Deep Learning for Genomics Using DragoNN

Johnny Israeli
Biophysics PhD Candidate & SIGF Bio-X Fellow, Stanford University

Charles Killiam
Certified Instructor, NVIDIA Deep Learning Institute
NVIDIA Corporation

May 9th 2017

DEEP LEARNING INSTITUTE

DLI Mission

Helping people solve challenging problems using AI and deep learning.

- Developers, data scientists and engineers
- Self-driving cars, healthcare and robotics
- Training, optimizing, and deploying deep neural networks

TOPICS

- Lab Perspective
- Genomics
- DragoNN / SimDNA / Keras / DeepLIFT
- Lab
  - Discussion / Overview
  - Lab Environment
  - Lab Review
- Example Application
- State of the Field

LAB PERSPECTIVE

WHAT THIS LAB IS

- Introduction to large-scale data in genomics
- Applications of deep learning in genomics
- Guided, hands-on exercise using the DragoNN (Deep Regulatory Genomic Neural Network) toolkit for learning how to train and interpret simple deep learning models for genomics

WHAT THIS LAB IS NOT

- Introduction to machine learning from first principles
- Detailed explanation of biology and genomics
- Rigorous mathematical formalism of neural networks
- Survey of all the features and options of the DragoNN package

ASSUMPTIONS

- You are familiar with:
  - Convolutional neural networks (CNNs)
- Helpful to have:
  - Rudimentary coding background

TAKE AWAYS

- Ability to setup your own DragoNN network for genomic research
- Know where to go for more info on DragoNN
- Familiarity with DragoNN workflow
- Downstream applications

GENOMICS

Decoding genome function

TGCCAAGCAGCAAAGTTTTGCTGCTGTTTATTTTTGTAGCTCTTACTATATT
CTACTTTTACCATTGAAAATATTGAGGAAGTTATTTATATTTCTATTTTTTAT
ATATTATATATTTTATGTATTTTAATATTACTATTACACATAATTATTTTTTAT
ATATATGAAGTACCAATGACTTCCTTTTCCAGAGCAATAATGAAATTTCAC
AGTATGAAAATGGAAGAAATCAATAAAATTATACGTGACCTGTGGCGAAG
TACCTATCGTGGACAAGGTGAGTACCATGGTGTATCACAAATGCTCTTTCC
AAAGCCCTCTCCGCAGCTCTTCCCCTATGACCTCTCATCATGCCAGCATTA
CCTCCCTGGACCCCTTTCTAAGCATGTCTTTGAGATTTTCTAAGAATTCTTA
TCTTGGCAACATCTTGTAGCAAGAAAATGTAAAGTTTTCTGTTCCAGAGCC
TAACAGGACTTACATATTTGACTGCAGTAGGCATTATATTTAGCTGATGAC
ATAATAGGTTCTGTCATAGTGTAGATAGGGATAAGCCAAAATGCAATAAG
AAAAACCATCCAGAGGAAACTCTTTTTTTTTTCTTTTTCTTTTTTTTTTTTCCA
GATGGAGTCTCGCACTTCTCTGTCACCCGGGCTGGAGCGCAGTGGTGCAA
TCTTGGCTCACTGCAACCTCCACCTCCTGGGTTCAGGTGATTCTCCCACCTC
AGCCTCCCGAGTAGTAGCTGGAATTACAGGTGCGCGCTCCCACACCTGGC
TAATTTTTTGTATTCTTAGTAGAGATGGGGTTTCACCATGTTGGCCAGGCT
GGTCTCAAACTCCTGCCCTCAGGTGATCTGCCCACCTTGGCCTCCCAGTGT
TGGGTTTACAGGCGTGAGCCACCGCGCCTGGCCTGGAGGAAACTCTTTAT
AACTACCGAGTGGTGATGCTGAAGGGAGACACAGCCTTGGATATGCGAG
GACGATGCAGTGCTGGACAAAAGGCAGGTATCTCAAAAGCCTGGGGAGC
CAACTCACCCAAGTAACTGAAAGAGAGAAACAAACATCAGTGCAGTGGA
AGCACCCAAGGCTACACCTGAATGGTGGGAAGCTCTTTGCTGCTATATAA
AATGAATCAGGCTCAGCTACTATTATT ……

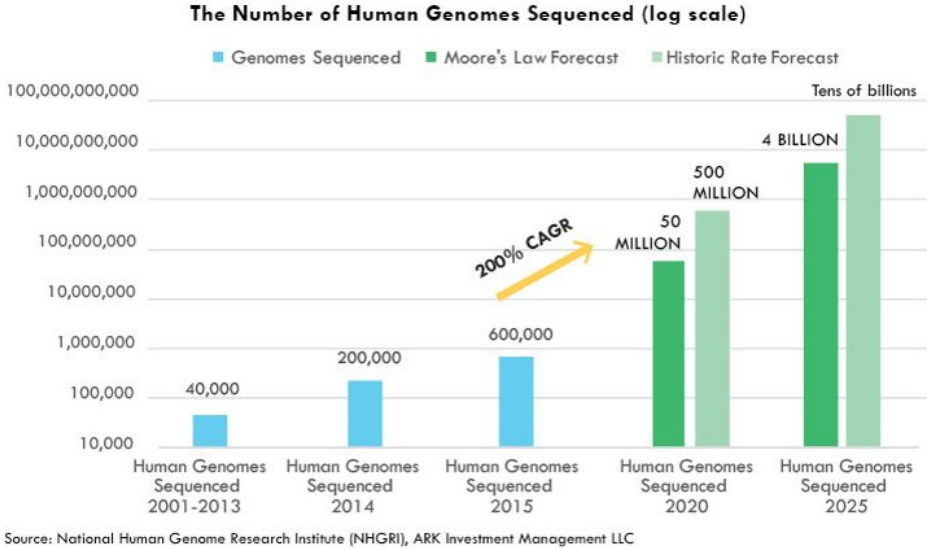
Function?

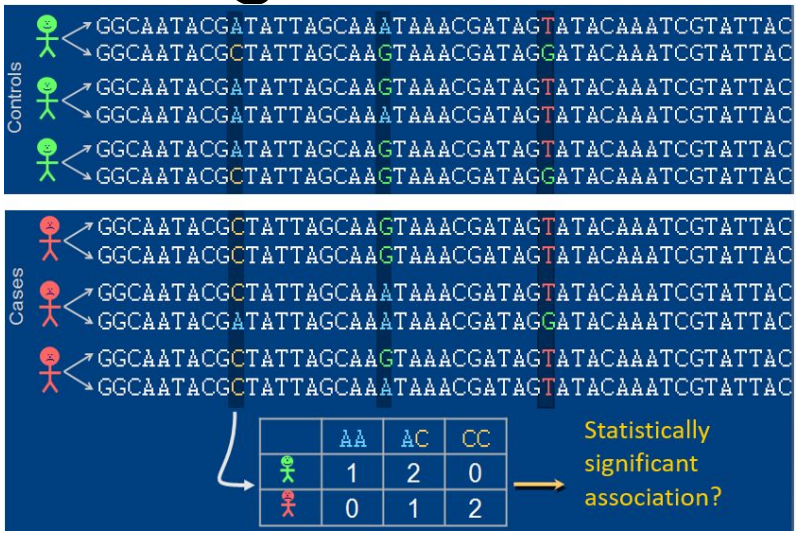
The Human Genome Project

2003

~ 3 billion nucleotides

Population sequencing identifies genetic variants

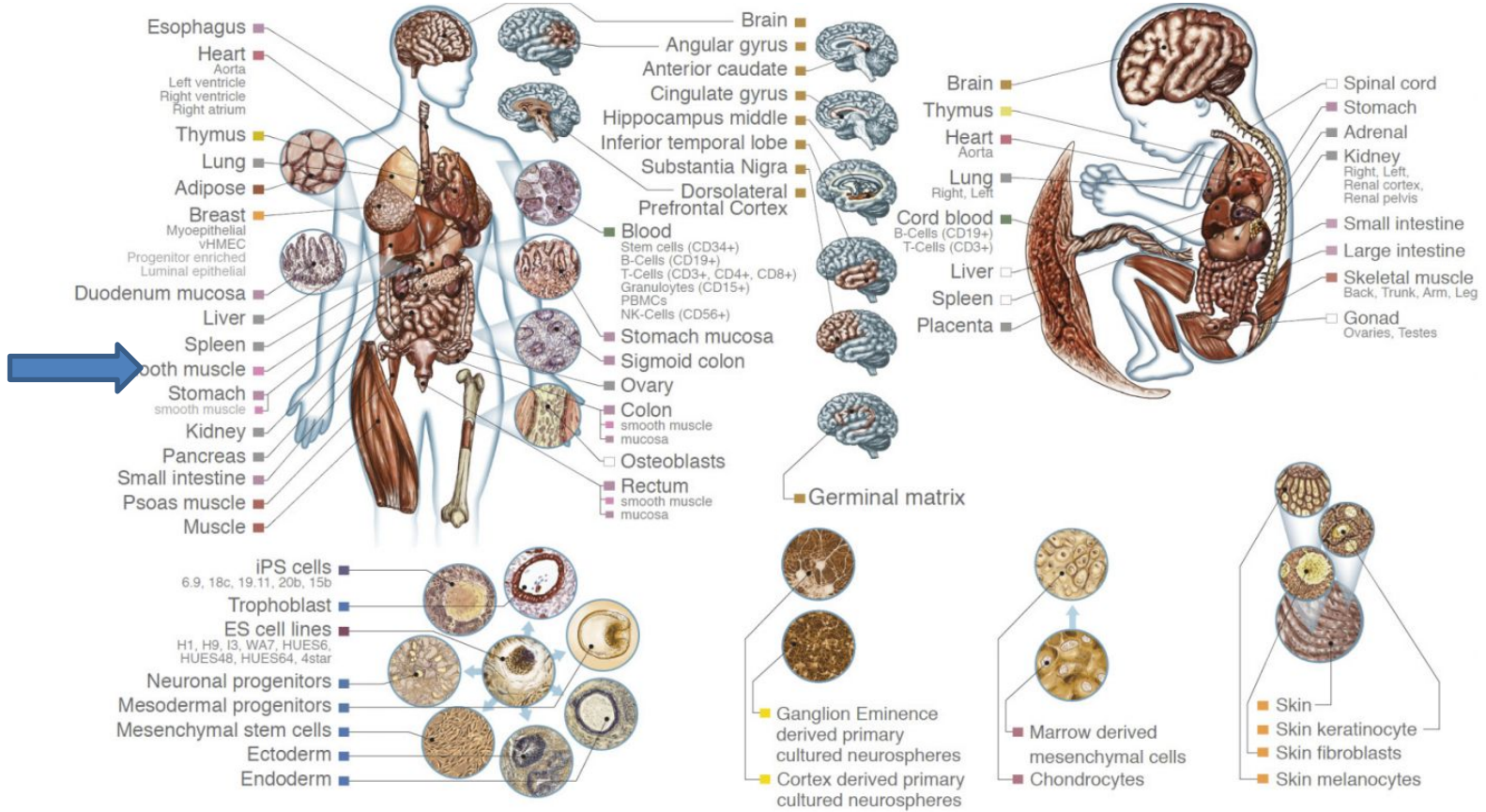




- 100,000s of personal genomes (**population-scale sequencing**)
- Millions of **genetic variants** (mutations) across individuals
- **Which variants are** benign and which ones are **related to disease**?
- What **functional DNA words** are these disease-associated variants disrupting?

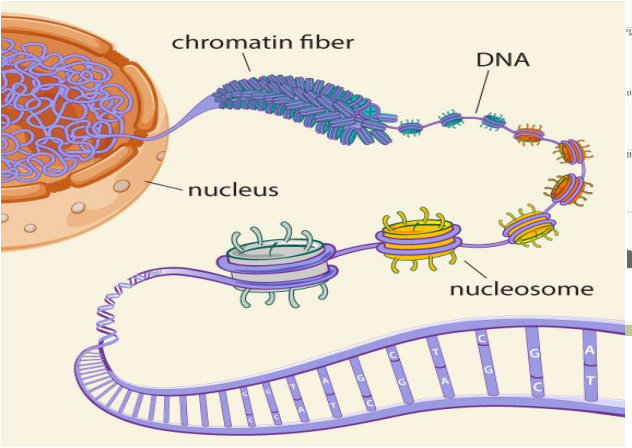
One genome ⇔ many cell types

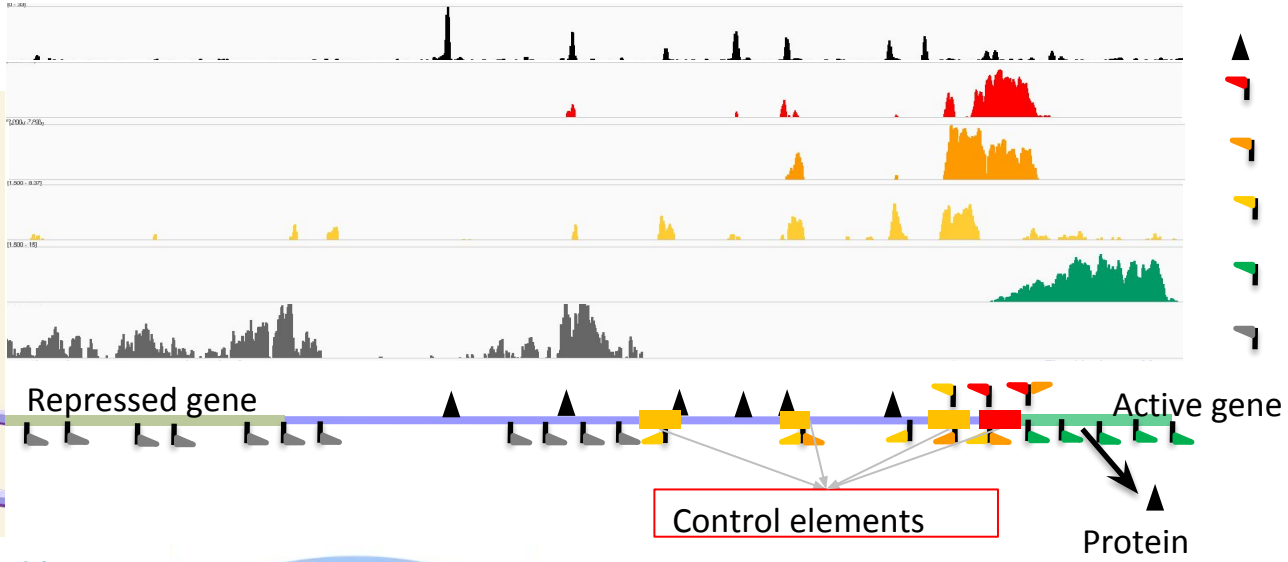
ACCAGTTACGACGG
TCAGGGTACTGATA
CCCCAAACCGTTGA
CCGCATTTACAGAC
GGGGTTTGGGTTTT
GCCCCACACAGGTA
CGTTAGCTACTGGT
TTAGCAATTTACCG
TTACAACGTTTACA
GGGTTACGGTTGGG
ATTTGAAAAAAAGT
TTGAGTTGGTTTTT
TCACGGTAGAACGT
ACCTTACAAA……



http://www.roadmapepigenomics.org/

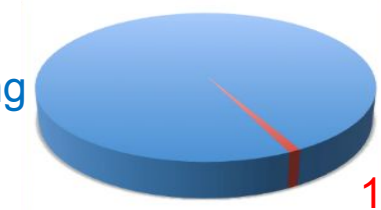
Biochemical markers of functional elements





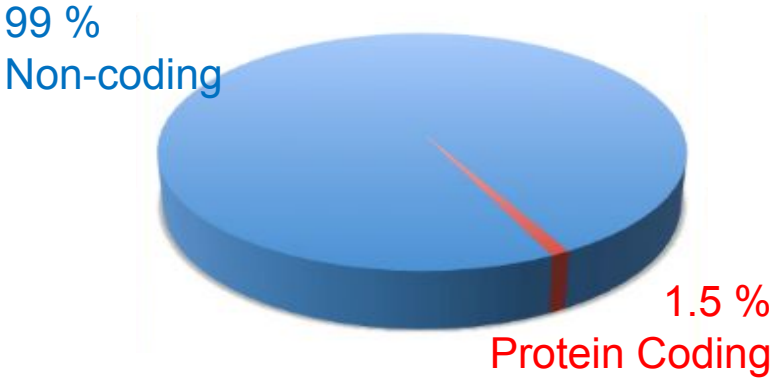
https://www.broadinstitute.org/news/1504

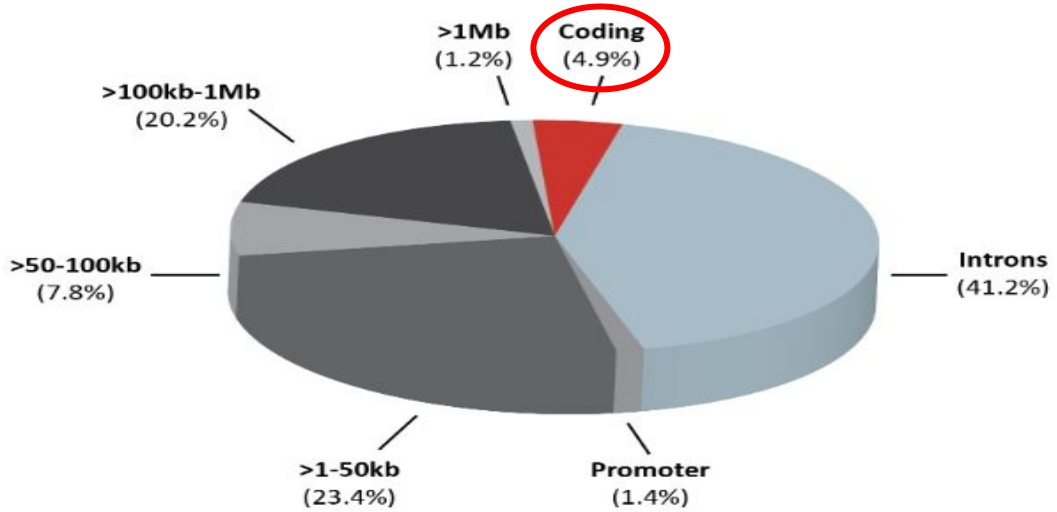
99 %
Non-coding



1.5 % Protein Coding

Why are control elements relevant in disease?



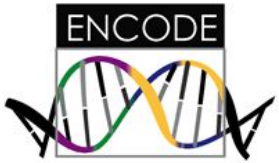


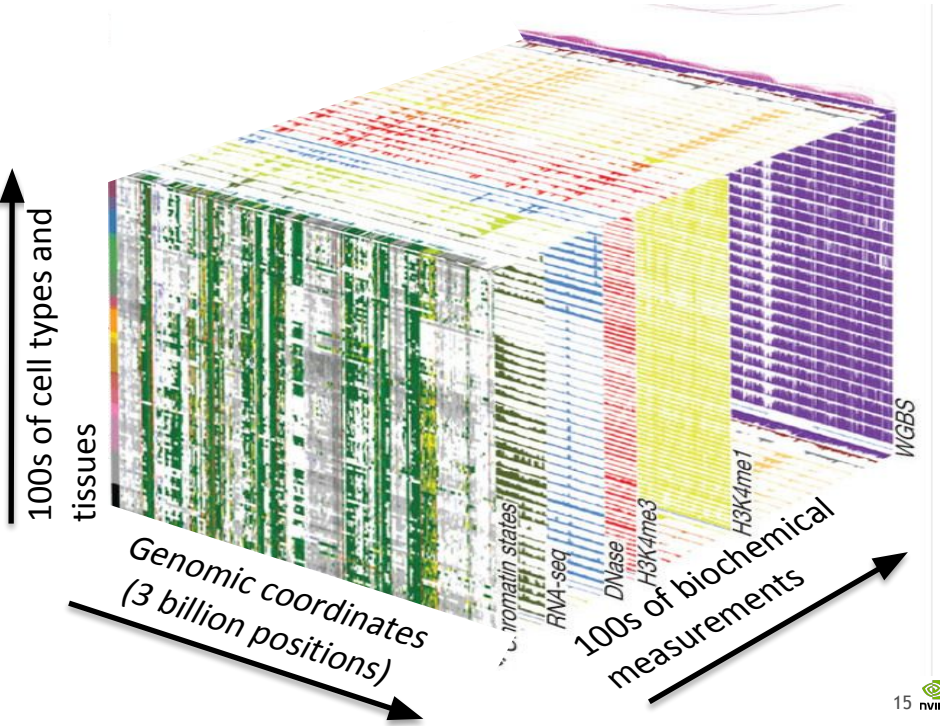
>90% of genetic variants associated with complex diseases do not disrupt protein-coding genes! Instead disrupting control elements!

