Algorithms for Feature Selection: Structure Preservation, Scale Invariance, and Stability

By
Snehalika Lall

A thesis submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in *Computer Science* at *Indian Statistical Institute*

Supervisor
Prof. Sanghamitra Bandyopadhyay
Machine Intelligence Unit
Indian Statistical Institute
203 B. T. Road, Kolkata-700108
December, 2022

Dedication

To my family and supervisor

Acknowledgements

Finally, it is time for me to acknowledge all those who inspired me, supported me and helped me to get to the place where I am today.
I take this opportunity to express a deep sense of gratitude to Prof. (Dr.) Sanghamitra Bandyopadhyay for her supervision and invaluable co-operation. This thesis would not have been possible without constant inspiration and unbelievable support of her over the last five years.
I am also grateful to Dr. Debarka Sengupta, Dr. Abhik Ghosh, and Dr. Sumanta Ray for their external support to understand biology and statistics.
I have had an amazing group of Labmates. Each of them deserves my gratitude: Monalisa Pal, Sucheta dawn, Suparna Saha, and Debajyoti Sinha. Working with them was a great experience that brought together several good and fruitful ideas. Thank you to all of you for the unselfish help, insights, feedback, and for making the job search a collaborative effort.
I would also like to register my heartfelt gratitude to all of my teachers who taught me during Ph.D. coursework and always inspired me to go beyond the limit.
Finally, most important of all, I would like to dedicate the thesis to my parents Mr. Arun Kumar Lall and Mrs. Kanchana Lall, to honor their love, patience, and support during my research. I would also express my appreciation to my husband Dr. Sumanta Ray and my sister Ms. Nikita Lall for their unwavering support and love.

Date: July, 2022 (Snehalika Lall)

List of Publications

Papers in Journals

- **Snehalika Lall**, Debajyoti Sinha, Sanghamitra Bandyopadhyay, Debarka Sengupta, Structure-aware principal component analysis for single-cell rna-seq data, **Journal of Computational Biology**, 25, 1365-1373, 2018.

- **Snehalika Lall**, Abhik Ghosh, Debajyoti Sinha, Debarka Sengupta, Sanghamitra Bandyopadhyay, Stable feature selection using copula, **Pattern Recognition**, Volume 112, April 2021, 107697.

- Sumanta Ray, **Snehalika Lall**, Sanghamitra Bandyopadhyay, CODC: A copula based model to identify differential coexpression, **NPJ System Biology and Application**, 6, 1-13, 2020, https://www.nature.com /articles/s41540-020-0137-9 .

- **Snehalika Lall**, Sumanta Ray, Sanghamitra Bandyopadhyay, RgCop-A regularized copula based method for gene selection in single cell rna-seq data, **PLoS Computational Biology**, 17(10): e1009464. https://doi.org/10.1371/ journal.pcbi.1009464, 2021.

- **Snehalika Lall**, Abhik Ghosh, Sumanta Ray, Sanghamitra Bandyopadhyay, sc-REnF: An entropy guided robust feature selection for clustering of single-cell rna-seq data, **Briefings in Bioinformatics**, Volume 23, Issue 2, March 2022, bbab517, https://doi.org/10.1093/bib/bbab517

- **Snehalika Lall**, Sumanta Ray, and Sanghamitra Bandyopadhyay, LSH-GAN enables in-silico generation of cells for small sample high dimensional scRNA-seq data., **Nat. Communication Biology**, 577 (2022). https://doi.org/10.1038/ s42003-022-03473-y.

- **Snehalika Lall**, Sumanta Ray, Sanghamitra Bandyopadhyay, A topology preserving graph convolution network for clustering of single-cell RNA seq data, **PLoS Computational Biology**,18(3): e1009600. DOI: https://doi.org/ 10.1371/journal.pcbi.1009600, 2022.

- Sumanta Ray, **Snehalika Lall**, and Sanghamitra Bandyopadhyay, A deep integrated framework for predicting SARS-CoV2–Human protein-protein interaction, **IEEE Transactions on Emerging Topics in Computational Intelligence**, doi: 10.1109/TETCI.2022.3182354., 2022

List of Presentations in National/International Conference:

- Snehalika Lall, Sanghamitra Bandyopadhyay, A l1 -Norm Regularized Copula based Feature Selection **ISCSIC 2019: Proceedings of the 2019 3rd International Symposium on Computer Science and Intelligent Control(ACM)**, September 2019 Article No.:30 Pages 1–6,2019.
- Snehalika Lall, Sumanta Ray, Sanghamitra Bandyopadhyay, Identifying novel SARS-CoV2–human protein interactions using graph embedding, **28th International Conference on Intelligent Systems for Molecular Biology (ISMB /ECCB)**, COVID-19 special track 2020.
- Snehalika Lall, Sumanta Ray, Sanghamitra Bandyopadhyay, and A Schönhuth, Deep variational graph autoencoders for novel host-directed therapy options against COVID-19, **29th Conference on Intelligent Systems in Molecular Biology ISMB/ECCB 2021, COVID-19 special track.**

Abstract

With the advancement of science and technology, data has increased both in sample size and dimension. Examples of high-dimensional data include genomic data, text data, image retrieval, bioinformatics, etc. One of the major problems in handling such data is that all the features are not equally important. Hence, feature engineering, feature selection and feature reduction are considered important pre-processing tasks to discard redundant, irrelevant features while preserving the prominent features of the data as much as possible. Feature selection, in practice, often improves the accuracy of down-stream machine learning problems, including clustering and classification.

In this thesis, we aim to devise some novel and robust feature selection mechanisms in diverse domains of applications with a special focus on high dimensional biological data such as gene expression and single cell transcriptomic data. We develop a series of feature selection techniques equipped with structure-aware data sampling at its core. We adopt several concepts from statistics (e.g. copula and its variant), information theory (entropy), and advanced machine learning domain (variational graph autoencoder, generative adversarial network, and its variant) to design the feature selection models for high dimensional and noisy data. The proposed models perform extremely well both in supervised and unsupervised cases, even if the sample size is very low. Important outcomes from all the proposed methods are discussed in chapters. Moreover, an overall discussion about the applicability along with a brief mention of the shortcomings of all the discussed methods is provided. Some suggestions and guidance are provided to overcome the disadvantages which direct the future scope of improvement of all the devised methods.

Contents

List of Figures

List of Tables

List of abbreviations

Notation	Explanation
FS	Feature Selection
CBFS	Stable Copula Based Feature Selection
RgCop	Regularized Copula based Feature Selection
GAN	Generative Adversarial Network.
LSH	Locality Sensitive Hashing.
LSH-GAN	Locality Sensitive Hashing based GAN
PCA	Principal Component Analysis.
LSPCA	Locality Sensitive PCA
CODC	A copula based model to identify differential coexpression
sc-REnF	Robust entropy based feature selection for single cell data analysis
sc-CGconv	Copula based Graph Convolution Network for Single cell Clustering.
DNN	Deep Neural Networks
HDSS	High Dimensional Small Sample
C(.)	Copula Function
G(.)	Generator
D(.)	Discriminator
F(.)	Marginal Distribution Function
$D_{m\times n}$	Data Matrix with m number of samples and n number of features.
F	Total Feature Set.
S	Selected Feature subset.
G	Total Gene set
G_s	Selected gene subset.
X,Y,Z	Random Variables
t-SNE	t-Distributed Stochastic Neighbour Embedding (t-SNE)
UMAP	Uniform Manifold Approximation and Projection (UMAP)

1 Introduction and Scope of the Thesis

1.1 Introduction

Recent technological advances in biomedical engineering produce large volumes of biological data, analysis of which is extremely important for many medical and biological applications, including disease diagnosis, biomarker discovery, drug development, and forensics. Generally, such datasets are high-dimensional (i.e., they have huge number of features) and contain complex nonlinear patterns. Examples of such high-dimensional data include genomic data [2], text data [3], image retrieval [4], bioinformatics [5], etc. To discover the hidden patterns, machine learning techniques such as genome-wide association studies, gene selection, and dimensionality reduction techniques have been successfully applied. Recently nonlinear models, in particular, deep neural networks (DNNs) are emerging as a potential supplement for machine learning tools to analyze the hidden complex patterns within voluminous datasets. Unfortunately, these models are difficult to train because of the significantly high number of parameters. Hence, the following two questions naturally arise:

1. is all the features equally important/necessary to build an effective prediction model?

2. is it possible to modify the existing machine-learning methods to efficiently process such high-dimensional data?

The answer to the first question leads to the development of efficient feature selection (FS) methods, while the second question brings out the necessity of modifying the existing machine learning models. In this thesis, we address these two challenges, independently and in combination.
Identifying important features is a persistent problem in machine learning, which is generally known as the feature selection (FS) problem. The process generally consists of identifying a smaller subset of relevant features (i.e., smaller than the original dataset) that contains relevant features such that the subset retains

the predictive capability of the data/model while eliminating the redundant or irrelevant features. The application of FS techniques in bioinformatics is often treated as an indispensable step before model building. A plethora of FS techniques exist for analyses of high-dimensional models in bioinformatics.
Here, we focus on developing novel FS techniques for high-dimensional biological data modeling. Unlike other dimensionality reduction techniques which are based on projection (e.g., principal component analysis) or based on information (e.g., using information theory), the FS technique does not change the original structure of the variables, instead selects a subset from the full set of variables. Hence, FS techniques can retain the original structure and semantics of the variables.

1.2 Feature Extraction and Feature Selection Methods: A Background Study

In today's world, dimensionality reduction is often considered to be one of the most important tasks for extracting knowledge from pre-processed data, where the data may be representative of a gene-gene interaction network, a disease network, a social network, etc. The dimensionality of data can be reduced through two techniques, viz., feature extraction and feature selection.
The task of feature extraction [6] is to transform the feature space into a more informative space based on several transformations or combinations of the original features. It is therefore, a mapping from higher-dimensional feature space to a lower-dimensional one from which the most relevant information about the underlying recognition tasks can be retrieved more efficiently and rapidly. These new reduced sets of features should then be able to summarize most of the information contained in the original set of features. Principal component analysis [7], independent component analysis [8], t-Distributed Stochastic Neighbour Embedding (t-SNE) [9] are widely used methods in the area of feature extraction.
In general, datasets are assumed to contain some redundant, irrelevant, or noisy features, the inclusion of which do not offer any extra benefit while searching for valuable knowledge in those datasets. Thus, feature selection is often adopted to discard such redundant, irrelevant features, while at the same time preserving the prominent features of the data as much as possible. Depending on the availability of the class labels, feature selection can also be grouped into two categories: supervised and unsupervised feature selection [10]. Unsupervised feature selection does not require class label information but supervised feature selection requires class label information. Major parts of the earlier developed feature selection algorithms are based on supervised learning, whereas a comparably smaller amount of research has been accomplished in the direction of unsupervised feature selection. In the cases of supervised approaches, several statistical measures

are based on t-test [11], chi–square test [12], Wilcoxon Mann–Whitney test [13], mutual information [14], Pearson correlation coefficients [15], etc. On the other hand, Unsupervised Feature Selection using Feature Similarity measure (UFSFS) [16], Laplacian Score for Feature Selection (LSFS) [17], Spectral Feature Selection (SPFS) [18], and Multi–Cluster Feature Selection (MCFS) [19] are some of the most recognized algorithms.

1.2.1 Feature Extraction Methods

In this section, some widely used feature extraction methods are discussed.

Principal Component Analysis (PCA)

—It is the most widely used linear dimensionality reduction technique. PCA [7] takes the original data as input and tries to find out a combination of the input features that can best summarize the original data distribution. PCA operates by maximizing variances and minimizing the reconstruction error by looking at pair wise distances. In PCA, the original data is projected into a set of orthogonal axes and each of the axes is ranked in terms of decreasing eigenvalues.

t-Distributed Stochastic Neighbour Embedding (t-SNE)

—Like other dimensionality reduction methods, t-SNE [9] generates a 2-dimensional (or 3D) visualization of data that allows close similarities between samples. It strives to retain the proximity of similar samples while keeping the dissimilar samples at a distance. t-SNE's has the ability to control the trade-off between local and global relationships among points. Due to its non-linearity, it usually generates more visually-compelling clusters when compared with the other methods [20]. It is widely applied to the transcriptomic data as well as other large high-dimensional datasets such as single cell data to produce 2-dimensional visual representations [21].

Uniform Manifold Approximation and Projection (UMAP)

—-Similar to t-SNE, Uniform Manifold Approximation and Projection (UMAP) [22] is a non-linear dimension reduction technique. However, it is a more recent method developed by McInnes et al. (2018), a few advantages over t-SNE . Unlike t-SNE, UMAP is scalable on large datasets. It can preserve the global structure of the data, and it is more memory efficient. UMAP is built upon mathematical foundations on Laplacian eigenmaps [23]. It uses a combination of Riemannian geometry and algebraic topology [24]. It assumes that the data is uniformly

distributed on a Riemannian manifold, the Riemannian metric is locally constant, and the manifold is locally connected. With these assumptions, the manifold is modelled with a fuzzy topological structure. The embeddings are constructed from searching for a low dimensional projection of the data that has the closest possible equivalent fuzzy topological structure.

Locality Sensitive Hashing (LSH)

— Hashing is a technique to index and retrieve data records. By incorporating special hash functions, data is retrieved efficiently by representing the data values as smaller keys. The unique keys correspond to hash 'buckets' and often map to unique memory locations. During a search of a given record, the hash function computes the respective key and the corresponding map to the memory location. This mechanism enables quick access to a searchable record typically in O(1) time. The efficiency of mapping depends on the hash function used. Ideally, hash collisions, where multiple keys map into the same location, are discouraged, but for specific problems, collisions are beneficial.
In contrast to the usual hashing techniques, LSH encourages the collision of records that are similar to each other [25]. The chosen hash function is such that the nearest neighbors of each point (single data record) have keys mapping to the same bucket. When searching for k-nearest neighbors (k-NN) of a new record, the bucket corresponding to the new record may contain sufficient points to qualify as neighbors. Traditionally, to search for k-nearest neighbors, all the remaining points had to be investigated for similarity. This simple scheme is not only time consuming, but it also becomes infeasible for large datasets.
For a hash function based on spatial mapping of the data values, projection of each data point on random hyperplanes is used. The dimension of the hyperplane is the same as the dimension of the data point. Each encodes 0/1 depending on which side of the hyperplane the projection lies. For a set of h such random hyperplanes, a h bit encoded binary string is generated. Each of these strings is designated as the hash key for respective data points. Computing the projection is a simple vector dot product (See Fig. 1.1, taken from web source). During the query, the same set of hyperplanes is used to obtain the hash key. This operation is carried out in constant time. The nearest neighbors are searched for within the candidates sharing the same hash key. Thus, LSH allows querying the k-NN for all the data points in a time-efficient manner.
It is evident that the hash function discussed above are stochastic in nature which makes the k-NN search an approximate process. Therefore, in order to improve the accuracy, the system of hyperplane based encoding is repeated over multiple sets of hyperplanes. This allows us to obtain a greater number of candidates to search for neighbours. This process is described in the more advanced version of

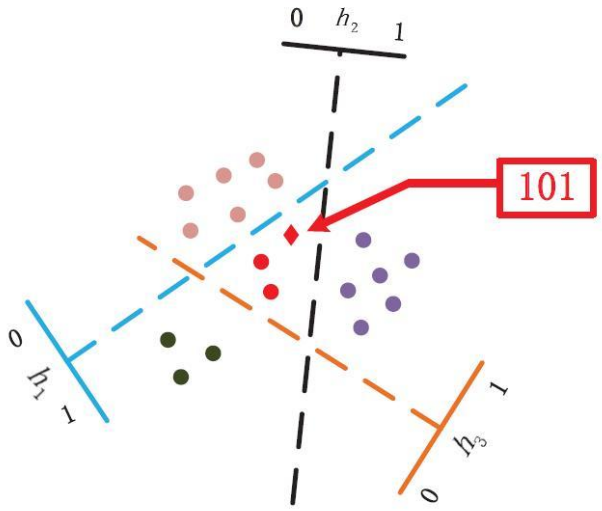


Figure 1.1: Hash function defined by a system of three hyperplanes: h1, h2, h3, each point is a three bit binary encoded string after projection.

LSH known as LSH Forest [26]. For a given query, the index is developed such that only a small set of "candidate" objects is retrieved for comparison with the query. In an attempt to address the limitations of the simple LSH, LSH Forest uses a B+ tree indexing scheme on an ensemble of several hash functions without affecting the retrieval time.

1.2.2 Feature Selection Methods

FS methods are of three broad types - *filters* [27], *wrapper* models [28] and *embedded* techniques[29]. Filters typically measure the association between explanatory variables and the dependent variable. Some of these association measures are -

Pearson's correlation coefficient [30]

— It measures the linear dependence between two random variables. It is defined as the covariance between two variables, divided by the product of their standard deviations. It can be treated as a normalised measurement of the covariance, such that the result always has a value between −1 and 1. Given paired data $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ consisting of n pairs, the Pearsons correlation coefficient, r_{xy} can be defined as

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{1.1}$$

where n is sample size x_i, y_i are the individual sample points indexed with i, $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ (the sample mean); and analogously for $\bar{y}$.

Chi (χ)-squared test [31]

—-A chi-square test is used in statistics to test the independence of two events. For a random variable $(x_1, x_2, \cdots, x_n)$,

$$\chi^2 = \sum_{i=1}^{n} \frac{(x_i - m_i)^2}{m_i} \tag{1.2}$$

where, x_i and m_i are the observed value, and the expected value of the i^{th} sample respectively.
When two features are independent, the observed value (x_i, y_i) is close to the expected value (m_{xi}, m_{yi}), thus we will have smaller Chi-Square value. So high Chi-Square value indicates that the hypothesis of independence is incorrect.

Mutual information [32]

—It is a measure that is often used in filter-based feature-selection methods. Mutual Information based FS methods simultaneously assess both the relevance of a subset of features and the redundancy concerning other feature variables. For two variables X and Y, the mutual information $I(X;Y)$ is defined as

$$\begin{aligned} I(X;Y) =& H(X) - H(X|Y) \\ =& \sum_{x \in X, y \in Y} p(x,y) \frac{\log p(x,y)}{p(x)p(y)} \end{aligned} \tag{1.3}$$

where H(X) is the marginal entropy of X and H($X|Y$) is the conditional entropy of X given Y.
Some widely used wrapper approaches are - Mutual information-based Feature Selection (MIFS), Conditional Mutual Information Maximization (CMIM), Minimal-Redundancy-Maximal-Relevance (mRMR). These are mutual information based FS approaches to find a subset of relevant features. There are also genetic algorithms based FS approaches [33, 34, 35], where a hybrid genetic algorithm is adopted to find a subset of features most relevant to the classification task.
Embedded methods are slightly different from wrappers. The embedded FS [36] techniques take the advantage of both the wrapper and filter-based approaches. Zhang et.al. [37] proposed the horseshoe regularization penalty for feature subset selection, demonstrating its theoretical and computational advantages. Wang et.

al. [38] proposed a novel unsupervised feature selection algorithm EUFS, which embeds feature selection into a clustering algorithm via sparse learning without the transformation. Liu et. al. [39] developed an embedded feature selection method using weighted Gini index (WGI).

1.2.3 Entropy based Correlation Measure for Feature Selection

This subsection describes some concepts of entropy and mutual information, and explains the reasons for employing them in FS. The entropy of a random variable can be described as a measure of its uncertainty. It is also a measure of the average amount of information needed to describe the random variable [40].
Here, some entropy measures and their applications in FS are discussed.

Shannon Entropy

— For a random variable X, the Shannon entropy is defined as

$$H(X) = -\sum_{x \in X} p(x) \log p(x). \tag{1.4}$$

For more than one variables, one can suitably construct the joint or the relative entropy measures. For example, the Shannon conditional entropy of the random variable X given random variable Y is defined as

$$H(X|Y) = -\sum_{x \in X, y \in Y} p(x, y) \log p(x|y). \tag{1.5}$$

Renyi **Entropy**

— It was first developed by *Alfred Renyi* [41] in the context of information science. The *Renyi* entropy of the random variable X is defined in terms of a positive real number q, with $q \neq 1$, as

$$H_q(X) = \frac{q}{1-q} \log \left(\sum_x (p(x))^q \right)^{1/q}, \quad q \neq 1. \tag{1.6}$$

Interestingly, note that, this *Renyi* entropy reduces to the Shannon entropy when $q \to 1$. It can also be extended for the two random variables X and Y; their joint *Renyi* entropy is given by

$$H_q(X, Y) = \frac{q}{1-q} \log \left(\sum_{x,y} (p(x, y))^q \right)^{1/q}, \quad q \neq 1. \tag{1.7}$$

Tsallis **Entropy**

— It is another generalization of Shannon entropy developed from the context of statistical mechanics that yields q-normal distribution as an equilibrium probability distribution [42]. Mathematically, the *Tsallis* joint entropy of two random variables X and Y is defined in terms of a tuning parameter $q > 0$ as:

$$H_{T_q}(X, Y) = \frac{1}{q-1}\left[1 - \sum_{x,y}(p(x, y))^q\right]. \tag{1.8}$$

It also coincides with the Shannon entropy as $q \to 1$.
Renyi and *Tsallis* entropies [43] have interesting characteristics. An appropriate choice of the tuning parameter q, in either of the two entropies, makes them less sensitive (more robust) against different noises present in the data. Therefore, the use of these Renyi and Tsallis entropy strengthens the robustness of our objective function proposed for feature selection. Although these entropies are widely-researched, but its application in the single-cell domain is less explored [44]. In a part of the present thesis the utility of Renyi, and Tsallis entropies for analysing single cell data is explored.
Mutual Information-based filter methods have gained popularity due to their ability to capture the non-linear association between dependent and independent variables in a machine learning setting. Mutual information-based Feature Selection (MIFS) is among the earliest algorithms in this segment [45]. It is a greedy algorithm that considers both mutual information of a candidate feature with class label information and the prior selected features. The objective function (J_{MIFS}) of MIFS is defined as

$$J_{MIFS}(X_k) = I(X_k; Y) - \beta \sum_{X_j \in S} I(X_j, X_k), \tag{1.9}$$

where, X_k is the feature under consideration, S is the already selected feature subset, Y is the class label and β is a configurable parameter.
Conditional Mutual Information Maximization (CMIM) [46] maximizes mutual information concerning the class while conditioning upon the selected feature. The objective function (J_{CMIM}) of CMIM is defined as

$$J_{CMIM}(X_k) = I(X_k; Y) - \max_{X_j \in S}(I(X_k; X_j) - I(X_k; X_j|Y)), \tag{1.10}$$

where, X_k is the feature under consideration, S is the already selected feature subset, and Y is the class label.

One of the popular mutual information-based approaches is the Minimal-Redundancy-Maximal-Relevance (mRMR) method [14], which considers feature relevance with respect to the class labels and also ensures that redundant features are not present in the final feature subset. The objective function (J_{mRMR}) of mRMR is defined as

$$J_{mRMR}(X_k) = I(X_k; Y) - \frac{1}{|S|} \sum_{X_j \in S} I(X_k; X_j). \tag{1.11}$$

Another mutual information based FS method is Joint Mutual Information Maximisation (JMIM) [47]. This method overcomes the limitations of the filter based FS approaches. It introduces a new goal function which is based on joint mutual information and the 'maximum of the minimum' nonlinear approach. The objective function (J_{JMIM}) of JMIM is defined as

$$J_{JMIM}(X_k) = \arg \max_{X_k \in (F-S)} \left[\min_{X_j \in S} I(X_k; X_j; Y) \right]. \tag{1.12}$$

1.3 Preliminaries of Molecular Biology

Cells, in biology, can be treated as a basic membrane-bound unit which is the fundamental unit of life. A single cell is a complete organism in itself, such as a bacterium or yeast. Other cells cooperate with other specialized cells and become the building blocks of large multicellular organisms, such as humans and other animals.

A protein is built following a 'recipe' or a set of instructions for protein production, encoded in the genetic material of the organism known as the Deoxyribonucleic acid (DNA). The same DNA is replicated in all cells of the individual through cell division. The DNA is a thread-like chain of units called nucleotides. Each nucleotide is composed of one of the four nitrogen-containing nucleobases Adenine (A), Thymine (T), Cytosine (C), or Guanine (G), a sugar called deoxyribose, and a phosphate group. It is the variations of the DNA sequence that make each individual unique. The DNA is represented as a sequence of the initial characters of the four nucleobases: A, T, C or G. Long chains of nucleotides are tightly packed into structures called chromosomes (see Fig. 1.2, taken from web source). A gene is a specific sequence of nucleotides at a given position on a given chromosome that encodes for a specific protein or another molecule called Ribonucleic Acid (RNA). It is also considered the basic physical and functional unit of heredity. Specific genes are 'activated' in different cell types which specify proteins or other functional RNAs.

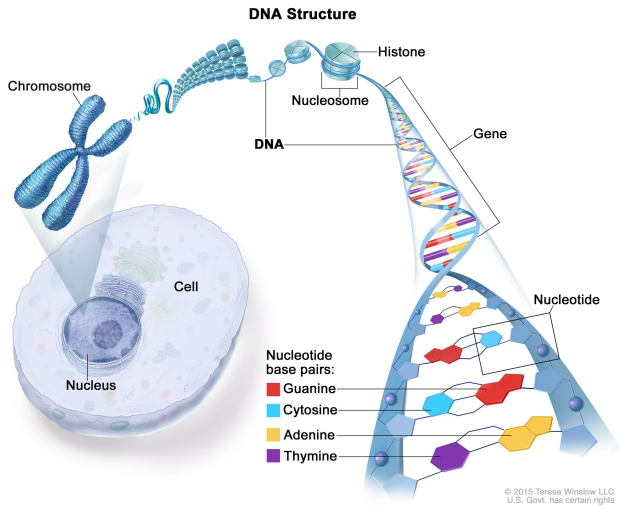


Figure 1.2: DNA, Chromosome, and Gene

1.3.1 The Central Dogma

The 'Central Dogma' is the fundamental process of protein synthesis. It was first proposed in 1958 by Francis Crick, the discoverer of the structure of DNA. The central dogma of molecular biology defines the process of conversion from DNA to RNA, to make a functional product, a protein. The process has two fundamental steps: transcription and translation (see Fig. 1.3,taken from web source).
In the transcription stage, the information of the DNA gets converted into small, portable RNA messages. These messages travel to the ribosomes where they are 'read' and translated to specific proteins. The central dogma states that the pattern which occurs most frequently in the cells is

- existing DNA to new DNA through replication.
- new RNA from DNA through transcription.
- new proteins from RNA through translation.

1.3.2 Next-generation Sequencing

Uninterrupted breakthroughs in genomics technologies have generated a wealth of mineable biological "big data". A specific genre of bioinformatic problems deals with datasets that are curated from biological material through a process called genome sequencing. The revolution in sequencing technology has brought down the cost of sequencing the genome. Compared to the more primitive microarray

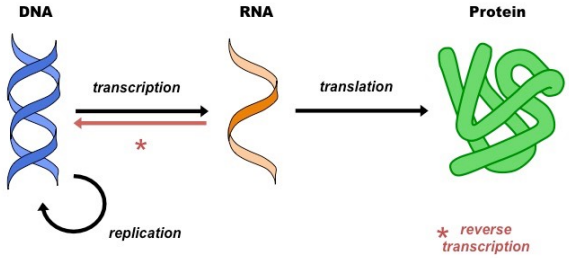


Figure 1.3: Diagrammatic view of transcription and translation processes in the central dogma

technology, next-generation sequencing (NGS) has opened the flood gates of NGS-derived omics datasets and a gold mine for knowledge discovery. Current omics data on whole-exome sequencing, RNA sequencing or methylation sequencing are being produced in vast quantities. The raw data generated by a sequencer often exceeds a few gigabytes, even for a single sample. For example, in RNA sequence data, the raw data is processed into expression matrices where each row represents a single sample having $30,000$ features.

Transcriptome data

— Measuring RNA abundance in a sample at a given moment is performed using next-generation RNA sequencing technology (RNA-seq). It allows an unbiased, high-throughput analysis of all transcripts the complete set of transcripts in a specific type of cell or tissue. In addition to mRNA transcripts, RNA-Seq can be used to look at different populations of RNA including total RNA, and small RNA, such as miRNA, tRNA, and ribosomal profiling. Unlike microarrays of the previous generation which often have a limited dynamic range, and rely on hybridization, RNA-seq analysis allows detection of low abundance transcripts and has very low background signals. It, therefore, has a much larger dynamic range and allows the transcriptome to be sequenced at higher coverage in high throughput and quantitative manner. This technology is not based on a priori knowledge of targets and is advantageous for the discovery of new transcripts, as it does not rely on known genomic sequences.

1.3.3 Common Transcriptomic Assays

Bulk RNA sequencing

— Over the last decade, the advancement of new technology enabled genome-wide profiling of DNA, RNA, protein and epigenetic modifications within individual cells [48]. Bulk RNA sequencing measures the average expression level for each

gene across a large population of input cells. It is useful for identifying common expression patterns from the transcripts of multiple individuals. For example, in disease studies, analysis involves finding the different expression patterns in the two groups of data, a diseased group, and a control group. However, bulk RNA sequencing is insufficient for studying heterogeneous systems, e.g., early development studies or complex tissues of the brain.
Functional genomics deals with the analysis of large datasets obtained from various biological data sources using large-scale experimental approaches. Examples include the simultaneous monitoring of expression levels of thousands of genes under a particular condition, which is termed gene expression analysis. Microarray technology makes it possible to quantify the expression of genes on a large scale. During the last few decades, the advancement of DNA microarray analysis invent a new line of research both in bioinformatics and machine learning. This technology produces data that is used to collect information from tissue and cell samples. A typical challenge in this data is to classify samples to separate healthy patients from cancer patients based on their gene expression profiles (binary approach). Another way to analyze the data is to distinguish among different types of tumors (multiclass approach). Experimental complications such as noise and variability render the analysis of microarray data an exciting domain for the machine learning researcher [49].

Single cell RNA sequencing

— A more recent Single Cell RNA sequencing (scRNA-seq) technology enables the screening of genome-wide transcription profiles of tissue at the resolution of a single-cell. Single-cell RNA sequence data promises to understand the dynamics of diseases at the resolution of a single cell. It is now possible to identify the heterogeneity in cell-types, the discovery of rare subtypes, and their roles in specific pathways/diseases. With the evolution of single-cell transcriptome technology, it is now possible to capture the RNA landscape (transcriptome) of individual cells to discover new cell types and study their functions. It is therefore also possible to obtain a higher resolution picture of organ development or disease processes. It can be estimated to witness a sharp increase in the average sample size of future investigations through the availability of affordable commercial platforms. Recent work produced an unprecedented $250k$ single-cell expression profile as part of a single study. This gives us an idea about the scale of future single-cell experiments. The number of genes mapped is the same as that for bulk RNA-seq. Until a few years ago, only a low-resolution molecular dissection could be obtained [50].
A schematic view of scRNA seq is given in Fig. 1.4 (taken from web source).

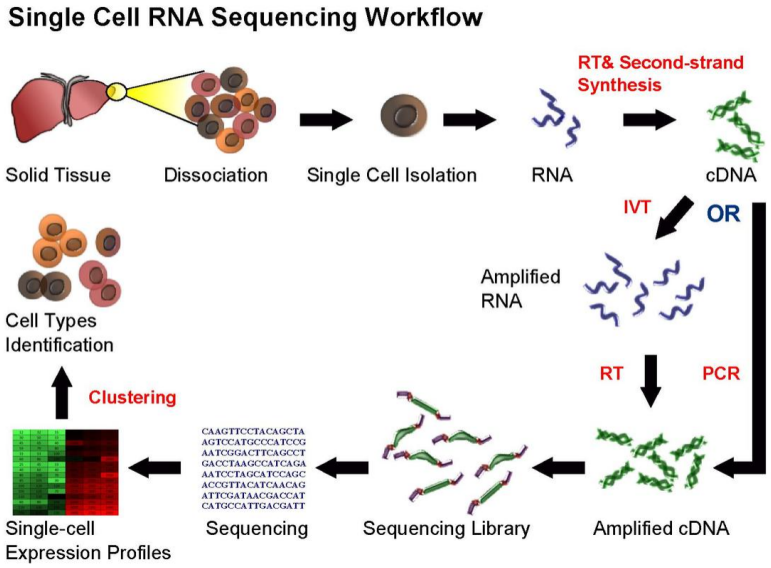


Figure 1.4: Diagrammatic view of single cell RNA transcriptome sequence.

1.3.4 Dimension Reduction and Feature Selection Techniques in Single Cell RNA Sequence Data

FS in single cell RNA seq data can be treated as identifying most relevant 500–2000 genes from the preprocessed *cell* × *gene* expression matrix. The usual way to select genes is by computing either their coefficient of variation (highly variable genes [51]) or average expression level (highly expressed genes) across all cells [52].

Starting from raw counts, scRNA-seq data analysis typically goes through the following steps before clustering: i) normalization, ii) feature selection, and iii) dimensionality reduction. While normalization/log-normalization adjusts the differences between the samples of individual cells and reduces the skewness of the data, feature selection seeks to identify the most relevant features (genes) from the large feature space. There exist several methods for normalization followed by feature (gene) selection [53, 54, 55, 56]. As an example, sctransform [53], defined within the Seurat package [57], constructs a regression model between gene expression and (cell) total counts, and uses the Pearson residuals of the model as normalized data. In Linnorm [54], the expression levels are adjusted by using a normality based normalizing transformation method. The top genes are identified from the normalized expression data by using diversified procedures. Scanpy used several dispersion based methods [58, 59] for selecting highly variable genes (HVG). In Seurat package [57], standardized variance is calculated from the normalized data to find out the HVGs. Principal component analysis (PCA) [60] is treated as the most popular dimension reduction technique, which is utilized in

Seurat and scanpy.
Although there is a plethora of methods [61, 62, 63, 58] available for performing each task within the pipeline, the standard approaches considers a common sequence of steps for the preprocessing of scRNA-seq data [64]. This includes normalization by scaling of sample-specific size factors, log transformation, and feature selection by using coefficient of variation (highly variable genes[65, 66]) or by using average expression level (highly expressed genes). Alternatively, some methods exist for gene selection, such as GLM-PCA [64] selects features (genes) by ranking genes using deviance, M3Drop [63] selects genes leveraging the dropouts effects in the scRNA-seq data.

1.3.5 Challenges in Single Cell RNA Sequencing

There exist several challenging themes which are common in all single-cell analyses, irrespective of the particular assay or data modality generated [67]. In this thesis, our objective is to address some of these challenges specific to the primary and later stages of downstream analysis of the single-cell data. Based on the conventional procedure of downstream analysis (see subsection 1.3.4), here we outline some of the challenges that are most common and yet to be unleashed by the existing approaches.

Variable gene selection in preprocessing step

— The selection of top genes has a good impact on the cell clustering (or classification) process in the later stage of downstream analysis. A good clustering (or classifying cell samples) can be ensured by the following characteristics of features/genes: the features/genes should have useful information about the biology of the system, while not including features containing any random noise. Thus the selected genes reduce the dimension of the while retaining the useful biological structure, reducing the computational cost of later steps.
The performance of downstream analysis, mainly the clustering process, is heavily dependent on the quality of selected top features/genes. The typical characteristics of good features/genes are: i) it should encode useful information about the biology of the system, ii) should not include features that contain random noise, and iii) preserve the useful biological structure while reducing the size of the data to reduce the computational cost of later steps.

Detection of rare cellular identities using supervised technique

— To overcome unsupervised clustering problem, we need methods that automatically determine cell labels. *Supervised learning* based approaches [68] addresses this by performing automatic and hassle free cell type detection, where labelled

data is given as input. Although there exits a plethora of methods [69] which address the problem of cell type detection using supervised approaches, it is hard to identify rare cell (or subtype of a major cell) from a given cell population. Recently Wegmann et al. [70] demonstrate that most of the conventional technique (supervised/unsupervised) meant for identifying cell populations shows good performance in identifying populations defined by more than 2% of total cells. Until now there exist a very few dedicated tools [70, 71] which can identify rarer populations. Note that, such rare types may be the ones of major interest in single cell typing, because in bulk experiments rare types yield signals that one easily confounds with noise. Yet, none of the supervised methods exists that could accurately identify poorly covered cells because of the little representation of the rare cell population in the training data.

Simulate realistic cell samples of poorly covered cells

— High dimensional small sample (HDSS) data is prevalent in the single cell domain due to the budgetary constraint of single cell experiments or simply because of the small number of available patient samples. Whatever the reason is, too few observations (cell sample) in the single cell data may create problems in the downstream analysis. This is because a small sample size may not reflect the whole population very well, which surely hinders any model to perform accurately. Recently computational researchers gaining interest in this field. Some methods like cscGAN [72], Splatter [1] are already developed and use different techniques (like the generative model, and statistical framework) to successfully simulate the samples of specific cell types or subpopulations. The challenge in this task is to handle the sparsity and heterogeneity of the cell populations which define the specific characteristics scRNA-seq data.

Finding out a set (or combination) of optimal markers

— Conventional approaches for marker selection only allow the identification of markers that distinguish particular cell labels from all of the other labels (i.e. one-vs-all). In these techniques, markers are treated as differentially expressed (DE) genes across two groups, and are identified by comparing within-group expression with across-group expression using a statistical test. The popular and widely used scRNA-seq analysis tools such as Seurat V3/V4 [73] and Scanpy [74] often used these methods for differential gene analysis using the Wilcoxon Rank sum test after clustering of the cells.
Recently Avermann et al. [75] demonstrate that the ranked set of DE genes cannot be used to find out the individual marker or the combination of markers.The 'ideal marker gene' must show a 'binary expression' pattern [75]. These genes should

have high expression levels in all individual cells of a given type, and not be expressed in the cells of other types. Most of cases these genes are expressed at a high level in the target cell cluster, but maintain lower (measurable) expression in other cell clusters. This presents problems in many downstream analyses such as in RT-PCR, or spatial transcriptomics analysis.
Identification of 'ideal markers' is not much explored in the literature. A single binary marker may not be available for a cell cluster of a specific type. This requires the identification of a combination of markers for optimal classification.

1.4 Copula Measure in Feature Selection

Copula based multivariate dependency is not much explored in feature selection literature. In this thesis, we leverage the dependency measure in the feature selection domain of the single cell RNA-seq data. In the next and subsequent paragraph, a brief introductory note about copula and its applications is provided.

1.4.1 Copula: Background and Preliminary Definitions

Since the introduction of Copula in the early '50s, a considerable surge of interest has been noticed in applying it in several fields such as applied mathematics, finance, time-series data analysis, portfolio optimization, bioinformatics, and many more. Copulas are mathematical objects which describe the dependence structure between random variables. It allows to model and estimates the marginals of a random vector, thus offering great flexibility to build a multivariate stochastic model. The name 'Copula' comes from a Latin word *copulare* which means 'joins together'. The word is first used by famous statistician *Sklar* in 1959 in one of his famous 'Sklar's Theorem'. The copula is extensively used in high dimensional data applications to obtain joint distributions from random vectors, easily by estimating their marginals. Apart from finance, applied mathematics, and other interdisciplinary fields, recently copulas are applied in bioinformatics as well. In Kim (2008) [76], dependencies between genes are measured using copula, which overcomes the difficulties of Bayesian network to measure gene-gene interactions. According to Fisher [77], there is main two advantage of copulas: it is a way of studying scale-free measures of dependences and it is a starting point for constructing families of bivariate distributions. More about copulas and multivariate distributions can be found from Balakrishnan (2009) [78], Joe (1997) [79], Nelson (2007) [80], Nelson (1997) [81].
Mathematically, Copula is defined as follows
Definition 1: Copula is an n dimensional function, $C : [0,1]^n \rightarrow [0,1]$, which satisfies the following properties:

1. $C(u_1, \cdots, u_{i-1}, 0, u_{i+1}, \cdots, u_n) = 0$, i.e., the copula is 0 if one of any variable is 0.

2. $C(1, \cdots, 1, u, 1, \cdots, 1) = u$, i.e., the copula function is just u if one of the variables is u with all others being 1.

3. C is an n-increasing function.

Copula in probability — Let, $(X_1, X_2, \cdots, X_n)$ be the random vectors whose marginal distributions $(U_1, U_2, \cdots, U_n)$ are uniformly distributed in $[0, 1]$. A copula function $C : [0, 1]^n \rightarrow [0, 1]$ is defined as the joint probability distribution

$$C(u_1, u_2, \cdots, u_n) = F(U_1 \leq u_1, U_2 \leq u_2, \ldots, U_n \leq u_n). \tag{1.13}$$

Sklar's following theorem extends this definition to more general random variables with possibly non-uniform marginals, which is presented below.

Sklar's theorem — Let $(X_1, X_2, \cdots, X_n)$ be the random vectors whose marginals are $F_1(x_1), F_2(x_2), \cdots, F_n(x_n)$. So, for any joint distribution F, their exists a copula function C of its univariate marginal distributions such that,

$$F(x_1, x_2, \cdots, x_n) = C(F_1(x_1), F_2(x_2), \cdots, F_n(x_n)). \tag{1.14}$$

So, Copula is also known as joint distribution generating function with a separate choice of marginals.
Assuming $F(X_1, X_2, \cdots, X_n)$ has *nth* order partial derivatives, relation between the joint probability density function and the copula density function, say c, can be obtained from Equation (1.14) as,

$$\begin{aligned} f(x_1, x_2, \cdots, x_n) &= \frac{\partial^n (F(X_1, X_2, \cdots, X_n))}{\partial X_1 \partial X_2 \cdots \partial X_n} \\ &= \frac{\partial^n (C(F_1(x_1), F_2(x_2), \cdots, F_n(x_n))}{\partial X_1 \partial X_2 \cdots \partial X_n} \\ &= c(F_1(x_1), F_2(x_2), \cdots, F_n(x_n)) \prod_i f_i(x_i) \end{aligned} \tag{1.15}$$

where, we define,

$$c(t_1, \ldots, t_n) = \frac{\partial^n C(t_1, \ldots, t_n)}{\partial t_1 \cdots \partial t_n}. \tag{1.16}$$

There are many variants of copula [80]. Three widely used copula families are: Archimedean copula, Empirical copula, Gaussian copula. These are disussed below.

Archimedean copula — An n dimensional copula C is called Archimedean if it can be represented as

$$C(X) = \phi(\phi^{-1}(x_1), \phi^{-1}(x_2), \cdots, \phi^{-1}(x_n)) \tag{1.17}$$

ϕ is an Archimedean Copula generator function. According to generator function copula functions are different. $\phi^{-1}(x_i)$is inverse marginal distributions of individuals. The different Archimedean Copula families are:

- $\phi(x) = (x^{-\theta} - 1)^{\theta}$, known as *Clayton Copula* where $\theta \in [-1, \text{inf})$
- $\phi(x) = \ln\left(\frac{1-\theta(1-x)}{x}\right)$, known as *Ali-Mikhail-Haq Copula* where $\theta \in [-1, 1)$
- $\phi(x) = (-\ln(x))^{\theta}$, known as *Gumbel-Hougaard Copula* where $\theta \in [1, \text{inf})$

These are the variations of the Archimedean Copula. There is another kind of copula called non-parametric Copula or *Empirical Copula* which does not require initial parameters [82].

Empirical copula — Let $X_1, X_2, \cdots, X_n$ be the random variables with marginals distribution function $F_1(x_1), F_2(x_2), \cdots, F_n(x_n)$, respectively.
The empirical estimate of $(F_i, i = 1, \cdots, n)$, based on a sample, $\{x_{i1}, x_{i2}, \cdots, x_{im}\}$ of size m is given by

$$\widehat{F_i}(x) = \frac{1}{m}\sum_{j=1}^{m} 1_{\{X_{ij} \leq x\}}, [i = 1, \cdots, n] \tag{1.18}$$

The *Empirical copula* of $X_1, X_2, \cdots, X_n$ is then defined as

$$\begin{aligned} &\widehat{C}(u_1, u_2, \cdots, u_n) \\ &= \frac{1}{m}\sum_{j=1}^{m} 1\{\widehat{F_1}(x_{1,j}) \leq u_1, \widehat{F_2}(x_{2,j}) \leq u_2, \cdots, \widehat{F_n}(x_{n,j}) \leq u_n\}, \end{aligned} \tag{1.19}$$

for $u_i \in [0, 1], [i = 1, \cdots, n]$.

Gaussian copula — Let $X_1, X_2, \cdots, X_n$ be the random variables with univariate standard normal marginals distribution function $F_1(x_1), F_2(x_2), \cdots, F_n(x_n)$, respectively. Let, R be the correlation matrix of X.
The *Gaussian Copula* of $X_1, X_2, \cdots, X_n$ is then defined as

$$\begin{aligned} &C_{Gauss(R)}(u_1, u_2, \cdots, u_n) \\ &= F_R(F_1^{-1}(u_1), F_2^{-1}(u_2), \cdots, F_n^{-1}(u_n)) \end{aligned} \tag{1.20}$$

where F_R is the distribution function of the n variate normal distribution, with mean as zero vector and co-variance matrix R, and F_i is an univariate standard normal distribution function for each $i = 1, 2, \ldots, n$.

Dependence measure using copula — Here we model the dependence between two random variable using Kendall tau(τ) [83] measure. This can be expressed as the difference between the probability of concordance and discordance between two random variables. Formally it can be stated as:

$$\tau_{XY} = [P(x_1 - x_2)(y_1 - y_2) \geq 0] - [P(x_1 - x_2)(y_1 - y_2) \leq 0] \tag{1.21}$$

Kendall tau using the Copula function is described below [80]. Let, $X = \{x_1, x_2\}$ and $Y = \{y_1, y_2\}$ are two bivariate random variables with joint and marginal distributions as F_{XY}, $F_X(x)$ and, $F_Y(y)$ respectively, suppose C be a Copula function, then by Sklar's theorem (see equation 1.14)

$$F_{XY}(x, y) = C(F_X(x), F_Y(y)). \tag{1.22}$$

Now the Kendall tau can be expressed using the Copula function C as:

$$\tau_{C(X,Y)} = \tau_{XY} = 4 \int_0^{+1}\int_0^{+1} C(u, v)\, dC(u, v) - 1, \tag{1.23}$$

where $u \in F_X(x)$ and $v \in F_Y(y)$. We have used $\tau_{C(X,Y)}$ to model the dependency between transcriptomic profiles of two gene sets.

