@CSoneson

Clustering of scRNA-seq data

Charlotte Soneson

Friedrich Miescher Institute for Biomedical Research &
SIB Swiss Institute of Bioinformatics

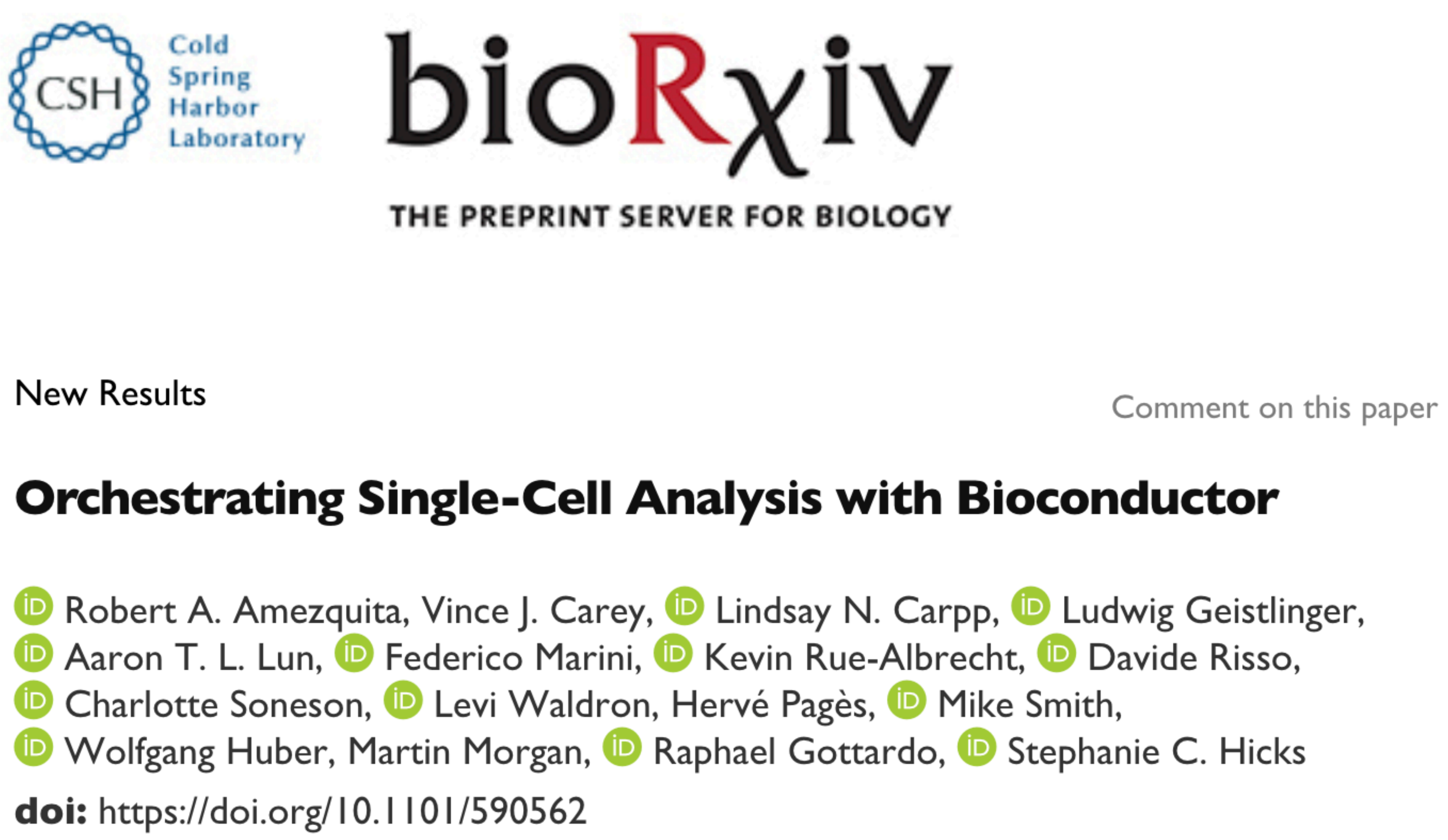
Single Cell Autumn School 2019

SIB
Swiss Institute of
Bioinformatics

FMI
Friedrich Miescher Institute
for Biomedical Research

Quick plug: the OSCA book

- Companion to "Orchestrating single-cell analysis with Bioconductor"
- https://osca.bioconductor.org/



Cold Spring Harbor Laboratory

bioRxiv

THE PREPRINT SERVER FOR BIOLOGY

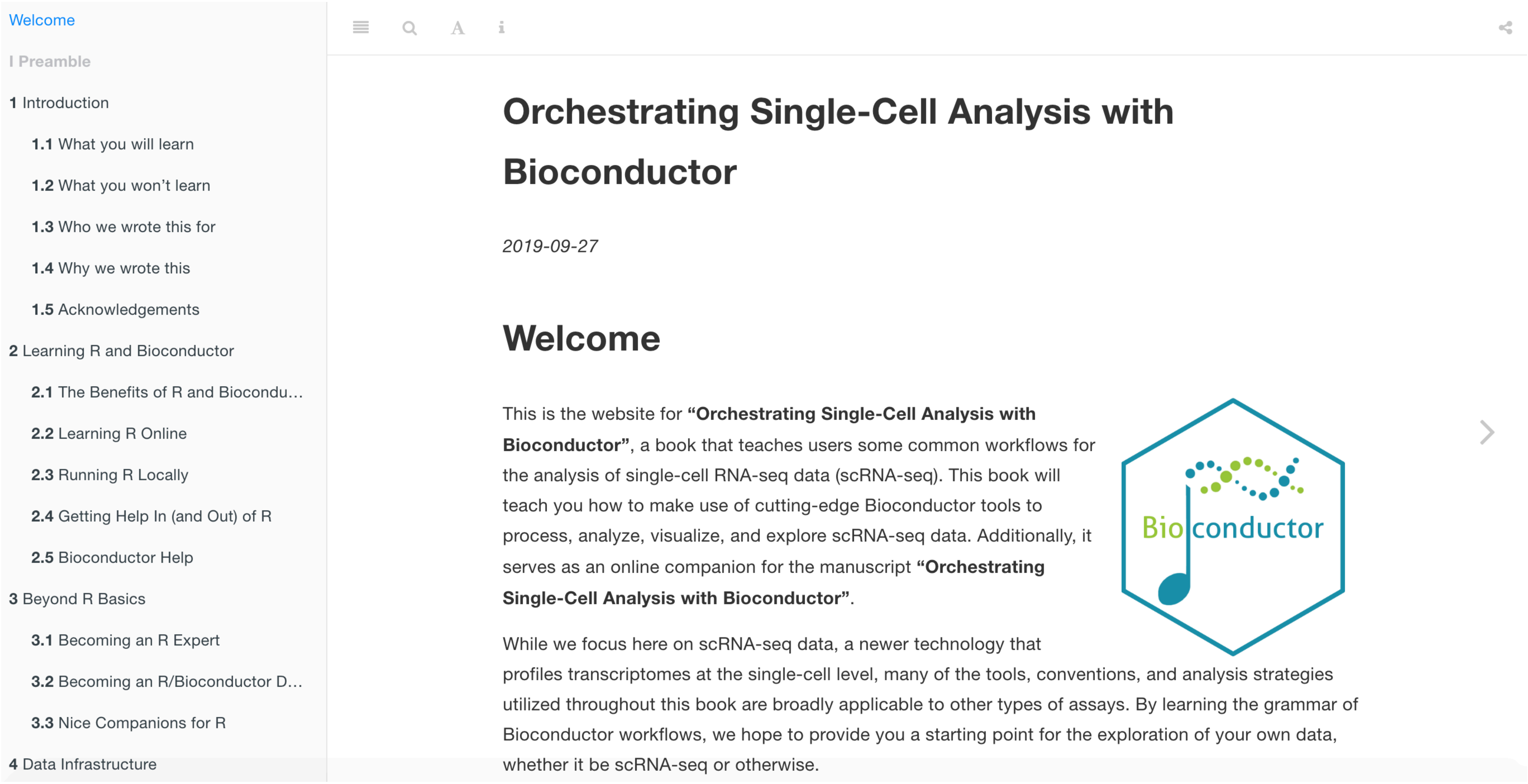
New Results

Comment on this paper

Orchestrating Single-Cell Analysis with Bioconductor

Robert A. Amezquita, Vince J. Carey, Lindsay N. Carpp, Ludwig Geistlinger, Aaron T. L. Lun, Federico Marini, Kevin Rue-Albrecht, Davide Risso, Charlotte Soneson, Levi Waldron, Hervé Pagès, Mike Smith, Wolfgang Huber, Martin Morgan, Raphael Gottardo, Stephanie C. Hicks

doi: https://doi.org/10.1101/590562



Welcome

I Preamble

1 Introduction

1.1 What you will learn

1.2 What you won't learn

1.3 Who we wrote this for

1.4 Why we wrote this

1.5 Acknowledgements

2 Learning R and Bioconductor

2.1 The Benefits of R and Biocondu...

2.2 Learning R Online

2.3 Running R Locally

2.4 Getting Help In (and Out) of R

2.5 Bioconductor Help

3 Beyond R Basics

3.1 Becoming an R Expert

3.2 Becoming an R/Bioconductor D...

3.3 Nice Companions for R

4 Data Infrastructure

Orchestrating Single-Cell Analysis with Bioconductor

2019-09-27

Welcome

This is the website for **"Orchestrating Single-Cell Analysis with Bioconductor"**, a book that teaches users some common workflows for the analysis of single-cell RNA-seq data (scRNA-seq). This book will teach you how to make use of cutting-edge Bioconductor tools to process, analyze, visualize, and explore scRNA-seq data. Additionally, it serves as an online companion for the manuscript **"Orchestrating Single-Cell Analysis with Bioconductor"**.

Bioconductor

While we focus here on scRNA-seq data, a newer technology that profiles transcriptomes at the single-cell level, many of the tools, conventions, and analysis strategies utilized throughout this book are broadly applicable to other types of assays. By learning the grammar of Bioconductor workflows, we hope to provide you a starting point for the exploration of your own data, whether it be scRNA-seq or otherwise.

What is clustering, and why do we do it?

- Partitioning of the objects (here, cells) into *groups*

- Cells in the same group should be *similar* to each other, and *different* from cells in other groups

- The aim is often to simplify and summarize the complex data and aid interpretation

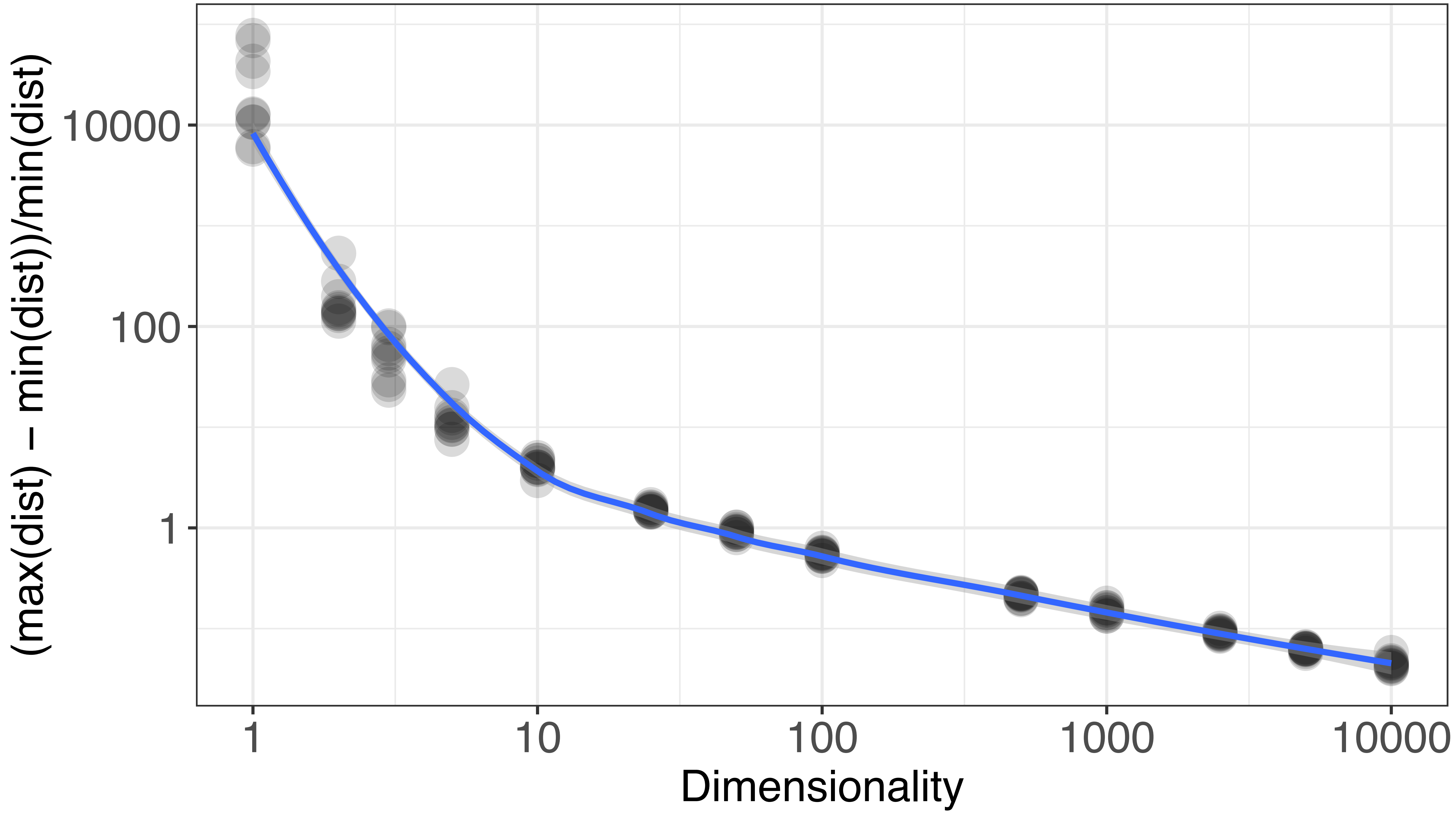
- In particular, clusters are often treated as proxies for cell types or states

Input

- Most clustering methods rely on some form of distance calculations.

- Distances are notoriously difficult to interpret in high-dimensional spaces (the “curse of dimensionality” - all distances tend to become similar).

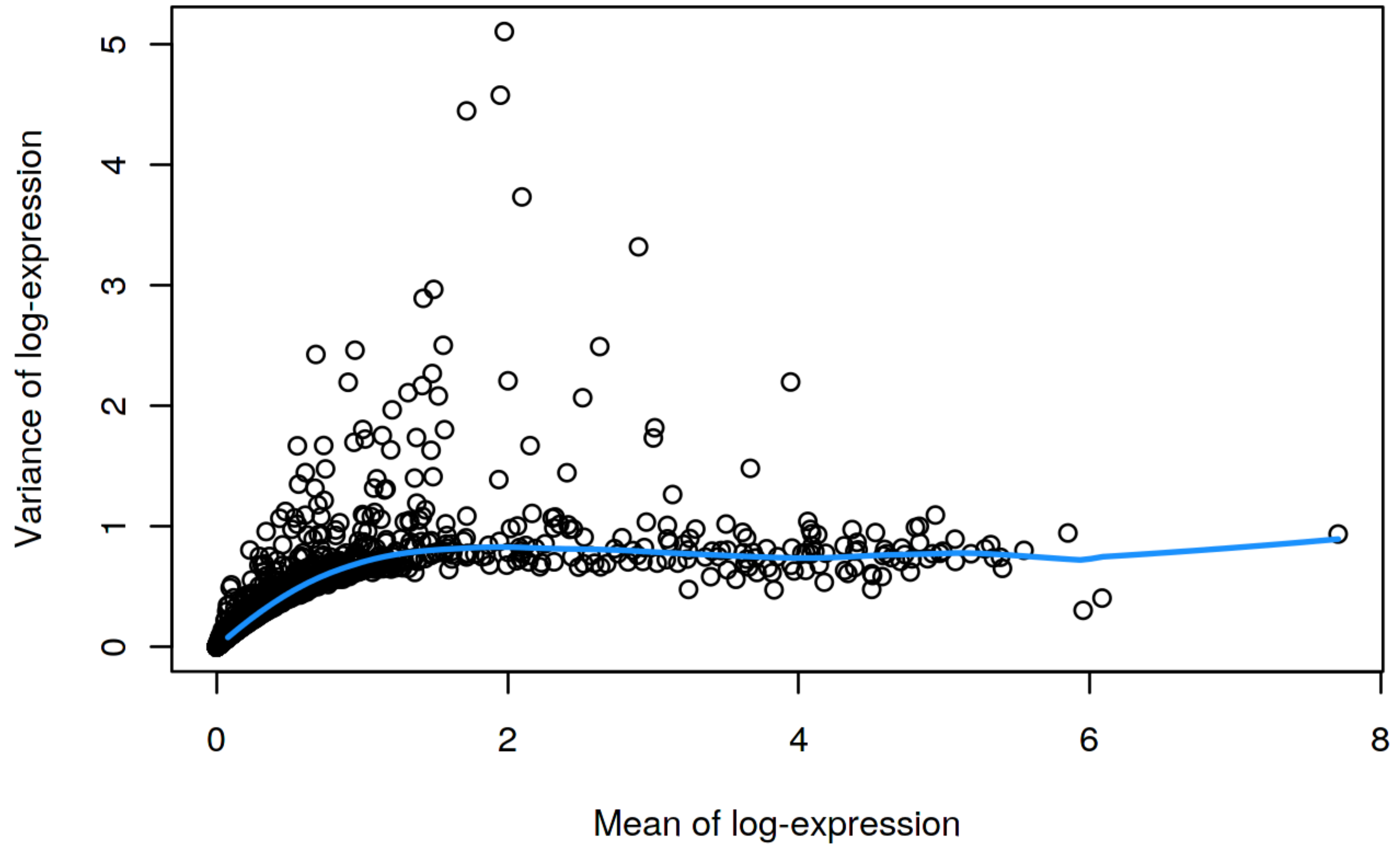
- Clustering is often applied on a reduced dimension representation of the data (often PCA).

