International Journal of
Environmental Research and Public Health

MDPI

Article

Reconstructing Clonal Evolution—A Systematic Evaluation of Current Bioinformatics Approaches

Sarah Sandmann [1,*], Silja Richter [1], Xiaoyi Jiang [2] and Julian Varghese [1]

[1] Institute of Medical Informatics, University of Münster, 48149 Münster, Germany
[2] Department of Computer Science, University of Münster, 48149 Münster, Germany
* Correspondence: sarah.sandmann@uni-muenster.de

Abstract: The accurate reconstruction of clonal evolution, including the identification of newly developing, highly aggressive subclones, is essential for the application of precision medicine in cancer treatment. Reconstruction, aiming for correct variant clustering and clonal evolution tree reconstruction, is commonly performed by tedious manual work. While there is a plethora of tools to automatically generate reconstruction, their reliability, especially reasons for unreliability, are not systematically assessed. We developed clevRsim—an approach to simulate clonal evolution data, including single-nucleotide variants as well as (overlapping) copy number variants. From this, we generated 88 data sets and performed a systematic evaluation of the tools for the reconstruction of clonal evolution. The results indicate a major negative influence of a high number of clones on both clustering and tree reconstruction. Low coverage as well as an extreme number of time points usually leads to poor clustering results. An underlying branched independent evolution hampers correct tree reconstruction. A further major decline in performance could be observed for large deletions and duplications overlapping single-nucleotide variants. In summary, to explore the full potential of reconstructing clonal evolution, improved algorithms that can properly handle the identified limitations are greatly needed.

Keywords: clonal evolution; single-nucleotide variant; copy number variant; simulation

Citation: Sandmann, S.; Richter, S.; Jiang, X.; Varghese, J. Reconstructing Clonal Evolution—A Systematic Evaluation of Current Bioinformatics Approaches. *Int. J. Environ. Res. Public Health* **2023**, *20*, 5128. https://doi.org/10.3390/ijerph20065128

Academic Editor: Paul B. Tchounwou

Received: 8 February 2023
Revised: 4 March 2023
Accepted: 13 March 2023
Published: 14 March 2023

1. Introduction

According to the International Agency for Research on Cancer, more than 19.2 million new cases of cancer were registered in 2020. Accounting for almost 10 million deaths, it is one of the world's leading causes of death [1]. The survival rates are influenced by many factors, e.g., the type of cancer or the stage at which it was diagnosed. For prostate cancer, as an example, the 1-year survival rate is 97% (5-year survival rate 87%), while it is only 41% for lung cancer (5-year survival rate 17%; data for the United Kingdom [2]). Similarly, the probability of recurrence considerably differs: for childhood acute myeloid leukemia (AML), it is reported to vary between 9% and 29% [3] and, for melanoma, even between 5% and 67% (within 180 days) dependent on stage [4].

The causes of cancer are diverse. In addition to genetic predisposition, environmental factors play a major role. Exposition to physical, chemical, or biological carcinogens favor the development of mutations, which can lead to the transformation of normal cells to tumor cells [5]. For many types of cancer, the precise mutational characterization of a tumor is essential, influencing diagnosis, prognosis, and therapy. In the case of myeoldysplastic syndromes (MDS), the Revised International Prognostic Scoring System (IPSS-R)—considering, among others, the presence of cytogenetic abnormalities—is commonly used to stratify patients for high vs. low risk of AML transformation [6]. A recent study conducted by [7] provides evidence that the further subgrouping of low-risk MDS patients can be performed based on their mutational profile. For Burkitt lymphoma (BL), there is evidence that the presence of a double-hit event, that is, two variants affecting the gene *TP53*, is associated

with relapse [8]. For acute lymphoblastic leukemia (ALL), therapy with a tyrosine kinase inhibitor, such as Imatinib, is assumed to be highly beneficial in the presence of an BCR:ABL translocation [9].

In addition to characterizing a tumor with respect to the presence or absence of certain variants at the time of initial diagnosis, monitoring the evolution of the mutational profile in the course of a disease is equally important. It allows for the early detection of newly developing, highly aggressive subclones that might be resistant to therapy and can potentially lead to relapse [8] (see Figure 1).

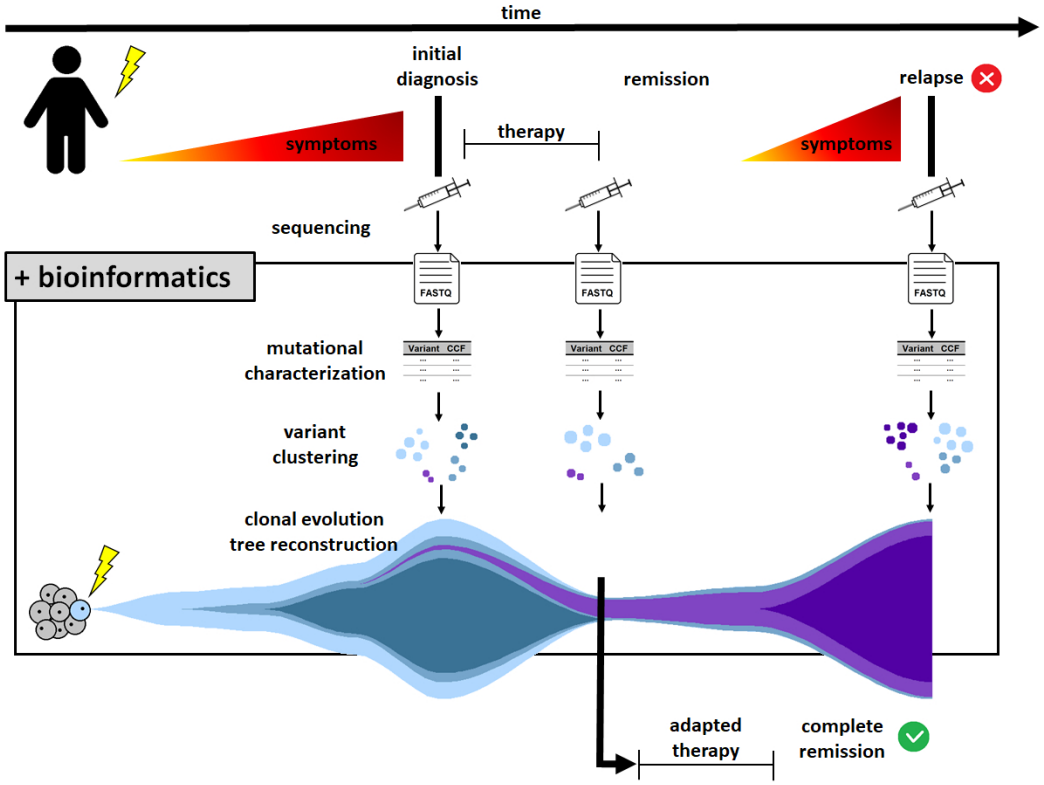

Figure 1. The role of bioinformatics in precision oncology. When performing sequencing experiments in the course of disease, bioinformatics analyses allow for performing mutational characterization of a tumor. Subsequently, data can be used to reconstruct clonal evolution, including variant clustering and clonal evolution tree reconstruction. Thereby, aggressive subclones (purple), which are resistant to therapy and may lead to relapse, can be identified early and therapy adapted accordingly.

To determine a tumor's mutational profile and its development over time, two main approaches can be distinguished: bulk and single-cell DNA sequencing (scDNA-seq). In terms of bulk sequencing, small variants are determined by, e.g., targeted or whole-exome sequencing (WES). The cancer cell fraction (CCF) estimated for each variant represents its average abundance across all the analyzed cells. The information on which variants co-occur within each cell is not directly available from the data. Instead, deconvolution has to be performed to decipher the underlying clonal populations and, subsequently, reconstruct clonal evolution. High intra-tumor heterogeneity [10,11] poses a major challenge for this approach. Unique results cannot always be retrieved. On the contrary, single-cell sequencing allows—in theory—to precisely determine the mutation profile of every analyzed cell. Technical challenges, e.g., allelic drop-out or amplification bias leading to low sensitivity [12,13], however, still hamper valid variant calling and the subsequent reconstruction of clonal evolution using this technique.

Apart from small variants, structural and copy number variants (CNVs) play a major role in clonal evolution. In addition to next-generation sequencing techniques, these variants are commonly detected using microarrays, fluorescence in situ hybridization

(FISH), and/or karyotypin. While microarrays, such as SNP-arrays, only allow for a rough estimation of CCFs on the bulk level, FISH and karyotyping provide data on single-cell level. However, the resolution is considerably lower compared to scDNA-seq (100–200 interphase nucleoli for FISH, 10–25 metaphases for karyotyping). However, the valid reconstruction of clonal evolution is only possible if the data on both small and large variants are integrated and jointly evaluated [14,15].

As clonal evolution thrives to analyze cancer development over time, the samples are usually collected at several time points within the course of the disease. Solid tumors additionally hold the option for collecting multi-regional samples—within one tumor and/or at different sites (e.g., primary tumor, lymph node, and metastasis).

A plethora of approaches that perform the reconstruction of clonal evolution fully automatically is available. In a previous study, we aimed at assessing the performance of tools for the analysis of bulk sequencing data, considering two sets of well-characterized real data from the patients with MDS [16] and BL [8]. Our analysis indicated that the performance of the currently available approaches does not warrant their safe usage in research or clinical routine. The reasons for these observations, however, remained unclear. The data neither allowed us to identify those characteristics that were primarily responsible for the unreliable performance of the evaluated tools, nor to develop countermeasures [17].

To the best of our knowledge, a systematic evaluation of the tools reconstructing clonal evolution, exploring their strengths and weaknesses in relation to, e.g., the number of clones or time points, has not been performed yet. To provide a means for systematic evaluation, we developed clevRsim—a simulation approach for clonal evolution in R. clevRsim has been designed to simulate single-nucleotide variants (SNVs) in bulk sequencing data as well as (overlapping) CNVs on the basis of a user-definable clonal evolution pattern. Simulated data can then be used as an input for tools performing variant clustering and clonal evolution tree reconstruction. Considering different levels of difficulty, we simulated 88 data sets with 10 patients each. We perform a detailed systematic evaluation of nine tools for variant clustering and four tools for clonal evolution tree reconstruction that were previously identified by systematic search.

2. Materials and Methods

2.1. clevRsim

An overview of the simulation tool clevRsim is provided in Figure 2 (screenshots provided in Appendix A.1. clevRsim, Figure A1–A4).

Creating a simulation in clevRsim can be divided into two main steps: (1) simulating the phylogeny and (2) simulating the variants. A download option allows for exporting all the simulated data, including a CCF matrix (information on the phylogeny and CCFs at every simulated time point), variant calls (SNVs and CNVs), visualization of clonal evolution, and selected input parameters. Optionally, files can be uploaded to clevRsim again to retrieve a previous simulation. Our simulation tool is programmed in R [18]. A graphical user interface for intuitive usage also used by non-computer scientists was developed using R shiny. The software is freely available at https://github.com/sandmanns/clevRsim, accessed on 4 March 2023.

2.1.1. Simulating the Phylogeny

When executing a simulation using clevRsim, the phylogeny is simulated first. As an input, the algorithm takes into account: the model of evolution, the number of time points, clones and variants (basic settings), the detection threshold, and the minimum clonal distance (advanced settings).

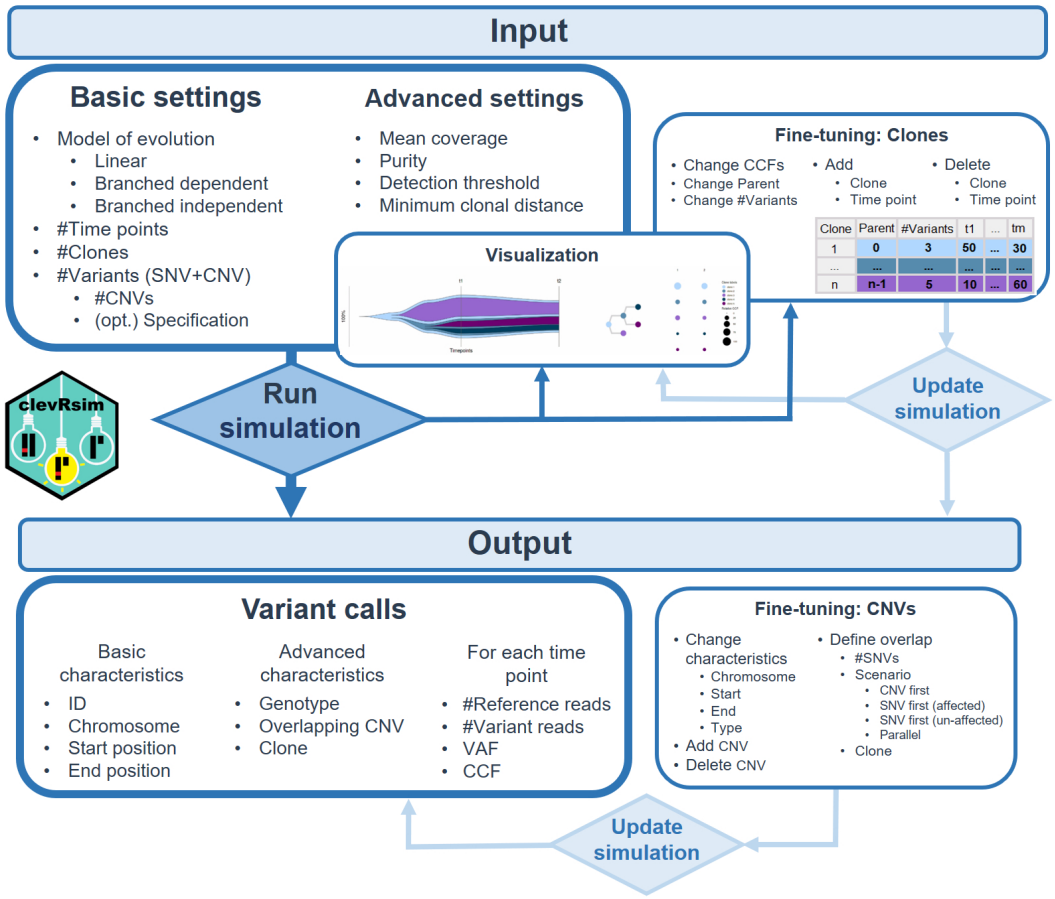

Figure 2. Overview of the simulation tool clevRsim. A graphical user interface is provided to generated customizable clonal evolution patterns. Basic information, including clones, their parents, and the cancer cell fraction at every time point, and a visualization of clonal evolution are provided. Additionally, variant calls—supporting SNVs as well as (overlapping) CNVs—are simulated. Output generated using clevRsim may serve as direct input for tools performing variant clustering and clonal evolution tree reconstruction.

