MSc Bioinformatics

Generating an R Shiny Front End to Visualise Pseudotime Analysis

Supervisor: Dr. Thomas Otto

Carol Clark, 2623287c

A report submitted in partial fulfilment of the requirements for the MSc Bioinformatics Degree at The University of Glasgow October 2021

Links to the Applications

Upload Application

The Upload application is available on GitHub at: https://github.com/2623287c/Upload-Application

Malaria Application

The Malaria application is available on todata1.mvls.gla.ac.uk at (this requires connection to the VPN): (please note for this application heatmaps may take 5-10 minutes to load)

The Malaria application is available on GitHub at: https://github.com/2623287c/Malaria-Application

T Cell Application

The T cell application is available on todata1.mvls.gla.ac.uk at (this requires connection to the VPN): /data/2623287c/Project1/newtcell

The T cell application is available on GitHub at: https://github.com/2623287c/TCell-Application

Each GitHub contains the necessary code and files to run the application as well as supporting documents explaining the code. The steps to test the Upload application are provided on GitHub (it should take 1.5-6 hours per resolution)

Link to the existing applications that were adapted for this project:
http://cellatlas.mvls.gla.ac.uk/

Acknowledgements

I would like to thank my supervisor Dr. Thomas Otto for his support and guidance. I would also like to thank William Haese-Hill and the rest of the Otto lab group for their help and feedback.

Contents

Abbreviations:

RNA sequencing (**RNA-Seq**)

Uniform Manifold Approximation and Projection (**UMAP**)

Potential of Heat-diffusion for Affinity-based Trajectory Embedding (**PHATE**)

Reversed graph embedding (**RGE**)

minimal spanning tree (**MST**)

false discovery rate (**FDR**)

Single cell RNA sequencing (**scRNA-Seq**)

unique molecular identified (**UMI**)

Summary

Background: Single-cell analysis allows users to investigate the gene expression of individual cells. Pseudotime analysis provides further depth to this. It involves the arrangement of cells along time points known as pseudotimes, based on expression patterns. From this, a path between the time points can be inferred. The data gained from pseudotime analysis provides insight into different biological processes such as the cell cycle, immune response, and parasite life cycle.

Problem: There are many different pseudotime tools, each suited to different trajectories and, therefore, different datasets. Furthermore, analysis with these tools often relies on programming knowledge, making it difficult to navigate for many users who require single-cell analysis but lack experience coding. Some applications offer a user-friendly interface alternative to these inaccessible tools; however, they are limited and often only one tool is offered. Investigation of these applications has highlighted a need for an application that: does not require extensive programming knowledge, provides multiple tools for pseudotime analysis, is customisable, and has an option for users to upload data.

Aims: There already exists an atlas produced at the University of Glasgow that allows users to investigate single-cell analysis results and change the resolution to view the impact. The existing atlas, available: http://cellatlas.mvls.gla.ac.uk/, was adapted to include the additional requirements of uploading data and pseudotime analysis. Coded in R using the Shiny framework, a package available in R, the applications are made up of tabs that the user can navigate to provide a user interface to previously inaccessible pseudotime tools.

Product: Based on the concerns over the suitability of tools for different datasets, four different tools were offered (Slingshot, tradeSeq (using trajectory inference outputs of Slingshot), Monocle 2 and Monocle 3). In the application, each tool offers a plot illustrating the inferred trajectories and a heatmap of the genes identified as significant to pseudotime by the tool. From these plots, the user can interpret the results and identify the tool best suited for their analysis

Three applications were produced; T cell, malaria, and an application where users can upload data. The Upload application was evaluated using the T cell application dataset. Furthermore, an informal questionnaire was answered by five users of the Malaria application to provide feedback on the ease of use, customisability, and speed.

Evaluation/Discussion: Evaluation of all the applications highlighted the speed of the application as an issue. Future work would aim to address this. Another possible next step would aim to improve the robustness of the Upload application. The applications did, however, provide good biological insight with four different tools available so that the user could find results that best reflect data and account for the limitations of other tools.

1. Introduction

Single-Cell

Single-cell analysis offers a high resolution to better understand transcription at the level of individual cells[1,2]. It is especially beneficial in complex, heterogenous cells where the previously employed bulk microarrays and RNA sequencing (RNA-Seq) are inadequate[1,3]. A commonly used R package in single-cell analysis is Seurat[4].

Single-cell research has been beneficial in many fields of research. For example, a study by Hentzschel, Gibbins([5]) identified important factors in *Plasmodium* parasite adaptation through the investigation of host-parasite interactions and recognised the importance of host cell maturation state in this adaptation. This study aided the understanding of these causative agents of malaria. Furthermore, a single-cell study in γδ T cells identified novel mechanisms of T cell regulation, a possible future route for immunotherapy[6].

Pseudotime

In addition to single-cell analysis, pseudotime analysis has become more prominent, allowing the study of cells over time[1]. Since its initial description by Trapnell, Cacchiarelli([7]), the field of pseudotime analysis has expanded with over 70 different tools recorded in 2019 and more curated since[8]. Pseudotime inference first involves the arrangement of single-cell data along time points known as pseudotimes[1]. Arrangement of these cells is based on transcriptional changes with the assumption that these gene expression changes are related to the pseudotime[1]. The path that the cells follow along the time points is referred to as the trajectory or lineage.

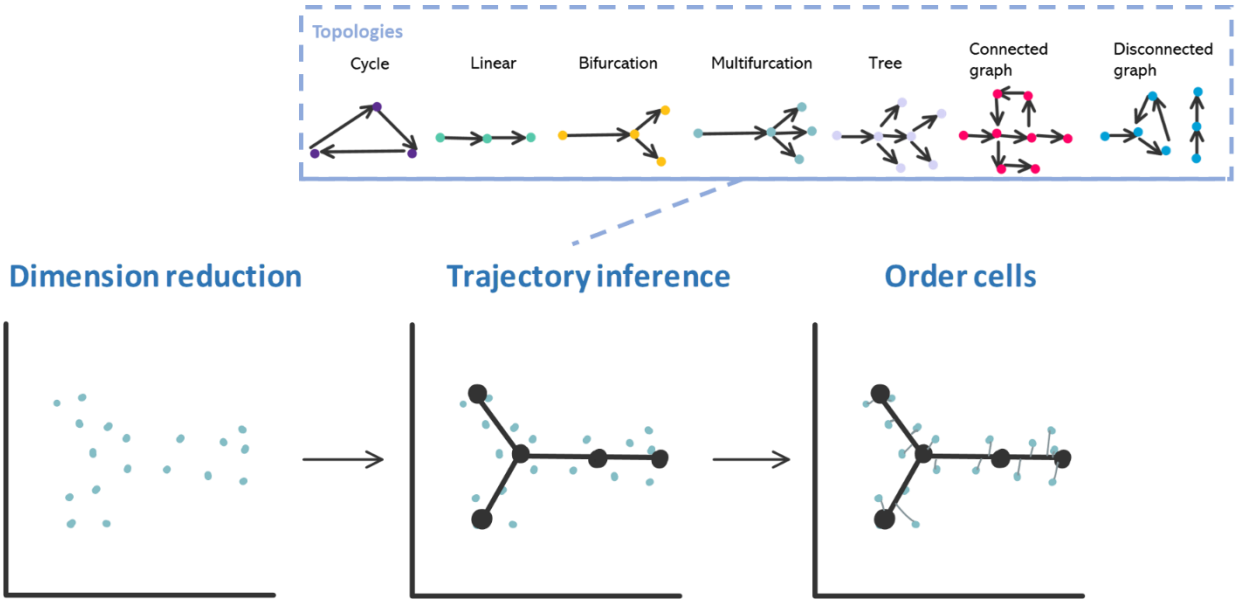

*Figure 1: **Pseudotime Stages**. Overview of the pseudotime analysis stages. Cells represented by blue dots, black lines indicating trajectories, black dots showing time points.*

The steps of pseudotime analysis are summarised in Figure 1. First, the dimensionality is reduced to minimise the complexity of the data[8]. From this reduced dimension, trajectory inference can take place. Here the path between the points is inferred based on gene expression patterns. Finally, the cells are ordered and assigned to lineages. For many tools, the pseudotime is calculated as a measurement of the distance from the cell to the original point (i.e., root node), along the trajectory[2]. Furthermore, differential expression, offered by some tools, can help pinpoint the genes that change at different time points.

The trajectories inferred are not always linear and may instead include branches, with branching points indicating a change in expression, i.e., a cell fate decision (Figure 1). Furthermore, trajectories may even be cyclical or disconnected graphs[9]. This is called the topology and is an area of interest in pseudotime analysis[8]. The varying range of trajectories poses a big issue in many pseudotime tools as different tools are beneficial for the inference of different trajectories[8].

The data provided by pseudotime analysis adds extra insight to the progression of cells in time when this data is not initially available. It can be used to better understand biological processes, such as the cell cycle, parasite life cycle and cell differentiation[8]. It has a range of applications in many different fields of study such as cancer, immunotherapy, and

Parasitology. Previous research using pseudotime analysis has provided a single-cell atlas outlining the maternal-fetal interface to better understand the immune response[10].

Pseudotime Tools

Many pseudotime inference tools lack proper analysis of differential expression with relation to lineages. Tools also differ in accuracy due to different methods used to infer these lineages. Another issue, indicated by Saelens, Cannoodt([8]), is the question of their stability and scalability, i.e., tools are preferred that can produce stable results with similar datasets and tools that can handle increased numbers of cells, respectively. For example, problems with scalability can result in increased running times when faced with high cell numbers.

Many tools have been produced which aim to infer pseudotimes of single-cell data, each with different benefits and detriments. As will be discussed more, four prevailing tools are Slingshot, tradeSeq, Monocle 2 and Monocle 3 (Table 1). The selection of these tools will be discussed more in section 3.1 Pseudotime Tool Selection.

Method	Dimensionality reduction	Cluster-based	Graph	Pseudotime Calculation	Branching	Supervision
Slingshot	Any	Yes	MST on clusters	Simultaneous principal curves, orthogonal projection	Yes	Start cluster, end clusters
Monocle 2	Reversed graph embedding (DDRTree)	No – can add cluster information	Principal graph on cells	Distance to root	Yes	Root state
Monocle 3	t-SNE or UMAP	Supergroups of similar cells inform trajectory inference	Principal graph on cells	Distance to root	Yes	Root node
tradeSeq	N/A	N/A	N/A	From previous tool (i.e., Slingshot)	N/A	Number of knots

Table 1: ***Overview of the Four Pseudotime Tools.***

The four tools used in the application, Slingshot, tradeSeq, Monocle 2 and Monocle 3, are further described here. N/A is used to indicate the features of tradeSeq that are dependent on the upstream trajectory inference tool. Table adapted from Street, Risso([1]) to also include Monocle 3 and tradeSeq

The different tools have different dimension reduction methods. This can impact the results and the extent to which different methods can be compared. Slingshot (and therefore tradeSeq, used downstream of Slingshot) provide Uniform Manifold Approximation and Projection (UMAP) and Potential of Heat-diffusion for Affinity-based Trajectory Embedding (PHATE) reduction. PHATE is judged to provide better biological insight but is not currently widely available. UMAP is available for Monocle 3 while Monocle 2 instead uses the reversed graph embedding (RGE) algorithm DDRTree[11].

Slingshot

Slingshot uses two steps for pseudotime analysis. First, the lineages are identified, beginning from a root cluster, and ending at different clusters for each lineage. This utilises minimal spanning tree (MST) which connects edges with minimal distance. For Slingshot, this MST is applied to clusters instead of individual cells, making this a cluster-based approach. Next, Slingshot uses these lineages and detects the associated pseudotimes. From a starting cluster, the expression is tracked to the end state[1]. Principal curves are extended allowing for various branching lineages[1]. It requires no supervision; however, the user may specify start or end clusters.

Monocle 2

The reversed graph embedding (RGE) algorithm is utilised in Monocle 2 to learn a principal graph[11]. Pseudotime is calculated based on the distance from a root state (often selected by the user) to the cell; Monocle focuses on these so-called functional states that cells progress through, characterised by differences in expression. This means Monocle 2 does require some supervision by the user. Unlike Slingshot, Monocle 2 is not cluster-based, however, information about clusters can be added to the data after ordering of cells.

Research has highlighted limitations in Monocle 2[2,8]. For example, the tool assumes there is only one trajectory present in the cells and cannot identify more than one starting point.

Monocle 3

Monocle 2 has been deprecated in place of Monocle 3[12]. Like Monocle 2, Monocle 3 uses RGE to learn the principal graph. It differs from Monocle 2, however, as it allows the user to select more than one starting node and therefore, can have more than one trajectory; Pseudotime is

calculated as the distance from the cell to the nearest starting node. It groups similar cells into "supergroups" to help identify the trajectories of individual cells; cells of a supergroup are associated with different lineages[12].

tradeSeq

tradeSeq can be used downstream of other trajectory inference tools, such as Slingshot or Monocle[2]. It focuses on the analysis of the differential expression and its association with pseudotime. This allows the characterisation of genes that change both within and between lineages. It uses a negative binomial generalized additive model (NB-GAM) to analyse these relationships. tradeSeq does require user input of the number of knots to be used by the model.

tradeSeq, like Monocle 3, is a recent tool with limited in-depth analysis available to identify the issues with the tools. However, it has been noted that tradeSeq does not provide an accurate false discovery rate (FDR) due to uncalibrated P-values which are only suitable for the internal ranking of the genes[13]. tradeSeq is also markedly time-consuming when confronted with over 1000 cells as noted by Van den Berge, Roux de Bézieux([2])

2. Analysis and Aims

2.1 Problem Statement

Single-cell and pseudotime analysis provide many benefits in the study of gene expression. However, they are often inaccessible to lay users; single-cell and pseudotime analysis in the programming language R predominantly relies on the user's ability to program and employ several packages. There do exist some applications and servers more suitable for users without in-depth programming knowledge. The applications aim to provide a user-friendly interface for single-cell analysis; however, they often have limited analysis (as will be discussed more in Other Applications).

One such application has previously been created at the University of Glasgow, accessed at http://cellatlas.mvls.gla.ac.uk/. This single-cell atlas displays single-cell analysis with the ability to change parameters (namely, the resolution). The user can therefore explore how a higher resolution can result in more clusters and how this impacts the clustering, marker genes and differential expression results. It has already been adapted with applications available for different datasets outlining the results of previous single-cell research.

