FLATIRON INSTITUTE
Center for Computational Mathematics

Clustering in low dimensions

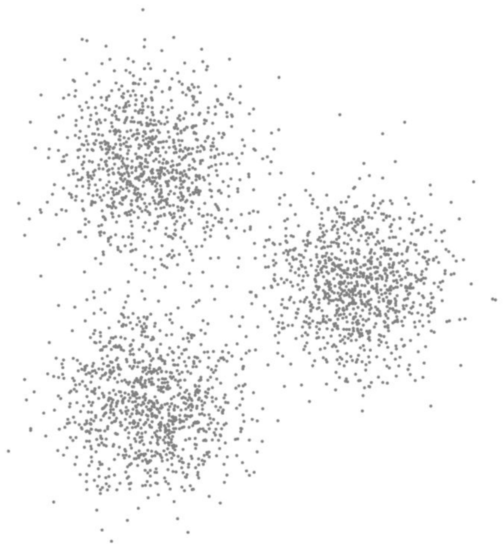
Jeremy Magland

1 November 2019
Flatiron-wide Algorithms and Mathematics (FWAM) conference

Objectives

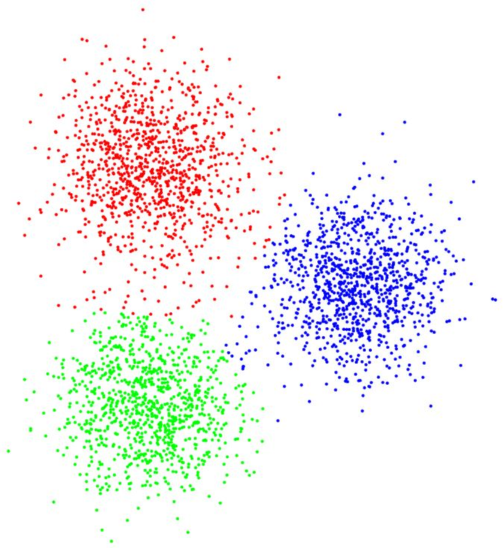
- Focus on 2-d (for visualization purposes)
- Provide overview of popular clustering algorithms
  - Algorithmic concepts
  - Example performance
  - Instructions on how to run
  - Pros / cons
- Web application for comparison and evaluation
- Motivate further research - there is room for improvement

How many clusters do you see?

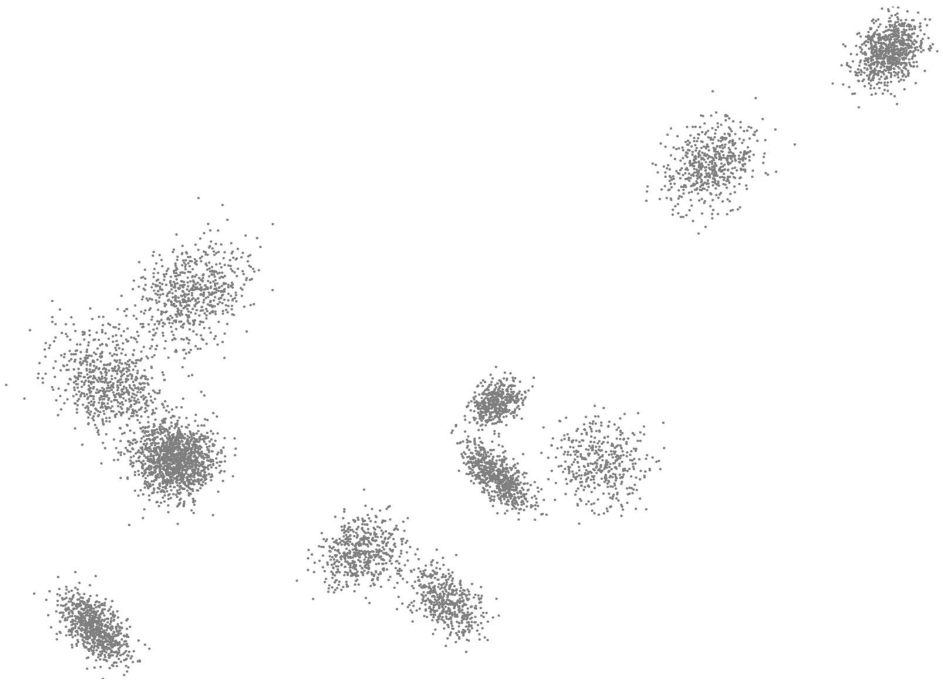


Center for Computational Mathematics

How many clusters do you see?

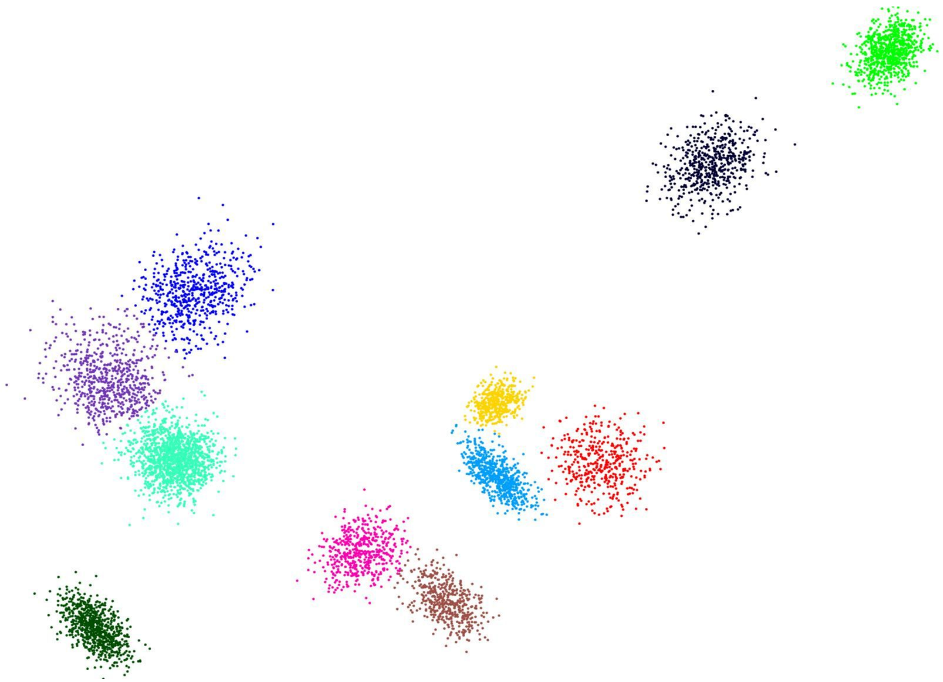


How many clusters do you see?



FLATIRON INSTITUTE
Center for Computational Mathematics

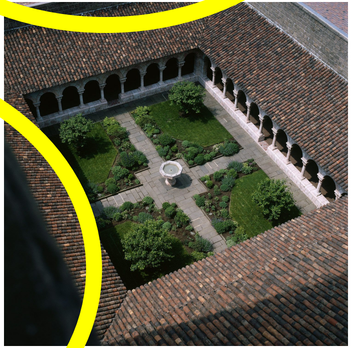
How many clusters do you see?



FLATIRON INSTITUTE
Center for Computational Mathematics

How many clusters do you see?

How many clusters do you see?



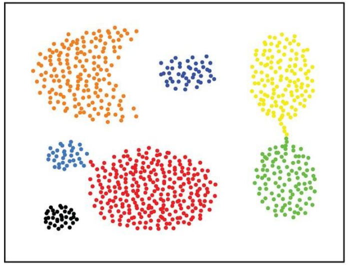
FLATIRON INSTITUTE
Center for Computational Mathematics

What is clustering?

Given points $x_1, \ldots x_n$ in R^d, determine the number of clusters K and integer labels $l_1, \ldots, l_n$, where $l_i \in \{1, \ldots, K\}$ **such that ...**

Loosely speaking: similar points should be in the same cluster.

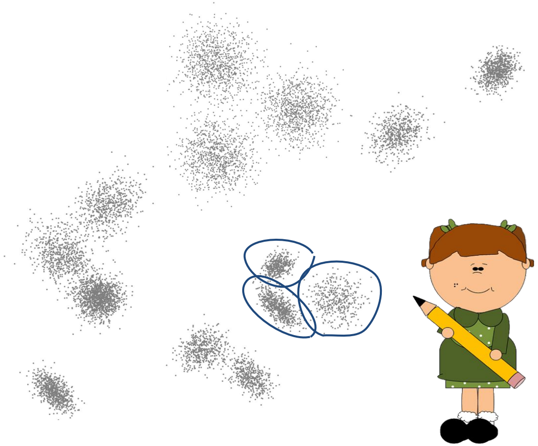
The classification should agree with the human eye in the obvious cases.



FLATIRON INSTITUTE
Center for Computational Mathematics

Spoiler

In two dimensions, I'm not sure if any existing clustering techniques can do as well as an eight-year-old human.



Good news: there is room for improvement

K-means

Step 1: Optimize the K cluster representatives.

Step 2: Optimize assignment to the K clusters.

Repeat until convergence

In practice, convergence is rapid, even though it is technically NP-hard in general.

