Лекция 13

Определение траекторий дифференцировки клеток в scRNA-seq

Даниил Бобровский

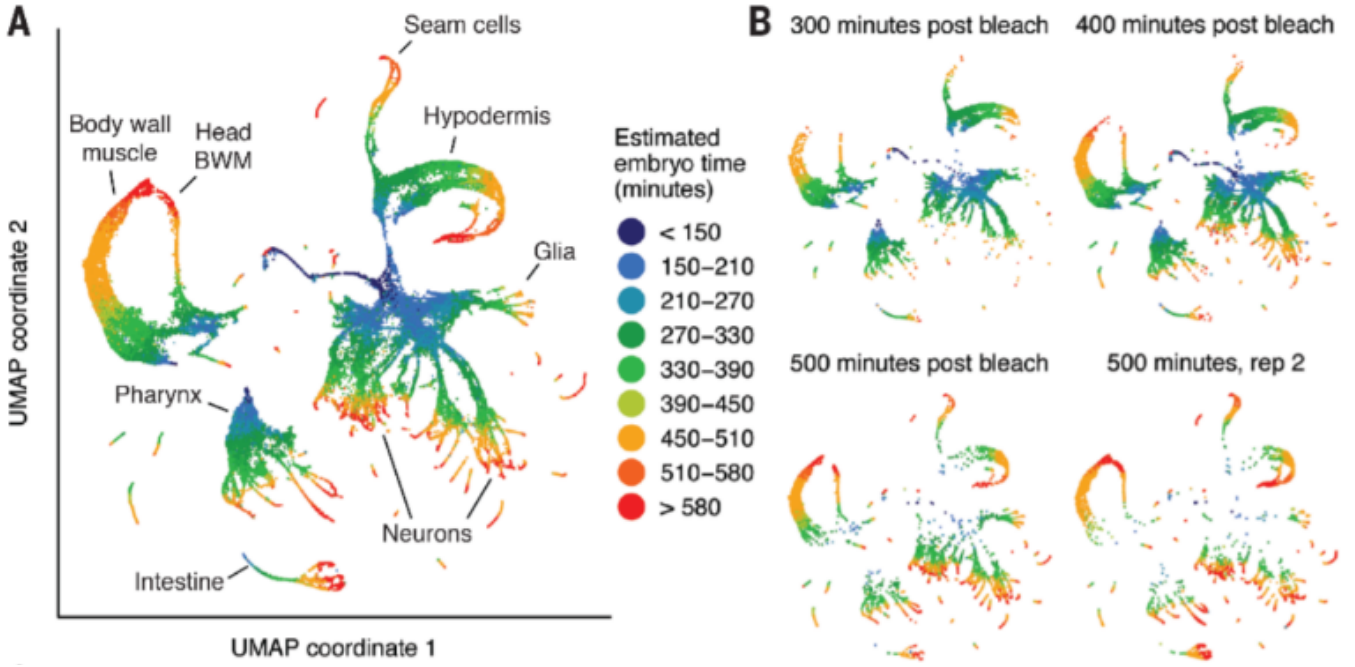
магистрант EPFL

сотрудник лаборатории системной биологии нейроразвития, EPFL

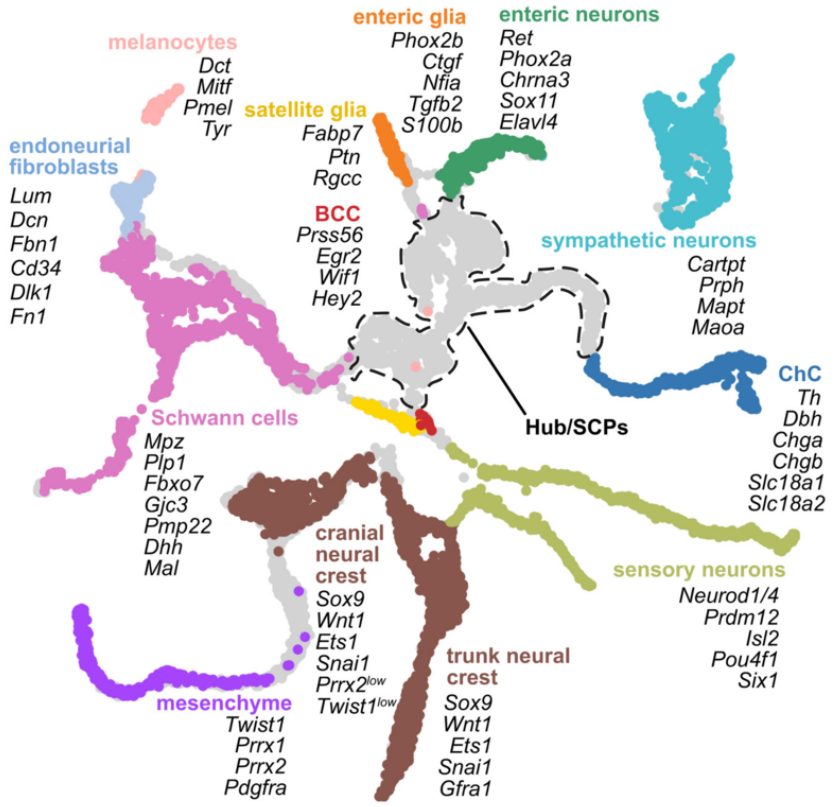
Задача

scRNA-seq успешно применяют для изучения дифференцировки клеток

По маркерным генам мы можем устанавливать типы клеток, но как можно с помощью вычислительных методов определить, какие клетки происходят из каких?



https://doi.org/10.1126/science.aax1971



https://doi.org/10.15252/embj.2021108780

Подзадачи

Pseudotime/latent time/differentiation potential

Изобрести какую-то меру степени дифференцированности, чтобы по ней можно было упорядочивать разные клетки

Trajectory inference

Установить траекторию дифференцировки как кривую в пространстве экспрессии генов или как ориентированный граф, найти точки бифуркации, ...

Fate mapping

Подсчитать для каждой клетки вероятность разных путей её дальнейшего развития

Давным-давно, в 2002 году...

Reconstructing the temporal ordering of biological samples using microarray data FREE

Paul M. Magwene, Paul Lizardi, Junhyong Kim

Постановка задачи:

- Траектория дифференцировки - это непрерывная гладкая функция $\vec{f}(t) = [x_1(t), x_2(t), ..., x_d(t)]$, d - число генов
- Транскриптомные данные - набор точек $V = \{\vec{\psi}(s_0), \vec{\psi}(s_1), ..., \vec{\psi}(s_n)\}$ в случайные моменты времени s_i, где $\vec{\psi}(s) = \vec{f}(s) + \vec{\delta}(s)$, $\vec{\delta}(s)$ - технический и биологический шум
- Определение траектории - задача восстановления кривой $\vec{f}$

Восстановление кривой

Подходы к восстановлению кривой :

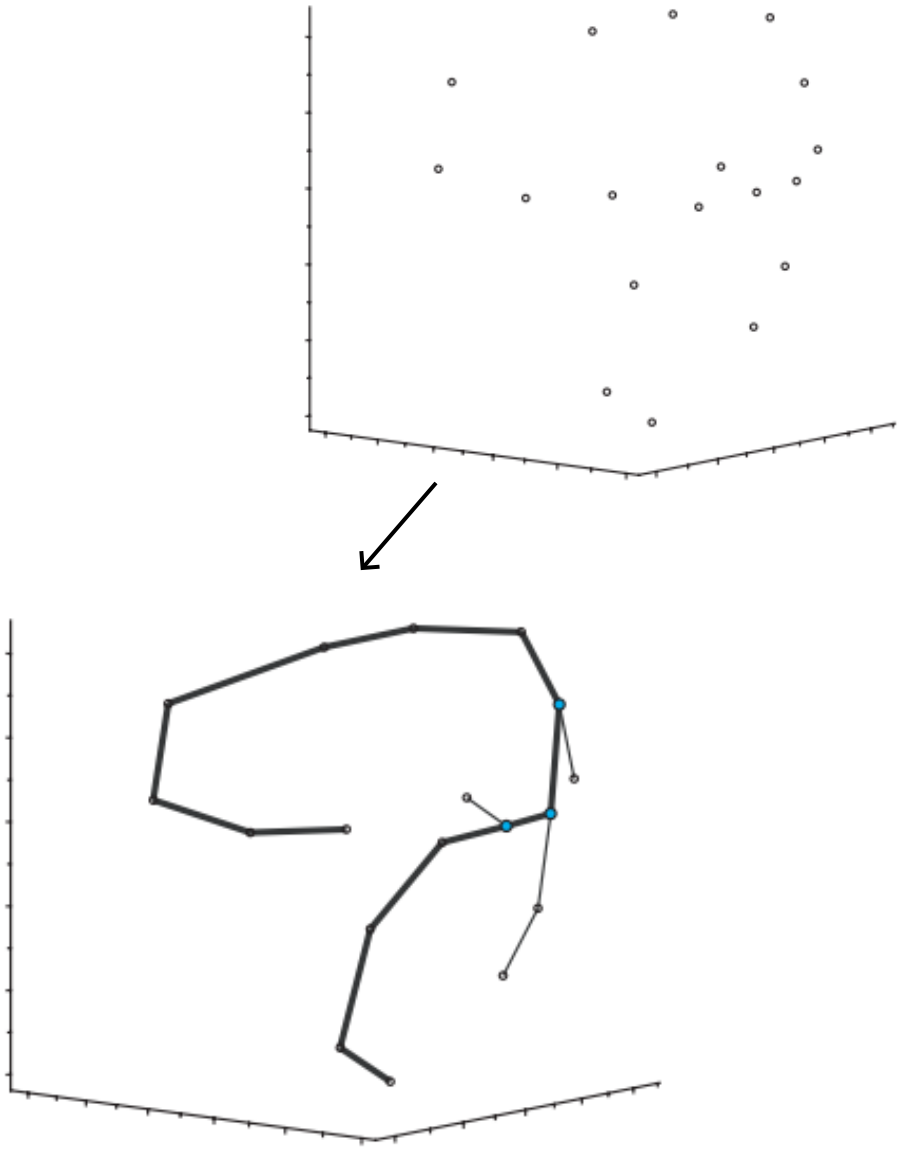
- polygonal reconstruction - ищем граф такой, что в нем соединены все образцы, соседние на $\vec{f}$
- principal curve - проводим кривую через центр "облака" точек в многомерном пространстве

Поскольку точек в случае данных с микрочипов не очень много, используется polygonal reconstruction

Polygonal reconstruction

Алгоритм:

1. Находим минимальное остовное дерево (minimum spanning tree, MST) в взвешенном полном графе; веса ребер - расстояния между образцами. Если полученное дерево - путь, ура!

