

# dyno: INFERRING, VISUALIZING AND INTERPRETING TRAJECTORIES



Wouter Saelens\*<sup>1,2</sup>, Robrecht Cannoodt\*<sup>1,2,3</sup>, Helena Todorov<sup>1,2,4</sup>, Yvan Saeys<sup>1,2</sup>

\* Equal contribution

<sup>1</sup> Data mining and Modelling for Biomedicine, VIB Center for Inflammation Research, Ghent, Belgium. <sup>2</sup> Department of Applied Mathematics, Computer Science and Statistics, Ghent University, Ghent, Belgium. <sup>3</sup> Center for Medical Genetics, Ghent University Hospital, Ghent, Belgium.

<sup>4</sup> Centre International de Recherche en Infectiologie, Inserm, U1111, Université Claude Bernard Lyon 1, CNRS, UMR5308, École Normale Supérieure de Lyon, Univ Lyon, F-69007, Lyon, France

► dyno is a **toolkit** for inferring, visualizing and interpreting **trajectories** in R

[dyno.dynverse.org](https://dyno.dynverse.org)



## Selecting the most optimal method(s)

### ► Choice of method depends on the:

**User** Expectations about the topology  
Available computing resources  
Relative importance to particular trajectory aspects

**Data** Number of cells and features  
Prior information (such as start cells)

### ► Examples:

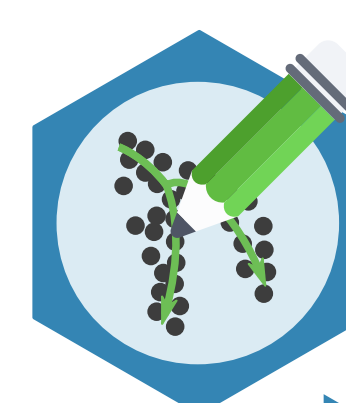
► Our app gives **recommendations** based on these criteria

Install locally in R: `devtools::install_github("dynverse/dynguidelines")`

Or go to: [guidelines.dynverse.org](https://guidelines.dynverse.org)

► Recommendations are based on the results from our **benchmark**:

Robrecht Cannoodt @ **Poster 1033** and [benchmark.dynverse.org](https://benchmark.dynverse.org)



## Inferring trajectories

### ► 71 tools for trajectory inference

...and counting  
...each with their own input/output interface

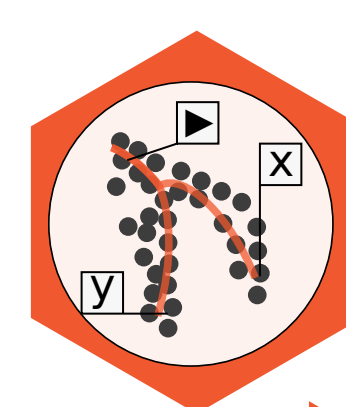
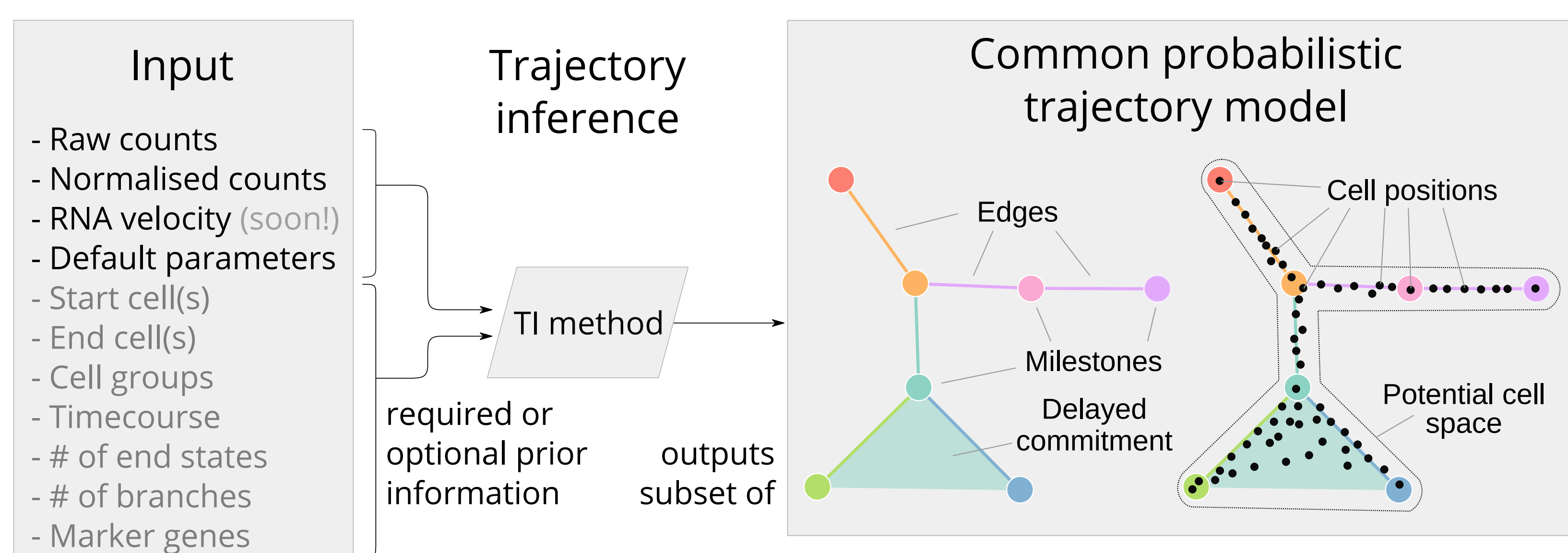
► We developed a **common input and output interface** for **55 methods**

