Benchmarking clustering methods for single cell sequencing cancer data

Simone Ciccolella[†], Murray Patterson[†,*], Paola Bonizzoni and Gianluca Della Vedova

Experimental Algorithmics Lab (algolab.eu), Dipartimento di Informatica, Sistemistica e Comunicazione (DISCo), Università degli Studi di Milano-Bicocca, Milan, Italy

[†] Joint first author
[*] Corresponding author: murray.patterson@unimib.it

Abstract

Background. Single cell sequencing (SCS) technologies provide a level of resolution that makes it indispensable for inferring from a sequenced tumor, evolutionary trees or phylogenies representing an accumulation of cancerous mutations. A drawback of SCS is elevated false negative and missing value rates, resulting in a large space of possible solutions, which in turn makes infeasible using some approaches and tools. While this has not inhibited the development of methods for inferring phylogenies from SCS data, the continuing increase in size and resolution of these data begin to put a strain on such methods. One possible solution is to reduce the size of an SCS instance — usually represented as a matrix of presence, absence and missing values of the mutations found in the different sequenced cells — and infer this tree from the reduced-size instance. Previous approaches have used k-means to this end, clustering groups of mutations and/or cells, and using these means as the reduced instance.

Results. In this work, we benchmark a variety of commonly used methods aimed at clustering vector, or matrix data — here representing SCS instances: with a focus on how effective these methods are for the purpose of inferring tumor evolutionary trees from these SCS instances. A trend we observe is that those methods designed for clustering data of a *categorical* nature (SCS data having three categories: present, absent and missing) — namely k-modes, and a method we have devised called *celluloid* (see Methods) — perform much better than all of the other methods. We demonstrate that these categorical methods cluster mutations with high precision: never pairing too many mutations that are unrelated in the ground truth, but also obtain accurate results in terms of the phylogeny inferred downstream from the reduced instance produced by such a method. Finally, we demonstrate the usefulness of a clustering step by applying the entire pipeline (clustering + inference method) to a real dataset, showing a significant reduction in the runtime, raising considerably the upper bound on the size of SCS instances which can be solved in practice.

Keywords. Clustering, Single Cell Sequencing, Cancer progression

Background

A tumor, at the time of detection, usually by performing a biopsy on the extracted tissue, is the result of a tumultuous evolutionary process, originating from a single tumor cell — the founder cell [23] — that has acquired a *driver* mutation, which inhibits control on the proliferation of subsequent cancer cells. From that moment, the combination of unrestrained proliferation and a very hostile environment — as the immune system fights for survival, and so the tumor cells, under extreme selection pressure, have to disguise themselves to avoid being attacked, compete with each other, all while having to thrive while getting low levels of oxygen — produces the accumulation of highly elevated number of mutations, including structural variations. This model of tumor evolution is called the clonal model [23], since a clone is a population of cells carrying the same set of mutations. Understanding this clonal evolutionary history can help clinicians understand the cell heterogeneity in the tumors of various types of cancer and, more importantly, gives insights on how to devise therapeutic strategies [22, 30].

The rise of *next-generation sequencing* (NGS) technologies has led to the computational problem of tumor phylogeny inference from NGS bulk sequencing data [7, 20, 27, 11]. This idea is very cost-effective, since NGS data is cheap to obtain and very reliable. The procedure consists of extracting different samples of the tumor and aligning the NGS reads against the reference genome: this allows to determine the approximate fraction of reads of each sample that are affected by any given mutation. This fraction is taken as a proxy of the fraction of cells in each sample that are affected by that mutation. The main difficulties of this technique are the fact that a sample contains a mix of both healthy cells and cancer cells, while the cancer cells are an unknown mixture of different clones.

Since both the composition of the samples and the evolutionary history are unknown in this model, most of the approaches have needed some simplifying assumption, such as the *infinite sites assumption* (ISA) that postulates that each mutation is acquired exactly once, and never lost, in the entire tree. While this assumption has only limited biological validity [19, 2] it reduces greatly the space of all possible solutions, making feasible several approaches [7, 11], which are at least partially inspired by a classical linear time algorithm [10] for reconstructing the phylogeny from noise-free character data: this latter computational problem is called the *perfect phylogeny* problem.

The newer *single cell sequencing* (SCS) technology provides a much finer level of resolution: in fact we can determine whether or not a given cell *has* a mutation, therefore avoiding the notion of sample and the approximations implied by the use of samples. Still, SCS is expensive and plagued by high dropout, *i.e.*, missing values, and false negative rates.

Nowadays, we are witnessing a decrease in SCS costs, coupled with improvements in the dropout and false negative rates, stimulating the research on tools for tumor evolution inference from SCS data [16, 25, 4, 5, 31, 6]. We believe that this line of research is going to become even more important in the next few years, since currently available SCS data is associated with a very large solution space. Moreover, the high missing value and false negative rates allow a huge number of possible phylogenies with near optimal values of the objective function: this fact makes difficult to determine which methods actually produce better solutions and shows that the objective function is not able to fully capture the biological soundness of the phylogeny.

These advances in costs and quality of the data produced will result in larger, but more constrained, instances: the net effect is a considerable reduction in the number of likely solutions. Since most of the currently available methods do not scale well to large instances (usually their running time is quadratic with respect to the number of mutations), a preprocessing step can be useful to reduce the size of an instance: for example, SPhyR [6] uses k-means [21, 1] to such purpose. However, k-means is designed for continuous data — where means are usually based on a Euclidean

distance — while SCS data, specifying the presence (1) or absence (0) of a mutation in cell, or the fact that it is missing (2), can be thought of as categorical.

Clustering categorical data is an active field of research in data mining [18], where massive databases of categorical data are handled, *e.g.*, finding groups of members of an insurance policy who regularly travel overseas.

In this paper we analyze a variety of different clustering approaches on SCS data. Our comparison has three goals: (1) to assess which clustering methods are more precise on SCS data by comparing the actual clustering produced, (2) to evaluate the effect of a clustering step on the downstream phylogeny inference step, and (3) to assess the usefulness of a clustering step on real data, by showing its capability to reduce a instance that is too large for some current methods so that the clustered instance can be easily solved.

Methods

We now give an overview of the methods we consider for clustering mutations (columns) of single-cell sequencing (SCS) datasets. We chose clustering methods for this study based on: (a) the method clusters vector or matrix data into a smaller such set of vectors or matrix, as is the nature of SCS data; and (b) the method has an implementation available.

k-means

Given m objects $X = \{\overline{x}_1, \ldots, \overline{x}_m\}$ on n real values, *i.e.*, each $x_i \in \mathbb{R}^n$, and an integer k, the k-means algorithm [21, 1] finds the vector of k values, called *means*, $Q = \{\overline{q}_1, \ldots, \overline{q}_k\}$, and an $m \times k$ *partition matrix* [12] that minimize the following objective function

$$\sum_{l=1}^{k} \sum_{j=1}^{m} p_{jl} d(\overline{x}_j, \overline{q}_l), \tag{1}$$

subject to

$$\begin{aligned} \sum_{l=1}^{k} p_{jl} = 1, \quad & 1 \leq j \leq m, \\ p_{jl} \in \{0, 1\}, \quad & 1 \leq j \leq m, 1 \leq l \leq k. \end{aligned} \tag{2}$$

Here, d is a *distance measure* that is usually the Euclidean distance, *i.e.*,

$$d(\overline{x}_j, \overline{q}_l) = \sum_{i=1}^{n} (\overline{x}_{ji} - \overline{q}_{li})^2. \tag{3}$$

The k-means algorithm starts with some initial set Q^0 of k means, and an initial collection $\mathbb{C}^0$ of k disjoint subsets of X, and then iterates the operations

1. compute $d(\overline{x}_j, \overline{q}_l)$, where $\overline{q}_l$ is the mean of cluster $C_l^0 \in \mathbb{C}$, for each $\overline{x}_j, C_l^0$;
2. allocate $\overline{x}_j$ to the cluster $C_{l'}^1$ minimizing $d(\overline{x}_j, q_{l'})$; and
3. recompute the new set of means Q^1 according to the new clusters $\mathbb{C}^1$, minimizing Equation 1;

until convergence, *i.e.*, $Q^{t+1} = Q^t$. This algorithm runs in time $O(tkmn)$, where t is the number of iterations of the above set of operations. It has been shown to converge in [21].

We used the implementation of k-means clustering available at `https://scikit-learn.org/stable/modules/clustering.html#k-means`

Affinity propagation

The affinity propagation algorithm [9] uses as input a set of similarities between data points. Those similarities are clustered by choosing a point as a representative for each class. Such points gradually emerge iteratively using a message-passing procedure, where each point exchanges messages to all other points.

