A Deep Neural Network Approach to Splice Site Prediction

Tilman Hinnerichs

Knowledge Mining Lab – KAUST

October 27, 2019

Outline

1. Problem Description
2. Dataset description
3. Simple classifiers
4. DiProDB database
   Application of CNN
5. Improvements on simple approach
6. repDNA
7. Application of ensemble methods
8. Influence of nucleotide position

Problem Description

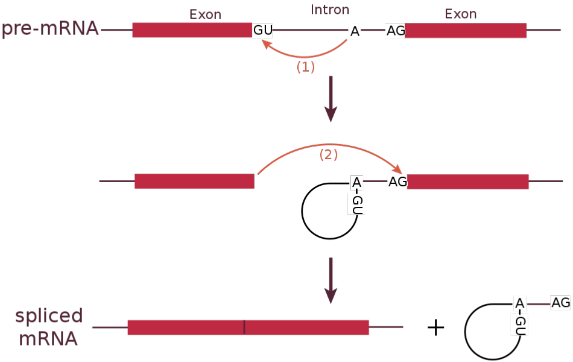


Figure: RNA splicing reaction (en.wikipedia.org)

Problem Description

Splice site prediction on Arabidopsis thaliana genome

Problem Description

Splice site prediction on Arabidopsis thaliana genome

- Acceptor site:
  . . . CGTATCTAGATGAGCA. . .
- Donor site:
  . . . ATGATTTGTGCAGTCA. . .

Problem Description

Splice site prediction on Arabidopsis thaliana genome

- Acceptor site:
  ...CGTATCT AG ATGAGCA...
- Donor site:
  ...ATGATTT GT GCAGTCA...

Problem Description

Splice site prediction on Arabidopsis thaliana genome

- Acceptor site:
  ...CGTATCT AG ATG AG CA...
- Donor site:
  ...ATGATTT GT GCA GT CA...

Dataset description

Example file, e.g., acceptor site

$$\begin{bmatrix}
CT \ldots 300 \text{ nt} \ldots AG \ldots 300 \text{ nt} \ldots GC \\
AG \ldots 300 \text{ nt} \ldots AG \ldots 300 \text{ nt} \ldots TT \\
GA \ldots 300 \text{ nt} \ldots AG \ldots 300 \text{ nt} \ldots AA \\
\vdots \\
100{,}000 \text{ records} \\
\vdots \\
TT \ldots 300 \text{ nt} \ldots AG \ldots 300 \text{ nt} \ldots CC
\end{bmatrix}$$

Simple non-convolutional NN

- Models built on one-hot-encoded data
- Dense networks with dropout

Approach	Samples	Depth	Acceptor acc.	Donor acc.
DNN	20,000	7	92.38	93.43
DNN	200,000	7	93.34	93.34

Figure: Binary classification results

Application of CNN to the DiProDB data

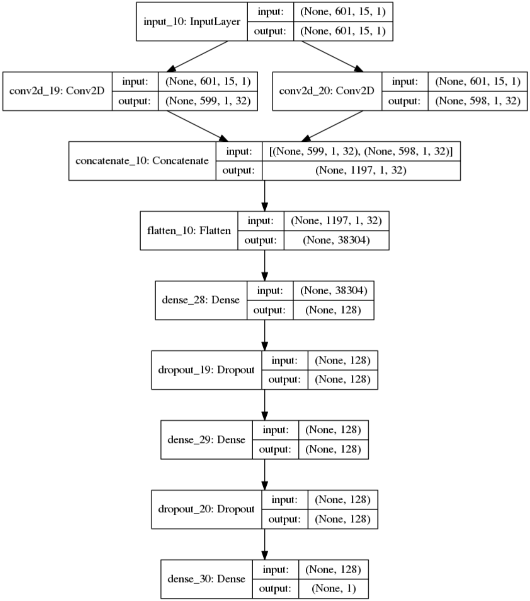
- DiProDB is database for the physicochemical properties of dinucleotides (127 entries)
- Applied PCA yielding 15 dimensions

