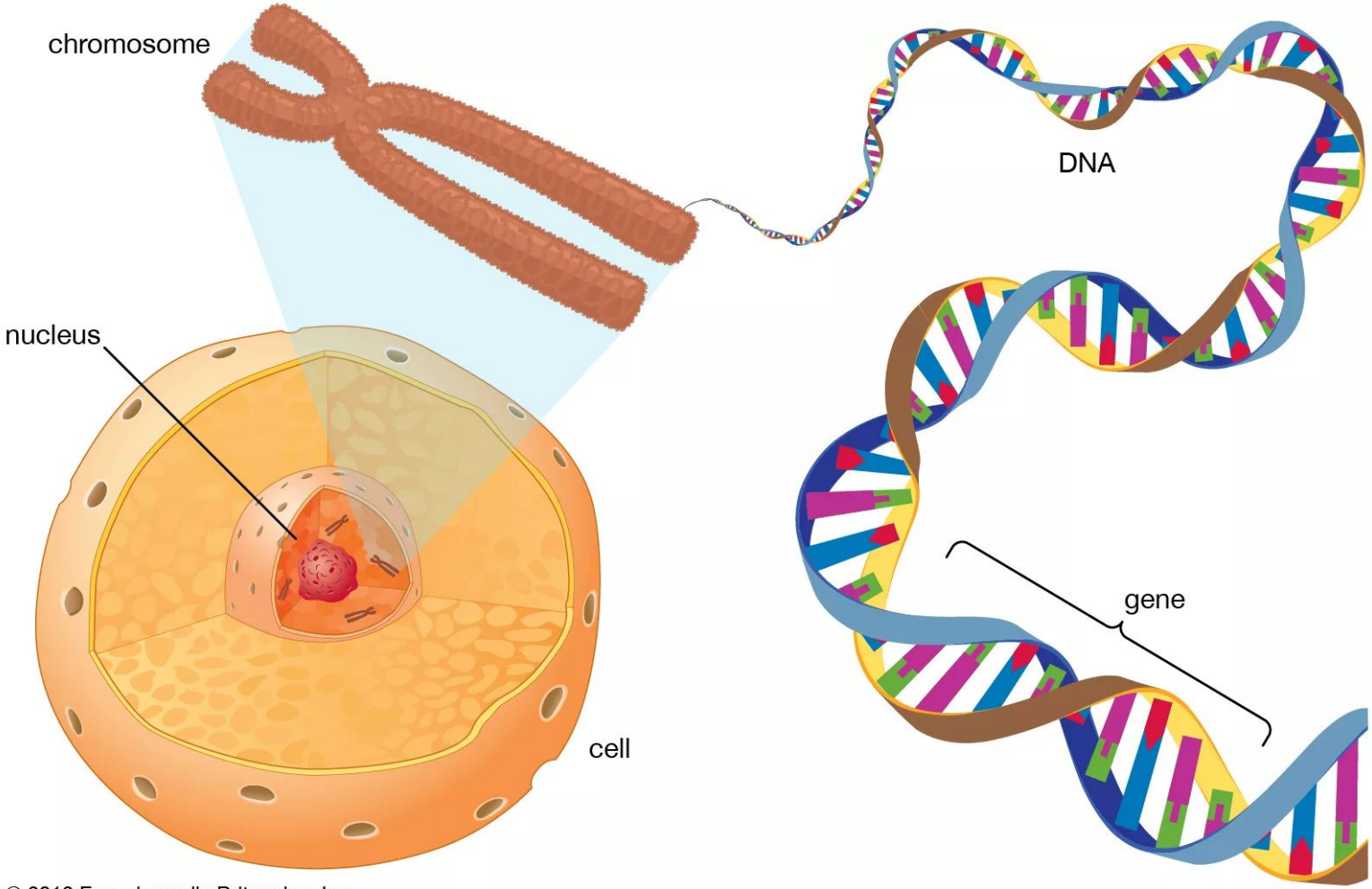
Iterative class discovery and feature selection using Minimal Spanning Trees

