# Instructions for applying Protein Block prediction models

## Preparation of input data for PBs prediction

### Generate tool/service outputs on which PBs prediction models depend

To apply PBs prediction models, it is necessary to generate outputs using the following tools/services:

* Spider3   
  <https://sparks-lab.org/server/spider3/>
* MassPred for disorder predictors listed in the paper except for VLXT <http://bioinfo.matf.bg.ac.rs/home/downloads.wafl?cat=Software&project=MassPred>
* PONDR VLXT predictor   
  <http://www.pondr.com/>
* DN repeats and IN repeats with minimal length 2 in amino acid sequence <http://bioinfo.matf.bg.ac.rs/home/downloads.wafl?cat=Software&project=RepeatsPlus>

It is required to save outputs of the listed tools/services per protein chain (with id *protein\_id*) in files using the following format for names:

* Spider3: *protein\_id*.spd33
* MassPred: *protein\_id*.disorder.txt
* PONDR VLXT: *protein\_id*.txt
* DN repeats: *protein\_id*.dn.txt
* IN repeats: *protein\_id*.in.txt

Obtained files must be saved in directories with the following names:

* Spider3: **spider3**
* MassPred:**disorders\_masspred**
* PONDR VLXT: **pondr**
* DN repeats: **dn\_repeats**
* IN repeats: **in\_repeats**

Required directories and files for chains given as examples are in the directory **example\_protein\_data**.

## Data preparation for PBs prediction models based on fixed-window data format

For merging outputs of listed tools/services and preparation data in the required format for PBs prediction models based on the fixed-window data format, use file **merge\_protein\_info\_fix\_window\_data.py** with command

python merge\_protein\_info\_fix\_window\_data.py *path\_to\_data*

Argument *path\_to\_data* is a path to a directory that contains all required directories with input files. If *path\_to\_data* is not specified, the required directories are considered to be in directory **example\_protein\_data**.

Chains without disorder regions or repeats do not need to have corresponding files for disorder regions or repeats.

The result per protein chain will be saved in a file with the name ***protein\_id*\_for\_pbp.csv** in a directory **data\_for\_pbp**.

## PBs prediction models developed using Python packages

Python packages used in the development of PBs prediction models are:

* Keras, version 2.3.1
* TensorFlow, version 1.13.2
* pandas, version 1.1.4
* scikit-learn, version 0.23.2
* NumPy, version 1.18.5
* pickle
* joblib, version 0.16
* pypmml, version 0.9.11

### PBs prediction models based on the fixed-window data format

For the PBs prediction can be used two models constructed using the scikit-learn package:

* **nn** which was developed using MLPerceptron and ratio 50:50 for partition on training and test parts
* **rf** which was developed using RandomForest and ratio 50:50 for partition on training and test parts

Directory **scklearn\_models** contains material for using of obtained models:

* subdirectory **models** contains obtained models
* subdirectory **apply\_on\_training\_test\_part** contains training and test parts and scripts for application of the obtained models on training and test parts. To apply models on training and test parts, use script **rf\_apply\_on\_training\_test\_data.py** for **rf** model, andscript **nn\_apply\_on\_training\_test\_data.py** for **nn** model. As a result, the calculated accuracy of each part will be shown and corresponding confusion matrices will be saved as HTML files.
* script **apply\_on\_chain.py** isfor PBs prediction for a single protein chain. Both models will be applied. The script can be called with a specified path to the directory with prepared data of chains for prediction. Information of one chain must be in a CSV file whose name has suffix **\_for\_pbp** and prepared as described in *Data preparation for PBs models based on the fixed-window data format*. If a path to a directory isn't specified, the current directory is searched for files with chain information.

python apply\_on\_chain.py path\_to\_dir\_with\_chains\_info

The output for each chain file found in the directory is in the format:

>ALG *model\_label*  
 >PROT\_ID *protein\_id*  
 >AA *amino\_acid\_sequence*   
 >PB *sequence\_of\_predicted\_protein\_block*s

where *model\_label* is rf or nn.

### PBs prediction models based on the sequence data format

Two LSTM-BRNN models constructed using Keras and TensorFlow packages can be used for the PBs prediction:

* **seq\_model\_subsets** which was developed using partition based on sequences (for details see the paper):
  + *subset1* which was used as the training part
  + *subset2* which was used as the test part
* **seq\_model50** which was developed using a partition on trаining and test parts with a ratio of 50:50

Directory **seq\_models** contains material for using obtained models:

* subdirectory **models** contains obtained models
* subdirectory **apply\_on\_training\_test\_part\_subsets** contains corresponding files of *subset1* and *subset2* and script **apply\_seq\_model\_on\_subsets.py** for application of the model **seq\_model\_subsets** on each part. As a result, the calculated accuracy of each part will be shown and corresponding confusion matrices will be saved as HTML files.
* subdirectory **apply\_on\_training\_test\_part\_50** contains files of training and test parts obtained with a ratio of 50:50 and script **apply\_seq\_model\_on\_partitions50.py** for application of the model **seq\_model50** on each part. As a result, the calculated accuracy of each part will be shown and corresponding confusion matrices will be saved as HTML files.
* script **seq\_apply\_on\_chain.py** for PBs prediction of a protein chain. Both models will be applied. The script can be called with a specified path to directory with outputs from required tools/services on which PBs prediction models depend. Output files of tools/services must be in subdirectories as described in *Preparation of input data for PBs prediction*. If a path to a directory isn't specified, the directory **example\_protein\_data** is used.

python seq\_apply\_on\_chain.py path\_to\_dir\_with\_chains\_info

The output for each chain file found in the directory is in the format:

>ALG *model\_label*  
 >PROT\_ID *protein\_id*  
 >AA *amino\_acid\_sequence*   
 >PB *sequence\_of\_predicted\_protein\_block*s

where *model\_label* is seq50:50 or seq\_subsets.

## PBs prediction models developed using IBM SPSS Modeler

IBM SPSS Modeler, version 18.2.1, (also trial and academic version) can be downloaded from <https://www.ibm.com/products/spss-modeler>.

Directory **spss\_models** contains material for the usage of models obtained by IBM SPSS Modeler. Material corresponding to the best-obtained model of each used algorithm is in a separate subdirectory with name in a format apply\_*suffix.* *suffix* corresponds to a used algorithm for building the model:

* *PBC5.0d* for C5.0
* *bayes\_net* for Bayes Net
* *cart* for CART
* *ex\_chaid* for Exhaustive CHAID
* *quest* for QUEST
* *nn\_mlp* for Multilayer Perceptron
* *xgboost* for XGBoost Tree
* *rt\_without\_cost* for Random Trees

Subdirectory of each algorithm contains material for the usage of the corresponding model:

* stream **apply\_on\_partitioned\_data\_*suffix*.str**, and stream **apply\_on\_partitioned\_data.str** for application of the PBs prediction model on a specified data part.

In IBM SPSS Modeler Client, a user must specify a path to a part file in the *input* node in the stream. If IBM SPSS Modeler in batch mode is used, the user can execute script

* + **apply\_on\_training\_data\_*suffix*.txt** with command  
    clemb @apply\_on\_training\_data\_*sufix*.txt  
    for PBs prediction on the training part
  + **apply\_on\_test\_data\_*suffix*.txt** with command  
    clemb @apply\_on\_test\_data\_*sufix*.txt  
    for PBs prediction on the test part
  + **apply\_on\_validation\_data\_*suffix*.txt** with command  
    clemb @apply\_on\_validation\_data\_*sufix*.txt  
    for PBs prediction on the validation part

As a result, a report in a form of HTML file with the calculated accuracy and a confusion matrix will be generated.

* stream **call\_for\_one\_chain\_*suffix*.str** for PBs prediction of a specified protein chain. The stream can be executed using the GUI of IBM SPSS Modeler Client or IBM SPSS Modeler in batch mode using the clemb command. The input to a stream is a chain file that must be prepared as described in *Data preparation for PBs prediction models based on the fixed-window data format*.

In IBM SPSS Modeler Client, a user must specify a path to a chain file in the input node in the stream. If IBM SPSS Modeler in batch mode is used, the user can execute script **call\_for\_one\_chain\_*suffix*.txt** with command

clemb @call\_for\_one\_chain\_sufix.txt

Path to chain file must be specified via a parameter *-Pinput.full\_filename* in a script.

The output for specified chain file as input is in the format:

>ALG model\_label

>PROT\_ID protein\_id

>AA amino\_acid\_sequence

>PB sequence\_of\_predicted\_protein\_blocks

Subdirectory **dataset** contains files of training, validation and test parts (training\_part.csv, test\_part.csv, validation\_part.csv) used during PBs model development.

## PBs prediction model developed using IBM Intelligent Miner

The model developed using IBM Intelligent Miner and SPRINT algorithm was exported using PMML format.

Directory **im** contains model and

* script **apply\_on\_training\_test\_part**.**py** for application of the PBs model on training and test parts. As a result, the calculated accuracy of each part will be shown and corresponding confusion matrices will be saved as HTML files.
* script **apply\_on\_chain.py** for PBs prediction for single protein chain. The script can be called with a specified path to the directory with prepared data of chains for prediction. Information of one chain must be in a CSV file whose name has suffix **\_for\_pbp** and prepared as described in *Data preparation for PBs prediction models based on the fixed-window data format*. If a path to a directory isn't specified, the current directory is searched for files with chain information.

python apply\_on\_chain.py *path\_to\_dir\_with\_chains\_info*

The output for each chain file found in the directory is in the format:

>ALG SPRINT

>PROT\_ID protein\_id

>AA amino\_acid\_sequence

>PB sequence\_of\_predicted\_protein\_blocks

* files of training and test parts (training\_part.csv, test\_part.csv) used during PBs prediction model development.