

# Computational Systems Biology Deep Learning in the Life Sciences

6.802 6.874 20.390 20.490 HST.506

David Gifford  
Lecture 7  
March 16, 2017

## Characterizing Cellular Regulation



Massachusetts  
Institute of  
Technology

<http://mit6874.github.io>

# Overall goal for today

- Understand models of the two levels of cellular regulation: chromatin accessibility and transcription factor cis-effects
- Recall issues in model selection given limited data

# Today's lecture

- Model selection
  - Model performance as a function of capacity
- Models of chromatin accessibility
  - Log-linear models based on k-mers
  - What can we learn about biology from models?
- Models of gene expression
  - Regression trees provide interpretable models

# Part 1 - Model Capacity

# Model capacity

- Capacity - ability to fit a wide range of functions (hypothesis space)
- Vapnik-Chervonenkis dimension - size of largest unique training set of examples a binary classifier can label arbitrarily is a measure of capacity

# Generalization and Capacity

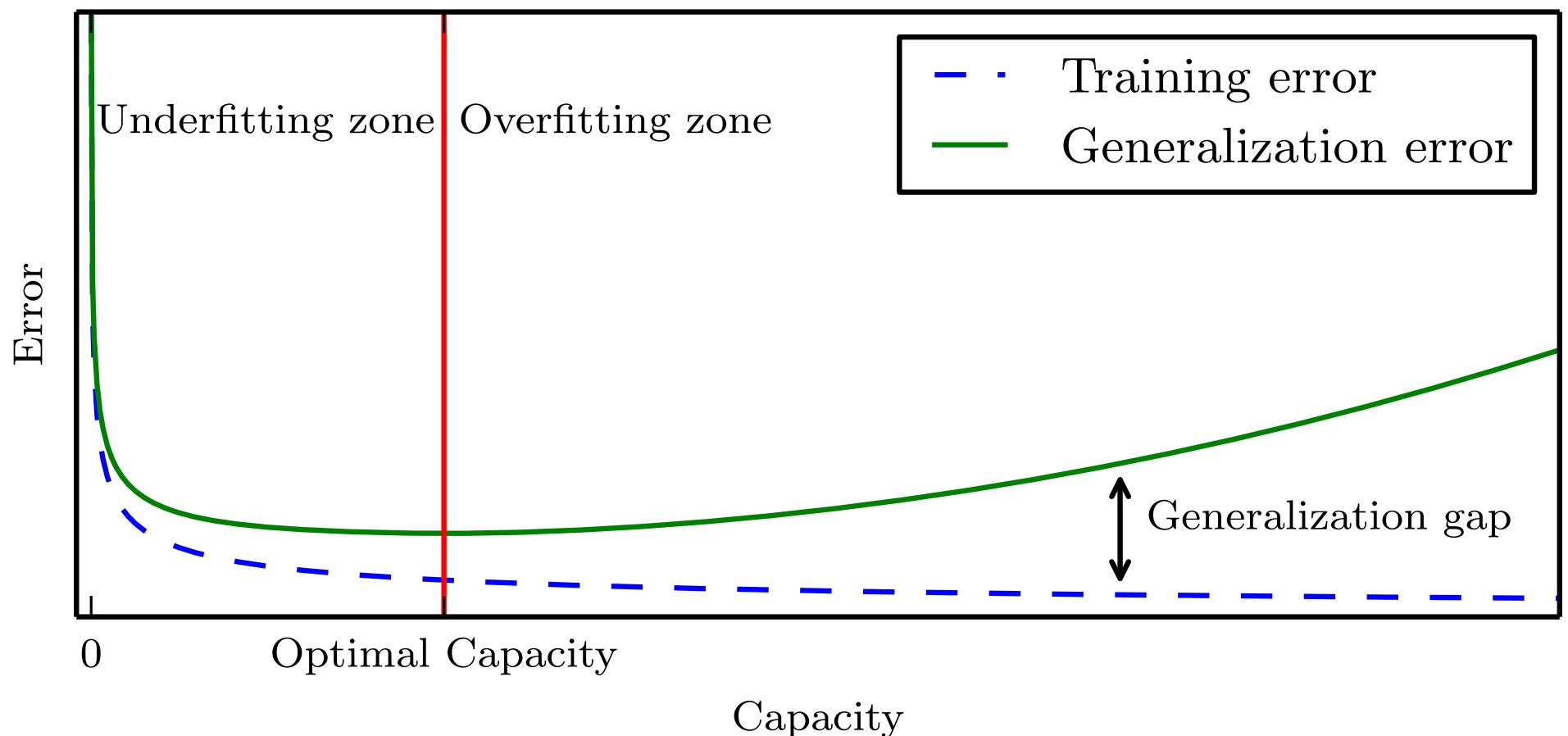


Figure 5.3

The capacity of non-parametric models is defined by the size of their training set

- k-nearest neighbor (KNN) regression computes its output based upon the  $k$  “nearest” training examples
- Often the best method, and certainly a baseline to beat

# Bayes error is residual noise from confounders and observation noise

- Bayes error is the error made by an Oracle given the training set
- For example, if there is an unobserved variable that affects the labels the Oracle's predictions will be noisy

# Sufficient training data are necessary to generalize well

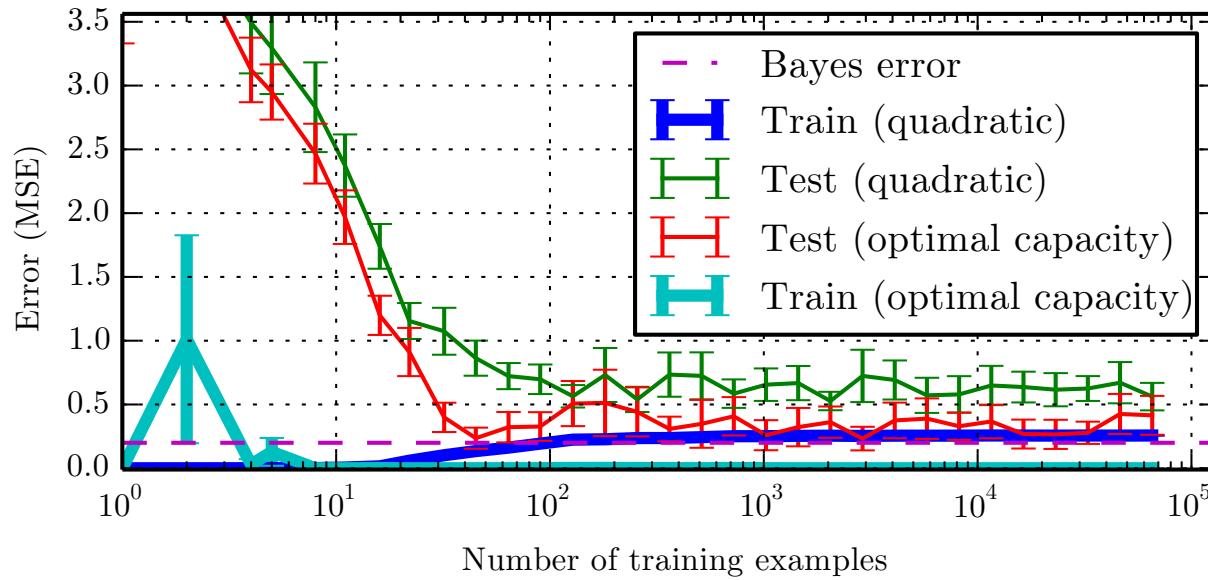
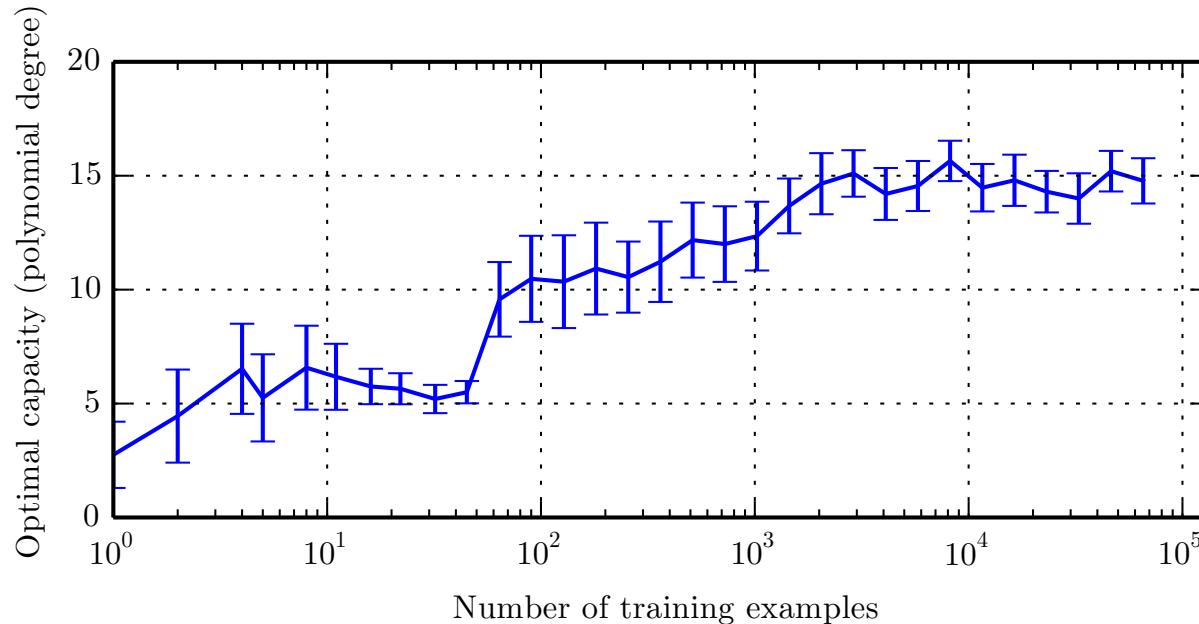


Figure 5.4

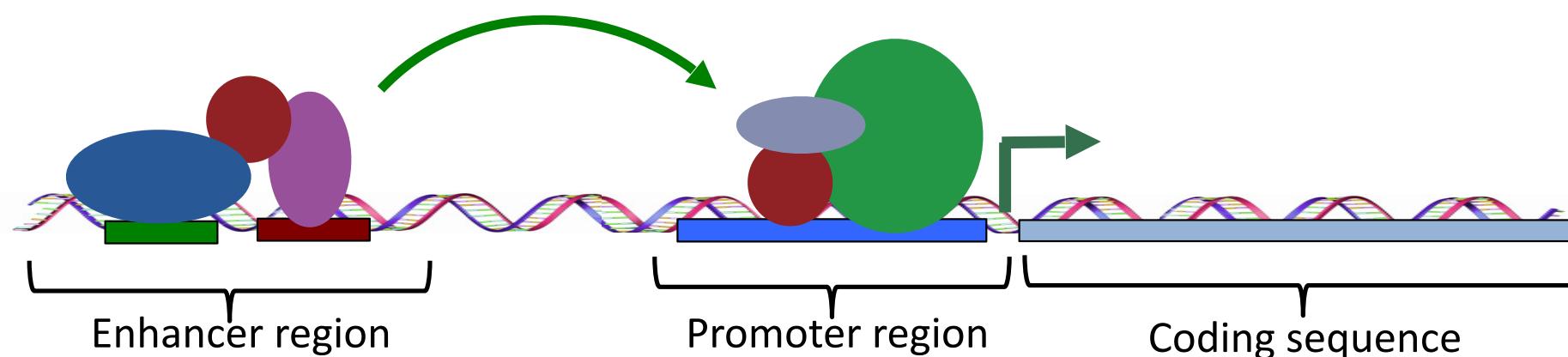


# Machine Learning has limits

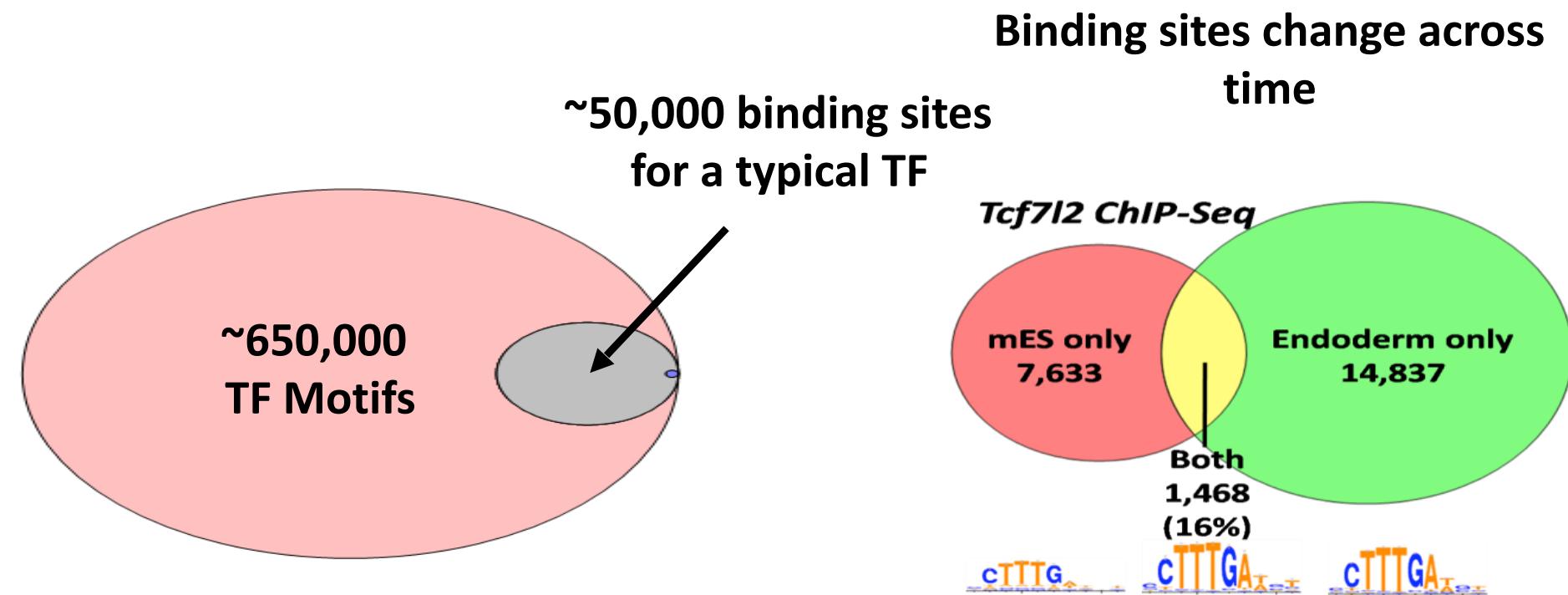
- The *no free lunch* theorem: You can only obtain generalization from finitely many training examples if the algorithm searches a limited hypothesis space.
- We desire a learning algorithm to perform *generalization* and to be *stable*. It generalize if the training error on a data set will converge to the expected error. It is stable if small perturbations in the data result in only small perturbations in the output hypothesis.

# Part 2 - Modeling Genome Accessibility

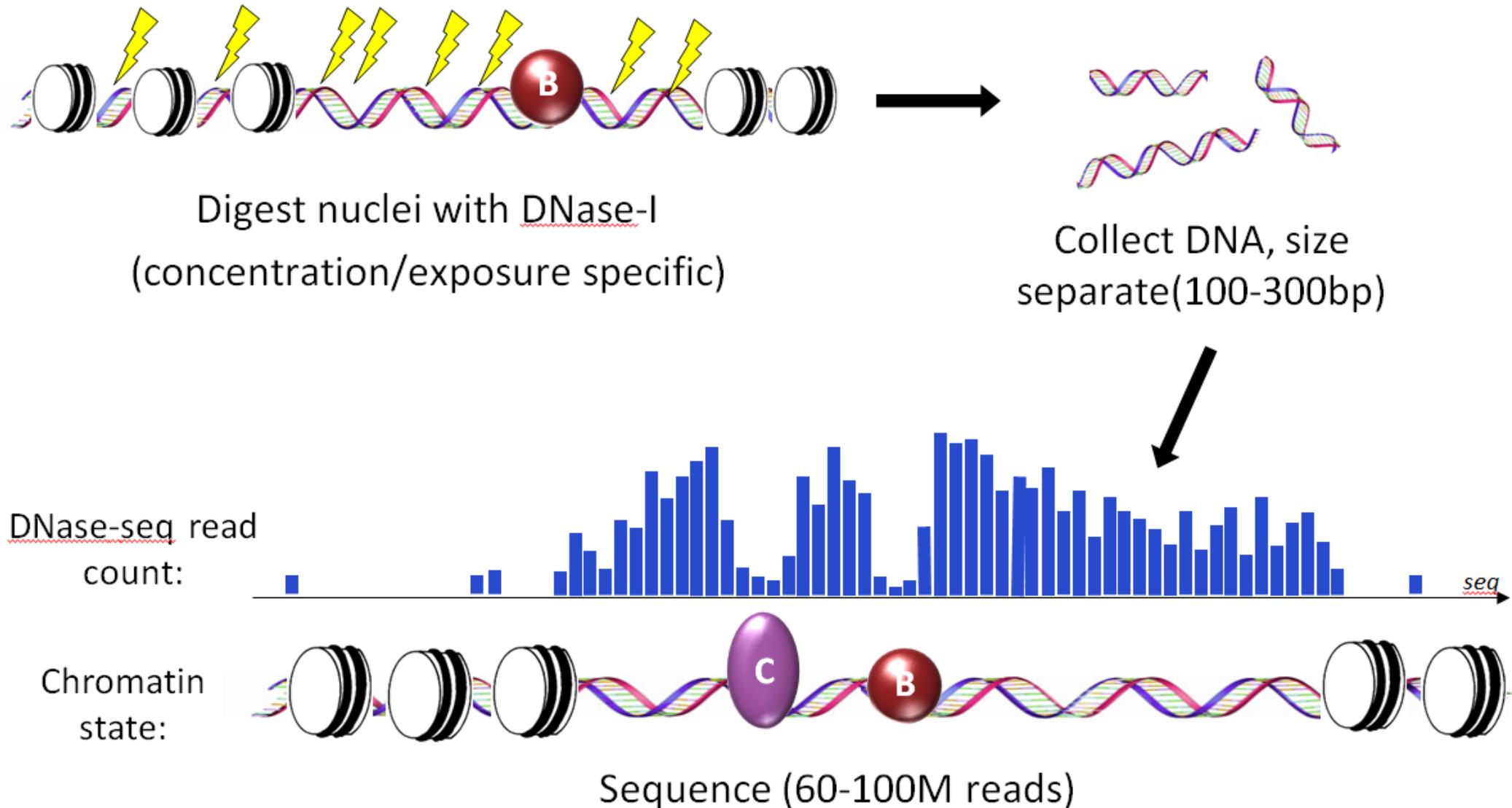
# How do transcription factors control activation of cell-type-specific promoters and enhancers?



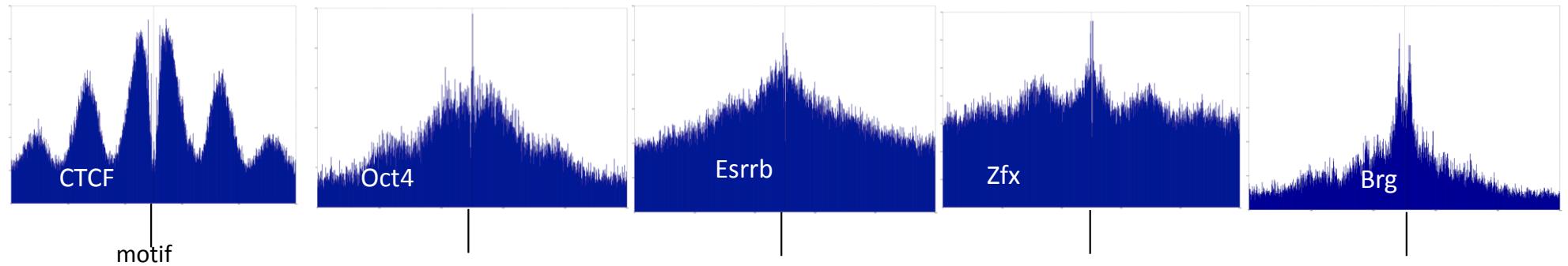
# Motifs are insufficient to predict TF binding



# DNase-seq reveals genome protection profiles



# Bound factors leave distinct DNase-seq profiles

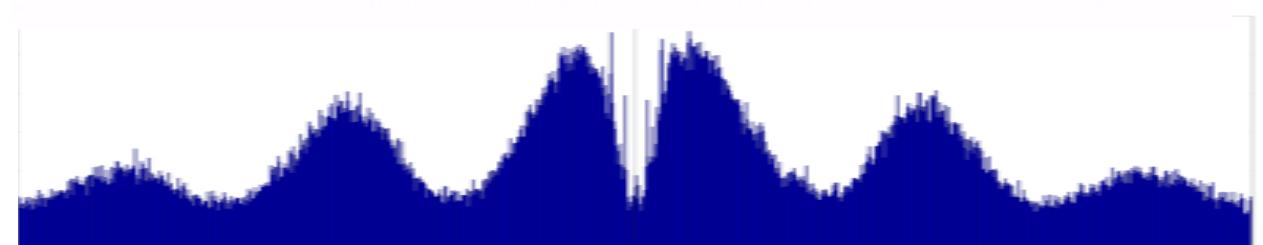


**Individual binding site prediction is difficult**

Individual CTCF:



Aggregate CTCF:



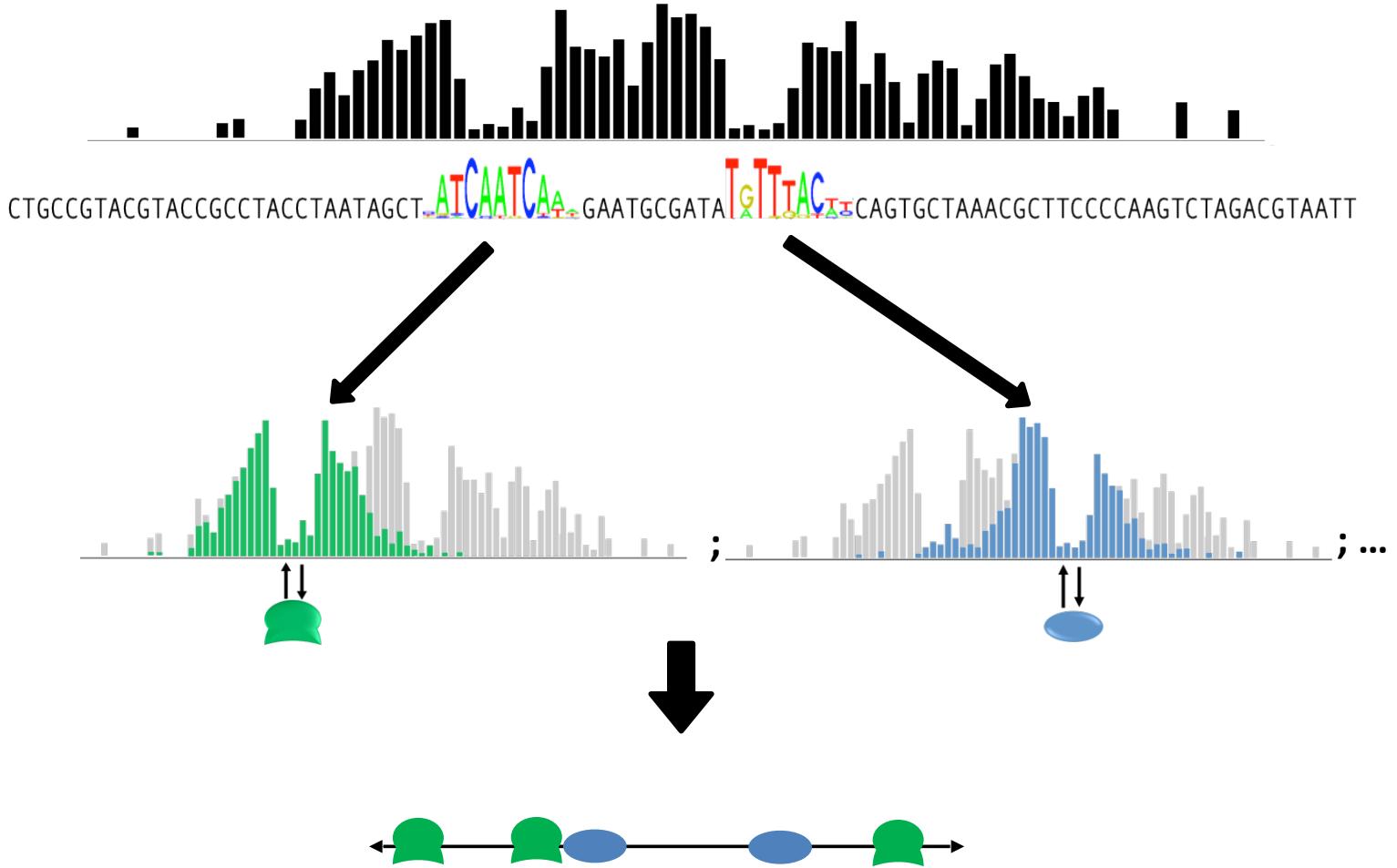
# ***PIQ*: algorithm to predictively model TF binding from DNase-seq + Sequence**

