A long time ago, in a room in the basement of EE...

Representation learning of genomic sequence motifs with convolutional neural networks

Peter K. Koo and Sean R. Eddy

CompBio Faculty Candidate, same time

CompBio Seminar
February 10, 2020
Alyssa LaFleur and Erin Wilson

Representation learning of genomic sequence motifs with convolutional neural networks

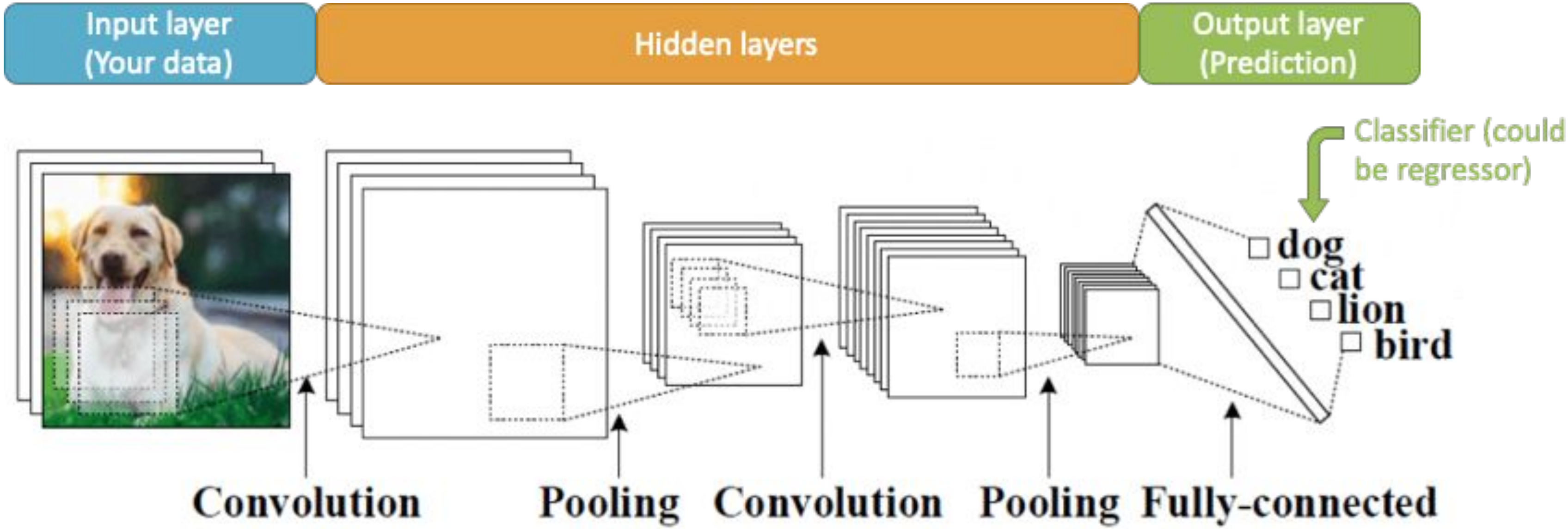
Peter K. Koo and Sean R. Eddy

CompBio Seminar
March 366, 2020
Erin Wilson
(slides co-created with Alyssa!)

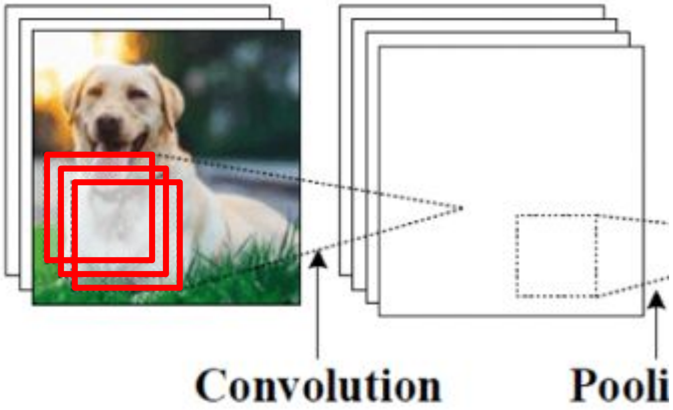
Overview

- Background
  - How do CNNs work?
  - Why do people care about finding motifs?
- Methods
  - Synthetic dataset used in this paper
  - Experimental setup for CNN architectures
  - Model vs Motif evaluation metrics
- Results
  - Pulling various CNN architecture levers!
- Main takeaways & discussion

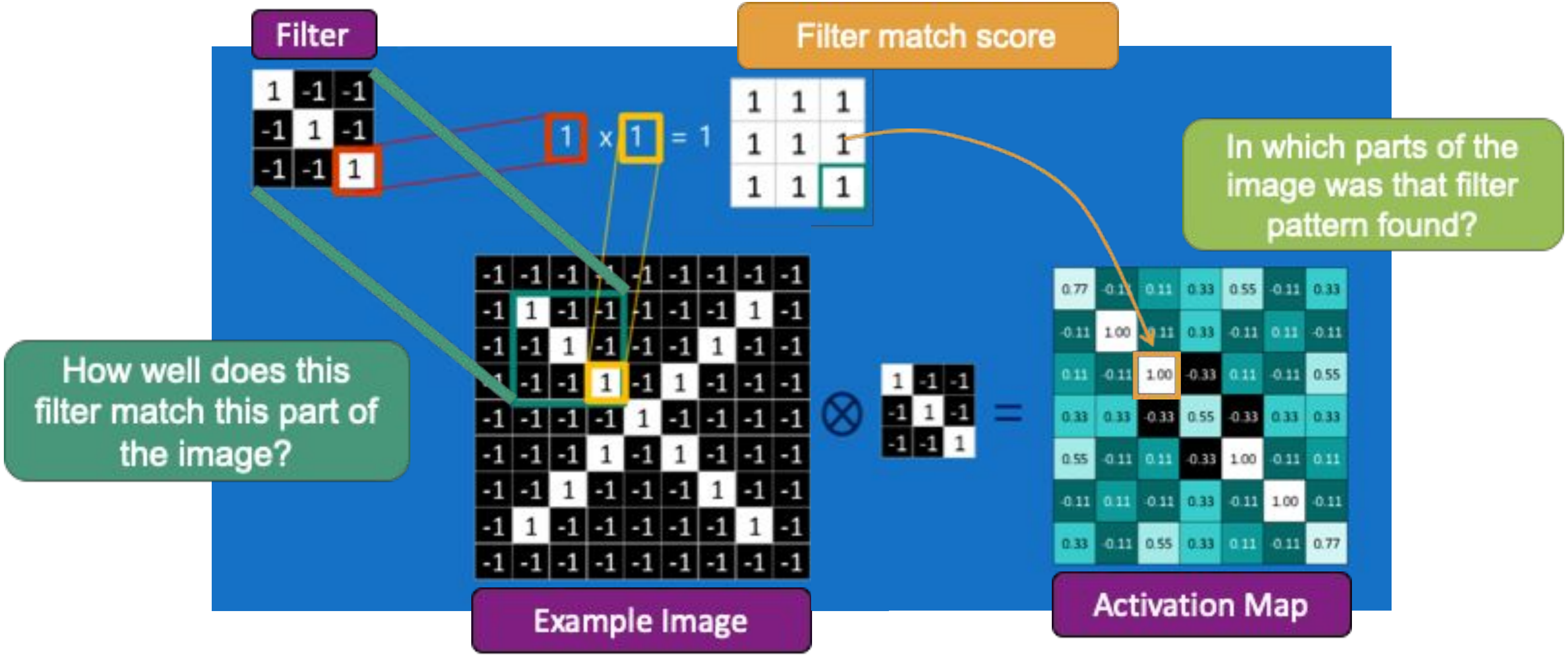
A shallow dive into deep learning…



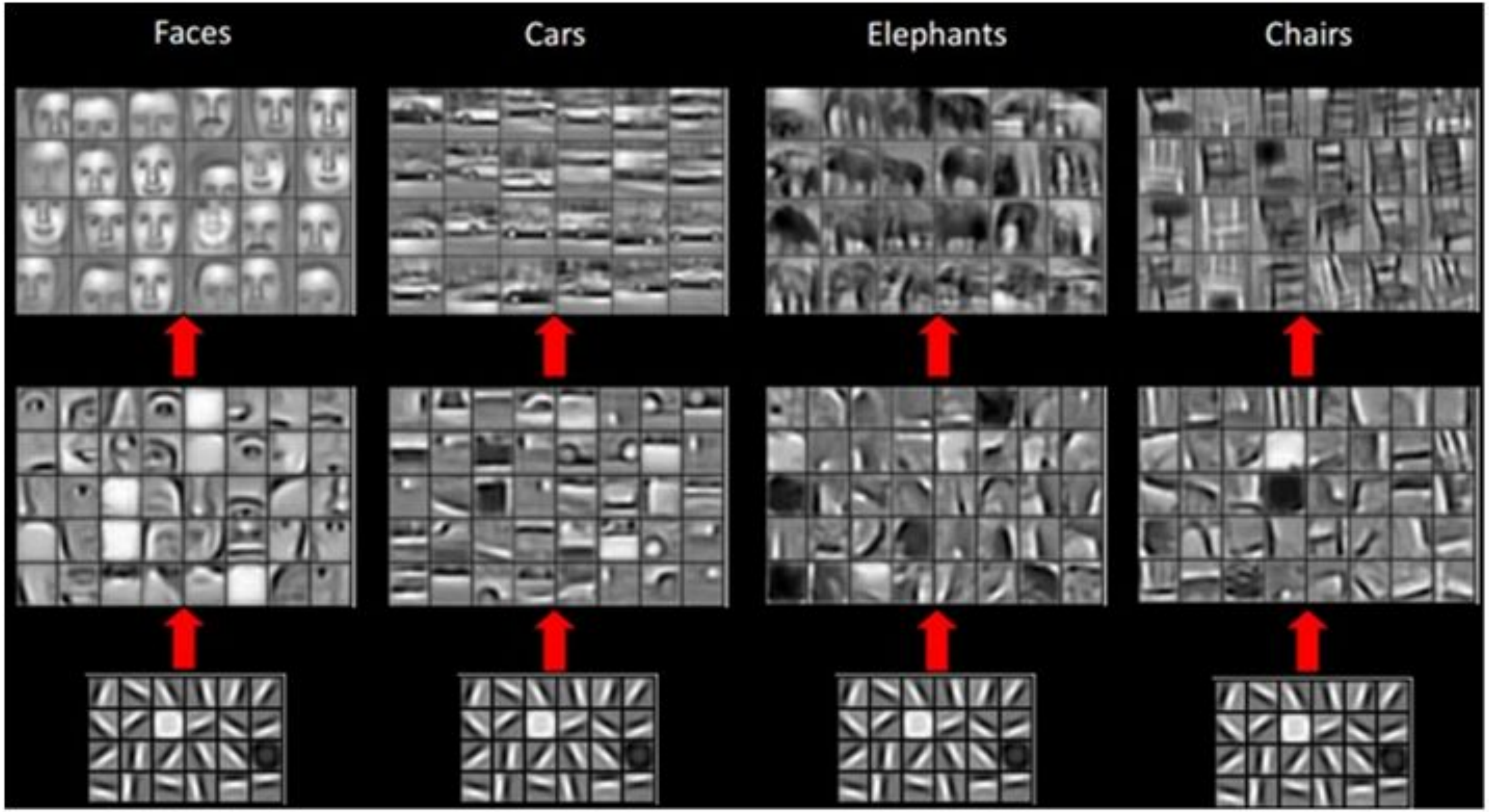
CNNs capture local spatial information between pixels in an image



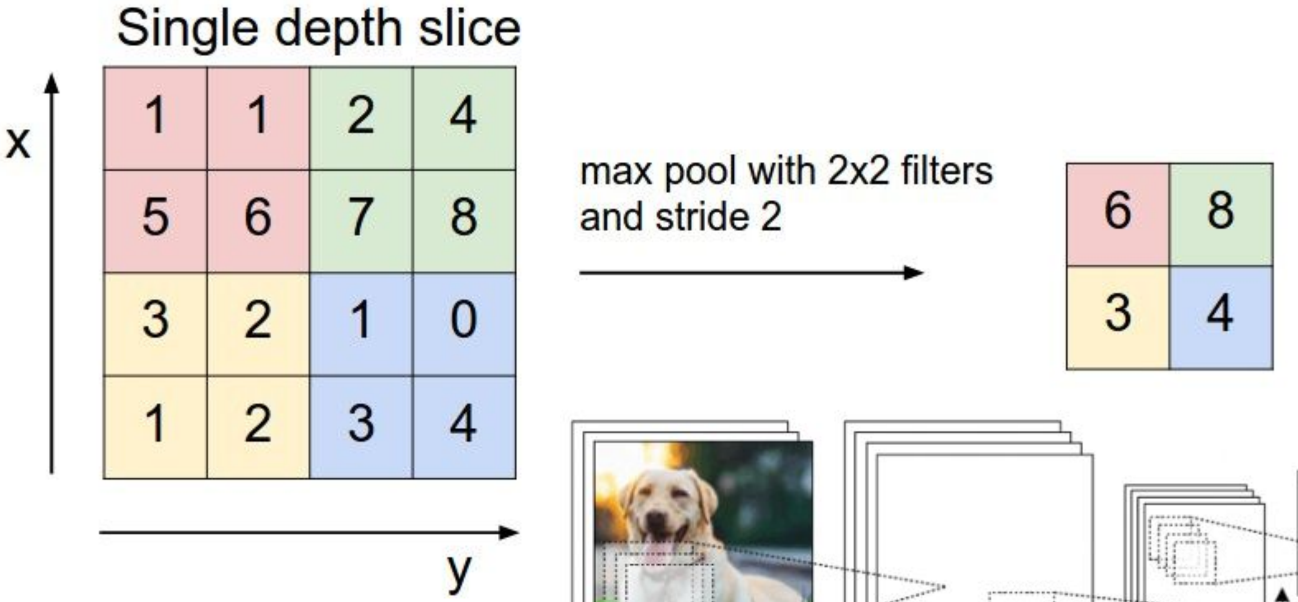
Filters are like small patterns. You can identify areas of the image containing that pattern.

