A Deep Neural Network Approach to Splice Site Prediction

Tilman Hinnerichs

Knowledge Mining Lab – KAUST

October 3, 2019

Outline

Problem Description

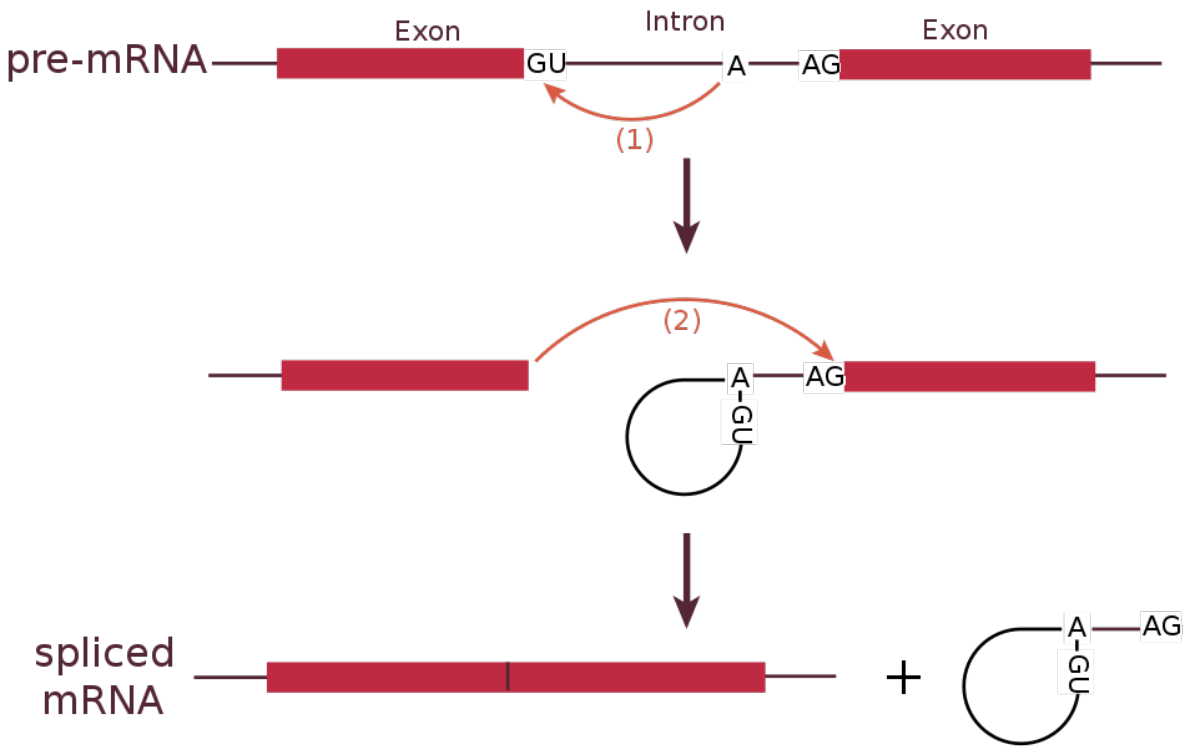


Figure: RNA splicing reaction (en.wikipedia.org)

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

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

Next Steps from two weeks ago

- Adapt convolution techniques from other papers with same goal
- Vary pre and post marker nt sequence length
- Use DiProDB[Friedel et al., NAR, 2009] for input
- Utilize electron io interaction potential (EIIP) for prediction

Application of NN to the DiProDB data

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

Application of NN to the DiProDB data

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

Approach	Samples	Depth	Acceptor			Donor		
			Acc.	Prec.	Rec.	Acc.	Prec.	Rec.
DiProDB	40,000	7	92.38	92.1	93.5	93.6	92.9	93.6
DiProDB	40,000	9	92.28	90.4	94.2	93.4	92.0	93.9

DiProDB BINARY CLASSIFICATION APPROACH
Data shape: (40000, 601, 15)

Layer (type)	Output Shape	Param #
input_10 (InputLayer)	(None, 601, 15)	0
flatten_10 (Flatten)	(None, 9015)	0
dense_46 (Dense)	(None, 150)	1352400
dropout_37 (Dropout)	(None, 150)	0
dense_47 (Dense)	(None, 80)	12080
dropout_38 (Dropout)	(None, 80)	0
dense_48 (Dense)	(None, 30)	2430
dropout_39 (Dropout)	(None, 30)	0
dense_49 (Dense)	(None, 10)	310
dropout_40 (Dropout)	(None, 10)	0
dense_50 (Dense)	(None, 1)	11