These existing applications by the University of Glasgow, however, do not allow pseudotime analysis. Other applications also often lack this analysis. Furthermore, even when pseudotime analysis is present, it is often limited to one tool and not more than two.

Moreover, this atlas and many other existing applications do not have an option for the user to upload data to analyse it.

To summarise, no application exists that fulfils the following criteria:

1. Is run as a server (not as separate packages)
2. Users can upload their own Seurat data for analysis
3. Users can change parameters and investigate their effect on the analysis
4. Pseudotime analysis of the data (with multiple tools)

2.2 Other Applications

There are many applications currently available that aim to provide more accessible single-cell analysis. However, an existing application that satisfied the criteria outlined above was not found.

Shiny is an R package framework for creating web applications with a user interface and server, commonly used to produce single-cell applications[14]. Table 2 shows the Shiny applications currently available, including the existing Cell Atlas (highlighted in blue), and how they address these criteria.

App	Own Seurat Data	Pseudotime	Parameters (resolution)	Server	Description
Cell Atlas UofG	X	X	✓	✓	Existing cell atlas http://cellatlas.mvls.gla.ac.uk
Cytoscope	✓ Seurat object	X	X	X (locally run application)	Shiny app to visualize single cell data (https://github.com/sjessa/cytoscope, accessed: 01/10/21)
scClustViz	X	X	✓	✓	An interactive R Shiny tool for visualizing scRNA-Seq clustering results from common analysis pipelines[15]
SCHNAPPs	counts matrix or SingleCellExperiment object	X	✓	X	Shiny app for the exploration and analysis of scRNA-Seq data[16]
Kee single cell RNAseq	X	✓ monocle	X	✓	Results from Kee, Volakakis([17]) scRNA-Seq
shinyCortex	X	~ pseudotime vs expression/ heatmap[1]	✓	✓	Resource that brings together data from recent scRNA-Seq[18]
Granatum	expression matrix and optional meta-data table	✓ monocle 2	✓[2]	✓	Granatum is a graphical single-cell RNA-Seq (scRNA-Seq) analysis pipeline for genomics scientists[3].
TSCAN	expression matrix	✓ Monocle 1 or TSCAN	✓[2]	✓	Tool for differential analysis of single-cell RNA-Seq data. The TSCAN algorithm uses a cluster-based minimum spanning tree (MST) method for the pseudo-temporal ordering of cells[19].
SCRAT	bam files	X	✓[2]	✓	Single-Cell Regulome Analysis Toolbox with a graphical user interface, for studying cell heterogeneity using single-cell regulome data[20].

Table 2: ***Overview of Existing Applications and the Objectives of this Project***

The existing application from the university of Glasgow is outlined in blue. Descriptions are taken from the publications/applications for each. scRNA-Seq = Single cell RNA sequencing. Cytoscope has no associated publication so the link to the GitHub is provided here.

[1]does not specify the pseudotime tool used. It only allows user to view a pseudotime versus expression plot and a heatmap

[2]the user can change the number of clusters (impacted by the resolution) but not specifically the resolution.

The existing cell atlas allows the user to analyse single-cell analysis results with the ability to change the resolution and cluster names. However, it does not include pseudotime analysis nor does it allow the user to upload their own Seurat object. Therefore, this existing tool was adapted to better suit the problems outlined.

Some of the other applications satisfied some of the criteria, such as Granatum[3] which allowed the user to upload data, however, it was not a Seurat object[3]. Furthermore, it only used Monocle 2 for pseudotime analysis. Finally, although it allowed the alteration of some parameters, the number of clusters once chosen were fixed and could not be compared amongst the results produced to find the best resolution, as is offered here.

Similar to the existing application, Kee single cell RNA-Seq provides an interface for users to better explore the results outlined in a previous publication[17]. It has limited customisation (users can only choose what to colour plots by) and has no option to upload other data sets. It does, however, have pseudotime analysis, offering Monocle 2.

Overall, this represents a need for an application that can allow a user to upload data for a Seurat pipeline which includes pseudotime analysis with user input to change variables and view their effect on the data. Moreover, there is a requirement for a pseudotime analysis application that allows a choice of pseudotime tools and not just Monocle; when pseudotime is offered by these applications, it is limited to two or fewer tools, often Monocle.

2.3 Objectives

This project adapted the existing cell atlas' to also provide pseudotime analysis. First, tools were identified for pseudotime analysis. These were then implemented in the Shiny application.

A further objective concerns the ability of users to upload data. Single-cell and pseudotime analysis were run on the data with the results visible in the application.

A secondary consideration was to ensure the application, where possible, was user friendly; these pseudotime tools are all available separately, but their amalgamation allows users with limited previous programming knowledge or no specific knowledge in R to use the application.

2.4 Approach to be Taken and Work Plan

The existing University of Glasgow atlas will be adapted. Shiny, used in the previous atlas, was used here. It was produced on R version 3.6.0. The atlas was run on the server allowing users to view the single-cell analysis and the additional pseudotime analysis. Where the user wants to upload data, a different application (Upload application) is available, which can be run locally on the user's machine. The files produced will be saved to the user's specified folder.

The existing atlas is available for different datasets and two of these were used here: T cell and Malaria. The malaria data was obtained from Hentzschel, Gibbins([5]). The study looked at parasite invasion and sexual differentiation in *Plasmodium*. The T cell data, from McIntyre, Monin([6]), investigated γδ T cell regulation. The Upload application was evaluated using the dataset used in the T cell application. An informal questionnaire was used to further evaluate the Malaria application. It was decided that it would be best to offer three different applications for each dataset (T cell, Malaria, and the users' data) as it would be too memory intensive to combine applications. Table 3 describes the three applications produced, each with the same framework of tabs and plots produced.

Name	**Pre-processed?**	**Dataset**
T cell application	YES	McIntyre, Monin([6])
Malaria application	YES	Hentzschel, Gibbins([5])
Upload application	NO	Any – evaluation here used data used in the T cell application[6]

Table 3: **Summary of the Three Applications Produced**

The T cell, Malaria and Upload applications produced for this project, adapted from the existing applications from the University of Glasgow.

3. Product

3.1 Pseudotime Tool Selection

First, the pseudotime tools were selected for the analysis. Of the many tools trialled, four were chosen: slingshot, tradeSeq, Monocle 2 and Monocle 3. This was based on examination of the literature and the use of the tools on the data[8]. Other tools tested were not found to provide biological insight beyond what was offered by the chosen tools. tradeSeq was used downstream of Slingshot. The use of tradeSeq downstream from the Monocle packages was considered, however, as shown in Van den Berge, Roux de Bézieux([2]), this is not recommended.

3.2 Program Design

The existing application, found at http://cellatlas.mvls.gla.ac.uk/, allows the user to adjust the resolution and view the changes to the UMAP and differential expression of genes. All figures and tables in the application are updated when the user changes the resolution. Users can navigate between the different tabs on the Shiny application. Figures 2 and 3 depict what was present in the existing atlas (blue) and what was added in this project (green).

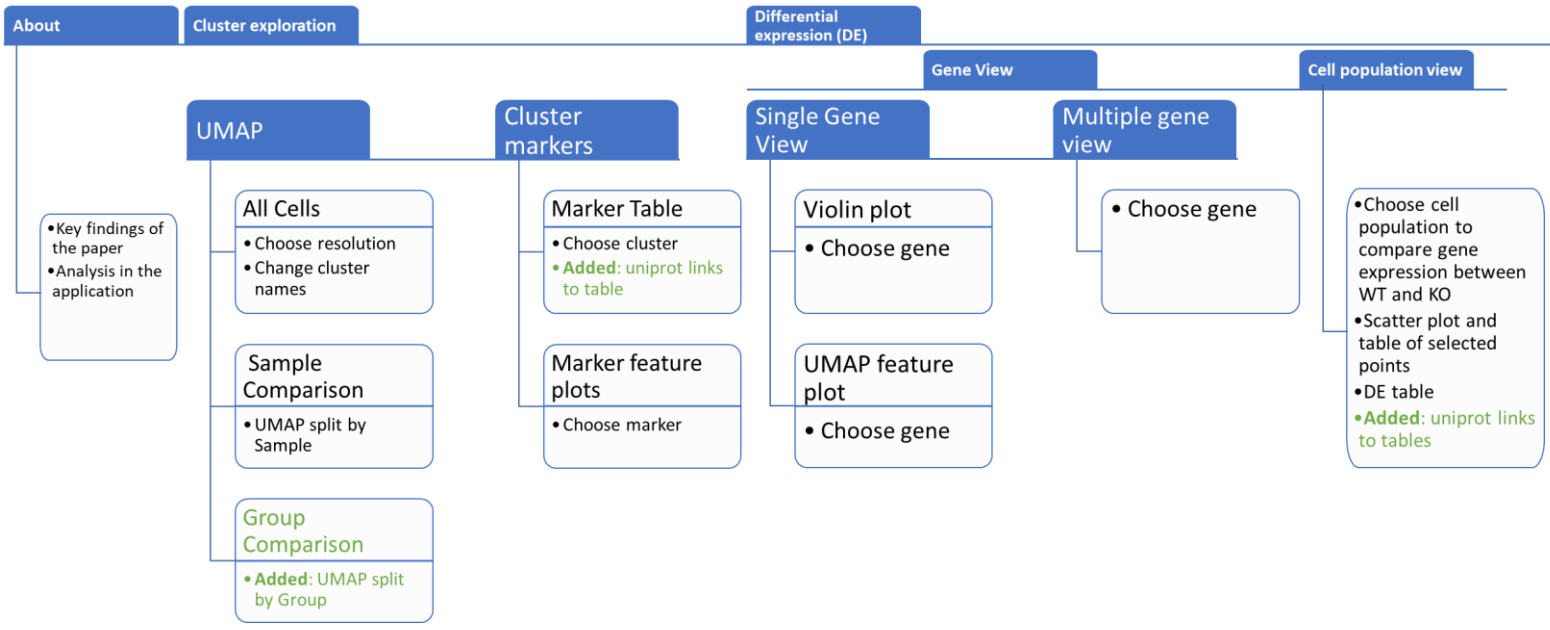

*Figure 2: **Overview of Single Cell Analysis Tabs.** The tabs available for cluster exploration and differential expression. Nested tabs are indicated by layers. Blue = present in existing atlas, green = added in this application.*

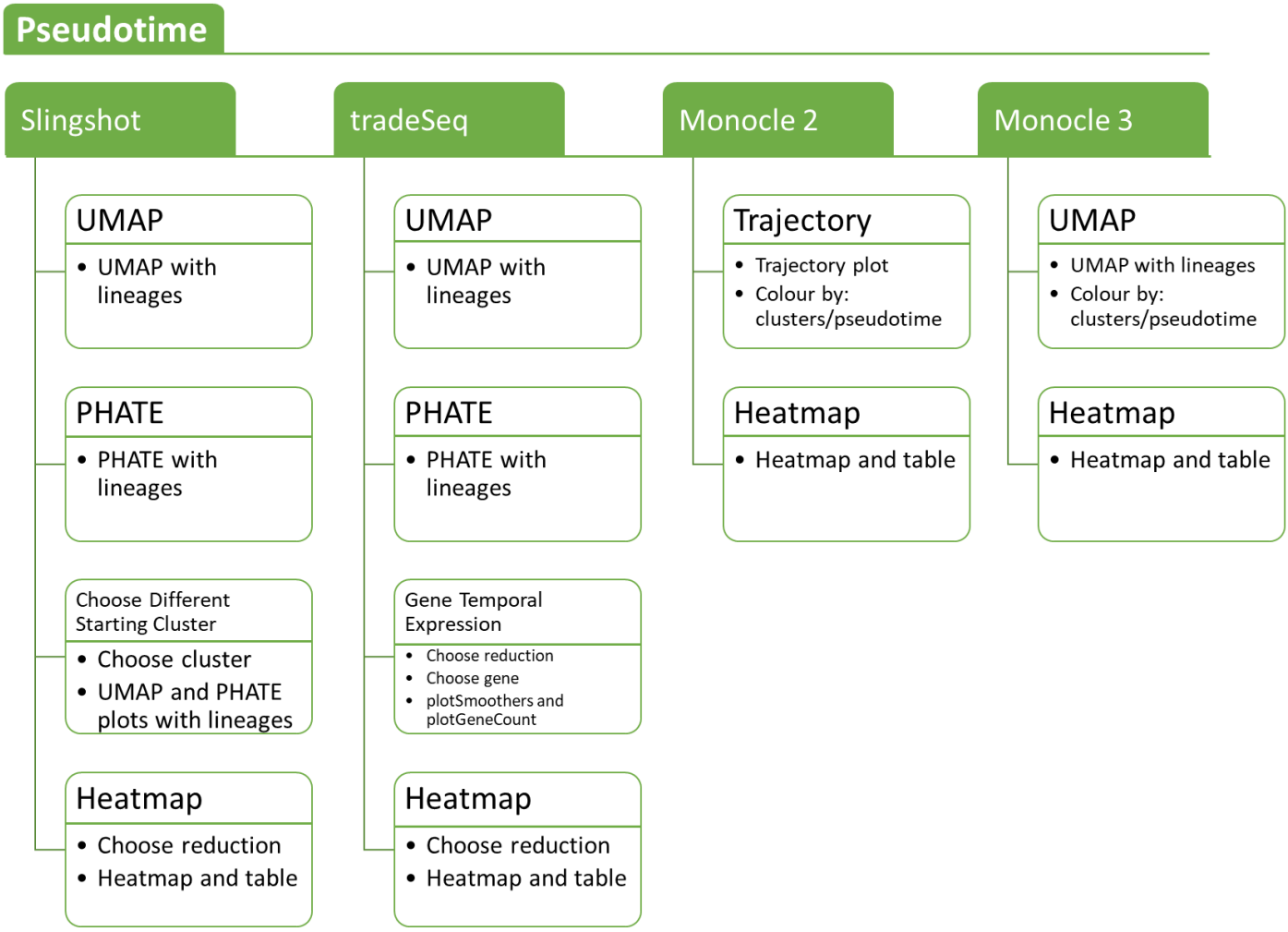

*Figure 3: **Overview of Pseudotime Analysis Tabs.** The tabs available for Pseudotime analsis. Each of the four tools have plots showing the trajectories and the heatmaps. Nested tabs are indicated by layers.*

To the existing single-cell analysis tabs, a plot to allow the user to view the UMAP split by groups was added (under the UMAP tab). Furthermore, links to UniProt were included in the tables for each of the genes to allow the user to further investigate the genes (Figure 4).

