**Software Project Lab - III**

**CISTRON  
(A Trajectory Based Single Cell Analysis Tool)**

**Course No: SE – 801**

Submitted by

**Ishrat Jahan Emu**

BSSE 0927

Session: 2016–17

Supervised By

**Dr.Sumon Ahmed**

Assistant Professor

Submitted to

**Software Project Lab – III Coordinators**



Institute of Information Technology (IIT)

University of Dhaka

**Date of Submission**: 23rd February 2021

## Letter of Transmittal

February 23,2021

Software Project Lab – III Coordinators

Institute of Information Technology

University of Dhaka.

Dear Sir,

I was assigned to develop an R shiny application named Cistron (A Trajectory Based Single Cell Analysis Tool). I am submitting my report with due respect. I have tried my best for the report.

So, may I therefore, hope that you would and oblige thereby.

Sincerely yours,

\_\_\_\_\_\_\_\_\_\_\_\_\_

Ishrat Jahan Emu

BSSE 0927

Institute of Information Technology

University of Dhaka

## Document Authentication

This project document has been approved by the following persons.

\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Approved by

Dr. Sumon Ahmed

Assistant Professor

Institute of Information Technology

University of Dhaka

Prepared by

Ishrat Jahan Emu

BSSE 0927

Institute of Information Technology

University of Dhaka

## Acknowledgement

By the grace of Almighty Allah, I have completed my technical report on Cistron (A Trajectory Based Single Cell Analysis Tool).

I am grateful to my supervisor Dr. Sumon Ahmed for his direction throughout the working time.

I am also thankful to the Single Cell Research Lead of The University of Manchester. They helped me a lot by sharing their valuable knowledge with me.

## Abstract

In the field of cellular biology, single-cell analysis is the study of genomics, transcriptomics, proteomics, metabolomics and cell–cell interactions at the single cell level. Being a new technology, sc-rna technology has been established as an important tool in biological analysis. As cell-to-cell variation is a very common and natural property for both healthy and diseased tissues, in most cases single cell RNA analysis provides better outcome than bulk RNA sequencing.

Trajectory inference is a computational technique used in single-cell analysis to determine the pattern of a dynamic process experienced by cells.

Trajectory inference methods are used to infer the developmental dynamics of a continuous biological process such as stem cell differentiation and cancer cell development. Although there are a bunch of trajectory methods, users often face difficulties using the same trajectory method for different datasets. Cistron provides a comparative analysis of Different Trajectory Interface.

Although last year , the world has faced the COVID-19 pandemic situation. There is no doubt that Bioinformatics have to do many things to protect mankind.

This document contains the technical report for the Software Project Lab-III which entitled Cistron (A Trajectory Based Single Cell Analysis Tool). This document provides the overview of the of the scenario-based model, class-based model, and data flow model including the methodology for using Augmented Reality. It also contains the description of tools and technologies used in this application and user manual for this application. Using this document as a guide, we are describing the requirements, necessary diagrams, procedures and working sequence of our project.

Here I will discuss how I will identify the requirements, how to analyze them and how to present a recommended solution for the system.

This will help to make the software according to the demand of the stakeholders

# CHAPTER 01: INTRODUCTION

# This document contains the system requirements “Cistron”, a single-cell RNA-sequencing analysis tool. This specification document includes descriptions of the functions and the specifications of the project. In this section, a review of the entire document is provided. The reader would get familiarized with the contents before the further details are described.

Single cell RNA-seq data raises computational challenges in data analysis due to high variability. Bulk RNA-seq technologies have been widely used to study gene expression patterns at population level in the past decade.The advent of single-cell RNA sequencing (scRNA-seq) provides unprecedented opportunities for exploring gene expression profile at the single-cell level. Currently, scRNA-seq has become a favorable choice for studying the key biological questions of cell heterogeneity and the development of early embryos (only include a few number of cells), since bulk RNA-seq mainly reflects the averaged gene expression across thousands of cells.

Trajectory inference is a computational technique used in [single-cell](https://en.wikipedia.org/wiki/Single-cell_transcriptomics) analysis to determine the pattern of a dynamic process experienced by cells.

Trajectory inference methods are used to infer the developmental dynamics of a continuous biological process such as stem cell differentiation and cancer cell development. Although there are a bunch of trajectory methods, users often face difficulties  using the same trajectory method for different datasets. Cistron provides a comparative analysis of Different Trajectory Interface.

