



«Анализ транскриптомных данных»

Лекция #9.  
**Методы снижения размерности**

**Серёжа Исаев**

аспирант ФБМФ МФТИ  
аспирант MedUni Vienna

# Содержание курса

## 1. Bulk RNA-Seq:

- a. экспериментальные подходы,
- b. выравнивания и псевдовыравнивания,
- c. анализ дифференциальной экспрессии,
- d. функциональный анализ;

## 1. Single-cell RNA-Seq:

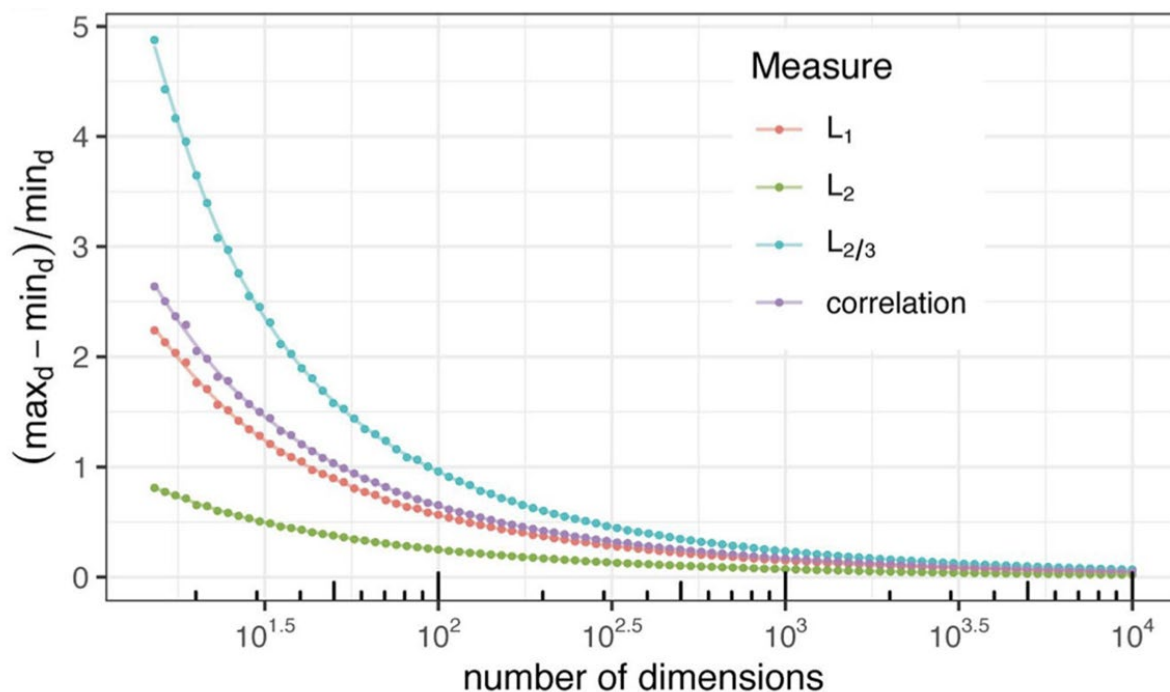
- a. экспериментальные подходы,
- b. отличия от процессинга bulk RNA-Seq,
- c. **методы снижения размерности,**
- d. кластера и траектории,
- e. мультимодальные омики одиночных клеток.

# Зачем нам снижение размерности?

1. В данных много шума а также автокорреляций, необходимо от них корректно избавиться
2. Сделать анализ более быстрым (например, в случае с кластеризацией)
3. Визуализировать данные
4. Избавиться от “проклятия размерности”

# Проклятие размерности

Если мы работаем с размерностями, превосходящими количество точек или даже сопоставимыми с ним (10000 клеток, 20000 генов), то в итоге расстояние между самыми далёкими точками становится сопоставимо расстоянию между самыми близкими точками



# Feature Selection

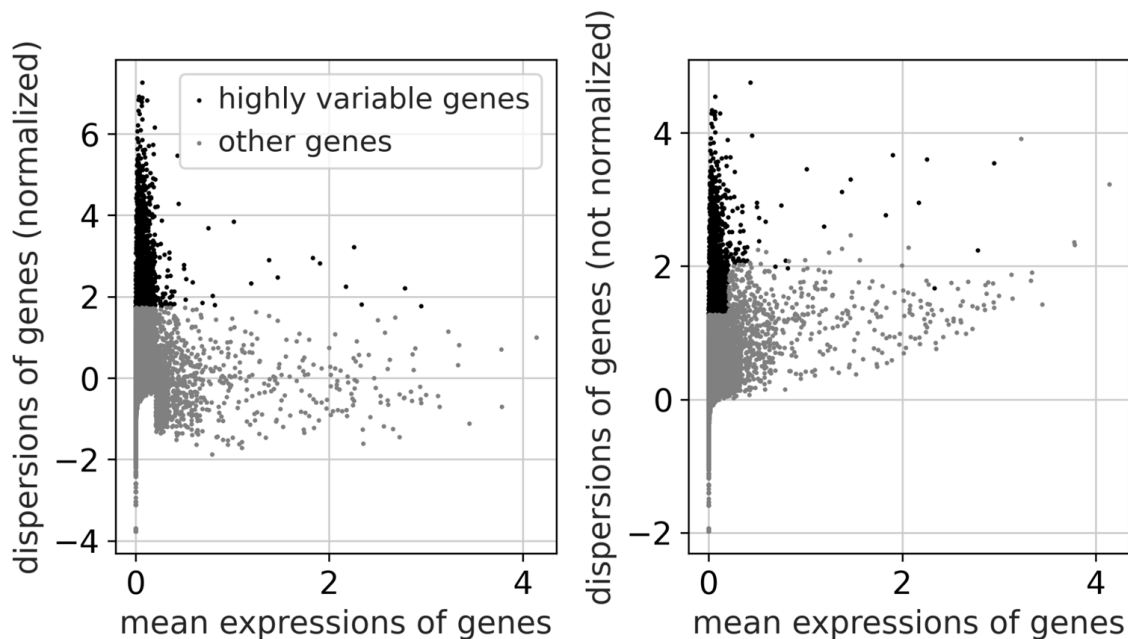
В классическом машинном обучении для того, чтобы предварительно снизить размерность данных и уменьшить количество параметров модели используется L1-регуляризация, или регрессия Лассо

Однако в scRNA-Seq нет переменной, которую мы хотим предугадать, поэтому в данном случае такой метод нам не подойдёт

# Highly Variable Genes (Seurat v. 2)

В таком случае гены упорядочиваются по средней экспрессии, дальше они делятся на N (default: 20) равных бина, а экспрессии внутри бина нормируются на среднюю дисперсию генов этого бина