We consider linear, branched dependent, and branched independent evolution [8]. Additional models 'neutral' and 'punctuated' evolution, previously described by [19], can be easily simulated by defining a high number of clones for branched dependent (=neutral) or linear evolution (=punctuated).

The number of time points n_{tp} defines the number of samples collected in the course of disease (default: 3 [1–50]). The number of clones n_c corresponds to the number of clusters to be simulated (default: 5 [1–50]). Each cluster is assumed to define a distinct population of tumor cells characterized by the same mutational profile. The number of variants n_v defines the overall number of both SNVs and CNVs to be simulated and split across the simulated clones (default: 20 [1–200]). The detection threshold min_{th} defines the minimum CCF at which a clone can still be detected (default: 2 [0–50]), while the minimum clonal distance min_{dist} defines the minimum CCF separating the two clones (default: 4 [0–50]).

First, the CCFs of the founding clone c_1 are simulated at random for all time points n_{tp} (uniform distribution). Subsequently, the CCFs of the remaining $n_c - 1$ clones are successively simulated considering the underlying model of evolution: For linear evolution, a new clone c_i always develops from the latest clone simulated c_{i-1}. For branched dependent evolution, clone c_2 develops linearly from c_1, while c_3 develops as the first dependent branch (parent: c_1). All subsequent clones randomly develop as either linear or branched dependent. For branched independent evolution, clone c_2 develops as the first independent branch (parent: normal cells). All subsequent clones randomly develop as either linear or branched independent.

For every new clone, its CCFs are simulated under the following constraints: A child clone cannot exceed its parent at any time. The sum over all dependent child clones cannot exceed their joined parent. The sum over all independent clones cannot exceed 100%. An eradicated clone cannot re-appear, i.e., once a clone's CCF has fallen to zero, it remains zero for all subsequent time points. The CCF of each clone has to be $\geq min_{th}$ for at least one time point. The Δ CCF for two clones has to be $\geq min_{dist}$ for at least one time point. If one of the constraints is violated, a new clone is simulated at random. If no solution is found after 100 attempts, a new simulation, including a new founding clone, is executed (maximum 700 attempts). Finally, if a phylogeny was successfully generated, one variant is assigned to every clone. The remaining $n_v - n_c$ variants are randomly split.

The results of the algorithm simulating the phylogeny are visualized using the R/Bioconductor package clevRvis [20]. Additionally, a CCF table is displayed, providing an option for fine-tuning the simulation on a clonal level. All major characteristics of the clonal evolution (parental relations, number of variants per clone, and CCFs) can be changed, and the clones and time points can be added as well as deleted.

2.1.2. Simulating Variants

As a second step when executing a simulation with clevRsim, the genetic variants are simulated under the infinite-allele assumption [21]. The simulation of CNVs is executed first. As an input, the algorithm takes into account the previously simulated phylogeny including the CCFs of every clone at every time point and the number of variants assigned to each clone. When simulating SNVs, the mean coverage, purity, and minimum clonal distance (advanced settings), as well as the simulated CNVs, are additionally considered.

For each CNV, a genomic location is generated at random (chromosomes 1–22, positions according to GRCh37, random CNV length, but $\leq$10,000 bp). clevRsim supports the simulation of deletions, duplications, and loss of heterozygosity (LOH). Every selected type of CNV is simulated at least once. Subsequently, the types of the remaining CNVs are determined at random. The CNVs are randomly assigned to the simulated clones.

Subsequently, the SNVs are generated. In the absence of CNVs, SNVs are simulated completely at random (chromosomes 1–22, positions according to GRCh37). In the presence of CNVs, all the SNVs are initially simulated as "non-overlapping". If the genomic locations of the simulated SNVs overlap the CNVs by chance, the simulation of the corresponding SNVs is repeated. The reads are simulated on the basis of a log–normal distribution ($\mu = log(\text{user-defined mean coverage})$, $\sigma = log(0.7)$; the standard deviation σ was determined empirically to best approximate the coverage distribution, evaluating the exemplary targeted and WES data [8,16]). The variant allele frequency (VAF) is determined as $VAF = CCF/2$. Random variation is added ($\mathcal{N}(0,1)$) to account for imperfect sequencing.

The results of the simulated variants are tabulated. For every call, information on the basic and advanced characteristics—also including the number of simulated reads and VAF for SNVs—is provided. Based on these results, the overlap of CNVs and SNVs can be configured (Fine-tuning: CNVs). This includes the number of overlapping SNVs, the affected clone as well as the scenario of overlap. We distinguish four scenarios: CNV first, SNV first affected, SNV first un-affected, and parallel [14] (the influence of the different scenarios on the genotype is summarized in Appendix A.1. clevRsim, Table A1). According to the formula provided in [14], the VAF is adjusted with respect to the scenario and CNV type. For example, in the case of a deletion present in the scenario 'SNV first un-affected', we expect to observe the cells characterized by SNV (genotype AB) and SNV+CNV (genotype B). Assuming that the CCF of clone c_{SNV} is 100% and the CCF of clone $c_{SNV+CNV}$ is 50%, the adjusted VAF (that is assumed to be observed in a sequencing experiment) is 67%, not 50%. The number of simulated reads is adjusted accordingly. The overlapping SNVs are randomly positioned within the corresponding CNVs. A clone is assigned at random, restricted only by the selected scenario of overlap and the number of variants per clone.

2.2. Simulated Data Sets

Altogether, 88 data sets were simulated using clevRsim. An overview of the data sets and their main characteristics is provided in Table 1. To account for random variation in the simulated data and their influence on the tools' performance, we simulated ten patients per data set leading to a total of 880 simulated patients.

Table 1. Data sets simulated using clevRsim. Altogether, 88 data sets with 10 patients each were simulated. Simulated sets were used to evaluate performance of variant clustering tools in a basic (from sim01 to sim40) and an advanced setting (from sim41 to sim 52). Additionally, the performance of tools for clonal evolution tree reconstruction was evaluated (from sim01 to sim20 and from sim53 to sim88).

Data Set	Model of Clonal Evolution	#Time Points	#Clones	#SNVs	Mean Coverage	#CNVs
sim01–sim10	linear	1–10	5	20	300x	0
sim11–sim20	linear	3	1–10	20	300x	0
sim21–sim30	linear	3	5	5–50 (step 5)	300x	0
sim31–sim40	linear	3	5	20	10x, 20x, 50x, 100x, 200x, 300x, 500x, 1000x, 1500x, 2000x	0
sim41–sim44	linear	3	5	20	300x	6 (del)
sim45–sim48	linear	3	5	20	300x	6 (dup)
sim49–sim52	linear	3	5	20	300x	6 (LOH)
sim53–sim62	branched dependent	1–10	5	20	300x	0
sim63–sim70	branched dependent	3	3–10	20	300x	0
sim71–sim80	branched independent	1–10	5	20	300x	0
sim81–sim88	branched independent	3	3–10	20	300x	0

The simulated data sets can be split into 3 groups. The first group—from sim01 to sim40—serves as a basic evaluation of the tools performing variant clustering. The data are characterized by a varying number of time points (from 1 to 10), number of clones (from 1 to 10), number of SNVs (from 5 to 50 in steps of 5), and mean coverage (from 10x to 2000x). For all sets, we chose 'linear' as the model of evolution. CNVs were not simulated.

The second group—from sim41 to sim52—serves an advanced evaluation of the tools performing variant clustering, also considering the present CNVs. As we could not identify any algorithm that was capable of clustering the SNVs and non-overlapping CNVs, only overlapping CNVs were simulated. Every data set contains 20 SNVs and 6 overlapping CNVs: 3 CNVs are overlapping 1 SNV each, 2 further CNVs are overlapping 2 SNVs each, and 1 CNV is overlapping 3 SNVs. Thereby, 10 out of 20 simulated SNVs per data set are affected by the overlapping CNVs. For every data set, we chose a different scenario of overlap (CNV first, SNV first affected, SNV first un-affected, and parallel).

The third group—from sim53 to sim88—serves, together with simulated data sets from sim01 to sim20, an evaluation of the tools performing clonal evolution tree reconstruction. Commonly, the tools evaluate the information on clones and their CCFs at every time point. Therefore, we evaluate a varying number of time points (from 1 to 10) and number of clones (from 3 to 10), considering all three models of evolution: linear, branched dependent, and branched independent. Of note, the branched dependent and branched independent evolution cannot be simulated in the presence of only 1 or 2 clones as a branch requires at least 3 clones.

2.3. Tools for Variant Clustering

We base our systematic evaluation of tools for the reconstruction of clonal evolution on a previously performed evaluation using two sets of real data [17]. In our previous study, the systematic search identified 40 algorithms performing variant clustering. However, 29 out of the 40 tools had to be excluded from the detailed evaluation as they had a different

scope, were unavailable, outdated, defective, or had missing documentation, which caused interpretation of the results to be impossible.

Aiming at identifying the reasons for the previously observed poor performance of the tools on real data, we reconsidered the remaining 11 tools cloneHD [22], clonosGP [23], DeCiFer [24], PyClone [25], PyClone-VI [26], QuantumClone [27], sciClone [28], Canopy [29], Cloe [30], LICHeE [31], and SPRUCE [32]. However, three tools had to be excluded from the detailed evaluation: cloneHD (missing clustering information in the output), LICHeE, and SPRUCE (both: high level of missing data). Accounting for recently published approaches performing variant clustering, the tool PICTograph [33] was additionally considered.

Altogether, a detailed evaluation of 9 tools performing variant clustering at different scenarios was executed (see Appendix A.2 for the tools for variant clustering and detailed information on all tools).

2.4. Tools for Clonal Evolution Tree Reconstruction

Similar to the systematic evaluation of tools for variant clustering, we reconsidered the previously evaluated tools for clonal evolution tree reconstruction. In our study, evaluating real data, we identified 30 algorithms performing this task. However, the majority of the 22 tools had to be excluded due to having a different scopes, being unavailable, outdated, defective, or lacking documentation. Furthermore, 4 algorithms (Canopy, Cloe, LICHeE, and SPRUCE) were excluded from detailed evaluation as they must perform tree reconstruction on their own (partly erroneous) variant clustering results. The remaining tools taken into account were ClonEvol [34], ClonalTREE [35], SCHISM [36], and TrAP [37].

Two additional recently published tools were identified: ClonalTREE2 [38] and SubMARine [39]. However, both tools were not considered for detailed evaluation (ClonalTREE2: additionally performing variant clustering; SubMARine: not reporting explicit clonal evolution trees for a majority of samples) (see Appendix A.3 for tools for clonal evolution tree reconstruction and detailed information on all tools).

2.5. Statistical Analysis

The statistical analysis was performed using R 4.1.2 [18]. To evaluate the tools' performance on variant clustering, we determined the variation of information (VI), using the R package 'clevr' [40]. The metric allows for calculating the entropy-based distance between two clusterings, with lower values corresponding to greater similarity.

To evaluate the tools' performance on clonal evolution tree reconstruction, we determined the discrete spectral distance (DSD), using the R package 'NetworkDistance' [41]. We defined the clonal evolution trees as directed graphs using the R package 'igraph' [42]. Subsequently, the graphs were transformed to adjacency matrices, for which the DSD was calculated as a similarity measure.

3. Results

3.1. Reliability of clevRsim

The simulation tool clevRsim was used to generate 88 data sets (880 patients) on clonal evolution. These sets form the basis for a systematic evaluation of tools reconstructing clonal evolution. Both variants and phylogeny are simulated at random—only restricted by user-defined input parameters and basic constraints of clonal evolution. To verify that our simulated data warrants an unbiased evaluation of the tools, the reliability of clevRsim is investigated.

With respect to variants, it can be observed that all SNVs are randomly split over the simulated clones. This observation holds for varying numbers of time points, clones, variants, and coverage (see Appendix B.1. The reliability of clevRsim, Figure A5). Exemplarily considering a varying number of clones, no biasing influence of the underlying model of evolution can be observed (Figure A6).

Considering phylogeny, a random number of branches is simulated for both the branched dependent and independent evolution (see Appendix B.1. The reliability of

clevRsim, Figure A7). No significant difference can be observed comparing the two models of evolution. Of note, the average number of simulated branches is usually lower than the maximum number of possible branches. However, this behavior of clevRsim was intended as branched evolutionary patterns were observed to commonly consist of both branched and linear development (e.g., [8,16]).