Large-scale functional genomic data

- **“Functional Genomics”**
  Application of sequencing technology for **profiling 100s of biochemical markers of function** across the entire genome in 100s of different cell types and tissues

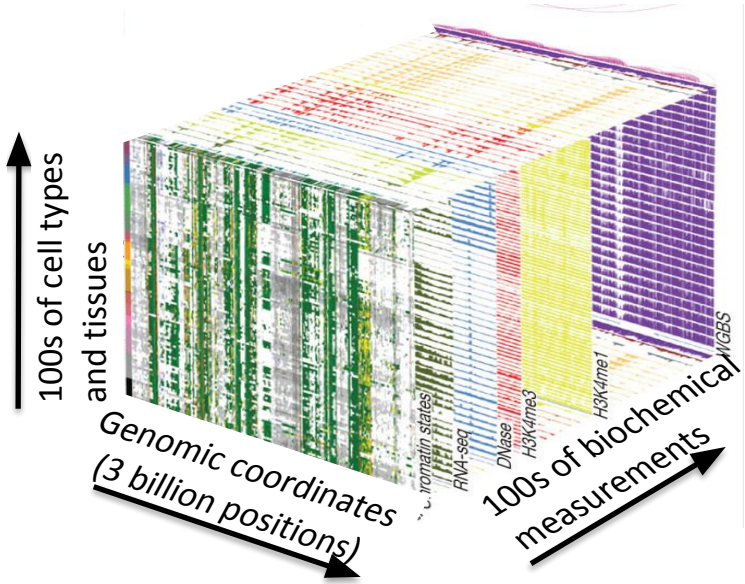
NIH funded collaborative consortia

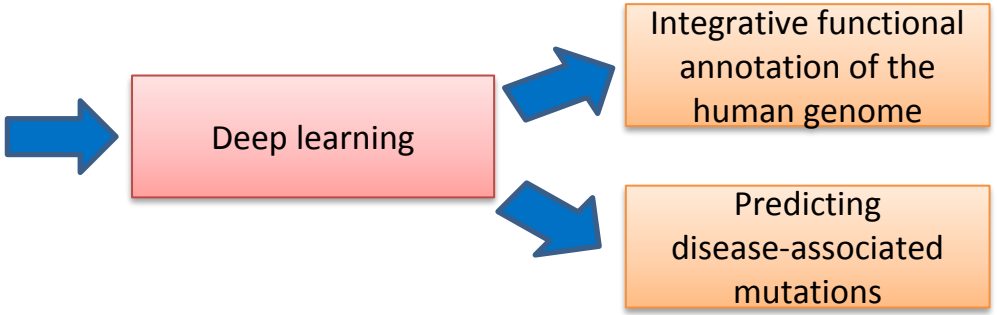




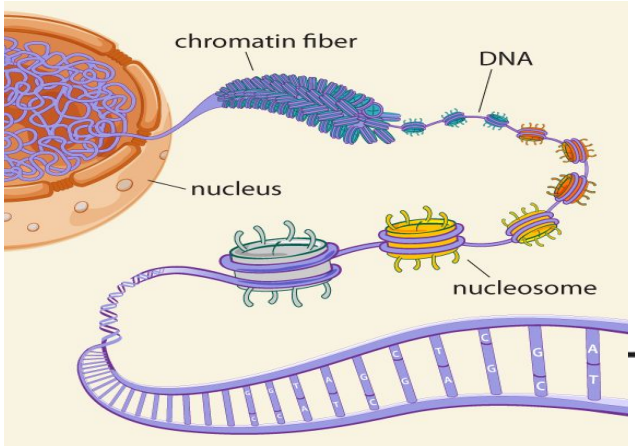
Why deep learning and functional genomics?

- **Terascale data cube** is rich and complex - deep learning is poised to integrate the heterogeneous data to decode genome function and its role in disease

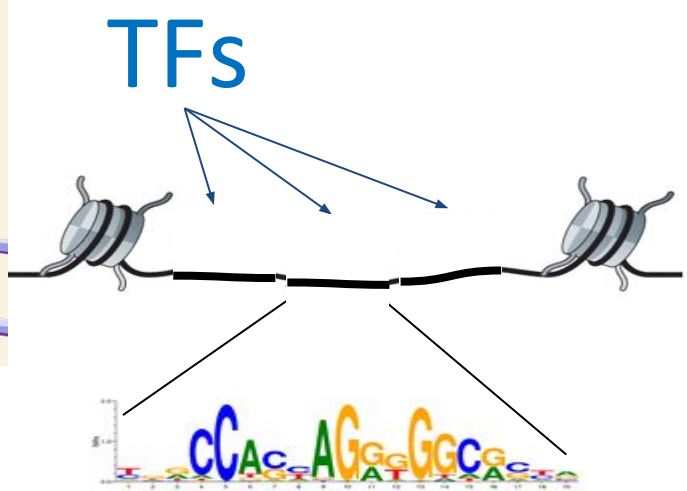




Proteins called transcription factors (TFs) bind **functional DNA words** to activate control elements

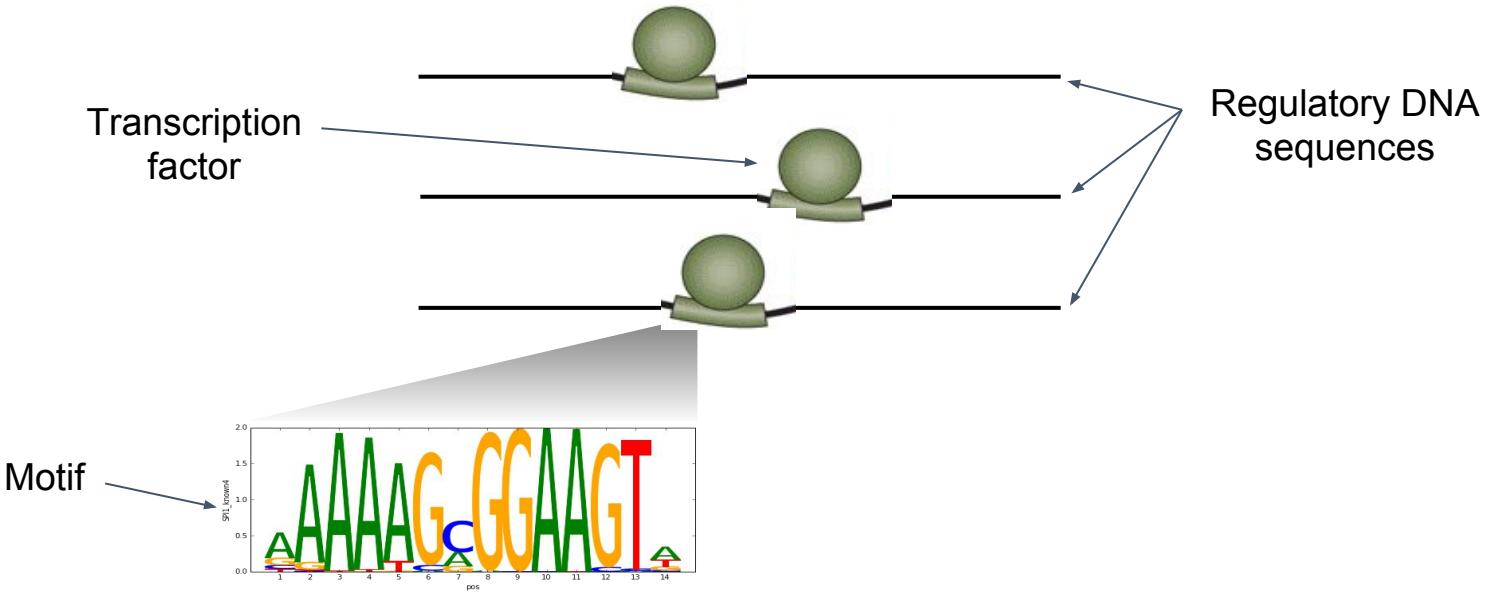


https://www.broadinstitute.org/news/1504



Adapted from Shlyueva et al. (2014) Nature Reviews Genetics.

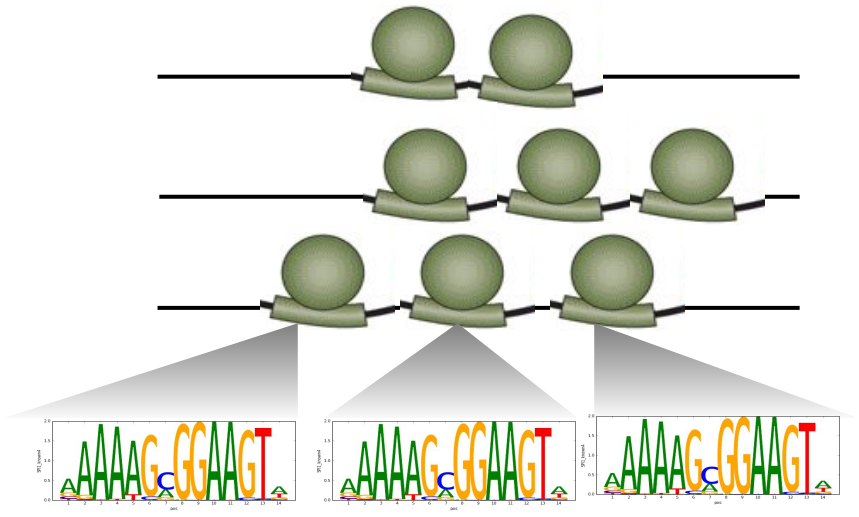
Key properties of control element sequences



TRANSCRIPTION FACTOR BINDING

Regulatory proteins called **transcription factors (TFs)** bind to high affinity sequence patterns (**motifs**) in regulatory DNA

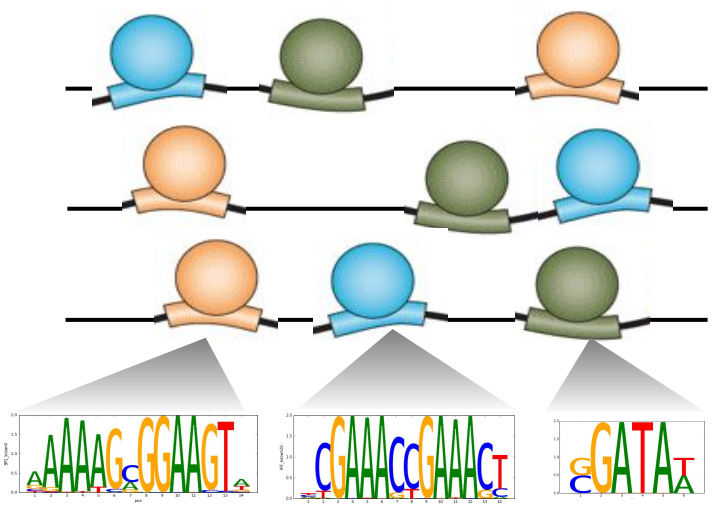
Key properties of control element sequences



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**

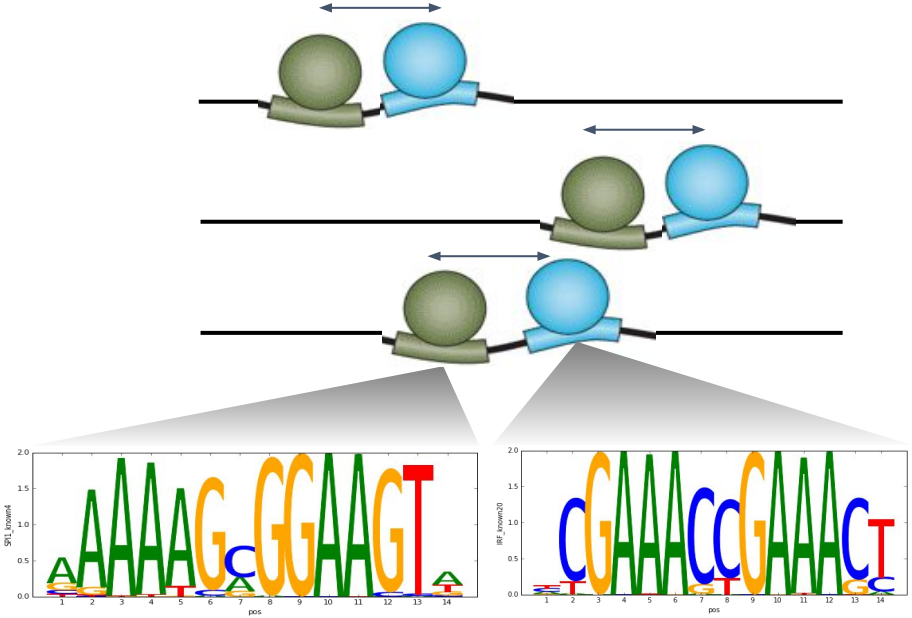
Key properties of control element sequences



HETEROTYPIC MOTIF COMBINATIONS

Regulatory sequences often bound by **combinations of TFs** resulting in **heterotypic clusters of motifs of different TFs**

Key properties of control element sequences



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**

DragoNN / SimDNA / KERAS / DeepLIFT

DragoNN

- A pedagogical tool for deep learning for genomics.
- Systematic model design and interpretation. Runs on
  - SimDNA for simulations
  - Keras for model development
  - DeepLIFT for model interpretation
- IPython notebook tutorials.
- A command line interface for modeling and interpretation.
- Docker build + image for cloud usage.

kundajelab / dragonn

Unwatch 15 | Unstar 75 | Fork 26

Code | Issues 2 | Pull requests 0 | Projects 0 | Wiki | Pulse | Graphs | Settings

A toolkit to learn how to model and interpret regulatory sequence data using deep learning.
http://kundajelab.github.io/dragonn/

Edit

deep-learning genomics Manage topics

107 commits | 6 branches | 3 releases | 5 contributors | MIT

Systematic simulations

Tutorials

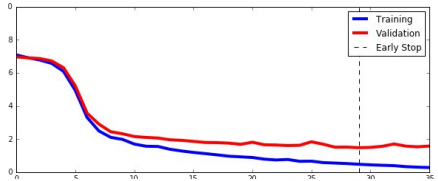
Command-line interface

Cloud-ready

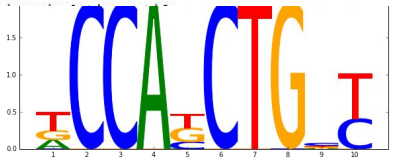
```
motif_density_localization_simulation_parameters = {
    "motif_name": "TAL1_known4",
    "seq_length": 1000,
    "center_size": 150,
    "min_motif_counts": 2,
    "max_motif_counts": 4,
    "num_pos": 3000,
    "num_neg": 3000,
    "GC_fraction": 0.4}
```

```
one_filter_dragonn_parameters = {
    'seq_length': 1000,
    'num_filters': [1],
    'conv_width': [10],
    'pool_width': 35}
```

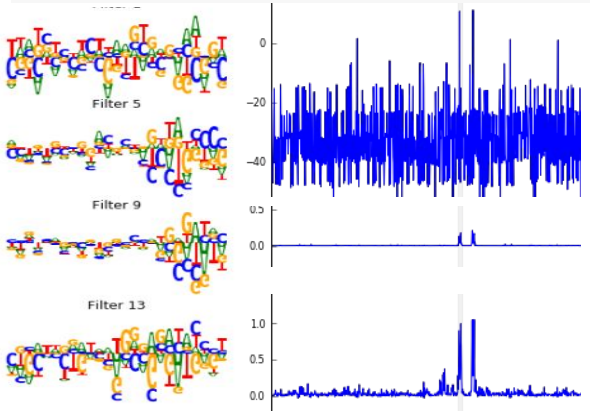
```
SequenceDNN_learning_curve(one_filter_dragonn)
```

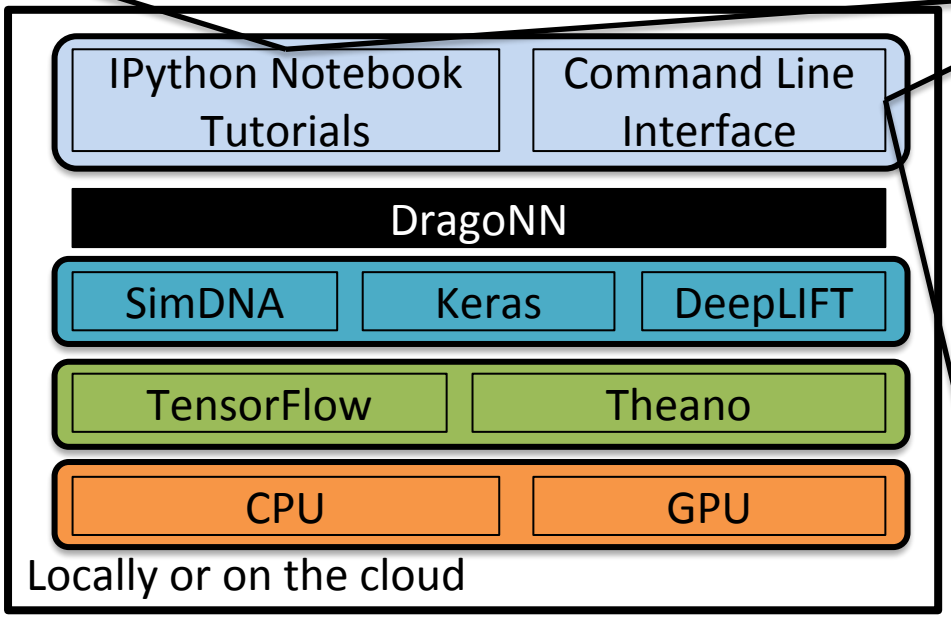



```
interpret_SequenceDNN_filters(multi_layer_dragonn, simulation_data)
```



```
interpret_data_with_SequenceDNN(multi_filter_dragonn, simulation_data)
```



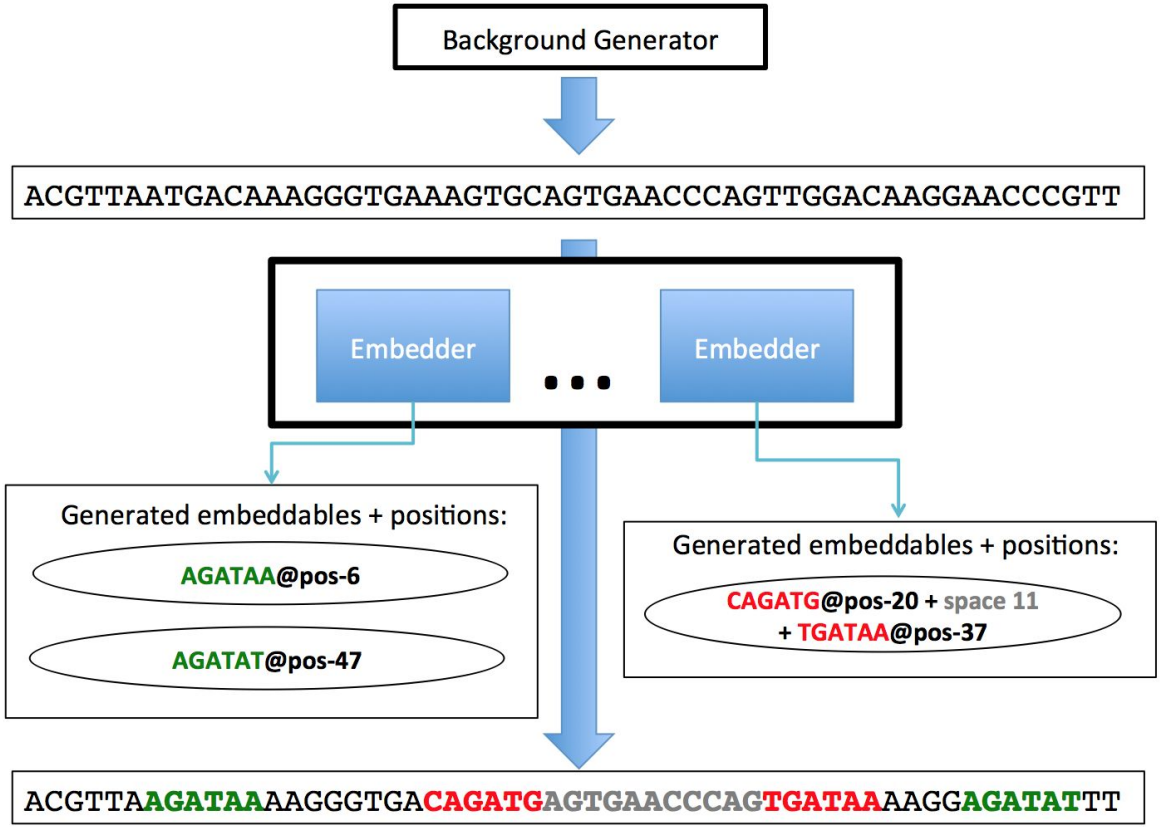



```
usage: dragonn [-h] {train,test,predict,interpret}

main script for DragoNN modeling of sequence data.

positional arguments:
  {train,test,predict,interpret}
                        dragonn command help
    train               model training help
    test                model testing help
    predict             model prediction help
    interpret           model interpretation help
```

SimDNA: Simulations of DNA

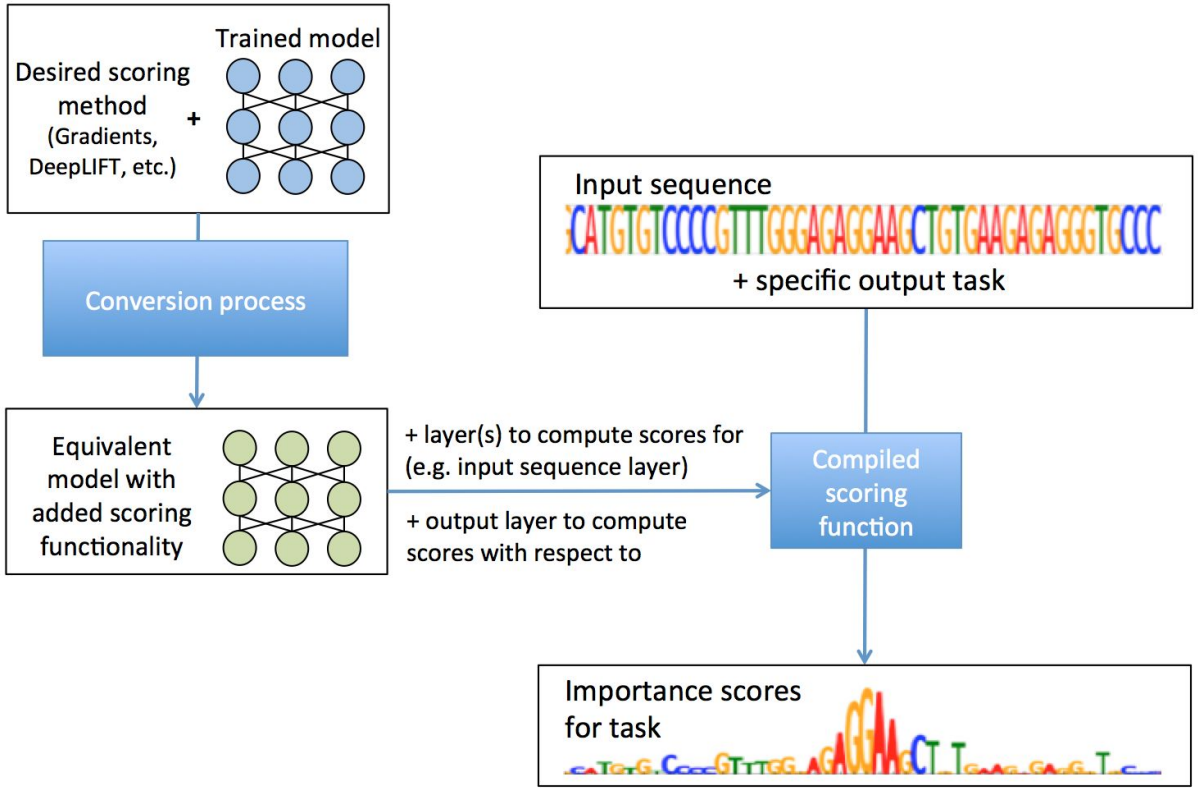


KERAS: Neural Network API

- Modular neural network written in Python
- Runs on TensorFlow and Theano
  - Theano excels at RNNs / LSTMs
- Keras library allows for easy and fast prototyping
- Runs on GPUs and CPUs
- Compatible with Python 2.7 – 3.5