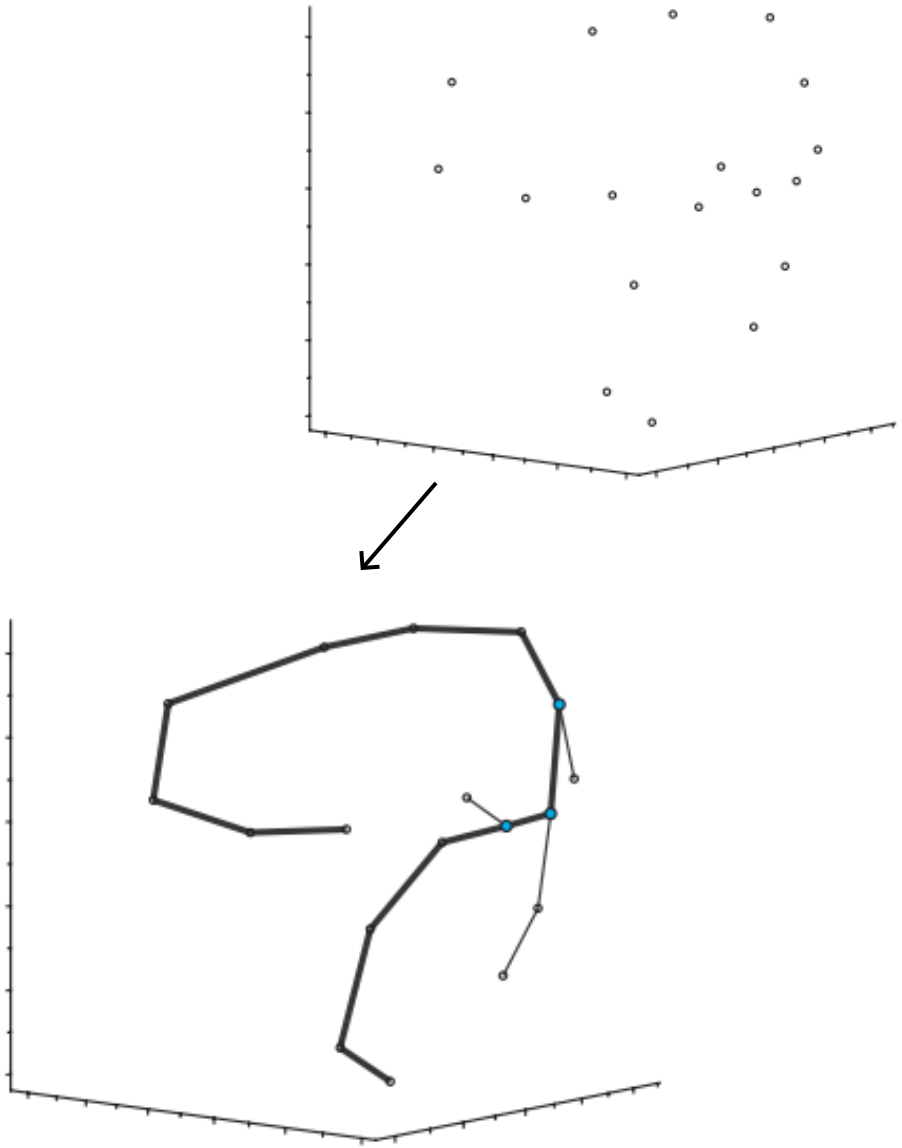


doi.org/10.1093/bioinformatics/btg081

Polygonal reconstruction

Алгоритм:

1. Находим минимальное остовное дерево (minimum spanning tree, MST) в взвешенном полном графе; веса ребер - расстояния между образцами. Если полученное дерево - путь, ура!
2. Если ответвлений мало и они короткие веточках, будем считать это шумом

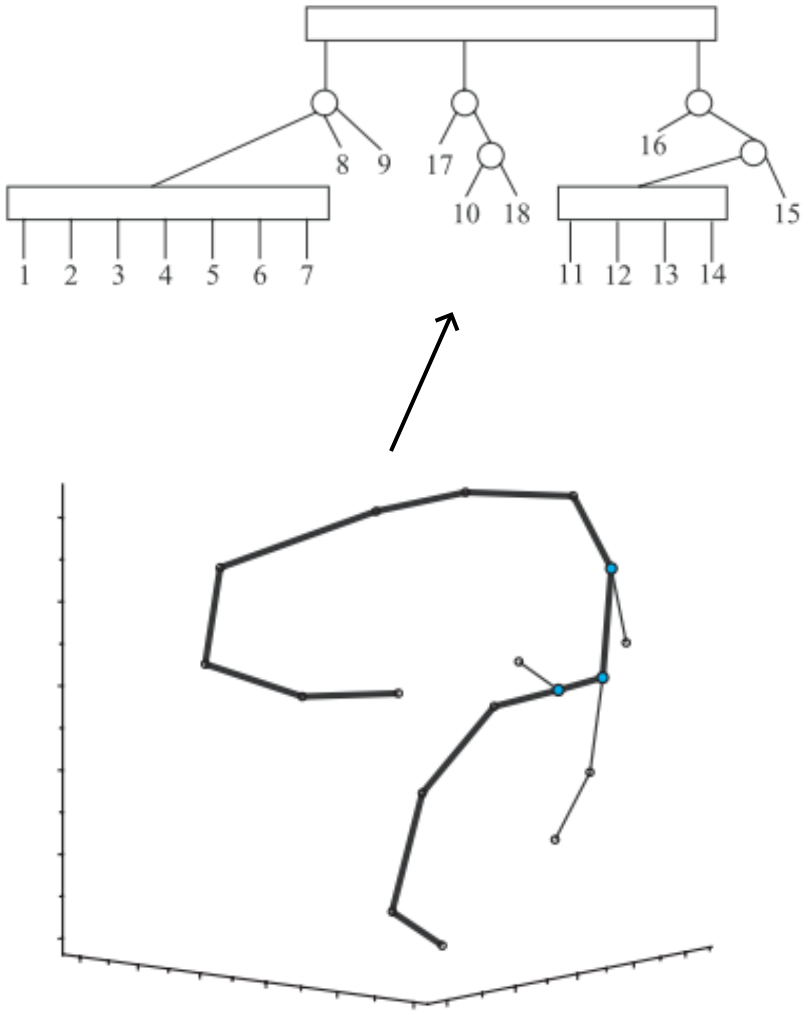


doi.org/10.1093/bioinformatics/btg081

Polygonal reconstruction

Алгоритм:

1. Находим минимальное остовное дерево (minimum spanning tree, MST) в взвешенном полном графе; веса ребер - расстояния между образцами. Если полученное дерево - путь, ура!
2. Если ответвлений мало и они короткие веточках, будем считать это шумом
3. Получим все возможные последовательности без разветвлений, удовлетворяющие полученному графу

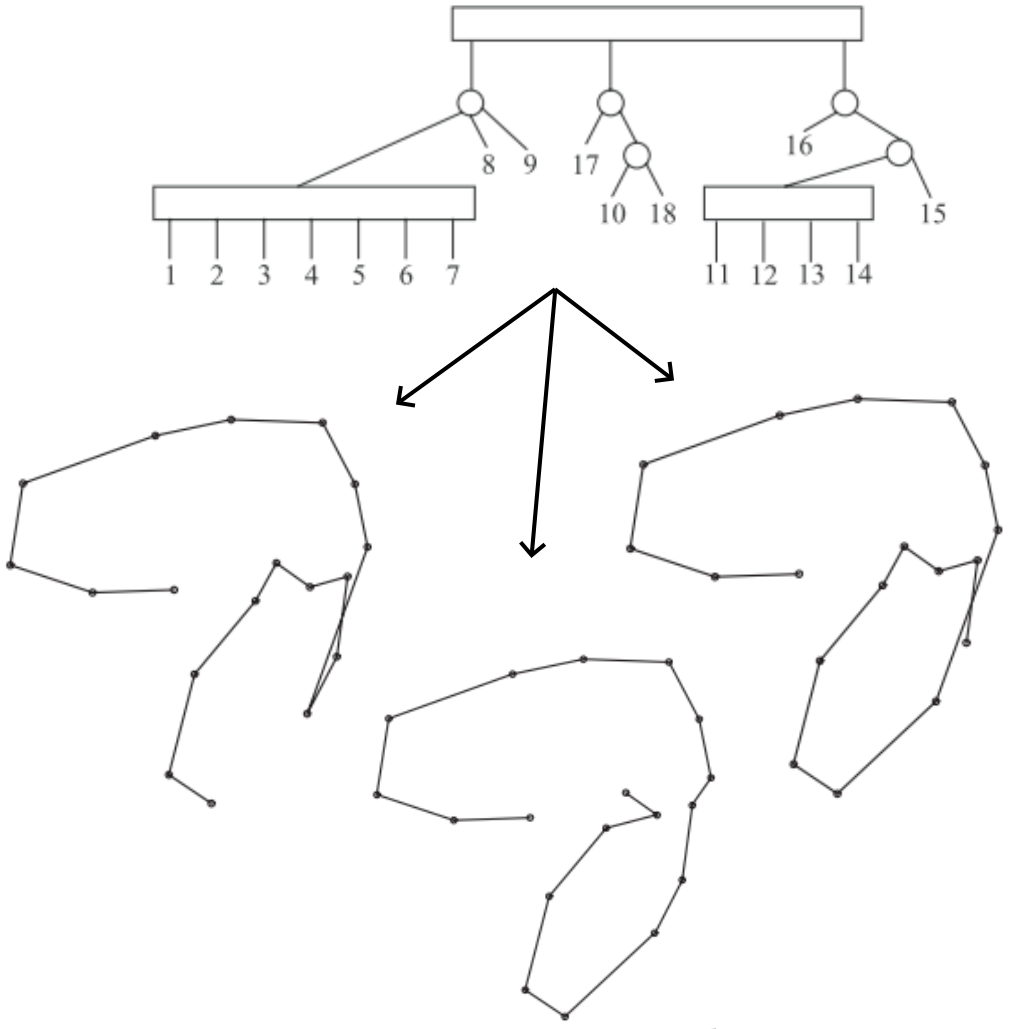


doi.org/10.1093/bioinformatics/btg081

Polygonal reconstruction

Алгоритм:

1. Находим минимальное остовное дерево (minimum spanning tree, MST) в взвешенном полном графе; веса ребер - расстояния между образцами. Если полученное дерево - путь, ура!
2. Если ответвлений мало и они короткие веточках, будем считать это шумом
3. Получим все возможные последовательности без разветвлений, удовлетворяющие полученному графу



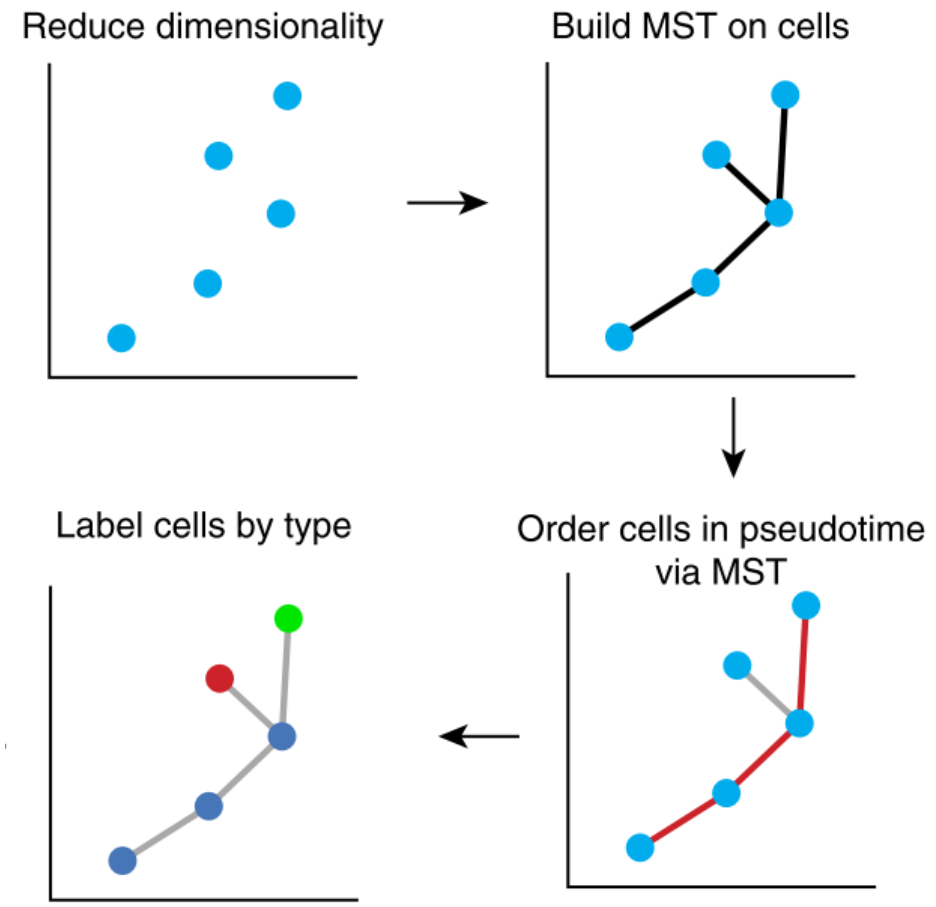
doi.org/10.1093/bioinformatics/btg081

Monocle

Trapnell et al., 2014

По сути - реализация предыдущего алгоритма для scRNA-seq

- Снижение размерности с помощью independent component analysis (ICA)
- Введено понятие **псевдовремени** - длина пути по MST от начальной клетки, указанной пользователем
- Пользователь указывает число терминально дифференцированных клеток k, и k самых длинных ответвлений считаются за альтернативные траектории



doi.org/10.1038/nbt.2859

GAM

Определение дифференциально экспрессируемых по ходу псевдовремени генов - с помощью генерализованных аддитивных моделей (GAM)

Распределение \ Функция f	$f(x_i^T) = x_i^T * \beta$	**непараметрическая f**
Нормальное	Линейная регрессия	Ядерное сглаживание (kernel smoothing)
Экспоненциальное семейство	Генерализованные линейные модели (GLM)	Генерализованные аддитивные модели (GAM)

$$g(\mathbb{E}[Y]) = \beta_0 + f_1(x_1) + f_2(x_2) + ... + f_m(x_m)$$

GAM

$$g(\mathbb{E}[Y]) = \beta_0 + f_1(x_1) + f_2(x_2) + ... + f_m(x_m)$$

- Левая часть та же, что и в GLM: оцениваем матожидание распределения Y; если нужно, применяем link function
- Правая часть: сумма гладких функций; например, кусочно-заданных (сплайнов)

