Clustering

Aaron Meyer

Outline

- Administrative Issues
- Clustering
  - K-Means
  - Agglomerative
- Clustering examples
- Implementation

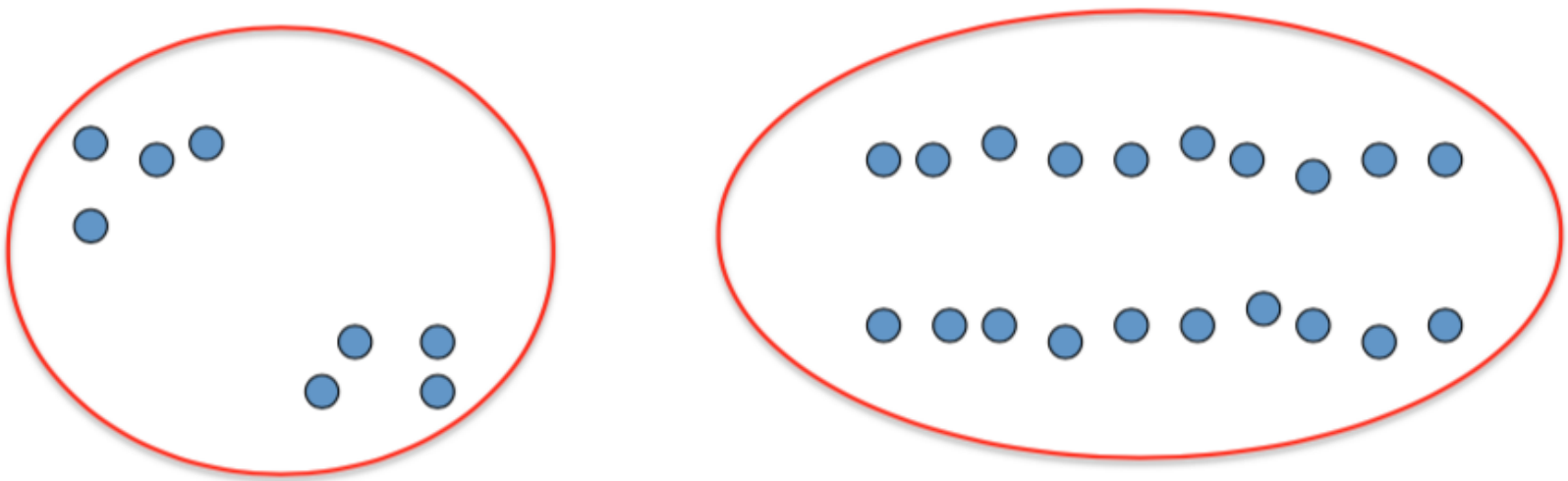
Slides adapted from David Sontag.

Clustering

- Unsupervised learning
- Requires data, but no labels
- Detect patterns e.g. in
  - Gene expression between patient samples
  - Images
  - Really any sets of measurements across samples
- Useful when don't know what you're looking for
- But: **can be gibberish**

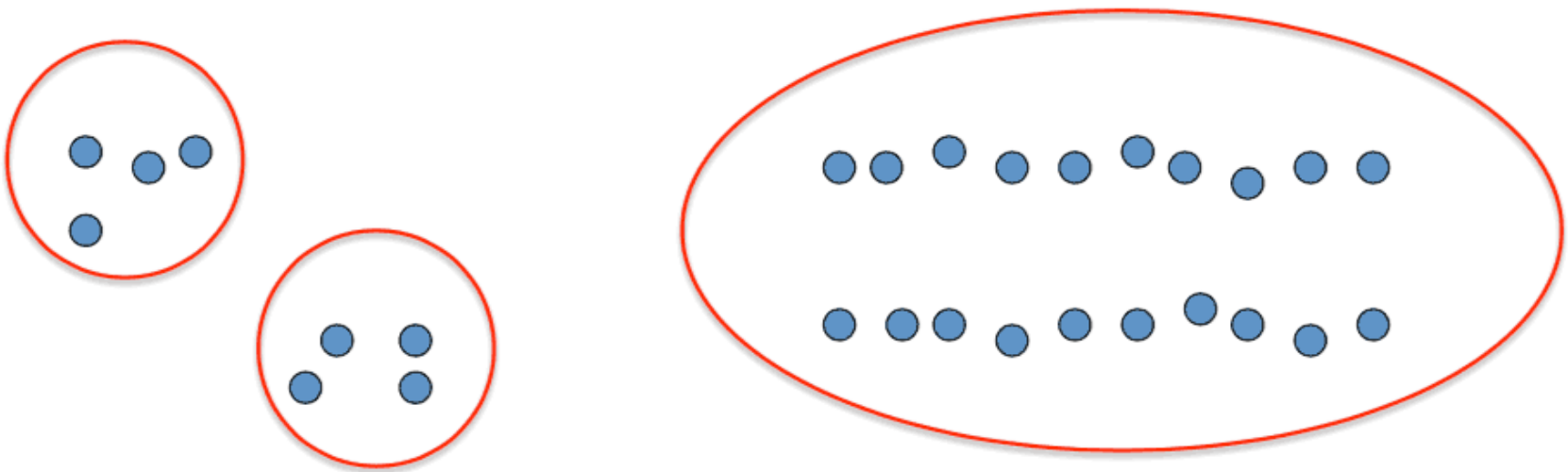
Clustering

- Basic idea: group together similar instances
- Example: 2D point patterns



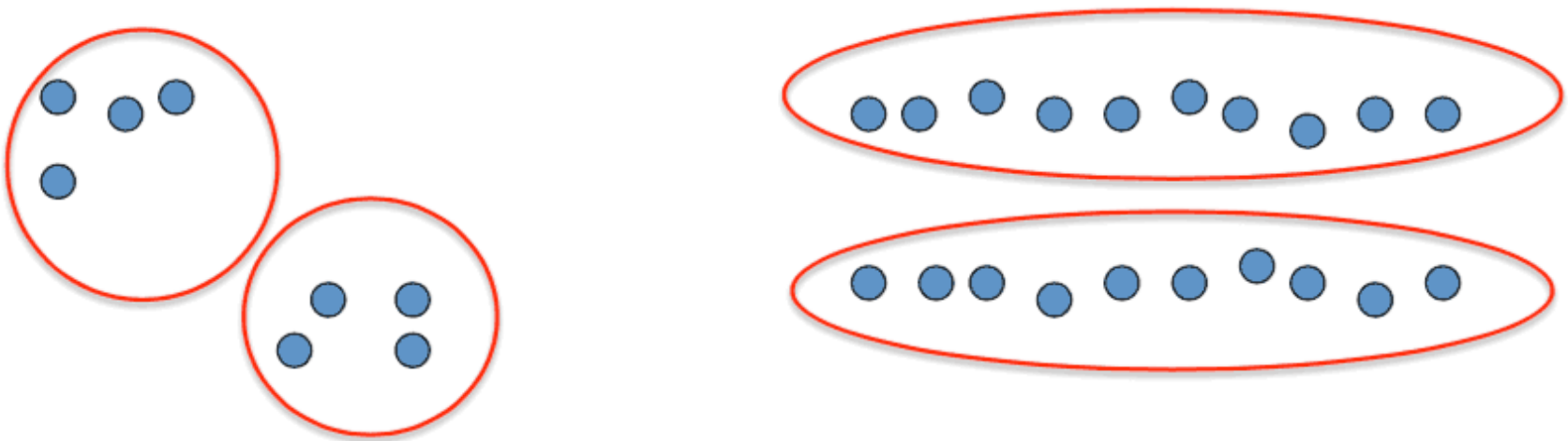
Clustering

- Basic idea: group together similar instances
- Example: 2D point patterns



Clustering

- Basic idea: group together similar instances
- Example: 2D point patterns



What could "similar" mean?

- One option: Euclidean distance

$$\text{dist}(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|$$

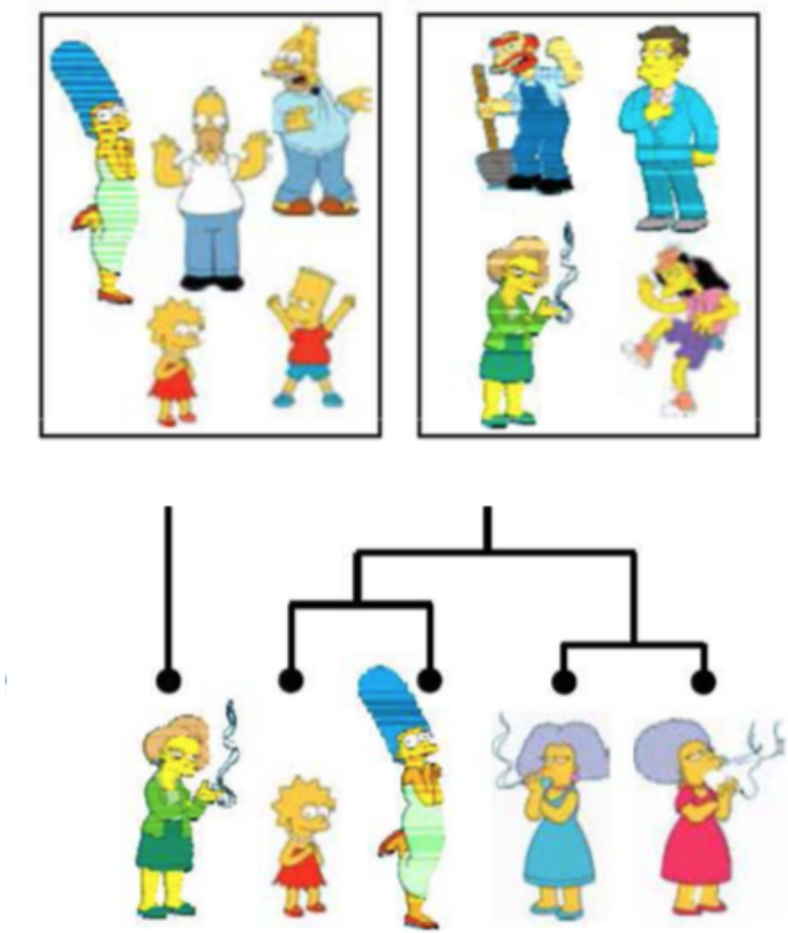
- Clustering results are **completely** dependent on the measure of similarity (or distance) between "points" to be clustered

What properties should a distance measure have?

- Symmetric
  - $D(A, B) = D(B, A)$
  - Otherwise we can say A looks like B but not vice-versa
- Positivity and self-similarity
  - $D(A, B) > 0$, and $D(A, B) = 0$ iff $A = B$
  - Otherwise there will be objects that we can't tell apart
- Triangle inequality
  - $D(A, B) + D(B, C) \geq D(A, C)$
  - Otherwise one can say "A is like B, B is like C, but A is not like C at all"

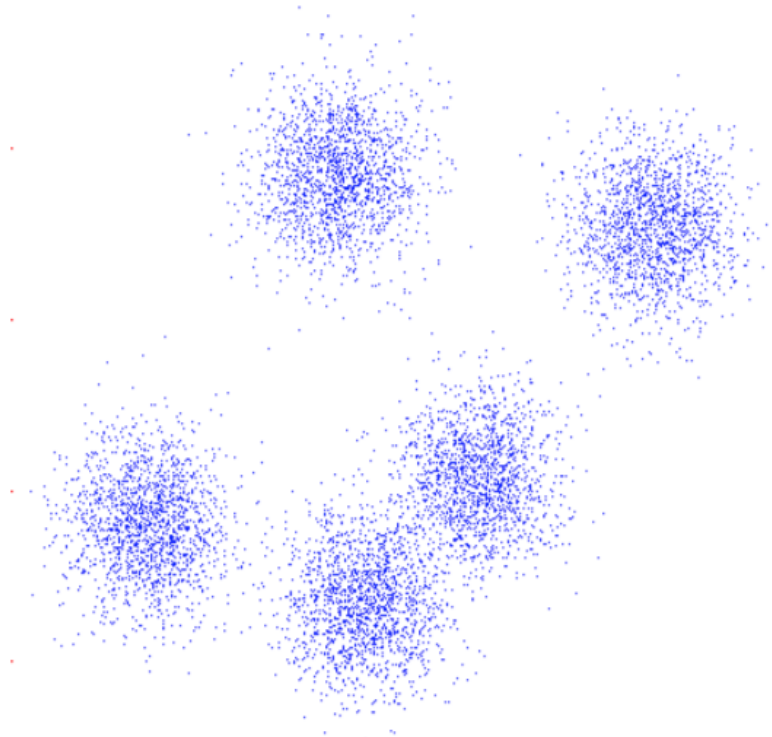
Clustering algorithms

- Partition algorithms (flat)
  - K-means
  - Gaussian mixtures
- Heirarchical algorithms
  - Bottom up - agglomerative
  - Top down - divisive



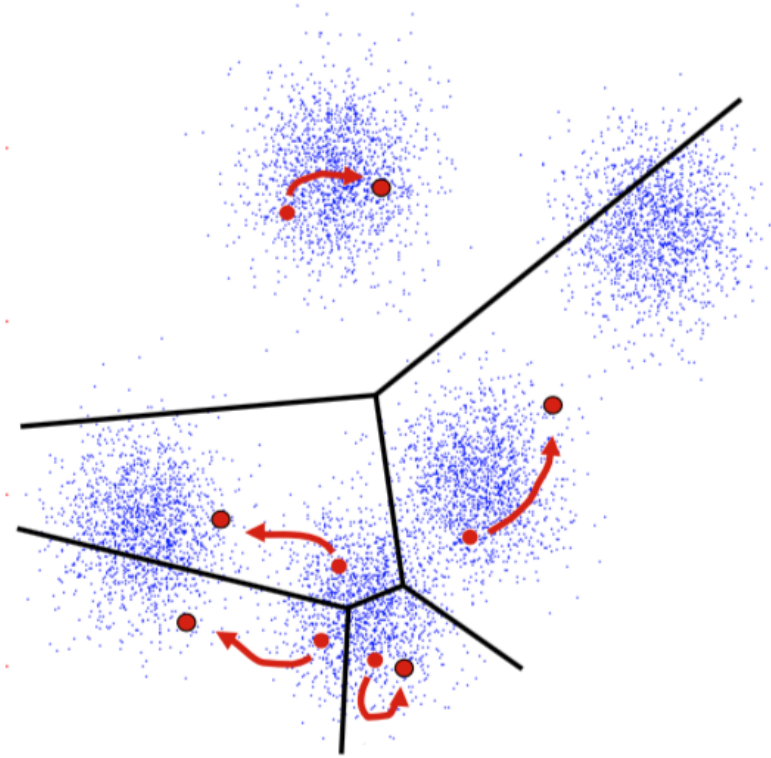
K-means, an iterative clustering algorithm

- Initialize: Pick K random points as cluster centers
- Alternate until no assignments change:
  1. Assign data points to closest cluster center
  2. Change the cluster center to the average of its assigned points