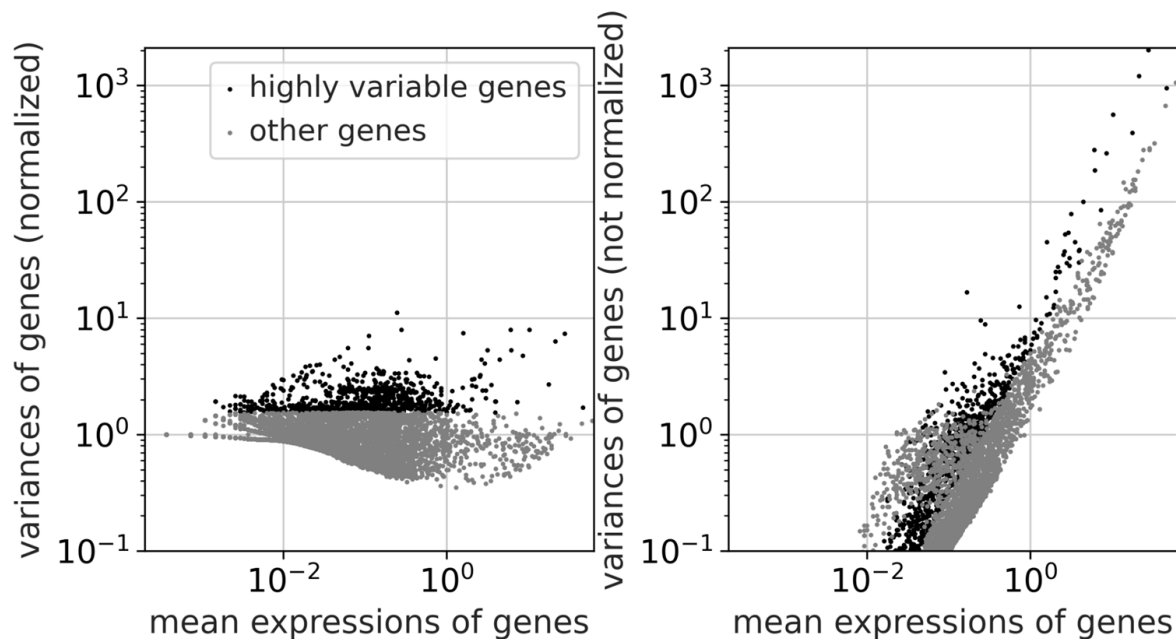
В итоге HVG выделяются по нормированной дисперсии



# Highly Variable Genes (Seurat v. 3)

В таком случае строится зависимость  $\log(\text{Var})$  от  $\log(E)$ , в которую фитится полином

На ожидаемое значение дисперсии в данной точке нормируется общая дисперсия, в итоге это значение используется для выбора HVG



# Методы снижения размерности

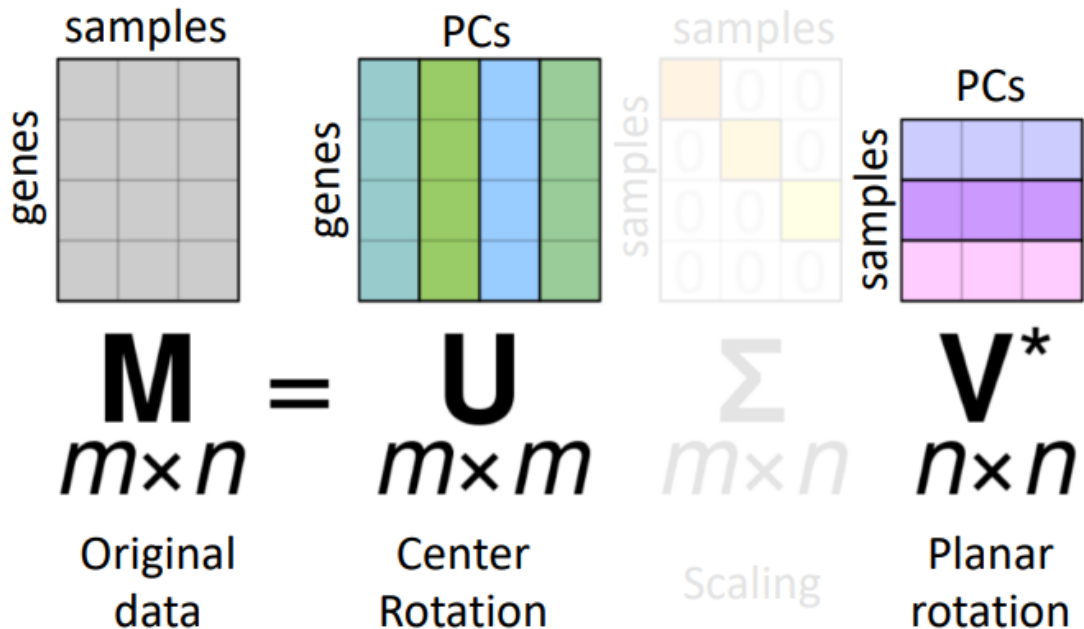
→ PCA	linear	Matrix Factorization		
ICA	linear	Matrix Factorization		
MDS	non-linear	Matrix Factorization		
Sparse NMF	non-linear	Matrix Factorization	2010	<a href="https://pdfs.semanticscholar.org/664d/40258f12ad28ed0b7d4c272935ad72a150db.pdf">https://pdfs.semanticscholar.org/664d/40258f12ad28ed0b7d4c272935ad72a150db.pdf</a>
cPCA	non-linear	Matrix Factorization	2018	<a href="https://doi.org/10.1038/s41467-018-04608-8">https://doi.org/10.1038/s41467-018-04608-8</a>
ZIFA	non-linear	Matrix Factorization	2015	<a href="https://doi.org/10.1186/s13059-015-0805-z">https://doi.org/10.1186/s13059-015-0805-z</a>
ZINB-WaVE	non-linear	Matrix Factorization	2018	<a href="https://doi.org/10.1038/s41467-017-02554-5">https://doi.org/10.1038/s41467-017-02554-5</a>
Diffusion maps	non-linear	graph-based	2005	<a href="https://doi.org/10.1073/pnas.0500334102">https://doi.org/10.1073/pnas.0500334102</a>
Isomap	non-linear	graph-based	2000	<a href="https://doi.org/10.1126/science.290.5500.2319">10.1126/science.290.5500.2319</a>
→ t-SNE	non-linear	graph-based	2008	<a href="https://lvdmaaten.github.io/publications/papers/JMLR_2008.pdf">https://lvdmaaten.github.io/publications/papers/JMLR_2008.pdf</a>
- BH t-SNE	non-linear	graph-based	2014	<a href="https://lvdmaaten.github.io/publications/papers/JMLR_2014.pdf">https://lvdmaaten.github.io/publications/papers/JMLR_2014.pdf</a>
- Flt-SNE	non-linear	graph-based	2017	<a href="https://arxiv.org/abs/1712.09005">arXiv:1712.09005</a>
LargeVis	non-linear	graph-based	2018	<a href="https://arxiv.org/abs/1602.00370">arXiv:1602.00370</a>
→ UMAP	non-linear	graph-based	2018	<a href="https://arxiv.org/abs/1802.03426">arXiv:1802.03426</a>
PHATE	non-linear	graph-based	2017	<a href="https://www.biorxiv.org/content/biorxiv/early/2018/06/28/120378.full.pdf">https://www.biorxiv.org/content/biorxiv/early/2018/06/28/120378.full.pdf</a>
scvis	non-linear	Autoencoder (MF)	2018	<a href="https://doi.org/10.1038/s41467-018-04368-5">https://doi.org/10.1038/s41467-018-04368-5</a>
VASC	non-linear	Autoencoder (MF)	2018	<a href="https://doi.org/10.1016/j.gpb.2018.08.003">https://doi.org/10.1016/j.gpb.2018.08.003</a>



