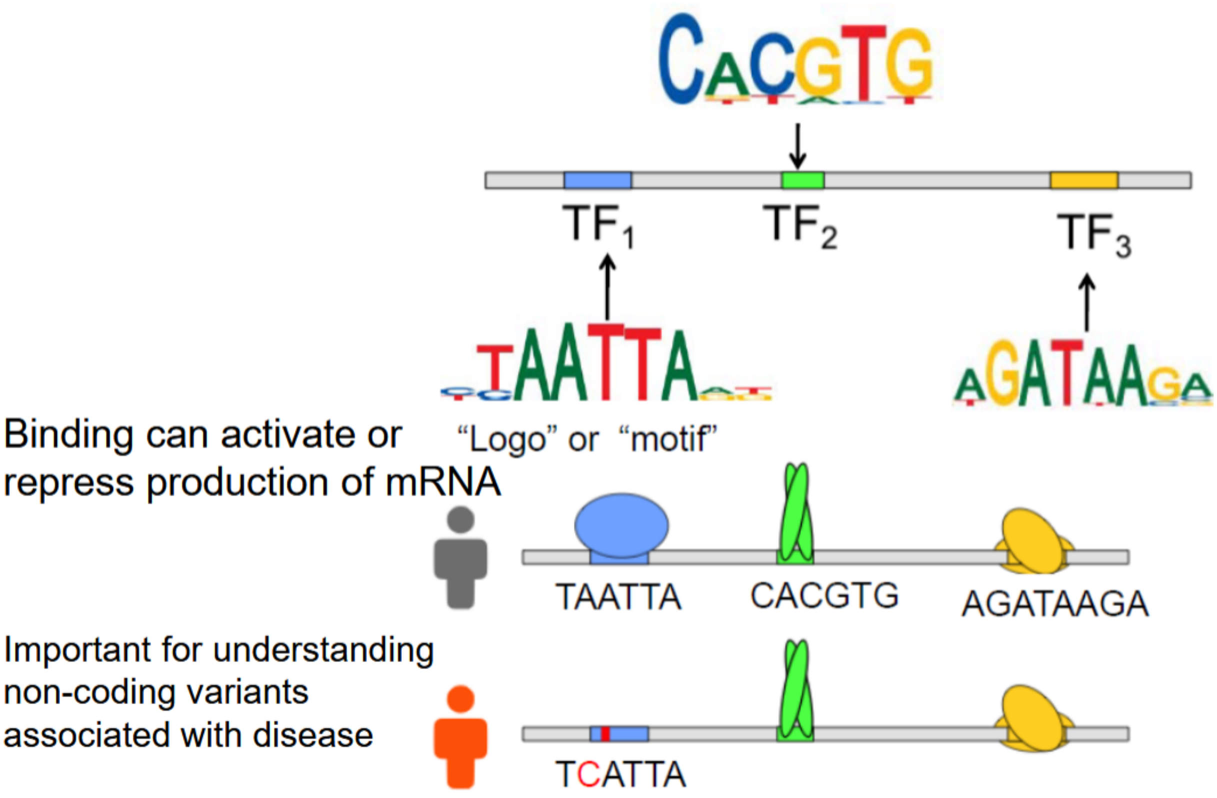
COM SCI 122 Week 8

Aidan Jan

February 28, 2025

Sequence Prediction

Understnading TF Binding Important to Interpreting Sequence Variants


CACGTG
TF1
TF2
TF3
TAATTA
AGATAAGA
Binding can activate or repress production of mRNA
"Logo" or "motif"
TAATTA
CACGTG
AGATAAGA
Important for understanding non-coding variants associated with disease
TCATTA

Using PWMs for Variant Effect Prediction

Strategy to predict variant effect with PWM

- Score reference and mutated sequence with PWM
- Check if at least one meets a score threshold
- Score change between the two sequences

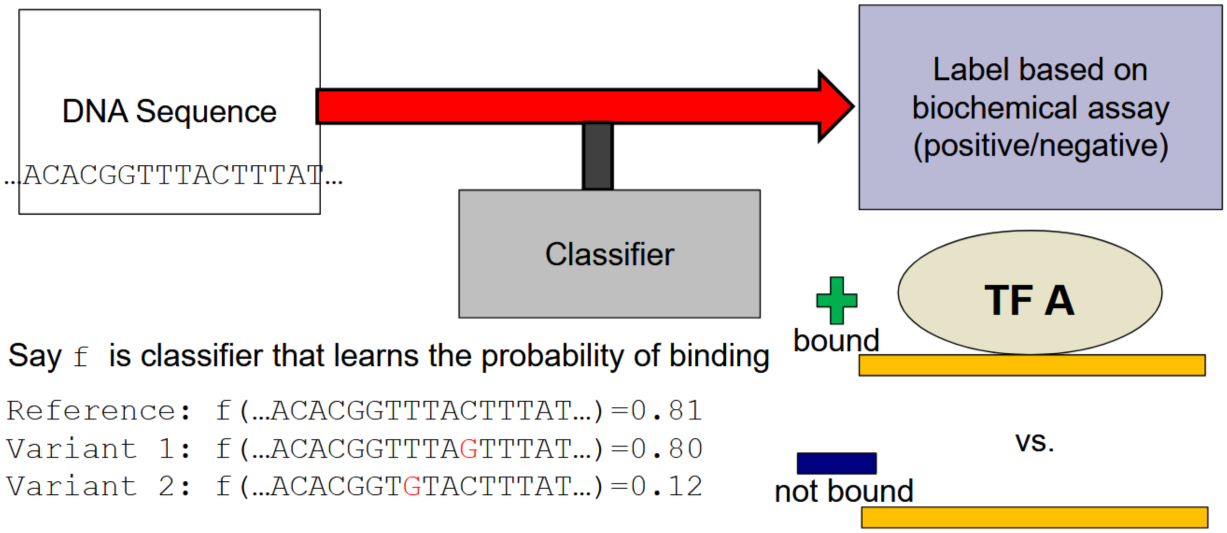
Suppose the reference sequence is `CAT`, under the PWM model below how would you rank the three mutations in terms of greatest predicted impact?

CAT CAT CAT

CTT CAA CAG

	1	2	3
A	1/7	4/7	1/7
C	4/7	1/7	2/7
G	1/7	1/7	2/7
T	1/7	1/7	2/7

Sequence to Biochemical Assay Prediction



Say f is classifier that learns the probability of binding

```
Reference:  f(…ACACGGTTTACTTTAT…)=0.81
Variant 1:  f(…ACACGGTTTAGTTTAT…)=0.80
Variant 2:  f(…ACACGGTGTACTTTAT…)=0.12
```

Question: How could such a classifier be used for variant effect prediction?