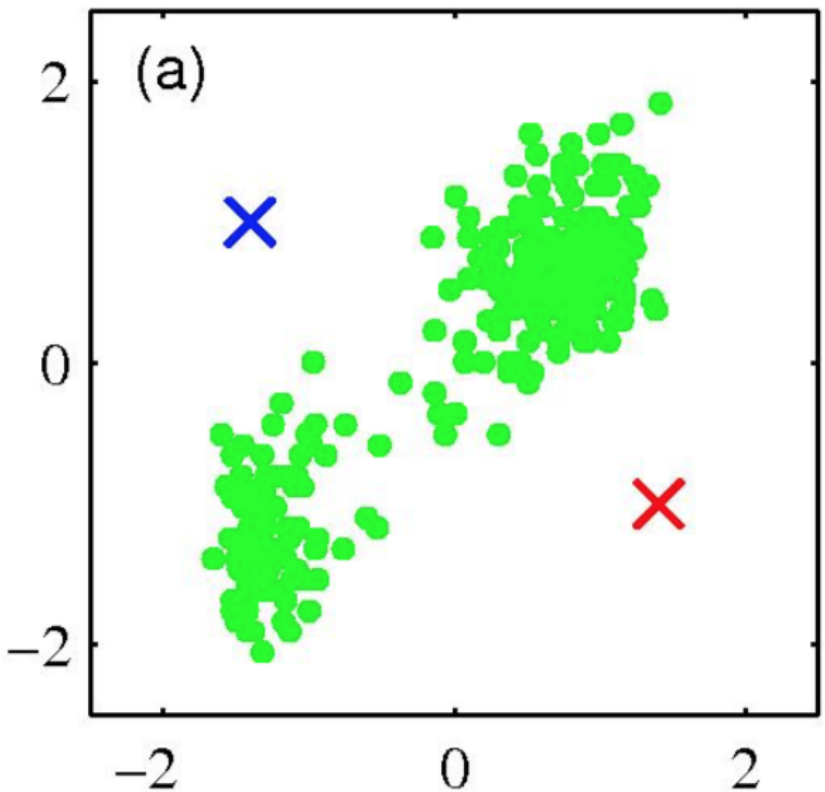
K-means, an iterative clustering algorithm

- Initialize: Pick K random points as cluster centers
- Alternate until no assignments change:
  1. Assign data points to closest cluster center
  2. Change the cluster center to the average of its assigned points



K-means clustering: Example

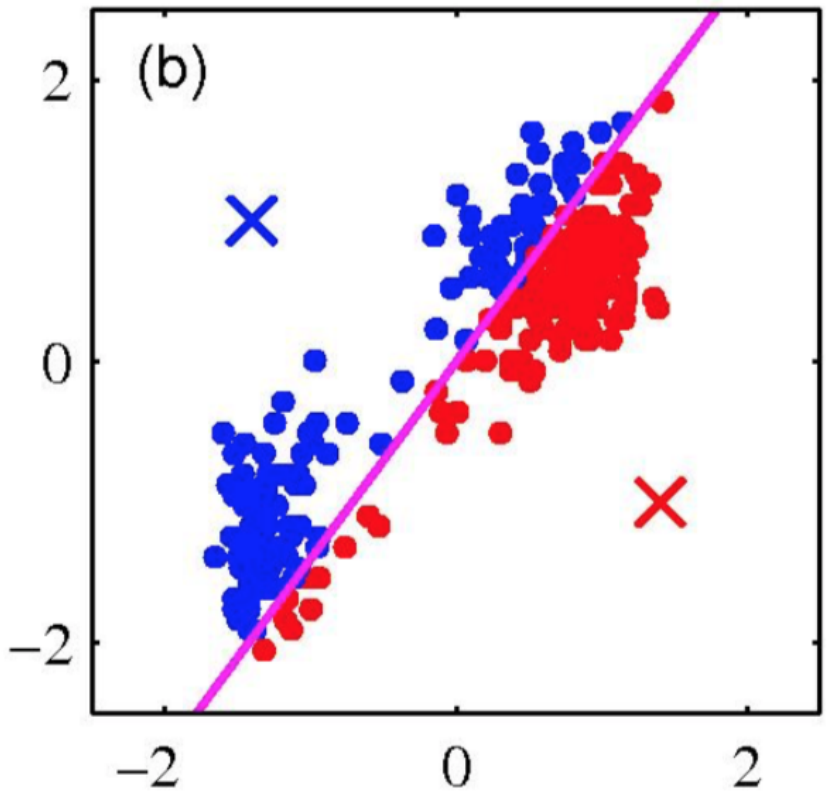
- Pick K random points as cluster centers (means)
  - Shown here for K=2



K-means clustering: Example

Iterative Step 1

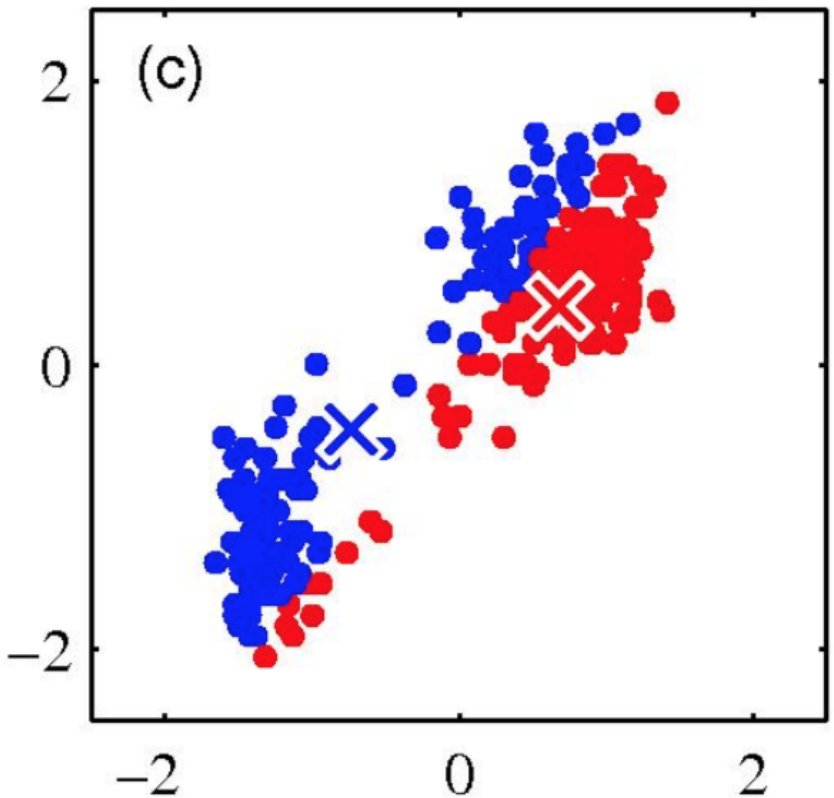
Assign data points to closest cluster center



