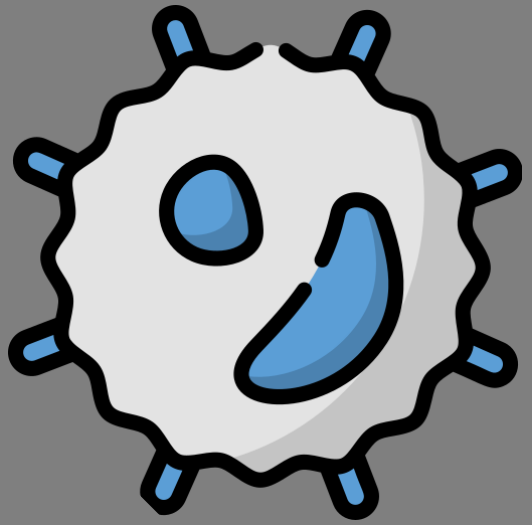
Trajectory inference methods – Progress report



George Alehandro Saad
george-alehandro.SAAD@etu-univ.amu.fr
M2 Bioinformatics - DLAD

Ciphe
Designing genes
Profiling immunity

CLIP

Aix*Marseille université
Initiative d'excellence

1

Introduction

Introduction

Trajectory inference (TI) method that uses the vector space of RNA velocity (E.g. Dynamo).

Employs various notions of mathematics (E.g. DeRahm map, Hodge decomposition, Laplacian matrix)

A new way to calculate the pseudotime starting from a root cell.

Method

2

Method

This approach uses the KNN graph built by the tviblindi workflow to build a transition matrix.

Method

This approach uses the KNN graph built by the tviblindi workflow to build a transition matrix.

Then a weighted graph is built from the transition matrix (weights are reversely proportional to the number of edges of each node).

Method

This approach uses the KNN graph built by the tviblindi workflow to build a transition matrix.

Then a weighted graph is built from the transition matrix (weights are reversely proportional to the number of edges of each node).

The boundary matrix is extracted from this graph.

Method

Derahm Cohomology will map nodes from 0-forms to edges in 1-forms space. The velocity calculated by Dynamo will be used here, alongside the boundary matrix and the coordinates of the points.

Method

Derahm Cohomology will map nodes from 0-forms to edges in 1-forms space. The velocity calculated by Dynamo will be used here, alongside the boundary matrix and the coordinates of the points.

A Laplacian matrix is built based on the transition matrix (the root cell is also pre-selected).

Method

Derahm Cohomology will map nodes from 0-forms to edges in 1-forms space. The velocity calculated by Dynamo will be used here, alongside the boundary matrix and the coordinates of the points.

A Laplacian matrix is built based on the transition matrix (the root cell is also pre-selected).

At this point, we will solve a system of equations. This system is the result of the Hodge decomposition for cohomology groups.

Method

Derahm Cohomology will map nodes from 0-forms to edges in 1-forms space. The velocity calculated by Dynamo will be used here, alongside the boundary matrix and the coordinates of the points.

A Laplacian matrix is built based on the transition matrix (the root cell is also pre-selected).

At this point, we will solve a system of equations. This system is the result of the Hodge decomposition for cohomology groups.

The resulting vector of values contains the pseudotimes of all the cells.

3

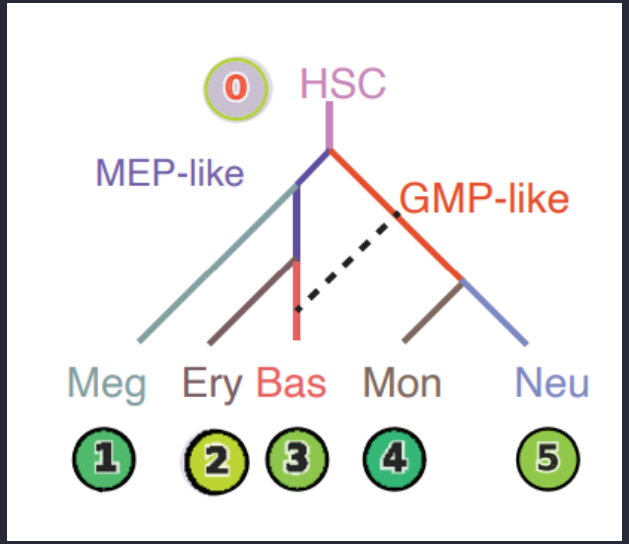
Dataset

Dataset

dyn.sample_data.hematopoiesis()

