Doctoral Thesis

Reproducible Classification and Analysis for Single Cell, rCASC

Author : Alessandrì Luca

Tutor: Prof. Raffaele A. Calogero

5 October 2020

Ph.D. Program in Complex Systems for Life Sciences

Dottorato in Sistemi Complessi per le Scienze della Vita

UNIVERSITY OF TURIN

XXXIII cycle

Contents

Abstract **v**

1 Introduction **1**

1.1 First generation sequencing 1

1.1.1 Short read sequencing 1

1.2 Single cell Technologies 2

1.2.1 Single cell RNA sequencing technologies 3

1.2.1.1 FACS 3

1.2.1.2 Fluidigm C1 4

1.2.1.3 Drop-seq and Indrop-seq 4

1.2.1.4 Microwell 5

1.2.1.5 Split-seq 5

1.2.1.6 Spatial transcriptomics 6

1.2.2 Single Cell ATAC-seq 6

1.2.3 Single Cell RNA-seq preprocessing 8

1.2.3.1 Data preprocessing, filtering and normalization 8

1.2.4 scRNAseq data Clustering 10

1.2.5 Hierarchical clustering algorithm 11

1.2.5.1 Griph 11

1.2.5.2 Seurat 12

1.2.5.3 Scanpy 12

1.2.5.4 SHARP 12

1.3 Non hierarchical clustering algorithm 14
1.3.0.1 t-Sne + K-Means 14
1.3.0.2 SIMLR 14
1.4 Feature selection 16
1.5 Aim of the Thesis 17
2 Results 18
2.1 rCASC Workflow 18
2.2 Quality control and Preprocessing 19
2.2.1 Variance and expression Filtering 22
2.2.2 Data Normalization 23
2.2.3 Detecting and removing cell cycle bias 25
2.3 Subpopulation discovery 27
2.3.1 Defining the optimal number of clusters 28
2.3.2 Cluster stability 29
2.3.3 Henning Cluster stability score 29
2.3.4 Cell Stability Score (CSS) 30
2.4 Spatial Transcriptomics Analysis 35
2.5 Features Selection 36
2.5.1 SIMLR/Seurat and differential expression for feature selection . . 36
2.5.2 COMET 38
2.6 Neural Network (NN) for Single cell data mining 40
2.6.1 Autoencoders 40
2.6.2 Sparse Autoencoders 42
2.6.3 Validation of SCA analysis on a PBMC derived dataset (setA) . . 45
2.6.4 Investigating the effect of normalization on SCA latent space frequency matrix on QCC/QCM scores- SetA 48
2.6.5 Validation of variational Sparsely Connected Autoencoders (vSCA) analysis on a PBMC derived dataset (setA) 50

2.6.6 Validation of Sparse Sparsely Connected Autoencoders (SSCA) analysis on a PBMC derived dataset (setA) 51
2.6.7 Application of SCA analysis on spatially resolved transcriptomics of breast cancer histological section. 53
2.7 GUI and cloud implementation 57
2.7.1 Galaxy, Laniakea implementation 57
2.7.2 Single cell ATAC-seq 58
2.7.2.1 Prediction algorithm 59

3 Conclusions 62

4 Materials and Methods 64

4.1 Availability and Requirements 64
4.2 Datasets 64
4.3 Counts table generation 67
4.4 Counts table exploration and manipulation 68
4.5 Clustering 69
4.6 Feature selection 70
4.7 Scalability 70
4.8 Autoencoder model coding and hyperparameter selection 70
4.9 SCA and SSCA latent space definition 71
4.10 QCM and QCC metrics 71
4.11 COMET analysis 72
4.12 SCA handling functions in rCASC 72
4.13 Jaccard Value 73
4.13.0.1 CSS Algorithm detailed 73
4.13.0.2 Example 75
4.14 Reproducibility 77
4.14.1 Docker implementation 78

5 Publications 80

6 Acknowledgments **81**

Abstract

Single-cell RNA sequencing is an essential tool to investigate cellular heterogeneity, and to highlight cell subpopulation specific signatures. Single-cell sequencing applications are now spreading from the most conventional RNAseq to epigenomics. Single-cell sequencing led to the development of a large variety of algorithms and tools. However, there are few computational workflows providing analysis flexibility and achieving at the same time functional (i.e. information about data and the utilized tools are saved in terms of meta-data) and computational reproducibility (i.e. real image of the computation environment used to generate the data is stored) through a user-friendly environment. rCASC is a modular workflow providing integrated analysis environment. It provides preprocessing tools to remove low quality cells and/or specific bias, e.g. cell cycle. Subpopulations discovery can be achieved using different clustering techniques based on different distance metrics. Quality of clusters is then estimated through a new metric namely Cell Stability Score (CSS), which describes the stability of a cell in a cluster as consequence of a perturbation induced by removing a random set of cells from the overall cells' population. Moreover, rCASC provides tools for the identification of clusters-specific gene-signature by mean of neural network analysis. In conclusion rCASC is a modular workflow embedding new features helping researchers in defining cells subpopulations and in detecting subpopulation specific markers. It exploits docker containerization to make easier its installation and to achieve a computation reproducible analysis. A Java GUI, is also provided to make friendly the use of the tool even for users without computational skills in R. rCASC is also available on the cloud on Laniakea, Galaxy platform.

1 Introduction

> People always have weird thoughts around camp fire.
>
> *Finn the Human- Adventure time*

1.1 First generation sequencing

In the middle of 70s the field of genetic research was radically modified by the possibility to sequence DNA fragments using Sanger sequence [1]. The development of Sanger sequencing method represented the first step to the definition of the Human genome sequence [2] [3]. However, at the beginning of the years 2000 sequencing was mainly limited to DNA/cDNA characterization

1.1.1 Short read sequencing

Second generation sequencing, also known as short read sequencing or Next Generation Sequencing (NGS), is a parallel sequencing approach, in which nucleic acids are fragmented and each fragment is sequenced multiple times. The sequencing accuracy for each individual nucleotide can be quite high, i.e. the error rate can be less than 1/10000 nucleotides. NGS extended the area of application of sequencing at the quantification of nucleic acid molecules, which was till 2010 mainly done by microarray technology [4]. Short read sequencing of transcriptomes, also known as RNA-Seq, provides both the sequence and frequency of RNA molecules that are present at any particular time in a specific cell type, tissue or organ.

1.2 Single cell Technologies

Single cell analysis provides information aiming to understand the functional differences existing between cells located in a tissue. Individual cells of the same phenotype are commonly viewed as identical functional units of a tissue or organ. However, short read sequencing of RNA from single cells suggests a more complex organization of heterogeneous cell states that together produce system-level functions. Since the analysis of heterogeneous cell population in bulk only provides an averaged view of the subpopulations, the information about small potentially relevant subpopulations might be lost by averaging population data. Sequencing of DNA and RNA from single cells can provide more comprehensive footprint of cellular functions. Single cell analysis focus on the understanding of differences that characterize single cells within a population of cells, which could be phenotypically identical. Nowadays, isolation and separation of single cells is still technically challenging. A large variety of technologies for Single-Cell separation, isolation and sorting are available and they can be chosen accordingly to the scientific objective of interest. Technologies currently in use for the isolation of Single Cells from tissues or liquid culture are :

- FACS/flow cytometry.
- Random seeding or liquid dilution into microplates.
- Manual cell picking.
- Microfluidics.
- Laser microdissection.
- Electric fields.
- Non-contact dispensing/printing.
- Optical tweezers.
- Capillary-based isolation.

1.2.1 Single cell RNA sequencing technologies

Single-Cell RNA sequencing (scRNA-seq)[5] provides the expression profile of individual cells. Through genes clustering analyses, cell subpopulations detection can be achieved, showing the identification of rare subpopulation of cells, which cannot be detected using conventional bulk RNAseq. scRNA-seq technology improved significantly in the last few years, both in terms of transcript quantitation and experimental throughput. Recent advances in microfluidic technologies have enabled high-throughput single cell profiling by which researchers can examine hundreds to thousands of cells per experiment in a cost-effective manner. Whereas low capture efficiency and high levels of technical noise limit the sensitivity and accuracy of scRNA-seq, sophisticated analytical frameworks are emerging to facilitate the interpretation of scRNA-seq data.

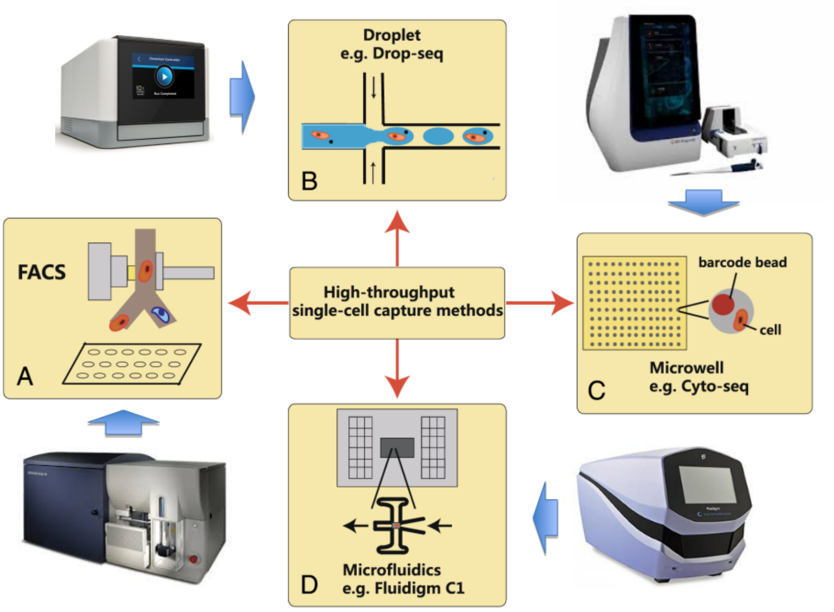


Figure 1.1: Single cell sequencing platforms

1.2.1.1 FACS

FACS, fig. 1.1A, was the first approach used to isolate and sequence single cells. It requires expensive equipment, expert technician and and throughput is limited to few hundred of cells.

1.2.1.2 Fluidigm C1

C1, fig. 1.1D, was the first instrument designed specifically to analyze single cell at RNA level. C1 system isolates single cells into individual reaction chambers in the Fluidigm integrated fluidic circuit (IFC), which is a complex microfluidic chip. The optically clear IFC automatically stain captured cells. After staining, cells are automatically lysed and template is prepared for qPCR or sequencing analysis. The Fluidigm platform has a lower throughput with respect to droplet based single cell sequencing, but offers the possibility to take a picture of the cells before lysis, which avoid the acquisition of data from empty chambers and chambers with cell dublets or debris [6].

1.2.1.3 Drop-seq and Indrop-seq

Drop-seq, fig. 1.1B, was developed by Macosko [7], The procedure incapsulates a single cell in a droplet. The procedure uses highly diluted cells, to avoid doublets, and highly diluted beads, on which are covalently bound multiple copies of a oligonucleotide encompassing sequencing primer, cell barcode, UMI (Unique molecular identifier), and oligo-dT. The need of having only one bead and one cell in a droplet leads to the generation of large amount of empty droplets, droplets with a bead, droplets with a cell. Thus, a subset of cells, those incapsulated in the droplet without bead, are excluded from the sequencing. The cell encapsulation in the droplet is done in presence of lysis buffer, which disrupts the cell and allows the hybridisation of mRNAs to the oligo-dT. Subsequently, the droplet is disrupted and cDNA is generated and sequenced. InDrop-seq was developed by Zilionis [8], fig. 1.1B. The procedure is similar to drop-seq, but uses large soft hydrogel beads instead of small hard beads. Because of the hydrogel size, only one hydrogel bead can be incapsulated in a droplet, thus, there are no cells that are excluded from sequencing, as in the case of drop-seq. This technology is used by 10XGenomics, which is today the leader in single-cell sequencing.

1.2.1.4 Microwell

Microwells, fig. 1.1C, technology is an alternative approach to dropseq/inDrop-seq. Rhapsody, from BD, is a recent platform based on microwells and offers the possibility to sequence up to twenty thousands cells in a unique run. Rhapsody is essentially an evolution of drop-seq which uses microwells instead of droplets.

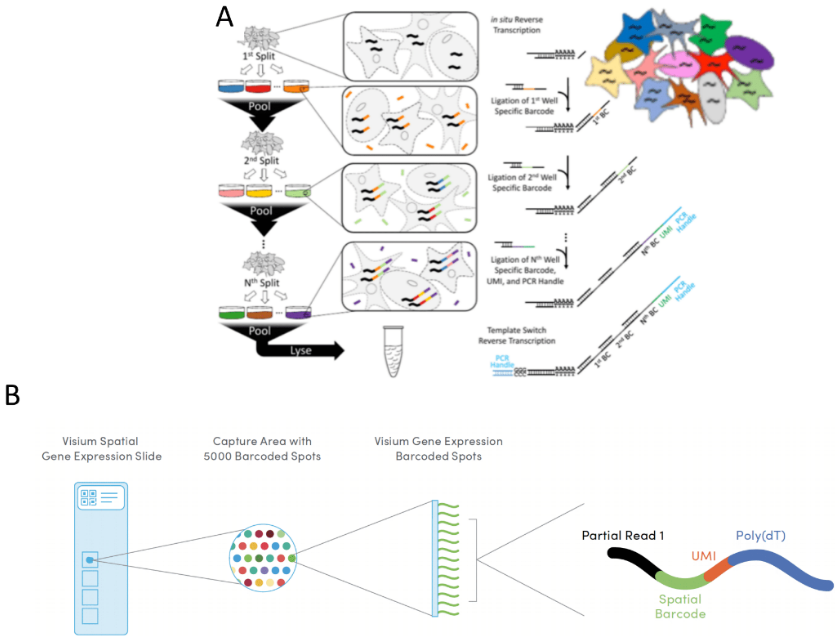

A
1st Split
Pool
2nd Split
Pool
Nth Split
Pool
Lyse
in situ Reverse Transcription
Ligation of 1st Well Specific Barcode
1st BC
Ligation of 2nd Well Specific Barcode
2nd BC
Ligation of Nth Well Specific Barcode, UMI, and PCR Handle
Nth BC
UMI
PCR Handle
Template Switch Reverse Transcription
PCR Handle
1st BC
2nd BC
Nth BC
UMI
PCR Handle
B
Visium Spatial Gene Expression Slide
Capture Area with 5000 Barcoded Spots
Visium Gene Expression Barcoded Spots
Partial Read 1
Poly(dT)
UMI
Spatial Barcode

Figure 1.2: A) Split-seq [9]. B) In 10XGenomics visium spatial transcriptomics platform 4 subarrays, 6.5 x 6.5 mm are available. Each subarray contains 5000 circular spatial spots, each with a unique spatial barcode and an approximate diameter of 50 μm. The spots are arranged with a center-to-center distance of 100 μm.

1.2.1.5 Split-seq

Split-seq is a single-cell RNA sequencing technique, developed few years ago [9], it is of particular interest because it allows to sequence a massive number of cells. This technique, fig. 1.2A, can be indeed used to sequence up to a million cells at the same time. The main idea of the method is to use the cell as reactor for the library preparation. Cells are permealized so that RNA or DNA oligonucleotides can enter inside the cell.

Using pulling and random splitting of cells it is possible to generate, using a relative little number of different probes, a complex barcode that is cell specific, fig. 1.2A.

1.2.1.6 Spatial transcriptomics

Spatial transcriptomics, fig. 1.2B, is acquiring a lot of interest today since it offers the possibility to combine both single-cell transcription profiling and spatial coordinates of the cells in the tissue section. Several papers have been recently published on spatial transcriptomics approaches [10] [11] [12]. Recently 10X Genomics commercialized ”visium” spatial transcriptomics technology (fig. 1.2B), which represents a slight improvement with respect to the spatial transcriptomics platform, published in [10]. The visium technology extends the number of acquisition spot up to 5000, where each spot has a diameter of 50μm, which however, does not guarantee single cell resolution. Visium technology uses frozen samples embedded in OTC. A fresh-frozen tissue section is then placed on an array containing capture probes that bind to RNA, fig. 1.2B. Tissue is fixed and permeabilized to release RNA, which binds to adjacent capture probes, allowing the capture of gene expression information. cDNA is then synthesized from captured RNA and libraries are prepared and sequenced.

1.2.2 Single Cell ATAC-seq

ATAC-seq stands for Assay for Transposase-Accessible Chromatin using sequencing, that means that a transposase can access to open chromatin and then the product of transposase excision is sequenced. A brief description of the method is given in fig. 1.3. Nucleosomes, have DNA wrapped around them and in between them, transposase cuts the DNA in regions not protected by nucleosome [13], sequenced fragments are then aligned to genome to identify accessible chromatin regions. The Atac-seq fragments are aligned to genome to identify chromatin accessible regions.

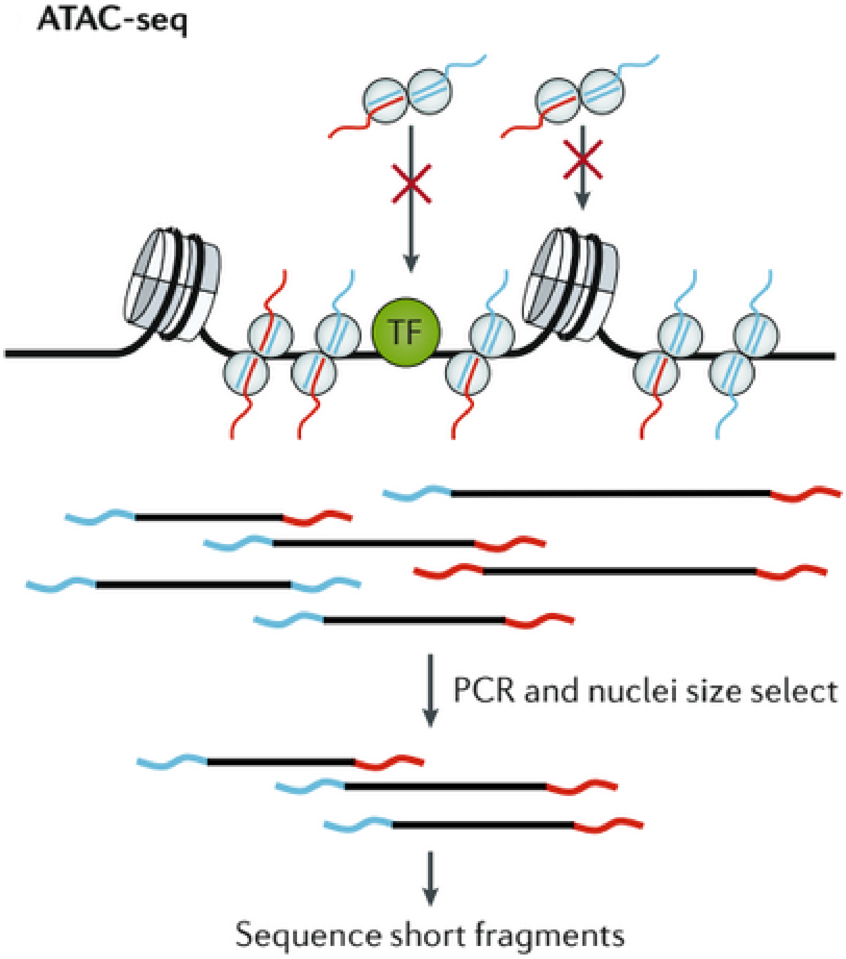


Figure 1.3: Assay for transposase-accessible chromatin using sequencing (ATAC-seq). An hyperactive transposase (Tn5) simultaneously cleaves genomic DNA and ligates adaptors to the excised DNA fragments.

1.2.3 Single Cell RNA-seq preprocessing

A first step in scRNAseq data analysis is data preprocessing, which is required to evaluate the quality of the overall experiment. Single cell raw sequencing data also need to be preprocessed to moderate the noise caused by zero inflations [14]. In single-cell RNA sequencing data, technical factors may induce an overabundance of zero measurements, which is defined as zero inflations. Transcription data are transformed, e.g. log_{10} converted or normalized [15] [16] [17], then transformed data undergo to data reduction [18] [19] to reduce the overall complexity of the dataset. After preprocessing, data are organized in subgroups, by clustering, to identify cell subpopulations. Subsequently, each cell subpopulation is further investigated to extract specific features, e.g. cluster's specific genes to be used as target for antibody staining in FACS analysis, or to identify cell type specific signatures.

1.2.3.1 Data preprocessing, filtering and normalization

Many preprocessing methods have been developed [20]. Notably, Diaz [21] and coworkers developed an approach, named Lorenz filtering, for the identification of low quality/dead cells in full transcript scRNAseq experiments (i.e. those experiments based on the use of smartseq sequencing protocol [22]).

Lorenz filtering is based on Lorenz statistics. In economy, Lorenz curve is a graphical representation of the distribution of income or wealth. It was developed by Max O. Lorenz in 1905 to represent inequality of the wealth distribution, fig. 1.4A. Diaz and coworker used this statistics to detect and remove cells characterized by a low quality transcriptome, i.e. cells with few expressed genes and overall low expression. Lorenz filtering [21], detects genes expressed at background levels in each sample, fig. 1.4B. Libraries with significantly high background levels are then discarded, fig. 1.4C .

Despite the removal of low quality cells, scRNAseq data requires to be normalized before being used for clusters analysis. Nowadays, the best way to normalize single-cell RNA-seq data has not defined yet [17], especially in the case of UMI data. Preprocessing is used to identify and remove low quality cells but cannot be used to remove technical

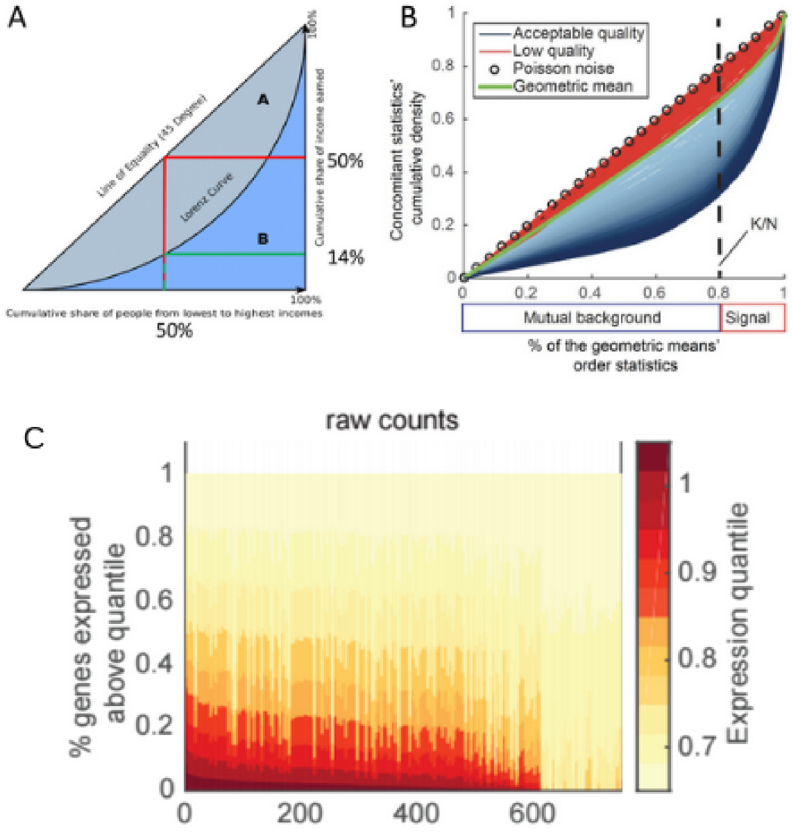


Figure 1.4: A) An example of Lorenz curve. In a perfect condition of equality, i.e. where 50 % of the population (y-axes) has an income that is exactly in the middle between the lowest and the highest income (x-axes, red lines), the hypotenuse of the triangle is the line of equality. In a condition of inequality, Lorenz curve shows which is the fraction of the population that has a specific income, e.g. green lines indicate that only 14 % of the population has an income that is exactly in the middle between the lowest and the highest income; B) An example of the Lorenz curve to estimate low quality cells; C) Fractions of each cells expressed above quantile. Those cells characterized by the smallest fraction of genes above quantiles, i.e. right part of the plot, are discarded.

and biological biases. SCnorm [15], is particularly interesting since performs a quantile regression approach for a robust normalization of single cell RNAseq data. SCnorm groups genes based on their counts-depth relationship, fig. 1.5A, then applies a quantile regression to each group in order to estimate the scaling factors needed to remove the effect of sequencing depth from the counts, fig. 1.5B. Scone [16], provides different global scaling methods that can be applied to UMI single-cell data. The scone Bioconductor package embeds the following normalization procedures: CLR (Centered log-ratio) , RLE (Relative log-expression), FQ (Full-quantile normalization), SSN (Sum scaling normalization), TMM (Weighted trimmed mean of M-values), UQ (Upper-quartile).

