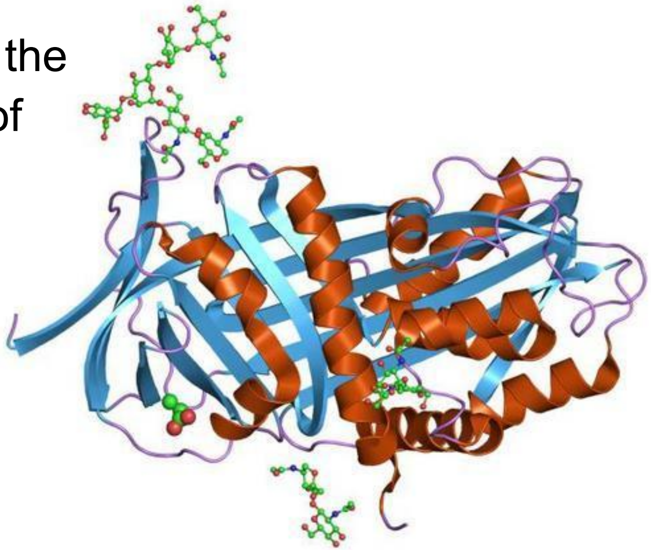
Protein Similarity

Cecilia Huang, Evelina Danielsson, Joel Bäcker

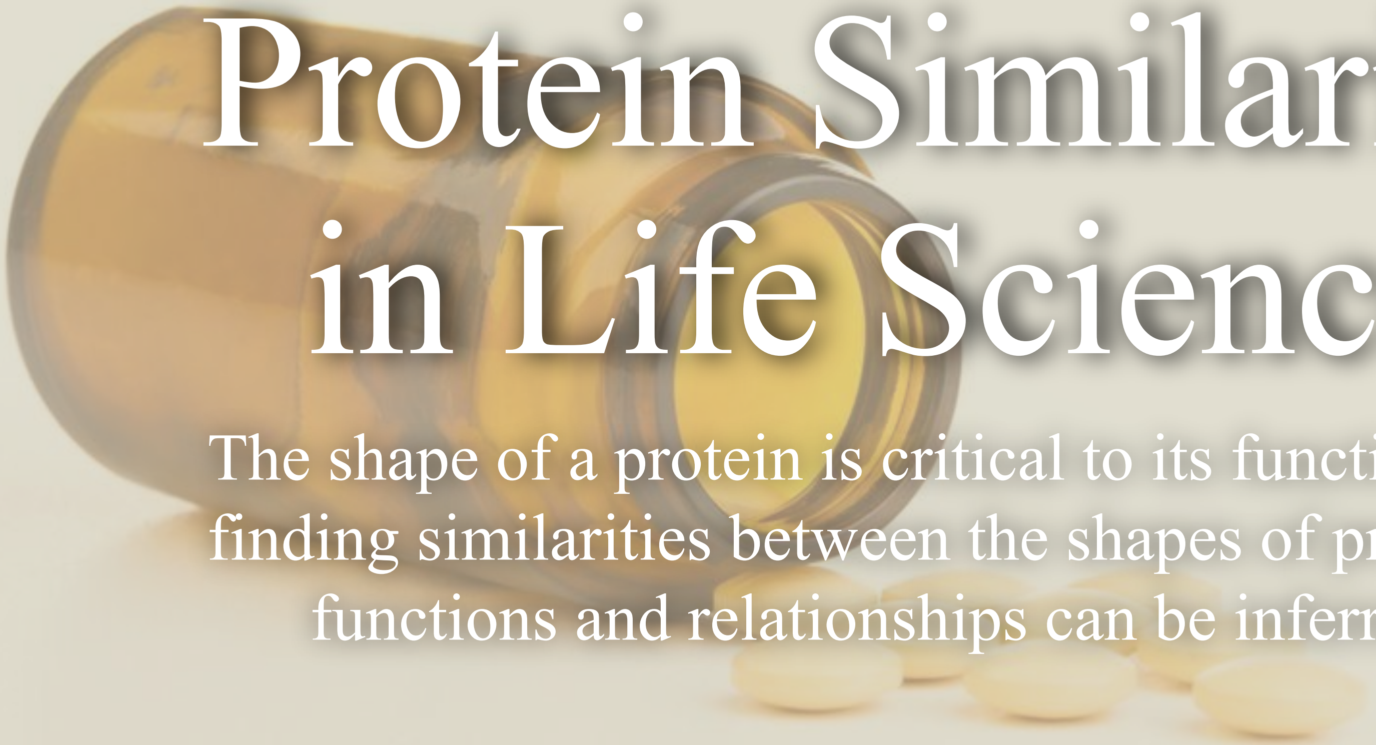
Supervisor: Daniel Varela

Andre Lab

Combination of computational and experimental methods to understand the structure, interactions and evolution of proteins.