1974 cells

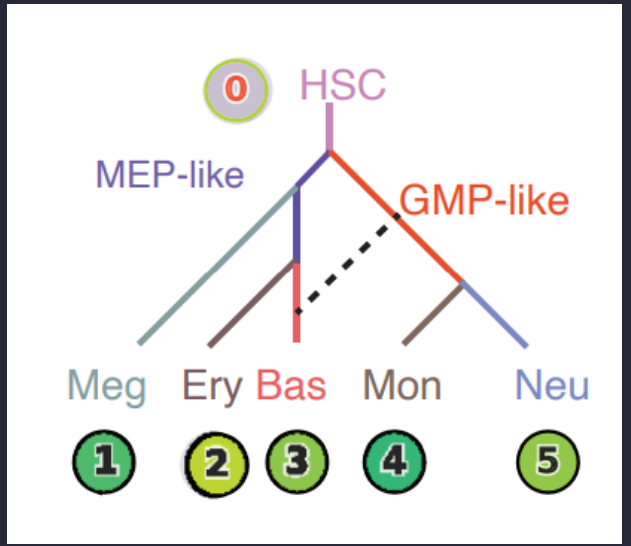
Hematopoietic stem cells (HSC)
Megakaryocyte progenitor cells (MEP)
Granulocyte-macrophage progenitor cells (GMP)
Erythrocytes, basophils, neutrophils, and monocytes.



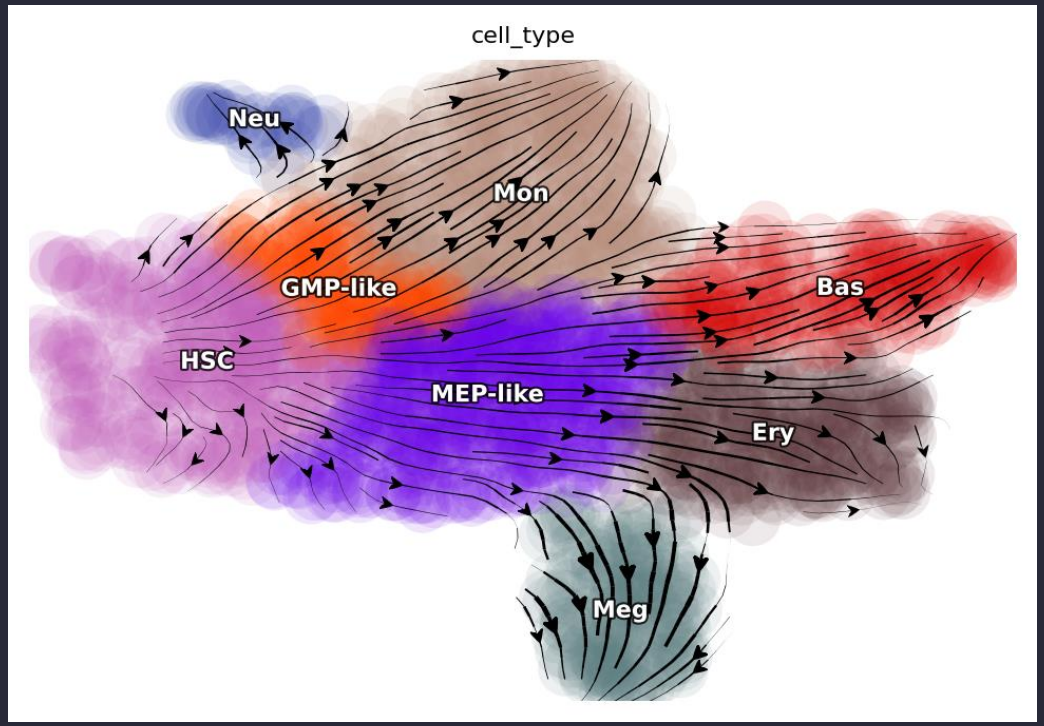
https://pitt.box.com/shared/static/kyh3s4wrxdywupn9wk9r2j27vzlvk8vf.h5ad

Dataset

Population	Number of cells
HSC	389
Megakaryocyte progenitor cells (MEP)	456
Granulocyte-macrophage progenitor cells (GMP)	161
Erythrocytes	234
Basophils	177
Neutrophils	32
Monocytes	423



