Biology program, 4056 Zürich, Switzerland (e-mail: olocin.issor@gmail.com, nicolo.rossi@bsse.ethz.ch).

Nicola Gigante is with the Faculty of Computer Science, Free University of Bozen-Bolzano, 39100 Bolzano, Italy (e-mail: nicola.gigante@uniud.it).

Nicola Vitacolonna and Carla Piazza are with the Department of Mathematics, Computer Science, and Physics, University of Udine, 33100 Udine, Italy (e-mail: nicola.vitacolonna@uniud.it; carla.piazza@uniud.it).

This article has supplementary downloadable material available at https://doi.org/10.1109/TCBB.2023.3337258, provided by the authors.

Digital Object Identifier 10.1109/TCBB.2023.3337258

© 2023 The Authors. This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

subsequent cell generations. These are very favorable properties for computational models.

In this paper, we describe a method to extract probabilistic models, which we call *Cancer Progression Markov Chains* (*CPMC*), from DNA SCS datasets, which describe the mutational history of the cells of a sampled tissue. In such models all the possible evolutionary paths witnessed by the data are represented. So, our approach is "conservative" with respect to other proposals which exploit statistical method to extract only the most likely paths. In this direction, we allow in our models convergent evolutions, which are also attracting the attention in the literature as possible cancer progression mechanisms (see, e.g., [6], [7]).

Moreover, being our method based on a minimal set of assumptions it has the advantage of highlighting the critical aspects in the use of mutational matrices extracted from DNA SCS data. In particular, we formally proved that the expressiveness of such data can be insufficient, independently from the inference method, to properly disentangle convergent trajectories.

CPMC are a particular kind of *Discrete Time Markov Chains (DTMC)* tailored to our use case, embedding useful mathematical properties. We first show that the kind of SCS datasets currently available can be modeled by CPMCs Then, we prove that a unique CPMC can be inferred from data, when there are no convergent evolutions. An algorithmic method to find such a solution is described. When the uniqueness is not guaranteed, we define a heuristic for inferring one of the possible models. Lastly, we propose a new tool, called *CIMICE*, which implements the described methods, and we evaluate its results on both synthetic and real datasets. CIMICE has been published as R package on Bioconductor [8].

A. Related Work

One of the earliest computational models of oncogenesis [9] represents the accumulation of mutations as an *oncogenetic tree* of causal dependencies among alterations.In this formalism, the root denotes the wild-type, and each path in the tree describes a sequence of causally related events. As in our approach, the nodes in an oncogenetic tree correspond only to observed genetic alterations, with no inferred genotypes. Despite some similarities with our approach, there are two fundamental differences between CPMC and oncogenetic trees: first, CPMC are *Directed Acyclic Graphs (DAGs)*, a more general graph topology allowing for path convergence, thus having trees just as a special case; second, and more importantly, the interpretaion of the edges' probabilities are different: in oncogenetic trees, the probability assigned to an edge is the probability of the event "this edge exists". In a CPMC, they instead represent the probabilities of transitioning from one state to the next.

One of the first computational methods to infer an evolutionary mutation tree from SCS data was proposed in [10]. Rather than inferring a phylogenetic tree, their method directly describes temporal ordering relationships among mutations sites by also taking into account sequencing errors. The idea is to compute a "pairwise order relation", which is a partial temporal ordering on the observed genotypes, represented by a genealogical tree, whose leaves are labeled by the observed genotypes and whose internal nodes correspond to putative common ancestors of the lineages of the samples. Then, mutations are superimposed on the branches of the tree, so that either a mutation temporally precedes another, or two mutations are considered independent. In other words, the ordering is determined by set inclusion: when a genotype has a subset of mutations of another, then the former must temporally precede the latter. For instance, if 00 encodes the wild-type, and 01, 11 are two other observed genotypes (with one and two mutations at the considered sites, respectively) then the inferred temporal ordering is $00 \rightarrow 01 \rightarrow 11$. To deal with situations that are inconsistent with the above rule, e.g., a triple 01, 10, and 11 of observed genotypes encoding a branching evolution from the wild-type, a Bayesian approach is incorporated into the method. A CPMC provides information similar to the genealogical tree of [10], but since CPMCs are DAGs, branching lineages of evolution can be trivially modeled in the graph structure. The previous "inconsistent" example would be modeled as a CPMC with four edges and a diamond topology: $00 \rightarrow 01$, $00 \rightarrow 10$, $01 \rightarrow 11$, $10 \rightarrow 11$.

OncoNEM [11] is an automated method based on a nested effects model for reconstructing clonal lineage trees from noisy somatic SNV data of single cells. OncoNEM works by clustering together cells with similar profiles; then, it infers their genotypes and unobserved ancestral genotypes; finally, it outputs the inferred tumor subclonal compositions, an evolutionary tree describing the history of such subpopulations, and posterior probabilities of the occurrence of mutations. OncoNEM's algorithm works by assigning a probabilistic score to sets of mutations and by searching for high-scoring models in the space of possible trees.

SCITE [12] is also a max-likelihood search algorithm that infers the evolutionary history of a tumor from noisy and incomplete SNV data, but, unlike OncoNEM, it focuses on mutation trees. SCITE makes the infinite sites assumption, hence it assumes that the input matrix describes a perfect phylogeny. Such a tree cannot have two nodes labelled with the same genotype. This condition together with the absence of convergent paths (intrinsic in the definition of tree) excludes convergent evolution. The infinite sites assumption justifies this choice by stating that there are so many possible mutation sites, that the probability of picking a specific site twice at random is negligible. However, contradictions of this hypothesis can be found in SNV databases for cancer, such as COSMIC [13].

An important limitation of both OncoNEM and SCITE is that they work under the infinite sites assumption, i.e., under the hypothesis that each mutation may only occur once in the evolutionary tree. Evidence has been brought forward to show that real SCS data violates that assumption, and that finite-site models taking into account chromosomal deletions, loss of heterozygosity and convergent evolution lead to more accurate inference of tumor phylogenies [14], [15], [16], [17]. Although our approach assumes that mutations are never lost, we do allow for convergent evolution.

Classic phylogenetic approaches, such as UPGMA and neighbor-joining [18], [19], [20], and other kinds of clustering methods [21], have also been applied to SCS data. Building correct phylogenies with such methods can be done efficiently under the infinite sites assumption if the data contains no errors

and mutations persist generation after generation [22]; under less restrictive hypotheses, however, they tend to be outperformed by the more focused approaches described above.

To improve the accuracy of variants detection, single-cell specific variant callers should be used. Monovar [23] and SC-caller [24] were the first two callers developed specifically for SCS data; SCIΦ [25] and SCAN-SNV [26] are two more recent approaches to solve the same problem.

One limitation of our proposal is that it assumes that subclonal reconstruction has already been performed, and clonal genotypes have been resolved. Rather than including a specific inference method into our model, we rely on tools such as SiCloneFit [16], Single Cell Genotyper [27] or BEAM [28] to provide the required input.

Although the technology is continually improving, the number and size of published SCS data sets are still limited. A few tools exist that permit generating simulated SCS data sets, and in some cases also inferring their phylogenies [29], [30].

Another way to tackle the lack of high-quality high-throughput SCS data is to develop statistical models that combine such data with traditional bulk sequencing data [31], [32], [33], [34], [35]. In this paper, however, we consider only SCS data.

Finally, the literature on SCS and computational analysis is too large to be summarized exhaustively. Several surveys on various methods and tools for inferring tumor histories from single-cell genomic data have been published to date, including [2], [3], [4], [22], [36], [37], [38], [39], [40], [41], [42], [43], [44].

II. Results and Discussion

In this work, we consider a DAG model suggested by the following general intuition. Phylogenetic trees put each existent taxon (e.g., a cell from an SCS experiment) in a leaf, and the internal nodes are their inferred extinct ancestors. However, in a tumor more complex evolutionary progression are possible and DAG models allow to better represent such trajectories especially in the case of convergent evolution. In fact, our method does not assume that a perfect phylogeny exists for a set of cells, i.e., a cell having a given mutation m_1 and a cell having another mutation m_2 may both generate cells having mutations m_1 and m_2.

We identify instead a minimal set of assumptions on tumor evolution, ensuring that DTMCs having DAGs as support correctly model the disease progression. Intuitively, DTMCs are probabilistic models in which the next state of a system only depends on the current one. In our context, the state of the system is the genotype of a tumor cell. The tumor cell will generate new cells whose genotypes will represent the next state.

We assume that:

- ($\emptyset$) the evolution starts from "normal" cells, i.e., cells which exhibit the same genotype as the healthy cells of the same patient;
- ($\cup$) mutations can only be acquired along the progression of the disease;
- (MC) the probability that a cell will generate cells with new mutations only depend on the genotype of the cell itself;
- ($\triangledown$) a minimal number of mutations is acquired in each new generation of cells.

DNA SCS data support hypothesis ($\emptyset$) and ($\cup$) since the DNA sequence is not influenced by the cell's life cycle and all mutations can be detected independently of the gene expression levels.

We are not pretending that these assumptions completely describe the high level of complexity of tumor evolution. Instead, we are trying to reason on the smallest possible set of hypotheses that allows us to rely on DTMCs as modeling formalism and to infer the underlying chain from a dataset. While hypothesis (MC) allows the use of DTMCs, the other hypotheses guarantee that the model has a simple topology, i.e.:

- it is acyclic[1] thanks to ($\cup$);
- it has a single source thanks to ($\emptyset$);
- it has no "forward" edges allowing to jump intermediate states thanks to ($\triangledown$).

