REVIEW

Assessing the Performance of Methods for Cell Clustering from Single-cell DNA Sequencing Data

Rituparna Khan and Xian Mallory*

*Correspondence: xfan2@fsu.edu
Department of Computer Science, Florida State University, Tallahassee, Florida, USA
Full list of author information is available at the end of the article

Abstract

Background: Many cancer genomes have been known to contain more than one subclone inside one tumor, the phenomenon of which is called intra-tumor heterogeneity (ITH). Characterizing ITH is essential in designing treatment plans, prognosis as well as the study of cancer progression. Single-cell DNA sequencing (scDNAseq) has been proven effective in deciphering ITH. Cells corresponding to each subclone are supposed to carry a unique set of mutations such as single nucleotide variations (SNV). While there have been many studies on the cancer evolutionary tree reconstruction, not many have been proposed that simply characterize the subclonality without tree reconstruction. While tree reconstruction is important in the study of cancer evolutionary history, typically they are computationally expensive in terms of running time and memory consumption due to the huge search space of the tree structure. On the other hand, subclonality characterization of single cells can be converted into a cell clustering problem, the dimension of which is much smaller, and the turnaround time is much shorter. Despite the existence of a few state-of-the-art cell clustering computational tools for scDNAseq, there lacks a comprehensive and objective comparison under different settings.

Results: In this paper, we benchmarked three state-of-the-art cell clustering tools–SCG, BnpC and SCClone–on simulated datasets given a variety of parameter settings and a real dataset. We designed a simulator specifically for cell clustering, and compared the three methods' performances in terms of their clustering accuracy, genotyping accuracy and running time.

Conclusion: From the benchmark study, we conclude that BnpC's clustering accuracy is the highest of all three methods. SCG's accuracy is very close to BnpC's, but it is much faster than the other two methods especially when the cell number is within 1000. When there are a large number of single cells (> 1500), BnpC is highly recommended due to its scalability while not sacrificing the clustering accuracy. Of the three methods, SCClone has the highest accuracy in estimating the number of clusters especially when the underlying number of cluster is high. When the variance of the cluster sizes is high, all three methods' clustering accuracy drops. To improve the clustering accuracy while cluster sizes' variance is high is potentially a future work in scDNAseq cell clustering.

Keywords: Single-cell DNA sequencing; Cell clustering; Single nucleotide variation; Benchmarking; Intra-tumor heterogeneity

Background

Cancer progresses with acquired mutations [1–3]. During this process, different cancer cells inside the same tumor may acquire different set of mutations, leading to a notoriously difficult-to-solve problem, the intra-tumor heterogeneity (ITH) [4–14].

In particular, the co-existence of multiple subclones of cancer cells leads to the malignancy of cancer, and even metastasis, which is responsible for 90% of the death of cancer [15]. ITH also is a confounding factor in treatment and prognosis due to the lack of a comprehensive knowledge of the clonality in the tumor. Therefore, it is important to fully characterize the clonality in a cancer genome to guide the prescription of drug or chemotherapy, to obtain an accurate prognosis, as well as to gain a better understanding of the mechanism of cancer progression.

Single-cell DNA sequencing (scDNAseq) data, due to its sequencing each cell separately, makes it a unique type of data advantageous in understanding the clonality and ITH [16, 17]. Technical challenges of scDNAseq lie in the whole genome amplification (WGA) of the tiny amount of the DNA (6pg) in the cell by 3 to 9 orders of magnitude [18] for sequencing library construction [19]. Particularly, multiple displacement amplification (MDA)-based approach [20–22] generates scDNAseq data that have high depth of the sequencing, facilitating the detection of single nucleotide variants (SNVs) [19, 23, 24]. However, MDA-based WGA suffers from high allele dropout (ADO) rate. ADO is a technical failure to provide measurement of both alleles, and is the leading cause of false negatives (FNs) in SNV detection [19,25]. On the other hand, the infidelity of polymerase enzymes may lead to false positive (FPs) SNVs in SNV detection [23]. Such false positive SNVs are extremely excessive in C:G>T:A transitions [26]. Finally, MDA-based WGA technologies render low amplification breadth over the genome [26], leading to severe missing calls in SNV detection. Technically, FNs are generated due to the lack of variant-supporting reads whereas missing calls are due to the lack of reads, both variant- and reference-supporting, covering the mutation site.

In the past decade, a myriad of bioinformatics tools have been developed to tackle the ITH problem specifically designed for scDNAseq data while considering its unique error profile. These bioinformatics tools mainly have three functions, inferring the evolutionary history of cancer cells, characterizing the clonal composition of the cells, and inferring the genotype of all the mutation sites for each cell. Some of the tools infer all of the three, whereas some only aim at the latter two. In general, if a tool infers the evolutionary history of cancer cells, it can also characterize the clonality of the cells and infer the genotype of the mutation sites. Those tools that can jointly infer the evolutionary history and characterize the clonality of cancer cells include but are not limited to SCITE [27], OncoNEM [28], Sifit [29], SiCloneFit [30], SASC [31], SPhyR [32], and GRMT [33].

The output of these tools typically consist of an evolutionary tree delineating the cancer evolution history. The class of the tree varies among phylogenetic trees in which each node represents a single cell, mutation trees in which each node represents a new mutation, and clonal lineage trees in which each node represents a subclone of single cells [34]. No matter which class of the tree is used, the single cells can always be placed on the tree. Some tools, like SCITE [27], added an additional step to attach them to the leaf nodes.

A post-processing step is then required to convert the placement of the cells on the tree to the clustering of the cells. In particular, on a clonal lineage tree or a mutation tree, the cells attached to the same node are assigned to the same cluster. On a phylogenetic tree in which each leaf node represents a sequenced cell, clustering of

the cells has to be done by cutting the branches at a certain level on the tree so that the cells attached to the same branch can be assigned to the same cluster.

To obtain the consensus genotype of the mutations for each cluster, one can traverse the path on the tree from the root to the corresponding leaf node (or the node below which a branch is cut) that the cells of the same cluster are attached to. This consensus genotype profile also serves as the corrected genotype profile for all the cells assigned to the same cluster.

While it is desirable to have both the cancer evolutionary history and the clonal composition inside a tumor, it is computationally expensive to infer the cancer evolutionary tree due to the vast number of trees that the algorithms have to search and evaluate. The ever-increasing number of cells in single-cell sequencing field makes it even more challenging to infer an evolutionary tree in terms of the cost of computational resources as well as the turn-around time [25]. Clinically, what a medical provider needs is mainly the number of clones in the tumor for the prognosis purpose, not necessarily the whole evolutionary tree, although the cancer evolutionary tree is helpful in understanding how cancer cells evolve. In such sense, clustering the cells without inferring the evolutionary tree becomes essential due to its less demanding of the computational resources and fast turn-around time [25]. As a matter of fact, in the case when the evolutionary history of the cancer development is of interest, separating the clustering of the cells into subclones and the inference of the cancer evolutionary history is becoming a future trend, considering the increasing number of single cells sequenced in each sample [25, 35]. In this way, not only can the genotype errors be corrected during the cell clustering stage, but also the lineage tree is searched based on the subclones (clusters of the cells) instead of the individual cells. In all, a separate cell clustering step provides a tremendously more precise and smaller search space for the search of lineage tree.

The problem of clustering, the procedure of identifying similar groups given a set of data, has been addressed by many machine learning tools. Some of them require the knowledge of the number of clusters (in our context, it is the number of clones of cells), for example, K-means [36], mixture model using EM algorithm [37], hierarchical clustering [38], etc. Others do not require the knowledge of the number of the clusters, for example, mean-shift [39], Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [40], and more advanced ones such as shared nearest neighbor (SNN) [41] algorithm. In recent years clustering has also been related with the problem of community detection where each data point (in our case, cell) represents a node in a graph, and the similarity between every pair of data points represents an edge between two nodes. Community detection algorithms, such as Infomap [42] and Louvain [43] method, can be re-purposed for the detection of clusters given the distance or similarity between each pair of the cells in terms of their mutation profiles.

Despite the plethora of clustering method in the realm of machine learning, only a few clustering methods are available that are specifically designed for scDNAseq data. Yet the conventional clustering method cannot be directly applied to scDNAseq data because of scDNAseq's unique error profile. Specifically, the false negative, false positive and missing rates shall be incorporated in the design of the

clustering algorithm. The few scDNAseq-based tools that aim at only the cell clustering, not the lineage tree inference, are SCG [44], BnpC [25], SCClone [45], RobustClone [46] and ARCANE-ROG [47]. Here we skip discussing RobustClone because it does not have a friendly user interface and is not as popular. ARCANE-ROG is also omitted in our benchmark due to that it is not open-source. Of SCG, BnpC and SCClone, SCG is the first tool published to cluster scDNAseq data. It uses mean-field variational inference to infer the probabilities of cell assignment and the corrected genotype. SCG is unique in that it was designed for multiple data types, including binary genotypes, where the mutation is either existent or absent; ternary genotypes, where the mutation could be homozygous variant, heterozygous variant and homozygous reference; as well as the data with doublets. When SCG is run in the doublet mode, it is called D-SCG. BnpC [25] is a fully non-parametric Bayesian method that can handle noisy data in a large quantity. BnpC is scalable to handle a large number of clusters as well as thousands of cells. SCClone [45] uses a probability mixture model to cluster single cells into distinct clusters and uses an Expectation Maximization (EM) algorithm to infer the model parameters. Since mixture model requires the prior knowledge of the number of clusters, SCClone performs the EM algorithm for each possible cluster number and searches for the optimal cluster number. SCClone's results depend heavily on the initial values of the model parameters such as false positive and false negative rates. It is worth noting that celluloid [48] and AMC [49] are two mutation clustering methods for scDNAseq data. Since this study focuses on clustering cells instead of clustering mutations, we do not include them in the benchmark and discussion.