Dataset



Markers to follow

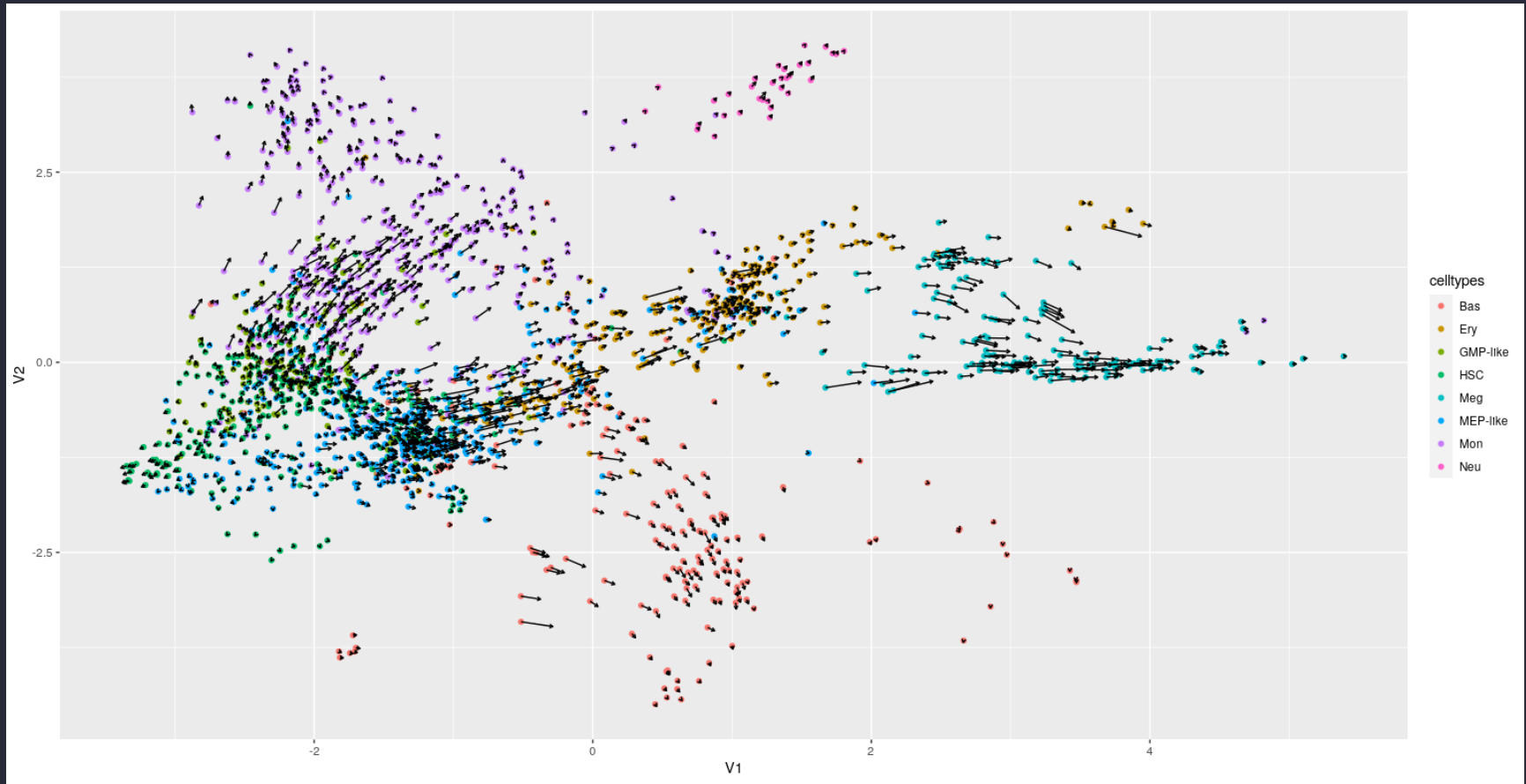
List of markers of interest that (among others) segregate between the pops.

Population	Genes / markers
HSC	CD34, CD38, CD90
Megakaryocyte progenitor cells (MEP)	CD41, CD61
Granulocyte-macrophage progenitor cells (GMP)	CD43, CD117
Erythrocytes	CD235a
Basophils	CD123, CD203c
Neutrophils	CD16, CD66b
Monocytes	CD14, CD16