Although last year , the world has faced the COVID-19 pandemic situation. There is no doubt that Bioinformatics have to do many things to protect mankind.

# **CHAPTER 02: BACKGROUND STUDIES**

## This part of this document contains necessary terms which be helpful to understand the next Usage Scenario and Methodology of this project.

## **2.1 Single Cell RNA Sequencing**

Single cell sequencing examines the sequence information from individual cells with optimized next-generation sequencing (NGS) technologies, providing a higher resolution of cellular differences and a better understanding of the function of an individual cell in the context of its microenvironment.

### These single-cell analyses will allow researchers to uncover new and potentially unexpected biological discoveries relative to traditional profiling methods that assess bulk populations. Single-cell RNA sequencing (scRNA-seq), for example, can reveal complex and rare cell populations, uncover regulatory relationships between genes, and track the trajectories of distinct cell lineages in development.

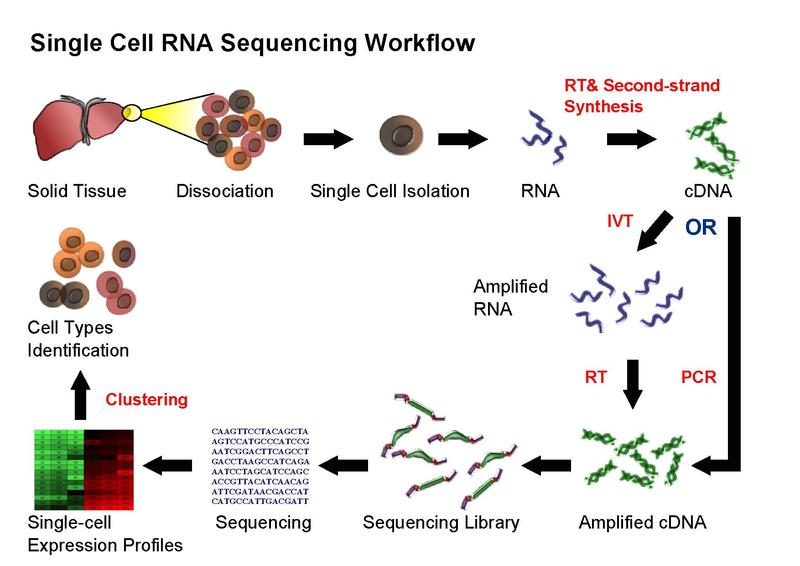


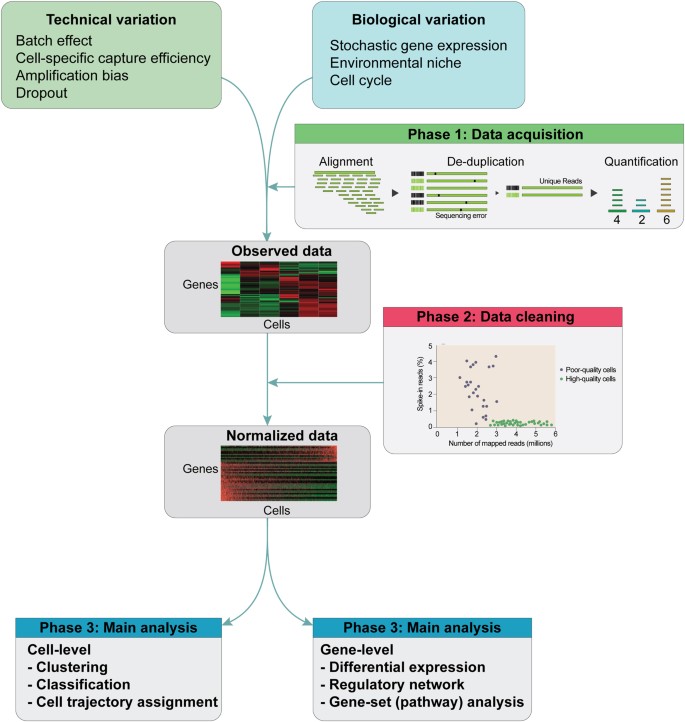
Figure 1: Single cell sequencing

### **2.2** **Computational challenges in scRNA-seq**

Although experimental methods for scRNA-seq are increasingly accessible to many laboratories, computational pipelines for handling raw data files remain limited. Some commercial companies provide software tools, such as 10× Genomics and Fluidigm, but this area remains in its infancy, and gold-standard tools have yet to be developed. In the sections below, we will discuss current bioinformatics tools available for the analysis of scRNA-seq data.

