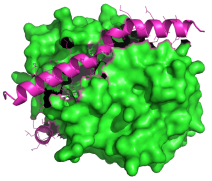
Project Specification

Structural Bioinformatics
2019

(9 May 2019)

Identification of Linear Interacting Peptides in PDB structures



Linear Interacting Peptides (LIPs) are intrinsically disordered regions (IDRs) that fold when interacting with other proteins or DNA/RNA molecules. In the PDB (https://www.rcsb.org/) LIPs are always found in complex with other molecules and can be easily recognized visually because retaining a certain degree of structural linearity. Compared to globular regions, LIPs are characterized by a larger surface of interaction and a larger fraction of inter-chain over intra-chain contacts. An example is PDB ID 2O8A, chain I, residues 12-17/44-56/137-165.

- Each team of student is requested to develop a software that calculates the propensity of a residue to be a LIP from a PDB protein complex.
- The software will be implemented in Python (2.7.x or 3.6.x). The use of the BioPython.PDB, Argparse and Logging modules is strongly encouraged.
- The software has to be documented extensively both inside the code and in an external file by including algorithm description, user guide, usage and instructions to re-train the parameters. Documentation will be generated using Markdown language (https://daringfireball.net/projects/markdown/).
- The training procedure has to be fully reproducible. The team will provide the necessary code, data and documentation.

Input
A PDB file in the PDB format.
Optional parameters: configuration file with algorithm parameters, output directory.

```
$ python lip_predictor.py -configuration parameters.ini -out_dir results 1jsu.pdb
```

Output
A text file in the following format.

```
>pdb_id
<residue_id (model/chain/position/insertion_code/name)> <LIP score> <LIP class>


>1jsu
0/C/25//K 0.343 0
0/C/26//P 0.453 0
0/C/27//S 0.574 1
0/C/28//A 0.615 1
0/C/29//C 0.831 1
0/C/30//R 0.808 1
0/C/31//N 0.747 1
0/C/32//L 0.304 0
0/C/33//F 0.233 0
```

The LIP score will be in the range 0.0-1.0, with 0.5 representing the threshold to switch between LIP/non-LIP classes (1.0 means LIP; 0.0 means non-LIP). The LIP score will have no more than 3 decimals.

LIP score implementation
The implementation can be rationalized in two steps: calculation of (relevant) residue features and training.

Features
Some residue features are provided by external tools like DSSP which provides the accessible surface area (ASA) and the secondary structure. Other features, like the ratio of inter- and intra-chain contacts, will be implemented.

- ASA. Can be used as it is provided by DSSP.
- Secondary structure. Assign 1.0 when DSSP predicts any secondary structure, otherwise 0.0.
- Inter-/intra-chain contacts ratio. Different alternatives can be tested, for example separated counts for main-chain contacts and side-chain contacts or different types of contacts (ionic, vdw, H-bonds, etc.), identified based on the type of interacting atoms. Contacts parameters can be found here http://protein.bio.unipd.it/ring/about. The RING software can be used to calculate contacts.

Both secondary structure and the contacts ratio can be further processed by calculating a moving average (sliding window). For example, calculate the inter-/intra-chain contact ratio over a small window and assign the result to the central residue. The contact ratio can be re-scaled to lie in the 0.0-1.0 range.

Training
Each feature can be considered a different LIP predictor and probably have significantly different performance/accuracy. The training will consist in two parts: identify optimal parameters for the features and perform an ensemble average (https://en.wikipedia.org/wiki/Ensemble_averaging_(machine_learning)).

- Single feature training. For each feature calculate the confusion matrix and relevant indexes (balanced accuracy, specificity, precision, sensitivity, F-score, AUC, MCC, ...). For the contact ratio feature, perform a grid search to optimize the window size, contact thresholds, etc. Also, try alternative implementations of the ratio considering the type of contacts, main/side-chain, etc.
- Ensemble average. Calculate a new ensemble score from the weighted sum of the score for the single features. The weights should be proportional to the accuracy (or other indexes) calculated previously and the sum of all weights should be 1.

Parameters will be optimized against a training set provided by the teacher. A ten fold cross-validation is recommended.

Project evaluation
The algorithm will be blindly tested against a validation dataset not available to the students. Clarity of the documentation and the reproducibility of the training results will be evaluated along with the performance of the predictor which should perform significantly better than random.