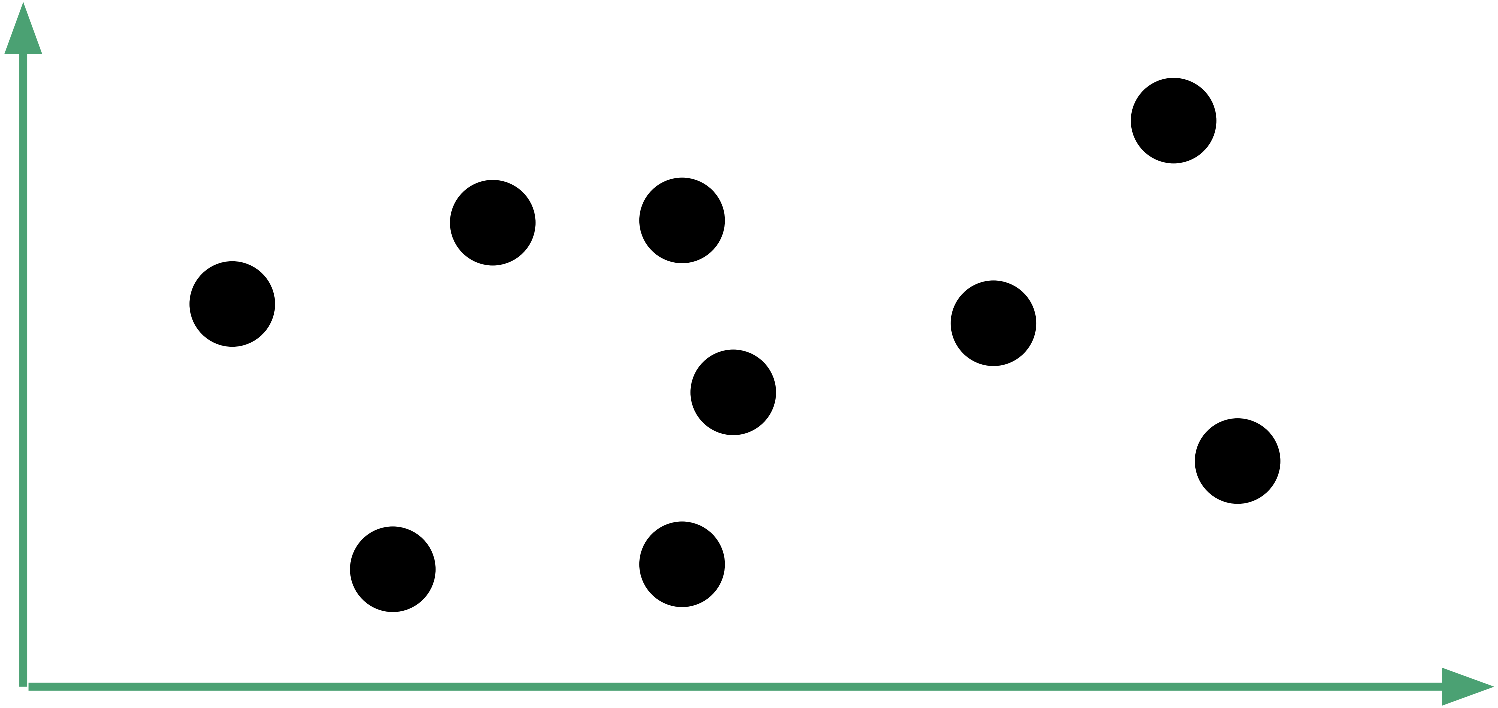
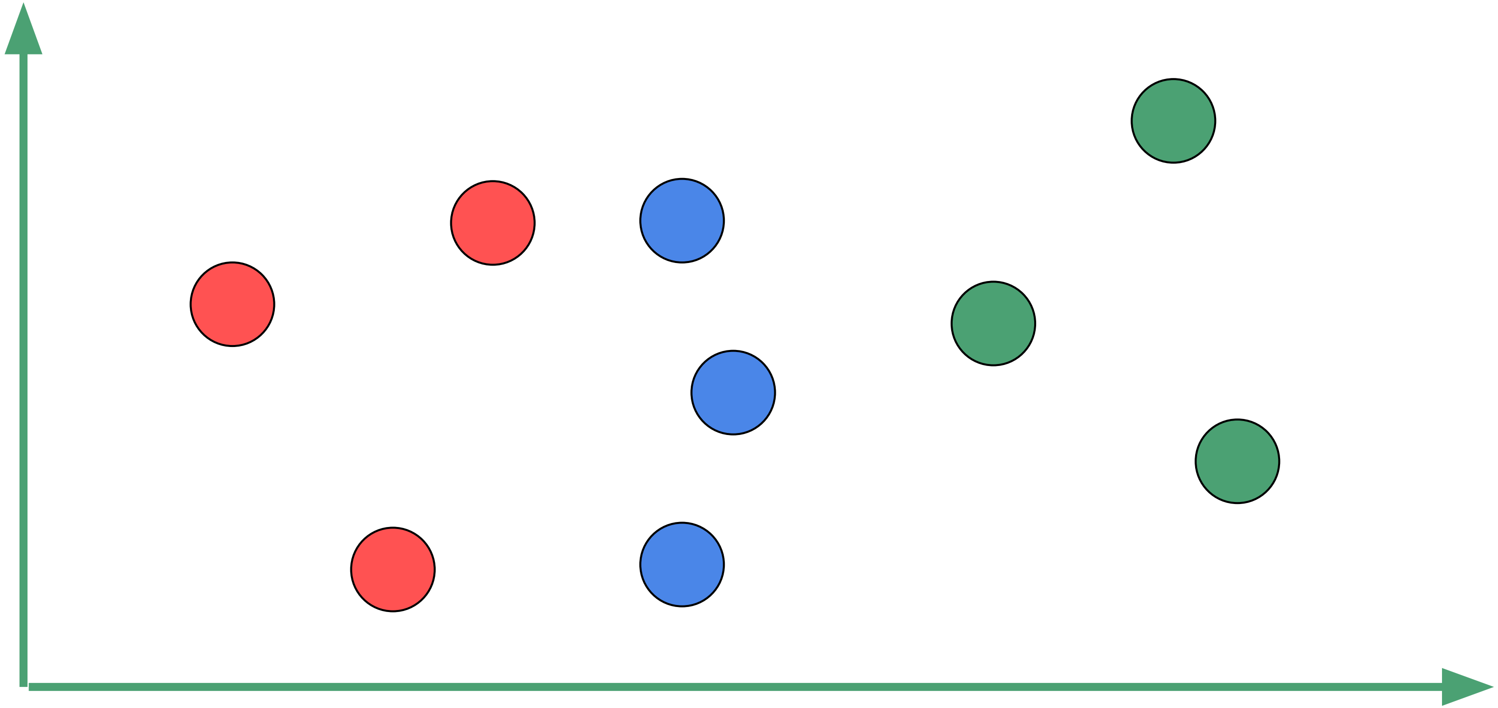
FoCS:
Clustering

