Subcellular Localization Prediction

Problem

For years a significant hurdle to understanding many biological questions has been the complexity of the systems that are attempted to be studied. Even when studying a single cell, there are hundreds of thousands of protein components interacting to produce an observed result. Mass spectrometry is a widely used technique that attempts to describe biological phenomena by identifying all of the proteins in a given reaction. This technique provides unparalleled insights into the protein components of biological events and has given rise to the field of proteomics (the large scale study of proteins).

The popularity of proteomic studies has led to an increase in protein discovery. However, little is known about these newly discovered proteins. Mass spectrometry analysis is able to identify the peptide sequence of these new proteins, but little else. In order to make inferences about the structure and function of these proteins computational tools are extremely valuable. Many tools have been created to predict the protein structure and identify repeated motifs. However the identification of more complex traits like function and cellular localization are only beginning to be explored.

Computational tools to identify the traits of a novel protein enable better hypothesis generation by researchers before performing experiments at the bench. More accurate hypotheses result in more efficiently designed experiments, saving researcher time and money on wasted reagents. The goal of this project is to identify the subcellular localization of a protein based only on the traits of its amino acid sequence. Knowing the subcellular localization of a protein will provide new hints to its function, as well as guide researchers to the best ways to design experiments around a novel protein.

Dataset

To generate this dataset the UniProt database was used. UniProt is an open access database that contains data on millions of proteins uploaded by researchers. These data include the primary amino acid sequence, structural data, subcellular localization data, and many other annotations describing a given protein. To mine this database the Uniprot Proteins REST API was used. This API allows for the request of up to 1,000 accession numbers (unique protein identifiers) at a time. As an initial quality control only Swiss-Prot reviewed (an internal UniProt validation) proteins were included in the mined data.

1. **Generate a list of Swiss-Prot reviewed human protein accession numbers.**
   The first step in the data wrangling process was the generation of a list of proteins to request information for. Using the UniProt REST Api for proteome (all proteins for a given organism), the accession numbers for all human proteins were requested. The conditions set were only human proteins that had been reviewed were included. This initial request generated a json file that was normalized into a list of around 20,000 strings containing Swiss-Prot reviewed, human protein accession numbers.

2. **Download all protein data for Swiss-Prot reviewed human proteins from UniProt.** Once the list of accession numbers to be mined was finalized, requests were made for each individual protein from the database. The UniProt API only allows for 1,000 requests to be made at a time and each request took around 15 minutes to load. To load the protein data the list of accession numbers was split into sublists, each containing 1,000 accession numbers. Individual requests were queued using the Jupyter notebook, and each request was saved locally into a .txt file.

3. **Create a list of dictionaries containing all of the protein data.** All of the .txt files were read back into python using the glob module. A loop was written that keyed each dictionary entry to the protein's accession number, and then updated a master dictionary that contains all proteins. A pandas dataframe was then generated from this dictionary.

4. **Unnesting protein data.** The data from UniProt was heavily nested. Where possible data were unnested and joined back into the main dataframe. When this was not possible, a new dataframe was created indexed to the accession number of the protein (e.g features dataframe). From these indexed dataframes counts were taken of key features, and merged back into the main dataframe. Rows containing NaN values were dropped.

5. **Analyzing the primary peptide sequence.** The column containing the primary amino acid sequence of the protein was sliced. Using ` regular expressions amino acid content of the sequences were performed.

