Base-resolution models of transcription factor binding reveal soft motif syntax

Avsec et al. 2020

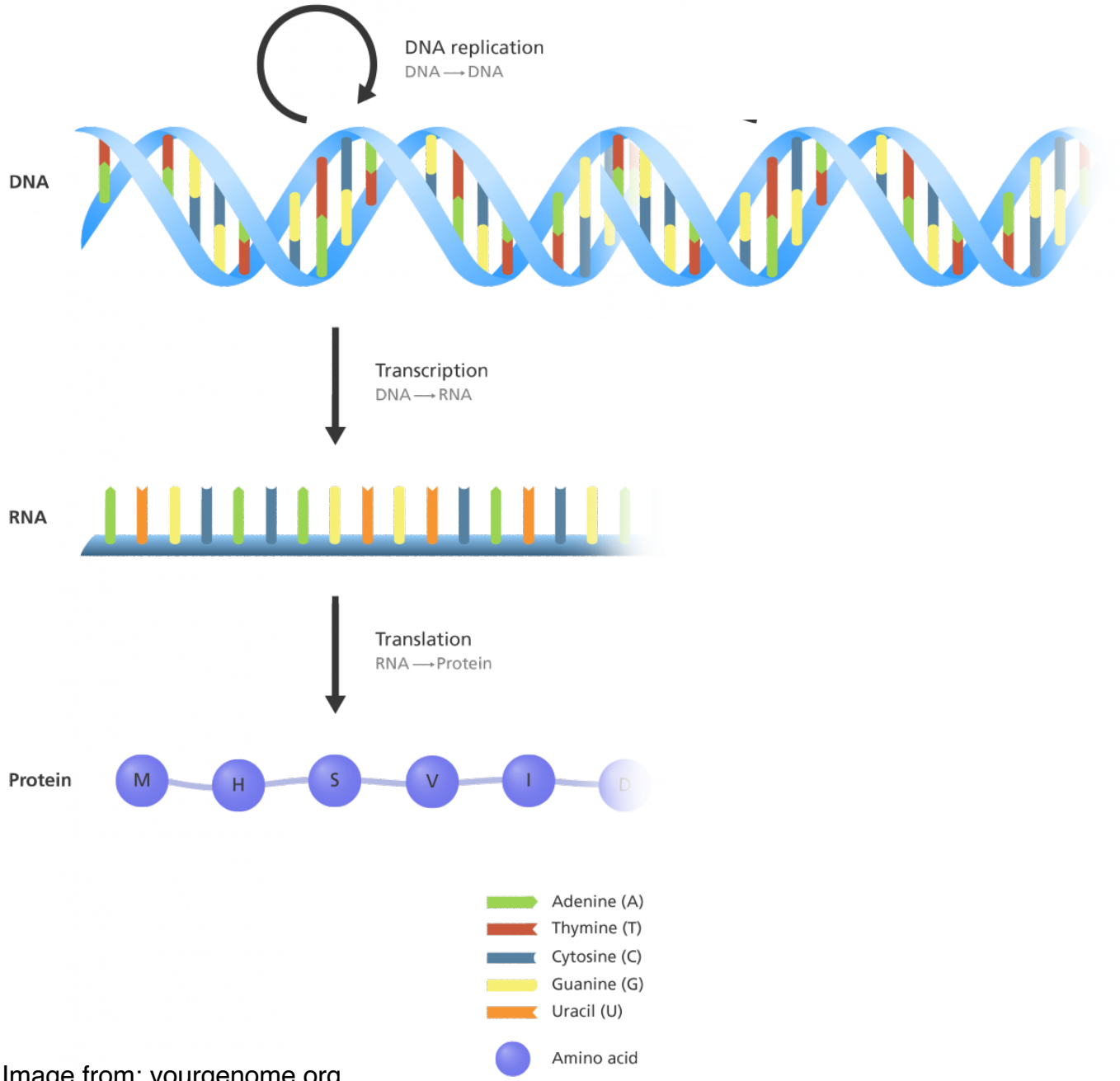


Image from: yourgenome.org

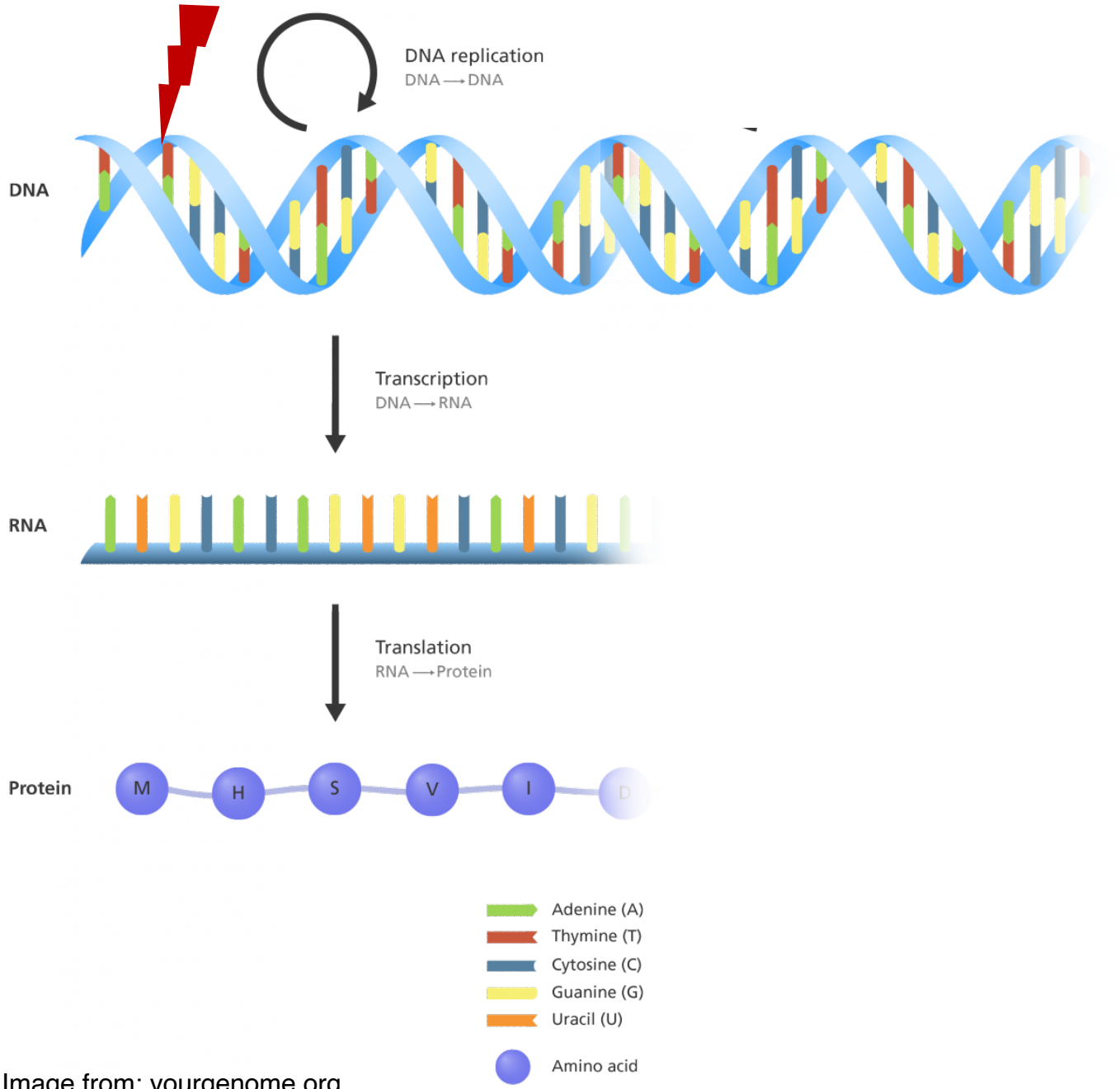


Image from: yourgenome.org

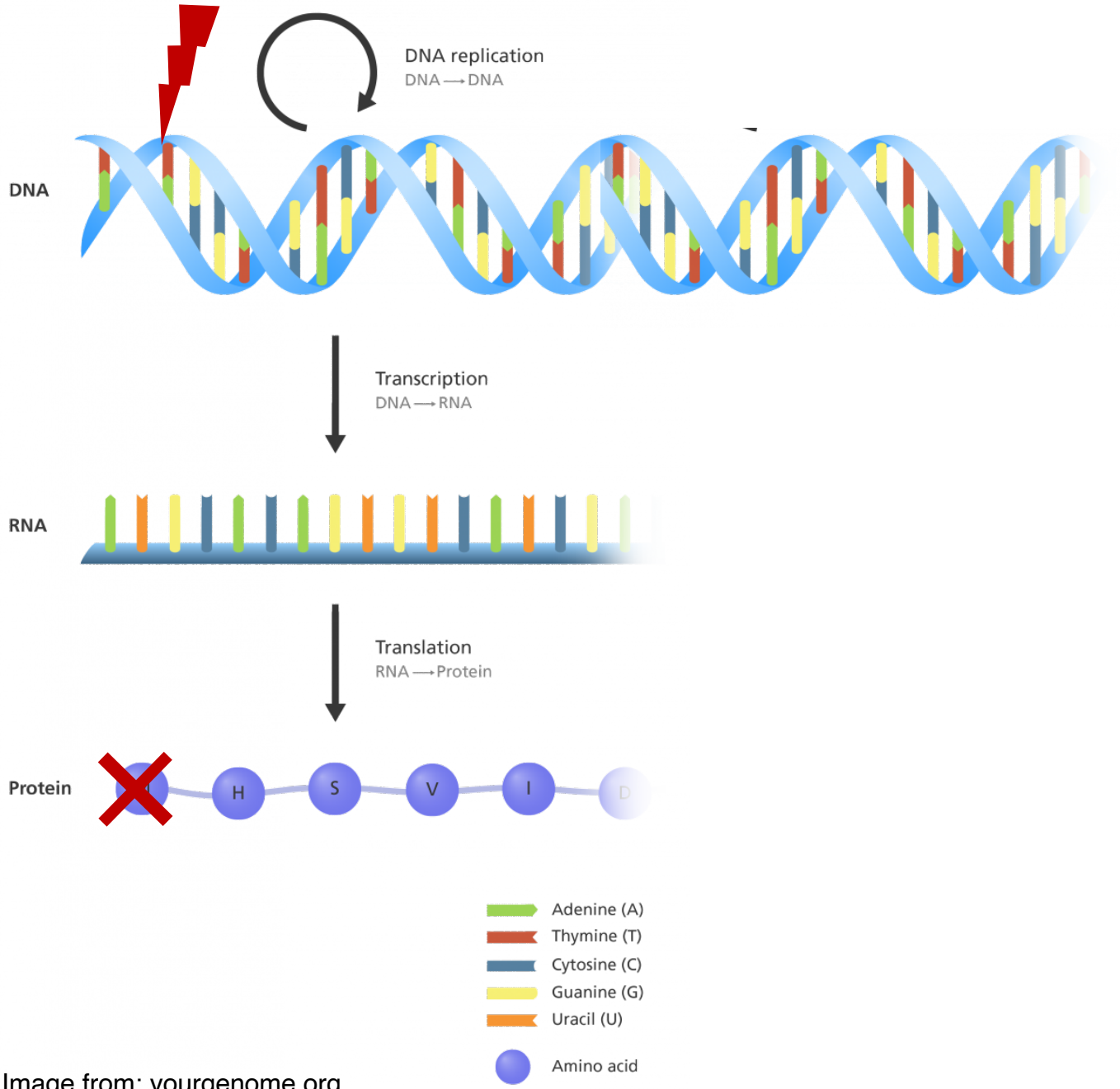


Image from: yourgenome.org

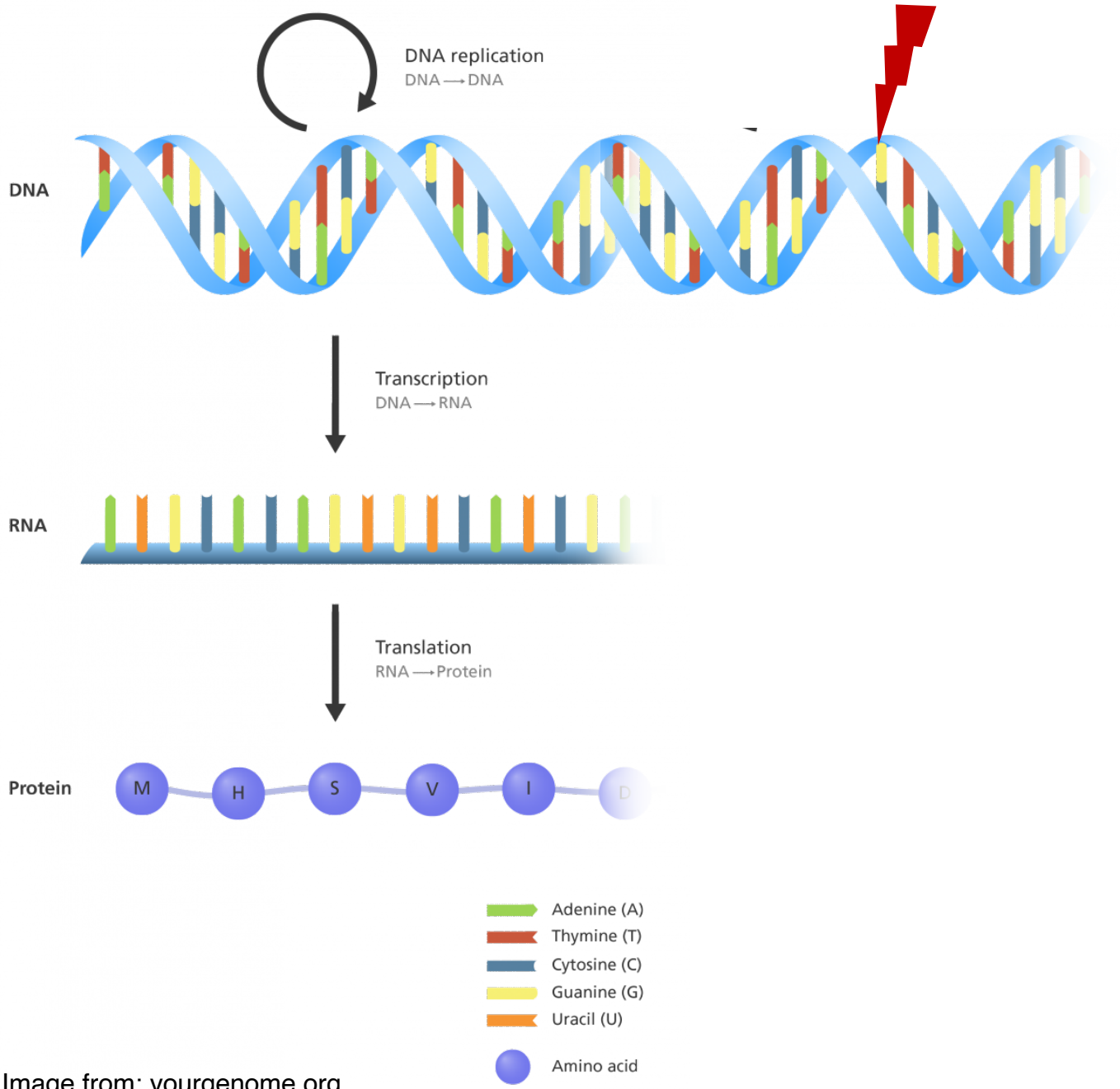


Image from: yourgenome.org

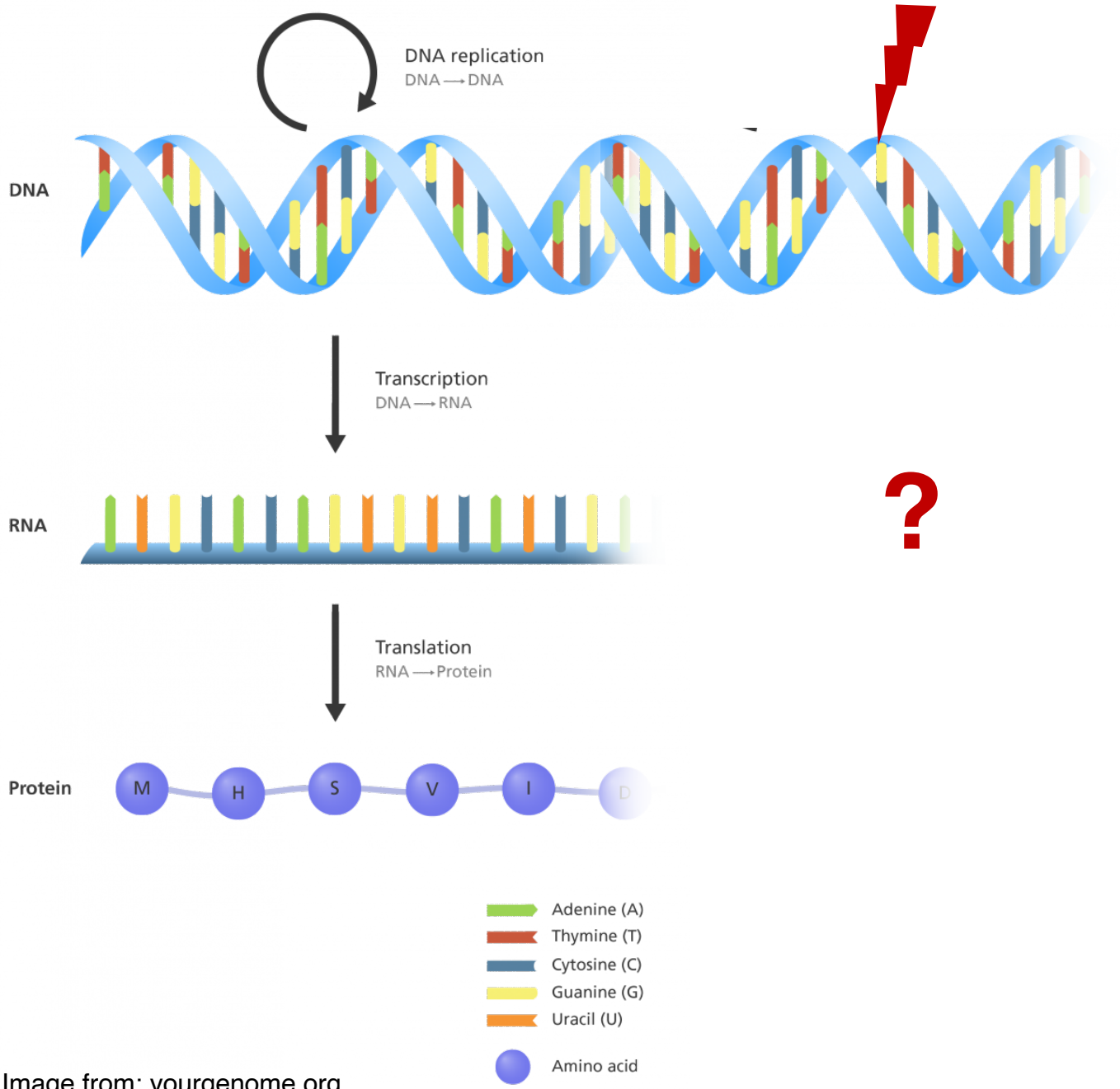
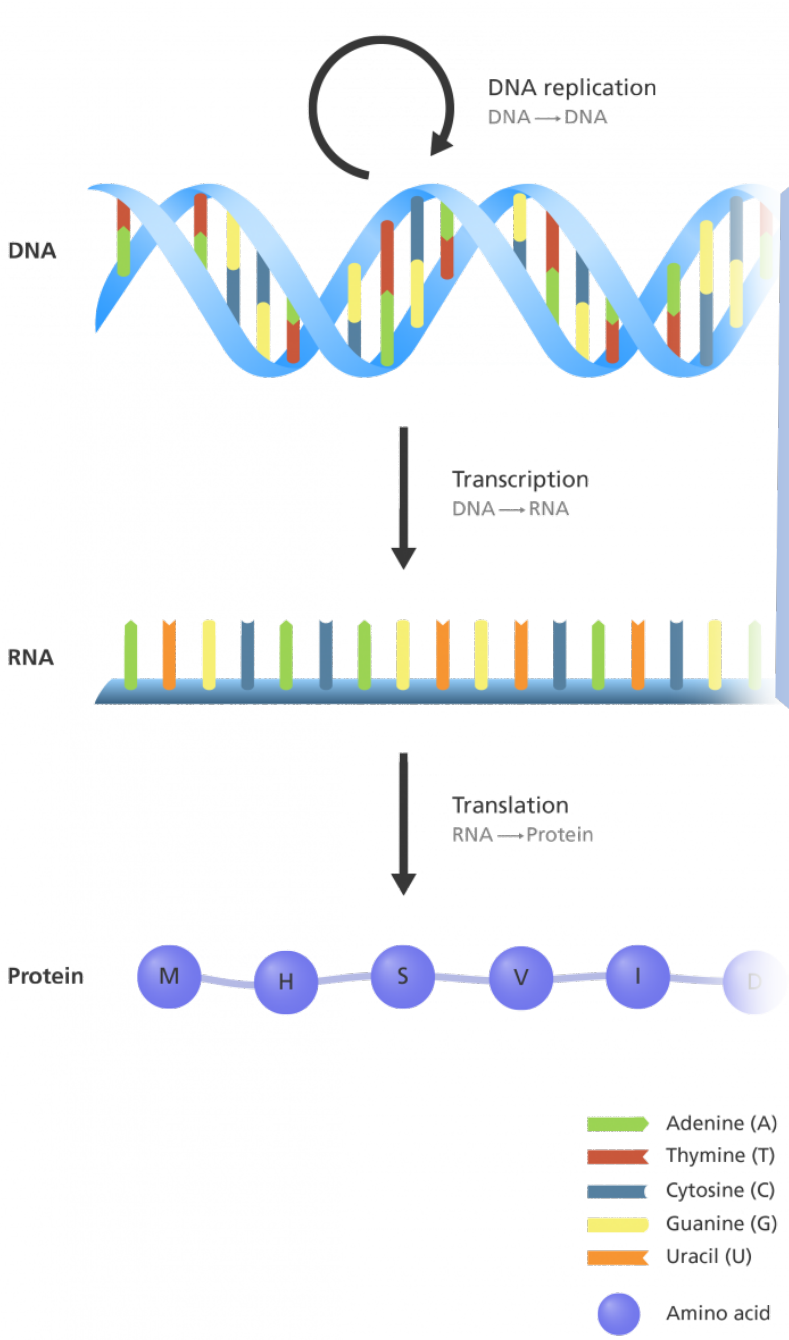
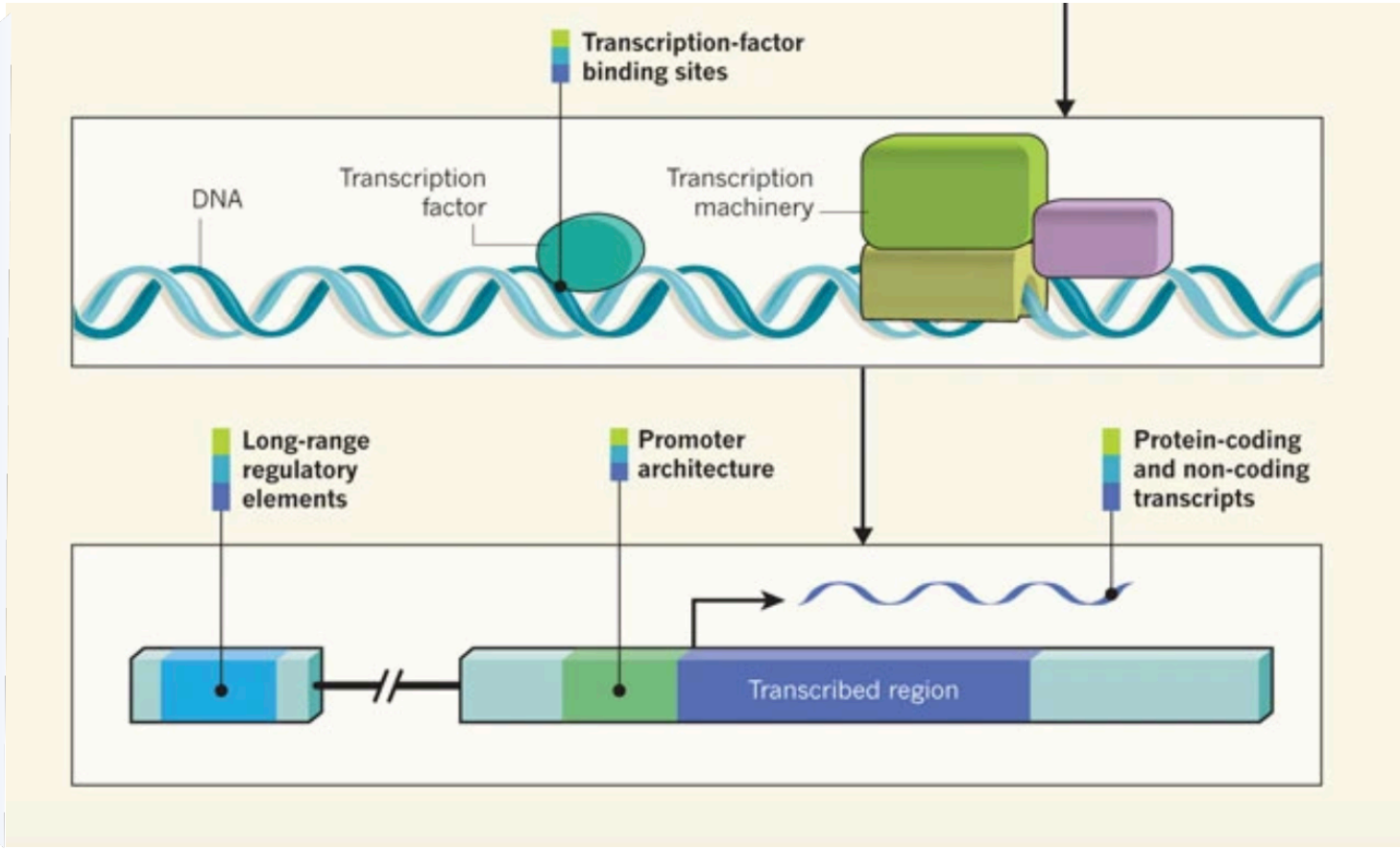


Image from: yourgenome.org





Ecker, J., Bickmore, W., Barroso, I. *et al.* ENCODE explained. *Nature* **489,** 52–54 (2012). https://doi.org/10.1038/489052a

Goal for paper

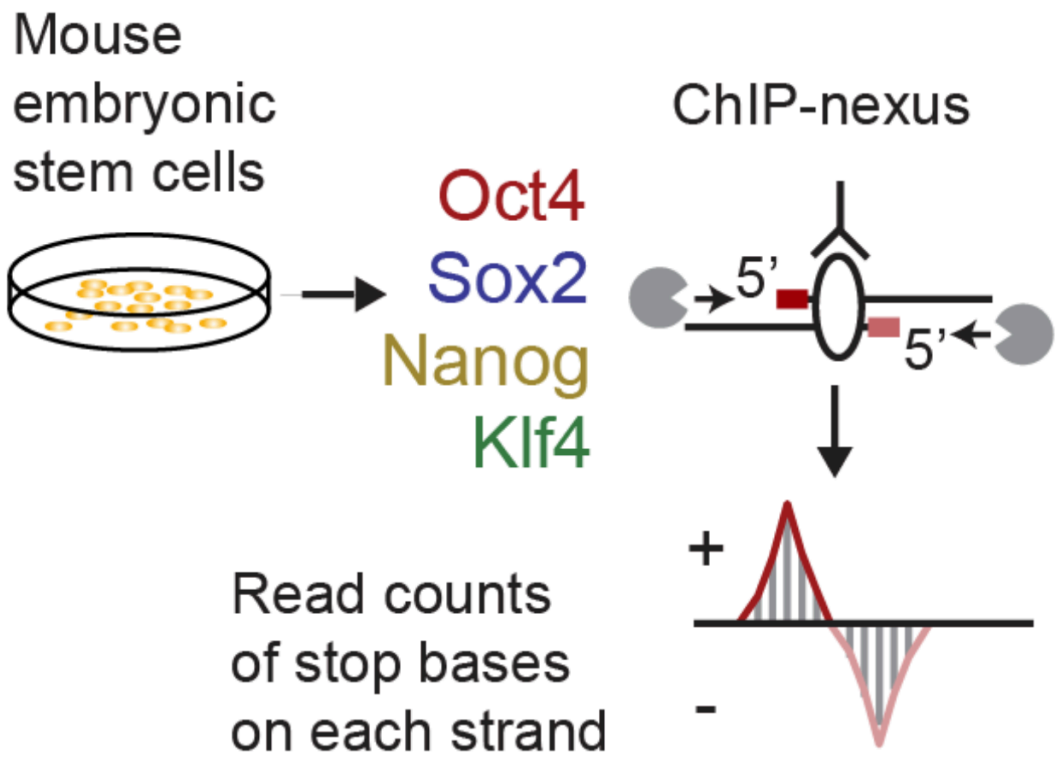
- Learn sequence motifs that are predictive of TF binding
- Learn the “syntax” (rules of arrangement) of motifs for TF binding
- Approach:
  - Train a neural network that takes as input sequence data and outputs TF binding profiles at base resolution
  - Using a combination of feature attribution and *in silico* mutagenesis, figure out what that neural network learned

Goal for my presentation

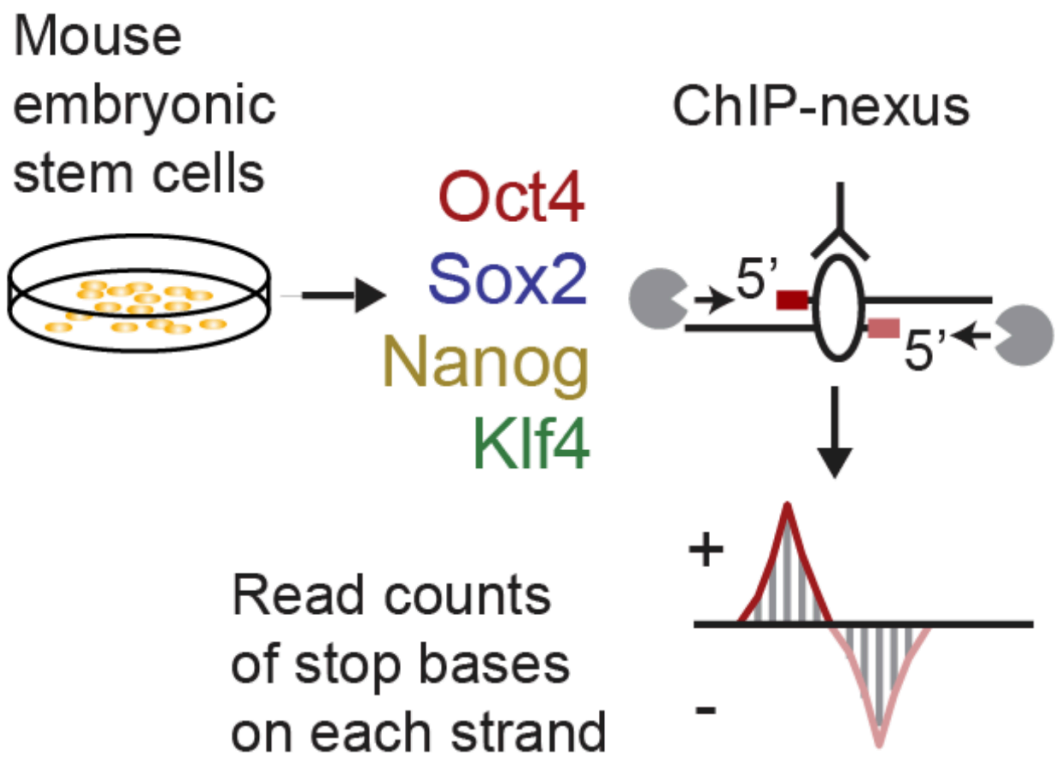
- Talk in detail about:
  - How their model is trained and evaluated
  - How feature attributions were generated
  - How interactions between motifs were found

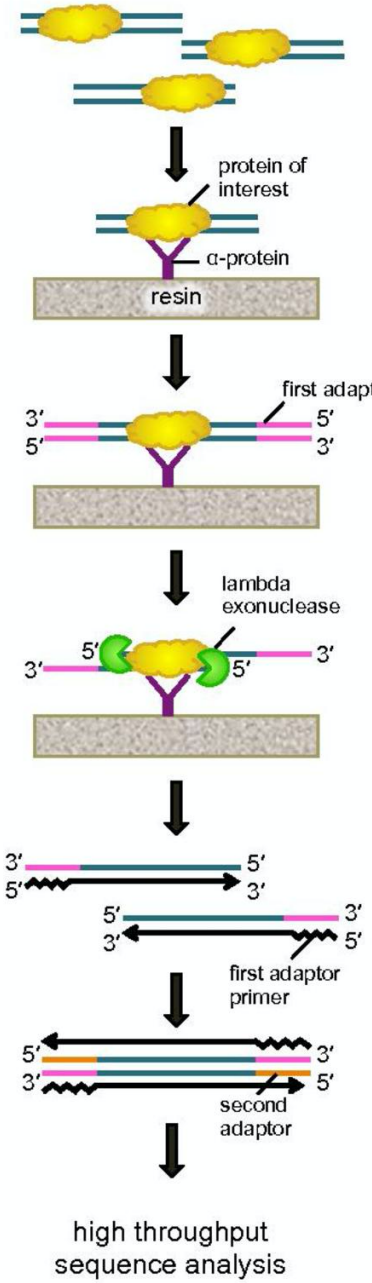
Figure 1
Predictive model

ChIP-nexus data for pluripotency TFs



ChIP-nexus data for pluripotency TFs





Crosslink proteins to DNA, shear DNA

Chromatin-immunoprecipitation

Ligated first adaptor, fill in ends

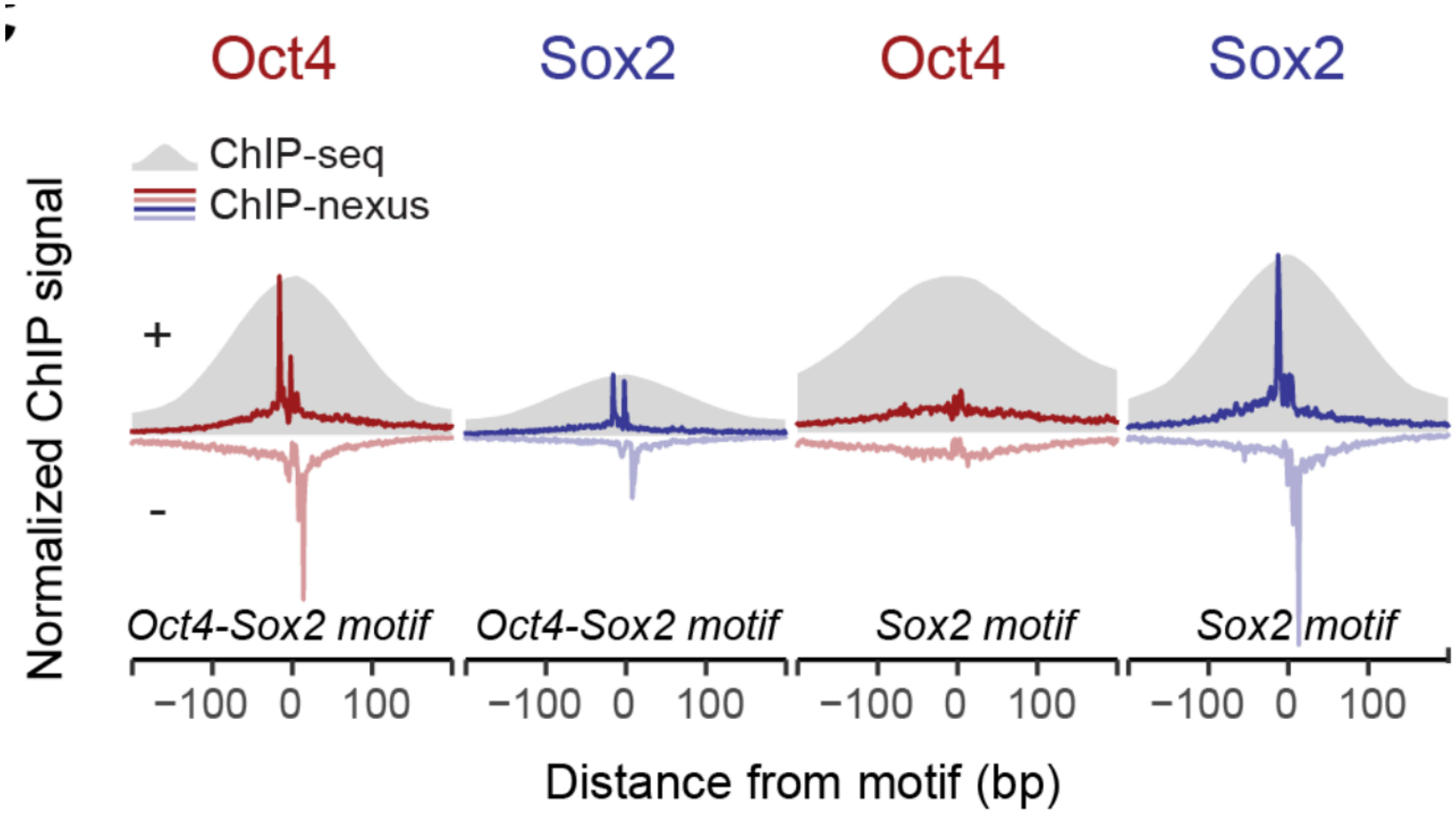
Exonuclease digestion

LM-PCR with primers to first adaptor

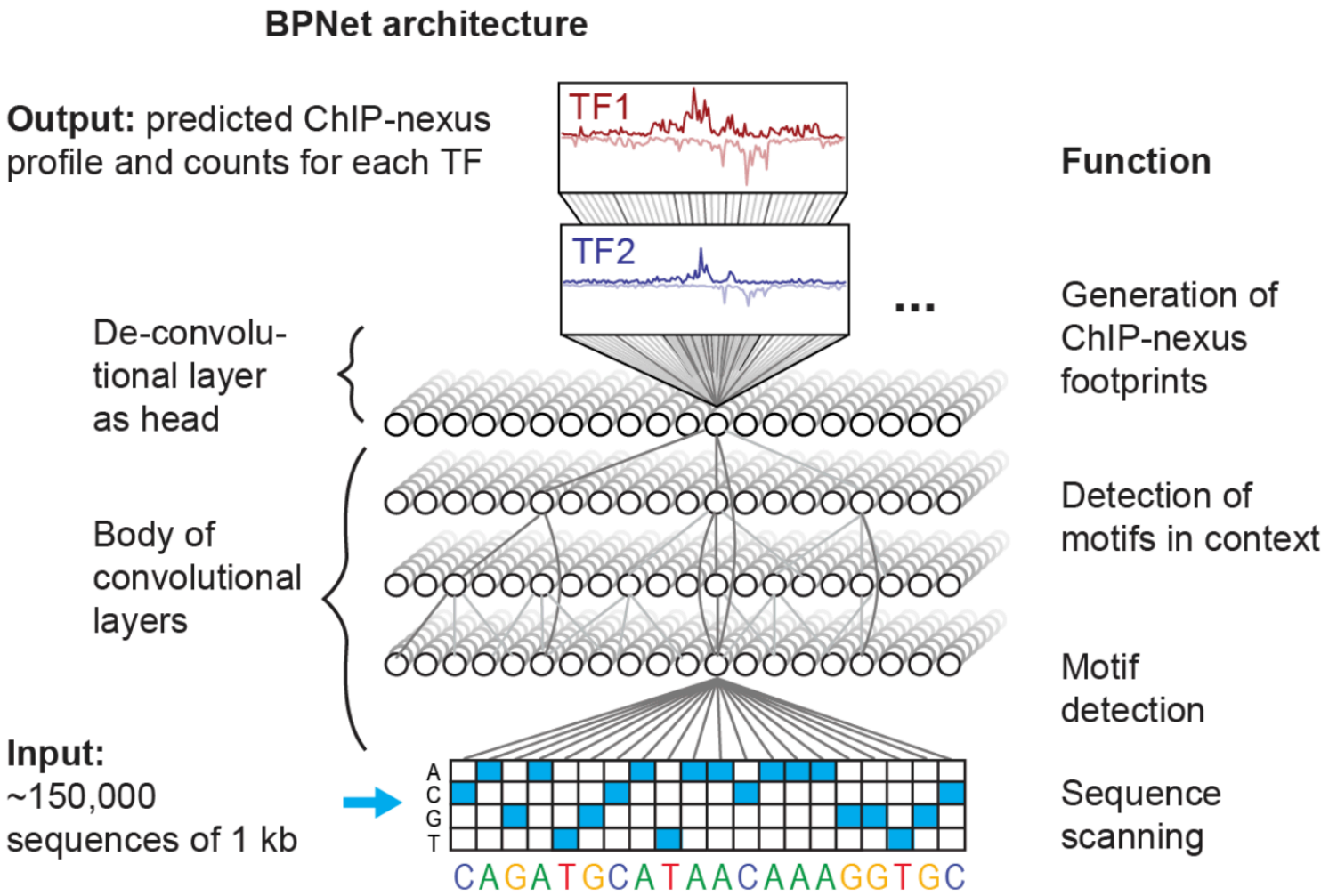
Ligate second adaptor, amplify DNA with LM-PCR

