VU
VRIJE
UNIVERSITEIT
AMSTERDAM

Amsterdam
UMC

Using single cell snapshot data to study the dynamics and reverse engineering of gene expression during B cell differentiation in germinal centers

By Martin Banchero

Minor Internship(XM0072)(30EC)

Supervisors: Dr. Perry Moerland & Dr. Aldo Jongejan
Examiner: Dr. Douwe Molenaar
March 18, 2021

Abstract

During the immune adaptive response, naïve B cells undergo a fast maturation process that leads to antibody-secreting cells, that is plasma cells (PC), or memory B cells (MBC). The maturation process occurs in structures called germinal centers located in peripheral lymphoid organs. The small gene regulatory network (GRN) that largely controls the process of B cell maturation and differentiation into PCs is composed of three genes: BCL6, IRF4, and BLIMP1. The dynamics of this small GRN was modeled using RNA-seq data nearly a decade ago by Martinez et al[1]. Recently, single-cell RNAseq (scRNA-seq) data has enabled measuring gene expression at the level of individual cells. Here, we intend to test whether the small GRN proposed by Martinez et al. explains single-cell gene expression data . To address this goal, we use a modular framework, InferenceSnapshot[2], to infer the GRN using publicly available data of human lymphoid organs[3] more precisely from spleen and tonsils . The results obtained in this project revealed that the InferenceSnapshot framework is not suitable for the biological process under study. However, considering the modularity of InferenceSnapshot it is possible for future studies to incorporate different modules to this framework to estimates the small GRN that controls the B cell differentiation in germinal centers.

1 Introduction

The immune system is the main barrier of defense of an organism. One of these barriers is the protection of the extracellular space. This is very important because many pathogens can spread through it. The extracellular space is protected by the humoral immune response via antibodies produced by mature B cells. Antibodies localize pathogens in extracellular space and can inhibit the infection by tagging the pathogen to be destroyed by cells like macrophages or neutrophils, preventing the binding of an antigen to its target, or through the activation of the complement system. The humoral response requires fast maturation of naive B cells into antibody-secreting cells, or plasma cells (PCs), and memory cells (MCs).

The humoral response is initiated in secondary lymphoid organs including the spleen, gut-associated tonsils, and lymph nodes. It comprises B cell proliferation and high-affinity antibody production. These processes are carried out in specialized compartments called germinal centers (GCs) (Figure 1). The internal structure of a germinal center shows two key compartments, the dark zone (DZ) and the light zone (LZ). In the DZ there is a high rate of proliferation of B cells, called centroblasts at this stage and the B cells go through somatic hypermutation (SHM). During this process the locus of the unique B cell receptor (BCR) experiences an intense rate of somatic mutation[4]. The BCR is a trans-membrane immunoglobulin (Ig), that contains the domain of interaction with the antigen (variable domain) encoded in V(D)J genes that are in the BCR locus. SHM leads to a mutated BCR with high, low or no affinity for the antigen. Following this step, B cells migrate from DZ to the LZ, where these cells are denominated as centrocytes. Centrocytes with high affinity for the antigen in the LZ can capture the antigen presented by follicular dendritic cells (FDC), then process and present the antigen through molecules called major histocompatibility complex (MHC) class II to T follicular helper (Tfh) cells. Tfh cells are able to promote survival and proliferation signals in centrocytes[5]. After this process, the selected centrocytes repeat the cycle by migrating to the DZ where they proliferate, undergo SHM and followed by migration to the LZ for positive selection.

For B cells presenting low affinity for the antigen can differentiate into memory B cells (MBCs). This process of differentiation of B cells depends on the encounter of antigen in the LZ and involved the assistance of T follicular helper (Tfh) cells that protect the B cells from being sent to apoptosis. On the other hand, those B cells with low or high affinity but that avoid contact with antigens experience apoptosis in the LZ. B cells with mutations that lead to damage or depletion of the BCR undergo apoptosis within the DZ. B cells, with self-reactive BCR are depleted from the GC[4].

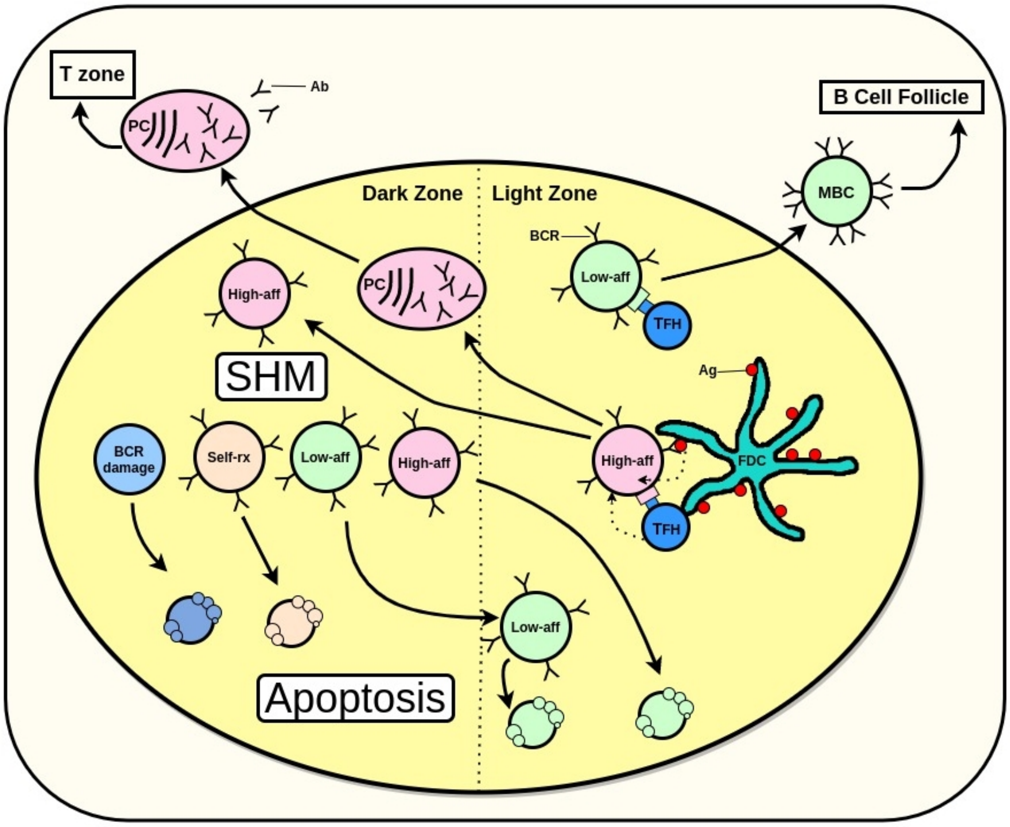


Figure 1: Overview of the process of selection, differentiation, and death of B cells in germinal centers. Adapted from "Selection in the germinal centers" by A.W. Lau and R. Brink, 2020, Current Opinion in Immunology, 63, p.29-34.[4]

The differentiation of B cells into PCs is largely controlled by a small gene regulatory network (GRN) of three transcription factors (Figure 2). B cell lymphoma 6 (BCL6) is a transcriptional repressor highly expressed in GC B cells and is a crucial controller of GC initiation[6]. BCL6 directly represses another important regulator of the differentiation of GC B cells into PCs[7], the B-lymphocyte-induced maturation protein 1 (BLIMP1). The other critical transcription factor is IRF4, a regulator of the PC development[8] , which downregulates BCL6, and transactivates BLIMP1. IRF4 is up regulated via signals generated by the BCR together with the CD40 receptor. These receptors sense co-stimulatory signals in B cells. In addition to this, different cytokines are involved in the activation of IRF4.

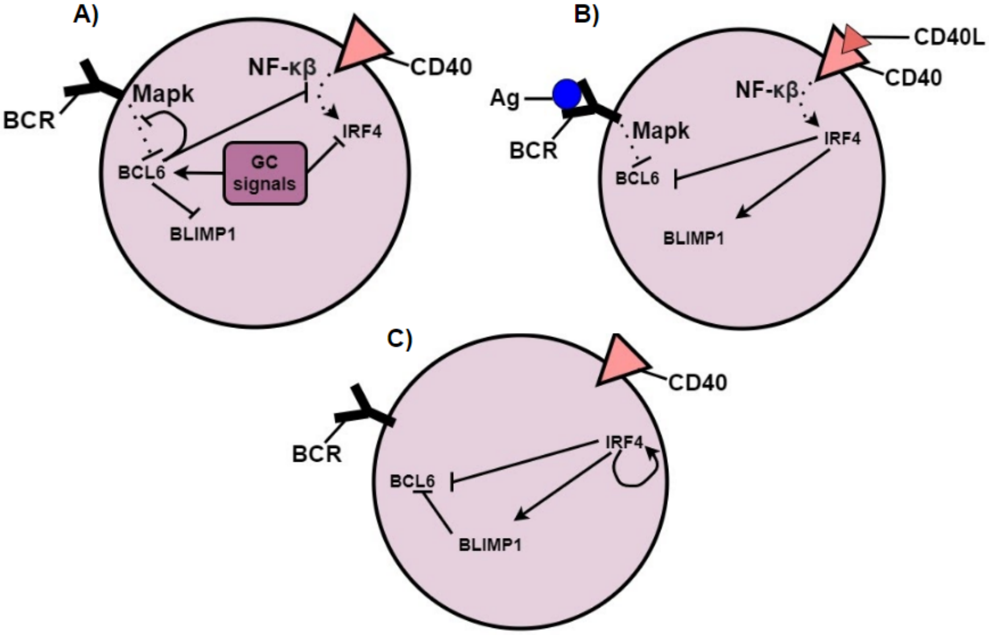


Figure 2: Regulatory network of GC B cells. A) In the first stage, B cells called centroblasts show high expression of BCL6 upregulated by upstream signals. BCL6 is an important repressor that controls the GC regulatory program. **B)** In the next stage, B cells called centrocytes compete for survival signals delivered by BCR and CD40 receptors. Here BCL6 is degraded and IRF4 upregulated. **C)** In this last differentiation stage of plasma cells, IRF4 and BLIMP1 are responsible for transcriptional silencing of BCL6. The cell is locked in this differentiation stage by the action of IRF4 which is upregulated by a self-positive loop. Adapted from "Quantitative modeling of the terminal differentiation of B cells and mechanisms of lymphomagenesis" by M. Rodriguez Martinez, 2012, PNAS, 109, 2672–2677 [1]

This small GRN regulating PC differentiation was previously modelled by Martinez et al. using ordinary differential equations[1] . The parameters of the model were estimated using microarray-based gene expression data measured from mature human B cells associated with the GC reaction. The problem with standard bulk RNA-seq data is that it measures the average gene expression across a population of cells, thereby losing information on gene expression of individual cells. In the last decade, single-cell RNA sequencing data (scRNA-seq)[9] has gained a broad popularity allowing to improve resolution at the cellular level. Recently, single-cell gene expression data in normal human GC B cells isolated from tonsil and spleen tissue from human samples has become publicly available[3,10].