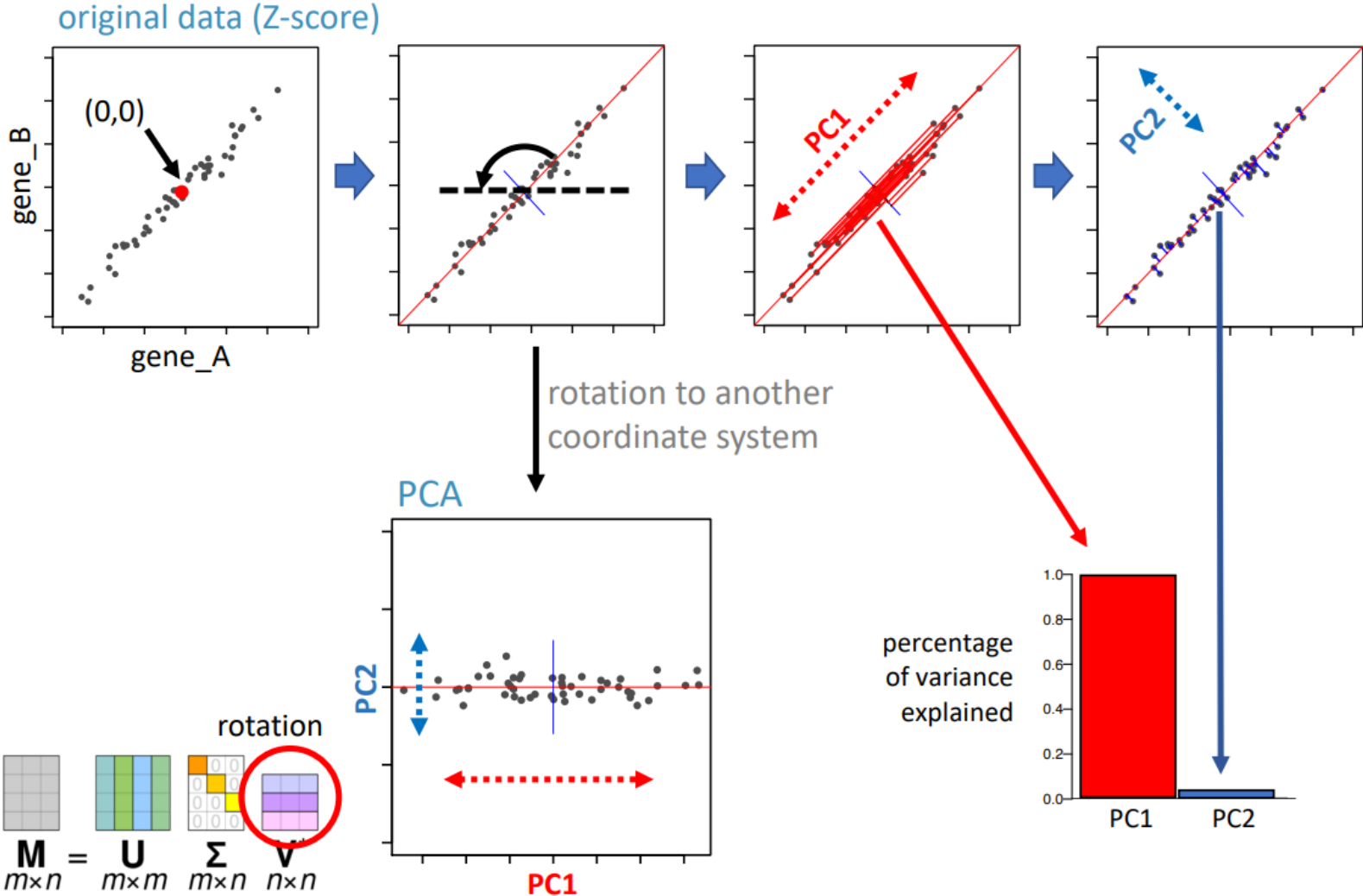
# Principal Component Analysis (PCA)

Это метод линейного снижения размерности (то есть каждая координата в пространстве сниженной размерности — это линейная комбинация координат пространства высокой размерности)

Частный случай SVD (Singular Value Decomposition)



# Как PCA работает?

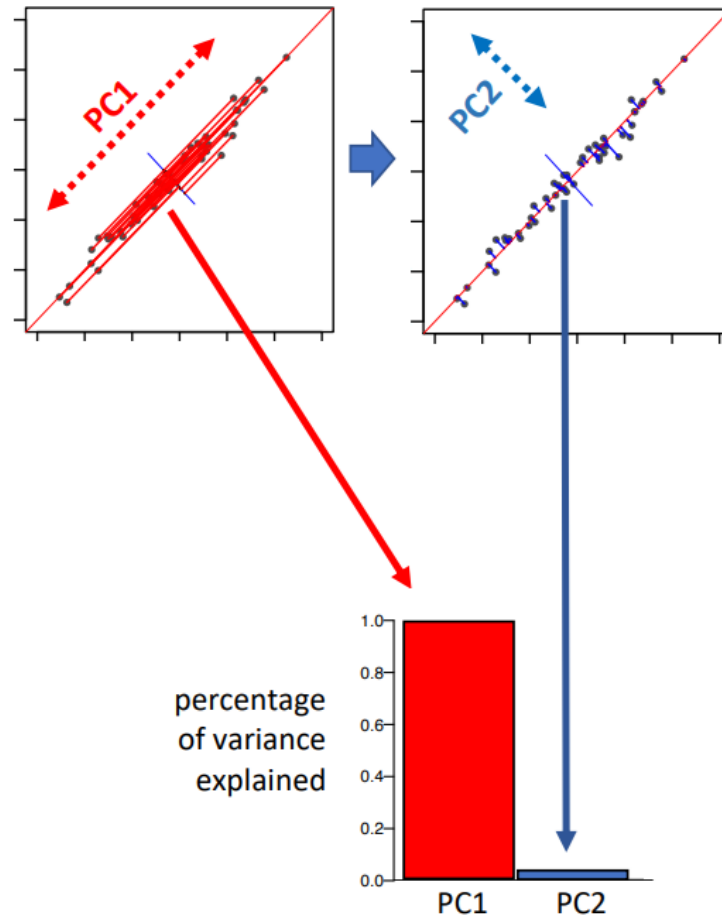
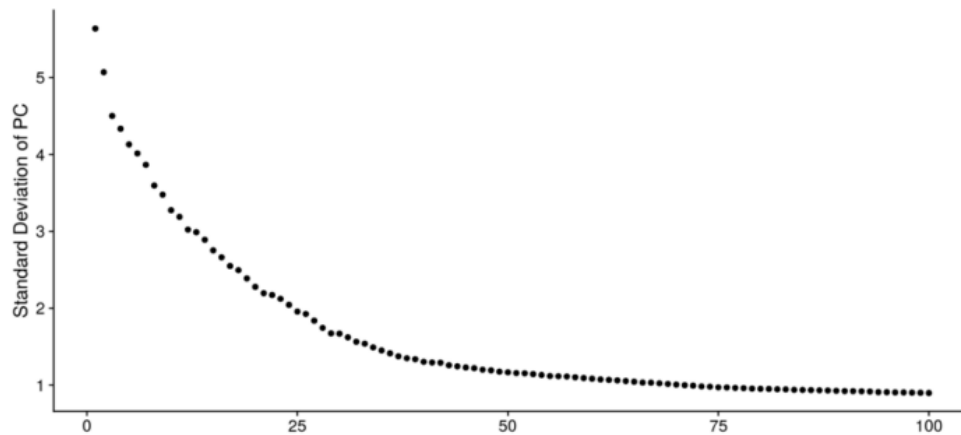


