Guest Lecture: Clustering

at IST Austria Core Course 2018/19

Christoph Lampert, Machine Learning and Computer Group

some material: courtesy of Shai Ben-David

What is the goal of Science?

“The purpose of science is to find meaningful simplicity

in the midst of disorderly complexity”

Herbert Simon

What is the goal of Science?

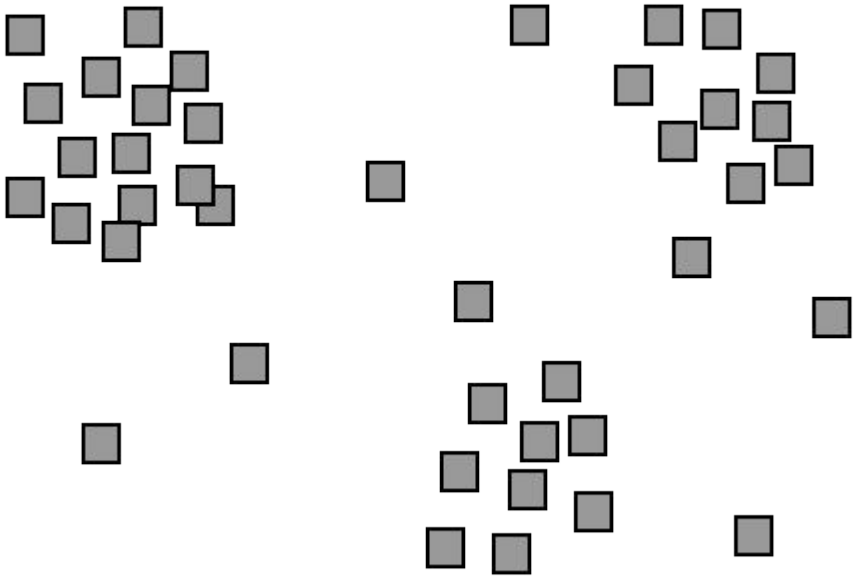
“The purpose of science is to find meaningful simplicity in the midst of disorderly complexity”

Herbert Simon

This can also serve to describe the goal of **clustering.**

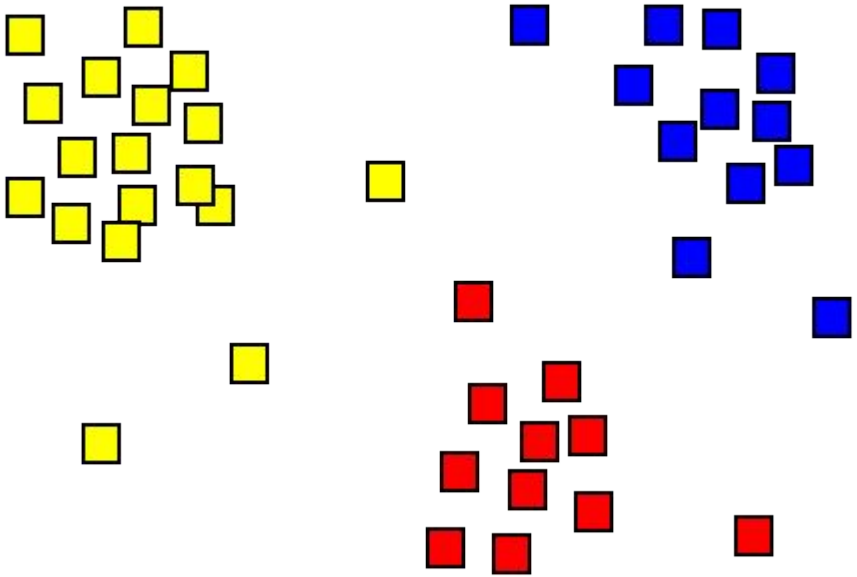
- given a lot of complex data, find simple structures

Clustering aka Cluster Analysis



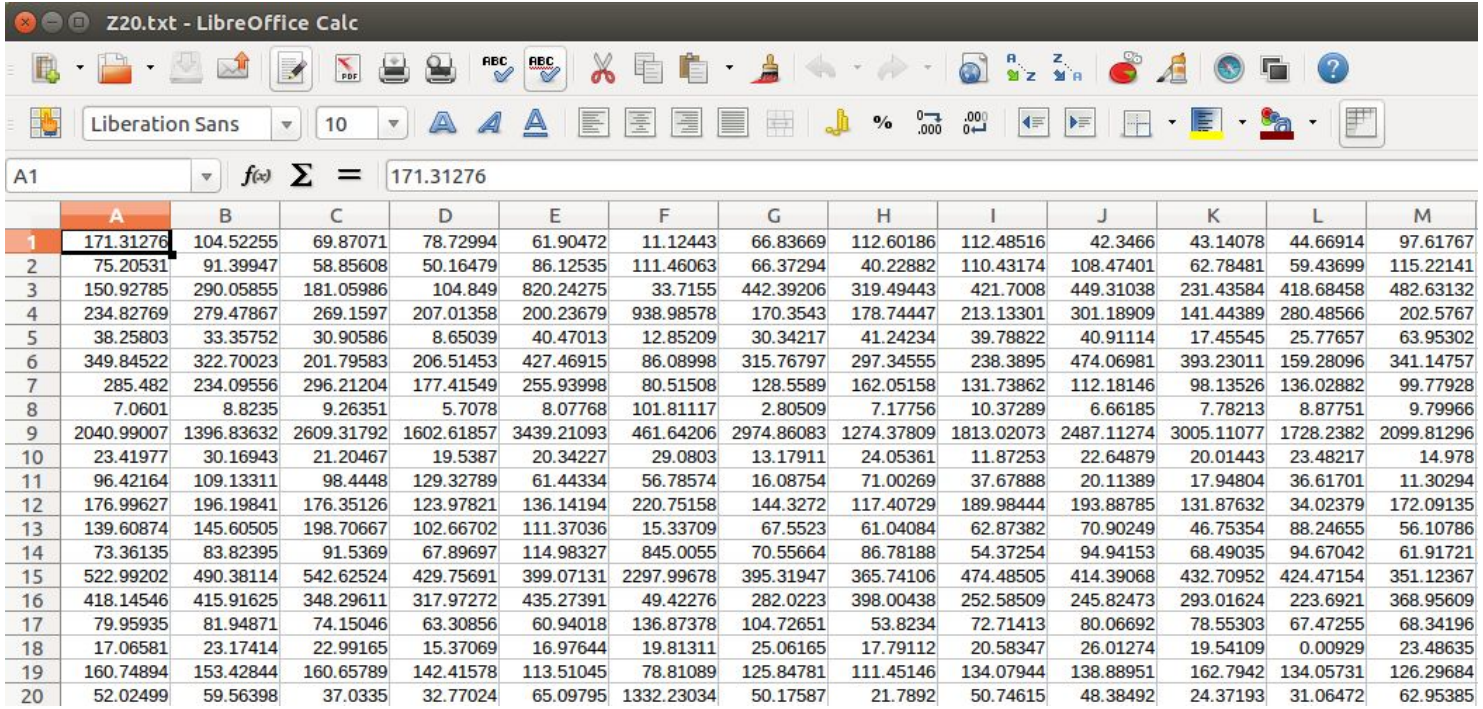
data

Clustering aka Cluster Analysis



clustered data

Real data



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	171.31276	104.52255	69.87071	78.72994	61.90472	11.12443	66.83669	112.60186	112.48516	42.3466	43.14078	44.66914	97.61767
2	75.20531	91.39947	58.85608	50.16479	86.12535	111.46063	66.37294	40.22882	110.43174	108.47401	62.78481	59.43699	115.22141
3	150.92785	290.05855	181.05986	104.849	820.24275	33.7155	442.39206	319.49443	421.7008	449.31038	231.43584	418.68458	482.63132
4	234.82769	279.47867	269.1597	207.01358	200.23679	938.98578	170.3543	178.74447	213.13301	301.18909	141.44389	280.48566	202.5767
5	38.25803	33.35752	30.90586	8.65039	40.47013	12.85209	30.34217	41.24234	39.78822	40.91114	17.45545	25.77657	63.95302
6	349.84522	322.70023	201.79583	206.51453	427.46915	86.08998	315.76797	297.34555	238.3895	474.06981	393.23011	159.28096	341.14757
7	285.482	234.09556	296.21204	177.41549	255.93998	80.51508	128.5589	162.05158	131.73862	112.18146	98.13526	136.02882	99.77928
8	7.0601	8.8235	9.26351	5.7078	8.07768	101.81117	2.80509	7.17756	10.37289	6.66185	7.78213	8.87751	9.79966
9	2040.99007	1396.83632	2609.31792	1602.61857	3439.21093	461.64206	2974.86083	1274.37809	1813.02073	2487.11274	3005.11077	1728.2382	2099.81296
10	23.41977	30.16943	21.20467	19.5387	20.34227	29.0803	13.17911	24.05361	11.87253	22.64879	20.01443	23.48217	14.978
11	96.42164	109.13311	98.4448	129.32789	61.44334	56.78574	16.08754	71.00269	37.67888	20.11389	17.94804	36.61701	11.30294
12	176.99627	196.19841	176.35126	123.97821	136.14194	220.75158	144.3272	117.40729	189.98444	193.88785	131.87632	34.02379	172.09135
13	139.60874	145.60505	198.70667	102.66702	111.37036	15.33709	67.5523	61.04084	62.87382	70.90249	46.75354	88.24655	56.10786
14	73.36135	83.82395	91.5369	67.89697	114.98327	845.0055	70.55664	86.78188	54.37254	94.94153	68.49035	94.67042	61.91721
15	522.99202	490.38114	542.62524	429.75691	399.07131	2297.99678	395.31947	365.74106	474.48505	414.39068	432.70952	424.47154	351.12367
16	418.14546	415.91625	348.29611	317.97272	435.27391	49.42276	282.0223	398.00438	252.58509	245.82473	293.01624	223.6921	368.95609
17	79.95935	81.94871	74.15046	63.30856	60.94018	136.87378	104.72651	53.8234	72.71413	80.06692	78.55303	67.47255	68.34196
18	17.06581	23.17414	22.99165	15.37069	16.97644	19.81311	25.06165	17.79112	20.58347	26.01274	19.54109	0.00929	23.48635
19	160.74894	153.42844	160.65789	142.41578	113.51045	78.81089	125.84781	111.45146	134.07944	138.88951	162.7942	134.05731	126.29684
20	52.02499	59.56398	37.0335	32.77024	65.09795	1332.23034	50.17587	21.7892	50.74615	48.38492	24.37193	31.06472	62.95385

what are the “clusters”?

Clustering real data

In real life, data is often:

- high-dimensional
  - each “data point” is described by thousands of numbers
- abundant
  - there might be hundreds of thousands of data points
- structured
  - data points are not just a bunch of numbers,
    but e.g. DNA sequences or networks themselves
- noisy
  - data has measurement errors, missing data, …

how to find the clusters?

General principle

Split the data into (disjoint) groups, such that

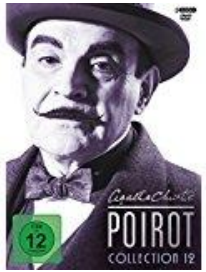
- **similar things are in the same groups,**

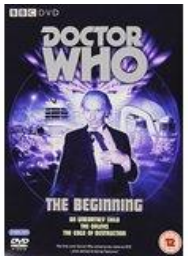
and

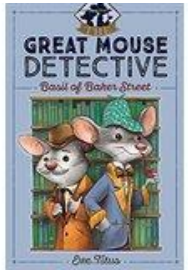
- **non-similar things are in different groups.**

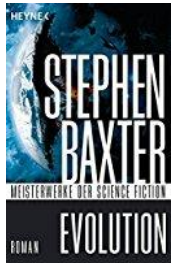
The groups we call clusters.

Clustering real data









what are the “clusters”?

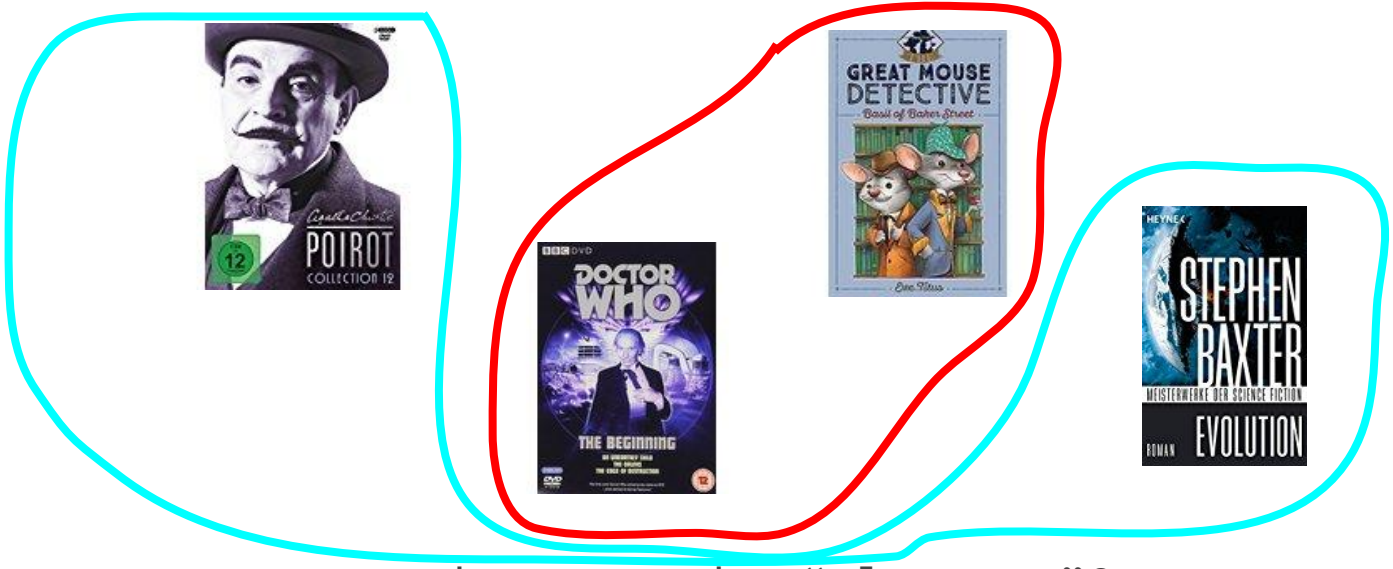
Clustering real data

what are the “clusters”?

