STA426 treekoR project

Anouk Petitpierre and Tudor Jumuga

Version January 4, 2022

Contents

Chapter 1

Introduction

The development of high-throughput single-cell technologies provide bioinformaticians with a large amount of high-dimensional cytometry data. High-dimensional data are data with more features than samples. In the case of cytometry data, which are data from cells and their properties, this means that more cells are measured than patients, from which the cells were taken.

Usually, these data are analysed using either Manual Gating or Hierarchical Clustering. Manual Gating clusters cells into discrete populations based on shared marker expression, Hierarchical Clustering stratifies cell subsets without a predetermined hypothesis.

Both methods have their advantages, but unfortunately, they have their disadvantages, too: Manual Gating is time-consuming and potentially biased, as the markers chosen for the hierarchical clustering are based on expert opinion. Hierarchical clustering is more time-efficient, but it dismisses cell hierarchy. treekoR is a novel framework developed to solve these disadvantages. It uses an automated hierarchical clustering algorithm to make the analysis more time-efficient and neutral, while still taking into account cell hierarchy. The treekoR framework consists of five steps:

1. Clustering

The data are clustered using an existing clustering algorithm. In the paper, flowSom was used.

2. Hierarchy

The clusters are put into a hierarchical tree. Different hierarchical clustering algorithms can be used within the treekoR package. The paper itself preferes hopach, as it allows for multiple children at each node.

3. Proportions

For each node in the hierarchical tree, the proportion of that node's expression and its parent node's expression (%parent) as well as the proportion of that node's expression and all the nodes' expression (%total) are calculated.

4. Significance Testing

For each node, a t-test is used to test for a significant difference between the proportions of the diseased and the healthy samples. This can be done for both the %parent and the %total.

5. Visualisation

treekoR plots a heatmap, which shows the statistical test result for each cluster and each marker. It also provides a scatterplot with the T-scores of the two proportions plotted against each other.

In this project, we want to reproduce a treekoR analysis on the raw Age Chronic data from Shen-Orr *et al.* (2016). Additionally, we want to use prepped datasets from the authors of treekoR to reproduce their Benchmarking of treekoR: They used twelve datasets from eleven papers and run treekoR on them, once computing $\%parent$, once computing $\%total$ and using "average linkage" as the hierarchical clustering algorithm, and once computing %total and using "hopach" as the hierarchical clustering algorithm. For all three results, they computed the balanced accuracy and provided a boxplot of the balanced accuracy of all twelve datasets. Apart from reproducing this analysis, we will also use a different performance evaluation measure for our model, namely X.

Chapter 2

Methods

For our project, we used R version 4.1.2 and BiocManager version 4.13.

To use the treekoR package from BiocManager on data, the data must first be converted to a Single Cell Experiment object. According to the treekoR paper's specifications, the count data were transformed using an arcsinh transformation with cofactor 5. The data were then clustered using the FlowSOM-based function cluster() from the CATALYST package. FlowSOM uses a self-organising map in order to analyse cytometry data. Its goal is to prevent the potential loss in subset detection that comes with the increasing number of markers measured in cytometry. To this end, it uses two-level clustering and star charts.

2.1 part I: reproduction of treekoR analysis on Age Chronic data set

First, the flowSOM files (fcs) were read in and transformed to a Single Cell Experiment object (sce) in order to make them available for the treekoR functions. Metadata regarding the age of the individuals ('Old' or 'Young' based on a cutoff of 40 years) were appended manually. Next, the data were arcsinh-transformed with a cofactor 5, and clustered using the flowSom-based function cluster() built into the CATALYST package. Cell hierarchy was achieved by using treekoR with both the "HOPACH" and "average linkage" algorithms.

"HOPACH" works by recursively partitioning a data set with the PAM algorithm, short for Partitioning Around Medioids, while ordering and possibly collapsing clusters at each level. "PAM" works by calculating the so-called medioids through a dissimilarity matrix, where dissimilarities can be either Euclidean distance or based on correlation of the individual elements. The product is a hierarchically structured tree of nodes. Different from other hierarchical clustering algorithms, "HOPACH" allows for more than two children at every node. This becomes visible in the visualisation of the treekoR analysis.

"Average linkage" on the other hand works by calculating the average distance between each pair of observations between clusters.

2.2 part II: Benchmarking

For the benchmarking step we used the datasets provided by the authors of treekoR.

To do the benchmarking, the authors computed probabilities for each cell to belong to either one of the conditions, for example diseased or healthy (prob.pos and prob.neg). Both were obtained using machine learning tools from the package mlr3 to train and predict the binary clinical outcome using %total based on either HOPACH or average linkage and %parent. On top of prob.pos and prob.neg, the authors provided a coloumn called truth, which showed the true condition of each single cell, as well as a coloumn called response, which showed the condition predicted by their model. These values were used by the authors to compute balanced accuracy, which is the weighted average of specificity and sensitivity.

For our Benchmarking, we used a different performance evaluation measure: the area under the receiver operating characteristics (ROC) curve (AUC). ROC curves plot the true positive rate (TPR) / Specificity against the false positive rate (FPR) / 1 - Sensitivity against each other to show a model's evaluation performance compared to random evaluation.

Chapter 3

Results

part I: reproduction of treekoR analysis - provide hierarchical trees - provide treekoR heatmap plots in the end - report significant findings / observations

part II: Benchmarking

```
## Loading required package:  survival
```

```
## Warning in ncases * ncontrols:  NAs produced by integer overflow
```

```
## Warning in ncases * ncontrols:  NAs produced by integer overflow
## Error in xy.coords(x, y, xlabel, ylabel, log):  ’x’ is a list, but does not
have components ’x’ and ’y’
```

titleROC curves.

- provide Benchmarking plots, once using balanced accuracy as a measure of comparison, once using our "own" - report findings / observations

Chapter 4

Discussion

part I: - why does our analysis look different from theirs? - what has remained unclear? - what could be explored further?

part II: - why does our analysis look different than theirs? - are there differences when using other measure instead of balanced accuracy? why? what does that mean? - what remains unclear? - what could be explored further?

Appendix A

Appendix

- code - where the data can be found

References

Shen-Orr, S. S., Furman, D., Kidd, B. A., Hadad, F., Lovelace, P., Huang, Y. W., Rosenberg-Hasson, Y., Mackey, S., Grisar, F. A., Pickman, Y., Maecker, H. T., hsiu Chien, Y., Dekker, C. L., Wu, J. C., Butte, A. J., and Davis, M. M. (2016). Defective signaling in the jak-stat pathway tracks with chronic inflammation and cardiovascular risk in aging humans. *Cell Systems*, **3**, 374–384.e4. 6