

Analysis of flow cytometry data with R

Training for life scientists

João Lourenço, Tania Wyss & Nadine Fournier

Translational Data Science – Facility

SIB Swiss Institute of Bioinformatics

Outline

Day 5

01

Presentation of the workflow

02

Normalization with flowStats

03

Clustering with the PhenoGraph algorithm

04

Diffusion maps for dimensionality reduction

05

Trajectory/pseudotime analysis with slingshot

01

Presentation of the workflow

“Source” of the presented workflow

PREPROCESSING			
Compensation	Export population	Transformation	Normalization
<p>Marker Y</p> <p>Marker X</p>	<p>Marker Y</p> <p>Marker X</p>	<p>Sample</p> <p>Marker X</p> <p>Marker Y</p>	<p>Marker X</p> <p>Marker Y</p> <p>A</p> <p>B</p> <p>C</p>
Standard processing gating software		<p>Manual arcsinh, FlowVS, FlowCore</p>	FlowStats


DIMENSIONALITY REDUCTION	CLUSTERING	PSEUDOTIME
Visualization	Grouping phenotypically similar cells	Trajectory analysis
<p>HSNE2</p> <p>HSNE1</p> <p>DC2</p> <p>DC1</p> <p>Marker X</p>	<p>HSNE2</p> <p>HSNE1</p> <p>DC2</p> <p>DC1</p>	<p>DC2</p> <p>DC1</p> <p>Pseudotime</p>
HSNE, Diffusion map, tSNE, UMAP, PCA	Gaussian mean shift, PhenoGraph, FlowSom	Slingshot

“Source” of the presented workflow

PREPROCESSING			
Compensation	Export population	Transformation	Normalization
<p>Marker Y</p> <p>Marker X</p>	<p>Marker Y</p> <p>Marker X</p>	<p>Sample</p> <p>Marker X</p> <p>Marker Y</p>	<p>A</p> <p>B</p> <p>C</p> <p>Marker X</p> <p>Marker Y</p>
Standard processing gating software		R Manual arcsinh, FlowVS, FlowCore	FlowStats

DIMENSIONALITY REDUCTION	CLUSTERING	PSEUDOTIME
Visualization	Grouping phenotypically similar cells	Trajectory analysis
<p>HSNE2</p> <p>HSNE1</p> <p>DC2</p> <p>DC1</p> <p>Marker X</p>	<p>HSNE2</p> <p>HSNE1</p> <p>DC2</p> <p>DC1</p>	<p>DC2</p> <p>DC1</p> <p>Pseudotime</p>
HSNE, Diffusion map, tSNE, UMAP, PCA	Gaussian mean shift, PhenoGraph, FlowSom	Slingshot

Some interesting points

- Importance of transformation optimization
 - Combining R with other software, eg HSNE and GMS clustering : export flowSet as fcs files with flowCore
- > `write.flowSet(x=flowSet, outdir="output_dir", filename, ...)`
- identifier of individual flowFrame objects within flowSet,
with fcs extension by default, i.e. `sampleNames(flowSet)` 
- The biological conclusions may depend on the tools/methods used:
 - quadrant gating vs GMS clustering of CD4⁺ T cells.

Let's setup the workflow

- Open the day 5 assignment in Posit

or

- Download and unzip the data from <https://taniawyss.github.io/flow-cytometry-analysis-with-R/flowCyt/material/#day-5>
- Obtain rmd from https://taniawyss.github.io/flow-cytometry-analysis-with-R/flowCyt/day5/exercises_d5/

Install packages:

```
> BiocManager::install("flowStats")  
> BiocManager::install("destiny")  
> BiocManager::install("slingshot")  
> devtools::install_github("JinmiaoChenLab/cytofkit2", dependencies=TRUE)
```