Despite the importance of cell clustering based on scDNAseq, there has not been a comprehensive benchmark on the existing cell clustering methods. Therefore their performance is unclear under different settings. Moreover, despite the fact that SCClone benchmarked the above-mentioned methods on their simulated data, we found that the parameters of the methods could have been better tuned to improve the performance of the tools, with the help of the correspondences with the authors. Thus the conclusion SCClone made regarding the comparison of the three methods is not necessarily reliable. On the other hand, although SCClone's simulation covered a few important parameters, the simulation is still not comprehensive. What is especially interesting but missing is the effect of the contrast of the cluster sizes on the clustering results. This is relevant because small subclones are more challenging to characterize due to the lack of representative cells.

In this context, we developed a simulator which allowed us to comprehensively benchmark the three state-of-the-art scDNAseq cell clustering methods, SCG, BnpC and SCClone, in terms of their accuracy of clustering, accuracy of the inference of the genotypes, and the consumption of computational resources. The various parameters allowed us to discuss the advantages and disadvantages of the three methods under different settings. In particular, what is new but essential in this benchmark study is the simulation of varying variance of cluster sizes. We modified the tree construction algorithm, the Beta splitting model used in SimSCSnTree [50, 51], so that we can tune a parameter in the simulator to increase or decrease the variance of the cluster sizes. The benchmark of the resulting simulated dataset is indicative of the robustness of the clustering method for the cases when the

contrast of the cluster sizes is big or small. In the following sections, we describe our simulator in detail, followed by the design of the simulation experiments, i.e., the list of datasets generated from varying parameters in the simulator. We briefly describe the comparison metrics that will be applied to measure the results of all three methods, SCG, BnpC and SCClone, as well as the parameter setting of these three methods. We then discuss the simulation results for each dataset, followed by the discussion of the results of a real dataset CRC2 [52]. We summarize the study by making recommendation of which tool to use under which setting in our discussion and conclusion.

Description of Our Simulator

To fully examine the performance of SCG, BnpC and SCClone, we designed a set of simulated data. For each simulated data set, we first simulated a clonal tree on which each leaf node represented a subclone of cells. Thus the number of leaves was the true number of clusters in cell clustering. The branches on the clonal tree were simulated such that each branch followed an exponential distribution with λ equaled to 1. The simulator distributed a given total number of mutations, M, to all the branches according to their lengths. The tree structure was simulated in the following way. Starting from the root node that was free from any mutations, we split a leaf node in the tree. When there was only a root node, it had to be selected to split. Once a node was split, it became the parent node of the two newly formed leaf nodes and itself was no longer a leaf node. Each node was given an interval whose start and end were within 0 and 1, and whose length was ≤ 1. Root node's interval was $[0, 1]$. The union of the two leaf nodes' intervals was the same as that of the parent node's interval, i.e., the child nodes split the parent node's interval. The ratio of the interval length of the two child nodes was decided by a sampling from the Beta distribution whose α value was fixed as 0.5 and β value was a variable that we varied in a set of simulated data. This variable was referred to as "Beta splitting variable". The chance of a leaf node being selected to split was in proportion to one minus its interval length instead of the interval length. This is a subtle but key difference between our approach and SimSCSnTree [50] whose leaf node's chance to be selected to split is in proportion to its interval length. Our "reverse splitting" strategy was intentionally designed in such a way so that nodes with big intervals had less chance to split and thus remained big clones, whereas nodes with small intervals continued to split and became even smaller clones. This procedure helped us to effectively use the Beta splitting variable to control the contrast of the sizes of the clusters and further allowed us to examine the three methods' accuracy in cases when cluster sizes differed to different degrees. The whole process stopped once the number of leaf nodes reached the desired number.

After generating the tree structure, we assigned a given number of cells, N, to all the leaf nodes in proportion to their interval lengths.

Once the cells were assigned to leaf nodes, for each leaf node x, we walked the path from the root node to x and assigned the SNVs on the branches along the path to the cells assigned to x. We then formed the genotype matrix G which was a cell locus matrix with the cells on the rows and SNV loci on the columns. $G_{i,j} = 1$ if SNV j was present in cell i. $G_{i,j} = 0$ if SNV j was absent in cell i. We used a

False positive rate	0.001, 0.01 (d), 0.05
False negative rate	0.1, 0.2 (d), 0.3, 0.4
Missing rate	0.2 (d), 0.3
# of cells	100, 500 (d), 1000, 1500
# of mutations	50, 200 (d), 500
Cluster number	4, 8 (d), 16, 32
Doublet rate	0 (d), 0.01, 0.05, 0.1
Beta splitting variable	0.05, 0.2 (d), 0.5

Table 1 Simulated datasets. Each line has a varying variable (first column) with the values in the second column. The default value is denoted by "(d)" on its right.

binary genotype matrix instead of a ternary one so that all of the three methods can be fairly compared. Notice that G matrix was the underlying true genotype matrix without any error or missing data. For a cell i, G_i was in fact the same as the true underlying consensus genotype for the leaf node (or cluster, as each leaf node represents a cluster) that cell i was assigned to. We then imputed the missing data, false positives and false negatives on the G matrix. We called the resulting noisy data matrix the D matrix. In the real scenario, D matrix is what can be observed from the sequencing data, whereas G matrix is the desired matrix to be inferred from the computational tools. In imputing the missing data, since we used the number "3" to represent the missing data, we randomly turned an entry, no matter whether its value was a 0 or 1, to be 3, according to a pre-set missing data probability. In imputing the false positives and false negatives, of all the entries that were 0 and 1, respectively, we flipped it to be 1 and 0 according to a pre-set false positive rate and false negative rate, respectively. The resulting matrix was the D matrix which was the input to the three tools as the cell locus matrix.

The Design of Simulation Experiments

We designed the simulation experiments in a comprehensive way in order to test the three methods from a variety of perspectives. Particularly, we tested the three methods by varying false positive rates, false negative rates, missing rates, number of cells, number of mutations, cluster number (or leaf node number), doublet rate and the Beta splitting variable that controlled the variance of cluster sizes. Beta splitting variable has never been investigated in the previous study and thus is unique in this study.

For each variable, we set up a default value. For each dataset, we varied only one variable while setting all other variables to be the default value, as seen in Table 1. We selected the ranges for each variable mainly according to the discussion in [19]. For example, since the doublet rate varies mainly up to 10%, we set up the maximum of doublet rate to be 10%. Notice that since BnpC and SCClone do not explicitly model doublets, we set the default doublet rate to be zero. To further examine how the existence of doublet cells may affect the accuracy of clustering, we varied doublet rates between 1% and 10%.

In terms of varying cell numbers, we set up a big range (100 to 1500) to test the algorithms since BnpC claimed to be scalable to a large number of cells. Similar to the number of cells, we set up a big range (50 to 500) of mutations to test the accuracy as well as the running time of all three methods.

For each such combination of variables, we repeatedly created the tree structure, assigned the mutations and cells, imputed the errors on the G matrix and created the D matrix for five times to overcome the random extremity.

Comparison Metrics

To compare the clustering results, we used four different metrics, and they are V-measure, running time, sensitivity and specificity. V-measure measures the clustering results given the ground truth, whereas sensitivity and specificity measure the consensus genotype inference. We also measured the genotyping accuracy of all three methods, but decided not to include it in this manuscript because of its redundancy with the sensitivity and specificity. In addition to these four metrics, we also evaluated the accuracy of the estimated number of clusters for two datasets, the varying number of clusters and the varying Beta splitting variable. Whether the three methods can correctly estimate the number of clusters on these two datasets is interesting because at a large number of underlying clusters or a large variance of cluster sizes, the estimation of the cluster number is extremely challenging.

Usage of SCG, BnpC and SCClone

The parameter setting to run each of the methods is described as follows.

- For SCG, we set the maximum number of clusters to be $N/4$ where N is the number of cells. The number of maximum iteration was set to be 10^6, and the gamma prior was set as "[9.99, 0.01, 1.0e-15][2.5, 7.5, 1.0e-15][1.0e-15, 1.0e-15, 1]". We used "[1, 1, 1.0e-15]" for the state prior and set the number of restart to be 20 with a random seed value selected from 0 to 10000. After running SCG for 20 restarts we chose the best seed value depending on the lower bound value, as suggested in [44]. We then used this best seed value and the above-mentioned parameters for a final run for SCG, whose results were taken as the final result.
- We ran BnpC by the default setting in all our simulated data sets. In the real data, we set *pp* value to be "0.75 0.75" as suggested by the authors since our real data set has less than 40 mutations.
- For SCClone, we used default parameters in which α was initialized to be 0.01 and β was selected from the grid search.

Since SCG does not directly output the cluster assignment for each cell but provides the posterior probability of each cell belonging to each cluster, we obtained the cluster assignment as follows. For each cell, we selected the cluster that the cell had the highest posterior probability with. If there were two clusters that had the tied highest probability for a cell, we randomly selected one of the clusters. Likewise, since SCG does not directly output the consensus genotype for each cluster but provides the posterior probability of a mutation being a certain genotype in a cluster, we binarized such a probability into the consensus genotype in the following way. For each inferred cluster and each mutation, we obtained the genotype by choosing the one with the highest posterior probability. If there were two genotypes that had the tied highest probability, we used the genotype that was supported by the most cells assigned to this cluster.

We intended to run D-SCG on our simulated data sets since D-SCG is doublet-aware. However, it takes extremely long time (longer than one day) to finish the jobs. We reasoned this is because D-SCG's computational complexity is $O(NK^2M)$, in which K is the number of clusters and M is the number of mutations. Thus D-SCG is quadratic with the number of clusters. We then switched back to SCG

and measured the clustering accuracy only for the non-doublet cells. Thus all three methods are not doublet-aware and the measurements do not include doublet cells throughout the entire benchmark study. However, the accuracy of clustering non-doublet cells in the presence of different levels of doublet cells is still of interest and we included this study in one of our simulation experiment.