https://en.wikipedia.org/wiki/File:ChIP-exo_process_diagram.pdf

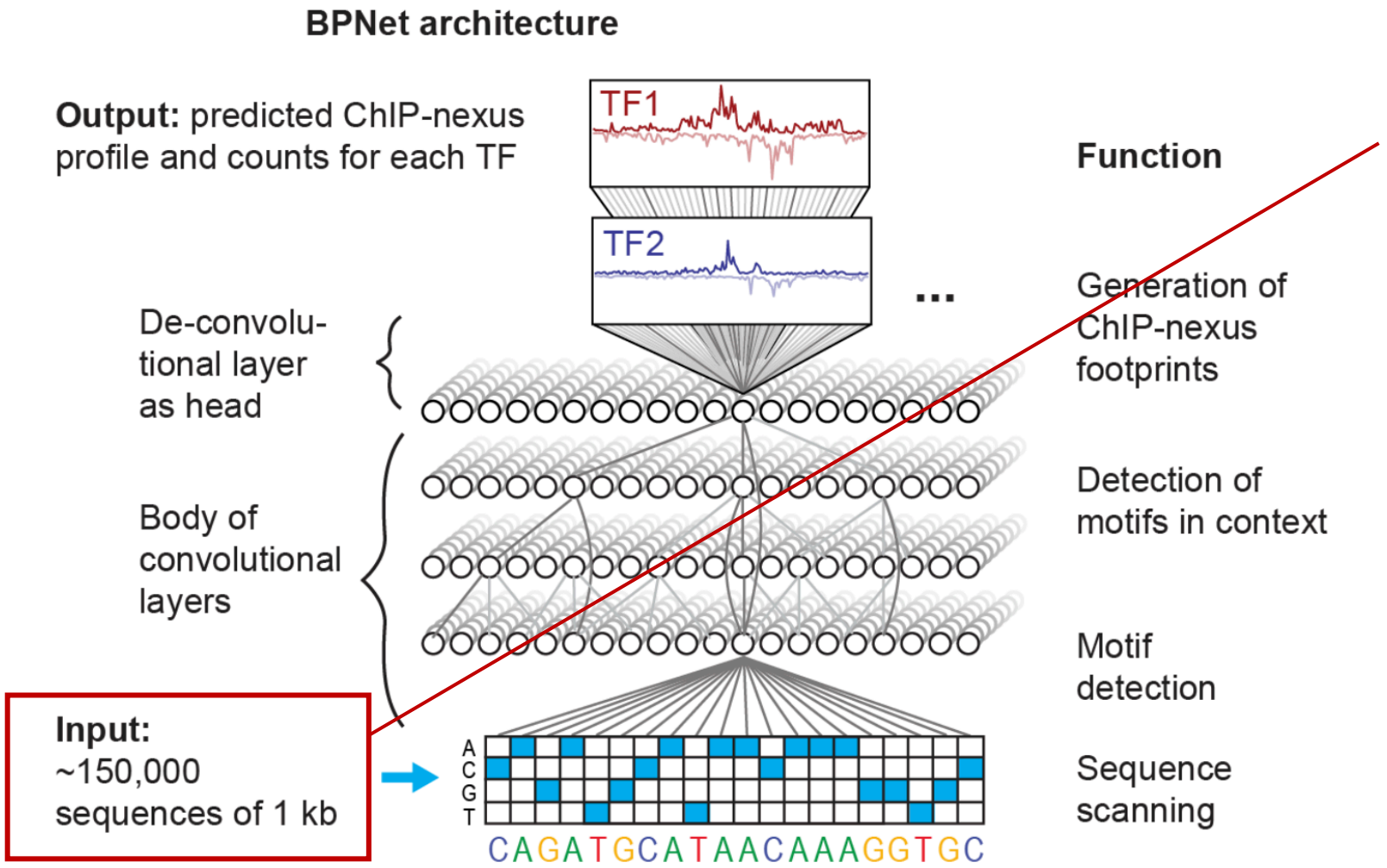
ChIP-nexus is higher resolution than ChIP-seq



BPNet: Base resolution conv net



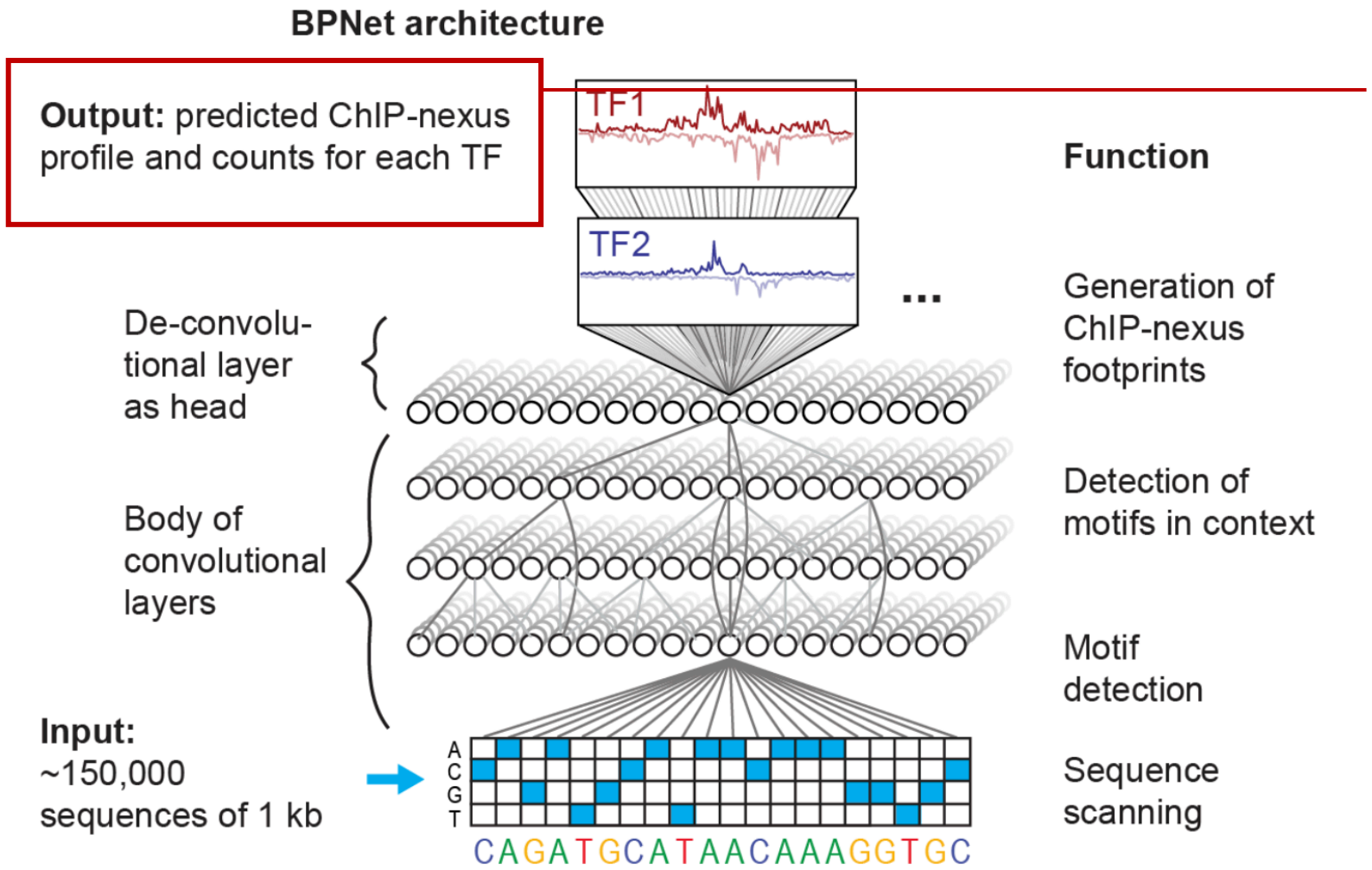
BPNet: Base resolution conv net



147,974 genomic regions w/ statistically significant & reproducible enrichment of ChIP-nexus signal for at least 1 of the 4 TFs

Is this the most reasonable population of genomic regions to use as training data? i.e. would it be better or worse to include regions where none of these TFs are bound?

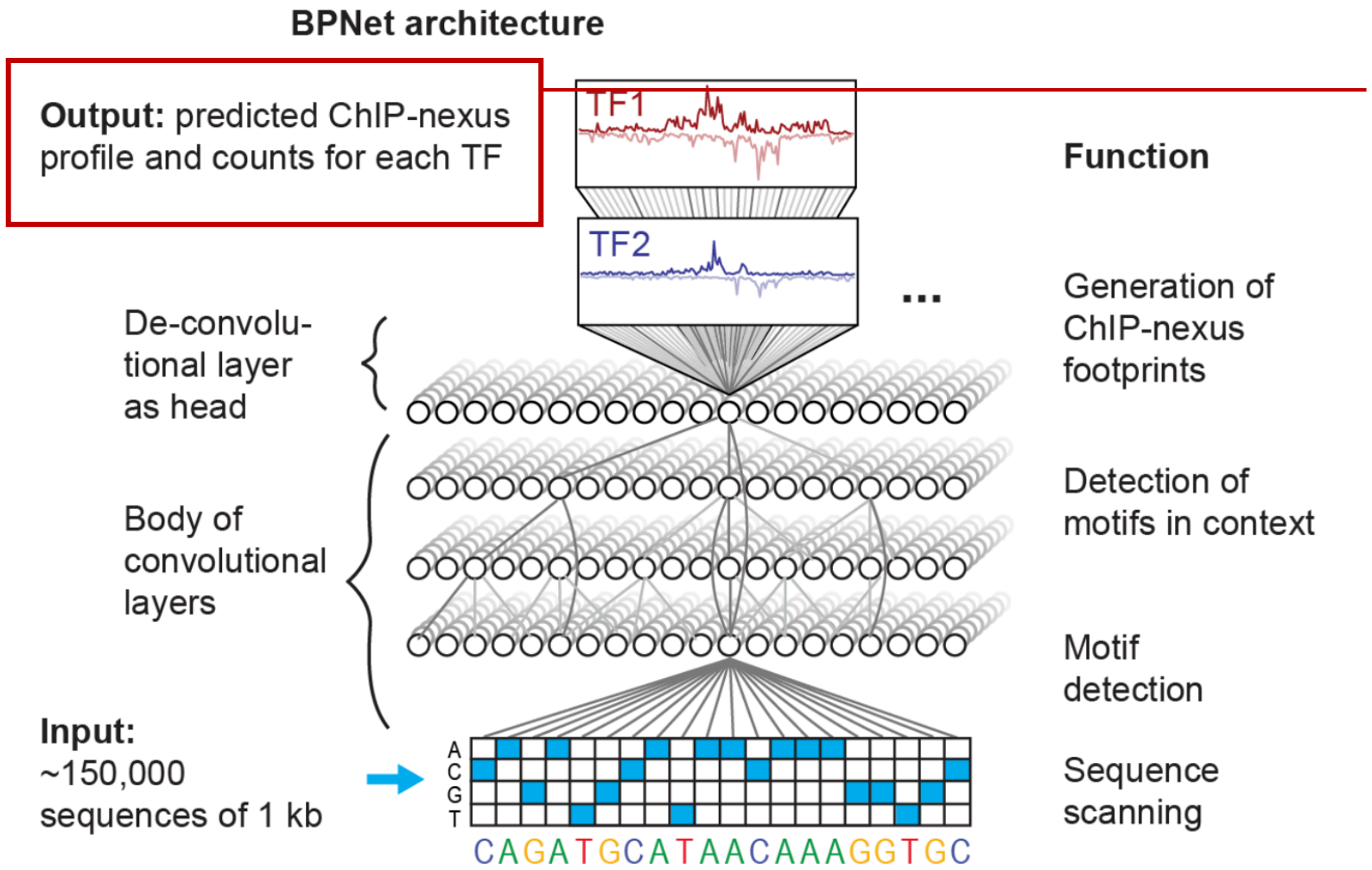
BPNet: Base resolution conv net



Multi-task prediction for 4 TFs

Maybe would have been interesting to see quantitatively how addition of each TF impacts model predictions for other TFs

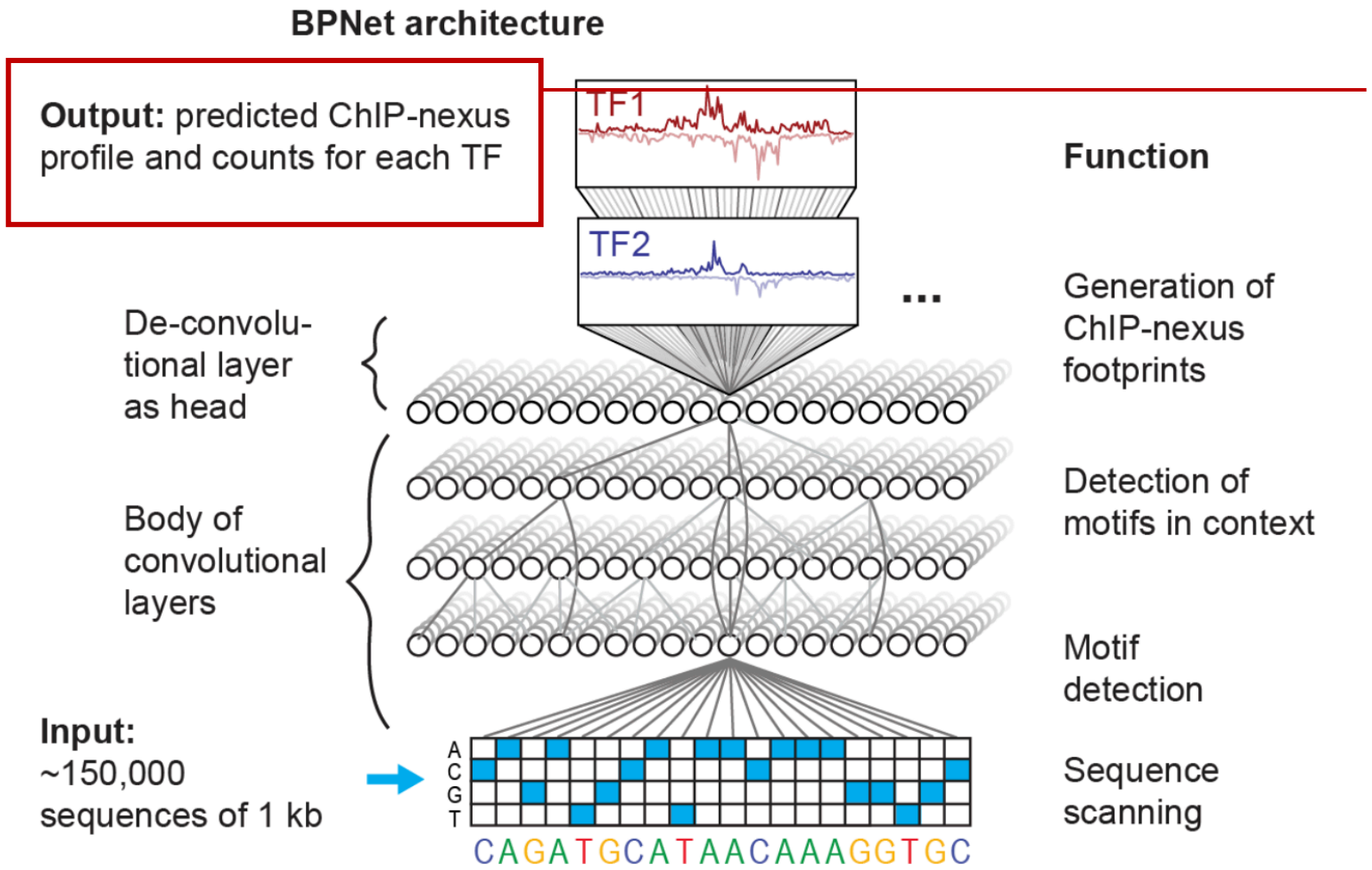
BPNet: Base resolution conv net



Output is actually factored into 2 heads per TF

- Total reads mapped to 1 kb region (mse loss)
- Profile shape (multinomial loss)

BPNet: Base resolution conv net

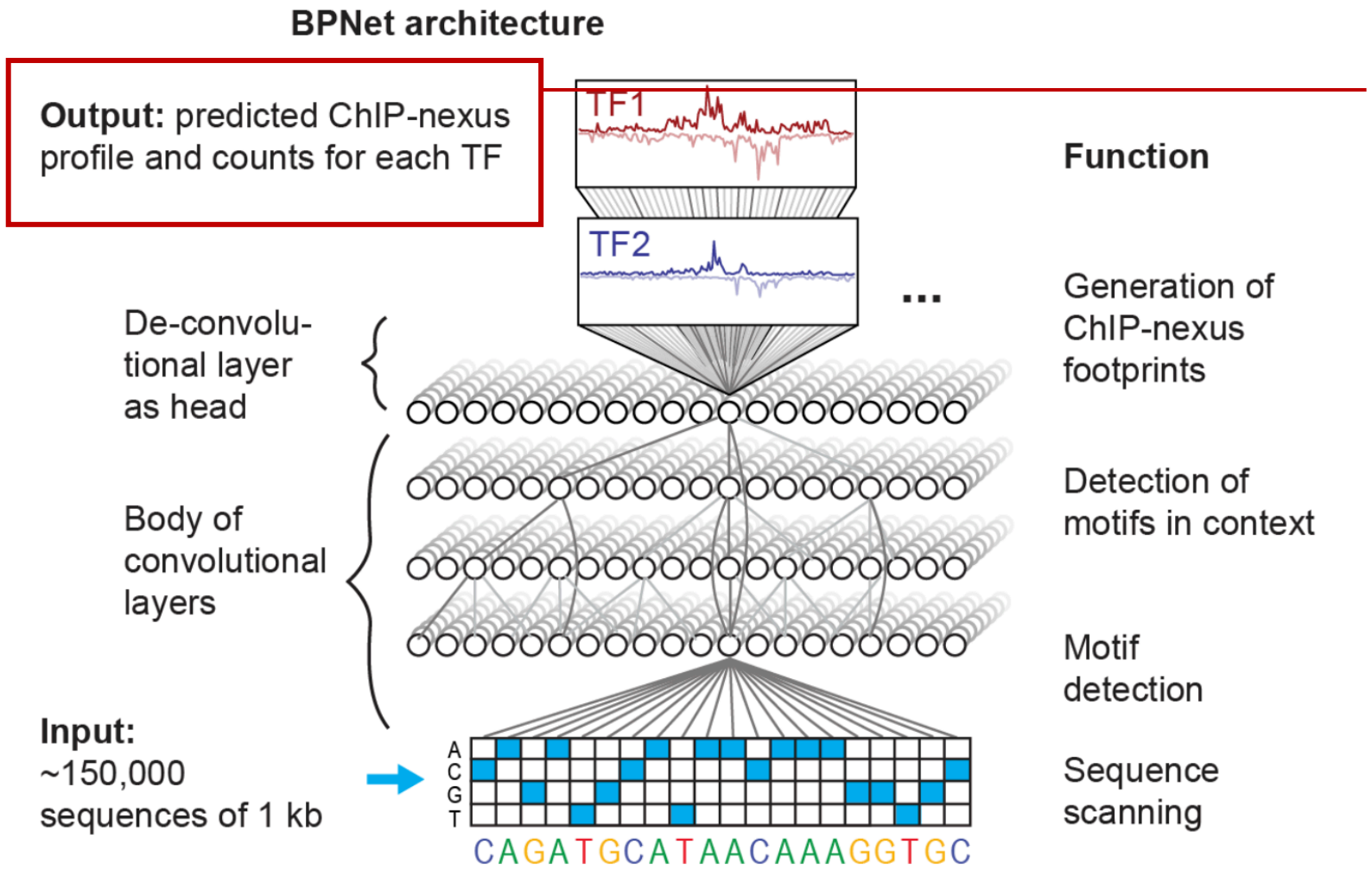


Output is actually factored into 2 heads per TF

- Total reads mapped to 1 kb region (mse loss)
- Profile shape (multinomial loss)

Assume you have k independent Poisson-distributed random variables $(X_1, \ldots, X_k)$ each with different means λ_k. Given the total number of counts, $n = X_1 + \ldots + X_k$, the conditional distribution of $(X_1, \ldots, X_k)$ is given as Mult(n, π), where π is just the vector of Poisson parameters normalized to sum to 1.