► Each method can be run with one line

`infer_trajectory(dataset, "my_favorite_ti_method")`

► New methods can be included through

[methods.dynverse.org](https://methods.dynverse.org)



## Annotating the trajectory

► Include additional information on top of the trajectory to make it **interpretable**

- Rooting based on markers (or manually)
- (soon!) Directionality of edges based on RNA velocity
- Labelling milestones based on markers (or manually)
- We welcome additional ideas about common operations



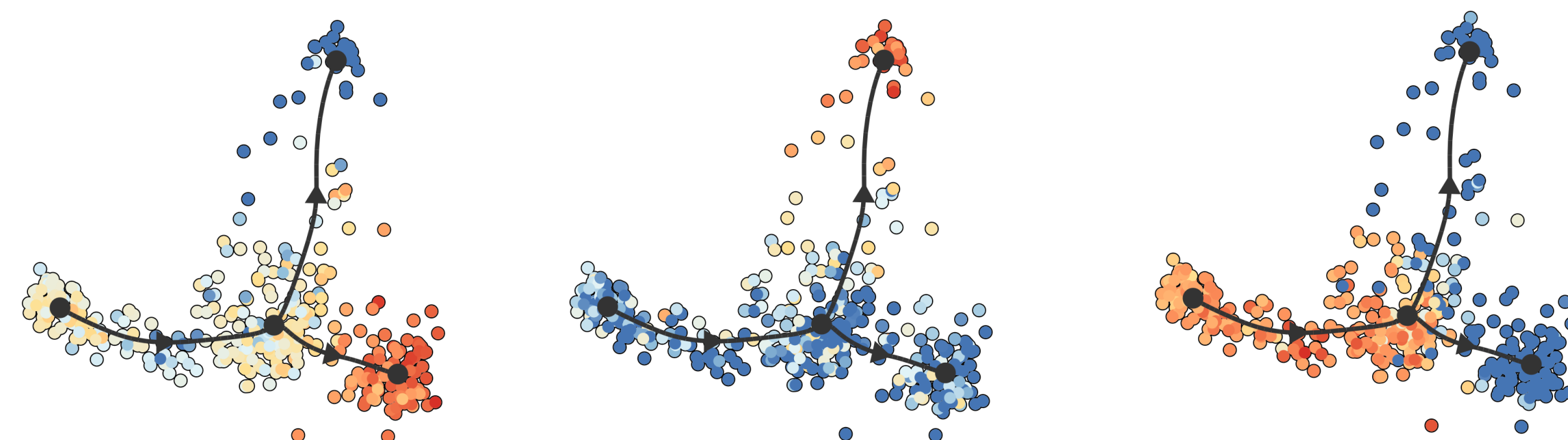
## Detecting differential expression

► Various **expression patterns** can be detected

**Overall**

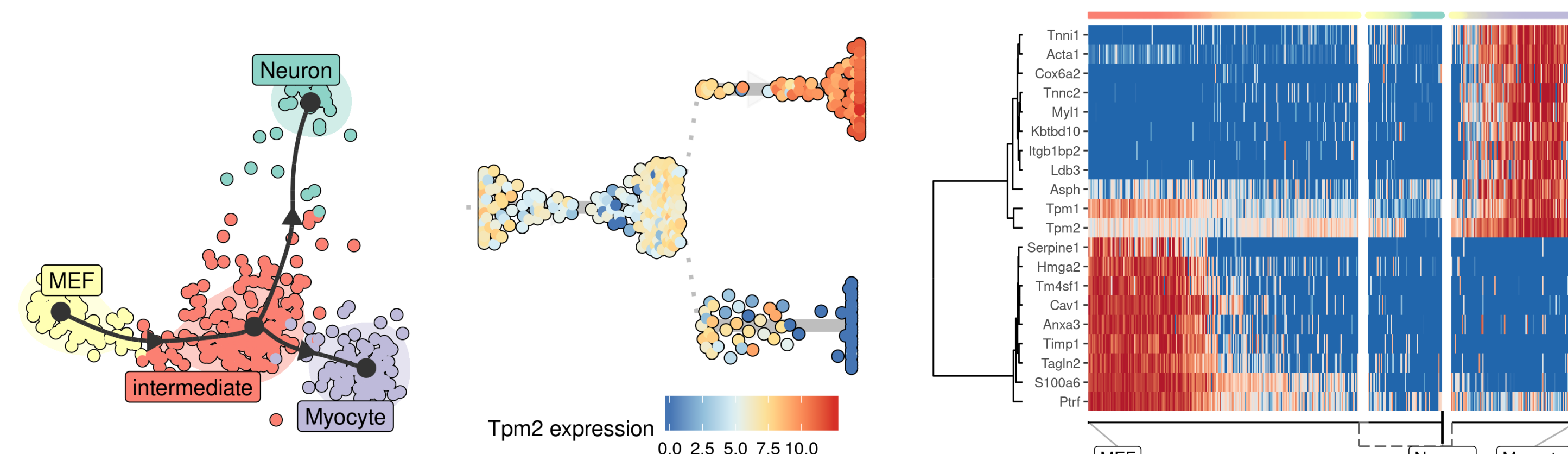
**Branch**

**Branch point**



## Visualizing the trajectory

► Visualize a model in **multiple ways**



► Compare models on a **common embedding**

