ML4FG Final Presentation:
Modeling Colorectal Cancer Gene Expression Distributions using Mixture Models

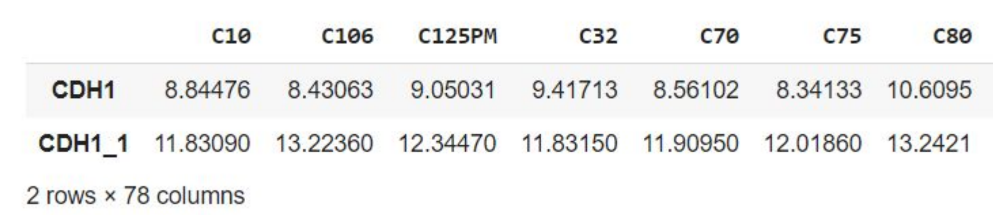
Shomik Ghose & Austin Tao

Introduction

Goal: Use mixture modeling (unsupervised ML) to fit continuous gene expression distributions in colorectal cancer cells

Data:

- Bodmer Microarray Phenotype: https://github.com/jeffliu6068/GMMchi/blob/main/Bodmer_microarray_phenotype.zip
- Rows (genes) x Columns (cell lines). Each (row, column) entry represents the logarithmic gene expression value for that cell line.
- Preprocessing: Handling *duplicates* and *null values*



	C10	C106	C125PM	C32	C70	C75	C80
CDH1	8.84476	8.43063	9.05031	9.41713	8.56102	8.34133	10.6095
CDH1_1	11.83090	13.22360	12.34470	11.83150	11.90950	12.01860	13.2421

2 rows × 78 columns

Methods I

Existing packages:

- GMMchi
- Gaussian Mixture Models (sklearn.mixture)
- Student-t Mixture Models (smm)

Novel extension of existing packages:

- Weighted Average (of Gaussian and student-t)

Own implementation:

- Gaussian Mixture Models
- Novel: Shifted Asymmetric Laplace (SAL) Mixture Models

Model Selection Metrics

- Bayesian Information Criterion (BIC)

$$\mathrm{BIC} = k\log(n) - 2\log L_{M,G}(x \mid \hat{\theta})$$

- Adjusted Least Squares (ALS)

$$\mathrm{ALS} = \sum_{i=1}^{n}(y_i - b_j)^2$$

- Area Under Difference (AUD)

$$AUD = AU_{\text{curve}} - AU_{\text{histogram}}$$

$$\mathrm{AU} = \sum_{i=1}^{n-1} y_i(x_{i+1} - x_i)$$

where

Methods II

Method to Generate Weights:

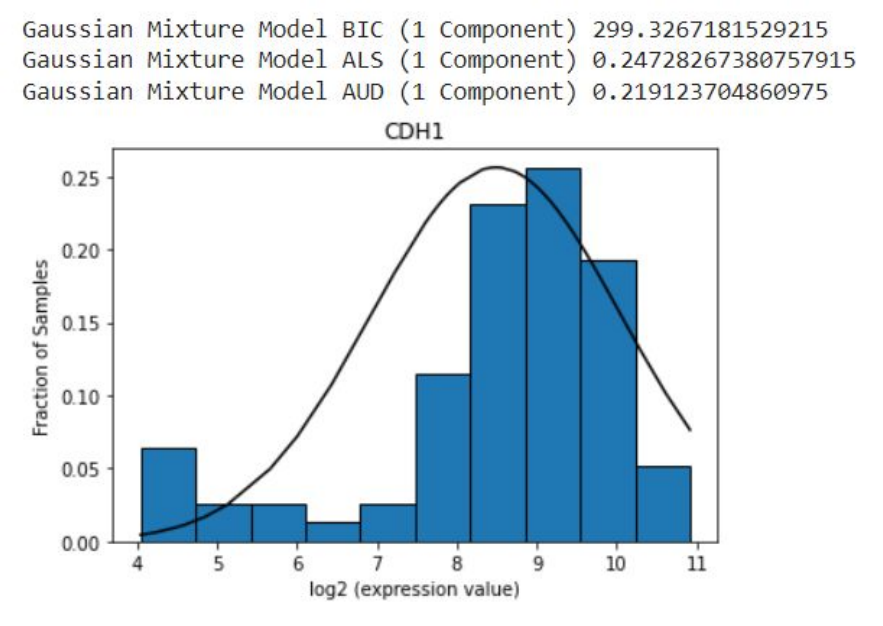
1. Keep 2 “running scores” across the dataset, one for the Gaussian mixture and one for the student-t mixture.
2. For each datapoint, calculate the ALS metric for each of the two fits. Choose min(ALS_Gaussian, ALS_t), and add that to the corresponding running score.
3. Normalize the running scores so they sum to 1, and use them as weights.