BPNet: Base resolution conv net



Output is actually factored into 2 heads per TF

- Total reads mapped to 1 kb region (mse loss)
- Profile shape (multinomial loss)

Assume you have k independent Poisson-distributed random variables $(X_1, \ldots, X_k)$ each with different means λ_k. Given the total number of counts, $n = X_1 + \ldots + X_k$, the conditional distribution of $(X_1, \ldots, X_k)$ is given as Mult(n, π), where π is just the vector of Poisson parameters normalized to sum to 1.

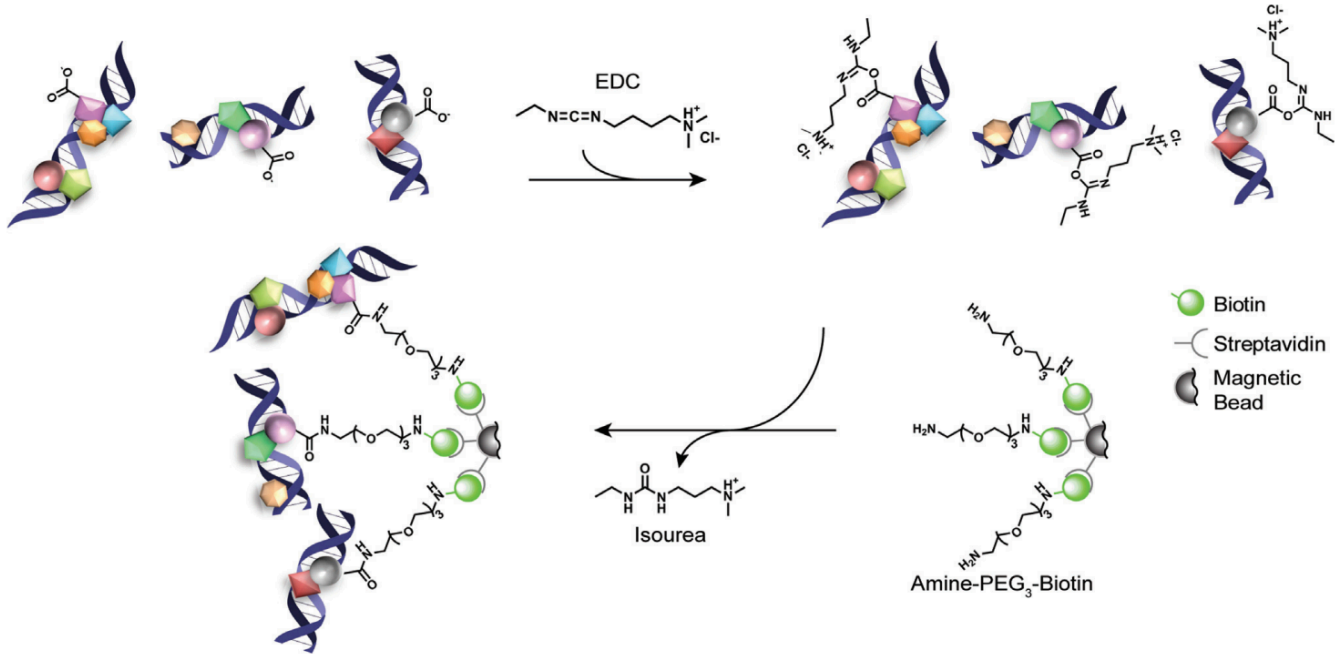
They up-weight the profile loss

Bias control

To account for experimental artifacts, analysis of ChIP-seq data relies on control experiments

Isolate cellular DNA, crosslink, but either use IgG or whole cell extract

PAtCh-Cap: protein attached chromatin capture



Bias control

To account for experimental artifacts, analysis of ChIP-seq data relies on control experiments

Isolate cellular DNA, crosslink, but either use IgG or whole cell extract

PAtCh-Cap: protein attached chromatin capture

Actual model fit is:
$y = f_{model}(seq) + f_{ctr}(ctrl\ track)$

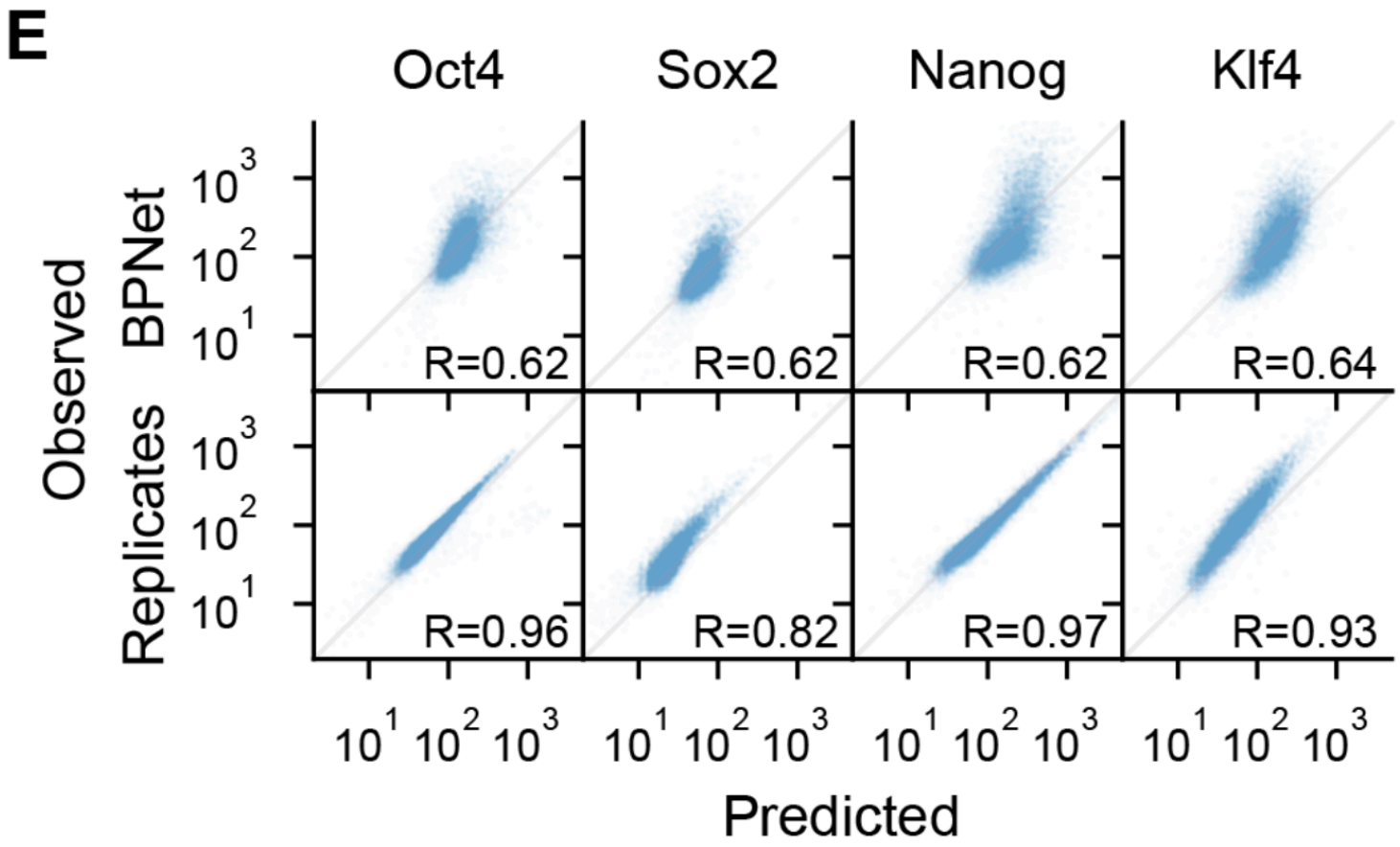
For the total counts heads, the control model is just a scalar weight times the log of the total number of counts in the control track

For the profile head, the control model is a weighted sum of the raw counts from the control track and smoothed version of the control track (50bp sliding window)

Jointly optimized

Evaluation

- For total counts, they just look at spearman R (Sup. Fig. 2)

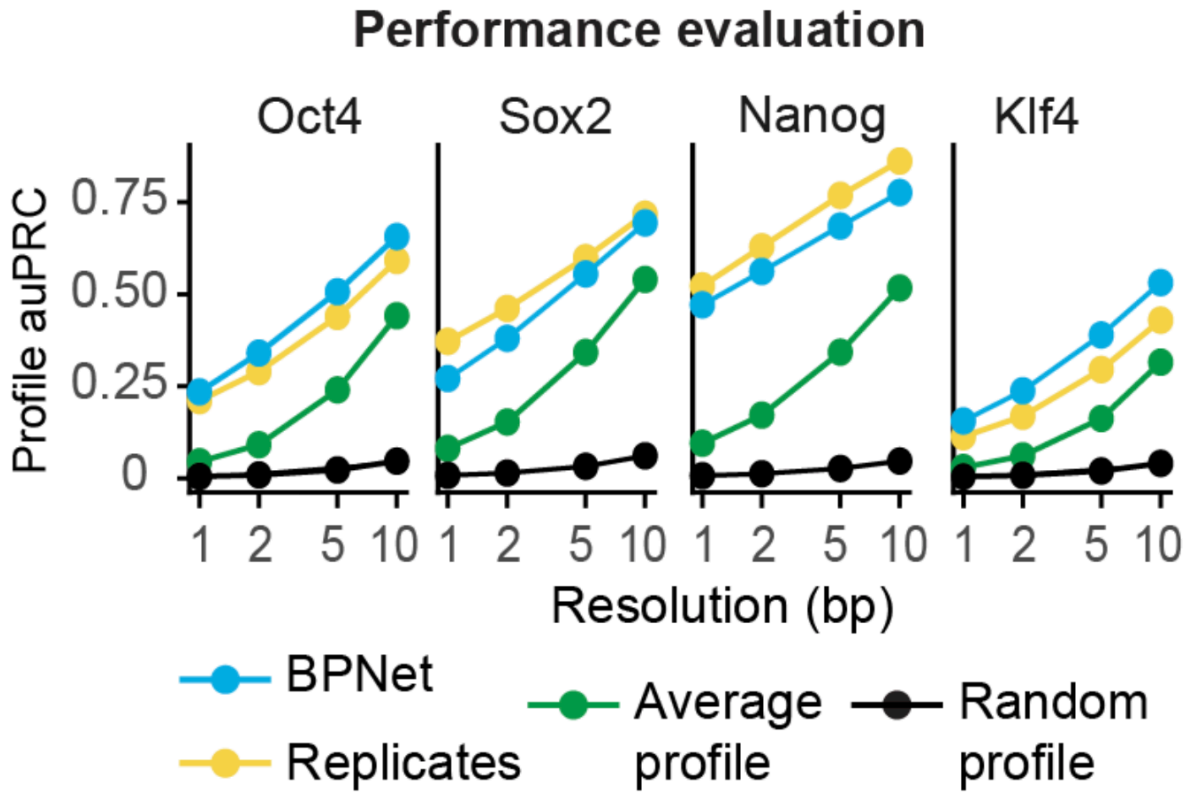


Evaluation

- For profile shape, they think of each bin as a binary classification problem: does shape of profile correctly identify high- and low-count bins
- Each base pair was labeled as positive if it had > 1.5% of the total reads in the 1kb region, and negative if it had < 0.5% of the total reads in the 1kb region
  - Thresholds manually determined by visual examination
  - Why not just CV?
- Then binned at different resolutions (2bp – 10bp)
- A bin was called positive if any bp in the bin had a positive label, negative if all bps were negative, and ambiguous otherwise
- For predicted probabilities, they used the max over the bin

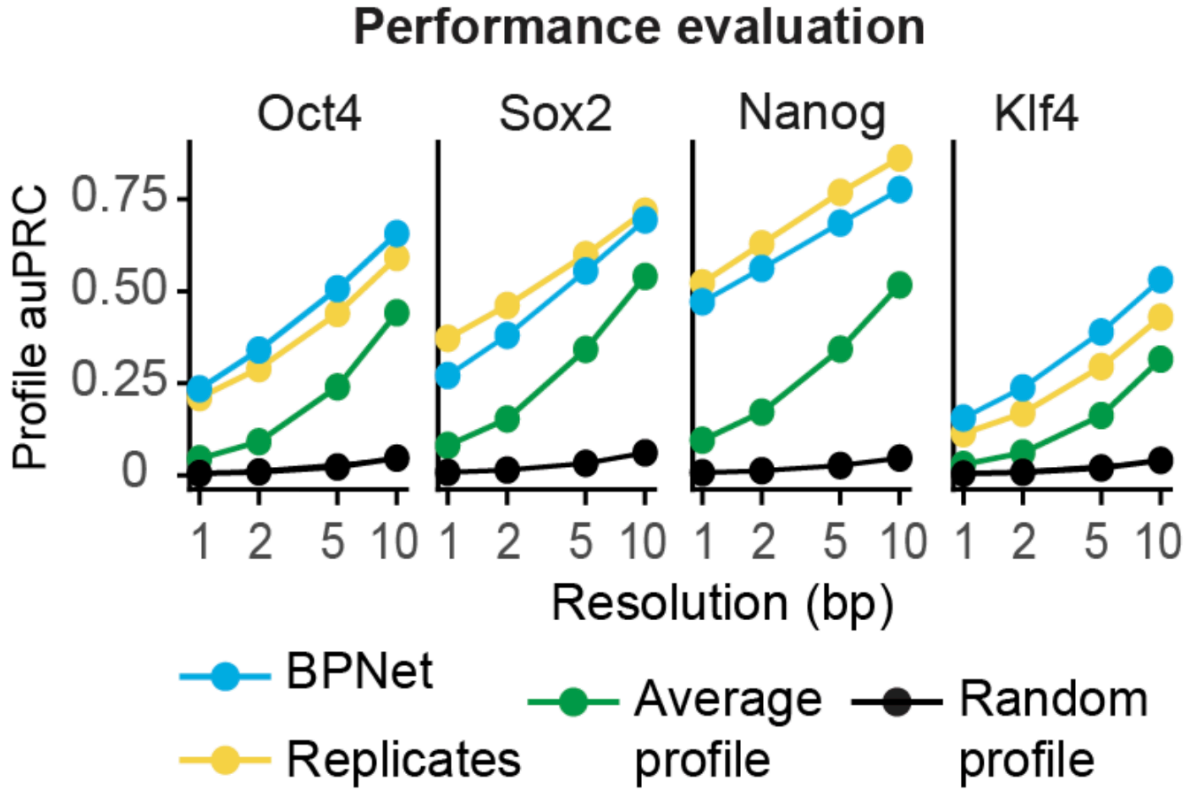
Evaluation

- BPNet achieves replicate level performance at this metric
- Random profile is generated using shuffled regions
- They don't really mention the what the average baseline is, other than saying that "The positional concordance was on par with replicate experiments and substantially better than randomized profiles or average profiles at resolutions ranging from 1-10 bp"

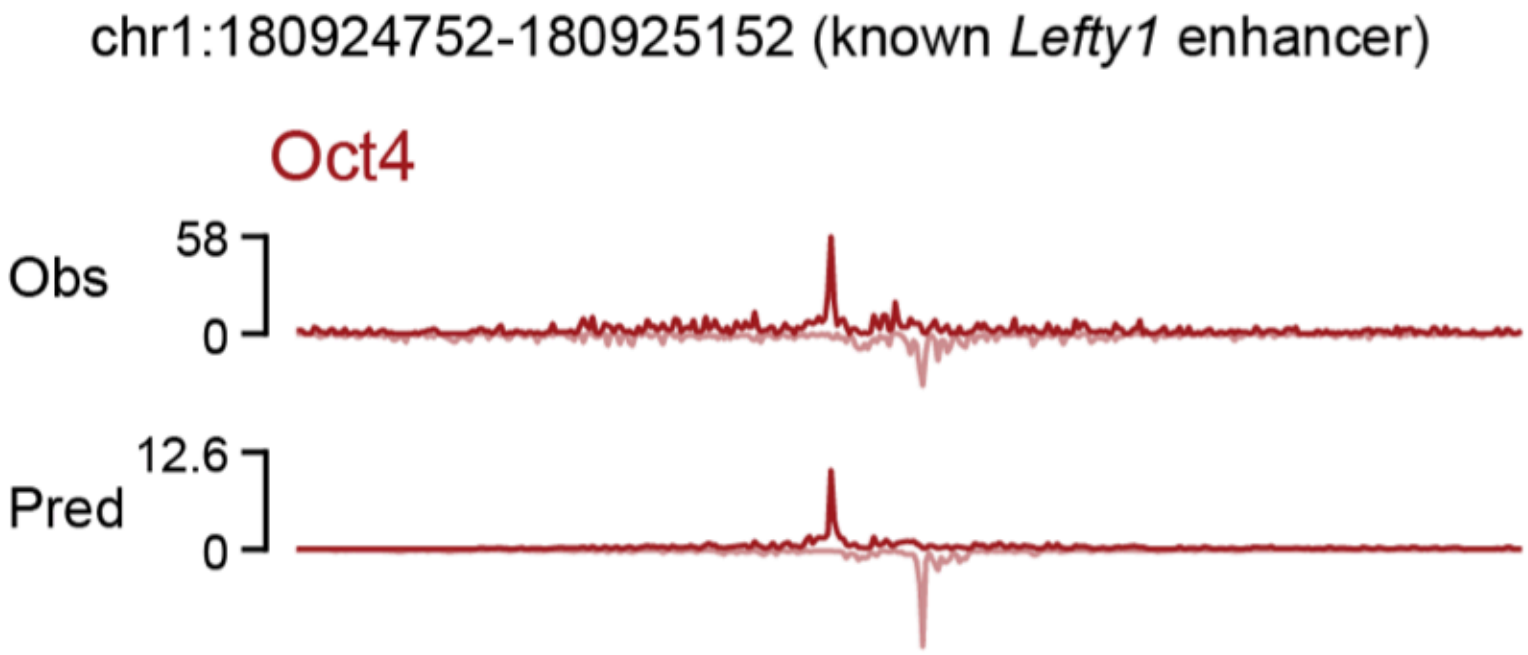


Evaluation

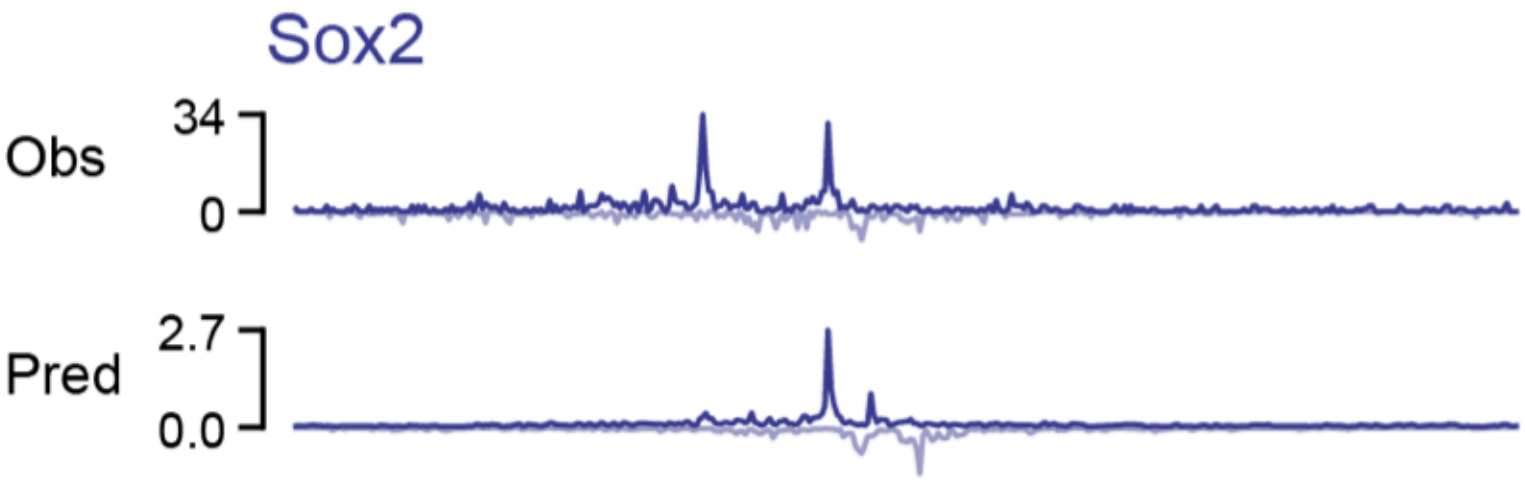
- From looking at the code, I think average profile is the average profile for each TF over all regions tested, but I'm not 100% sure
- What performance would you get if you did average positive profile and average negative profile for each TF and applied those either w/ the ground truth for whether the region is bound or w/ the model's prediction of whether the region is bound?
- Uncertainty measures for these points? You can see that sometimes BPNet is visibly above replicates the same amount that replicates is above average profile (see Klf4)