Protein Similarity in Life Science

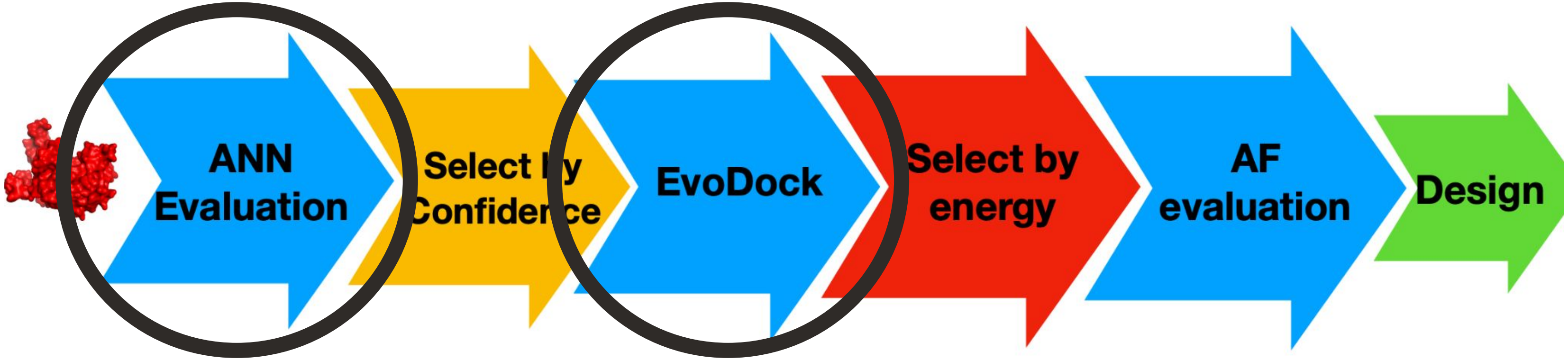


The shape of a protein is critical to its function. By finding similarities between the shapes of proteins, functions and relationships can be inferred.

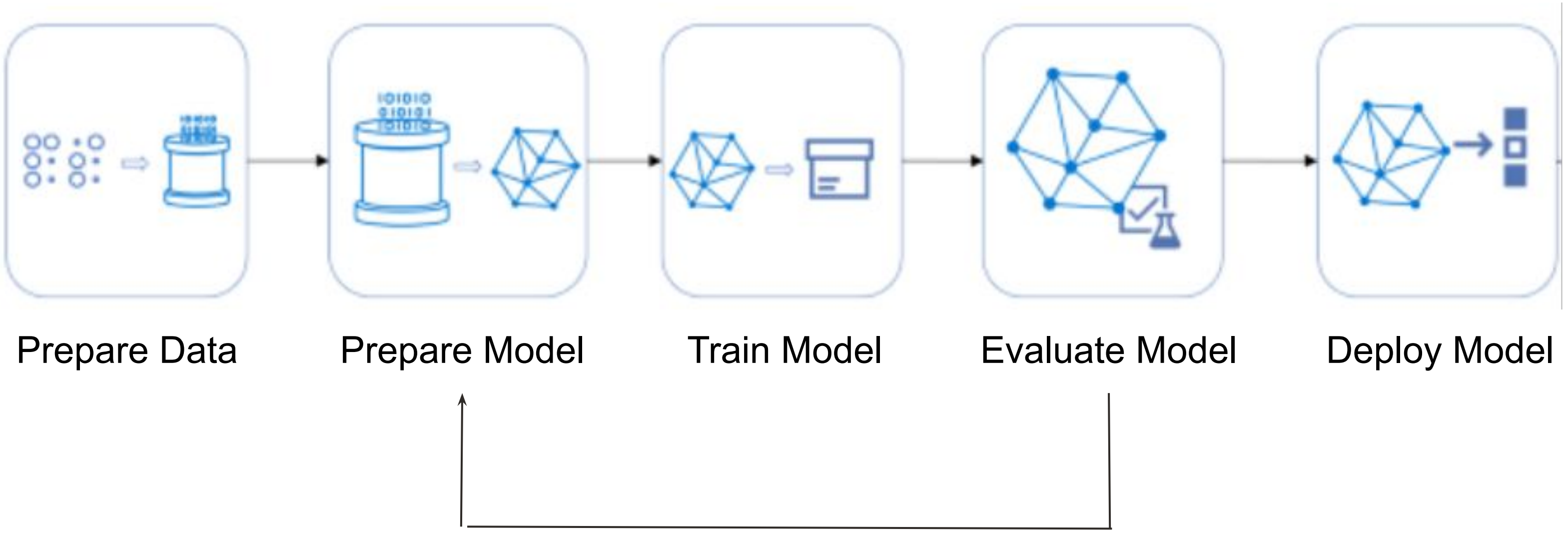
Predicting protein structure: Challenges