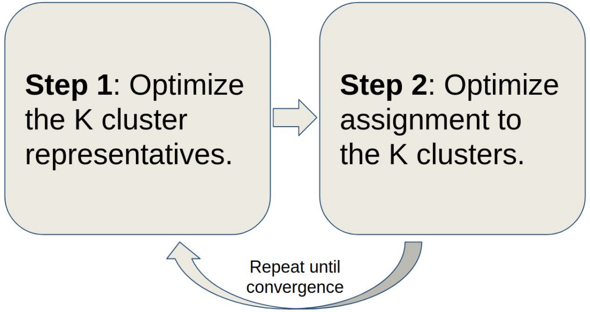
Objective:

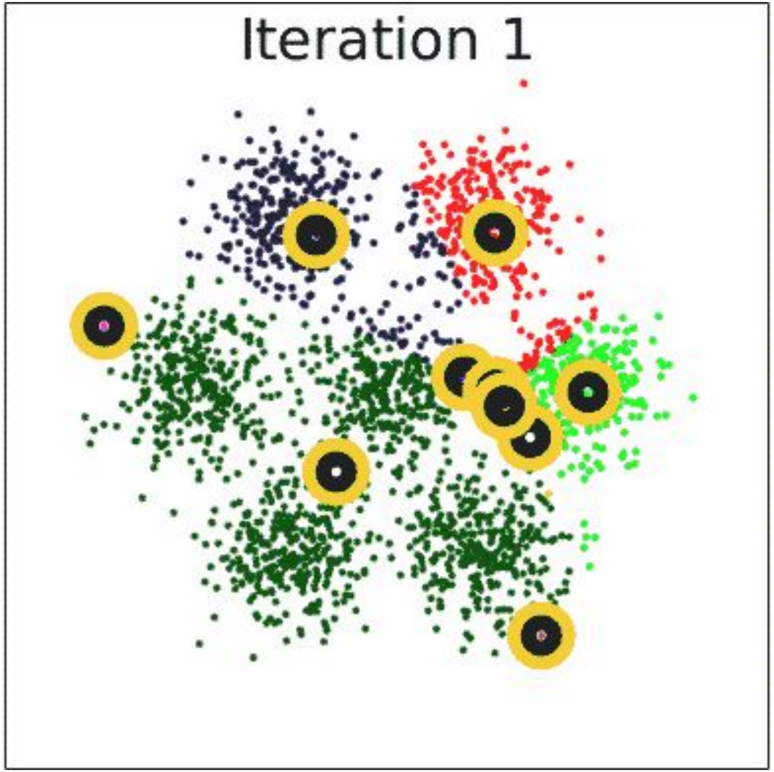
For a fixed number of clusters K, minimize the sum of the squared distances to the cluster centroids.

Both steps reduce the following quantity:

$$\sum_{i=1}^{N} \|x_i - \mu_{k_i}\|^2$$

K-means



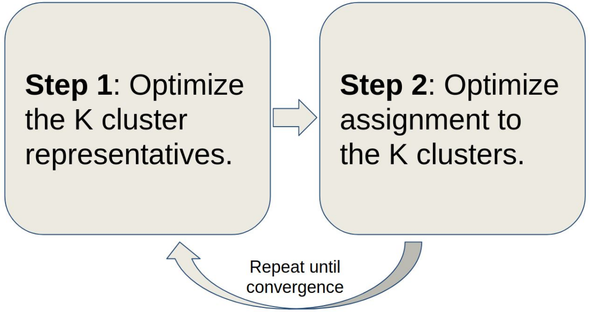


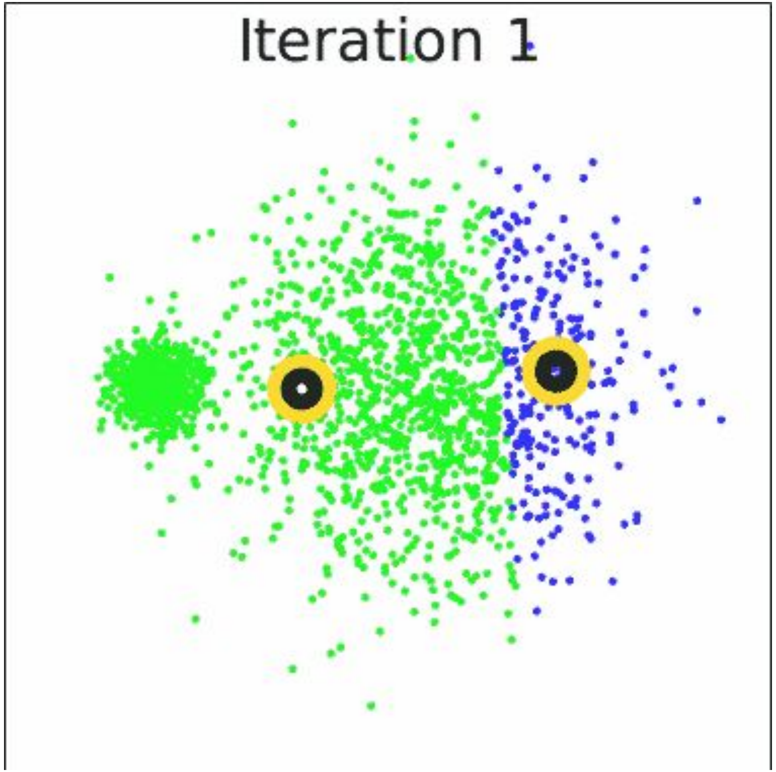
Problem:

How to choose K?

In general, we don't know ahead of time the number of clusters.

K-means



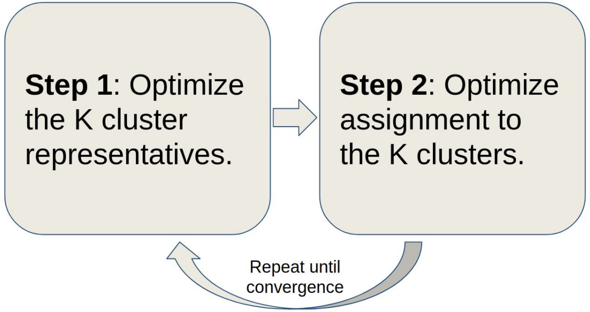


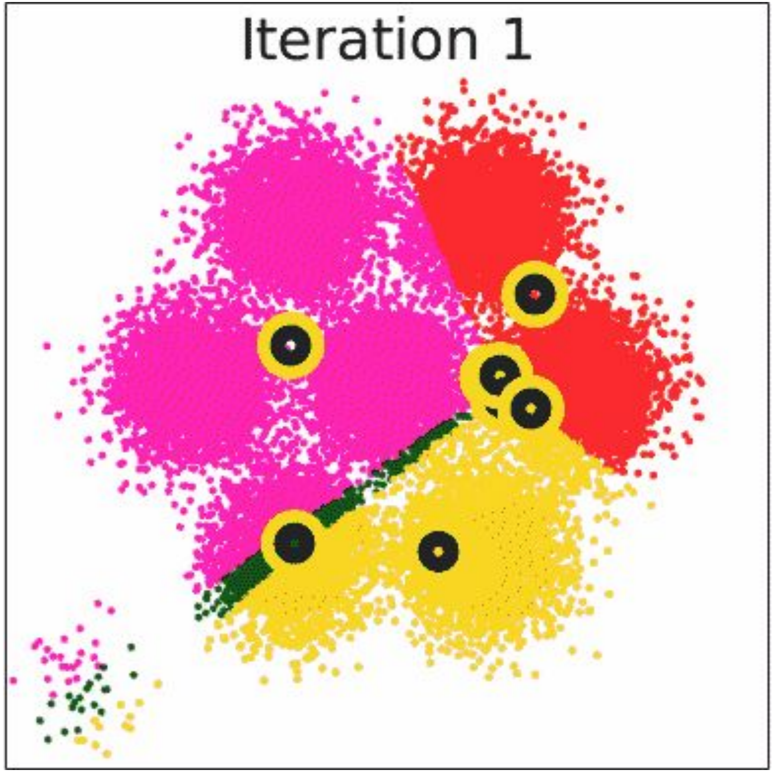
Problem:

Unequal variances

K-means can fail to give the correct decision boundaries.