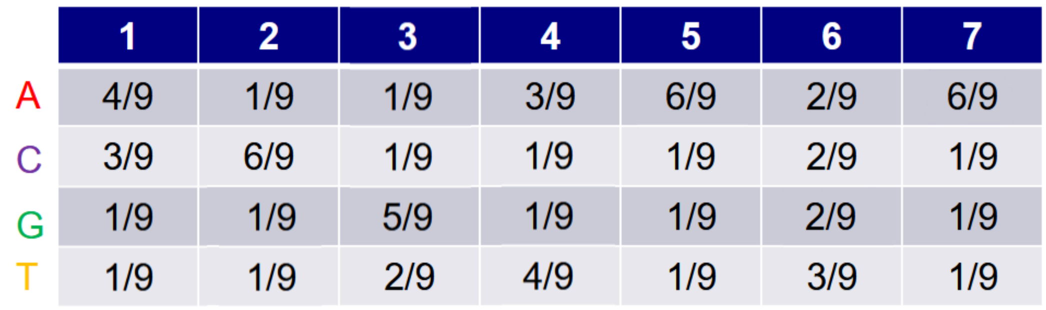
- Compare prediction probability for reference and mutation

Prediction of Binding Based on a single PWM

- Scan a sequence based on a single PWM (known or discovered)
- Predict based on recorded maximum PWM score for any sub-sequence

Scoring a Sequence with a PWM

Score each sub-sequence that is length of the PWM and record score of the subsequence with the best match.



	1	2	3	4	5	6	7
A	4/9	1/9	1/9	3/9	6/9	2/9	6/9
C	3/9	6/9	1/9	1/9	1/9	2/9	1/9
G	1/9	1/9	5/9	1/9	1/9	2/9	1/9
T	1/9	1/9	2/9	4/9	1/9	3/9	1/9

$\boxed{\text{ACTTATC}}\text{GA}$ $$\frac{4}{9}\times\frac{6}{9}\times\frac{2}{9}\times\frac{4}{9}\times\frac{6}{9}\times\frac{3}{9}\times\frac{1}{9}=\boxed{0.000723}$$

$\text{A}\boxed{\text{CTTATCG}}\text{A}$ $$\frac{3}{9}\times\frac{1}{9}\times\frac{2}{9}\times\frac{3}{9}\times\frac{1}{9}\times\frac{2}{9}\times\frac{1}{9}=0.00000753$$

$\text{AC}\boxed{\text{TTATCGA}}$ $$\frac{1}{9}\times\frac{1}{9}\times\frac{1}{9}\times\frac{4}{9}\times\frac{1}{9}\times\frac{2}{9}\times\frac{6}{9}=0.0000100$$

In this case, the highest score is 0.000723. Now, we apply a variant to the sequence, and do it again.

$\boxed{\text{GCTTATC}}\text{GA}$ $$\frac{1}{9}\times\frac{6}{9}\times\frac{2}{9}\times\frac{4}{9}\times\frac{6}{9}\times\frac{3}{9}\times\frac{1}{9}=0.00018075$$

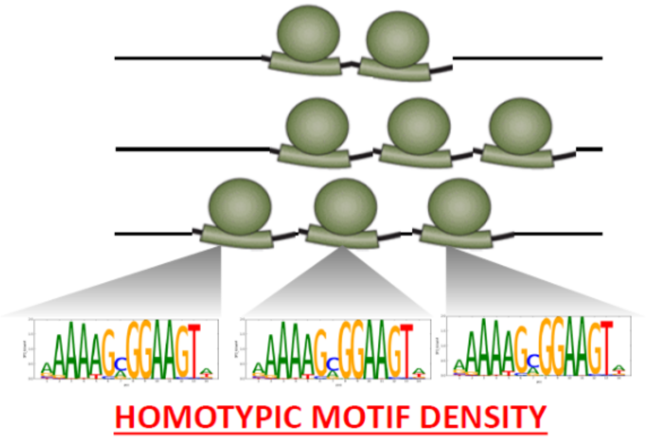
$\text{G}\boxed{\text{CTTATCG}}\text{A}$ $$\frac{3}{9}\times\frac{1}{9}\times\frac{2}{9}\times\frac{3}{9}\times\frac{1}{9}\times\frac{2}{9}\times\frac{1}{9}=0.00000753$$

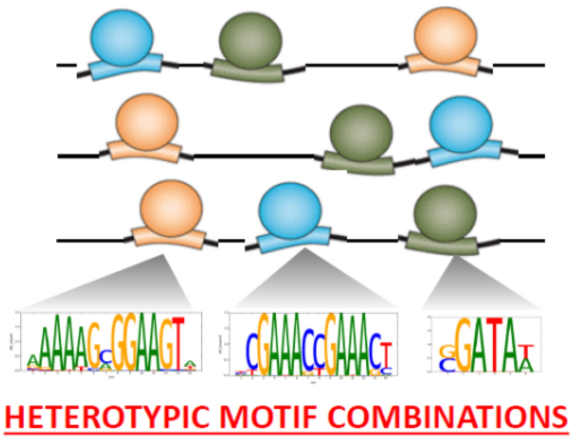
$\text{GC}\boxed{\text{TTATCGA}}$ $$\frac{1}{9}\times\frac{1}{9}\times\frac{1}{9}\times\frac{4}{9}\times\frac{1}{9}\times\frac{2}{9}\times\frac{6}{9}=0.0000100$$

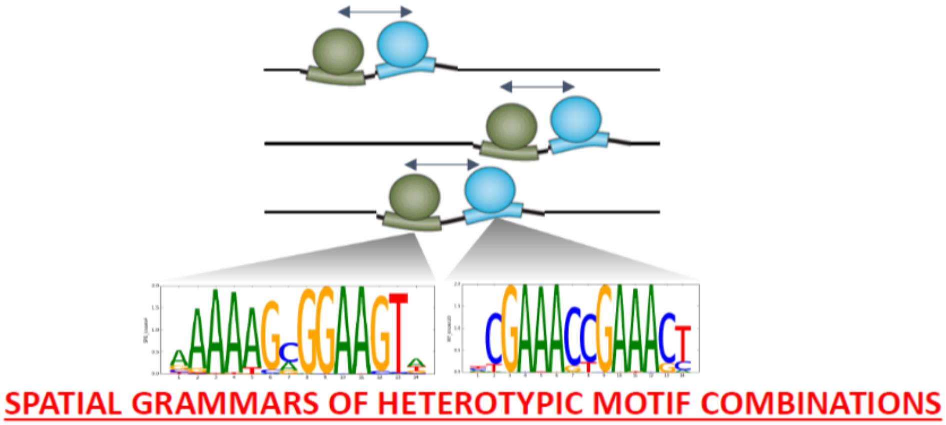
Limitations of Binding Predictions Based on PWM Scanning

- Many motif instances are not actually bound and there is additional information in sequence context for predicting binding
- **Question:** Suppose we have a ChIP-seq experiment for a transcription factor, what is another strategy we could use to predict transcription factor binding?
  - Through supervised machine learning models

Properties of Regulatory Sequences Not Captured by a PWM







- **Homotypic Motif Density:** Regulatory sequences often contain **more than one binding instance** of a TF resulting in **homotypic clusters of motifs of the same TF**
- **Heterotypic Motif Combinations:** Regulatory sequences often bound by **combinations of TFs** resulting in **heterotypic clusters of motifs of different TFs**
- **Spatial Grammars of Heterotypic Motif Combinations:** Regulatory sequences are often bound by **combinations of TFs** with specific **spatial and positional constraints** resulting in distinct **motif grammars**

K-mer based / Logistic Regression

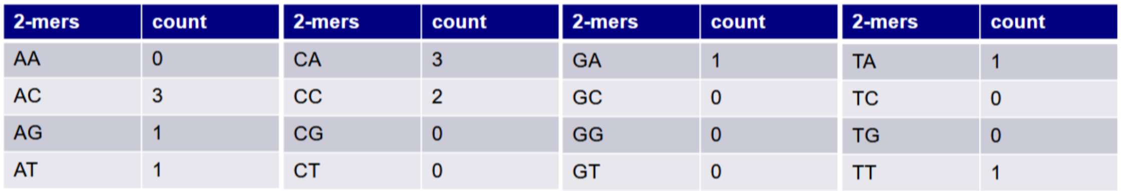
Defining Features for Classification

- Many standard machine learning classifiers take an explicit set of features.
- **Question:** How to define features for a DNA sequence?