### **2.3** **Pre-processing the data**

Once reads are obtained from well-designed scRNA-seq experiments, quality control (QC) is performed.



scRNA-seq data are inherently noisy with confounding factors, such as technical and biological variables. After sequencing, alignment and de-duplication are performed to quantify an initial gene expression profile matrix. Next, normalization is performed with raw expression data using various statistical methods. Additional QC can be performed when using spike-ins by inspecting the mapping ratio to discard low-quality cells. Finally, the normalized matrix is then subjected to main analysis through clustering of cells to identify subtypes. Cell trajectories can be inferred based on these data and by detecting differentially expressed genes between clusters.

### **2.3** **Trajectory Interface**

Trajectory inference methods are used to infer the developmental dynamics of a continuous biological process such as stem cell differentiation and cancer cell development. Most of the current trajectory inference methods infer cell developmental trajectories based on the transcriptome similarity between cells, using single cell RNA-Sequencing (scRNA-Seq) data. These methods are often restricted to certain trajectory structures like trees or cycles, and the directions of the trajectory can only be partly inferred when the root cell is provided.

Pseudotime analyses of single-cell RNA-seq data have become increasingly common. Typically, a latent trajectory corresponding to a biological process of interest – such as differentiation or cell cycle – is discovered. However, relatively little attention has been paid to modelling the differential expression of genes along such trajectories.

The trajectory plot above shows the trajectory followed by olfactory neurons as the develop in mice. Each point is a cell, where are connected into a minimum spanning tree, the core data structure Monocle uses to find the trajectory, shown in black. Each cell’s pseudotime value is measured as the distance along the trajectory from its position back to the beginning. In order to describe complex differentiation processes in which cells make fate decisions,

# **CHAPTER-03: PROJECT DESCRIPTION**

After discussing the inception phase, I need to focus on the Elicitation phase. So, this chapter specifies the Elicitation phase.

## **3.1 INTRODUCTION**

Requirements Elicitation is a part of requirements engineering that is the practice of gathering requirements from the users, customers and other stakeholders. I have faced many difficulties, like understanding the problems, making questions for the stakeholders, problems of scope and volatility. Though it is not easy to gather requirements within a very short time, I have surpassed these problems in an organized and systematic manner.

## **3.2 ELICITING REQUIREMENTS**

Unlike the beginning, Elicitation uses a requirements format that incorporates problem solving, preparation, negotiations and specification components, in which questions were answered. A group of end-users and developers must cooperate in order to generate the demands.

**3.3 COLLABORATIVE REQUIREMENTS GATHERING**

There are many different approaches to collaborative requirements gathering. Each approach makes use of a slightly different scenario. We followed the subsequent steps to do it:

I. Meetings were conducted with the research fellows of The University of Manchester. They were questioned about their requirements and expectations from the tool.

II. They were asked about the comparative analysis of Trajectory Interface.

III. At last we selected our final requirement list from these meetings.

### **3.4 QUALITY FUNCTION DEPLOYMENT**

Quality Function Deployment (QFD) is a technique that translates the needs of the customer into technical requirements for software. It concentrates on maximizing customer satisfaction from the software engineering process. So,I have followed this methodology to identify the requirements for the project. The requirements, which are given below, are identified successfully by the QFD.

#### 

#### **3.4.1 NORMAL REQUIREMENTS**

Normal requirements are generally the objectives and goals that are stated for a product or system during meetings with the stakeholders. The presence of these requirements fulfills stakeholders’ satisfaction. The normal requirements of my project-

* Comparison of the different trajectory inference methods
* Gene Expression
* An easy way to access the data without any bioinformatic expertise.

#### **3.4.2 EXPECTED REQUIREMENTS**

The requirements that are implicit to the system might not be brought up during the meeting because of their fundamental nature. Despite being not explicitly mentioned their presence must be ensured. Otherwise, the product will leave customers dissatisfied. These requirements are called expected requirements and these are stated below.