K-means



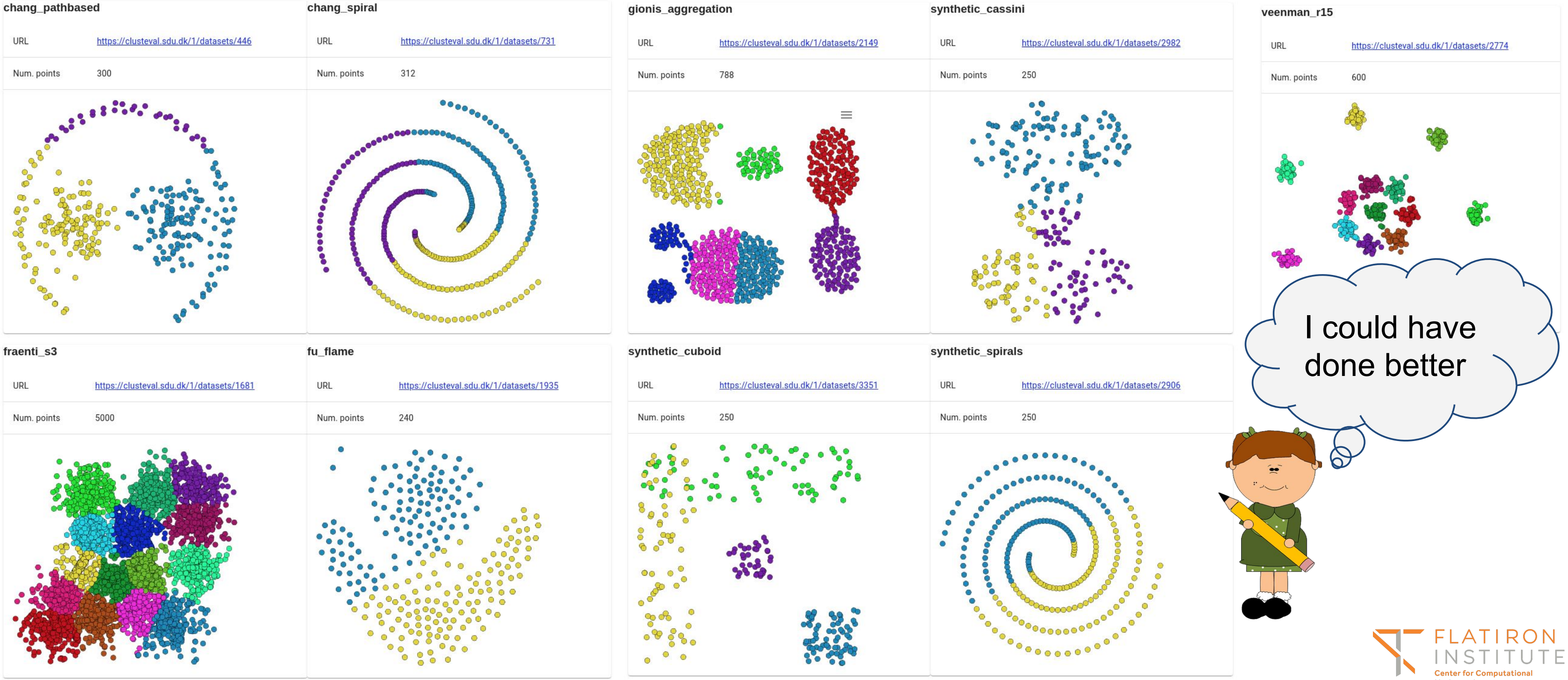


Problem:

Unequal populations

K-means can fail to separate sparse clusters.

K-means (true K)



FLATIRON INSTITUTE
Center for Computational Mathematics

How to run k-means?

From Python:

```
from sklearn.cluster import KMeans
A = KMeans(n_clusters=n_clusters).fit(X)
labels = A.labels_
```

Notes:
Fast, robust, simple
Need to choose K ahead of time
Assumptions about cluster shapes and sizes

Web application on colab notebook

Jeremy Magland, Ph.D.

Senior Data Scientist, Center for Computational Mathematics, Flatiron Institute

Miscellaneous links

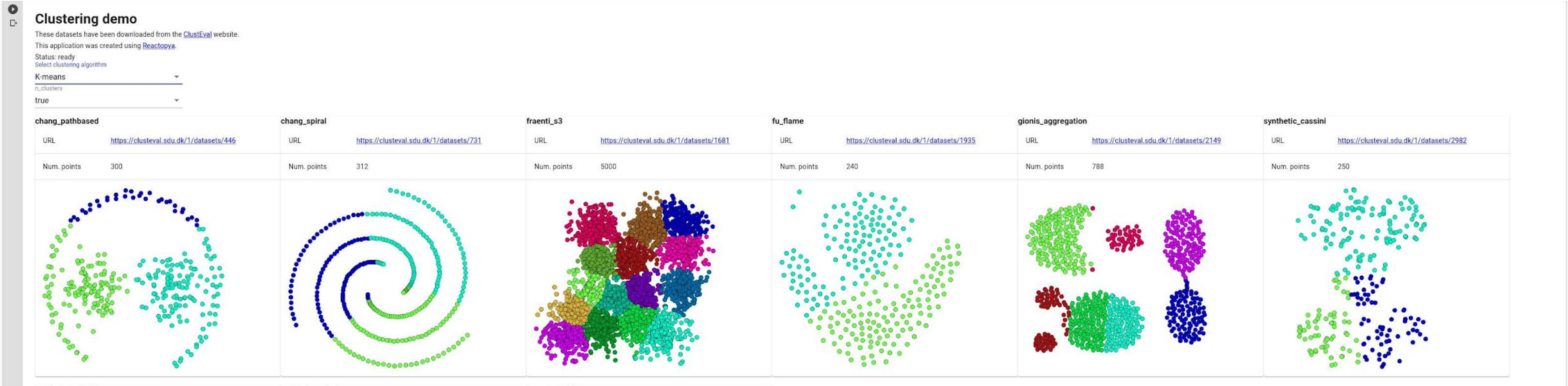
- Spike sorting comparison: SpikeForest
- Example static reactopya snapshots
- Colab notebook: Clustering

https://users.flatironinstitute.org/~magland/

Miscellaneous projects under construction

- Spike sorting tools for Python: SpikeInterface
- Widgets deployable to notebook, desktop, web: reactopya
- Content-addressable storage: kachery
- Represent HDF5 as JSON: h5_to_json
- Widgets for neurophysiology visualization: ephys-viz
- Spike sorting algorithm: MountainSort4

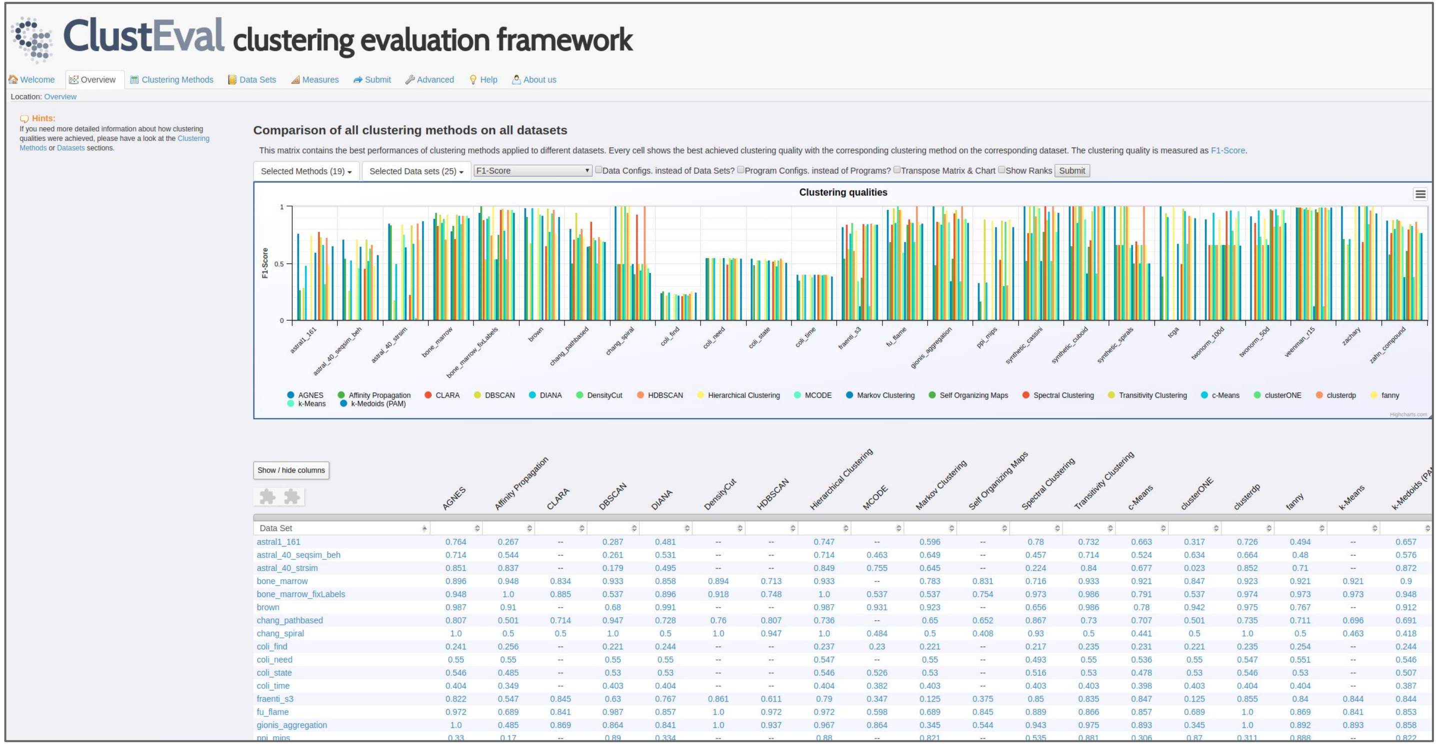
Try out the various algorithms with different parameters using google's compute resources.



https://clusteval.sdu.dk

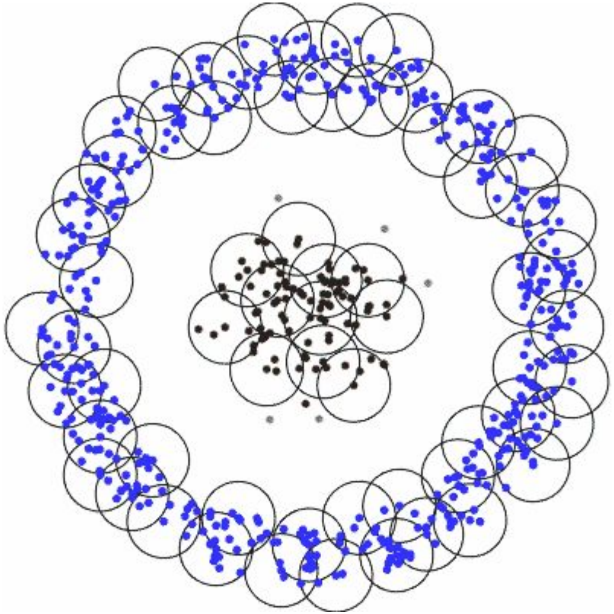
Public website

Evaluates 19 clustering algorithms on 25 datasets.



