Title: SVM buried versus exposed

## Introduction

Support vector machine (SVM) is a supervised machine learning technique which means that data for training is supplied with correct classification. It is non-parametric so there is no assumption on how the data is distributed. The algorithm is trying to find the hyperplane to separate training examples and in turn, assign them to different classes with the highest possible accuracy. The number of classes is defined by user beforehand(Mountrakis, Im, & Ogole, 2011). One of easily accessible ways to use SVM is through scikit-learn library(Varoquaux et al., 2015). There are 4 main types of kernels in SVM provided by scikit-learn: linear, polynomial, rbf and sigmoid. These kernel types define the function used for separating hyperplane determination. Different kernels might give the best results depending on the underlying problem one is trying to solve.

Assessing the accuracy of the model is important for getting an estimate of its capabilities and also for model selection. One way of accuracy assessment is cross-validation(Kohavi, 1995). In this technique, the training database is split into training set on which algorithm is trained and test set which is only used for testing the performance of the model. This approach allow for generalization the accuracy of the model, as test set is simulating real problem(need citation).

Trans-membrane β-barrels are transmembrane proteins formed by antiparallel β-sheet forming a barrel shape structure. Two neighbour residues inside in this β-sheet are always pointing in the opposite directions forming the in/out pattern with reference to the centre of the barrel. The residues pointing outside will always be nonpolar, as they are facing nonpolar inner part of transmembrane protein and residues pointing inside will always be polar(Zvelebil, M., & Baum, 2007). There are very little β-barrels structures experimentally solved so far. Predicting transmembrane regions is also difficult since they lack recognizable features such as stretch of 15-30 consecutive hydrophobic residues or positive inside rule present in helical transmembrane protein(Singh, Goodman, Walter, Helms, & Hayat, 2011). Prediction of the exposure status of residues (buried or exposed) is important because of its possible applications in side-specific mutational studies and in channel engineering(Singh et al., 2011).

There are several different approaches for prediction the exposure status of Trans-membrane β barrel residues.

For SVM its around 75-80% - best approach seems to be HMM – paper references

TMBHMM – 83%

## Methods

* Dataset

The dataset used in this project consisted of 69 transmembrane β barrel non-homologous proteins. For each protein, the exposure status in given position was provided. The dataset was organized in repeating three line pattern: protein ID, protein sequence and exposure status in separate lines for each entry.

* Extracting features from dataset

Task is to develop a SVM model and optimize it by in example checking the optimal window length and kernel and addition of more information i.e. evolutionary info to further increase its accuracy also to compare our SVC model with other stuff like random forest and simple decision tree to further asses its performance

Because they lack the pattern, not many proteins are known and no positive inside rule

- Exposed vs Buried

It is important for reasons stated in the article I previously read

- Some examples of best currently available with their accuracies

For SVM its around 75-80% - best approach seems to be HMM – paper references

Methods:

introduce SVM

And some python PSSM stuff how it works

Results:

Comparison of PSSM with no PSSM

Comparison different kernels

Comparison of different window lengths

Comparison of the SVM with random forest and single decision tree methods

Things that did improved the linear SVM

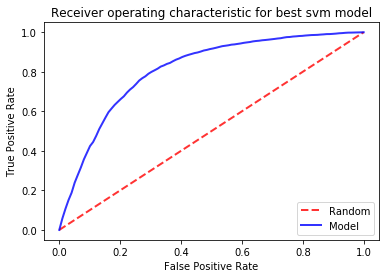
Measurement of the Accuracy ROC curve MCC maybe

How our results correspond to the currently available predictor and also how SVM does in general compared to hmm

Conclusion:

What was achieved and how my predictor does what might be possible applications

## Results and discusion



## Conclusions