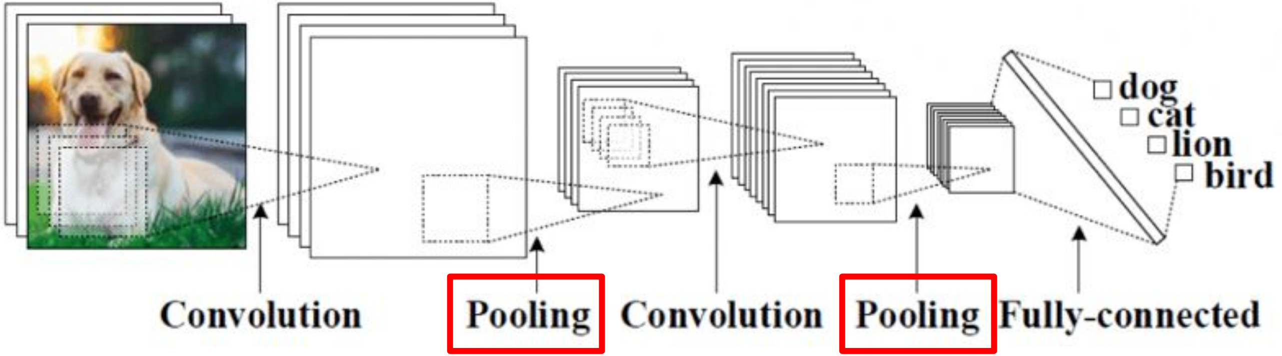


Filters learn basic patterns that can be composed into more complex features

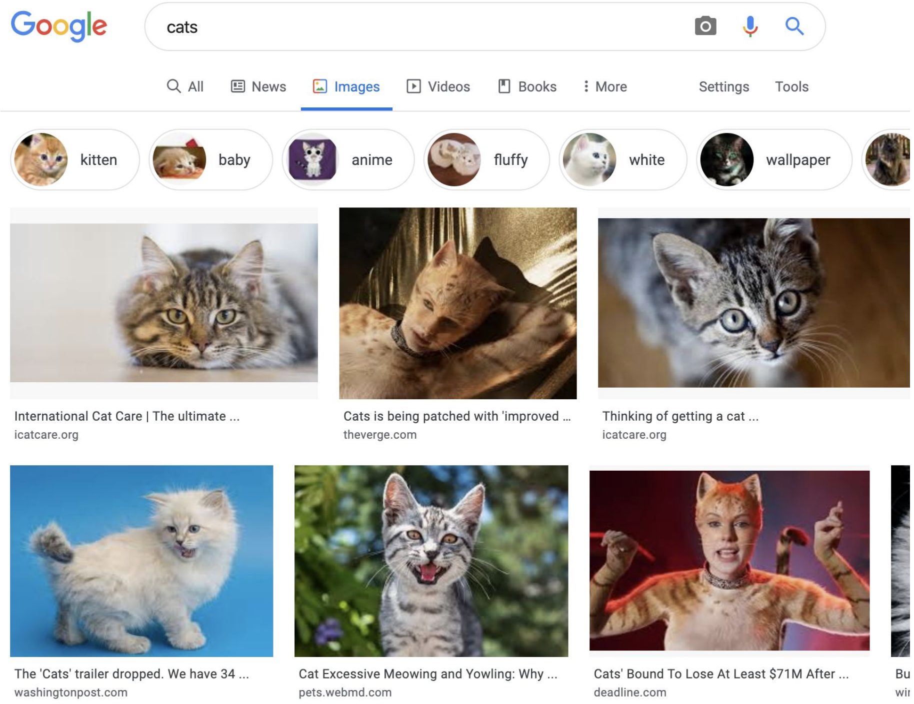


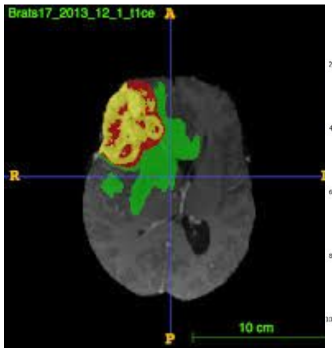
Max pooling: reduce image features by taking the max value from a window





How are CNNs helpful in biology?



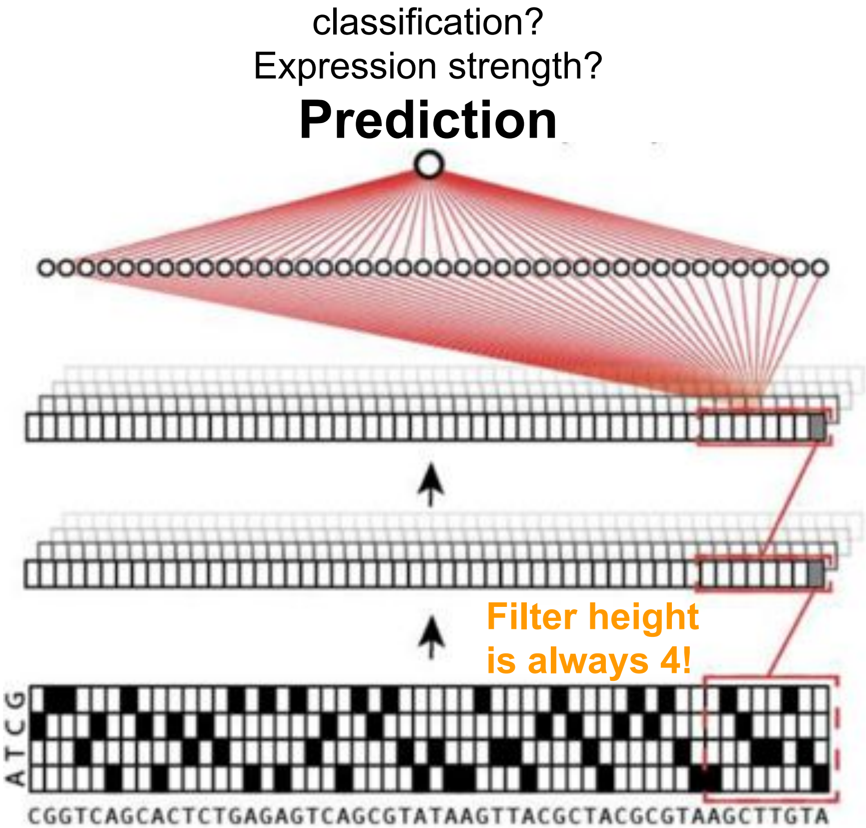


How to pretend your DNA is a cat.

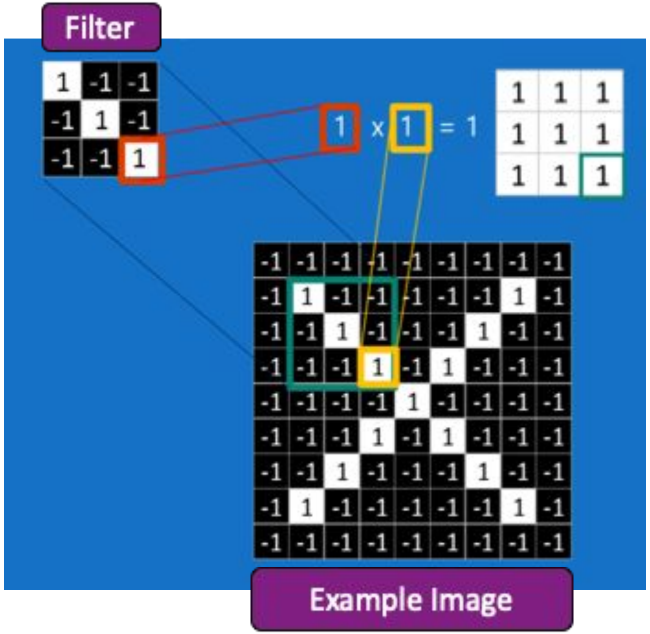
A	[1,0,0,0]
C	[0,1,0,0]
G	[0,0,1,0]
T	[0,0,0,1]

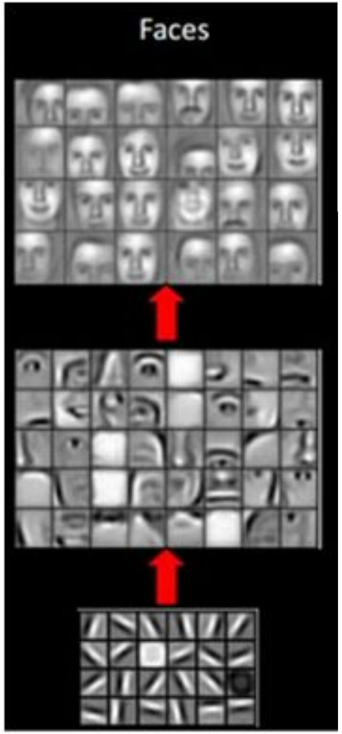
A	T	G	G	C	T	C	A	T
1	0	0	0	0	0	0	1	0
0	0	0	0	1	0	1	0	0
0	0	1	1	0	0	0	0	0
0	1	0	0	0	1	0	0	1

C A T



What do filters learn?





A	0.0	1.0	0.0	1.0	1.0	0.0
C	0.0	0.0	0.0	0.0	0.0	0.0
G	0.0	0.0	0.0	0.0	0.0	0.0
T	1.0	0.0	1.0	0.0	0.0	1.0
	T	A	T	A	A	T

A	0.1	0.7	0.2	0.4	0.4	0.1
C	0.0	0.0	0.1	0.2	0.2	0.0
G	0.1	0.1	0.1	0.2	0.2	0.0
T	0.8	0.2	0.6	0.2	0.2	0.9
	T	A	T	A	A	T

CNN filters can learn motifs relevant to the prediction task

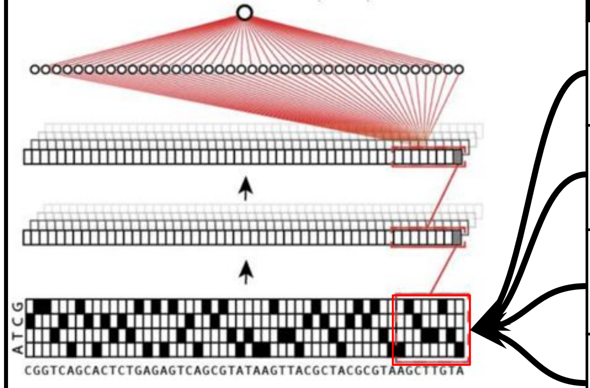
How do upstream sequences influence gene expression strength?

Learned filters:

Predicts high expression!

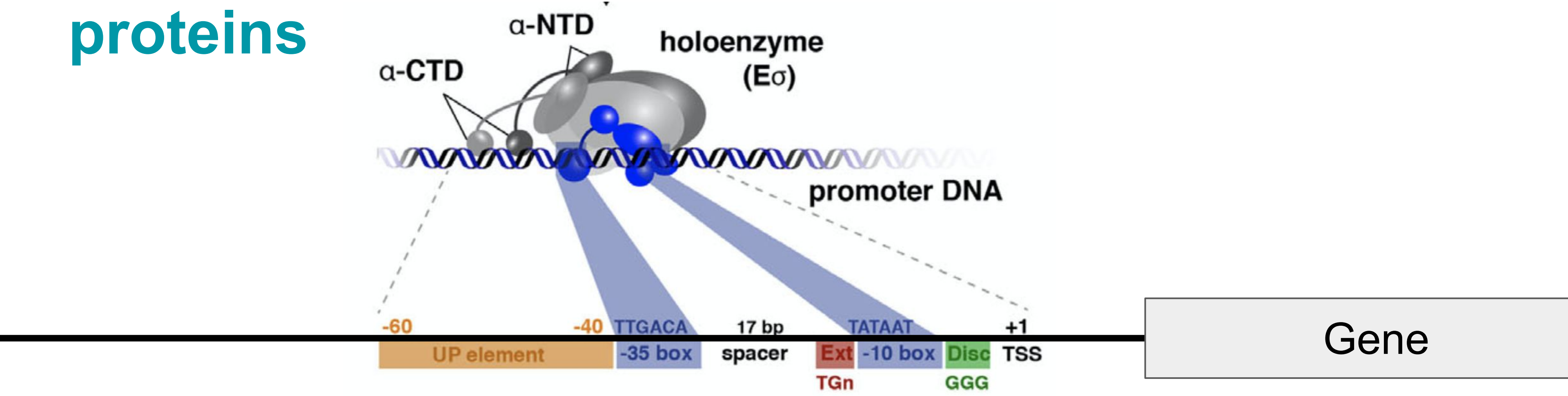
Predicts low expression!

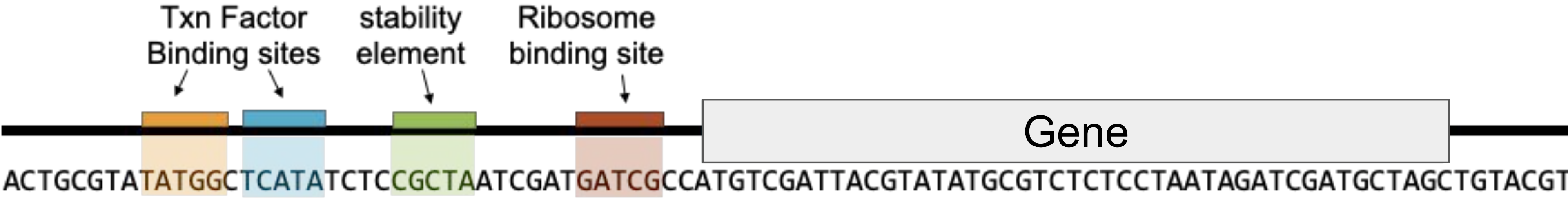
Expression strength



Upstream sequence	Expression strength
ATGGCTCATATCTCCG...	204
TATCTCCGCTAATCGA...	50
CTAATCGAACATCGCA...	3
CATCGCATGTCGATTA...	186

Motifs are landing zones for various DNA binding proteins





JASPAR CORE

Total 1964 profiles

http://jaspar.genereg.net/

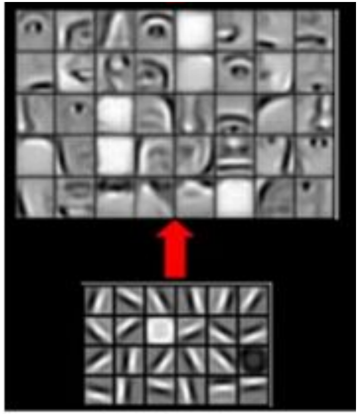
Display 10 profiles

Filter:

	ID	Name	Species	Class	Family	Sequence logo
	MA0001.1	AGL3	Arabidopsis thaliana	MADS box factors	MADS	
	MA0001.2	AGL3	Arabidopsis thaliana	MADS box factors		
	MA0002.1	RUNX1	Homo sapiens	Runt domain factors	Runt-related factors	
	MA0002.2	RUNX1	Mus musculus	Runt domain factors	Runt-related factors	
	MA0003.1	TFAP2A	Homo sapiens	Basic helix-span-helix factors (bHSH)	AP-2	
	MA0003.2	TFAP2A	Homo sapiens	Basic helix-span-helix factors (bHSH)	AP-2	

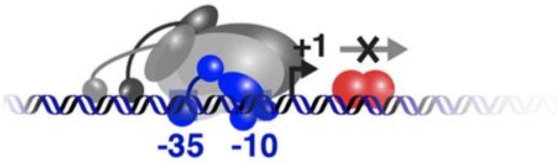
Main takeaways:

1.) CNN filters are good at finding small areas of patterns within a bigger pattern that are useful for prediction tasks



2.) For DNA sequence inputs, CNN filters learn DNA motifs

3.) Motifs usually contain some biological relevance for how, when, and where proteins bind to DNA