Application

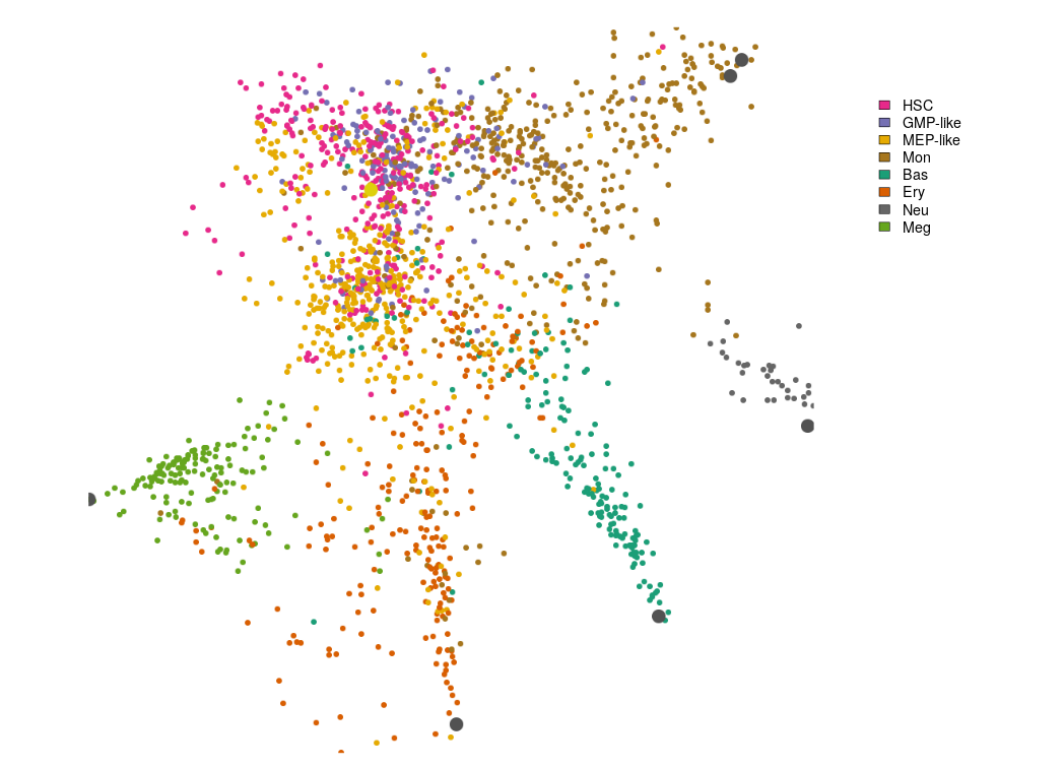
4

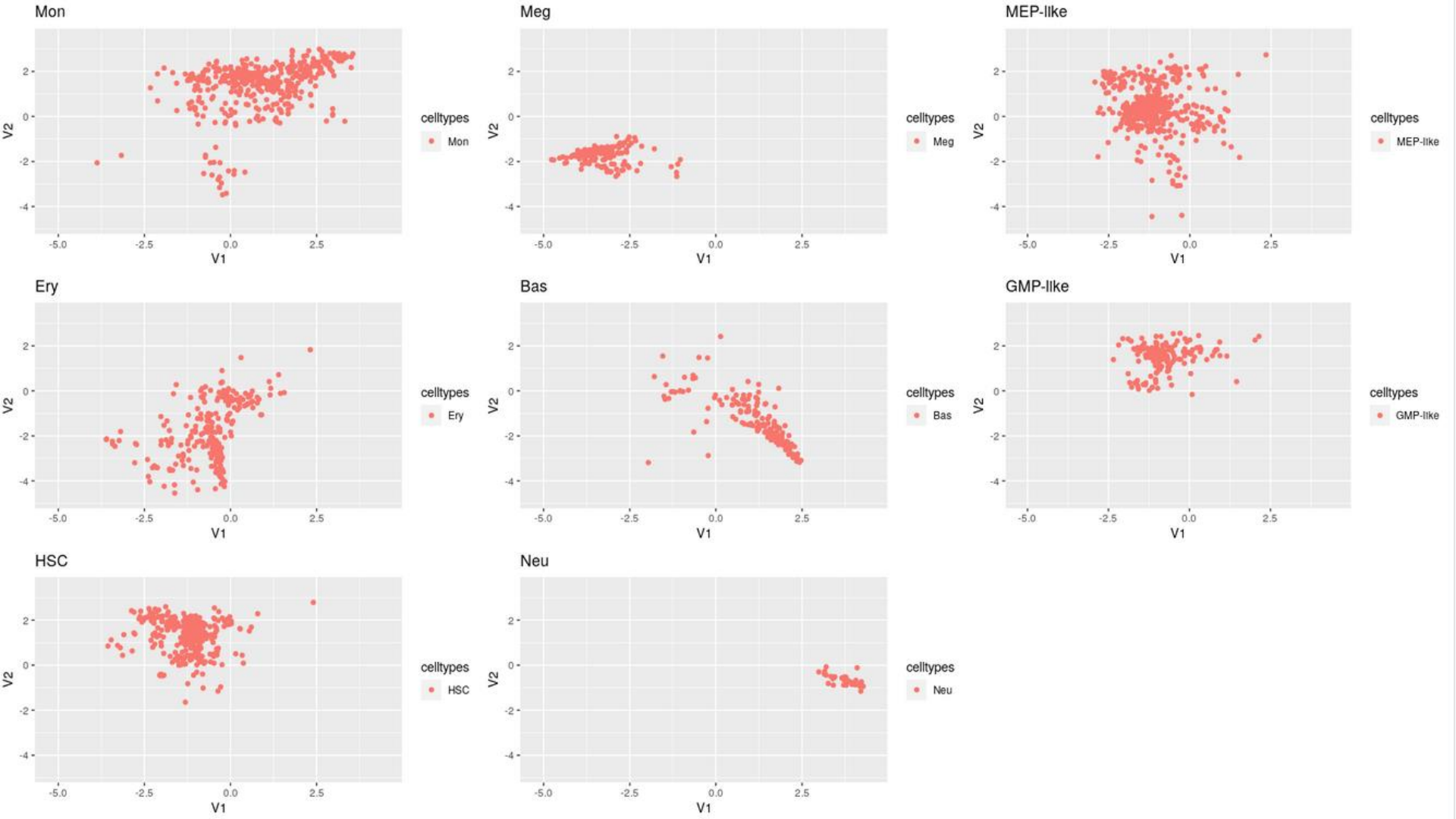
Vaevictis + Velocity vectors Embedded



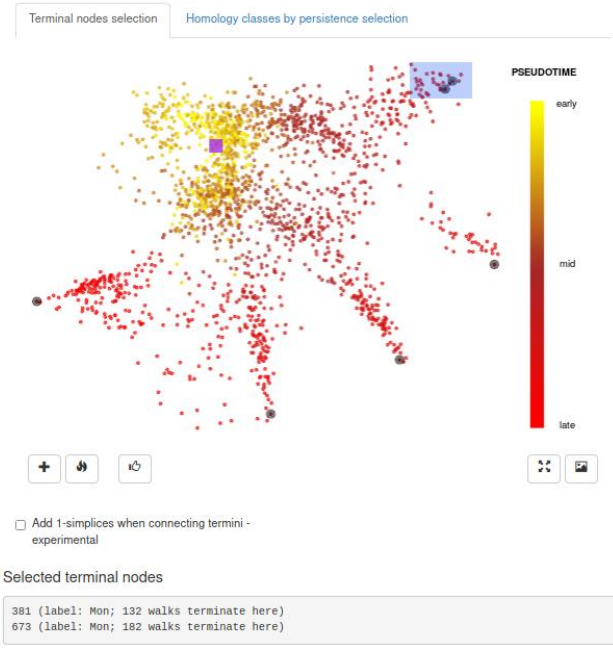
Tviblindi – Vaevictis output