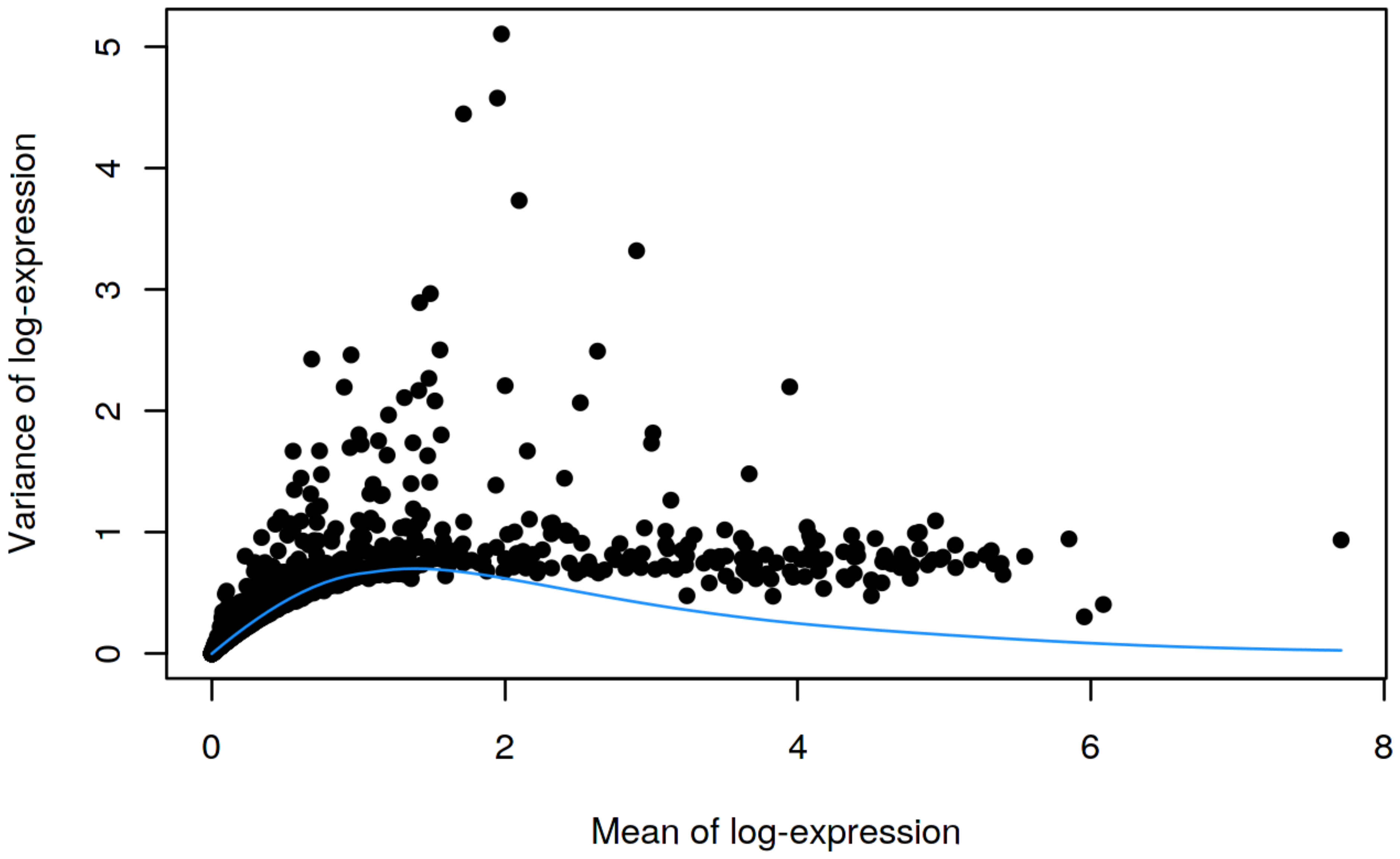


Input

- Alternatively, apply clustering to a subset of (e.g., highly variable) genes
- With `scran`, decompose observed variance into 'technical' and 'biological' variance

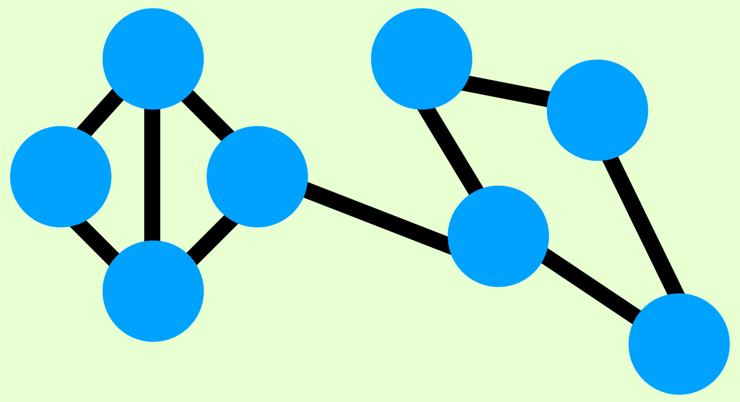
Side note: This can be used to determine the number of principal components (in the `scran::denoisePCA()` function) - remove PCs (from the end) until you have explained the total technical variance.



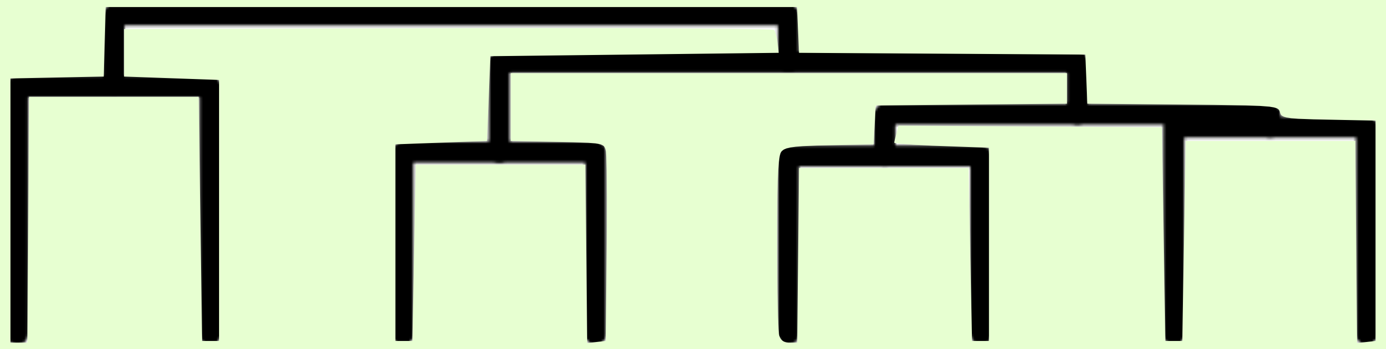


Clustering methods

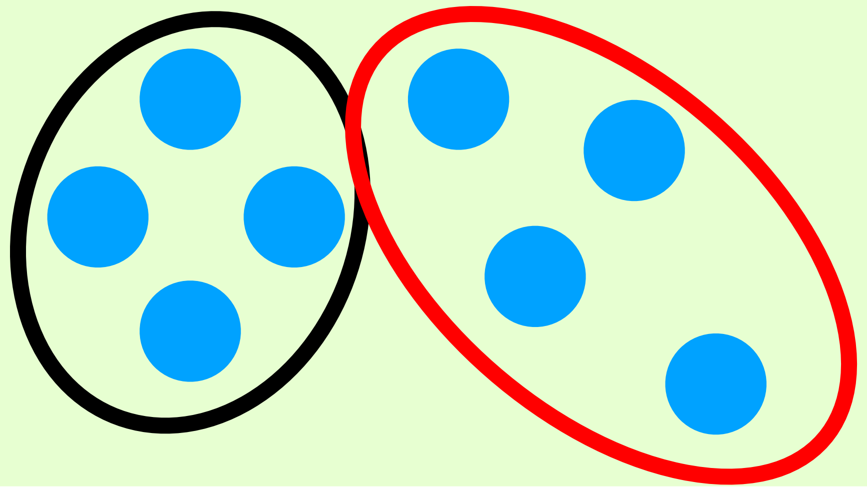
Graph-based



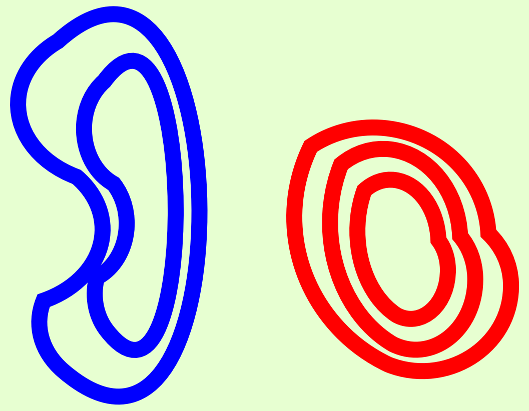
Hierarchical



Centroid-based



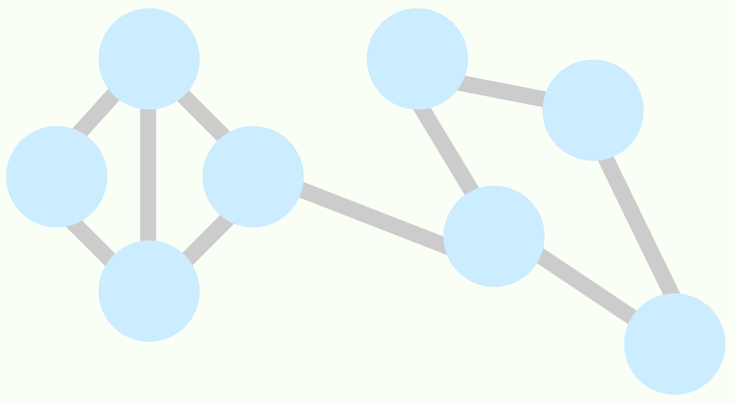
Density-based



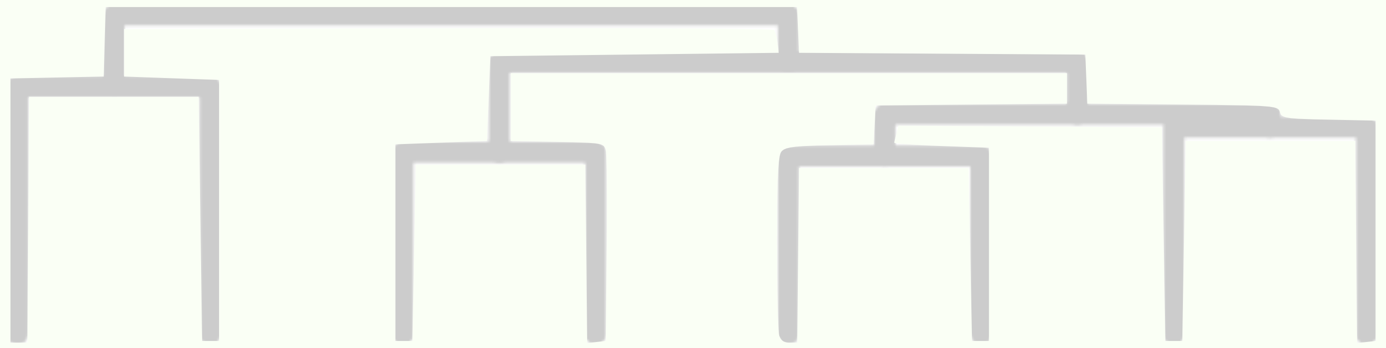
Consensus clustering

Clustering methods

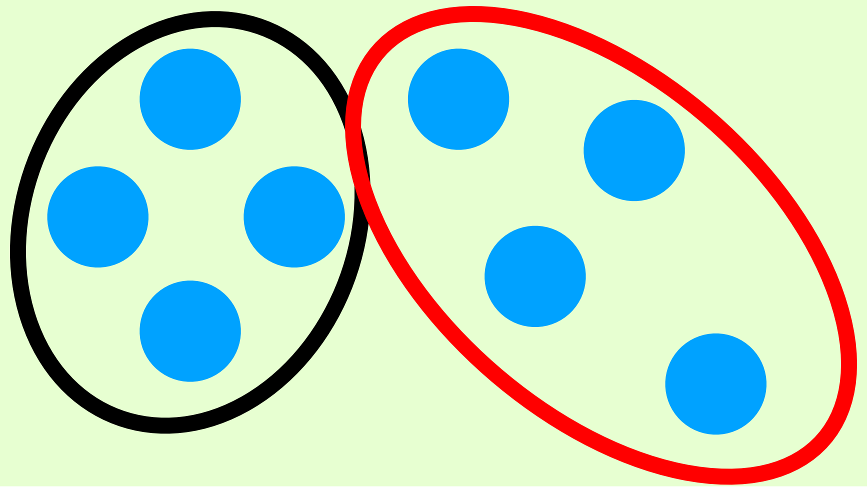
Graph-based



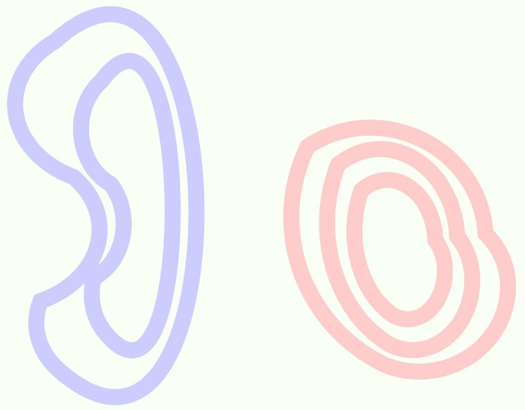
Hierarchical



Centroid-based



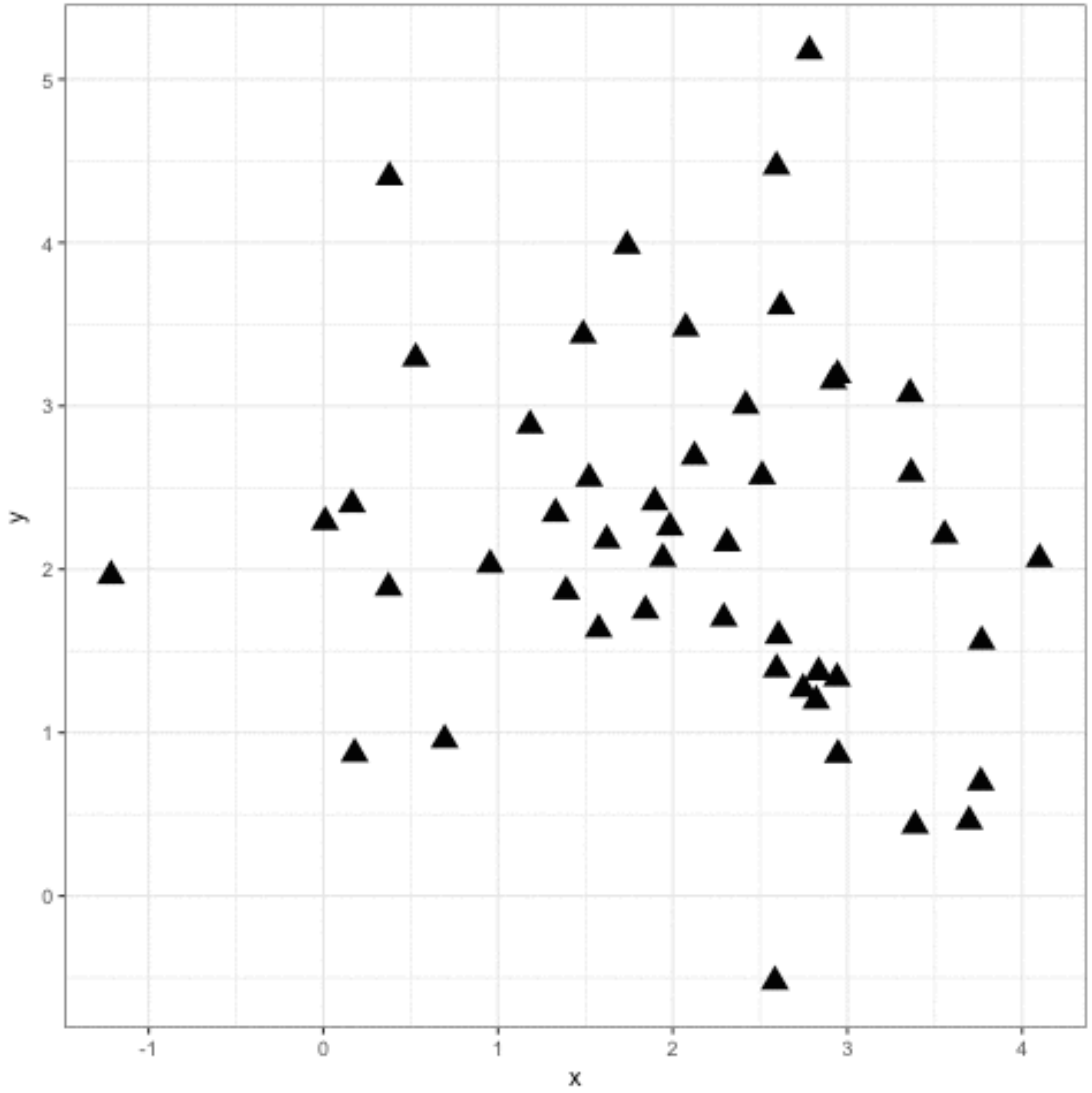
Density-based



Consensus clustering

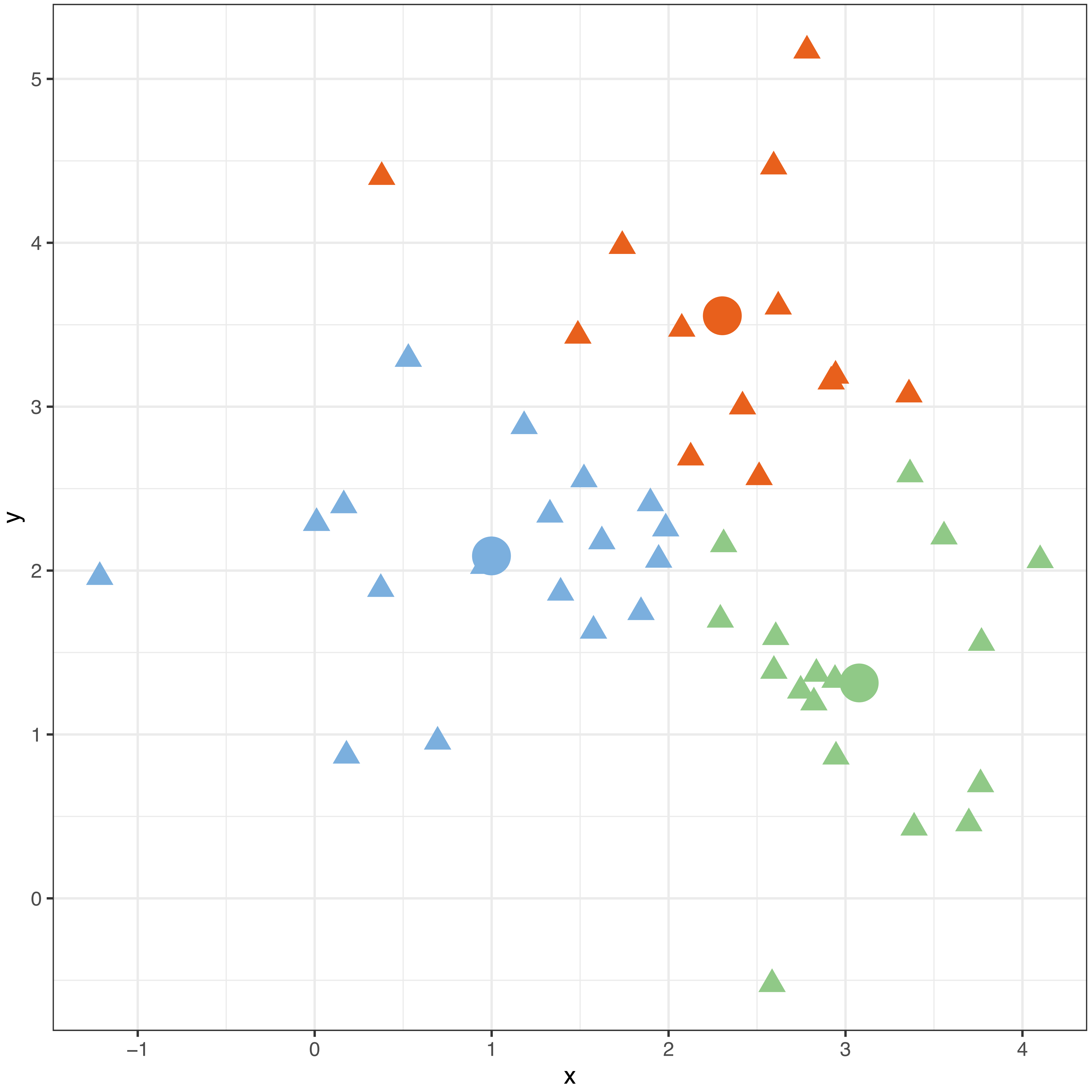
k-means

- Partitions cells into k clusters (where k is predefined)
- Initialize k cluster centers (randomly)
- Iterate until convergence:
    - assign cells to the closest center
    - recalculate cluster centers



k-means

- Implicitly favors spherical clusters
- k must be prespecified, and the number of clusters is forced to be k, even if there is strong evidence that there are more clusters
- Results may depend on the initialization - run multiple times for stability



How to decide on the number of clusters?