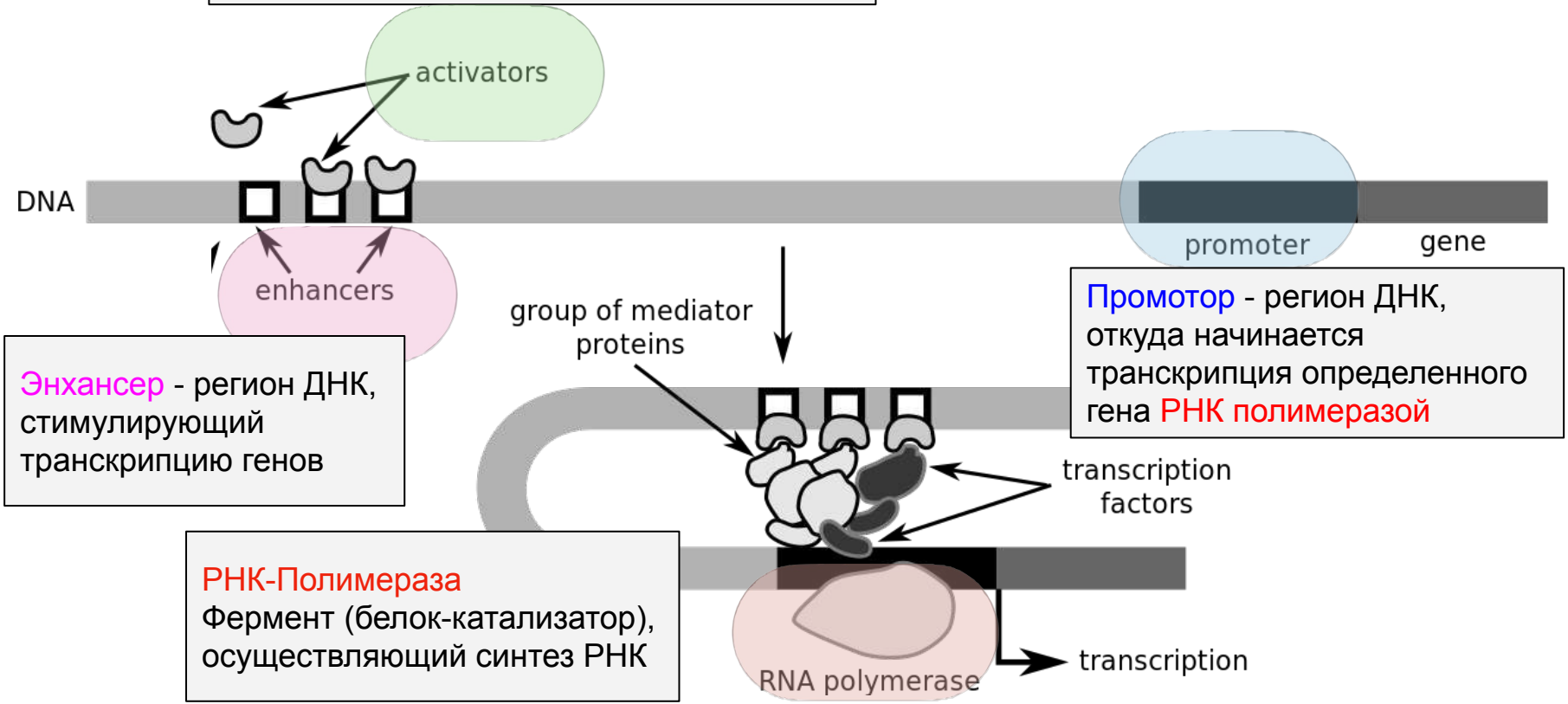
Sudhir Varma and Richard Simon

Доклад подготовлен студентами:

Смирнова Валерия
Иванникова Анастасия
Евтушенко Олег
Муштаков Макар
Кутников Александр
Цеквава Акакий