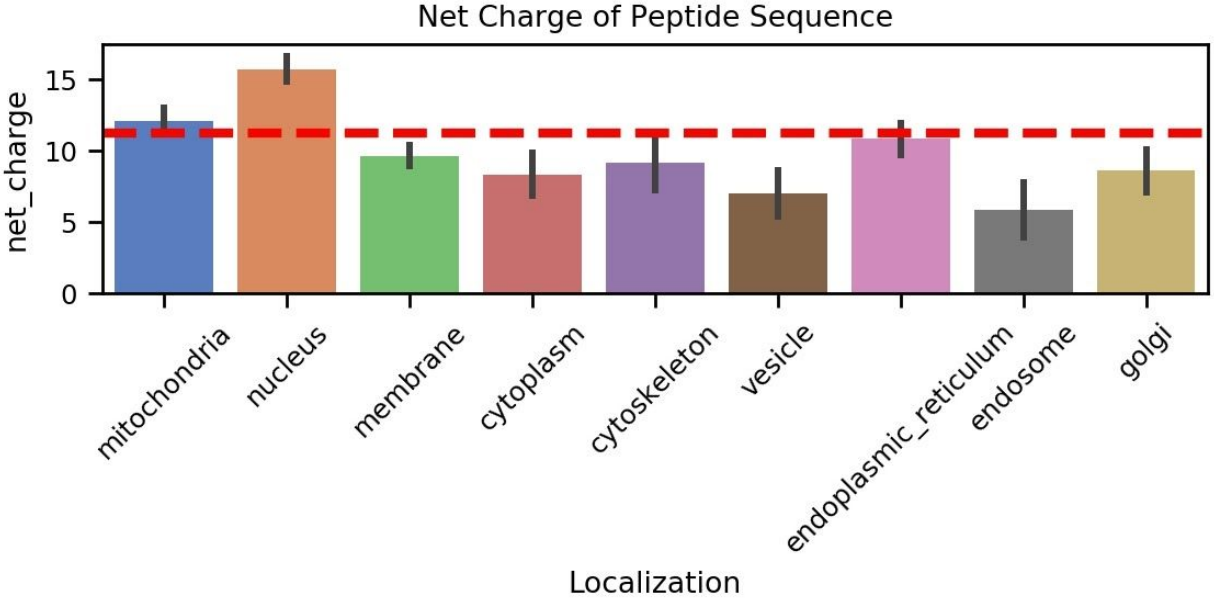
Because much of the data on UniProt is qualitative (i.e keywords pertaining to protein function), one-hot encoding was used to shape the dataset into an all numerical format. This process involved assigning True (1) or False (0) to a keyword if it is or is not associated with a given protein. Through this encoding scheme all of the data was made numerical[1], but at the expense of generating many features per protein. The final shape of the dataset was 12167 rows × 1017 columns.

Initial Findings

Many features of the primary peptide sequence appear to be predictive of the subcellular localization of a protein. Due to the large number of features per peptide included in this dataset picking which might be predictive of subcellular localization is difficult. For the initial data exploration features relating to the traits of the primary peptide amino acid sequence were focused on. To identify features of interest a consistent methodology was applied, described below in the context of examining the possible role of net peptide charge in protein localization.

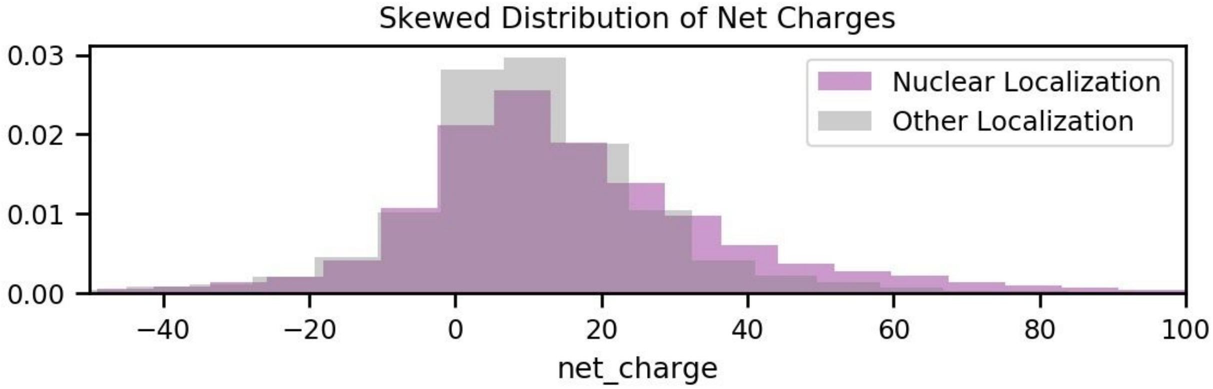
[1] The protein localization column was left categorical for ease of telling the data story. This column will be converted using one-hot encoding before machine learning is applied to these data.