- Large dataset of proteins (500.000)
- Computationally expensive (300 seconds comparing proteins!)

AndreLab protein design approach



Standard ML pipelining

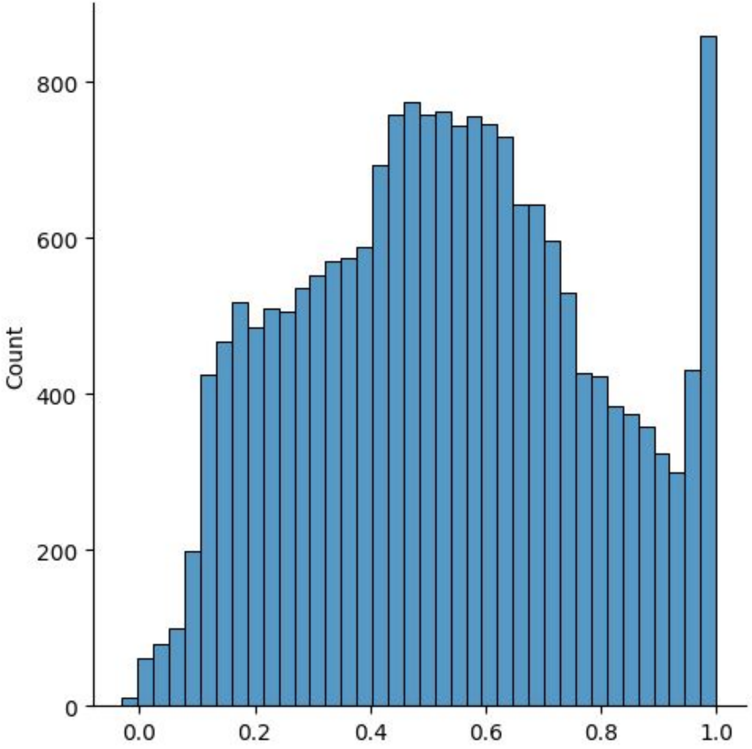


Data Generation

- Random selection
- Zernike-Canterakis shape descriptor
- ZEAL score

How to create one sample

ZEAL score distribution



Features and targets

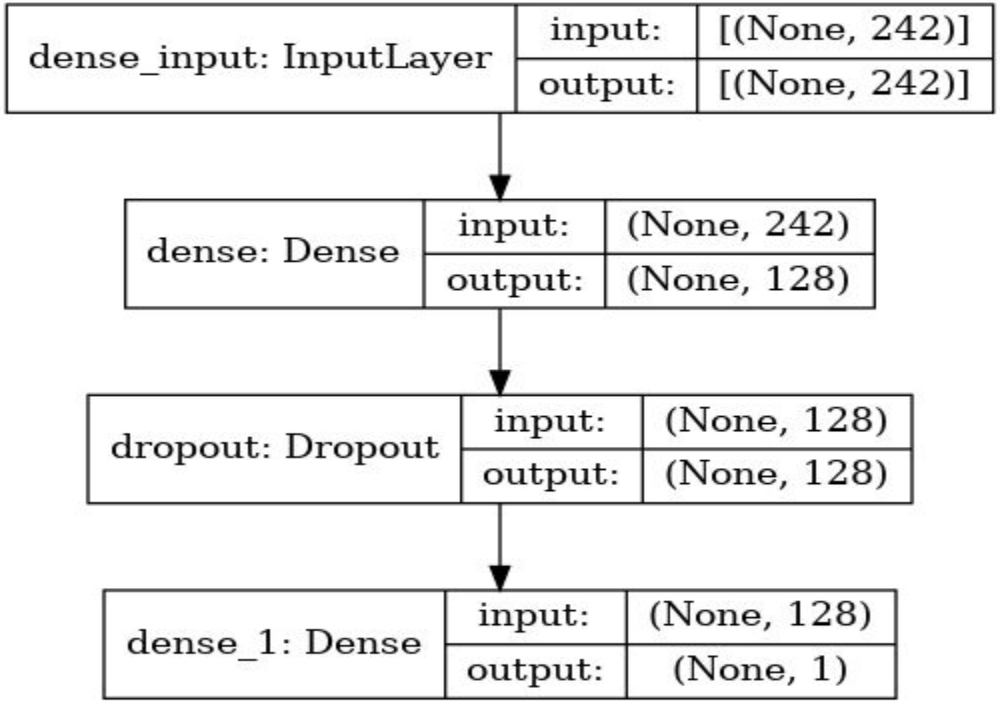
$$X = \begin{bmatrix} zd^{1}{}_{T0} & zd^{1}{}_{T1} & \cdots & zd^{1}{}_{T120} & zd^{1}{}_{R0} & \ldots & zd^{1}{}_{R120} \\ zd^{2}{}_{T0} & zd^{2}{}_{T1} & \ldots & zd^{2}{}_{T120} & zd^{2}{}_{R0} & \ldots & zd^{2}{}_{R120} \\ zd^{3}{}_{T0} & & \ldots & & \ddots & & \\ \vdots & & & & & \ddots & \\ zd^{n}{}_{T0} & & \ldots & & & & zd^{n}{}_{R120} \end{bmatrix}$$