Regarding the development of CCF over time, an influence of the clone considered as well as the model of evolution can be observed, indicating the correct functioning of clevRsim (see Appendix B.1. The reliability of clevRsim, Figure A8). The first clone (stemline) is commonly characterized by the highest CCFs. For linear clonal evolution, subsequent clones follow with gradually decreasing CCFs as a child clone cannot exceed its parent. For both branched dependent and independent evolution, however, the variation in the CCFs of the subsequent clones can be observed, indicating the correct independent development of clones belonging to different branches.

3.2. Variant Clustering in the Absence of CNVs

To evaluate the basic performance of tools for variant clustering, we simulated 40 data sets (400 patients) only containing SNVs. We consider three time points, five clones, 20 variants, and a coverage of 300x as our baseline configuration. From the simulation, we analyze the influence of each of the four parameters on the variant clustering. The results are summarized in Figure 3 (see Appendix B.2. For the variant clustering in the absence of CNVs, see Table A2 for the exact numbers; information on the available clustering data is provided in Figure A9; the reported numbers of clusters for each tool and data set are provided in Figure A10).

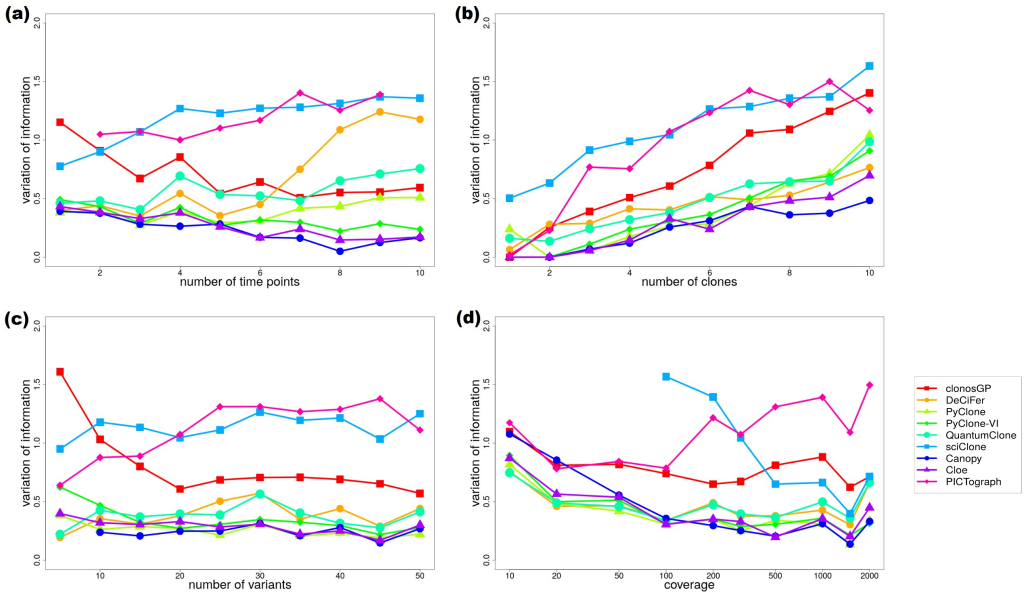

Figure 3. Variation of information comparing variant clustering performed using clonosGP, DeCiFer, PyClone, PyClone-VI, QuantumClone, sciClone, Canopy, Cloe, and PICTograph in the absence of CNVs to simulated truth. Three time points, five clones, 20 variants, and 300x coverage define the baseline configuration. We evaluate the influence of varying (**a**) number of time points; (**b**) number of clones; (**c**) number of variants; and (**d**) coverage. Of note, a missing dot indicates missing clustering results for all 10 patients per simulated data set.

When manually performing variant clustering, the results commonly improve as the number of time points increases. The more information available on the variants and their development of CCFs over time, the more clearly the clusters emerge. In Figure 3a, it can be observed that the number of available time points has only a small but diverse effect on the tools' performance. clonosGP, QuantumClone, Canopy, and Cloe show the expected

behavior of the decreasing VI as the number of time points increases. The remaining tools, on the contrary, show a minor (PyClone, PyClone-VI, and PICTograph) up to major (sciClone and DeCiFer) increase in dissimilarity compared to the ground truth. However, most tools share poor performance in the presence of only one time point.

Regarding an increasing number of clones, we expect the performance of the tools to decrease. The more clones are present, the higher the chance of clones showing similar evolution with only minor differences in CCF. All the considered tools show the expected behavior (see Figure 3b). It is interesting to observe that, even in the presence of only one clone, some tools such as sciClone do not safely succeed in performing perfect clustering but instead overestimate the number of clones (average number of clones 2; $VI_{sciClone} = 0.50$).

For the number of variants, it can be observed that their influence on most tools is mainly insignificant (see Figure 3c). Just in the case of clonosGP, an improved performance in the presence of more variants can be observed, while PICTograph shows poorer performance.

With respect to coverage, we expect to observe improved performance of the tools as coverage increases. Low coverage increases the variation in the estimated CCFs, characterizing the SNVs to cluster. As these estimated CCFs may thus show considerable differences from the true CCFs, valid clustering is challenging. As expected, all the tools show a decrease in VI as the coverage increases (see Figure 3d). The only exception from this development is PICTograph, showing the best results between 20x and 100x. Of note, the data indicate a general decline in performance at a very high coverage of 2000x.

To analyze whether the observed differences in the performance do not just result from random variation, we considered the robustness of the results at the exemplary case of the varying number of time points. The evaluation of the quartiles Q_1 and Q_3 indicates that the true differences in the performance exist and that the simulated number of $n = 10$ patients per configuration is apt to observe and assess these differences (see Appendix B.2. For the variant clustering in the absence of CNVs, see Figure A11.

3.3. Variant Clustering in the Presence of CNVs

To evaluate the performance of tools for variant clustering in an advanced, more challenging setting, we simulated 12 data sets (120 patients) containing 20 SNVs and and 6 overlapping CNVs each. Figure 4 sums up the results, considering deletions, duplications, and LOH in comparison to the baseline configuration just containing 20 SNVs (see Appendix B.3. For variant clustering in the presence of CNVs, see Table A2 for the exact numbers; information on the available clustering data is provided in Figure A12; the reported numbers of clusters for each tool and data set are provided in Figure A13).

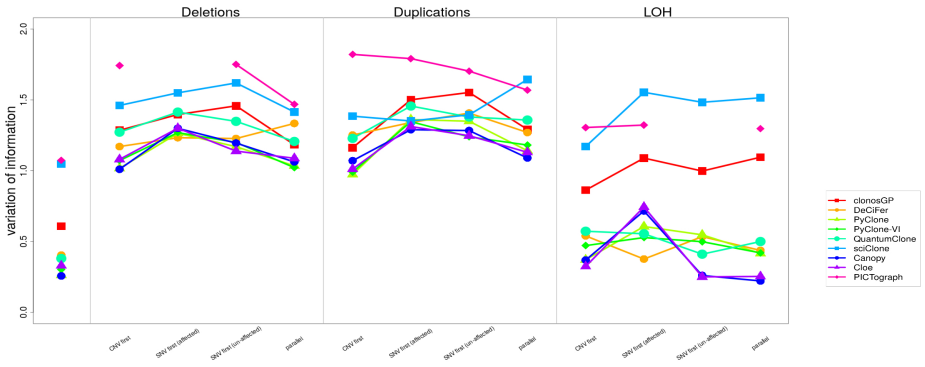

Figure 4. Variation of information comparing variant clustering performed using clonosGP, DeCiFer, PyClone, PyClone-VI, QuantumClone, sciClone, Canopy, Cloe, and PICTograph in the presence of CNVs to simulated truth. We evaluate the influence of deletions, duplications, and LOH, considering four scenarios of overlap: CNV first, SNV first (affected), SNV first (un-affected), and parallel. For comparison, the performance of each tool at the baseline configuration is included. Of note, a missing dot indicates missing clustering results for all 10 patients per simulated data set.

We aimed at simulating the data as close as possible to real data. We therefore assumed that the SNVs and CNVs were detected by performing different experiments, e.g., WES and SNP arrays. Consequently, we only know about the presence of (overlapping) variants, but not necessarily about the underlying scenario of the overlap. If a deletion and an SNV are detected in one clone, it is unclear whether the underlying genotype is B (CNV first or SNV first un-affected), A (SNV first affected), or AB and A (parallel; compare Appendix A, Table A1). For all tools, therefore, we just defined the copy number as '1' instead of '2' for the SNVs overlapping deletions. Duplications and LOH were handled accordingly.

It can be observed that, in the presence of deletions and duplications, the performance of all tools is considerably worse compared to our baseline configuration. The differences between the scenarios of overlap are usually small. However, the data indicate the lowest performance for scenarios that are SNV first (affected and un-affected) for most tools.

Considering the influence of LOH, different results can be observed. In general, the tools' performances are much better and even comparable to baseline. However, as the copy number is 2 and the genotype is AB for two out of four scenarios, these results are mainly expected.

3.4. Clonal Evolution Tree Reconstruction

The performance of tools for clonal evolution tree reconstruction is evaluated by 56 data sets (560 patients), considering varying numbers of time points and clones. For each, we simulated the underlying linear, branched dependent, and branched independent clonal evolution. The results are summed up in Figure 5 (see Appendix B.4. For the clonal evolution tree reconstruction, see Table A3 for the exact numbers; information on the available clustering data is provided in Figure A14; the reported numbers of clusters for each tool and data set are provided in Figure A15).

Regarding the influence of the available time points, a slight tendency toward better performance can be observed as the number of time points increases (Figure 5a,c,e). For ClonEvol and TrAP, the data indicate a decline in the performance as more challenging clonal evolution patterns are simulated (linear $>$ branched dependent $>$ branched independent). In the context of branched independent evolution, ClonEvol only succeeds in reconstructing clonal evolution trees for 2 out of 10 settings (2 and 3 time points; analysis of 1 time point generally not supported), while SCHISM shows a particularly poor performance, independent of the number of time points. ClonalTREE is the only tool showing comparable performance across all underlying models of clonal evolution.

With respect to a varying number of clones, the behavior of the tools for tree reconstruction is comparable to the tools for variant clustering (Figure 5b,d,f). An increasing number of clones leads to an increasing DSD and, thus, to the lower performances of the tools. Comparing linear vs. branched dependent evolution, the performances of the tools are usually similar. Considerable differences can, however, be observed for branched independent evolution, while SCHISM shows especially poor performance in the presence of $\geq$6 clones, ClonEvol and TrAP do not succeed in generating any trees for seven or more clones. ClonalTREE is the only tool showing a relatively stable performance independent of the underlying model of evolution.

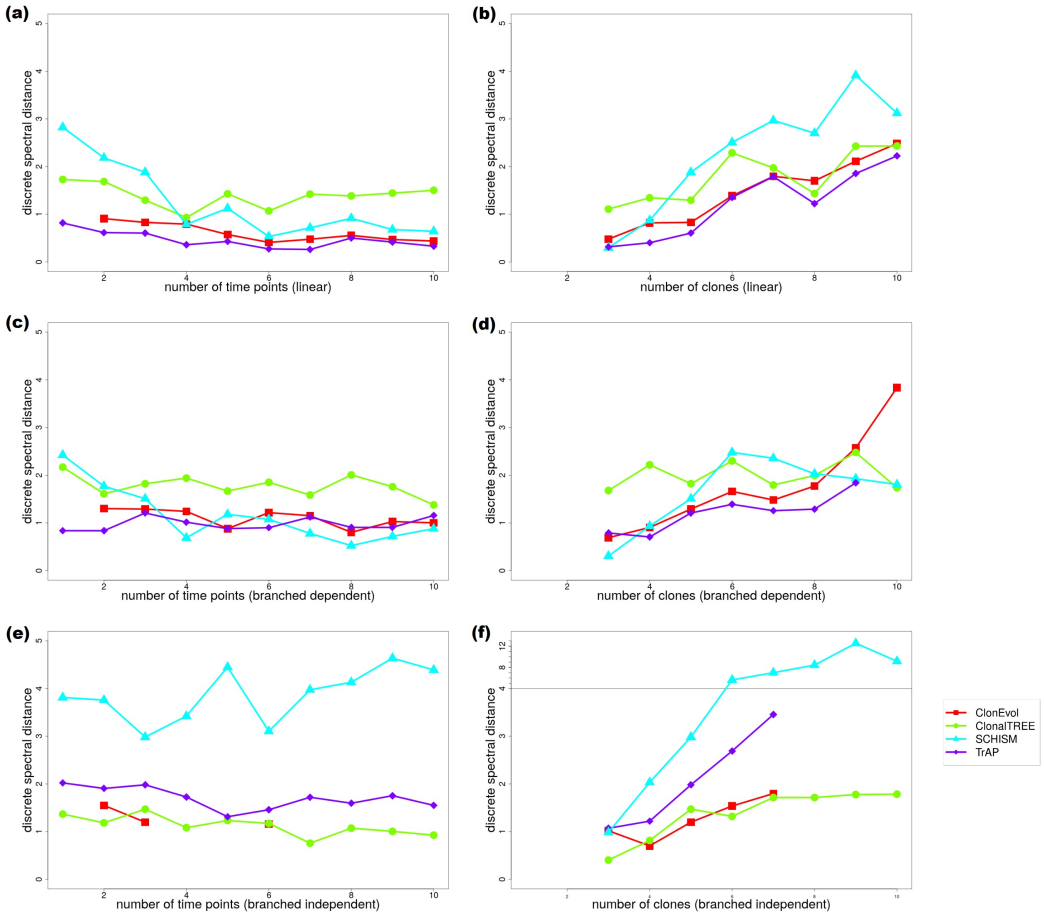

Figure 5. Discrete spectral distance comparing the trees reconstructed usinh ClonEvol, ClonalTREE, SCHISM, and TrAP to simulated truth. If tools report more than one tree, the average distance over all reported trees is calculated. We evaluate the influence of varying (**a**) number of time points at linear evolution; (**b**) number of clones at linear evolution; (**c**) number of time points at branched dependent evolution; (**d**) number of clones at branched dependent evolution; (**e**) number of time points at branched independent evolution; and (**f**) number of clones at branched independent evolution. Of note, branched dependent and independent evolution can only be considered in the presence of $\geq$3 clones. A missing dot indicates missing clustering results for all 10 patients per simulated data set.