Benchmark of Simulation Experiments

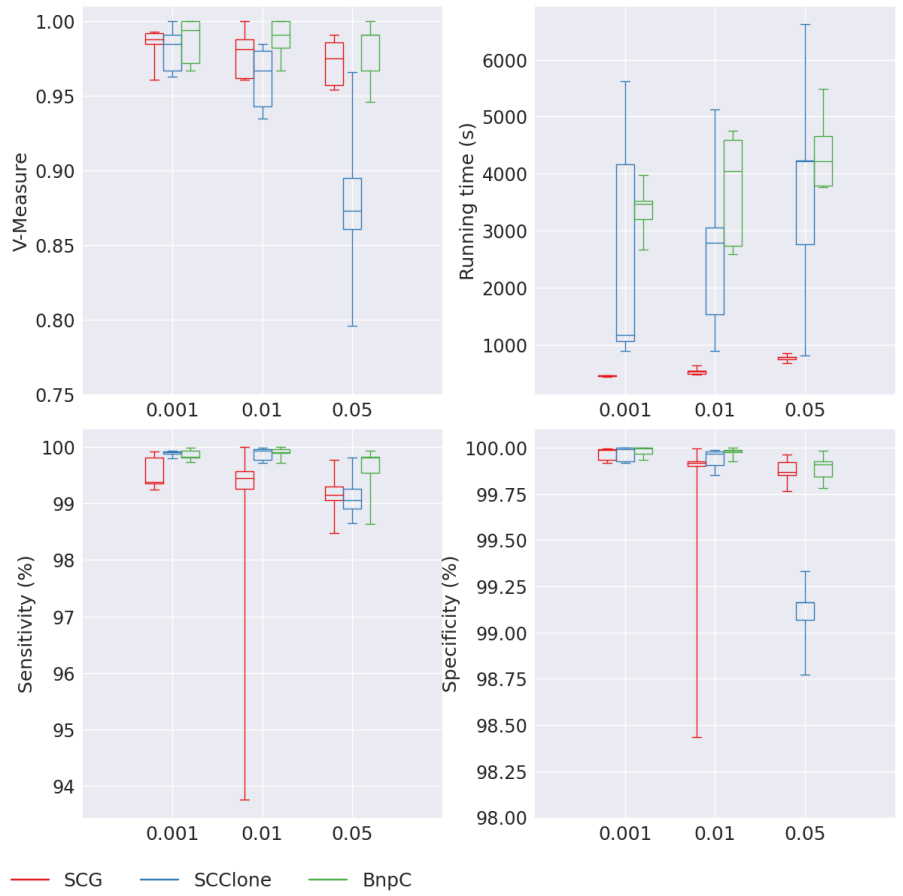

Figure 1 Performance of SCG (red), SCClone (blue) and BnpC (green) for varying false positive rate, the values of which are shown on the x-axes. The upper left, upper right, bottom left and bottom right panels are the V-measure, running time in seconds, genotyping sensitivity and genotyping specificity, respectively.

First, we varied the false positive rates in the simulated data while keeping all other parameters at their default values. As expected, all three methods' performance dropped when false positive rate increases (Fig. 1). Among the three methods, BnpC was the most stable method whose performance only slightly dropped with the increase of false positive rates. Specifically, BnpC's median of V-measure dropped from 0.994 to 0.991, compared with 0.985 to 0.873 for SCClone and 0.988 to 0.975 for SCG. BnpC's median of sensitivity did not drop, compared with 99.909% to 99.055% for SCClone and 99.375% to 99.153% for SCG. BnpC's median of specificity dropped from 99.995% to 99.911% , compared with 99.992% to 99.165% for

SCClone and 99.99% to 99.87% for SCG. Of all three methods, SCClone's performance was the worst when the false positive rate was high (0.05), whose V-measure's median was only 0.873, whereas both SCG and BnpC's V-measure were > 0.975 at the same false positive rate. SCG's V-measure was comparable to that of BnpC's for all three false positive rates, but its sensitivity and specificity were generally lower than that of BnpC's. All three methods' running time increased when false positive rate increased (Fig. 1). SCG's median running time increased from 447s to 783s, and it remained the lowest of all three methods for all false positive rates. SCClone and BnpC's median running time increased from 1170s to 4213s, and 3461s to 4215s, respectively. Thus SCClone's running time increased the most of all three methods with the increase of false positive rate.

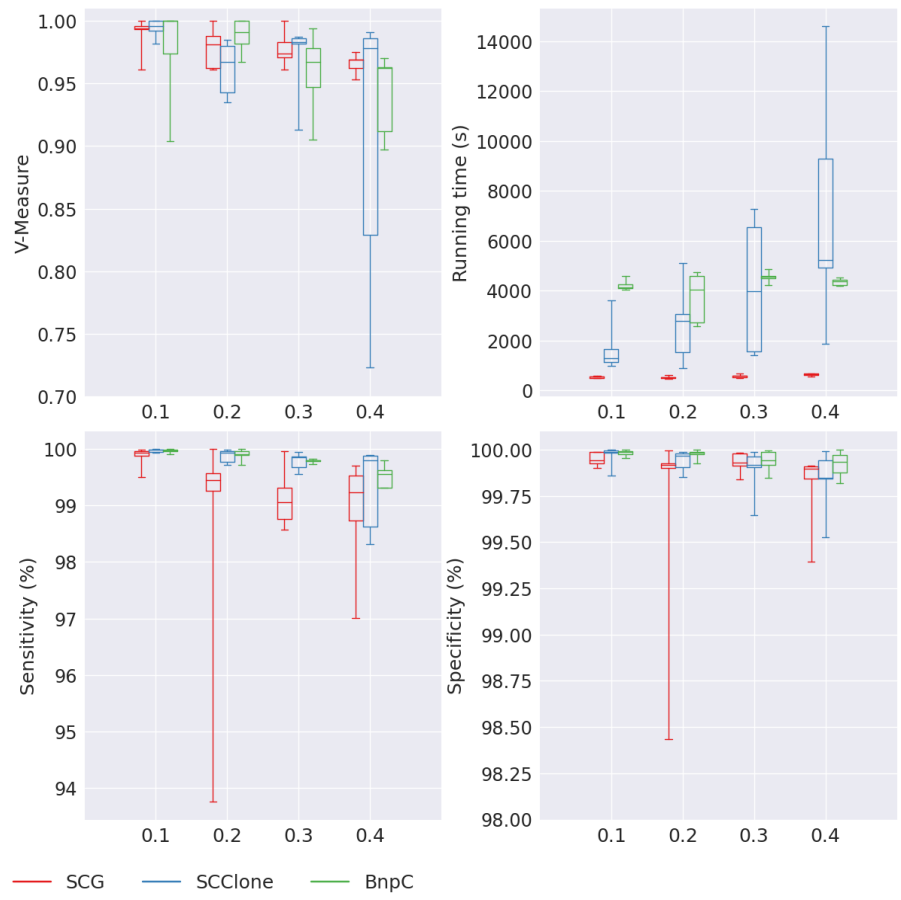

Figure 2 Performance of SCG (red), SCClone (blue) and BnpC (green) for varying false negative rates, the values of which are shown on the x-axes. The upper left, upper right, bottom left and bottom right panels are the V-measure, running time in seconds, genotyping sensitivity and genotyping specificity, respectively.

We further varied the false negative rate from 0.1 to 0.4. We found that the median of V-measure for SCG and BnpC both slightly dropped (0.994 to 0.969 for SCG and 1 to 0.962 for BnpC), seen in Fig. 2. Although the median of V-measure for SCClone also dropped slightly (from 0.996 to 0.978), its variance increased greatly (from 5.6E-05 for 0.1 to 0.0145 for 0.4), to be compared with that of SCG (from 2.5E-

04 for 0.1 to 7.08E-05 for 0.4) and BnpC (from 1.9E-04 for 0.1 to 2.8E-04 for 0.4). In terms of sensitivity, we noticed an obvious drop for BnpC (the median dropped from 99.976% for 0.1 to 99.561% for 0.4). Similarly, SCG's median sensitivity also dropped from 99.929% for 0.1 to 99.235% for 0.4. SCClone's median sensitivity did not drop as much as SCG and BnpC (from 99.985% for 0.1 to 99.803% for 0.4), but it showed an obvious pattern of increasing variance (from 0.001 for 0.1 to 0.59 for 0.4). Of all false negative rates, SCClone's median sensitivity stayed the highest compared with SCG and BnpC. All three methods were comparable in terms of specificity in the increasing false negative rates, although for all three methods the overall specificity decreased when false negative rate increased. Of all three methods, SCClone's specificity dropped the most with the increase of false negative rate. Its median specificity dropped from 99.989% to 99.848%, to be compared BnpC whose median specificity dropped dropped from 99.989% to 99.936%, and SCG whose median specificity dropped from 99.942% to 99.897%. SCClone's high sensitivity and dropping specificity indicated that SCClone tends to over-estimate the false negative rate when the false negative rate is high, thus it over-corrects the false negative entries. From the aspect of running time, it was interesting to observe that SCClone's running time dramatically increased with the increase of false negative rate. Its median running time increased from 1281s to 5246s as shown in Fig. 2. SCG's running time increased negligibly slightly whereas BnpC's did not show an increasing pattern. Of all three methods, SCG's running time stayed the least, within 654s for all false negative rates. When false negative rate was at 0.1, SCClone's running time (median at 1281s) was less than that of BnpC's (median at 4142s). When false negative rate was at 0.4, SCClone's running time (median at 5246s) surpassed that of BnpC's (median at 4389s) and became the slowest of the three methods.

In terms of missing rate, SCG and BnpC's V-measure decreased slightly with the increase of the missing rate (Fig. 3), whereas SCClone's V-measure increased with the increasing missing rate. In fact, SCClone also showed a reverse trend of the running time when the missing rate increased, i.e., its median running time decreased from 2787s to 1122s when the missing rate increased from 0.2 to 0.3. These data showed that SCClone's V-measure and running time were robust in the event of increasingly uncertain data. On the other hand, we did not observe a dramatic increase of running time for SCG and BnpC in terms of the increasing missing rate (Fig. 3).