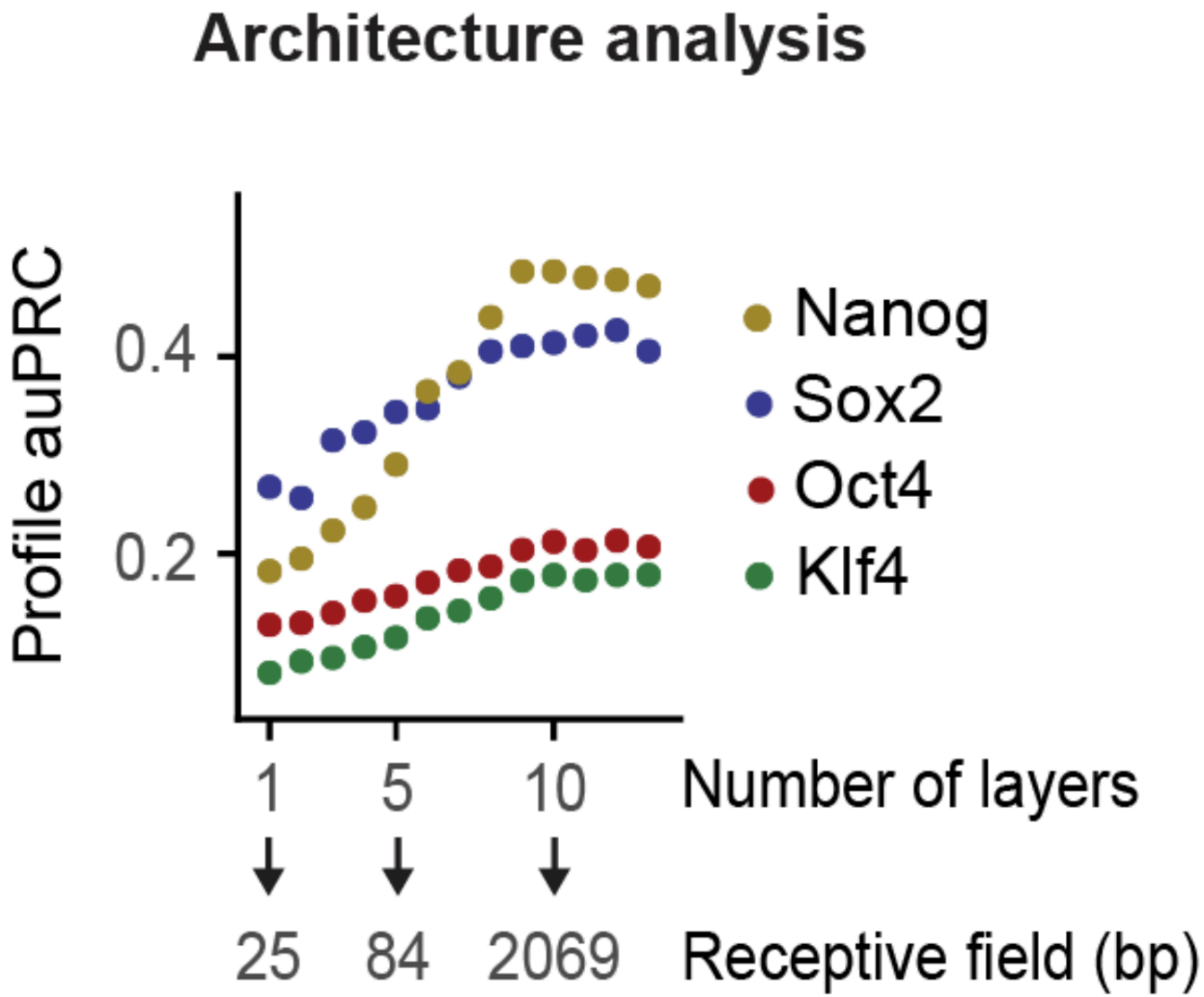
Predictions qualitatively look good



Predictions qualitatively look good

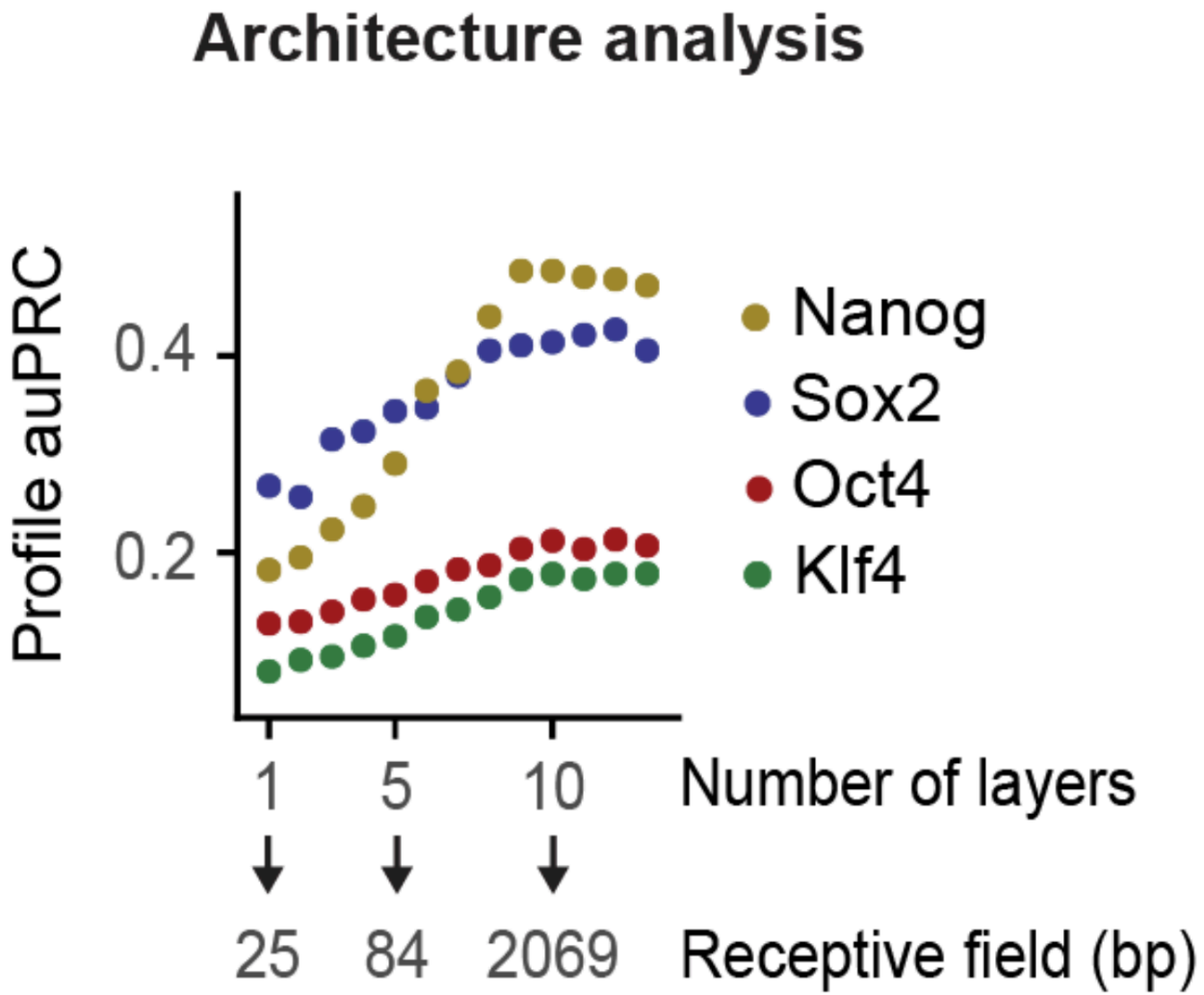


Receptive field size is important for Nanog



(For each position in the predicted profile, how many input bases are considered in the input)

Stacking more layers improves performance



- Does improvement stop at input sequence length?
- If input sequence length were longer, would receptive field continue to add performance? Like, what is the reasonable length of receptive field?
- Basically, I'm not necessarily convinced that stacking more layers improves performance because there are complex, compositional giant motifs and not just because the deeper res-net optimizes more easily or something?

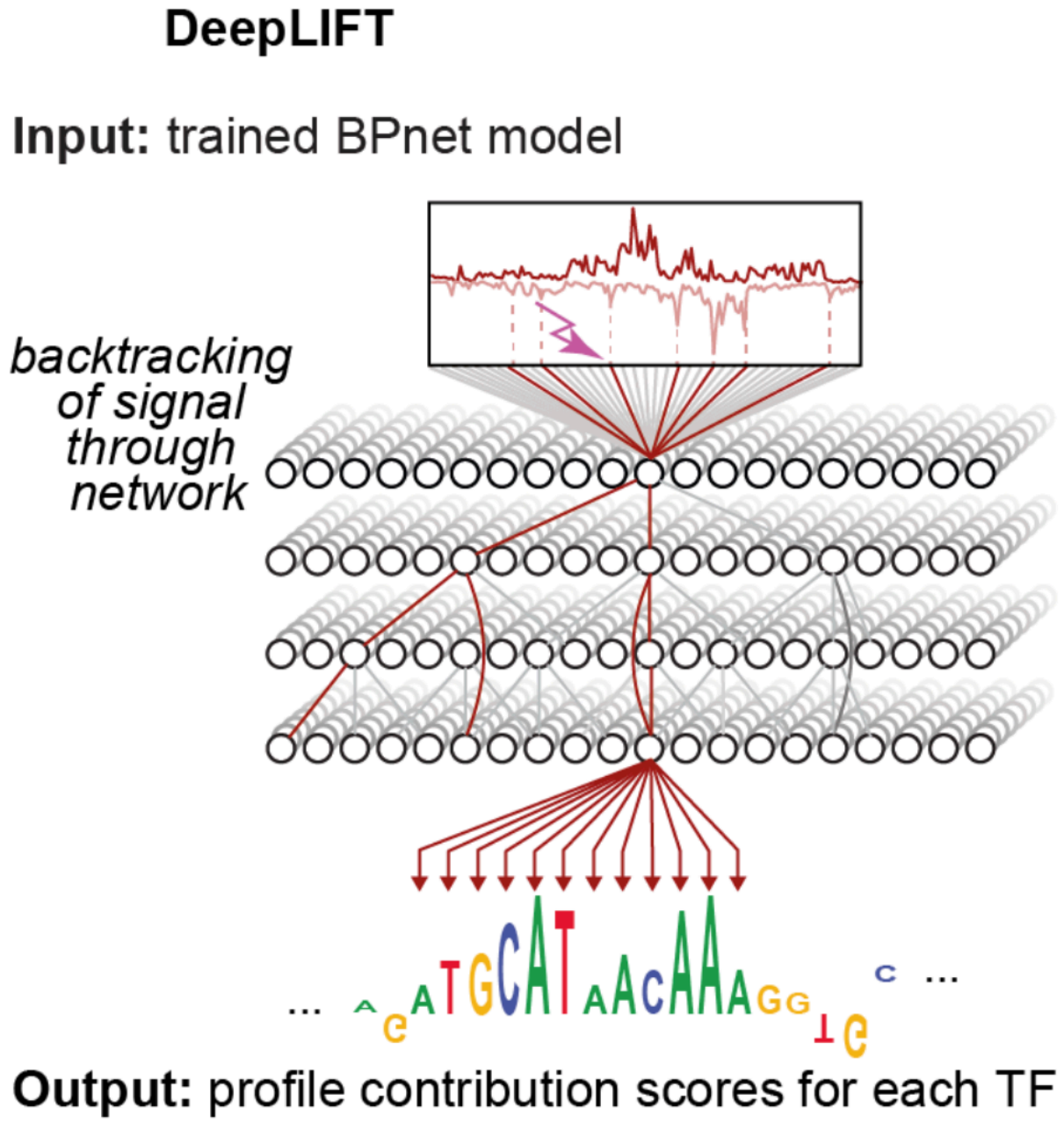
Figure 2
Model interpretation

Feature Attribution

- Find importance of input features in terms of output prediction
- Model output will be the sum of the feature attributions
- For a linear network, the contribution of each feature would just be:

$$(x_i - bi) * \prod w$$

- For non-linear networks, you calculate the (approximate) Shapley value for each non-linearity encountered and back-propagate it back through linear components



Feature Attribution

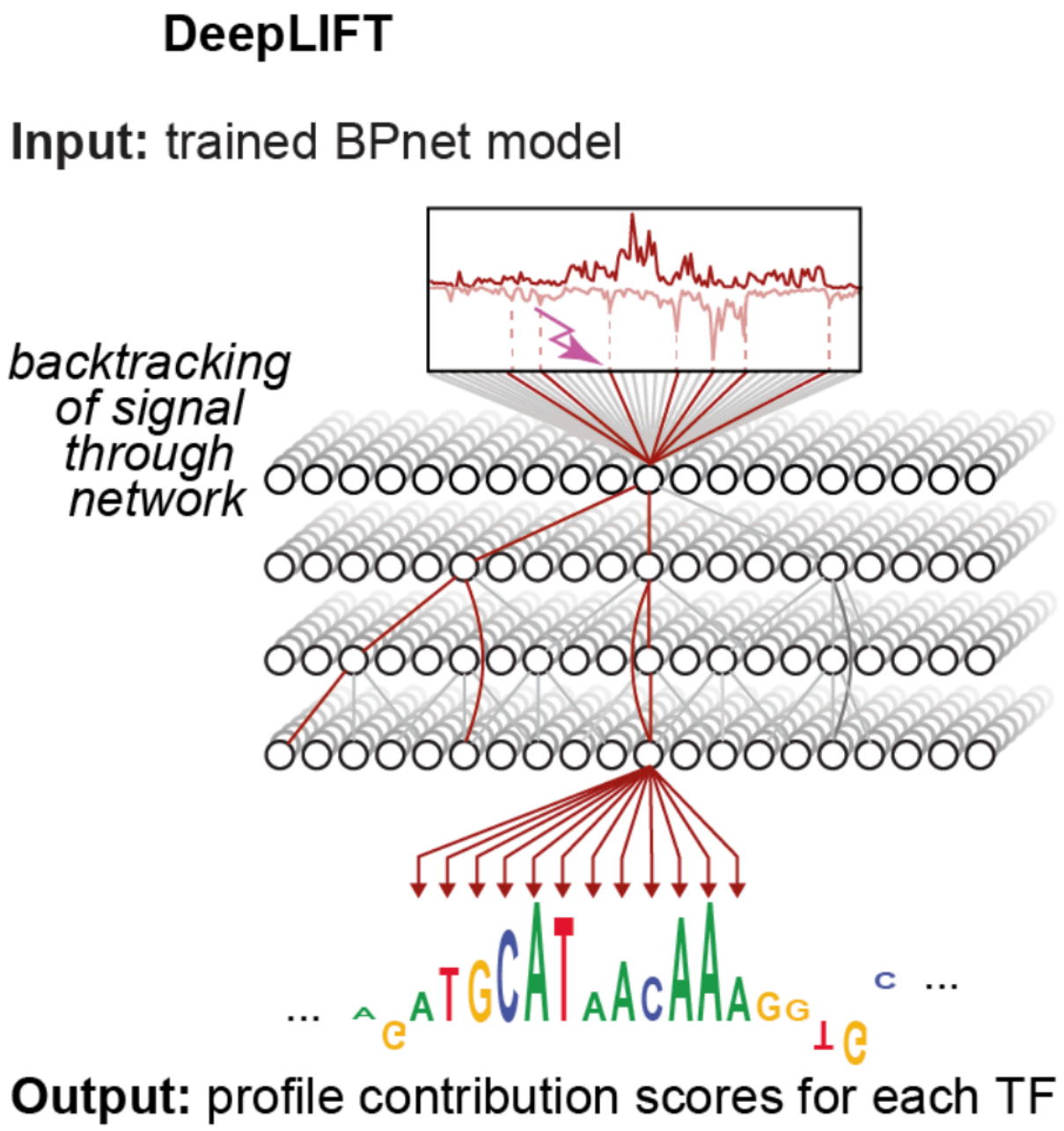
- DeepLIFT divides a scalar output between each of the contributing input features
- How to get the importance for an entire profile (L x S matrix, where L is 1kb, S is 2 strands)
- Scalar attributions for a base:

$$f(x) - f(b) = \sum_{i}^{D} c_i$$

- Profile attributions for a base:

$$c^{(profile),i} = \sum_{j,s} c_{js}^{i}\, p_{js}$$

where c_{js}^{i} is the DeepLIFT attribution for input sequence position *i* to output position *j* on strand *s* and p_{js} is the *j,s* index of **p** = softmax(f(**x**))

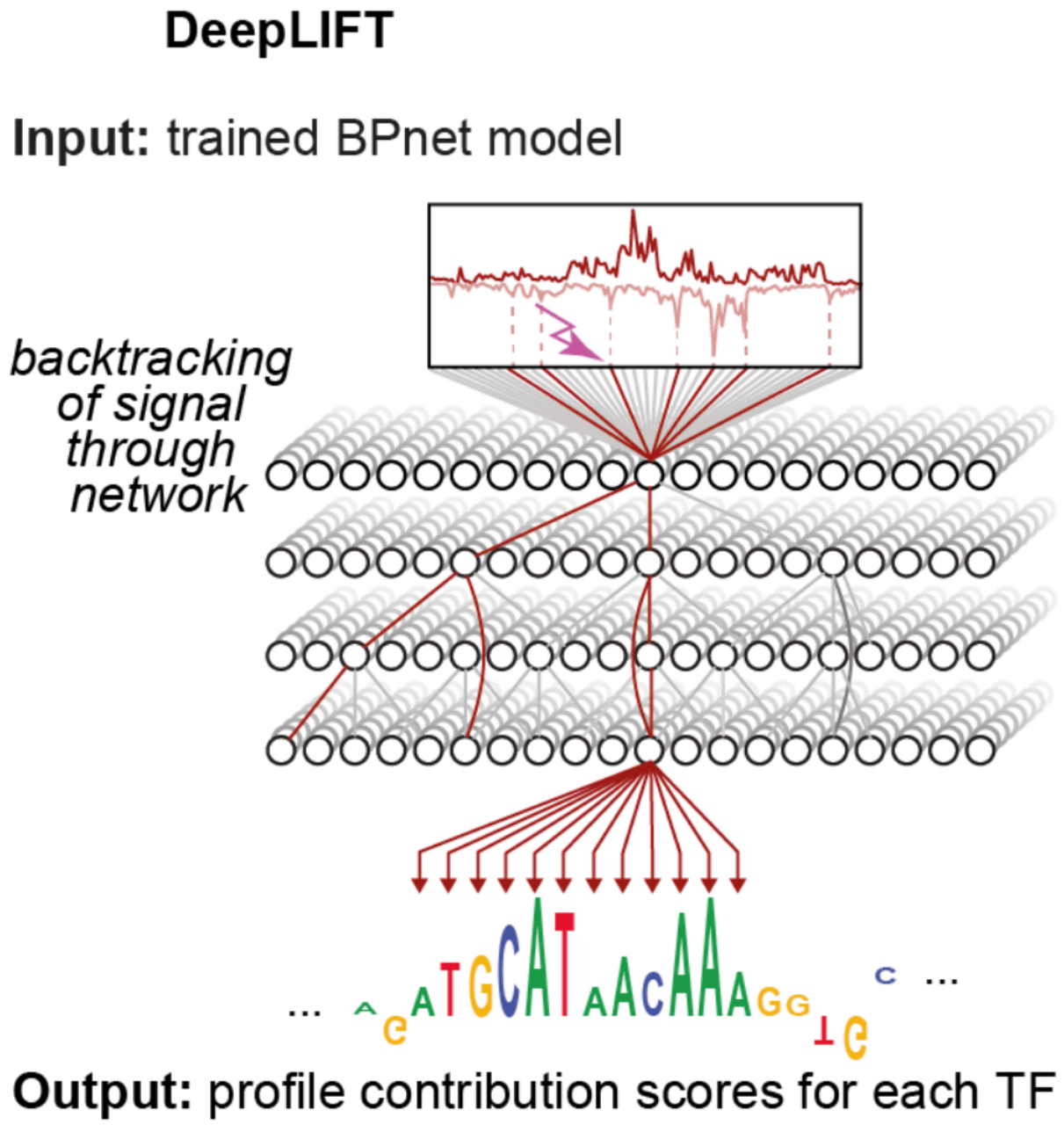


Feature Attribution

- Profile attributions for a base:

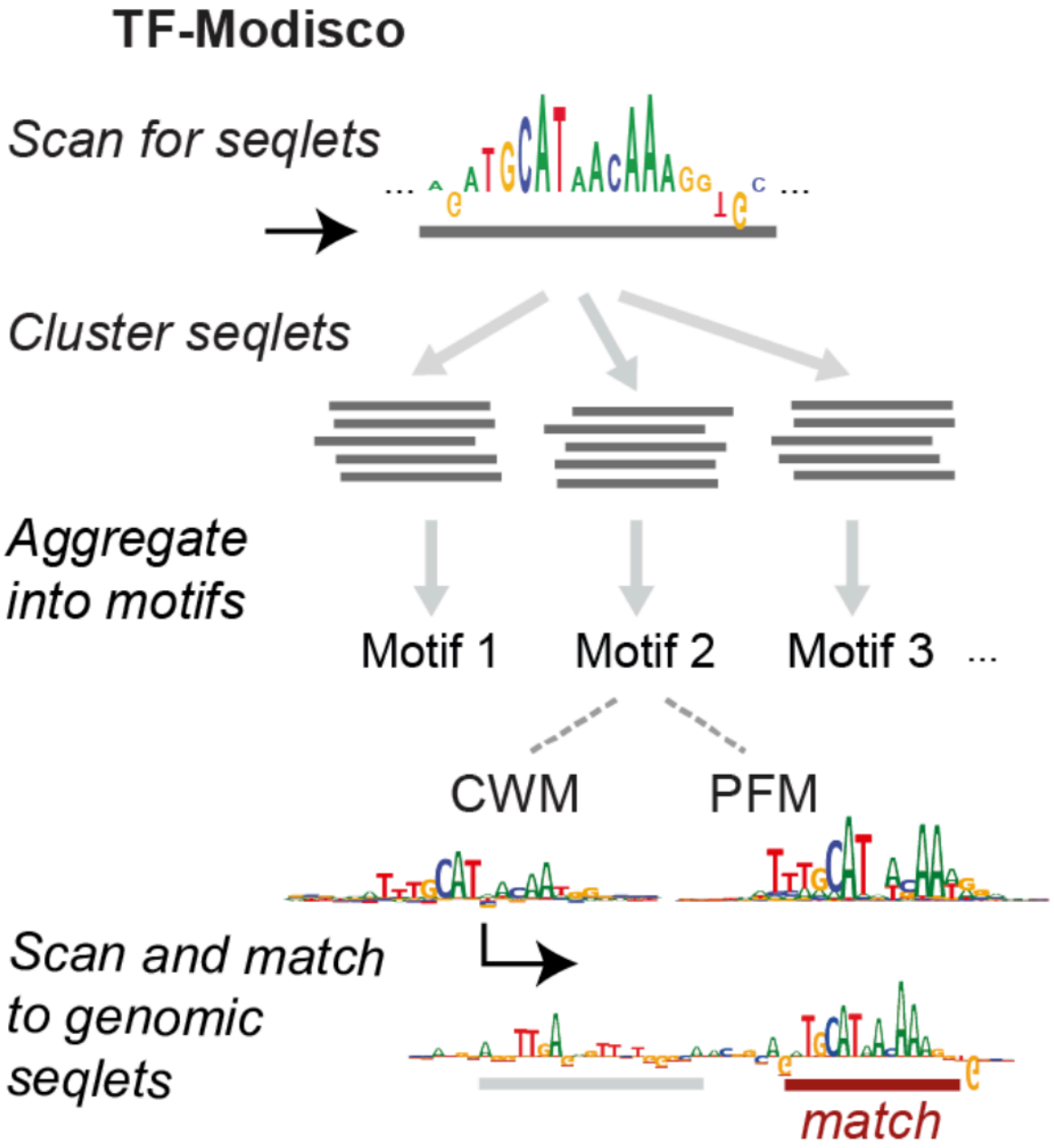
$$c^{(profile),i} = \sum_{j,s} c_{js}^{i}\, p_{js}$$

  where c_{js}^{i} is the DeepLIFT attribution for input sequence position *i* to output position *j* on strand *s* and p_{js} is the *j,s* index of **p** = softmax(f(**x**))
- So p is just the function output in probability space instead of logit space
- They say “the rationale for performing a weighted sum is that positions with high predicted profile output values should be given more weight than positions with low predicted profile output values.”
- I think it’s weird though, this really removes any weight for places where the model is confident that there’s no binding (large negative magnitude in logit space, 0 in prob. space)
- Places where the model is confident are already scaled by the magnitude of their logit output



Cluster attributions into motifs

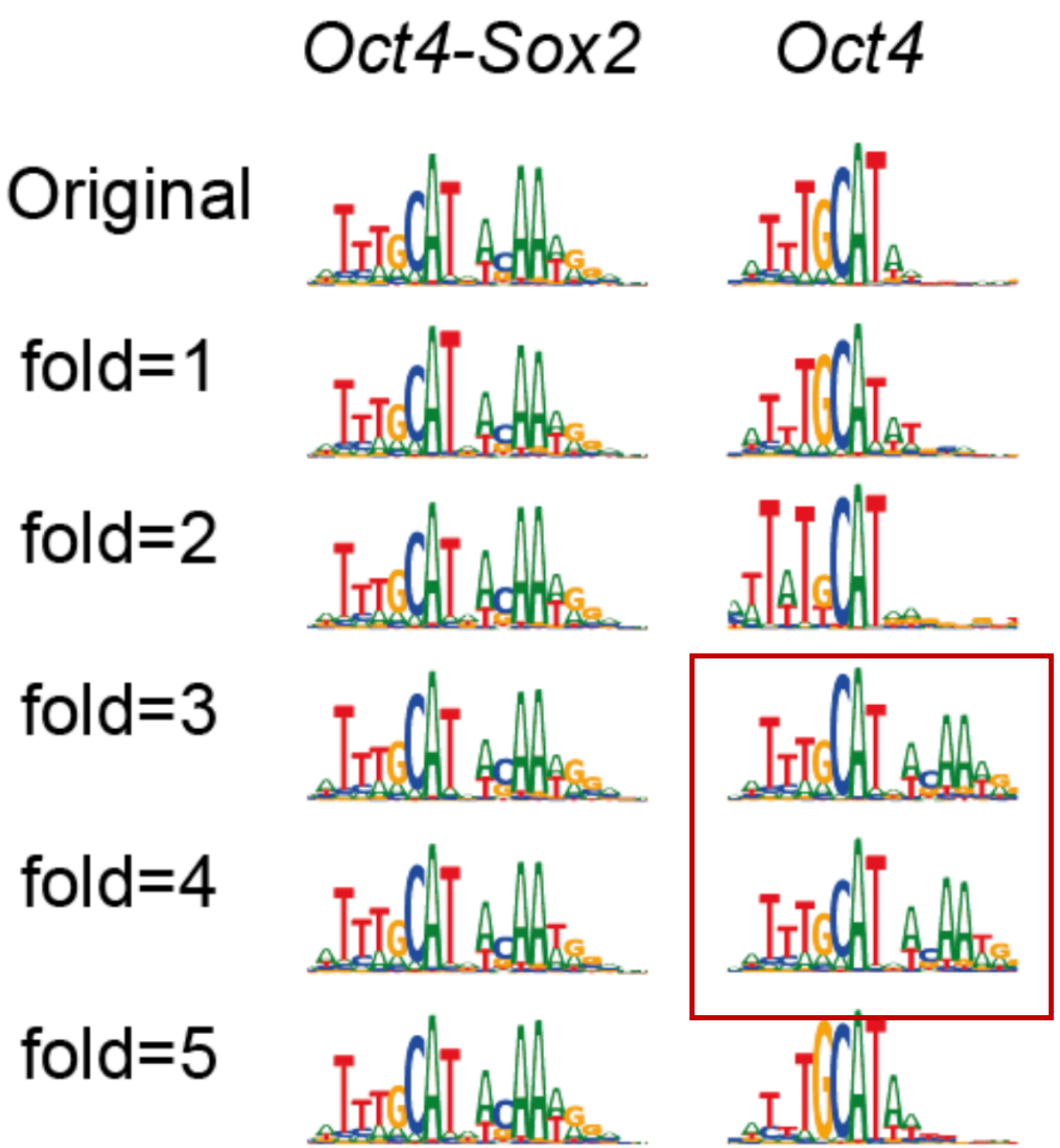
- “Seqlets” are short sequences w/ statistically significantly higher attribution than shuffled sequences
- Cluster these using a community detection algorithm
- Do some heuristic processing to merge clusters and throw out bad looking clusters
- Average attributions into CWM motifs over all aligned sequences
- Also generate PFMs by looking at frequencies of bases at each position in aligned sequences



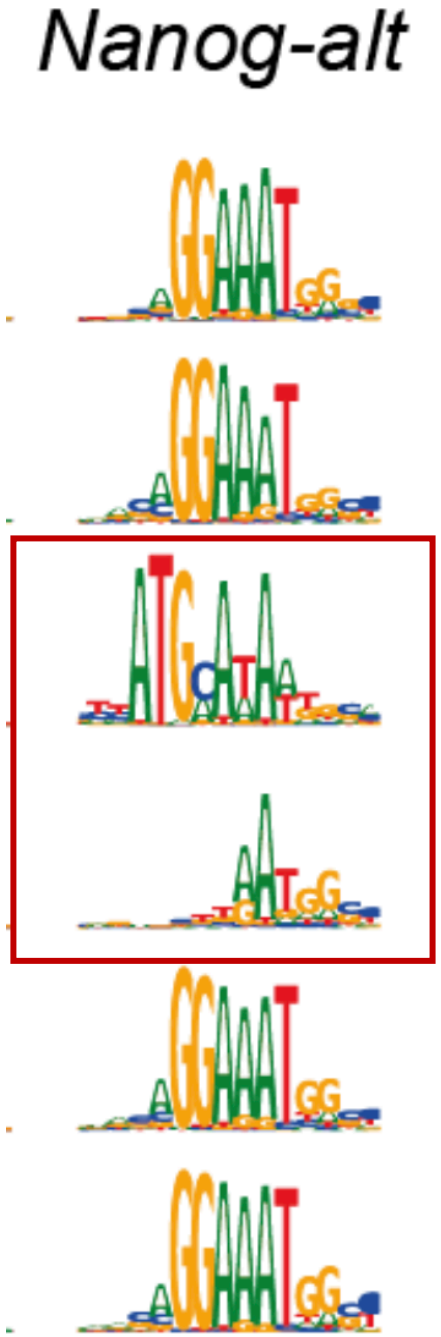
Computational validation of motifs (supplemental fig 6)

- Are the motifs learned by models robust?
- Train 5 additional models on different subsets of the data and generate motifs for these

Validation of motifs



Validation of motifs



Validation of motifs

- Is this really that robust (40% of the time different for some motifs)
- Why not just average over re-trainings?

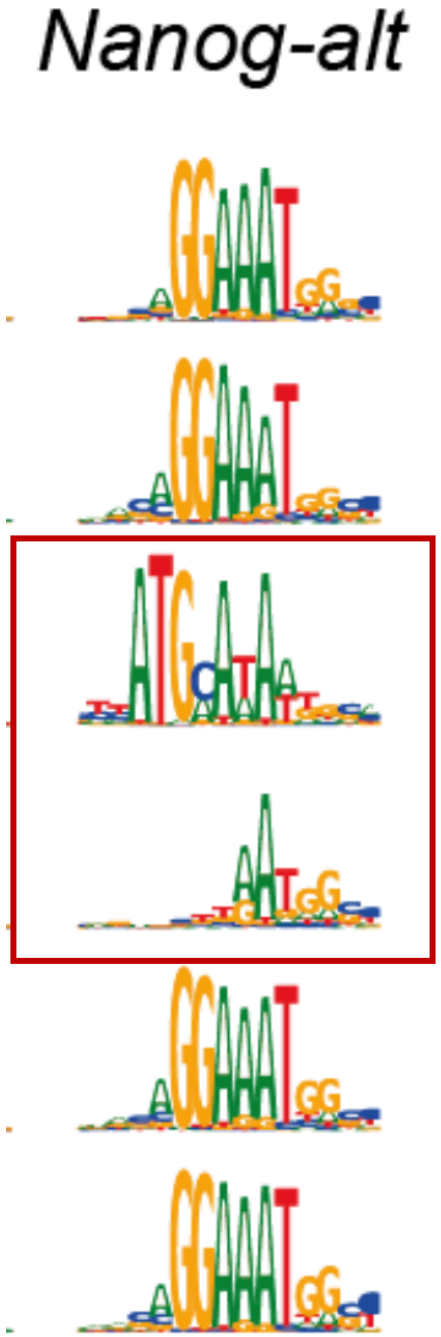


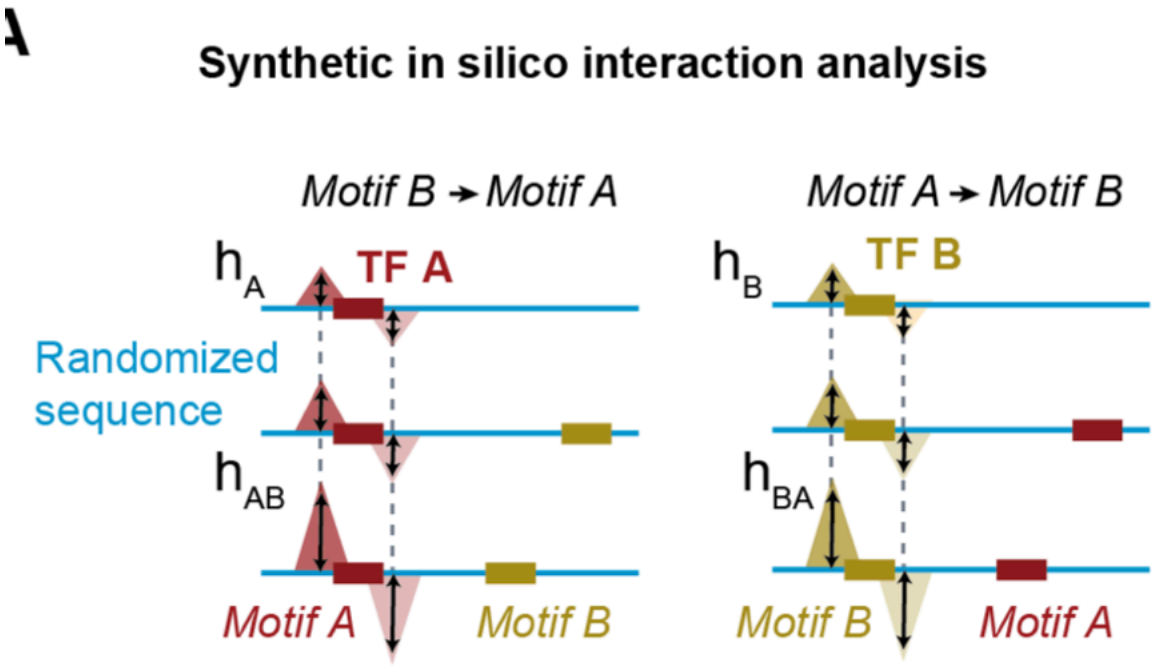
Figure 4
Higher order syntax

Two approaches to motif syntax

- To extract rules of cooperativity, measure how the binding of a TF to its motif is enhanced by a second motif (and how this depends on the distance between these motifs)
- Synthetic approach
- Naturally occurring motifs in sequences

Synthetic approach

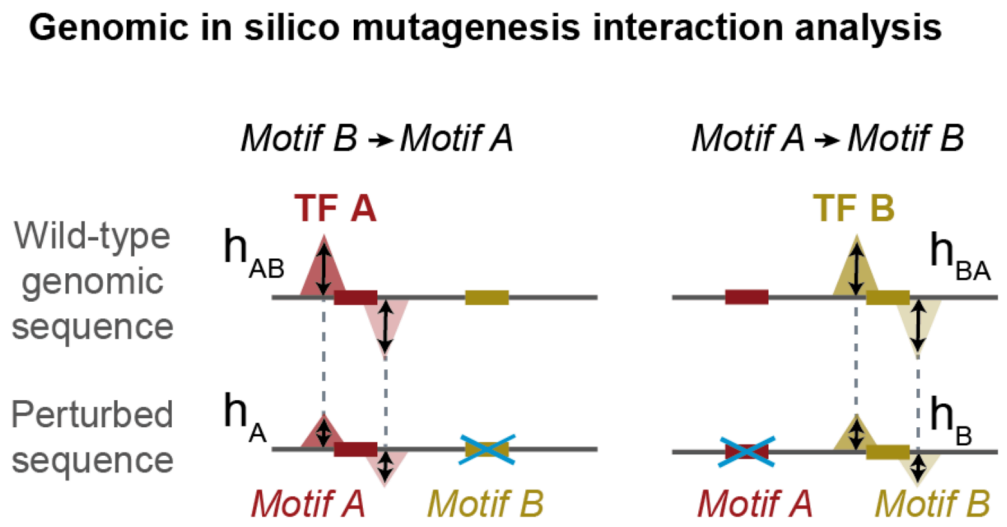
- Create 128 sequences where each base is independent uniform random
- Replace the central bases by *Motif A*
- Insert *Motif B d* bases downstream of *Motif A* (where the distance is measured from the centers of the motifs)
- Predict the strand-specific ChIP-nexus profile for the primary TF of *Motif A* (e.g. Oct4 for the *Oct4-Sox2 Motif*)
- Average the predictions across the 128 random background sequences
- Strand-specific summit is then h_{AB}
- Just add *Motif A* to the center of the 128 sequences, predict, and average
- Strand-specific summit in this case is h_A
- Just add *Motif B* to position *d* off center, predict, and average
- Strand-specific summit in this case is h_B
- Average the prediction across the 128 sequences when neither motif has been added
- Strand-specific summit in this case is $h_{\emptyset}$
- Binding fold change is $(h_{AB} – (h_B - h_{\emptyset}))/ h_A$
  - > 1 means positive interaction, <1 means neg. interaction, 1 means no interaction