K-mer Features

- K-mer features - count for each substring of length k of how often it occurs in the sequence.
- Consider sequence `ACACCATTAGACCA`, and $k = 2$.

- If we count the 2-mers, we get:

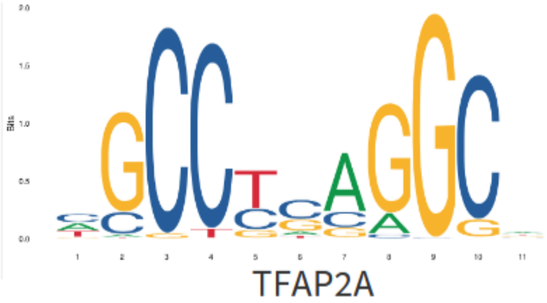


2-mers	count	2-mers	count	2-mers	count	2-mers	count
AA	0	CA	3	GA	1	TA	1
AC	3	CC	2	GC	0	TC	0
AG	1	CG	0	GG	0	TG	0
AT	1	CT	0	GT	0	TT	1

- **Question:** How many possible k-mers for a value of k?
  - 4^k.
- A small k might not be informative enough to capture some motifs, but also, a large k might be observed too infrequently.
- This raises scability challenges.

Extending K-mer Features

For some transcription factors there are degenerate positions between informative positions



- **Question:** What are limitations of regular k-mer features in such cases?
- **Question:** What can be done instead?

Gapped K-mer Features

`ACACCATTAGACCA`

- An alternate strategy is to define gapped k-mers
  - Allow k fixed characters
  - Allow m wild card positions
  - $k + m$ positions total
- Example of a gapped k-mer for $k = 2$ and $m = 1$; "?" denotes wild card
  - `A?A` - what is the frequency this gapped k-mer in the above sequence? (2, since `ACA` and `AGA` occur once each.)

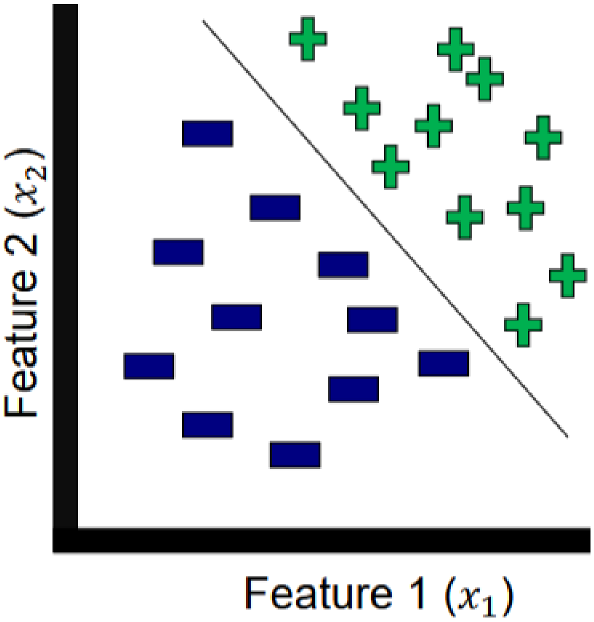
Classifier

- Many classifiers could be applied to discriminate two classes (e.g., logistic regression, random forest, SVM, etc.)
- We will discuss logistic regression

Logistic Regression

- A probabilistic classification based on weighted linear combination of features, with two features:

$$z = w_0 + w_1 x_1 + w_2 x_2$$



- In the example, different combinations of feature values that lie on the same line determine by w will have the same classification probability.
- Labels are binary so different than the setting for linear regression.

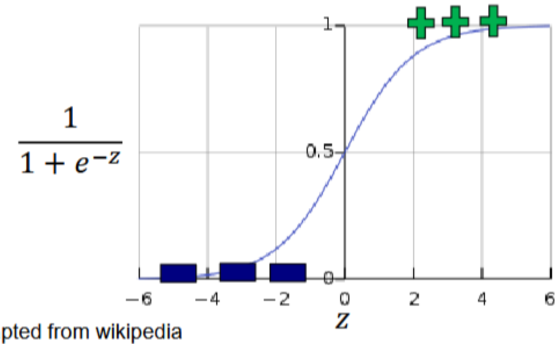
Binary Logistic Regression

- Let Y be a binary variable for the label, e.g.,
  - $Y = 1$ is a transcription factor binds a sequence
  - $Y = 0$ if it does not.
- Let X be a variable for a vector x of d input features about the sequence based on which we want to make predictions (e.g., k-mer features)
- Let w be a vector of feature weights which we will learn.

$$P(Y=1|X=x,w) = \frac{1}{1+e^{-\underbrace{(w_0+w_1x_1+\cdots+w_dx_d)}}}$$

Let
$z = (w_0 + w_1x_1 + \cdots + w_dx_d)$

Logistic function stays bounded between 0 and 1



Logistic image adapted from wikipedia

Then,

$$P(Y=1|X=x,w) = \frac{1}{1+e^{-(w_0+w_1x_1+\cdots+w_dx_d)}}$$

$$P(Y=0|X=x,w) = 1 - P(Y=1|X=x,w) = \frac{e^{-(w_0+w_1x_1+\cdots+w_dx_d)}}{1+e^{-(w_0+w_1x_1+\cdots+w_dx_d)}}$$

$$\log\frac{P(Y=1|X=x,w)}{P(Y=0|X=x,w)} = w_0 + w_1x_1 + \cdots + w_dx_d$$

Log-odds is a linear function of the input features.

