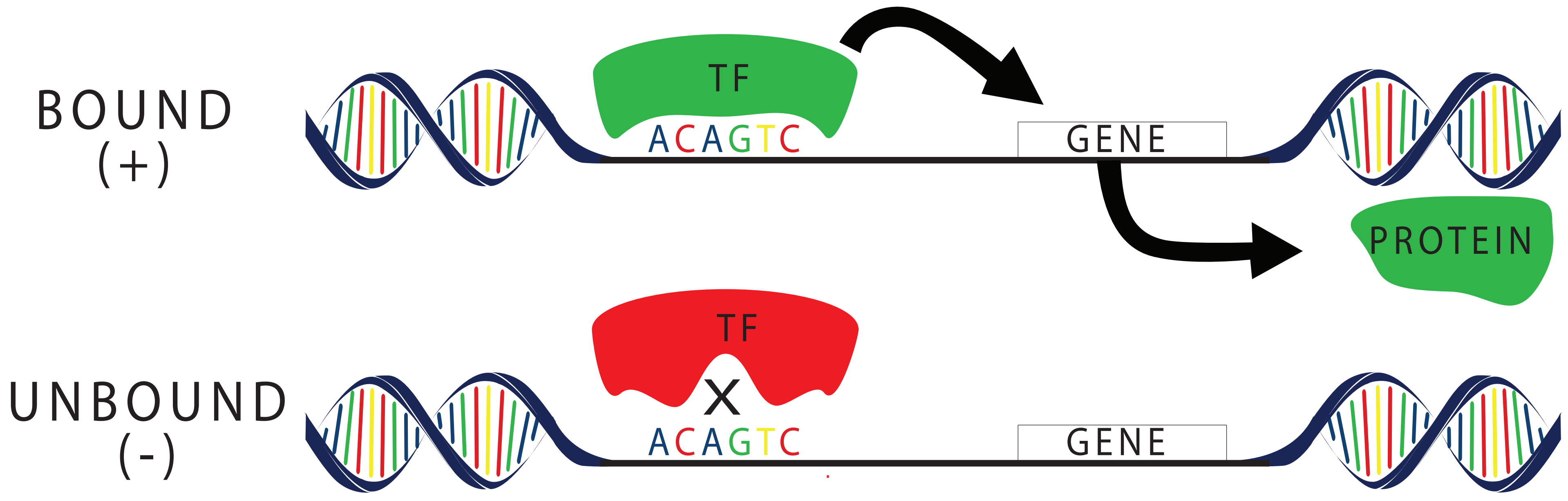
Convolutional Kitchen Sinks for Transcription Factor Binding Site Prediction

Alyssa Morrow*, **Vaishaal Shankar***
Anthony Joseph, Benjamin Recht, Nir Yosef

Transcription Factor?

A protein that binds to DNA to regulate gene expression



Transcription Factors (TF) attach to specific "binding sites" on DNA

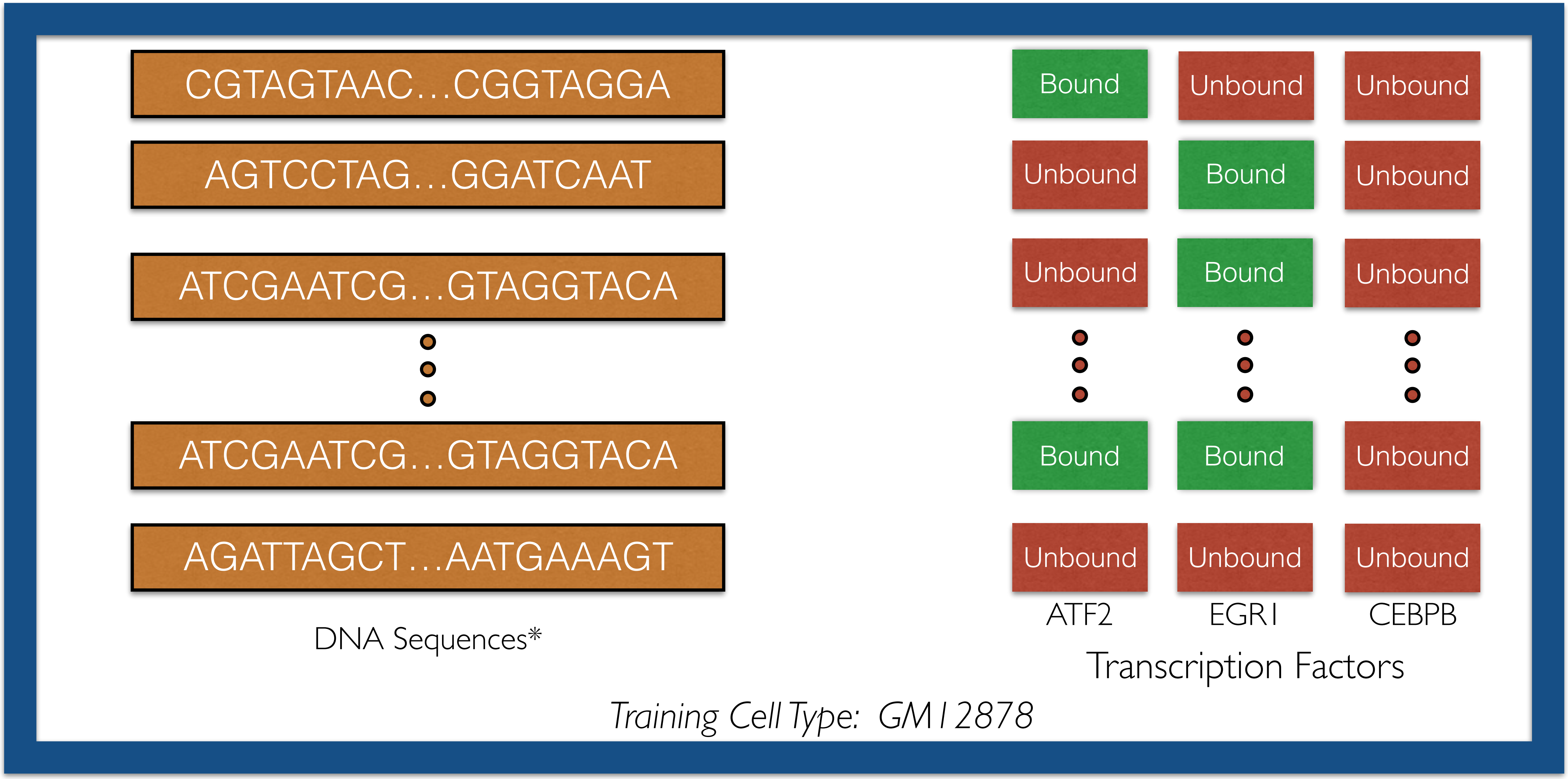
Why are binding sites important?

- Locating TF binding sites can greatly help understanding of gene regulation
- Presence/Absence of specific TF binding sites can help differentiate between benign and malignant mutations
- Genetic variation in TF binding sites linked to common diseases

Can we predict TF binding sites with data?

