Feature importance project

Proteins have various important features. In this study, we have chosen 14 different features namely Cov441, Pre441, Pre231, Cov231, aa_trROS, frobnorm_trROS, seqpro_trROS, ss_pdnet, pssm_pdnet, sa_pdnet, entropy_pdnet, ccmpred_pdnet, freecontact_pdnet, potential_pdnet and tried to rank the features based on their importance. The importance here is a measure of predictability based on protein contact maps.

Task A: Creating Pre231 and Cov231 features for 3456 proteins of the Deepcov dataset

Background

Cov441, one among the primary features generated using DEEPCON architecture and model is one among state-of-the-art contact map prediction. Pre441 was generated using DEEPCON_PRE that uses DEEPCON architecture and precision features generated by incorporating RESPRE's ridge regression model on DEEPCON's data pipeline.

Goal

The volume of 441 features occupies expensive SSDs and longer training times. To increase the efficiency, it is needed to reduce the training times without losing the accuracy in the prediction. The covariance matrices by their nature is symmetric along their diagonals. So, if the upper triangle values are added with the lower triangle values, the information conveyed in a matrix is retained and the channel size is reduced to 231. This idea is stated in equation 2 (page 4) of the paper "Protein contact prediction using metagenome sequence data and residual neural networks by Qi Wu1, Zhenling Peng2,*, Ivan Anishchenko3,4, Qian Cong3,4, David Baker3,4 and Jianyi Yang 1"

This idea has been incorporated in to DEEPCON's data pipeline and thus pre231 and cov231 features were generated.

The code for converting 441 channels to 231 channels for the protein is attached here.



compress441to231_pre.txt



deep_cov_matrix_from_aln_1_2forloop.txt

Pre231 and Cov231 features were generated for 3456 proteins in Deepcov dataset. Thus, the goal was achieved

Task B: Studying the Feature Importance

Background

The intent behind this task was to see which feature contributes the most to the model accuracy and rank the importance of features accordingly.

Sub Task 1: Create a model comprising of all the features

The intent was to create a huge model comprising of all the important features.

So, for this first a model was created that was trained using all the feature channels (1022) using 3456 proteins of the Deepcov dataset and validated using the first 130 proteins of the Deepcov dataset.

Model of 1022 channels included these features i.e., 441 channels of precision features(pre441), 441 channels of covariance features(cov441), 55 channels of PDNET features, 85channels of trRosetta features.

The code for training and the corresponding log is attached here.


train.all.txt


allfeatures_2420.log.txt

Subtask 1 is complete.

Sub Task 2: Creating Noise/mean matrices

The model created in Subtask1 has the average precision accuracy of

Top L/5 79.19 Top L 53.52 NC 50.07. So, in this sub task the intent is to calculate the Noise/Mean matrix.

Every feature is in the shape of [L, L, X], where X denotes the dimension of the feature and L denotes the length of the protein chosen. For e.g. X = 441 for the covariance/precision feature. To calculate the Noise/Mean matrix for the different features, the top 200 out of 3456 proteins from Deepcov dataset was selected based on max L criteria and mean of those features were calculated. Mean was calculated using the numpy method nanmean. Here we initialize the mean matrix with np.nan to overcome the ‘zero’s affecting the average problem’, since every chosen protein (among the 200) will be of varying lengths, so when a value at a particular place in a sequence is zero, this could create a bias in a particular place in a protein sequence.

Sample Code for Mean matrix pre231



Avg_pre231.txt

Similarly, Noise/Mean matrices were created for pre441, cov441, pdnet55 and trROS features.

[Separately pre231/cov231 mean matrix was also created and a separate model for 231+231 = 462 features was created to test which among these two features did perform better. This was extra not pertaining to the original 1022 model. Surprising finding was that the exact same loss in precision was observed when both covariance and precision matrices was altered].

Thus, six different mean matrices were created.

Subtask2 is complete.

Sub Task 3: Blurring/Zeroing of features

The generated model has 1022 channels. To find out how the model performs when every individual feature was masked separately was tested. So, for e.g. if the intention is to see how pre441 feature performs, when we feed the pre441 features as input while predicting the accuracy, in the place of pre441 features the pre441 mean matrix is substituted 441 times.

Similarly, to assess the importance of each feature, every feature was blurred individually at a time using their respective noise matrices and prediction accuracy was calculated.

The following table enumerates the feature importance when noise matrices were used

Feature altered	Precision after replacing the feature with a noise (mean) matrix		
	L5	L	NC
Nothing (all 1022 channels intact)	79.19	53.52	50.07
pdnet (secondary structures)	26.48	15.45	14.55
trRos (sequence profiles)	38.52	21.6	20.11
pdnet (solvent accessibility)	52.35	32.29	29.76
pdnet (pssm)	59.41	37.2	34.02
pdnet (entropy)	60.11	37.39	34.95
trRos (aa composition)	66.72	41.58	38.3
pdnet (ccmpred)	67.55	44.2	40.89
precision matrix	74.51	48.72	45.35
covariance matrix	72.31	49.17	46.12
pdnet (freecontact)	77.58	51.44	48.23
trRos (frob. Norm)	78.53	52.54	49.19
pdnet (potential)	79.14	53.46	49.89

Our results show that one-dimensional features such as secondary structures, sequence profiles, and solvent accessibility are critically important for a deep learning method if we have them as input.

To cross validate the findings we replaced zero matrices in the place of noise matrix and did the same predictions again. The results are given below.

Feature altered	Precision after replacing the feature with 0s		
	L5	L	NC
Nothing (all 1022 channels intact)	79.19	53.52	50.07
pdnet (secondary structures)	44.93	26.28	24.55
pdnet (solvent accessibility)	50.64	31.23	29.06
trRos (frob. Norm)	58.8	33.89	31.26
pdnet (pssm)	56.32	34.61	32.22
pdnet (entropy)	58.09	36.38	34.34
pdnet (ccmpred)	67.53	44.31	40.89
trRos (aa composition)	69.27	44.72	41.18
trRos (sequence profiles)	69.94	45.63	42.1
precision matrix	72.65	47.14	43.62
covariance matrix	74.6	50.1	46.93
pdnet (freecontact)	77.62	51.64	48.46
pdnet (potential)	79.11	53.49	49.88

Surprisingly, even after zeroing the features the model still performs reasonably well.

Subtask3 is complete.

Excel attachments


feature importance results template.xlsx


Feature_Importance _Ranking_blur+zero


Validation dataset comparison_Noise n


Validation dataset comparison_Zero ma

Future Work

Feature importance in distance prediction. [Googledoc maintained by Dr.Badri Adhikari. The manuscript sections written by Dr. Badri Adhikari is the future work]

References to code in prayog09 server

Code for 441-231 compression: /nvme2tb/nachammai/homedir/PDNETpre/

Code for training with 1022 channels and location of hdf5 files:
/nvme2tb/nachammai/pdnet-test/

Code for mean matrices and noising/blurring of matrices:
/nvme2tb/nachammai/FeatureExtraction/