All three methods' V-measure, sensitivity and specificity increased when the number of cells increased whereas BnpC's V-measure, sensitivity and specificity remained the highest for all the number of cells among the three methods (Fig. 4). All three methods' running time increased dramatically (Fig. 4), and SCClone's rate of increased running time in terms of the increasing cell number was the highest, from 108s to 12418s when the cell number increased from 100 to 1500. Specifically, when cell number was 100, SCClone, whose median running time was 108s, was just slightly slower than SCG's whose median running time was 65s. Both were much faster than BnpC, whose median running time was 2793s. When cell number was 500, SCClone's median running time (2787s) surpassed that of SCG's (522s) and was comparable to BnpC (4046s). When the number of cells increased to 1000,

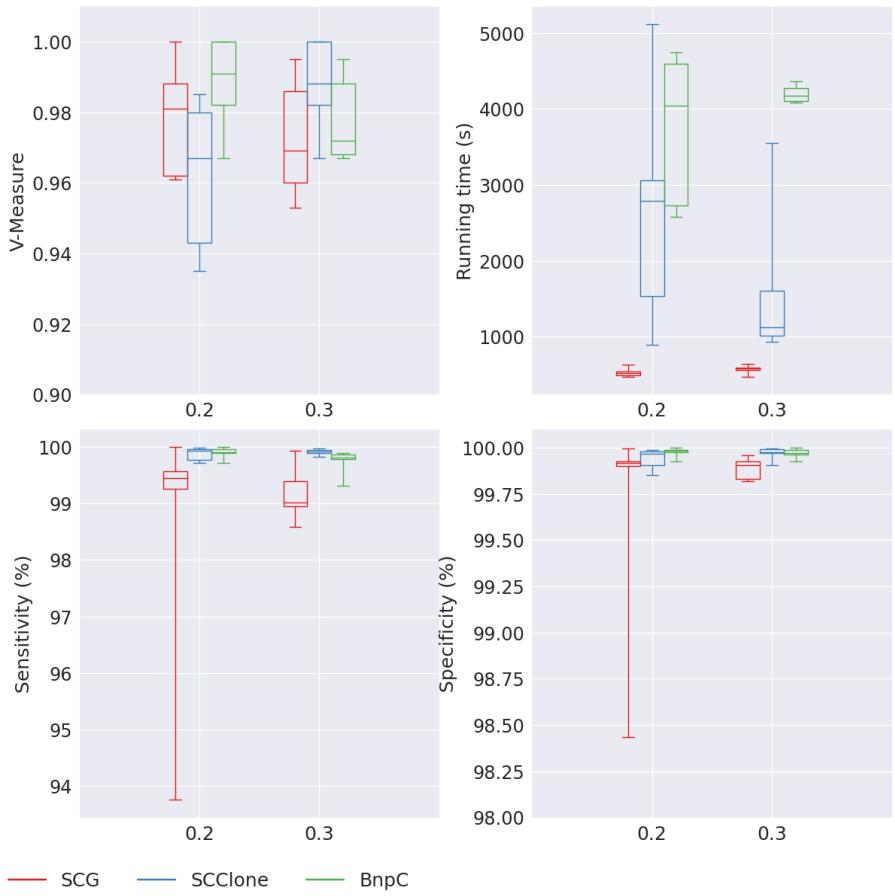

Figure 3 Performance of SCG (red), SCClone (blue) and BnpC (green) for varying number of missing rates, the values of which are shown on the x-axes. The upper left, upper right, bottom left and bottom right panels are the V-measure, running time in seconds, genotyping sensitivity and genotyping specificity, respectively.

SCClone's median running time increased to 8620s, much higher than that of SCG (2703s) and BnpC (3747s). Such situation worsened when the number of cells further increased to 1500, when SCClone's median running time reached 12418s, to be compared with 5712s and 7178s for BnpC and SCG, respectively. For cell numbers at 100, 500 and 1000, SCG's running time was always the lowest of the three methods. This was not the case when cell number further increased to 1500, at which SCG's running time surpassed that of BnpC's, resulting in BnpC being the fastest among the three methods. To summarize, the increase of running time in terms of the increasing cell number was almost linear for BnpC (p-value ≈ 0.03), and quadratic for SCClone and SCG (p-value ≈ 0.01 and ≈ 0.005, respectively).

Similar to the case of increasing number of cells, the V-measure, sensitivity, and specificity of all three methods increased with the increasing number of mutations (Fig. 5). Again we observed that BnpC's V-measure, sensitivity, and specificity remained the highest of all three different numbers of mutations among the three methods. As expected, the running time increased for all three methods with the increasing number of mutations (Fig. 5). Specifically, medians of SCG and BnpC's

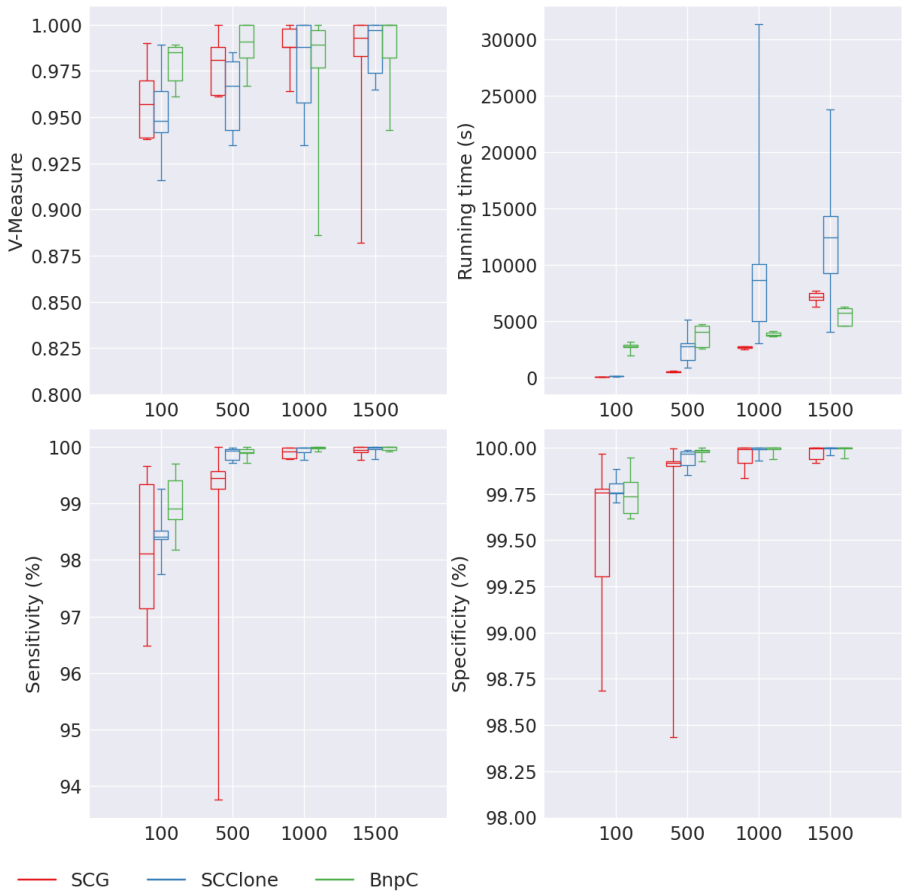

Figure 4 Performance of SCG (red), SCClone (blue) and BnpC (green) for varying number of cells, the values of which are shown on the x-axes. The upper left, upper right, bottom left and bottom right panels are the V-measure, running time in seconds, genotyping sensitivity and genotyping specificity, respectively.

running time increased from 386s to 963s, and from 1199s to 8196s, respectively. Unlike the case of increasing number of cells, SCClone's median running time did not increase as much as that of BnpC's for the increasing number of mutations, from 974s to 2258s. However, the variance of SCClone's median running time increased dramatically when the number of mutation was 500.To summarize, BnpC was affected the most in terms of the running time in the increasing number of mutations. When the number of mutations was as high as 500, it became the slowest of the three methods.

We observed decreasing V-measure, sensitivity and specificity when the number of clusters increased for all three methods (Fig. 6). The drop of BnpC's performance was the smallest, showing that BnpC was the most stable of the three in the face of growing cluster size. Specifically, BnpC's median V-measure dropped from 1 to 0.919 when clone size increased from 4 to 32, to be compared with from 0.975 to 0.868 for SCG, and from 1 to 0.914 for SCClone. Moreover, we also observed that BnpC's V-measure remained the highest on almost all the cluster sizes among the three methods. Comparing SCClone and SCG, the drop of V-measure