4. Discussion

In this work, we performed a systematic evaluation of tools for variant clustering and clonal evolution tree reconstruction. To our knowledge, a comparable thorough analysis has not been performed yet. With the help of simulated data sets generated with our novel approach clevRsim, we analyzed the influence of a varying number of time points, clones, SNVs, coverage, CNVs, and the underlying model of evolution.

Our results indicate that a high number of clones poses a major challenge for all variant clustering tools. This observation is in line with our previous results on real dat: the correct clustering could not be automatically determined for any patient characterized by >3 clones [17]. Regarding clinical practice, this observation implies a considerable challenge. Several studies reported on the high level of intra-tumor heterogeneity to be observed in many cancers, e.g., [10]. However, heterogeneity is expected to hamper the valid automatic variant clustering, leading to an underestimated number of clusters.

For a varying number of time points, diverse tool-dependent effects could be observed. A majority of tools showed poor performance in the presence of only one time point. With respect to clinical practice, this observation may be similarly challenging. At the beginning of a disease, only limited amount of data are available. However, in order to use clonal

evolution analysis to support treatment decisions, valid results—even in the presence of only one time point—are needed.

A clear negative influence on the tools' performance could be observed for low coverage. This challenge may be overcome by increasingly performing deep targeted sequencing or—in case information on commonly mutated hotspot genes is lacking—high-coverage WES. However, it was also observed that a very high coverage of 2000x again leads to performance degradation. The detailed analysis of the results showed that variants of one cluster are often split among multiple clusters for all tools. It appears likely that the reason for this observation is the "over-interpretation" of small differences in the estimated CCFs. However, unexpectedly, we could also identify some clusters, partly showing differences of >20% CCF, being merged. This observation remains unclear as higher coverage is expected to lead to higher confidence in the estimated CCFs.

A particularly negative impact on variant clustering could be observed for deletions or duplications overlapping SNVs. For all tools, dissimilarity between the proposed and the true clustering was considerably increased. Several issues underlie this observation: First, a majority of the tools just allow for defining the underlying copy number for every SNV and not the fraction of cells characterized by a certain copy number. Second, the scenario of overlap is not considered despite having a major influence on the genotype and requiring an adjusted calculation of the CCF (formula provided in [14]). Third, none of the considered tools are capable of jointly clustering CNVs and SNVs in the case of lacking overlap.

Reconstructing clonal evolution trees on the basis of correctly clustered variants is not free from flaws either. The presence of a high number of clones as well as branched independent evolution increased the dissimilarity between the correct and the reported trees. When introducing their tool, SCHISM, [36] observed similar results: in the presence of many clones, SCHISM only succeeded in reporting the correct tree under ideal conditions—many time points, high coverage, and high purity of the samples. However, it should be mentioned that [36] just determined whether the correct tree was upon all trees reported by SCHISM. For 55% of all the simulated patients in our study, the tool reported an ambiguous consensus tree.

Furthermore, all the results we observed on tree reconstruction in our study were generated in an optimized setting. Commonly, the output of variant clustering tools is provided as input, which—as our results showed—does not necessarily match the correct clustering. Thus, moving from a list of variants to properly reconstructed clonal evolution, errors are expected to multiply.

To provide a means for developing new optimized algorithms, our simulation tool clevRsim is expected to provide a valuable input. Randomly simulating phylogeny and variants, our approach is able to generate an unlimited number of valid clonal evolution patterns. As future work, we plan to extend clevRsim considering a user-defined input on clinical parameters, which is expected to modulate the birth and death rates of simulated cell populations. For example, a therapy being applied is likely leading to a general decrease in CCFs and an eradication of some cell populations, while relapse is probably preceded by several new (branching) clones. Furthermore, the simulation of driver vs. passenger mutations and their effect on CCF development will be explored. As a result, the future updated version of clevRsim will allow for generating more specific clonal evolution patterns automatically—without the need for manual fine-tuning that exists at present.

It can be discussed why we focused our evaluation of bioinformatics approaches for reconstructing clonal evolution on the—compared to scDNA-seq—relatively old technique of bulk sequencing. Despite bearing the potential to revolutionize the field of clonal evolution, scDNA-seq is currently still characterized by major technical and practical challenges. Allelic drop-outs and amplification bias cause low sensitivity for detecting variants. Sequencing costs are high, which hampers its use in large study cohorts or even clinical routine. Additionally, missing suitable material prohibits the retrospective analysis

of samples. As a consequence, we expect that the reconstruction of clonal evolution on the basis of bulk sequencing data will continue to play a major role within the next decade.

5. Conclusions

With respect to optimizing cancer treatment by the detailed study of the underlying clonal evolution, our study shows that the current algorithms for fully automatic reconstruction are still characterized by major flaws. While the coverage of the sequencing experiment can be actively influenced, the tools' inability to perform correct clustering and tree reconstruction in the presence of many clones and few time points is likely to lead to false results. The same is true for additionally present, possibly overlapping CNVs. There is a pressing need for better algorithms, automatically determining the underlying scenario of overlap as well as jointly clustering CNVs and SNVs with and without overlap. Only when these current limitations are overcome and clonal evolution does no longer have to be reconstructed by tedious manual work, its full potential may be explored.

Author Contributions: S.S. conceptualized the project, generated simulated data sets, performed data analyses, and wrote the manuscript. S.R. and S.S. implemented the algorithm. J.V. and X.J. supervised the project and reviewed the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All data necessary to reproduce the reported results are available with the article. The approach clevRsim, used to simulate the data sets analyzed, is freely available at https://github.com/sandmanns/clevRsim, accessed on 4 March 2023.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ALL	Acute lymphoblastic leukemia
AML	Acute myeloid leukemia
BL	Burkitt lymphoma
CCF	Cancer cell fraction
CNV	Copy number variant
DSD	Discrete spectral distance
FISH	Fluorescence in situ hybridization
IPSS-R	Revised International Prognostic Scoring System
LOH	Loss of heterozygosity
MDS	Myeoldysplastic syndromes
scDNA-seq	Single-cell DNA sequencing
SNV	Single-nucleotide variant
VI	Variation of information
WES	Whole-exome sequencing

Appendix A. Additional Materials and Methods

Appendix A.1. clevRsim

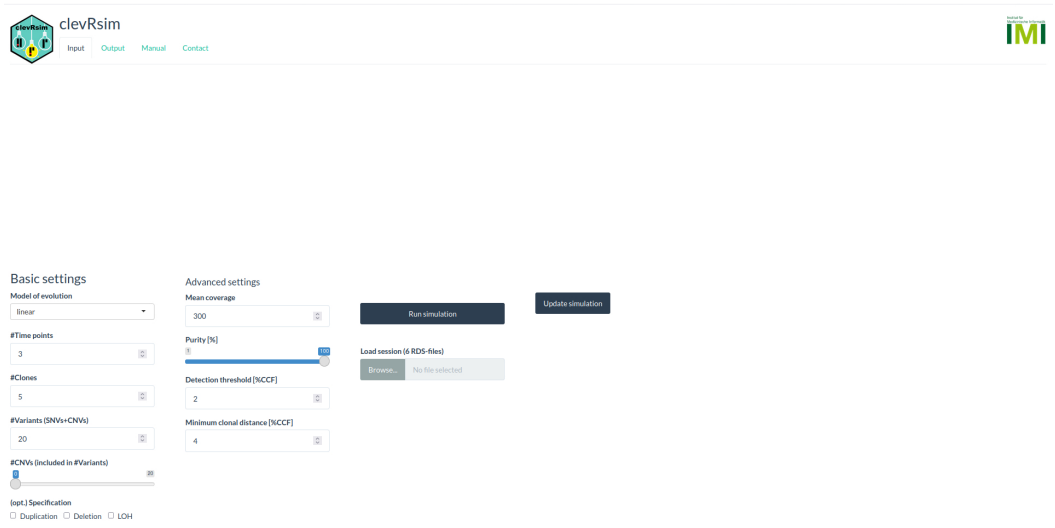

Figure A1. Defining input for the analysis using clevRsim. Basic and advanced settings can be configured for the simulation. Alternatively, a previously exported simulation can be loaded using RDS files.

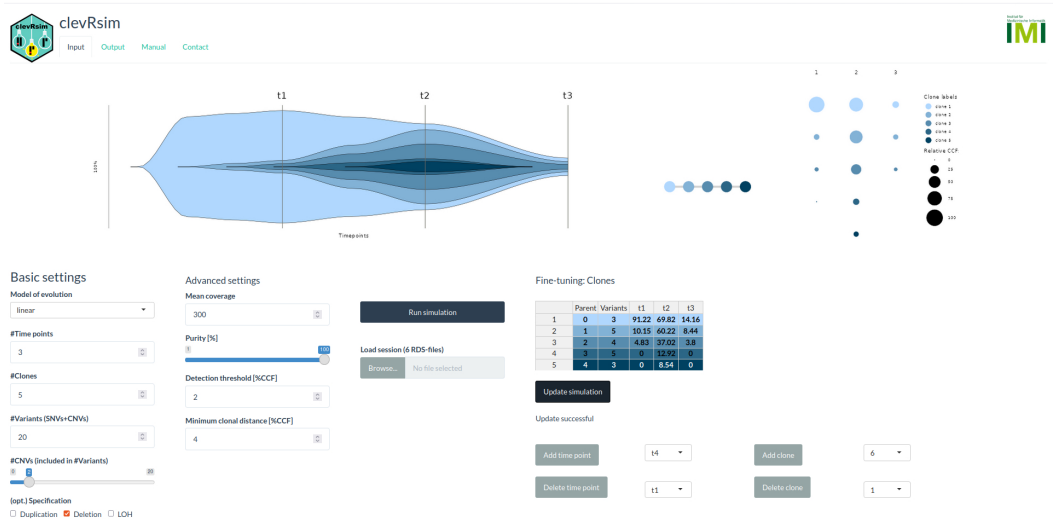

Figure A2. Exemplary simulation generated using clevRsim. When executing a simulation, the clonal evolution is visualized in the input panel using the R package 'clevRvis'. Additionally, an option for fine-tuning clones is displayed. Every cell in the table, showing a clone, its parent, the number of associated variants, and the CCFs at every time point, can be changed. Additionally, clones and time points can be added and/or removed. Subsequently, clonal evolution is updated by clicking 'Update simulation'.

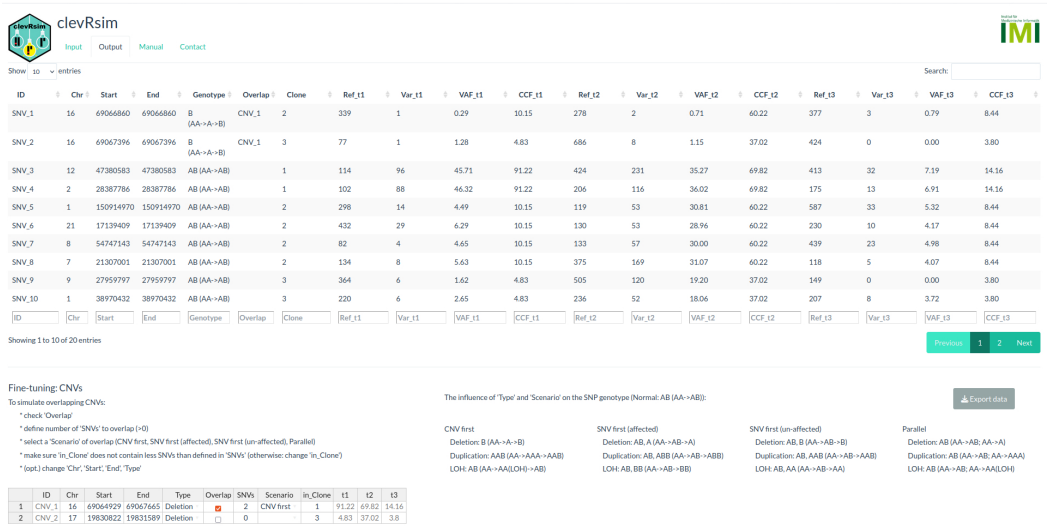

Figure A3. Exemplary simulation generated using clevRsim. When executing a simulation, the variant calls are available in the output panel. Information on basic and advanced call characteristics is provided in addition to the number of reference reads, variant reads, VAF, and CCF for every simulated time points. Additionally, an option for fine-tuning CNVs is displayed. Call characteristics can be changed, CNVs added and/or deleted, and the overlap of CNVs and SNVs defined. Subsequently, clonal evolution is updated by clicking 'Update simulation'.