Application of CNN to the DiProDB data

- Convolutional NN to adapt ideas from other papers

Application of CNN to the DiProDB data

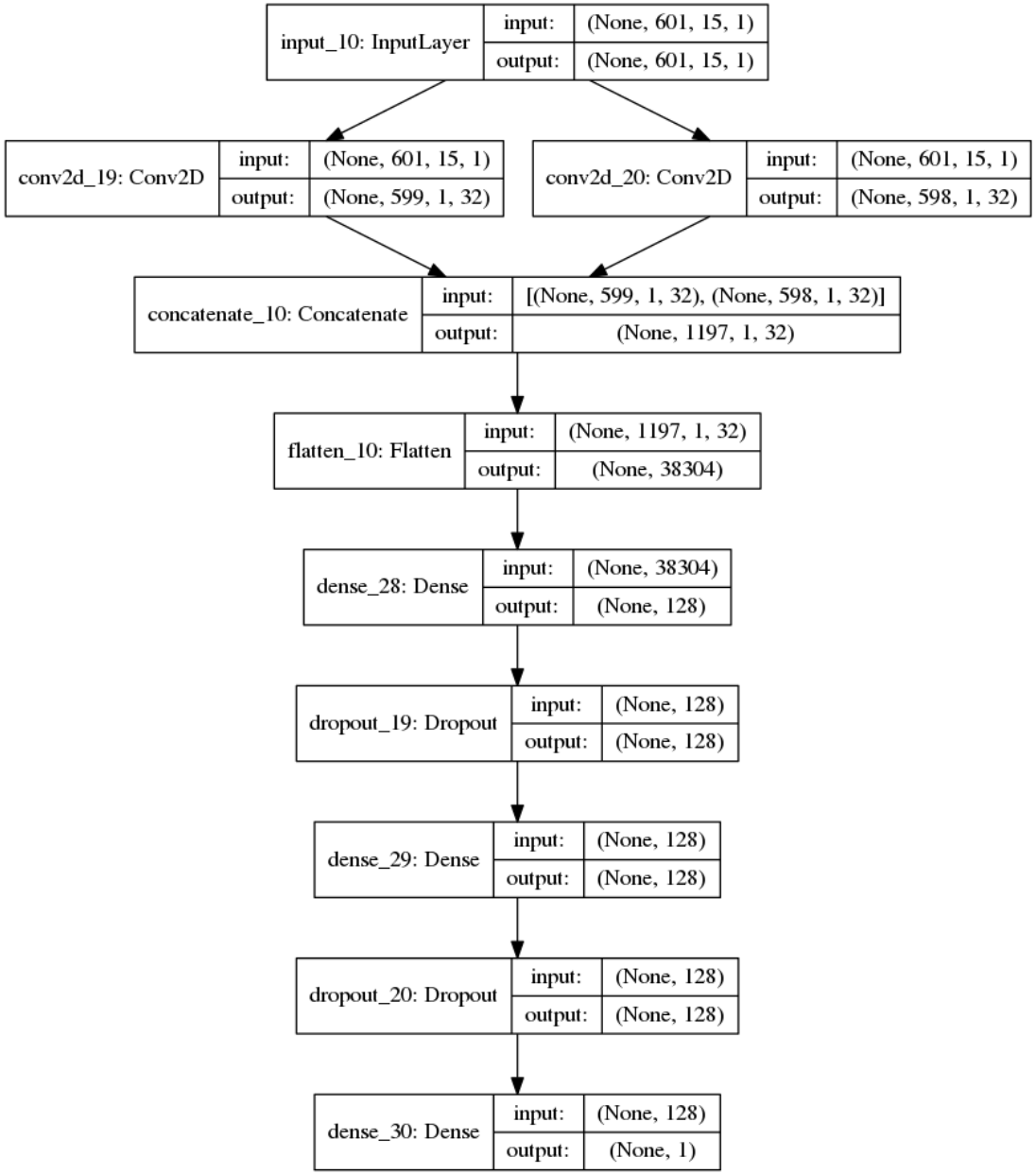
- Convolutional NN to adapt ideas from other papers

