Optimization of a memory gene selection for annotating cell-families in scRNAseq data, a machine learning approach

Maffei Theo, Dormann Alexia, Ben M'Rad Imane
Laboratory Supervisor: Eisele Almut, from UPSUTER EPFL laboratory
EPFL Lausanne, Switzerland

***Abstract*—In the past few decades, a multitude of novel high-throughput techniques in biology have allowed researchers to collect vast amounts of data. However, when too many variables are included, learning algorithms tend to overfit and produce overall poor performance. [1] The dimensionality of the data needs to be reduced. A typical way to reduce the dimension of the data without altering the inputs is by feature selection. In the present paper, this dimension reduction technique was performed to extract gene sets able to predict cell families in scRNAseq data. It has previously been shown that some genes are more stably inherited over cell divisions than others. From data sets in which lineage tracing was combined with scRNAseq these can now be assessed widely and in different cell types. Thus, the identification of these memory genes could lead the way for a novel method of lineage tracing.**

I. Introduction

Single cell RNA sequencing enables to asses gene expression genome-wide in single cells at high throughput. Different scRNAseq methods exist, among which plate (ex.SMART-seq) and microfluidic based (10X CHormium, InDrop) systems, and all are widely used. ScRNAseq profiles are a snapshot view of gene expression and a high interest exist to add temporal information to such data. [2] One way to do so today is through assessing the relation of cells through their family belonging. That can be done using lineage tracing techniques. Typically, researchers use cellular barcoding of cells before scRNAseq. A unique nucleotide sequence is introduced in each cell. Since this barcode is introduced in the genome of the cells, all progenitor cells inherited this sequence. Thus, cells with the same barcode are identified as a family.

This has recently been implemented in BIDDY [3] and Weinreb [4] papers, giving insights into reprogramming and hematopoietic differentiation respectively. The technique however requires the previous isolation, lentiviral transduction and culture of cells and is thereby restricted to few, mainly in vitro, experimental setups. A way to assess lineage relation in scRNAseq data without prior handling would be very valuable. An analysis by Eisele et al. of scRNAseq with annotated family information revealed that in different cell types the levels of some genes are more stable than of others within cell families. These genes are called memory genes. Furthermore an overlap of these stable genes was found between different cell types opening up the possibility to predict cell families using these genes.

An initial statistical analysis of the transcriptome performed by Eisele and al. identified a subset of genes that could predict the family of a given cell with a somewhat high accuracy. However, the precision of the prediction could be improved. Thus, the aim of this project is to discover the best set of genes enabling to predict a cellular family using a machine learning approach. Firstly, the set of genes was optimized on each data set independently using feature selection. The aim was to find a subset of genes that works well across data sets. Finally, an ideal overlap of the genes in the individual optimization was identified.

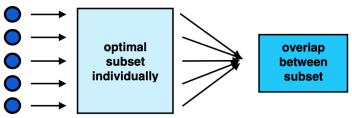

Fig. 1. Project structure

II. Data sets

Multiple family-annotated scRNAseq data sets were used to optimize the subset of memory genes, a total of 15 data sets were analyzed. Two data sets were provided by the host laboratory at EPFL, whereas all others came from recent publications. Moreover, all the different sources analyzed the transcriptome of different cell types and cells were cultured 48h to 72h:

- UPSUTER, Mouse embryonic stem cells(mESCs)
- Biddy, Mouse embryonic fibroblasts(MEF) [3]
- Kimmerling, CD8+ T cells and L1210 lymphocytic leukemia cells [5]
- Weinreb/LARRY, Hematopoietic progenitor cells (LSK and LK) [4]

Note that the data in the different publications were obtained using different scRNAseq technologies. Furthermore, each of the data set will be referred by their abbreviation throughout this paper: UPSUTER: AE3 and AE4, Biddy: D0, D02, D6, D62, D15, and D152, Kimmerling: KCD8 and L1210, and Weinreb: LK_1 and LSK_1 More details on each data set can be found in the appendix.