Data Set	AGNES	Affinity Propagation	CLARA	DBSCAN	DIANA	DensityCut	HDBSCAN	Hierarchical Clustering	MCODE	Markov Clustering	Self Organizing Maps	Spectral Clustering	Transitivity Clustering	c-Means	clusterONE	clusterdp	fanny	k-Means	k-Medoids (PA
astral1_161	0.764	0.267	--	0.287	0.481	--	--	0.747	--	0.596	--	0.78	0.732	0.663	0.317	0.726	0.494	--	0.657
astral_40_seqsim_beh	0.714	0.544	--	0.261	0.531	--	--	0.714	0.463	0.649	--	0.457	0.714	0.524	0.634	0.664	0.48	--	0.576
astral_40_strsim	0.851	0.837	--	0.179	0.495	--	--	0.849	0.755	0.645	--	0.224	0.84	0.677	0.023	0.852	0.71	--	0.872
bone_marrow	0.896	0.948	0.834	0.933	0.858	0.894	0.713	0.933	--	0.783	0.831	0.716	0.933	0.921	0.847	0.923	0.921	0.921	0.9
bone_marrow_fixLabels	0.948	1.0	0.885	0.537	0.896	0.918	0.748	1.0	0.537	0.537	0.754	0.973	0.986	0.791	0.537	0.974	0.973	0.973	0.948
brown	0.987	0.91	--	0.68	0.991	--	--	0.987	0.931	0.923	--	0.656	0.986	0.78	0.942	0.975	0.767	--	0.912
chang_pathbased	0.807	0.501	0.714	0.947	0.728	0.76	0.807	0.736	--	0.65	0.652	0.867	0.73	0.707	0.501	0.735	0.711	0.696	0.691
chang_spiral	1.0	0.5	0.5	1.0	0.5	1.0	0.947	1.0	0.484	0.5	0.408	0.93	0.5	0.441	0.5	1.0	0.5	0.463	0.418
coli_find	0.241	0.256	--	0.221	0.244	--	--	0.237	0.23	0.221	--	0.217	0.235	0.231	0.221	0.235	0.254	--	0.244
coli_need	0.55	0.55	--	0.55	0.55	--	--	0.547	--	0.55	--	0.493	0.55	0.536	0.55	0.547	0.551	--	0.546
coli_state	0.546	0.485	--	0.53	0.53	--	--	0.546	0.526	0.53	--	0.516	0.53	0.478	0.53	0.546	0.53	--	0.507
coli_time	0.404	0.349	--	0.403	0.404	--	--	0.404	0.382	0.403	--	0.403	0.403	0.398	0.403	0.404	0.404	--	0.387
fraenti_s3	0.822	0.547	0.845	0.63	0.767	0.861	0.611	0.79	0.347	0.125	0.375	0.85	0.835	0.847	0.125	0.855	0.84	0.844	0.844
fu_flame	0.972	0.689	0.841	0.987	0.857	1.0	0.972	0.972	0.598	0.689	0.845	0.889	0.866	0.857	0.689	1.0	0.869	0.841	0.853
gionis_aggregation	1.0	0.485	0.869	0.864	0.841	1.0	0.937	0.967	0.864	0.345	0.544	0.943	0.975	0.893	0.345	1.0	0.892	0.893	0.858
ppi_mips	0.33	0.17	--	0.89	0.334	--	--	0.88	--	0.821	--	0.535	0.881	0.306	0.87	0.311	0.888	--	0.822

DBSCAN

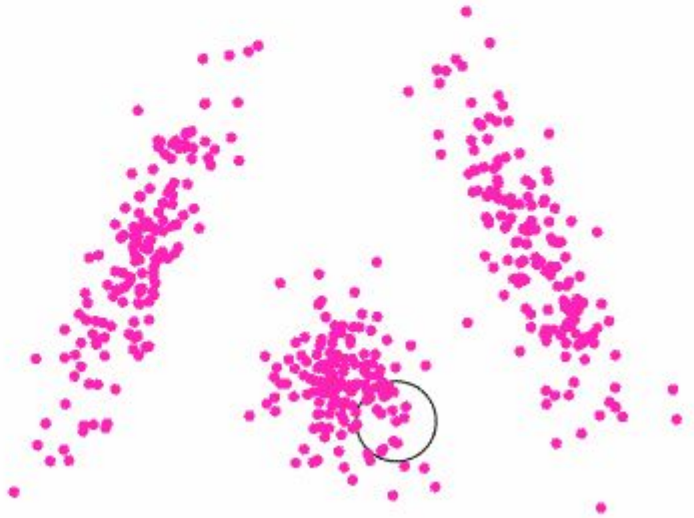


Two parameters: ϵ, minPts

A core neighborhood is an ϵ-neighborhood containing at least minPts.

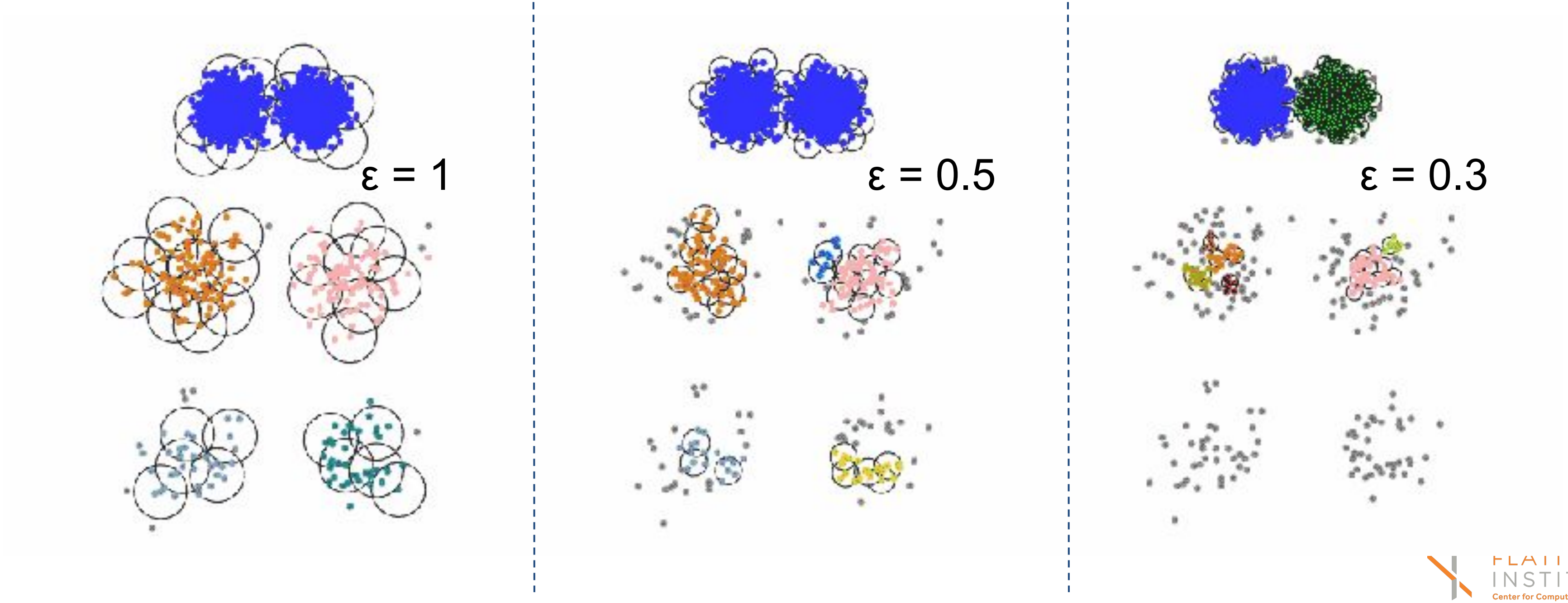
A cluster is the points in a connected union of core neighborhoods.

DBSCAN



DBSCAN

Has trouble with clusters of varying densities
No single choice of parameter works for all clusters



FLATIRON INSTITUTE
Center for Computational Mathematics

DBSCAN

chang_pathbased

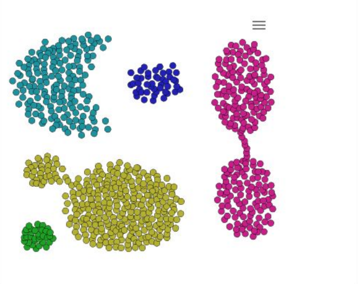
URL	https://clusteval.sdu.dk/1/datasets/446
Num. points	300

chang_spiral

URL	https://clusteval.sdu.dk/1/datasets/731
Num. points	312

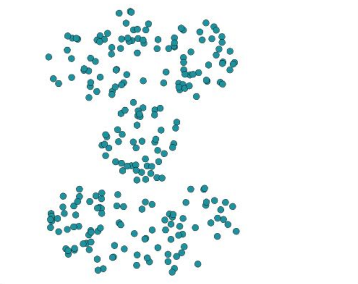
gionis_aggregation

URL	https://clusteval.sdu.dk/1/datasets/2149
Num. points	788



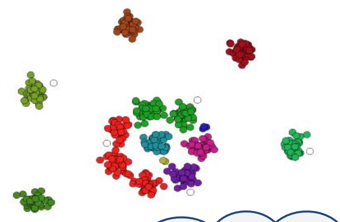
synthetic_cassini

URL	https://clusteval.sdu.dk/1/datasets/2982
Num. points	250



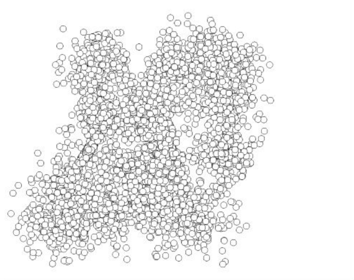
veenman_r15

URL	https://clusteval.sdu.dk/1/datasets/2774
Num. points	600



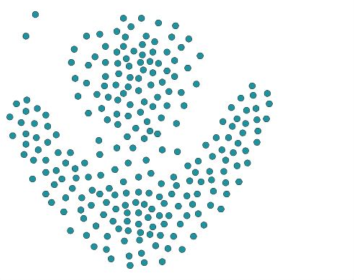
fraenti_s3

URL	https://clusteval.sdu.dk/1/datasets/1681
Num. points	5000



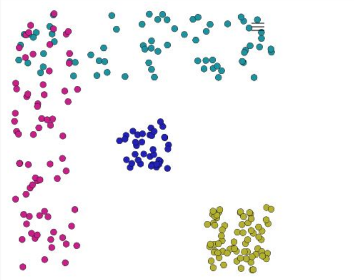
fu_flame

URL	https://clusteval.sdu.dk/1/datasets/1935
Num. points	240



synthetic_cuboid

URL	https://clusteval.sdu.dk/1/datasets/3351
Num. points	250



synthetic_spirals

URL	https://clusteval.sdu.dk/1/datasets/2906
Num. points	250

How to run DBSCAN?