# Как PCA работает?

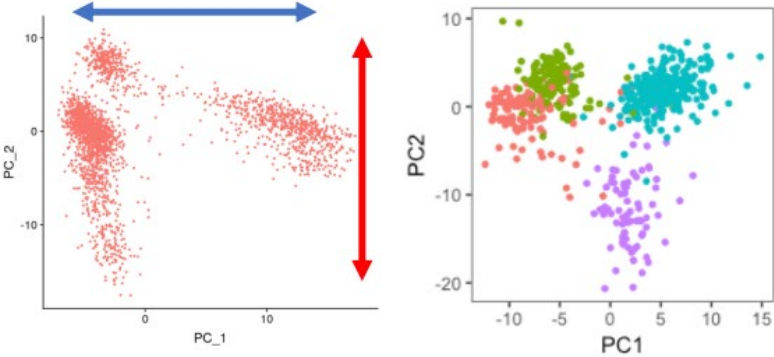
В нашем случае первая главная компонента описывает  $>98\%$  дисперсии данных

Вторая компонента незначима

В настоящей же жизни всё гораздо сложнее:

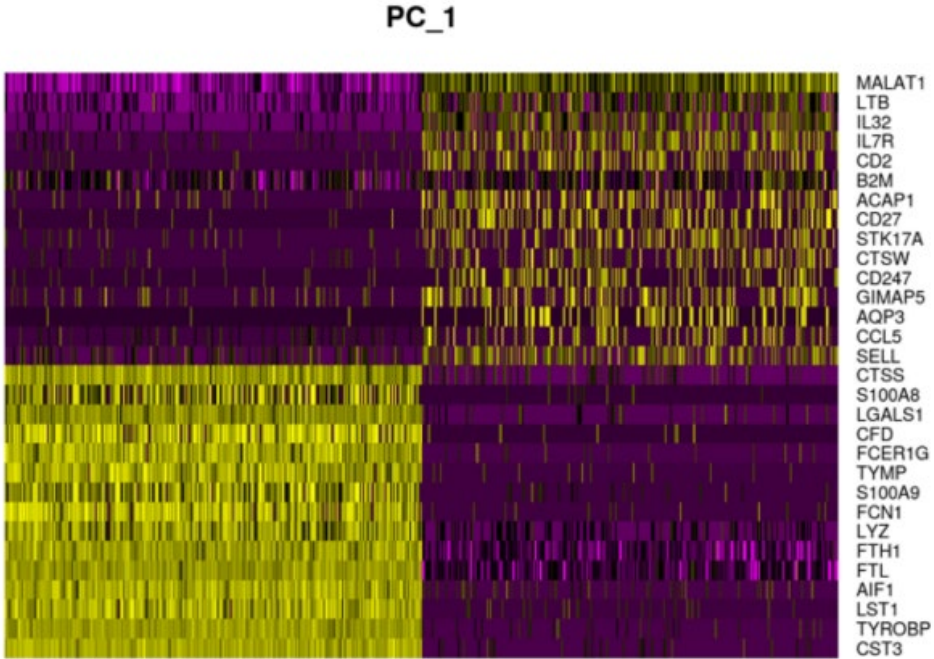
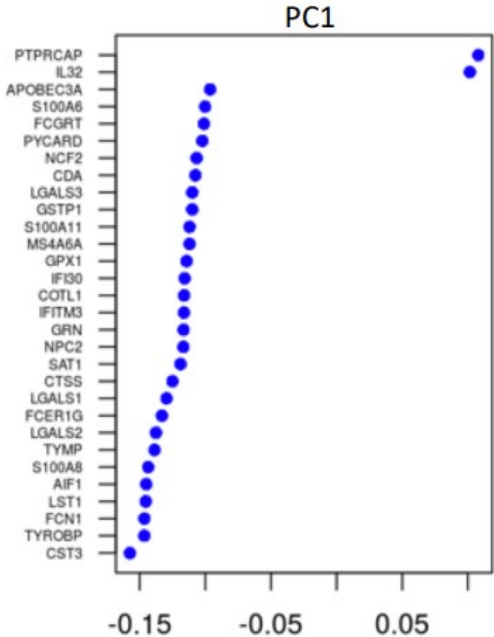


# PCA в scRNA-Seq



- ILC1
- ILC2
- ILC3
- NK

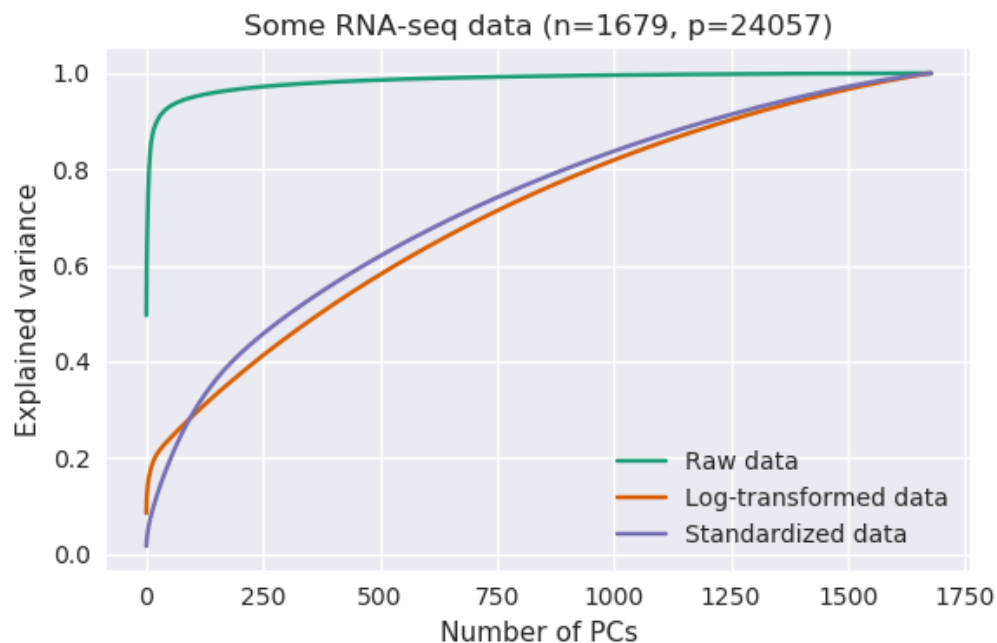
PC1 и PC2 обычно коррелируют с глубиной секвенирования клетки и гетерогенностью ткани