The Logistic Loss Function

$$\sum_{i=1}^{t} -y_i \log(P(Y=1|X=x_i,w)) - (1-y_i)\log(P(Y=0|X=x_i,w))$$

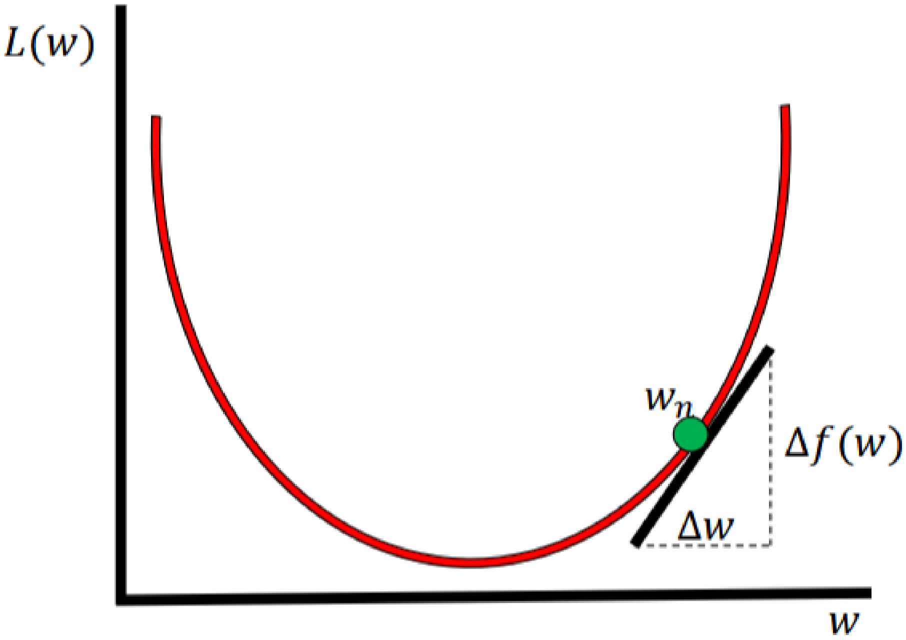
- w is set to minimize the above expression. Equivalent to maximizing the log-likelihood of the data.
- y_i is the label of the i-th data point.
- If $y_i = 1$, the above expression within sum simplifies to
$$-\log(P(Y=1|X=x_i,w))$$
- If $y_i = 0$, the above expression within sum simplifies to
$$-\log(P(Y=0|X=x_i,w))$$
- **In general, requires numerical methods such as gradient descent to optimize.**

Gradient Descent

$$w_{n+1} = w_n + \gamma\frac{\partial L(w)}{\partial w}$$

- γ is the learning rate
- L is the loss function

- w are the weight(s)



Logistic Loss Function with Ridge (L_2) Regularization

$$\sum_{i=1}^{t} -y_i \log(P(Y=1|X=x_i,w)) - (1-y_i)\log(P(Y=0|X=x_i,w)) + \lambda \sum_{i=1}^{d} w_i^2$$

- Compared to the original logistic loss function, we added an extra 'regularization' term.
- λ is a non-negative parameter

Without regularization weights could be arbitrarily large in magnitude both positive and negative and may not generalize well to unseen data.

Logistic Loss Function with Lasso (L_1) Regularization

$$\sum_{i=1}^{t} -y_i \log(P(Y=1|X=x_i,w)) - (1-y_i)\log(P(Y=0|X=x_i,w)) + \lambda \sum_{i=1}^{d} |w_i|$$

Lasso encourages sparsity meaning typically only a subset of features have non-zero weight.

Limitations

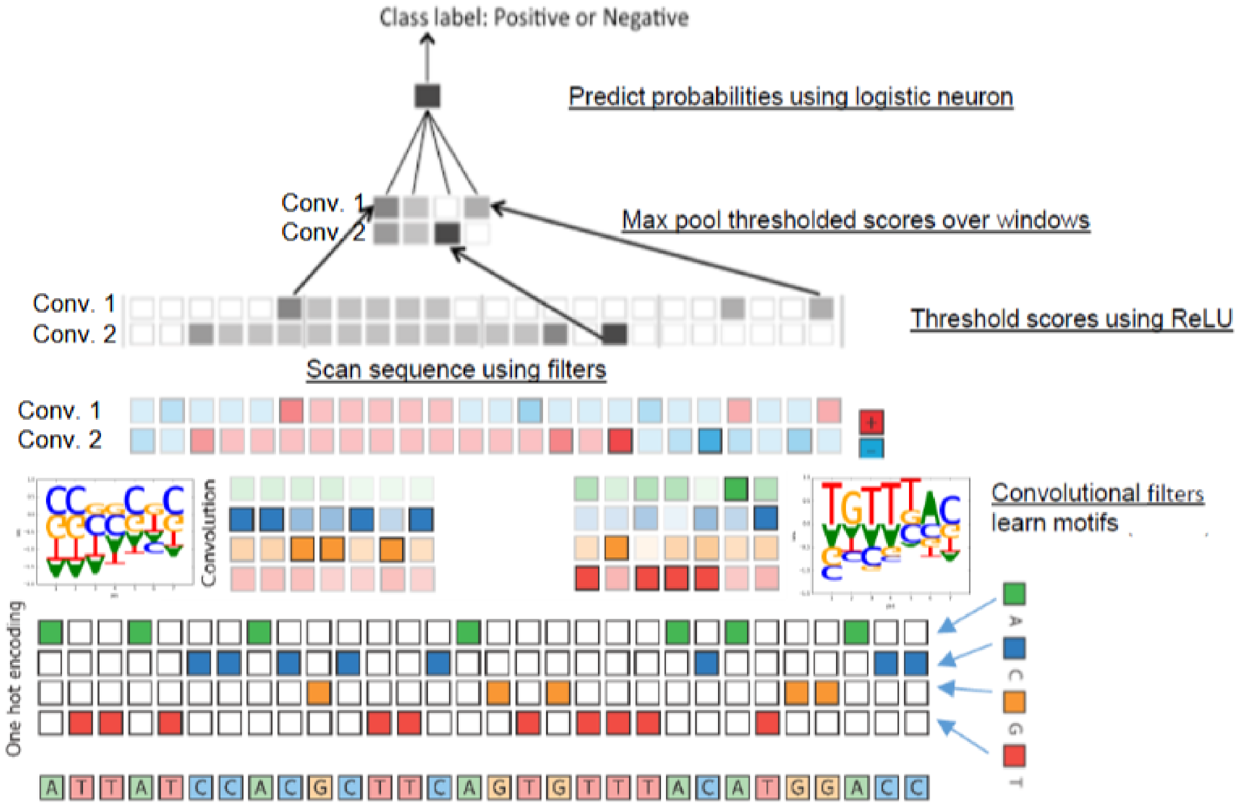
What are potential limitations of k-mer based approach and/or logistic regression that we may want to address for classification?