Agreeing on the above assumptions, we propose the use of such DTMCs, later called CPMCs, as models for the mutational evolution of a tumor as inferred from a dataset of genotypes collected from cancer cells. These models can be used as *generators* for SCS mutational matrices that we can exploit as synthetic data. In addition, we define also the *jump* version of a CPMC, another DTMC in which at each step we force a mutational event to occur, hence eliminating any self-loop. Then, we consider the problem of inferring this latter model from data. In particular, we propose a method that takes in input a Mutational Array containing the genotypes of a set of cells taken from the tumor at a single time and outputs the jump version of a CPMC that could explain the data. The dataset has to be representative of all the genotypes present in the tumor, i.e., it has to reflect the actual genotype distribution (see Section Sample Size Evaluation in the supplementary material, for some statistical considerations). When the dataset does not support two or more possible explanations for a genotype (no convergences) we formally prove that the CPMC that we output is the only model that satisfy our hypothesis. When there are convergences, we output one of the possible CPMCs that explains the data.

We do not aim at correcting errors in the input data within our method. For this task we rely instead on other tools capable of clustering and cleaning mutational matrices based on known or predicted false positive, false negative and missing value rates. As an example of such an approach, the CIMICE package provides an easy-to-use interface to SiCloneFIT's preprocessing algorithm.

To further reduce the dependency of the results from random noise in the data, we implemented a bootstrap-based approach that consists in the random resampling of the input mutational matrix's rows. Bootstrapping allows us to evaluate how much support there is in the data for both the nodes and the transitions that we intend to reconstruct. This allows CIMICE to produce multiple CPMC models that are finally merged, helping the user to identify nodes and edges that might be generated only because

[1] A part from the possible presence of self-loops.

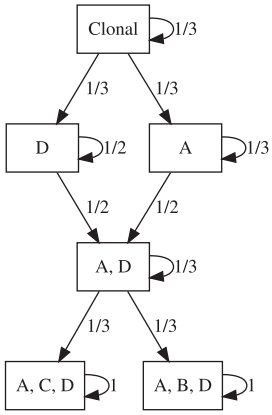
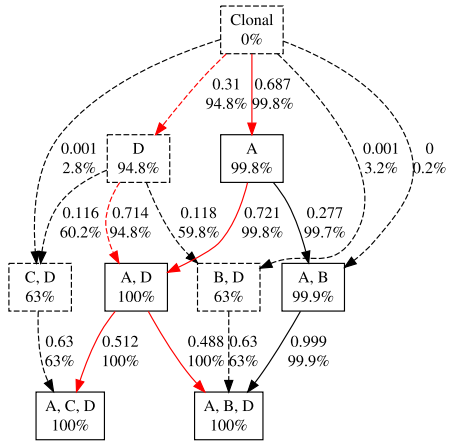
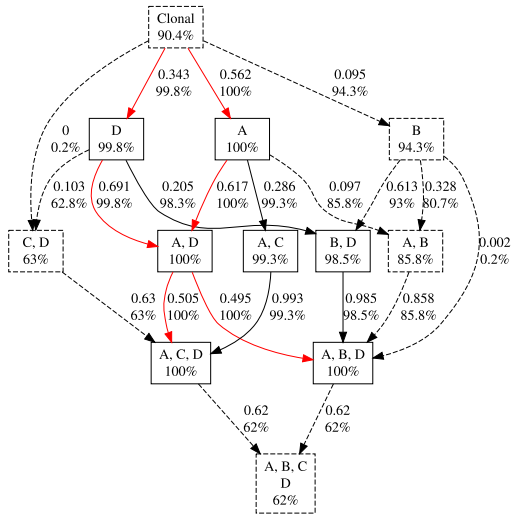


Fig. 1. CPMC used to generate the example datasets. Random paths of fixed length are simulated from the clonal node to generate the genotype of a single cell.



(a) $FP = 0.01$ and $FN = 0.05$.



(b) $FP = 0.01$ and $FN = 0.15$.

Fig. 2. Examples without SiCloneFit preprocessing. Dashed components have a bootstrap probability less than 95%. Red arrows were present in the original topology.

of the noisy nature of SCS data. The merging operation is done naturally by averaging the weighted adjacency matrices of the CPMCs produced by running CIMICE on the different sampled datasets.

In order to assess the performance of CIMICE, we test it on both artificial datasets generated accordingly to our model with different levels of noise and two real world case studies.

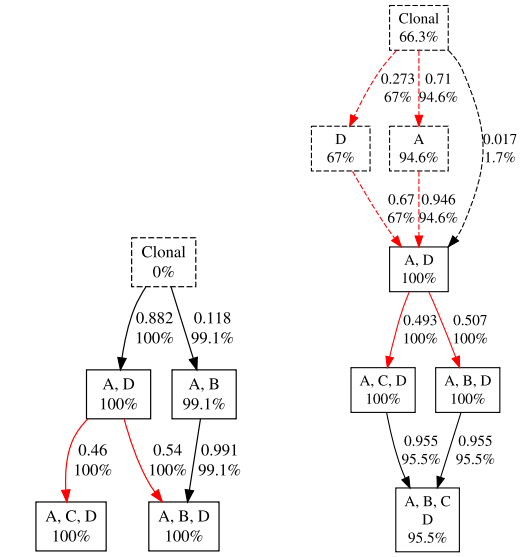
As for the simulation, the datasets were generated from the graph in Fig. 1, setting the length of the generated path k to 5 and simulating 100 cells. The length of the generated path represents how far in the cancer progression the dataset has been sampled, while the number of simulated cells is the number of samples in the SCS dataset. In order to include in our generated data errors that are typical of real SCS experiments, we considered different false positive FP and false negative FN rates. The FP rate is the probability of detecting a mutation in a cell when it is not present. Similarly, the FN rate is the probability of not detecting a mutation that is present. We repeated the simulation 4 times for the following FP and FN rates:

- $FP = 0.01$ and $FN = 0.05$
- $FP = 0.01$ and $FN = 0.10$
- $FP = 0.01$ and $FN = 0.15$

In Figs. 2 and 3, we test and compare results between CIMICE and CIMICE coupled with SiCloneFit's preprocessing algorithm. To produce each result, we resampled 100 cells from the generated dataset 1000 times. The sample size selected here was decided to adhere with most of the publicly available real datasets and to display the performance of the method in a challenging scenario. As the reader can notice, SiCloneFit allows us to improve the genotype/node selection in presence of high noise data. This results in more compact graphs which exclude most of the spurious transitions. However, also the models produced by our method alone are consistent with the generator, if considering only the solid nodes/edges that have passed the bootstrapping test. This means that such nodes have been represented in a fraction of at least $1 - p$ of the models produced through the bootstrapping procedure. Note that p is a hyperparameter to be chosen between 0 and 1 which balances the sensitivity-specificity tradeoff. In the examples its $p = 0.05$. Fig. 4 shows the sensitivity and the false negative rates obtained in recollecting the edges in the mentioned examples. Note that even if SiCloneFit helps in reducing the size of the final model, it may be too restrictive and limit the correctness of the reconstructed topology.

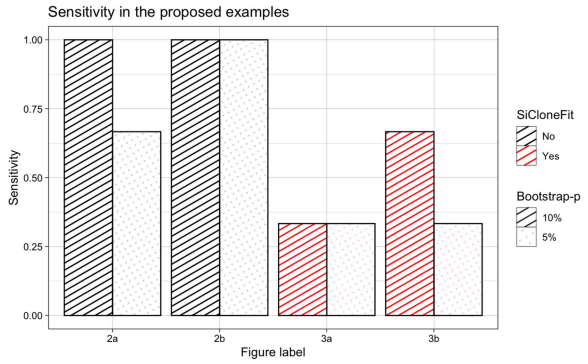
The supplementary material provides other simulations based on the generator of Fig. 1. For each dataset the models reconstructed by CIMICE and by CIMICE coupled with SiCloneFit are compared with those obtained by SCITE's, highlighting the impact of its assumptions (Supplementary Figs. 2–13).

As anticipated in the introduction, the solution to the inference problem may be not unique. We show in the Methods Section that the uniqueness is guaranteed when there are no convergences in the evolutionary trajectories. This does not hold in the general case and we rely on a heuristic that assumes uniform proportional distribution of the possible sources for a given sample. In Section Validation of the supplementary material, we consider

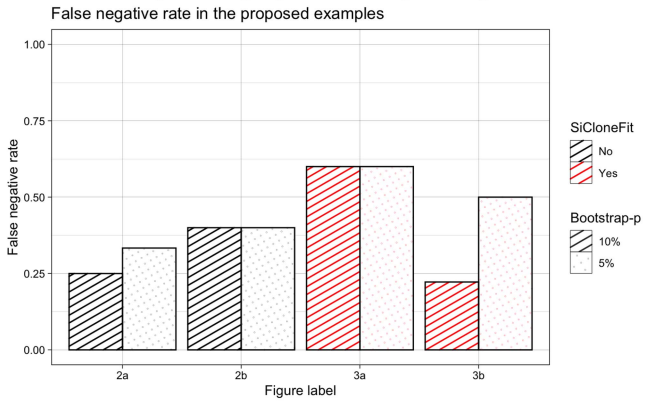


(a) $FP = 0.01$ and $FN = 0\ 05$ (b) $FP = 0.01$ and $FN = 0\ 15$

Fig. 3. Examples with SiCloneFit preprocessing. Red arrows were present in the original topology.