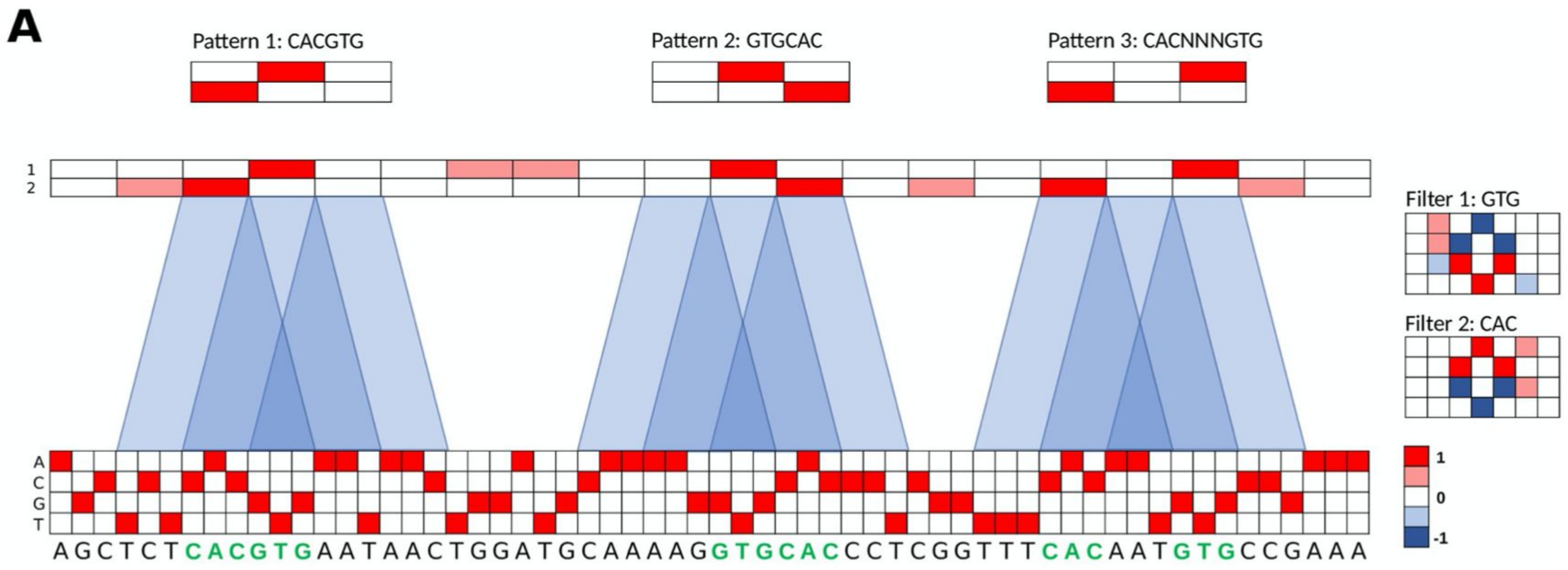


Representation learning of genomic sequence motifs with convolutional neural networks

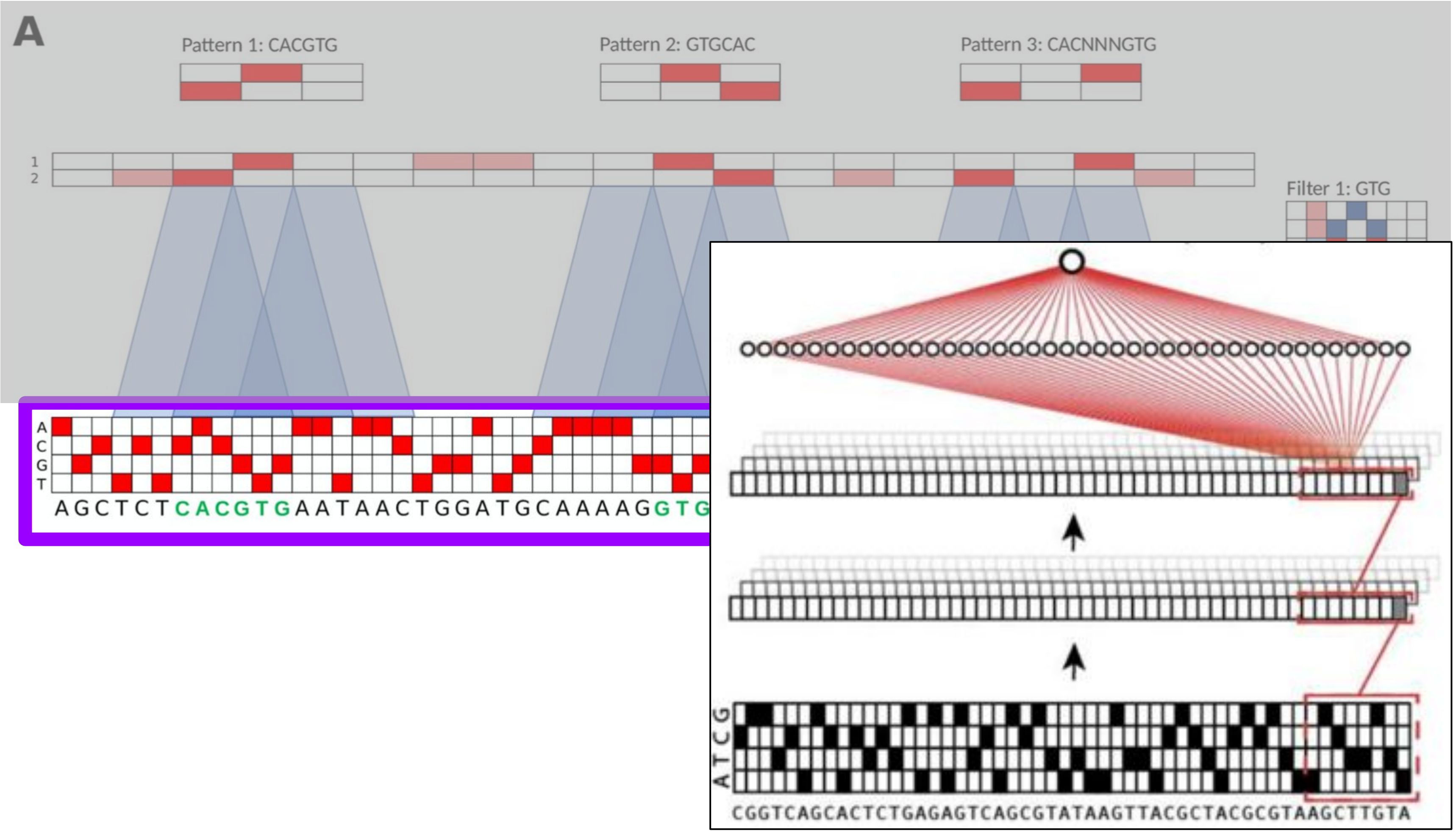
Peter K. Koo[1¤*], Sean R. Eddy[1,2*]

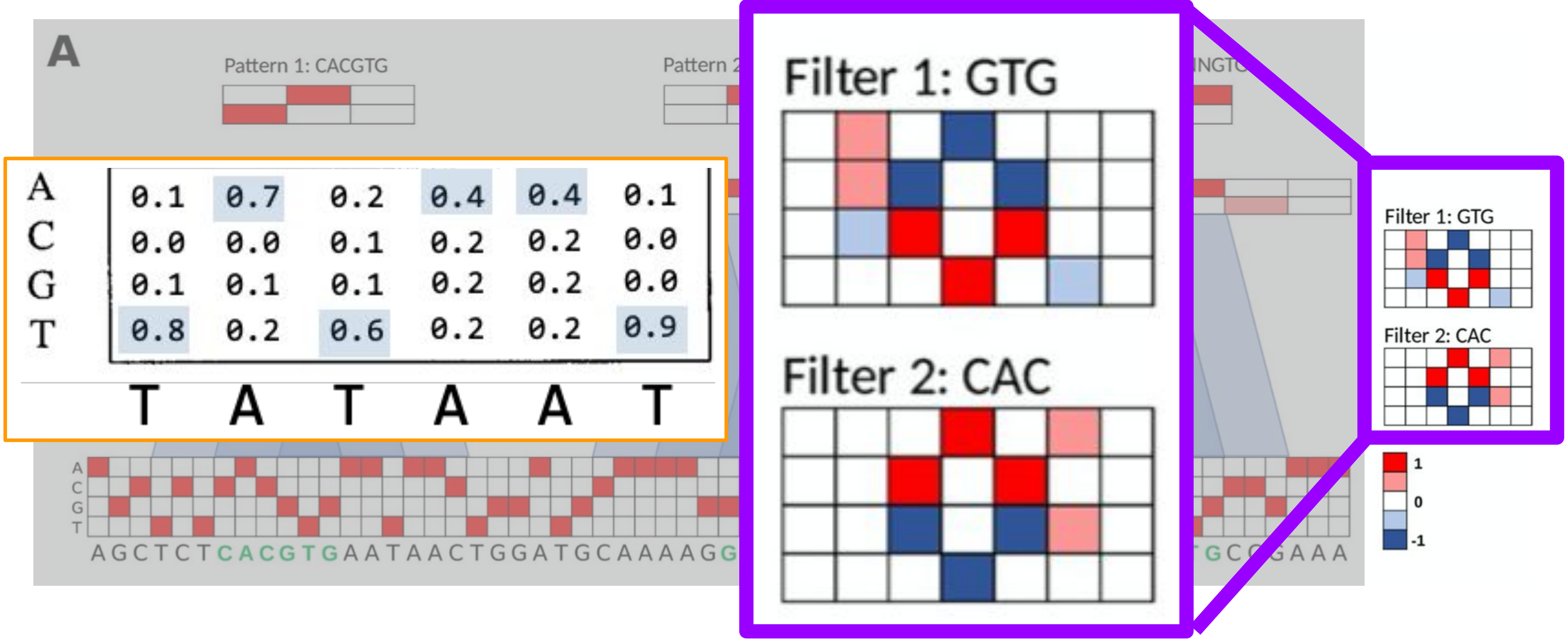
Main question:

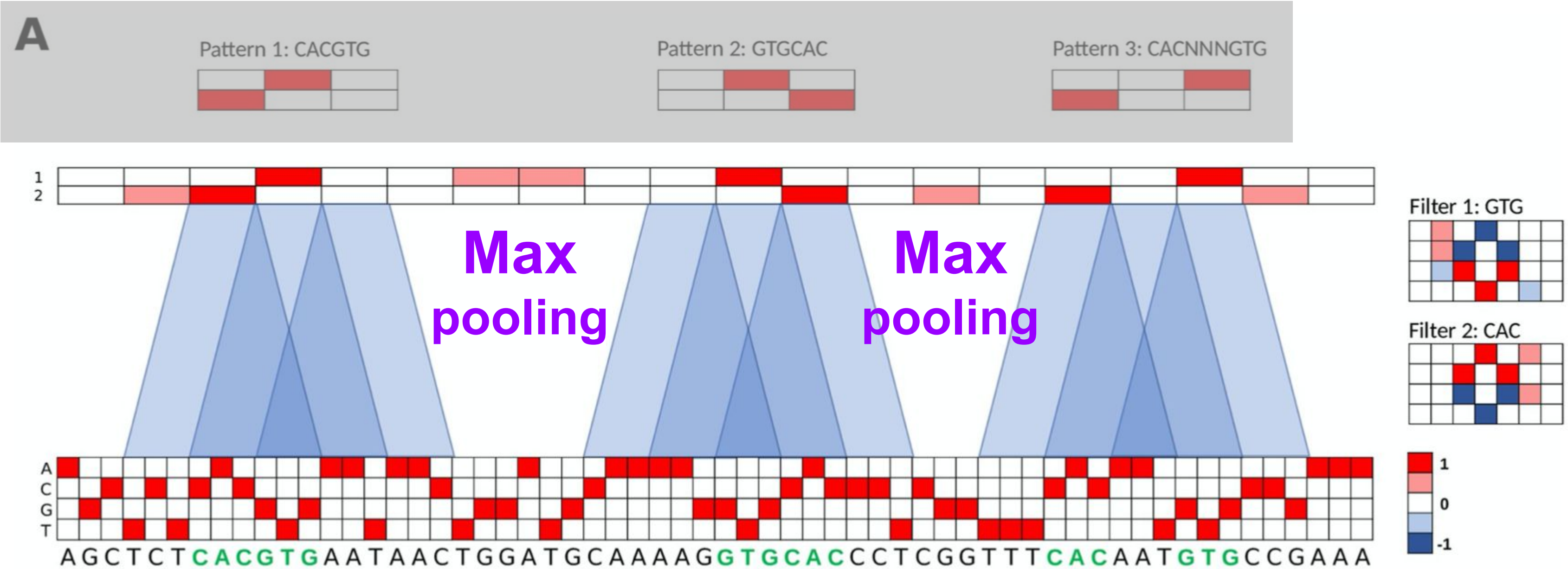
How does the ***architecture*** of the CNN influence its ability to ***learn whole motifs*** in the first convolutional layer?