- Does not offer flexibility of PWM like representation
- Does not capture spatial constraints
- May not capture certain types of combinatorial relationships

Convolutional Neural Networks (CNNs) / Deep Learning

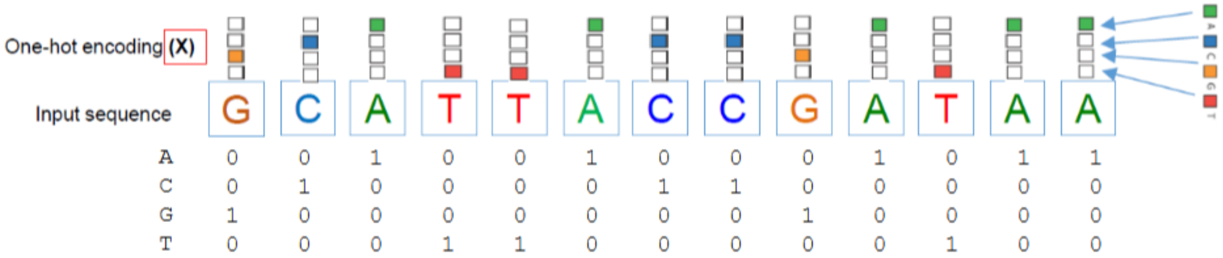
- Convolutional Neural Networks exploit structure of problem with specially designed hidden layers
- Transformed computer vision field
- Applications found in many other domains
- Avoids pre-specifying features
- Can potentially learn higher level features

CNN for Sequence Based Prediction


Class label: Positive or Negative
Predict probabilities using logistic neuron
Conv. 1
Conv. 2
Max pool thresholded scores over windows
Conv. 1
Conv. 2
Threshold scores using ReLU
Scan sequence using filters
Conv. 1
Conv. 2
+
−
Convolution
Convolutional filters learn motifs
One hot encoding
A
C
G
T
A T T A T C C A C G C T T C A G T G T T T A C A T G G A C C

- Bottom of image: One-hot encoding
- Next layer: Convolutional Filters
- Next layer: CNN Filters
- Next two layers: ReLU() and maxpool()
- Top layer: Fully saturated neural network

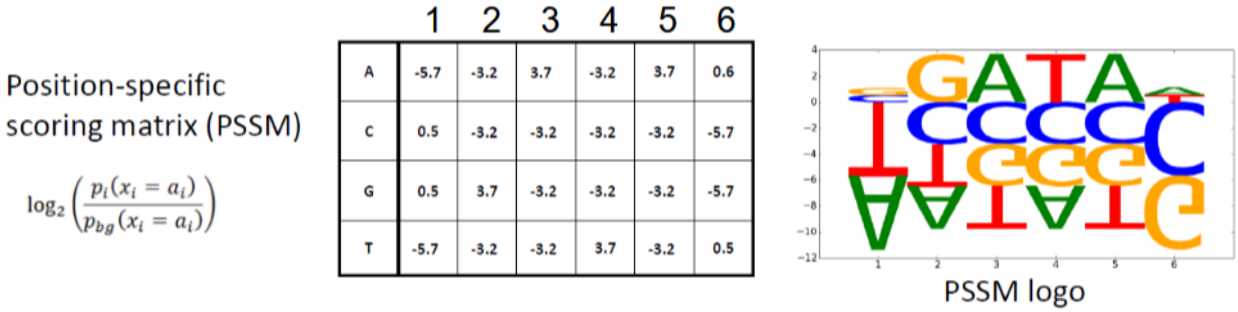
One-Hot Encoding



	G	C	A	T	T	A	C	C	G	A	T	A	A
A	0	0	1	0	0	1	0	0	0	1	0	1	1
C	0	1	0	0	0	0	1	1	0	0	0	0	0
G	1	0	0	0	0	0	0	0	1	0	0	0	0
T	0	0	0	1	1	0	0	0	0	0	1	0	0

Convolutional Filters

- Matrix of real values where rows correspond to nucleotides and columns correspond to motif widths
- Analogous to PWMs but values can be outside of a range 0 and 1 and will be combined additively instead of multiplicatively
- More similar to position-specific scoring matrix (PSSM) in values but unlike in PSSM values are not explicitly tied to a background distribution

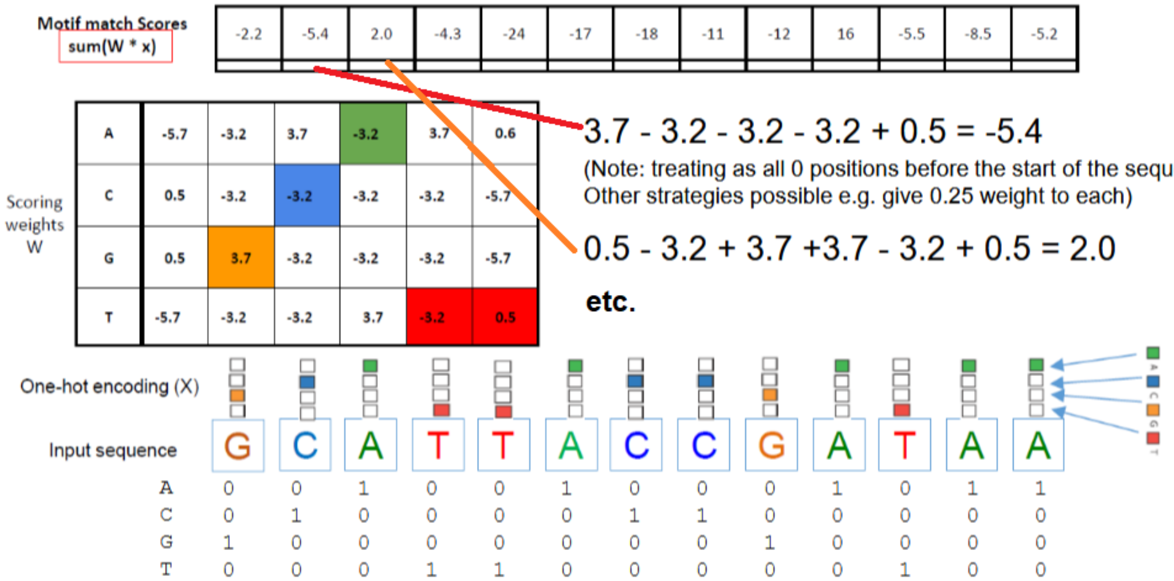

Position-specific scoring matrix (PSSM)

$$\log_2\left(\frac{p_i(x_i = a_i)}{p_{bg}(x_i = a_i)}\right)$$

	1	2	3	4	5	6
A	-5.7	-3.2	3.7	-3.2	3.7	0.6
C	0.5	-3.2	-3.2	-3.2	-3.2	-5.7
G	0.5	3.7	-3.2	-3.2	-3.2	-5.7
T	-5.7	-3.2	-3.2	3.7	-3.2	0.5

PSSM logo

To score a sequence:



Motif match Scores sum(W * x)

-2.2	-5.4	2.0	-4.3	-24	-17	-18	-11	-12	16	-5.5	-8.5	-5.2

Scoring weights W

A	-5.7	-3.2	3.7	-3.2	3.7	0.6
C	0.5	-3.2	-3.2	-3.2	-3.2	-5.7
G	0.5	3.7	-3.2	-3.2	-3.2	-5.7
T	-5.7	-3.2	-3.2	3.7	-3.2	0.5

$3.7 - 3.2 - 3.2 - 3.2 + 0.5 = -5.4$

(Note: treating as all 0 positions before the start of the sequence. Other strategies possible e.g. give 0.25 weight to each)

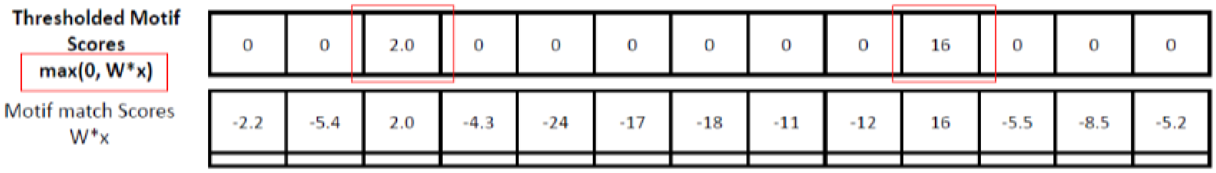
$0.5 - 3.2 + 3.7 + 3.7 - 3.2 + 0.5 = 2.0$

etc.