DeepLIFT: Neural Network Interpretation API

- Modular feature importance API in Python 2.7 – 3.5
- Runs on TensorFlow and Theano
- Allows for easy interpretation of neural networks
- Runs on GPUs and CPUs



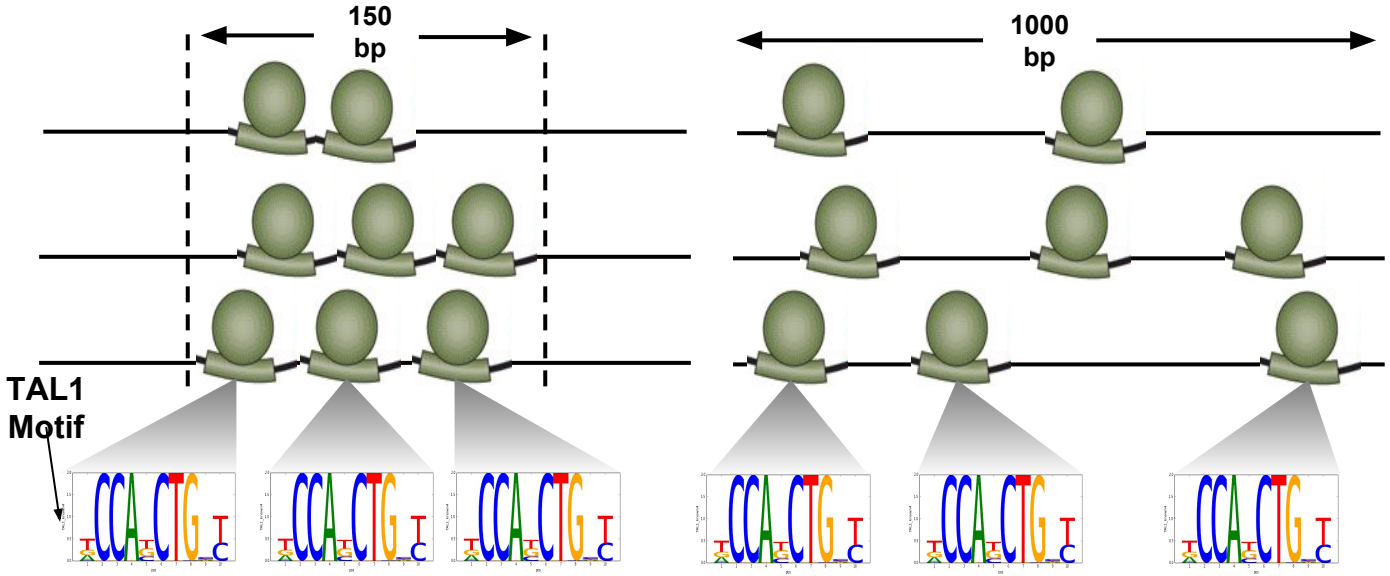
LAB DISCUSSION / OVERVIEW

LAB PROCESS

1. Simulate data for homotypic motif density localization
2. Architect DragoNN model
3. Train your model
4. Visualize results / interpret data
5. Modify architecture to enhance predictions and interpretation

Simulate homotypic motif density localization

```
simulation_parameters = {
    "motif_name":
    "TAL1_known4",
    "seq_length": 1000,
    "center_size": 150,
    "min_motif_counts": 2,
    "max_motif_counts": 4,
    "num_pos": 3000,
    "num_neg": 3000,
    "GC_fraction": 0.4}
```



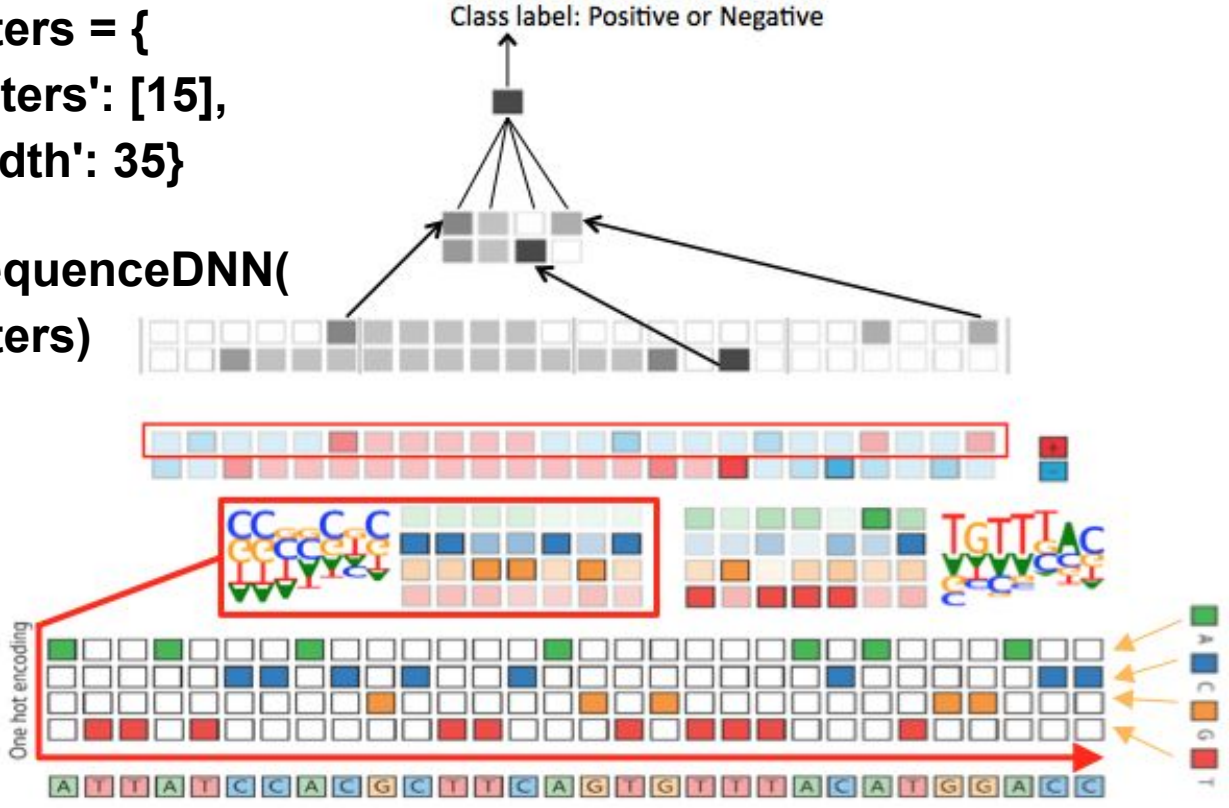
Positive set: 1000 base pairs (bp) sequences with 2-4 instances of TAL1 motif **in central 150bp**

Negative set: 1000 base pairs (bp) sequences with 2-4 instances of TAL1 motif **positioned anywhere in the 1000bp**

Architecting a DragoNN model

```
multi_filter_dragonn_parameters = {
    'seq_length': 1000, 'num_filters': [15],
    'conv_width': [10], 'pool_width': 35}

multi_filter_dragonn = get_SequenceDNN(
multi_filter_dragonn_parameters)
```



TRAIN A DRAGONN MODEL

multi_filter_dragonn = get_SequenceDNN(multi_filter_dragonn_parameters)

train_SequenceDNN(multi_filter_dragonn, simulation_data)

Epoch 1:

Train Loss: 0.7184 Balanced Accuracy: 50.27% auROC: 0.502 auPRC: 0.492 Recall at 5%|10%|20% FDR: 0.1%|0.1%|0.1% Num Positives: 1957 Num Negatives: 2043

Valid Loss: 0.7212 Balanced Accuracy: 48.60% auROC: 0.470 auPRC: 0.498 Recall at 5%|10%|20% FDR: 0.0%|0.0%|0.0% Num Positives: 528 Num Negatives: 472 *

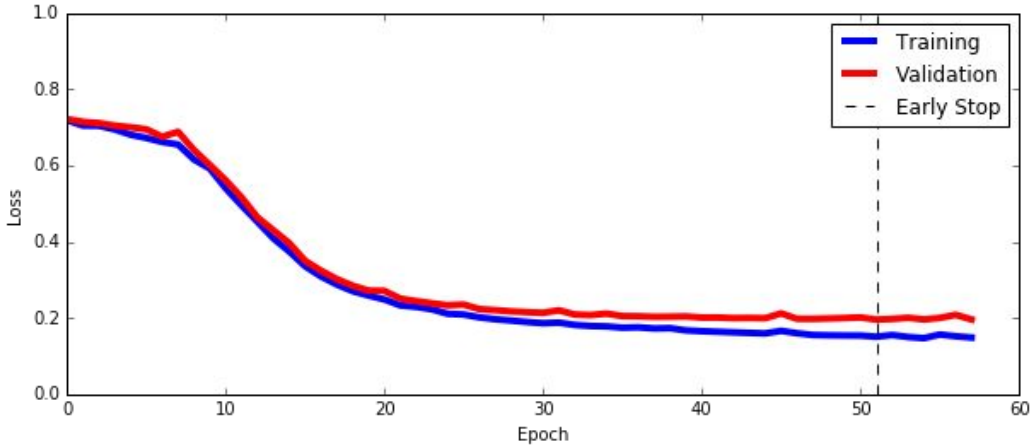
.....

Epoch 58:

Train Loss: 0.1484 Balanced Accuracy: 94.40% auROC: 0.988 auPRC: 0.987 Recall at 5%|10%|20% FDR: 93.4%|97.8%|99.7% Num Positives: 1957 Num Negatives: 2043

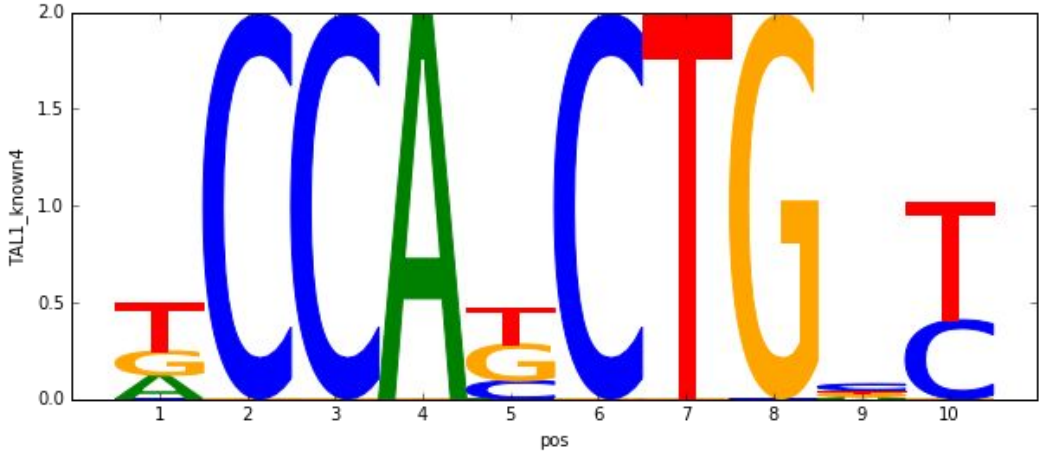
Valid Loss: 0.1958 Balanced Accuracy: 91.97% auROC: 0.976 auPRC: 0.977 Recall at 5%|10%|20% FDR: 88.6%|95.3%|99.2% Num Positives: 528 Num Negatives: 472

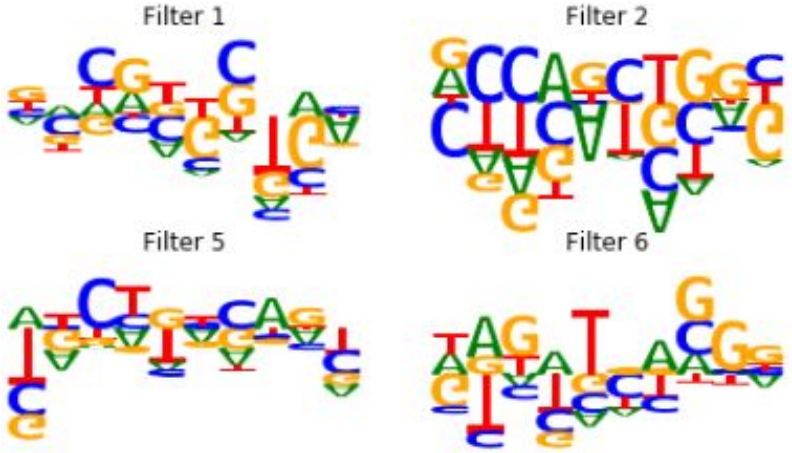
Finished training after 58 epochs.



Evaluate Learned Filters

interpret_SequenceDNN_filters(multi_filter_dragonn, simulation_data)



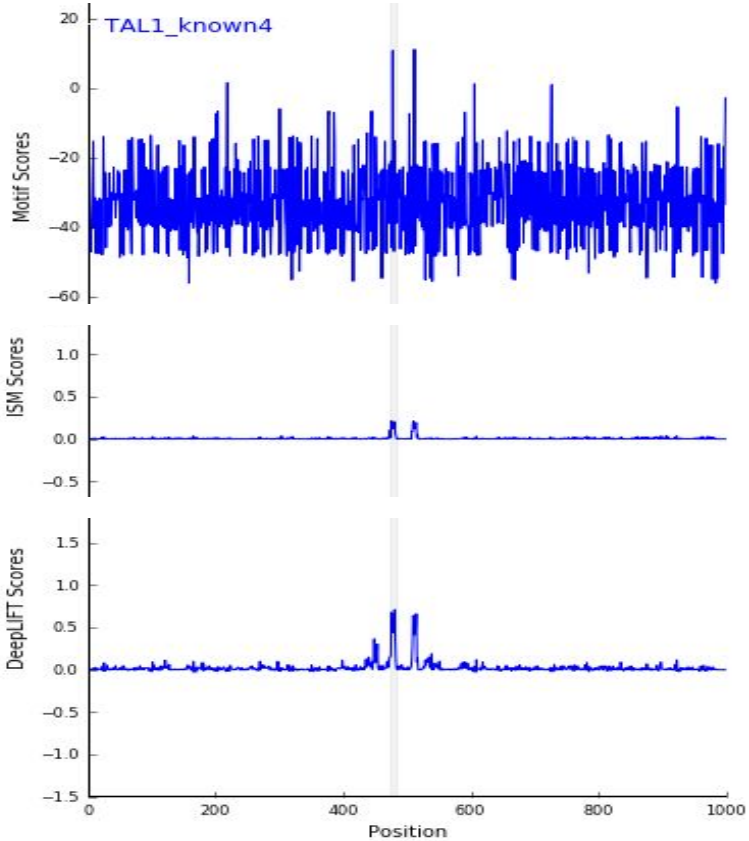


- Simulated motif is the “ground truth” feature
- Learned filters evaluated wrt simulated motif

Evaluate Feature Importances

interpret_data_with_SequenceDNN(
multi_filter_dragonn,
simulation_data)

- Motif scores are the “ground truth” feature importances
- Inferred feature importances inferred evaluated wrt motif scores



LAB ENVIRONMENT

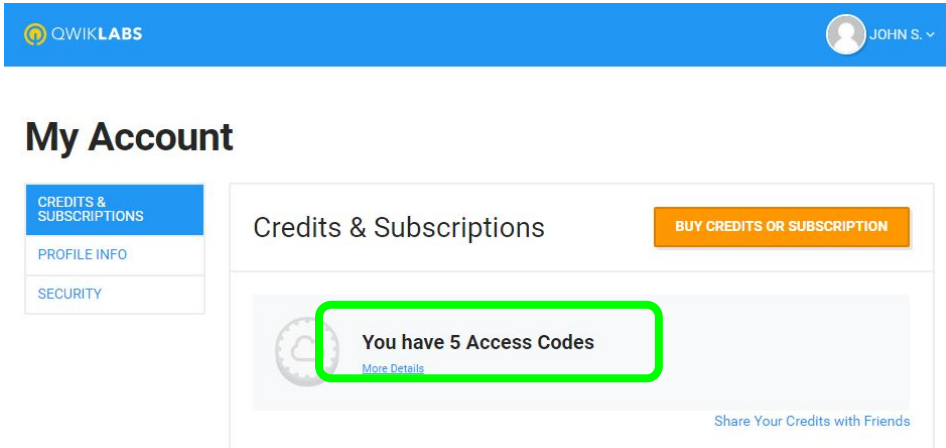
LAB CONNECTION INSTRUCTIONS - Part 1

Go to nvlabs.qwiklab.com

Sign in or create an account

Check for Access Codes (each day):

- Click My Account
- Click Credits & Subscriptions



If no Access Codes, ask for paper one from TA.

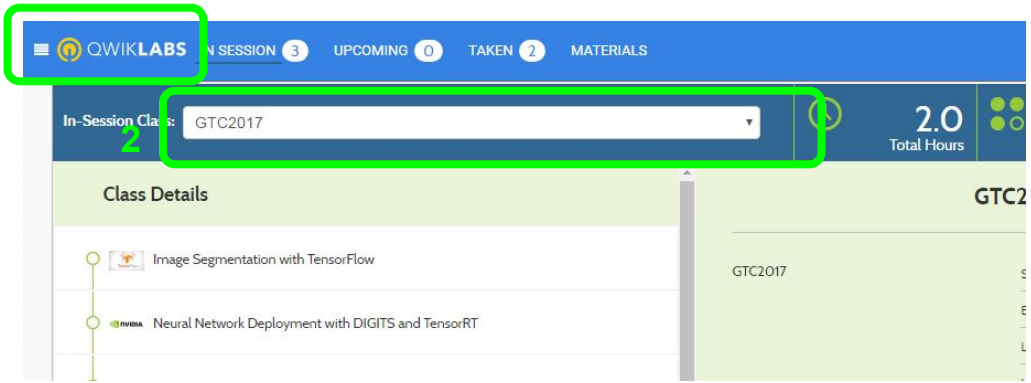
Please tear in half once used

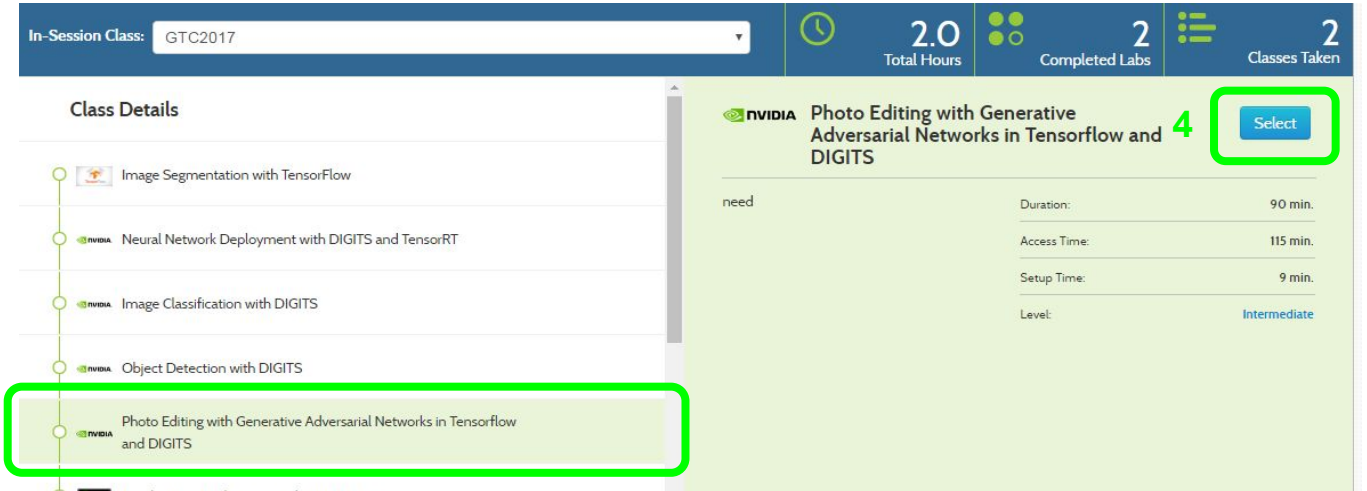
An Access Code is needed to start the lab

WIFI SSID: **GTC_Hands_On**	Password: **HandsOnGpu**

LAB CONNECTION INSTRUCTIONS - Part 2

1. Click Qwiklabs in upper-left
2. Select GTC2017 Class
3. Find lab and click on it
4. Click on Select
5. Click Start Lab



