# STAT_465-0 Project Report
# Single Cell RNA Sequencing Analysis

Kaile Yuan

## 1 Introduction

What attributes constitute an ideal single-cell RNA-seq? The answer might vary, yet the widely accept criteria might be, as taught in one computational biology class at Caltech (Beltrame et al. 2019) (widely used terms are italicized and boldfaced),

- ***Universal*** in terms of cell size, type and state
- ***In situ*** measurements
- No ***minimum input*** of number of cells to be assayed
- Every cell is assayed, i.e. 100% ***capture rate***
- Every transcript in every cell is detected, i.e. 100% ***sensitivity***
- Every transcript is identified by its ***full-length sequence***
- Transcripts are assigned correctly to cells, e.g. no ***doublets***
- Additional ***multimodal*** measurement
- ***Cost*** effective per cell
- ***Easy*** to use
- ***Open source***

As of February 2019, the largest single-cell RNA-seq dataset has been published consists of 690,000 Drop-seq adult mouse brain cells (Saunders et al. 2018). The rapid adoption of single-cell RNA-seq is evident in the growth of records in public sequence database, reported by Ben Langmead, shown in **Figure** 1.

### 1.1 Brief history of scRNA-seq

Single-cell RNA sequencing technology could be traced back to the research on electrophysiological characterization of single live neuron (Eberwine et al. 1992) . People used the whole-cell patch-clamp technique to understand the electrophysiological properties of single cells. One limitation encountered was that it's not possible to obtain electrophysiological recordings from the individual cells prior to in situ hybridizations (ISH). Before ISH, since the amount of mRNA within a single cell is minuscule, estimated to be between 0.1 and 1 pg, therefore PCR is usually employed. However, for PCR there was certain obstacles as well. When isolating RNA from a single cell, there was tendency that RNA interacts nonspecifically with plastic and glass. Several technical hurdles as such were overcome in that research, and they have accurately measured the amount of mRNAs of individual pyramidal neurons of the rat hippocampus.
After the maturity of microarray technology, marked by (Schena et al. 1995), people started to think about how to adapt single-cell RNA-seq to microarray platform. However labor-intensive Sanger sequencing was still the major sequencing technology at that time, which has created an insurmountable barrier for sequencing the whole cell. The next-gen sequencing technology didn't come into play till 2007, with a dramatic drop of sequencing cost, as shown by a well-cited graph made available by NIH. In 2009, scientist reported the adaptation of next-gen sequencing technology, marking the beginning of scRNA-seq technology (Tang et al. 2009).

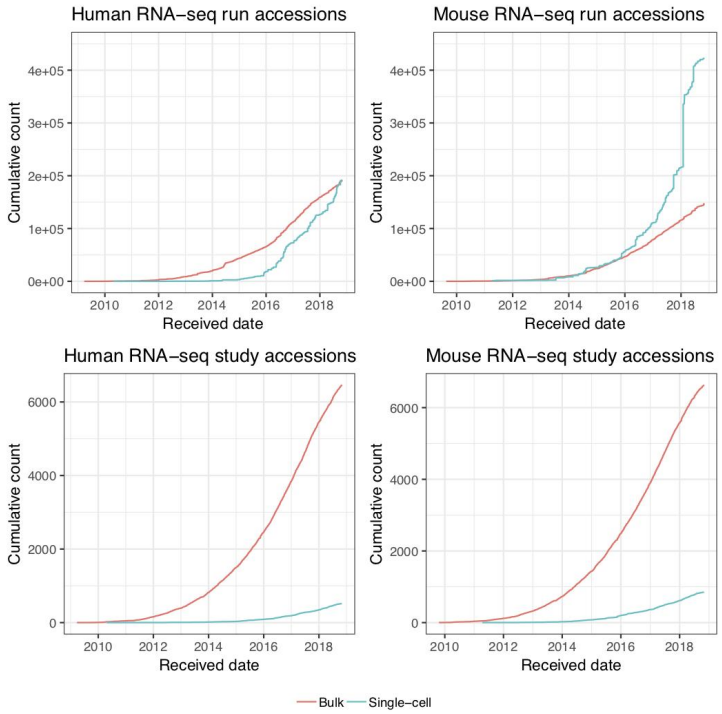


**Figure 1:** SRA RNA-seq growth for human and mouse, distinguishing bulk vs single-cell and run vs study accessions

## 1.2 Applications of scRNA-seq

ScRNA-seq enables the field to surpass the descriptive studies of specific cell states, and makes it possible that people can quantify the transcriptomes of cells of interest. Therefore, the technology is potentially endowed with broad medical applications. For example, in field of tumor studies, the tumor heterogeneity is a common phenomenon that can occur both within and between tumors, (Hwang, Lee, and Bang 2018) has foreseen that the application of scRNA-seq helps illuminate unknown tumor features that cannot be discerned from conventional bulk transcriptomic studies. In phylogenetics, scRNA-seq can be applied to reconstruct clonal and phylogenetic relationships between cells by modeling transcriptional kinetics. In developmental biology, scRNA-seq could be employed to ask the lineage tracing question, the fundamental question of this field. Particularly, the expression of mRNA in a single cell was used to construct lineage phylogeny over many generations, such as (Frieda et al. 2017) has done.