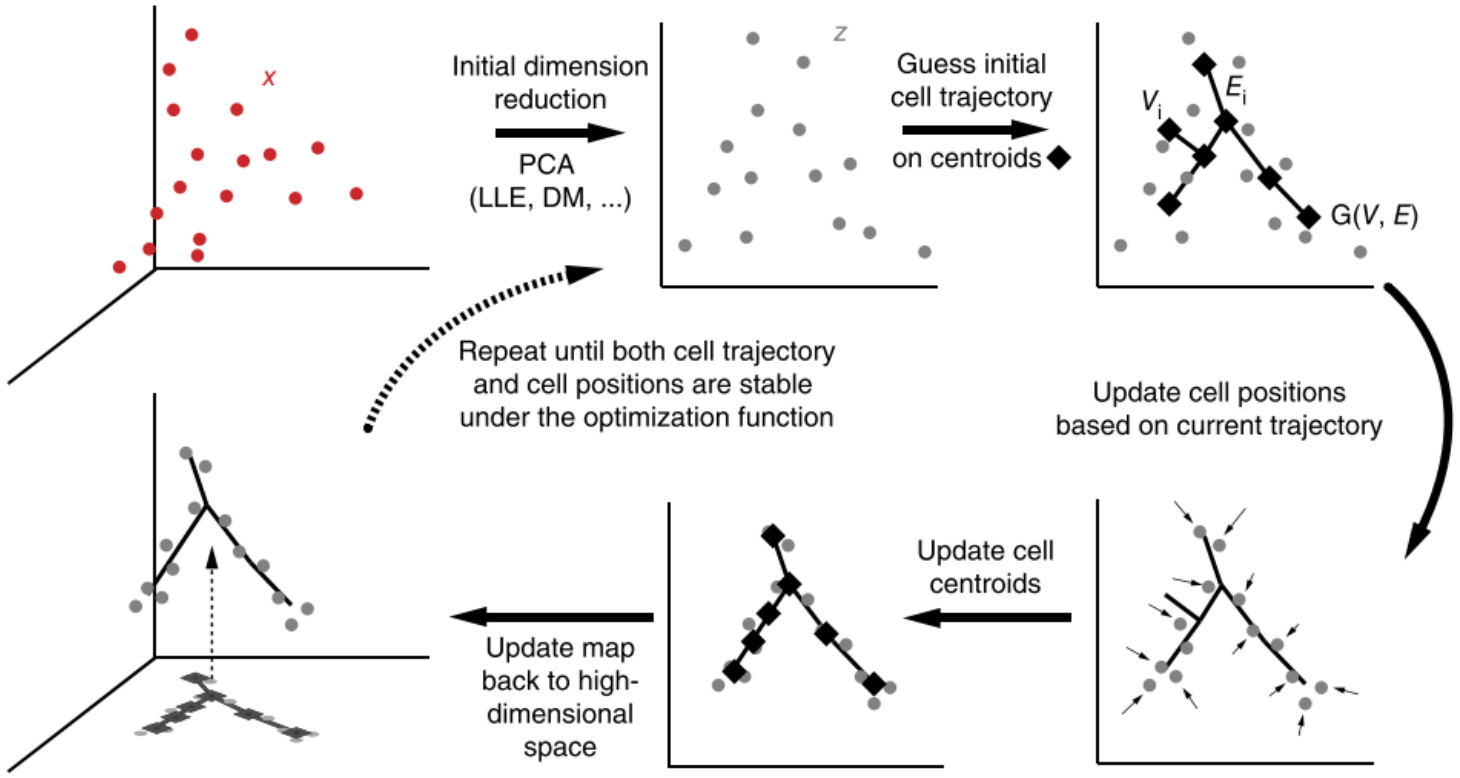
В нашем случае, Y - экспрессия гена, x - псевдовремя

GAM можно использовать с практически любыми методами trajectory inference, и мы самостоятельно реализуем GAM на семинаре

Monocle2

polygonal reconstruction + principal curves = **principal graph**

Совместим задачу получения латентного представления с задачей построения графа



doi.org/10.1038/nmeth.4402

Reversed graph embedding

- $X = \{x_1, ..., x_N\}$ - входные данные, экспрессия для N клеток в многомерном пространстве
- $Z = \{z_1, ..., z_N\}$ - латентное представление в пространстве сниженной размерности
- Граф G в латентом пространстве, вершины V_i соответствуют точкам z_i
- $b_{i,j}$ - вес ребра (V_i, V_j); чем выше $b_{i,j}$, тем больше схожи точки z_i и z_j
- $f_G(z_i)$ - отображение точки z_i в какую-то точку пространства исходной размерности

Построение такого графа - задача оптимизации:

$$\min_{G \in G_b} \min_{f_g \in F} \min_{Z} \sum_{i=1}^{N} ||x_i - f_G(z_i)||^2 + \frac{\lambda}{2} \sum_{(V_i, V_j) \in \epsilon} b_{i,j} ||f_G(z_i) - f_G(z_j)||^2$$

Reversed graph embedding

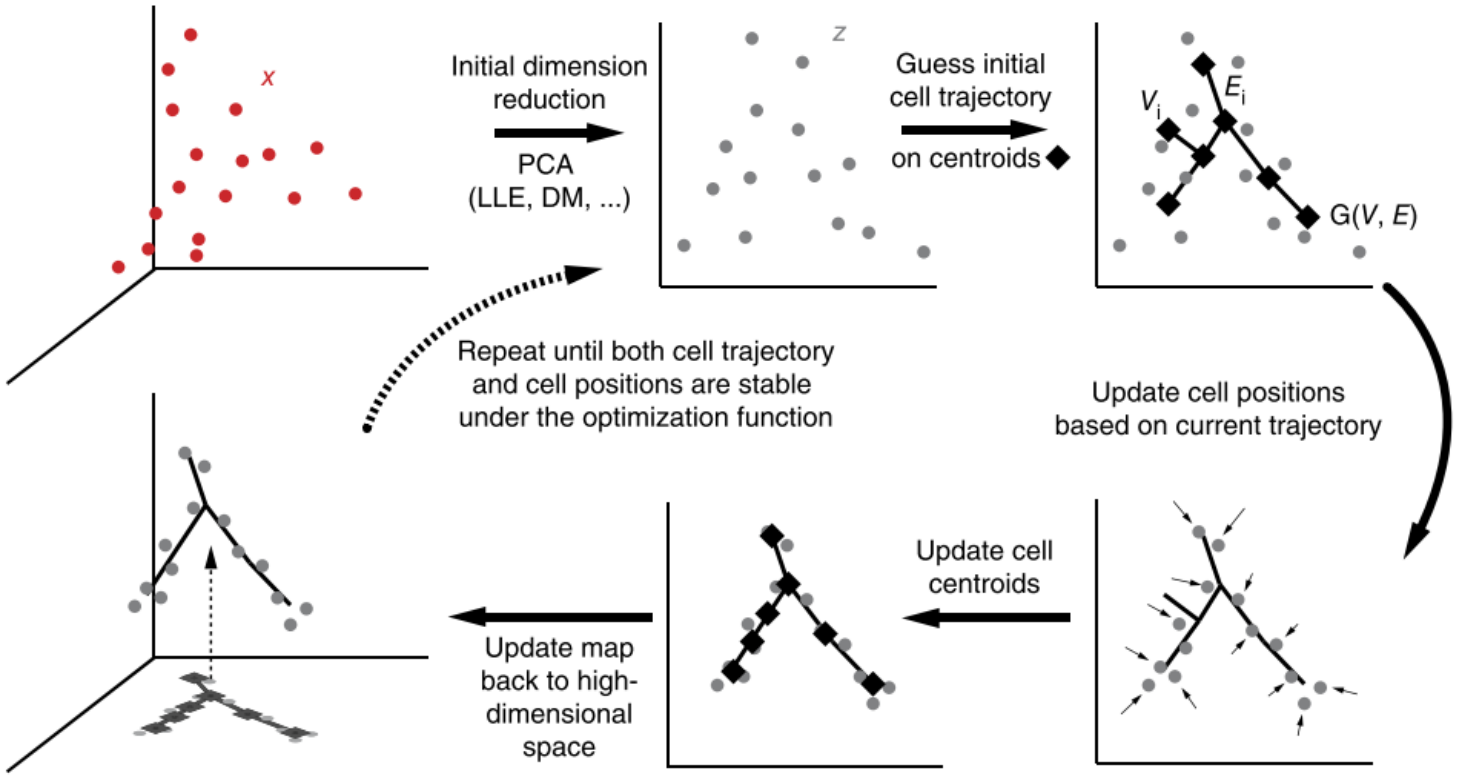
$$\min_{G \in G_b} \min_{f_g \in F} \min_{Z} \sum_{i=1}^{N} ||x_i - f_G(z_i)||^2 + \frac{\lambda}{2} \sum_{(V_i, V_j) \in \epsilon} b_{i,j} ||f_G(z_i) - f_G(z_j)||^2$$

Минимизируя эти две суммы, мы получаем G, Z и f_G такие, что:

- Функция f_G - хорошее отображение из латентного пространства в исходное (первая сумма)
- Если z_i и z_j схожи в латентном пространстве, их отображения расположены рядом в исходном (вторая сумма)

Алгоритмы

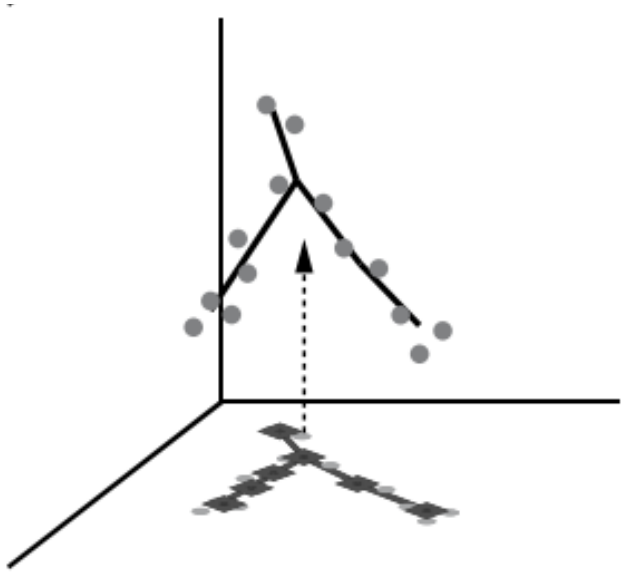
- SimplePPT - учит граф на всех клетках
- DDRTree - в процессе делает кластеризацию, и граф строится для центров кластеров; дефолтный алгоритм в Monocle2



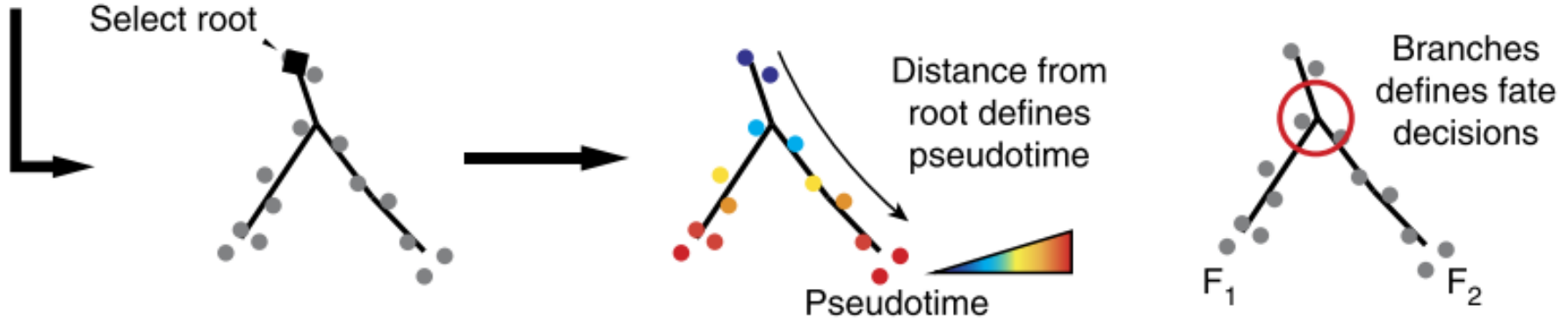
Алгоритм Monocle2 при выборе DDRTree

doi.org/10.1038/nmeth.4402

Определение траектории



- Клетки проецируются на полученный граф
- Для проекций строится MST, по нему вычисляется псевдовремя
- Точки ветвления определяются самой структурой principal tree, пользователю не нужно указывать число бифуркаций



doi.org/10.1038/nmeth.4402

Diffusion maps

Предложены как метод снижения размерности, основанный на моделировании дифференцировки как Марковского процесса