From Python:

```
from sklearn.cluster import DBSCAN
A = DBSCAN(eps=3, min_samples=2).fit(X)
labels = A.labels_
```

Notes:

Fast - even for large numbers of points

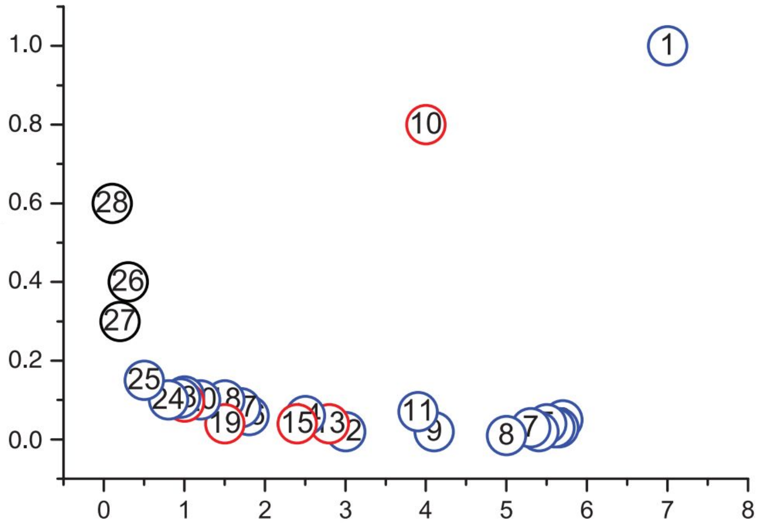
Good for irregular cluster shapes

Choice of neighborhood size and minPts is a problem

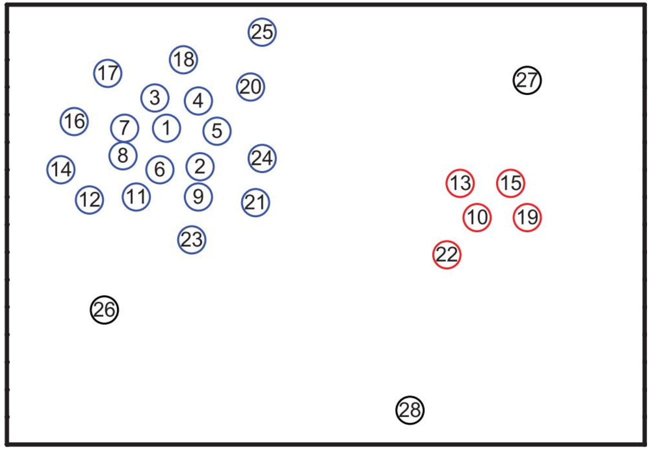
Not good for higher # dimensions

Rodriguez-Laio

Distance to nearest point with higher density

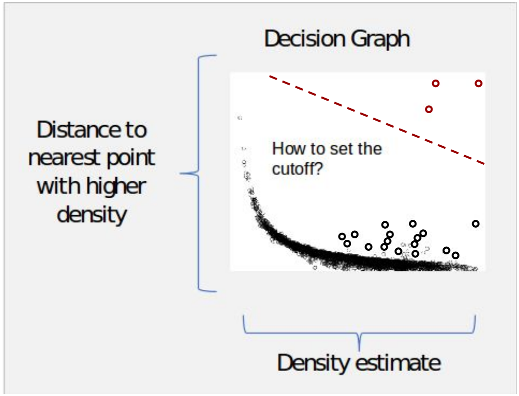


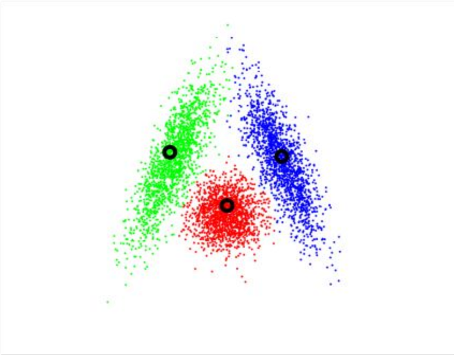
Density estimate



Rodriguez and Laio, Science, 2014

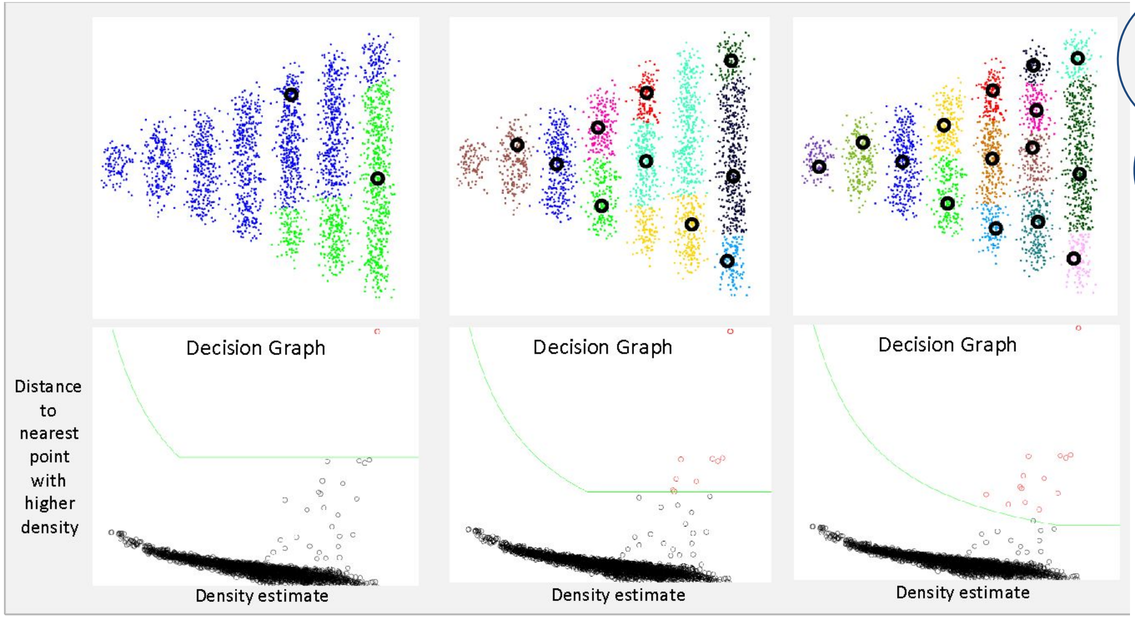
Rodriguez-Laio





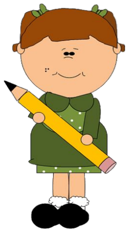
Flatiron Institute
Center for Computational Mathematics

Rodriguez-Laio



No single choice of cutoff parameter works for all clusters

This technique looks promising. That example was cherry-picked to make R-L fail, so I think further investigation is required to determine whether I could do better in general.



How to run Rodriguez-Laio?

Not sure the best way to run Rodriguez-Laio in Python.

Some methods require input of the full distance matrix.

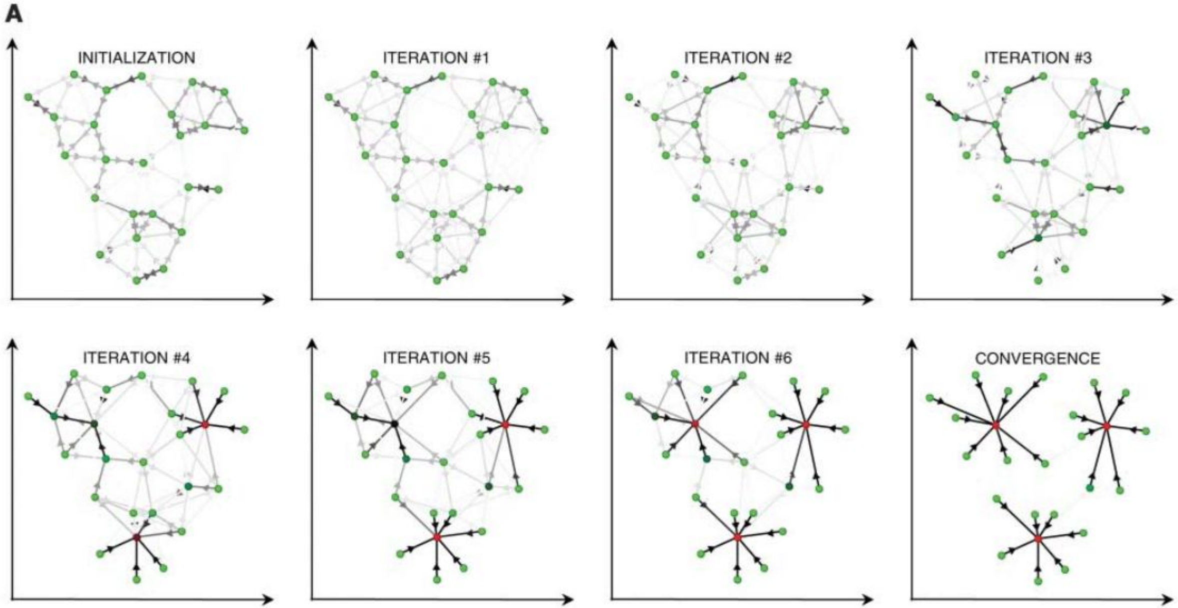
Also known as dpclust or density-peak clustering.