(a) Sensitivity in the proposed examples (higher is better).



(b) False negative rate in the proposed examples (lower is better).

Fig. 4. Performance metrics in the shown examples. The *Bootrstrap-p* value indicates that edges with support less than $1 - p$ in the bootstrap test were excluded from the final model. We remind again that these examples show a highly challenging scenario with strong lack of data.

two possible CPMCs for generating synthetic data and we use them for evaluating the performances of our inference method. While the first CPMC is designed to agree with our heuristic, the second one is explicitly studied to represent a "worst case" scenario in this regard. In both cases the topology is correctly reconstructed, while as expected the weights of the edges are more sensitive.

Finally, we test our approach on real datasets, specifically in two settings. In the first we considered a dataset on clear cell renal cell carcinoma from [20], and in the latter metastatic colorectal cancer data from [45] (CRC1). In Figs. 10 and 11 we show the results of our method coupled with SiCloneFit preprocessing, using the False positive, negative and missing rates reported in [20] and [45], respectively. We set the bootstrapping method to resample 100 datasets with the same size of the original ones.

We consider the results from our tool on the dataset of [20] as it is a common benchmark for tumor phylogenetics. In the reconstructed model depicted in Fig. 10 we label the nodes with the sample identifiers that correspond to the genotype identifiers in our context. The amount of considered cells is extremely low (17 samples), far from the requirements needed to successfully recapitulate the evolutionary history (see the supplementary material, section Sample Size Evaluation). However, our method distinguishes two clusters of cells: the early-stage ones are represented in the first layer of the generated chain and the remaining ones in the second layer (10). A possible interpretation of our reconstruction, is to consider the first group as the set of possible transient states, that evolve to eventually converge to a single final state. Notice that the phylogenetic tree reported in [20] is built without applying any technique for error correction. As a consequence, some samples are considered different in the phylogenetic tree, while they exihibit the same genotype in our model due to the SiCloneFit preprocessing step. Despite these differences we can notice some similarities, for instance `RC-6` and `RC-7` are far from the normal cell in both cases.

As far as a comparison between the results reported in [45] and ours (Fig. 11), we notice a consistency between the reconstructed models. In particular, the mutations associated with the metastatic phase almost coincide and appear in the deeper nodes in both models. We notice a difference on the positioning of GATA1. In [45] it is reported as a metastatic mutation, while in our model it is introduced in early stages. Evidence supporting our reconstruction can be found in the literature (see, e.g., [46], [47]), where GATA1 is suggested as a driver gene, aggressiveness predictor, and potential therapeutic target.

III. Conclusion

In this manuscript we have presented CIMICE, a framework for inferring tumor evolution as a Discrete Time Markov Chain from single cell DNA sequencing (scDNASeq) data. Compared to other methods, our tool is based on a limited set of assumptions and is capable to incorporate convergent evolutionary trajectories.

The proposed approach embeds a heuristic to reconstruct the probabilities of the tumor taking different paths through its progression. This simplification assumes that the paths that have

originated a given genotype have equal proportional probabilities. If more information about the history of certain cell classes is available, e.g., through expert knowledge, our approach can be straightforwardly extended to include it in the inference.

The fulfillment of our assumptions and a sufficient quantity of data are necessary for obtaining statistically significant results. However, we have shown that some information can still be extracted when data is scarce. The development of new sequencing methods, like the Tapestri platform [48], will help in generating larger and more reliable datasets for our method, approaching the sample sizes required by our analyses.

The final outcome of the proposed method is a Discrete Time Markov Chain and, as such, it can be analyzed with model checkers such as PRISM [49]. Such an approach would allow to find and evaluate temporal properties about the system and, therefore, of the modelled tumor. In addition, this representation can be naturally exploited to consider the interplay of drug effects, possibly relying also on more sophisticated hybrid models (see, e.g., [50]). Unfortunately, this direction requires mapping such effects to logical models based solely on the genotypes of the cells, which could be both very challenging and limiting in certain circumstances.

Another direction for improvements would be to include additional information that is not available from the genotypes alone, such as the RNA expression levels, epigenetic markers and histological/spatial data. Such data can help in providing a deeper characterization of the tumor, allowing for the refinement of the proposed subclones. Even if coping with this less stable data is, from the modelling point of view, much more complex, it could be exploited to resolve part of the ambiguity on convergences present in mutational matrices.

In conclusion, our method is a step forward in describing mutation accumulation in scDNASeq data. It develops over a limited set of clearly stated assumptions and it is flexible enough to cope with noisy data if paired with a proper preprocessor. In addition, it is reasonable to foresee that, with the progress in sequencing technologies, data limitations will be overcome in the near future, bringing our method to its full potential.

IV. Methods

A. Setting the Biological and Experimental Context

We represent the state of a cell as the set of mutations present in it. A normal cell is considered to harbor no mutations. Such absence of variants can be defined by exploiting either an external reference or the the healthy cells of the patient. The first method requires attention in mutation selection, while the latter may hide genetic predisposition to tumor development. A cell in any other state than the normal one is possibly dangerous.

Formally, let $G = \{g_1, \ldots, g_n\}$ be a set of *gene mutations*. A set $S \subseteq G$ denotes the *genotype* over G for a cell in which the mutations of S are present, while the mutations in $G \setminus S$ are not. In particular, $\emptyset$ is the genotype of a normal cell. The set of all possible genotypes over G is $\mathcal{P}(G)$, the power set of G.

We are interested in the reconstruction of a probabilistic model representing the mutational history of a cell, i.e., the temporal sequence of the genotypes of the cell's ancestors. To this aim, some assumptions on the mechanisms underlying the mutational events are needed. We formulate the following **Model's Hypotheses**:

- ($\emptyset$) The normal cell $\emptyset$ is an ancestor of every cell.
- ($\cup$) Mutations can only be acquired, and multiple mutations may be acquired from one cell generation to the next.
- (MC) The probability of a mutational event in a cell only depends on its current genotype—that is, it does not depend on how the cell reached a certain state.
- ($\triangledown$) An evolutionary history is anti-transitive and minimal, in the sense that it does not contain another evolutionary history that can explain the same observed genotypes, subject to the requirement that it must account for every plausible trajectory—that is, if $X \subset Y$ are two genotypes then there must be a path from X to Y.

The *empty set* hypothesis ($\emptyset$) states that each mutational history always starts from a normal cell. This hypothesis can be relaxed without significantly affecting the results in this paper. For instance, if there are some mutations that are present in all the cells of the system under analysis, the $\emptyset$ genotype can be replaced by a given genotype containing the mutations acquired at birth by the patient. The *union* hypothesis ($\cup$) specifies that mutations are never lost, i.e., the genotype of an ancestor of the current genotype is a subset of the current genotype. The *homogeneous Markov chain* hypothesis (MC) states that the acquisition of mutations is probabilistic and can be modeled through Markov chains, since each genotype uniquely determines the probability of transitioning to any other genotype. The *anti-transitivity* hypothesis ($\triangledown$) asserts that whenever it is possible to observe a sequence of transitions from a genotype to another, it is not possible to observe any of its subsequences. This is a sort of parsimony assumption, because it implies that each new mutated generation only acquires a minimal number of mutations.

Example 1 from the Additional examples section of the supplementary material shows the impact of these hypothesis in a simplified setting.

B. Details and Limitations of the Used Data

We focus on SCS data and in particular on DNAseq data, as the DNA molecule offers the chemical stability properties needed for our hypothesis that the RNA cannot provide. Moreover, the DNA sequence is not influenced by the cell's life cycle and all mutations can be detected independently of the gene expression levels.

Aware of the limitations and errors of the current SCS technologies, in this work we consider an ideal setting in which all the relevant mutations are correctly detected and numerous cells from a tumor region are analyzed. As the technology improves, it is reasonable to assume that larger datasets will become available and that the rate of errors will decrease. Currently, to approximate this ideal setting, the data may be preprocessed with tools that impute missing values and resolve clonal genotypes [16], [27], [28]. A possibility to derive relatively large datasets is to preprocess data from bulk sequencing experiments and extract plausible single cell explanations [51].

As we will see, working on SCS data has the following advantages with respect to bulk sequencing data:

- we can drastically simplify the model inference engine, since the set of genotypes present in the tumor are represented in the data and do not need to be inferred;
- we can formally prove that when each cell has a unique possible set of ancestors, the produced model is *correct*, i.e., no other information is needed;
- we propose models that can be used to generate artificial data.

An SCS experiment consists in the sequencing of a set of cells taken from either in vivo or in vitro samples. Hence, the genotype of each analyzed cell is known. Usually, the results of such experiments are represented through Boolean matrices, called *Mutational Arrays* [3], in which each row represents a cell and each column represents a mutation. The value in position (i, j) of a mutational array is 1 if and only if the j-th mutation is present in the i-th cell. Mutational arrays have a broad usage among many tools in the field of tumor phylogenetics (see, e.g., [52], [53]).

As for the underlying models, we need some assumptions on the data as well. In particular, our **Data Hypotheses** are:

(ONE) All the analyzed cells are taken at the same time from a single site, i.e., they represent *one* snapshot of a cancer tissue.