SAL Equation:

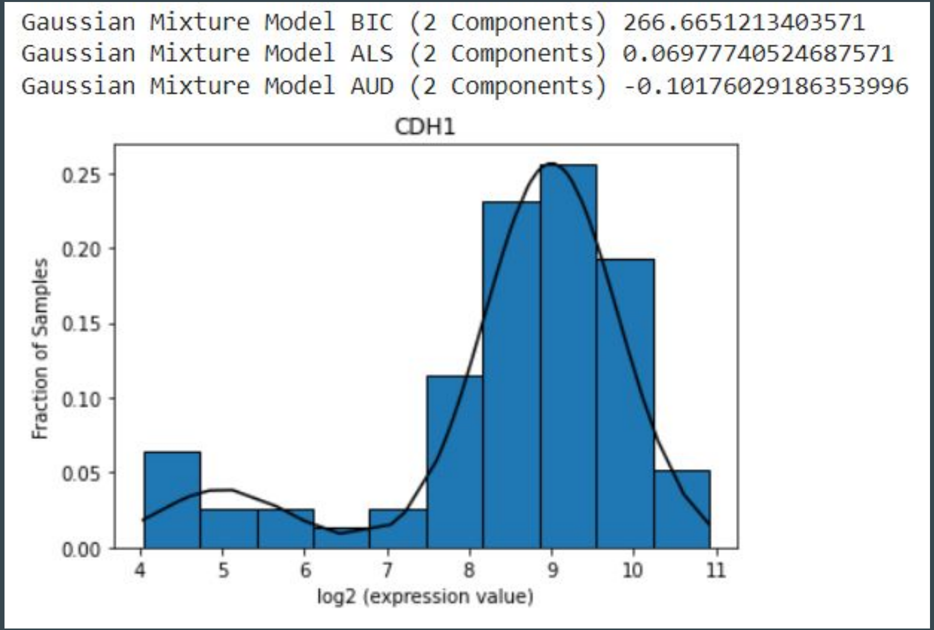
$$\mathcal{L} = \prod_{i=1}^{n} \prod_{g=1}^{G} \left[\pi_g \phi\left(\mathbf{x}_i \mid \boldsymbol{\mu}_g + w_i \boldsymbol{\alpha}_g, w_i \boldsymbol{\Sigma}_g\right) h\left(w_i\right)\right]^{\tau_{ig}}$$

where $\phi \sim \mathcal{N}(\boldsymbol{\mu}_g + w_i \boldsymbol{\alpha}_g, w_i \boldsymbol{\Sigma}_g)$

Results I



```
Gaussian Mixture Model BIC (1 Component) 299.3267181529215
Gaussian Mixture Model ALS (1 Component) 0.24728267380757915
Gaussian Mixture Model AUD (1 Component) 0.219123704860975
```

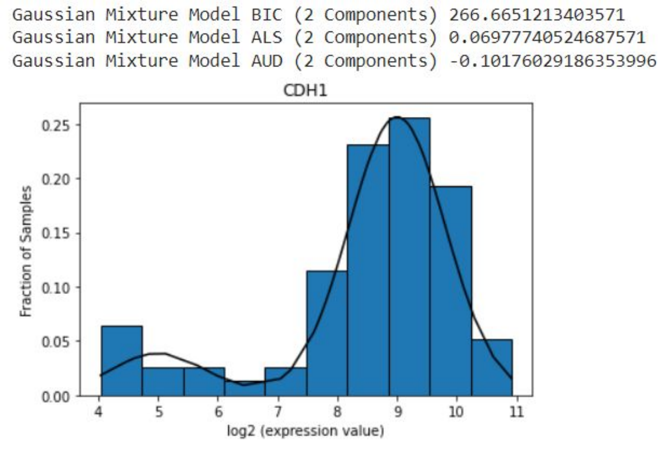


```
Gaussian Mixture Model BIC (2 Components) 266.6651213403571
Gaussian Mixture Model ALS (2 Components) 0.06977740524687571
Gaussian Mixture Model AUD (2 Components) -0.10176029186353996
```

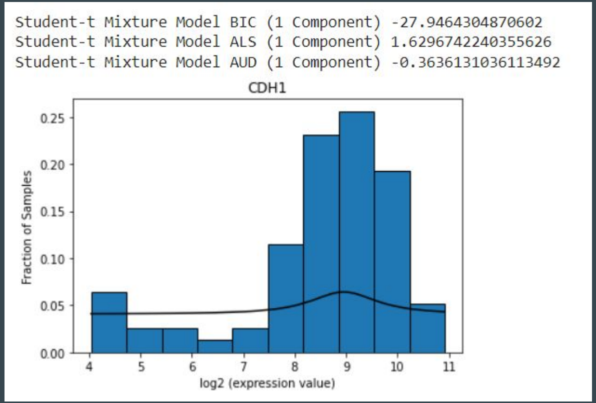
Results II

Existing Packages

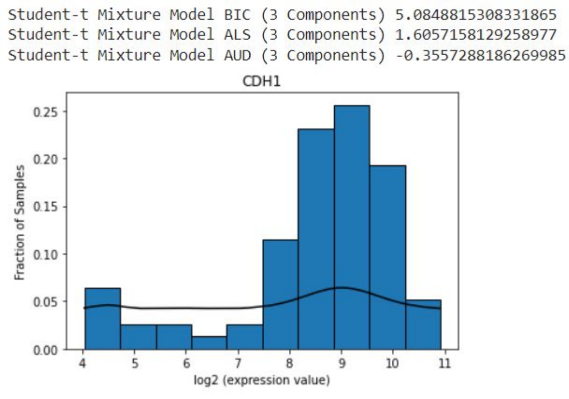
- Weighted average method best accounts for non-normal tail
- BIC is not effective at determining best fit; ALS and AUD may be better differentiators