1.4.2 Application of Copula in Existing Literature

In [84], Copula modeling is named as copula craze. The copula dependence measure is the principle view of this paper. In [85], a basic introduction about the multivariate distribution based on copula is discussed. The author concentrate on the theoretical aspects of copula and its applications. Schmid et al. [86] proposed the copula-based multivariate model association. Nonparametric estimation of multivariate distribution measures is discussed in this paper using empirical copula. In a review work [87], different types of copulas modeling for multivariate control chart is discussed. Multivariate control charts are used to simultaneously observe the quality characteristics to detect the mean changes in manufacturing industries. In [88], the author gives a brief study about the various types of concavity and convexity in the class of multivariate copulas. A method for constructing multivariate Schur-concave copulas is also given in this paper. Frees et al., [89] develop some practical applications using copulas, including estimation of joint life

mortality and multi decrements models. They also discussed the basic properties of copulas and their relationship to the measure of dependences.
There is also wide application of copula in time series, finance, and economics. Patton et al.[90] provide a brief review of copula application in finance and economics. They also introduce the modeling of Markov processes and general nonlinear time series models using copula. In [91] a novel approach is proposed to model and forecast realized volatility (RV) measures based on the copula functions. As the marginal distributions and the dependence structure can be constructed separately, the copula-based approach permits for a great deal of flexibility in the construction of an appropriate multivariate distribution.

1.5 Scope of The Thesis

1.5.1 Structure Aware Principal Component Analysis for High Dimensional Data.

With the emergence of droplet-based technologies, it has now become possible to profile transcriptomes of several thousand cells in a day. While such a large single-cell cohort may favor the discovery of cellular heterogeneity, it also brings new challenges to the prediction of minority cell types. Identification of any minority cell type holds a special significance in knowledge discovery. In the analysis of single-cell expression data, the use of Principal Component Analysis (PCA) is surprisingly frequent for dimension reduction. The principal directions obtained from PCA are usually dominated by the major cell types in the concerned tissue. Thus, it is very likely that using a traditional PCA may endanger the discovery of minority populations. To this end, we propose Locality Sensitive PCA (*LSPCA*), a scalable variant of PCA equipped with structure aware data sampling at its core. Structure aware sampling provides PCA with a neutral spread of the data, thereby reducing the bias in its principal directions arising from the redundant samples in a dataset. We benchmarked the performance of the proposed method on eleven publicly available single cell expression datasets including one very large, annotated dataset. Results have been compared with traditional PCA and PCA with random sampling. Clustering results on the annotated datasets also show that *LSPCA* can detect minority populations with higher accuracy.

1.5.2 Stable Feature Selection using Copula in a Supervised Framework

FS is a key step in many machine learning tasks. A majority of the existing methods of FS address the problem by devising some scoring function while treating the features independently, thereby overlooking their interdependencies. We lever-

age the scale invariance property of copula to construct a greedy, supervised FS algorithm that maximizes the feature relevance while minimizing the redundant information content. Multivariate copula is used in the proposed copula Based FS (*CBFS*) to discover the dependence structure between features. The incorporation of copula-based multivariate dependency in the formulation of mutual information helps avoid averaging over multiple instances of bivariate dependencies, thus eliminating the average estimation error introduced when the bivariate dependency is used between a pair of feature variables.
We also developed a novel feature selection method called *RCFS* (**R**egularized **C**opula based **F**eature **S**election) based on regularized copula. l_1 regularization is used, as it penalizes the redundant co-efficient of features and makes them zero, resulting in non-redundant effective features set. Scale-invariant property of copula ensures good performance in noisy data, thereby improving the stability of the method. Three different forms of copula viz., Gaussian copula, Empirical copula, and Archimedean copula are used with l_1 regularization. Results prove a significant improvement in the accuracy of the prediction model to any non-regularized FS method.
This regularization based copula can also be used in gene selection of single cell RNA sequencing (scRNA-seq) data. Gene selection in unannotated large single cell RNA sequencing (scRNA-seq) data is a crucial step in the preliminary step of downstream analysis. The existing approaches are primarily based on high variation (highly variable genes) or significant high expression (highly expressed genes) and failed to provide a stable and predictive feature set due to technical noise present in the data.
A novel **reg**ularized **cop**ula based method (*RgCop*) for gene selection from large single cell RNA-seq data. *RgCop* utilizes copula correlation (*Ccor*), a robust equitable dependence measure that captures multivariate dependency among a set of genes in single cell expression data. The objective function is developed by adding a l_1 regularization term with *Ccor* to penalizes the redundant co-efficient of features/genes, resulting non-redundant effective features/genes set.

1.5.3 Feature Selection Using Copula in an Unsupervised Framework

Differential coexpression has recently emerged as a new way to establish the fundamental difference in expression patterns among a group of genes between two populations. Primitive methods detect the similarity of gene expression in one group and seek to identify significant differences in other groups. Some scoring techniques have been utilized to detect the changes in correlation patterns of a gene pair in two groups. However, modeling differential coexpression utilizing finding differences in dependence structure of gene pair has hitherto not been carried out.

Copula based framework is exploited to model differential coexpression between gene pairs in two different conditions, named *CODC*: A copula based model to identify differential coexpression. Copula is used here to model the pattern of expression profiles between a pair of genes in two conditions. For a gene pair, the distance between two joint distributions produced by copula is served as differential coexpression. We have utilized five bulk RNA sequence data to evaluate the model. Moreover, the proposed model can detect the mild change in coexpression pattern across two conditions which also be treated as differential coexpression.
Copula based framework is also used in single cell RNA sequence dataset. Single-cell RNA sequencing (scRNA-seq) data analysis is a powerful tool to study the molecular mechanisms underlying various biological processes. It can be used to study any process at the single-cell level, where gene expression heterogeneity is involved. To reveal valuable information about the data, such as recognizing the genes that are actively expressed in specific cell types, we need to sift through the high amount of technical noise. Noise is a persistent problem that occurs with scRNA-seq. The low amount of RNA that can be extracted from a single cell contributes to the high technical noise in scRNA-seq, and also makes the data-sparse in nature. A common problem with many existing gene selection methods is that they select genes with high expression variability, and are, thus, vulnerable to technical noise, due to the sparse nature of scRNA-seq datasets. Copulas are equipped to handle data sparseness and hence are better suited for such situations. This inspired us to come up with a robust unsupervised gene selection technique based on copula based graph convolution network (*sc-CGconv*). *sc-CGconv* is a stepwise robust unsupervised feature extraction and clustering approach that formulates and aggregates cell-cell relationships using copula correlation (Ccor), followed by a graph convolution network-based clustering approach. It can also handle substantially smaller sample sizes to identify stable gene sets. *sc-CGconv* can model the expression co-variability of a large number of genes, thereby outperforming state-of-the-art gene selection/extraction methods for clustering. Moreover, it preserves the cell-to-cell variability within the selected gene set by constructing a cell-cell graph through copula correlation measure.

1.5.4 Entropy based Feature Selection for High Dimensional Single Cell RNA Sequence Data

Annotation of cells in single-cell clustering requires a homogeneous grouping of cell populations. Since single-cell data is susceptible to technical noise, the quality of genes selected before clustering is of crucial importance in the preliminary steps of downstream analysis. Therefore, interest in robust gene selection has gained considerable attention in recent years. We introduce *sc-REnF*, (**r**obust **en**tropy based **f**eature (gene) selection method), aiming to leverage the advantages of

Rényi and *Tsallis* entropies in gene selection for single cell clustering. Experiments demonstrate that with tuned parameter (q), *Rényi* and *Tsallis* entropies select genes that improved the clustering results significantly, over the other competing methods. *sc-REnF* can capture relevancy and redundancy among the features of noisy data extremely well due to its robust objective function. Moreover, the selected features/genes can be able to determine the unknown cells with high accuracy. Finally, *sc-REnF* yields good clustering performance in the small sample, large feature scRNA-seq data.

1.5.5 A Deep Generative Framework for FS in Small Sample Large Dimensional Data

A fundamental problem of downstream analysis of scRNA-seq data is the unavailability of enough cell samples compared to the feature size. This is mostly due to the budgetary constraint of single cell experiments or simply because of the small number of available patient samples. Here, we present an improved version of the generative adversarial network (GAN) called LSH-GAN to address this issue by producing new realistic cell samples. We update the training procedure of the generator of GAN using LSH which speeds up the sample generation, thus maintaining the feasibility of applying the standard procedures of downstream analysis. LSH-GAN outperforms the benchmarks for the realistic generation of quality cell samples. Experimental results show that generated samples of LSH-GAN improves the performance of the downstream analysis such as feature (gene) selection and cell clustering.

2

Structure Aware Principal Component Analysis for High Dimensional Data

2.1 Introduction

A droplet based bar-coding technology has been developed that allows RNA sequencing of single cells (scRNA-seq) at a massive scale producing large datasets [92]. This has resulted in the sky-rocketing of sample sizes from 18 samples in 2012 to 1.3 million samples in 2016 [93]. A key promise of single-cell technology is to uncover the cellular heterogeneity within a tissue through transcriptomic analysis [94]. Within a single scRNA-seq data there are subpopulations, some of which may be large while others may represent minor (rare) groups. The challenge to discover these minor subpopulations from the large cohort is non-trivial [95]. Principal component analysis (PCA) is widely used as a dimension reduction tool when investigating a transcriptome dataset. Owing to the mechanics of traditional PCA, for a scRNA-seq dataset with typically ~25K genes and a few hundred to thousands of samples, the principal directions obtained are likely to be biased by the larger groups in the dataset. To put it simply, the top principal components are dominated by the sheer number of data points in the larger groups, thereby superseding the effect of small subpopulations. This poses a major setback in the analysis of scRNA-seq datasets.

Inspired by the benefits of sampling in imbalanced datasets [96, 97], in this chapter, a systematic down-sampling step before PCA is introduced in a way so that the structure of the dataset is preserved. We call this step as *structure aware data sampling*. To this end, we have built a scalable dimension reduction framework - **LSPCA** or **L**ocality **S**ensitive **P**rincipal **C**omponent **A**nalysis.

We evaluated *LSPCA* on multiple single-cell datasets including a large dataset of ~68K Peripheral Blood Mononuclear Cell (PBMC) transcriptomes where the true cell type annotations were available. When evaluating the clustering outcome on the annotated dataset using *LSPCA*, we obtained a better Adjusted Rand Index compared to traditional PCA. As a noteworthy benefit of *LSPCA*, we also observed

that *LSPCA* performs better in discovering minority cell groups. Details of the results on all datasets have been discussed in Section 2.4.

2.2 Background

2.2.1 Locality Sensitive Hashing

LSH [98, 25, 99] operates on a reduced dimension to find approximate nearest neighbors. LSH uses special locality-sensitive hash functions where the chances of similar objects hashing into a same bucket are more than the collision of dissimilar objects [26]. The details of LSH is given in Section 1.2.1 of Chapter 1.

2.2.2 Principal Component Analysis

Principal Component Analysis (PCA) [100], is a statistical technique which converts the observation points of correlated variables to linearly uncorrelated variables using orthogonal transformations. PCA is widely used as a dimension reduction technique [101]. Mathematically, PCA is defined as an orthogonal linear transformation of data into a new coordinate system by iteratively projecting instances on the direction of maximum variance [102]. PCA of a data matrix $\mathbf{D}_{m\times n}$ may be performed by diagonalizing its corresponding covariance matrix (a symmetric matrix) to obtain $\mathbf{E}_{n\times n}$. The diagonal matrix consists of eigenvalues in the decreasing order on the diagonal. The eigenvectors are called principal axes or principal directions of the data.
Singular Value Decomposition (SVD) is a matrix factorization method providing a robust computational framework to compute the principal components accurately for a variety of datasets. It is the generalization of the eigen decomposition. It is used to obtain $d = k < n$ principal components without the need to compute the diagonal matrix of the whole covariance matrix which also overcomes the restrictions caused by ill-conditioned matrices.
The steps to transform the original matrix into the new space are enumerated below.

1. Construct the co-variance matrix, **CV**, of the mean centered data.
2. Compute the eigenvalue matrix ($\mathbf{E}_{n\times n}$) of **CV**.
3. Compute **E′** by sorting **E** according to its corresponding eigenvalues in decreasing order.
4. Transform **D** to **PC** by projecting **D** on **E′**.

Note that the first principal component of $\mathbf{d_{(i)}}$ in the transformed co-ordinates is given by

$$\mathbf{PC_1} = \mathbf{D} \cdot \mathbf{E'}_{(1)}$$

where, $\mathbf{E'}_{(k)} = (E'_1, \ldots, E'_m)_{(k)}$, $k = 1, \ldots n$, is a unit vector of weights that transforms each row vector $\mathbf{d_{(i)}}$ of $\mathbf{D}$ to a new vector of principal component scores $\mathbf{PC}_{(k)} = (PC_1, \ldots, PC_d)_{(k)}$ where, $\mathbf{PC_k}$ denotes the k$^{\text{th}}$ principal component.

2.3 Materials and Methodology

The *LSPCA* framework is carried out in three major steps: (1) LSH based sampling, (2) computing principal components, (3) post-hoc projection of all data points onto the PCA space. The details of each step are described in the subsections. The flowchart of the whole process is outlined in Fig. 2.1.

2.3.1 LSH based Sampling

1. **Input:** Pre-processed data matrix, **D**, containing normalized counts of filtered cells and top dispersed genes.

2. **LSH Forest:** In this step, the hash codes of the input data points are produced. Unique hash codes which depict local regions or neighborhood are then computed. The python sklearn implementation of LSHForest module is used. The LSH Forest is applied on the pre-processed dataset with n_estimators = 30 and rest of the parameters are set to their defaults.

3. **K-NN graph:** An approximate neighborhood graph is obtained from the LSH tables. It is followed by searching for the 5-nearest neighbors (KNN) of each point. This involves computing the euclidean distances between the query point and its candidate neighbors.

4. **Sampling:** Sampling is carried out in a 'greedy' fashion. Each data point is *visited* sequentially in the same order as it appears in the original dataset. During each *visit*, its respective 5-NN are flagged and never visited again. In this way, a sub-set of samples is obtained. The residue is further down-sampled by performing the *visit* step recursively. For example, in the 68K PBMC dataset, after 5 such iterations, $\sim$ 2000 samples were retained.

2.3.2 Computing *LSPCA* Rotation Matrix

LSPCA uses the structure preserving samples, $\mathcal{SS} \subset \mathbf{D}$, obtained in the previous step to compute its eigenvector matrix by using a traditional PCA implementation.

Performing SVD on LSH based samples produces a transformation matrix similar to a PCA rotation matrix.

2.3.3 Post-hoc Projection onto the PCA Space

The obtained transformation matrix, O, is used to project the entire dataset onto the new projection bias. The features in projected data are referred to as *LSPCA* components.

$$\textit{LSPCA} \text{ Component Matrix } = \mathbf{D} \times O$$

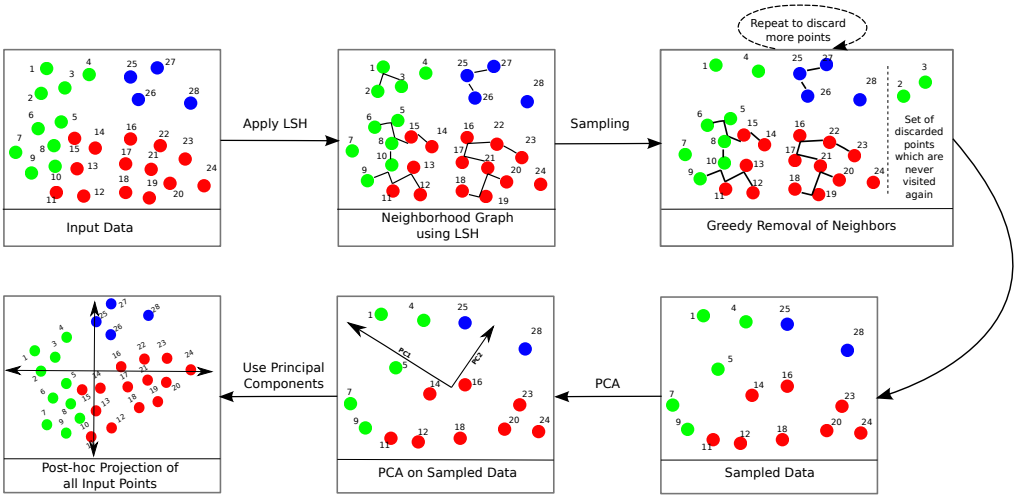


Figure 2.1: Pictorial view of the *LSPCA* algorithm. The solid circles represent individual samples of a dataset. The numbers adjacent to each sample indicate the respective index.

2.4 Results and Discussion

2.4.1 Data Description

Two types of single cell datasets were used for the evaluation. The unannotated recount single cell datasets comprised 9 datasets sourced from the *recount2* project [103]. The second type contained two datasets with annotations. The summary of the datasets are provided in Table 2.1.

- recount is a multi-experiment resource of analysis-ready RNA-seq gene and exon count datasets, *recount2* is the later version of the recount project.

This online resource consists of RNA-seq gene and exon counts along with their corresponding the coverage *bigWig* files for 2041 different studies. All unannotated datasets used in our experiments was downloaded from https://jhubiostatistics.shinyapps.io/recount/.

- The PBMC dataset prepared by [58], was downloaded from https://support.10xgenomics.com/single-cell-gene-expression/datasets. The data is sequenced on Illumina NextSeq 500 high output with 20,000 reads per cell.
- Another annotated dataset (GSE79374) originated from profiling of 333 single-cells isolated from the mouse brain tissues across 3 developmental time points. Visual cortex, Hippocampus, and thalamus regions were profiled on Embryonic day 18.5(e18.5), 12 days old postnatal (p12) and adult mice (n=5 each). The unannotated cells were removed from the dataset in our analysis.

Table 2.1: The Table gives a brief summary of the datasets used in the experiments.

Serial #	Dataset Name	Domain	Features	Instances
1	GSE57872	Human Tumor Cells	58037	875
2	DRP001358	Human Cancer Cells	58037	337
3	DRP002435	Human HeLa Cells	58037	477
4	GSE53529	Human Myoblasts	58037	384
5	GSE64016	Human Embroyonic Stem Cells	58037	460
6	GSE70580	Human Tonsil Cells	58037	648
7	GSE67835	Human Brain Cells	58037	466
8	GSE66357	Human Tonsil Cells	58037	643
9	GSE75140	Human Cerebral Tissues	58037	734
10	PBMC	Fresh PBMC, Healthy Doner	32738	68793
11	GSE79374	Mouse Brain Cells	22074	280

2.4.2 Data Preprocessing

Datasets containing raw UMI read counts were downloaded from multiple sources: (1) using the Bioconductor *recount* package by [103], (2) from the www.support.10xgenomics.com website and (3) the mouse embryos scRNA-seq dataset by [104]. Each dataset was preprocessed using the steps outlined in Fig. 2.2.

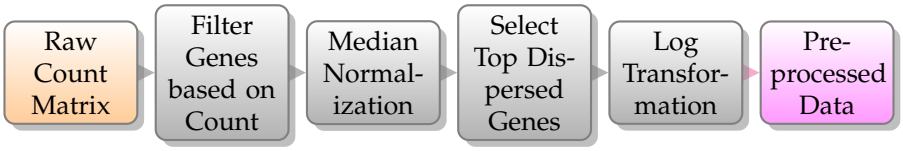


Figure 2.2: Data pre-processing stages

1. **Cell and Gene filtering:** Raw count matrix was filtered for poor quality cells. Typically single-cell datasets have more than 20,000 genes. Genes having count greater than 2 in at least 4 cells were kept for subsequent pre-processing.

2. **Median Normalization:** The filtered data matrix was median normalized. The median normalization step then involved division of UMI counts in the filtered matrix by the total UMI counts in each cell. These scaled counts were multiplied by the median of the total UMI counts across cells [105].

3. **Gene Selection:** In order to select the top dispersed genes, 2000 most variable genes were selected based on their relative dispersion (variance/mean) with respect to the expected dispersion across genes with similar average expression [92, 105].

4. **Log Normalization:** The resulting matrix (of dimension n cells $\times$ 2000 genes) was further subjected to $\log_2$ transformation after addition of 1 as a pseudo count.

2.4.3 Simulation Parameter Settings

LSPCA was compared with (1) a traditional PCA on whole dataset and (2) PCA on a subset of data obtained through random sampling (RSPCA). The random sampling PCA uses random sampling instead of LSH based sampling. The number of random samples were kept equal to the number of samples obtained from LSH based sampling. We included this method to contrast the effectiveness of LSH based sampling against random sampling. The *PCA* module from the python-sklearn implementation was used for the traditional PCA with *svd_solver='full'*. Every raw dataset was processed as per the steps as outlined in Fig. 2.2. The preprocessed datasets were then subjected to the three feature extraction variants: PCA, *LSPCA*, and RSPCA. For the clustering experiments, k-means was applied to the top 50 principal components obtained from the respective PCA variant. The best k for each of the unannotated datasets was determined using two strategies viz. Bayesian Information Criterion (BIC) [106], and Silhouette Scores [107]. The k corresponding to the highest score was chosen. Both the strategies reported the same value of k for each dataset. On the other hand, the k for the annotated datasets was set according their respective known number of cell types.

Execution time

PCA took approximately one hour on the PBMC dataset compared to only ~11 minutes by *LSPCA*. All experiments were carried out on a PC having an Intel Core i7-3770 3.40 GHz processor and 32GB of RAM.

2.4.4 Simulation Results

Clustering accuracy

The clustering performance by respective PCA methods was reported using the Silhouette Score for the unannotated datasets and the Adjusted Rand Index (ARI) for the annotated datasets. In addition, the clustering accuracy of the minor cell type populations of the annotated datasets was examined exclusively.

Silhouette score

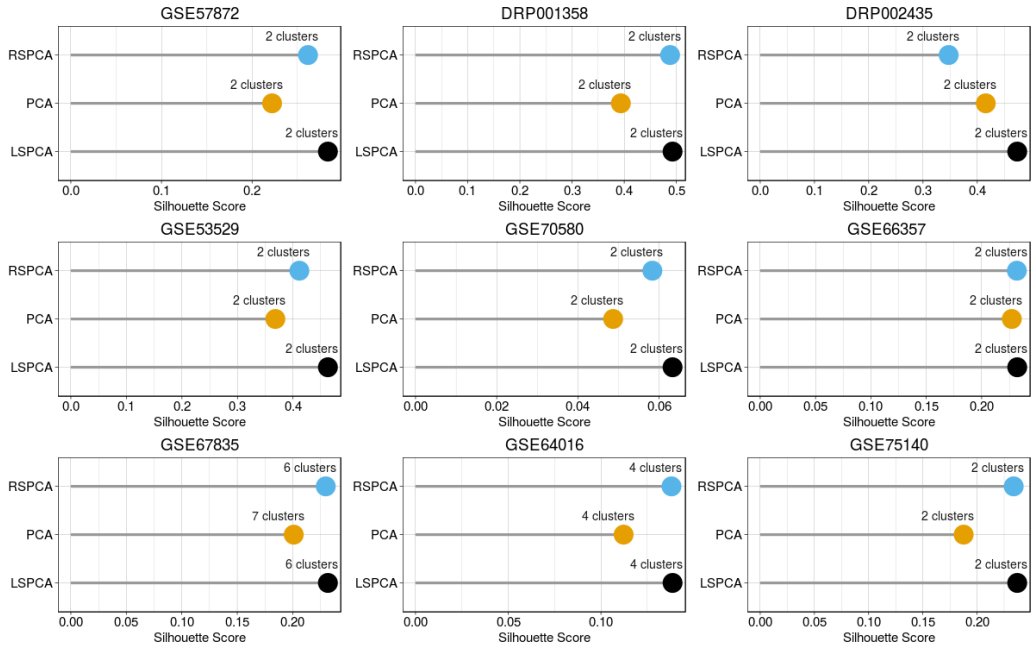
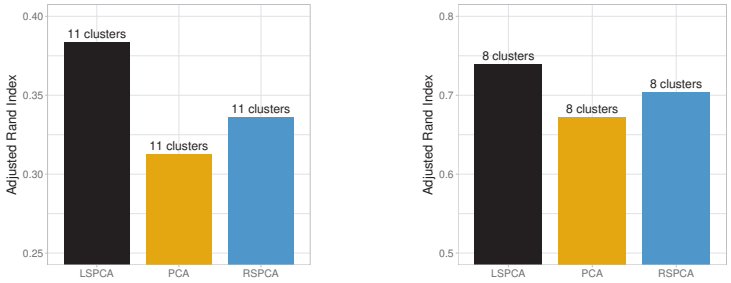


Figure 2.3: Silhouette Index of the unannotated datasets. The values adjacent to the bubble indicate the number of clusters for which the maximum Silhouette Score could be attained. the x-axis indicate the Silhouette Score.

The Silhouette Score (or Index) is an internal validity measure based on similarity of points within a cluster and separation across other clusters [107]. It is used for validation of consistency within clusters of data when data labels are not present. A high average silhouette score indicates that the objects within a cluster are tightly placed and the individual clusters are well separated. The Fig. 2.3 describes the Silhouette score for nine single-cell recount datasets clustered on the top 50 components obtained from the respective PCA variants. The comparative performance shows that the proposed *LSPCA* method gives a better lower

dimensional representation of original datasets than the alternative variants of PCA.

Adjusted rand index



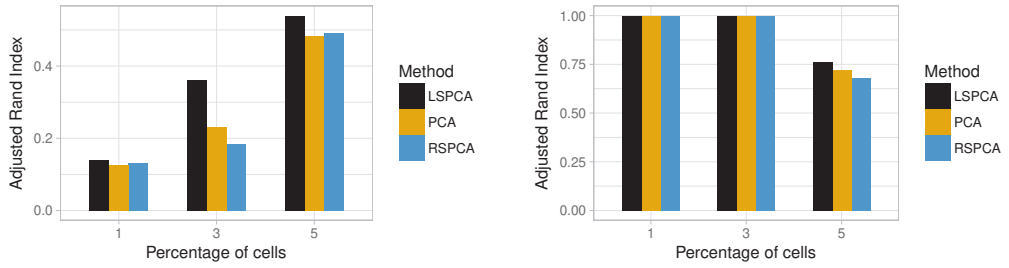
(a) (b)

Figure 2.4: Comparing the *k*-means clustering accuracy evaluated using Adjusted Rand Index (ARI) of three PCA variants on (a) PBMC dataset and (b) Mouse Brain tissue dataset with the best K determined using Silhouette score metric.

For the datasets whose label information is available, Adjusted Rand Index (ARI) metric is a suitable choice to evaluate clustering performance. The value of ARI is close to 0 when the clustering prediction is random, whereas the ARI is 1 when the clustering is perfectly identical to the true labels [108]. The Fig. 2.4 illustrates the clustering performance on the PBMC and Mouse Brain Cell datasets for the different PCA methods. It is observed that the ARI scores obtained from PCA with LSH based sampling is higher than traditional PCA or PCA with random sampling.

Discovery of small clusters

The effectiveness of sampling is more pronounced when the performance of clustering is measured only for the minority groups. For this measurement, we selected only those transcriptomes which shared the annotated groups of size <= 5% in the entire dataset. Fig. 2.5 (a) shows the ARI computed for the transcriptomes appearing in minor groups of the ~68K PBMC dataset for all the methods. For all the categories, more accurate predictions were made when *LSPCA* was used.
Similarly, Fig. 2.5 (b) depicts the ARI computed for the transcriptomes of the minority groups in the Mouse Brain dataset (GSE79374) containing a mixture of 280 annotated transcriptomes. Only the selected cells are used for computing the ARI. All the methods could detect the smallest known cluster *Newly formed Oligodendrocyte (5 cells)* belonging to the 1% category. The *unknown subtype in the visual cortex (purple, 7 cells)* included in the 3% category were also detected by all the



(a) (b)

Figure 2.5: ARI to compare the discordance with respect to the known annotation as discovered by the three PCA variants. (a) **PBMC dataset**: Members from the *CD34+ (262 cells)* and the *CD4+ T Helper2 (19 cells)* groups together constitutes the 1% category. The 3% category contains the *Dendritic Cells (1865 cells)* in addition to the cells under the 1% category. The third subset includes the *CD4+/CD45RA+/CD25- Naive T (2793 cells)* and the *CD4+/CD45RO+ Memory (3126 cells)*; (b) **Mouse Brain dataset**: *Newly formed Oligodendrocyte (5 cells)* belongs to the 1% category. The *unknown subtype in the visual cortex (purple, 7 cells)* is also present in the 3% category. The 5% category includes the *Oligodendrocyte progenitor (16 cells)*.

three methods. However, *LSPCA* clearly produced better principal components that allowed *k*-means to detect the *Oligodendrocyte progenitor (16 cells)* appearing in the 5% category.

2.5 Conclusion

In this chapter, a systematic down-sampling method called *LSPCA* is proposed to improve PCA while keeping the structure of the dataset. The components obtained from *LSPCA* are utilized to cluster high dimensional single cell data. The results show that *LSPCA* can identify small subpopulation of cells from single cell data identification which is difficult for the traditional PCA technique.
It may be noted that *LSPCA* is applicable wherever PCA is used. In particular, *LSPCA* will be useful for large datasets like single-cell expression matrices. It is fast and produces components almost identical to the traditional PCA components. The method provides flexibility to a user to adjust the number of samples to execute the PCA. It must be noted that *LSPCA* is not a new dimension reduction method, but when the dataset is large, it assists PCA to work on a smaller, subset whose members adequately represent the whole dataset. Compared to random sampling, structure aware sampling is a more effective way to sample from a large dataset. *LSPCA* performs dimension reduction by operating on a subset of less redundant samples without significantly altering the performance of the traditional PCA.

Although the *LSPCA* performs well in any high dimensional single cell data, the principal components are essentially linear combinations of the original features/genes. Therefore, it is difficult to ascribe any meaning to the principal components. In contrast, feature/gene selection methods yield a subset of features each of which corresponds to the original features/genes, and are hence easily understandable. The next chapter introduces a new copula based feature selection method that possesses desirable properties.

3

Stable Feature Selection using Copula in a Supervised Framework

3.1 Introduction

With the advancement of science and technology, data has increased both in the number of samples and number of dimensions [109]. Examples of high-dimensional data include genomic data [2], text data [3], social image data [4], transcriptome data [5], etc. Over the last five decades, feature engineering has emerged as a necessary ingredient of machine learning and has become a field of study by itself [110]. Feature selection and feature extraction are two broad components of feature engineering. In this chapter, we focus only on the feature selection task.

The importance of feature selection [111] is twofold - it reduces the computational cost [112] and helps avoid model overfitting. Feature selection, in practice, often improves the accuracy of down-stream machine learning tasks, including clustering and classification. In [113], the authors proposed a new feature selection algorithm based on dynamic mutual information, which is only estimated on unlabeled instances. In [114], the authors introduced a local discriminant model in the feature selection framework of subspace learning. The model preserves both the local discriminant structure and the local geometric structure of the data. In [115], a stratified sampling method was employed to select the feature subset for random forests with high dimensional data. Features were separated into two groups. One group contained strong informative features and the other weak informative features. Then, random features were chosen from each group proportionally. In [116], the authors proposed a space division strategy based on the feature importance, which can choose relevant features into the same subspace with a low computational cost. In [117], an information-theoretic approach was proposed to extract the hidden common structure shared by a set of random variables.

Feature selection methods are of three broad types - *filters* [27], *wrappers* model [28]

and *embedded* techniques[29]. Filters typically measure the association between explanatory variables and the dependent variable. Some of these association measures are - Pearson's correlation coefficient [30], Chi-squared test [31], mutual information [32], etc. Wrapping methods compute models with a certain subset of features and evaluate the importance of each feature. Some of these measures are - Forward selection [118], Backward selection [119], and Stepwise selection [120]. Embedded methods combine the qualities of filter and wrapper methods. The most Common embedded technique are the tree algorithm's like RandomForest [121], Lasso Regression [122], Ridge Regression [123]. The details of feature selection methods are given in Section 1.2.2 of Chapter 1.

Feature selection, in the context of supervised learning, aims to identify an optimal feature subset such that the model accuracy is maximized. Finding the exact solution amounts to an NP-hard problem [124]. Approximate methods [125] are devised in the intersection of mathematics, statistics, and algorithms to circumvent this problem.

Of late, mutual information-based filter methods have gained popularity due to their ability to capture the non-linear association between dependent and independent variables in a machine learning setting. Mutual Information-based Feature Selection (MIFS) is among the earliest algorithms in this segment [45]. It is a greedy algorithm that considers both mutual information of a candidate feature with class label information and the prior selected features. Conditional Mutual Information Maximization (CMIM), along the same line, maximizes mutual information concerning the class while conditioning upon the selected features [46]. One of the popular mutual information-based approaches is the Minimal-Redundancy-Maximal-Relevance criterion (MRMR) [14], which considers feature relevance to the class labels and ensures that redundant features are not present in the final feature subset. In [126], the authors introduced a method called Double Input Symmetric Relevance (*DISR*), where joint mutual information was estimated with symmetrical relevance [126]. The limitation of the above methods is, that they are using an average of bivariate redundancy measure between all pairs of features. These result an average estimation error in redundancy measure [14, 45].

To address the above issues, in this thesis, we propose a copula based feature selection technique (*CBFS*). Thereafter it is extended to provide a more robust and stable feature selection method by introducing a regularization term. The extended version of the *CBFS* is called *RCFS* (Regularized Copula based Feature Selection). This is further extended to provide *RgCop* (A regularized copula based method for gene selection in single cell RNA-seq data), for use in relevant gene selection problems of single cell RNA-seq data. In the following subsections the theoretical background, the proposed method and its extensions, and the experimental results are described in detail.

3.2 Theoretical Background and Formal Details

In this section, we illustrate some basic preliminaries about copulas and mutual information.

3.2.1 Copula

The name Copula comes from a Latin word *copulare*, which means joint together. Copulare word is first used by famous statistician *Sklar* in 1959 in one of his famous Sklar's Theorem. Copula produces a multivariate probability distribution from multiple uniform marginal distributions. Copula [80] is also extensively used in high dimensional data applications to obtain joint distributions from a random vector, easily by estimating their marginal functions. Mathematically, Copula is defined as follows:
Definition 1: Copula is an n dimensional function, $C : [0,1]^n \rightarrow [0,1]$, which satisfies the following properties:

1. $C(u_1, \cdots, u_{i-1}, 0, u_{i+1}, \cdots, u_n) = 0$, i.e., the copula is 0 if one of any variable is 0.

2. $C(1, \cdots, 1, u, 1, \cdots, 1) = u$, i.e., the copula function is just u if one of the variables is u with all others being 1.

3. C is an n-increasing function.

The detail description of copula theory is given in Section 1.4 of Chapter 1.

Copula correlation measure

Let, $Y = \{y_1, y_2\}$ and $Z = \{z_1, z_2\}$ are two bivariate random variables and their joint and marginal distributions are H_{YZ}, $F_Y(y)$ and, $F_Z(z)$ respectively. Now H_{YZ} can be expressed as: $H_{YZ}(y,z) = C(F_Y(y), F_Z(z))$, where C is a copula function.
Kendall tau(τ), the measure of association, [83] can be expressed in terms of concordance and discordance between random variables. Kendall tau is the difference between probability of concordance and discordance of (y_1, y_2) and (z_1, z_2). It can be described as

$$\tau_{YZ} = [P(y_1 - y_2)(z_1 - z_2) \geq 0] - [P(y_1 - y_2)(z_1 - z_2) \leq 0] \tag{3.1}$$

According to Nelson [127] Kendall tau can be expressed using copula function:

$$\tau(C_{Y,Z}) = \tau_{YZ} = 4 \iint_0^{+1} C(u,v)\, dC(u,v) - 1 \tag{3.2}$$

Where, $u \in F_Y(y)$ and $v \in F_Z(z)$. $\tau(C_{Y,Z})$ is termed as copula-correlation (*Ccor*) in our study.

Regularization

Regularization is a type of regression that penalizes the coefficient of redundant feature towards zero [128]. The simplest regularization is l_1 norm or Lasso Regression, which adds absolute value of magnitude" of coefficient as penalty term to the loss function. Another widely used regularization is l_2 norm or Ridge Regression, which adds "squared magnitude" of coefficient as penalty term to the loss function. The key difference between these two is that Lasso shrinks the less important feature's coefficient to zero and thus, removes some features as well. So, this will be applicable where we would have huge number of features. On the contrary, l_1 norm regularization produces sparse solutions by making higher coefficients of the loss function to zero. l_1 norm or Lasso Regression is used in our model to handle the scRNA-seq data with the large number of features.
For any vector $A \in \mathcal{R}^m$, the l_1 norm is $\|A\|_1 = \gamma \sum_{i=1}^{m} |A_i|$, where γ is a tuning parameter, controls penalization. For $\gamma = 0$ regularization effect is none. When γ value increases, it starts to penalizes the larger coefficients to zero. However, after a certain value of γ, the model starts losing important properties, increasing bias in the model and thus causes under-fitting. We tuned γ using eight synthetic Gaussian mixture dataset in this study.

3.2.2 Information Theory

Here, we discuss some basic parts of information theory based on entropy and Mutual Information. The entropy is defined as a measure of uncertainty and average information in a random variable [129]. The entropy of discrete random variable, $X = (x_1, x_2, \cdots, x_n)$ is defined as:

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log_2 p(x_i) \tag{3.3}$$

where, $p(x_i)$ is the probability mass function, defined as:

$$p(x_i) = \frac{\text{Number of events occur with, } x_i}{\text{Total number of events, } n} \tag{3.4}$$

The conditional entropy of a random variable X given another random variable $Y = (y_1, y_2, \cdots, y_n)$ is defined as:

$$H(X|Y) = -\sum_{j=1}^{m}\sum_{i=1}^{n} p(x_i, y_j) \log_2 \frac{p(x_i, y_j)}{p(y_j)} \tag{3.5}$$

The joint entropy between two discrete random variables X and Y is defined as:

$$H(X, Y) = -\sum_{j=1}^{m}\sum_{i=1}^{n} p(x_i, y_j) \log_2 p(x_i, y_j) \tag{3.6}$$

Mutual Information is a mutual dependence measure between two random variables. For any two continuous random variable X and Y, the mutual information is given in form of their probability distribution functions $p(x)$, $p(y)$ and $p(x, y)$ as:

$$I(X, Y) = \iint p(x, y) \log \frac{p(x, y)}{p(x)p(y)}\, dx\, dy \tag{3.7}$$

For any two discrete random variables x and Y, mutual information is then given by

$$\begin{aligned} I(X, Y) &= \sum_{x,y} p(x, y) \log \frac{p(x, y)}{p(x)p(y)} \\ &= \sum_{x,y} p(x, y) \log \frac{p(x|y)}{p(x)} \\ &= H(X) - H(X|Y) \end{aligned} \tag{3.8}$$

3.2.3 Relation of Copula with Mutual Information

From Equations (1.15), (3.5) and (3.6) the mutual information between two random variables X and Y can be described in terms of the copula function as

$$\begin{aligned} I(X, Y) &= \iint p(x, y) \log \frac{p(x, y)}{p(x)p(y)}\, dx\, dy \\ &= \iint c(P(x), P(y))p(x)p(y) \\ &\quad \log \frac{c(P(x), P(y))p(x)p(y)}{p(x)p(y)}\, dx\, dy \\ &= \iint c(u, v) \log c(u, v)\, du\, dv \\ &= -H(C(u, v)) \end{aligned} \tag{3.9}$$

where, u and v are the individual marginal distributions, respectively, $P(x)$ and $P(y)$. So, It can be seen that the Mutual Informations of the random variables are similar as negative entropy of their corresponding Copula distributions.

3.2.4 Limitations of the Previous Works

Earlier feature selection methods are mainly concern on the optimization of two criteria: maximizing relevance and minimizing redundancy. The limitations of these methods are the following:

- *MIFS* is not efficient for a large number of features. If the magnitude of the redundancy term increases, some irrelevant features may be selected. This limitation is removed in *MRMR* and *NMIFS* by dividing the redundancy term with the total number of subsets.
- Most of the existing methods employ the cumulative summation and forward search to approximate the solution which causes overestimation of significance (*MIFS*, *CMIM*, and *MRMR*). The overestimation of significance can occur when the candidate features have a high correlation with preselected features. Still, at the same time, these are almost independent of the remaining subsets. This situation causes high redundancy for the candidate feature sets.
- Existing feature selection works are dataset dependent [47]. Changes in dataset results different selected features (*MIFS*, *CMIM*, and *MRMR*). Significance of all the discussed methods depends on the characteristics of datasets.

3.3 Materials and Methodology

The proposed method for feature selection addresses the issue of overestimation of some feature due to the cumulative summation of Mutual Information.
Let, a dataset be D with N dimensions. The total feature space is $F = \{f_1, f_2, \cdots, f_N\}$. We want to select a sub set of features with n dimensions, where $n << N$. The feature subset will be $S = \{f_1, f_2, \cdots, f_n\}$. We aim to select a feature subset S, which have same or better classifier accuracy than the feature set F.
After introducing the proposed method (*CBFS*), in this section, we will also show that our *CBFS* method is equivalent to max dependency search method [130]. We also compute the optimal bound of our method. Let us start with defining the related optimality criteria.

Max dependency The mutual information-based feature selection method which selects the feature set having the largest dependency with the class variable, is

known as Maximal Dependency(MD) method [130]. This can be mathematically expressed as:

$$\widehat{S}_{\text{selected}} = \arg\max_{S} I(S, Y) = \arg\max_{f_s \in F - \dot{S}} I(f_s, Y|\dot{S}) \tag{3.10}$$

Here, $S = (\dot{S}, f_s)$ is a subset of features, where f_s is the last feature in selected set S and Y is class level. Often direct implementation of the first minimization in Equation 3.9 is infeasible when the second optimization implemented it, known as the first order incremental search. The method describes that one feature is selected at each iteration, and that feature will be optimum, i.e., maximum dependent with the class variable.

Relevancy A feature f_i is more relevant to class level Y than another feature f_j, if f_i has higher mutual information with the class label Y than f_j. It is the *Relevancy* test for the features[130]. Mathematically f_i is more relevant than f_j if

$$I_c(f_i, Y) > I_c(f_j, Y), \tag{3.11}$$

Where $I_c(x, y)$ is a mutual information measure, which we will use to be the copula mutual information. As an alternative to the Max-dependency method, the maximum-relevancy method selects the feature set S as

$$\widehat{S}_{max-rel} = \arg\max_{S} \frac{1}{|S|} \sum_{f_s \in S} I_c(f_s, Y). \tag{3.12}$$

Minimum redundancy It is often the case that the feature set S obtained by max-relevancy criterion (3.12) contains features that are mutually inter-dependent to a high extent to be redundant. Redundancy is measured using the minimization of mutual information between all selected features f_s and non selected feature, say f_i. It can be mathematically expressed as:

$$min_{s:f_s \in S} I_c(f_i; f_s) \tag{3.13}$$

3.3.1 Copula Based Feature Selection

As mentioned in [130] there is an overestimation to find the redundancy (*MIFS, CMIM and MRMR*). Most of the previous feature selection works, discussed in Section 3.1 (including the mRMR method of [14]), were dataset dependent. Any noise in the dataset may change the selected feature subset. We develop a copula-based feature selection (*CBFS*) method to optimize the relevancy and redundancy, which is much more stable than the existing methods. We minimize the copula

mutual information between f_i and f_s (to reduce the redundancy between them) and maximizing the copula mutual information between class label Y and f_i. So, we are indeed using the first order incremental search to select one feature in each step but propose to use (empirical) copula-based mutual information instead of the standard information measure (as in [14]) to achieve more stability. Further, after selecting more than one feature, we use multivariate mutual information in contrast with the average in (3.13). Mathematically, it can be expressed as follows. After selecting $f_1, \ldots, f_s \in S$, we select the next feature ($f_{s+1} = f_{\text{CBFS}}$) by

$$\begin{aligned} f_{CBFS} &= \arg \max_{f_i \in (F-S)} [I_c(f_i; Y) - I_c(f_i; f_1; f_2; \cdots; f_s)] \\ &= \arg \max_{f_i \in (F-S)} [-H(C(P(f_i), P(Y)) + H(C(P(f_i), P(f_1), \cdots, P(f_s)))] \end{aligned} \tag{3.14}$$

The feature selection method (*CBFS*) is illustrated in Fig. 3.1. The Venn diagram A denotes the entropy of non selected features, $H(f_i)$, B denotes the entropy of selected features, $H(f_s)$, and C denotes the entropy of target class, $H(Y)$. The first diagram represents the copula based relevancy, $I_c(f_i, Y)$. The second diagram represents the copula based redundancy, $I_c(f_i, f_1, f_2, \ldots, f_s)$. The third diagram depicts the objective function of the algorithm. The optimum feature is obtained by maximization of the objective criteria iteratively.