- k-means tries to minimize the observed within-cluster sum of squares

- The **gap statistic** compares the observed within-cluster sum of squares to that *expected* under a suitable null distribution (e.g., obtained by randomly placing the points within the original bounding box)

- Large gap statistic - better clustering

- Choose most parsimonious k beyond which the gap statistics doesn't increase much

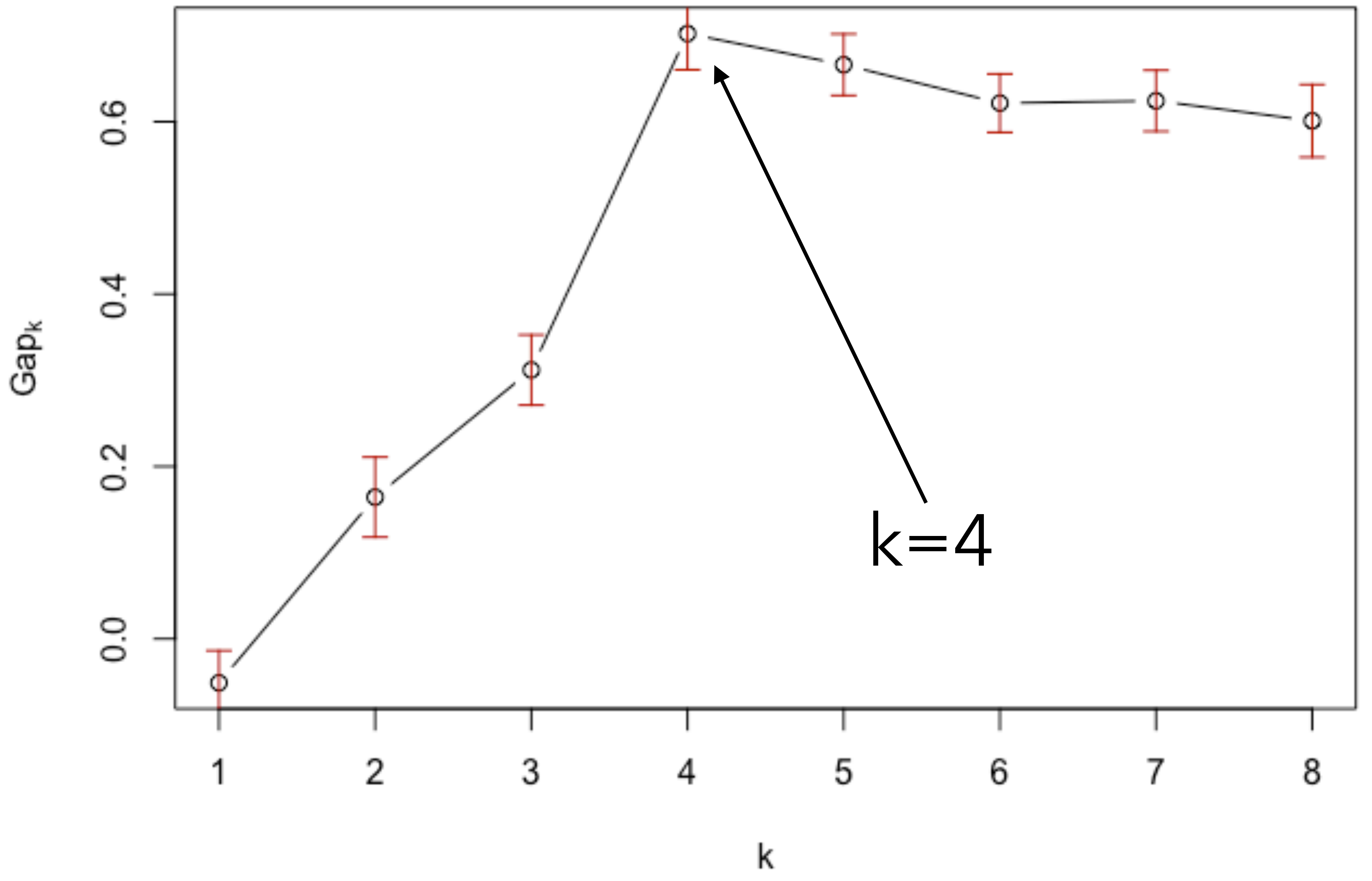
J. R. Statist. Soc. B (2001)
63, *Part* 2, *pp.* 411–423

Estimating the number of clusters in a data set via the gap statistic

Robert Tibshirani, Guenther Walther and Trevor Hastie

Stanford University, USA

cluster::clusGap(X, kmeans, K.max=8)



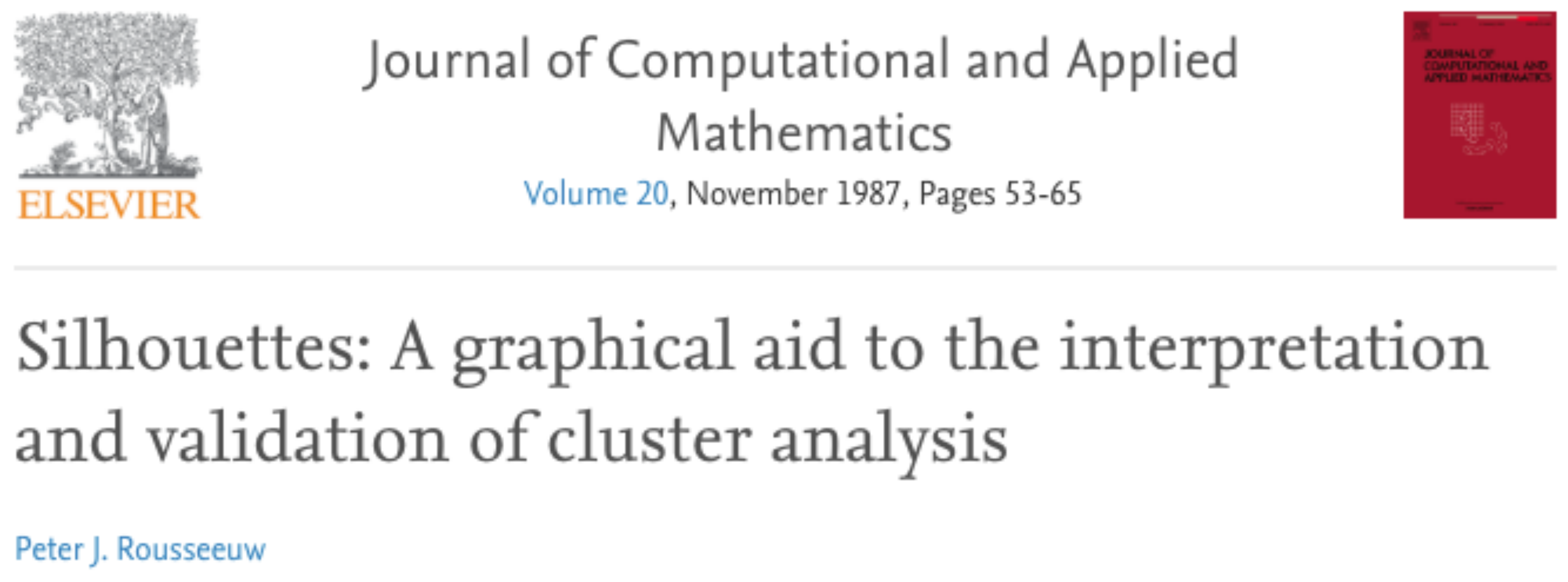
How to decide on the number of clusters?

- The **silhouette score** compares (for each cell *i*) the average distance to the cells in the same cluster as *i* (*a(i)*) with the average distance to the cells in the nearest cluster (*b(i)*)

- Large silhouette scores - compact, well-separated clusters

$$s(i) = \frac{b(i) - a(i)}{max(a(i), b(i))}$$

$$-1 \leq s(i) \leq 1$$



ELSEVIER

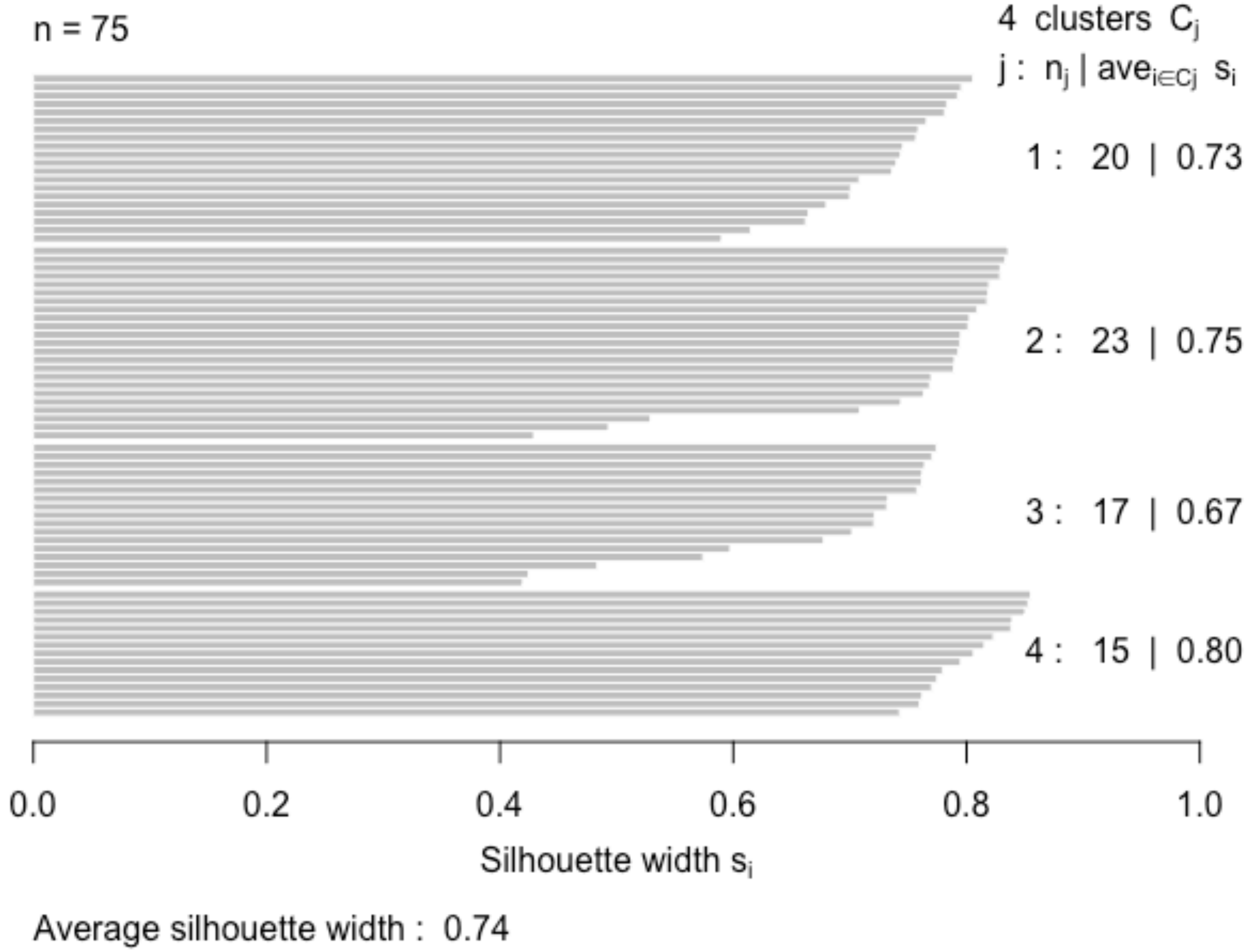
Journal of Computational and Applied Mathematics

Volume 20, November 1987, Pages 53-65

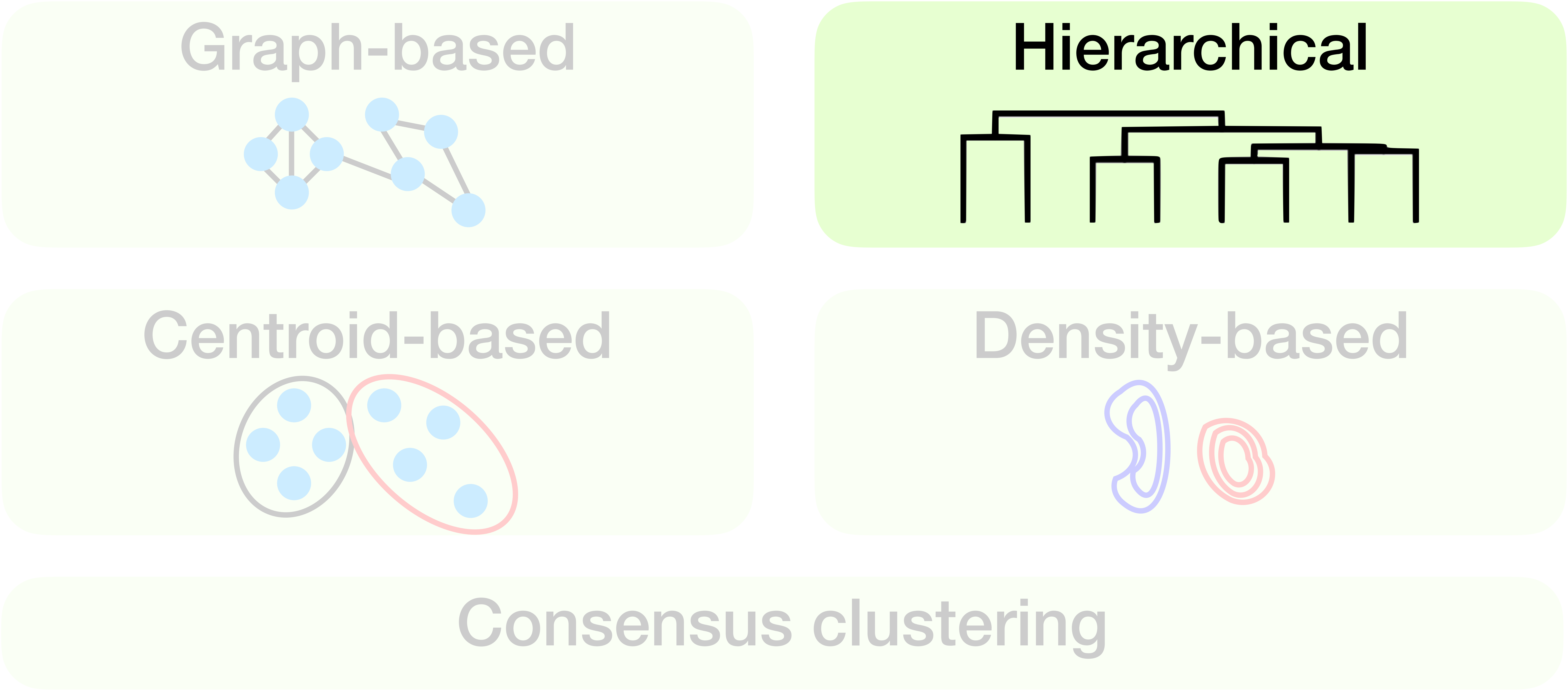
Silhouettes: A graphical aid to the interpretation and validation of cluster analysis

