Effective evaluation of clustering algorithms on single-cell CNA data

Effective Evaluation of Clustering Algorithms on Single-Cell CNA data

Marilisa Montemurro
Politecnido di Torino
Turin, Italy
marilisa.montemurro@polito.it

Gianvito Urgese
Politecnido di Torino
Turin, Italy
gianvito.urgese@polito.it

Elena Grassi
University of Turin
Candiolo (TO), Italy
elena.grassi@ircc.it

Carmelo Gabriele Pizzino
University of Turin
Candiolo (TO), Italy
carmelogabriele.pizzino@ircc.it

Andrea Bertotti
Univerity of Turin
Candiolo Cancer Institute – FPO
IRCCS
Candiolo (TO), Italy
andrea.bertotti@ircc.it

Elisa Ficarra
Politecnido di Torino
Turin, Italy
elisa.ficarra@polito.it

Abstract

Clustering methods are increasingly applied to single-cell DNA sequencing (scDNAseq) data to infer the subclonal structure of cancer. However, the complexity of these data exacerbates some data-science issues and affects clustering results. Additionally, determining whether such inferences are accurate and clusters recapitulate the real cell phylogeny is not trivial, mainly because ground truth information is not available for most experimental settings. Here, by exploiting simulated sequencing data representing known phylogenies of cancer cells, we propose a formal and systematic assessment of well-known clustering methods to study their performance and identify the approach providing the most accurate reconstruction of phylogenetic relationships.

CCS Concepts: • **Applied computing → Computational genomics**.

Keywords: clustering, copy-number, single-cell, CNA, benchmarking

ACM Reference Format:
Marilisa Montemurro, Gianvito Urgese, Elena Grassi, Carmelo Gabriele Pizzino, Andrea Bertotti, and Elisa Ficarra. 2020. Effective Evaluation of Clustering Algorithms on Single-Cell CNA data. In *Proceedings of 7th International Conference on Biomedical and Bioinformatics Engineering (ICBBE 2020).* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

ICBBE 2020, November 06-09, 2020, Kyoto, Japan

ACM ISBN 978-1-4503-8822-1…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

Cancer cells accumulate genetic alterations at every cell division, including both sequence variants and structural variations with gross copy number changes of entire genomic regions (i.e. copy number alterations, CNAs). On these premises, similarities in the genomic structure of individual cancer cells can be exploited to estimate the phylogenetic distance across different cells and consequently infer the subclonal structure of a tumor. For this reason, scDNAseq is becoming an increasingly popular technique [24, 28].

The most common way of inferring a single-cell CNA (scCNA) phylogeny is by performing hierarchical clustering on the CN profiles [3, 16], assuming that similar cells are very likely to have experienced the same mutational events. However, a number of biases could affect this kind of approach and vitiate the accuracy of the outcome. Specifically, clustering single-cell data exacerbates some biological data-science issues [30]. Indeed, the increasing number of cells which can be sequenced together expands the space of possible cluster assignments and determining the most meaningful results is not trivial without knowing the underlying biological truth. Additionally, the high-dimensional nature of such data harbors the "curse of dimensionality" [20]: distance metrics stop to behave as expected based on our low-dimensional intuition and clustering algorithms fail in determining distance between points. Moreover, the infinite-sites model does not apply to cancer CNAs [17], which intrinsically diminish the power of exploiting similarities in the genomic structure to predict phylogenies.

Although some of these issues have been partially addressed in the context of single-cell RNA methodology [18], in the case of scDNAseq, the extent of available data is still limited and there is need for the development of dedicated data analysis methods.

On these premises, the aim of the present work is to propose a first formal and systematic performance evaluation of nine well-known clustering methods on scCNA data.

We generated a synthetic scCNA dataset to evaluate the accuracy, the stability, the run time and scalability of nine clustering methods. Moreover, we compared the performance of the algorithms following different pre-processing steps. Finally, we tested the best performing methods on a real scCNA dataset obtained from colorectal cancer cells. The code used to perform our analysis is available at https://github.com/mmontemurro/clusteringbenchmarking.

2 Materials and Methods

In the following, we will describe the procedure to generate the simulated and the real scCNA datasets and the evaluation methods we have used for this work.

2.1 Simulations

We designed a simulation experiment to compare the performance of the clustering methods on datasets of three different sizes (100, 200 and 400 cells). Each experiment was iterated 50 times, for a total of 150 datasets.