Load packages in « libraries » chunk



Normalization with flowStats

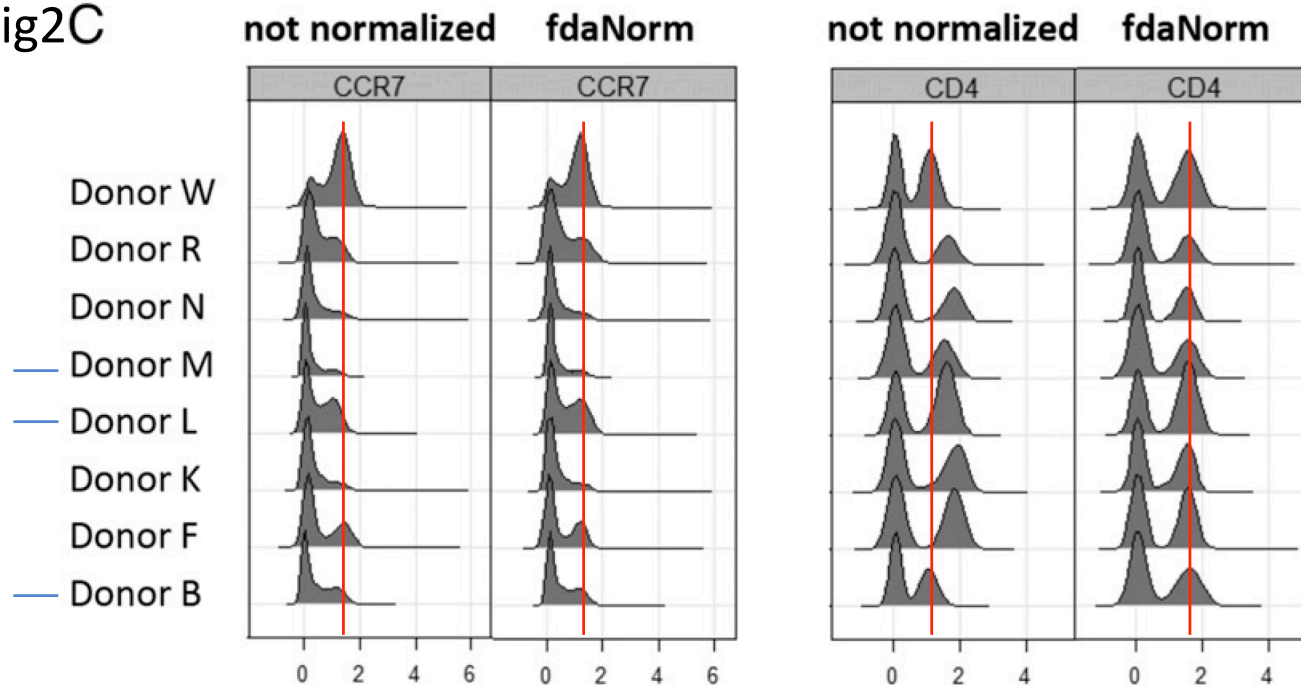
flowStats

<https://www.bioconductor.org/packages/release/bioc/html/flowStats.html>

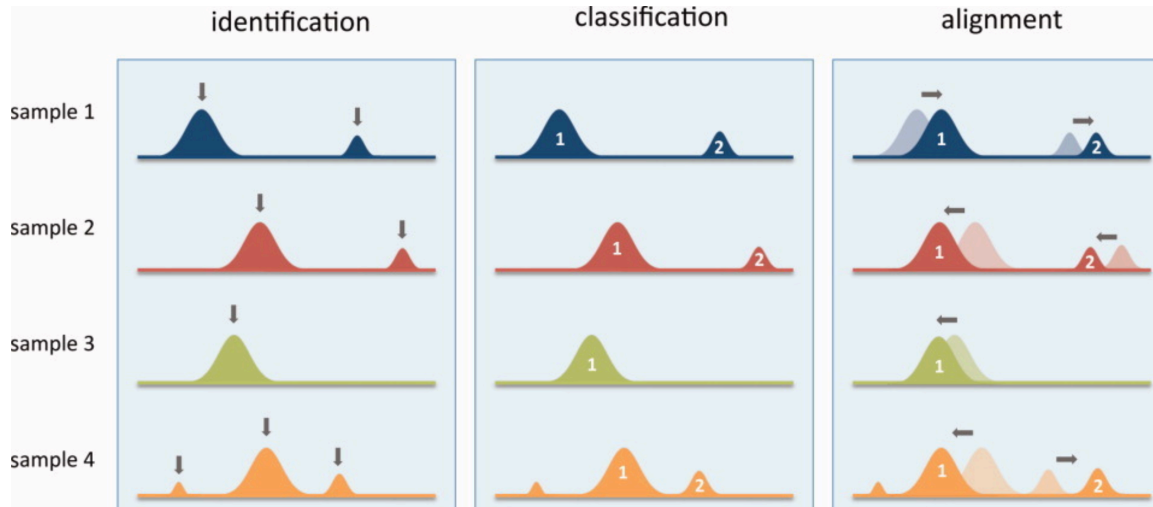
<https://onlinelibrary.wiley.com/doi/10.1002/cyto.a.20823>

Methods and functionality to analyze flow data that are beyond the basic infrastructure provided by the flowCore package.