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
SpliceRover	4	2	96.1	93.9	97.4	95.4	95.6	96.7
CNN DPDB(*)	2	4	94.3	95.6	94.3	95.3	96.9	94.4

SpliceRover[Zuallaert et al., 2018]

DiProDB: BINARY CLASSIFICATION APPROACH
Data shape: (40000, 601, 15, 1)
Epochs: 5, Batch size: 500
Model: "model_10"

Layer (type)	Output Shape	Param #
input_10 (InputLayer)	(None, 601, 15, 1)	0
conv2d_28 (Conv2D)	(None, 599, 1, 32)	1472
max_pooling2d_28 (MaxPooling	(None, 299, 1, 32)	0
conv2d_29 (Conv2D)	(None, 297, 1, 64)	6208
max_pooling2d_29 (MaxPooling	(None, 148, 1, 64)	0
conv2d_30 (Conv2D)	(None, 146, 1, 128)	24704
max_pooling2d_30 (MaxPooling	(None, 48, 1, 128)	0
flatten_10 (Flatten)	(None, 6144)	0
dense_28 (Dense)	(None, 64)	393280
dropout_19 (Dropout)	(None, 64)	0
dense_29 (Dense)	(None, 64)	4160
dropout_20 (Dropout)	(None, 64)	0
dense_30 (Dense)	(None, 1)	65


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	200000	94.6	93.3	96.7	95.0	92.5	96.3

Single nucleotides model

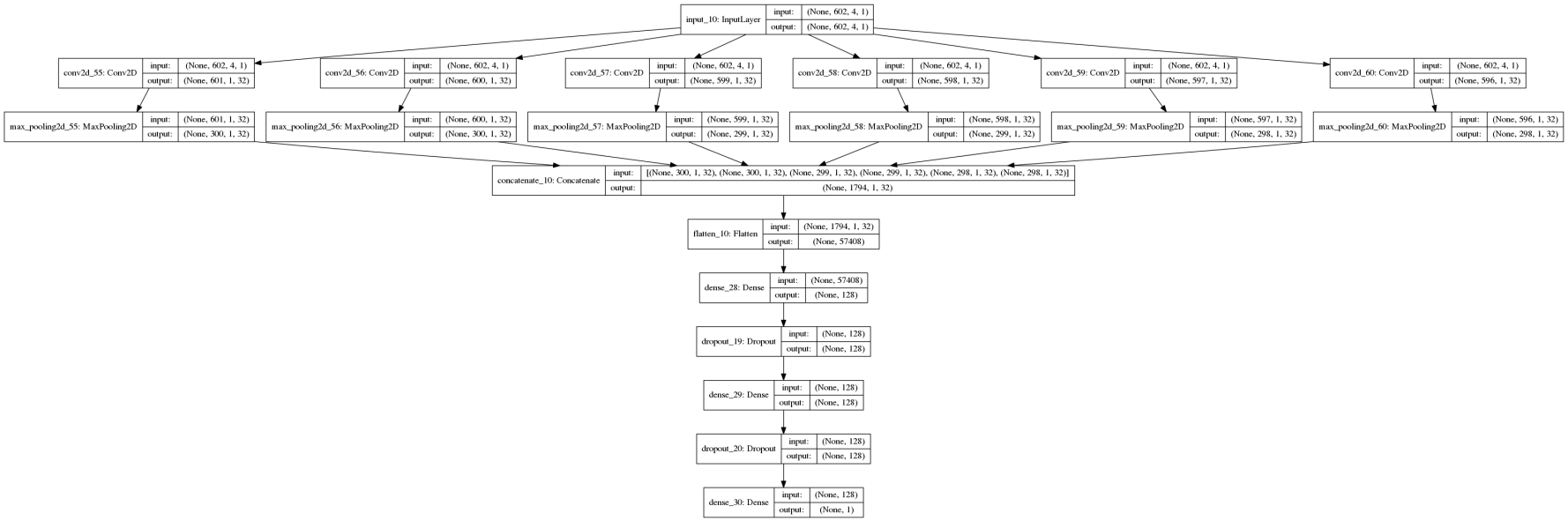


Figure: Convolutional model with filter sizes (2x4), . . . , (7x4)