$$y = \begin{bmatrix} zeal_{1} \\ zeal_{2} \\ \vdots \\ zeal_{n} \end{bmatrix}$$

Model Selection

- Regression
    - Linear Regression
    - Random Forest Regressor
- Neural Networks
    - Feed-Forward Network
    - 1D Convolutional Neural Network
    - 1D Fully-Convolutional Network

Feed Forward Network



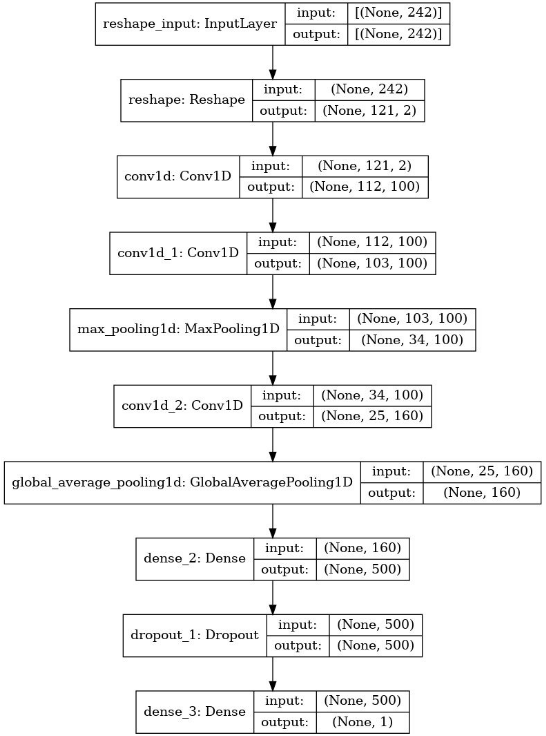
dense_input: InputLayer | input: | [(None, 242)]
output: | [(None, 242)]

dense: Dense | input: | (None, 242)
output: | (None, 128)

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

dense_1: Dense | input: | (None, 128)
output: | (None, 1)

Convolutional Neural Network



reshape_input: InputLayer | input: [(None, 242)] | output: [(None, 242)]

reshape: Reshape | input: (None, 242) | output: (None, 121, 2)

conv1d: Conv1D | input: (None, 121, 2) | output: (None, 112, 100)

conv1d_1: Conv1D | input: (None, 112, 100) | output: (None, 103, 100)

max_pooling1d: MaxPooling1D | input: (None, 103, 100) | output: (None, 34, 100)

conv1d_2: Conv1D | input: (None, 34, 100) | output: (None, 25, 160)

