Forest Fire Clustering: Kernel-based Affinity Clustering and Monte Carlo Verification Inspired by Forest Fire Dynamics

Zhanlin Chen[1], Jeremy Goldwasser[1], Philip Tuckman[2], Jing Zhang[3], Mark Gerstein[4,5,*]
[1.]Program in of Statistics and Data Science, Yale University, New Haven, CT 06520, USA
[2.]Program in of Atmosphere, Oceans, and Climate, Massachusetts Institute of Technology, Boston, MA 02139, USA [3.]Department of Computer Science, University of California, Irvine, CA 92617, USA [4.]Department of Molecular Biophysics and Biochemistry, and [5.]Department of Computer Science, Yale University, New Haven, CT 06520, USA

*To whom correspondence should be addressed.

Abstract

Summary: Clustering methods group similar data points together and assign them group-level labels. Here, we introduce a novel kernel-based clustering method modeled after forest fire dynamics. After constructing a graph, we utilized network properties as parameters to simulate the propagation of labels similar to the spread of forest fires. Through this method, we can drastically reduce the number of input parameters down to just one - a temperature term describing how easily one label propagates from one node to the next. By iteratively starting small controlled burns through the network, we can discover the number of clusters in the data with minimum prior assumptions. Further, we can verify our predictions and probe the posterior probability distribution of the labels using Monte Carlo methods. Lastly, our iterative method is better for inference because we don't need to refit the model to predict the labels on new data. Here, we describe the method and provide a summary of common clustering benchmarks.
Contact: pi@gersteinlab.org

Introduction

Clustering analysis is an important task in statistical data analysis and data mining. They have been utilized in a wide variety of application scenarios, from finding clusters in social networks to detecting fraudulent bank activity [1-5]. In biology, the rise of single-cell sequencing technologies allows researchers to examine the genetic information down to the resolution of a single cell [6, 7]. One of the powerful applications of this technology is to cluster and categorize individual cells into cell types based on genomic features, especially in detecting subtypes of cancer cells in molecularly targeted therapy [8]. However, a single-cell dataset could contain millions of cells and hundreds of thousands of genomic features. The increase in information challenges current data analysis methods, especially high dimensional clustering algorithms. The presence of rare and unknown cell types could go against previous assumptions about cellular composition [9]. Therefore, any prior assumption about the data could bias the analysis and would fail to capture the nuances of rare and potentially influential cell types. In addition, the doublet or multiplet cell effect in single-cell sequencing occurs when two or more cells are

mistakenly sequenced and tagged as one cell [10-12]. These artifactual data points show cell type characteristics from two or more cell types and could not be removed by merely examining the number of sequencing reads per cell. There is currently no effective method for distinguishing and handling these artifacts, and they severely confound the clustering analysis because such noise is treated with equal weight in the clustering process.

There are many definitions of what constitutes a cluster and many ways to find them efficiently. However, current clustering algorithms are heavily parameterized with assumptions about the data. Further, there is almost no way to evaluate how confident a clustering method is in their output due to methodological limitations. Here, we introduce a method that could find clusters without prior knowledge on the number of clusters in the data and calculate a pointwise confidence score of the predicted labels with Monte Carlo methods. We provide a summary of our method's performance compared to other clustering methods using common clustering benchmarks. Furthermore, we demonstrate its usefulness in discovering rare cell types in single-cell sequencing.

Related Works

Previously, there have been four general approaches in clustering data: connectivity-based, centroid-based, distribution-based, and density-based methods.

Connectivity-based clustering methods, also known as hierarchical clustering, iteratively merge data points into the same group based on some evaluation of distance to obtain a hierarchical structure [13]. These types of methods could often be represented with a dendrogram, with the data points on one axis while tracking the merge history on another axis. The benefit is that there are no assumptions about the prior distribution of the data. However, the continuous nature of dendrograms provides no definitive cut-off on the resolution of the clusters, therefore it cannot detect cluster boundaries.