* Sample and cluster overview panels.
* Tables of most expressed genes and marker genes for samples and clusters.
* Tables of enriched pathways for samples and clusters.
* Different Methods for Trajectory Interface

#### **EXCITING REQUIREMENTS**

* Interactive plot and all plots can be exported to PNG.
* Standalone Application

### **3.5 Usage Scenario**

Single cell technologies are becoming increasingly important tools in biological analysis. Complementing average measurements on bulk populations of cells, single-cell measurements provide a finer-grained picture of complex biology and unmask heterogeneity that is present in tissues.

Cistron , a trajectory based single cell analysis tool is a standalone application. This tool will take a normalised biological dataset as input. It will show different trajectory interfaces with some common features of single cell technology such as Gene Expression, Enriched Pathway etc. As trajectory interface methods differ for different datasets and users are able to show the different methods result analysis, users can decide the best methods for the particular dataset.

Being a standalone application ,all operating system users can use Cistron without any additional process. Users can export the plot into JPG/PNG format for their further use.

# **CHAPTER-04: SCENARIO BASED MODELING**

For developing our software, we are giving the highest priority to user satisfaction. To identify the requirements to establish meaningful analysis and design model we determine how users want to interact with the system. Thus, our requirements modeling begins with scenario generation in the form of use cases, activity diagrams.

4.1 Use Case

Use case diagrams are usually referred to as behavior diagrams used to describe a set of actions that some system or sub-systems can perform in collaboration with one or more external users of the system.

The first step in writing a Use Case is to define that set of “actors” that will be involved in the story. Actors are the different people that use the system or product within the context of the function and behavior that is to be described. Actors represent the roles that people play as the system operators. Every user has one or more goals when using the system.

4.1.1 Primary Actor

Primary actors interact directly to achieve required system function and derive the intended benefit from the system. They work directly and frequently with the software. In our system both users the system both are primary actor.

4.2 Activity diagram

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In this chapter we did try to provide each use case and its corresponding activity diagram together.

**4.3 Use Case and Activity Diagram**

**4.3.1 Level 0 Use Case Diagram of Cistron**

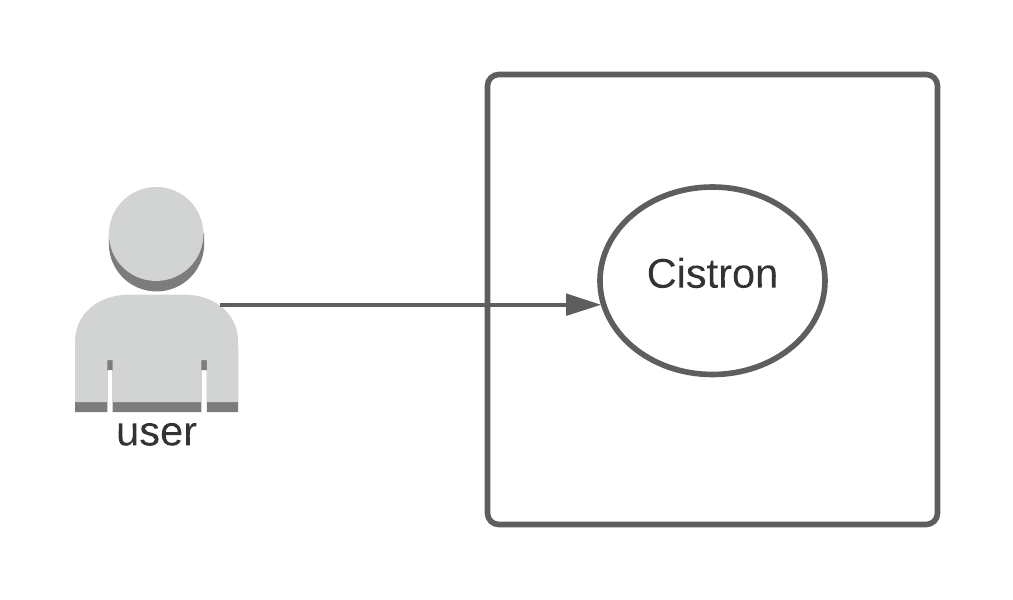


Figure 1: Level 0 use case diagram of Cistron

Table 6: Information about level 0 use case diagram

|  |  |
| --- | --- |
| **Name:** | Cistron |
| **ID:** | L-0 |
| **Primary Actor:** | User |
| **Secondary Actor:** | None |

#### Description of Level 0 Use Case Diagram

After analyzing usage scenario, we found that user interact with our system as primary actor.