Clustering real data

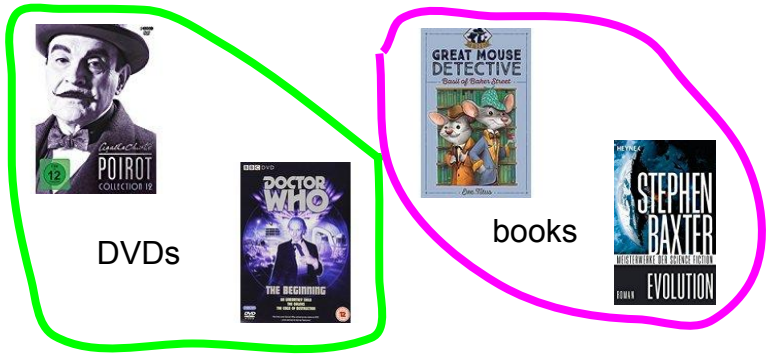
what are the “clusters”?

Clustering real data



what are the “clusters”?

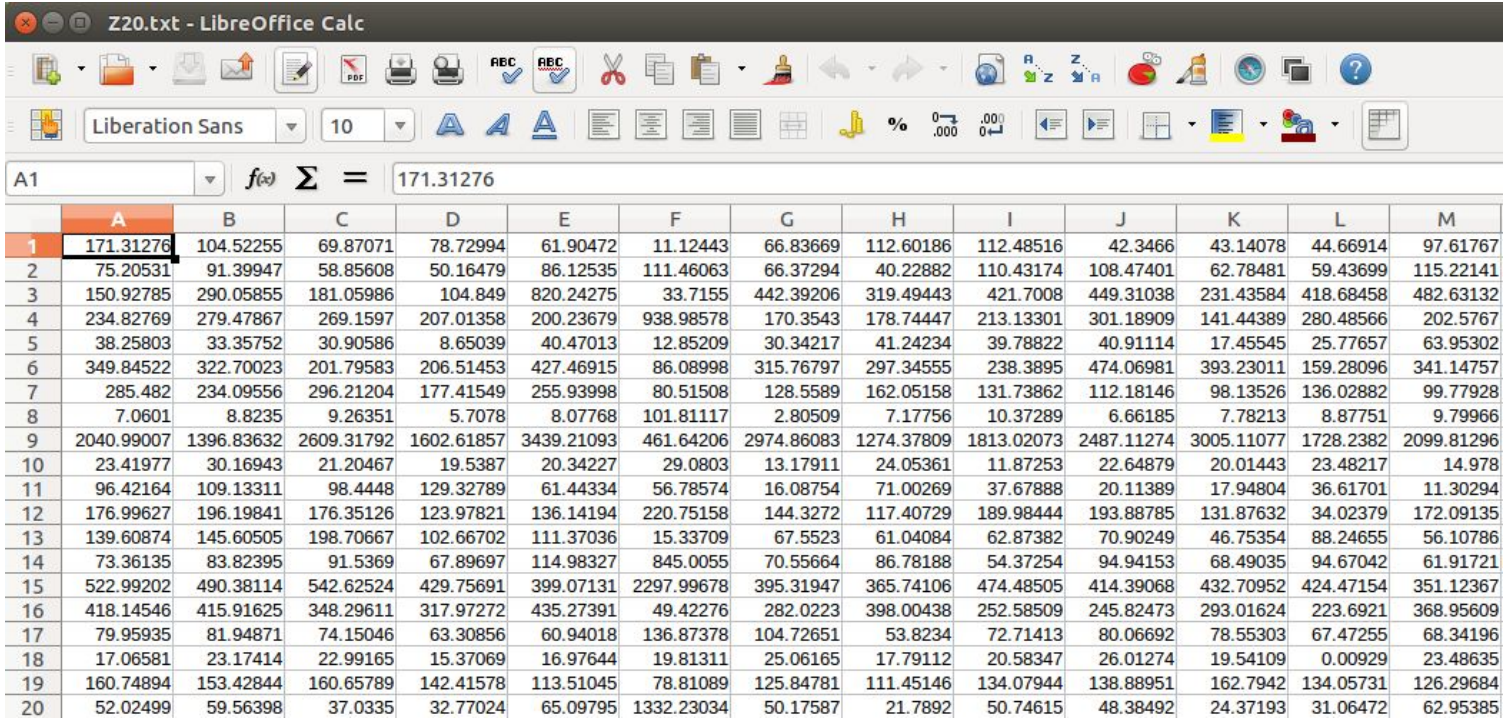
Clustering real data



Insight 1:

- there are no “true” clusters in the data
- what is the “correct” clustering depends on the observer/task/goal/interests/application/...

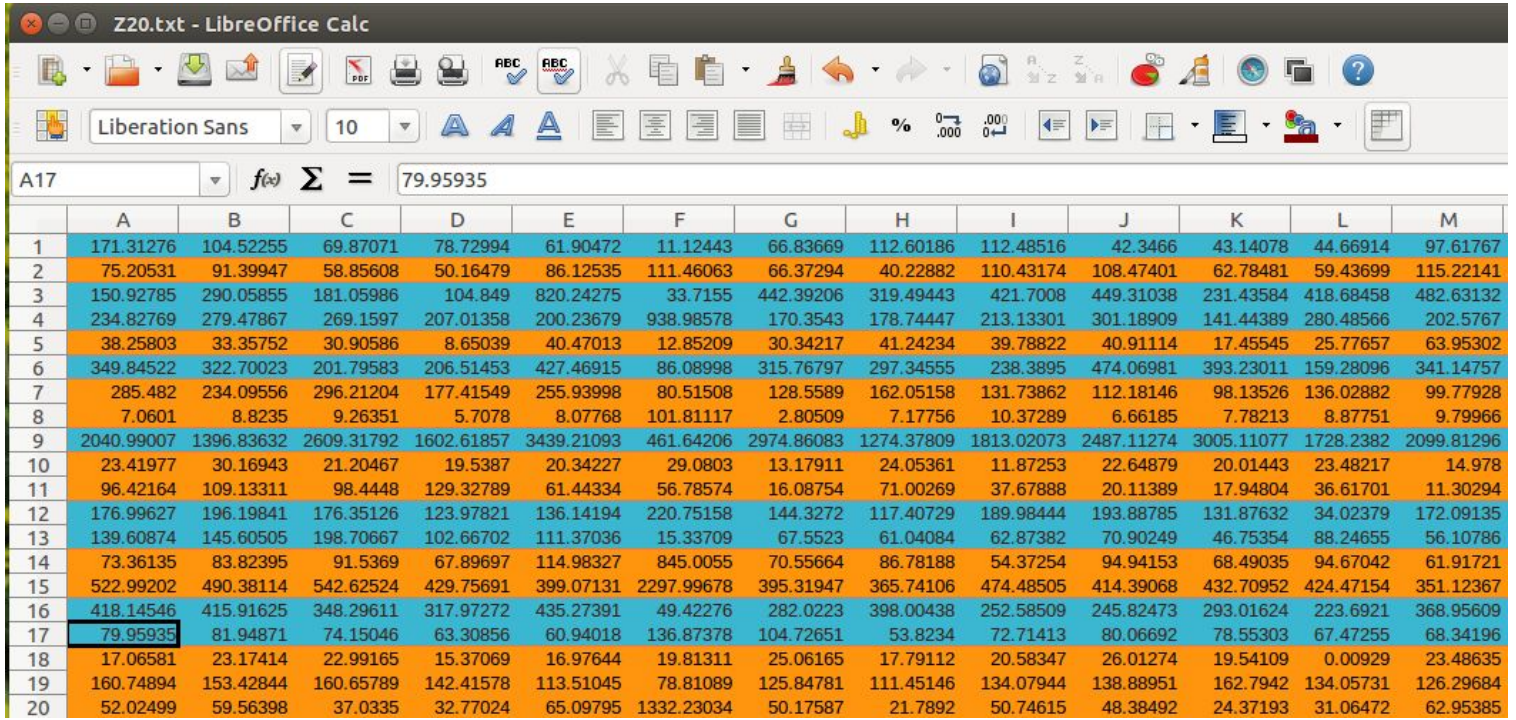
Real data



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	171.31276	104.52255	69.87071	78.72994	61.90472	11.12443	66.83669	112.60186	112.48516	42.3466	43.14078	44.66914	97.61767
2	75.20531	91.39947	58.85608	50.16479	86.12535	111.46063	66.37294	40.22882	110.43174	108.47401	62.78481	59.43699	115.22141
3	150.92785	290.05855	181.05986	104.849	820.24275	33.7155	442.39206	319.49443	421.7008	449.31038	231.43584	418.68458	482.63132
4	234.82769	279.47867	269.1597	207.01358	200.23679	938.98578	170.3543	178.74447	213.13301	301.18909	141.44389	280.48566	202.5767
5	38.25803	33.35752	30.90586	8.65039	40.47013	12.85209	30.34217	41.24234	39.78822	40.91114	17.45545	25.77657	63.95302
6	349.84522	322.70023	201.79583	206.51453	427.46915	86.08998	315.76797	297.34555	238.3895	474.06981	393.23011	159.28096	341.14757
7	285.482	234.09556	296.21204	177.41549	255.93998	80.51508	128.5589	162.05158	131.73862	112.18146	98.13526	136.02882	99.77928
8	7.0601	8.8235	9.26351	5.7078	8.07768	101.81117	2.80509	7.17756	10.37289	6.66185	7.78213	8.87751	9.79966
9	2040.99007	1396.83632	2609.31792	1602.61857	3439.21093	461.64206	2974.86083	1274.37809	1813.02073	2487.11274	3005.11077	1728.2382	2099.81296
10	23.41977	30.16943	21.20467	19.5387	20.34227	29.0803	13.17911	24.05361	11.87253	22.64879	20.01443	23.48217	14.978
11	96.42164	109.13311	98.4448	129.32789	61.44334	56.78574	16.08754	71.00269	37.67888	20.11389	17.94804	36.61701	11.30294
12	176.99627	196.19841	176.35126	123.97821	136.14194	220.75158	144.3272	117.40729	189.98444	193.88785	131.87632	34.02379	172.09135
13	139.60874	145.60505	198.70667	102.66702	111.37036	15.33709	67.5523	61.04084	62.87382	70.90249	46.75354	88.24655	56.10786
14	73.36135	83.82395	91.5369	67.89697	114.98327	845.0055	70.55664	86.78188	54.37254	94.94153	68.49035	94.67042	61.91721
15	522.99202	490.38114	542.62524	429.75691	399.07131	2297.99678	395.31947	365.74106	474.48505	414.39068	432.70952	424.47154	351.12367
16	418.14546	415.91625	348.29611	317.97272	435.27391	49.42276	282.0223	398.00438	252.58509	245.82473	293.01624	223.6921	368.95609
17	79.95935	81.94871	74.15046	63.30856	60.94018	136.87378	104.72651	53.8234	72.71413	80.06692	78.55303	67.47255	68.34196
18	17.06581	23.17414	22.99165	15.37069	16.97644	19.81311	25.06165	17.79112	20.58347	26.01274	19.54109	0.00929	23.48635
19	160.74894	153.42844	160.65789	142.41578	113.51045	78.81089	125.84781	111.45146	134.07944	138.88951	162.7942	134.05731	126.29684
20	52.02499	59.56398	37.0335	32.77024	65.09795	1332.23034	50.17587	21.7892	50.74615	48.38492	24.37193	31.06472	62.95385

what are the “clusters”?