Sequence Classification

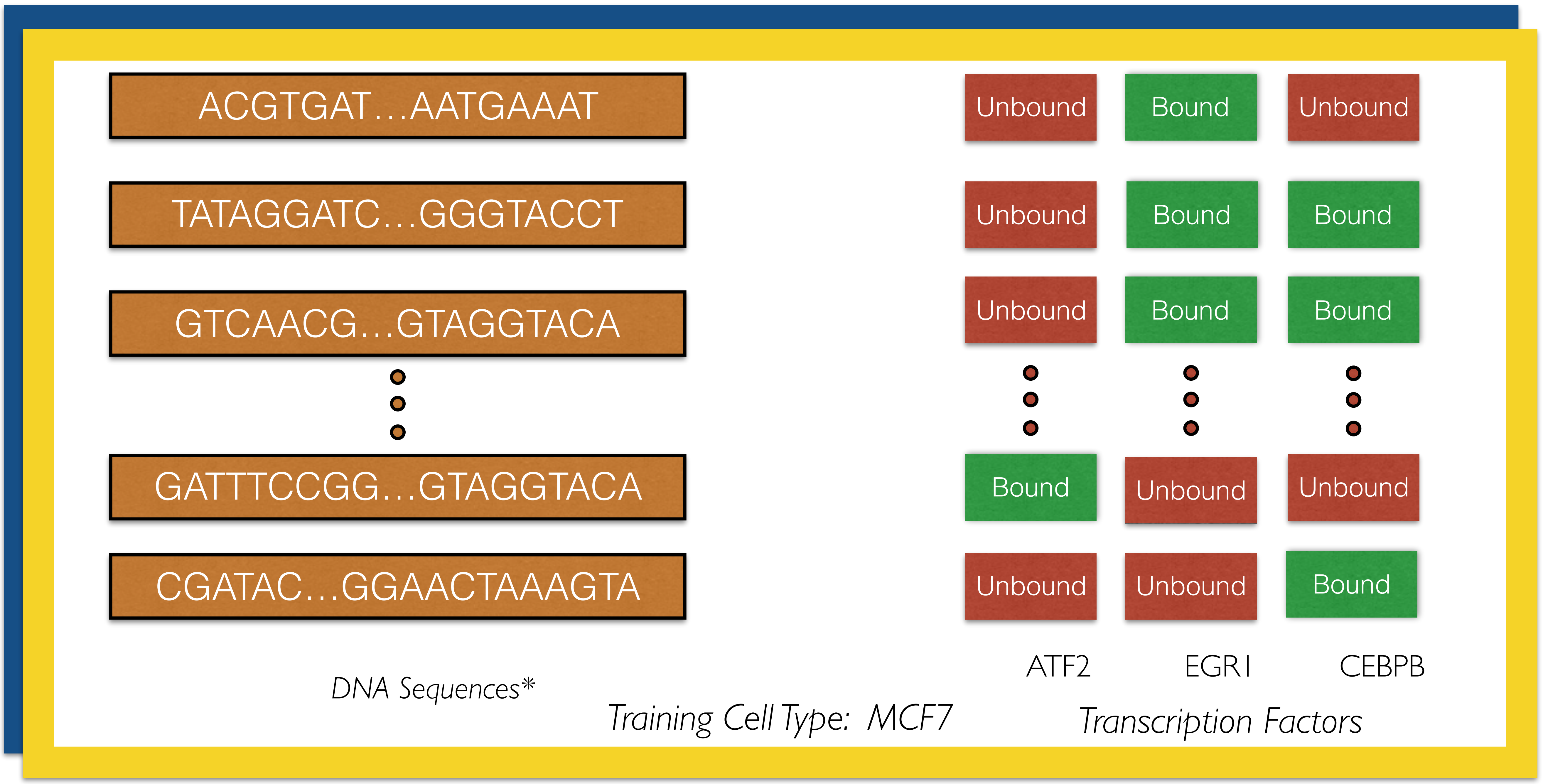
Lets set up the machine learning problem



*extracted from ENCODE ChIP-seq dataset

Sequence Classification

Lets set up the machine learning problem

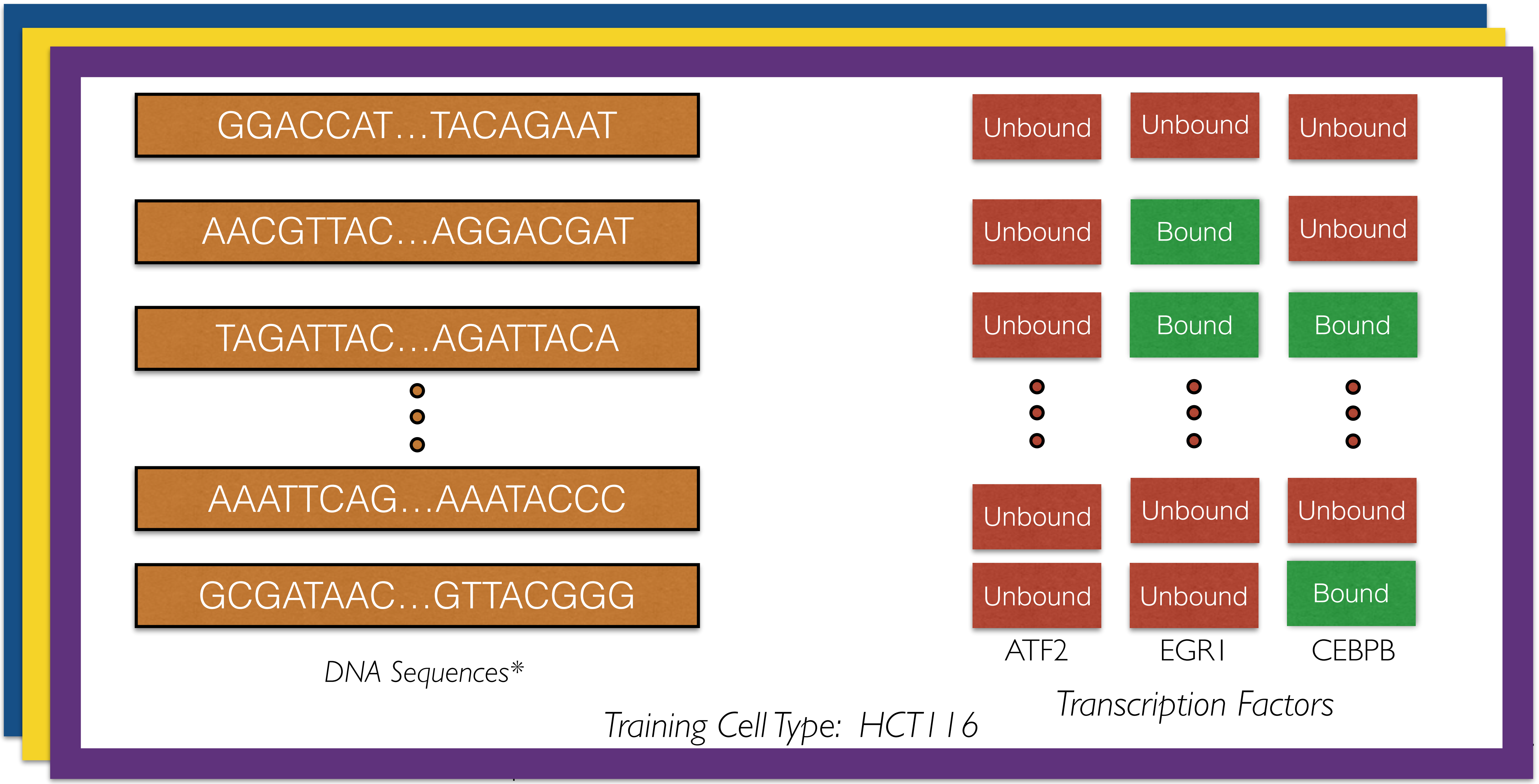


DNA Sequences*	ATF2	EGR1	CEBPB
ACGTGAT…AATGAAAT	Unbound	Bound	Unbound
TATAGGATC…GGGTACCT	Unbound	Bound	Bound
GTCAACG…GTAGGTACA	Unbound	Bound	Bound
⋮	⋮	⋮	⋮
GATTTCCGG…GTAGGTACA	Bound	Unbound	Unbound
CGATAC…GGAACTAAAGTA	Unbound	Unbound	Bound