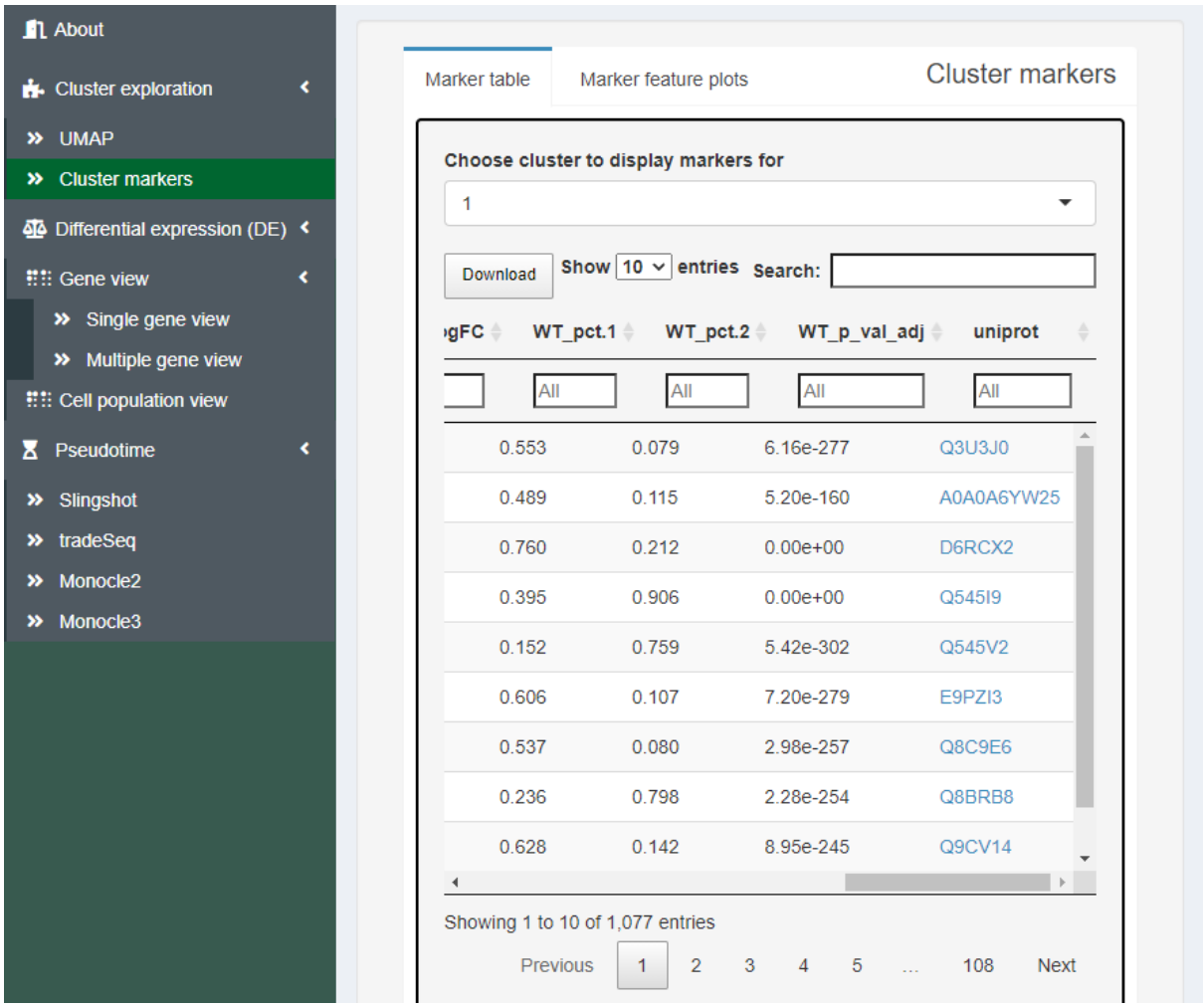

Figure 4: ***Marker Table with UniProt Links.*** *Example of UniProt links added to the existing Markers Table in the application.*

Each tab also includes an information box to provide insight into the analysis. Furthermore, there are options to download the plots and change the download specifications. These features were adapted from the existing atlas.

3.2.1 Pseudotime Application Design

In the additional pseudotime section, the user can view the different tabs for each of the four pseudotime tools (Figure 3). Each tool has tabs of different plots and a heatmap (Figure 5). The plots and tabs vary between each tool, but each tool has a heatmap and a table of the associated genes. The heatmaps are produced using the appropriate method to select genes of interest for the tools. As with the existing atlas, the outputs are updated when the user changes the resolution.

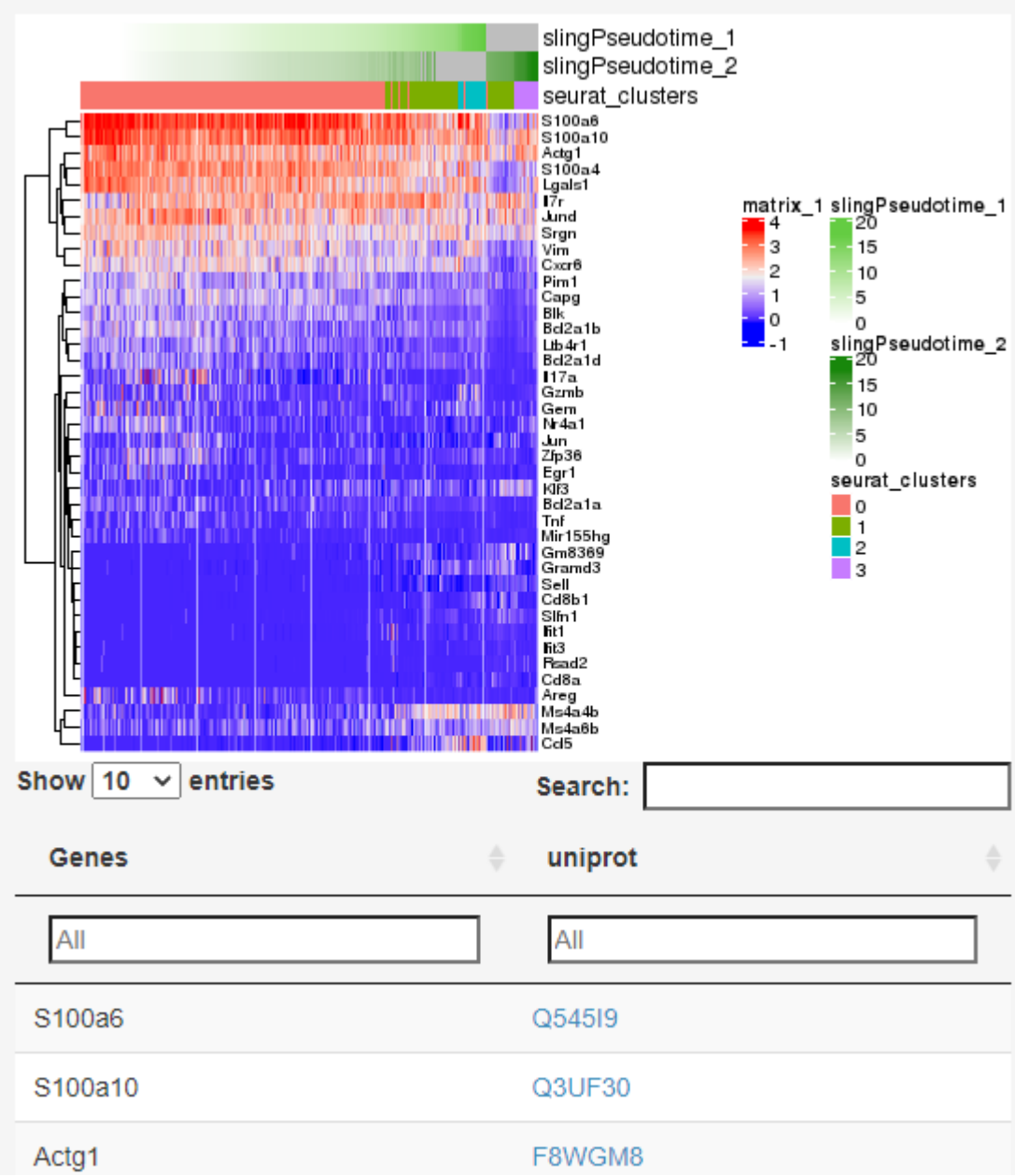

Figure 5: ***Heatmap Screenshot.*** *Screenshot of the heatmap and associated gene table taken from the application. UniProt links are present in the table. Example of Slingshot UMAP heatmap from the T cell application. Names may differ in UniProt links due to issues with gene entry names with UniProt and is not an issue with the application.*

Slingshot

The Slingshot tab has both UMAP and PHATE reduction lineage plots (Figure 6). These show the clusters and the lineages identified by the tool. Additionally, the user can choose a different starting cluster and view the UMAP and PHATE plots to see the effect on lineages inferred (Figure 7).

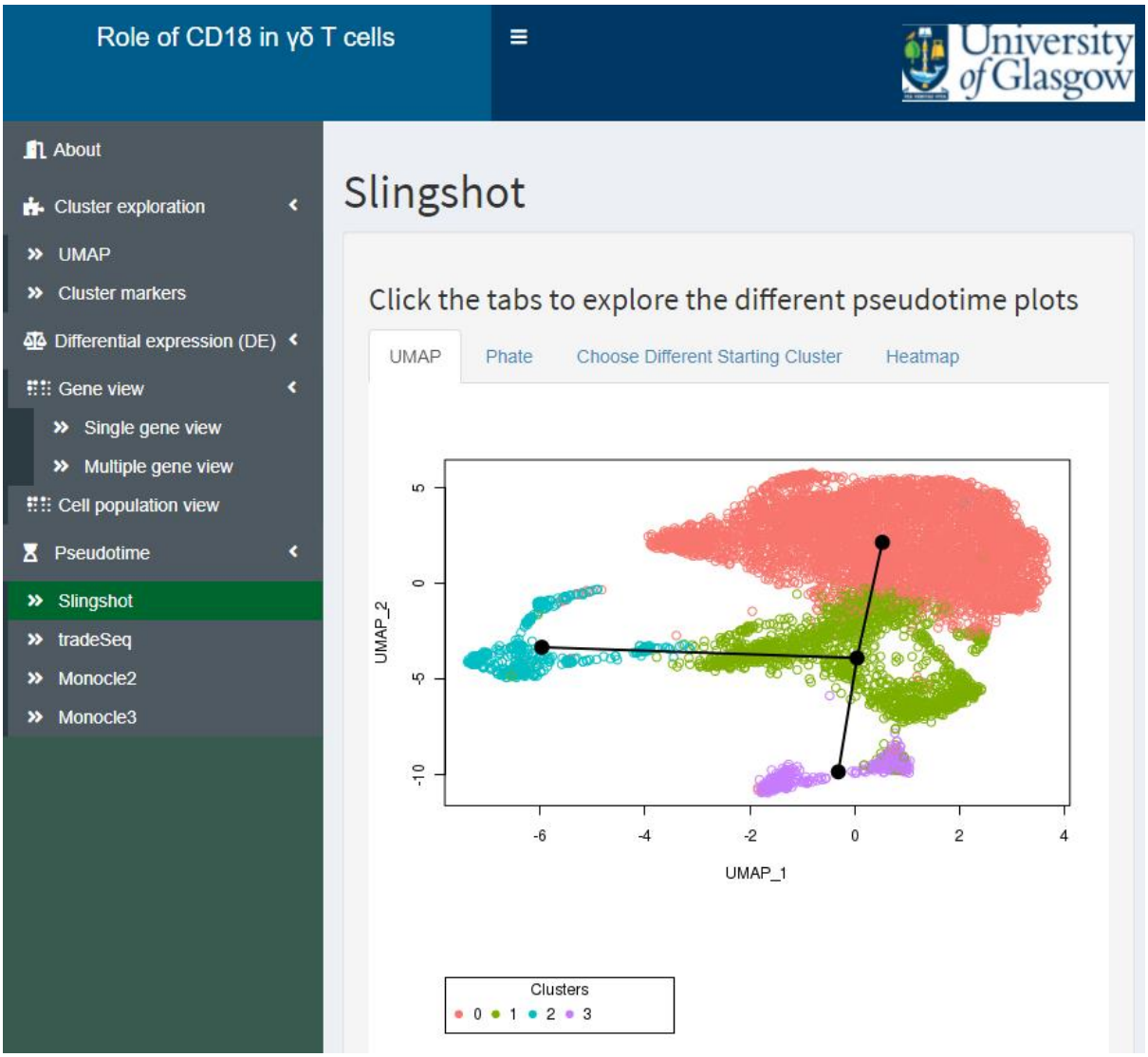

Figure 6: ***Slingshot tab from the Application****. Example of UMAP plot from T Cell pseudotime application*