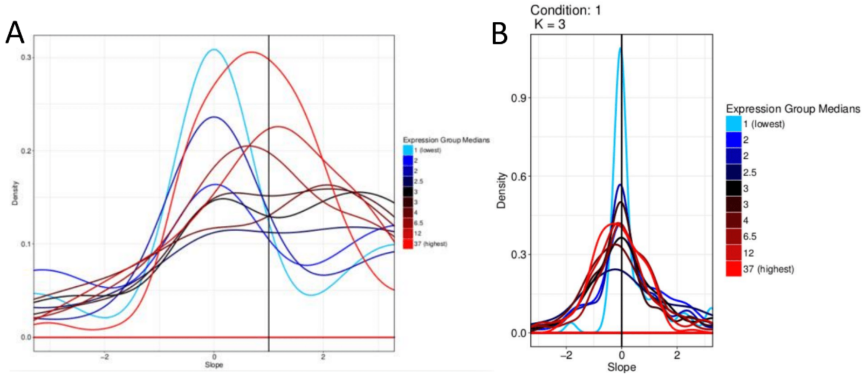


Figure 1.5: Count-depth relationship groups before (A) and after (B) normalization with SCnorm.

1.2.4 scRNAseq data Clustering

Clustering is the task of grouping a set of cell transcriptomes, in a way that cells in the same group are more similar to each other than those in other groups. A lot of single-cell studies use computational and statistical methods being developed primarily for the analysis of traditional bulk RNA-seq methods [23] [24] . These methods do not address the unique characteristics that make single-cell expression data challenging to be analyzed:

- outlier cell populations

- transcript amplification noise
- biological effects, such as cell cycle.

In additions, it has been shown that many statistical methods fail to alleviate other underlying biases, such as dropout events, where zero expression measurements occur due to sampling or stochastic oligo hybridization failures. A lot of single cell clustering tools uses hierarchical clustering methods, which usually rely on specific similarity metrics across the objects to be clustered. However, standard similarity metrics may not be generalize well across platforms and biological experiments. Clustering algorithm can be divided in two main group:

- Hierarchical
- Not Hierarchical

The main difference is that a hierarchical algorithm estimates the best numbers of clusters according to the dataset, instead, a not hierarchical algorithm divides the dataset in K clusters, with K defined by the user. The clustering tool described below provide a representation of the most used algorithms in single cells analysis today. Specifically all the clustering methods embed a data reduction procedure followed by a clustering algorithm.

1.2.5 Hierarchical clustering algorithm

1.2.5.1 Griph

The clustering performed by griph [25] is graph-based and uses the community detection method Louvain modularity [26]. Griph algorithm is close to agglomerative clustering methods, since every node is initially assigned to its own community and communities are subsequently built by iterative merging.

1.2.5.2 Seurat

Seurat was shown to deliver the overall best performance in cell clustering in two independent papers [27] [28], evaluating the efficacy of clustering tools for scRNAseq. Seurat aggregates cells using PCA information, reducing the clustering problem dimensionality by choosing a specific subset of PC dimensions. The clustering algorithm is also in this case based on Lovain modularity method [26].

1.2.5.3 Scanpy

Scanpy is a scalable toolkit for the analysis of large single-cell gene expression data [29]. It runs faster than Seurat and griph as the number of cells increases. Its Python-based implementation efficiently deals with datasets of more than one million cells, also supporting sparse data and allowing HDF5-based files. Its clustering algorithm is based on Leiden algorithm [30], that was designed to fix the issue of disconnected communities, which is a critical side effect associated to Louvain modularity approach.

1.2.5.4 SHARP

SHARP [31], Single-cell RNA-seq Hyper-fast and Accurate processing via ensemble Random Projection, is based on ensemble random projection (RP) and multi-layer meta-clustering, which well preserves cell-to-cell distance in reduced-dimensional space. By default, SHARP automatically determines the optimal number of clusters by integrating Silhouette index, Calinski-Harabasz index and hierarchical heights. SHARP adopts hierarchical clustering as the basic clustering method in individual RP clustering and weighted-base meta-clustering. SHARP, defines the number of reduced dimension as a function of the number of cells ($ceiling(\frac{log_2(ncells)}{(0.2^2)})$), where ceiling function rounds a number up to the nearest integer or to the nearest multiple of significance. To make the performance robust, SHARP by default adopts 15 runs of RPs for small-size datasets, whereas 5 runs of RPs for large-size datasets, fig. 1.6.

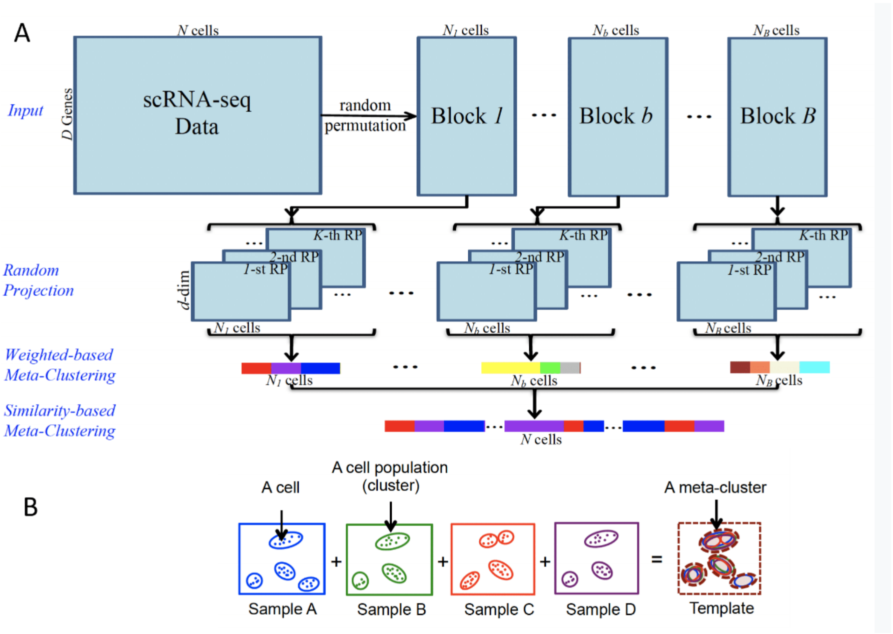


Figure 1.6: SHARP clustering algorithm; A) The dataset is splitted in multiple random blocks. Each block is transformed by random projection. The random projections for each block are clustered by weighted-based meta-clustering. Blocks clusters are combined by similarity-based meta-clustering; B) An example of the aggregation performed by meta-clustering algorithm. The present concept applies both to weighted-based and similarity-based meta-clustering.

1.3 Non hierarchical clustering algorithm

1.3.0.1 t-Sne + K-Means

t-Sne + K-means is often used in single cell rna-seq data. After data reduction by t-Sne method, cells are clustered by k-means method [18], which aims to partition the points into k groups, such that the sum of squares from points to the assigned cluster centres is minimized. At the minimum, all cluster centres are at the mean of their Voronoi sets, i.e. set of data points which are nearest to the cluster centre.

1.3.0.2 SIMLR

SIMLR, single-cell interpretation via multi-kernel learning [19], fig. 1.7, approaches the key question of "which cells are similar or different", learning an appropriate cell-to-cell similarity function from the input single-cell data. SIMLR offers three main unique advantages over other methods:

- it learns a distance metric that best fits the structure of the data by combining multiple kernels;
- SIMLR is unsupervised, thus allowing de novo discovery from data;
- SIMLR addresses the challenge of high levels of dropout events that can significantly weaken cell-to-cell similarities even under an appropriate distance metric, by employing graph diffusion, which improves weak similarity measures, which are likely to result from noise or dropout events.

SIMLR requires as input log_{10} transformed gene counts for each cell and the number of possible clusters to be used for partitioning the cells. Given a log_{10} gene expression matrix X, with N cells and M genes as input, where M has to be greater then N, SIMLR solves for S, a NxN symmetric matrix that contain cell pairwise similarities. In particular, S_{ij} represents similarities between cell i and cell j. S matrix is solved generating a matrix with C blocks on the diagonal, fig. 1.7, where, in every blocks, cells

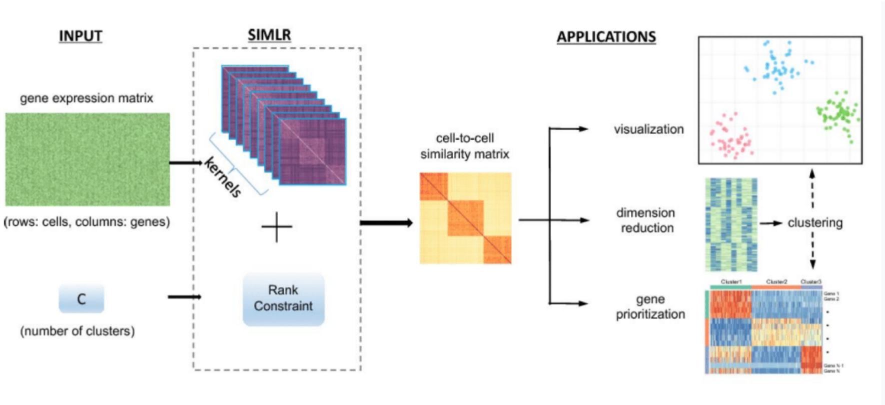


Figure 1.7: SIMLR clustering algorithm. The input count data matrix together with the K number of clusters are used to identify the subset of kernels that better fit the data structure. Then, kernel information is used to build a cell-to-cell similarity matrix that is used for data reduction by t-Sne and, the space reduced information are used for K-mean clustering and gene prioritization.

have larger similarities to other cells within the same subpopulation. SIMLR computes cell-to-cell similarities through the following algorithm

$$Minimize \sum_{i,j} D(x_i, x_j)S_{i,j} + \beta \|S\|_F^2 + \gamma Tr(L^T(I_N - S)L) + p \sum_l w_l log(w_l) \qquad (1.1)$$

where

$$D(X_i, X_k) = \sum_l w_l D_l(x_i, x_j), \sum w_l = 1, w_l > 0 \qquad (1.2)$$

$$L^T L = I_c, \sum : jS_{ij} = 1, S_{ij} >= 0 forall(i, j) \qquad (1.3)$$

where X_i and X_j are gene expression vector for ith and jth cell; $D(X_i, X_j)$ is the distance between cell i and cell j expressed as linear combination of distance metrics D_l with weights w_l ; I_N and I_C are identity matrix of N and C dimension respectively; β and γ are non negative tuning value. S between two cells should be small if the distance between them is large and, should be large, if the distance between them is small. One critical component of this optimization problem is the choice of the distance measure $D(x_i, x_j)$ between pairs of cells. It is well-known that the distance metric defined for the input space is critical to the performance of clustering and visualization algorithms designed for high-dimensional data [32]. Due to the presence of unusual zero-inflated distributions in single-cell data [11], standard metrics like the euclidean distance may fail to perform well. Instead, SIMLR incorporates multiple kernel learning, that flexibly combines multiple distance metrics. The algorithm works optimizing S, L and w :

- Fixing L and w to update S
- Fixing S and w to update L
- Fixing S and L to update W
- Similarity enhancement with a diffusion based step reduces the effects of noise and dropouts in single-cell data.

SIMLR iterates the four steps above until convergence. On the basis of our experience SIMLR is very precise, but very slow and works optimally for dataset up to 1000 cells.

1.4 Feature selection

Feature selection is a useful tool to detect genes,which are more representative for each cluster. Seurat and SIMLR integrated in their algorithm a feature selection option that automatically select the cluster specific signatures. Differential expression approaches as Anova-like method [33] can also be used as feature selection methods. COMET, with respect to other tools, uses a derivative of the hypergeometric test to identify single gene or combination of genes, up to 4, which provide an optimal representation of each cluster.

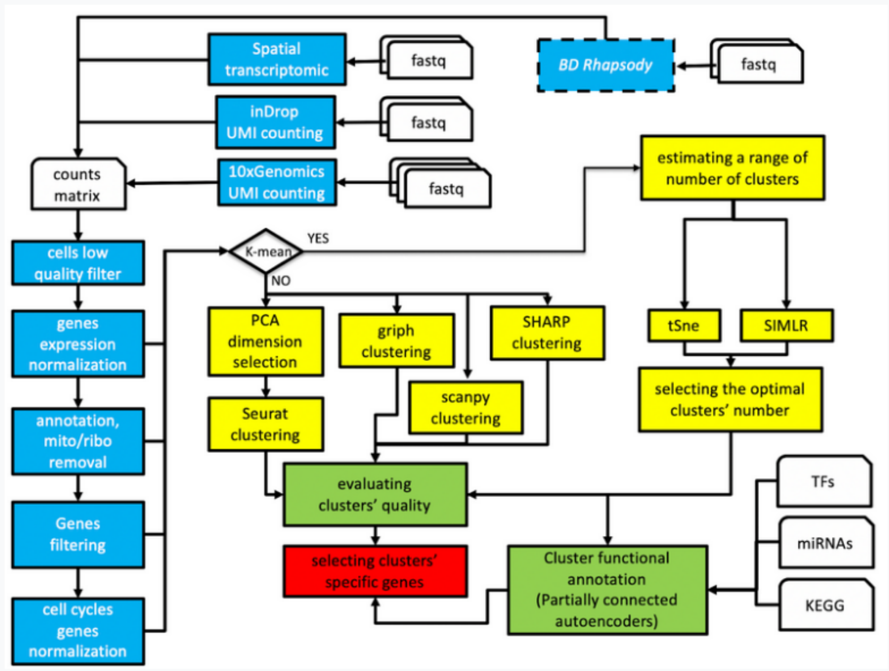
1.5 Aim of the Thesis

Reproducibility of a research is a key element in the modern science and it is mandatory for any industrial application. It represents the ability of replicating an experiment independently by the location and the operator. Therefore, a study can be considered reproducible only if all used data are available and the exploited computational analysis workflow is clearly described. However, to reproduce a complex bioinformatics analysis, the raw data and the list of tools used in the workflow could be not enough to guarantee the reproducibility of published results. Indeed, different releases of the same tools and/or of the system libraries (exploited by such tools) might lead to sneaky reproducibility issues. I am part of the group that founded the reproducible bioinformatics community [34]. The Reproducible Bioinformatics Project (RBP, http://www.reproducible-bioinformatics.org/) provides a general schema and an infrastructure to distribute robust and reproducible workflows. Thus, it guarantees to final users the ability to repeat consistently any analysis independently by the used UNIX-like architecture. As part of this project, I focused my research activity in the development of an analysis workflow for scRNaseq, fulfilling the requirements of being functional (i.e., information about the data and the tools used are saved as metadata) and computational reproducible (i.e., a real image of the computational environment used to generate the data is stored) through a user-friendly environment. Thus, my thesis work was devoted to the development of a SCRNAseq tool that could be used both as stand-alone or as cloud-based application.

2 Results

> Everything small is just a smaller version of something big
>
> *Finn the Human- Adventure time*

2.1 rCASC Workflow



Spatial transcriptomic
fastq
BD Rhapsody
fastq
inDrop UMI counting
fastq
counts matrix
10xGenomics UMI counting
fastq
estimating a range of number of clusters
cells low quality filter
K-mean
YES
NO
genes expression normalization
PCA dimension selection
griph clustering
SHARP clustering
tSne
SIMLR
scanpy clustering
selecting the optimal clusters' number
annotation, mito/ribo removal
Seurat clustering
evaluating clusters' quality
TFs
Genes filtering
miRNAs
selecting clusters' specific genes
Cluster functional annotation (Partially connected autoencoders)
KEGG
cell cycles genes normalization

Figure 2.1: Overview of rCASCC workflow. Blue boxes refer to data preprocessing, yellow boxes to clustering, green boxes to the characterization of the generated clusters. Red box indicates the set of tools for cluster-specific feature selection.

Reproducible classification and analysis for single cell data (rCASC) [35] workflow is divided in 4 modules, fig. 2.1. In brief:

- The first module implements different procedures, which are platform dependent, to generate the cells counts matrix. Actually spatial transcriptomics, InDrop and 10XGenomics are supported. BD Rhapsody fastq to counts tool is under development.
- Second module includes various quality control, normalization and filtering tools.
- Third module provides different options for subpopulation clustering (Griph, SIMLR, t-Sne+K-mean, Seurat, Scanpy). This module includes the cell stability score metrics (CSS), which is used to evaluate how stable is the subpopulation organization generated by clustering [35].
- The fourth module includes various feature selection tools, allowing the discovery of cluster specific signatures.
- The fifth module includes a feature discovery tool based on Sparsely-connected Autoencoder, allowing the identification of key genes which might play a role in the phenotype of the cell subpopulation.

rCASC workflow can be controlled by both command line (https://github.com/kendomaniac/rCASC) and java graphical interface (https://github.com/mbeccuti/4SeqGUI). rCASC is also available as cloud based application on the Elixir Galaxy framework Laniakea [36].

2.2 Quality control and Preprocessing

We implemented Lorenz statistic [21] as quality filter, for the removal of low quality cells in full transcript scRNAseq. Furthermore, we have developed a quality control plot called RiboMito plot, which facilitates, the identification of stressed and low quality cells [37]. RiboMito plot is associated to the function called scannobyGTF, which allows the removal of cells detected as outlier by RiboMito Plot. In RiboMito plot on the x-axis is shown the percentage of ribosomal mRNA counts calculated with respect to all UMI counts in each cell. In y-axis, it is given the percentage of mitochondrial mRNA counts for each cell. Furthermore, scannobyGTF offers the possibility to remove ribosomal and mitochondrial genes from the dataset. In fig. 2.2A, it is shown an example of RiboMito Plot for mouse bone marrow cells. Various groups of cells characterized by a different ribosomal content are present, few cells are also characterized by a high fraction of mitochondrial genes associated to low fraction of ribosomal counts and few genes expressed in each cells, fig. 2.2A rectangle area. Cells subset can be removed because probably constituted by stressed cells with a little informative transcriptome, i.e. less then 100 genes called present (N.B. ribosomal and mitochondrial genes are in total 114 in mouse). In fig. 2.2B, it is shown the RiboMito Plot for $CD8^{+}T$ cells activated by *Listeria* infection [38]. Also in this case low quality cells are characterized by high fraction of mitochondrial genes and a poor transcript content, left side of the plot in fig. 2.2B. Ribosomal RNA and ribosomal proteins represent a significant part of cell cargo. Large cells and actively proliferating cells have respectively more ribosomes and more active ribosome synthesis [39]. Thus, ribosomal proteins expression represents one of the major confounding factor in cluster formation between active and quiescent cells. Furthermore, the main function of mitochondria is to produce energy through aerobic respiration and the number of mitochondrial mRNAs depends on cell metabolic demands [40]. This might also affect clustering, favoring the separation between metabolic active and resting cells. In fig. 2.2C, it is shown the effect of removal of ribosomal and mitochondrial genes from the dataset using scannoByGTF function.

RiboMito Plot was used in the paper of Ordonez et al [37]. In this paper we evaluated the possibility to remove apoptotic and proapoptotic cells using only computational filters. In fig. 2.3A, it is summarized the experiment made in [37] using HEK293 cells, in which apoptosis was induced. Squares in fig. 2.3A indicates apoptotic, proapoptotic and healthy cells isolated by FACS and sequenced by 10XGenomics platform. In fig. 2.3B, it is shown the t-Sne plot for the three populations, only apoptotic cell are relatively well separated as instead proapoptotic and healthy cells are clustering together. In

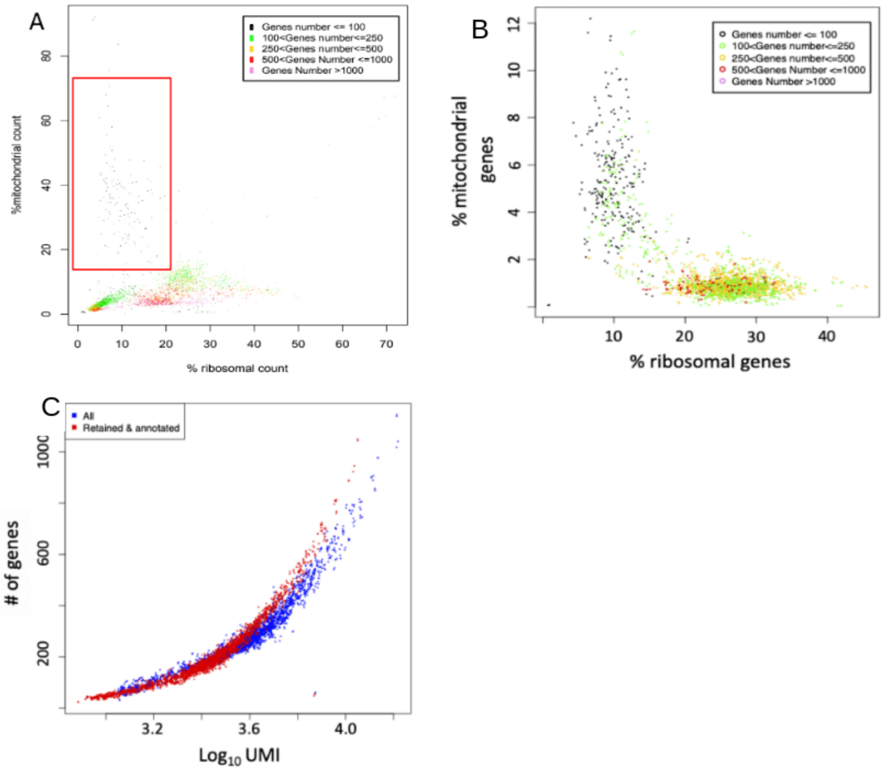


Figure 2.2: A) Mouse bone marrow scRNAseq; B) $CD8^+$ T cell activated by Lysteria infection; C) Effect on number of detected genes after the removal of ribosomal and mitochondrial genes.

fig. 2.3C-E, it is showed the RiboMito Plot for healthy, proapoptotic and apoptotic cells respectively. Also in RiboMito Plot only apoptotic cells show different distribution of cells with high number of detected genes with respect to health and proapoptotic cells. The above mentioned observations combined with other analysis in [37] suggested that the only way to remove apoptotic and proapoptotic cells was preprocessing cells before scRNAseq.

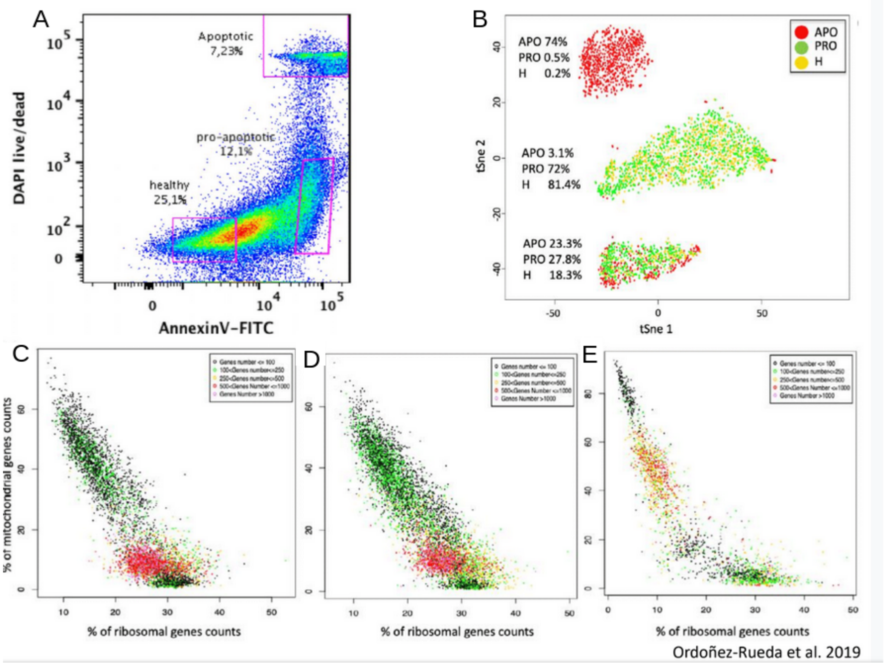


Figure 2.3: A) Live/dead cells represented with DAPI and Annexin V, in order to detect healthy, pre-apoptotic and apoptotic cells; B) t-Sne plot, apoptotic can be separated instead pro-apoptotic and healthy are clustered together; C) Health cells MitoRibo Plot; D) Pro-apoptotic cells MitoRibo Plot; E) Apoptotic cells MitoRibo Plot.

2.2.1 Variance and expression Filtering

An ”ever green” way to remove the noise from single cell RNA-seq count matrix is filtering genes by expression. rCASC provides a function named topX, which ranks genes, according to gene expression/variance and removes genes that are below a user defined threshold value. Specifically gene variance is calculated using edgeR tagwise dispersion

[33]. The method estimates the gene-wise dispersion implementing a conditional maximum likelihood procedure [41] . On the basis of our experience it is useful to combine the two features, filtering the full dataset, fig. 2.4A, by variance, fig. 2.4B, in order to keep the most variable genes, thus enhancing the difference between cells, and then reducing noise by removing low expressed genes, fig. 2.4C.