In particular, there exist two types of messages: (1) *responsibility* $r(i, k)$ is sent from point i to the candidate representative point k reflecting the cumulative evidence of how well-suited k is to serve as representative of i; and (2) *availability* $a(i, k)$ sent from the candidate representative k to i reflecting the cumulative evidence of how well-suited k should be chosen by i to be its representative. Both variables take into account other potential candidates.

Such messages are exchanged iteratively, each time refining $r(\cdot, \cdot)$ and $a(\cdot, \cdot)$, until a stop condition is fulfilled. At any point these variables can be combined to identify the representatives. For each point i the value of k that maximizes $r(i, k) + a(i, k)$ identifies the representative k of i, where k and i can be the same point.

We used the implementation of affinity propagation clustering available at `https://scikit-learn.org/stable/modules/clustering.html#affinity-propagation`

Hierarchical agglomerative clustering

The hierarchical agglomerative clustering method [17] produces, in an hierarchical fashion, groups of disjoint subsets of a given set, each maximizing an internal similarity score. The procedure is executed in a bottom-up fashion in which initially each point is in a subset by itself. At each step, two sets are merged together, so that the union maximizes a given criterion. The procedure is then repeated until only one group remains, thus having the complete hierarchical structure of the clustering.

In our experiments we used the Manhattan metric, *i.e.*, the sum of the horizontal and vertical distance of two points, to compute the similarity scores, as this metric is more suited for matrix containing categorical data, as opposed to the Euclidean distance more suited for a continuous-valued matrix. The similarity of two sets of observations is computed using the average of the distances between elements in the respective sets. Consequently, as expected, the hierarchical agglomerative clustering, when using the Manhattan metric performed better (see Results) than the Euclidean metric, and so we only included the former in the comparison.

We used the implementation of hierarchical clustering available at `https://scikit-learn.org/stable/modules/clustering.html#hierarchical-clustering`

BIRCH clustering

The BIRCH clustering procedure [32] takes as input a set of points as well as the desired number of clusters, and operates in four steps. (1) In the first step it computes the clustering feature (CF) tree while computing measures using a predefined metric. (2) The second optional phase tries to build a smaller CF tree while removing outliers and regrouping crowded subclusters into larger ones. (3) In the third step, a hierarchical clustering algorithm, usually an adaptation of the agglomerative clustering (briefly described in the previous subsection), is used to cluster all the leaves of the CF

tree. During this phase, the centroids of each cluster are computed. (4) The centroids are then used for the final fourth step to further refine the clusters, since minor and localized misplacements could occur during the previous step. This last step can also be used to label the points with the cluster they are placed in — we used this feature to obtain the groups of mutations in our experiments.

We used the implementation of BIRCH clustering available at `https://scikit-learn.org/stable/modules/clustering.html#birch`

Spectral clustering

The spectral clustering algorithm [26] — see [29] for a gentle and comprehensive treatment of the topic — first performs dimensionality reduction on the data, and then clusters, using a standard clustering technique such as k-means, the data in this reduced dimension. In order to reduce the dimensionality, it first constructs a similarity *graph* from the initial matrix of similarities of the input set of data points — usually a sparse representation of this similarity matrix, *e.g.*, the k-nearest neighbor graph [29]. It then takes the Laplacian matrix [3] on this graph, and a subset of the (relevant) eigenvectors (spectrum) of this Laplacian matrix — itself a matrix, is the reduced-dimension data. It is then this matrix which is then passed to the clustering technique.

We used the implementation of spectral clustering available at `https://scikit-learn.org/stable/modules/clustering.html#spectral-clustering`

k-modes

Given m objects X on n *categorical* attributes $A = \{\overline{a}_1, \ldots, \overline{a}_n\}$, the k-modes algorithm [13, 14] finds k *modes* Q — that is, for each $\overline{q}_l \in Q$, $\overline{q}_{li}$ is one of the $\overline{a}_i$ possible categories at position i. Note that every $\overline{a}_i = \{0, 1, 2\}$ when considering single cell sequencing data for inferring tumor evolutions. Instead of the Euclidean distance between object $\overline{x}_j$ and (mode) $\overline{q}_l$ we compute a *dissimilarity* measure d_1:

$$d_1(\overline{x}_j, \overline{q}_l) = \sum_{i=1}^{n} \delta(\overline{x}_{ji}, \overline{q}_{li}). \tag{4}$$

according to some dissimilarity δ. A common dissimilarity is the simple *matching dissimilarity* between categories:

$$\delta_M(\overline{x}_{ji}, \overline{q}_{li}) = \begin{cases} 0 & \text{if } \overline{x}_{ji} = \overline{q}_{li} \\ 1 & \text{o.w. } (\overline{x}_{ji} \neq \overline{q}_{li}) \end{cases} . \tag{5}$$

Now, the *mode* of cluster $C_l \subseteq X$ is a vector $\overline{q}_l$ which minimizes

$$D(C_l, \overline{q}_l) = \sum_{\overline{x}_j \in C_l} d_1(\overline{x}_j, \overline{q}_l). \tag{6}$$

Note that $\overline{q}_l$ is not necessarily an element of C_l. The k-modes algorithm then starts with some initial set Q^0 of k modes, and an initial collection $\mathbb{C}^0$ of k disjoint subsets of X, and then iterates operations 1–3 as in the k-means algorithm, but using *dissimilarity* instead of distance, and computing *modes* instead of means, that minimize instead the objective function

$$\sum_{l=1}^{k} \sum_{j=1}^{m} \sum_{i=1}^{n} p_{jl} \delta(x_{ji}, q_{li}), \tag{7}$$

subject again to the constraints in Equation 2, by finding modes according to the following theorem, where c_{ri} is the r-th category of attribute $\overline{a}_i$ and $f(\overline{a}_i = c_{ri}|X)$ is its relative frequency in X:

Theorem 1 ([14]) *Eq. 6 is minimized iff* $f(\overline{a}_i = q_{li}|X) \geq f(\overline{a}_i = c_{ri}|X)$ *for* $q_{li} \neq c_{ri}\ \forall\ i \in [n]$.

In other words, Eq. 6 is minimized by selecting the mode category for each attribute. Note that this theorem implies that the mode of X is not necessarily unique.

Since the solution depends on the initial set Q^0 of k modes, we consider two procedures for initializing Q^0. The first one is quite simple: a random selection of k objects from the set X of objects as the initial k modes — which we refer to as *random initialization*. The second one, devised in [14], is a more complicated procedure, based on the frequencies $f(\overline{a}_i = c_{ri}|X)$ of all categories, which we refer to as *Huang initialization*. We focus on this second procedure, since it achieves the best results:

1. order the categories of each attribute $\overline{a}_i$ in descending order of frequency, *i.e.* $f(c_{r_1 i}) \geq f(c_{r_2 i}) \geq f(c_{r_3 i})$, etc.;
2. assign uniformly at random the most frequent categories to the initial k modes;
3. for each $\overline{q}_l$, select the $\overline{x}_j \in X$ most similar to $\overline{q}_l$ and make this the mode $\overline{q}_l^0 \in Q^0$, such that $\overline{q}_l^0 \neq \overline{q}_{l'}^0$ for $l \neq l'$.

This final step is to avoid empty clusters in $\mathbb{C}^0$. This initial selection procedure is to have a diverse set of initial modes Q^0, which can lead to better clustering results — see more details in [14].

The *Python Package Index* (pypi) has a package for computing a clustering based on k-modes `https://pypi.org/project/kmodes/`, including an implementation of Huang (and random) initialization. This is what we used in our tests.

Celluloid

We have designed a new dissimilarity, based on the notion of *conflict*, to be used within the k-modes framework. Since a 2 represents *missing* data in our context, we are only interested in cases where a 2 is not involved, obtaining what we call the *conflict dissimilarity*, formally described in Eq. 8 and used as dissimilarity in k-modes.

$$\delta_C(\overline{x}_{ji}, \overline{q}_{li}) = \begin{cases} 0 & \text{if } \overline{x}_{ji} = 2 \text{ or } \overline{q}_{li} = 2 \\ \delta_M(\overline{x}_{ji}, \overline{q}_{li}) & \text{o.w.} \end{cases} . \tag{8}$$

Our approach, *celluloid: clustering single cell sequencing data around centroids*, is the combination of k-modes with Huang initialization and our conflict dissimilarity, and is available at `https://github.com/AlgoLab/celluloid/` under an MIT license.

Results

To evaluate the accuracy of the clustering methods, we designed a two-fold experimentation on synthetic datasets: we first measure the quality of the clusters found by the methods, and later we evaluate how such clusters impact the quality of the phylogenies returned by a cancer progression inference method.