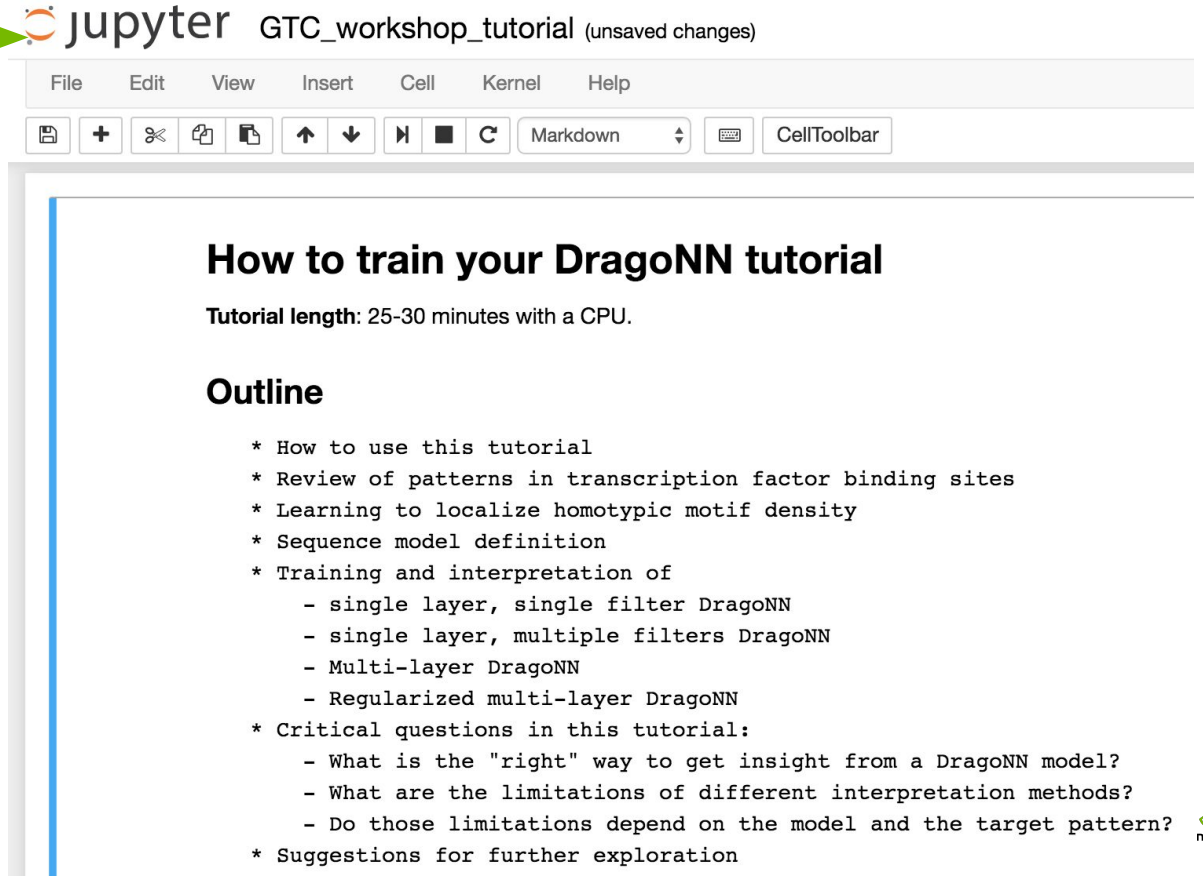


WIFI SSID:
GTC_Hands_On

Password:
HandsOnGpu

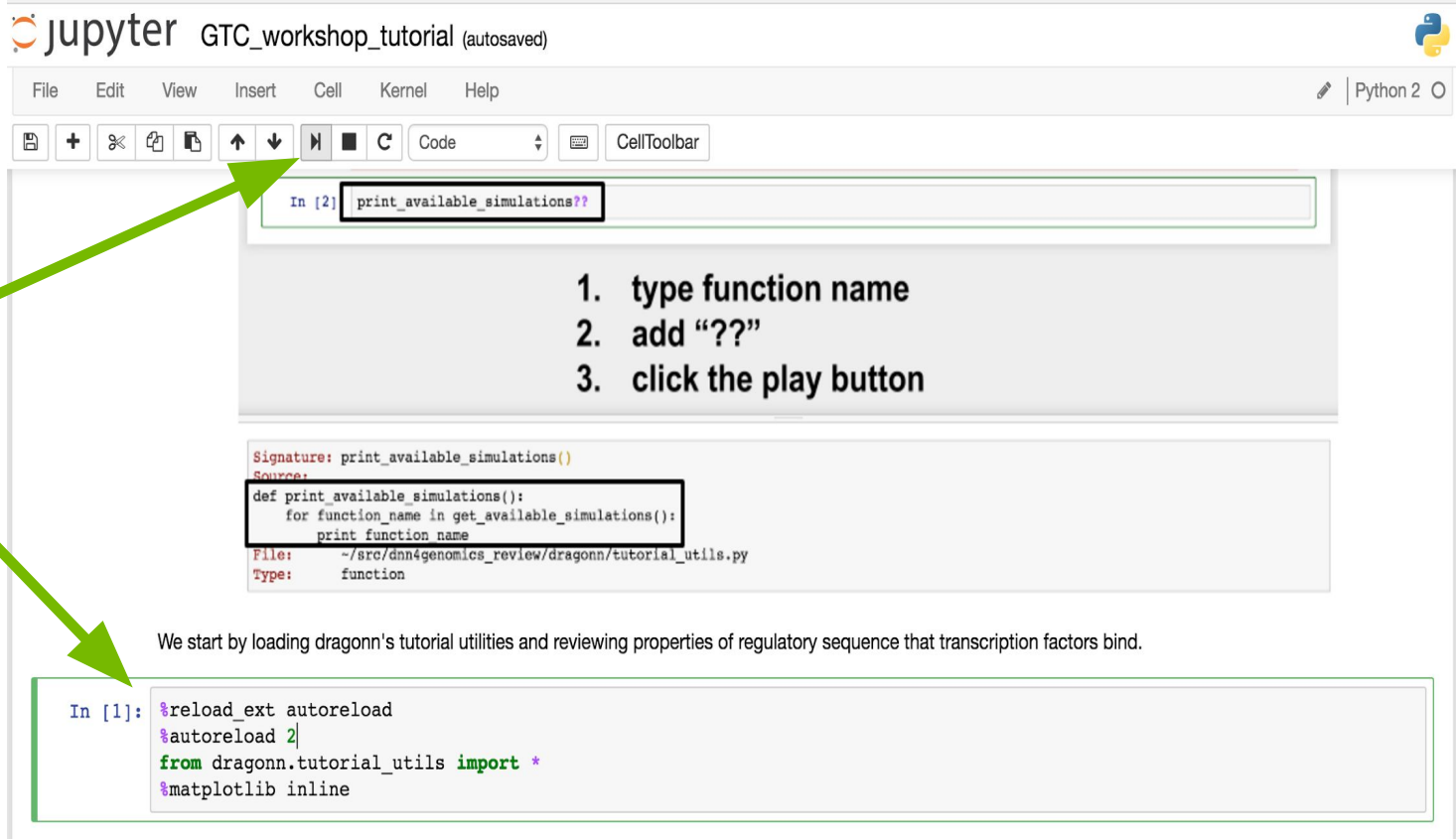
ACCESSING LAB INSTRUCTIONS

Should see Jupyter notebook



ACCESSING LAB INSTRUCTIONS

Place cursor in code block and click execute button



LAB REVIEW

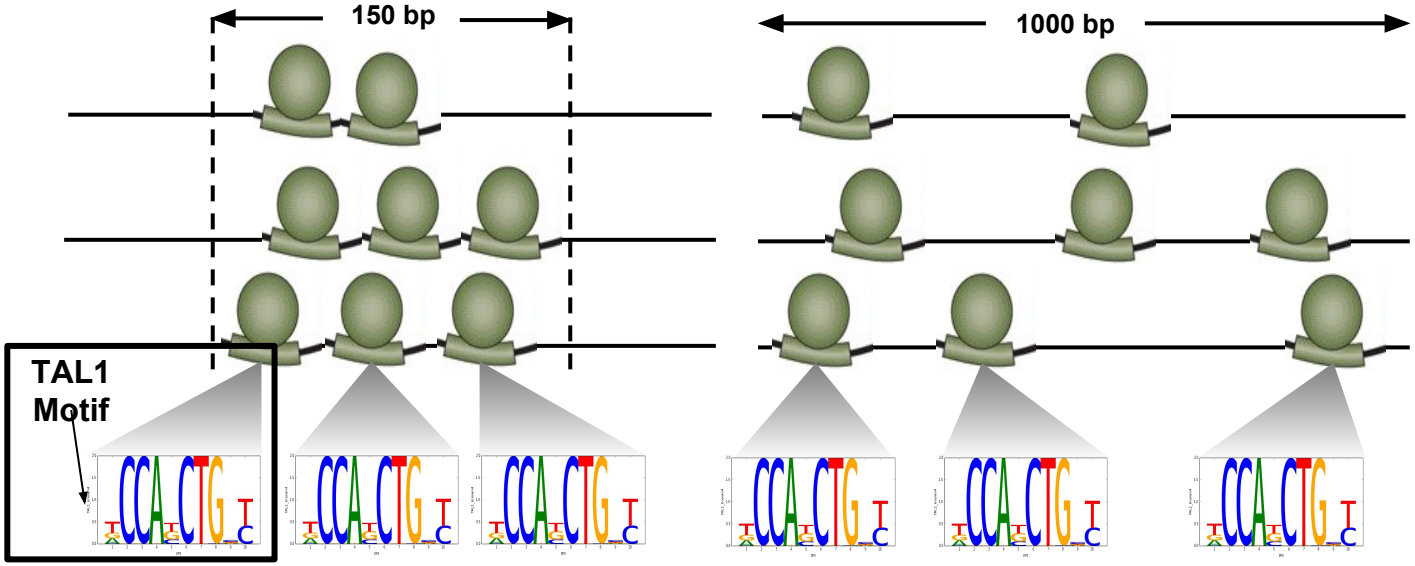
LAB REVIEW

1. Generate simulation data
2. Architect and train single filter DragoNN model
3. Architect and train multi-filter model
4. Visualize results
5. Interpret model and data
6. Introduce enhancements

Generating simulation data

Homotypic motif density localization

```
simulation_parameters = {
    "motif_name":
    "TAL1_known4",
    "seq_length": 1000,
    "center_size": 150,
    "min_motif_counts": 2,
    "max_motif_counts": 4,
    "num_pos": 3000,
    "num_neg": 3000,
    "GC_fraction": 0.4}
```

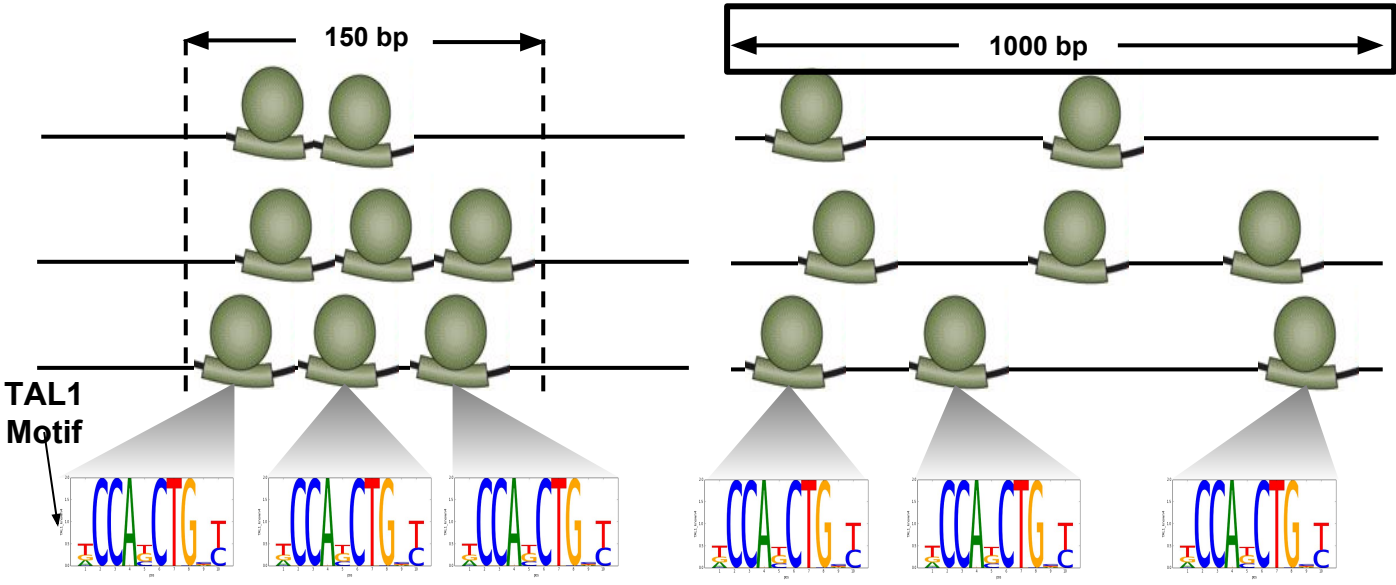


Positive set: 1000 base pairs (bp) sequences with 2-4 instances of TAL1 motif **in central 150bp**

Negative set: 1000 base pairs (bp) sequences with 2-4 instances of TAL1 motif **positioned anywhere in the 1000bp**

Homotypic motif density localization

```
simulation_parameters = {
    "motif_name":
    "TAL1_known4",
    "seq_length": 1000,
    "center_size": 150,
    "min_motif_counts": 2,
    "max_motif_counts": 4,
    "num_pos": 3000,
    "num_neg": 3000,
    "GC_fraction": 0.4}
```

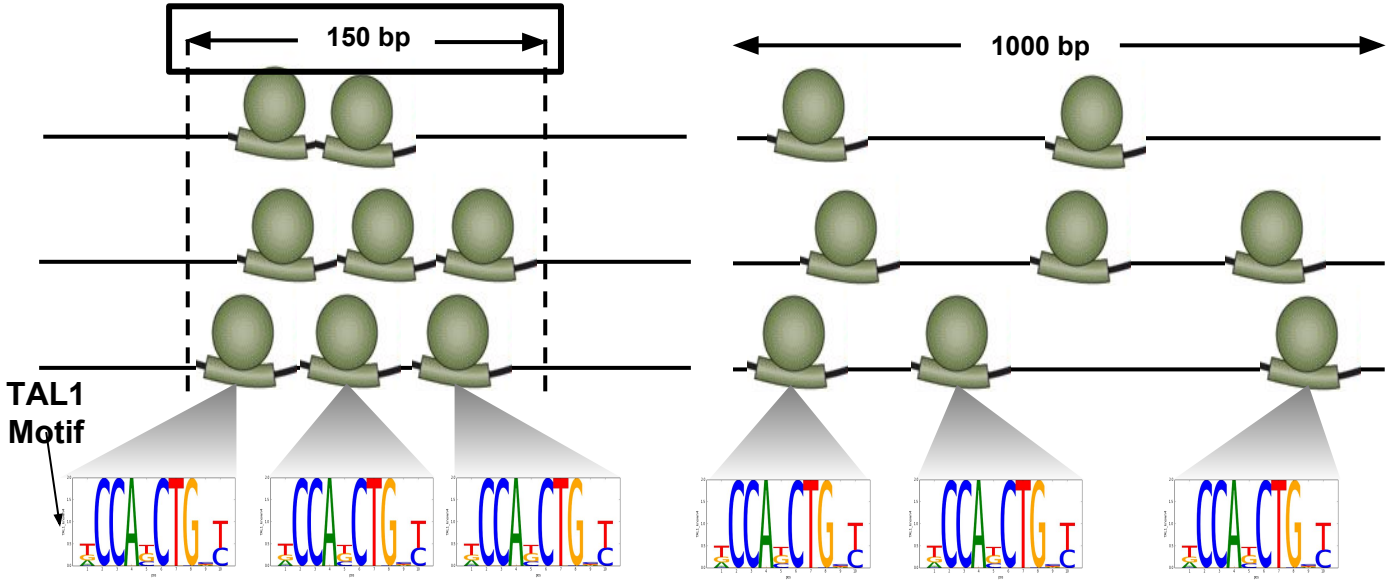


Positive set: 1000 base pairs (bp) sequences with 2-4 instances of TAL1 motif **in central 150bp**

Negative set: 1000 base pairs (bp) sequences with 2-4 instances of TAL1 motif **positioned anywhere in the 1000bp**

Homotypic motif density localization

```
simulation_parameters = {
    "motif_name":
    "TAL1_known4",
    "seq_length": 1000,
    "center_size": 150,
    "min_motif_counts": 2,
    "max_motif_counts": 4,
    "num_pos": 3000,
    "num_neg": 3000,
    "GC_fraction": 0.4}
```

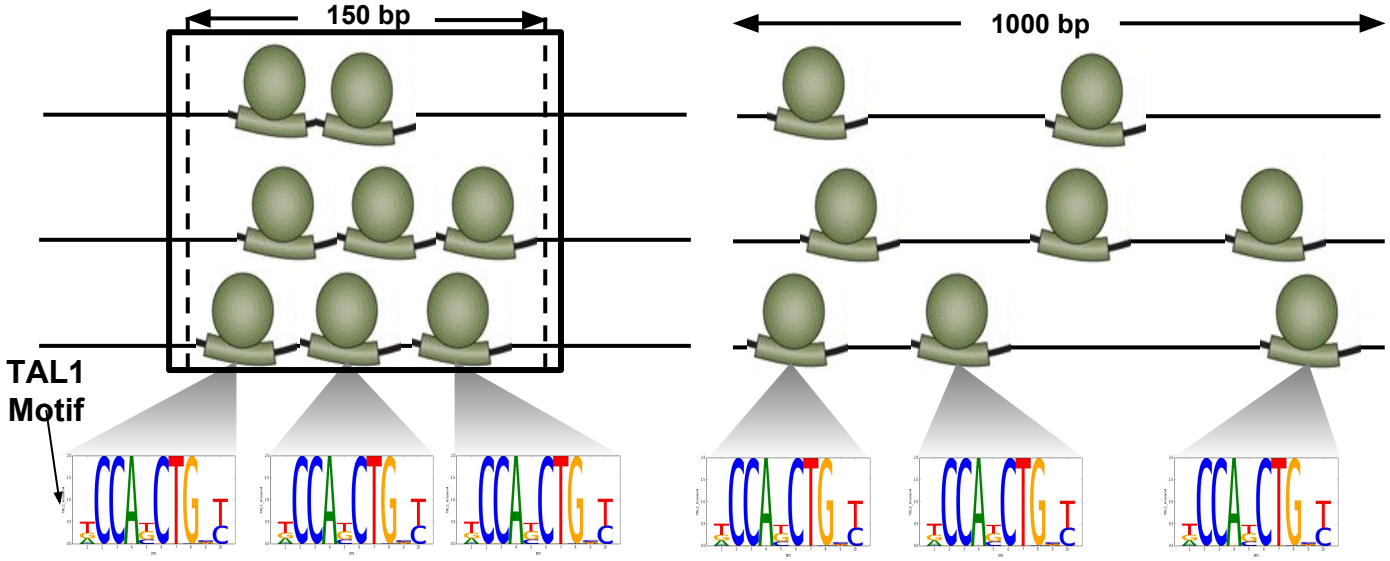


Positive set: 1000 base pairs (bp) sequences with 2-4 instances of TAL1 motif **in central 150bp**

Negative set: 1000 base pairs (bp) sequences with 2-4 instances of TAL1 motif **positioned anywhere in the 1000bp**

Homotypic motif density localization

```
simulation_parameters = {
    "motif_name":
    "TAL1_known4",
    "seq_length": 1000,
    "center_size": 150,
    "min_motif_counts": 2,
    "max_motif_counts": 4,
    "num_pos": 3000,
    "num_neg": 3000,
    "GC_fraction": 0.4}
```



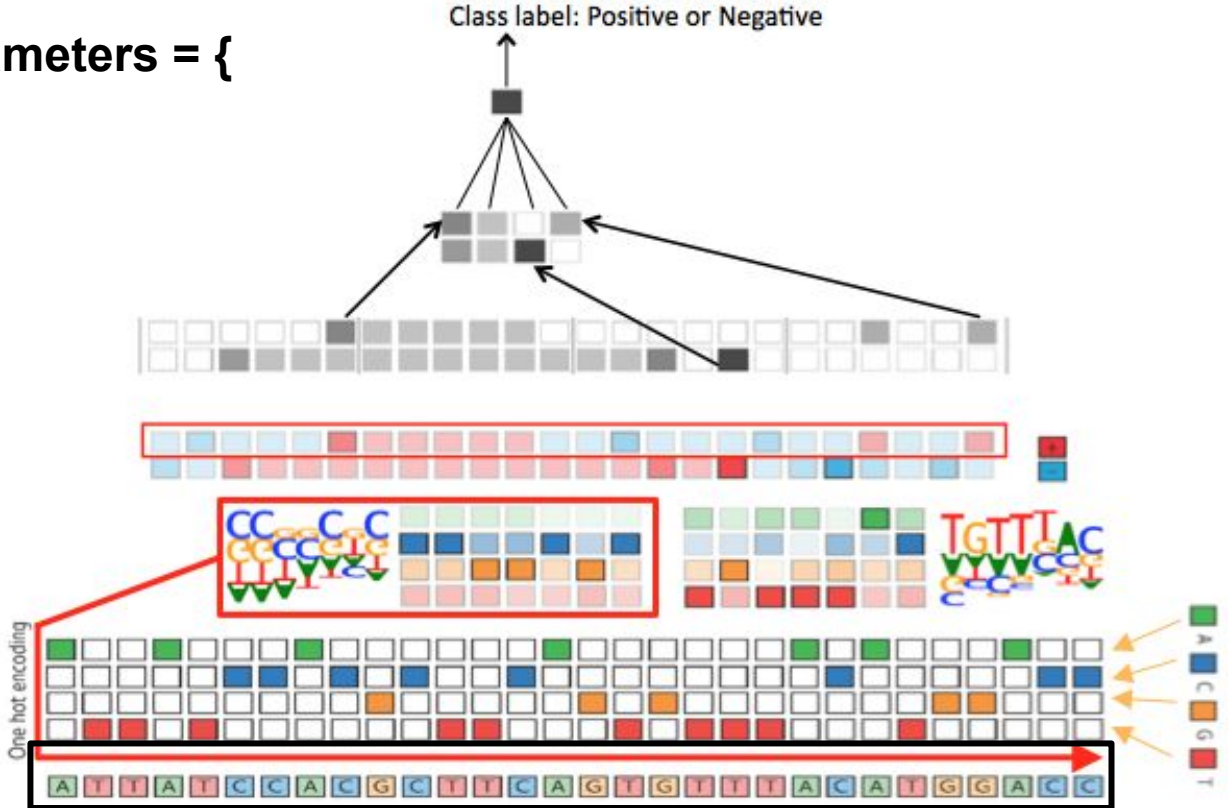
Positive set: 1000 base pairs (bp) sequences with 2-4 instances of TAL1 motif **in central 150bp**

