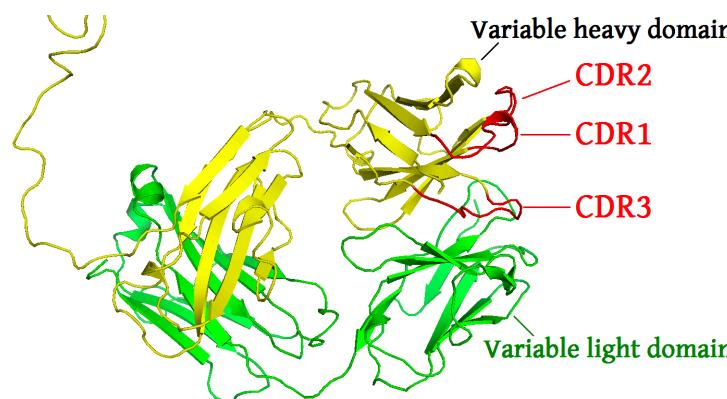


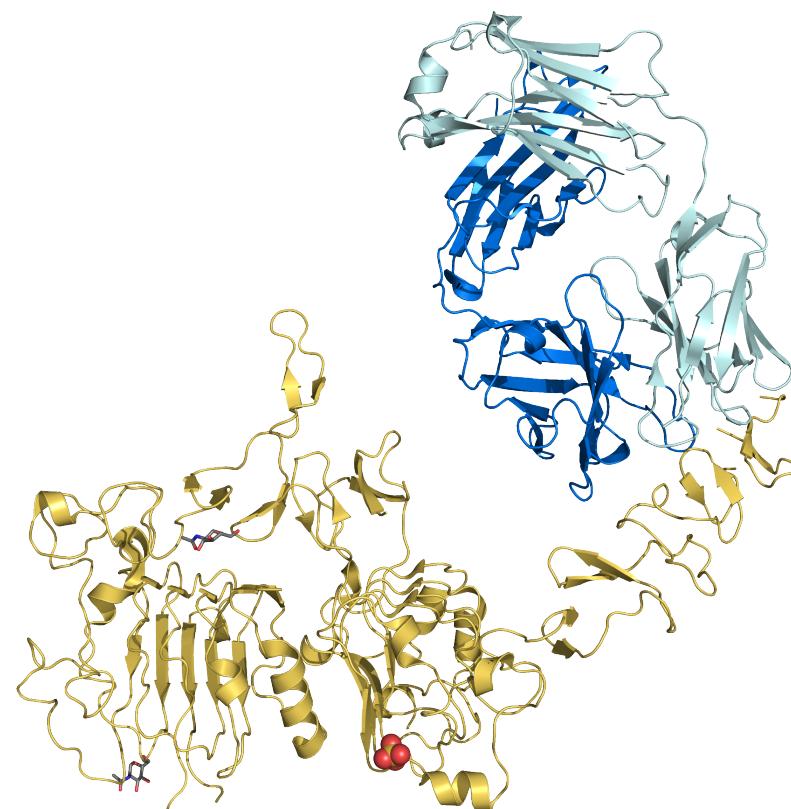
Interview Case Study – Engineering Trastuzumab

By: Jinyan Tao

Task 1: Predicting the binding status of novel Trastuzumab Variants



Trastuzumab



ERBB2

Task 1: Predicting the binding status of novel Trastuzumab Variants

Key outline:



Task 1: Predicting the binding status of novel Trastuzumab Variants

Quality control and exploratory data analysis

Data quality check

- Applied Q score to ensure sequence quality control:
- Phred Score (Q score), threshold **25**
 - **Exclude** any sequence with if any base with Q score **less than 25**
 - A rule of thumb cut-off is 20~30, apply an intermediate threshold 25 should guarantee high sequence quality

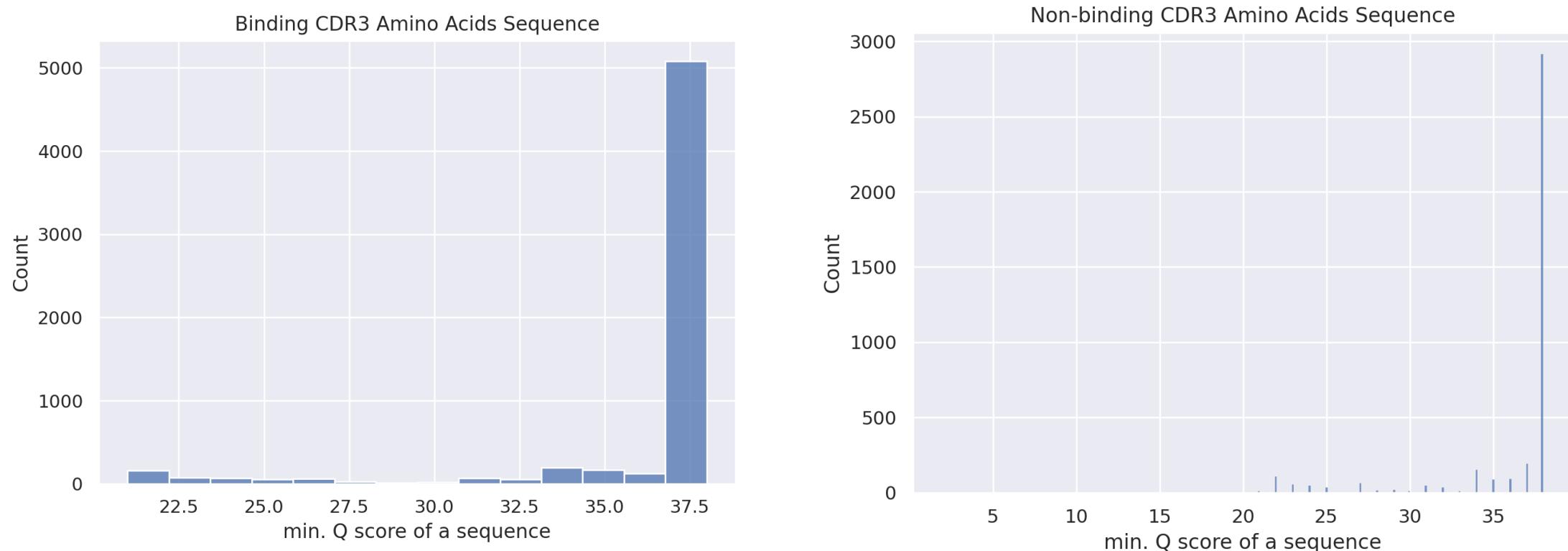


Figure 1: histograms of binders and non-binders min.Q score distribution. All sequences show min.Q score larger than 20

Data quality check

1. Applied Q score to ensure sequence quality control (QC):

- Q score with threshold **25**
 - **Exclude** any sequence with if any base with Q score **less** than **25**
 - A rule of thumb cut-off is 20~30, apply a moderate threshold 25 should guarantee high sequence quality

2. Deleted repetitive binders and non-binders with identical CDR3 amino acid sequence, however with altered CDR3 nucleotide sequence (due to codon degeneracy)

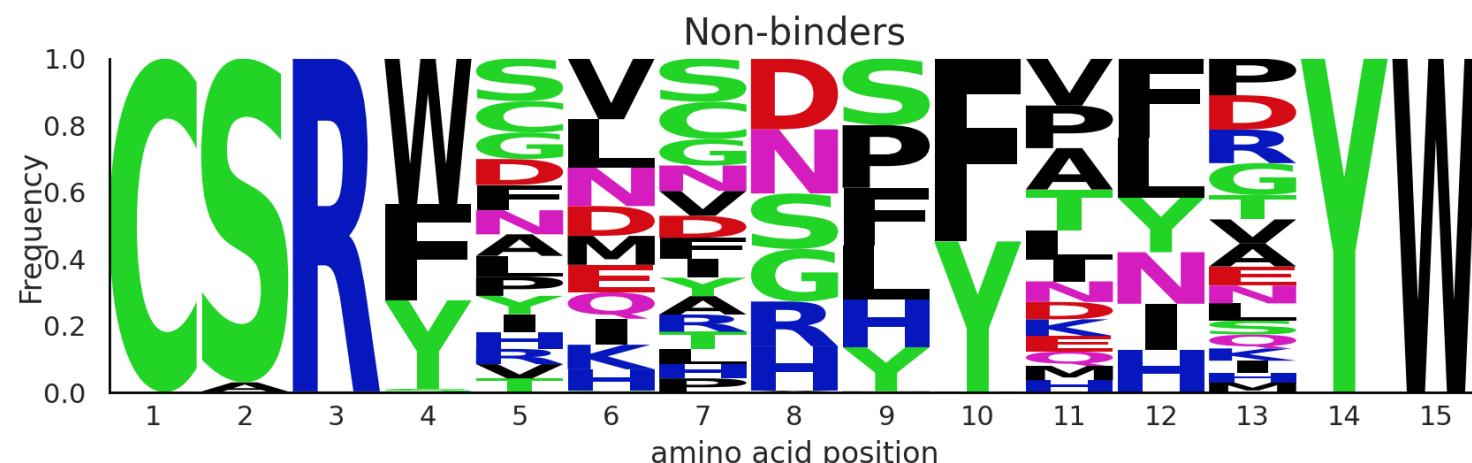
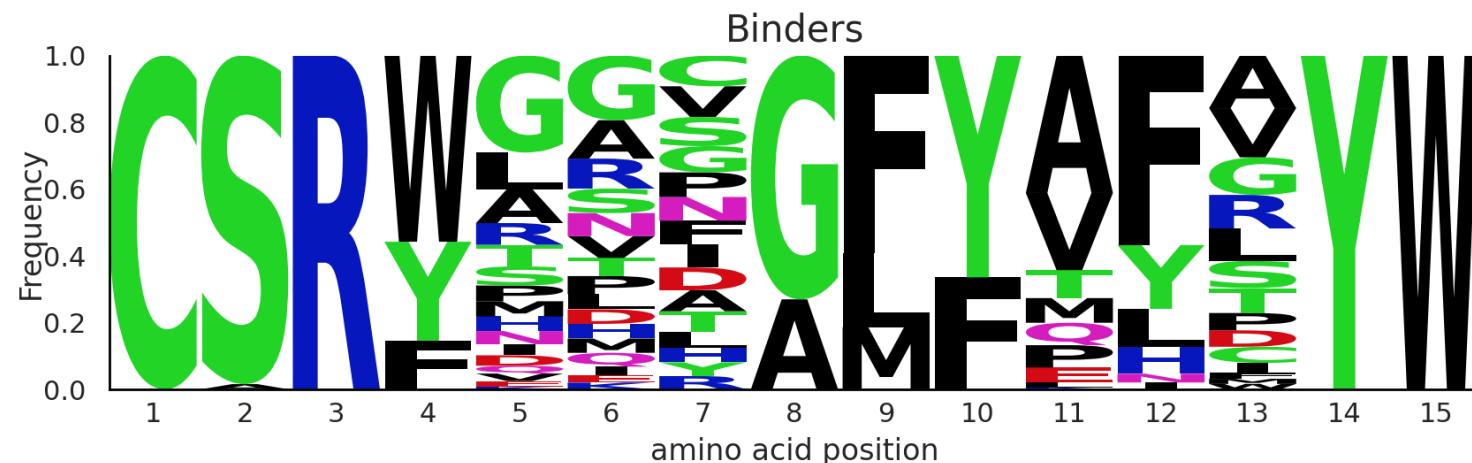
Number of Binder* entries		Number of Non-binder entries	
Before QC	After QC	Before QC	After QC
6146	3725	3888	3620