A. Data pre-processing

The raw data of scRNAseq was already analyzed and a few pre-processing steps were applied by Eisele and al. The

cells with high transcription rate of mitochondrial genes were filtered out as they are most likely dead cells. In addition, cells with abnormally high or low read numbers were filtered out of our analysis. The read count was also normalized, so that all read values in a cell sum up to 40'000, making it possible to compare read numbers between cells. Finally, only cells that belonged to relatively small families, between 2-5 cells depending on the data set, were taken into account since the cells were only cultured 48h or 72h. Indeed, families that contain a large number of cells or only one cell are probably not true families.

B. Initial gene selection

A memory gene should have the following characteristics:

- high correlation between family members
- low intrafamily variability, high interfamily variability
- close to variability-mean regression line
- overlapping marker for family clusters

These characteristics for each gene were evaluated by comparing them to random samples using several statistical methods by Eisele and al. All genes that were considered significantly potential memory genes were kept in our initial subset. In addition, a popular R package for scRNAseq analysis was used: Seurat.[6] It identifies genes with differential expression that define clusters. The genes that were estimated important markers of cluster identity were also included in our initial preselection of potential memory genes. Furthermore, only the genes that were expressed in at least 50% of the cells were considered, otherwise the prediction would certainly not be general.

III. Methods

A. Prediction algorithm and scoring

To handle this task, several approaches were investigated by Eisele et al. These methods include K-means, hierarchical clustering on distance matrix and hierarchical clustering on correlations. Indeed, agglomerative hierarchical clustering revealed itself to be very appropriate for the task at hand. A measure of dissimilarity and linkage criterion are required to determine which clusters should be merged. [7] The similarity of the cells was determined by taking the Spearman correlation between genes expression, and the ward criterion was used. Unlike other commonly used criterion, the ward method does not directly compute a distance. Instead, it tries to minimize the variance of the clusters.

$$minD(A,B) = \frac{n_A n_B}{n_A + n_B}\delta(A,B)^2$$

with n_j the number of point in cluster j, and $\delta(A,B)$ the distance between A and B (here Spearman correlation). [8]

One problem that arose was that simply segregating the cells into a fixed number N of clusters did not perform well. The predicted clusters were either very large or contained only one cell. Thus, the number of final clusters was constraint using a convenient python library: scipy_cut_tree_balanced. [9]

Numerous methods exist to evaluate clustering predictions. A lot of these computations involved in some way a confusion matrix, a matrix summarizing the performance of the classification algorithm, containing true/false positive and false/true negative. In the context of this paper, they are defined as follow:

- True positive (TP), two cells in same cluster belong to the same family
- False positive (FP), two cells in same cluster belong to different families
- True negative (TN), two cells in different clusters belong to different families
- False negative (FN), two cells in different clusters belong to the same family

Empirically, the precision and the sensitivity were deemed the most appropriate to estimate the models. They give a good estimate of how well the clustering algorithm is able to correctly cluster cells together. Thus, they were chosen as the score to maximize during feature selection. Depending on the data sets, one or the other gave better results.

$$precision = \frac{TP}{TP+FP}, sensitivity = \frac{TP}{TP+FN}$$

For the sake of optimization and time, cross-validation was not performed in the algorithms mainly for two reasons. First of all, as the data sets have initially on average more than 30'000 features, executing cross-validation is computationally too costly and it would waste some time to do cross-validation while that time could have been used to test several additional methods. And secondly, usually cross-validation is used when there are training steps for example for finding optimal weights and for model selection. However, the used prediction model did not involve any training, thus it is expected that the performance of the model on new data and the data at hand should be similar if both data sets are relatively homogeneous.

Furthermore, ideally a train/test split should have been used to get a better understanding of the performance of the different investigated gene subsets and avoid overfitting. However, all the investigated data sets are very small (a bit less than a hundred in most data sets) and performing this split would even further decrease the number of samples used for feature selection thus leading to poorer performance. This is why a train/test split was not performed.