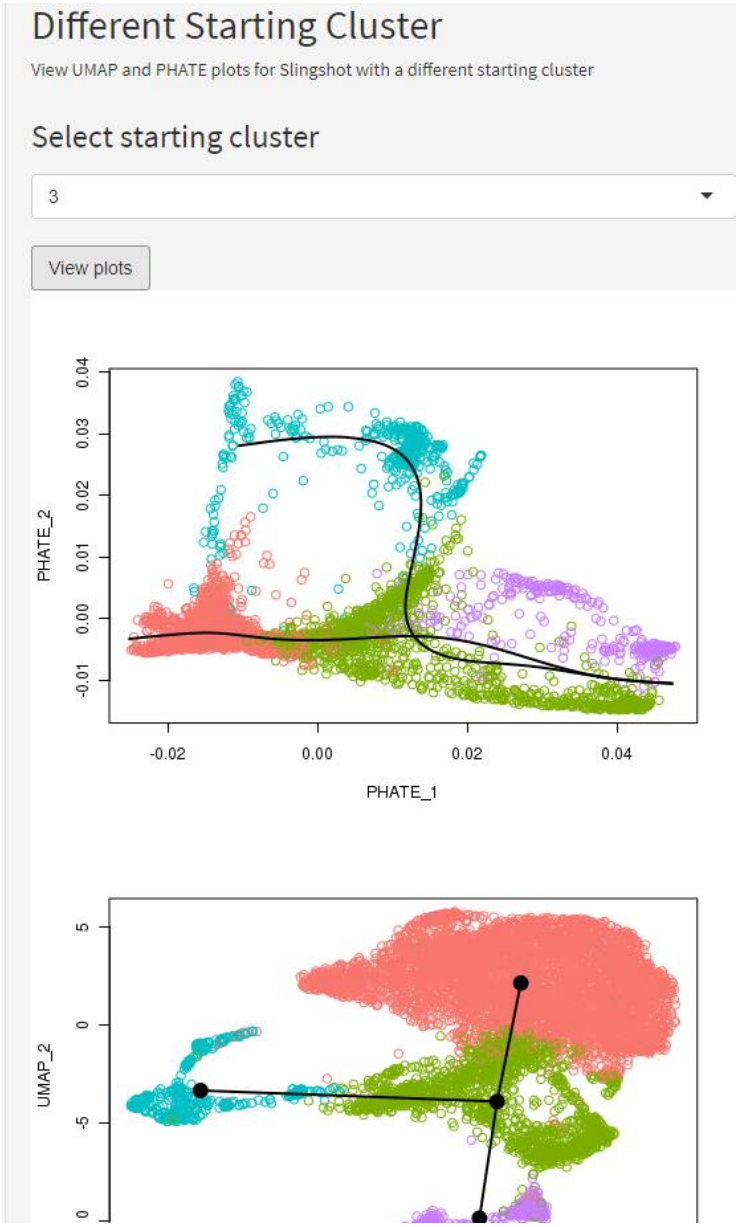

Figure 7: ***Slingshot Choose Different Starting Cluster Tab****. Figure from T cell application where the user can choose a different starting cluster and dynamically load the objects for the PHATE and UMAP plots for only the specified cluster and resolution.*

tradeSeq

UMAP and PHATE plots are also available in the tradeSeq tab (Figure 8). The user can view the 'Gene Temporal Expression' tab, unique to tradeSeq (Figure 9). This tab allows the user to select the genes identified in the heatmap, i.e., associated with pseudotime, and investigate their expression as a function of pseudotime or on the UMAP space. The user also must choose the reduction (PHATE or UMAP) in the 'Heatmap' and 'Gene Temporal Expression' tabs.

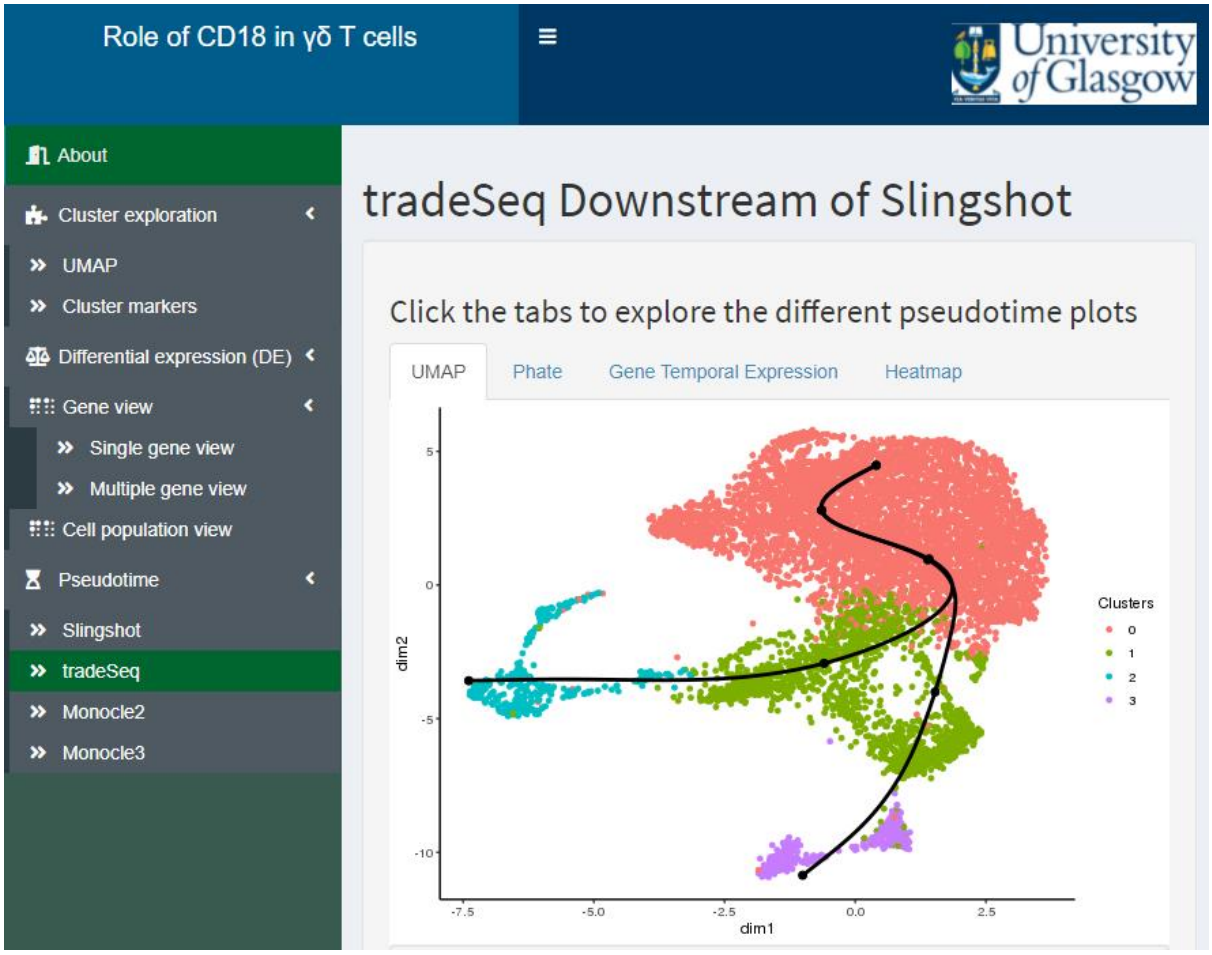

Figure 8: ***tradeSeq tab from the Application****. Example of UMAP plot from the T cell application*

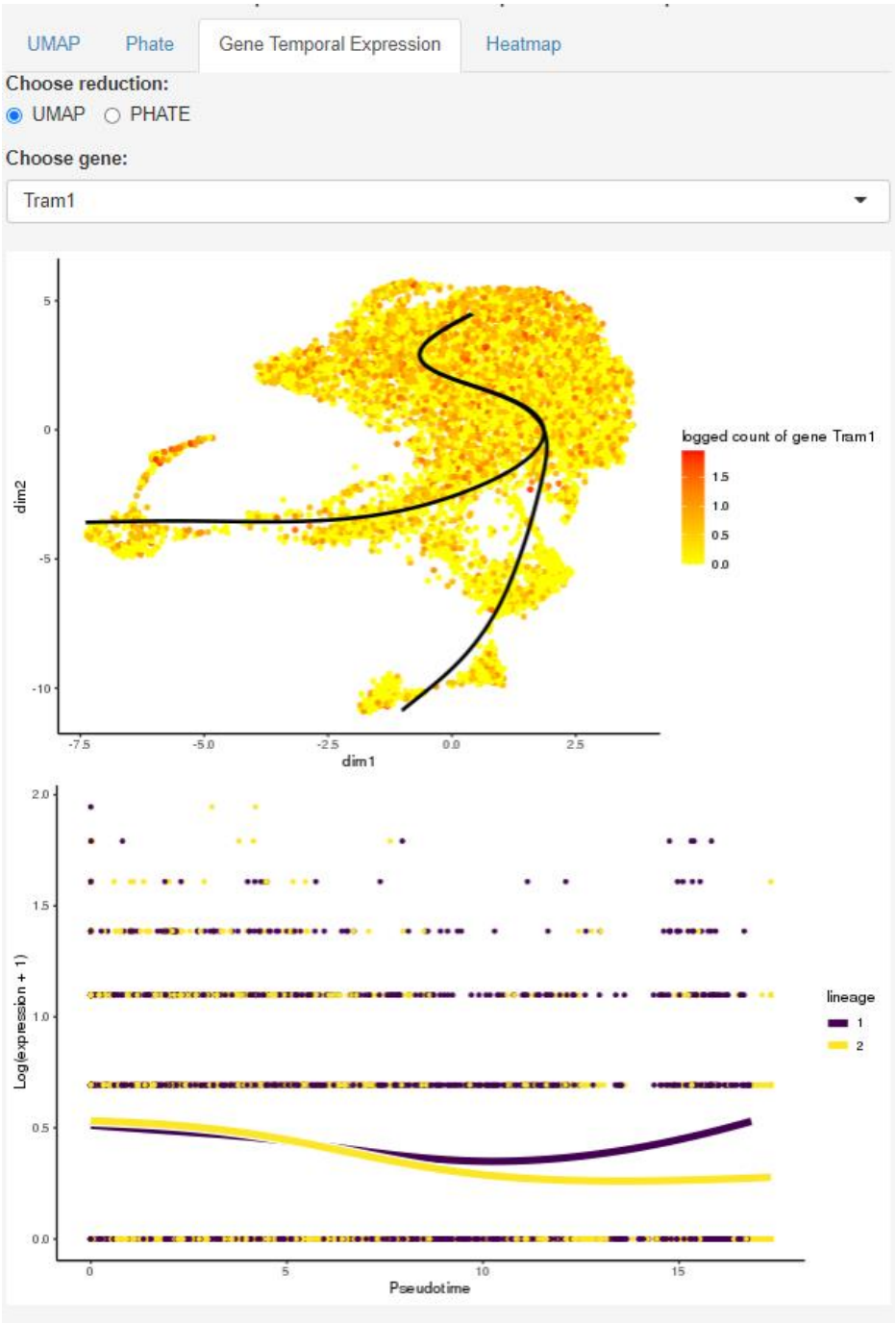

Figure 9: ***tradeSeq Gene Temporal Expression Tab.*** *Screenshot from T cell application where the user can choose a gene and view the expression as either a function of pseudotime or on the UMAP space. User has the choice of UMAP or PHATE reduction.*

Monocle 2

A trajectory plot is offered in the Monocle 2 tab (Figure 10). It allows the user to view the branches associated with cell fate decisions. This plot can be coloured by clusters or pseudotime.

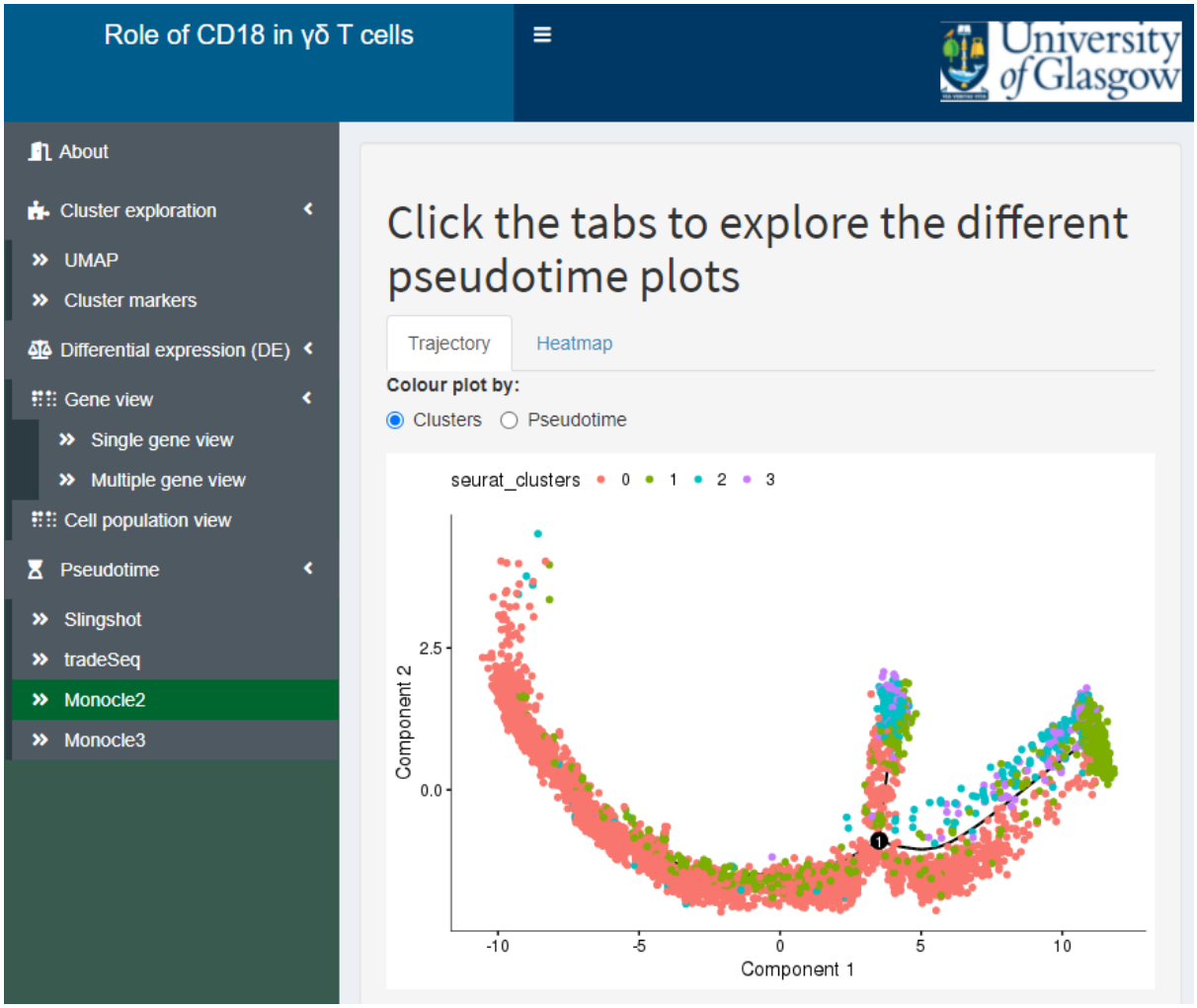

Figure 10: ***Monocle 2 tab from the Application****. Example of trajectory plot from the T cell application.*

Monocle 3

The Monocle 3 tab also has a UMAP plot that can be coloured based on clusters or pseudotime (Figure 11).