Centroid-based clustering methods rely on optimizing a central vector to find data clusters, such as k-means clustering [14]. This type of method clusters data into a Voronoi diagram and is interpretable. However, it is heavily parameterized on the number of clusters in the data and assumes a spherical distribution centered around the central vector. Also, Lloyd's algorithm optimizes this NP-hard process but only finds the local minimum.

In distribution-based clustering methods, clusters are defined as samples from certain distributions. Most prominently, Gaussian Mixture Models assume a gaussian distribution of the data [15]. However, there is also an assumption on the shape and number of distributions. It is easy to overfit the labels by increasing the complexity of the model.

Density-based clustering methods rely on the sparse regions to distinguish between clusters. Characteristic algorithms in this category such as DBSCAN and OPTICS are fast and produce interpretable clusters based on density [16, 17]. However, they require a range parameter and do not perform well when there are overlapping dense clusters.

Lastly, Louvain and Leiden community detection methods do provide us with an efficient approach to find the number of clusters in the data given a desired resolution [18]. However, since cluster labels do not correspond from one run to another, these types of analysis are methodologically limited in providing us a confidence measure.

Notably, though, there are important characteristics in these five general approaches that would help us deal with diverse and complex datasets such as single-cell sequencing. For example, connectivity-based methods deal with linkage between nodes, and consequently, assumes little about the number of clusters in the data. Density-based methods borrowed some ideas from connectivity-based methods and create non-linear clusters based on reachability. Distribution-based clustering assumes centers located at the mean of distributions, which makes the cluster shapes interpretable and enhances performance during overlapping dense clusters. We reference these central ideas when constructing our clustering algorithm and are able to create a model without the drawbacks associated with these methods.

Background

Our clustering method took inspiration from the controlled burns used by Native Americans to protect and cultivate the land [19]. By setting flints at random locations in the forest, there could be multiple clusters of fire burning, and the fire would only sustain when it is propagating through a highly dense patch of trees. Once it reaches the edges of the patch, the sparsity would not be able to sustain the fire, and the fire would slowly extinguish. Furthermore, in forest fire dynamics, if nearby trees are on fire, it increases the probability of the surrounded tree to catch on fire as well [20].

We operationalized this concept by drawing parallels between forest fires and clustering analysis. Here, we represent a dataset as a graph with data points as nodes and distances between them as edges. Each node would take after the behavior of a tree, and the fire would represent the label for a particular cluster. Because fires consume the trees and are mutually exclusive, we ensure that there could not be more than one label for every node.

Further, we utilize the kernel method to transform the distance matrix into a non-linear affinity matrix for clustering [21]. Diffusion maps also utilize the affinity matrix to extract high dimensional diffusion trajectories [22]. By starting with an affinity matrix, we could extract similar high-dimensional information and use it to identify clusters in the data without the need for dimensionality reduction.

Monte Carlo method is a computational algorithm that allows us to simulate and reconstruct the posterior probability distribution by repeating a large number of random trials [23]. Similarly, we utilize the Monte Carlo method by randomly selecting nodes for label propagation to build a posterior probability distribution of the labels on each data point. Through this, we could provide a pointwise confidence score in our predicted labels.

Method and Theory

In this section, we describe our forest fire clustering method. For a data matrix W with N rows and M features, we calculate the pairwise distance matrix D between every data point N. Then, we utilize the kernel trick to transform the distance matrix into an affinity matrix A. The affinity matrix could be seen as a graph with each data point as nodes and the pairwise affinity as edges, and could also be understood as the forest in the forest fire analogy. Further, by using an adaptive kernel based on K-nearest neighbors, we could not only preserve local distances

but also encoding a longer trajectory structure. Using this graph, we can extrapolate network properties that control the dynamics of how labels (or the controlled burns as per the analogy) spread throughout similar nodes.

First, we observe that there are two important features in forest fire dynamics: the threshold above which a node would catch on fire (T) and the temperature of the fire (F).