Figure A4. Manual with detailed information on the use of clevRsim.

Table A1. Influence of overlapping CNVs on SNVs, dependent on the underlying scenario of overlap. For CNV first, a genomic location is first affected by a copy number variant. Subsequently, the location is affected by a point mutation (allele A changed to B). For SNV first, a genomic location is first affected by a point mutation. For 'affected', the mutated allele is affected by the subsequently acquired copy number variant, e.g., the mutated allele being deleted. For 'un-affected', the mutated allele is not affected by the subsequently acquired copy number variant, e.g., the healthy allele being deleted. For parallel, SNV and CNV are assumed to affect different cells.

Scenario	CNV Type	Genotype						
CNV first	Deletion	AA	→	A	→	B		
	Duplication	AA	→	AAA	→	AAB		
	LOH	AA	→	AA	→	AB		
SNV first (affected)	Deletion	AA	→	AB	→	A		
	Duplication	AA	→	AB	→	ABB		
	LOH	AA	→	AB	→	BB		
SNV first (un-affected)	Deletion	AA	→	AB	→	B		
	Duplication	AA	→	AB	→	AAB		
	LOH	AA	→	AB	→	AA		
Parallel	Deletion	AA	→	AB		AA	→	A
	Duplication	AA	→	AB		AA	→	AAA
	LOH	AA	→	AB		AA	→	AA

Appendix A.2. Tools for Variant Clustering

The tools for variant clustering, such as cloneHD, clonosGP, DeCiFer, PyClone, PyClone-VI, QuantumClone, sciClone, Canopy, Cloe, LICHeE, and SPRUCE, were executed as previously described in the Supplementary Information of 'Exploring Current Challenges and Perspectives for Automatic Reconstruction of Clonal Evolution' [17]. However, the cloneHD, LICHeE, and SPRUCE tools had to be excluded from detailed evaluation.

Although cloneHD could be successfully executed on the data simulated using clevR-sim, the reported output only contained information on the number of clusters, not on the assignment of a variant to a cluster. As this missing information hampers a detailed statistical evaluation of cloneHD's performance, the tool was excluded from any subsequent analysis.

For data sets from sim01 to sim40, LICHeE generated results for only 57% of the simulated patients. Furthermore, only 25% (from sim21 to sim30) to 35% (from sim11 to sim20) of the simulated SNVs per sample were assigned to a cluster. The remaining variants were filtered as 'germline' calls. No option to change the filtration settings could be identified.

For SPRUCE, we similarly observed a high level of missing data (the results are only available for data set sim21 and for four out of ten patients in sim22). In the presence of >10 variants, the tool appears to require considerable resources. The calculations did not finish within 10 min. An analysis with unlimited time was automatically terminated as it exceeded our server's resources (48 cores, 250 GB RAM, and 22 TB storage).

Additionally, the performance of the tool PICTograph was evaluated. Considering patient 1 in data set sim01, the tool was executed using the following command:

```
category<-"Timepoints"
values<-c(1:10)
for(i in values){
  message("i=",i)
  for(j in 1:10){
    message("j=",j)
    y<-read.table(paste0(dir,category,"/",i,"/Patient",
                         j,"/y.txt"),
                  header=T,sep="\t")
```

```
    y<-as.matrix(y)
    n<-read.table(paste0(dir,category,"/",i,"/Patient",
                         j,"/n.txt"),
                  header=T,sep="\t")
    n<-as.matrix(n)
    purity<-rep(1,length(y[1,]))
    tcn<-m<-y
    for(k in 1:length(y[,1])){
      for(l in 1:length(y[1,])){
        tcn[k,l]<-2
        m[k,l]<-1
      }
    }
    input.data<-list(y=y,
                     n=n,
                     purity=purity,
                     tcn=tcn,
                     m=m,
                     I=length(y[,1]),
                     S=length(y[1,]))
    clustering = tryCatch({
      all.set.results <- clusterSep(input.data,
                                    n.iter = 1000,
                                    n.burn = 100,
                                    thin = 1,
                                    max_K = 10)
      set.k.choices <- writeSetKTable(all.set.results)
      best.set.chains <- collectBestKChains(all.set.results,
         chosen_K = set.k.choices$chosen_K[!is.na(
                                set.k.choices$chosen_K)])
      chains <- mergeSetChains(best.set.chains, input.data)
      cluster.table<-writeClusterCCFsTable(chains$w_chain)
      cluster.mutations<-writeClusterAssignmentsTable(
                                    chains$z_chain)
    }, error = function(e) {
      message("fail")
      NULL
    })
  }
}
```

Appendix A.3. Tools for Clonal Evolution Tree Reconstruction

The tools for clonal evolution tree reconstruction, such as ClonEvol, ClonalTREE, SCHISM, and TrAP, were executed as previously described in the Supplementary Information of 'Exploring Current Challenges and Perspectives for Automatic Reconstruction of Clonal Evolution' [17]. The tools Canopy, Cloe, LICHeE, and SPRUCE perform both variant clustering and subsequent tree reconstruction. However, the trees are always reconstructed on the (partly erroneous) tools' clustering results. It is not possible to interfere and provide correct clustering results instead. In order to perform a fair comparison, executing all tools on the same correct clustering input, the four tools were excluded from detailed evaluation.

The recently developed tool ClonalTREE2 had to be excluded from detailed evaluation as it—different from its included predecessor ClonalTREE—is additionally performing variant clustering. No option to provide the correct clustering as input could be identified.

Additionally, the recently developed tool SubMARine was excluded from detailed evaluation. The tool could be successfully executed on all simulated data sets. However, the results on reconstructed clonal evolution trees, did not contain information on the explicit trees in from 77% (from sim11 to sim20) to 98% (from sim53 to sim62) of the simulated patients. Instead, the relations matrices were provided with several relations being marked as 'possible' (value '−1' in the file sample_basic.zmatrix). SubMARine was executed using the following command:

```
python3 ~/Downloads/submarine/submarine.py --basic_version \
        --freq_file $dir/Timepoints/$sample/Sample.csv \
        --output_prefix $dir/Timepoints/$sample/sample_basic
```

Appendix B. Additional Results

Appendix B.1. Reliability of clevRsim

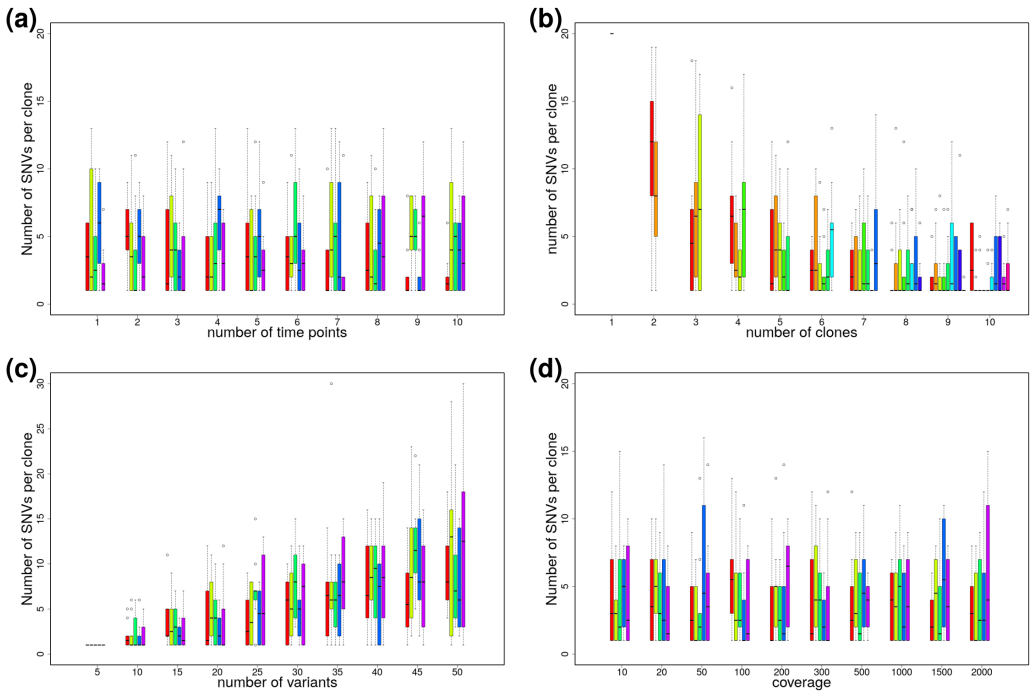

Figure A5. Boxplots showing the number of SNVs per cluster considering different numbers of (**a**) time points, (**b**) clones, (**c**) variants, and (**d**) coverage.

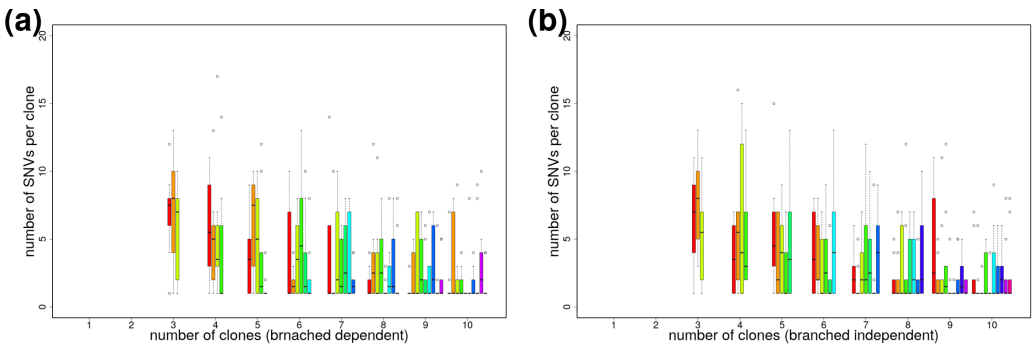

Figure A6. Boxplots showing the number of SNVs per cluster considering different numbers of clones. (**a**) Branched dependent evolution and (**b**) branched independent evolution.

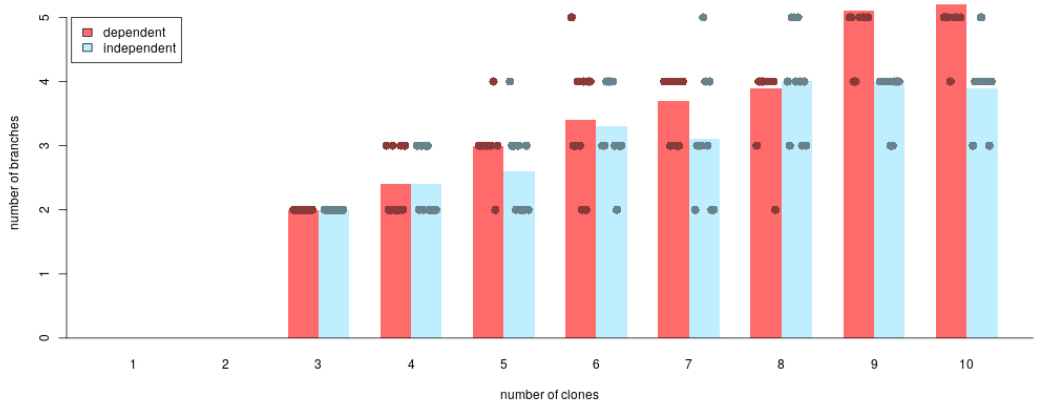

Figure A7. Average number of branches considering different numbers of clones; red: dependent branches at simulated branched dependent evolution, and blue: independent branches at simulated branched independent evolution. Red and blue dots indicate the precise number of branches, respectively.

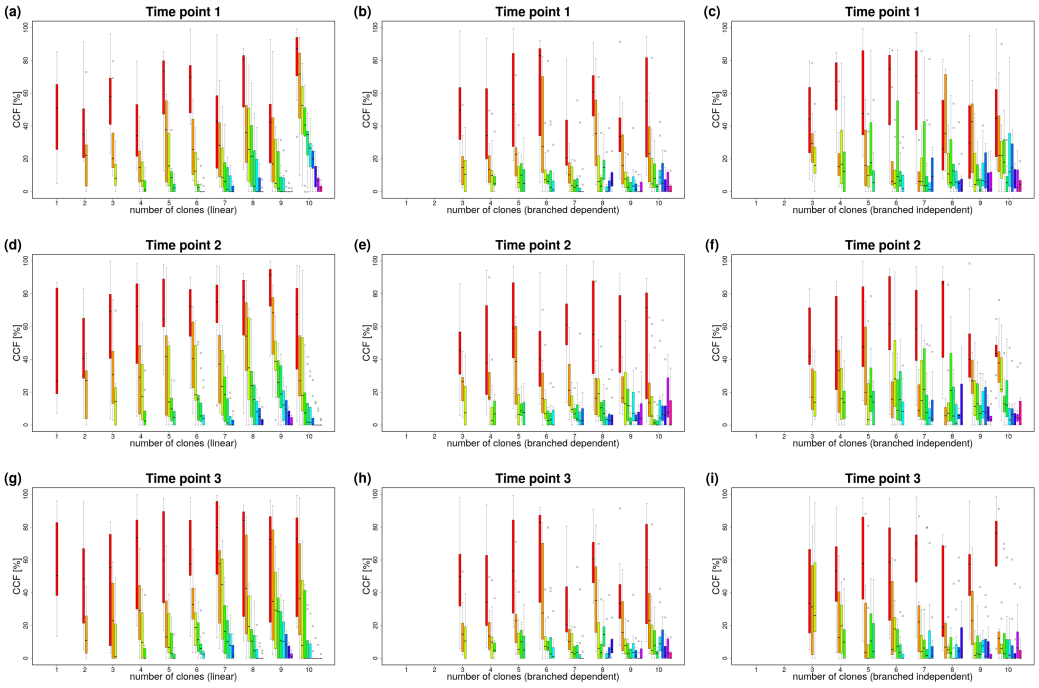

Figure A8. Boxplots showing the average CCF per cluster considering different numbers of clones at (**a**) time point 1, linear evolution, (**b**) time point 1, branched dependent evolution, (**c**) time point 1, branched independent evolution, (**d**) time point 2, linear evolution, (**e**) time point 2, branched dependent evolution, (**f**) time point 2, branched independent evolution, (**g**) time point 3, linear evolution, (**h**) time point 3, branched dependent evolution, and (**i**) time point 3, branched independent evolution.