Application of CNN to the DiProDB data

- DiProDB is database for the physicochemical properties of dinucleotides (127 entries)
- Applied PCA yielding 15 dimensions

Approach	Layers		Acceptor			Donor		
	Conv.	Others	Acc.	Prec.	Rec.	Acc.	Prec.	Rec.
CNN DPDB	4	5	94.4	95.4	94.6	94.9	94.4	94.7
CNN DPDB	4	7	93.5	93.3	94.5	94.0	94.0	93.3
CNN DPDB	6	5	94.0	93.9	94.9	94.2	95.4	91.6
CNN DPDB	6	5	94.4	97.0	93.8	95.2	96.5	93.7
CNN DPDB	2	4	94.3	95.6	94.3	95.3	96.9	94.4
SpliceRover	4	2	96.1	93.9	97.4	95.4	95.6	96.7
Splice2Deep	-	-	95.2	–	94.9	95.6	–	98.8

SpliceRover[Zuallaert et al., 2018]
Splice2Deep[Albaradei et al., 2019]



input_10: InputLayer | input: (None, 601, 15, 1) | output: (None, 601, 15, 1)

conv2d_19: Conv2D | input: (None, 601, 15, 1) | output: (None, 599, 1, 32)

conv2d_20: Conv2D | input: (None, 601, 15, 1) | output: (None, 598, 1, 32)

concatenate_10: Concatenate | input: [(None, 599, 1, 32), (None, 598, 1, 32)] | output: (None, 1197, 1, 32)

flatten_10: Flatten | input: (None, 1197, 1, 32) | output: (None, 38304)

dense_28: Dense | input: (None, 38304) | output: (None, 128)

dropout_19: Dropout | input: (None, 128) | output: (None, 128)

dense_29: Dense | input: (None, 128) | output: (None, 128)

dropout_20: Dropout | input: (None, 128) | output: (None, 128)

dense_30: Dense | input: (None, 128) | output: (None, 1)

Improvements on simple approach

Applying convolutional models to one hot encoding of

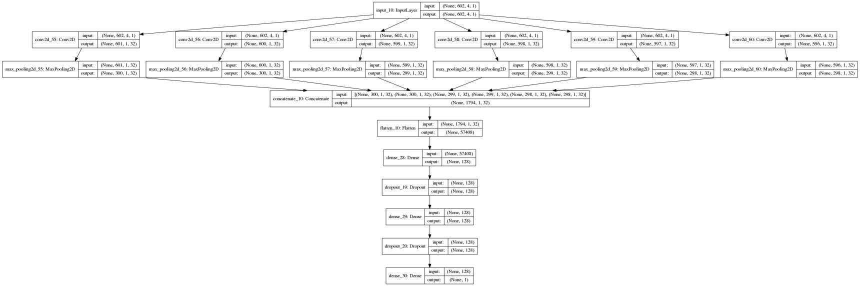
- single nucleotides

Approach	Samples	Acceptor			Donor		
		Acc.	Prec.	Rec.	Acc.	Prec.	Rec.
Simple	200000	94.5	95.6	93.3	95.3	96.7	94.5

- trinucleotides

Approach	Samples	Acceptor			Donor		
		Acc.	Prec.	Rec.	Acc.	Prec.	Rec.
Simple	40000	94.6	93.3	96.7	95.0	92.5	96.3
Simple	200000	95.6	96.6	94.6	95.8	96.7	95.0