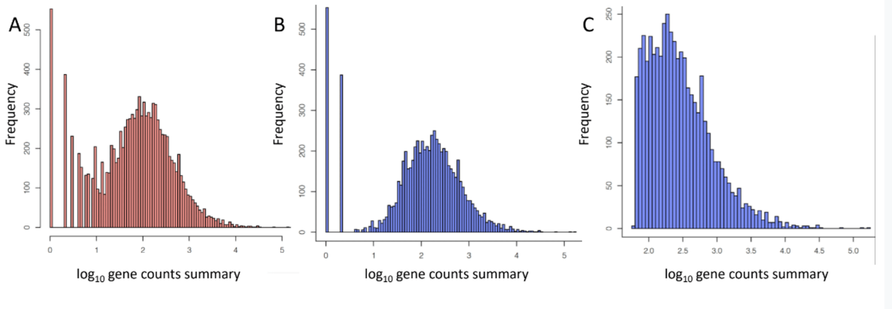


Figure 2.4: Example of TopX function usage; A)Unfiltered dataset. B) Top 10000 most variant genes; C) Top 5000 most expressed genes out of 10000 most variant genes

2.2.2 Data Normalization

In rCASC workflow are available two different tools:

- SCnorm [15], which works best on datasets based on full length transcript scRNAseq, i.e. library prep based on Smart Seq protocol. SCnorm performs a quantile-regression as approach for robust normalization of single-cell RNA-seq data. In brief, SCnorm groups genes based on their count-depth relationship then applies a quantile regression to each group in order to estimate scaling factors which will remove the effect of sequencing depth from counts. SCnorm is not intended for datasets with more than 80% zero counts, because of lack of algorithm convergence in these situations. In fig. 2.5, it is shown an example of SCnorm normalization.

- scone [16], which provides different global scaling methods that can be applied to smart-seq and UMI single-cell data. The scone Bioconductor package embeds the following normalization procedures:

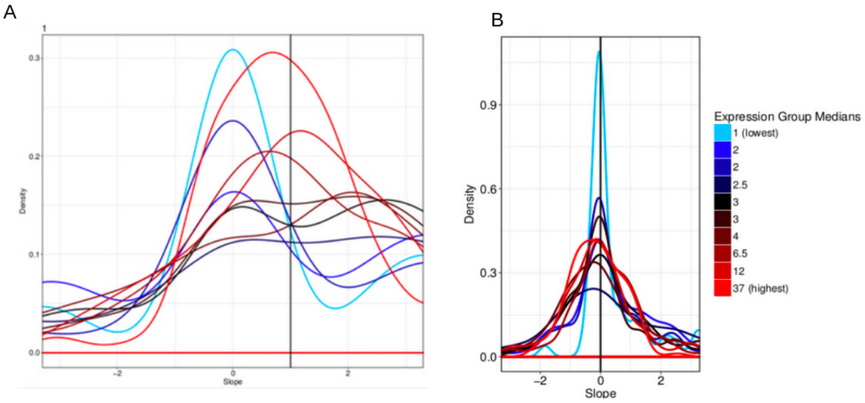


Figure 2.5: A)Evaluation of count-depth relationship in un-normalized blood mouse cells dataset. Transcript of different length are sampled differently by short transcript; B) Effect of SCnorm normalization. SCnorm eliminate the differences in sampling between transcripts of different length.

- CLR, Centered log-ratio, in this normalization procedure sample vectors undergo a transformation based on the logarithm of ratio between individual elements and the geometric mean of the vector.
- RLE, Relative log-expression, here the scaling factors are calculated for each row as median of the ratio, for each gene, of its read count of its geometric mean across all lanes.
- FQ, Full-quantile normalization uses an approach developed for microarrays normalization. As first step each column of the count matrix is ranked from the smaller to the larger value. As second step the mean of each row is calculated and it is used to replace all values in the row. As third step each column is sorted on the basis of the gene values. The result of the normalization is a new count table that has for each column the same distribution.
- SSN, Sum scaling normalization, in this normalization gene counts are divided by the total number of mapped reads (or library size) associated with their column and multiplied by the mean total count across all the samples of the dataset.

- TMM, Weighted trimmed mean of M-values. To compute the TMM factor, one column is considered as reference sample and the others columns of the count matrix as test samples, with TMM being the weighted mean of log ratios between test and reference, after excluding the most expressed genes and the genes with the largest log ratios.
- UQ, Upper-quartile, in this normalization, the total counts are replaced by the upper quartile of counts different from 0 in the computation of the normalization factors.

2.2.3 Detecting and removing cell cycle bias

Single-cell RNA-Sequencing measurement of expression is often affected by large systematic bias. One of the major source of this bias is cell cycle, which introduces large within-cell-type heterogeneity that can obscure the differences in expression between cell types. Since not all datasets are affected by cell cycle bias, it is essential to address if the removal of cell cycle effect is really needed. reCAT [42] is a modelling framework for unsynchronized single-cell transcriptome data, which is able to reconstruct cell cycle time-series. Thus, reCAT cell cycle prediction step can be used to check if cell cycle effect can be detected in a dataset, fig. 2.6A, and then cell cycle normalization approach can be applied, e.g. using ccRemove [43]. In case cell cycle effect is not detected, fig. 2.6B, it is not necessary to perform any cell cycle normalization.

ccRemove [43] is a cell cycle effect remover which removes cell cycle effects preserving other biological signals. In fig. 2.7, it is shown the effect of the ccReove, implemented in rCASC, using the Buettner dataset [44]. The removal of the cell cycle effect, fig. 2.7B, is clearly shown by a reduction of the variance explained by PC1 and PC2 in the PCA plot with respect to the untreated dataset, fig. 2.7.

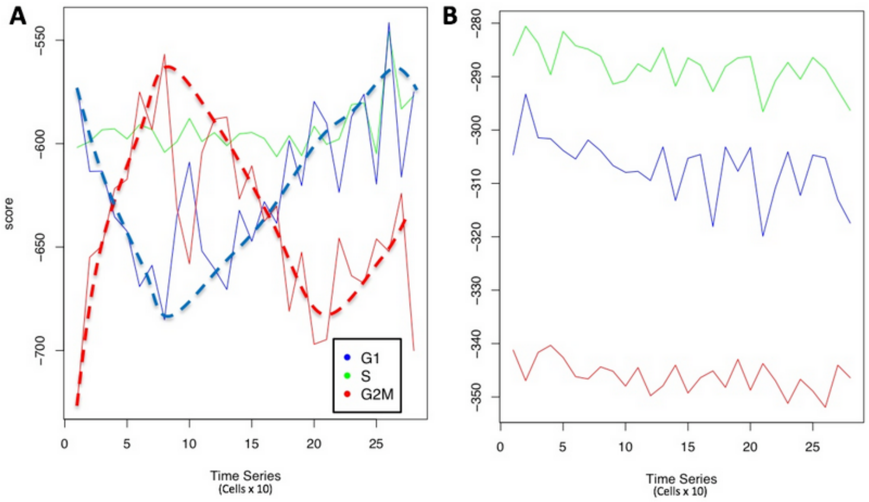


Figure 2.6: Cell cycle assignment to the cells using reCAT. A) Buettner dataset, cells are expected to be distributed in G1, S and G2M, B) Naive T cells, expected to be mainly in G0, reCAT does not manage to detect any cell cycle trend

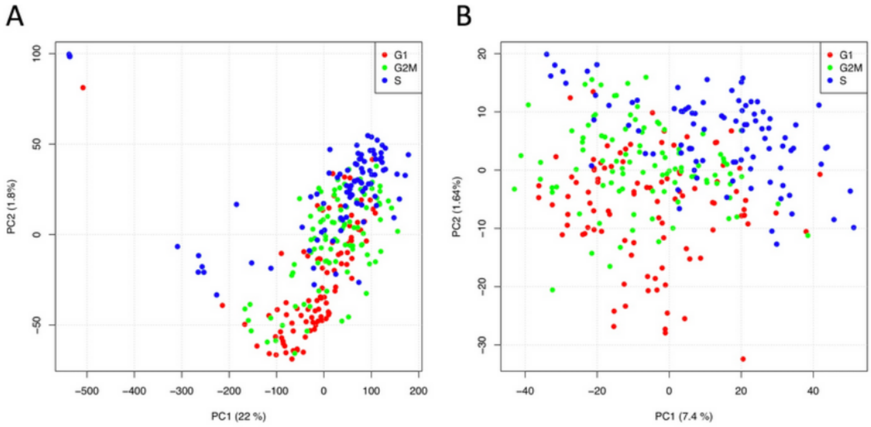


Figure 2.7: rCASC implementation of the ccRemove. A) PCA analysis of Buettner et al. [44] log_{10} transformed raw data, B) PCA analysis of the same dataset after cell cycle effect removal.

2.3 Subpopulation discovery

Subpopulation discovery is the main topic of scRNAseq. The partition of cells in different groups can be done using multiple clustering approaches. In clustering there are two main decisions that have to be taken:

- Which data preprocessing has to be applied.
- Which clustering algorithm has to be used.

Different combinations of algorithm and data reduction technique can affect the cells groups organizations. Furthermore, perturbation of the dataset, e.g. removal of a subset of cells, can also affect the distribution of the cells in different clusters. A golden standard solution to cluster single cells is not available at the present time. To provide more flexibility to the final user, in rCASC we have implemented different clustering algorithms and data reduction methods. Specifically, we have implemented :

- K-means clustering using tSne as data reduction approach. tSne is a non linear data reduction method, that was designed for big data [45] and later was applied to single cell Rna-seq.
- K-means clustering using as data reduction method a machine learning approach [19]. Kernel learning selects the best combination of kernels providing an optimal separation between groups of cells.
- Meta-clustering using as data reduction random projections [31]. This clustering approach splits the cells in multiple groups and, after data reduction, uses a meta-clustering approach to group them.
- Clustering using Louvain methods for community detection and as data reduction PCA meta features [46]. In this clustering approach only a subset of PCs is passed to Louvain algorithm to find the subpopulations organization.
- Clustering using Louvain method for community detection and tSne as data reduction method [25]. In this clustering approach the data are passed to louvain

algorithm to find the optimal subpopulation organization.

- Clustering using the Leiden algorithm and tSne as data reduction method. [29]

2.3.1 Defining the optimal number of clusters

SHARP, griph, Seurat and Scanpy are able to define the optimal number of clusters to be used for cells partition. K-means clustering instead, requires that the user provides the number of clusters to be used by the k-means algorithm. To help the user to define the optimal number of clusters we followed the idea that optimal clusters number should not be much affected by perturbation applied to the input dataset. Then, we used griph, because of its high clustering speed, together with bootstraps of the original dataset, i.e. in every bootstrap we remove randomly 10% of the cells. The aim of this procedure is the identification of the most represented number of clusters along all the permutations. The most frequent number of clusters identified by griph clustering, is used in k-means clustering. The function that uses griph to detect the most represent value of number of clusters generates an histogram of the detected number of clusters, fig. 2.8. We tested this function on the set A-D (See Methods section). These datasets are characterized by a progressive increase of similarity among subpopulations, which makes progressively more challenging the clustering. In this specific case, we observe that the sensitivity to perturbation increase from setA to setD, indicating that subset of cells characterized by similar transcription profiles generate clusters that are sensitive to dataset perturbation.

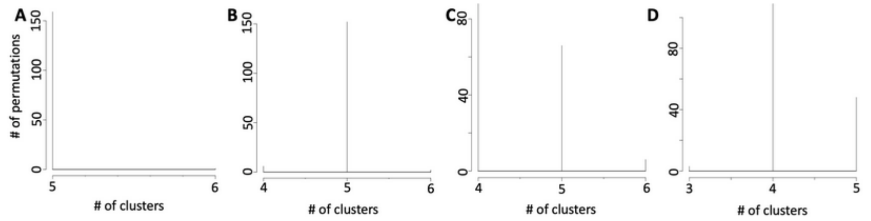


Figure 2.8: Clusters number is dependent on cell subpopulation similarity: A) number of clusters detectable by griph in setA, B) number of clusters detectable by griph in setB, C) number of clusters detectable by griph in setC, D) number of clusters detectable by griph in setD

2.3.2 Cluster stability

Since the combination of clustering algorithm and data reduction methods can significantly affect the subpopulation partitioning, the question which is the best clustering approach cannot have a general answer. A typical approach would be repeating clustering using different algorithm, but the question ”which is the best partitioning approach” remains questionable, because of the lack of a robust quality metrics.

2.3.3 Henning Cluster stability score

Few years ago Henning [47] proposed a method to measure clusters stability published on R archive as fpc package. The function ”clusterboot” from the fpc R package [48] allows the evaluation of cluster stability measurement (CSM). We implemented, in rCASC, CSM for clusters generated with SIMLR, Seurat and k-means. We analyzed setA and setC, where SetA is made of monocytes (M), natural killer (NK), B-cells (B), hematopoietic stem cells (S), Naive T cells (N), as instead setC is made of cytotoxic T cell (C), monocytes (M), T-helper cells (H), Natural killer cells (NK), and Naive T-cells (N). The presence of three subpopulations of T-cells makes setC more challenging for clustering. Henning’s CSM identifies as highly stable all setA clusters, fig. 2.9A, i.e. CSM close to 1, as instead, fig. 2.9B, only cluster 5 (M), was detected with a high CSM in setC. Unfortunately, this method is quite slow and does not scale well as the number of cells increase.

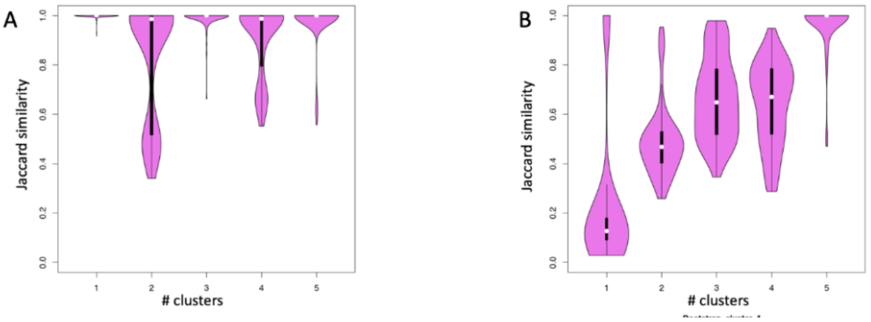


Figure 2.9: Estimating cluster stability output for two SetA (A) and SetC (B) using CSM and SIMLR clustering.

2.3.4 Cell Stability Score (CSS)

On the basis of the results obtained using Henning method we decided to approach the problem of cluster stability from a different point of view. Instead of evaluating the cluster stability we decided to evaluate the stability of cells in a cluster as consequence of dataset perturbations. We designed a metrics called cell stability score (CSS), fig. 2.10. These are the CSS evaluation steps :

- Single cell RNA-seq data S_0, (fig. 2.10A), are grouped by a clustering algorithm, fig. 2.10B.
- A new dataset S_j is generated by randomly removing a random set of cells (usually is 10%), fig. 2.10C.
- The dataset S_j is clustered, fig. 2.10C.
- A score is assigned on the basis of the cells clustering together in S_0 and S_j ,fig. 2.10D, fig. 2.10E.
- The above steps are repeated N times
- The final visualization show S_0 clusters, where cells are labelled according to their stability score using a symbol assigned to each cell, fig. 2.10H.

CSS is characterized by some interesting properties. In fig. 2.11, the result of the analysis with SIMLR and CSS of setA and setC are shown. CSS stability, fig. 2.11C, for setA clusters, fig. 2.11A, is very high for all clusters, i.e. $>=$ 75%, and the clusters composition is relatively homogeneous with clusters made by more then 92 % by the same cell type fig. 2.11A. In setC, fig. 2.11B, , we observe that clusters are quite heterogeneous in composition, fig. 2.11B, and unless for few cases, clusters are contaminated by more then 10% of other cell types. Interestingly, the CSS for the setC clusters, fig. 2.11D, shows an inverse relation with the homogeneity of the clusters. This observation suggests that CSS has the peculiar propriety to measure the cell homogeneity of the cluster.

The CSS propriety previously introduced was further confirmed in the experiment shown in fig. 2.12. We have reanalyzed the scRNAseq data from Pace [38]. In her

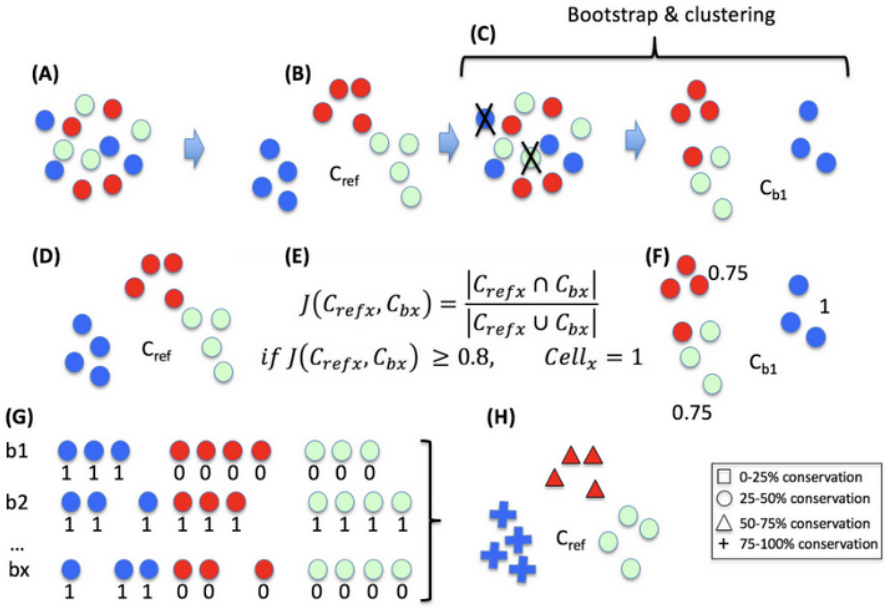


Figure 2.10: CSS algorithm

paper Pace generated scRNAseq for the following populations of T-cells: naive WT $CD8^+$ T-cells, naive Suv39h1 KO $CD8^+$T-cells Lysteria activated WT and Suv39h1 KO $CD8^+$T-cells. We have combined 400 cells for each experimental group and we have used SIMLR with 160 bootstrap to estimate CSS for each cell, fig. 2.12. Clusters 3 and 5 refers to naive WT/KO T-cells. All clusters are very stable. Interestingly, carefully looking at the boundaries between clusters it is possible to observe cells, fig. 2.12B rectangles, with lower stability, since they shuffle between two clusters, probably because the transcriptome is not sufficiently robust to assign those cells to one of the two clusters. This observation, brings us to speculate that these cells at the boundaries of clusters, represent partially differentiated cells. However, to further investigate this point we need a different benchmark dataset which we could not manage to find, yet. The ability of CSS to identify unstable cells offers also the possibility to refine clusters, removing unstable cells.

The effect of the perturbations induced in the clustering upon the removal of 10%, 20%, 30% and 50% of the data set was also investigated in SetA, fig. 2.13. Increasing the

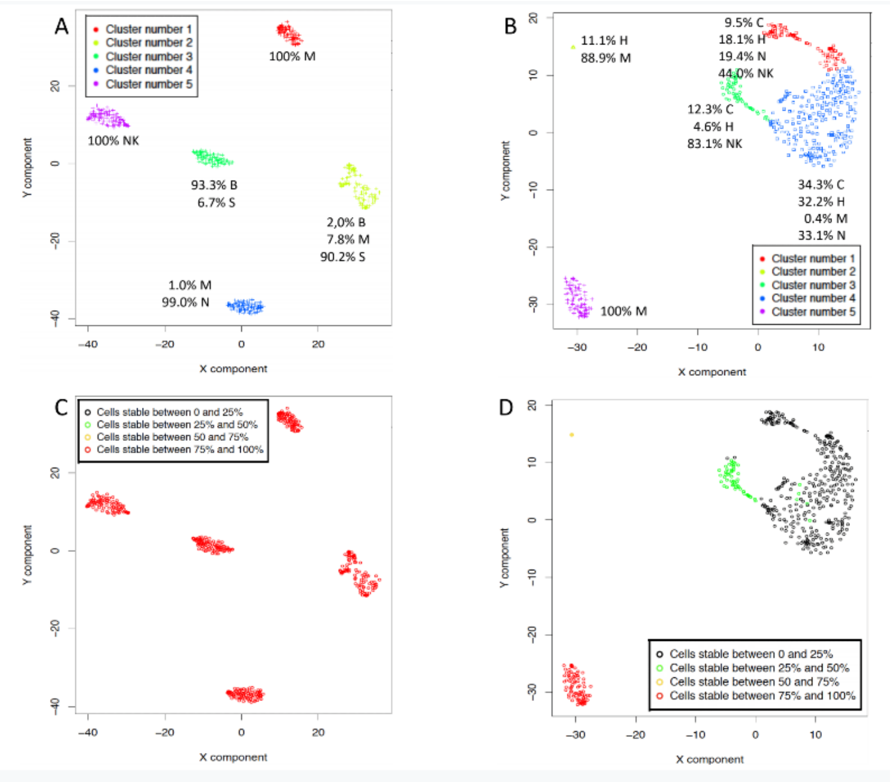


Figure 2.11: A) SetA cells clusters are made at least of 90 % of a unique cell type; B) Clusters referring to setC are characterized by different heterogeneity, only cluster 5 is made of 100 % of the same cell type; C)CSS of setA by color code. All clusters show a CSS greater then 75 %, red dots; D) Cluster 3 CSS score is only between 25 to 50%. Clusters 4 and 1 have CSS below 25 %.

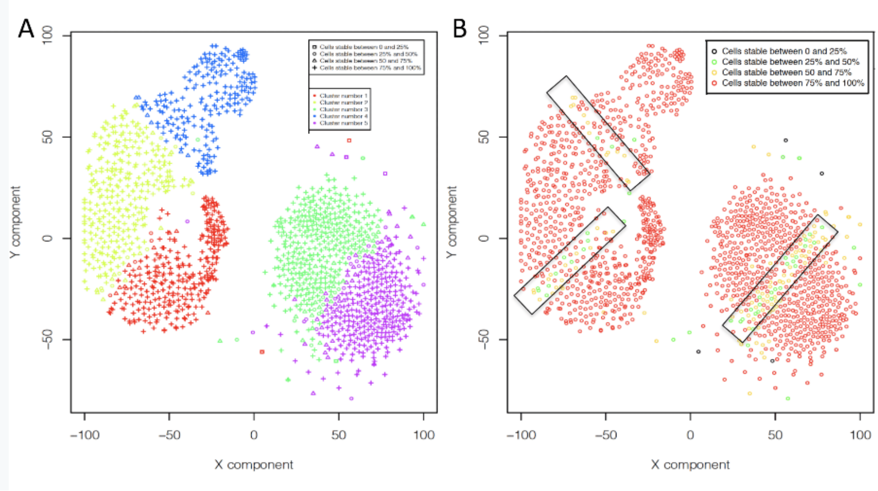


Figure 2.12: Cell Stability score results for Pace Dataset

fraction of cells removed at each permutation affects, in a negative way, the overall cell stability score of each cell in each cluster. However, the reduction in CSS is not identical for all clusters. In fig. 2.13, it is clear that cluster 2, completely made of NK cells, is the most stable cluster to the perturbations induced by increasing the number of removed cells. On the other side, cluster 4, mainly made by stem cells (92 cells), together with few B-cells (2 cells) and Monocytes (7 cells) is the least stable. Sorting by increasing CSS the cells in cluster 4, fig. 2.13D, B-cells and Monocytes are found within the first 15 most unstable cells. This observation suggests that studying CSS, increasing the fraction of cells removed at each permutation, could highlight the presence of cells located at the boundaries of different clusters.

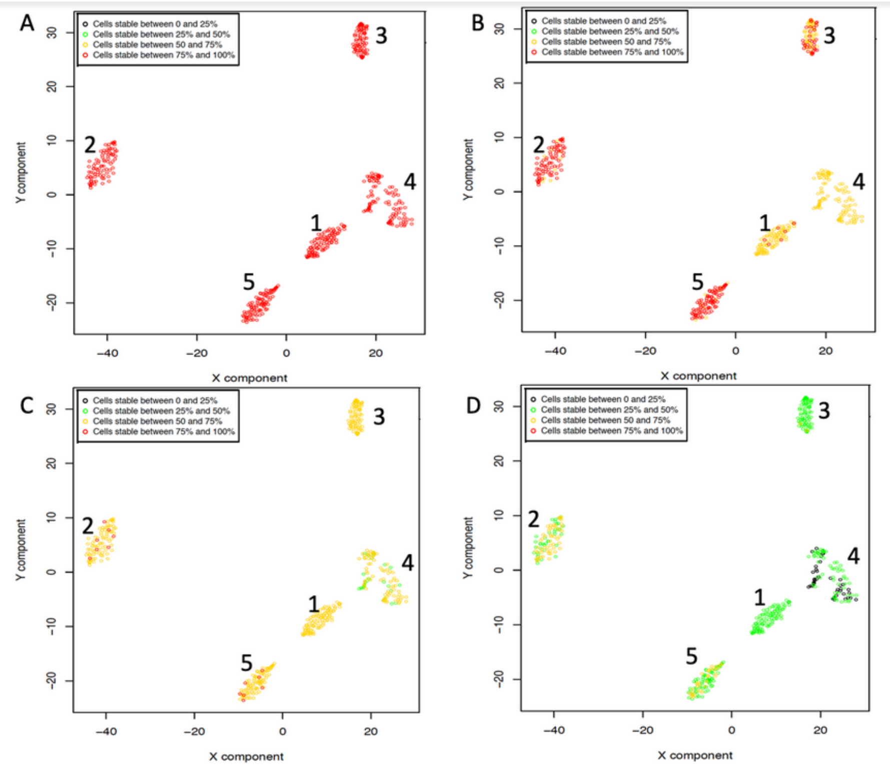


Figure 2.13: Cell stability score for SIMLR analysis of setA with k=5 clusters, removing progressively 10 (A), 20 (B), 30 (C) and 50 (D) percent of the cells at each permutation.