In the last years, several frameworks have been developed to infer GRN dynamics using scRNA-seq data as input, i.e. SCODE[11], GRISLI[12] and InferenceSnapshot proposed by Ocone et al.[2].
In this project we focus on InferenceSnapshot summarized in Figure 3. This is a modular framework that takes as input single-cell snapshot data. This type of data contains gene expression values of a set of genes in a number of cells at a single time-point. The first step in the framework is to apply dimensionality reduction. A low dimensionality representation of high-dimensional snapshot data allows the identification of cell clusters i.e. branches alongside differentiation trajectories, based on their gene expression values. The identified branches can be separated by an ad-hoc clustering algorithm. To obtain GRN dynamics from snapshot data, it is necessary to extract dynamic information from static data. Considering gene expression as a function of time, cells can be ordered along the clustered branches, previously determined in the low-dimensional space, to obtain a pseudo-time series[13]. The latter is used during parameter optimization of the ordinary differential equations (ODE) models and model selection. The transcriptional ODE model space is restricted using a network inference method.

In this project, the main goal is to use the modular framework of InferenceSnapshot to recover temporal behavior from single cell snapshot data and reverse engineer the dynamics of gene expression during B cell differentiation.

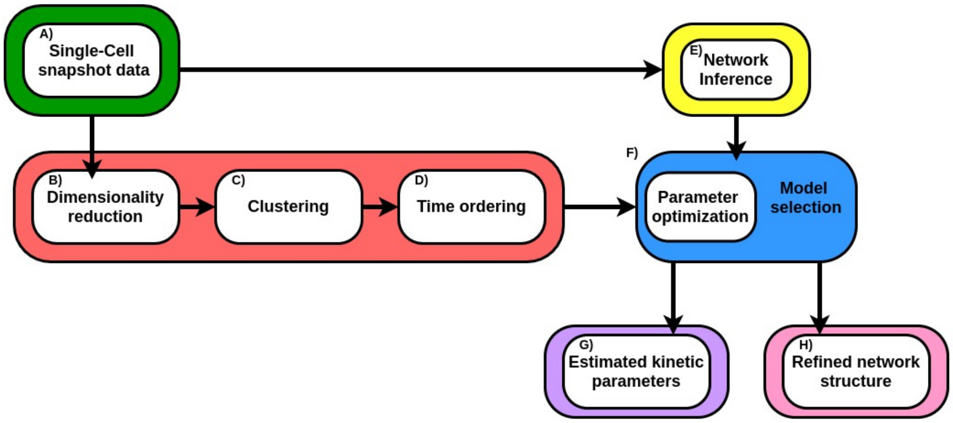


Figure 3: Schematic representation of InferenceSnapshot framework. Single cell snapshot data is used in two paths **(A)**. The first path combined dimensionality reduction **(B)**, clustering **(C)**, and cell time-ordering **(D)**. The second path is a network inference algorithm to infer coarse GRN**(E)**. The result of both paths are combined for model selection and parameter optimization **(F)**. The final output is a refined GRN **(H)** with the estimated kinetic parameters **(G)**. Adapted from "Reconstructing gene regulatory dynamics from high-dimensional single-cell snapshot data", by A. Ocone, 2015, 31, i89–i96.[2]

.

2 Data

2.1 Human lymphoid organs data

The publicly available data used for the analysis was taken from Milpied *et al.*[3] consisting of B cells from human normal (non-cancerous) lymphoid organs, spleen (SP) or tonsil (TS), and follicular lymphoma (FL) tissue. The FL samples were not relevant for our analysis and were therefore removed. The single-cell gene expression analysis by Milpied *et al.* was performed by applying real-time quantitative PCR (qPCR)[14] over 91 pre-selected genes. qPCR allows a sensitive detection of specific transcript levels in a sample. This led to a better resolution of the gene expression profiles of each cell in comparison with scRNA-seq data containing technical dropouts, where only a small proportion of the transcriptome is captured for each cell. As a result of the sc-qPCR analysis we considered only 767 cells that passed quality control (QC). The number of cells used for the analysis is shown in Table 1.

Table 1: Statistics for sc-qPCR data. The spleen samples, SP1, SP2 and SP3 correspond to female donors of 52, 60 and 63 years old respectively. TS1 indicates tonsil sample.

Sample	# sorted single cells	# Cells passing QC
SP1	212	188
SP2	234	186
SP3	223	201
TS1	234	192

The sc-qPCR gene expression matrix was integrated with surface phenotype (phenotype index) data. The phenotypic index allows the separation of B cells into different subsets (Table 2). In the case of GC B cells, the subsets correspond to those cells located in the dark zone (DZ) or in the light zone (LZ) and includes other cells that are also present in the GC. The plasma cells (PC) are divided into three subsets: early plasmablast (Early PB), late plasmablast (Late PB) and mature plasma cells (Mature PC). The memory B cells (MC) are divided into 4 subsets based on the expression of IgD, IgM and IgG. The statistics for each of these subsets are shown in Table 2.

Table 2: Statistics for the different subsets of sorted cells.

Simple sort phenotype	Indexed phenotype	Number of cells
GC	DZ	256
GC	LZ	230
GC	Other	17
PB/PC	Early PB(CD19$^+$ CD20$^+$)	33
PB/PC	Late PB(CD19$^+$ CD20$^-$)	39
PB/PC	Mature PC(CD19$^-$ CD20$^-$)	23
MC	Mem (IgM$^+$ IgD$^+$)	45
MC	Mem (IgM- IgD$^+$)	42
MC	Mem (IgM$^+$ IgD$^-$)	42
MC	Mem (IgM$^-$ IgD$^-$)	40

2.2 ODEs solution data

A second data set used in this project was generated in house (E. Merino Tejero 2020, personal communication). This data set was obtained from the numerical solution of the ODEs proposed by Martinez *et al.*. It contains 400 time-points with time-steps of 0.5, and the gene expression values for the three genes that form the small regulatory network, BCL6, BLIMP1, and IRF4.

3 Methods

In the following sections the different modules of the InferenceSnapshot framework (Figure 3) are explained in more detail.

3.1 Dimensionality reduction

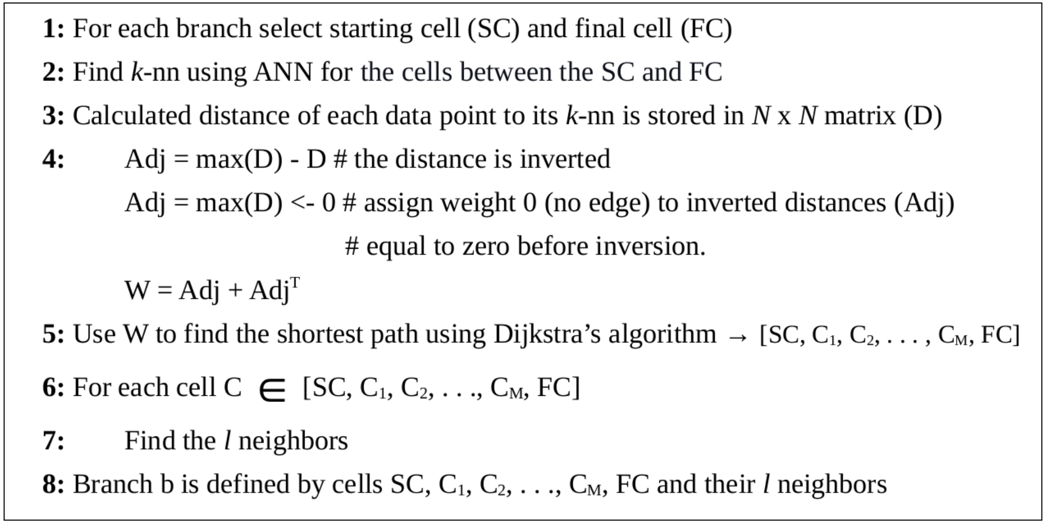
The first module of the InferenceSnapshot framework applies dimensionality reduction to high-dimensional snapshot data. The representation of the data in a low-dimensional space allows the visualization of cells in two or three dimensions. Many dimensionality reduction methods are used to visualize scRNA-seq data, e.g. linear methods like PCA or non-linear methods like UMAP[15]. Despite the capability of these methods to highlight cell subpopulations, these methods are not well suited for visualization of continuous trajectories like those involved in differentiation processes. For this reason, a suitable method to use here is diffusion maps[16]. This non-linear method can conserve the global geometry of data as a continuum and is robust to noise[17]. The cell coordinates are given by eigendecomposition of a sparse transition probability matrix with NxN dimensions, where N denotes the number of cells. The cell to cell probabilities are based on a Gaussian kernel with width σ. The first eigenvectors with decreasing eigenvalues are then considered as diffusion components. Diffusion maps consider similarities between N cells by computing the Euclidean distance in the lower dimensional space determined by the first few eigenvectors. Another parameter, besides the number of eigenvectors to obtain the diffusion map is σ. To obtain the optimal σ I reimplemented the MATLAB code provided in Haghverdi et al.[17] into Python.

The low-dimensional representation obtained with a diffusion map facilitates the identification of continuous trajectories based on their gene expression values. In this context cell clusters can represent different branches present in the differentiation pathways during cell differentiation. This information can be used later in the framework for time ordering and to learn GRNs.

3.2 Branch clustering

To extract relevant information, i.e. cellular differentiation processes from snapshot data, the diffusion map algorithm is applied followed by an ad hoc clustering method. Here, I restructured an in-house Python implementation (A. Jongejan, 2020, personal communication) from the original Matlab implementation provided in Ocone *et al.* This clustering approach separates cells belonging to different processes underlying the snapshot data. The branch clustering strategy starts by choosing a starting cell (SC) and a final cell (FC) for each branch. Next, we search for the k nearest neighbors of the cells between the SC and FC. This step is performed by using a Python implementation of the approximate nearest neighbors (ANN)[18] algorithm named Annoy.
It is important to remark that in our case we are dealing with a data set that contains 767 cells, that is a relatively small number of cells. Because of this, it is possible to apply an exhaustive k-NN method to predict the exact nearest neighbors. However, this method presents a drawback, the computation time increases linearly with the size of the data set, in this case 767 samples (cells) and 91 variables (genes). Taking this into consideration, we decided to follow the same design as in Ocone et al. and continue using ANN, in case of further research that may include larger data sets. The resulting output of Annoy is an adjacency matrix that is used as input for Dijkstra's shortest path algorithm[19]. Here, I used Dijkstra's algorithm to determine the shortest path of a weighted and directed graph. In a weighted graph, the sum of edge weights is minimized while in an unweighted graph the number of nodes is minimized. In a directed graph, the edges connecting the nodes have a direction. In our analysis, the idea is to compute the shortest path using the distances (weights) computed by Annoy and from starting to final cell, a directed graph therefore is the logical option. The final output of the clustering approach is a branch composed of cells belonging to the shortest path between starting and final cell, including their l nearest neighbors. The pseudocode for the branch clustering algorithm (Pc-BCA) is summarized below. This pseudocode was obtained by following the original Matlab implementation of Ocone *et al.*.