# Подготовка данных к PCA

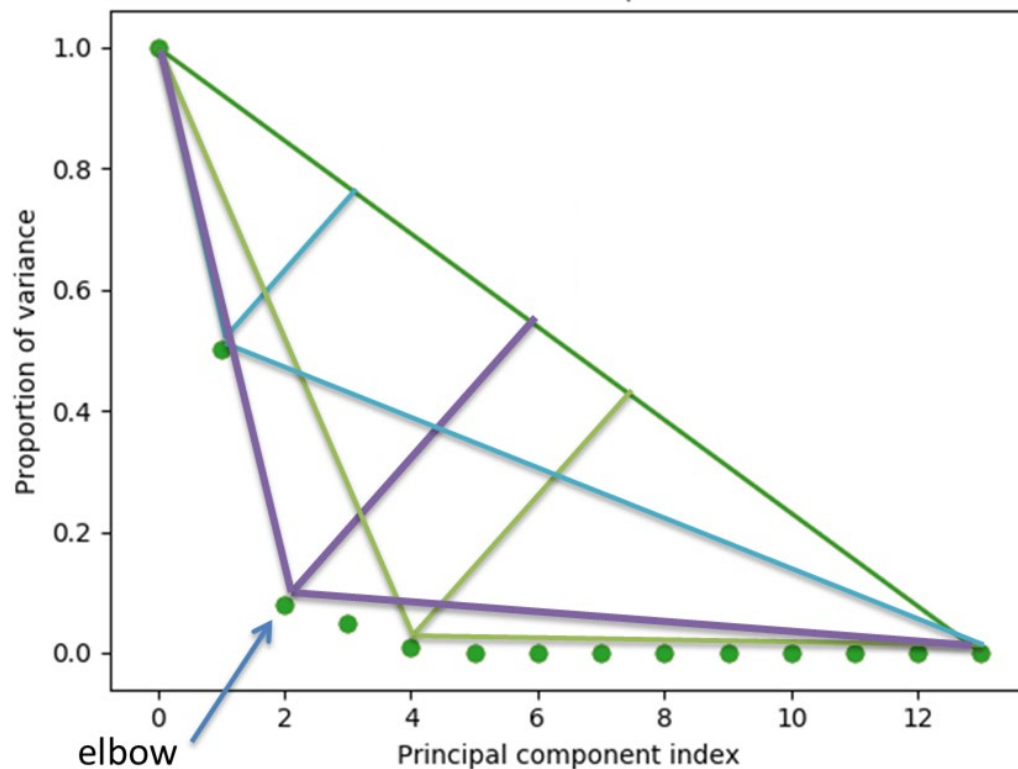
Задача PCA — описать дисперсию данных. Если у вас есть очень высоко дисперсные переменные, то PCA будет стремиться описать именно их

Именно поэтому нам был необходим шаг контроля за дисперсией



# Сколько компонент использовать в анализе?

Можно воспользоваться “методом локтя” (см. иллюстрацию ниже), однако на практике берут первые 20-30 компонент



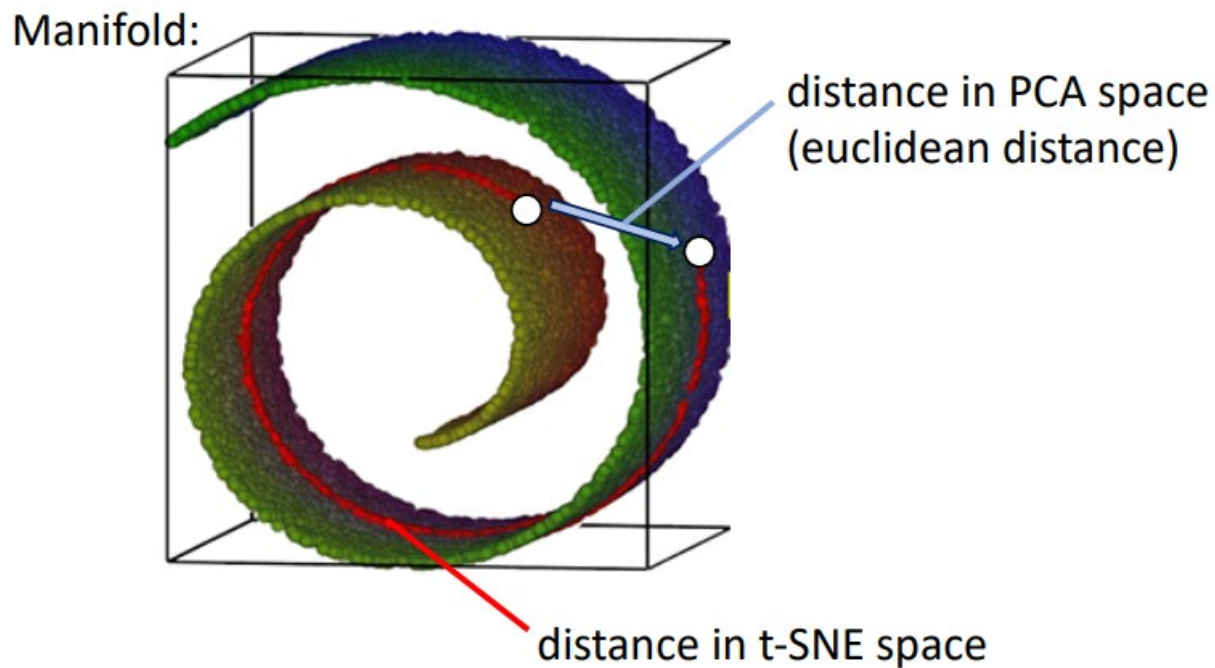
# РСА: выводы

- Это линейный метод уменьшения размерности. Координаты имеют физический смысл, их можно интерпретировать, расстояния имеют физический смысл
- Данные, как правило, шкалируются перед тем, как выполнить РСА (нужно ли шкалировать Pearson Residuals?)
- Топ первых главных компонент описывает почти всю дисперсию в данных

# t-SNE

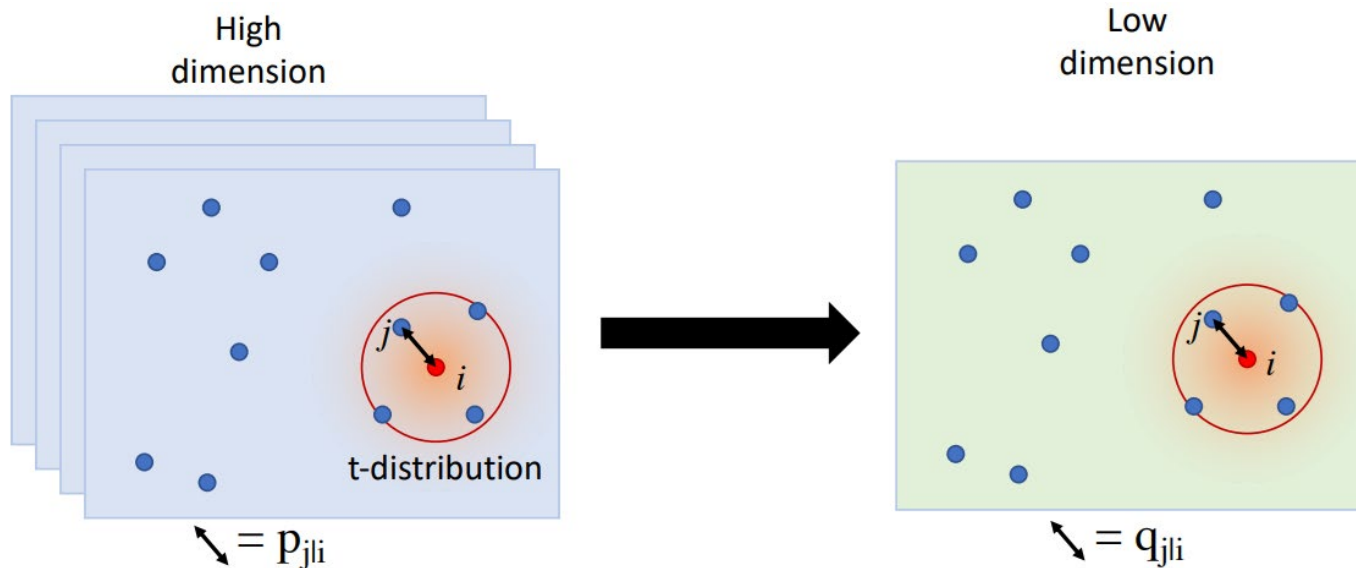
t-SNE — это **нелинейный** метод снижения размерности