1. Compare the mean net charge value between the different localization groups.



A bar graph was generated to visually compare the mean values between groups. The red dotted line represents the mean net charge value of the entire dataset. In this case the net charge of proteins localized to the nucleus appeared to be higher than average, and the net charge of proteins localized to vesicles and endosomes appeared to be much lower than average. Proteins localized to the nucleus were focused on because of their increase over the mean.

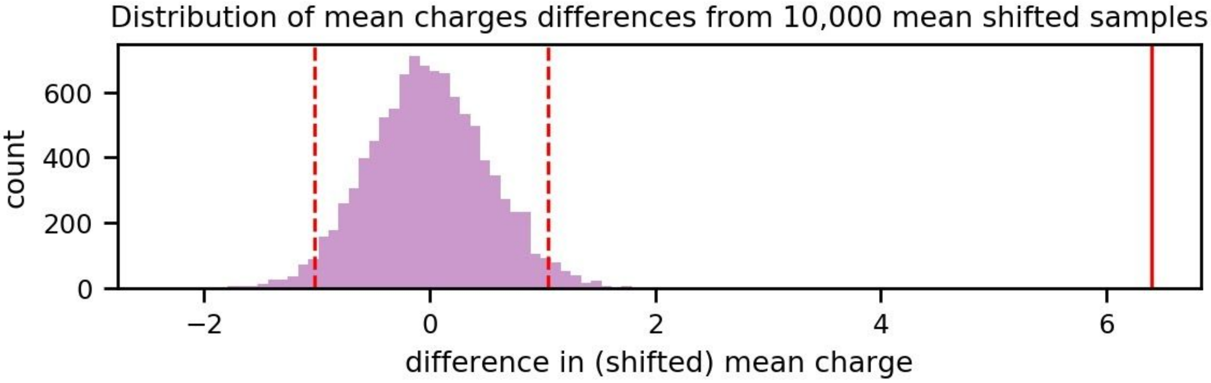
2. Next the overall distribution of the data was visualized with a kernel density normalized histogram comparing the distribution of the net charges of nucleus-associated proteins to that of non-nuclear proteins.. This type of histogram was used to show the trend in net charge normalized to the number of data points present, as there are different counts per localization.



With this graph it is clear that proteins that localize to the nucleus have a much wider distribution of net charges, skewing towards higher charges. All of the other localizations appear to have a closer to normal distribution of charges.

3. From this graphical exploration, a hypotheses were generated:
   H_0: There is no difference in net charge between nucleus-associated proteins and non-nuclear proteins.
   H_1: Proteins that localize to the nucleus exhibit a higher net charge than those that do not associate with this organelle.

4. To test these hypotheses bootstrapping was performed because of the relatively high number of samples: n=3686 for nuclear and n=8481 for non-nuclear, and the comparison that has to be made between normally and non-normally distributed data. The first step in this analysis was to shift both arrays to simulate that each array has the same mean. This was done by subtracting the mean value of each array by every data point in the array, and then add the mean value of both arrays to each value in the array. Bootstrapping was then performed 10,000 times to simulate many replicates of the experiment being performed for each array (nuclear and non-nuclear). These arrays were then subtracted from one another to generate the differences in the simulated data.



   This histogram represents the change that is observed from shifting the values of each array towards the mean of the whole array, bootstrapping to generate new samples and subtracting those samples to get the simulated difference. The dotted lines represent the 95% confidence intervals, and the solid line shows the observed change in charge. This shows that shifting each respective array toward the mean of the entire data no differences were observed as extreme has the observed data.

5. The p-value of these data was calculated by summing the number of simulated differences that were higher than in the observed set and dividing this number by the number of bootstrap replicates. In 10,000 replicates there was not a single instance of a difference greater than what was in the observed data, making the p-value zero.