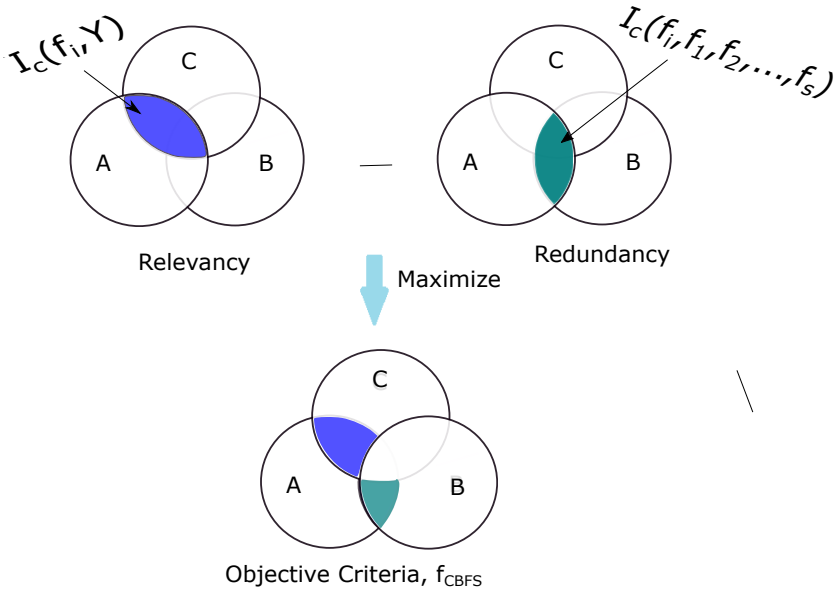


Figure 3.1: Pictorial form of the feature selection method (*CBFS*)

Algorithm for the Copula Based Feature Selection

CBFS algorithm is described in Algorithm **1**. The algorithm is discussed below.

1. This algorithm takes data matrix D with class label Y, and the number of features to be selected k as the input parameter.

2. The first most relevant feature is selected by maximizing the copula based mutual information between class label Y and all features in F.

3. M contains the copula based multivariate mutual information values between each non selected feature (f_i) with all selected feature ($f_1, f_2, \ldots, f_s$).

4. R contains the copula based multivariate mutual information values between each non selected feature (f_i) with class label Y.

5. Subtraction of values in M from values in R are kept in E.

6. A feature is selected with maximum value, obtained from E, and merged in selected feature list S iteratively.

7. The above process (step 3 to step 6) is repeated k times.

8. Thus, an optimal feature subset is obtained in S.

9. S is returned as output.

Algorithm 1 Copula Based Feature Selection Algorithm(*CBFS*)

Input: Data Matrix **D**, Target class **Y**, Number of Selected Features, **d**.
Output: Optimal Feature subset,(**S**).

Initialisation:

S = $\emptyset$, {**S** will hold sub-set feature indices.}
S[1] $\leftarrow Max_{f_i} I_c(f_i, Y)$ {Initial most relevant feature}
for $i = 0$ to $(\mathbf{k} - \mathbf{1})$ **do**
  E = $\emptyset$
  M $\leftarrow I_c(f_i; f_{1s}; \cdots; f_{ks})$ {Redundancy Criterion}
  R $\leftarrow I_c(f_i; Y)$ {Relevancy Criterion}
  E $\leftarrow (R - M)$
  S $\leftarrow \{\mathbf{S} \bigcup \arg\max(\lim_{f_i}\{E\})$
  F $\leftarrow F - \{f_i\}$
end for
return S

3.3.2 Optimality of Copula Based Feature Selection

In article [14], authors proved that their method (mRMR) with first order incremental search is optimum in the sense that it becomes equivalent to the max dependency criteria. It is proved in this section that Copula Based Feature Selection is also optimum, which is equivalent to max dependency criteria (and hence also equivalent to the mRMR method).

Theorem 1. *The proposed method CBFS is equivalent to the Maximal Dependency criteria (3.10).*

Proof. Since we are also using the first order incremental search, we assume that without loss of generality, the s feature $S_s = \{f_1, \ldots, f_s\}$ has already been selected optimally and we select the $(s+1)$-th feature $f_{s+1} = f_{\text{CBFS}}$ by (3.28). Let us denote $S_{s+1} = \{f_1, \ldots, f_s, f_{\text{CBFS}}\}$. Then the max dependency criteria is to maximize $I(S_{s+1}, Y)$. So, we need to show that this is equivalent to our proposed method (3.28). From the article [14], it can be shown that the maximization of $I(S_{s+1}, Y)$ is equivalent to simultaneous maximization of relevancy and minimization of redundancy. So, it is enough to show that our *CBFS* method also satisfies the max-relevancy and min-redundancy criteria. We can easily show that the minimum redundancy criteria satisfy when the second term of Equation (3.28) has a minimum bound over zero. It can be shown as below:

$$
\begin{aligned}
& I_c(f_1; \cdots; f_s) = \\
& \int \cdots \int p(f_1; \cdots; f_s) \log \frac{p(f_1; \cdots; f_s)}{p(f_1) \cdots p(f_s)} \, df_1 \ldots df_s \\
& = \int \cdots \int c(P(f_1), \cdots, P(f_s)) \prod_i P(f_i) \\
& \log \frac{c(P(f_1), \cdots, P(f_s)) \prod_i P(f_i)}{p(f_1) \cdots p(f_s)} \, df_1 \ldots df_s \\
& = \int_0^1 \cdots \int_0^1 c(u_1, \cdots, u_s) \log c(u_1, \cdots, u_s) \, du_1 \ldots du_s \\
& \geq \int_0^1 \cdots \int_0^1 (\prod_i u_i) \log(\prod_i u_i) \, du_1 \ldots du_s \\
& = 0
\end{aligned}
\tag{3.15}
$$

where, u_i is the individual uniform marginal distribution functions, $P(f_i)$. It is observed that when $(c(u_1, \cdots, u_s) = \prod_i u_i))$,all the features are independent to each other, then mutual information among selected feature is minimized. It is the minimum redundancy criteria.
Now, we estimate the upper bound of the Equation (3.28), which will satisfy the

maximum relevance criteria. It can be shown as below:

$$\begin{aligned} I_c(f_i; Y) &= \iint p(f_i, Y) \log \frac{p(f_i, Y)}{p(f_i)p(Y)}\, df_i\, dY \\ &= \iint c(p(f_i), p(Y))p(f_i)p(Y) \log c(p(f_i), p(Y))\, df_i\, dY \\ &= \int_0^1 \int_0^1 c(u_i, v) \log c(u_i, v)\, du_i\, dv \\ &\leq min(u_i, v) \end{aligned} \tag{3.16}$$

where, u_i and v are the individual uniform marginal distribution functions, $P(f_i)$ and $P(Y)$. It is verified from the above results that the maximum of $I_c(f_i; Y)$ in Equation (3.28) is minimum of the feature variables and class label i.e. when the feature variable is maximally dependent with the class variable. □

3.3.3 Stability: The Advantage of *CBFS*

Most of the existing mutual information based feature selection methods are dataset dependent. One of the striking advantages of the proposed method is that it overcomes the downside of the primitive mutual information based approach due to its *scale invariant* property [80].

Proposition 1: Consider, there are two random variables X and Y, and their copula function is C_{XY}. If α and β are two functions of X and Y respectively, then relation of the Copula of $(\alpha(X), \beta(Y))$ and (X, Y) are as follows.

- If α and β are strictly increasing functions, then the copula of $(\alpha(X), \beta(Y))$ can be expressed as:

$$C_{\alpha(X)\beta(Y)}(u, v) = C_{XY}(u, v) \tag{3.17}$$

- If α is strictly increasing and β is strictly decreasing , then we have

$$C_{\alpha(X)\beta(Y)}(u, v) = u - C_{XY}(u, 1 - v) \tag{3.18}$$

- If α is strictly decreasing and β is strictly increasing function, then we have

$$C_{\alpha(X)\beta(Y)}(u, v) = v - C_{XY}(v, 1 - u) \tag{3.19}$$

- If α and β both are strictly decreasing function then their copula function can be expressed as:

$$C_{\alpha(X)\beta(Y)}(u, v) = u + v - 1 - C_{XY}(1 - u, 1 - u) \tag{3.20}$$

These propositions are used to prove that the proposed copula based mutual information measure satisfied scale invariant property of copula. In case of noisy data, it keeps the original subset of features stable in the dataset. Theoretical proves of this statement is described here, whereas the simulation result is given later in Section 3.4.

Theorem 2. *Let, the functions α and β both be strictly increasing according to random variables X and y respectively. Copula based mutual information $I_c(\alpha(X), \beta(Y))$ is same as $I_c(X, Y)$*

Proof. From Equations (3.8) and (3.17), we get the copula mutual information of $I_c(\alpha(X), \beta(Y))$,between $\alpha(X)$ and $\beta(Y)$ to have the form:

$$\begin{aligned} & I_{c(\alpha(X),\beta(Y))}(u,v) = -H_{c(\alpha(X),\beta(Y))}(u,v) \\ & = -\int_0^1 \int_0^1 -c_{(\alpha(X),\beta(Y))}(u,v) \\ & \log c_{(\alpha(X),\beta(Y))}(u,v))\, du\, dv \\ & = \int_0^1 \int_0^1 c_{(X,Y)}(u,v) \log c_{(X,Y)}(u,v)\, du\, dv \\ & = -H_{c(X,Y)}(u,v) \\ & = I_{c(X,Y)}(u,v) \end{aligned} \tag{3.21}$$

□

Thus, copula distribution function of two increasing functions remain same with copula function of two random variables.

Theorem 3. *Consider, the functions α is strictly increasing and β is strictly decreasing according to random variables X and Y respectively.Copula based mutual information $I_c(\alpha(X), \beta(Y))$ is same as $I_c(X, Y)$*

Proof. As, α is strictly increasing and β is strictly decreasing function, their copula distribution function will be as Equation (3.18) and hence their copula density function is:

$$\begin{aligned} c_{(\alpha(X),\beta(Y))}(u,v) &= \frac{\partial^2 C_{(\alpha(X),\beta(Y)}}{\partial u \partial v} \\ &= \frac{\partial^2 (u - C_{XY}(u, 1-v))}{\partial u \partial v} \\ &= c_{XY}(u, 1-v) \end{aligned} \tag{3.22}$$

From Equation (3.22), copula mutual information $I_c(\alpha(X), \beta(Y))$, in this case is given by

$$\begin{aligned}
&I_{c(\alpha(X),\beta(Y))}(u,v) = -H_{c(\alpha(X),\beta(Y))}(u,v) \\
&= -\int_0^1 \int_0^1 -c_{(\alpha(X),\beta(Y))}(u,v) \\
&\log c_{(\alpha(X),\beta(Y))}(u,v))\, du\, dv \\
&= \int_0^1 \int_0^1 c_{X,Y}(u,1-v) \\
&\log c_{X,Y}(u,1-v)\, du\, dv \\
&= \int_0^1 \int_0^1 c_{X,Y}(u,p) \log c_{X,Y}(u,p)\, du\, dp \\
&[\text{By changing variable},(1-v=p)] \\
&= -H_{c(X,Y)}(u,p) \\
&= I_{c(X,Y)}(u,p) \\
&= I_{c(X,Y)}(u,v)
\end{aligned} \tag{3.23}$$

□

Similarly, if α is strictly decreasing and β is strictly increasing function of X and Y respectively, their copula mutual information of $I_c(\alpha(X), \beta(Y))$ can be shown to satisfy the relation:

$$I_{c(\alpha(X),\beta(Y))}(u,v) = I_{c(X,Y)}(u,v) \tag{3.24}$$

Theorem 4. *Consider, the functions α and β both to be strictly decreasing according to random variables X and y respectively. Copula based mutual information $I_c(\alpha(X), \beta(Y))$ is same as $I_c(X, Y)$*

Proof. Now, when both α and β are strictly decreasing functions, their copula distribution function will be as in Equation (3.20) and hence their copula density function is given by:

$$\begin{aligned}
c_{(\alpha(X),\beta(Y))}(u,v) &= \frac{\partial^2 C_{(\alpha(X),\beta(Y)}}{\partial u \partial v} \\
&= \frac{\partial^2 (u+v-1-C_{XY}(1-u,1-v))}{\partial u \partial v} \\
&= c_{XY}(1-u,1-v)
\end{aligned} \tag{3.25}$$

From Equation (3.25), copula mutual information of $I_c(\alpha(X)\beta(Y))$, when α and β

are both strictly decreasing functions of X and Y respectively, is described below.

$$
\begin{aligned}
& I_{c(\alpha(X),\beta(Y))}(u,v) = -H_{c(\alpha(X),\beta(Y))}(u,v) \\
& = -\int_0^1 \int_0^1 -c_{(\alpha(X),\beta(Y))}(u,v) \\
& \log c_{(\alpha(X),\beta(Y))}(u,v))\, du\, dv \\
& = \int_0^1 \int_0^1 c_{X,Y}(1-u,1-v) \\
& \log c_{X,Y}(1-u,1-v)\, du\, dv \\
& = \int_0^1 \int_0^1 c_{X,Y}(p,q) \log c_{X,Y}(p,q)\, du\, dv \\
& [\text{By changing variable},(1-u=p,1-v=q)] \\
& = -H_{c(X,Y)}(p,q) \\
& = I_{c(X,Y)}(p,q) \\
& = I_{c(X,Y)}(u,v)
\end{aligned}
\tag{3.26}
$$

□

Thus we have shown that copula based mutual information of random variables $\alpha(X)$ and $\beta(Y)$ are here same as that of the random variables X, Y, where α and β are increasing or decreasing function of X and Y. Since the proposed *CBFS* method is strictly based on this copula based mutual information, the selected feature set with this method also remains the same under such transformations. The proof of correctness of the optimal solution is provided below.

Theorem 5. (Proof of Correctness of *CBFS*) *Let, a feature subset* $F_s = (f_1, f_2, \ldots, f_j, f_{j+1})$ *be obtained from feature set* F *using CBFS. Our claim is that the feature subset* F_s *is the optimal feature set.*

Proof. Let us consider the claim of the theorem is false.
Let, another optimal feature subset (F'_s) which has most of the frequent features with our claim feature set F_s.
The new optimal feature subset is, $F'_s = \{f_1, f_2, \ldots, f_j, f_x\}$ which is similar with feature subset F_s but except the feature(f_{j+1}). The feature f_{j+1} is replaced with feature f_x in new optimal feature subset F'_s.
CBFS selects features according to minimum correlation with all other non selected features and maximum correlation with the class label. So, It is true that,

$$
I_{c(f_k \in (F_N - F_s))}(f_{j+1}; f_k) \leq I_{c(f_k \in (F_N - F'_s))}(f_x; f_k) \tag{3.27}
$$

$I_{C(X,Y)}$ is the copula mutual information between random variables X and Y. Hence,

CBFS will exchange feature f_x with the feature f_{j+1}.
Then, new optimal solution will be, $F_s = \{f_1, f_2, \ldots, f_j, f_{j+1}, \ldots\}$, which is contradiction of our claim. □

Now, we extend the above method to a regularized copula based feature selection (*RCFS*), it is described below.

3.3.4 Feature Selection with *RCFS*

Objective Function: According to [130], an average estimation error occurred due to bivariate mutual information based dependency methods, (*MIFS, DISR, and MRMR*). Most of the methods in feature selection, discussed in Section 3.1, are dataset dependent and cease in locally optimum solutions. In this chapter, a copula based multivariate dependency measure with l_1 regularized is employed in feature selection. Multivariate copula based dependency is computed between $f_i \in (F - S)$ and $f_t \in S$ (to reduce the redundancy between them) as redundancy metric. Bivariate copula dependency is computed between class label Y and f_i as relevancy metric. Here, F and S denotes the whole feature set and selected feature subset respectively. *RCFS* uses a forward selection search to select one single feature in each step, using multivariate copula-based dependency instead of the classical information measure. Mathematically, it can be expressed as follows.

After selecting $(f_1, \ldots, f_s) \in S$, we select the next feature $(f_{s+1} = f_{\text{RCFS}})$ by

$$f_{RCFS} = \arg \max_{f_i \in (F-S)} [\tau[C(f_i; Y)] - \tau[C(f_i; f_1; f_2; \cdots; f_s)] + \gamma \|\tau[C(f_i; Y)] * Var(f_i)\|_1] \tag{3.28}$$

Where, $\tau[C(f_i; f_j)] = [4 \iint_0^{+1} C(f_i; f_j)\, dC(f_i; f_j) - 1]$ is Kendall tau dependency score of three types of copula (Gaussian, Empirical, and Archimedean) between two features f_i and f_j. Here, γ represents coefficient of regularization. An overview of our proposed work is given in algorithm **1**.

Algorithm for the Regularized Copula based Feature Selection

. *RCFS* is described in Algorithm 2.

- *RCFS* takes data matrix D with class label Y. The number of features to be selected d is considered as an input parameter.
- First relevant feature is selected by maximizing the copula dependency measure between class label Y and all features inset F

- A loop is repeated $(d-1)$ times in line number 4 to 10.
- Then, Copula based multivariate dependency is employed between each non selected feature (f_i) with all selected feature set ($f_s \in S$) and kept in list M.
- Copula based bivariate dependency is employed between each non selected feature (f_i) with class label Y and kept in list R.
- Difference of two lists R and M is kept in E
- Maximize the list E and the optimal feature is merged in list S.
- Thus, an optimal feature subset is returned in list S.
- S is returned as output.

This algorithm is applied for three variants of the copula (Gaussian, Archimedean, and Empirical). The correctness proof of *RCFS* is given below.

Algorithm 2 Regularized Copula Based Feature Selection (*RCFS*)

Input: Data Matrix **D**, Target class **Y**, Number of Selected Features, **d**.
Output: Optimal Feature subset,(**S**).

Initialisation:
S = $\emptyset$, {**S** will hold sub-set feature indices.}
S[1] $\leftarrow Max_{f_i}\tau[C(f_i, Y)]$, {Maximum Relevancy}
for all $i = 0$ to $(\mathbf{d}-\mathbf{1})$ **do**
  E = $\emptyset$
  M $\leftarrow \tau[C(f_i; f_{1s}; \cdots; f_{ds})]$, {Redundancy Criterion}
  R $\leftarrow \tau[C(f_i; Y)]$, {Relevancy Criterion}
  E $\leftarrow (R - M)$
  S $\leftarrow \{\mathbf{S} \bigcup \arg\max(\lim_{f_i}\{E\})$
  F $\leftarrow F - \{f_i\}$
end for
return S

Theorem 6. (Proof of Correctness of *RCFS* Algorithm) *Let, a feature subset* $F_s = \{f_1, f_2, \ldots, f_i, f_{i+1}, \ldots, f_d\}$ *are procured from a feature set F using RCFS. Our aim is to proof the feature subset* F_s *is the optimal feature set.*

Proof. Let, consider the claim of the theorem is not true.
Their exist another optimal feature subset (F'_s) which has most of the common features with our claim feature set F'_s.
The new feature subset $F'_s = \{f_1, f_2, \ldots, f_i, f_j, \ldots, f_d\}$ which is similar with optimal feature subset F_s except f_{i+1}. The feature f_{i+1} is replaced with feature f_j in new

optimal feature subset (F'_s).
RCFS selects a feature based on minimum correlation(kendall tau) with other non selected features (f_l) to maintain minimum redundancy criterion. So, It is true that,

$$\tau_{(f_l \in (F - F_s))} C(f_{i+1}; f_l) \leq \tau_{(f_k \in (F - F'_s))} C(f_j; f_l) \tag{3.29}$$

$\tau_C(X, Y)$ is Kendall tau dependency of copula between any two random variables X and Y. Hence, *RCFS* will exchange the feature f_j with feature f_{i+1}.
Then, optimal solution will be: $F_s = \{f_1, f_2, \ldots, f_i, f_{i+1}, \ldots, f_d\}$, which contradicts our claim. □

3.3.5 Feature/Gene Selection in scRNA-seq Data using *RgCop*

Here we describe the methodology of regularized copula based feature selection on single cell RNA sequence dataset *RgCop* is described here. The complete workflow of *RgCop* is discussed below.

Workflow of *RgCop*

Fig. 3.2 provides a workflow of the whole analysis performed here. Following subsections discussed the important steps:

A. Preprocessing of raw datasets See -A of panel-'*RgCop* framework for gene selection' of Fig. 3.2. Raw scRNA-seq datasets are obtained from public data sources. The counts matrix $D \in \mathcal{R}^{c \times g}$, where c is number of cells and g represents the number of genes, is normalized using a transformation method (Linnorm) [54]. We choose that cells which have more than a thousand genes expression values (non zero values) and choose that genes which have the minimum read count greater than 5 in at least 10% among all the cells. log_2 normalization is employed on the transformed matrix by adding one as a pseudo count.

B. RgCop framework for feature selection See -B of panel-'*RgCop* framework for gene selection' of Fig. 3.2. The preprocessed data is used in the proposed copula-correlation (*Ccor*) based feature/gene selection models. First, a feature ranking is performed based on the *Ccor* scores between all features and class labels. We assume the feature having a larger *Ccor* value is the most relevant one and we include it in the selected list. Next, *Ccor* is computed between the selected relevant features and the remaining features. The feature with a minimum score is called the most essential (and not redundant) feature and included in the selected list. The process continued in an iterative way by including the most relevant and minimum redundant features in each step every time in the list. Feature selection in this way ensures the list of genes will be optimal (see proof of correctness). An

l_1 regularization term is added with the objective function to penalize the large coefficient of relevancy term. The resulting matrix with selected features is utlized for further downstream analysis.

C. Validation through clustering See panel-'validation-A' of the Fig. 3.2. We adopt the conventional clustering steps of scanpy [74] package to cluster the resulting matrix obtained from the previous step. We employed two clustering techniques (SC3 [131], and Leiden clustering [132]) for clustering the neighborhood graph of cells. To validate the clusters we utilize the Adjusted Rand Index (ARI) metric which is usually used as a measure of agreement between two partitions. We compare the ARI score of *RgCop* with different state-of-the-art unsupervised feature selection method.

D. Validation through classification See panel-'validation-A' of the Fig. 3.2. We validate the selected features by employing several classifiers to train the resulting matrix obtained from step-B of the workflow. The features are selected by several supervised feature selection algorithm and the classification accuracy are compared with *RgCop*.

E. Annotating unknown cells See panel-'validation-B' of the Fig. 3.2. For cells of the unknown type, *RgCop* can able to accurately cluster/classify the cells using the genes selected in the previous step. The filtered and preprocessed data is divided into train-test ratio 7:3 and the train set is utilized to obtain the selected features using *RgCop*. Several classifier models are trained on the train set with the selected feature set and applied to the test set. The test data with selected features are also used for clustering. This provides the validation of our approach to work in practice.

F. Marker identification We detect highly differentially expressed (DE) genes within each cluster obtained from step-C in the workflow. Here we utilized Wilcoxon Ranksum test to identify DE genes in each cluster. The top five DE genes are chosen from each cluster according to their p-values.

The objective function:

RgCop utilizes a forward selection wrapper approach to select gene iteratively from a gene set. It uses multivariate copula-based dependency instead of the classical information measure. The objective function integrates the relevancy and the redundancy terms defined using the *Ccor*. Mathematically, it can be expressed as follows.

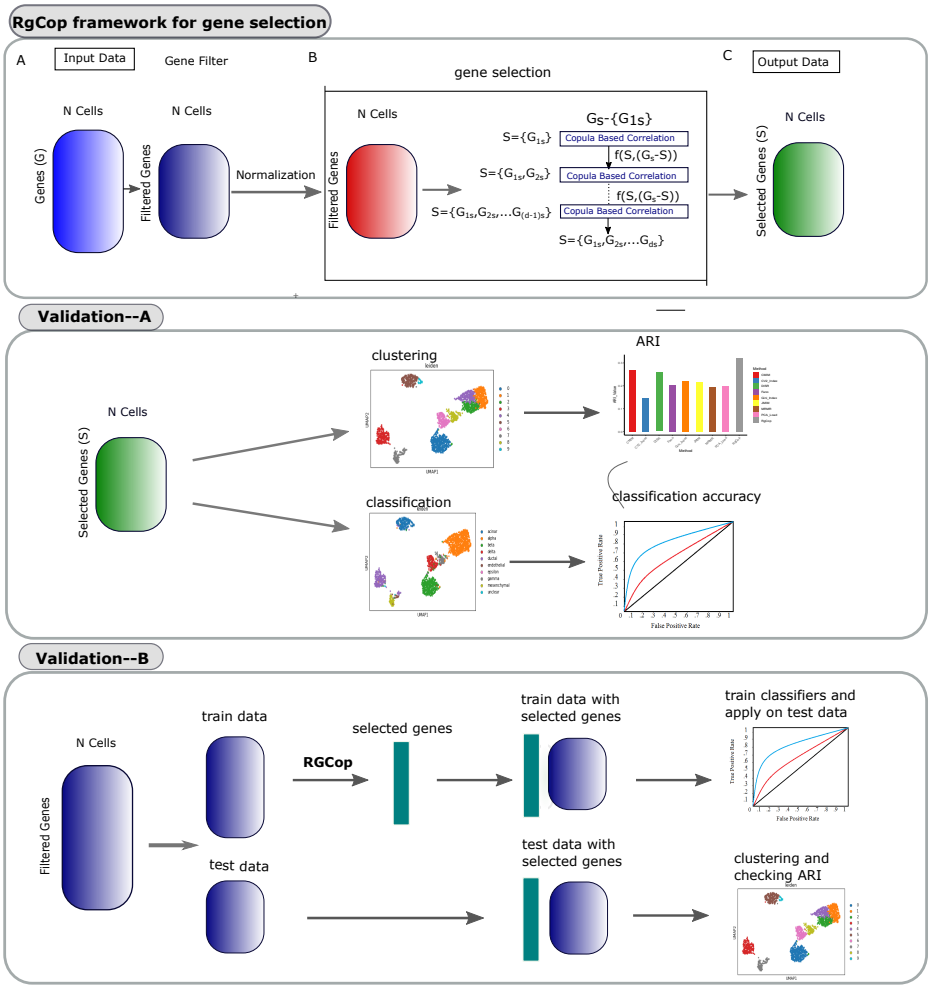


Figure 3.2: The whole workflow of the methodology: *RgCop* framework for gene selection is provided in the top panel. Clustering and classification is performed with the genes obtained from *RgCop* to validate the method (shown in middle panel). *RgCop* is validated for detection of unknown sample by splitting the data into train-test ratio of 7:3 (shown in the bottom panel). The test data is utilized for validation of the selected genes by *RgCop*.

Let us assume genes $(g_1, \ldots, g_i)$ are in the selected list G_s. The next gene $g_{i+1} \in (G - G_s)$ at $(i+1)$ iteration used the objective function

$$f = \arg \max_{g_i \in (G - G_s)} [(\tau(C_{g_i;C_D}) - \tau(C_{g_i;g_1;g_2;\cdots;g_s})) + \gamma \|\tau(C_{g_i;C_D}) * Var(g_i)\|_1] \tag{3.30}$$

Where, $\tau(C_{g_i;g_j}) = [4 \int_0^{+1}\int C(g_i; g_j)\, dC(g_i; g_j) - 1]$ is Kendall tau dependency score of Empirical copula between two genes g_i and g_j. Here, γ represents the regularization coefficient.

Theorem 7. (Proof of correctness of *RgCop*) *Suppose* $G_s = \{g_1, g_2, \ldots, g_i, g_{i+1}, \ldots g_d\}$ *denotes a subset of genes obtained from a gene set* G *using RgCop. Here* g_i *represents selected gene at iteration* i*. We claim that the set* G_s *is optimal.*

Proof. Let us prove this by the method of contradiction. If we assume the claim is not true, then there should exist some another optimal gene set G'_s. Without loss of generality, let us assume G'_s has a maximum number of initial genes (i number genes) common with G_s.
Now G'_s can be written as $G'_s = \{g_1, g_2, \ldots, g_i, g_k, \ldots, g_d\}$. So, G'_s contains $\{g_1, g_2, \ldots, g_i\}$ from G_s, but not g_{i+1}. Following our assumption g_{i+1} cannot be included in any of the optimal gene lists (G'_s has maximum i number of initial genes overlapped with G_s).
Now we claim that $k > (i+1)$. This is because k cannot be equal to $i+1$, otherwise G'_s would have $(i+1)$ genes overlapped with G_s. Similarly, $k \not\leq i$, because otherwise G'_s will contains redundant genes.
Now by the definition of our objective function (f) we can write: $f(g_k) < f(g_{i+1})$. So we can substitute g_k with g_{i+1} in the G'_s list, and the list will be still optimal. This contradicts our assumption that g_{i+1} cannot be included in any optimal list. This proves our claim.

□

The algorithm 3 describes the method of *RgCop*.

3.4 Results and Discussion

3.4.1 Simulation Parameter Settings for *CBFS*, *RCFS* and *RgCop*

Four well known mutual information-based feature selection methods: *CMIM*, *MIFS*, *DISR*, and *MRMR* to compare with the proposed method *CBFS* and *RCFS*. The reason for choosing these three works is that all the methods are based on joint mutual information and made an excellent performance in feature selection [130].

Four well known gene selection methods in scRNA-seq data are selected for comparisons: *Gini Clust* [133], *PCA Loading* [92], CV^2 *Index* and *Fano Factor* [134]. *Gini Clust* uses Gini Index in feature selection which is used in [133] for rare cell detection in scRNA-seq data. *PCA Loading* selects feature with principal component analysis, which is very common and widely used in scRNA-seq data analysis.

Algorithm 3 l_1 Regularized Copula Based Feature Selection (*RgCop*)

Input: Preprocessed Data Matrix $\mathbf{D}$, Cell Type $\mathbf{C_D}$, Number of Selected Features, $\mathbf{d}$.
Output: Optimal Feature subset,($\mathbf{G_s}$).
Initialisation:
$\mathbf{G_s} = \emptyset$, {**S** will hold sub-set feature indices.}
$\mathbf{g_{1s}} \leftarrow \arg\max_{g_i} \tau(C_{g_i,C_D})$, {Maximum Relevancy}
for all $i = 0$ to $(\mathbf{d-1})$ **do**
  $\mathbf{E} = \emptyset$
  $\mathbf{R} \leftarrow \tau(C_{g_i;C_D})$, {Relevancy Criterion}
  $\mathbf{M} \leftarrow \tau(C_{g_i;g_{1s};\cdots;g_{ds}})$, {Redundancy Criterion}
  $\mathbf{E} \leftarrow (R - M)$
  $\mathbf{G_s} \leftarrow \{\mathbf{G_s} \bigcup \arg\max(\lim_{g_i}\{E\})$
  $\mathbf{G} \leftarrow G - \{g_i\}$
end for
return $\mathbf{G_s}$

CV^2 *Index* is defined as variance to mean ratio of a variable. Features/genes having higher CV^2 Index is selected from scRNA-seq data. *Fano Factor* is a measure of dispersion among the features. It is also defined as a ratio of variance to mean of a variable. In scRNA-seq data the genes having the highest Fano factor is selected.

For Gini-Clust, the R package with the default parameter as provided in the original paper [133] is used. For PCA loading, the first three PC components are considered as the default parameter. 'praznik' R package is employed with default parameters for supervised methods (MRMR, DISR, JMIM, and CMIM). For *RgCop*, we use regularization coefficient γ as 0.3 (see simulation result on synthetic data in Result section). Number of selected features is user defined in all method (*CBFS*, *RCFS*, *RgCop*. In this thesis, all experiments are performed on top 100 selected features using *CBFS*, and *RCFS*, and top 500 selected genes using *RgCop*.

Multiple Classifiers and Clustering Methods

CBFS, and *RCFS* method is a filter-based feature selection approach. So, We anticipate that feature selected with the *CBFS* method has better performance with any classifier. To validate this, we have considered four widely used classifiers [135], Support Vector Machine (*SVM*), Neural Network (*NN*), Naive Bayes (*NB*) and Gradient Boosting Machine (*GBM*). The caret package in R provided all the classifier implementations.

- SVM is a discriminative supervised learning method. We configured the

SVM with l_2-regularizer and linear kernel.

- NN is also a supervised learner that emulates the human brain neuron structure. We have used the default caret NN layout to train our models.
- NB is one of the probabilistic supervised learning methods for classification. NB has shown excellent classification performance on many real datasets. It uses Bayes' theorem to compute the necessary probabilities.
- GBM is a machine learning method for classification and regression problems. It ensembles multiple weak prediction models to build a single strong learner.

The single cell clustering procedure using the Seurat [136] and SC3 R package [131] with default parameters are employed to validate selected features using *RgCop*.

3.4.2 Datasets Description

Two types of datasets are used here, the first one is a synthetic Gaussian mixture dataset, and another is real datasets. We have used four well-known classifiers to measure accuracy. In the next three subsections, multiple classifiers, synthetic Gaussian datasets, and real datasets are described respectively.

Synthetic gaussian mixture data

To test the efficacy of our algorithm, we generated eight synthetic Gaussian mixture datasets using 2, 3, 4 and 5 clusters. We used 50 relevant features and 250 irrelevant features for each dataset. We created K component of Gaussian mixture model with the 50 relevant features. The covariance matrices(Σ) were generated using the formula described below:

$$\Sigma = (\rho^{|i-j|}) \tag{3.31}$$

i, j are the row and column of the covariance matrix. We have consider $\rho = 0.5$.
The covariance matrix is kept the same for all eight synthetic datasets, and fifty relevant features are generated by variation of the mean (μ). We have assumed the above for simplicity of datasets.
White Gaussian noise [137] was added to these synthetic datasets as 250 irrelevant features. The function uses the random normal distribution function to create the normally distributed noise and adds it to the input matrix. Add.Gaussian.noise package in R is utilized here to generate Gaussian noise. The magnitude of the Gaussian noise is taken as mean 0 and standard deviation 1 for all experiment. 500 samples with $k = (2, 3, 4, 5)$ number of classes are generated for each synthetic dataset. We generated overlapping clusters and non-overlapping clusters data for

each class $k = (2, 3, 4, 5)$. The sample generation for each dataset was repeated 100 times, and average results are provided in Section 3.4.3. Synthetic Data descriptions and distribution parameters are given below in Table 3.1 and 3.2. To make the data noisy we add contaminated noise in the constructed synthetic data by using 8 : 2 mixing ratio among the classes. The following steps are considered for contamination of one synthetic data (assuming 2 class c and $\widehat{c}$ data):

- 20 percent of the samples of one class (c) (with mean μ and covariance matrix Σ) are replaced with samples generated with mean ($3\mu_{\widehat{c}} + 3$) and covariance matrix Σ of other class ($\widehat{c}$), where $\mu_{\widehat{c}}$ is the mean of the other class ($\widehat{c}$).
- The process is repeated for all eight Gaussian mixture datasets.

Table 3.1: Non-Overlapping Synthetic Gaussian Mixture Data

# Classes	Mixing Proba-bilities	# Features		Range of Means (μ)				
		Relevant	Irrelevant	Cluster_1	Cluster_2	Cluster_3	Cluster_4	Cluster_5
2	[0.6,0.4]	50	250	50 values from $(5, 15)$	50 values from $(-15, -5)$	-	-	-
3	[0.4,0.3,0.3]	50	250	Same as above	Same as above	25 values from $(5, 15)$ 25 values from $(-15, -5)$	-	-
4	[0.4,0.2,0.2,0.2]	50	250	Same as above	Same as above	Same as above	25 times 0 25 values from $(5, 15)$	-
5	[0.2,0.2,0.2,0.2,0.2]	50	250	Same as above	Same as above	Same as above	Same as above	50 values from $(15, 20)$

Table 3.2: Overlapping Synthetic Gaussian Mixture Data

# Classes	Mixing Proba-bilities	# Features		Range of Means (μ)				
		Relevant	Irrelevant	Cluster_1	Cluster_2	Cluster_3	Cluster_4	Cluster_5
2	[0.6,0.4]	50	250	50 values from $(2, 3)$	50 values from $(-3, -2)$	-	-	-
3	[0.4,0.3,0.3]	50	250	Same as above	Same as above	25 values from $(2, 3)$ 25 values from $(-3, -2)$	-	-
4	[0.4,0.2,0.2,0.2]	50	250	Same as above	Same as above	Same as above	25 times 0 25 values from $(2, 3)$	-
5	[0.2,0.2,0.2,0.2,0.2]	50	250	Same as above	Same as above	Same as above	Same as above	25 times 0 25 values from $(-3, -2)$

The synthetic gaussian mixture datasets are plotted using tSNE visualization in Fig. 3.3. The t-Distributed Stochastic Neighbor Embedding (t-SNE) is a non-linear technique for dimensionality reduction that is particularly employed to visualize of high-dimensional datasets.

Real life dataset

UCI datasets Ten continuous and discrete datasets were used for evaluation. Among them, first five are UCI repository datasets https://archive.ics.uci.

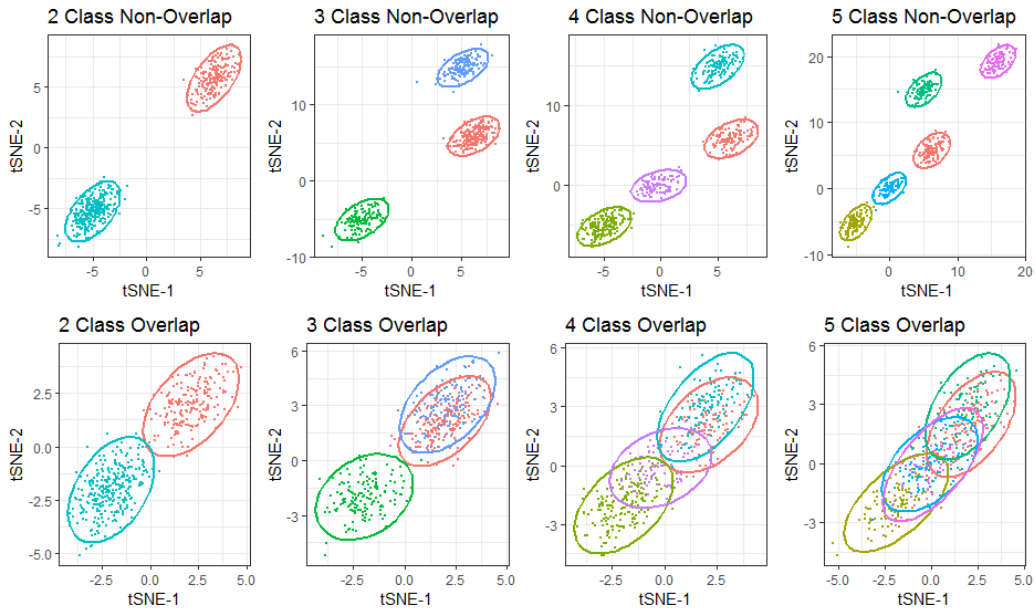


Figure 3.3: Eight synthetic Gaussian Mixture Datasets are embedded using tSNE visualization. The figures represent non-overlapping classes and overlapping classes in the respective rows.

edu/ml/datasets.php, the next two are gene microarray datasets, then the two are face image datasets, and last one is handwritten image dataset. Raw datasets were preprocessed with discretization process, [138], using which the continuous datasets with equal frequencies. A short description of the datasets is given below in Table 3.3.

Table 3.3: Summary of the real datasets used in the experiments.

Serial #	Dataset Name	#Instances	#Features	#Class
1	Arrhythmia	452	279	2
2	Musk	476	168	2
3	Libra	360	91	15
4	Semieon	1593	256	10
5	Handwritten Numerals	2000	649	10
6	Lymphoma	96	4026	9
7	Leukemia	72	7070	2
8	ORL	400	1024	40
9	warpPIE10P	210	2420	10
10	USPS	9298	256	10

- The Arrhythmia Data contains 452 samples and 279 features. It is also a

binary classification problem. The two classes are male and female.

- The dataset Musk contains 476 samples and 168 features. It is also a binary class dataset. It is a continuous type of dataset.

- The Libras Movement dataset contains 360 samples and 91 attributes as a feature, representing the coordinates of movement. It includes 15 classes, where each class references to a hand movement type in LIBRAS.

- The Semieon is a popularly used handwritten dataset which contains 1593 samples and 256 gray scale pixel values. Pixel values are regarded as features. It contains ten classes,$(0, 1, \cdots, 9)$.

- The Handwritten (HDR) consists of features extracted from handwritten numerals collected from Dutch utility maps. It has ten classes of '0-9' digits. HDR consists of total 2000 observations with 200 patterns per class. Data contains 649 different shapes for each model which are regarded as features.

- The Lymphoma Dataset [139] is an unevenly distributed data. It contains 96 sample points. It has 4026 genes expression values for each sample point as features. The target class has nine subtypes of lymphoma.

- The Leukemia dataset [140] is one of the popular gene expression datasets. It contains 72 observations. It consists of 7070 gene expression values for each observation. It is a binary class dataset, 'AML', 'ALL' are two class labels.

- The ORL face database (developed at the Olivetti Research Laboratory, Cambridge, U.K.) contains 400 images of size $112 * 92$. The dataset accommodates images of 40 persons, ten images per each person, which are at different times, lighting, and facial expressions. It has 40 class labels.
- The warpPIE10P [141] is an image database of over 40,000 facial images of 68 people. Images are captured with each person across 13 different poses, under 43 different illumination conditions, and with four different expressions. It contains 10 class labels.
- The USPS database [142] is an image database for handwritten text recognition research. Digital images of approximately 5000 city names, 5000 state

Table 3.4: A brief summary of the real scRNA sequence Dataset

Dataset Name	Features	Instances	Class
Yan	20214	90	7
Muraro	19127	2126	10
Pollen	23794	299	11
PBMC	32738	68793	11

names, 10000 ZIP Codes, and 50000 alphanumeric characters are included in the dataset. It contains 10 class labels.

Single cell RNA sequence datasets The *RgCop* study used the following single-cell RNA sequence datasets: Yan [143], Pollen [144], Muraro [145] and PBMC68k [58] (see Table 3.4). Three (Pollen, Yan, and Muraro) are downloaded from Gene Expression Omnibus (GEO) https://www.ncbi.nlm.nih.gov/geo/.The description of daatsets are discussed below.

- Yan: The dataset consists of a transcriptome of 124 individual cells from a human preimplantation embryo and embryonic stem cell. The 7 unique cell types accommodates labelled 4-cell, 8-cell, zygote, Late blastocyst, and 16-cell.[GEO under accession no. GSE36552; [143]].

- Pollen: Single cell RNA seq pair-end 100 reads from single cell cDNA libraries were quality trimmed using Trim Galore with the flags. It contains 11 cell types. [GEO accession no GSM1832359; [144]]

- Muraro: Single-cell transcriptomics was carried out on live cells from a mixture using an automated version of CEL-seq2 on live, FACS sorted cells. It contains 2126 number of cells. It is a human pancreas cell tissue with 10 cell types. The dataset was downloaded from GEO under accession no GSE85241 [145].

- PBMC68k: The dataset[58], is downloaded from 10x genomics website https://support.10xgenomics.com/single-cell-geneexpression/datasets. The data is sequenced on Illumina NextSeq 500 high output with 20,000 reads per cell.

3.4.3 Simulation Results of Feature Selection Using *CBFS*

Two alternative approaches were used to demonstrate the performance of the proposed feature selection algorithm (*CBFS*). Each approach addresses specific evaluation criteria, as outlined below.

1. To investigate whether *CBFS* produces a better informative feature subset: Feature subsets obtained from different feature selection methods were subsequently used for building their corresponding prediction models. After that, prediction accuracy was used to compare the model performance across multiple real datasets. For the synthetic datasets, the clustering accuracy was evaluated using *CBFS*.

2. To verify the scale-invariant property of *CBFS*: Gaussian noise was added to the real datasets and then tested to compare stability across the different feature selection methods.

Clustering accuracy on synthetic datasets

The clustering performance on all eight synthetic Gaussian mixture datasets was reported using the Adjusted Rand Index (ARI). The ARI can be used as a measure of agreement between the assigned clusters and the known groups. The value of the ARI is close to 0 when the clustering prediction is random, and 1 when the clustering is perfectly coherent to the known labels [146]. Table 3.5 illustrates the clustering performance evaluated on eight synthetic Gaussian mixture datasets for all four feature selection methods. As the number of clusters increases, the ARI score decreases for all methods. Also, the ARI score for the overlapping Gaussian mixture datasets is less than non-overlapping Gaussian mixture datasets. The results in Table 3.5 depict that *CBFS* outperforms all competing methods (Section 3.4) excepts 4-class and 5-class overlapping datasets, whereas MIFS and CMIM have higher ARI values than *CBFS*. The plausible reason is that *CBFS* leverages the multivariate dependency measure and scale-invariant property of the copula.

Table 3.5: Adjusted Rand Index for Synthetic Data

Datasets	Method	2-Class Data	3-Class Data	4-Class Data	5-Class Data
Non-Overlapping	*CBFS*	1 ± 0	**0.88 ± 0.1**	**0.72 ± 0.2**	**0.68 ± 0.07**
	CMIM	1 ± 0	0.83 ± 0.05	0.70 ± 0.13	0.57 ± 0.02
	MIFS	1 ± 0	0.71 ± 0.02	0.68 ± 0.04	0.61 ± 0.05
	MRMR	1 ± 0	0.70 ± 0.04	0.67 ± 0.004	0.59 ± 0.02
Overlapping	*CBFS*	**0.98 ± 0.04**	**0.78 ± 0.11**	0.64 ± 0.1	0.65 ± 0.02
	CMIM	0.97 ± 0.02	0.76 ± 0.03	0.58 ± 0.12	**0.67 ± 0.1**
	MIFS	0.96 ± 0.03	0.75 ± 0.07	**0.65 ± 0.023**	0.64 ± 0.03
	MRMR	0.97 ± 0.01	0.76 ± 0.04	0.63 ± 0.05	0.65 ± 0.07

Table 3.6: Four setups for generating simulated scRNA-seq datasets using Splatter [1]

Setups	Group Proportions (%)	Dropout rate	DE Gene Proportion (%)
S1	(10, 10, 10, 70)	0.2	40
S2	(25, 25, 25, 25)	0.5	40
S3	(10, 10, 10, 70)	0.5	10
S4	(25, 25, 25, 25)	0.2	10

3.4.4 Simulation Results of Feature/Gene Selection on scRNA-seq data Using *RgCop*

Performance in synthetic scRNA-seq data

For single-cell clustering the most common challenge is to discriminate samples between major cell types and its sub-types. Samples of similar cell types tend to overlap within one cluster, discriminating of which required sophisticated method that can extract features from overlapped samples. To explore whether *RgCop* can address this issue we apply it on simulated data generated by a widely used method called Splatter [1]. We make four experimental setup (S_1 to S_4) to comprehensively evaluate *RgCop*. Splatter is utilized to generate the data in each case.
S_1: generated four groups of 500 cells with the sample ratio of $10 : 10 : 10 : 70$. Low dropout rate is set (~ 0.2) over 2000 genes
S_2: generated four equal-sized groups of cells, each group consisting 25% of the total (500) cells, over 2000 genes at a high dropout rate (~ 0.5).
S_3: generated four groups of 500 cells, with the sample ratio $10 : 10 : 10 : 70$ over 2000 genes with a high dropout rate (~ 0.5)
S_4: generated four equal-sized groups of 500 cells over 2000 genes at a low dropout rate ~ 0.2.
The proportions of differentially expressed (DE) genes in S_1 to S_4 were selected as $40\%, 40\%, 10\%, 10\%$ respectively. The details of the simulation settings are shown in Table 3.6.

