Report

Hierarchical visualization of single cell RNA-seq data

Bioinformatics Laboratory

Philip van Kuiken (s1585827)
Perry Moerland (supervisor)

Major internship 30 ECTS (X_405027)
13 February 2017 - 31 October 2017

Contents

Abstract

Over the years the number of cells in single cell RNA-seq (scRNA-seq) experiments have increased to several thousands of cells. Exploring and visualizing such an amount of single cells with conventional dimensionality reduction techniques, such as Principal Component Analysis (PCA) and t-Distributed Stochastic Neighbor Embedding (t-SNE) is becoming a challenge. The aim of these algorithms is to project cells to a 2-dimensional visualization space. However, with the significant increase in the number of cells these 2-dimensional representations become dense and packed of cells. As a result, potentially interesting structures might be hidden. We therefore propose to use a hierarchical approach involving multiple 2-dimensional visualization spaces to explore scRNA-seq data, called a hierarchical mixture of probabilistic PCA (hmPPCA). We compared the low-dimensional representation of hmPPCA to low-dimensional representations of PCA and t-SNE by quantifying how well different cell types are separated from each other. HmPPCA proved to be effective in revealing underlying substructures on simulated scRNA-seq datasets. HmPPCA tested on experimental scRNA-seq data achieved similar results compared to t-SNE.

1 Introduction

Multicellular organisms consist of a diversity of cell types with different functions and identities (Jiang et al., 2016). Identification, subtypification and classification of these different cell types is one of the core tasks in molecular cell biology. Investigating these cell types and interpreting differences in the functional elements of the cells is essential in understanding cell development and disease (Jiang et al., 2016). A gold standard does not yet exist for cell type classification, but over the years investigating gene expression profiles has become one of the widely used approaches (Lukk et al., 2010; Su et al., 2002).

Since many years a population of cells can be investigated by profiling cells at the transcriptome level (Wang et al., 2009). The transcriptome is the complete set of transcripts expressed in a cell for a specific developmental stage or physiological function (Wang et al., 2009). In order to express a gene DNA is transcribed to mRNA and translated into a protein. Hence, examining the quantities of mRNA transcripts can be used to profile gene expression levels. With the advances of next generation sequencing technologies (NGS) over the last 10 - 15 years, RNA sequencing (RNA-seq) and more recently single cell RNA sequencing (scRNA-seq) provide expression profiles of the whole transcriptome (Svensson et al., 2018; Wang et al., 2009). While bulk RNA-seq expression experiments have proven to be useful in thousands of studies, the insights found by this technique are only at the cell population level (Stegle et al., 2015). These types of profiles give insight into the average gene expression levels across a population of cells. scRNA expression analysis has the advantage to compare single cells (Stegle et al., 2015). This has led to a wide range of applications where unknown cell types could be recognized in different types of tissues. For instance, Darmanis et al. (2015) sequenced single cells of the adult human brain to interrogate molecular heterogeneity of the human brain. Molecular characterization of the different cell types in the human brain helped in further understanding this complex organ. Another interesting experiment comes from Nestorowa et al. (2016), where they investigated hematopeiotic stem and progenitor cells (HSPCs). By sequencing HSPC transcripts they were able to reconstruct the differentiation trajectories of different HSPC types. This revealed dynamic expression changes associated with early lymphoid, erythroid and granulocyte-macrophage differentiation.

To obtain single cell expression profiles from sample tissue several experimental steps are needed (Figure 1). Over the years different scRNA-seq protocols have been developed to perform these steps (Figure 2). In the earliest scRNA-seq protocols only a few single cells were manually picked under the microscope and analyzed (Tang et al., 2009). Over the years methodologies have been improved, allowing to analyze more cells per experiment. A first step was made with multiplex approaches, where a bar code sequence is added to each cDNA library and libraries are pooled (Figure 2; STRT-seq). Pooling libraries increased the number of cells added to a single tube up to ~100 cells (Islam et al., 2011). This technique was followed by more automated approaches for capturing the cells, such as fluorescence assisted cell sorting (FACS) (Figure 2; Smart-seq, Smart-seq2) and robotic automation (Figure 2; MARS-seq), increasing the number of cells per experiment to several thousands (Jaitin et al., 2014). The first form of a passive capture techniques was called Microfluidics (Figure 2; Fluidigm C1), which was developed by Fluidigm (Brennecke et al., 2013; Svensson et al., 2018). Cells are loaded onto a chip and captured passively in isolated chambers. The most recent studies use droplet based techniques (Figure 2; Drop-seq, inDrop, 10x Genomics), picowell plates (Figure 2; Seq-Well) or in situ bar coding (Figure 2; SPLiT-seq). These techniques increased the number of cells in scRNA-seq studies up to several ten thousands (Klein et al., 2015; Bose et al., 2015; Rosenberg et al., 2018).

Single cell gene expression profiles are high-dimensional due to the high number of expressed genes per cell. Therefore, almost all of the applied visualization approaches in scRNA-seq are dimensionality reduction techniques. These type of methods reduce the number of dimensions, while preserving as much variation and structure of the data as possible. In the end data is visualized in a two-dimensional (2D) or three-dimensional (3D) space that is displayed as a scatter plot.

One of the most popular dimensionality reduction techniques in life sciences is principal component analysis (PCA) (Pierson & Yau, 2015). The aim of this technique is to find directions in the data that explain the largest variance (Bishop, 2006). First eigenvalues and eigenvectors of the covariance matrix are determined. Eigenvalues correspond to the variance of the data in the direction of the corresponding

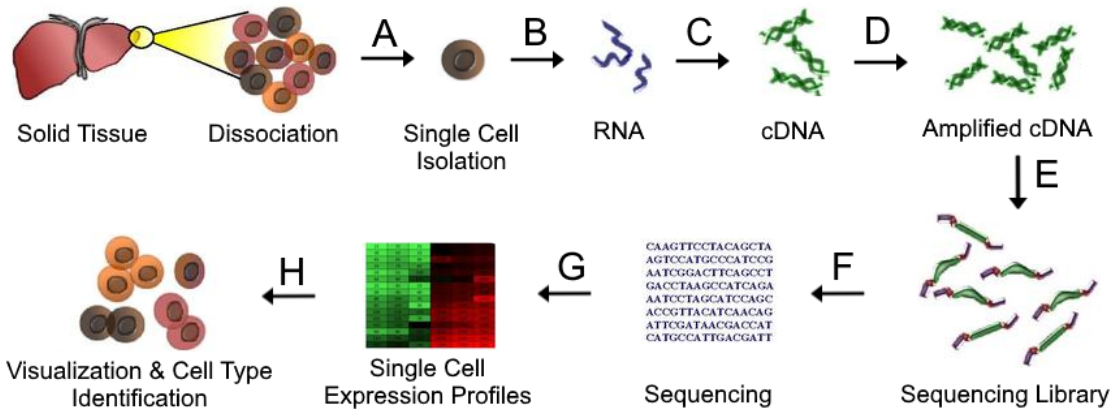

Figure 1: (a) First cells have to be dissociated from the tissue sample and isolated, (b) next the RNA is isolated from the single cells, (c) followed by reverse transcription of RNA to cDNA and cDNA second-strand synthesis, (d) cDNA is amplified, (e) and cDNA is fragmented. (f) Once the cDNA sequences are fragmented they are ready to be sequenced with NGS techniques. (g) With NGS reads are obtained which are mapped back to a reference genome to infer which transcripts are expressed. (h) In the end expression levels are clustered and visualized to investigate cell type heterogeneity (h). Figure has been adapted from Yijyechern (2014).

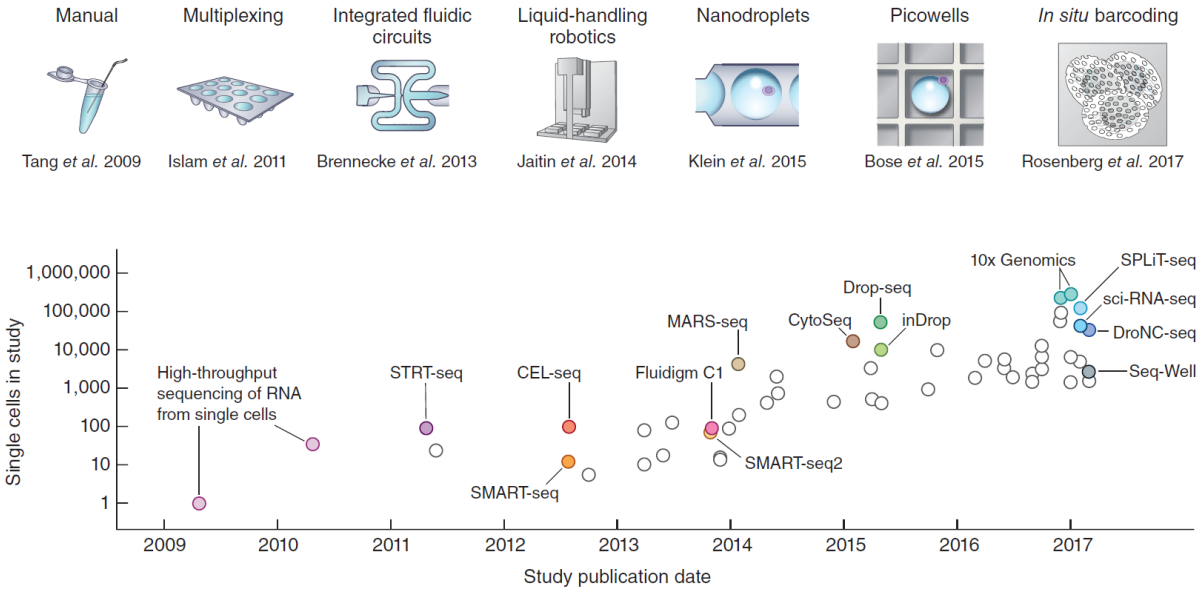

Figure 2: Number of cells handled per technique in scRNA-seq experiments. Over the years the number of cells in scRNA-seq experiments increased from 1 manually picked cell to several hundred thousands captured by picowell technologies and random capture technologies such as nanodroplets. The last increase is made with in situ barcoding techniques reaching almost 1 million cells (e.g. SPLiT-seq). Figure has been adapted from Svensson et al. (2018)

.

eigenvectors. By arranging eigenvectors in decreasing order of the corresponding eigenvalues one finds the eigenvectors explaining most of the variance, from this point on called the principal components. In the end principal components are defined as the linear projections of the data onto a lower dimensional space, making up the principal subspace (Bishop, 2006). The principal subspace can be inspected and visualized in for instance 2D or 3D. In such a case the first two or three principal components explaining the largest variance are selected and visualized.

A more recent dimensionality reduction technique that gained popularity for the visualization of scRNA-seq data is t-distributed stochastic neighbour embedding (t-SNE) (van der Maaten & Hinton, 2008). In contrast to PCA, t-SNE is a nonlinear approach for reducing the number of dimensions (van der Maaten & Hinton, 2008). The idea of t-SNE is to place similar data points nearby each other and dissimilar points more distant from each other in the 2D or 3D visualization space. First the high-dimensional Euclidean distances between data points are converted into conditional probabilities expressing pairwise similarities. These probabilities can be interpreted as the likelihood that a point will have another point as its neighbour (Anchang et al., 2016). The same calculations can be performed in the lower dimensional visualization space resulting in a second matrix of similarities. t-SNE searches for a representation of data points in lower visualization space that minimizes the information loss (Kullback-Leibler divergence) between the two matrices of similarities. (van der Maaten & Hinton, 2008). A drawback of t-SNE is that on high-dimensional datasets the algorithm has more difficulty in finding an interpretable low-dimensional representation of the data. In such a case data is not well clustered together and scattered over the grid of the plot. Therefore, often a PCA preprocessing step is needed to reduce the number of dimensions of the dataset before the t-SNE algorithm is initialized (van der Maaten & Hinton, 2008). As a consequence some of the variance is already removed. In some cases this corresponds to noise, but in other cases low-variance genes are implicitly removed holding important information about the different cell types.

The discussed algorithms visualize the data in 2D and 3D scatterplots. Such an approach is adequate when the number of cells in an experiment is not too large, because differences between clusters of cells, i.e. cell types, can then easily be distinguished. But with scRNA-seq experiments reaching up to thousands of cells, scatter plots become denser and packed of cells. In such a case the overall structure of the data is visualized, but exploring and separating cell types is more difficult. Consequently, potentially underlying internal structures, which could for instance represent an interesting group of rare cell types, cannot be found.

A solution to infer underlying internal structures in scRNA-seq data is by applying hierarchical algorithms. One approach called pcaReduce is based on clustering the scRNA-seq data in a hierarchical manner (Žurauskienė & Yau, 2016). With this approach the authors assumed that broad cellular classes are captured in the first principal components, while refined subclasses are found in lower principal components. The algorithm starts clustering gene expression data on all principal components. The number of clusters is set to a relatively high number to ensure most cell types are captured. This is followed by an iterative loop, where in each iteration clusters with the highest probability to belong to each other are merged. This is followed by removeing of the principal component with the lowest variance. Merging of clusters and removal of the principal component with the lowest variance is repeated until a single cluster remains. In the end this results in a cellular hierarchy of clusters. This method is better able to capture the cellular classes in broad and in detailed gene expression data (Žurauskienė & Yau, 2016). Another technique is called cellTree, which finds hierarchical relationships between cells (duVerle et al., 2016). This method adapted a natural language processing method to compare 'topic histograms' of different cells. The assumption is that each cell consists of a mixture of topics, with gene expression levels as the frequencies in the topic histograms. By fitting a Latent Dirichlet Allocation (LDA) model the topic histograms of the cells can be compared in lower dimensional space. A following step is to infer similarity by calculating a distance matrix in the calculated lower dimensional space. This can be used as input to a hierarchical clustering algorithm to infer hierarchical relationship between cells. More approaches exist which visualize hierarchical relationships in a tree-based manner. One algorithm called SPADE is a computational method for trajectory inference. Trajectory inference algorithms aim to reconstruct the underlying process of cell development over time (Anchang et al., 2016; Cannoodt et al., 2016. For instance, if a set of cells is sampled in different stages of the cell cycle, then SPADE will order cells

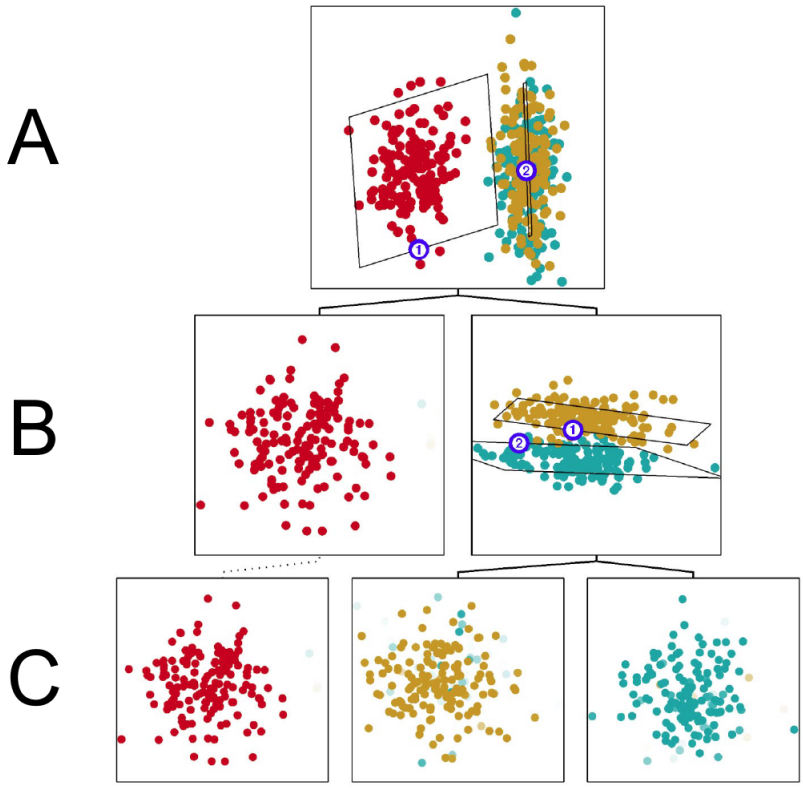

Figure 3: Hierarchical 2D visualization of the toy dataset which consists of 450 data points and 3 dimensions. At the top level (a) the user initializes centers of apparent clusters to extend the hierarchy to the second level (b). The user is then able to investigate other directions of variance in the data, which in this case reveals an internal structure in the cyan/orange cluster. In a next step the user decides to partition the 2 apparent cluster into 2 different components. These are shown at the lowest level (c). Figure has been copied from Bishop & Tipping (1998)

from immature to mature. This allows to investigate the transition stages in cell development, which can reveal novel intermediate states (Cannoodt et al., 2016).