Активатор - белок, связывающий свой регион энхансер с промотором генов

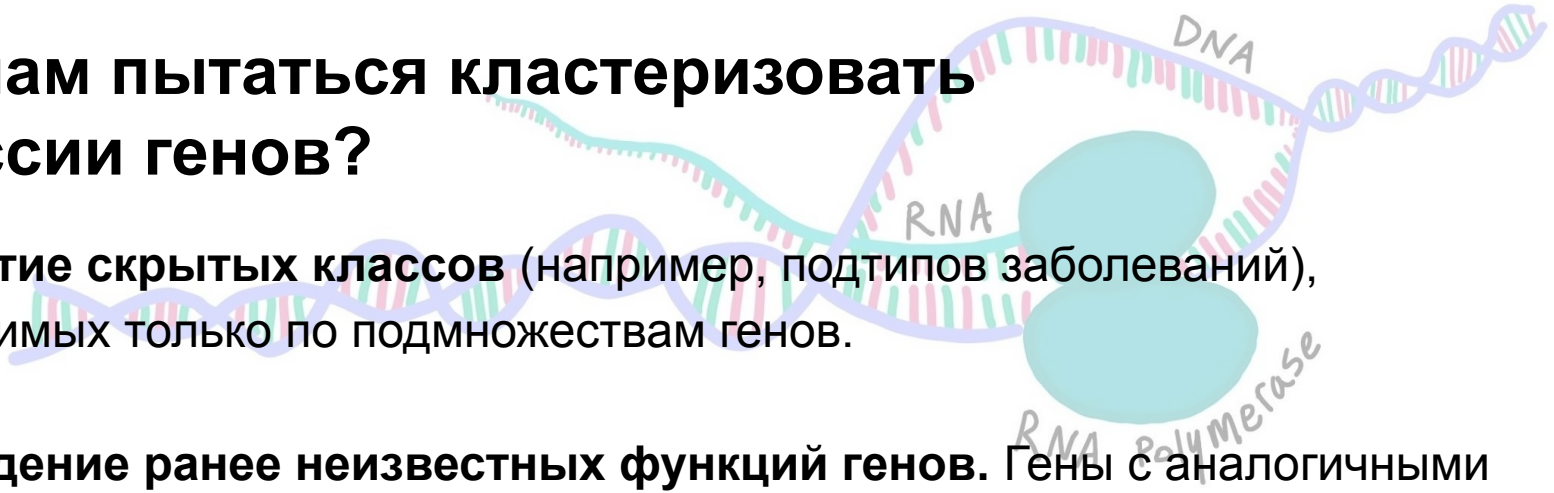


Энхансер - регион ДНК, стимулирующий транскрипцию генов

Промотор - регион ДНК, откуда начинается транскрипция определенного гена РНК полимеразой

РНК-Полимераза
Фермент (белок-катализатор), осуществляющий синтез РНК

Зачем нам пытаться кластеризовать экспрессии генов?



- **Открытие скрытых классов** (например, подтипов заболеваний), различимых только по подмножествам генов.

- **Нахождение ранее неизвестных функций генов.** Гены с аналогичными образцами экспрессии (совместно выраженные гены) группируются в кластеры вместе с подобными клеточными функциями.

- Поиск общих последовательностей ДНК в зонах промоторов генов внутри одного и того же кластера. Это позволяет **определять регуляторные элементы, специфичные для каждого кластера генов**.

Проблема

При кластеризации генов (а это пространства высокой размерности) зачастую используются все гены из выборки, в том числе noise/irrelevant .

Как следствие, небольшое количество информативных сигналов/признаков часто теряется на фоне большого количества шума.