Fig2C



flowStats (≠ CytoNorm !)



```
> fs_normfda <-  
warpSet(fs_transf,  
stains=c("CD8","CD27"))
```

Select the markers which require normalization.

- High density areas represent particular sub-types of cells.
- Markers are binary. Cells are either positive or negative for a particular marker.
- Peaks should align if the above statements are true.

The algorithm in warpSet performs the following steps:

1. Identify landmarks for each parameter
2. Estimate the most likely total number (k) of landmarks
3. Perform k-means clustering to classify landmarks
4. Estimate functions for each sample and parameter that best align the landmarks, given the underlying data. This step uses functionality from the fda package.
5. Transform the data using the estimated functions

03

PhenoGraph algorithm

PhenoGraph algorithm

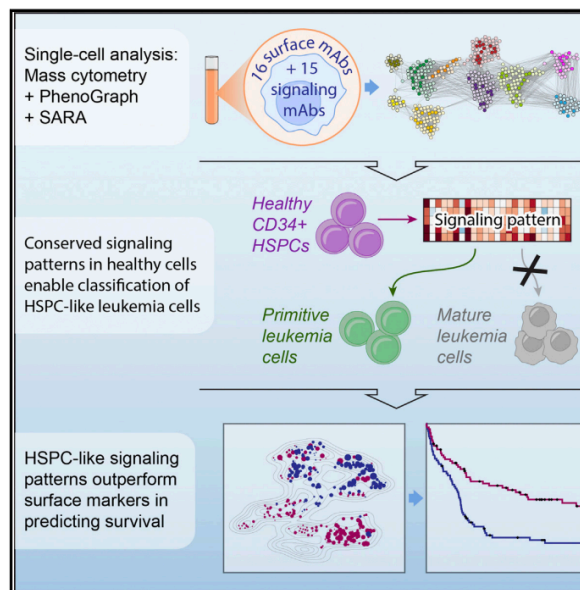
Clustering method designed for high-dimensional single-cell data analysis

Cell

Resource

Data-Driven Phenotypic Dissection of AML Reveals Progenitor-like Cells that Correlate with Prognosis

Graphical Abstract



Authors

Jacob H. Levine, Erin F. Simonds, Sean C. Bendall, ..., James R. Downing, Dana Pe'er, Garry P. Nolan

Correspondence

dpeer@biology.columbia.edu (D.P.), gnolan@stanford.edu (G.P.N.)

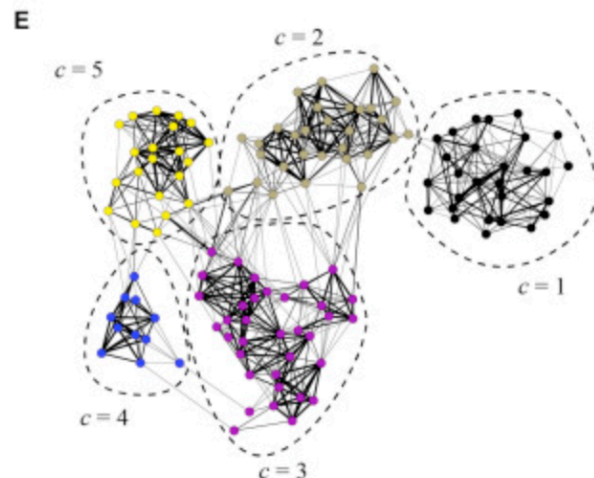
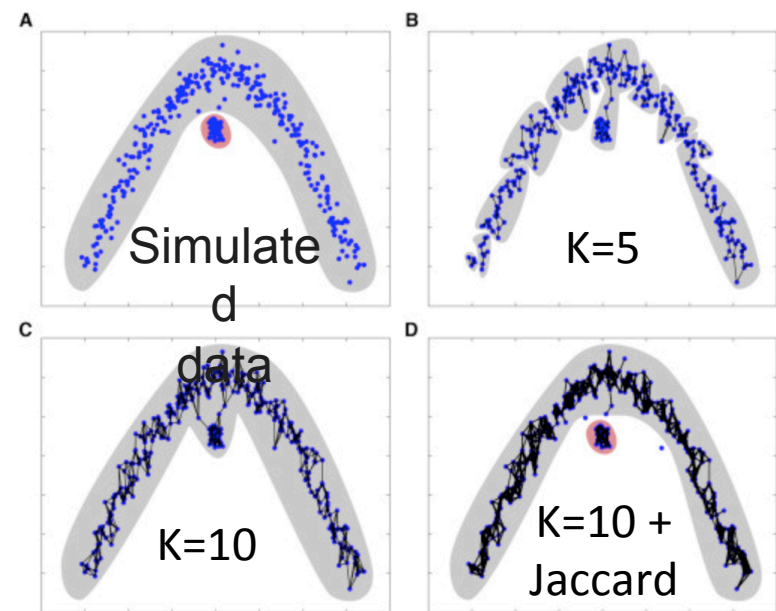
In Brief

