1 Introductions

Single-cell high-throughput transcriptomics is a powerful approach to investigate the heterogeneity of gene expression activities on cell level, which is otherwise hard to detect by bulk transcriptomics experiments. Many of the single-cell transcriptomics researches focus on the differentiation of various cell types and hope to find how the expression of genes and key regulatory factors changes over the differentiation process [1,2]. For individual cells during the differentiation process, each cell may experience quite varying schedule of transcriptional changes even though the overall trajectory of changes is highly identical. To depict such variation between individual cells, it is essential to reconstruct the differentiation lineage and order each single cell to the correct cellular states on the lineage.

Several computational methods specifically tailored for single-cell experiments have been proposed to reconstruct the trend of cellular transitions during differentiation. For example, SPADE [3] uses an unsupervised spanning-tree progression approach to organize cells in a hierarchy of related phenotypes when applied
to flow cytometry data. Such approach can reveal complex cellular hierarchies of
differentiation and rare cell states. Similarly, Monocle [4] also takes advantages
of unsupervised minimum spanning tree to order whole-transcriptome profiles
of single cells along 'pseudotime', a hypothesized time course which can quanti-
tatively measure the real biological process of differentiation. This pseudotime
course is then used to study how gene expressions change over the differentiation
process.
If we assume the validity of the pseudotemporal ordering approach, several prob-
lems still remain to be solved. First, it is obvious that there could be numerous
ways to construct such pseudotime course for a given single-cell transcriptome
profile. Thus how to evaluate these pseudotime courses in an unbiased way is a
critical issue before any follow-up analysis. Second, a completely unsupervised
learning strategy (e.g. without using any prior biological information of certain
marker genes) may lead to quite unreliable results. In Monocle’s case, since
the MST (minimum spanning tree) algorithm cannot automatically designate
a starting or ending point of the tree, it is possible that the MST will falsely
switch the starting and ending point of the true biological process and lead to
the opposite results. As reliable prior biological information do exist for most
biological system in which single-cell experiments are conducted, we hypothesize
that these information could be incorporated into constructing the pseudotime
course to generate more reliable results.
To address the first problem, we propose pseudotemporal ordering score (POS),
a quantitative measure of the reliability of numerous possible pseudotime course.
We also propose a new pseudotime course construction method, travelling sales-
man problem algorithm (TSP), which aims to find a pseudotime course to min-
imize the total distance of linking the transcriptome profiles of all single cells.
Using POS, we argue that TSP is more reliable than the MST algorithm in
several testing datasets.
A new software, TSCAN, is proposed to facilitate the construction of pseudotem-
poral cell ordering using TSP algorithm on single-cell RNA-seq data. TSCAN
first uses principal component analysis to perform dimension reduction on the
preprocessed gene expression data. Then the travelling salesman problem (TSP)
algorithm is applied to construct a pseudotemporal ordering of cells so that the
total distance along the path is nearly the shortest one. To address the problem
that TSP does not have a designated starting/ending point, users can provide
prior biological information of gene expression to select the optimal starting
point of the ordering. After the pseudotemporal ordering and pseudotime are
constructed, TSCAN can search for differentially expressed genes.
TSCAN comes with a user-friendly GUI written in shiny which provides pow-
ervful and convenient functions to tune pseudotemporal ordering. For example,
users can trim unwanted cells or branches after an initial construction of pseu-
dotime ordering. Users can use multiple marker genes to tune the starting point
of the TSP path. Users can also easily calculate POS for different cell orderings
using TSCAN GUI by uploading cell ordering and sub-population information.
Many new features can be added to TSCAN in the future. Most importantly,
TSCAN will be able to analyze single-cell experiments data other than gene ex-
pression data and construct pseudotemporal ordering correspondingly. TSCAN will also be able to automatically determine the change points of the differentially expressed genes and order the genes according to these change points. Genes with similar change points and change patterns may provide interesting biological findings about the biological processes in the single-cell experiments. Now we use the single-cell RNA-seq data for BMDC cells before and after 6h of LPS stimulation [5] to demonstrate how TSCAN can be used to construct pseudotemporal cell ordering using TSP.