=> After two steps, data was used for downstream analysis

* note: for binders, two dataset was merged and performed quality check

Positional amino acids (AAs) diversity analysis in CDR3 sequences

- Length of AA sequences : 15 residues
- Binders and non-binders differ mainly from position 5-13**
 - Position 1-3 are mostly CSR, ending residues are YW
 - Frequency and compositions of AAs from pos. 5-13 vary



Task 1: Predicting the binding status of novel Trastuzumab Variants

Feature engineering

Feature engineering of CDR3 AA sequences representation

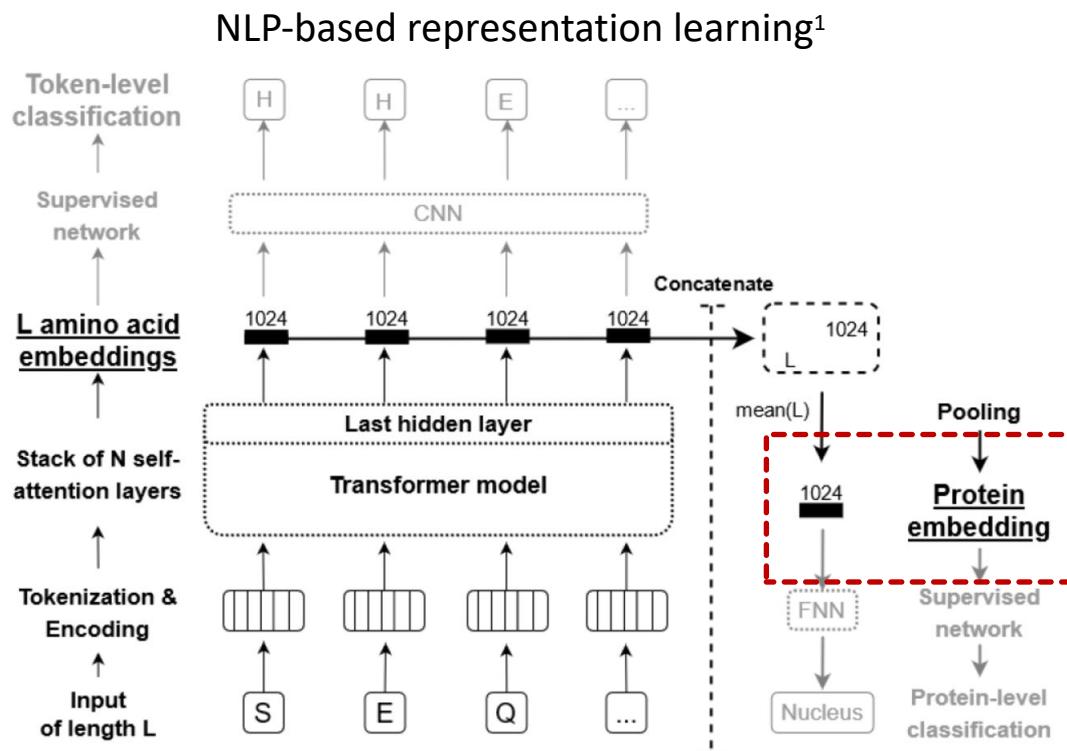
- Key challenges:
 1. To transform string-like AA sequences to machine readable numeric values
 - ✓ **Solutions:**
 - I. Natural language processing (NLP) approach to learn the data representation
 - II. Binary encoding representing ab /presence of AAs
 - III. Protein descriptors (protein fingerprints, electrostatic potential properties descriptors etc.)
 - 2. To reserve positional relationships of AA
 - ✓ **Solutions:**
 1. Positional one-hot encoding of AAs
 2. Explicit protein descriptors
 - 3. To reserve functional properties of AA
 - ✓ **Solutions:**
 - I. Implicit categorical encoding of 20 natural AAs
 - II. Explicit protein descriptors

- ❖ select AA sequences rather than nucleotide sequences, as:
 - AA impacts protein interaction directly
 - More diverse categories (20) to encode

Feature engineering of CDR3 AA sequences

- Two feature engineering approaches were selected

- Natural language processing (NLP) approach to learn 1024-vector numeric representation for each sequence**
- Positional one-hot encoding deriving 15 X 20 matrix to represent each sequence



Key ideas:

Leverage BERT (Bidirectional Encoder Representations from Transformers) model pretrained with large-scale protein sequences to generate representation

Key steps:

1. Tokenize protein sequence and create positional encoding, feed forward to BERT
2. Generate 1024-vector embedding for each AA
3. Concatenate embeddings and average them into one 1024-vector representation

Feature engineering of CDR3 AA sequences

- Two feature engineering approaches were selected
 - I. Natural language processing (NLP) approach to learn **1024-vector** numeric representation for each sequence
 - II. **Positional one-hot encoding deriving **15 X 20 matrix** to represent each sequence**

Example sequence

	A	C	D	E	F	G	B	J	MLQRS	T	V	W	G
C	0	1	0	0	0	0	0	0	...	0	0	0	0
S	0	0	0	0	0	0	0	0	...	0	0	0	0
R	0	0	0	0	0	0	0	0	...	0	0	0	0
Y	0	0	0	0	0	0	0	0	...	0	0	0	0
G	0	0	0	0	0	0	1	0	0	...	0	0	0
V	0	0	0	0	0	0	0	0	...	0	1	0	0
G	0	0	0	0	0	0	1	0	0	...	0	0	0
G	0	0	0	0	0	0	0	1	0	0	...	0	0
M	0	0	0	0	0	0	0	0	...	0	0	0	0
F	0	0	0	0	0	0	1	0	0	...	0	0	0
P	0	0	0	0	0	0	0	0	...	0	0	0	0
Y	0	0	0	0	0	0	0	0	...	0	0	1	0
V	0	0	0	0	0	0	0	0	...	0	1	0	0
T	0	0	0	0	0	0	0	0	...	0	0	0	1
W	0	0	0	0	0	0	0	0	...	0	0	1	0

=>

Key ideas:

- each row is a 20-d vector with one-hot encoding represents presence of one AA given 20 natural AAs
- 15 rows presents AA positions in the whole sequence
- **Keep positional relationships and diversity representation of AAs**

Task 1: Predicting the binding status of novel Trastuzumab Variants

Classification

Classification with NLP-learned & one-hot encoded representations: Positional one-hot encoding outperforms NLP-based representations

Dataset:

- 3725 binding sequences
- 3620 non-binding sequences
train:test = 80:20

Input:

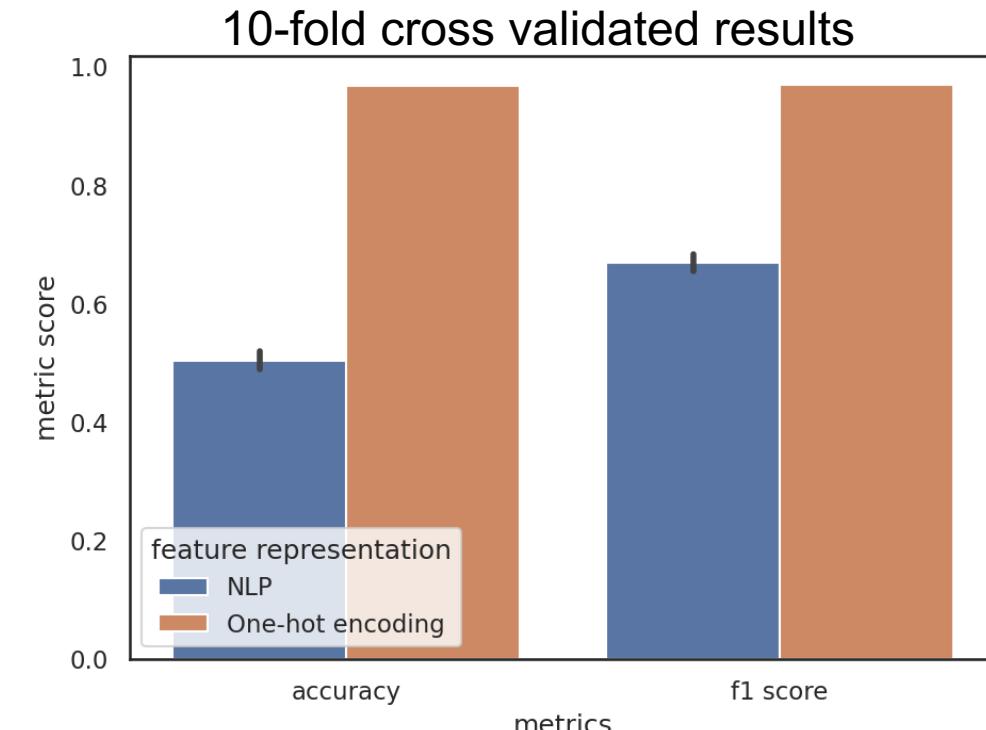
- 1024-dimensional feature vector
- 15 X 20 positional encoded matrix
 - Flattened to 300-d vector for algorithmic convenience