Negative set: 1000 base pairs (bp) sequences with 2-4 instances of TAL1 motif **positioned anywhere in the 1000bp**

Architecting a DragoNN model

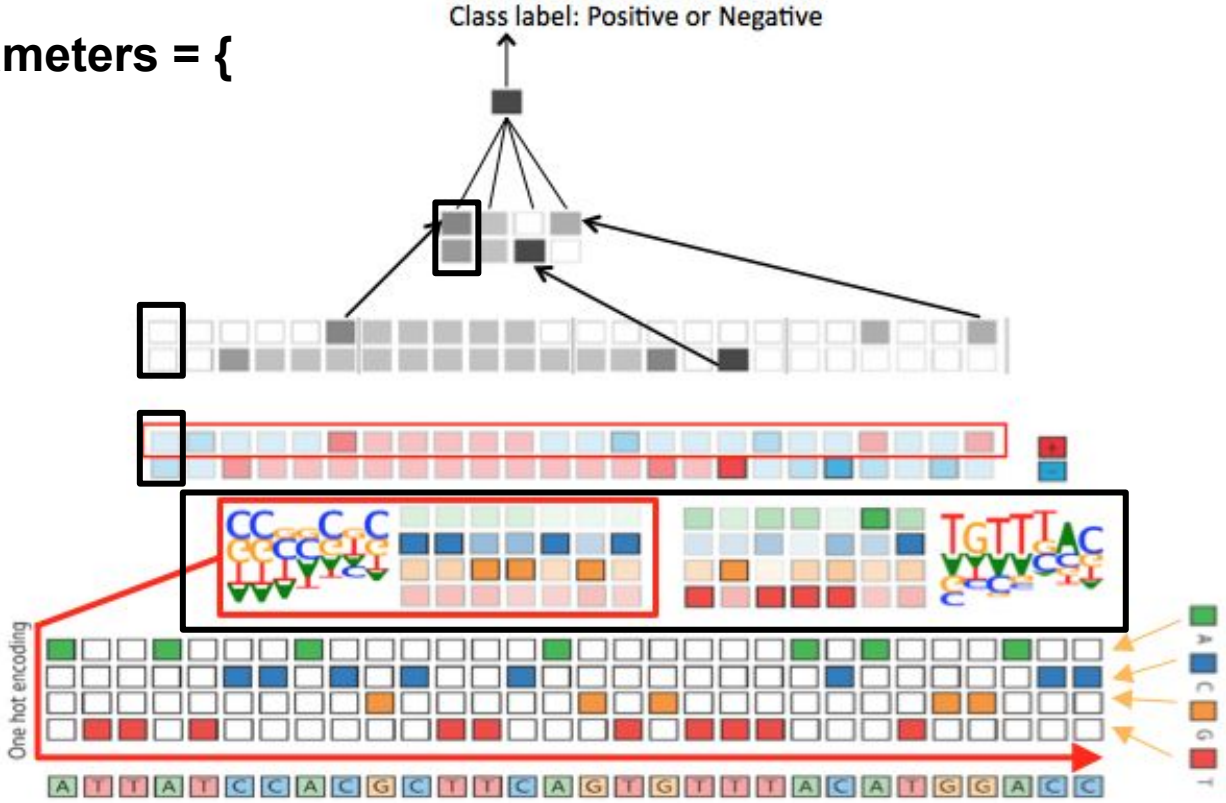
ARCHITECT one_filter_dragonn MODEL

```
one_filter_dragonn_parameters = {
    'seq_length': 1000,
    'num_filters': [1],
    'conv_width': [10],
    'pool_width': 35}
```



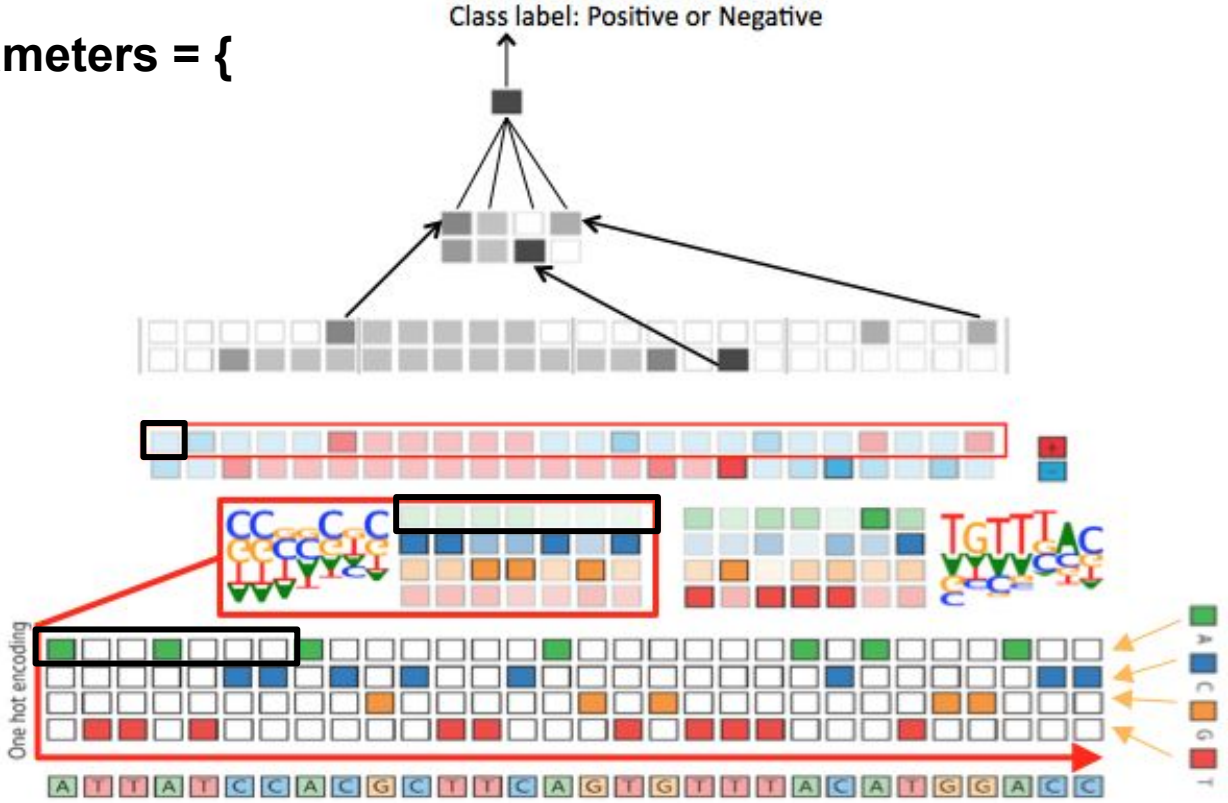
ARCHITECT one_filter_dragonn MODEL

```
one_filter_dragonn_parameters = {
    'seq_length': 1000,
    'num_filters': [1],
    'conv_width': [10],
    'pool_width': 35}
```



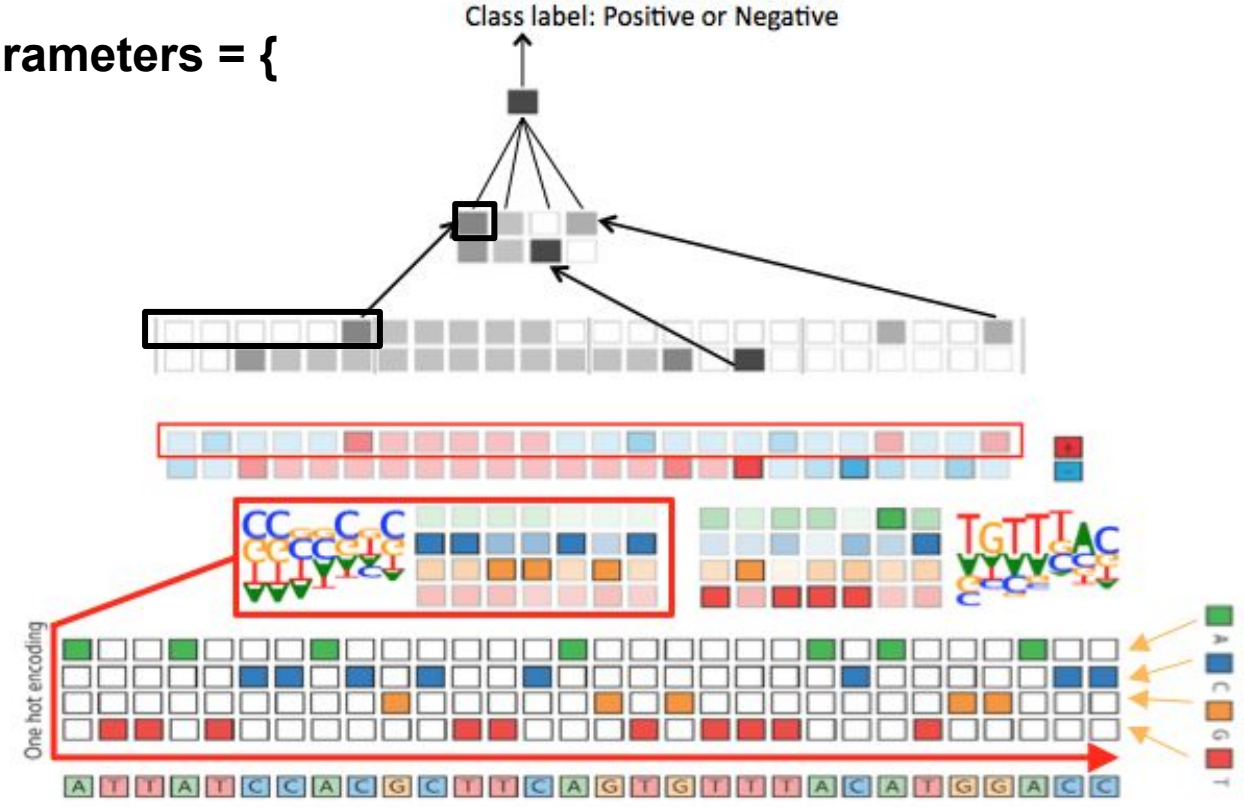
ARCHITECT one_filter_dragonn MODEL

```
one_filter_dragonn_parameters = {
    'seq_length': 1000,
    'num_filters': [1],
    'conv_width': [10],
    'pool_width': 35}
```



ARCHITECT one_filter_dragonn MODEL

```
multi_filter_dragonn_parameters = {
    'seq_length': 1000,
    'num_filters': [1],
    'conv_width': [10],
    'pool_width': 35}
```



Training the Model

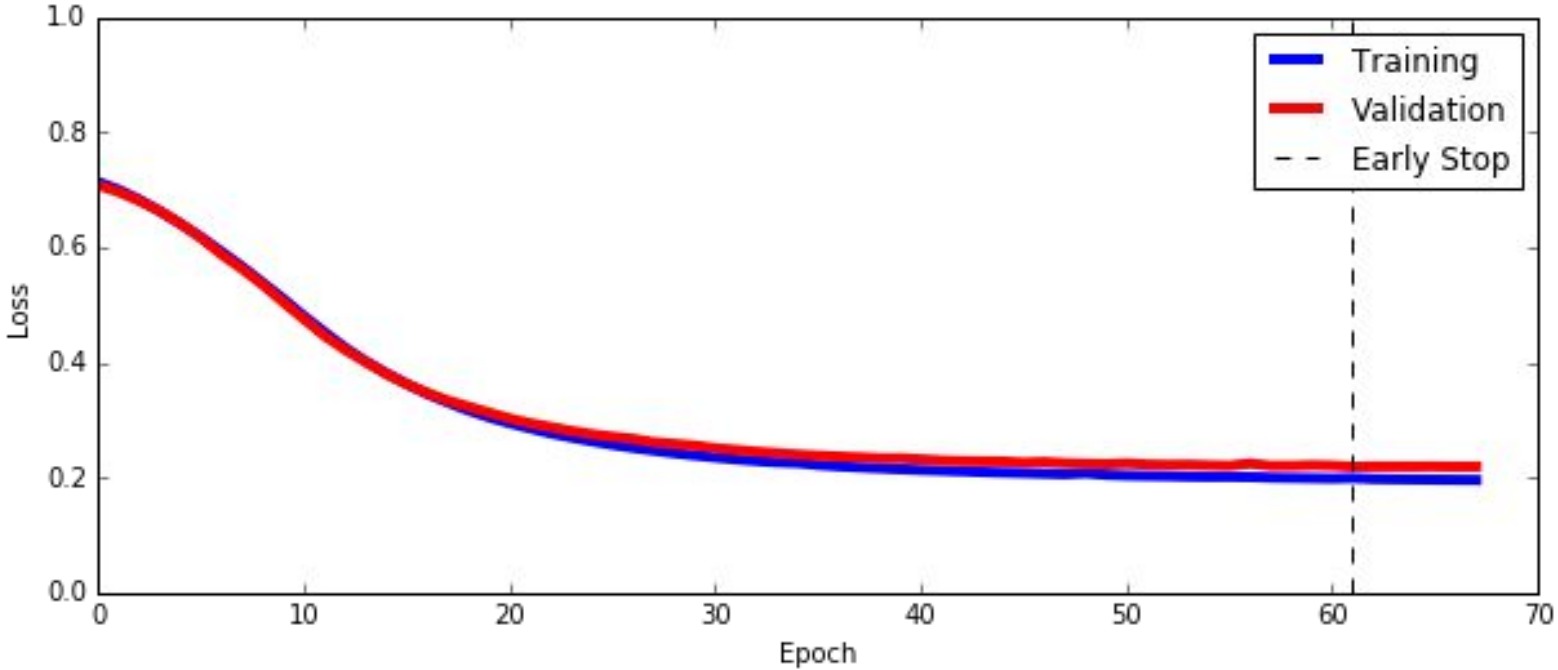
TRAIN one_filter_dragonn MODEL

one_filter_dragonn = get_SequenceDNN(one_filter_dragonn_parameters)
train_SequenceDNN(one_filter_dragonn, simulation_data)

Epoch 1:
Train Loss: 0.7134 Balanced Accuracy: 50.86% auROC: 0.509 auPRC: 0.502 Recall at 5%|10%|20%
FDR: 0.0%|0.0%|0.0% Num Positives: 1957 Num Negatives: 2043
Valid Loss: 0.7077 Balanced Accuracy: 53.47% auROC: 0.524 auPRC: 0.552 Recall at 5%|10%|20%
FDR: 0.6%|0.6%|0.6% Num Positives: 528 Num Negatives: 472 *
.....
Epoch 68:
Train Loss: 0.1956 Balanced Accuracy: 91.63% auROC: 0.977 auPRC: 0.975 Recall at 5%|10%|20%
FDR: 86.1%|93.5%|98.6% Num Positives: 1957 Num Negatives: 2043
Valid Loss: 0.2189 Balanced Accuracy: 91.40% auROC: 0.970 auPRC: 0.972 Recall at 5%|10%|20%
FDR: 78.4%|94.9%|98.9% Num Positives: 528 Num Negatives: 472
Finished training after 68 epochs.

VISUALIZE one_filter_dragonn RESULTS

SequenceDNN_learning_curve(one_filter_dragonn)



ARCHITECT MULTI-FILTER DRAGONN MODEL

```
multi_filter_dragonn_parameters = {
    'seq_length': 1000,
    'num_filters': [15], ## notice the change from 1 filter to 15 filters
    'conv_width': [10],
    'pool_width': 35}
multi_filter_dragonn = get_SequenceDNN(multi_filter_dragonn_parameters)
train_SequenceDNN(multi_filter_dragonn, simulation_data)
SequenceDNN_learning_curve(multi_filter_dragonn)
```

TRAIN MULTI-FILTER DRAGONN MODEL

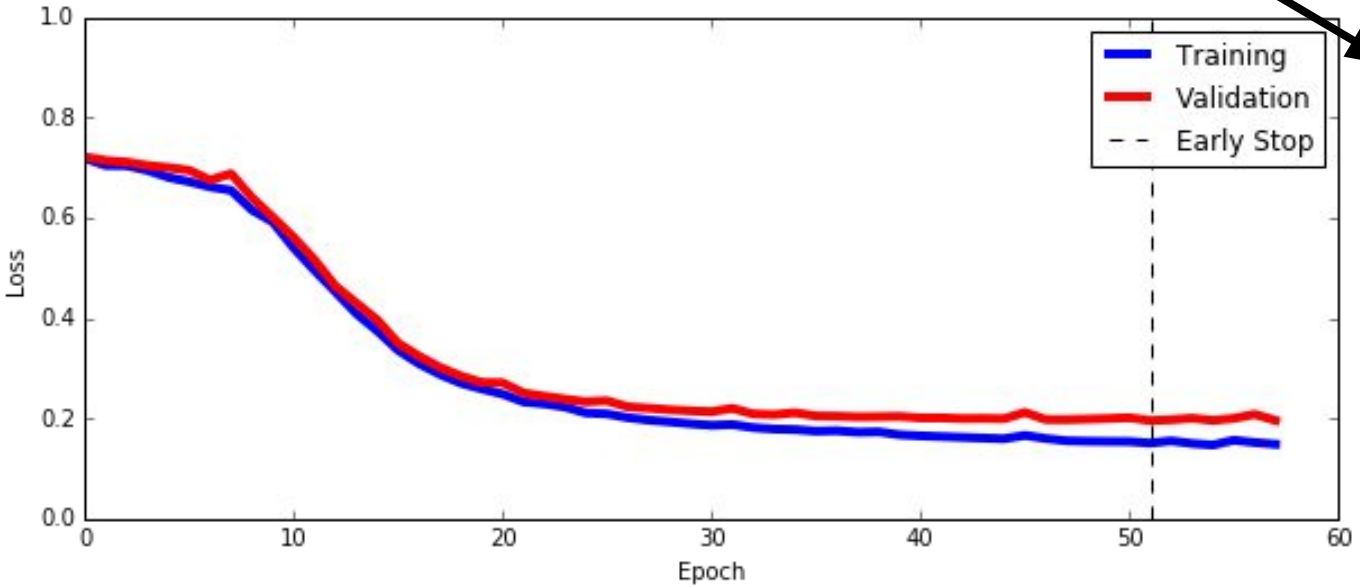
Epoch 58:

Train Loss: 0.1484 Balanced Accuracy: 94.40% auROC: 0.988 auPRC: 0.987

Recall at 5%|10%|20% FDR: 93.4%|97.8%|99.7% Num Positives: 1957 Num Negatives: 2043

Valid Loss: 0.1958 Balanced Accuracy: 91.97% **auROC: 0.976** auPRC: 0.977

Recall at 5%|10%|20% FDR: 88.6%|95.3%|99.2% Num Positives: 528 Num Negatives: 47

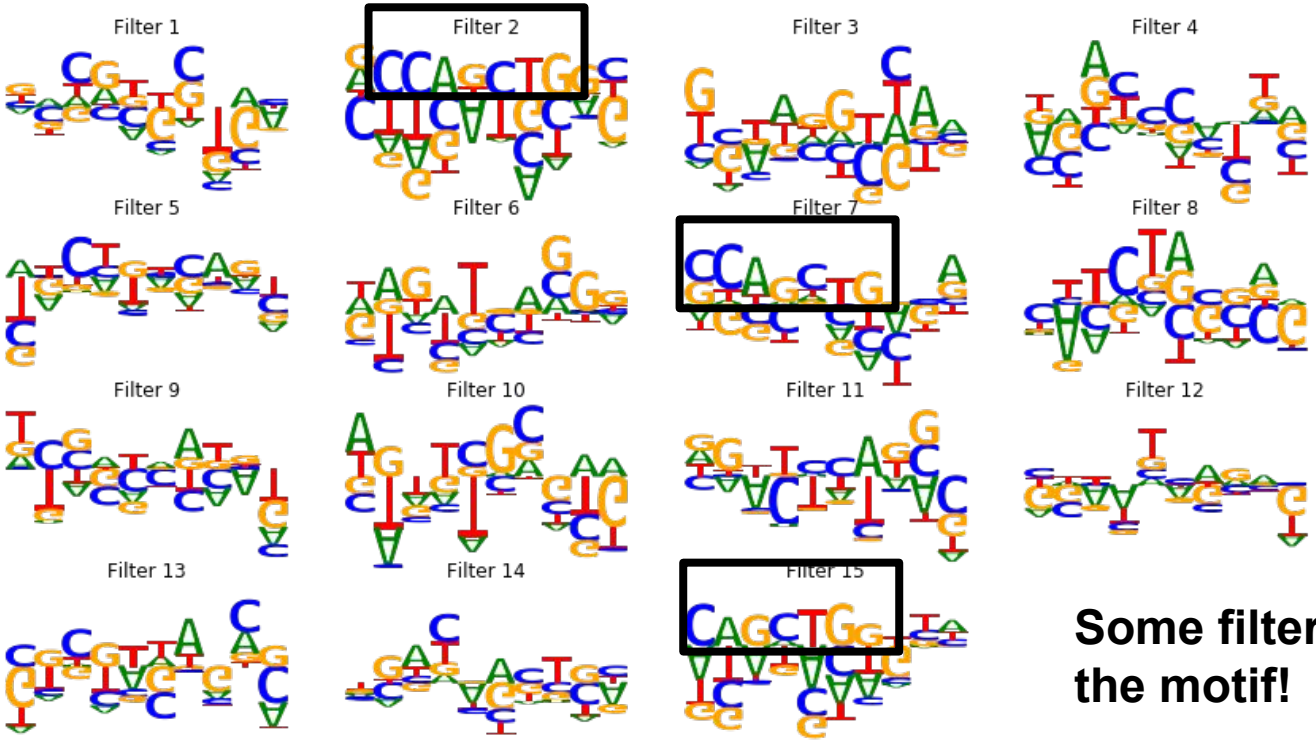


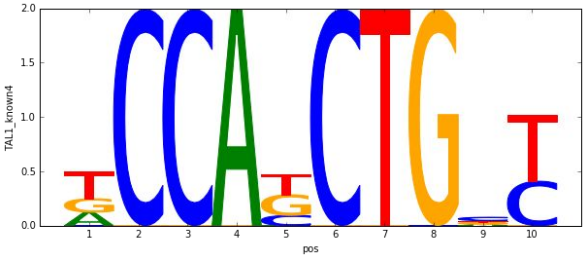
Better than one filter model?