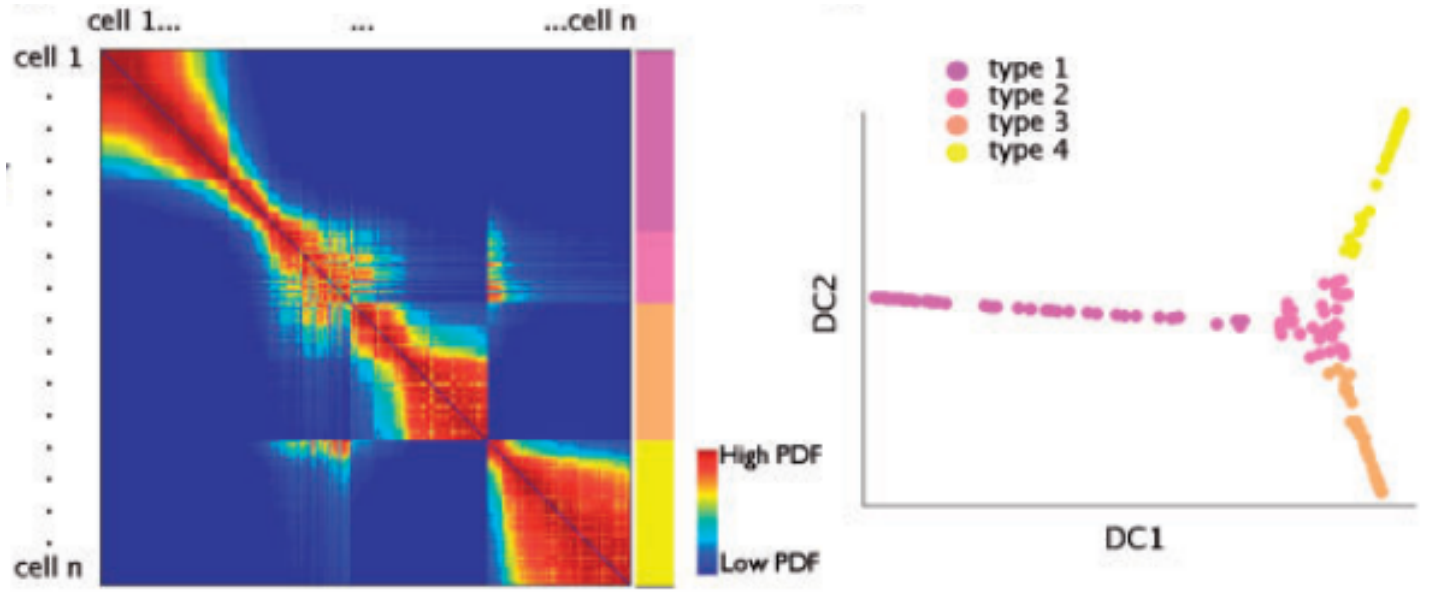
Наши клетки - это возможные состояния системы

В ходе дифференцировки система переходит от одного состояния к другому

- Пусть мы находимся в клетке x
- Позволим себе "диффундировать" в пространстве экспрессии генов по Гауссовой функции с центром в x
- Чем ближе другая клетка, тем выше вероятность перейти в нее
- Получаем $\tilde{P}$ - матрицу вероятностей переходов между клетками

Diffusion maps

Собственные векторы этой матрицы $\tilde{P}$ сооветствуют возможным направлениям "диффузии" клеток, и их можно использовать как оси для снижения размерности



doi.org/10.1093/bioinformatics/btv325

Упражнение*

*Для тех, кому очень хочется разобраться в моделировании дифференцировки с помощью Марковских процессов

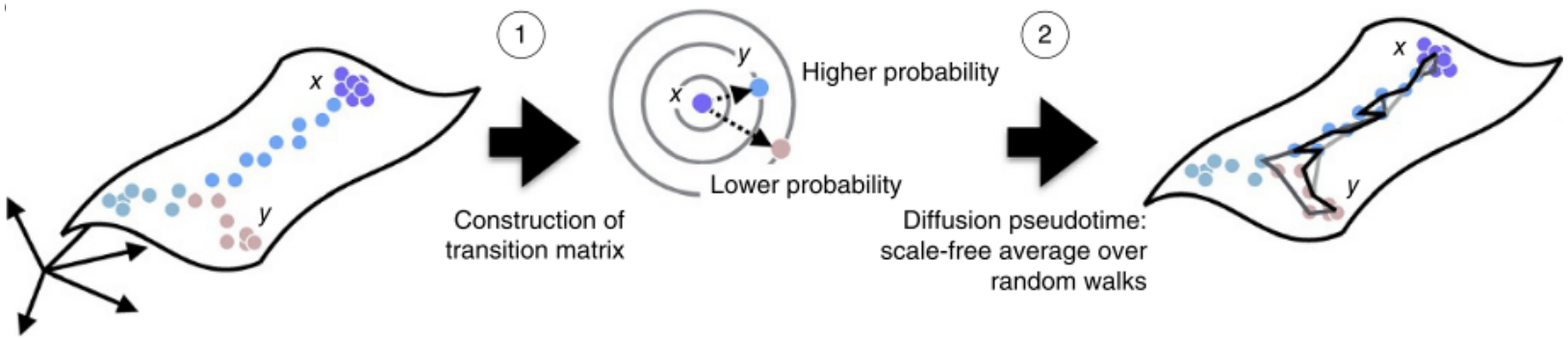
Докажите, что правый собственный вектор матрицы Марковского процесса перехода клеток в клетки с собственным значением $\lambda = 1$ соответствует стационарному распределению

Подсказки:

- Прочитайте про стационарное распределение: https://brilliant.org/wiki/stationary-distributions/
- Осознайте, почему всегда существует левый собственный вектор с собственным значением $\lambda = 1$
- Подумайте, почему в нашем случае правые собственные векторы совпадают с левыми собственными векторами

Diffusion pseudotime

- Получаем $\tilde{P}$ и случайно бродим из клетки в клетку
- Начиная бродить из каждой клетки, подсчитываем вероятность оказаться в любой другой через t шагов, $t \in \{1, 2, ..., \infty\}$
- Diffusion pseudotime (DPT) клетки - евклидово расстояние между векторами для этой клетки и для указанной пользователем корневой клетки: чем ближе клетка к корню, тем более похожа с корневой клеткой ее возможная дальнейшая судьба

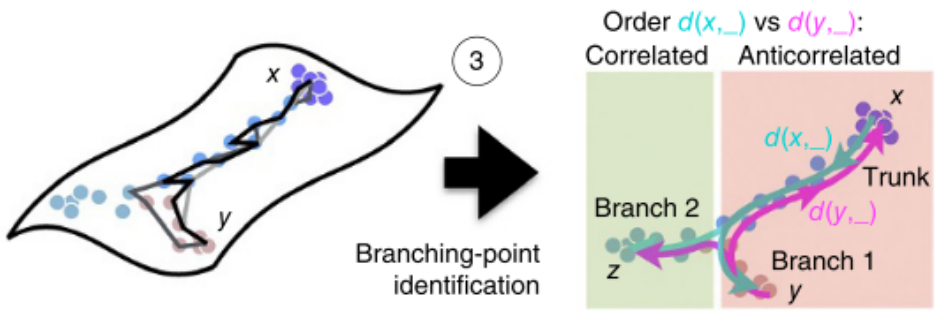


doi.org/10.1038/nmeth.3971

Diffusion pseudotime

- Получаем $\tilde{P}$ и случайно бродим из клетки в клетку
- Начиная бродить из каждой клетки, подсчитываем вероятность оказаться в любой другой через t шагов, $t \in \{1, 2, ..., \infty\}$
- Diffusion pseudotime (DPT) клетки - евклидово расстояние между векторами для этой клетки и для указанной пользователем корневой клетки: чем ближе клетка к корню, тем более похожа с корневой клеткой ее возможная дальнейшая судьба

- Поиск бифуркаций: сравниваем DPT от корневой клетки и от максимально далекой от нее до всех остальных - они антикоррелируют до точки ветвления, а потом начинают коррелировать



doi.org/10.1038/nmeth.3971

PAGA

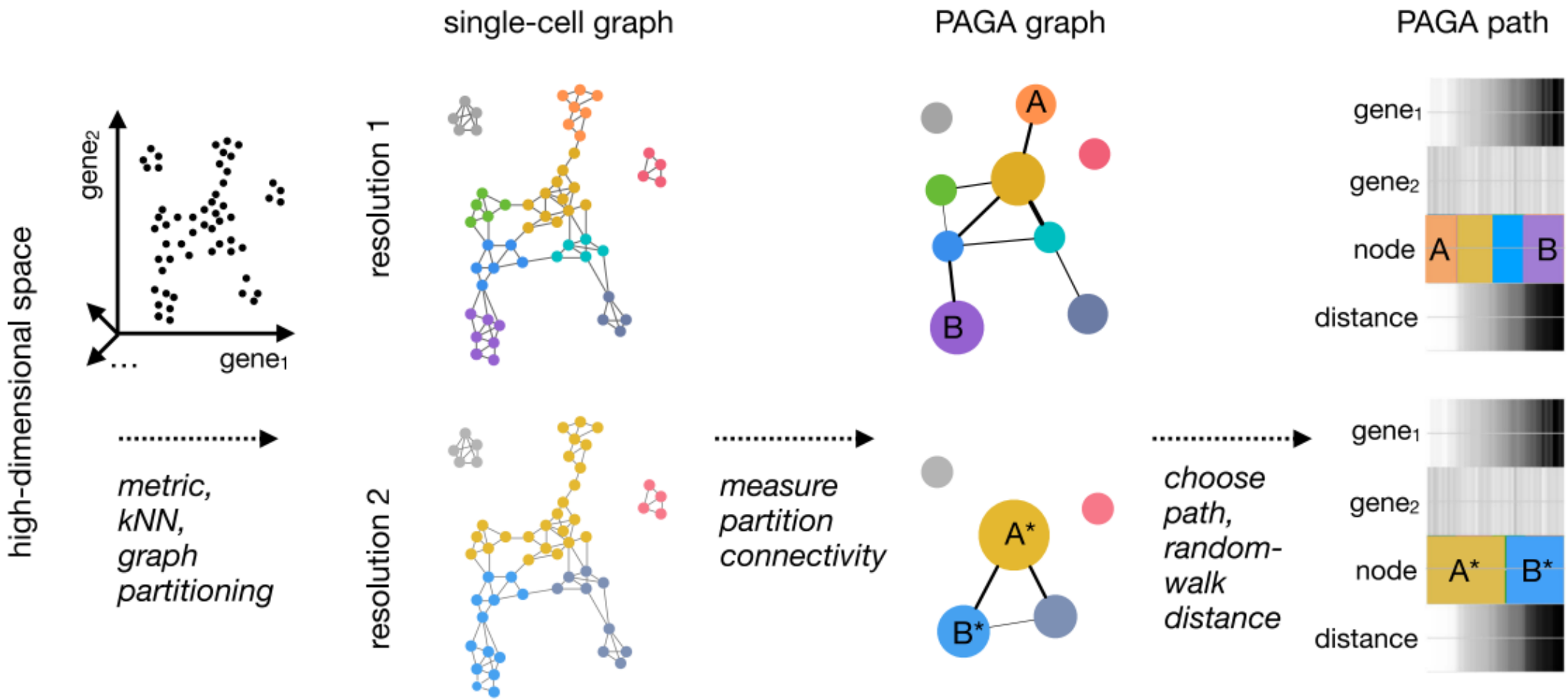
Partition-based graph abstraction; не относится к методам trajectory inference, но используется некоторыми из них

Алгоритм:

1. Строим граф ближайших соседей
2. Делаем кластеризацию
3. Между всеми кластерами считаем **PAGA connectivity measure** - насколько ребер между двумя кластерами больше, чем было бы в случайном графе

