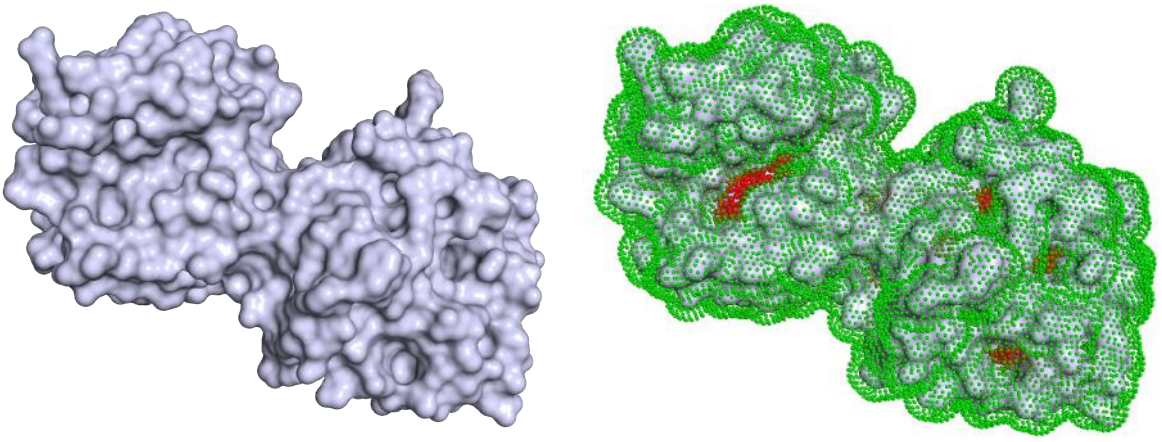
NPFL054 Introduction to Machine Learning

Charles University, November 2018

Protein Ligandability Recognition – task description

A) Brief description of the machine learning task and the provided data set

You will do the task of *Protein Ligandability Recognition*, which comes from the field of bioinformatics. Proteins perform their biological functions namely by their interaction with other molecules such as ligands, other proteins etc. A ligand is a small molecule that specifically binds to a larger one. There exist a number of points on protein surface where ligands might bind. The figure below to the left visualizes a protein and below to the right the red points on the protein's surface binding some ligands.



You should build an automatic predictor that, using a provided development data set of points, should be able to predict for the points on the protein's surface either their ligandability, i.e. ability to bind ligands (a binary classification task), or their distance to the closest ligand (a regression task).

You will be provided with a development data set of points on surface of selected proteins. Each point belongs to just one particular protein. Also, a blind test data of points on different proteins will be provided without true predictions. Each example in the data is a feature vector, which represents physico-chemical properties of a point lying on the surface of a protein. A brief description of the features can be found in the posted document `plr.attributes`.

B) General remarks on working with data and methods

(1) You will do either regression, or classification task. We provide three different data sets (A, B, C) for classification, and one data set (D) for regression.

(2) Important note on `protein_id` column: Data points should be always split by proteins. This means that when you are splitting development examples into train and test subsets, you should first decide which proteins (identified by `protein_id`) go to which subset and move all data points of those proteins there. The same applies to splitting data sets into folds in cross validation.

If you randomized all datapoints without taking `protein_id` into account, you would get unrealistically good results. In other words, your generalization error estimate would be biased and would seem too good. In the end, you will do predictions for new, unseen proteins!

Do not use `protein_id` as a feature for training!

(3) This note is relevant only to the regression task. Data set D contains examples with `ligand_distance` value for regression experiments. Ligand distance is the distance from the point represented by a given example to the closest ligand atom measured in Angstrom (A).

(4) Recommendation: First, look at the data and find how many proteins appear in your data set. Then:

- When doing regression – For each protein, make a boxplot with visualisation of the distribution of `ligand_distance`.
- When doing classification – Compare the proportion of the binary output value `class` within each protein. Print a summarising table.

C) Technical hints

- To load the data sets quickly into the memory use `fread()` (R package `data.table`).
- To learn your predictors use the following R packages: `rpart` for Decision Trees, `randomForest` and `adabag` for ensemble learning, and `e1071` for SVM.
- Variable importance values are computed by methods implemented in the mentioned packages:
    - `importance()` function extracts variable importance measures as produced by randomForest
    - `boosting()` function directly gives the importance values in `model$importance`