Pseudocode for branch clustering algorithm (Pc-BCA)



```
For each branch select starting cell (SC) and final cell (FC)
Find k-nn using ANN for the cells between the SC and FC
Calculated distance of each data point to its k-nn is stored in $N$ x $N$ matrix (D)
    Adj = max(D) - D # the distance is inverted
    Adj = max(D) <- 0 # assign weight 0 (no edge) to inverted distances (Adj)
                          # equal to zero before inversion.
    $W = Adj + Adj^T$
Use W to find the shortest path using Dijkstra’s algorithm → [SC, $C_1$, $C_2$, . . . , $C_M$, FC]
For each cell C $\in$ [SC, $C_1$, $C_2$, . . ., $C_M$, FC]
    Find the $l$ neighbors
Branch b is defined by cells SC, $C_1$, $C_2$, . . ., $C_M$, FC and their $l$ neighbors
```

3.3 Time ordering

The initial input for the framework, as mentioned before, is high-dimensional static snapshot data. However, considering that the final objective of InferenceSnapshot is to obtain the dynamics of GRN, the extraction of dynamic information is needed. Considering gene expression as a function of time, it is possible to recover dynamic information as pseudo-time. This is defined as an ordering of cells based in the trajectories followed by the cells in a differentiation process. The physical time in which a differentiation process occurs is lost during the capture of the process, since cells are destroyed to be sequenced. However, the ordering can be statistically inferred generating pseudo-time.

Cell ordering is carried out by the Wanderlust algorithm[20]. This algorithm can map cells onto a one-dimensional developmental trajectory using high-dimensional data as input. Wanderlust represents cells as nodes in a k-nn graph, these nodes are connected to similar nodes through edges for which weights are computed by a distance metric, e.g. Euclidean or cosine. The trajectory then is defined by computing the shortest path that minimizes the sum of the edge weights in the graph. The starting cell (SC) selected in the branch clustering module is used by Wanderlust to compute the shortest path. Furthermore, the performance of Wanderlust is not considerably affected by a small variation of the SC location[20].

It is important to remark that Wanderlust assumes non-branching trajectories. In the InferenceSnapshot framework, this is overcome by applying the branch clustering strategy described in the previous section. In this module to estimate the time ordering I restructured an in-house Python implementation (A. Jongejan, 2020, personal communication) translated from the original Matlab implementation provided in Ocone et al.

3.4 Network inference

The pseudo-time series obtained in the previous module can be used to infer kinetic parameters of ODE-based transcriptional models. The ODE-based models include knowledge of network structures like different inputs for a target gene, types of regulation referring to inhibitory or activating inputs, and different logical functions (i.e. AND/OR gates) that operate over the target gene. The combination of all these factors leads to a combinatorial explosion of the possible number of ODE models.

In order to reduce the model space, the InferenceSnapshot framework uses the GENIE3 algorithm[21] to predict a coarse GRN. Moreover, the coarse GRN serves as a basis from which to build the ODE models. The main assumption of GENIE3 is that the expression of each (target) gene is a function of other genes (regulators). The main idea behind GENIE3 is to look for regulators that are predictive of the expression of a target gene. This is cast as a regression problem including feature (regulator) selection. The output is an ordered ranking from most to least important features based on a predictive score. The prediction of a regulator of a target gene is significant when the weight assigned to it is larger than a certain threshold. For each set containing a G gene, GENIE3 solves G supervised (non-parametric) regression problems using tree-based ensemble methods. To perform GENIE3 I used a Python package publicly available on GitHub link.

4 Results

4.1 Dimensionality reduction

To begin with our analysis, I performed a 3-dimensional plot using only three genes, BLIMP1, IRF4 and BCL6 that compose the regulatory network described by Martinez *et al.*using the sc-qPCR data of Milpied *et al.* (Figure 4). As mentioned in section 3.1, it is necessary to determine the location of a starting cell (root cell) and final cell to obtain the shortest path in the branch clustering step and later to run the Wanderlust algorithm in the time-ordering module. Clearly, it is not possible to identify the starting cell in this 3-dimensional plot. This means that diffusion maps must be applied to recover an approximate location of the starting cell.

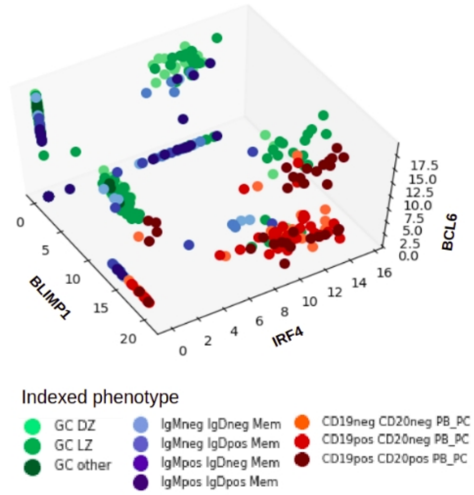


Figure 4: Three-dimensional plot of the three genes that form the small regulatory network. BCL6, BLIMP1 and IRF4. Each point corresponds to a cell and, the different colors indicates each of the different human B cell subsets described in Table 2.

.

The resulting diffusion map is shown in Figure 5. In this plot three branches can be observed, which are associated with different phenotypes. Green indicates cells that belong to the GC and can be subdivided into cells that belong to the DZ, LZ and other cells (i.e. follicular dendritic cells and T follicular helper cells). In blue, memory cells and in red plasma cells, subdivided in early plasma cells (CD19$^+$ CD20$^+$), late plasma cells (CD19$^+$ CD20$^-$) and mature plasma cells (CD19$^-$ CD20$^-$).

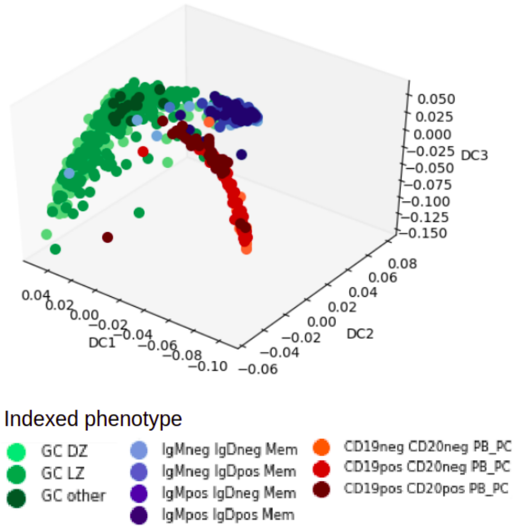


Figure 5: Three-dimensional plot of the first three diffusion components (DCs) obtained with Human lymphoid data.

To obtain the diffusion map shown in Figure 5 only one parameter has to be estimated, the Gaussian kernel width (σ). The plot in Figure 6 represents the dimensionality of the data manifold as a function of log10(σ). For the human lymphoid organs data set, the estimated optimal value of σ is 19.95.

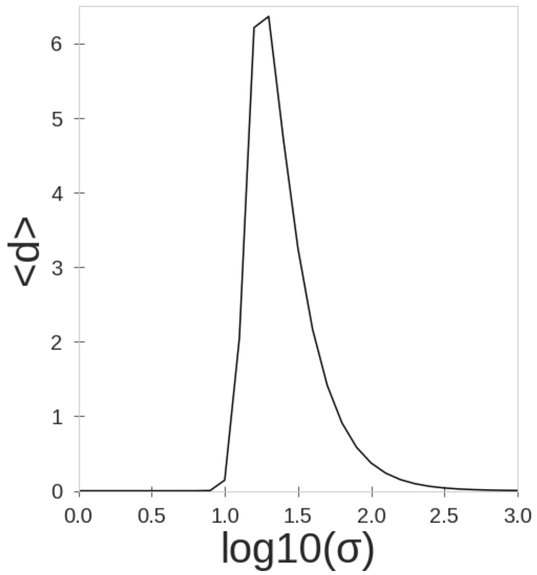


Figure 6: Optimal Gaussian kernel width (σ). The x-axis represents different values of sigma. On the y-axis the average intrinsic dimensionality(<d>) is shown. The optimal value for log10(σ) is 1.29, that is σ 19.95.

In addition to the diffusion map shown before (Figure 5), the same plot was used but, in this case, I colored the cells by gene expression of the three genes that are part of the GRN, (BCL6, BLIMP1, and IRF4; Figure 7). These plots are in agreement with the literature. BCL6 shows high expression within GC cells and low expression for plasma cells and memory cells. This is expected since BCL6 is an important regulator of GC initiation. On the other hand, IRF4 and BLIMP1 show low expression in cells belonging to the GC and high expression levels in plasma cells. This is also expected because BCL6 represses BLIMP1, which is an important regulator of the differentiation of GC B cells into plasma cells. IRF4 is highly expressed in plasma cells, in agreement with its role as a regulator of PC development, where it downregulates BCL6 and transactivates BLIMP1. The up regulation of IRF4 is produced by different signals generated by the BCR together with the CD40 receptor in the LZ.

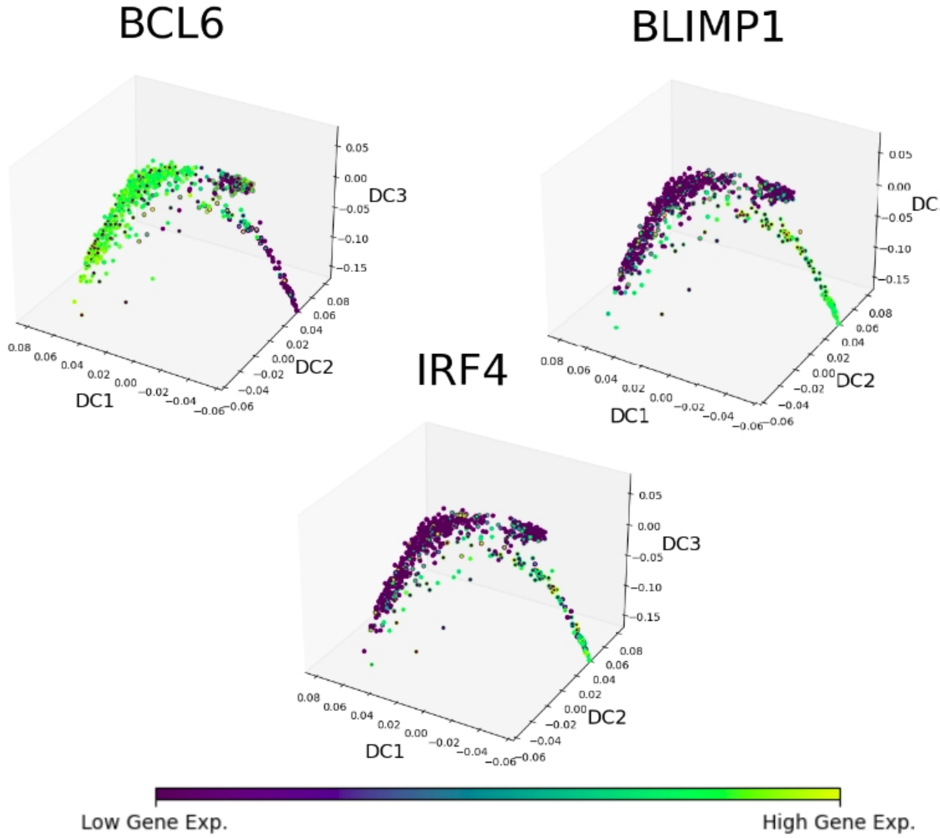


Figure 7: Diffusion map colored by gene expression values for BCL6 (upregulated in GC cells), IRF4 (upregulated in PCs) and BLIMP1 (upregulated in PCs).