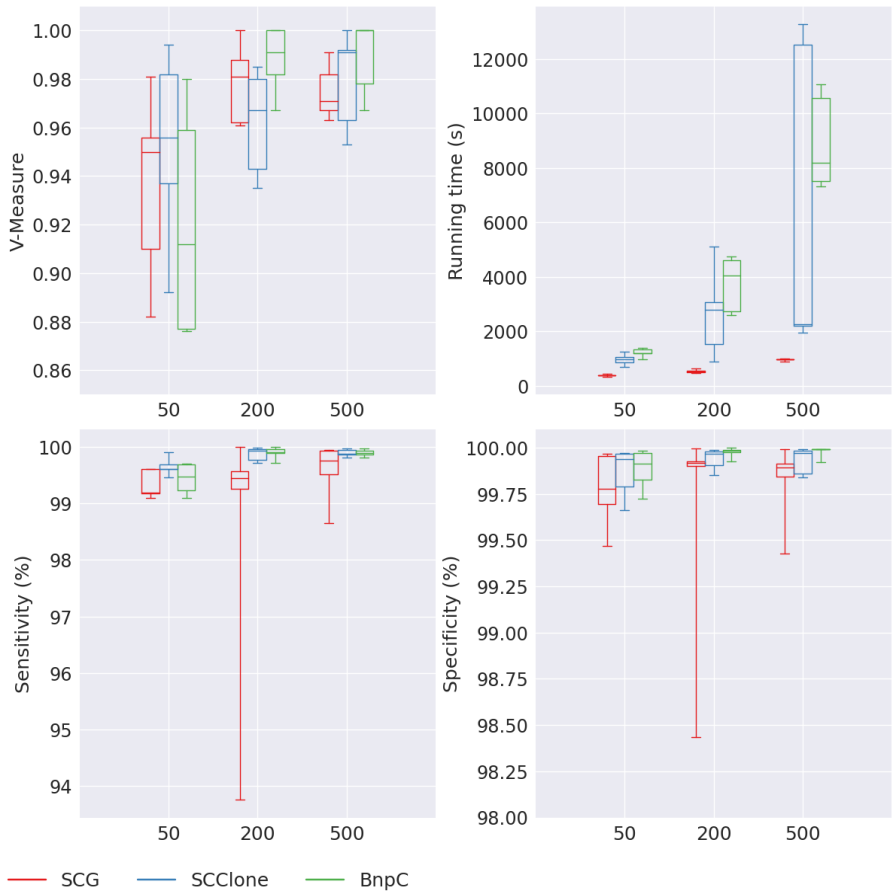

Figure 5 Performance of SCG (red), SCClone (blue) and BnpC (green) for varying number of mutations, the values of which are shown on the x-axes. The upper left, upper right, bottom left and bottom right panels are the V-measure, running time in seconds, genotyping sensitivity and genotyping specificity, respectively.

for SCClone was not as much as that of SCG's in the increasing cluster size. The drop of both sensitivity and specificity was lower for BnpC than SCG and SCClone. In detail, BnpC's sensitivity dropped from 100% to 98.478%, to be compared with SCClone's dropping from 100% to 97.835% and SCG's dropping from 99.771% to 97.55%. BnpC's specificity dropped from 100% to 99.816%, to be compared with SCClone's dropping from 100% to 99.697% and SCG's dropping from 99.99% to 99.306%. This showed that BnpC was the most accurate and stable method among the three when there were a lot of subclones in the data. In terms of running time, SCClone's running time increased the most when the number of underlying clusters increased (Fig. 6). Specifically, its median running time increased from 654s when cluster number was 4 to 23036s when cluster number was 32, and the increase is almost exponential (p-value ≈ 0.13). SCG and BnpC's running time with respect to the increase of the number of clusters was relatively stable compared with that of SCClone's, median from 482s to 662s for SCG and from 3321s to 4636s for BnpC. In addition to the accuracy and speed, in this experiment we were also interested in the accuracy of the estimated number of clusters for all three methods. We ob-

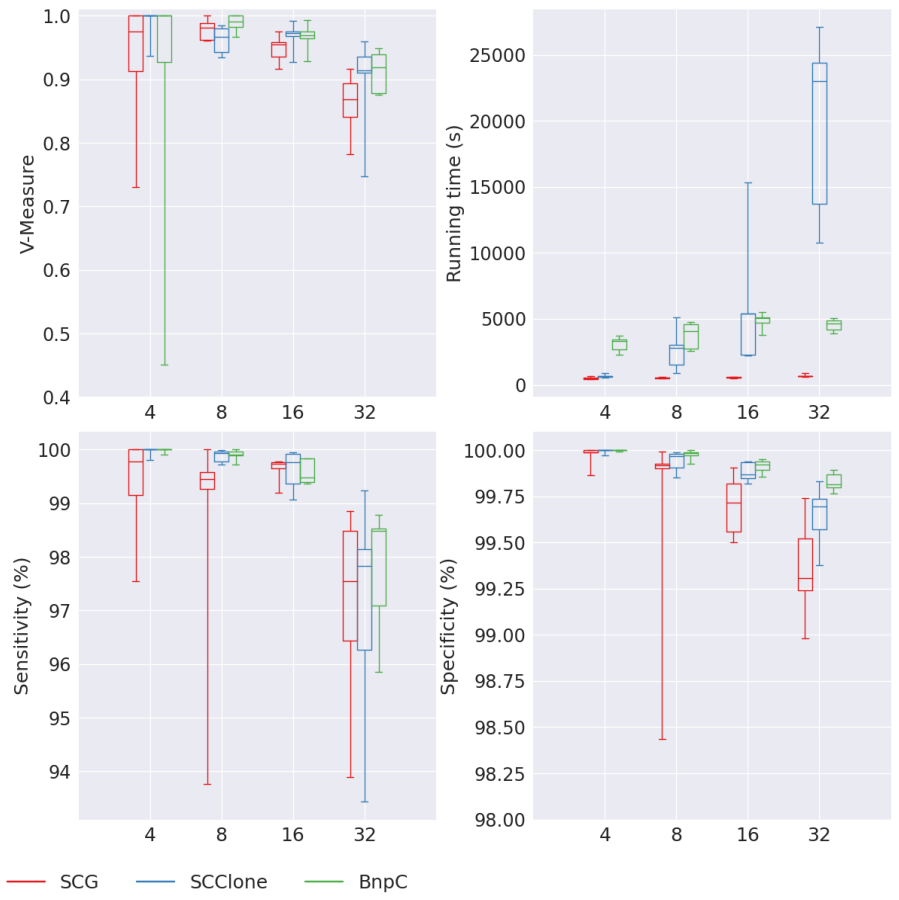

Figure 6 Performance of SCG (red), SCClone (blue) and BnpC (green) for varying number of clones, the values of which are shown on the x-axes. The upper left, upper right, bottom left and bottom right panels are the V-measure, running time in seconds, genotyping sensitivity and genotyping specificity, respectively.

served that of all three methods, SCClone was the most accurate in estimating the number of clusters (Fig. 7). Specifically, SCClone correctly estimated the number of clusters when the number of clusters was 4 and 8. When the number of clusters was 16, SCClone's estimated median number of clusters was estimated to be 15, thus very close to the true value. When the number of clusters was 32, SCClone's estimated median number of clusters was 28, to be compared with 11 for SCG and 14 for BnpC. Of all three methods, SCG's estimation of the number of clusters was the most inaccurate. When the number of clusters was small (4 or 8), SCG tended to over-estimate the number of clusters, whereas when the number of clusters was large (16 or 32), SCG tended to under-estimate the number of clusters. In fact, while both SCClone and BnpC estimated an increasing number of clusters when the underlying number of clusters increased from 4 to 32, SCG's estimated number of clusters plateaued when the true number of clusters increased from 16 to 32. While BnpC showed a trend of increasing estimated number of clusters and its estimation number of clusters was accurate when the underlying number of clusters was small (4 and 8), it under-estimated the number of clusters when the underly-

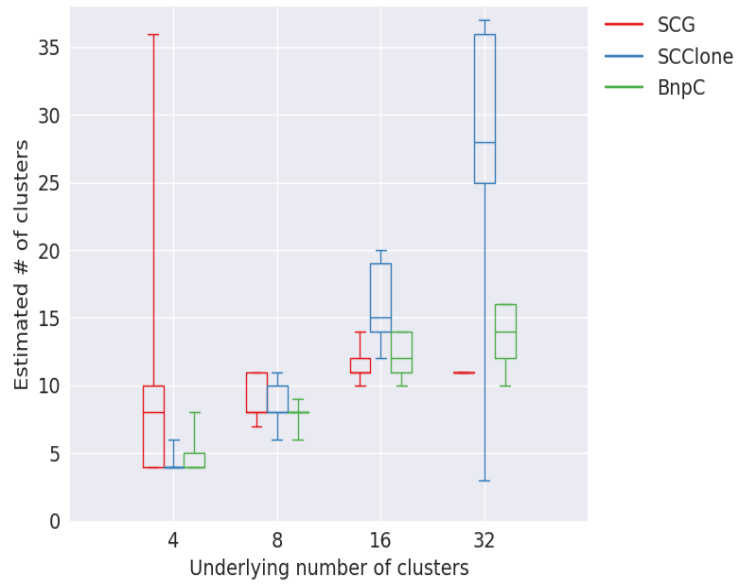

Figure 7 Estimated number of clusters (y-axis) for varying underlying number of clusters (x-axis).

ing number of clusters was large (16 and 32). Considering both the estimation of the number of clusters and the running time, we observed that although all three algorithms were agnostic of the number of clusters, SCClone's running time was sensitive to the underlying number of clusters and it rendered the most accurate estimated number of clusters especially when the underlying number of clusters was high. We thus concluded that SCClone had a trade-off of the running time for a more accurate estimation of the number of clusters.