Appendix B.2. Variant Clustering in the Absence of CNVs

Table A2. Variation of information comparing variant clustering performed using clonosGP, DeCiFer, PyClone, PyClone-VI, QuantumClone, sciClone, Canopy, Cloe, and PICTograph in the absence of CNVs to simulate truth. Different numbers of time points, clones, variants, and coverage were simulated.

Time Points									
Values	**clonosGP**	**DeCiFer**	**PyClone**	**PyClone-VI**	**QuantumClone**	**sciClone**	**Canopy**	**Cloe**	**PICTograph**
1	1.15	0.40	0.38	0.49	0.46	0.78	0.39	0.43	NA
2	0.91	0.44	0.40	0.43	0.48	0.90	0.38	0.38	1.05
3	0.67	0.35	0.27	0.29	0.41	1.07	0.28	0.33	1.07
4	0.86	0.54	0.39	0.42	0.69	1.27	0.26	0.38	1.00
5	0.54	0.35	0.29	0.27	0.54	1.23	0.28	0.26	1.10
6	0.64	0.45	0.31	0.32	0.52	1.27	0.17	0.17	1.17
7	0.51	0.75	0.42	0.30	0.48	1.28	0.16	0.24	1.40
8	0.55	1.09	0.43	0.22	0.65	1.31	0.05	0.15	1.26
9	0.56	1.24	0.51	0.29	0.71	1.37	0.13	0.15	1.39
10	0.59	1.18	0.51	0.24	0.76	1.36	0.17	0.17	NA
Clones									
Values	**clonosGP**	**DeCiFer**	**PyClone**	**PyClone-VI**	**QuantumClone**	**sciClone**	**Canopy**	**Cloe**	**PICTograph**
1	0.00	0.06	0.24	0.00	0.16	0.50	NA	0.00	0.02
2	0.26	0.28	0.00	0.00	0.14	0.63	0.00	0.00	0.23
3	0.39	0.29	0.06	0.11	0.24	0.92	0.07	0.06	0.77
4	0.51	0.41	0.18	0.24	0.32	0.99	0.12	0.15	0.76
5	0.61	0.40	0.26	0.30	0.38	1.05	0.26	0.33	1.07
6	0.78	0.52	0.28	0.36	0.51	1.27	0.31	0.24	1.23
7	1.06	0.49	0.44	0.51	0.63	1.29	0.43	0.43	1.43
8	1.09	0.53	0.63	0.65	0.64	1.36	0.36	0.48	1.30
9	1.25	0.64	0.71	0.69	0.65	1.37	0.38	0.51	1.50
10	1.40	0.77	1.04	0.91	0.99	1.63	0.48	0.70	1.25
Variants									
Values	**clonosGP**	**DeCiFer**	**PyClone**	**PyClone-VI**	**QuantumClone**	**sciClone**	**Canopy**	**Cloe**	**PICTograph**
5	1.61	0.19	0.38	0.62	0.22	0.95	NA	0.40	0.64
10	1.03	0.35	0.26	0.47	0.43	1.18	0.24	0.32	0.88
15	0.80	0.31	0.29	0.32	0.37	1.13	0.21	0.30	0.89
20	0.61	0.38	0.27	0.27	0.40	1.05	0.25	0.33	1.07
25	0.69	0.50	0.21	0.31	0.39	1.11	0.25	0.28	1.31
30	0.71	0.57	0.31	0.35	0.56	1.27	0.32	0.31	1.31
35	0.71	0.35	0.20	0.32	0.40	1.20	0.21	0.22	1.27
40	0.69	0.44	0.23	0.29	0.32	1.21	0.28	0.25	1.29
45	0.65	0.29	0.19	0.22	0.28	1.03	0.15	0.17	1.38
50	0.57	0.44	0.22	0.28	0.41	1.25	0.27	0.30	1.11
Coverage									
Values	**clonosGP**	**DeCiFer**	**PyClone**	**PyClone-VI**	**QuantumClone**	**sciClone**	**Canopy**	**Cloe**	**PICTograph**
10	1.10	0.76	0.82	0.89	0.75	NA	1.08	0.87	1.17
20	0.81	0.46	0.48	0.50	0.49	NA	0.85	0.56	0.78
50	0.82	0.47	0.42	0.51	0.46	NA	0.56	0.54	0.84
100	0.74	0.34	0.31	0.31	0.34	1.57	0.36	0.31	0.79
200	0.65	0.49	0.35	0.35	0.47	1.39	0.30	0.35	1.22
300	0.67	0.38	0.25	0.29	0.40	1.05	0.25	0.33	1.07
500	0.81	0.38	0.34	0.31	0.36	0.65	0.20	0.20	1.31
1000	0.88	0.43	0.31	0.36	0.50	0.66	0.31	0.36	1.39
1500	0.62	0.30	0.13	0.21	0.36	0.40	0.14	0.20	1.09
2000	0.71	0.66	0.34	0.31	0.66	0.72	0.33	0.45	1.50

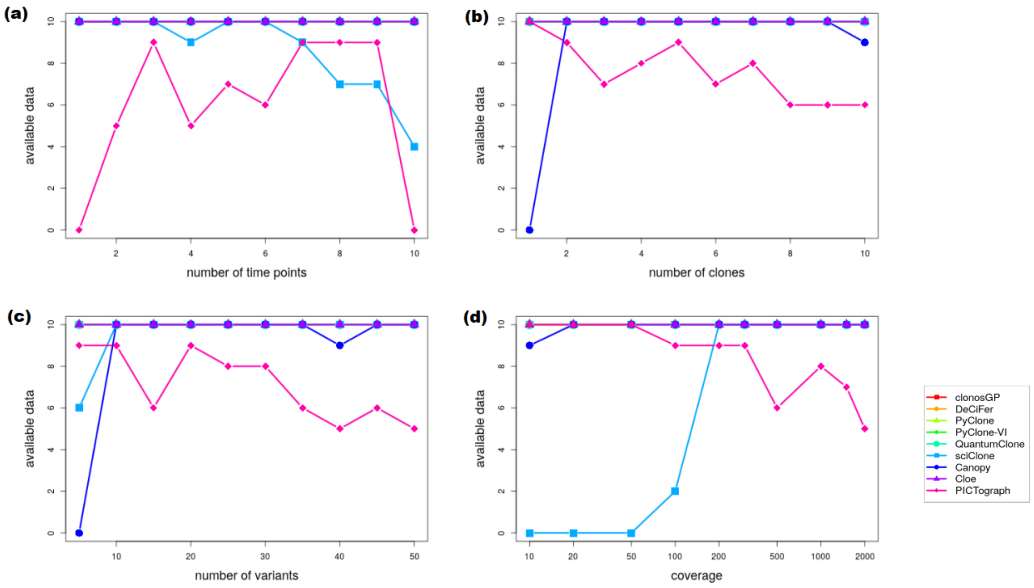

Figure A9. Available clustering output generated using clonosGP, DeCiFer, PyClone, PyClone-VI, QuantumClone, sciClone, Canopy, Cloe, and PICTograph in the absence of CNVs. For each considered number of (**a**) time points, (**b**) clones, (**c**) variants, and (**d**) coverage, 10 patients were simulated.

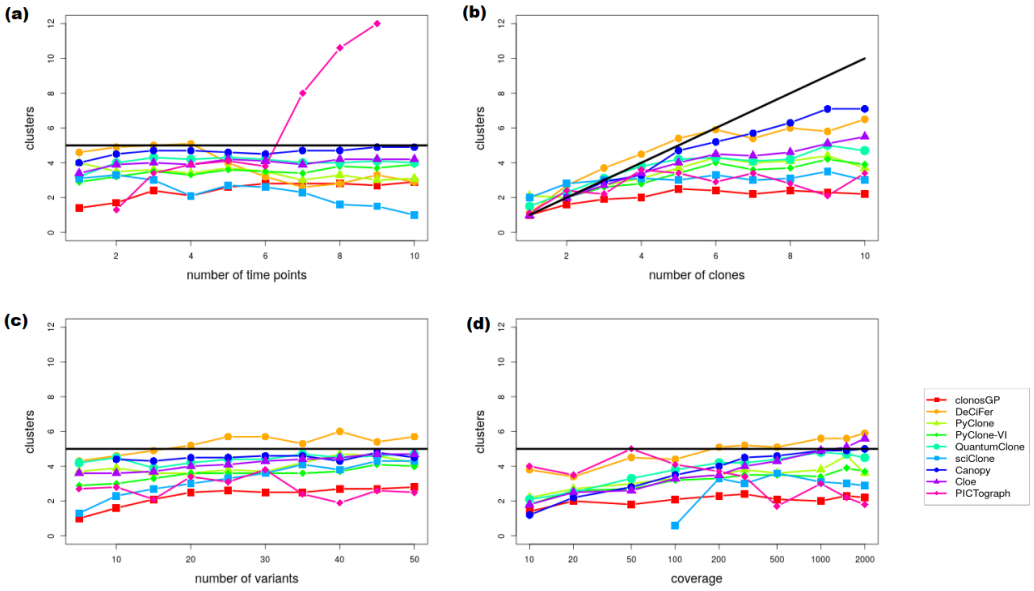

Figure A10. Number of clusters reported using clonosGP, DeCiFer, PyClone, PyClone-VI, QuantumClone, sciClone, Canopy, Cloe, and PICTograph in the absence of CNVs for varying numbers of (**a**) time points, (**b**) clones, (**c**) variants, and (**d**) coverage. The black line indicates the true number of simulated clones.

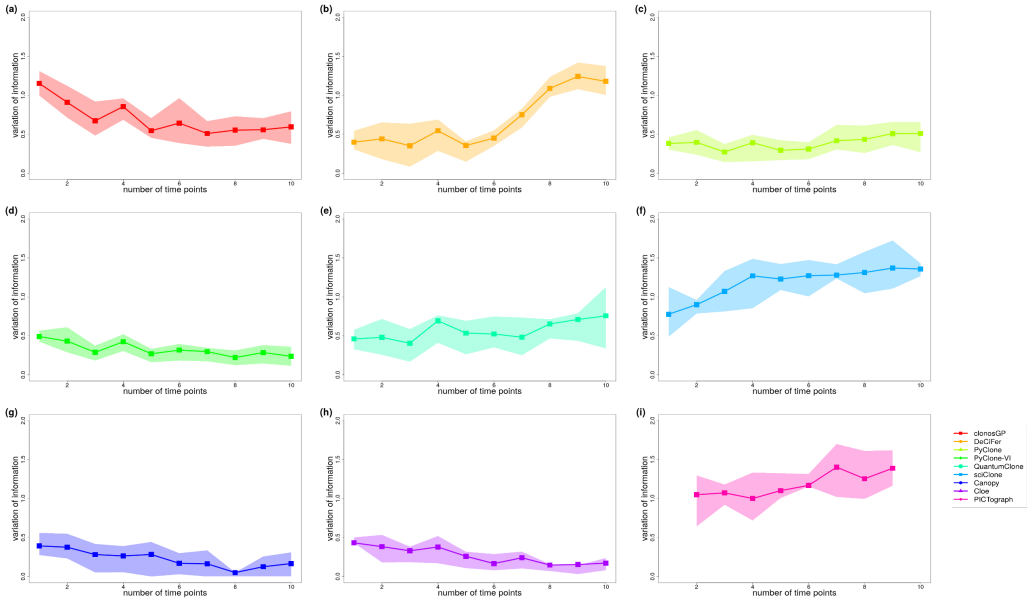

Figure A11. Number of clusters reported using (**a**) clonosGP, (**b**) DeCiFer, (**c**) PyClone, (**d**) PyClone-VI, (**e**) QuantumClone, (**f**) sciClone, (**g**) Canopy, (**h**) Cloe, and (**i**) PICTograph, in the absence of CNVs for varying numbers of time points. Light background colors indicate the Q_1 and Q_3 quartiles.

Appendix B.3. Variant Clustering in the Presence of CNVs

Table A3. Variation of information comparing variant clustering performed using clonosGP, DeCiFer, PyClone, PyClone-VI, QuantumClone, sciClone, Canopy, Cloe, and PICTograph in the presence of CNVs to simulate truth. Deletions, duplications, and LOH were simulated, considering the scenarios of overlap: CNV first, SNV first (affected), SNV first (un-affected), and parallel.