Peter J. Rousseeuw

```
cluster::silhouette(clusters, distances)
```



Clustering methods



Graph-based

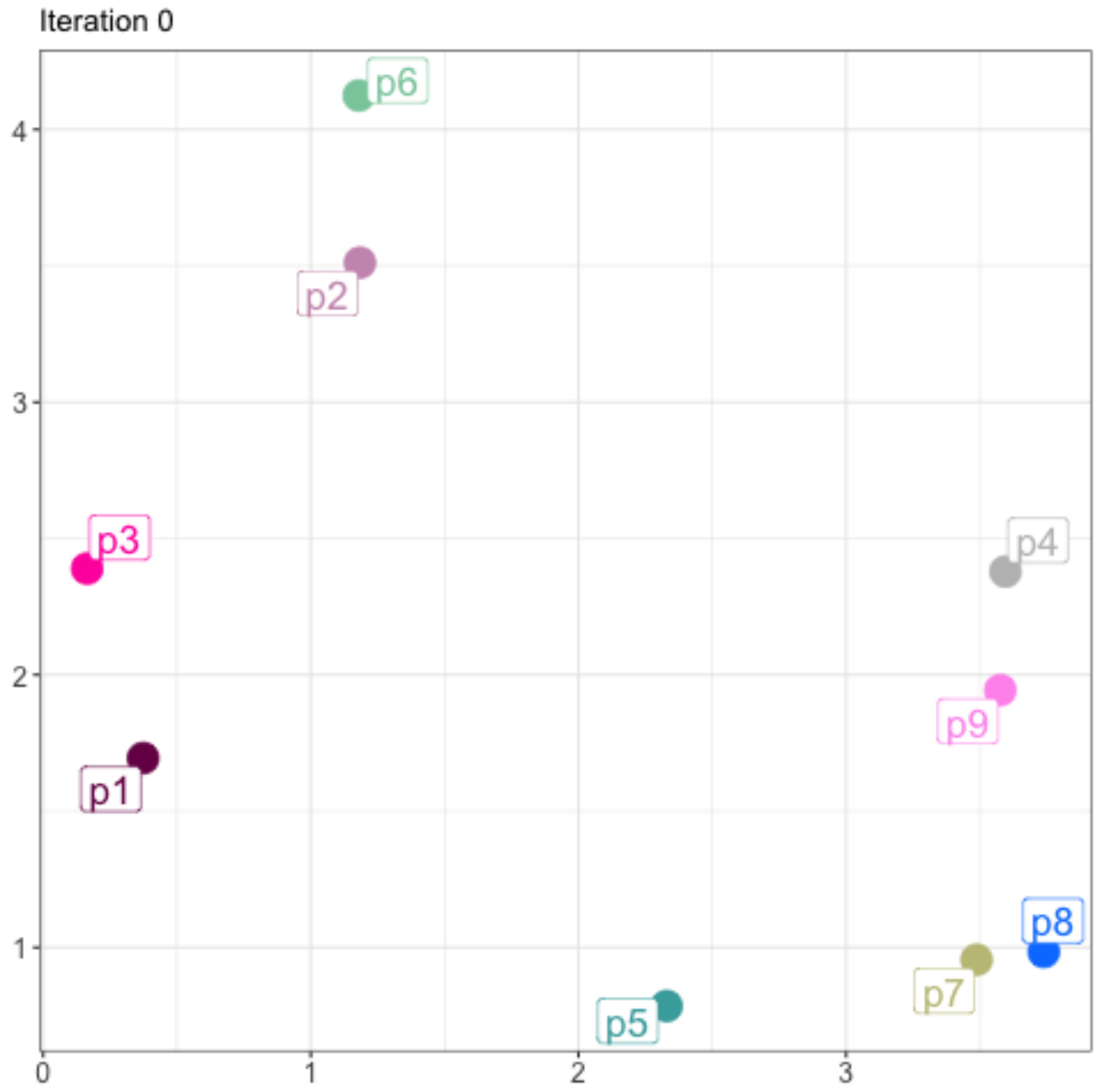
Hierarchical

Centroid-based

Density-based

Consensus clustering

(Agglomerative) hierarchical clustering

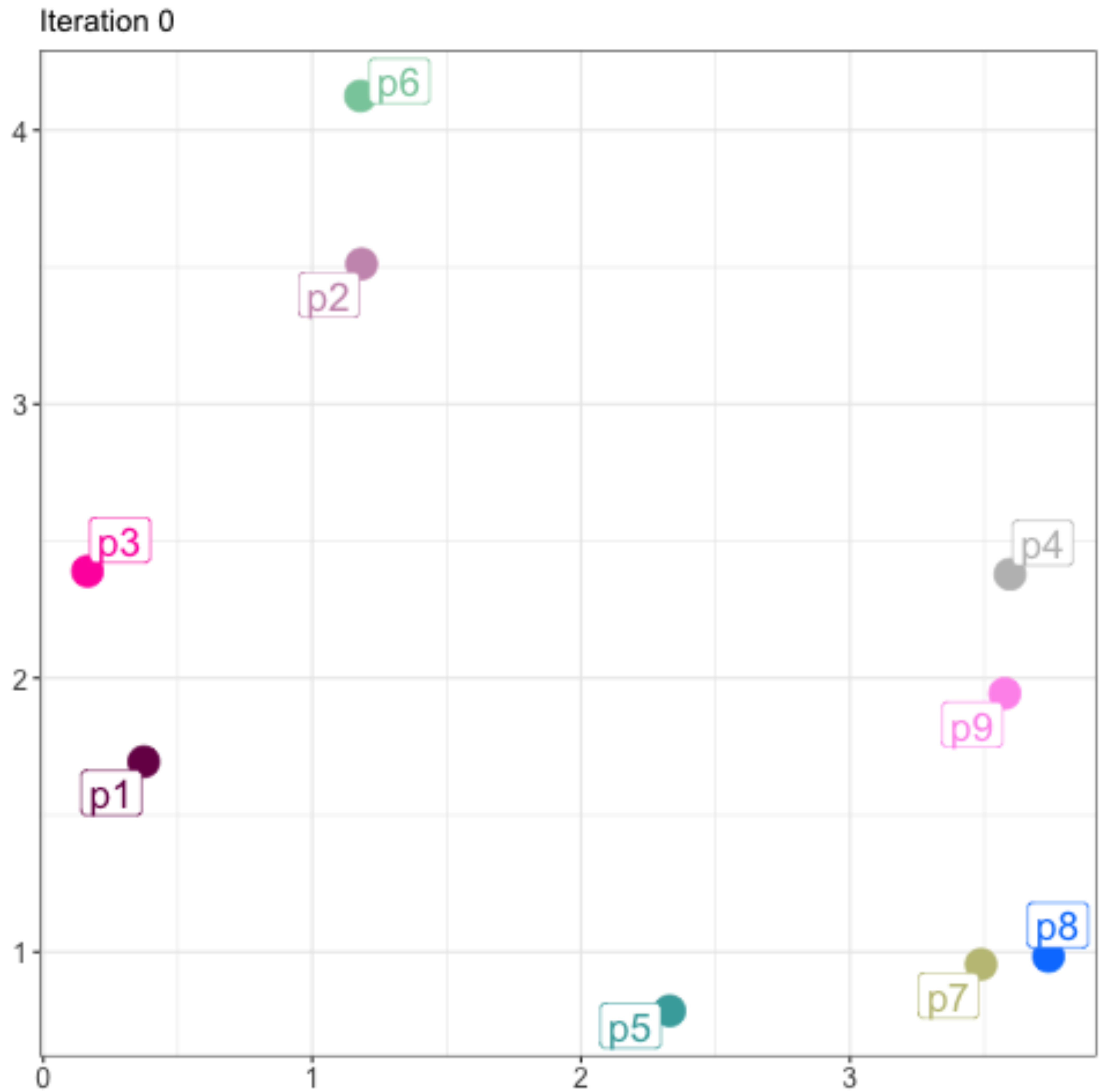


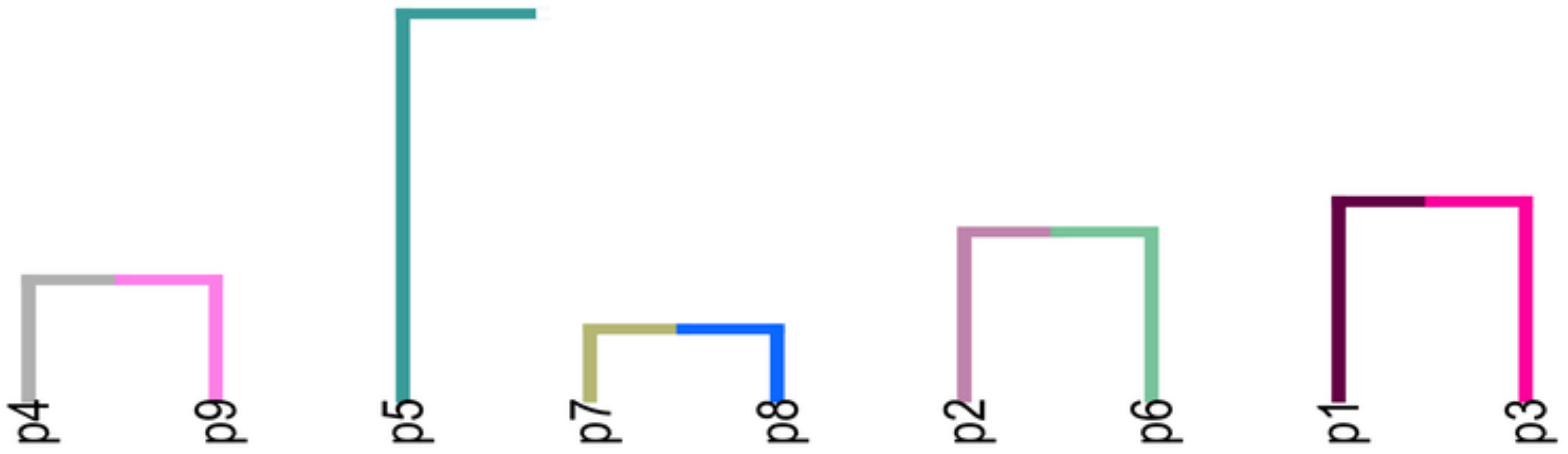
- Start with each cell as its own cluster

- In each step, join the two most similar clusters into a new cluster

- Continue until there is only one cluster, containing all the cells

Hierarchical clustering

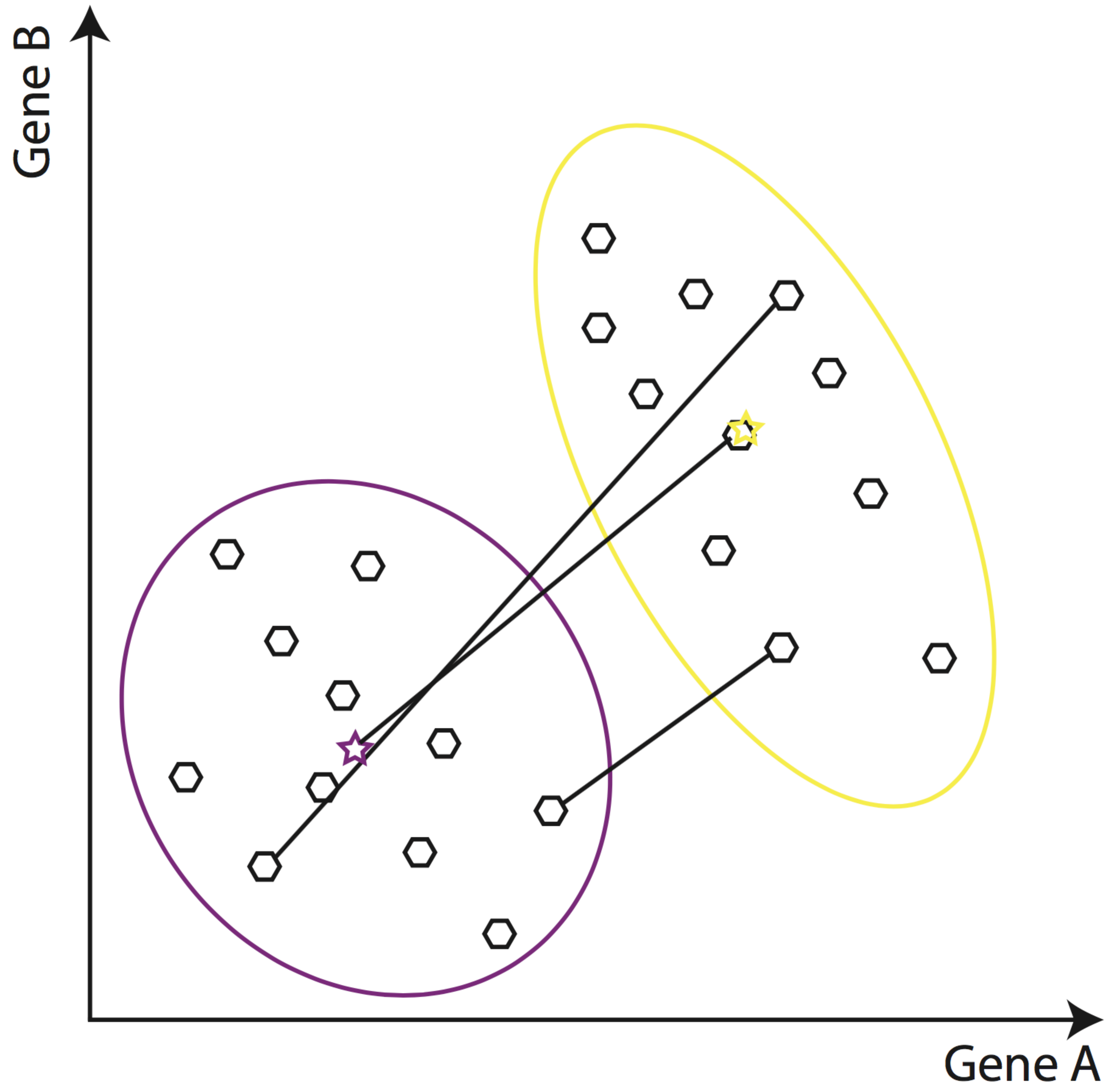




Linkage

- The linkage indicates how dissimilarities among *clusters* are calculated, and thus how to decide which to clusters to merge in each step

- Common choices:
  - complete
  - single
  - average
  - centroid
  - Ward (in R: “ward.D2”) - use only when individual point-to-point distances are Euclidean



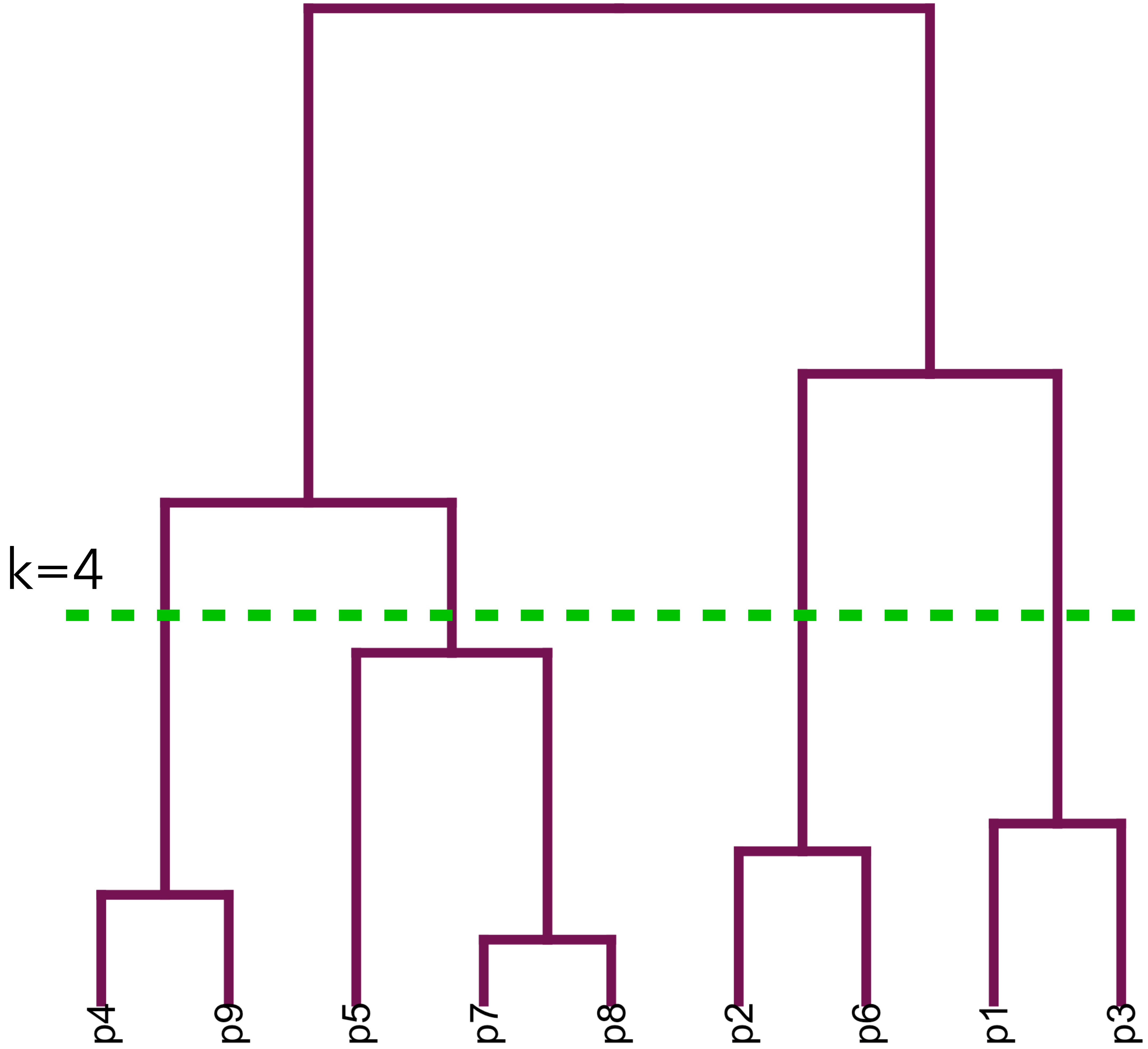
Getting the partition

- The dendrogram provides a rich summary, also of relationships among clusters at varying resolution

- To get a partitioning of the cells, cut the tree

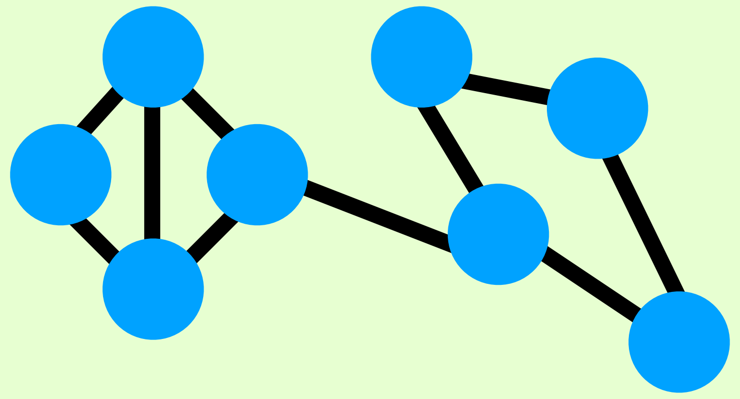
    - fixed height: stats::cutree()

    - dynamically: dynamicTreeCut::cutreeDynamic()