K-means clustering: Example

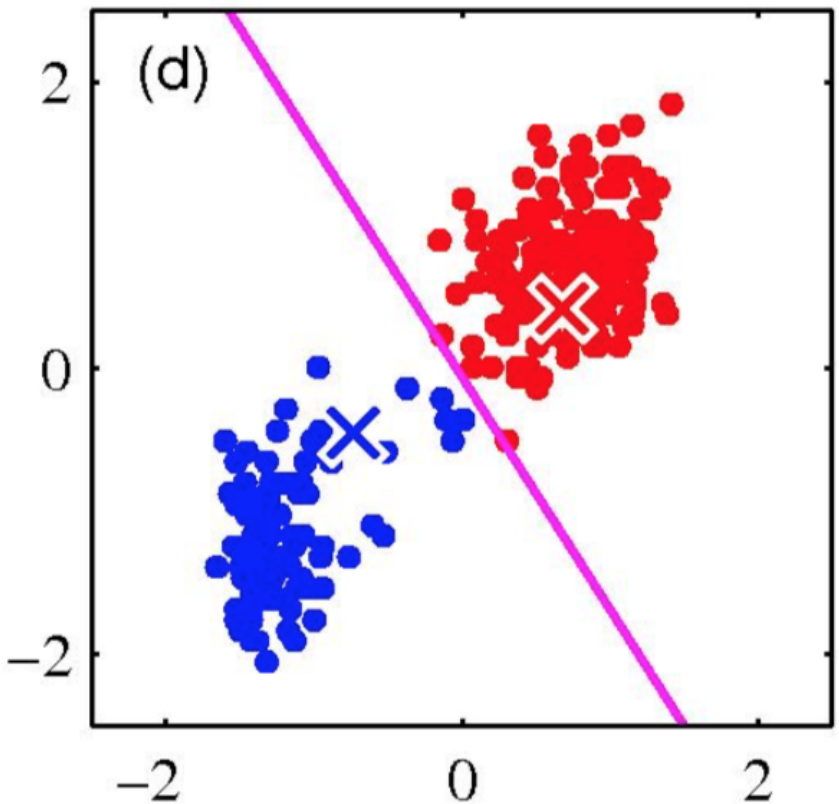
Iterative Step 2

Change the cluster center to the average of the assigned points

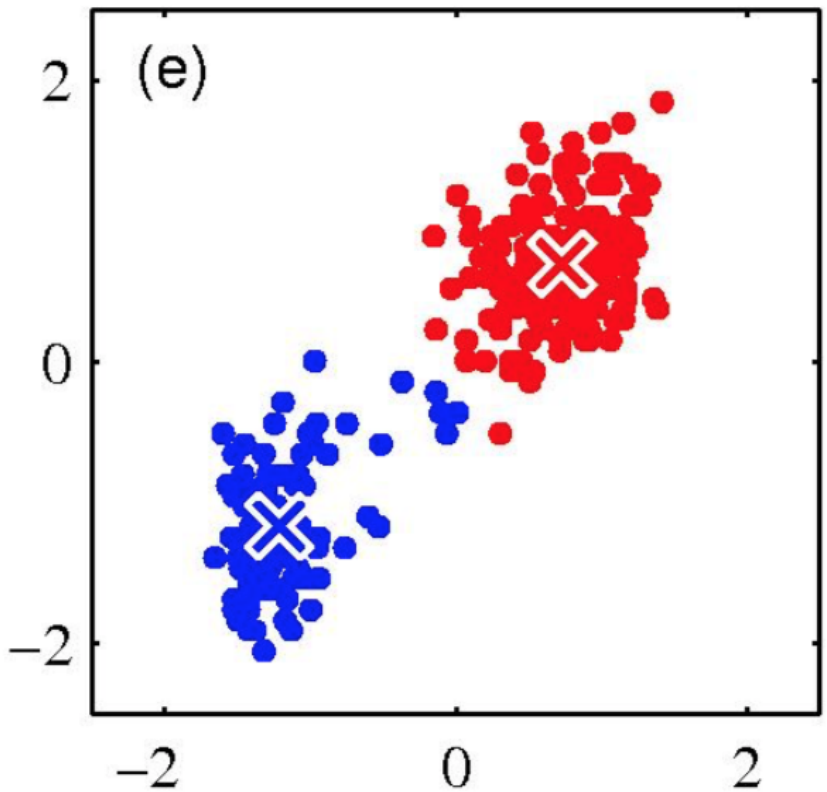


K-means clustering: Example

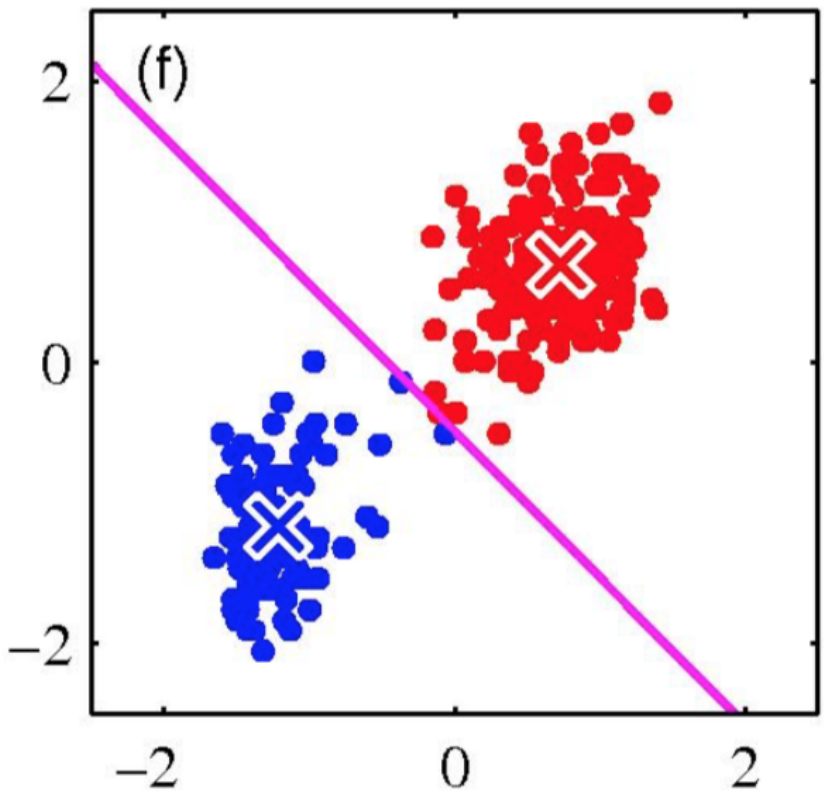
Repeat until convergence



K-means clustering: Example



K-means clustering: Example



Another Example

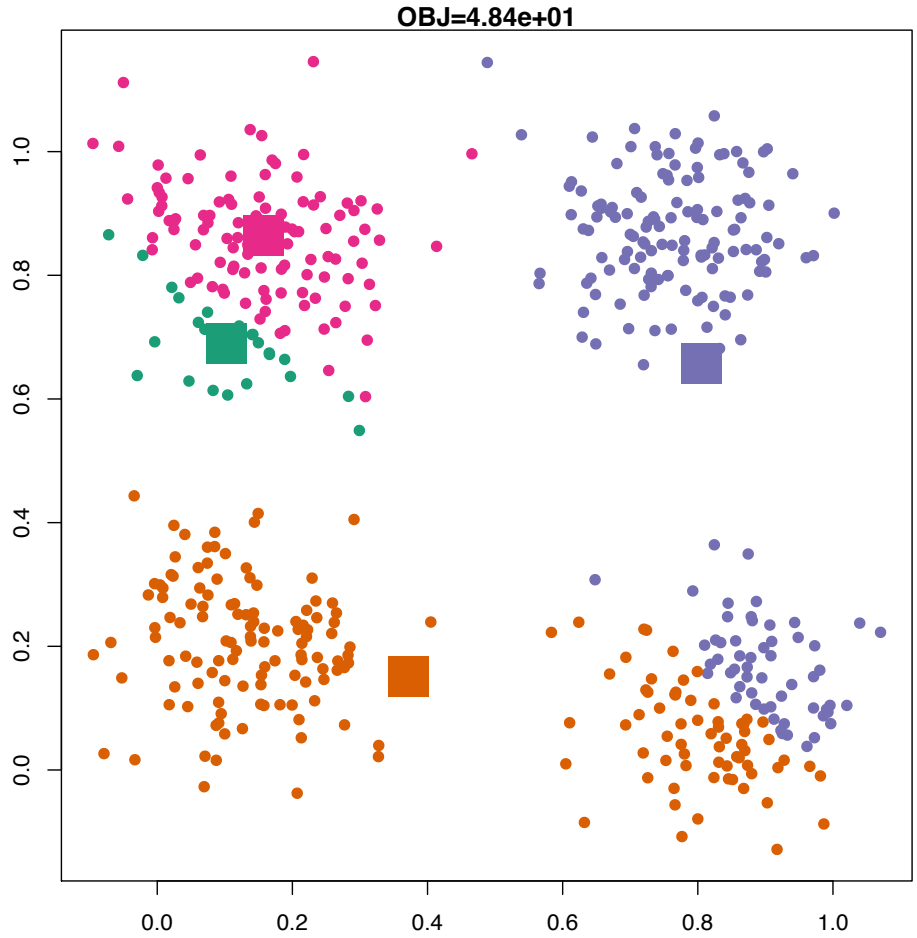


Figure: Step 1

Another Example

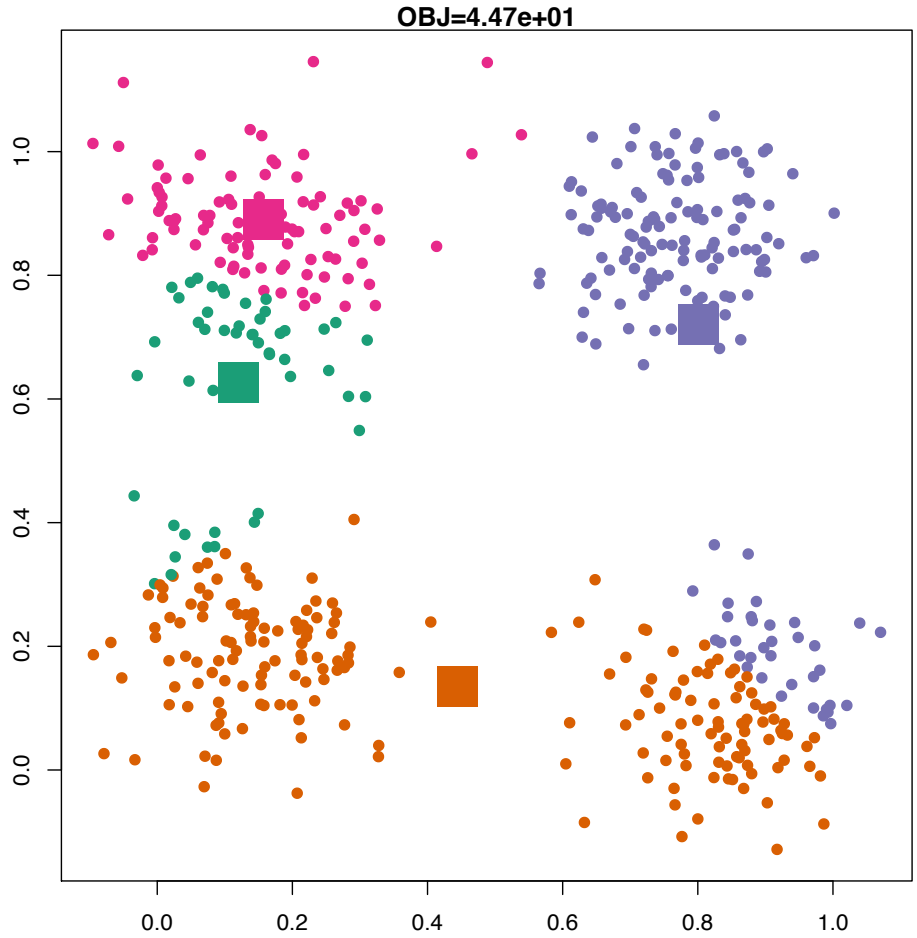


Figure: Step 2

Another Example

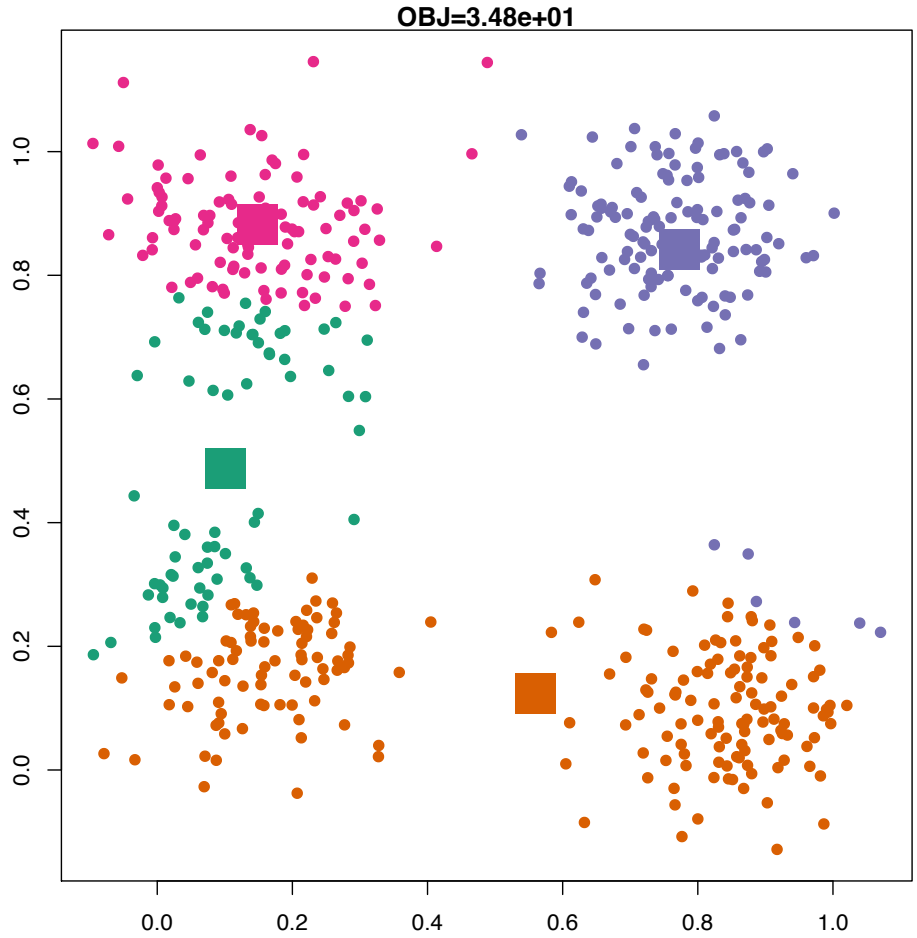


Figure: Step 3

Another Example

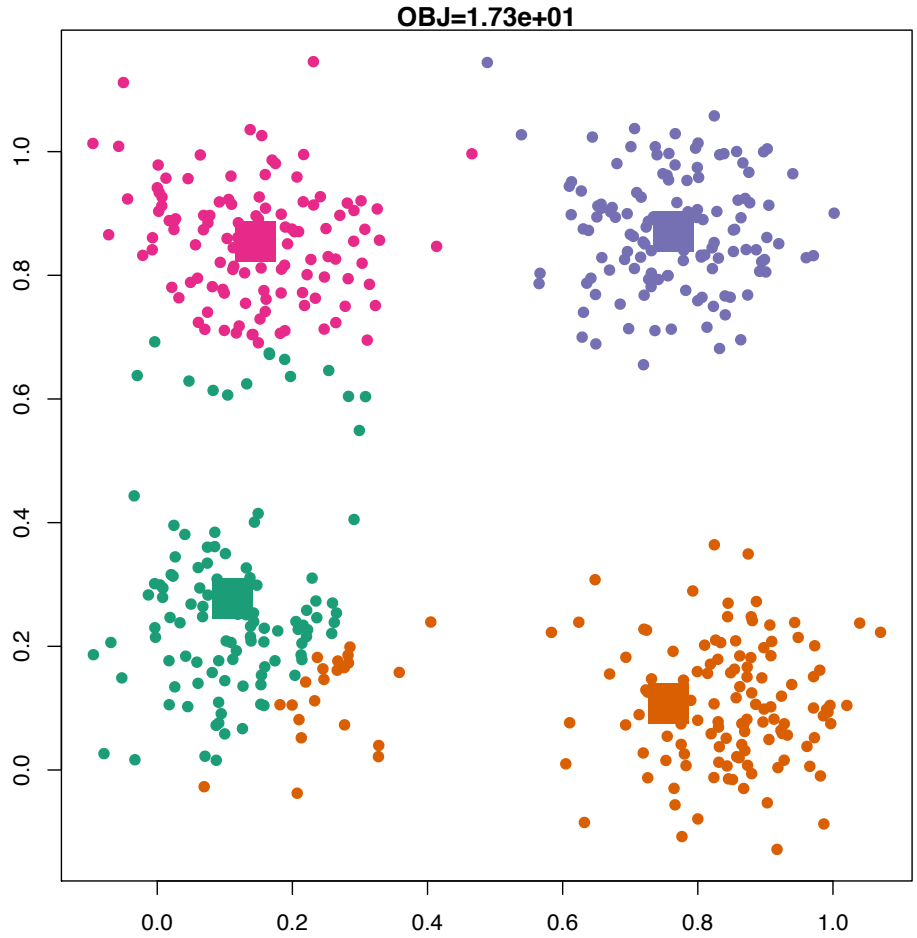


Figure: Step 4

Another Example

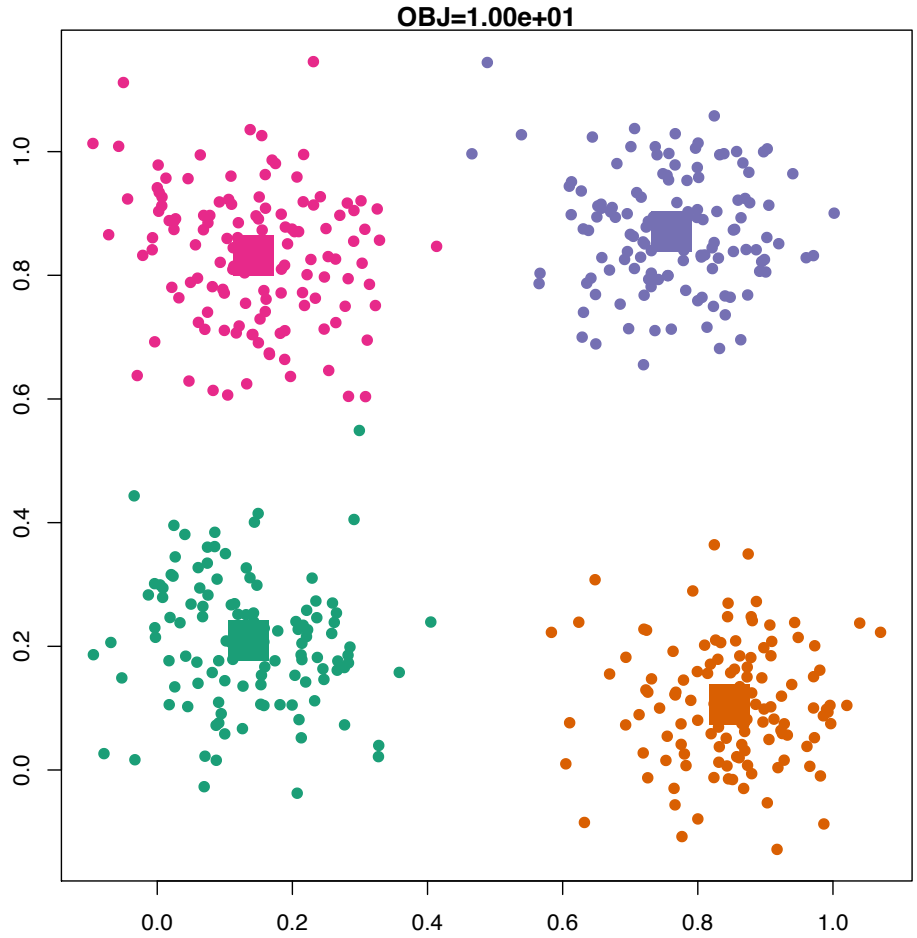


Figure: Step 5

Another Example

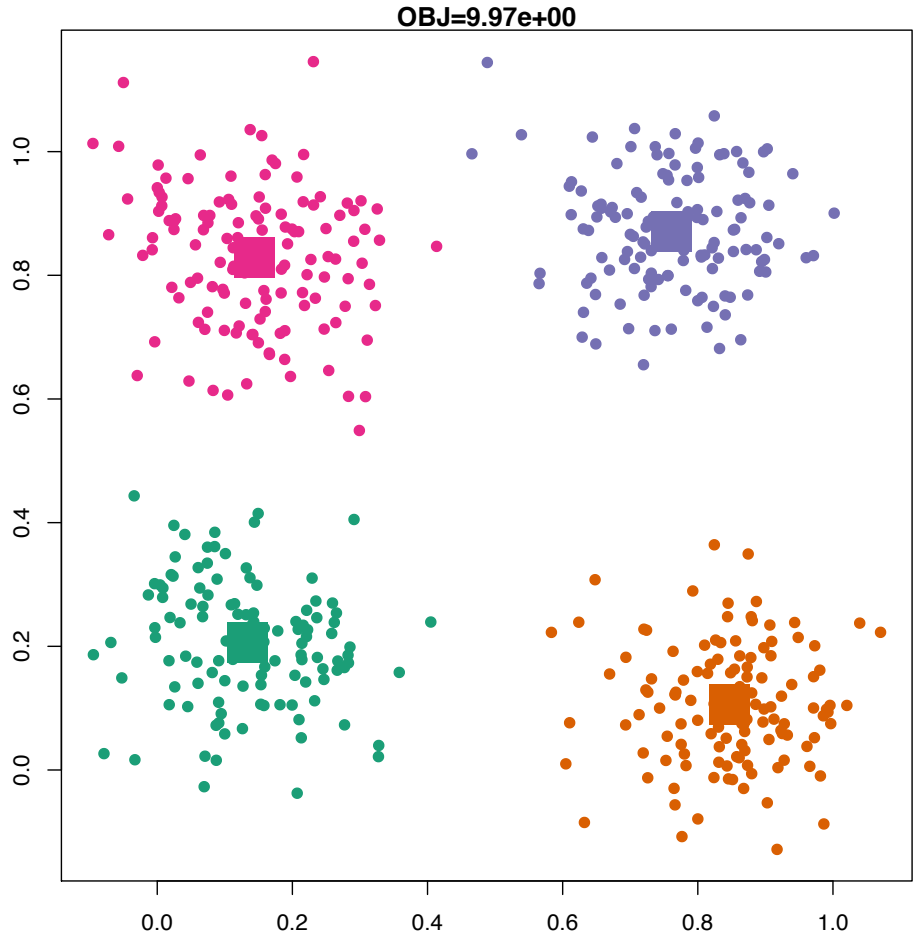


Figure: Step 6

Another Example

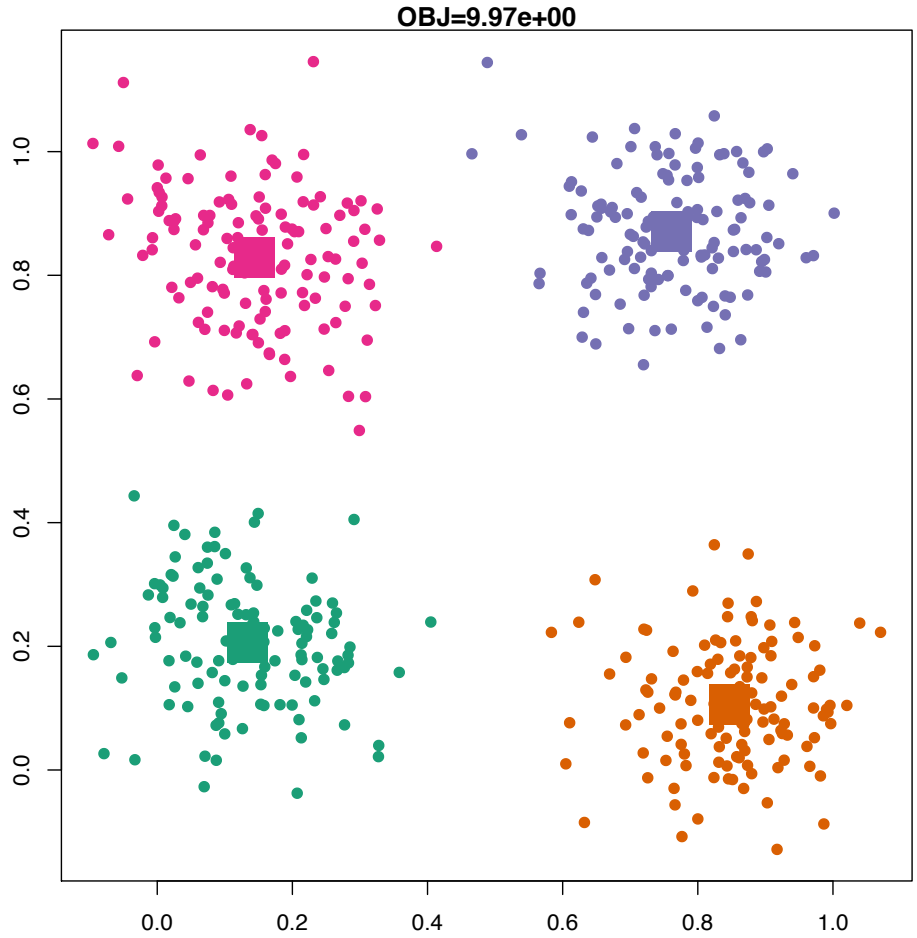


Figure: Step 7

Properties of K-means algorithm

- Guaranteed to converge in a finite number of iterations
- Running time per iteration:
  1. Assign data points to closest cluster center:
     - **O(KN) time**
  2. Change the cluster center to the average of its assigned points:
     - **O(N) time**