Trinucleotides model

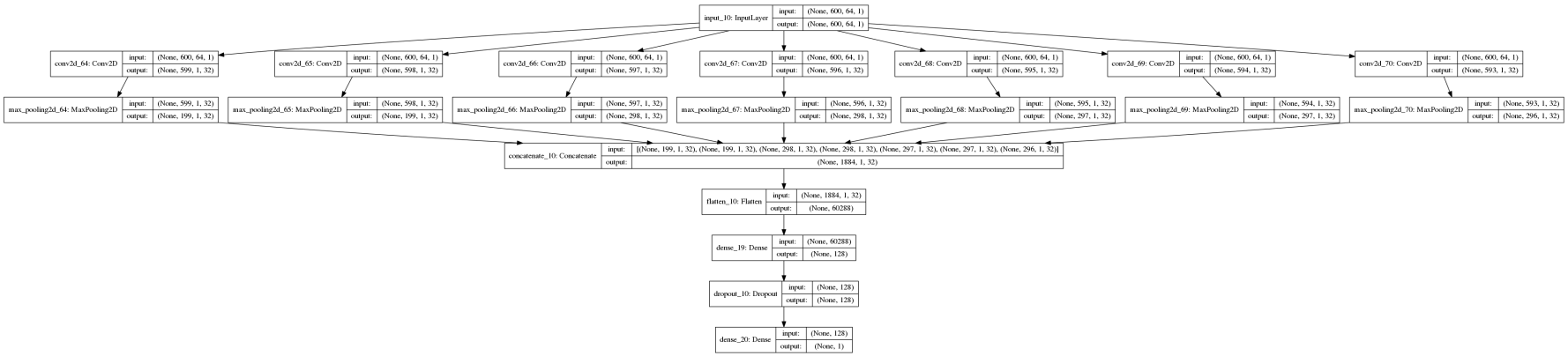


Figure: Convolutional model with filter sizes (2 x 64), . . . , (8x64)

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)