We made a comparison between the five clustering tools implemented in rCASC by mean of CSS measurements, fig. 2.14. From this analysis, we observed that Griph, Seurat, SIMLR and SHARP (fig. 2.14B,C,D,E) generate clusters with high CSS, which fit well with the homogeneity of clusters. On the other side tSne + kmean generates clusters with lower cluster homogeneity and therefore lower CSS, fig. 2.14A. The worst clustering was generated by scanpy, fig. 2.14F. In the case of scanpy we tested various combination of perplexity and PCs values, we failed to find any condition able to detect 5 clusters and the clusters stability was always very poor. We are now implementing a new function that can work as skeleton for the implementation of any clustering tool within the permutation framework. From the point of view of the computing time, the above mentioned analysis took 159 mins for tSne, 168 mins for SIMLR, 194 mins for Seurat, 61 mins for griph and only 9 min for scanpy. Thus, we further controlled the performance of the tools giving different number of cells, fig. 2.15. From this analysis is clear that not all tools scale well as the number of cells increases. SIMLR and t-Sne/kmean become very slow if the number of cells grow over 2000, fig. 2.15A. Instead griph, Seurat and scanpy are still very fast if the number of cells is up to 100000, instead the number of genes used in clustering are not affecting very much the speed of the analysis, fig. 2.15B. It has to be noted that if the cells number increases the request of RAM for analysis can be very high.

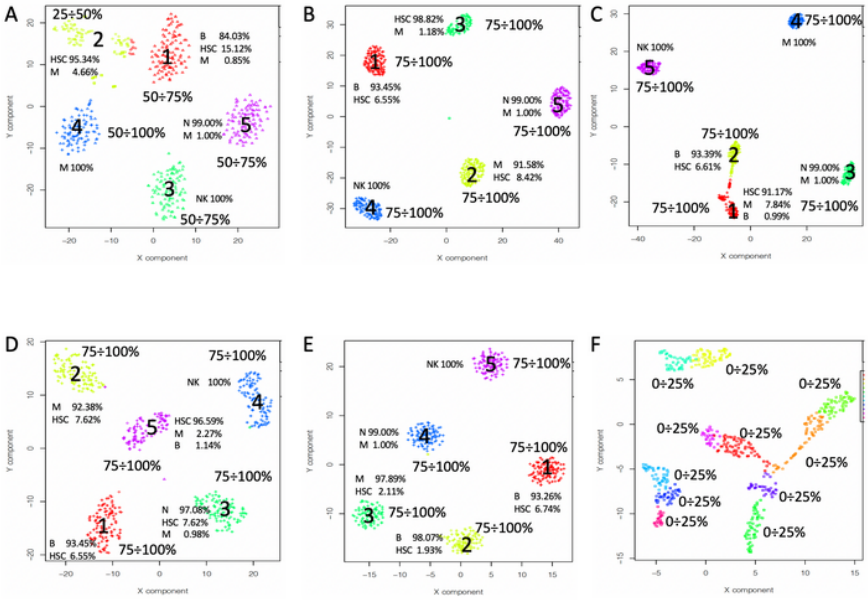


Figure 2.14: A) tSne+K=mean, B) griph, C) SIMLR, D) Seurat, E) SHARP, F) scanpy. N.B. Colors do not identify the same cluster in different panels

2.4 Spatial Transcriptomics Analysis

Spatial transcriptomics allows to link each cell transcriptome to cell spatial location within a tissue sample. The functions embedded in rCASC allow to overlay clustering results on the spatial localization of the cells in the tissue section under analysis. An example of rCASC output for 10XGenomics visium spatial transcriptomics is shown in

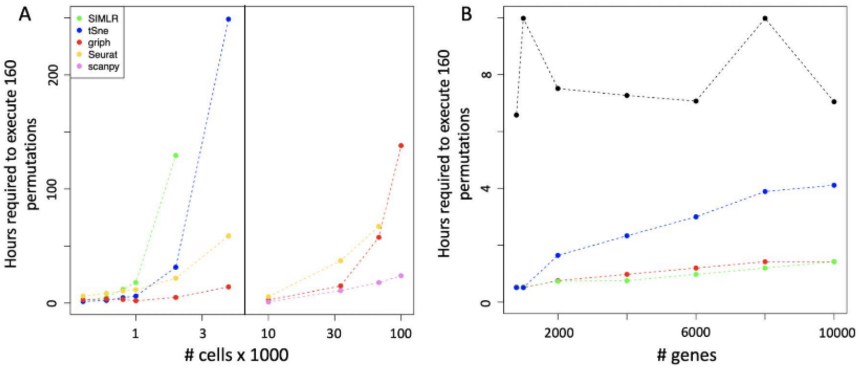


Figure 2.15: Computing times for SIMLR(green), tSne(blue), griph(red), Seurat(yellow), scanpy(violet); A) Computing time increasing the number of cells; B) Computing time increasing the number of genes.

fig. 2.16

A critical issue of 10XGenomics spatial transcriptomics is the possibility to have in a spot between 1 to 30 cells, depending on the tissue type and cells density. Since some spots can be made of a mixture of different cells, we have refined CSS including spatial information.

Each spot, in the spatial transcriptomics design, is surrounded by 6 other spots. CSS adapted for spatial transcriptomics assign a "bonus" to CSS score, if at least a certain number of surrounding spots belong to the same cluster of the central spot. For example as shown in fig. 2.16C, the cell in the center belongs to cluster 1 together with other 3 cells that surround it, while the other three cells belong to the cluster 5. This spatial bonus increases the CSS score as the number of surrounding spots of the same type increases.

2.5 Features Selection

2.5.1 SIMLR/Seurat and differential expression for feature selection

SIMLR and Seurat can identify subsets of cluster-specific genes. Independently by the feature selection tool used, the cluster-specific genes expression can be visualized among

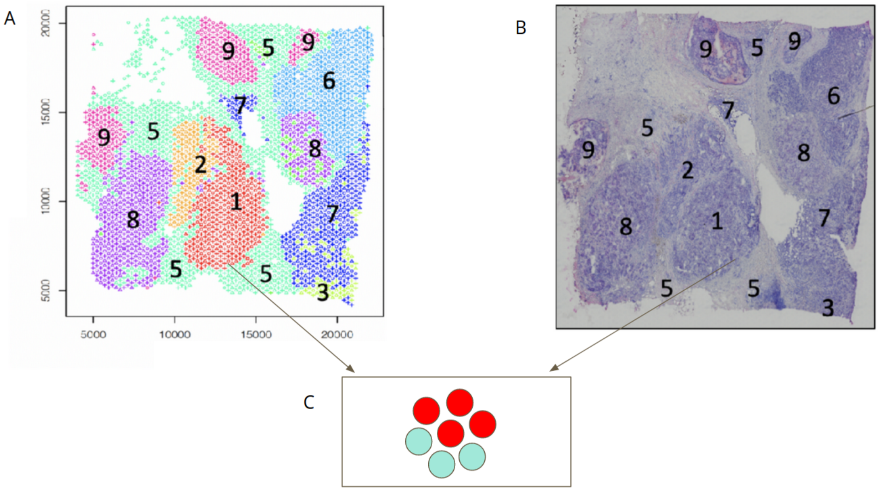


Figure 2.16: Breast cancer histological section available as part of the demo data for visium technology.A) SIMLR clusters location in the tissue section. B) Hematoxylin and eosin image c) Zoom on spots organization.

clusters. In fig. 2.17, it is shown an example of a subset of genes detected as cluster specific by SIMLR in fig. 2.17A, the plot allows the evaluation of the maxDeltaConfidence (x-axis) and minLogMean (y-axis),which are required by SIMLR genesSelection, in this case they are respectively 0.01 and 20. In fig. 2.17B, high number of counts is showed by bright yellow color as instead low number of counts is shown by bright blue color. Clusters are represented on the top of the figure as bars of different colors and, in fig. 2.17C, it is shown the Z-score heatmap, high positive Z-score is indicated by bright yellow color as instead high negative Z-core is indicated by bright blue color. In fig. 2.17F, cells are colored on the basis of the amount of CPMs for the gene under analysis, blue color refers to low CPM as instead bright yellow indicates high CPM. The expression is generated for each of the genes coming from SIMLR genesSelection output.A limitation of the above approach is the high time required for genes selection. Another possible approach to detect cluster specific genes is ANOVA analysis. We have implemented in rCASC the ANOVA-like method available within the edgeR [33] package, which can be used to detect clusters differential expression with respect to a reference cluster.

We also implemented DESeq2 differential expression analysis, since it was shown to

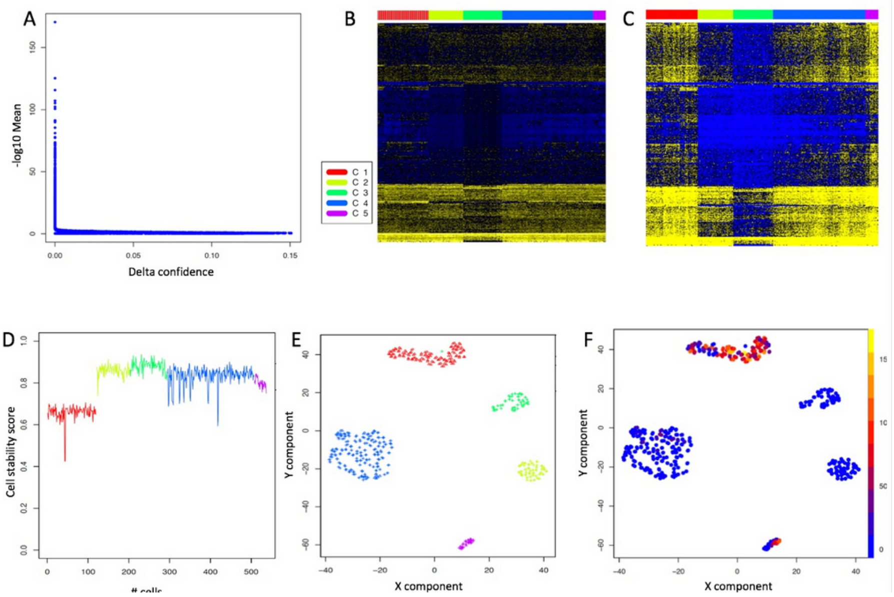


Figure 2.17: SIMLR visualization of gene cluster specific. A) SIMLR Prioritization plot, B) log_{10} counts heatmap from the output of genesSelection. C) Z-score heatmap D) Cell stability score. E) Cell stability score plot of the dataset clustered with SIMLR. F) Single gene CPM expression in SIMLR clusters

represent one of the most efficient tool for scRNAseq data [49]. On the basis of our observations both SIMLR/Seurat prioritization tools, as well as differential expression approaches as edgeR ANOVA-like and DESeq2 are very much dependent on the statistical characteristic of the clusters and some specific features might be missed. Recently a new tool become available, COMET, which is based on a different approach with respect to those described above.

2.5.2 COMET

COMET, fig. 2.18, implements the XL-minimal HyperGeometric test (XL-mHG), which is a semiparametric test to assess enrichment in ranked list. This method binarizes gene expression data in a gene-specific and cluster-specific manner, assessing for each gene G and cluster K, the extent to which gene G could be a good marker for cluster K. XL-mHG asks whether exists an unusual accumulation of a subset of those “interesting items” at the “top of the list”. In fig. 2.19, it is shown an example of COMET output (B) on a Seurat clusterization of a spatial transcriptomics lymph node dataset (A). CXCL5 (B) was found as the first top ranked gene for cluster 4, indicating that its expression is prevalently associated to cluster 4.

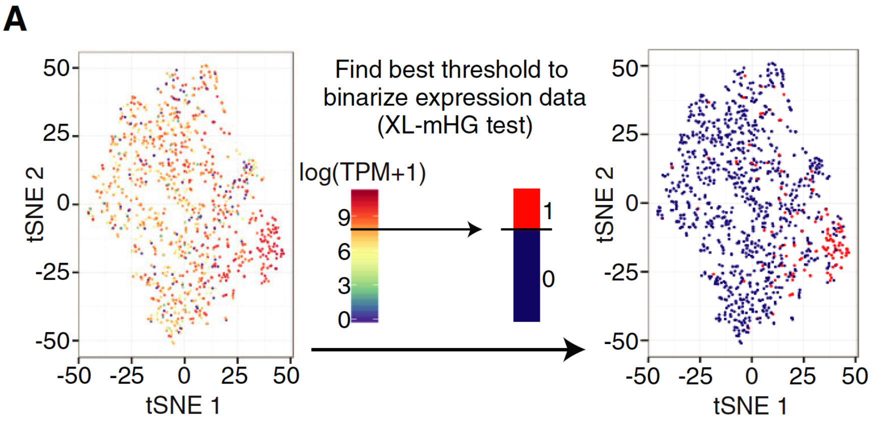


Figure 2.18: An illustration of the binarization procedure applied by COMET to each gene in a cluster-specific manner via the non-parametric XL-mHG test. For each gene, an expression threshold of maximal classification strength for the given cluster is annotated with the XL-mHG test.The XL-mHG P-value measures the significance of the chosen threshold index.This threshold index is then matched to an expression cutoff, which is used to binarize gene expression values.

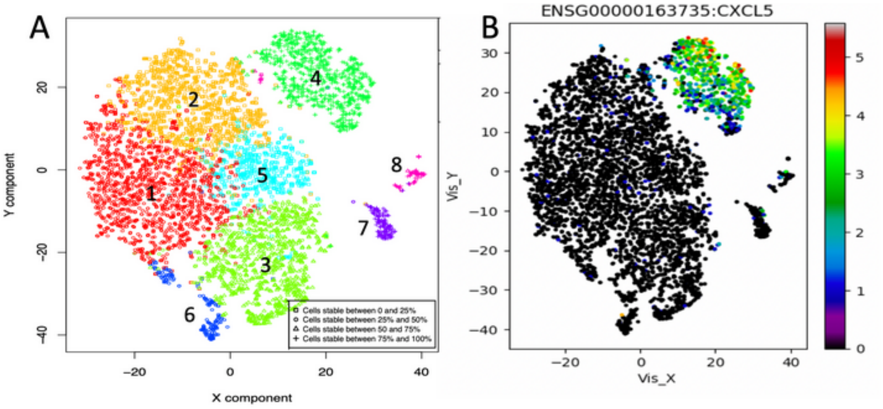


Figure 2.19: COMET genes prioritization. A) Seurat clustering for T-cell lymph node dataset B) Top ranked gene for cluster 4

2.6 Neural Network (NN) for Single cell data mining

The tools described above are able to detect cluster specific genes, however there is no indication of the importance of such genes in the molecular activities associated to a specific cluster. We decided to try to grasp the molecular activities underlying the transcriptome characterizing each cluster. Thus, we decided to see if neural network analysis could help in defining molecular features, which are specific of a cluster.

Artificial neural networks are computing systems inspired by the biological neural network that constitute animal brains. NN is made by three parts, fig. 2.20:

- input Layer : input Data.
- hidden Layer: a black box,"where the magic happen".
- output Layer : the result outputs. These results will be compared with the expected ones. On the basis of the level of convergence between output layer and expected results the middle layer is changed until the output layer looks like the expected results.

Neural networks are powerful tools, but they are affected by two main problems:

- NN cannot be used on small datasets
- The core of the NN architecture, the one that will 'understand' the problem, is a black box. So it is helpful to solve problems but it might not be useful to understand in which way the problems were solved.

Frequently, NN are used in bioinformatics as dimensionality reduction tools using a particular type of neural network called autoencoder [50].

The interesting features of scRNAseq dataset are the sparsity and the numerosity of samples, which makes scRNAseq a suitable input for NN analysis.

2.6.1 Autoencoders

Autoencoder is an unsupervised artificial neural network, which efficiently compresses and encodes data, then learns how to reconstruct the data back from this reduced encod-

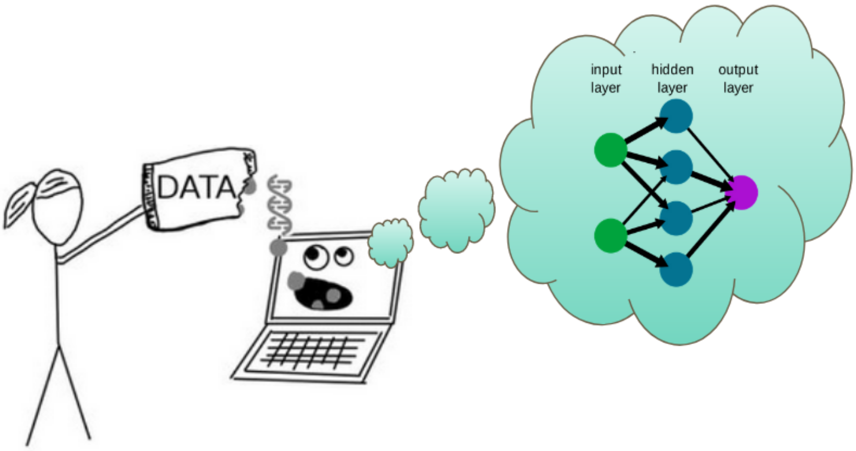


Figure 2.20: Example of NN. Laptop is feeded with DNA information and learning through the neural network architecture is performed.

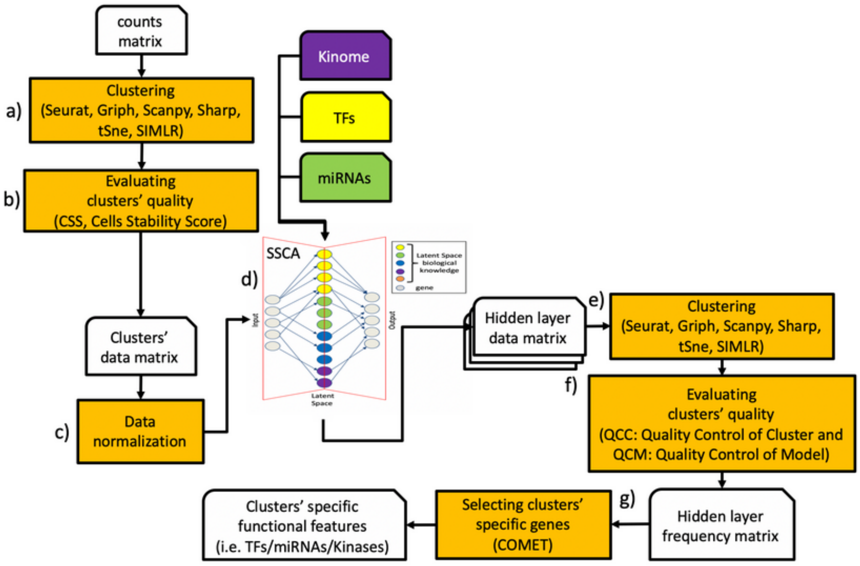

counts matrix
a)
Clustering (Seurat, Griph, Scanpy, Sharp, tSne, SIMLR)
b)
Evaluating clusters' quality (CSS, Cells Stability Score)
Clusters' data matrix
c)
Data normalization
Kinome
TFs
miRNAs
SSCA
d)
Input
Output
Latent Space
Latent Space biological knowledge
gene
Hidden layer data matrix
e)
Clustering (Seurat, Griph, Scanpy, Sharp, tSne, SIMLR)
f)
Evaluating clusters' quality (QCC: Quality Control of Cluster and QCM: Quality Control of Model)
g)
Hidden layer frequency matrix
Selecting clusters' specific genes (COMET)
Clusters' specific functional features (i.e. TFs/miRNAs/Kinases)

Figure 2.21: SCA workflow in rCASC. Clusters defined by any of the clustering tools implemented in rCASC are used for feeding SCA. SCA is made of experimentally validated information as transcription factors target genes, miRNA target genes etc. SCA is used to see if the biological knowledge implemented in the latent space is able to reconstruct any of the input clusters. Then, cluster specific key features are extracted from the latent space.

ing to produce an output that is as close as possible to the original input. Autoencoder reduces data dimensions by learning how to ignore the noise in the data. Autoencoder-based approaches have been used to cluster single cell data [51], to impute single cell data [52], data denoising [53], batch correction [54]. Recently, Gold and co-workers [55] have evaluated the use of autoencoders for data interpretation, implementing sparsely-connected autoencoder (SCA) to gene set analysis. In rCASC we have implemented SCA to mine the biological content of cells subpopulations generated by clustering, fig. 2.21.

2.6.2 Sparse Autoencoders

A SCA encoding/decoding function

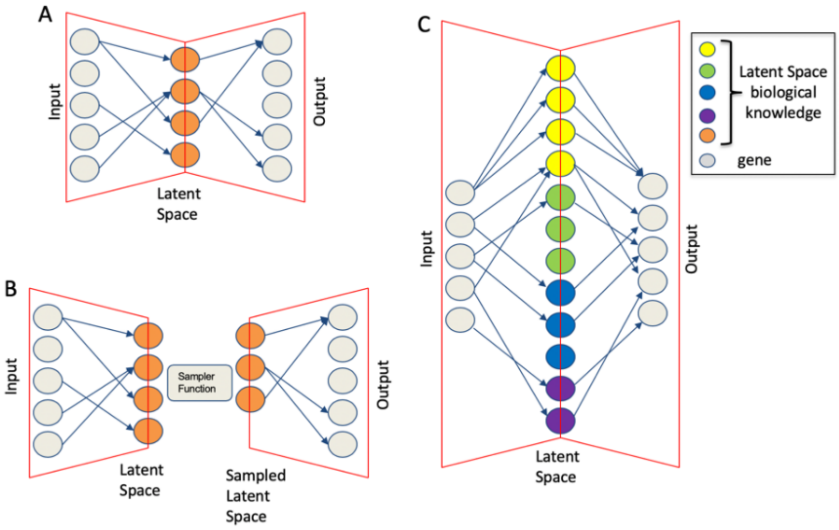


Figure 2.22: Autoencoders architecture. A) Sparsely-Connected Autoencoders (SCA), B) variational Sparsely-Connected Autoencoders (vSCA), C) Sparse Sparsely-Connected Autoencoders (SSCA). Grey nodes refer to genes. Gene-level expression profile for each gene in each cell are used as input and reconstructed as output on the basis of the latent space. The latent space is made of nodes where each node is associated to a transcription factor, a miRNA, a kinase or a functional signature or other biological knowledge. The vertices connecting input/output nodes to latent space are based on experimentally validated biological knowledge.

consists of a single sparse layer (fig. 2.22, latent space), with connections based on known biological relationships[[55], [56]]. Each node represents a known biological relationship, e.g. transcription factors (TFs) target, miRNA target (miRNAs), cancer-related immune-signatures (ISs), kinase specific protein targets (Ks), etc. and only receives inputs from gene nodes associated with the biological relationship. With respect to the Gold paper [55], which uses gene sets [57], in our implementation the latent space is based on experimentally validated data, TRRUST [58], miRTarBase [59], RegPhos [60], and a manually curated cancer-based immune-signature. Cancer immune-signature was manually curated, retrieving genes ids from PUBMED articles related to the keyword "cancer immune signature" and genes derived from KEGG "Immune system" pathways [61] . Genes associated to KEGG pathways were manually extracted from KEGG pathways public repository [62] . SSCA latent space is made by the union of TFs, miRNAs, IS and Kinases data.