Niema Moshiri
UC San Diego SPIS 2019

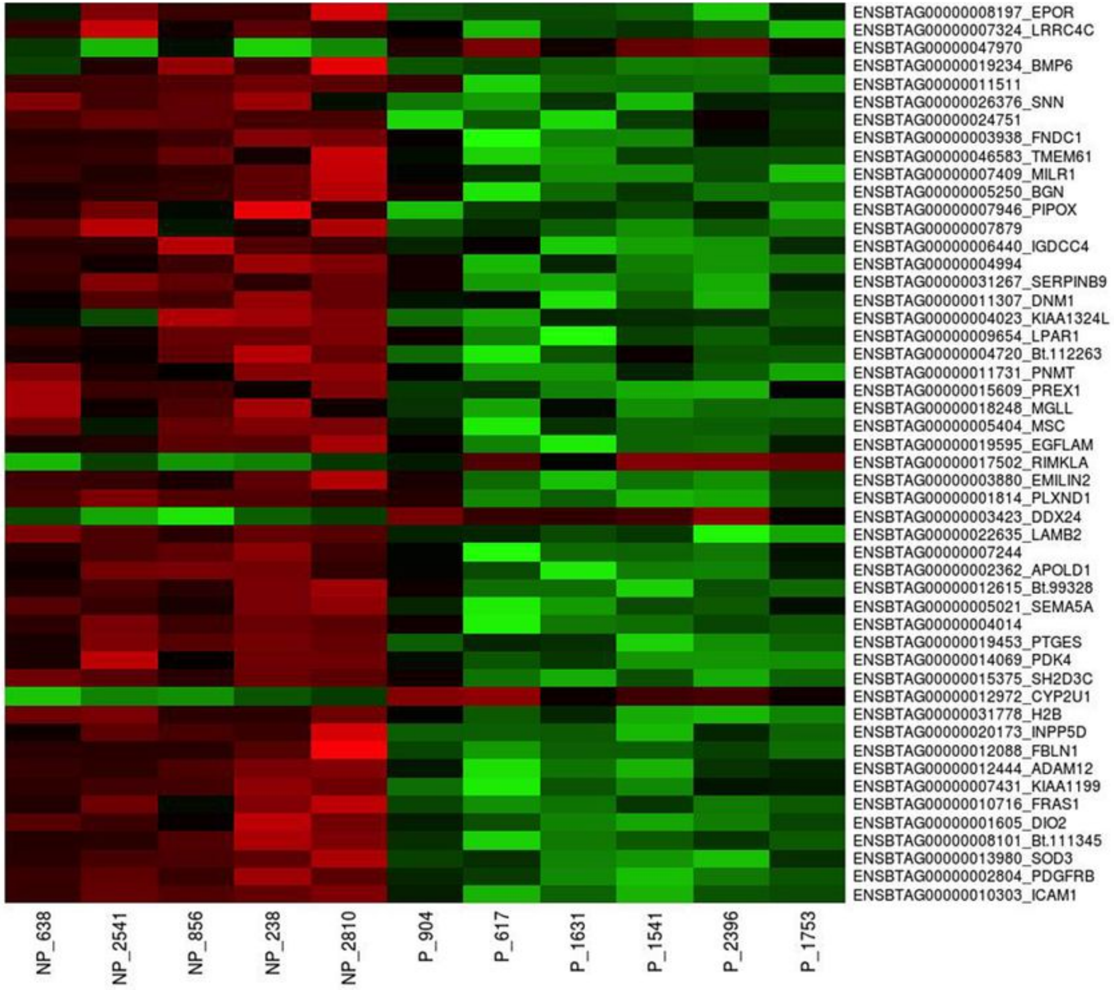
Clustering



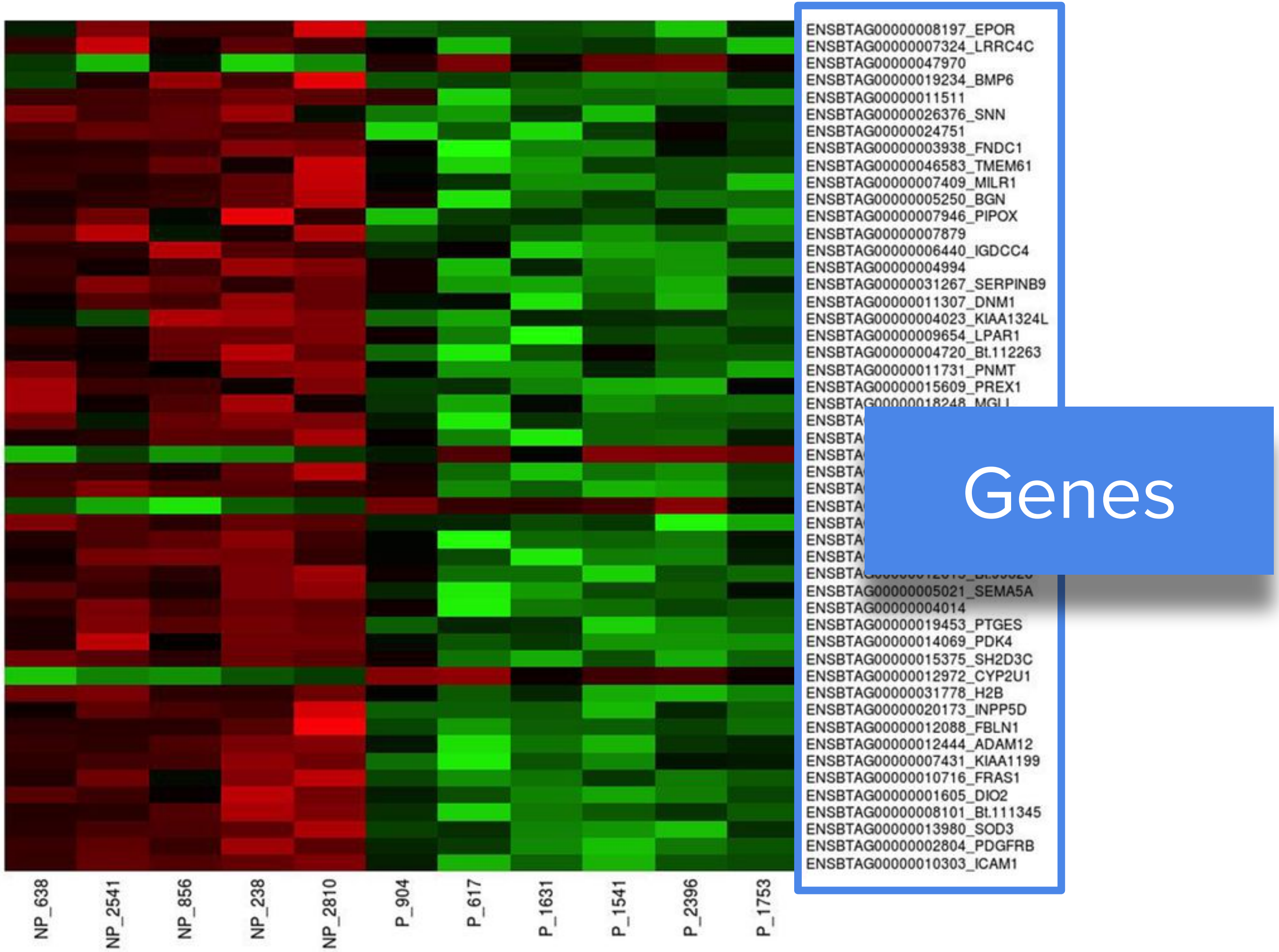
Clustering



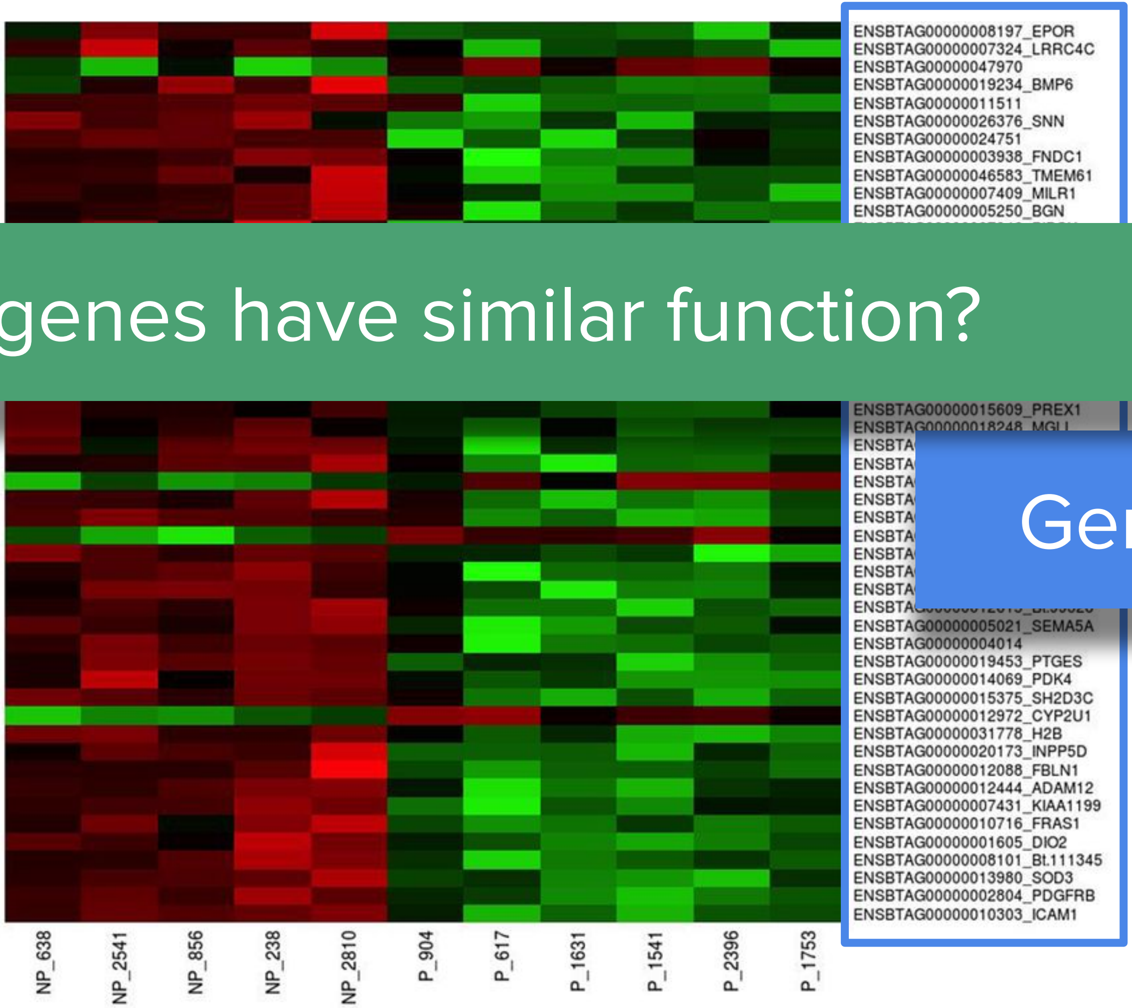
Clustering


ENSBTAG00000008197_EPOR
ENSBTAG00000007324_LRRC4C
ENSBTAG00000047970
ENSBTAG00000019234_BMP6
ENSBTAG00000011511
ENSBTAG00000026376_SNN
ENSBTAG00000024751
ENSBTAG00000003938_FNDC1
ENSBTAG00000046583_TMEM61
ENSBTAG00000007409_MILR1
ENSBTAG00000005250_BGN
ENSBTAG00000007946_PIPOX
ENSBTAG00000007879
ENSBTAG00000006440_IGDCC4
ENSBTAG00000004994
ENSBTAG00000031267_SERPINB9
ENSBTAG00000011307_DNM1
ENSBTAG00000004023_KIAA1324L
ENSBTAG00000009654_LPAR1
ENSBTAG00000004720_Bt.112263
ENSBTAG00000011731_PNMT
ENSBTAG00000015609_PREX1
ENSBTAG00000018248_MGLL