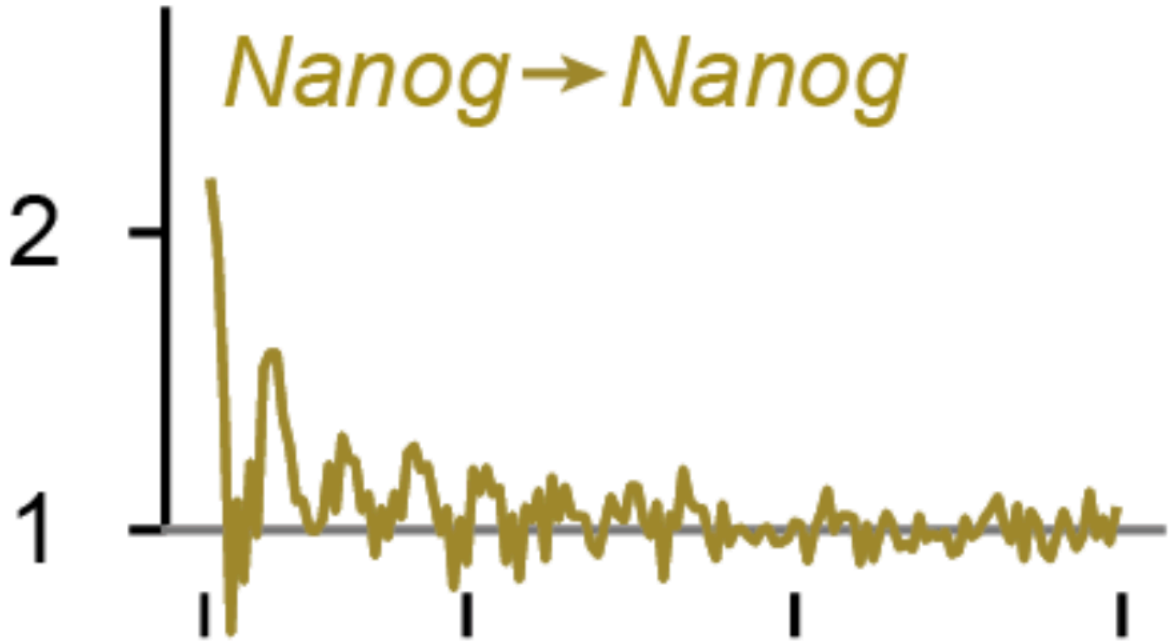
Genomic approach

- Find instances of co-occurring motifs in the genome
- Now replace either *Motif A*, *Motif B*, or both with random sequence
- Add pseudo-counts to both numerator and denominator of fold change
  - “20th percentile of the considered quantity”

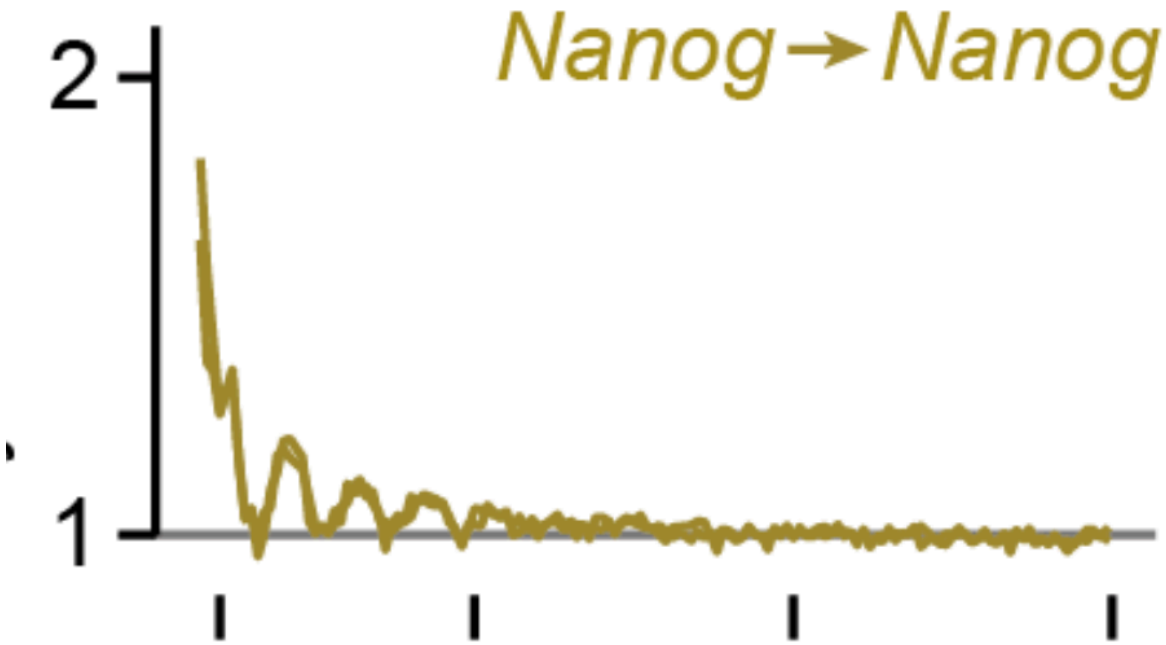
$$(h_{AB} - (h_B - h_{\emptyset}) + PC_{AB}) / (h_A + PC_A)$$



