Task: Given 5 different time points of genes and their gene expression values, determine the groups of genes that exhibit similar properties according to their gene ontology.

**Using Hidden Markov Models for Clustering**

Why is using HMMs a good idea?

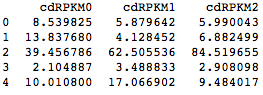
**It captures the sequential behaviour between time points, as opposed to k-means clustering. E.g. In k-means clustering, randomly permuting the time points while using the Euclidean distance metric would give us the same answer. Moreover, for HMMs, we do not have to define a distance function.**

So how are HMMs used to solve this problem?  
**Start from an initial collection of HMMs, an iterative procedure finds cluster models and an assignment of data points to these models that maximizes the join likelihood of the clustering.**

We use ***hmmlearn*** library, which uses the Baum Welch algorithm to estimate parameters by assuming the emissions are from a Gaussian distribution (ref: Exploratory Analysis). This library is reasonably fast (3 seconds for ~ 11000 datapoints, 7 hidden states)

**Procedure**

1. Assign the initial number of components, n=3.
2. Fit the data (using EM) (first 5 rows given below) to get the Hidden State for each sequence in the 11087 observations we have.



1. Check number of samples belonging to each state.

Result: {0: 5983, 1: 4151, 2: 953}

1. Get the transition matrix, means and variances of each state (assume diagonal covariance, i.e. each hidden state has a diagonal covariance matrix. These parameters have been estimated using Baum Welch Algo (EM).

Transition matrix

[[ 0.54088498 0.3791243 0.07999072]

[ 0.5353866 0.37579376 0.08881964]

[ 0.52178083 0.35773401 0.12048517]]

Means and vars of each hidden state

Hidden state 1

mean = [ 6.1464168 6.30849677 7.09616838]

var = [ 18.49816135 19.10115191 22.95627763]

Hidden state 2

mean = [ 29.51966805 32.28184107 32.04601375]

var = [ 266.4165085 303.65605753 271.48850803]

Hidden state 3

mean = [ 205.10211459 224.31116124 182.87103404]

var = [ 81402.08106441 90082.84812639 44742.93741073]

1. For each sequence, plot it according to its predicted hidden state.

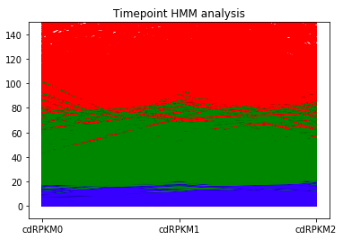


Fig. All gene expression values clustered

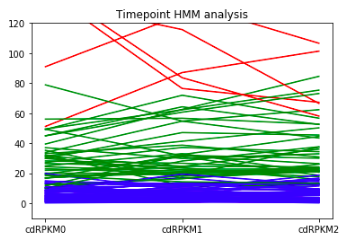


Fig. Gene expression values for 120 sequences, clustered

Blue: State 1, Green: State 2 ,Red: State 3

1. Get descriptive statistics for each cluster, and modify the prior means and prior covariance for actual clustering (not implemented yet).

Descriptive statistics for cluster 1 is

0 1 2

count 5983.000000 5983.000000 5983.000000

mean 6.139204 6.310284 7.087727

std 4.257891 4.336890 4.745488

min 0.074259 0.073034 0.074316

25% 2.593439 2.677915 3.147669

50% 5.395196 5.470832 6.376432

75% 9.020419 9.347439 10.203773

max 21.829329 21.232039 22.683164

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Descriptive statistics for cluster 2 is

0 1 2

count 4151.000000 4151.000000 4151.000000

mean 29.732860 32.484161 32.274286

std 16.295154 17.341424 16.429757

min 0.928242 1.893303 1.757234

25% 17.847245 19.295431 20.120568

50% 25.437777 27.579762 27.691935

75% 37.623006 42.018830 41.055019

max 101.319960 96.745990 100.798797

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Descriptive statistics for cluster 3 is

0 1 2

count 953.000000 953.000000 953.000000

mean 206.643537 226.082221 184.150928

std 286.466793 301.279603 212.304157

min 7.692122 26.774226 26.606169

25% 82.970543 95.886629 87.766905

50% 117.812188 131.070226 117.446857

75% 196.832557 219.257593 185.131477

max 2656.834519 2849.584643 2268.660153

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. We can also get samples from the model, as well as get log-likelihood scores for samples we provide.

Generate 10 samples:

(array([[ 8.16368561, 4.10142349, 13.12486475],

[ 38.96375391, 29.24911109, 19.0020877 ],

[ 6.63347733, 7.66973802, 4.87582102],

[ 5.98370251, -2.1842652 , 11.04637392],

[ 2.96879384, 1.78530831, 5.42850131],

[ 10.69734801, 47.33830186, 18.03384904],

[ 5.43147868, 3.09367272, 9.70172824],

[ 489.36455322, 389.68573477, 521.71837249],

[ 9.75992579, 16.48335967, 5.49995756],

[ 41.91398146, 36.29348922, 25.78110123]])

**Problems**

1. Calling the fit method from ***hmmlearn*** is giving me the hidden state from which each sample is generated, not a sequence of hidden states. I don’t know if this is accurate-my idea (from the paper I read) is that we get a sequence of hidden states for each

HMM.

1. Don’t really understand the significance of a diagonal vs spherical covariance matrix.