ENSBTAG00000005404_MSC
ENSBTAG00000019595_EGFLAM
ENSBTAG00000017502_RIMKLA
ENSBTAG00000003880_EMILIN2
ENSBTAG00000001814_PLXND1
ENSBTAG00000003423_DDX24
ENSBTAG00000022635_LAMB2
ENSBTAG00000007244
ENSBTAG00000002362_APOLD1
ENSBTAG00000012615_Bt.99328
ENSBTAG00000005021_SEMA5A
ENSBTAG00000004014
ENSBTAG00000019453_PTGES
ENSBTAG00000014069_PDK4
ENSBTAG00000015375_SH2D3C
ENSBTAG00000012972_CYP2U1
ENSBTAG00000031778_H2B
ENSBTAG00000020173_INPP5D
ENSBTAG00000012088_FBLN1
ENSBTAG00000012444_ADAM12
ENSBTAG00000007431_KIAA1199
ENSBTAG00000010716_FRAS1
ENSBTAG00000001605_DIO2
ENSBTAG00000008101_Bt.111345
ENSBTAG00000013980_SOD3
ENSBTAG00000002804_PDGFRB
ENSBTAG00000010303_ICAM1
NP_638
NP_2541
NP_856
NP_238
NP_2810
P_904
P_617
P_1631
P_1541
P_2396
P_1753