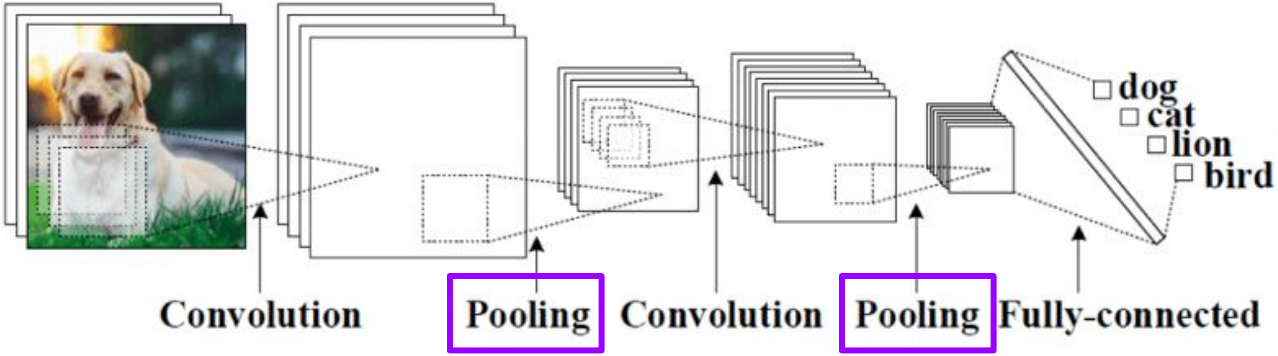


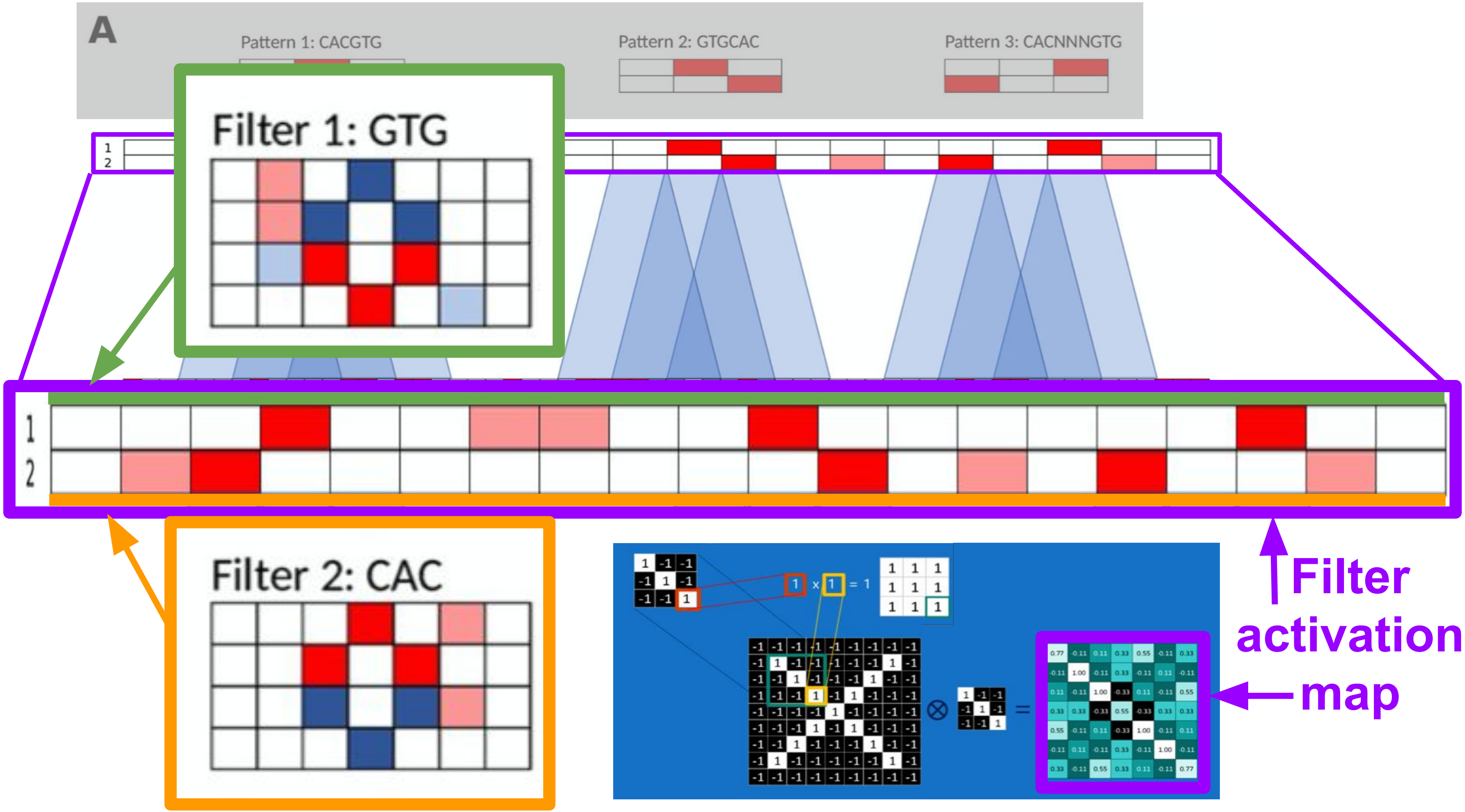
A
Pattern 1: CACGTG
Pattern 2: GTGCAC
Pattern 3: CACNNNGTG
1
2
Filter 1: GTG
Filter 2: CAC
1
0
-1
A
C
G
T
AGCTCTCACGTGAATAACTGGATGCAAAAGGTGCACCCTCGGTTTCACAATGTGCCGAAA


A
Pattern 1: CACGTG
Pattern 2: GTGCAC
Pattern 3: CACNNNGTG
Filter 1: GTG
1
2
A
C
G
T
AGCTCTCACGTGAATAACTGGATGCAAAAGGTG
G
C
T
A
CGGTCAGCACTCTGAGAGTCAGCGTATAAGTTACGCTACGCGTAAGCTTGTA

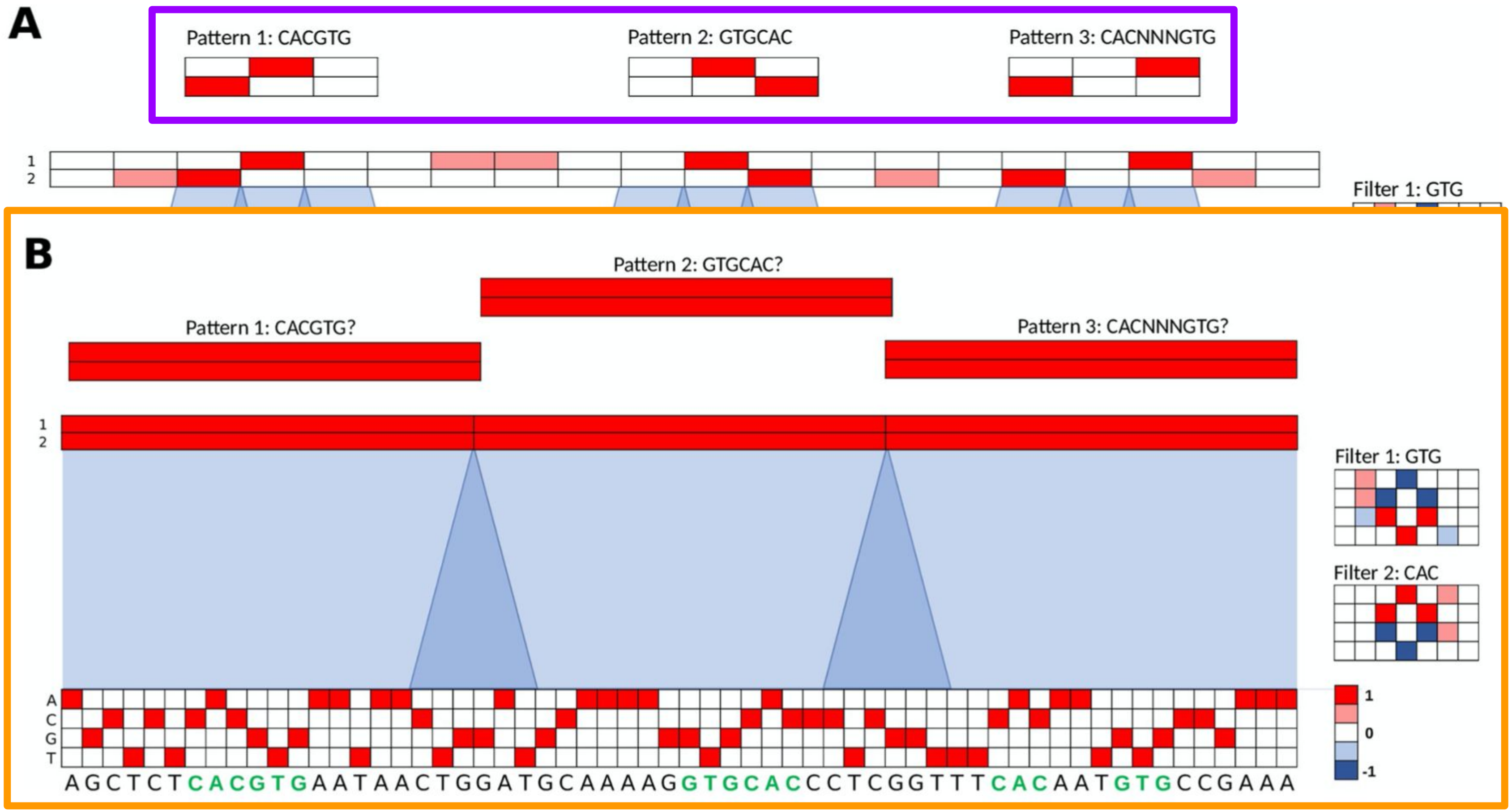

A
Pattern 1: CACGTG
Pattern 2
NGTG
A C G T
0.1 0.7 0.2 0.4 0.4 0.1
0.0 0.0 0.1 0.2 0.2 0.0
0.1 0.1 0.1 0.2 0.2 0.0
0.8 0.2 0.6 0.2 0.2 0.9
T A T A A T
A C G T
AGCTCTCACGTGAATAACTGGATGCAAAAGG
TGCCGAAA
Filter 1: GTG
Filter 2: CAC
Filter 1: GTG
Filter 2: CAC
1
0
-1


A
Pattern 1: CACGTG
Pattern 2: GTGCAC
Pattern 3: CACNNNGTG
1
2
Max pooling
Max pooling
Filter 1: GTG
Filter 2: CAC
A
C
G
T
1
0
-1
AGCTCTCACGTGAATAACTGGATGCAAAAGGTGCACCCTCGGTTTCACAATGTGCCGAAA


dog
cat
lion
bird
Convolution
Pooling
Convolution
Pooling
Fully-connected


A
Pattern 1: CACGTG
Pattern 2: GTGCAC
Pattern 3: CACNNNGTG
Filter 1: GTG
1
2
Filter 2: CAC
Filter
activation
map

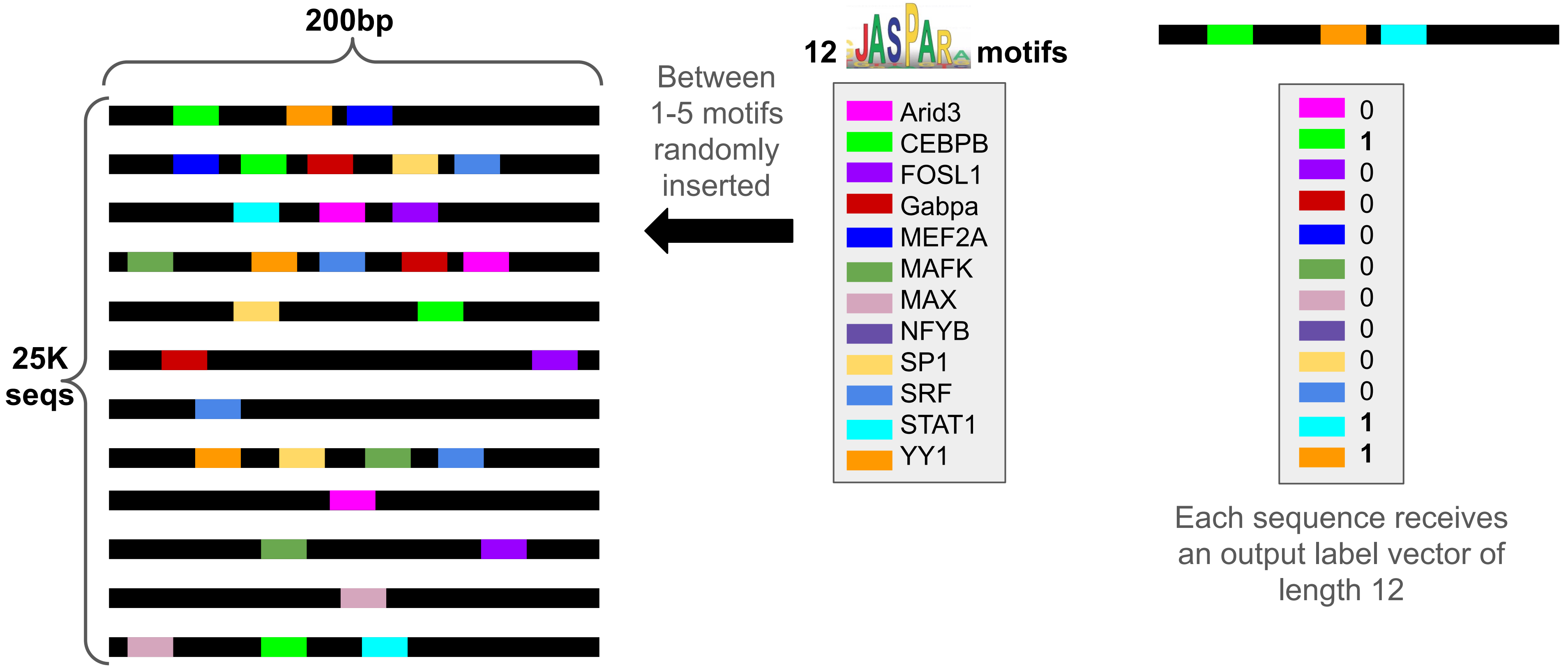
Second layer convolutional filters



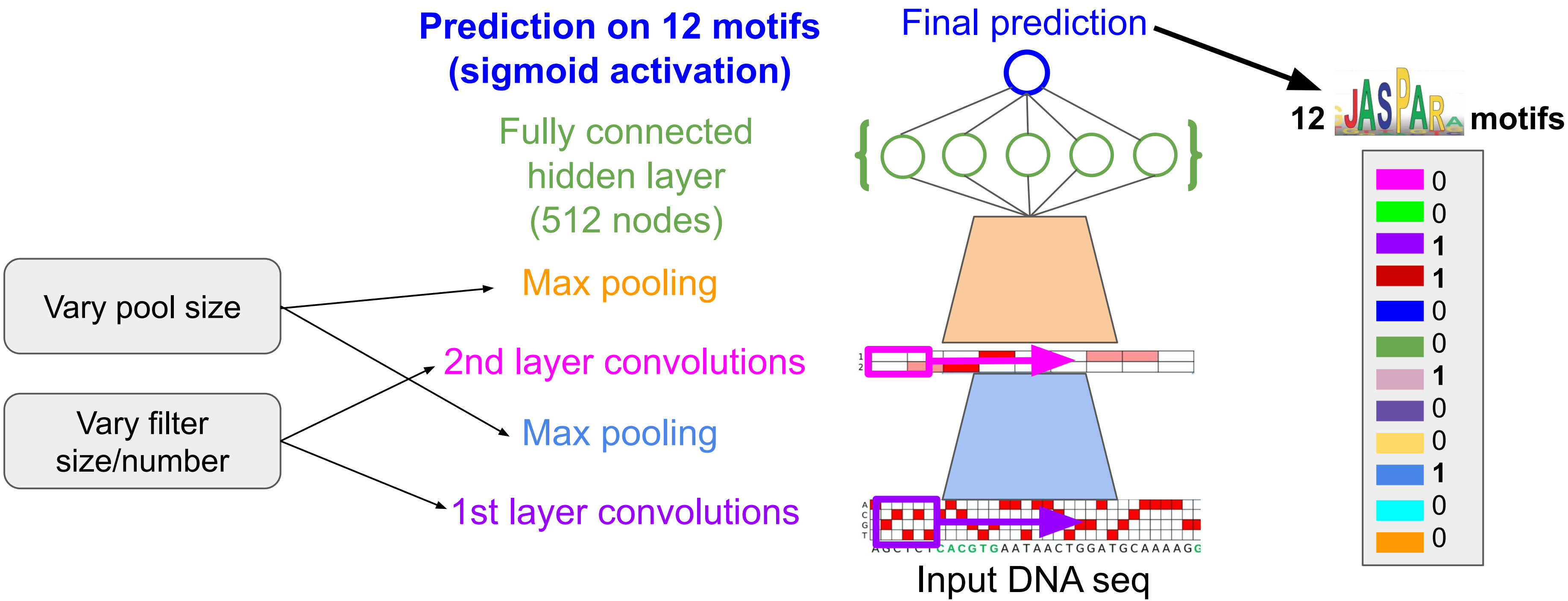
Overview

- Background
  - How do CNNs work?
  - Why do people care about finding motifs?
- **Methods**
  - **Synthetic dataset used in this paper**
  - **Experimental setup for CNN architectures**
  - **Model vs Motif evaluation metrics**
- Results
  - Pulling various CNN architecture levers!
- Main takeaways & discussion

