clustering

April 4, 2022

1 CLUSTERING - Bioinformatics on Genomics 2022

1.1 THEMES

During this session we will work on: * PCA * k-means * hierarchical clustering

1.2 DATA

We will look at data from this publication: "RNAseq analysis of heart tissue from mice treated with atenolol and isoproterenol reveals a reciprocal transcriptional respons" (*2016, Prunotto et al.*)

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5015234/

1.3 PROGRAMMING

Python 3

1.3.1 STEP1: Libs

These are the libs we will be using, import them: * `pandas` as `pd` for file I/O

[1]:
```
import numpy
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt
import scipy.stats as stats
from pandas import DataFrame
from sklearn.cluster import KMeans
from scipy.cluster.hierarchy import dendrogram, linkage
from sklearn.cluster import AgglomerativeClustering
```

1.3.2 STEP2: data import

Import the following data (found under input/PGX_data). * tmm_data.tab * genes_entrezid.tab * samples.tab

First, import expression data and gene ids into into data structures: numpy arrays are a good choice. Then import samples: here you need to open the file and read a list of strings.

[2]:
```
# import expression data and gene ids
expression_data = ...
genes = ...
```

```
# import samples
samples = ...
```

1.3.3 STEP3: inspect and prepare your data

Can you infer what they contain and how they go together? How many rows and columns in each data structure? Visualize a few elements to get an idea of the contents.

[3]:
```
print("Number of rows and columns for each data matrix:")
...
print("Expression matrix (first sample, first 10 genes):")
...
```

```
Number of rows and columns for each data matrix:
Expression matrix (first sample, first 10 genes):
```

Now, prepare the date for our analysis by normalizing them: log transform, then zscore (first by sample, then by gene). Visualize a few elements before and after normalization. Finally, assempble all the data in one data structure (for example, a pandas dataframe).

[4]:
```
# normalize the data
expression_data_log = ...
expression_data_normalized = ...
# visualize a few elements
...
# assemble all data into one data structure
data = DataFrame(...
```

```
  File "<ipython-input-4-6bb66b0bc033>", line 7
    data = DataFrame(…
                     ^
SyntaxError: unexpected EOF while parsing
```

1.3.4 STEP4: perform PCA dimensionality reduction

Run PCA on expression data. Capture the loadings corresponding to the first two components.

[5]:
```
pca = ...
```

Plot loading on cartesian axes: color according to treatment. Look at the sample name ending in ATE/ISO/CTR. * mice treated with the -blocker atenolol (sample name ending in ATE) * mice trated with the -agonist isoproterenol (sample name ending in ISO) * controls (sample name ending in CTR)

[6]:
```
plt.scatter(...
```

```
  File "<ipython-input-6-09bd95c1e97e>", line 1
    plt.scatter(…
                ^
SyntaxError: unexpected EOF while parsing
```

1.3.5 STEP5: perform k-means clustering

Run k-means clustering on expression data. Look again for 3 clusters. Note that, depending in the library you use, you might have to rotate the data to cluster by sample.

[7]:
```
kmeans = ...
```

Repeat the PCA plot above, this time coloring the points according to the labels that were assigned by the clustering algorithm. Does it make sense, visually?

[8]:
```
plt.scatter(...
```

```
  File "<ipython-input-8-09bd95c1e97e>", line 1
    plt.scatter(…
                ^
SyntaxError: unexpected EOF while parsing
```

1.3.6 STEP6: perform h-clustering clustering

Perform hierarchical clusterin: plot a dendrogram of our expression data. Lable the leaves using the sample names: what do they mostly cluster by?

[9]:
```
# Calculate the distance between each sample
...

# plot the dendogram
...
```

[ ]:
```

```

[ ]:
```

```