Deletions									
Scenario of Overlap	**clonosGP**	**DeCiFer**	**PyClone**	**PyClone-VI**	**QuantumClone**	**sciClone**	**Canopy**	**Cloe**	**PICTograph**
CNV first	1.29	1.17	1.02	1.07	1.27	1.46	1.01	1.08	1.74
SNV first (affected)	1.40	1.23	1.26	1.27	1.41	1.55	1.30	1.30	NA
SNV first (un-affected)	1.46	1.23	1.17	1.20	1.35	1.62	1.19	1.14	1.75
Parallel	1.18	1.33	1.03	1.02	1.21	1.41	1.06	1.09	1.47
Duplications									
Scenario of Overlap	**clonosGP**	**DeCiFer**	**PyClone**	**PyClone-VI**	**QuantumClone**	**sciClone**	**Canopy**	**Cloe**	**PICTograph**
CNV first	1.16	1.25	0.97	0.99	1.23	1.39	1.07	1.01	1.82
SNV first (affected)	1.50	1.34	1.36	1.35	1.46	1.35	1.29	1.31	1.79
SNV first (un-affected)	1.55	1.41	1.35	1.24	1.38	1.39	1.28	1.25	1.70
Parallel	1.29	1.27	1.14	1.18	1.36	1.64	1.09	1.13	1.57
LOH									
Scenario of Overlap	**clonosGP**	**DeCiFer**	**PyClone**	**PyClone-VI**	**QuantumClone**	**sciClone**	**Canopy**	**Cloe**	**PICTograph**
CNV first	0.86	0.54	0.37	0.47	0.57	1.17	0.37	0.32	1.30
SNV first (affected)	1.09	0.38	0.61	0.53	0.55	1.55	0.72	0.74	1.32
SNV first (un-affected)	1.00	0.53	0.55	0.50	0.41	1.48	0.26	0.25	NA
Parallel	1.10	0.44	0.42	0.42	0.50	1.51	0.22	0.25	1.30

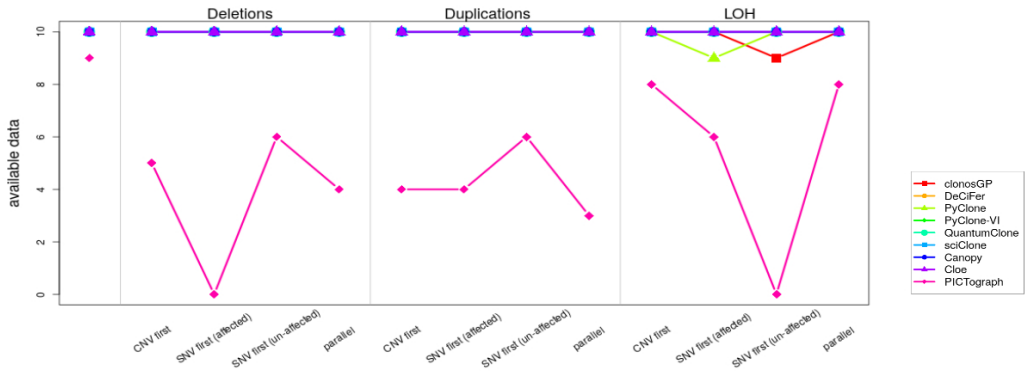

Figure A12. Available clustering output generated using clonosGP, DeCiFer, PyClone, PyClone-VI, QuantumClone, sciClone, Canopy, Cloe, and PICTograph in the presence of CNVs. For each considered number of time points, clones, variants, and coverage, 10 patients were simulated.

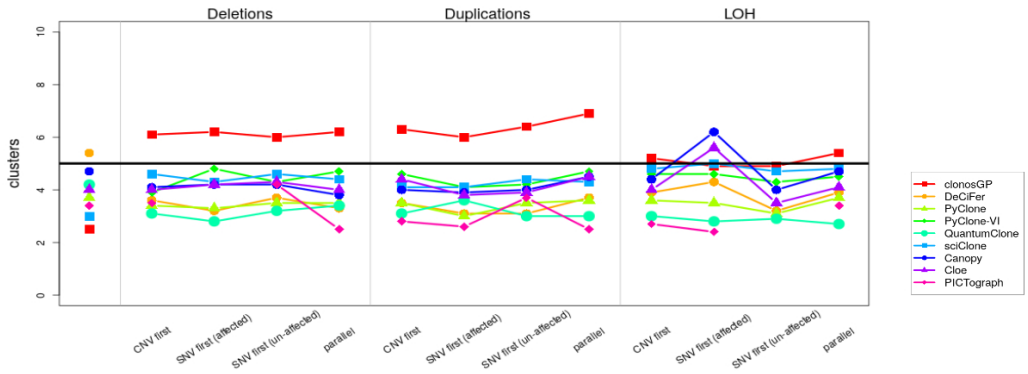

Figure A13. Number of clusters reported using clonosGP, DeCiFer, PyClone, PyClone-VI, QuantumClone, sciClone, Canopy, Cloe, and PICTograph in the presence of CNVs. The black line indicates the true number of simulated clones.

Appendix B.4. Clonal Evolution Tree Reconstruction

Table A4. Discrete spectral distance comparing the trees reconstructed using ClonEvol, ClonalTREE, SCHISM, and TrAP to simulate truth. Different numbers of time points and clones under the models of clonal evolution linear, branched dependent, and branched independent were simulated.

	Time Points (Linear)				Clones (Linear)			
Values	**ClonEvol**	**ClonalTREE**	**SCHISM**	**TrAP**	**ClonEvol**	**ClonalTREE**	**SCHISM**	**TrAP**
1	NA	1.73	2.82	0.82				
2	0.91	1.69	2.18	0.62				
3	0.83	1.3	1.88	0.61	0.48	1.11	0.3	0.32
4	0.79	0.93	0.79	0.36	0.82	1.35	0.87	0.40
5	0.57	1.43	1.12	0.43	0.83	1.3	1.88	0.61
6	0.41	1.07	0.53	0.27	1.39	2.29	2.51	1.36
7	0.48	1.42	0.71	0.26	1.8	1.97	2.97	1.79
8	0.56	1.39	0.92	0.50	1.7	1.43	2.7	1.22
9	0.47	1.44	0.68	0.42	2.11	2.43	3.91	1.86
10	0.44	1.5	0.64	0.33	2.48	2.43	3.12	2.22

Table A4. *Cont.*

Values	Time Points (Branched Dependent) ClonEvol	ClonalTREE	SCHISM	TrAP	Clones (Branched Dependent) ClonEvol	ClonalTREE	SCHISM	TrAP
1	NA	2.17	2.43	0.84				
2	1.3	1.61	1.77	0.84				
3	1.29	1.82	1.51	1.21	0.69	1.68	0.31	0.79
4	1.24	1.94	0.69	1.02	0.91	2.22	0.93	0.71
5	0.88	1.67	1.18	0.88	1.29	1.82	1.51	1.21
6	1.22	1.85	1.08	0.90	1.66	2.3	2.48	1.39
7	1.15	1.59	0.78	1.12	1.48	1.8	2.36	1.26
8	0.8	2.01	0.52	0.91	1.77	2	2.03	1.29
9	1.03	1.76	0.72	0.91	2.58	2.48	1.93	1.84
10	1	1.38	0.88	1.16	3.84	1.74	1.81	NA
Values	**Time Points (Branched Independent) ClonEvol**	**ClonalTREE**	**SCHISM**	**TrAP**	**Clones (Branched Independent) ClonEvol**	**ClonalTREE**	**SCHISM**	**TrAP**
1	NA	1.37	3.81	2.02				
2	1.55	1.18	3.76	1.91				
3	1.2	1.47	2.98	1.98	1.02	0.4	0.98	1.07
4	NA	1.08	3.42	1.73	0.7	0.81	2.03	1.22
5	NA	1.23	4.45	1.31	1.2	1.47	2.98	1.98
6	1.16	1.17	3.1	1.46	1.54	1.32	5.63	2.69
7	NA	0.76	3.97	1.72	1.8	1.71	7	3.46
8	NA	1.07	4.13	1.60	NA	1.72	8.43	NA
9	NA	1.01	4.64	1.75	NA	1.78	12.54	NA
10	NA	0.92	4.39	1.55	NA	1.78	9.15	NA

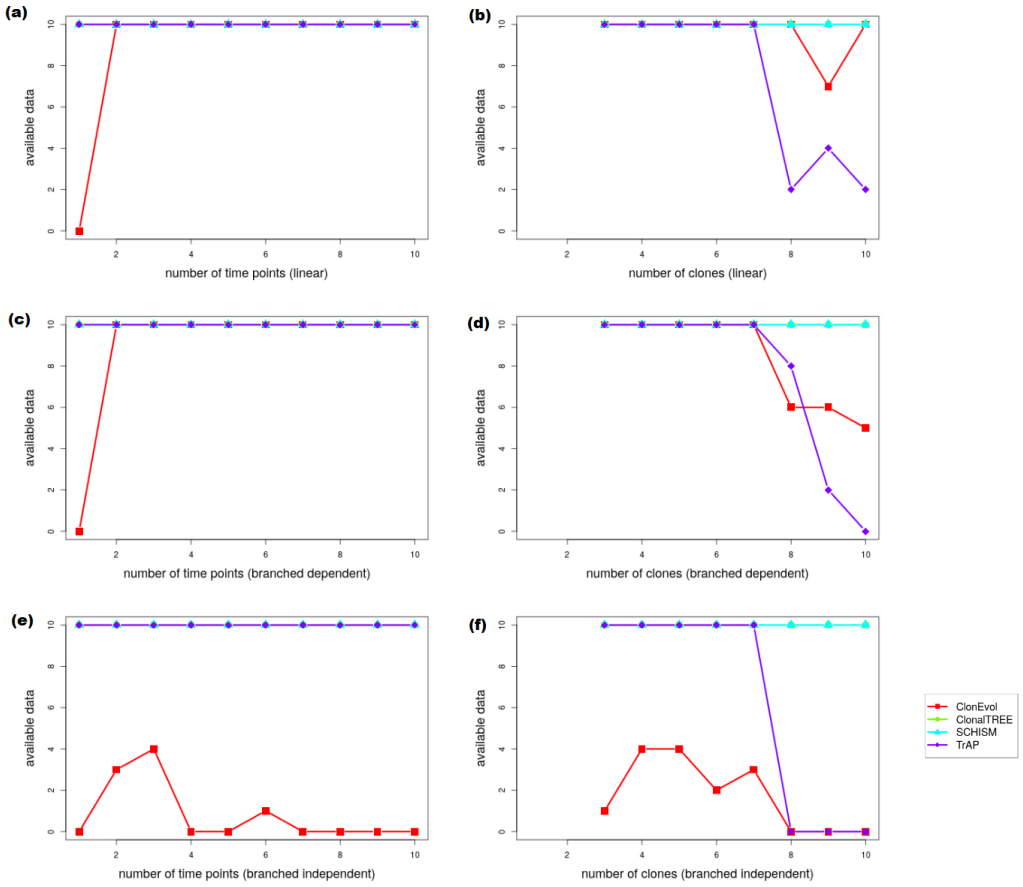

Figure A14. Available tree output generated using ClonEvol, ClonalTREE, SCHISM, and TrAP. For each considered number of time points and clones under the models of clonal evolution linear, branched dependent, and branched independent, 10 patients were simulated; (**a**) linear evolution, varying number of time points, (**b**) linear evolution, varying number of clones, (**c**) branched dependent evolution, varying number of time points, (**d**) branched dependent evolution, varying number of clones, (**e**) branched independent evolution, varying number of time points, and (**f**) branched independent evolution, varying number of clones.

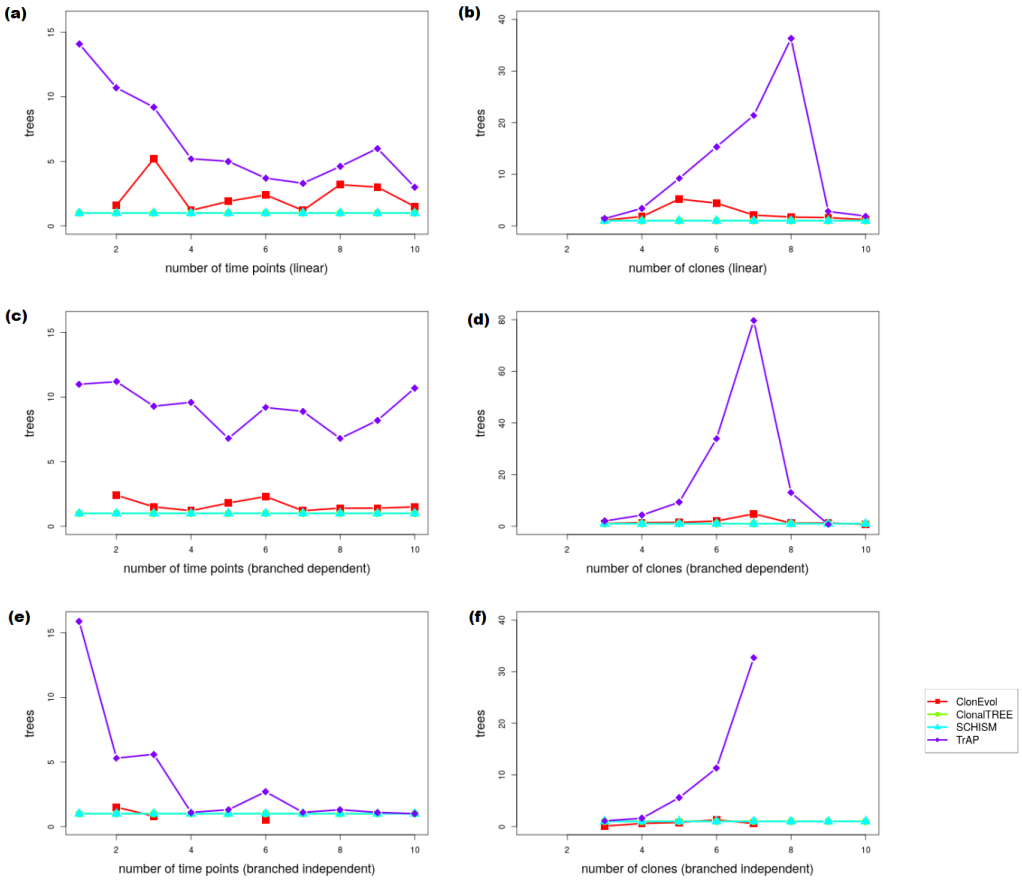

Figure A15. Number of trees reported using ClonEvol, ClonalTREE, SCHISM, and TrAP, while ClonalTREE and SCHISM always report a single tree, ClonEvol and especially TrAP usually report more than one possible tree; (**a**) linear evolution, varying number of time points, (**b**) linear evolution, varying number of clones, (**c**) branched dependent evolution, varying number of time points, (**d**) branched dependent evolution, varying number of clones, (**e**) branched independent evolution, varying number of time points, and (**f**) branched independent evolution, varying number of clones.