Model:

- random forest classifier
 - Parameters tuned by grid-search

Performance evaluation

- 10-fold cross-validated
- f1, accuracy score



Classification with NLP-learned & one-hot encoded representations:

Positional one-hot encoding (OHE) outperforms NLP-learned representations

Dataset:

- 3725 binding sequences
- 3620 non-binding sequences
train:test = 80:20

Input:

- 1024-dimensional feature vector
- 15 X 20 positional encoded matrix
 - Flattened to 300-d vector for algorithmic convenience

Model:

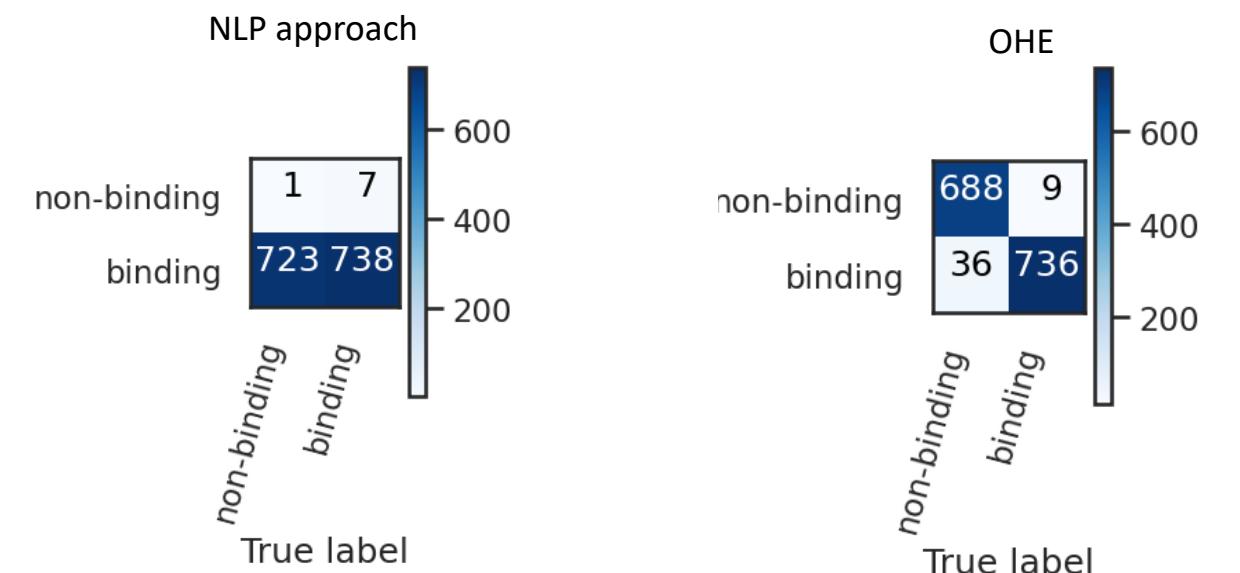
- random forest classifier
 - Parameters tuned by grid-search

Performance evaluation

- 10-fold cross-validated
- f1, accuracy score

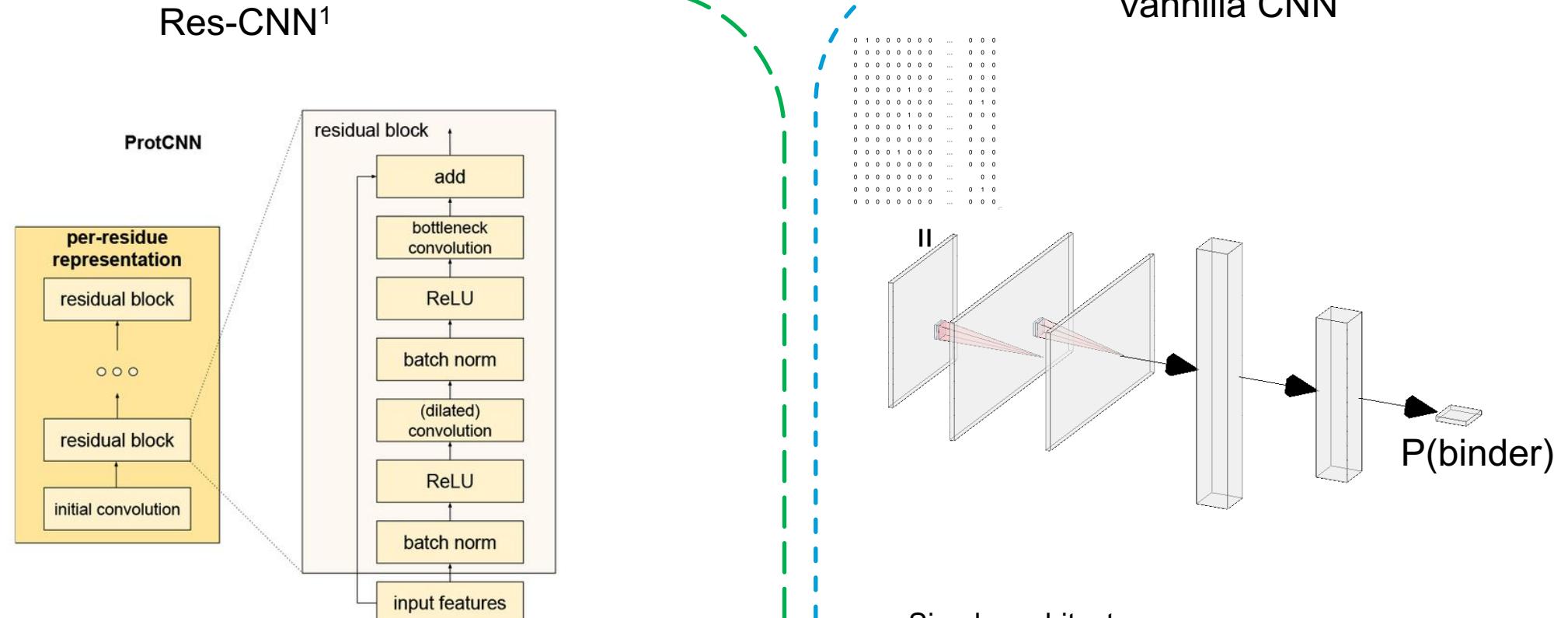
Test set result

	Test - F1 score	Test - accuracy
NLP	0.669	0.503
OHE	0.970	0.968



Therefore, positional one-hot encoder was selected to benchmark CNN performance (see discussion)

Classification with one-hot encoded representation and Convolutional Neural Networks (CNNs): Res-CNN & Vanilla CN



Key architecture: CNN + residual blocks + dilated convolution

- Residual blocks: avoid vanishing/exploding gradient problems²
- Dilated convolution : increase receptive field size, include more spatial features with less computations

Simple architecture:

- Two convolution + max pooling layers
 - Extract spatial features and reserve spatial relationships (fits AA sequence structure)
- Two dense layers with batch normalization and dropouts to avoid overfitting

Classification with one-hot encoded representation:

Res-CNN slightly outperforms vanilla CNN architecture

Dataset:

- 3725 binding sequences
 - 3620 non-binding sequences
- Train : validation : test = 60:20:20

Input:

- 15 X 20 positional encoded matrix

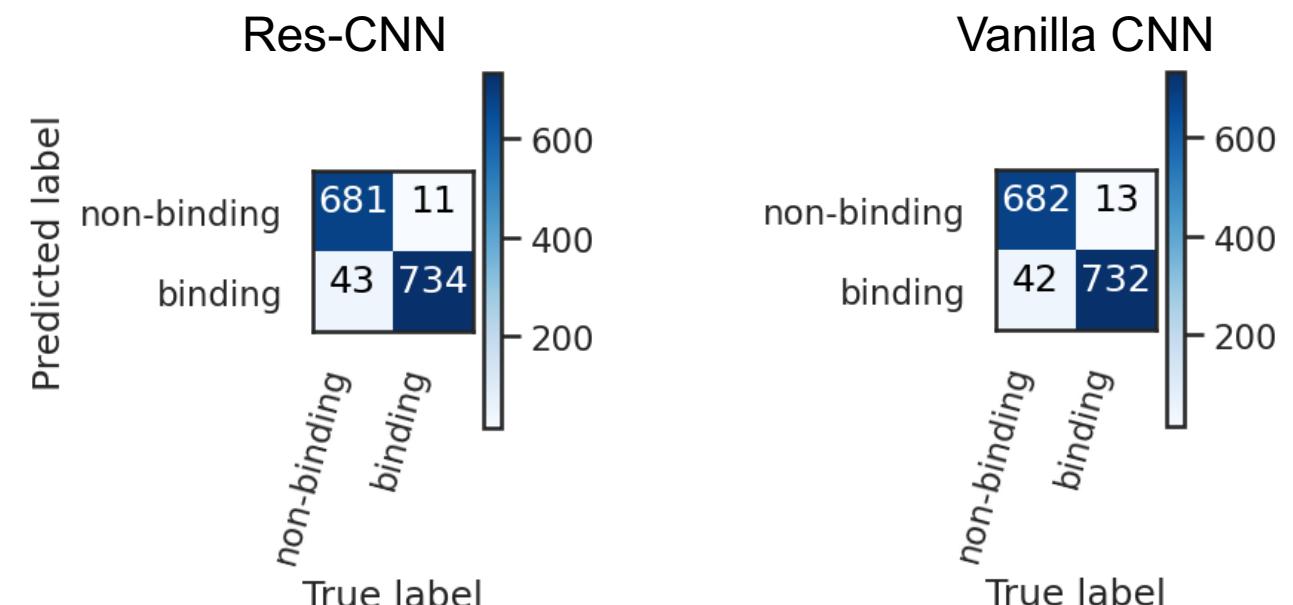
Model:

- ResCNN & CNN
 - Trained for 200 epochs with early stopping and regularization

Performance evaluation

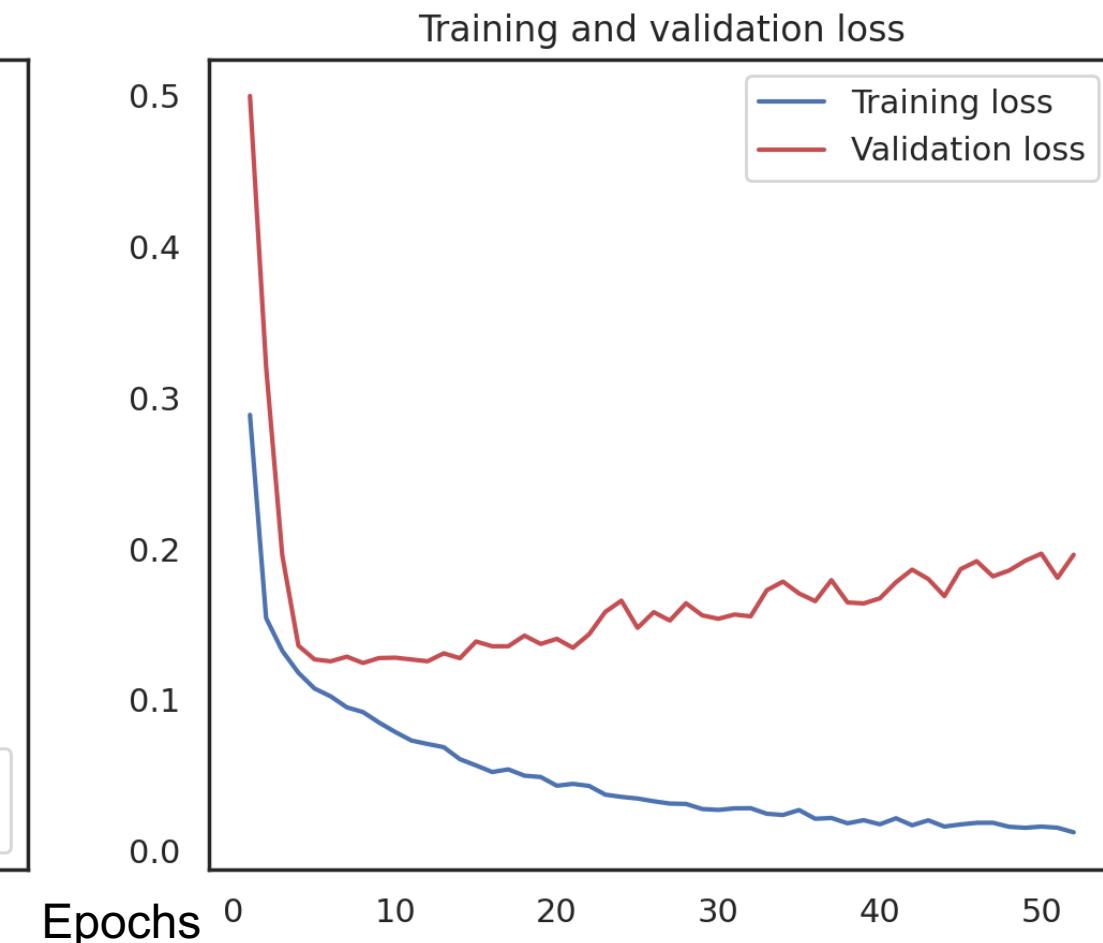
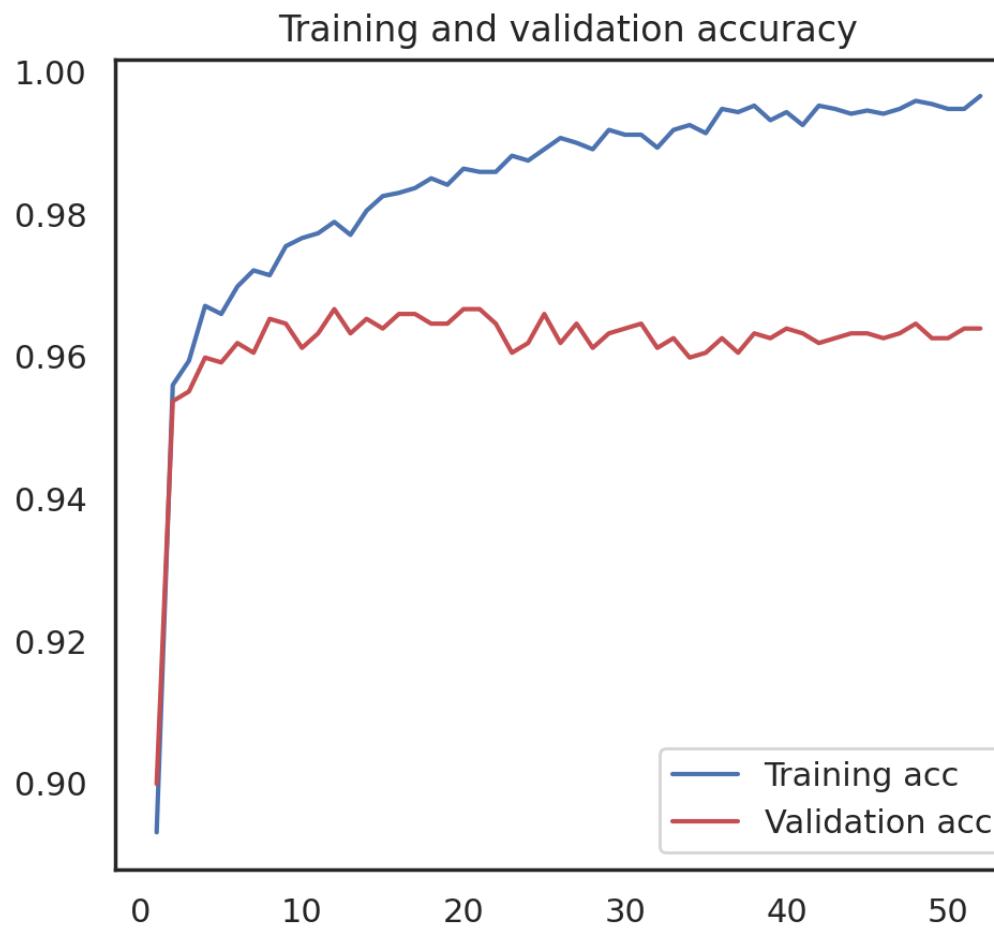
- Binary entropy loss
- F1, accuracy score

	Test - F1 score	Test - accuracy
Res-CNN	0.9634	0.9632
Vanilla CNN	0.9627	0.9626



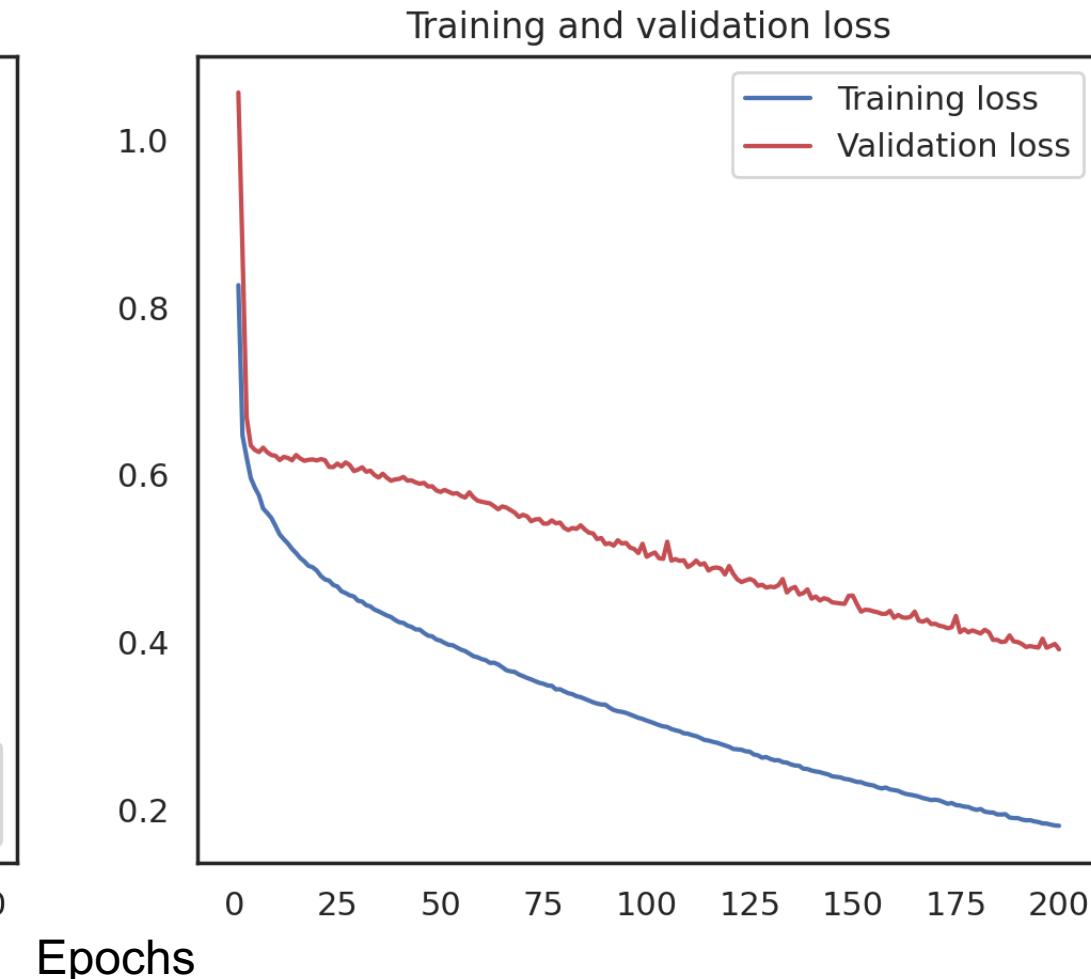
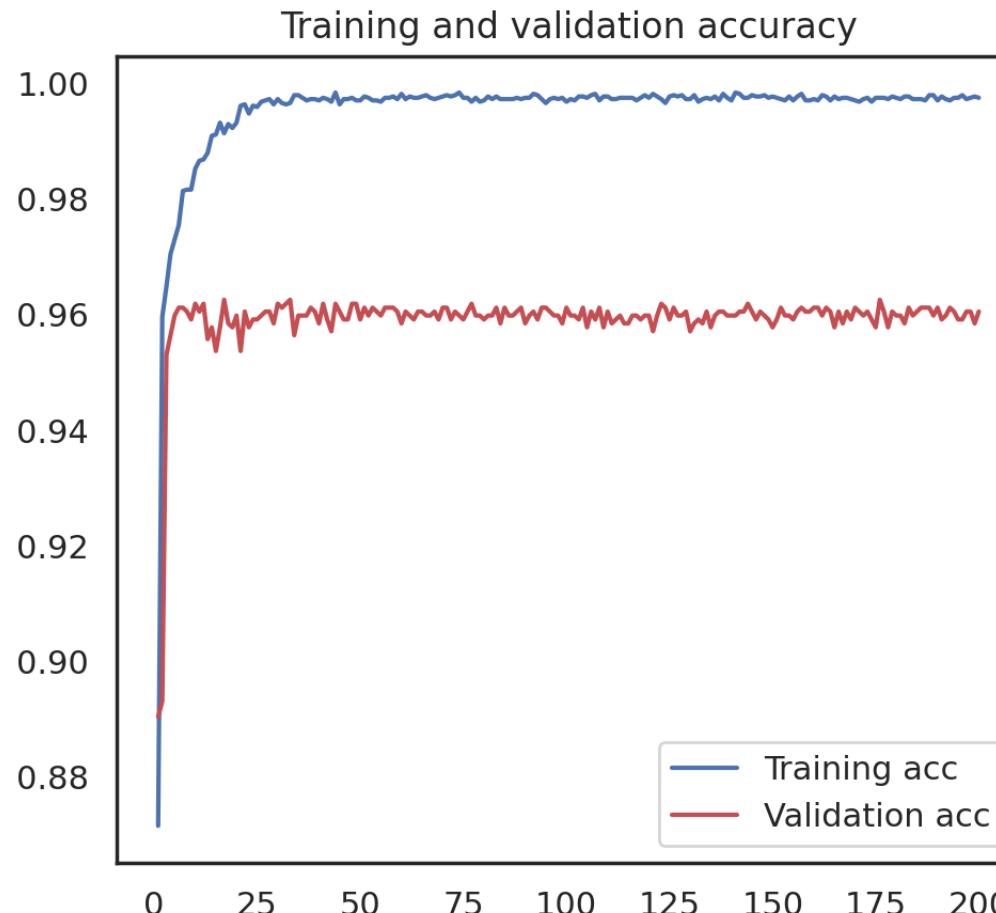
Classification with one-hot encoded representation:

Vanilla CNN architecture training history shows train loss approaches 0 with more epochs, validation loss oscillates after epoch 20



Classification with one-hot encoded representation:

Res-CNN architecture training history shows train loss approaches 0 with more epochs, validation loss keep decreasing with accuracy reaching plateau after epoch 50



Task 1: Predicting the binding status of novel Trastuzumab Variants

Conclusion:

1. Highly accurate prediction of binding status of Trastuzumab can be achieved:
 - Highest: with random forest classifier (positional one-hot encoded) f1 score 0.97, accuracy 0.968
 - Res CNN slightly outperforms vanilla CNN, slightly less performs the random forest classifier
2. CDR3 sequence embedding with positional one-hot encoding outperforms representations learned by pretrainind BERT model

	Test - F1 score	Test - accuracy
Res-CNN with OHE	0.9634	0.9632
Vanilla CNN with OHE	0.9627	0.9626
Random forest with NLP encoding	0.669	0.503
Random forest with OHE	0.970	0.968

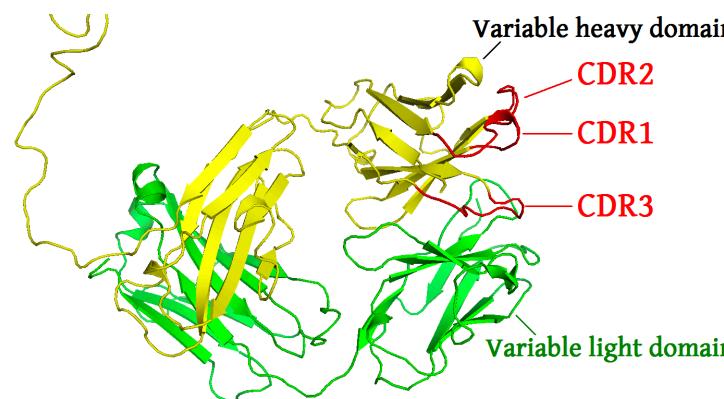
Task 1: Predicting the binding status of novel Trastuzumab Variants

Discussion & future perspective

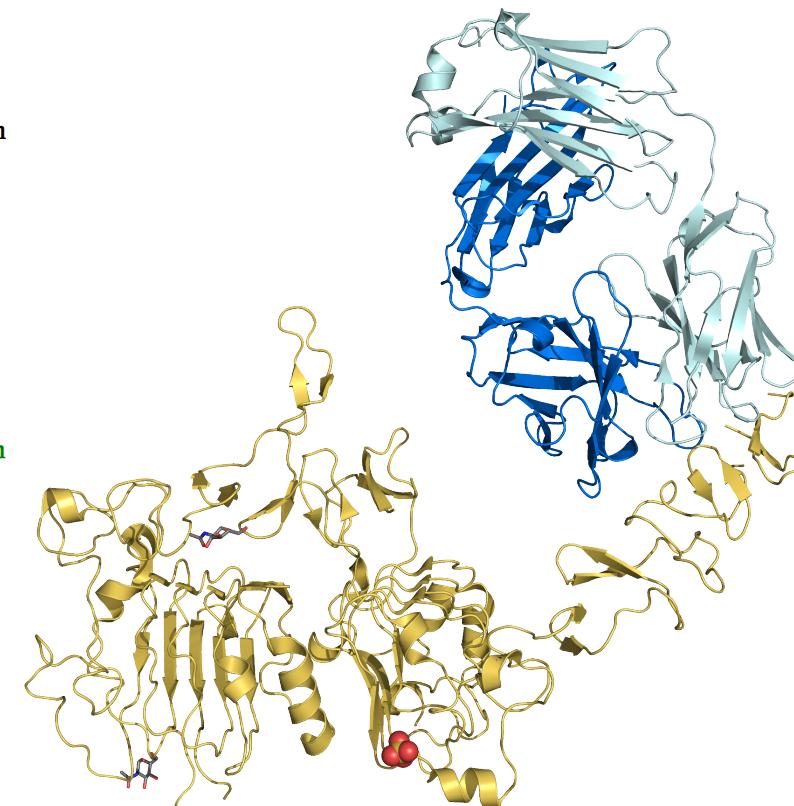
- Task 1 is a friendly start ☺, the key is to extract feature from the AA sequences
 1. Positional one-hot encoding scheme retains positional relationships between AAs in a sequence
 2. Learned sequence representations are 1024-dimensional numerical vector, might not reserve structural information
 - ⇒ Further reasonings can be investigated and postulated
 - ⇒ NLP-based approach was chosen out of personal interest/curiosity in experimenting new methods; positional OHE was chosen by its demonstrated good performance in literatures¹ and importance of AA and structural information to protein-antibody binding
 - ⇒ Other representations can be experimented: such as protein descriptors
 3. Skipping conventional tokenized & padding approach because:
 1. Sequence is too short (15 residues) to be padded
 2. Padding might lead to incomplete truncation on sequence, results in inaccurate feature extraction

Task 2: Predicting the Affinity of Novel Trastuzumab Variants Towards HER2

- Very challenging problem due to small dataset (90 entries)



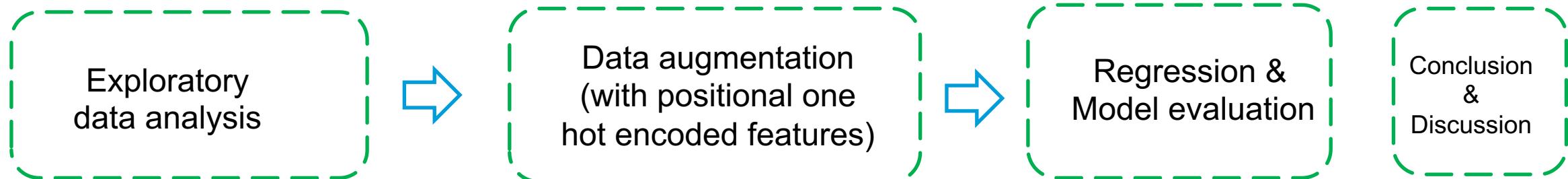
Trastuzumab



ERBB2/HER2

Task 2: Predicting the Affinity of Novel Trastuzumab Variants Towards HER2

Key outline:



Task 2: Predicting the Affinity of Novel Trastuzumab Variants Towards HER2

Exploratory data analysis

Log(natural) normalization of KD values

- KD values exhibits right skewed distribution and with large fold gap between small and large values
- Log normalization of KD values gives less sample variance and smaller fold gap
 - Sequencing quality control was skipped due to absence of sequencing quality data (assuming high quality)
 - No repetitive data entries