4.2 Branch clustering

Continuing with the analysis, the diffusion map showed previously (Figure 5) was visualized in two dimensions and colored by the gene expression values of BCL6 (Figure 8). The decision to start the clustering with cells that belong to GC is arbitrary and based on the idea that the process of B cell differentiation into PCs or MBCs starts in the GC. In two dimensions (Figure 8, left) it is possible to distinguish a potential branch cluster, with high expression values for BCL6 corresponding to positive values of DC1 and negative values of DC2. The plot representing DC1 vs. DC2 shows a clearer visualization of the GC branch compared to the plot representing DC1 vs, DC3 (Figure 8, right). To begin the clustering strategy, we therefore picked a starting cell (SC) and a final cell (FC) using the plot shown in Figure 8, left.

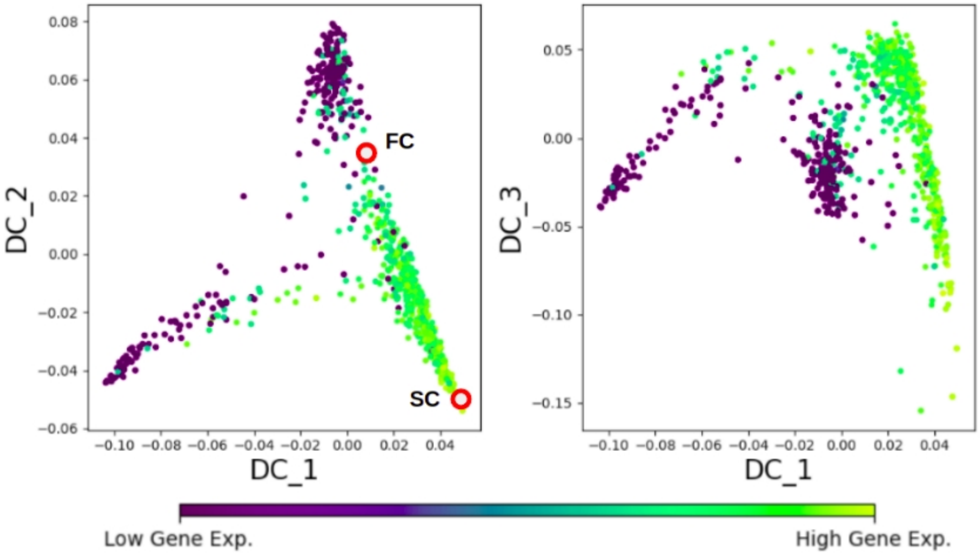


Figure 8: Two-dimensional visualization of the diffusion map obtained for the human lymphoid organs data.The color represents the gene expression values of BCL6. Left: DC1 vs DC2, in red the starting and final cell from which the branch clustering strategy is applied. Right: DC1 vs DC3.

In this step of the InferenceSnapshot framework (Pc-BCA, line 2) to determine the optimal value of k for the approximate nearest neighbors algorithm I tested values of k ranging from 2 to 20. For values of $k < 5$ the nn-graph is disconnected, and results obtained with k between 5 and 20 are not presenting major differences. I therefore continued the analysis using $k = 8$. The ANN algorithm implemented in Annoy determines the k nearest neighbors of all cells using Euclidean distance. Annoy returns an adjacency matrix, which is a matrix representation of a graph. The adjacency matrix is then used as input to determine the shortest path using Dijkstra's algorithm (Figure 9, left). The cells belonging to this path together with their nearest neighbors are considered part of the same branch (Figure 9, right). One can observe that the shortest path (Figure 9, left) presents a deviation from the main group of data points and, clearly, does not represent the expected shortest path.

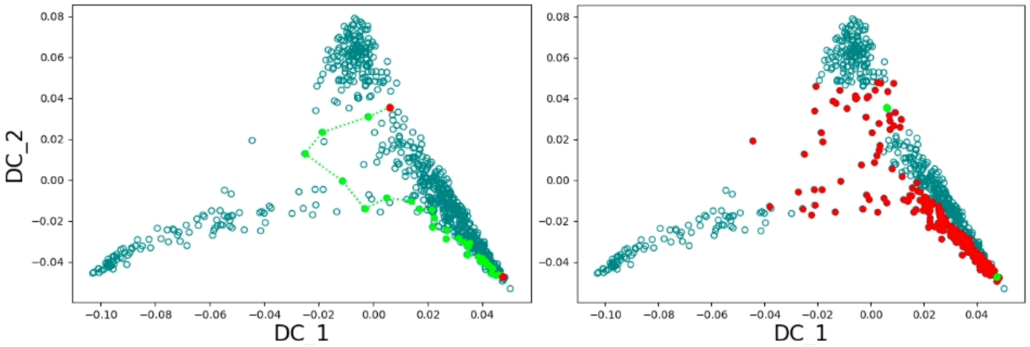


Figure 9: Two-dimensional visualization of diffusion map.Left: In light green, the obtained shortest path from starting to final cell (red points). **Right:** In red, the clustered cells in one branch, starting and final cell are represented in light green.

To get a better insight into why this is the case, I created a toy data set to see whether the ANN or the parameters used to obtain the shortest path were causing this behavior (see details in Supplementary information, section 7.1). The analysis using the toy data set revealed that the InferenceSnapshot implementation applies a distance inversion to the adjacency matrix (see PcBCA, line 4). This step makes that points that originally are close to each other are further away after distance inversion. Based on the analysis using the toy data set I decided to remove the distance inversion step from our implementation. As a result, we obtained a better estimation of the shortest path (Figure 10, left) in comparison with the default method used by InferenceSnapshot. The final clustered branch (Figure 10, right), resulted in a more compact branch in comparison with the results obtained using distance inversion.

The clustered branch shown in Figure 10 (right) can also be visualized in a three-dimensional plot of the diffusion map, see Figure 11(left). The cells in the clustered branch belong to the GC B cells branch (Figure 11, right). The clustering strategy was also applied for the remaining branches, containing PCs and MBCs (Supplementary Figure S5).

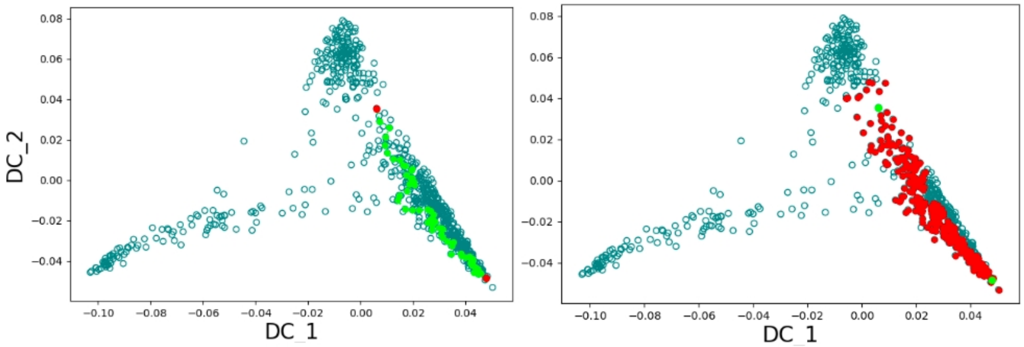


Figure 10: Two-dimensional visualization of diffusion map.Left: In light green the shortest path obtained without distance inversion, the starting and final cells are represented in red. **Right:**In red, the clustered cells in one branch, in light green the starting and final cell.

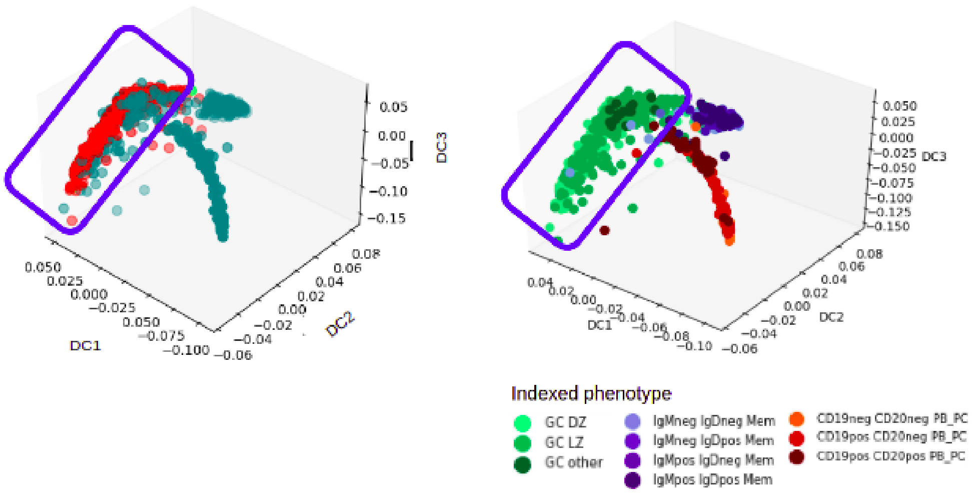


Figure 11: Left:Diffusion map showing in red the clustered branch. Right:Diffusion map colored by different cell phenotypes.

4.3 Time ordering

To obtain the pseudo-time series for BCL6, IRF4 and BLIMP1 the cells must be ordered by time within the clustered branch for GC cells, obtained in the previous step (Figure 11, left). The cells were ordered using Wanderlust algorithm. Before running Wanderlust on the cells in the clustered branch, ten waypoints were chosen (Supplementary Figure S6). These waypoints are used by Wanderlust to build a more reliable trajectory.

The result provided by Wanderlust (Figure 12), is not informative due to the presence of noise. I repeated the same analysis for MBC and PC branches. The results obtained by Wanderlust shown non-informative results for MBC (Supplementary Figure S7 A). However, for PCs (Supplementary Figure S7 B) Wanderlust output shows a pattern of pseudo-time series that agrees what we would expect. This is, PCs showing high expression of BLIMP1 and IRF4 and low expression of BCL6.

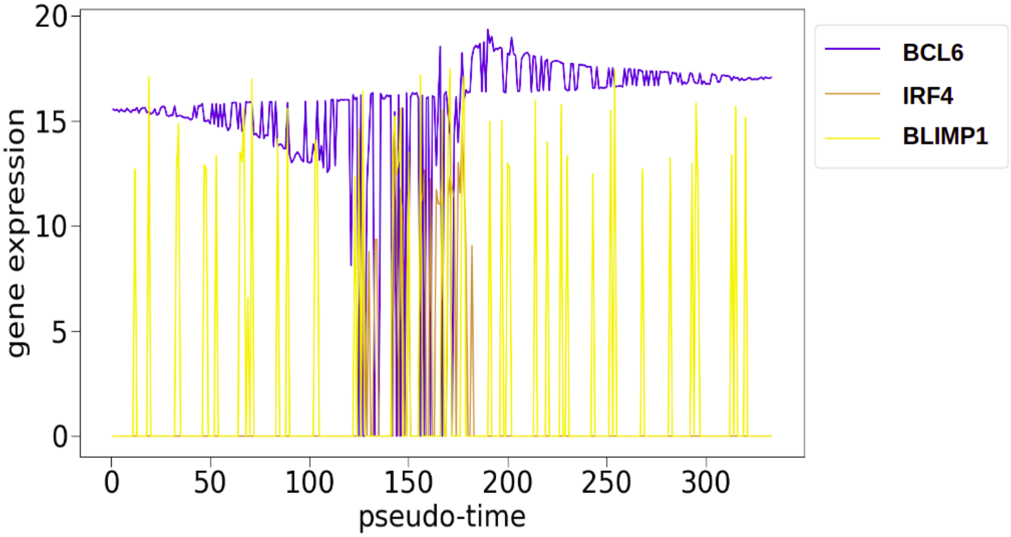


Figure 12: Wanderlust output. The x-axis represents pseudo-time and on the y-axis gene expression values for BCL6, IRF4 and BLIMP1 shown.