PAGA + DPT

Получив PAGA-граф, можно считать DPT внутри каждой компоненты связности

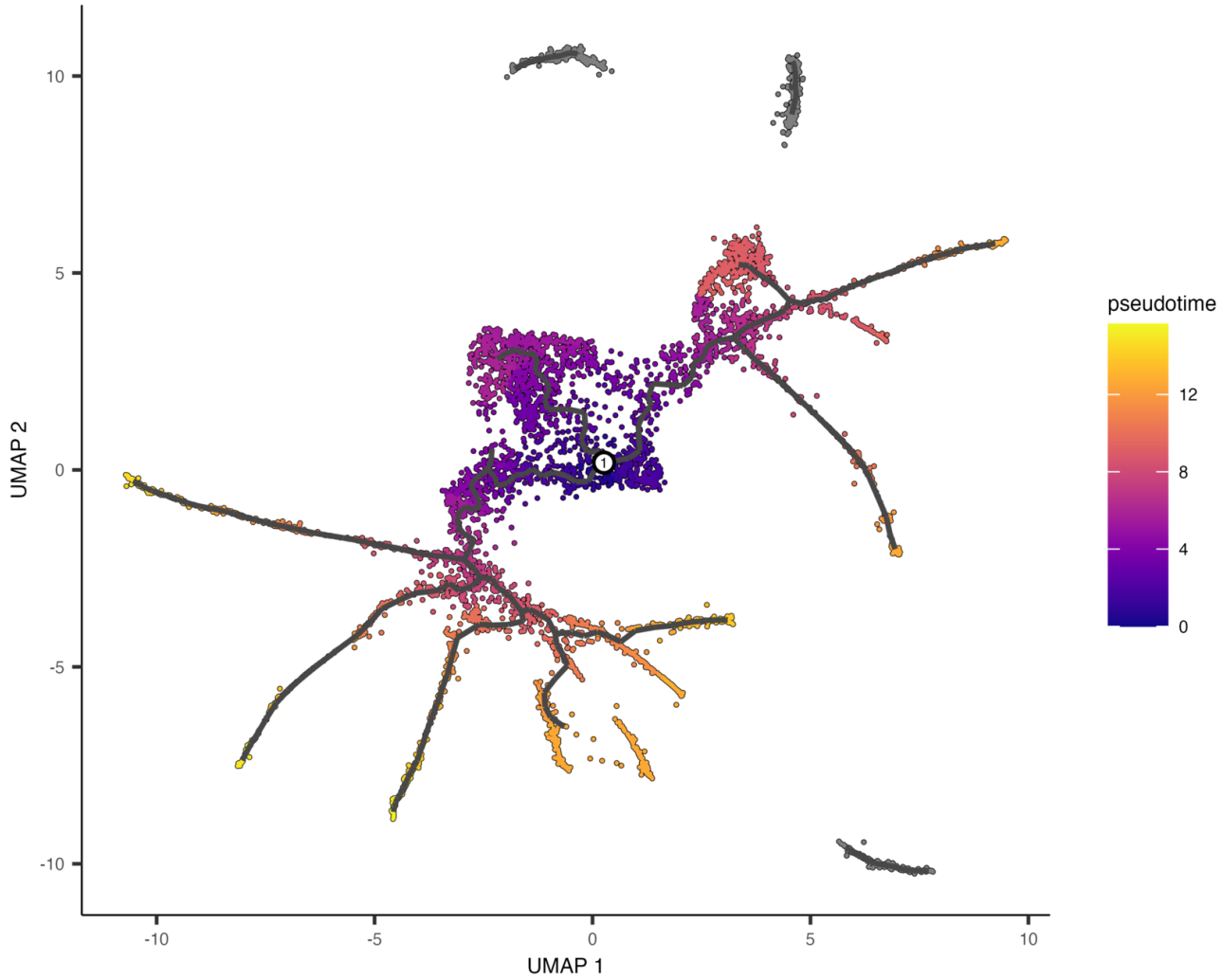


Monocle3

Наверное, самый популярный метод trajectory inference

Основан на эмбеддинге UMAP, очень чувствительном к параметрам

Доступен в виде пакета на R; на первый взгляд user-friendly, пока вы не пытаетесь совмещать его с другими методами

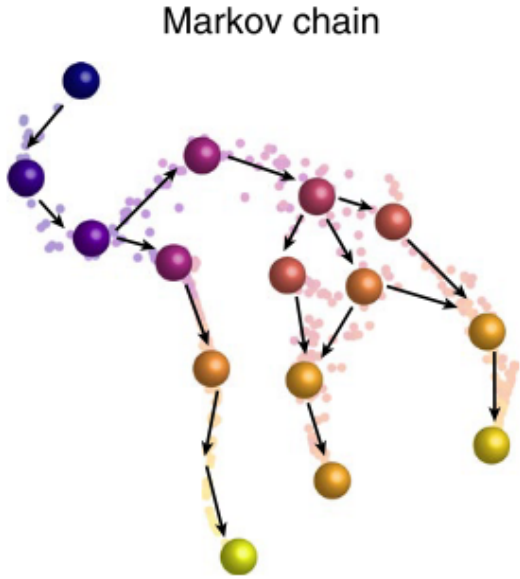


https://cole-trapnell-lab.github.io/monocle3/docs/trajectories/

Алгоритм Monocle3

1. Снижение размерности (UMAP), получение PAGA-графа
2. Для каждой компоненты связности учим principal graph, используется SimplePPT с модификациями:
   - Учим граф только для клеток, близких к центрам кластеров (K-means кластеризация внутри PAGA-компонент)
   - Если находим короткое ответвление, убираем его из графа
   - В графе могут быть циклы
3. Проецируем клетки на ближайшие ребра графа
4. Пользователь указывает корневые вершины, от них по графу считается псевдовремя
5. Поиск дифференциально экспрессируемых по полученному графу генов

Palantir



Дифференцировка как Марковский процесс

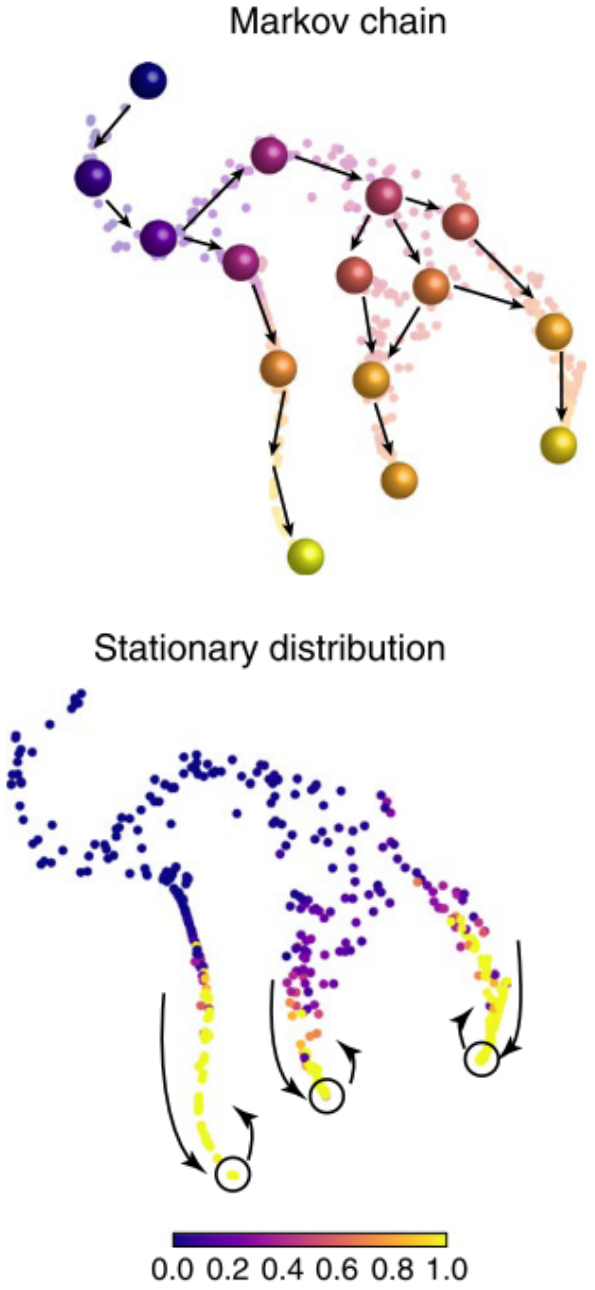
1. Строим граф ближайших соседей в пространстве diffusion map
2. Псевдовремя - расстояние по кратчайшему пути в этом графе от начальной клетки
    - Построение графа и подсчет расстояния с учетом общей топологии эмбеддинга

doi.org/10.1038/s41587-019-0068-4

Palantir

Дифференцировка как Марковский процесс

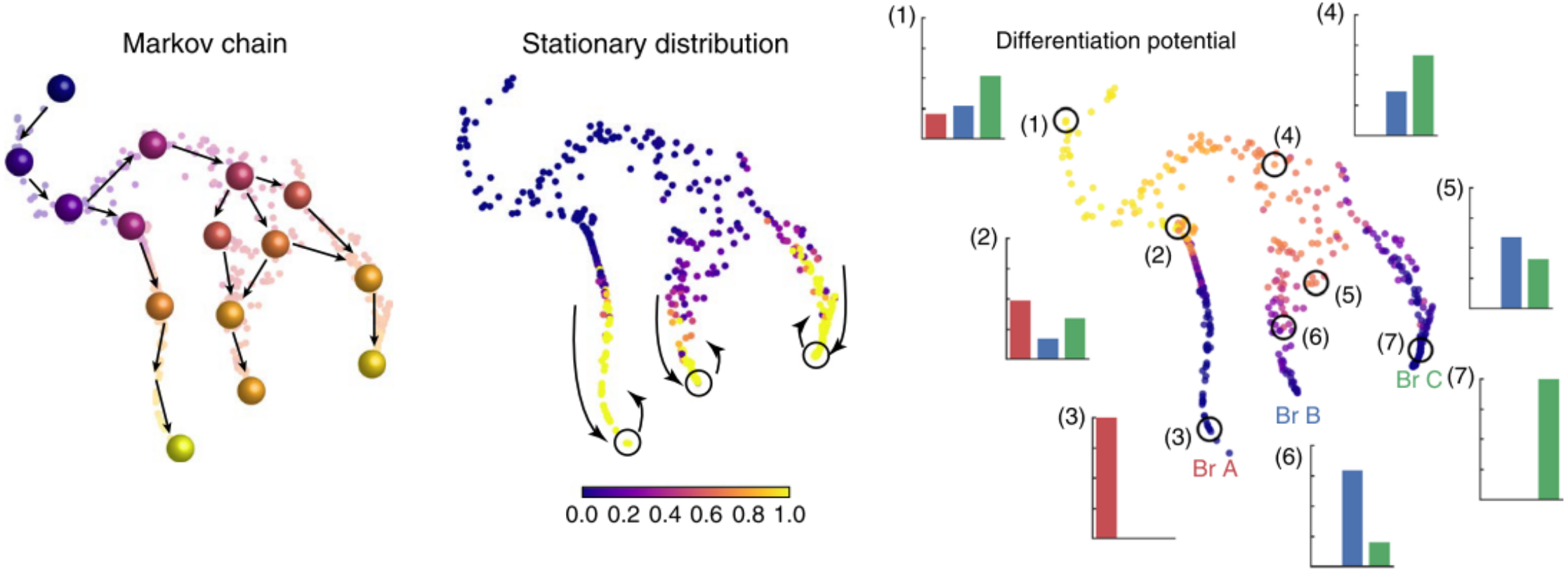
1. Строим граф ближайших соседей в пространстве diffusion map
2. Псевдовремя - расстояние по кратчайшему пути в этом графе от начальной клетки
   - Построение графа и подсчет расстояния с учетом общей топологии эмбеддинга
3. Выбираем набор конечных состояний и для каждой клетки считаем вероятности перехода в каждое из них



doi.org/10.1038/s41587-019-0068-4

Differentiation potential

Для каждой клетки можем подсчитать энтропию, соответствующую неопределенности судьбы клетки: чем выше differentiation potential, тем больше конечных состояний клетка может достичь