B. Feature selection

Feature selection is commonly used to reduce the dimensionality of biological data. Ideally, the resulting set will be a small subset of relevant features. The selected subset should lead to better performance, decrease the computational complexity and increase the interpretability of the model, which may help better understand the underlying processes.[10] Usually, the best subset contains a set of features that are complementary to each other and that are useful for classification. Note that a feature that is relevant on its own, may be redundant when it is considered together with other features. In principle, every possible subset should be investigated. However, given the very large search space of possible subsets, an exhaustive search is infeasible in almost all situations. [11] Thus, a large variety of sub-optimal feature selection methods were developed over the years. In

the subsequent sections, only the feature selection methods that performed well were described. A overview of the other methods can be found in the appendix.

Commonly, feature selection algorithms are classified into three broad categories, but only two of these categories will be used: filter and wrapper methods. Indeed, the third one, embedding method, was not deemed appropriate as it integrates the features selection process into the prediction algorithm as its name suggest. However, the prediction method was already chosen beforehand by the supervisor and integrating the feature selection would therefore not be a possibility.

1) Filtering methods: Filter methods are interested in the statistical properties of the data. Given a set of N features, filter feature selection chooses a subset with K features, which maximizes some criterion. These methods are independent from the learning algorithm making them computationally less complex and unbiased to the prediction method. [11]

- Mutual information maximization (MIM) utilizes the mutual information. It is an estimate of the statistical dependency between random variables. It is able to capture non-linear dependencies.

$$MI = I(X;Y) = H(X)–H(X|Y)$$

$$H(X) = -\sum_{i=1}^{n} P(x_i)log_2(P(x_i))$$

  The simplest way of using this scoring criterion is by ranking the features with decreasing mutual information content and choosing the top N features.
- Mutual information feature selection (MIFS)
- Joint mutual information (JMI)
- Double Input Symmetrical Relevance (DISR)
- Fisher's score criterion
- Laplacian score criterion
- RelieF feature selection algorithm
- Fast correlation-based Feature Selection (FCBF)
- Hilbert Schmidt Independence Criterion Lasso (HSIC Lasso)

2) Wrapping methods: Contrary to filter methods, wrapper methods include a learning algorithm. A score is attributed to a subset by computing some accuracy value of the model. Generally, wrapper approaches achieve better results than filter methods, but they tend to be very computationally expensive. [11]

- Stochastic optimization of features selection (SO)
- Stimulated annealing (SA) is inspired by the annealing of solids. When a solid is hot, the particles composing it are rapidly rearranging, it helps strengthen the material. The frequency of these rearrangements decreases as the solid cools down. Each feature can be seen as a particle and the collection of variables as the material. A set of features is selected at random, the model is then evaluated on this subset. Then, a few features are included or excluded from the subset. Again, the subset is evaluated, and if the computed score is better, we keep the new set. Otherwise, if the new accuracy is worse, then an acceptance probability, which is a function of time, is calculated:

$$Pr[accept] = \exp(\frac{i}{c}\frac{old - new}{old})$$

  Next, a random number is drawn, if it is smaller than the acceptance, then the new subset is kept. Otherwise, we reject the new subset. This helps escape local minimum. [12]
- Genetic algorithm (GA) is inspired by the biological evolution of bacteria populations. An initial population composed of multiple subsets is generated. The score is computed for each member of the population. Then, the fittest subsets are selected for the following generation. The subsequent subsets are composed of the winning features with some cross-over and where mutations can also occur. This process is repeated over multiple generations. Finally, the final optimal subset maximizing some criterion is chosen in the last generation. [12]

3) Hybrid methods: MIM outperformed all other filter feature selection methods (see results). Thus, hybrid versions of MIM and wrapper methods were implemented. First, N features were selected using MIM. Then, SA or GA was run on this new subset.

- Mutual information maximizer followed by stimulated annealing (MI/SA)
- Mutual information maximizer followed by genetic algorithm (MI/GA)

IV. Result and discussion

A. Optimization of memory genes on every data sets

As mentioned previously, the optimization of memory genes subset was run independently on each data set with every mentioned methods. The detailed results of this optimization process can be found in the appendix. A lot of the methods selected subsets that did not improve the families prediction, some even decreased the initial established baseline.