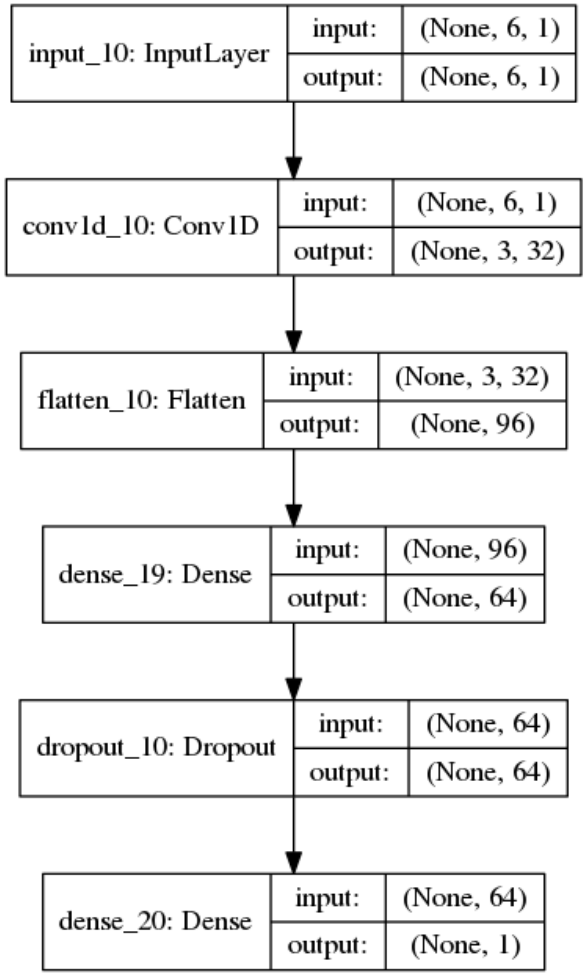
repDNA content

- 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

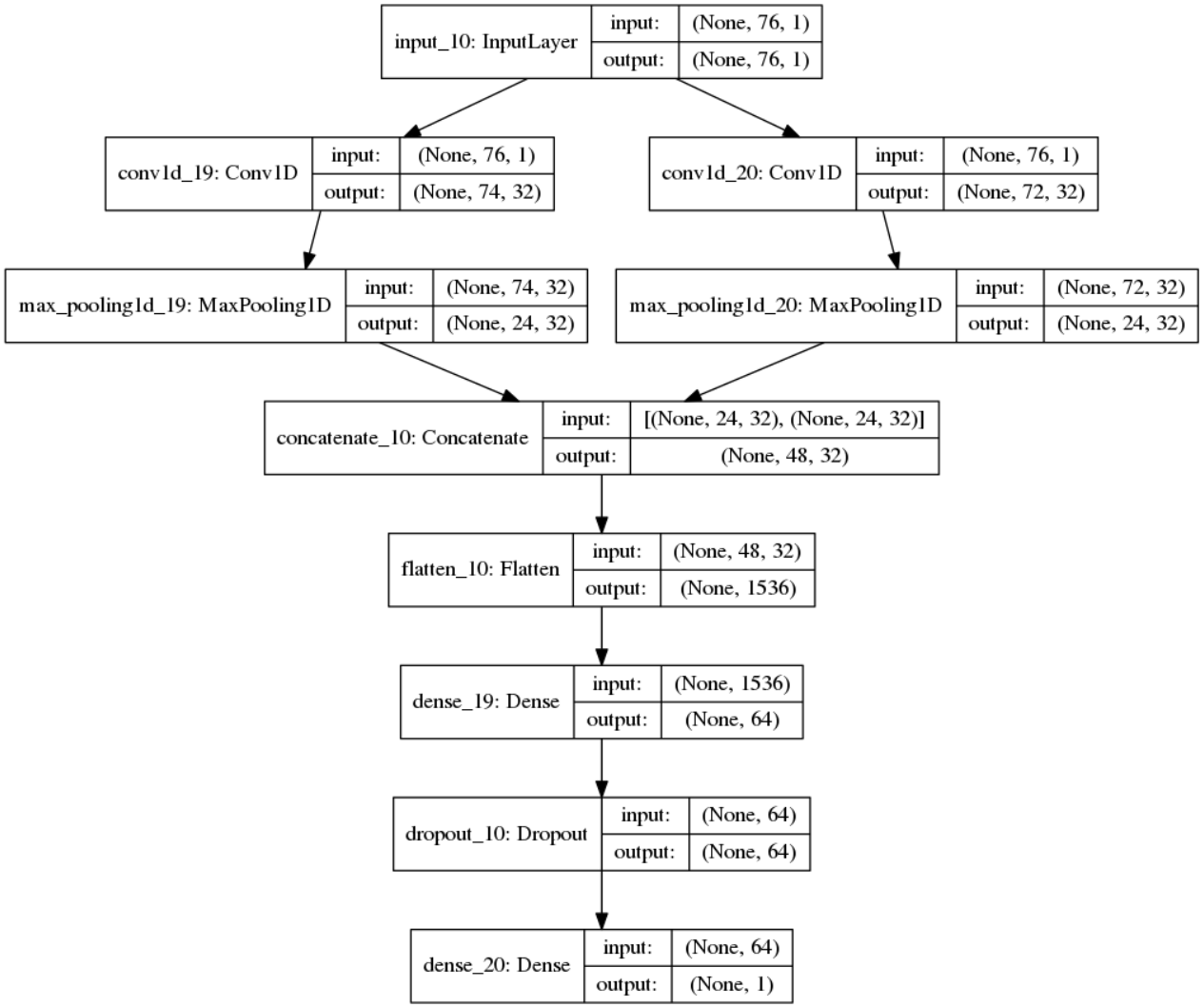
Classifier model on repDNA features: Results

Approach	Samples	Acceptor			Donor		
		Acc.	Prec.	Rec.	Acc.	Prec.	Rec.
IDkmer	40000	75.4	71.4	78.5	73.2	69.6	76.5
DAC	40000	74.1	74.2	74.3	72.9	72.2	73.2
DCC	40000	75.1	80.0	77.7	74.5	75.2	80.0
PC-PseDNC	40000	78.0	76.5	80.1	76.4	75.1	77.9
PC-PseTNC	40000	80.5	76.1	84.2	78.8	76.9	81.6
SC-PseDNC	40000	79.2	74.5	82.4	77.5	77.0	78.8
SC-PseTNC	40000	80.6	76.3	84.8	78.7	77.3	81.5