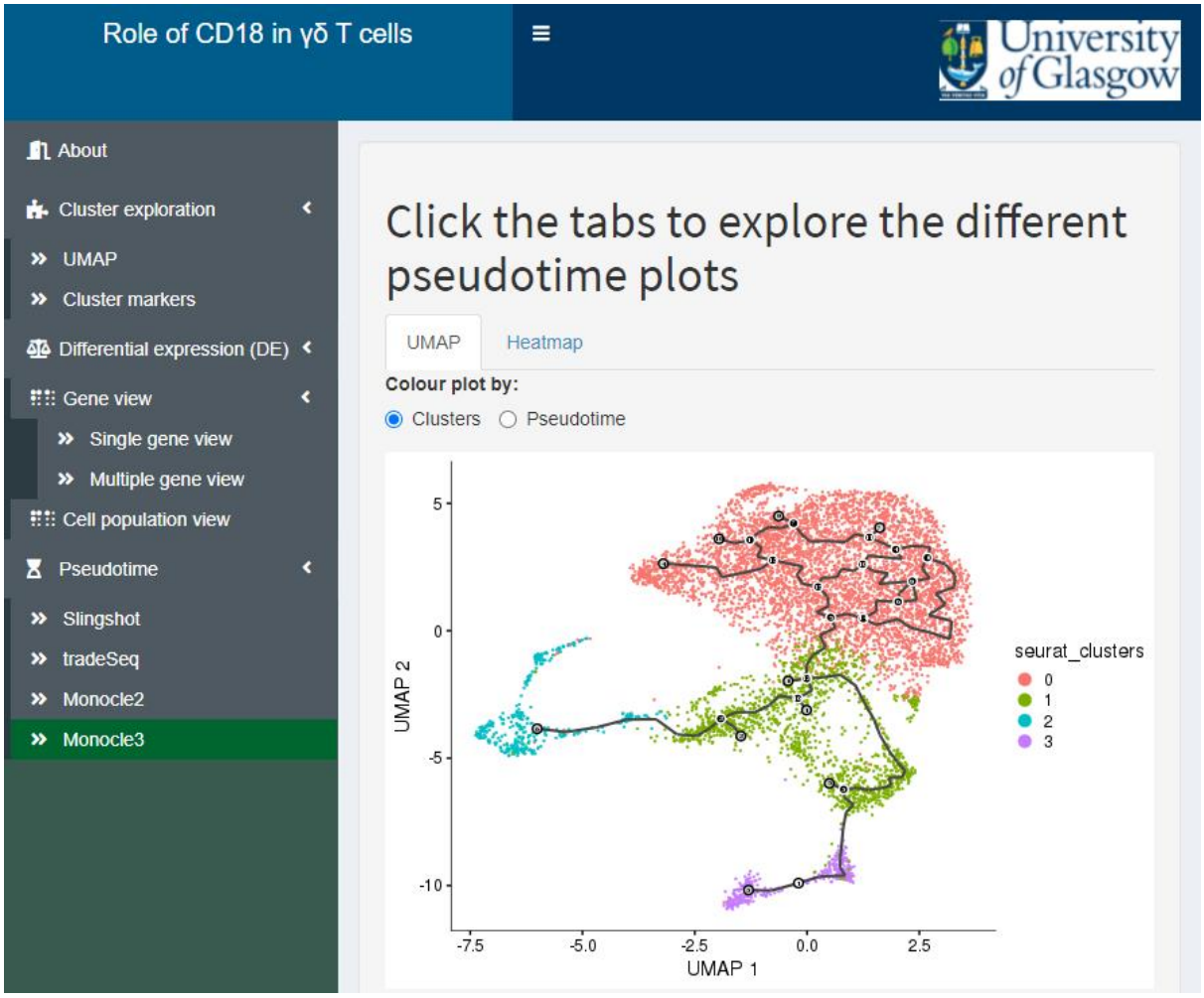

Figure 11: ***Monocle 3 tab from the Application****. Example of UMAP plot from the T cell application.*

3.2.2 Upload Application Design

All the features mentioned previously, apart from the tab to choose a different Slingshot starting cluster, are available on the Upload application. There also exists an extra tab, 'Upload Data'. This is deeply linked to the analysis pipeline, with user input required for each step of the standard Seurat pipeline. The different tabs and input boxes are controlled via panels which are shown and hidden, dependent on criteria.

Figure 12 depicts the input and output to the user when uploading data. Each tab will be described further below. When required, users can either provide the necessary parameters on the current tab (by clicking a checkbox) or can wait to view the relevant plots on the next tab and then input the values. This allows the user to go through the analysis more quickly if they are already aware of what values suit their data.

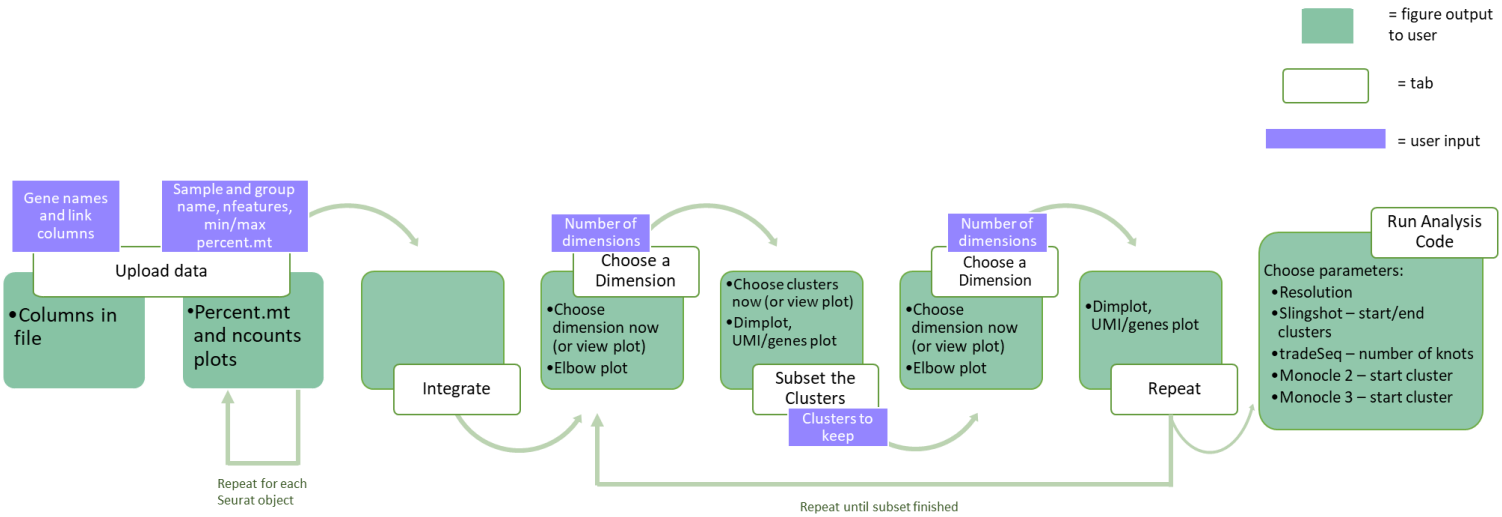

*Figure 12: **Upload Application User Input/Output**. Describes the different inputs required from users on each tabs and the plots shown to the users. Tabs are navigated using buttons in the direction indicated by arrows.*

Figure 13 illustrates the tabs shown in the application.

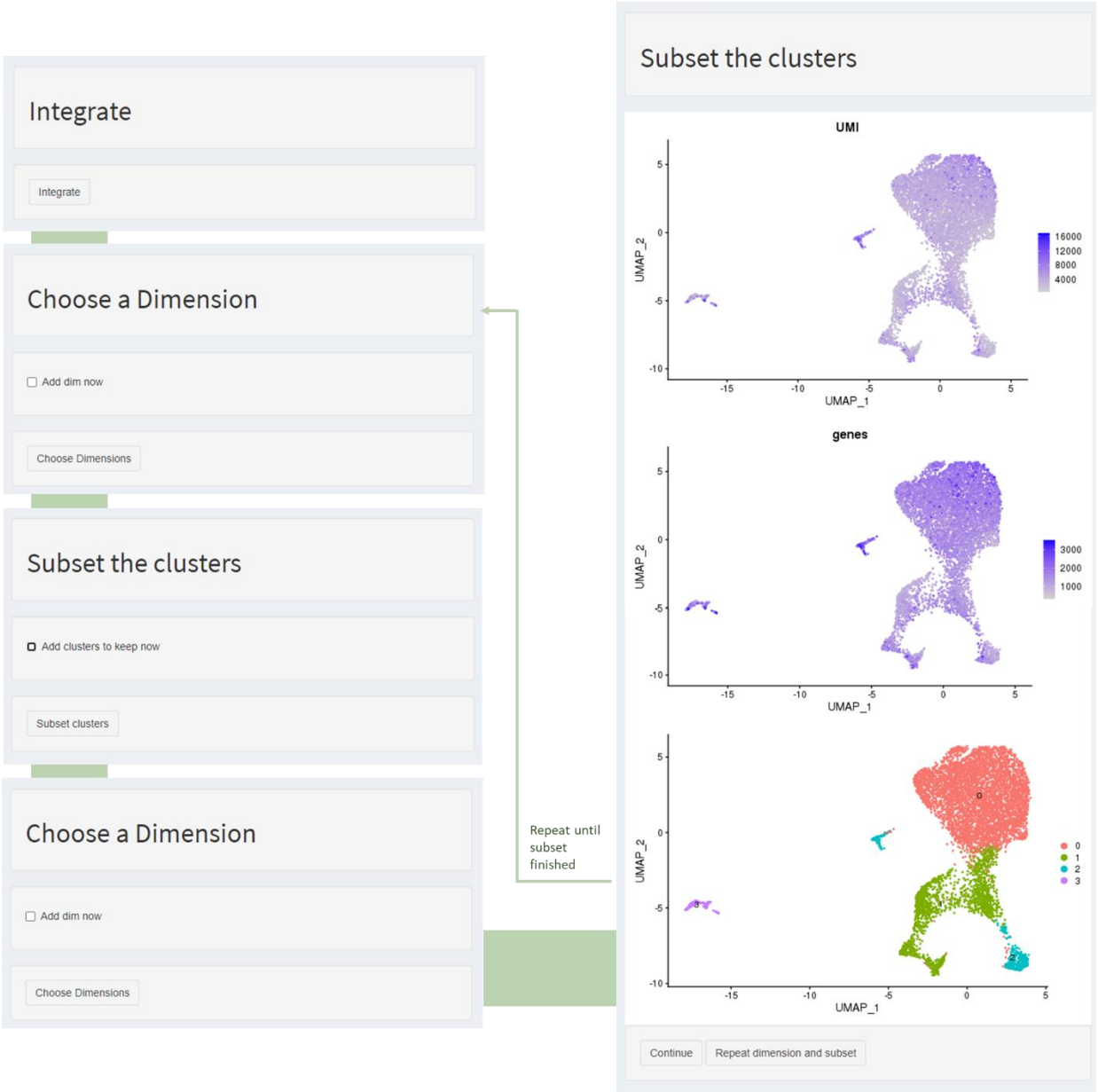

Figure 13: ***Upload Application Single Cell Analysis.*** *Screenshots from the Upload application demonstrating the single cell analysis workflow. In order of Green line starting from Integrate.*

First, the user uploads the objects. For the UniProt file, the user selects the columns for the gene names and the links. The user also uploads the Seurat object and provides a sample and group name (i.e., WT1 and WT, respectively). The user also inputs the mitochondria and number of features to subset (percent.mt and min/max nfeatures, respectively). The user can choose to view an elbow plot before providing the subset-specific parameters.

After the data is subset, the user can choose to upload a new Seurat object (and repeat the preceding steps) or to proceed to integration. This is achieved via buttons that are active/inactive based on the decisions the user makes.

The then user clicks a button to integrate the data. Next, the user chooses how many dimensions of the data to use. The user can choose to view an elbow plot to inform their decision. The user selects the clusters they want to keep. They can choose to first view the DimPlot, unique molecular identified (UMI) and gene plots or to select the clusters there. The user then chooses the number of dimensions again to re-cluster the data based on the newly subset clusters.

The DimPlot, UMI and gene plots for the new clusters are displayed and the user is asked if they want to further subset the clusters or if they want to proceed. If they choose to subset again, they will be taken to the subset page and will have to select the dimensions once more. If they choose to proceed, they will continue to the next page to run the single-cell and pseudotime analysis.

To upload and perform the standard Seurat pipeline (as has been described so far) takes 10-20 minutes. This, however, can depend on the size and number of Seurat objects uploaded.

Finally, the user can select the variables to use as the code runs through the loops, repeated for each resolution, to generate the objects required in the application (Figure 14). The user can specify the range of resolutions they want to look at. They can also specify any pseudotime parameters they want to use. For each tool, the user can choose (once more using checkboxes) if they want to change the parameter. If they do not click the box, default parameters will be used.