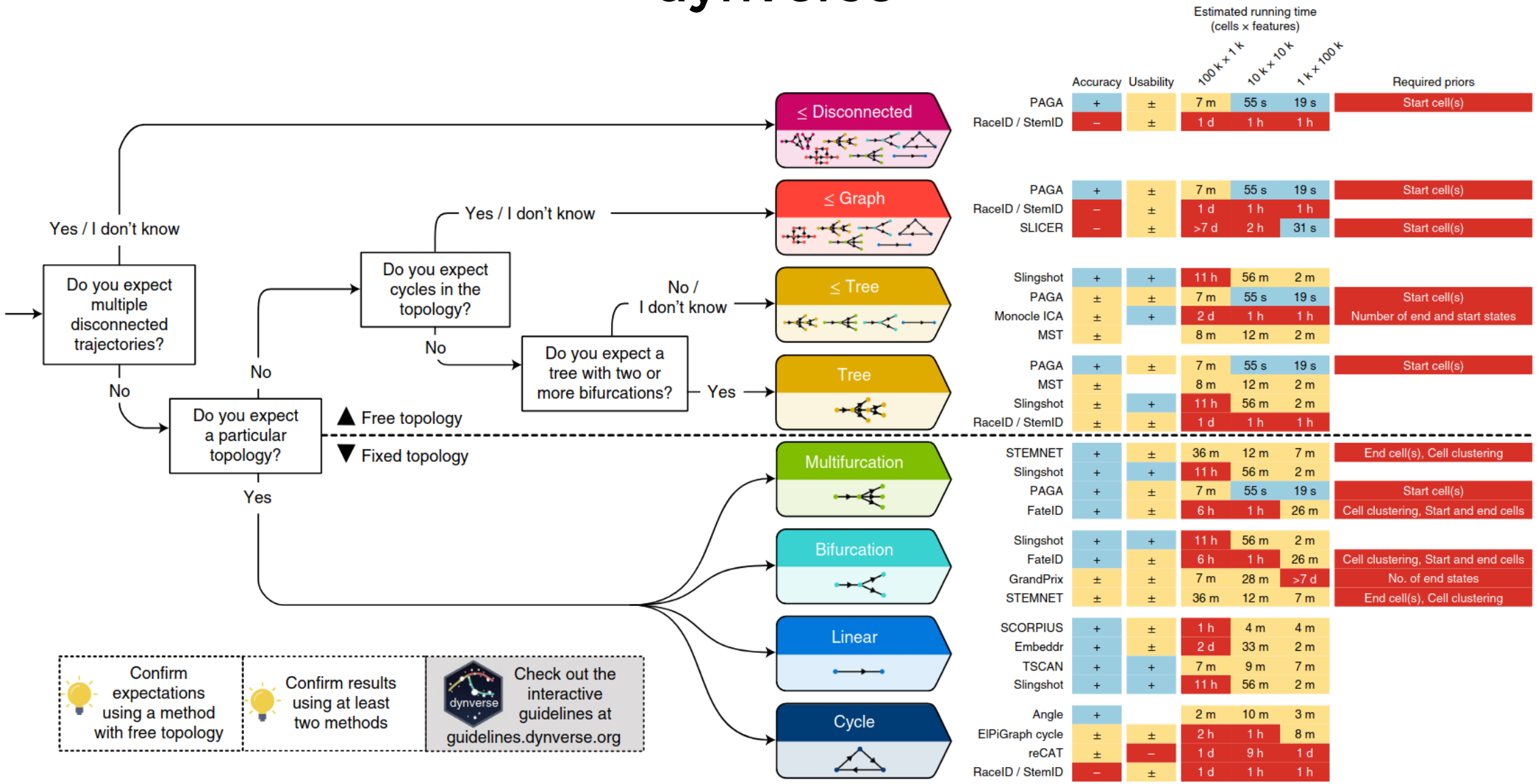
doi.org/10.1038/s41587-019-0068-4

И много других подходов

	Dimensionality reduction	Cluster based	Graph	Pseudotime calculation	Branching	Supervision
Diffusion Pseudotime	Diffusion maps	No	Weighted k-NN graph on cells	Transition probabilities over arbitrary length random walks	Yes	Starting cell
Embeddr	Laplacian eigenmaps	No	N/A	Principal curve, orthogonal projection	No	Path direction[1], subsetting[2]
Monocle	ICA	No	MST on cells	Diameter path, PQ trees	Yes[3]	Path direction[1], number of lineages
Monocle 2	Reversed graph embedding	No	Principal graph on cells	Distance to root	Yes	Starting cluster
TSCAN	PCA	Yes	MST on clusters	Cluster centers, orthogonal projection	Yes	Starting cluster
Waterfall	PCA	Yes	MST on clusters	Cluster centers, orthogonal projection	Yes[4]	Path direction[1]
Wishbone	Diffusion maps	No	Ensemble of k-NN graphs on cells	Distance refinement by waypoints	Yes[5]	Starting cell
Slingshot	Any	Yes	MST on clusters	Simultaneous principal curves, orthogonal projection	Yes	Starting cluster, end clusters (optional)

doi.org/10.1186/s12864-018-4772-0

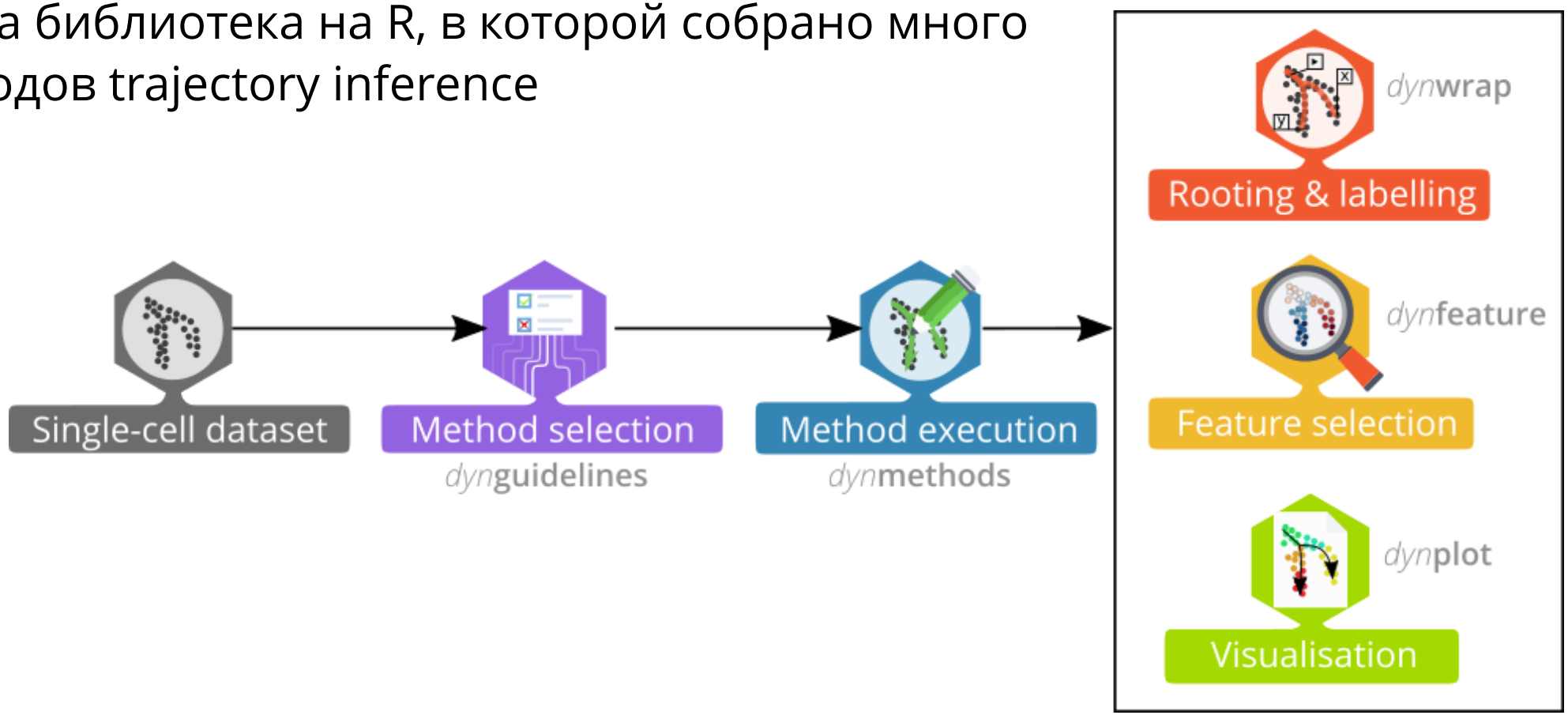
dynverse



https://doi.org/10.1038/s41587-019-0071-9

dynverse

Одна библиотека на R, в которой собрано много методов trajectory inference

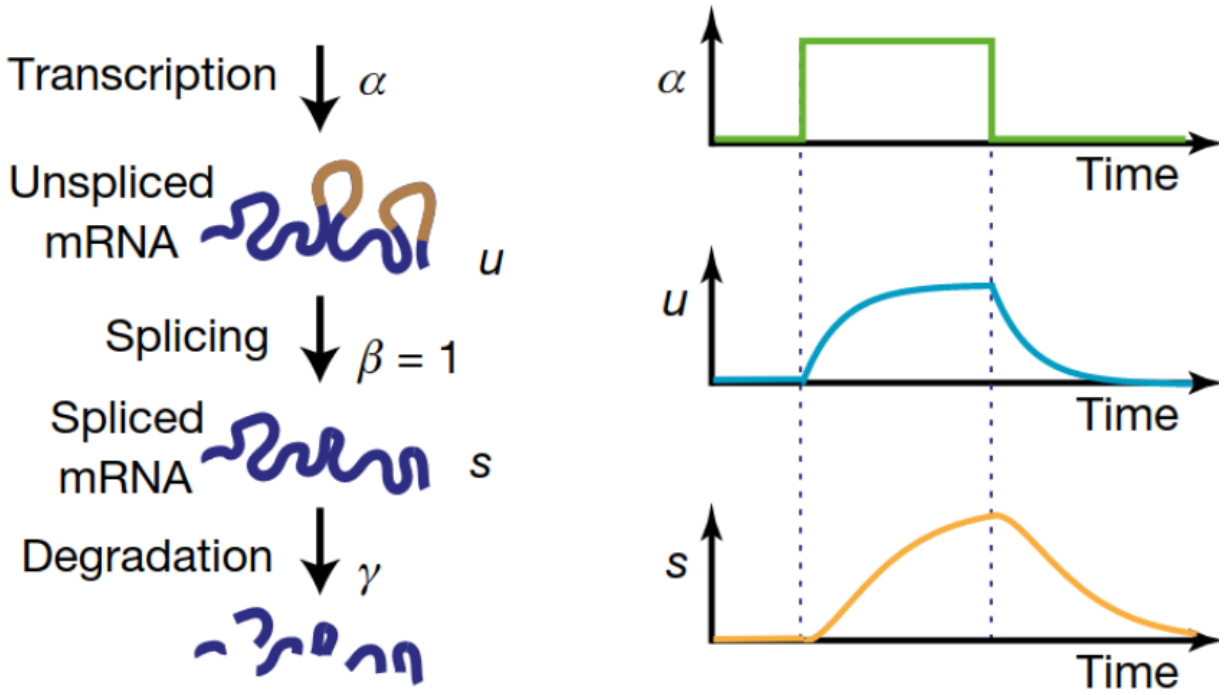


https://dynverse.org/

RNA velocity

По соотношению пре-РНК и зрелых РНК можем судить об изменениях в экспрессии

u - unspliced, s - spliced, α - transcription rate, β - splicing rate, γ - degradation rate



https://doi.org/10.1038/s41586-018-0414-6

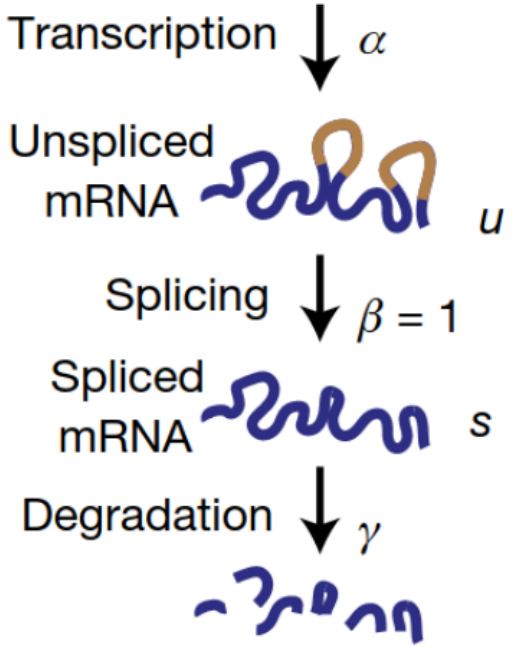
RNA velocity

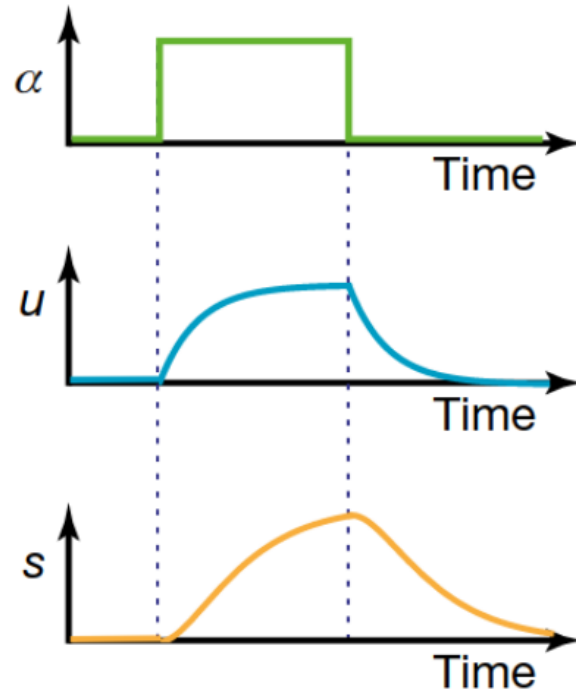
По соотношению пре-РНК и зрелых РНК можем судить об изменениях в экспрессии