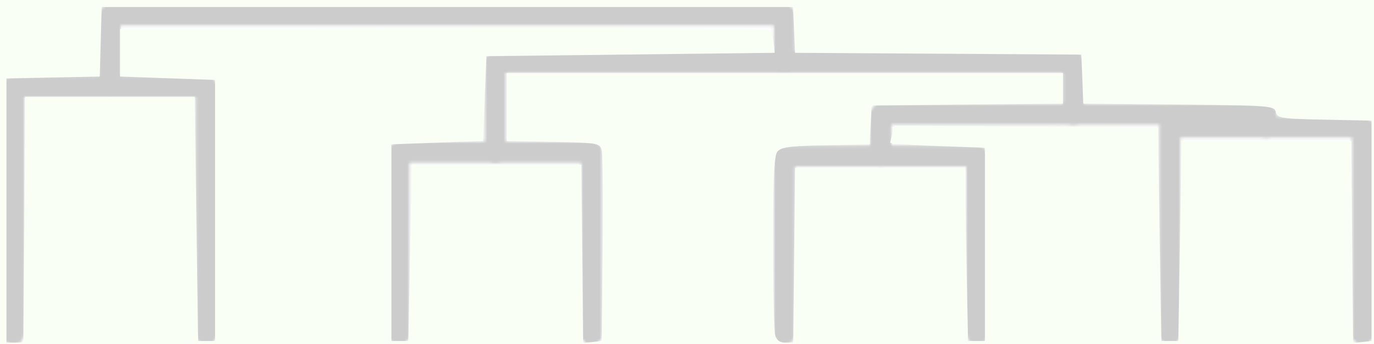


Clustering methods

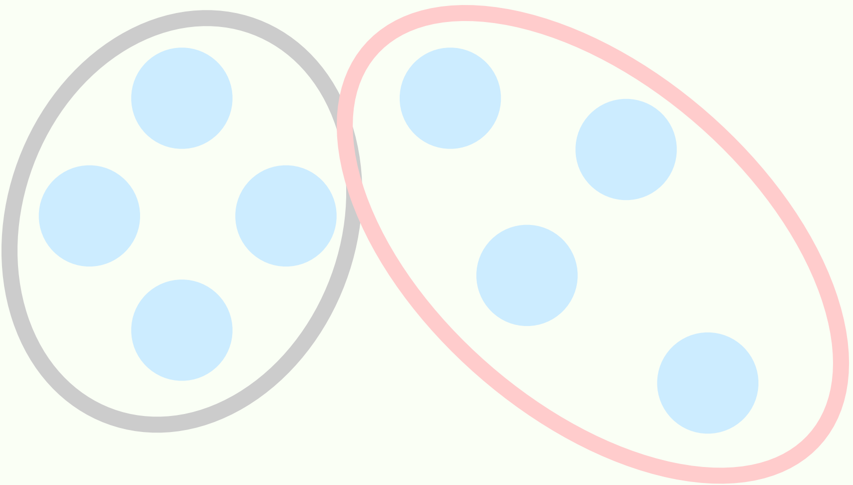
Graph-based



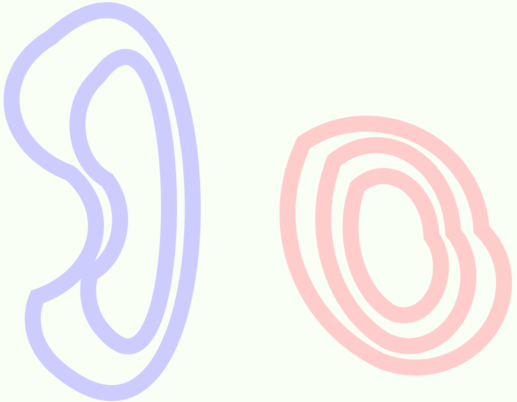
Hierarchical



Centroid-based



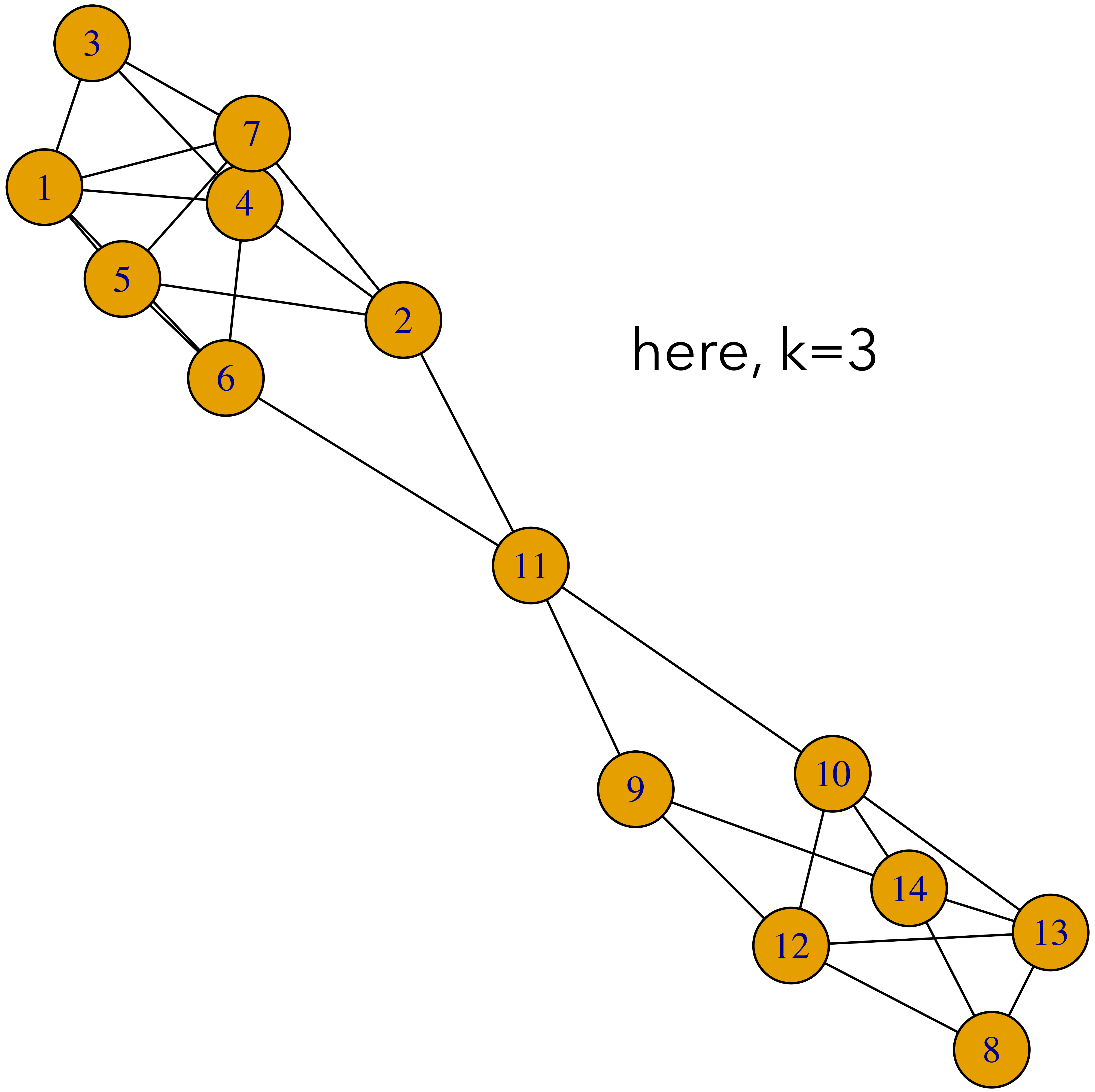
Density-based



Consensus clustering

Graph-based clustering

- Build a graph connecting each node to its k nearest neighbors in the input space
- k controls the resolution - higher k yields a more interconnected graph, and larger clusters



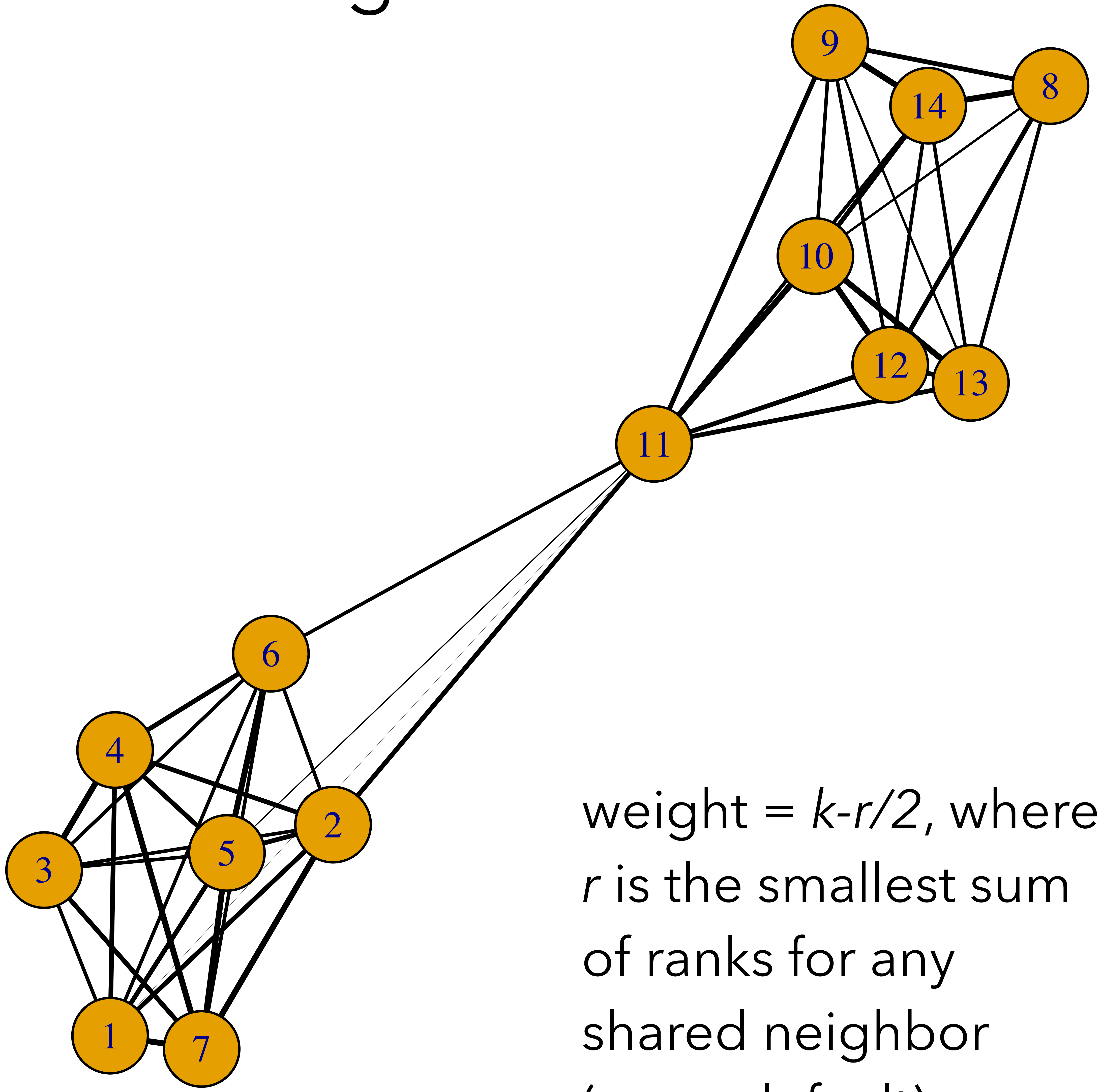
here, k=3

Graph-based clustering

- Next, create the *shared* nearest neighbor (SNN) graph:

- Draw an edge between each pair of cells that share at least one neighbor

- Edges are weighted by the characteristics of the shared nearest neighbors (different options available)
    - more shared neighbors or neighbors that are very close to both cells -> higher weight



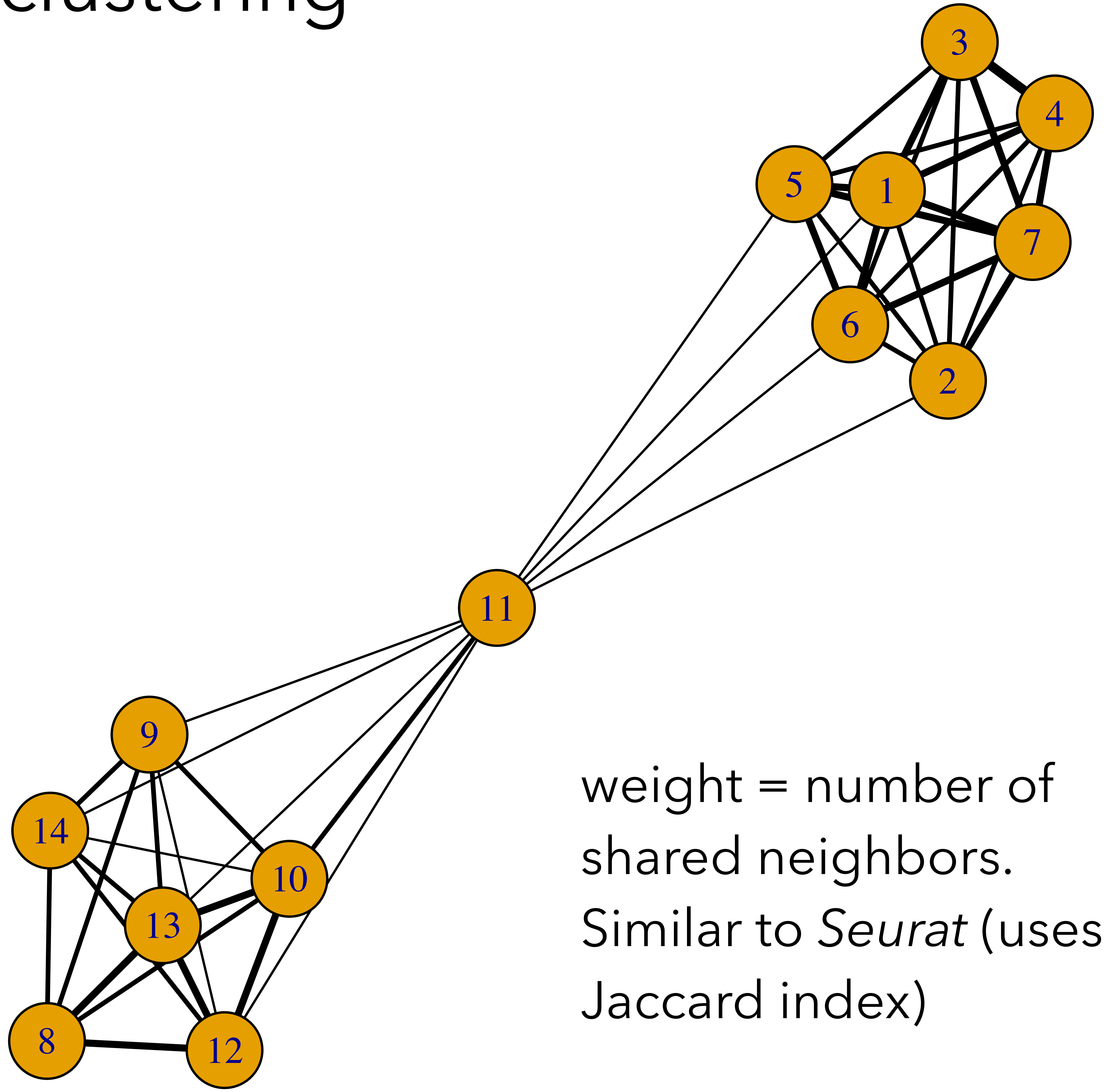
weight = *k-r/2*, where *r* is the smallest sum of ranks for any shared neighbor (*scran* default)

Graph-based clustering

- Next, create the *shared* nearest neighbor (SNN) graph:

- Draw an edge between each pair of cells that share at least one neighbor

- Edges are weighted by the characteristics of the shared nearest neighbors (different options available)
    - more shared neighbors or neighbors that are very close to both cells -> higher weight



weight = number of shared neighbors. Similar to *Seurat* (uses Jaccard index)

Graph-based clustering

- Find clusters using a *community detection* algorithm (many options available)

- The quality of a partition is often measured by its **modularity** (M)

- A network with high modularity has many connections between nodes *within* communities, and few connections *between* communities

- Maximizing modularity is an NP-hard problem -> heuristics!

resolution

$$M = \frac{1}{2m} \sum_{i,j} \left(A_{ij} - \gamma \frac{k_i k_j}{2m} \right) \delta(c_i, c_j)$$

sum of all edge weights

sum over node pairs

weight of edge between *i* and *j*

"probability of an edge between *i* and *j* if edges are randomized"