The threshold is closely related to the flashpoint in forest fire dynamics. If the temperature of the environment is above a certain threshold, then the tree would start burning. For each tree, it could have a state (S) of burned, denoted by i from the ith fire, or unburned, denoted by 0, corresponding to whether a node has been labeled or not.

$$S = \begin{cases} i, F \geq T \\ 0, F < T \end{cases} \tag{1}$$

Here, a node would inherit labels from nearby nodes if the temperature is above a certain threshold. In order for the fire to slowly stop at the cluster edges where the data is sparse, the threshold would have to be closely related to the density of the graph, as the more dense a region is, the more likely it is to belong to a cluster. Therefore, we set the threshold to be the inverse of the degree of a node and standardize the proportionality constant for the threshold to be one. (2)

$$Threshold = \frac{1}{Degree} \tag{2}$$

For the fire temperature, it governs how much the label of nearby unlabeled nodes is influenced by the label of the current node. The temperature would radiate onto other nodes with a non-linear decay function, and we directly utilize the affinity between two nodes i and j. (3) The proportionality constant c for the fire temperature is the only parameter of our clustering algorithm. With a higher fire temperature, the resulting clusters will also be bigger, since fire representing one label is more likely to spread to nearby trees. One could also discover smaller clusters by decreasing the fire temperature of the model. The proportionality constant c increases in conjunction with the kernel bandwidth, where a higher momentum from a higher fire temperature could compensate for the decay of large distances.

$$F_{i,j} = c * A_{i,j} \tag{3}$$

Looking at a single tree, it is possible that one tree on fire is not enough to set the nearby tree on fire initially, but as nearby trees catch on fire over time, the fire temperature could accumulate from nearby trees, which could push over the threshold and light the current tree on fire. Therefore, the fire temperature for at a unlabeled node k could be seen as the temperature of the surrounding nodes $1 \ldots n$. (4)

$$F_k = F_{1,k} + F_{2,k} + \ldots + F_{n,k} \tag{4}$$

The clustering algorithm recursively checks each node with breadth-first search (BFS) whether the cumulated fire temperature is higher than the threshold and iteratively adds neighboring nodes to the current cluster. After each label propagation slowly stops, another unlabeled node will be randomly selected as a starting node, and the process continues until all nodes have been labeled. The pseudo-code for the algorithm is shown below.

```
Calculate pairwise distance between data points
Apply kernel and transform into affinity matrix
If there are nodes with no labels, randomly choose a node to cast a label
       Continue until labels gradually stop propagating
              For every other unlabeled node
                     Calculate the mean temperature from neighboring labeled nodes
                     Determine if the total temperature is higher than the threshold
                            If higher, node inherits nearby labels
                            If not higher, do nothing
Return labels for each node
```

For Monte Carlo, after predicting a set of labels, we randomly choose a node to start the propagation of the predicted label. By evaluating the number of times a node inherits different labels, we could obtain a posterior probability distribution of the cluster labels.

Our method is different yet references ideas from existing methods. For example, we utilize linkage between data points in the form of affinity to generate clusters similar to hierarchical clustering, but we are able to detect cluster borders through forest fire propagation dynamics. In addition, unlike centroid- and distribution- based methods, we do not have any prior assumptions on the number and distribution of points in a cluster. Further, our method is closely related to density-based methods, but does not use a range parameter. Similar to density-based methods, our clustering method could be used to infer the label of new data, without the need of reclustering a union of the new and the old dataset. We also averaged the total fire temperature by the number of unlabeled nearby nodes to centralize the existing clusters, which improves its performance on overlapping dense clusters compared to density-based methods.

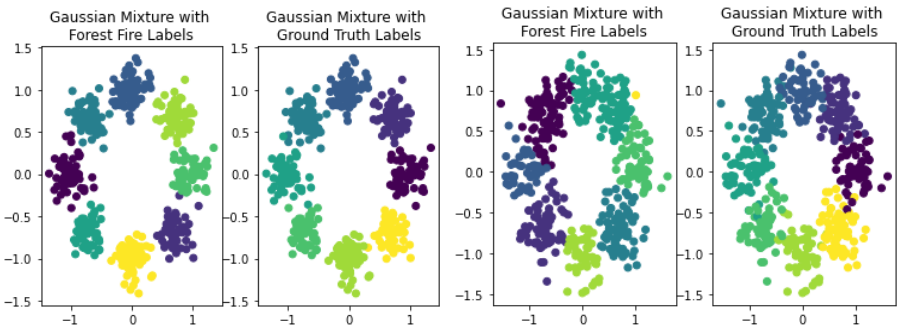
All of the previously mentioned methods suffer from the curse of dimensionality. High dimensional datasets skew distances between data points, which negatively affects the clustering results. Diffusion maps have been shown to be an effective manifold learning technique, and it utilizes affinity matrices for dimensionality reduction. Equivalently, our method also uses affinity matrices and maintains its robust performance on high dimensional datasets. Therefore, our technique is not limited by other dimensionality reduction techniques.