(POP) The analyzed cells reflect the genotype distribution of the *population* of all the cells in a given site.

Under these assumptions, given a mutational array, it makes sense to define a frequency distribution over the genotypes.

Definition 1 (Dataset Distribution): Let G be a set of gene mutations. A *dataset distribution* D *over* G is a frequency distribution over the genotypes of G, i.e., a function $D : \mathcal{P}(G) \longrightarrow [0, 1]$, with $\sum_{S \in \mathcal{P}(G)} D(S) = 1$.

In what follows, we will omit the underlying mutational arrays and refer to the corresponding dataset distributions, called simply *datasets* hereafter. A dataset is typically defined by its support: although the size of $\mathcal{P}(G)$ is exponential with respect to the size of G, usually only a limited number of genotypes is observed, so the support usually has a small size. Summarizing, $\mathcal{P}(G)$ is essentially the observed frequency of the genotype G in the dataset D (see Example 2 from the Additional examples section of the supplementary material).

Our goal is to find a plausible probabilistic model of the mutational histories of the cells in a given dataset, within a certain class of Markov models, based on the previously stated assumptions. In general, such a problem does not have a unique solution, even in an ideal generalized setting in which an infinite sequence of datasets corresponding to temporal snapshots of a sequenced tissue is available (see Example 5). In order to overcome such difficulty, we will (a) identify a few additional conditions guaranteeing the uniqueness of the model reconstructed from a dataset, and when uniqueness cannot be achieved, explicitly describe what missing information prevents that; and (b) when such additional information is not available, propose a reasonable criterion for the reconstruction of an admissible model.

C. Basics of Discrete Time Markov Chains

Given our model's hypotheses, it is reasonable to consider DTMCs as the underlying mechanism generating the data. The nodes of such chains correspond to the possible genotypes of a cell, while the edges model the probability of a genotype to mutate, i.e., to acquire new mutations. In the supplementary material, section Introduction to Discrete Time Markov Chains, we briefly report some ground definitions, such as the concept of *jump chain*, and notations used in this manuscript. We refer the reader to [54], [55] for a complete presentation on the topic.

D. Cancer Progression Markov Chains

In our context, we refer to a subset of DTMCs that we call Cancer Progression Markov Chains (CPMCs). CPMCs have additional properties which make them admissible as models of cancer progression. Intuitively, the vertexes of a CPMC represent the genotypes involved in the cancer progression under analysis. The empty genotype is the normal one that is at the origin of every mutational history. Every other vertex is reachable from the empty genotype. Since, under our hypotheses, mutations cannot be removed, a vertex representing a genotype cannot reach another vertex representing a genotype with fewer mutations. As a consequence, CPMCs are always acyclic. Moreover, since we are assuming that the evolutionary history is always the one involving fewer mutations, whenever there is a path of length at least 2 from one vertex to another, there cannot be an edge connecting the two vertices. This implies that CPMCs are anti-transitive. The above observations lead to the formulation of the following definition of CPMCs.

Definition 2 (Cancer Progression Markov Chain (CPMC)): Consider a set of genes $G = \{g_1, \ldots, g_n\}$ and let $\mathcal{S} = \{S_1, \ldots, S_m\}$ be a set of genotypes over G, with $S_1 = \emptyset$. A *Cancer Progression Markov Chain* $C = (\mathcal{S}, p)$ *over* $\mathcal{S}$ is a DTMC such that:

1) $S_1 = \emptyset$ reaches any other genotype of the chain;
2) for every $i, j \in [1, m]$, $p(S_i, S_j) > 0$ if and only if $S_i \subseteq S_j$ and there is no $k \neq i, j$ such that $S_i \subseteq S_k \subseteq S_j$.

The first condition of our definition of CPMC states that the normal genotype $\emptyset$ is always present, and it is the initial state of any mutational evolution. In other terms, in CPMCs we are always implicitly considering the initial distribution that at time 0 gives probability 1 to $\emptyset$ and 0 to all the other states.

In the second condition of the above definition, we have been more restrictive than stated in our hypotheses. In particular, we have imposed that whenever a genotype S_i is one of the minimal explanations for a genotype S_j, the probability of going from S_i to S_j is greater than 0. This restriction is not too demanding, since such probability can be arbitrarily small. It allows us to uniquely define the topology (i.e., the set of edges) of the chain for a given set of genotypes. However, it is possible to drop such restriction when further information on the topology is available.

Example 1: Let us consider the set of genes $G = \{A, B, C\}$ and the set of genotypes $\mathcal{S} =$

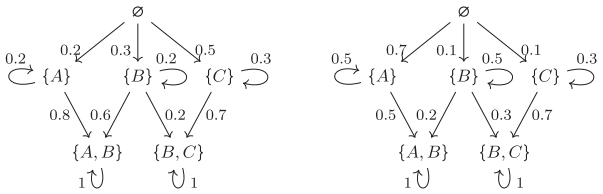


Fig. 5. CPMCs over $\{\emptyset, \{A\}, \{B\}, \{C\}, \{A, B\}, \{B, C\}\}$.

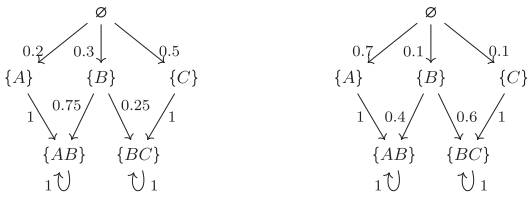


Fig. 6. Jump chains. The jump chains associated with the CPMCs depicted in 5.

$\{\emptyset, \{A\}, \{B\}, \{C\}, \{A, B\}, \{B, C\}\}$. In Fig. 5 we represent two possible CPMCs over $\mathcal{S}$.

Since a CPMC is a DTMC, given a CPMC C, we can build the jump DTMC $J(C)$ associated to C. The properties of C ensure that also $J(C)$ is acyclic, with a single source vertex, and anti-transitive. In particular, $J(C)$ is still a CPMC.

Lemma 1: Let C be a CPMC. Then C is acyclic, anti-transitive, and $J(C)$ is a CPMC.

Example 2: Let us consider the two CPMCs depicted in Fig. 5. Fig. 6 represents their associated jump chains.

E. Cancer Progression Markov Chains and Datasets

Let us assume that we know that the evolution of a type of cancer is regulated by a given CPMC C. We can use C to generate simulations of the evolution of the cancer. Moreover, we can use C to determine the probability that a cell with a given genotype will degenerate into another one.

Notice that in CPMCs time evolves, i.e., edges are crossed, when a cell cycle is completed. However, it makes no difference in our context to replace single cell cycles with their multiples, e.g., consider the new state after 100 cell cycles, or even with periodic observations of the system. On the other hand, we could have referred to Continuous Time Markov models in which time can be expressed in days, months, years (depending on the desired granularity). In that case, probabilities would have been replaced by transition rates. However, without more specific knowledge on proliferation/death rates of different genotypes, continuous time models would give us an equivalent view.

Interestingly, CPMCs can be used as data generators to validate other inference methods, provided that such methods agree on our four Model Hypothesis. In particular, we can randomly generate a CPMC C, use it to generate a dataset D_k, where $D_k(S)$ is the probability that C is in state S at time k, i.e., $D_k(S) = P[X(k) = S]$, apply the inference method on D_k and check whether the inferred knowledge is correct with respect to the underlying chain C. This process can be repeated until we are able to either accept or reject the inference method. The CPMCs can also be artificially engineered in order to test the behavior of the method on limit cases. Section II of the supplementary material presents in more details this idea and exploits it for validating our proposal.

Fig. 7. Two CPMCs that generate the same datasets at different time instants.

We recall that, since $C = (\mathcal{S}, p)$ is a Markov Chain and we assume that at time 0 the process starts from the normal genotype $\emptyset$, we have:

$$D_0(T) = P[X(0) = T] = 0 \text{ if } T \neq \emptyset$$
$$D_0(\emptyset) = P[X(0) = \emptyset] = 1$$

and

$$D_k(T) = P[X(k) = T] = \sum_{S \in \mathcal{S}} P[X(k-1) = S] \times p(S, T)$$

for each $k > 0$

Example 3: Let us consider again the CPMC C depicted in Fig. 5 on the left. The dataset D_0 generated by C at time 0 is $D_0(\emptyset) = 1$. The dataset D_1 generated by C at time 1 is $D_1(\{A\}) = 0.2$, $D_1(\{B\}) = 0.3$, and $D_1(\{C\}) = 0.5$. The dataset D_2 generated by C at time 2 is $D_2(\{A\}) = 0.2^2$, $D_2(\{B\}) = 0.3 \times 0.2$, $D_2(\{C\}) = 0.5 \times 0.3$, $D_2(\{A, B\}) = 0.2 \times 0.8 + 0.3 \times 0.6$, and $D_2(\{B, C\}) = 0.3 \times 0.2 + 0.5 \times 0.7$. The dataset D_3 generated by C at time 3 is $D_3(\{A\}) = 0.008$, $D_3(\{B\}) = 0.012$, $D_3(\{C\}) = 0.045$, $D_3(\{A, B\}) = 0.408$, and $D_3(\{B, C\}) = 0.527$.

F. Ambiguous Origin of Mutational Matrix

A given dataset may be generated by different CPMCs at different times (Example 4). Besides, different CPMCs can even generate the same (infinite) sequence of datasets (Example 5).

