Data loading

- load the model
  - what is the input sequence length?
  - what are the dimensions of the biological / technical features?
- load the enhancer sequences
  - calculate the GC-content of each enhancer
- **set and save the random seed**

Predict features and quantify GC-bias

- window-ize the enhancer sequences
  - Calculate the GC-content of each "center focus window"
  - Note: this part is actually always the same for models of the same input length, maybe it would be better to store the data (windows) and load it later?

- predict the technical and biological features for all sequences ("center focus windows") with padding
  - Calculate the correlation of non-aggregated technical/biological features with the "center focus window" GC-content
    - **Save this table (shape = (n_bio_features + n_tech_features, 1))**
- aggregate the windows using the mean / max
  - **Save the features after aggregation (for examply in a .npy or torch tensor file), make sure to also save the labels/enhancer names**
    - **shape = (n_bio_features, n_enhancer) and (n_tech_features, n_enhancer)**
  - calculate the correlation of the aggregated technical/biological features with the enhancer GC-content (the one calculated in the beginning)
    - **Save this table (shape = (n_bio_features + n_tech_features, 1))**

PCA

- calculate PCA on the aggregated features
  - calculate how much variance is explained by each dimension
    - save this information
  - how many principal components does it take to explain 90% of the variance?
    - save this information
  - save the PCA-results (save the python object or the table with the loadings)
  - plot PC1 vs PC2
    - color: active vs inactive
    - color: GC-content
    - color: tissues
  - plot PC3 vs PC4
    - color: active vs inactive
    - color: GC-content
    - color: tissues

Enhancer prediction task

- fit elastic net / Ridge models on the biological / technical embeddings to predict active / inactive
  - calculate the performance
    - save the performance
  - for the elastic net model, check the numberr of non-zero parameters
  - save the models / model paramters

- calculate the performance of using just GC-content
  - save the performance

- remove the GC-content dependency from the features, you can use the function below to do this
  - calculate the performance with GC-content removed
    - save the performance
  - for the elastic net model, check the number of non-zero parameters
  - save the models/ model parameters

tissue specificity prediction task

- fit elastic net / Ridge models on the biological / technical embeddings to predict tissue
  - calculate the performance
    - save the performance
  - for the elastic net model, check the number of non-zero parameters
  - save the models / model paramters

- calculate the performance of using just GC-content

    - save the performance

- remove the GC-content dependency from the features, you can use the function below to do this
    - calculate the performance with GC-content removed
        - save the performance
    - for the elastic net model, check the number of non-zero parameters
    - save the models/ model parameters