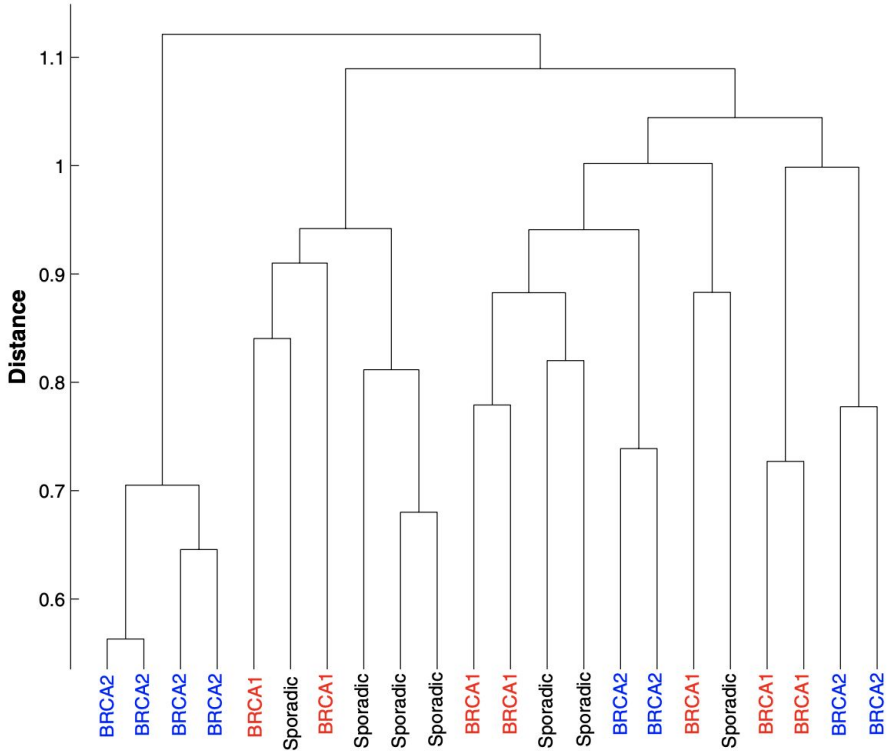


Figure 1
Hierarchical clustering of BRCA data using all genes. Hierarchical clustering of BRCA data using centered correlation with average linkage. Inclusion of all genes in the clustering swamps out the differences between samples with BRCA1 and BRCA2 mutation.

Общие идеи авторов

- Чередование между кластеризацией и отбором релевантных признаков/сигналов - позволяет постепенно избавляться от шума
- Использование MST для кластеризации вместо полного перебора
- Отбор генов с помощью t-statistic для определения кластеров
- В итоге получаем множество бинарных разбиений и наборы генов, значимых для этих разбиений

Minimal Spanning Tree

Минимальное остовное дерево (MST) — это дерево, которое соединяет все точки таким образом, что сумма длин ребер минимальна.

δ - длина удаленного ребра

Удаляя ребро длины δ, мы гарантируем разбиение, в котором два кластера разделены как минимум на δ.

Все возможные бинарные разбиения (без MST): 2^(N-1)-1

Наш алгоритм рассматривает только разбиения, полученные удалением одиночных ребер из MST: N-1

MST используется не для того, чтобы найти одно лучшее разбиение, а чтобы отобрать множество перспективных кандидатов на разбиение.

Метрика кластеризации. Fukuyama-Sugeno

$$FS(S) = \sum_{k=1}^{2} \sum_{j=1}^{N_k} \left[\left\| x_j^k - \mu_k \right\|^2 - \left\| \mu_k - \mu \right\|^2 \right]$$

Чем меньше значение FS(S), тем лучше разбиение: кластеры плотные и далекие друг от друга.

Отбор признаков

Для выбранного разбиения мы хотим определить, какие гены имеют значимую разницу в экспрессии между кластерами.

Для каждого гена вычисляется t-статистика для сравнения среднего уровня экспрессии в двух кластерах.

Параметр порога в процентах $P\_thresh \in (0,100)$ используется для вычисления T_thresh. Значение T_thresh соответствует перцентилю P_thresh/2 распределения Стьюдента с нулевым средним и N-2 степенями свободы (где N — число образцов).

Выбираются гены с абсолютным значением t-статистики, превышающим порог T_thresh.

Описание алгоритма


Np < MaxNp?
Выходим из алгоритма
Набор генов F = F_set, t = T_thresh / 2
Достаточно ли нам генов для разбиения?
Выходим из алгоритма
Строим MST, считаем F-S, выбираем лучшее разбиение P*
Для каждого гена в P* считаем t_g, собираем F_new = {g: t_g > t}
Получили ли мы полное покрытие? F_new = F, t = T_thresh
F = F_new, увеличиваем t
Выводим P*, F
F_set = F_set - F, Np += 1