Clustering


ENSBTAG00000008197_EPOR
ENSBTAG00000007324_LRRC4C
ENSBTAG00000047970
ENSBTAG00000019234_BMP6
ENSBTAG00000011511
ENSBTAG00000026376_SNN
ENSBTAG00000024751
ENSBTAG00000003938_FNDC1
ENSBTAG00000046583_TMEM61
ENSBTAG00000007409_MILR1
ENSBTAG00000005250_BGN
ENSBTAG00000007946_PIPOX
ENSBTAG00000007879
ENSBTAG00000006440_IGDCC4
ENSBTAG00000004994
ENSBTAG00000031267_SERPINB9
ENSBTAG00000011307_DNM1
ENSBTAG00000004023_KIAA1324L
ENSBTAG00000009654_LPAR1
ENSBTAG00000004720_Bt.112263
ENSBTAG00000011731_PNMT
ENSBTAG00000015609_PREX1
ENSBTAG00000005021_SEMA5A
ENSBTAG00000004014
ENSBTAG00000019453_PTGES
ENSBTAG00000014069_PDK4
ENSBTAG00000015375_SH2D3C
ENSBTAG00000012972_CYP2U1
ENSBTAG00000031778_H2B
ENSBTAG00000020173_INPP5D
ENSBTAG00000012088_FBLN1
ENSBTAG00000012444_ADAM12
ENSBTAG00000007431_KIAA1199
ENSBTAG00000010716_FRAS1
ENSBTAG00000001605_DIO2
ENSBTAG00000008101_Bt.111345
ENSBTAG00000013980_SOD3
ENSBTAG00000002804_PDGFRB
ENSBTAG00000010303_ICAM1
Genes
NP_638
NP_2541
NP_856
NP_238
NP_2810
P_904
P_617
P_1631
P_1541
P_2396
P_1753

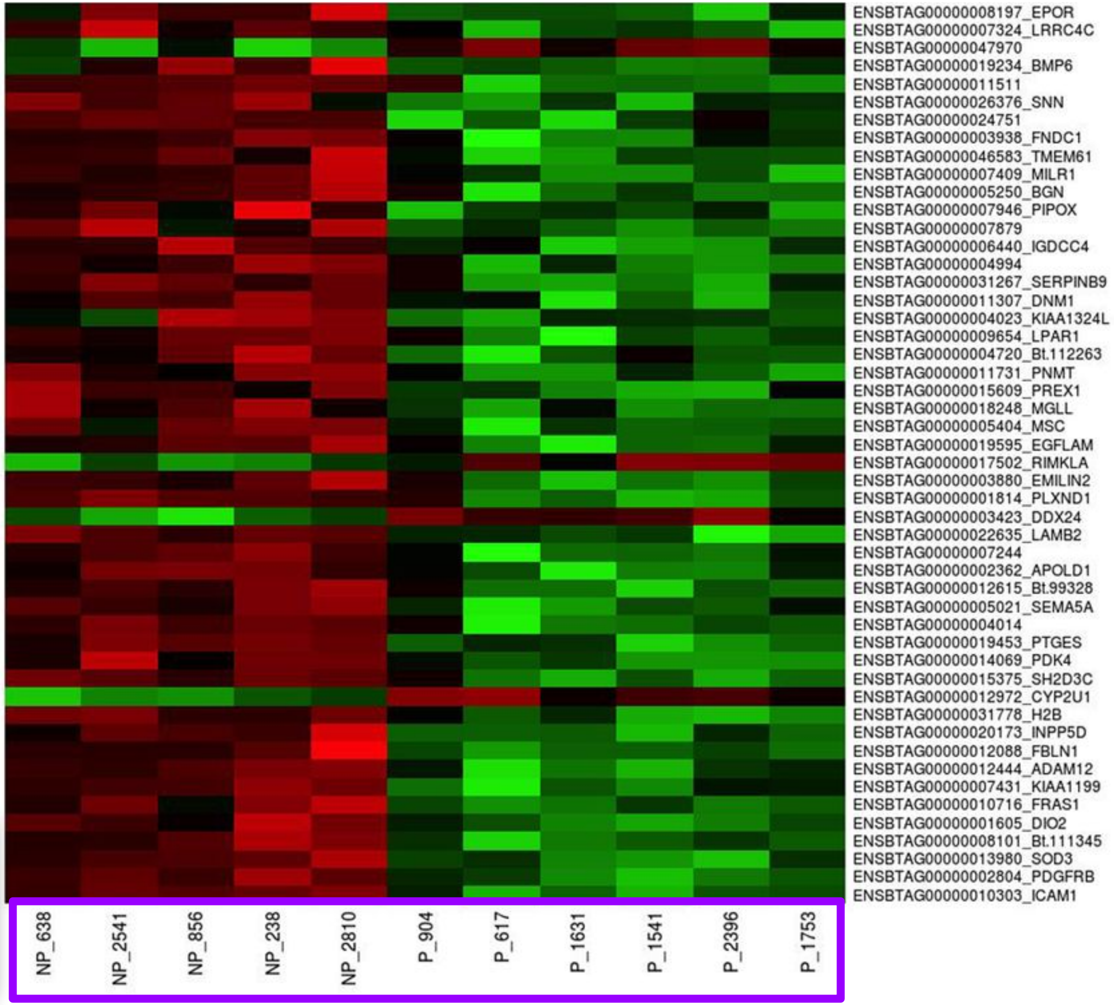
Clustering



What genes have similar function?