*extracted from ENCODE ChIP-seq dataset

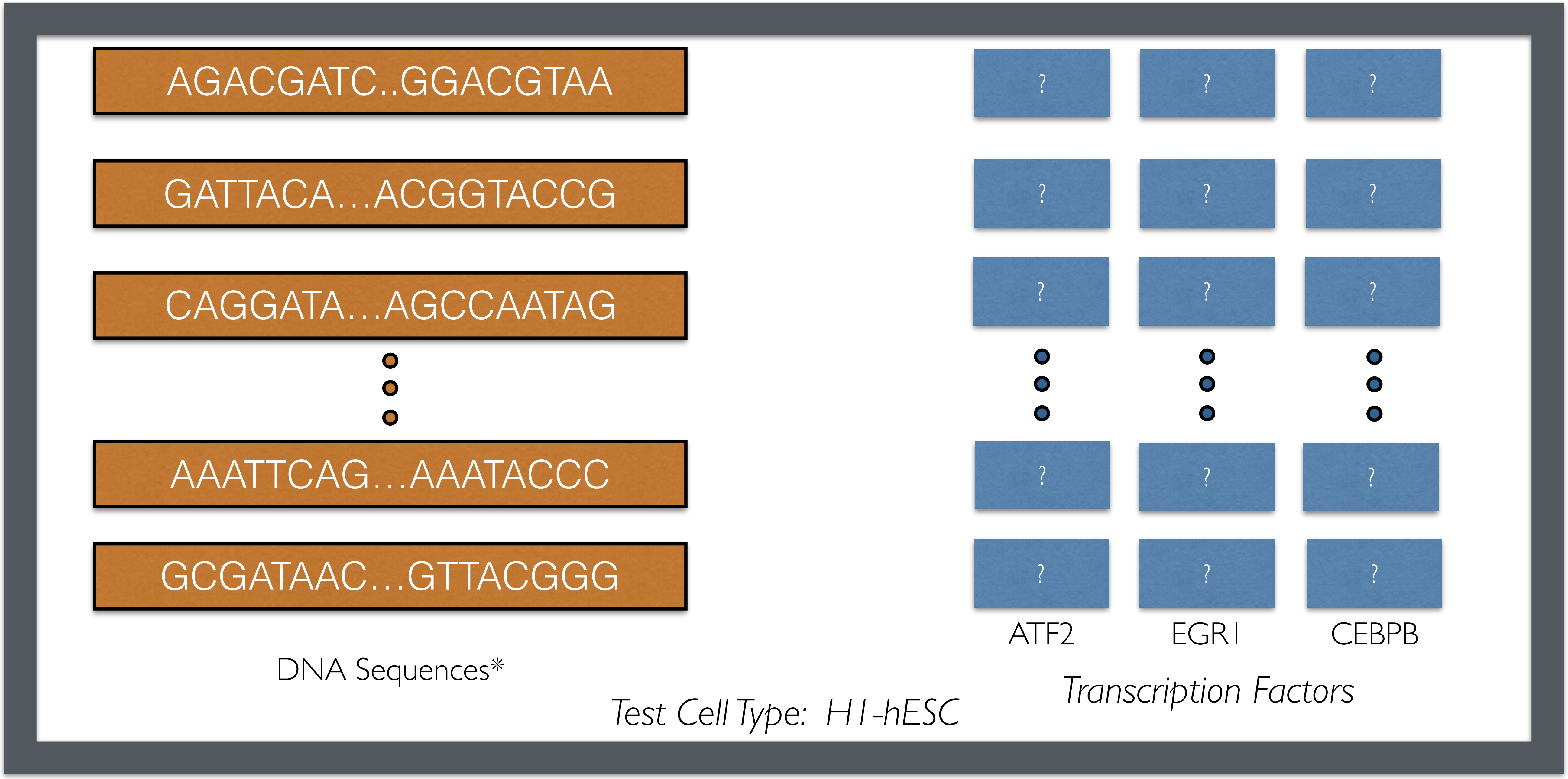
Sequence Classification

Lets set up the machine learning problem



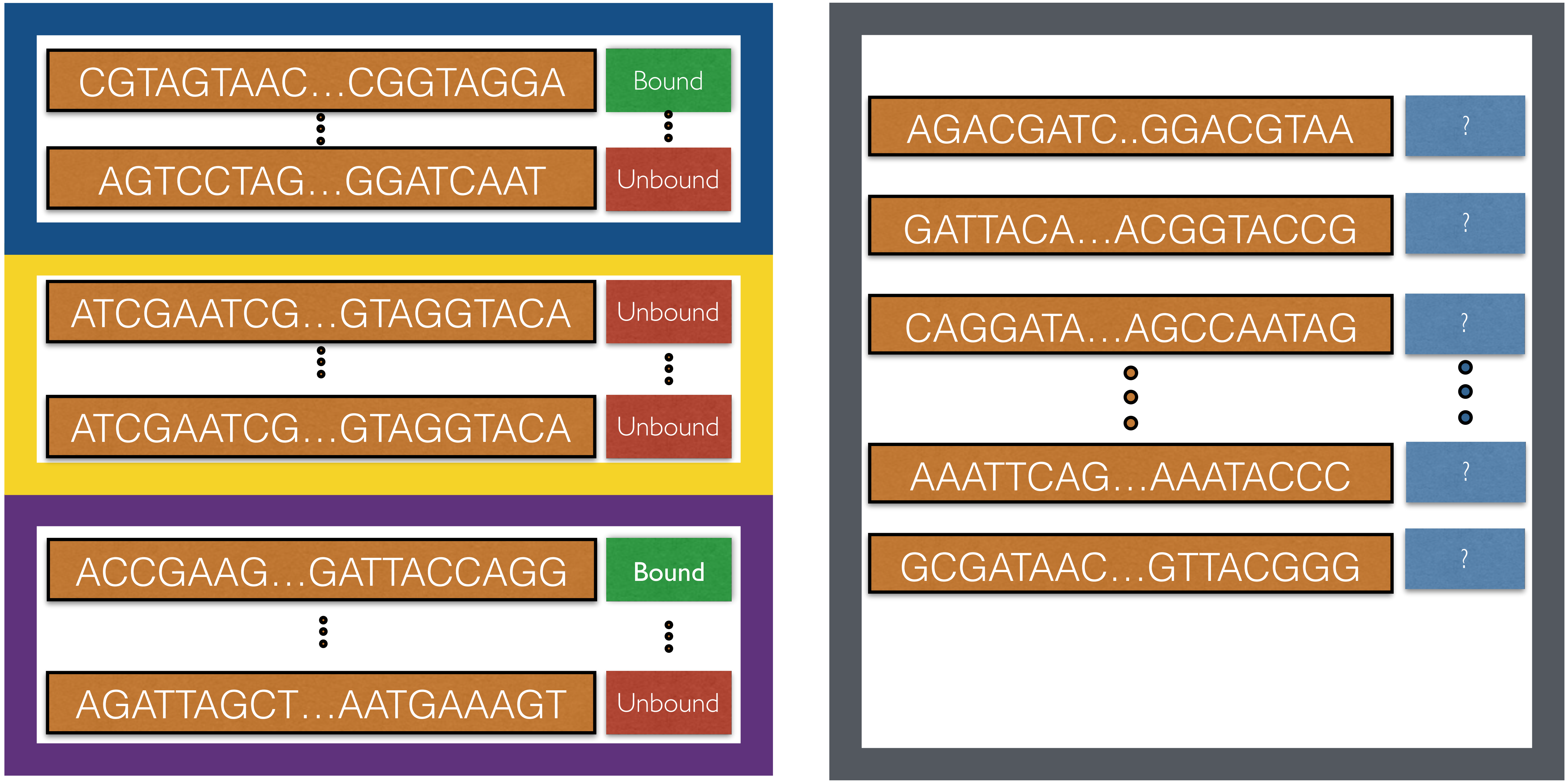
Sequence Classification

Lets set up the machine learning problem



*extracted from ENCODE ChIP-seq dataset

Train/Test on single TF



ATF2 Training Data
(Multiple cell types)

ATF2 Test Data
(One Foreign Cell Type)

We could use a Convolutional Neural Network

- Deep Convolutional Neural Networks (CNNs) are state of the art for many classification tasks
- Recently successful for TF binding site prediction
  - DeepBind
  - Basett
  - DeepSea

Challenges with Deep Nets

- Scales to large dataset sizes
  - Slow to train
  - Difficult to parallelize