Concerning the filter methods, the MIM method outperformed all of them. Indeed, it was able to select a set of genes that increased significantly all our precision measures in all data sets. . In addition, the HSIC Lasso method also performed quiet well on some of the data sets. The optimization attempts with the wrapper methods resulted in large subsets of genes which improve the prediction performance. But to expend what was already done here, it could be interesting to try to constraint the final subset to a smaller number of genes since they tend to give better results. Moreover, the hybrid methods that first selected a subset using MIM, then run stimulated annealing or genetic algorithm outperformed the simple MIM implementation in most cases. For example, in the data set L1210, the precision and specificity with MIM were respectively 0.780 and 0.770, whereas it increases to 0.817 and 0.930 with MI/GA.

The optimization on most of the Biddy and Kimmerling data sets resulted in very satisfactory results (D0, D6, D62, D15, D152, KCD8, and L1210). On the bigger data sets (AE3, DO2), the analysis gave significant improvement, but to a

lesser extend. On one hand, machine learning analysis of data with fewer samples tend to be prone to overfitting As mentioned before, a good practice to avoid overfitting is to split the data into train and test sets. Since this split was not performed, it would be wise to further investigate if the found subsets actually overfit the data. But, on the other hand, this trend was also observed by Eisele and al. in their analysis. The variation of performance may be explained by this inherent difficulty to identify memory genes in some of the data sets.

Data set	Best method	size subset	precision	sensitivity
AE3	MI/GA	293	0.446	0.870
D0	MI/SA	48	1.0	1.0
D0_2	MI/GA	559	0.726	0.957
D6	MI/GA	137	1.0	0.983
D6_2	MI/SA	212	1.0	1.0
D15	MI/SA	198	1.0	1.0
D15_2	MI/GA	124	1.0	1.0
CD8	MI/GA	335	0.898	0.846
L1210	MI/GA	169	0.817	0.930

TABLE I
BEST FEATURE SELECTION METHOD ON EACH DATA SET

Due to the time concern, the hyperparameters were not finely tuned on each data set. Instead, we found an appropriate set of parameters on the AE3 data set and used them in the subsequent optimization on other data sets. These hyperparameters can be found in TabII. Methods without tunable parameters are not included in the table. Nevertheless, one should consider that finer tuning of the parameters for each data set could improve the reported results.

Method	method parameters
MIM	n_neighbors = 3
MIFS	$\beta = 1$
ReliefF algorithm	n_neighbors = 5
SO	n_iter = 1000, p_mutate= 0.1
SA	n_iters = 1000, p_mutate = 0.1, c = 1
GA	n_population = 300, crossover_proba = 0.5, mutation_proba = 0.2, n_generations= 40, tournament_size = 3
MI/SA	initial_MI_selection = 400, n_neighbors = 3, n_iters = 5000, p_mutate = 0.05, c = 1
MI/GA	initial_MI_selection = 850, n_neighbors = 3, n_generations= 200, rest same as GA

TABLE II
HYPERPARAMETERS USED FOR FEATURE SELECTION METHODS

Unfortunately, some of the wrapper methods were not performed on the AE4 data set as it was computationally too expensive because of the large number of cells. Thus, the optimization on the AE4 data will not be used in the following overlap analysis. Moreover, data sets LK1 and LSK1 were kept for evaluating the performance of the optimal overlap subset.

B. Optimal set across data sets

In order to find the optimal overlap between the subsets, the best subset on each data set is chosen independently (see Tab. I) The criterion to do so is the sum of the precision and the sensitivity as these seemed to be relevant to evaluate the clustering. Moreover, when two methods gave similar results, the larger subset was chosen. As the data is heterogeneous, feature selection methods resulted in different optimal subsets with low overlap (see Fig 2) . Therefore, choosing the larger subset should guarantee a better overlap between the subsets.