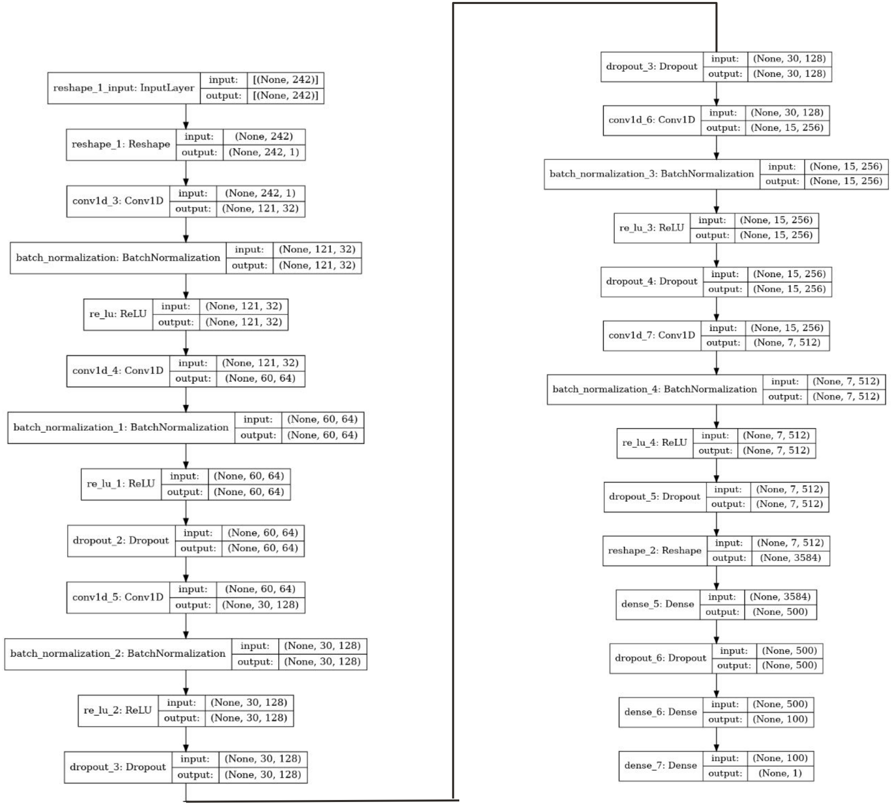
global_average_pooling1d: GlobalAveragePooling1D | input: (None, 25, 160) | output: (None, 160)

dense_2: Dense | input: (None, 160) | output: (None, 500)

dropout_1: Dropout | input: (None, 500) | output: (None, 500)

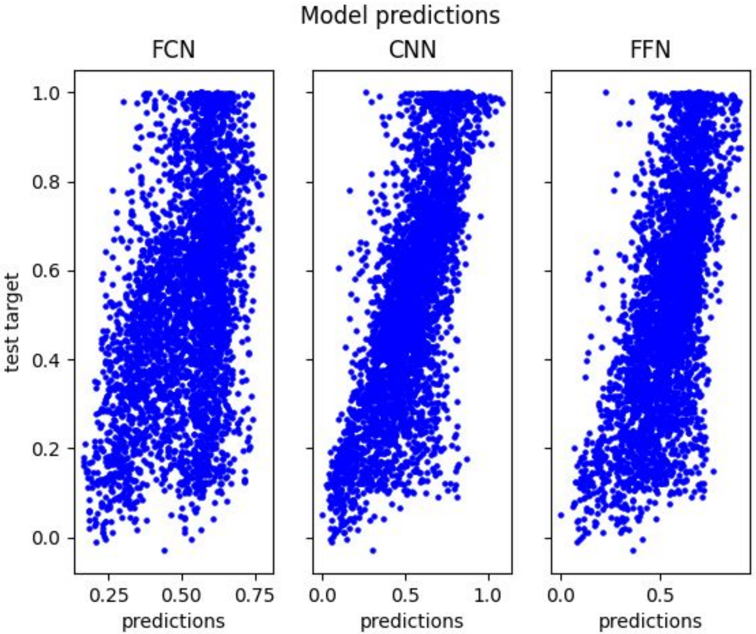
dense_3: Dense | input: (None, 500) | output: (None, 1)