Interpreting the Model & Data

Compare learned filters to simulated motif

interpret_SequenceDNN_filters(multi_filter_dragonn, simulation_data)



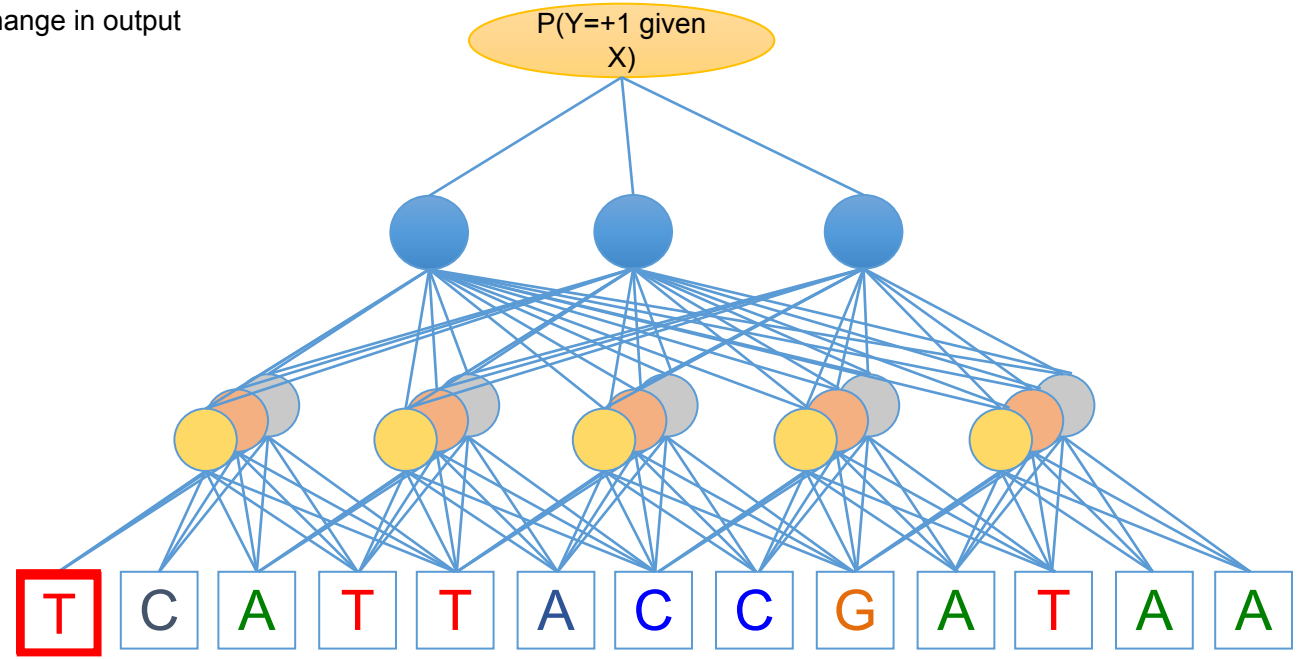


Some filters correspond to the motif!

Interpretation: In-silico mutagenesis

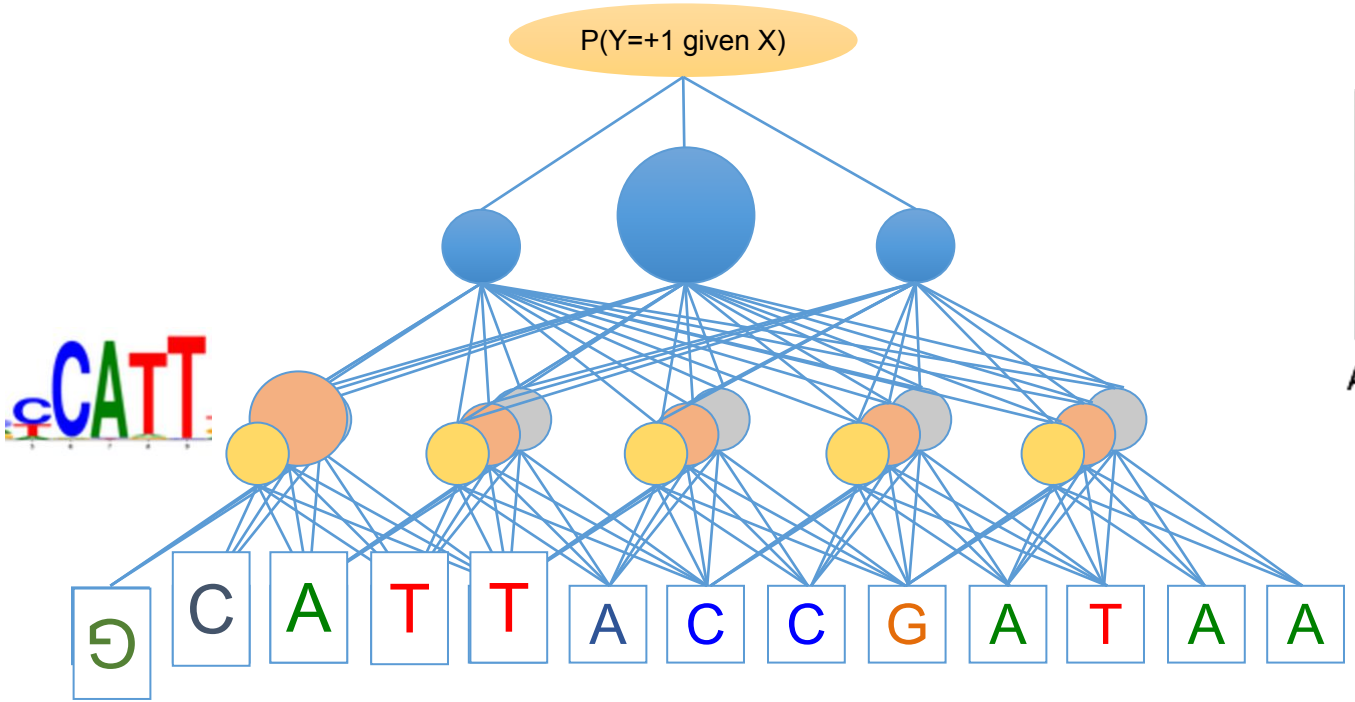
Output: Bound (+1) vs. not bound (0)

Assess change in output



Input: One-hot encoded DNA sequence

Interpretation: DeepLIFT (Deep learning feature importance)



Avanti Shrikumar

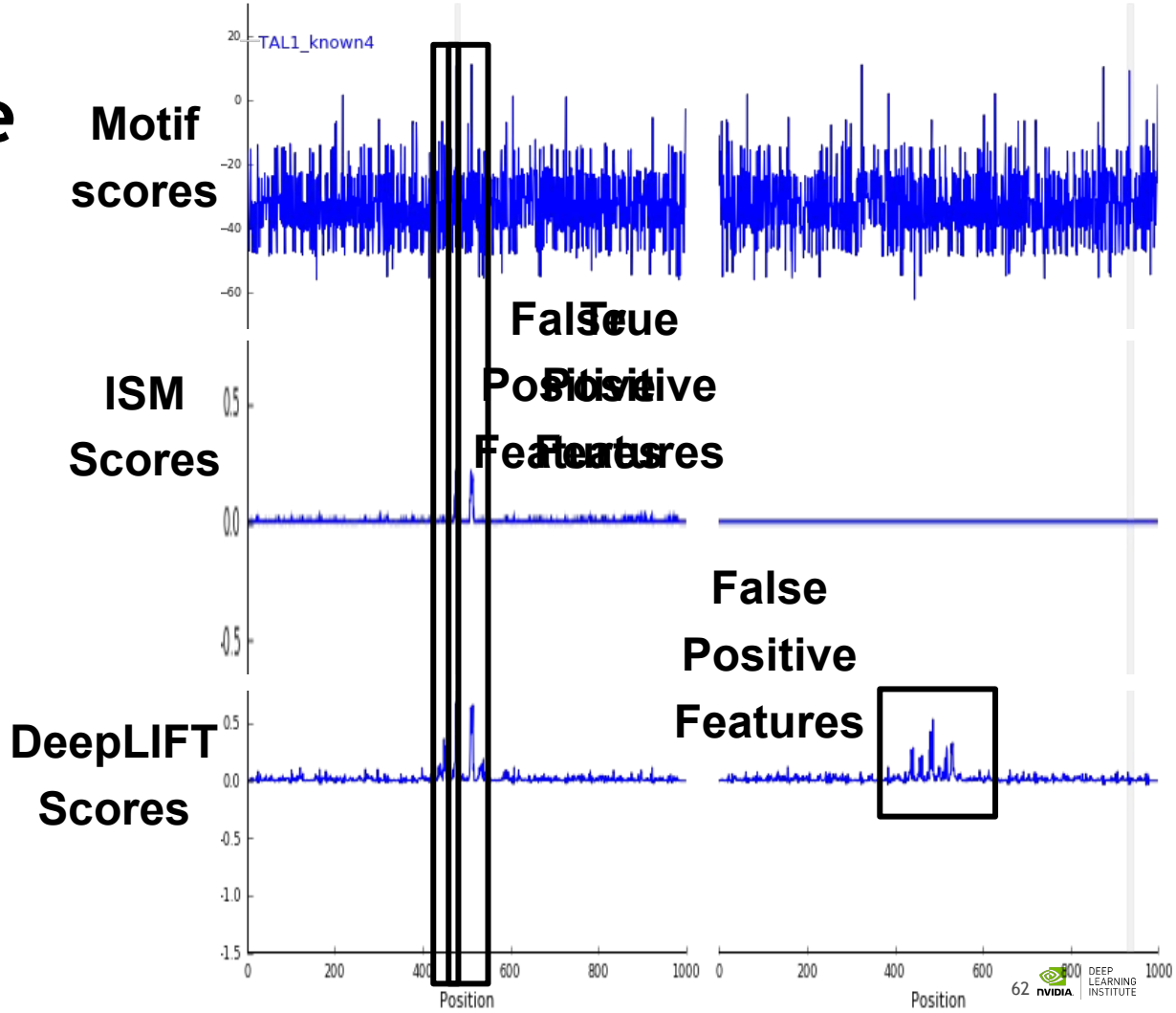
https://arxiv.org/abs/1605.01713

Compare feature importances to motif scores

interpret_data_with_SequenceDNN(
multi_filter_dragonn,
simulation_data)

ISM

DeepLIFT



Enhancement #1: multiple conv layers

ARCHITECT MULTI-LAYER DRAGONN MODEL

```
multi_layer_dragonn_parameters = {
    'seq_length': 1000,
    'num_filters': [15, 15, 15], ## notice the change to multiple filter values, one for each layer
    'conv_width': [10, 10, 10],
    'pool_width': 35}
multi_layer_dragonn = get_SequenceDNN(multi_layer_dragonn_parameters)
train_SequenceDNN(multi_layer_dragonn, simulation_data)
SequenceDNN_learning_curve(multi_layer_dragonn)
```

TRAIN MULTI-LAYER DRAGONN MODEL

Epoch 36:

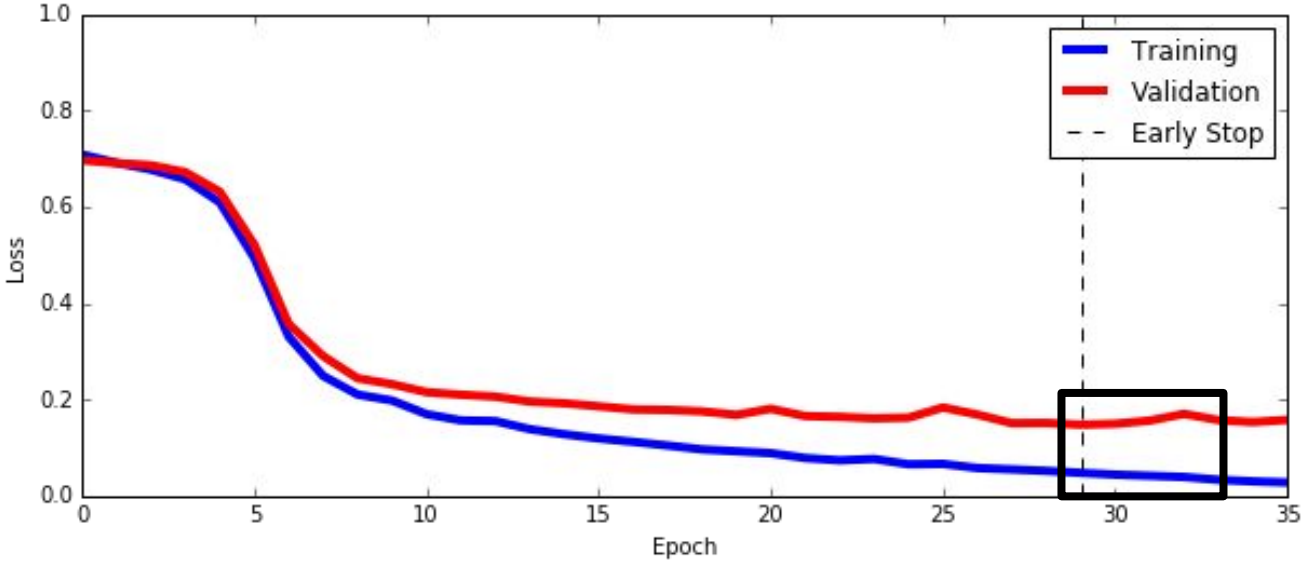
Train Loss: 0.0278 Balanced Accuracy: 99.73% auROC: 1.000 auPRC: 1.000

Recall at 5%|10%|20% FDR: 100.0%|100.0%|100.0% Num Positives: 1957 Num Negatives: 2043

Valid Loss: 0.1577 Balanced Accuracy: 94.34% **auROC: 0.987** auPRC: 0.985

Recall at 5%|10%|20% FDR: 95.6%|99.2%|99.8% Num Positives: 528 Num Negatives: 47

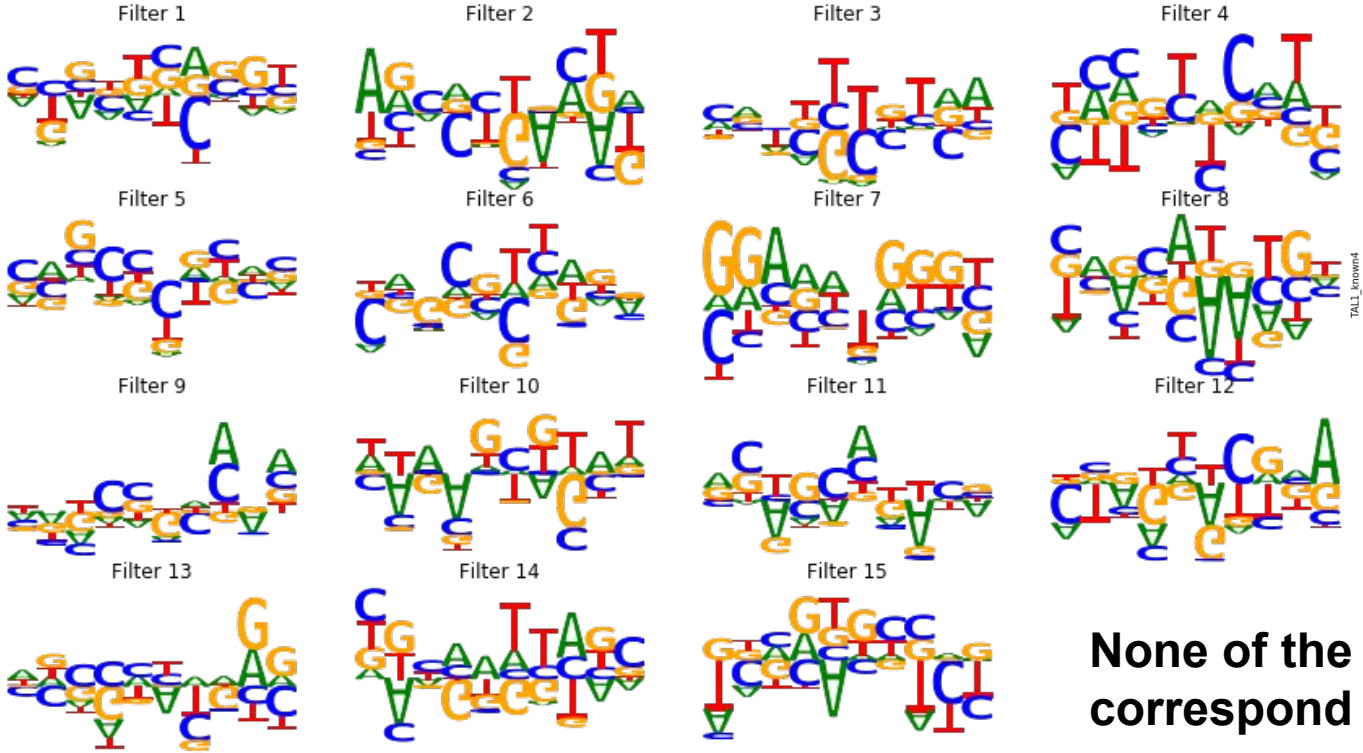
Better than one layer model

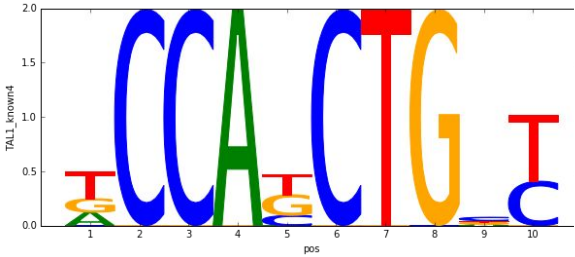


It's overfitting

Compare learned filters to simulated motif

interpret_SequenceDNN_filters(multi_layer_dragonn, simulation_data)





None of the filters correspond to the motif!