Fig. 2. Overlap of optimized subset between AE3, CD8, and D62

It is noteworthy that the independent optimization yielded very different subsets of genes. It is the reason why the overlap between the different sets is that low. Nevertheless, the final optimal subset composed of all genes that appear at least twice in the individual optimization subsets does not perform so badly. Its size is 370 genes. It increases most of the baselines' precision and sensitivity. However, it is observed that it does not perform very well on D62 and L1210. One possible explanation to that result is that the number of cells in these sets is very low. Thus, a general subset cannot perform very well on them.

Data set	precision	sensitivity	Data set	precision	sensitivity
AE3	0.302	0.569	AE4	0.032	0.140
DO	1.0	0.964	DO_2	0.632	0.822
D6	0.857	0.736	D6_2	0.666	0.842
D15	1.0	0.977	D15_2	1.0	0.956
CD8	0.574	0.453	L1210	0.391	0.225
LK_1	0.080	0.249	LSK_1	0.024	0.161

TABLE III
PERFORMANCE ON EACH DATA SET WITH OVERLAP SUBSET

V. CONCLUSION

In conclusion, the optimized feature selection enabled to increase the performance of the family clustering in most data sets. To improve our work, some additional feature selection methods could be used to enhance the clustering and may lead to better results. Furthermore, the optimization was run on an already preselected set of genes. A more unbiased approach could have been taken by starting with every genes in the data set that is expressed at least at some relatively high threshold. Concern of overfitting also need to be addressed. However, as mentioned most data sets are too small to be split into a train/test train. One solution to this problem could be to augment the sample size by boot-strapping. A technique in which the training set is generated by random selection of samples with replacement. [13] Finally, the overall workflow should probably be modified. Instead, of optimizing the subset of memory genes on each data set independently, a more integrated approach could be taken, meaning that the calculated score should also take into account the performance of a given subset on all data sets.

APPENDIX

A. More details on the data sets

The following table shows the source, cell type, culture time and sequencing method for each data set.

Source	ID	Cell type	Cultured time	Sequencing
UPSUTER	AE3	mESC	48h	10X
	AE4	mESC	72h	10X
Biddy	D0	MEF	48h	10X
	D0_2	MEF	48h	10X
	D6	MEF	72h	10X
	D6_2	MEF	72h	10X
	D15	MEF	48h	10X
	D15_2	MEF	48h	10X
Kimmerling	CD8	CD8+ T cells	-	SmartSeq
	L1210	Leukimia cell line	-	SmartSeq
Larry	D2_LSK	LSK	48h	inDrop
	D2_LSK_2	LSK	48h	inDrop
	D2_LS_2	LS	48h	inDrop
	D2_LS_2	LS	48h	inDrop
	D2_mix	mix LSK and LS	48h	inDrop

TABLE IV
DETAILS ON EACH DATA SET

B. Details of all feature selection methods not mentioned in the main part

- A known limitation of MIM method is that each feature is considered independently. However, as seen previously, an input that is complementary with the already selected subset should be included. Thus, when features are interdependent, which is often the case, it is sup-optimal. [14] To address this issue, a variety of methods were proposed that find the relevant and non-redundant features. For example, the **mutual information feature selection** (MIFS) introduces a penalty term that helps select non-redundant features. An other way to maximize this relevance-redundancy trade-off is by **Joint mutual information** (JMI). Instead of the individual MI of each feature, it uses the joint mutual information $I(X_K) = \sum I(X_k X_j, Y)$ to rank the features. Lastly, **Double Input Symmetrical Relevance** (DISR) estimates a complementary value, that estimates the additional information contained in a set of features about the output. [15]
- **Fisher's score** is one of the most used filter features selection methods given its good performance on a variety of data. The fisher's scoring is calculated for each variable independently. The main idea is to choose the features such that the distances between sample of different class is as large as possible, whereas the distances between objects of the same class is small. [16]
- **Laplacian score** method ranks the features according to their locality preserving power. This is determined using a Laplacian eigenmaps and locality preserving projection. Note that this is an unsupervised method. [17]
- **RelieF** estimates an approximation of each feature quality and relevance given the target variable. [18]
- **Fast correlation-based Feature Selection** (FCBF) uses correlation as a measure of the relevance of the features. Indeed, a variable is deemed relevant if it is highly correlated to the class, but not to other features. FCBF uses a correlation measure based on entropy.[19]
- **Hilbert Schmidt Independence Criterion Lasso** (HSIC Lasso) aims to find the best trade-off between redundancy and relevance. To do so, it computes the HSIC criterion, a kernel-based independence score. Additionally, thanks to the l_1-regularizer, lasso, it forces features with score close to zero to be eliminated. [20]
- **Stochastic optimization of features selection** (SO). The possible subset space is search stochastically. At each iteration, each feature included in the initial subset has a probability p of not being included in the new subset and inversely. At each iteration, a few features are flipped, a score is calculated and if this new subset improves the previous score, the subset is kept, otherwise it does not change. However, this approach will have a tendency to get stuck in a local minimum.