Основная цель — сохранить локальное соседство каждой клетки при переходе из многомерного пространства в дву- или трёхмерное

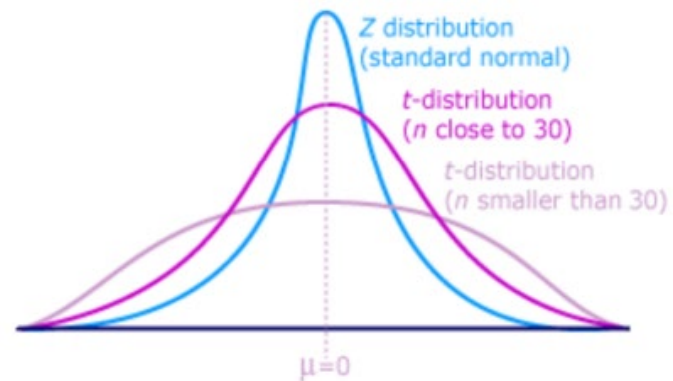




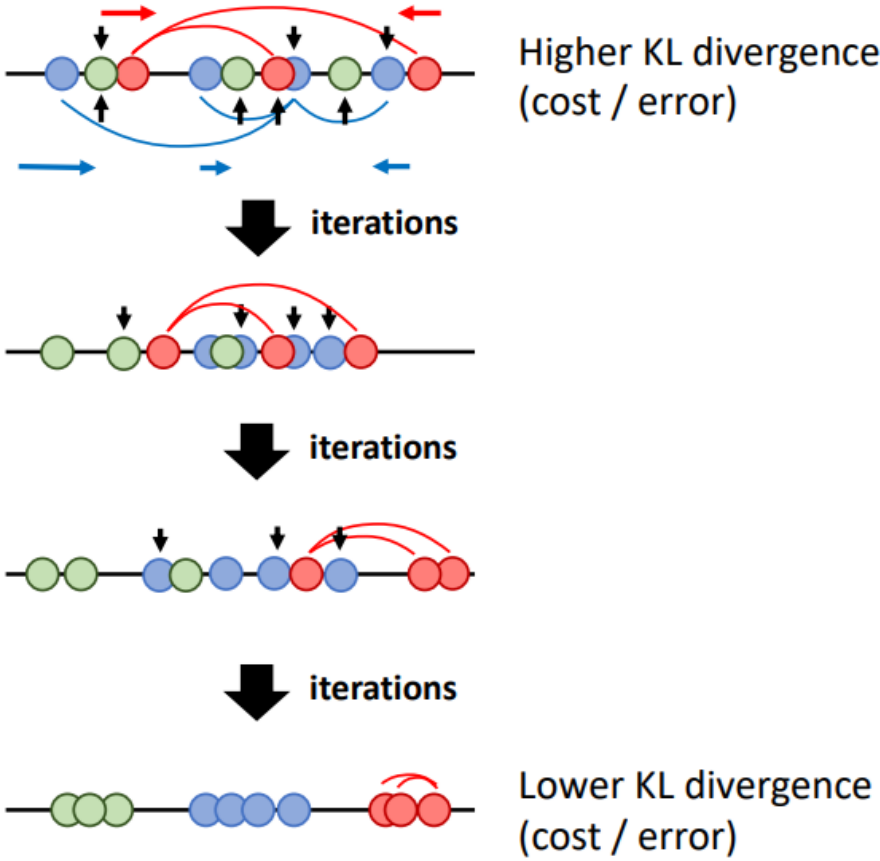
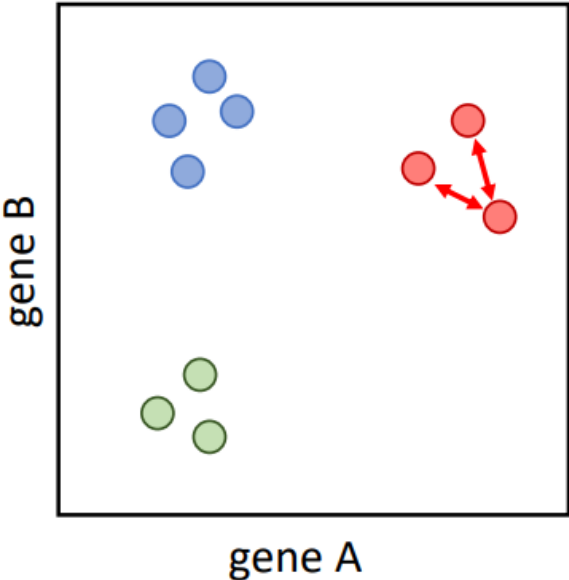
# Как работает t-SNE?



$p(j | i)$  и  $q(j | i)$  измеряют условную вероятность, что точка  $i$  будет иметь точку  $j$  в качестве своего ближайшего соседа



# Как работает t-SNE?



# Гиперпараметры t-SNE

t-SNE, как правило, строится из представления данных в пространстве PCA (то есть не с сырых экспрессий!)

Основные гиперпараметры:

1. Perplexity
2. Число итераций
3. Learning rate

Также играет роль, какое количество главных компонент вы взяли для t-SNE

<https://distill.pub/2016/misread-tsne/>

# Важные пункты насчёт t-SNE

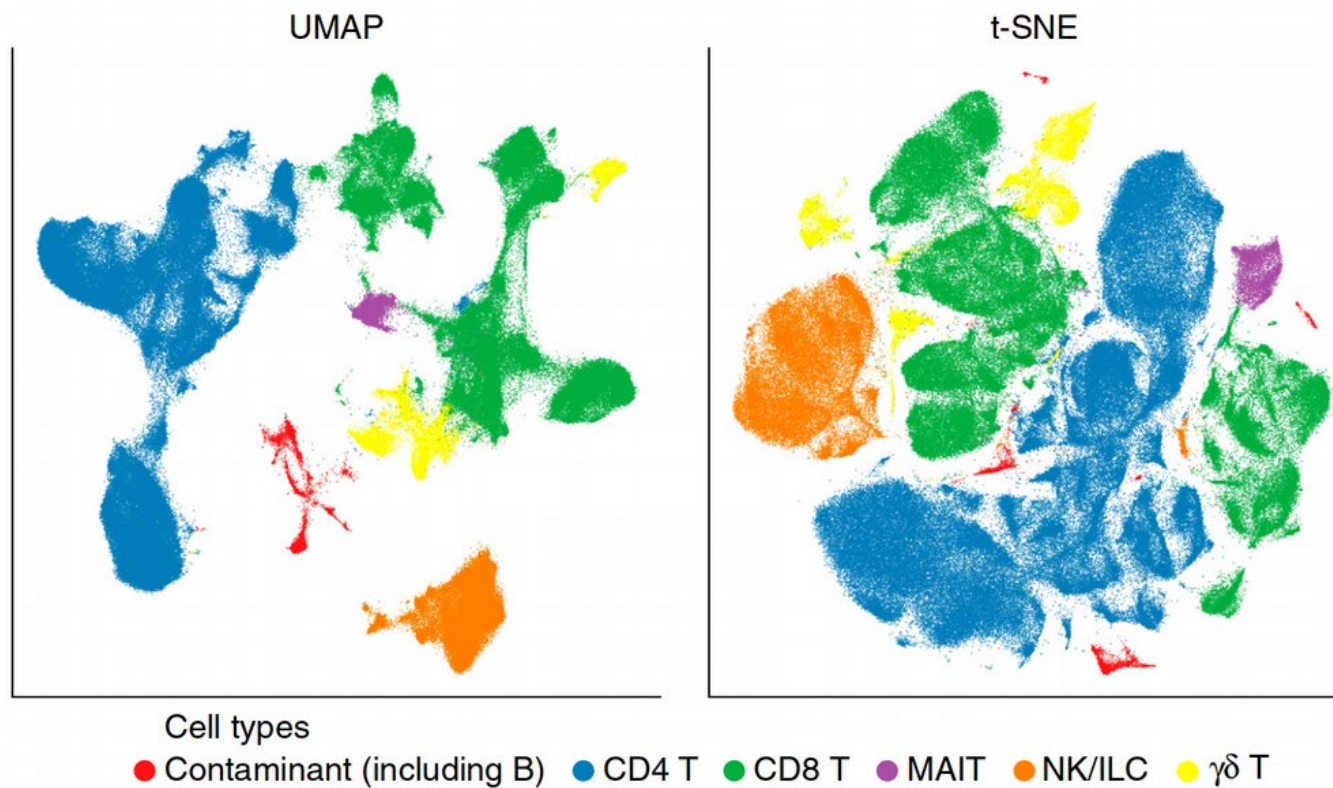
В отличие от PCA, это стохастический алгоритм, поэтому он может продуцировать разные результаты в зависимости от того, какой вы используете seed

Функция потерь t-SNE оптимизирует локальные расстояния, а не глобальные (из-за особенности t-распределения). Взаимное расположение кластеров друг относительно друга не имеет физического смысла

Чтобы добавить новые точки на представление, необходимо перезапускать алгоритм (исключение: пакет openTSNE)

# UMAP

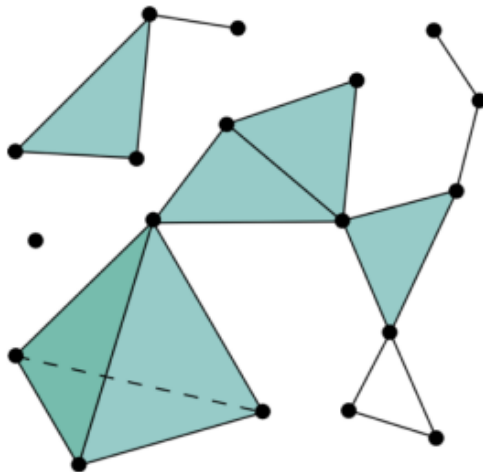
UMAP — это алгоритм, концептуально очень похожий на tSNE, однако работающий по немного другим принципам



# Как работает UMAP?

Алгоритм основан на топологических структурах в пространстве высокой размерности. Клетки, находящиеся на расстояниях меньше определённого порога ( $\text{min\_dist}$ ), объединяются рёбрами и образуют симплексы

Дальше UMAP оптимизирует функциональную зависимость отображения данных из многомерного пространства в маломерное (обычно двухмерное), инициация происходит не случайно, а на собственных значениях матрицы Лапласа графа симплексов