Input:

Scan 1331  
motifs

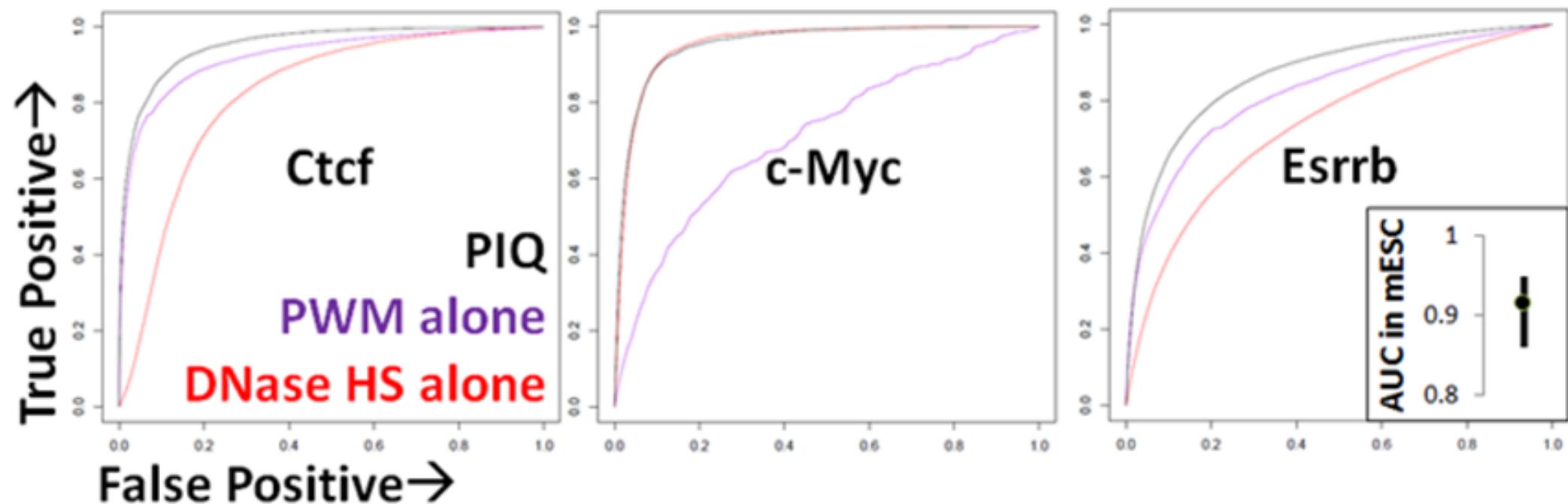
Modeling:

Predictions:



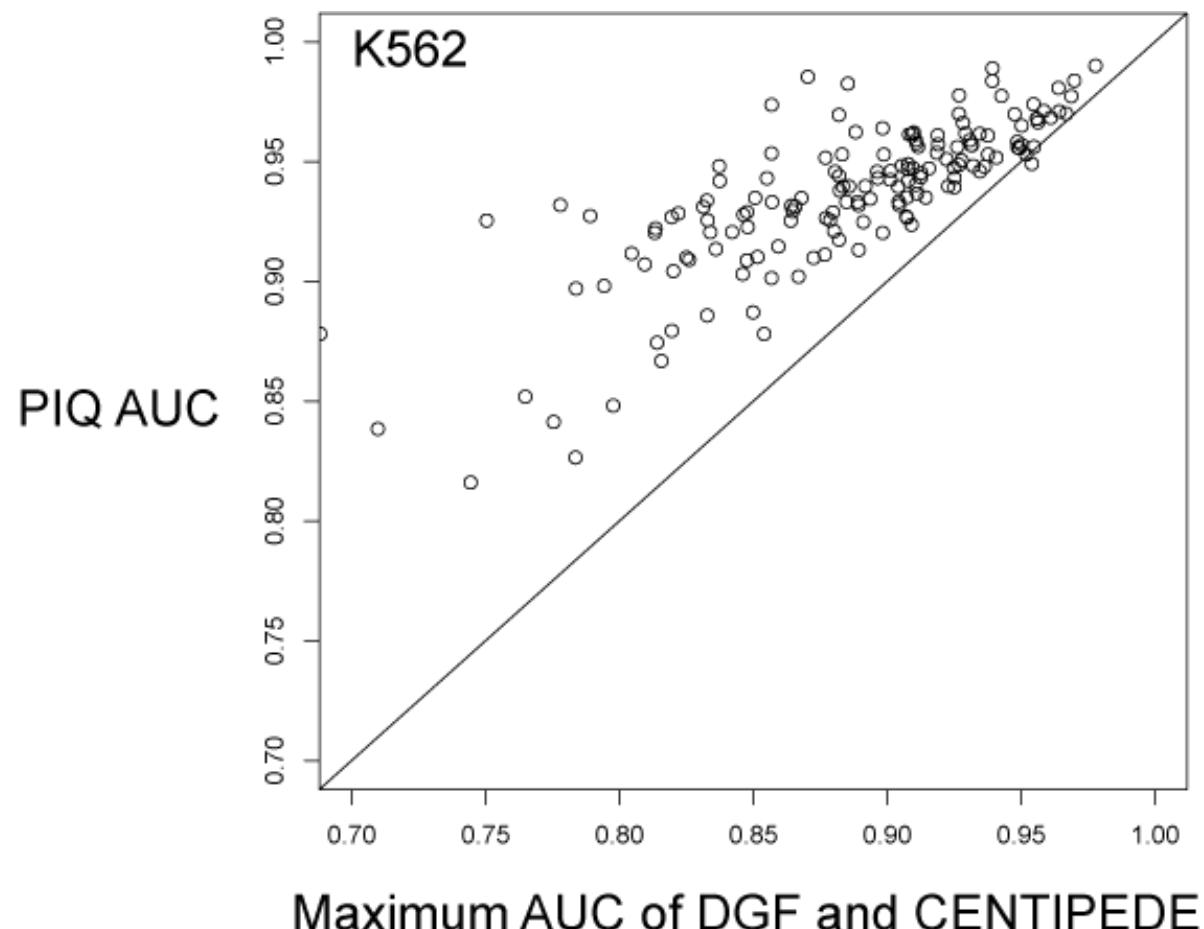
# PIQ accurately predicts mESC TF binding

Receiver operating characteristic (ROC) curves show PIQ matches closely with ChIP-seq data.

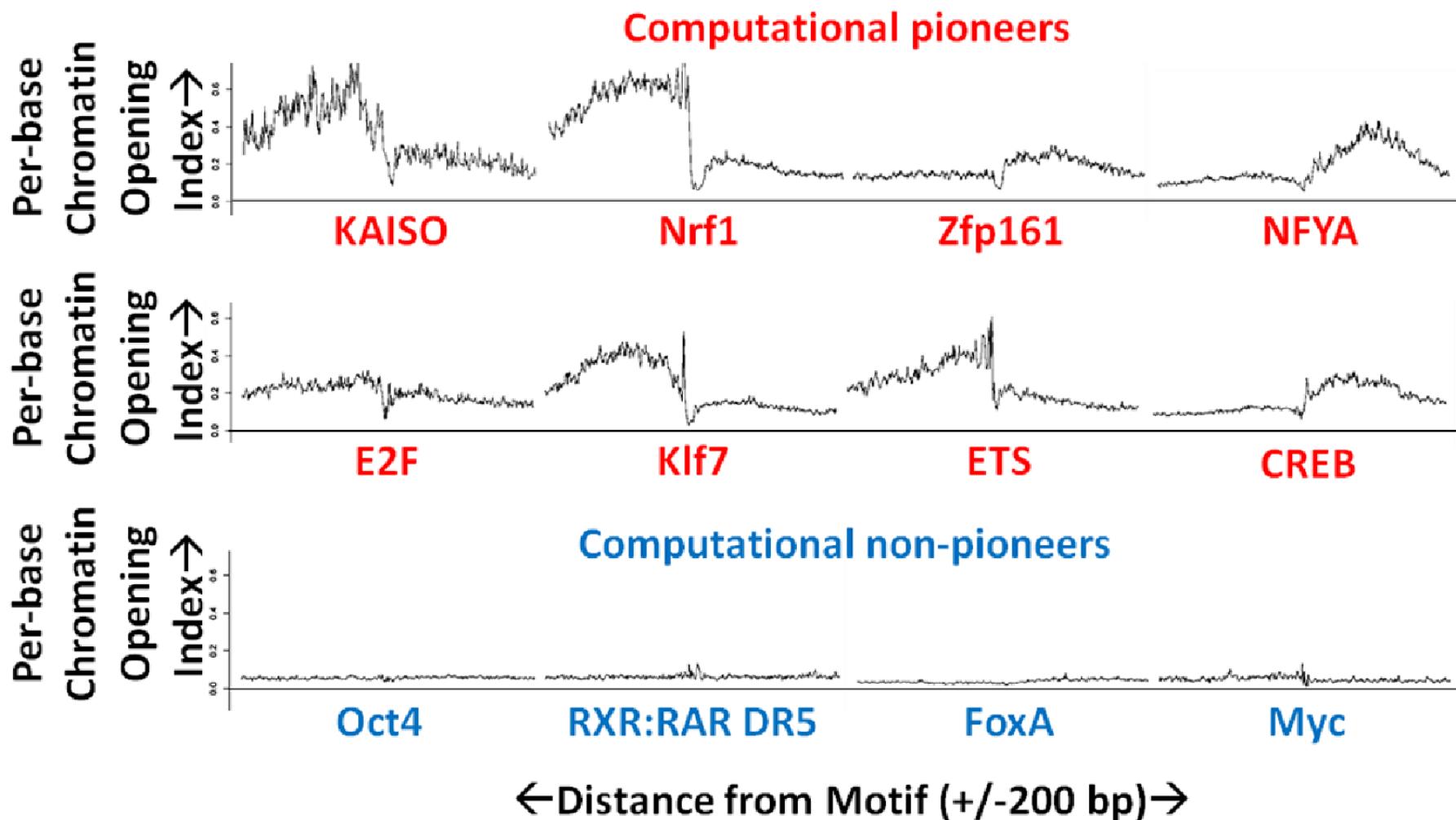


# PIQ outperforms existing methods when predicting binding for 313 ENCODE ChIP-seq experiments

PIQ (.93 Mean AUC); Centipede (.87); DGF (.65)

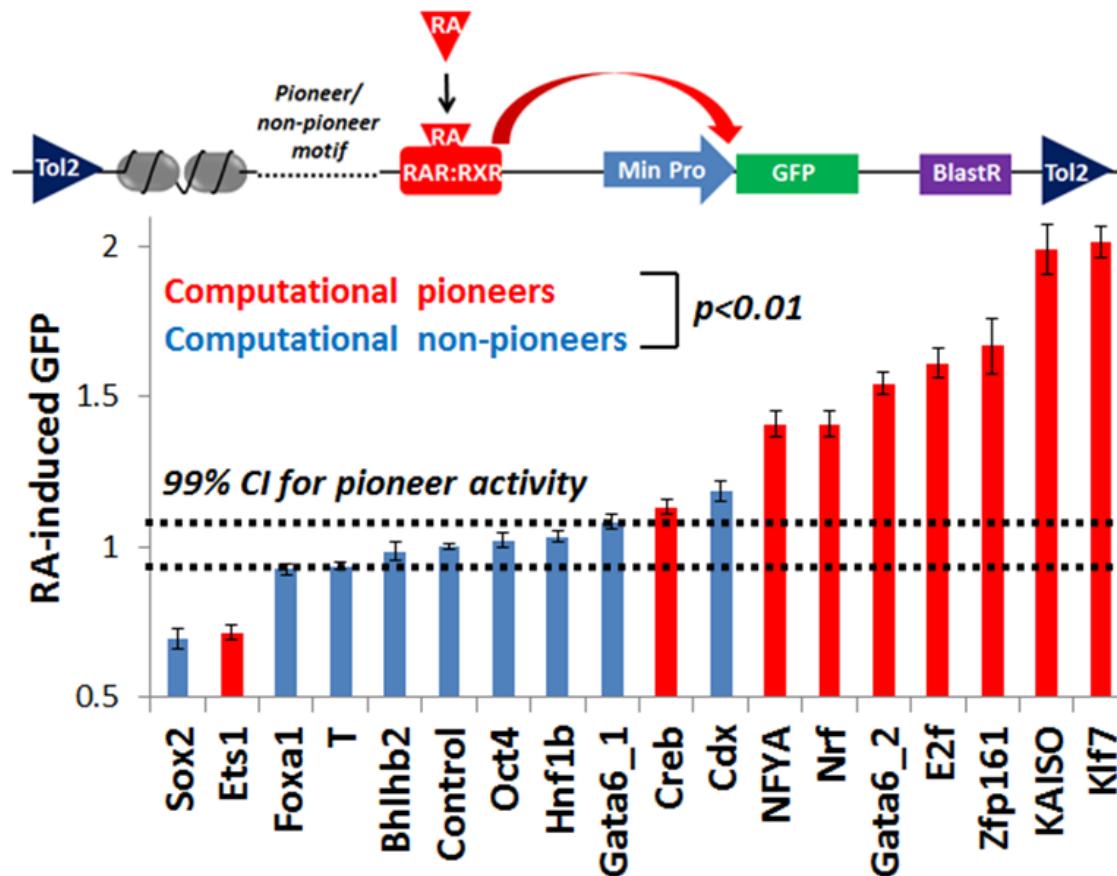


# Pioneer TFs have identifiable profiles

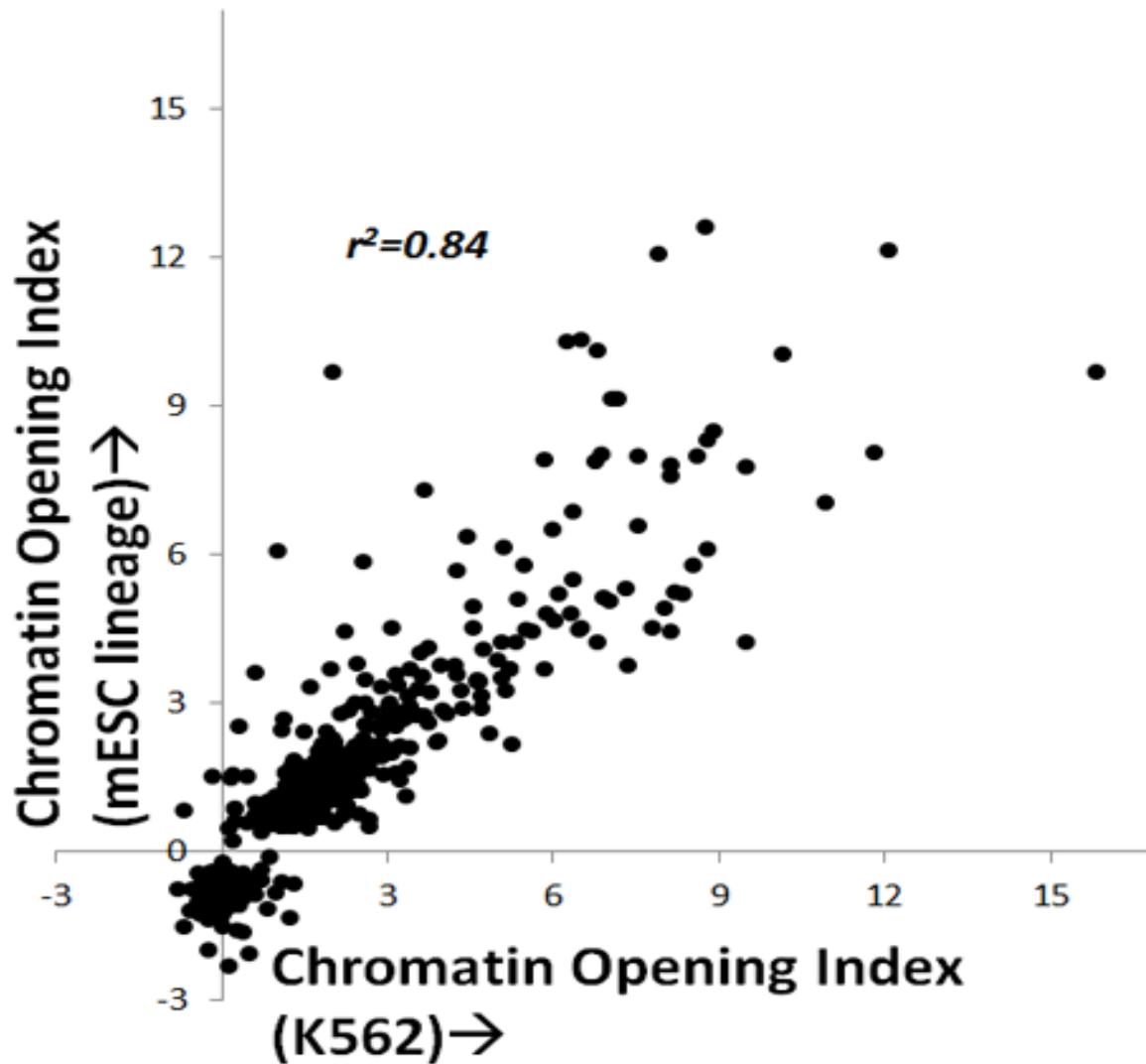


# In vitro reporter assays recapitulate computational predictions

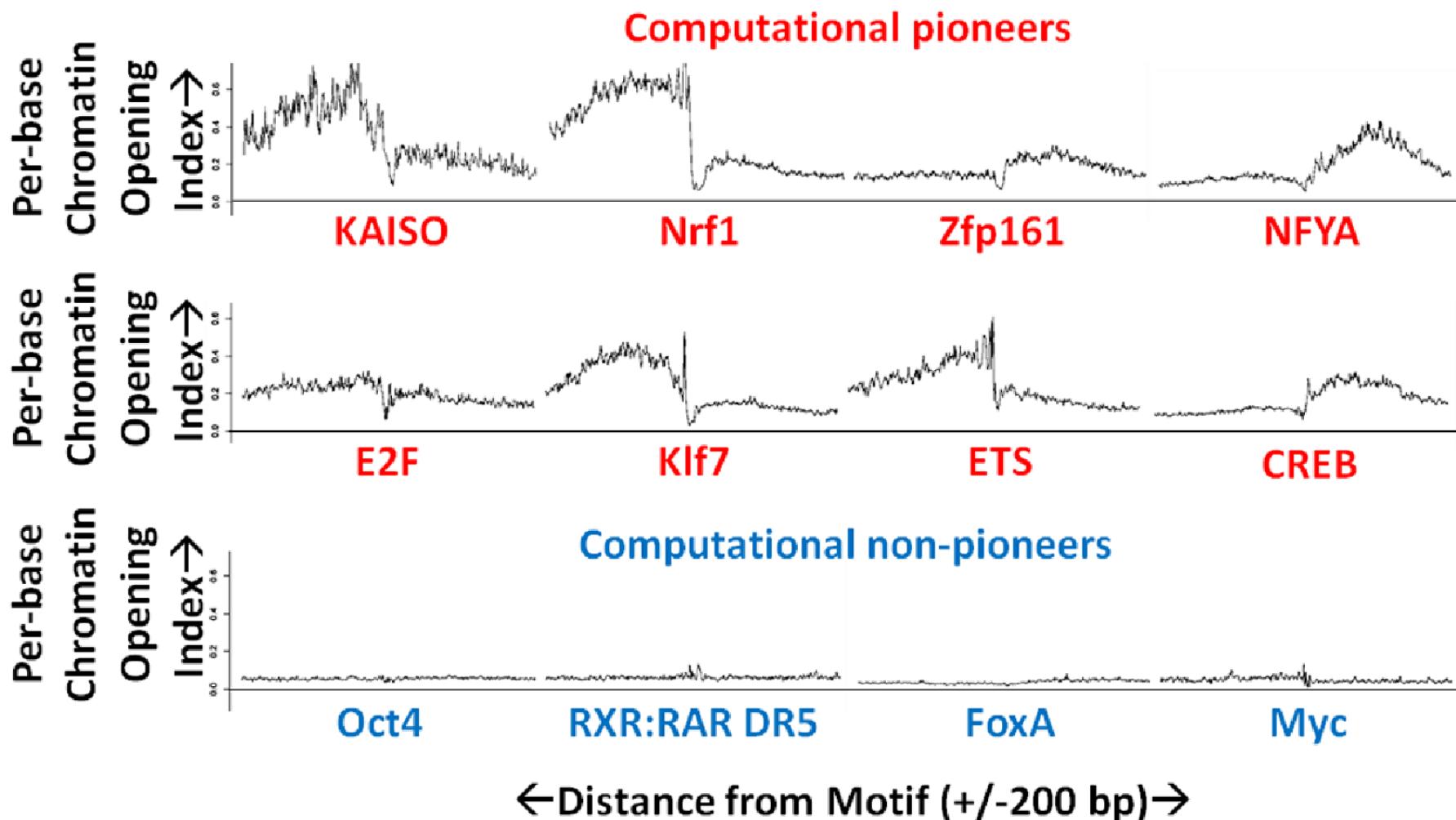
Using a Tol2 based GFP reporter, we confirm finding that these pioneers create new enhancers.



# Pioneers appear to be conserved between human/mouse

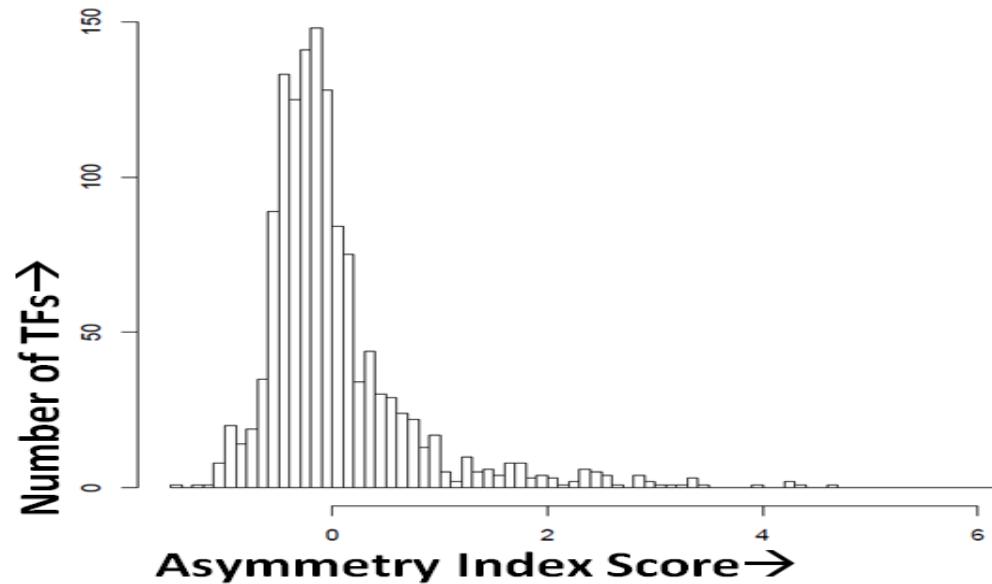


# Pioneer TFs have identifiable profiles

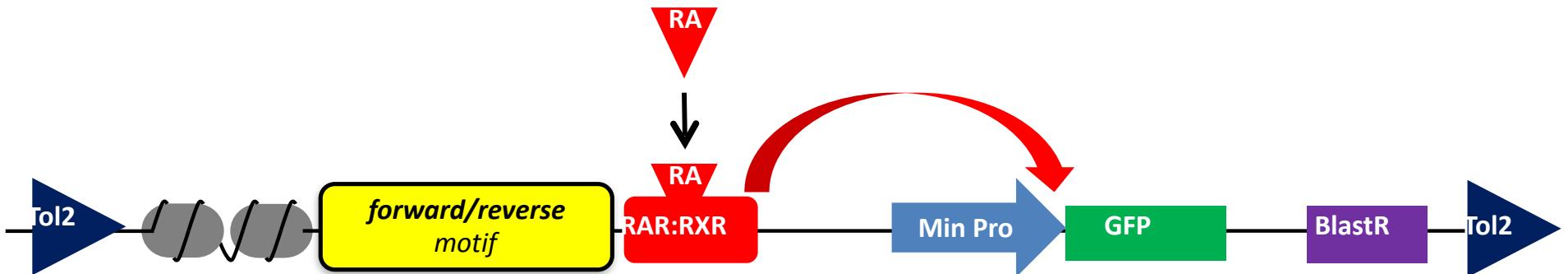


# Certain pioneer TFs are directional

- We define asymmetry index as the expected change between left and right sides in (squared) chromatin opening index score