delta function, =1 if *i* and *j* are in the same community

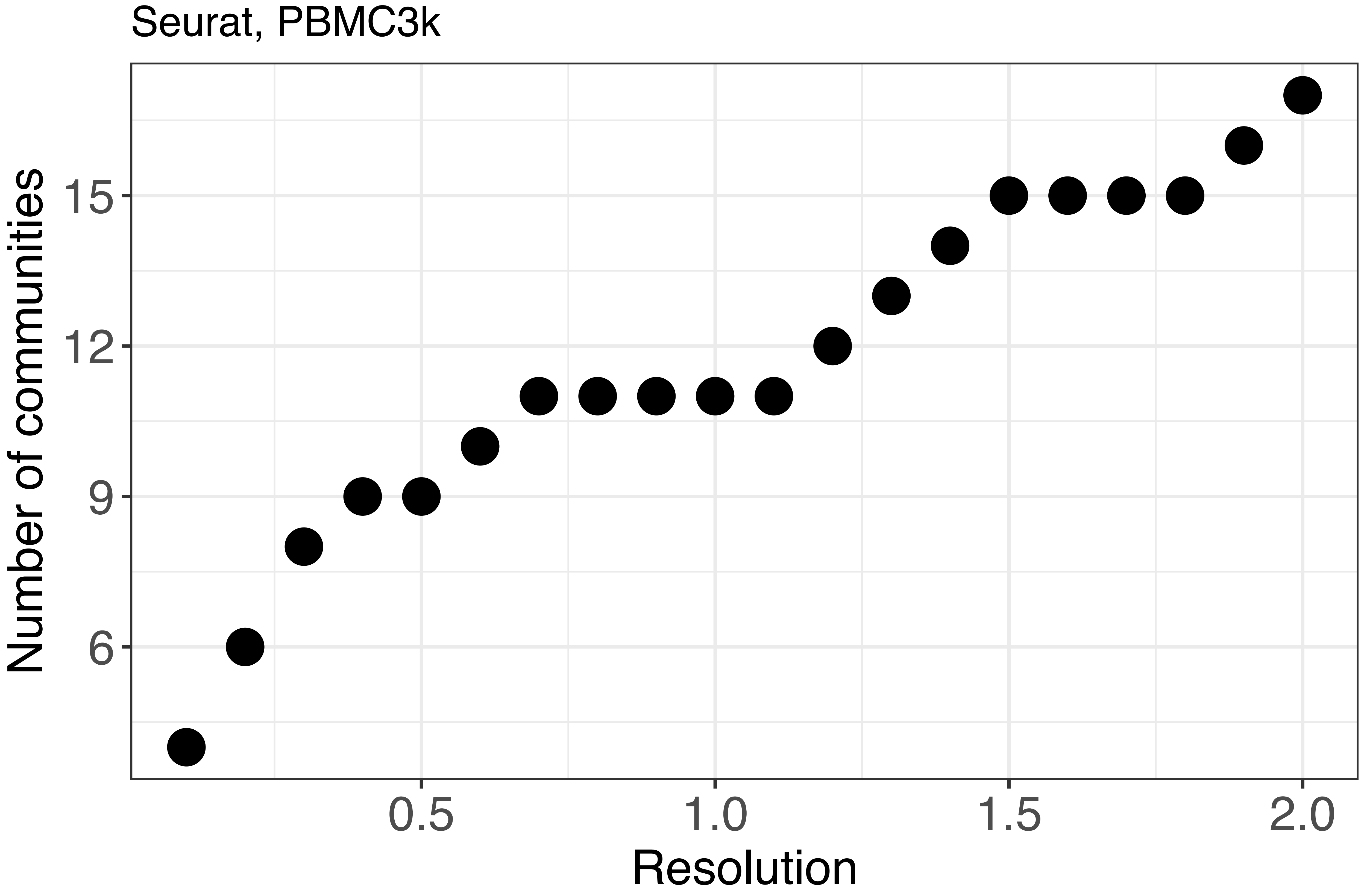
$$k_i = \sum_j A_{ij}$$

Graph-based clustering - number of communities

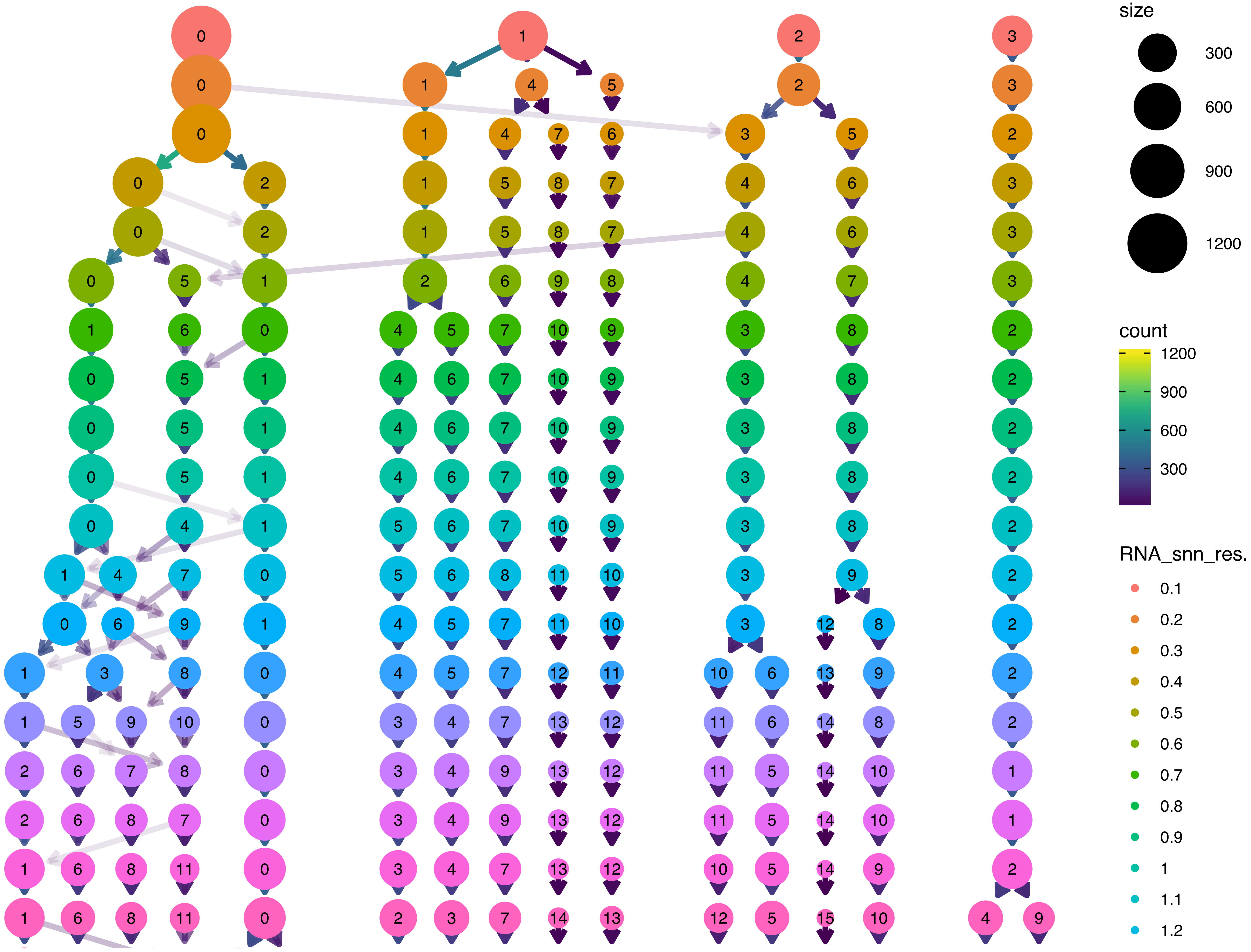
- The number of communities is determined by the resolution parameter (not explicitly set)

- Higher resolution -> more communities

- No strong assumptions on the shape of the communities



Compare different resolutions with clustree



Zappia and Oshlack, GigaScience (2018)

Some community detection algorithms (from igraph)

- **louvain** - aim at optimizing modularity. Start with each cell in its own community, apply a greedy algorithm to move cells between communities to increase modularity. Iterate.

- **walktrap** - attempt to find densely connected subgraphs via random walks, the idea being that short random walks tend to stay in the same community.

- **fast-greedy** - fast, greedy algorithm aimed at optimizing modularity in large networks.

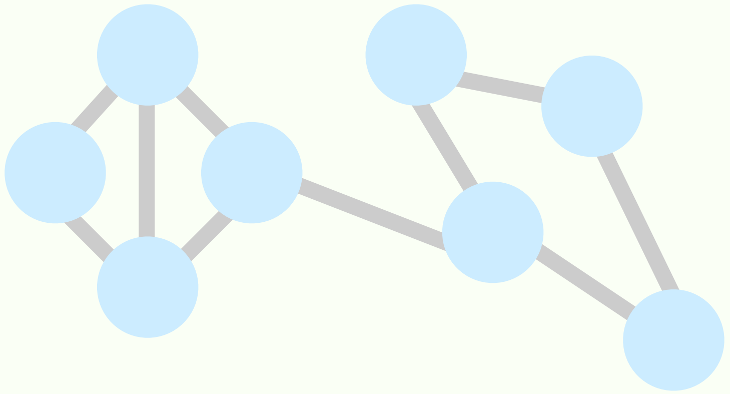
- **edge_betweenness** - successively remove edges with high “betweenness” (= large number of shortest paths going via the edge (“bottlenecks”))

Efficiency of graph-based clustering

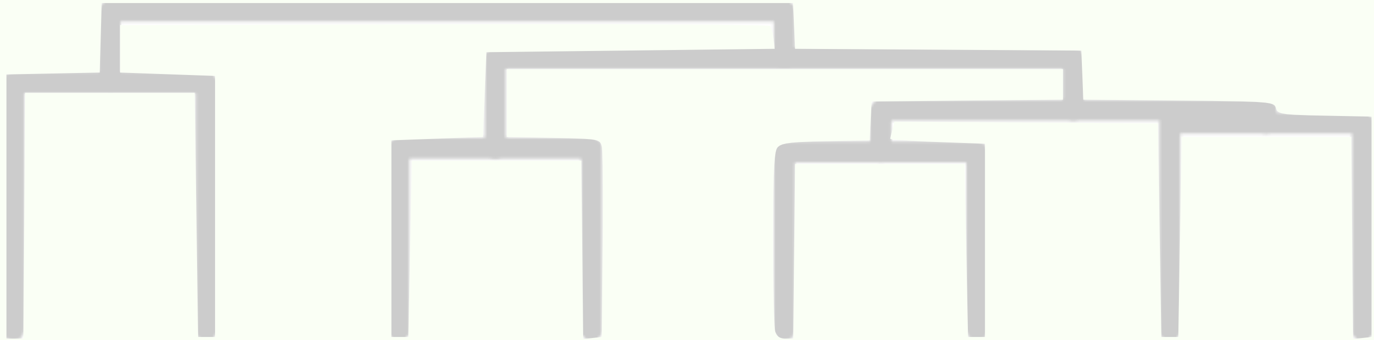
- Requires only a nearest neighbor search
- Does not retain any information beyond the nearest neighbors

Clustering methods

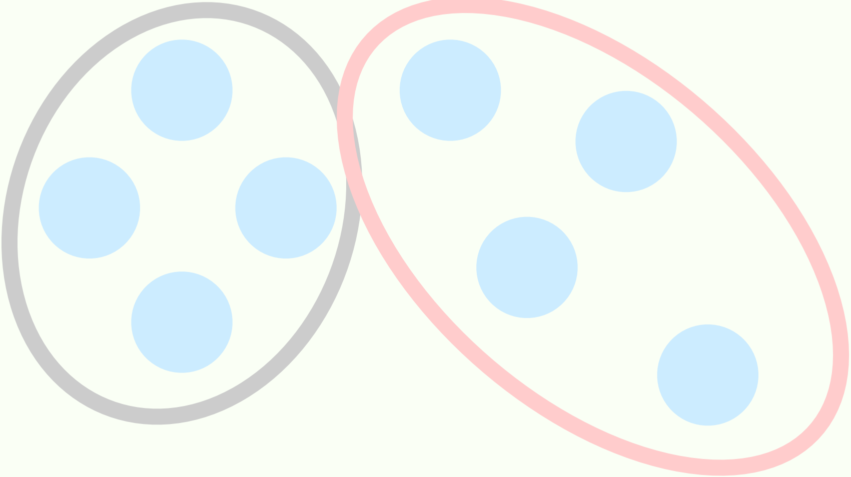
Graph-based



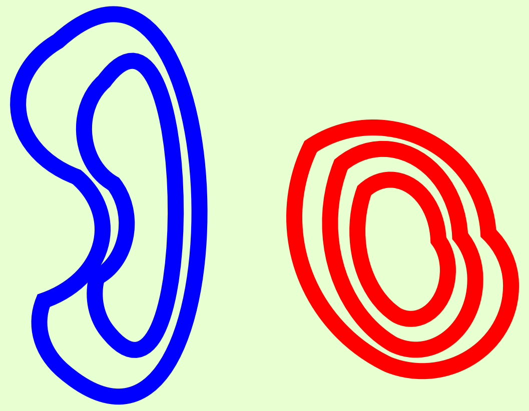
Hierarchical



Centroid-based



Density-based



Consensus clustering

Density-based clustering - DBSCAN

- Given a value *e* and an integer *N*, a point *P* is classified as:
    - A core point, if at least *N* points are within distance *e* from *P*
    - Directly reachable (from *Q*) if it is within distance *e* from a core point *Q*
    - Reachable (from *Q*) if there's a path *Q*,…,*P* where each point is directly reachable from the previous
    - An outlier if it's none of the above
- If *P* is a core point, then it forms a cluster together with all points that are reachable from it

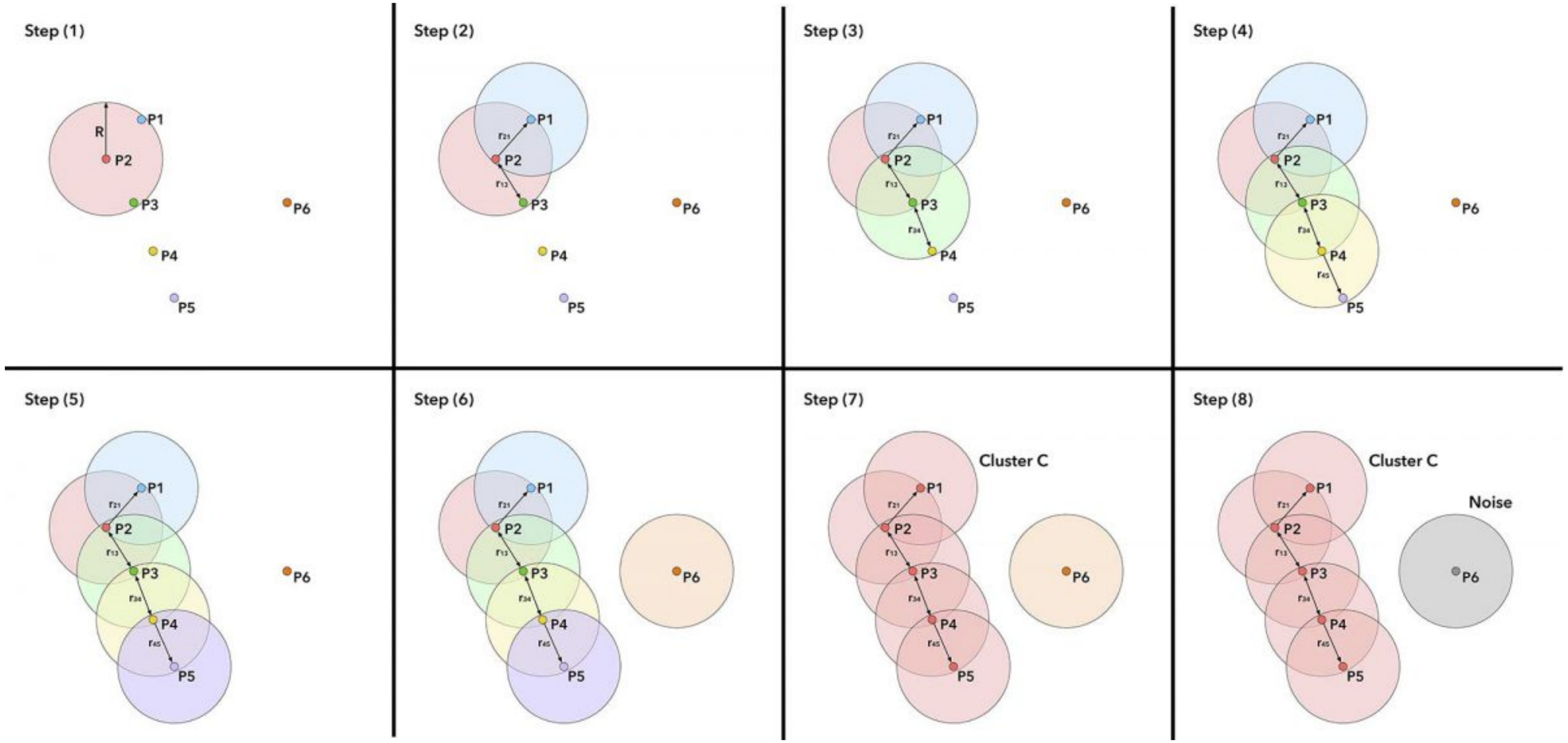
Ester *et al*, Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining (1996)

Density-based clustering - DBSCAN

- Does not require the number of clusters to be specified in advance

- Points are not “forced” to be part of any cluster

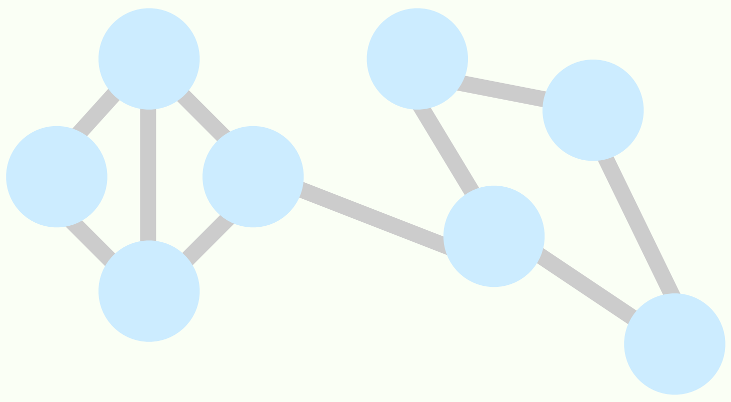
- No strong limitations on cluster shape



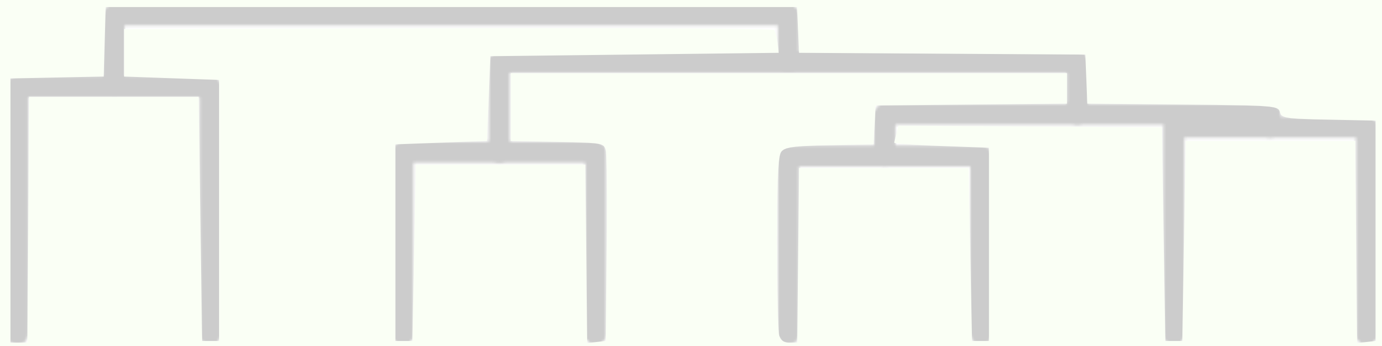
http://www.francescogrigoli.it/tutorial/how-dbscan-clustering-algorithm-works-an-easy-guide-thorugh-sketches.html

Clustering methods

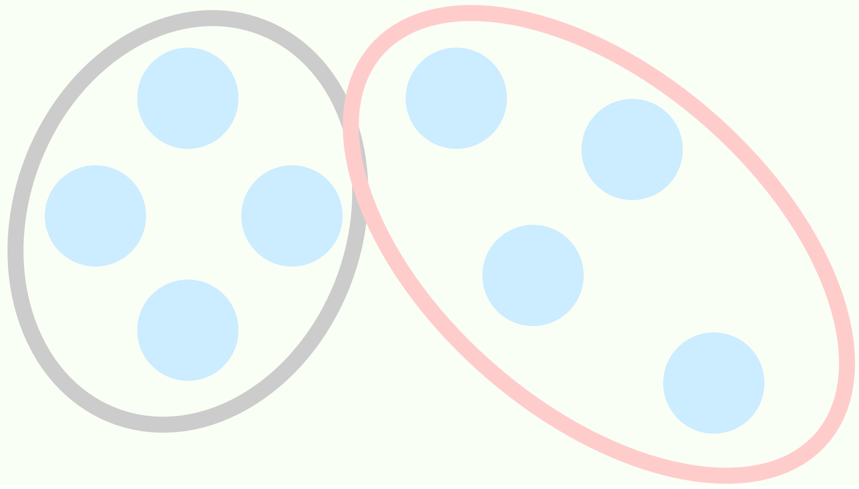
Graph-based



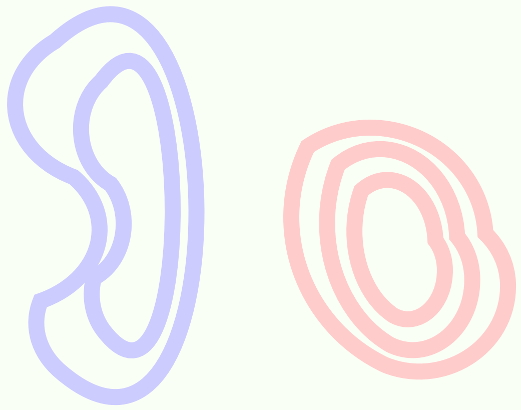
Hierarchical



Centroid-based



Density-based



Consensus clustering

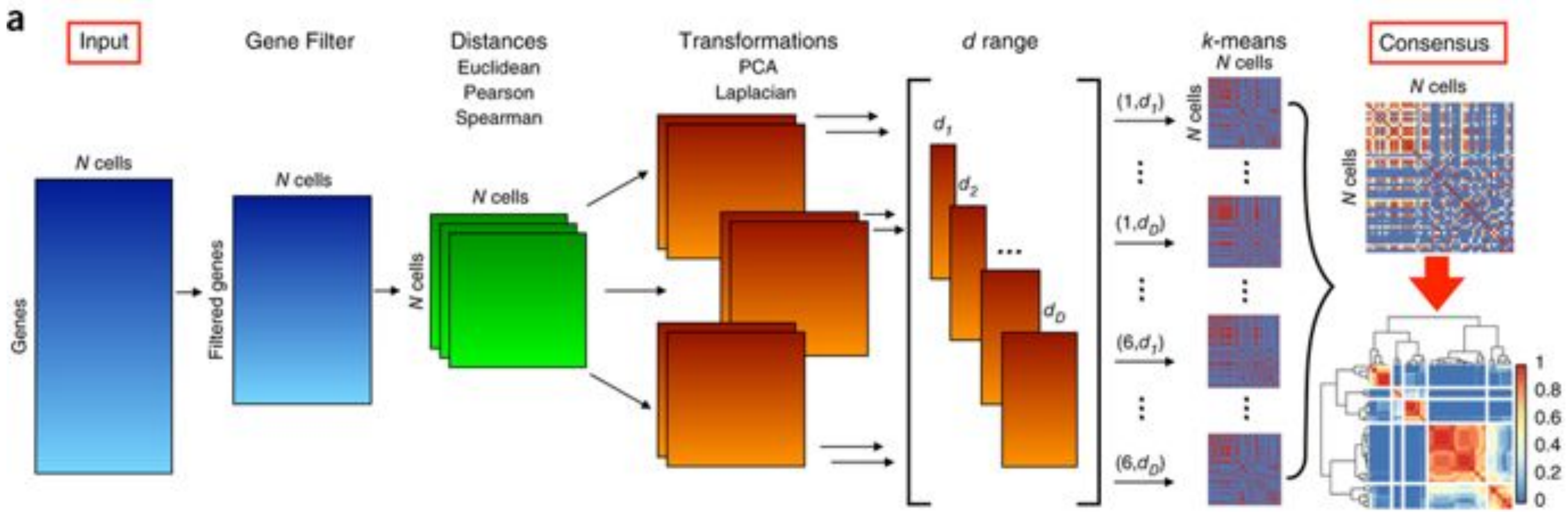
Consensus clustering

- Clustering results can be sensitive to the chosen method, or to the initialization of a given method