Single nucleotides model



input_10: InputLayer | input: (None, 602, 4, 1) | output: (None, 602, 4, 1)
conv2d_55: Conv2D | input: (None, 602, 4, 1) | output: (None, 601, 1, 32)
conv2d_56: Conv2D | input: (None, 602, 4, 1) | output: (None, 600, 1, 32)
conv2d_57: Conv2D | input: (None, 602, 4, 1) | output: (None, 599, 1, 32)
conv2d_58: Conv2D | input: (None, 602, 4, 1) | output: (None, 598, 1, 32)
conv2d_59: Conv2D | input: (None, 602, 4, 1) | output: (None, 597, 1, 32)
conv2d_60: Conv2D | input: (None, 602, 4, 1) | output: (None, 596, 1, 32)
max_pooling2d_55: MaxPooling2D | input: (None, 601, 1, 32) | output: (None, 300, 1, 32)
max_pooling2d_56: MaxPooling2D | input: (None, 600, 1, 32) | output: (None, 300, 1, 32)
max_pooling2d_57: MaxPooling2D | input: (None, 599, 1, 32) | output: (None, 299, 1, 32)
max_pooling2d_58: MaxPooling2D | input: (None, 598, 1, 32) | output: (None, 299, 1, 32)
max_pooling2d_59: MaxPooling2D | input: (None, 597, 1, 32) | output: (None, 298, 1, 32)
max_pooling2d_60: MaxPooling2D | input: (None, 596, 1, 32) | output: (None, 298, 1, 32)
concatenate_10: Concatenate | input: [(None, 300, 1, 32), (None, 300, 1, 32), (None, 299, 1, 32), (None, 299, 1, 32), (None, 298, 1, 32), (None, 298, 1, 32)] | output: (None, 1794, 1, 32)
flatten_10: Flatten | input: (None, 1794, 1, 32) | output: (None, 57408)
dense_28: Dense | input: (None, 57408) | output: (None, 128)
dropout_19: Dropout | input: (None, 128) | output: (None, 128)
dense_29: Dense | input: (None, 128) | output: (None, 128)
dropout_20: Dropout | input: (None, 128) | output: (None, 128)
dense_30: Dense | input: (None, 128) | output: (None, 1)

Figure: Convolutional model with filter sizes (2x4), . . . , (7x4)

Trinucleotides model

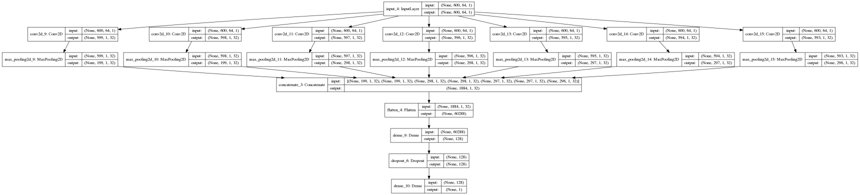

input_4: InputLayer
input: (None, 600, 64, 1)
output: (None, 600, 64, 1)
conv2d_9: Conv2D
input: (None, 600, 64, 1)
output: (None, 599, 1, 32)
conv2d_10: Conv2D
input: (None, 600, 64, 1)
output: (None, 598, 1, 32)
conv2d_11: Conv2D
input: (None, 600, 64, 1)
output: (None, 597, 1, 32)
conv2d_12: Conv2D
input: (None, 600, 64, 1)
output: (None, 596, 1, 32)
conv2d_13: Conv2D
input: (None, 600, 64, 1)
output: (None, 595, 1, 32)
conv2d_14: Conv2D
input: (None, 600, 64, 1)
output: (None, 594, 1, 32)
conv2d_15: Conv2D
input: (None, 600, 64, 1)
output: (None, 593, 1, 32)
max_pooling2d_9: MaxPooling2D
input: (None, 599, 1, 32)
output: (None, 199, 1, 32)
max_pooling2d_10: MaxPooling2D
input: (None, 598, 1, 32)
output: (None, 199, 1, 32)
max_pooling2d_11: MaxPooling2D
input: (None, 597, 1, 32)
output: (None, 298, 1, 32)
max_pooling2d_12: MaxPooling2D
input: (None, 596, 1, 32)
output: (None, 298, 1, 32)
max_pooling2d_13: MaxPooling2D
input: (None, 595, 1, 32)
output: (None, 297, 1, 32)
max_pooling2d_14: MaxPooling2D
input: (None, 594, 1, 32)
output: (None, 297, 1, 32)
max_pooling2d_15: MaxPooling2D
input: (None, 593, 1, 32)
output: (None, 296, 1, 32)
concatenate_3: Concatenate
input: [(None, 199, 1, 32), (None, 199, 1, 32), (None, 298, 1, 32), (None, 298, 1, 32), (None, 297, 1, 32), (None, 297, 1, 32), (None, 296, 1, 32)]
output: (None, 1884, 1, 32)
flatten_4: Flatten
input: (None, 1884, 1, 32)
output: (None, 60288)
dense_9: Dense
input: (None, 60288)
output: (None, 128)
dropout_6: Dropout
input: (None, 128)
output: (None, 128)
dense_10: Dense
input: (None, 128)
output: (None, 1)