Given the non-informative output of Wanderlust, I decided to test the Python implementation to search for possible errors in the implementation.
To do this, I used the in-silico data set described in section 2.2. This data set contained the numerical solution of the ODE models proposed by Martinez *et al.* (Figure 13, left). The results for Wanderlust (Figure 13, right), after applying the InferenceSnapshot framework, return output similar to the expected output given by the ODE numerical solution showing that the implementation is performing correctly on these data (see supplementary section 7.4 for more detail).

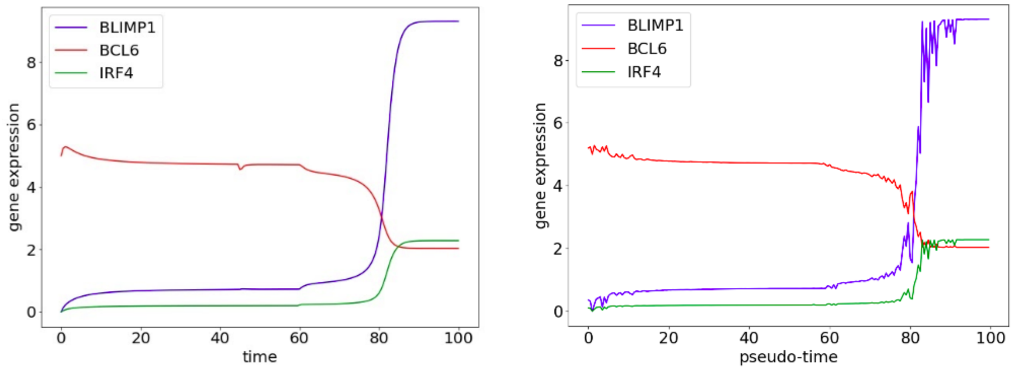


Figure 13: Gene expression profile plot. Left: Plot of ODE solution data. Right: Plot obtained with Wanderlust.

The analysis using the ODE solution data suggests that the non- informative Wanderlust output obtained using the human lymphoid organs data may not be due to errors in our Python implementation but truly represent difficulties in properly identifying the underlying pseudo time.

4.4 Network inference

To explore the network inference module of InferenceSnapshot I applied GENIE3 on the human lymphoid organs data. As a result of GENIE3, a ranking is provided with edge weights. These weights correspond to the importance of a regulator gene to predict the expression of a target gene. The weights are sorted from most to least importance for each of the target genes. In this case, I used as regulators the three genes that form the small regulatory network, BCL6, IRF4, and BLIMP1.

For a regulatory edge to be predicted for a given target gene, the value representing that edge must be above a threshold. However, the selection of a proper threshold remains open as is described by Huynh-Thu *et al.*[21]. In this analysis, I followed the approach used by Ocone *et al.*[2]. Here, the threshold is determined empirically, that is, by looking at all edge weight values and setting the threshold where there is a gap between these values.

The selection of more restrictive thresholds leads to fewer models to be compared during the selection of the ODE models. In our case (Figure 14, left) one can observe two gaps between the edge weight values, one corresponding to a more restrictive threshold at 0.45 and a less restrictive one in the range of 0.30-035. In the latter case I arbitrarily set the threshold at 0.35. Using the 0.35 threshold, we can predict the regulators of BLIMP1, BCL6 and IRF4 and represent the inferred network as a directed graph (Figure 14, right). For IRF4 the predicted regulators are BCL6 and BLIMP1. In the case of BCL6 both IRF4 and BLIMP1 are below the threshold and therefore are not predicted as regulators by GENIE3.

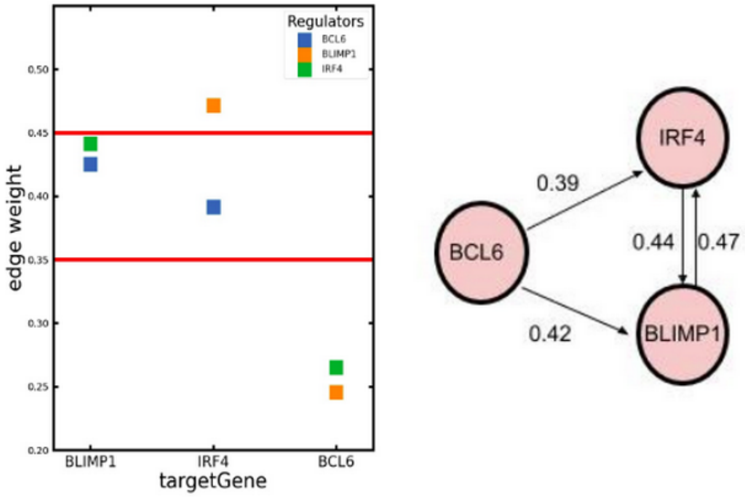


Figure 14: GENIE3 results. Left:Edge weight plot. Right:Inferred small network represented as a directed graph. The numbers represent the edge weights from the left panel.

Once the coarse GRN is inferred, one can apply a correlation analysis on the single-cell snapshot data to further reduce the number of ODE models to compare. The values obtained with Pearson correlation analysis can be associated to regulatory signs of the directed edges (Figure 15) when the correlation coefficient is significant. Here, I considered as significative an absolute correlation value (|r|) above 0.3.
The correlation values of BCL6 agree with the literature, BCL6 downregulates IRF4 and BLIMP1 during B cell differentiation in GC. On the other hand, IRF4 and BCL6 show a correlation value of 0.51. These two genes are both upregulated during B cell differentiation into PCs.

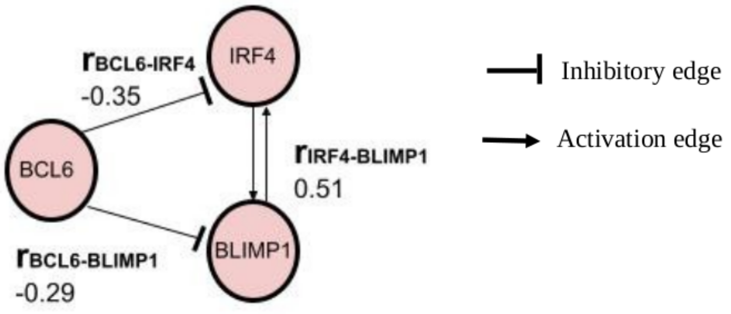


Figure 15: Directed graph including regulatory signs. Correlation values were obtained by using Pearson correlation analysis.

To extend the small GRN shown before, I also considered several other genes that are part of the regulatory network of B cell differentiation within the GC. I added the genes that are described by Nutt *et al.*[22] and are also present in the Milpied data set. These genes are BACH2, IRF8, CIITA, and PAX5. The edge weight plot (Figure 16, left) shows three possible thresholds (red solid lines). I arbitrarily choose the one at 0.2, making model selection less restrictive. The directed graph (Figure 16, right) shows the regulatory edges above the 0.2 threshold. Interestingly, the extended network only keeps the edge between BLIMP1 and IRF4 from those belonging to the small GRN shown in (Figure 15).

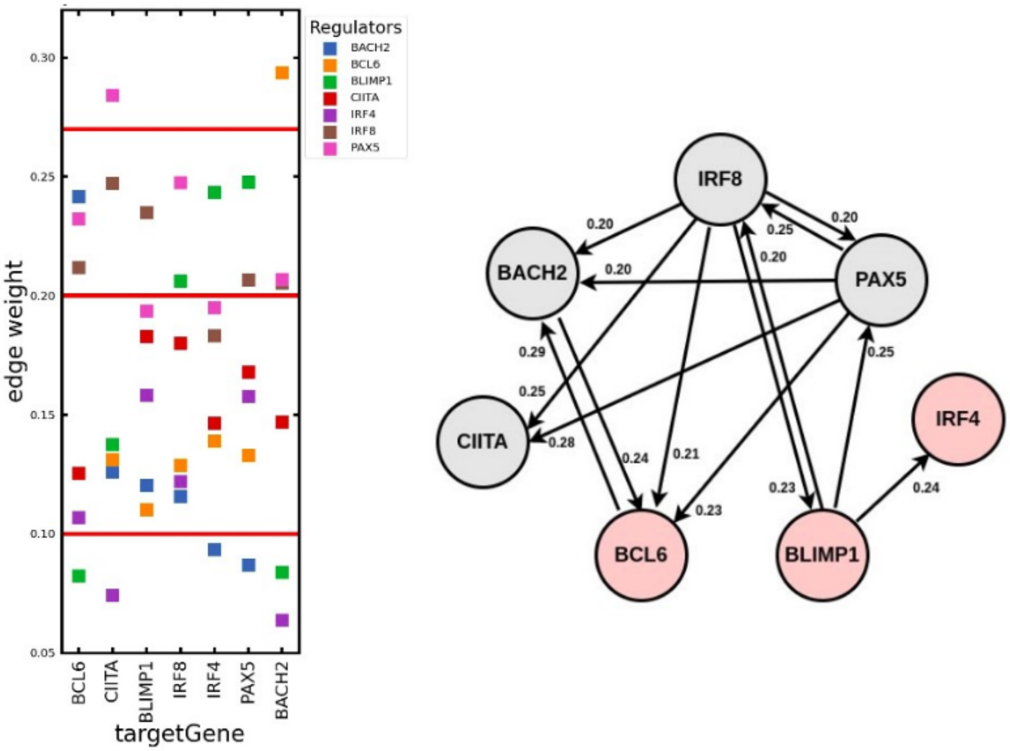


Figure 16: GENIE3 extended network. Left: Edge weight plot. Right: Directed graph. In pink the nodes of the small GRN, in gray the nodes added to extend the small GRN.

The next step consisted in calculating the Pearson correlation to obtain the signs of the edges in the directed graph inferred before. Therefore, I consider as significant correlation values above 0.3.
The results (Figure 17) show a strong positive correlation for BCL6 and BACH2 (r=0.62). According to literature, BCL6 and BACH2 cooperate to suppress genes in GC B cells[23]. Another interesting result is BLIMP1 showing a negative correlation of -0.38 with IRF8, this is in line with IRF8 inducing BCL6 in GC B cells and BCL6 inhibiting BLIMP1[24]. Also, BLIMP1 is negatively correlated to PAX5, an important regulator of the B cell differentiation program. However, how BLIMP1 interacts with PAX5 remains an open question[22] .