- Combining many clustering results may give a more robust partitioning

- The methods that are combined, as well as the way in which they are combined, will influence the accuracy of the consensus

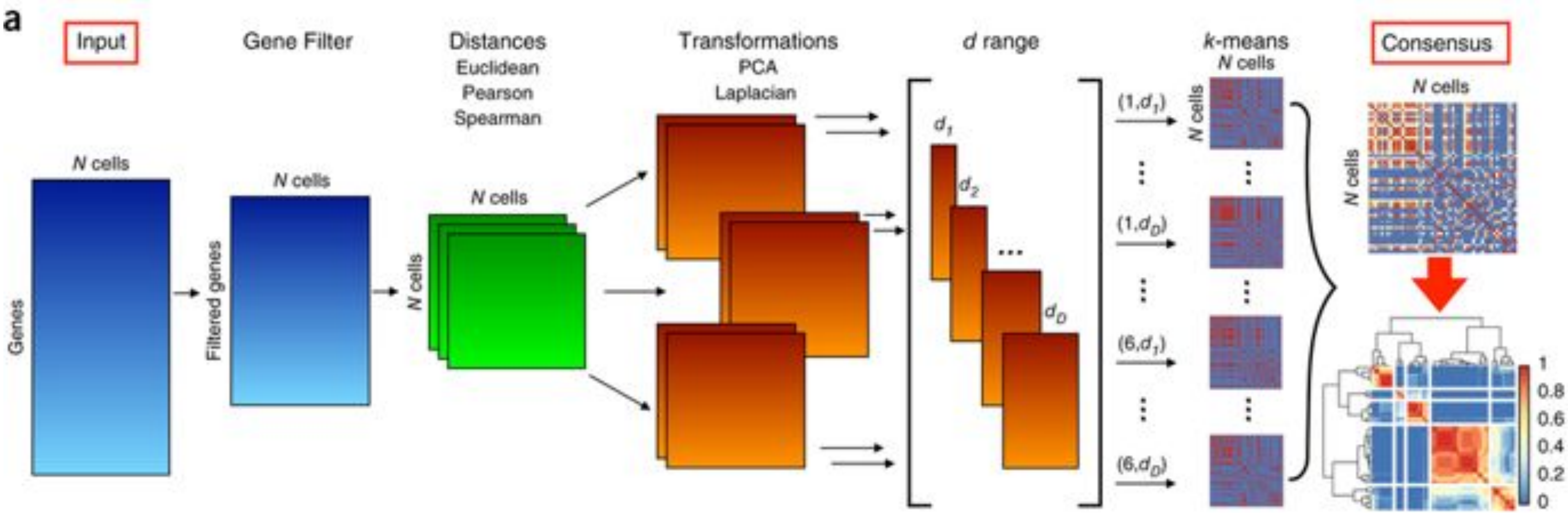
SC3



- Filter out genes that are expressed in very few, or almost all, cells (not informative for the clustering)

Kiselev *et al*, Nature Methods (2017)

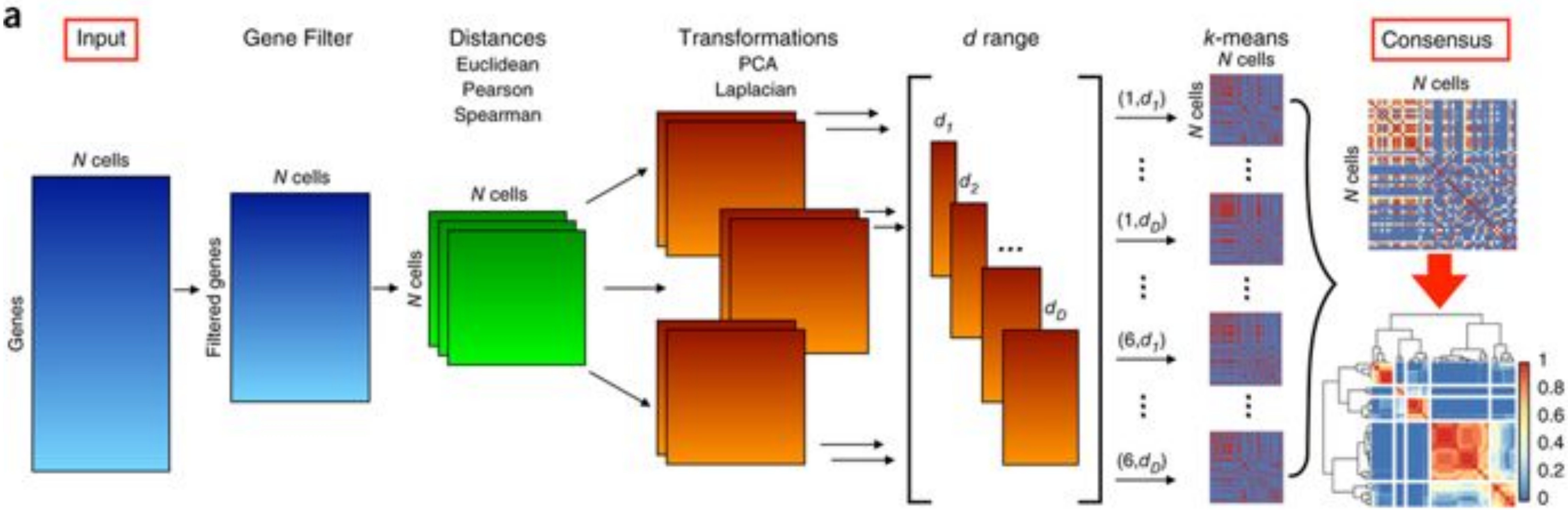
SC3



- Calculate multiple distance/dissimilarity matrices among cells

Kiselev *et al*, Nature Methods (2017)

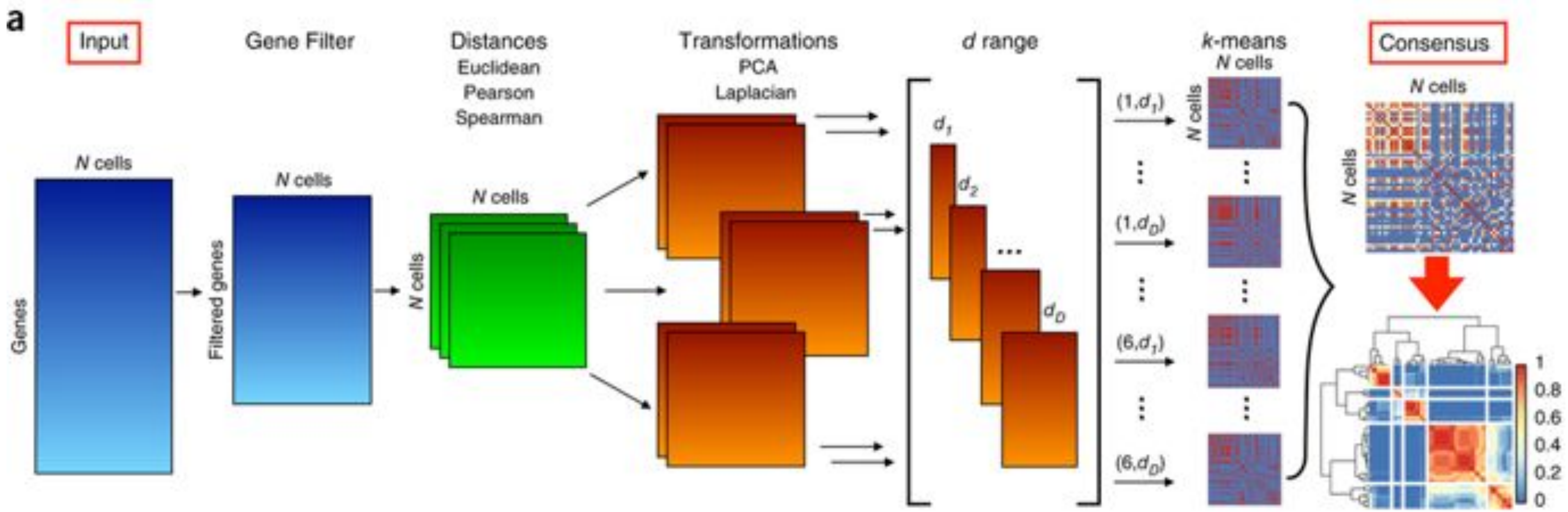
SC3



- Transform distance matrices using either PCA or with the graph Laplacian, sort columns by ascending eigenvalue

Kiselev *et al*, Nature Methods (2017)

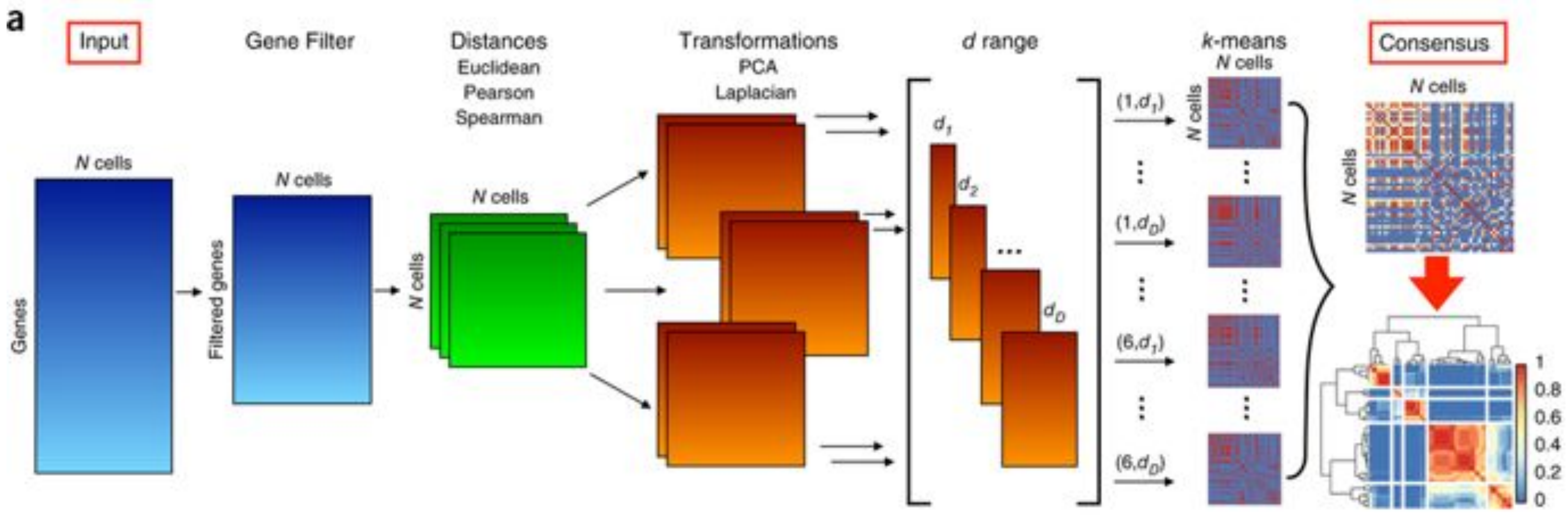
SC3



- Run k-means clustering (with multiple starts) on the first d eigenvectors of the transformed distance matrix.

Kiselev *et al*, Nature Methods (2017)

SC3

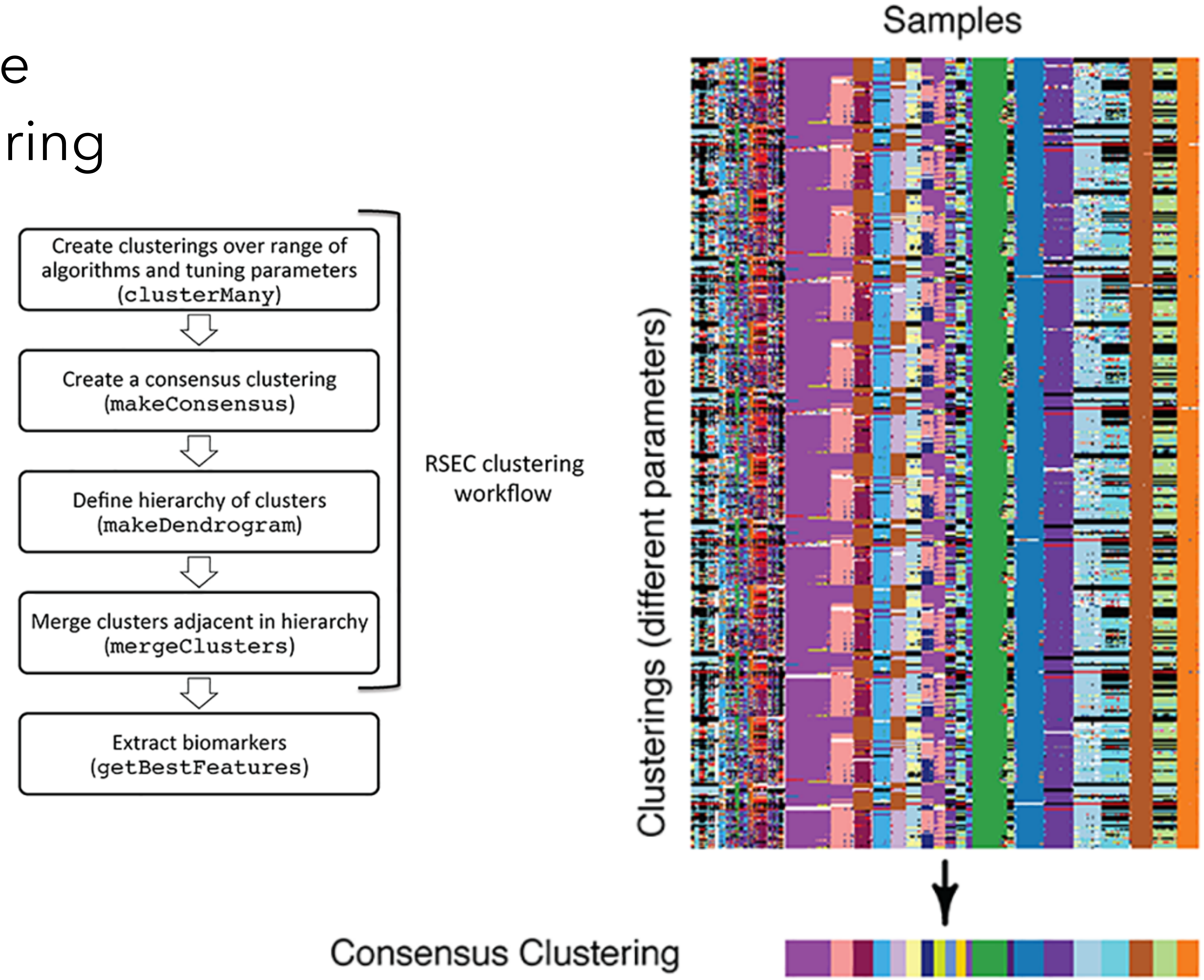


- For each clustering, make a binary similarity matrix (1 = in the same cluster, 0 = in different clusters). Average across all clusterings.
- Cluster the average similarity matrix using hierarchical clustering, and cut the dendrogram.

Kiselev *et al*, Nature Methods (2017)

clusterExperiment

Framework for formalizing the application of multiple clustering algorithms, on differently processed input data, with different number of clusters, and generation of a consensus.



Create clusterings over range of algorithms and tuning parameters (`clusterMany`)

Create a consensus clustering (`makeConsensus`)

Define hierarchy of clusters (`makeDendrogram`)

Merge clusters adjacent in hierarchy (`mergeClusters`)

Extract biomarkers (`getBestFeatures`)

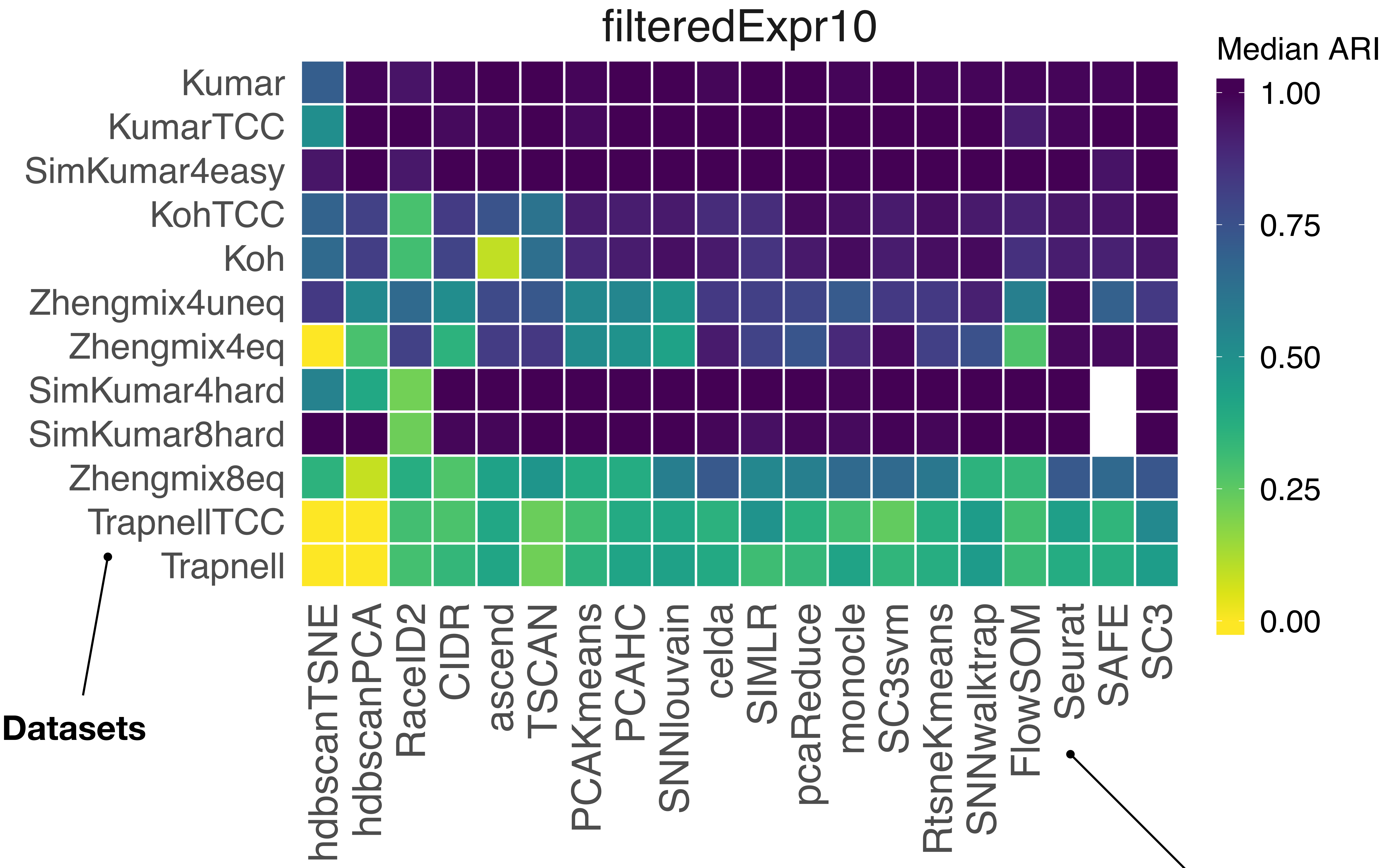
RSEC clustering workflow

Risso *et al*, PLoS Computational Biology (2018)

A couple of observations

- Consensus methods (SC3, SAFE) are among the top performing clustering methods

The k that gives the best agreement may differ between methods!