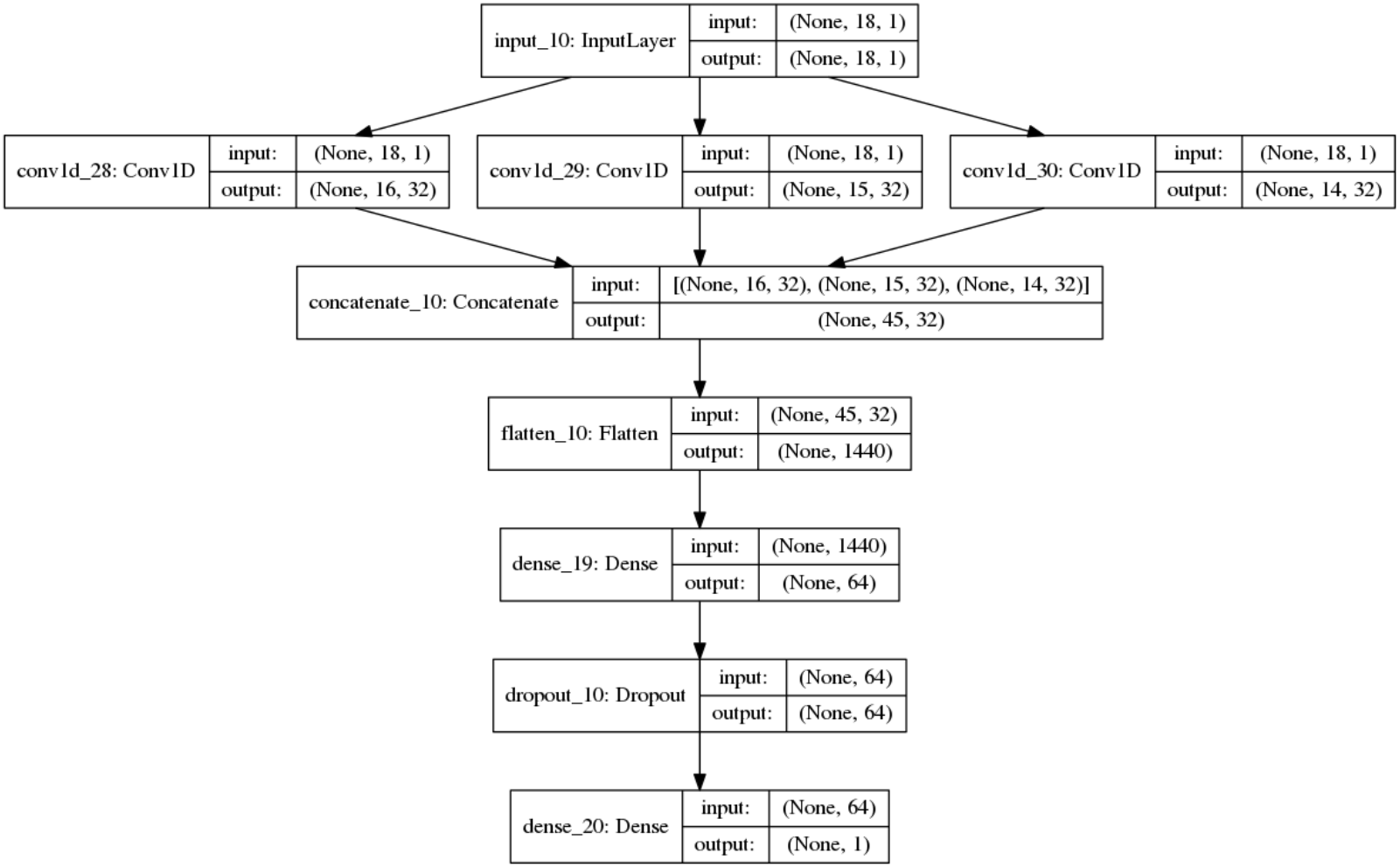
Classifier model on repDNA features: IDkmer