In rCASC, SCA analysis is executed multiple times on a cell dataset, previously partitioned in clusters, using any of the clustering tools implemented in rCASC: k-mean+t-Sne [18], SIMLR [19], griph [25], scanpy [29] and SHARP [31] . Cell Stability Score [35] is used on SCA outputs to generate two quality scores metrics: QCC (Quality Control for Clusters) and QCM (Quality Control of Model). QCC is generated comparing cell partitions generated by rCASC clustering, i.e. the reference clusters, with the clusters generated using the latent space data after each SCA run. This metric measures the efficacy of the latent space in describing cells aggregations corresponding to at least part of the reference clusters. Specifically, QCC measures the frequency by which cells, belonging to a reference cluster, are found to be part of the same cluster in multiple SCA runs. QCC ranges from 0 to 1, where 0 indicates total lack of correspondence between a reference cluster and the corresponding SCA cluster over multiple runs of SCA. QCC equal to 1 instead indicates that cells, being part of a reference cluster are detected as part of the same cluster in all runs of SCA analysis. We suggest as threshold for QCC mean a value ≥ 0.5 , which indicates that at least in 50% of SCA runs a latent space cluster retains the cell content of the corresponding reference cluster. Instead, QCM measures cluster consistency between SCA runs. This metric is designed to evaluate the reproducibility of the model defined by SCA latent space. Specifically, if a set of biological information describing the latent space is important for the definition of a cluster, it is expected that the majority of the SCA runs will converge to a similar solution for that cluster. Thus, comparison, via QCM, of random couples of clusters selected over multiple SCA runs must show a conserved cluster(s) organization, fig. 2.23.

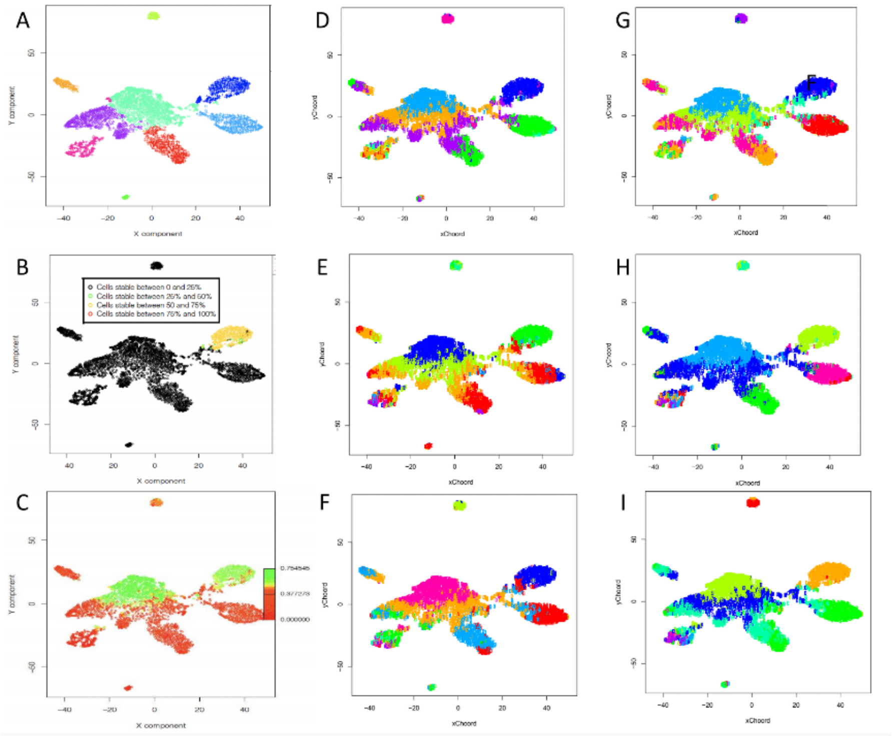


Figure 2.23: Exemplary description QCC and QCM calculation. A) setA clustered with SIMLR. B) QCC, each cell of setA (A) is compared against the different runs of SCA to generate a stability score for each cell. i.e. A vs D, A vs E, A vs F, A vs G, A vs H, A vs I. C) QCM, each cell of a randomly chosen latent space (D-I) is compared against another randomly chosen latent space (D-I), e.g. D vs I, E vs D, G vs D, H vs E, I vs F, F vs G. D-I, stability range is shown on the side of the picture. D-I) 6 SCA runs of TF based latent space.

QCM ranges from 0 to 1, where 0 indicates that, in any pairs of SCA runs comparisons, there is a total lack of correspondence between the cells content of a cluster detected in a SCA run compared to the corresponding cluster in another randomly selected SCA run. Instead, QCM equal to 1 indicates that cells, being part of a SCA cluster, are always detected as part of the same cluster in any pairs of SCA runs comparisons. We suggest as threshold for QCM mean a value ≥ 0.5, which indicates that at least 50% of SCA runs retain the structure of the cell content in any pair of SCA runs comparison. Thus, a reference cluster explainable by SCA analysis should be characterized by both QCC and QCM ≥ 0.5. As result of multiple runs of SCA, fig. 2.24 , it is possible to build a frequency matrix for the latent space representations. This frequency matrix is used to detect the latent space nodes, which are the most important for a cluster characterized by QCM and QCC means ≥ 0.5. The matrix describing the frequency of the latent space variables is used to extract cluster specific signatures using COMET tool [63].

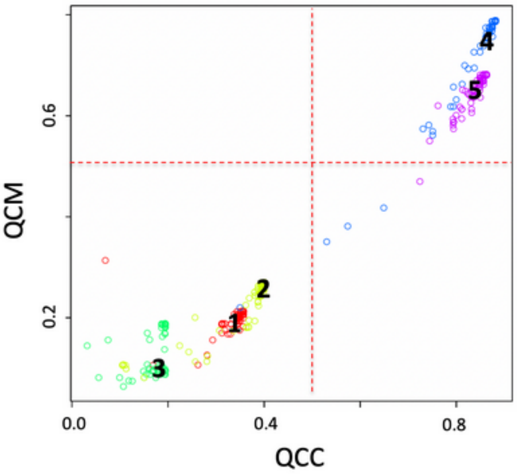


Figure 2.24: SCA analysis using a manually curated cancer-immune-signature (IS) on setA dataset. Input counts table for SCA is setA, log_{10} transformed. 160 permutations were run and latent space clustering was done with SIMLR. Numbers are related to the cluster number assigned with previous analysis made on setA.

2.6.3 Validation of SCA analysis on a PBMC derived dataset (setA)

We used a data set (setA), based on FACS purified cell types [64], to investigate the SCA (fig. 2.22A) performance. We chose this dataset for technical reasons, since these cells are FACS separated and they are clean and easily separable. SetA was previously used to estimate the strength of CSS metric [35]. We tested a SCA embedding a TFs-based latent space, where each latent space node was associated with a TF and arches connecting input and output nodes to each latent space node represented experimentally validated TF target genes from TRRUST database [58]. From this analysis, we observed that only cluster 1 and 2, fig. 2.25A,

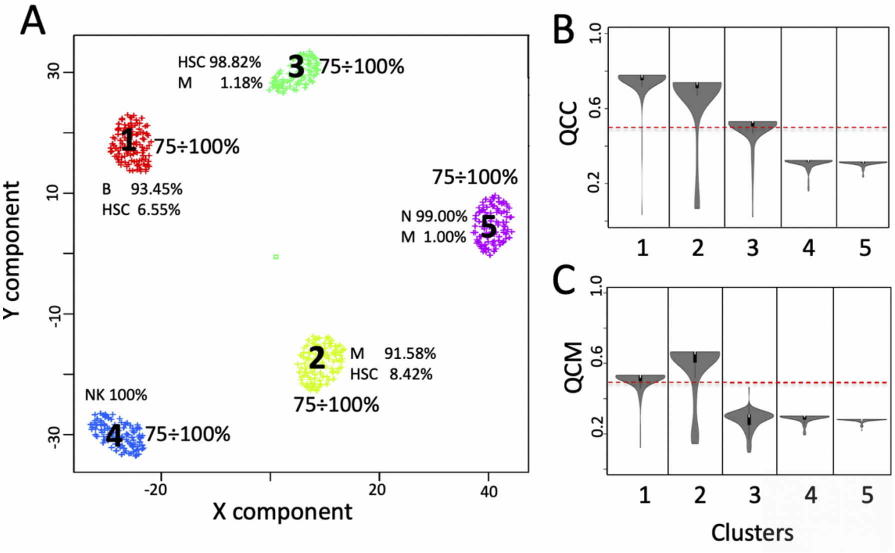


Figure 2.25: SCA analysis using a TF-based latent space. A) Five clusters were detected analysing setA with griph using log_{10} transformed counts table. Each cluster is made by more than 90% by one cell type. A little amount of HSC is contaminating B cells, monocytes and naïve T cells. Latent space clustering was done with SIMLR B) QCC violin plot. The metric is an extension of CSS and it measures the ability of latent space to keep aggregated cells belonging to predefined clusters, i.e. those in panel A. C) QCM violin plot, this metric is also an extension of CSS and it measures the ability of the neural network to generate consistent data over multiple SCA runs. Dashed red line indicates the defined threshold to consider the latent space information suitable to support cells' clusters. Input counts table for SCA analysis is log_{10} transformed. Monocytes (M), natural killer (NK), B-cells (B), Naive T cells (N), Hematopoietic stem cells (HSC).

could be reconstructed by this type of SCA, since only these two clusters were supported by a QCC and QCM greater then 0.5 (fig. 2.25B, C). The matrix describing the frequency of the latent space variables was used to extract cluster specific signatures using COMET tool, which is also implemented in rCASC. The optimal maker panel definition for cluster 1 was made by four genes PAX5, NFAT5, RFXANK and

CHD4, fig. 2.26A. The four genes detected by COMET for cluster 1, which was composed mainly by B-cells, recapitulated very well some of the key elements that have been shown already to be involved in development and differentiation of this cell type. The transcription factor PAX5 is essential for the commitment of lymphoid progenitors to B-lymphocyte lineage [65]: PAX5 fulfills a dual role by repressing B lineage 'inappropriate' genes and simultaneously activating B lineage-specific genes [65]. NFAT5 is important for optimal antibody productivity [25]. CHD4 is critical for early B-cell 24development [66] and RFXANK is involved in activation of MHC-II genes, which in turn MHC-II molecules are largely restricted to thymic epithelial cells and professional

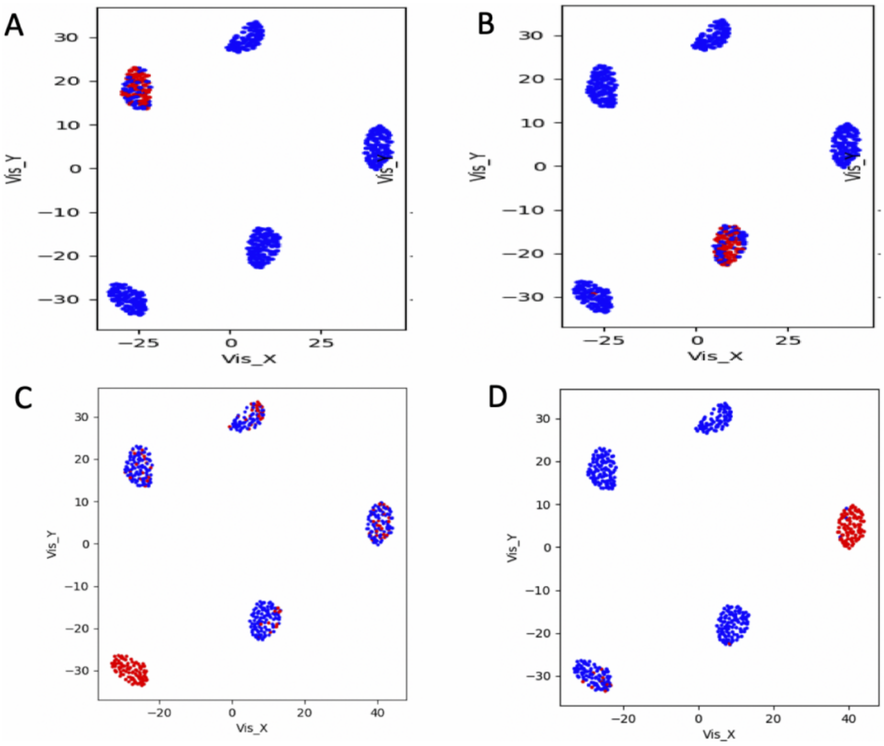


Figure 2.26: COMET analysis of latent space frequency matrix. A) Set of 4 genes (PAX5, NFAT5, RFXANK and CHD4) characterizing cluster 1 (B-cells). B) Set of 4 genes (CEBPA, KHSRP negation, CEBPB, CREBBP) characterizing cluster 2 (Monocytes). Dots are cells, blue and red indicates the expression of the gene of interest below significant threshold and above respectively. C) Rank 1 NK signature [32] specifically characterizing cluster 4 (NK). D) Rank 2 Asthma KEGG negation, specifically characterizing Naive T-cells (Cluster 5).

antigen-presenting cells, including dendritic cells, macrophages, and B-cells. Moreover, for cluster 2, which was mainly composed by monocytes, the best maker panel was made of four genes, CEBPA, KHSRP (lack of expression), CEBPB, CREBBP, fig. 2.26B, which have been shown already to be strongly involved in monocyte functionalities. The transcription factor CCAAT/enhancer-binding protein β (CEBPB) is highly expressed in monocytes/macrophages and is a critical factor for Ly6C-monocyte survival [63]. The downregulated expression of the KH-Type Splicing Regulatory Protein (KHSRP) during monocytopoiesis and its up-regulated expression during granulopoiesis suggested that KHSRP has divergent roles during monocytic and granulocytic differentiation [67]. CREB is involved in anti-apoptotic survival signaling in monocytes and macrophages [68] and CREBBP specifically binds to the active phosphorylated form of CREB.

A SCA analysis based on a latent space made of manually curated cancer-immune-signatures was also performed on setA reference clusters, fig. 2.24. This SCA analysis showed, for cluster 4 (Natural Killer cells) and cluster 5 (Naive T-cells), QCM/QCC values greater than 0.5 for the majority of the cells. Analyzing the IS-based latent space frequency table with COMET, we identified one feature of the immune-signature derived from Nirmal's paper, which was characteristic of NK (fig. 2.26C) and a KEGG immune-signature, which expression absence was specific for Naive T-cells cluster (fig. 2.26D). SCA analysis on setA was also done using a latent space based on kinase targets, but we cannot find any robust association with reference clusters (not shown).

2.6.4 Investigating the effect of normalization on SCA latent space frequency matrix on QCC/QCM scores- SetA

SCA analysis based on validated target genes for miRNAs (fig. 2.27A) showed that clusters 2 (Monocytes) and 3 (Hematopoietic Stem Cells) had a potentially interesting trend, although they were not supported by a QCM and a QCC greater then 0.5. Consequently, we investigated the effect of different normalization procedures of the SCA input counts table on the modelled results, to see if normalization of SCA input data could help in improving QCM and a QCC scores. In fig. 2.27, it is shown how normalization affected

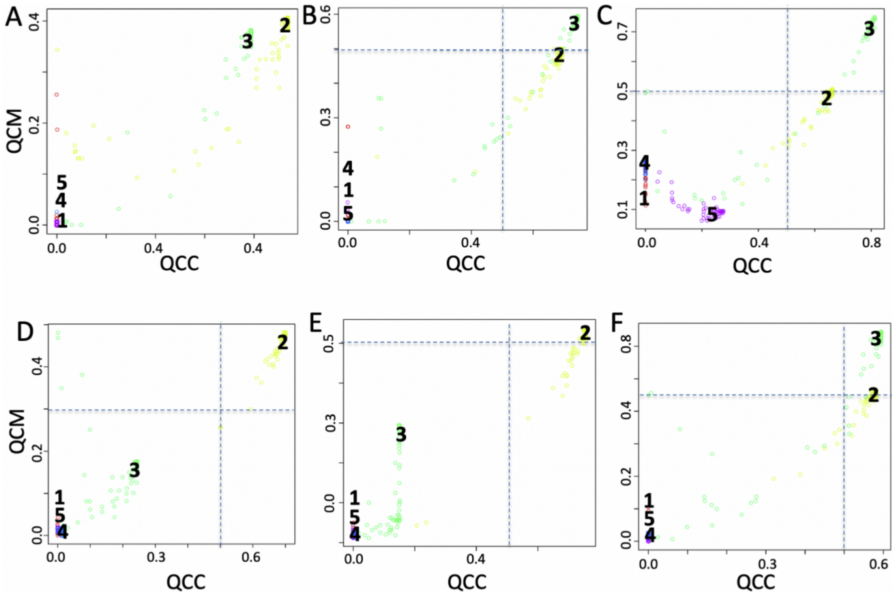


Figure 2.27: QCM/QCC plots using different normalizations for the SCA input counts table. A) Log_{10} transformed, B) Centered log-ratio normalization (CLR), C) relative log-expression (RLE), D) full-quantile normalization (FQ), E) sum scaling normalization (SUM), F) weighted trimmed mean of M-values (TMM).

both QCM and QCC scores for clusters closed to the suggested significant threshold (0.5), i.e. cluster 3 (fig. 2.27, C, F) cluster 2 (fig. 2.27D, E). Low quality clusters, i.e. those lacking robust latent space information, were minimally affected by normalization, i.e. clusters 1, 4 and 5. This observation suggested that it is important to assess the effect of different normalization procedures on SCA input data, specifically if SCA clusters show QCM and QCC scores near to the significant threshold (0.5) suggested for those metrics. Analyzing the latent space of miRNAs frequency table with COMET we identified miR-191 as top marker for cluster 3 (HSC) and ranked 1, 5 and 205 markers using CLR, RLE and TMM normalizations, respectively. miR-191 has been already associated with the appearance of stem cell-like phenotype in liver epithelial cells. Using TMM normalization for cluster 3, rank 1 marker was miR-132-3p, which has been linked to HSC maintenance. miR-187-3p was detected as rank 1 marker for cluster 2 (M) with FQ normalization and as rank 4 marker in SUM normalization, respectively. miR-187 has been demonstrated to play a central role in the physiological regulation of IL-10-driven anti-inflammatory responses in TLR4-stimulated monocytes. It was notable that different normalizations retained a certain amount of consistency in the top ranked markers.

2.6.5 Validation of variational Sparsely Connected Autoencoders (vSCA) analysis on a PBMC derived dataset (setA)

A variational autoencoder (VAE) consists of an encoder, a decoder, and a loss function. VAEs have one fundamentally unique property that separates them from other autoencoders: their latent spaces are, by design, continuous, allowing easy random sampling and interpolation. We applied the concept of VAE to SCA (vSCA, fig. 2.22B). We tested a vSCA based on TF-targets using setA (fig. 2.28D). From the point of view of the QCM/QCC of clusters vSCA results were nearly superimposable to those of a TF-based SCA fig. 2.28. The analysis of COMET for cluster 1 and cluster 2 detected as the best markers respectively CHD4 and CEBPA, which were part of the 4 genes SCA signature for cluster 1 and 2. Taken together, these observations indicated that vSCA, although more complex, did not provide any specific improvement with respect to a simple SCA.

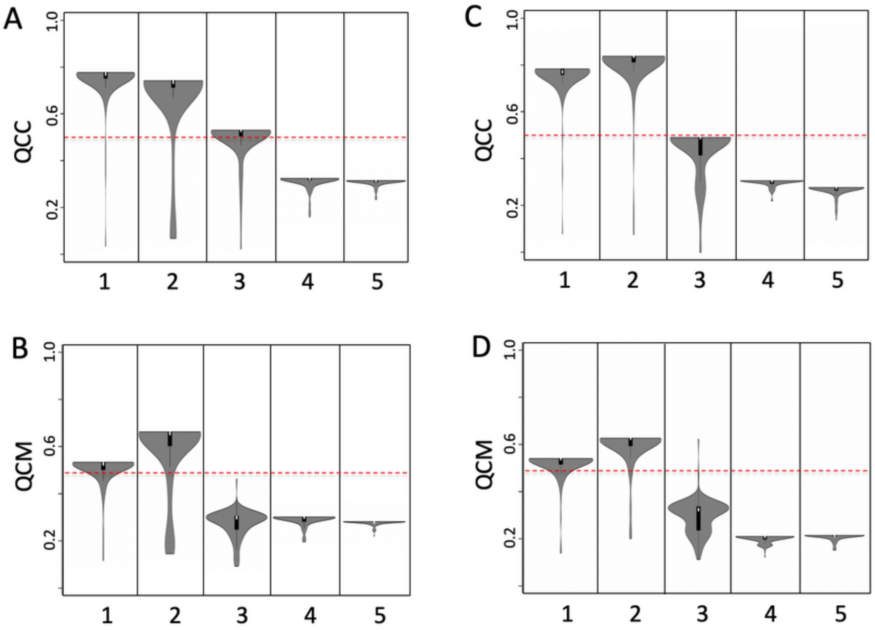


Figure 2.28: vSCA based on TFs targets. A) QCC TFs based SCA, B) QCM TFs based SCA, C) QCC TFs based vSCA, D) QCM TFs based vSCA . Input counts table for SCA is log_{10} transformed, analysis was performed using 160 permutations and latent space clustering was done with SIMLR.

2.6.6 Validation of Sparse Sparsely Connected Autoencoders (SSCA) analysis on a PBMC derived dataset (setA)

Sparse autoencoder may include more hidden units than inputs, although only a small number of the hidden units are allowed to be active at once. This sparsity can improve classification performance. Usually, the sparsity is possible thanks to combinations of activation functions, sampling steps and different kinds of penalties [63]. In our implementation sparsity was generated combining TF, miRNA, IS and kinase SCA latent spaces. The addition of miRNA targets to TF latent space (fig. 2.29B) repositioned cluster 3 (HSC) inthe significant area, i.e. mean of both QCM and QCC greater then 0.5, which could not be possible using only miRNA targets as latent space (fig. 2.27A). For cluster 3 the best marker was miR-129-2-3P. Interestingly, miR-132,detected as top

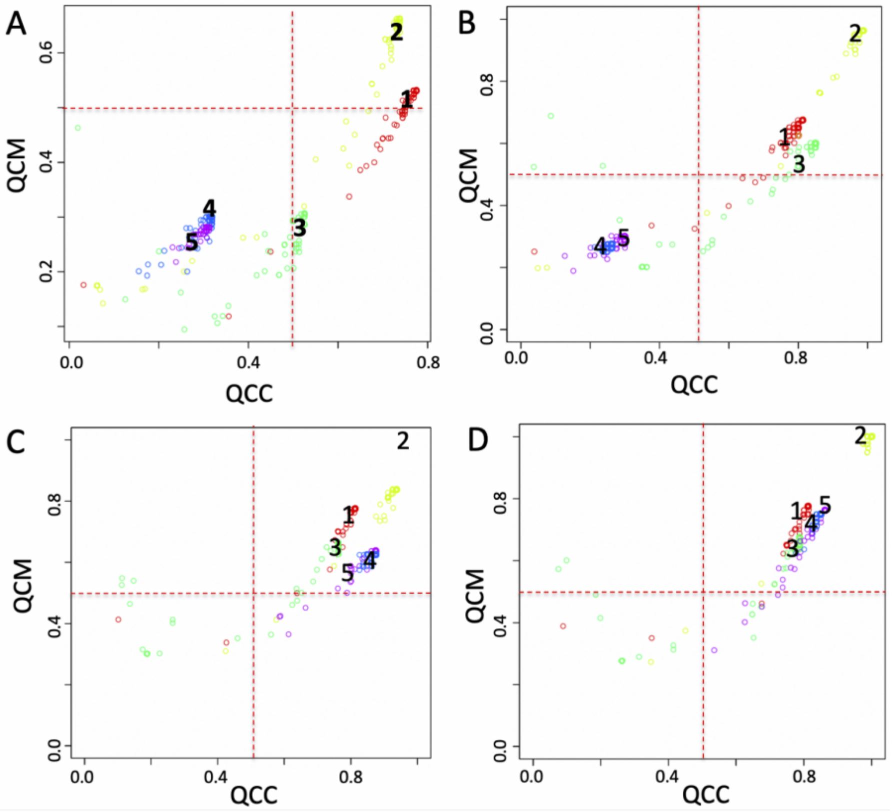


Figure 2.29: QCM/QCC plots for SSCA. A) TF latent space. B) TF + miRNA latent space, C) TF + miRNA + IS latent space, D) TF + miRNA + IS + Kinase latent space. Input counts table for SCA is log_{10} transformed.

ranked for cluster 3 upon TMM normalization (fig. 2.27F). This miRNA together with miR-129, detected only from SSCA analysis, were distinct faces of the same coin, since miR-132 has been already shown to be linked to HSC maintenance and miR-129 was found to be associated to self-renewal and lineage differentiation of stem cells [69]. The addition of the IS to TF + miRNA latent space relocated in the significant area cluster 4 (NK cells) and 5 (naïve T-cells) (fig. 2.29D). For cluster 4, the first top ranked item was NK signature. Instead, for cluster 5 (Naive T cells) the best marker was of the expression absence of transcription factor POU2F2, which is expressed only in activated T cells. ASTHMA KEGG pathway lacks of expression was also detected using IS-based SCA alone. Notably, the Kinase based latent space alone was not able to bring any cluster in the significant area (not shown). However, when Kinases are added to TF + miRNA + IS latent space (fig. 2.29C), the QCM and QCC scores for cluster 4 and 5 improved slightly, but kinases were not present in the top ranked genes for cluster 4 and 5. The analysis of the setA using a latent space embedding integrated biological knowledge, i.e. TF + miRNA (fig. 2.29B); TF + miRNA + IS (fig. 2.29C); TF + miRNA + IS + Kinase (fig. 2.29D), showed a notable improvement in the overall QCM/QCC scores with respect to the analysis done using each specific knowledge group alone. Furthermore, the combined latent space reduced the computing time with respect to the time needed for the independent analysis of each individual latent space.