Example 4: Let us consider the two CPMCs depicted in Fig. 7. Let C_1 be the chain on the left and C_2 be the one on the right. It is immediate to observe that the dataset D_2^1 generated by C_1 at time 2 is $D_2^1(\emptyset) = 0.01$ and $D_2^1(\{A\}) = 0.99$. Such dataset coincides with the dataset D_1^2 generated by C_2 at time 1.

We will come back to this example in the next section. The problem here lies in the inference of the probability of the self-loop on $\emptyset$. As a matter of fact, both C_1 and C_2 have the same jump chain, and we will prove that such jump chain can be inferred exactly.

When inferring a CPMC from a single dataset D, it may not be possible to accurately estimate the time at which the snapshot was taken. The example above shows that, in general, D may be supported by different CPMCs, which generate D at different times.

Unfortunately, in the worst case, two different CPMCs can generate the same datasets at each time instant (Example 5). This is not a problem when CPMCs are used as data generators

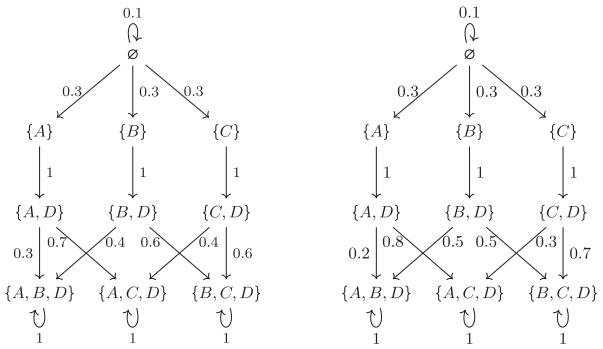


Fig. 8. Two CPMCs that generate the same datasets at each time instant.

because the CPMC is known, it for inference it means that in general uniqueness of the model cannot be guaranteed. In the next section we will prove that this can happen only in presence of convergences, i.e., when genotypes have many possible ancestors. In that case, we will provide a heuristic which allows us to infer one of the possible underlying jump chains.

Example 5: Let us consider the two CPMCs depicted in Fig. 8. They generate the same datasets at each time instants. As a matter of fact, the first 3 levels of the chains are equal with equiprobable branching, while at the last level for each node the sum of the probabilities of the incoming edges is the same in the two chains. The dataset D_3 at time 3 for both chains is $D_3(\{A,B,D\}) = 0.3 * 0.7,\ D_3(\{A,C,D\}) = 0.3 * 1.1,$ and $D_3(\{B,C,D\}) = 0.3 * 1.2$.

G. Properties of Cancer Progression Markov Chains

Let $f_i(S)$ be the event $X(i) = S \wedge X(i-1) \neq S \wedge \cdots \wedge X(0) \neq S$, i.e., the chain is in state S at time i and has never been in S before. Hence, $P[f_i(S)]$ is the probability of $f_i(S)$, i.e., the probability of reaching for the first time the vertex S after i steps. This is also known as the *first passage probability*. The probability of being in state $T \in Adj[S]^-$ for the first time after at most k steps, passing through the edge from S to T, can be expressed as

$$\sum_{i=1}^{k} P[f_i(T)] \times P[X(i-1) = S \mid f_i(T)]$$

Since C is acyclic this is equivalent to

$$\sum_{i=1}^{k} P[X(i-1) = S \wedge X(i) = T]$$

and can be computed on CPMCs as stated by the following lemma.

Lemma 2: Let $C = (\mathcal{S}, p)$ be a CPMC and let $S, T \in \mathcal{S}$ be such that $S \neq T$ and $p(S,T) > 0$. Then:

$$\sum_{i=1}^{k} P[f_i(T)] \times P[X(i-1) = S \mid f_i(T)]$$

$$= p(S,T) \times \sum_{j=0}^{k-1} P[X(j) = S] \tag{1}$$

Proof: See the Proofs section in the supplementary material. □

As a consequence, we get an alternative method to compute the probabilities on the jump chain $J(C)$. Let $height(C)$ be the length of the longest path in C, without crossing self-loops. Since C is acyclic and the normal genotype $\emptyset$ reaches any other genotype, $height(C)$ is the length of the longest path which does not uses self-loops from $\emptyset$ to a leaf in C.

Theorem 1: Let $C = (\mathcal{S}, p)$ be a CPMC and $k \geq height(C)$. Let $J(C) = (\mathcal{S}, jp)$ be the jump DTMC associated to C. Let $S, T \in \mathcal{S}$ with $S \neq T$ and $p(S,T) > 0$.

$$jp(S,T) = \frac{\sum_{i=1}^{k} P[f_i(T)] \times P[X(i-1) = S \mid f_i(T)]}{\sum_{W \in Adj^-[S]} \sum_{i=1}^{k} P[f_i(W)] \times P[X(i-1) = S \mid f_i(W)]} \tag{2}$$

Proof: See the Proofs section in the supplementary material. □

Notice that in (2) the denominator is just a normalization factor which ensures that the sum of the probabilities of the edges from S is 1.

H. The Inference Method

Let D_k be a dataset satisfying our data hypotheses, representing a snapshot of a tumor after k evolution steps. As discussed before, the number of steps in our context models the time elapsed from normality to the observed snapshot. We will see that in our method we do not assume to know the value of k. Assuming that D_k has been generated by a CPMC (i.e., by a model satisfying our model hypotheses) one may wonder whether it is possible to infer such a CPMC, i.e., a CPMC such that $D_k(S) = P[X(k) = S]$ for each genotype S, where k is not known a priori. To be more precise, since at this point of the construction, we do not want to add information to the dataset, we can say that we are interested in inferring the part of C that is visible from the dataset, i.e., the genotypes of C will be the normal genotype and the ones having positive frequency in D_k. Formally, this means that $C = (\mathcal{S}, p)$, where $\mathcal{S} = \{\emptyset\} \cup \{S \mid D_k(S) > 0\}$. If during the evolution there had been genotypes that have disappeared and are not represented in the dataset our method will not infer such genotypes, since it is our aim to reconstruct a model able to represent the current situation without introducing unobserved knowledge.

Lemma 2 provides a way to compute $p(S,T)$, but only when $\sum_{j=0}^{k-1} P[X(j) = S]$ is known. Unfortunately, there is no way to determine such a quantity from the dataset alone. On the other hand, if we consider $J(C)$ instead of C then Theorem 1 can be used to compute the transition probabilities—in some cases exactly, in general using some heuristics.

By definition, the topology of $J(C)$ is uniquely determined by the set of observed genotypes, as follows:

$$T \in Adj^-[S]$$

if and only if

$$S \subseteq T \wedge \forall T' \neq S, T' \neq T.\ (S \not\subseteq T' \vee T' \not\subseteq T)$$

According to Theorem 1, in order to infer $jp(S,T)$ the following probabilities must be estimated:

a. for each $T \in \mathcal{S} \setminus \{\emptyset\}$ and for each $i \in [1,k]$, the probability $P[f_i(T)]$;
b. for each $S \in \mathcal{S}$, for each $T \in Adj^-[S]$, and for each $i \in [1,k]$, the probability $P[X(i-1) = S \mid f_i(T)]$.

We say that there is a *convergence* in C whenever a genotype T has two different predecessors, that is, when there is a genotype T such that $|Pred^-[T]| > 1$. We distinguish two cases:

1) C (or equivalently, $J(C)$) has no convergences;
2) C (or equivalently, $J(C)$) has at least a convergence.

I. No Convergences

If C has no convergences, then for each $S \in \mathcal{S}$, for each $T \in Adj^-[S]$, and for each $i \in [1,k]$ it holds that $P[X(i-1) = S \mid f_i(T)] = 1$. This is trivial since S is the only predecessor of T. Hence, by Theorem 1 we get

$$jp(S,T) = \frac{\sum_{i=1}^{k} P[f_i(T)]}{\sum_{W \in Adj^-[S]} \sum_{i=1}^{k} P[f_i[W]]}$$

Since the denominator is just a normalization factor, we have to find a way to compute the numerator from D_k.

We can proceed by induction from the leaves to the root of C:

- if T is a "leaf" of C, i.e., $Adj^-[T] = \emptyset$, then:

$$\sum_{i=1}^{k} P[f_i(T)] = D_k(T);$$

- otherwise:
$\sum_{i=1}^{k} P[f_i(T)] = D_k(T) + \sum_{V \in Adj^-[T]} \sum_{i=1}^{k} P[f_i[V]]$.

As a consequence, we have proved the following corollary.

Corollary 1: Let $C = (\mathcal{S}, p)$ be an unknown CPMC without convergences and let D_k be a dataset generated from C at time k. The chain $J(C)$ can be uniquely inferred from D_k, provided that all the genotypes of C are represented in D_k.

In other terms, the fact that all the genotypes have to be represented in D_k means that the time instant at which the data are taken is neither too early, so that some genotypes have not yet been discovered, nor too late, so that some genotypes are no more present. Notice that we do not assume to know the value of k.