Simulations were performed using the method presented by Fan et al. [11] which generates a phylogenetic tree starting from a reference genome, using a generalization of the Beta-Splitting model [6]. When a new edge enters the tree, a number of new CNAs is generated by sampling from a Poisson distribution (default $\lambda = 2$). The CNA size is determined by sampling from an exponential distribution (default mean=5Mbp), plus a minimum CNA size (default 2Mbp). The kind of alteration (*gain* vs. *loss*) is decided by a binomial distribution (default $p = 0.5$). If a CN gain is sampled, the number of copies to be gained is determined by means a geometric distribution (default $p = 0.5$). If a CN loss is sampled, the whole sequence on that region of the allele is deleted. The allele is chosen by drawing from a binomial distribution (default $p = 0.5$). The chromosome and the starting position of the CNA are sampled from a uniform distribution, bounded between 0 and the genome size. The daughter cell inherits all CNAs from the parent node, in addition to its unique CNAs. In agreement with the finite-site model of CNA evolution, new mutations may occur on already mutated sites. Additionally, to mimic the behaviour of punctuated evolution [15], at the edges to the root, whole-chromosome amplifications may occur, in addition to focal CNAs. The probability of a chromosome to be amplified at this step is set by means of a binomial distribution (default $p = 0.2$). Finally, the number of CNAs generated at this step may be increased by a given multiplying factor. At the end, the leaves of the generated tree represent the cells sampled from the patient, while the internal nodes represent intermediate CN states, which do not exist anymore.

In order to evaluate the ability of clustering methods to produce group of cells phylogenetically related, we converted the generated trees into easy-to-be-handled Newick format [12] and defined a set of clusters, directly from the trees, to be used as ground truth. The clusters are extracted as proposed from Balaban et al. [5], by solving an optimization problem that, given an arbitrary tree, returns the minimum number of clusters such that the maximum pairwise cophenetic distance between leaves in each cluster is lower than a given threshold. The threshold has been chosen according to the empiric observation that using a value equal to the height of each tree, a set of balanced clusters is obtained.

2.2 Clustering algorithms and evaluation methods

Since there is no formal evidence that hierarchical clustering should be preferred to other clustering paradigms in this scenario, we decided to take in consideration six among the mostly used methods, which implementation is available: *Affinity Propagation* [13], *Agglomerative Hierarchical clustering* [19], *Birch* [31], *DBSCAN* [10], *HDBSCAN* [26] and *K-Means* [23]. Additionally, we tested four variants of the agglomerative method [19]: *average linkage*, *complete linkage*, *single linkage* and *ward linkage*.

We applied each clustering method on every simulated dataset in three different scenarios: (i) without any preprocessing stage; (ii) after low variance feature filtering and PCA-based dimensionality reduction and (iii) after low variance feature filtering and UMAP-based dimensionality reduction.

The whole pipeline is fully automated. The Silhouette score maximization heuristic [8] has been used to determine the cluster number for the algorithms requiring it. Through this, we simulated a real world scenario, in which the cluster number is not known a priori and must be, arbitrary, chosen. For each dataset, the optimal number of PCs has been defined based on a randomization method, as described in Peres-Neto et al. [29]. This method consists in shuffling the dataset a number of times (default $N\_iter = 50$) and computing the percentage of variance explained by the PCs at every iteration. The significance of each PC is then defined as the probability that the permuted variance is greater than that observed one. Based on this, all the PCs characterized by a a p-value equal or below the threshold significance level (default $\alpha = 0.05$) are considered informative.

For each clustering method, we measured the execution time and computed the following indices:

- (*stability*) the *Average Proportion of Non-overlapping* (APN). This score measures the average incoherence between full data clustering and clustering based on data in which one dimension was removed. Values closer to 0 indicate good algorithm stability.
- (*accuracy*) the *Adjusted Rand index* (ARI), the *Adjusted Mutual Information* (AMI), the *V-Measure* (VM) and the *Fowlkes-Mallows Index* (FMI). These indices measure the similarity between the ground truth and clustering results. Values closer to 1 indicate good algorithm accuracy.

2.3 Single-cell sequencing

A real dataset has been generated by executing a scDNA-seq experiment on the human non-metastatic colorectal cancer-derived cell-line, SW480.

To this purpose, cells were cultured in L-15 medium supplemented with 10% FBS and 1% penicillin–streptomycin. To perform nuclei isolation, we proceeded accordingly to 10X Genomics protocol [1]. Briefly, 1 million cells were centrifuged (300 rcf for 5 minutes, at 4°C). Cell membranes were then lysed using a pre-chilled lysis buffer, and nuclei were pelleted by centrifugation (850 rcf for 5 minutes, at 4°C). Supernatant was removed, and nuclei were washed twice in PBS (0.04% BSA). After it, nuclei were counted, and re-suspended to a 1000 nuclei/ul concentration. Three thousand nuclei were processed accordingly to manufacturer protocol [2], to generate a barcoded DNA library from each nucleus. After QC check, libraries were sequenced on a Novaseq 6000 S1 flow cell (Illumina).

We used 10X Genomics proprietary pipeline [3], *Cell Ranger DNA*, to filter-out sequencing noise, align the reads against the GrCh38 reference genome and assign them to valid cell identifiers. We, then, demultiplexed the alignment file into single-cell .bam files, filtering out poor quality reads ($MAPQ < 30$), multimapping and secondary alignments. We, finally, used a customized version of *Ginkgo* [16] to extract scCNA profiles. The choice to use Ginkgo to call CNAs was motivated by the need of flexibility which is not fully provided by Cell Ranger DNA.