All discussed hierarchical algorithms are able to divide broad cellular classes into smaller subpopulations. We propose to use a hierarchical algorithm, which constructs a hierarchy involving multiple low-dimensional representations of the gene expression data. This hierarchical algorithm preserves the original structure as much as possible, while potentially interesting internal structures can be explored in a hierarchical and interactive manner. For this task we adopted a hierarchical latent variable model from Bishop & Tipping (1998), called PhiVis. To further explain this model we will use an example. In Figure 3 we see a hierarchical tree built on a toy dataset of 450 data points in 3 dimensions. Because the dataset consists of 3 dimensions, dimensionality reduction is needed to create the 2D top level plot (Figure 3A). For this task the model uses a probabilistic PCA (PPCA) algorithm, which is closely related to PCA, and is described in more detail in the Methods section. After dimensionality reduction we observe in the scatter plot a structure of two clusters on top of each other, suggesting there is an internal structure we cannot clearly observe yet (Figure 3A). The strength of PhiVis is that we can dive deeper into the data. This can be achieved by extending the plot to a two-level hierarchy. At the second level new PPCA submodels are initialized based on manually selected centers of apparent clusters in the top level plot. Data points are assigned to the nearest selected center in the original data space. Then a so-called mixture model is fitted at the second level in which points can belong to multiple submodels with a

certain probability. The probability of belonging to a model is visualized ~~in~~ a color gradient per point, where points with a lower probability to belong to a model become more transparent. The advantage of modeling only on a part of the data is that new directions with the largest variance in the data can be found. This is illustrated with the right hand plot in Figure 3B, where the principal components with the largest variance better separate the two clusters. This hierarchical approach of exploring the data also seems suitable for large scRNA-seq datasets, since this might enable to investigate the internal structure of high-dimensional gene expression profiles.

In this report we compare PhiVis with PCA and t-SNE. We investigate if this model performs better in finding internal structures and if cell types can be better distinguished from each other. Bishop & Tipping (1998) tested PhiVis on datasets with just a few dimensions. One of the key aspects of scRNA-seq data is the high number of dimensions. We therefore expect some computational issues which we should recognize and if possible solve. The solutions we have found are discussed in the Methods section. In the Methods section we also explain some important aspects of the hierarchical latent variable model of Bishop & Tipping (1998). We focus on the basics of the PPCA algorithm and the expectation maximization (EM) algorithm used to estimate the parameters of the model. In the Methods section we also propose an extension to the PhiVis algorithm to give the user more guidance in the initial placement of the centers. After that we test PhiVis, PCA and t-SNE on simulated scRNA-seq datasets and scRNA-seq datasets chosen from literature. We evaluate the performance per model by assessing cell type separability per created low-dimensional space. An overview of the results can be found in Section 3. In the last section we discuss the results and possible extensions to the model. In the end this research will answer our research question if a hierarchical visualization of scRNA-seq data outperforms traditional techniques such as PCA and t-SNE in visualizing and investigating the high-dimensional gene expression profiles.

2 Methods

2.1 Overview

For the task of hierarchical visualization of scRNA-seq data we adopted a hierarchical latent mixture model built on probabilistic principal component analysis (PPCA) (Bishop & Tipping, 1998). In this section we explain the key concepts of this model visually and mathematically. In the Introduction we explained the key concepts of PCA, namely that points from high-dimensional space are mapped to a low-dimensional principal subspace. With PPCA similar results are obtained as with PCA, but in a probabilistic way. The advantage of PPCA is that a mixture of PPCA models can be defined, which allows to model more complex data structures. Another advantage is that PPCA can be extended to a hierarchical mixture of PPCA models. The optimal solution of the model is found using an expectation maximization (EM) algorithm to save computational cost.

2.2 Probabilistic principal component analysis

We can denote single cell gene expression levels by d-dimensional vectors $\mathbf{x}_i$ in data space $\mathbf{x} = [\mathbf{x}_1, ..., \mathbf{x}_n]$, where n denotes the number of cells and d the number of genes. PPCA assumes that data is generated from a k-dimensional Gaussian distributed latent space $\mathbf{z} = [\mathbf{z}_1, ..., \mathbf{z}_n]$ with $k \ll d$ (Figure 4A). Cells from this latent space can then be mapped to a higher dimensional data space by a linear transformation (Figure 4B). Because data is not expected to be exactly a flat sheet after transformation to data space, Gaussian distributed noise $\boldsymbol{\epsilon}$ is added. This gives a linear transformation function in the form of:

$$\mathbf{x} = \mathbf{W}\mathbf{z} + \boldsymbol{\mu} + \boldsymbol{\epsilon} \tag{1}$$

where $\mathbf{W}$ denotes a $d \times k$ transformation matrix, $\boldsymbol{\mu}$ is a $d \times 1$ mean vector and $\boldsymbol{\epsilon}$ is a $d \times 1$ noise vector. Because PPCA is limited to Gaussian distributions the noise distribution is modeled as:

$$\boldsymbol{\epsilon} \sim Normal(\mathbf{0}, \sigma^2 \mathbf{I}_d) \tag{2}$$

where $\mathbf{I}$ denotes the $d \times d$ identity matrix. The final density model for the mapping from latent space

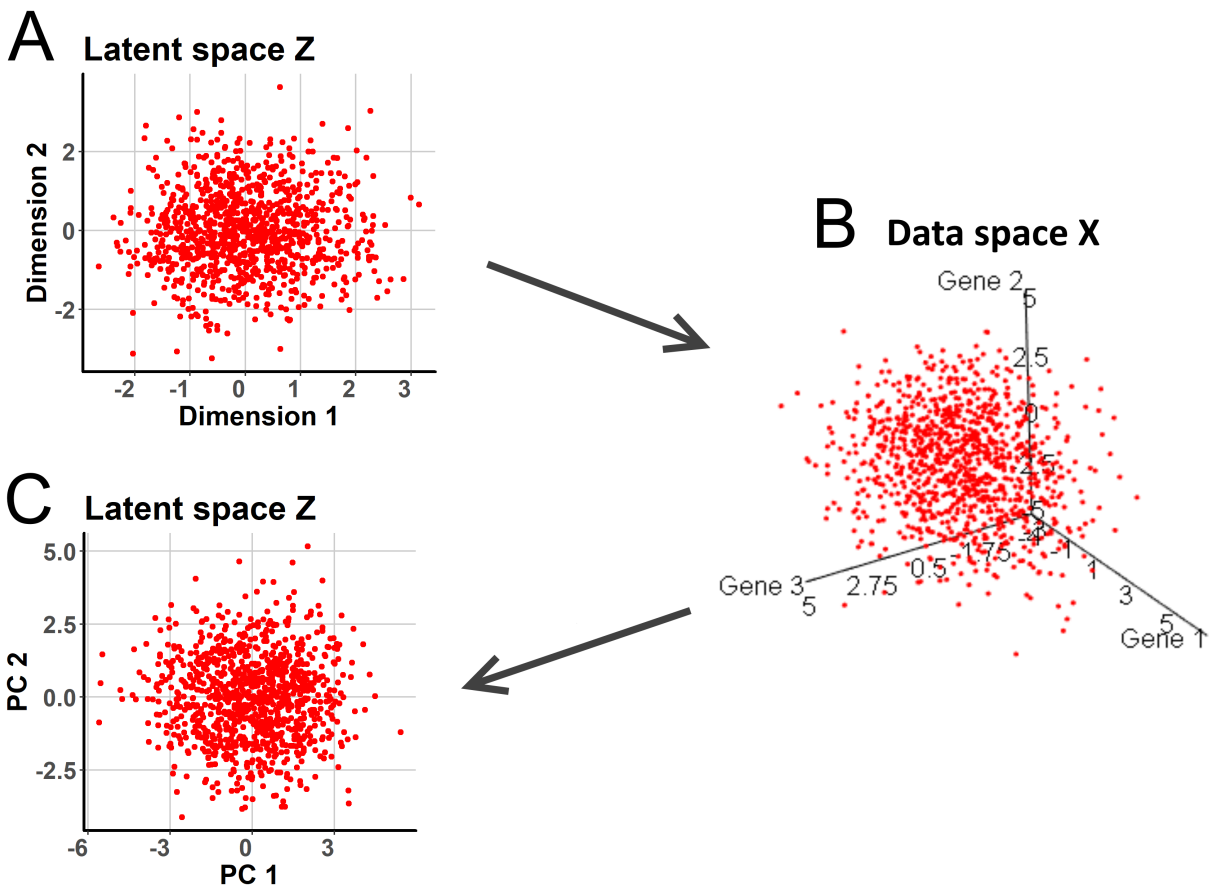

Figure 4: Example of dimensionality reduction with PPCA. (a) In this example 1000 cells are randomly distributed in a 2 dimensional latent space Z. (b) PPCA-based transformation with a linear mapping to a 3 dimensional space consisting of 3 genes. (c) PPCA is a dimensionality reduction technique, thus PPCA reduces the number of dimensions from a high-dimensional space to low-dimensional space Z, where Z illustrates the first 2 principal components (PCs) after dimensionality reduction. The parameters of the linear transformation from high-dimensional space to low-dimensional space Z are estimated by maximum likelihood estimation.

to data space can be defined as:

$$p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z}, \tag{3}$$

with,

$$\mathbf{z} \sim Normal(\mathbf{0}, \mathbf{I}_k) \tag{4}$$

$$\mathbf{x}|\mathbf{z} \sim Normal(\mathbf{W}\mathbf{z} + \boldsymbol{\mu}, \sigma^2\mathbf{I}_d) \tag{5}$$

Analogy to the previous equations, we get:

$$\mathbf{x} \sim Normal(\boldsymbol{\mu}, \mathbf{W}\mathbf{W}^T + \sigma^2\mathbf{I}_d) \tag{6}$$

To calculate the coordinates in latent space $\mathbf{z}$ (Figure 4C) from the coordinates in data space $\mathbf{x}$ (Figure 4B), we have to estimate the unknown parameters $\mathbf{W}$, $\boldsymbol{\mu}$ and σ^2 with maximum likelihood estimation.

The log likelihood is written in the form:

$$L = \sum_{n=1}^{N} ln \int p(\mathbf{x}_n|\mathbf{z}_n)p(\mathbf{z}_n)d\mathbf{z}_n \tag{7}$$

where N denotes the number of cells. The values $\mathbf{z}_n$ are assumed to be missing. Expected values of $\mathbf{z}_n$ can be obtained by calculation of the posterior distribution $\mathbf{z}_n$ with Bayes theorem, given the observed $\mathbf{x}_n$ and model parameters:

$$p(\mathbf{z}|\mathbf{x}) = \frac{p(\mathbf{x}|\mathbf{z})p(\mathbf{z})}{p(\mathbf{x})} \tag{8}$$

The expected values of $\mathbf{z}_n$ can be obtained in the E-step of the EM algorithm in the form:

$$\langle \mathbf{z}_n \rangle = \mathbf{M}^{-1}\mathbf{W}^T(\mathbf{x}_n - \boldsymbol{\mu}) \tag{9}$$

$$\langle \mathbf{z}_n \mathbf{z}_n^T \rangle = \boldsymbol{\sigma}^2 \mathbf{M}^{-1} + \langle \mathbf{z}_n \rangle \langle \mathbf{z}_n^T \rangle \tag{10}$$

where $\mathbf{M} = \mathbf{W}^T\mathbf{W} + \sigma^2\mathbf{I}_k$ and $\boldsymbol{\mu}$ is given by the sample mean of $\mathbf{x}_n$, written as:

$$\tilde{\boldsymbol{\mu}} = \frac{1}{N}\sum_n \mathbf{x}_n \tag{11}$$

The M-step of the EM algorithm maximizes the log-likelihood to give:

$$\tilde{\mathbf{W}} = \mathbf{SW}(\sigma^2\mathbf{I} + \mathbf{M}^{-1}\mathbf{W}^T\mathbf{SW})^{-1} \tag{12}$$

$$\tilde{\sigma}^2 = \frac{1}{d}\,\mathrm{Tr}(\mathbf{S} - \mathbf{SWM}^{-1}\tilde{\mathbf{W}}^T) \tag{13}$$

We refer to Bishop & Tipping (1998), Section 2 for a detailed explanation of equations 12 and 13.

2.3 Hierarchical mixture of probabilistic principal component analyzers

One of the advantages of PPCA versus PCA is that PPCA can be extended to a (Gaussian) mixture of PPCA models. A mixture of PPCAs allows to model more complex data structures by using multiple Gaussian densities. Figure 5 illustrates an example of a Gaussian mixture model (GMM) trained using EM. This example is closely related to a mixture of PPCAs, only each component in the mixture defines a Gaussian density model instead of a PPCA model. We use this example to explain the main steps of training a GMM, which we later extend with additional steps for training a mixture of PPCAs. So first we explain the steps of training a GMM using EM. The initial GMM in Figure 5A can be written as:

$$p(\mathbf{x}) = \sum_{i=1}^{C} \pi_i p(\mathbf{x}|i) \tag{14}$$

where C denotes the number of components in the mixture model and π_i the mixing coefficients of the different mixture components $p(\mathbf{x}|i)$. Mixing coefficient π_i has the constraint:

$$\sum_i \pi_i = 1 \tag{15}$$

Per component each data point gets a value assigned corresponding to the probability of belonging to that specific model (Figure 5B). These values can be calculated and updated in the E-step of the EM algorithm using this equation:

$$R_{ni} = \frac{\pi_i p(\mathbf{x}_n|i)}{\sum_{i'} \pi_{i'} p(\mathbf{x}_n|i')} \tag{16}$$

where i denotes the component in the mixture. After the first E-step the obtained probabilities from (16) are substituted in the M-step of the EM algorithm to obtain new values for the mixing coefficients $\tilde{\pi}_i$ and the $\tilde{\boldsymbol{\mu}}_i$ and $\tilde{\boldsymbol{\Sigma}}_i$ of the Gaussian distributions, we get:

$$\tilde{\pi}_i = \frac{1}{N} \sum_n R_{ni} \tag{17}$$

$$\tilde{\boldsymbol{\mu}}_i = \frac{\sum_n R_{ni} \mathbf{x}_n}{\sum_n R_{ni}} \tag{18}$$

$$\tilde{\boldsymbol{\Sigma}}_i = \frac{\sum_n R_{ni} (\mathbf{x}_n - \tilde{\boldsymbol{\mu}}_i)(\mathbf{x}_n - \tilde{\boldsymbol{\mu}}_i)^\top}{\sum_n R_{ni}} \tag{19}$$

Updating these values results in a better fit of the GMM to the data (Figure 5C). In the end the EM algorithm iterates until convergence (Figure 5D-F).