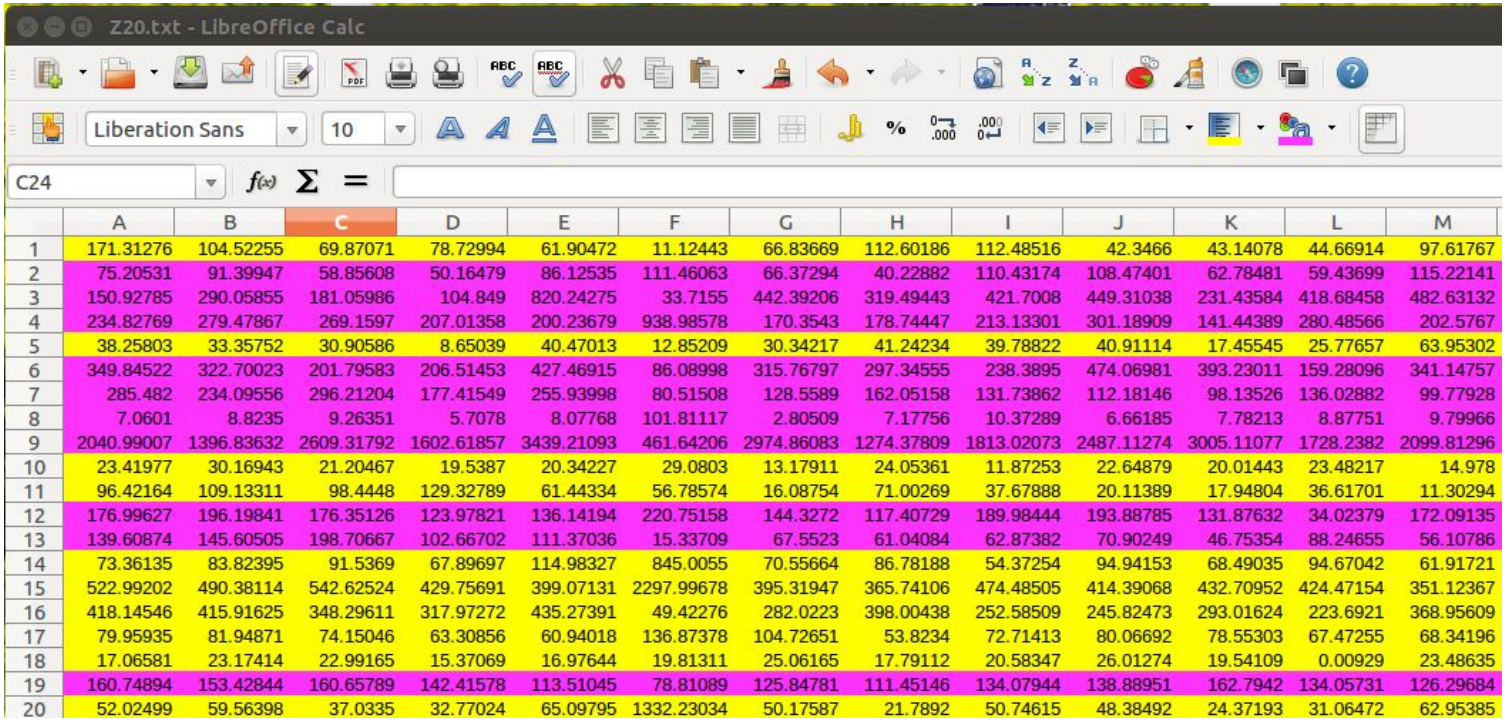
Real data



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	171.31276	104.52255	69.87071	78.72994	61.90472	11.12443	66.83669	112.60186	112.48516	42.3466	43.14078	44.66914	97.61767
2	75.20531	91.39947	58.85608	50.16479	86.12535	111.46063	66.37294	40.22882	110.43174	108.47401	62.78481	59.43699	115.22141
3	150.92785	290.05855	181.05986	104.849	820.24275	33.7155	442.39206	319.49443	421.7008	449.31038	231.43584	418.68458	482.63132
4	234.82769	279.47867	269.1597	207.01358	200.23679	938.98578	170.3543	178.74447	213.13301	301.18909	141.44389	280.48566	202.5767
5	38.25803	33.35752	30.90586	8.65039	40.47013	12.85209	30.34217	41.24234	39.78822	40.91114	17.45545	25.77657	63.95302
6	349.84522	322.70023	201.79583	206.51453	427.46915	86.08998	315.76797	297.34555	238.3895	474.06981	393.23011	159.28096	341.14757
7	285.482	234.09556	296.21204	177.41549	255.93998	80.51508	128.5589	162.05158	131.73862	112.18146	98.13526	136.02882	99.77928
8	7.0601	8.8235	9.26351	5.7078	8.07768	101.81117	2.80509	7.17756	10.37289	6.66185	7.78213	8.87751	9.79966
9	2040.99007	1396.83632	2609.31792	1602.61857	3439.21093	461.64206	2974.86083	1274.37809	1813.02073	2487.11274	3005.11077	1728.2382	2099.81296
10	23.41977	30.16943	21.20467	19.5387	20.34227	29.0803	13.17911	24.05361	11.87253	22.64879	20.01443	23.48217	14.978
11	96.42164	109.13311	98.4448	129.32789	61.44334	56.78574	16.08754	71.00269	37.67888	20.11389	17.94804	36.61701	11.30294
12	176.99627	196.19841	176.35126	123.97821	136.14194	220.75158	144.3272	117.40729	189.98444	193.88785	131.87632	34.02379	172.09135
13	139.60874	145.60505	198.70667	102.66702	111.37036	15.33709	67.5523	61.04084	62.87382	70.90249	46.75354	88.24655	56.10786
14	73.36135	83.82395	91.5369	67.89697	114.98327	845.0055	70.55664	86.78188	54.37254	94.94153	68.49035	94.67042	61.91721
15	522.99202	490.38114	542.62524	429.75691	399.07131	2297.99678	395.31947	365.74106	474.48505	414.39068	432.70952	424.47154	351.12367
16	418.14546	415.91625	348.29611	317.97272	435.27391	49.42276	282.0223	398.00438	252.58509	245.82473	293.01624	223.6921	368.95609
17	79.95935	81.94871	74.15046	63.30856	60.94018	136.87378	104.72651	53.8234	72.71413	80.06692	78.55303	67.47255	68.34196
18	17.06581	23.17414	22.99165	15.37069	16.97644	19.81311	25.06165	17.79112	20.58347	26.01274	19.54109	0.00929	23.48635
19	160.74894	153.42844	160.65789	142.41578	113.51045	78.81089	125.84781	111.45146	134.07944	138.88951	162.7942	134.05731	126.29684
20	52.02499	59.56398	37.0335	32.77024	65.09795	1332.23034	50.17587	21.7892	50.74615	48.38492	24.37193	31.06472	62.95385

similar gene expression profiles

Real data



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	171.31276	104.52255	69.87071	78.72994	61.90472	11.12443	66.83669	112.60186	112.48516	42.3466	43.14078	44.66914	97.61767
2	75.20531	91.39947	58.85608	50.16479	86.12535	111.46063	66.37294	40.22882	110.43174	108.47401	62.78481	59.43699	115.22141
3	150.92785	290.05855	181.05986	104.849	820.24275	33.7155	442.39206	319.49443	421.7008	449.31038	231.43584	418.68458	482.63132
4	234.82769	279.47867	269.1597	207.01358	200.23679	938.98578	170.3543	178.74447	213.13301	301.18909	141.44389	280.48566	202.5767
5	38.25803	33.35752	30.90586	8.65039	40.47013	12.85209	30.34217	41.24234	39.78822	40.91114	17.45545	25.77657	63.95302
6	349.84522	322.70023	201.79583	206.51453	427.46915	86.08998	315.76797	297.34555	238.3895	474.06981	393.23011	159.28096	341.14757
7	285.482	234.09556	296.21204	177.41549	255.93998	80.51508	128.5589	162.05158	131.73862	112.18146	98.13526	136.02882	99.77928
8	7.0601	8.8235	9.26351	5.7078	8.07768	101.81117	2.80509	7.17756	10.37289	6.66185	7.78213	8.87751	9.79966
9	2040.99007	1396.83632	2609.31792	1602.61857	3439.21093	461.64206	2974.86083	1274.37809	1813.02073	2487.11274	3005.11077	1728.2382	2099.81296
10	23.41977	30.16943	21.20467	19.5387	20.34227	29.0803	13.17911	24.05361	11.87253	22.64879	20.01443	23.48217	14.978
11	96.42164	109.13311	98.4448	129.32789	61.44334	56.78574	16.08754	71.00269	37.67888	20.11389	17.94804	36.61701	11.30294
12	176.99627	196.19841	176.35126	123.97821	136.14194	220.75158	144.3272	117.40729	189.98444	193.88785	131.87632	34.02379	172.09135
13	139.60874	145.60505	198.70667	102.66702	111.37036	15.33709	67.5523	61.04084	62.87382	70.90249	46.75354	88.24655	56.10786
14	73.36135	83.82395	91.5369	67.89697	114.98327	845.0055	70.55664	86.78188	54.37254	94.94153	68.49035	94.67042	61.91721
15	522.99202	490.38114	542.62524	429.75691	399.07131	2297.99678	395.31947	365.74106	474.48505	414.39068	432.70952	424.47154	351.12367
16	418.14546	415.91625	348.29611	317.97272	435.27391	49.42276	282.0223	398.00438	252.58509	245.82473	293.01624	223.6921	368.95609
17	79.95935	81.94871	74.15046	63.30856	60.94018	136.87378	104.72651	53.8234	72.71413	80.06692	78.55303	67.47255	68.34196
18	17.06581	23.17414	22.99165	15.37069	16.97644	19.81311	25.06165	17.79112	20.58347	26.01274	19.54109	0.00929	23.48635
19	160.74894	153.42844	160.65789	142.41578	113.51045	78.81089	125.84781	111.45146	134.07944	138.88951	162.7942	134.05731	126.29684
20	52.02499	59.56398	37.0335	32.77024	65.09795	1332.23034	50.17587	21.7892	50.74615	48.38492	24.37193	31.06472	62.95385

similar clinical symptoms

General principle

Split the data into (disjoint) groups, such that

- **similar things are in the same groups,**

and

- **non-similar things are in different groups.**

This makes sense only if we specify what we mean by **similar**.

- similar genre? media format? target audience? …

Clustering in a computer

Computers famously only do what you tell them to do.

Clustering needs a similarity, so have to provide one:

```
def sim(x,y):  # pairwise similarity between x and y
    do something that involves x and y
    return value
```

Typically, similarities are most useful if they are
- non-negative: **sim(x,y) >= 0**
- symmetric: **sim(x,y) = sim(y,x)**

Sometimes, more convenient to think of *distances* instead (high similarity <-> low distance and vice versa).

Back to the general principle

Split the data into (disjoint) groups, such that

- **similar things are in the same groups,**

and

- **non-similar things are in different groups.**

Do we really need both?

Back to the general principle

Split the data into (disjoint) groups, such that

- **similar things are in the same groups,**

and

- **non-similar things are in different groups.**

Do we really need both? **YES!**

Clustering in a computer

Computers are also famously pedantic.

Imagine you tell the computer to

Split the data into groups, such that

- **similar things are in the same groups.**

What would happen?

Clustering in a computer

Computers are also famously pedantic.

Imagine you tell the computer to

Split the data into groups, such that

- **similar things are in the same groups.**

What would happen?

It would create a single group that contains all things.

Clustering in a computer

Computers are also famously pedantic.

Imagine you tell the computer to

Split the data into groups, such that

- **non-similar things are in different groups.**

What would happen?

Clustering in a computer

Computers are also famously pedantic.

Imagine you tell the computer to

Split the data into groups, such that

- **non-similar things are in different groups.**

What would happen?

It would create as many groups as there are things, each group containing a single thing.

General principle

Now we are convinced that we need both conditions.

Split the data into (disjoint) groups, such that

- **similar things are in the same groups,**

and

- **non-similar things are in different groups.**

General principle

Now we are convinced that we need both conditions.

Split the data into (disjoint) groups, such that

- **similar things are in the same groups,**

and

- **non-similar things are in different groups.**

Punchline: **We can’t always fulfill both at the same time!**

General principle

Now we are convinced that we need both conditions.

Split the data into (disjoint) groups, such that

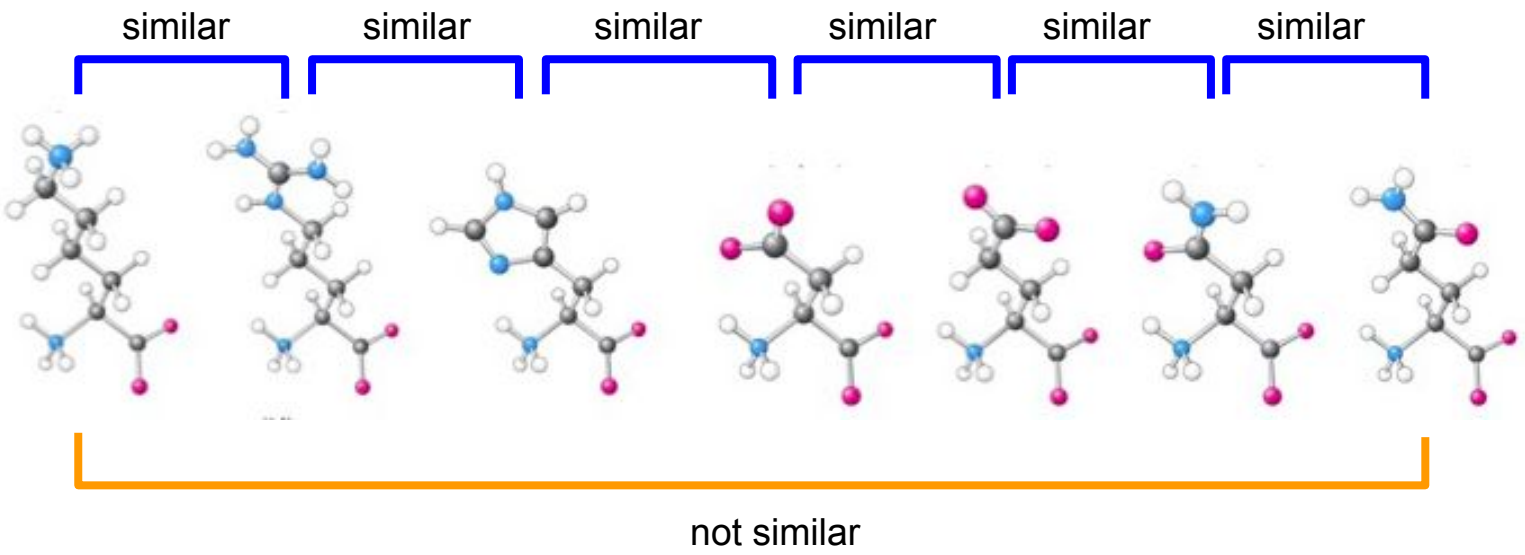
- **similar things are in the same groups,**

and

- **non-similar things are in different groups.**

Punchline: ~~**We can’t always fulfill both at the same time!**~~
For real data: **We can never fulfill both at the same time!**

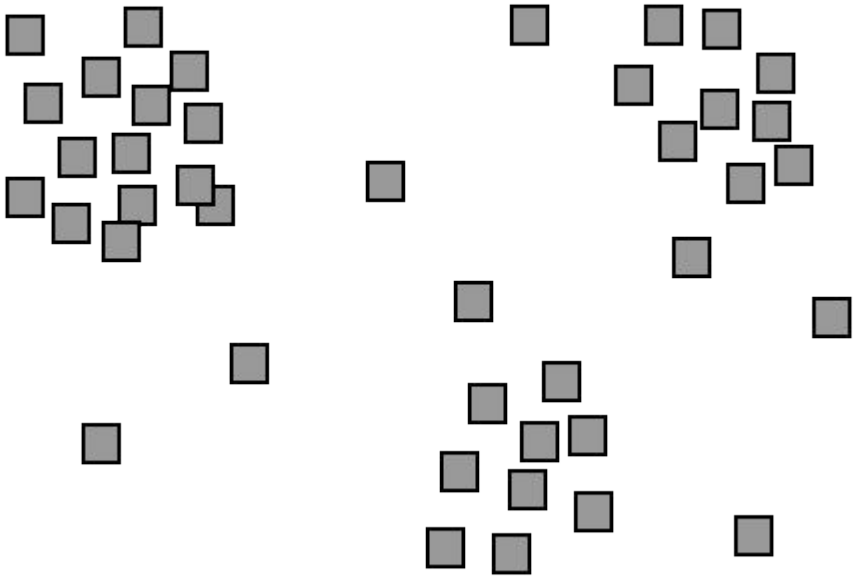
General principle



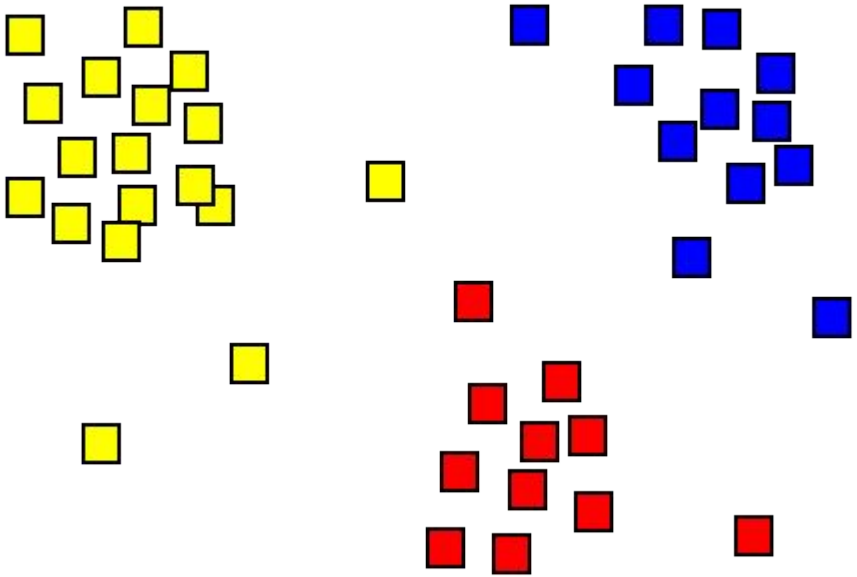
how to cluster?