Результаты

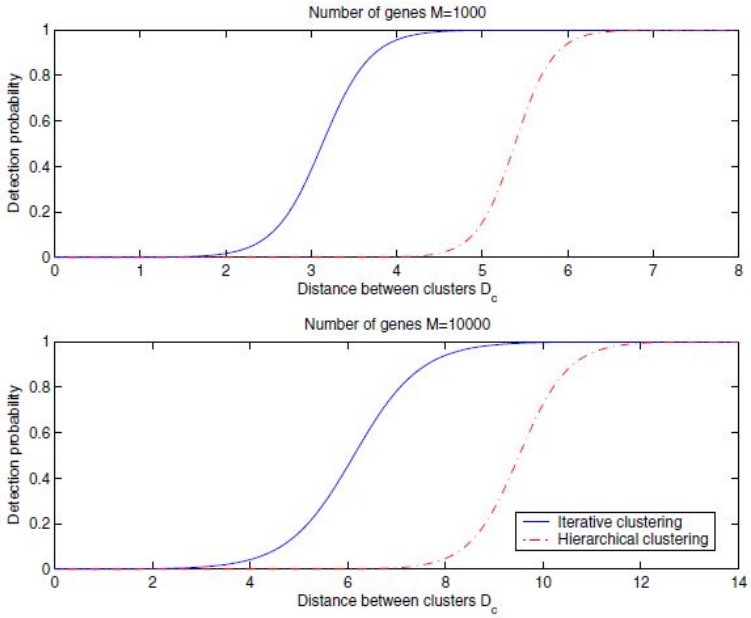


Figure 2
Detection probability vs. cluster separation. Probability of detection of the planted partition as a function of the distance between the clusters in the partition.

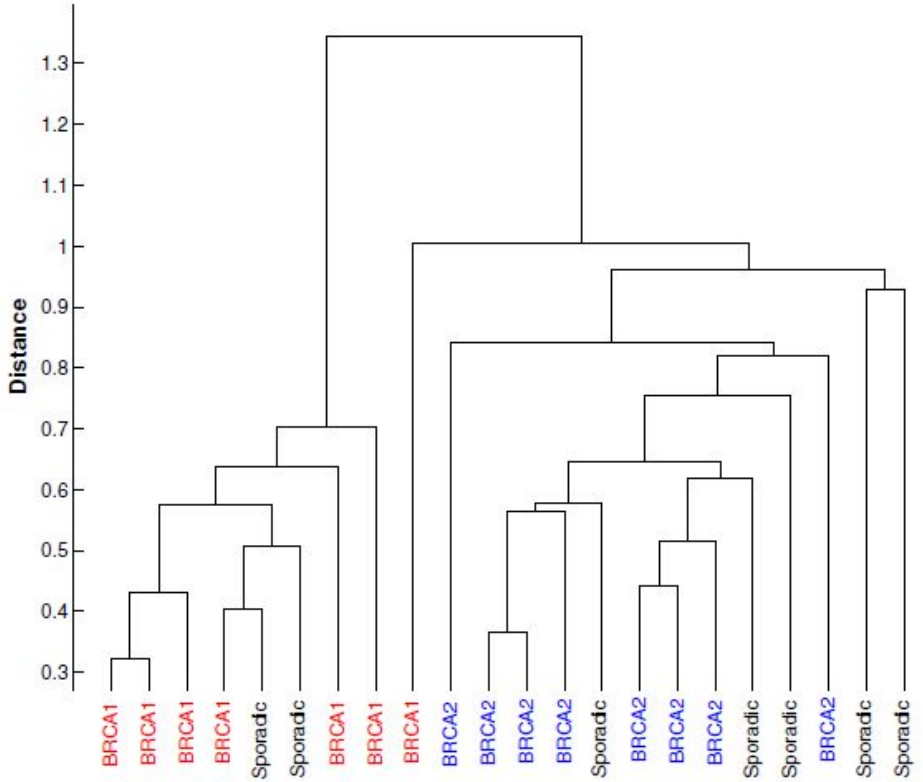


Figure 3
Hierarchical clustering of BRCA data using selected genes. Hierarchical clustering of BRCA data using only the genes supporting Partition 4. BRCA1 and BRCA2 are separated with one misclassification.