K-Means Convergence

Objective: $\min_\mu \min_C \sum_{i=1}^{k} \sum_{x \in C_i} \|x - \mu_i\|^2$

1. Fix μ, optimize C:

$$\min_C \sum_{i=1}^{k} \sum_{x \in C_i} \|x - \mu_i\|^2 = \min_C \sum_{i}^{n} \|x_i - \mu_{x_i}\|^2$$

2. Fix C, optimize μ:

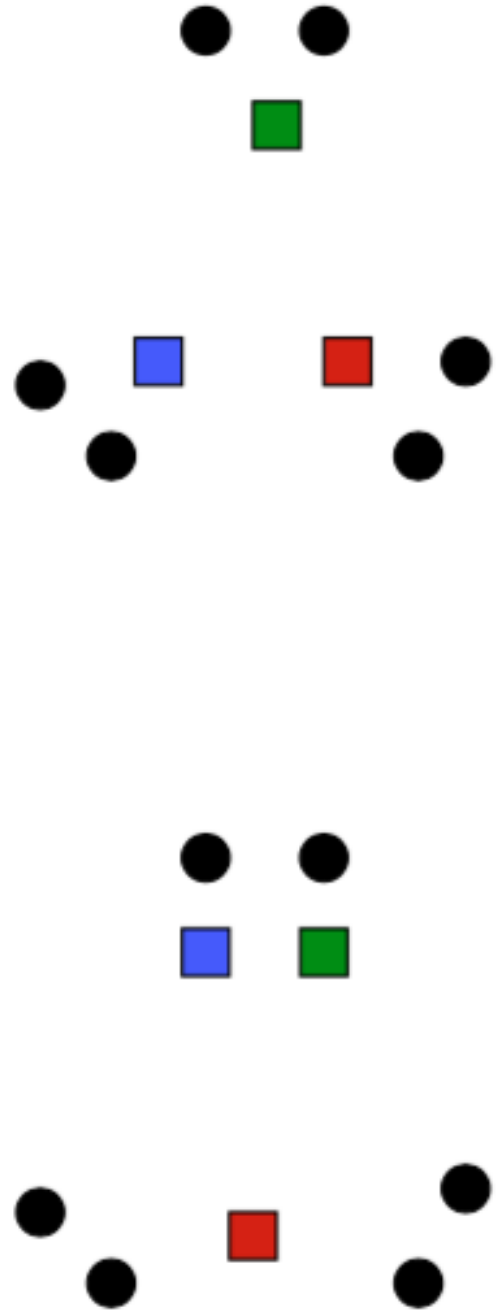
$$\min_\mu \sum_{i=1}^{k} \sum_{x \in C_i} \|x - \mu_i\|^2$$

- Take partial derivative of μ_i and set to zero, we have

$$\mu_i = \frac{1}{\|C_i\|} \sum_{x \in C_i} x$$

Kmeans takes an alternating optimization approach. Each step is guaranteed to decrease the objective—thus guaranteed to converge.

Initialization

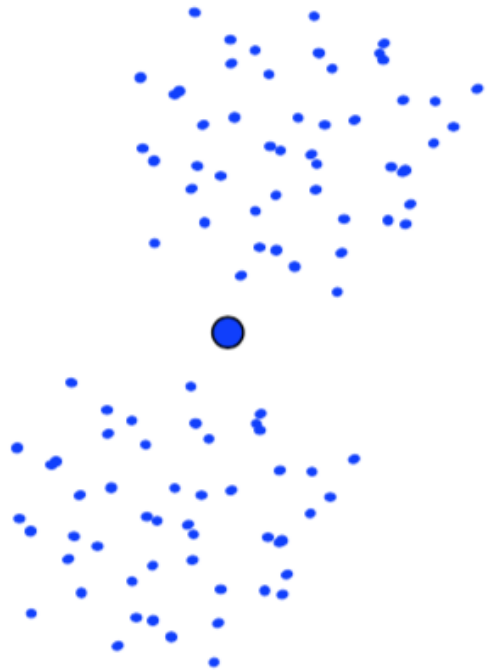


K-means algorithm is a heuristic:

K-Means Getting Stuck

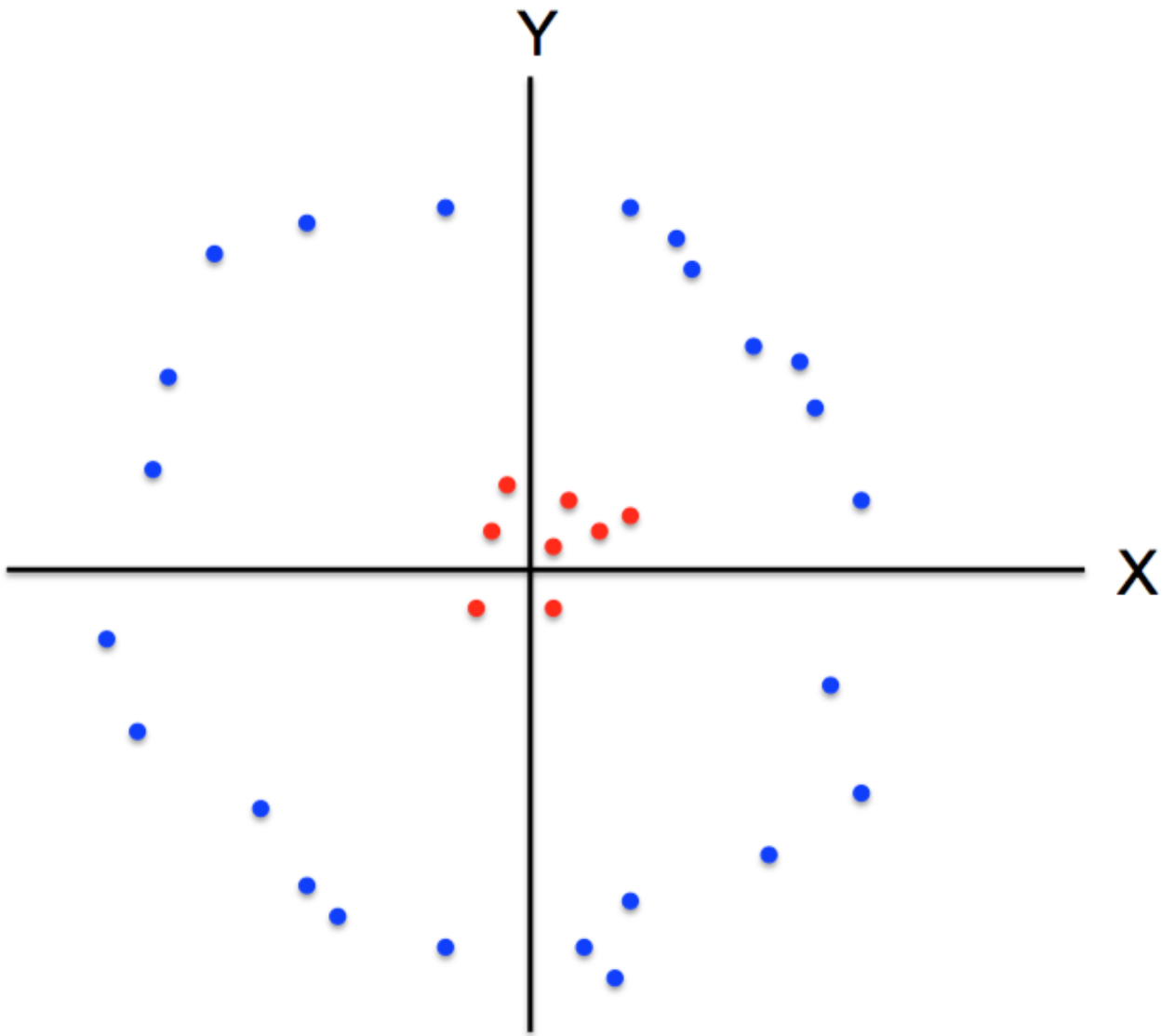
Local optima dependent on how the problem was specified:

Would be better to have
one cluster here

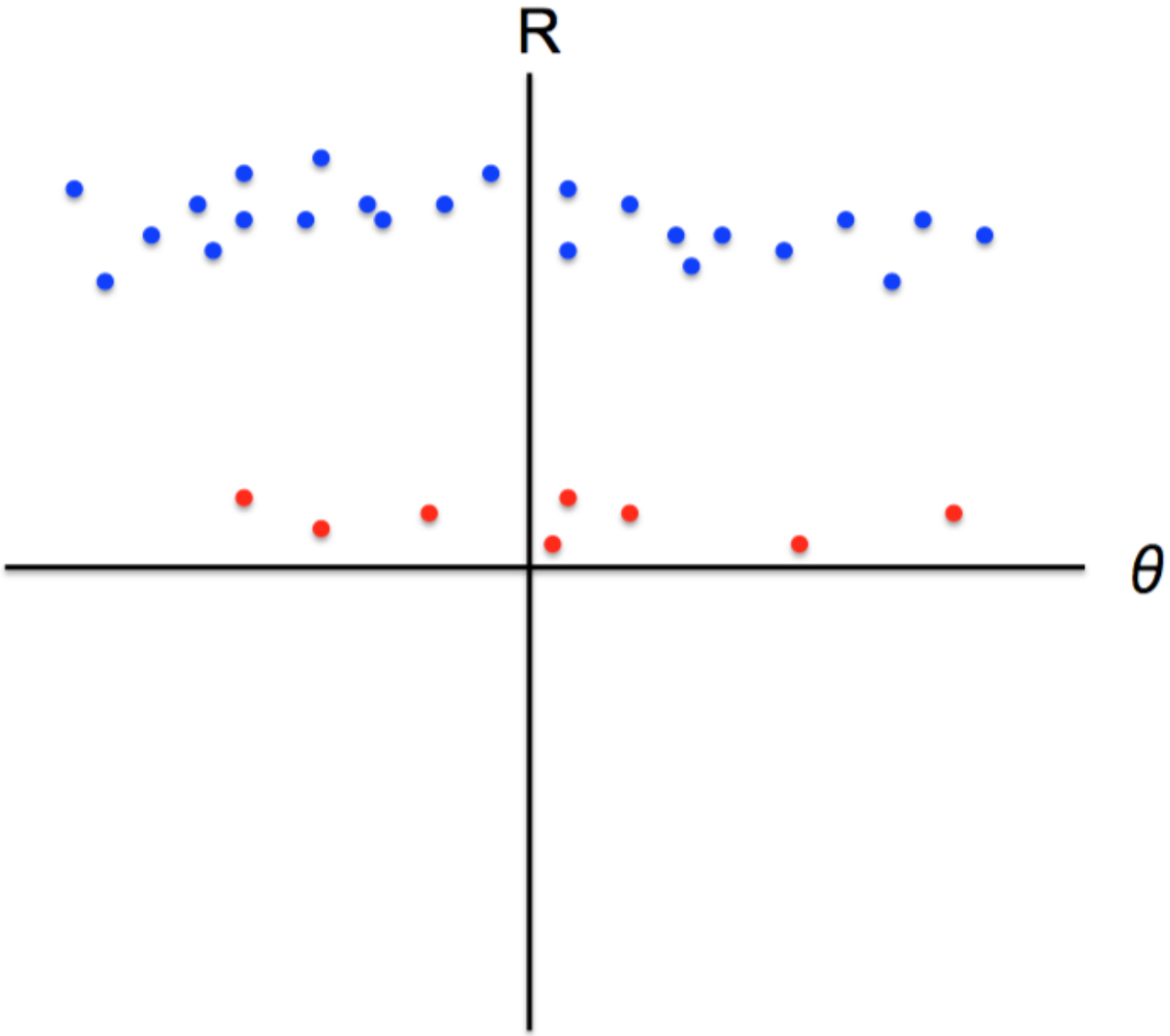


… and two clusters here

K-means not able to properly cluster

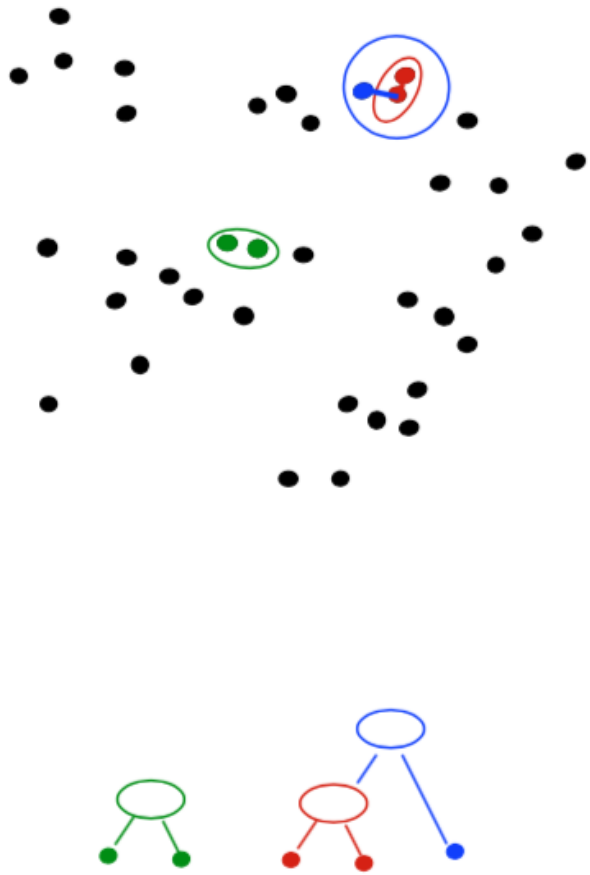


Changing the features (distance function) can help



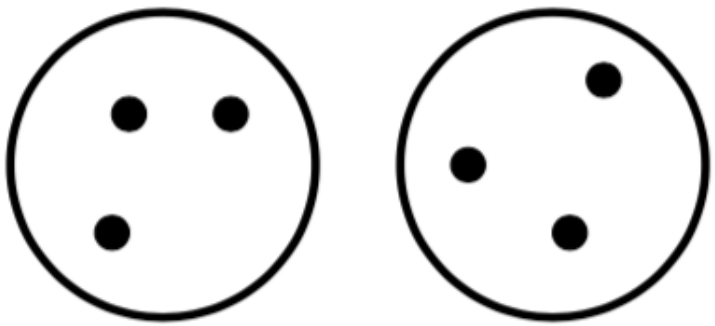
Agglomerative Clustering

- Agglomerative clustering:
  - First merge very similar instances
  - Incrementally build larger clusters out of smaller clusters
- Algorithm:
  - Maintain a set of clusters
  - Initially, each instance in its own cluster
  - Repeat:
    - Pick the two closest clusters
    - Merge them into a new cluster
    - Stop when there's only one cluster left
- Produces not one clustering, but a family of clusterings represented by a dendrogram



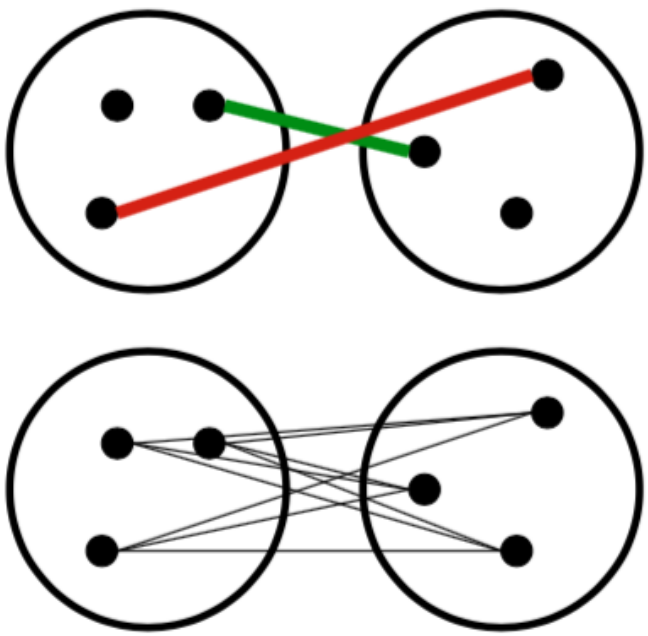
Agglomerative Clustering

How should we define "closest" for clusters with multiple elements?