### Level 1 Use Case Diagram of Cistron

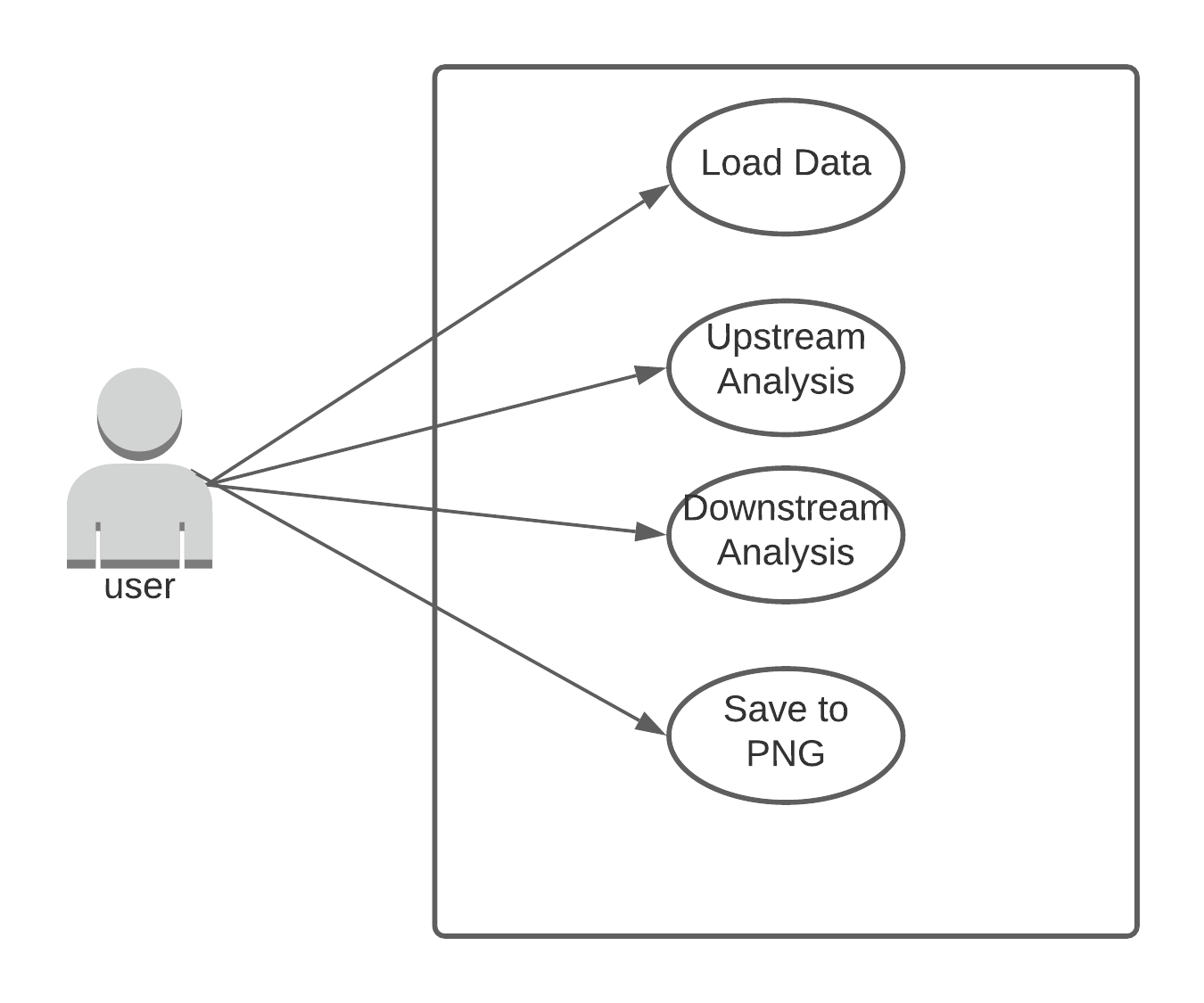


Figure 2: Level 1 use case diagram of Cistron

Table: Information about level 1 use case diagram

|  |  |
| --- | --- |
| **Name:** | Cistron |
| **ID:** | L-1 |
| **Primary Actor:** | User |
| **Secondary Actor:** | None |

# **CHAPTER-05: CLASS BASED MODELING**

In this chapter, our designed class-based model represents the objects that our “Cistron” will manipulate, the operation that will applied to the objects, relationships between and the collaboration that occur between the classes that are defined.