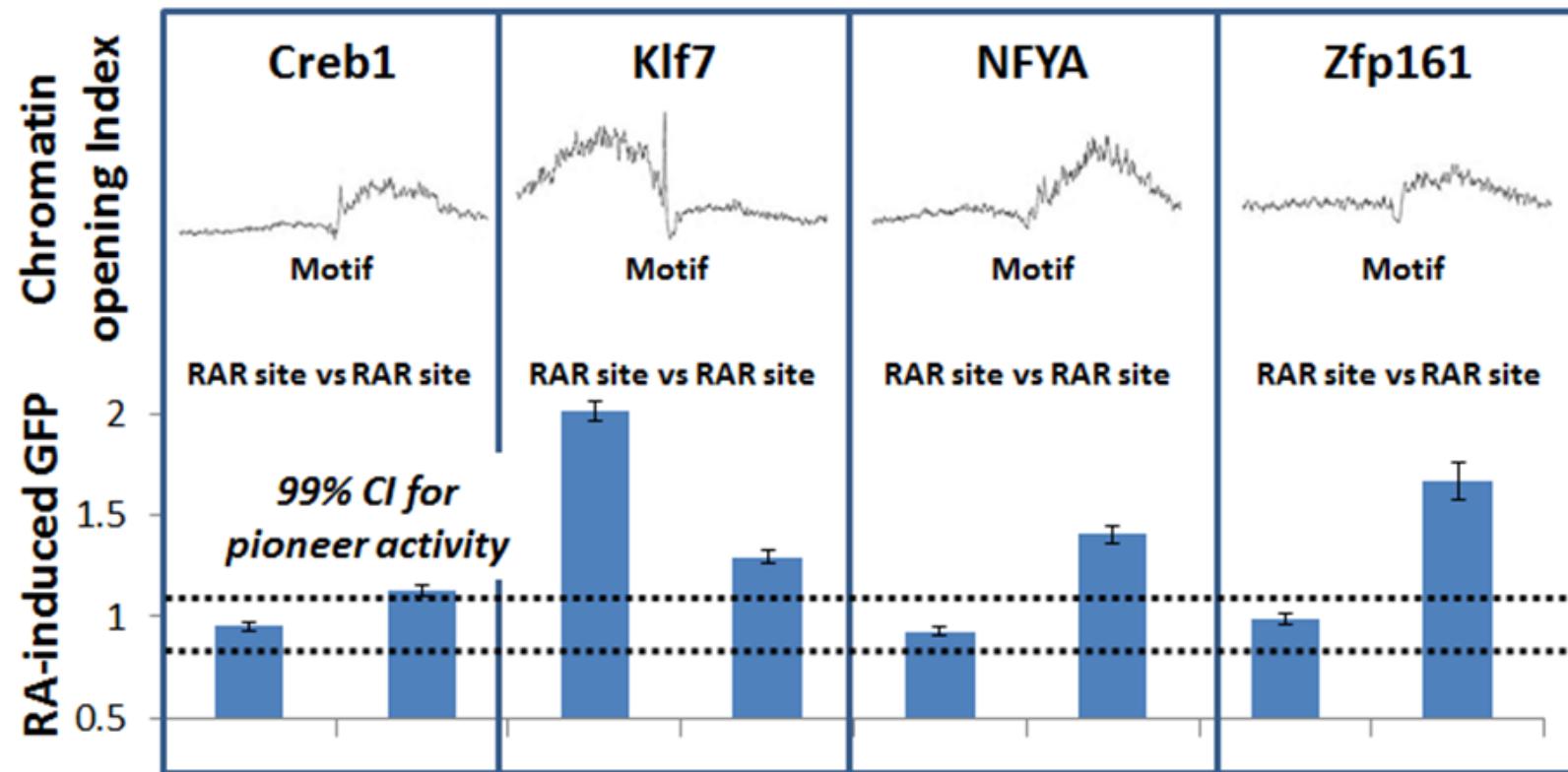


- Biological validation by testing both motif orientations



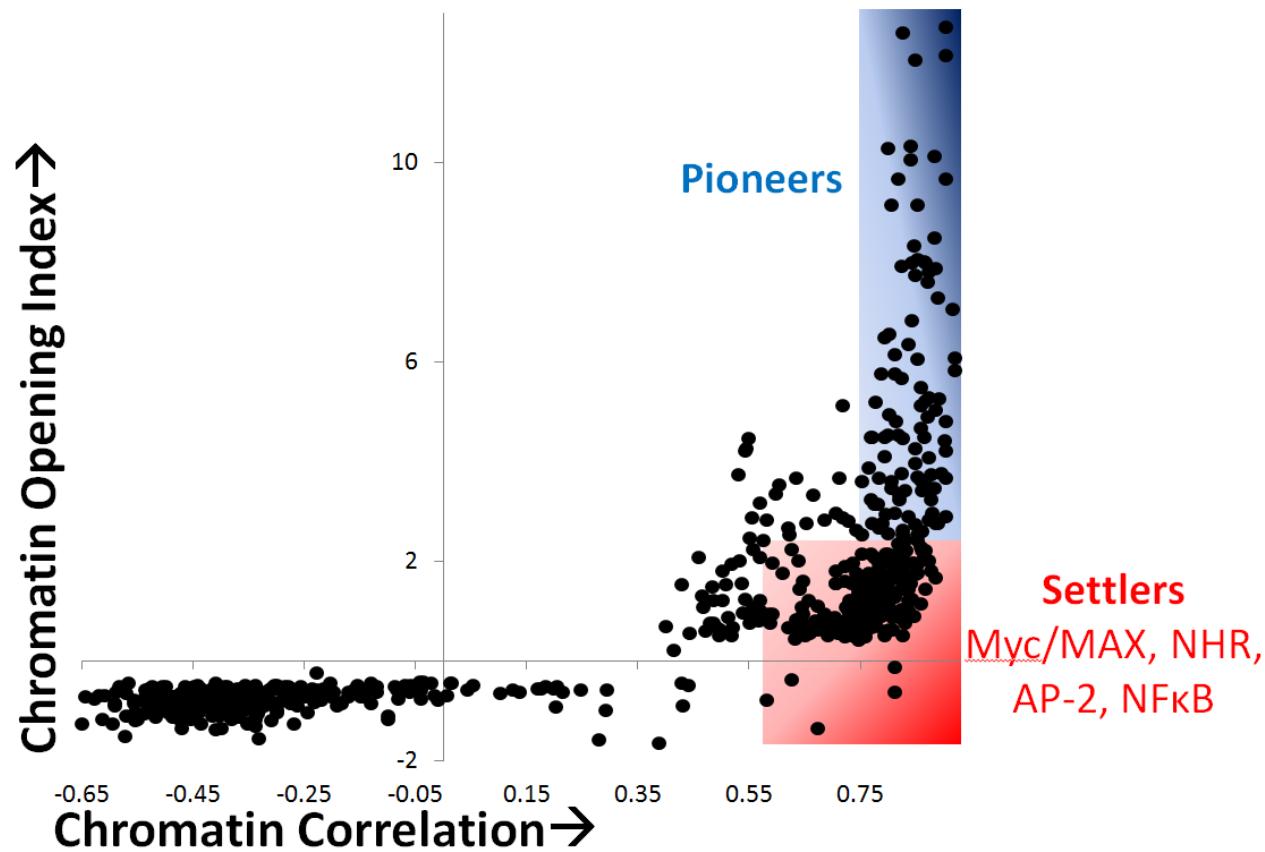
# Certain pioneer TFs are directional

Orienting the motif direction in the reporter recapitulates expected directional behaviors.



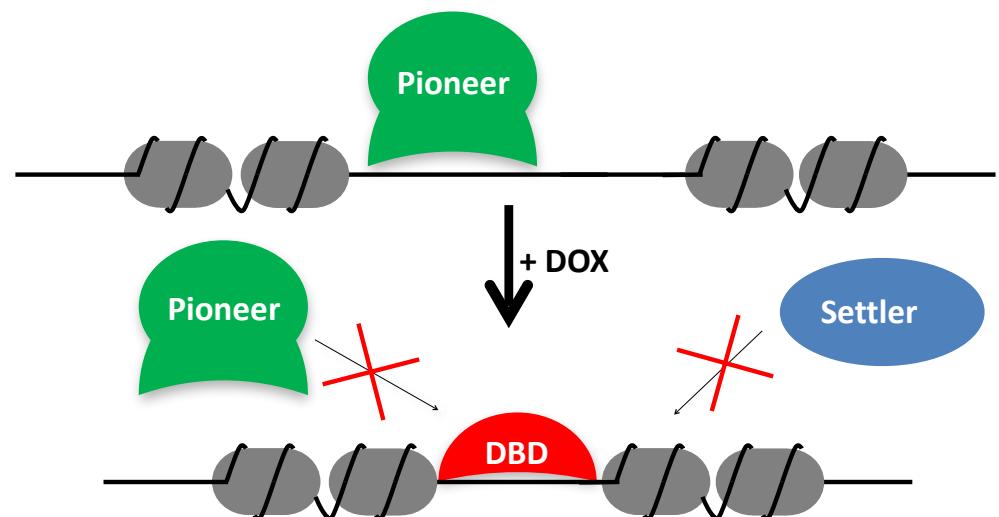
## Settler factors follow pioneer factor binding and loss of pioneer binding causes chromatin to return to a closed state

Pioneers (chromatin opening and dependent) are rare and distinct, while there exists a class of chromatin dependent, but non-opening factors.



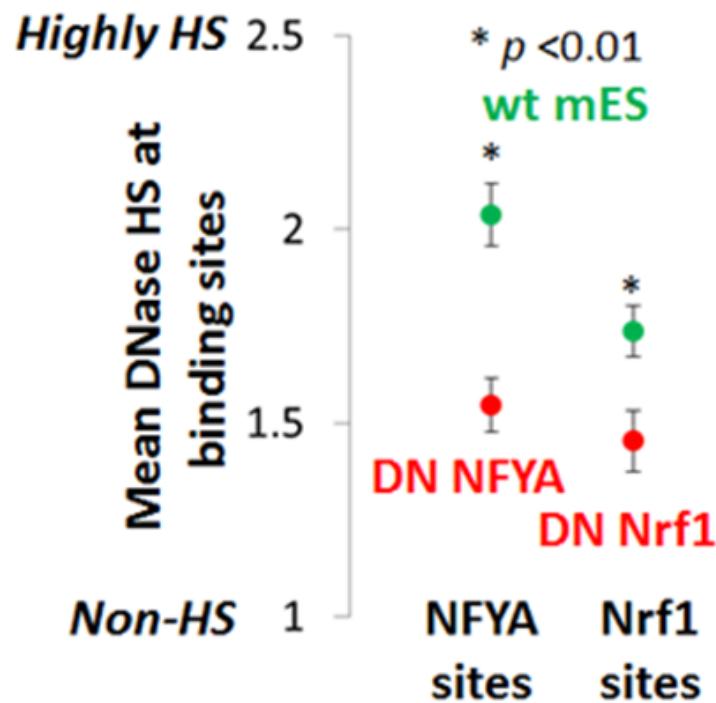
# Validate pioneer/settler model via dominant-negative competition assay

- Construct pioneer DBD protein that retains no pioneering function
- Induction of DBD protein competes for genomic binding, reducing local chromatin accessibility settlers rely on
- Compare proximal chromatin openness
- Compare ChIP levels for neighboring settler binding

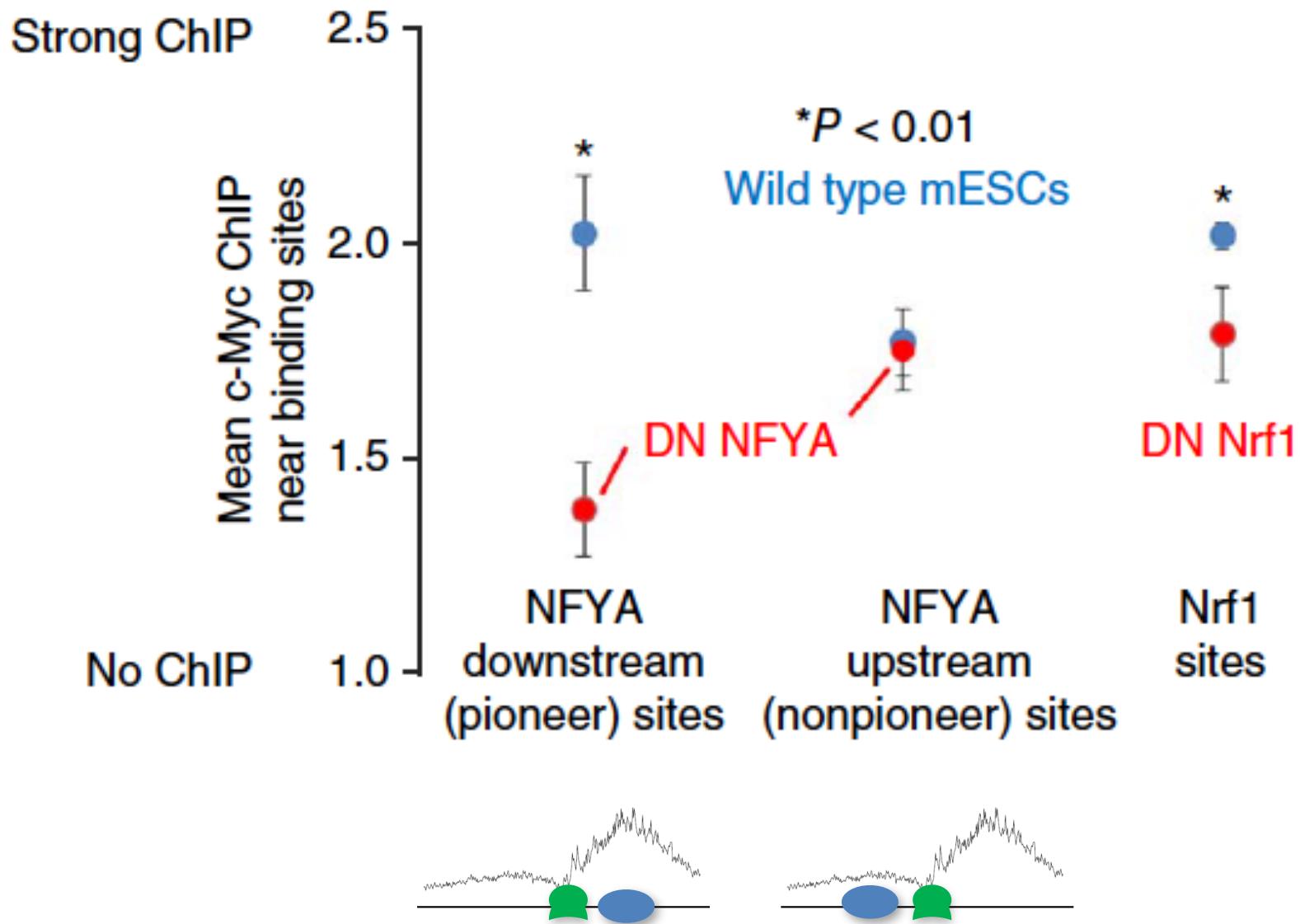


# Dominant negative pioneers reduce proximal DNase HS

We created dominant negative versions of the NFYA and Nrf1 pioneers and measured DNase accessibility at native NFYA and Nrf1 sites after induction of dominant negatives.



# Dominant negative pioneers reduce proximal binding of c-Myc

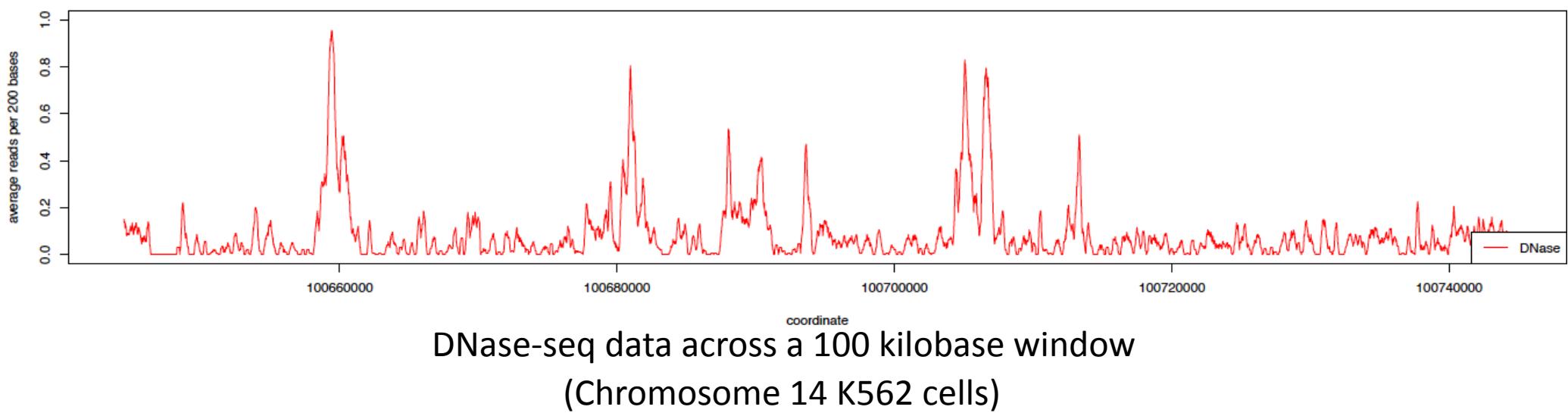


# Chromatin accessibility influences transcription factor binding

- Modeling accessibility profiles yields binding predictions and pioneer factor discovery
- Asymmetric accessibility is induced by *directional pioneers*
- The binding of *settler factors* can be enabled by proximal pioneer factor binding

Sherwood, RI, et al. “Discovery of directional and nondirectional pioneer transcription factors by modeling DNase profile magnitude and shape”  
*Nat. Biotech* 2014.

# Can we predict chromatin accessibility directly from DNA sequence?



Motivation –

1. Understand the fundamental biology of chromatin accessibility
2. Predict how genomic variants change chromatin accessibility

# Can we discover DNA “code words” encoding chromatin accessibility?

- The DNA “code words” encoding chromatin accessibility can be represented by k-mers ( $k \leq 8$ )
- K-mers affect chromatin accessibility locally within  $\pm 1$  kb with a fixed spatial profile
- A particular k-mer produces the same effect wherever it occurs

# We model accessibility read counts with a sum of k-mer influences

The input variable  $c$  is a vector of length  $N$  representing counts and  $c_i$  represents the read-count observed at base  $i$ .

The latent variable  $\lambda$  is a vector of length  $N$  representing the current estimate for  $c$  using  $\theta$ .

$\theta^k$  is the parameter matrix of size  $4^k \times 2M$  associated with the set of all  $k$ -mers.

The variable  $g^k$  is a mapping from genomic coordinate  $i$  to the  $k$  mer starting at  $i$ . The  $k$ -mer for  $g^k$  is represented as an integer that maps to rows of  $\theta$  such that the  $g^k$ th row of  $\theta^k$  is the effect of a  $k$ -mer starting at coordinate  $i$ .

For instance,  $g_i^4$  is the 4-mer starting at coordinate  $i$ . If this is **ATCG**, then the row  $\theta_{g_i^4}^{k_4}$  must be the effect that **ATCG** exerts on its neighbors.

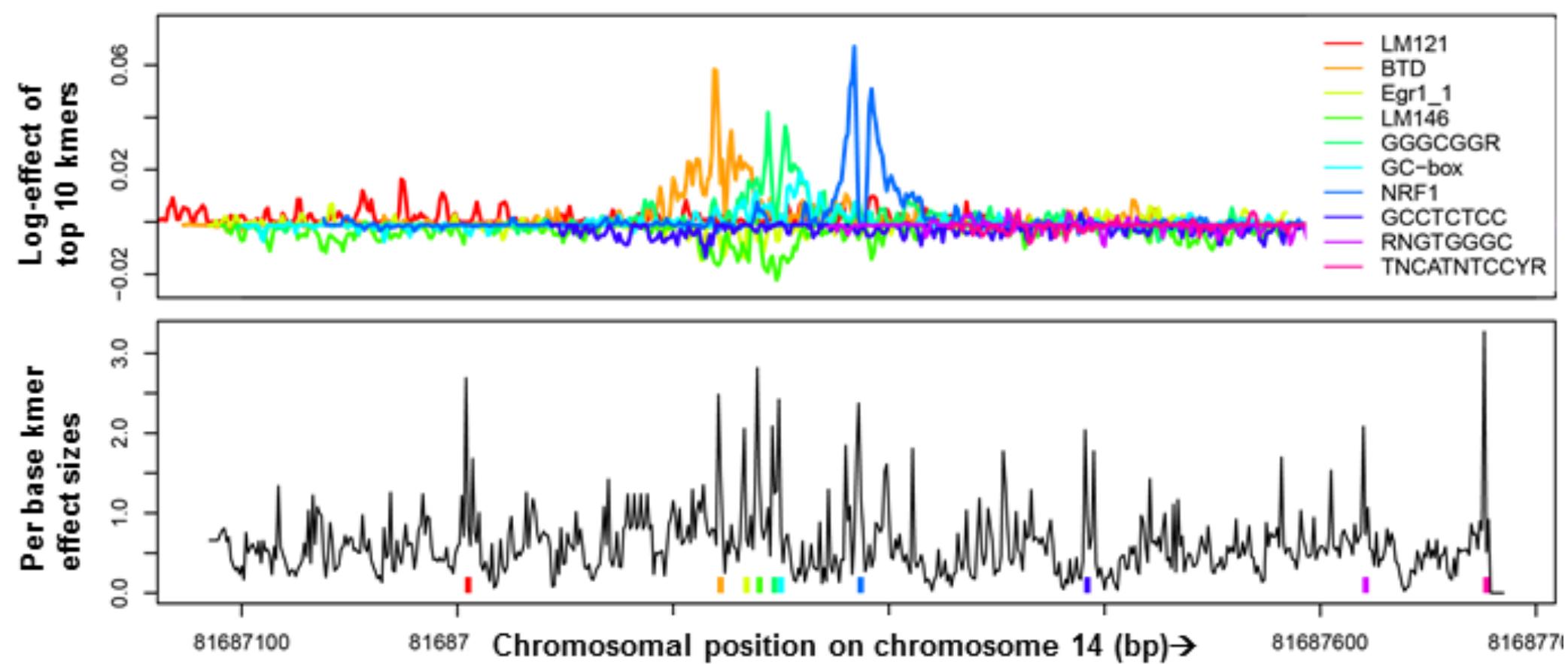
The special parameter  $\theta_0$  is used to set the average read rate of the genome globally.