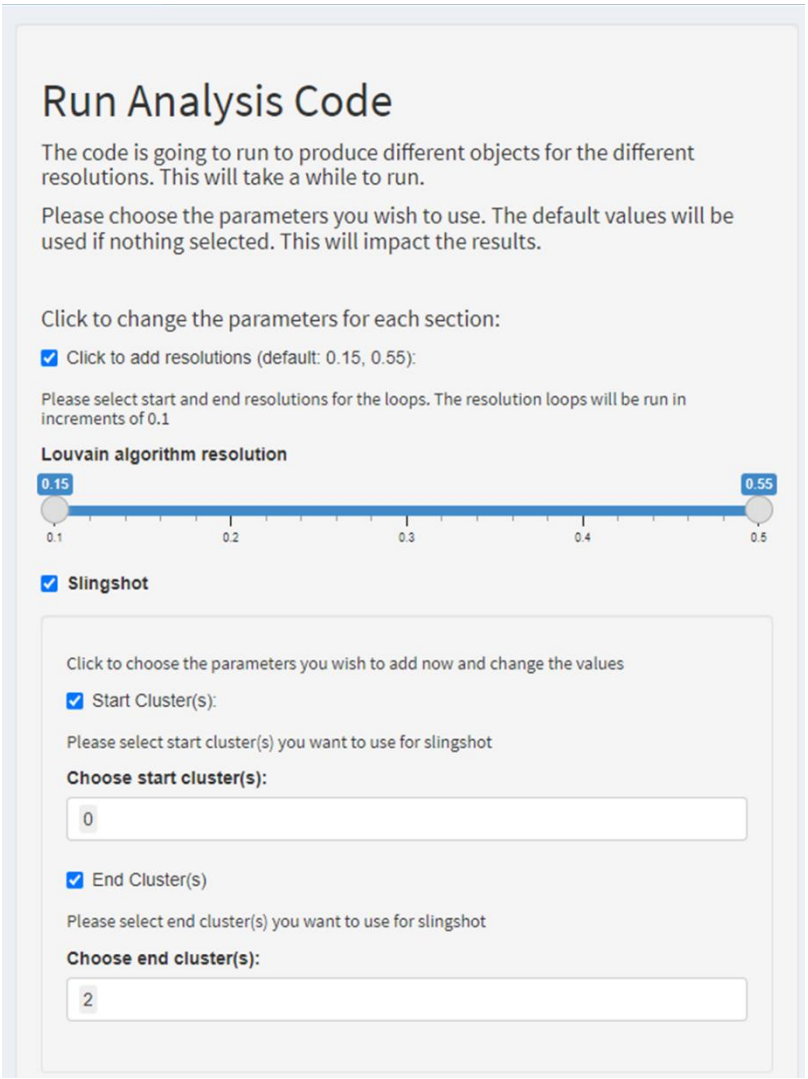

*Figure 14: **Pseudotime Parameters from Application**. Screenshot of the tab from the Upload application where the user can specify the parameters for Pseudotime analysis if they want. If no parameters are selected, default values/no parameters are used.*

The user is then shown a page instructing them not to close the application as the process is running. When the pseudotime analysis is finished, a button will appear that the user can click and view the tabs with the results.

3.3 Program Analysis Pipeline

3.3.1 Existing Single-Cell Pipeline

The existing atlas, Figure 15 (blue), hardcoded the upload of datasets to create Seurat objects which were integrated. The integrated object was clustered. The following single-cell analysis was repeated for each resolution. First, the clusters were found. Next, the conserved markers. Markers were also found and finally, plots and tables were produced for differential expression analysis. The existing application allows the user to read these files produced previously.

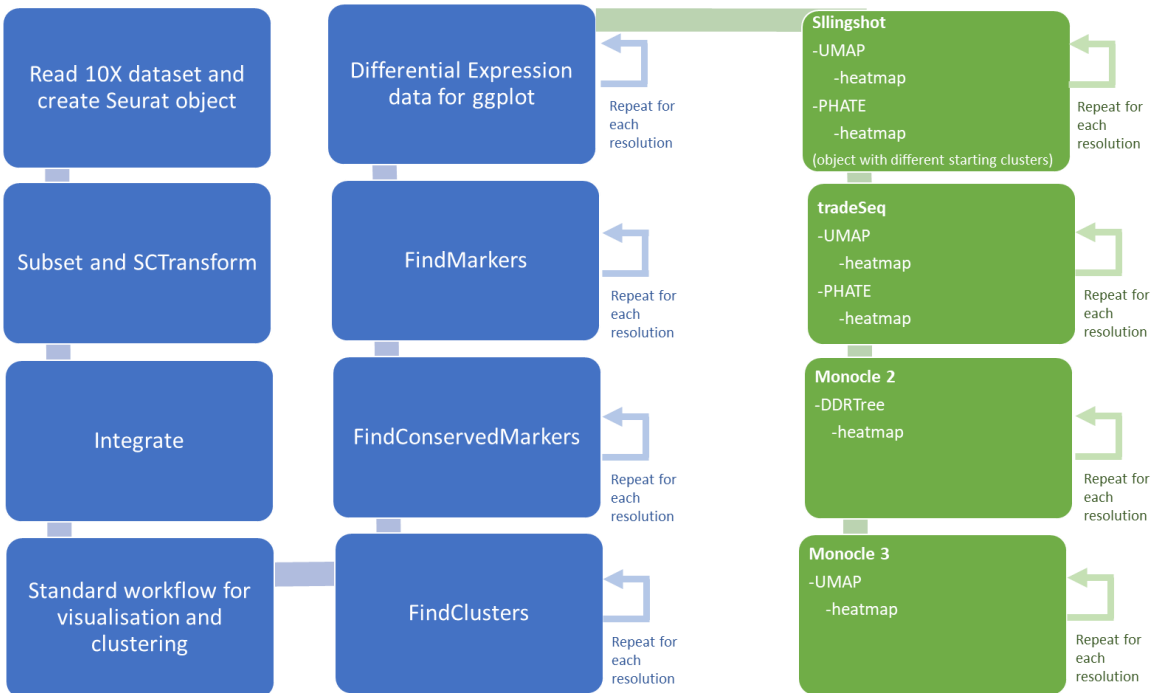

*Figure 15: **Workflow of the Pre-Processed Analysis for the Applications**. Pre-processing steps of the T cell and Malaria applications. Blue = from existing application, green = added here. Arrows indicate where steps are repeated for each resolution. Workflow starts at top left corner and ends at bottom right.*

3.3.1 Added Pseudotime Pipeline

Pseudotime analysis of the application, Figure 15 (green), was performed using the four tools selected previously: Slingshot, tradeSeq, Monocle 2 and Monocle 3. Each tool was used according to their vignettes and using the Seurat object created previously from the standard Seurat pipeline.

Heatmaps were produced for each tool using the R package ComplexHeatmaps. The top 40 genes were selected based on the tool's method of identifying variable genes for pseudotime. Slingshot does not have a method for this so a method from Seurat was used instead to find variable features rather than those associated with Pseudotime. It should be noted that there are still caveats in the comparison of the different methods in identifying the genes affected by

pseudotime. For example, all heatmaps apart from tradeSeq were sorted by q value. This is because tradeSeq does not provide an accurate FDR. It was instead sorted by p-value, a less effective alternative.

3.3.1 Upload Application Pipeline

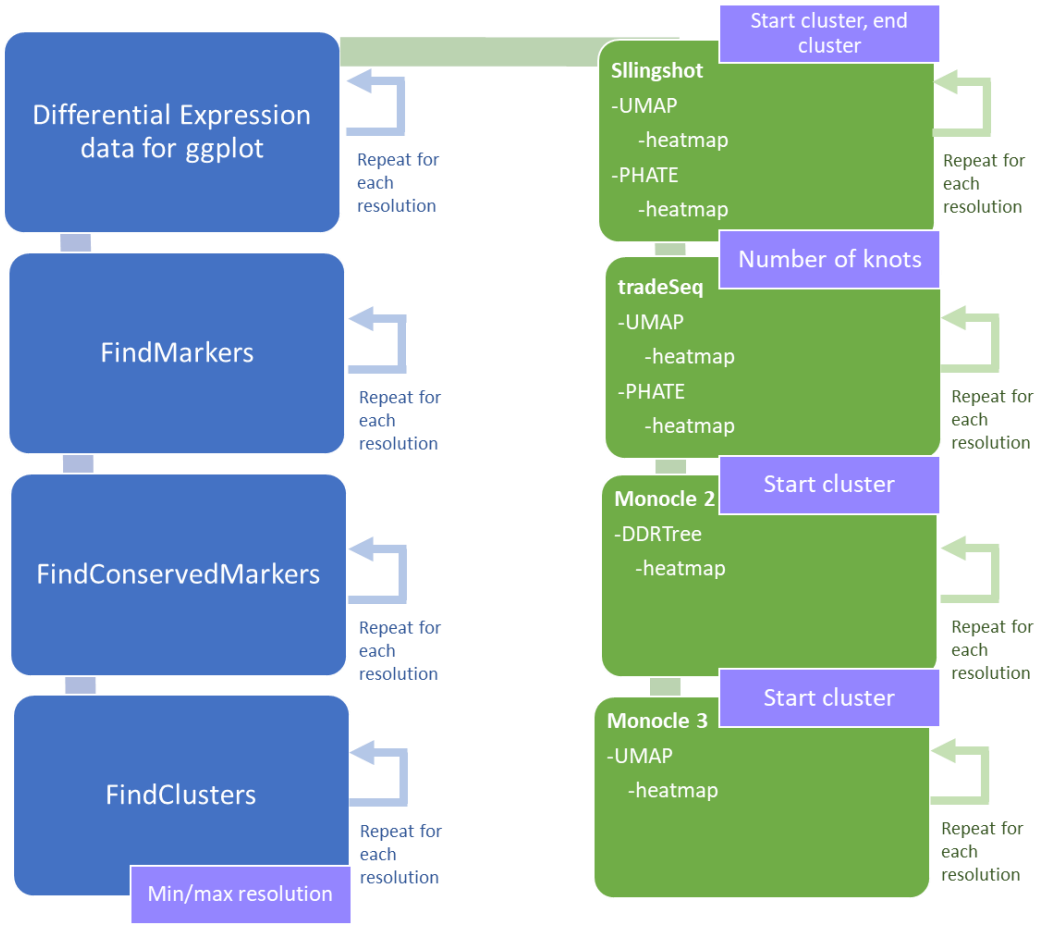

*Figure 16: **Workflow of the Upload Application**. Blue = from existing application, green = added. Purple boxes indicate parameters from the user. Arrows indicate where steps are repeated for each resolution. Workflow starts at bottom left and ends at bottom right.*

The pipeline is performed similarly to the other applications except the user can specify parameters for the analysis of the data. As shown in Figure 16 and Table 4, these include the minimum and maximum resolutions to use for the looping of the analysis; 0.15-0.55 in increments of 0.1. For Slingshot, the user can choose end and start cluster(s). This is an optional feature of Slingshot analysis, and if not selected, no clusters are specified. The number of knots can be specified for tradeSeq, here a default of 3 is used (to reflect the default of Monocle 2). For Monocle 2, the user can provide the root cluster for the pseudotime

analysis. If not specified, Monocle 2 will calculate this. The starting cluster can also be specified for Monocle 3.

Tool	Parameter	Description	Default
Slingshot	Start cluster	Root cluster(s) for pseudotime analysis	NULL
	End cluster	End cluster(s) for pseudotime analysis	NULL
tradeSeq	Number of knots	Used in fitGAM	3
Monocle 2	Root state	Monocle 2 is not cluster-based so it finds the state associated with that cluster to use as the root state[1]	NULL
Monocle 3	Root node	Monocle 3 is not cluster-based so it finds the cells associated with that cluster to use as the root node[1]	NULL

Table 4: ***Summary of Upload Application Pseudotime Parameters***

[1]The root node/state from Monocle 2 and 3 are used to find the associated starting cluster

4. Evaluation

The pseudotime application was evaluated with two different datasets: T cell and malaria. The Upload application was evaluated using the T cell dataset. The Malaria application was further evaluated by other users, who then filled out a questionnaire.

The differences in the tools were investigated by looking at the trajectories identified, the heatmaps produced and other features. A significant barrier to the applications is the time they take to load. This was also explored.

4.1 Evaluation with T Cell Dataset

T cell data from McIntyre, Monin([6]) was investigated using the T cell application. The existing T cell application was modified to include pseudotime analysis. There are five different resolutions available to the user (0.15, 0.25, 0.35, 0.45, 0.55). All resolutions were explored, 0.15 is shown here.

All tools followed a similar pattern in start and end clusters with variation in the nature of the trajectories inferred (Figure 17). Monocle 3 was able to identify topologies omitted by the other tools; Monocle 3 identified cyclical trajectories that the other tools did not.

Furthermore, Monocle 3 identified many more points. This could be viewed as adding too much noise to the analysis with many different branches on the plots. However, it can also aid in understanding and may detect biologically relevant branches missed by other tools. The inclusion of the four tools allows the user to decide this for the data.

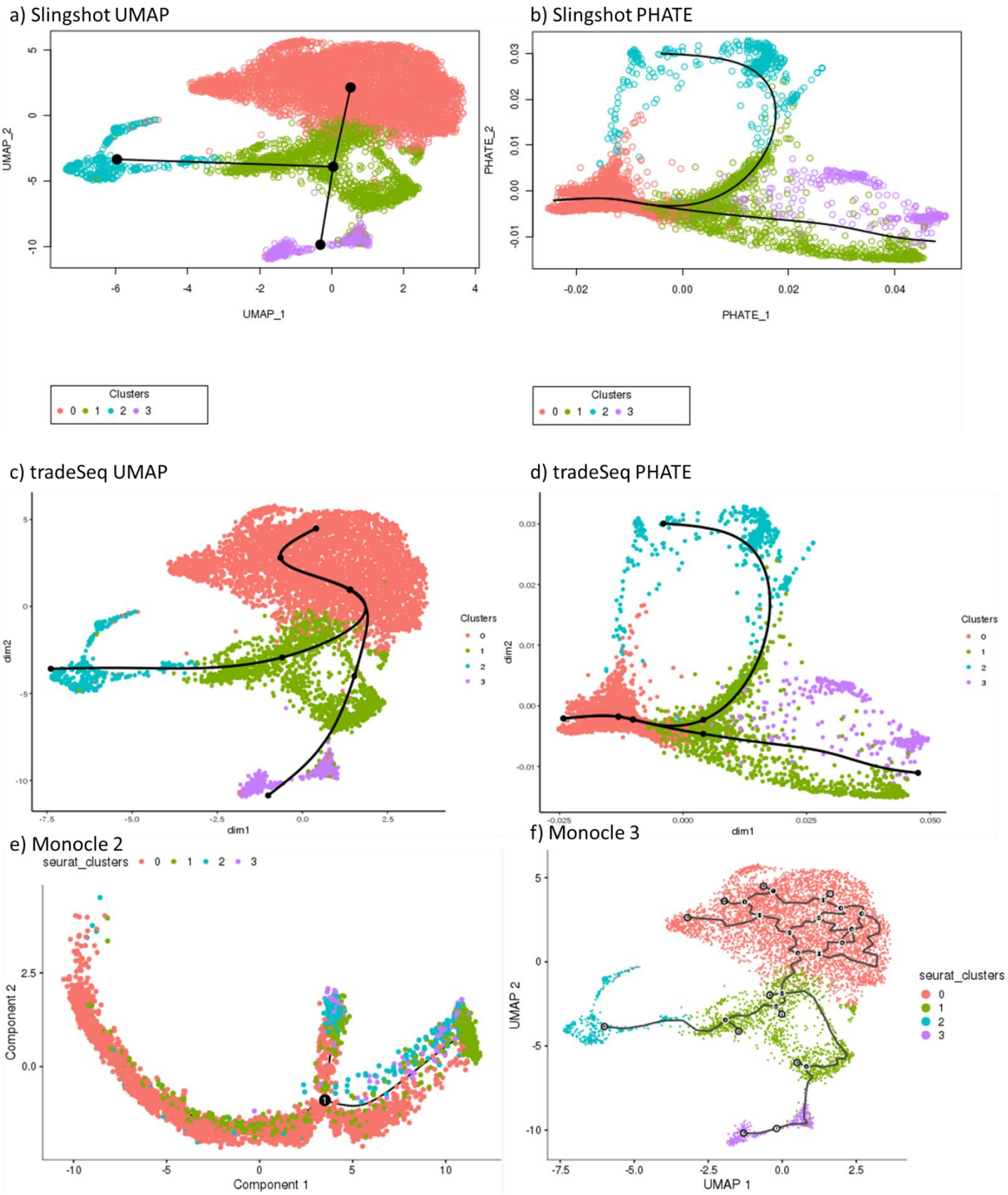

Figure 17: ***T Cell Application Trajectory Plots****. Screenshots taken from the T cell application for resolution 0.15*

The response time of the application was also evaluated. All plots, except the heatmaps, took less than 5 seconds to load. A limitation of using the ComplexHeatmap package is the time it takes for the heatmaps to load. However, each heatmap for the T cell application loads in under a minute (Table 5) and a loading sign is shown to the user as they load, suggesting this is not a major constraint. These loading times were tested with the five different resolutions and had little variation.