The resulting dataset contained 399 scCNA profiles.

3 Results and discussion

3.1 Evaluating clustering

We applied each clustering method on every simulated dataset, in the three preprocessing scenarios. Each clustering algorithm was therefore executed 450 times, for a total of 4050 clustering results. The evaluation metrics were computed for each algorithm run and then aggregated to summarize the results.

In the following, we will summarize the main results of our analysis.

3.1.1 Computation time. Figure 1 shows how the mean computation time increases as the input datasets become larger, in the no-preprocessing scenario. When dealing with small datasets, all algorithms achieve comparable performance; as the dataset size increases, density based algorithms (DBSCAN, HDBSCAN) behave worse than the others. This result was expected since it reflects the complexity of the algorithms.

3.1.2 Stability. Table 1 shows the mean APN score over different sizes of the input datasets, for the three preprocessing scenarios. All algorithm demonstrated good performance (APN near to 0), in terms of stability, in all tested conditions.

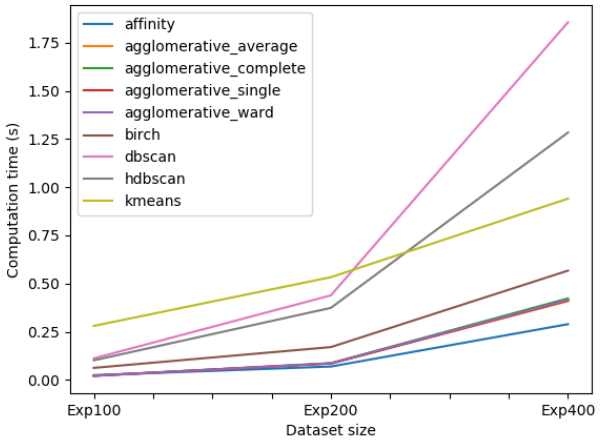


Figure 1. Clustering algorithm evaluation: mean computation time on non-reduced datasets.

However, in the absence of any preprocessing stage, K-Means and DBSCAN achieve the worse scores. Moreover, all the algorithms were less stable when applied to data preprocessed through PCA or UMAP. This is expected and coherent with the notion that following dimensionality reduction all the selected features are relevant for classification. As a final remark, it is interesting to notice that increasing the input dataset size, from 200 to 400 cells, improved the stability of DBSCAN.

3.1.3 Accuracy. Figures 2, 3 and 4 summarize the results of our analysis on clustering accuracy. We ranked algorithms to identify the most accurate one, for each input dataset size and preprocessing scenario. To this purpose, we first assigned a rank to each algorithm based on each validation index and then computed the overall performance as the average of the ranks.

The only algorithm which demonstrated good accuracy even in the absence of data preprocessing, is Affinity Propagation (AP) clustering. This is reasonable since the AP algorithm was already shown to perform well in various data-science fields, dealing with various kind of high-dimensional data [9, 14, 21, 22]. The reason of the good performance of AP is likely related to the fact that it does not take random samples for cluster centers but considers all points as possible exemplars [4].

On the contrary, it is interesting to notice that Agglomerative clustering based on single and average linkage consistently performed worse than the others, possibly because they are very sensitive to noise and, as a consequence, tend to produce a high number of little, singleton, clusters. In contrast, Agglomerative clustering with ward linkage performed better, in accordance with the notion that it generally produces more balanced clusters, and should be preferred

	Exp100			Exp200			Exp400		
	No preproc.	PCA	UMAP	No preproc.	PCA	UMAP	No preproc.	PCA	UMAP
affinity	0.0	0.061	0.246	0.0	0.052	0.345	0.001	0.051	0.33
agglomerative_average	0.0	0.019	0.106	0.0	0.019	0.148	0.0	0.006	0.209
agglomerative_complete	0.0	0.025	0.124	0.0	0.034	0.177	0.001	0.053	0.25
agglomerative_single	0.0	0.016	0.074	0.0	0.014	0.108	0.0	0.002	0.157
agglomerative_ward	0.0	0.028	0.119	0.0	0.026	0.172	0.0	0.023	0.24
birch	0.0	0.026	0.097	0.0	0.029	0.156	0.0	0.026	0.206
dbscan	0.003	0.123	0.459	0.089	0.087	0.388	0.042	0.081	0.225
hdbscan	0.0	0.046	0.001	0.002	0.052	0.0	0.003	0.057	0.0
kmeans	0.057	0.031	0.129	0.079	0.048	0.182	0.078	0.077	0.251

Table 1. Clustering algorithm evaluation: Mean APN scores.

when performing hierarchical clustering on non-reduced scCNA data.