2.6.7 Application of SCA analysis on spatially resolved transcriptomics of breast cancer histological section.

As example of different application of SCA analysis, we analyzed a breast cancer histological section available as part of the demo data proposed for visium, i.e. spatially resolved transcriptomics. We obtained the best clustering organization of expression data using SIMLR (fig. 2.30A),which generated a partition made of 9 clusters, where 6 clusters (1, 2, 3, 6, 7, 9) showed very high CSS (fig. 2.30B), while other two clusters (5 and 8) showed an intermediate but still significant CSS (fig. 2.30B). Clusters were then localized on the histological session (fig. 2.30C, D). Despite the low magnification

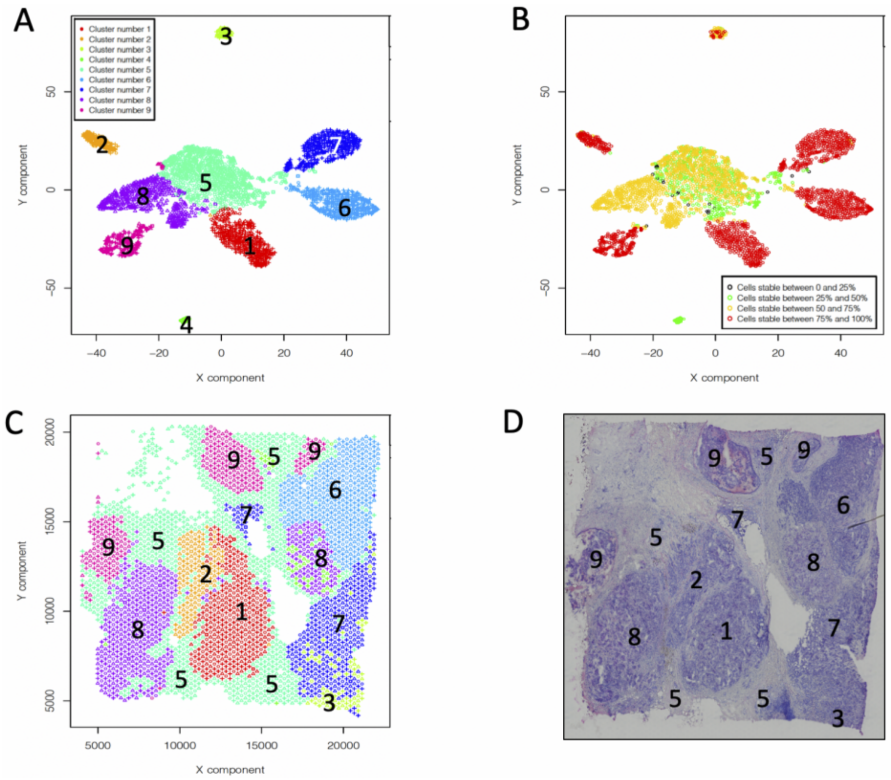


Figure 2.30: Analysis of human breast cancer, from 10XGenomics Visium Spatial Gene Expression 1.0.0. demonstration samples. A) SIMLR partitioning in 9 clusters. B) Cell stability score plot for SIMLR clusters in A. C) SIMLR clusters location in the tissue section. D) Hematoxylin and eosin image.

of histological picture (fig. 2.30D) obtained from the frozen section, the pathologists were able to assign cluster 5 to areas predominantly corresponding to tumor stroma; cluster 9 to ductal carcinoma in situ with micropapillary features; clusters 1 and 8 corresponded to roundish areas annotated as invasive carcinoma, showing the same dyeability and possibly histological similarity (micropapillae) with areas associated with cluster 9. Clusters 6 and 7 were allocated to invasive carcinomas with comparable features at low magnification (smaller cell clusters infiltrating the stroma). Cluster 3 and 4 could not be classified by the pathologists, because of the limited number of cells. We tested the ability of SCA to associate TFs with the detected clusters and only cluster 7 could be described by SCA analysis fig. 2.31.

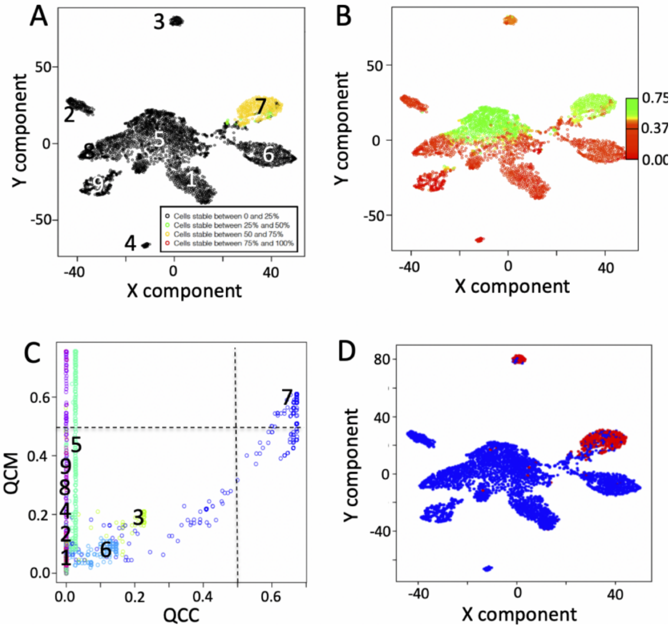


Figure 2.31: Information contents extracted by SCA analysis using a TF-based latent space. A) QCC. B) QCM. C) QCM/QCC plot, where only cluster 7 shows, for the majority of the cells, both QCC and QCM greater than 0.5. D) COMET analysis of SCA latent space. SOX5 was detected as first top ranked gene specific for cluster 7, using as input for COMET the latent space frequency table. Input counts table for SCA analysis is made by raw counts.

COMET analysis of the latent space frequency table provided the detection of SOX5

(COMETsc statistics = 2.88E-184, TP = 0.623 and TN = 0.968) as top ranked transcription factor specific for cluster 7 (fig. 2.31D). Notably, SOX5 has been recently associated with breast cancer proliferation and invasion [39], suggesting a peculiar aggressive phenotype for the invasive carcinoma associated with cluster 7. We also tested miRNA, immune signature and kinase based SCA, but we could not find any robust association with reference clusters (not shown). These observations suggested that the knowledge present in SCA based on miRNA, immune signature and kinase targets were not sufficient to describe the complexity of tumors clusters observed in this specific dataset. At the same time the results obtained for cluster 7, using the TF-based latent space, highlighted the ability of SCA to grasp specific biological knowledge associated with transcription control in this experimental setting.

2.7 GUI and cloud implementation

To simplify the use of the rCASC package for users without scripting experience, R functions can be controlled by a dedicated GUI, integrated in the 4SeqGUI tool [70] fig. 2.32.

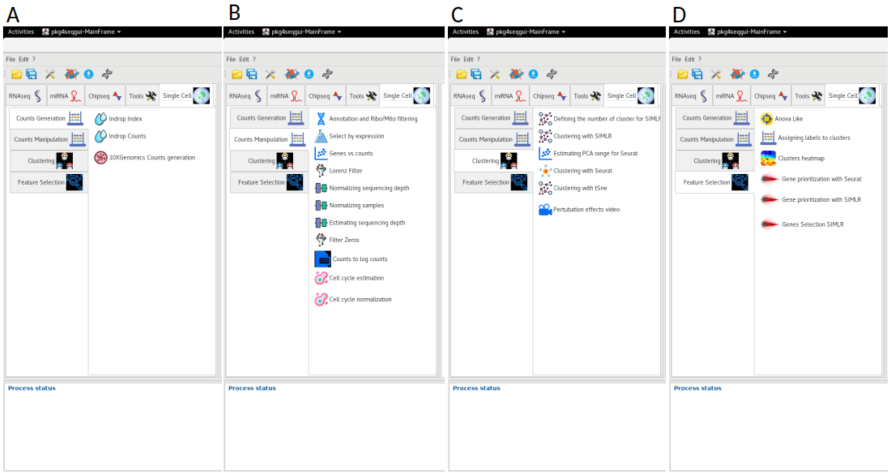


Figure 2.32: rCASC graphical interface within 4seqGUI. A) Count table generation menu: this set of functions is devoted to the conversion of fastq to a count table. B) Count table manipulation menu: this set of functions provides inspection, filtering, and normalization of the count table. C) Clustering menu: these functions allow the use of SIMLR, K-mean + t-SNE, Seurat, griph, scanpy and SHARP, to group cells in subpopulations. D) Feature selection menu: this set of functions allows the identification of cluster-specific subsets of genes and their visualization using heatmaps

2.7.1 Galaxy, Laniakea implementation

Laniakea is a complete software solution to set up a ”Galaxy on-demand” platform as a service. Building on the INDIGO-DataCloud software stack, Laniakea can be deployed over common cloud architectures usually supported both by public and private e-infrastructures. The user interacts with a Laniakea-based service through a simple front-end that allows a general setup of a Galaxy instance, and then Laniakea takes care of the automatic deployment of the virtual hardware and the software components [36]. rCASC workflow is also implemented in Laniakea. The full set of rCASC tools are implemented in Galaxy and they can be combined in a user-specific workflow. There are also some workflows ready to be used, an example is shown in fig. 2.33. Galaxy output for rCASC can be saved as a tar file or they can be visualized in html, fig. 2.34

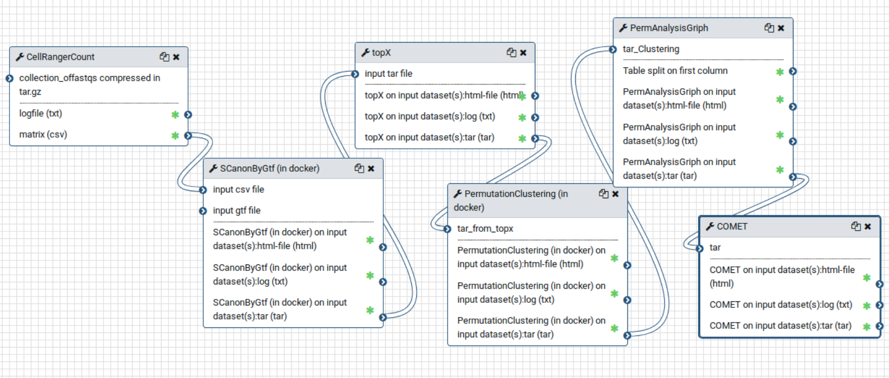
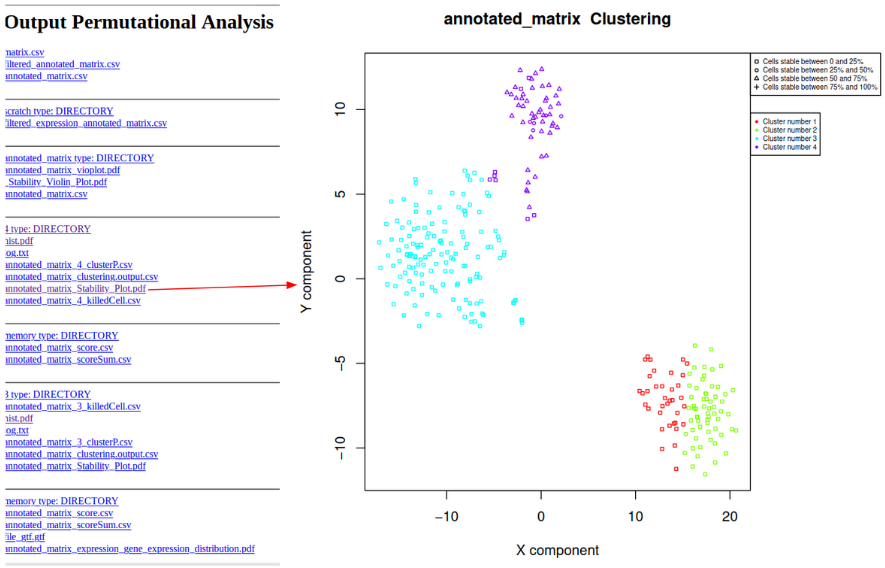

CellRangerCount
collection_offastqs compressed in tar.gz
logfile (txt)
matrix (csv)
SCanonByGtf (in docker)
input csv file
input gtf file
SCanonByGtf (in docker) on input dataset(s):html-file (html)
SCanonByGtf (in docker) on input dataset(s):log (txt)
SCanonByGtf (in docker) on input dataset(s):tar (tar)
topX
input tar file
topX on input dataset(s):html-file (html)
topX on input dataset(s):log (txt)
topX on input dataset(s):tar (tar)
PermutationClustering (in docker)
tar_from_topx
PermutationClustering (in docker) on input dataset(s):html-file (html)
PermutationClustering (in docker) on input dataset(s):log (txt)
PermutationClustering (in docker) on input dataset(s):tar (tar)
PermAnalysisGriph
tar_Clustering
Table split on first column
PermAnalysisGriph on input dataset(s):html-file (html)
PermAnalysisGriph on input dataset(s):log (txt)
PermAnalysisGriph on input dataset(s):tar (tar)
COMET
tar
COMET on input dataset(s):html-file (html)
COMET on input dataset(s):log (txt)
COMET on input dataset(s):tar (tar)

Figure 2.33: rCASC on Galaxy example workflow in Galaxy.


Output Permutational Analysis

natrix.csv
iltered_annotated_matrix.csv
nnotated_matrix.csv

cratch type: DIRECTORY
iltered_expression_annotated_matrix.csv

nnotated_matrix type: DIRECTORY
nnotated_matrix_vioplot.pdf
_Stability_Violin_Plot.pdf
nnotated_matrix.csv

4 type: DIRECTORY
hist.pdf
og.txt
nnotated_matrix_4_clusterP.csv
nnotated_matrix_clustering.output.csv
nnotated_matrix_Stability_Plot.pdf
nnotated_matrix_4_killedCell.csv

nemory type: DIRECTORY
nnotated_matrix_score.csv
nnotated_matrix_scoreSum.csv

3 type: DIRECTORY
nnotated_matrix_3_killedCell.csv
hist.pdf
og.txt
nnotated_matrix_3_clusterP.csv
nnotated_matrix_clustering.output.csv
nnotated_matrix_Stability_Plot.pdf

nemory type: DIRECTORY
nnotated_matrix_score.csv
nnotated_matrix_scoreSum.csv
ile_gtf.gtf
nnotated_matrix_expression_gene_expression_distribution.pdf

annotated_matrix Clustering
Y component
X component
10
5
0
−5
−10
−10
0
10
20
□ Cells stable between 0 and 25%
○ Cells stable between 25% and 50%
△ Cells stable between 50 and 75%
+ Cells stable between 75% and 100%
🟥 Cluster number 1
🟩 Cluster number 2
● Cluster number 3
● Cluster number 4

Figure 2.34: Example of rCASC output visualization in Galaxy

2.7.2 Single cell ATAC-seq

My interest in NN started when I was visiting fellow at Stein Aerts's lab for six months during the second year of the PhD program. Stein's lab is part of the KU Leuven Cen-

ter for Human Genetics and the VIB Center for Brain and Disease Research. Stein's main project is focused in decoding the genomic regulatory code and understanding how genomic regulatory programs drive dynamic changes in cellular states, both in normal and disease processes. Transcriptional states emerge from complex gene regulatory networks. The nodes in these networks are cis-regulatory regions such as enhancers and promoters, where usually multiple transcription factors bind to regulate the expression of their target genes. The aim of my work was the prediction of gene expression using ATAC-seq data generated from Drosophila eye disk. ScRNAseq for Drosophila eye disk were also available. Although ATAC-seq and RNA-seq were generated from independent experiments, Stein's group linked them together using biological features (unpublished results). Although I tried multiple NN configurations I always failed in predicting a quantitative gene expression only using ATAC-seq data. Then, I tried to predict if a gene was simply in on or off state. Thus, RNA-seq matrix was binarized in the following way :
let be S the single cell RNA-seq matrix, i the cell and j the gene :

$$S_{ij} > 3- > S_{ij} = 1 \tag{2.1}$$

$$S_{ij} <= 3- > S_{ij} = 1 \tag{2.2}$$

For this analysis I used a NN made by 5 dense layers with the respective dropout layer and imputed with binarized genes fig. 2.35.
NN hyperparameters used are : $LearningRate = 0.001$, $LossFunction =$ Mean squared error, $nEpochs = 1000$, $Optimizer =$ Adam with accuracy as metrics, $Batchsize = 64$.

2.7.2.1 Prediction algorithm

The ATAC-seq matrix was divided in regions of variable size depending from the continued accessible part of the chromatin. The ATAC-seq matrix was used without normalization, but rescaled to a window around the studied gene, fig. 2.36. The window

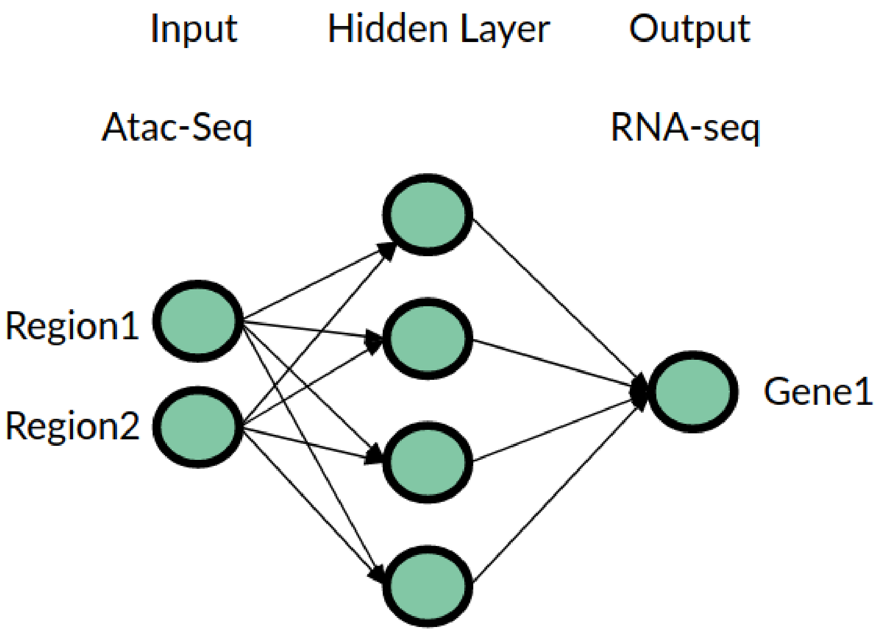


Figure 2.35: Neural Network Architecture to predict gene expression from ATAC-seq

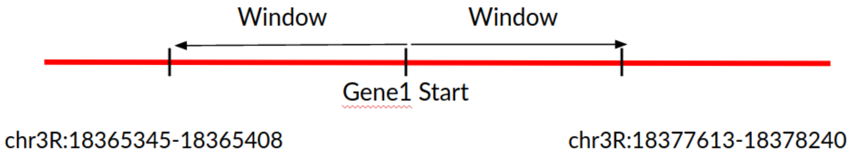


Figure 2.36: ATAC-seq window, the example is based on gl Gene using window=7, where the window indicates the number of ATAC regions surrounding the gene promoter position.

represents the number of open chromatin regions detected by ATAC-seq, independently by their length and located upstream and downstream the gene promoter. Specifically window=7 means that the ATAC-seq taken in account is made by the 14 regions around the gene promoter. Test set and train set were created sampling N ATAC-cells, where N correspond to the number of RNA-seq Cells. In fig. 2.37B, it is given an example of the prediction provided by my NN using virtual spatial information from [71]. The gene was predicted, compared with the real expression, fig. 2.37A.

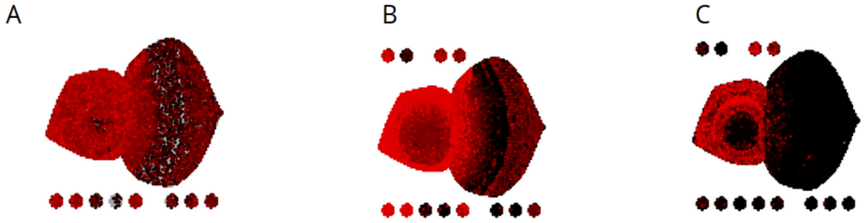


Figure 2.37: Example of expression prediction. A) hth mrna expression by SC-Rna seq. B) Predicted expression of Hth gene.C) Null prediction by random shuffling of the ATAC regions computed multiple times.

3 Conclusions

> Sometimes I'll start a sentence, and I don't even know where it's going. I just hope I find it along the way. Like an improv conversation. An improversation.
>
> *Michael- The office*

When we started the design of rCASC we wanted to create a tool that could provide some useful tips to the final users. We wanted to build a robust tool providing computational and functional reproducibility being user-friendly and easy to install. After three years of work, I think we managed to realize what we have planned. The problem of reproducibility is an important issue in bioinformatics [72] and we built rCASC as part of the Reproducible Bioinformatics Project, a community project, which provides a controlled but flexible workspace to distribute Docker based workflows under the umbrella of a reproducibility framework. The implementation in docker containers of the rCASC tools makes the overall installation of the package also extremely easy for the final user, for more information see rCASC web page: https://github.com/kendomaniac/rCASC. rCASC does not simply provides an integration of previously published tools, but, with the definition of the cell stability score (CSS), it provides an innovative approach to quantify the stability and the purity of a cluster. Furthermore, the implementation of sparsely-connected-autoencoders rCASC provides an effective instrument to extract important biological features characterizing the cell subpopulation clusters. The seminal paper of Gold and coworkers [55] proposed SCA as a promising tool for projecting gene-level data onto gene sets. Indeed, their results suggest that SCA can be efficiently exploited in the identification of transcription factors with differential activity between conditions or cell types. However, Gold and coworkers [8] did not release any implementation of these methods, and they did not report a well-defined scoring metric to evaluate the efficacy of SCA in grabbing biological information in scRNAseq experiments. Instead in rCASC, SCA is exploited to query cell subpopulations to discover the functional features (e.g. TFs, miRNAs, Kinases, etc.) driving cell clustering. We have shown that different hidden layers, derived by experimentally validated data (TF targets, miRNA targets, Kinase targets, and cancer-related immune signatures), can be used to grasp single cell cluster-specific functional features. Then, when SCA encoding is able to reconstruct at least one of the clusters, that means that the encoded biological knowledge is mandatory to obtain a specific aggregation of cells. In our implementation, SCA efficacy comes from its ability to reconstruct only specific clusters, thus indicating only those clusters where the SCA encoding space is a key element of a specific cell subgroup. A very important element of our SCA implementation is the availability of metrics estimating the robustness of the SCA encoding. ScRNAseq, although powerful, has the limit of being very noisy [73]. A particularly prominent aspect of noise is dropout, i.e. scRNA-seq produces more zeros than expected and this bias is greater for poorly expressed genes [74]. Transcription factors and kinases are encoded by genes characterized by a relatively low expression in cells, thus they can be notably affected by dropout. Furthermore, nowadays it is not possible to quantify, at single cell level, miRNAs together with mRNAs. Therefore, important functional networks, e.g. TF-miRNA circuits, characterizing a cell subpopulation, cannot be directly measured. In rCASC, SCA is able to grasp the hidden knowledge present in cell subpopulations. SCA offers a fresh view of regulatory genes that, because of scRNAseq noise, cannot be efficiently quantified, such as transcription factors and kinases, or not detected at all, i.e. miRNAs. In conclusion, we think that rCASC is a workflow with valuable new features that could help researchers in defining cells subpopulations as well as detecting sub-population specific markers, under the umbrella of data reproducibility.