We model accessibility read counts with a sum of k-mer influences

$$\max_{\theta} \left( \sum_i c_i \log(\lambda_i) - \lambda_i \right) - \eta \sum |\theta^k|_1$$

$$\lambda_i = \exp \left( \left( \sum_{k \in [1..K]} \sum_{j \in [-M, M-1]} \theta_{(g_{i+j}^k - j)}^k \right) - \theta_0 \right)$$

# The Synergistic Chromatin Model (SCM) is a K-mer model



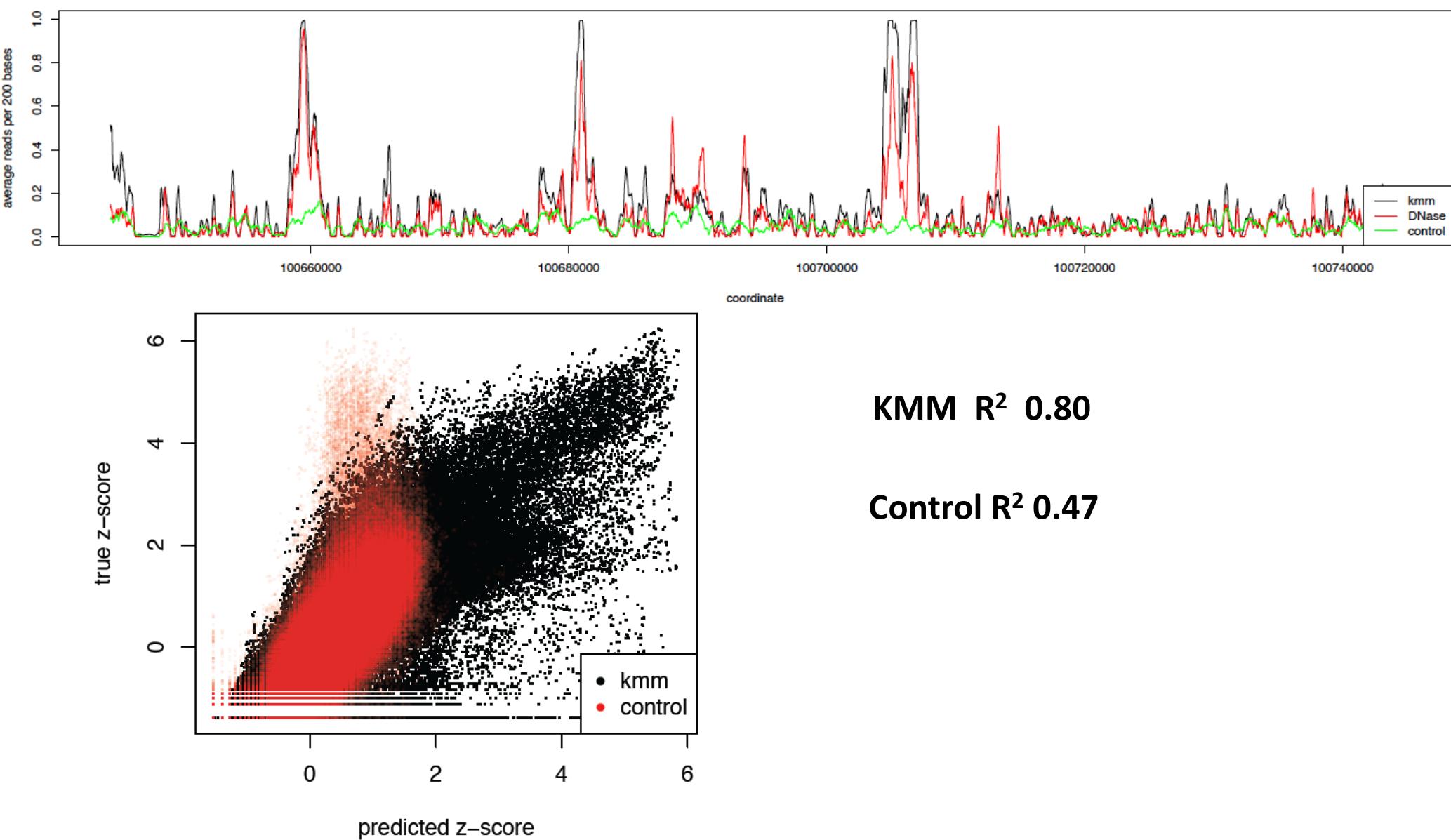
~40,000 K-mers in model

~5,000,000 parameters

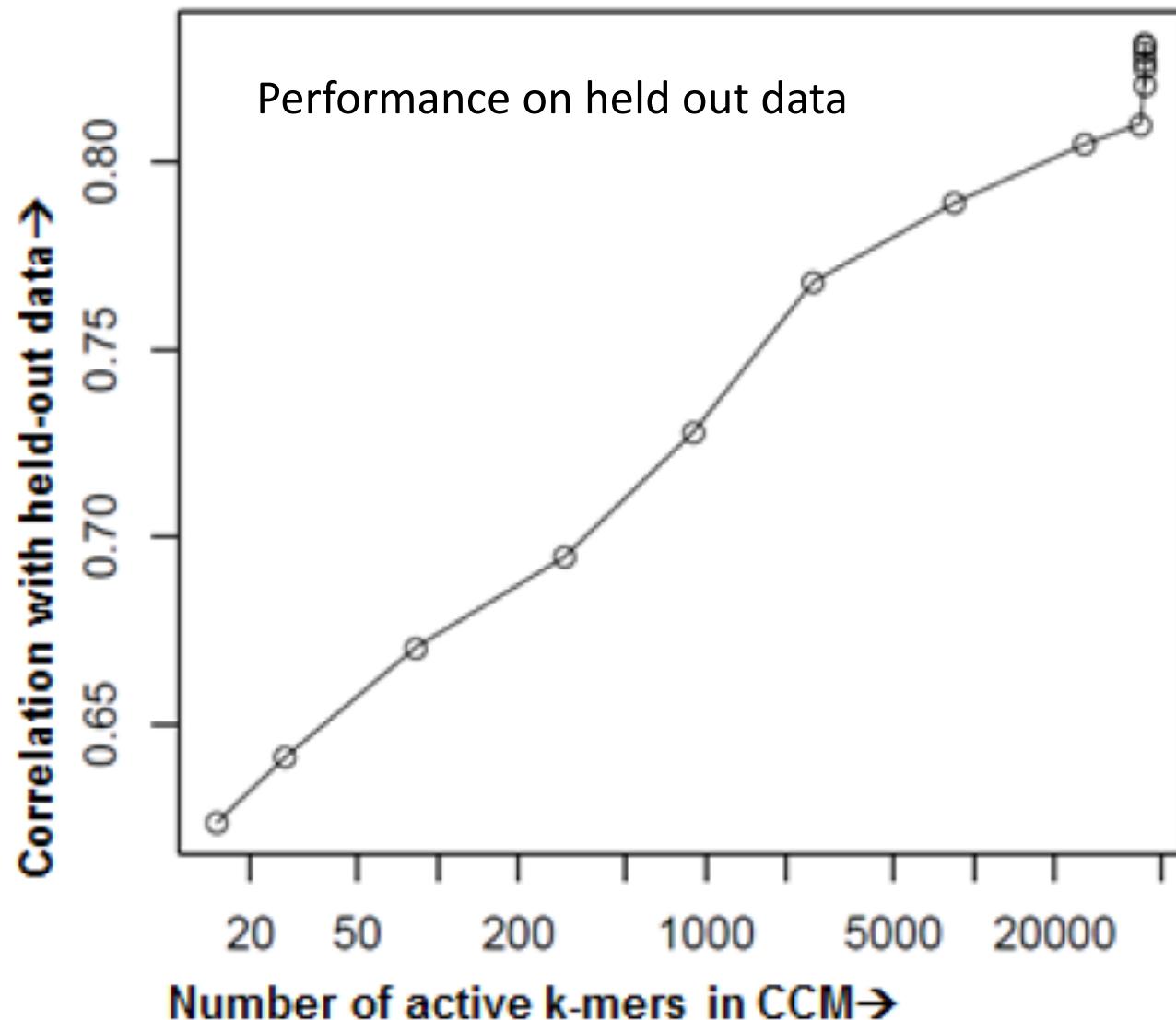
543 iterations \* 360 seconds / iteration \* 40 cores

= ~ 90 days

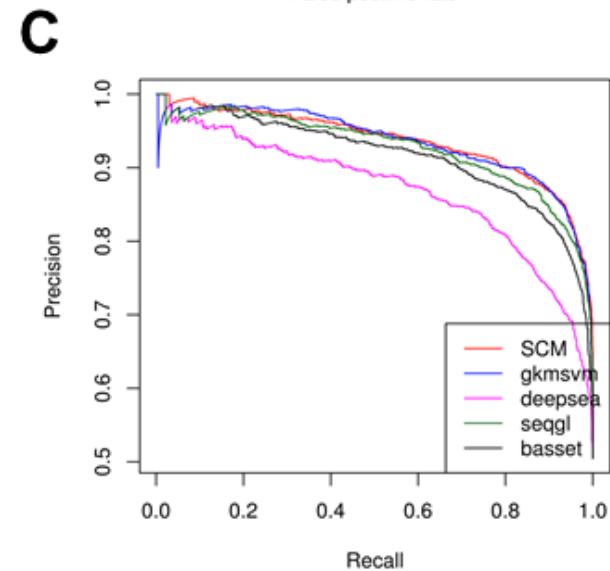
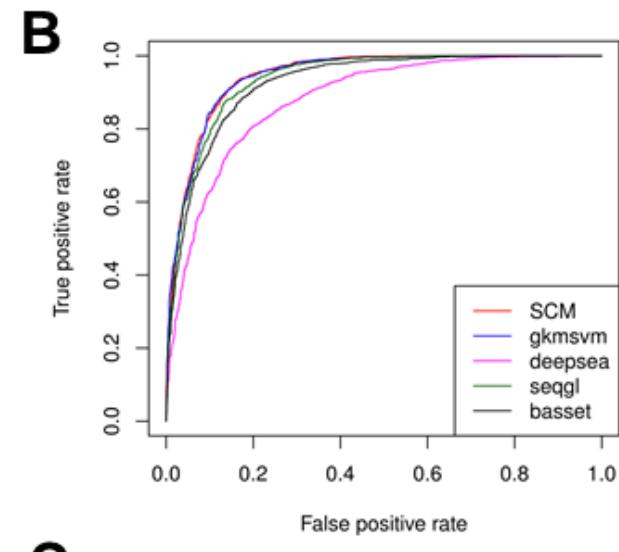
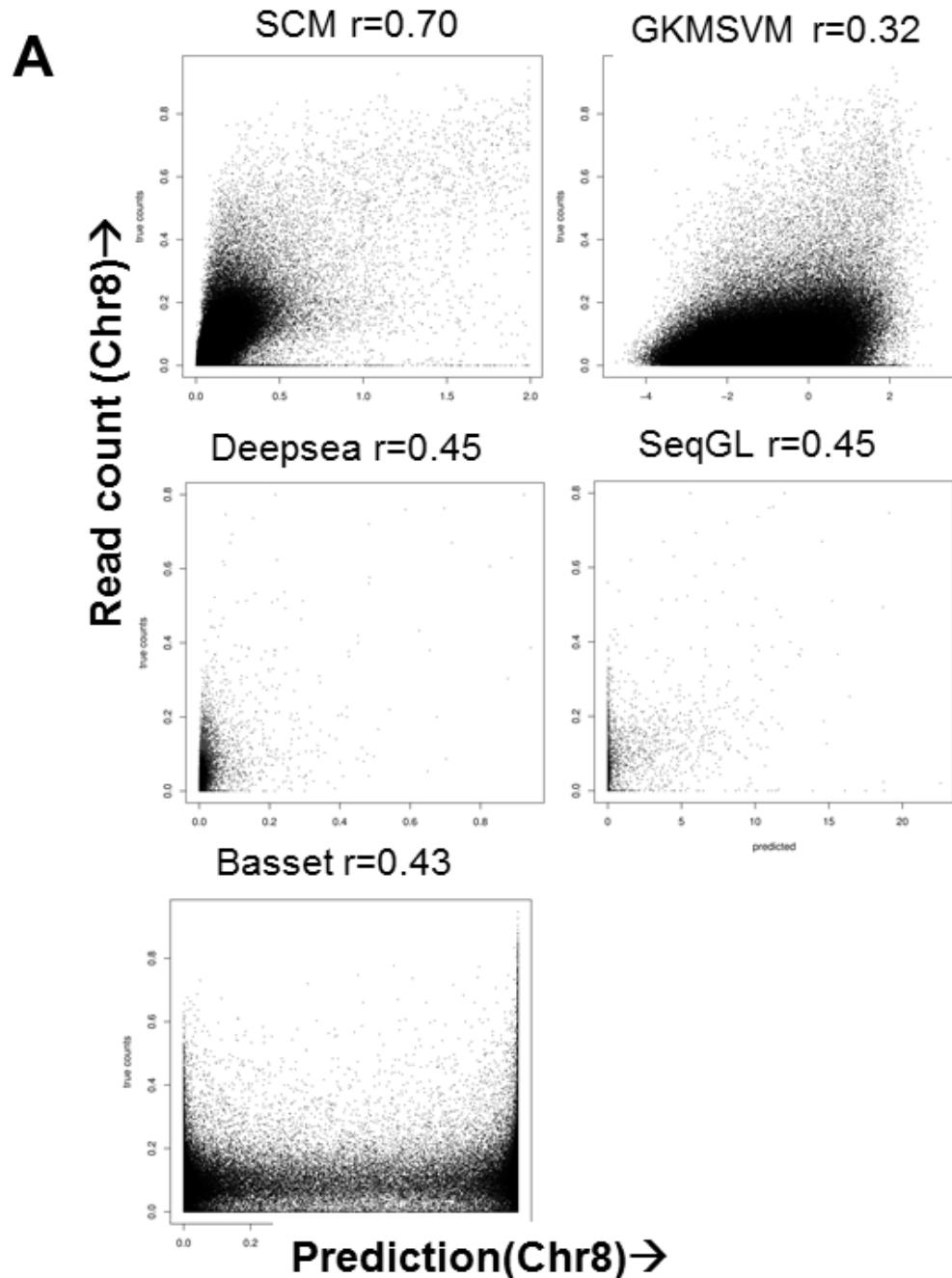
Training on K562 DNase-seq data from chromosomes 1 – 13  
predicts chromosome 14 (black line)



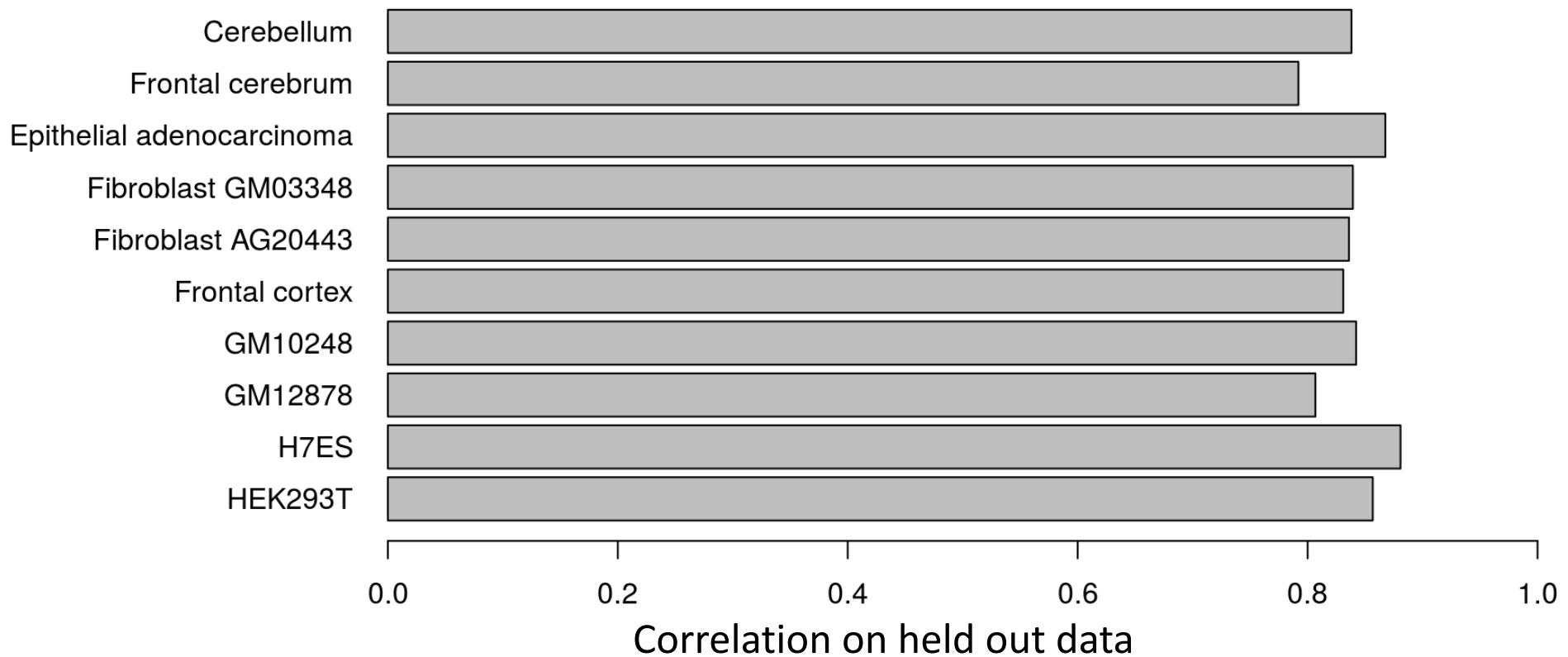
SCM model performance improves with more k-mer features



# SCM outperforms contemporary models at predicting chromatin accessibility from sequence (K562)

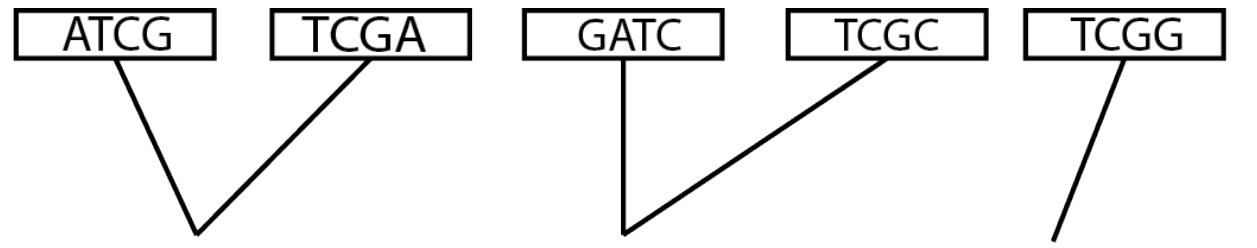


# SCM models have similar predictive power for other cell types

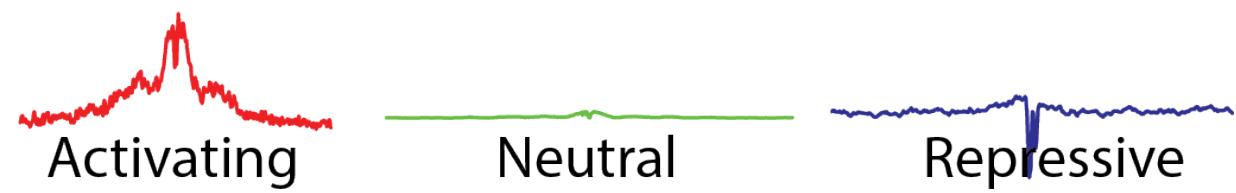


We created synthetic “phrases” each of which contains k-mers that are similar in chromatin opening score

All K-mers



Sort into classes



Construct Debrujin Graph

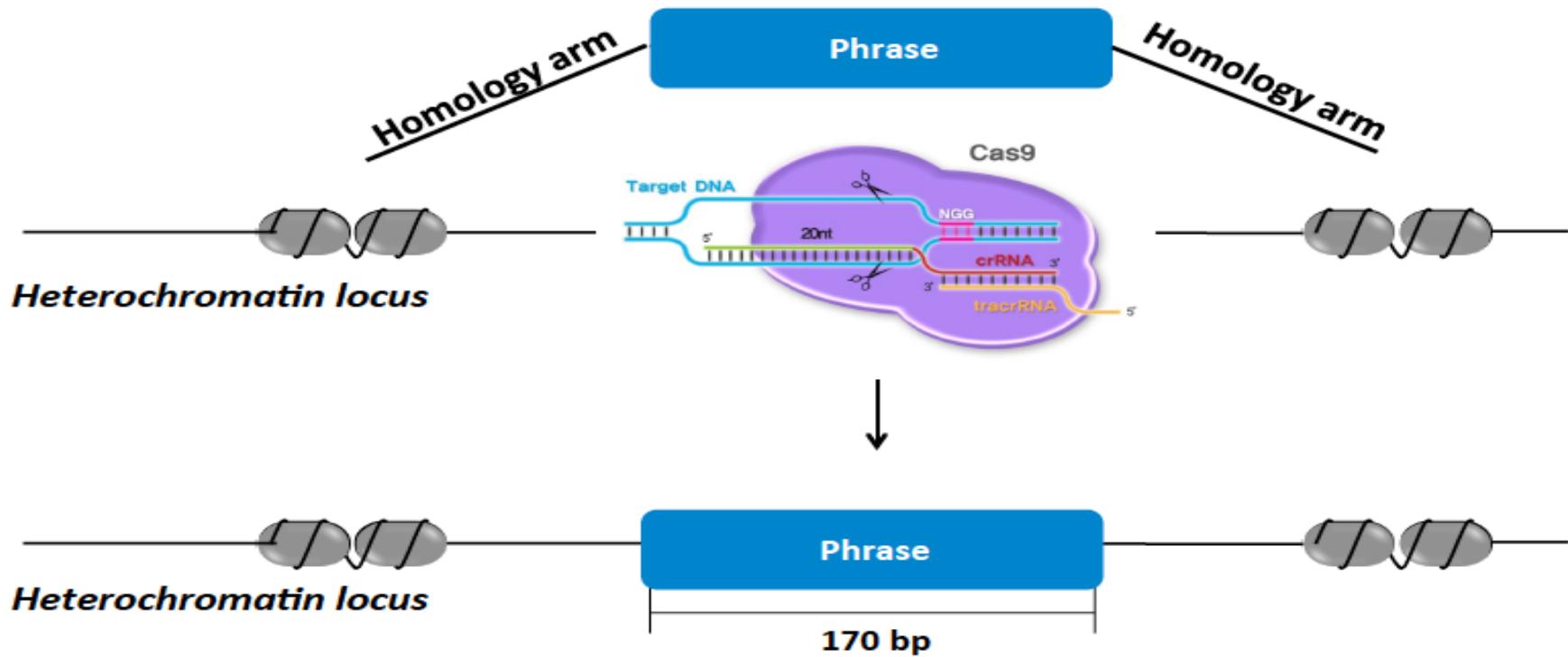


Biased random walk

1. GATCGC
2. GATCGA

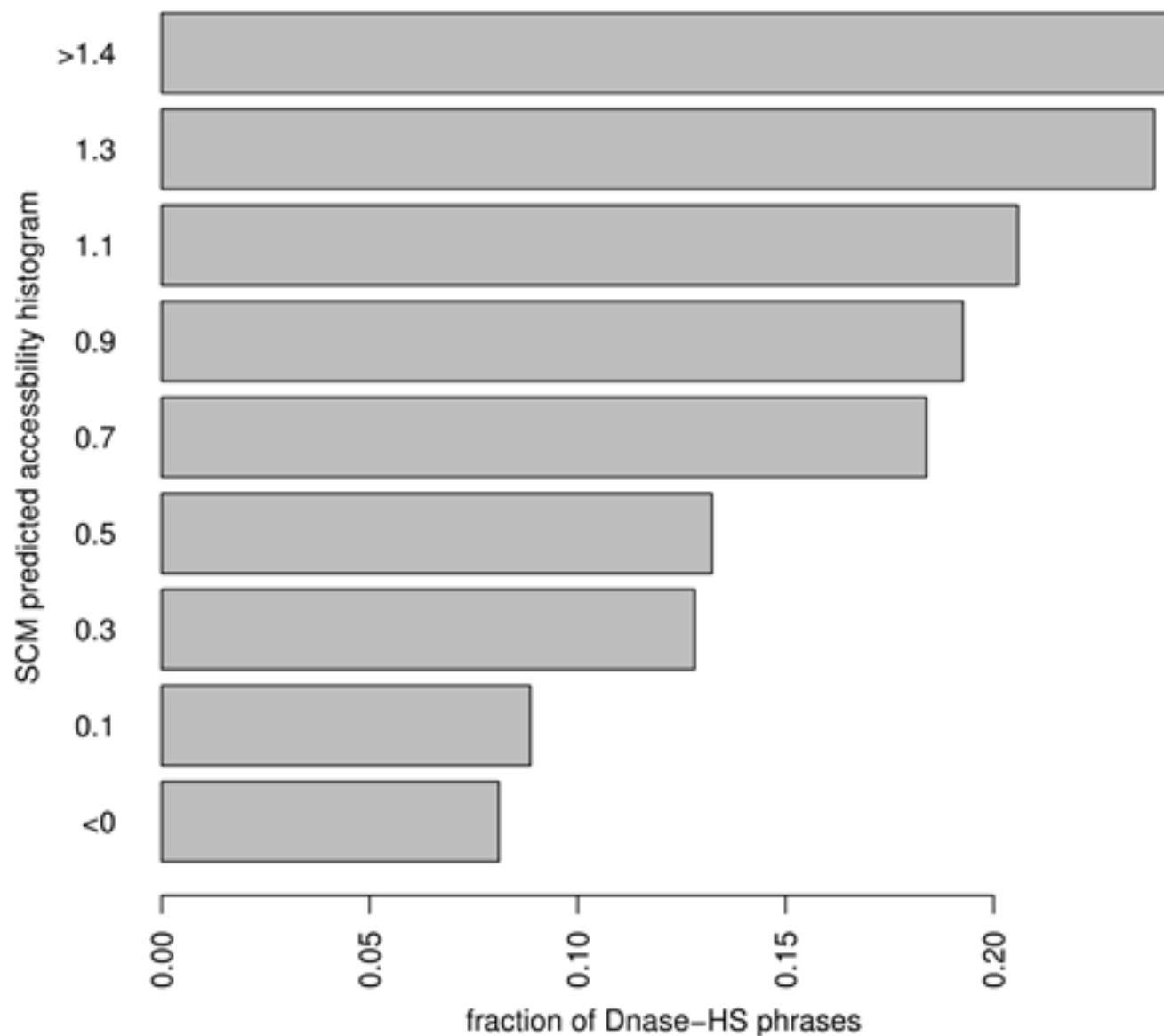
# Single Locus Oligonucleotide Transfer

>6,000 designed phrases into a chromosomal locus

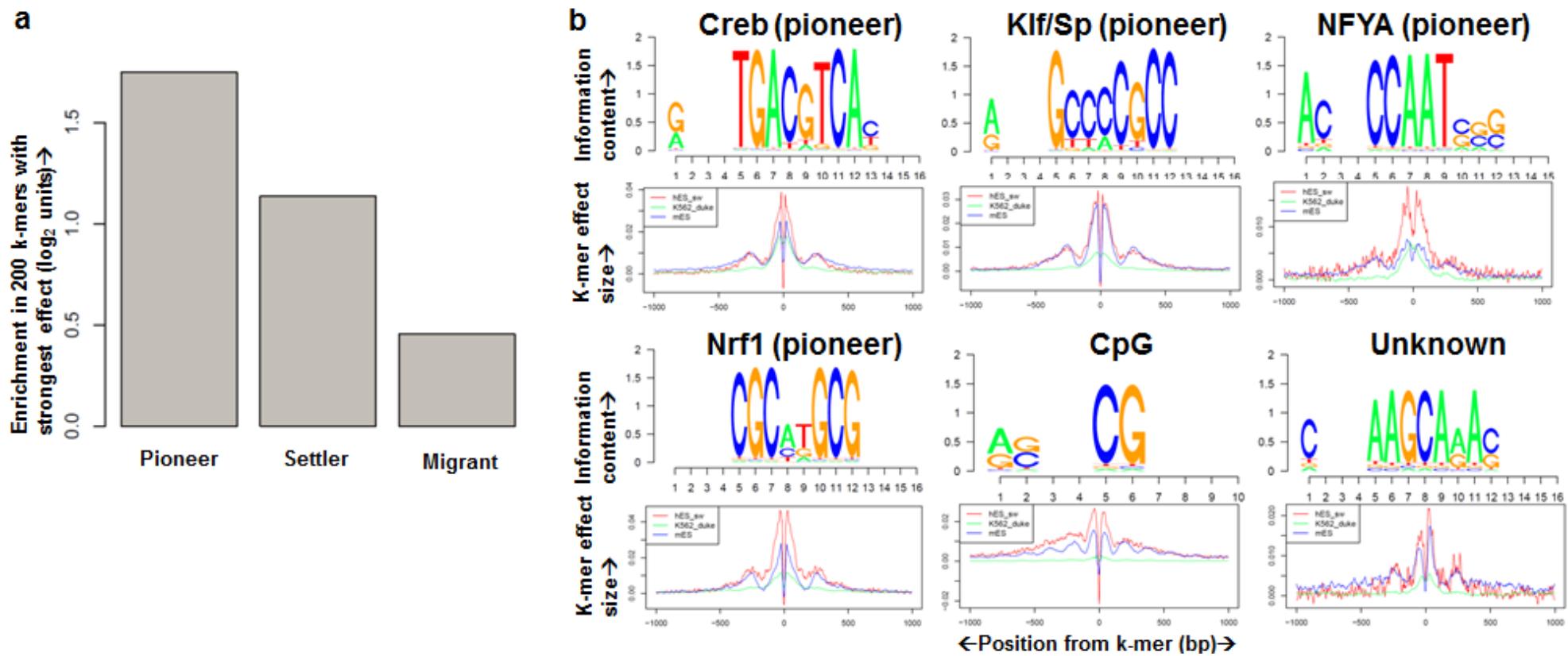


Heterochromatin locus	A	B	C
% alleles with phrase integration	35	15	15
# unique integrations	350,000	150,000	150,000

