GraphProt2: A graph neural network-based method for predicting binding sites of RNA-binding proteins Supplementary Material

Michael Uhl, Van Dinh Tran, Florian Heyl, and Rolf Backofen

July 24, 2020

Supplementary methods

Program modes

GraphProt2 is logically split into five different program modes: training set generation, prediction set generation, model training, model evaluation, and model prediction. Here we briefly introduce their functionalities. For a complete and up-to-date description of available options and parameters, please refer to the GitHub documentation.

Training set generation

This mode (`graphprot2 gt`) is used to generate a training dataset from a given set of RBP binding sites, which can be sequences, genomic regions, or transcript regions. In case of sequences, negative sequences can be supplied or generated by k-nucleotide shuffling of positive sequences. In case of regions, negatives can be supplied or selected randomly from gene or transcript regions containing positive sites. The number of generated negative sequences can be further specified, as well as the regions from which to extract them. Output site lengths can be of variable or fixed size, and various filtering options are available to filter the sites by score, sequence complexity, region, or length. Depending on the input type (see Table 1), different features can be selected for annotation of positive and negative regions. Annotations are taken from the input GTF file, while sequences are extracted from the input .2bit genomic sequence file. An HTML report can be generated, providing statistics and visualizations to compare the positive with the negative set. The dataset is stored in a folder which forms the main input to the model training mode.

Prediction set generation

The prediction set generation mode (`graphprot2 gp`) resembles the training set generation mode, but instead of generating a training set containing positives and negatives, it generates a prediction set from a given set of sites or sequences to predict on. Just like with `graphprot2 gt`, sites can be filtered and different feature annotations can be added depending on their input type. The set is again stored inside a folder, which acts as the main input to the model prediction mode.

Model training

After generating a training set, a model can be trained on the dataset in model training mode (`graphprot2 train`). By default, all features of the training set are used to train the model, but specific features can be selected as well. Cross validation is used to estimate model accuracy, and hyperparameter optimization can be enabled by providing value lists for each hyperparameter to test. Current hyperparameters that can be changed are: gradient descent batch size, number of training epochs, patience, learning rate, weight decay, fully connected layer and hidden layer node feature dimensionality. After the optimal hyperparameters have been determined (in case of several combinations, otherwise set parameters are used), a final model is trained with these parameters and output in a new folder, which serves as input to the model evaluation and model prediction modes. For the models used to create the benchmark results, we set the following default hyperparameters: patience=10, batch size=50, epochs=100, fully connected layer and hidden layer node feature dimensionalities=128, learning rate=0.0001, weight decay=0.0001.

Model evaluation

This mode (`graphprot2 eval`) is used to visualize binding preferences of the model trained with `graphprot2 train`. Sequence and additional feature logos of various lengths can be output, as well as position-wise scoring profiles for a user-defined subset of training sites (see Figure 4 and Supplementary Figure 1 for visualization examples).

The support of variable-sized inputs (i.e., graphs of variable size) allows GraphProt2 to calculate position-wise scoring profiles using a sliding window approach. Each window corresponds to a subgraph, which is scored by the model, and the score is assigned to the center (backbone) position of the graph. To generate a logo, GraphProt2 extracts top-scoring positions from a specified number of top scoring positive profiles, and extends them to a defined logo length. The extracted subsequences are then converted into a weight matrix and plotted with Logomaker [2].

Model prediction

Model prediction mode (`graphprot2 predict`) is used to predict whole binding sites as well as position-wise binding profiles for a given set of sequences, genomic sites, or transcript sites. The prediction dataset needs to be generated by `graphprot2 gp` beforehand, as well as the model which needs to be trained through `graphprot2 train`. Note that the same input type and set of features needs to be used to generate the model and the prediction set. In case of whole site prediction, a list of input sites with associated scores is output. In case of profile prediction, position-wise binding profiles are generated, and high-scoring sites within the profiles are extracted and output together with the profiles.

Supplementary figures

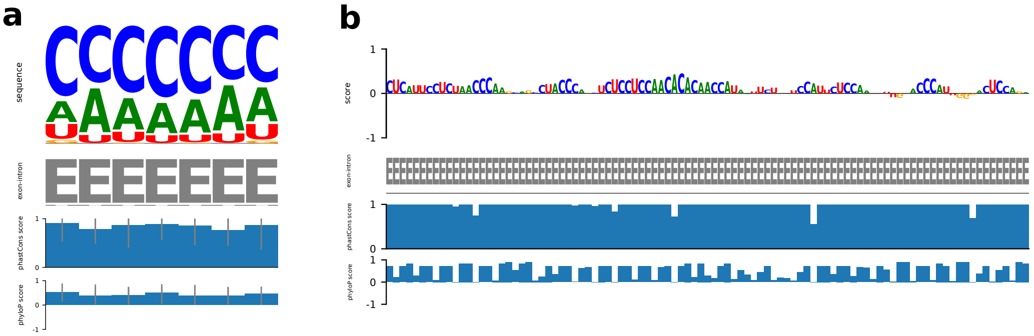

Figure 1: GraphProt2 feature logo and top profile for the IGF2BP1 model, trained with the following features: exon-intron annotation, phastCons and phyloP scores. (a) GraphProt2 feature logo, with one subplot for each model feature. (b) GraphProt2 top scoring profile, with one subplot for each model feature. Logo and profile (preference for CA repeats and exonic regions) is in agreement with described binding preferences of IGF2BP1 (CACA as one of the main recognition motifs and 3'UTR binding) [1]. Top 500 scoring profiles were used to generate the logo. A window size of 5 was used to calculate the profiles.

Supplementary tables

Table 1: Key attributes of GraphProt2 compared to GraphProt and current CNN-based methods.