References

1. Global Cancer Observatory: Cancer Today. International Agency for Research on Cancer. Available online: https://gco.iarc.fr/today (accessed on 29 January 2023).
2. World Cancer Research Fund International. Cancer Survival Statistics. Available online: https://www.wcrf.org/cancer-trends/cancer-survival-statistics/ (accessed on 29 Janury 2023).
3. Alexander, T.B.; Wang, L.; Inaba, H.; Triplett, B.M.; Pounds, S.; Ribeiro, R.C.; Pui, C.H.; Rubnitz, J.E. Decreased relapsed rate and treatment-related mortality contribute to improved outcomes for pediatric acute myeloid leukemia in successive clinical trials. *Cancer* **2017**, *123*, 3791–3798. [CrossRef]
4. Rockberg, J.; Amelio, J.M.; Taylor, A.; Jörgensen, L.; Ragnhammar, P.; Hansson, J. Epidemiology of cutaneous melanoma in Sweden-Stage-specific survival and rate of recurrence. *Int. J. Cancer* **2016**, *139*, 2722–2729. [CrossRef]
5. Das, S.; Kundu, M.; Jena, B.C.; Mandal, M. Chapter 25—Causes of cancer: Physical, chemical, biological carcinogens, and viruses. In *Materials Today: Biomaterials for 3D Tumor Modeling*; Kundu, S.C., Reis, R.L., Eds.; Elsevier: Amsterdam, The Netherlands, 2020; pp. 607–641.
6. Greenberg, P.L.; Tuechler, H.; Schanz, J.; Sanz, G.; Garcia-Manero, G.; Solé, F.; Bennett, J.M.; Bowen, D.; Fenaux, P.; Dreyfus, F.; et al. Revised international prognostic scoring system for myelodysplastic syndromes. *Blood* **2012**, *120*, 2454–2465. [CrossRef]
7. Malcovati, L.; Crouch, S.; De Graaf, A.O.; Sandmann, S.; Tobiasson, M.; Kosmider, O.; van der Reijden, B.A.; Painter, D.; Van de Loosdrecht, A.A.; Symeonidis, A.; et al. Mutation Profiles Identify Distinct Clusters of Lower Risk Myelodysplastic Syndromes with Unique Clinical and Biological Features and Clinical Endpoints. *Blood* **2020**, *136*, 29. [CrossRef]
8. Reutter, K.; Sandmann, S.; Rohde, J.; Müller, S.; Wöste, M.; Khanam, T.; Michgehl, U.; Klapper, W.; Wößmann, W.; Seggewiß, J.; et al. Reconstructing clonal evolution in relapsed and non-relapsed Burkitt lymphoma. *Leukemia* **2021**, *35* 639–643. [CrossRef] [PubMed]

9. Byun, J.M.; Koh, Y.; Shin, D.Y.; Kim, I.; Yoon, S.S.; Lee, J.O.; Bang, S.M.; Kim, K.H.; Jung, S.H.; Lee, W.S.; et al. BCR-ABL translocation as a favorable prognostic factor in elderly patients with acute lymphoblastic leukemia in the era of potent tyrosine kinase inhibitors. *Haematologica* **2017**, *102*, e187–e190. [CrossRef]
10. Gerlinger, M.; Rowan, A.J.; Horswell, S.; Math, M.; Larkin, J.; Endesfelder, D.; Gronroos, E.; Martinez, P.; Matthews, N.; Stewart, A.; et al. Intratumor heterogeneity and branched evolution revealed by multiregion sequencing. *N. Engl. J. Med.* **2012**, *366*, 883–892. [CrossRef]
11. Zhang, J.; Fujimoto, J.; Zhang, J.; Wedge, D.C.; Song, X.; Zhang, J.; Seth, S.; Chow, C.W.; Cao, Y.; Gumbs, C.; et al. Intratumor heterogeneity in localized lung adenocarcinomas delineated by multiregion sequencing. *Science* **2014**, *346*, 256–259. [CrossRef] [PubMed]
12. Luquette, L.J.; Bohrson, C.L.; Sherman, M.A.; Park, P.J. Identification of somatic mutations in single cell DNA-seq using a spatial model of allelic imbalance. *Nat. Commun.* **2019**, *10*, 3908. [CrossRef] [PubMed]
13. Navin, N.E. Cancer genomics: One cell at a time. *Genome Biol.* **2014**, *15*, 452. [CrossRef]
14. Sandmann, S.; Behrens, Y.L.; Davenport, C.; Thol, F.; Heuser, M.; Dörfel, D.; Löhr, F.; Castrup, A.; Steinemann, D.; Varghese, J.; et al. Clonal Evolution at First Sight: A Combined Visualization of Diverse Diagnostic Methods Improves Understanding of Leukemic Progression. *Front. Oncol.* **2022**, *12*, 888114. [CrossRef]
15. Ding, L.; Raphael, B.J.; Chen, F.; Wendl, M.C. Advances for studying clonal evolution in cancer. *Cancer Lett.* **2013**, *340*, 212–219. [CrossRef] [PubMed]
16. Da Silva-Coelho, P.; Kroeze, L.I.; Yoshida, K.; Koorenhof-Scheele, T.N.; Knops, R.; van de Locht, L.T.; de Graaf, A.O.; Massop, M.; Sandmann, S.; Dugas, M.; et al. Clonal evolution in myelodysplastic syndromes. *Nat. Commun.* **2017**, *8*, 15099. [CrossRef] [PubMed]
17. Sandmann, S.; Richter, S.; Jiang, X.; Varghese J. Exploring Current Challenges and Perspectives for Automatic Reconstruction of Clonal Evolution. *Cancer Genom. Proteom.* **2022**, *19*, 194–204. [CrossRef] [PubMed]
18. R Core Team. *R: A Language and Environment for Statistical Computing*; R Foundation for Statistical Computing: Vienna, Austria, 2021. Available online: https://www.R-project.org/ (accessed on 29 January 2023).
19. Davies, A.; Gao, R.; Navin, N. Tumor evolution: Linear, branching, neutral or punctuated? *Biochim. Biophys. Acta Rev. Cancer* **2017**, *1867*, 151–161. [CrossRef]
20. Sandmann, S.; Inserte, C.; Varghese, J. clevRvis: Visualization Techniques for Clonal Evolution. *GigaScience* 2022, *accepted*.
21. Kimura, M.; Crow, J.F. He number of alleles that can be maintained in a finite population *Genetics* **1964**, *49*, 725–738. [CrossRef]
22. Fischer, A.; Vázquez-García, I.; Illingworth, C.J.R.; Mustonen, V. High-definition reconstruction of clonal composition in cancer. *Cell Rep.* **2014**, *7*, 1740–1752. [CrossRef]
23. Vavoulis, D.V.; Cutts, A.; Taylor, J.C.; Schuh, A. A statistical approach for tracking clonal dynamics in cancer using longitudinal next-generation sequencing data. *Bioinformatics* **2021**, *37*, 147–154. [CrossRef]
24. Satas, G.; Zaccaria, S.; El-Kebir, M.; Raphael, B. DeCiFering the elusive cancer cell fraction in tumor heterogeneity and evolution. *Cell Syst.* **2021**, *12*, 1004–1018.e10. [CrossRef] [PubMed]
25. Roth, A.; Khattra, J.; Yap, D.; Wan, A.; Laks, E.; Biele, J.; Ha, G.; Aparicio, S.; Bouchard-Côté, A.; Shah, S.P. PyClone: Statistical inference of clonal population structure in cancer. *Nat. Methods* **2014**, *11*, 396–398. [CrossRef] [PubMed]
26. Gillis, S.; Roth, A. PyClone-VI: Scalable inference of clonal population structures using whole genome data. *BMC Bioinform.* **2020**, *21*, 571. [CrossRef] [PubMed]
27. Deveau, P.; Colmet Daage, L.; Oldridge, D.; Bernard, V.; Bellini, A.; Chicard, M.; Clement, N.; Lapouble, E.; Combaret, V.; Boland, A.; et al. QuantumClone: Clonal assessment of functional mutations in cancer based on a genotype-aware method for clonal reconstruction. *Bioinformatics* **2018**, *34*, 1808–1816. [CrossRef]
28. Miller, C.A.; White, B.S.; Dees, N.D.; Griffith, M.; Welch, J.S.; Griffith, O.L.; Vij, R.; Tomasson, M.H.; Graubert, T.A.; Walter, M.J.; et al. SciClone: Inferring Clonal Architecture and Tracking the Spatial and Temporal Patterns of Tumor Evolution. *PLoS Comput Biol.* **2014**, *10*, e1003665. [CrossRef]
29. Jiang, Y.; Qiu, Y.; Minn, A.J.; Zhang, N.R. Assessing intratumor heterogeneity and tracking longitudinal and spatial clonal evolutionary history by next-generation sequencing. *Proc. Natl. Acad. Sci. USA* **2016**, *113*, E5528–E5537. [CrossRef]
30. Marass, F.; Mouliere, F.; Yuan, K.; Rosenfeld, N.; Markowetz, F. A phylogenetic latent feature model for clonal deconvolution. *Ann. Appl. Stat.* **2016**, *10*, 2377–2404. [CrossRef]
31. Popic, V.; Salari, R.; Hajirasouliha, I.; Kashef-Haghighi, D.; West, R.B.; Batzoglou, S. Fast and scalable inference of multi-sample cancer lineages. *Genome Biol.* **2015**, *16*, 91. [CrossRef]
32. El-Kebir, M.; Satas, G.; Oesper, L.; Raphael, B.J. Inferring the Mutational History of a Tumor Using Multi-state Perfect Phylogeny Mixtures. *Cell Syst.* **2016**, *3*, 43–53. [CrossRef] [PubMed]
33. Zheng, L.; Niknafs, N.; Wood, L.D.; Karchin, R.; Scharpf, R.B. Estimation of cancer cell fractions and clone trees from multi-region sequencing of tumors. *Bioinformatics* **2022**, *38*, 3677–3683. [CrossRef] [PubMed]
34. Dang, H.X.; White, B.S.; Foltz, S.M.; Miller, C.A.; Luo, J.; Fields, R.C.; Maher, C.A. ClonEvol: Clonal ordering and visualization in cancer sequencing. *Ann. Oncol.* **2017**, *28*, 3076–3082. [CrossRef] [PubMed]
35. Ismail, W.M.; Tang, H. Clonal reconstruction from time course genomic sequencing data. *BMC Genom.* **2019**, *20*, 1002. [CrossRef] [PubMed]

36. Niknafs, N.; Beleva-Guthrie, V.; Naiman, D.Q.; Karchin, R. SubClonal Hierarchy Inference from Somatic Mutations: Automatic Reconstruction of Cancer Evolutionary Trees from Multi-region Next Generation Sequencing. *PLoS Comput Biol.* **2015**, *11*, e1004416. [CrossRef]
37. Strino, F.; Parisi, F.; Micsinai, M.; Kluger, Y. TrAp: A tree approach for fingerprinting subclonal tumor composition. *Nucleic Acids Res.* **2013**, *41*, e165. [CrossRef]
38. Ismail, W.M.; Tang, H. A scalable algorithm for clonal reconstruction from sparse time course genomic sequencing data. *bioRxiv* **2021**. [CrossRef]
39. Sundermann, L.K.; Wintersinger, J.; Rätsch, G.; Stoye, J.; Morris, Q. Reconstructing tumor evolutionary histories and clone trees in polynomial-time with SubMARine. *PLoS Comput Biol.* **2021**, *17*, e1008400. [CrossRef] [PubMed]
40. Marchant, N.; Steorts, R. clevr: Clustering and Link Prediction Evaluation in R. version 0.1.1. 2020. Available online: https://CRAN.R-project.org/package=clevr (accessed on 29 January 2023).
41. You K. NetworkDistance: Distance Measures for Networks. version 0.3.4. 2021. Available online: https://CRAN.R-project.org/package=NetworkDistance (accessed on 29 January 2023).
42. Csardi, G.; Nepusz, T. The igraph software package for complex network research. *Inter J. Complex Syst.* **2006**, *1695*, 1–9.