Methods: creating a synthetic dataset

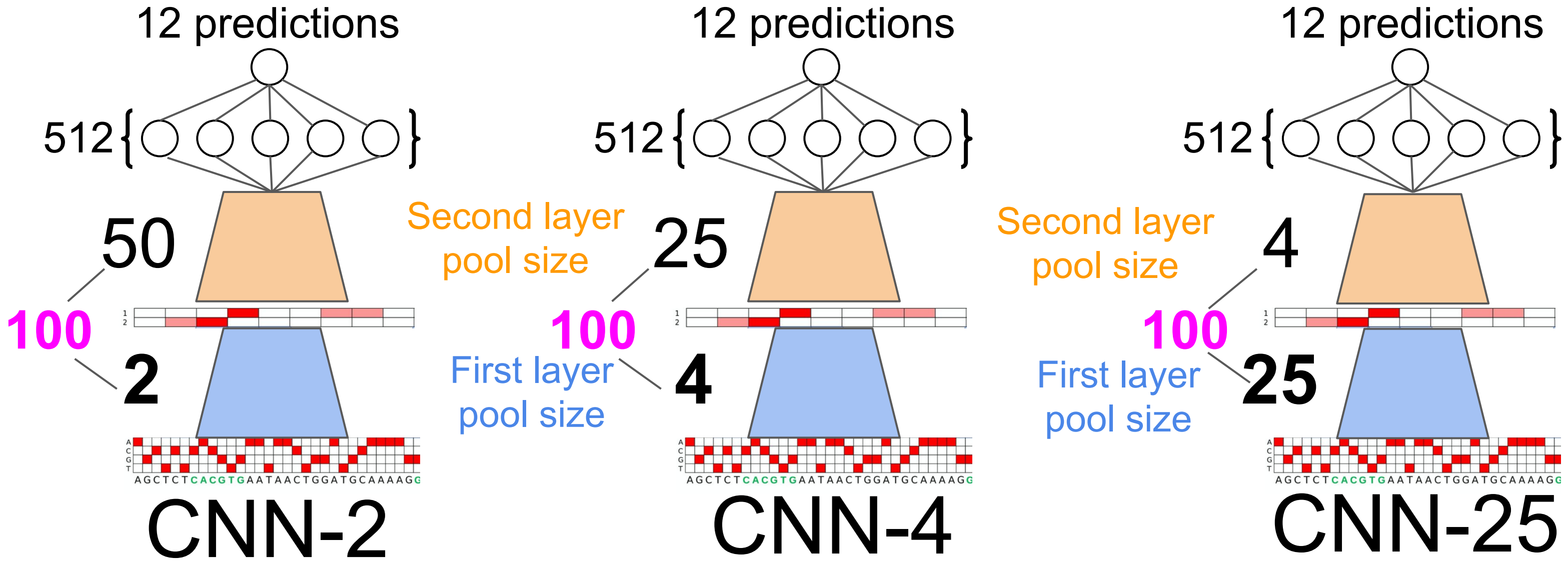


Methods: Network architecture framework



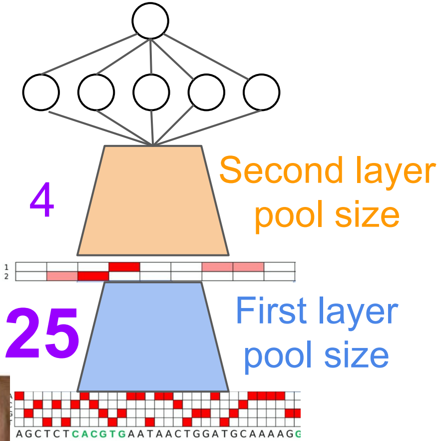
Methods: Network naming scheme

Max-pooling: **product** of first and second pool sizes is **100.**

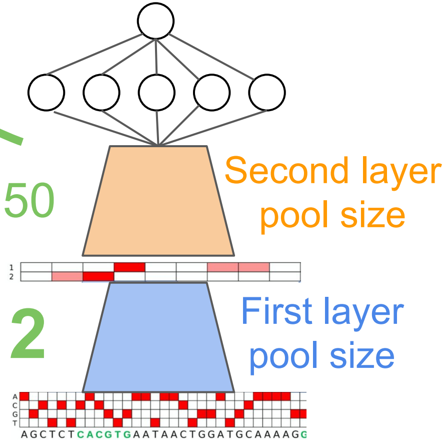


Models are (mostly) named for their first pool size

Model	Average AU-ROC
CNN-1	0.972±0.001
CNN-2	0.966±0.000
CNN-4	0.955±0.002
CNN-10	0.964±0.007
CNN-25	0.973±0.001
CNN-50	0.961±0.011
CNN-100	0.958±0.012
CNN_9-4	0.954±0.002
CNN_9-25	0.958±0.008
CNN_3-50	[illegible]
CNN_3-2	[illegible]
CNN-50-2	[illegible]
CNN_{19-1}-2	[illegible]
CNN-25 (60)	[illegible]
CNN-25 (90)	[illegible]
CNN-25 (120)	0.963±0.001



CNN-25



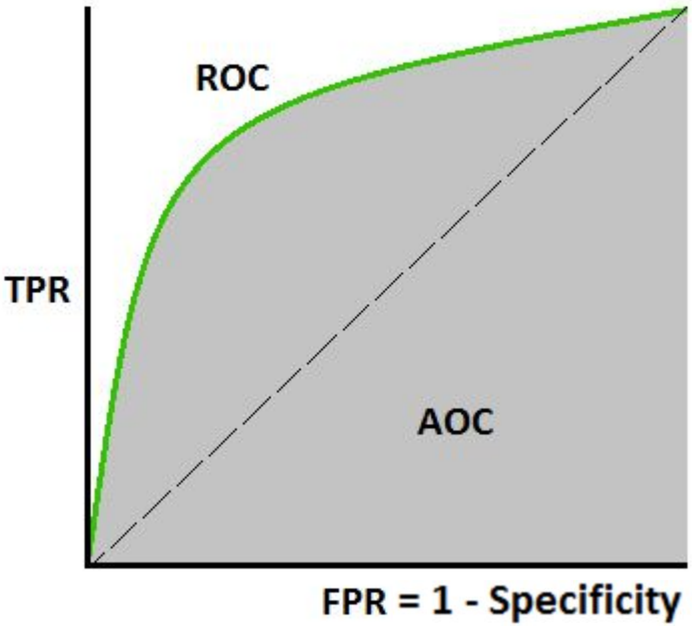
CNN-2

Methods: evaluate models using AU-ROC

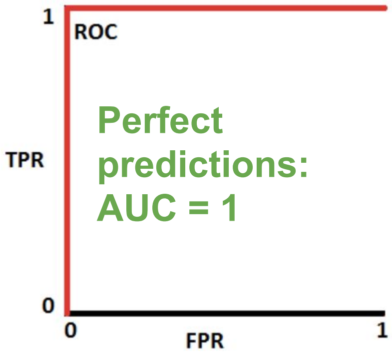
True positive rate

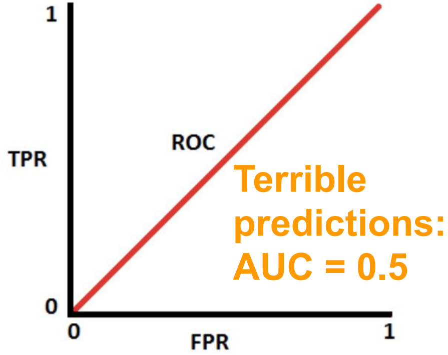
TPR /Recall / Sensitivity

$$\frac{TP}{TP + FN}$$



False positive rate $= \dfrac{FP}{TN + FP}$





https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5

Evaluate models for consistent AU-ROC, not best!

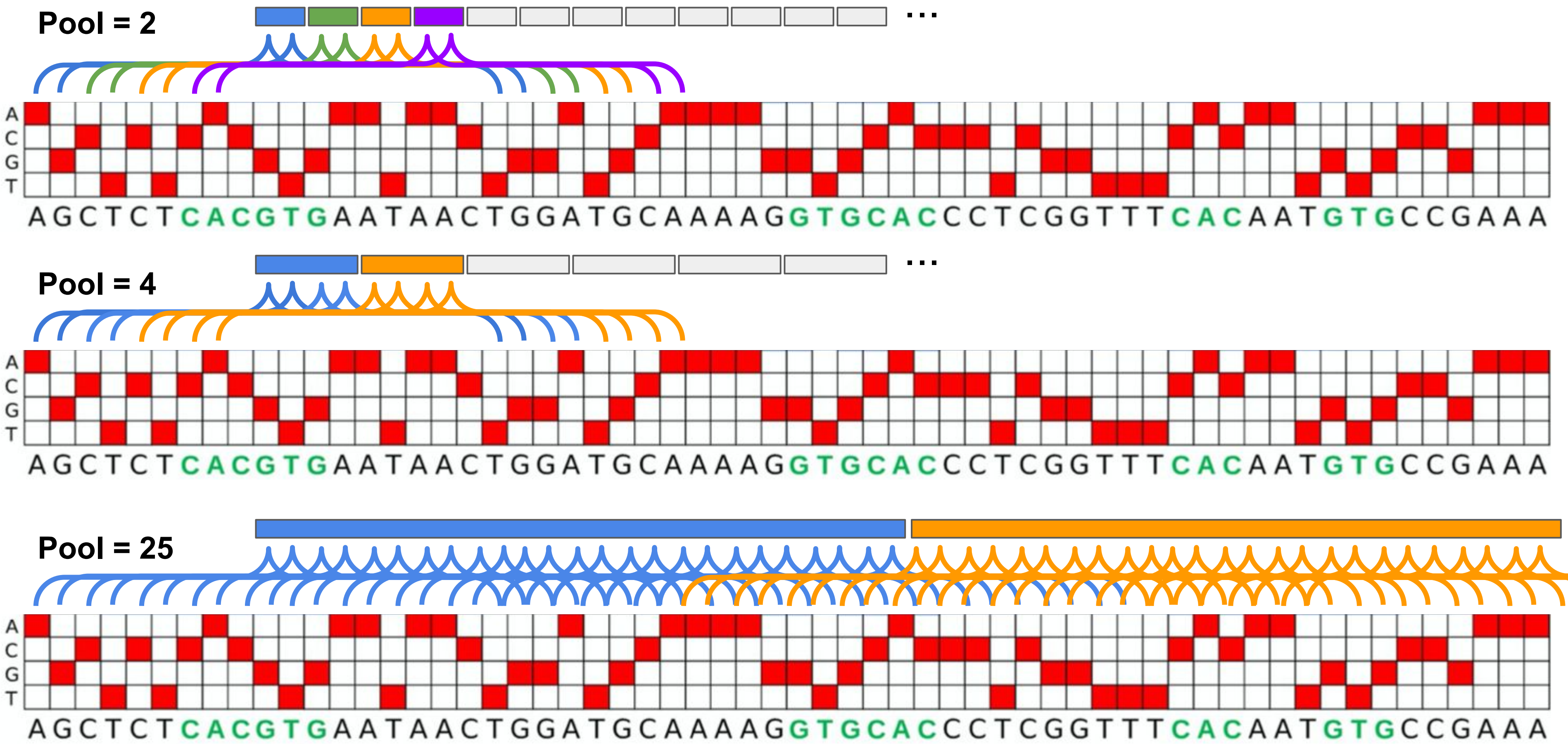
Model	Average AU-ROC
CNN-1	0.972±0.001
CNN-2	0.966±0.000
CNN-4	0.955±0.002
CNN-10	0.964±0.007
CNN-25	0.973±0.001
CNN-50	0.961±0.011
CNN-100	0.958±0.012
CNN_9-4	0.954±0.002
CNN_9-25	0.958±0.008
CNN_3-50	0.648±0.008
CNN_3-2	0.968±0.001
CNN-50-2	0.921±0.012
$CNN_{19\text{-}1}$-2	0.969 ±0.002
CNN-25 (60)	0.972±0.001
CNN-25 (90)	0.968±0.001
CNN-25 (120)	0.963±0.001

- **Not** concerned with maximizing AU-ROC - want to be **consistent**
- Real question: after change some aspect of network **structure**, and evaluate the **motifs learned** by first layer filters

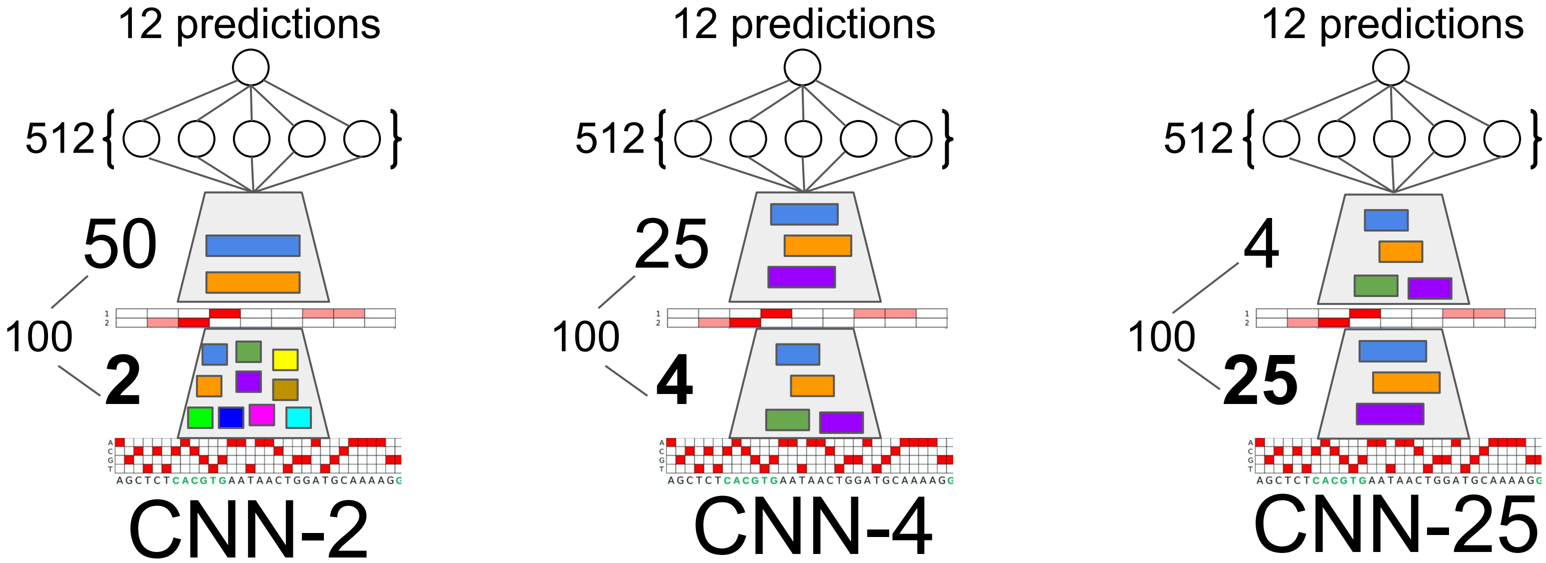
Overview

- Background
  - How do CNNs work?
  - Why do people care about finding motifs?
- Methods
  - Synthetic dataset used in this paper
  - Experimental setup for CNN architectures
  - Model vs Motif evaluation metrics
- **Results**
  - **Pulling various CNN architecture levers!**
- Main takeaways