# Predicted accessibility matches measured accessibility

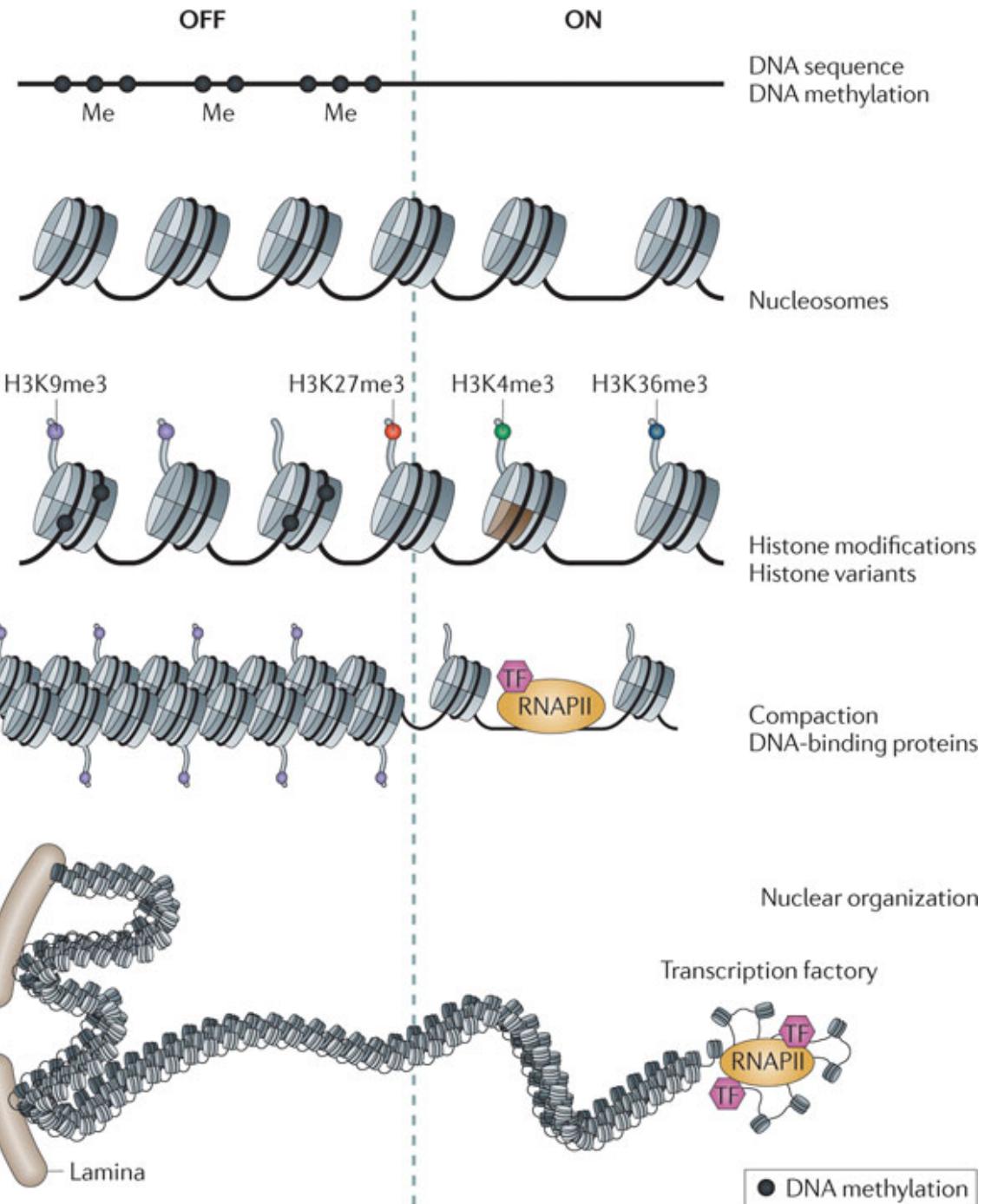


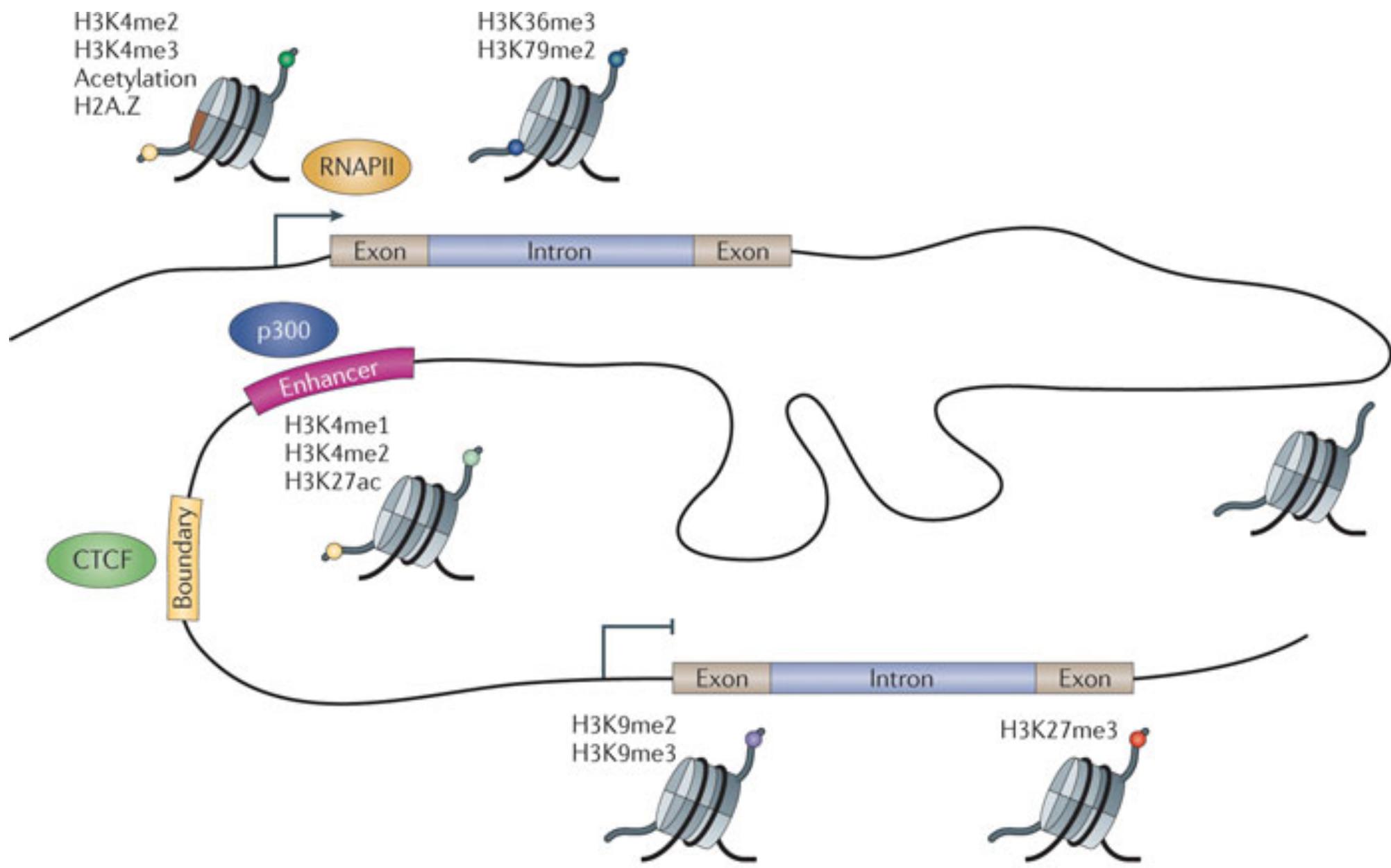
# Chromatin accessibility arises from interactions, largely among pioneer TFs

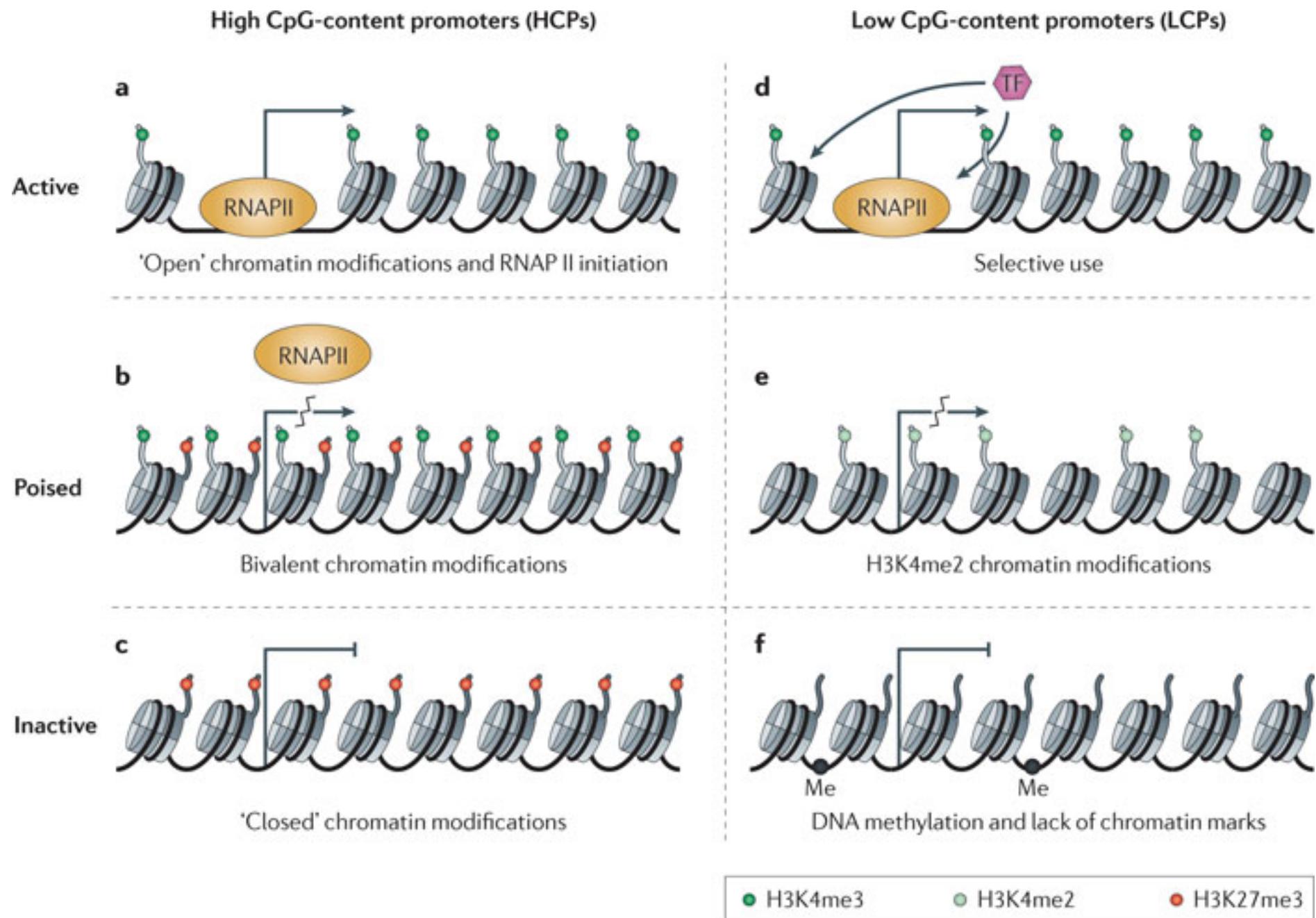


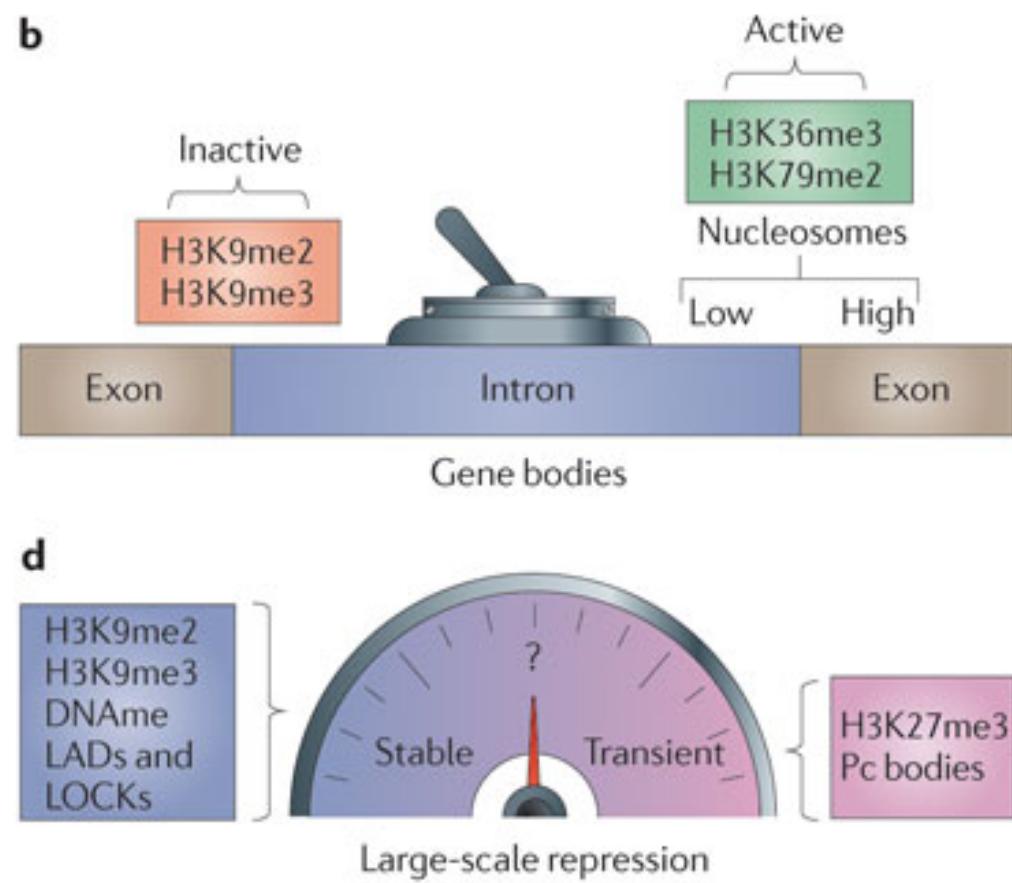
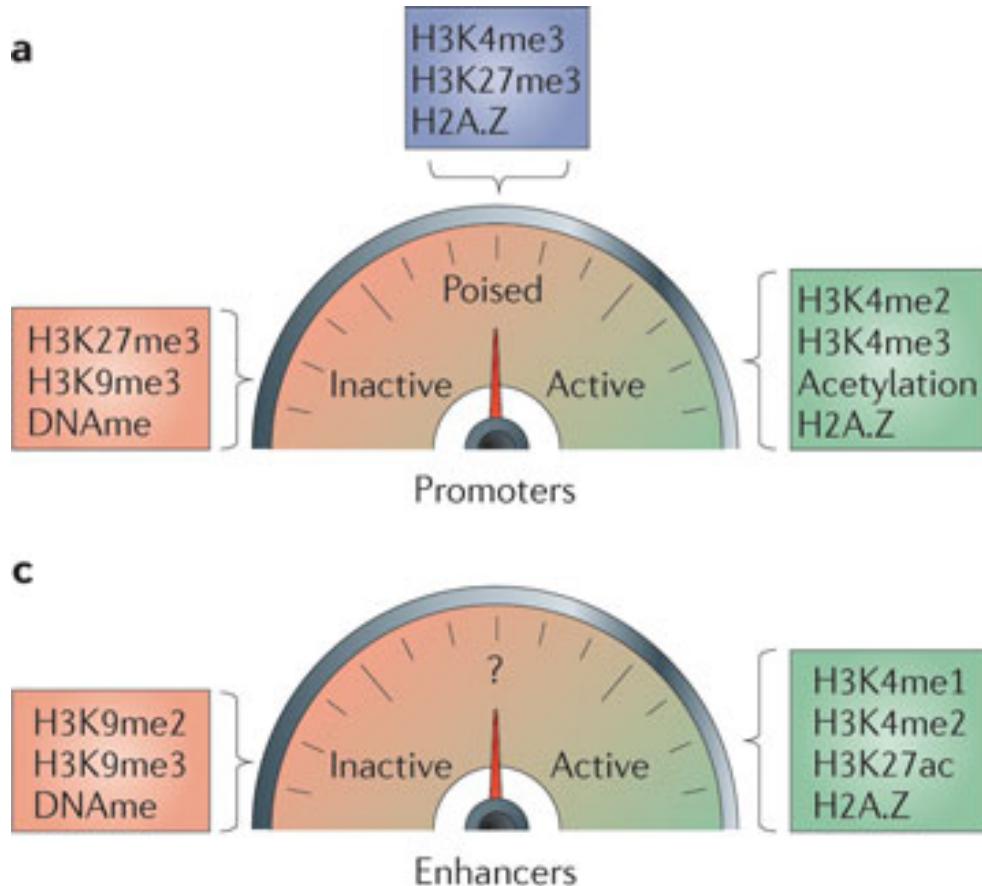
Chromatin organization has multiple structural layers and organizes chromatin into “domains”

Both DNA methylation and chromatin marks contain important functional information









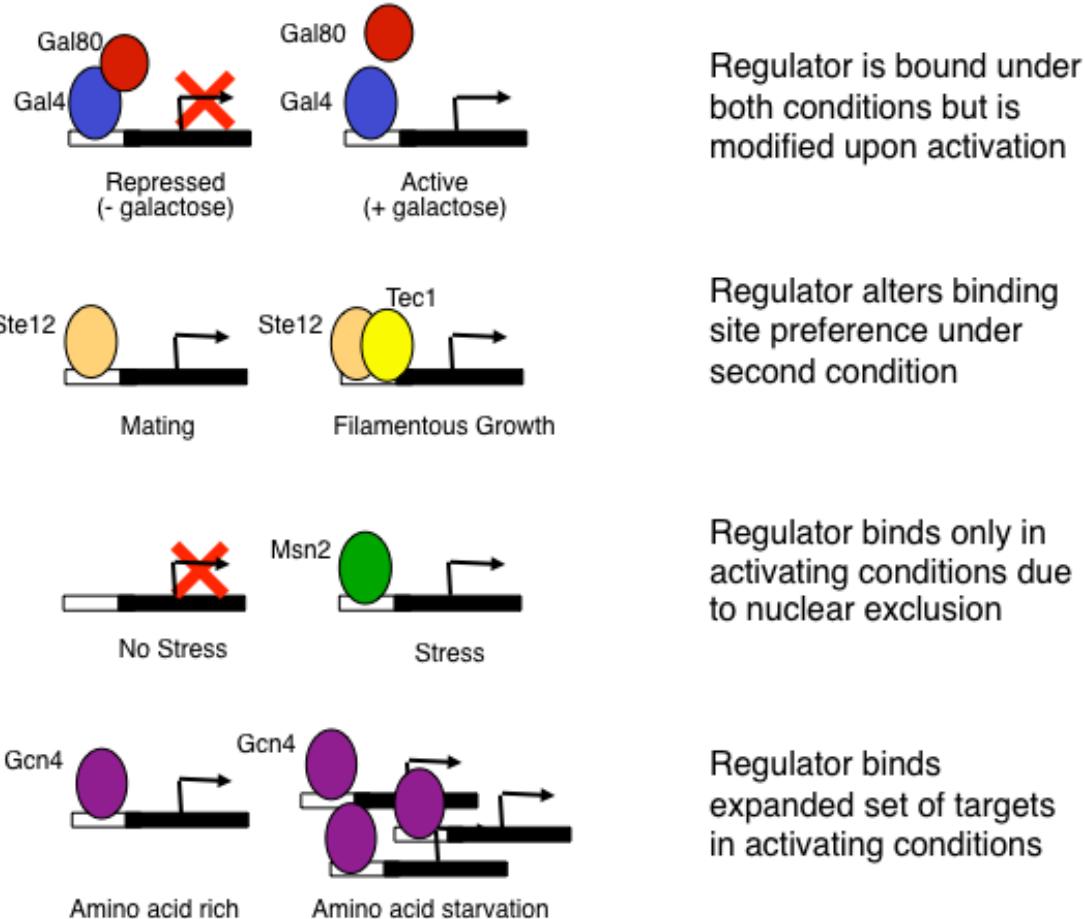
# Part 3 - Modeling Gene Expression



# Modules and modularity

- Modularity guides most engineering systems
  - separates implementation from interface/function
  - creates building blocks, easy combination
- Biological systems are more “integrated” and involve greater robustness (compensation)
- Finding (abstracting) units of (partial) autonomy in cells is therefore a more challenging task
  - the focus on larger groups of genes nevertheless gives us some statistical power

# TF expression, mechanisms



(Young, 2003)

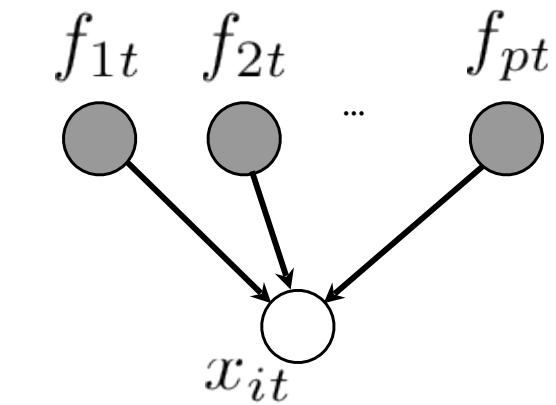
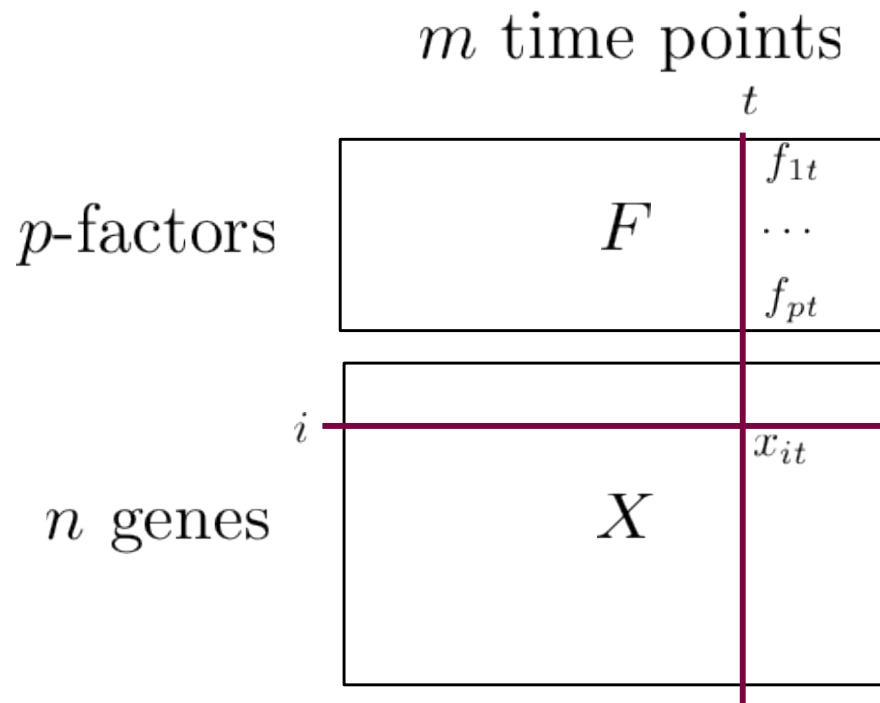