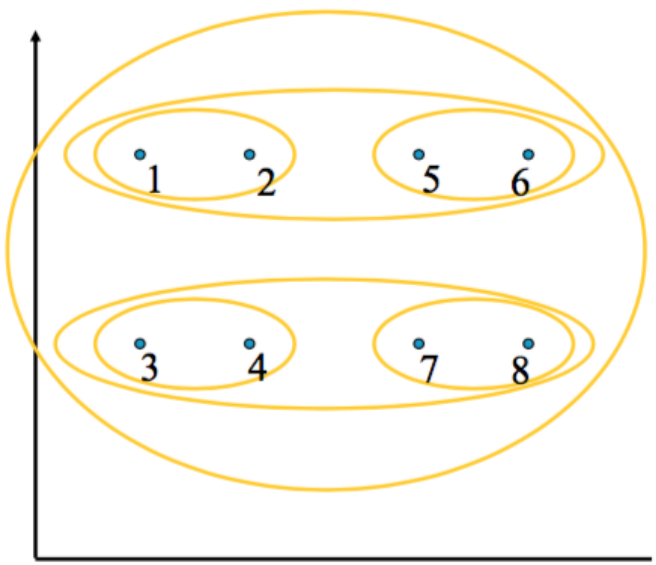
Agglomerative Clustering

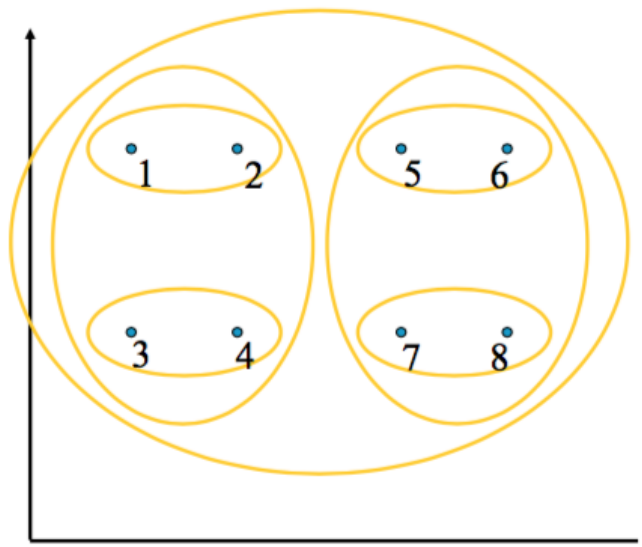
- How should we define "closest" for clusters with multiple elements?
- Many options:
  - Closest pair (single-link clustering)
  - Farthest pair (complete-link clustering)
  - Average of all pairs
- Different choices create different clustering behaviors



Agglomerative Clustering

- How should we define "closest" for clusters with multiple elements?
- Many options:
  - Closest pair (left)
  - Farthest pair (right)
  - Average of all pairs
- Different choices create different clustering behaviors





Clustering Behavior

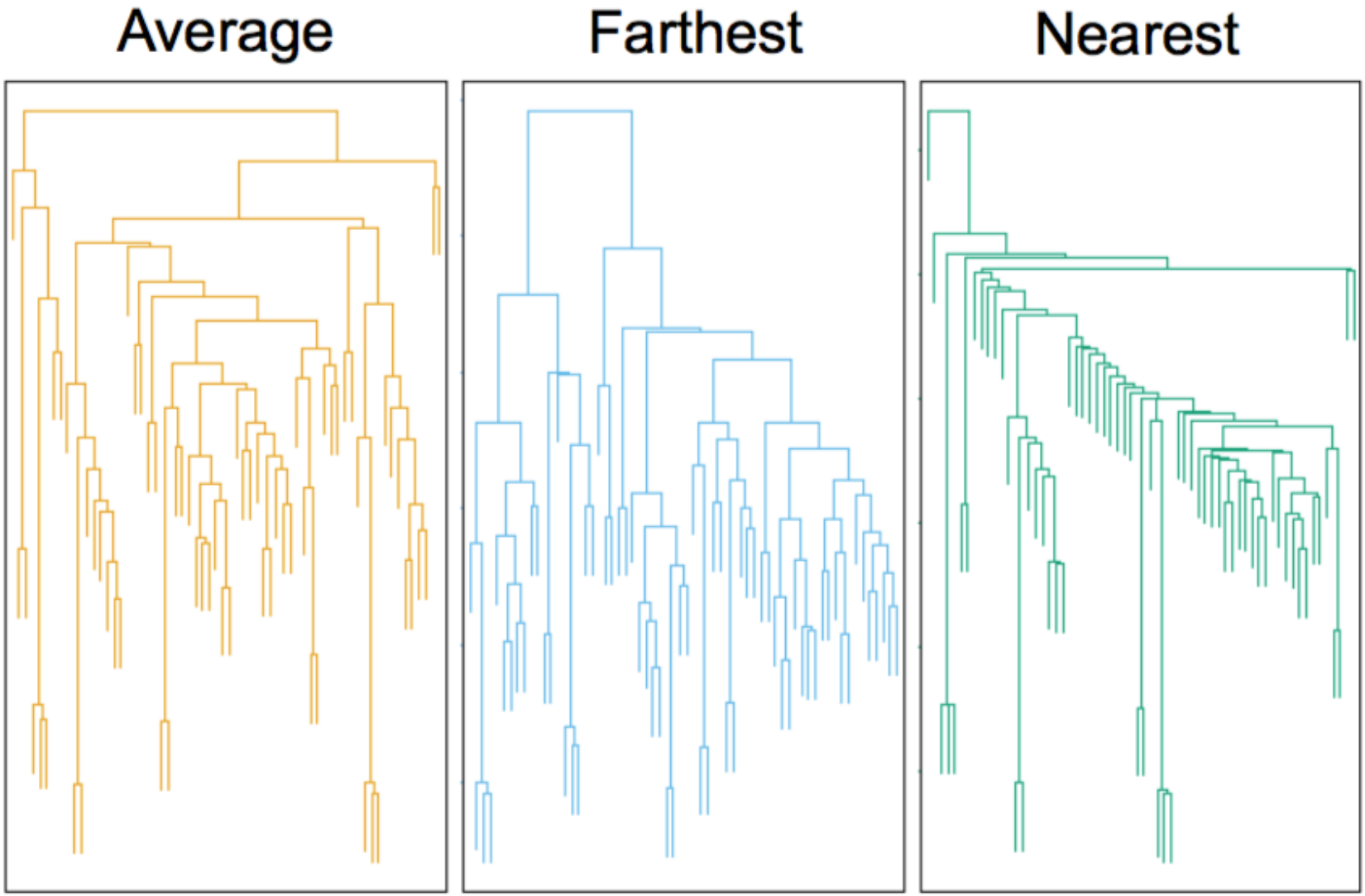


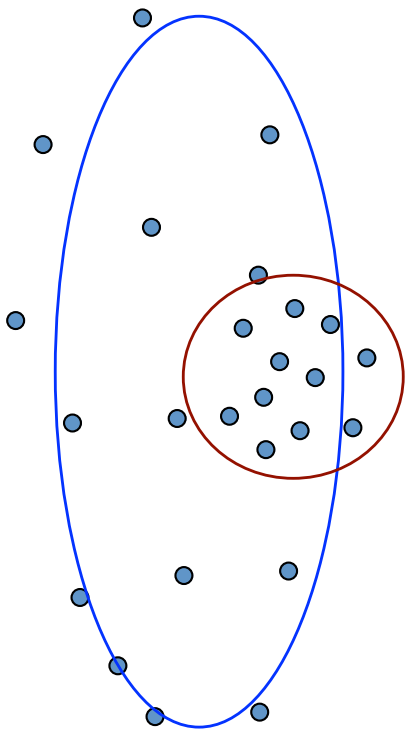
Figure: Mouse tumor data from Hastie et al.

Agglomerative Clustering Questions

- Will agglomerative clustering converge?
  - To a global optimum?
- Will it always find the true patterns in the data?
- Do people ever use it?
- How many clusters to pick?

Reconsidering “hard assignments”?

- Clusters may overlap
- Some clusters may be “wider” than others
- Distances can be deceiving!



Applications

- Clustering patients into groups based on molecular or eitiological measurements
- Cells into groups based on molecular measurements
- Neuronal signals

ARTICLE

doi:10.1038/nature10983

The genomic and transcriptomic architecture of 2,000 breast tumours reveals novel subgroups

Christina Curtis[1,2]†*, Sohrab P. Shah[3,4]*, Suet-Feung Chin[1,2]*, Gulisa Turashvili[3,4]*, Oscar M. Rueda[1,2], Mark J. Dunning[2], Doug Speed[2,5]†, Andy G. Lynch[1,2], Shamith Samarajiwa[1,2], Yinyin Yuan[1,2], Stefan Gräf[1,2], Gavin Ha[3], Gholamreza Haffari[3], Ali Bashashati[3], Roslin Russell[2], Steven McKinney[3,4], METABRIC Group‡, Anita Langerød[6], Andrew Green[7], Elena Provenzano[8], Gordon Wishart[8], Sarah Pinder[9], Peter Watson[3,4,10], Florian Markowetz[1,2], Leigh Murphy[10], Ian Ellis[7], Arnie Purushotham[9,11], Anne-Lise Børresen-Dale[6,12], James D. Brenton[2,13], Simon Tavaré[1,2,5,14], Carlos Caldas[1,2,8,13] & Samuel Aparicio[3,4]

The elucidation of breast cancer subgroups and their molecular drivers requires integrated views of the genome and transcriptome from representative numbers of patients. We present an integrated analysis of copy number and gene expression in a discovery and validation set of 997 and 995 primary breast tumours, respectively, with long-term clinical follow-up. Inherited variants (copy number variants and single nucleotide polymorphisms) and acquired somatic copy number aberrations (CNAs) were associated with expression in ~40% of genes, with the landscape dominated by *cis*- and *trans*-acting CNAs. By delineating expression outlier genes driven in *cis* by CNAs, we identified putative cancer genes, including deletions in *PPP2R2A*, *MTAP* and *MAP2K4*. Unsupervised analysis of paired DNA–RNA profiles revealed novel subgroups with distinct clinical outcomes, which reproduced in the validation cohort. These include a high-risk, oestrogen-receptor-positive 11q13/14 *cis*-acting subgroup and a favourable prognosis subgroup devoid of CNAs. *Trans*-acting aberration hotspots were found to modulate subgroup-specific gene networks, including a TCR deletion-mediated adaptive immune response in the 'CNA-devoid' subgroup and a basal-specific chromosome 5 deletion-associated mitotic network. Our results provide a novel molecular stratification of the breast cancer population, derived from the impact of somatic CNAs on the transcriptome.

Clustering Patients

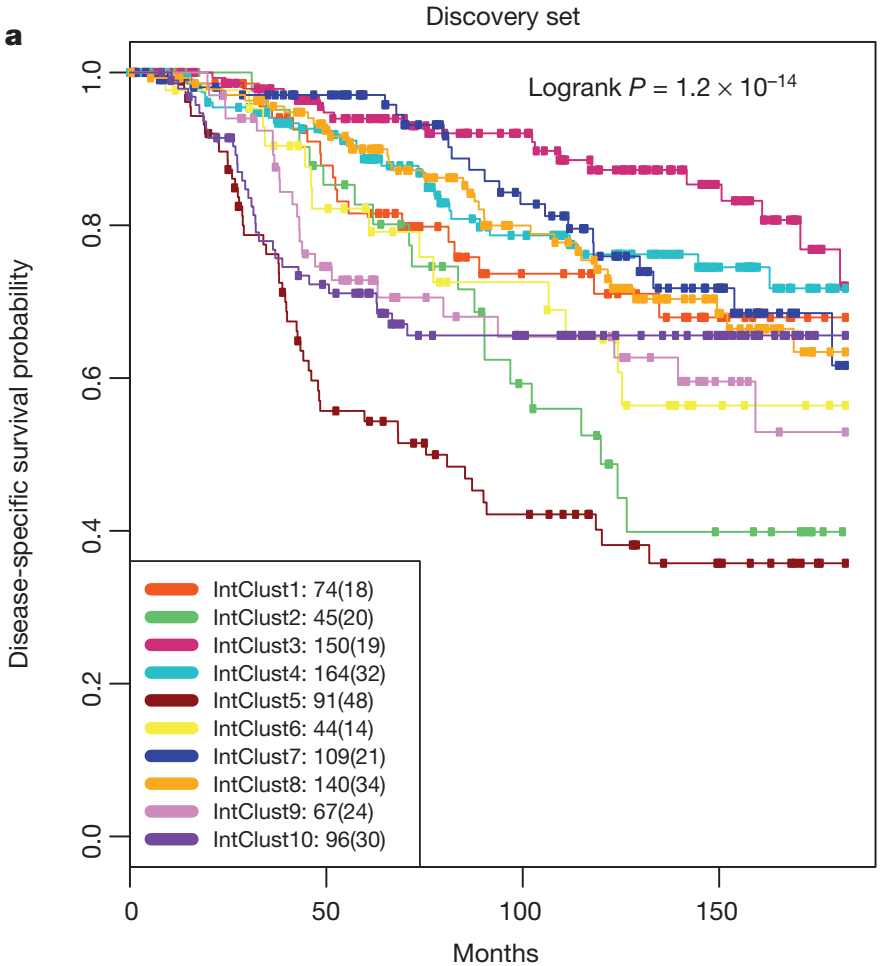


Figure 5 | The integrative subgroups have distinct clinical outcomes. **a**, Kaplan–Meier plot of disease-specific survival (truncated at 15 years) for the integrative subgroups in the discovery cohort. For each cluster, the number of samples at risk is indicated as well as the total number of deaths (in parentheses).