To tune the regularization parameter γ (see equation 3.30), the feature selection process is repeated for nine set of values ranging from 0 to 0.5.
$\gamma = \{0, 0.002, 0.005, 0.009, 0.02, 0.07, 0.09, 0.3, 0.5\}$. We trained random forests classifier to measure overall accuracy over 100 simulation replicates. Table 3.7 reports median accuracy for the nine γ-values in four simulation setups. High accuracy is observed for the γ-parameter in the range of $\gamma \in [0.07, 0.3]$ (see Table 3.7). The selected range of γ values are utilized for the later stage of analysis.

Table 3.7: Classification Accuracy are reported for different values of γ using *RgCop*

Method	Classifier	Setups	q-Values								
			0	0.002	0.005	0.009	0.02	0.07	0.09	0.3	0.5
RgCop	Random Forest	S1	0.85	0.90	0.90	0.90	0.91	0.93	0.94	**0.95**	**0.95**
		S3	0.78	0.81	0.80	0.81	0.81	0.84	**0.87**	**0.90**	**0.89**
		S2	0.96	0.98	0.99	**0.99**	**0.99**	**0.99**	**0.99**	**0.99**	**0.98**
		S4	0.86	0.90	0.91	0.91	0.90	0.90	0.90	**0.98**	**0.97**

3.4.5 Stability of *CBFS*, *RCFS* and *RgCop*

This section represents that the proposed method has the potential to select features from noisy data. As discussed in this chapter [80], the copula has the power to preserve the same dependency measure in transformed noisy features. The transformation of features may occur due to technical noise, which is much more common in real-life datasets. In that case, existing feature selection algorithms will fail on the noisy structure of the data, which results in poor performance.
white Gaussian noise with a mean (μ= 0) and standard deviation 1 is mixed to each gene/feature of a dataset. *Add.Gaussian.noise* package in R was used to generate Gaussian noise. Next, the top 100 features using *CBFS*, *RCFS*, and top 500 features using *RgCop* were selected with each noisy dataset. The percentage of matching features was employed as a scale invariance metric. The number of matching features is the intersection of the most informative feature subset from a noisy dataset with the original optimal feature subset. The matching feature score (percentage) is defined below:

$$Simscore = ((n - r)/n) * 100 \tag{3.32}$$

n is the total number of features in a dataset, and r represents the number of discrepancies between the feature subsets of original and noisy data.
Fig. 3.4 reports a correlation plot of the similarity score among all the competing methods in ten real datasets. For each case, we perform 100 trials and compute Kendall tau correlation among the similarity scores returned by all the methods.
It can be observed from the figure that the correlation between the *CBFS* and CMIM is lesser than the other competing pairs, which represents *CBFS* has a higher similarity score than the well known CMIM method. The possible reason behind this is the scale invariant property of copula. Thus, the number of informative feature subset almost remains the same as the original feature subset for a noisy dataset using *CBFS*, which results in a high similarity score.
Table 3.8 reports the number of indistinguishable features subset between original and noisy datasets among all the methods. The result depicts that *RCFS* surpasses all other state-of-art in stable feature selection for noisy datasets. *RCFS* obtains the highest percentage of similarity in Arrhythmia and Musk datasets.

Table 3.8: Comparison of stability performance among different methods. Table shows the similarity score (*S_Score*) value of each method in noisy data

Dataset	DISR	MIFS	MRMR	RCFS_A	RCFS_E	RCFS_G
Arrhythmia	38	25	12	45	55	**56**
Musk	49	52	51	61	**63**	65
Lymphoma	15	26	28	**55**	50	53
Leukemia	11	18	31	49	**52**	45

For *RgCop*, We perform 5 iterations, each contains 100 such experiments. For a competing method, each trial gives 100 scores for one dataset and the median of these scores are shown in Fig. 3.7. Each row of the figure shows bar plots of the median values for three scRNA-seq datasets. It can be observed from the figure that *RgCop* achieves better *SimilarityScore* for all the datasets, particularly for small sample data.

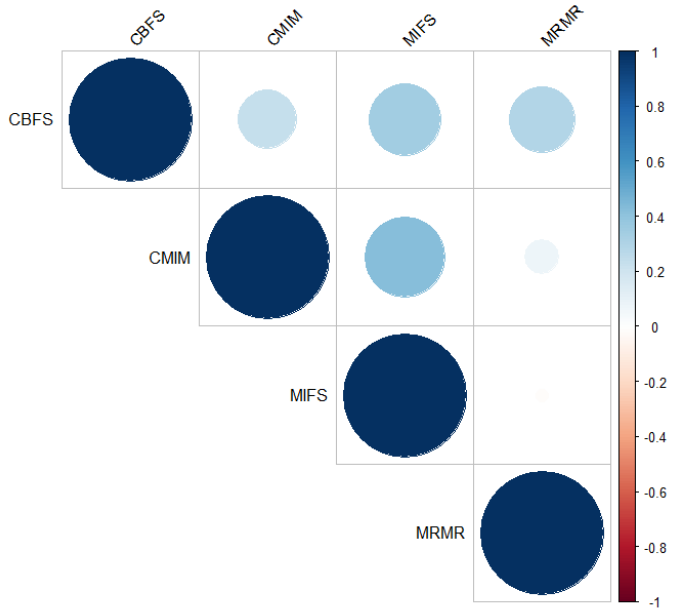


Figure 3.4: Correlation plot of similarity score for all four methods on ten datasets.

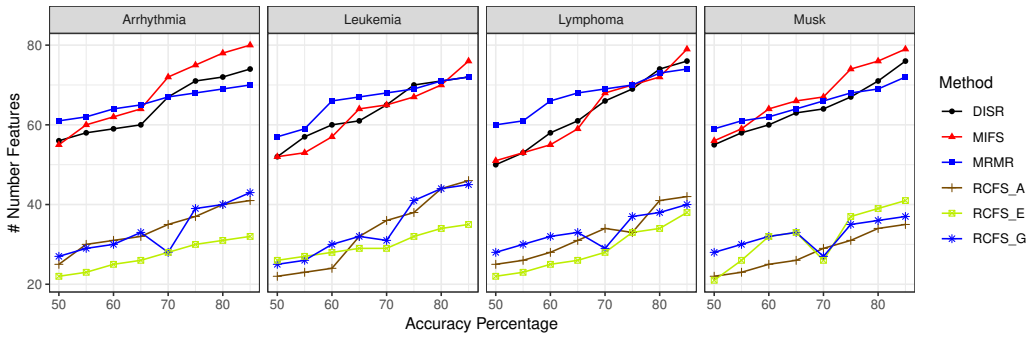


Figure 3.5: Figure shows the comparisons among the different methods. The Y axis denotes number of required features while X axis represents Accuracy Percentages.

Table 3.9: Classification results on Real Datasets using Supervised Methods

Classifier	Dataset Name	MRMR	DISR	JMIM	CMIM	*RgCop* ($\gamma = 0.3$)
GBM	Muraro	0.90 ± 0.05	0.89 ± 0.04	0.89 ± 0.010	0.85 ± 0.02	0.96 ± 0.009
	Pollen	0.75 ± 0.03	0.76 ± 0.01	0.75 ± 0.05	0.74 ± 0.02	0.88 ± 0.002
	Yan	1 ± 0	0.99 ± 0.02	0.98 ± 0.02	1 ± 0	0.97 ± 0.01
NNET	Muraro	0.82 ± 0.02	0.81 ± 0.01	0.81 ± 0.05	0.84 ± 0.03	0.89 ± 0.07
	Pollen	0.71 ± 0.01	0.72 ± 0.003	0.69 ± 0.02	0.70 ± 0.01	0.85 ± 0.02
	Yan	0.99 ± 0.01	0.99 ± 0.02	0.99 ± 0.03	0.98 ± 0.003	0.98 ± 0.02
SVM	Muraro	0.87 ± 0.04	0.88 ± 0.01	0.87 ± 0.011	0.89 ± 0.01	0.95 ± 0.01
	Pollen	0.74 ± 0.06	0.75 ± 0.03	0.70 ± 0.03	0.71 ± 0.01	0.87 ± 0.001
	Yan	1 ± 0	0.99 ± 0.02	1 ± 0	0.99 ± 0.04	0.98 ± 0.003

3.4.6 Comparisons with the State-of-the-art

Comparisons of *CBFS* with state-of-the-art

The classification performance of four classifiers (discussed in Section 3.4.1) using the respective feature selection methods has been reported here. Leave-group-out cross-validation with 100 repetitions, and the ratio of training to testing at 8.5:1.5 were used for evaluation purposes. In order to inspect how the number of selected features affects a classifier, the classifiers were trained separately by varying the number of top features, $top_n = (10, 20, 30, 40, 50, 60, 70, 80)$ obtained from respective feature selection methods. Fig. 3.8 explains the classification accuracy using the Gradient Boost Machine(GBM) classifier for ten real datasets. The result reports that when the number of features increases, the classification accuracy with the *CBFS* methods also increases. The plausible reason is the multivariate dependency of the copula based feature selection method. For all multi class dataset (HDR, Libra, Lymphoma, ORL, USPS, warpPie10P, and Semieon), our proposed method performs well among other methods. In three datasets (Musk, Arrhythmia, and Semieon), CMIM and MRMR perform well than *CBFS* for the small number of features.

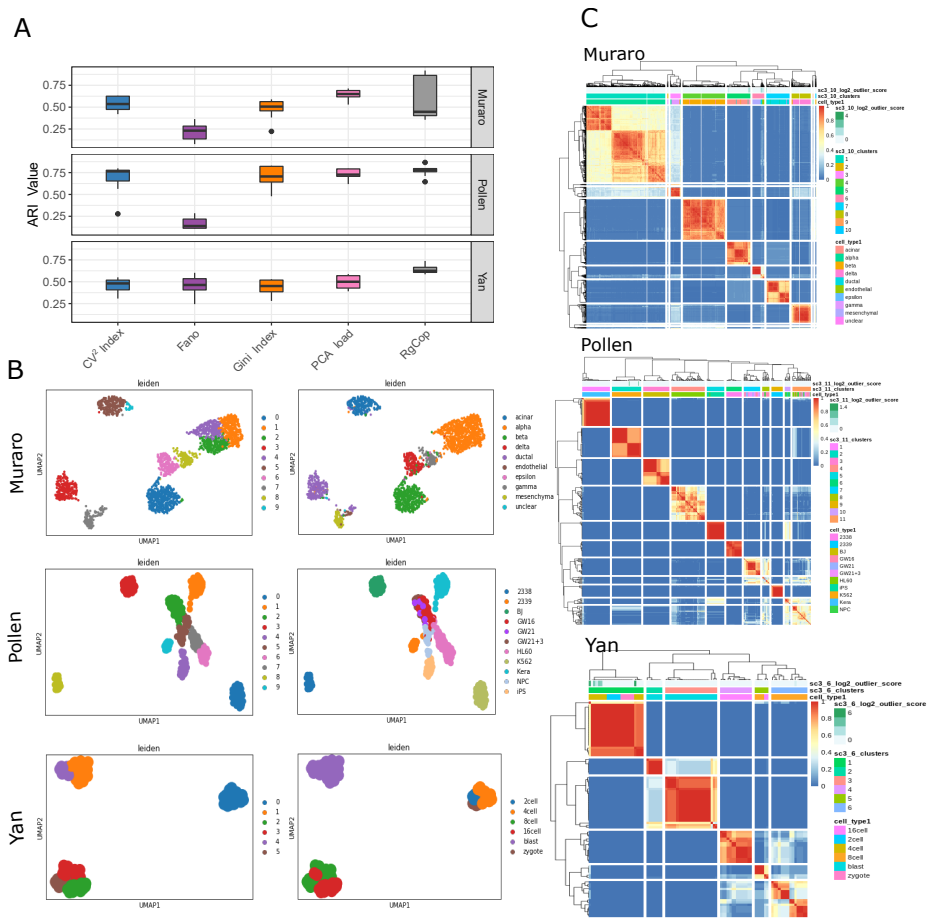


Figure 3.6: Figure shows the comparisons of clustering performance. Panel-A shows the boxplot of ARI values computed from the clustering results of each competing method. Each box represents ten ARI scores of clustering results for selected 6 sets of features ranging from 500 to 1000. Panel-B shows the 2 dimensional UMAP visualization of clustering results of three datasets for *RgCop*. Panel-C shows the consensus clustering plots of obtained clusters from *RgCop*.

Comparisons of *RCFS* with state-of-the-art

The main approach of *RCFS* is that it uses a regularized parameter (γ) in objective function. So, optimal regularized parameter finding is also important. To select the tuning parameter γ for *RCFS* we have made an analysis. For a fixed accuracy value we compute the optimal number of features for different γ values.
Three variants of regularized copula such as RCFS_A, RCFS_G, and RCFS_E have been utilized in this thesis, which denotes *RCFS* with **A**rchimedean, **G**aussian, and **E**mpirical copula respectively.
The classification performance of two classifiers (discussed in Section 3.4.1) using

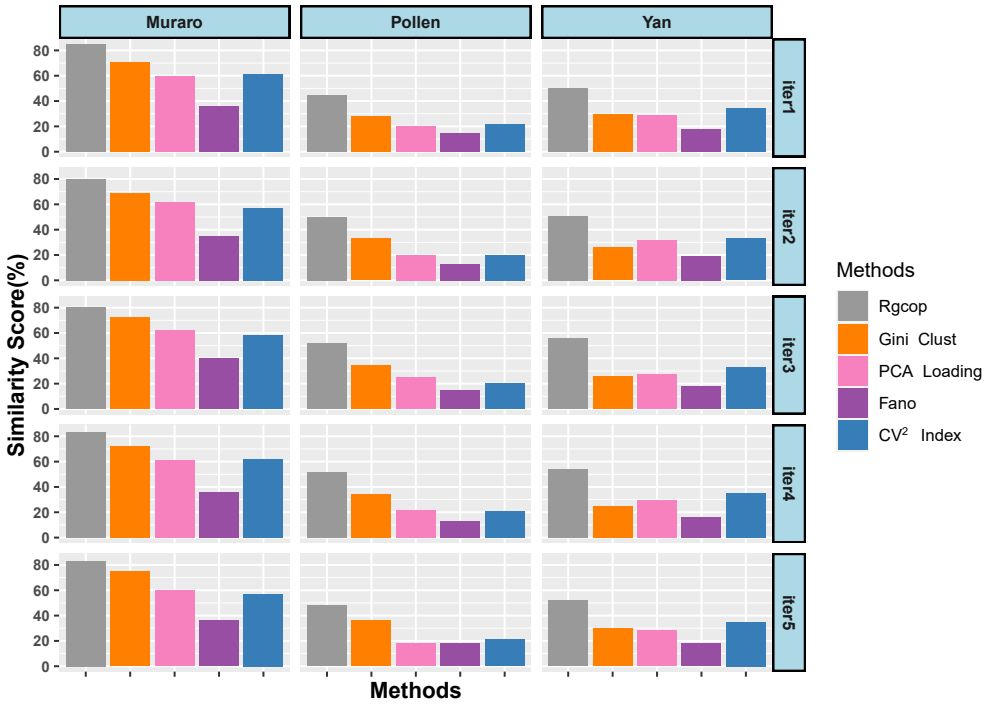


Figure 3.7: Figure shows the median of *match_score* (percentage) of five different competing methods including *RgCop*. Five iterations (iter1,iter2, iter3, iter4, iter5) are performed with 100 repetition in each iteration to compute the median of *SimilarityScore*.

the selected feature sets of the comparing methods has been reported. We use Leave-one-out cross-validation with 100 repetitions, and the keep the training to testing ratio at 80:20. Table 3.10 depicts the optimum γ values for each dataset, for each copula. We use these optimum γ values in *RCFS* for comparing the results with other methods. In order to evaluate the optimum number of selected features for different accuracy values, the classifiers were trained separately by varying the percentage of accuracy, $Accuracy\_Percentage = 50, 55, 60, 65, 70, 75, 80, 85$. The result is shown in Fig. 3.5.
We vary the accuracy values from 50 to 90% and observe the required number of optimal features in each case. It can be noticed from the Fig. 3.5 that *RCFS* achieves the accuracy value same as other methods, with a significantly lower number of selected features.

Comparisons of *RgCop* with state-of-the-art

We compared the efficacy of *RgCop* by comparing with four well known techniques for identifying highly dispersed genes in scRNA-seq data: *Gini Clust* [133], *PCA*

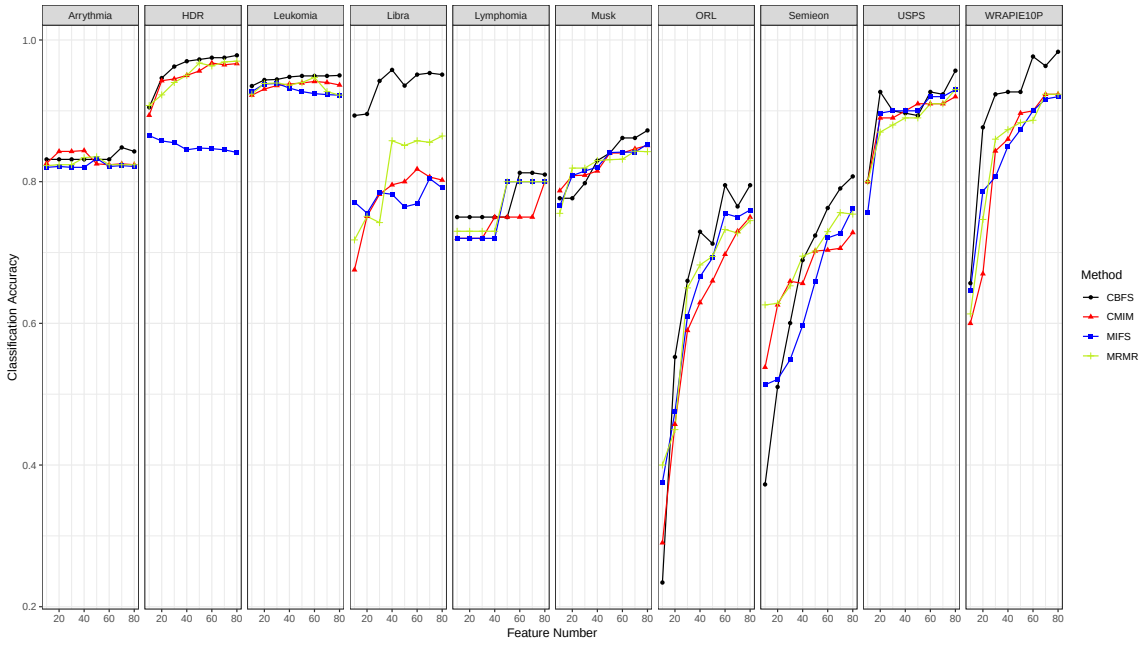


Figure 3.8: Classification accuracy for the ten datasets with Gradient Boost Machine(GBM) Classifier. There are ten boxes. Each box represents one dataset that contains four color lines (Methods). The X-axis represents the number of selected features(10, 20, 30, 40, 50, 60, 70, 80), Y-axis represents the classification accuracy values.

Table 3.10: Selection of Optimum Tuning parameter γ for three Copulas used in *RCFS*.

Dataset	RCFS_A	RCFS_E	RCFS_G
Arrhythmia	0.5	0.2	0.5
Musk	0.5	0.5	0.09
Lymphoma	0.2	0.5	0.2
Leukemia	0.09	0.5	0.5

Loading [92], CV^2 *Index* and *Fano Factor* [134]. We also compared the performance of *RgCop* with four widely used supervised feature selection techniques:CMIM [46], JMIM [47], DISR [147], MRMR [14].

Clustering performance on real dataset using unsupervised method

Here single cell Consensus clustering (SC3) method [131] is employed for clustering expression matrix with selected features. In Fig. 3.6, panel-A illustrates the boxplots of ARI Values of the clustering results on Yan, Muraro, and Pollen

datasets. We vary the number of selected features in the range from 500 to 1000 and compute the ARI scores for each method. It can be seen from the figure that *RgCop* achieves high ARI values in almost all the three datasets. For the Yan dataset, while the performance of other methods is relatively low, *RgCop* achieves a good ARI value, demonstrating the capability of *RgCop* to perform in small sample data. We also create a visualization of the clustering performance of *RgCop* in Muraro, yan, and Pollen datasets. In Fig. 3.6, panel- B shows two dimensional t-SNE plot of predicted clusters and their original labels. Panel-C of this figure shows heatmaps of cell × cell consensus matrix representing how often a pair of cells is assigned to the same cluster considering the average of clustering results from all combinations of clustering parameter [131]. Zero score (blue) means two cells are always assigned to different clusters, while score '1' (red) represents two cells are always within the same cluster. The clustering will be perfect if all diagonal blocks are completely red and off-diagonals are blue. A perfect match between the predicted clusters and the original labels can be seen from panel-B and panel-C of Fig. 3.6.

Classification performance on real dataset using supervised method

We compare *RgCop* with four well known supervised feature selection methods and compute the classification accuracy. Three widely used classifiers are considered in our work, Neural Network (*NNET*), Support Vector Machine (*SVM*), and Gradient Boosting Machine (*GBM*) for learning the expression matrix with selected features. Table 3.9 shows the average test accuracy and the corresponding standard errors over 50 runs for each of the competing methods. Results demonstrate that *RgCop* outperforms the other existing supervised feature selection methods.

Classifying test samples using the selected features

Classifying new cell samples is crucial for the scRNA-seq data analysis pipeline. Here, we address this by performing an analysis to show how the selected genes are important for discriminating the unknown cell samples. We first split the data in train-test ratio of 8:2 and use *RgCop* to select 500 most informative genes from the training set. Next, we train a random forest classifier with this data and retain the trained model. Table 3.11 shows the classification performance of the trained model on the test sample using the selected genes as the feature set. The experiment is repeated 100 times with a random split of train-test data with 8:2 ratio in each case. High classification accuracy demonstrates that the selected feature sets are equally important for discriminating the cells of the completely independent test samples.

Table 3.11: Classification Accuracy on test datasets using *RgCop*.

Datasets	Classifier	Proposed Methods
		RgCop
Yan	Random Forest	0.94 ± 0.05
Muraro		0.88 ± 0.01
Pollen		0.97 ± 0.01
PBMC		0.67 ± 0.05

Marker gene selection using *RgCop*

We have chosen marker genes (DE genes) for different cell types from the clustering results. Differentially Expressed (DE) genes are identified from every cluster using Wilcoxon rank-sum test. We use this to directly assess the separation between distributions of expression profiles of genes from different clusters. Fig. 3.9 illustrates the top five DE genes from each cluster of Pollen dataset (panel-A), and Yan dataset (panel-B). The higher expression values of the top five DE genes (shown in the heatmap of panel-A, B) for a particular cluster suggests the presence of marker genes within the selected gene sets. The results are detectable from violin plots of the expression profiles of top DE genes within each cluster (Fig. 3.9, panel-A, and -B).

Execution time

All experiments were carried out on a Linux server having 50 cores and *X*86_64 platform. As our proposed method is a wrapper-based step wise feature selection method, so it takes more time than any filter-based feature selection technique (e.g. $CV^2 index$, *GiniClust*). To check how the competing methods scale with the number of cells (and classes) we performed an analysis. We have generated simulated data (using splatter) by varying the number of cells (and classes). Four simulated data are generated with the number of cells and classes as follows: 500 cells with two classes, 1000 cells with three classes, 1500 cells with four classes, and 2000 cells with five classes. All data are generated with equal group probabilities, 2000 number of features, fixed dropout rate (0.2), and 40% DE gene proportion. 500 features are selected in each case and the runtime is compared with different competing methods. The execution time (minute) for each dataset is given in Table 3.12

3.5 Conclusions

In this chapter, we have provided a detailed description of three proposed methods of feature selection namely, *CBFS* and its two extensions *RCFS* and *RgCop*.
The main characteristics of the these methods are:

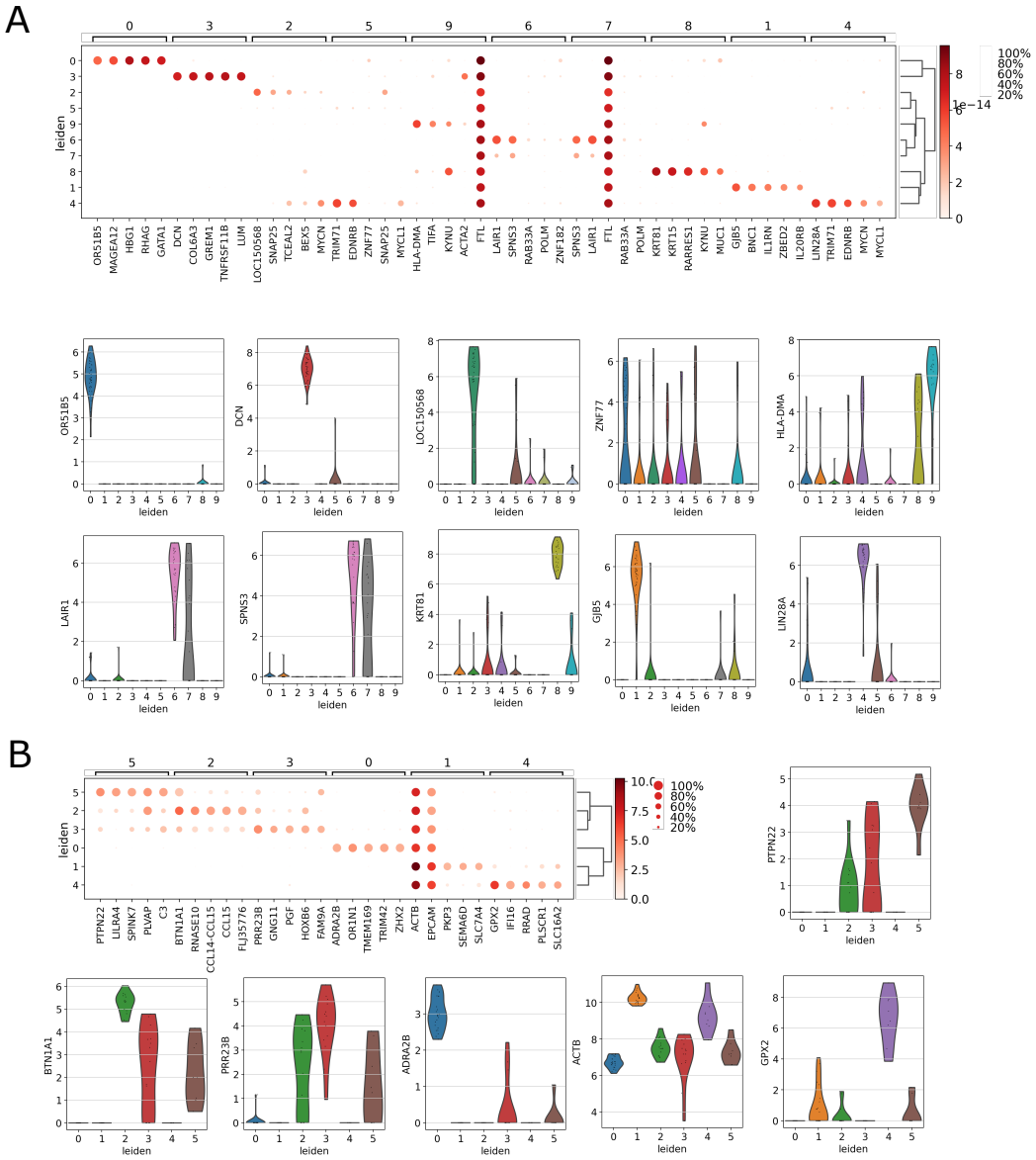


Figure 3.9: Figure shows marker analysis for Pollen dataset (panel-A), and Yan dataset (panel-B). The average expression values of the top five DE genes are shown in heatmap of panel-A, and -B. The violin plots of the expression profiles of those top DE genes within each cluster are shown in panel-A and -B.

1. Multivariate dependency- Incorporation of copula-based multivariate dependency in mutual information helps to remove the need of using bivariate dependency measures,

Table 3.12: Execution time in minute for five competing methods for *RgCop*.

Datasets	# Selected Feature	# Cells	# Class	Execution Time (in Minute)				
				RgCop	Gini Clust	CV^2 Index	Fano	PCA Loading
Data1	500	500	2	9	2	1	1	3
Data2		1000	3	13	2	1	1	7
Data3		1500	4	17	3	1	3	11
Data4		2000	5	20	5	3	5	14

2. Scale Invariance- Due to the scale invariance property of copula, all three methods achieve superior results compared to other methods on noisy datasets.

It may be noted that no regularization parameter has been considered in CBFS. This may sometimes make the algorithm susceptible to overfitting. To address this issue we have developed a regularized feature selection method called *RCFS* which employed regularization within the copula-based correlation measure. As a result, *RCFS* is robust due to the use of l_1 regularization. This is employed in three variants of the copula. *RCFS* yields the optimal number of features than competing methods on UCI machine learning datasets.

Both the methods (*CBFS* and *RCFS*) provide robust and stable feature selection technique in general. However, these are not implemented on high dimensional datasets such as single cell (scRNA-seq) datasets. Because of the high dimension and large scRNA-seq data, selecting important genes is a challenging task that has an immense effect on clustering and cell type prediction. The proposed method *RgCop* addresses this task by employing a robust and scale invariance dependence measure called copula-correlation (*Ccor*) with an l_1 regularization term. *RgCop* performs well using empirical copula with l_1 regularization. It is applied on four scRNA-seq datasets, including PBMC, and yields good results in clustering, classification of test samples, and marker gene selection. To summarize, the results of the proposed CBFS and its extensions demonstrate that these methods may be treated as important tools for machine learning researchers as well as for computational biologists to investigate the informative features in any type of high dimensional data.

All the methods in this chapter correspond to the supervised framework, where the labels of the data are easily available. However, there exist several scenarios where it is difficult to get the labels of the data samples. In that case, we need to devise appropriate algorithms which do not need the labels as input but still possess the advantages of copula namely, capturing the multivariate dependency among the features in high dimensional data and robustness due to scale invariance property. In the next chapter, we introduce such novel feature selection methods which operate in an unsupervised framework.

4

Feature Selection using Copula in an Unsupervised Framework

4.1 Introduction

With the advancement of science and technology, data has increased both in the number of samples and number of dimensions [109]. Examples of high-dimensional data include genomic data [2], text data [3], social image data [4], transcriptome data [5], etc. Over the last five decades, feature engineering has emerged as a necessary ingredient of machine learning and has become a field of study by itself [110]. Feature selection and feature extraction are two broad components of feature engineering. In this chapter, we focus on both the feature selection and extraction task in an unsupervised setup.

Nowadays, feature selection and extraction in supervised learning have been well studied. However, for unsupervised learning, the research is relatively rare. When the class information is sufficient, supervised feature selection and extraction methods usually outperform unsupervised feature selection. In this chapter, two methods are provided that address the problem of feature selection and extraction in an unsupervised setup. We have utilized copula dependency measure to sort out this problem in two valid domains. First, we develop a method called *CODC* (*CODC*: A copula based model to identify differential coexpression) to select differentially co-expressed genes in TCGA cancer datasets. Next, we have updated it to apply on large single cell RNA-seq datasets. We termed this extension, called *sc-CGconv* (**c**opula based **g**raph **conv**olution network for **s**ingle cell **c**lustering). Both the proposed methodology is utilized in unsupervised settings. In the following paragraphs, we describe the main challenges and related works for these two works.

Microarray based gene coexpression analysis has been demonstrated as an emerging field that offers opportunities to the researcher to discover coregulation patterns among gene expression profiles. Genes with similar transcriptomic expression are more likely to be regulated by the same process. Coexpression analysis seeks to identify genes with similar expression patterns which can be believed to associate

with the common biological process [148, 149, 150]. Recent approaches are interested to find the differences between the coexpression pattern of genes in two different conditions [151, 152]. This is essential to get a more informative picture of the differential regulation pattern of genes under two phenotype conditions. Identifying the difference in coexpression patterns, which is commonly known as Differential Coexpression (DC) is no doubt a challenging task in computational biology. Several computational studies exist for identifying change in gene coexpression patterns across normal and disease states [153, 154, 153, 155, 156].
For example, CoXpress [157] utilized hierarchical clustering to model the relationship between genes. Another approach called DiffCoex [158] utilized a statistical framework to identify DC modules. DICER(Differential Correlation in Expression for meta-module Recovery) [159] identified gene sets whose correlation patterns differ between disease and control samples. In [160], the authors proposed a multiobjective framework called DiffCoMO to detect differential coexpression between two stages of HIV-1 disease progression.
Most of the methods proposed scoring techniques to capture the differential coexpression pattern and utilized some searching algorithm to optimize it. Here, we have proposed *CODC*, **Co**pula based model to identify **D**ifferential **C**oexpression of genes under two different conditions. Copula [80, 84] produces a multivariate probability distribution from multiple uniform marginal distribution. It is extensively used in high-dimensional data applications. In the proposed method, first, a pairwise dependency between gene expression profiles is modeled using an empirical copula. As the marginals of gene expressions are unknown, so we have used an empirical copula to model the joint distribution between each pair of gene expression profiles. To investigate the difference in the coexpression pattern of a gene pair across two conditions, we compute a statistical distance between the joint distributions.
The above approach is updated for single cell RNA sequence data. A fundamental goal of scRNA-seq data analysis is cell type detection [67]. The most immediate and standard approach performs clustering to group the cells, which are later labeled with specific type [131, 66]. This provides an unsupervised method of grouping similar cells into clusters that facilitate the annotation of cells with specific types present in the large population of scRNA-seq data [161, 162, 163].
Several methods exist for feature/gene selection and extraction, such as GLM-PCA [64] extract features by ranking genes using deviance, M3Drop [63] selects genes leveraging the dropouts effects in the scRNA-seq data. The standard approaches (such as methods utilized in Seurat/Scanpy analysis pipeline) for feature selection/extraction failed to produce a stable and predictive feature set for higher dimension scRNA-seq data [70]. Moreover, the exiting approaches overlook the cellular heterogeneity and patterns across the transcriptional landscape, which

ultimately affects cell clustering. This motivates us to go for a robust and stable technique that can deal with the larger dimension of the single cell data while preserving the cell-to-cell variability.
Here we develop *sc-CGconv* (**c**opula based **g**raph **conv**olution network for **s**ingle cell **c**lustering), a stepwise robust unsupervised feature extraction and clustering approach that formulates and aggregates cell-cell relationships using copula correlation (Ccor), followed by a graph convolution network based clustering approach. *sc-CGconv* formulates a cell-cell graph using *Ccor* that is learned by a graph-based artificial intelligence model, graph convolution network. The learned representation (low dimensional embedding) is utilized for cell clustering. *sc-CGconv* features the following advantages. a. *sc-CGconv* works with substantially smaller sample sizes to identify homogeneous clusters. b. *sc-CGconv* can model the expression covariability of a large number of genes, thereby outperforming state-of-the-art gene selection/extraction methods for clustering. c. *sc-CGconv* preserves the cell-to-cell variability within the selected gene set by constructing a cell-cell graph through copula correlation measure. d. *sc-CGconv* provides a topology-preserving embedding of cells in low dimensional space.

4.2 Background Theory and Formal Details

Copula The term *Copula* [80] originated from a Latin word *Copulare*, which means 'join together'. The Copula is utilized in several domains in statistics to obtain joint distributions from uniform marginal distributions. Following the famous *Sklar's* theorem, Copula (C) function is specified [164] as
C is an n dimensional function, $C : [0,1]^n \rightarrow [0,1]$, satisfying the following properties:

1. $C(u_1, \cdots, u_{i-1}, 0, u_{i+1}, \cdots, u_n) = 0$, i.e., the function value is 0 if any of it argument is 0.

2. $C(1, \cdots, 1, u, 1, \cdots, 1) = u$ i.e the function value is 1 if one argument is u and all others are 1.

3. $C(u_1, \cdots, u_n)$ is n-increasing function. This means, C volume of any hyper rectangle R must be non negative, where $R = \{(x_i, y_i) : i$ is an integer and $n \geq i \geq 1\}$, for $(x_i, y_i) \in [0,1]$. This may be formally stated as follows

$$V_R(C) = \sum_{i_1=1}^{2} \cdots \sum_{i_n=1}^{2} (-1)^{i_1+\cdots+i_n} C(u_{1,i_1}, \cdots, u_{n,i_n}) \geq 0 \tag{4.1}$$

   here, $u_{j,1} = x_j$ and $u_{j,2} = y_j$, $j \in (1, \cdots, n)$.

Among several categories of Copulas [85], Clayton Copula from Archimedean family is one of the most widely used function for high dimensional datasets [165].

Clayton Copula: Let, ϕ be a strictly decreasing function such that $\phi(1) = 0$, and $\phi^{[-1]}(x)$ is the pseudo inverse of $\phi(t)$ such that $\phi^{[-1]}(x) = \phi^{-1}(x)$ for $x \in [0, \phi(0))$ and $\phi^{[-1]}(t) = 0$ for $x \geq \phi(0)$. Let $U_1, U_2, \ldots, U_n$ be the random variables having uniform marginal distributions. Then, the general family of Archimedean copula is described as,

$$\begin{aligned} &C_{Archi}(u_1, u_2, \cdots, u_n) \\ &= \phi^{[-1]}(\phi(u_1) + \phi(u_2), \cdots, +\phi(u_n)), \end{aligned} \tag{4.2}$$

where, $\phi(.)$ is called the generator function. The Clayton Copula is a particular Archimedian copula when the generator function ϕ is given by ,

$$\phi(x) = \frac{(x^{-\theta} - 1)}{\theta}, \tag{4.3}$$

with $\theta \in [-1, \infty)$.

The detail description of copula theory is given in Section 1.4 of Chapter 1.

4.3 Materials and Methodology

Here we first provide the proposed method for selecting differetially coexpressed genes using *CODC*. Next, we describe the methodology for gene selection in single cell data using '*sc-CGconv*'

4.3.1 Modeling differential coexpression using *CODC*

CODC stands for **Co**pula based model to identify **D**ifferential **C**oexpression of genes. *CODC* is applied on the TCGA gene expression data for finding differentailly coexpressed genes. Differential coexpression is simply defined as the change of coexpression patterns of a gene pair across two conditions. A straightforward method to measure this is to take the absolute difference of correlations between two gene expression profiles in two conditions. For a gene pair g_i and g_j, this can be formally stated as: $DC\_Score_{i,j}^{p1,p2} = |Sim(g_i, g_j)^{p1} - Sim(g_i, g_j)^{p2}|$, where p_1, p_2 are two different phenotype conditions. Here $Sim(g_i, g_j)^p$ signifies Pearson correlation between g_i and g_j for phenotype p. In the statistical analysis, the simple way to measure the dependence between the correlated random variable is to use copulas [166].

Here, we model the dependence between each pair of gene expression profile using

empirical copulas. As we were unaware of the distributions of expression profiles, so empirical copulas are the only choice here. Notably, we have estimated joint empirical copula density from the marginals of each gene expression profile. We have used beta kernel estimation to determine the copula density directly from the given data. The smoothing parameters are selected by minimizing the Asymptotic Mean integrated squared error (AMISE) using the Frank copula as the reference copula. The input to the copula density estimator is of size $n \times 2$, where n is the number of samples in different datasets. For each pair of samples, we estimate the empirical copula density using beta kernel estimator.
To model the differential coexpression of a gene pair, we have measured a statistical distance between two joint distribution provided by the copulas. We have utilized the Kolmogorov-Smirnov (K-S) test to quantify the distance between two empirical distributions. Value of d-statistic represents the distance here. Thus, the distance obtained for a gene pair is treated as a differential coexpression score.

4.3.2 Feature Extraction and Clustering using *sc-CGconv*

sc-CGconv takes a stepwise approach for feature extraction from the scRNA-seq data: first, it obtains a sub-sample of genes using locality sensitive hashing, next it generates a cell neighborhood graph by utilizing the copula correlation (*Ccor*) measure, and finally, a graph representation learning algorithm (here GCN) is utilized to get the low dimensional embedding of the constructed graph. A short workflow of *sc-CGconv* is depicted in the Fig. 4.1.

Structure preserving feature sampling using LSH

LSH [99] reduces the dimensionality of higher dimension datasets using an approximate nearest neighbour approach. LSH uses a random hyperplane based hash function, which maps similar objects into the same bucket. LSH is used to partition the data points (genes) of the preprocessed data matrix ($D_{(CxG)}$) into k (here $k = 10$) different buckets such that $|g_i \in G| > 2^k$, where $G = \{g_i, i = 1, \cdots, n\}$ is the set of genes in D, where $|G| = n$. A k-nn graph is formed by searching the five nearest neighbours within the bucket for each gene. Each gene is *visited* sequentially in the same order as it appears in the original dataset and is added to the *selected* list while discarding its nearest neighbours. If the visited gene is discarded previously, then it will be skipped and its neighbors will be discarded. Thus a sub-sample of genes is obtained, which is further down-sampled by performing the same procedure recursively. The number of iteration for downsampling is user defined and generally depends on the size of the data points. We use cosine distance to compute the nearest neighbours of a gene. LSHForest [167] python package is utilized to implement the whole process.

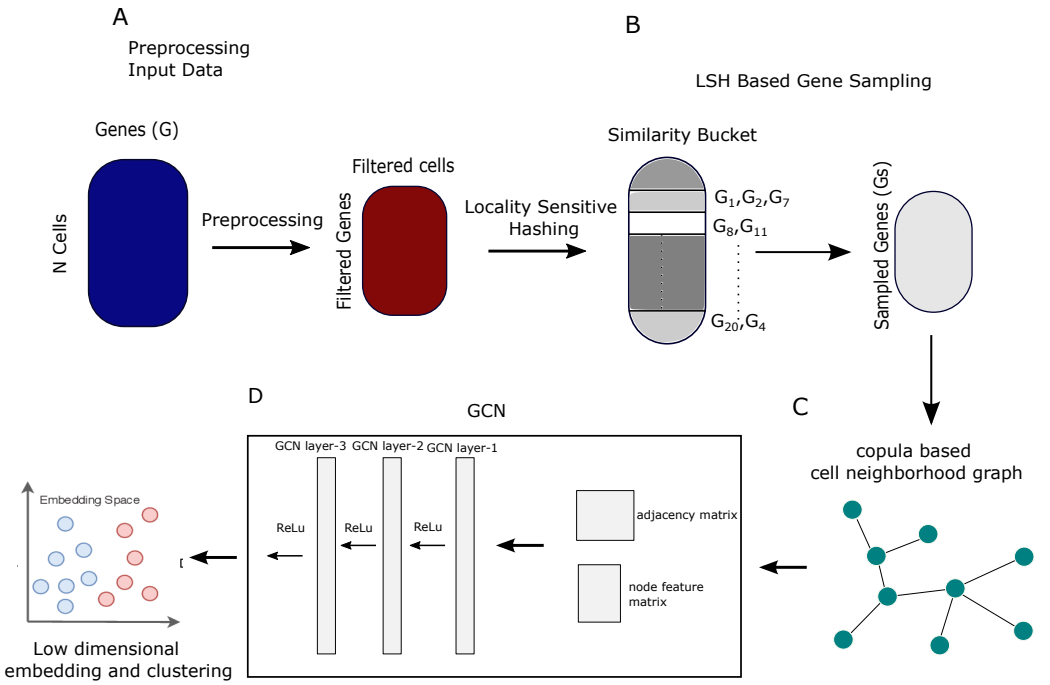


Figure 4.1: workflow of the analysis. A. scRNA-seq count matrix are downloaded and preprocessed using limnorm. B. LSH based sampling is performed on the preprocessed data to obtained a subsample of features. C. A cell neighbourhood graph is constructed using copula correlation. D. A three layer graph convolution neural network is learned with adjacencey matrix and node feature matrix as input. It aggregates information over neighbourhoods to update the representation of nodes. The final representation obtained is called as graph embedding which is utilized for cell clustering.

Thus, a subset of d number of genes, where,$(d < n)$ are obtained from the above sampling stage. These genes are considered as feature set, $F_{LSH} = \{f_s : s = 1, \cdots, d\}$ for the next feature selection stage. The next stage of feature selection using Copula is described below .