Empirical Results

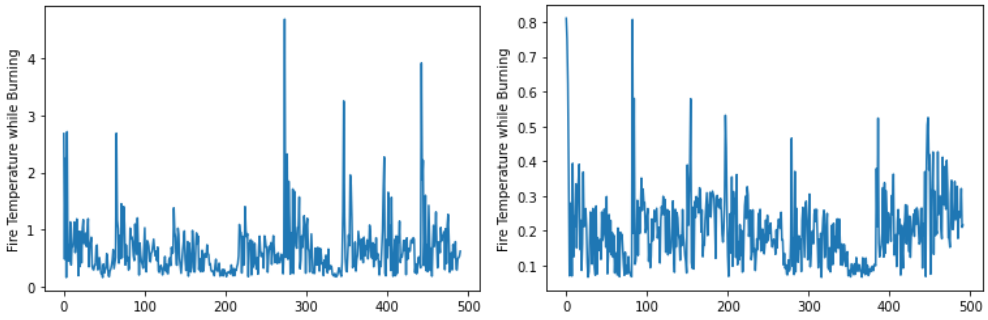
1. Gaussian Mixture Model

We constructed a Gaussian Mixture Model with 500 data points sampled from 8 distributions located evenly around the unit circle ($\sigma = 0.15, \sigma = 0.2$). Figures below show the

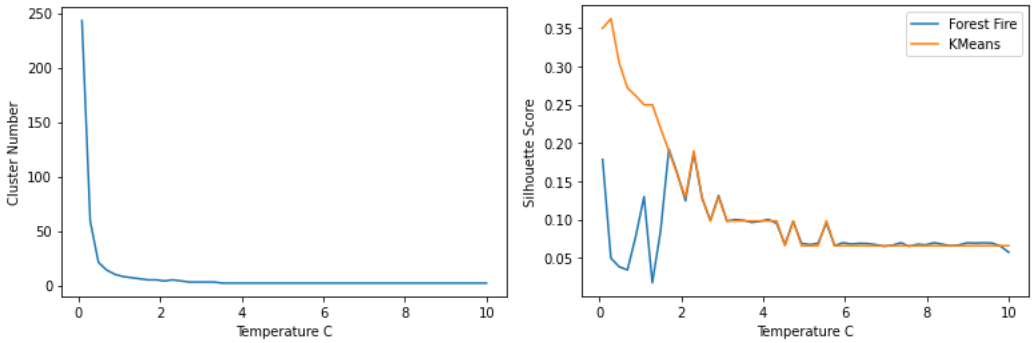
clustering results from our clustering method with $c = 1$. The model maintains robust performance as a density-based method even with overlapping clusters, which could be attributed to our modification on averaging the fire temperature across nodes of the existing cluster while spreading. This example demonstrates that our model has the ability to infer the number of clusters and improves upon current density-based models.