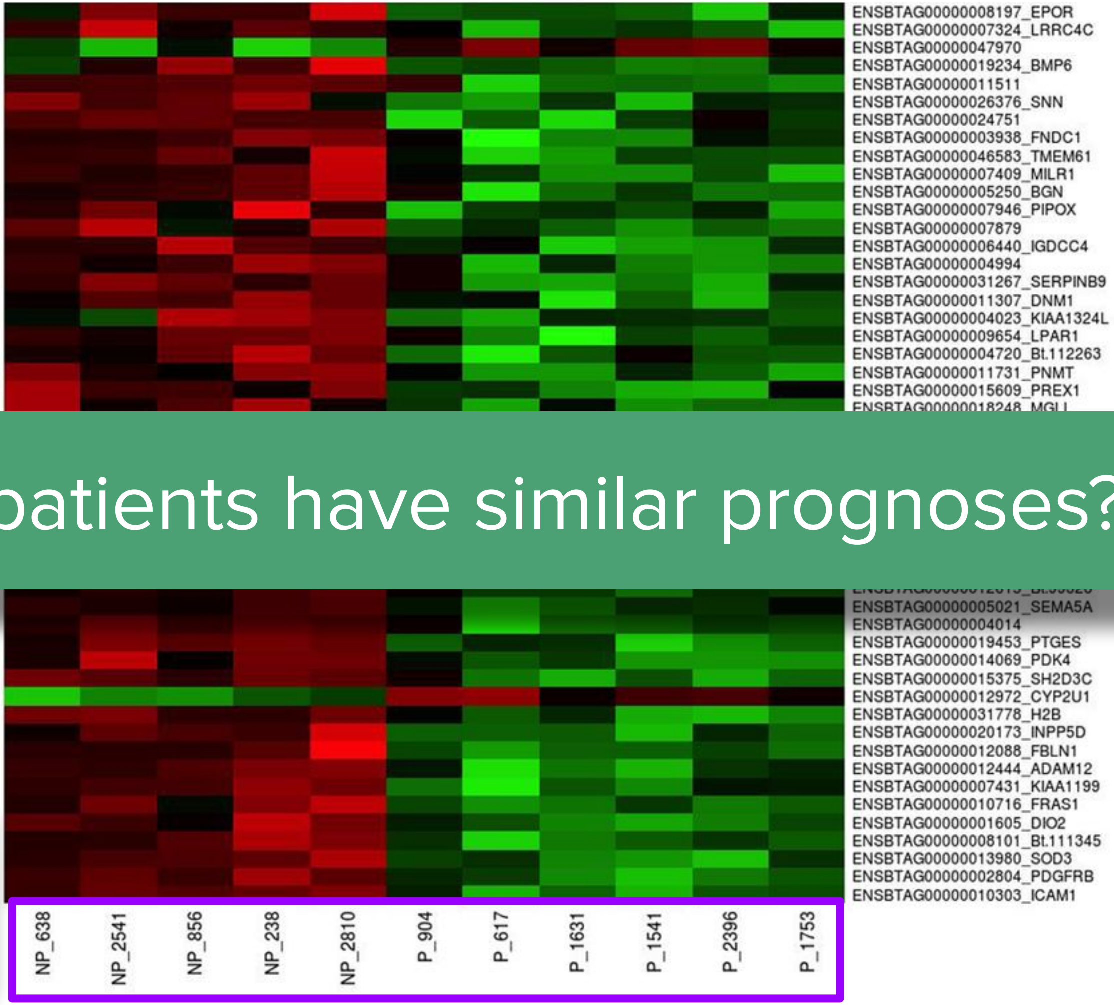
Genes

Clustering



Patients

Clustering



What patients have similar prognoses?

Patients

Cancer Research at UCSD (at a glance)

Jill
Mesirov

Trey
Ideker

Hannah
Carter

Olivier
Harismendy

Cancer Research at UCSD (at a glance)

Jill
Mesirov

Trey
Ideker

Hannah
Carter

Olivier
Harismendy

Cancer Genomics Cross-Lab Meeting

What is **clustering**?

Clustering

- Given data points, we want to “maximally separate” them

Clustering

- Given data points, we want to “maximally separate” them
    - **Low intra**-cluster (i.e., *within* cluster) pairwise distances

Clustering

- Given data points, we want to “maximally separate” them
    - **Low intra**-cluster (i.e., *within* cluster) pairwise distances
    - **High inter**-cluster (i.e., *across* clusters) pairwise distances

Clustering

- Given data points, we want to “maximally separate” them
  - **Low intra**-cluster (i.e., *within* cluster) pairwise distances
  - **High inter**-cluster (i.e., *across* clusters) pairwise distances
- Generally, we want to try to **minimize** the **number of clusters**

Supervised vs. Unsupervised Learning

- Previously, we looked at the **classification** problem

Supervised vs. Unsupervised Learning

- Previously, we looked at the **classification** problem
    - Given labeled points, train a computer to classify new data

Supervised vs. Unsupervised Learning

- Previously, we looked at the **classification** problem
  - Given labeled points, train a computer to classify new data
  - That was an example of **supervised** learning (we know labels)

Supervised vs. Unsupervised Learning

- Previously, we looked at the **classification** problem
  - Given labeled points, train a computer to classify new data
  - That was an example of **supervised** learning (we know labels)
- **Clustering** is an example of **unsupervised** learning

Supervised vs. Unsupervised Learning

- Previously, we looked at the **classification** problem
  - Given labeled points, train a computer to classify new data
  - That was an example of **supervised** learning (we know labels)
- **Clustering** is an example of **unsupervised** learning
  - We don’t know labels or anything like that in advance

Supervised vs. Unsupervised Learning

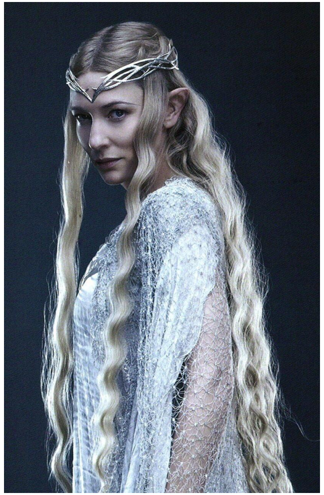
- Previously, we looked at the **classification** problem
  - Given labeled points, train a computer to classify new data
  - That was an example of **supervised** learning (we know labels)
- **Clustering** is an example of **unsupervised** learning
  - We don’t know labels or anything like that in advance
  - We want the computer to learn the underlying structure of the data