In case of a mixture of PPCAs each component in the mixture defines a single latent variable model with parameters $\mathbf{W}_i$, $\boldsymbol{\mu}_i$, and σ_i^2. In contrast to the ~~previous explained~~ EM algorithm a two stage EM algorithm is needed to obtain new values for $\mathbf{W}_i$ and σ_i^2. Namely, after the first E-step (16) the obtained probabilities are substituted in the first stage M-step of the EM-algorithm to obtain new values for $\tilde{\pi}_i$ and $\tilde{\boldsymbol{\mu}}_i$ obtained from (17) and (18). These new values are then substituted into (9) and (10) to calculate new expectations for the posterior distribution of $\mathbf{z}_{ni}$ in a second E-step. Finally, these values can be substituted into (12) and (13) to maximize in a second stage M-step the values of $\mathbf{W}_i$ and σ_i^2. Models are again initialized with the updated model parameters, which results in a better fit to the data ~~(Figure 5C)~~. The algorithm iterates until convergence ~~(Figure 5D-F). It is important to mention that the first stage M-step and second stage E-step both are not visualized in Figure 5. Namely, these steps should occur in between Figure 5B and Figure 5C.~~ For a detailed explanation how to extend PPCA equations (9), (10), (12) and (13) to equations for a mixture of PPCAs, we refer to Section 3 of Bishop & Tipping (1998).

Algorithm 1: Expectation maximization algorithm for a mixture of PPCAs

initialize model parameters $\mathbf{W}_i$, $\boldsymbol{\mu}_i$, σ_i, π_i;
while *model parameters not converged* **do**
  first stage E-step: compute $\mathbf{R}_i$;
  first stage M-step: update $\tilde{\pi}_i$, $\tilde{\boldsymbol{\mu}}_i$ with use of $\mathbf{R}_i$;
  second stage E-step: compute $\langle \mathbf{z}_{ni} \rangle$ with use of new $\tilde{\boldsymbol{\mu}}_i$ and compute $\langle \mathbf{z}_{ni} \mathbf{z}_{ni}^T \rangle$ from $\langle \mathbf{z}_{ni} \rangle$;
  second stage M-step: update $\tilde{\mathbf{W}}_i$, $\tilde{\sigma}_i^2$;
end

The mixture of PPCAs can be extended to a hierarchical mixture of PPCAs (hmPPCA). There the top level is a single PPCA model and the second level is a mixture of PPCA models. The user initializes the means $\boldsymbol{\mu}_i$ of the PPCA models in the mixture of the second level by interactively placing centers in the top level plot. Model parameters $\mathbf{W}_i$, σ_i^2 are initialized according to Appendix D of Bishop & Tipping (1998). Model parameter π_i is initialized by calculating the proportion of data points which are assigned to their nearest selected center in original data space. Once these model parameters are initialized the EM algorithm converges to a local maximum (Algorithm 1). Again the user can decide to extend the hierarchy to a third level. Mathematical details of extending to a third level and further can be found in Section 4 of Bishop & Tipping (1998).

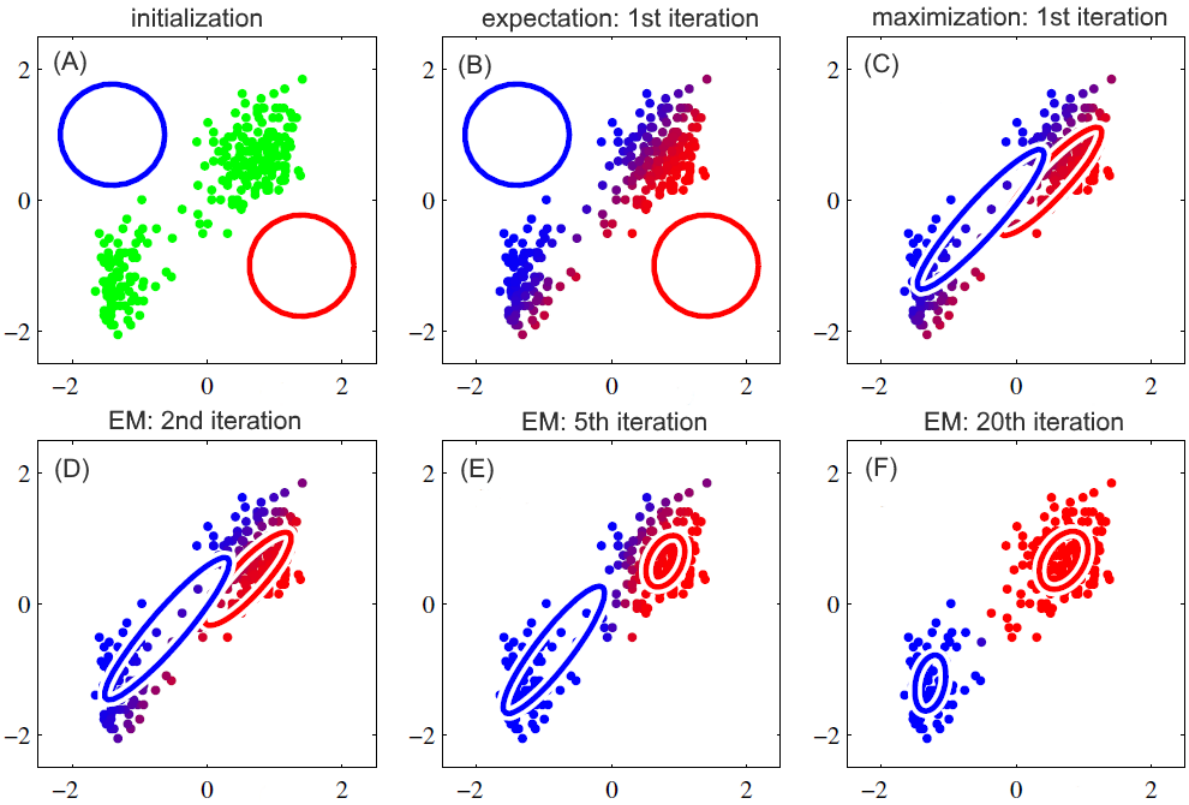

Figure 5: Illustration of the EM algorithm for a 2-component Gaussian mixture model. Data points are visualized in green and the two Gaussian densities in red and blue. (a) Two Gaussian density models are initialized. (b) In the expectation step points are softly assigned to the models based on their posterior probabilities. Each data point is colored with a gradient ~~equally~~ to the posterior probability of a data point for belonging to one of the density models. (c) In the maximization step model parameters are updated. (d-f) EM algorithm iterates until convergence. Figure has been copied from Bishop (2006).

2.4 Data

Our approach is to test the performance of a hierarchical mixture of PPCAs against PCA and t-SNE. We used several scRNA-seq datasets to compare performance of these algorithms in performing dimensionality reduction. We generated two simulated scRNA-seq datasets with the function *splatSimulateGroups* from the R package *splatter* (version 1.2.1) (Zappia et al., 2017). The following parameters can be set for this simulation:

- group.prob - The probabilities that cells come from particular groups
- de.prob - Probability that a gene is differentially expressed in each group
- de.facLoc - Location (meanlog) parameter for the differential expression factor log-normal distribution
- de.facScale - Scale (sdlog) parameter for the differential expression factor log-normal distribution.

An overview of our simulation settings can be found in Table 2. We set the *seed* to 1 in order to keep results reproducible. Varying the location and scale parameters for the differential expression factor log-normal distribution results in more heterogeneous group clusters for low values and well defined clusters for high values.

We also obtained two experimental scRNA-seq datasets. The first dataset was obtained from Darmanis et al. (2015)[1]. 466 single cells were captured from human and embryonic brain tissue. Expression levels of 22088 genes were measured. The authors obtained an unbiased cell classification using a Gaussian mixture model, where the number of components (clusters) was selected using the Bayesian information criterion (R package *mclust*). This resulted in 10 distinct cell types (unbiased groups) annotated according to Table 1. The downloaded dataset contained only 9 cell types, because hybrid cells (unbiased group 2)

Table 1: Annotation of the unbiased groups obtained after hierarchical clustering performed by Darmanis et al. (2015)

Cluster	**Annotation**
Unbiased group 1	Oligodendrocyte precursor cells (OPC)
Unbiased group 2	Hybrid
Unbiased group 3	Oligodendrocytes
Unbiased group 4	Astrocytes
Unbiased group 5	Microglia
Unbiased group 6	Unannotated
Unbiased group 7	Neurons
Unbiased group 8	Endothelial
Unbiased group 9	Fetal replicating
Unbiased group 10	Fetal quiescent

and unnannotated cells (unbiased group 6) were combined into a single cell type annonated as hybrid cells. The second dataset was obtained from Nestorowa et al. (2016)[2]. This dataset consists of 1656 hematopoietic stem and progenitor cells. Single cells were isolated into three gates based on protein expression levels. This resulted in progenitor (Prog) cell types, hematopoietic stem and progenitor cell (HSPC) types and encompassing long-term hematopoietic stem cell (LT_HSC) types. They measured expression levels of 4774 genes. For all datasets we removed genes with an expression level of zero in all cells and transformed counts to $log2$ scale with an offset of 1.

To perform PCA we applied *prcomp* using the R package *stats* (version 3.4.3). The following settings were used, $centering = True$ and $scaling = False$. For t-SNE we applied *Rtsne* from the R package *Rtsne* version (0.12). We explained in the Introduction that with t-SNE often a PCA preprocessing step is needed to find an interpretable representation of the data. With the $initial\_dims$ parameter in t-SNE the dimensionality of the data can be reduced to a given number. Without setting this parameter t-SNE gave only scattered low-dimensional representations. We set this parameter for all datasets to $initial\_dims = 50$. A parameter which also needs tuning is the perplexity parameter. The parameter is an estimate of the expected size of each cluster (van der Maaten & Hinton, 2008). We tested the optimal perplexity value qualitatively by eye in a range from 5 to 70, with steps of 5. The values we chose for the different datasets can be found in Table 2. The check for duplicated data points was set to $check\_duplicates = False$. Different solutions can be found with t-SNE, because the algorithm is non-deterministic. Therefore, we set the *seed* to 1 when running *Rtsne*.

2.5 Sampling

In some cases the PhiVis algorithm was not able to find an optimal log likelihood solution. This was the case for larger datasets with a high number of genes. We illustrate this with an example in Figure 6, where the left plot shows that after several iterations an optimal solution could be found, while in the right plot this is not the case. Finding the mathematical solution for this problem did not fit in the scope of this project. We therefore solved this issue by down sampling the number of genes for both

[1] https://hemberg-lab.github.io/scRNA.seq.datasets/human/brain/
[2] http://blood.stemcells.cam.ac.uk/single_cell_atlas.html

Table 2: Overview of the 4 different datasets we used to test PCA, t-SNE and hmPPCA. The number of genes after removing genes with an expression level of zero for all cells are between single parentheses. The number of genes after sampling the scRNA-seq data (see Section 2.5) are between double parentheses. Also the simulations setting for the datasets Splatter 1 and Splatter 2 can be found in this table.

Name	**Nr. cells**	**Nr. genes**	**Nr. cell types**	**Group. prob**	**De. prob**	**De. facLoc**	**De. facScale**	**Perplexity**
Splatter 1	500	400 (395)	5	0.2, 0.2, 0.2, 0.2, 0.2	0.05, 0.05, 0.05, 0.05, 0.05	3	0	30
Splatter 2	600	400 (395)	6	0.24, 0.12, 0.10, 0.02, 0.37, 0.15	0.20, 0.20, 0.20, 0.20, 0.20	0.1	0.4	20
Darmanis	466	22088 (21630) ((216))	9	-	-	-	-	35
Nestorowa	1656	4774 (4773) ((239))	3	-	-	-	-	70

the Darmanis dataset and Nestorowa dataset. First we sorted genes on variance of expression levels in decreasing order. Then we selected the top fraction of genes with the highest variance. Namely, for the Darmanis dataset a fraction of 0.01 of the 21630 genes left after removing genes with zero expression for all cells. For the Nestorowa dataset we selected a fraction of 0.05 of the 4773 genes left after removing genes with zero expression for all cells. For consistency purpose we applied PCA and t-SNE to the same sampled datasets.

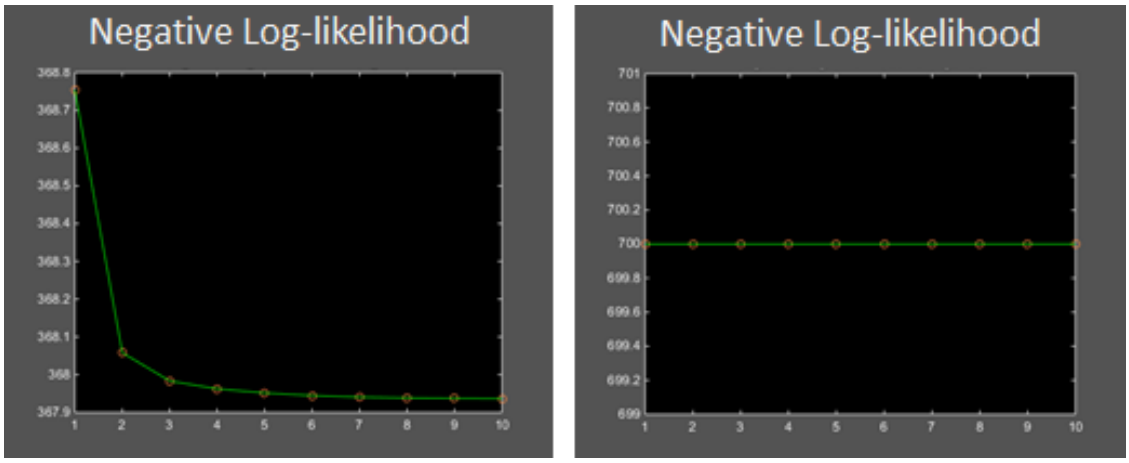

Figure 6: Calculations of the negative log likelihood, where on the left an optimal solution is found and on the right the solution did not converge to an optimum.

2.6 Performance evaluation

All tested dimensionality reduction algorithms created 2-dimensional visualization spaces from the high-dimensional scRNA-seq data. We adapted a method from Pierson & Yau (2015) to objectively assess the quality of the resulting 2-dimensional representations in terms of cell type separability after dimensionality reduction. The rationale behind this method is that hidden cell type substructures are more difficult to separate in a 2D plot, which decreases the ability to separate the different cell types. When training a prediction model on complex data with hidden substructures, the model will have more difficulty in finding decision boundaries segregating the data into different groups. Eventually this leads to lower accuracies in predicting the correct cell type. We trained a multinomial logistic regression model on the x and y coordinates of cells in visualization space to predict their corresponding cell type. Accuracy of PCA, t-SNE and for each 2-dimensional space of a single component in the mixture of PPCAs was obtained using leave one out cross validation (LOOCV).

With hmPPCA we also want to objectively quantify the accuracy of a whole level in the hierarchy. To obtain the accuracy for the whole level some extra calculations were needed. Namely, as we explained above points are softly assigned to each component in a mixture model of PPCAs. Figure 3C shows points with a certain gradient, this gradient corresponds to the probability of belonging to that specific model. To objectively measure the quality of a whole level in the hierarchy we have to take these probabilities into account. To obtain a weighted accuracy of a whole level in the hierarchy, we multiplied the predicted outcome, i.e. 1 for a correctly predicted cell type and 0 for a incorrectly predicted cell type, for each cell with the corresponding probabilities (Equation 16). For instance, take a point at the third level (Figure 3C) belonging to the middle plot with a probability of 0.8 and to the right plot with a probability of 0.2 (and therefore a probability of 0 for the left component). If the cell type is correctly predicted in the middle plot and incorrectly predicted in the right plot, the accuracy for this point in the entire level is then $0.8 \times 1 + 0.2 \times 0 = 0.8$. Summing the weighted accuracies for all points at this level gives a weighted accuracy per level.