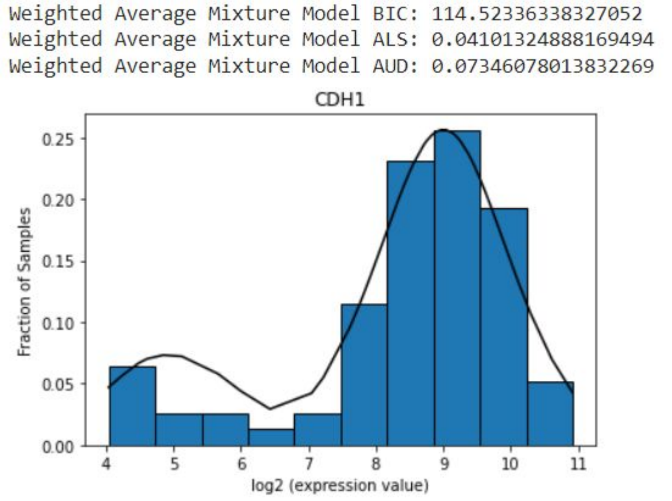
```
Gaussian Mixture Model BIC (2 Components) 266.6651213403571
Gaussian Mixture Model ALS (2 Components) 0.06977740524687571
Gaussian Mixture Model AUD (2 Components) -0.10176029186353996
```



```
Student-t Mixture Model BIC (1 Component) -27.9464304870602
Student-t Mixture Model ALS (1 Component) 1.6296742240355626
Student-t Mixture Model AUD (1 Component) -0.3636131036113492
```



```
Student-t Mixture Model BIC (3 Components) 5.0848815308331865
Student-t Mixture Model ALS (3 Components) 1.6057158129258977
Student-t Mixture Model AUD (3 Components) -0.3557288186269985
```



```
Weighted Average Mixture Model BIC: 114.52336338327052
Weighted Average Mixture Model ALS: 0.04101324888169494
Weighted Average Mixture Model AUD: 0.07346078013832269
```

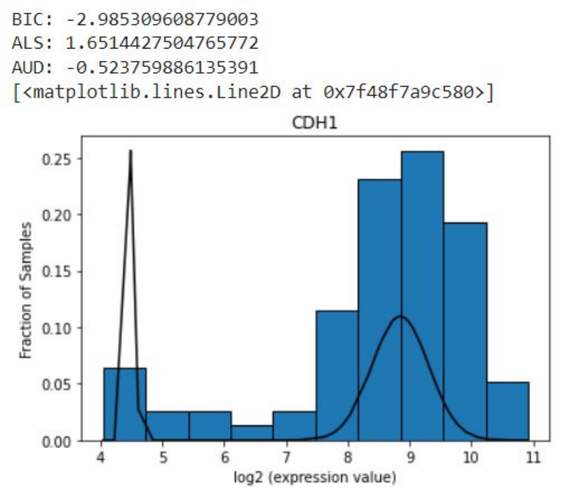
Results III

Own Implementation

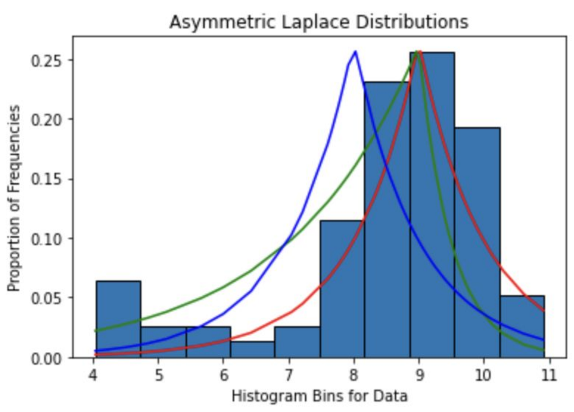
- Our GMM implementation accounts for the non-normal tail but does not adjust the peak appropriately
- Our SAL implementation forms a relatively accurate fit but is inefficient with regards to runtime

Conclusion

- The weighted average method generally produces the best fit across normal and non-normal distributions



```
BIC: -2.985309608779003
ALS: 1.6514427504765772
AUD: -0.523759886135391
[<matplotlib.lines.Line2D at 0x7f48f7a9c580>]
```



Discussion

Next Steps

- Improve our own mixture model implementation
    - Add automatic initialization of parameters
    - Improve runtime efficiency for SAL
    - Add additional distribution mixtures (e.g. Noncentralized Beta)
    - Generalize to multidimensional mixture models

- Demonstrate the benefits of having a better-fitting distribution
    - Torrente *et. al.*: Determining the optimal fit could improve accuracy of supervised techniques such as identifying prognostic marker genes