Cluster These Tolkien Characters

Legolas

Frodo

Aragorn



Galadriel

Bard

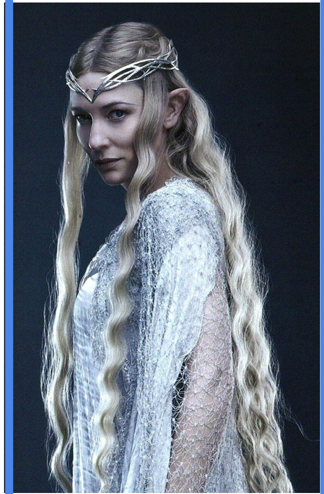
Cluster These Tolkien Characters

Legolas

Frodo

Aragorn

Bard



Galadriel

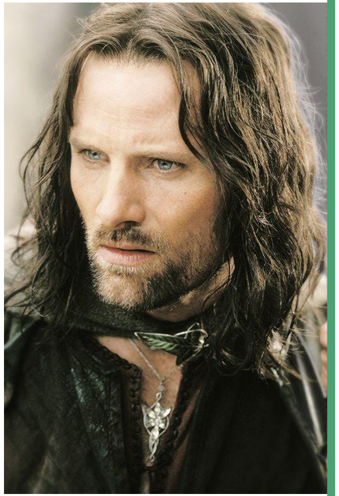
Male

Female

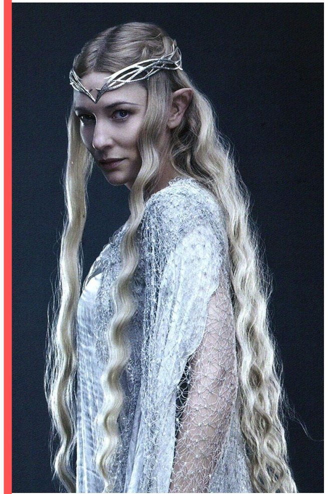
Cluster These Tolkien Characters

Legolas

Frodo



Aragorn



Galadriel

Bard

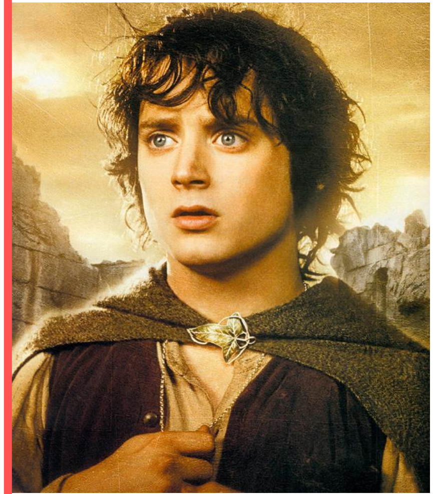
Part of the Fellowship of the Ring

Cluster These Tolkien Characters

Legolas

Aragorn

Galadriel



Frodo

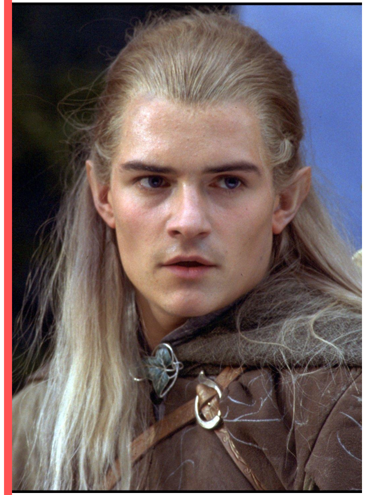
Bard

Royalty

Not Royalty

Cluster These Tolkien Characters

Bard



Legolas

Frodo

Aragorn

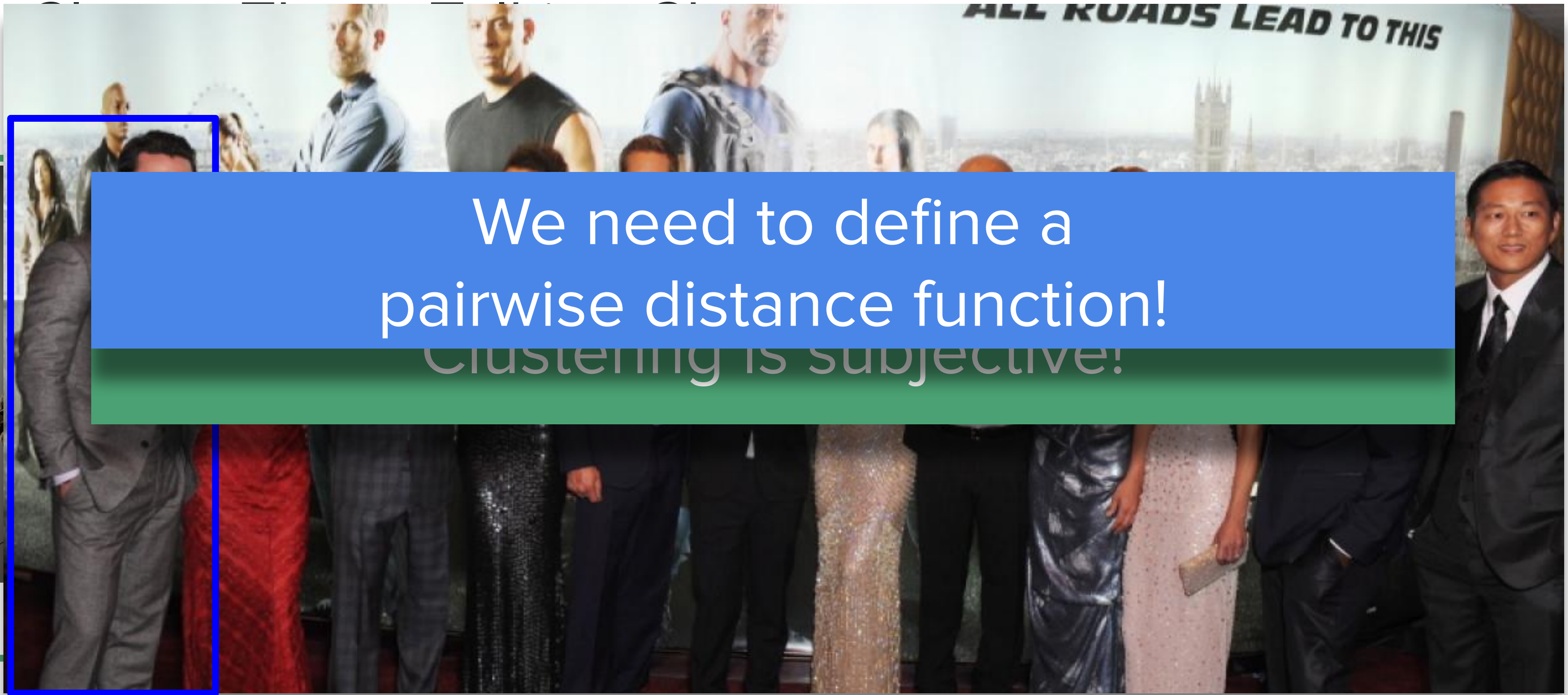
Galadriel

Starred in a Fast & Furious Movie

Starred in a Fast & Furious Movie

Clustering is subjective!

Starred in a Fast & Furious Movie



We need to define a pairwise distance function!

Clustering is subjective!

Starred in a Fast & Furious Movie

Defining a Pairwise Distance Function

- We need to define a **pairwise distance function**

Defining a Pairwise Distance Function

- We need to define a **pairwise distance function**
    - Let *u* and *v* denote two objects from the universe of possibilities

Defining a Pairwise Distance Function

- We need to define a **pairwise distance function**
  - Let *u* and *v* denote two objects from the universe of possibilities
  - ***D*(*u*,*v*)** denotes the distance between *u* and *v*

Defining a Pairwise Distance Function

- We need to define a **pairwise distance function**
    - Let *u* and *v* denote two objects from the universe of possibilities
    - ***D*(*u*,*v*)** denotes the distance between *u* and *v*
    - E.g. Euclidean distance

Defining a Pairwise Distance Function

- We need to define a **pairwise distance function**
  - Let *u* and *v* denote two objects from the universe of possibilities
  - ***D*(*u*,*v*)** denotes the distance between *u* and *v*
  - E.g. Euclidean distance
- We can alternatively define a **similarity function** (and negate it)

Desirable Properties of a Clustering Algorithm

Desirable Properties of a Clustering Algorithm

- Scalability (in terms of both time and space)

Desirable Properties of a Clustering Algorithm

- Scalability (in terms of both time and space)
- Ability to deal with different data types

Desirable Properties of a Clustering Algorithm

- Scalability (in terms of both time and space)
- Ability to deal with different data types
- Minimal requirements for domain knowledge to determine inputs

Desirable Properties of a Clustering Algorithm

- Scalability (in terms of both time and space)
- Ability to deal with different data types
- Minimal requirements for domain knowledge to determine inputs
- Interpretability and usability

Desirable Properties of a Clustering Algorithm

- Scalability (in terms of both time and space)
- Ability to deal with different data types
- Minimal requirements for domain knowledge to determine inputs
- Interpretability and usability
- *Optional: Incorporation of user-specified constraints*