We made some extra changes to the LOOCV script, namely in cases with only a single class to train the model, we automatically assigned an accuracy of 1 (multiplied with the weights per data point, in case of obtaining the weighted accuracy of a level in hmPPCA). Also, to prevent leave one out training errors when one of the classes is of size 1, we removed points belonging to the class of size 1 from the training set and automatically predicted the majority class. We trained a *multinom* model using the R package *nnet* version (7.3-12). The *multinom* function calls a neural net (*nnet* function). With a neural net the numbers of hidden layers and the range of output probabilities can be set. We set the number of hidden layers to zero ($size = 0$). We limited the output probabilities to values between zero and one by setting $softmax = True$. One of the advantages of this model is that in the case of hmPPCA we can give weights to the data points corresponding to the calculated probabilities per data point (Equation 16). Another advantage is that the model can predict multilevel labels, which in our case are the different cell types per dataset.

To visualize the decision boundaries of the trained model in 2D we needed to train an extra *multinom* model with different settings. Namely, this model is not trained in LOOCV, because plotting decision boundaries of all models trained in LOOCV in one 2D plot is unclear. Furthermore, we removed cells with a probability lower than 0.05 from the training set to make plots for the hmPPCA model easier to interpret. PhiVis has a similar approach of removing points in its visualizations, but we did not extensively test if a more stringent threshold is more accurate. The reported accuracy for these 2D plots are calculated separately in LOOCV. It is important to mention that this strategy differs from the calculation of the earlier explained weighted accuracy per level. Namely, for calculation of the weighted accuracy per level all cells were included in the training set. We created the visualization with code adapted from the *plotLearnerPrediction* function of the R package MLR (version 2.11).

2.7 Estimation of cluster centers

In PhiVis an option is available to color points based on a given label vector, illustrated in Figure 3. This label vector comes from an earlier performed clustering on the data. To give more guidance to the user,

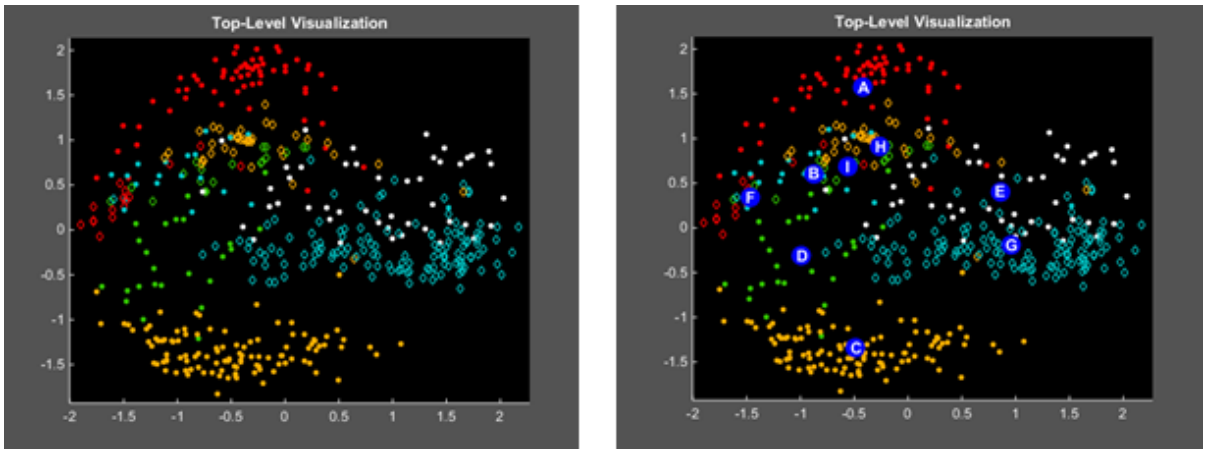

Figure 7: Illustration of the top level plot after dimensionality reduction of the Darmanis dataset. Dimensionality reduction is performed with the PhiVis package (Hierarchical mixture of PPCA). The left figure illustrates an example where 9 different cell types are colored based on labels obtained from a clustering algorithm (Darmanis et al., 2015). The right figure illustrates the same example, only now the calculated centers per cluster are also plotted with symbols A - ~~G~~.

we extended the functionality of coloring clusters with the possibility of plotting the centers per colored cluster. The x-y coordinates of each center were calculated by taking the weighted mean (according to 16) of principal component 1 and principal component 2 obtained from Equation 9. An example of the end result is given by Figure 7.

3 Results

In this section we discuss and compare the performance of PCA, t-SNE and a hierarchical mixture of PPCAs on 2 simulated scRNA-seq datasets and 2 experimental scRNA-seq datasets. Each dataset is discussed in a different subsection.

Abbreviations in figures: DR = Dimensionality reduction, CV = Cross validation, LOO = Leave one out, ML = Machine learning, MN = Multinomial logistic regression, ACC = Accuracy.

3.1 Dimensionality reduction on simulated scRNA-seq data: Splatter 1

The dataset Splatter 1 served as a test for confirmation if the dimensionality reduction techniques worked as expected. This dataset has therefore a simple structure with 5 well separated clusters. If the algorithms worked properly and captured the most important structures in the data, we also would expect 5 well separated clusters in 2-dimensional space. Quantification of the cell type separability would then reach an accuracy of 1. In Figure 8 2-dimensional spaces for Splatter 1 after dimensionality reduction with PCA and t-SNE are shown. We observed in both plots that PCA and t-SNE represented the data in 5 well separated clusters. The main difference between both plots is the distance between the clusters. With PCA the relative distance between points is preserved after dimensionality reduction. The aim of t-SNE is to display similar points (Euclidean distances expressed in conditional probabilities) close to each other and dissimilar points further from each other (van der Maaten & Hinton, 2008). This is the nonlinear component of t-SNE and a reason that distances between clusters in both plots differed. The top level plot in hmPPCA is a single PPCA model. We should therefore find a similar result as with PCA after dimensionality reduction. When we investigated the top level plot (Figure 9A), we indeed observed similar structures as with PCA (Figure 8A). The difference we found was in the orientation along the first

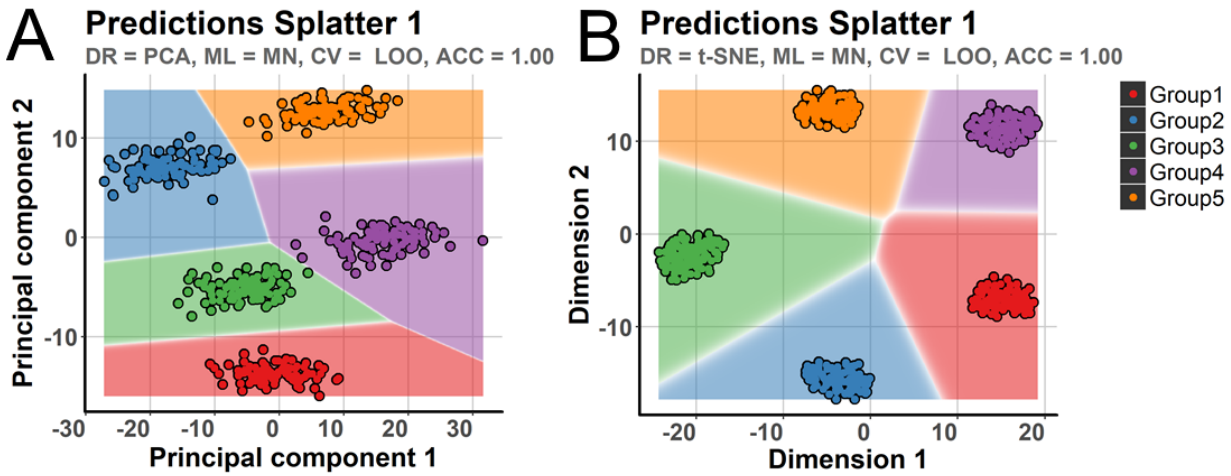

Figure 8: Visualization of simulated dataset Splatter 1. Dataset Splatter 1 consists of 500 cells and 5 cell types. These plots illustrate dimensionality reduction from 395 genes to 2 dimensions with PCA (a) and t-SNE (b).

PC corresponding to an eigenvector with opposite sign. Both 2-dimensional spaces reached an accuracy of 1 in cell type separability. The difference between both models is that we can create a hierarchy with hmPPCA. We found 5 well distinguished cell structures at the top level and therefore initialized 5 centers to create a hierarchical mixture model with 5 PPCA models. Individual models in the mixture are visualized in Figure 9B. There are no hidden structures in the data, therefore no hidden cell type structures were revealed at this second level. Obviously, this led to the same accuracies as before, namely for all plots an accuracy of 1.

We conclude from these results that all three dimensionality reduction algorithms worked as expected on a simple simulated dataset.

3.2 Dimensionality reduction on simulated scRNA-seq data: Splatter 2

To further test the different algorithms we simulated a more complex dataset named Splatter 2. This dataset is noisier with a more complex structure. We therefore expected that not all 6 cell types could be revealed and explored in a 2D plot.

First we applied PCA to find the principal components with the largest variation. The first 2 principal components are visualized in Figure 10A. We can clearly distinguish 2 cell types, namely Group 1 and Group 5. On the other hand, the structures of the other 4 cell types are difficult to explore and are strongly heterogeneous. Consequently, perfect cell type separability is not possible, reaching an accuracy of 0.76. With t-SNE we tried to improve results. This technique is nonlinear and tries to group similar cells close to each other. This can be helpful in revealing some of the hidden structures of Group 2,3,4,6. By investigating Figure 10B we found an improvement in the number of cell types we can explore, namely also Group 6 could be better distinguished from the other cell types. Another improvement is found with Group 2, where a large part of the cells from that cell type are grouped together. Clearly, t-SNE revealed that Group 2 and Group 6 are separate cell types. Investigating this dataset with t-SNE is therefore an improvement to PCA, reaching an accuracy of 0.86.

We have illustrated that this dataset has hidden structures in 2D plots after dimensionality reduction with PCA. With t-SNE some of the hidden structures were revealed, but we could not clearly observe the simulated cell types Group 3 and Group 4. We therefore wanted to investigate if hmPPCA could reveal these hidden structures. In Figure 12A we can find the top level plot. Here we placed three centers for initialization of the mixture of PPCAs at the second level. Center 1 and center 3 are placed on the group

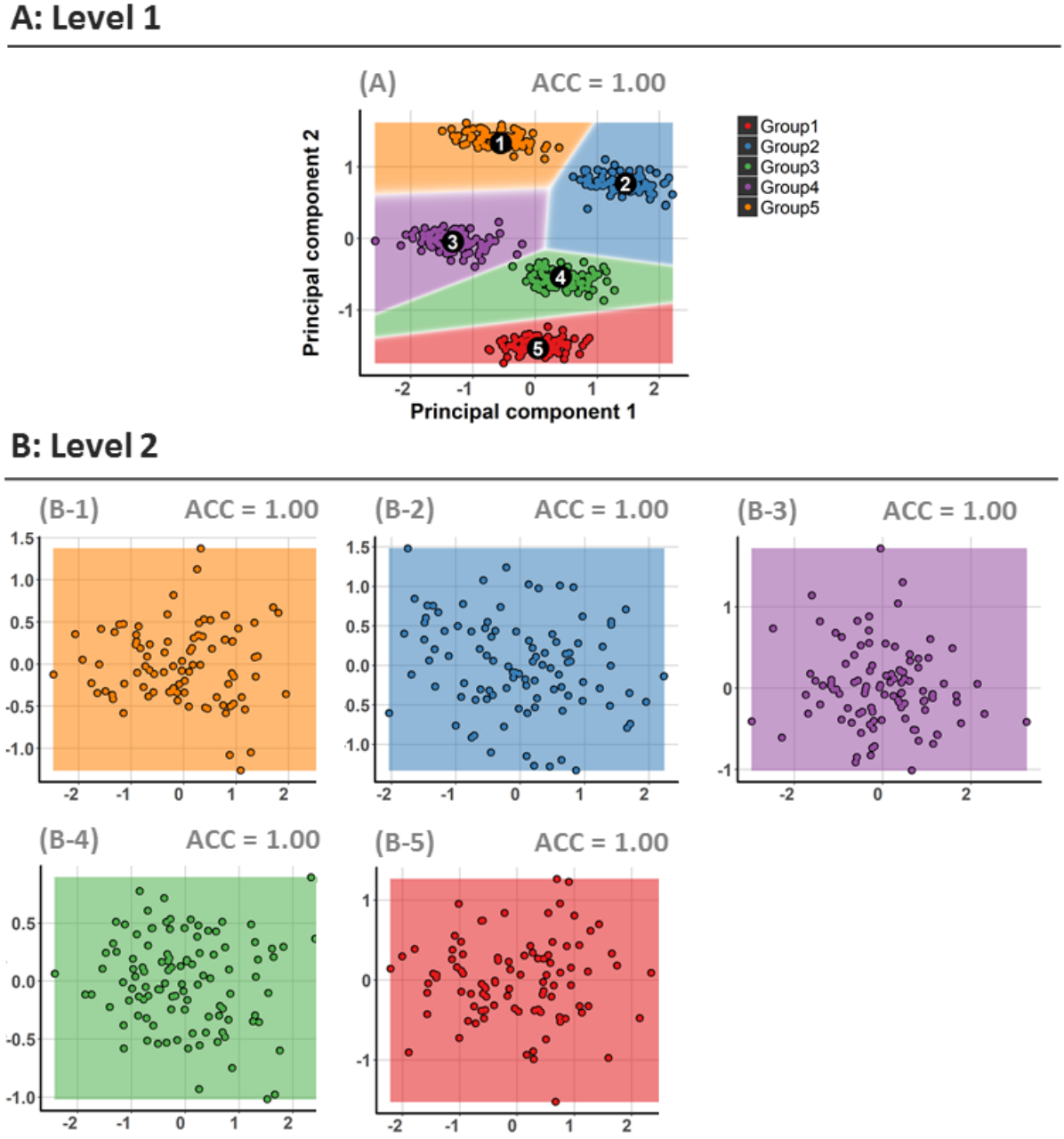

Figure 9: Visualization of simulated dataset Splatter 1. Dataset Splatter 1 consists of 500 cells and 5 cell types. The number of dimensions is reduced from 395 genes to 2 dimensions with hmPPCA. Level 1 visualizes the first 2 principal components after dimensionality reduction with PPCA. By initializing 5 new centers in the top level plot, a second level is created consisting of 5 components. The letter between parentheses for each plot correspond to the level of the hierarchy. Numbers correspond to the user selected centers at the top level plot indicated with numbers 1 - 5.

of cells we could clearly separate from the rest of the cells. Center 2 was placed on the heterogeneous structure to further explore that part of the dataset. In Figure 12B we can find the results of the second

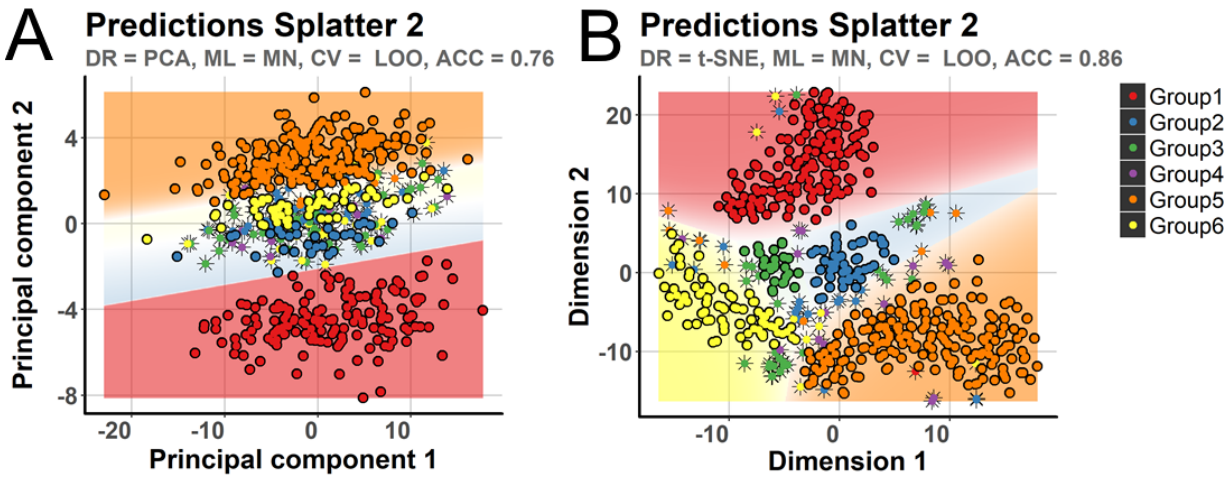

Figure 10: Visualization of simulated dataset Splatter 2. Dataset Splatter 2 consists of 600 cells and 6 cell types. These plots illustrate dimensionality reduction from 395 genes to 2 dimensions with PCA (a) and t-SNE (b). Cells highlighted with * are cell types misclassified by the classifier.