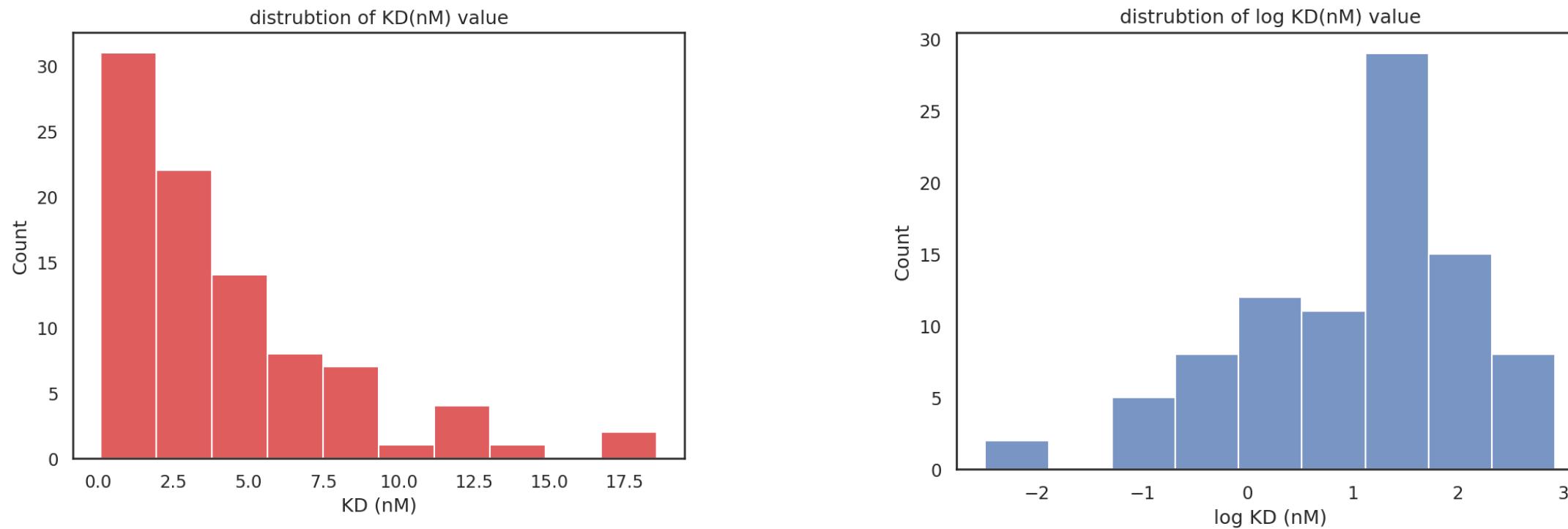
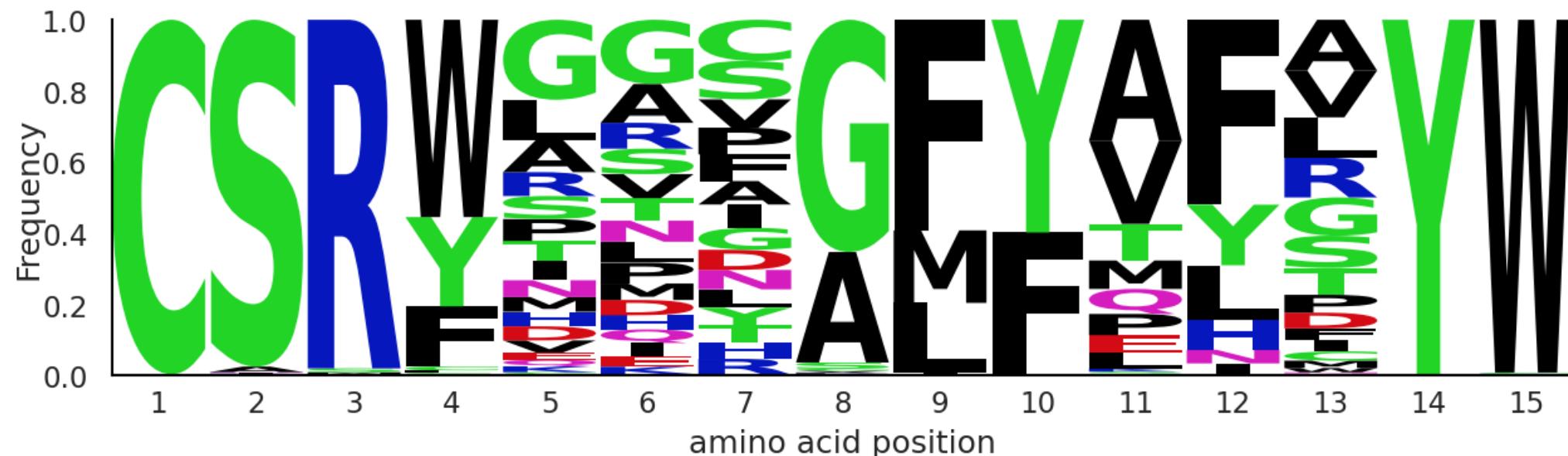


Figure: histograms of binding affinity distributions

Sequence diversity analysis

- Length of AA sequences : 15 residues
- similar residue compositions and positions as data from task 1,
 - Pos 1-3 : CSR mostly
 - Pos 14-15: YW



Task 2: Predicting the Affinity of Novel Trastuzumab Variants Towards HER2

Data Augmentation

Data augmentation scheme

- Key challenges: 90 entry points

- predict numerical values with such big range with small amount of data available
- in the case of small data regime, here presented my **decision process**:
 - Dashed arrow = no go
 - **Solid arrow, bold fonts = go**

Whether we can collect more data?

If Yes

Go and collect more data,
So far there are no other HER2-antibody
binding available after keyword search on
public database

No

1. Learning with original data
2. Learning with data augmentation
3. Transfer learning with pretrained affinity
model (no large-scale model founded)

Protein/AA sequence augmentation methods

Protein/AA sequence augmentation methods

- Literature research has benchmarked AA sequence augmentation methods and highlighted its generalizability
 - ❖ Replacing residues (with probability p) with residues of most similar chemical properties^{1,2}
 - Retains AA properties that are crucial to binding such as. Hydrogen donor/receptors

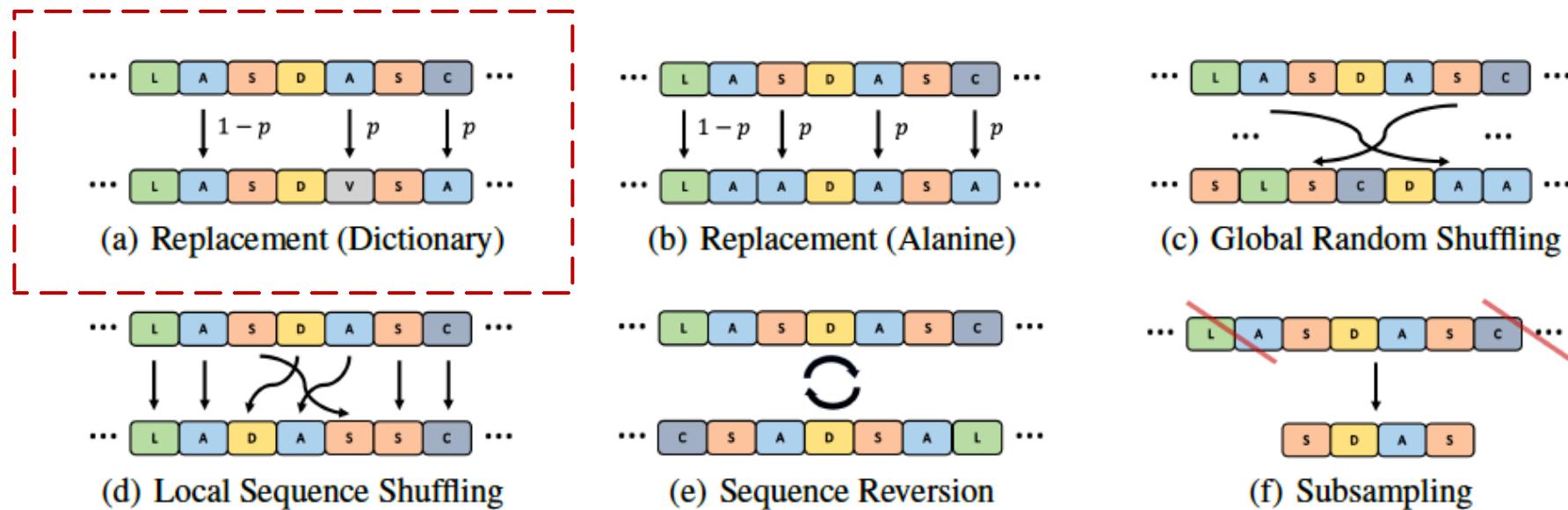


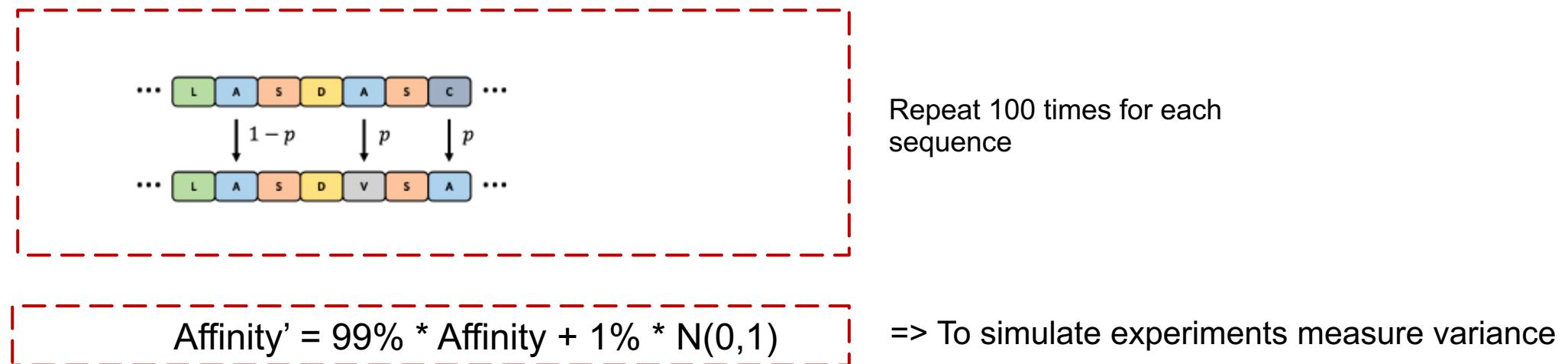
Figure 1: Diagram of data augmentations. We study randomly replacing residues (with probability p) with (a) a chemically-motivated dictionary replacement or (b) the single amino acid alanine. We also consider randomly shuffling either (c) the entire sequence or (d) a local region only. Finally, we look at (e) reversing the whole sequence and (f) subsampling to a subset of the original.