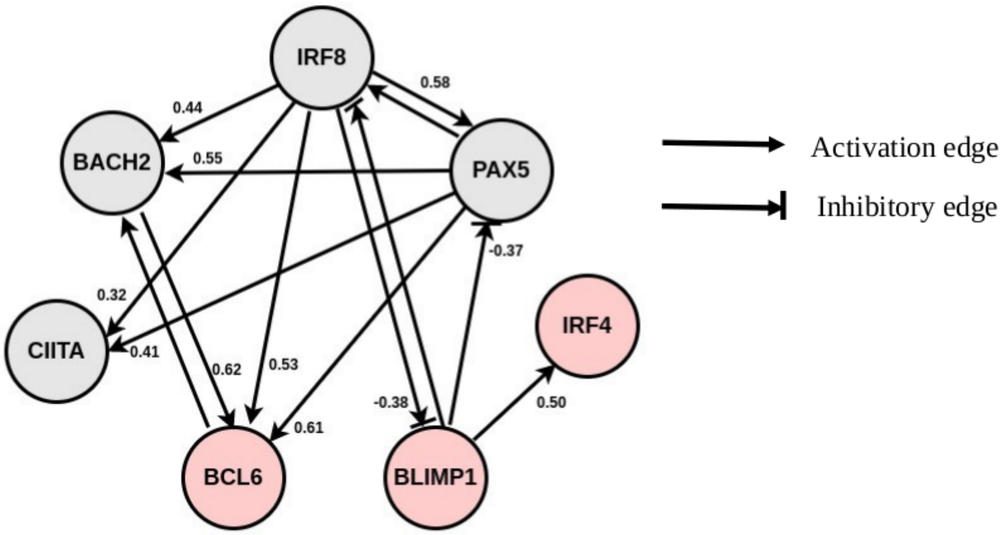


Figure 17: Directed graph including regulatory signs. Correlation values were obtained by Pearson correlation analysis.

5 Discussion and Conclusions

Considering that most B cell lymphomas are derived from GC-experienced B cells[25] the process under study here, differentiation of B cells in GC is of crucial importance to contribute to a better understanding of lymphoma pathogenesis. Here I focused on the quantitative model proposed by Martinez *et al.* to explain the exit of B cells from GC. To test whether this model proposed by Martinez *et al.* was able to explain single-cell expression data of human lymphoid organs, we used the modular framework InferenceSnapshot to reconstruct gene regulatory dynamics.
As a result, we observed that the dimensionality reduction module, using diffusion maps, improved branch clustering, and facilitates determining a starting cell for trajectory inference. The branch clustering approach showed good performance by leaving out from our implementation the distance inversion proposed in the original implementation of InferenceSnapshot.
Results obtained by the time-ordering module resulted to not be suitable for the biological process under study in this project. This may be to the impossibility of Wanderlust to compute accurate pseudo-time series due to the presence of circular trajectories in germinal centers.
Results corresponding to GENIE3 by using three genes (BCL6, IRF4, and BLIMP1) showed a coarse GRN that combined with correlation computed using snapshot data exhibited a correspondence with literature. In addition to this, the extended GRN brings to the spotlight important genes involved in the process of B cell differentiation like BACH2 and IRF8 not considered in the original model proposed by Martinez *et al.*

The analysis in more detail of the different modules leads to different important observations. The diffusion maps obtained during the dimensionality reduction module using human lymphoid data showed three possible branches corresponding to GC B cells, MBCs, and PCs. However, it is important to remark that it is possible to consider these as two partially overlapping branches. This means that the first branch can be defined by cells in the GC to MBCs and the second branch from by cells in the GC to PCs. This last possibility was not considered for the final analysis, but it is important to consider for further studies.
The second module, branch clustering, showed a good performance when not including the distance inversion proposed by Ocone *et al.*. However, it remains an open question why the inverted distances are used in the original

implementation of InferenceSnapshot. Following this strategy, it was possible to obtain three branches, corresponding to GC B cells, MBCs, and PCs respectively.
The third module of InferenceSnapshot, time ordering, showed that the Wanderlust algorithm is not suitable to obtain pseudo-time series from the human organ lymphoid data. One explanation for this is that Wanderlust assumes linear trajectories. Moreover, from the literature, we know that the process of B cell differentiation in GC is circular[3]. An alternative to Wanderlust is to apply more general methods for trajectory inference[26] . A possible suitable method to use in our case is PAGA[27] , a graph-based method that can infer pseudo-time from circular trajectories.
To obtain a coarse GRN using GENIE3, we selected a threshold that leads to a small number of models. The selection of this threshold remains an open problem. The network obtained with GENIE3 was extended by adding regulatory signs based on correlation analysis of the underlying snapshot data. This approach was used for the small gene regulatory network (BCL6, IRF4, and BLIMP1) and an extended network (BCL6, IRF4, BLIMP1, CIITA, BACH2, IRF8, and PAX5). Despite that GENIE3 was originally designed to be used with bulk RNA-seq data, benchmarking analysis of methods to infer GRNs from single-cell gene expression data have shown that GENIE3 remains a top performer in terms of consistency and accuracy[28]. However, GENIE3 presents low performance for noisy single-cell gene expression data[29,30] , low gene expression values (drop-outs).
Recently developed methods to infer GRNs, such as GRISLI[12], can directly infer de novo the GRN from scRNA-seq data. In contrast with GENIE3, GRISLI is dynamic and can also be applied to time-series expression data. However, is not clear whether GRISLI can mitigate drop-outs effects[12]. In addition to this, a recent adaptation of GENIE3 was proposed, called dynGENIE3[31]. This algorithm conserves the scalability of GENIE3, while it can infer GRNs from both static and dynamic data.
The last observation to add is the comparison of GRISLI with InferenceSnapshot. With both methods it is possible to infer GRNs. However, the InferenceSnapshot framework is more informative providing methods for clustering, to recover pseudo-time series and kinetic parameters.

The main goal of this study was focused in testing the quantitative model proposed by Martinez *et al.* that explains the exit of PCs from the GC, by using single-cell expression data. To intend to reach this goal, we used the InferenceSnapshot framework. The results obtained by using the default modules for this framework showed to be not suitable for our research question. However, we were able to develop a better understanding of the biological process and different problems that arose during the application of InferenceSnapshot to human lymphoid organs data. In addition to this I proposed possible approaches to be applied in the future, for example replace the Wanderlust algorithm with another algorithm able to capture circular trajectories and generate a pseudo-time series, i.e PAGA. Furthermore, considering recent developments in GRN inference algorithms it would be interesting to replace the inference network module using GENIE3 with more recently developed methods i.e GRISLI or dynGENIE3. Many tools were developed recently in the field of single cell genomics. This open a wide spectrum of possible methods to be used in the future to answer our question and continue improving our understanding within systems immunology.

6 Acknowledgments

I would like to thank Dr. Perry Moerland and Dr. Aldo Jongejan for giving me the chance to carry out this challenging project under their supervision. To be open and supportive specially during this pandemic times. To all the staff from the Bioinformatics laboratory at the Amsterdam UMC for the support and good predisposition to help when was needed, specially Prof. Antoine Van Kampen and PhD student Elena Merino.

References

[1] María Rodríguez Martínez et al. "Quantitative modeling of the terminal differentiation of B cells and mechanisms of lymphomagenesis". In: *Proceedings of the National Academy of Sciences of the United States of America* 109.7 (Feb. 2012), pp. 2672–2677. ISSN: 00278424. DOI: 10.1073/pnas.1113019109. URL: www.pnas.org/cgi/doi/10.1073/pnas.1113019109.

[2] Andrea Ocone et al. "Reconstructing gene regulatory dynamics from high-dimensional single-cell snapshot data". In: *Bioinformatics* 31.12 (2015), pp. i89–i96. ISSN: 14602059. DOI: 10.1093/bioinformatics/btv257.

[3] Pierre Milpied et al. "Human germinal center transcriptional programs are de-synchronized in B cell lymphoma". In: *Nature Immunology* 19.9 (2018), pp. 1013–1024. ISSN: 15292916. DOI: 10.1038/s41590-018-0181-4. URL: http://dx.doi.org/10.1038/s41590-018-0181-4.

[4] Angelica Wy Lau and Robert Brink. "Selection in the germinal center". In: *Current Opinion in Immunology* 2020 (2019), pp. 29–34. DOI: 10.1016/j.coi.2019.11.001. URL: https://doi.org/10.1016/j.coi.2019.11.001.

[5] Shane Crotty. "T Follicular Helper Cell Biology: A Decade of Discovery and Diseases". In: *Immunity* 50.5 (May 2019), pp. 1132–1148. ISSN: 10974180. DOI: 10.1016/j.immuni.2019.04.011.

[6] Masahiro Kitano et al. "Bcl6 Protein Expression Shapes Pre-Germinal Center B Cell Dynamics and Follicular Helper T Cell Heterogeneity". In: *Immunity* 34.6 (June 2011), pp. 961–972. ISSN: 10747613. DOI: 10.1016/j.immuni.2011.03.025. URL: https://pubmed.ncbi.nlm.nih.gov/21636294/%20https://pubmed.ncbi.nlm.nih.gov/21636294/?otool=inlvulib.

[7] Calame Angelin-Duclos et al. "Plasmacytic Differentiation by Bcl-6 Inhibits prdm1 Direct Repression of". In: *J Immunol References* 173 (2004), pp. 1158–1165. DOI: 10.4049/jimmunol.173.2.1158. URL: http://www.jimmunol.org/content/173/2/1158http://www.jimmunol.org/content/173/2/1158.full%7B%5C#%7Dref-list-1.

[8] Runqing Lu. "Interferon regulatory factor 4 and 8 in B-cell development". In: *Trends in Immunology* 29.10 (Oct. 2008), pp. 487–492. ISSN: 14714906. DOI: 10.1016/j.it.2008.07.006. URL: /pmc/articles/PMC2823592/?report=abstract%20https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2823592/.

[9] Malte D Luecken and Fabian J Theis. "Current best practices in single-cell RNA-seq analysis: a tutorial". In: *Molecular Systems Biology* 15.6 (June 2019). ISSN: 1744-4292. DOI: 10.15252/msb.20188746. URL: https://onlinelibrary.wiley.com/doi/abs/10.15252/msb.20188746.

[10] Antony B. Holmes et al. "Single-cell analysis of germinal-center B cells informs on lymphoma cell of origin and outcome". In: *Journal of Experimental Medicine* 217.10 (2020). ISSN: 15409538. DOI: 10.1084/JEM.20200483. URL: https://pubmed.ncbi.nlm.nih.gov/32603407/.

[11] Hirotaka Matsumoto et al. "SCODE: an efficient regulatory network inference algorithm from single-cell RNA-Seq during differentiation". In: *Bioinformatics* 33.15 (2017), pp. 2314–2321. DOI: 10.1093/bioinformatics/btx194. URL: https://academic.oup.com/bioinformatics/article/33/15/2314/3100331.

[12] Pierre-Cyril Aubin-Frankowski and Jean-Philippe Vert. "Gene regulation inference from single-cell RNA-seq data with linear differential equations and velocity inference". In: *Bioinformatics* 36.18 (Sept. 2020). Ed. by Jan Gorodkin, pp. 4774–4780. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btaa576. URL: https://academic.oup.com/bioinformatics/article/36/18/4774/5858974.

[13] Cole Trapnell et al. "The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells". In: *Nature Biotechnology* 32.4 (Mar. 2014), pp. 381–386. ISSN: 15461696. DOI: 10.1038/nbt.2859. URL: https://www-nature-com.vu-nl.idm.oclc.org/articles/nbt.2859.