The simulated data are generated as follows. First we simulate a random tree topology on s nodes, each representing a tumor clone, by first creating a root (the germline) and then iteratively

attaching the $s-1$ remaining nodes uniformly at random to any other node in the tree. The nodes are then randomly labeled with m mutations — meaning that each mutation is acquired at the node that it labels. Then, a total of n cells are associated to the nodes, uniformly at random. A binary matrix M is then extracted from these cells giving rise to a genotype profile for each cell (a row in M), which is the presence (1) or absence (0) of each mutation (column in M) in the cell, given the presence or absence of the mutations on the path in the tree from the root to this cell. The binary matrix is then perturbed according to the false negative, false positive and missing value rates, to simulate a real-case scenario. Each of the s nodes is therefore considered as a natural (true) cluster of the simulated dataset.

For each experiment, we consider 100, 200 and 300 cells, respectively Experiment 1, 2, and 3. For each such value we generated 50 simulated datasets, where we fixed the number s of clones to 20 and the number m of mutations to 1000. While this number of mutations is at the high end in terms of currently available real cases, it will be a typical size in the near future — some such cases already exist today (see Application on read data).

We performed clustering on the datasets of our experiments to obtain instances with a reduced number of columns (mutations), which can in turn be given as input to such a method above. The clustering methods we used were all the ones described in the Methods section, *i.e.*, k-means, affinity propagation, hierarchical agglomerative, BIRCH, spectral clustering, k-modes and celluloid. Since the cancer inference methods tend to scale quadratically with the number of mutations, we reduced with all clustering methods the size of mutations to a total of 100, which is a reasonable number of mutations in the currently available literature.

Evaluating a clustering

To evaluate to the clusters obtained, we used standard precision and recall measures, adapted to the particular goal, as follows.

Precision: measures how well mutations are clustered together. For each pair of mutations appearing in the same clone in the simulated tree, we check if they are in the same cluster, resulting in a true positive (TP). For each pair of mutations clustered together that are not in the same clone, we encounter a false positive (FP). The value of the precision is then calculated with the standard formula: $\frac{TP}{TP+FP}$.

Recall: measures how well mutations are separated. For each pair of mutations in the same clone, we now also check if they are not in the same cluster, resulting in a false negative (FN). The recall is then calculated as: $\frac{TP}{TP+FN}$.

It is important to highlight that we are mostly interested in obtaining a high *precision* since, while cancer phylogeny inference algorithms can later cluster together mutations by assigning them to same subtree or the same path, they cannot separate mutations that have been erroneously clustered together.

In Figures 1, 2 and 3 — representing the respective experiments 1, 2 and 3 — a common trend is evident: indeed standard clustering methods (k-means, affinity propagation, hierarchical agglomerative, BIRCH and spectral clustering) perform much more poorly than celluloid and k-modes, presenting a gap from all the other methods in terms of both precision and recall. On the other hand, celluloid and k-modes differ slightly in terms of precision.

It is interesting to notice that the precision of the conflict dissimilarity rapidly increases when the amount of cells increase, thus being well-suited for future increases on the size of SCS datasets.

We have also evaluated the quality of the clusters with some standard clustering metrics such as Adjusted Rand index [15], Fowlkes-Mallows index [8], Completeness score [24] and V-Measure [24]. In this case, we measure how much the computed clustering is similar to the ground truth.

In Figures 4, 5 and 6 — representing again the respective experiments 1, 2 and 3 — we see the similar trend of celluloid and k-modes performing much better in terms of precision than all the other methods, presenting this same gap from all the other methods, while differing slightly from each other in terms of precision.

We decided also to compute Adjusted Rand index, Fowlkes-Mallows index, Completeness score and V-Measure for all pairs of techniques to allow observation of the similarity, according to the measures, between different algorithms. The heatmaps in figures 7, 8 and 9 — representing respectively experiments 1, 2 and 3 — show the average value of the scores of each simulated data for each pair of clustering method.

The first column of all the heatmaps represents the comparison with the ground truth, where in all the cases celluloid, thus more resembling the ground truth. Furthermore, we compare all pairs of clustering algorithms, to detect similarities between them; as expected k-modes and celluloid are very similar to each other, while the other methods tend to be quite dissimilar to each other.

Assessing the impact of a clustering

To better understand the impact of the clustering on the actual cancer progression inference, we considered SCITE [16] and SPhyR [6], two published, publicly available inference tools tailored to SCS data, and have been shown to scale instances of the size we consider in our study — the latter tool has a built-in clustering preprocessing step which uses k-means. We executed SCITE [16] and SPhyR [6] on both the obtained clusters and on the unclustered datasets, to understand the effect of the clustering on the tools. We performed the inference with all the clustering methods used as preprocessing step.

For assessing the accuracy of the methods we used the measures defined in [5, 4]:

Ancestor-Descendant accuracy: for each pair of mutations in an ancestor-descendant relationship in the ground truth, we check whether the relationship is conserved in the inferred tree (TP) or whether it is not (FN). For each pair of mutations in an ancestor-descendant relationship in the inferred tree, we also check if such relationship does not exist in the ground truth tree (FP).

Different lineages accuracy: similarly to the previous measure, we check whether mutations in different branches are correctly inferred or if any pair of mutation is erroneously inferred in different branches.

Figures 10, 11 and 12 show very interesting results. Regarding SCITE's results, as expected, we observe a severe drop in performance when used in combination with k-means, affinity propagation, hierarchical agglomerative, BIRCH and spectral clustering methods. This fact is supported by the gap in the precision of the methods — a low precision indeed leads to a low accuracy in the tree reconstruction. The trend is still present in the different lineages accuracy, but to a lower extent — this is because, as previously discussed, a cancer inference method can separate *clusters* of mutations, but when a cluster is computed it is not possible to separate mutations *within* this cluster.

On the other hand, the results obtained by SCITE when celluloid is used are much better, in particular using celluloid with our conflict measure as preprocessing allows SCITE to score higher in both Ancestor-Descendant and Different lineages accuracies than it is able to using unclustered

data as input. For this inference tool, using celluloid as a preprocessing step actually helps SCITE to achieve a better score than without it; experiment 3, being the largest of the three, shows the biggest improvements. SCITE on unclustered data scores an average of 0.755 and 0.954 for Ancestor-Descendant and Different lineages respectively, against an average of 0.936 and 0.991 after clustering the datasets using celluloid with Huang initialization and conflict as measure. Moreover, celluloid allows SCITE to achieve a 20x speedup in runtime, on average.

Unlike the previous tool, SPhyR shows very interesting results since the method itself performs a k-means clustering on both mutations and cells given as input. For this reason clustering already clustered data can be a hit-or-miss case, as seen in Figures 10, 11 and 12. In almost all cases, preprocessing the data causes SPhyR to obtain extremely low values in the Ancestor-Descendant measure, while achieving a very good Different lineages score, with the exception of Spectral clustering in all experiments and celluloid with conflict similarity in experiment 1. On the other hand, the Agglomerative clustering seems to have no impact or to slightly improve the tool, this could be a consequence of the high recall and low precision achieved by the clustering algorithm — indeed it is possible that separating well clusters but not merging them very much leads the clustering step of SPhyR to achieve a better result.

Application on real data

Finally, we run our entire pipeline (clustering + inference method) on a oligodendroglioma IDH-mutated tumor [28] consisting of 1842 mutations over 926 cells. Such computation was performed using SCITE which on unclustered data takes as long as 68 hours, while adding a preprocessing step with celluloid, we were able to compute the tree within 1 hour; most likely providing a better phylogeny than SCITE alone as seen in the simulated experimentations.

Discussion

We have shown how to compare various clustering methods on single cell sequencing data, with three distinct experiments. More precisely, we describe two experiments on synthetic data: the first experiment measures the quality of the clusters computed by the methods, while the second experiment considers the effect of the clustering on the quality of trees obtained downstream from a phylogeny inference tool. Finally, we have described an experiment on real data that measures the usefulness of the clustering procedure, by computing the improvement in the runtime for a large instance.

We have made available at `https://github.com/AlgoLab/celluloid/` the entire pipeline that generates the simulated data, run the clustering tools on those data, and computes the plots and tables used in the analysis. Our comparison is reproducible and can be easily extended by modifying a Snakemake file.

One of the main conclusions that we can draw from our comparison is that k-means is not an adequate choice to cluster single cell data for inferring tumor phylogenies. The second main finding of our paper is that a suitable clustering step, such as celluloid, not only decreases the runtime of the phylogeny inference methods, but can also improve the quality of the inferred phylogenies, as we have shown for SCITE.

Declarations

Availability of data and material. Instructions on how to use and replicate all experiments can be found at `https://github.com/AlgoLab/celluloid/`

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

Funding. We acknowledge the support of 2018-ATE-0575 and 2017-ATE-0534 grants.