Table 1: Results on BRCA data-set

Partition number	Cluster number	BRCA1	BRCA2	Number of genes selected
1	1	0	4	80
	2	7	4	
2	1	7	4	110
	2	0	4	
3	1	5	3	73
	2	2	5	
4	1	6	0	61
	2	1	8	

Table 2: Results on Leukemia data-set

Partition number	Cluster number	AML	ALL	Number of genes selected
1	1	1	46	578
	2	24	1	
2	1	4	29	650
	2	21	18	
3	1	25	38	108
	2	0	9	
4	1	20	27	81
	2	5	20	

Другие подходы

- **Ben-Dor et al. — полный перебор бинарных разбиений** (симулированный отжиг (simulated annealing)) (дорого)
- **Xing & Karp — Normalized Cut** (графовый разрез) (внутри сильные связи, снаружи слабые) (риск маленьких кластеров)
- **Von Heydebreck, Tang et al. — отбор генов через критерий качества кластеров** (завязано на эту метрику)
- **Hastie et al. (gene shaving)** — итеративное сбривание генов с наибольшим вкладом (работает для генов, не образцов).

Слабые стороны MST (Varma & Simon)

- Требует настройки параметров (порог t-теста, выбор метрики расстояния)
- Работает только с бинарными разбиениями (две группы за раз).
- T-тест может быть нестабильным при малом числе образцов.

Новые улучшенные методы: Sparse k-means

Daniela M Witten, Robert Tibshirani, 2010 — A framework for feature selection in clustering

Принципиальное отличие алгоритмов

- **K-means:** минимизирует *внутрикластерную дисперсию*
- **Sparse K-means:** максимизирует *взвешенную межкластерную дисперсию*

$$\text{maximize}_{C_1,\ldots,C_K,\mathbf{w}} \left\{\sum_{j=1}^{p} w_j\left(-\sum_{k=1}^{K}\frac{1}{n_k}\sum_{i,i'\in C_k} d_{i,i',j}\right)\right\}$$
$$\text{subject to } ||\,\mathbf{w}\,||^2 \le 1, ||\,\mathbf{w}\,||_1 \le s, w_j \ge 0\ \forall j.$$

  - Большие *дисперсии* → признак реально различает группы.
  - Маленькие *дисперсии* → нулевой вес → признак бесполезен, выбрасывается.

Это позволяет одновременно находить кластеры и отбирать информативные признаки.

Решение проблем итеративного MST

1. Веса признаков отбираются автоматически через *L1*-регуляризацию. Параметр *s* подбирается по перестановочным тестам → не требует ручной настройки. (Как это было с порогом *t*-статистики)
2. За счёт зануления шумовых признаков снижается влияние «плохих» измерений, метрика становится более устойчива.

   Существует множество вариантов sparse K-means (робастные, kernelized) → если евклидова геометрия не подходит, можно заменить фазу кластеризации на kernel-kmeans / робастную версию (RSKC и др.)

3. Использует совместную (*embedded*) регуляризацию (*L1*) при оптимизации кластеров и весов; регуляризация снижает дисперсию оценки признаков.
4. Работает сразу с *K* центрами → учитывает различия между всеми группами одновременно.

Новые улучшенные методы: Seurat

Stuart et al., 2019 — Comprehensive Integration of Single-Cell Data

Работа алгоритма

1. **Чистка данных**: например, убираем клетки, где слишком мало экспрессированных генов
2. **Нормализация**: например, учитывать глубину прочтений, SCTransform возвращает стабилизированные по дисперсии образцы
3. **Выбор «самых вариабельных генов»:** с помощью среднего и дисперсии для каждого гена, берем гены с самыми лучшими показателями
4. **Снижение размерности**: с помощью PCA переводим из пространства тысячи генов в 20-50 главных компонент
5. **Построение графа соседей:** строим k-NN граф для каждого образца по PCA-признакам, затем строим SNN граф
6. **Кластеризация графа:** Алгоритмы кластеризации Louvein и Leiden увеличивают модульность
7. **Поиск маркерных генов:** ищемс помощью GLM моделей или непараметрических статистических тестов

Решение проблем итеративного MST

1. Seurat использует графовую многокластерную оптимизацию (Louvain/Leiden → community detection).

2. Метрика смещается в PCA-пространство, а затем используется SNN (Jaccard) — локальная согласованность соседств, а не абсолютные расстояния;
3. Seurat выбирает гены с наибольшей изменчивостью по клеткам; использует GLM модели для нормализации дисперсии и ищет дифференциально экспрессируемые гены между кластерами
4. Кластеризация по соседям в kNN-графе помогает принять решение на уровне связности графа, а не одного гена