# UMAP: число компонент из PCA

5 PCs



10 PCs



20 PCs



30 PCs



50 PCs

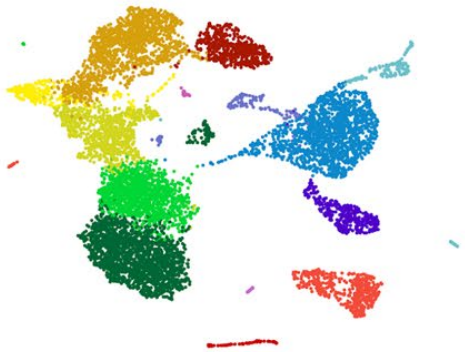


All data



# UMAP: число соседей в графе

5 neighbors



10 neighbors



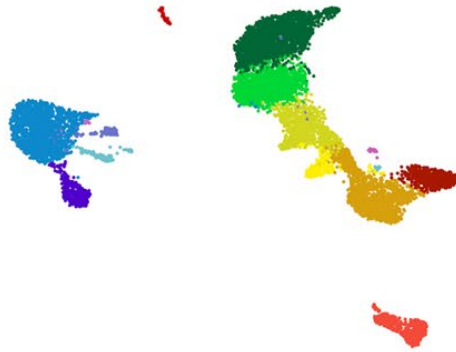
50 neighbors



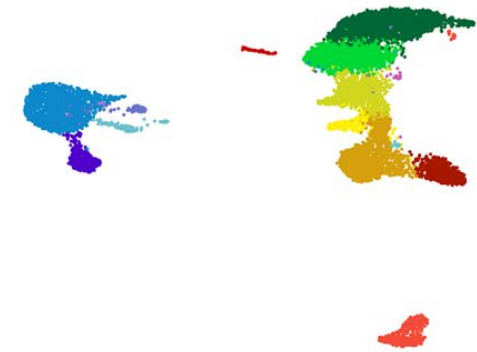
100 neighbors



200 neighbors



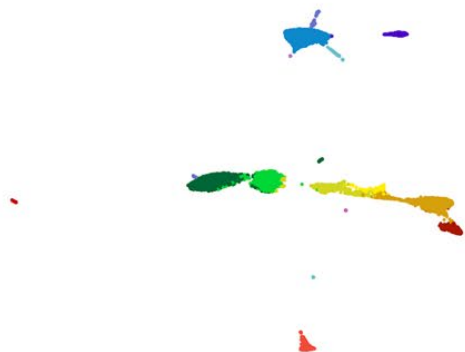
500 neighbors



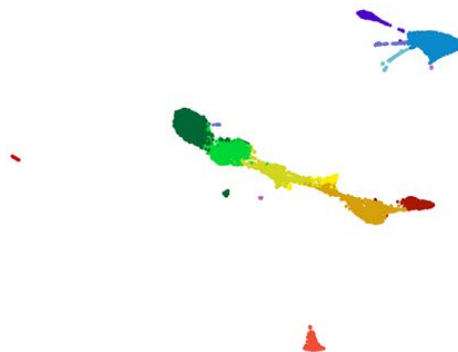


# UMAP: минимальное расстояние для симплекса

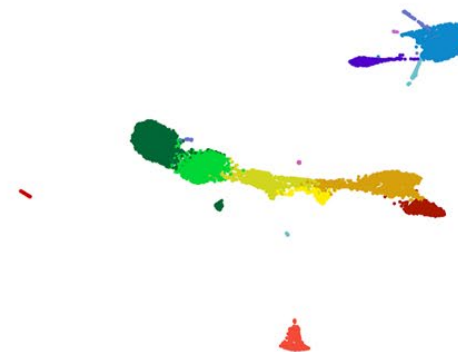
min\_dist = 0.01



min\_dist = 0.1



min\_dist = 0.2



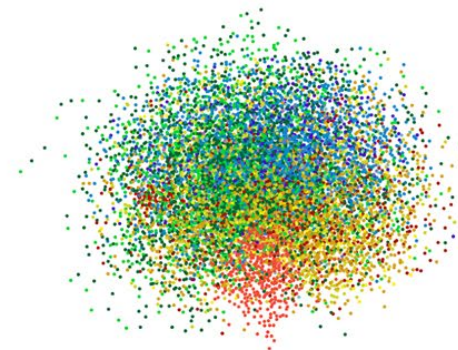
min\_dist = 0.5



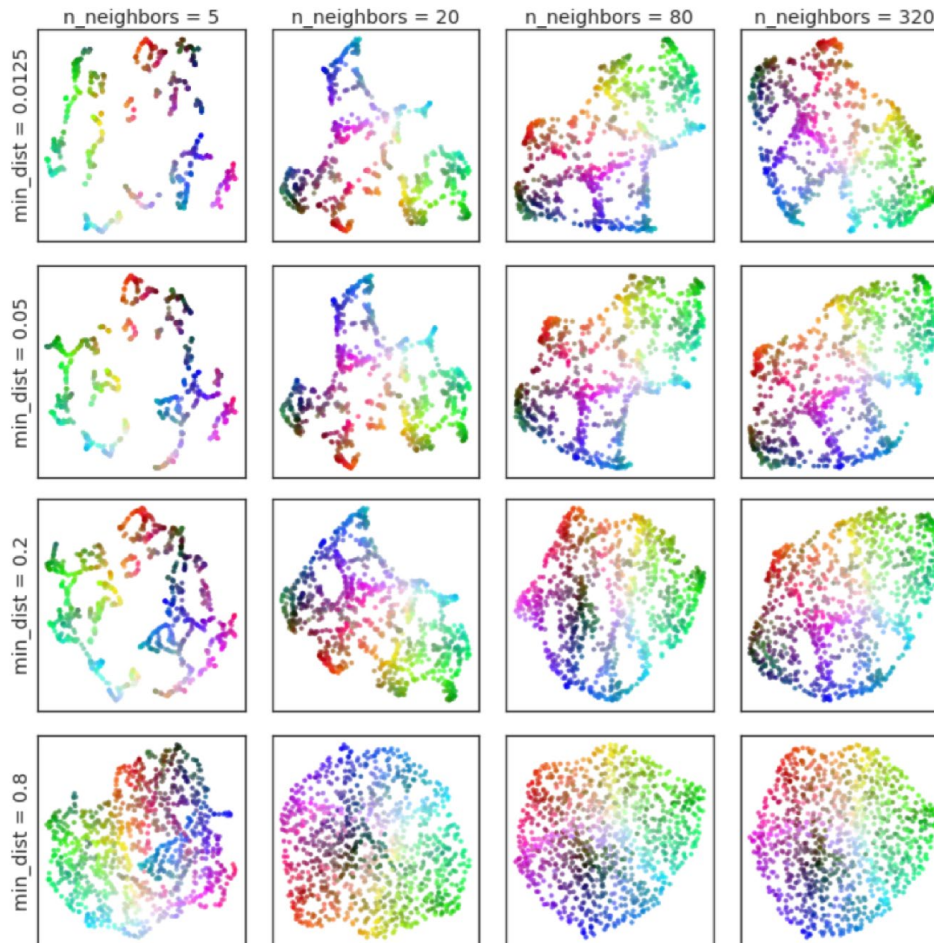
min\_dist = 1



min\_dist = 5



# UMAP может находить структуры в шуме



# UMAP лучше описывает глобальную структуру

Published: 03 December 2018

## Dimensionality reduction for visualizing single-cell data using UMAP

Etienne Becht, Leland McInnes, John Healy, Charles-Antoine Dutertre, Immanuel W H Kwok, Lai Guan Ng, Florent Gehoux & Evan W Newell 

*Nature Biotechnology* **37**, 38–44(2019) | [Cite this article](#)

**47k** Accesses | **468** Citations | **274** Altmetric | [Metrics](#)

*A new algorithm, called uniform manifold approximation and projection (UMAP) has been recently published and is claimed to preserve as much of the local and more of the global data structure than t-SNE, with a shorter run time.*

# На самом деле нет

Contradictory Results

[Comment on this paper](#)

## UMAP does not preserve global structure any better than t-SNE when using the same initialization

 Dmitry Kobak, George C. Linderman

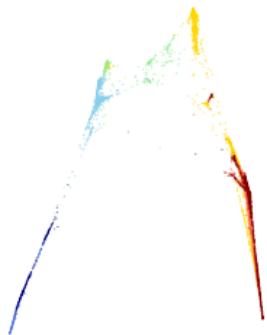
doi: <https://doi.org/10.1101/2019.12.19.877522>

This article is a preprint and has not been certified by peer review [what does this mean?].

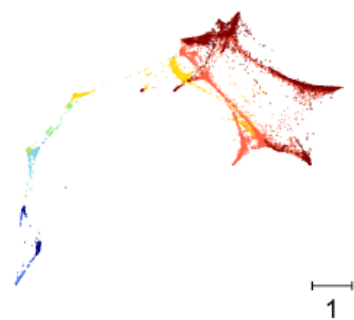
*We show that UMAP with random initialization preserves global structure as poorly as t-SNE with random initialization while t-SNE with informative initialization performs as well as UMAP with informative initialization. Hence, contrary to the claims of Becht et al., their experiments do not demonstrate any advantage of the UMAP algorithm per se, but rather warn against using random initialization.*

# Оптимизация лосса t-SNE, exaggeration rate

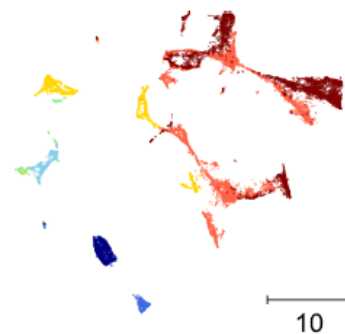
**a** Laplacian Eigenmaps



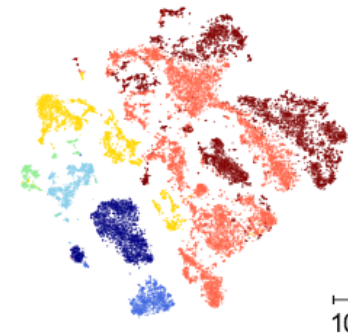
**b**  $\rho = 30$



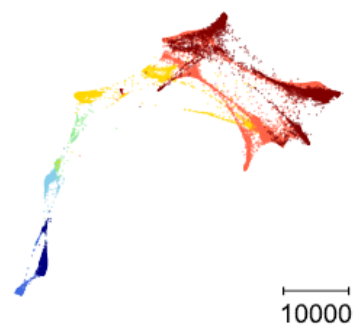
**c**  $\rho = 4$



**d** t-SNE



**e** ForceAtlas2



**f** UMAP