Results: vary max pooling size

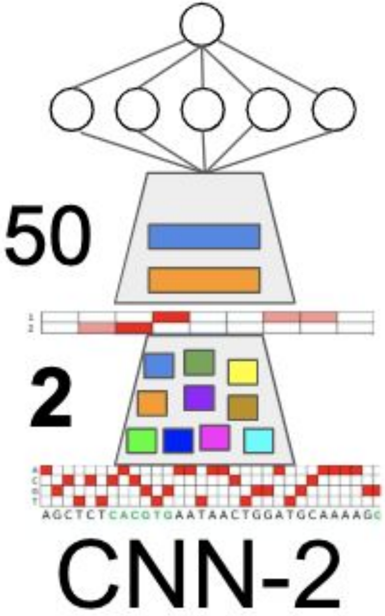


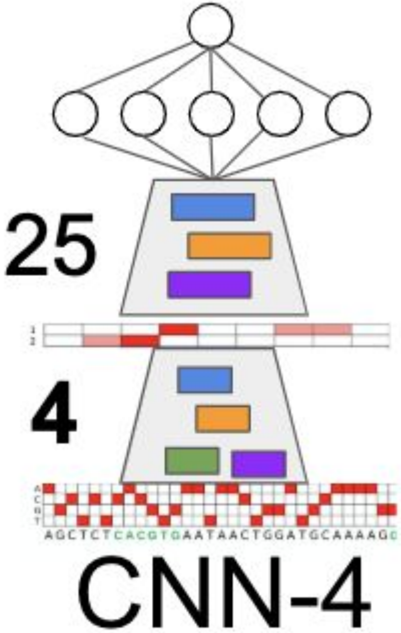
Results: vary max pooling size

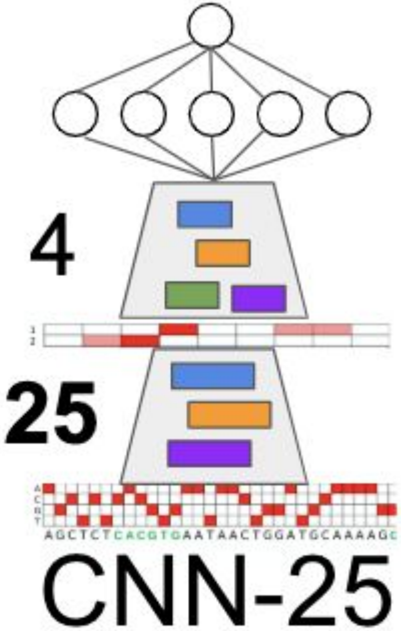


Results: vary max pooling size

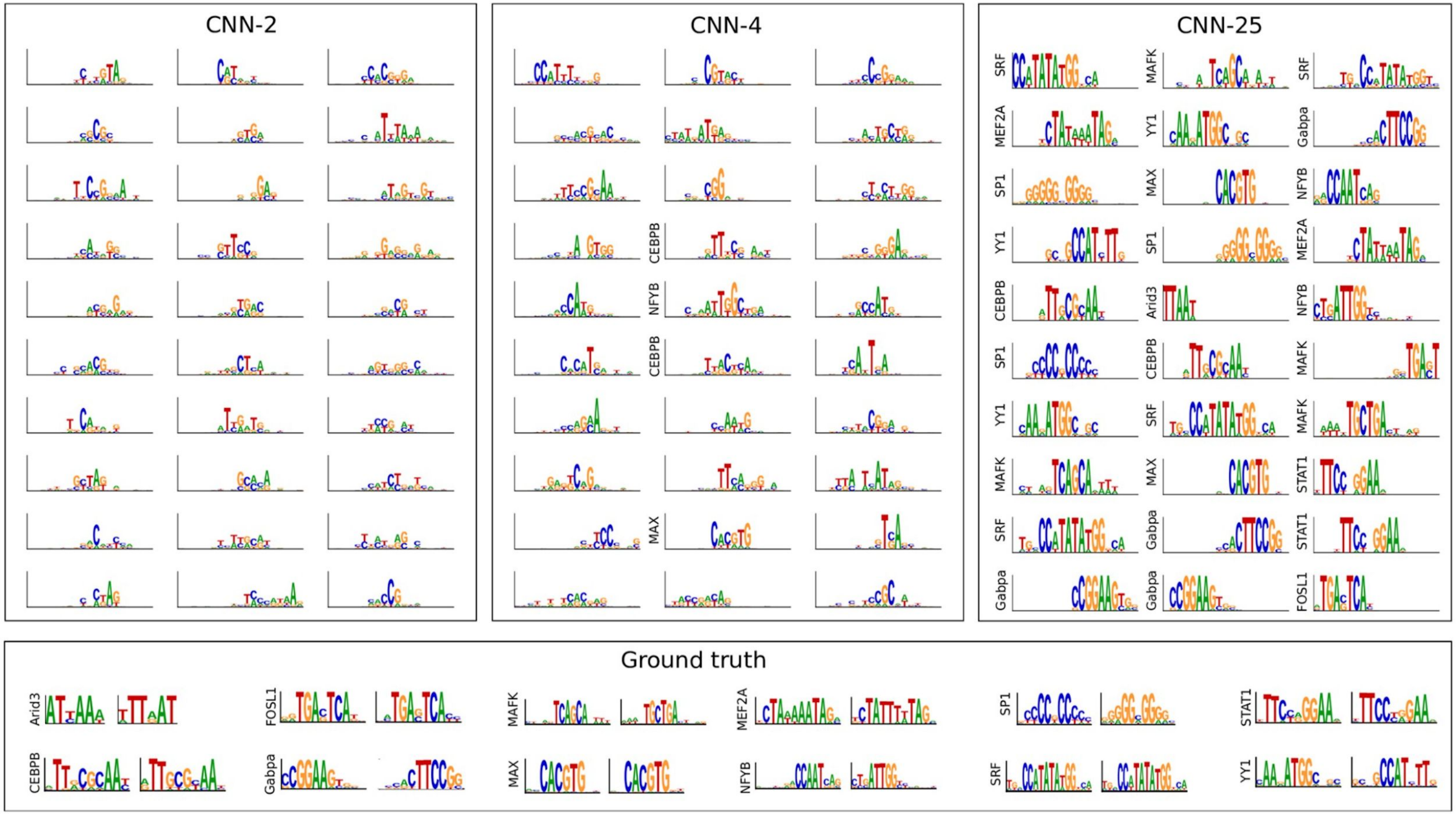
(image versions to use without fixing font size everytime)



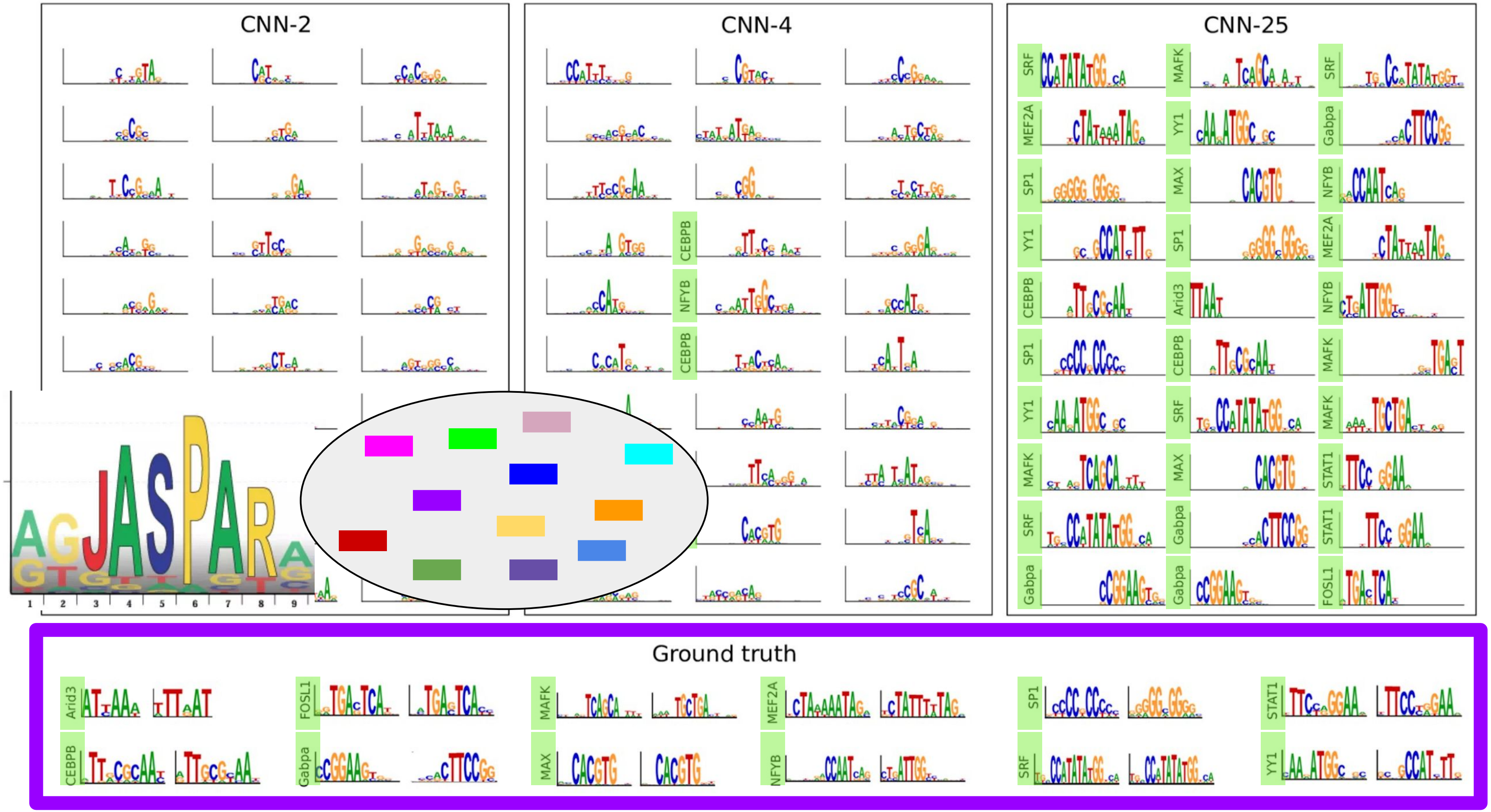




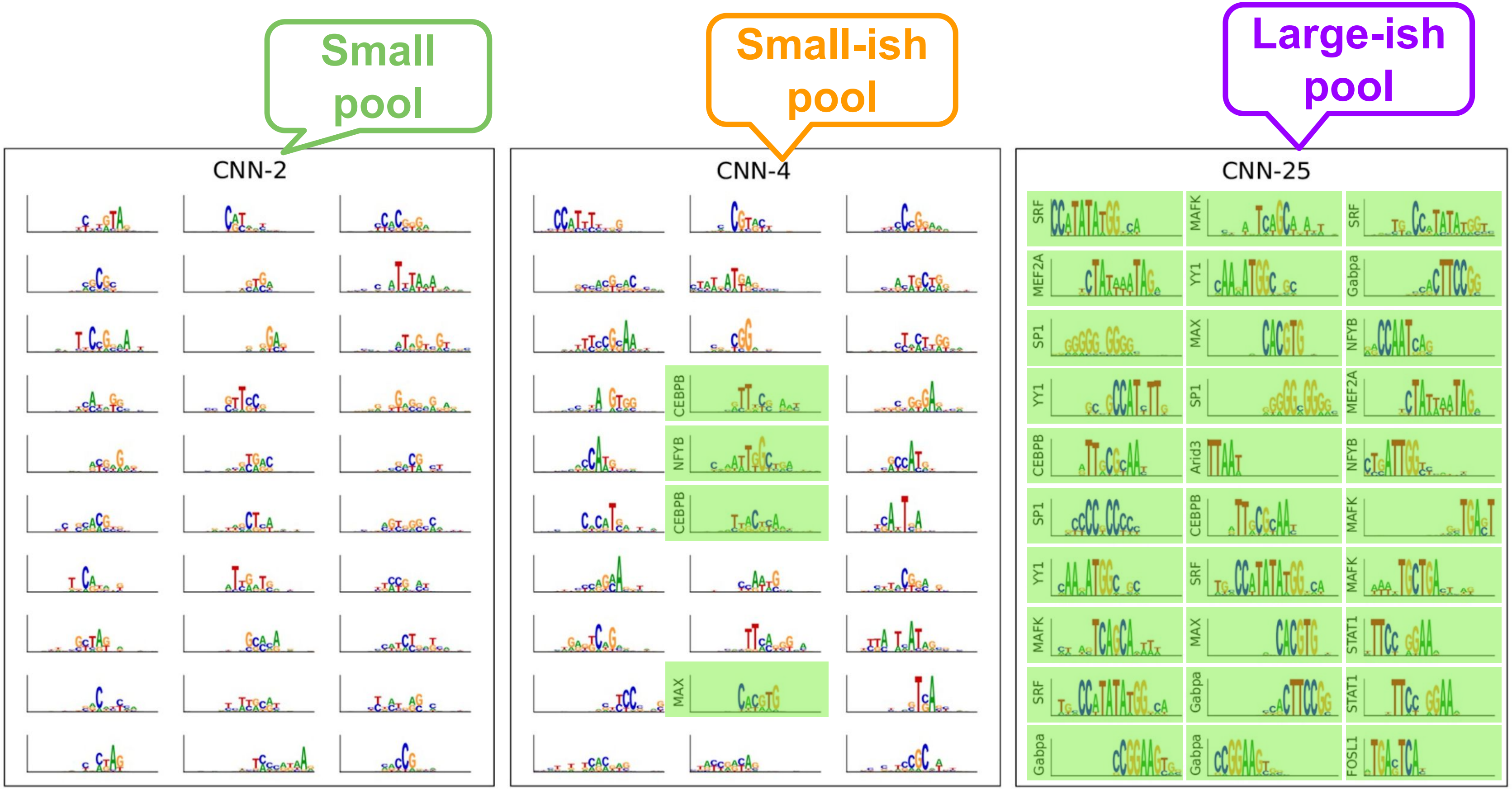
Results: vary max pooling size



Results: vary max pooling size



Results: vary max pooling size



Results: vary max pooling

12 Ground Truth motifs

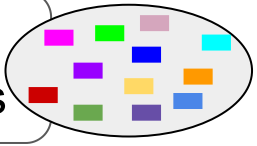


Table 1. Performance on the synthetic dataset.