Tool	Time
Slingshot	7 seconds
tradeSeq	42 seconds
Monocle 2	24 seconds
Monocle 3	22 seconds

Table 5: ***T Cell Application Heatmap Loading Times***

The time it takes to load the heatmaps for each tool in the T cell application.Results here for resolution 0.15 but there was little variation between the resolutions.

Venn diagrams of the genes identified by the different tools for the heatmaps show variation within and between tools (Figure 18). Furthermore, the Slingshot heatmaps identified variable features and not those explicitly association with psuedotime, however there was still overlap with other tools which are more specific to pseudotime. This suggests the use of find variable features from Seurat still provided relevant results, although they should be taken with caution and reference to the other tools. Overall, Figure 18 supports that the different tools offer different information and the inclusion of the four tools is necessary to convey this.

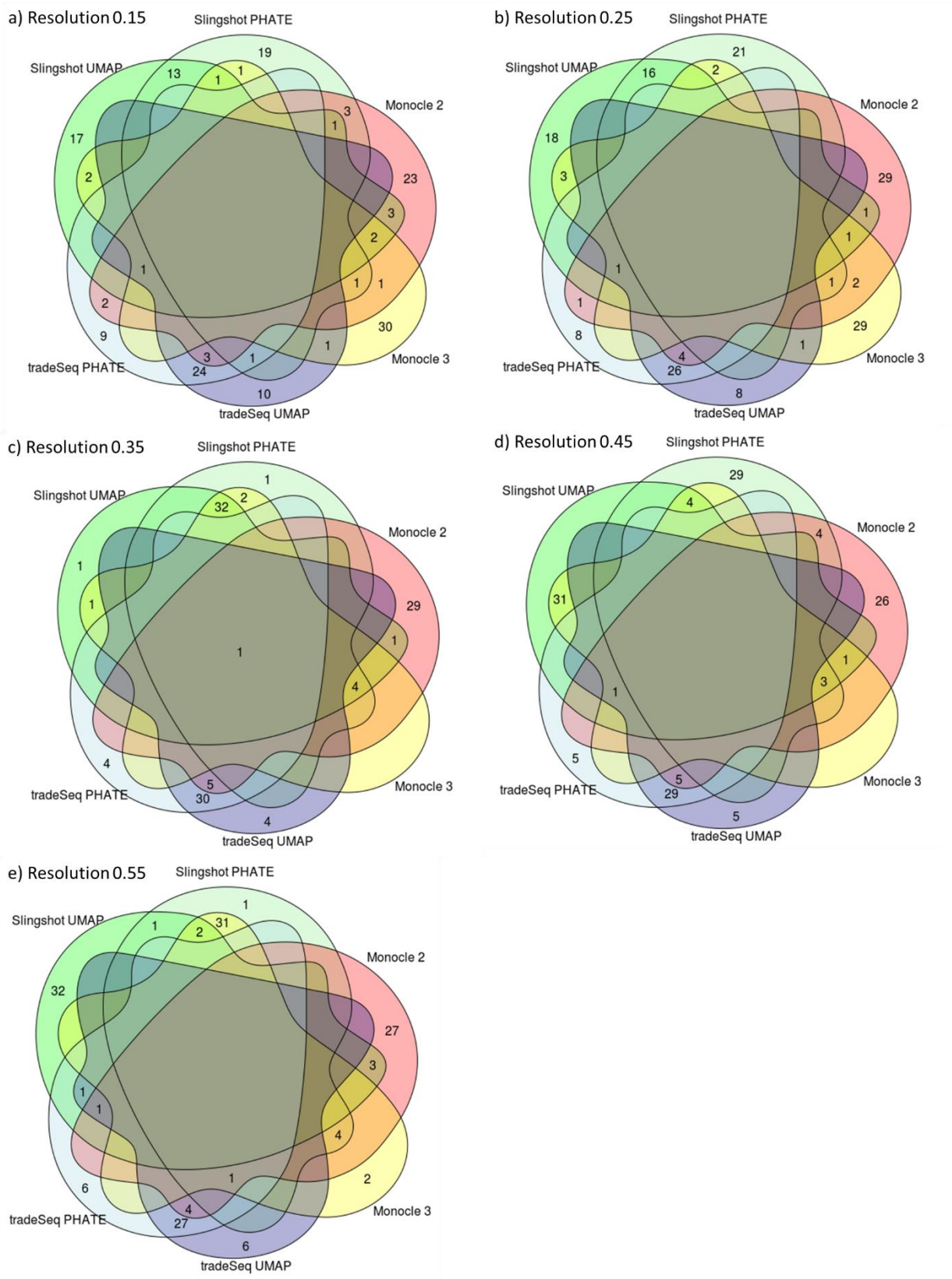

Figure 18: ***T Cell Heatmap Genes Venn Diagrams.*** *Venn diagrams of the genes identified in the heatmaps for each tool. Venn diagrams for each of the five resolutions of the T cell application. Produced using the package Venn in R. Appendix B. Venn Diagrams Code*

4.2 Evaluation with Malaria Dataset

An application was also produced for the malaria data produced from Hentzschel, Gibbins([5]). Only one resolution (0.51) is available, to match the existing single-cell atlas.

In addition to offering more than one tool, several reductions are also offered. Evaluation of these reductions supported that they allow the user to best find biologically relevant results. Based on the biology of the malaria life cycle, outlined in Hentzschel, Gibbins([5]), the lineages present for this dataset should have three endpoints. Slingshot UMAP reduction did not find all three, only two were identified (Figure 19). PHATE did however identify all the expected endpoints.

As expected, the same was found in tradeSeq. Furthermore, Monocle 2 identified two branches and Monocle 3 only identified two endpoints. The Monocle 3 UMAP plot suggests there are many more than three endpoints, and the endpoints did not follow the pattern expected and supported by Slingshot and tradeSeq PHATE reduction, i.e., cluster 8 to 9, 11, 12.

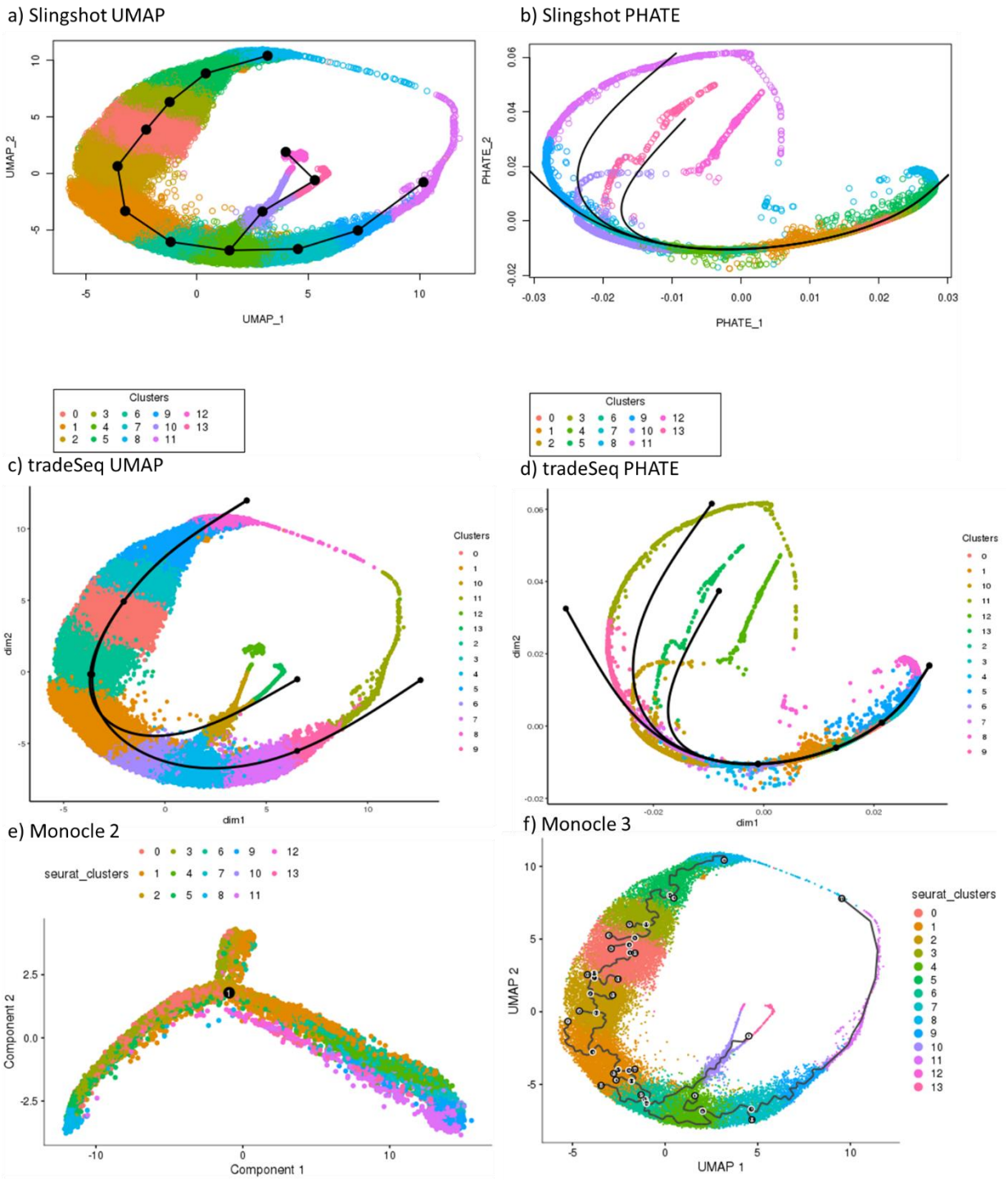

*Figure 19: **Malaria Application Trajectory Plots**. Screenshots taken from the Malaria application. The colours of the clusters in tradeSeq do not match the other tools, this is because the numbers over 10 are ordered incorrectly and therefore colours are assigned incorrectly.*

Figure 20 illustrates that there were no genes in common for the two reductions in Slingshot, supporting the offering of both reductions as they both offered different results. tradeSeq showed overlap but still some variation of genes identified. A similar trend can also be seen

between the different tools. This further illustrates the differences between the tools and the adaptability allowed in offering different tools and reductions.

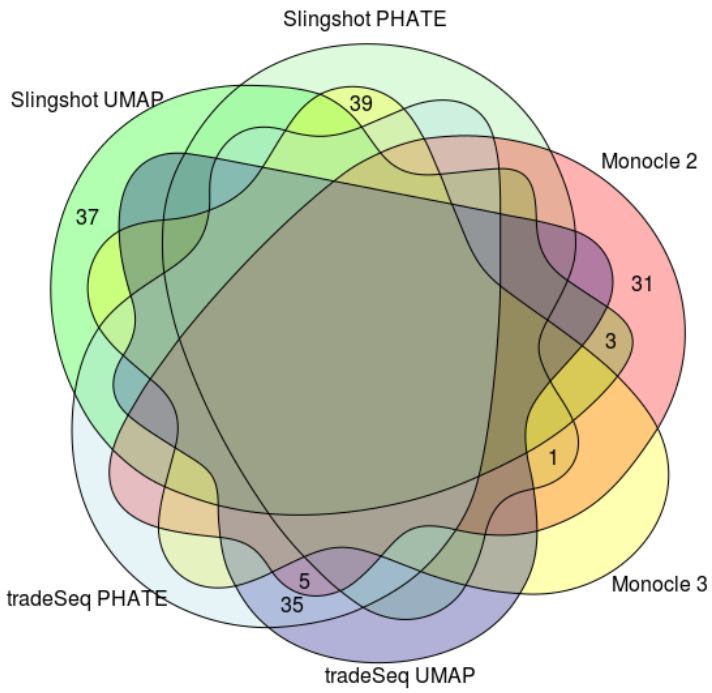

Figure 20: ***Malaria Heatmap Genes Venn Diagrams****. Venn diagrams of the genes identified in the heatmaps for each tool. Venn diagrams for the one resolution. Produced using the package Venn in R*

The heatmaps for this application took significantly longer to load compared to the T cell application (Table 6).

Tool	**Time**
Slingshot	7 seconds
tradeSeq	10 min 19 seconds
Monocle 2	4 min 45 seconds
Monocle 3	5 min 14 sec

Table 6: ***Malaria application Heatmap Loading Times***

The time it takes to load the heatmaps for each tool in the Malaria application.