C. Optimization result on each data sets

Find here the detailed results obtained for each data sets using all methods. The best results are highlighted in blue.

Method	subset size	precision	sensitivity
Baseline preselected genes	5380	0.390	0.298
MIM	89	0.391	0.895
MIFS	600	0.185	0.436
JMI	500	0.343	0.118
Fisher's score	600	0.190	0.430
Laplacian score	600	0.172	0.374
ReliefF algorithm	600	0.221	0.456
FCBF	600	0.158	0.381
HSIC Lasso	600	0.280	0.5
SO	2698	0.322	0.550
SA	2678	0.306	0.556
GA	1155	0.300	0.616
MI/SA	238	0.410	0.890
MI/GA	293	0.446	0.870

TABLE V
RESULTS OF FEATURE SELECTION METHODS FOR THE DATA SET AE3

Method	subset size	precision	sensitivity
Baseline preselected genes	1790	0.087	0.171
MIM	139	0.144	0.483
Fisher's score	500	0.033	0.188
Laplacian score	600	0.027	0.126
ReliefF algorithm	600	0.034	0.170
FCBF	500	0.037	0.201
HSIC Lasso	600	0.067	0.241
SO	944	0.068	0.306

TABLE VI
RESULTS OF FEATURE SELECTION METHODS FOR THE DATA SET AE4

Method	subset size	precision	sensitivity
Baseline preselected genes	4021	0.98	0.875
MIM	40	1.0	0.982
MIFS	360	0.674	0.564
JMI	741	0.939	0.821
DISR	400	0.658	0.454
Fisher's score	1076	0.920	0.821
Laplacian score	1151	1.0	0.856
ReliefF algorithm	566	0.98	0.875
FCBF	789	0.927	0.678
HSIC Lasso	85	1.0	0.928
SO	1996	1.0	0.964
SA	1995	1.0	0.964
GA	1719	1.0	0.982
MI/SA	48	1.0	1.0
MI/GA	44	1.0	1.0

TABLE VII
RESULTS OF FEATURE SELECTION METHODS FOR THE DATA SET BIDDY-D0

Method	subset size	precision	sensitivity
Baseline preselected genes	3121	0.425	0.619
MIM	230	0.733	0.904
MIFS	350	0.151	0.396
JMI	900	0.313	0.821
DISR	500	0.174	0.378
Fisher's score	2408	0.5	0.597
Laplacian score	2598	0.468	0.622
ReliefF algorithm	1540	0.412	0.585
FCBF	2940	0.470	0.619
HSIC Lasso	950	0.478	0.688
SO	1603	0.451	0.711
SA	1575	0.410	0.715
MI/SA	207	0.654	0.954
MI/GA	559	0.726	0.957

TABLE VIII
RESULTS OF FEATURE SELECTION METHODS FOR THE DATA SET BIDDY-D0-2

Method	subset size	precision	sensitivity
Baseline preselected genes	2285	0.707	0.509
MIM	100	0.980	0.879
MIFS	900	0.682	0.536
JMI	750	0.619	0.464
DISR	680	0.643	0.474
Fisher's score	978	0.704	0.564
Laplacian score	1232	0.756	0.564
ReliefF algorithm	1276	0.750	0.526
FCBF	1695	0.75	0.579
HSIC Lasso	274	0.804	0.638
SO	1173	0.822	0.673
SA	1173	0.822	0.685
GA	1668	0.807	0.778
MI/SA	190	0.964	0.947
MI/GA	137	1.0	0.983

