Predicting Signal Peptides

Lab Notebook:

- 12/12/2017:

  Decided to choose the project topic from the given list of projects.
  Signal Peptide Classification seemed to be a reasonable choice as it was more aligned to a machine learning paradigm.
  Decided to familiarize with terms and problem before proceeding with the project.

- 13/12/2017:
  The data was taken from the course web: https://www.kth.se/social/files/547d95adf276544940c0ad5f/spdata.tar.gz and spent some time going over the data to get a feel of scheme of things at hand.

- 14/12/2017:
  After initial internet research, I think one needed to understand protein transport and targeting first.
  It seemed that creating good data features would be a challenging task. It also occured to me that HMM could be a useful tool in this problem.The notable work was Improved prediction of signal peptides by Bendtsen et al. This is the paper behind the SignalP service which is very popular. They use HMM and NN. They also trim the sequences to 60 AAs.

- 17/12/2017:
  After getting a better understanding of what signal peptides are and what they do. I came across a paper titled: Machine learning approaches for the prediction of signal peptides Link :http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.113.3288&rep=rep1&type=pdf and other protein sorting signals Authors:Nielsen e.t.al The paper focussed on a HMM and NN model. The best approach seemed to be a hybrid NN-HMM. I found that windowing that was used to create feature for the hidden markov model was a bit complex.. Classification will be performed by scoring the sequence using each of the models. We decided against HMMs.

- 20/12/2017:
  Created a feature space using Scikit learn's K-gram vectorizing followed by Feature Hashing method. his can be found here :http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html

- 22/12/2017:
  After realising that the feature space that I had created the day before was very sparse - Since of K-grams do not appear at all in some sequences. I decided to form a dimensionality reduction to a 2D space using SVD using `TruncatedSVD()` function in sklearn.

- 23/12/2017:
  A set of classical classifiers available over sklearn were chosen to begin our experimentations. Classifiers considered:

    - KNN
    - SVC -With Linear Kernel
    - SVC -With Radial Basis Function Kernel
    - Decision Tree
    - Random Forest
    - Adaboost
    - Multi-Layer Perceptron

- 27/12/2017:
    - Plotting functions were made, where each one of the classifiers were plotted with its accuracy alongside its decision boundary. Confusion matrices were also plotted as to better understand the results. The day was spent trying to empirically discover the arguments that would lead to the best accuracy for finding the signal peptide, for both transmembrane and non-transmembrane samples where we found that MLP and SVC- rbf seem to be slightly better than the others.
    - Performance on transmembrane data was similar but however the was a marked increase in false negative rate in MLP as the data was biased towards the negative in the transmembrane data set this can be seen below. The ensemble methods such as adaboost and random forest better managed this bias.

- 28/12/2017:
    - Visualizing signal peptides using weblogo. I decided to take one sequence file from the positive examples and compare it with one of the negatives.I found out that the A and L proteins affected the signal peptides. I also removed the first amino acids as it made visualisation difficult, it was too heavily loaded in all files.

- 29/12/2017:
    - Started early with task of implementation of the trained models on new proteomes. Initially there were some doubts as to which proteomes where right ones - which filters to put on biomart. After some research, I decided as per the information available on the ftp site for biomart http://www.ensembl.org/info/data/ftp/index.html. The species I chose were Homo Sapien and Mus musculus.

- 2/1/2018:
    - Implemented the code to get predictions on the new proteomes. Initially I had incorrectly created new vectorizers for the new datasets. I corrected this after some errors from the predict() methods. There were some things that I needed to consider when I was writing this code - first I made the mistake of creating new n-grams instead of using those created in training - this created runtime dimensionality errors.

    - Results Obtained:
        -

Classifier	Predicted Peptides Signal
SVC-RBF	41710
KNN	46124
SVC Linear	30143
Decision Tree	45830
Random Forest	48916
MLP	38492
AdaBoost	49505

- The results seems to be a bit biased upwards compared to other sources where they found only about 20000 proteins that contain a signal peptide. Since the training data a biased towards those that had signal peptides this is expected.

- 5/1/2018:
    - Wrote the the findings in the report including the weblogo difference in the sequences. Observation: The sequence logos are very sparse in the sequences without signal Peptides and for sequences with signal peptides tend to have a high densities of amino acids Lysine (L) and Alanine (A) in posItions 5 to 20. Sequences without peptides tend not have any dominant amino acids.

- 6/1/2018:
    - Combing all the results and experiments and started compiling reports.

- 10/1/2018:
    - Finished compiling reports.