Furthermore, the other pseudotime plots available do have slightly longer waiting times for tradeSeq (UMAP took 30 seconds and PHATE took 47 seconds). This most likely can be attributed to the increased number of cells and clusters in this dataset. It is unlikely future work would be able to improve this, but it would still be worthy of investigation.

4.3 Evaluation of Malaria application by Users

A questionnaire was made available along with the Malaria application. Users were asked about speed, ease of use, customisability, and any other concerns they had (Appendix A. Malaria application Evaluation Responses).

Five users filled out the questionnaire. The results are displayed in Table 7.

Question	**Number of users who gave the response**				
	1 (worst)	**2**	**3**	**4**	**5 (best)**
How easy was it to use?	0	0	0	3	1
How quick was it?	1	3	1	0	0
How good was it to use for biological insight?	0	0	2	2	1
How customisable was it?	0	0	1	3	0

Table 7: ***Summary of the Questionnaire Responses.***

Questionnaire responses for the Malaria application with the number of users who gave that response. Further results are shown in Appendix A. Malaria application Evaluation Responses

This highlights the issue of the speed of the application, the only question to receive a score of 1. This was also noted in the evaluation above. Considering this, a warning was added to the application to inform the user of the long waiting times.

The feedback allowed users to write anything they liked or disliked. Users noted the plots were nice and the application was easy to navigate. It was highlighted, however, there were some minor errors in the application. Namely, the DE scatterplot did not load, however, this was due to the wrong object being loaded and has since been corrected.

The informal questionnaire on the malaria dataset application also identified further work. It showed a need to change cluster names in the pseudotime results when the names are changed by the user. It also noted that it could be useful to provide more details when errors occur. These were not possible due to time constraints but could be of interest in the future.

4.4 Evaluation with Upload application using T Cell Dataset

This application loaded quicker compared to the T cell and Malaria applications where the objects are pre-processed. The Upload application took 27 seconds compared to the 1 minute 16 seconds required to load the libraries and files for the T cell and Malaria applications.

In testing the same five resolutions used in the T cell application pre-processed data, it was noted that the resolution 0.55 failed due to an error in the single-cell analysis to find the clusters (FindClusters). It is likely an error in the Seurat package used, however, future work must attempt to resolve this as it may pose issues in any analysis using the Upload application.

Therefore, only four resolutions were used to test the Upload application (0.15, 0.25, 0.35, 0.45). The data was uploaded using the same parameters as the existing application. The upload took 12 minutes, and the analysis took 8 hours 44 minutes. It mostly produced the same plots found in the T cell application (Figure 21). However, Figure 22 shows some variation in the PHATE plots of Slingshot and tradeSeq, likely due to the nature of the tools and their stability. Future work should also address this.

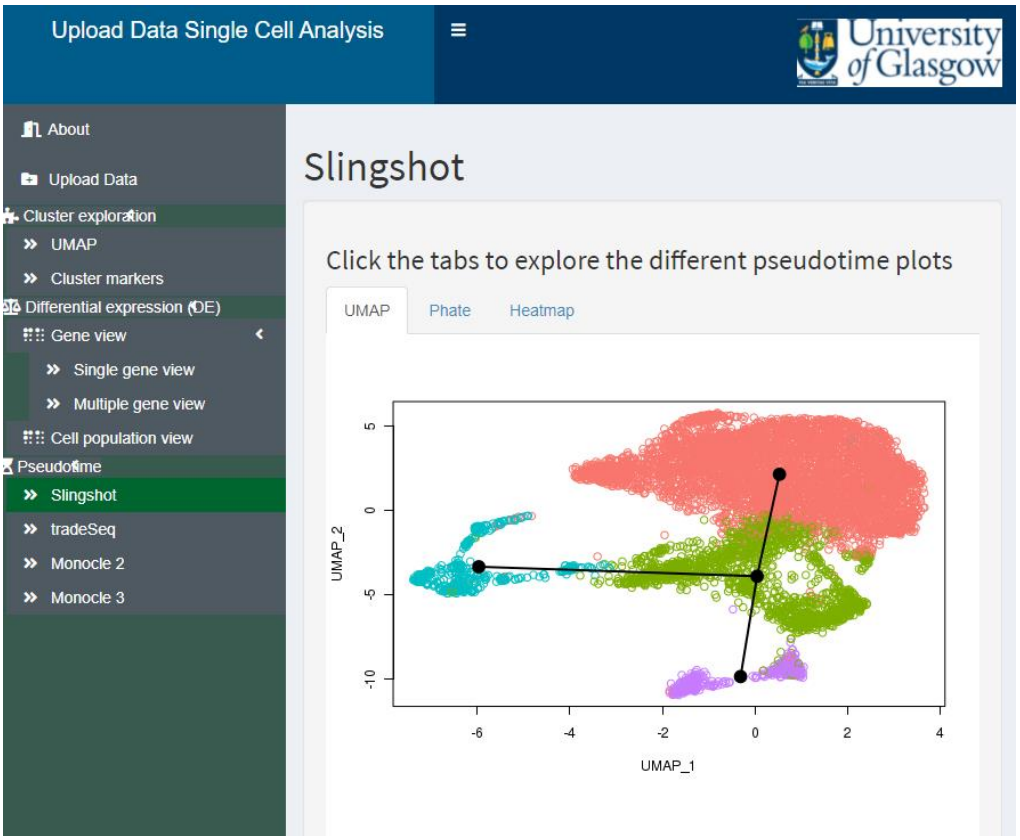

*Figure 21: **Upload Application Screenshot**. Screenshot of the Upload Application evaluated using the T cell dataset, resolution 0.15. Reflected the results from the pre-processed T cell application*

Figure 22: ***Upload Application Compared to T Cell Application Heatmap Genes Venn Diagrams****.Venn diagrams of the the genes identified in the heatmaps for each tool in T Cell compared to the same dataset in the Upload application. Vatiation likely due to the nature of the tools rather than an inherent issue. Produced using the package Venn in R*

A drawback to the Upload application is that the errors and output of code being run are not displayed in the application and are instead shown in the RStudio console. It was reasoned that as the application is run on the user's local machine, they will have access to the console and be able to see any output here. Future work could consider adding text output to the user interface so users can better see these details.

It should be noted that the application is not fully robust with no constraints over the parameters input nor the type of file uploaded. If the user encounters an error, there is no option to go back and re-do steps. Instead, they need to close the application or the error results in the exiting of the application. The user then needs to begin the application. Therefore, implementing effective error handling should be a clear next step to improve the application that was not possible here due to time restrictions.

5. Discussion and Conclusion

Many pseudotime tools are available and it can be difficult for users without programming expertise to find the tool best suited for the data. The applications produced here aim to address this. Four different tools are offered for the same dataset allowing users to trial them to find the optimal tool. Three separate applications were produced to be more favourable on the memory and speed up the application: Malaria, T cell and Upload. This was achieved using the existing Shiny atlas from the University of Glasgow with the addition of a pseudotime analysis section. The Upload application, unlike Malaria and T cell, did not use pre-processed data and instead allowed the user to produce the objects to be used in the application. The two main objectives were evaluated on these applications using the Malaria and T cell datasets.

5.1 Comparison to Other Applications

The features of the completed application were compared to other available single-cell applications. The application developed here allows the user to choose from four different pseudotime tools and view the results for each. The existing applications were limited to at most two tools and often used Monocle. Monocle, although a beneficial tool, hence being offered here, has often been found to be inadequate for pseudotime analysis. This application instead has offered three more tools alongside Monocle 2. The evaluation found that this was beneficial as different trajectories were identified in these different tools which would have been omitted if only Monocle 2 was offered.

5.2 Further work

Many possible future adaptations to the application have been mentioned previously, including changing cluster names, and providing an output in the application of code being run when uploading data and of any errors encountered. Furthermore, the Upload application is not robust and evaluation using the T cell dataset highlighted possible errors in the analysis which may be due to the instability of the tools. Future work must address this.

Further work may also aim to better quantify the differences of the tools to allow the users to compare them. This was not possible here due to time constraints and the differing natures of the tools.

The main concern of any further work would be to streamline the project. Little can be done to decrease the time taken to run the pipeline; however, the application may be sped up. For example, dynamically loading objects only when the user requires them could allow a faster start-up speed and prevent excessive use of storage space. This was shown in the 'Select starting cluster' section of Slingshot, where only the object at the relevant cluster is loaded. This does mean that if the user chooses the same cluster twice, it will still re-load the same object, which is inefficient and would need to be considered. Sparce matrices in the pre-processing may also reduce storage space and could be tested as a next step to this application.

5.3 Conclusion

Highlighted in the evaluation above, offering different tools and reductions for pseudotime analysis provides different trajectory inference and different results. Furthermore, the bias provided by different pseudotime tools can be accounted for by using multiple tools to support findings.

The ability to examine different pseudotime tools was lacking from the previous applications identified by this project. Few allowed more than one tool and a way to compare these tools. This is especially prudent here as not all tools used the same reduction method. The heatmaps allowed a better insight into these tools, with an understanding that the method of creating

the heatmaps for each tool varied and it is not directly comparable providing only a snapshot of the 40 genes.

Overall, the applications produced here provide biological insight into single-cell data. The three applications followed the same framework allowing for pseudotime analysis on existing atlases or the user's data. This will hopefully help make pseudotime analysis more accessible to lay users and allow them a novel way to view the results of multiple tools in one application.

References

1. Street K, Risso D, Fletcher RB, Das D, Ngai J, Yosef N, et al. (2018) Slingshot: cell lineage and pseudotime inference for single-cell transcriptomics. BMC Genomics.19(1):477, 10.1186/s12864-018-4772-0.
2. Van den Berge K, Roux de Bézieux H, Street K, Saelens W, Cannoodt R, Saeys Y, et al. (2020) Trajectory-based differential expression analysis for single-cell sequencing data. Nature Communications.11(1):1201, 10.1038/s41467-020-14766-3.
3. Zhu X, Wolfgruber TK, Tasato A, Arisdakessian C, Garmire DG, Garmire LX (2017) Granatum: a graphical single-cell RNA-Seq analysis pipeline for genomics scientists. Genome Medicine.9(1):108, 10.1186/s13073-017-0492-3.
4. Stuart T, Butler A, Hoffman P, Hafemeister C, Papalexi E, Mauck WM, 3rd, et al. (2019) Comprehensive Integration of Single-Cell Data. Cell.177(7):1888-902.e21, 10.1016/j.cell.2019.05.031.
5. Hentzschel F, Gibbins MP, Attipa C, Beraldi D, Moxon CA, Otto TD, et al. (2021) Host cell maturation modulates parasite invasion and sexual differentiation in Plasmodium. bioRxiv.2021.07.28.453984, 10.1101/2021.07.28.453984.
6. McIntyre CL, Monin L, Rop JC, Otto TD, Goodyear CS, Hayday AC, et al. (2020) β2 Integrins differentially regulate γδ T cell subset thymic development and peripheral maintenance. Proceedings of the National Academy of Sciences.117(36):22367-77, 10.1073/pnas.1921930117.
7. Trapnell C, Cacchiarelli D, Grimsby J, Pokharel P, Li S, Morse M, et al. (2014) The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. Nature Biotechnology.32(4):381-6, 10.1038/nbt.2859.
8. Saelens W, Cannoodt R, Todorov H, Saeys Y (2019) A comparison of single-cell trajectory inference methods. Nature Biotechnology.37(5):547-54, 10.1038/s41587-019-0071-9.
9. Campbell KR, Yau C (2018) Uncovering pseudotemporal trajectories with covariates from single cell and bulk expression data. Nature Communications.9(1):2442, 10.1038/s41467-018-04696-6.
10. Vento-Tormo R, Efremova M, Botting RA, Turco MY, Vento-Tormo M, Meyer KB, et al. (2018) Single-cell reconstruction of the early maternal–fetal interface in humans. Nature.563(7731):347-53, 10.1038/s41586-018-0698-6.
11. Qiu X, Mao Q, Tang Y, Wang L, Chawla R, Pliner HA, et al. (2017) Reversed graph embedding resolves complex single-cell trajectories. Nat Methods.14(10):979-82, 10.1038/nmeth.4402.
12. Cao J, Spielmann M, Qiu X, Huang X, Ibrahim DM, Hill AJ, et al. (2019) The single-cell transcriptional landscape of mammalian organogenesis. Nature.566(7745):496-502, 10.1038/s41586-019-0969-x.
13. Song D, Li JJ (2021) PseudotimeDE: inference of differential gene expression along cell pseudotime with well-calibrated p-values from single-cell RNA sequencing data. Genome Biology.22(1):124, 10.1186/s13059-021-02341-y.
14. Winston Chang JC, JJ Allaire, Carson Sievert, Barret Schloerke, Yihui Xie, Jeff Allen, Jonathan McPherson, Alan Dipert, Barbara Borges. shiny: Web Application Framework for R. R package version 1.6.0. 2021.
15. Innes BT, Bader GD (2018) scClustViz - Single-cell RNAseq cluster assessment and visualization. F1000Res.7:ISCB Comm J-1522, 10.12688/f1000research.16198.2.
16. Jagla B, Rouilly V, Puceat M, Hasan M (2020) SCHNAPPs - Single Cell sHiNy APPlication(s). bioRxiv.2020.06.07.127274, 10.1101/2020.06.07.127274.
17. Kee N, Volakakis N, Kirkeby A, Dahl L, Storvall H, Nolbrant S, et al. (2017) Single-Cell Analysis Reveals a Close Relationship between Differentiating Dopamine and Subthalamic Nucleus Neuronal Lineages. Cell Stem Cell.20(1):29-40, 10.1016/j.stem.2016.10.003.

18. Kageyama J, Wollny D, Treutlein B, Camp JG (2018) ShinyCortex: Exploring Single-Cell Transcriptome Data From the Developing Human Cortex. Frontiers in Neuroscience.12(315), 10.3389/fnins.2018.00315.
19. Ji Z, Ji H (2016) TSCAN: Pseudo-time reconstruction and evaluation in single-cell RNA-seq analysis. Nucleic Acids Research.44(13):e117-e, 10.1093/nar/gkw430.
20. Ji Z, Zhou W, Ji H (2017) Single-cell regulome data analysis by SCRAT. Bioinformatics (Oxford, England).33(18):2930-2, 10.1093/bioinformatics/btx315.