We observed a trend of dropping V-measure in the increasing doublet rate for SCG (Fig. 8). Its median decreased from 0.981 when there was no doublet to 0.957 when the doublet rate was 0.1. Notice that we measured the V-measure only for the non-doublet cells, thus the measurement focused on how non-doublet cells can be clustered when there were doublet cells in the same pool. BnpC showed a similar trend, whose median dropped from 0.991 when there was no doublet to 0.98 when the doublet rate was 0.1. SCClone's median of V-measure did not show a clear trend with the increasing doublet rate, although when doublet rate was the highest (0.1), its median V-measure was the lowest (0.959). Among the three methods, BnpC's V-measure was the highest when the doublet rate was above 0.05. Interestingly, none of the three methods showed a clear trend of dropping sensitivity in the increasing doublet rate. Moreover, BnpC's did not show a trend of dropping specificity when the doublet rate increased. However, SCG and SCClone showed a slight decrease of median specificity (from 99.92% to 99.791% for SCG and from 99.969% to 99.717% for SCClone). In terms of running time, BnpC's increased the most of all with the increase of doublet rate, whose median running time increased from 4046s to 6535s, and remained the slowest of all three methods for all doublet rates (Fig. 8). SCG's running time was not affected by the increase of the doublet rate. Similarly, there

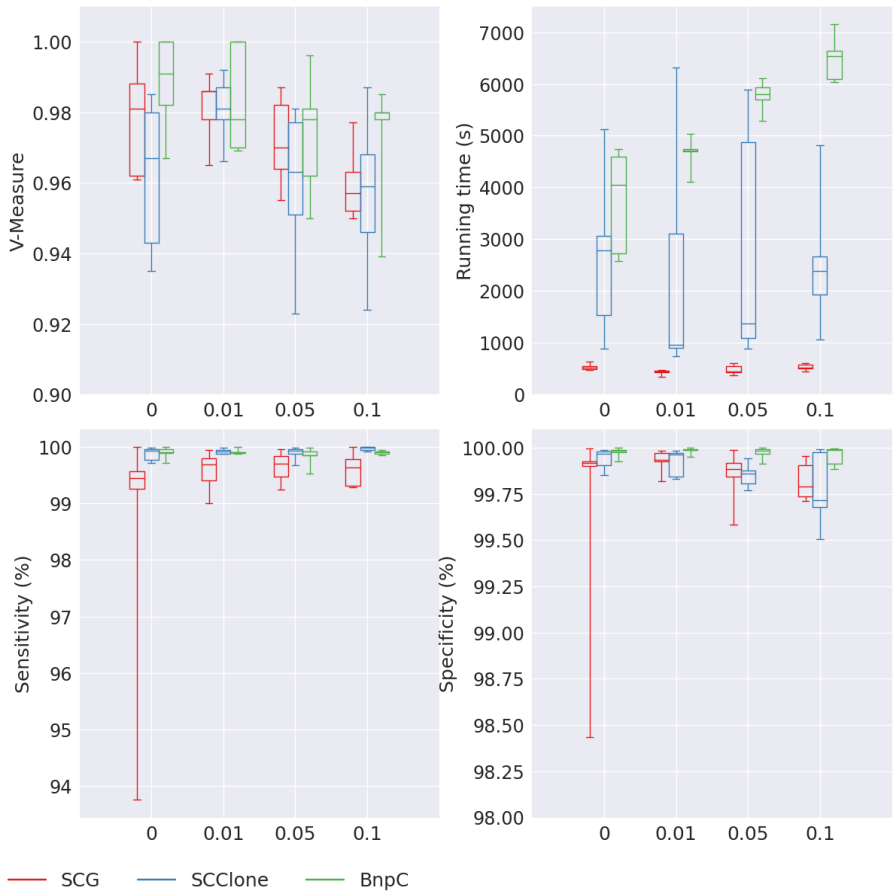

Figure 8 Performance of SCG (red), SCClone (blue) and BnpC (green) for varying doublet rate, the values of which are shown on the x-axes. The upper left, upper right, bottom left and bottom right panels are the V-measure, running time in seconds, genotyping sensitivity and genotyping specificity, respectively.

was not a clear trend of SCClone's running time when the doublet rate increased, although SCClone was slower than SCG for all doublet rates.

Lastly, we applied the three methods to the dataset with varying Beta splitting variable. The Beta splitting variable was used to control the variance of the cluster sizes so that the smaller the Beta splitting variable, the bigger the variance. Specifically, in this dataset, corresponding to the Beta splitting variable at 0.05, 0.2, and 0.5, the average standard deviation of the cluster sizes are 102, 71.1 and 60.32, respectively. We found that all three methods' performance dropped with the decreasing Beta splitting variable and the increasing variance of the cluster sizes (Fig. 9). We notice that when the Beta splitting variable was the smallest (0.05), both SCClone and BnpC's performance was not stable. The lowest V-measure of SCClone and BnpC dropped as low as 0.709 and 0.458, respectively, compared with that of SCG's whose lowest V-measure was 0.88. All three methods had high V-measure when the Beta splitting variable was the highest (0.5), all above 0.95. The same pattern can be observed in the three measurements in terms of genotype sensitivity and specificity. Specifically, when the Beta splitting variable was

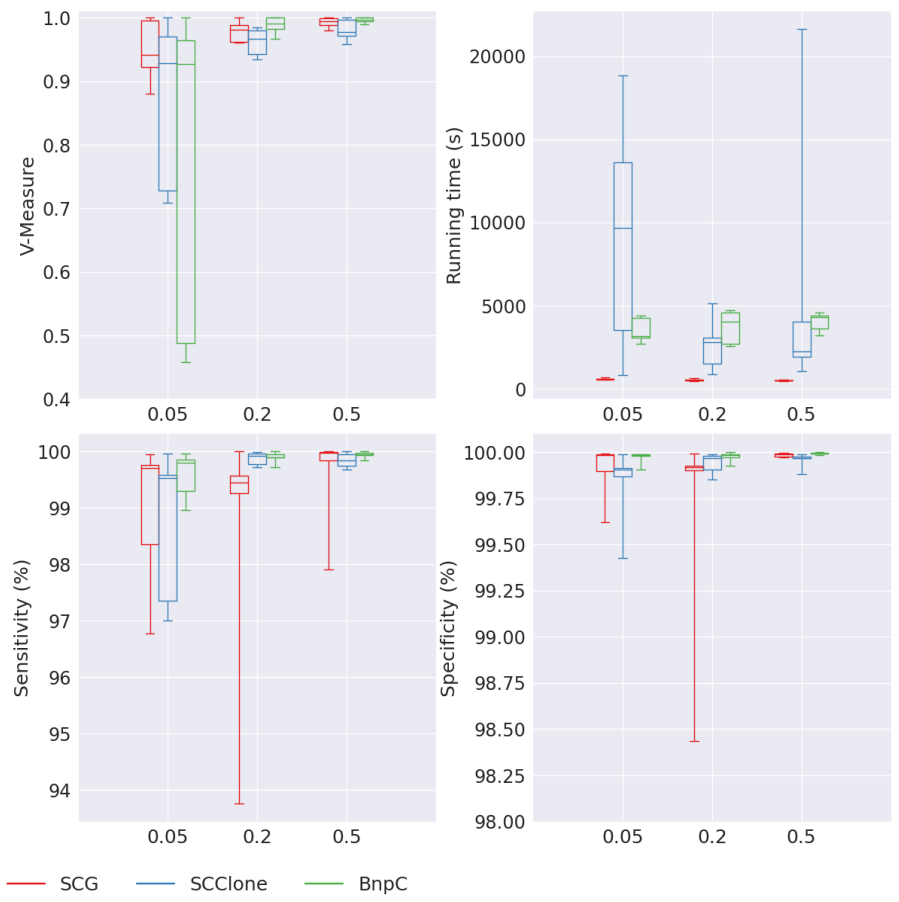

Figure 9 Performance of SCG (red), SCClone (blue) and BnpC (green) for varying Beta splitting variable, the values of which are shown on the x-axes. The upper left, upper right, bottom left and bottom right panels are the V-measure, running time in seconds, genotyping sensitivity and genotyping specificity, respectively.

the smallest (0.05), the lowest sensitivity was worse for SCG and SCClone than that of BnpC's (96.78% for SCG, 97.01% for SCClone and 98.959% for BnpC). But almost all three methods' V-measure were above 0.99 when the Beta splitting variable was 0.5. This showed that when the clone sizes did not differ from each other greatly, all three methods achieved good results and their performance was comparable. However, when there are both both big and small clones in a tumor, it should be expected that the clustering accuracy drops for all three methods. In terms of running time, both SCG and SCClone's median running time increased with the decrease of the Beta splitting variable (Fig. 8). Nevertheless, SCClone's running time increased much more dramatically than that of SCG's (p-value ≈ 0.14 for the exponential significance test). Specifically, its median running time increased from 2249s to 9674s, to be compared to SCG's that increased from 486s to 598s. BnpC's running time showed an opposite trend, however, whose median running time decreased from 4304s to 3179s when the Beta splitting variable decreased from 0.5 to 0.05. In all, all three methods showed that it is easier to cluster the cells when the variance of the cluster sizes is smaller. When the variance of the cluster sizes

is large, neither SCClone nor BnpC's performance is reliable. Moreover, SCClone's running time increased exponentially when the variance of cluster sizes increased. We further looked into the accuracy of the estimation of the number of clusters for the three methods for the varying Beta splitting variable. This is an interesting study because when Beta splitting variable is small, the variance of cluster sizes is large, and it is expected to be more challenging to correctly estimate the number of clusters. Specifically, small clusters with only a few cells are especially difficult to single out since these clusters have weak signals and together with the noises in the data, the signal-to-noise ratio is low. We found that all three methods correctly estimated the underlying number of clusters for all Beta splitting variables, i.e., their medians equal to eight, except SCClone which over-estimated the number of clusters when Beta splitting variable was as low as 0.05. This indicates that SCClone tends to over-segment the clusters when there are both small and large clusters in the pool.

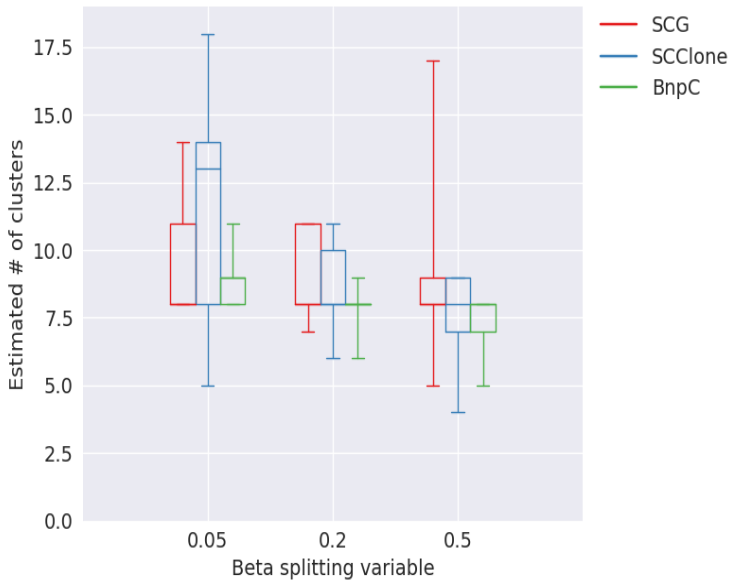

Figure 10 Estimated number of clusters for varying Beta splitting variable. The underlying number of clusters is 8 for all Beta splitting variables.