J. Convergences

From the above discussion, it emerges that in the case with convergences we have to find a way to estimate $P[X(i-1) = S \mid f_i(T)]$, for each $i \in [1,k]$. This means that for any $i \in [1,k]$ we have to estimate the probability that since we are for the first time in T at time i we were in S at time $i-1$. As already stated in the previous sections, Markov Chains are time homogeneous, but this is not in true in the general case for their reverse. So it is possible that $P[X(i-1) = S \mid f_i(T)] \neq P[X(j-1) = S \mid f_j(T)]$, for some $i, j \in [1,k]$. However, without any additional knowledge, the best one can do is approximate such values. In the following, we will approximate all the values uniformly with a single quantity denoted $Split(S,T)$. In this way, by Theorem 1 we get

$$jp(S,T) \approx \frac{Split(S,T) \times \sum_{i=1}^{k} P[f_i(T)]}{\sum_{W \in Adj^-[S]} Split(S,W) \times \sum_{i=1}^{k} P[f_i(W)]} \quad (3)$$

Again, the denominator is a normalization factor, so we focus on the numerator.

- $Split(S,T)$ is an approximation that we attribute to all the possible values of $P[X(i-1) = S \mid f_i(T)]$ and has to be computed by exploiting only D_k.
- $\sum_{i=1}^{k} P[f_i(T)]$ has to be computed by induction from the leaves to the root, but some more caution will be necessary with respect to the case without convergences.

We have already showed that no unique solution may exist in the presence of convergences, i.e., the function $Split(S,T)$ is not uniquely determined in general (see Fig. 8 and Example 5). Our heuristic for $Split(S,T)$ is based on the following simple considerations.

1) Since $Split(S,T)$ represents the probability of reaching T through S, $\sum_{X \in Pred^-[T]} Split(X,T) = 1$.
2) For $S, S' \in Pred^-[T]$, if S is more frequent than S' in the dataset, then it is more likely that T is reached from S than from S'.
3) To be more precise, in the previous item not only the frequencies of S and S' have to be taken into account, but also those of their ancestors.
4) Also, the number of outgoing edges from S and S' must be taken into account. If S has many outgoing edges, but S' reaches only T, then, intuitively, even if S and S' have the same frequency, the probability of reaching T from S should be lower than the probability of reaching T from S'.

Based on the above, we elaborate the following iterative definition for $\overline{Split(S,T)}$, that will be then normalized to obtain $Split(S,T)$:

$$\overline{Split(S,T)} = \begin{cases} \frac{D_k(\emptyset)}{|Adj^-[\emptyset]|} & \text{if } S = \emptyset \\ \frac{1}{|Adj^-[S]|} \times (D_k(S) + \sum_{U \in Pred^-[S]} \overline{Split(U,S)}) \end{cases}$$

Intuitively, S is assigned a weight proportional to its frequency in the dataset and, recursively, to the weight of its ancestors; then, such weight is uniformly distributed over S's outgoing edges. In principle, such distribution should be proportional to $jp(S,T)$, but since we are still in the process of evaluating it we apply a uniform distribution.

Once all the $\overline{Split(S,T)}$ have been computed, we can normalize them, thus obtaining the values for $Split(S,T)$.

In order to compute the values $\sum_{i=1}^{k} P[f_i(T)]$, we proceed iteratively from the leaves to the root. However, since a node can have many parents, we cannot assign all its probability to every parent. We use again the heuristic $Split$ to distribute such probability among all parents. In particular, we have:

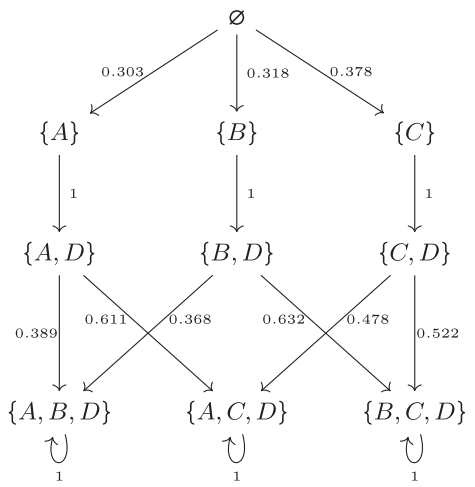


Fig. 9. Jump CPMC inferred from the dataset D_3 of Example 5.

- if T is a "leaf" of C, i.e., $Adj^-[T] = \emptyset$, then: $\sum_{i=1}^{k} P[f_i(T)] = D_k(T)$;
- otherwise: $\sum_{i=1}^{k} P[f_i(T)] = D_k(T) + \sum_{V \in Adj^-[T]} Split(T, V) \times \sum_{i=1}^{k} P[f_i[V]]$.

Finally, we can exploit (3) to get the probabilities $jp(S, T)$.

If the above heuristic is applied to a topology with no convergences, then all the $Split(S, T)$ are 1, hence the heuristic computes the same jump probabilities that can be obtained by applying the method described in the previous section for the special case without convergences.

Example 6: By applying (3) to the dataset D_3 of Example 5 we obtain the jump CPMC depicted in Fig. 9. This is the approximation we compute for the jump chains of the models in Fig. 8. We recall that both models in Fig. 8 are plausible generators for the dataset. Notice that despite the high symmetry of the dataset over the first 7 genotypes, the chain we extract is not completely symmetric. However, we are not inferring the self-loops. It is possible to define a CPMC with self-loop whose jump chain is that presented in Fig. 9 and that generates the dataset D_3 by solving a system of equations whose unknowns are the probabilities of the self-loops.

In the supplementary material we consider the effectiveness of the heuristic on data generated by different underlying models (Section Validation).

K. All Together

In order to prove the correctness of our method, we described it assuming that the dataset D_k has been generated from a CPMC C. We demonstrated when and with which accuracy we are able to infer $J(C)$ from D_k. Summing up, as long as all present genotypes are detected and the observations are exact, we proved the following results.

1) When there are no convergences, we exactly infer $J(C)$.
2) When there are convergences, if the probability of reaching T from S is time homogenous and has been estimated, e.g., using further data and expert knowledge, then we can exactly infer $J(C)$. Notice that such further information is necessary only for the nodes with convergences.
3) When there are convergences and no further information is available, we provided a heuristic for inferring a plausible $J(C)$.

Notice that the inference method is deterministic, i.e., on a given dataset it always returns the same CPMC.

L. The Implementation: CIMICE

The R package *CIMICE-R: (Markov) Chain Method to Infer Cancer Evolution* implements the above described methods. It takes in input a dataset in the form of a mutational matrix, i.e., a Boolean matrix representing altered genes in a collection of samples obtained with SCS DNA analysis.

CIMICE data processing and analysis can be divided in four section: input management, preliminary analysis of the dataset, graph topology reconstruction, chain weight computation, output presentation.

The tool requires a Boolean dataframe as input in which each column represents a gene, each row represents a sample (or a genotype), and each 0/1 represents whether a given gene is mutated in a given sample. It is possible to load this information from different file formats. The default one is the "CAPRI/CAPRESE" TRONCO [56] format: the file is a tab or space separated file; the first line starts with the string "s/g" (or any other word) followed by the list of genes (or loci) to be considered in the analysis. Each subsequent line begins with a sample identifier string, followed by the bit set representing its genotype. Another option is to directly define such data frame in R. In the case of data composed by samples with associated frequencies, it is possible to use an alternative format that we call *CAPRIpop*, consisting of unique samples as rows and respective frequencies reported in a special column named `freq`. Finally, to extend CIMICE interoperability with different variant callers, it is possible to compute mutational matrices directly from Mutation Annotation Format (MAF) files. The definition of the variant calling pipeline and the related error management is left to the user and is out of the scope of this manuscript. In some of the above presented examples we show how a preprocessing tool such as SiCloneFit can be exploited to mitigate experimental errors.

The tool includes simple functions to quickly analyze the distributions of mutations among genes and samples. Correlation plots are also available. In case of huge dataset, it could be necessary to focus only on a subset of the input samples or genes. CIMICE provides an easy way to do so when the goal is to use the most (or least) mutated samples and/or genes.

The subsequent stage's goal is to obtain the topology for the final Cancer Progression Markov Chain. Once the topology has been computed, it can be plotted, e.g., using igraph. Finally, the probabilities that label the edge of the jump chain are computed. The tool first computes the $\overline{Split(S, T)}$'s. These are called UP weights in the implementation. Then, these are normalized to obtain the $Split(S, T)$'s (called normalized UP weights). From these, the probabilities can be derived (also called normalized DOWN weights).

In order to show the results of the analysis exploiting different libraries, three output methods are provided. These libraries improve on the default igraph output visualization.

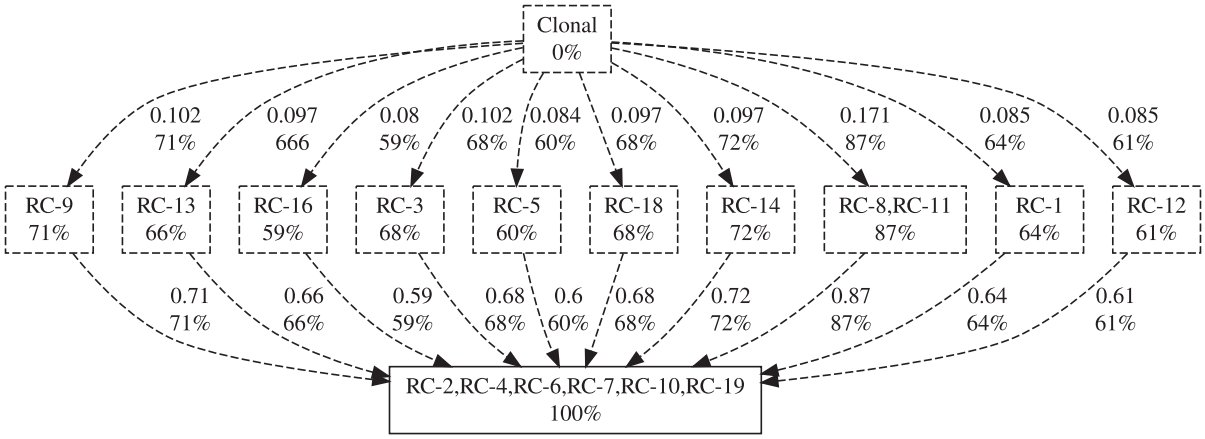


Fig. 10. Our method's results on the dataset from [20]. The False positive, False Negative and Missing Value Rates reported in the original manuscript for this particular dataset are 2.67×10^{-5}, 0.1643, and 0.2117, respectively. Two different stages are clearly separated, even if the scarcity and noisiness of data does not allow our method to establish a preferred progression.