Proposed copula based correlation measure

We model the dependence between two random variables using Kendall tau(τ) [83] measure. Note that we defined Kendall tau by using copula. Kendall's tau (τ) is the measure of concordance between two variables; defined as the probability of concordance minus the probability of discordance. Formally this can be expressed as

$$\tau_{XY} = [P(x_1 - x_2)(y_1 - y_2) \geq 0] - [P(x_1 - x_2)(y_1 - y_2) \leq 0] \tag{4.4}$$

The concordance function (Q) is the difference of the probabilities between concordance and discordance between two vectors (x_1, y_1) and (x_2, y_2) of continuous random variables with different joint distribution $H1$ and $H2$ and common margins F_X and F_Y. It can be proved that the function Q depends on the distribution of (x_1, y_1) and (x_2, y_2) only through their copulas. According to Nelson [80], there is a relation between Copula and Kendall tau that can be expressed as:

$$\tau_{C(X,Y)} = \tau_{XY} = 4 \iint_{0}^{+1} C(u, v)\, dC(u, v) - 1, \tag{4.5}$$

Where, $u \in F_X(x)$ and $v \in F_Y(y)$. $\tau(C_{X,Y})$ is termed as copula-correlation (*Ccor*) in our study. Here the copula density $C(u, v)$ is estimated through the clayton copula defined in the previous section.
We have used $\tau_{C(X,Y)}$ to model the dependency between transcriptomic profiles among the cells.

Feature extraction using *sc-CGconv*

sc-CGconv takes a stepwise approach for feature extraction from the scRNA-seq data: first, it obtains a sub-sample of genes using locality sensitive hashing, next it generates a cell neighborhood graph by utilizing the copula correlation *Ccor* measure, and finally, a graph representation learning algorithm (here GCN) is utilized to get the low dimensional embedding of the constructed graph.

Cell neighbourhood graph construction

For graph construction, we rank each node (cell) according to the *Ccor* values. For a node (cell) we compute the *Ccor* values to all of its possible pairs. A k-nearest neighbour list is prepared for each node based on the *Ccor* values. A high value of *Ccor* assumes there is a high similarity between the cell pair over the transcriptomic profile, while a smaller value signifies low similarity. the output of this step is an adjacency matrix representing the connection among the cells according to the k-nearest neighbour list and a node feature matrix storing the *Ccor* values for each node pair.

Extracting node features using GCN

We have utilized graph convolution network (GCN) [168] to learn the low dimensional embedding of nodes from the cell-cell graph. Given a graph $G = (V, E)$, the goal is to learn a function of signals/features on G which takes i) A optional feature matrix $X \in M \times F$, where x_i is a feature description for every node i, ,M is the number of nodes and F is number of input features and ii) A description

of the graph structure in matrix form which typically represents adjacency matrix A as inputs, and produces a node-level output $Z \in M \times O$, where O represents the output dimension of each node feature. The graph-level outputs are modeled by using indexing operation, analogous to the pooling operation uses in standard convolutional neural networks [169]. In general every layer of neural network can be described as a non-linear function: $H^{(l+1)} = f(H^{(l)}, A))$, where $H^{(0)} = X$ and $H^{(L)} = Z$, L representing the number of layers, $f(.,.)$ is a non linear activation function like *ReLU*. Following the definition of layer-wise propagation rule proposed in [168] the function can be written as $f(H^{(l)}, A) = \sigma(\hat{D}^{-1/2}\hat{A}\hat{D}^{1/2}H^{(l)}W^{(l)})$, where $\hat{A} = A + I$, I represents identity matrix, $\hat{D}$ is the diagonal node degree matrix of $\hat{A}$, $\hat{D_{ii}} = \sum_j A_{ij}$, W represents trainable weight matrix of the neural network. Intuitively, the graph convolution operator calculates the new feature of a node by computing the weighted average of the node attribute of the node itself and its neighbours. The operation ensures identical embedding of two nodes if the nodes have identical neighboring structures and node features. We adopted the GCN architecture similar to [168], a 3-layer GCN architecture with randomly initialized weights. For the cell-cell graph, we take the adjacency matrix (A) of the neighbourhood graph and put feature matrix (X) as the node features. The 3-layer GCN performs three propagation steps during the forward pass and effectively convolves the 3rd-order neighborhood of every node.

4.4 Results and Discussions

4.4.1 Dataset Description

We first describe the TCGA cancer data which we utilized for differentially coexpressed gene selection. Next we describe the single-cell RNA-seq datasets utlized for validating the proposed '*sc-CGconv*' method.

TCGA RNA-seq data preparation

We have evaluated the performance of the proposed method in five RNAseq expression data downloaded from TCGA data portal (https://www.cancer.gov/about-nci/organization/ccg/research/structural-genomics/tcga). We have downloaded matched pair of tumor and normal samples from five pan cancer data sets: Breast invasive carcinoma (BRCA, #samples=112), head and neck squamous cell carcinoma (HNSC, #samples = 41), liver hepatocellular carcinoma (LIHC, #samples = 50),thyroid carcinoma (THCA, #samples = 59) and Lung Adenocarcinoma (LUAD, #samples = 58). For preprocessing the dataset we first take those genes that have raw read count greater than two in at least four cells. The filtered data matrix is then normalized by dividing each UMI (Unique Molecular Identifiers) counts

by the total UMI counts in each cell and subsequently, these scaled counts are multiplied by the median of the total UMI counts across cells [58]. Top 2000 most variable genes were selected based on their relative dispersion (variance/mean) with respect to the expected dispersion across genes with similar average expression. Transcriptional responses of the resulting genes were represented by the log2(fold-change) of gene expression levels from paired tumor and normal samples. A brief description of the datasets used in this article is summarized in table 4.1. Fig. 4.2-panel(A) and Panel-B represent box and violin plot of average expression value of samples for each dataset.

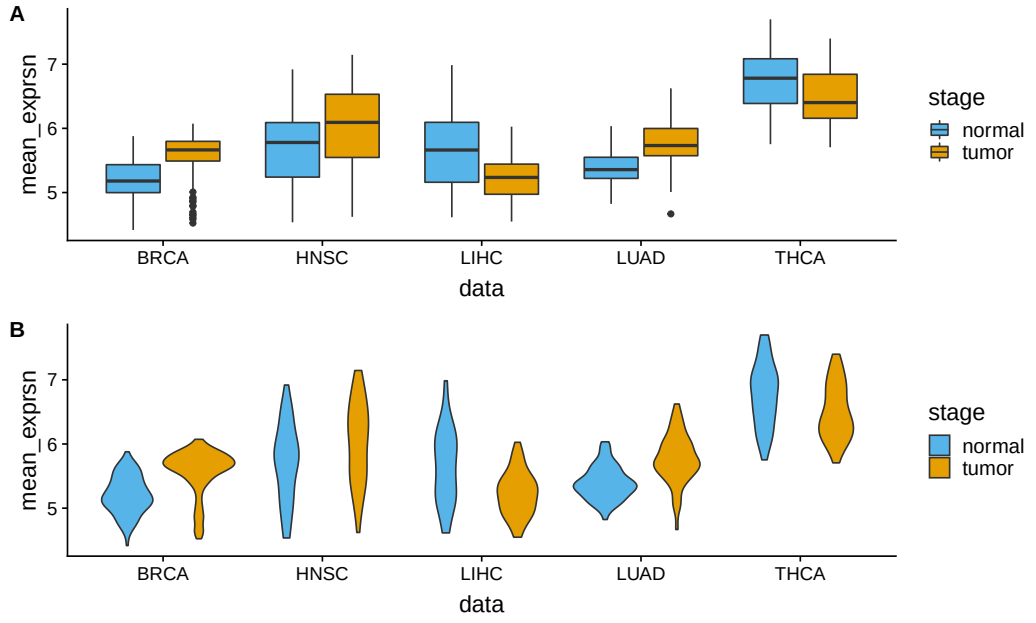


Figure 4.2: The figure describes box (panel-A) and violin plots (panel-B) of mean expression values of the used datasets.

Single-cell RNA-seq data

We used four public single-cell RNA sequence datasets downloaded from Gene Expression Omnibus (GEO) https://www.ncbi.nlm.nih.gov/geo/ and 10X genomics (https://support.10xgenomics.com/single-cell-gene-expression/datasets). Table 6.1 shows a summary of the used datasets. The details description of the used datasets is given below.

- Baron: The dataset is invoked with inDrop, a droplet-based single-cell RNA-Seq method, to determine the transcriptomes of over 12,000 individual pan-

Table 4.1: Tumor types and number of TCGA RNA-seq samples used in the analysis

Sl No.	Cancer type	#matched pair samples
1	Breast invasive carcinoma (BRCA)	112
2	Head and neck squamous cell carcinoma (HNSC)	51
3	Liver hepatocellular carcinoma (LIHC)	50
4	Thyroid carcinoma (THCA)	59
5	Lung Adenocarcinoma (LUAD)	58

creatic cells from four human donors and two strains of mice. Cells could be divided into 15 clusters that matched previously characterized cell types: all endocrine cell types, including rare ghrelin-expressing epsilon-cells, exocrine cell types, vascular cells, Schwann cells, quiescent and activated pancreatic stellate cells, and four types of immune cells. It contains 20125 number of genes and 8569 number of cells with 8 cell types.

- Klein: This dataset was generated by the droplet barcoding method with an average total read count of 20,033.40 reads in the expression matrix. A total of eight single cell data sets are submitted: 3 for mouse embryonic stem (ES) cells (1 biological replicate, 2 technical replicates); 3 samples following LIF withdrawal (days 2,4, 7); one pure RNA data set (from human lymphoblast K562 cells); and one sample of single K562 cells. The dataset was downloaded from GEO under accession no.GSE65525. The dataset contains 24175 number of genes and 2717 number of cells with 4 cell types.

- Melanoma: The dataset describes the diversity of expression states within melanoma tumors, it is obtained freshly resected samples, dissagregated the samples, sorted into single cells and profiled them by single-cell RNA-seq., It is downloaded from GEO under accession no. GSE72056. It contains 19783 number of genes and 68579 cells with 14 cell types.

- PBMC: It is downloaded from https://support.10xgenomics.com/single-cell-geneexpression/datasets. The data is sequenced on Illumina NextSeq 500 high output with 20,000 reads per cell. It contains approx ~68k number of cell with 11 cell types.

Table 4.2: A brief summary of the dataset used here

Dataset	Dataset Descrition	#Features	#Instances	#Class
Baron [170]	Human pancreas cell	20125	8569	8
Klein [171]	Mouse Embryo Cell	24175	2717	4
Melanoma [172]	Human Tumor Cell	19783	68579	14
PBMC68k [58]	Human Blood tissue	32738	68793	11

4.4.2 Results on the detection of DC gene pair using *CODC*

Detection of DC gene pair

Differential coexpression between a gene pair is modeled as a statistical distance between the joint distributions of their expression profiles in a paired sample. Joint distribution is computed by using empirical copula which takes expression profile of a gene as marginals in normal and tumor sample. The K-S distance, computed between the joint distribution is served as differential coexpression score between a gene pair. The score for a gene pair (g_i, g_j) can be formulated as: $DC\_Copula(g_i, g_j) = KS\_dist(e.c(g_i^{tumor}, g_j^{tumor}), e.c(g_i^{normal}, g_j^{normal}))$, where KS-dist represents Kolmogorov-Smirnov distance between two joint probability distribution, e.c represents empirical copula, g_i^P represents the expression profile of gene g_i at phenotype P. For each RNA-seq data, we have computed the *DC_Copula* matrix, from which we identify differentially coexpressed gene pairs.

To know how the magnitude of differential coexpression is changing with the score we plot the distribution of correlation values of gene pairs with their scores in Fig. 4.3. The figure also shows the number of gene pairs having positive and negative correlations in each stage (normal/tumor). It can be noticed from the figure that high scores produce differentially coexpressed gene pairs having a higher positive and negative correlation. We collected the gene pairs having the score greater than 0.56 and plot the correlations values in Fig. 4.4. This figure shows plots of all gene pairs having a positive correlation in normal and the negative correlation in tumor (shown in the panel-A) and vice-versa (shown in the panel-(B)). The density of the correlation values is shown in panel-C and panel-D for each case. In Fig. 4.5 we create a visualization of top differentially coexpressed gene pairs in BRCA data which shows a strong positive correlation in tumor stage and negative correlation in normal stage. The Figure shows a heatmap of binary matrix constructed from the expression data of those gene pairs in tumor and normal stages. When the expression values showing the same pattern for a gene pair it is assumed 1, while 0 representing a non-matching pattern. From the Figure, it is quite understandable that most of the entry in the normal stage is 0 (non-match) while in tumor stage is 1 (match).

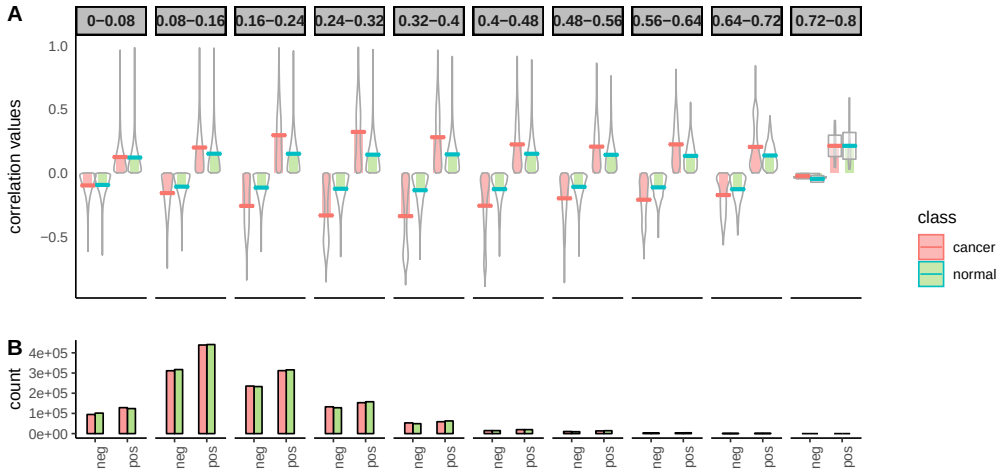


Figure 4.3: The Figure shows the distribution of correlation values in normal and cancer samples of BRCA data with the DC_Copula score. Panel-A shows the distribution for different DC_Copula scores. Here, four pirate plots are shown in each facet, two for positive and two for negative correlations. The violins in each facet represent the distribution of positive and negative correlations of gene pair in normal and cancer samples. Panel-B shows a bar plot representing the number of positive and negatively correlated gene pairs in normal and cancer samples in each facet.

Stability performance of *CODC*

To prove the stability of *CODC* we have performed the following analysis:
First, we add Gaussian noise to the original expression data of normal and cancer sample to transform these into noisy datasets. We use the rnorm function of R to create normally distributed noise with mean 0 and standard deviation 1 and we add this into the input data. We have utilized BRCA data for this analysis.
First, we compute the K-S distance and then obtain *DC_Copula* matrix for both original and noisy datasets. Let us denote these two matrices as D and D'.
The usual way is to pick a threshold t for D (or D') and extract the gene pair (i,j) for which $D(i,j)(orD'(i,j)) \geq t$. First, we set t as the maximum of D and D', and then decreases it continuously to extract the gene pairs. For each t, we observe the number of common gene pairs obtained from D and D'. Fig. 4.6 shows the proportion of common genes selected from D and D' for different threshold selection and different level of noise. Theoretically, *CODC* produces D with scores no more than D' (See the Section 3.3.3 for details). So, it is quite obvious that the number of common genes increased with a lower threshold value. From the property (See the Section 3.3.3 for details), it can be noticed that the scores in D get

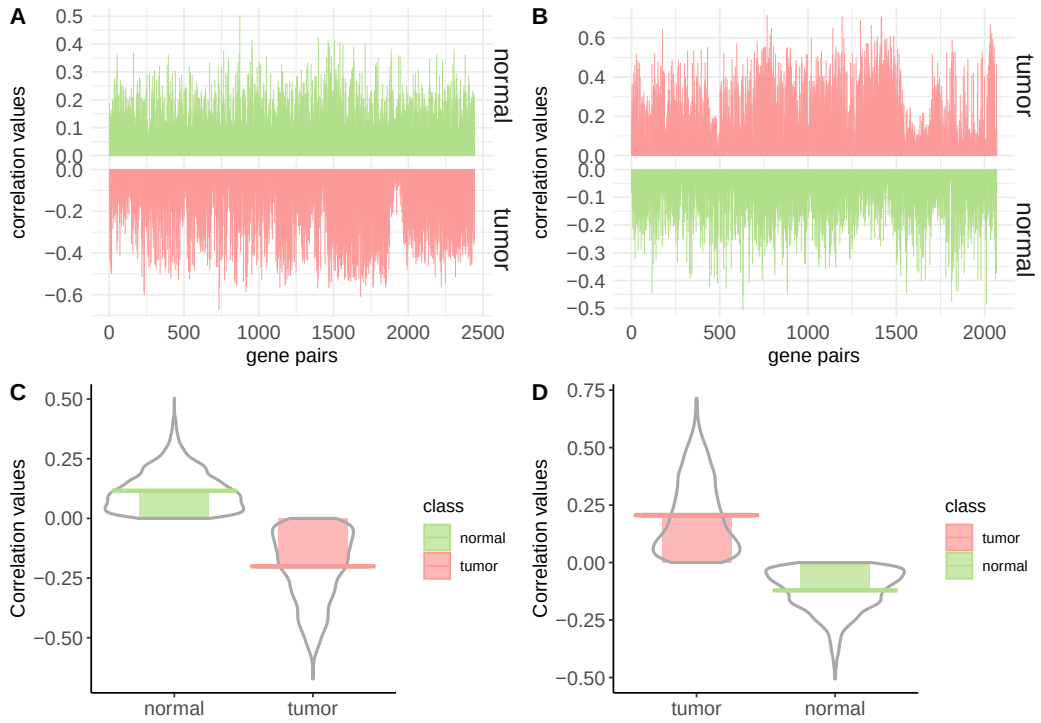


Figure 4.4: The Figure shows visualizations of gene pairs having DC_copula score greater than 0.56. Panel-A and Panel-B show the visualization of correlation values of gene pair having a positive correlation in normal and negative correlation in tumour and vice-versa, respectively. Panel-C and Panel-D represent the distribution of correlation values according to panel-A and panel-B, respectively.

preserved in D'. So, it is expected that obtained gene pairs from original data are also preserved in noisy data. Fig. 4.6 shows the evidence for this case. As can be seen from the figure that even the noise label is 80%, for threshold value above 0.25 more than 55% of the gene-pairs are common between noisy and original datasets.

Execution time: Competing methods took approximately twenty minutes on the TCGA dataset compared to only ~5 minutes by *CODC*. All experiments were carried out on a PC having an Intel Core i7-3770 3.40 GHz processor and 32GB of RAM.

Comparisons with competing methods

For comparison purpose, we have taken three competing techniques such as Diffcoex, coXpress, and DiffCoMO and compared them with our proposed method.

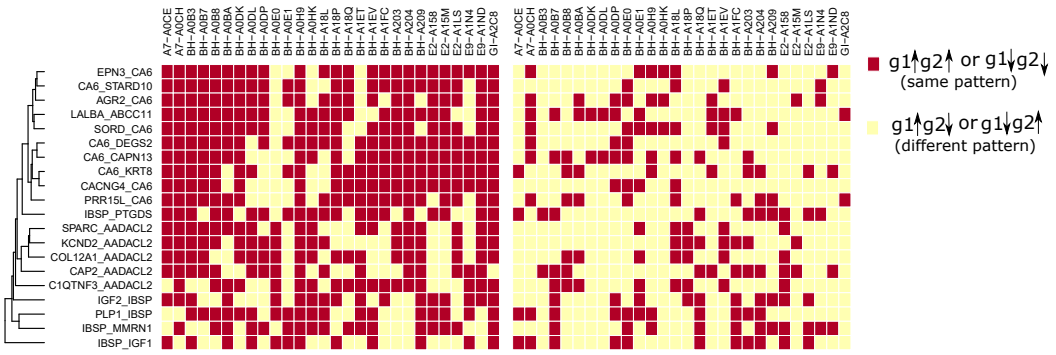


Figure 4.5: The Figure shows a heatmap representation of binary matrix constructed from the expression matrix of top differentially co-expressed gene pairs in normal and tumor stages. Expression values of a gene pair showing the same pattern are indicated as 1 and showing a different pattern is indicated as 0 in the matrix. The columns representing differentially co-expressed gene pairs while rows are the samples of BRCA data.

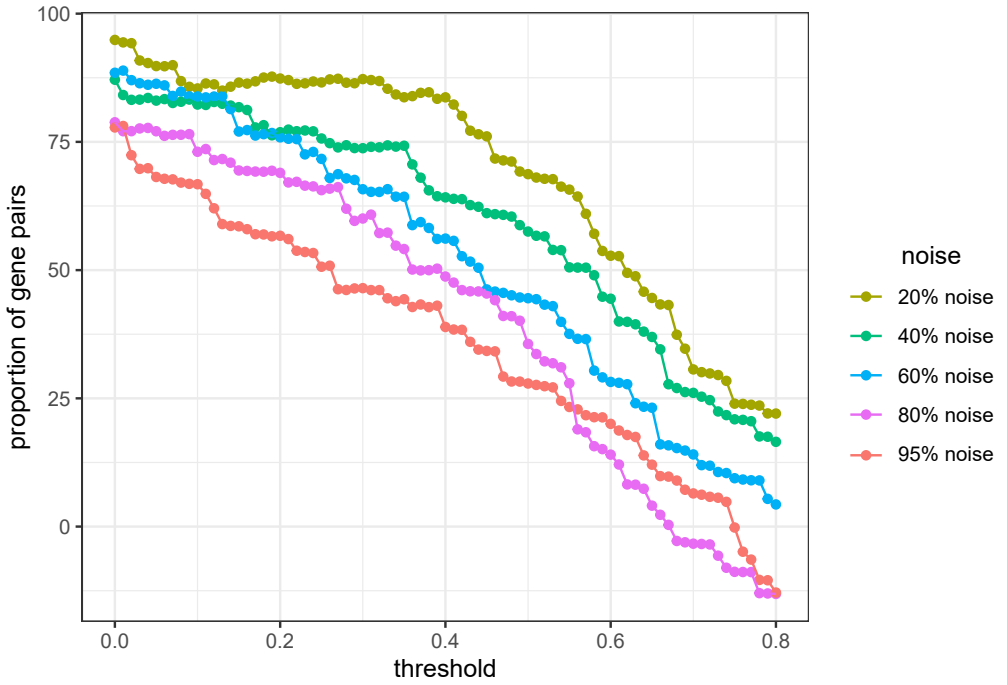


Figure 4.6: The proportion of common gene pairs obtained from noisy and original dataset with different threshold values and different noise level.

All these methods are extant DC based, which look for gene modules with altered coexpression between two classes. DiffCoEx performed hierarchical clustering on the distance matrix complied from correlation matrices of two phenotype stages. CoXpress detect correlation module in one stage and find the alternation of the correlation pattern within the module in other class. DiffCoMO uses the multiobjective technique to detect differential coexpression modules between two phenotype stages. We have made two approaches for comparing our proposed method with competing methods. We first compare the efficacy of these methods for detecting differential coexpressed gene pairs and next compare the modules identified in each case. For the first case, we take top 1000 gene pairs having high *DC_Copula* scores from the DC matrix, and perform classification using normal and tumor samples. Expression ratio of each DC gene pairs from the expression matrix was taken and compiled a $n \times 1000$, where n represents the number of samples in each data. For the other three methods, we have also selected the same number of differentially coexpressed gene pairs for the classification task. Table 4.3 shows some parameters we have used for the selection of the gene pairs. For CoXpress, first, we have used 'cluster.gene' and 'cuttree' function with default parameters provided in the R package of *CoXpress* to get the gene clusters according to the similarity of their expression profiles. These groups are then examined by the *coXpress* R function to identify the differentially coexpressed modules by comparing with the t-statistics generated by randomly resampling the dataset 10,000 times for each group. We have taken top 10 modules based on the robustness parameter, which tells the number of times that the group was differentially coexpressed in 1000 randomly resampled data. Now we have selected 1000 gene pairs randomly from those modules. For DiffCoEx method, we collected the DC gene pairs before partitioning them in modules. We used the code available in the supplementary file of the original paper of DiffCoEx, to get the distance score matrix which is used in the hierarchical clustering for module detection. We sort the score of the distance matrix and pick top 1000 gene pairs based on the scores. For DiffCoMO we use the default parameters to cluster the network to obtained differentially coexpressed modules. As it utilized multiobjective method, so all the Pareto optimal solutions of the final generation is taken as selected modules. We then choose 1000 gene pairs randomly from the identified modules. Classification is performed by treating normal and tumor samples as class label. A toy example of the comparison is shown in Fig. 4.7. Please note that all these methods are meant for differentially coexpressed module detection. So, for comparison, we collected the DC gene pairs before partitioning them in modules. We train four classifiers Boosted GLM, Naive Bayes, Random Forest and SVM with the data and take the classification accuracy. The classification results are shown in the Fig. 4.8. It can be noticed from the figure that for most of the dataset proposed method

achieved high accuracy compare to the other methods.

Table 4.3: table shows the different parameters/threshold we have used for selecting differentially coexpressed gene pairs for other methods

Method	No of gene pairs selected	Parameters used
CoXpress	1000	Used cluster.gene and cutree function with corr.coef threshold 0.6 and cutting height of hierarchical tree h=0.4. Robustness parameter threshold = 800
DiffCoEX	1000	Used Spearman correlation to compute adjacency matrix for each phenotype condition. Use default soft thresholding parameter $\beta = 6$ for computation of distance score matrix.
DiffCoMO	1000	No of modules (population size) is taken as 50 and the number of generation is 200.

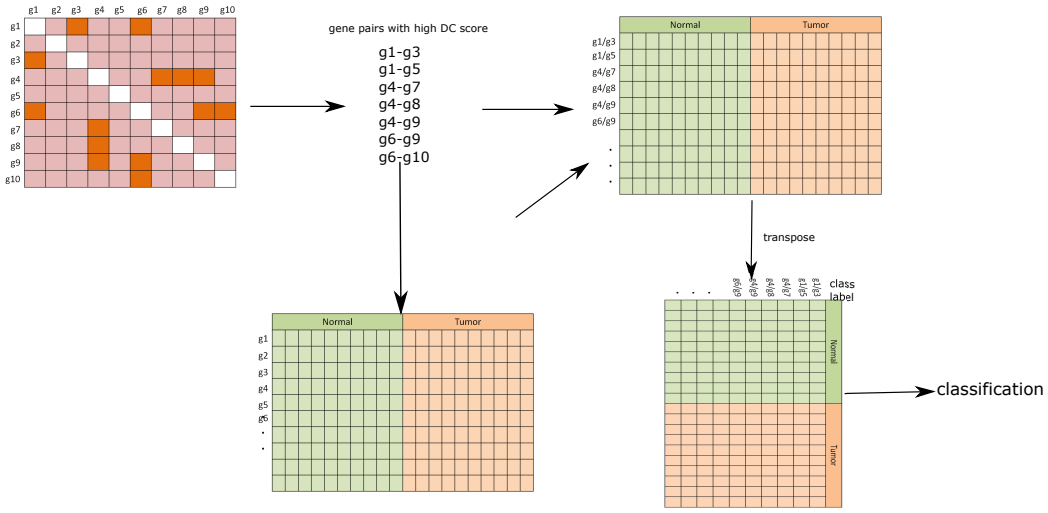


Figure 4.7: A toy example of performing classification on differentially coexpressed gene pairs. From the DC matrix top gene pairs are selected based on DC_copula score. Expression ratio is computed for each gene pairs for normal and tumor samples. The final matrix is then transposed and subsequently, classification is performed using normal and tumor sample as class label.

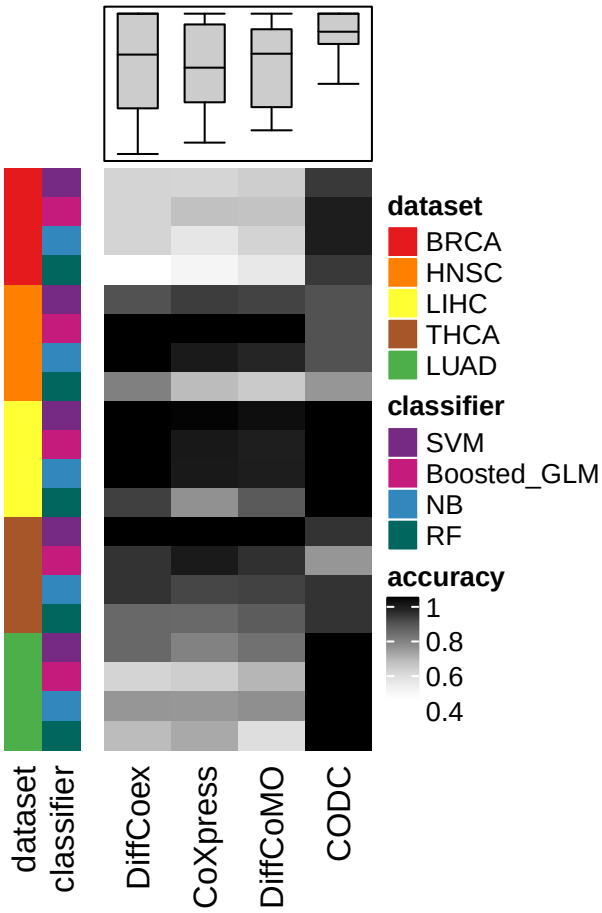


Figure 4.8: Comparison of classification accuracy for five datasets with four classifiers GBM, Naive Bayes, Random Forest and SVM.

4.4.3 Results on Single cell RNA sequence dataset using *sc-CGconv*

Training of graph convolution network on cell-cell graph

To train the GCN model with our datasets we first randomly split the cell-cell graph into an 8:1:1 ratio of the train, validation, and test sets. The test edges are not included in the training set, however, we keep all the nodes of the graph in the training set. Now, we train the model using the training edges and check the performance of the trained model for recovering the removed test edges. The model is trained with 50 epochs using Adam optimizer with a learning rate of 0.001 and a dropout rate of 0.1. ReLu is used as an activation function. Table 4.4 shows the average precision and ROC score for the four networks obtained from the datasets. We took the low dimensional embeddings from the output of the

Table 4.4: Performance of GCN on networks created from four datasets: First two columns of the table shows total number of nodes and number of edges of the four networks. Rest of the columns show ROC and average precision score for validation and test edges. V. ROC and V. AP refer to validation ROC and validation average precision score, whereas T. ROC and T. AP refer the same for test set.

Dataset	#edges	#nodes	V. ROC	V. AP	T. ROC	T. AP
Baron	41876	8569	87.32	87.08	85.87	86.39
Klein	13885	2717	84.79	83.21	83.46	82.81
Melanoma	340875	68579	83.38	86.48	83.1	82.30
PBMC68k	342890	68793	84.98	86.78	82.9	83.8

encoder of the trained model

***sc-CGconv* can produces topology-preserving single-cell embedding**

The resulting embedding of *sc-CGconv* can be utilized for manifold learning and graph drawing algorithms such as UMAP and t-SNE. Here we used *sc-CGconv* to generate the single-cell embeddings throughout this paper for clustering. To quantify how similar the topology of low-dimensional embedding within the space Z is to the topology of the high-dimensional space X, we adopted a procedure similar to wolf et al. [173]. Here, we define a classification setup where the ground truth is defined as a kNN graph $G_X = (V, E_X)$ fitted in the high dimensional space X. The edge set E_{FC} which defines all possible edges is the state space of the classification problem. In this setting, the embedding algorithm predicts whether an edge $e \in E_{FC}$ is an element of E_X. We put label '1' for the edge $e \in E_{FC}$ if $e \in E_X$, otherwise put label '0'. For each edge $e \in E_{FC}$, the embedding will put label 1 with the probability q_e and put label 0 with probability $(1 - q_e)$. The cost function to train such a classifier is form as a binary cross-entropy function $H(P, Q)$ or logloss, which is equivalent to the negative log-likelihood of the labels under the model. It is defined as

$$H(P,Q) = \sum_{e \in E_{FC}} \sum_{l \in 0,1} p_e log(q_e) \quad = \sum_{e \in E_{FC}} p_e log(\frac{1}{q_e}) + (1 - p_e) log(\frac{1}{1 - q_e}) \quad (4.6)$$

Now the KL divergence of the predicted distribution Q and the reference distribution P is measured as $KL(P, Q) = H(P, Q) - H(P)$, where $H(P) = -\sum_{e \in E_{FC}} p_e$., which ultimately leads to the equation

$$KL(P,Q) = \sum_{e \in E_{FC}} p_e log(\frac{p_e}{q_e}) + (1 - p_e) log(\frac{1 - p_e}{1 - q_e}). \quad (4.7)$$

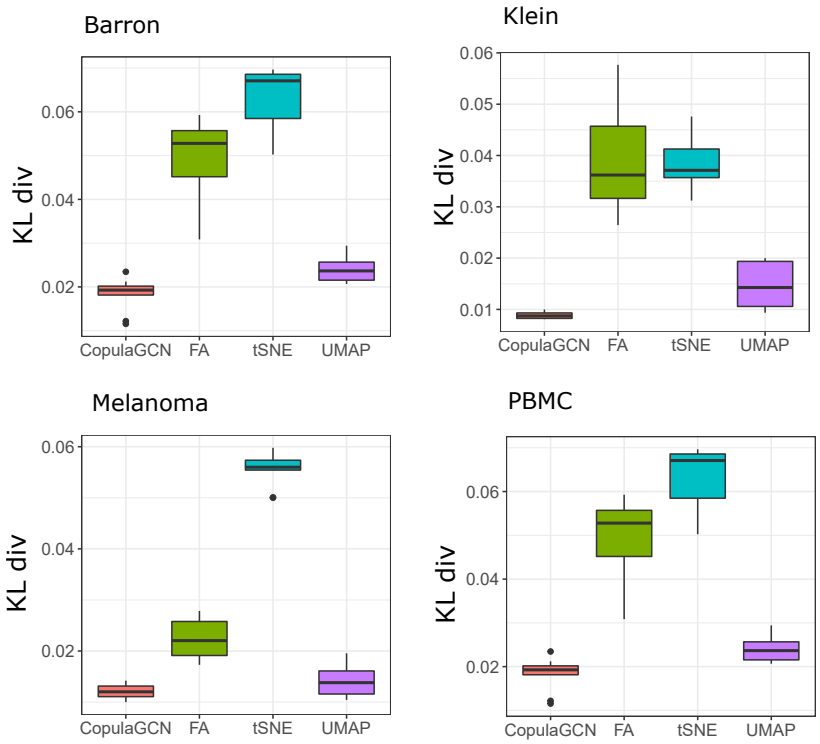


Figure 4.9: Performance of different embedding algorithm on four datasets. Kl divergence is computed by rerunning embedding algorithms 50 times.

We measured the KL divergenece between P and Q for t-SNE [9], UMAP [22], ForceAtlas2 [174] and the *sc-CGconv*. Fig. 4.9 shows the statistics of KL measures for the different embeddings in the four used datasets.

Comparison with state-of-the-arts

In scRNA-seq datasets, single cells are the unit of analysis, and it is crucial to identify the clusters to which they belong accurately. These reference clusters are typically based on the expression profiles of many cells. Misclassification of cells is the common issue for annotating clusters as single-cell gene expression datasets often show a high level of heterogeneity even within a given cluster. To establish the efficacy of *sc-CGconv* over such procedures, we have selected four state-of-the-art methods that are widely used for gene selection and clustering of the single cell data.

Here, we compare *sc-CGconv* with the following five methods I) *Gini Clust* [133]: a feature selection scheme using Gini-index followed by density-based spatial clustering of applications with noise, DBSCAN [175]. II) *GLM-PCA* [64]: a multinomial

model for feature selection and dimensionality reduction using generalized principal component analysis (GLM-PCA) followed by k-means clustering III) Seurat V3/V4 [136]: a single cell clustering pipeline which selects *Highly Variable Gene (HVG)* that exhibit high cell-to-cell variation in the dataset (i.e, they are highly expressed in some cells, and lowly expressed in others) followed by Louvain clustering IV) *Fano Factor* [134], a measure of dispersion among features. Features having the maximum Fano Factor are chosen for clustering. V) scGene-Fit [176], a marker selection method that jointly optimizes cell label recovery using label-aware compressive classification, resulting in a substantially more robust and less redundant set of markers. Vi) SC3, a single-cell consensus (k-means) clustering (SC3) tool for unsupervised clustering of scRNA-seq data.
The R package for Gini-Clust [133] is employed with default settings. For GLM-PCA and HVG, we consider the default settings as described in [64, 136]. For scGene-Fit, the parameters (redundancy=0.25, and method='centres') are used as suggested by their github page. For SC3 we adopted the default parameters for clustering all the datasets.

sc-CGconv requires the number of iterations (*iter*) as the input parameters of LSH step. we set as *iter* as 1 for all datasets. The parameter of Clayton Copula is set as $\theta = -0.5$. For GCN, we used 3-layer GCN architecture which performs three propagation steps during the forward pass and convolves 3rd order neighborhood of every node. We take the dimension of the output layer of the first and second layers as 256 and 128. For decoder, we use a simple inner product decoding scheme.

Execution Time: All experiments were carried out on a Linux server having 50 cores and *X*86_64 platform. As our proposed method is a deep learning based feature extraction method, so it takes more time than any filter-based feature selection technique (e.g. $Fano, Gini - clust$). To check how the competing methods scale with the number of cells (and classes) we performed an analysis. We have generated simulated data (using splatter) by varying the number of cells (and classes). Four simulated data are generated with the number of cells and classes as follows: 500 cells with two classes, 1000 cells with three classes, 1500 cells with four classes, and 2000 cells with five classes. All data are generated with equal group probabilities, 2000 number of features, fixed dropout rate (0.2), and 40% DE gene proportion. 1000 features are selected in each compared method and 128 embedded features are chosen using sc-CGconv . The runtime is compared with all seven different competing methods. The execution time (minute) for each dataset is given in Table 4.5.

To validate the clustering results, we utilized two performance metrics, Adjusted Rand Index (ARI) and Silhouette Score [107].

The Table 4.6 depicts the efficacy of *sc-CGconv* over the other methods. For the other

Table 4.5: Execution time in minute for eight competing methods.

Datasets	# Cells	# Class	Execution Time (in Minute)							
			sc-CGconv	Gini Clust	GLM-PCA	Fano	Seurat	scGeneFit	SC3	M3drop
Data1	500	2	9	2	1	1	3	3	5	4
Data2	1000	3	13	2	1	1	7	5	8	6
Data3	1500	4	17	3	1	3	11	10	13	12
Data4	2000	5	20	5	3	5	14	13	17	15

Table 4.6: Comparison with state-of-the-arts: Adjusted Rand Index (ARI) and Average Silhouette Width (ASW) are reported for six competing methods on four datasets.

Dataset	Method													
	sc-CGconv		Gini Clust		GLM-PCA+Kmeans		Fano+Louvain		Seurat		scGeneFit+Kmeans		SC3	
	ARI	ASW	ARI	ASW	ARI	ASW	ARI	ASW	ARI	ASW	ARI	ASW	ARI	ASW
Baron	0.68	0.52	0.6	0.48	0.42	0.4	0.52	0.46	0.62	0.47	0.62	0.43	0.60	0.4
Melanoma	0.56	0.52	0.15	0.29	0.18	0.24	0.42	0.29	0.43	0.45	0.25	0.4	0.38	0.35
Klein	0.86	0.8	0.76	0.7	0.43	0.58	0.4	0.3	0.8	0.72	0.82	0.75	0.80	0.66
PBMC	0.51	0.46	0.38	0.29	0.31	0.26	0.29	0.14	0.50	0.3	0.47	0.48	0.48	0.31

competing methods, we select the top 1000 features/genes in the gene selection step and use the default clustering technique meant for this method. For *sc-CGconv*, after obtaining the low dimensional embedding of 128-dimension, we performed a simple k-means for clustering the cells. It is evident from the table that *sc-CGconv* with k-means provides higher ARI (and average silhouette width) values for all four datasets.

sc-CGconv preserves cell-to-cell variability

Once features are estimated to be important, it is essential to ask whether the cell-to-cell variability has been preserved within the extracted features. To determine this, we computed the Euclidean distance between each pair of cells. Thus two Euclidean distance matrices are obtained, one for the original feature space, and the other for the reduced feature space. The Correlation score (Kendall tau) is computed between these two distance matrices. Fig. 4.10 depicts the correlation measures for all the four scRNA-seq datasets.

sc-CGconv can identify marker genes

We followed the conventional procedure of Scanpy to find markers (DE genes) from the clustering results. Scanpy utilized Wilcoxon rank-sum test [177] to find out the significant ($p < 0.05$) DE genes for each cluster which are treated as marker genes. We took the top 50 marker genes with their p-value threshold 0.05 on PBMC 68*k* dataset.

We found that 19 marker genes from the melanoma dataset and 13 marker genes from the PBMC dataset that are biologically significant according to cell marker

Table 4.7: Marker genes identified from the clustering results with sc-CGconv. 19 (for Melanoma) and 12 (for PBMC68k) markers are found to be overlapped with CellMarker database

Dataset	cell type	markers (pubmed id)
Melanoma	CD8 T cell	TNFRSF9 (28622514),KLRC4 (28622514), CXCL13 (28622514)
	CD4 T cell	CTSW (28457750), CD69 (28566371), LTB (28263960), CD4 (12000723)
	Regulatory T cell	IL32 (30093597), LAG3 (28929191), FCRL3 (25762785), TNFRSF18 (23929911), LAT2 (28622514)
	Naive T cells	CD7 (7539656)
	T helper1 (Th1) cell	IFNG (20868565), STAT4
	CD4+ memory T cells	CCR7 (28929596)
	NK cell	BCL11B
	B cell	CD79B (29230012)
	Megakaryocyte	CTSW (30093597)
PBMC	Regulatory T cell	IL32 (30093597)
	CD8 T cell	CCL5 (30093597)
	NK cells	NKG7 (8458737), GNLY (12884856)
	Effector CD8+ memory T cell	GZMH (28622514)
	Plasmacytoid dendritic cells	GZMB (19965634)
	CD4+ cytotoxic T cell	CST7 (28622514)
	B cell	CD79A (11396639), CD37 (24952935)
	Monocyte derived dendritic cells	CST3 (19956698)
	Megakaryocyte progenitors	PPBP (27084257), PF4 (30645026)

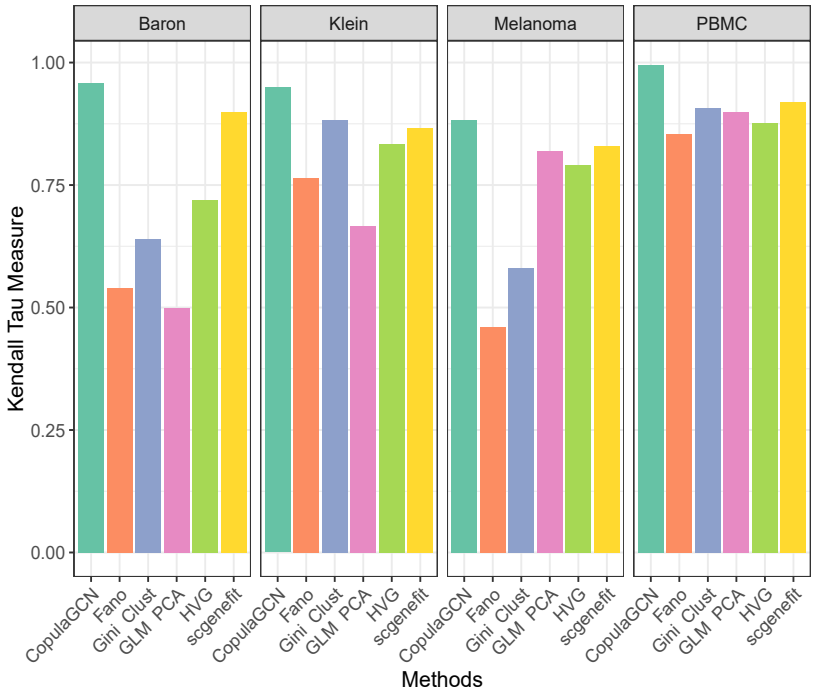


Figure 4.10: Correlation score between two distance matrices, defined on original feature space and reduced feature space. Figure shows the comparisons among the competing methods based on the correlation scores obtained for different set of features.

database [178]. The list of biologically significant marker genes is given in Table 4.7.

4.5 Conclusions

In this chapter, we have described two methods (i) *CODC*, a copula based model to select differential coexpression of genes in TCGA RNA-seq datasets, and (ii) *scCGconv*, a copula based unsupervised feature extraction and clustering technique to select stable and non-redundant features/genes in single cell RNA-seq data.
CODC seeks to identify the dependency between expression patterns of a gene pair in two conditions separately. The Copula is used to model the dependency in the form of two joint distributions. Kolmogorov-Smirnov distance between two joint distributions is treated as a differential coexpression score of a gene pair. We have compared *CODC* with three competing methods DiffCoex, CoXpress, and DiffCoMO in five pan-cancer RNA-Seq data of TCGA. *CODC* has the ability for delineating minor changes of coexpression in two different samples making it unique and suitable for differential coexpression analysis. The scale-invariant

property of copula is inherited into *CODC* to make it robust against noisy expression data. It is advantageous for detecting the minor change in correlation across two different conditions which is the most desirable feature of any differential coexpression analysis.

Next, we have developed a step-wise sampling based feature extraction method for scRNA-seq data leveraging the Copula dependency measure with a graph convolution network. On one hand, LSH based sampling is used to deal with ultra-large sample sizes, whereas Copula dependency is utilized to model the interdependence between features (genes) to construct the cell-cell graph. Graph convolution network has been utilized to learn low dimensional embedding of the constructed graph. There are two striking characteristics of the proposed method : I) It can sample a subset of features from original data keeping the structure intact. Therefore, minor clusters are not ignored. This sampling is achieved by using the LSH based sampling method. II) *sc-CGconv* utilizes scale invariant dependency measure which gives a superior and stable measure for constructing the dependency graph among the cells. III) GCN provides topology-preserving low dimensional embedding of the cell graphs. It can effectively capture higher-order relations among cells. IV) LSH based structure aware sampling of features showed a significant lift in accuracy (Correlation, ARI values) in large single cell RNA-seq datasets.