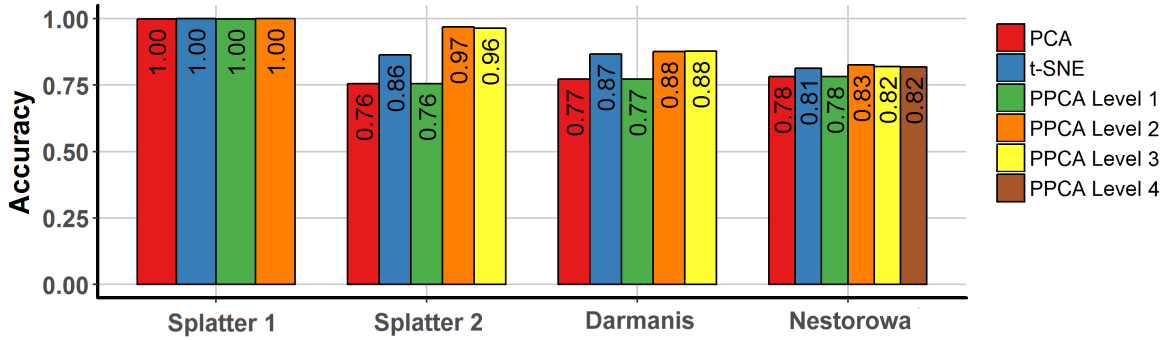

Figure 11: For each dataset we calculated accuracies of predicting the correct cell type in the 2-dimensional space after dimensionality reduction.

level of hmPPCA. Figure 12B-1 shows some interesting results. Namely, Group 6 is clearly separated from the other cell types in contrast to the top level plot. In Figure 12B-2 we observe a clear separation of Group 2 and Group 3. Where t-SNE (Figure 10) already showed a small improvement in separating these cell types, cells were still not really grouped together. To further separate the cell types the hierarchy was extended to a third level, this level can be found in Appendix 7.1. With these results hmPPCA proved its strength, namely we can explore the data in different directions to find possible hidden structures. All three plots in the second level increased in accuracy, ranging from 0.90 - 1.00. The overall weighted accuracy of this level is 0.97 (Figure 11).

We can conclude that hmPPCA improved both on PCA and t-SNE. We also showed that for simulated scRNA-seq datasets hmPPCA can reveal hidden structures which could not be properly found with PCA and t-SNE. In the next section we will investigate if hmPPCA achieved similar results on experimental scRNA-seq datasets.

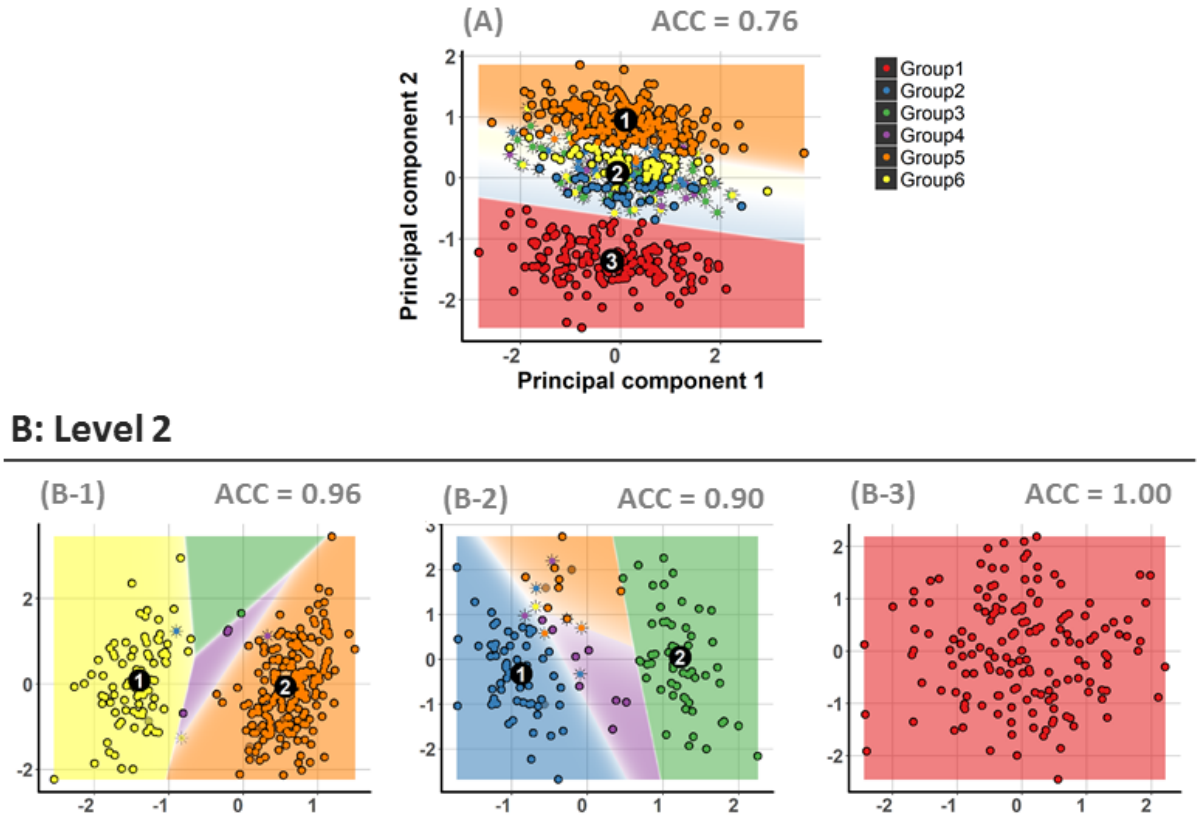

Figure 12: Visualization of simulated dataset Splatter 2. Dataset Splatter 2 consists of 600 cells and 6 cell types. The number of dimensions is reduced from 395 genes to 2 dimensions with hmPPCA. Level 1 visualizes the first 2 principal components after dimensionality reduction with PPCA. By initializing 3 new centers in the top level plot a second level is created consisting of 3 components. The letter between parentheses for each plot correspond to the level of the hierarchy. Numbers correspond to the user selected centers in the top level plot indicated with numbers 1 - 3. By again initializing new centers at the second level, the hierarchy can be extended to a third level. Level 3 can be found in Appendix 7.1. Cells highlighted with * are cell types misclassified by the classifier.

3.3 Dimensionality reduction on experimental scRNA-seq data: Darmanis

The dataset of Darmanis et al. (2015) consists of human brain tissue from 8 adult and 4 embryonic samples. Darmanis et al. performed a hierarchical clustering on the gene expression profiles without prior knowledge (unbiased). This resulted in 10 distinct cell types. Based on the top 20 enriched genes they were able to annotate 8 of the 10 clusters. Cells were annotated as astrocytes, oligodendrocytes, oligodendrocyte precursor cells (OPC), neurons, microglia, endothelial, replicating neuronal progenitors (fetal replicating) and quiescent newly born neurons (fetal quiescent) cells. One of the two unannotated clusters was enriched with a mixture of neuronal-, oligodendrocyte-, and OPC-specific genes making up a hybrid cluster. This mixture can be caused by contamination of the cells, or that OPCs partly express oligodendrocyte genes (Darmanis et al., 2015). The other unannotated cluster displayed characteristics of neurons and astrocytes. The authors ruled out the possibility of contamination. A possibility is that this cluster is a distinct undiscovered cell type. From these annotations we expect the hybrid cluster to

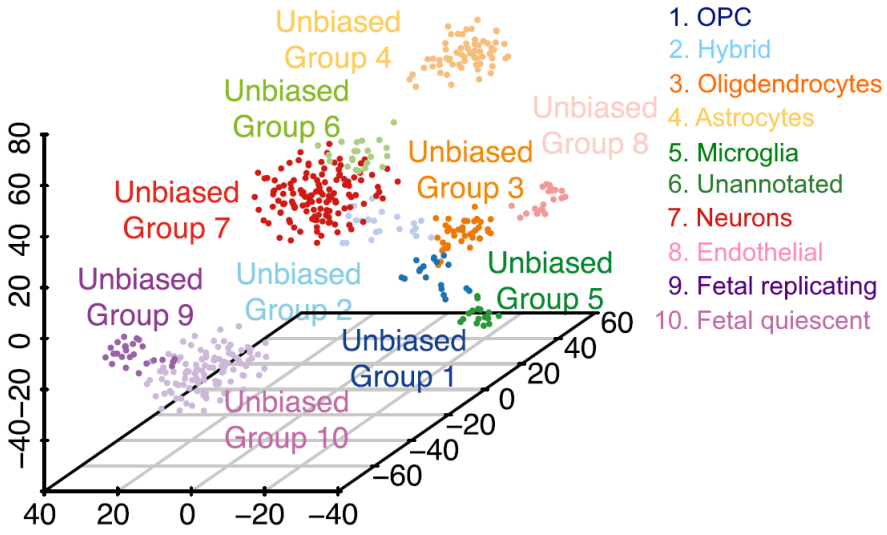

Figure 13: Overview of the scRNA-seq dataset from Darmanis et al. (2015). t-SNE plot created from 466 cells with 10 different cell types. Figure has been copied from Darmanis et al. (2015)

be positioned between the neuronal, oligodendrocyte and OPC clusters. We also expect the fetal clusters to have more similar gene expression profiles and therefore to be close together. For the unannotated cluster we expect a positioning between the neuron and astrocyte cluster. We indeed observe in the 3D plot (Figure 13) of Darmanis et al. (2015) that the hybrid cluster is positioned between neuronal, oligodendrocyte and OPC cells. We directly notice the problems of visualizing the results in this type of plot. We were able to roughly interpret the position of the hybrid cluster, but interpretation of distances between points and the amount of heterogeneity of the clusters is more difficult. We noticed the same for the unannotated cluster. Darmanis et al. (2015) hypothesized that this cluster could correspond to a distinct undiscovered cell type. Interpreting in this 3D plot how distinct the unannotated hybrid cells are from the neuron cells is difficult and almost impossible.

Also in this section we investigate if t-SNE and hmPPCA more clearly show the differences between these cell types. Unfortunately, we cannot investigate improvements in visualization for the unannotated cluster, because the downloaded dataset did not contain this cluster. In Figure 14 the results after dimensionality reduction with PCA and t-SNE are shown. In the PCA plot we can distinguish the astrocyte cells, fetal quiescent cells and partly the neuron cells and fetal replicating cells from the rest of the cells. In the middle we found a heterogeneous bulk of cells, from which we could not extract any information. For instance, we cannot distinguish if the hybrid cells are positioned in between the neuronal, oligodendrocyte and OPC cells. Reaching an accuracy of 0.77 indicated there is room for improvement of visualizing the different cell types.

The noisy bulk in the middle of the plot is quite similarly structured to the structure we found with PCA for Splatter 2 (Figure 10). Darmanis et al. created the 3D plot (Figure 13) with viSNE, which is a t-SNE Barnes-Hut implementation (van der Maaten, 2014). This implementation is built on the t-SNE algorithm with an acceleration in finding the best representation of the data in low-dimensional space. Visualization of this dataset with t-SNE should therefore lead to almost the same result as in Figure 13. The only difference is a 2D visualization versus a 3D visualization. If we compare Figure 13 to the t-SNE plot in Figure 14B we indeed observe that in both plots the fetal cells are grouped together with some distance to the rest of the cells. The same is observed for the hybrid and neuron cells. On the other hand also some differences are observed. In Figure 13 each cell type is grouped together and there is no scattering of individual cells over the plot, while in Figure 14B some individual cells are scattered over the plot. For instance, endothelial cells are not fully grouped together in Figure 14, while this is the case

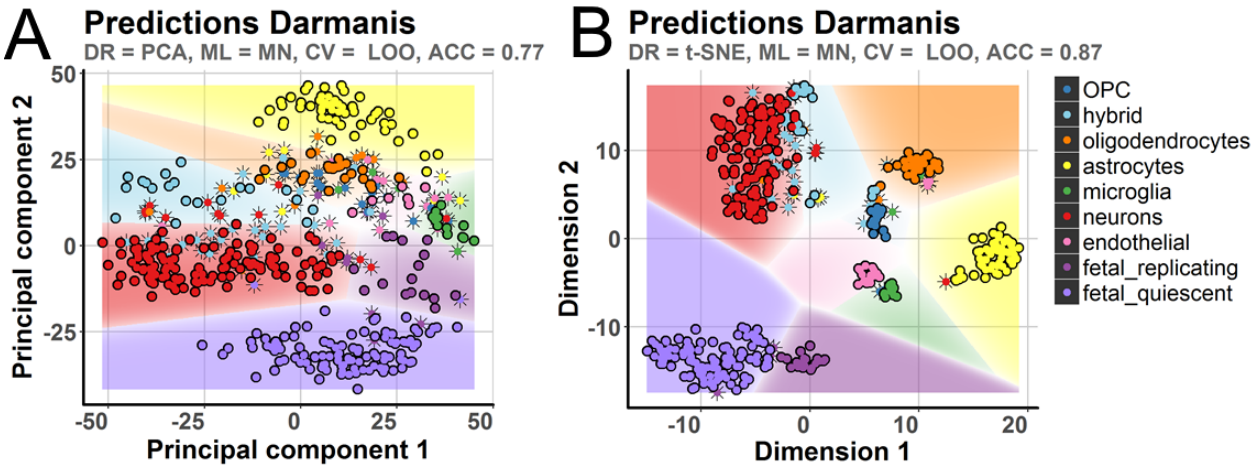

Figure 14: Visualization of experimental dataset of Darmanis et al. (2015). Dataset Darmanis consists of 466 cells and 9 cell types. These plots illustrate dimensionality reduction from 216 genes to 2 dimensions with PCA (a) and t-SNE (b). Cells highlighted with * are cell types misclassified by the classifier.

in Figure 13. Both low-dimensional spaces are created with a t-SNE algorithm, but settings between the algorithms can vary such as perplexity and the PCA preprocessing step (*initial_dim* parameter of the t-SNE algorithm) resulting in different representations in low-dimensional space. When we compare the t-SNE plot to the PCA plot, an improvement is observed in separating OPC and endothelial cells. We are also better able to interpret the distance of the hybrid cluster to rest of the cells. Namely, the hybrid cells are much closer to the neuron cells than to the oligodendrocyte cells and OPC cells. The hybrid and neuron cells are still packed and heterogeneous in this plot. We cannot conclude if this is caused by a heterogeneous structure, or caused by visualizing the results in a 2D plot. Investigating this structure in different directions of variance with hmPPCA should confirm if this structure is heterogeneous. For t-SNE cell type separability quantification improved to an accuracy of 0.87.