4 Materials and Methods

> In learning you will teach, and in teaching you will learn
>
> *Phil Collins- Tarzan son of Man*

4.1 Availability and Requirements

Project name: rCASC: reproducible Classification Analysis of Single Cell sequencing data
Project page: https://github.com/kendomaniac/rCASC; https://github.com/mbeccuti/4SeqGUI
Operating system: Linux
Programming language: R and JAVA
Other Requirements: None
License: The GNU Lesser General Public License, version 3.0 (LGPL-3.0)
Any restrictions to use by non-academics: None

4.2 Datasets

Dataset setA, B, C, D are based on FACS purified cell types [64]. setA is made of 100 cells randomly selected from each cell types of the following cell types:

- B cell, 25K reads/cell
- Monocytes, 100K reads/cell

- Stem cells, 24.7K reads/cell
- Natural killer, 29K reads/cell
- Naive T cells, 19K reads/cell

.

setB is made of 100 cells randomly selected from each of the following cell types:

- B-cells
- Monocytes
- T-helper cells (21K reads/cell)
- Natural killer
- Naive T-cells

SetC is made of 100 cells randomly selected from each of the following cell types:

- Cytotoxic T-cells (28.6K reads/cell)
- Monocytes
- T-helper cells
- Natural Killer
- Naive T-cells

setD is made of 100 cells randomly selected from each of the following cell types:

- Cytotoxic T-cells
- Naive cytotoxic T-cells (20K reads/cell)
- T-helper cells
- Natural killer

- Naive T-cells

Moving from SetA to setD we added an increasing number of cells coming from T-cell populations, making the cell-type partitioning more challenging, because of the similarities between T-cell sub-populations. Breast cancer dataset derived from 10XGenomics spatial transcriptomics datasets resources. The filtered sparse matrix from 10XGenomics repository was transformed in a dense matrix using rCASC h5tocsv function. Dataset was annotated using ENSEMBL Homo sapiens GRCh38.99 GTF file using the rCASC scannobyGtf function. After annotation, ribosomal and mitochondrial protein genes were removed together with all ENSEMBL ID not belonging to protein coding ENSEMBL biotype. Cells with less than 250 detected genes were also removed (i.e. a gene is called detected if it is supported by at least 3 UMIs). After filtering rCASC topX function was used to select the 10000 most dispersed genes and from them the 5000 most expressed genes. The final matrix was made by 5000 genes and 3432 cells out of the initial 3813 cells. Data were log_{10} transformed before clustering.

Pace dataset is from Pace et al. paper [38], in which the role of histone methyltransferase Suv39h1 in murine $CD8+$ T cells activated after Listeria monocytogenes infection is explored. GSE106264 dataset is made of 10,035 cells and published by Pace and coworkers in 2018 [38] and the 10,000/33,000/68,000 cells. PBMC human datasets, available at 10xGenomics repository (www.10xgenomics.com).

Drosophila eye disk SCRNA-seq and ATACseq data are from Stein Aerts lab.

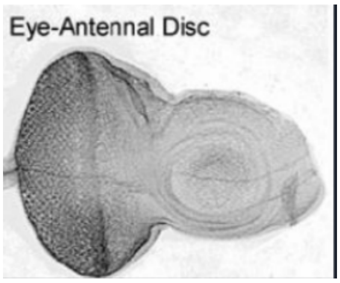


Figure 4.1: Drosophila Eye-Antennal disc. [75]

The Drosophila Eye disk cells dataset is made of [71]:

- 3531 Rna-seq cells
- 15387 ATAC-seq cells

The subpopulations identified in Drosophila eye-disk were:

- $PMmedial$
- $Glia$
- $PMFPRLate$
- $Hemocytes$
- $PMFPREarly$
- $PMFConeCells$
- $AntennaA1$
- $AntennaA3arita$
- $HeadVertex$
- $PMlateral$
- $MFMorphogeneticFurrow$
- $twiells$

4.3 Counts table generation

InDrop single-cell sequencing approach was originally published by Klein [76]. Then, the authors published the detailed protocol in Nature Methods in 2017 [77]. In rCASC, the generation of the count table starting from fastq files refers to the version 2 of the inDrop chemistry described in [77], which is commercially distributed by 1CellBio. The procedure described in the inDrop github is embedded in a docker image. rCASC function indropIndex allows the generation of the transcripts index required to convert fastq in counts, and indropCounts function converts reads in UMI counts. 10XGenomics Cellranger is packed in a docker image and the function cellrangerCount converts fastq to UMI matrix using any of the genome indexes with cellrangerIndexing function.

4.4 Counts table exploration and manipulation

rCASC provides various data inspection and preprocessing tools. genesUmi function generates a plot where the number of detected genes are plotted for each cell with respect to the number of UMI. MitoRibo Plot calculates the percentage of mitochondrial/ribosomal genes with respect to the total number of detected genes in each cell and plots percentage of mitochondrial genes with respect to percentage of ribosomal genes. Furthermore, cells are colored on the basis of the number of detected genes (MitoRibo Plot allows to identify cells with low information content, i.e. those cells with a little number of detectable genes e.g. $<$ 100 genes/cell, little ribosomal content and high content of mitochondrial genes, which indicate cell stress. The function scannobyGtf uses ENSEMBL gtf and the R package refGenome to associate gene symbol with the ENSEMBL gene ID. Furthermore, scannobyGtf allows one to remove mitochondrial/ribosomal genes and "stressed" cells detectable with MitoRibo Plot function. The function lorenzFilter embeds the Lorenz statistics developed by Diaz [21], a cell quality statistics correlated with cell live-dead staining. As counts table preprocessing steps, we implemented the functions checkCountDepth/scnorm to detect the presence of sample specific count–depth relationship (i.e. the relationship existing between transcript-specific expression and sequencing depth) and to adjust the counts table for it. Furthermore, we added two other functions recatPrediction and ccRemove, which are based respectively on the paper of Liu [42] and Barron [43]. The function recatPrediction organizes the single cell data to reconstruct cell cycle pseudo time-series and it is used to understand if a cell cycle effect is present. The above function embeds reCAT software [42], which models the reconstruction of time-series as a traveling salesman problem, thus identifying the shortest possible cycle by passing through each cell exactly once and returning to the start. Since the traveling salesman problem is a NP-hard problem, reCAT is based on a heuristic algorithm, which is used to find the solution. ccRemove function is instead based on the work of Barron and Li [43] and embeds their scLVM (single-cell latent variable model) algorithm, which uses a sophisticated Bayesian latent variable model to reconstruct hidden factors in the expression profile of the cell-cycle genes. This algorithm is able to remove cell-cycle effect from real scRNA-Seq datasets. Thus, ccRemove is used to mitigate the cell cycle effect of the inter-samples transcriptome, when it is detected by recatPrediction function.

4.5 Clustering

For the identification of cell subpopulations we implemented 5 approaches: tSne, Seurat [46], SIMLR [19], griph [25], SHARP [31] and scanpy [29] . The function seuratPCAEval has to be run before executing the clustering program to identify the ‘meta features’, i.e. the subset of PCA components describing the relevant source of cells’ heterogeneity, to be used for clustering. seuratBootstrap function implements data reduction and clustering.. Differently SIMLR implements a k-mean clustering, where the number of clusters (i.e. k) is taken as input. SIMLR, requires as input raw counts log_{10} transformed. SIMLR is capable of learning an appropriate cell-to-cell similarity metric from the input single-cell data and to exploit it for the clustering task. The function simlrBootstrap controls the clustering procedure and the function nClusterEvaluationSIMLR, a wrapper for the R package griph (Graph Inference of Population Heterogeneity), is exploited to estimate the (sub)optimal number “k” of clusters. Griph clustering is based on Louvain modularity. Griph algorithm is closer to agglomerative clustering methods, since every node is initially assigned to its own community and communities are subsequently built by iterative merging. Also scanpy uses for clustering a heuristic method based on modularity optimization. SHARP is executed using SHARP Bootstrap function. The metric called CSS (Cell Stability Score), which describes the persistence of a cell in a specific cluster upon Jackknife resampling and therefore offers a peculiar way of describing cluster stability, is embedded in seuratBootstrap, simlrBootstrap, scanpyBootstrap and griphBootstrap, SHARPBootstrap.

4.6 Feature selection

To select the most important features of each cluster we implemented in the anovaLike function the edgeR ANOVA-like method for single cells [33] and in the functions seuratPrior and genesPrioritization/genesSelection respectively the Seurat and SIMLR genes prioritization methods. hfc function allows the visualization of the genes prioritized with the above methods as heatmap and provides plots of prioritized genes in each single cell.

4.7 Scalability

To estimate the scalability of rCASC clustering we used the GSE106264 dataset. We randomly generated from the 10035 cells (27998 ENSEMBL GENE IDs) the following subsets of cells: 400, 600, 800, 1000, 2000, 5000. Moreover for the subsets with more than 600 cells we randomly sampled the genes: 10000, 8000, 6000, 4000, 2000, 1000, 800. We run SIMLR, tSne, griph and Seurat using 160 permutations within SeqBox hardware : Intel i7 3.5GHz (4 cores), 32 GB RAM and 500 GB SSD disk. We extended, for Seurat, griph and scanpy, the scalability analysis to 10K, 33K, 68K and 101K cells, using 10,000/33,000/68,000 cells from PBMC human datasets, available at 10xGenomics repository (www.10xgenomics.com), and 101,000 cells dataset, made assembling the above mentioned 33,000 and 68,000 PBMC datasets. The analysis was executed on a SGI server (10 x CPU E5-4650 2.4GHz (16 cores), 1TB RAM, 30 TB SATA raid disk) allocating 40 threads for each analysis.

4.8 Autoencoder model coding and hyperparameter selection

We implemented the models in python (version 3.7) using TensorFlow package (version 2.0.0), Keras (version 2.3.1), pandas (version 0.25.3), numpy (version 1.17.4), matplotlib (version 3.1.2), sklearn (version 0.22), scipy (version 1.3.3). Optimisation was done using Adam (Adaptive moment estimation) using the following parameters lr=0.01, beta1=0.9, beta2=0.999, epsilon=1e-08, decay=0.0, loss='mean squared error'. RELU (rectified linear unit) was used as activation function for dense layer. The SCA input gene table could be made by raw or log_{10} transformed or normalized counts

4.9 SCA and SSCA latent space definition

SCA latent space is generated using as input a tab delimited text file having as first column the feature id associated with the latent space node and a second column having the input/output gene associated to the latent space node. Third column is compulsory and includes the reference from which the feature/gene relation was taken. Experimentally validated transcription factors' target genes and the associated transcription factor were retrieved from TRRUST v2.0. Experimentally validated miRNA gene targets and their corresponding miRNA were retrieved from miRTarBase v8.0. Kinases target genes were retrieved from RegPhos v2.0 database. Cancer immune-signature was manually curated, retrieving genes ids from PUBMED articles related to the keyword "cancer immune signature" and genes derived from KEGG "Immune system" pathways. Genes associated with KEGG pathways were manually extracted from KEGG pathways public repository. SSCA latent space is made by the union of TF, miRNA, IS and Kinases data.

4.10 QCM and QCC metrics

QCM and QCC are extensions of CSS. QCC describes the cell stability of a reference cluster with respect to a cluster generated using SCA latent space information. Reference clusters are those generated using any of the clustering tools implemented in rCASC. In QCC, reference clusters are compared to multiple runs SCA, where clusters are constructed using latent space information. High coherence between a reference cluster and a SCA cluster indicates that latent space is able to properly describe reference cluster organization using only the biological knowledge embedded in it. The QCC threshold for an informative latent space cluster is a value grater or equal to 0.5, i.e. in 50% of the SCA runs cells are colocalizing as in corresponding reference cluster. QCM is instead measuring the robustness of the SCA model. Each run of the SCA the latent space starts from a random configuration, which is modelled on the basis of the information provided to the SCA, i.e. gene counts. Thus, if the SCA latent space describes properly some of the reference clusters, then those clusters should remain similar among various runs of SCA. QCM measures the reproducibility of each single cluster over a large set of randomly pairs of SCA runs. The lack of reproducibility between clusters indicates that latent space information is not relevant or not robust enough to support conserved cluster structures. The QCM threshold describing a robust model for a cluster is a value grater or equal to 0.5, i.e. in 50% of random pairs of SCA runs cells are colocalizing in the same cluster.

4.11 COMET analysis

The cluster-specific markers detection was done using COMET [63], which is implemented in rCASC. COMET was set to extract up to 4 features characterizing each cluster. Although COMET analyses all available clusters, marker features are investigated only for those clusters characterized by the mean of both QCM and QCC greater then 0.5.

4.12 SCA handling functions in rCASC

A full description of the SCA handling functions, available in rCASC, is described in SCAtutorial github (https://github.com/kendomaniac/SCAtutorial), which includes a vignette (https://kendomaniac.github.io/SCAtutorial/articles/SCAvignette.html) and the outputs of the exemplary analysis
(https://github.com/kendomaniac/SCAtutorial/tree/master/vignettes/setA).

4.13 Jaccard Value

To verify if a cell is stable the Jaccard coefficient was used.

$$J(A,B) = |\frac{A \cap B}{A \cup B}| \tag{4.1}$$

Calling A and B the two vector that represent the cells belonging to cluster K_i and K_j , be P_0 the cluster generated with all cells and $P_1..P_n$ the clusters generated by bootstrap, J is a percentage value that represents the number of common cells divided for the total cells between P_0 and P_n. Whenever this value is greater than a specific percentage value (80 % was set as default value of threshold) all cells in cluster gain a point. Points for each cells divided by number of permutation define the stability score of each cell. It is very important to highlight that the threshold values have to be at least better than 50 % to be taken in account.

4.13.0.1 CSS Algorithm detailed

Let be C the count matrix with $N \times M$ dimension where N is the gene number and M is cell number.

$$C = \begin{bmatrix} c_{11} & c_{12} & \cdots \\ \vdots & \ddots & \\ c_{N1} & & c_{NM} \end{bmatrix}$$

Then, we define C^p the matrix generated by p^{th} permutation removing q random columns from the original matrix C. Moreover considering the p^{th} permutation we denote L^p the set of all the removed cells in p^{th} permutation with $|L^p| = q$ and $\mathbf{cl}^p$ the vector with length $M - q$ encoding the relation between cells and clusters in p. Hence, $\mathbf{cl}^p_i$ identified the cluster in which the ith cell is inserted in permutation p. Finally, we use the notation $\mathbf{cl}^C$ for indicating the output of the clustering algorithm obtained by all the cells (i.e. matrix C).

The relation symmetric matrix R^p with dimension $M \times M$ is defined as follows:

$$R^p = \begin{bmatrix} r^p_{1,1} & r^p_{1,2} & \dots \\ \vdots & \ddots & \\ r^p_{M,1} & & r^p_{M,M} \end{bmatrix}$$

where $r^p_{i,j}$ is:

$$r^p_{i,j} = \begin{cases} 1 \; if \; \mathbf{cl}^p_i = \mathbf{cl}^p_j \\ 0 \; otherwise \end{cases} \tag{4.2}$$

Similarly we defined R^C the relation symmetric matrix obtained considering all cells and R^{C-L^P} as the relation symmetric matrix R^C in which the columns and the rows associated with cells in L^p are removed. Observe that the sum $R^{C-L^P} + R^p$ will always gives as results a matrix with 3 possible values: 0,1 or 2.

$$R^{C-L^p}[j,i] + R^p[j,i] = \begin{cases} 1 & if \; cell \; i \; and \; cell \; j \; clustered \; together \; in \; R^{C-L^p} \\ & or \; in \; R^P \; only; \\ 2 & if \; cell \; i \; and \; cell \; j \; are \; always \; in \; the \; same \; cluster; \\ 0 & if \; cell \; i \; and \; cell \; j \; are \; never \; in \; the \; same \; cluster. \end{cases}$$

Let $\delta(i,k)$ be the kronecker delta, a function of two variables that return 1 if the variables are equal, and 0 otherwise:

$$\delta(i,k) = \begin{cases} 1 \; if \; i = k \\ 0 \; otherwise \end{cases} \tag{4.3}$$

then we define the function *length* that counts the occurrence of a value k fixing the row j in the matrix $R^{C-L^P} + R^p$.

$$length(j,p,k) = \sum_{i=1}^{M} \delta(R^{C-L^P}[j,i] + R^p[j,i], k) \tag{4.4}$$

We use the function *length* to count the occurrence of 1 or 2 in in the matrix $R^{C-L^P} + R^p$.

Finally, we define the permutation score $pscore_{j,p}$ as:

$$pscore_{j,p} = \frac{length(j,p,2)}{length(j,p,2) + length(j,p,1)} \tag{4.5}$$

where p is a permutation and j a cell. Then, this returns the percentage of cells initially clustered with cell j that remain clustered with cell j in the permutation p.

Then, we define $tscore_{j,s}$ as follow:

$$tscore_{j,s} = \frac{1}{P} \sum_{p \in P} 1_{pscore_{j,p} >= s} \tag{4.6}$$

where P is the total number of permutations and s is user-defined threshold. This metric compute the probability that a cell j is always clustered with the same set of cells given that $pscore_{j,p} >= s$.

4.13.0.2 Example

Be $\mathbf{cl} = \left\{1 \quad 2 \quad 2 \quad 1 \quad 2 \quad 1\right\}$

Be $\mathbf{L} = \left\{6 \quad 2 \quad 2 \quad 4\right\}$

Be $\mathbf{cl}^1 = \left\{1 \quad 2 \quad 1 \quad 1 \quad 2\right\}$

Be $\mathbf{cl}^2 = \left\{1 \quad 2 \quad 1 \quad 2 \quad 2\right\}$

Be $\mathbf{cl}^3 = \left\{1 \quad 2 \quad 1 \quad 1 \quad 2\right\}$

Be $\mathbf{cl}^4 = \left\{1 \quad 2 \quad 2 \quad 1 \quad 2\right\}$

$\forall\ p \in \{1, 2, 3, 4\},\ R_p$ is calculated

for instance hereafter I reported R and R^1

$$R = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}$$

$$R^1 = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

$\forall\ p\ R^{p'} + R^p$ *is calculated*

for instance hereafter I reported $R^{p'} + R^1$

$$R'^1 + R^1 = \begin{bmatrix} 2 & 0 & 1 & 2 & 0 \\ 0 & 2 & 1 & 0 & 2 \\ 1 & 1 & 2 & 1 & 1 \\ 2 & 0 & 1 & 2 & 0 \\ 0 & 2 & 1 & 0 & 2 \end{bmatrix}$$

$\forall\ p,$ *pscore is evaluated with* $S = 0.6$ for instance hereafter I reported $pscore_1$

$$pscore_1 = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

This means that in permutation 1 the third cell is unstable "jumping" from cluster number 1 to cluster number 2 in P. $\mathrm{tscore}_{m,s}$ is evaluated then for each cell $tscore_s =$

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0.25 \\ 0.25 \\ 0 \end{bmatrix}$$

In this Example number of permutation is 4, for statistical relevance, an higher number of permutation is required.

4.14 Reproducibility

Since the end of the 90s omics high-throughput technologies have generated an enormous amount of data, reaching today an exponential growth phase. The analysis of omics big data is a revolutionary means of understanding the molecular basis of disease regulation and susceptibility, and this resource is made accessible to the biological/medical community via bioinformatics frameworks. However, owing to the increasing complexity and the fast evolution of omics methods, the reproducibility crisis [78] demands that we find a way to guarantee robust and reliable results to the research community. Single-cell analysis is instrumental to understanding the functional differences among cells within a tissue. Individual cells of the same phenotype are commonly viewed as identical functional units of a tissue or an organ. However, single-cell sequencing results [44] suggest the presence of a complex organization of heterogeneous cell states that together produce system-level functionalities. A mandatory element of single-cell RNA sequencing (RNA-seq) is the availability of dedicated bioinformatics workflows. rCASC (reproducible classification and analysis for single cells)) is one of the tools developed under the umbrella of the Reproducible Bioinformatics project [34], an open-source community aimed at providing to biologists and medical scientists an easy-to-use and flexible framework, which also guarantees the ability to reproduce results independently by the underlying hardware,using Docker containerization (computational reproducibility). The Reproducible Bioinformatics project was founded and is maintained by the research team of the Elixir node at the University of Turin. An example of stand-alone hardware/software infrastructure for bulk RNA-seq, developed within the Reproducible Bioinformatics project, was described by Beccuti et al. [70]. Indeed, it was developed following the best-practice rules for reproducible computational research, proposed in 2013 by Sandveet al. [79]. It is also listed within the tools developed by the Italian Elixir node.

4.14.1 Docker implementation

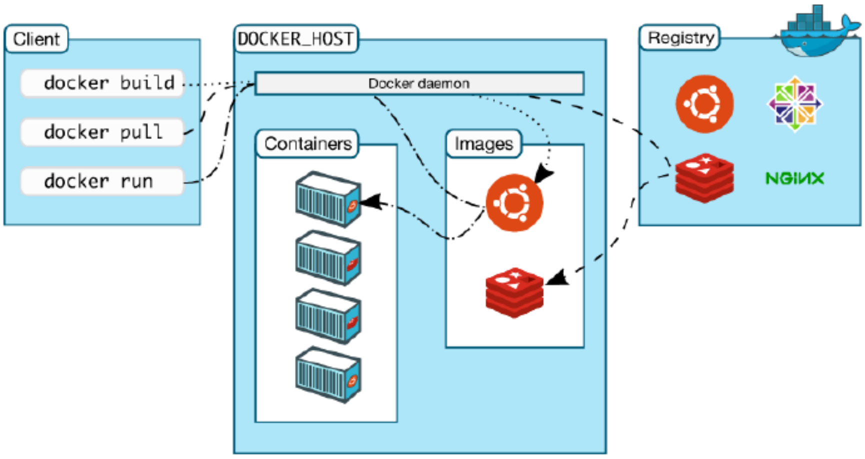


Figure 4.2: Docker's architecture [80]

All the computational tools in rCASC are embedded in Docker images stored in a public repository on the Docker hub. Parameters are delivered to Docker containers via a set of R functions, part of the rCASC R github package [10]. Docker container wraps a piece of software in a complete filesystem that contains everything needed to run: code, runtime, system tools, system libraries – anything that can be installed on a server. This guarantees that the software will always run the same, regardless of its environment (fig. 4.2). Docker Containers and virtual machine are different :

- Containers share the same operating system kernel of the machine in which is running.
- Docker containers are based on open standards, enabling containers to run on all

major Linux distributions and on Microsoft Windows and on top of any infrastructure.

- Container isolates applications from each other and the underlying infrastructure, while providing an added layer of protection for the application.
- Virtual machines include the application, the necessary binaries and libraries, and an entire guest operating system all of which can amount to tens of Gigabyte.
- Container include the application and all of its dependencies but share the kernel with other containers, running as isolated processes in user space on the host operating system. Docker containers are not tied to any specific infrastructure: they run on any computer, on any infrastructure, and in any cloud.

5 Publications

> Wait, aren't you gonna teach us first?
> I believe in learning on the job.
>
> *Snotlout,Gobber- Dragon trainer*

1. Alessandri et al. Differential Expression Analysis in Single-Cell Transcriptomics. Methods Mol Biol. 2019;1979:425-432.

2. Alessandri et al. rCASC: reproducible classification analysis of single-cell sequencing data. Gigascience. 2019 Sep 1;8(9):giz105.

3. Kulkarni et al. Reproducible bioinformatics project: a community for reproducible bioinformatics analysis pipelines. BMC Bioinformatics. 2018 Oct 15;19(Suppl 10):349.

4. Ordoñez-Rueda et al. Apoptotic Cell Exclusion and Bias-Free Single- Cell Selection Are Important Quality Control Requirements for Successful Single- Cell Sequencing Applications. Cytometry A. 2020 Feb;97(2):156-167. doi:

5. Iampietro et al. Molecular and functional characterization of urine-derived podocytes from patients with Alport syndrome. J Pathol. 2020 Jul 11.

6. Alessandri et al. Sparsely-Connected Autoencoder (SCA) for single cell RNAseq data mining-Under revision on npj System Biology and Applications

6 Acknowledgments

> The longer you will wait, the stronger you will like it
>
> *Alessandrì Maurizio*

That was fast! I did not realized how quickly the time passed from the start of my PhD and how many different things I have learned. Cannot say that I arrived where I am now just by myself. There is a lot of people that I wish to thank and I will start from my friend Davide, always available to help me in every kind of weird mathematical algorithm. Also for the mathematical help I would like to thanks Prof. Marco Beccuti, always super happy to help me especially in the formal definition of CSS in materials and methods. Prof. Piero Fariselli was very kind in helping me in understanding the ”neural network world”, kindly answering to all my questions (they were really a lot of questions). I must mention my colleague Maddalena, she helped me not only in overviewing this thesis but she was also my biology Rosetta stone, highlighting tips and trick for optimizing presentations and speech. She is also a very good friend, always ready to help whenever I need. Of course a very special thank to my tutor Prof. Raffaele Calogero. He always believed and supported me. Since I stepped in the lab he was carefully supervising me and I really appreciate that. He created a peaceful and pleasant working environment and I think he has done a great job as tutor and, even more then a tutor. Thanks also to whom is reading my thesis, I hope you enjoyed our work.

Bibliography

[1] F. Sanger, “Nucleotide sequences in bacteriophage ribonucleic acid. the eighth hopkins memorial lecture,” *Biochemical Journal*, vol. 124, pp. 833–843, Oct. 1971.

[2] “Initial sequencing and analysis of the human genome,” *Nature*, vol. 409, pp. 860–921, Feb. 2001.