Figure: Convolutional model with filter sizes (2 x 64), . . . , (8x64)

repDNA (Liu, 2014)

A „Python package to generate various modes of feature vectors for DNA sequences“:

repDNA content

- Nucleic acid composition
  - kmer
  - Increment of diversity (ID)
- Autocorrelation
  - Dinucleotide-based auto covariance (DAC)
  - Dinucleotide-based cross covariance (DCC)
  - Dinucleotide-based auto-cross covariance (DACC)
  - Trinucleotide-based auto covariance (TAC)
  - Trinucleotide-based cross covariance (TCC)
  - Trinucleotide-based auto-cross covariance (TACC)

repDNA (Liu, 2014)

repDNA content

- Pseudo nucleotide composition
  - Pseudo dinucleotide composition (PseDNC)
  - Pseudo k-tupler nucleotide composition (PseKNC)
  - Parallel correlation pseudo dinucleotide composition (PC-PseDNC)
  - Parallel correlation pseudo trinucleotide composition (PC-PseTNC)
  - Series correlation pseudo dinucleotide composition (SC-PseDNC)
  - Series correlation pseudo trinucleotide composition (SC-PseTNC)

repDNA (Liu, 2014)

repDNA content

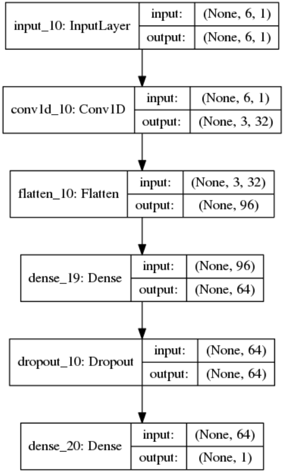
- Pseudo nucleotide composition
  - Pseudo dinucleotide composition (PseDNC)
  - Pseudo k-tupler nucleotide composition (PseKNC)
  - Parallel correlation pseudo dinucleotide composition (PC-PseDNC)
  - Parallel correlation pseudo trinucleotide composition (PC-PseTNC)
  - Series correlation pseudo dinucleotide composition (SC-PseDNC)
  - Series correlation pseudo trinucleotide composition (SC-PseTNC)

- Build model for each encoding and reuse filters for overall model

Classifier model on repDNA features: Results

Approach	Samples	Acceptor			Donor		
		Acc.	Prec.	Rec.	Acc.	Prec.	Rec.
IDkmer	200000	75.2	72.3	76.7	72.7	77.05	75.6
DAC	200000	75.5	72.8	77.0	75.2	68.9	78.8
DCC	200000	75.1	80.0	77.7	74.5	75.2	80.0
TAC	200000	68.0	58.3	72.4	68.2	57.1	73.3
TCC	200000	73.6	75.5	72.7	75.3	65.0	82.0
PseKNC	200000	78.1	76.0	79.4	75.84	71.3	78.4
PC-PseDNC	200000	78.0	76.5	80.1	76.4	75.1	77.9
PC-PseTNC	200000	80.5	76.1	84.2	78.8	76.9	81.6
SC-PseDNC	200000	79.2	74.5	82.4	77.5	77.0	78.8
SC-PseTNC	200000	80.6	76.3	84.8	78.7	77.3	81.5