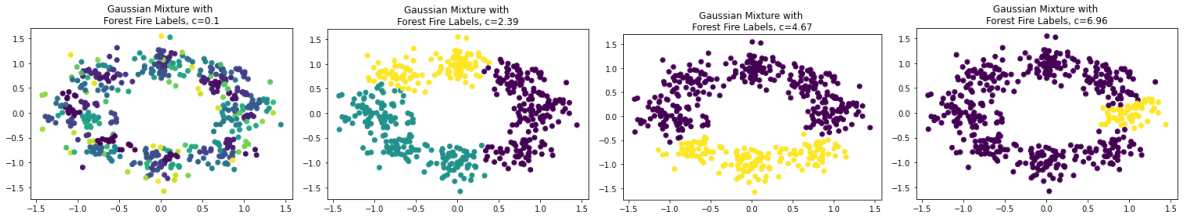
Interestingly, the number of clusters could also be inferred from the fire temperature over time plot, where it records the cumulative fire temperature of each point when it was labeled. The plot sheds new light on the clustering process and makes the clustering results more interpretable. In the plot, every spike corresponds to a meaningful cluster that had been discovered. Particularly, in the Gaussian Mixture with higher noise ($\sigma = 0.2$), there were eight clusters, but one of which only has one data point (shown in yellow). We could see that from the fire temperature over time plot as there were only seven spikes. Hence, the plot could be used as a heuristic measure to judge the quality of clustering results. Further, it shows that our model is able to detect the cluster edges by gradually decreasing in temperature and stopping the label propagation at sparse regions.



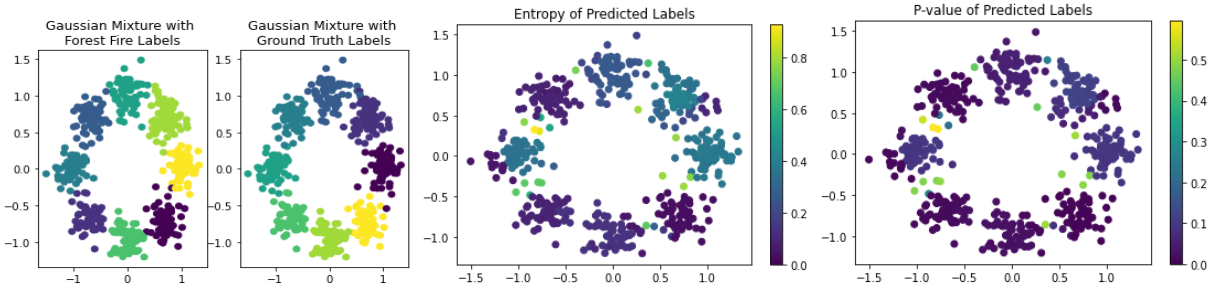
Since there is only one parameter in our model, we also performed experiments on the robustness of the parameter. We verified our hypothesis that as the fire temperature increases, the resolution of the cluster decreases. The silhouette score for K-means is locally optimal when given the number of clusters K. Here, we discovered the number of clusters using our clustering method and compared the silhouette score of our method with K-means. Given the number of clusters, our method converges with K-means, showing that the forest fire dynamics produces near optimal clustering results.



On one end, when the fire temperature is low, each node would not have enough momentum to propagate its label to nearby nodes. On the other end, when the fire temperature is too high, one label could ravage through the entire graph. Lastly, we observe a degree of stochasticity in the clustering results, where nodes are not always assigned their own cluster, particularly when the noise is high.