Image: http://www.iop.org/

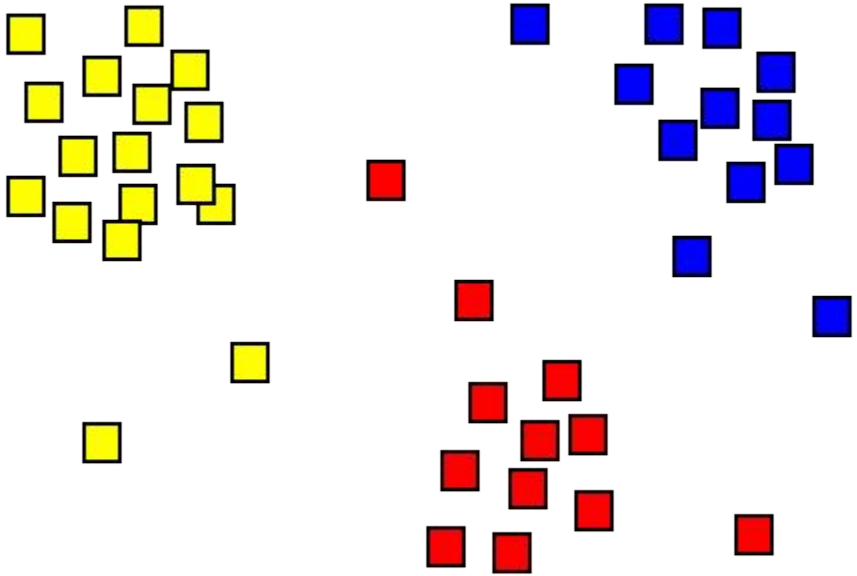
General principle



General principle



General principle



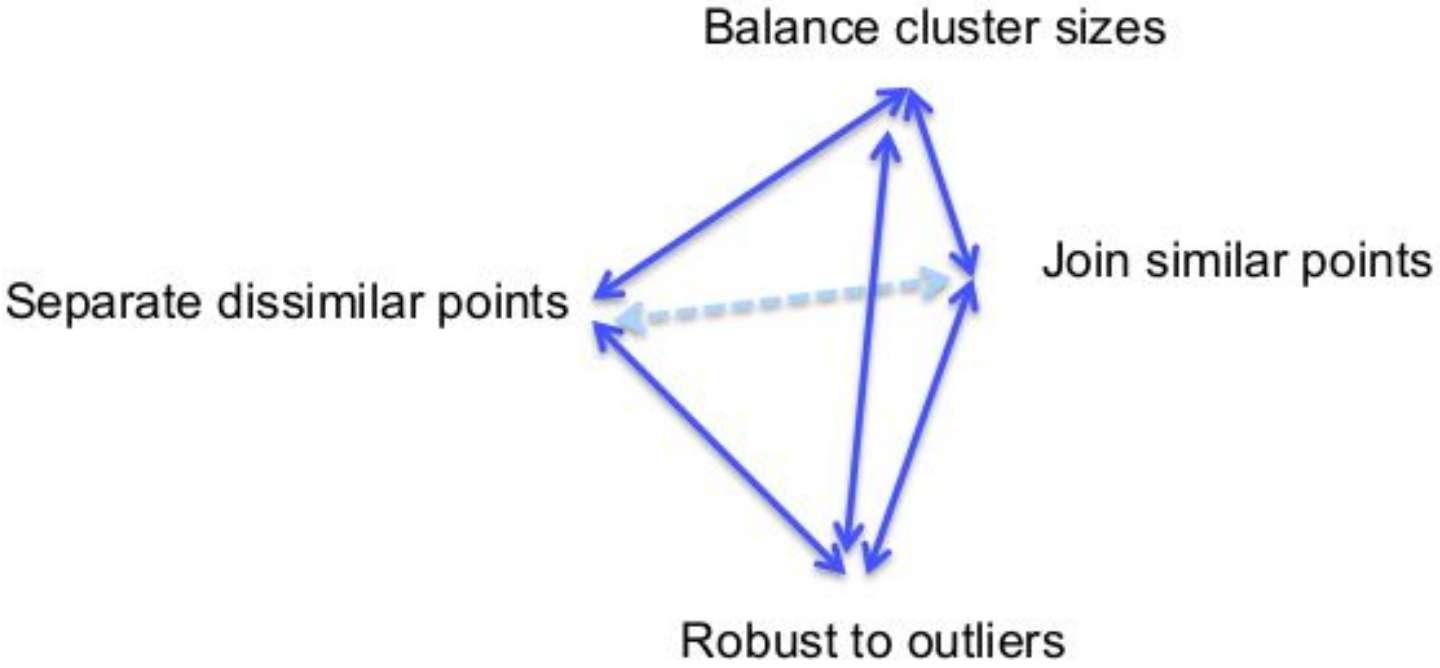
Real world clustering needs a trade-off

separate
dissimilar
things

join
similar
things

Real world clustering needs a trade-off

Really, it's a trade-off between more than two properties….



Clustering in a computer

Let’s look at some real clustering algorithms:

single linkage clustering, complete linkage clustering, average linkage clustering, spectral clustering, min-cut clustering, normalized cut clustering, biclustering, correlation clustering, k-means, k-medians, k-medoids, Gaussian mixture clustering, mean shift, medoid shift, affinity propagation, ant colony clustering

Too many…

Clustering in a computer

Let’s cluster the clustering algorithms ;-)

- Connectivity-based clustering
  - single linkage clustering, complete linkage clustering, average linkage clustering
- Graph-based clustering
  - spectral clustering, min-cut clustering, normalized cut, correlation clustering
- Centroid-based clustering
  - k-means, k-medians, k-medoids
- Distribution-based clustering
  - Gaussian mixture clustering
- Density-based clustering
  - mean shift, median shift
- Others
  - affinity propagation, ant colony clustering, biclustering

Excurse… how DO people select a clustering method?

Many reasons, typically ad hoc:

- “easy to use, no need to tune parameters”
- “available in my favorite software package”
- “freely downloadable from the web”
- “runs really fast”
- “worked for a friend of mine (on a different problem)”

Excurse… how DO people select a clustering method?

Many reasons, typically ad hoc:

- “easy to use, no need to tune parameters”
- “available in my favorite software package”
- “freely downloadable from the web”
- “runs really fast”
- “worked for a friend of mine (on a different problem)”
- “it’s called *cluster*, you only have to click on the icon, and the publication has 13000 citations”

Excurse… how DO people select a clustering method?

Many reasons, typically ad hoc:

- “easy to use, no need to tune parameters”
- “available in my favorite software package”
- “freely downloadable from the web”
- “runs really fast”
- “worked for a friend of mine (on a different problem)”
- “it’s called *cluster*, you only have to click on the icon, and the publication has 13000 citations” ← IST faculty

Excurse… how DO people select a clustering method?

Many reasons, typically ad hoc:

- “easy to use, no need to tune parameters”
- “available in my favorite software package”
- “freely downloadable from the web”
- “runs really fast”
- “worked for a friend of mine (on a different problem)”
- “it’s called *cluster*, you only have to click on the icon, and the publication has 13000 citations” ← IST faculty

We already saw: the “right” clustering depends on the task.
There is no one best clustering algorithm for everything.
(and that’s actually a mathematically provable statement)

Excurse… how DO people select a clustering method?

(example courtesy of S. Ben-David)

Imagine, your data is an **illness** you have and the clustering algorithms are the **medicine.**

Excurse… how DO people select a clustering method?

(example courtesy of S. Ben-David)

Imagine, your data is an **illness** you have and the clustering algorithms are the **medicine.**

Then

- “easy to use, no need to tune parameters”

becomes

- “take this pill, it’s really easy to swallow”

Excurse… how DO people select a clustering method?

(example courtesy of S. Ben-David)

Imagine, your data is an **illness** you have and the clustering algorithms are the **medicine.**

Then

- “available in my favorite software package”

becomes

- “take this pill, I found it in my medicine cabinet”

Excurse… how DO people select a clustering method?

(example courtesy of S. Ben-David)

Imagine, your data is an **illness** you have and the clustering algorithms are the **medicine.**

Then

- “freely downloadable from the web”

becomes

- “take this pill, I got it as a free sample”

Excurse… how DO people select a clustering method?

(example courtesy of S. Ben-David)

Imagine, your data is an **illness** you have and the clustering algorithms are the **medicine.**

Then

- “worked for a friend of mine (on a different problem)”

becomes

- “take this pill, it worked for a friend of mine (when he had a different illness)”

Excurse… how DO people select a clustering method?

(example courtesy of S. Ben-David)

Imagine, your data is an **illness** you have and the clustering algorithms are the **medicine**.

Then

- “it’s called *cluster*, you only have to click on the icon, and the publication has 13000 citations”

becomes

- “take this pill, it’s call *medicine*, it’s easy to take out of the box, and its inventor got rich and famous.”

How SHOULD you select a clustering method?

1) find out what properties you want

2) select a method with appropriate properties

How SHOULD you select a clustering method?

1) find out what properties you want

2) select a method with appropriate properties

For that, of course you have to know:

1) what you want ← I can’t help you with that.

2) which method has which properties ← the rest of the lecture

Clustering in a computer

Different groups of clustering algorithms:

- Connectivity-based clustering
  - single linkage clustering, complete linkage clustering, average linkage clustering
- Graph-based clustering
  - spectral clustering, min-cut clustering, normalized cut, correlation clustering
- Centroid-based clustering
  - k-means, k-medians, k-medoids
- Distribution-based clustering
  - Gaussian mixture clustering
- Density-based clustering
  - mean shift, median shift
- Others
  - affinity propagation, ant colony clustering, biclustering

Connectivity-based/Agglomerative Clustering

A family of algorithms using the same strategy:

- initially: each data point is a cluster of its own
- repeat
  - identify the two clusters that are most similar to each other and merge them
- until only 1 cluster is left

Connectivity-based/Agglomerative Clustering

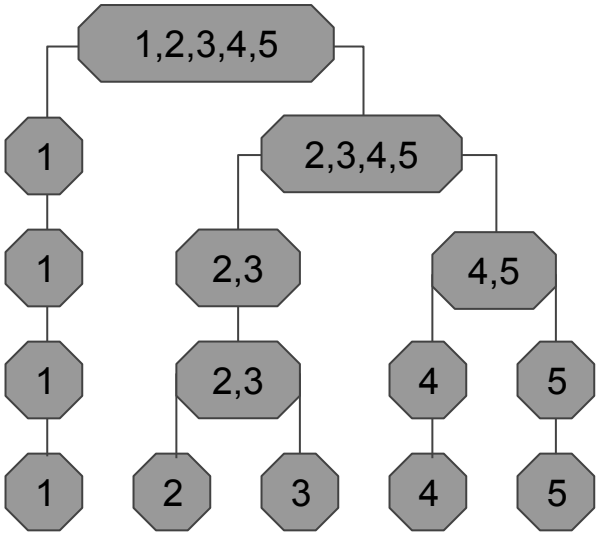
A family of algorithms using the same strategy:

- initially: each data point is a cluster of its own
- repeat
  - identify the two clusters that are most similar to each other and merge them
- until only 1 cluster is left

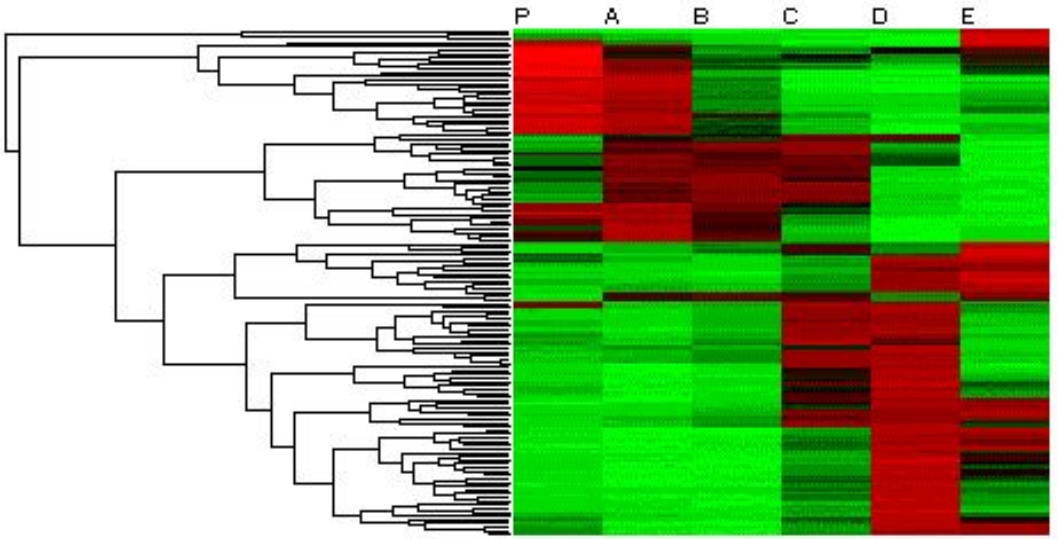
What is the result?

- not just one clustering but a sequence of clusterings with fewer and fewer clusters → a hierarchy
- the user must decide which level is appropriate

Connectivity-based/Agglomerative Clustering



schematics



hierarchical clustering of expression data

Connectivity-based/Agglomerative Clustering

Family of algorithms using the same strategy:

- initially: each data point is a cluster of its own
- repeat
  - identify the two clusters that are most similar to each other and merge them
- until only 1 cluster is left

What do we need?