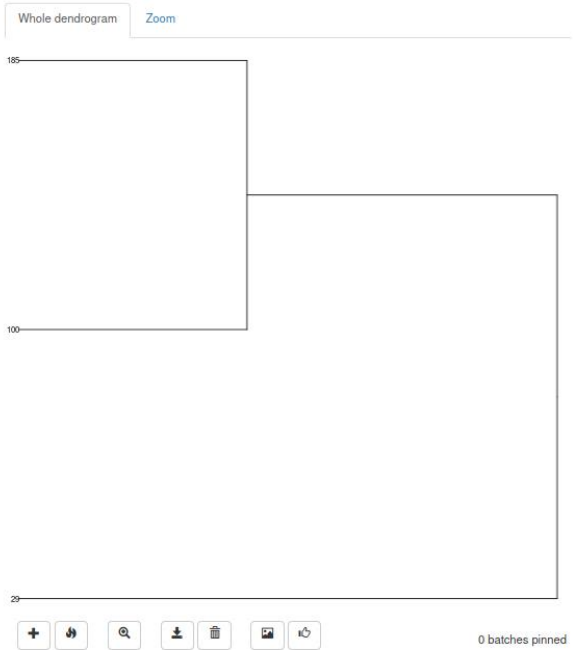
Using the PCA dataset over 30 principal components

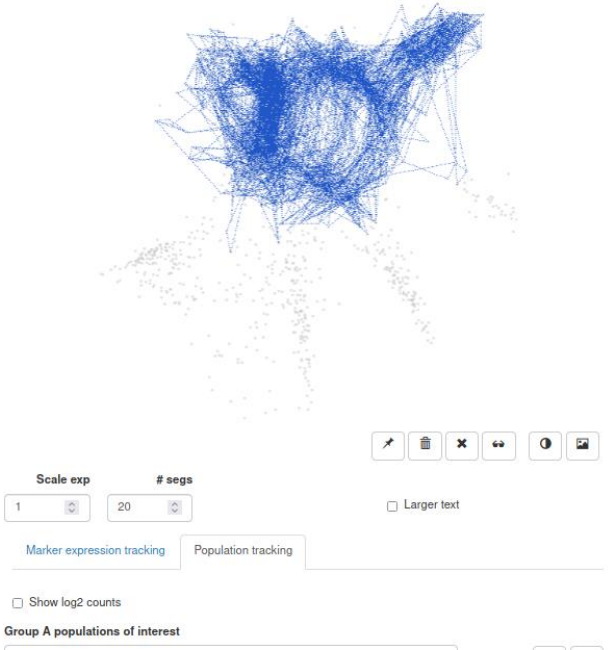



Mon
Meg
MEP-like
Ery
Bas
GMP-like
HSC
Neu
V1
V2
celltypes
Mon
Meg
MEP-like
Ery
Bas
GMP-like
HSC
Neu