However, for all dataset sizes, a better performance was achieved when clustering was applied following feature selection and dimensionality reduction. This confirms that the high-dimensional and noisy nature of this data negatively affects clustering results. In this scenario, PCA preprocessing was more effective when dealing with smaller datasets, while UMAP worked better with the larger ones. It is generally believed that clustering following UMAP embeddings should be avoided, since UMAP affects the global data structure, while maintaining the local relationships between data points [27]. UMAP can also create false tears in clusters, resulting in excessively fined grained clustering. Despite these concerns there are still valid reasons to use UMAP as a preprocessing step before clustering. Specifically, UMAP is particularly effective in uncovering the underlying signals from data with a very large number of dimensions, most of which are noisy or redundant. When this is the case, UMAP preprocessing may be therefore beneficial, provided that a manual inspection of the results is performed [25].

Indeed, at least in our experiment, on average the best performance was obtained when applying UMAP preprocessing, particularly when combined with density-based clustering approaches, which suggests that UMAP preprocessing may be useful to reduce scCNA data dimensionality before clustering.

In general, it is worth noting that the clustering methods which provided, on average, the most accurate results are those which does not require to be seeded with the cluster number. This may be a consequence of the automatic selection of the K, determined by maximizing the Silhouette score. This led us to conclude that, when dealing with large-scale and high-dimensional data, where the number of clusters is unknown, clustering methods which are able to infer the number of clusters, from the data, are always the best choice.

Moreover, to obtain an indication of the average accuracy of each algorithm in the three preprocessing scenarios, we rescaled the indices to the interval $[0, 1]$ and computed a mean accuracy score across the dataset sizes. Table 2 shows that UMAP should be preferred over PCA, especially, when used before running DBSCAN, or HDBSCAN. On the other side, to exploit the full resolution of the data, AP is the most accurate algorithm.

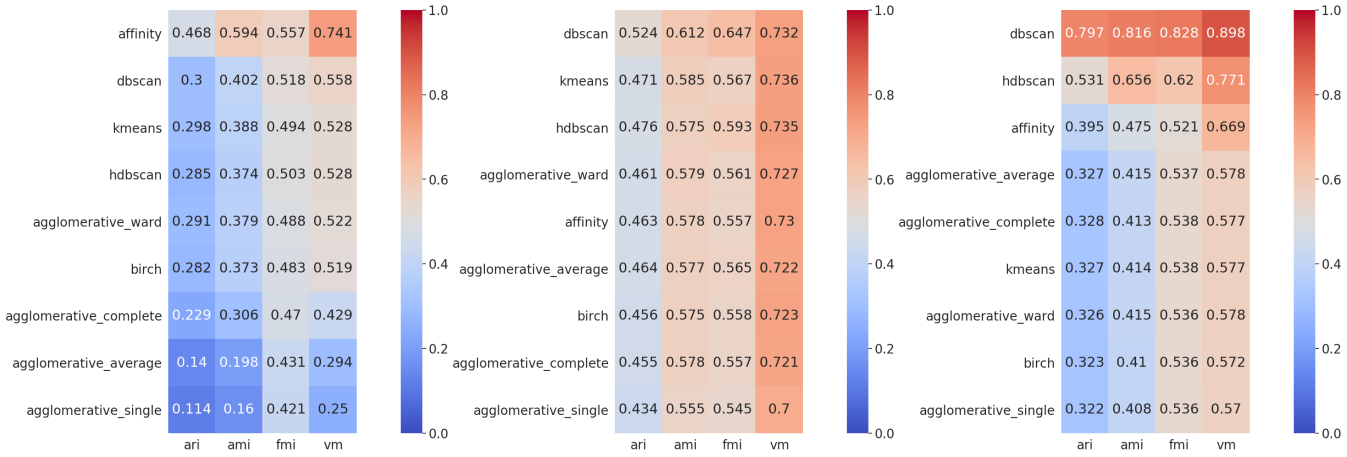
	No preproc.	PCA	UMAP
dbscan	0.401	0.715	0.783
kmeans	0.467	0.627	0.556
hdbscan	0.435	0.606	0.698
affinity	0.723	0.658	0.390
agglomerative_ward	0.465	0.602	0.557
birch	0.462	0.572	0.546
agglomerative_complete	0.351	0.460	0.550
agglomerative_average	0.164	0.364	0.547
agglomerative_single	0.141	0.298	0.538

Table 2. Clustering algorithm evaluation: overall mean accuracy scores.