Small pool

Small-ish pool

Large-ish pool

Model	Average AU-ROC	% Motif match (JASPAR)	% Motif match (Relevant)
CNN-1	0.972±0.001	0.240±0.083	0.007±0.013
CNN-2	0.966±0.000	0.240±0.071	0.007±0.013
CNN-4	0.955±0.002	0.453±0.131	0.127±0.080
CNN-10	0.964±0.007	0.987±0.016	0.973±0.033
CNN-25	0.973±0.001	0.987±0.016	0.980±0.027
CNN-50	0.961±0.011	0.933±0.037	0.920±0.045
CNN-100	0.958±0.012	0.887±0.034	0.880±0.034
CNN_9-4	0.954±0.002	0.260±0.039	0.033±0.030
CNN_9-25	0.958±0.008	0.993±0.013	0.980±0.016
CNN_3-50	0.648±0.008	0.160±0.049	0.000±0.000
CNN_3-2	0.968±0.001	0.233±0.067	0.000±0.000
CNN-25 (120)	0.963±0.001	0.933±0.015	0.887±0.025

Take away: wider pooling size (like CNN-25) forces first row filters to learn WHOLE motifs

Results: vary filter number

12 Ground Truth motifs

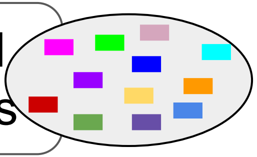


Table 1. Performance on the synthetic dataset.

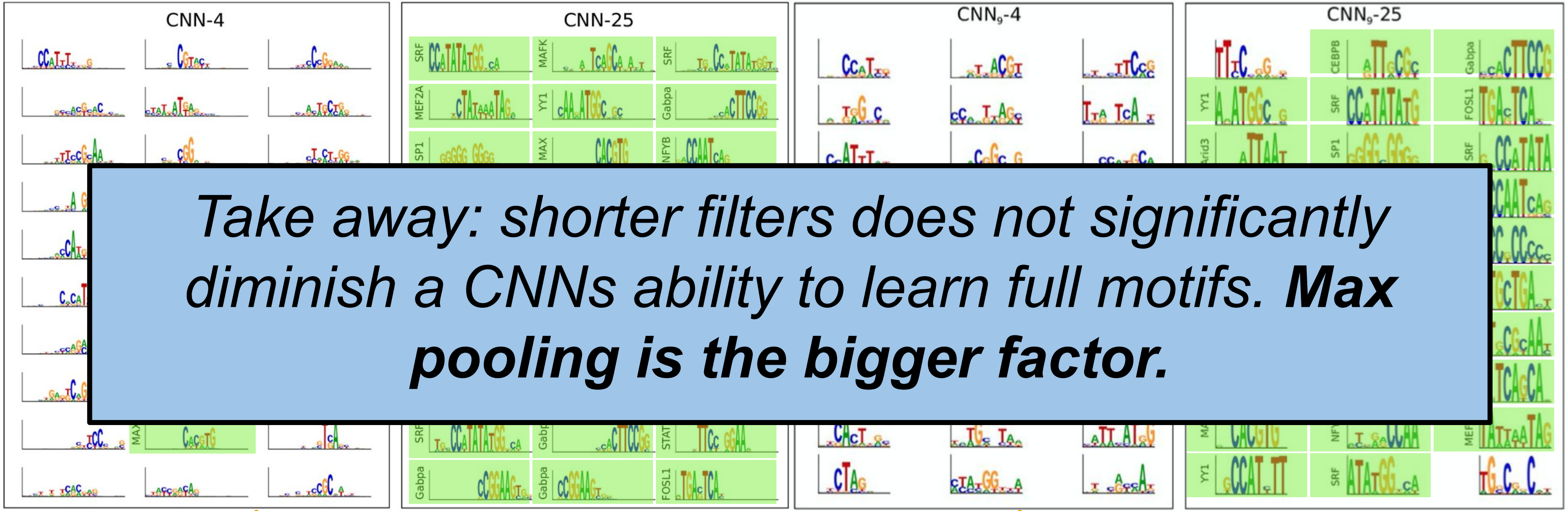
Model	Average AU-ROC	% Motif match (JASPAR)	% Motif match (Relevant)
CNN-1	0.972±0.001	0.240±0.083	0.007±0.013
CNN-2	0.966±0.000	0.240±0.071	0.007±0.013
CNN-4	0.955±0.002	0.453±0.131	0.127±0.080
CNN-10	0.964±0.007	0.987±0.016	0.973±0.033
Default: 30 filters > CNN-25	0.973±0.001	0.987±0.016	0.980±0.027
CNN-50	0.961±0.011	0.933±0.037	0.920±0.045
CNN-100	0.958±0.012	0.887±0.034	0.880±0.034
CNN-50-2	0.921±0.012	0.913±0.050	0.893±0.044
$CNN_{19\text{-}1}$-2	0.969 ±0.002	0.867±0.056	0.747±0.096
CNN-25 (60)	0.972±0.001	0.973±0.013	0.960±0.023
CNN-25 (90)	0.968±0.001	0.940±0.023	0.909±0.028
CNN-25 (120)	0.963±0.001	0.933±0.015	0.887±0.025

Take away: more filters does not improve accuracy and ***% of filters that learn motifs decreases***

Vary # filters

Results: vary filter size

Default: filter size = 19 | Test filter size = 9

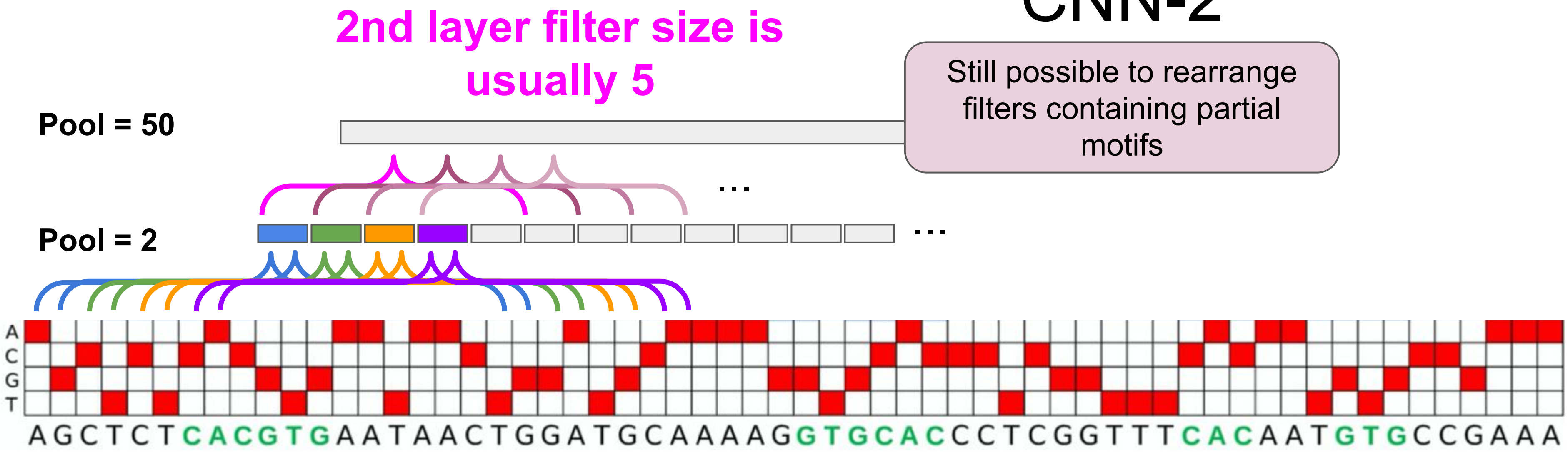


Take away: shorter filters does not significantly diminish a CNNs ability to learn full motifs. ***Max pooling is the bigger factor.***

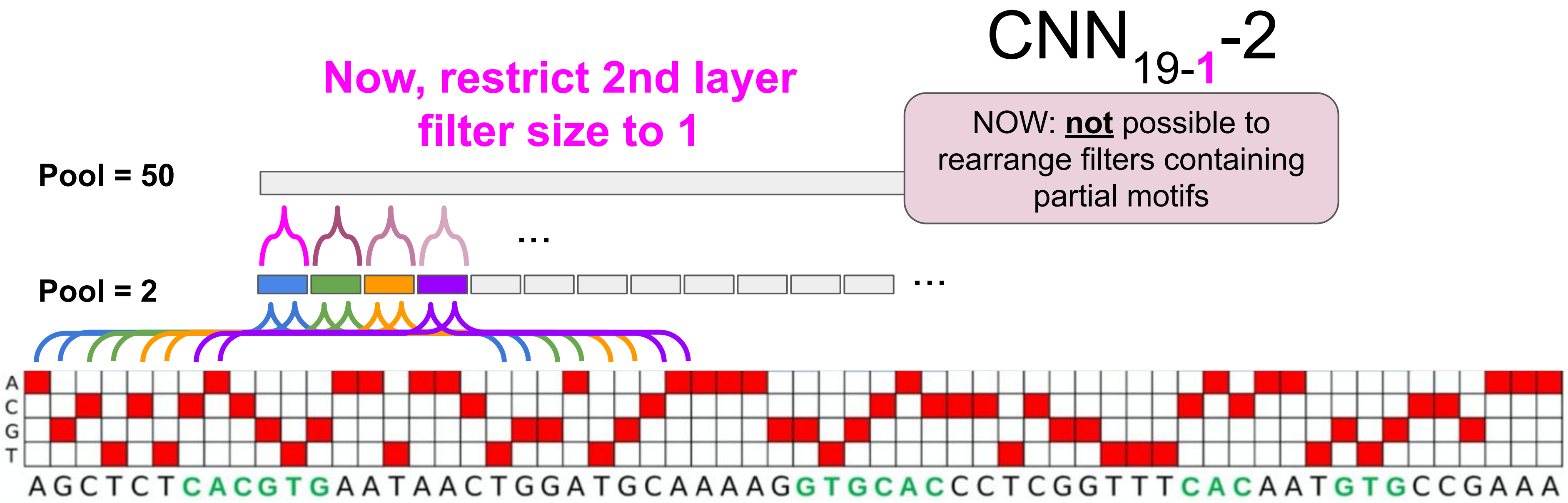
Small-ish pool | **Large-ish pool** | **Small-ish pool** | **Large-ish pool**

Results: Restricting deeper layer assemblies

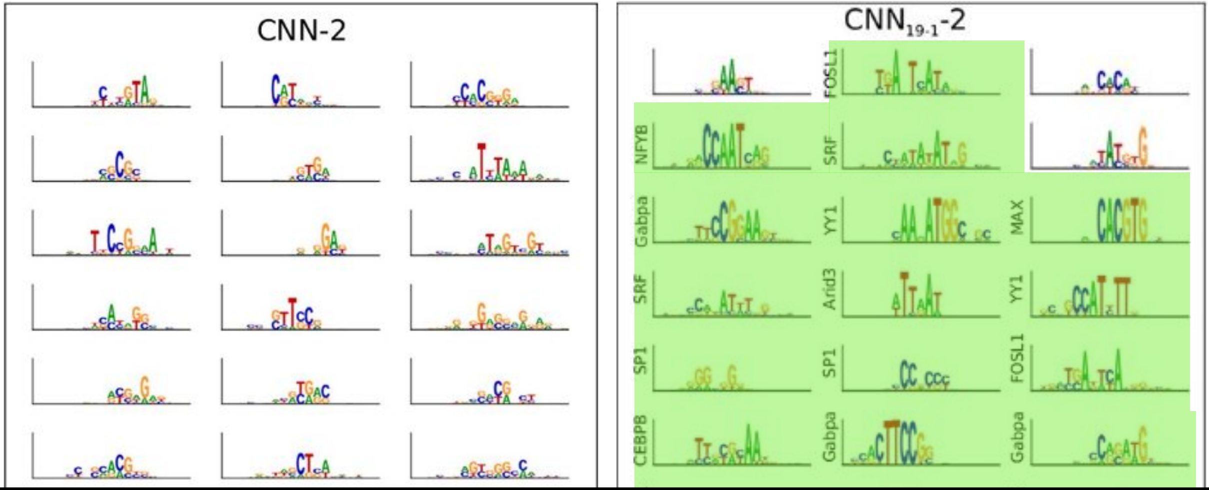
CNN-2



Results: Restricting deeper layer assemblies



CNN-2 CNN$_{19\text{-}1}$-2



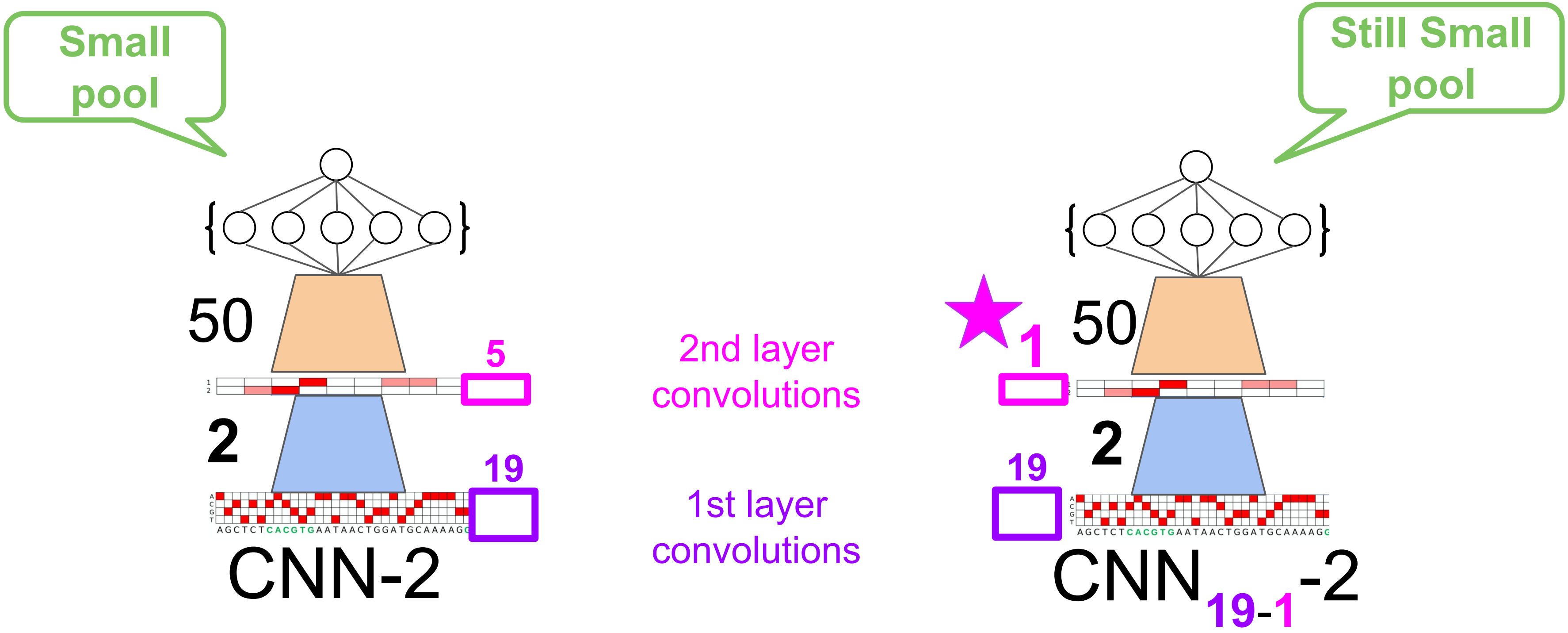
*Take away: learning WHOLE motif representations in first layer is affected by the ability of **deeper layers to hierarchically build motifs.***

Results: Restricting deeper layer assemblies

Table 1. Performance on the synthetic dataset.