Someone should write a nice Python implementation for points in low dimensions.

Notes:
Further exploration required
Minimal assumptions about cluster shapes
How to choose the cutoff rule?
Certain cluster shapes are not handled well

Affinity propagation

Messages are passed between nearby data points to determine optimal “exemplars” for clusters.

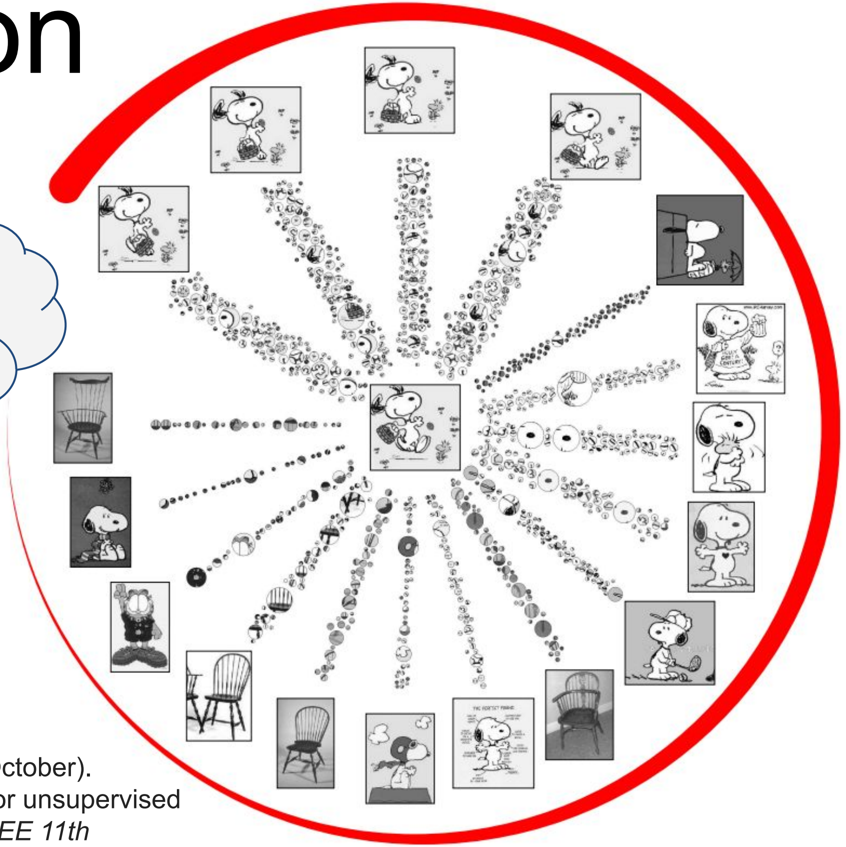


Frey, B. J., & Dueck, D. (2007). Clustering by passing messages between data points. *science*, *315*(5814), 972-976.

Affinity propagation

Advantages

- Works with similarities between data points (no need to embed in vector space)
- No need to specify # clusters *a priori*

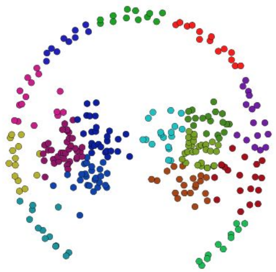


Dueck, D., & Frey, B. J. (2007, October). Non-metric affinity propagation for unsupervised image categorization. In *2007 IEEE 11th International Conference on Computer Vision* (pp. 1-8). IEEE.

Affinity propagation

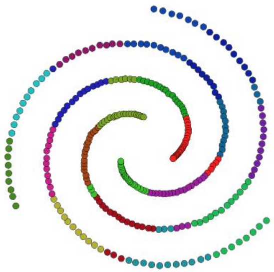
chang_pathbased

URL	https://clusteval.sdu.dk/1/datasets/446
Num. points	300



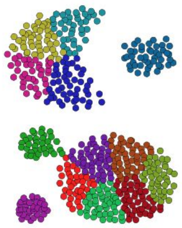
chang_spiral

URL	https://clusteval.sdu.dk/1/datasets/731
Num. points	312



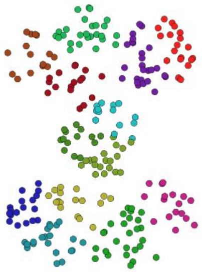
gionis_aggregation

URL	https://clusteval.sdu.dk/1/datasets/2149
Num. points	788



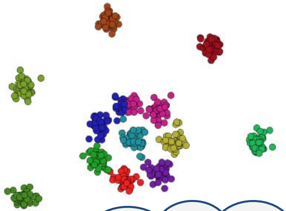
synthetic_cassini

URL	https://clusteval.sdu.dk/1/datasets/2982
Num. points	250



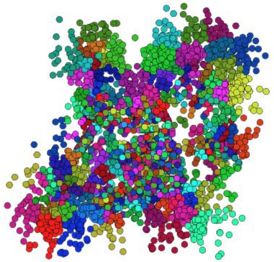
veenman_r15

URL	https://clusteval.sdu.dk/1/datasets/2774
Num. points	600



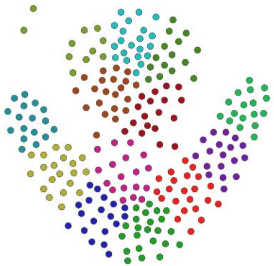
fraenti_s3

URL	https://clusteval.sdu.dk/1/datasets/1681
Num. points	5000



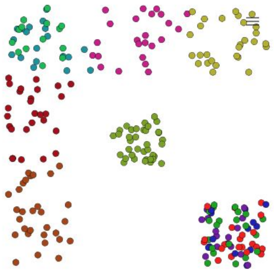
fu_flame

URL	https://clusteval.sdu.dk/1/datasets/1935
Num. points	240



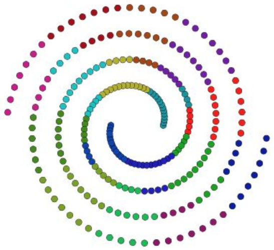
synthetic_cuboid

URL	https://clusteval.sdu.dk/1/datasets/3351
Num. points	250



synthetic_spirals

URL	https://clusteval.sdu.dk/1/datasets/2906
Num. points	250



How to run affinity propagation?

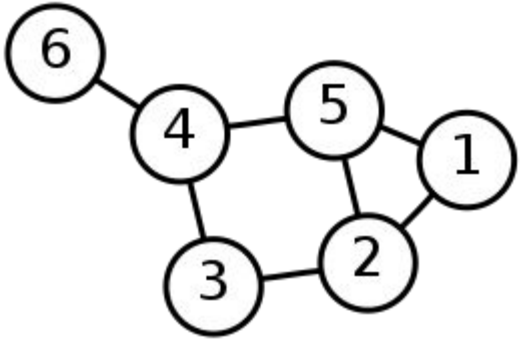
From Python:

```
from sklearn.cluster import AffinityPropagation
A = AffinityPropagation(bandwidth=0.5).fit(X)
labels = A.labels_
```

Notes:
Does not require embedding in vector space
Good for irregular cluster shapes and variable sizes
SLOW - not scalable with number of datapoints
Seems to fail in simple cases