u - unspliced, s - spliced, α - transcription rate, β - splicing rate, γ - degradation rate

$$\frac{du(t)}{dt} = \alpha(t) - \beta(t)u(t)$$

$$\frac{ds(t)}{dt} = \beta(t)u(t) - \gamma(t)s(t)$$





https://doi.org/10.1038/s41586-018-0414-6

RNA velocity

- По соотношению пре-РНК и зрелых РНК можем судить об изменениях в экспрессии
- u - unspliced, s - spliced, α - transcription rate, β - splicing rate, γ - degradation rate

$$\frac{du(t)}{dt} = \alpha(t) - \beta(t)u(t)$$

$$\frac{ds(t)}{dt} = \beta(t)u(t) - \gamma(t)s(t)$$

$$\longrightarrow$$

$$\frac{du(t)}{dt} = \alpha - u(t)$$

$$\frac{ds(t)}{dt} = u(t) - \gamma s(t)$$

RNA velocity

- По соотношению пре-РНК и зрелых РНК можем судить об изменениях в экспрессии
- u - unspliced, s - spliced, α - transcription rate, β - splicing rate, γ - degradation rate

$$\frac{du(t)}{dt} = \alpha(t) - \beta(t)u(t)$$

$$\frac{ds(t)}{dt} = \beta(t)u(t) - \gamma(t)s(t)$$

$$\longrightarrow$$

$$\frac{du(t)}{dt} = \alpha - u(t)$$

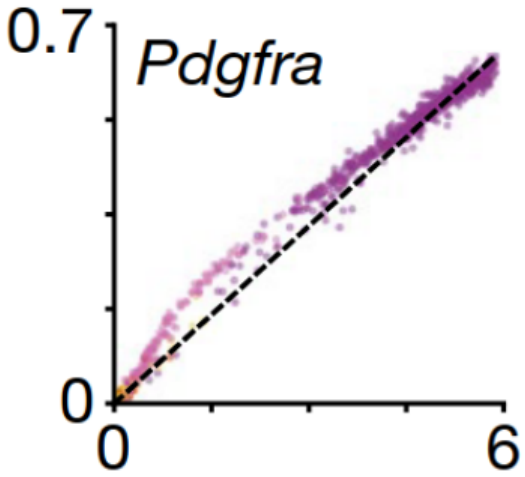
$$\boxed{\frac{ds(t)}{dt} = u(t) - \gamma s(t)}$$

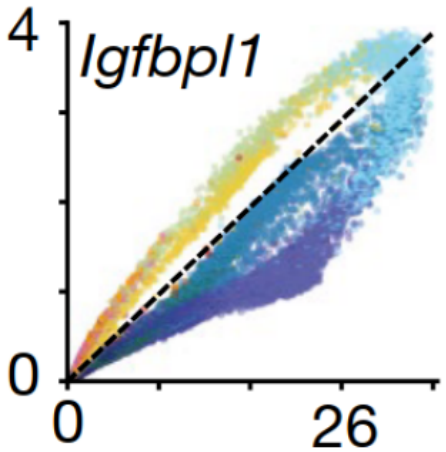
Это и назовем "RNA velocity"

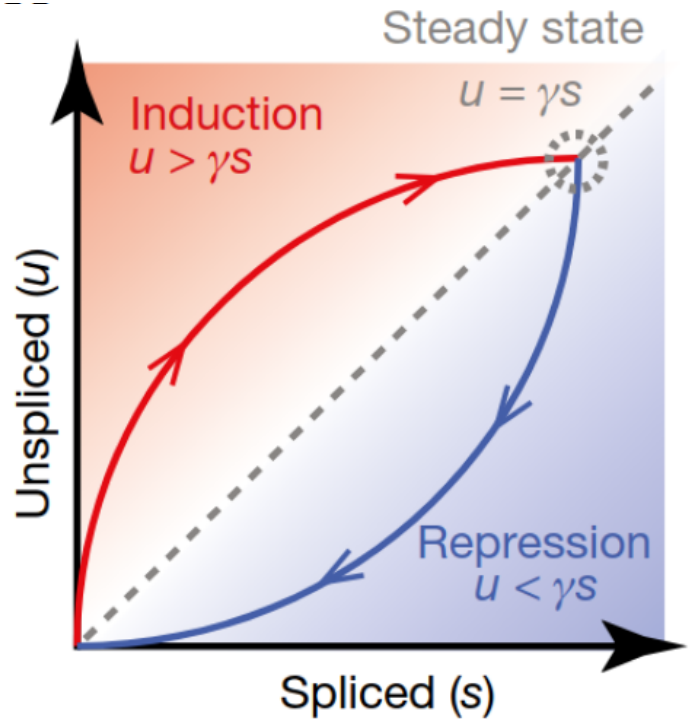
Если сможем оценить эту величину, получим возможность узнать, в какую сторону менялась экспрессия в клетке в момент эксперимента

"Фазовые портреты"

- Для любого гена можем построить график зависимости unspliced от spliced







- Предположим, что клетки крайних квантилей в стационарном состоянии, и оценим γ

Оценка параметров

- Если $ds/dt = 0$, то все просто:

$$\frac{du(t)}{dt} = \alpha - u(t)$$
$$\frac{ds(t)}{dt} = u(t) - \gamma s(t)$$

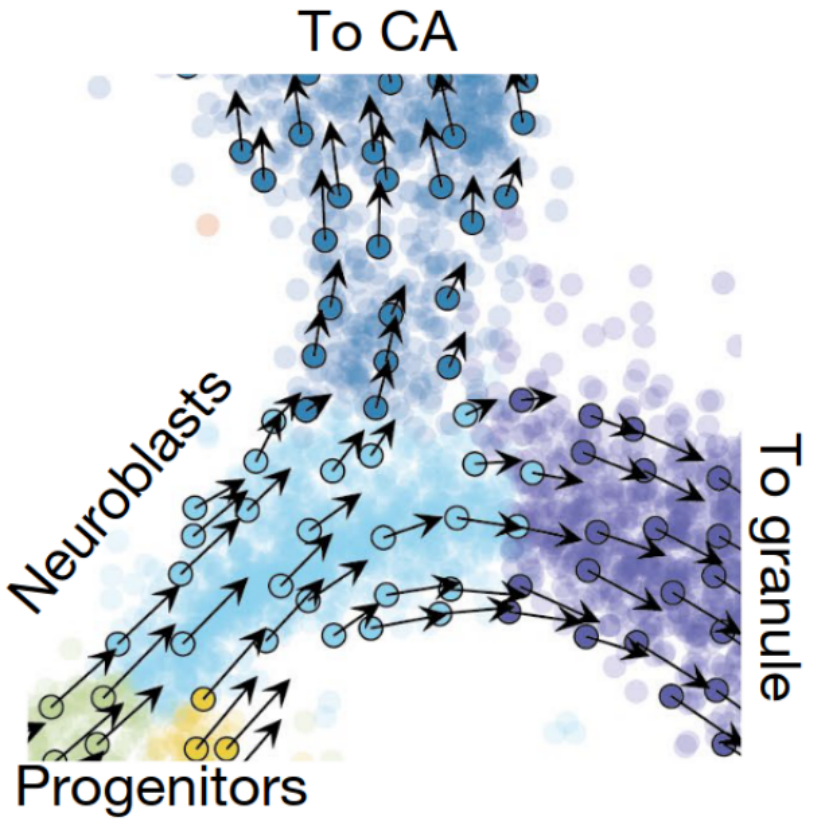
$$\longrightarrow$$

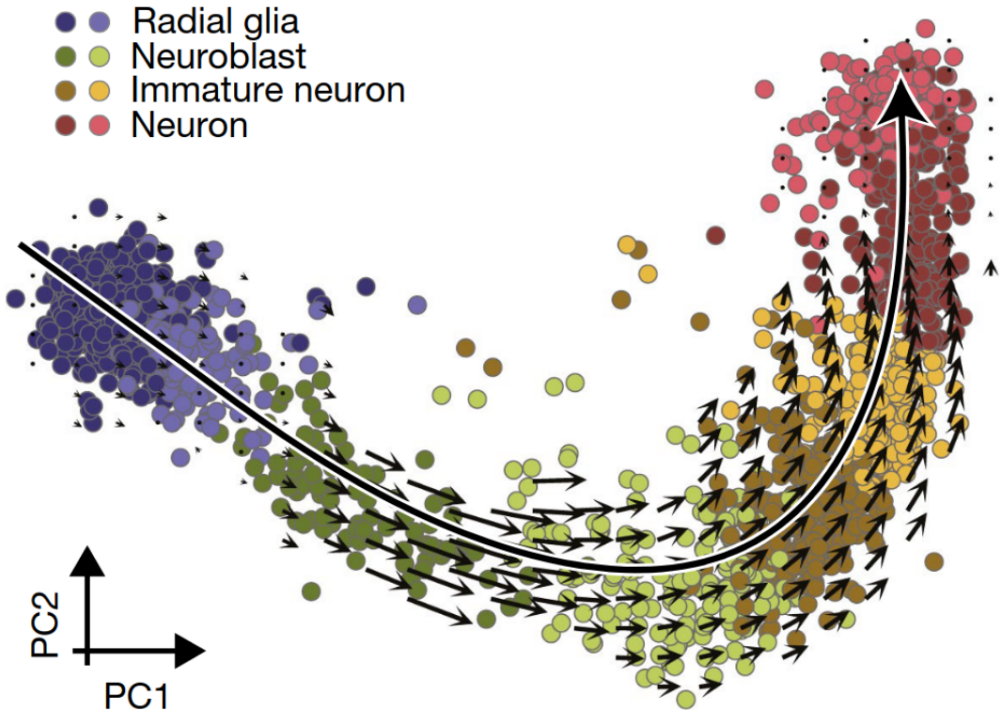
$$\gamma = \frac{u}{s}$$
$$\alpha = u$$

- Оценим γ исходя из крайних квантилей распределения клеток

Проекция на UMAP

Можем спроецировать векторы RNA velocity на UMAP





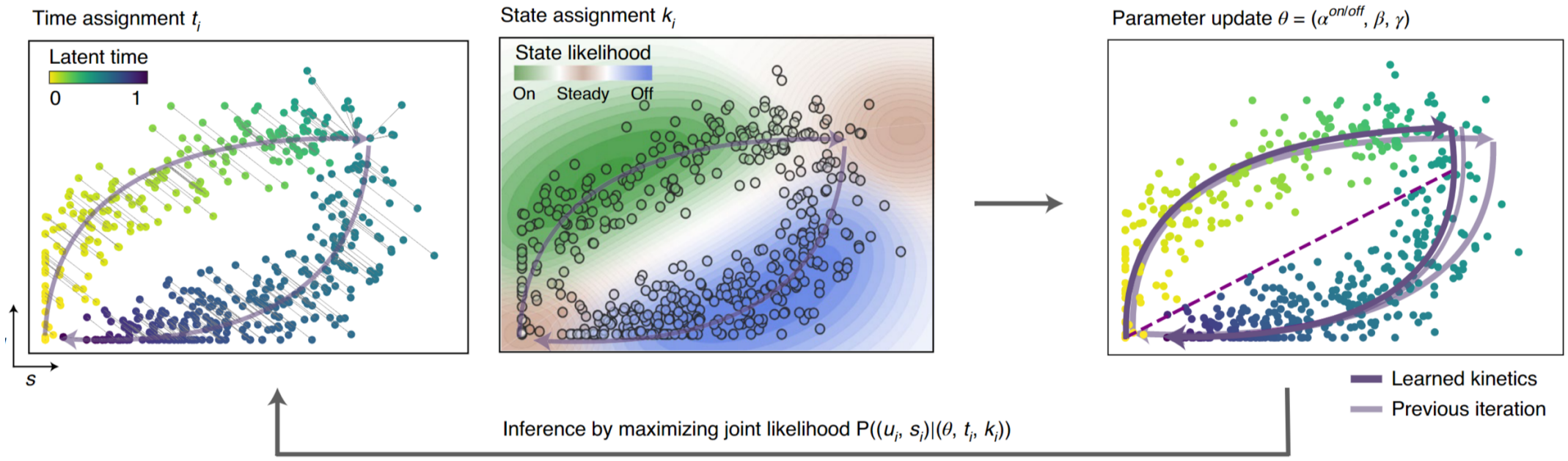
https://doi.org/10.1038/s41586-018-0414-6