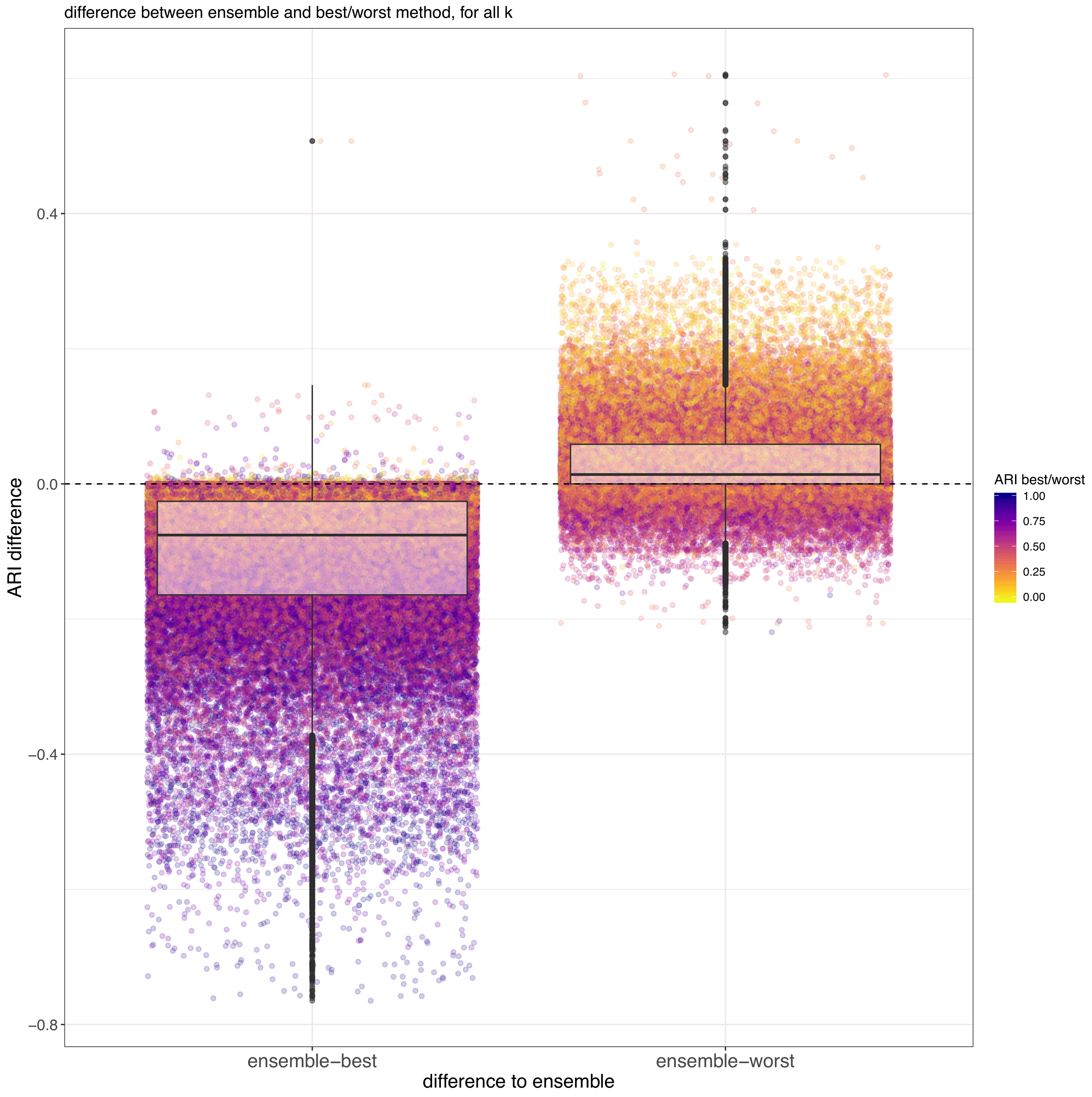
Datasets

Methods

Modified from Duò *et al*, F1000Research (2018)

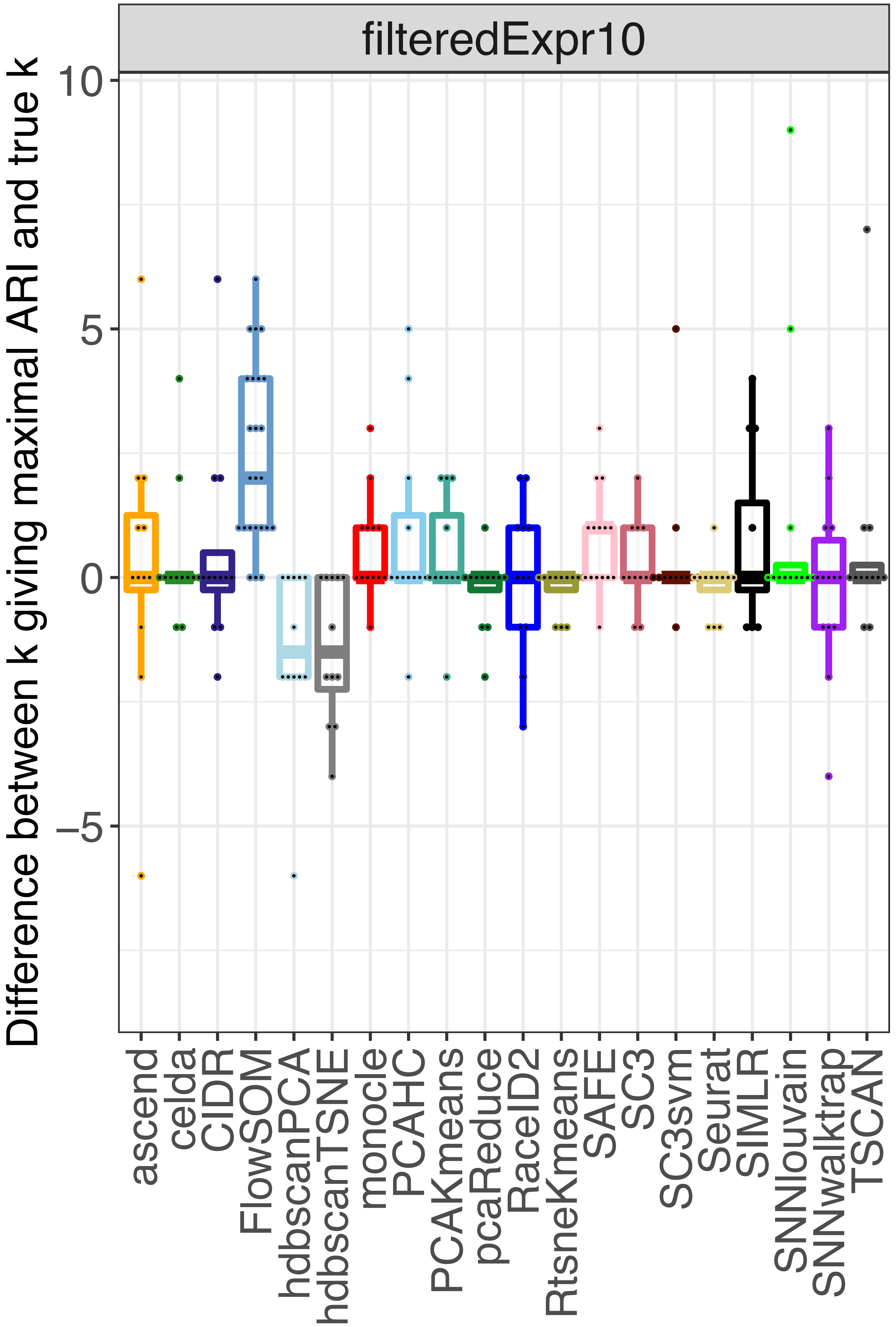
A couple of observations

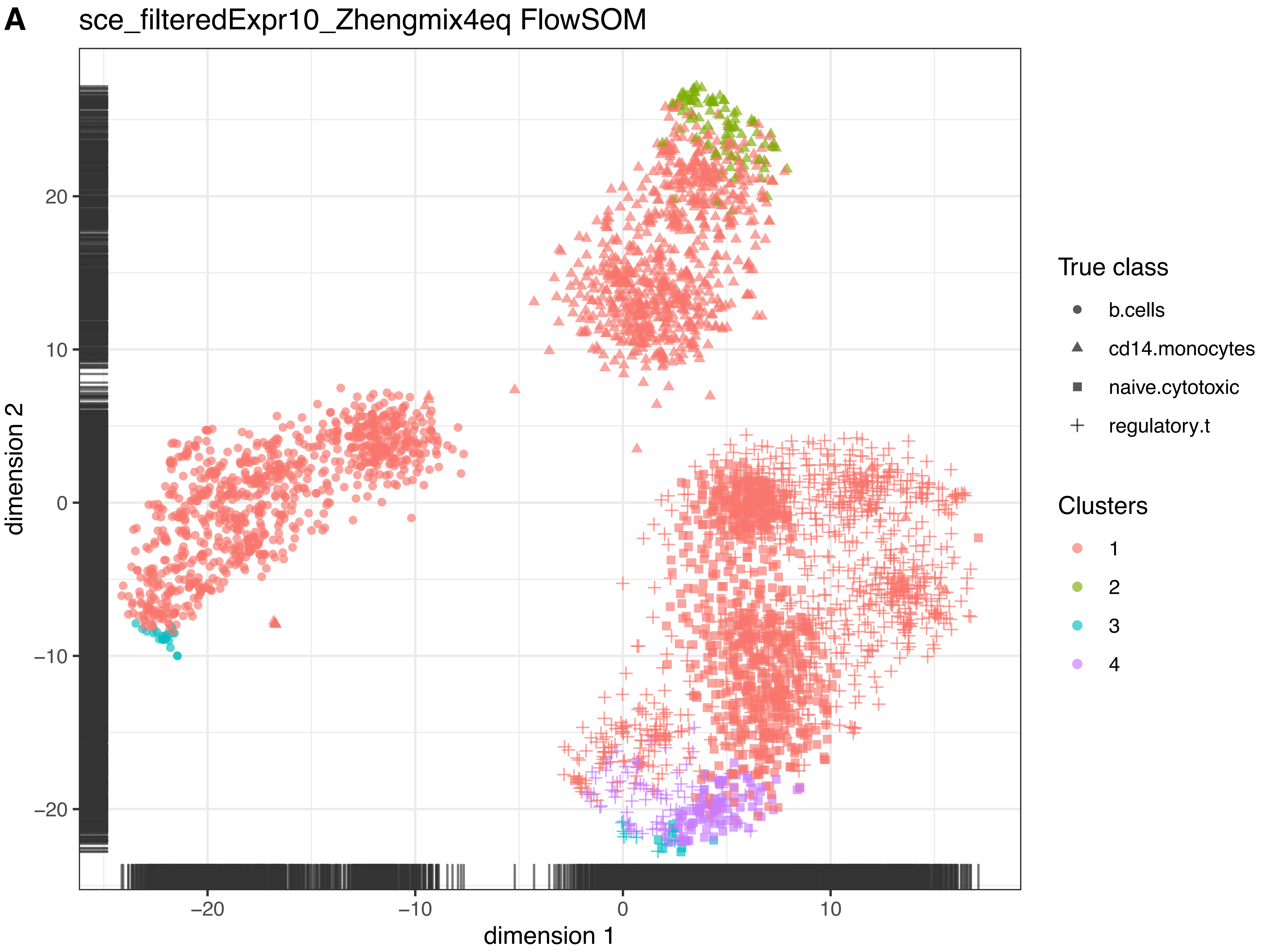
- Ensembles (of two methods) rarely performed better than the best of the individual methods (**but** of course we don’t know which method that is from the beginning!)



Modified from Duò *et al*, F1000Research (2018)

The “best” k is not always the “true” k



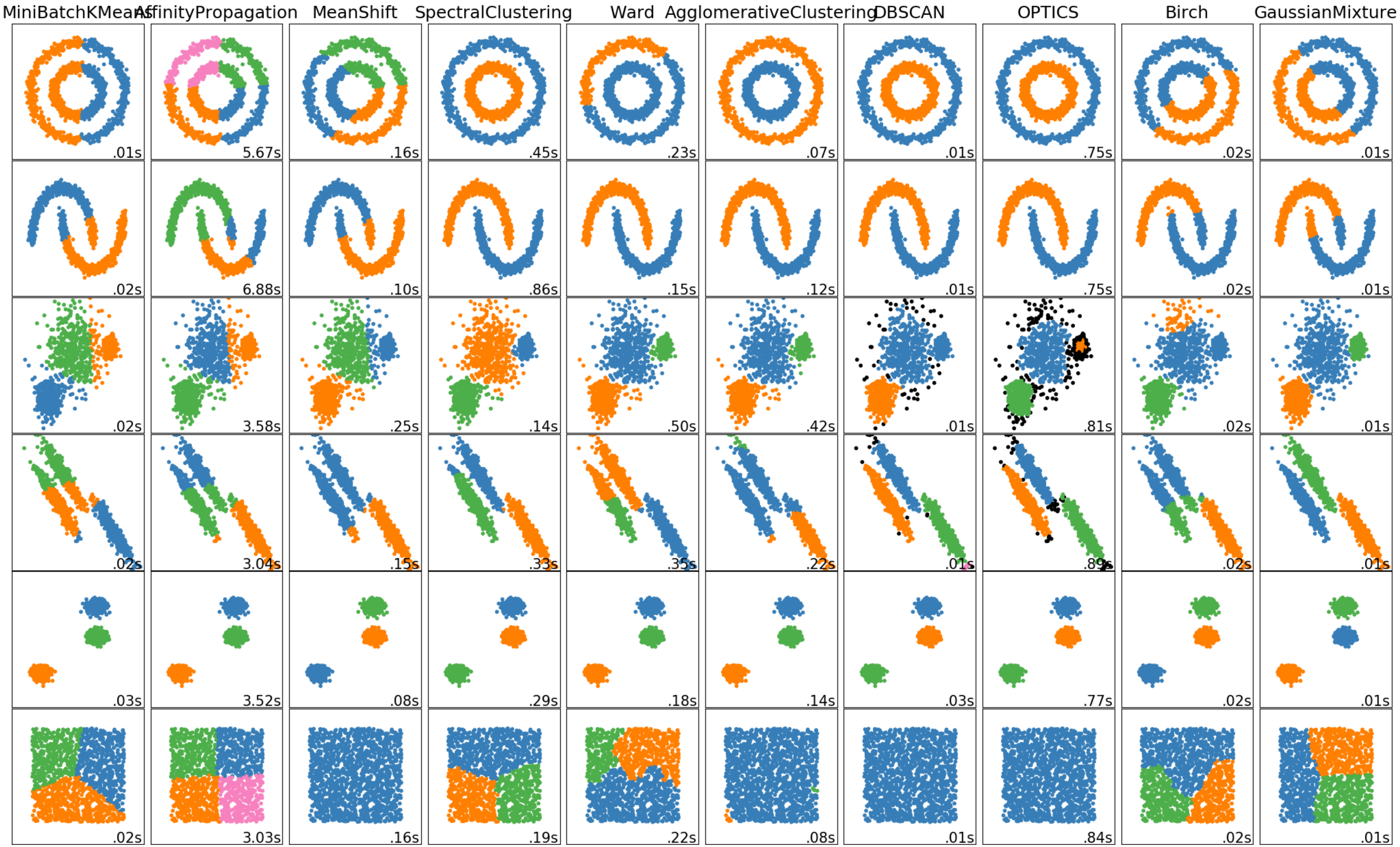


Modified from Duò *et al*, F1000Research (2018)

Interpretation of clusters

- Often many different partitions are “correct” or “interpretable”
    - Different resolutions (major vs more specific cell types)
    - Different types of signals or aspects of the data

Interpretation of clusters



https://scikit-learn.org/stable/auto_examples/cluster/plot_cluster_comparison.html

References

- Amezquita *et al*: Orchestrating single-cell analysis with Bioconductor. bioRxiv doi:10.1101/590562 (2019)
- Blondel *et al*: Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment (2008)
- Clauset *et al*: Finding community structure in very large networks. arXiv:cond-mat/0408187
- Duò *et al*: A systematic performance evaluation of clustering methods for single-cell RNA-seq data. F1000Research 7:1141 (2018)
- Ester *et al*: A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining (1996)
- Freytag *et al*: Comparison of clustering tools in R for medium-sized 10x Genomics single-cell RNA-sequencing data. F1000Research 7:1297 (2018)
- Kiselev *et al*: SC3: consensus clustering of single-cell RNA-seq data. Nature Methods 14:483-486 (2017)
- Newman: Analysis of weighted networks. arXiv:cond-mat/0407503v1 (2004)
- Pons and Latapy: Computing communities in large networks using random walks. arXiv:physics/0512106 (2005)
- Rousseeuw: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics 20:53-65 (1987)
- Tibshirani *et al*: Estimating the number of clusters in a data set via the gap statistic. Journal of the Royal Statistical Society Series B 63:411-423 (2001)
- Zappia and Oshlack: Clustering trees: a visualization for evaluating clusters at multiple resolutions. GigaScience 7(7):giy083 (2018)