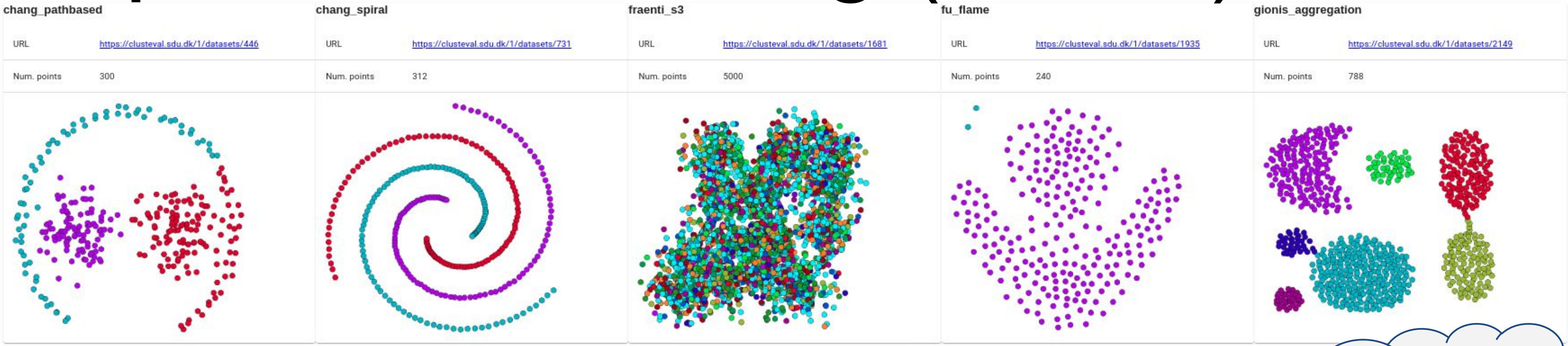
Spectral clustering

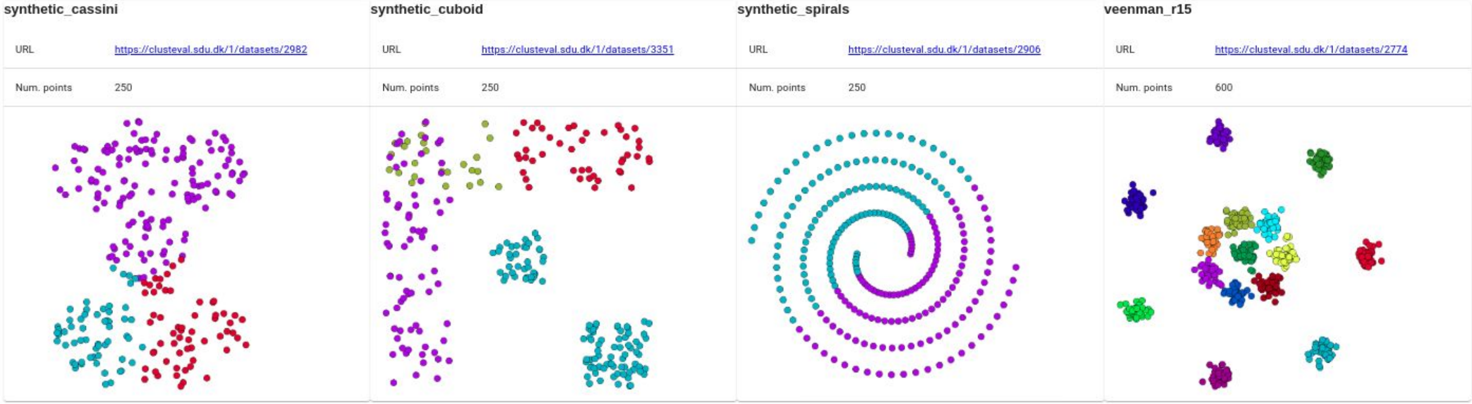
Was discussed earlier by Marina.



$$\begin{pmatrix} 2 & -1 & 0 & 0 & -1 & 0 \\ -1 & 3 & -1 & 0 & -1 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & -1 & 3 & -1 & -1 \\ -1 & -1 & 0 & -1 & 3 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 \end{pmatrix}$$

Spectral clustering (true K)





How to run spectral clustering?

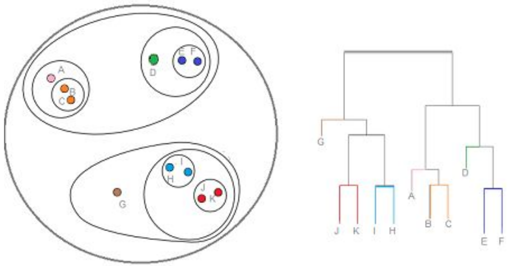
From Python:

```
from sklearn.cluster import SpectralClustering
A = SpectralClustering(n_clusters=5).fit(X)
labels = A.labels_
```

Notes:
Mostly an embedding - requires a second step for labeling
Medium scalability

Agglomerative clustering

- Begin with each point in its own cluster
- Merge clusters one at a time based on a measure of similarity
- Different options for similarity
  - **Single linkage** - based on closest pair of points
  - **Complete linkage** - based on furthest pair of points
  - **Average linkage** - based on average distances between all pairs
  - **Ward** - global objective function (minimizes within-cluster variances)
  - etc.
- Construct a dendrogram
- A second step is needed to obtain labels from dendrogram (e.g., specify K)



https://www.statisticshowto.datasciencecentral.com/hierarchical-clustering

FLATIRON INSTITUTE
Center for Computational Mathematics

Agglomerative clustering (true K)

chang_pathbased

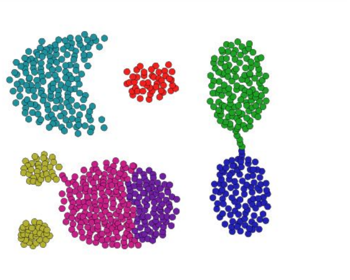
URL	https://clusteval.sdu.dk/1/datasets/446
Num. points	300

chang_spiral

URL	https://clusteval.sdu.dk/1/datasets/731
Num. points	312

gionis_aggregation

URL	https://clusteval.sdu.dk/1/datasets/2149
Num. points	788

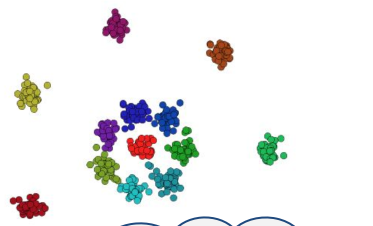


synthetic_cassini

URL	https://clusteval.sdu.dk/1/datasets/2982
Num. points	250

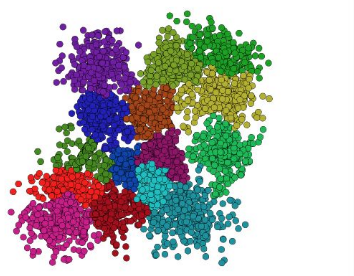
veenman_r15

URL	https://clusteval.sdu.dk/1/datasets/2774
Num. points	600



fraenti_s3

URL	https://clusteval.sdu.dk/1/datasets/1681
Num. points	5000

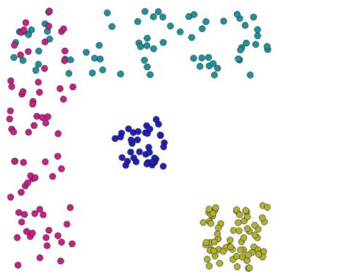


fu_flame

URL	https://clusteval.sdu.dk/1/datasets/1935
Num. points	240

synthetic_cuboid

URL	https://clusteval.sdu.dk/1/datasets/3351
Num. points	250



synthetic_spirals

URL	https://clusteval.sdu.dk/1/datasets/2906
Num. points	250

FLATIRON INSTITUTE
Center for Computational Mathematics

How to run agglomerative clustering?

From Python:

```
from sklearn.cluster import AgglomerativeClustering
A = AgglomerativeClustering(n_clusters=5).fit(X)
labels = A.labels_
```

Notes:

Fast and scalable

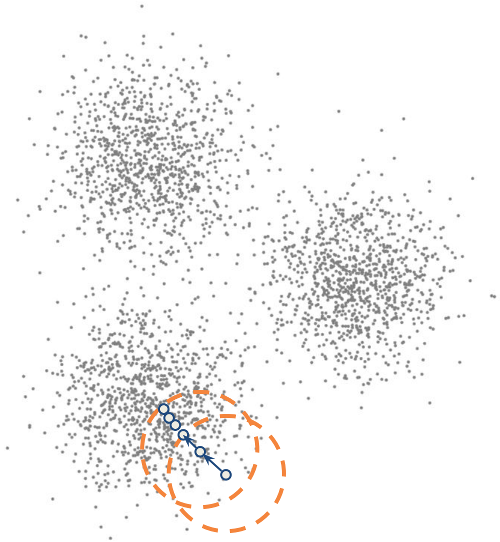
Does not require embedding in vector space

Good for irregular cluster shapes and variable sizes

Need to specify number of clusters or some other criteria for cutting dendrogram

Many choices for linkage / merge criteria

Mean shift



- Clusters are basins of attraction for the mean shift operation
- Mean shift iteration:
  - Compute weighted centroid of nearby points
  - Move to centroid’
- Requires choice of bandwidth / kernel function

Mean shift (bandwidth = auto)