The PhenoGraph algorithm robustly partitions high-parameter single-cell data into phenotypically distinct subpopulations, aiding the study of complex tissues and disease cohorts. Applying PhenoGraph to a pediatric acute myeloid leukemia dataset revealed a recurrent population of leukemic cells with variable cell surface markers, but consistent signaling dynamics that mimicked normal hematopoietic progenitors.

<https://pubmed.ncbi.nlm.nih.gov/26095251/>

PhenoGraph algorithm

- *k*-nearest neighbor graph based on euclidean distances in PCA space
- Each cell is represented by a node and connected by a set of edges to a neighborhood of its most similar cells
- Edge weights between cells are refined based on the shared overlap in their local neighborhoods (*Jaccard coefficient*)

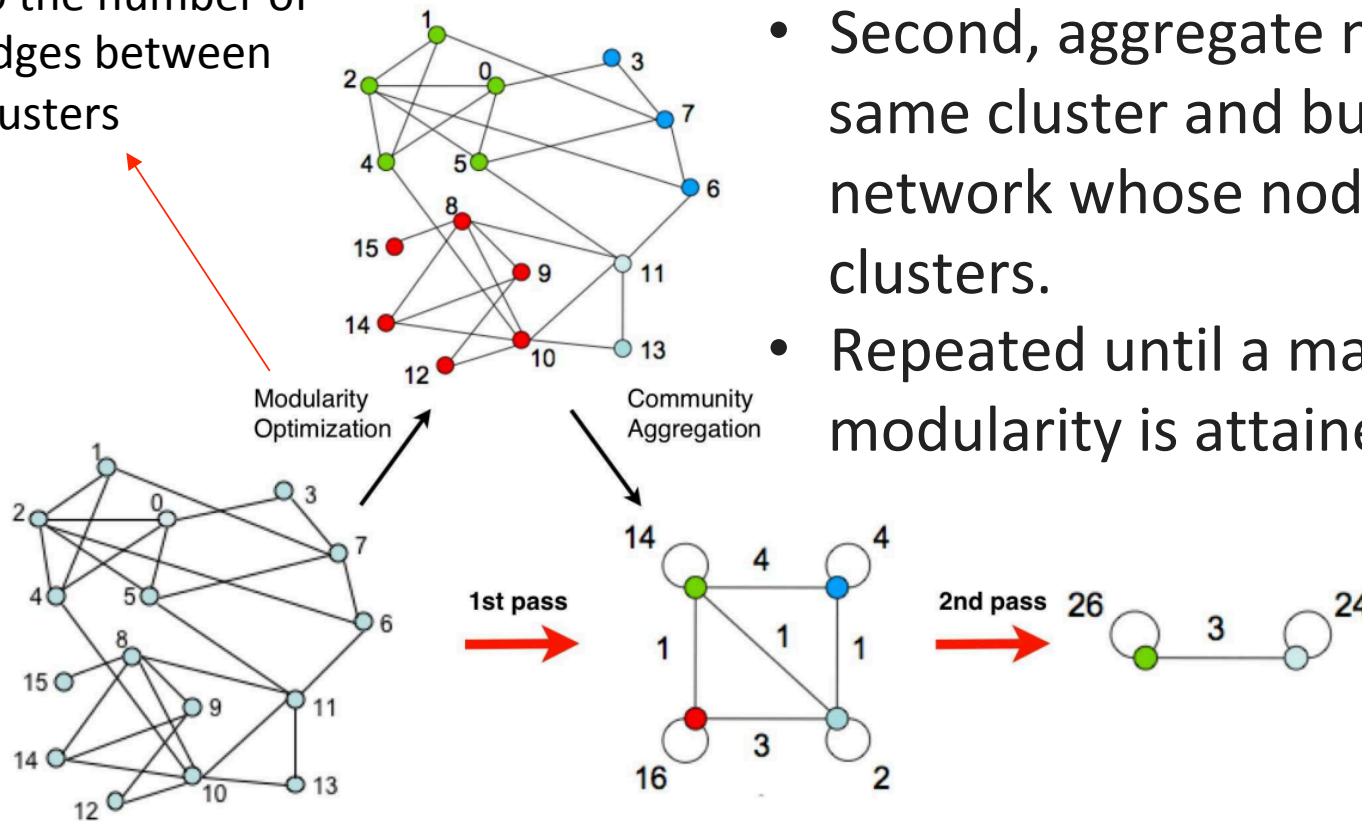


- Cluster cells by optimizing for modularity (*Louvain algorithm*)

Louvain method

Maximize the number of edges within clusters compared to the number of edges between clusters

- First, look for "small" clusters by optimizing *modularity* locally
- Second, aggregate nodes of the same cluster and builds a new network whose nodes are the clusters.
- Repeated until a maximum of modularity is attained



PhenoGraph algorithm

- An unsupervised approach to clustering: there is no assumption about the size, number, or form of the clusters
- Like other unsupervised methods, it is suitable for less predictable or under-studied tissues such as cancer, where new phenotypes can occur