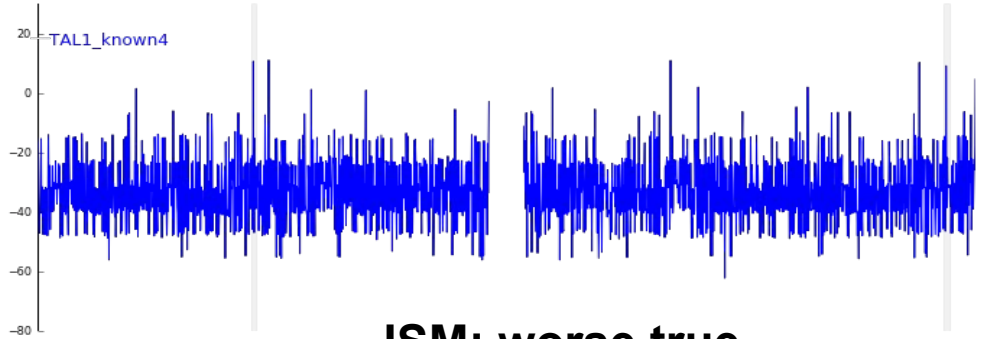
Compare feature importances to motif scores

interpret_data_with_SequenceDNN(
multi_layer_dragonn,
simulation_data)

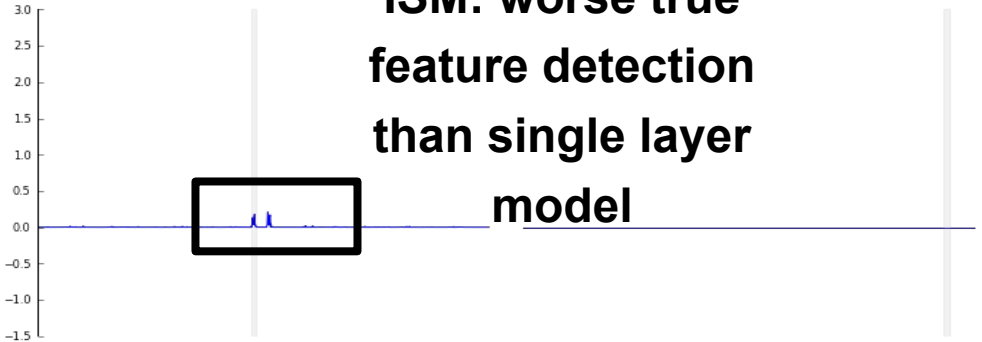
ISM

DeepLIFT

Motif scores

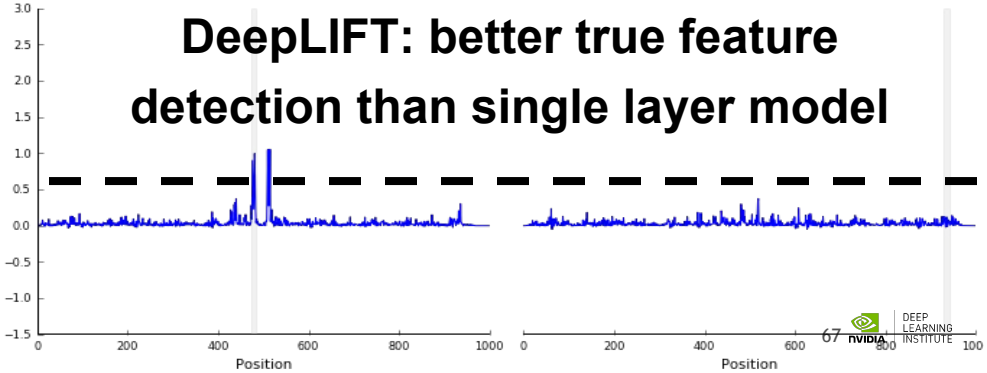


ISM Scores



ISM: worse true feature detection than single layer model

DeepLIFT Scores



DeepLIFT: better true feature detection than single layer model

Enhancement #2: dropout regularization

ARCHITECT REGULARIZED DRAGONN MODEL

```
regularized_multi_layer_dragonn_parameters = {
    'seq_length': 1000,
    'num_filters': [15, 15, 15],
    'conv_width': [10, 10, 10],
    'pool_width': 35,
    ‘dropout’: 0.2} ## we introduce dropout of 0.2 on every convolutional layer for regularization
regularized_multi_layer_dragonn = get_SequenceDNN(
        regularized_multi_layer_dragonn_parameters)
train_SequenceDNN(regularized_multi_layer_dragonn, simulation_data)
SequenceDNN_learning_curve(regularized_multi_layer_dragonn)
```

TRAIN REGULARIZED DRAGONN MODEL

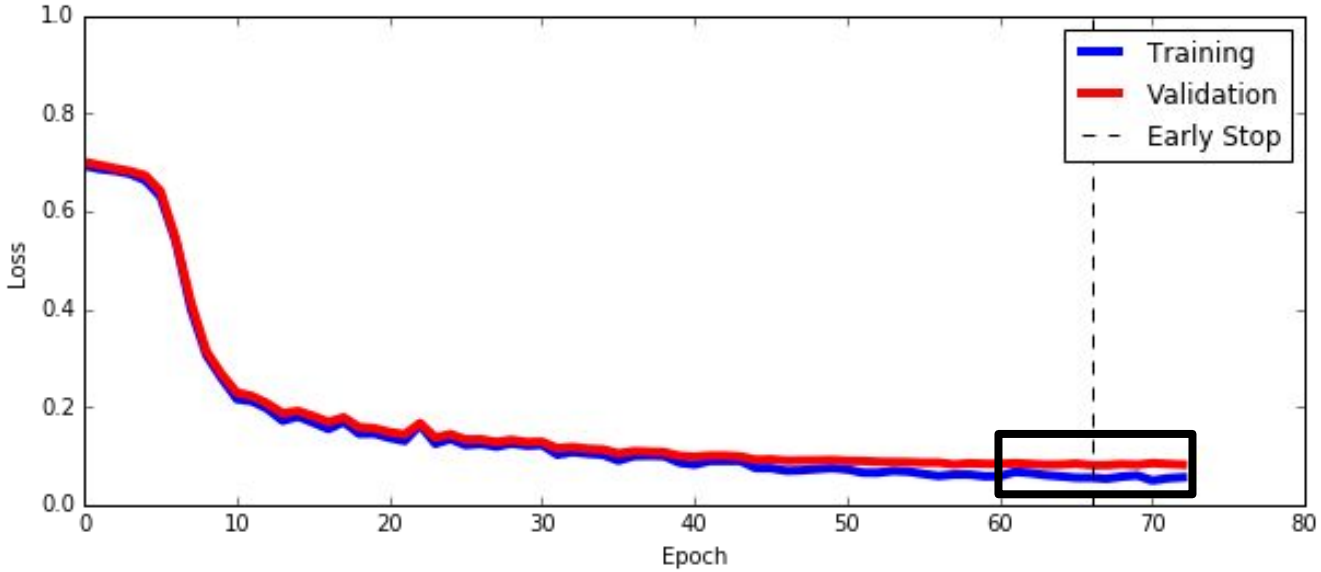
Epoch 73:

Train Loss: 0.0558 Balanced Accuracy: 98.36% auROC: 0.998 auPRC: 0.998

Recall at 5%|10%|20% FDR: 99.8%|100.0%|100.0% Num Positives: 1957 Num Negatives: 2043

Valid Loss: 0.0817 Balanced Accuracy: 97.23% **auROC: 0.994** auPRC: 0.992

Recall at 5%|10%|20% FDR: 98.5%|100.0%|100.0% Num Positives: 528 Num Negatives: 472



Better than without dropout

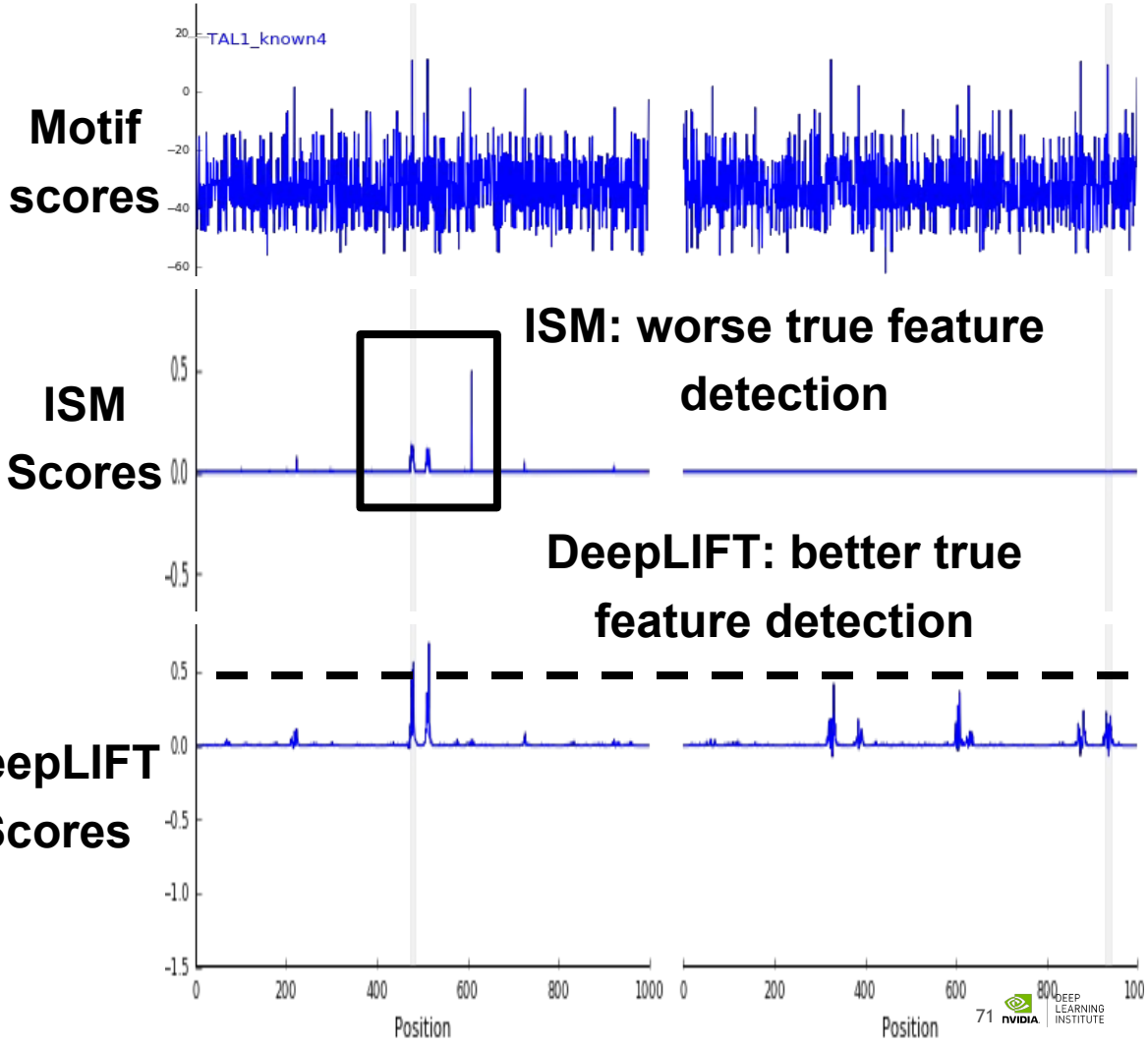
Dropout fixed the overfitting!

Compare feature importances to motif scores

interpret_data_with_SequenceDNN(
regularized_multi_layer_dragonn,
simulation_data)

DeepLIFT!

ISM!



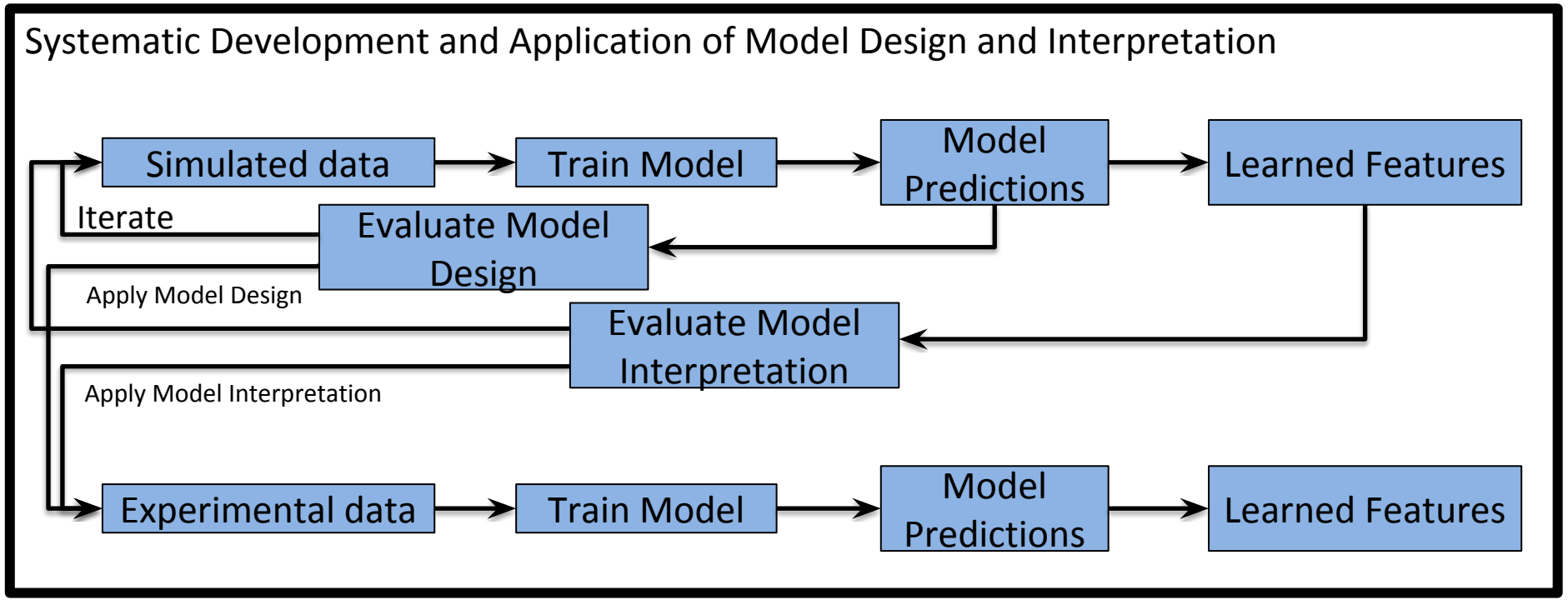
Discussion

Single layer -> multi-layer -> regularized multi layer led to

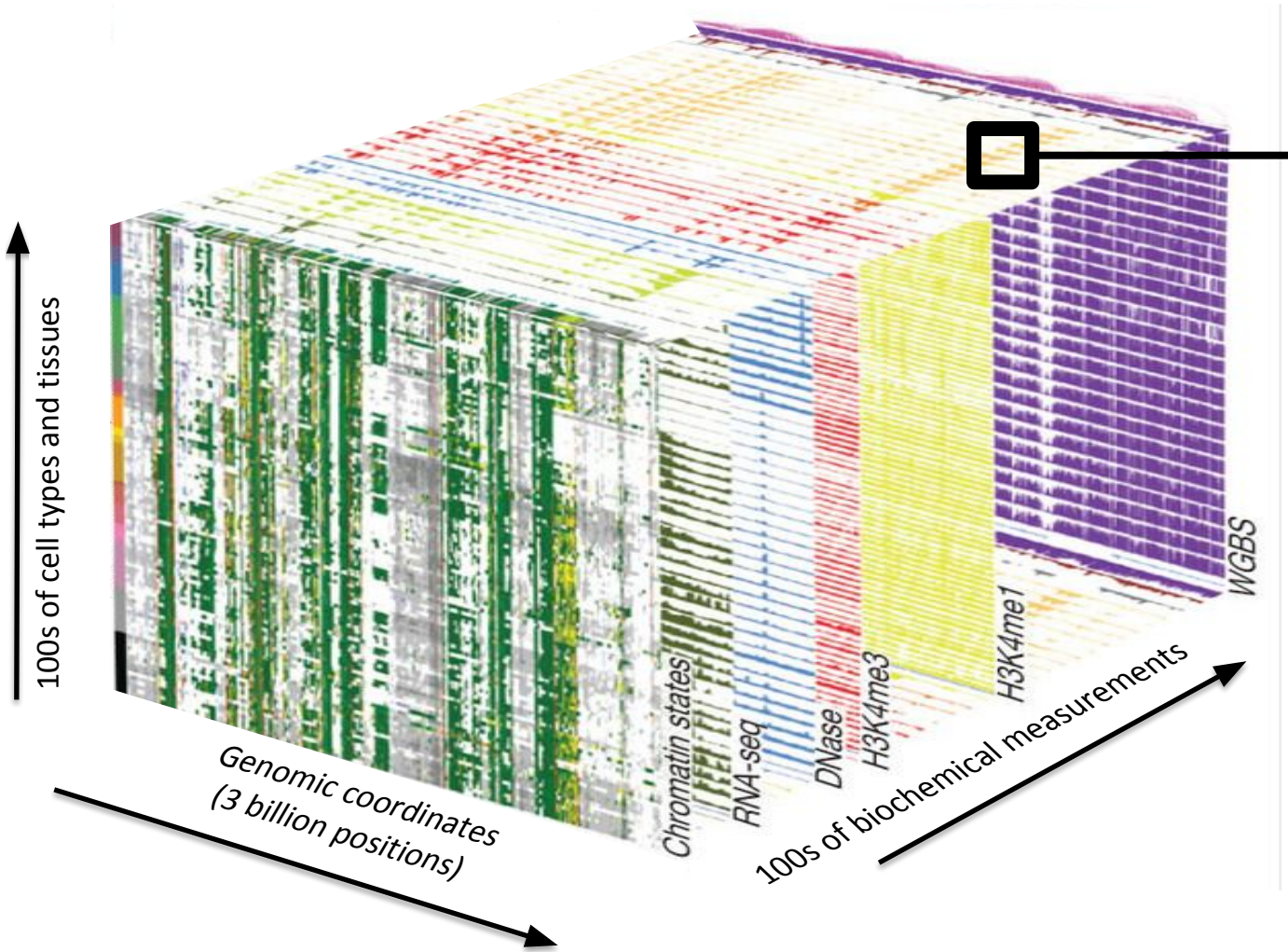
- improved auROC
- Less interpretable 1st layer filters
- Worse ISM feature importance
- Better DeepLIFT feature importance

Why? What is the expected ISM behavior in the limit of the “perfect” model of this simulation?

The DragoNN workflow

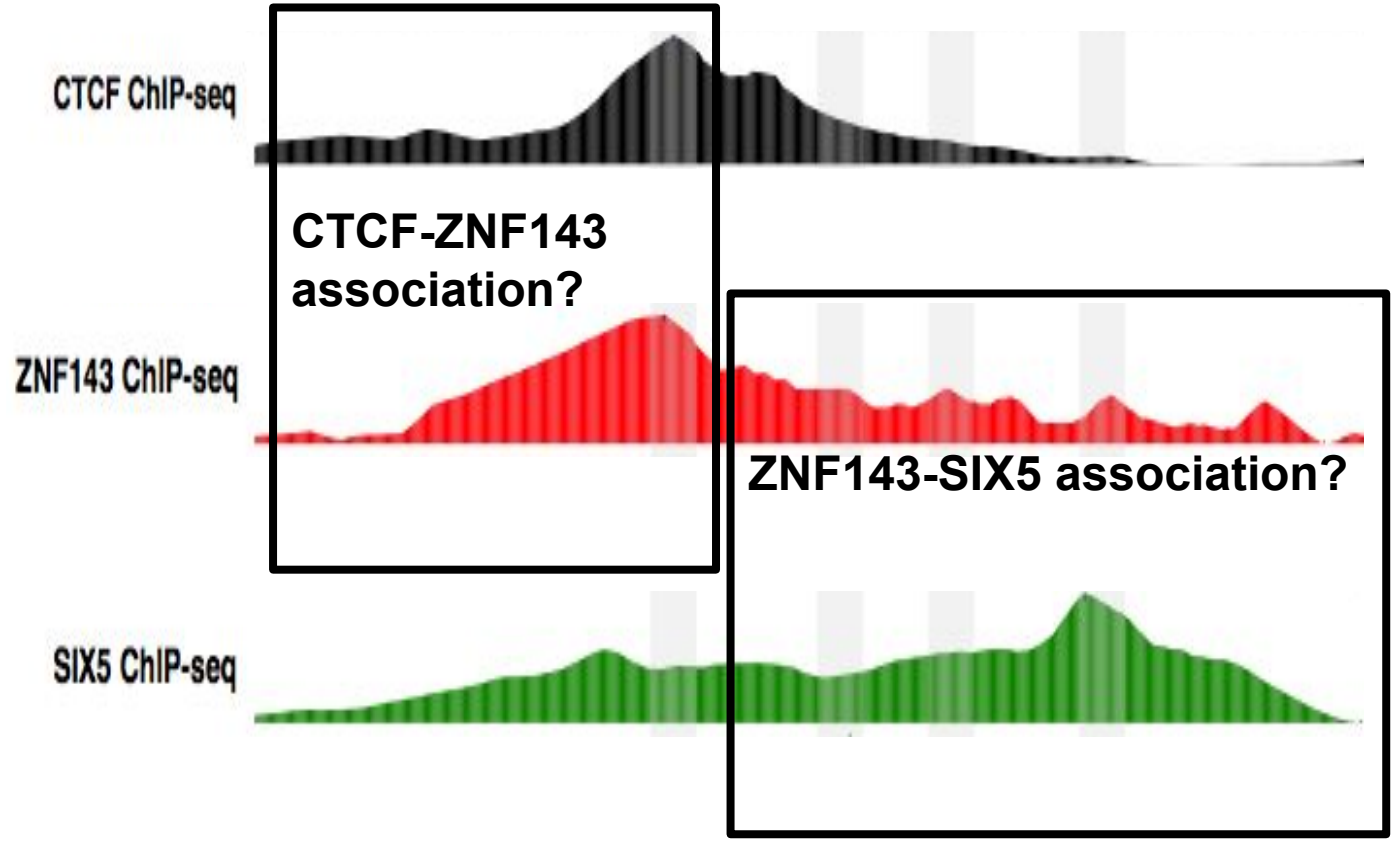


Example Application: in-vivo TF binding


100s of cell types and tissues
Genomic coordinates
(3 billion positions)
100s of biochemical measurements
Chromatin states
RNA-seq
DNase
H3K4me3
H3K4me1
WGBS

DragoNN

Example Application



Identified predictive sequence

No CTCF-SIX5 direct association?

CTCF-SIX5 association conditional on ZNF143 first discovered in:

Architecture of the human regulatory network derived from ENCODE data

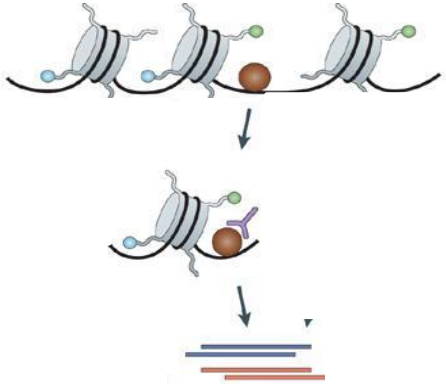
Mark B. Gerstein, Anshul Kundaje et al.

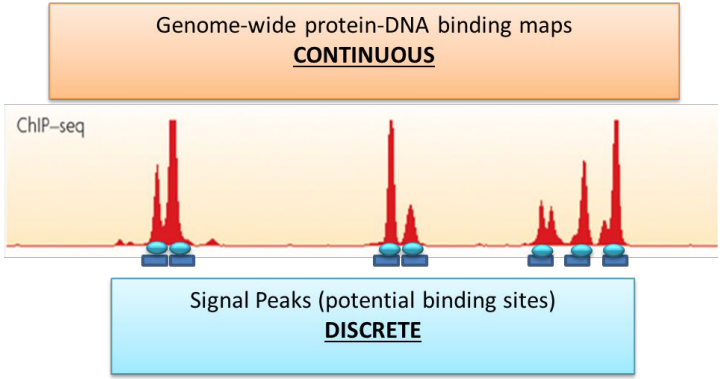
Nature **489**, 91–100 (06 September 2012) | doi:10.1038/nature11245

Received 09 December 2011 | Accepted 22 May 2012 | Published online 05 September 2012

State of the Field: TF Binding Models

Challenge Data: “Ground truth”





Genome-wide in-vivo TF binding data

- **Ground truth:** High quality ChIP-seq data targeting specific TF in specific cell type
- **Peaks**: Regions of enrichment => likely binding events. Peaks ranked by enrichment and reproducibility across replicates

Challenge Data: Output variable

Binary Classification problem

- Is a genomic region bound or not by TF of interest?