In Figure 15 the hierarchy created with hmPPCA is illustrated. In the top level plot (Figure 15A) we initialized 5 centers which extends the hierarchy to a second level. Similarly to Splatter 2 we expected that some hidden structures might be revealed, for instance the clusters of OPC and oligodendrocytes are quite heterogeneous in the top level plot. What we directly observe in Figure 15B-2 is that the OPC and oligodendrocytes are now separated from each other. In the t-SNE plot (Figure 14B) oligodendrocytes were also visualized separately from the rest of the cell types, therefore hmPPCA improves in visualization of these cell types compared to PCA. This is confirmed with an overall accuracy of 0.87 in the second level of hmPPCA (Figure 11). We found for the hybrid and neuron cells a similar result as with t-SNE. Namely, in Figure 15B-3 they are grouped together as a heterogeneous structure with some distance to the OPC and oligodendrocytes cells. In the t-SNE plot we were able to easily interpret distances between all the cell types. This becomes more difficult in Level 2 of the hierarchy. Namely, the OPC and oligodendrocytes cells (Figure 15B-2) are not captured in the same PPCA model as the hybrid and neuron cells (Figure 15B3) making interpretation of distances between cell types more difficult. We further extended the hierarchy to a third level (Figure 15C) in an attempt to further explore the heterogeneous hybrid and neuron structure, but no clear separation of the hybrid and neuron cells was observed in Figure 15C-3.1. In Figure 13 it was not clear if hybrid and neuron cells were separated from each other or more heterogeneous. Our visualization confirmed that these cell types are not separated from each other. This can indicate that the hybrid cluster mainly consists of neuron type of cells. Extending the hierarchy with more levels is not always an improvement in terms of visualization and cell type separability. An interesting result is found in the extension from Figure 15B-3. Namely, in one of the extensions (Figure 15C-3.2) the accuracy dropped to a value of 0.54. We can find the cause of these drop in accuracy in

the capture of mostly the heterogeneous part of the data of Figure 15B-3, while most of neuron cells are captured in the PPCA model of Figure 15C-3.1. This brings us to the caveat of the hmPPCA algorithm, when does one stop extending the levels? Per level we calculated a weighted accuracy quantifying cell type separability. In Figure 11 the overall accuracies per level are shown. We observed almost no improvement from level 3 to level 4 for the Darmanis dataset. This indicates cell type separability is not improved by extending the hierarchy, thus a convergence in the weighted accuracy per level indicates when a local optimal solution is found.

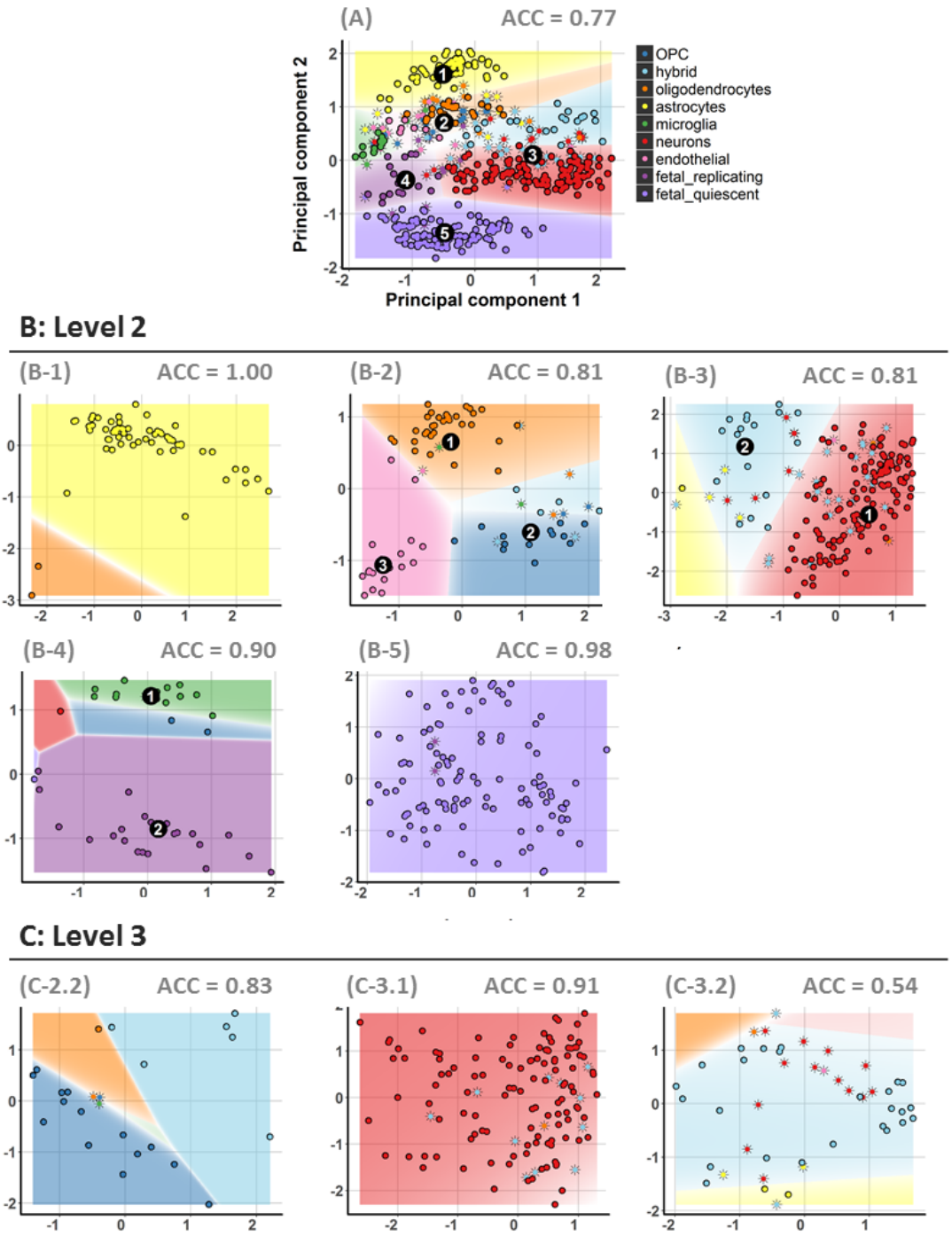

Figure 15: Visualization of experimental dataset of Darmanis et al. (2015). Dataset Darmanis consists of 466 cells and 9 cell types. The number of dimensions is reduced from 216 genes to 2 dimensions with hmPPCA. Level 1 visualizes the first 2 principal components after dimensionality reduction with PPCA. By initializing 5 new centers in the top level plot a second level is created consisting of 5 components. The first number between parentheses for each plot at Level 2 correspond to user selected centers in the top level plot indicated with numbers 1 - 5. By again initializing new centers in the second level, the hierarchy can be extended to a third level. The second number between parentheses for each plot at Level 3 correspond to user selected centers in Level 2. In Level 3 we only showed the 3 most interesting plots. The full level can be found in Appendix 7.2. Cells highlighted with * are cell types misclassified by the classifier.

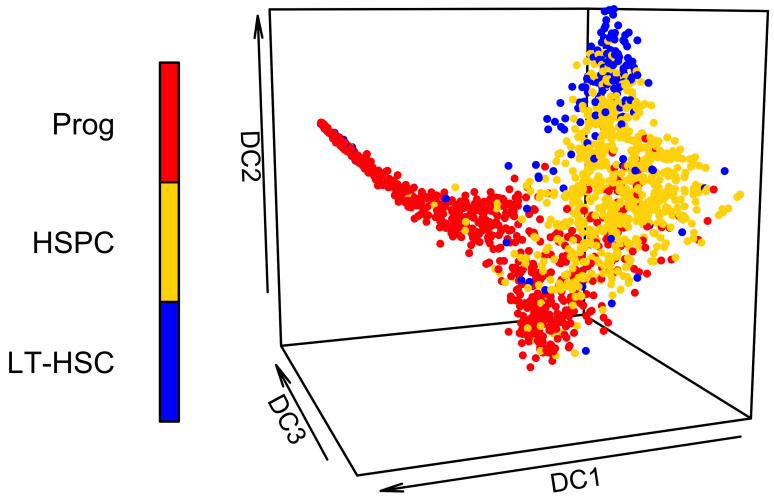

Figure 16: Overview of the scRNA-seq dataset from Nestorowa et al. (2016). Diffusion map created from 1656 cells with 3 different cell types. Namely, Progenitor (Prog), Hematopeiotic stem and Progenitor cells (HSPC) and long-term Hematopeiotic stem cells (LT-HSC). Figure has been copied from *Gene and protein expression in adult haematopoiesis* (2016).

From these results we can conclude that with hmPPCA we are better able to explore the cell types compared to PCA on an experimental dataset. On the other hand we found no improvement in cell type separability compared to t-SNE. With t-SNE we were better able to interpret distances between cell types. But with hmPPCA we could explore a structure more intensively at a deeper level, which helped in understanding the heterogeneity of the data.

3.4 Dimensionality reduction on experimental scRNA-seq data: Nestorowa

The last dataset we tested is much larger in number of cells as the previous datasets, namely 1656 hematopoietic stem and progenitor cells (HSPCs). Nestorowa et al. (2016) collected HSPCs from bone marrow of 10 female 12-week-old mice. Isolating the single cells into three gates based on protein expression levels resulted in Progenitor cell types, hematopoietic stem and progenitor cell (HSPC) types and encompassing long-term hematopoietic stem cell (LT_HSC) types. HSPCs and LT_HSCs were first captured in the same gate, LT_HSCs were additionally sorted into a separate gate. We expect that these cell types are more similar to each other than to the Progenitor cells. Nestorowa et al. (2016) visualized the 3 cell types in a 3D plot obtained using a diffusion map method (Nestorowa et al., 2016). From Figure 16 we can roughly conclude that Progenitor cells are more distant from LT_HSCs than from HSPCs. But we can clearly observe the problems arising when visualizing such a large dataset. The overall structure of the 3 different cell types can be explored, but for instance many of the Progenitor cells are hidden and could be much closer to the LT_HSCs than we can observe in this figure. PCA and t-SNE will both have a similar problem when representing such a large dataset in a 2D plot. We therefore expect that exploring the results with an hierarchical approach would reveal some of the hidden substructures.

In Figure 17 the first 2 dimensions are visualized after dimensionality reduction with PCA and t-SNE. Both plots are really dense and packed of cells making exploration of the results quite challenging. For instance, LT_HSC cells are hidden in both plots and can not be properly explored. t-SNE achieved a higher accuracy than PCA, but overall the results are similar. Nestorowa et al. (2016) performed their analysis on a different cell type classification, which was obtained using a hierarchical clustering algorithm on expression levels. Directly comparing the visualized clusters of Figure 17 with the clusters in Figure

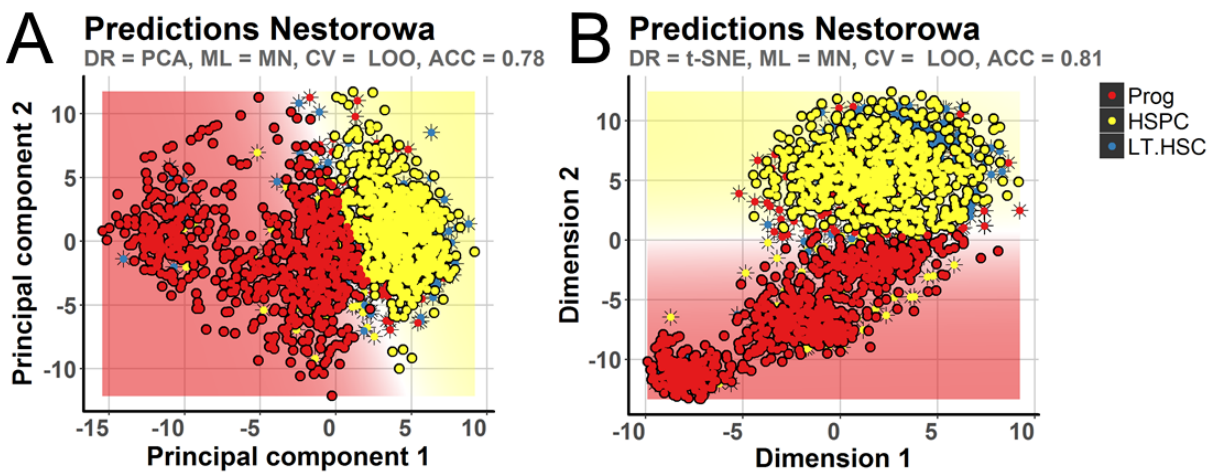

Figure 17: Visualization of experimental dataset of Nestorowa et al. (2016). Dataset Nestorowa consists of 1656 cells and 3 cell types. These plots illustrate dimensionality reduction from 239 genes to 2 dimensions with PCA (a) and t-SNE (b). Cells highlighted with * are cell types misclassified by the classifier.

16 is therefore not possible.

Hierarchically exploring this dataset might reveal some of the hidden structures. In Figure 18 the hierarchy after dimensionality reduction with hmPPCA is shown. In the top level we initialized 2 centers in an attempt to reveal the structure of the hidden LT_HSC cells. Figure 18B-1 mainly consists of Progenitor cells. Based on a full red colored decision boundary we expect that all cells are classified as Progenitor cells. Hence, an accuracy of 0.89 indicates that around 89% of the cells are Progenitor cells. In Figure 18B-2 we found all three cell types in large numbers. This makes further understanding distances between the different cell type structures still a challenge. In the end a small increase in weighted accuracy of level 2 is observed from 0.78 to 0.83 (Figure 11). We again initialized centers to extend the hierarchy to a third level. The LT_HSC cells are mainly visualized in 18C-2.1, while Progenitor cells are mainly visualized in 18C-2.2. This indicates these cell types are more distant from each other than both are from the HSPC cells. Further investigating the data in a fourth level scattered mostly HSPC cells over all plots. Investigating the accuracy per level showed that only an extension to the second level increased the overall level accuracy. The overall level accuracies converged when we extended to a third and fourth level.

We can conclude that hmPPCA showed a small improvement in visualization of such a large amount of cells compared to PCA and t-SNE. We were able to further explore the data in different directions of variance. This gave an indication that LT_HSCs were closer to HSPCs than to Progenitor cells. But in the end no clear separation of cell types was achieved and data remained heterogeneous.