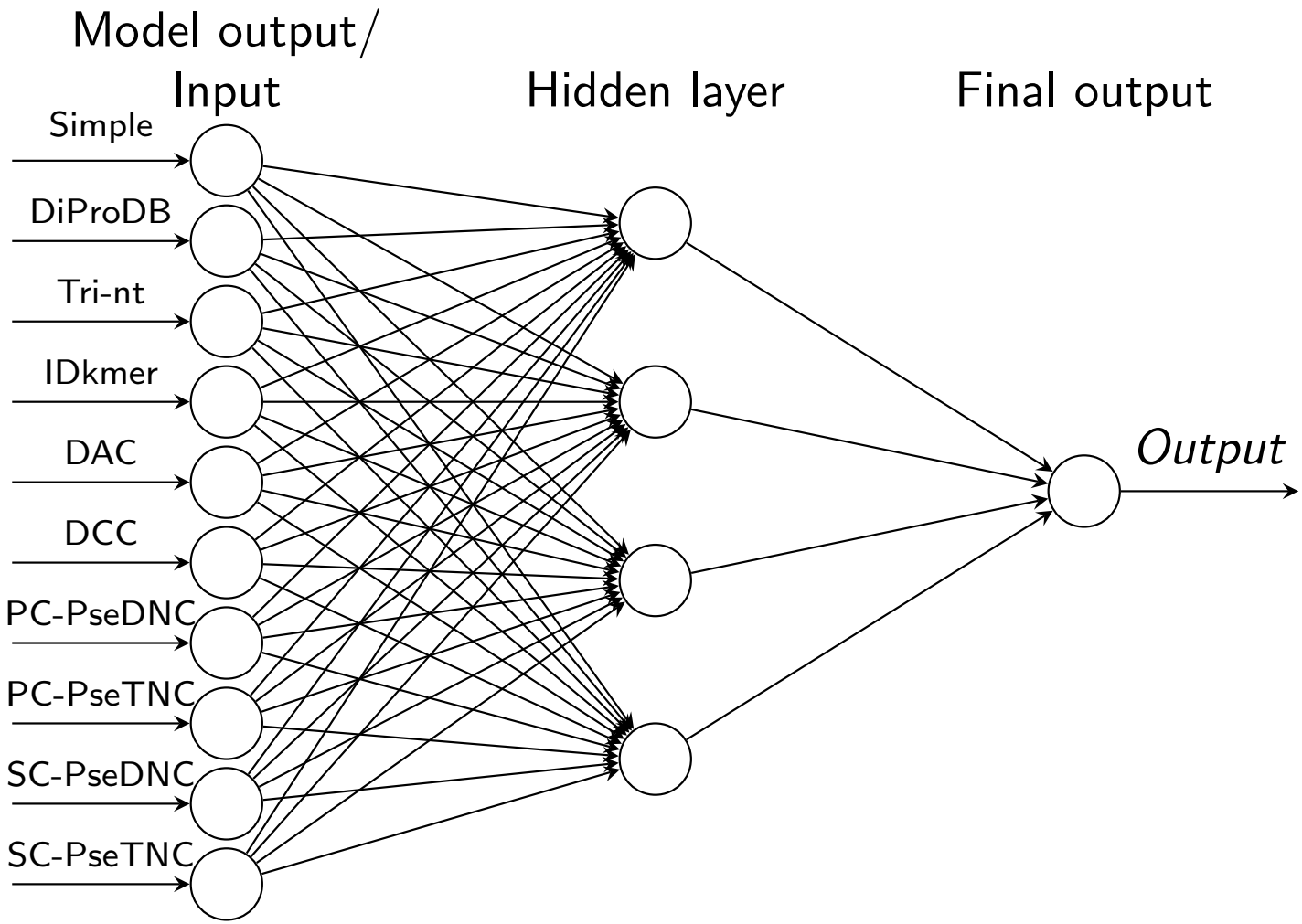
Classifier model on repDNA features: DAC/DCC



Classifier model on repDNA features: PseNAC



Funneling method: Model



Funneling method: Results

Sampl.	S	Di	IK	Pse	D	Tr	Acceptor			Donor		
							Acc.	Prec.	Rec.	Acc.	Prec.	Rec.
40000	X	X					97.7	98.1	96.7	—	—	—
40000	X	X	X				97.7	98.1	96.7	—	—	—
40000	X	X	X	X			97.6	98.2	96.2	—	—	—
40000	X	X	X	X	X		97.5	98.2	96.5	97.4	97.3	96.3
40000	X	X				X	98.0	98.4	97.7	98.7	1.27	(*)

Next steps

- Follow improvements of SpliceRover
- repDNA (Liu et al., 2015) for different embeddings of the sequences
- Train CNN on all embeddings
- Freeze convolutional layers and funnel them into DNN

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