Partitional vs. Hierarchical Clustering

Partitional vs. Hierarchical Clustering

- **Partitional:** Construct partitions and evaluate them by some criterion

Partitional vs. Hierarchical Clustering

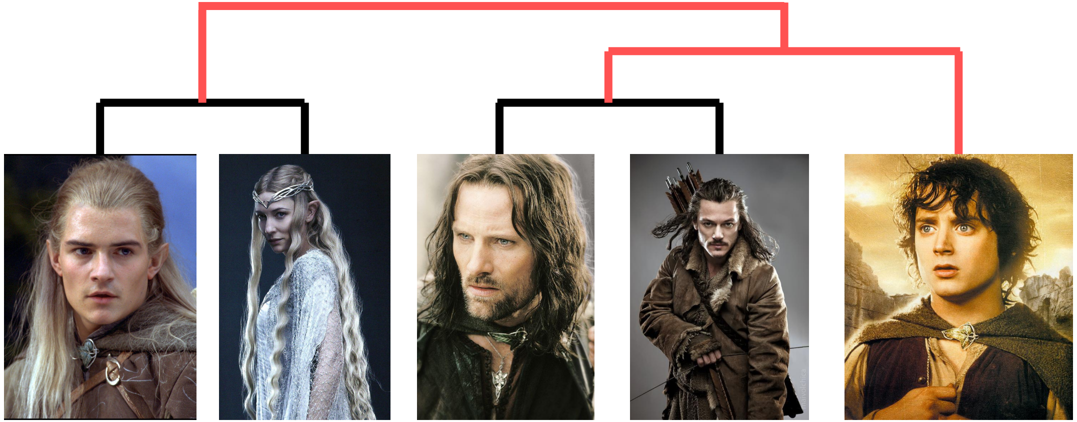
- **Partitional:** Construct partitions and evaluate them by some criterion
- **Hierarchical:** Create a hierarchy using some criterion

Partitional vs. Hierarchical Clustering

- **Partitional:** Construct partitions and evaluate them by some criterion
- **Hierarchical:** Create a hierarchy using some criterion
    - A hierarchical clustering can be “cut” into a partitional clustering

Partitional vs. Hierarchical Clustering

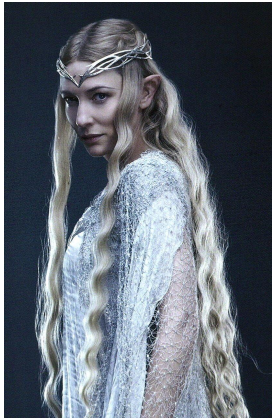
- **Partitional:** Construct partitions and evaluate them by some criterion
- **Hierarchical:** Create a hierarchy using some criterion
    - A hierarchical clustering can be “cut” into a partitional clustering



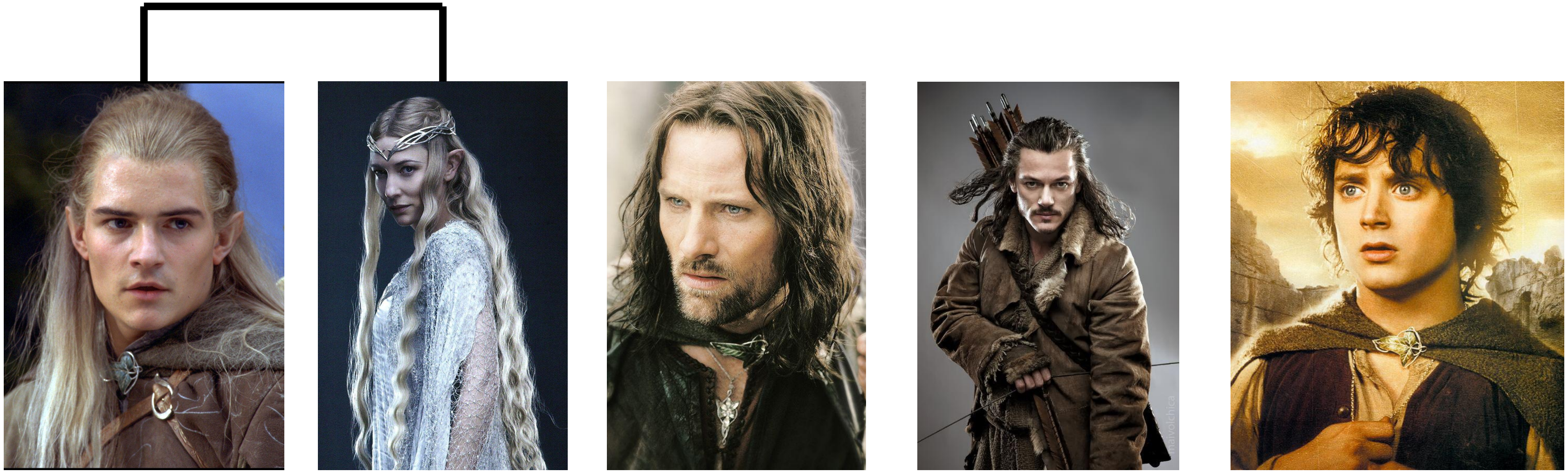
Partitional vs. Hierarchical Clustering

- **Partitional:** Construct partitions and evaluate them by some criterion
- **Hierarchical:** Create a hierarchy using some criterion
  - A hierarchical clustering can be “cut” into a partitional clustering