Author's contributions. SC, MDP and GDV designed the benchmarking methodology. SC and MDP chose the clustering methods. SC did the experiments. SC, MDP, PB and GDV wrote and revised the manuscript. All authors read and approved the final version of the manuscript.

Acknowledgments. We thank Iman Hajirasouliha and Dana Silverbush for several illuminating discussions.

References

[1] M.R. Anderberg. *Cluster Analysis for Applications*. Academic Press, 1973.

[2] David Brown, Dominiek Smeets, Borbála Székely, et al. Phylogenetic analysis of metastatic progression in breast cancer using somatic mutations and copy number aberrations. *Nature Communications*, 8:14944, 2017.

[3] F. Chung. Spectral graph theory. In *Conference Board of the Mathematical Sciences Regional Conference Series in Mathematics*, volume 92.

[4] Simone Ciccolella, Mauricio Soto Gomez, Murray Patterson, Gianluca Della Vedova, Iman Hajirasouliha, and Paola Bonizzoni. Inferring cancer progression from single cell sequencing while allowing loss of mutations. *bioRxiv*, 268243, 2018.

[5] Simone Ciccolella, Mauricio Soto, Murray Patterson, Gianluca Della Vedova, Iman Hajirasouliha, and Paola Bonizzoni. gpps: An ILP-based approach for inferring cancer progression with mutation losses from single cell data. In *IEEE Computational Advances in Bio and medical Sciences, 8th International Conference (ICCABS)*, 2018. to appear.

[6] Mohammed El-Kebir. SPhyR: tumor phylogeny estimation from single-cell sequencing data under loss and error. *Bioinformatics*, 34(17):i671–i679, 2018.

[7] Mohammed El-Kebir, Layla Oesper, Hannah Acheson-Field, and Benjamin J. Raphael. Reconstruction of clonal trees and tumor composition from multi-sample sequencing data. *Bioinformatics*, 31(12):i62–i70, 2015.

[8] E. B. Fowlkes and C. L. Mallows. A method for comparing two hierarchical clusterings. *Journal of the American Statistical Association*, 78(383):553–569, 1983.

[9] Brendan J. Frey and Delbert Dueck. Clustering by passing messages between data points. *Science*, 315(5814):972–976, 2007.

[10] Dan Gusfield. Efficient algorithms for inferring evolutionary trees. *Networks*, 21(1):19–28, 1991.

[11] Iman Hajirasouliha and Benjamin J. Raphael. *Reconstructing Mutational History in Multiply Sampled Tumors Using Perfect Phylogeny Mixtures*, pages 354–367. Lecture Notes in Computer Science. Springer Nature, 2014.

[12] D.J. Hand. *Discrimination and Classification.* John Wiley & Sons, 1981.

[13] Z. Huang. A fast clustering algorithm to cluster very large categorical data sets in data mining. In *the SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery*, pages 1–8, 1997.

[14] Z. Huang. Extensions to the k-modes algorithm for clustering large data sets with categorical values. *Data Mining and Knowledge Discovery*, 2(3):283–304, 1998.

[15] Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of Classification*, 2(1):193–218, Dec 1985.

[16] Katharina Jahn, Jack Kuipers, and Niko Beerenwinkel. Tree inference for single-cell data. *Genome Biology*, 17(1):86, 2016.

[17] Joe H. Ward Jr. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58(301):236–244, 1963.

[18] W. Klosgen and J.M. Zytkow. Knowledge discovery in databases terminology. In *Advances in Knowledge Discovery and Data Mining*, pages 573–592. AAAI Press/The MIT Press, 1996.

[19] Jack Kuipers, Katharina Jahn, Benjamin J. Raphael, and Niko Beerenwinkel. Single-cell sequencing data reveal widespread recurrence and loss of mutational hits in the life histories of tumors. *Genome Research*, 27:1885–1894, 2017.

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

[21] J.B. McQueen. Some methods for classification and analysis of multivariate observations. In *the 5th Berkely Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967.

[22] A. Sorana Morrissy, Livia Garzia, David J. H. Shih, et al. Divergent clonal selection dominates medulloblastoma at recurrence. *Nature*, 529(7586):351–357, 2015.

[23] Peter C. Nowell. The clonal evolution of tumor cell populations. *Science*, 194(4260):23–28, 1976.

[24] Andrew Rosenberg and Julia Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. In *EMNLP-CoNLL*, 2007.

[25] Edith M. Ross and Florian Markowetz. Onconem: inferring tumor evolution from single-cell sequencing data. *Genome Biology*, 17(1):69, 2016.

[26] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. Technical report, 2000.

[27] F. Strino, F. Parisi, M. Micsinai, and Y. Kluger. TrAp: a tree approach for fingerprinting subclonal tumor composition. *Nucleic Acids Research*, 41(17):e165, 2013.

[28] Andrew S. Venteicher, Itay Tirosh, Christine Hebert, Keren Yizhak, Cyril Neftel, Mariella G. Filbin, Volker Hovestadt, Leah E. Escalante, McKenzie L. Shaw, Christopher Rodman, Shawn M. Gillespie, Danielle Dionne, Christina C. Luo, Hiranmayi Ravichandran, Ravindra Mylvaganam, Christopher Mount, Maristela L. Onozato, Brian V. Nahed, Hiroaki Wakimoto, William T. Curry, A. John Iafrate, Miguel N. Rivera, Matthew P. Frosch, Todd R. Golub, Priscilla K. Brastianos, Gad Getz, Anoop P. Patel, Michelle Monje, Daniel P. Cahill, Orit Rozenblatt-Rosen, David N. Louis, Bradley E. Bernstein, Aviv Regev, and Mario L. Suvà. Decoupling genetics, lineages, and microenvironment in idh-mutant gliomas by single-cell rna-seq. *Science*, 355(6332), 2017.

[29] Ulrike von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4), 2007.

[30] Jiguang Wang, Emanuela Cazzato, Erik Ladewig, et al. Clonal evolution of glioblastoma under therapy. *Nature Genetics*, 48(7):768–776, 2016.

[31] Hamim Zafar, Anthony Tzen, Nicholas Navin, Ken Chen, and Luay Nakhleh. SiFit: inferring tumor trees from single-cell sequencing data under finite-sites models. *Genome Biology*, 18(1):178, 2017.

[32] Tian Zhang, Raghu Ramakrishnan, and Miron Livny. Birch: An efficient data clustering method for very large databases. *SIGMOD Rec.*, 25(2):103–114, June 1996.

Figures

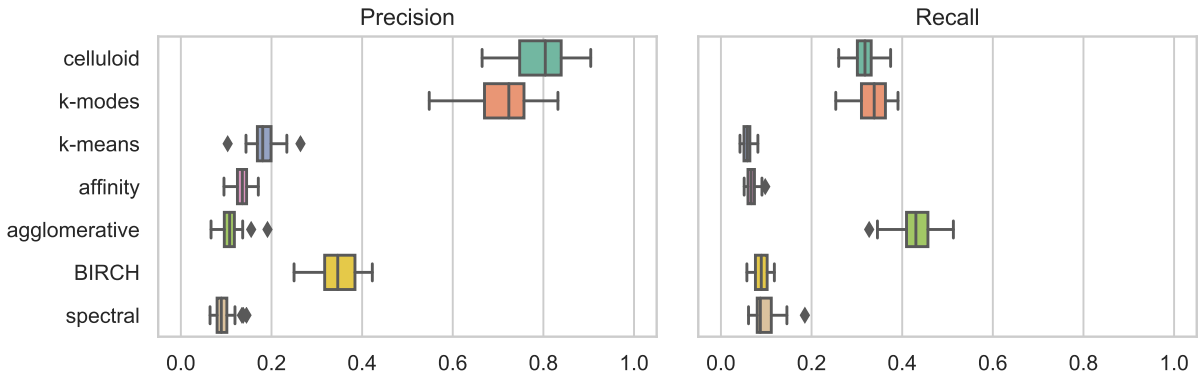

Figure 1: Precision and recall results for experiment 1, generated with a total of 1000 mutations, 100 cells and a clustering size of $k = 100$. The plots include results for celluloid, k-means, k-modes, affinity, agglomerative, BIRCH and spectral clustering.