3.2 Test case: SW480 cells

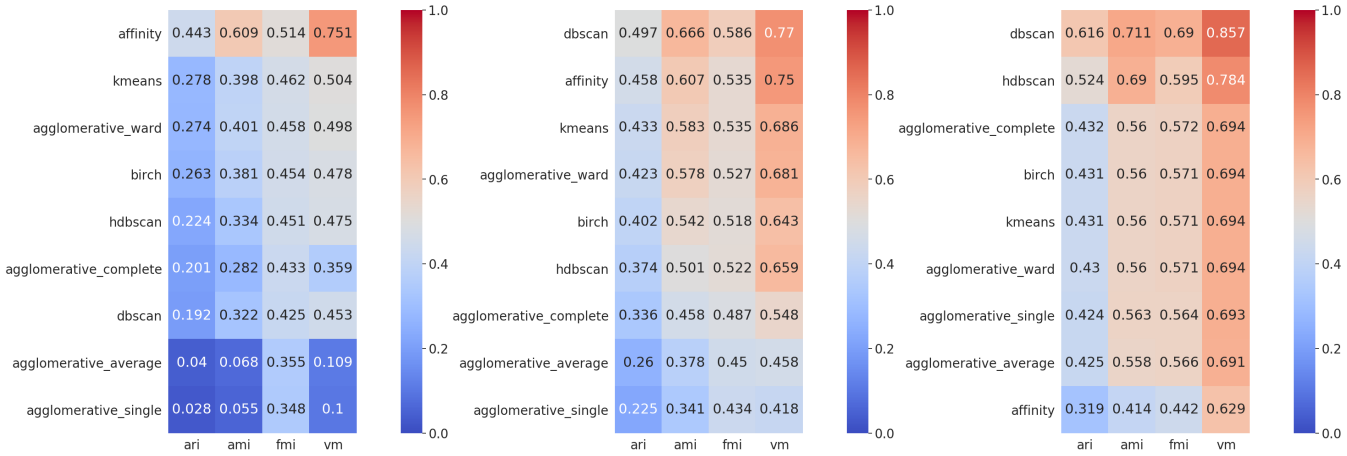
We tested the algorithms which achieved the best performance in the experiment with 400 cells, on SW480 cell data. As a general preprocessing step, we filtered out the cells characterized an high MAD ($> 90^{th}$ percentile). The MAD is the median absolute deviation of all pair-wise differences in read counts between neighboring bins and reflects the bin count dispersion due to technical noise. After that, we applied Affinity Propagation (AP) clustering to the non-reduced dataset and HBSCAN to UMAP-preprocessed data. In order to determine the model with better separation between the clusters, we computed the Davies-Bouldin score [7] (lower values signifies better cluster separation).

Figures 5a and 6a show AP results. Clusters composed of less than 10 items were excluded and only the 7 major clusters were kept for the further analysis. The scatter-plot (Figure 6a) shows that the clusters were well separated, with the exception of a few cells which have been mis-classified.



(a) No preprocessing **(b)** PCA **(c)** UMAP

Figure 2. Clustering algorithm accuracy on 100 cells dataset: mean external validation indices scores. Clustering algorithms are sorted according to the average ranking computed on all indices.



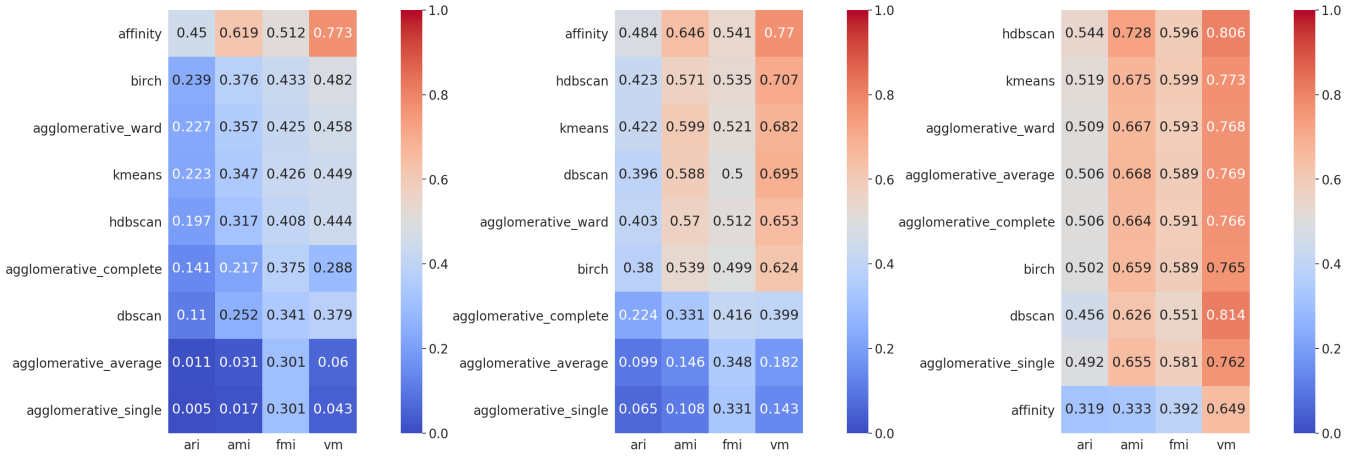
(a) No preprocessing **(b)** PCA **(c)** UMAP

Figure 3. Clustering algorithm accuracy on 200 cells dataset: mean external validation indices scores. Clustering algorithms are sorted according to the average ranking computed on all indices.

The heatmap shows that clusters were internally cohesive and each of them contained CNA profiles which were clearly distinguishable from those of the other clusters. The Davies-Bouldin score had value 9.933.