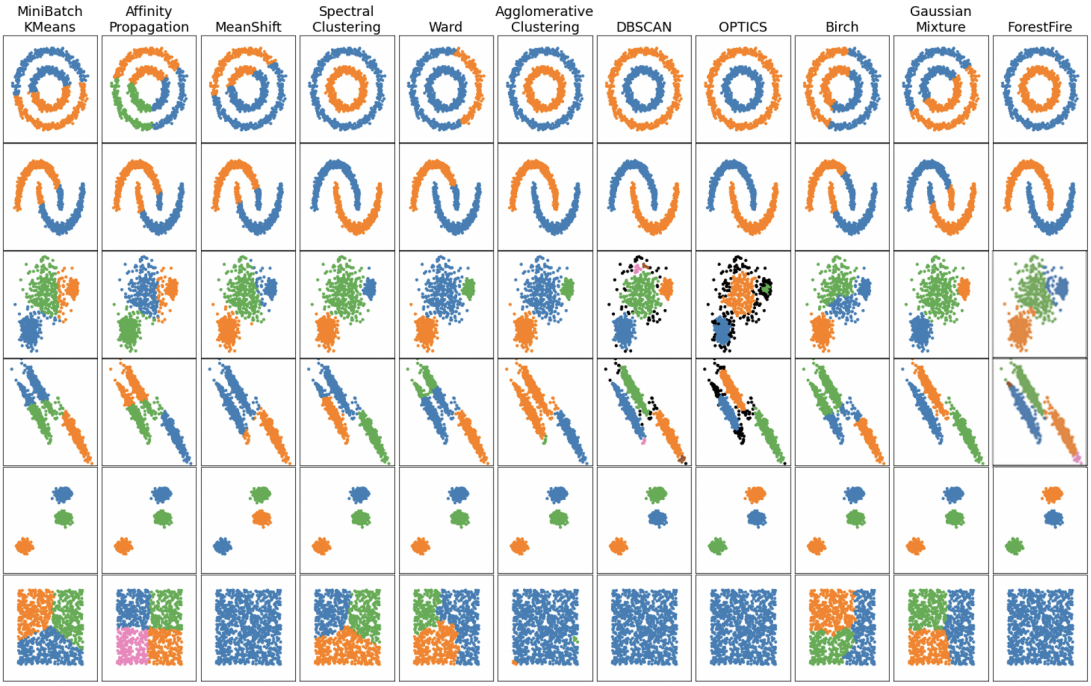


We further investigated the difference and ambiguity in labeling across different runs. By utilizing the Monte Carlo method through permuting the starting node of the labeled cluster, the posterior probability distribution of the labels could be found, and the entropy of the label for each data point could be calculated. From the entropy plot, we see that there is more ambiguity on particular clusters and on the edges of certain clusters. Using the same logic, we could also calculate a p-value for each prediction. In contract, other clustering methods are limited in calculating the confidence score because there is no correspondence between runs due to cluster label variance. For single-cell analysis, this pointwise confidence score could be used to detect and filter doublet cell artifacts.



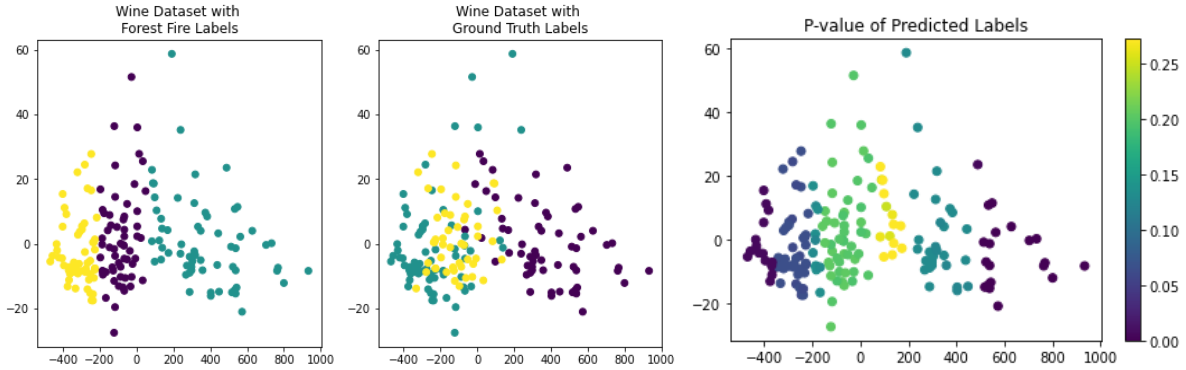
2. Toy datasets

We then benchmarked the performance of our clustering method with existing clustering methods on toy datasets, as shown in the image panel below. With a non-linear transformation using kernel methods, our clustering results span over manifold spaces rather than the original data space. Further, we show that our method has the ability to detect clusters without assuming the shape of the clusters, contrasting with Gaussian Mixture Models. We also maintain robust performance on scenarios with overlapping clusters as compared to density-based methods like DBSCAN and OPTICS. Lastly, we are able to discover the number of clusters in the data without any prior assumptions, whereas Spectral Clustering and Mini-Batch K-means are significantly biased with those assumptions.