## 1.3 scRNA-seq method comparison

There are a few platforms available for performing scRNA-seq, including the leading commercial platform Smart-seq/C1, one of the most popular full-length methods Smart-seq2 (Picelli et al. 2014), a UMI-method (unique molecular identifier) that uses in-vitro transcription for cDNA amplification from manually isolated cells CEL-seq (Hashimshony et al. 2012), a UMI-method that has a very high throughput Drop-seq, and a UMI-method that allows single-cell isolation by FACS (fluorescence activated cell sorter), SCRB-seq (Soumillon et al. 2014).

### 1.3.1 SMART-seq2

(Picelli et al. 2014) (also for library preparation for SMART-seq2) introduced Smart-seq2 for single cell transcriptome analysis, with improved sensitivity, accuracy and full-length cover-

age based on the previously introduced Smart-seq protocol. SMART is a clever acronym for **S**witching **M**echanism **A**t the 5' end of the **R**NA **T**ranscript. There are certain advantages over the commercially available widely used SMARTer kit, for example. Smart-seq2 was reported with improved quality of library at the cost of 12% of the previous generation. It also allows a high degree of multiplexing: up to 96 samples can be pooled and sequenced on a single lane of an Illumina sequencer.

### 1.3.2 TIVA-seq

TIVA-seq employs a transcriptome *in vivo* analysis(TIVA) tag, which upon photoactivation enables mRNA capture from single cells in the live tissue (Lovatt et al. 2014). It was reported as the first noninvasive approach for capturing mRNA from live single cells in their natural microenvironment.

## 1.4 R packages for analyzing scRNA-seq data

### 1.4.1 Seurat

Seurat is R toolkit developed in Satija lab targeting at the joint analysis of multiple scRNA-seq datasets and downstream comparative analysis (Butler et al. 2018). The integrated computational framework allows for robust and insightful comparison of heterogenous tissues in health and disease, integration of data from diverse technologies, and the comparison of single-cell data from different species. Several features of Seurat includes:

- Unsupervised clustering and discovery of cell types and states
- Spatial reconstruction of single cell data
- Integrated analysis of scRNA-seq across conditions, technologies and species.

Among them, the spatial reconstruction functionality of importance and this report will employ it to gain insight into scRNA-seq data.

### 1.4.2 URD

URD is an R package developed in Schier lab for reconstructing transcriptional trajectories underlying specification or differentiation process in the form of a branching tree, provided scRNA-seq data (Farrell et al. 2018) . It is named after the Norse mythological figure who nurtures the world tree and decides all fates. It has been demonstrated to reconstruct the developmental trajectories during zebrafish embryogenesis. It has been demonstrated further that using Drop-seq data, the gene expression map of Hydra nervous system can be constructed (Siebert et al. 2018).
The typical dimensionality reduction method used in the these two packages includes principal component analysis (PCA), non-negative matrix factorization (NMF) and t-distributed stochastic neighbor embedding (tSNE).

# 2 Specific Aims

With the data provided, combining the R package Saurat and URD, to recapitulate the work of (Farrell et al. 2018). Based on the understanding of these two toolkits, appropriations and modifications of the source code will be made.

# 3 Analysis

(Farrell et al. 2018) has performed the drop-seq analysis of wild-type embryos (38,731 cells, 12 timepoints, 28 samples total, 20-40 embryos per sample), SMART-seq2 analysis of wild-type and MZoep mutant embryos (52 wild-type and 364 MZoep cells from 50% epiboly stage), and 10X single-cell sequencing analysis of WT and MZoep mutant embryos (3,000 WT and 2,200 MZoep cells from 6-somites stage). The following table has summarized the sampled zebrafish development stage with the abbreviation used in the following analysis.