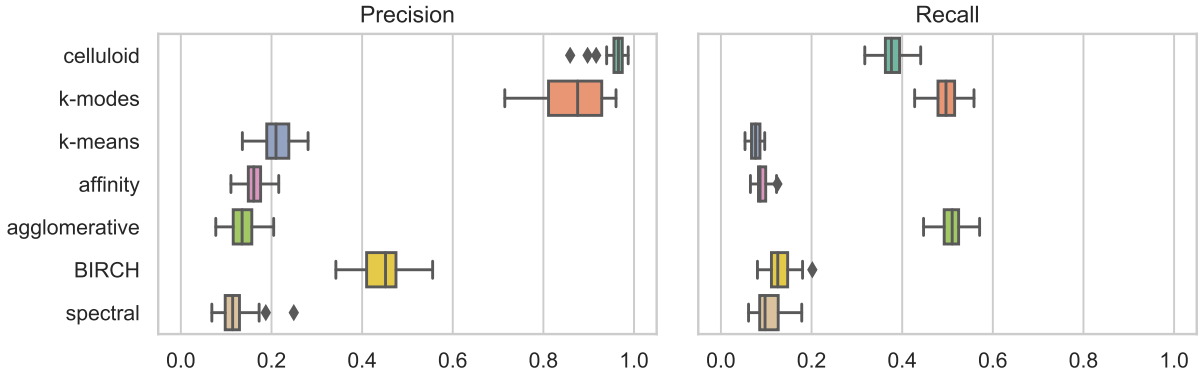

Figure 2: Precision and recall results for experiment 2, generated with a total of 1000 mutations, 200 cells and a clustering size of $k = 100$. The plots include results for celluloid, k-means, k-modes, affinity, agglomerative, BIRCH and spectral clustering.

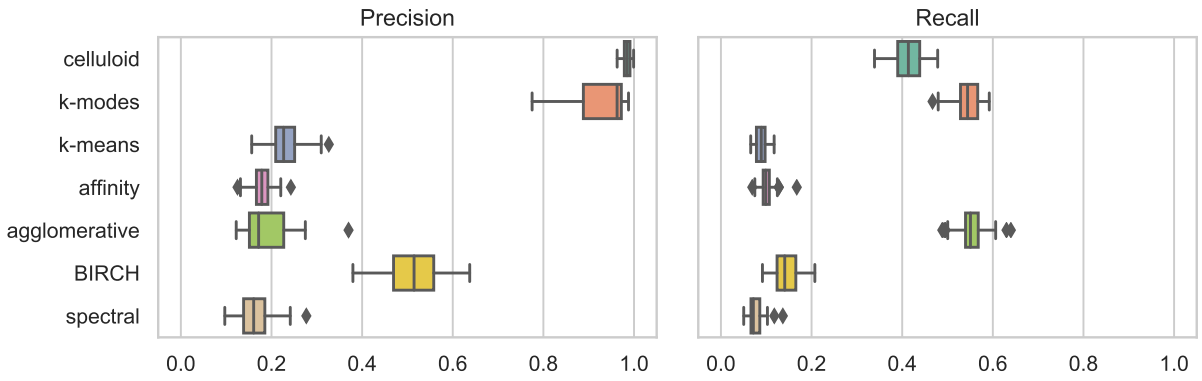

Figure 3: Precision and recall results for experiment 3, generated with a total of 1000 mutations, 300 cells and a clustering size of $k = 100$. The plots include results for celluloid, k-means, k-modes, affinity, agglomerative, BIRCH and spectral clustering.

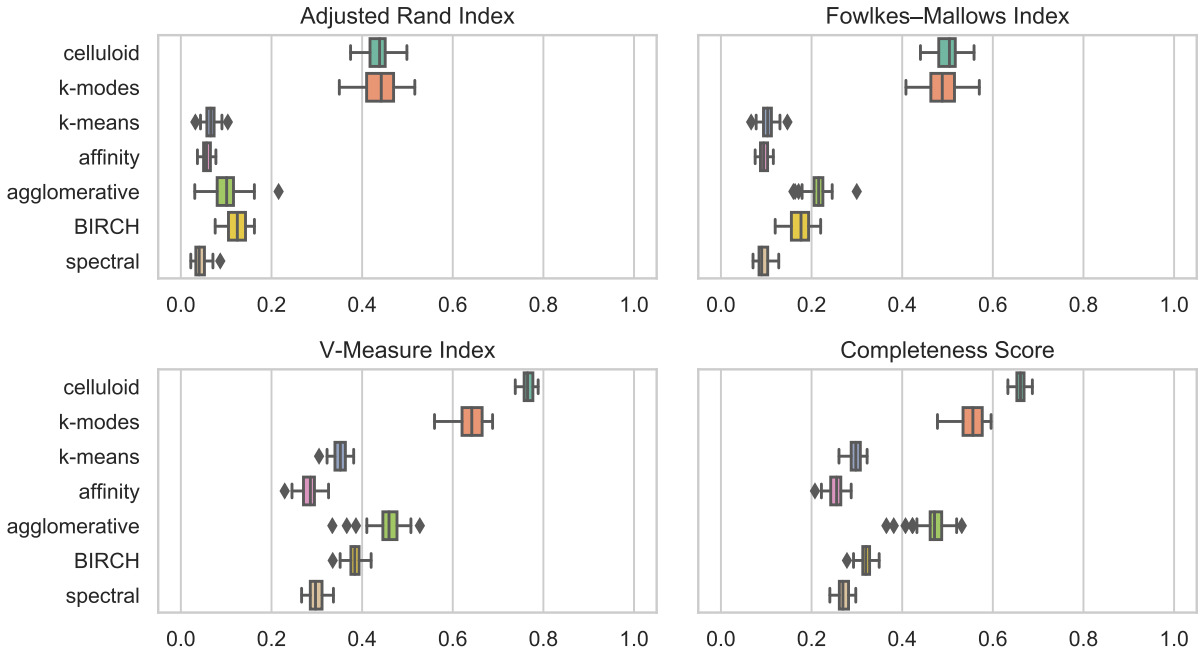

Figure 4: Adjusted Rand index, Fowlkes-Mallows index, Completeness score and V-Measure between all clustering methods and the ground truth for experiment 1, generated with a total of 1000 mutations, 100 cells and a clustering size of $k = 100$. The plots include results for celluloid, k-means, k-modes, affinity, agglomerative, BIRCH and spectral clustering.

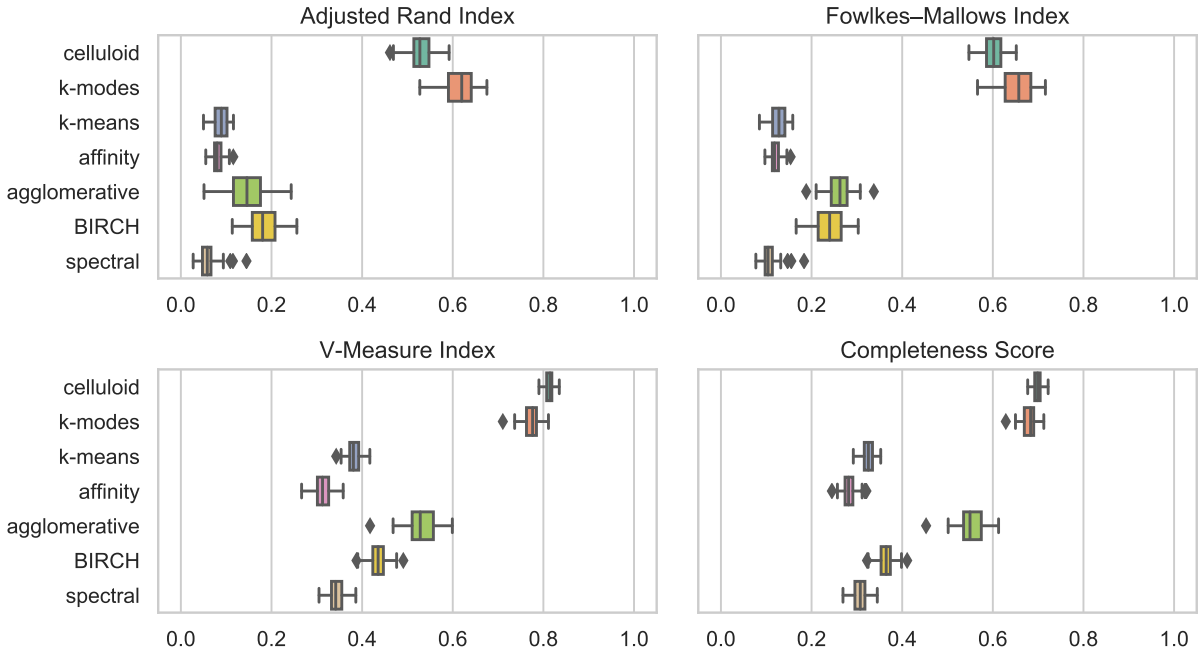

Figure 5: Adjusted Rand index, Fowlkes-Mallows index, Completeness score and V-Measure between all clustering methods and the ground truth for experiment 2, generated with a total of 1000 mutations, 200 cells and a clustering size of $k = 100$. The plots include results for celluloid, k-means, k-modes, affinity, agglomerative, BIRCH and spectral clustering.