Protein/AA sequence augmentation methods with replacement of chemically similar residues

- The adaptation of replacement augmentation has following steps:
 - Randomly select 1 position in each AA sequence (typically position 4-13, as 1-3, 14-15 are highly conserved)
 - With probability 0.1, the residue will be replaced, chance of 0.9 it won't be replaced
 - Replacement follows pairwise rule: *

A:V, S:T, F:Y, K:R, C:M, D:E, N:Q, V:I, P:P, G:G, W:W, H:H, T:S, Y:F, M:C, E:D, Q:N, L:L, R:K, I:V

 - Repeat 1-3 steps 100 times for each sequence
- Corresponding affinity value is simulated from parent log KD value plus 1% Gaussian noise $\sim N(0,1)$



* note: proline (only backbone cyclic), glycine (only an H side-chain), tryptophan (indole side chain), or histidine (basic side chain with no size or chemical reaction equivalent) remains same residue

Feature engineering of CDR3 AA sequences

- Positional one-hot encoding deriving **15 X 20 matrix** to represent each sequence:
 - Selected by task 1 benchmark result

Positional one-hot encoding²

	A	C	D	E	F	G	B	J	MLQRS	T	V	W	G
C	0	1	0	0	0	0	0	0	...	0	0	0	0
S	0	0	0	0	0	0	0	0	...	0	0	0	0
R	0	0	0	0	0	0	0	0	...	0	0	0	0
Y	0	0	0	0	0	0	0	0	...	0	0	0	0
G	0	0	0	0	0	1	0	0	...	0	0	0	0
V	0	0	0	0	0	0	0	0	...	0	1	0	0
G	0	0	0	0	0	1	0	0	...	0	0	0	0
G	0	0	0	0	0	1	0	0	...	0	0	0	0
M	0	0	0	0	0	0	0	0	...	0	0	0	0
F	0	0	0	0	1	0	0	0	...	0	0	0	0
P	0	0	0	0	0	0	0	0	...	0	0	0	0
Y	0	0	0	0	0	0	0	0	...	0	0	1	0
V	0	0	0	0	0	0	0	0	...	0	1	0	0
T	0	0	0	0	0	0	0	0	...	0	0	0	1
W	0	0	0	0	0	0	0	0	...	0	0	1	0

=>

Task 2: Predicting the Affinity of Novel Trastuzumab Variants Towards HER2

Regression

Regression with original data and data augmentation:

With simplest K nearest neighbor regressor (no restriction on data size)

Dataset:

- original data: 90
 - train:test = 80:20
- Augmented training set: 4000
 - Randomly selected from 5700 augmented training data

Input:

- 15 X 20 positional encoded matrix
 - Flattened to 300-d vector for algorithmic convenience

Model:

- KNN regressor: k=4
 - Hamming loss as distance metrics
 - Similar to Levenshtein distance but normalized by length of vector
 - Parameters tuned

Performance evaluation

- Spearman rank correlation & mean squared error (MSE) (based on natural logarithm level)

Test set performance

	Rank correlation	MSE (log KD)
With augmentation	0.324 (p-value<0.01)	1.200
Without augmentation	0.690 (p-value<0.01)	0.760

KNN performs better without data augmentation

- Rank correlation shows moderate-strong correlation

Regression with original data and data augmentation:

With simplest K nearest neighbor regressor (no restriction on data size)

Dataset:

- original data: 90
 - train:test = 80:20
- Augmented training set: 4000
 - Randomly selected from 5700 augmented training data

Input:

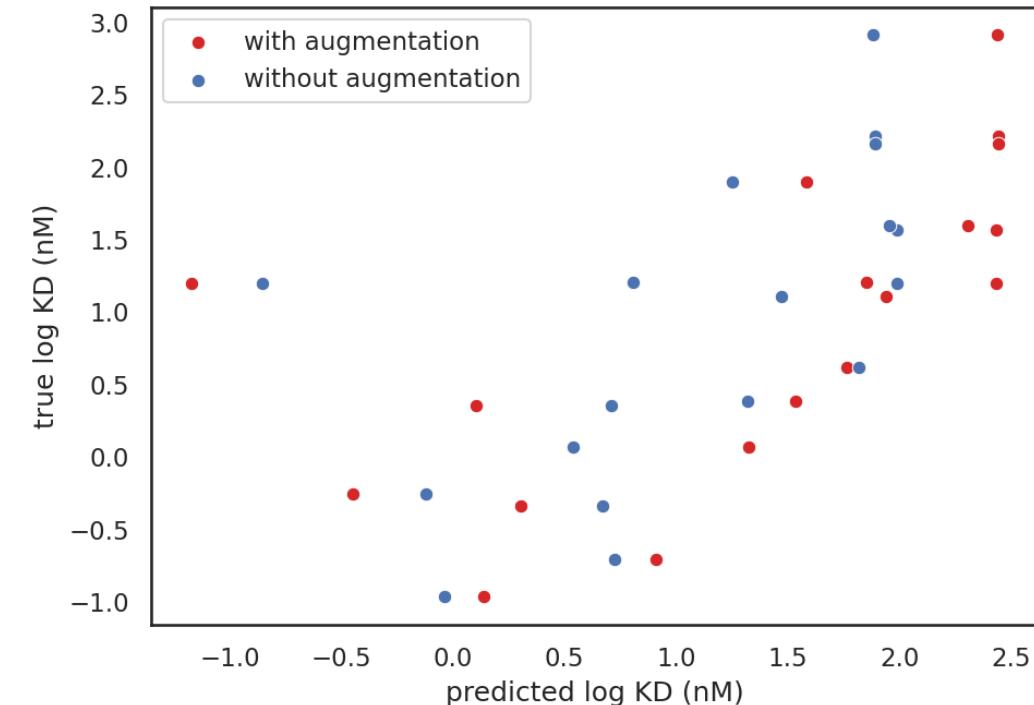
- 15 X 20 positional encoded matrix
 - Flattened to 300-d vector for algorithmic convenience

Model:

- KNN regressor : $k = 4$
 - Hamming loss as distance metrics
 - Similar to Levenshtein distance but normalized by length of vector
 - Parameters tuned

Performance evaluation

- Spearman rank correlation & mean squared error (MSE) (based on natural logarithm level)



KNN performs better without data augmentation

- predicted value shifted by larger error with augmentation
- KNN can't handle noise very well, only extrapolate values from distance relationships

Regression with data augmentation and Res-CNN

Dataset:

- Augmented training set: 4000
 - Randomly selected from 5700 augmented training data
 - Train: Test = 80:20

Input:

- 15 X 20 positional encoded matrix

Model:

- Res-CNN
 - Similar model architecture to task1 adapted to regression

Performance evaluation

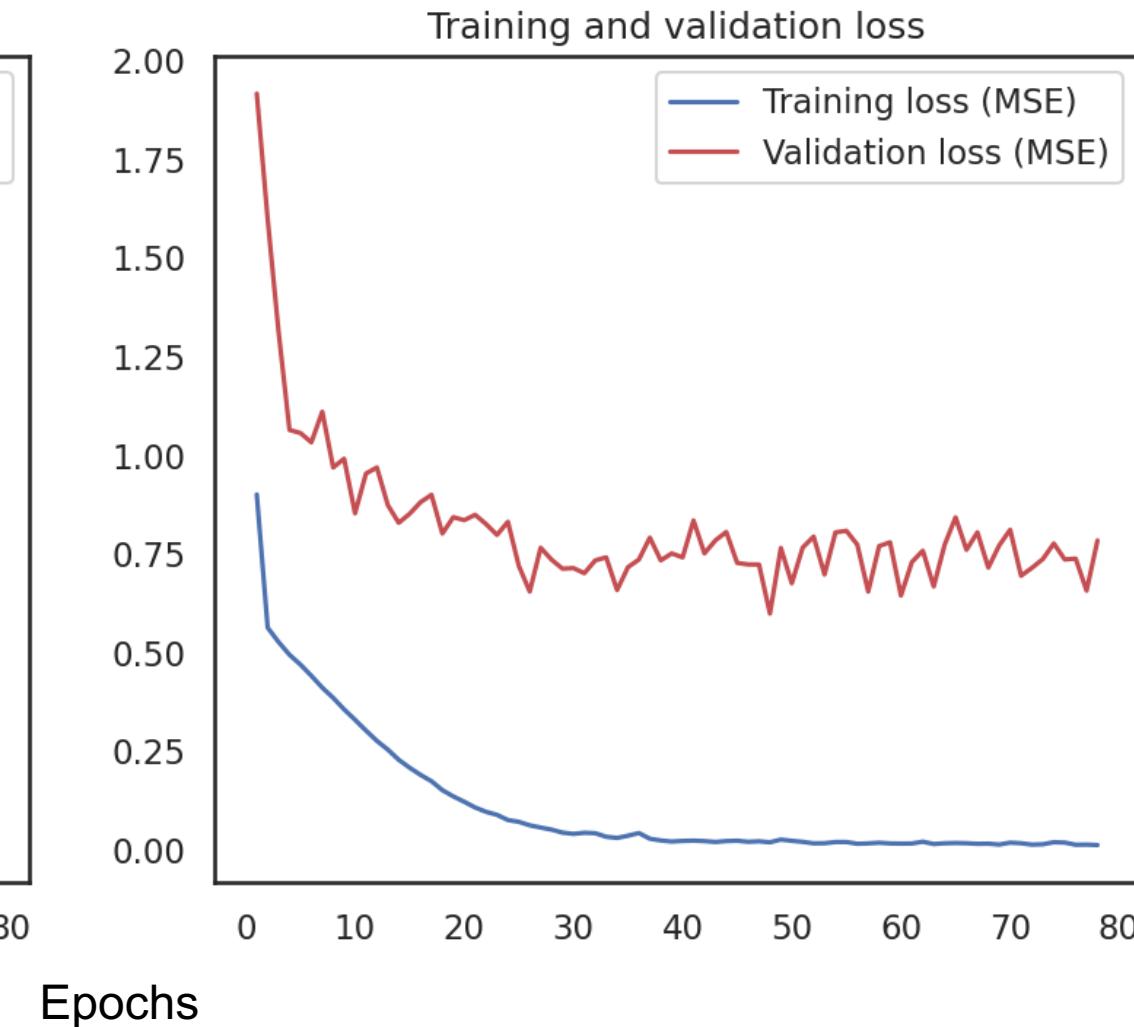
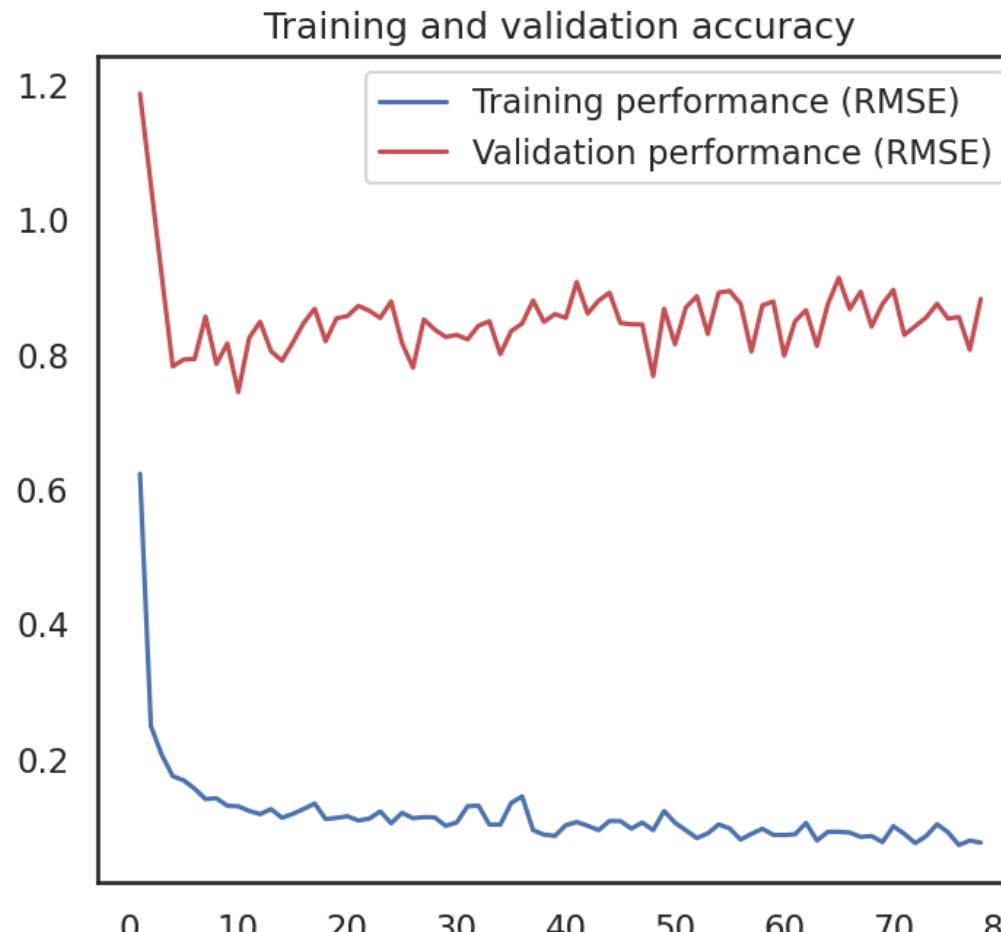
- Spearman rank correlation & mean squared error (MSE) (based on natural logarithm level)