Tviblindi – Vaevictis output

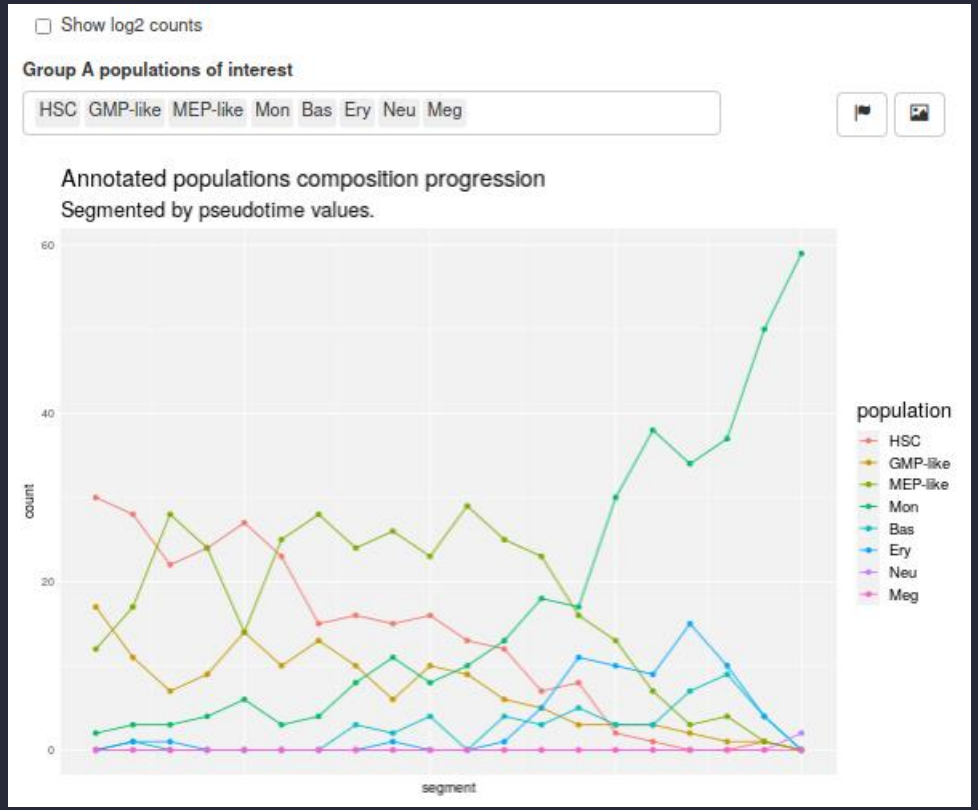




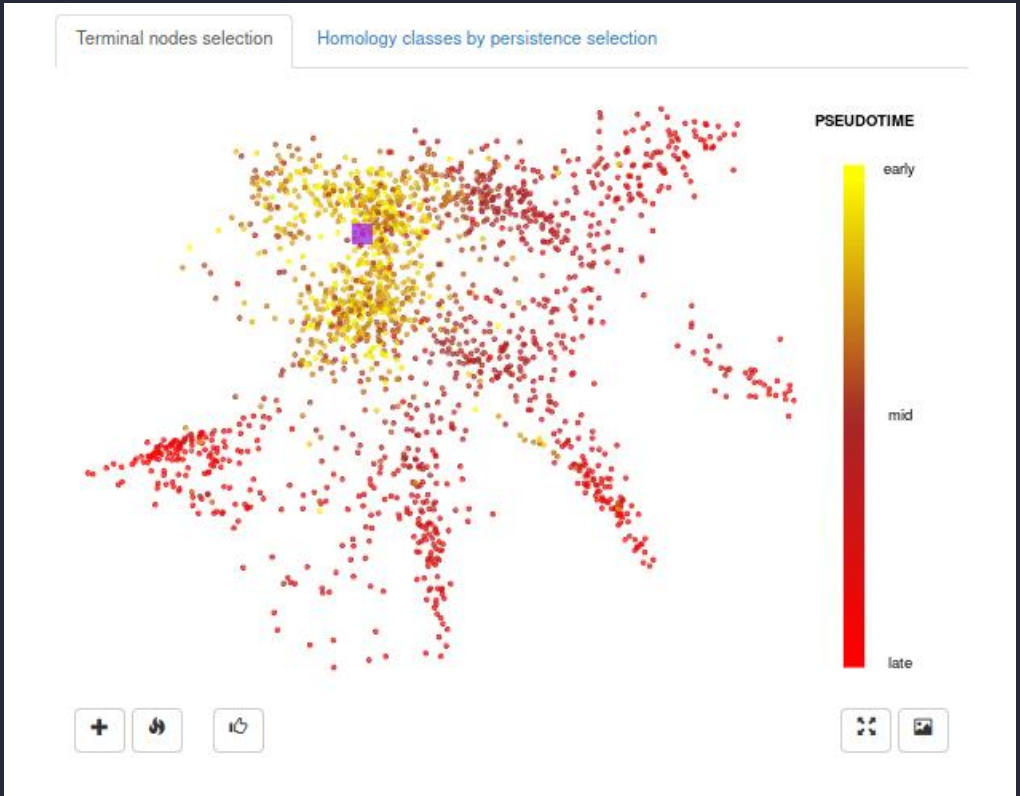


Using the standard pseudotime calculation of tviblindi.