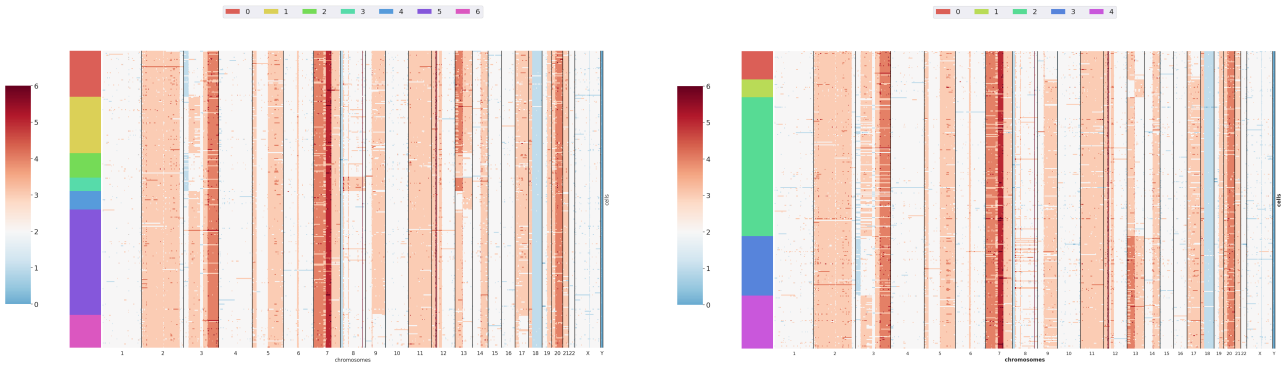
Figures 5b and 6b show HDBSCAN results. The cells marked as "noise" by the algorithm were excluded. Additionally, HDBSCAN library implements the GLOSH outlier detection algorithm which can detect outliers that may be noticeably different from points in its local region (for example points not on a local submanifold) but that are not necessarily outliers globally. So we took advantage of this feature to filter out also the cells with a high outlier-score ($> 90^{th}$ percentile). In the end, we obtained 5 clusters. The scatter-plot (Figure 6b) shows that, in this case, all cells were assigned to the most appropriate cluster. The heatmap (Figure 5b) shows quite consistent cluster, even if clusters 2 and 3 could have been splitted in two subclusters. The Davies-Bouldin score had value 10.950.

Remarkably, the clusters returned by the two methods, applied once on the raw dataset and once after an aggressive dimensionality reduction, performed with UMAP, are very similar. This means that UMAP may be used as a preprocessing step for clustering, as long as some manual validation of the results is performed. AP produced more clusters than HDBSCAN, because it was able to separate the cells which the latter algorithm put into clusters 2 and 3, as reflected



(a) No preprocessing **(b)** PCA **(c)** UMAP

Figure 4. Clustering algorithm accuracy on 400 cells dataset: mean external validation indices scores. Clustering algorithms are sorted according to the average ranking computed on all indices.



(a) AP **(b)** UMAP+HDBSCAN

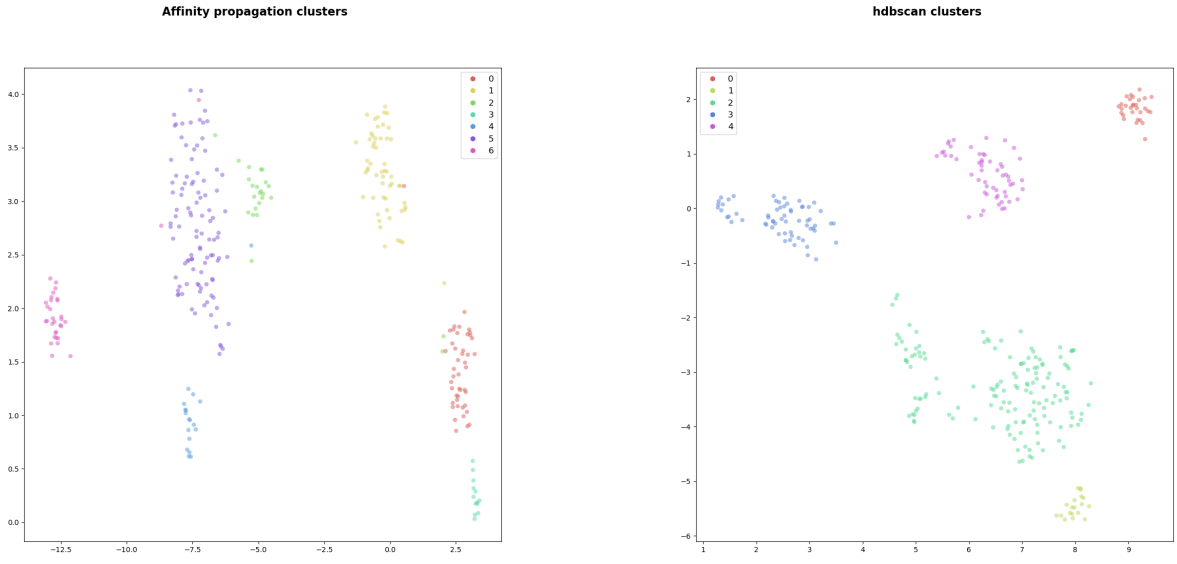
Figure 5. SW480 clusters. We applied AP on the non-reduced dataset (5a) and HBSCAN on the UMAP-reduced one (5b). The colored labels on the left-side of the heatmaps indicate the cluster which each cell was assigned to.

by the Davies-Bouldin score. On the other side, HDBSCAN is cluster shape independent and is resilient to noise and outliers.