- Design decisions
  - How many layers?
  - How wide?
  - Learning rate?

Can we try something faster & simpler while preserving predictive performance?

Yes.

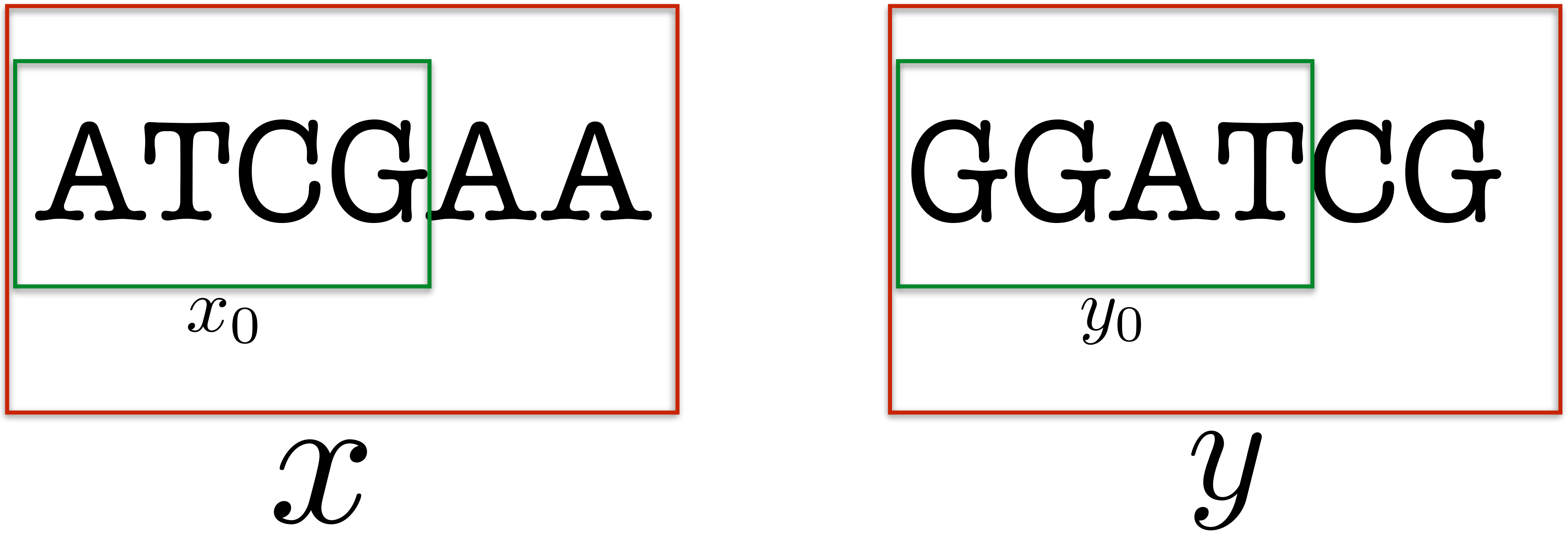
Kernels

- Well understood theoretically
- Previously successful for biological sequence classification
  - Spectrum Kernel
  - Gapped k-mer Kernel
- Single point of design: kernel function

How to Design Kernel?

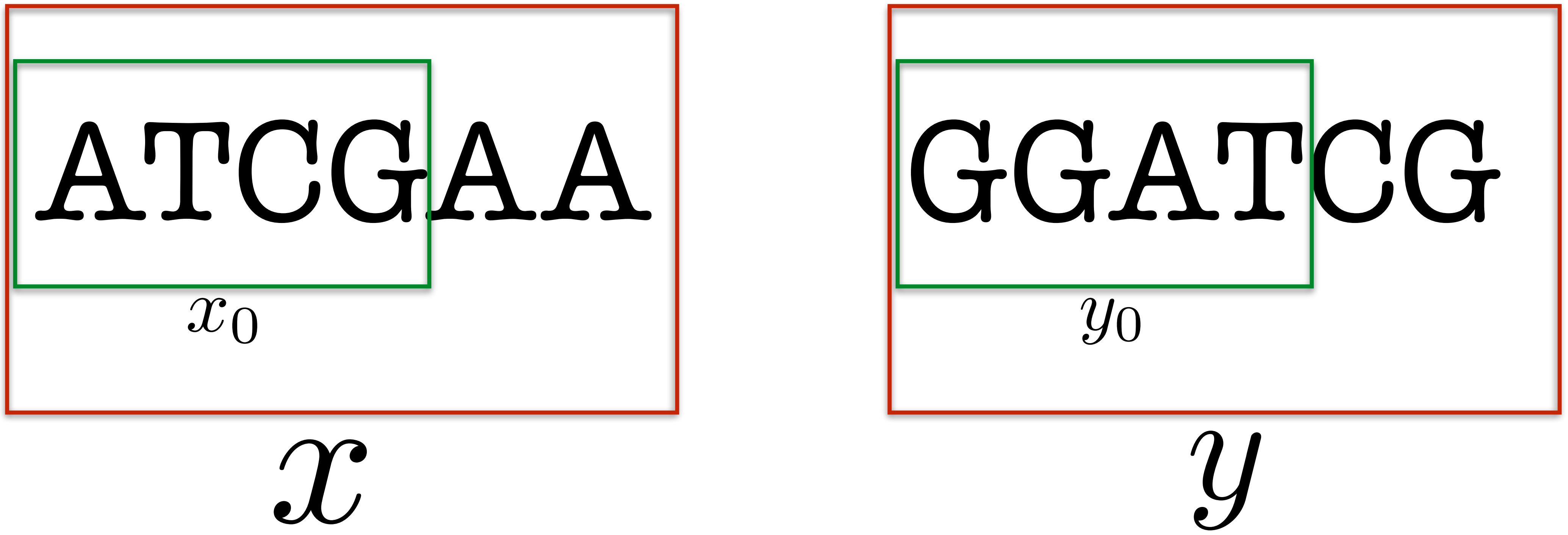
- k(x,y) large for "similar" sequences
    - sequences x,y that both bind/don't bind to a particular transcription factor
- k(x,y) small for "dissimilar" sequences
- Binding sites are 6-30 base pairs long
    - Local sequence comparisons
    - Aggregation of local comparisons