TABLE IX
RESULTS OF FEATURE SELECTION METHODS FOR THE DATA SET BIDDY-D6

Method	subset size	precision	sensitivity
Baseline preselected genes	2033	0.826	0.95
MIM	130	1.0	1.0
MIFS	400	0.818	0.818
JMI	250	0.667	0.8
DISR	250	0.826	0.864
Fisher's score	400	0.909	1.0
Laplacian score	300	0.818	0.857
ReliefF algorithm	300	0.909	1.0
FCBF	400	0.75	0.818
HSIC Lasso	103	0.909	1.0
SO	1449	0.826	0.864
SA	1426	0.826	0.864
GA	416	1.0	0.977
MI/SA	212	1.0	1.0
MI/GA	149	0.909	1.0

TABLE X
RESULTS OF FEATURE SELECTION METHODS FOR THE DATA SET BIDDY-D6-2

Method	subset size	precision	sensitivity
Baseline preselected genes	2033	0.458	1.0
MIM	40	1.0	1.0
MIFS	400	0.974	0.840
JMI	250	0.794	0.614
DISR	400	0.925	0.841
Fisher's score	800	0.975	0.886
Laplacian score	400	0.975	0.886
ReliefF algorithm	400	0.975	0.909
FCBF	400	0.972	0.795
HSIC Lasso	150	1.0	0.954
SO	1026	1.0	0.886
SA	1036	1.0	0.886
GA	670	1.0	0.841
MI/SA	198	1.0	1.0
MI/GA	138	1.0	1.0

TABLE XI
RESULTS OF FEATURE SELECTION METHODS FOR THE DATA SET BIDDY-D15

Method	subset size	precision	sensitivity
Baseline preselected genes	2033	0.458	1.0
MIM	25	1.0	0.972
MIFS	400	0.974	0.841
JMI	250	0.084	0.289
DISR	400	0.925	0.841
Fisher's score	800	0.975	0.886
Laplacian score	400	0.975	0.886
ReliefF algorithm	400	0.975	0.909
FCBF	400	0.897	0.795
HSIC Lasso	150	1.0	0.955
SO	1033	1.0	0.932
SA	972	1.0	1.0
GA	304	1.0	0.909
MI/SA	212	0.527	0.854
MI/GA	124	1.0	1.0

TABLE XII
RESULTS OF FEATURE SELECTION METHODS FOR THE DATA SET BIDDY-D15-2

Method	subset size	precision	sensitivity
Baseline preselected genes	6081	0.462	0.2906
MIM	146	0.863	0.901
MIFS	400	0.380	0.385
JMI	250	0.323	0.338
DISR	400	0.333	0.308
Fisher's score	600	0.388	0.244
Laplacian score	350	0.307	0.333
ReliefF algorithm	300	0.454	0.394
FCBF	400	0.370	0.310
HSIC Lasso	250	0.870	0.74
SO	3001	0.535	0.575
SA	3064	0.619	0.619
GA	857	0.407	0.578
MI/SA	182	0.831	0.988
MI/GA	335	0.898	0.946

TABLE XIII
RESULTS OF FEATURE SELECTION METHODS FOR THE DATA SET CD8

Method	subset size	precision	sensitivity
Baseline preselected genes	6679	0.697	0.329
MIM	41	0.770	0.780
MIFS	100	0.112	0.25
JMI	80	0.134	0.375
DISR	400	0.326	0.246
Fisher's score	150	0.245	0.333
Laplacian score	500	0.323	0.323
ReliefF algorithm	400	0.589	0.5
FCBF	250	0.333	0.339
HSIC Lasso	100	0.705	0.727
SO	3406	0.763	0.512
SA	3345	0.763	0.531
GA	2256	0.775	0.633
MI/SA	193	0.732	0.923
MI/GA	169	0.817	0.930

TABLE XIV
RESULTS OF FEATURE SELECTION METHODS FOR THE DATA SET L1210

REFERENCES

[1] Pedro Larrañaga Yvan Saeys, Iñaki Inza. A review of feature selection techniques in bioinformatics. *Bioinformatics*, 23:2507–2517, October 2007.