- Outperforms other methods in terms of computation time, which allows to analyse datasets of unprecedented size

PhenoGraph algorithm

The PhenoGraph clustering is implemented in the *cytofkit2* package (<https://github.com/JinmiaoChenLab/cytofkit2>)

```
> library(cytofkit2)
```

```
> phenograph <- Rphenograph(df, k=50)
```

Expression data to be analysed

“Resolution”. Number of nearest neighbours, default is 30. Lower to get more clusters (smaller ones) and higher to get fewer clusters (bigger ones)

04

Diffusion maps

Diffusion maps

Implemented in the R package *destiny* (<https://bioconductor.org/packages/release/bioc/html/destiny.html>)

Bioinformatics, 32(8), 2016, 1241–1243

doi: 10.1093/bioinformatics/btv715

Advance Access Publication Date: 14 December 2015

Applications Note

OXFORD

Gene expression

***destiny*: diffusion maps for large-scale single-cell data in R**

Philipp Angerer¹, Laleh Haghverdi¹, Maren Büttner¹, Fabian J. Theis^{1,2}, Carsten Marr^{1,*} and Florian Buettner^{1,†,*}

¹Helmholtz Zentrum München - German Research Center for Environmental Health, Institute of Computational Biology, Ingolstädter Landstr. 1, 85764 Neuherberg, Germany and ²Technische Universität München, Center for Mathematics, Chair of Mathematical Modeling of Biological Systems, Boltzmannstr. 3, 85748 Garching, Germany

*To whom correspondence should be addressed.

†Present address: European Molecular Biology Laboratory, European Bioinformatics Institute (EMBL-EBI), Wellcome Trust Genome Campus, Cambridge, CB10 1SD, UK.