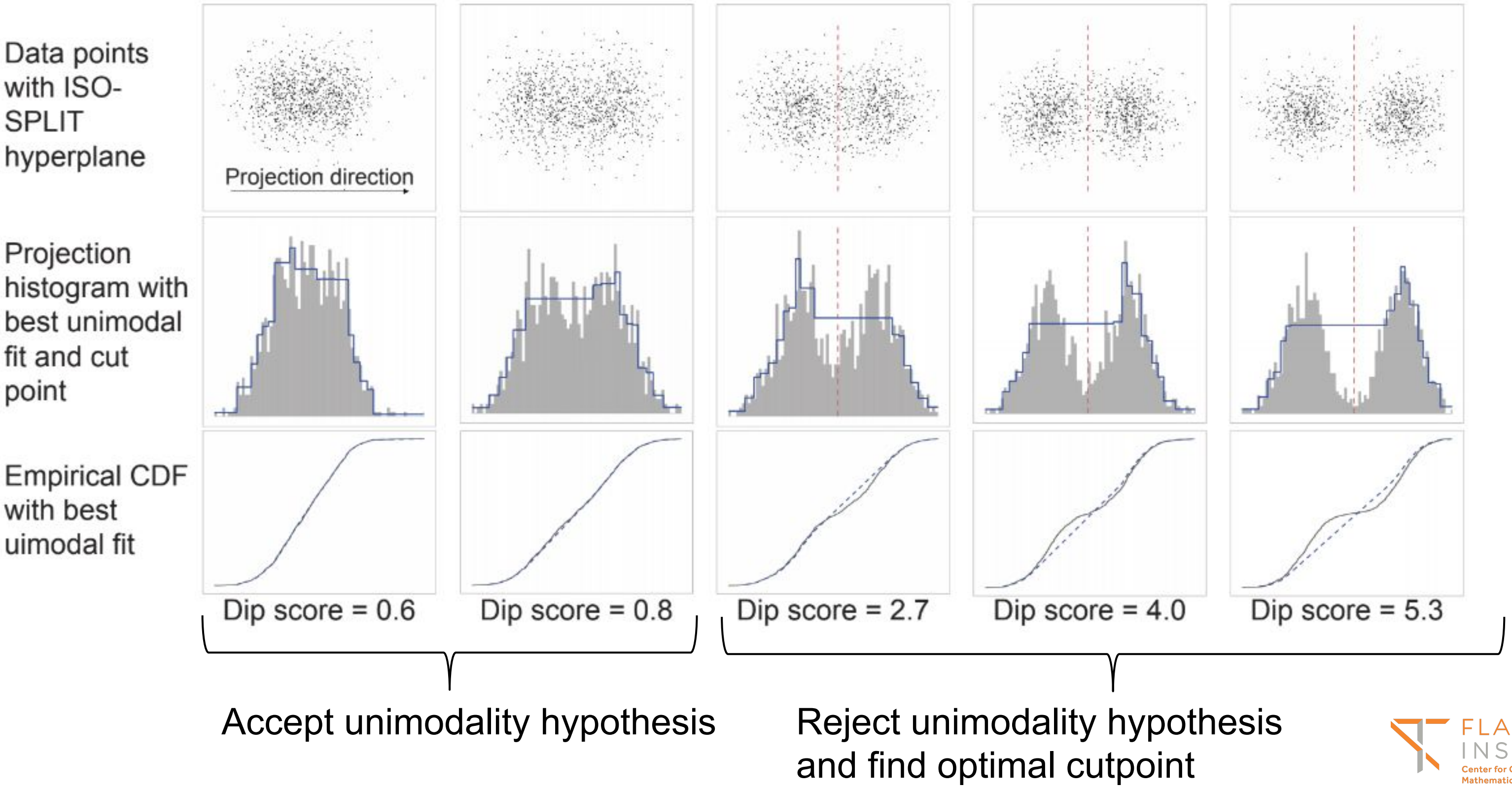
How to run mean shift clustering?

From Python:

```
from sklearn.cluster import MeanShift
A = MeanShift(bandwidth=3).fit(X)
labels = A.labels_
```

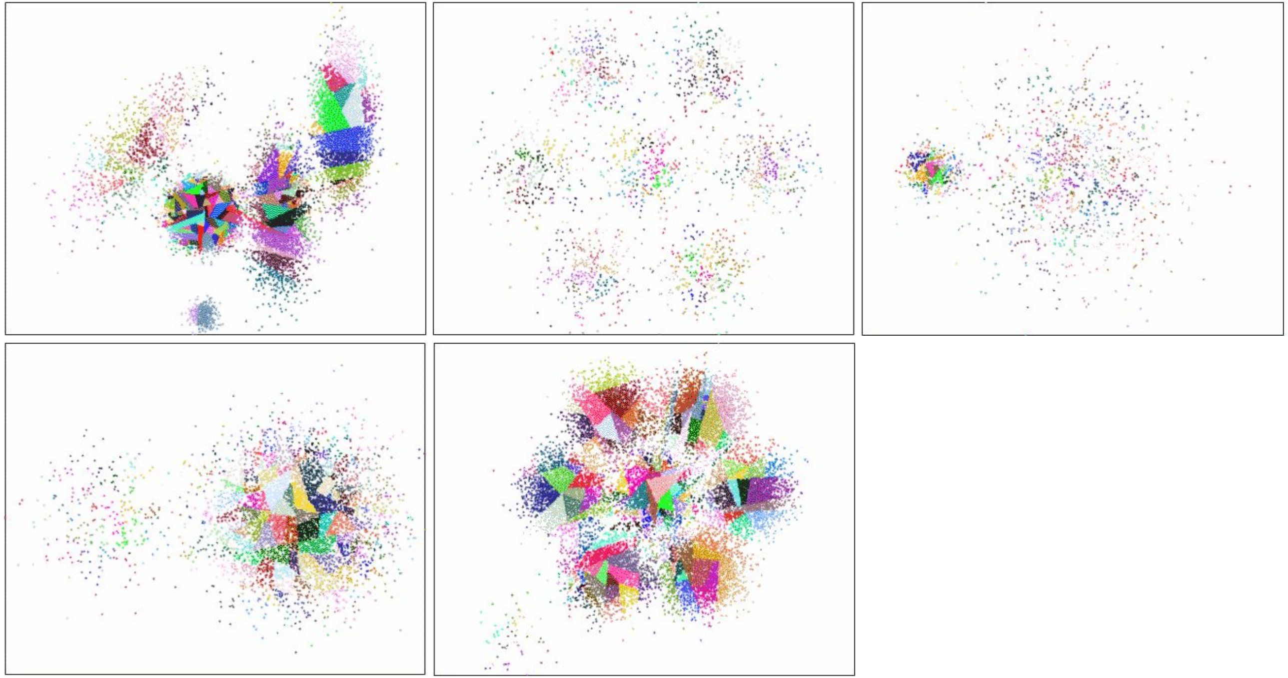
Notes:
Slow and not very good, it seems

ISO-SPLIT (1-d)



FLATIRON INSTITUTE
Center for Computational Mathematics

ISO-SPLIT (2-d)

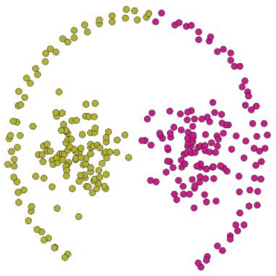



Looks good to me!
But I think you were biased in picking those examples to make your own method look good.

ISO-SPLIT

chang_pathbased

URL	https://clusteval.sdu.dk/1/datasets/446
Num. points	300

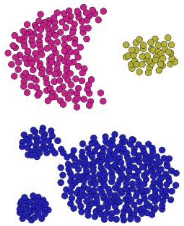


chang_spiral

URL	https://clusteval.sdu.dk/1/datasets/731
Num. points	312

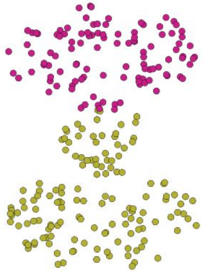
gionis_aggregation

URL	https://clusteval.sdu.dk/1/datasets/2149
Num. points	788



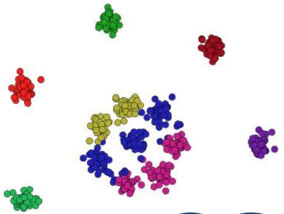
synthetic_cassini

URL	https://clusteval.sdu.dk/1/datasets/2982
Num. points	250



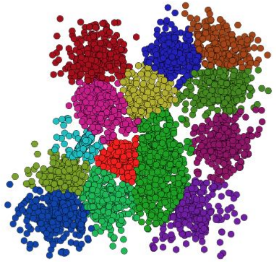
veenman_r15

URL	https://clusteval.sdu.dk/1/datasets/2774
Num. points	600



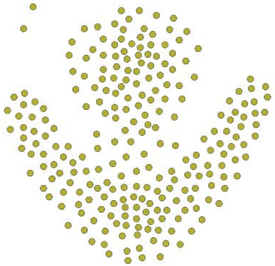
fraenti_s3

URL	https://clusteval.sdu.dk/1/datasets/1681
Num. points	5000



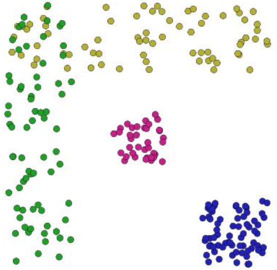
fu_flame

URL	https://clusteval.sdu.dk/1/datasets/1935
Num. points	240



synthetic_cuboid

URL	https://clusteval.sdu.dk/1/datasets/3351
Num. points	250



synthetic_spirals

URL	https://clusteval.sdu.dk/1/datasets/2906
Num. points	250

FLATIRON INSTITUTE
Center for Computational Mathematics

How to run ISO-SPLIT?

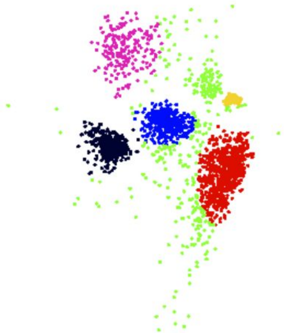
From Python:

```
from isosplit5 import isosplit5
labels = isosplit5(X)
```

Notes:
Unimodal clusters
No adjustable parameters
Fast
Not great for small datasets (statistical test benefits from large number of points)
Cannot handle non-unimodal clusters

Did not cover mixture models

- Mixture models require *a priori* assumptions about cluster distributions
- Clustering fails when assumptions are not met
- Typically there are many adjustable parameters
- Overfitting via overlapping clusters



Gaussian mixture model with known K

Summary

- Clustering in low dimensions is difficult to define
- A variety of techniques attempt to tackle the problem
- It appears that in simple 2-d examples, an eight-year-old child can do a better job than any of the techniques we explored
- There is room for improvement
- There is a web app to help decide which technique and parameters are suitable
- Future directions
  - Expand the web app (more datasets, more algorithms)
  - Improve ISO-SPLIT or create yet another alternative

Thank you for listening!

- Colab notebook: Clustering

https://users.flatironinstitute.org/~magland/