OPEN ACCESS Freely available online

PLoS COMPUTATIONAL BIOLOGY

MCAM: Multiple Clustering Analysis Methodology for Deriving Hypotheses and Insights from High-Throughput Proteomic Datasets

Kristen M. Naegle[1,2], Roy E. Welsch[3], Michael B. Yaffe[1,2,4], Forest M. White[1,2], Douglas A. Lauffenburger[1]*

1 Department of Biological Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts, United States of America, **2** Koch Institute for Integrative Cancer Research, Massachusetts Institute of Technology, Cambridge, Massachusetts, United States of America, **3** Sloan School of Management, Massachusetts Institute of Technology, Cambridge, Massachusetts, United States of America, **4** Department of Biology, Massachusetts Institute of Technology, Cambridge, Massachusetts, United States of America

Abstract

Advances in proteomic technologies continue to substantially accelerate capability for generating experimental data on protein levels, states, and activities in biological samples. For example, studies on receptor tyrosine kinase signaling networks can now capture the phosphorylation state of hundreds to thousands of proteins across multiple conditions. However, little is known about the function of many of these protein modifications, or the enzymes responsible for modifying them. To address this challenge, we have developed an approach that enhances the power of clustering techniques to infer functional and regulatory meaning of protein states in cell signaling networks. We have created a new computational framework for applying clustering to biological data in order to overcome the typical dependence on specific *a priori* assumptions and expert knowledge concerning the technical aspects of clustering. Multiple clustering analysis methodology ('MCAM') employs an array of diverse data transformations, distance metrics, set sizes, and clustering algorithms, in a combinatorial fashion, to create a suite of clustering sets. These sets are then evaluated based on their ability to produce biological insights through statistical enrichment of metadata relating to knowledge concerning protein functions, kinase substrates, and sequence motifs. We applied MCAM to a set of dynamic phosphorylation measurements of the ERRB network to explore the relationships between algorithmic parameters and the biological meaning that could be inferred and report on interesting biological predictions. Further, we applied MCAM to multiple phosphoproteomic datasets for the ERBB network, which allowed us to compare independent and incomplete overlapping measurements of phosphorylation sites in the network. We report specific and global differences of the ERBB network stimulated with different ligands and with changes in HER2 expression. Overall, we offer MCAM as a broadly-applicable approach for analysis of proteomic data which may help increase the current understanding of molecular networks in a variety of biological problems.

Clustering molecular signals

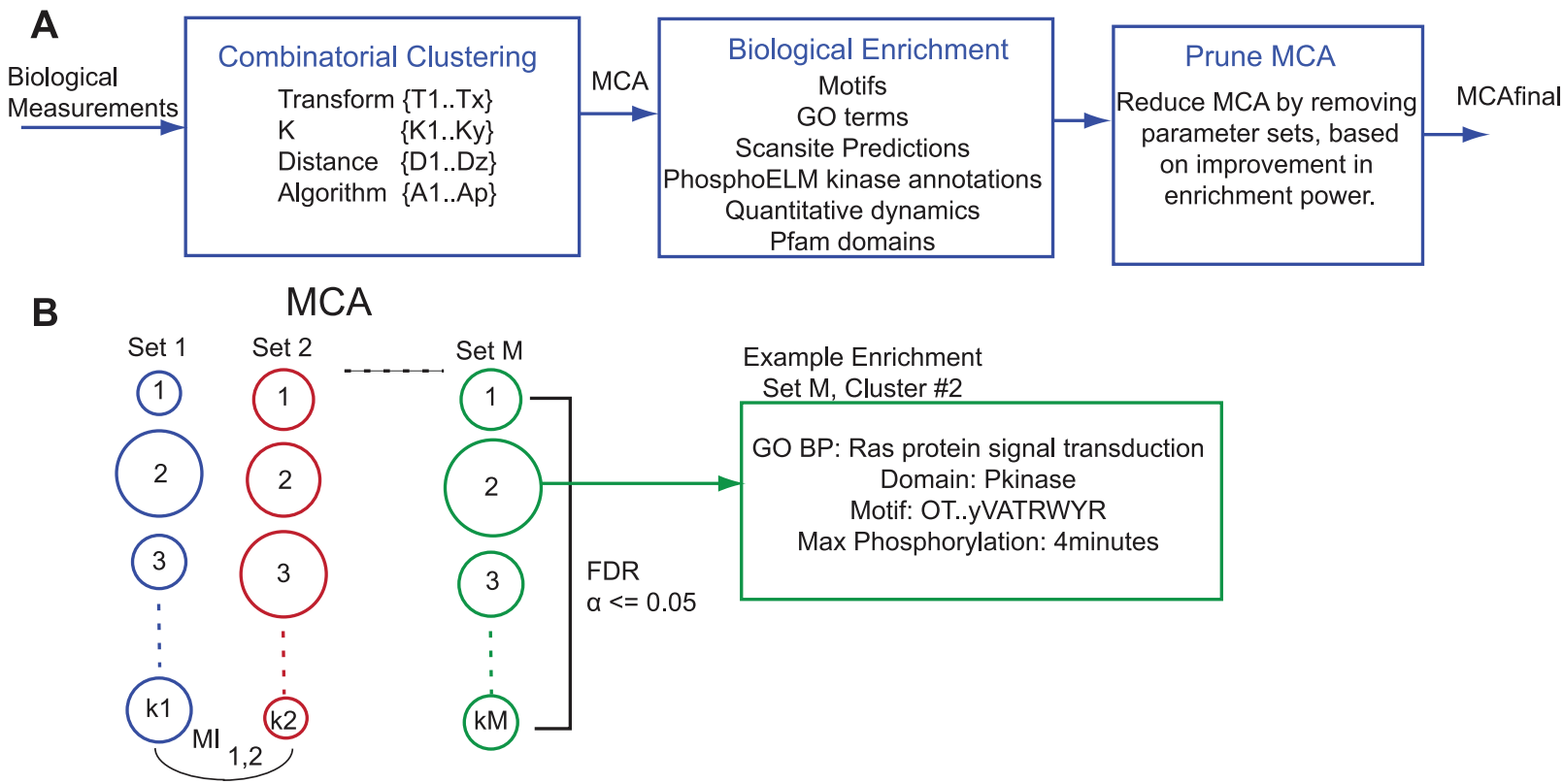


Figure 1. Multiple Clustering Analysis Method. A) MCAM begins with clustering a biological dataset through the combinatorial application of a set of clustering parameters, followed by biological enrichment testing in various categories of information. Following this, the enrichment is used to prune those parameters that contribute little biological information. B) The depiction of an MCA, which contains M sets, with each set having some number of k clusters and produced by a particular combination of clustering parameters. Biological enrichment is corrected for multiple hypothesis testing by using the False Discovery Rate procedure across a set and within a category of biological information. Mutual Information can be used to compare the resulting clustering solution between any two sets.
doi:10.1371/journal.pcbi.1002119.g001

Clustering molecular signals

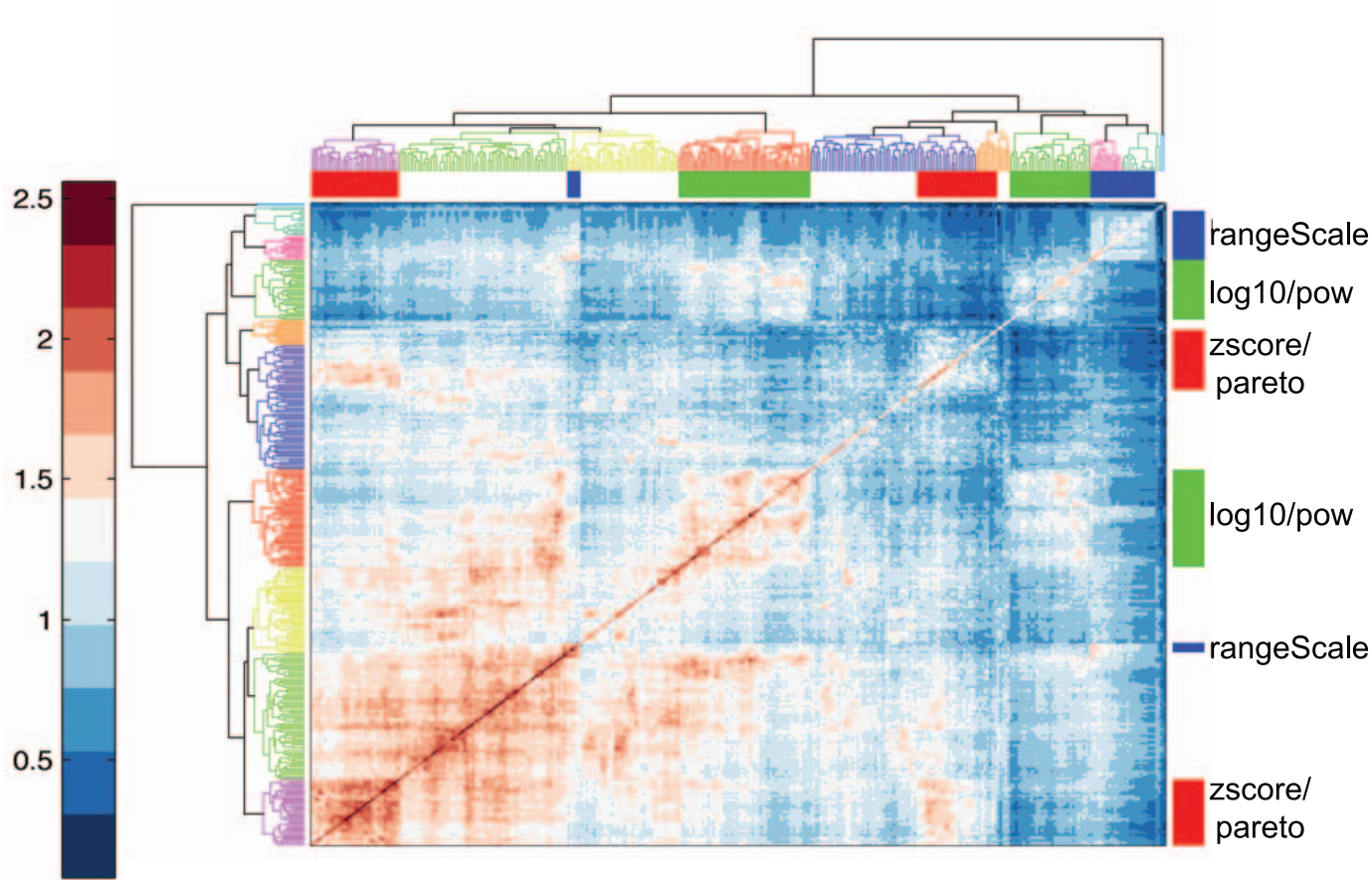


Figure 3. Comparison of parameters and metrics. B) Hierarchical clustering of pairwise mutual information between every set in the MCA. Self-MI is highest along the diagonal. Highlighted groups indicate dendrogram cutoffs for which the full group is composed of the denoted parameter. The labels log10/pow denote normMax_log10, log10 and the pow transformations, pareto/zscore contain zscore and pareto transformations. The topmost zscore/pareto group contains one outlier (out of the group of 41) created using the transform pow.
doi:10.1371/journal.pcbi.1002119.g003

Clustering molecular signals

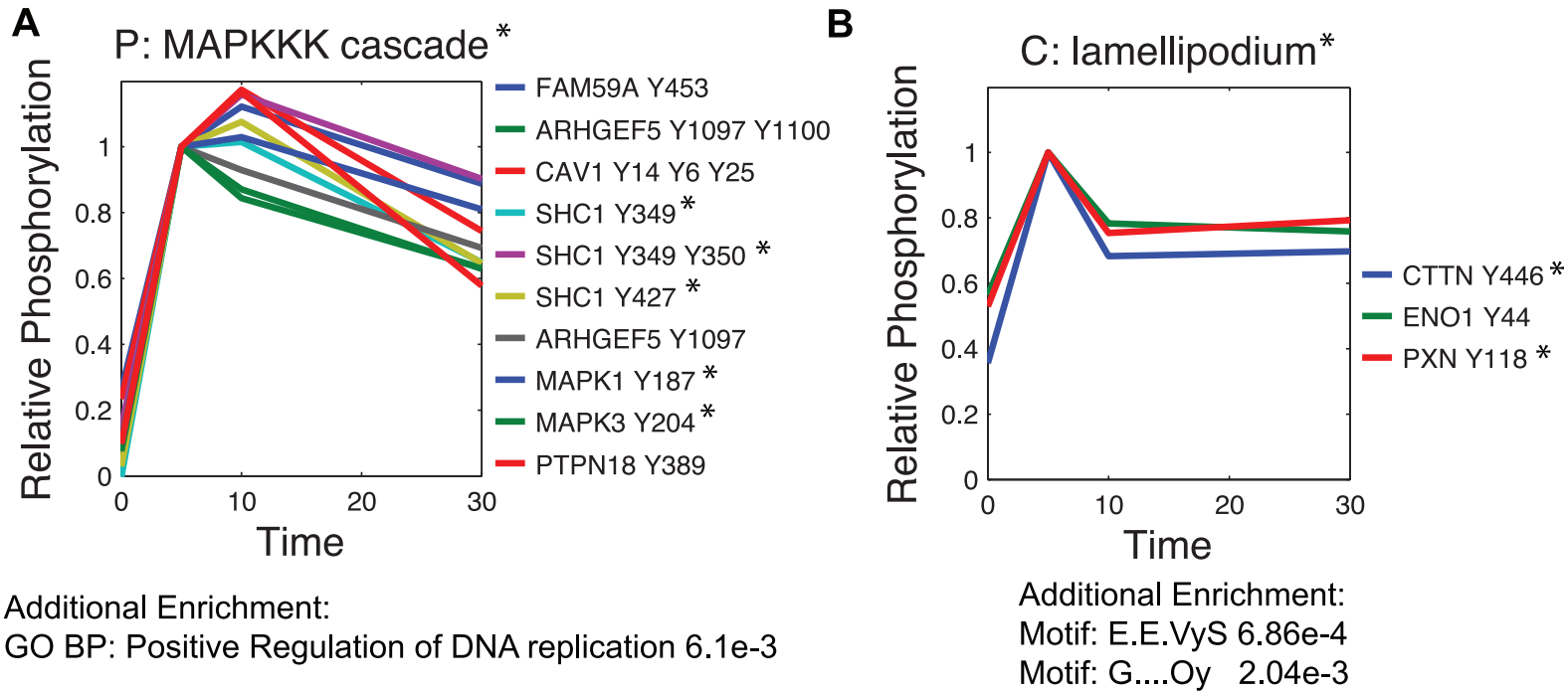


Figure 4. Biological inference based on robust clustering results. A) The group of phosphopeptides that participate at least 50% of the time in a cluster with enrichment for GO Biological Process term "MAPKKK Cascade", those proteins with the term are starred. This new group is enriched for GO BP term "positive regulation of DNA replication". B) These three phosphopeptides always appear when GO Cellular Compartment term "lamellipodium" is enriched, CTTN and PXN are the proteins annotated as being localized in lamellipodium. This new group is enriched for two sequence motifs as well.