Another important observation is that *sc-CGconv* yields the highest ARI values for Human Klein and Pollen in comparison to other state-of-art methods. The rationale behind this is that *sc-CGconv* utilized copula correlation measure, which correctly models the correlations among the feature set. From the holistic viewpoint, the *sc-CGconv* algorithm performs much better than the other methods.

The computation time of *sc-CGconv* is equivalent to the number of sampled features. The process may be computationally expensive when a large number of features are selected in the LSH step. However, as copula correlation returns stable and non-redundant features, in reality, a small set of selected features will be effective to construct the cell-cell graph. We observed in scRNA-seq data 1000 sampled features will serve the purpose.

Taken together, the two proposed methods *CODC* and *sc-CGconv* not only outperform in the domain of gene selection/extraction in RNA-seq/scRNA-seq datasets but also can able to identify good modules (clusters) for large bulk/single cell data. It can be observed from the results that *sc-CGconv* leads both in the domain of single cell clustering by extracting informative features and generating low dimensional embedding of cells from large scRNA-seq data. The results prove that both the methods may be treated as an important tool for computational biologists to investigate the important genes from bulk/single cell RNA-seq data.

In the next chapter, we introduce two entropies (*Renyi* and *Tsallis*), which remain unexplored in the feature selection domain, so far. We utilized these to build a model for feature selection from high dimensional single cell data.

5

Entropy based feature selection for high dimensional single cell RNA sequence data.

5.1 Introduction

In recent times technological advances have made it possible to study RNA-seq data at single cell resolution [179]. Single cell RNA sequencing (scRNA-seq) is a powerful tool to capture gene expression snapshots in individual cells. Cell type detection is one of the fundamental steps in downstream analysis of scRNA-seq data [67]. A widely used approach for this is to cluster the cells into different groups, and determine the identity of cells within the individual groups/clusters [131, 66]. This provides an unsupervised method of grouping similar cells into clusters that facilitate the annotation of different cell types present in the large population of scRNA-seq data [161, 162, 163]. Starting from raw counts, scRNA-seq data analysis typically goes through the following steps before clustering: i) normalization, ii) feature selection, and iii) dimensionality reduction. While normalization/log-normalization adjusts the differences between the samples of individual cells and reduces the skewness of the data, feature selection seeks to identify the most relevant features (genes) from the large feature space.

Although there exist a plethora of methods [61, 62, 63, 58] for performing each task within the pipeline, the standard approaches consider a common sequence of steps for the preprocessing of scRNA-seq data [64]. This includes normalization by scaling of sample-specific size factors, log transformation, and feature selection by using the coefficient of variation (highly variable genes[65, 66]) or by using average expression level (highly expressed genes). Alternatively, some methods exist for gene selection, such as GLM-PCA [64] selects features (genes) by ranking genes using deviance, M3Drop [63] selects genes leveraging the dropouts effects in the scRNA-seq data.

The performance of downstream analysis, mainly the clustering process, is heavily

dependent on the quality of the selected top features/genes. The typical characteristics of good features/genes are: i) it should encode useful information about the biology of the system, ii) should not include features that contain random noise, and iii) preserve the useful biological structure while reducing the size of the data to reduce the computational cost of later steps.
In this chapter, we show that employing *Renyi*, and *Tsallis* entropies in the gene filtering process introduces major advantages both in terms of clustering performance, and in terms of a biologically meaningful interpretation (a.k.a. marker selection) of the results. *Renyi*, and *Tsallis* entropies have major advantages over the Shannon method for robustness against noisy observations/data. An appropriate choice of the tuning parameter q, in either of the two entropies, makes them less sensitive (more robust) against different noises present in the data. On the other hand, Shannon entropy and the associated Kullback-Leibler divergence directly lead to the likelihood based inference which is known to be extremely non-robust against data contamination. Therefore, the use of these *Renyi*, and *Tsallis* entropies strengthens the robustness of our objective function proposed for feature selection (see Section 5.2). As a result, our proposed procedure leads to the robust selection of truly important features, suppressing the ill-effects of different kinds of noises present in single-cell data.
We have compared the proposed method with five state-of-the-art gene selection techniques: entropy based method: Shannon [180] entropy; and four methods are well known for gene selection in scRNA-seq data (*Gini Clust* [133], *GLM-PCA* [64], *M3Drop* [63] and *HVG* of Seurat V3 [136]). We have also utilized three well known scRNA-seq clustering method (Seurat [66], SC3 [131] and CIDR [181]) to validate the features (genes) selected by different competing methods.
Beyond the selection of good informative features, we also demonstrate that by using a simple classification model the selected genes can correctly classify cells of completely independent test data. To check this, we split the data into training and test sets and demonstrate that the selected features in the training set are equally effective for the classification of the test samples. We also carry out a comprehensive simulation study to establish the effectiveness of the proposed method on simulated scRNA-seq data with four separate settings. The results show that the proposed method not only selects genes with high accuracy but is also robust when the parameters are appropriately tuned.

5.2 Methods

5.2.1 Deriving *Renyi* and *Tsallis* Risk Functions

Let us consider three discrete random variables X, Y and Z with supports $\{x_1, \cdots, x_d\}$, $\{y_1, \cdots, y_p\}$ and $\{z_1, \ldots, z_n\}$, respectively. For each $i = 1, 2, \ldots, d$, $j = 1, 2, \ldots, p$ and

$k = 1, 2, \ldots, n$, let us denote $p_i = P(X = x_i)$, $p_{ijk} = P(X = x_i, Y = y_j, Z = z_k)$, $p_{i|jk} = P(X = x_i | Y = y_j, Z = z_k)$ and so on. The *Renyi* entropy of the random variable X is defined in terms of a positive real number q, with $q \neq 1$, as

$$H_q(X) = \frac{q}{1-q} \log \|P_X\|_q = \frac{q}{1-q} \log \left(\sum_i p_i^q \right)^{1/q}, \quad q \neq 1, \tag{5.1}$$

where, $\|P_X\|_q$ is q-norm of the discrete probability distribution $P_X = (p_1, \ldots, p_d)$ of X, interpreted as a vector in $\mathbb{R}^d$. Interestingly, note that, this *Renyi* entropy reduces to the Shannon entropy when $q \to 1$. It can also be extended for the three random variables X, Y, and Z, so that their joint *Renyi* entropy is given by

$$H_q(X, Y, Z) = \frac{q}{1-q} \log \left(\sum_{i,j,k} p_{ijk}^q \right)^{1/q}, \quad q \neq 1. \tag{5.2}$$

Accordingly, the conditional *Renyi* entropy can be defined as

$$H_q(X|Y, Z) = \frac{q}{1-q} \log \left(\sum_{j,k} \left(\sum_i p_{i,j,k}^q \right)^{1/q} \right), \quad q \neq 1 \tag{5.3}$$

It is important to note here that there are several other existing definitions of Conditional *Renyi* entropy (see, e.g., [182, 183, 184]). Our particular form of Conditional *Renyi* divergence in (3) was originally proposed by [185] in the context of channel coding in information theory; more recent works (see, e.g., [186]) have further justified the usefulness of this definition in comparison to the other definitions. Note that, our conditional *Renyi* entropy is a decreasing function of q. It can also be shown that the conditional *Renyi* entropy closely corresponds to a risk function, i.e. as an expected error when we try to estimate the value of X, given the values of Y and Z. We refer to the associated risk as the *Renyi Risk Function* which is defined as

$$\begin{aligned} R_q(X|Y, Z) =& \|P_X\|_q - \sum_{j,k} p_{j,k} \left(\sum_i p_{i|j,k}^q \right)^{1/q}, \\ =& \sum_{j,k} p_{j,k} \left(\|P_X\|_q - \|P_{X|y_j,z_k}\|_q \right) \\ =& E_{Y,Z} \left[L_q \left(X; (Y, Z) \right) \right], \end{aligned} \tag{5.4}$$

where $P_{X|y_j,z_k} = \left(p_{1|j,k}, \ldots, p_{d|j,k} \right)$ and $L_q \left(X; (y, z) \right) = \|P_X\|_q - \|P_{X|y,z}\|_q$ is a loss function measuring the discrimination between unconditional distribution of X and its

conditional distribution given $(Y = y, Z = z)$. This formulation clearly proves that the quantity $R_q(X|Y,Z)$, being the expected loss, is indeed a risk function.
Next, the *Tsallis* joint entropy of the three random variables X, Y and Z is mathematically defined, also in terms of a tuning parameter $q > 0$, as

$$H_{T_q}(X,Y,Z) = \frac{1}{q-1}\left[1 - \sum_{i,j,k} p_{i,j,k}^q\right], \tag{5.5}$$

It also coincides with the Shannon entropy as $q \to 1$. The conditional *Tsallis* entropy of random variable X, given the values of Y and Z is defined as

$$H_{T_q}(X|Y,Z) = \frac{1}{q-1}\left[1 - \left(\frac{\sum_{i,j,k} p_{i,j,k}^q}{\sum_{j,k} p_{j,k}^q}\right)\right], \quad q \neq 1. \tag{5.6}$$

So, we can again define a *Tsallis* risk function, i.e. another form of the expected error when we try to estimate the value of X given the values of Y and Z, as

$$\begin{aligned} R_{T_q}(X|Y,Z) &= \|P_X\|_q - \sum_{j,k} {}_e p_{j,k}^q \left(\sum_i p_{i|j,k}^q\right), \\ &= E_{Y,Z}^q\left[L_q\left(X;(Y,Z)\right)\right], \end{aligned} \tag{5.7}$$

where ${}_e p_{j,k}$ is the joint escort probability distribution [187] of (Y,Z), given by ${}_e p_{j,k} = \frac{p_{j,k}^q}{\sum_{j,k} p_{j,k}^q}$, and $E_{Y,Z}^q$ represents the expectation with respect to this escort distribution.
Note that, the loss function L_q in the *Tsallis* risk (Equation 5.7) is the same as considered in the *Renyi* risk (Equation 5.4), but we have now taken the expectation with respect to joint q-escort distribution of (Y,Z) instead of the same with respect to the usual joint distribution of (Y,Z). Such expectation with respect to the escort distribution is quite common in the literature of non-extensive entropies including the *Tsallis* one [188, 189, 42]. The details of entropies is given in Section 1.2.3 of Chapter 1.

5.2.2 *sc-REnF* Algorithm for Feature (gene) Selection

Let, any dataset be arranged in a matrix $D_{m\times n}$, where n is number of samples and d is number of features. Let, F be the set of features, $F = \{f_1, f_2, f_3, \ldots, f_n\}$, and Y be the set of target variable. Our algorithm is wrapper based forward selection approach which constructs a monotonically increasing sequence $\{S\}$ of subset of F. At step i, subset $\{f_i\}$ is selected according to the objective function and added to the previous

feature subset S_{i-1}. The objective function is defined with a feature relevancy and redundancy criteria driven by entropy measure $\mathcal{E}$.(we utilized *Renyi*, and *Tsallis* entropies as the specific choices for $\mathcal{E}$.)

Feature relevance

Feature f_i is more relevant to the class label C than feature f_j in the context of the already selected subset S, when

$$\mathcal{E}(C|f_i) \geq \mathcal{E}(C|f_j), \tag{5.8}$$

where $\mathcal{E}(\cdot|\cdot)$ is a (bivariate) conditional entropy function.

Feature redundance

Selected feature f_i is more redundant to the feature f_j than feature f_k, if the following holds:

$$\mathcal{E}(C|f_i, f_j) \geq \mathcal{E}(C|f_i, f_k), \quad f_j, f_k \in (F - S), \quad f_i \in S \tag{5.9}$$

where $\mathcal{E}(\cdot|\cdot, \cdot)$ is an appropriate conditional entropy.

The objective Function

We minimize the conditional entropy function between $f_i \in (F - S)$ and $f_s \in S$ (to reduce the redundancy between them) and maximize the conditional entropy function between class label C and $f_i \in (F - S)$ to select the first feature, where $f_s \in S$ is already selected feature. The selected feature subset, $\{S\}$ and the feature $f_i \in (F - S)$ are inductively defined as

$$\begin{aligned} S =& \emptyset \\ f_1 =& arg\, max_{(f_i \in F)} \mathcal{E}(C|f_i), \quad \text{see Equation 5.8} \\ S =& S \cup \{f_1\}, \\ f_{i+1} =& arg\, min_{(f_i \in (F-S), f_j \in S)} \mathcal{E}(C|f_i, f_j), \quad \text{(see Equation 5.9)} \\ S =& S \cup \{f_{i+1}\}, \end{aligned} \tag{5.10}$$

sc-REnF utilizes a wrapper based stepwise forward selection approach to select gene iteratively from a gene set. An overview of the *sc-REnF* Algorithm is given in algorithm **4**.
The selected features using the objective function (with *Renyi* and *Tsallis* entropies as a choice of $\mathcal{E}$) are optimal in the sense of minimizing the corresponding risk functions, defined in Equations 5.4 and 5.7, respectively, as stated by the proposition:

Theorem: At every step, the selected feature f_{i+1} minimizes the *Renyi* and *Tsallis* risk functions, i.e.,

$$\mathcal{R}(C|f_s, f_{i+1}) \leq \mathcal{R}(C|f_s, f), \forall f \in (F - S), \quad f_s \in S, \tag{5.11}$$

where $\mathcal{R}$ denotes either R_q or R_{T_q}, defined in Equations 5.4 and 5.7, respectively.

Proof. In order to proof the theorem, We will start from the objective function in Equation 5.10 .
According to our objective function

$$\mathcal{E}(C|f_s, f_{i+1}) \leq \mathcal{E}(C|f_s, f), \forall f \in (F - S), \quad f_s \in S. \tag{5.12}$$

Let, u, v', v represent generic value tuples and values of $f_s \in S$ (Selected feature), f_{i+1} (To be selected feature at $(i+1)^{th}$ step), and $f \in (F - S)$ (Non selected features) respectively. After putting the generic representation, Equation 5.12 for the *Renyi* entropy can be written as

$$\mathcal{E}(C|u, v') \leq \mathcal{E}(C|u, v)$$

$$\implies \frac{q}{1-q} \log \sum_{u,v'} \left(\sum_c p^q_{c,u,v'} \right)^{1/q}$$

$$\leq \frac{q}{1-q} \log \sum_{u,v} \left(\sum_c p^q_{c,u,v} \right)^{1/q}, \quad \text{(see Equation 5.3.)}$$

$$\implies \text{If q} > 1, \text{it leads to}$$

$$\sum_{u,v'} \left(\sum_c p^q_{c,u,v'} \right)^{1/q} \geq \sum_{u,v} \left(\sum_c p^q_{c,u,v} \right)^{1/q} \implies \sum_{u,v'} p_{u,v'} \left(\sum_c p^q_{c|u,v'} \right)^{1/q} \geq \sum_{u,v} p_{u,v} \left(\sum_c p^q_{c|u,v} \right)^{1/q}. \tag{5.13}$$

On the other hand, for $q < 1$, we get

$$\sum_{u,v'} \left(\sum_c p^q_{c,u,v'} \right)^{1/q} \leq \sum_{u,v} \left(\sum_c p^q_{c,u,v} \right)^{1/q} \implies \sum_{u,v'} p_{u,v'} \left(\sum_c p^q_{c|u,v'} \right)^{1/q} \leq \sum_{u,v} p_{u,v} \left(\sum_c p^q_{c|u,v} \right)^{1/q}. \tag{5.14}$$

Now, multiplying a constant $k = \frac{q}{1-q}$ in Equations 5.13 and 5.14 we get, for any $q \neq 1$, that

$$k \sum_{u,v'} p_{u,v'} \left(\sum_c p^q_{c|u,v'} \right)^{1/q} \geq k \sum_{u,v} p_{u,v} \left(\sum_c p^q_{c|u,v} \right)^{1/q}. \tag{5.15}$$

Then, by definition of *Renyi* risk function in Equation 5.4, we get

$$R_q(C|f_s, f_{i+1}) \leq R_q(C|f_s, f). \tag{5.16}$$

In case of minimization of *Tsallis* risk function, after putting the generic representation, the Equation 5.12 for the *Tsallis* entropy can be written as

$$\begin{aligned}
&\mathcal{E}(C|u, v') \leq \mathcal{E}(C|u, v) \\
&\implies \frac{1}{q-1}\left[1 - \left(\sum_{c,u,v'} p^q_{c,u,v'} / \sum_{u,v'} p^q_{u,v'}\right)\right] \leq \frac{1}{q-1}\left[1 - \left(\sum_{c,u,v} p^q_{c,u,v} / \sum_{u,v} p^q_{u,v}\right)\right], (\\
&\qquad \text{see Equation 5.6.)}
\end{aligned} \tag{5.17}$$

Now, for $q > 1$, we have

$$\begin{aligned}
-\left(\sum_{c,u,v'} p^q_{c,u,v'} / \sum_{u,v'} p^q_{u,v'}\right) &\leq -\left(\sum_{c,u,v} p^q_{c,u,v} / \sum_{u,v} p^q_{u,v}\right) \\
\implies \left(\sum_{c,u,v'} p^q_{c,u,v'} / \sum_{u,v'} p^q_{u,v'}\right) &\geq \left(\sum_{c,u,v} p^q_{c,u,v} / \sum_{u,v} p^q_{u,v}\right), \\
\implies \left(\sum_{u,v'} p^q_{u,v'} \sum_c p^q_{c|u,v'}\right) / \sum_{u,v'} p^q_{u,v'} &\geq \left(\sum_{u,v} p^q_{u,v} \sum_c p^q_{c|u,v}\right) / \sum_{u,v} p^q_{u,v}, \\
\implies \left(\sum_{u,v'} {}_e p^q_{j,k} \sum_c p^q_{c|u,v'}\right) &\geq \left(\sum_{u,v'} {}_e p^q_{j,k} \sum_c p^q_{c|u,v}\right).
\end{aligned} \tag{5.18}$$

Also, for $q < 1$, Equation 5.17 can be written as

$$\left(\sum_{u,v'} {}_e p^q_{j,k} \sum_c p^q_{c|u,v'}\right) \leq \left(\sum_{u,v'} {}_e p^q_{j,k} \sum_c p^q_{c|u,v}\right). \tag{5.19}$$

Now, Multiplying a constant $k = \frac{1}{q-1}$ in Equation 5.19, we get

$$k\left(\sum_{u,v'} {}_e p^q_{j,k} \sum_c p^q_{c|u,v'}\right) \geq k\left(\sum_{u,v'} {}_e p^q_{j,k} \sum_c p^q_{c|u,v}\right). \tag{5.20}$$

Then, by definition of *Tsallis* risk function, described in Equation 5.7, we have

$$R_{T_q}(C|f_s, f_{i+1}) \leq R_{T_q}(C|f_s, f). \tag{5.21}$$

This completes the proof. □ □

Algorithm 4 **R**obust **En**tropy based **F**eature (gene) selection method (*sc-REnF*)

Input: Preprocessed data matrix **D**, Cell type **C**, Number of selected features **d**.
Output: Optimal feature subset (**S**).

Initialisation:
$\mathbf{S} = \emptyset$,
$\mathbf{f_{1s}} \leftarrow \arg\max_{f_i} \mathcal{E}(f_i|C)$, {Maximum Relevancy}
for all $i = 0$ to $(\mathbf{d} - \mathbf{1})$ **do**
  $\mathbf{E} = \emptyset$
  $\mathbf{E} \leftarrow arg\, min_{(f_i \in (F-S), f_j \in S)} \mathcal{E}(C|f_i, f_j)$
  $\mathbf{S} \leftarrow \{\mathbf{S} \bigcup \arg\max(\lim_{f_i}\{E\})$
  $\mathbf{F} \leftarrow F - \{f_i\}$
end for
return **S**

5.3 Results and Discussion

In the following, we will describe the workflow of our analysis pipeline.

5.3.1 Workflow of *sc-REnF*

The Fig. 5.1 describes the workflow of our analysis pipeline. All important steps are discussed in this subsection.

Preprocessing of raw datasets: Single-cell RNA sequence raw datasets are downloaded from publicly available sources. The RNA counts are organised as a matrix $D_{C \times G}$, where C is the number of cells and G is the number of genes. Each element $[D]_{ij}$ represents count of the i^{th} cell and the j^{th} gene. If more than a thousand genes are expressed (non zero values) in one cell, then the cell is termed as good. We assume one gene is expressed if the minimum read count of it exceeds 5 in at least 10% of the good cells. The data matrix M with expressed genes and good cells is normalized using a linear model and normality based normalizing transformation method (Linnorm) [54]. The resulting matrix ($D_{C' \times G'}$) is then log_2 transformed by adding one as a pseudo count.

Feature/gene selection using *Renyi*, and *Tsallis* entropies See panel-A of the Fig. 5.1. The feature/gene selection is driven by an iterative algorithm that select most relevant and non-redundant features using *Renyi*, and *Tsallis* entropies in each step. First, relevancy between all features and class labels is computed to select the most relevant feature (see Equation 5.8). Next, non-redundant features are selected by calculating redundancy between the remaining features and the relevant one (see Equation 5.9).

Validation of selected features/genes See Fig. 5.1, panel-B. The validation is performed by employing benchmark single cell clustering techniques to group

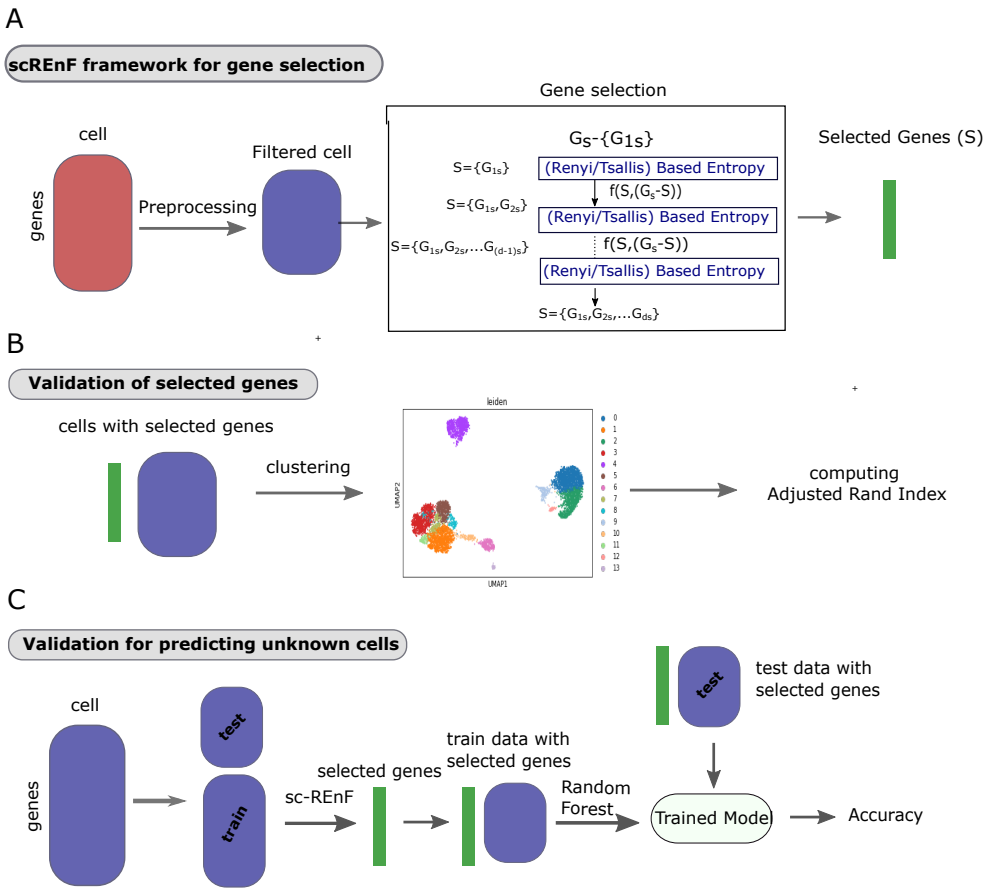


Figure 5.1: A brief framework of our study: Panel-A: scRNA-seq count matrix are downloaded and preprocessed. *sc-REnF* is applied for gene/feature selection using *Renyi*, *Tsallis* entropy measures. Panel-B: Selected genes are validated by adopting scRNA-seq clustering techniques. Panel-C: validating the selected genes (from a training set) in an unknown test samples using clustering and ARI method.

the cells with selected features/genes. The clustering results are evaluated using Adjusted Rand Index (ARI) score which ensures proper and accurate partitioning of cells. We compute differentially expressed (DE) genes from each cluster which can be treated as marker genes of specific groups of cells

Classification of cell samples with unknown labels See Fig. 5.1, panel-C. Prediction of unknown cells in scRNA-seq data analysis is crucial and can be addressed by supervised or unsupervised way. For unknown cell types, the selected genes obtained from our method can able to discriminate the unknown cells in the test data. We split the whole expression data in train:test ratio 8:2, and select features/genes from training set using *sc-REnF*. Training data with the selected genes are used to train a classifier model, which is further used to predict cell types of

Table 5.1: Four setups for generating simulated datasets using Splatter [1]

Setups	Group Proportions (%)	Dropout rate	DE Gene Proportion (%)
S1	(10, 10, 10, 70)	0.2	40
S2	(25, 25, 25, 25)	0.5	40
S3	(10, 10, 10, 70)	0.5	10
S4	(25, 25, 25, 25)	0.2	10

Table 5.2: Classification Accuracy are reported for different values of $q-parameter$ for *Renyi*, and *Tsallis* entropies

Method	Group Prob	Classifier	DE(%)	q-Values									
				0.1	0.3	0.5	0.7	1.3	1.5	1.7	2	2.5	3
sc-REnF_Renyi	0.7:0.1:0.1:0.1	Random Forest	40	0.85	0.89	**0.94**	**0.96**	0.95	0.92	0.91	0.88	0.83	0.82
			10	0.7	0.75	0.85	**0.93**	**0.92**	0.91	0.9	0.85	0.84	0.88
	0.25:0.25:0.25:0.25		40	0.92	0.94	0.94	**0.95**	**0.95**	**0.96**	0.94	0.92	0.9	0.89
			10	0.82	0.88	**0.87**	**0.88**	**0.88**	0.87	0.86	0.86	0.84	0.81
sc-REnF_Tsallis	0.7:0.1:0.1:0.1		40	0.88	**0.89**	**0.9**	**0.9**	0.89	0.87	0.86	0.88	0.87	0.85
			10	0.72	**0.75**	**0.76**	**0.76**	0.74	0.74	0.73	0.7	0.68	0.68
	0.25:0.25:0.25:0.25		40	0.9	**0.91**	**0.9**	**0.9**	0.89	0.88	0.87	0.87	0.86	0.87
			10	**0.85**	**0.84**	**0.85**	**0.85**	0.86	0.83	0.83	0.82	0.8	0.79

the test cell samples.

5.3.2 Feature Selection on Synthetic scRNA-seq Data

Data generation

We generated simulated data to evaluate the tuning parameter ($q-$value) of *sc-REnF*. Splatter [1] is utilized to generate the data with four experimental setups:
S1: generated 500 cells in four groups, with sample a ratio of 10 : 10 : 10 : 70, keeping low dropout rate ($\sim$ 0.2) over 2000 genes
S2: generated four equal-sized groups of cells, each group consisting 25% of the total (500) cells, over 2000 genes at a high dropout rate ($\sim$ 0.5)
S3: generated four groups of 500 cells, with the sample ratio 10 : 10 : 10 : 70 over 2000 genes with a high dropout rate ($\sim$ 0.5)
S4: generated four equal-sized groups of 500 cells over 2000 genes at a low dropout rate $\sim$ 0.2.
The proportions of differentially expressed (DE) genes in S1 to S4 were varied from 40%, 40%, 10%, 10% respectively. The details of the simulation settings are shown in Table 5.1.

Feature selection using different range of q parameter

To investigate the tuning parameter (q-parameter of *Renyi*, and *Tsallis* entropies, see Method section for explanation) of the proposed method, we trained random forests classifier to measure overall accuracy over 100 simulation replicates. Furthermore, we reported the median of accuracy in Table 5.2 for each of the twelve

q-values in four simulation conditions for the proposed method(*sc-REnF_Renyi*, and *sc-REnF_Tsallis*).
High accuracy is observed for the q-parameter ranging from 0.5 to 1.3 for the *Renyi* entropy, and from 0.3 to 0.7 for the *Tsallis* entropy (see Table 5.2). The selected range of q-values are utilized for the later stage of analysis.

5.3.3 Comparisons with State-of-the-art

Here, we compared *sc-REnF* with four state-of-the-art gene selection algorithms widely used in scRNA-seq analysis: *Gini Clust* [133], *M3Drop* [63], *GLM-PCA* [64], and highly variable gene selection *HVG* of Seurat V3/V4 [136]. To know how the performance of the proposed method is improved over the other entropy measure we also include Shannon entropy as a competing method. For Gini-Clust, we use the R package with the default parameter as provided in the original paper. For GLM-PCA, we consider the first three PC components as the default parameter. For highly variable genes (HVG), the standard pipeline of the Seurat package is adopted here.
Three benchmark scRNA-seq clustering algorithm (Louvain clustering of Seurat [66] pipeline, SC3 [131] and CIDR [181]) are employed to validate the selected features obtained from different state-of-the-arts gene selection techniques. We have retained the default parameters as specified in each of the scRNA-seq clustering packages.
Top 500 genes are selected for each competing method. The ARI score for each clustering result is reported in Table 5.3. It is observed that *sc-REnF* (using *Renyi* and *Tsallis* entropies) responds well compare to the other five competing methods.

5.3.4 Classifying Test Samples using Selected Features

Classifying a new cell samples is crucial for the scRNA-seq data analysis pipeline. Here, we address this by performing an analysis to show how the selected genes are important for discriminating the unknown cell samples. We first split the data in train-test ratio of 8:2 and use *sc-REnF* to select 500 most informative genes from the training set. Next, we train a random forest classifier with this data and retain the trained model. Table 5.4 shows the classification performance of the trained model on the test sample using the selected genes as the feature set. The experiment is repeated 100 times with a random split of train-test data with 8:2 ratio in each case. High classification accuracy demonstrates that the selected feature sets are equally important for discriminating the cells of the completely independent test samples.

Table 5.3: Comparisons with five state-of-the-arts feature selection methods on five scRNA-seq datasets: ARI scores are reported for three clustering methods SC3 Seurat and CIDR

Dataset	Clustering Methods	State-of-the-arts						
		***sc-REnF**_Renyi*	***sc-REnF**_Tsallis*	Gini Clust	HVG	M3Drop	GLM-PCA	Shannon
CBMC		0.35	**0.63**	0.2	0.39	0.42	0.22	0.4
Yan		0.68	**0.71**	0.62	0.47	0.67	0.65	0.53
Klein	SC3	**0.86**	0.82	0.76	0.83	0.8	0.82	72
CellBench10x		**0.92**	0.89	0.82	0.9	0.88	0.89	0.78
Melanoma		0.42	**0.51**	0.15	0.38	0.4	0.5	0.24
CBMC		0.29	**0.6**	0.25	0.45	0.36	0.19	0.35
Yan		0.72	**0.74**	0.61	0.55	0.62	0.68	0.69
Klein	Seurat	**0.78**	0.75	0.7	0.86	0.77	0.76	0.68
CellBench10x		**0.88**	0.87	0.81	0.88	0.86	0.86	0.77
Melanoma		0.38	**0.47**	0.14	0.35	0.38	0.45	0.22
CBMC		0.36	**0.72**	0.28	0.36	0.38	0.25	0.36
Yan		**0.71**	0.7	0.67	0.52	0.65	0.67	0.65
Klein	CIDR	**0.87**	0.81	0.72	0.84	0.82	0.84	0.7
CellBench10x		**0.93**	0.9	0.86	0.9	0.9	0.9	0.8
Melanoma		0.45	**0.5**	0.2	0.39	0.42	0.48	0.23

Table 5.4: Classification Accuracy on test datasets using *sc-REnF*.

Datasets	Classifier	Proposed Methods	
		sc-REnF_Renyi	*sc-REnF_Tsallis*
CBMC	Random Forest	0.72 ± 0.02	0.81 ± 0.01
Yan		0.94 ± 0.05	0.92 ± 0.03
Klein		0.98 ± 0.01	0.95 ± 0.02
Cellbench		0.97 ± 0.01	0.92 ± 0.03
Melanoma		0.77 ± 0.05	0.75 ± 0.04

5.3.5 Selected Genes have Reliable Overlap with Marker Genes

Here the aim is to investigate the overlap between the selected gene set and the marker genes of specific cell types. For this, we use the datasets with a full gene set and follow the general framework of Scanpy for clustering. We obtained the marker genes from the identified clusters by utilizing Wilcoxon Rank Sum test embedded in the Scanpy package. It results in highly differential genes (DE) for each cluster. We compare the overlap between selected gene sets of six competing methods (*sc-REnF_Renyi*, *sc-REnF_Tsallis*, Gini Clust, GLM-PCA, M3Drop, and HVG) and the DE genes obtained from Scanpy. Table 5.5 shows the result of overlaps for 500 selected genes in each case. It can be observed from the table that the selected set of genes obtained from *sc-REnF* (using *Renyi*, and *Tsallis* entropies) shows a higher number of overlaps (common gene) with the DE genes than the other competing methods. We have also identified several experimentally verified markers from the common gene set using a literature search. Table 5.6 shows some of the markers of specific cell types obtained from the selected gene set of *sc-REnF*. The PubMed-id for each search is given in the table for reference.

Table 5.5: Table shows overlap between the marker genes and top 500 selected genes using the competing methods

Data	*sc-REnF_Renyi*	*sc-REnF_Tsallis*	Gini Clust	HVG	M3Drop	GLM-PCA
Yan	30	39	5	33	25	4
CBMC	75	160	46	123	89	58
Klein	92	78	40	88	74	75
Cellbench	84	76	65	82	68	73
Melanoma	59	42	29	53	35	40

Table 5.6: Markers identified from the selected gene set of *sc-REnF* for three datasets CBMC, Melanoma, and Cellbench.

Dataset	cell type	markers (pubmed id)
Melanoma	CD8 T cell	TNFRSF9 (28622514),KLRC4 (28622514), CXCL13 (28622514)
	CD4 T cell	CTSW (28457750), CD69 (28566371), LTB (28263960), CD4 (12000723)
	Regulatory T cell	IL32 (30093597), LAG3 (28929191), FCRL3 (25762785), TNFRSF18 (23929911), LAT2 (28622514)
	Naive T cells	CD7 (7539656)
	T helper1 (Th1) cell	IFNG (20868565), STAT4
	CD4+ memory T cells	CCR7 (28929596)
	NK cell	BCL11B
	B cell	CD79B (29230012)
	Megakaryocyte	CTSW (30093597)
CBMC	CD8 T cell	APEX1 (28457750), NASP(28622514), DCAF11 (28622514)
	NK cells	ZNF683 (29361178), GNLY (12884856)
	CD4+ T cell	CST7 (28622514), GSPT1 (28622514), PRF1 (28622514), CCL5 (28622514), CHST12 (28622514), SLC40A1 (28622514)
	B cell	CD79B (28428369), LYL1 (30093597) , NAA50 (30093597) , NXPH4 (30093597), FAM177B (30093597), CD74 (30093597), SRM(30093597)
	CD16+ Mono	S100A12 (28428369)
	MK Cell	PLEK(28622514)
	Eryth	HBG1 (28830992), HBA1 (28830992)
	Plasmacytoid dendritic cell	NOP56 (28428369), IRF8 (28428369), CD83 (30093597)
Cellbench	Epithelial cell	EPCAM (24768153), KRT19 (24972717), FZD4 (24768153)

5.3.6 Stability of *sc-REnF*

Here we explore an additional advantage of *sc-REnF* over the other measures by validating the stability of its performance. A non-parametric statistical test Kruskal-Wallis Test [190] is utilized to examine the stability of ARI scores resulted from the clustering. Here population distributions in each group considered are the same in particular, if the distributions are coming from a common family, then the hypothesis specifies that the median of each group is equal. We vary the number of selected features/genes from range 100 to 500 and for each case, we compute the ARI scores after clustering. Thus for any method (e.g. *Renyi*), we get five ARI scores (for #feature=100, 200, 300, 400, 500) representing the clustering performance with selected features/genes. To know the variation of ARI scores across all the datasets for a particular method (e.g. *Renyi*), we performed Kruskal-Wallis Test. Although all methods produce stable results with low p-values, nevertheless the *sc-REnF* (with *Tsallis* and *Renyi* entropies) show more stable ($p - value$=9.01E-03 and 1.11E-02) performance among the other methods (see Table 5.7). These results may be treated as a straightforward implication of the theoretical proof presented in the method section.

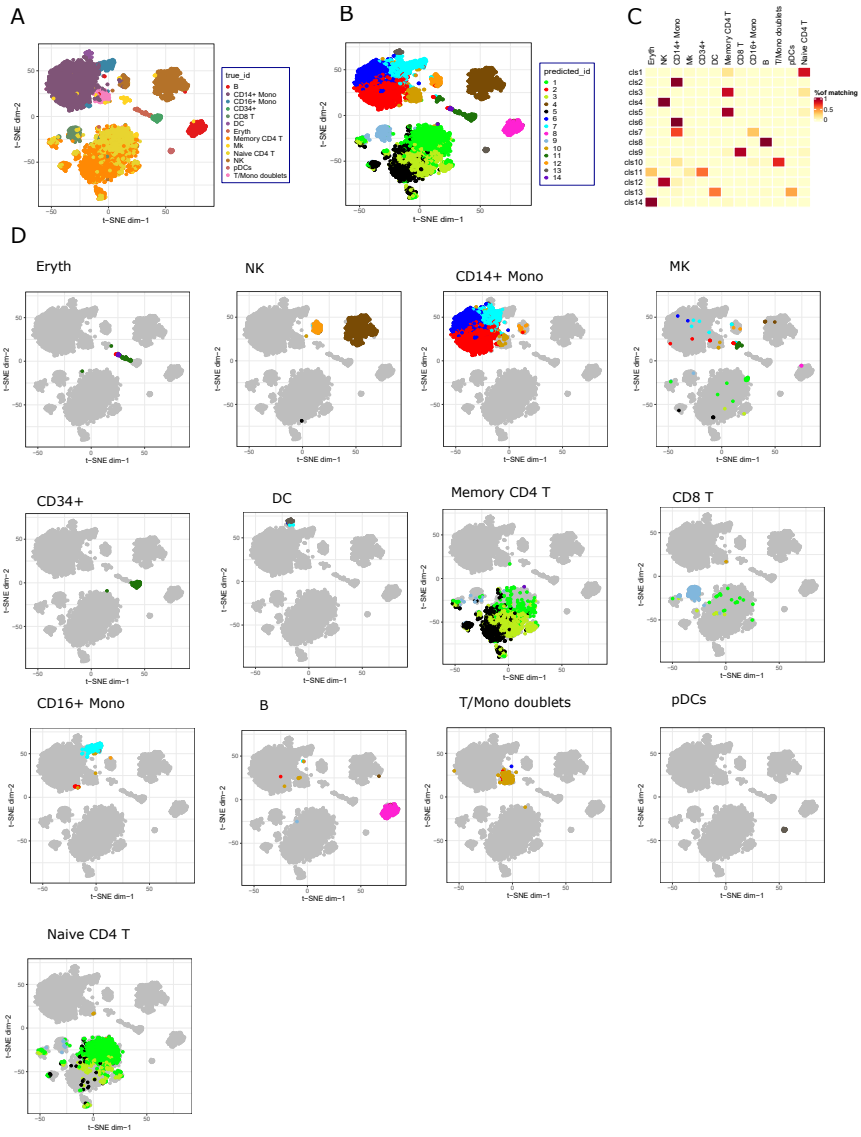


Figure 5.2: Clustering results of CBMC data after gene selection. Panel-A and -B represents t-SNE visualization of data with original and predicted cluster labels respectively. Panel-C shows a heatmap that represents the percentage of matching samples between 14 identified clusters and 13 different cell types. Panel-D depicts a visualization of samples coming from different immune cells and their corresponding predicted clusters (color-coded)

5.3.7 Execution Time

All experiments were carried out on a Linux server having 50 cores and *x*86_64 platform. As our proposed method is a wrapper based step wise feature selection

Table 5.7: Stability performance of *sc-REnF*: p-values (Kruskal-Wallis test) are reported on ARI scores obtained from the clustering results of scRNA-seq data

Sl No.	Method	chi-squared Value	p-Value
1	*sc-REnF_Tsallis* (q-parameter=0.3)	13.5	**9.01E-03**
2	*sc-REnF_Renyi* (q-parameter=0.7)	13.01	**1.11E-02**
3	Gini Clust	12.38	3.02E-02
4	HVG	12.05	5.64E-02
5	M3Drop	12.91	1.51E-02
6	GLM-PCA	12.6	1.80E-02
7	Shannon	13.06	1.21E-02

Table 5.8: Execution time in minute for six methods.

Datasets	# Selected Feature	# Cells	# Class	Execution Time (in Minute)					
				sc-REnF_Renyi	*sc-REnF_*Tsallis	Gini Clust	M3Drop	HVG	GLM-PCA
Data1	500	500	2	4	9	2	1	1	3
Data2		1000	3	9	13	2	1	1	7
Data3		1500	4	12	19	3	1	1	11
Data4		2000	5	15	21	5	3	1	14

method, so it takes more time than any filter based feature selection technique (e.g. HVG, M3Drop). To check how the competing methods scales with the number of cells (and classes) we performend an analysis. We have generated simulated data (using splatter) by varying the number of cells (and classes). Four simulated data are generated with the number of cells and classes as follows: 500 cells with two classes, 1000 cells with three classes, 1500 cells with four classes, and 2000 cells with five classes. All data are generated with equal group probabilities, 2000 number of features, fixed dropout rate (0.2), and 40% DE gene proportion. 500 features are selected in each case and the runtime is compared with different competing methods. The execution time (minute) for each dataset is given in Table 5.8.

5.3.8 Visualization of Clustering Results on CBMC Data

This analysis aims to show the features selected by sc-REnF leads to approximately pure clustering of cells. We provide a t-SNE visualization of clustering results on 7895 cord blood mononuclear cells (CBMCs) using the selected features with *sc-REnF* (utilizing *Tsallis* entropy). For clustering, we use Leiden clustering algorithm embedded in the pipeline of the Scanpy package. Fig. 5.2 panel-A and B shows the t-SNE visualization of cells with original labels and predicted cluster labels, respectively. Most of the clusters such as cluster-2, cluster-4 and cluster-14 determine unique cells in the data. For example, cluster-2 captures most of the samples of CD14+ Mono cells, while cluster-4 and cluster-14 represent samples of 'Nk' (Natural killer) and erythrocyte cells. Some clusters represent more than one cell, such as cluster-13 includes DCs and pDCs, cluster-11 includes erythrocyte, Mks, and CD34+ cells. Individual mapping of cells to different clusters is depicted

in Fig. 5.2, panel-D. For example 'NK' cells are captured by two clusters, cluster-4 and cluster-12. Despite being multiple associations of clusters into one particular cell type, in most cases, one cluster captures major samples of a particular cell type.

5.4 Conclusions

Clustering of cells in scRNA-seq data is an essential step for cell type discovery from a large population of cells. Owing to the large feature/gene set of scRNA-seq data, the selection of the most variable genes is crucial in the preprocessing step, which has an immense effect in the later stages of downstream analysis. The proposed method *sc-REnF* addressed this issue by using (*Renyi*,and *Tsallis*) based feature selection method for identifying possible informative genes in the preprocessing steps. *sc-REnF* has the advantage over the conventional statistical approach in that it can consider cell-to-cell dependency based on generalized and wide spectrum entropy measures *Renyi* and *Tsallis*. We demonstrated that *sc-REnF* using *Renyi* and *Tsallis* methods introduces major advantages both in terms of clustering accuracy and marker gene detection in the downstream analysis of scRNA-seq data.

sc-REnF yields a stable feature/gene selection with a controlling parameter (q) for *Renyi* and *Tsallis* entropy. The optimal controlling parameter (q) is determined by applying it in simulated scRNA-seq datasets. We later demonstrated that the range of selected $q - values$ is applicable in the real-life scRNA-seq data clustering task. The four scRNA-seq data where we apply *sc-REnF* yields accurate clustering results which are validated by the ARI score. The stability of *sc-REnF* is demonstrated by evaluating its performance using KruskalWallis test. While applying *sc-REnF* multiple times with a varying number of features, the resulting ARI scores employ a minimum deviation (p-value $\ll 0.05$ for Kruskal-Wallis) for *Renyi* and *Tsallis* entropy.

Although the primary objective of *sc-REnF* is variable gene selection in the preprocessing step of scRNA-seq data, we extend the process towards the later stage of downstream analysis. We employ the clustering technique to group the cells using those selected genes. A precise clustering of cells also demonstrates the efficacy of our method for selecting the most variable genes in the first stage. This facilitates the selection of novel marker genes within each cluster. We pinpointed several markers, which show a high expression level within a particular cluster, among them some are also identified in previously published cell marker database.

Classifying unknown cells based on the reference data is a crucial problem for the identification of cell types in scRNA-seq cell classification. We addressed the problem by using a classification model trained with selected genes of the

reference data and applying it for classifying the unknown cell samples of test data. We demonstrate the advantage of using selected genes by *sc-REnF* in classifying the unknown test sample. We observed good classification accuracy, suggesting that selected genes from the reference data are also effective to produce a perfect classification in a completely unknown test sample.

The execution time of *sc-REnF* is directly proportional to the number of selected features and can be expensive when one needs to select a large number of features. However, this can be easily tackled with ever-increasing computing power in advanced servers.

Taken together, the proposed method *sc-REnF* not only has good performance on informative gene selection in the preprocessing step but also can explore the classification of unknown cells in the scRNA-seq data. Despite being applied in feature selection of different domains, the application of *Renyi*, and *Tsallis* entropies show good potential in gene selection and cell clustering of scRNA-seq data. Results show that *sc-REnF* not only leads in the domain of robust feature (gene) selection analyses but accelerates the investigations of cell type definition in large scRNA-seq data as well. We believe that *sc-REnF* may be an important tool for computational biologists to explore the most informative genes and marker genes in the downstream analysis of scRNA-seq data.