A Convolutional String Kernel



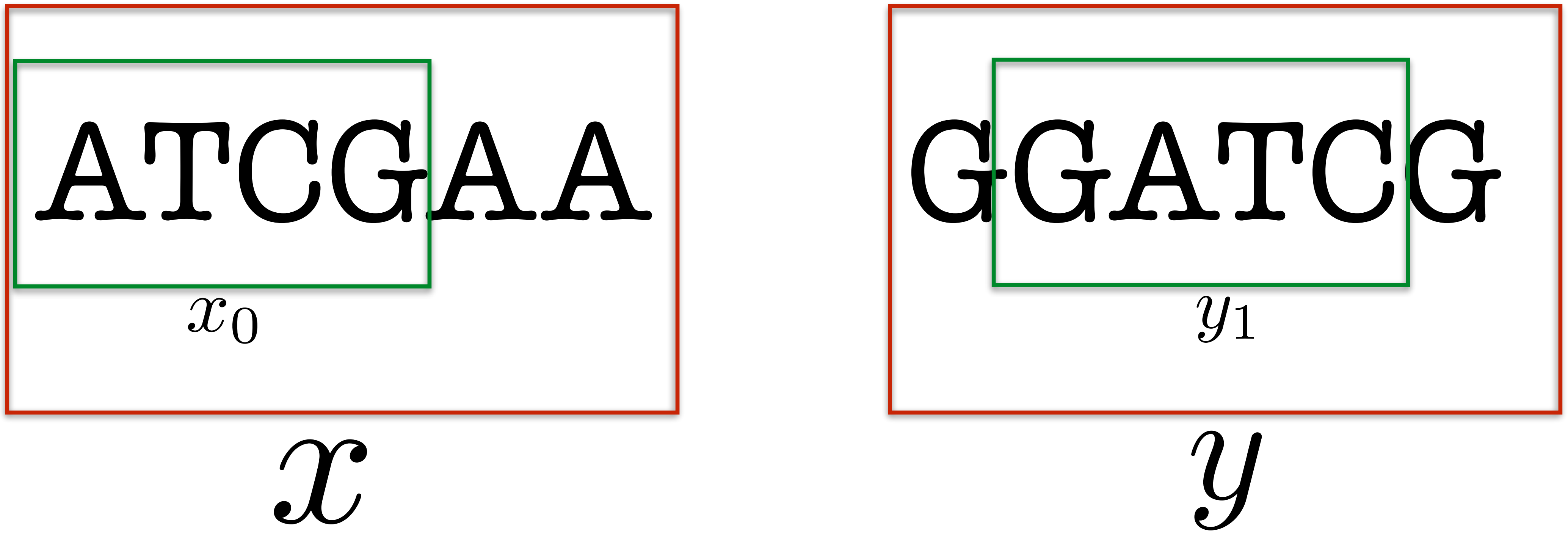
$$k(x, y) =$$

A Convolutional String Kernel



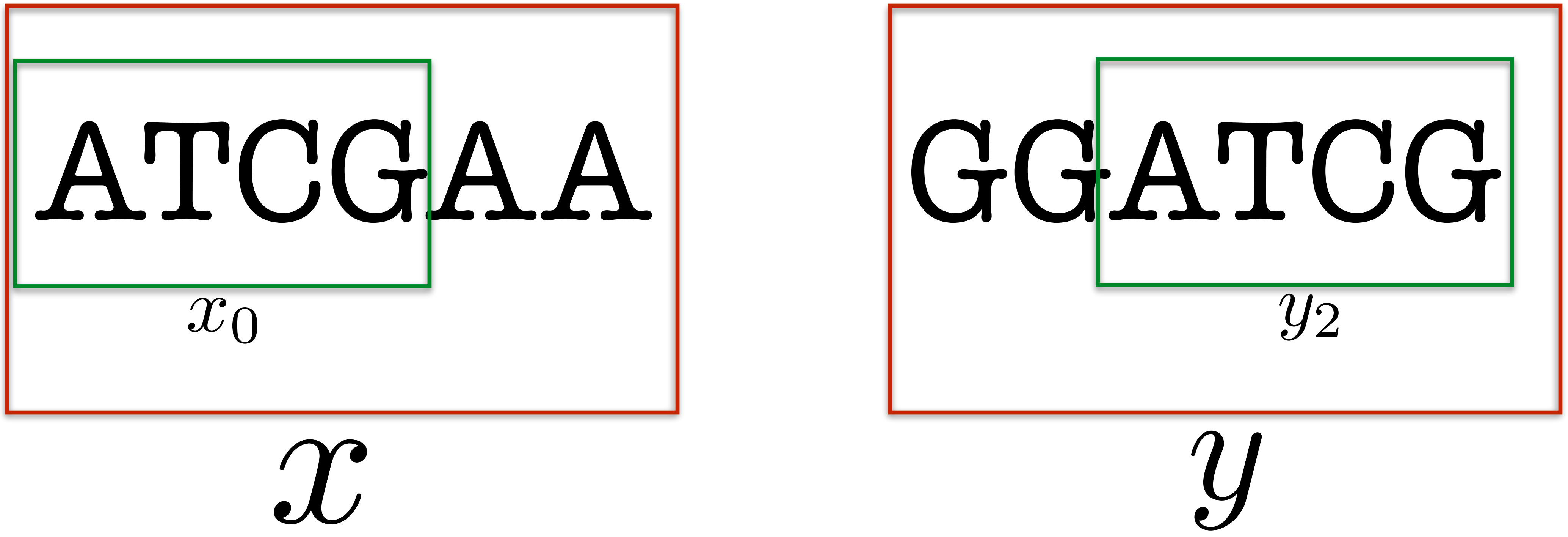
$$k(x,y) = exp(-\gamma \mathbb{H}^2(x_0, y_0))$$