- not just a way to measure similarity between objects, but a similarity between clusters

Connectivity-based/Agglomerative Clustering

Different choice leads to very different algorithms:

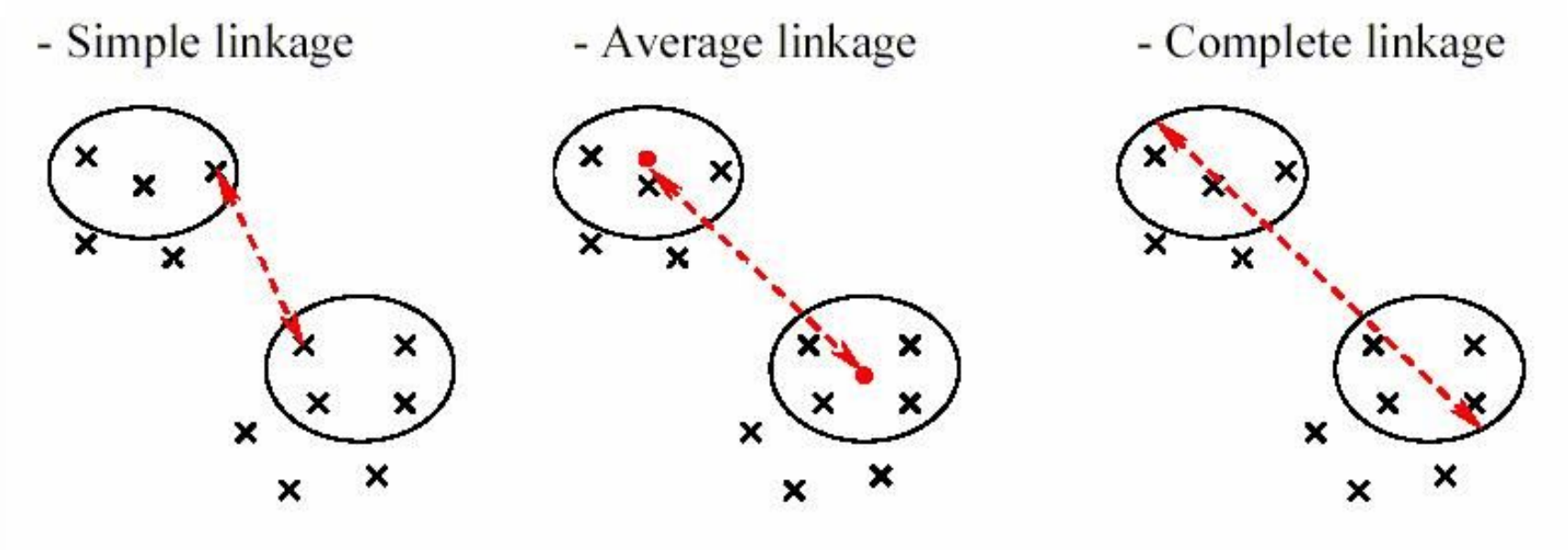
“single linkage”: similarity between clusters is the highest similarity between any of their elements

“average linkage”: similarity between clusters is the average similarity between any of their elements

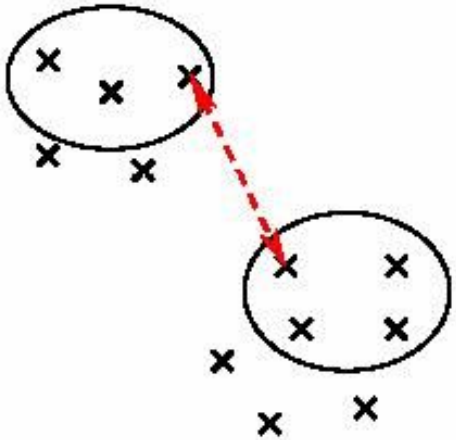
“complete linkage”: similarity between clusters is the smallest similarity between any of their elements

Connectivity-based Clustering

Different choice leads to very different algorithms:



- Simple linkage

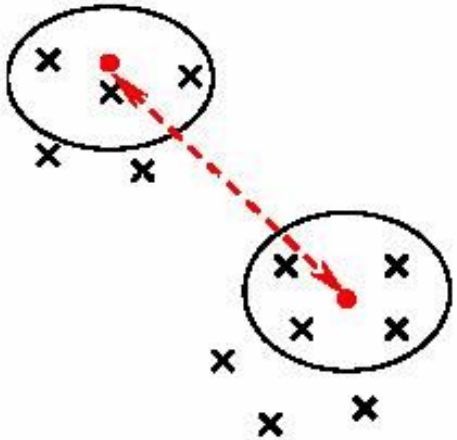


+ similar things are groups together

- dissimilar things are separated

- balanced clusters

- Average linkage

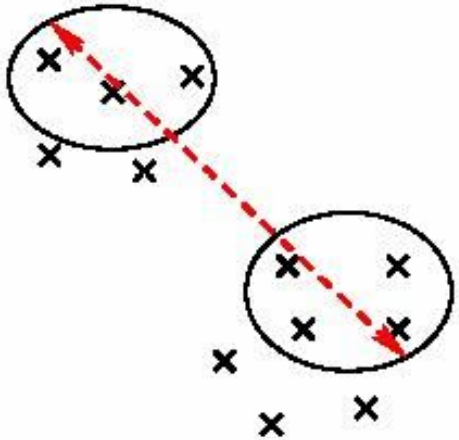


~ similar things are groups together

~ dissimilar things are separated

+ balanced clusters

- Complete linkage



- similar things are groups together

+ dissimilar things are separated

- balanced clusters

Connectivity-based/Agglomerative Clustering

Family of algorithms using the same strategy:

- initially: each data point is a cluster of its own
- repeat
  - identify the two clusters that are most similar to each other and merge them
- until only 1 cluster is left

What don’t you get?

- the number of clusters
- representatives for each cluster

Centroid-based clustering

Family of algorithms using the same strategy:

- identify a set of “cluster centers”
- assign each data point to its most similar center

What would you prefer as a “cluster centers”?

- only original data points
  - k-medoid or vector quantization
- linear combinations of data points (for vector data)
  - k-means

k-medoid clustering

Parameter: number of clusters you want, *k*

1) randomly pick k data points as cluster centers
2) repeat
   - assign each points to its closest cluster center
   - within each cluster, identify the point with highest average similarity to all others
3) until the assignments do not change anymore

Results: *k* cluster centers (that are original data points)
assignments of each data point to a cluster

k-means clustering

Parameter: number of clusters you want, *k*

1) choose as many data points as necessary
2) repeat
   - assign each points to its closest cluster center
   - within each cluster, compute a new cluster center as the average vector of all points
3) until the assignments do not change anymore

Results: *k* cluster center vectors (not original data points)
assignments of each data point to a cluster

k-medoid, k-means clustering

- similar things are groups together

+ dissimilar things are separated

+ balanced cluster sizes

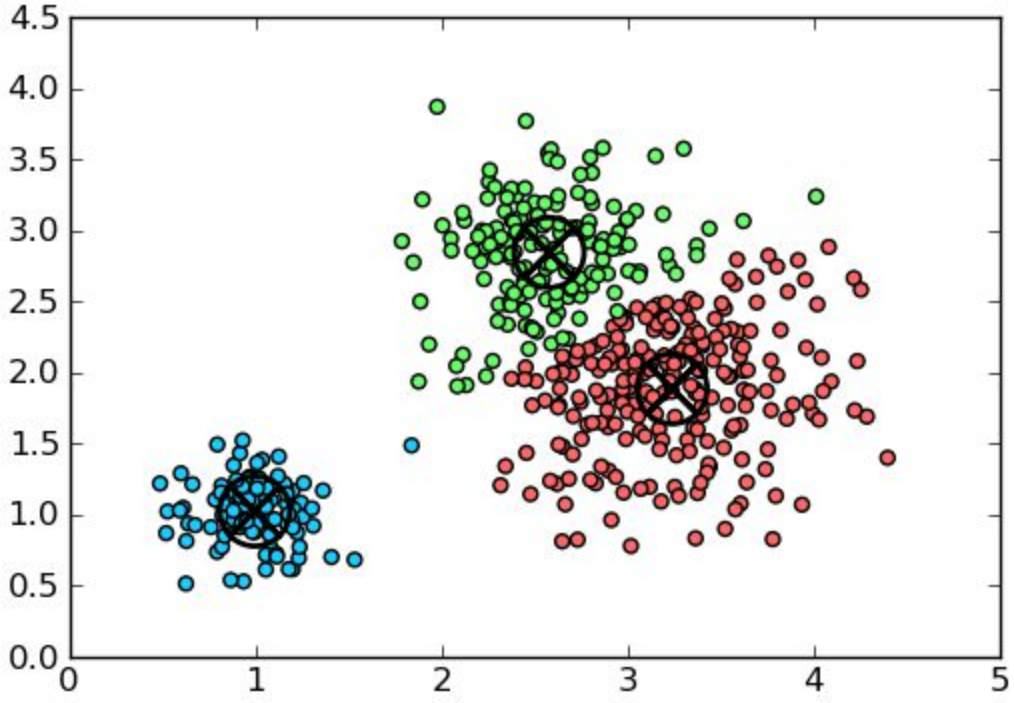


Image: http://blog.mpacula.com/2011/04/27/k-means-clustering-example-python/

Learned vector quantization

Parameter: max.distance of points to their cluster center, *d*
1) select best subset data points as cluster centers such that
   - for each data points a center lies at distance at most *d*
2) assign each data points to its closest

How to select? Integer-Linear Optimization problem (ILP)

Results: some number of cluster centers (data points)
assignments of each data point to a cluster