In next chapter, we sort out the problem of feature selection in High Dimensional Small Sample (HDSS) datasets using a generative model.

6

Generating realistic cell samples for gene selection in scRNA-seq data: A novel generative framework

6.1 Introduction

Recently, the emergence of high dimensional biological data such as single cell RNA sequence (scRNA-seq) data has posed a significant challenge to machine learning researchers [191, 163]. The high dimension, and small sample size (HDSS) data handling is difficult for downstream analysis, particularly for feature selection (FS). It affects later stages of downstream analysis such as cell clustering, marker selection, and annotation of cell clusters. A few outliers can drastically affect the FS techniques, and the selected feature sets may not be adequate to discriminate the classes [192]. Moreover, high dimensionality increases the computational time beyond acceptability.

High dimensional small sample (HDSS) data is prevalent in the single cell domain due to the budgetary constraint, ethical consideration of single cell experiments, or simply because of the small number of available patient samples. Whatever the reason is, too few observations (cell sample) in the single cell data may create problems in the downstream analysis. This is because a small sample size may not reflect the whole population accurately, which surely degrades the performance of any model. The general pipeline of scRNA-seq downstream analysis starts with pre-processing (normalization, quality control) of the raw count matrix and then going through several steps which include identification of relevant genes, clustering of cells, and annotating cell clusters with marker genes [52, 193, 194, 161, 162]. Each step has a profound effect on the next stage of analysis. The gene selection step identifies the most relevant features/genes from the normalized/preprocessed data and has an immense impact on cell clustering. The general procedure for selecting relevant genes which are primarily based on high variation (highly variable genes) [195, 65] or significantly high expression (highly expressed genes) [52]

suffers from a small sample effect. The general FS techniques also failed to provide a stable and predictive feature set in this case due to the large size of the feature (gene). One way to solve this issue is to go for a robust and stable technique that does not overfit the data. A few attempts [196, 197, 198] were observed recently which embed statistical and information-theoretic approach. Although these methods result in stable features, however, these are not performed well in small sample scRNA-seq data.

Recently computational researchers gaining interest in this field. Some methods like cscGAN [72], Splatter [1], SUGAR [199] are already developed which use different techniques (like the generative model, statistical framework) to successfully simulate the samples of specific cell types or subpopulations. The challenge in this task is to handle the sparsity and heterogeneity of the cell populations which define the specific characteristics of scRNA-seq data. In this chapter, we propose a generative model to sort out this problem in HDSS scRNA-seq data. We use the generative adversarial model to generate realistic cell samples from a small number of available samples of HDSS scRNA-seq data. Generative Adversarial Network (GAN) [200] has already been shown to be a powerful technique for learning and generating complex distributions [201, 202]. However, the training procedure of GAN is difficult and unstable. The training suffers from instability because both the Generator (G) and the (D) model are trained simultaneously in a game that requires a Nash equilibrium to complete the procedure. Gradient descent does this, but sometimes it does not which results in a costly time-consuming training procedure.

The main contribution here is in modifying the G input which results in a fast training procedure. We create a subsample of original data based on the Locality Sensitive Hashing(LSH) technique and augment this with noise distribution, which is given as input to the G. Thus, the (G) does not take pure noise as input, instead, we introduce a bias in it by augmenting a subsample of data with the noise distribution. We theoretically proved that the global minimum value of the virtual training criterion of the G is less than the traditional GAN ($< -log4$). We develop a method, named *LSH-GAN*(Locality Sensitive Hashing based Generative Adversarial Network). *LSH-GAN* can able to generate realistic samples in a faster way than the traditional GAN. This makes *LSH-GAN* more feasible to use in the feature (gene) selection problem of scRNA-seq data. Gene selection and clustering on the generated samples of *LSH-GAN* provide excellent results for small-sample and high dimensional single cell data.

6.2 Methods

In the following, we will describe the workflow of our analysis pipeline.

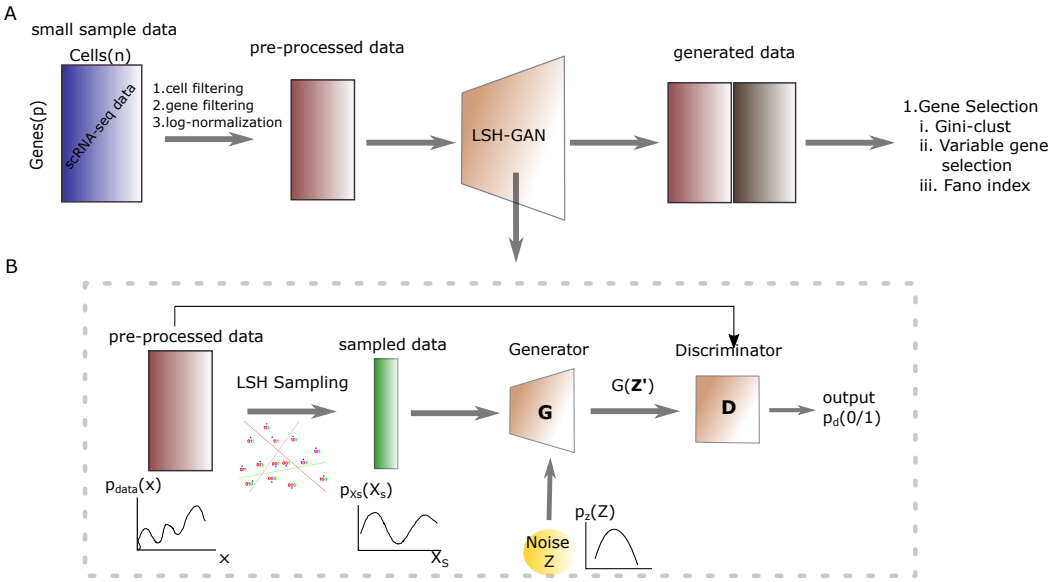


Figure 6.1: Panel-A: Figure shows the workflow for gene selection in HDSS scRNA-seq data using generated samples with *LSH-GAN* model. Panel-B shows the general architecture of *LSH-GAN*.

6.2.1 Proposed Model: *LSH-GAN*

The Fig. 6.1 describes the workflow of our analysis pipeline. Fig. 6.1, panel-A, describes the application of the proposed *LSH-GAN* model in the feature selection task of the HDSS scRNA-seq data, while Panel-B depicts basic building blocks of the model. The following subsections describe the framework in brief.

LSH step: sampling of input data

LSH [98, 203, 197] is widely used in nearest neighbor searching to reduce the dimensionality of data. LSH utilizes locality-sensitive hash functions which hash similar objects into the same bucket with a high probability. The number of buckets is much lesser than the universe of possible items, thus reduces the search space of the query objects (see chapter 1 for a detailed description of LSH technique).
In this work first, the unique hash codes which depict the local regions or neighborhoods of each data point are produced. For this, we utilized python sklearn implementation of *LSHForest* module with default parameters.
An approximate neighborhood graph (k-nn graph) is constructed by using $k = 5$ for each data point. This step computes the euclidean distances between the query point and its candidate neighbors. Sampling is carried out in a 'greedy' fashion where each data point is *traversed* sequentially and its corresponding five nearest

neighbors are flagged out which never visited again. Thus after one traversing a sub-set of samples is obtained which is further down-sampled by performing the same step iteratively.

Generator of *LSH-GAN*

The $G(.)$ function is modified by augmenting its taken input data. Instead of giving the pure noise ($p_z(z)$) as input we augment a subsample of real data distribution ($p_{data}(x)$) with it. The sampling of the input data is done in the LSH step. Thus the $G(.)$ function builds a mapping function from $\widehat{z}$ to data space (x) as $G(\widehat{z}; \theta_g)$ and is defined as: $G(.) : \widehat{z} \rightarrow x$. Modifying the $G(.)$ in this way we claim that it can increase the probability of generating samples of real data in lesser time.

Discriminator of *LSH-GAN*

Here $D(.)$ takes both the real data $p_{data}(x)$ and generated data coming from ($G(\widehat{z})$), with probability density ($p_{\widehat{z}}(\widehat{z})$) and returns the scalar value, $D(x)$ that represents the probability that the data x is coming form the real data: $D(.) : x \rightarrow [0, 1]$.
So, the value function can be written as:

$$L(D, G) = \min_G \max_D (E_{x \sim p_{data}(x)} \log(D(x)) + E_{\widehat{z} \sim p_{\widehat{z}}(\widehat{z})} \log(1 - D(G(\widehat{z})))) \tag{6.1}$$

Discriminator, and Generator form a two-player minimax game with value function $L(D, G)$. We train D to maximize the probability of correctly validate the real data and generated data. We simultaneously train G to minimize $\log(1 - D(G(\widehat{z})))$, where $G(\widehat{z})$ represents the generated data from the G by taking the noise (p_z) and the sampled data $p_{x_s}(x_s)$ as input.

Feature/gene selection using *LSH-GAN*

The generated cell samples are utilized for the gene selection task. We have employed five well known gene selection methods (with default parameters) of scRNA-seq data are adopted for validation: *GLM-PCA* [64], *CV^2 Index*, M3Drop [204] , *Fano Factor* [134] and Highly Variable Gene (HVG) selection of Seurat V4[57]. Single cell clustering method (SC3) technique is utilized to validate the selected genes from the generated samples.
The whole algorithm and the sampling procedure are described in the '*LSH-GAN* algorithm'.

6.2.2 Theoretical Analysis of *LSH-GAN*

In this section, we first provide a short description of Generative Adversarial Network (GAN) and then explain the theoretical foundation of *LSH-GAN* model.

Algorithm *LSH-GAN* Algorithm

Input: Data Matrix (**X**), number of Training iterations, number of nearest neighbor (**k**), number of iterations for sub-sampling (**t**)

Output: Generated data ($\mathbf{G_{out}}$).

for number of training iterations **do**
  X_s=LSH-SAMPLING(X,k,t)
  augment $p_{x_s}(X_s)$ with prior noise $p_z(z)$ and give this ($p_{\widehat{z}}(\widehat{z})$) to the, G.
  real data $p_{data}(x)$ and generated data $p_g(x)$ is given to D.
  Update the , D(.)
  $\Delta_d = \sum_{i=1}^{n} \log(D(x_i)) + \log(1 - D(G(\widehat{z})_i)))$
  Update the G(.)
  $\Delta_g = \sum_{i=1}^{n} \log(1 - D(G(\widehat{z})_i)))$
end for
{The adaptive momentum gradient decent rule is used in our experiment.}
procedure LSH-sampling(x, k, t)
Execute LSH (LSH) on x and prepare a k-Nearest Neighbour list for each data point.
for number of iteration of sub-sampling t **do**
  visit each data point sequentially in the order as it appears in data.
  if the data point is not visited earlier, select the data point and discard all its k neighbors from its nearest-neighbour list.
end for
end procedure

Generative adversarial network

Generative adversarial network (GAN) is introduced in [200] which was proposed to train a generative model. GAN consists of two blocks: a generative model (G) that learn the data distribution ($p(x)$), and a discriminative model (D) that estimates the probability that a sample came from the training data (X) rather than from the G. These two models can be non-linear mapping functions such as two neural networks. To learn the G distribution p_g over data x, a differentiable mapping function is built by G to map a prior noise distribution $p_z(z)$ to the data space as $G(z;\theta_g)$. The D function $D(x;\theta_d)$ returns a single scalar that represents the probability of x coming from the real data rather than from G distribution p_g. The goal of the G is to fool the D, which tries to distinguish between true and generated data. Training of D ensures that the D can properly distinguish samples coming from both training samples and the G. G and D are simultaneously trained to minimize $log(1-D(G(z))$ for G and maximize $log(D(x))$ for D. It forms a two-player min-max game with value function $V(G, D)$

$$\min_G \max_D V(G, D) = E_{x\sim p_x(x)}[\log(D(x))] + E_{z\sim p_z(z)}[1 - log(D(G(z)))] \tag{6.2}$$

LSH Generative adversarial network

For *LSH-GAN*, a sub-sampling of real data $p_{x_s}(x_s)$ is augmented with the prior noise distribution, $p_z(z)$. Due to this additional information in G, we assume that the probability $D(G(\widehat{z}))$ will increase by a factor, ζ.

Proposition 1: $L(D, G)$ is maximized with respect to (D), for a fixed (G), when

$$D^*_G(x) = \frac{p_{data}(x)(1-\zeta)}{p_{data}(x) + p_g(x)} \tag{6.3}$$

Proof. Equation 6.2 can be written as

$$\begin{aligned} L(D,G) &= \int_x p_{data}(x)\log(D(x))dx + \int_{\widehat{z}} p_{\widehat{z}}(\widehat{z})\log(1-\{D(G(\widehat{z}))+\zeta\})d\widehat{z} \\ &= \int_x p_{data}(x)\log(D(x)) + p_g(x)\log(1-\{D(x)+\zeta\})dx \\ &\qquad \text{[As the range of } D(G(\widehat{z})) \text{ is within the domain of real data } x \text{ so we can write this]} \end{aligned} \tag{6.4}$$

We know that, the function $y = a\log x + b\log(1-(x+\zeta))$ will have maximum value, at $x = \frac{a(1-\zeta)}{a+b}$, for any $(a,b) \in R^2\{0,0\}$ and $\zeta \in (0,1)$. So, the optimum value of D for a fixed G is:

$$D^*_G(x) = \frac{p_{data}(x)(1-\zeta)}{p_{data}(x) + p_g(x)} \tag{6.5}$$

The training objective for D is to maximize the log-likelihood of the conditional probability $P(Y = y|x)$, where Y signify whether x is coming from real data distribution($y = 1$) or coming from the $G(y = 0)$. Now the equation 6.2 can be written as

$$\begin{aligned} C(G) &= \max_D L(G,D) \\ &= (E_{x\sim p_{data}(x)}\log(D^*_G(x)) + E_{\widehat{z}\sim p_g(\widehat{z})}\log(1-D^*_G(G(\widehat{z})))) \\ &= (E_{x\sim p_{data}(x)}\log(D^*_G(x)) + E_{x\sim p_g(x)}\log(1-D^*_G(x))) \\ &= E_{x\sim p_{data}(x)}\log\frac{p_{data}(x)(1-\zeta)}{p_{data}(x)+p_g(x)} + E_{x\sim p_g(x)}\log(1-\frac{p_{data}(x)(1-\zeta)}{p_{data}(x)+p_g(x)}) \end{aligned} \tag{6.6}$$

Theorem 1: At $p_g(x) = p_{data}(x)$ (global minimum criterion of value function $L(G,D)$), the value of $C(G)$ is less than $(-\log 4)$.

proof From equation 6.6 we get

$$
\begin{aligned}
C(G) &= E_{x\sim p_{data}(x)} \log\left(\frac{p_{data}(x)(1-\zeta)}{p_{data}(x)+p_g(x)}\right) + E_{x\sim p_g(x)} \log\left(1 - \frac{p_{data}(x)(1-\zeta)}{p_{data}(x)+p_g(x)}\right) \\
&= E_{x\sim p_{data}(x)} \log\left(\frac{p_{data}(x)(1-\zeta)}{p_{data}(x)+p_g(x)}\right) + E_{x\sim p_g(x)} \log\left(\frac{\zeta p_{data}(x)+p_g(x)}{p_{data}(x)+p_g(x)}\right) \\
&= \left[\log(1-\zeta) + E_{x\sim p_{data}(x)} \log\left(\frac{p_{data}(x)}{p_{data}(x)+p_g(x)}\right)\right] \\
&+ \left[E_{x\sim p_g(x)} \log\left(1 + \frac{\zeta p_{data}(x)}{p_g(x)}\right) + E_{x\sim p_g(x)} \log\left(\frac{p_g(x)}{p_{data}(x)+p_g(x)}\right)\right] \\
&= \left[\log(1-\zeta) + E_{x\sim p_g(x)} \log\left(1 + \frac{\zeta p_{data}(x)}{p_g(x)}\right)\right] \\
&+ \left[E_{x\sim p_{data}(x)} \log\left(\frac{p_{data}(x)}{p_{data}(x)+p_g(x)}\right) + E_{x\sim p_g(x)} \log\left(\frac{p_g(x)}{p_{data}(x)+p_g(x)}\right)\right] \\
&= \left[\log(1-\zeta) + E_{x\sim p_g(x)} \log\left(1 + \frac{\zeta p_{data}(x)}{p_g(x)}\right)\right] + \left[(-\log 4) + 2JSD(p_{data}(x)\|p_g(x))\right]
\end{aligned}
\tag{6.7}
$$

where, $JSD(p_{data}(x)\|p_g(x))$ represents Jensen–Shannon divergence between two distributions p_{data} and p_g. Now, if the two distribution are equal, Jensen–Shannon divergence (JSD) will be zero. Thus, for global minimum criterion of the value function ($p_g = p_{data}$) the Equation 6.7 is reduces to,

$$
C(G) = \log(1-\zeta) + \log(1+\zeta) + (-\log 4) = \log\frac{(1-\zeta^2)}{4} \leq (-\log 4) \tag{6.8}
$$

This completes the proof.

6.3 Results and Discussions

6.3.1 Datasets Description

The brief description of dataset is given here. The single-cell RNA sequence datasets used for evaluation of our proposed approach are downloaded from Gene Expression Omnibus (GEO) https://www.ncbi.nlm.nih.gov/geo/.Those are discussed below.

- **Yan Dataset:** This is a human preimplantation embryo and embryonic stem cell dataset. The average total read count in the expression matrix is 25,228,939 reads. There are 7 cell types, including labelled 4-cell, 8-cell, zygote, Late blastocyst and 16-cell, downloaded from GEO under accession no. GSE36552 [143].

- **Klein Dataset:** This dataset was generated by the droplet barcoding method with an average total read count of 20,033.40 reads in the expression matrix. A total of eight single cell data sets are submitted: 3 for mouse embryonic stem (ES) cells (1 biological replicate, 2 technical replicates); 3 samples following LIF withdrawal (days 2,4, 7); one pure RNA data set (from human lymphoblast K562 cells); and one sample of single K562 cells. The dataset was downloaded from GEO under accession no.GSE65525 [171].

- **Pollen:** scRNA-seq: Strand-specific reads were aligned to the human reference genome, Ensembl GRCh37/hg19 release 75, using TopHat v2.0.10 with the flags (–library-type fr-firststrand –microexon-search). De novo transcriptome assembly was performed separately on rRNA depletion total RNA-seq alignments, and on polyA selection RNA-seq alignments, using Cufflinks v2.2.1 with the flags Dataset Libraries were generated from 600 individual cells in parallel. It contains 11 cell types. The dataset was downloaded from GEO under accession no GSM1832359 [144].

- **Darmanis:** It contains single cell RNA sequencing on 466 cells to capture the cellular complexity of the adult and fetal human brain at a whole transcriptome level. Healthy adult temporal lobe tissue was obtained from epileptic patients during temporal lobectomy for medically refractory seizures. The dataset was downloaded from GEO under accession no GSE67835 [205].

- **Melanoma [172]**: The dataset describes the diversity of expression states within melanoma tumors, it is obtained freshly resected samples, disaggregated the samples, sorted into single cells, and profiled them by single-cell RNA-seq. It is downloaded from GEO under accession no. GSE72056. It contains 19783 number of genes and 68579 cells with 14 cell types. Tumors were disaggregated, sorted into single cells, and profiled by Smart-seq2.

Table 6.1: A brief summary of the datasets used in the experiments.

# Serial	Dataset Name	Features	Instances	Class
1	Yan [143]	20214	90	7
2	Klein [171]	24175	2717	4
3	Darmanis [205]	22088	466	9
4	Pollen [144]	23794	299	11
5	Melanoma [172]	19783	68579	14

6.3.2 Data Preprocessing

The raw count matrix $D \in \mathcal{R}^{C \times G}$, where C and G represents the number of cells and genes, respectively, is normalized using *Linnorm* [54] Bioconductor package of R. We select cells having more than a thousand expressed genes (non zero values)

and choose genes having a minimum read count more than 5 in at least 10% of the cells. log_2 normalization is performed on the transformed matrix by adding one as a pseudo count.

6.3.3 Experimental Settings

The number of nearest neighbor (k) and the number of iteration (t) are two main parameters of the LSH-step (see Algorithm), tuning of which affects the amount of sampling given to the G for training the *LSH-GAN* model. We vary k and t in the range $\{5, 10, 15, 20\}$ and $\{1, 2\}$, respectively, and choose that value for which the Wasserstein distance [201] between generated and real samples are reported to be minimum. We fixed the amount of sampling using $k = 5, t = 1$ for Pollen, Yan, Darmanis datasets and $k = 5, t = 2$ for Klein dataset and Melanoma datasets (see Fig. 6.2) . Similarly, we choose the epoch (e_{opt}), which results in the lowest Wasserstein metric. For example, we take e_{opt} as 10k, 30k, 10k, 15k, and 25k for the dataset Darmanis, Yan, Pollen, Klein, and Melanoma respectively (see Fig. 6.2). For generating hash code from LSH sampling, *LSHForest* of *scikit-learn* version 0.19.2 is utilized.
We take the adaptive learning rate optimization algorithm implemented in ADAM optimizer in python Tensorflow version 1.9.2. $G(.)$ and $D(.)$ uses 2-layer multilayer perceptrons with hidden layer dimension as $(16, 16)$. For traditional GAN, we retain the same settings as *LSH-GAN* for G and D networks.
For benchmarking our method we have utilized three state-of-the-art techniques widely used for sample generation: cscGAN [72], SUGAR [199], and Splatter [1]. For these three methods, We adopted the code (with default parameters) provided on the Github page of the original publications.
Five well known gene selection methods (with default parameters) of scRNA-seq data are adopted for validation: *GLM-PCA* [64], CV^2 *Index*, M3Drop [204] , *Fano Factor* [134] and Highly Variable Gene (HVG) selection of Seurat V4[57]. We select the top 500 features (genes) using all three feature selection methods on scRNA-seq datasets. For validation purposes, Wasserstein metric [201] is utilized to estimate the quality of the generated data. Clustering of scRNA-seq data is performed using *SC3* [131] technique with default parameters. Clustering performance is evaluated using the Adjusted Rand Index (ARI).

6.3.4 Parameter Selection of *LSH-GAN*

The number of nearest neighbor (k) and the number of iteration (t) are two main parameters of the LSH-step (see Algorithm-1 in main text), tuning of which affects the amount of sampling given to the G for training the *LSH-GAN* model. We vary k and t in the range $\{5, 10, 15, 20\}$ and $\{1, 2\}$, respectively, and choose that value for

Table 6.2: Wasserstein distance between generated and real samples for different range of parameters k and t.

	t=1					t=2				
	Yan	Pollen	Darmanis	Melanoma	Klein	Yan	Pollen	Darmanis	Melanoma	Klein
k=5	0.23	0.21	0.29	0.37	0.39	0.29	0.28	0.35	0.31	0.3
k=10	0.24	0.28	0.35	0.4	0.43	0.32	0.3	0.39	0.36	0.32
k=15	0.29	0.33	0.4	0.46	0.48	0.37	0.38	0.41	0.38	0.4
k=20	0.3	0.37	0.41	0.5	0.5	0.43	0.44	0.48	0.4	0.42

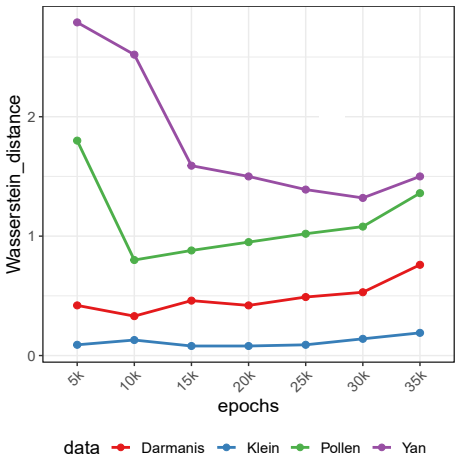


Figure 6.2: Figure shows Wasserstein metric between real and generated data distribution across different epochs for scRNA-seq datasets.

which the Wasserstein distance between generated and real samples is reported to be minimum. We fixed the amount of sampling using $k = 5, t = 1$ for Pollen, Yan, Darmanis datasets and $k = 5, t = 2$ for Klein dataset and Melanoma datasets (see Table 6.4)

We trained the LSH-GAN model in five scRNA-seq datasets: Darmanis, Yan, Pollen, Klein and Melanoma. Here, a sub-sample of real data distribution is augmented with prior noise and used as the input to the G network. The generated data using *LSH-GAN* (with k=5) is validated by computing the Wasserstein metric between the real and generated data distribution for different epochs (see Fig. 6.2). For each data, we note the epoch (e_{opt}), which results in the lowest Wasserstein metric. For example, we take e_{opt} as 10k, 30k, 10k, 15k, and 25k for the dataset Darmanis, Yan, Pollen, Klein, and Melanoma respectively.

Table 6.3: Wasserstein distance between generated and real data distribution. Model is trained on synthetic data of size 100×1000 Gaussian mixture data with 2 non-overlapping classes.

Nearest Neighbour	Model	Epoch			
		10000	15000	20000	25000
$k = 5$		0.46	0.35	0.33	0.45
$k = 10$		1.09	0.89	0.83	0.82
$k = 15$	*LSH-GAN*	1.36	0.89	1.45	0.87
$k = 20$		1.53	1.35	1.19	0.83
	GAN	1.71	1.73	1.75	1.70

6.3.5 *LSH-GAN* Improves Performance of Traditional GAN on Simulated Data

First, we train the *LSH-GAN* on HDSS synthetic data and generate realistic samples to compare against the traditional GAN model. For this, we create a 2-class non-overlapping Gaussian mixture data consisting 100 samples and 1000 features by taking the mean (μ) of the data in the range of 5 to 15 for class-1 and -15 to -5 for class-2. The covariance matrix (Σ) is taken for all the samples using the formula $\Sigma = (\rho^{|i-j|})$, where i, j are row and column index, and ρ is equal to 0.5. We calculate Wasserstein metric to estimate the quality of the generated data. The Wasserstein distance between the real data distribution (p_{data}) and the generated data distribution (p_g) to estimate the quality of the generated data. We use different settings of k^{th} (k=5, 10, 15, 20) nearest neighbor to generate sub-sample of data from LSH sampling procedure. In each case, the sampled data (p_{x_s}) is augmented with prior noise (p_z) and given to the G of *LSH-GAN* for model training.

For comparison with the traditional GAN model, we use the data with train: test split of 80:20 and calculate the Wasserstein metric between the test sample and the generated sample. Table 6.3 shows the values of the metric for *LSH-GAN* and traditional GAN model in different range of epochs and nearest neighbors k. A closer look into the Table 6.3 reveals that the performance of *LSH-GAN* (at 10000 epoch and $k = 5$) is far better than the traditional GAN model with 25000 epochs. Notably, for less amount of sampling (larger k), *LSH-GAN* needs more iterations for training. As for particular example, the performance of *LSH-GAN* achieved on 20000 epoch and $k = 20$ is rivaled only at 10000 epoch for $k = 10$. Thus it is evident from the results that reducing the amount of sampling needs more epochs and thus needs more training time for the *LSH-GAN* model to converge. Fig. 6.3 also supports this statement. Here, the two models (*LSH-GAN*, and traditional GAN) are trained to simulate a two dimensional synthetic data of known distribution, for which the *LSH-GAN* can able to generate samples that are more real than the traditional GAN, in a lesser number of iteration.

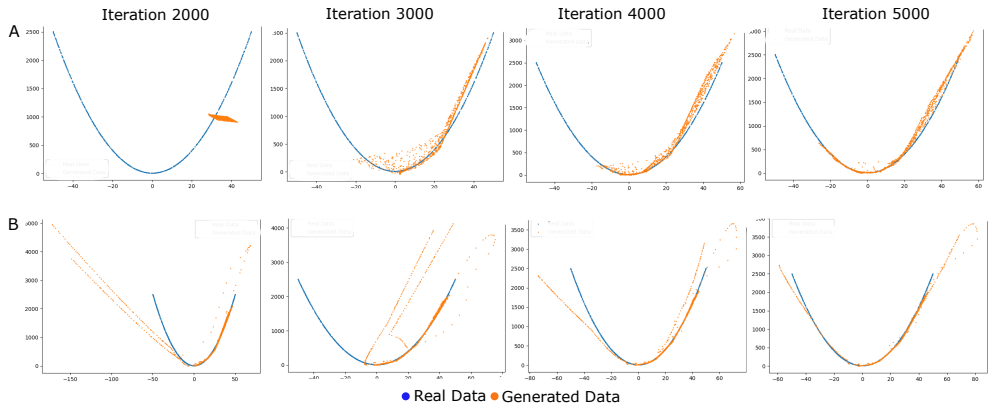


Figure 6.3: Generation of two dimensional synthetic data using traditional GAN (upper row, Panel-A) and *LSH-GAN* (lower row, Panel-B) model for different epochs.

6.3.6 Comparison of *LSH-GAN* with Benchmarks in HDSS scRNA-seq Data

We compare *LSH-GAN* with four existing benchmarks: cscGAN [72], splatter [1], SUGAR [199] and traditional GAN [200]. Since the evaluation of the generative model is notoriously difficult, we first use Wasserstein distance to compare the real data distribution and generated data distribution coming from different competing models. We also used UMAP visualization, and marker genes expression to visualize the generated cell samples. Fig. 6.4, panel-A-C shows the two-dimensional UMAP representation of generated and real cell sample from the test data for four competing models. Melanoma data is utilized for this experiment. As can be seen from this figure, *LSH-GAN* can able to retain the distribution of the original cell samples. This can also be supported by the Wasserstein distance (see Fig. 6.4, panel-F) measured between real data and generated data distribution. To know how the expression of the marker genes are retained in the generated data, we plot the expression of marker gene CD8A (marker for CD8T cell) and MS4A1 (marker for B cell) in the two dimensional UMAP space for both the real and generated samples of *LSH-GAN* (see Fig. 6.4, panel-D-E). It reveals from the figure that marker genes CD8A and MS4A1 show similar expression patterns (high expression) both in real and generated cell samples.

We also validate the generated samples by training a classifier (random forest) to see whether it can able to distinguish the samples coming from two different distributions (real and generated). The aim is to see whether the model can discriminate between the real and generated cell samples accurately. Table 6.4 shows the cross validation AUC score of the random forest classifier for five scRNA-seq datasets. It reveals from the table that for *LSH-GAN*, the AUC scores

Table 6.4: Table shows results of applying random forest classifier for discriminating real and generated samples coming from different competing methods. The average AUC score (with 5-fold cross validation) is reported for each dataset.

	AUC Score				
	Yan	Darmanis	Pollen	Klein	Melanoma
cscGAN	0.65 ±0.02	0.68 ±0.01	0.64 ±0.02	0.62 ±0.02	0.66 ±0.01
Splatter	0.69 ±0.01	0.69 ±0.02	0.67 ±0.03	0.65 ±0.01	0.72 ±0.02
SUGAR	0.67 ±0.02	0.66 ±0.03	0.61 ±0.02	0.64 ±0.02	0.68 ±0.01
GAN	0.72 ±0.02	0.71 ±0.02	0.73 ±0.02	0.72 ±0.02	0.76 ±0.03
LSH-GAN	0.59 ±0.01	0.60 ±0.02	0.58 ±0.02	0.57 ±0.01	0.60 ±0.01

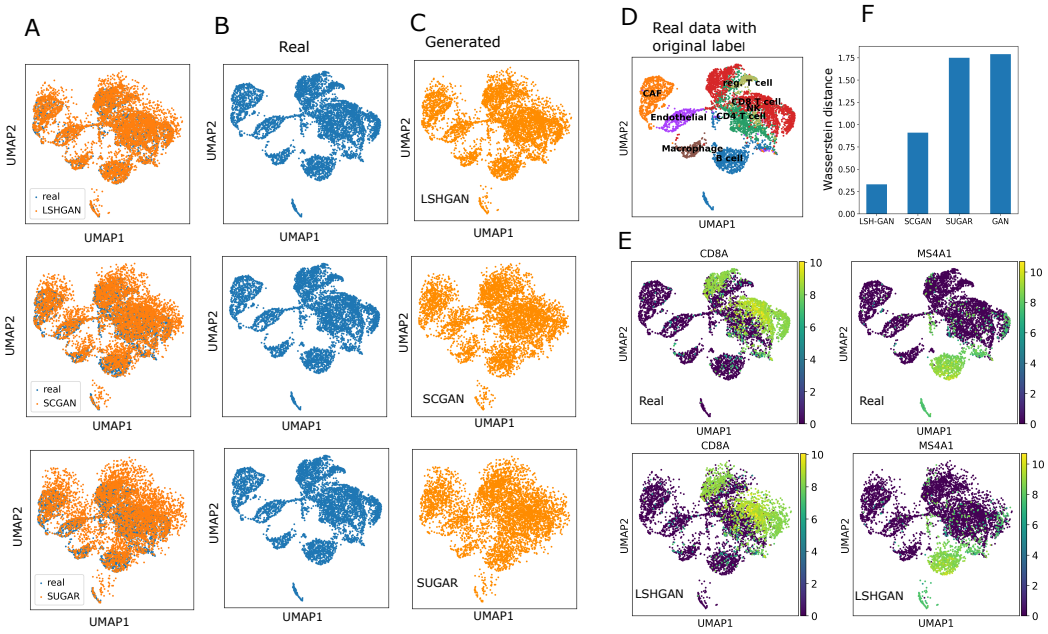


Figure 6.4: Panel–A-C: UMAP visualization of real and generated cell samples of melanoma data. Panel-D shows real data with the original labels. Panel-E shows the expression of two markers CD8A (marker of CD8 T cell) and MS4A1 (marker of B cell) in real and generated data. Panel-F shows a barplot which describe the Wasserstein distance between the generated and real cell sample.

hardly reach 0.6 (only for melanoma data) suggesting a chance-level performance of RF model. This suggests the generated data obtained from *LSH-GAN* is highly similar to the real data.

6.3.7 Gene Selection in HDSS scRNA-seq Data

Here, we aim to address the problem of gene selection in HDSS scRNA-seq data using the generated samples. We augment the generated sample with original

Table 6.5: Table shows the Adjusted Rand Index (ARI) scores of clustering results on the cell samples generated by the five competing methods.

Data	FS method	*LSH-GAN*	SUGAR	cscGAN	Splatter	GAN	without model
Darmanis	GLM-PCA	0.634	0.413	0.531	0.420	0.129	0.4
	Fano Factor	0.535	0.319	0.457	0.380	0.270	0.340
	CV^2 index	0.598	0.420	0.510	0.481	0.461	0.457
	M3Drop	0.648	0.513	0.580	0.507	0.480	0.460
	HVG (Seurat V4)	0.680	0.510	0.556	0.539	0.460	0.430
Yan	GLM-PCA	0.895	0.709	0.798	0.715	0.62	0.66
	Fano Factor	0.821	0.790	0.801	0.768	0.730	0.713
	CV^2 index	0.891	0.801	0.825	0.793	0.719	0.70
	M3Drop	0.898	0.802	0.796	0.790	0.761	0.71
	HVG (Seurat V4)	0.910	0.811	0.891	0.802	0.810	0.80
Pollen	GLM-PCA	0.835	0.780	0.819	0.793	0.788	0.780
	Fano Factor	0.933	0.878	0.916	0.880	0.815	0.712
	CV^2 index	0.94	0.906	0.908	0.890	0.831	0.81
	M3Drop	0.918	0.864	0.897	0.790	0.758	0.735
	HVG (Seurat V4)	0.958	0.916	0.897	0.868	0.801	0.820
Klein	GLM-PCA	0.815	0.769	0.784	0.731	0.581	0.66
	Fano Factor	0.8	0.742	0.782	0.770	0.669	0.796
	CV^2 index	0.82	0.710	0.761	0.709	0.690	0.680
	M3Drop	0.837	0.794	0.769	0.718	0.61	0.607
	HVG (Seurat V4)	0.898	0.861	0.857	0.785	0.730	0.739

data to make the sample to feature ratio as 1.5. The augmented data is utilized for gene selection. Here, we have employed five feature selection methods (Highly Variable Gene (HVG) selection of Seurat V3/V4, M3Drop, GLM-PCA and *Fano Factor*, CV^2-index), widely used for the gene selection task in scRNA-seq data and one single cell clustering method (SC3) technique to validate the selected genes from the augmented data.

LSH-GAN is compared with five other state-of-the-arts in four HDSS scRNA-seq datasets (Darmanis, Yan, Pollen, and Klein datasets). We exclude Melanoma data for this analysis as it already has larger sample size compared to the feature size (sample:feature is 3.46). The aim is to know whether the selected features/genes from the generated combined data can lead to a pure clustering of cells. Table 6.5 shows the comparisons of the Adjusted Rand Index (ARI) values resulting from the cell clustering. It is evident from the table that features/genes selected from the generated combined data of the *LSH-GAN* model produce better clustering results than the other competing models. The last column of the Table 6.5 shows the ARI scores of clustering results with the original feature (gene) set.

6.3.8 Selected Genes using *LSH-GAN* can Effectively Predict Cell Clusters

Here we provide the detailed results of clustering on four datasets using the genes selected from the generated samples. For this, we adopted a widely used single

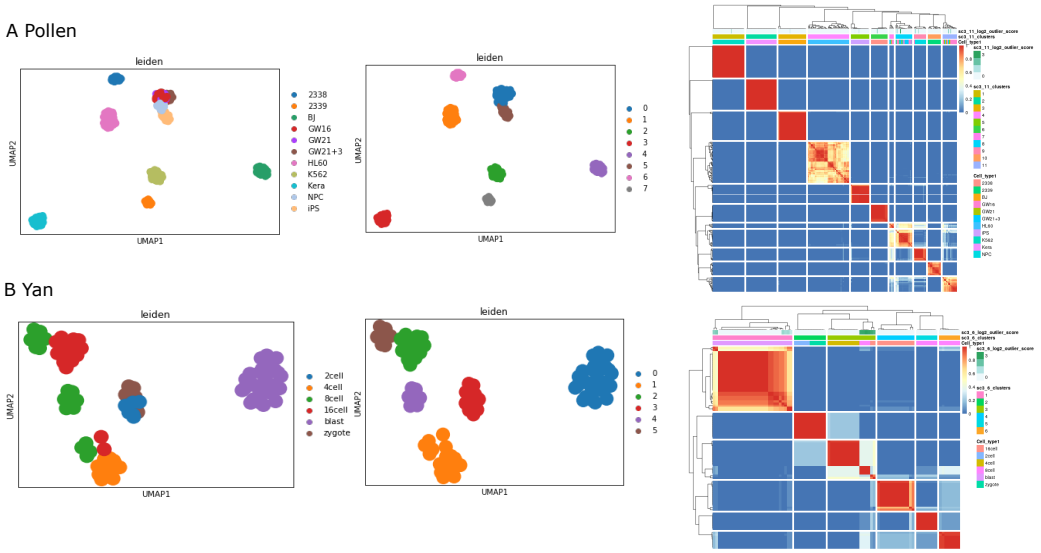


Figure 6.5: Figure shows the clustering results of Pollen and Yan data sets. Panel-A shows the t-SNE visualization of clustering results (original and predicted labels), whereas panel-B shows the consensus clustering plots of obtained clusters.

cell clustering method SC3 [131]. Fig. 6.5 panel- A, depicts the t-SNE visualization of predicted clusters and their original labels for Yan and Pollen datasets. Panel-B of Fig. 6.5 represents heatmaps of cell × cell consensus matrix. Each heatmap signifies the number of times a pair of cells is appearing in the same cluster [131]. Here two cells are said to be in different clusters if the score is zero (blue color). Similarly, a score '1' (red) signifies two cells are belonging to the same class. Thus completely red diagonals and blue off-diagonals represent a perfect clustering. A careful notice on the Fig. 6.5, panel-A and -B reveals a perfect match between the original and predicted labels for YAN and Pollen datasets.

6.4 Conclusions

In this paper, we present a novel and faster way of generating cell samples of HDSS single cell RNA-seq data using a generative model called *LSH-GAN*. We update the training procedure of generative adversarial network (GAN) using LSH which can produce realistic samples in a lesser number of iterations than the traditional GAN model. We utilized the generated data in the standard procedure of downstream analysis for analyzing real life scRNA-seq data. Particularly, we demonstrated that the recent and benchmark approaches of gene selection and cell clustering produce excellent results on the generated cell samples of *LSH-GAN*.

Our preliminary simulation experiment also suggests that for a fixed number of training iterations the proposed model can generate more realistic samples than the traditional GAN model. This observation is also established theoretically by proving that the cost of value function is less than $-log4$ which is the cost for traditional GAN at the global minimum of virtual training criterion ($p_g = p_{data}$).
We demonstrated that the generated samples of *LSH-GAN* are useful for gene selection and cell clustering in HDSS scRNA-seq data. particularly the excellent results of *LSH-GAN* over the recent benchmark methods support its usability for generating realistic cell samples. For validation of the generated cell samples, we use the conventional steps of downstream analysis for scRNA-seq data. We employ five widely used gene selection techniques and one single cell clustering technique for gene selection and grouping of cells. The precise clustering of cells demonstrates the quality of generated cell samples using the *LSH-GAN* model.
One limitation of our method is that for feature selection we hardly found any linear relationship between the clustering results with the sample size of generated scRNA-seq data. The correct sample size should be selected by using a different range of values between $0.25p$ to $1.5p$, where p is the feature size. There may be some effects of different parameters related to single cell clustering ($SC3$ method) and feature selection (e.g. different FS methods, number of selected features, etc.) which may play a critical role in the clustering performance. However, we found clustering results are always better for the generated data with more than 1p (p is the feature size) sample size. This observation suggests that for feature selection in HDSS data, whenever we produce samples larger than the feature size we will end up with a better clustering. The feasibility of generating such samples is justified by the faster training procedure of *LSH-GAN* model.
It may interesting to speculate how well *LSH-GAN* can be useful for generating data from other biological domains, particularly data generated from spatial transcriptomics. The obtained data from this technology has a spatial arrangement of cell types within a tissue and is thus extremely useful to understand normal development and disease pathology. In-silico generation of this data may find great interest to the machine learning researcher as the model should capture the location-wise heterogeneity of the real samples.
Taken together, the proposed model can generate good quality cell samples from HDSS scRNA-seq data in a lesser number of iteration than the traditional GAN model. Results show that *LSH-GAN* not only leads over the benchmarks in the cell sample generation of scRNA-seq data but also accelerates the way of gene selection and cell clustering in the downstream analysis. We believe that LSH-GAN may be an important tool for computational biologists to explore the realistic cell samples of HDSS scRNA-seq data and its application in the downstream analysis.

7

Conclusions and Future Scope of Research

Developing novel computational methods and algorithms for feature/gene selection in large biological datasets (mainly for bulk/single cell expression data) is the main focus of the thesis. The most relevant features/genes from the dataset are further analyzed through classification/clustering techniques to establish the efficacy of the selected feature subset.

In this thesis, first, we proposed Locality Sensitive PCA (*LSPCA*), a scalable variant of PCA equipped with structure aware data sampling at its core. *LSPCA* is applicable for all types of applications where PCA can be used, more specifically for large datasets like single-cell expression matrices. It is fast and produces components almost identical to the vanilla PCA components. The method provides flexibility to a user to adjust the number of samples to train the PCA. Compared to random sampling, structure aware sampling is a more effective way to sample from a large dataset. *LSPCA* performs dimension reduction by operating on a subset of less redundant samples without significantly altering the performance of the classic PCA.

We have also proposed *CBFS*, a multivariate copula based approach for feature selection which leverages the advantages of copula, namely scale invariance and ability to model multivariate dependencies. *CBFS* algorithm is applied on synthetic Gaussian Mixture datasets and real-life datasets where it is found to outperform other competing methods in most of the cases. This makes *CBFS* well accepted in a large domain of datasets (text, image, and biological).

We have extended this copula based dependency measure for the selection of informative genes in scRNA-seq data. The proposed method *RgCop* addressed this gene selection problem by employing a robust and equitable dependence measure called copula-correlation (*Ccor*). It can accurately measure relevance and redundancy simultaneously between two sets of genes. *RgCop* also adds simple l_1 regularization technique with its objective function to control the large coefficients of relevancy terms. *RgCop* provides a stable feature/gene selection which is robust to noise in the data due to *scale invariant property* of copula.

RgCop is a supervised feature/gene selection method for single cell RNA-seq data.

In some applications where the class labels are not available, unsupervised feature selection methods are the only way to reduce the dimensionality of the data. We have developed a Graph Convolution Network(GCN) based feature (gene) selection method called *sc-CGconv* for scRNA-seq data which leverages the Copula dependency measure. Copula is utilized to generate a cell-cell dependency graph from large single cell data. Graph convolution network is utilized to extract the low dimensional embedding from the constructed graph which is utilized as the extracted features from the data. There are two striking characteristics of *sc-CGconv*: i) It can capture the cell-to-cell variability of the single cell RNA-seq data. ii) GCN utilizes a dependency graph and extracts a low dimensional embedding.

We also developed another application of copula in the domain of unsupervised feature selection from the bulk microarray data. We developed *CODC*, a copula based unsupervised model to detect differential coexpression of genes in two different samples. *CODC* seeks to identify the dependency between the expression patterns of a gene pair in two different conditions. Copula is used to model the dependency in the form of two joint distributions. Kolmogorov-Smirnov distance between two joint distributions is treated as the differential coexpression score of a gene pair. The scale invariant property of copula is inherited into *CODC* to make it robust against noisy expression data. It is advantageous for detecting a minor change in correlation across two different conditions which is the most desirable feature of any differential coexpression analysis. We have also analyzed the identified modules enriched with different biological pathways and highlighted gene pairs such as 'POSTN-KONK5', 'ALPL-RDH16', 'LPO-LEMP2' 'KONK5-GPX2 as highly differentially coexpressed which would be potential biomarkers for the corresponding disease.