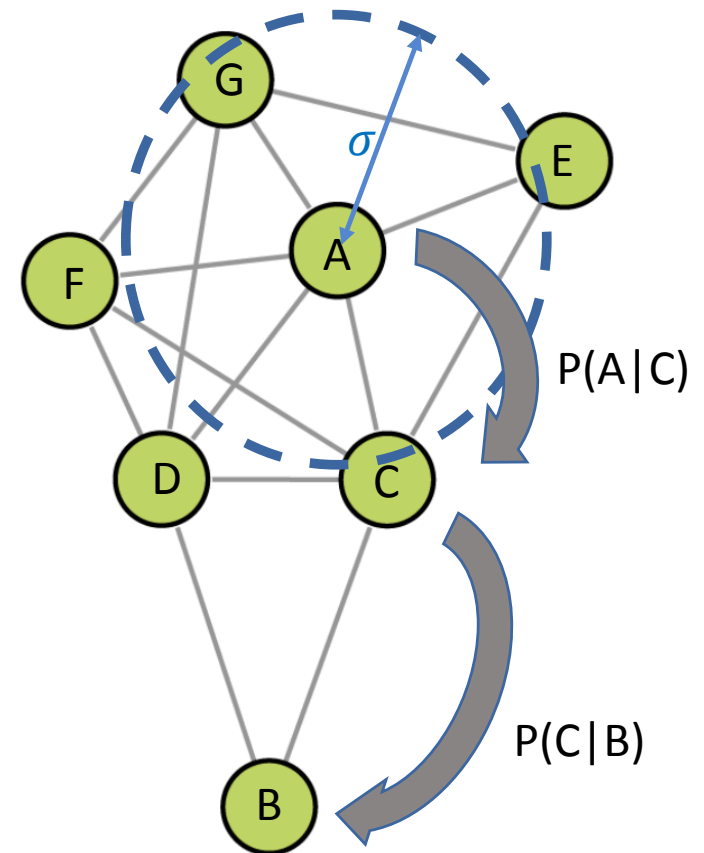
Associate Editor: Ziv Bar-Joseph

Received on 5 August 2015; revised on 28 October 2015; accepted on 1 December 2015

<https://pubmed.ncbi.nlm.nih.gov/26668002/>

Diffusion maps

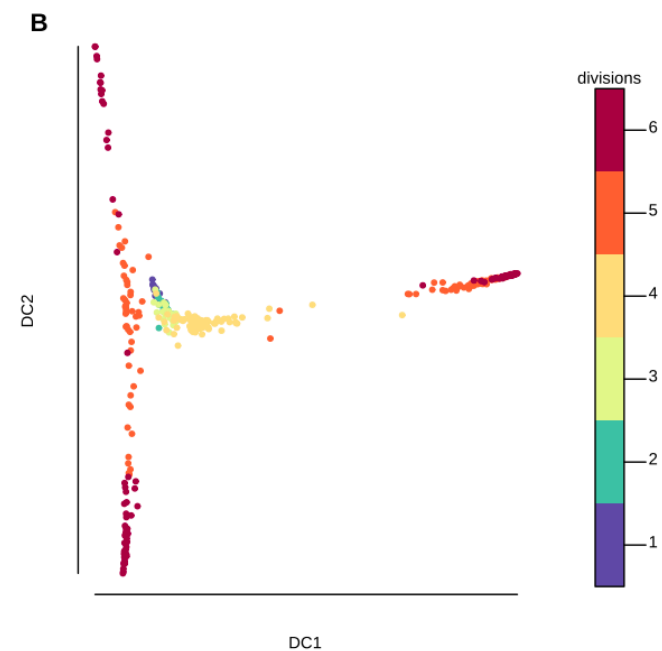
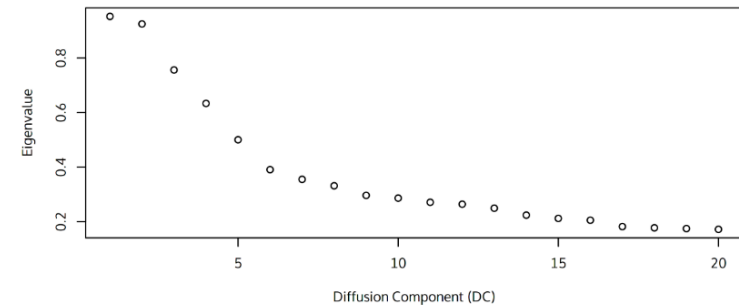
- Non-linear dimensionality reduction algorithm
- Based on a network of cells (nodes), in which phenotypically similar cells are connected
- The distance between two cells is defined by the probability of going from one to the other in K steps (*transition probabilities*)
- Estimation heuristic to derive the parameters (σ) of the *Gaussian kernel*.