2 Preprocess Gene Expression Profiles

The function preprocess can be used to filter out genes which are not expressed in most cells or not differentially expressed.

```r
library(TSCAN)
data(lpsdata)
procdata <- preprocess(lpsdata)
```

By default, preprocess will first take log2 on the original dataset after adding a pseudocount of 1. Then the function will rule out genes which have expression values of less than 1 in at least half of all cells. Genes whose coefficient of covariance of the expression values is less than 1 are also filtered out.

3 Constructing Pseudotime

The function exprmclust will perform dimension reduction using principal component analysis and model-based clustering.

```r
lpsmclust <- exprmclust(procdata)
```

Use plotmclust function to visualize the clustering based on the data after dimension reduction.
Use the TSCANorder function to obtain the TSCAN ordering.

```r
lpsorder <- TSCANorder(lpsmclust)
lpsorder
```

```
## [1] "Unstimulated_S27" "Unstimulated_S38" "Unstimulated_S60"
## [4] "Unstimulated_S46" "Unstimulated_S49" "Unstimulated_S39"
## [7] "Unstimulated_S28" "Unstimulated_S82" "Unstimulated_S61"
## [10] "Unstimulated_S78" "Unstimulated_S66" "Unstimulated_S93"
## [13] "Unstimulated_S51" "Unstimulated_S63" "Unstimulated_S3"
## [16] "Unstimulated_S2" "Unstimulated_S15" "Unstimulated_S52"
## [19] "Unstimulated_S34" "Unstimulated_S71" "Unstimulated_S26"
## [22] "Unstimulated_S80" "Unstimulated_S6" "Unstimulated_S70"
## [25] "Unstimulated_S24" "Unstimulated_S48" "Unstimulated_S75"
## [28] "Unstimulated_S17" "Unstimulated_S54" "Unstimulated_S76"
```
## 4 Testing Differentially Expressed Genes

Use the `difftest` function to detect differentially expressed genes given a constructed pseudotemporal ordering. The function will compare fitting a generalized additive model (gam) and fitting a constant. If gam fits significantly better than a constant fit than the gene is supposed to be differentially expressed. Original p-values as well as q-values after adjusting for multiple testing will be the output.
diffval <- difftest(procdata,lpsorder)

#Selected differentially expressed genes under qvalue cutoff of 0.05
head(row.names(diffval)[diffval$qval < 0.05])

## [1] "STX6" "MRPL28" "CUTA" "AI413582" "SNRPC" "MTCH1"

Use singlegeneplot function to plot the expression value of a single gene against
a given pseudotime. Notice that here orderonly should be set to FALSE.

STAT2expr <- log2(lpsdata["STAT2",]+1)
singlegeneplot(STAT2expr, TSCANorder(lpsmclust,flip=TRUE,orderonly=FALSE))
Comparing Different Pseudotemporal Ordering

Given sub-population information of the single-cell information, function orderscore can calculate the pseudotemporal ordering score (POS) of multiple pseudotemporal ordering.

First prepare the ordering information. We want to compare TSP ordering with or without marker gene information.

```r
subpopulation <- data.frame(cell = colnames(procdata), sub = ifelse(grepl("Unstimulated",colnames(procdata)),0,1), stringsAsFactors = FALSE)
#Comparing ordering with or without marker gene information
order1 <- TSCANorder(lpsmclust)
order2 <- TSCANorder(lpsmclust, c(1,2,3))
orders <- list(order1,order2)

Run orderscore function to get the comparison result:

```r
orderscore(subpopulation, orders)
```

## 

|   | 0.8786155 | 0.8024513 |

From the previous example we can see that when prior information of MYOG gene expression is available, POS will dramatically increase compared to the ordering without MYOG gene expression information.

Reference