[14] Anders Ståhlberg and Mikael Kubista. "Technical aspects and recommendations for single-cell qPCR". In: *Molecular Aspects of Medicine* 59 (Feb. 2018), pp. 28–35. ISSN: 18729452. DOI: 10.1016/j.mam.2017.07.004.

[15] Etienne Becht et al. "Dimensionality reduction for visualizing single-cell data using UMAP". In: *Nature Biotechnology* 37.1 (Jan. 2019), pp. 38–47. ISSN: 15461696. DOI: 10.1038/nbt.4314. URL: https://www-nature-com.vu-nl.idm.oclc.org/articles/nbt.4314.

[16] Ronald R. Coifman and Stéphane Lafon. "Diffusion maps". In: *Applied and Computational Harmonic Analysis* 21.1 (July 2006), pp. 5–30. ISSN: 10635203. DOI: 10.1016/j.acha.2006.04.006.

[17] Laleh Haghverdi, Florian Buettner, and Fabian J. Theis. "Diffusion maps for high-dimensional single-cell analysis of differentiation data". In: *Bioinformatics* 31.18 (July 2015), pp. 2989–2998. ISSN: 14602059. DOI: 10.1093/bioinformatics/btv325. URL: https://pubmed.ncbi.nlm.nih.gov/26002886/%20https://pubmed.ncbi.nlm.nih.gov/26002886/?otool=inlvulib.

[18] Alexandr Andoni, Piotr Indyk, and Ilya Razenshteyn. "Approximate Nearest Neighbor Search in High Dimensions". In: *Proceedings of the International Congress of Mathematicians, ICM 2018* 4 (June 2018), pp. 3305–3336. arXiv: 1806.09823. URL: http://arxiv.org/abs/1806.09823.

[19] E. W. Dijkstra. "A note on two problems in connexion with graphs". In: *Numerische Mathematik* 1.1 (Dec. 1959), pp. 269–271. ISSN: 0029599X. DOI: 10.1007/BF01386390. URL: https://link-springer-com.vu-nl.idm.oclc.org/article/10.1007/BF01386390.

[20] Sean C. Bendall et al. "Single-cell trajectory detection uncovers progression and regulatory coordination in human b cell development". In: *Cell* 157.3 (Apr. 2014), pp. 714–725. ISSN: 10974172. DOI: 10.1016/j.cell.2014.04.005. URL: /pmc/articles/PMC4045247/?report=abstract%20https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4045247/.

[21] V A Huynh-Thu et al. "Inferring Regulatory Networks from Expression Data Using Tree-Based Methods". In: *PLoS ONE* 5.9 (2010), p. 12776. DOI: 10.1371/journal.pone.0012776. URL: www.plosone.org.

[22] Stephen L. Nutt et al. "The generation of antibody-secreting plasma cells". In: *Nature Reviews Immunology* 15.3 (Mar. 2015), pp. 160–171. ISSN: 14741741. DOI: 10.1038/nri3795. URL: www.nature.com/reviews/immunol.

[23] Nilushi S. De Silva and Ulf Klein. “Dynamics of B cells in germinal centres”. In: *Nature Reviews Immunology* 15.3 (Mar. 2015), pp. 137–148. ISSN: 14741741. DOI: 10.1038/nri3804. URL: /pmc/articles/PMC4399774/?report=abstract%20https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4399774/.

[24] Hoon Lee Chang et al. “Regulation of the germinal center gene program by interferon (IFN) regulatory factor 8/IFN consensus sequence-binding protein”. In: *Journal of Experimental Medicine* 203.1 (Jan. 2006), pp. 63–72. ISSN: 00221007. DOI: 10.1084/jem.20051450. URL: /pmc/articles/PMC2118063/?report=abstract%20https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2118063/.

[25] Ralf Küppers. “Mechanisms of B-cell lymphoma pathogenesis”. In: *Nature Reviews Cancer* 5.4 (Apr. 2005), pp. 251–262. ISSN: 1474175X. DOI: 10.1038/nrc1589. URL: https://pubmed.ncbi.nlm.nih.gov/15803153/%20https://pubmed.ncbi.nlm.nih.gov/15803153/?otool=inlvulib.

[26] Wouter Saelens et al. “A comparison of single-cell trajectory inference methods: towards more accurate and robust tools * Equal contribution ^{1} Data mining and Modelling for”. In: (2018). DOI: 10.1101/276907. URL: https://doi.org/10.1101/276907.

[27] F. Alexander Wolf et al. “PAGA: graph abstraction reconciles clustering with trajectory inference through a topology preserving map of single cells”. In: *Genome Biology* 20.1 (Mar. 2019), pp. 1–9. ISSN: 1474760X. DOI: 10.1186/s13059-019-1663-x. URL: https://genomebiology.biomedcentral.com/articles/10.1186/s13059-019-1663-x.

[28] Aditya Pratapa et al. “Benchmarking algorithms for gene regulatory network inference from single-cell transcriptomic data”. In: *Nature Methods* 17.2 (Feb. 2020), pp. 147–154. ISSN: 15487105. DOI: 10.1038/s41592-019-0690-6. URL: /pmc/articles/PMC7098173/?report=abstract%20https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7098173/.

[29] Shuonan Chen and Jessica C. Mar. “Evaluating methods of inferring gene regulatory networks highlights their lack of performance for single cell gene expression data”. In: *BMC Bioinformatics* 19.1 (June 2018), p. 232. ISSN: 14712105. DOI: 10.1186/s12859-018-2217-z. URL:

https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-018-2217-z.

[30] Payam Dibaeinia and Saurabh Sinha. “SERGIO: A Single-Cell Expression Simulator Guided by Gene Regulatory Networks”. In: *Cell Systems* 11 (2020), pp. 252–271. DOI: 10.1016/j.cels.2020.08.003. URL: https://doi.org/10.1016/j.cels.2020.08.003.

[31] Vân Anh Huynh-Thu and Pierre Geurts. “DynGENIE3: Dynamical GENIE3 for the inference of gene networks from time series expression data”. In: *Scientific Reports* 8.1 (2018), p. 3384. ISSN: 20452322. DOI: 10.1038/s41598-018-21715-0. URL: www.nature.com/scientificreports/.

7 Supplementary Information

7.1 Toy data set

The generated toy dataset contains 12 data points divided into 3 clusters (Suppl. Figure S1). For the analysis I used a range of k between 4 and 20 to test the default strategy proposed in InferenceSnapshot. In particular, results obtained by inverting distances when calculating the adjacency matrix (see Pc-BCA, line 4) were compared with results obtained without distance inversion.

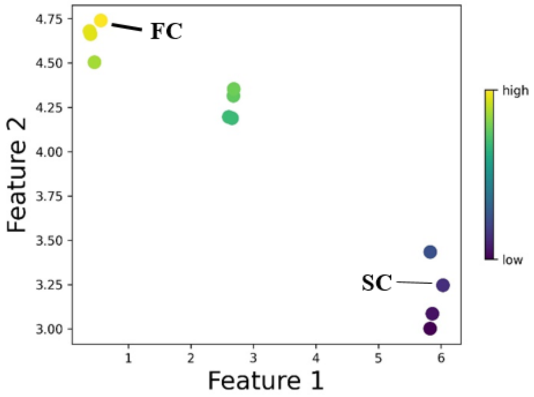


Figure S1: Plot of toy dataset. The points are colored based on feature 2.

The analysis of the toy data set using values of k equals 4 shows the same results for both with distance and without distance inversion (Suppl. Figure S2). In this case the graph used to compute the shortest path is disconnected, since the ANN algorithm considers as nearest neighbors for a data point, the data point itself, leaving i.e. for k=4, 3 nearest neighbors. Therefore, as a result there is no shortest path computed between the SC and FC.

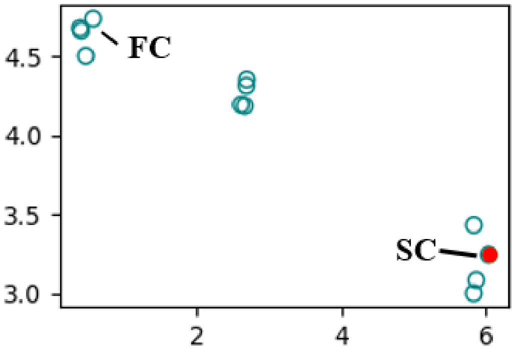


Figure S2: Resulting shortest path using k=4. The same result was obtained with and without distance inversion.

The other case for which the shortest path remains unchanged is for k ≥ 10 using distance inversion (Suppl. Figure S3, left). Here, considering the nearest neighbors of SC, for k=10 the graph is including as nearest neighbors all points in the cluster around value 6, cluster around value 3 and, one point of cluster around value 1, this representing the FC. Therefore, there is a path that goes directly from SC to FC. The final shortest path is minimizing the edge weights (inverted distance) connecting the starting cell with the final cell by one edge. The estimation of the shortest path without distance inversion (Suppl. Figure S3, right) shows that for k ≥ 12 the shortest path remains unchanged. Here, the weights (distances) are smaller in comparison with those distances obtained by using the distance inversion approach. The shortest path minimizes the sum of edge weights by including more nodes in contrast with the shortest path obtained with distance inversion (Suppl. Figure S3, left) in which the minimum sum of edge weights is obtained by considering only one edge.

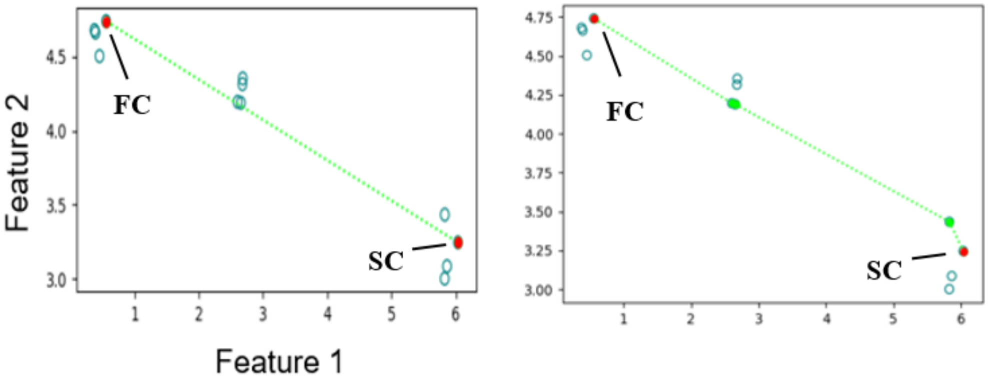


Figure S3: Shortest path plot. Left: With distance inversion using k=10. Right: Without distance inversion for k=12.

Another interesting result was obtained for k = 6, in which the shortest path obtained using distance inversion (Figure S4, left) shows a deviation from what is expected for the shortest path. The distance inversion is leading to points that were close in space before distance inversion, now to be further away from each other. This explains the edge connecting the starting cell with one of the points in the middle cluster instead of connecting it to one of the adjacent points.