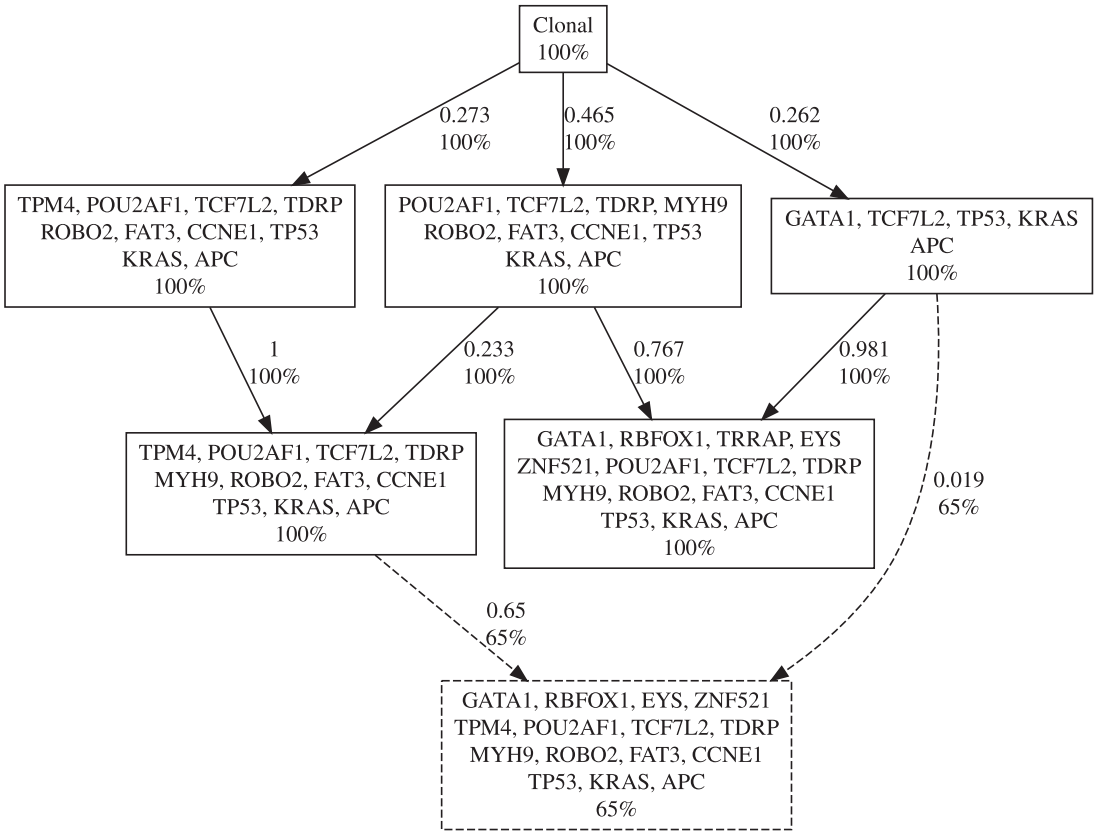


Fig. 11. In this example, we consider the dataset CRC1 together with the False Positive (0.0152), False Negative (0.0789), and Missing Value (0.0671) rates from [45]. In this case our method reconstructs several progression trajectories, that are mostly in agreement with the subdivision between genes present in metastatic and non-metastatic gene assets given in the original paper.

The computational complexity of the tool resides in two steps, the compression of the initial dataset to extract the frequency of each genotype and the computation of the DAG's topology. These two steps have computational cost $O(|\mathcal{C}||G|)$ and $O(|\mathcal{S}|^2|G|)$ respectively, where $|\mathcal{C}|$ is the number of samples present in the dataset (e.g., the sequenced cells), $|G|$ is the number of different possible mutations, and $|\mathcal{S}|$ is the number of unique genotypes present in the dataset. These dependencies allow this method to scale well even for large dataset if the number of unique genotypes is limited. Clustering and filtering of mutations can be taken into consideration to reduce the $|G|$ term if needed. All other steps have lower computational cost, note that the bootstrapping approach requires to compute a model for each resampling.

An in depth analysis of the required number of samples required for successfully running the method is detailed in the supplementary material in Section Sample Size Evaluation. A script is provided to compute such requirement under user specified conditions. These allow to select the confidence of estimating the genotypes' probabilities (and therefore the DAG structure) with an error below a given threshold. In addition, the methods can also consider the effect of random disturbances in the data with respect to the required sample size.

All other aspects required for usage, with running examples, are included in the detailed user manual provided at [8].

References

[1] P. C. Nowell, "The clonal evolution of tumor cell populations," *Science*, vol. 194, no. 4260, pp. 23–28, 1976.

[2] A. Davis, R. Gao, and N. Navin, "Tumor evolution: Linear, branching, neutral or punctuated?," *Biochimica et Biophysica Acta (BBA)- Rev. Cancer*, vol. 1867, no. 2, pp. 151–161, 2017.

[3] R. Schwartz and A. A. Schäffer, "The evolution of tumour phylogenetics: Principles and practice," *Nature Rev. Genet.*, vol. 18, no. 4, 2017, Art. no. 213.

[4] R. Schwartz, "Computational models for cancer phylogenetics," *Bioinf. Phylogenetics: Seminal Contributions Bernard Moret*, pp. 243–275, 2019.

[5] N. Navin et al., "Tumour evolution inferred by single-cell sequencing," *Nature*, vol. 472, pp. 90–95, Apr. 2011.

[6] H. Chen and X. He, "The convergent cancer evolution toward a single cellular destination," *Mol. Biol. Evol.*, vol. 33, no. 1, pp. 4–12, 2016.

[7] K. J. Pienta, E. U. Hammarlund, R. Axelrod, S. R. Amend, and J. S. Brown, "Convergent evolution, evolving evolvability, and the origins of lethal cancerevolving evolvability and the origins of lethal cancer," *Mol. Cancer Res.*, vol. 18, no. 6, pp. 801–810, 2020.

[8] N. Rossi, "CIMICE-R: (Markov) chain method to infer cancer evolution," 2021, [Online]. Available: https://bioconductor.org/packages/release/bioc/html/CIMICE.html, doi: 10.18129/B9.bioc.CIMICE.

[9] R. Desper, F. Jiang, O.-P. Kallioniemi, H. Moch, C. H. Papadimitriou, and A. A. Schäffer, "Inferring tree models for oncogenesis from comparative genome hybridization data," *J. Comput. Biol.*, vol. 6, no. 1, pp. 37–51, 1999.

[10] K. I. Kim and R. Simon, "Using single cell sequencing data to model the evolutionary history of a tumor," *BMC Bioinf.*, vol. 15, 2014, Art. no. 27.

[11] E. M. Ross and F. Markowetz, "OncoNEM: Inferring tumor evolution from single-cell sequencing data," *Genome Biol.*, vol. 17, no. 1, 2016, Art. no. 69.

[12] K. Jahn, J. Kuipers, and N. Beerenwinkel, "Tree inference for single-cell data," *Genome Biol.*, vol. 17, no. 1, 2016, Art. no. 86.

[13] S. Bamford et al., "The cosmic (catalogue of somatic mutations in cancer) database and website," *Brit. J. Cancer*, vol. 91, no. 2, pp. 355–358, 2004.

[14] 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 Biol.*, vol. 18, no. 1, 2017, Art. no. 178.

[15] S. Ciccolella et al., "Inferring cancer progression from single-cell sequencing while allowing mutation losses," *Bioinf.*, vol. 37, no. 3, pp. 326–333, 2021.

[16] H. Zafar, N. Navin, K. Chen, and L. Nakhleh, "SiCloneFit: Bayesian inference of population structure, genotype, and phylogeny of tumor clones from single-cell genome sequencing data," *Genome Res.*, vol. 29, pp. 1847–1859, 2019.

[17] A. Kozlov et al., "CellPhy: Accurate and fast probabilistic inference of single-cell phylogenies from scDNA-seq data," *Genome Biol.*, vol. 23, no. 1, 2020.

[18] P. Eirew et al., "Dynamics of genomic clones in breast cancer patient xenografts at single-cell resolution," *Nature*, vol. 518, no. 7539, 2015, Art. no. 422.

[19] C. Yu et al., "Discovery of biclonal origin and a novel oncogene SLC12A5 in colon cancer by single-cell sequencing," *Cell Res.*, vol. 24, no. 6, 2014, Art. no. 701.

[20] X. Xu et al., "Single-cell exome sequencing reveals single-nucleotide mutation characteristics of a kidney tumor," *Cell*, vol. 148, no. 5, pp. 886–895, 2012.