| Period | Stage | Gene | UMIs | Abbreviation |
|---|---|---|---|---|
| blastula | high stage | 1,000-7,500 | 1,500-40,000 | ZFHIGH |
| 2.25-5.25h | oblong stage | 625-7,500 | 1,500-30,000 | ZFOBLONG |
| | dome stage | 800-3,800 | 2,000-20,000 | ZFDOME |
| | 30% epiboly | 625-3,000 | 1,000-17,500 | ZF30 |
| gastrula | 50% epiboly | 600-4,000 | 1,500-25,000 | ZF50 |
| 5.25-10.33h | shield stage | 600-2,500 | 1,000-15,000 | ZFS |
| | 60% epiboly | 600-3,500 | 1,500-22,500 | ZF60 |
| | 75% epiboly | 600-3,200 | 1,400-20,000 | ZF75 |
| | 90% epiboly | 500-3,500 | 1,000-20,000 | ZF90 |
| | bud stage | 500-3,200 | 1,000-17,500 | ZFB |
| segmentation | 3-somite stage | 500-3,000 | 1,000-12,500 | ZF3S |
| 10.33-24h | 6-somite stage | 500-3,000 | 1,000-12,500 | ZF6S |

**Table 1:** Zebrafish Developmental Staging Series Sampled (Farrell et al. 2018)

## 3.1 Drop-seq analysis

UMI (unified molecular identifier) is used to label distinct cDNA sequences in scRNA-seq experimental method. We have to distinguish here in particular that UMIs are used to label molecules whereas barcodes are used to label cells. As one example shown to the right, (Islam et al. 2014) used 5-digit UMI for labeling up to 1024 distinct molecules and 6-digit barcode for labeling different cells. Approximately $10^5 \sim 10^6$ mRNA molecules are present in a typical single mammalian cell, and up to 10,000 different genes may be expressed. However, many genes are expressed from multiple promoters or have promoters with diffuse transcription start sites, so that the number of identical mRNA molecules is expected to be <100 for most genes.

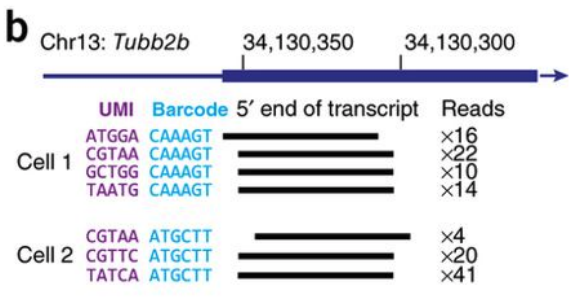


**Figure 2:** UMI alignment and mRNA molecule counting example from (Islam et al. 2014)

"createUDR" method takes matrix comprised of different genes and their UMI counts. The data fed to URD packages are characteristically in the form of "ZFDOME_WT_DS5_AAAATCAAGAGG". Here "ZFDOME" denotes the tissue taken from zebrafish embryos, "WT" stands for cell types, "DS5" marks the specific cell, and the final 12-digit nucleotide sequence serves as the combination of UMI and barcode for mapping different cells and their characteristic molecules. Conceivably, barcode and UMI are both 6-digit respectively. One thing to note here is that the 12-digit deoxynucleotide sequence is hooked to one mRNA sequence, thus the whole sequence is comprised of 5 different necleobases. To gain a sense of the unique identifier such

UMI can encode, a simple calculation we can do here is that by a 8-digit nucleotide sequence, we can encode $4^8 = 2^{16} \approx 150,000$ unique identifier.
"createURD" method also takes another metadata matrix, with rows as cells (should match column name of count.data), and every single cell in the row corresponds to "NUM_READS", "NUM_TRANSCRIPTS", "NUM_GENES", "MT_TRANSCRIPTS", "PERCENT_MT" as additional information.

### 3.2 Find variable genes

The coefficient of variation (CV), defined as the ratio between the standard deviation ($\sigma$) of a variable and its mean ($\mu$), is a natural measure of a gene's extent of variation. However, in the count-based data (UMI based data), ranking genes based on CV solely could lead to declaring those genes with comparatively low mean as variable gene, which is the false positive. By accounting for the mean-CV relationship in droplet dataset, there should be a better way of selecting variable genes (Pandey et al. 2018). The simplest null model is that the transcript counts follow the Poisson distribution, namely,

$$X_g \sim Poisson(\mu_g) \tag{1}$$

where $X_g$ is the UMI counts for gene $g$ in a cell, and $\mu_g$ is the sampling rate equal to the average count of gene $g$ across all cells. Since the variance of Poisson distribution is equal to mean, this predicts a relationship: $CV_g = 1/\sqrt{\mu_g}$. The Poisson model, which is parameter-free, provides a tight lower bound of the CV for lowly expressed genes, i.e., the actual CV values for lowly expressed genes are equal to or higher than the Poisson CV. However, in practice, the high expression values observed that the model significantly underestimated the minimum CV in the data. More specifically, the CV of genes in the data appear to plateau at high mean expression, whereas the Poisson model predicts a square root decrease (**Figure** 3).