Real Data Analysis

We applied the three methods to a human colorectal cancer sample CRC2 [52]. Thirty-four primary cells (denoted as PA cells) and thirty-three metastatic cells (denoted as MA cells) had been sequenced by scDNAseq in [52] and used in this study. According to [52], thirty-six mutations were genotyped and shown in the cell locus matrix, out of which 137 entries were missing. [52] used SCITE [27] to infer the mutation tree and the underlying genotype of the cells. SCITE inferred the existence of two metastatic clones, categorizing 15 of the 33 metastatic cells to the first metastasis (M1) and 13 to the second metastasis (M2). For the rest of the 5 MA cells, SCITE categorized them as primary cells. According to both SCITE and

the original resection position from the patient, it is clear that in total there are three underlying clusters in this dataset. SCITE also estimated the false positive and negative rates for this sample to be 0.0174 and 0.1256, respectively.

To examine the clustering performance of SCG, BnpC and SCClone on the real datasets, we applied these methods on CRC2. We kept the parameter settings for the three methods the same as the setting for simulated datasets except increasing the "-pp" value for BnpC to 0.75 0.75 as suggested by the authors due to the decreased number of mutation sites.

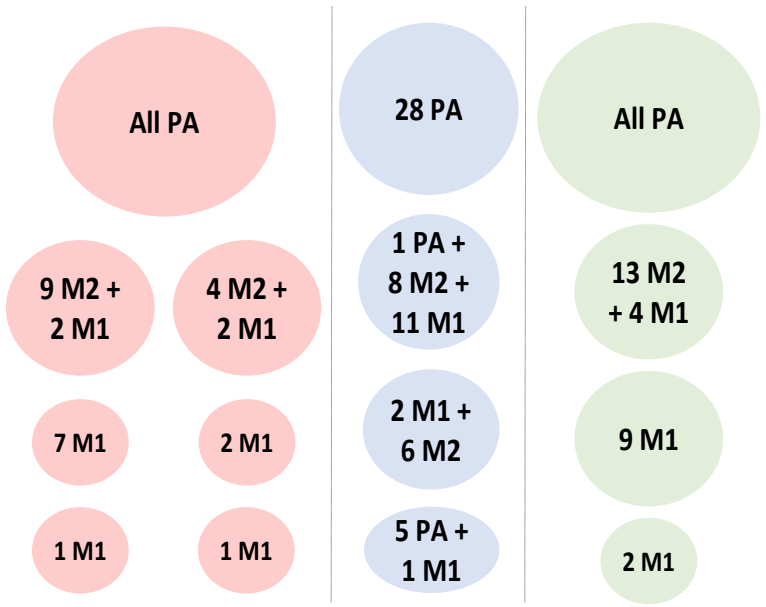

Figure 11 Illustration of the clustering results from SCG (red background), SCClone (blue background) and BnpC (green background).

We found that five MA cells (MA39, MA41, MA42, MA44 and MA9) that were categorized as primary cells by SCITE were clustered with other metastatic cells instead of primary cells by all the three clustering methods being investigated (Fig. 11). This indicates that clustering methods, due to its simplicity and smaller search dimension, may lead to a higher clustering accuracy than the methods that consider the tumor lineage and clonality simultaneously. Among the three methods, we observed that BnpC's clustering result was the most consistent with the tumor cells' original location and SCITE's inference. In specific, we observed that BnpC and SCG clustered all the PA cells together in one cluster whereas SCClone broke the PA cells into two clusters, one of which was also mixed with a M1 cell. BnpC clustered all the M2 cells together, but mixed four M1 cells (MA27, MA48, MA88 and MA94) with this M2 cluster. SCG separated M2 cells into two clusters and also mixed the four M1 cells mentioned above, two for each, with these two M2 clusters.

BnpC clustered the remaining M1 cells into two clusters, whereas SCG broke these M1 cells into four clusters, two of which were composed of only one cell. This is consistent with the observation we found from the simulation study, where SCG over-segmented clusters when the underlying cluster number was small. Interestingly, both SCG and BnpC contained the same cluster that was composed of only

two M1 cells, MA29 and MA90, showing the challenge of clustering these two cells together with other MA cells. Different from SCG and BnpC which clearly separated M1 and M2 cells, SCClone mixed up M1 and M2 cells together into two cocktail clusters. Based on these observations, we conclude that SCG tended to break large clusters into smaller ones and does not cluster all the cells together according to their original tumor location. BnpC captured the clonality of the cells according to their original tumor location. Even though SCClone's estimation of the cluster number is the most accurate, it's clustering result is not as accurate as BnpC. These conclusions are consistent with what we observed in the simulation study.

Discussion

We examined three scDNAseq clustering methods, SCG, SCClone and BnpC in terms of their clustering and genotyping accuracy, running time and the accuracy of estimating the number of clusters from a benchmark study and a real data sample. In the benchmark study, we varied eight variables to comprehensively examine the performance of the three methods. We summarize our observation and make our recommendation under different settings in the following text.

In the face of high false positive rate, SCClone's performance in terms of both clustering accuracy and running time is the worst. Of the rest two methods, BnpC is slightly more accurate than SCG. However, SCG's running time is the least among the three for all false positive rates and is one order of magnitude faster than the other two methods. We thus recommend SCG in terms of high false positive rate.

When false negative rate is low, all three methods are comparable in their accuracy in clustering and genotyping. Since SCG's running time is the lowest, we recommend using SCG when the data has a relatively low false negative rate (≤ 0.2). When false negative rate is high (≥ 0.3), we recommend using SCClone due to its high V-measure and sensitivity. However, since SCClone takes one order of magnitude more time than SCG, and its running time increases with the increase of false negative rate, we recommend using SCG as well for a faster turnaround time at the cost of some genotyping accuracy (< 0.05 for V-measure and $< 0.6\%$ for sensitivity).

For different missing rates, we recommend using SCG due to its high clustering accuracy and speed.

The three methods' performance vary mostly on their speed in terms of different number of cells. We recommend using SCG when the cell number is smaller than 1000, and BnpC when the cell number is greater than 1000. When cell number is around 100, SCClone is also a good choice since its turnaround time is comparable to that of SCG's.

Since SCG remains the fastest for all numbers of mutations with a slightly lower clustering accuracy compared with BnpC in the face of large number of mutations, we recommend using SCG in terms of extremely large number of mutations.

We recommend both SCG and SCClone for small number of clusters (≤ 4) due to their speed and accuracy. However, when the number of clusters increases, we recommend BnpC due to its high clustering and genotyping accuracy and slightly increasing running time. However, should it be the case when a fast turnaround time is desired, we recommend SCG in the face of large underlying cluster number at the cost of some clustering accuracy. If an accurate estimation of the cluster

number is desired, we recommend SCClone throughout all the underlying number of clusters due to that SCClone is much more accurate in estimating the number of clusters than the other two methods.

In the face of high doublet rate, we recommend SCG for its speed. However, when accuracy is more desired than speed, we recommend BnpC due to its high clustering and genotyping accuracy.

Lastly, we recommend SCG when the variance of cluster sizes is either big or small due to that SCG is robust in terms of both the clustering and genotyping accuracy and the speed. Specifically, when the variance of cluster sizes is large, we do not recommend SCClone and BnpC due to their poor clustering accuracy in response to the data with both large and small clusters. In addition, SCClone tends to over-segment the clusters when there are both large and small clusters in the pool.

To summarize, although BnpC's clustering accuracy stays the highest for most cases, its running speed is less advantageous than SCG except when the number of cells is as high as 1500. SCG's clustering accuracy is next to BnpC's and there is only a small difference between the two methods' V-measures for most cases. When there are a large number of clusters (≥ 32), BnpC is advantageous in clustering and genotyping accuracy over SCG despite the cost of running time.

Although SCClone's accuracy is slightly worse than that of SCG and BnpC on most of the datasets, SCClone is much more accurate in estimating the number of clusters especially when the underlying number of clusters is high.

Conclusion

We developed a simulator that has eight varying parameters to benchmark three state-of-the-art scDNAseq cell clustering methods, which are SCG, BnpC and SCClone. We conclude that BnpC has the highest clustering accuracy of the three. SCG is the fastest and its clustering accuracy is very close to BnpC's for most of the datasets. However, we recommend BnpC when the cell number is high due to its scalability to the increasing cell numbers. SCClone is the most accurate in estimating the number of clusters in both small and large numbers of clusters, but its clustering and genotyping accuracy is slightly worse than SCG and BnpC. Moreover, its accuracy and running time are sensitive to many variables such as false negative rate, the number of cells, number of clusters and the variance of the cluster sizes.

All three methods are sensitive to the variance of cluster sizes. The bigger the variance, the worse the performance. To develop a method that is robust to high variance of cluster sizes is potentially a future work in scDNAseq cell clustering.

Acknowledgement

R.K. and X.M. were supported by the startup funding from Florida State University.

Data and Code Availability

The simulator described in this study, as well as the instructions to run the three methods along with the post-processing scripts are publicly available at https://github.com/compbiofan/clusteringBenchmark.