REVIEW

COMPUTATIONAL BIOLOGY

Avoiding common pitfalls when clustering biological data

Tom Ronan, Zhijie Qi, Kristen M. Naegle*

Clustering is an unsupervised learning method, which groups data points based on similarity, and is used to reveal the underlying structure of data. This computational approach is essential to understanding and visualizing the complex data that are acquired in high-throughput multidimensional biological experiments. Clustering enables researchers to make biological inferences for further experiments. Although a powerful technique, inappropriate application can lead biological researchers to waste resources and time in experimental follow-up. We review common pitfalls identified from the published molecular biology literature and present methods to avoid them. Commonly encountered pitfalls relate to the high-dimensional nature of biological data from high-throughput experiments, the failure to consider more than one clustering method for a given problem, and the difficulty in determining whether clustering has produced meaningful results. We present concrete examples of problems and solutions (clustering results) in the form of toy problems and real biological data for these issues. We also discuss ensemble clustering as an easy-to-implement method that enables the exploration of multiple clustering solutions and improves robustness of clustering solutions. Increased awareness of common clustering pitfalls will help researchers avoid overinterpreting or misinterpreting the results and missing valuable insights when clustering biological data.

Introduction

Technological advances in recent decades have resulted in the ability to measure large numbers of molecules, typically across a smaller number of

Differentiating between a meaningful and a random clustering result can be accomplished by applying cluster validation methods, determining statistical and biological significance, accounting for noise, and evaluating

Review

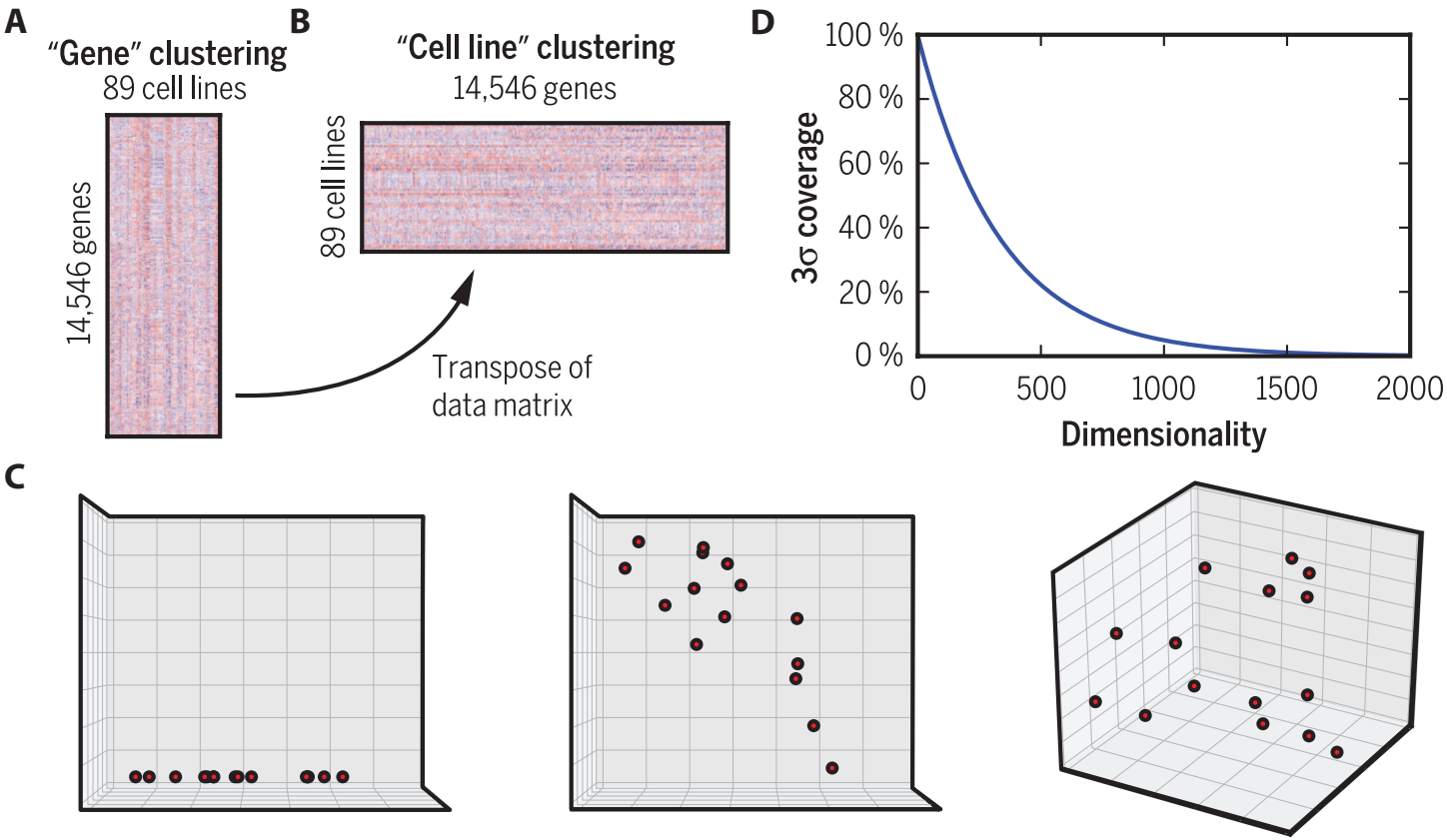


Fig. 1. Determining the dimensionality of a clustering problem. (**A** and **B**) Representation of the mRNA clustering problem consisting of >14,000 mRNAs measured across 89 cell lines. Data are from Lu *et al.* (*6*). When the mRNAs are clustered, the mRNAs are the objects and each cell line represents a feature, resulting in an 89-dimensional problem (A). When attempting to classify normal and tumor cell lines using gene expression, the cells lines are the objects and each mRNA is a feature, resulting in a clustering problem with thousands of dimensions (B). (**C**) Effect of dimensionality on sparsity. (**D**) Effect of dimensionality on coverage of the data based on SD from the mean.

Review

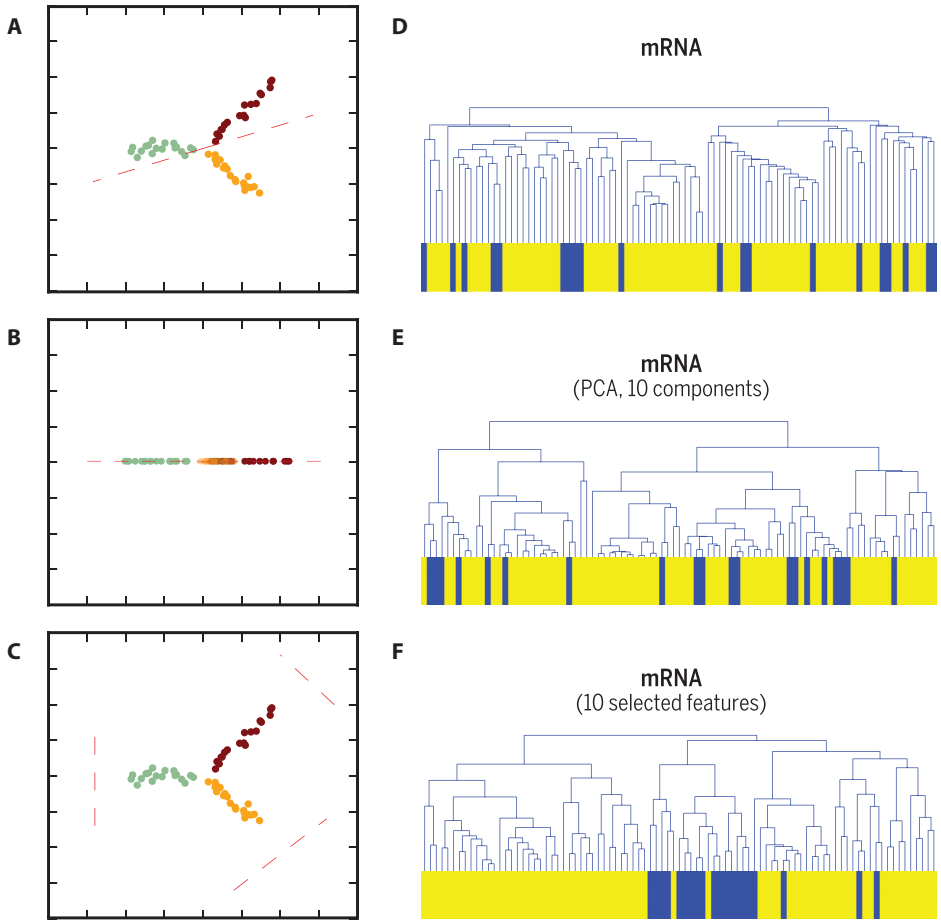


Fig. 2. Dimensionality reduction methods and effects. Comparison of PCA and subspace clustering. (**A**) Three clusters are plotted in two dimensions. The dashed red line indicates the one-dimensional line upon which the original two-dimensional data is projected as determined by PCA. (**B**) The clusters are plotted in the new single dimension after reducing the dimensionality from two to one. (**C**) Three alternate one-dimensional projections (dashed red lines) onto which the data can be projected, each demonstrating better separability for some clusters than the projection identified using PCA. (**D** to **F**) Comparison of the original clustering results of 89 cell lines in ~14,000-dimensional mRNA data (D) to clustering results after PCA (E) and after subspace clustering (F). Blue bars, gastrointestinal cell lines; yellow bars, nongastrointestinal cell lines.

Review

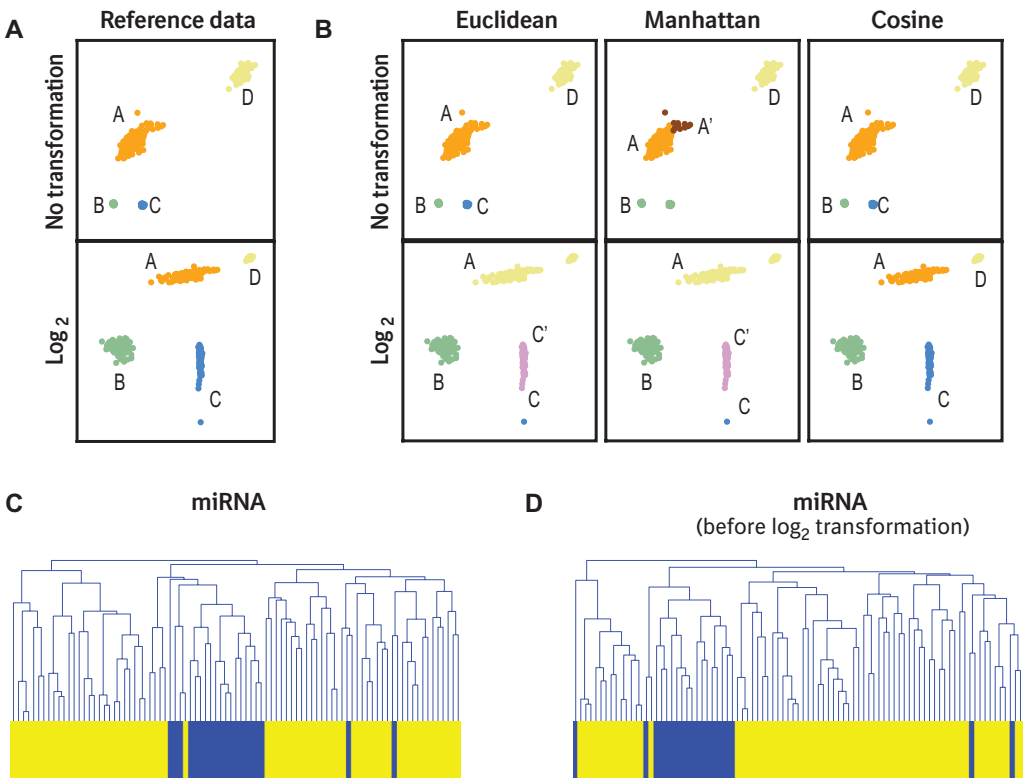


Fig. 3. Effect of transformations and distance metric on clustering results. (**A**) Demonstration of how transformations affect the relationship of data points in space. A toy data set (reference set, https://github.com/knaegle/clusteringReview) was clustered into four clusters with agglomerative clustering, average linkage, and Euclidean distance. The four reference clusters without transformation (upper panel) and after $\log_2$ transformation (lower panel). (**B**) Transformations and distance metrics change clustering results when compared to the reference clustering result. With no transformation (upper panels), Euclidean and cosine distance do not change cluster identity, but with Manhattan distance, a new cluster A′ is added, and cluster C is merged into cluster B. With the $\log_2$ transformation (lower panels), the Euclidean and Manhattan metrics caused cluster C′ to emerge and cluster D to be lost. (**C**) Dendrogram from the microRNA (miRNA) clustering experiment result from 89 cell lines and 217 microRNAs (*6*). Gastrointestinal-derived cell lines (blue bars) predominantly cluster together in the full-dimensional space. Note: The data were $\log_2$-transformed as part of the preclustering analysis. (**D**) Same microRNA data as in (C) but without $\log_2$ transformation.

Review

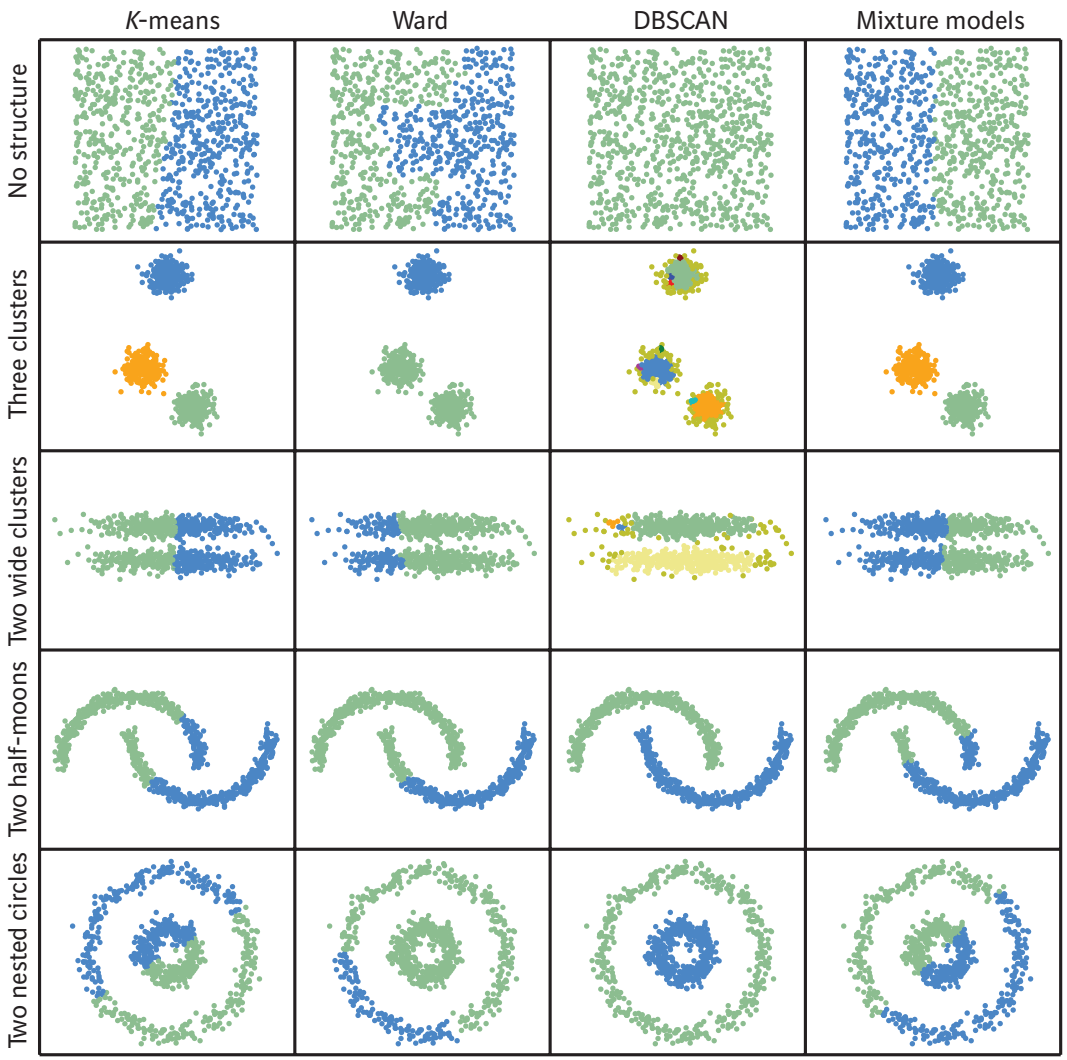


Fig. 4. The effect of algorithm on clustering results. Four toy data sets (https://github.com/knaegle/clusteringReview) demonstrate the effects of different types of clustering algorithms on various known structures in two-dimensional data.