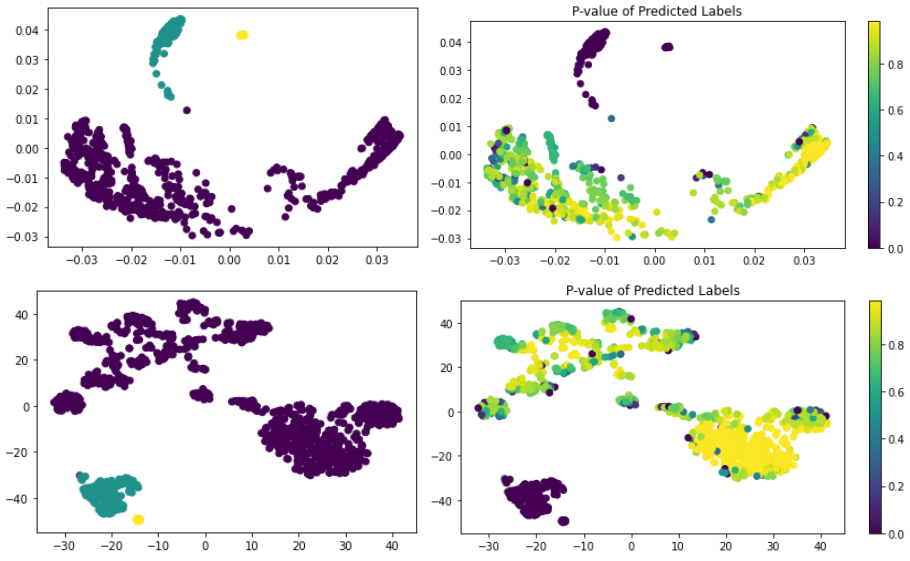
3. Wine dataset

Subsequently, we evaluated the performance of our model on high dimensional toy data. Below, we predicted clusters in the wine dataset with data points viewed on PCA projections (PC 1-2). We show that our method is still able to discover three clusters without any prior assumptions. It performs better on points farther away from the center, which corresponds to more distinguishable data points. The performance decreases for central points as the data becomes more homogenous.



4. Single-cell Data (Retinal Bipolar Cells)

Lastly, we utilized a real-world single-cell transcriptomic experiment on retinal bipolar cells to demonstrate the effectiveness of our method. Shekar et al. found distinct and previously undocumented subtypes of retinal bipolar cells from a dataset with 25,000 mouse retinal bipolar cells. With preprocessing, we show that we were able to discover rare cell types with the top 100 features that corroborated with conclusions in Shekhar et al., as shown in the t-SNE and PHATE projected coordinates below.



Conclusion

Clustering analysis is a great way to label and classify datasets by identifying the commonalities between data. As an unsupervised learning technique, it has been ubiquitous in various real world application scenarios. However, many of the current clustering algorithms have their own drawbacks. Connectivity-based methods cannot detect cluster edges. Centroid-

and distribution-based methods are heavily parameterized. Density-based methods do not perform well with overlapping clusters and have a discrete range parameter. Previous approaches also could not handle high dimensional data. Here, we propose a clustering method inspired by forest fire dynamics that address the drawbacks of previous methods. It has only one parameter that controls the temperature of the fire and could be used to discover the number of clusters in the data with minimum prior assumptions. Further, we show that it outperforms previous methods on toy datasets, especially density-based methods with overlapping clusters. Lastly, our method also remains robust in high dimensions, and it is able to discover rare cell types in real-world single-cell experiments.

The future of single-cell sequencing analysis is to ultimately construct a cellular network using the genetic interactions between different cell types [24]. However, in order to create cellular networks, we need to first define cell types and cluster similar cells together. Previous methods like Louvain and Leiden can discover the number of clusters in the data, but they cannot evaluate the confidence of the cluster labels. They treat noisy data equally, and the network built on these clustering results could be severely confounded by experimental artifacts. Using our method, we could define and cluster cell types and offer the possibility of constructing a cellular network using high confidence and robust cells and clusters.