Owing to the large feature/gene set of scRNA-seq data, the selection of most variable genes is crucial in the preprocessing step, which has an immense effect on the later stage of downstream analysis. We have developed *sc-REnF* (Robust entropy based feature selection on single cell RNA seq data) to address this issue by using an entropy (*Renyi*, *Tsallis*) based feature selection method for identifying possible informative genes in the preprocessing steps. *sc-REnF* has the advantage over the conventional statistical approach in that it can consider the cell-to-cell dependency based on generalized and wide spectrum entropy measures *Renyi* and *Tsallis*. We demonstrated that *sc-REnF* using *Renyi* and *Tsallis* entropies introduces major advantages both in terms of clustering accuracy and in terms of marker gene detection in the downstream analysis of scRNA-seq data.

A fundamental problem of downstream analysis of scRNA-seq data is the unavailability of enough cell samples compared to the feature size. This is mostly due to the budgetary constraint of single cell experiments or simply because of the small number of available patient samples. In this thesis, we presented a

novel and fast way of generating cell samples of High dimensional Small Sample (HDSS) single cell RNA-seq data using a generative model called *LSH-GAN*. We update the training procedure of generative adversarial network (GAN) using locality-sensitive hashing which can produce realistic samples in smaller number of iterations than the traditional GAN model. We utilized the generated data in the standard procedure for analyzing real-life scRNA-seq data. Particularly, we demonstrated that the recent and benchmark approaches of gene selection and cell clustering produce excellent results on the generated cell samples of *LSH-GAN*.

Single cell technology provides an exciting opportunity to analyze data at the cell level and can be useful for different bioinformatic analyses which were not possible in the era of bulk sequencing. On the other hand, this also gives us numerous challenges for analyzing the data computationally, most important for small sample data, a researcher trying to utilize several machine learning and deep learning models to fix the problem in the analysis of the data. Most of the devised methods like *RgCop*, *sc-REnF*, *CBFS*, *LSH-GAN* are wrapper based feature selection techniques. So it may consume more time than any other filter based method. For the future direction, it may be a good idea to use generative models for simulating spatial transcriptomic data which is a recently groundbreaking molecular profiling method that allows scientists to measure all the gene activity in a tissue sample and map where the activity is occurring. It may be interesting to speculate how well *LSH-GAN* can be useful for generating data from other biological domains, particularly data generated from spatial transcriptomics. The obtained data from this technology has spatial arrangement of cell types within a tissue and is thus extremely useful to understand normal development and disease pathology. In-silico generation of this data may find great interest to the machine learning researcher as the model should capture the location-wise heterogeneity of the real samples.

Bibliography

[1] L. Zappia, B. Phipson, and A. Oshlack, "Splatter: simulation of single-cell rna sequencing data," *Genome biology*, vol. 18, no. 1, pp. 1–15, 2017.

[2] S. K. Baliarsingh, S. Vipsita, K. Muhammad, B. Dash, and S. Bakshi, "Analysis of high-dimensional genomic data employing a novel bio-inspired algorithm," *Applied Soft Computing*, vol. 77, pp. 520–532, 2019.

[3] X.-Y. Zhang, Y. Bengio, and C.-L. Liu, "Online and offline handwritten chinese character recognition: A comprehensive study and new benchmark," *Pattern Recognition*, vol. 61, pp. 348–360, 2017.

[4] J.-P. van Oosten and L. Schomaker, "Separability versus prototypicality in handwritten word-image retrieval," *Pattern Recognition*, vol. 47, no. 3, pp. 1031–1038, 2014.

[5] P. Boileau, N. S. Hejazi, and S. Dudoit, "Exploring high-dimensional biological data with sparse contrastive principal component analysis," *Bioinformatics*, vol. 36, no. 11, pp. 3422–3430, 2020.

[6] Ø. D. Trier, A. K. Jain, and T. Taxt, "Feature extraction methods for character recognition-a survey," *Pattern recognition*, vol. 29, no. 4, pp. 641–662, 1996.

[7] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and intelligent laboratory systems*, vol. 2, no. 1-3, pp. 37–52, 1987.

[8] A. Hyvärinen and E. Oja, "Independent component analysis: algorithms and applications," *Neural networks*, vol. 13, no. 4-5, pp. 411–430, 2000.

[9] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.

[10] M. Dash and H. Liu, "Feature selection for classification," *Intelligent data analysis*, vol. 1, no. 1-4, pp. 131–156, 1997.

[11] J. Hua, W. Tembe, and E. R. Dougherty, "Feature selection in the classification of high-dimension data," in *2008 IEEE international workshop on genomic signal processing and statistics*. IEEE, 2008, pp. 1–2.

[12] X. Jin, A. Xu, R. Bie, and P. Guo, "Machine learning techniques and chi-square feature selection for cancer classification using sage gene expression profiles," in *International workshop on data mining for biomedical applications*. Springer, 2006, pp. 106–115.

[13] C. Liao, S. Li, and Z. Luo, "Gene selection using wilcoxon rank sum test and support vector machine for cancer classification," in *International Conference on Computational and Information Science*. Springer, 2006, pp. 57–66.

[14] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1226–1238, 2005.

[15] J. Biesiada and W. Duch, "Feature selection for high-dimensional data—a pearson redundancy based filter," in *Computer recognition systems 2*. Springer, 2007, pp. 242–249.

[16] P. Mitra, C. Murthy, and S. K. Pal, "Unsupervised feature selection using feature similarity," *IEEE transactions on pattern analysis and machine intelligence*, vol. 24, no. 3, pp. 301–312, 2002.

[17] X. He, D. Cai, and P. Niyogi, "Laplacian score for feature selection," *Advances in neural information processing systems*, vol. 18, 2005.

[18] Z. Zhao and H. Liu, "Spectral feature selection for supervised and unsupervised learning," in *Proceedings of the 24th international conference on Machine learning*, 2007, pp. 1151–1157.

[19] D. Cai, C. Zhang, and X. He, "Unsupervised feature selection for multi-cluster data," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2010, pp. 333–342.

[20] E. Taskesen and M. J. Reinders, "2d representation of transcriptomes by t-sne exposes relatedness between human tissues," *PloS one*, vol. 11, no. 2, p. e0149853, 2016.

[21] K. Shekhar, S. W. Lapan, I. E. Whitney, N. M. Tran, E. Z. Macosko, M. Kowalczyk, X. Adiconis, J. Z. Levin, J. Nemesh, M. Goldman *et al.*, "Comprehensive classification of retinal bipolar neurons by single-cell transcriptomics," *Cell*, vol. 166, no. 5, pp. 1308–1323, 2016.

[22] L. McInnes, J. Healy, and J. Melville, "Umap: Uniform manifold approximation and projection for dimension reduction," *arXiv preprint arXiv:1802.03426*, 2018.

[23] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural computation*, vol. 15, no. 6, pp. 1373–1396, 2003.

[24] D. I. Spivak, "Metric realization of fuzzy simplicial sets," *Self published notes, available online at https://www. semanticscholar. org/paper/METRIC-REALIZATION-OF-FUZZY-SIMPLICIAL-SETS-Spivak/a73fb9d562a3850611d2615ac22c3a8687fa745e*, 2012.

[25] P. Indyk, R. Motwani, P. Raghavan, and S. Vempala, "Locality-preserving hashing in multidimensional spaces," in *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*. ACM, 1997, pp. 618–625.

[26] M. Bawa, T. Condie, and P. Ganesan, "Lsh forest: self-tuning indexes for similarity search," in *Proceedings of the 14th international conference on World Wide Web*. ACM, 2005, pp. 651–660.

[27] S. Kashef and H. Nezamabadi-pour, "A label-specific multi-label feature selection algorithm based on the pareto dominance concept," *Pattern Recognition*, vol. 88, pp. 654–667, 2019.

[28] J. González, J. Ortega, M. Damas, P. Martín-Smith, and J. Q. Gan, "A new multi-objective wrapper method for feature selection–accuracy and stability analysis for bci," *Neurocomputing*, vol. 333, pp. 407–418, 2019.

[29] J. Zhang, Z. Luo, C. Li, C. Zhou, and S. Li, "Manifold regularized discriminative feature selection for multi-label learning," *Pattern Recognition*, vol. 95, pp. 136–150, 2019.

[30] H. S. Badr, H. Du, M. Marshall, E. Dong, M. M. Squire, and L. M. Gardner, "Association between mobility patterns and covid-19 transmission in the usa: a mathematical modelling study," *The Lancet Infectious Diseases*, 2020.

[31] S.-W. Hwang, Y.-C. Chu, S.-R. Hwang, and J.-H. Hwang, "Association of periodic limb movements during sleep and tinnitus in humans," *Scientific Reports*, vol. 10, no. 1, pp. 1–5, 2020.

[32] J. Ircio, A. Lojo, U. Mori, and J. A. Lozano, "Mutual information based feature subset selection in multivariate time series classification," *Pattern Recognition*, vol. 108, p. 107525, 2020.

[33] C.-F. Tsai, W. Eberle, and C.-Y. Chu, "Genetic algorithms in feature and instance selection," *Knowledge-Based Systems*, vol. 39, pp. 240–247, 2013.

[34] S. Bandyopadhyay and U. Maulik, "Genetic clustering for automatic evolution of clusters and application to image classification," *Pattern recognition*, vol. 35, no. 6, pp. 1197–1208, 2002.

[35] S. Jiang, K.-S. Chin, L. Wang, G. Qu, and K. L. Tsui, "Modified genetic algorithm-based feature selection combined with pre-trained deep neural network for demand forecasting in outpatient department," *Expert systems with applications*, vol. 82, pp. 216–230, 2017.

[36] B. Peralta and A. Soto, "Embedded local feature selection within mixture of experts," *Information Sciences*, vol. 269, pp. 176–187, 2014.

[37] X. Zhang, G. Wu, Z. Dong, and C. Crawford, "Embedded feature-selection support vector machine for driving pattern recognition," *Journal of the Franklin Institute*, vol. 352, no. 2, pp. 669–685, 2015.

[38] S. Wang, J. Tang, and H. Liu, "Embedded unsupervised feature selection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 29, no. 1, 2015.

[39] H. Liu, M. Zhou, and Q. Liu, "An embedded feature selection method for imbalanced data classification," *IEEE/CAA Journal of Automatica Sinica*, vol. 6, no. 3, pp. 703–715, 2019.

[40] T. M. Cover, *Elements of information theory*. John Wiley & Sons, 1999.

[41] A. Rényi *et al.*, "On measures of entropy and information," in *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*. The Regents of the University of California, 1961.

[42] C. Tsallis, *Introduction to nonextensive statistical mechanics: approaching a complex world*. Springer Science & Business Media, 2009.

[43] K. Gajowniczek, T. Zabkowski, and A. Orłowski, "Comparison of decision trees with rényi and tsallis entropy applied for imbalanced churn dataset," in *2015 Federated Conference on Computer Science and Information Systems (FedCSIS)*. IEEE, 2015, pp. 39–44.

[44] A. Rajagopal, A. S. Nayak, A. U. Devi *et al.*, "From the quantum relative tsallis entropy to its conditional form: separability criterion beyond local and global spectra," *Physical Review A*, vol. 89, no. 1, p. 012331, 2014.

[45] R. Battiti, "Using mutual information for selecting features in supervised neural net learning," *IEEE Transactions on Neural Networks*, vol. 5, no. 4, pp. 537–550, 1994.

[46] F. Fleuret, "Fast binary feature selection with conditional mutual information," *Journal of Machine Learning Research*, vol. 5, no. Nov, pp. 1531–1555, 2004.

[47] M. Bennasar, Y. Hicks, and R. Setchi, "Feature selection using joint mutual information maximisation," *Expert Systems with Applications*, vol. 42, no. 22, pp. 8520–8532, 2015.

[48] A. Tanay and A. Regev, "Scaling single-cell genomics from phenomenology to mechanism," *Nature*, vol. 541, no. 7637, pp. 331–338, 2017.

[49] Y. Saeys, I. Inza, and P. Larranaga, "A review of feature selection techniques in bioinformatics," *bioinformatics*, vol. 23, no. 19, pp. 2507–2517, 2007.

[50] S. Picelli, Å. K. Björklund, O. R. Faridani, S. Sagasser, G. Winberg, and R. Sandberg, "Smart-seq2 for sensitive full-length transcriptome profiling in single cells," *Nature methods*, vol. 10, no. 11, pp. 1096–1098, 2013.

[51] P. Brennecke, S. Anders, J. K. Kim, A. A. Kołodziejczyk, X. Zhang, V. Proserpio, B. Baying, V. Benes, S. A. Teichmann, J. C. Marioni *et al.*, "Accounting for technical noise in single-cell rna-seq experiments," *Nature methods*, vol. 10, no. 11, p. 1093, 2013.

[52] A. Duò, M. D. Robinson, and C. Soneson, "A systematic performance evaluation of clustering methods for single-cell rna-seq data," *F1000Research*, vol. 7, 2018.

[53] C. Hafemeister and R. Satija, "Normalization and variance stabilization of single-cell rna-seq data using regularized negative binomial regression," *Genome biology*, vol. 20, no. 1, pp. 1–15, 2019.

[54] S. H. Yip, P. Wang, J.-P. A. Kocher, P. C. Sham, and J. Wang, "Linnorm: improved statistical analysis for single cell rna-seq expression data," *Nucleic acids research*, vol. 45, no. 22, pp. e179–e179, 2017.

[55] C. A. Vallejos, J. C. Marioni, and S. Richardson, "Basics: Bayesian analysis of single-cell sequencing data," *PLoS Comput Biol*, vol. 11, no. 6, p. e1004333, 2015.

[56] X. Qiu, A. Hill, J. Packer, D. Lin, Y.-A. Ma, and C. Trapnell, "Single-cell mrna quantification and differential analysis with census," *Nature methods*, vol. 14, no. 3, p. 309, 2017.

[57] Y. Hao, S. Hao, E. Andersen-Nissen, W. M. M. III, S. Zheng, A. Butler, M. J. Lee, A. J. Wilk, C. Darby, M. Zagar, P. Hoffman, M. Stoeckius, E. Papalexi, E. P. Mimitou, J. Jain, A. Srivastava, T. Stuart, L. B. Fleming, B. Yeung, A. J. Rogers, J. M. McElrath, C. A. Blish, R. Gottardo, P. Smibert, and R. Satija, "Integrated analysis of multimodal single-cell data," *Cell*, 2021. [Online]. Available: https://doi.org/10.1016/j.cell.2021.04.048

[58] G. X. Zheng, J. M. Terry, P. Belgrader, P. Ryvkin, Z. W. Bent, R. Wilson, S. B. Ziraldo, T. D. Wheeler, G. P. McDermott, J. Zhu *et al.*, "Massively parallel digital transcriptional profiling of single cells," *Nature communications*, vol. 8, 2017.

[59] R. Satija, J. A. Farrell, D. Gennert, A. F. Schier, and A. Regev, "Spatial reconstruction of single-cell gene expression data," *Nature biotechnology*, vol. 33, no. 5, pp. 495–502, 2015.

[60] H. Hotelling, "Analysis of a complex of statistical variables into principal components." *Journal of educational psychology*, vol. 24, no. 6, p. 417, 1933.

[61] A. T. Lun, D. J. McCarthy, and J. C. Marioni, "A step-by-step workflow for low-level analysis of single-cell rna-seq data with bioconductor," *F1000Research*, vol. 5, 2016.

[62] D. J. McCarthy, K. R. Campbell, A. T. Lun, and Q. F. Wills, "Scater: preprocessing, quality control, normalization and visualization of single-cell rna-seq data in r," *Bioinformatics*, vol. 33, no. 8, pp. 1179–1186, 2017.

[63] T. S. Andrews and M. Hemberg, "Identifying cell populations with scrnaseq," *Molecular aspects of medicine*, vol. 59, pp. 114–122, 2018.

[64] F. W. Townes, S. C. Hicks, M. J. Aryee, and R. A. Irizarry, "Feature selection and dimension reduction for single-cell rna-seq based on a multinomial model," *Genome biology*, vol. 20, no. 1, pp. 1–16, 2019.

[65] A. Butler, P. Hoffman, P. Smibert, E. Papalexi, and R. Satija, "Integrating single-cell transcriptomic data across different conditions, technologies, and species," *Nature biotechnology*, vol. 36, no. 5, pp. 411–420, 2018.

[66] A. Gribov, M. Sill, S. Lück, F. Rücker, K. Döhner, L. Bullinger, A. Benner, and A. Unwin, "Seurat: visual analytics for the integrated analysis of microarray data," *BMC medical genomics*, vol. 3, no. 1, p. 21, 2010.

[67] D. Lähnemann, J. Köster, E. Szczurek, D. J. McCarthy, S. C. Hicks, M. D. Robinson, C. A. Vallejos, K. R. Campbell, N. Beerenwinkel, A. Mahfouz *et al.*, "Eleven grand challenges in single-cell data science," *Genome biology*, vol. 21, no. 1, pp. 1–35, 2020.

[68] T. Abdelaal, L. Michielsen, D. Cats, D. Hoogduin, H. Mei, M. J. Reinders, and A. Mahfouz, "A comparison of automatic cell identification methods for single-cell RNA-sequencing data," *bioRxiv*, p. 644435, 2019.

[69] H. A. Pliner, J. Shendure, and C. Trapnell, "Supervised classification enables rapid annotation of cell atlases," *BioRxiv*, p. 538652, 2019.

[70] R. Wegmann, M. Neri, S. Schuierer, Bilican *et al.*, "Cellsius provides sensitive and specific detection of rare cell populations from complex single-cell rna-seq data," *Genome biology*, vol. 20, no. 1, pp. 1–21, 2019.

[71] A. Jindal, P. Gupta, D. Sengupta *et al.*, "Discovery of rare cells from voluminous single cell expression data," *Nature communications*, vol. 9, no. 1, pp. 1–9, 2018.

[72] M. Marouf, P. Machart, S. Bansal, Bonn *et al.*, "Realistic in silico generation and augmentation of single-cell rna-seq data using generative adversarial networks," *Nature communications*, vol. 11, no. 1, pp. 1–12, 2020.

[73] Y. Hao, S. Hao, E. Andersen-Nissen, W. M. Mauck III, S. Zheng, A. Butler, M. J. Lee, A. J. Wilk, C. Darby, M. Zager *et al.*, "Integrated analysis of multimodal single-cell data," *Cell*, 2021.

[74] F. A. Wolf, P. Angerer, and F. J. Theis, "Scanpy: large-scale single-cell gene expression data analysis," *Genome biology*, vol. 19, no. 1, p. 15, 2018.

[75] A. Aevermann and D. Zhang, "A machine learning method for the discovery of minimum marker gene combinations for cell-type identification from single-cell rna sequencing," *Genome research*, pp. gr–275 569, 2021.

[76] J.-M. Kim, Y.-S. Jung, E. A. Sungur, K.-H. Han, C. Park, and I. Sohn, "A copula method for modeling directional dependence of genes," *BMC bioinformatics*, vol. 9, no. 1, p. 225, 2008.

[77] N. I. Fisher, "Copulas," *Encyclopedia of statistical sciences*, 1997.

[78] I. Olkin, "Continuous bivariate distributions, emphasising applications," *Journal of the American Statistical Association*, vol. 86, no. 416, pp. 1143–1145, 1991.

[79] H. Joe, *Multivariate models and multivariate dependence concepts*. CRC Press, 1997.

[80] R. B. Nelsen, *An introduction to copulas*. Springer Science & Business Media, 2007.

[81] R. Nelsen, J. Quesada-Molina, and J. Rodríguez-Lallena, "Bivariate copulas with cubic sections," *Journal of Nonparametric Statistics*, vol. 7, no. 3, pp. 205–220, 1997.

[82] J. Dedecker, P. Doukhan, G. Lang, and R. León, "S. and prieur, c.(2007). weak dependence: with examples and applications," *Lecture Notes in Statistics*, vol. 190.

[83] W. H. Kruskal, "Ordinal measures of association," *Journal of the American Statistical Association*, vol. 53, no. 284, pp. 814–861, 1958.

[84] P. Embrechts, "Copulas: A personal view," *Journal of Risk and Insurance*, vol. 76, no. 3, pp. 639–650, 2009.

[85] F. Durante and C. Sempi, "Copula theory: an introduction," in *Copula theory and its applications*. Springer, 2010, pp. 3–31.

[86] F. Schmid, R. Schmidt, T. Blumentritt, S. Gaißer, and M. Ruppert, "Copula-based measures of multivariate association," in *Copula theory and its applications*. Springer, 2010, pp. 209–236.

[87] P. Busababodhin and P. Amphanthong, "Copula modelling for multivariate statistical process control: a review," *Communications for Statistical Applications and Methods*, vol. 23, no. 6, pp. 497–515, 2016.

[88] A. Dolati and A. D. Nezhad, "Some results on convexity and concavity of multivariate copulas," *Iranian Journal of Mathematical Sciences and Informatics*, vol. 9, no. 2, pp. 87–100, 2014.

[89] E. W. Frees and E. A. Valdez, "Understanding relationships using copulas," *North American actuarial journal*, vol. 2, no. 1, pp. 1–25, 1998.

[90] A. J. Patton, "Copula–based models for financial time series," in *Handbook of financial time series*. Springer, 2009, pp. 767–785.

[91] O. Sokolinskiy and D. van Dijk, "Forecasting volatility with copula-based time series models," Tinbergen Institute Discussion Paper, Tech. Rep., 2011.

[92] E. Z. Macosko, A. Basu, R. Satija, J. Nemesh, K. Shekhar, M. Goldman, I. Tirosh, A. R. Bialas, N. Kamitaki, E. M. Martersteck *et al.*, "Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets," *Cell*, vol. 161, no. 5, pp. 1202–1214, 2015.

[93] A. Wagner, A. Regev, and N. Yosef, "Revealing the vectors of cellular identity with single-cell genomics," *Nature biotechnology*, vol. 34, no. 11, pp. 1145–1160, 2016.

[94] A. K. Shalek, R. Satija, X. Adiconis, R. S. Gertner, J. T. Gaublomme, R. Raychowdhury, S. Schwartz, N. Yosef, C. Malboeuf, D. Lu *et al.*, "Single-cell transcriptomics reveals bimodality in expression and splicing in immune cells," *Nature*, vol. 498, no. 7453, pp. 236–240, 2013.

[95] D. Grün, A. Lyubimova, L. Kester, K. Wiebrands, O. Basak, N. Sasaki, H. Clevers, and A. van Oudenaarden, "Single-cell messenger rna sequencing reveals rare intestinal cell types," *Nature*, vol. 525, no. 7568, pp. 251–255, 2015.

[96] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[97] P.-G. Martinsson, V. Rokhlin, and M. Tygert, "A randomized algorithm for the decomposition of matrices," *Applied and Computational Harmonic Analysis*, vol. 30, no. 1, pp. 47–68, 2011.

[98] L. Pauleve, H. Jegou, and L. Amsaleg, "Locality sensitive hashing: A comparison of hash function types and querying mechanisms," *Pattern Recognition Letters*, vol. 31, no. 11, pp. 1348–1358, 2010.

[99] J. Leskovec, A. Rajaraman, and J. D. Ullman, *Mining of massive datasets*. Cambridge University Press, 2014.

[100] S. Karamizadeh, S. M. Abdullah, A. A. Manaf, M. Zamani, and A. Hooman, "An overview of principal component analysis," *Journal of Signal and Information Processing*, vol. 4, no. 03, p. 173, 2013.

[101] C. Li, Y. Diao, H. Ma, and Y. Li, "A statistical pca method for face recognition," in *Intelligent Information Technology Application, 2008. IITA'08. Second International Symposium on*, vol. 3. IEEE, 2008, pp. 376–380.

[102] I. T. Jolliffe, "Principal component analysis and factor analysis," in *Principal component analysis*. Springer, 1986, pp. 115–128.

[103] Collado-Torres, Leonardo, Nellore, Abhinav, Kammers, Kai, Ellis, S. E, Taub, M. A, Hansen, K. D, Jaffe, A. E, Langmead, Ben, Leek, and J. T, "Reproducible rna-seq analysis using recount2," *Nature Biotechnology*, vol. 35, no. 4, pp. 319–321, 2017.

[104] Q. Deng, D. Ramsköld, B. Reinius, and R. Sandberg, "Single-cell rna-seq reveals dynamic, random monoallelic gene expression in mammalian cells," *Science*, vol. 343, no. 6167, pp. 193–196, 2014.

[105] G. X. Zheng, J. M. Terry, P. Belgrader, P. Ryvkin, Z. W. Bent, R. Wilson, S. B. Ziraldo, T. D. Wheeler, G. P. McDermott, J. Zhu *et al.*, "Massively parallel digital transcriptional profiling of single cells," *bioRxiv*, p. 065912, 2016.

[106] H. S. Bhat and N. Kumar, "On the derivation of the bayesian information criterion," *School of Natural Sciences, University of California*, 2010.

[107] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of computational and applied mathematics*, vol. 20, pp. 53–65, 1987.

[108] K. Y. Yeung and W. L. Ruzzo, "Details of the adjusted rand index and clustering algorithms, supplement to the paper an empirical study on principal component analysis for clustering gene expression data," *Bioinformatics*, vol. 17, no. 9, pp. 763–774, 2001.

[109] S. Dash, S. K. Shakyawar, M. Sharma, and S. Kaushik, "Big data in healthcare: management, analysis and future prospects," *Journal of Big Data*, vol. 6, no. 1, p. 54, 2019.

[110] S. Sharmin, M. Shoyaib, A. A. Ali, M. A. H. Khan, and O. Chae, "Simultaneous feature selection and discretization based on mutual information," *Pattern Recognition*, vol. 91, pp. 162–174, 2019.

[111] R. Sheikhpour, M. A. Sarram, S. Gharaghani, and M. A. Z. Chahooki, "A survey on semi-supervised feature selection methods," *Pattern Recognition*, vol. 64, pp. 141–158, 2017.

[112] I. A. Gheyas and L. S. Smith, "Feature subset selection in large dimensionality domains," *Pattern Recognition*, vol. 43, no. 1, pp. 5–13, 2010.

[113] H. Liu, J. Sun, L. Liu, and H. Zhang, "Feature selection with dynamic mutual information," *Pattern Recognition*, vol. 42, no. 7, pp. 1330–1339, 2009.

[114] R. Shang, Y. Meng, W. Wang, F. Shang, and L. Jiao, "Local discriminative based sparse subspace learning for feature selection," *Pattern Recognition*, vol. 92, pp. 219–230, 2019.

[115] Y. Ye, Q. Wu, J. Z. Huang, M. K. Ng, and X. Li, "Stratified sampling for feature subspace selection in random forests for high dimensional data," *Pattern Recognition*, vol. 46, no. 3, pp. 769–787, 2013.

[116] X.-f. Song, Y. Zhang, Y.-n. Guo, X.-y. Sun, and Y.-l. Wang, "Variable-size cooperative coevolutionary particle swarm optimization for feature selection on high-dimensional data," *IEEE Transactions on Evolutionary Computation*, 2020.

[117] S.-L. Huang, X. Xu, and L. Zheng, "An information-theoretic approach to unsupervised feature selection for high-dimensional data," *IEEE Journal on Selected Areas in Information Theory*, 2020.

[118] M. Reif and F. Shafait, "Efficient feature size reduction via predictive forward selection," *Pattern Recognition*, vol. 47, no. 4, pp. 1664–1673, 2014.

[119] R. Kohavi and D. Sommerfield, "Feature subset selection using the wrapper method: Overfitting and dynamic search space topology." in *KDD*, 1995, pp. 192–197.

[120] Y. Kim, S. Lee, M.-S. Kwon, A. Na, Y. Choi, S. G. Yi, J. Namkung, S. Han, M. Kang, S. W. Kim *et al.*, "Developing cancer prediction model based on stepwise selection by auc measure for proteomics data," in *2015 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 2015, pp. 1345–1350.

[121] M. S. Rahman, M. K. Rahman, M. Kaykobad, and M. S. Rahman, "isgpt: An optimized model to identify sub-golgi protein types using svm and random forest based feature selection," *Artificial intelligence in medicine*, vol. 84, pp. 90–100, 2018.

[122] M. Yamada, W. Jitkrittum, L. Sigal, E. P. Xing, and M. Sugiyama, "High-dimensional feature selection by feature-wise kernelized lasso," *Neural computation*, vol. 26, no. 1, pp. 185–207, 2014.

[123] S. Paul and P. Drineas, "Feature selection for ridge regression with provable guarantees," *Neural computation*, vol. 28, no. 4, pp. 716–742, 2016.

[124] A. Milan, S. H. Rezatofighi, R. Garg, A. R. Dick, and I. D. Reid, "Data-driven approximations to np-hard problems." in *AAAI*, 2017, pp. 1453–1459.

[125] D. Mo and Z. Lai, "Robust jointly sparse regression with generalized orthogonal learning for image feature selection," *Pattern Recognition*, vol. 93, pp. 164–178, 2019.

[126] P. E. Meyer, "Information-theoretic variable selection and network inference from microarray data," *Ph. D. Thesis. Université Libre de Bruxelles*, 2008.

[127] R. B. Nelsen, "Properties and applications of copulas: A brief survey," in *Proceedings of the First Brazilian Conference on Statistical Modeling in Insurance and Finance,(Dhaene, J., Kolev, N., Morettin, PA (Eds.)), University Press USP: Sao Paulo*, 2003, pp. 10–28.

[128] E. P. Xing, M. I. Jordan, R. M. Karp *et al.*, "Feature selection for high-dimensional genomic microarray data," in *ICML*, vol. 1, 2001, pp. 601–608.

[129] D. Pál, B. Póczos, and C. Szepesvári, "Estimation of rényi entropy and mutual information based on generalized nearest-neighbor graphs," in *Advances in Neural Information Processing Systems*, 2010, pp. 1849–1857.

[130] G. Brown, A. Pocock, M.-J. Zhao, and M. Luján, "Conditional likelihood maximisation: a unifying framework for information theoretic feature selection," *Journal of machine learning research*, vol. 13, no. Jan, pp. 27–66, 2012.

[131] V. Y. Kiselev, K. Kirschner, M. T. Schaub, T. Andrews, A. Yiu, T. Chandra, K. N. Natarajan, W. Reik, M. Barahona, A. R. Green *et al.*, "Sc3: consensus clustering of single-cell rna-seq data," *Nature methods*, vol. 14, no. 5, p. 483, 2017.

[132] V. A. Traag, L. Waltman, and N. J. van Eck, "From louvain to leiden: guaranteeing well-connected communities," *Scientific reports*, vol. 9, no. 1, pp. 1–12, 2019.

[133] L. Jiang, H. Chen, L. Pinello, and G.-C. Yuan, "Giniclust: detecting rare cell types from single-cell gene expression data with gini index," *Genome biology*, vol. 17, no. 1, p. 144, 2016.

[134] D. Grün, L. Kester, and A. Van Oudenaarden, "Validation of noise models for single-cell transcriptomics," *Nature methods*, vol. 11, no. 6, p. 637, 2014.

[135] A. S. Britto Jr, R. Sabourin, and L. E. Oliveira, "Dynamic selection of classifiers—a comprehensive review," *Pattern Recognition*, vol. 47, no. 11, pp. 3665–3680, 2014.

[136] T. Stuart, A. Butler, P. Hoffman, C. Hafemeister, E. Papalexi, W. M. Mauck III, Y. Hao, M. Stoeckius, P. Smibert, and R. Satija, "Comprehensive integration of single-cell data," *Cell*, vol. 177, no. 7, pp. 1888–1902, 2019.

[137] W. Liu and W. Lin, "Additive white gaussian noise level estimation in svd domain for images," *IEEE Transactions on Image processing*, vol. 22, no. 3, pp. 872–883, 2012.

[138] C.-F. Tsai and Y.-C. Chen, "The optimal combination of feature selection and data discretization: An empirical study," *Information Sciences*, vol. 505, pp. 282–293, 2019.

[139] A. A. Alizadeh, M. B. Eisen, R. E. Davis, C. Ma, I. S. Lossos, A. Rosenwald, J. C. Boldrick, H. Sabet, T. Tran, X. Yu *et al.*, "Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling," *Nature*, vol. 403, no. 6769, p. 503, 2000.

[140] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri *et al.*, "Molecular classification of cancer: class discovery and class prediction by gene expression monitoring," *Science*, vol. 286, no. 5439, pp. 531–537, 1999.

[141] T. Sim, S. Baker, and M. Bsat, "The cmu pose, illumination, and expression (pie) database," in *Proceedings of Fifth IEEE International Conference on Automatic Face Gesture Recognition*. IEEE, 2002, pp. 53–58.

[142] J. J. Hull, "A database for handwritten text recognition research," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 5, pp. 550–554, 1994.

[143] L. Yan, M. Yang, H. Guo, L. Yang, J. Wu, R. Li, P. Liu, Y. Lian, X. Zheng, J. Yan *et al.*, "Single-cell rna-seq profiling of human preimplantation embryos and embryonic stem cells," *Nature structural & molecular biology*, vol. 20, no. 9, p. 1131, 2013.

[144] A. A. Pollen, T. J. Nowakowski, J. Shuga, X. Wang, A. A. Leyrat, J. H. Lui, N. Li, L. Szpankowski, B. Fowler, P. Chen *et al.*, "Low-coverage single-cell mrna sequencing reveals cellular heterogeneity and activated signaling pathways in developing cerebral cortex," *Nature biotechnology*, vol. 32, no. 10, p. 1053, 2014.

[145] M. J. Muraro, G. Dharmadhikari, D. Grün, N. Groen, T. Dielen, E. Jansen, L. van Gurp, M. A. Engelse, F. Carlotti, E. J. de Koning *et al.*, "A single-cell transcriptome atlas of the human pancreas," *Cell systems*, vol. 3, no. 4, pp. 385–394, 2016.

[146] M. Hoffman, D. Steinley, and M. J. Brusco, "A note on using the adjusted rand index for link prediction in networks," *Social networks*, vol. 42, pp. 72–79, 2015.

[147] P. E. Meyer and G. Bontempi, "On the use of variable complementarity for feature selection in cancer classification," in *Workshops on Applications of Evolutionary Computation*. Springer, 2006, pp. 91–102.

[148] A. Ralston and K. Shaw, "Gene expression regulates cell differentiation," *Nature education*, vol. 1, no. 1, p. 127, 2008.

[149] Y. Yang, L. Han, Y. Yuan, J. Li, N. Hei, and H. Liang, "Gene co-expression network analysis reveals common system-level properties of prognostic genes across cancer types," *Nature communications*, vol. 5, p. 3231, 2014.

[150] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein, "Cluster analysis and display of genome-wide expression patterns," *Proceedings of the National Academy of Sciences*, vol. 95, no. 25, pp. 14 863–14 868, 1998.

[151] T. Ideker and N. Krogan, "Differential network biology," *Molecular Systems Biology*, vol. 565, no. 8, 2011.

[152] S. Ray and S. Bandyopadhyay, "Discovering condition specific topological pattern changes in coexpression network: an application to hiv-1 progression," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 11, no. 4, Dec 2015.

[153] S. Cho, J. Kim, and J. Kim, "Identifying set-wise differential co-expression in gene expression microarray data." *BMC Bioinformatics*, vol. 10, no. 109, 2009.

[154] D. Kostka and R. Spang, "Finding disease specific alterations in the co-expression of genes," *Bioinformatics*, vol. 20, no. 1, pp. i194–i199, March 2004.

[155] Y. Lai, B. Wu, L. Chen, and H. Zhao, "A statistical method for identifying differential gene–gene co-expression patterns," *Bioinformatics*, vol. 20, no. 17, pp. 3146–3155, 2004.

[156] D. Kostka and R. S. R, "Finding disease specific alterations in the co-expression of genes." *Bioinformatics*, vol. 20, no. Sup 1, pp. i194–199., 2005.

[157] M. Watson, "Coxpress: differential co-expression in gene expression data," *BMC Bioinformatics*, vol. 7, no. 509, 2006.

[158] B. Tesson, R. Breitling, and R. Jansen, "Diffcoex: a simple and sensitive method to find differentially coexpressed gene modules." *BMC Bioinformatics*, vol. 11, no. 497, 2010.

[159] D. Amar, H. Safer, and R. Shamir, "Dissection of regulatory networks that are altered in disease via differential co-expression," *Plos Comp Bio.*, vol. 9, no. 3, p. e1002955, 2013.

[160] S. Ray and U. Maulik, "Identifying differentially coexpressed module during hiv disease progression: A multiobjective approach," *Scientific reports*, vol. 7, no. 1, p. 86, 2017.

[161] M. Plass, J. Solana, F. A. Wolf, S. Ayoub, A. Misios, P. Glažar, B. Obermayer, F. J. Theis, C. Kocks, and N. Rajewsky, "Cell type atlas and lineage tree of a whole complex animal by single-cell transcriptomics," *Science*, vol. 360, no. 6391, 2018.

[162] C. T. Fincher, O. Wurtzel, T. de Hoog, K. M. Kravarik, and P. W. Reddien, "Cell type transcriptome atlas for the planarian schmidtea mediterranea," *Science*, vol. 360, no. 6391, 2018.

[163] S. Ray and A. Schonhuth, "Markercapsule: Explainable single cell typing using capsule networks," *bioRxiv*, 2020.

[164] P. Jaworski, F. Durante, W. K. Hardle, and T. Rychlik, *Copula theory and its applications*. Springer, 2010, vol. 198.

[165] D. Gunawan, M.-N. Tran, K. Suzuki, J. Dick, and R. Kohn, "Computationally efficient bayesian estimation of high-dimensional archimedean copulas with discrete and mixed margins," *Statistics and Computing*, vol. 29, no. 5, pp. 933–946, 2019.

[166] R. B. Nelsen, "Introduction," in *An Introduction to Copulas*. Springer, 1999, pp. 1–4.

[167] A. Andoni, I. Razenshteyn, and N. S. Nosatzki, "Lsh forest: Practical algorithms made theoretical," in *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 2017, pp. 67–78.

[168] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[169] D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams, "Convolutional networks on graphs for learning molecular fingerprints," in *Advances in neural information processing systems*, 2015, pp. 2224–2232.

[170] M. Baron, A. Veres, S. L. Wolock, A. L. Faust, R. Gaujoux, A. Vetere, J. H. Ryu, B. K. Wagner, S. S. Shen-Orr, A. M. Klein *et al.*, "A single-cell transcriptomic map of the human and mouse pancreas reveals inter-and intra-cell population structure," *Cell systems*, vol. 3, no. 4, pp. 346–360, 2016.

[171] A. M. Klein, L. Mazutis, I. Akartuna, N. Tallapragada, A. Veres, V. Li, L. Peshkin, D. A. Weitz, and M. W. Kirschner, "Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells," *Cell*, vol. 161, no. 5, pp. 1187–1201, 2015.

[172] I. Tirosh, B. Izar, S. M. Prakadan, M. H. Wadsworth, D. Treacy, J. J. Trombetta, A. Rotem, C. Rodman, C. Lian, G. Murphy *et al.*, "Dissecting the multicellular ecosystem of metastatic melanoma by single-cell rna-seq," *Science*, vol. 352, no. 6282, pp. 189–196, 2016.

[173] F. A. Wolf, F. K. Hamey, M. Plass, J. Solana, J. S. Dahlin, B. Göttgens, N. Rajewsky, L. Simon, and F. J. Theis, "Paga: graph abstraction reconciles clustering with trajectory inference through a topology preserving map of single cells," *Genome biology*, vol. 20, no. 1, pp. 1–9, 2019.

[174] M. Jacomy, T. Venturini, S. Heymann, and M. Bastian, "Forceatlas2, a continuous graph layout algorithm for handy network visualization designed for the gephi software," *PloS one*, vol. 9, no. 6, p. e98679, 2014.

[175] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *kdd*, vol. 96, no. 34, 1996, pp. 226–231.

[176] B. Dumitrascu, S. Villar, D. G. Mixon, and B. E. Engelhardt, "Optimal marker gene selection for cell type discrimination in single cell analyses," *Nature communications*, vol. 12, no. 1, pp. 1–8, 2021.

[177] T. P. Hettmansperger and J. W. McKean, *Robust nonparametric statistical methods*. CRC Press, 2010.

[178] X. Zhang, Y. Lan, J. Xu, F. Quan, E. Zhao, C. Deng, T. Luo, L. Xu, G. Liao, M. Yan *et al.*, "Cellmarker: a manually curated resource of cell markers in human and mouse," *Nucleic acids research*, vol. 47, no. D1, pp. D721–D728, 2019.

[179] V. Svensson, R. Vento-Tormo, and S. A. Teichmann, "Exponential scaling of single-cell rna-seq in the past decade," *Nature protocols*, vol. 13, no. 4, pp. 599–604, 2018.

[180] J. Lin, "Divergence measures based on the shannon entropy," *IEEE Transactions on Information theory*, vol. 37, no. 1, pp. 145–151, 1991.

[181] P. Lin, M. Troup, and J. W. Ho, "Cidr: Ultrafast and accurate clustering through imputation for single-cell rna-seq data," *Genome biology*, vol. 18, no. 1, pp. 1–11, 2017.

[182] A. Teixeira, A. Matos, and L. Antunes, "Conditional rényi entropies," *IEEE Transactions on Information Theory*, vol. 58, no. 7, pp. 4273–4277, 2012.

[183] V. M. Ilić, I. B. Djordjević, and M. Stanković, "On a general definition of conditional rényi entropies," in *Multidisciplinary Digital Publishing Institute Proceedings*, vol. 2, no. 4, 2017, p. 166.

[184] T. Villmann and T. Geweniger, "Multi-class and cluster evaluation measures based on renyi and tsallis entropies and mutual information," in *International Conference on Artificial Intelligence and Soft Computing*. Springer, 2018, pp. 736–749.

[185] S. Arimoto, "Information measures and capacity of order α for discrete memoryless channels," *Topics in information theory*, 1977.

[186] M. Iwamoto and J. Shikata, "Revisiting conditional rényi entropies and generalizing shannons bounds in information theoretically secure encryption," *Technical report, Cryptology ePrint Archive 440/2013*, 2013.

[187] S. Abe, "Geometry of escort distributions," *Physical Review E*, vol. 68, no. 3, p. 031101, 2003.

[188] A. Ghosh and A. Basu, "A scale-invariant generalization of the rényi entropy, associated divergences and their optimizations under tsallis' nonextensive framework," *IEEE Transactions on Information Theory*, vol. 67, no. 4, pp. 2141–2161, 2021.

[189] M. A. Kumar and I. Sason, "Projection theorems for the rényi divergence on \alpha −*convexsets*," *IEEE Transactions on Information Theory*, vol. 62, no. 9, pp. 4924–4935, 2016.

[190] S. Couch, Z. Kazan, K. Shi, A. Bray, and A. Groce, "Differentially private nonparametric hypothesis testing," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019, pp. 737–751.

[191] C. A. Vallejos, D. Risso, A. Scialdone, S. Dudoit, and J. C. Marioni, "Normalizing single-cell rna sequencing data: challenges and opportunities," *Nature methods*, vol. 14, no. 6, p. 565, 2017.

[192] S. Liao, Q. Gao, F. Nie, Y. Liu, and X. Zhang, "Worst-case discriminative feature selection." in *IJCAI*, 2019, pp. 2973–2979.

[193] M. D. Luecken and F. J. Theis, "Current best practices in single-cell rna-seq analysis: a tutorial," *Molecular systems biology*, vol. 15, no. 6, p. e8746, 2019.

[194] Z. Ji and H. Ji, "Tscan: Pseudo-time reconstruction and evaluation in single-cell rna-seq analysis," *Nucleic acids research*, vol. 44, no. 13, pp. e117–e117, 2016.

[195] G. Chen, B. Ning, and T. Shi, "Single-cell rna-seq technologies and related computational data analysis," *Frontiers in genetics*, vol. 10, p. 317, 2019.

[196] E. Vans, A. Patil, and A. Sharma, "Feats: Feature selection based clustering of single-cell rna-seq data," *bioRxiv*, 2020.

[197] S. Lall, D. Sinha, S. Bandyopadhyay, and D. Sengupta, "Structure-aware principal component analysis for single-cell rna-seq data," *Journal of Computational Biology*, 2018.

[198] S. Lall, S. Ray, and S. Bandyopadhyay, "Rgcop-a regularized copula based method for gene selection in single cell rna-seq data," *bioRxiv*, 2020.

[199] O. Lindenbaum, J. Stanley, G. Wolf, and S. Krishnaswamy, "Geometry based data generation," in *Advances in Neural Information Processing*

Systems, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., vol. 31. Curran Associates, Inc., 2018. [Online]. Available: https://proceedings.neurips.cc/paper/2018/file/c8ed21db4f678f3b13b9d5ee16489088-Paper.pdf

[200] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[201] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 70. PMLR, 2017, pp. 214–223.

[202] S. Nowozin, B. Cseke, and R. Tomioka, "f-gan: Training generative neural samplers using variational divergence minimization," *arXiv preprint arXiv:1606.00709*, 2016.

[203] X.-L. Mao, B.-S. Feng, Y.-J. Hao, L. Nie, H. Huang, and G. Wen, "S2jsd-lsh: A locality-sensitive hashing schema for probability distributions," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, no. 1, 2017.

[204] T. S. Andrews and M. Hemberg, "M3drop: dropout-based feature selection for scrnaseq," *Bioinformatics*, vol. 35, no. 16, pp. 2865–2867, 2019.

[205] S. Darmanis, S. A. Sloan, Y. Zhang, M. Enge, C. Caneda, L. M. Shuer, M. G. H. Gephart, B. A. Barres, and S. R. Quake, "A survey of human brain transcriptome diversity at the single cell level," *Proceedings of the National Academy of Sciences*, vol. 112, no. 23, pp. 7285–7290, 2015.