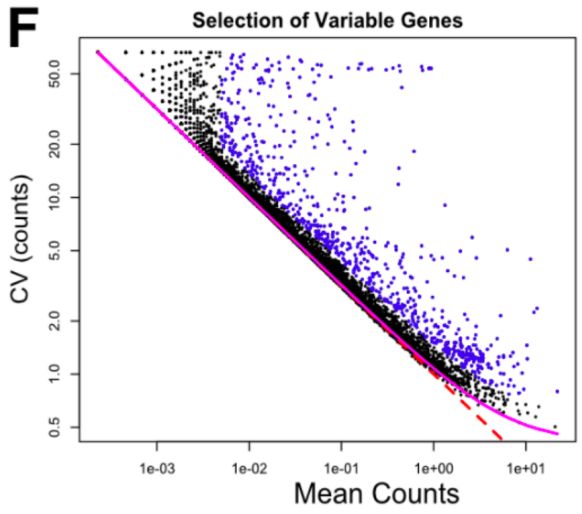


**Figure 3:** Plot of CV of UMI counts across cells versus mean UMI counts. Each black dot corresponds to a single gene, red dotted line is the empirical mean-CV relation predicted by the Poisson model, the magenta curve represents the modified Gamma-Poisson (Nagative Bimodal) model of the expected null CV as a function of mean counts, from (Pandey et al. 2018)

The question here is what accounts for the over-dispersion in the data at high mean expression values compared to the Poisson model. Since Poisson model treats genes individually, i.e., there's no dependence between different genes, the total number of transcript counts per cell ($N_{tot}$) is a sum of independent Possion r.v., which is also a Poisson r.v.. This conclusion is not supported by the dat a Pandey oberved, as the variance of $N_{tot}$ is approximately 389 times its mean in the larval droplet data. This can be caused by a lot of factors, if biological, can be the cell state and cell size, whereas if technical, could be the variations in cell lysis and efficiency of RT, number of captured oligonucleotides, or extent of RNA degradation between droplets. Based on this, Pandey made a simple modification to the Poisson model by positing that the sampling rate of a gene in a given cell depends on its relative library size $\eta$, and hypothesized that,

$$\eta = \frac{N_{tot,i}}{\mathbb{E}(N_{tot,i})} \tag{2}$$

where $N_{tot,i}$ is the total number of molecules in cell $i$ and $\mathbb{E}(N_{tot,i})$ is its expectation across all cells. Note that $\mathbb{E}(\eta) = 1$. They found that a Gamma distribution with mean of 1 provided an excellent fit for the empirical distribution of $\eta$ in all of the droplet datasets (**Figure** 4).

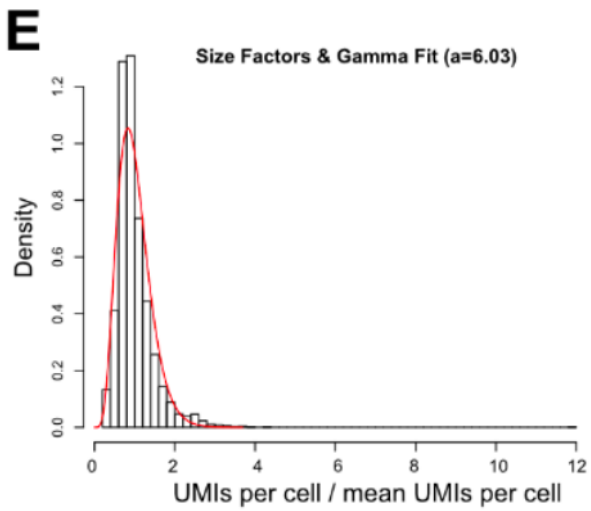


**Figure 4:** The distribution of the relative library size $\eta$ of cells in droplet data modeled as Gamma distribution (red curve). The particular parametrization of $\eta$ ensures that the mean of 1 in its distribution, making it a single parameter fit, from (Pandey et al. 2018)

As a result, every gene follows a negative binomial distribution, $X_g \sim NB(r, p_g)$. Here, $r$ and $p$ represent the canonical parameters of the negative binomial distribution, the number of failures($r$) and the success probability($p$). Using this distribution, they compute the CV-mean relationship as

$$CV_g^2 = \frac{1}{\mu_g} + \frac{1}{\alpha} \tag{3}$$

where $\alpha$ equals $r$ in the distribution, which is a fitting parameter in the model. The solid magenta line in **Figure** 3 shows this relationship. We can see that for lowly expressed genes, where $1/\mu_g \gg 1/\alpha$, the curve reduced to Poisson. For highly expressed genes, $\mu_g \gg \alpha$, the model reduces to $CV_g = 1/\sqrt{\alpha}$, explaining the saturation observed.