References

1. Feuk, L., Carson, A.R., Scherer, S.W.: Structural variation in the human genome. Nature Reviews Genetics **7**(2), 85 (2006)
2. Sharp, A.J., Cheng, Z., Eichler, E.E.: Structural variation of the human genome. Annu. Rev. Genomics Hum. Genet. **7**, 407–442 (2006)
3. Lupski, J.R., *et al.*: Structural variation in the human genome. New England Journal of Medicine **356**(11), 1169 (2007)
4. Aparicio, S., Mardis, E.: Tumor heterogeneity: next-generation sequencing enhances the view from the pathologist's microscope. Springer (2014)
5. El-Deiry, W.S., Taylor, B., Neal, J.W.: Tumor evolution, heterogeneity, and therapy for our patients with advanced cancer: How far have we come? American Society of Clinical Oncology Educational Book **37**, 8–15 (2017)
6. McGranahan, N., Swanton, C.: Clonal heterogeneity and tumor evolution: past, present, and the future. Cell **168**(4), 613–628 (2017)
7. Lawrence, M.S., Stojanov, P., Polak, P., Kryukov, G.V., Cibulskis, K., Sivachenko, A., Carter, S.L., Stewart, C., Mermel, C.H., Roberts, S.A., *et al.*: Mutational heterogeneity in cancer and the search for new cancer-associated genes. Nature **499**(7457), 214–218 (2013)
8. Burrell, R.A., McGranahan, N., Bartek, J., Swanton, C.: The causes and consequences of genetic heterogeneity in cancer evolution. Nature **501**(7467), 338–345 (2013)
9. Yap, T.A., Gerlinger, M., Futreal, P.A., Pusztai, L., Swanton, C.: Intratumor heterogeneity: seeing the wood for the trees. Science translational medicine **4**(127), 127–1012710 (2012)
10. Turajlic, S., Sottoriva, A., Graham, T., Swanton, C.: Resolving genetic heterogeneity in cancer. Nature Reviews Genetics **20**(7), 404–416 (2019)
11. Alizadeh, A.A., Aranda, V., Bardelli, A., Blanpain, C., Bock, C., Borowski, C., Caldas, C., Califano, A., Doherty, M., Elsner, M., *et al.*: Toward understanding and exploiting tumor heterogeneity. Nature medicine **21**(8), 846 (2015)
12. Oesper, L., Mahmoody, A., Raphael, B.J.: Theta: inferring intra-tumor heterogeneity from high-throughput dna sequencing data. Genome biology **14**(7), 80 (2013)
13. McGranahan, N., Swanton, C.: Clonal heterogeneity and tumor evolution: past, present, and the future. Cell **168**(4), 613–628 (2017)
14. Dagogo-Jack, I., Shaw, A.T.: Tumour heterogeneity and resistance to cancer therapies. Nature reviews Clinical oncology **15**(2), 81 (2018)
15. Dillekås, H., Rogers, M.S., Straume, O.: Are 90% of deaths from cancer caused by metastases? Cancer medicine **8**(12), 5574–5576 (2019)
16. Navin, N., Kendall, J., Troge, J., Andrews, P., Rodgers, L., McIndoo, J., Cook, K., Stepansky, A., Levy, D., Esposito, D., *et al.*: Tumour evolution inferred by single-cell sequencing. Nature **472**(7341), 90–94 (2011)
17. Wang, Y., Navin, N.E.: Advances and applications of single-cell sequencing technologies. Molecular cell **58**(4), 598–609 (2015)
18. De Bourcy, C.F., De Vlaminck, I., Kanbar, J.N., Wang, J., Gawad, C., Quake, S.R.: A quantitative comparison of single-cell whole genome amplification methods. PloS one **9**(8), 105585 (2014)
19. Zafar, H., Navin, N., Nakhleh, L., Chen, K.: Computational approaches for inferring tumor evolution from single-cell genomic data. Current Opinion in Systems Biology **7**, 16–25 (2018)
20. Dean, F.B., Hosono, S., Fang, L., Wu, X., Faruqi, A.F., Bray-Ward, P., Sun, Z., Zong, Q., Du, Y., Du, J., *et al.*: Comprehensive human genome amplification using multiple displacement amplification. Proceedings of the National Academy of Sciences **99**(8), 5261–5266 (2002)
21. Wang, Y., Waters, J., Leung, M.L., Unruh, A., Roh, W., Shi, X., Chen, K., Scheet, P., Vattathil, S., Liang, H., *et al.*: Clonal evolution in breast cancer revealed by single nucleus genome sequencing. Nature **512**(7513), 155 (2014)
22. Hou, Y., Song, L., Zhu, P., Zhang, B., Tao, Y., Xu, X., Li, F., Wu, K., Liang, J., Shao, D., *et al.*: Single-cell exome sequencing and monoclonal evolution of a jak2-negative myeloproliferative neoplasm. Cell **148**(5), 873–885 (2012)
23. Navin, N.E.: Cancer genomics: one cell at a time. Genome biology **15**(8), 1–13 (2014)
24. Mallory, X.F., Edrisi, M., Navin, N., Nakhleh, L.: Methods for copy number aberration detection from single-cell dna-sequencing data. Genome biology **21**(1), 1–22 (2020)
25. Borgsmüller, N., Bonet, J., Marass, F., Gonzalez-Perez, A., Lopez-Bigas, N., Beerenwinkel, N.: Bnpc: Bayesian non-parametric clustering of single-cell mutation profiles. Bioinformatics **36**(19), 4854–4859 (2020)
26. Estévez-Gómez, N., Prieto, T., Guillaumet-Adkins, A., Heyn, H., Prado-López, S., Posada, D.: Comparison of single-cell whole-genome amplification strategies. BioRxiv, 443754 (2018)
27. Jahn, K., Kuipers, J., Beerenwinkel, N.: Tree inference for single-cell data. Genome biology **17**(1), 1–17 (2016)
28. Ross, E.M., Markowetz, F.: Onconem: inferring tumor evolution from single-cell sequencing data. Genome biology **17**(1), 1–14 (2016)
29. Zafar, H., Tzen, A., Navin, N., Chen, K., Nakhleh, L.: Sifit: inferring tumor trees from single-cell sequencing data under finite-sites models. Genome biology **18**(1), 1–20 (2017)
30. Zafar, H., Navin, N., Chen, K., Nakhleh, L.: Siclonefit: Bayesian inference of population structure, genotype, and phylogeny of tumor clones from single-cell genome sequencing data. Genome research **29**(11), 1847–1859 (2019)
31. Ciccolella, S., Ricketts, C., Soto Gomez, M., Patterson, M., Silverbush, D., Bonizzoni, P., Hajirasouliha, I., Della Vedova, G.: Inferring cancer progression from single-cell sequencing while allowing mutation losses. Bioinformatics **37**(3), 326–333 (2021)
32. El-Kebir, M.: Sphyr: tumor phylogeny estimation from single-cell sequencing data under loss and error. Bioinformatics **34**(17), 671–679 (2018)

33. Yu, Z., Liu, H., Du, F., Tang, X.: Grmt: generative reconstruction of mutation tree from scratch using single-cell sequencing data. Frontiers in genetics, 970 (2021)
34. Davis, A., Navin, N.E.: Computing tumor trees from single cells. Genome biology **17**(1), 1–4 (2016)
35. Lan, F., Demaree, B., Ahmed, N., Abate, A.R.: Single-cell genome sequencing at ultra-high-throughput with microfluidic droplet barcoding. Nature biotechnology **35**(7), 640–646 (2017)
36. Lloyd, S.: Least squares quantization in pcm. IEEE transactions on information theory **28**(2), 129–137 (1982)
37. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the em algorithm. Journal of the Royal Statistical Society: Series B (Methodological) **39**(1), 1–22 (1977)
38. Nielsen, F.: Hierarchical clustering. In: Introduction to HPC with MPI for Data Science, pp. 195–211. Springer, ??? (2016)
39. Fukunaga, K., Hostetler, L.: The estimation of the gradient of a density function, with applications in pattern recognition. IEEE Transactions on information theory **21**(1), 32–40 (1975)
40. Ester, M., Kriegel, H.-P., Sander, J., Xu, X., *et al.*: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Kdd, vol. 96, pp. 226–231 (1996)
41. Ertöz, L., Steinbach, M., Kumar, V.: Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data. In: Proceedings of the 2003 SIAM International Conference on Data Mining, pp. 47–58 (2003). SIAM
42. Rosvall, M., Axelsson, D., Bergstrom, C.T.: The map equation. The European Physical Journal Special Topics **178**(1), 13–23 (2009)
43. Blondel, V.D., Guillaume, J.-L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. Journal of statistical mechanics: theory and experiment **2008**(10), 10008 (2008)
44. Roth, A., McPherson, A., Laks, E., Biele, J., Yap, D., Wan, A., Smith, M.A., Nielsen, C.B., McAlpine, J.N., Aparicio, S., *et al.*: Clonal genotype and population structure inference from single-cell tumor sequencing. Nature methods **13**(7), 573–576 (2016)
45. Yu, Z., Du, F., Song, L.: Scclone: Accurate clustering of tumor single-cell dna sequencing data. Frontiers in Genetics, 26 (2022)
46. Chen, Z., Gong, F., Wan, L., Ma, L.: Robustclone: a robust pca method for tumor clone and evolution inference from single-cell sequencing data. Bioinformatics **36**(11), 3299–3306 (2020)
47. Farswan, A., Gupta, R., Gupta, A.: Arcane-rog: Algorithm for reconstruction of cancer evolution from single-cell data using robust graph learning. Journal of Biomedical Informatics **129**, 104055 (2022)
48. Ciccolella, S., Patterson, M.D., Bonizzoni, P., Della Vedova, G.: Effective clustering for single cell sequencing cancer data. In: Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics, pp. 437–446 (2019)
49. Yu, Z., Du, F.: Amc: accurate mutation clustering from single-cell dna sequencing data. Bioinformatics **38**(6), 1732–1734 (2022)
50. Mallory, X.F., Nakhleh, L.: Simscsntree: a simulator of single-cell dna sequencing data. Bioinformatics **38**(10), 2912–2914 (2022)
51. Blum, M.G., François, O.: Which random processes describe the tree of life? a large-scale study of phylogenetic tree imbalance. Systematic Biology **55**(4), 685–691 (2006)
52. Leung, M.L., Davis, A., Gao, R., Casasent, A., Wang, Y., Sei, E., Vilar, E., Maru, D., Kopetz, S., Navin, N.E.: Single-cell dna sequencing reveals a late-dissemination model in metastatic colorectal cancer. Genome research **27**(8), 1287–1299 (2017)