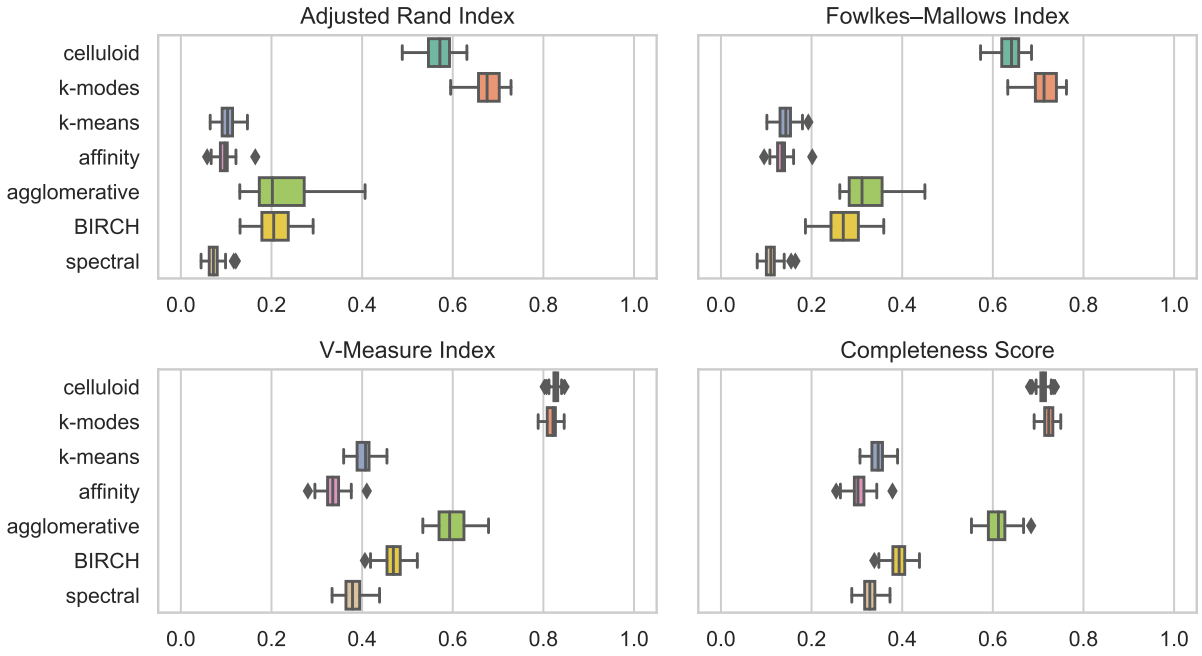

Figure 6: Adjusted Rand index, Fowlkes-Mallows index, Completeness score and V-Measure between all clustering methods and the ground truth for experiment 3, generated with a total of 1000 mutations, 300 cells and a clustering size of $k = 100$. The plots include results for celluloid, k-means, k-modes, affinity, agglomerative, BIRCH and spectral clustering.

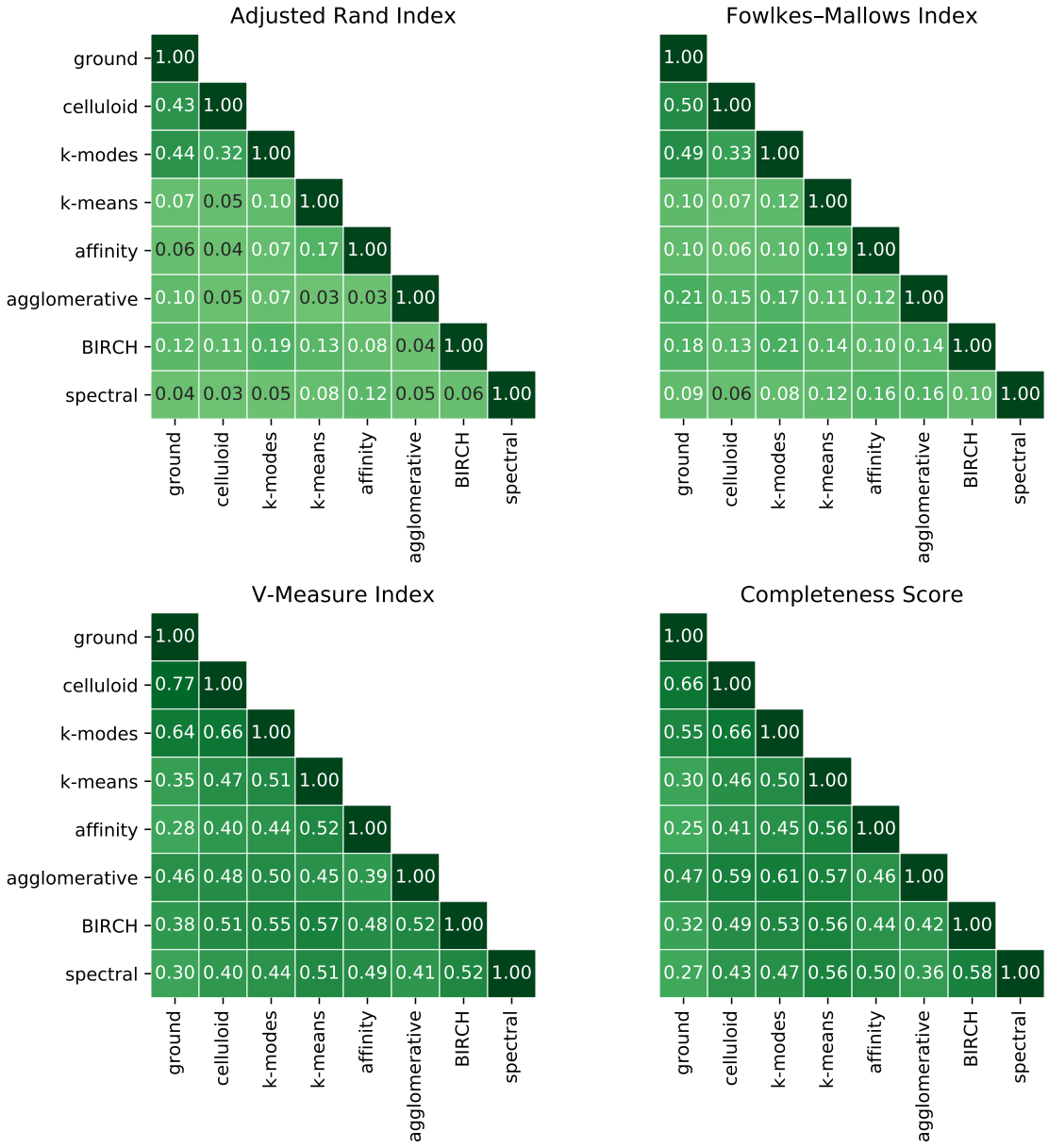

Figure 7: Adjusted Rand index, Fowlkes-Mallows index, Completeness score and V-Measure between all pairs of clustering methods for experiment 1, generated with a total of 1000 mutations, 100 cells and a clustering size of $k = 100$. The plots include results for celluloid, k-means, k-modes, affinity, agglomerative, BIRCH and spectral clustering. Each cell is the average the scores obtained for each simulated instance.