- Two types of TF ChIP-seq peaks called
  - **Conservative:** High-confidence, reproducible peaks labeled as **bound (B)**
  - **Relaxed:** Lower confidence, reproducible peaks labeled as **ambiguous (A)** – not used in evaluation
- Rest of the genome is labeled **unbound (U)**
- Resolution of data: 200 bp bins every 50 bp

chr	start	stop	GM12878	H1-hESC
chr10	600	800	B	B
chr10	650	850	U	U
chr10	700	900	U	U
chr10	750	950	U	U
chr10	800	1000	B	U
chr10	850	1050	U	U
chr10	900	1100	U	U
chr10	950	1150	A	B
chr10	1000	1200	U	U

Challenge Data: Problem set up

Held-out cell types

For each TF

- ChIP-seq data is provided for some **training cell types** for participants to train models
- Binding to be predicted in new **held-out cell types**

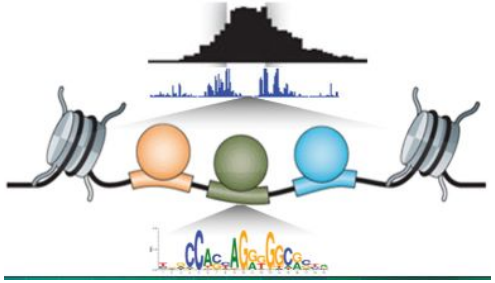
Held-out chromosomes

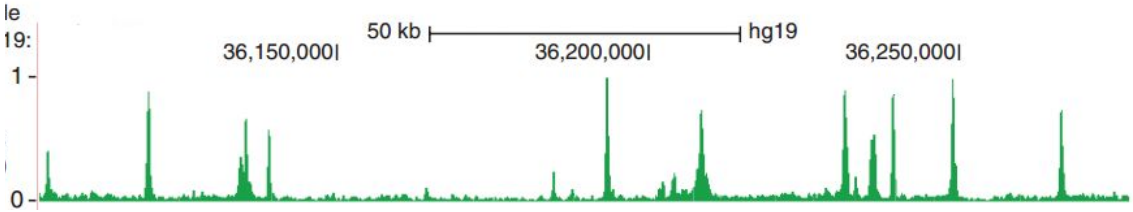
- Binding data in **chr1, 21 and 8** is never provided (even in training cell types)
- Performance evaluated on these 3 held-out chromosomes

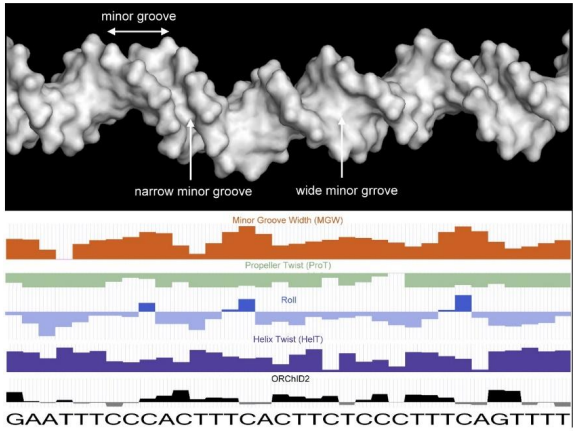
32 diverse TFs, 14 cell types (Training and held-out cell types are different for each TF)

Challenge Data: Predictor variables

1. **DNA Sequence features**
2. **Chromatin accessibility**
3. **Gene expression**
4. **Gene annotations**
5. **DNA shape features**







(*Zhou et al. 2013 Nucleic Acids Res., 41, W56-W62.*)

Challenge Questions

Binary prediction task: Predict probability (between 0 and 1) of being bound by TF for each 200 bp bin every 50 bp in specified chromosomes in specified cell type

Chromosome	Start Position	End Position	Predicted Probability
chr1	600	800	2.16e-39
chr1	650	850	5.65e-169
chr1	700	900	6.18e-37
chr1	750	950	2.70e-26
chr1	800	1000	6.40e-74

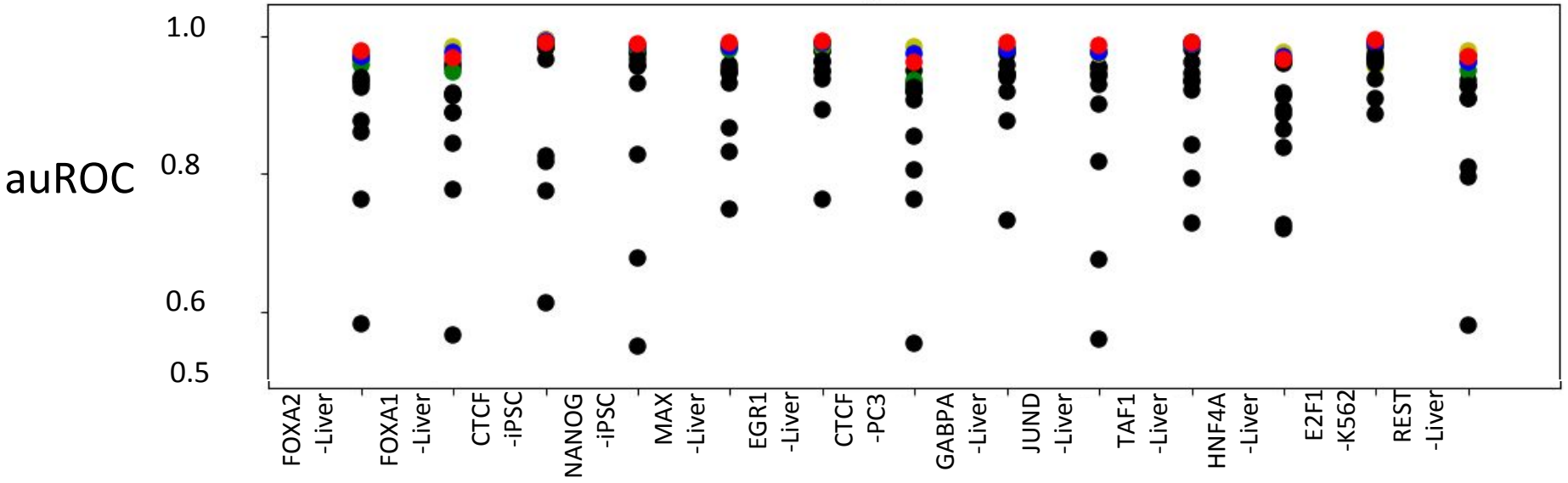
Main Challenge Question

Across cell-type prediction

Predict binding in hidden cell types on hidden chromosomes

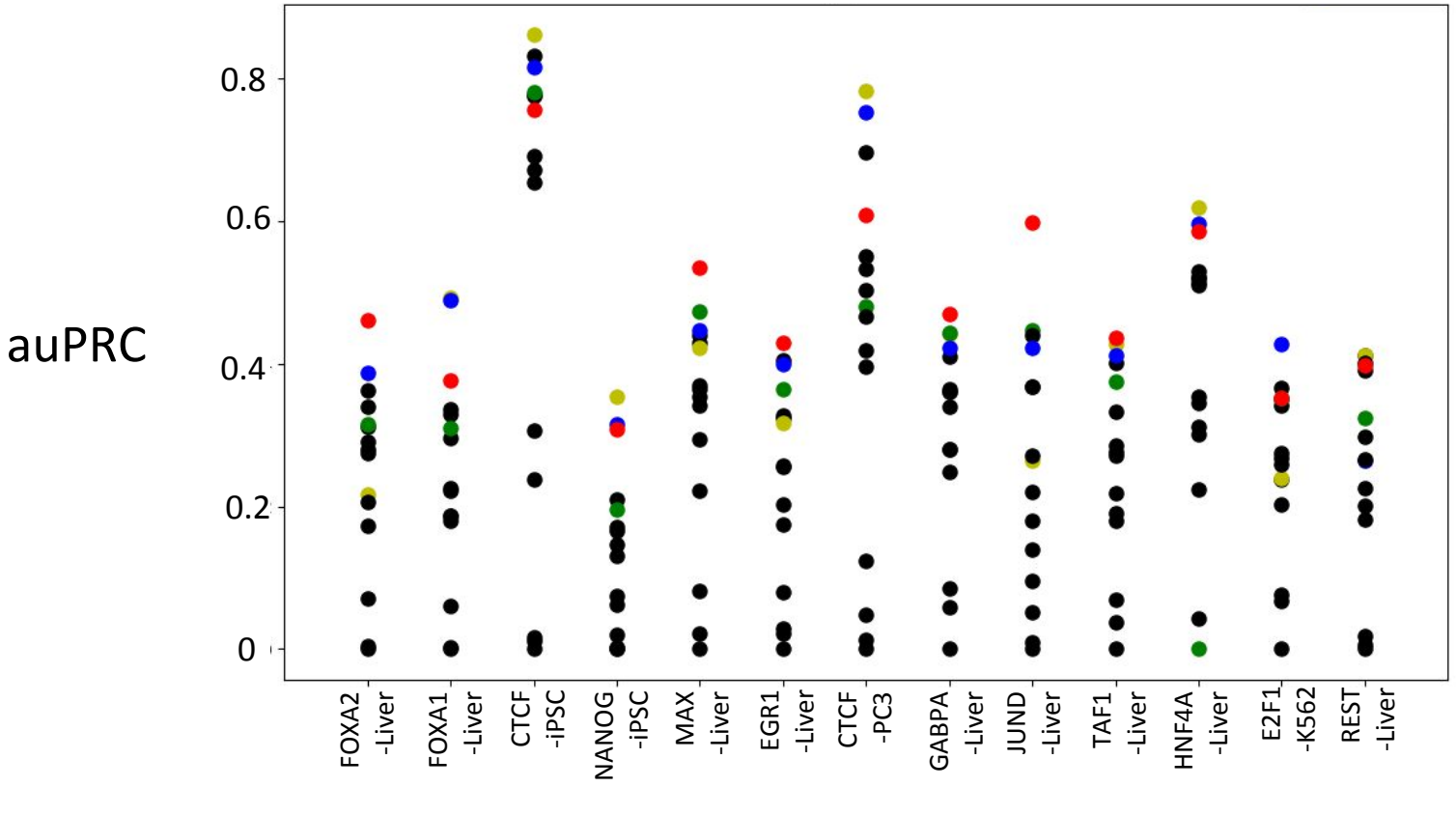
(Practical definition of “prediction”)

auROC across all TF/cell type



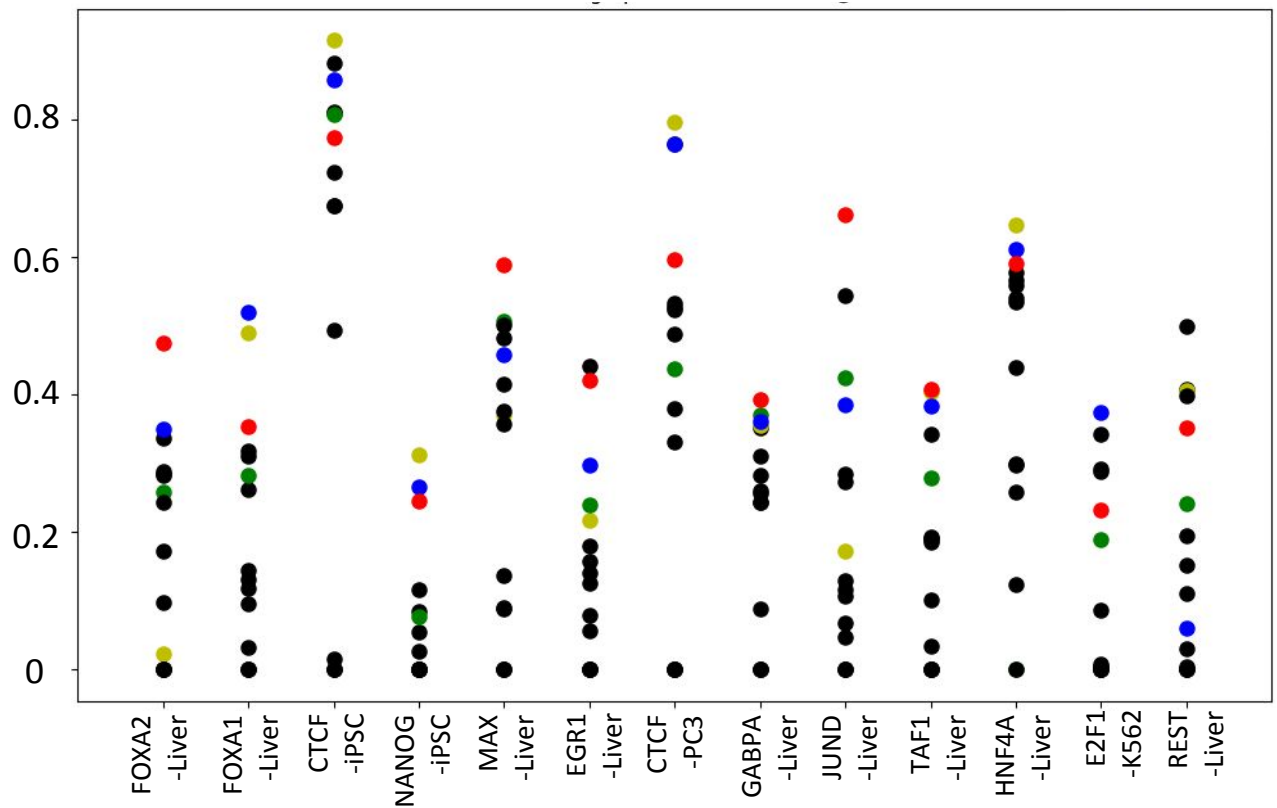
- Daniel Quang
- Autosome.ru
- J-Team
- Yuanfang Guan

auPRC across all TF/cell type



Recall @ 50%FDR across all TF/cell type

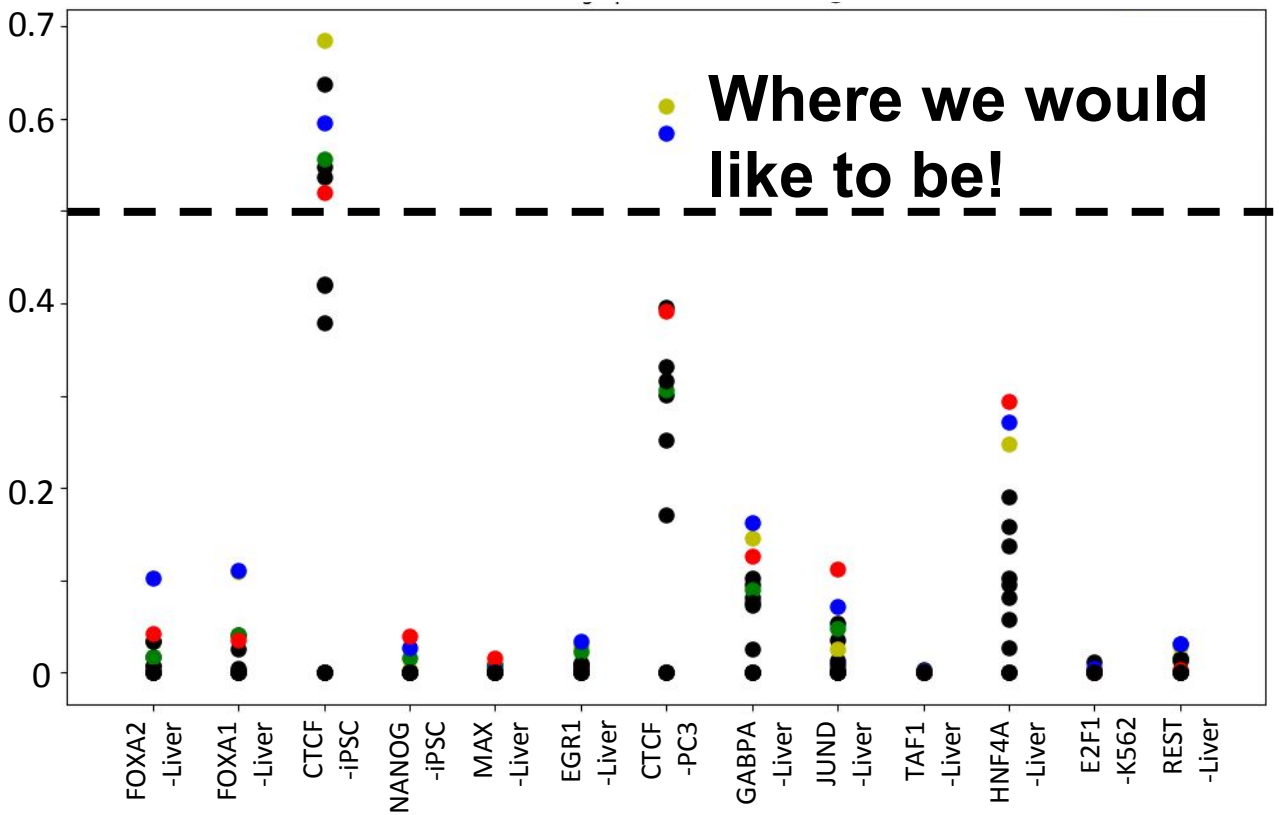
Recall @ 50% FDR



- Daniel Quang
- Autosome.ru
- J-Team
- Yuanfang Guan

Recall @ 10%FDR across all TF/cell type

Recall @ 10% FDR



- Daniel Quang
- Autosome.ru
- J-Team
- Yuanfang Guan

WHAT'S NEXT

WHAT'S NEXT

- Access tutorial on github: https://github.com/kundajelab/dragonn/blob/GTC/examples/GTC_workshop_tutorial.ipynb
- Follow DragoNN repo for discussions and more tutorials: https://github.com/kundajelab/dragonn/
- Reach out to NVIDIA and the Deep Learning Institute
- Reach out to Johnny Israeli and Anshul Kundaje

WHAT'S NEXT

TAKE SURVEY

…for the chance to win an NVIDIA SHIELD TV.

Check your email for a link.

ACCESS ONLINE LABS

Check your email for details to access more DLI training online.

ATTEND WORKSHOP

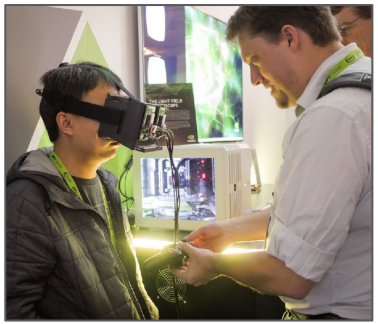
Visit www.nvidia.com/dli for workshops in your area.

JOIN DEVELOPER PROGRAM

Visit https://developer.nvidia.com/join for more.

GPU TECHNOLOGY CONFERENCE

May 8 - 11, 2017 | Silicon Valley | #GTC17
www.gputechconf.com

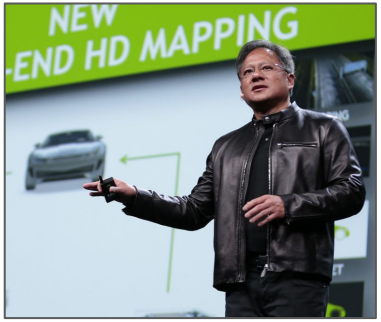


CONNECT

Connect with technology experts from NVIDIA and other leading organizations

LEARN

Gain insight and valuable hands-on training through hundreds of sessions and research posters



DISCOVER

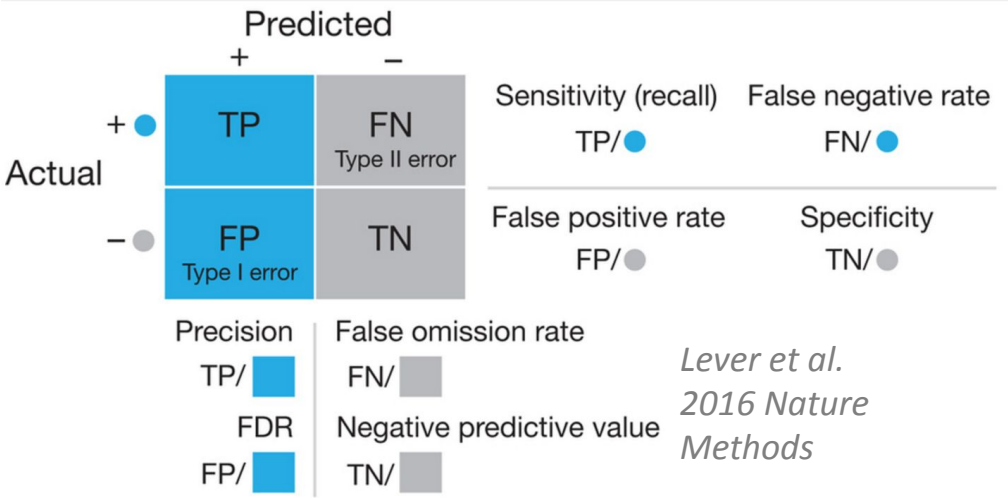
See how GPUs are creating amazing breakthroughs in important fields such as deep learning and AI

INNOVATE

Hear about disruptive innovations from startups

backup slides

Performance Measures



Lever et al. 2016 Nature Methods

- **Area under Receiver Operating Curve (auROC):** The area under the curve of Recall vs. False positive rate.
- **Area under Precision-Recall Curve (auPRC):** The area under the curve of Precision (1- False discovery rate) vs. Recall.
- **Recall @ X% FDR (10%, 50%):** The fraction of actual bound sites predicted as positives when controlling the False Discovery Rate (1-Precision) to be under X%.