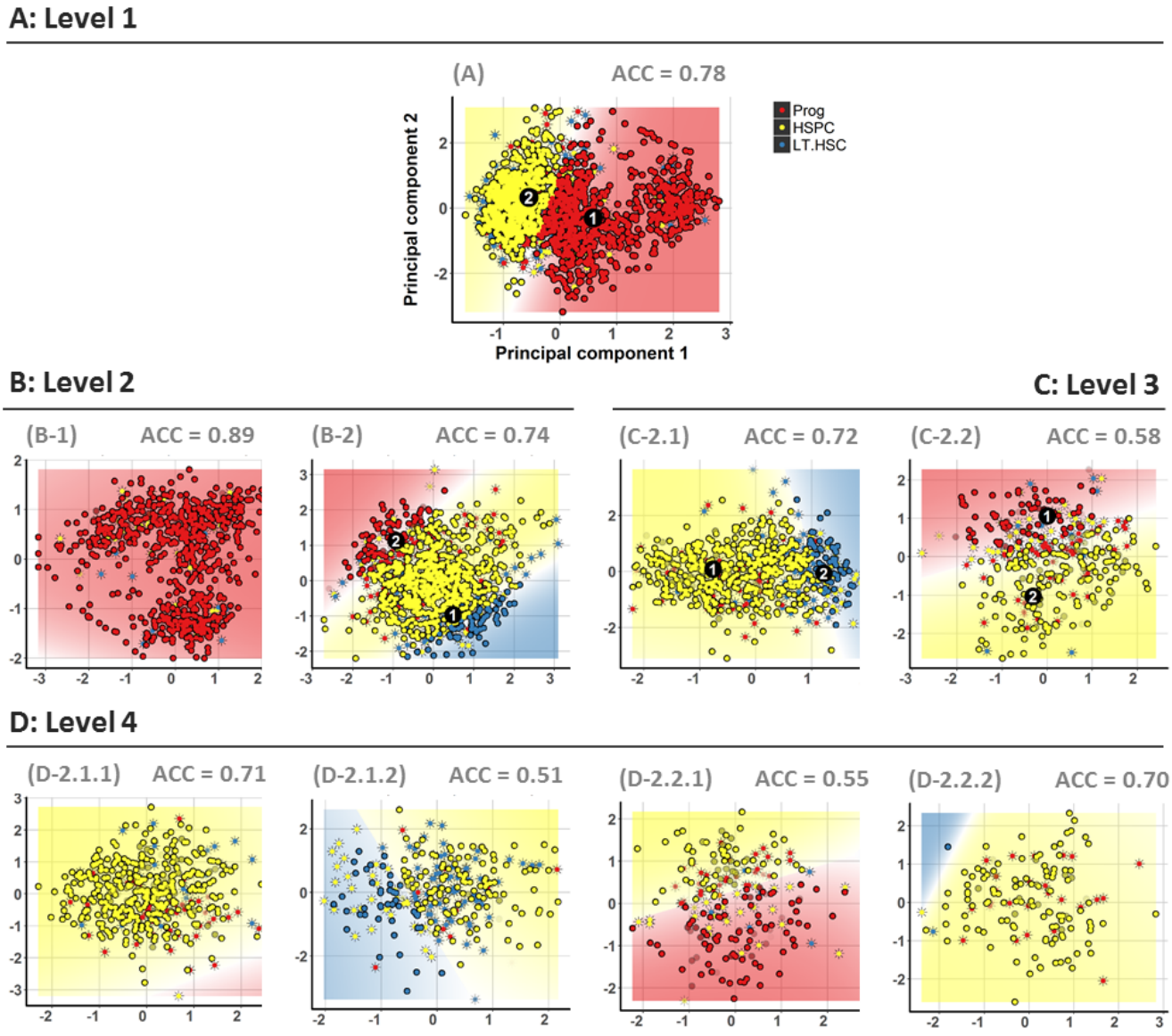

Figure 18: Visualization of experimental dataset of Nestorowa et al. (2016). Dataset Nestorowa consists of 1656 cells and 3 cell types. The number of dimensions is reduced from 239 genes to 2 dimensions with hmPPCA. Level 1 visualizes the first 2 principal components after dimensionality reduction with PPCA. By initializing 2 new centers in the top level plot a second level is created consisting of 2 components. The numbers between parentheses for each plot at Level 2 correspond to user selected centers in the top level plot indicated with numbers 1 - 2. By again initializing new centers in plot B-2, plot C-2.1 and plot C-2.2 the hierarchy can be extended to a third and fourth level. The second and third number between parentheses for each plot at Level 3 and Level 4 correspond to user selected centers in the preceding levels respectively. The full hierarchy of Level 3 and Level 4 can be found in Appendix 7.3. Cells highlighted with * are cell types misclassified by the classifier.

4 Discussion

In this research we compared three dimensionality reduction techniques. Each has shown its utility and strength in reducing the number of dimensions on high-dimensional scRNA-seq datasets. PCA already proved on many occasions to be effective in reducing the number of dimensions (Kolodziejczyk et al., 2015). In our case visualization of the first 2 principal components gave indeed the ability to explore results of the high-dimensional simulated and experimental datasets, but we noticed that substructures were often hidden in the 2D plots. This was confirmed with the lowest accuracy scores in quantification of cell type separability for the different datasets. By applying nonlinear dimensionality reduction with t-SNE, cells were better separated from each other in 2D space, which improved the ability of exploring the different cell types. First we found scattered and suboptimal low-dimensional representations when we applied t-SNE on datasets where all genes were included. Therefore, we manually tuned the *perplexity* parameter and *initial_dim* parameter (PCA preprocessing step in t-SNE algorithm) to find the optimal representation in 2-dimensional space. Exploring these parameter settings is time consuming and the optimal representation could even be missed. For instance, in the 2-dimensional space of Splatter 2 (Figure 10B) the cells of Group 3 were scattered, while in the 2-dimensional space of hmPPCA (Figure 12B-2) these cells grouped together. In this case the 2-dimensional representation of hmPPCA has the tendency to be a better representation than the representation of t-SNE. Different representations of the data can lead to a different interpretation of the similarity between cells and cell types. An option is to perform parameter optimization with a grid search based on minimization of the Kullback-Leibler divergence to find the optimal representation.

With the dataset of Darmanis et al. we showed that the choice of a dimensionality reduction technique has consequences for the interpretation of the similarity between cell types. While Darmanis et al. classified and visualized the hybrid cells as a separate cell type, we showed with t-SNE and hmPPCA that this cluster was much closer and overlapping with the neuron cell type. Hence, the choice of dimensionality reduction can give different results, which can affect the interpretation of cell type differences and how cell types are related. We also noticed that distances were more easily to interpret in a t-SNE plot than in hierarchical mixture of PPCA plots. This also has influence on interpretation of the relation and distance between cells and cell types. Depending on the chosen technique this can lead to a different understanding of how cells and cell types are related.

It is important to emphasize that objectively measuring the quality of a low-dimensional representation is difficult. Our approach quantified cell type separability in low-dimensional space. As explained earlier, a prediction model can more easily find decision boundaries in 2D plots with a low amount of hidden substructures. Consequently, accuracies in predicting the correct cell type per cell are higher when there are less hidden substructures. This measure proved its effectivity in our research. In addition to this we integrated the method into the algorithm of hmPPCA. Namely, we were able to quantify the overall weighted accuracy of cell type separability at each level in the hierarchy of hmPPCA. When extending the hierarchy of hmPPCA with a new level, we could evaluate if this leads to an improvement in cell type separability. When this accuracy converges to a maximum, we concluded that no further hidden structures in the data could be revealed when extending the hierarchy to a new level. After convergence of the accuracy one could suggest that a heterogeneous structure at the top level is also heterogeneous at the lowest level in the hierarchy. We observed this when we visualized the Darmanis dataset. Namely, we found at the lowest level in the hierarchy that the neuron and hybrid cell structures were heterogeneous.

While the discussed techniques are adequate for the task of dimensionality reduction, they were not specifically designed for scRNA-seq data. One prominent problem in scRNA-seq is the occurrence of dropout events leading to zero counts. These dropout events are caused by lowly expressed genes that are difficult to capture or due to sampling stochasticity (Lun et al., 2016). This can cause problems when applying dimensionality reduction techniques such as PCA and hmPPCA. Because PCA is not specially designed for data with a high abundance of zeros (Pierson & Yau, 2015) and hmPPCA assumes that data is normally distributed (Bishop & Tipping, 1998; van der Maaten & Hinton, 2008). This is not the case with a high abundance of zeros, making the algorithms less accurate in calculation of the low-dimensional representation. Several methods have especially been developed to solve this problem for

scRNA-seq data (Pierson & Yau, 2015; van Dijk et al., 2017). One approach is called Zero Inflation Factor Analysis (ZIFA), which explicitly takes the zero inflation into account in the model. This model is build on a latent variable model based on factor analysis (FA). The authors of ZIFA reasoned that genes with lower expression levels have a higher probability to be affected by dropout events than genes with higher expression levels (Pierson & Yau, 2015). Hence, the underlying assumption of the zero-inflation model is that the dropout rate for a gene depends on the expected expression level of a gene in a population (Pierson & Yau, 2015). Similar to PPCA, the FA framework of ZIFA assumes that points are mapped from a low-dimensional space to points in high-dimensional space with the addition of Gaussian distributed noise. ZIFA adds an extra layer to this step, in which some of the measurements in high-dimensional space are set to zero based on the dropout rate of the zero-inflation model. With ZIFA one wants to find the low-dimensional representation from the high-dimensional representation. Therefore similarly to PPCA, the model parameters are estimated via an EM algorithm. This also applies to the relationship between the dropout rate and expected gene expression, which are estimated in the EM algorithm. We can conclude that the setup of the ZIFA algorithm has a lot of similarities to the PPCA algorithm. An interesting approach is to extend the ZIFA algorithm into a hierarchical mixture of ZIFA. An advantage of this model would be the integrated hierarchical aspect combined with more accuracy in calculating the low-dimensional representation specifically for scRNA-seq data. MAGIC (Markov Affinity-based Graph Imputation of Cells) is another approach specifically designed for the dropout problem in scRNA-seq data (van Dijk et al., 2017). The idea of MAGIC is to restore the original structure of the data by imputation of the dropout values. These dropout values are imputed based on gene expression levels of similar cells. The authors of MAGIC tested the algorithm on datasets where the ground truth is known. They showed that MAGIC is able to recover developmental trends and improved cluster-specific gene expression (van Dijk et al., 2017). An advantage of MAGIC is that the algorithm can work standalone as a preprocessing step before dimensionality reduction. Therefore, incorporating MAGIC into the hmPPCA pipeline is likely to be much easier compared to the extension of ZIFA to a hierarchical ZIFA algorithm.

We showed that in some cases nonlinear t-SNE outperformed linear PCA in terms of cell type separability. We also showed that investigating scRNA-seq data with a linear hierarchical approach improved the ability of finding substructures in the data. Efforts have been made to develop nonlinear hierarchical algorithms. It is therefore interesting to compare hmPCCA to these nonlinear hierarchical approaches. Hierarchical GTM (Generative Topographic Mapping) is an extension of the PhiVis package, where the researchers incorporated nonlinear projection manifolds (Tino & Nabney, 2002). Another interesting nonlinear approach is hierarchical t-SNE (Pezzotti et al., 2016). This algorithm has the power of t-SNE to find an interpretable representation of the data in low-dimensional space, while the data can be explored hierarchically in more detail. We expect that in some cases a linear hierarchal approach is more suitable for a specific dataset, while in other cases a nonlinear hierarchical approach could reveal interesting underlying structures in the data. It is important to test these approaches on a large number of datasets.

With our research we created a pipeline to simulate scRNA-seq data and test performance of several dimensionality reduction techniques. A next step is to automate this pipeline which enables to test the dimensionality reduction techniques on a larger scale. Testing performance of PCA could easily be automated. If t-SNE is extended with a grid search option to find optimal values for the *perplexity* parameter and *initial_dim* parameter, then also this algorithm could be automated. Still the influence of automatically choosing these parameters should be investigated. Testing hmPPCA automatically is more difficult, because of the interactive component of the algorithm. We extended the algorithm with more guidance for the user by showing centers of apparent clusters. An option is to use these calculated centers to initialize centers at a new level in the hierarchy. The process could then be automated until the weighted accuracy per level converges to a maximum. This approach will not always lead to the optimal representation of the data and there is a danger of overfitting the model to the chosen clustering. Nevertheless, it gives a good indication of the power of a hierarchical approach to a specific dataset.

5 Conclusion

The various results in this research indicate that hmPPCA is a useful addition in exploring high-dimensional scRNA-seq datasets. The algorithm outperformed conventional non hierarchical dimensionality reduction techniques in several cases. With hmPPCA the user is able to interactively explore the data in more detail, which can reveal hidden substructures. We extended the algorithm with more functionality to give the user more guidance in the interactive steps of the algorithm. In addition to this we delivered a pipeline to compare different dimensionality reduction techniques and objectively assess their performance in terms of cell type separability. This pipeline can be further automated to assess performance on a larger set of simulated scRNA-seq datasets. One should consider to investigate the power of nonlinear hierarchical approaches versus hmPPCA in exploring substructures of the data. Also the common problem of dropout events in scRNA-seq data should be taken into account by applying suitable algorithms for imputing those values. In the end we can conclude that hmPPCA is a useful exploration tool and could be considered when exploring a scRNA-seq dataset.

6 Acknowledgments

I would like to thank Perry for the tremendous support and assistance during my master internship. In the beginning of the project I was mainly focused on understanding of the mathematical background of the subject. Perry explained which parts were important to understand, was patient when explaining complex parts and really encouraged me to fully understand the mathematical equations. I liked the discussions during the project about the choice of direction for the research we needed to make, they were most of the time serious but there was always time to joke. The feedback which Perry gave to my thesis drafts and presentations were of much detail. This really brought my work to a next level. Perry many thanks for this interesting journey of diving into the science of biology and machine learning.

I would also like to thank the team of the Bioinformatics Laboratory. They gave me useful feedback during the meetings and were very helpful and supportive during my internship.

Of course, I cannot forget to thank the fellow students who were at the same time working on their internship. Especially Jose and Michael, I really liked your company and the many jokes we made in the so called aquarium student room.

References

Anchang, B., Hart, T. D. P., Bendall, S. C., Qiu, P., Bjornson, Z., Linderman, M., ... Plevritis, S. K. (2016). Visualization and cellular hierarchy inference of single-cell data using SPADE. *Nature Protocols*, *11*(7), 1264-1279.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Secaucus, NJ, USA: Springer-Verlag New York, Inc.

Bishop, C. M., & Tipping, M. E. (1998). A hierarchical latent variable model for data visualization. *IEEE Transactions on Pattern Analysis and Machince Intelligence*, *20*(3), 281–293.

Bose, S., Wan, Z., Carr, A., Rizvi, A. H., Vieira, G., Pe'er, D., & Sims, P. A. (2015). Scalable microfluidics for single-cell RNA printing and sequencing. *Genome Biology*, *16*(1), 120.

Brennecke, P., Anders, S., Kim, J. K., Kolodziejczyk, A. A., Zhang, X., Proserpio, V., ... Heisler, M. G. (2013). Accounting for technical noise in single-cell RNA-seq experiments. *Nature Methods*, *10*, 1093-1095.

Cannoodt, R., Saelens, W., & Saeys, Y. (2016). Computational methods for trajectory inference from single-cell transcriptomics. *European Journal of Immunology*, *46*(11), 2496-2506.

Darmanis, S., Sloan, S. A., Zhang, Y., Enge, M., Caneda, C., Shuer, L. M., ... Quake, S. R. (2015). A survey of human brain transcriptome diversity at the single cell level. *Proceedings of the National Academy of Sciences*, *112*(23), 7285-7290.

duVerle, D. A., Yotsukura, S., Nomura, S., Aburatani, H., & Tsuda, K. (2016). CellTree: an R/bioconductor package to infer the hierarchical structure of cell populations from single-cell RNA-seq data. *BMC Bioinformatics*, *17*(1), 363.

Gene and protein expression in adult haematopoiesis. (2016). Retrieved 2018-03-05, from `http://blood.stemcells.cam.ac.uk/single_cell_atlas.html`

Islam, S., Kjllquist, U., Moliner, A., Zajac, P., Fan, J.-B., Lnnerberg, P., & Linnarsson, S. (2011). Characterization of the single-cell transcriptional landscape by highly multiplex RNA-seq. *Genome Research*, *21*(7), 1160-1167.

Jaitin, D. A., Kenigsberg, E., Keren-Shaul, H., Elefant, N., Paul, F., Zaretsky, I., ... Amit, I. (2014). Massively parallel single-cell RNA-seq for marker-free decomposition of tissues into cell types. *Science*, *343*(6172), 776–779.

Jiang, L., Chen, H., Pinello, L., & Yuan, G.-C. (2016). GiniClust: detecting rare cell types from single-cell gene expression data with Gini index. *Genome Biology*, *17*(1), 144.

Klein, A. M., Mazutis, L., Akartuna, I., Tallapragada, N., Veres, A., Li, V., ... Kirschner, M. W. (2015). Droplet Barcoding for Single-Cell Transcriptomics Applied to Embryonic Stem Cells. *Cell*, *161*(5), 1187-1201.

Kolodziejczyk, A., Kim, J. K., Svensson, V., Marioni, J., & Teichmann, S. (2015). The technology and biology of single-cell RNA sequencing. *Molecular Cell*, *58*(4), 610 - 620.

Lukk, M., Kapushesky, M., Nikkila, J., Parkinson, H., Goncalves, A., & Huber, W. (2010). A global map of human gene expression. *Nature Biotechnology*, *28*(4), 322-324.

Lun, A. T., Bach, K., & Marioni, J. C. (2016). Pooling across cells to normalize single-cell RNA sequencing data with many zero counts. *Genome Biology*, *17*(1), 75.