Test set performance

	Rank correlation	MSE (log KD)
With augmentation	0.702	0.82
Without augmentation	NA, model not converge due to too small data points	

Regression with Res-CNN

Res-CNN architecture training history shows training loss approaches 0 with more epochs, validation loss oscillates, perhaps due to stuck in local minimum or overfits the training set



Regression with simple KNN slightly outperforms Res-CNN

Dataset:

- Augmented training set: 4000
 - Randomly selected from 5700 augmented training data
 - Train: Test = 80:20

Input:

- 15 X 20 positional encoded matrix

Model:

- Res-CNN
 - Similar model architecture to task1 adapted to regression

Performance evaluation

- Spearman rank correlation & mean squared error (MSE) (based on natural logarithm level)

Test set performance comparison

	Rank correlation	MSE (log KD)
KNN With augmentation	0.324	1.200
Res-CNN with augmentation	0.702	0.82
KNN without augmentation	0.690	0.760
Res-CNN Without augmentation	NA, model not converge due to too small data points	

❖ Res-CNN shows worse MSE and slightly better rank correlation

Regression with simple KNN slightly outperforms Res-CNN with data augmentation

Dataset:

- Augmented training set: 4000
 - Randomly selected from 5700 augmented training data
 - Train: Test = 80:20

Input:

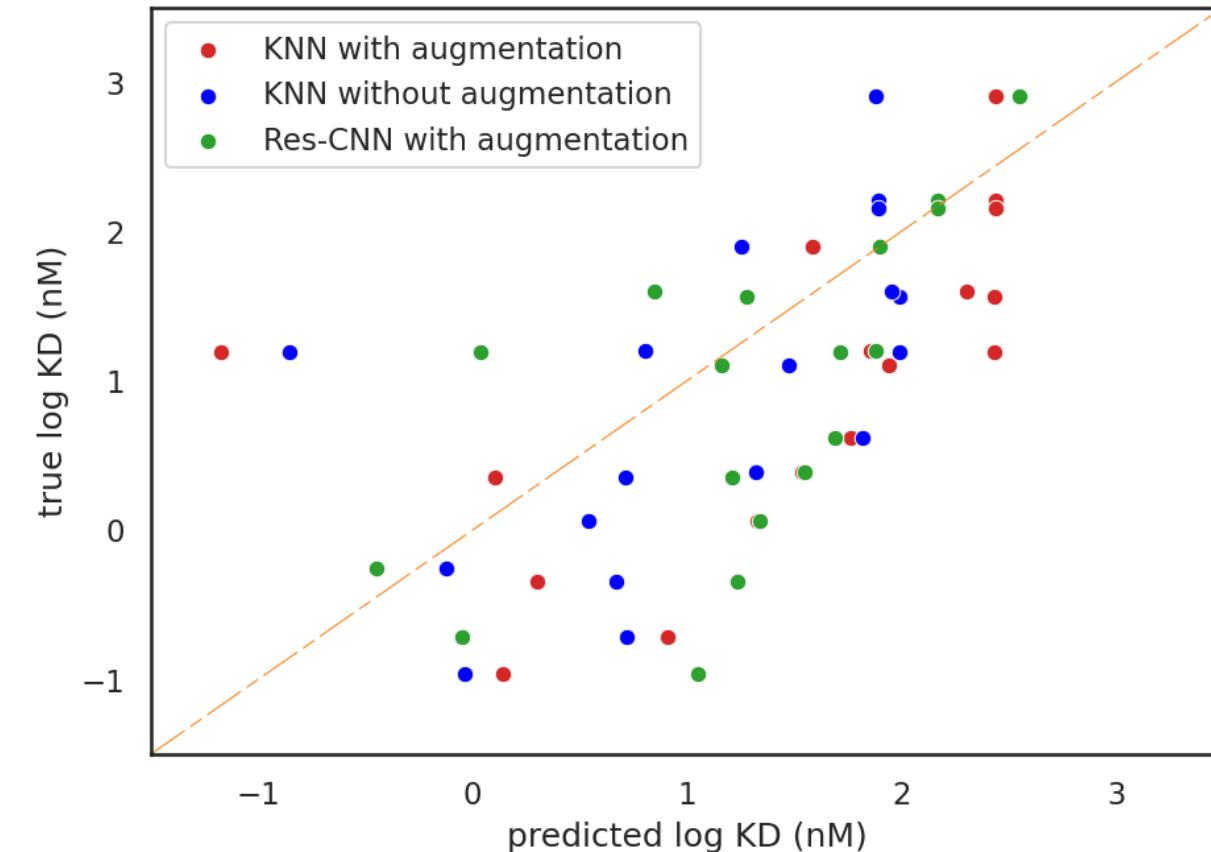
- 15 X 20 positional encoded matrix

Model:

- Res-CNN
 - Similar model architecture to task1 adapted to regression

Performance evaluation

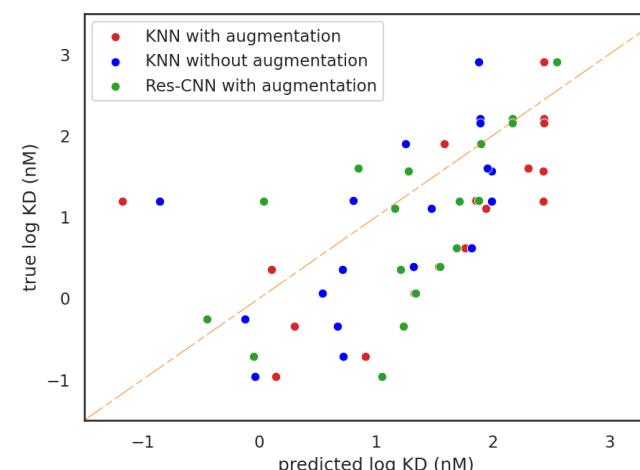
- Spearman rank correlation & mean squared error (MSE) (based on natural logarithm level)



Task 2: Predicting the Affinity of Novel Trastuzumab Variants Towards HER2

Conclusion:

1. Challenging task to predict accurately with low data regime
2. With data augmentation scheme:
 - i. KNN performs worse than without augmentation
 - i. **Pros:** Might perform well under low data regime given selection of representative training data
 - ii. **Cons:** KNN extrapolate distance relationships between data points and performs weighted sum of neighbors
 - ii. Res-CNN performs better with augmentation
 - i. **Pros:** Neural networks learns well with noisy and synthetic data
 - ii. **Cons:** Deep learning requires larger dataset



Task 2: Predicting the Affinity of Novel Trastuzumab Variants Towards HER2

Future perspective:

1. More robust data augmentation techniques can be experimented
 - e.g. masking tokens in AA sequences
 - Literature¹ highlights other robust augmentation skills however without providing example code and algorithm, challenging to implement within limited timeframe
2. Transfer learning with pretrained protein properties model
 - Not so many large-scale affinity prediction data available
 - there might be other protein properties prediction dataset available, pretrained model can be used to initiate training => needs further research
3. more robust prediction algorithms (or postulate mathematical relationships between features and affinity)
4. Check prediction outliers to diagnose wrong prediction
5. **Best approach:**
 - I. **Generate more data with experiment measurement ☺ (if possible)**
 - II. Molecular simulation to mimic protein-antibody binding affinity as ground truth...(less recommended)

Thank you!

For details of implementation (code) please check:

https://github.com/PinaColadast/CDR3_prediction_case



Back-up Slides