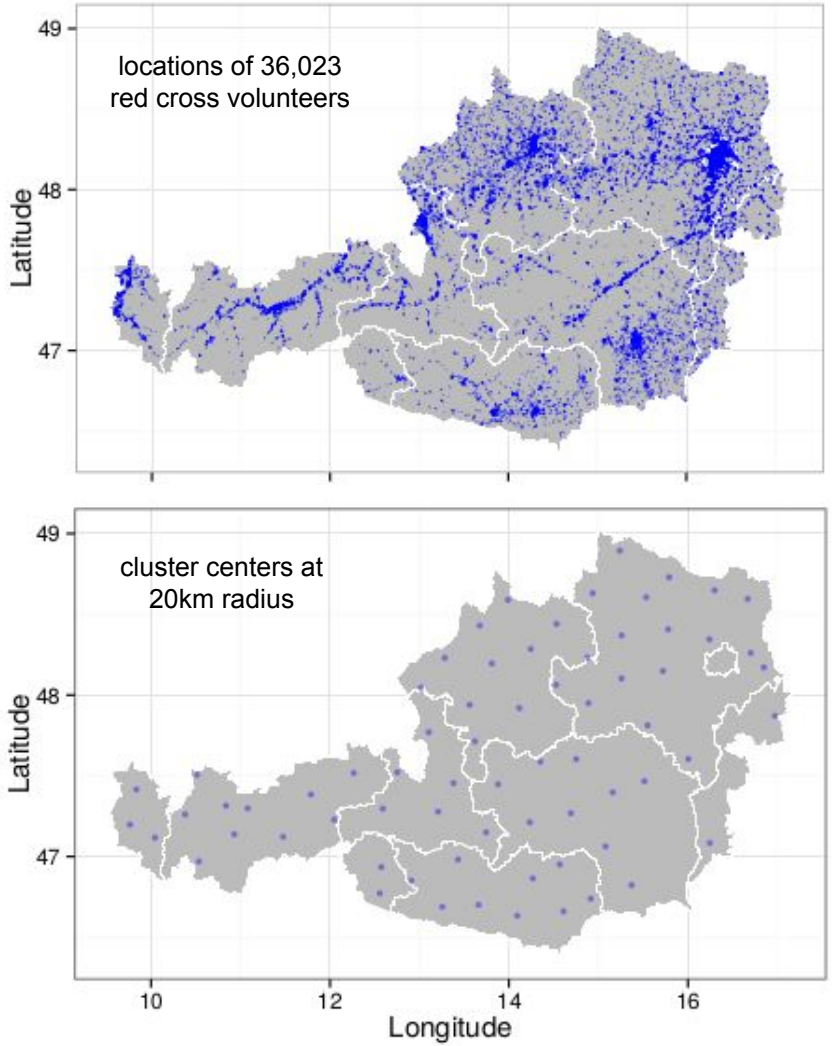
Learned vector quantization

- similar things are groups together

+ dissimilar things are separated

- balanced cluster sizes

+ balanced cluster radii



Learned vector quantization

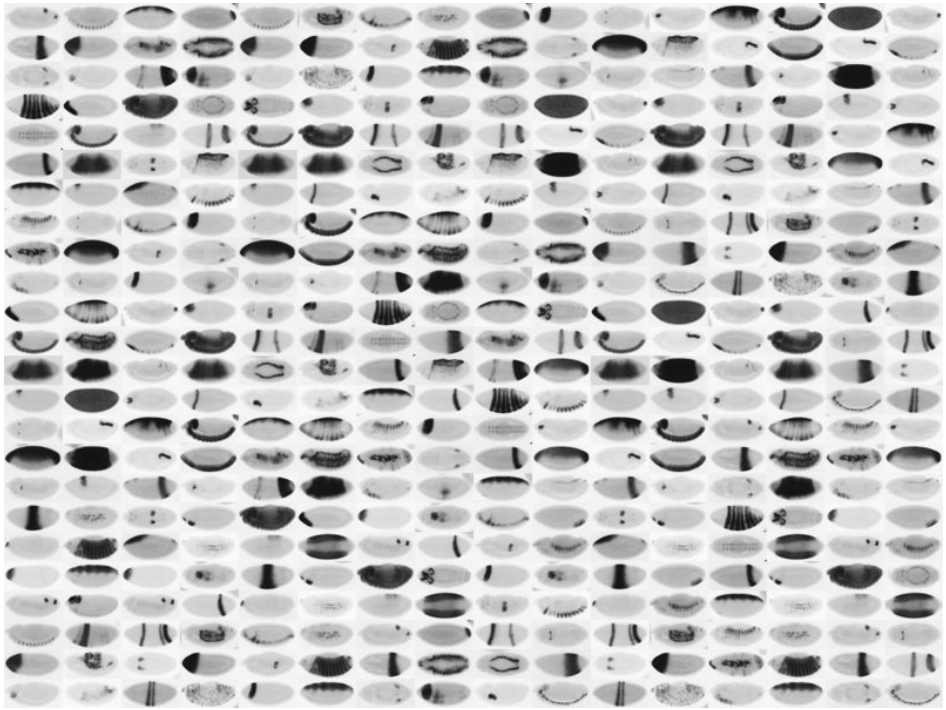
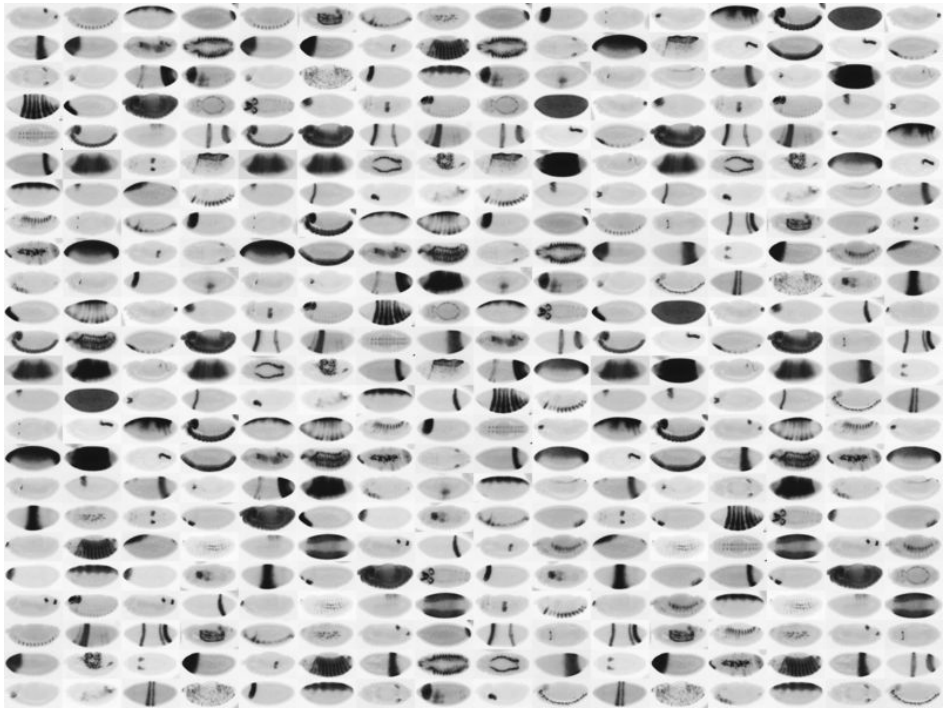
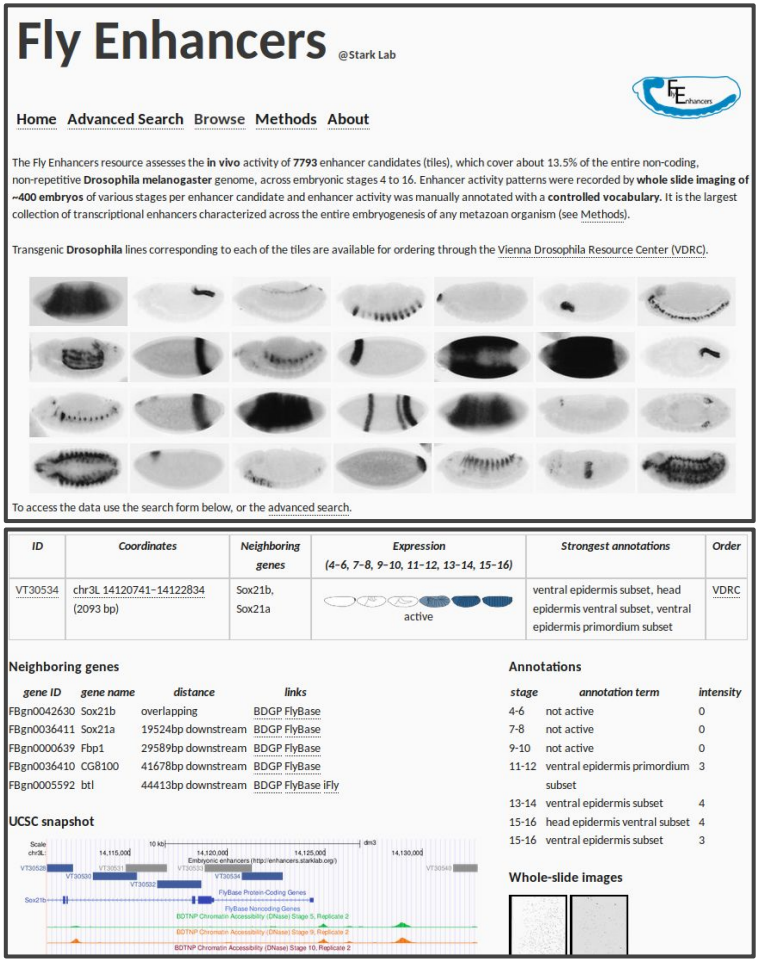


Image: Kvon EZ, Kazmar T, Stampfel G*, Yáñez-Cuna JO*, Pagani M, Schernhuber K, Dickson BJ, Stark A. Genome-scale functional characterization of Drosophila developmental enhancers in vivo. Nature, 2014.

Learned vector quantization





Fly Enhancers @Stark Lab

Home Advanced Search Browse Methods About

The Fly Enhancers resource assesses the **in vivo** activity of **7793** enhancer candidates (tiles), which cover about 13.5% of the entire non-coding, non-repetitive **Drosophila melanogaster** genome, across embryonic stages 4 to 16. Enhancer activity patterns were recorded by **whole slide imaging of ~400 embryos** of various stages per enhancer candidate and enhancer activity was manually annotated with a **controlled vocabulary.** It is the largest collection of transcriptional enhancers characterized across the entire embryogenesis of any metazoan organism (see Methods).

Transgenic **Drosophila** lines corresponding to each of the tiles are available for ordering through the Vienna Drosophila Resource Center (VDRC).

To access the data use the search form below, or the advanced search.

ID	Coordinates	Neighboring genes	Expression (4-6, 7-8, 9-10, 11-12, 13-14, 15-16)	Strongest annotations	Order
VT30534	chr3L 14120741-14122834 (2093 bp)	Sox21b, Sox21a	active	ventral epidermis subset, head epidermis ventral subset, ventral epidermis primordium subset	VDRC

Neighboring genes

gene ID	gene name	distance	links
FBgn0042630	Sox21b	overlapping	BDGP FlyBase
FBgn0036411	Sox21a	19524bp downstream	BDGP FlyBase
FBgn0000639	Fbp1	29589bp downstream	BDGP FlyBase
FBgn0036410	CG8100	41678bp downstream	BDGP FlyBase
FBgn0005592	btl	44413bp downstream	BDGP FlyBase iFly

Annotations

stage	annotation term	intensity
4-6	not active	0
7-8	not active	0
9-10	not active	0
11-12	ventral epidermis primordium subset	3
13-14	ventral epidermis subset	4
15-16	head epidermis ventral subset	4
15-16	ventral epidermis subset	3

UCSC snapshot

Whole-slide images

Image: Kvon EZ, Kazmar T, Stampfel G*, Yáñez-Cuna JO*, Pagani M, Schernhuber K, Dickson BJ, Stark A. Genome-scale functional characterization of Drosophila developmental enhancers in vivo. Nature, 2014.

Affinity Propagation

- create two matrices: **R** “responsibility”, **A** “availability”
  - *r(i,k)*: how well-suited is datapoint *i* to represent the datapoint *k*
  - a(i,k): how appropriate would it be for data *i* to represent data *k*
- initialize

$$r(i,k) \leftarrow s(i,k) - \max_{k' \neq k} \{a(i,k') + s(i,k')\}$$

- repeat

$$a(i,k) \leftarrow \min\left(0, r(k,k) + \sum_{i' \notin \{i,k\}} \max(0, r(i',k))\right) \text{ for } i \neq k \text{ and}$$

$$a(k,k) \leftarrow \sum_{i' \neq k} \max(0, r(i',k)).$$

- until no more changed, or a fixed number of times

Affinity Propagation

What properties will the clusters have?

Affinity Propagation

What properties will
the clusters have? no clue

?? similar things are
groups together

?? dissimilar things
are separated

?? balanced cluster sizes

Affinity Propagation

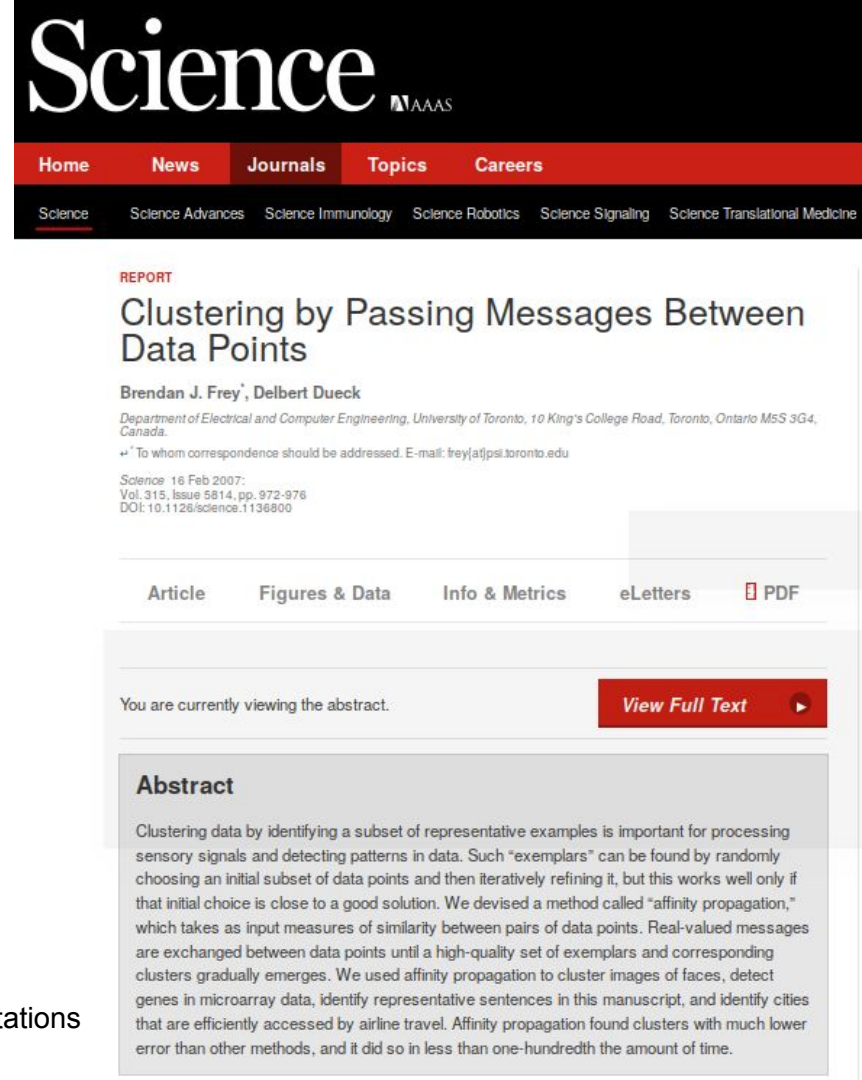
What properties will the clusters have? no clue

?? similar things are groups together

?? dissimilar things are separated

?? balanced cluster sizes

~4800 citations



Science AAAS

Home News Journals Topics Careers

Science Science Advances Science Immunology Science Robotics Science Signaling Science Translational Medicine

REPORT

Clustering by Passing Messages Between Data Points

Brendan J. Frey*, Delbert Dueck

Department of Electrical and Computer Engineering, University of Toronto, 10 King's College Road, Toronto, Ontario M5S 3G4, Canada.

* To whom correspondence should be addressed. E-mail: frey{at}psi.toronto.edu

Science 16 Feb 2007:
Vol. 315, Issue 5814, pp. 972-976
DOI: 10.1126/science.1136800

Article Figures & Data Info & Metrics eLetters PDF

You are currently viewing the abstract. View Full Text

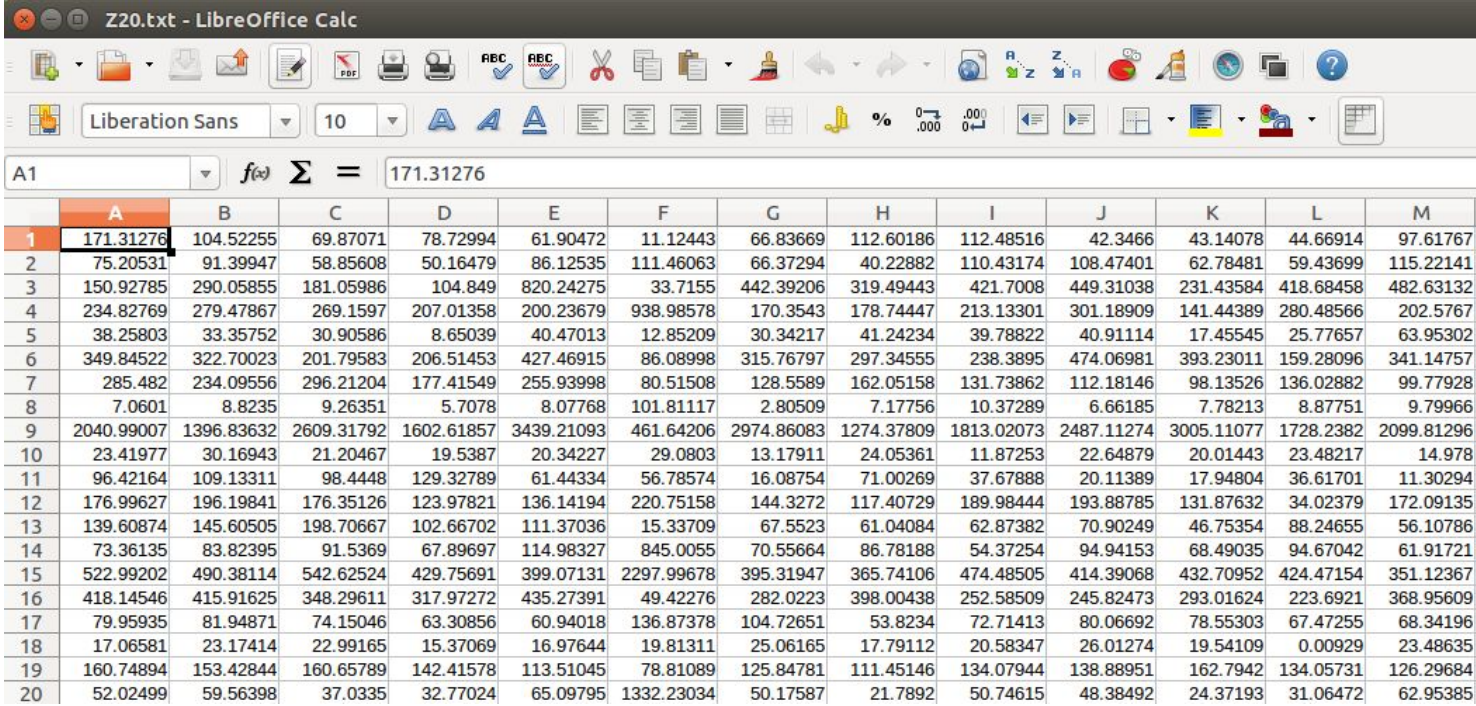
Abstract

Clustering data by identifying a subset of representative examples is important for processing sensory signals and detecting patterns in data. Such "exemplars" can be found by randomly choosing an initial subset of data points and then iteratively refining it, but this works well only if that initial choice is close to a good solution. We devised a method called "affinity propagation," which takes as input measures of similarity between pairs of data points. Real-valued messages are exchanged between data points until a high-quality set of exemplars and corresponding clusters gradually emerges. We used affinity propagation to cluster images of faces, detect genes in microarray data, identify representative sentences in this manuscript, and identify cities that are efficiently accessed by airline travel. Affinity propagation found clusters with much lower error than other methods, and it did so in less than one-hundredth the amount of time.