Bottom-Up Hierarchical Clustering

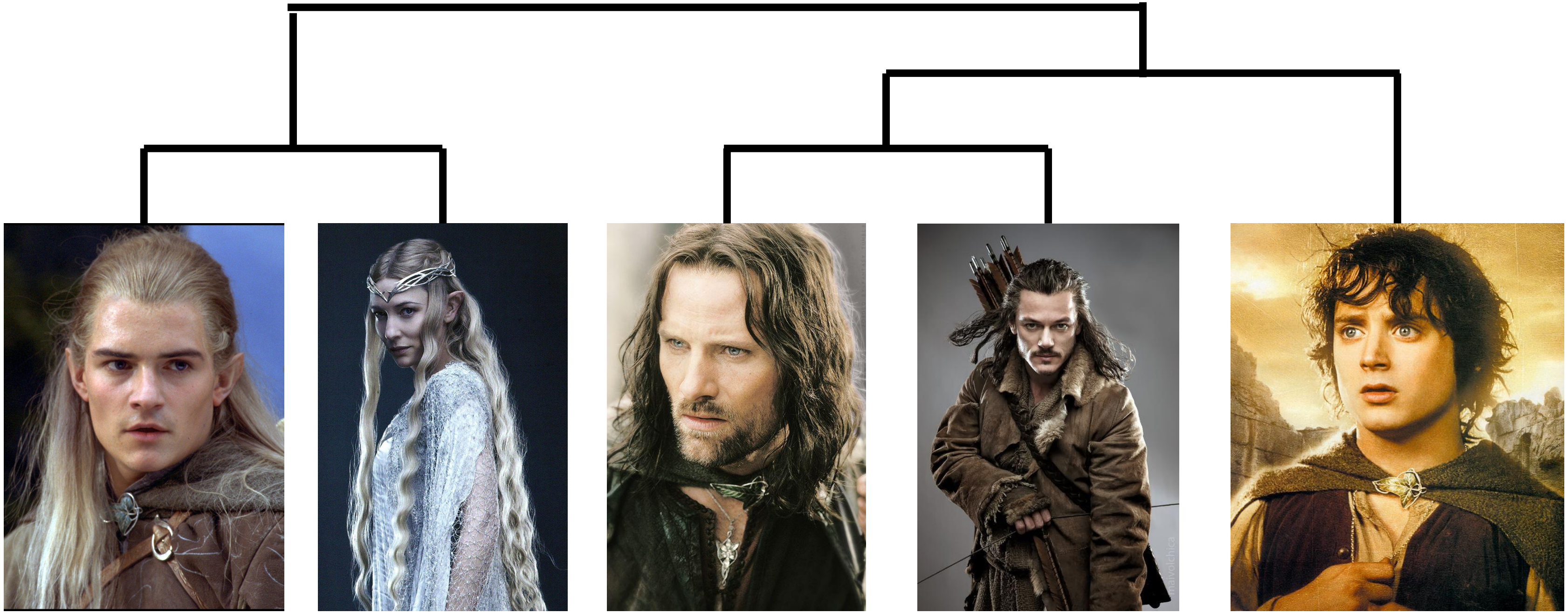
Bottom-Up Hierarchical Clustering



Bottom-Up Hierarchical Clustering

Bottom-Up Hierarchical Clustering

Bottom-Up Hierarchical Clustering



Bottom-Up Hierarchical Clustering

Computing Distances Between Clusters

- Given our pairwise distance function D, we know how to compute the distance between two individual objects u and v

Computing Distances Between Clusters

- Given our pairwise distance function D, we know how to compute the distance between two individual objects u and v
  - It's just $D(u,v)$

Computing Distances Between Clusters

- Given our pairwise distance function D, we know how to compute the distance between two individual objects u and v
  - It's just $D(u,v)$
- How do we compute the distance between two clusters A and B?

Computing Distances Between Clusters

- Given our pairwise distance function *D*, we know how to compute the

$$\boxed{D(A,B) = \min_{u \in A} \min_{v \in B} D(u,v)}$$

- How do we compute the distance between two clusters *A* and *B*?
  - **Minimum** distance between any element *u* in *A* and *v* in *B*

Computing Distances Between Clusters

- Given our pairwise distance function D, we know how to compute the

$$\boxed{D(A,B) = \max_{u \in A} \max_{v \in B} D(u,v)}$$

- How do we compute the distance between two clusters A and B?
  - **Maximum** distance between any element u in A and v in B

Computing Distances Between Clusters

- Given our pairwise distance function *D*, we know how to compute the

$$\boxed{D(A,B) = \frac{\sum_{u \in A} \sum_{v \in B} D(u,v)}{|A||B|/2}}$$

- How do we compute the distance between two clusters *A* and *B*?
    - **Average** distance between any element *u* in *A* and *v* in *B*

Computing Distances Between Clusters

- Given our pairwise distance function *D*, we know how to compute the

$$D(A, B) = \frac{\cdots (u, v)}{|A||B|/2}$$

Something else?

- How do we compute the distance between two clusters *A* and *B*?
    - **Average** distance between any element *u* in *A* and *v* in *B*

Soft vs. Hard Partitional Clustering

Soft vs. Hard Partitional Clustering

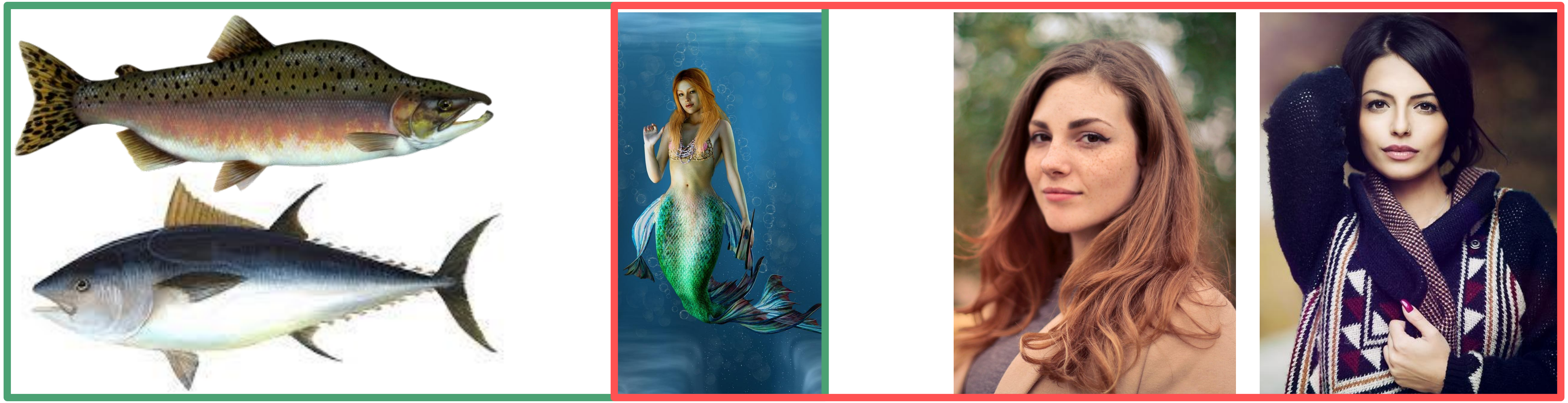
- In a **hard** clustering, each point is assigned to a single cluster

Soft vs. Hard Partitional Clustering

- In a **hard** clustering, each point is assigned to a single cluster
    - What about if a point looks like a mix of clusters?

Soft vs. Hard Partitional Clustering

- In a **hard** clustering, each point is assigned to a single cluster
  - What about if a point looks like a mix of clusters?
- In a **soft** clustering, each point is partially assigned to multiple clusters



Some Common Clustering Algorithms

Some Common Clustering Algorithms

- ***K*-means Clustering:** Cluster points into *k* clusters

Some Common Clustering Algorithms

- ***K*-means Clustering:** Cluster points into *k* clusters
    - Need to specify *k* (or can try multiple different values)

Some Common Clustering Algorithms

- ***K*-means Clustering:** Cluster points into *k* clusters
    - Need to specify *k* (or can try multiple different values)
- **UPGMA:** A common bottom-up hierarchical clustering algorithm

Some Common Clustering Algorithms

- ***K*-means Clustering:** Cluster points into *k* clusters
  - Need to specify *k* (or can try multiple different values)
- **UPGMA:** A common bottom-up hierarchical clustering algorithm
- **Gaussian Mixture Models:** Fit a mixture of distributions, and assign each point to its closest distribution