[3] J. C. Venter, M. D. Adams, E. W. Myers, P. W. Li, R. J. Mural, G. G. Sutton, H. O. Smith, M. Yandell, C. A. Evans, R. A. Holt, J. D. Gocayne, P. Amanatides, R. M. Ballew, D. H. Huson, J. R. Wortman, Q. Zhang, C. D. Kodira, X. H. Zheng, L. Chen, M. Skupski, G. Subramanian, P. D. Thomas, J. Zhang, G. L. G. Miklos, C. Nelson, S. Broder, A. G. Clark, J. Nadeau, V. A. McKusick, N. Zinder, A. J. Levine, R. J. Roberts, M. Simon, C. Slayman, M. Hunkapiller, R. Bolanos, A. Delcher, I. Dew, D. Fasulo, M. Flanigan, L. Florea, A. Halpern, S. Hannenhalli, S. Kravitz, S. Levy, C. Mobarry, K. Reinert, K. Remington, J. Abu-Threideh, E. Beasley, K. Biddick, V. Bonazzi, R. Brandon, M. Cargill, I. Chandramouliswaran, R. Charlab, K. Chaturvedi, Z. Deng, V. D. Francesco, P. Dunn, K. Eilbeck, C. Evangelista, A. E. Gabrielian, W. Gan, W. Ge, F. Gong, Z. Gu, P. Guan, T. J. Heiman, M. E. Higgins, R.-R. Ji, Z. Ke, K. A. Ketchum, Z. Lai, Y. Lei, Z. Li, J. Li, Y. Liang, X. Lin, F. Lu, G. V. Merkulov, N. Milshina, H. M. Moore, A. K. Naik, V. A. Narayan, B. Neelam, D. Nusskern, D. B. Rusch, S. Salzberg, W. Shao, B. Shue, J. Sun, Z. Y. Wang, A. Wang, X. Wang, J. Wang, M.-H. Wei, R. Wides, C. Xiao, C. Yan, A. Yao, J. Ye, M. Zhan, W. Zhang, H. Zhang, Q. Zhao, L. Zheng, F. Zhong, W. Zhong, S. C. Zhu, S. Zhao, D. Gilbert, S. Baumhueter, G. Spier, C. Carter, A. Cravchik, T. Woodage, F. Ali, H. An, A. Awe, D. Baldwin, H. Baden, M. Barnstead, I. Barrow, K. Beeson,

D. Busam, A. Carver, A. Center, M. L. Cheng, L. Curry, S. Danaher, L. Davenport, R. Desilets, S. Dietz, K. Dodson, L. Doup, S. Ferriera, N. Garg, A. Gluecksmann, B. Hart, J. Haynes, C. Haynes, C. Heiner, S. Hladun, D. Hostin, J. Houck, T. Howland, C. Ibegwam, J. Johnson, F. Kalush, L. Kline, S. Koduru, A. Love, F. Mann, D. May, S. McCawley, T. McIntosh, I. McMullen, M. Moy, L. Moy, B. Murphy, K. Nelson, C. Pfannkoch, E. Pratts, V. Puri, H. Qureshi, M. Reardon, R. Rodriguez, Y.-H. Rogers, D. Romblad, B. Ruhfel, R. Scott, C. Sitter, M. Smallwood, E. Stewart, R. Strong, E. Suh, R. Thomas, N. N. Tint, S. Tse, C. Vech, G. Wang, J. Wetter, S. Williams, M. Williams, S. Windsor, E. Winn-Deen, K. Wolfe, J. Zaveri, K. Zaveri, J. F. Abril, R. Guigó, M. J. Campbell, K. V. Sjolander, B. Karlak, A. Kejariwal, H. Mi, B. Lazareva, T. Hatton, A. Narechania, K. Diemer, A. Muruganujan, N. Guo, S. Sato, V. Bafna, S. Istrail, R. Lippert, R. Schwartz, B. Walenz, S. Yooseph, D. Allen, A. Basu, J. Baxendale, L. Blick, M. Caminha, J. Carnes-Stine, P. Caulk, Y.-H. Chiang, M. Coyne, C. Dahlke, A. D. Mays, M. Dombroski, M. Donnelly, D. Ely, S. Esparham, C. Fosler, H. Gire, S. Glanowski, K. Glasser, A. Glodek, M. Gorokhov, K. Graham, B. Gropman, M. Harris, J. Heil, S. Henderson, J. Hoover, D. Jennings, C. Jordan, J. Jordan, J. Kasha, L. Kagan, C. Kraft, A. Levitsky, M. Lewis, X. Liu, J. Lopez, D. Ma, W. Majoros, J. McDaniel, S. Murphy, M. Newman, T. Nguyen, N. Nguyen, M. Nodell, S. Pan, J. Peck, M. Peterson, W. Rowe, R. Sanders, J. Scott, M. Simpson, T. Smith, A. Sprague, T. Stockwell, R. Turner, E. Venter, M. Wang, M. Wen, D. Wu, M. Wu, A. Xia, A. Zandieh, and X. Zhu, “The sequence of the human genome,” *Science*, vol. 291, pp. 1304–1351, Feb. 2001.

[4] H. Ledford, “The death of microarrays?,” *Nature*, vol. 455, pp. 847–847, Oct. 2008.

[5] J. M. Levsky, “Single-cell gene expression profiling,” *Science*, vol. 297, pp. 836–840, Aug. 2002.

[6] R. Durruthy-Durruthy and M. Ray, “Using fluidigm c1 to generate single-cell full-

length cDNA libraries for mRNA sequencing," in *Methods in Molecular Biology*, pp. 199–221, Springer New York, 2018.

[7] E. Z. Macosko, A. Basu, R. Satija, J. Nemesh, K. Shekhar, M. Goldman, I. Tirosh, A. R. Bialas, N. Kamitaki, E. M. Martersteck, J. J. Trombetta, D. A. Weitz, J. R. Sanes, A. K. Shalek, A. Regev, and S. A. McCarroll, "Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets," *Cell*, vol. 161, pp. 1202–1214, May 2015.

[8] R. Zilionis, J. Nainys, A. Veres, V. Savova, D. Zemmour, A. M. Klein, and L. Mazutis, "Single-cell barcoding and sequencing using droplet microfluidics," *Nature Protocols*, vol. 12, pp. 44–73, Dec. 2016.

[9] A. B. Rosenberg, C. M. Roco, R. A. Muscat, A. Kuchina, P. Sample, Z. Yao, L. T. Graybuck, D. J. Peeler, S. Mukherjee, W. Chen, S. H. Pun, D. L. Sellers, B. Tasic, and G. Seelig, "Single-cell profiling of the developing mouse brain and spinal cord with split-pool barcoding," *Science*, vol. 360, pp. 176–182, Mar. 2018.

[10] F. Salmén, P. L. Ståhl, A. Mollbrink, J. F. Navarro, S. Vickovic, J. Frisén, and J. Lundeberg, "Barcoded solid-phase RNA capture for spatial transcriptomics profiling in mammalian tissue sections," *Nature Protocols*, vol. 13, pp. 2501–2534, Oct. 2018.

[11] S. Vickovic, G. Eraslan, F. Salmén, J. Klughammer, L. Stenbeck, D. Schapiro, T. Äijö, R. Bonneau, L. Bergenstråhle, J. F. Navarro, J. Gould, G. K. Griffin, Å. Borg, M. Ronaghi, J. Frisén, J. Lundeberg, A. Regev, and P. L. Ståhl, "High-definition spatial transcriptomics for in situ tissue profiling," *Nature Methods*, vol. 16, pp. 987–990, Sept. 2019.

[12] A. Sansone, "Spatial transcriptomics levels up," *Nature Methods*, vol. 16, pp. 458–458, May 2019.

[13] S. L. Klemm, Z. Shipony, and W. J. Greenleaf, "Chromatin accessibility and the regulatory epigenome," *Nature Reviews Genetics*, vol. 20, pp. 207–220, Jan. 2019.

[14] L. Alessandrì, M. Arigoni, and R. Calogero, "Differential expression analysis in single-cell transcriptomics," in *Methods in Molecular Biology*, pp. 425–432, Springer New York, 2019.

[15] R. Bacher, L.-F. Chu, N. Leng, A. P. Gasch, J. A. Thomson, R. M. Stewart, M. Newton, and C. Kendziorski, "SCnorm: robust normalization of single-cell RNA-seq data," *Nature Methods*, vol. 14, pp. 584–586, Apr. 2017.

[16] M. B. Cole, D. Risso, A. Wagner, D. DeTomaso, J. Ngai, E. Purdom, S. Dudoit, and N. Yosef, "Performance assessment and selection of normalization procedures for single-cell RNA-seq," *Cell Systems*, vol. 8, pp. 315–328.e8, Apr. 2019.

[17] 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, pp. 565–571, May 2017.

[18] R-project.org, "kmean for r," 2015.

[19] B. Wang, D. Ramazzotti, L. D. Sano, J. Zhu, E. Pierson, and S. Batzoglou, "SIMLR: A tool for large-scale genomic analyses by multi-kernel learning," *PROTEOMICS*, vol. 18, p. 1700232, Jan. 2018.

[20] N. Lytal, D. Ran, and L. An, "Normalization methods on single-cell RNA-seq data: An empirical survey," *Frontiers in Genetics*, vol. 11, Feb. 2020.

[21] A. Diaz, S. J. Liu, C. Sandoval, A. Pollen, T. J. Nowakowski, D. A. Lim, and A. Kriegstein, "SCell: integrated analysis of single-cell RNA-seq data," *Bioinformatics*, vol. 32, pp. 2219–2220, Apr. 2016.

[22] S. Picelli, O. R. Faridani, Å. K. Björklund, G. Winberg, S. Sagasser, and R. Sandberg, "Full-length RNA-seq from single cells using smart-seq2," *Nature Protocols*, vol. 9, pp. 171–181, Jan. 2014.

[23] D. Usoskin, A. Furlan, S. Islam, H. Abdo, P. Lönnerberg, D. Lou, J. Hjerling-Leffler, J. Haeggström, O. Kharchenko, P. V. Kharchenko, S. Linnarsson, and P. Ernfors,

"Unbiased classification of sensory neuron types by large-scale single-cell RNA sequencing," *Nature Neuroscience*, vol. 18, pp. 145–153, Nov. 2014.

[24] A. A. Kolodziejczyk, J. K. Kim, V. Svensson, J. C. Marioni, and S. A. Teichmann, "The technology and biology of single-cell RNA sequencing," *Molecular Cell*, vol. 58, pp. 610–620, May 2015.

[25] A. S. Panagiotis Papasaikas, Michael Stadler, "griph: Graph inference of population heterogeneity," 2017.

[26] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, p. P10008, Oct. 2008.

[27] S. Freytag, L. Tian, I. Lönnstedt, M. Ng, and M. Bahlo, "Comparison of clustering tools in r for medium-sized 10x genomics single-cell RNA-sequencing data," *F1000Research*, vol. 7, p. 1297, Dec. 2018.

[28] A. Duò, M. D. Robinson, and C. Soneson, "A systematic performance evaluation of clustering methods for single-cell RNA-seq data," *F1000Research*, vol. 7, p. 1141, Sept. 2018.

[29] F. A. Wolf, P. Angerer, and F. J. Theis, "SCANPY: large-scale single-cell gene expression data analysis," *Genome Biology*, vol. 19, Feb. 2018.

[30] V. A. Traag, L. Waltman, and N. J. van Eck, "From louvain to leiden: guaranteeing well-connected communities," *Scientific Reports*, vol. 9, Mar. 2019.

[31] S. Wan, J. Kim, and K. J. Won, "SHARP: hyperfast and accurate processing of single-cell RNA-seq data via ensemble random projection," *Genome Research*, vol. 30, pp. 205–213, Jan. 2020.

[32] Eric P. Xing, Andrew Y. Ng, Michael I. Jordan,Stuart Russell, "Distance metric learning, with applicationto clustering with side-information."

[33] M. D. Robinson, D. J. McCarthy, and G. K. Smyth, “edgeR: a bioconductor package for differential expression analysis of digital gene expression data,” *Bioinformatics*, vol. 26, pp. 139–140, Nov. 2009.

[34] N. Kulkarni, L. Alessandrì, R. Panero, M. Arigoni, M. Olivero, G. Ferrero, F. Cordero, M. Beccuti, and R. A. Calogero, “Reproducible bioinformatics project: a community for reproducible bioinformatics analysis pipelines,” *BMC Bioinformatics*, vol. 19, Oct. 2018.

[35] L. Alessandrì, F. Cordero, M. Beccuti, M. Arigoni, M. Olivero, G. Romano, S. Rabellino, N. Licheri, G. D. Libero, L. Pace, and R. A. Calogero, “rCASC: reproducible classification analysis of single-cell sequencing data,” *GigaScience*, vol. 8, Sept. 2019.

[36] M. A. Tangaro, G. Donvito, M. Antonacci, M. Chiara, P. Mandreoli, G. Pesole, and F. Zambelli, “Laniakea: an open solution to provide galaxy “on-demand” instances over heterogeneous cloud infrastructures,” *GigaScience*, vol. 9, Apr. 2020.

[37] D. Ordoñez-Rueda, B. Baying, D. Pavlinic, L. Alessandri, Y. Yeboah, J. J. Landry, R. Calogero, V. Benes, and M. Paulsen, “Apoptotic cell exclusion and bias-free single-cell selection are important quality control requirements for successful single-cell sequencing applications,” *Cytometry Part A*, vol. 97, pp. 156–167, Oct. 2019.

[38] L. Pace, C. Goudot, E. Zueva, P. Gueguen, N. Burgdorf, J. J. Waterfall, J.-P. Quivy, G. Almouzni, and S. Amigorena, “The epigenetic control of stemness in cd8 t cell fate commitment,” *Science*, vol. 359, pp. 177–186, Jan. 2018.

[39] L. Montanaro, D. Treré, and M. Derenzini, “Nucleolus, ribosomes, and cancer,” *The American Journal of Pathology*, vol. 173, pp. 301–310, Aug. 2008.

[40] C. M. Nasrallah and T. L. Horvath, “Mitochondrial dynamics in the central regulation of metabolism,” *Nature Reviews Endocrinology*, vol. 10, pp. 650–658, Sept. 2014.

[41] Yunshun Chen , Aaron Lun, Davis McCarthy , Xiaobei Zhou , Mark Robinson, Gordon Smyth , “edger,” 2017.

[42] Z. Liu, H. Lou, K. Xie, H. Wang, N. Chen, O. M. Aparicio, M. Q. Zhang, R. Jiang, and T. Chen, “Reconstructing cell cycle pseudo time-series via single-cell transcriptome data,” *Nature Communications*, vol. 8, June 2017.

[43] M. Barron and J. Li, “Identifying and removing the cell-cycle effect from single-cell RNA-sequencing data,” *Scientific Reports*, vol. 6, Sept. 2016.

[44] F. Buettner, K. N. Natarajan, F. P. Casale, V. Proserpio, A. Scialdone, F. J. Theis, S. A. Teichmann, J. C. Marioni, and O. Stegle, “Computational analysis of cell-to-cell heterogeneity in single-cell RNA-sequencing data reveals hidden subpopulations of cells,” *Nature Biotechnology*, vol. 33, pp. 155–160, Jan. 2015.

[45] L. van der Maaten and G. Hinton, “Visualizing high-dimensional data using t-sne,” *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.

[46] A. Butler, P. Hoffman, P. Smibert, E. Papalexi, and R. Satija, “Integrating single-cell transcriptomic data across different conditions, technologies, and species,” *Nature Biotechnology*, 2018.

[47] C. Hennig, “Cluster-wise assessment of cluster stability,” *Computational Statistics & Data Analysis*, vol. 52, pp. 258–271, Sept. 2007.

[48] Henning C, “fpc r package,” 2019.

[49] M. I. Love, W. Huber, and S. Anders, “Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2,” *Genome Biology*, vol. 15, Dec. 2014.

[50] R. Lopez, J. Regier, M. B. Cole, M. I. Jordan, and N. Yosef, “Deep generative modeling for single-cell transcriptomics,” *Nature Methods*, vol. 15, pp. 1053–1058, Nov. 2018.

[51] T. A. Geddes, T. Kim, L. Nan, J. G. Burchfield, J. Y. H. Yang, D. Tao, and P. Yang, "Autoencoder-based cluster ensembles for single-cell RNA-seq data analysis," *BMC Bioinformatics*, vol. 20, Dec. 2019.

[52] M. B. Badsha, R. Li, B. Liu, Y. I. Li, M. Xian, N. E. Banovich, and A. Q. Fu, "Imputation of single-cell gene expression with an autoencoder neural network," *Quantitative Biology*, vol. 8, pp. 78–94, Jan. 2020.

[53] J. Wang, D. Agarwal, M. Huang, G. Hu, Z. Zhou, C. Ye, and N. R. Zhang, "Data denoising with transfer learning in single-cell transcriptomics," *Nature Methods*, vol. 16, pp. 875–878, Aug. 2019.

[54] T. Wang, T. S. Johnson, W. Shao, Z. Lu, B. R. Helm, J. Zhang, and K. Huang, "BERMUDA: a novel deep transfer learning method for single-cell RNA sequencing batch correction reveals hidden high-resolution cellular subtypes," *Genome Biology*, vol. 20, Aug. 2019.

[55] A. L. Maxwell P. Gold and E. Fraenkel, "Shallow sparsely-connected autoencoders for gene set projection," *Pac Symp Biocomput*, vol. 24, Mar. 2019.

[56] C. Lin, S. Jain, H. Kim, and Z. Bar-Joseph, "Using neural networks for reducing the dimensions of single-cell RNA-seq data," *Nucleic Acids Research*, vol. 45, pp. e156–e156, July 2017.

[57] A. Subramanian, P. Tamayo, V. K. Mootha, S. Mukherjee, B. L. Ebert, M. A. Gillette, A. Paulovich, S. L. Pomeroy, T. R. Golub, E. S. Lander, and J. P. Mesirov, "Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles," *Proceedings of the National Academy of Sciences*, vol. 102, pp. 15545–15550, Sept. 2005.

[58] H. Han, J.-W. Cho, S. Lee, A. Yun, H. Kim, D. Bae, S. Yang, C. Y. Kim, M. Lee, E. Kim, S. Lee, B. Kang, D. Jeong, Y. Kim, H.-N. Jeon, H. Jung, S. Nam, M. Chung, J.-H. Kim, and I. Lee, "TRRUST v2: an expanded reference database of human

and mouse transcriptional regulatory interactions," *Nucleic Acids Research*, vol. 46, pp. D380–D386, Oct. 2017.

[59] C.-H. Chou, S. Shrestha, C.-D. Yang, N.-W. Chang, Y.-L. Lin, K.-W. Liao, W.-C. Huang, T.-H. Sun, S.-J. Tu, W.-H. Lee, M.-Y. Chiew, C.-S. Tai, T.-Y. Wei, T.-R. Tsai, H.-T. Huang, C.-Y. Wang, H.-Y. Wu, S.-Y. Ho, P.-R. Chen, C.-H. Chuang, P.-J. Hsieh, Y.-S. Wu, W.-L. Chen, M.-J. Li, Y.-C. Wu, X.-Y. Huang, F. L. Ng, W. Buddhakosai, P.-C. Huang, K.-C. Lan, C.-Y. Huang, S.-L. Weng, Y.-N. Cheng, C. Liang, W.-L. Hsu, and H.-D. Huang, "miRTarBase update 2018: a resource for experimentally validated microRNA-target interactions," *Nucleic Acids Research*, vol. 46, pp. D296–D302, Nov. 2017.

[60] K.-Y. Huang, H.-Y. Wu, Y.-J. Chen, C.-T. Lu, M.-G. Su, Y.-C. Hsieh, C.-M. Tsai, K.-I. Lin, H.-D. Huang, T.-Y. Lee, and Y.-J. Chen, "RegPhos 2.0: an updated resource to explore protein kinase–substrate phosphorylation networks in mammals," *Database*, vol. 2014, Jan. 2014.

[61] M. Kanehisa, M. Furumichi, M. Tanabe, Y. Sato, and K. Morishima, "KEGG: new perspectives on genomes, pathways, diseases and drugs," *Nucleic Acids Research*, vol. 45, pp. D353–D361, Nov. 2016.

[62] G. Dennis, B. T. Sherman, D. A. Hosack, J. Yang, W. Gao, H. C. Lane, and R. A. Lempicki, "DAVID: Database for Annotation, Visualization, and Integrated Discovery," *Genome Biol.*, vol. 4, no. 5, p. P3, 2003.

[63] C. Delaney, A. Schnell, L. V. Cammarata, A. Yao-Smith, A. Regev, V. K. Kuchroo, and M. Singer, "Combinatorial prediction of marker panels from single-cell transcriptomic data," *Molecular Systems Biology*, vol. 15, Oct. 2019.

[64] G. X. Y. Zheng, J. M. Terry, P. Belgrader, P. Ryvkin, Z. W. Bent, R. Wilson, S. B. Ziraldo, T. D. Wheeler, G. P. McDermott, J. Zhu, M. T. Gregory, J. Shuga, L. Montesclaros, J. G. Underwood, D. A. Masquelier, S. Y. Nishimura, M. Schnall-Levin, P. W. Wyatt, C. M. Hindson, R. Bharadwaj, A. Wong, K. D. Ness, L. W.

Beppu, H. J. Deeg, C. McFarland, K. R. Loeb, W. J. Valente, N. G. Ericson, E. A. Stevens, J. P. Radich, T. S. Mikkelsen, B. J. Hindson, and J. H. Bielas, “Massively parallel digital transcriptional profiling of single cells,” *Nature Communications*, vol. 8, Jan. 2017.

[65] C. Cobaleda, A. Schebesta, A. Delogu, and M. Busslinger, “Pax5: the guardian of b cell identity and function,” *Nature Immunology*, vol. 8, pp. 463–470, Apr. 2007.

[66] J. Ju, K. Zou, and H. Xie, “Downregulation of NFAT5 by RNA interference reduces monoclonal antibody productivity of hybridoma cells,” *Cell Research*, vol. 17, pp. 264–270, Feb. 2007.

[67] Y. Ueda, K. Yang, S. J. Foster, M. Kondo, and G. Kelsoe, “Inflammation controls b lymphopoiesis by regulating chemokine CXCL12 expression,” *Journal of Experimental Medicine*, vol. 199, pp. 47–58, Jan. 2004.

[68] A. Y. Wen, K. M. Sakamoto, and L. S. Miller, “The role of the transcription factor CREB in immune function,” *The Journal of Immunology*, vol. 185, pp. 6413–6419, Nov. 2010.

[69] S.-P. Liu, R.-H. Fu, H.-H. Yu, K.-W. Li, C.-H. Tsai, W.-C. Shyu, and S.-Z. Lin, “MicroRNAs regulation modulated self-renewal and lineage differentiation of stem cells,” *Cell Transplantation*, vol. 18, pp. 1039–1045, Sept. 2009.

[70] M. Beccuti, F. Cordero, M. Arigoni, R. Panero, E. G. Amparore, S. Donatelli, and R. A. Calogero, “SeqBox: RNAseq/ChIPseq reproducible analysis on a consumer game computer,” *Bioinformatics*, vol. 34, pp. 871–872, Oct. 2017.

[71] C. B. González-Blas, X.-J. Quan, R. Duran-Romaña, I. I. Taskiran, D. Koldere, K. Davie, V. Christiaens, S. Makhzami, G. Hulselmans, M. Waegeneer, D. Mauduit, S. Poovathingal, S. Aibar, and S. Aerts, “Identification of genomic enhancers through spatial integration of single-cell transcriptomics and epigenomics,” *Molecular Systems Biology*, vol. 16, May 2020.

[72] J. Williams, “Reproducibility in computational biology,” *Global engage*, Dec. 2017.

[73] G. Chen, B. Ning, and T. Shi, “Single-cell RNA-seq technologies and related computational data analysis,” *Frontiers in Genetics*, vol. 10, Apr. 2019.

[74] S. C. Hicks, F. W. Townes, M. Teng, and R. A. Irizarry, “Missing data and technical variability in single-cell RNA-sequencing experiments,” *Biostatistics*, vol. 19, pp. 562–578, Nov. 2017.

[75] M. Tare, O. R. Puli, and A. Singh, “Molecular genetic mechanisms of axial patterning: Mechanistic insights into generation of axes in the developing eye,” in *Molecular Genetics of Axial Patterning, Growth and Disease in the Drosophila Eye*, pp. 37–73, Springer New York, 2013.

[76] 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, pp. 1187–1201, May 2015.

[77] B. Tang, Z. Hu, Y. Li, S. Yuan, Z. Wang, S. Yu, and S. He, “[corrigendum] downregulation of opioid receptor by RNA interference enhances the sensitivity of BEL/FU drug-resistant human hepatocellular carcinoma cells to 5-FU,” *Molecular Medicine Reports*, Aug. 2018.

[78] D. B. Allison, R. M. Shiffrin, and V. Stodden, “Reproducibility of research: Issues and proposed remedies,” *Proceedings of the National Academy of Sciences*, vol. 115, pp. 2561–2562, Mar. 2018.

[79] G. K. Sandve, A. Nekrutenko, J. Taylor, and E. Hovig, “Ten simple rules for reproducible computational research,” *PLoS Computational Biology*, vol. 9, p. e1003285, Oct. 2013.

[80] “Docker documentation, https://docs.docker.com/engine/understanding-docker/.”