References

1. Ding, L., Z. Feng, and Y. Bai, *Clustering analysis of microRNA and mRNA expression data from TCGA using maximum edge-weighted matching algorithms.* BMC Med Genomics, 2019. **12**(1): p. 117.
2. Dunn, H., et al., *Cluster Analysis in Nursing Research: An Introduction, Historical Perspective, and Future Directions.* West J Nurs Res, 2018. **40**(11): p. 1658-1676.
3. Hoffman, M., et al., *Detecting Clusters/Communities in Social Networks.* Multivariate Behav Res, 2018. **53**(1): p. 57-73.
4. Kiselev, V.Y., T.S. Andrews, and M. Hemberg, *Challenges in unsupervised clustering of single-cell RNA-seq data.* Nat Rev Genet, 2019. **20**(5): p. 273-282.
5. Singh, P. and M. Singh, *Fraud Detection by Monitoring Customer Behavior and Activities.* International Journal of Computer Applications, 2015. **111**(11).
6. Shalek, A.K., et al., *Single-cell RNA-seq reveals dynamic paracrine control of cellular variation.* Nature, 2014. **510**(7505): p. 363-9.
7. Tang, F., et al., *mRNA-Seq whole-transcriptome analysis of a single cell.* Nat Methods, 2009. **6**(5): p. 377-82.
8. Saadatpour, A., et al., *Single-Cell Analysis in Cancer Genomics.* Trends Genet, 2015. **31**(10): p. 576-586.
9. Chen, B., et al., *Profiling Tumor Infiltrating Immune Cells with CIBERSORT.* Methods Mol Biol, 2018. **1711**: p. 243-259.
10. Bernstein, N.J., et al., *Solo: Doublet Identification in Single-Cell RNA-Seq via Semi-Supervised Deep Learning.* Cell Syst, 2020. **11**(1): p. 95-101 e5.
11. DePasquale, E.A.K., et al., *DoubletDecon: Deconvoluting Doublets from Single-Cell RNA-Sequencing Data.* Cell Rep, 2019. **29**(6): p. 1718-1727 e8.
12. McGinnis, C.S., L.M. Murrow, and Z.J. Gartner, *DoubletFinder: Doublet Detection in Single-Cell RNA Sequencing Data Using Artificial Nearest Neighbors.* Cell Syst, 2019. **8**(4): p. 329-337 e4.

13. Ding, C. and X. He, *Cluster merging and splitting in hierarchical clustering algorithms.* IEEE International Conference on Data Mining, 2002.
14. Hartigan, J.A. and M.A. Wong, *A K-Means Clustering Algorithm.* Journal of the Royal Statistical Society, 1979. **The 28**(1): p. 8.
15. Sahbi, H., *A particular Gaussian mixture model for clustering and its application to image retrieval.* Soft Computing, 2007. **12**: p. 9.
16. Ankerst, M., et al., *OPTICS: Ordering Points To Identify the Clustering Structure.* International Conference on Management of Data, 1999.
17. Ester, M., et al., *A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise.* Association for the Advancement of Artificial Intelligence Proceedings 1996.
18. Traag, V.A., L. Waltman, and N.J. van Eck, *From Louvain to Leiden: guaranteeing well-connected communities.* Sci Rep, 2019. **9**(1): p. 5233.
19. Kay, C.E., *Native Burning in Western North America: Implications for Hardwood Forest Management.* Proceedings: Workshop on Fire, People, and the Central Hardwoods Landscape, 2000.
20. Albini, F., *Estimating Wildfire Behavior and Effects* USDA Forest Service General Technical Report INT-30 1976.
21. Hofmann, T., B. Schölkopf, and A.J. Smola, *Kernel methods in machine learning.* The annals of statistics, 2008: p. 1171-1220.
22. Nadler, B., et al., *Diffusion maps, spectral clustering and eigenfunctions of Fokker-Planck operators.* Advances in neural information processing systems, 2005. **18**: p. 955-962.
23. Hammersley, J., *Monte carlo methods*. 2013: Springer Science & Business Media.
24. Aibar, S., et al., *SCENIC: single-cell regulatory network inference and clustering.* Nature methods, 2017. **14**(11): p. 1083-1086.