Classifier model on repDNA features: IDkmer



input_10: InputLayer | input: (None, 6, 1) | output: (None, 6, 1)

conv1d_10: Conv1D | input: (None, 6, 1) | output: (None, 3, 32)

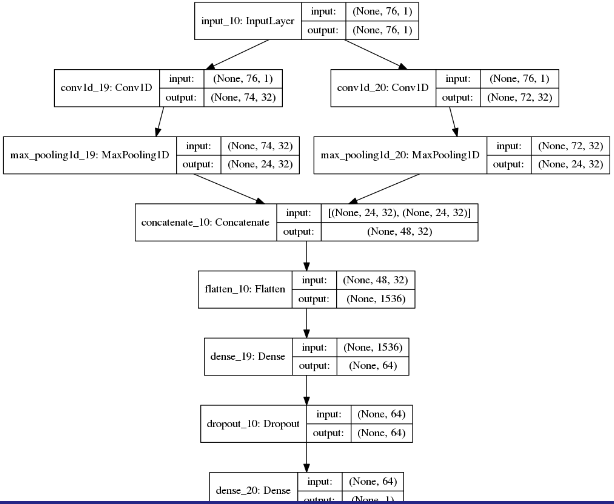
flatten_10: Flatten | input: (None, 3, 32) | output: (None, 96)

dense_19: Dense | input: (None, 96) | output: (None, 64)

dropout_10: Dropout | input: (None, 64) | output: (None, 64)

dense_20: Dense | input: (None, 64) | output: (None, 1)

Classifier model on repDNA features: DAC/DCC/TAC/TCC



input_10: InputLayer | input: (None, 76, 1) | output: (None, 76, 1)

conv1d_19: Conv1D | input: (None, 76, 1) | output: (None, 74, 32)

conv1d_20: Conv1D | input: (None, 76, 1) | output: (None, 72, 32)

max_pooling1d_19: MaxPooling1D | input: (None, 74, 32) | output: (None, 24, 32)

max_pooling1d_20: MaxPooling1D | input: (None, 72, 32) | output: (None, 24, 32)

concatenate_10: Concatenate | input: [(None, 24, 32), (None, 24, 32)] | output: (None, 48, 32)

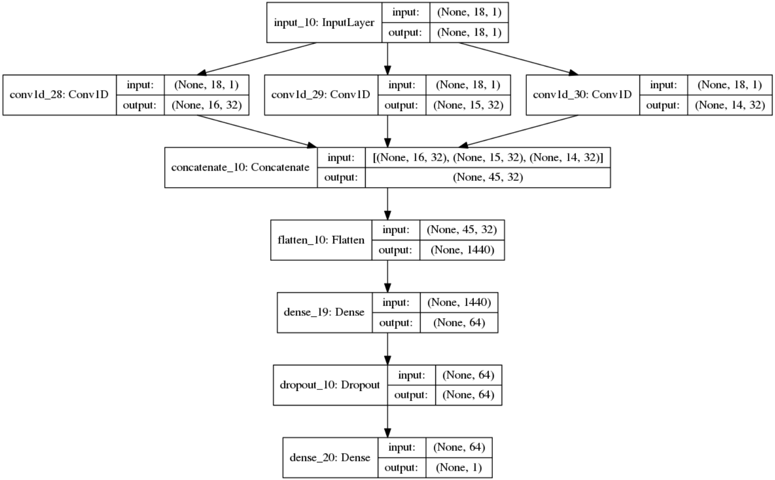
flatten_10: Flatten | input: (None, 48, 32) | output: (None, 1536)

dense_19: Dense | input: (None, 1536) | output: (None, 64)

dropout_10: Dropout | input: (None, 64) | output: (None, 64)

dense_20: Dense | input: (None, 64) | output: (None, 1)

