**Main Pipeline**

* Start with 2 main files: *M\_batch1\_batch2\_select\_no\_ffpe\_corrected.csv* and *phenotype\_mappings.csv* (called M\_batch and pheno respectively from now on)
  + M\_batch1 file contains methylation data (from Illumina EPIC array) for each sample (~750k cpg island each for ~1100 samples)
    - File is pre-processed by Idit
    - Values were standardized using SVA
    - ffpe samples were removed due to the process affecting the methylation values
  + Pheno file contains the phenotype information about all the samples (endo case vs control, batch, cycle phase, institute of analysis etc)
  + Both csv files are stored in my S3 storage
  + Need to remove samples from “MAMC” institute due to some error in analysis from their end
    - All notebooks below will have code to remove MAMC samples
    - Tried to remove the samples from the main 2 files itself but resulting files took a very long time to read for some reason. Had to use a work around
* Feature extraction (Meth\_full\_split\_and\_feature\_extract\_cycle\_phase.ipynb)
  + The initial models run on the would take ~40 mins. To reduce computation time we extracted the most relevant features with respect to Endo case vs control (stratified by cycle phase) from the full dataset (reducing features from ~750k to ~45k)
    - Run time went from ~40 mins to ~10 mins
  + Method:
    - Merge the 2 main files M\_batch and pheno
    - Split features and labels (cpg islands and endo case vs control)
    - Take out a test set (stratified by cycle phase) from the full dataset resulting in a combined train/val dataset so that feature extraction is only done on train/val dataset
    - Feature extraction was performed using linear regression of the cpg islands with respect to the output variable (endo case vs control)
    - Use linear regression to get p-values for cpg islands on the train/val dataset
    - Subset only the nominally significant features (~45k)
    - Filter the same 45k features present in train/val dataset for the test set
      * Both datasets need to have the same features for model
    - Split the train/val dataset into separate train and validation datasets (stratified by cycle phase)
    - Write all 3 datasets to csv
      * *Endometriosis\_subset\_train\_feature\_extract\_no\_mamc\_splitfunction.csv, Endometriosis\_subset\_val\_feature\_extract\_no\_mamc\_splitfunction.csv, Endometriosis\_subset\_test2\_feature\_extract\_no\_mamc\_splitfunction.csv*
* The models (Pytorch Endo Captum IG No MAMC.ipynb and tf\_regressor\_endo\_feature\_extract\_no\_mamc.ipynb)
  + 2 ANN model frameworks Pytorch and TensorFlow
  + Both framework notebooks go through similar pipelines
  + Method:
    - Load in feature extracted dataset (train, val and test)
    - Pre-process all 3 datasets
    - Split all 3 into features and labels
    - Train model on training dataset
      * Validation loop using val dataset
    - Predict on test dataset
    - Calculate train, val and test AUC scores
      * Plot the scores
    - Model interpretability methods:
      * Pytorch can use Captum package which includes a lot of model interpretability methods
        + Main one I used was Integrated Gradients (IG)
      * I used Feature Importance Permutation in Tensorflow
    - Used model interpretability methods to find the features that the model thinks are the most important
    - Took top 500 CPGs and write to txt file
  + CPGs to Pathways
    - Used MissMethyl (GoMeth function) to obtain pathways from CPG islands
    - Used Cytoscape to form pathway networks using MissMethyl output

**Other notebooks and pipelines**

* PCA (PCA.ipynb)
  + Load feature extracted train, test and validation datasets
    - Concatenate all 3 datasets
  + Use standard scaler on the datasets
  + Create color map to assign colors to various features (cycle phase, batch, institute etc)
  + Plot PCA and color by desired feature
  + Full dataset:
    - Perform same steps as above for the full dataset
* Feature extraction with respect to Cycle Phase (Meth\_full\_split\_and\_feature\_extract\_cycle\_phase.ipynb)
  + Extracted feature from the full dataset with Cycle Phase as y variable
  + Compared prediction accuracies for Endo and Cycle Phase with the feature extracted dataset with Endo as y variable
  + Train, val and test sets:
    - *Endometriosis\_subset\_train\_feature\_extract\_no\_mamc\_splitfunction\_cycle\_phase\_stratify\_by\_endo.csv, Endometriosis\_subset\_val\_feature\_extract\_no\_mamc\_splitfunction\_cycle\_phase\_stratify\_by\_endo.csv, Endometriosis\_subset\_test2\_feature\_extract\_no\_mamc\_splitfunction\_cycle\_phase\_stratify\_by\_endo.csv*
* Other models
  + Cycle Phase Predictions (Pytorch Cycle Phase Captum IG No MAMC.ipynb )
    - Predicted cycle phase using feature extracted dataset
    - Sanity check to confirm our results with previous study
  + Cycle phase predictions with cycle phase feature extracted dataset (Pytorch Cycle Phase Captum IG No MAMC.ipynb )
    - Just changed the train, val and test set to the cycle phase feature extracted dataset
  + Endo predictions with cycle phase feature extracted dataset (Pytorch Endo Captum IG No MAMC.ipynb)
    - Just changed the train, val and test set to the cycle phase feature extracted dataset

Notes:

This project is a continuation of Parker Grosjean’s rotation project. His code is available on the Box folder as well as my EC2 instance. His models focused more on model interpretability rather than prediction accuracy. Ours on the other hand, we had higher prediction accuracy but lower model interpretability. Our models were more complex (more layers, nodes and extensive hyperparameterization). His pipeline is built on pytorch whereas our pipeline was initially built on Tensorflow. However, due to Integrated Gradients being more compatible for tabular data on Pytorch, we translated our Tensorflow pipeline to Pytorch to use IG. Additionally in Parker’s pipeline, there was a data leakage problem (feature extraction being done on the full dataset rather than on just the combined train/val dataset).