A Convolutional String Kernel



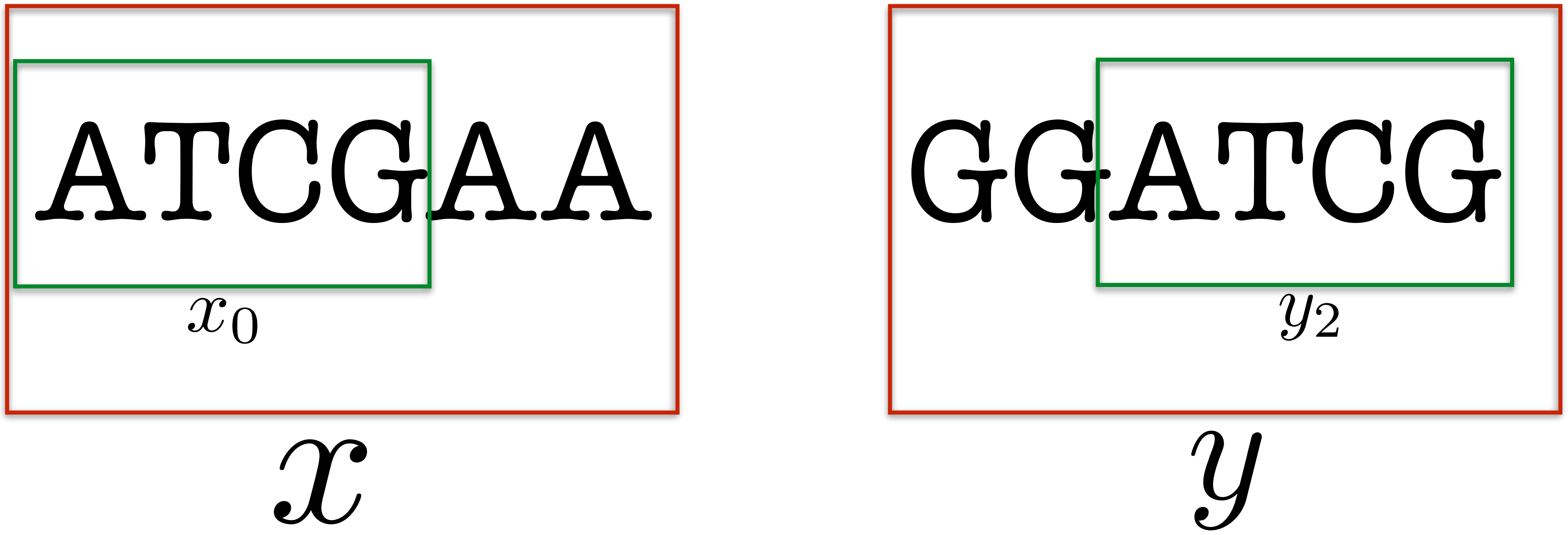
$$k(x,y) = exp(-\gamma \mathbb{H}^2(x_0,y_0)) + exp(-\gamma \mathbb{H}^2(x_0,y_1))$$

A Convolutional String Kernel



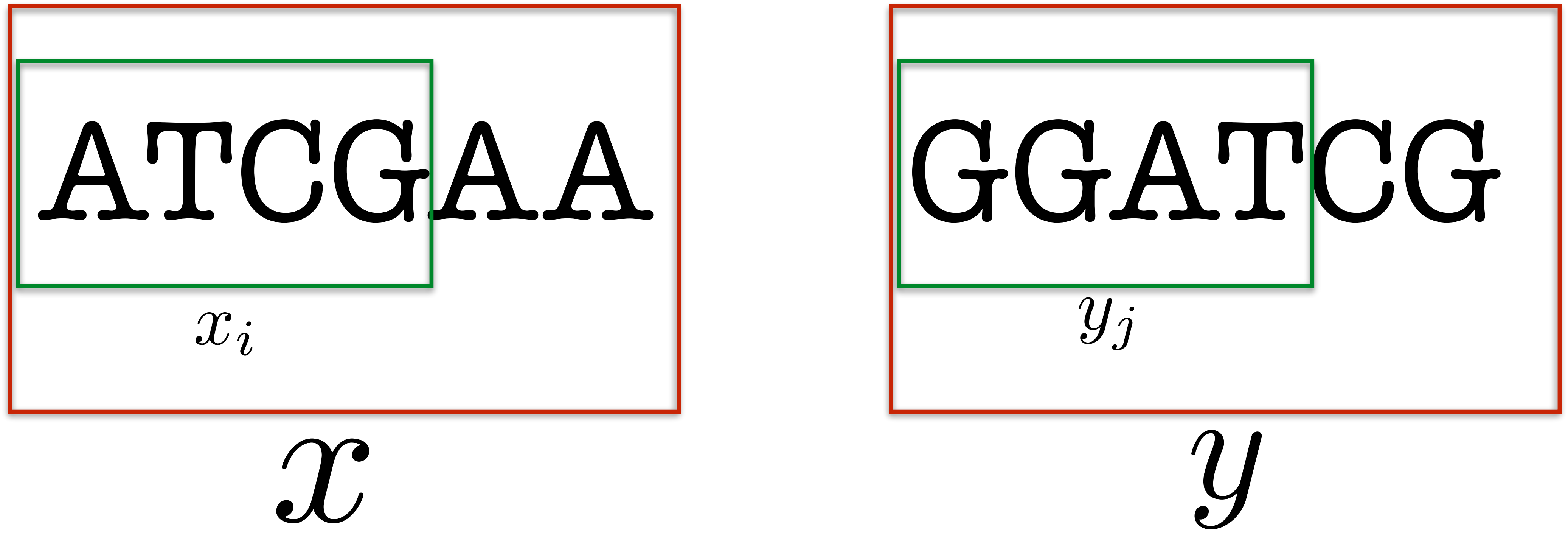
$$k(x,y) = exp(-\gamma \mathbb{H}^2(x_0, y_0)) + exp(-\gamma \mathbb{H}^2(x_0, y_1)) + exp(-\gamma \mathbb{H}^2(x_0, y_2))$$