Classifier model on repDNA features: PseNAC



input_10: InputLayer	input:	(None, 18, 1)
	output:	(None, 18, 1)
conv1d_28: Conv1D	input:	(None, 18, 1)
	output:	(None, 16, 32)
conv1d_29: Conv1D	input:	(None, 18, 1)
	output:	(None, 15, 32)
conv1d_30: Conv1D	input:	(None, 18, 1)
	output:	(None, 14, 32)
concatenate_10: Concatenate	input:	[(None, 16, 32), (None, 15, 32), (None, 14, 32)]
	output:	(None, 45, 32)
flatten_10: Flatten	input:	(None, 45, 32)
	output:	(None, 1440)
dense_19: Dense	input:	(None, 1440)
	output:	(None, 64)
dropout_10: Dropout	input:	(None, 64)
	output:	(None, 64)
dense_20: Dense	input:	(None, 64)
	output:	(None, 1)

Additional models

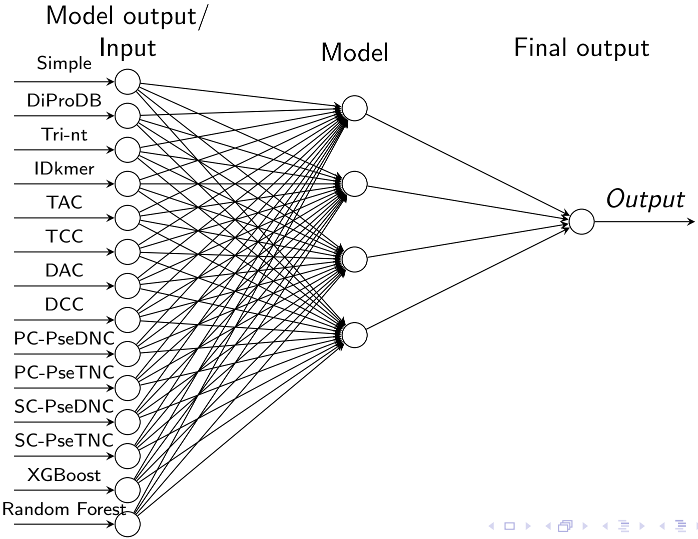
- XGBoost: Library for gradient boosting algorithms
- Random Forest

Additional models

- XGBoost: Library for gradient boosting algorithms
- Random Forest

Approach	Acceptor			Donor		
	Acc.	Prec.	Rec.	Acc.	Prec.	Rec.
XGBoost	90.8	89.5	92.0	92.0	90.6	93.3
Random Forest	83.5	83.0	83.8	86.0	86.3	85.8

Ensemble method: Model



Ensemble method: Results

See this as another classification problem:

Approach	Acceptor			Donor		
	Acc.	Prec.	Rec.	Acc.	Prec.	Rec.
Naive Bayes	83.5	83.0	83.8	85.9	86.3	85.8
Grad boost	83.5	83.0	83.8	86.0	86.3	85.8
Random Forest	93.7	94.2	93.3	94.1	94.9	93.4
NN	83.5	83.0	83.8	87.0	85.5	87.4

Ensemble method: Results

Minimization techniques over weights over training data:

- Nalder-Mead
- Powell

Ensemble method: Results

Minimization techniques over weights over training data:

- Nalder-Mead
- Powell

Approach	Acceptor			Donor		
	Acc.	Prec.	Rec.	Acc.	Prec.	Rec.
Soft Min.	83.8	83.5	83.0	86.0	86.3	85.8
Hard Min.	83.5	83.0	84.1	86.5	86.7	86.3

Ensemble methods: Results

Minimization over validation data:

- Random search
- Genetic algorithm

Ensemble methods: Results

Minimization over validation data:

- Random search
- Genetic algorithm

Acceptor:

	Weights								Results		
Mode	S	D	T	R	X	dc	PsePC	PseSC	Acc.	Prec.	Rec.
S	52	54	98	4	40	6	30	4	95.8	95.5	95.9
H	20	81	98	3	50	3	13	4	95.7	95.8	95.4