This model, most importantly, serves a handy tool for estimating the lower bound of CV across all genes. As (Farrell et al. 2018) did, they ranked the genes based on their distance from the null curve in the log-space, i.e., $log(CV_{observed}/CV_{NB})$. Then the cutoff value 0.3 is estimated, above which genes are considered highly variable. In zebrafish data, $CV_{observed} > 1.35CV_{NB}$ serve as the threshold for concluding "biologically variable".

After identifying those variable genes, data are put into next step, for calculating distance between cells in gene expression space, for calculating the diffusion map, and for building the tree. They are also privileged during differential expression analysis with lower threshold, as they are more likely to be interesting cell-type specific marker. As the genes that encode biological information vary over developmental time, we need to calculate variable genes separately for each stage, and take the union of them.

For all downstream analysis, the digital gene expression matrix was normalized to account for differing library sizes. Each cell's gene expression values were divided by the number of detected transcripts in that cell and multiplied by 10,000 (the median number of UMIs observed per cell across all of the data, rounded up to the nearest order of magnitude). Values were then $\log_2$ transformed.

### 3.3 Correcting batch effect using MNN

(Haghverdi et al. 2018) has published one novel batch effect correcting method named mutual nearest neighbors (MNN). The method firstly uses a cosine normalization and then identifies mutual nearest neighbors across two different batches. Each MNN pair represents cells in different batches that are of the same cell type/state, assuming that batch effects are mostly orthogonal to the biological manifold. MNN correction is implemented by `mnnCorrect` in the scran package.

### 3.4 PCA and tSNE representation projection for dimensionality reduction

URD package utilizes `RunPCA` method of the Seurat package, which largely replies on Lanczos bidiagonalization algorithm (IRLBA) for SVD computing. T-distributed stochastic neighbor embedding method (tSNE) needs the PCA result. URD package `calcTsne` directly uses `Rtsne`, i.e. the Barnes-Hut implementation of tSNE wrapper written in C++. Perplexity, in particular, is a measure for information that is defined as 2 to the power of the Shannon entropy. The perplexity of a fair die with k sides is equal to k. In t-SNE, the perplexity may be viewed as a knob that sets the number of effective nearest neighbors. It is comparable with the number of nearest neighbors k that is employed in many manifold learners.
Applying this to zebrafish data, gene expression of different developmental time is well segmented, yet the visualization here is not that informative since distances between different clusters in tSNE don't mean anything, so do the cluster sizes (since the stochastic algorithm could generate a slightly different visualization every time). However, in a positive sense, this graph provides us with the fundament on which we are going to construct diffusion map and transitional edges later on, for the trajectory inference.

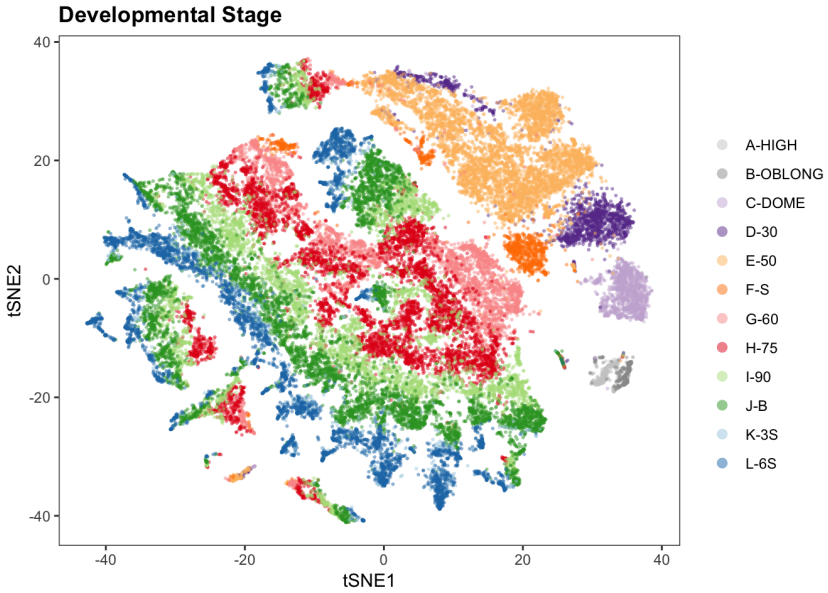


**Figure 5:** tSNE visualization of 38731 cells of differential gene expression across 12 timepoints

### 3.5 Diffusion maps

From the tSNE projection, we have to calculate the transition probabilities between cells. The eigendecomposition of the transition probabilities gives diffusion components, which comprises a diffusion map.
Since the diffusion map is calculated on a k-nearest neighbor graph in gene expression space, cells that are unusually far from their nearest neighbors in a k-nearest neighbor graph often result in poor diffusion maps because many of the highly ranked diffusion components will primarily represent variability of individual outlier cells. Thus, cropping cells based on their distance to their nearest neighbor, and cropping cells that have unusually large distances to an nth nearest neighbor (given the distance to their nearest neighbor) generally produces better, more connected diffusion maps.
Here URD used destiny package method `DiffusionMap` for map generating. When using this, the choice of parameter sigma is kind of tricky, and for this, there's one specific destiny method