# Expression programs

- We can try to relate the expression of transcription factors (TFs) and genes they regulate
- The key assumption we have to make is that TFs are themselves transcriptionally regulated, i.e., their mRNA levels are indicative of their activity
- The problem:

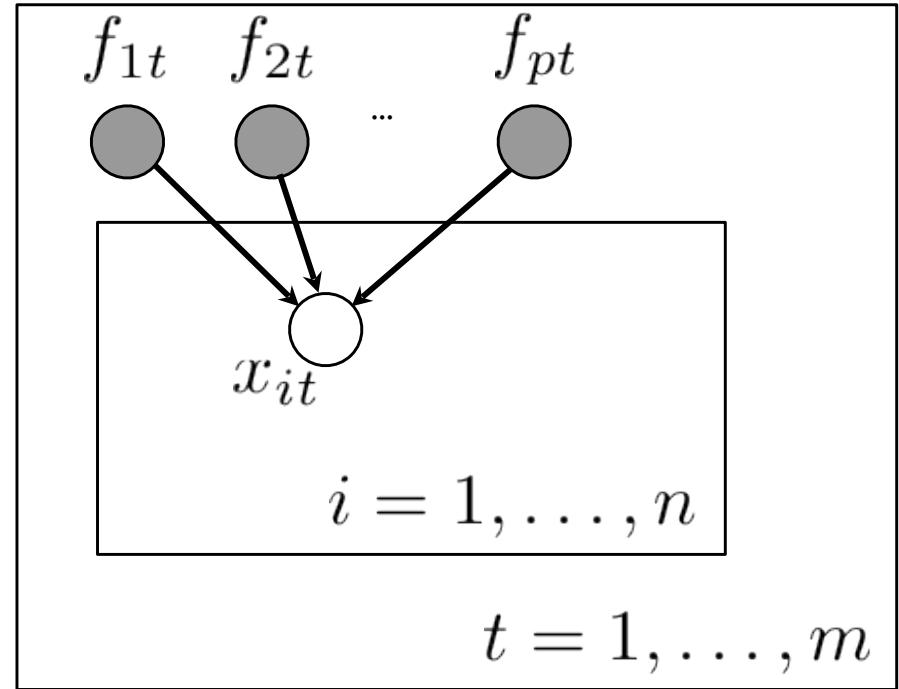
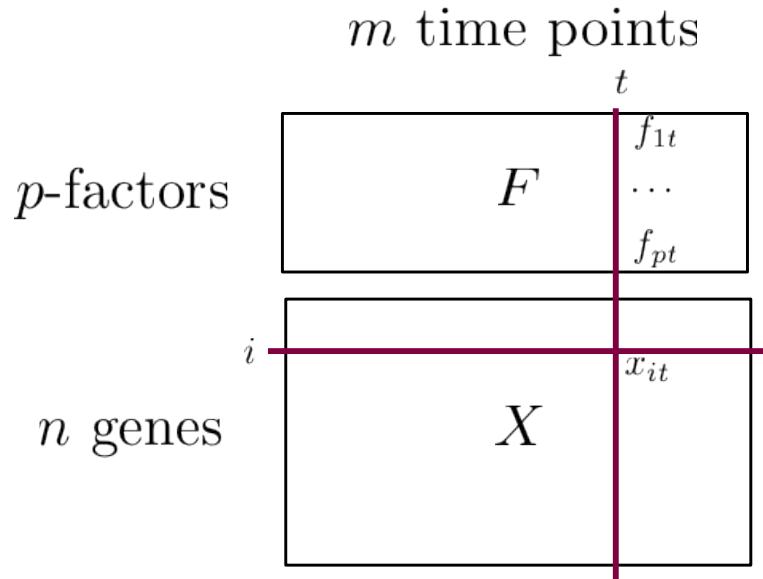
Find a subset of TFs and the genes they regulate along with the function (program) that relates their expression

# A regression model



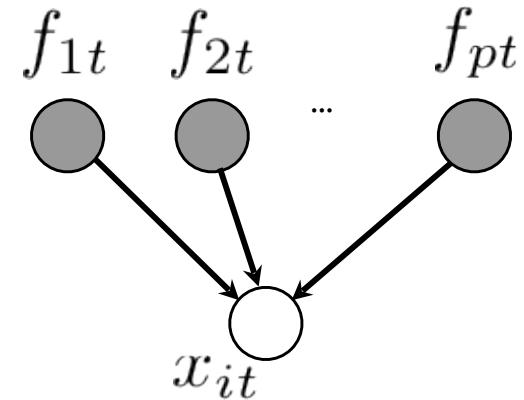
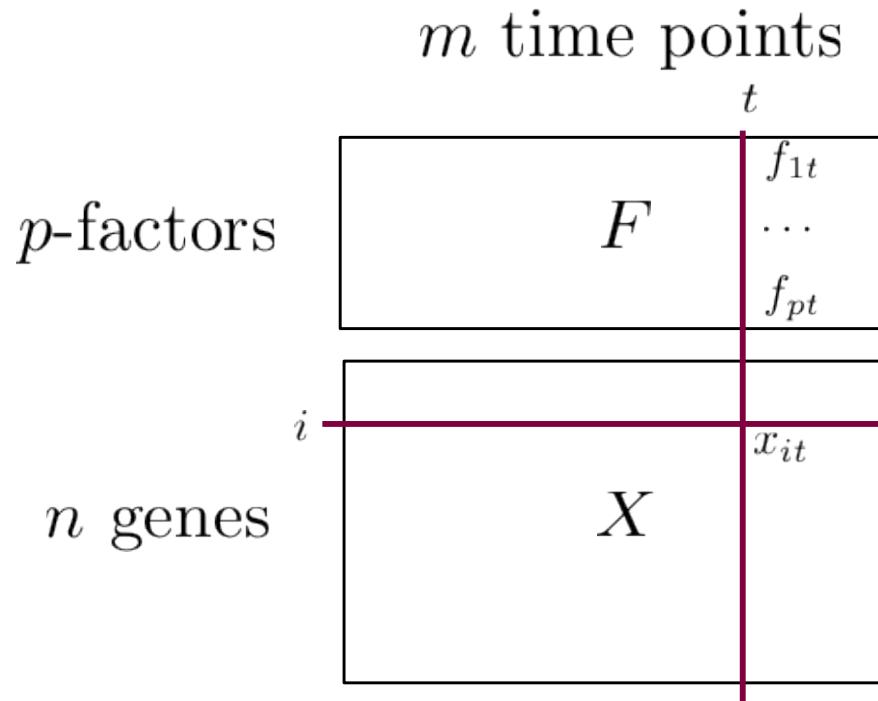
$$P(x_{it} | f_{1t}, \dots, f_{pt}, \Theta)$$

# A regression model: sampling



$$P(X|F, \Theta_1) = \prod_{t=1}^m \left[ \prod_{i=1}^n P(x_{it}|f_{1t}, \dots, f_{pt}, \Theta_1) \right]$$

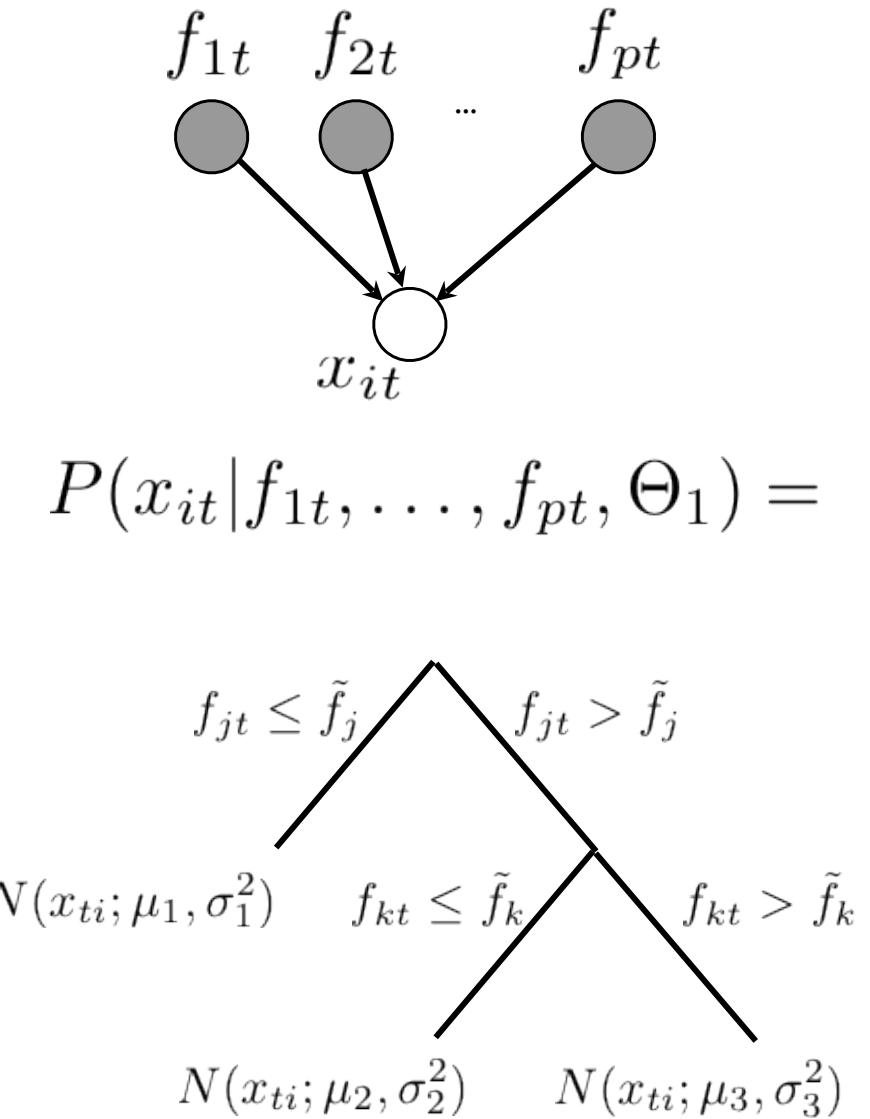
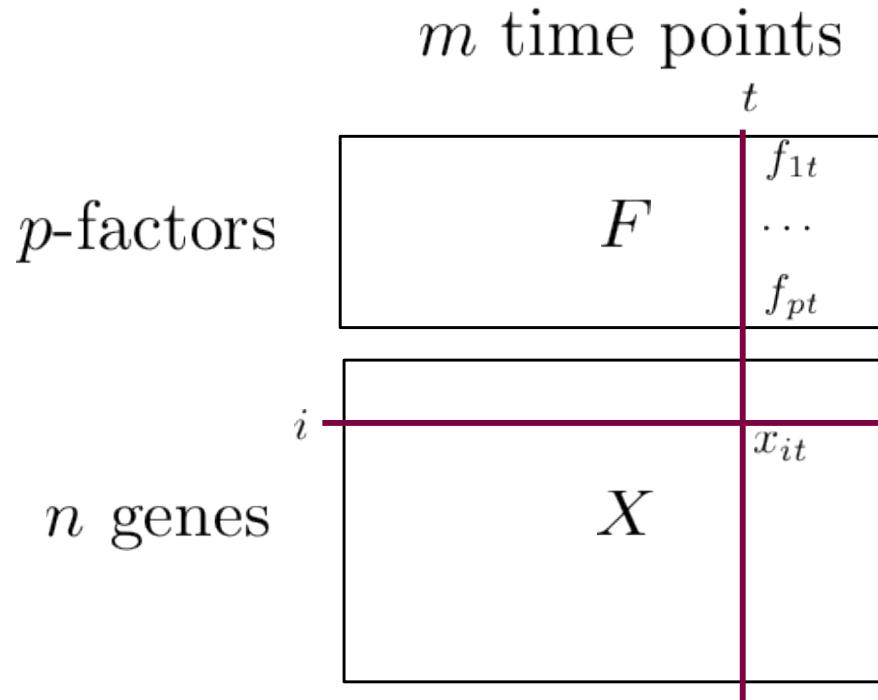
# Possible regression models



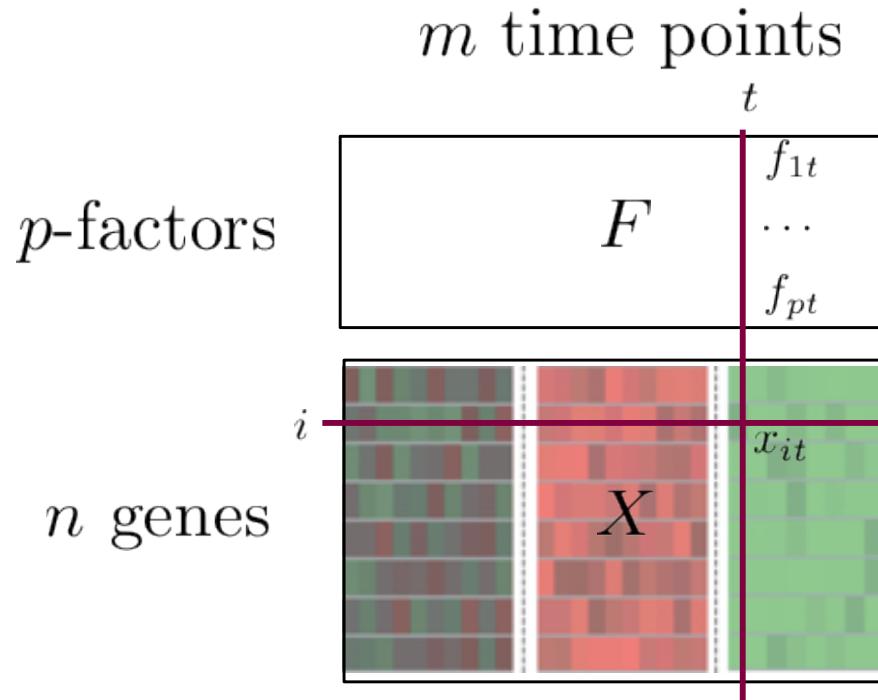
$$P(x_{it} | f_{1t}, \dots, f_{pt}, \Theta_1) =$$

- Linear regression (with feature selection)
- Regression tree
- Discretization + conditional table
- etc.

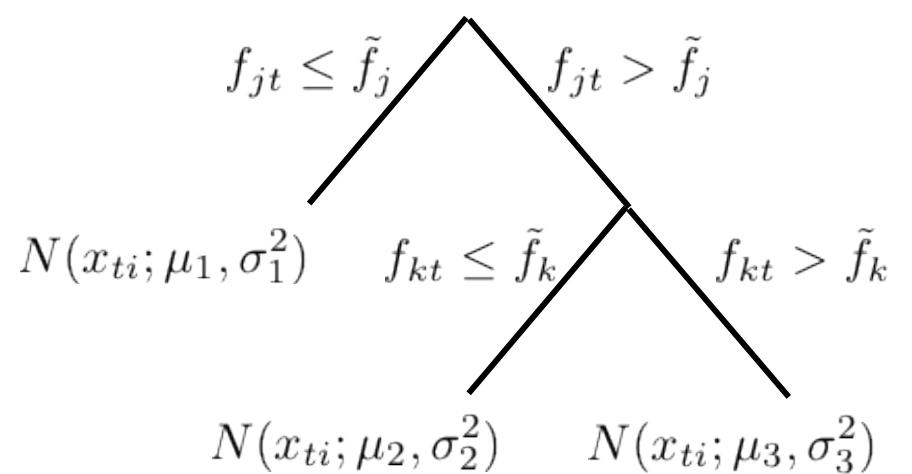
# A regression tree



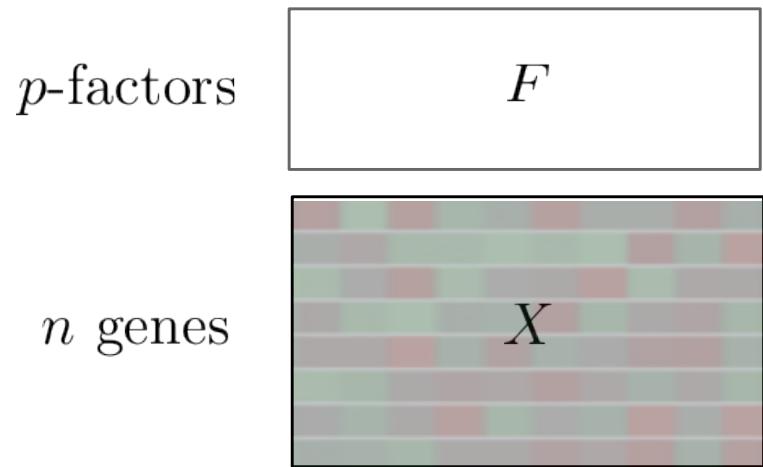
# A regression tree



$$P(x_{it} | f_{1t}, \dots, f_{pt}, \Theta_1) =$$



# Learning regression trees: null model

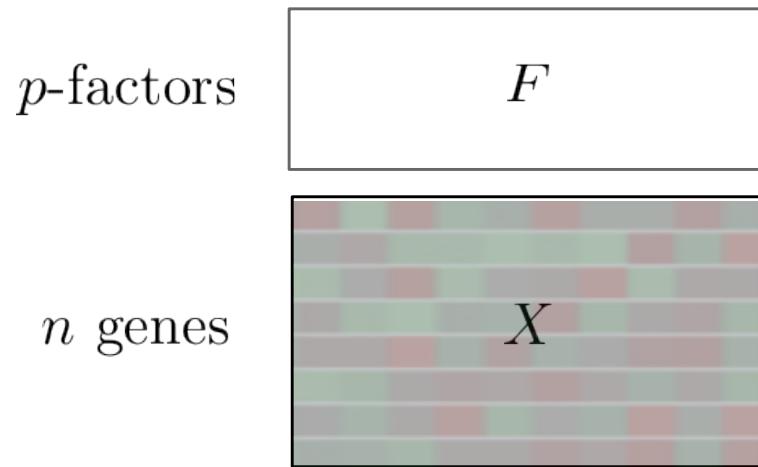


- The simplest case (null model): no dependence on the factors

$$P(X; \theta) = \prod_{i=1}^n \prod_{t=1}^m N(x_{it}; \mu, \sigma^2)$$

where  $\theta = \{\mu, \sigma^2\}$ .

# Learning regression trees: null model



- The simplest case (null model): no dependence on the factors

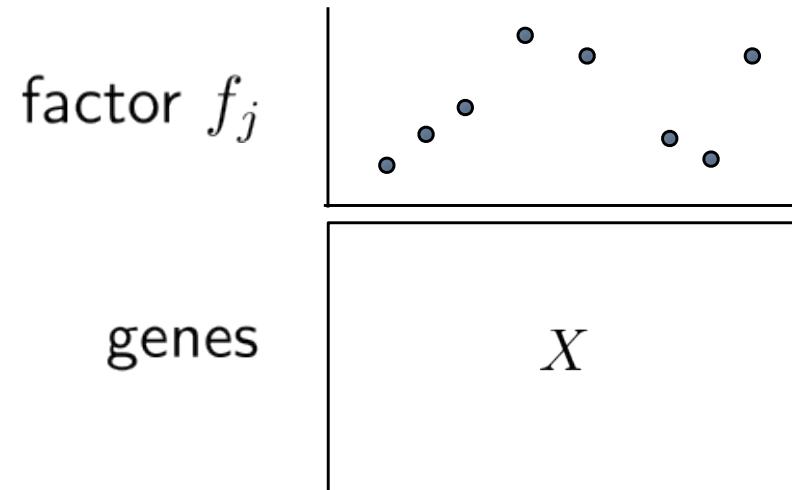
$$P(X; \theta) = \prod_{i=1}^n \prod_{t=1}^m N(x_{it}; \mu, \sigma^2)$$

where  $\theta = \{\mu, \sigma^2\}$ .

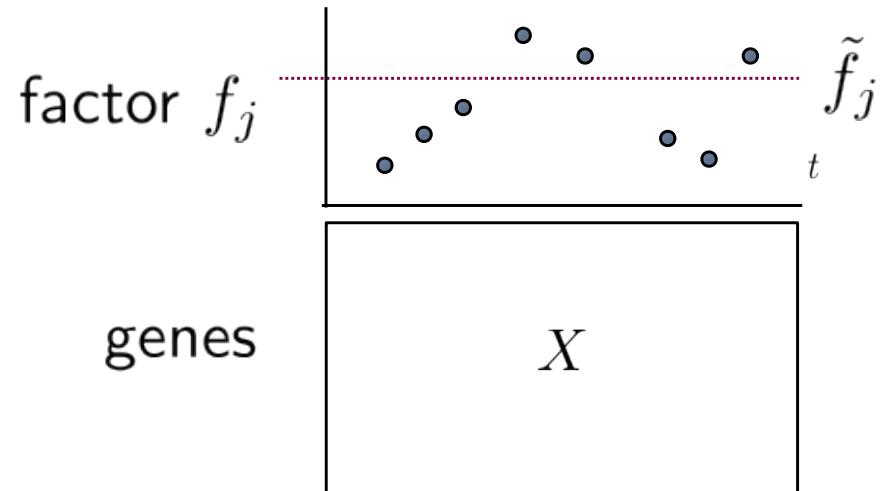
- By maximizing  $l(X; \theta) = \log P(X; \theta)$  with respect to the parameters (mean and variance), we get

$$l(X; \hat{\theta}) = nm \left[ -\frac{1}{2} - \frac{1}{2} \log(2\pi\hat{\sigma}^2) \right]$$