A Convolutional String Kernel



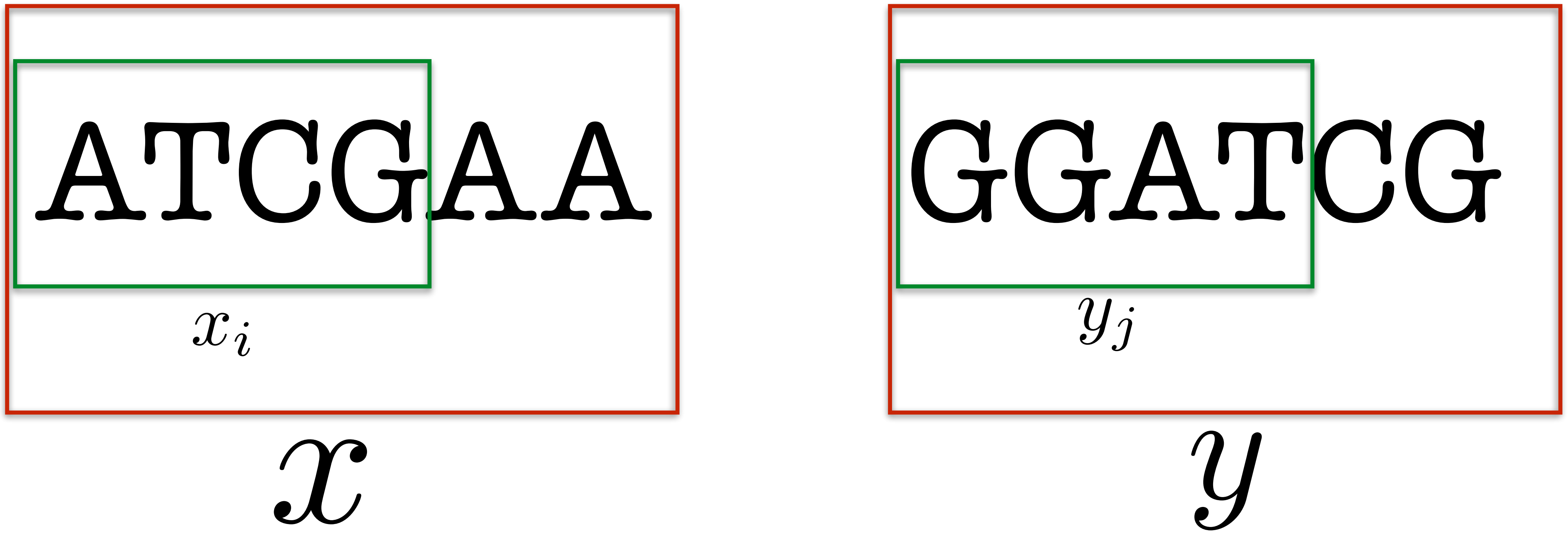
$$\begin{aligned} k(x,y) = & \, exp(-\gamma \mathbb{H}^2(x_0, y_0)) \\ & + exp(-\gamma \mathbb{H}^2(x_0, y_1)) \\ & + exp(-\gamma \mathbb{H}^2(x_0, y_2)) \\ & \vdots \end{aligned}$$

A Convolutional String Kernel



$$k(x,y) = \sum_{i=0}^{d-n} \sum_{j=0}^{d-n} \exp(-\gamma \mathbb{H}^2(x_i, y_j))$$

A Convolutional String Kernel



$$k(x,y) = \sum_{i=0}^{d-n} \sum_{j=0}^{d-n} \exp(-\gamma \mathbb{H}^2(x_i, y_j))$$

What's wrong?

Still have to solve learning problem!

Let K be the kernel matrix derived by kernel function and training data

Let y be the training labels

Solve for w

$$Kw = y$$

Matrix Inverse!

Unfortunately K grows quadratically with dataset size

Kernel Trick (in reverse)

$$K = \Phi\Phi^T$$

Due to representer theorem it suffices to compute:

$$\min_z ||\Phi z - y||_2^2$$

Least Squares!

Unfortunately Φ is potentially infinite dimensional

The convolutional kernel can be efficiently approximated!

Definitions

100,000	N	Number of training data points
101	d	Length of each string in training set
8	n	Local comparison window length
0.1	γ	Kernel bandwidth
4096	M	Approximation dimension

More Definitions

Let W be a $M \times n$ matrix Such that $W_{ij} \sim N(0, \gamma)$
Let $b \in \mathbb{R}^M$ Such that $\forall_i b_i \sim U(0, 2\pi)$
Let $\hat{\phi}$ be a map from $\mathbb{R}^n \to \mathbb{R}^M$ Such that $\hat{\phi}(x) = \cos(Wx + b)$

Random Approximation

$$K = \Phi\Phi^T$$

$$k(x,y) = \sum_{i=0}^{d-n}\sum_{j=0}^{d-n} \exp(-\gamma \mathbb{H}^2(x_i, y_j))$$

$$k(x,y) \approx \sum_{i=0}^{d-n}\sum_{j=0}^{d-n} \hat{\phi}(x_i)^T \hat{\phi}(y_j)$$

$$k(x,y) \approx \Big(\sum_{i=0}^{d-n} \hat{\phi}(x_i)\Big)^T \Big(\sum_{j=0}^{d-n} \hat{\phi}(y_j)\Big)$$

$$K \approx \hat{\Phi}\hat{\Phi}^T$$

Kernel Trick (in reverse)

$$K \approx \hat{\Phi}\hat{\Phi}^T$$

Due to representer theorem it suffices to compute:

$$\min_z ||\hat{\Phi}z - y||_2^2$$

$\hat{\Phi}$ is N x M dimensional

If M is small this problem is easy!

Putting it Together

1. Compute random map for each training point
2. Solve least squares system with output features
3. Compute *same* random map for each test point
4. Use least squares model to predict labels for test points