named `find_sigmas` for global sigma, the diffusion scale parameter of the Gaussian kernel. A small sigma requires cells to be closer to each other in transcriptional space in order to be connected in the tree, yet an overly small sigma will create disconnections in the map. Here, we calculated several diffusion maps on the zebrafish data with varying sigmas (5, 7, 8, 9 and 13), and generally, we found it's best to choose the smallest sigma possible that doesn't cause many disconnections in the data. From the graphs in the next page (showing 5, 8, 9, 13 due to the page limit), we can see that sigma 5 is too small and has many components that are essentially linear, while sigma 13 is too broad. Therefore sigma=8 here is appropriate for building the tree.

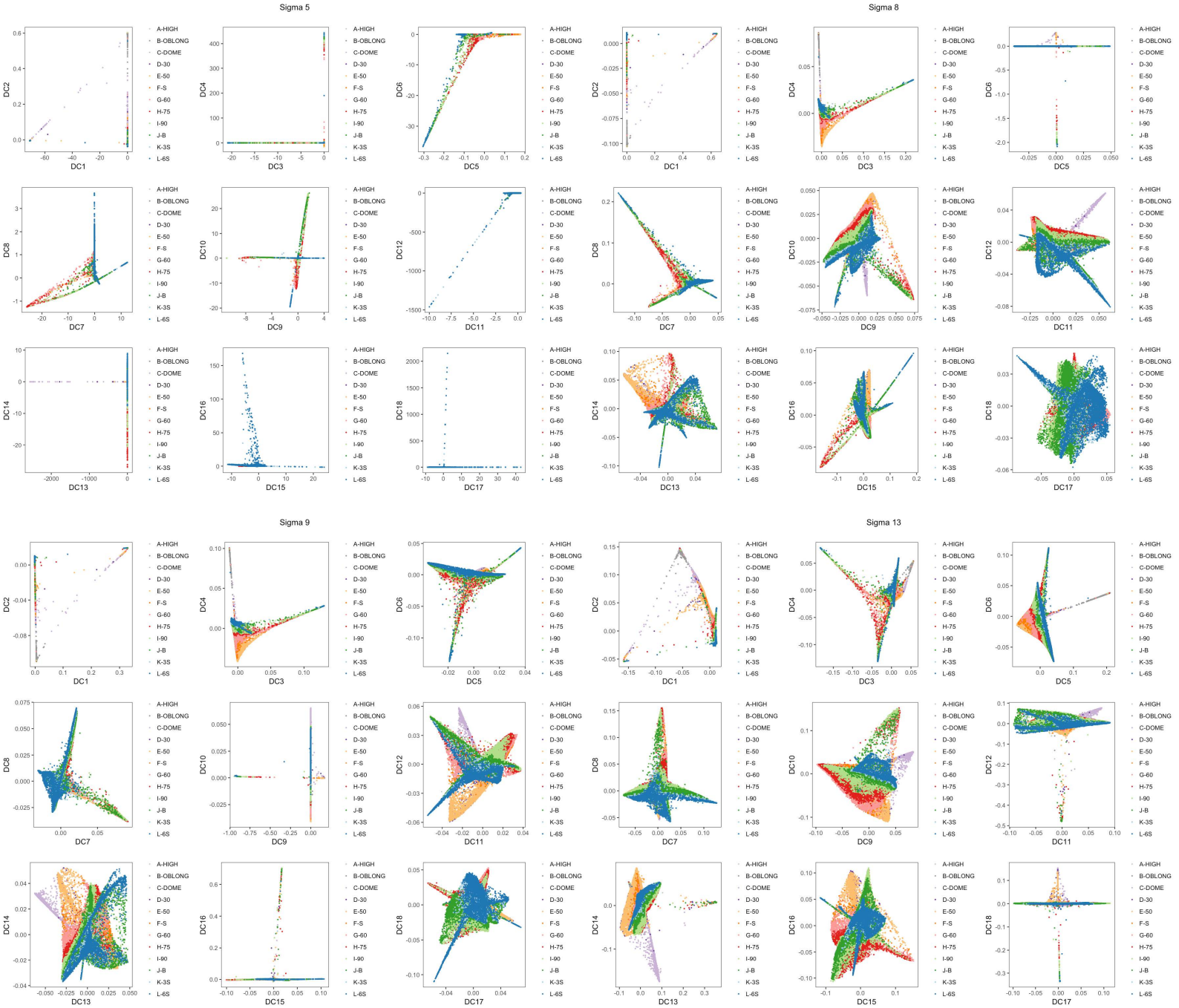


**Figure 6:** Parameter comparison for displaying diffusion components (DC), sigma = 5, 8, 9 and 13

### 3.6 Build tree and trajectory inference

For building the developmental trees, first pseudotime is assigned, and then do the random walk from the early phases reflected in pseudotime to late phases. Pseudotime is defined as the visitation times of the probabilistic breadth-first search of the constructed diffusion map. Since this procedure is not deterministic, the average between different simulations is calculated as the pseudotime of each node in the graph, i.e., each cell in the developmental trajectory. After adequate number of simulations, we see that the change of number of visitation has dropped to a lower level, suggesting almost all the possibilities of visiting the graph in such a manner have been iterated, and this is when it's trustworthy that a good yardstick of assigning time to cells is attained.