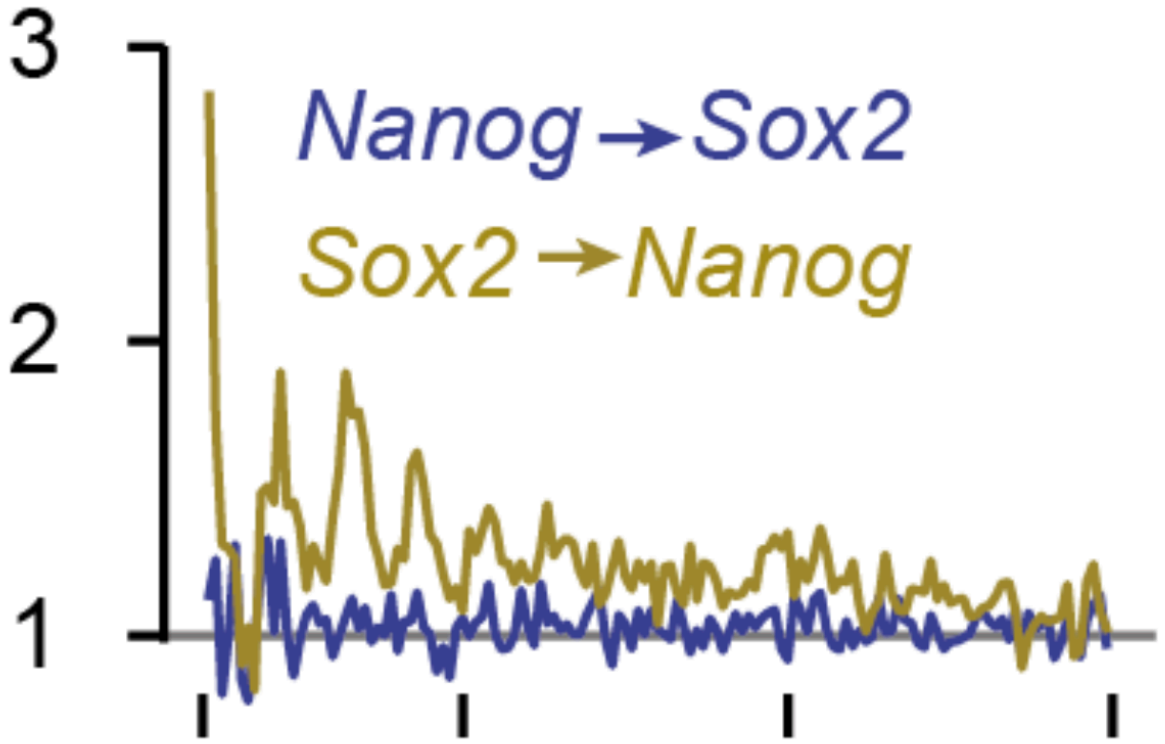
Synthetic approach



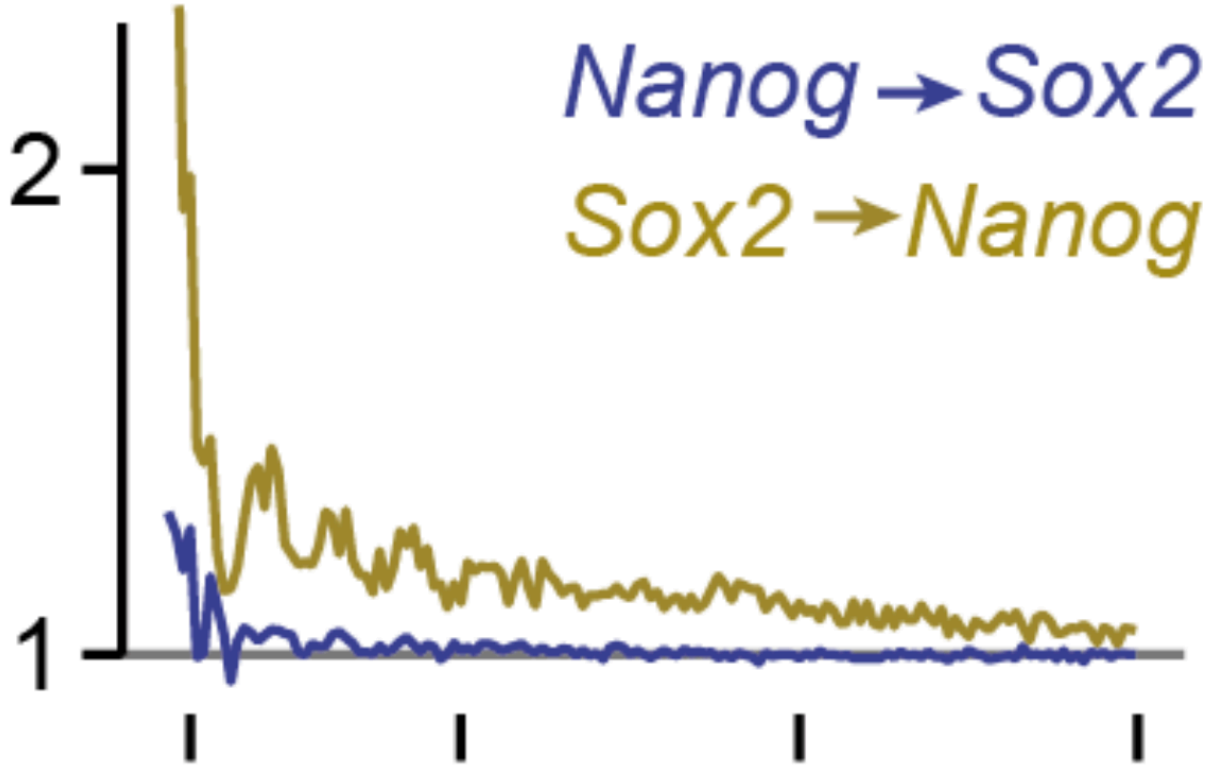
Genomic approach



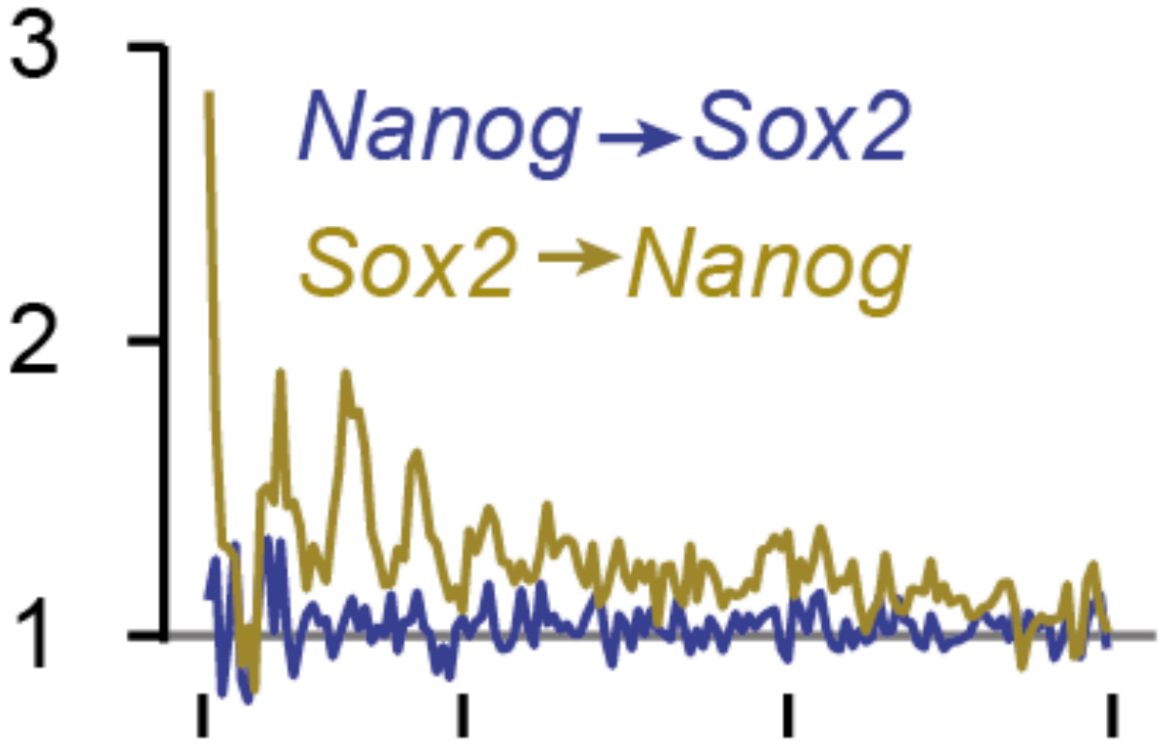
Synthetic approach



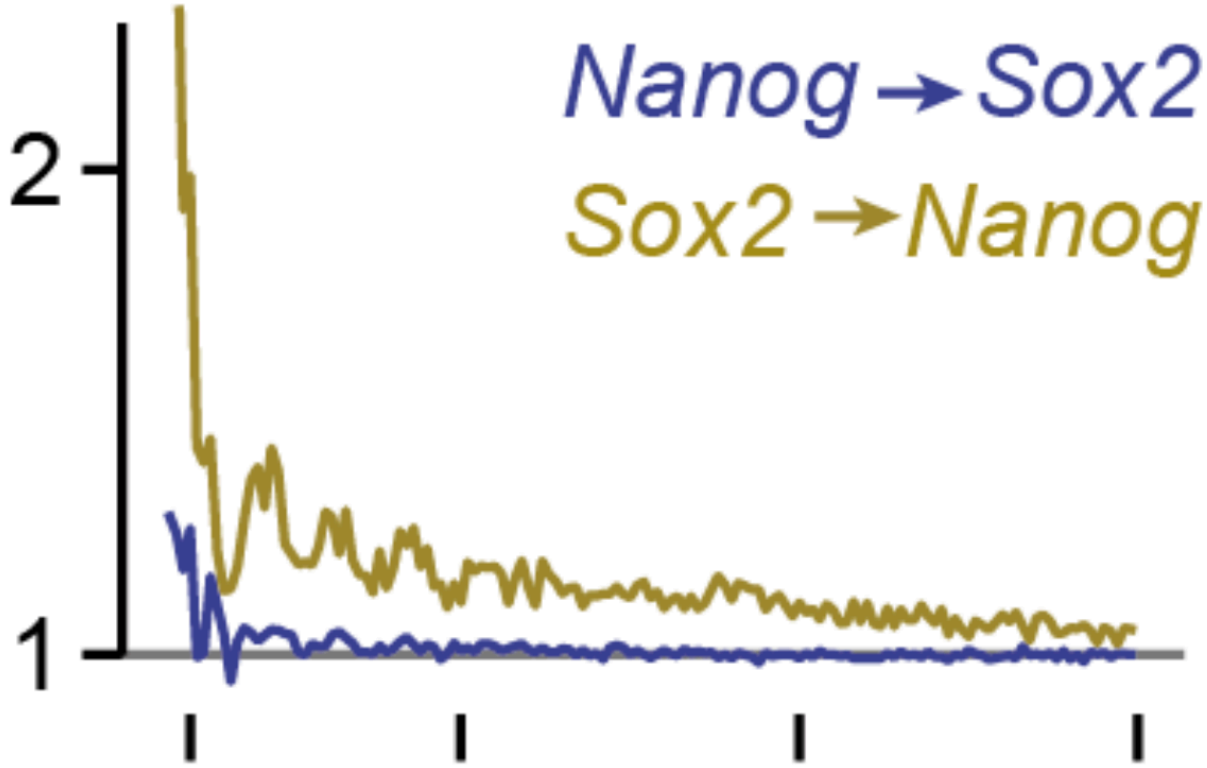
Genomic approach

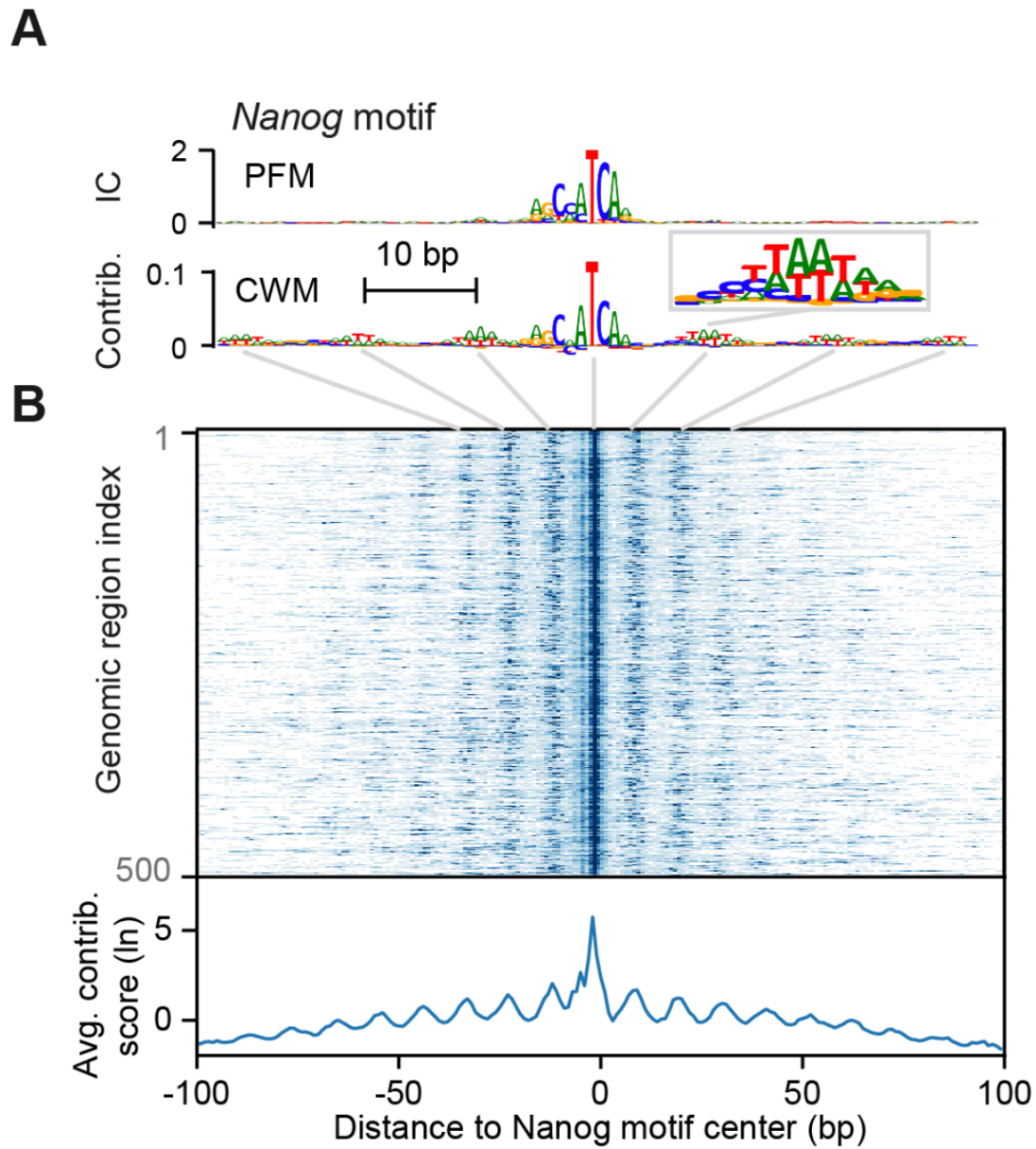


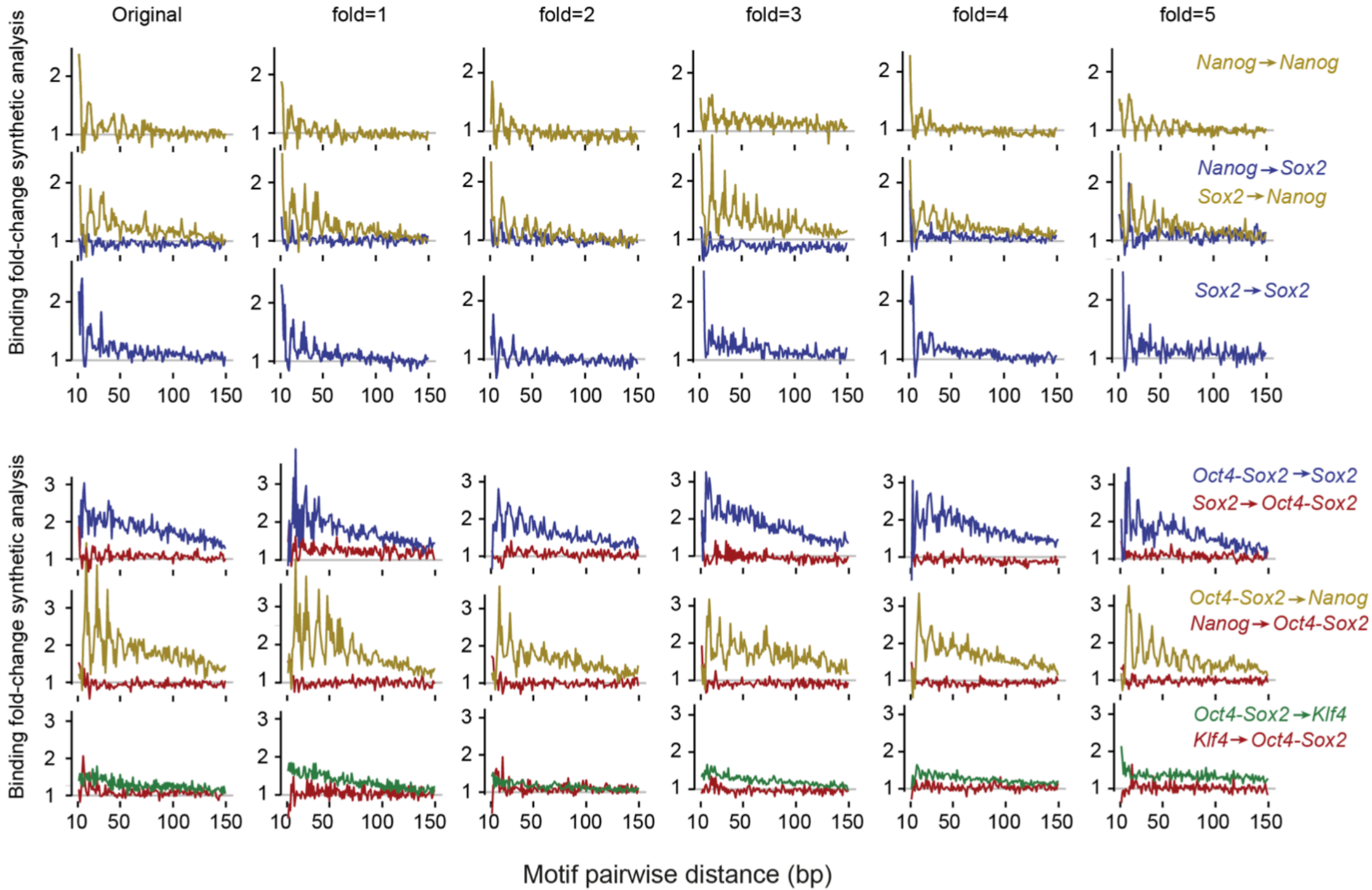
Synthetic approach

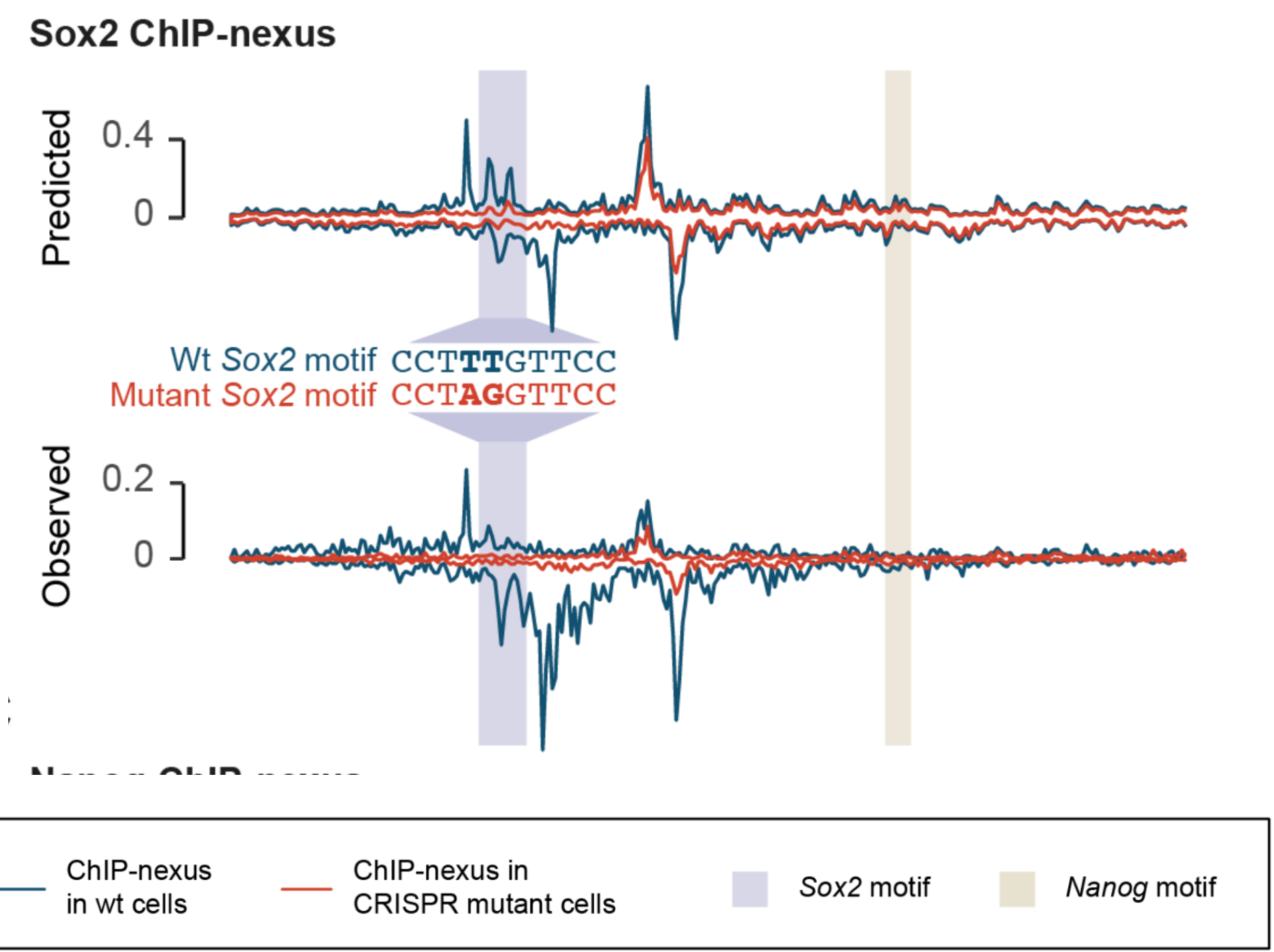


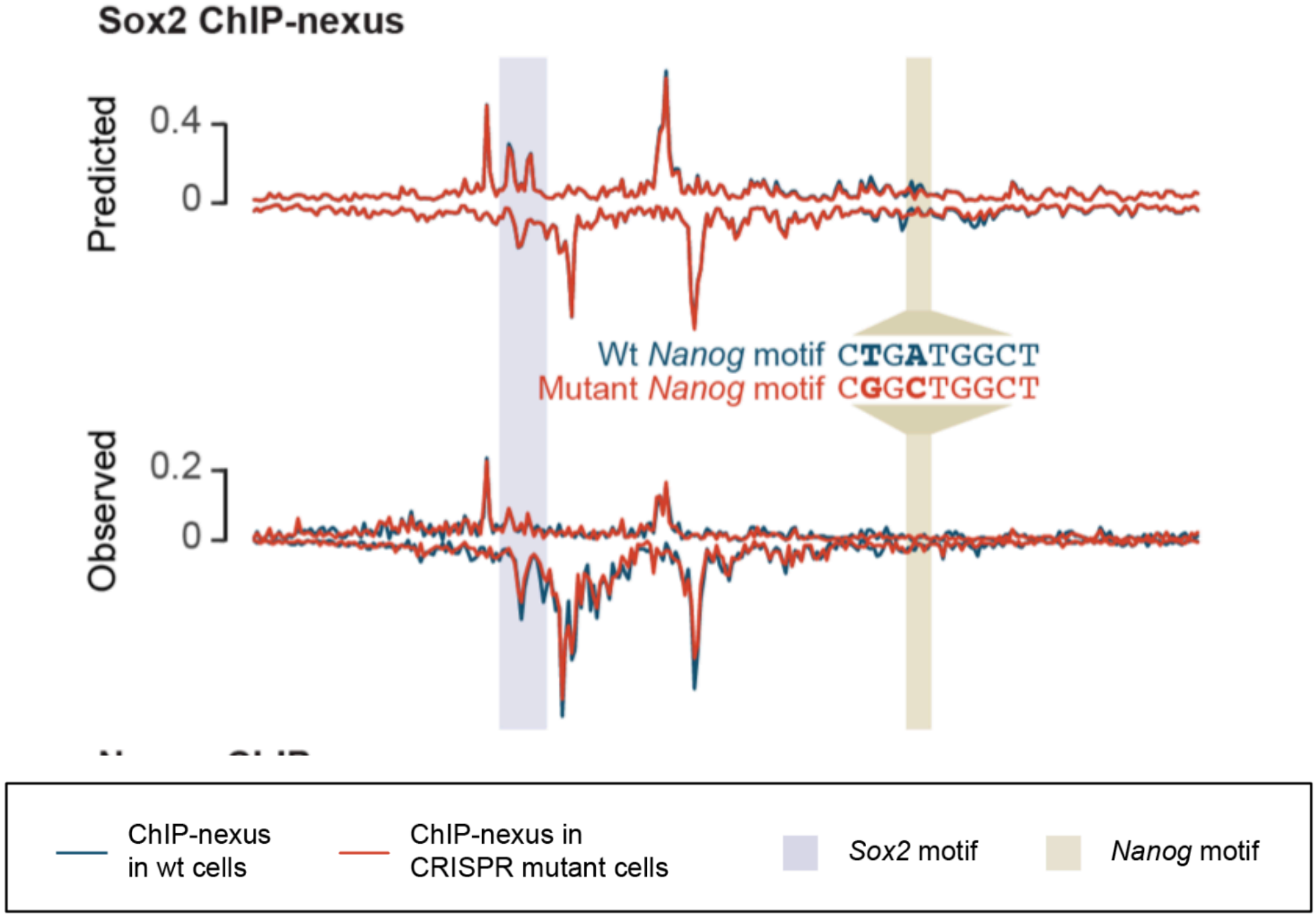
Genomic approach

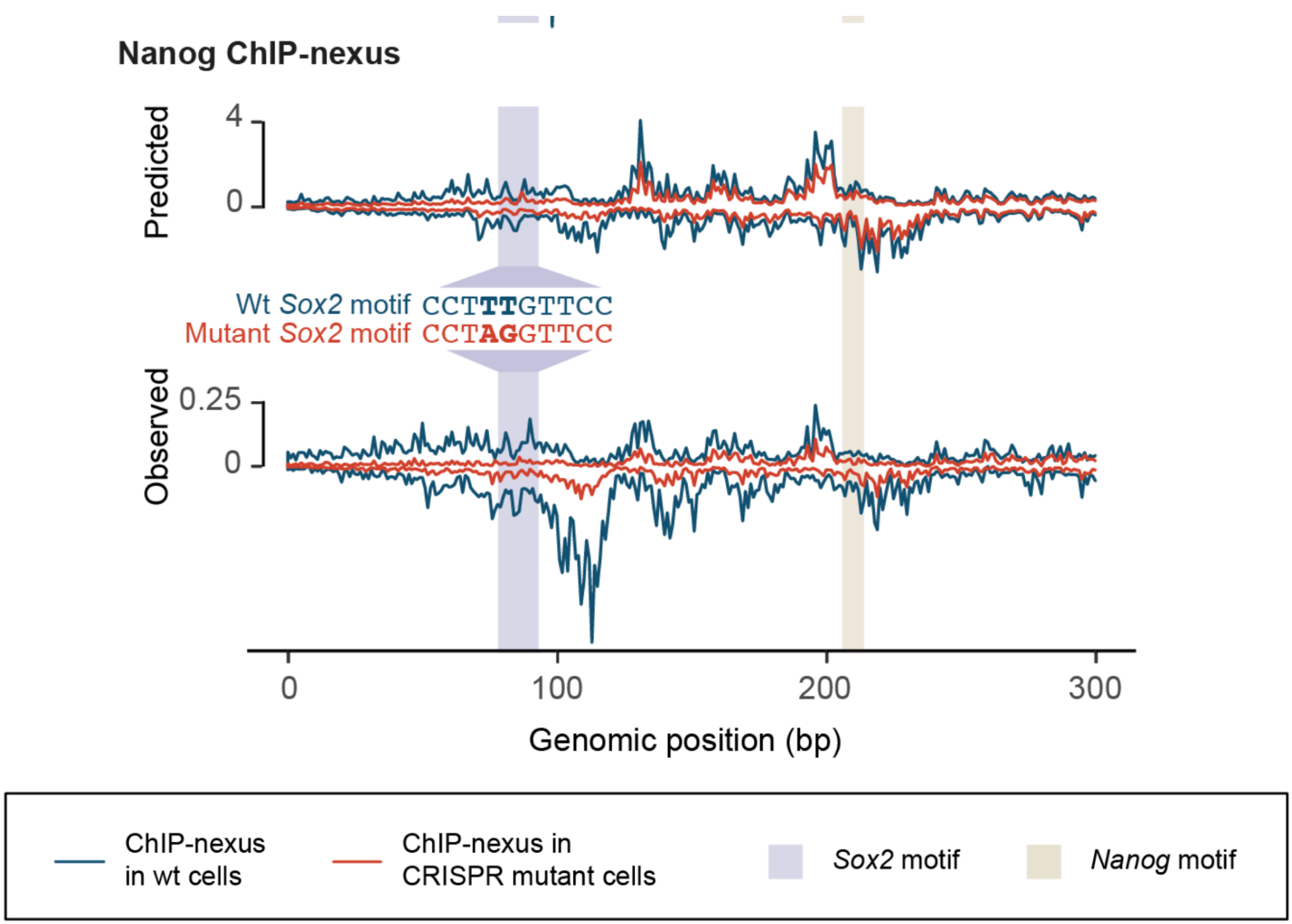


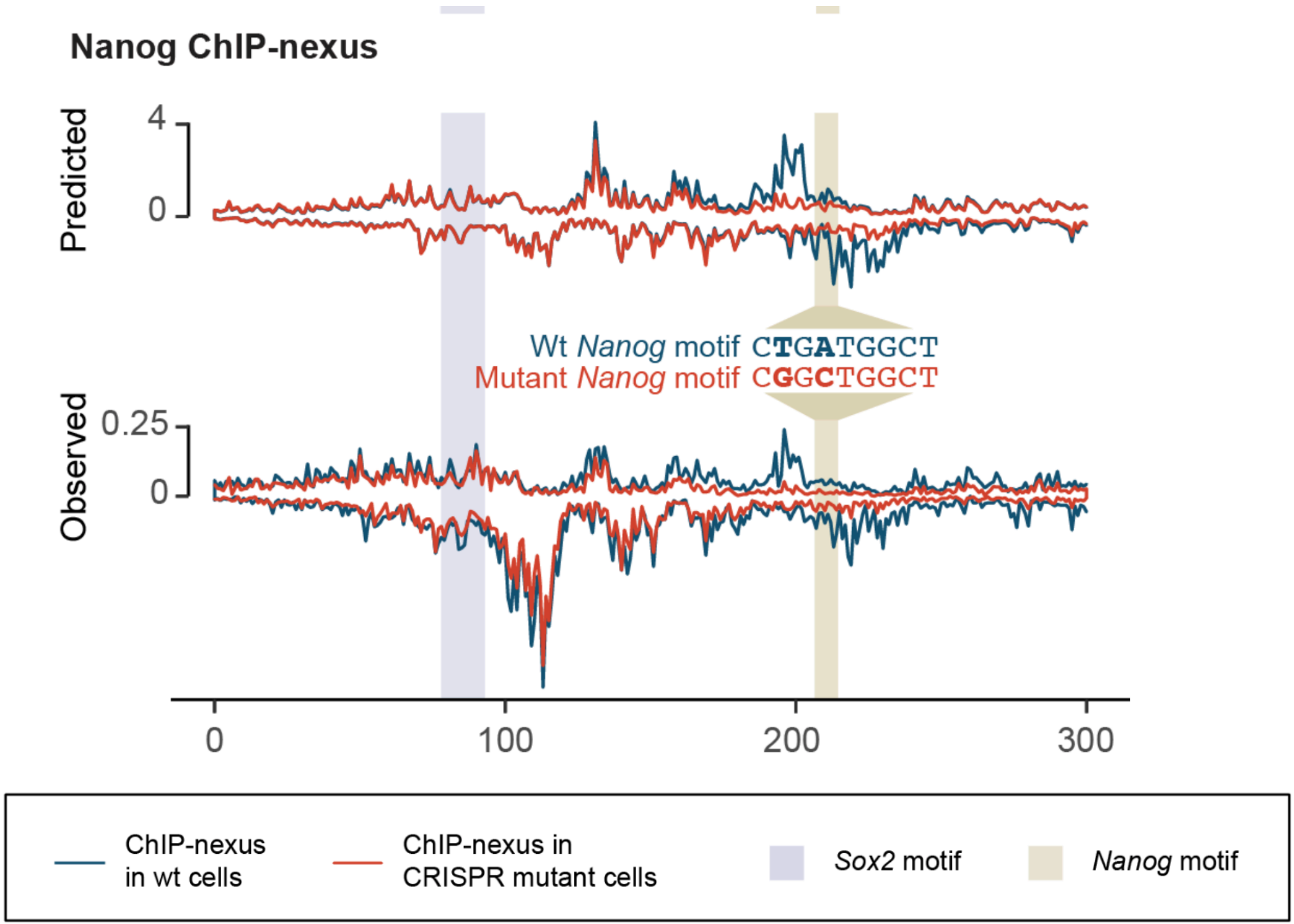

A
Nanog motif
IC
2
0
PFM
Contrib.
0.1
0
CWM
10 bp
B
Genomic region index
1
500
Avg. contrib. score (ln)
5
0
-100
-50
0
50
100
Distance to Nanog motif center (bp)


Original
fold=1
fold=2
fold=3
fold=4
fold=5
Binding fold-change synthetic analysis
Nanog→Nanog
Nanog→Sox2
Sox2→Nanog
Sox2→Sox2
Oct4-Sox2→Sox2
Sox2→Oct4-Sox2
Oct4-Sox2→Nanog
Nanog→Oct4-Sox2
Oct4-Sox2→Klf4
Klf4→Oct4-Sox2
10 50 100 150
Motif pairwise distance (bp)


Sox2 ChIP-nexus
Predicted
0.4
0
Wt Sox2 motif CCTTTGTTCC
Mutant Sox2 motif CCTAGGTTCC
Observed
0.2
0
ChIP-nexus in wt cells
ChIP-nexus in CRISPR mutant cells
Sox2 motif
Nanog motif


Sox2 ChIP-nexus
Predicted
0.4
0
Wt Nanog motif CTGATGGCT
Mutant Nanog motif CGGCTGGCT
Observed
0.2
0
ChIP-nexus in wt cells
ChIP-nexus in CRISPR mutant cells
Sox2 motif
Nanog motif


Nanog ChIP-nexus
Predicted
4
0
Wt Sox2 motif CCTTTGTTCC
Mutant Sox2 motif CCTAGGTTCC
Observed
0.25
0
0
100
200
300
Genomic position (bp)
ChIP-nexus in wt cells
ChIP-nexus in CRISPR mutant cells
Sox2 motif
Nanog motif


Nanog ChIP-nexus
Predicted
4
0
Wt Nanog motif CTGATGGCT
Mutant Nanog motif CGGCTGGCT
Observed
0.25
0
0
100
200
300
ChIP-nexus in wt cells
ChIP-nexus in CRISPR mutant cells
Sox2 motif
Nanog motif