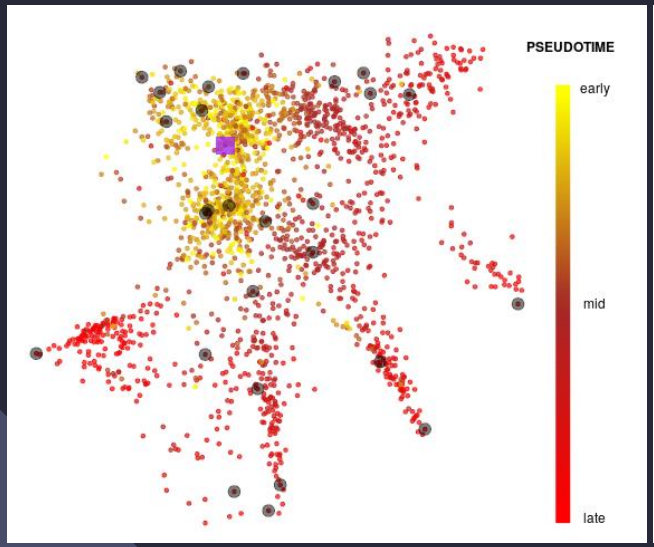
Tviblindi – Vaevictis output



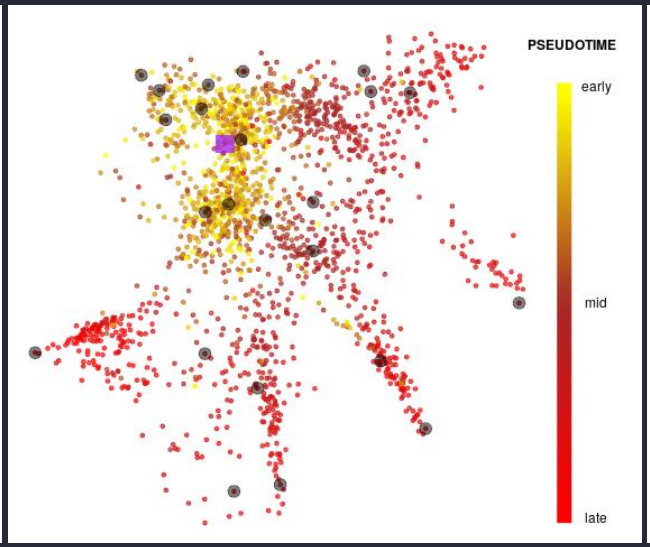
Tviblindi – Newly calculated pseudotime



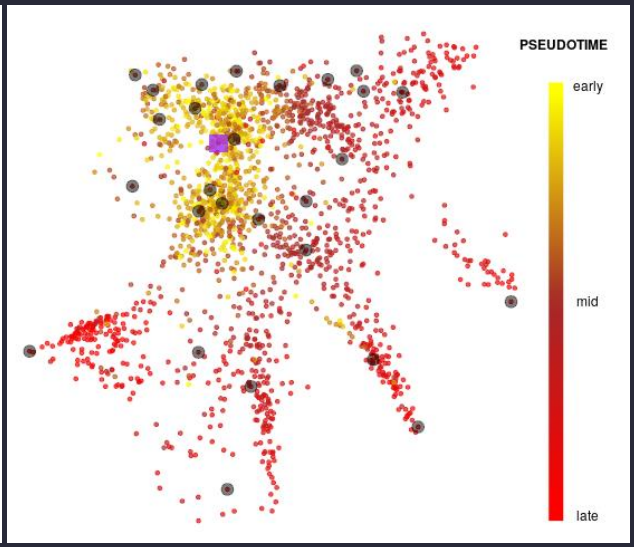
The choice of number of neighbors



10

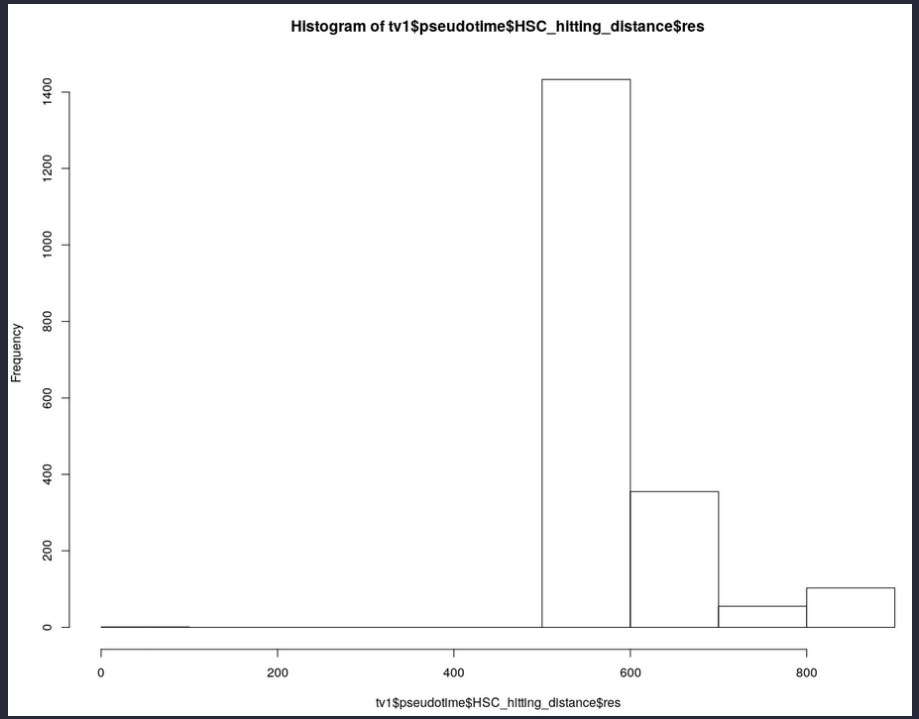


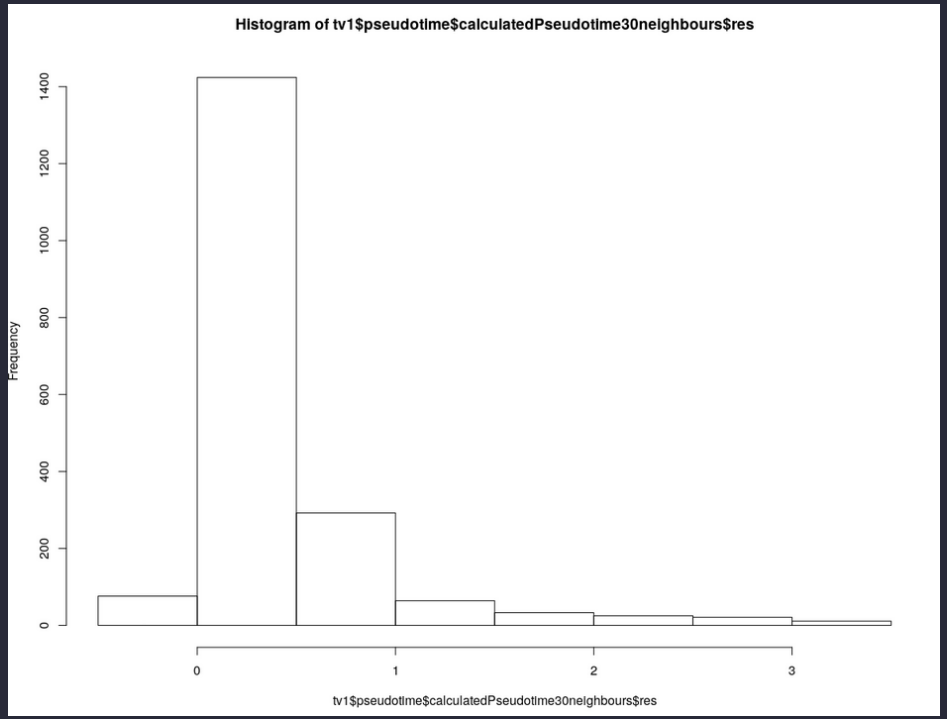
30



90

Comparison of methods





Things to do next

- Still have to test and benchmark with other datasets.
- Compare both pseudotimes, understand the differences.
- Should find an approach to combine both of the ways of calculation.
- Refactor the code and find a way to integrate the new calculations (if the results are promising).

THANK YOU!