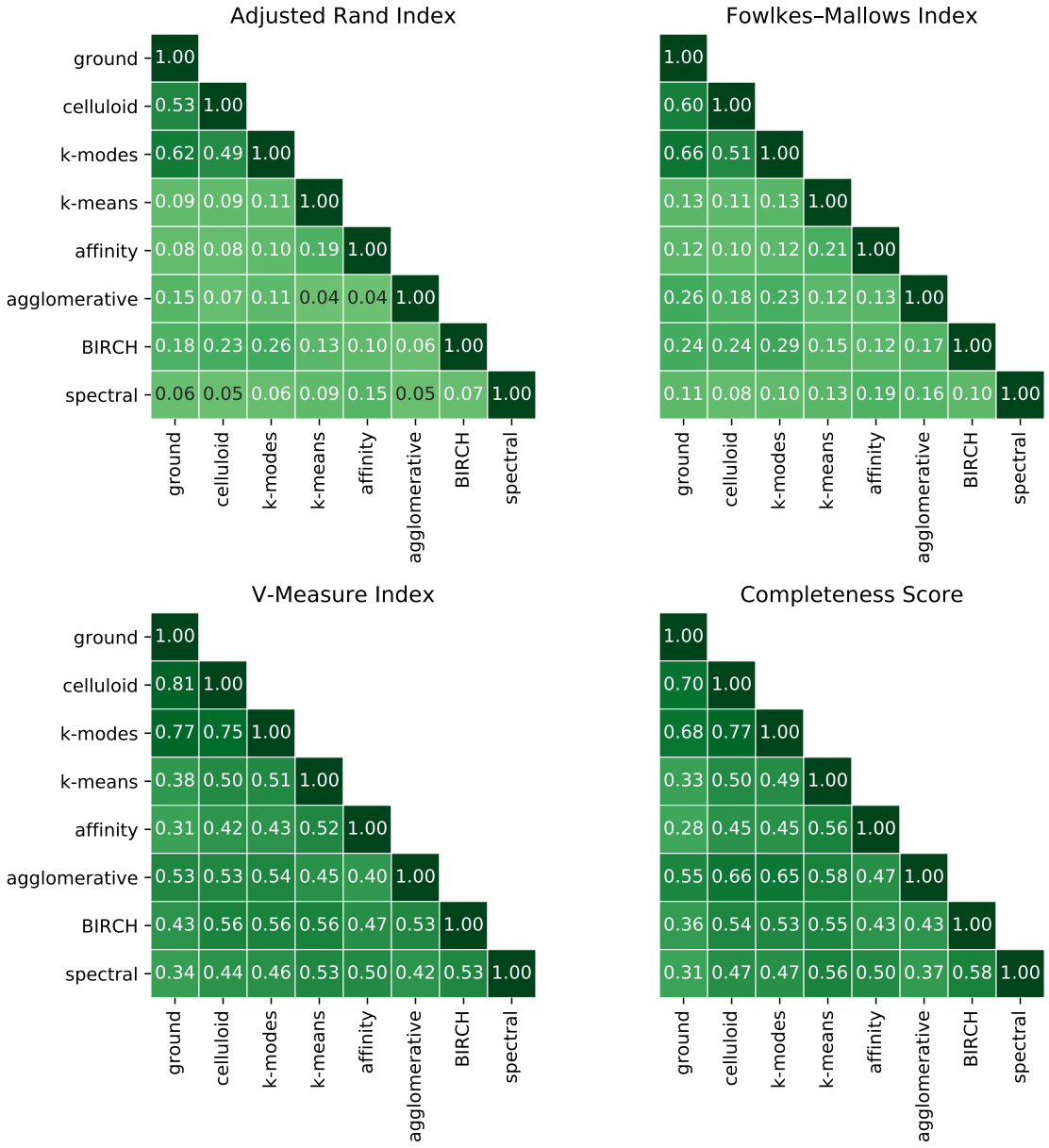

Figure 8: Adjusted Rand index, Fowlkes-Mallows index, Completeness score and V-Measure between all pairs of clustering methods for experiment 2, generated with a total of 1000 mutations, 200 cells and a clustering size of $k = 100$. The plots include results for celluloid, k-means, k-modes, affinity, agglomerative, BIRCH and spectral clustering. Each cell is the average the scores obtained for each simulated instance.

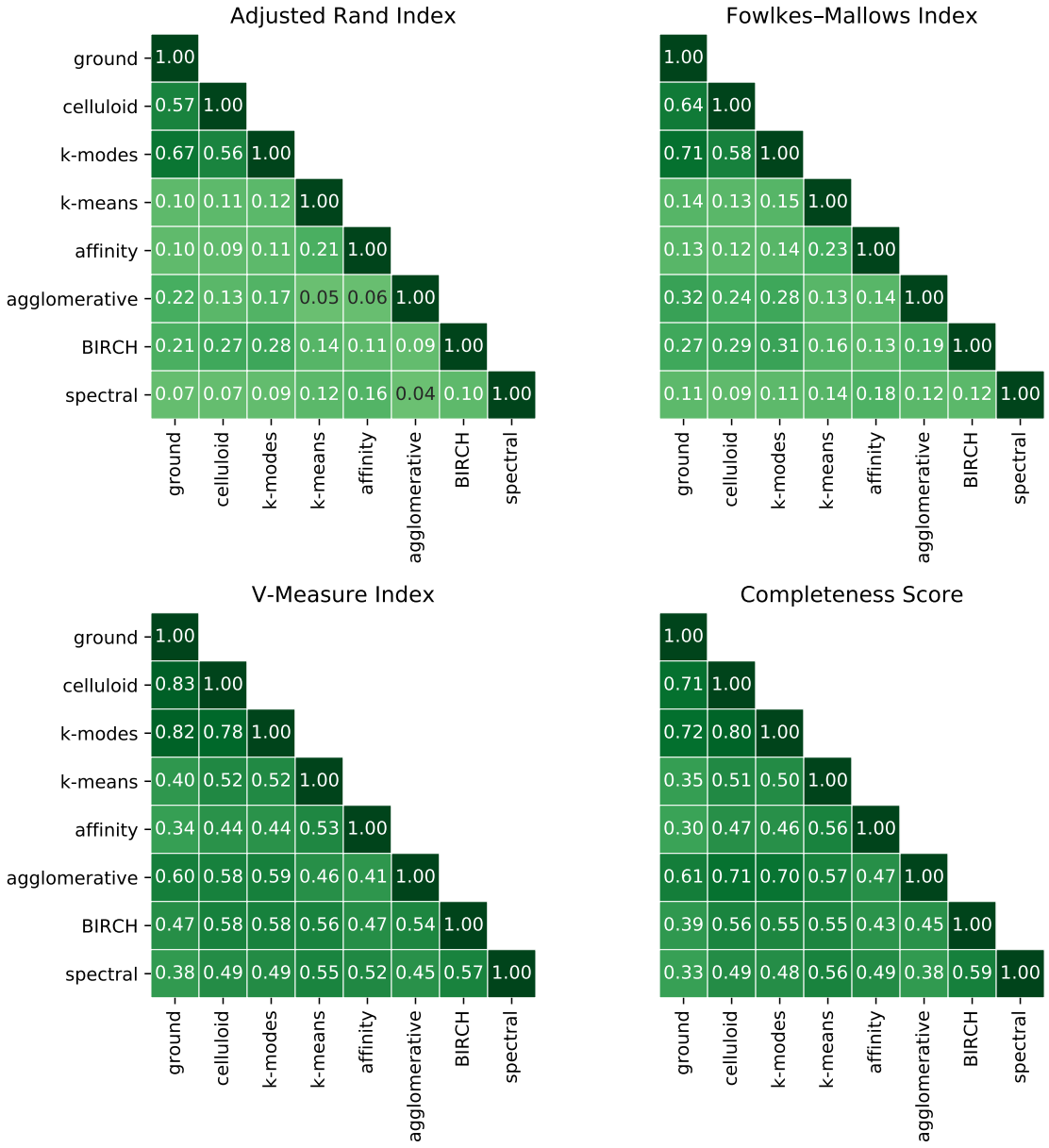

Figure 9: Adjusted Rand index, Fowlkes-Mallows index, Completeness score and V-Measure between all pairs of clustering methods for experiment 3, generated with a total of 1000 mutations, 300 cells and a clustering size of $k = 100$. The plots include results for celluloid, k-means, k-modes, affinity, agglomerative, BIRCH and spectral clustering. Each cell is the average the scores obtained for each simulated instance.