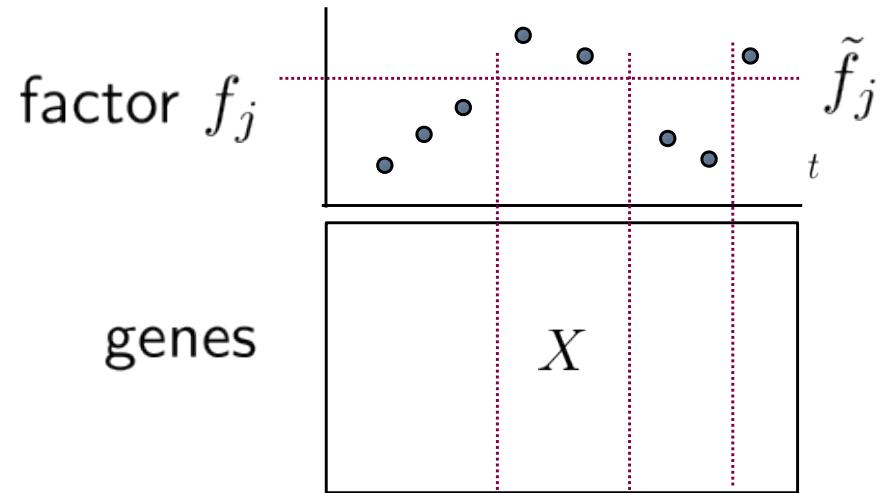
# Learning regression trees: one factor



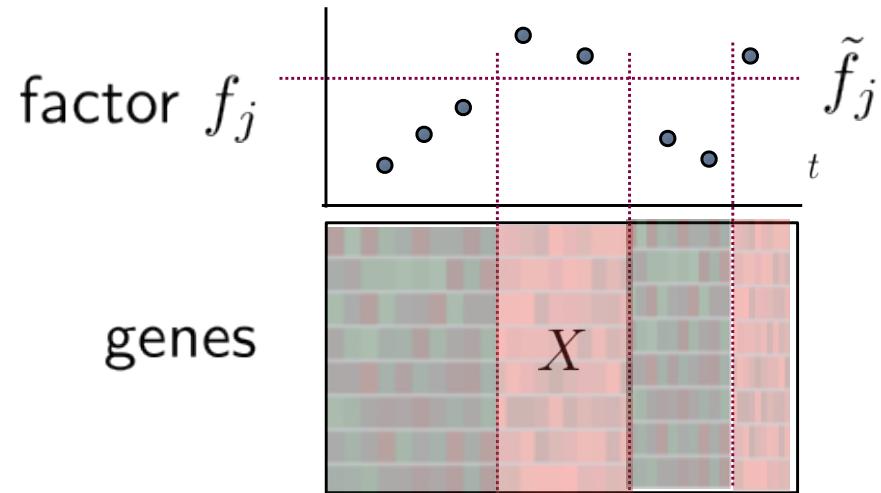
# Learning regression trees: one factor



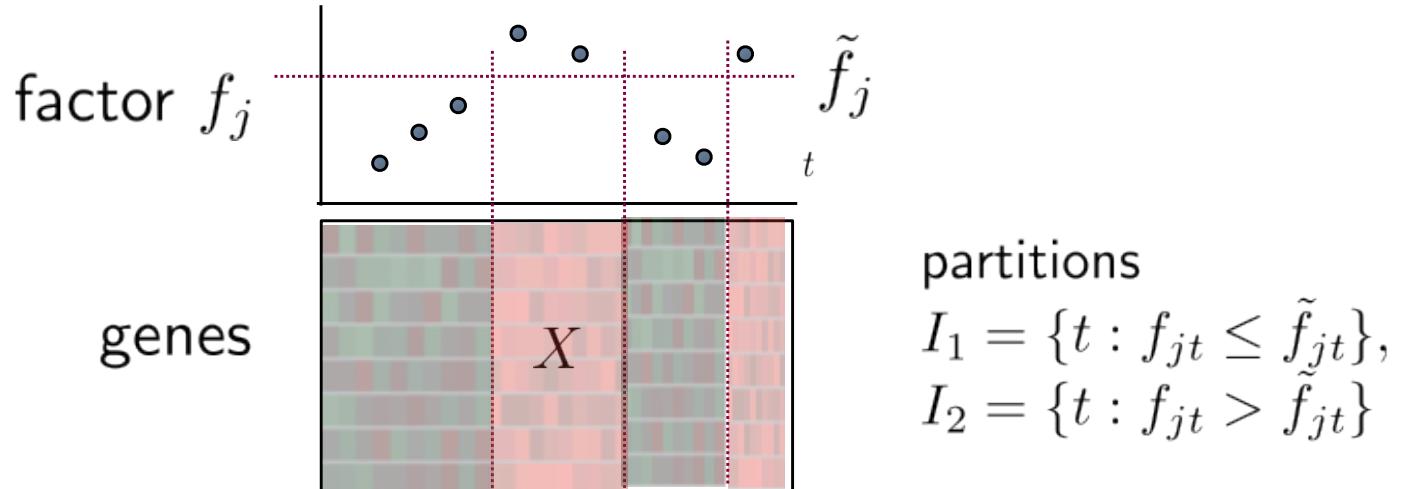
# Learning regression trees: one factor



# Learning regression trees: one factor



# Learning regression trees: one factor

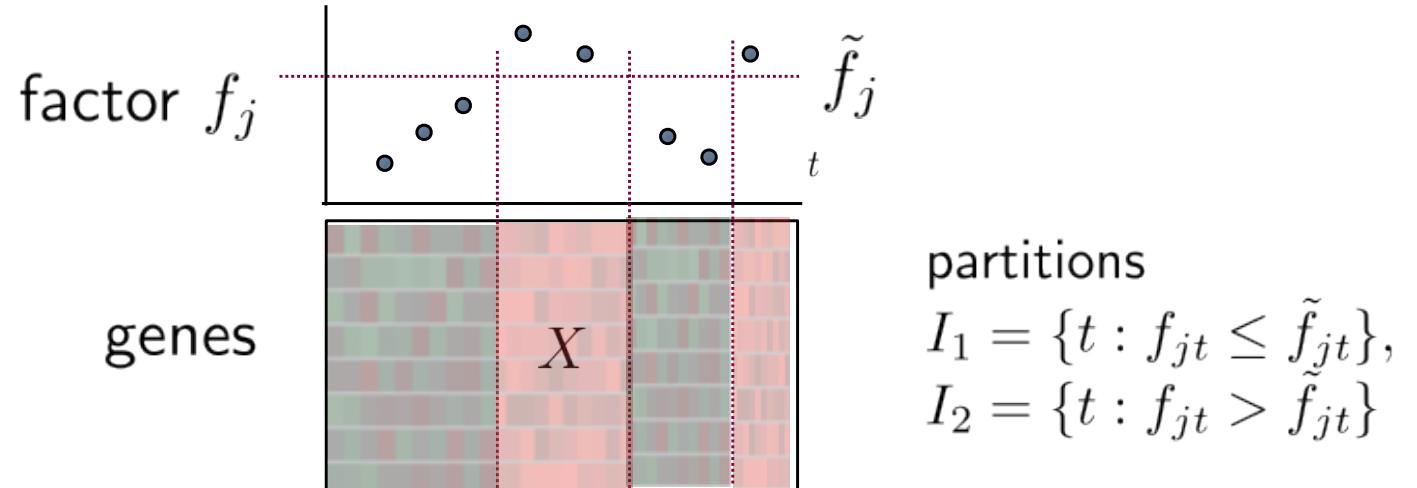


- Conditioning on the factor value partitions the expression matrix into two types of segments

$$P(X|f_j; \Theta) = \left[ \prod_{t \in I_1} \prod_{i=1}^n N(x_{it}; \mu_1, \sigma_1^2) \right] \left[ \prod_{t \in I_2} \prod_{i=1}^n N(x_{it}; \mu_2, \sigma_2^2) \right]$$

where  $\Theta = \{\tilde{f}_j, \mu_1, \mu_2, \sigma_1^2, \sigma_2^2\}$ .

# Learning regression trees: one factor



- Conditioning on the factor value partitions the expression matrix into two types of segments

$$P(X|f_j; \Theta) = \left[ \prod_{t \in I_1} \prod_{i=1}^n N(x_{it}; \mu_1, \sigma_1^2) \right] \left[ \prod_{t \in I_2} \prod_{i=1}^n N(x_{it}; \mu_2, \sigma_2^2) \right]$$

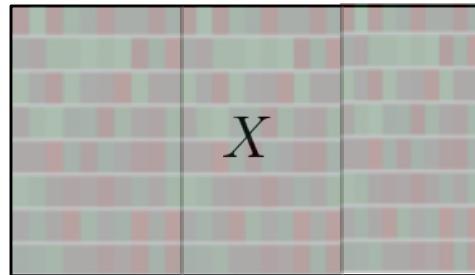
where  $\Theta = \{\tilde{f}_j, \mu_1, \mu_2, \sigma_1^2, \sigma_2^2\}$ .

- By maximizing  $l(X|f_j; \Theta) = \log P(X|f_j; \Theta)$  with respect to the parameters (threshold, means and variances), we get

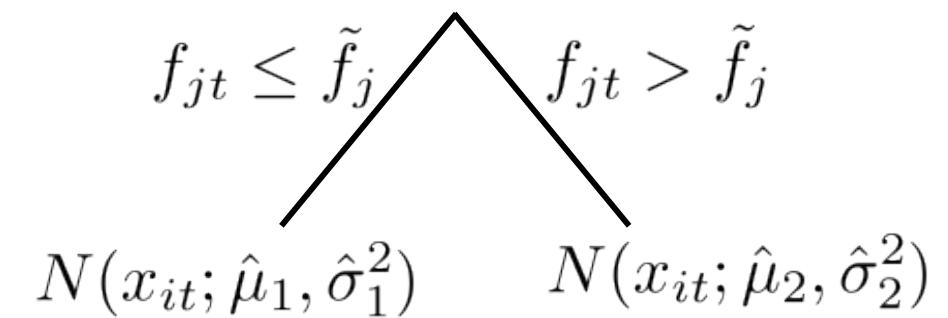
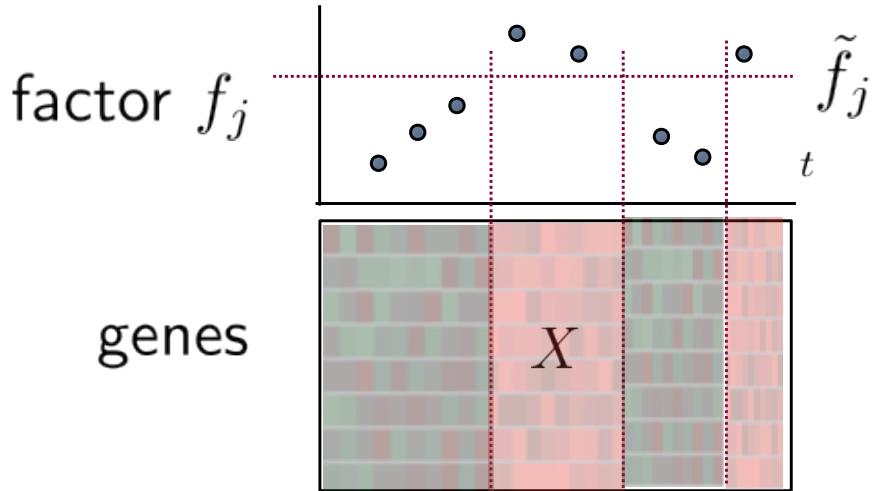
$$l(X|f_j; \hat{\Theta}) = n|\hat{I}_1| \left[ -\frac{1}{2} - \frac{1}{2} \log(2\pi\hat{\sigma}_1^2) \right] + n|\hat{I}_2| \left[ -\frac{1}{2} - \frac{1}{2} \log(2\pi\hat{\sigma}_2^2) \right]$$

# Competing “programs”

genes



$$N(x_{it}; \hat{\mu}, \hat{\sigma}^2)$$



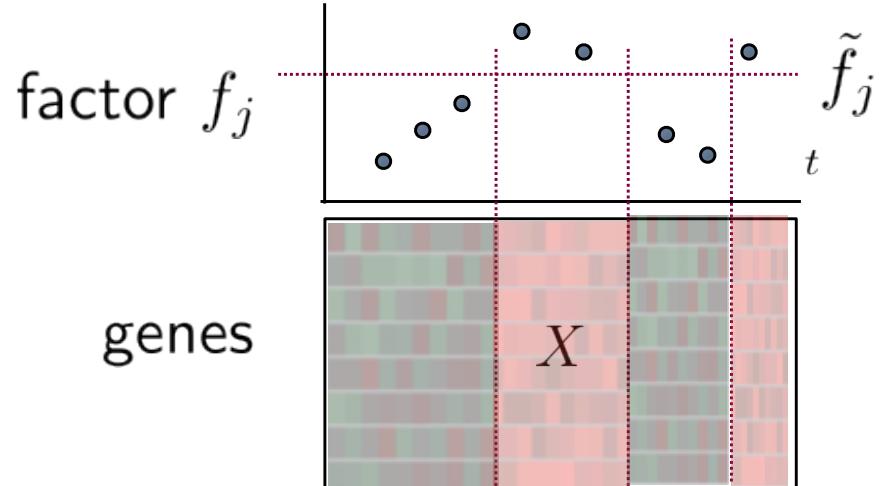
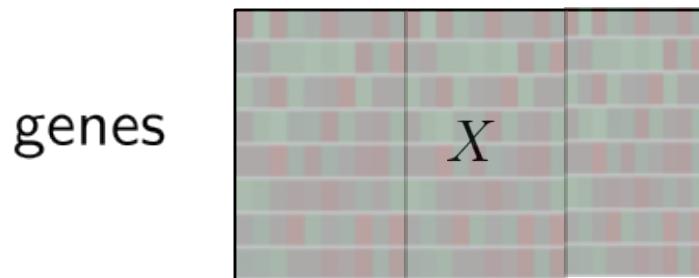
We need to penalize our likelihood models based on their complexity

Number of model parameters (2)

$$\text{BIC score} = l(X; \hat{\theta}) - \frac{2}{2} \log(nm)$$

Number of observations

# Competing “programs”



$$N(x_{it}; \hat{\mu}, \hat{\sigma}^2)$$

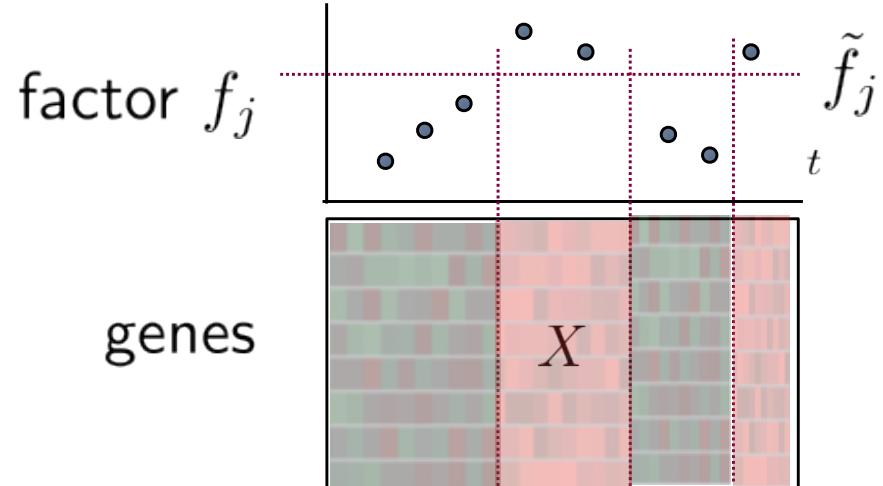
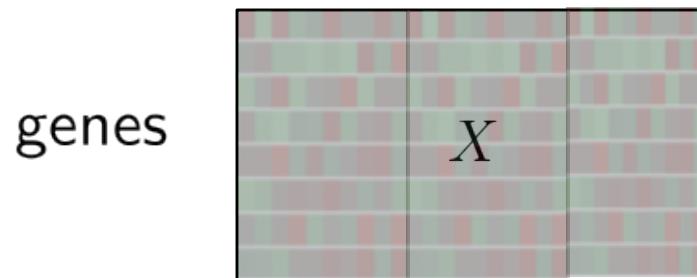
BIC score =  
$$l(X; \hat{\theta}) - \frac{1}{2} \log(nm)$$

$$f_{jt} \leq \tilde{f}_j \quad f_{jt} > \tilde{f}_j$$

$$N(x_{it}; \hat{\mu}_1, \hat{\sigma}_1^2) \quad N(x_{it}; \hat{\mu}_2, \hat{\sigma}_2^2)$$

BIC score =  
$$l(X | f_j; \hat{\Theta}) - \frac{5}{2} \log(nm)$$

# The algorithm



$$N(x_{it}; \hat{\mu}, \hat{\sigma}^2)$$

BIC score =

$$l(X; \hat{\theta}) - \frac{1}{2} \log(nm)$$

$$f_{jt} \leq \tilde{f}_j \quad f_{jt} > \tilde{f}_j$$

$$N(x_{it}; \hat{\mu}_1, \hat{\sigma}_1^2) \quad N(x_{it}; \hat{\mu}_2, \hat{\sigma}_2^2)$$

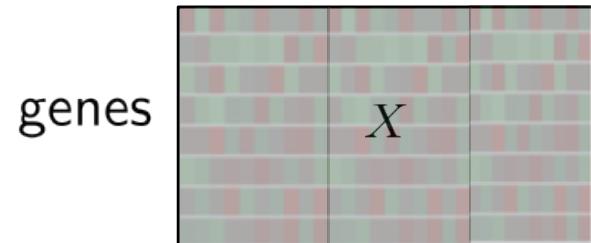
Algorithm:

- 1) Calculate BIC scores with and without factor
- 2) If null model wins, stop, otherwise recurse

BIC score =

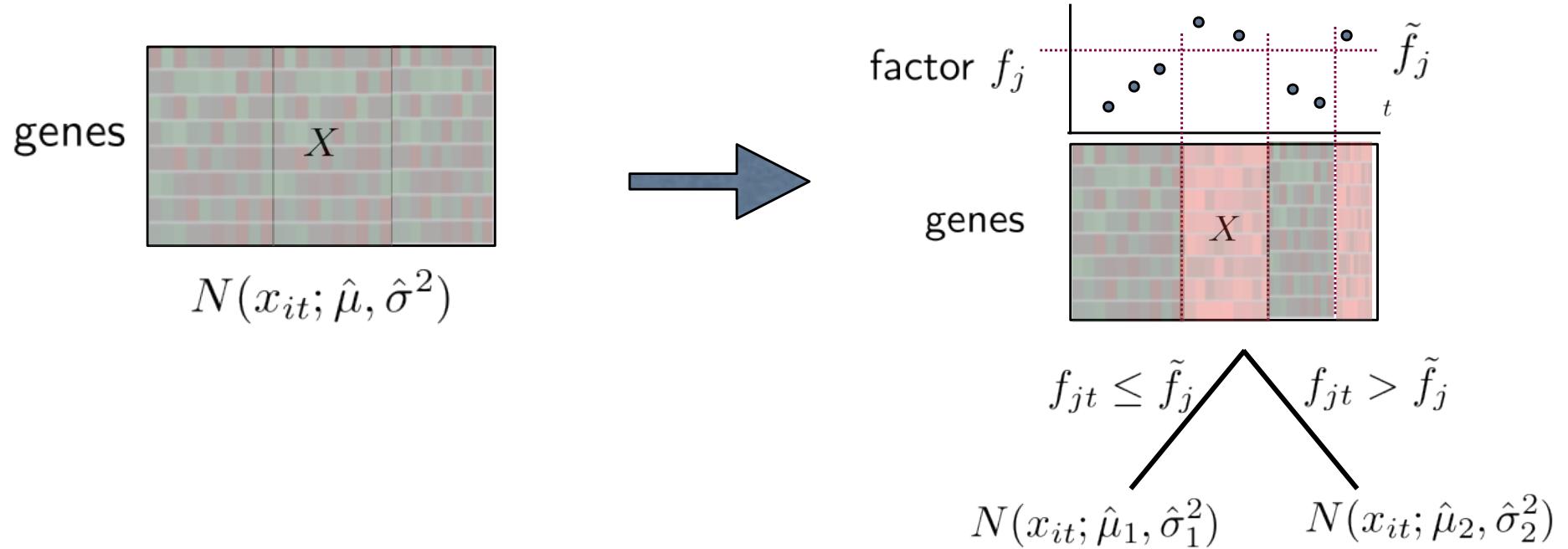
$$l(X|f_j; \hat{\Theta}) - \frac{5}{2} \log(nm)$$