$$P(A|B) \text{ in 2 steps} = P(A|C) \times P(C|B) + P(A|D) \times P(B|B)$$

Diffusion maps

- Matrix of transition probabilities between cells
- Dimensionality reduction is done by eigenvalue decomposition (like in PCA)
- Principal diffusion components (like in PCA)



Angerer et al., Bioinformatics 2015

Diffusion maps

- UMAP & tSNE:
 - Best represent the structure of the data
 - Separate cells into different clusters
- Diffusion maps:
 - Best represent the connections in the data
 - Place cells (clusters) into the trajectories through intermediate states
 - Especially suited for analysing single-cell data from differentiation experiments

Diffusion maps

```
> library(density)
> dm <- DiffusionMap(df,
  k=1000,
  suppress_dpt = TRUE,
  verbose=TRUE)
```

DiffusionMap
object

Expression data to be analyzed

The input parameter k controls the number of nearest neighbours for each cell to be considered.

Guideline for k is a small enough number to make the computation cost limited, but not too small to alter the connectivity of data as a graph, which would result in a noisy embedding.

A typical k is between 200 and 1000 cells.

To perform (**FALSE**) or not (**TRUE**)
pseudotime ordering and assigns cell to branches (
<https://bioconductor.org/packages/release/bioc/vignettes/destiny/inst/doc/DPT.html>)

Without downsampling,
this step can take hours !

05

Slingshot: trajectory / pseudotime analysis

Slingshot: trajectory / pseudotime analysis

- Method for inferring cell lineages and pseudotimes from single-cell gene expression data
- Designed for multiple branching lineages
- *Pseudotime*: one-dimensional variable representing each cell's transcriptional progression toward the terminal state

Slingshot: trajectory / pseudotime analysis

Implemented in the R package *slingshot* (
<https://bioconductor.org/packages/release/bioc/html/slingshot.html>)

Street et al. *BMC Genomics* (2018) 19:477
<https://doi.org/10.1186/s12864-018-4772-0>

BMC Genomics

METHODOLOGY ARTICLE

Open Access



Slingshot: cell lineage and pseudotime inference for single-cell transcriptomics

Kelly Street^{1,8}, Davide Risso², Russell B. Fletcher³, Diya Das^{3,9}, John Ngai^{3,6,7}, Nir Yosef^{4,8}, Elizabeth Purdom^{5,8} and Sandrine Dudoit^{1,5,8,9*} 

Abstract

Background: Single-cell transcriptomics allows researchers to investigate complex communities of heterogeneous cells. It can be applied to stem cells and their descendants in order to chart the progression from multipotent progenitors to fully differentiated cells. While a variety of statistical and computational methods have been proposed for inferring cell lineages, the problem of accurately characterizing multiple branching lineages remains difficult to solve.

Results: We introduce Slingshot, a novel method for inferring cell lineages and pseudotimes from single-cell gene expression data. In previously published datasets, Slingshot correctly identifies the biological signal for one to three branching trajectories. Additionally, our simulation study shows that Slingshot infers more accurate pseudotimes than other leading methods.

Conclusions: Slingshot is a uniquely robust and flexible tool which combines the highly stable techniques necessary for noisy single-cell data with the ability to identify multiple trajectories. Accurate lineage inference is a critical step in the identification of dynamic temporal gene expression.

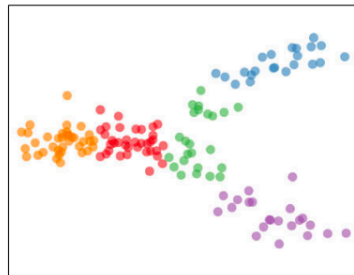
Keywords: RNA-Seq, Single cell, Lineage inference, Pseudotime inference

<https://pubmed.ncbi.nlm.nih.gov/29914354/>

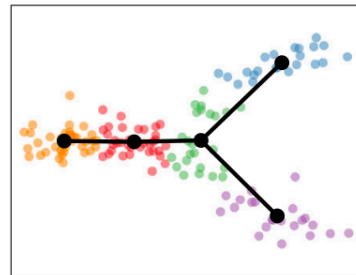
Slingshot: trajectory / pseudotime analysis

Two main stages:

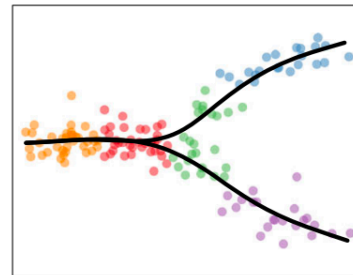
- the inference of the global lineage structure



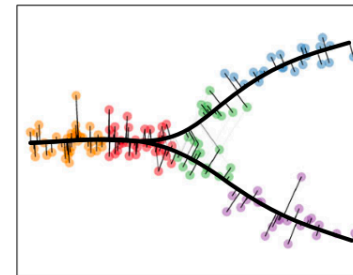
set of clusters
(& DR embedding)



Cluster-based
Minimum
Spanning Tree
(MST)



Simultaneous
principal curves
(smooth
representations of
lineages)



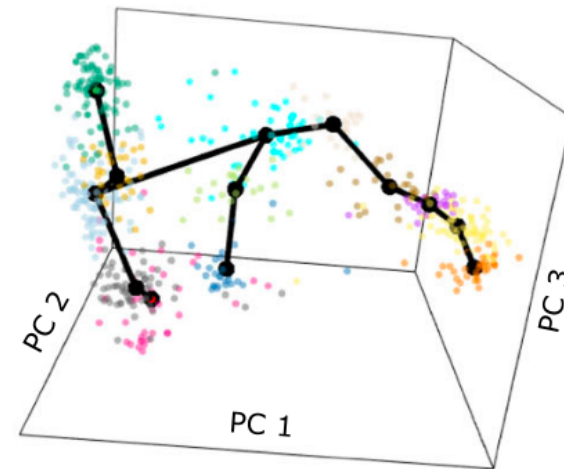
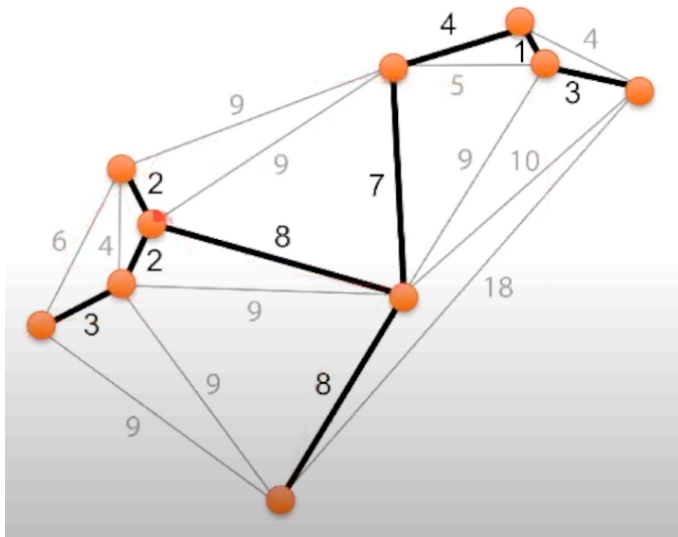
Pseudotime values
are obtained by
orthogonal
projection onto the
curves



- the inference of pseudotime variables for cells along each lineage

Minimum spanning tree

Slingshot treats clusters of cells as nodes in a graph and draws a minimum spanning tree



Street et. al. 2018

<https://www.youtube.com/watch?v=XmHDexCtjyw>

In a MST, nodes are connected in such a way that the total sum of distances is minimized. By definition there are no cycles.

Slingshot: trajectory / pseudotime analysis

- Sensitive to upstream analysis choices (clustering and dimensionality reduction)
- No cyclic trajectories (cell cycle...)

Slingshot

```
> library(slingshot)
> sce.slingshot <- slingshot(sce.slingshot,
  clusterLabels = "clusters_phenograph",
  reducedDim = "DiffusionMap",
  start.clus = "Naive")
```

A data object containing the matrix of expression. Supported types include matrix and a singleCellExperiment (sce) object

Slot from the sce object with each cell cluster assignment

The dimensionality reduction to be used.

The starting cluster from which lineages will be drawn. There is also an *end.clus* parameter, if you wish to set which cluster(s) will be forced to be leaf nodes in the graph.

Thank you for your attention!

Please share your opinion about this course!

Course feedback - Flow cytometry
data analysis with R - 2023