	G	C	A	T	T	A	C	C	G	A	T	A	A
A	0	0	1	0	0	1	0	0	0	1	0	1	1
C	0	1	0	0	0	0	1	1	0	0	0	0	0
G	1	0	0	0	0	0	0	0	1	0	0	0	0
T	0	0	0	1	1	0	0	0	0	0	1	0	0

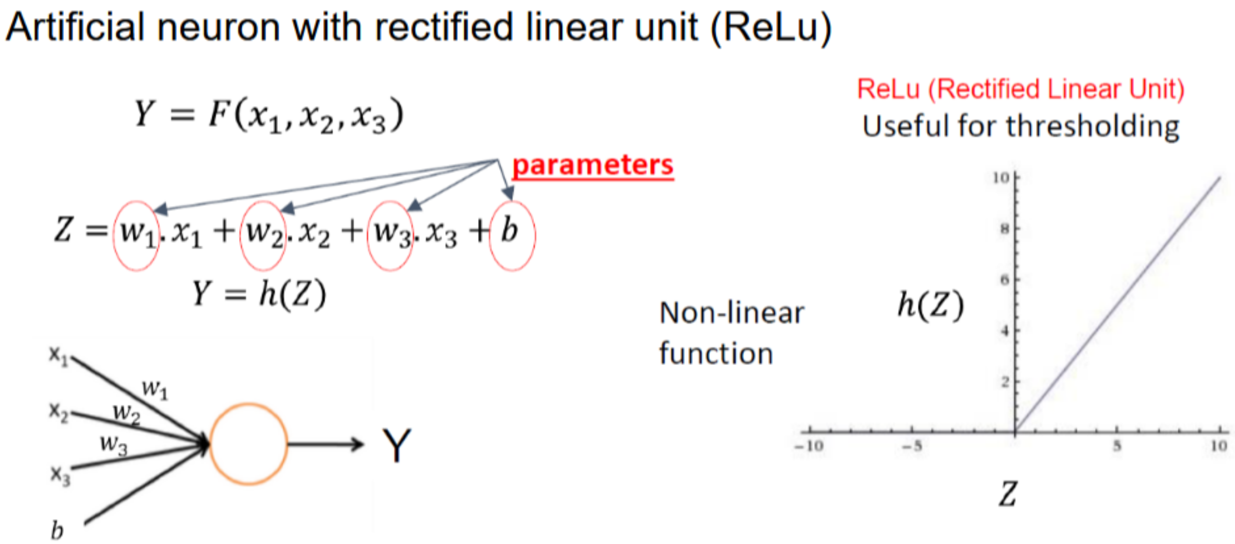
Thresholding

We use the ReLU() function, which keeps positive numbers the same, and sets all negative numbers to 0.



Thresholded Motif Scores max(0, W*x)	0	0	2.0	0	0	0	0	0	0	16	0	0	0
Motif match Scores W*x	-2.2	-5.4	2.0	-4.3	-24	-17	-18	-11	-12	16	-5.5	-8.5	-5.2

Representing a Motif with an Artificial Neuron



Artificial neuron with rectified linear unit (ReLu)

$$Y = F(x_1, x_2, x_3)$$

parameters

$$Z = w_1 . x_1 + w_2 . x_2 + w_3 . x_3 + b$$

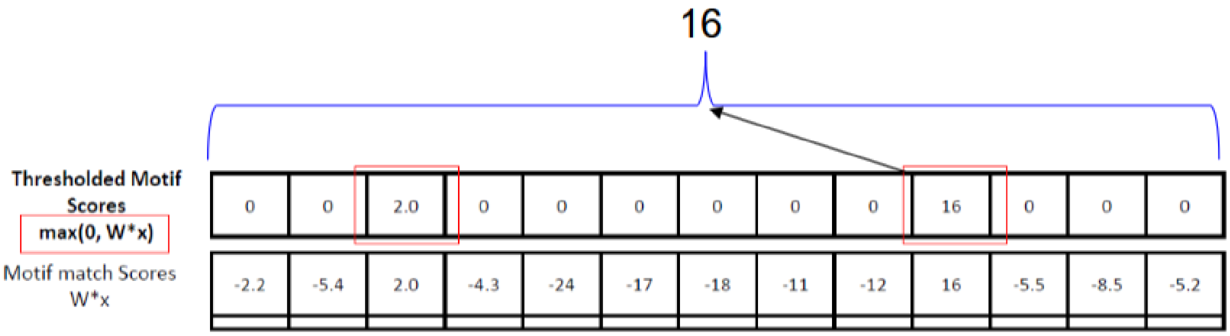
$$Y = h(Z)$$

ReLu (Rectified Linear Unit)

Useful for thresholding

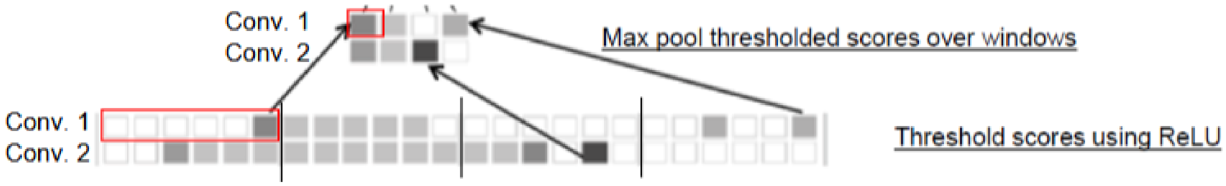
Non-linear function

Max Pooling



Thresholded Motif Scores max(0, W*x)	0	0	2.0	0	0	0	0	0	0	16	0	0	0
Motif match Scores W*x	-2.2	-5.4	2.0	-4.3	-24	-17	-18	-11	-12	16	-5.5	-8.5	-5.2

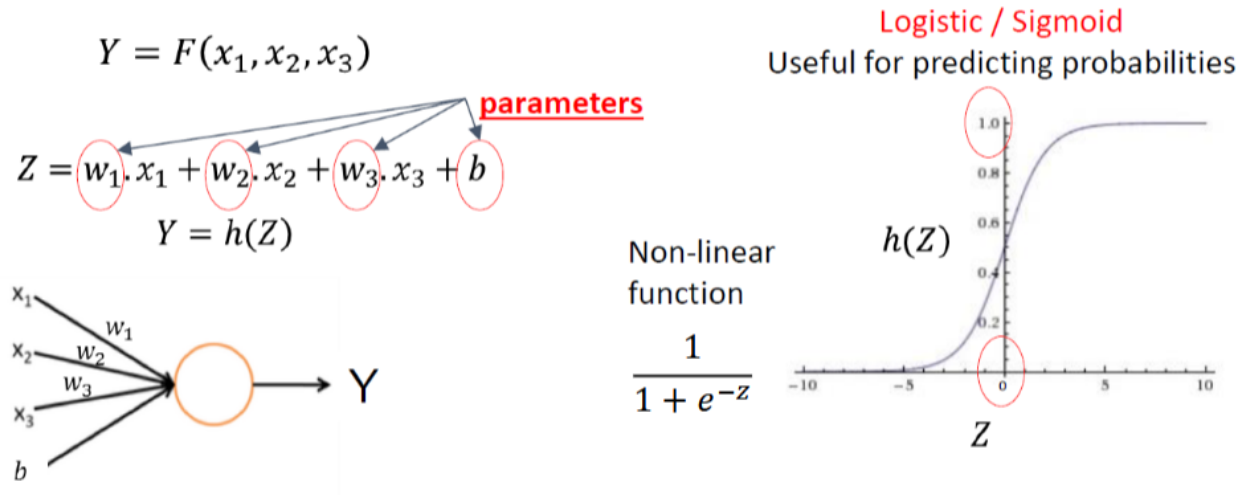
- Pooling can be done over only part of a sequence leading to multiple pooling outputs per sequence and convolution filter.