The next step is to find the tip in the diffusion map based either on late phases suggested by pseudotime or on prior knowledge of specific developmental stages. Then random walk is performed from the tip to the root in the graph by converting the undirected graph to a directed graph using pseudotime. Thus a developmental trajectory can be generally obtained in this way. For data analysis, due to the calculation capacity limit of my computer(the original random walks were performed on clusters) I used the subset of the raw data, the two specific cell type lineages, i.e. the notochord and prechodal plate(these two cell types are part of the axial mesoderm, so they share a common progenitor type) to show how it's implemented.

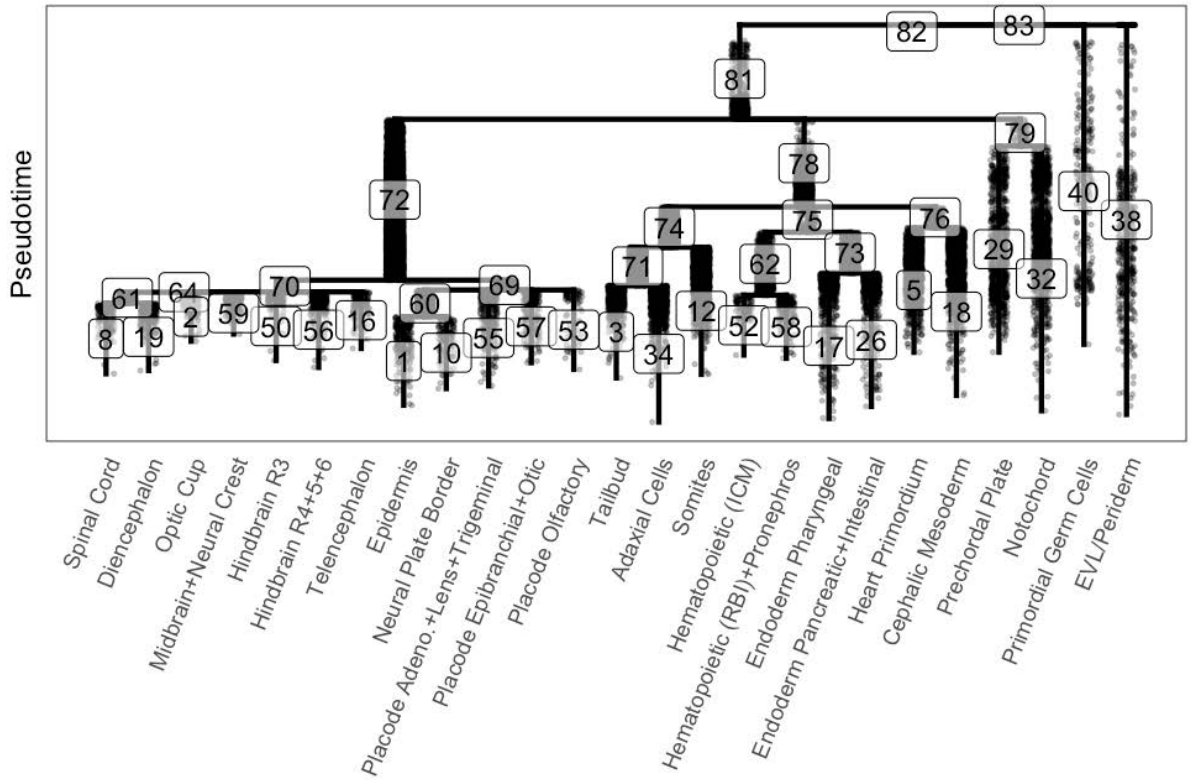


**Figure 7:** Tree structure achieved using random walks on the zebrafish drop-sep dataset

## 4 Summary

Using the zebrafish drop-set dataset, I have analyzed variable genes in the different developmental stages, performed PCA and subsequently obtained a tSNE projection of high dimensional

expression space. Then in order to create a connected graph based on tSNE projection, diffusion map is generated using the package *destiny*. Pseudotime is defined on the connected graph based on a probabilistic breadth-first search of the graph and subsequently it converts the undirected graph to a directed one. Finally random walk is performed from the tip (later phase/large pseudotime) to the root (early phase/small pseudotime) of the graph and thus a putative developmental tree is constructed.