Nestorowa, S., Hamey, F. K., Pijuan Sala, B., Diamanti, E., Shepherd, M., Laurenti, E., ... Göttgens, B. (2016). A single cell resolution map of mouse haematopoietic stem and progenitor cell differentiation. *Blood*, *128*(8), e20-e31.

Pezzotti, N., Höllt, T., Lelieveldt, B. P., Eisemann, E., & Vilanova, A. (2016). Hierarchical stochastic neighbor embedding. *Computer Graphics Forum (Proc. of EuroVis)*, *35*(3), 21-30.

Pierson, E., & Yau, C. (2015). ZIFA: Dimensionality reduction for zero-inflated single-cell gene expression analysis. *Genome Biology*, *16*(1), 241.

Rosenberg, A. B., Roco, C. M., Muscat, R. A., Kuchina, A., Sample, P., Yao, Z., ... Seelig, G. (2018). Single-cell profiling of the developing mouse brain and spinal cord with split-pool barcoding. *Science*, *360*(6385), 176-182.

Stegle, O., Teichmann, S. A., & Marioni, J. C. (2015). Computational and analytical challenges in single-cell transcriptomics. *Nature Reviews Genetics*, *16*(3), 133–145.

Su, A. I., Cooke, M. P., Ching, K. A., Hakak, Y., Walker, J. R., Wiltshire, T., ... Hogenesch, J. B. (2002). Large-scale analysis of the human and mouse transcriptomes. *Proceedings of the National Academy of Sciences*, *99*(7), 4465-4470.

Svensson, V., Vento-Tormo, R., & Teichmann, S. A. (2018). Exponential scaling of single-cell RNA-seq in the past decade. *Nature Protocols*, *13*, 599. (Perspective)

Tang, F., Barbacioru, C., Wang, Y., Nordman, E., Lee, C., Xu, N., ... Surani, M. A. (2009). mRNA-seq whole-transcriptome analysis of a single cell. *Nature Methods*, *6*(5), 377–382.

Tino, P., & Nabney, I. (2002). Hierarchical GTM: constructing localized nonlinear projection manifolds in a principled way. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*(5), 639-656.

van der Maaten, L. (2014). Accelerating t-SNE using tree-based algorithms. *Journal of Machine Learning Research*, *15*, 3221-3245.

van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, *9*, 2579-2605.

van Dijk, D., Nainys, J., Sharma, R., Kathail, P., Carr, A. J., Moon, K. R., ... Pe'er, D. (2017). MAGIC: A diffusion-based imputation method reveals gene-gene interactions in single-cell RNA-sequencing data. *bioRxiv:111591*.

Wang, Z., Gerstein, M., & Snyder, M. (2009). RNA-Seq: a revolutionary tool for transcriptomics. *Nature Reviews Genetics*, *10*(1), 57-63.

Yijyechern. (2014). *RNA-seq workflow.* Retrieved 2018-03-05, from `https://en.wikipedia.org/wiki/File:RNA-Seq_workflow-5.pdf`

Zappia, L., Phipson, B., & Oshlack, A. (2017). Splatter: simulation of single-cell RNA sequencing data. *Genome Biology*, *18*(1), 174.

Žurauskienė, J., & Yau, C. (2016). pcaReduce: hierarchical clustering of single cell transcriptional profiles. *BMC Bioinformatics*, *17*(1), 140.

7 Appendix

7.1 Splatter 2: level 3 of the hmPPCA hierarchy

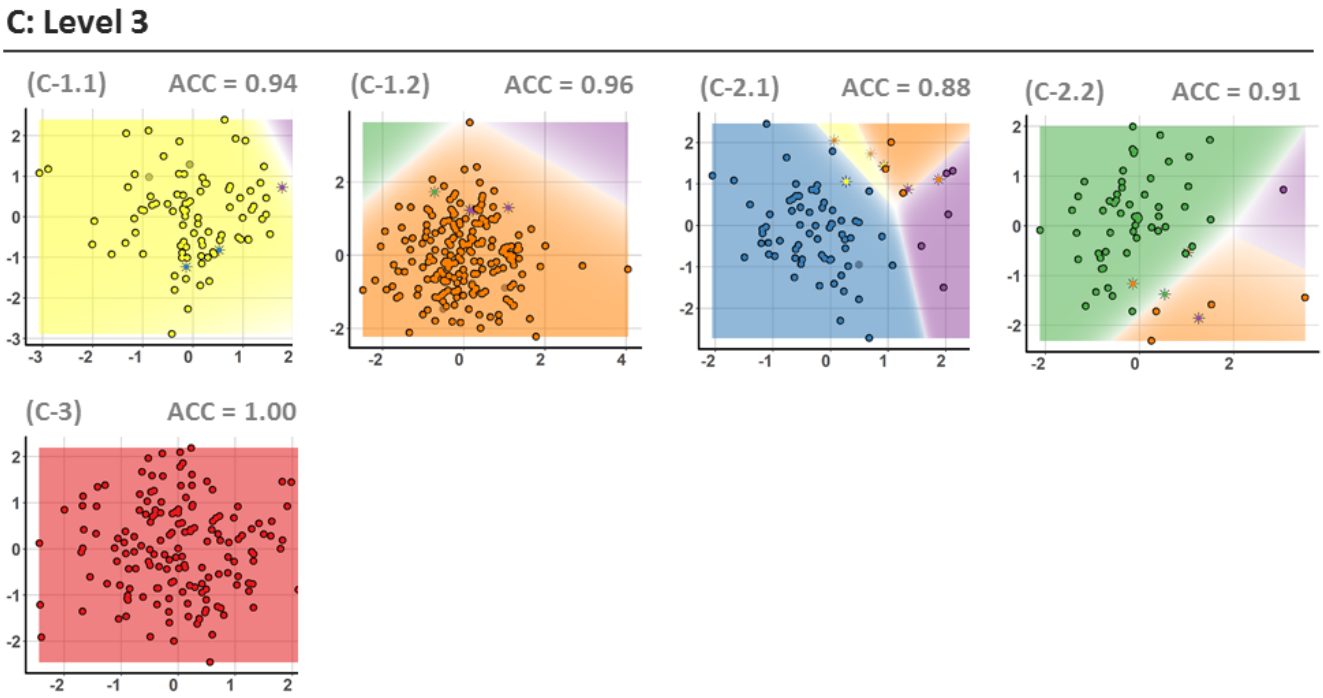

Figure 19: Visualization of simulated dataset Splatter 2. Dataset Splatter 2 consists of 600 cells and 6 cell types. The number of dimensions is reduced from 395 genes to 2 dimensions with hmPPCA. Here Level 3 is visualized. By initialization of 4 new centers at Level 2 of Figure 12 a third level is created consisting of 4 components. The letter between parentheses corresponds to the level of the hierarchy. Numbers correspond to the initialized center in preceding levels of the hierarchy (Figure 12). The first number refers to centers in Level 1 and the second number refers to centers in Level 2. Cells highlighted with * are cell types misclassified by the classifier.

7.2 Darmanis: level 3 of the hmPPCA hierarchy

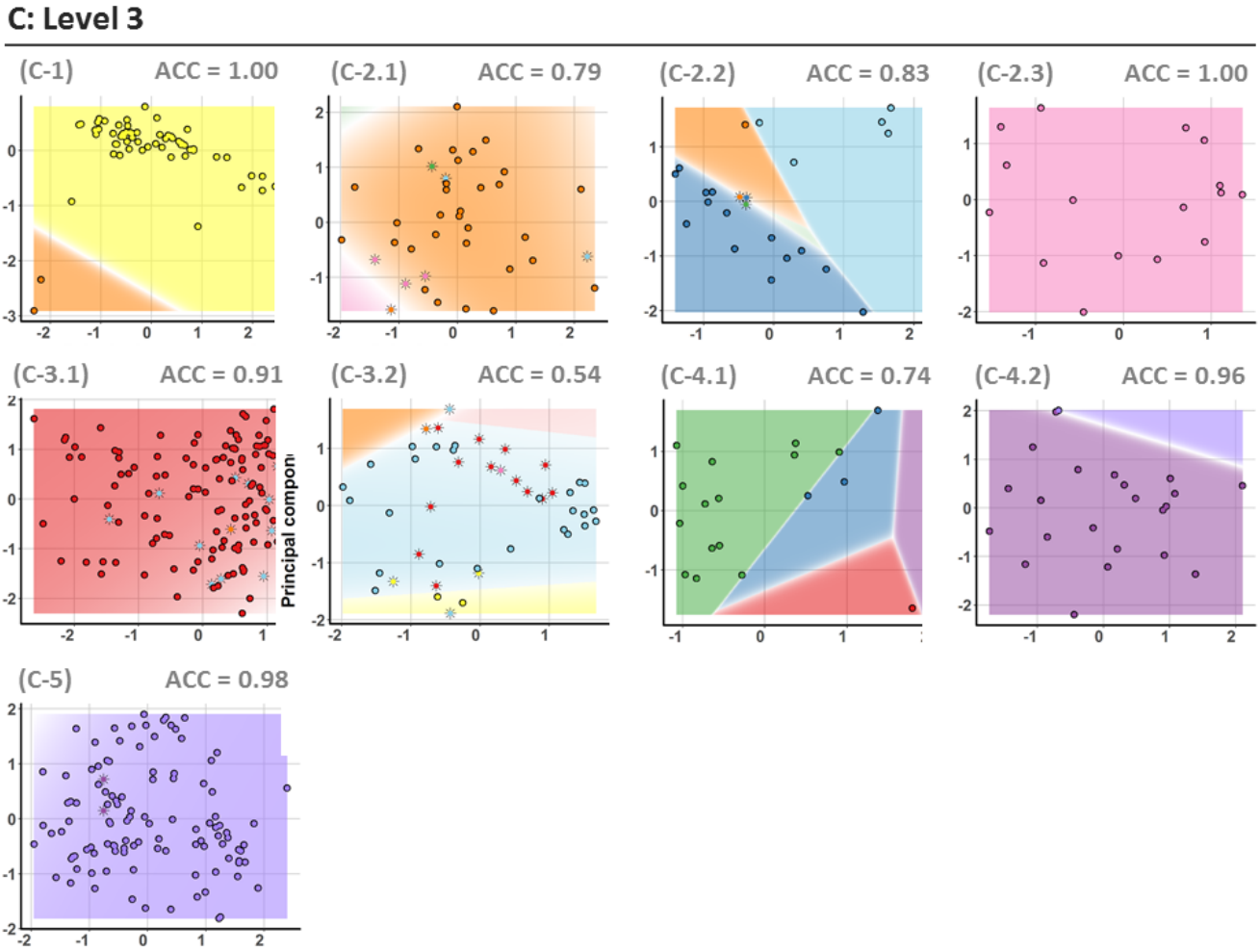

Figure 20: Visualization of experimental dataset of Darmanis et al. (2015). Dataset Darmanis consists of 466 cells and 9 cell types. The number of dimensions is reduced from 216 genes to 2 dimensions with hmPPCA. Here Level 3 is visualized. By initialization of 7 new centers at Level 2 of Figure 15 a third level is created consisting of 7 components. The letter between parentheses for each plot at Level 3 correspond to the level of the hierarchy. Numbers correspond to the user selected center in preceding levels of the hierarchy (Figure 15). The first number refers to centers in Level 1 and the second number refers to centers in Level 2. Cells highlighted with * are cell types misclassified by the classifier.

7.3 Nestorowa: level 3 and 4 of the hmPPCA hierarchy

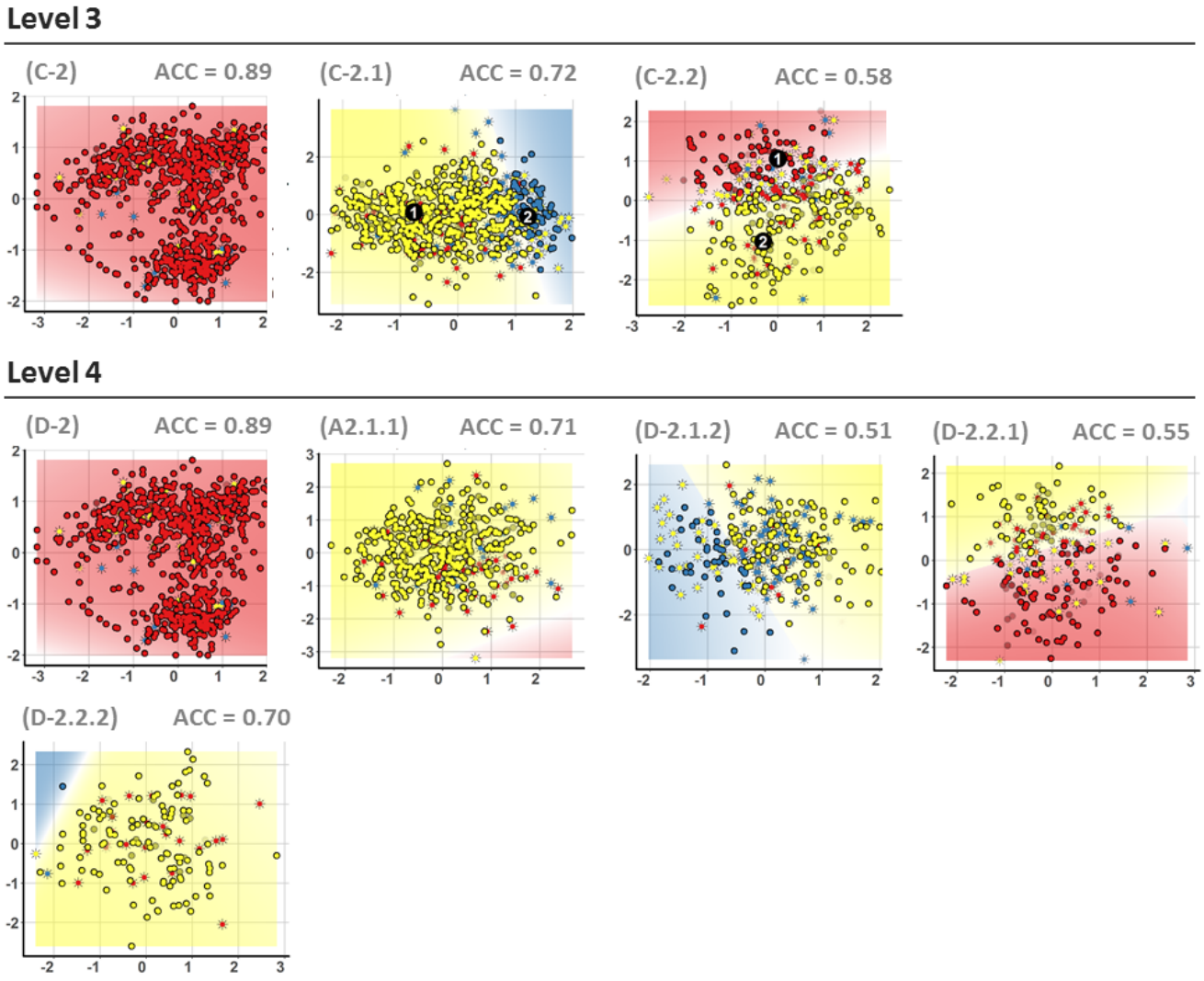

Figure 21: Visualization of experimental dataset of Nestorowa et al. (2016). Dataset Nestorowa consists of 1656 cells and 3 cell types. The number of dimensions is reduced from 239 genes to 2 dimensions with hmPPCA. Here Level 3 and Level 4 are visualized. By initialization of 2 new centers in Level 2 of Figure 18 a third level is created consisting of 3 components. The hierarchy is extended to a fourth level by initialization of 4 user selected centers in Level 3. The letter between parentheses for each plot correspond to the level of the hierarchy. Numbers correspond to the user selected center in preceding levels of the hierarchy (Figure 18). The first number refers to centers in Level 1, the second number refers to centers in Level 2, etc. Cells highlighted with * are cell types misclassified by the classifier.