[2] Alex R Lederer and Gioele La Manno. The emergence and promise of single-cell temporal-omics approaches. *Current Opinion in Biotechnology*, 63:70–78, 2020. Nanobiotechnology Systems Biology.

[3] Kong W. Kamimoto K. et al Biddy, B.A. Single-cell mapping of lineage and identity in direct reprogramming. *Nature*, 564:219–224, 2018.

[4] Camargo FD Klein AM. Weinreb C, Rodriguez-Fraticelli A. Lineage tracing on transcriptional landscapes links state to fate during differentiation. *Science*, 2020.

[5] Lee Szeto G. Li J. et al Kimmerling, R. A microfluidic platform enabling single-cell rna-seq of multigenerational lineages. *Nat Commun*, 7:10220, 2016.

[6] Integrated analysis of multimodal single-cell data. *Cell*, 184:p.3573–3587, 2021.

[7] P.Baby K.Sasirekha. Agglomerative hierarchical clustering algorithm-a review. *International Journal of Scientific and Research Publication*, 2013.

[8] Ke Li Yuxuan Hu and Anran Meng. Agglomertive hierarchical clustering using ward linkage.

[9] Vicente Reyes-Puerta. A balanced tree cutting method for hierarchical clustering, December 2020.

[10] Jiliang Tang, Salem Alelyani, and Huan Liu. *Feature selection for classification: A review*, pages 37–64. CRC Press, January 2014.

[11] Will N. Browne Bing Xue, Mengjie Zhang. Particle swarm optimisation for feature selection in classification: Novel initialisation and updating mechanisms. *Applied Soft Computing*, 18:261–276, October 2014.

[12] Max Kuhn and Kjell Johnson. *Feature Engineering and Selection: A Practical Approach for Predictive Models*. Taylor Francis, 2019.

[13] Inês Soares, Joana Matos Dias, Humberto Rocha, Maria do Carmo Lopes, and Brigida Ferreira. *Feature Selection in Small Databases: A Medical-Case Study*, pages 808–813. 04 2016.

[14] Ming-Jie Zhao Mikel Luján Gavin Brown, Adam Pocock. Conditional likelihood maximisation: A unifying framework for information theoretic feature selection. *Journal of Machine Learning Research (JMLR)*, 2012.

[15] Patrick Emmanuel Meyer, Colas Schretter, and Gianluca Bontempi. Information-theoretic feature selection in microarray data using variable complementarity. *IEEE Journal of Selected Topics in Signal Processing*, 2(3):261–274, 2008.

[16] Quanquan Gu, Zhenhui Li, and Jiawei Han. Generalized fisher score for feature selection. *CoRR*, abs/1202.3725, 2012.

[17] Xiaofei He, Deng Cai, and Partha Niyogi. Laplacian score for feature selection. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems*, volume 18. MIT Press, 2006.

[18] Ryan J. Urbanowicz, Melissa Meeker, William La Cava, Randal S. Olson, and Jason H. Moore. Relief-based feature selection: Introduction and review. *Journal of Biomedical Informatics*, 85:189–203, 2018.

[19] Lei Yu and Huan Liu. Feature selection for high-dimensional data: A fast correlation-based filter solution. In T. Fawcett and N. Mishra, editors, *Proceedings, Twentieth International Conference on Machine Learning*, Proceedings, Twentieth International Conference on Machine Learning, pages 856–863, December 2003. Proceedings, Twentieth International Conference on Machine Learning ; Conference date: 21-08-2003 Through 24-08-2003.

[20] Makoto Yamada, Wittawat Jitkrittum, Leonid Sigal, Eric P Xing, and Masashi Sugiyama. High-dimensional feature selection by feature-wise kernelized lasso. *Neural computation*, 26(1):185–207, 2014.