Donor:

	Weights								Results		
Mode	S	D	T	R	X	dc	PsePC	PseSC	Acc.	Prec.	Rec.
S	31	69	97	7	60	6	10	5	96.2	96.3	96.1
H	71	91	95	12	77	5	8	8	96.0	96.6	95.6

Influence of nucleotide position

1. Divide upstream and downstream sequences in $\{6,3,2\}$ parts
2. Stitch data back together
3. Apply classification

Dividing sequences into chunks

Example file, e.g., acceptor site

$$\begin{bmatrix} CT \ldots 300 \text{ nt} \ldots AG \ldots 300 \text{ nt} \ldots GC \\ AG \ldots 300 \text{ nt} \ldots AG \ldots 300 \text{ nt} \ldots TT \\ GA \ldots 300 \text{ nt} \ldots AG \ldots 300 \text{ nt} \ldots AA \\ \vdots \\ \rotatebox{90}{$100{,}000$ records} \\ \vdots \\ TT \ldots 300 \text{ nt} \ldots AG \ldots 300 \text{ nt} \ldots CC \end{bmatrix}$$

Influence of nucleotide position: Results

Accuracy for chunk pair:

Simple classifier:

	Acceptor						Donor					
	1	2	3	4	5	6	1	2	3	4	5	6
1	77.1	68.9	67.5	65.5	65.3	64.8	85.1	74.0	66.4	64.9	64.4	64.7
2	77.4	69.4	66.1	66.2	65.4	64.9	85.2	74.2	66.6	65.3	64.6	64.8
3												
4												
5												
6												

DiProDB classifier:

	Acceptor						Donor					
	1	2	3	4	5	6	1	2	3	4	5	6
1	77.1	70.0	66.2	66.4	65.84	65.5	85.2	75.2	66.9	65.5	65.1	65.2
2	77.7	70.3	66.7	66.3	65.9	65.6	85.2	75.4	67.3	65.5	64.9	65.2
3												
4												
5												
6												

Trinucleotide classifier:

	Acceptor						Donor					
	1	2	3	4	5	6	1	2	3	4	5	6
1	78.2	71.3	67.4	67.0	66.9	66.3	85.4	76.1	67.7	66.34	65.8	68.9
2	78.6	71.6	67.7	67.1	66.8	66.2	85.6	76.3	67.8	66.2	65.8	65.8
3												
4												
5												
6												

Gradient Boosting classifier:

	Acceptor						Donor					
	1	2	3	4	5	6	1	2	3	4	5	6
1	68.6	60.5	58.5	58.5	57.9	57.7	77.4	64.9	59.0	57.0	57.3	57.2
2	68.6	60.4	58.6	59.0	57.9		77.4	64.9	58.0	57.6	57.3	
3												
4												
5												
6												

Random Forest classifier:

	Acceptor						Donor					
Chunk	1	2	3	4	5	6	1	2	3	4	5	6
1	69.2	58.6	56.8	57.1	56.6	56.2	78.9	64.3	55.3	54.8	55.5	55.4
2	69.4	58.9	57.3	57.1	56.6		78.9	64.3	55.6	54.8	55.5	
3												
4												
5												
6												

Influence of nucleotide position: Results

Accuracy for chunk pair:

Chunk	1	2	3
1			
2			
3			

Citations

- Liu B, Liu F, Fang L, Wang X, Chou K-C.repDNA: a Python package to generate various modes of feature vectors for DNA sequences by incorporating user-defined physicochemical properties and sequence-order effects. Bioinformatics 2015;31(8):1307-1309.
- Jasper Zuallaert, Fréderic Godin, Mijung Kim, Arne Soete, Yvan Saeys, Wesley De Neve, SpliceRover: interpretable convolutional neural networks for improved splice site prediction, Bioinformatics, Volume 34, Issue 24, 15 December 2018, Pages 4180–4188, https://doi.org/10.1093/bioinformatics/bty497