Attribute	GraphProt2	GraphProt	CNN-based methods
Model architecture	GCN	Graph kernel + SVM	CNN
Additional nucleotide features	YES	NO	YES
Built-in profile prediction	YES	YES	NO
Variable length input	YES	YES	NO
Base pair annotation	YES	YES	NO

Table 2: Single model benchmark results over 30 individual RBP eCLIP sets for GraphProt, DeepBind, iDeepS, and GraphProt2. We report average accuracies obtained by 10-fold cross validation together with standard deviations (apart from GraphProt).

RBP	GraphProt	DeepBind	iDeepS	GraphProt2
AGGF1	70.29	71.75±1.26	78.65±1.56	**84.29±1.10**
BUD13	70.06	74.51±1.22	79.78±0.87	**88.66±0.62**
CSTF2T	83.58	85.03±0.71	**89.89±0.60**	**89.62±0.46**
DDX55	67.57	69.61±1.28	72.58±1.44	**85.50±0.69**
EFTUD2	78.52	80.40±1.03	82.58±1.27	**85.44±1.01**
EWSR1	75.65	77.84±1.58	**82.83±0.79**	**82.98±1.08**
FASTKD2	71.26	73.31±1.63	79.84±1.35	**90.31±0.53**
FMR1	74.46	76.96±1.52	82.42±1.65	**92.62±0.77**
FUS	71.51	74.06±0.93	**78.55±1.59**	**78.39±0.73**
FXR2	76.86	78.78±0.96	84.44±0.87	**96.36±0.53**
HNRNPA1	76.01	78.32±1.64	**85.90±1.07**	81.09±1.06
HNRNPC	82.29	87.28±0.85	**91.82±0.65**	89.32±0.99
HNRNPK	88.63	90.42±0.82	**93.76±0.54**	**93.10±0.59**
IGF2BP1	68.64	71.39±0.65	80.46±1.45	**92.97±0.74**
KHDRBS1	76.95	78.12±1.16	**82.64±1.61**	**82.03±0.75**
LIN28B	66.89	70.51±1.24	76.45±1.08	**90.13±0.66**
PCBP2	86.94	89.95±0.51	**93.17±0.78**	**93.16±0.42**
PTBP1	84.27	85.30±0.62	**89.40±0.85**	**90.04±1.01**
PUM2	58.06	63.16±0.81	65.79±2.02	**70.55±1.13**
QKI	75.86	80.79±1.31	**84.32±1.06**	**83.88±1.10**
RBFOX2	67.80	72.21±1.35	76.12±1.10	**78.44±1.24**
SF3B4	70.46	76.44±1.27	78.78±0.76	**90.19±0.56**
SFPQ	67.31	69.29±1.05	70.63±2.74	**74.47±1.58**
SMNDC1	77.86	79.25±1.05	81.37±2.01	**83.56±0.51**
SRSF1	82.64	85.06±1.18	90.06±0.45	**92.08±1.10**
TAF15	78.24	80.63±0.95	**83.66±1.01**	**83.58±0.97**
TARDBP	91.39	91.52±0.81	**94.20±0.95**	92.91±0.74
TIA1	68.64	73.61±1.45	80.67±1.09	**82.16±0.74**
U2AF2	69.88	79.99±1.77	**86.88±0.76**	83.24±0.99
UPF1	61.47	63.32±1.46	69.70±2.33	**91.91±0.80**
AVG	74.66	77.63	82.24	**86.43**

Table 3: Generic model benchmark results over a combined eCLIP set containing sites from 20 different RBPs, for GraphProt, DeepBind, iDeepS, and GraphProt2. In each round a model was trained on 19 RBP sets and tested on the remaining RBP set. We report test accuracies for each round, together with the name of the test set RBP.

RBP	GraphProt	DeepBind	iDeepS	GraphProt2
AGGF1	73.09	72.60	72.83	**79.14**
CSTF2T	**88.67**	88.20	86.48	**89.62**
DDX55	72.93	72.86	72.50	**85.02**
EWSR1	**88.56**	85.21	**88.34**	**88.10**
FMR1	79.65	78.52	80.03	**89.77**
FUS	**86.87**	84.22	85.61	**87.75**
FXR2	80.16	78.75	79.75	**92.58**
HNRNPA1	57.92	62.30	64.08	**70.67**
HNRNPK	85.15	84.06	**88.59**	85.87
IGF2BP1	70.62	70.91	73.67	**92.53**
KHDRBS1	42.41	43.67	45.91	**53.28**
LIN28B	67.35	66.07	65.38	**88.66**
PCBP2	86.79	83.79	84.92	**89.22**
PTBP1	69.54	73.24	**77.61**	**78.82**
PUM1	76.85	76.19	74.52	**91.97**
QKI	49.87	53.20	59.23	**69.03**
TAF15	73.81	73.52	**74.99**	**75.75**
TARDBP	**75.26**	70.61	69.99	**76.10**
TIA1	60.68	62.89	67.78	**83.24**
U2AF2	57.24	60.97	71.57	**82.61**
AVG	72.17	72.09	74.19	82.49

References

[1] Markus Hafner, Markus Landthaler, Lukas Burger, Mohsen Khorshid, Jean Hausser, Philipp Berninger, Andrea Rothballer, Manuel Ascano Jr, Anna-Carina Jungkamp, Mathias Munschauer, et al. Transcriptome-wide identification of rna-binding protein and microrna target sites by par-clip. *Cell*, 141(1):129–141, 2010.

[2] Ammar Tareen and Justin B Kinney. Logomaker: beautiful sequence logos in python. *Bioinformatics*, 36(7):2272–2274, 2020.