[21] C. Gawad, W. Koh, and S. R. Quake, "Dissecting the clonal origins of childhood acute lymphoblastic leukemia by single-cell genomics," in *Proc. Nat. Acad. Sci. USA*, vol. 111, no. 50, pp. 17 947–17 952, 2014.

[22] J. Kuipers, K. Jahn, and N. Beerenwinkel, "Advances in understanding tumour evolution through single-cell sequencing," *Biochimica et Biophysica Acta*, vol. 1867, pp. 127–138, 2017.

[23] H. Zafar, Y. Wang, L. Nakhleh, N. Navin, and K. Chen, "Monovar: Single-nucleotide variant detection in single cells," *Nature Methods*, vol. 13, no. 6, pp. 505–507, 2016.

[24] X. Dong et al., "Accurate identification of single-nucleotide variants in whole-genome-amplified single cells," *Nature Methods*, vol. 14, no. 5, pp. 491–493, 2017.

[25] J. Singer, J. Kuipers, K. Jahn, and N. Beerenwinkel, "Single-cell mutation identification via phylogenetic inference," *Nature Commun.*, vol. 9, no. 1, 2018, Art. no. 5144.

[26] L. J. Luquette, C. L. Bohrson, M. A. Sherman, and P. J. Park, "Identification of somatic mutations in single cell DNA-seq using a spatial model of allelic imbalance," *Nature Commun.*, vol. 10, no. 1, pp. 1–14, 2019.

[27] A. Roth et al., "Clonal genotype and population structure inference from single-cell tumor sequencing," *Nature Methods*, vol. 13, no. 7, 2016, Art. no. 573.

[28] S. Miura, L. A. Huuki, T. Buturla, T. Vu, K. Gomez, and S. Kumar, "Computational enhancement of single-cell sequences for inferring tumor evolution," *Bioinformatics*, vol. 34, no. 17, pp. i917–i926, 2018.

[29] D. Posada, "CellCoal: Coalescent simulation of single-cell sequencing samples," *Mol. Biol. Evol.*, vol. 37, no. 5, pp. 1535–1542, 2020.

[30] Z. Yu, F. Du, X. Sun, and A. Li, "SCSsim: An integrated tool for simulating single-cell genome sequencing data," *Bioinformatics*, vol. 1–2, 2019, Art. no. 359.

[31] S. Salehi, A. Steif, A. Roth, S. Aparicio, A. Bouchard-Côté, and S. P. Shah, "ddClone: Joint statistical inference of clonal populations from single cell and bulk tumour sequencing data," *Genome Biol.*, vol. 18, 2017, Art. no. 44.

[32] D. Ramazzotti, A. Graudenzi, L. De Sano, M. Antoniotti, and G. Caravagna, "Learning mutational graphs of individual tumour evolution from single-cell and multi-region sequencing data," *BMC Bioinf.*, vol. 20, no. 1, 2019, Art. no. 210.

[33] 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 Commun.*, vol. 10, no. 1, 2019, Art. no. 2750.

[34] S. Malikic et al., "PhISCS: A combinatorial approach for subperfect tumor phylogeny reconstruction via integrative use of single-cell and bulk sequencing data," *Genome Res.*, vol. 29, pp. 1860–1877, 2019.

[35] L. Baghaarabani, S. Goliaei, M.-H. Foroughmand-Araabi, S. P. Shariatpanahi, and B. Goliaei, "Conifer: Clonal tree inference for tumor heterogeneity with single-cell and bulk sequencing data," *BMC Bioinf.*, vol. 22, 2021, Art. no. 416.

[36] D. Lähnemann et al., "Eleven grand challenges in single-cell data science," *Genome Biol.*, vol. 21, no. 1, pp. 1–35, 2020.

[37] B. Lim, Y. Lin, and N. Navin, "Advancing cancer research and medicine with single-cell genomics," *Cancer Cell*, vol. 37, no. 4, pp. 456–470, 2020.

[38] H. Zafar, N. Navin, L. Nakhleh, and K. Chen, "Computational approaches for inferring tumor evolution from single-cell genomic data," *Curr. Opin. Syst. Biol.*, vol. 7, pp. 16–25, 2018.

[39] D. Tsoucas and G.-C. Yuan, "Recent progress in single-cell cancer genomics," *Curr. Opin. Genet. Develop.*, vol. 42, pp. 22–32, 2017.

[40] F. Vandin, "Computational methods for characterizing cancer mutational heterogeneity," *Front. Genet.*, vol. 8, 2017, Art. no. 83.

[41] A. Davis and N. E. Navin, "Computing tumor trees from single cells," *Genome Biol.*, vol. 17, no. 1, 2016, Art. no. 113.

[42] C. Gawad, W. Koh, and S. R. Quake, "Single-cell genome sequencing: Current state of the science," *Nature Rev. Genet.*, vol. 17, no. 3, 2016, Art. no. 175.

[43] N. Beerenwinkel, R. F. Schwartz, M. Gerstung, and F. Markowetz, "Cancer evolution: Mathematical models and computational inference," *Systematic Biol.*, vol. 64, no. 1, pp. e1–e25, 2015.

[44] N. E. Navin, "Cancer genomics: One cell at a time," *Genome Biol.*, vol. 15, 2014, Art. no. 452.

[45] M. L. Leung et al., "Single-cell DNA sequencing reveals a late-dissemination model in metastatic colorectal cancer," *Genome Res.*, vol. 27, no. 8, pp. 1287–1299, 2017.

[46] J. Yu, M. Liu, H. Liu, and L. Zhou, "GATA1 promotes colorectal cancer cell proliferation, migration and invasion via activating akt signaling pathway," *Mol. Cellular Biochem.*, vol. 457, pp. 191–199, 2019.

[47] I. Peters et al., "Decreased mRNA expression of GATA1 and GATA2 is associated with tumor aggressiveness and poor outcome in clear cell renal cell carcinoma," *Target. Oncol.*, vol. 10, pp. 267–275, 2015.

[48] D. W. Ruff, D. M. Dhingra, K. Thompson, J. A. Marin, and A. T. Ooi, "High-throughput multimodal single-cell targeted DNA and surface using the mission bio tapestri platform," *Single-Cell Protein Anal., Methods Mol. Biol.*, pp. 171–188, 2022.

[49] M. Kwiatkowska, G. Norman, and D. Parker, "Prism 4.0: Verification of probabilistic real-time systems," in *Proc. 23rd Int. Conf. Comput. Aided Verification*, Snowbird, UT, USA, 2011, pp. 585–591.
[50] A. Casagrande, T. Dreossi, and C. Piazza, "Hybrid automata and ϵ-analysis on a neural oscillator," in *Proc. 1st Int. Workshop Hybrid Syst. Biol.*, 2012, pp. 58–72.
[51] O. E. Ogundijo and X. Wang, "Seqclone: Sequential monte carlo based inference of tumor subclones," *BMC Bioinf.*, vol. 20, no. 1, pp. 1–15, 2019.
[52] N. Misra, E. Szczurek, and M. Vingron, "Inferring the paths of somatic evolution in cancer," *Bioinformatics*, vol. 30, no. 17, pp. 2456–2463, 2014.
[53] D. Ramazzotti et al., "CAPRI: Efficient inference of cancer progression models from cross-sectional data," *Bioinformatics*, vol. 31, no. 18, pp. 3016–3026, 2015.
[54] J. R. Norris and J. R. Norris, *Markov Chains*. Cambridge, U.K.: Cambridge Univ. Press, 1998, no. 2.
[55] J. G. Kemeny and J. L. Snell, *Finite Markov Chains*. Berlin, Germany: Springer, 1976.
[56] L. De Sano et al., "TRONCO: An r package for the inference of cancer progression models from heterogeneous genomic data," *Bioinformatics*, vol. 32, no. 12, pp. 1911–1913, 2016.

Nicolò Rossi received the MSc degree in computer science from the University of Udine, Italy, in 2021. He is currently working toward the PhD with the CTSB Group of ETH Zörich's D-BSSE (Basel, Switzerland). His current research interests include the application of computer science, especially formal methods, symbolic regression, and machine learning, to bioinformatics, and systems biology.

Nicola Gigante received the PhD degree in computer science from the University of Udine, in 2019 with a thesis on the expressiveness and complexity of timeline-based planning languages. He visited the University of Western Australia in Perth. After two years as a postdoc in Udine, he moved to the Free University of Bozen-Bolzano as a researcher. His research interest includes placed at the border between formal methods and artificial intelligence, with a focus on temporal reasoning and verification.

Nicola Vitacolonna received the PhD degree in computer science from the University of Udine, Italy, where he has been employed as a researcher since 2007. His research interests include bioinformatics algorithms and data modeling/data management applications, in the context of life sciences and humanities. He got interested in applications of formal methods to the verification of security protocols. He is currently working with Statice GmbH as a privacy researcher.

Carla Piazza received the master's degree in mathematics and the PhD degree in computer science. Since December 2021 she is full professor of computer science with the University of Udine, Italy. She is coordinator of the laboratory of Computational Biology and Bioinformatics with the Departments of Mathematics, Computer Science, and Physics, University of Udine. Her main research interests include concern system biology, formal methods, model checking, hybrid systems, and information flow security.

Open Access provided by 'Università degli Studi di Udine' within the CRUI CARE Agreement