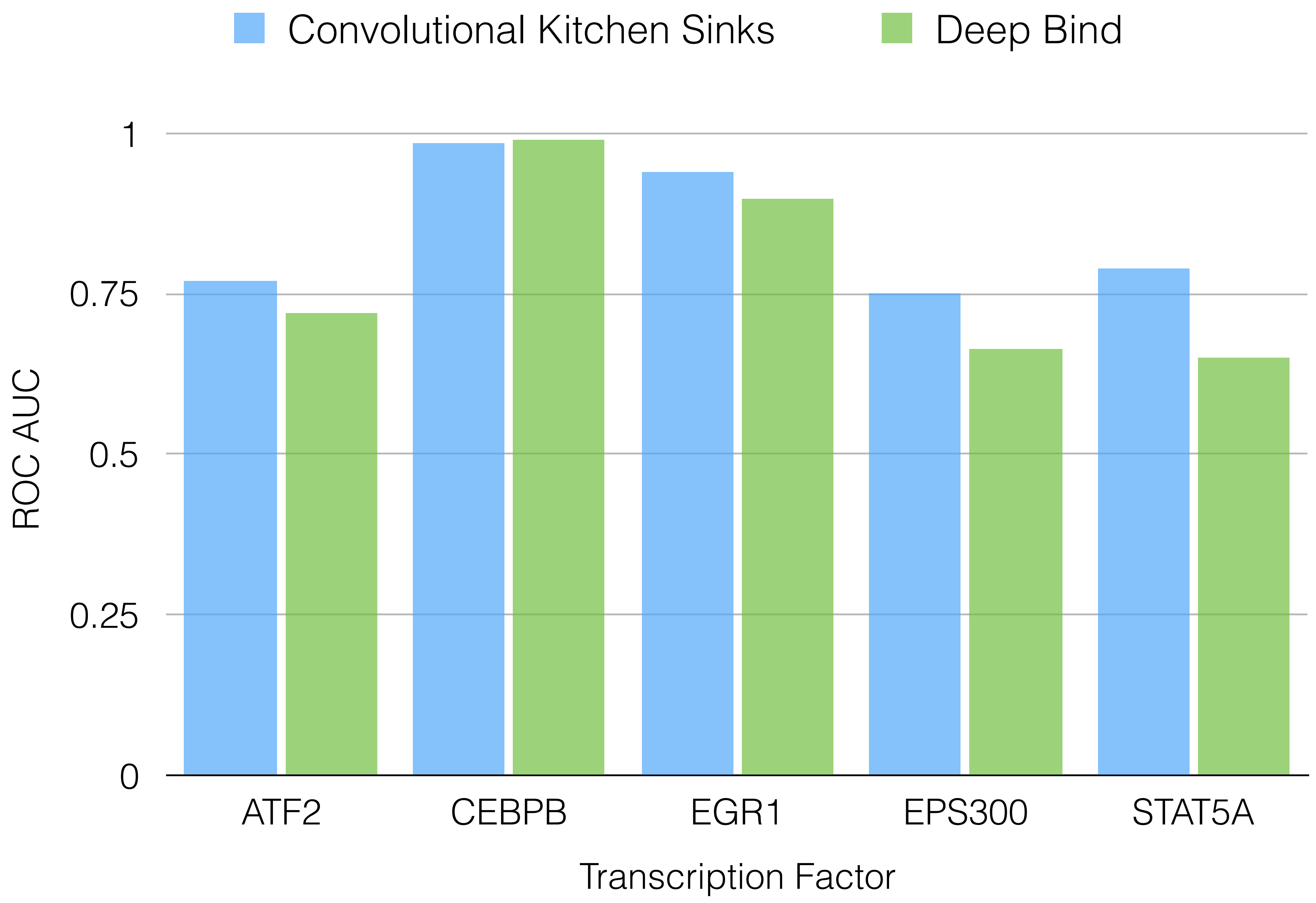
Comparison: DeepBind

- DeepBind is state of art deep learning method for predicting binding sites from DNA sequence
- Trained and tested on DeepBind's datasets:
  - Top 101 bp positive sequences from ENCODE
  - Synthetically generated negative sequences

Training Time

- Deep bind model takes 2 GPU hours to train for single TF
    - Over 8 hours with cross validation to find optimal hyper parameters
- Our convolutional kitchen sink model takes 10 minutes on a GPU to train for a single TF
    - Less than 20 minutes with cross validation to find optimal hyper parameters

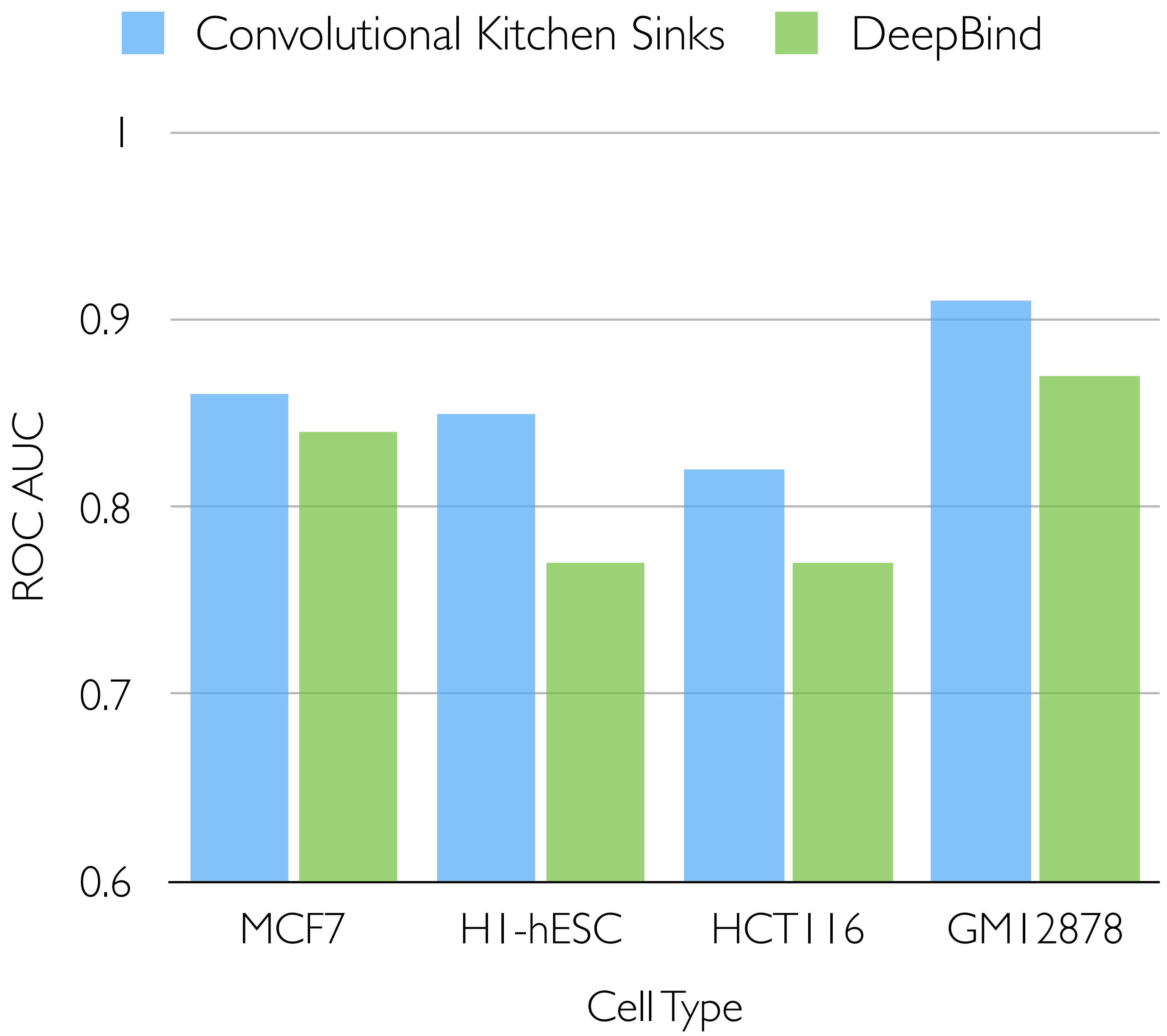
Accuracy Results



More Test Data

- Tested TF EGR1 on new sequences preprocessed from ENCODE:
    - Tested on 4 different (foreign) cell types
    - 101 bp positive sequences from ENCODE
    - True negative sequences from genome

More Accuracy Results



Conclusion & Future Work

- Fast simple learning algorithm is competitive for transcription factor binding site prediction
- Using epigenetic data such as Dnase hypersensitivity and Histone modification will further help binding site prediction
- Binding site detection (as opposed to recognition) more difficult problem
- Adaptive kernel learning

FIN

vaishaal@berkeley.edu

akmorrow@berkeley.edu

References

- Leslie, C., Eskin, E., Weston, J., Noble, W. The Spectrum Kernel: A String Kernel for SVM Protein Classification. Proceedings of the Pacific Symposium on Biocomputing, 2002. pp. 564–575.
- Ghandi M. et al. (2014) Enhanced regulatory sequence prediction using gapped k-mer features. PLoS Comput. Biol., 10, e1003711.
- ENCODE Project Consortium et al. The ENCODE (Encyclopedia of DNA elements) project. Science, 306(5696):636–640, 2004.
- Babak Alipanahi, Andrew Delong, Matthew T Weirauch, and Brendan J Frey. Predicting the se- quence specificities of DNA-and RNA-binding proteins by deep learning. Nature biotechnology, 2015.
- Ali Rahimi and Benjamin Recht. Weighted sums of random kitchen sinks: Replacing minimiza- tion with randomization in learning. In Advances in Neural Information Processing Systems, pages 1313–1320, 2009.
- Julien Mairal, Piotr Koniusz, Zaid Harchaoui, and Cordelia Schmid. Convolutional kernel networks. In Advances in Neural Information Processing Systems, pages 2627–2635, 2014.
- Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In Advances in Neural Information Processing Systems, pages 1177–1184, 2007.