Conv. 1
Conv. 2
Max pool thresholded scores over windows
Conv. 1
Conv. 2
Threshold scores using ReLU

- Average pooling is an alternative to max pooling, or both could be done.

Predict Probabilities with Logistic Neuron



$$Y = F(x_1, x_2, x_3)$$

$$Z = w_1 . x_1 + w_2 . x_2 + w_3 . x_3 + b$$

$$Y = h(Z)$$

Logistic / Sigmoid

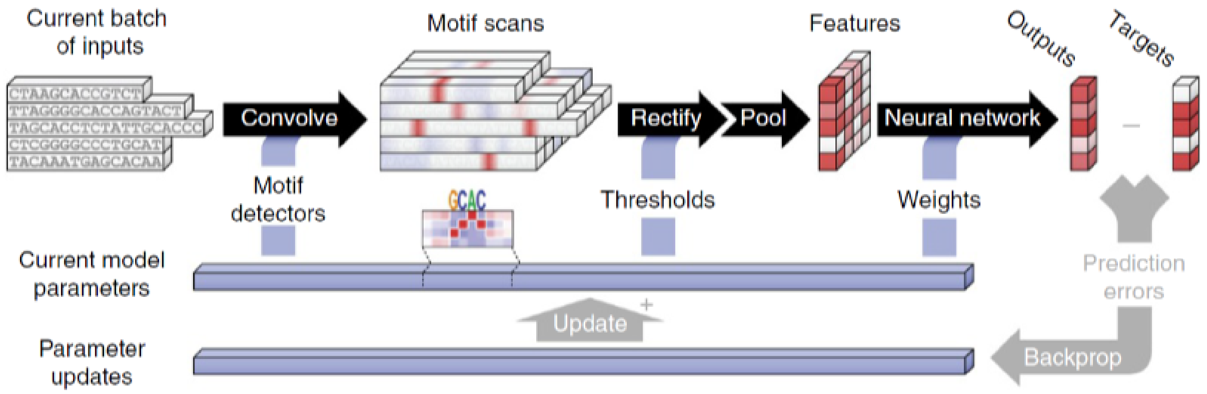
Useful for predicting probabilities

Non-linear function

$$\frac{1}{1+e^{-z}}$$

- Returns True or False (Positive or Negative), depending if the value is in the section where the logistics curve is 0 or 1.

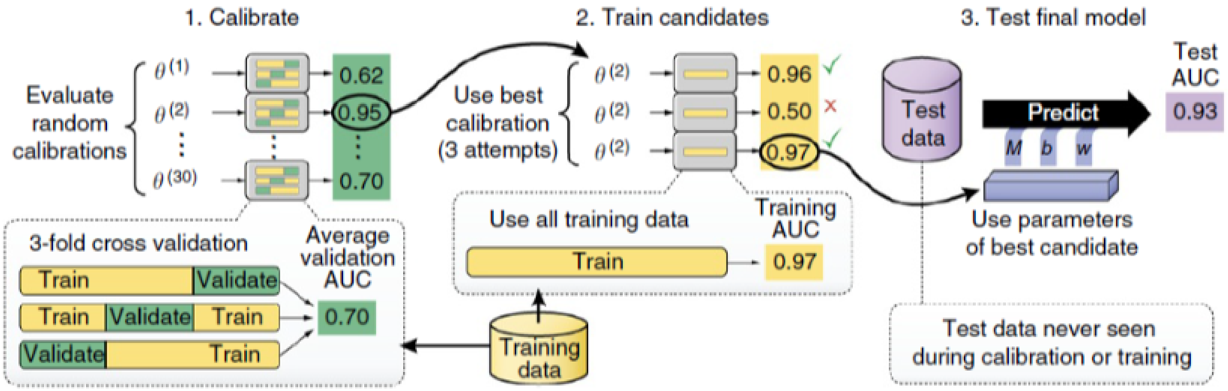
Training Model Parameters



Alipanahi et al, *Nature Biotech* 2015

- Gradients can be computed for model parameters with Back-propagation algorithm through gradient descent

