**Cancer Immunotherapy Grand Data Science Challenge - Challenge 3**

Part 1:

statistics s(.) => summary of gene expression of distribution Pi

To summarize the gene expression distribution Pi after knocking out gene i, we use a Kernel Principal Component Analysis (Kernel\_PCA). A Kernel\_PCA will capture the maximum variance of the gene expression in high dimensional space using a nonlinear kernel. In a standard PCA, covariance matrices are central in defining the linear dependence among random variables. As a dimensionality reduction technique, PCA utilizes eigen-decomposition of the covariance matrix to find low dimensional projections of data in linear space. Kernel\_PCA, however, generalizes standard PCA with a nonlinear kernel using the dot product of the matrix in high dimensional space (ref.1).

We can use a Kernel\_PCA to define our choice our gene expressions in the following manner:

* We choose a kernel function: K (x, y) = φ(x) \* φ(y) where φ(x) is the mapping function of the data into the high-dimensional feature space.
* We compute the kernel matrix: K = [K (x\_1, x\_1), ..., K (x\_n, x\_n)], where x\_i is a sample of the gene expression distribution Pi.
* We compute the eigenvalues and eigenvectors of the kernel matrix K: K = λV, where λ is the vector of eigenvalues and V is the matrix of eigenvectors.
* The first Kernel\_PCA score is the dot product of the gene expression distribution Pi with the first eigenvector
* Kernel\_PCA\_score = Pi \* V\_1, where V\_1 is the first column of the matrix V
* We solve for P0, Q, S^(Pi) using the Kernel\_PCA\_score
* Kernel\_PCA\_score (Pi, Q, S^(Pi)) = **w1 \* abs (Kernel\_PCA\_score(P0) - Kernel\_PCA\_score (Pi)) + w2 \* abs (Q - Kernel\_PCA\_score (Pi))**
* w1 and w2 are the weights that define the importance of each term
* The 1st part of the formula defines the difference between the score of the unperturbed data and the score of the perturbed data.
* The 2nd part of the formula defines the difference between the predicted cell state proportion Q and the score of the perturbed data.
* This allows us to summarize the gene expression in an unsupervised way, taking into consideration the unperturbed and perturbed data.

ref.1: arXiv:2211.13172v1 [stat.ML] 23 Nov 2022. KERNEL PCA FOR MULTIVARIATE EXTREMES. MARCO AVELLA MEDINA, RICHARD A. DAVIS, AND GENNADY SAMORODNITSKY.

Part 2:

scoring function => takes in P0, Q, s\_hat (Pi) => outputs score of gene i => a larger score = better perturbation

As an observation from the analysis of the target variable (gene state) in challenge 1, it was clear that there was a great imbalance in the data between the five states. A statistical metric that handles well imbalanced data is the Weight Mean Square Error (WMSE). While the Mean Square Error handles the error raised by the samples in the classes in an equal way, in WMSE the error raised is handled by a class influence term (ref.2).

* We refer to this class influence term as ßn.
* WMSE = 1/N \* ∑ (i=1 to N) ((yhat - ytrue) ²) \* βn

Let us solve for WMSE\_score for our challenge:

* WMSE (Q, s^(Pi), N) = 1/N \* ∑ (i=1 to N) ((Qi - s^(Pi)) ²) \* βj
* N is the number of genes
* i is the index of the gene
* βj is the weight of each gene
* βj = Cn / N
* βn is the ration between the number of samples in the nth class and the number of samples in the training dataset
* Cn is the number of instances in class n of the training dataset
* The relevance is greater for classes with a greater number of samples in the training dataset
* This means that classes with greater influence value contributes more to the error

ref.2: LAXMI SREE B R et al: A WEIGHTED MEAN SQUARE ERROR TECHNIQUE TO TRAIN DEEP BELIEF. A Weighted Mean Square Error Technique to Train Deep Belief Networks for Imbalanced Data. Laxmi Sree B R. Vijaya M S.