Model	Average AU-ROC	% Motif match (JASPAR)	% Motif match (Relevant)
CNN-1	0.972±0.001	0.240±0.083	0.007±0.013
CNN-2	0.966±0.000	0.240±0.071	0.007±0.013
CNN-4	0.955±0.002	0.453±0.131	0.127±0.080
CNN-10	0.964±0.007	0.987±0.016	0.973±0.033
CNN-25	0.973±0.001	0.987±0.016	0.980±0.027
CNN-50	0.961±0.011	0.933±0.037	0.920±0.045
CNN-100	0.958±0.012	0.887±0.034	0.880±0.034
CNN_9-4	0.954±0.002	0.260±0.039	0.033±0.030
CNN_9-25	0.958±0.008	0.993±0.013	0.980±0.016
CNN_3-50	0.648±0.008	0.160±0.049	0.000±0.000
CNN_3-2	0.968±0.001	0.233±0.067	0.000±0.000
CNN-50-2	0.921±0.012	0.913±0.050	0.893±0.044
$CNN_{19\text{-}1}$-2	0.969 ±0.002	0.867±0.056	0.747±0.096
CNN-25 (60)	0.972±0.001	0.973±0.013	0.960±0.023
CNN-25 (90)	0.968±0.001	0.940±0.023	0.909±0.028
CNN-25 (120)	0.963±0.001	0.933±0.015	0.887±0.025

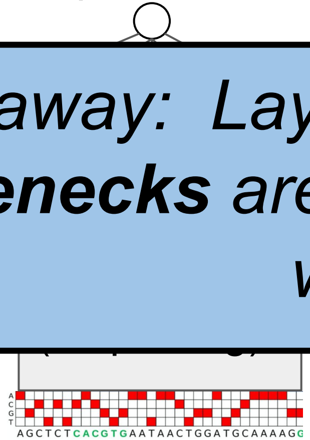
Results: Restricting deeper layer assemblies



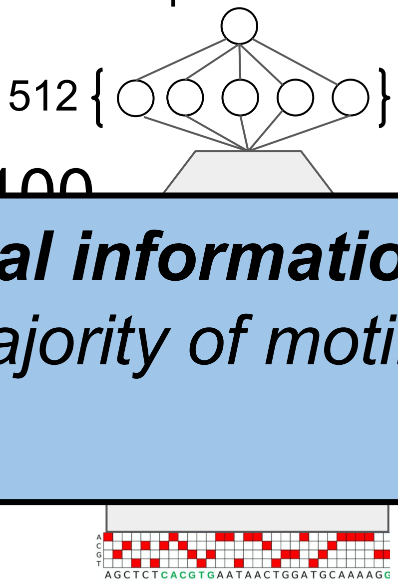
Results: Restricting deeper layer assemblies

Model	Average AU-ROC	% Motif match (JASPAR)	% Motif match (Relevant)
CNN-2	0.966±0.000	0.240±0.071	0.007±0.013
CNN-50-2	0.921±0.012	0.913±0.050	0.893±0.044
$CNN_{19\text{-}1}$-2	0.969 ±0.002	0.867±0.056	0.747±0.096

Results: filters can be learned in deeper layers



CNN-1



CNN-1-1-100

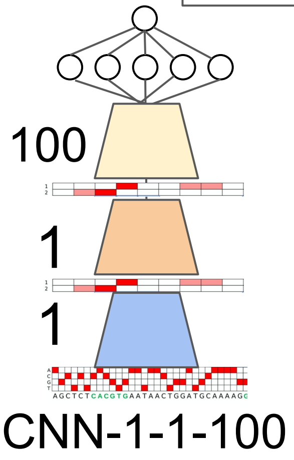
Take away: Layers with ***spatial information bottlenecks*** *are where the majority of motifs will be learned*

	First layer filters	Second layer filters
AU-ROC:	0.972	0.972
% JASPAR:	0.240	**0.900**
% Relevant:	0.007	**0.847**

	First layer filters	Second layer filters	Third layer filters
AU-ROC:	--	--	--
% JASPAR:	0.147	0.192	**0.927**
% Relevant:	0.000	0.006	**0.891**

Results: motifs are learned at the information bottleneck

Model	Average AU-ROC	% Motif match (JASPAR)	% Motif match (Relevant)
CNN-1	0.972±0.001	0.240±0.083	0.007±0.013
CNN-1 second layer filters →		**0.900 +/- 0.024**	**0.847 +/- 0.021**
CNN-1-1-100			
CNN-1-1-100 first layer filters →		0.147 +/- 0.045	0.0 +/- 0.0
CNN-1-1-100 second layer filters →		0.192 +/- 0.022	0.006 +/- 0.006
CNN-1-1-100 second layer filters →		**0.927 +/- 0.020**	**0.891 +/- 0.030**



*Take away: Layers with spatial **information bottlenecks** are where the majority of motifs will be learned*

Results: *In Vivo* Generalizations

Nat Methods. 2015 Oct;12(10):931-4. doi: 10.1038/nmeth.3547. Epub 2015 Aug 24.

Predicting effects of noncoding variants with deep learning-based sequence model.

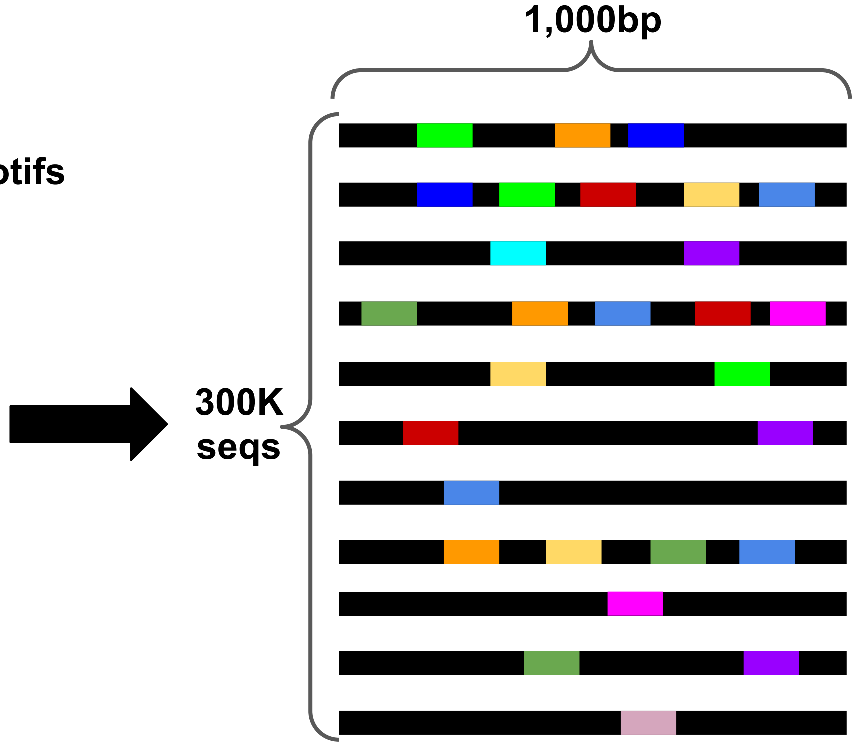
Zhou J[1,2], Troyanskaya OG[1,3,4]. **Cited ~800 times!**

Human ChIP-seq data

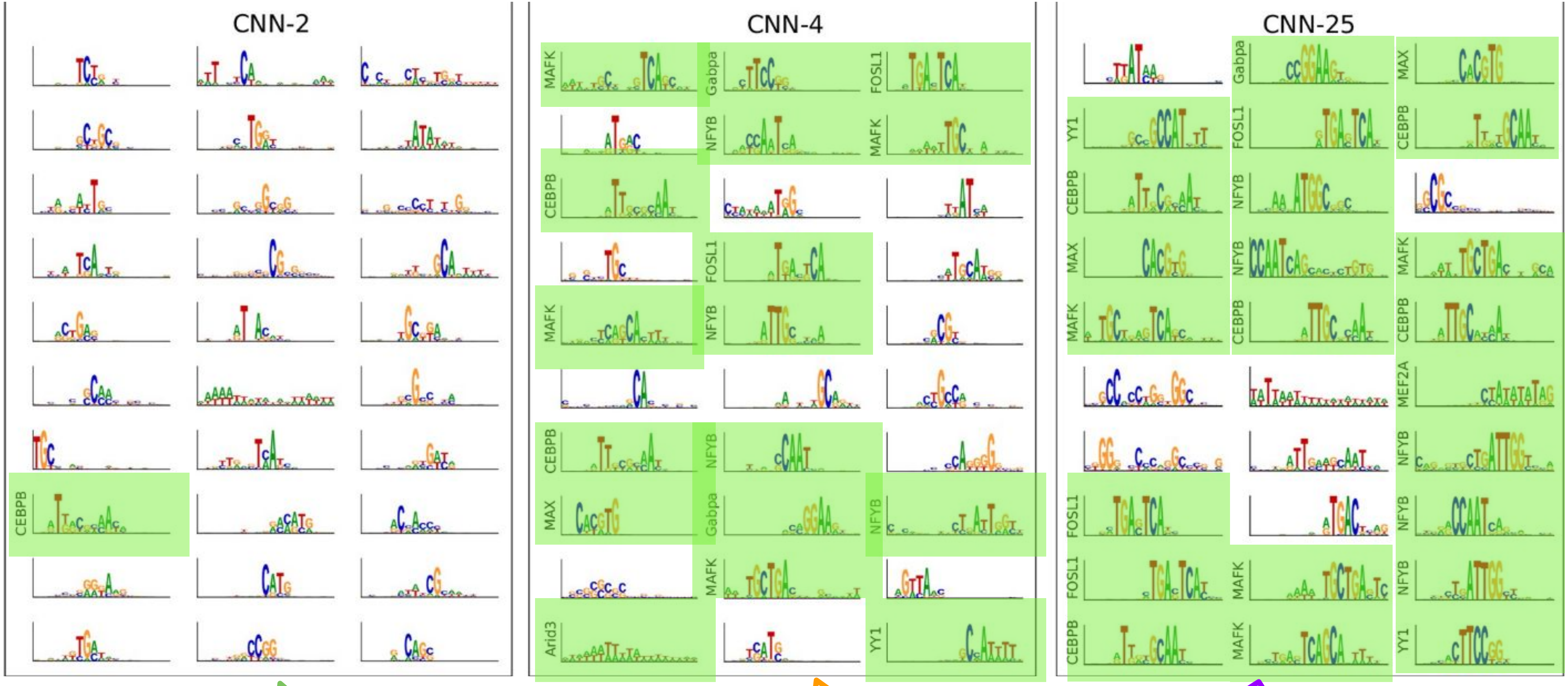
DeepSEA

12 JASPAR motifs

- Arid3
- CEBPB
- FOSL1
- Gabpa
- MEF2A
- MAFK
- MAX
- NFYB
- SP1
- SRF
- STAT1
- YY1



Results: *in vivo* dataset



Small pool

Small-ish pool

Large-ish pool

Table 2. Performance of deep learning models on the *in vivo* dataset.

Model	Average AU-ROC	Average AU-PR	Motif match (JASPAR)	Motif match (Relevant)
CNN-1	0.918±0.001	0.626±0.000	0.227±0.068	0.020±0.027
CNN-2	0.911±0.003	0.609±0.006	0.333±0.067	0.107±0.057
CNN-4	0.907±0.002	0.601±0.005	0.753±0.045	0.507±0.039
CNN-10	0.903±0.006	0.583±0.020	0.920±0.045	0.753±0.034
CNN-25	0.903±0.003	0.580±0.009	0.933±0.030	0.747±0.040
CNN-50	0.903±0.003	0.582±0.009	0.913±0.034	0.733±0.063
CNN-25 (90)	0.919±0.001	0.628±0.005	0.940±0.023	0.909±0.028
CNN-25 (120)	0.920±0.002	0.637±0.005	0.933±0.015	0.887±0.025

*Take away: Architectures may need **more filters** to perform better on **in vivo** sequences*

On Synthetic Data:
CNN-25 had the best Relevant match w/ 0.980 +/- 0.027

Now:
CNN-25: 0.747 +/- 0.040 Relevant match
CNN-25 (60): has the best Relevant match performance with 0.960 +/- 0.023

Results Summary

- CNN architecture choices affect how motifs are learned
  - **Wider pooling** size forces first layer filters to learn whole motifs
  - Filter **number** and filter **size** are less influential
  - **Restricting hierarchical assembly** in deeper layers can increase first layer motif learning
  - Motifs are learned at the **information bottleneck** (can be 1st, 2nd, 3rd layer)
  - With ***in vivo*** dataset **more filters helped** with distributed representation learning

Overview

- Background
  - How do CNNs work?
  - Why do people care about finding motifs?
- Methods
  - Synthetic dataset used in this paper
  - Experimental setup for CNN architectures
  - Model vs Motif evaluation metrics
- Results
  - Pulling various CNN architecture levers!
- **Main takeaways & discussion**

Main Takeaways & Discussion

- Exploration of various CNN architectures to better understand **how** and **where** CNNs learn motifs
    - *Was this a useful aspect to explore?*

- % of 1st layer filters that learn motifs is *not necessarily a useful metric* for assessing biological relevance because CNNs can **assemble partial motifs in deeper layers**
    - *Do you agree? Would you still want this reported?*

- If you want to **enforce** that your CNN learns whole motifs in the **1st layer**, be mindful of your architecture
    - *Would you consider doing this intentionally in your own work?*

Thanks!

Second Beach, La Push, WA