Ant Colony Optimization Clustering

Bonus track: Bi-Clustering

Real data



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	171.31276	104.52255	69.87071	78.72994	61.90472	11.12443	66.83669	112.60186	112.48516	42.3466	43.14078	44.66914	97.61767
2	75.20531	91.39947	58.85608	50.16479	86.12535	111.46063	66.37294	40.22882	110.43174	108.47401	62.78481	59.43699	115.22141
3	150.92785	290.05855	181.05986	104.849	820.24275	33.7155	442.39206	319.49443	421.7008	449.31038	231.43584	418.68458	482.63132
4	234.82769	279.47867	269.1597	207.01358	200.23679	938.98578	170.3543	178.74447	213.13301	301.18909	141.44389	280.48566	202.5767
5	38.25803	33.35752	30.90586	8.65039	40.47013	12.85209	30.34217	41.24234	39.78822	40.91114	17.45545	25.77657	63.95302
6	349.84522	322.70023	201.79583	206.51453	427.46915	86.08998	315.76797	297.34555	238.3895	474.06981	393.23011	159.28096	341.14757
7	285.482	234.09556	296.21204	177.41549	255.93998	80.51508	128.5589	162.05158	131.73862	112.18146	98.13526	136.02882	99.77928
8	7.0601	8.8235	9.26351	5.7078	8.07768	101.81117	2.80509	7.17756	10.37289	6.66185	7.78213	8.87751	9.79966
9	2040.99007	1396.83632	2609.31792	1602.61857	3439.21093	461.64206	2974.86083	1274.37809	1813.02073	2487.11274	3005.11077	1728.2382	2099.81296
10	23.41977	30.16943	21.20467	19.5387	20.34227	29.0803	13.17911	24.05361	11.87253	22.64879	20.01443	23.48217	14.978
11	96.42164	109.13311	98.4448	129.32789	61.44334	56.78574	16.08754	71.00269	37.67888	20.11389	17.94804	36.61701	11.30294
12	176.99627	196.19841	176.35126	123.97821	136.14194	220.75158	144.3272	117.40729	189.98444	193.88785	131.87632	34.02379	172.09135
13	139.60874	145.60505	198.70667	102.66702	111.37036	15.33709	67.5523	61.04084	62.87382	70.90249	46.75354	88.24655	56.10786
14	73.36135	83.82395	91.5369	67.89697	114.98327	845.0055	70.55664	86.78188	54.37254	94.94153	68.49035	94.67042	61.91721
15	522.99202	490.38114	542.62524	429.75691	399.07131	2297.99678	395.31947	365.74106	474.48505	414.39068	432.70952	424.47154	351.12367
16	418.14546	415.91625	348.29611	317.97272	435.27391	49.42276	282.0223	398.00438	252.58509	245.82473	293.01624	223.6921	368.95609
17	79.95935	81.94871	74.15046	63.30856	60.94018	136.87378	104.72651	53.8234	72.71413	80.06692	78.55303	67.47255	68.34196
18	17.06581	23.17414	22.99165	15.37069	16.97644	19.81311	25.06165	17.79112	20.58347	26.01274	19.54109	0.00929	23.48635
19	160.74894	153.42844	160.65789	142.41578	113.51045	78.81089	125.84781	111.45146	134.07944	138.88951	162.7942	134.05731	126.29684
20	52.02499	59.56398	37.0335	32.77024	65.09795	1332.23034	50.17587	21.7892	50.74615	48.38492	24.37193	31.06472	62.95385

What, if we think that there are clusters in the data, but only with respect to **some aspect of the measurements**? For example:

- rows 1-5 are similar to each other with respect to columns A-E
- rows 11-19 are similar to each other with respect to column H-M

Real data

very common for gene expression data under different conditions:

- under some conditions, some gene are co-expressed
- under other conditions, other genes are co-expressed

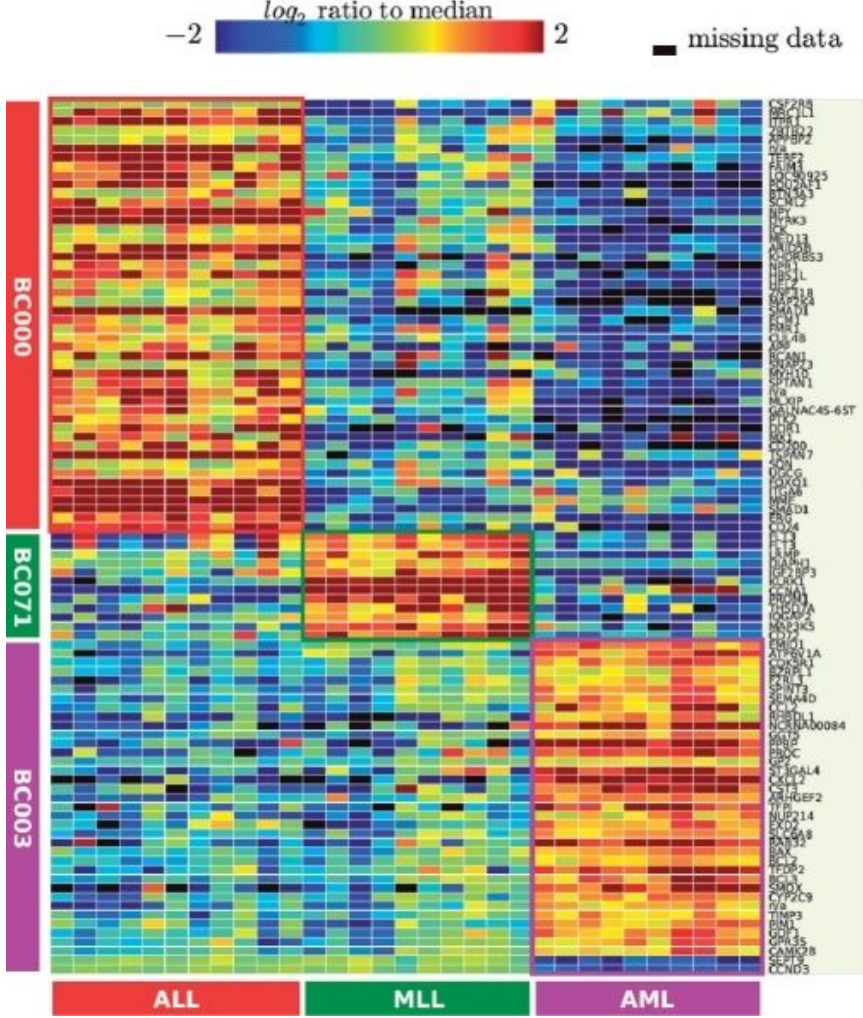
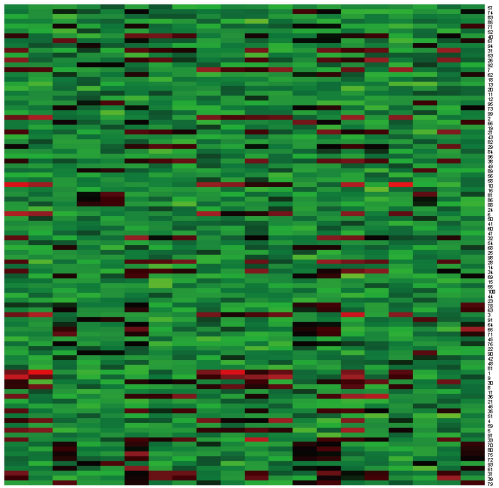


Image: Li G, Ma Q, Tang H, Paterson AH, Xu Y, “QUBIC: a qualitative biclustering algorithm for analyses of gene expression data.” Nucleic Acids Res. (2009)

Biclustering

Given: a matrix of similarity values

Goal: identify subsets of rows that are similar to each other with respect some subset of columns (or vice versa)

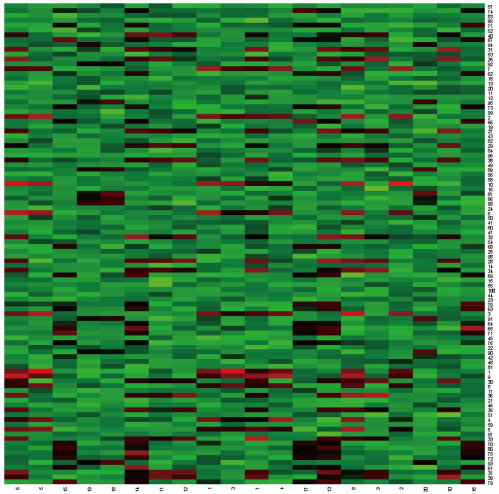


original data matrix

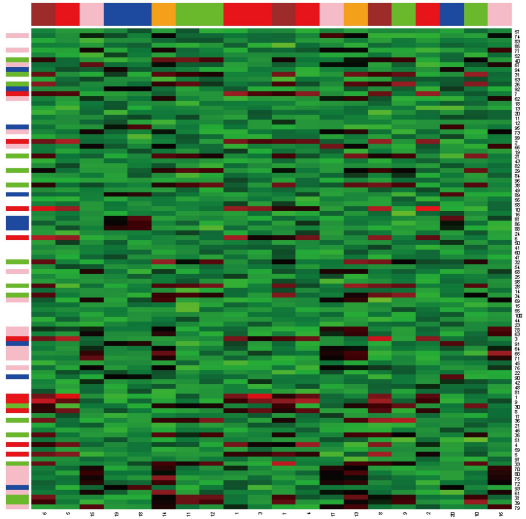
Biclustering

Given: a matrix of similarity values

Goal: identify subsets of rows that are similar to each other with respect some subset of columns (or vice versa)



original data matrix

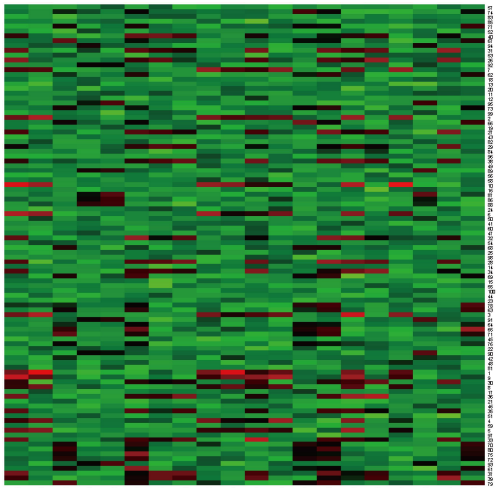


identified row/column groups

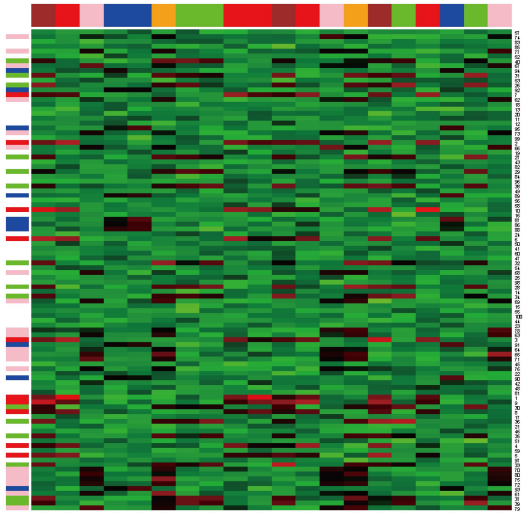
Biclustering

Given: a matrix of similarity values

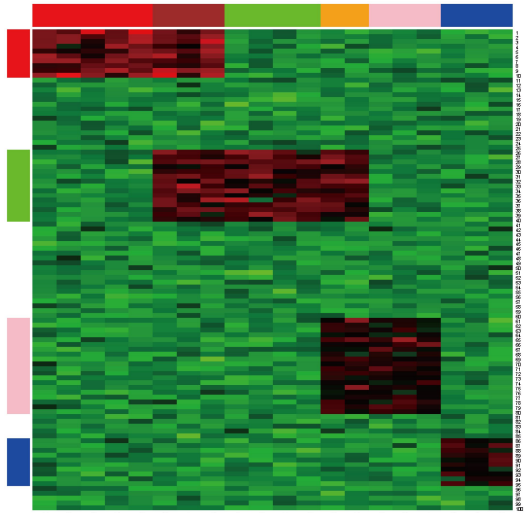
Goal: identify subsets of rows that are similar to each other with respect some subset of columns (or vice versa)



original data matrix



identified row/column groups



reshuffled matrix for visualization

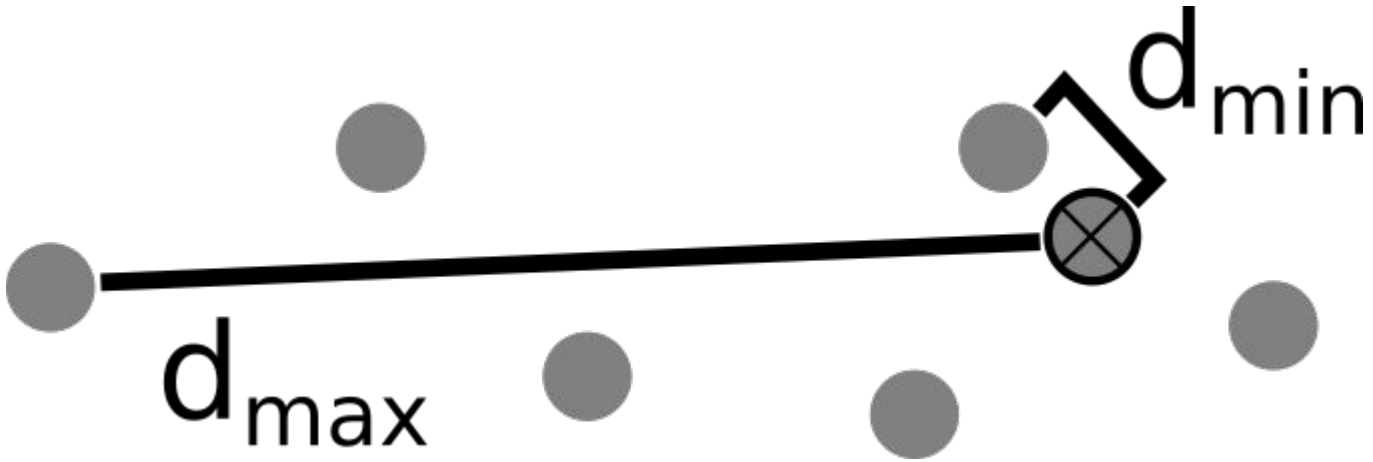
Bonus track: Dimensionality Reduction

Fact 1) Real data is often high-dimensional.