## **5.1 Final classes**

The final classes are identified from the scenario of this project. Those are:

1. UI

2. Server

The class cards of these classes are shown in tables below:

|  |  |
| --- | --- |
| **UI** | |
| Attributes | Methods |
| Dashboard | dashboardSidebar()  dashboardBody()  FluidRow()  plotoutput() |

*Table 3: Class card of UI class*

|  |  |
| --- | --- |
| **SERVER** | |
| Attributes | Methods |
| Data | Renderplot()  Renderplotly()  Ggplot() |

*Table 4: Class card of Server class*

### Class Diagram of Cistron

Figure 3: Class diagram of Cistron

# CHAPTER-06: ARCHETYPE

# CHAPTER-07: PRELIMINARY TEST PLAN

# CHAPTER-08: METHODOLOGY

# **Slingshot**

Slingshot is a tool for the identification of developmental trajectories in single-cell RNA-seq (scRNA-seq) data. The Slingshot algorithm can use prior knowledge via supervised graph construction.

Provides functions for inferring continuous, branching lineage structures in low-dimensional data. Slingshot was designed to model developmental trajectories in single-cell RNA sequencing data and serve as a component in an analysis pipeline after dimensionality reduction and clustering. It is flexible enough to handle arbitrarily many branching events and allows for the incorporation of prior knowledge through supervised graph construction.

TSCAN:

When analyzing single-cell RNA-seq data, constructing a pseudo-temporal path to order cells based on the gradual transition of their transcriptomes is a useful way to study gene expression dynamics in a heterogeneous cell population. Currently, a limited number of computational tools are available for this task, and quantitative methods for comparing different tools are lacking. Tools for Single Cell Analysis (TSCAN) is a software tool developed to better support in silico pseudo-Time reconstruction in Single-Cell RNA-seq ANalysis. TSCAN uses a cluster-based minimum spanning tree (MST) approach to order cells. Cells are first grouped into clusters and an MST is then constructed to connect cluster centers. Pseudo-time is obtained by projecting each cell onto the tree, and the ordered sequence of cells can be used to study dynamic changes of gene expression along the pseudo-time. Clustering cells before MST construction reduces the complexity of the tree space. This often leads to improved cell ordering. It also allows users to conveniently adjust the ordering based on prior knowledge.

Monocle3:

Monocle introduced the strategy of using RNA-Seq for single-cell trajectory analysis. Rather than purifying cells into discrete states experimentally, Monocle uses an algorithm to learn the sequence of gene expression changes each cell must go through as part of a dynamic biological process. Once it has learned the overall "trajectory" of gene expression changes, Monocle can place each cell at its proper position in the trajectory. You can then use Monocle's differential analysis toolkit to find genes regulated over the course of the trajectory, as described in the section Finding genes that change as a function of pseudotime . If there are multiple outcomes for the process, Monocle will reconstruct a "branched" trajectory. These branches correspond to cellular "decisions", and Monocle provides powerful tools for identifying the genes affected by them and involved in making them. You can see how to analyze branches in the section Analyzing branches in single-cell trajectories .

Partition-based Graph Abstraction (PAGA)

• Gives graph-like map of data manifold, based on estimating

connectivity of manifold partitions (e.g. cell clusters)

• Preserves global topology of data, allowing analysis at different

resolutions

• Unifies both the clustering and continuous change approaches

PAGA

How to run

• PAGA is python based

• Save Seurat object as loom and import into anndata (doesn’t always

work smoothly)

• Can run from scratch using the scanpy manual

• Can add matrix, UMAP coordinates and meta data separately

References

1. Saelens, Wouter, et al. "A comparison of single-cell trajectory inference methods." Nature

biotechnology 37.5 (2019): 547.

2. Wolf, F. Alexander, et al. "PAGA: graph abstraction reconciles clustering with trajectory inference

through a topology preserving map of single cells." Genome biology 20.1 (2019): 59.

3. Street, Kelly, et al. "Slingshot: cell lineage and pseudotime inference for single-cell

transcriptomics." BMC genomics 19.1 (2018): 477.

4. Trapnell, Cole, Davide Cacchiarelli, and Xiaojie Qiu. "Monocle: Cell counting, differential

expression, and trajectory analysis for single-cell RNA-Seq experiments." (2019): 10.

5. La Manno, Gioele, et al. "RNA velocity of single cells." Nature 560.7719 (2018): 494.