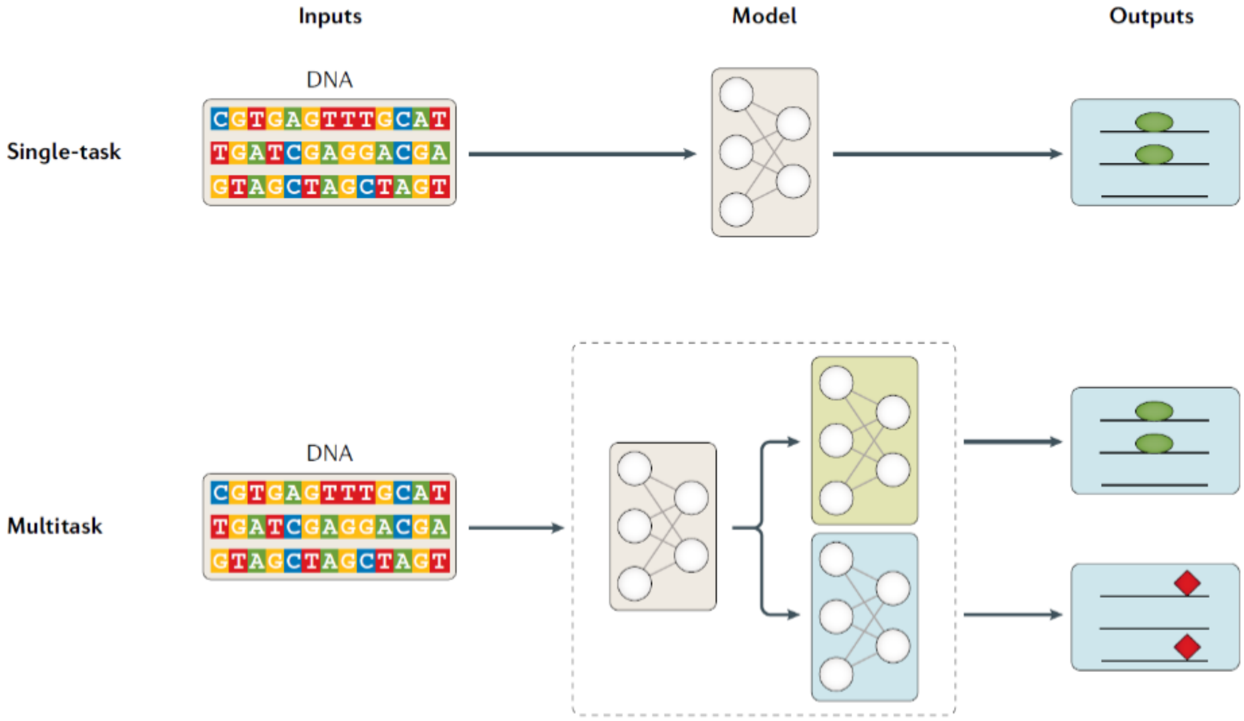
Hyperparameter Tuning



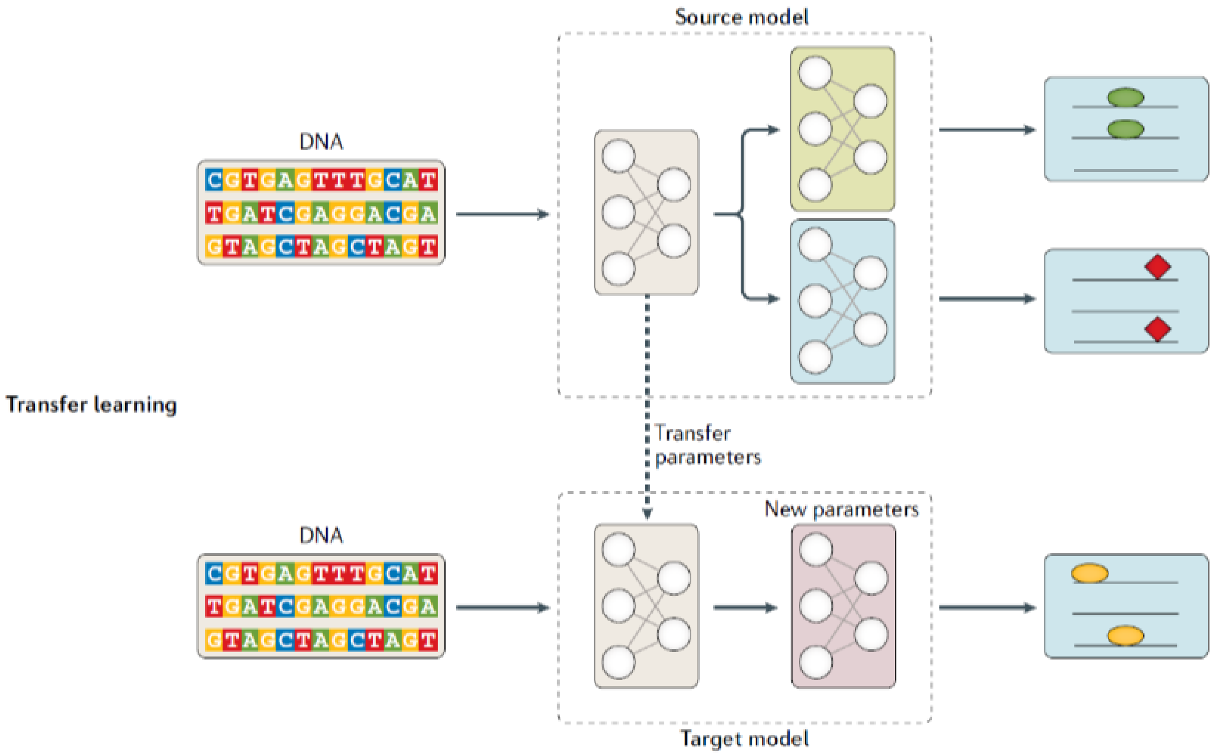
Alipanahi et al, *Nature Biotech* 2015

- Various hyper-parameters need to be set (number of motifs, motif length, number of hidden layers, learning rate, etc.)
- Selected based on empirical performance of model for different combinations

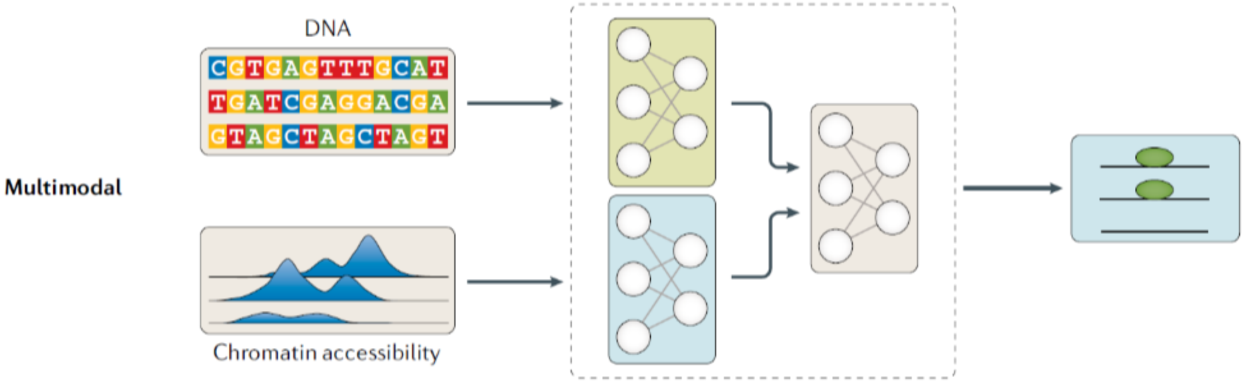
Single-Task vs. Multitask Training



Transfer Learning



Multimodal Training



Note: usually not best strategy for learning sequence variant effect prediction models since can short-circuit sequence information

Transformers for Sequence-Based Predictions

- Transformers, the architecture behind large language models (LLMs) such as ChatGPT, also have been used in some more recent DNA sequence-based prediction models

Enformer model

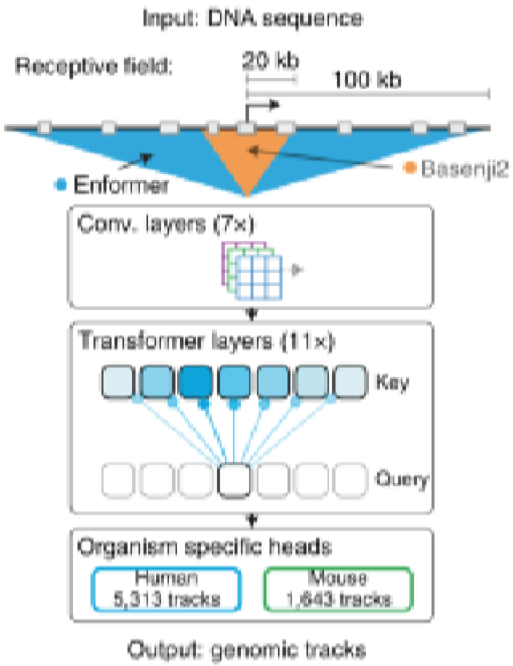


Image from Avsec et al, *Nature Methods* 2021