4 Conclusion

The task of evaluating clustering algorithms performance on scCNA data has shown to be challenging and insidious, mainly because ground truth information about cell phylogeny is not available for public scDNAseq datasets.

Here, we exploited a synthetic dataset of single-cell CNA profiles with a known underlying phylogeny to perform the first formal and systematic evaluation of clustering algorithms onto single-cell CNA data which raises some data science issues. We have compared the performance of nine well-known clustering algorithms highlighting the pros and cons of the methods in predicting the structure of the real cell phylogeny. We took in consideration three different dataset sizes and both situations in which data are reduced to a lower dimensional space (PCA/UMAP) and when they are not. For each algorithm run we estimated the computation time, algorithm stability (APN) and the algorithm accuracy (ARI, AMI, FMI, VM). All of them showed to produce highly stable results, while density based algorithms are those which computation time increases more rapidly by increasing the dataset size. As for the accuracy, we ranked the algorithms, based on the average of the four indices. The algorithms which do not require to be seeded with the cluster number outperformed the others. Specifically, Affinity Propagation won when no dimensionality reduction was performed, while density based algorithms had very good results on top of PCA and UMAP results (DBSCAN for 100 and 200 cells dataset, HDBSCAN for 400 cells dataset).



(a) AP **(b)** UMAP+HDBSCAN

Figure 6. 2D representation of clustering results (without outliers). The 2D representation of the dataset shows that AP (6a) assigned a few cells to the wrong cluster, while HDBSCAN (6b) failed in splitting cluster 2 and 3.

We tested Affinity Propagation and HDBSCAN on a real scCNA dataset. AP was applied on the non-reduced dataset while HDBSCAN was performed following UMAP preprocessing. They both extracted cohesive and well-separated clusters. Moreover, the clusters identified by the two algorithms were similar, suggesting that UMAP may be effectively exploited to perform dimensionality-reduction. AP outperformed HDBSCAN in separating the items of two subgroups, which may indicate that retaining the full set of features may increase the resolution in subclones identification.

The main limitation of the present work is that the algorithm benchmarking was performed on synthetic data, due to the lack of an available biological ground truth; for this reason, we believe that an ad-hoc experiment should be designed to produce real data and extend our analysis.

To conclude, we have proposed a framework to study clustering algorithms performance on scCNA data, which can be easily replicated to perform similar studies.

Acknowledgments

Computational resources were provided by HPC@POLITO, a project of Academic Computing within the Department of Control and Computer Engineering at the Politecnico di Torino (http://www.hpc.polito.it). This work has been supported by the SmartData@PoliTO center on Big Data and Data Science, the AIRC 5x1000 grant (21091) and the European Research Council Consolidator grant (724748 – BEAT).

References

[1] 10x Genomics. 2020. *Isolation of Nuclei for Single Cell DNA Sequencing.* https://support.10xgenomics.com/single-cell-dna/sample-prep/doc/demonstrated-protocol-isolation-of-nuclei-for-single-cell-dna-sequencing Last accessed June 26, 2020.

[2] 10x Genomics. 2020. *USER GUIDE. Chromium Single Cell DNA Reagent Kits.* https://assets.ctfassets.net/an68im79xiti/3BEOhYw96Bjtb8jlKOum9z/28612e6bcf479581cfd216855e8d6dcc/CG000153_UserGuideSingleCellDNA_ReagentKits_RevC.pdf Last accessed June 26, 2020.

[3] 10x Genomics. 2020. *What is Cell Ranger DNA?* https://support.10xgenomics.com/single-cell-dna/software/pipelines/latest/what-is-cell-ranger-dna Last accessed June 26, 2020.

[4] Osama Abu Abbas. 2008. Comparisons Between Data Clustering Algorithms. *International Arab Journal of Information Technology (IAJIT)* 5, 3 (2008).

[5] Metin Balaban, Niema Moshiri, Uyen Mai, Xingfan Jia, and Siavash Mirarab. 2019. TreeCluster: Clustering biological sequences using phylogenetic trees. *PloS one* 14, 8 (2019), e0221068.

[6] Michael GB Blum and Olivier François. 2006. Which random processes describe the tree of life? A large-scale study of phylogenetic tree imbalance. *Systematic Biology* 55, 4 (2006), 685–691.

[7] David L Davies and Donald W Bouldin. 1979. A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence* 2 (1979), 224–227.

