**Proteomics Expression Analysis with AI modeling for Detecting Missing Values from MS Acquisition with TMT labeling**

**Abstract**

**Motivation**

A common task in proteomics data analysis is estimating differential protein abundance between experimental conditions. While many applications exist to perform these calculations, they commonly filter any PSM with even one missing technical replicate. The impact of this missing data filter is often small, but if there are many missing replicates or if there are a low number of PSMs per protein than a method to impute missing data is valuable. Certain values may be missing at random, missing not at random, or missing completely at random. If a value is missing not at random the MS instrument may not have picked up a signal that was there. If this is the case, likely the majority of the other technical replicates would be present. For values missing at random the MS instrument likely would not pick up a signal for majority of technical replicates. A case where an MS instrument does not pick up a signal at one technical replicate where all others are present – this event likely is missing completely at random.

**Results**

We present a new tool, written in Python and R, that allows a user to impute missing data using a variety of algorithms, and then estimate differential abundance on the corrected matrix. The proteomics matrix is computed in Python using the model of choice among missForest, KNN or RegImpute from the missingpy and StaticImpute libraries. All PCA and Volcano plots are generated in R using the Shiny platform.

**Availability and Implementation**

**Introduction**

An important technique used to determine validity in Proteomics MS data is the percentage of missing values and the variance between technical replicates. With little variance and missing values it is possible to move forward with an improved matrix. Some PSMs may be left out of the final analysis if a matrix isn’t generated to fill in minor missing values. This is important in analyses when particular proteins may not show up as highly regulated due to one or two missing values; but, these are actually important for the outcome of the experiment.

Our tool hopes to eliminate the gray area about reasons for missing values in Proteomics data. Much of the debate is whether the missing values are missing due to a valid instance or if this was determined by a random instance. If we take a look at the minimal random noise level and compare the data to this, we can get a sense of the validity of the abundance during an MS run. Noting the validity of each MS abundance of a peptide spectrum match can help us get a better sense of the confidence in each Protein fold change despite a significant p-value.

**Materials and Methods**

It’s important to predict the different types of occurrences responsible for missing data, in order to compute each value accordingly. At times datasets could have many missing values, in this case any prediction matrix used to fit a model wouldn’t be best. If a dataset has moderate to high amounts of data filled with MS data, modelling is appropriate. In many cases this aspect of Proteomics data analysis is ignored; and, here we bring a tool that can pick up on the variability in a dataset and compute these blanks. In order to calculate this variability within each technical replicate we use an optimized method involving the testing and training dataset composed of 100 matrices of different MS datasets resembling the original MS data. A matrix is then computed with a percentage predicting whether a blank is missing at random or not at random for any blank entries in the original MS data matrix. Based on the chosen model, each value is calculated if the prediction of whether it is a blank or not is above 60%. This is an implementation of an existing AI method for Proteomics Quantification.

By generating results from 3 different models, KNN K-nearest neighbors, missForest, and RegImpute, we are able to decide which model represents the data best. KNN, applied first, is an algorithm based on K-nearest neighbors, or the proximity between data points. In our case k=10, thus, there are 10 points forming clusters that are used to determine this proximity. Data points are computed based on the cluster that they belong to. MissForest is an algorithm based on iterations of predicting the missing values. Each time a prediction is made for a particular missing value, the algorithm uses this as a new training set to predict the next value for the same missing value. Data points are calculated based on previous predictions. RegImpute uses values to calculate missing values based on a linear regression of the current dataset. Missing values in the dataset are computed based on points within the linear regression that closely resemble the points in the neighboring technical replicates. While the new dataset cannot match the original dataset completely we are looking for a new matrix that introduces minimal change with high accuracy in terms of nearest technical replicates. While it is important to maintain the integrity of the original dataset, we plot the difference and note any outliers that may have changed. The results of some models may create false fold changes or p-values that are artificially inflated, the model creating these results are rejected. The model that produces changes in any PSM fold change aligned with the assigned protein fold change remains as a solid parameter for the PEA tool. Each dataset may be different, so the user can decide which model best represents their data.

PEA starts out by generating a new matrix based on a percentage matrix and a chosen model. Then filters out all high confidence PSM matches with unique peptides at a number greater than two per protein. By uploading the new matrix into the Shiny app and choosing the channels for treatment and control, we can then move forward in the analyses. We analyze the noise with each channel through box-whisker plots. Each condition is then compared with a PCA plot, making sure that each channel within treatment and control does not contain any outliers. Volcano plots are produced based on significant fold changes and p-values. All plots and tables are saved for further evaluation.

After methods are applied for calculating and adjusting missing data in TMT Proteomics data, the file is further filtered with protein FDR confidence is high, unique peptides greater than 2, master proteins only, and no contaminants. Some of the graphs and tables produced include PCA plots, Volcano plots, and tables including all the statistics presented in the graphs. Applied here is a VSN normalization computed on the imputed matrix using a robust variant of the maximum-likelihood estimator for an additive-multiplicative error model and affine calibration. The model incorporates dependence of the variance on the mean intensity and a variance stabilizing data transformation. A linear model is fitted to the expression data for control and treatment, then *t*-statistics are computed by empirical Bayes moderation of standard errors towards a common value.

**Discussion**

A key note here is that many Proteomics experiments may have many to almost no empty abundance channels. While it may be difficult to decide whether to apply some sort of correcting model, the application and decision on which channels to fill are selected based on the percentage of missing values as well as the variability between valid technical replicates. The results should be comprised of either the original matrix or an improved matrix providing clearer PCA and Fold Change coverage with similar to lower p-values. Any abundance outliers or Fold Change based on a minimum noise level will be taken out of figures, providing clearer results to draw conclusions of whether particular Post-translational modifications have an effect on cell functions before and after treatment. Specific cases include O-GlcNAc sites from complex proteomes and computation of proteins of importance from these enrichment analyses of the glycoproteome followed by mass spectrometry (MS) analysis. Our method for Proteomics Expression Analysis determines custom workflows for different types of data for better interpretation of drug treatment results.