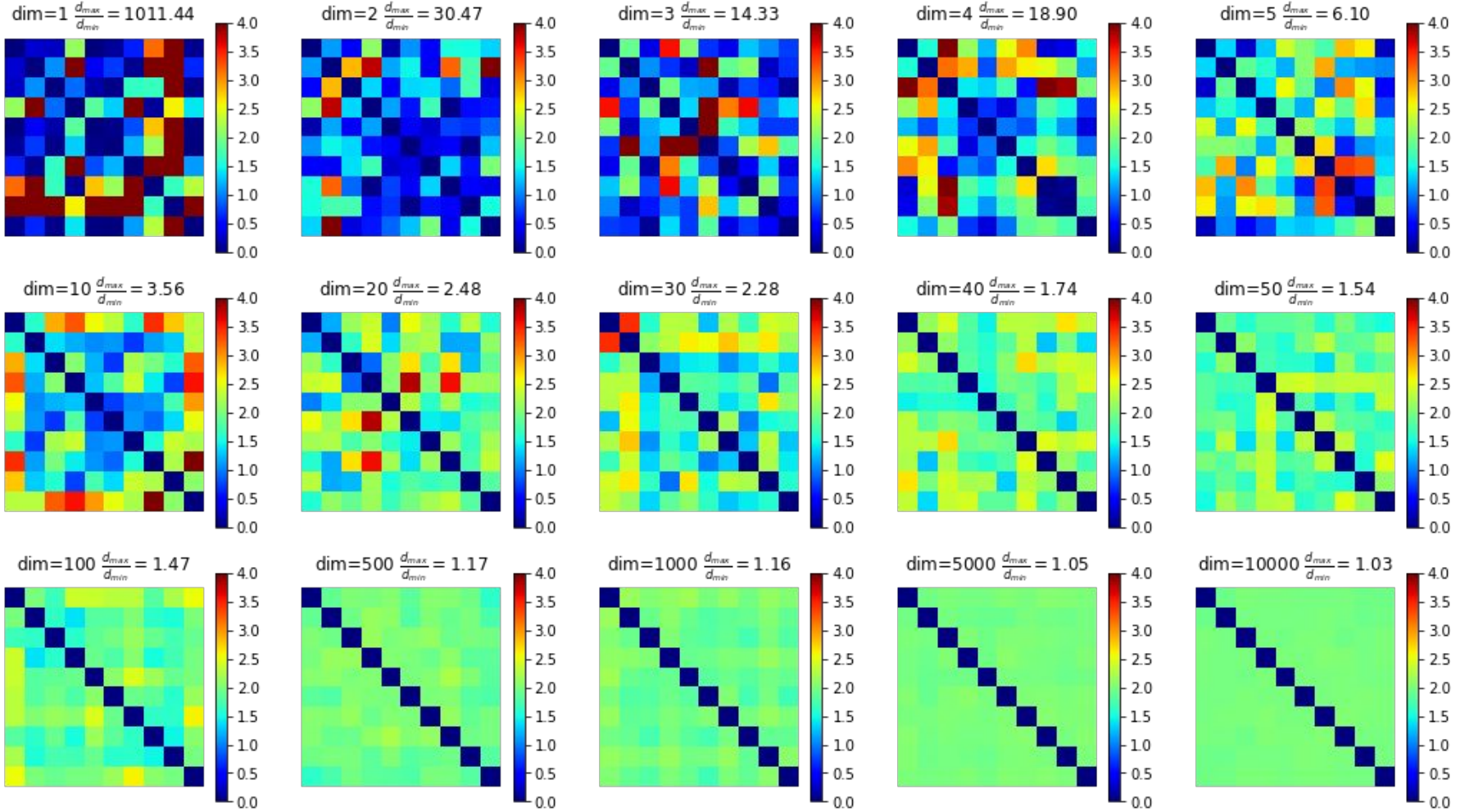
Fact 2) Distances in high dimensions behave in a weird way.

Fact 1) Real data is often high-dimensional.

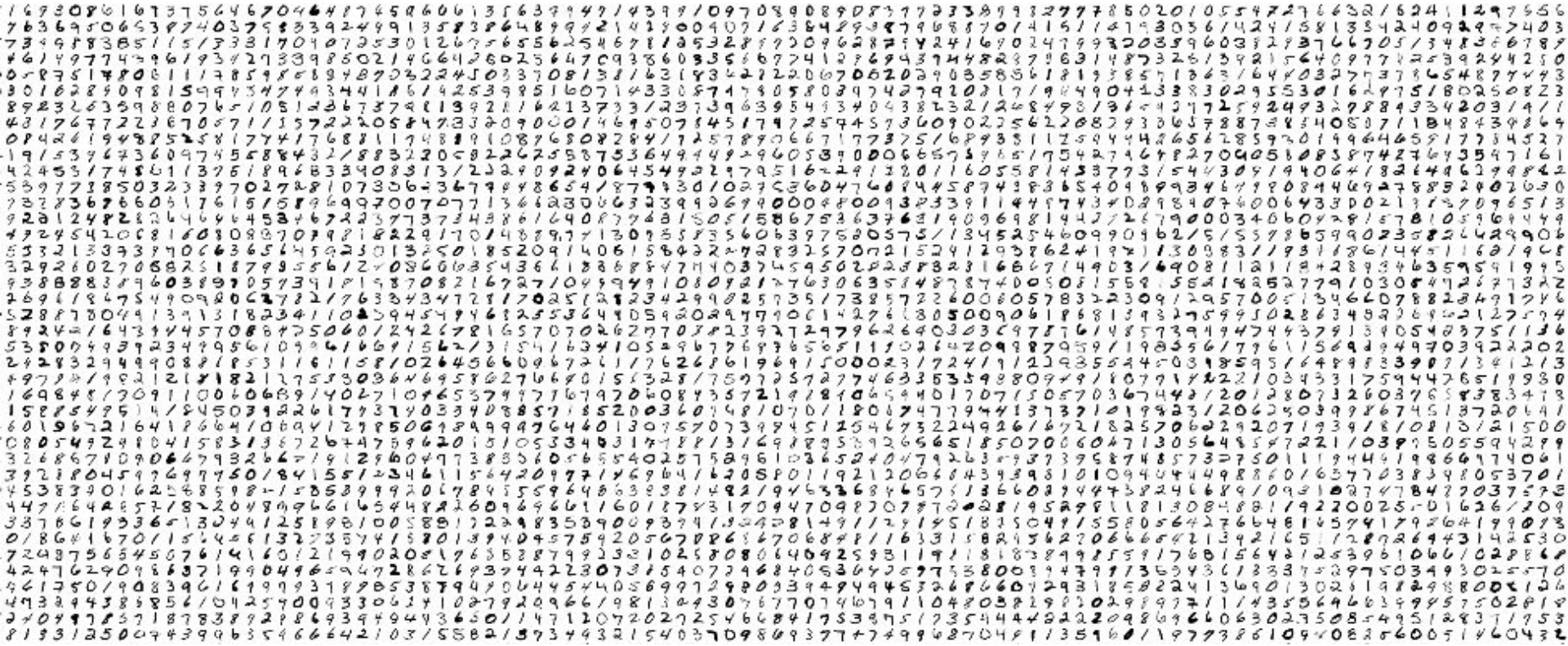
Fact 2) Distances in high dimensions behave in a weird way.



What's d_{max}/d_{min} for typical data?


dim=1 $\frac{d_{max}}{d_{min}} = 1011.44$
dim=2 $\frac{d_{max}}{d_{min}} = 30.47$
dim=3 $\frac{d_{max}}{d_{min}} = 14.33$
dim=4 $\frac{d_{max}}{d_{min}} = 18.90$
dim=5 $\frac{d_{max}}{d_{min}} = 6.10$
dim=10 $\frac{d_{max}}{d_{min}} = 3.56$
dim=20 $\frac{d_{max}}{d_{min}} = 2.48$
dim=30 $\frac{d_{max}}{d_{min}} = 2.28$
dim=40 $\frac{d_{max}}{d_{min}} = 1.74$
dim=50 $\frac{d_{max}}{d_{min}} = 1.54$
dim=100 $\frac{d_{max}}{d_{min}} = 1.47$
dim=500 $\frac{d_{max}}{d_{min}} = 1.17$
dim=1000 $\frac{d_{max}}{d_{min}} = 1.16$
dim=5000 $\frac{d_{max}}{d_{min}} = 1.05$
dim=10000 $\frac{d_{max}}{d_{min}} = 1.03$
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0

Idea: first reduce dimensionality of data, e.g. with **t-SNE**



Demo: http://colah.github.io/posts/2014-10-Visualizing-MNIST/

Spike sorting: which spikes belong to individual neurons?

1) use t-SNE to transform spikes from 12 to 2 dimensions

2) draw clusters by hand

3) (ideally) each cluster corresponds to one cell

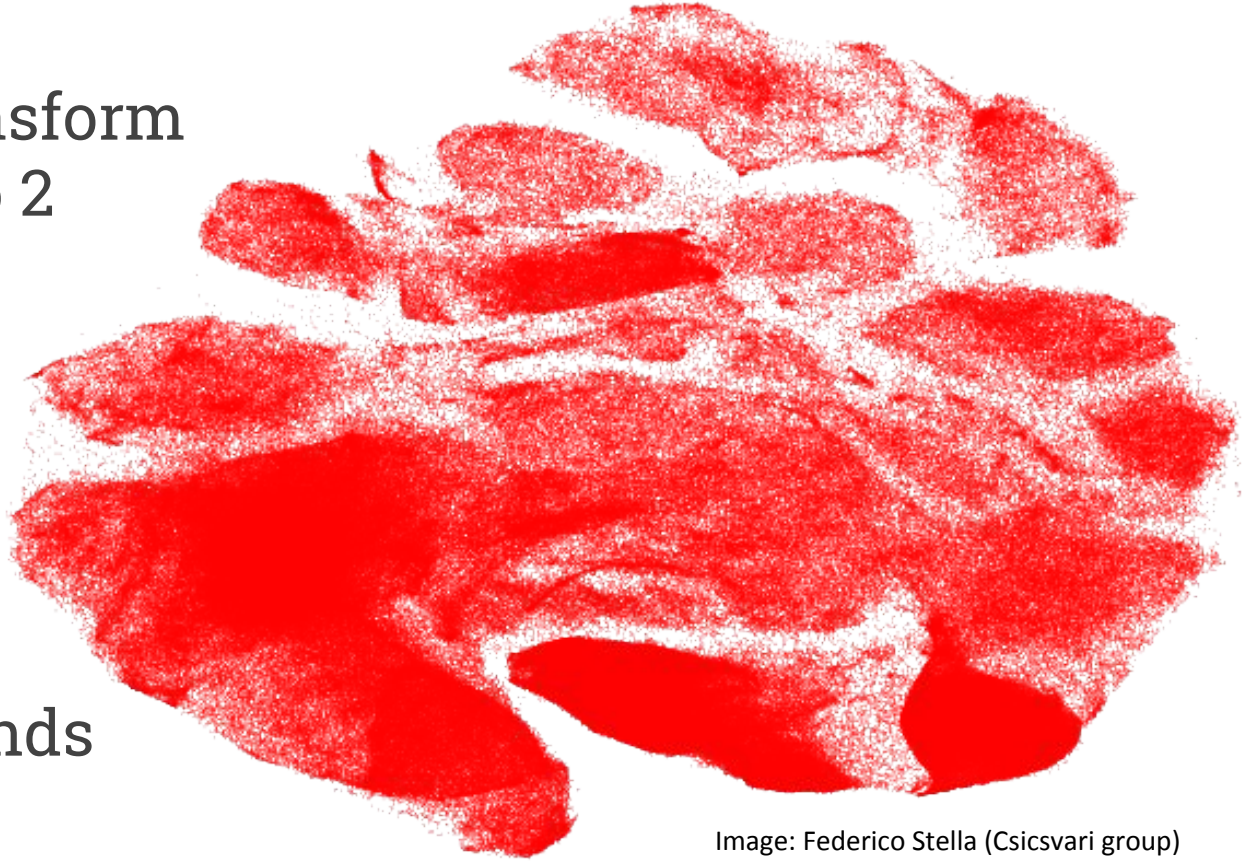


Image: Federico Stella (Csicsvari group)

Summary

- Clustering is a way to find structure in data.
  - Put similar things together, dissimilar things apart.
- Clustering is fundamentally task-dependent.
  - There are many methods with different characteristics.
  - There is no absolute “right” or “wrong” clustering.

What we didn’t speak about...

- How many clusters to use?
  - This question is so fundamental that it turned into a running gag.
- How efficient are these algorithms?