Review

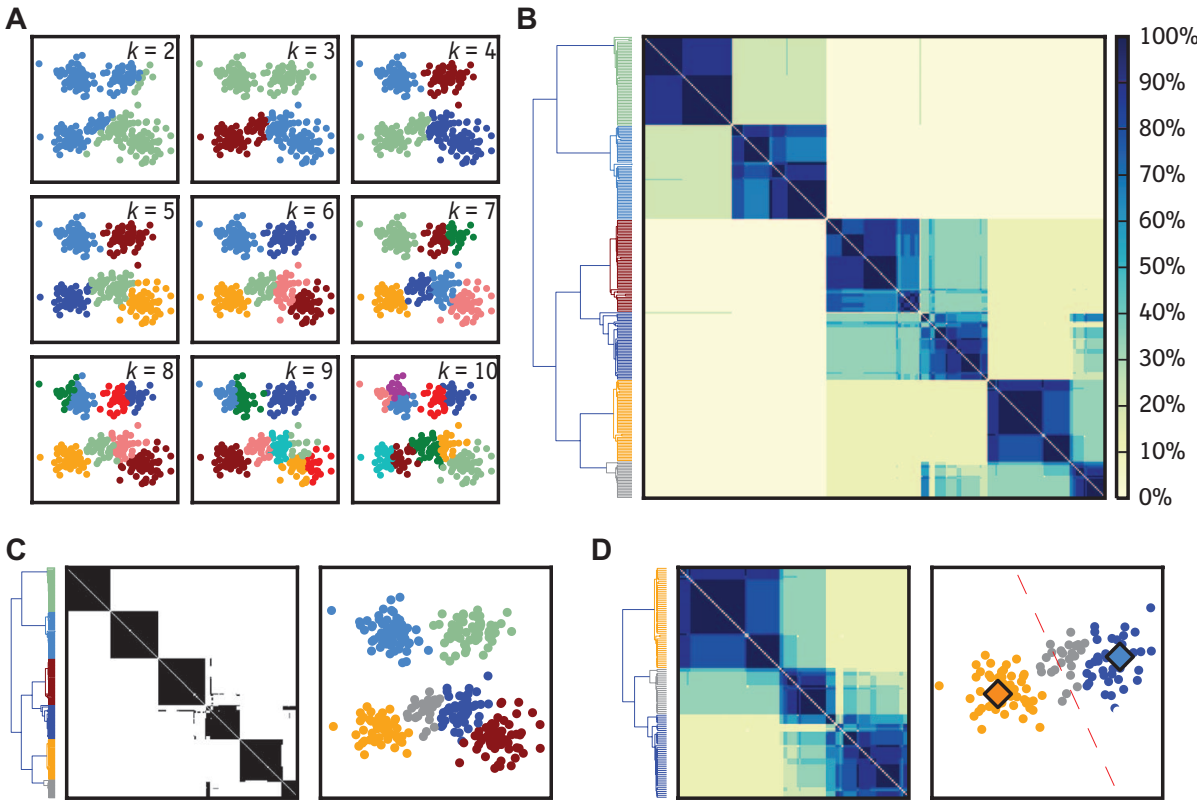


Fig. 5. Ensemble clustering overview. Finishing techniques were applied to random toy data (see file S1 for analysis details). (**A**) Set of clustering results obtained using the *k*-means algorithm with various values of *k* (a *k*-sweep). (**B**) Hierarchically clustered (Ward linkage) co-occurrence matrix for the ensemble of results in (A). The heatmap represents the percentage of times any pair of data points coclusters across the ensemble. (**C**) A majority vote analysis was applied (left panel) using a threshold of 50% on the co-occurrence matrix in (B). Six clusters (see dendrogram color groupings) result from the majority vote (right panel). (**D**) Application of fuzzy clustering to the ensemble. The left panel shows the details of the co-occurrence matrix for the blue, gray, and orange clusters, and the right shows the clustering assignments. The gray cluster provides an example of partially fuzzy clustering because it shares membership with the orange and dark blue clusters.

Review

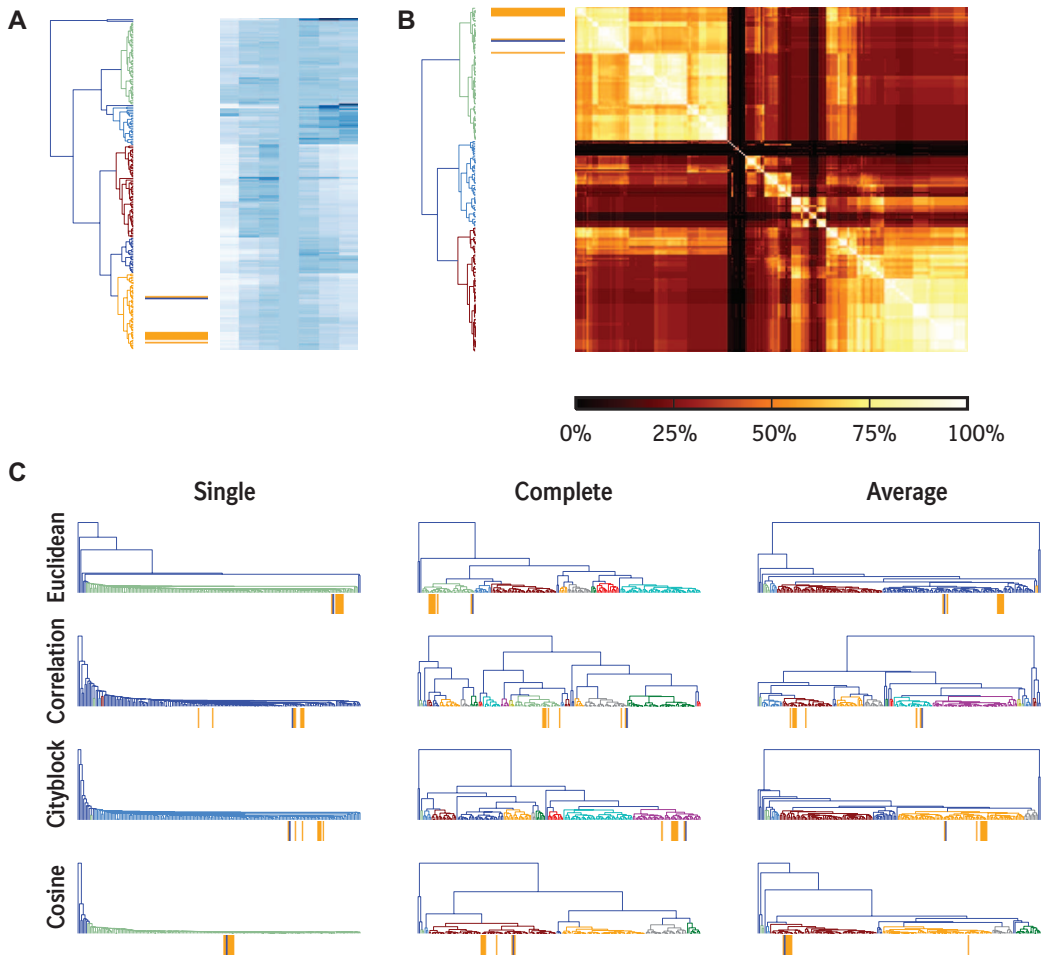


Fig. 6. Ensemble clustering on phosphoproteomic data. (A) Single clustering solution showing known interactors with EGFR (orange bars) and PDLIM1 (blue bar) coclustering in the phosphoproteomic data (blue heatmap). (B) Co-occurrence matrix heatmap demonstrating clustering of interactors with EGFR. The known interactors with EGFR (orange bars) and PDLIM1 (blue bar) are found in a single cluster (upper left). (C) Subset of clustering results across multiple distance metrics and clustering algorithms. Under the dendrogram, known interactors with EGFR are marked with orange bars and PDLIM1 is marked with a blue bar.

Implementation

K-means

`sklearn.cluster.KMeans`

- `n_clusters`: Number of clusters to form
- `init`: Initialization method
- `n_init`: Number of initializations
- `max_iter`: Maximum steps algorithm can take

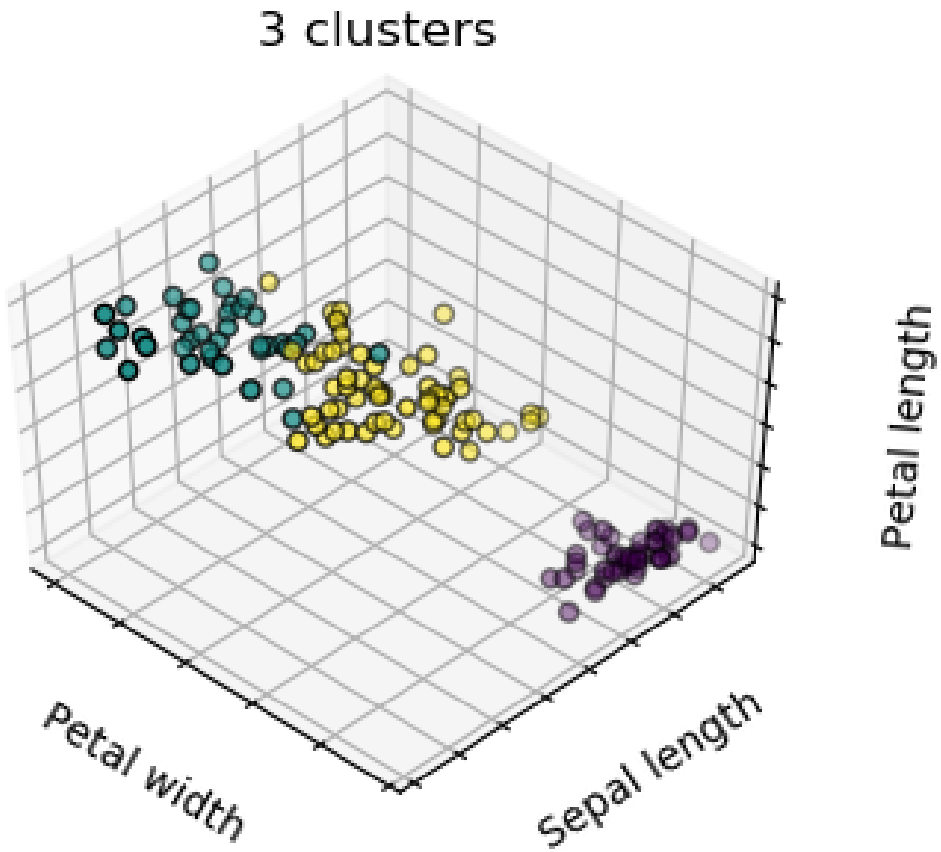
Agglomerative

`sklearn.cluster.AgglomerativeClustering`

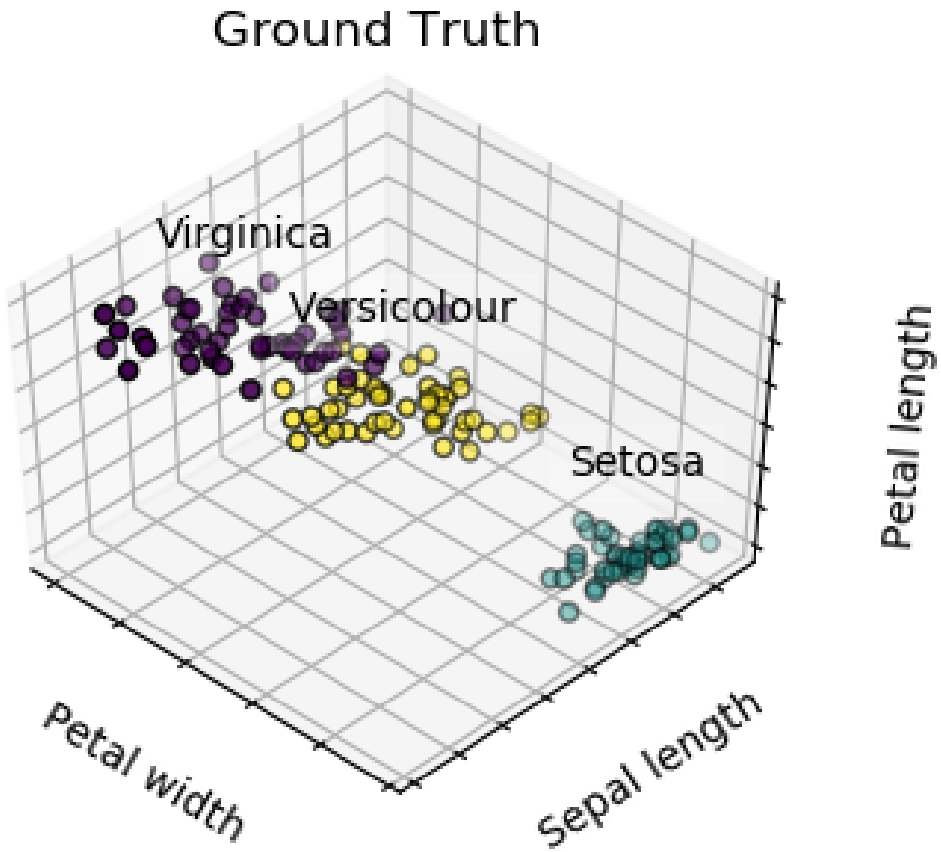
Implementation - K-means

```
import numpy as np
from matplotlib.pyplot import figure
from mpl_toolkits.mplot3d import Axes3D

from sklearn.cluster import KMeans
from sklearn import datasets

iris = datasets.load_iris()
X, y = iris.data, iris.target

est = KMeans(n_clusters=3)

ax = Axes3D(figure(), rect=[0, 0, .95, 1], elev=48, azim=134)
est.fit(X)
labels = est.labels_

ax.scatter(X[:, 3], X[:, 0], X[:, 2],
           c=labels.astype(np.float), edgecolor='k')

# ...
```

Implementation - K-means



Implementation - K-means



Further Reading

- sklearn: Clustering
- Avoiding common pitfalls when clustering biological data