Fully Convolutional Network

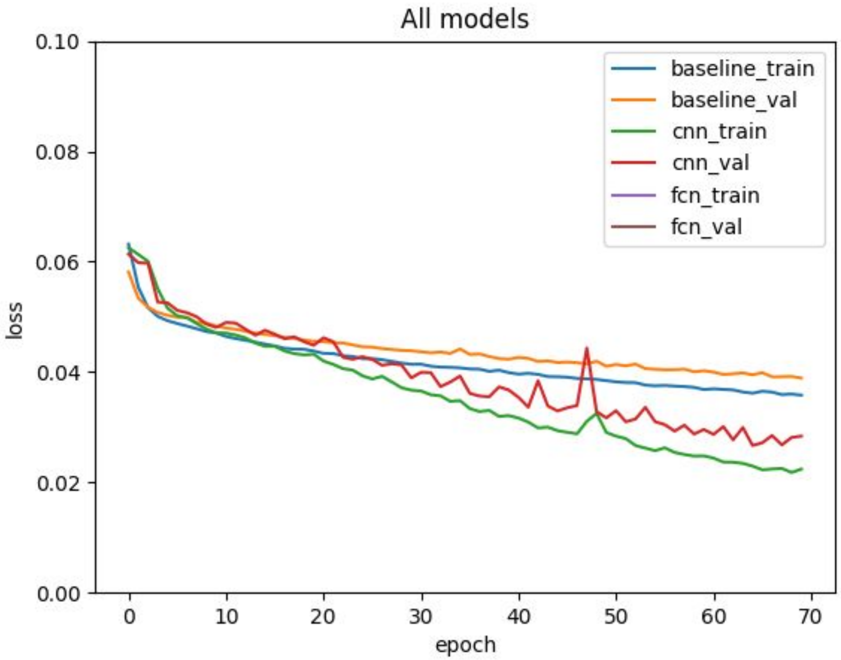


reshape_1_input: InputLayer	input:	[(None, 242)]
	output:	[(None, 242)]
reshape_1: Reshape	input:	(None, 242)
	output:	(None, 242, 1)
conv1d_3: Conv1D	input:	(None, 242, 1)
	output:	(None, 121, 32)
batch_normalization: BatchNormalization	input:	(None, 121, 32)
	output:	(None, 121, 32)
re_lu: ReLU	input:	(None, 121, 32)
	output:	(None, 121, 32)
conv1d_4: Conv1D	input:	(None, 121, 32)
	output:	(None, 60, 64)
batch_normalization_1: BatchNormalization	input:	(None, 60, 64)
	output:	(None, 60, 64)
re_lu_1: ReLU	input:	(None, 60, 64)
	output:	(None, 60, 64)
dropout_2: Dropout	input:	(None, 60, 64)
	output:	(None, 60, 64)
conv1d_5: Conv1D	input:	(None, 60, 64)
	output:	(None, 30, 128)
batch_normalization_2: BatchNormalization	input:	(None, 30, 128)
	output:	(None, 30, 128)
re_lu_2: ReLU	input:	(None, 30, 128)
	output:	(None, 30, 128)
dropout_3: Dropout	input:	(None, 30, 128)
	output:	(None, 30, 128)
dropout_3: Dropout	input:	(None, 30, 128)
	output:	(None, 30, 128)
conv1d_6: Conv1D	input:	(None, 30, 128)
	output:	(None, 15, 256)
batch_normalization_3: BatchNormalization	input:	(None, 15, 256)
	output:	(None, 15, 256)
re_lu_3: ReLU	input:	(None, 15, 256)
	output:	(None, 15, 256)
dropout_4: Dropout	input:	(None, 15, 256)
	output:	(None, 15, 256)
conv1d_7: Conv1D	input:	(None, 15, 256)
	output:	(None, 7, 512)
batch_normalization_4: BatchNormalization	input:	(None, 7, 512)
	output:	(None, 7, 512)
re_lu_4: ReLU	input:	(None, 7, 512)
	output:	(None, 7, 512)
dropout_5: Dropout	input:	(None, 7, 512)
	output:	(None, 7, 512)
reshape_2: Reshape	input:	(None, 7, 512)
	output:	(None, 3584)
dense_5: Dense	input:	(None, 3584)
	output:	(None, 500)
dropout_6: Dropout	input:	(None, 500)
	output:	(None, 500)
dense_6: Dense	input:	(None, 500)
	output:	(None, 100)
dense_7: Dense	input:	(None, 100)
	output:	(None, 1)

Evaluation


Model predictions
FCN
CNN
FFN
test target
1.0
0.8
0.6
0.4
0.2
0.0
0.25
0.50
0.75
0.0
0.5
1.0
0.0
0.5
predictions
predictions
predictions

Evaluation


All models
0.10
0.08
0.06
0.04
0.02
0.00
loss
0
10
20
30
40
50
60
70
epoch
baseline_train
baseline_val
cnn_train
cnn_val
fcn_train
fcn_val

Conclusions and further work

- Applicating the classical steps of ML pipelines on biology research

- Data generation was very tedious (surprise!)
  - No previous data
  - Data distribution (ZEAL)

- We are working on testing the model on a much larger data set that we will include in our report.

- Models results

LUND
UNIVERSITY