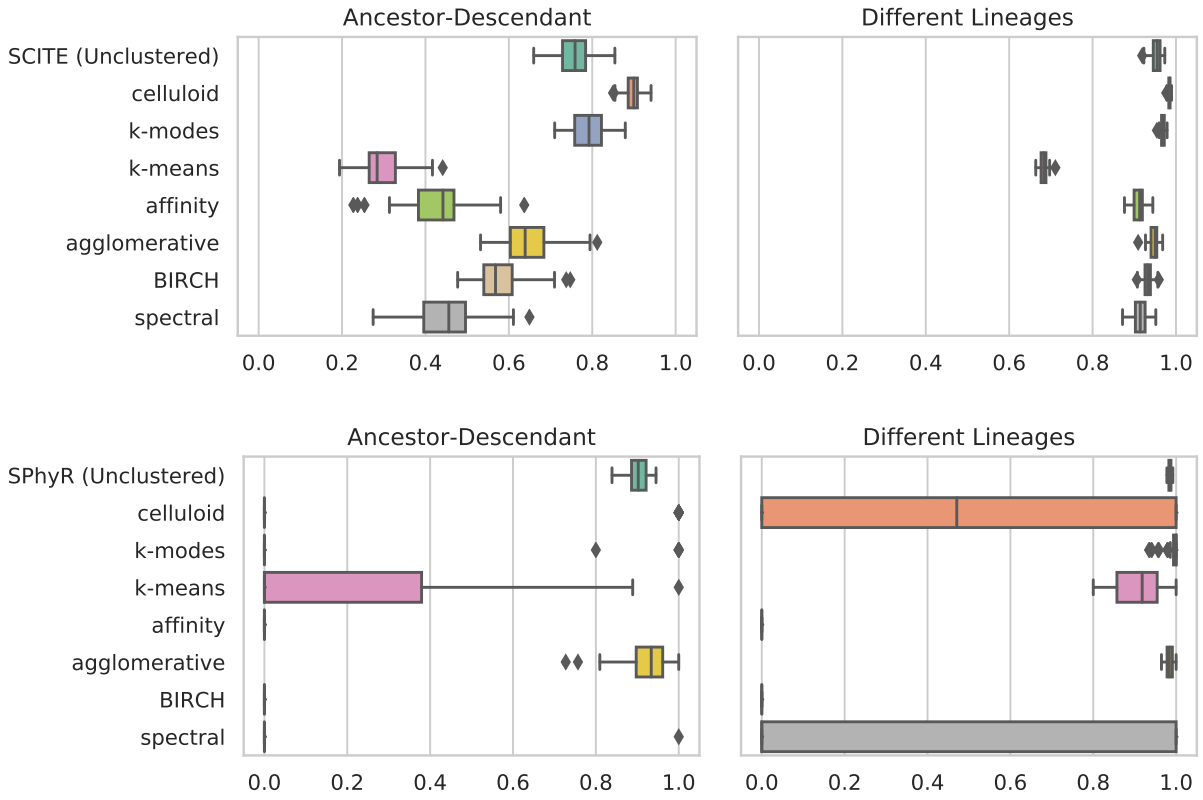

Figure 10: Ancestor-Descendant and Different lineages accuracy measures for experiment 1. Cancer phylogenies are inferred by SCITE (Top) and SPhyR (Bottom) using as input both the unclustered data as well as the clusters obtained by celluloid, k-means, k-modes, Affinity, Agglomerative, BIRCH and Spectral clustering.

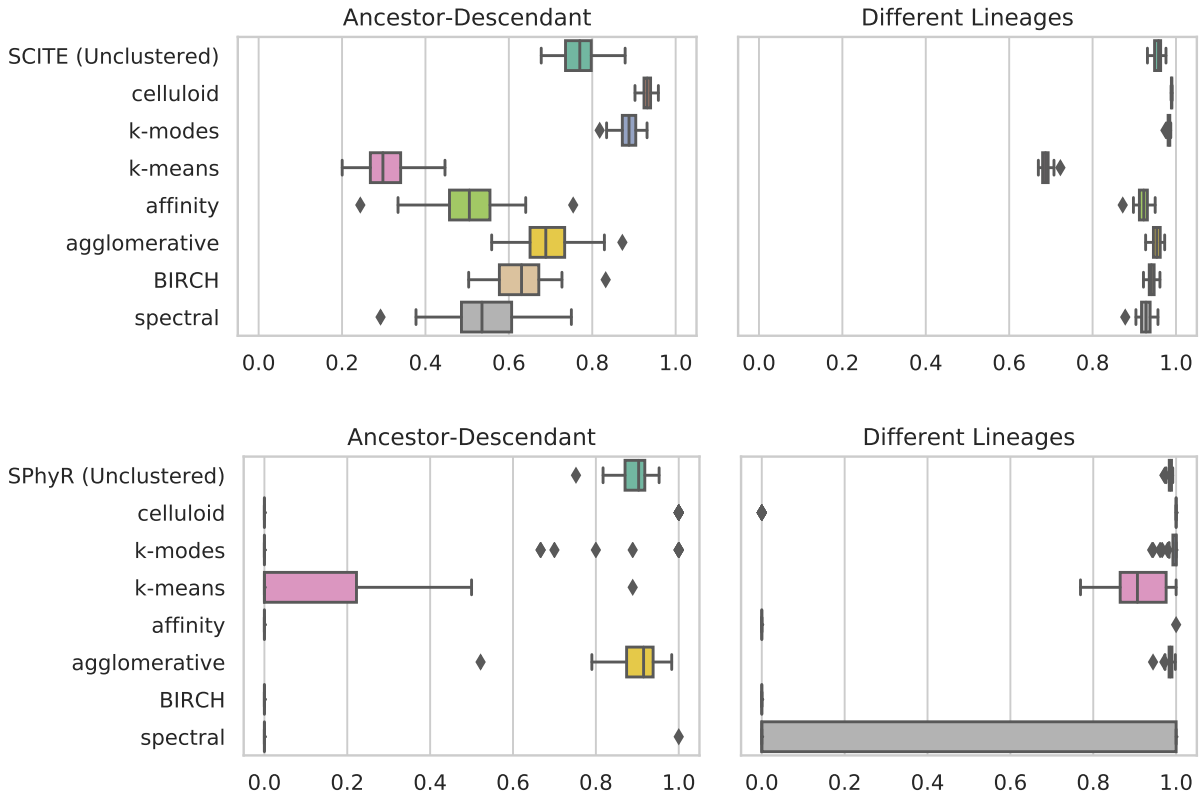

Figure 11: Ancestor-Descendant and Different lineages accuracy measures for experiment 2. Cancer phylogenies are inferred by SCITE (Top) and SPhyR (Bottom) using as input both the unclustered data as well as the clusters obtained by celluloid, k-means, k-modes, Affinity, Agglomerative, BIRCH and Spectral clustering.

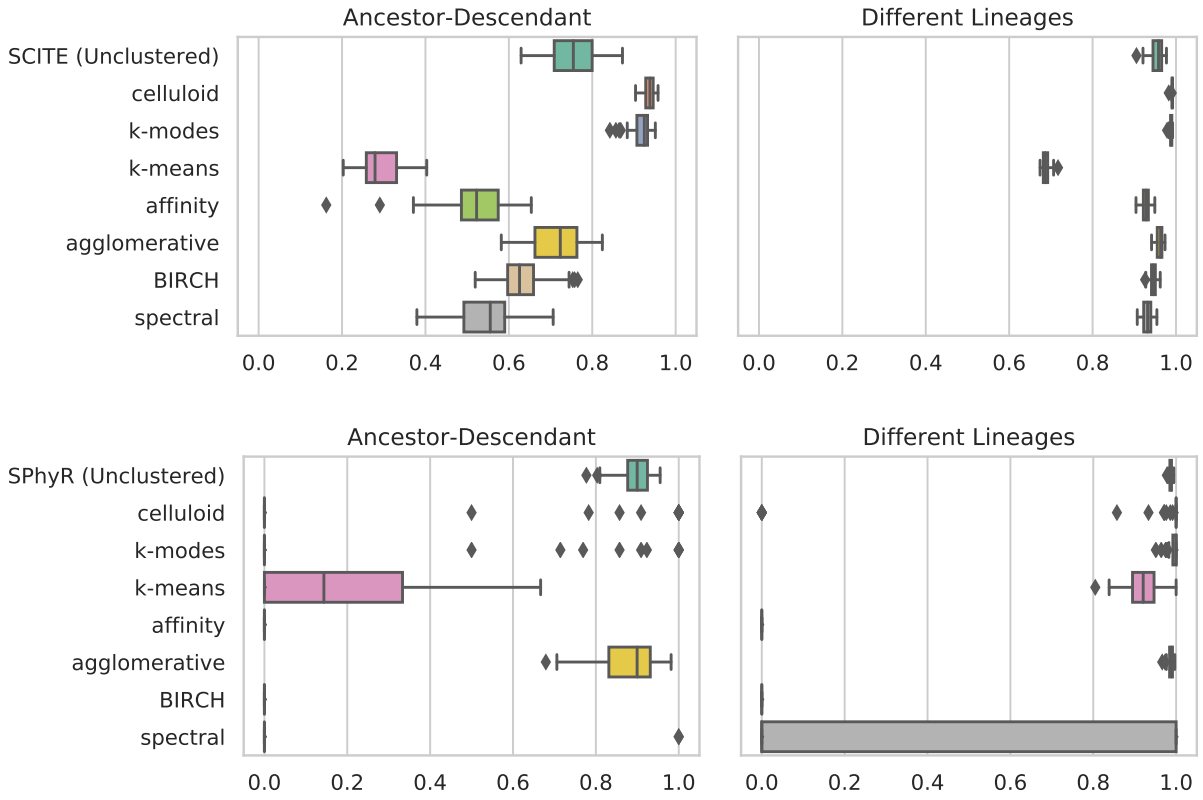

Figure 12: Ancestor-Descendant and Different lineages accuracy measures for experiment 3. Cancer phylogenies are inferred by SCITE (Top) and SPhyR (Bottom) using as input both the unclustered data as well as the clusters obtained by celluloid, k-means, k-modes, Affinity, Agglomerative, BIRCH and Spectral clustering.