Реализации

Первая библиотека - velocyto (La Manno et al., 2018)

Гораздо более удобная - scVelo (Bergen et al., 2020)

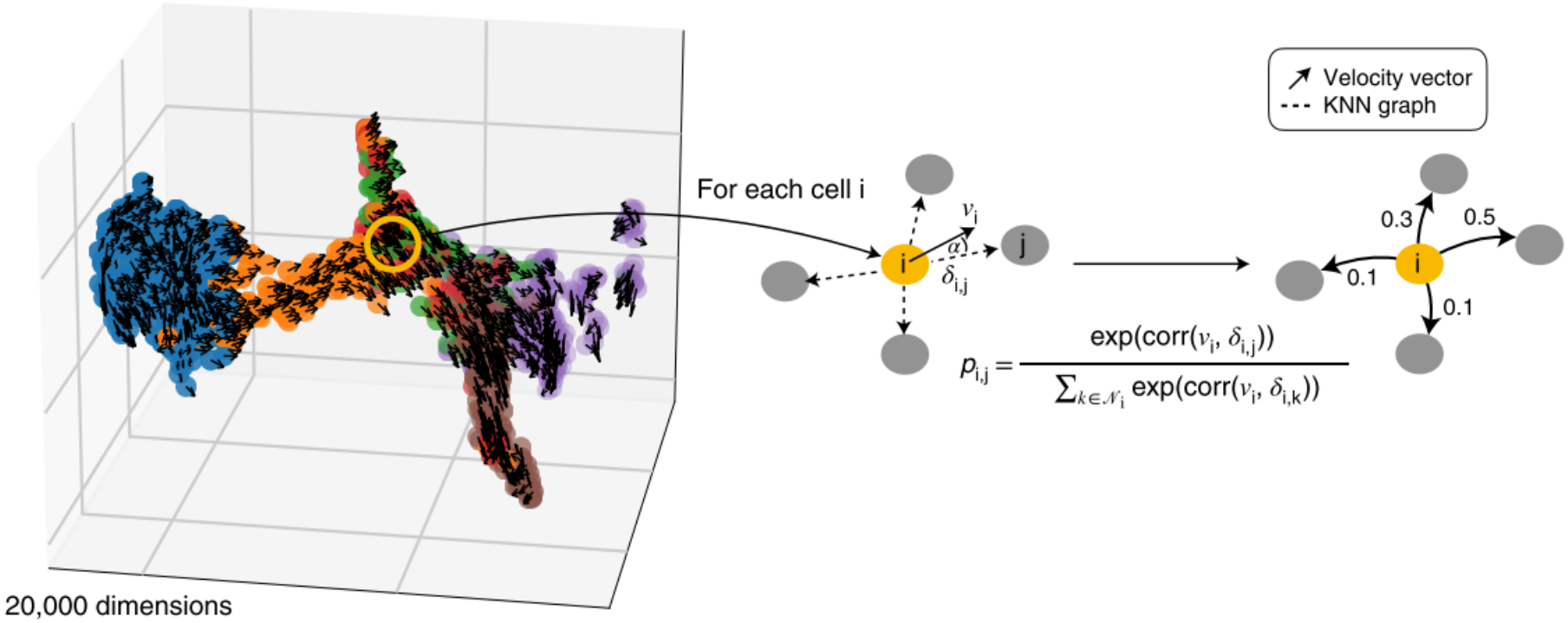
Кроме того, в модель добавлено латентное время и более сложная оценка параметров



https://doi.org/10.1038/s41587-020-0591-3

CellRank

Вычисляем вероятности перехода клетки в клетку, считая корреляцию между предсказанным RNA velocity изменением экспрессии и реально наблюдаемым

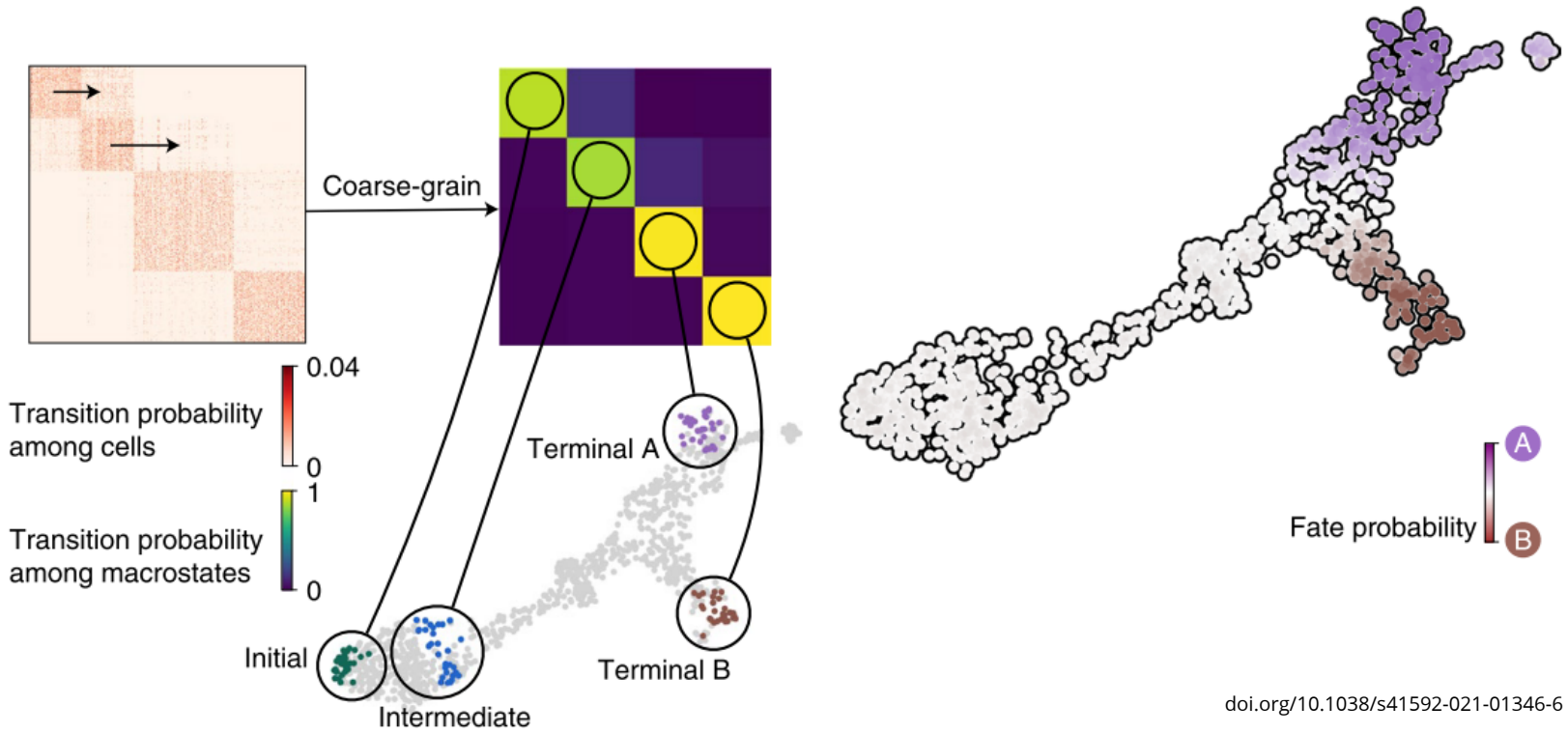


doi.org/10.1038/s41592-021-01346-6

CellRank

Задаем (или находим) начальные и терминальные макросостояния, пересчитываем Марковскую цепь с их учетом

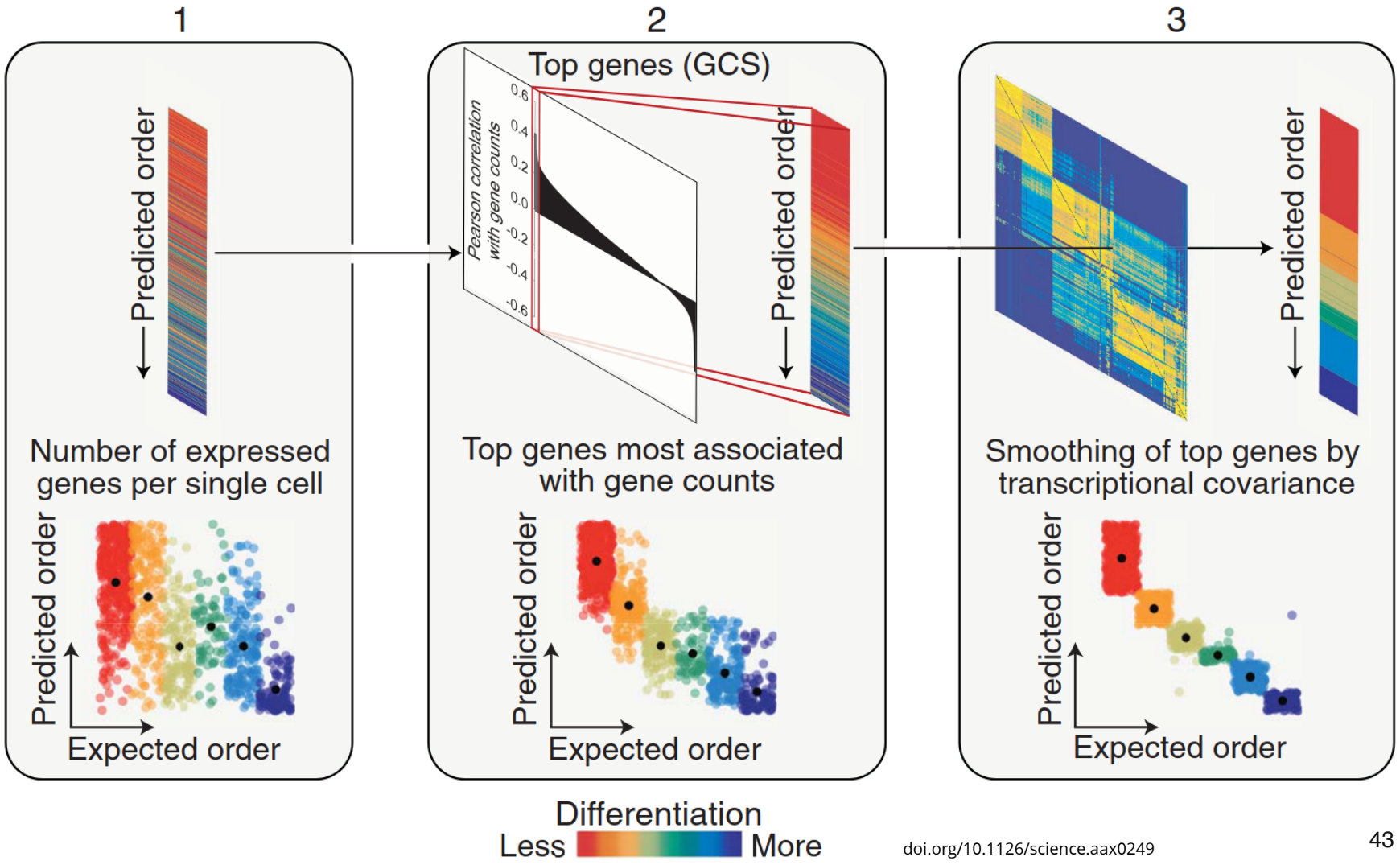
Для каждой клетки вычисляем вероятность дифференцировки в каждое из терминальных макросостояний



doi.org/10.1038/s41592-021-01346-6

CytoTRACE

Совсем другой подход: чем менее дифференцирована клетка, тем больше генов она экспрессирует



doi.org/10.1126/science.aax0249