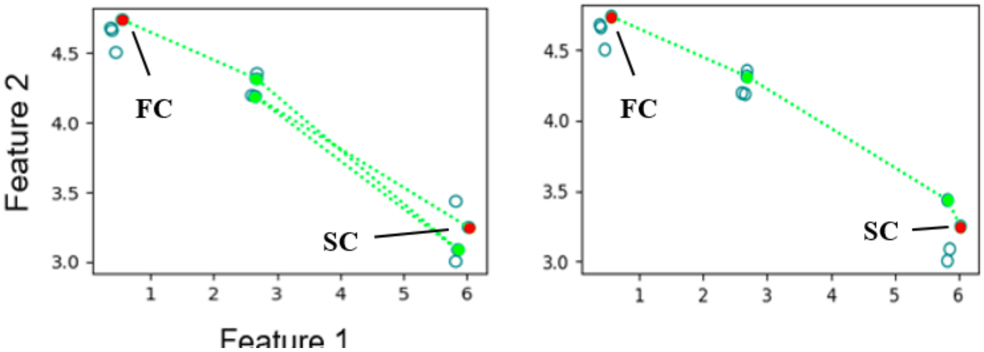


Figure S4: Shortest path plot for k = 6. Left: With distance inversion. Right: Without distance inversion.

7.2 Branches of memory B cells and plasma cells

The result after applied branch clustering to PCs and MBCs (Suppl. Figure S5).

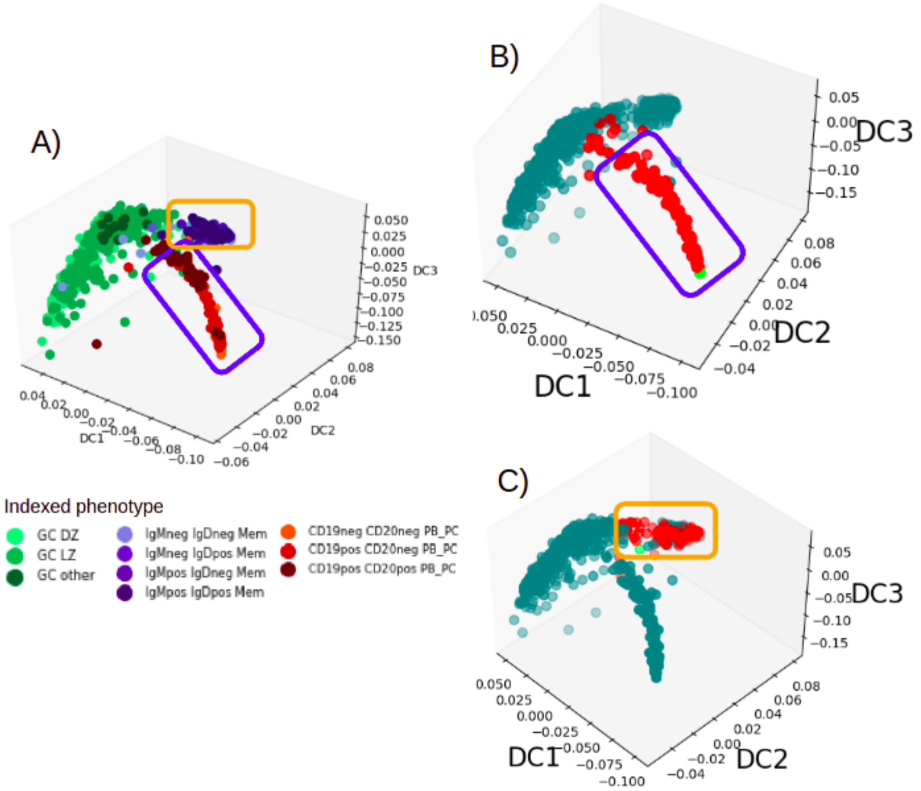


Figure S5: Branch clustering. A) Diffusion map colored by phenotype index. B) Clustered branch corresponding to PCs. C) Clustered branch corresponding to MBCs.

7.3 Time ordering

Once we obtained the clustered branch one step is needed before running Wanderlust to infer the pseudo-time ordering. This step consists of selecting 10 waypoints along the clustered branch (Suppl. Figure S6) and makes the algorithm more robust to noise[20].

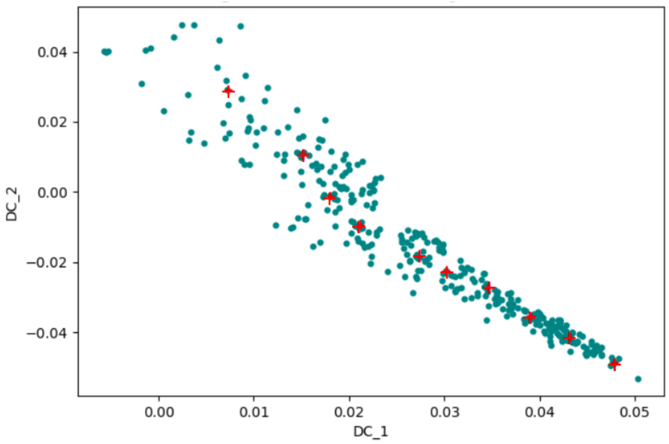


Figure S6: Selection of ten waypoints used by Wanderlust to infer pseudo-time.

I also applied Wanderlust to clustered branches of MBCs and PCs after applied the diffusion maps step followed by branch clustering. The results for MBCs (Suppl. Figure S7A) are non-informative due to the presence of noise across the whole pseudo-time series. The results for PCs (Suppl. S7B) also we obtained noise except for the second half of the pseudo-time series, above pseudo-time value of 60.

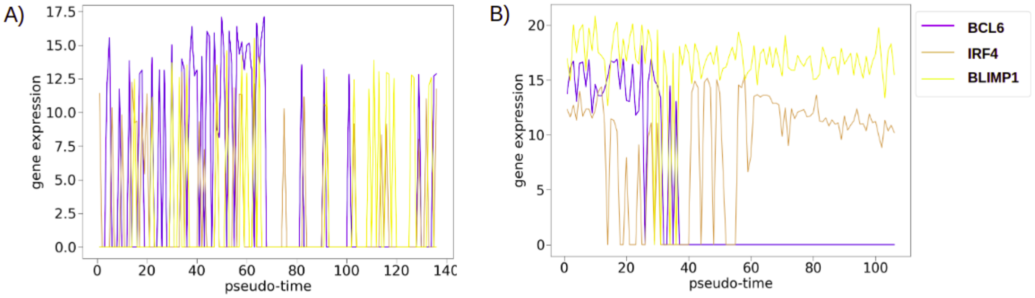


Figure S7: Pseudo-time vs Gene expression plot. A) Wanderlust output using clustered branch corresponding to MBCs. B) Wanderlust output using clustered branch corresponding to PCs.

7.4 Analysis of ODE solution data

In this section I show the steps in InferenceSnapshot as applied to ODE solution data consisting in only one branch. The analysis starts with selection of starting cell (SC) and final cell (FC) over the unique branch (Suppl. Figure S8). In this case dimensionality reduction is not needed due to the presence of only three genes. As a result, we obtained the shortest path (Suppl. Figure S9 A)., the clustered branch represented in 2 dimensions (Suppl. Figure S9 B) and three-dimensional plot of the clustered branch (Suppl. Figure S9 C).

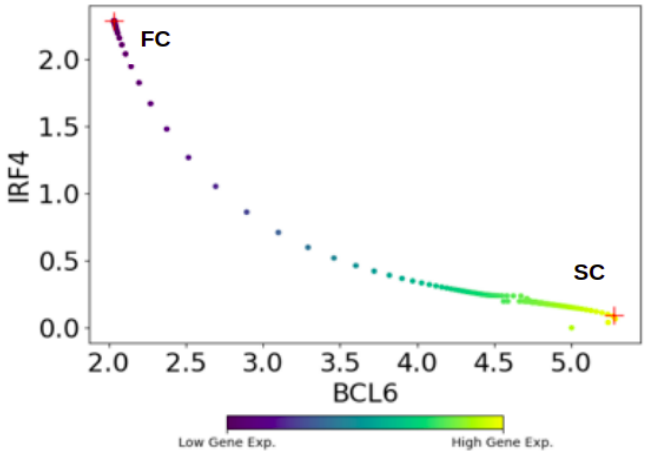


Figure S8: 2D plot of BCL6 vs IRF4 for ODE solution data. In red the starting cell (SC) and final cell (FC).

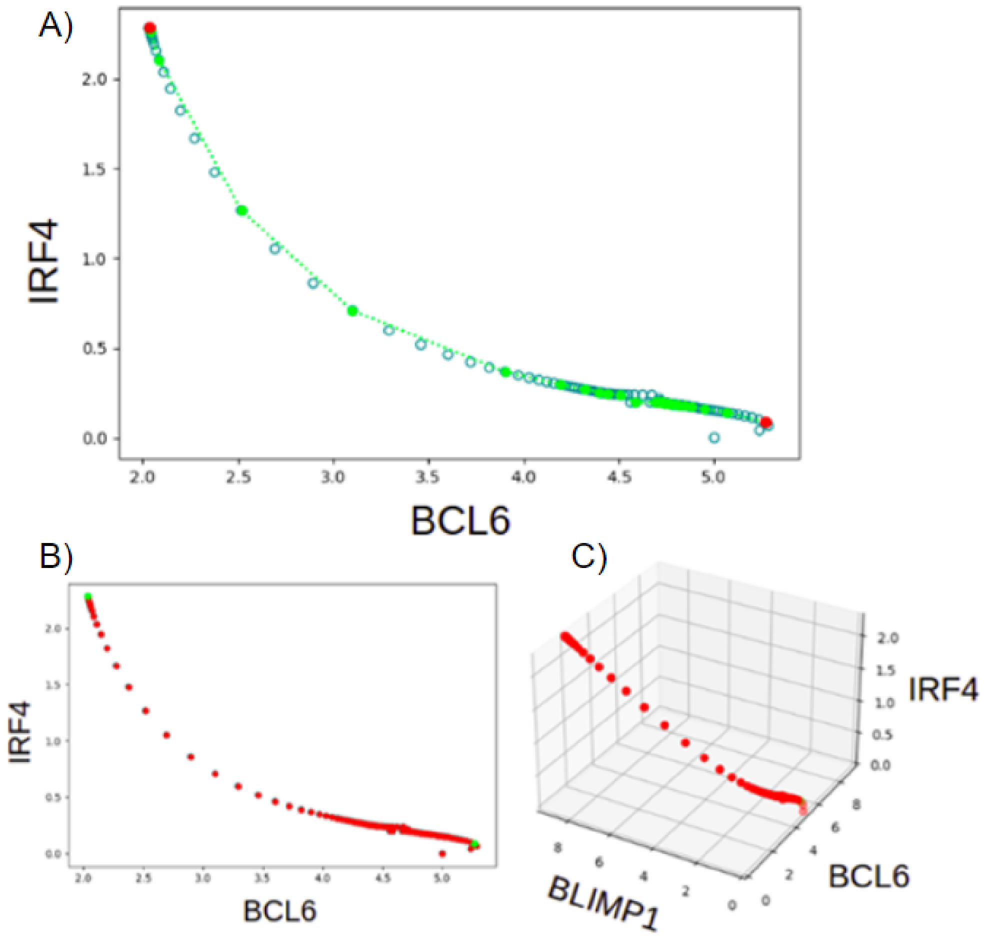


Figure S9: A)Shortest path obtained using ODE solution data. B)2D plot of the clustered Branch. C) 3D plot for the clustered branch.