[8] Renato Cordeiro De Amorim and Christian Hennig. 2015. Recovering the number of clusters in data sets with noise features using feature rescaling factors. *Information Sciences* 324 (2015), 126–145.

[9] Chunhua Du, Jie Yang, Qiang Wu, and Feng Li. 2007. Integrating affinity propagation clustering method with linear discriminant analysis for face recognition. *Optical Engineering* 46, 11 (2007), 110501.

[10] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise.. In *Kdd*, Vol. 96. 226–231.

[11] Xian Fan, Mohammadamin Edrisi, Nicholas Navin, and Luay Nakhleh. 2019. Benchmarking tools for copy number aberration detection from single-cell DNA sequencing data. *bioRxiv* (2019), 696179.
[12] Joseph Felsenstein, J Archie, W Day, W Maddison, C Meacham, F Rohlf, and D Swofford. 1986. The Newick tree format.
[13] Brendan J Frey and Delbert Dueck. 2007. Clustering by passing messages between data points. *science* 315, 5814 (2007), 972–976.
[14] Guojun Gan and Michael Kwok-Po Ng. 2015. Subspace clustering using affinity propagation. *Pattern Recognition* 48, 4 (2015), 1455–1464.
[15] Ruli Gao, Alexander Davis, Thomas O McDonald, Emi Sei, Xiuqing Shi, Yong Wang, Pei-Ching Tsai, Anna Casasent, Jill Waters, Hong Zhang, et al. 2016. Punctuated copy number evolution and clonal stasis in triple-negative breast cancer. *Nature genetics* 48, 10 (2016), 1119.
[16] Tyler Garvin, Robert Aboukhalil, Jude Kendall, Timour Baslan, Gurinder S Atwal, James Hicks, Michael Wigler, and Michael C Schatz. 2015. Interactive analysis and assessment of single-cell copy-number variations. *Nature methods* 12, 11 (2015), 1058. https://doi.org/10.1038/nmeth.3578
[17] Xin-Sheng Hu, Yang Hu, and Xiaoyang Chen. 2016. Testing neutrality at copy-number-variable loci under the finite-allele and finite-site models. *Theoretical Population Biology* 112 (2016), 1–13.
[18] Giovanni Iacono, Elisabetta Mereu, Amy Guillaumet-Adkins, Roser Corominas, Ivon Cuscó, Gustavo Rodríguez-Esteban, Marta Gut, Luis Alberto Pérez-Jurado, Ivo Gut, and Holger Heyn. 2018. bigSCale: an analytical framework for big-scale single-cell data. *Genome research* 28, 6 (2018), 878–890.
[19] Stephen C Johnson. 1967. Hierarchical clustering schemes. *Psychometrika* 32, 3 (1967), 241–254.
[20] Mario Köppen. 2000. The curse of dimensionality. In *5th Online World Conference on Soft Computing in Industrial Applications (WSC5)*, Vol. 1. 4–8.
[21] Darong Lai and Hongtao Lu. 2008. Identification of community structure in complex networks using affinity propagation clustering method. *Modern Physics Letters B* 22, 16 (2008), 1547–1566.
[22] Jianjun Liu and Jianquan Kan. 2018. Recognition of genetically modified product based on affinity propagation clustering and terahertz spectroscopy. *Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy* 194 (2018), 14–20.
[23] James MacQueen et al. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, Vol. 1. Oakland, CA, USA, 281–297.
[24] Andriy Marusyk and Kornelia Polyak. 2010. Tumor heterogeneity: causes and consequences. *Biochimica et Biophysica Acta (BBA)-Reviews on Cancer* 1805, 1 (2010), 105–117.
[25] Leland McInnes. 2020. *Using UMAP for Clustering*. https://umap-learn.readthedocs.io/en/latest/clustering.html Last accessed June 26, 2020.
[26] Leland McInnes and John Healy. 2017. Accelerated hierarchical density based clustering. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*. IEEE, 33–42.
[27] Leland McInnes, John Healy, and James Melville. 2018. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426* (2018).
[28] Nicholas Navin, Jude Kendall, Jennifer Troge, Peter Andrews, Linda Rodgers, Jeanne McIndoo, Kerry Cook, Asya Stepansky, Dan Levy, Diane Esposito, et al. 2011. Tumour evolution inferred by single-cell sequencing. *Nature* 472, 7341 (2011), 90.
[29] Pedro R Peres-Neto, Donald A Jackson, and Keith M Somers. 2005. How many principal components? Stopping rules for determining the number of non-trivial axes revisited. *Computational Statistics & Data Analysis* 49, 4 (2005), 974–997.
[30] Tom Ronan, Zhijie Qi, and Kristen M Naegle. 2016. Avoiding common pitfalls when clustering biological data. *Science signaling* 9, 432 (2016), re6–re6.
[31] Tian Zhang, Raghu Ramakrishnan, and Miron Livny. 1996. BIRCH: an efficient data clustering method for very large databases. *ACM sigmod record* 25, 2 (1996), 103–114.