## References


Beltrame, Eduardo da Veiga et al. (Feb. 2019). *Introduction to Single-Cell RNA-Seq Technologies*. DOI: 10.6084/m9.figshare.7704659.v1.

Butler, Andrew et al. (May 2018). Integrating Single-Cell Transcriptomic Data across Different Conditions, Technologies, and Species. en. Nature Biotechnology *36*.5, 411–420. ISSN: 1546-1696. DOI: 10.1038/nbt.4096.

Eberwine, J. et al. (Apr. 1992). Analysis of Gene Expression in Single Live Neurons. en. Proceedings of the National Academy of Sciences *89*.7, 3010–3014. ISSN: 0027-8424, 1091-6490. DOI: 10.1073/pnas.89.7.3010.

Farrell, Jeffrey A. et al. (June 2018). Single-Cell Reconstruction of Developmental Trajectories during Zebrafish Embryogenesis. en. Science *360*.6392, eaar3131. ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.aar3131.

Frieda, Kirsten L. et al. (Jan. 2017). Synthetic Recording and *in Situ* Readout of Lineage Information in Single Cells. en. Nature *541*.7635, 107–111. ISSN: 1476-4687. DOI: 10.1038/nature20777.

Haghverdi, Laleh et al. (May 2018). Batch Effects in Single-Cell RNA-Sequencing Data Are Corrected by Matching Mutual Nearest Neighbors. en. Nature Biotechnology *36*.5, 421–427. ISSN: 1546-1696. DOI: 10.1038/nbt.4091.

Hashimshony, Tamar et al. (Sept. 2012). CEL-Seq: Single-Cell RNA-Seq by Multiplexed Linear Amplification. Cell Reports *2*.3, 666–673. ISSN: 2211-1247. DOI: 10.1016/j.celrep.2012.08.003.

Hwang, Byungjin, Lee, Ji Hyun, and Bang, Duhee (Aug. 2018). Single-Cell RNA Sequencing Technologies and Bioinformatics Pipelines. En. Experimental & Molecular Medicine *50*.8, 96. ISSN: 2092-6413. DOI: 10.1038/s12276-018-0071-8.

Islam, Saiful et al. (Feb. 2014). Quantitative Single-Cell RNA-Seq with Unique Molecular Identifiers. en. Nature Methods *11*.2, 163–166. ISSN: 1548-7105. DOI: 10.1038/nmeth.2772.

Lovatt, Ditte et al. (Feb. 2014). Transcriptome *in Vivo* Analysis (TIVA) of Spatially Defined Single Cells in Live Tissue. en. Nature Methods *11*.2, 190–196. ISSN: 1548-7105. DOI: 10.1038/nmeth.2804.

Pandey, Shristi et al. (Apr. 2018). Comprehensive Identification and Spatial Mapping of Habenular Neuronal Types Using Single-Cell RNA-Seq. Current Biology *28*.7, 1052–1065.e7. ISSN: 0960-9822. DOI: 10.1016/j.cub.2018.02.040.

Picelli, Simone et al. (Jan. 2014). Full-Length RNA-Seq from Single Cells Using Smart-Seq2. en. Nature Protocols *9*.1, 171–181. ISSN: 1750-2799. DOI: 10.1038/nprot.2014.006.

Saunders, Arpiar et al. (Aug. 2018). Molecular Diversity and Specializations among the Cells of the Adult Mouse Brain. English. Cell *174*.4, 1015–1030.e16. ISSN: 0092-8674, 1097-4172. DOI: 10.1016/j.cell.2018.07.028.

Schena, Mark et al. (Oct. 1995). Quantitative Monitoring of Gene Expression Patterns with a Complementary DNA Microarray. en. Science *270*.5235, 467–470. ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.270.5235.467.

Siebert, Stefan et al. (Jan. 2018). Stem Cell Differentiation Trajectories in Hydra Resolved at Single-Cell Resolution. bioRxiv, 460154. DOI: 10.1101/460154.

Soumillon, Magali et al. (Mar. 2014). Characterization of Directed Differentiation by High-Throughput Single-Cell RNA-Seq. en. bioRxiv, 003236. DOI: 10.1101/003236.

Tang, Fuchou et al. (May 2009). mRNA-Seq Whole-Transcriptome Analysis of a Single Cell. en. Nature Methods *6*.5, 377–382. ISSN: 1548-7105. DOI: 10.1038/nmeth.1315.