# Recursive estimation

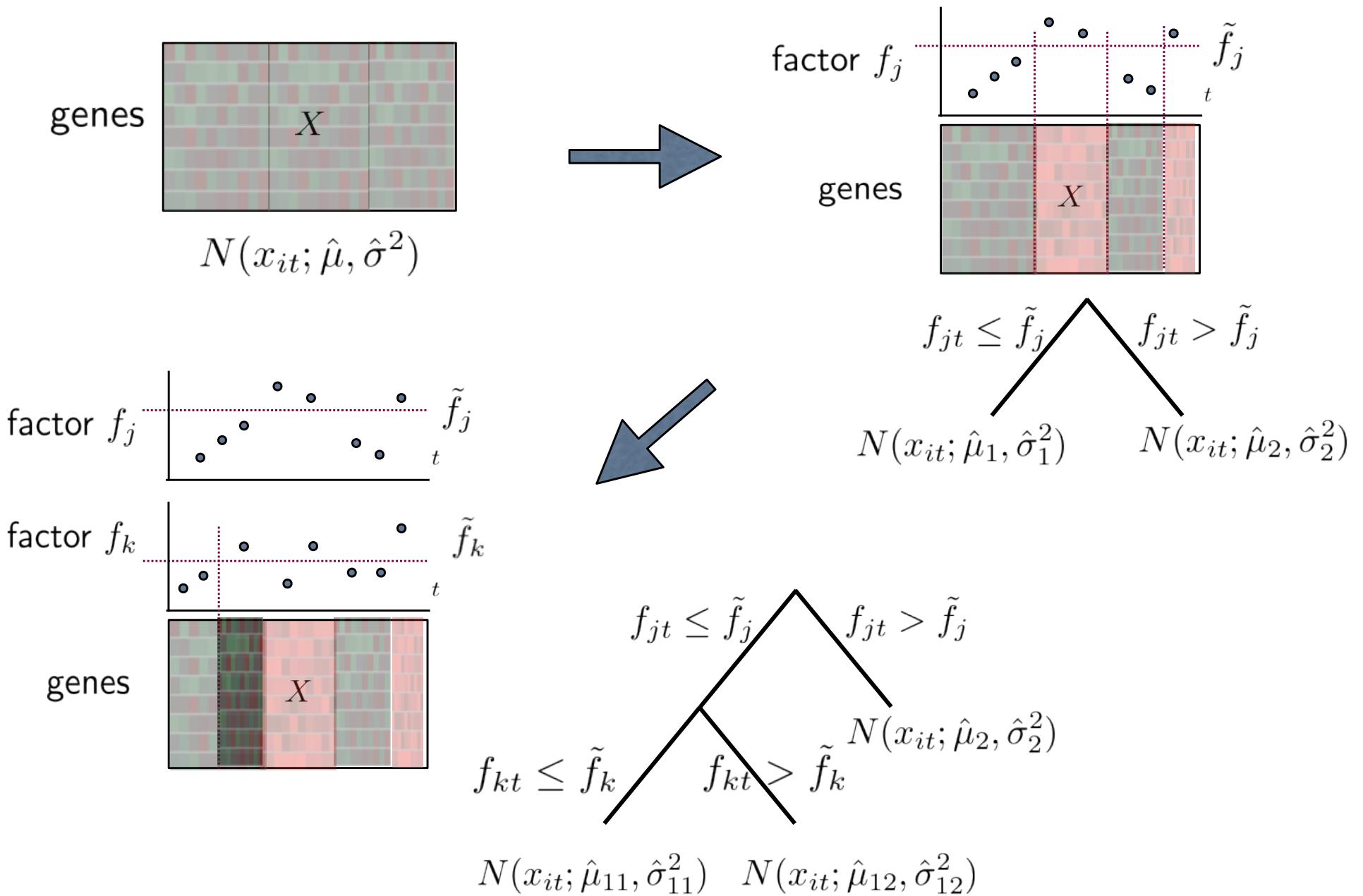


$$N(x_{it}; \hat{\mu}, \hat{\sigma}^2)$$

# Recursive estimation

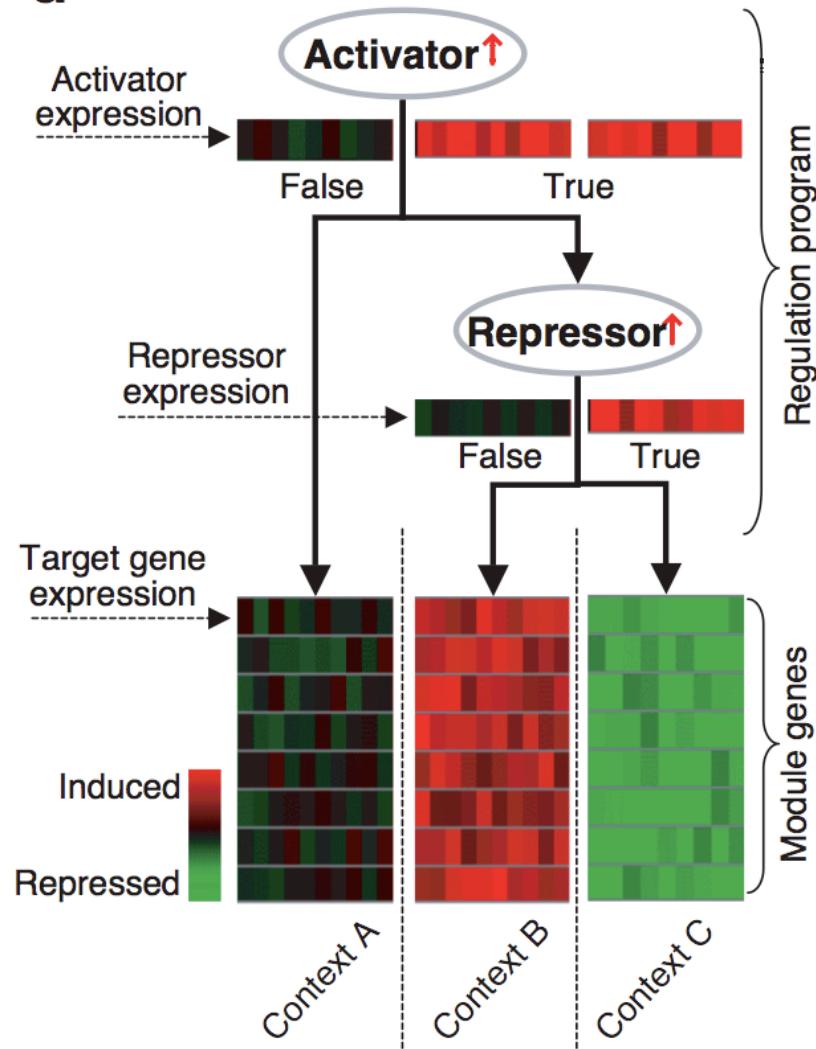


# Recursive estimation



# An ideal regression tree

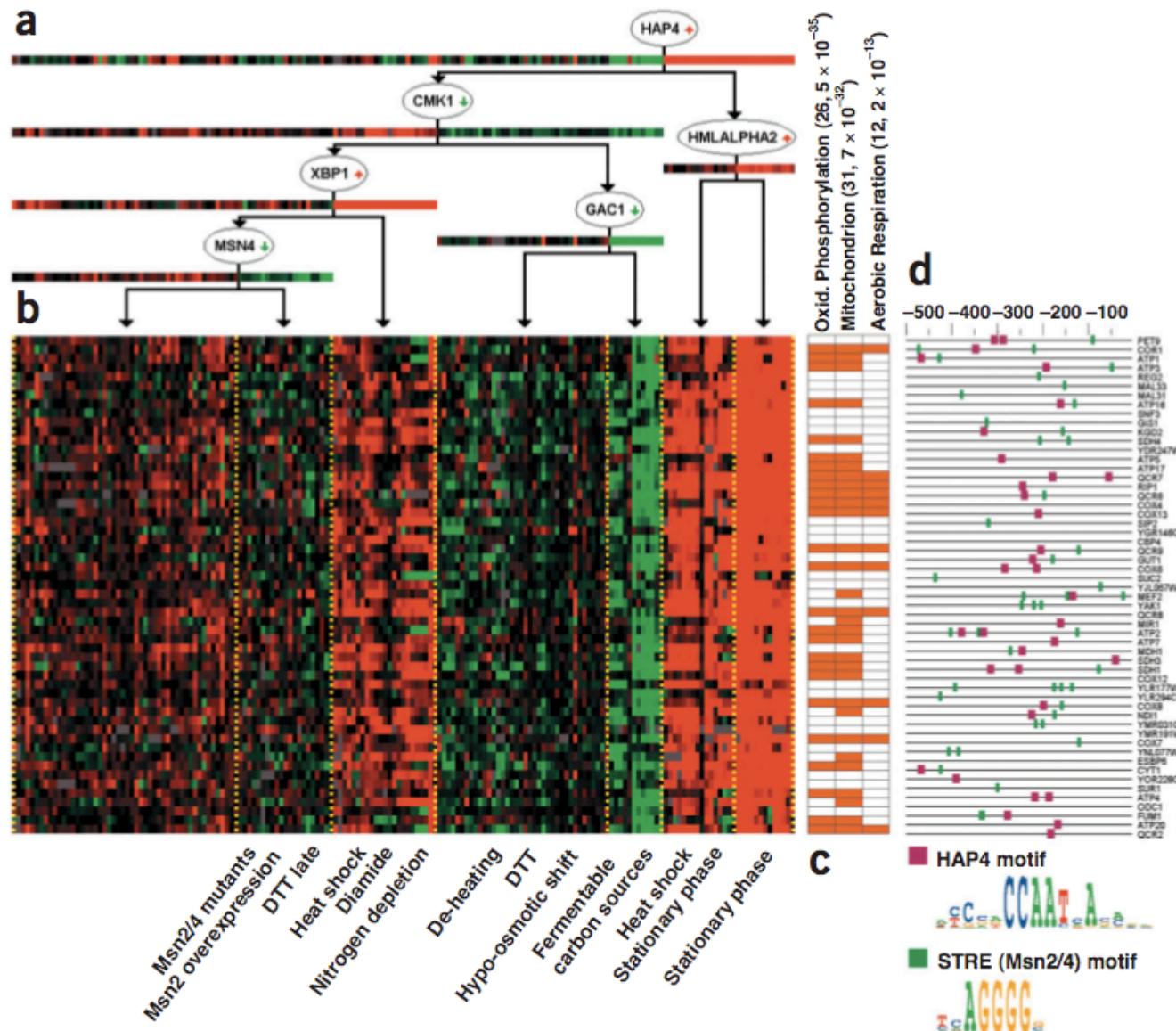
d



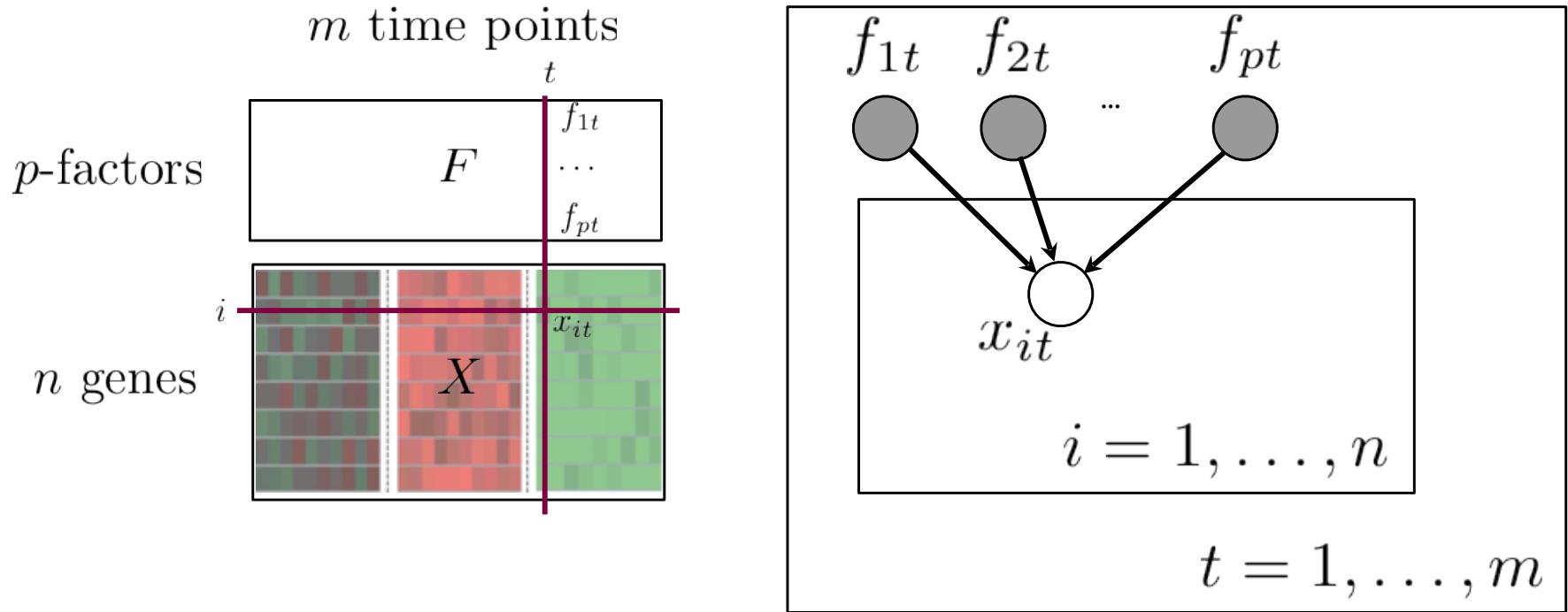
(Segal et al. 2003)

# Expression programs

- An example expression program found in yeast

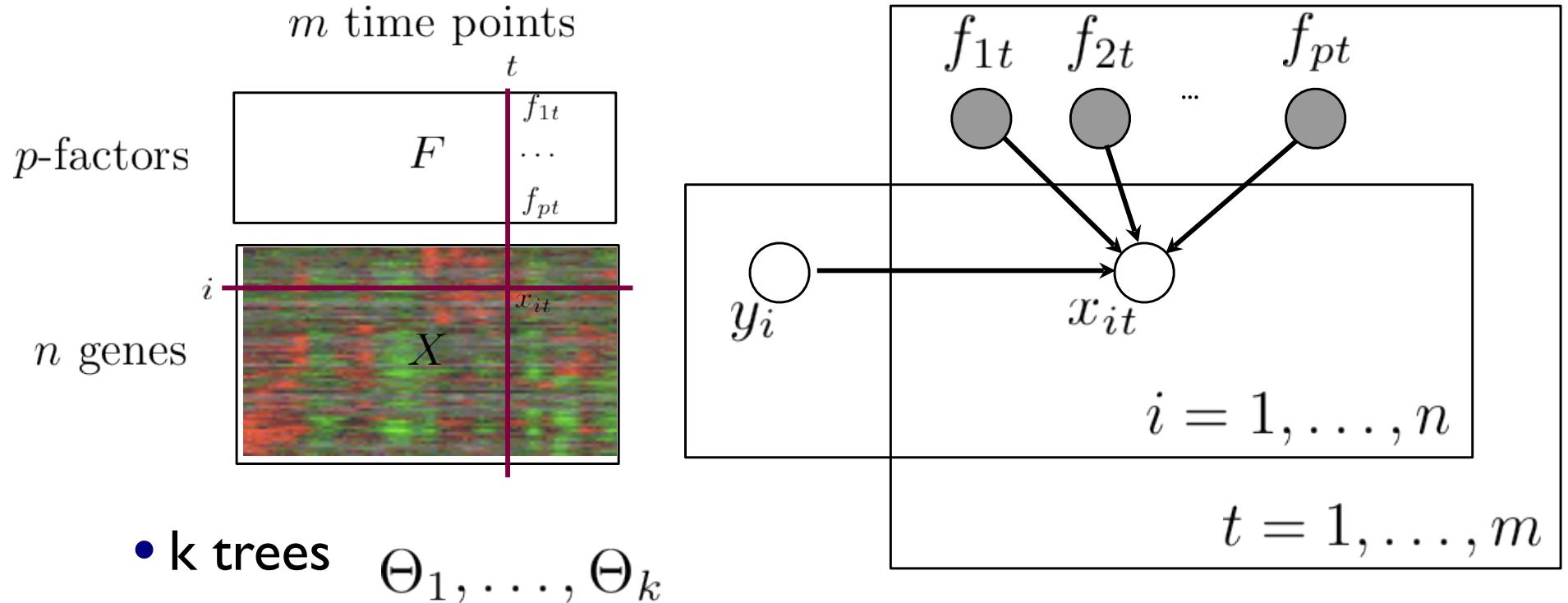


# A single regression tree: sampling model



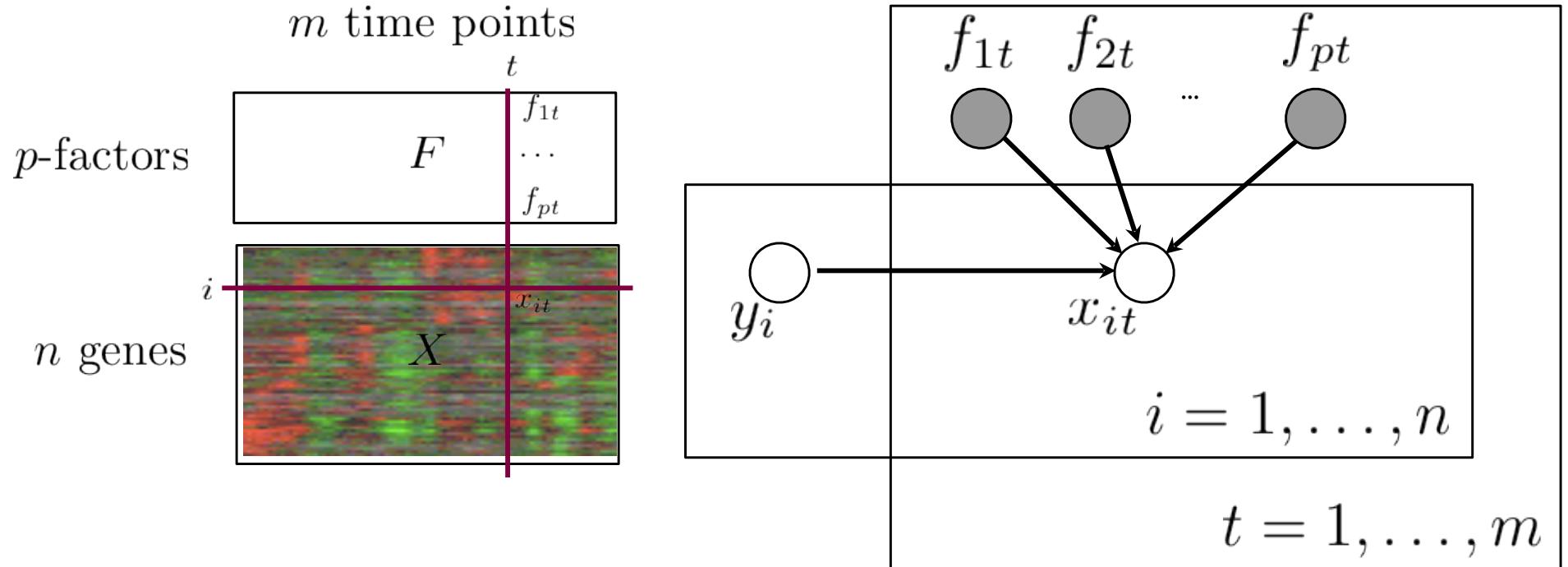
$$P(X|F, \Theta_1) = \prod_{t=1}^m \left[ \prod_{i=1}^n P(x_{it}|f_{1t}, \dots, f_{pt}, \Theta_1) \right]$$

# We can allow different regulation programs for k different “modules” of genes



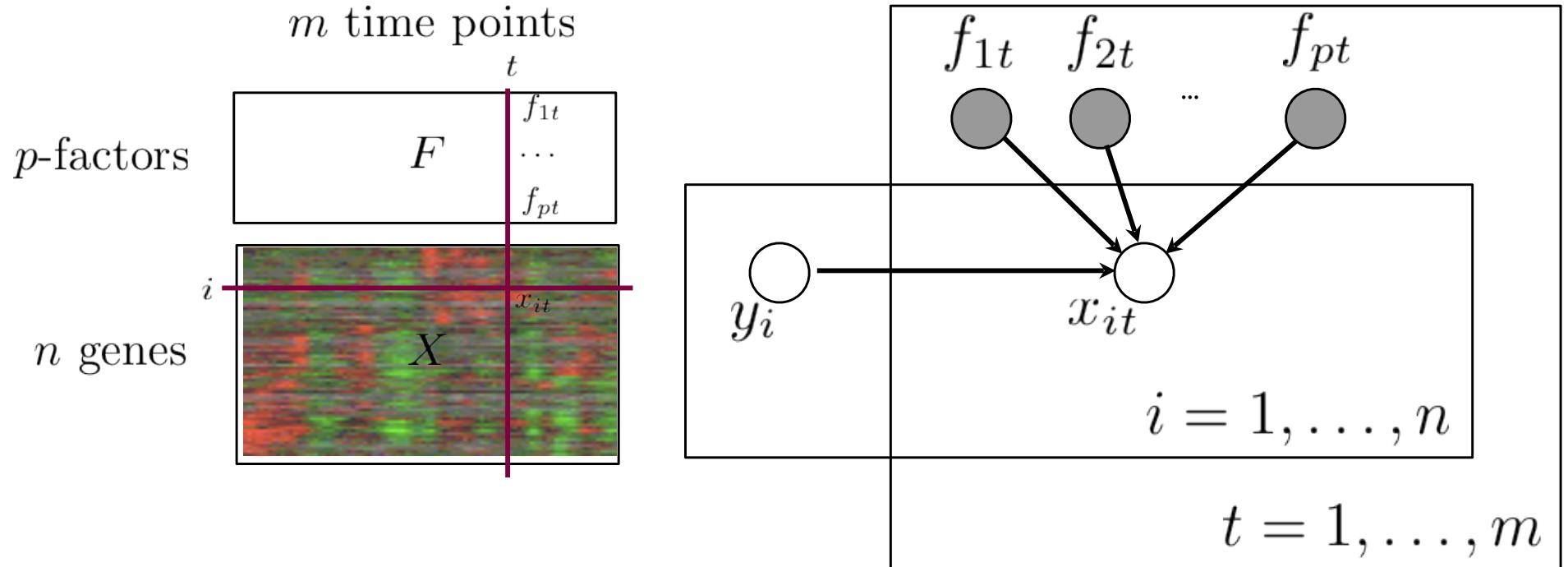
$$P(x_{it} | f_{1t}, \dots, f_{pt}, y) = P(x_{it} | f_{1t}, \dots, f_{pt}, \Theta_y)$$

# A mixture of trees: sampling model



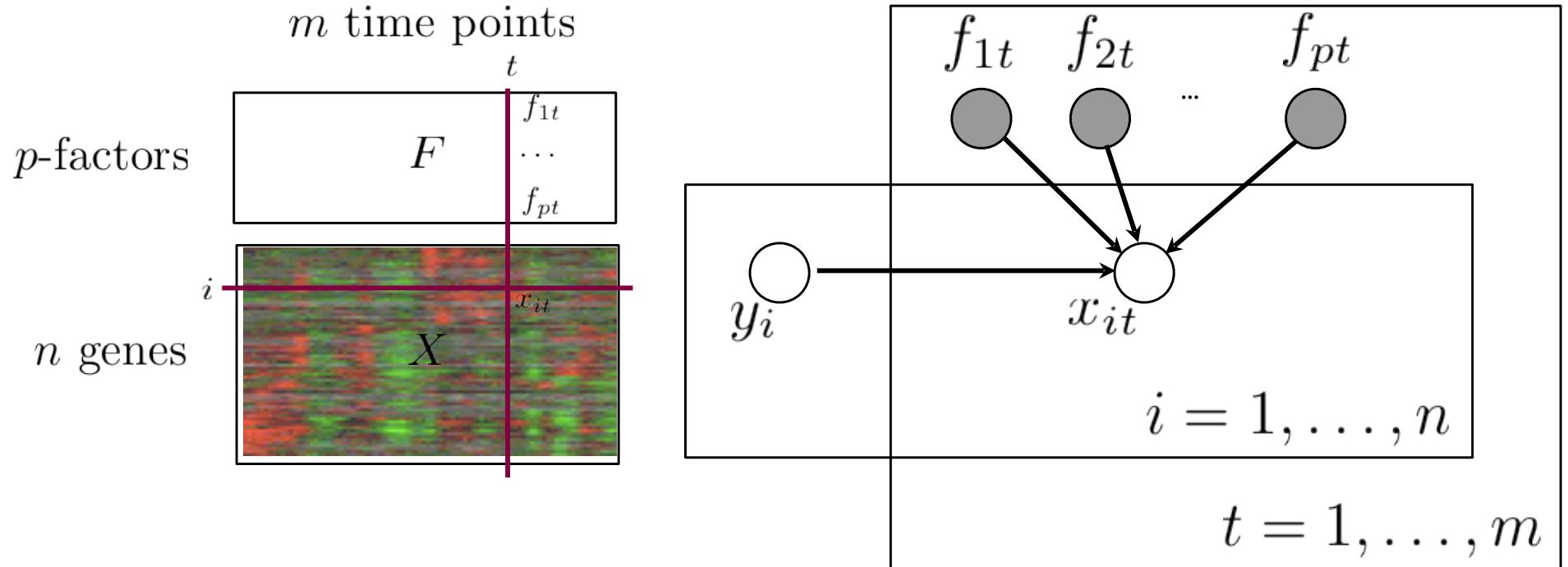
$$P(X|F, \Theta) = \prod_{i=1}^n \sum_{y_i=1}^k P(y_i) \prod_{t=1}^m P(x_{it}|f_{1t}, \dots, f_{pt}, \Theta_{y_i})$$

# A mixture of trees: sampling model



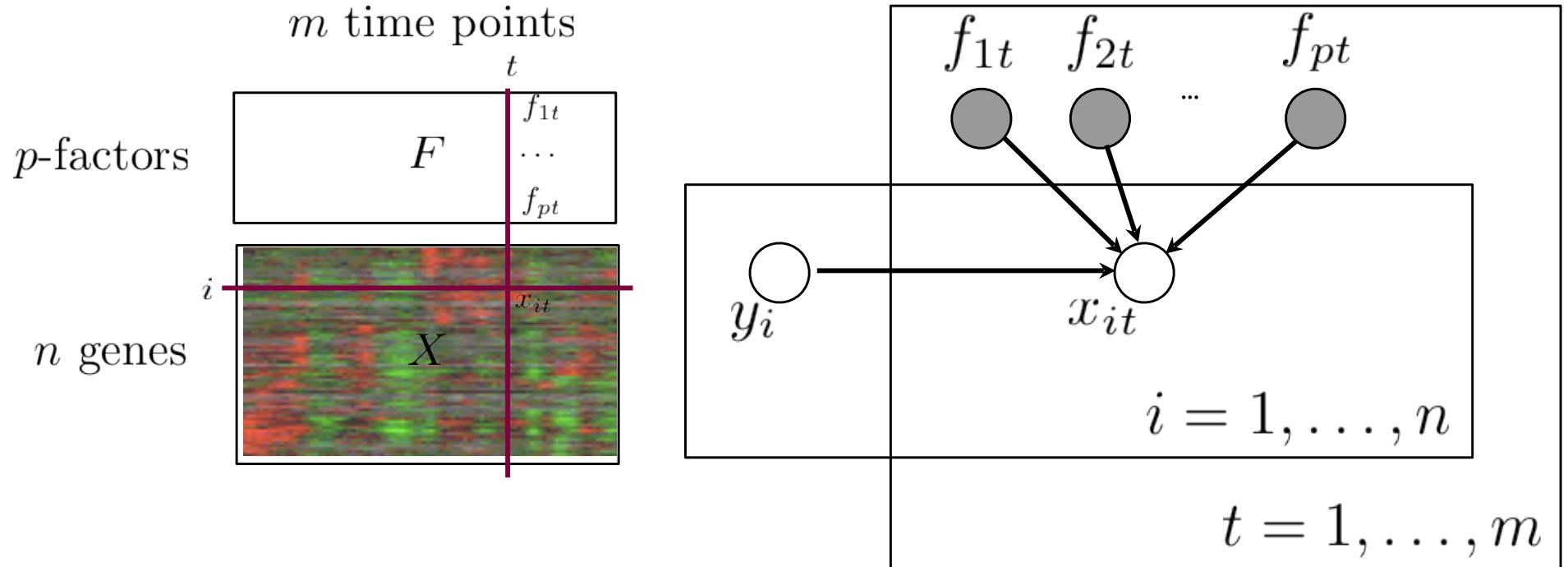
$$P(X|F, \Theta) = \prod_{i=1}^n \sum_{y_i=1}^k P(y_i) \prod_{t=1}^m P(x_{it}|f_{1t}, \dots, f_{pt}, \Theta_{y_i})$$

# A mixture of trees: sampling model



$$P(X|F, \Theta) = \prod_{i=1}^n \sum_{y_i=1}^k P(y_i) \prod_{t=1}^m P(x_{it}|f_{1t}, \dots, f_{pt}, \Theta_{y_i})$$

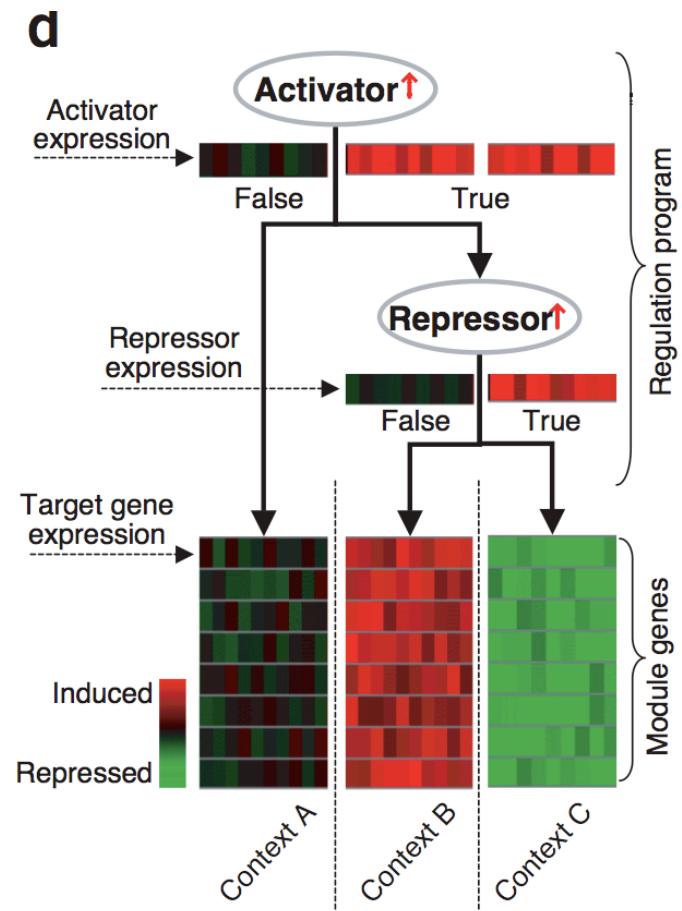
# A mixture of trees: sampling model



$$P(X|F, \Theta) = \prod_{i=1}^n \sum_{y_i=1}^k P(y_i) P(\underline{x}_i|F, \Theta_{y_i})$$

# Finding expression programs

- We can jointly search for transcriptionally regulated subsets of genes and the corresponding programs (Segal et al., 2003)



# EM algorithm

- We model the program assignments and expression profiles as

$$P(y_1, \dots, y_n, \underline{x}_1, \dots, \underline{x}_n | F, \Theta) = \\ \prod_{i=1}^n P(y_i) P(\underline{x}_i | F, \Theta_{y_i})$$

- We can optimize the parameters of this model via the EM algorithm

- (0) cluster genes into  $k$  clusters
- (1) re-estimate regression trees for each cluster (program)
- (2) (softly) re-assign each gene to the best program

$$p(y = j | i) \propto P(y) P(\underline{x}_i | F, \Theta_y)$$

# Regression trees as programs

- Each regression tree can be represented as an “if-then-else” program

Cond. distribution  $P(x|f_1, \dots, f_p) = P(x|F)$

if  $f_j \leq \tilde{f}_j$

$x \sim N(x; \mu_1, \sigma_1^2)$

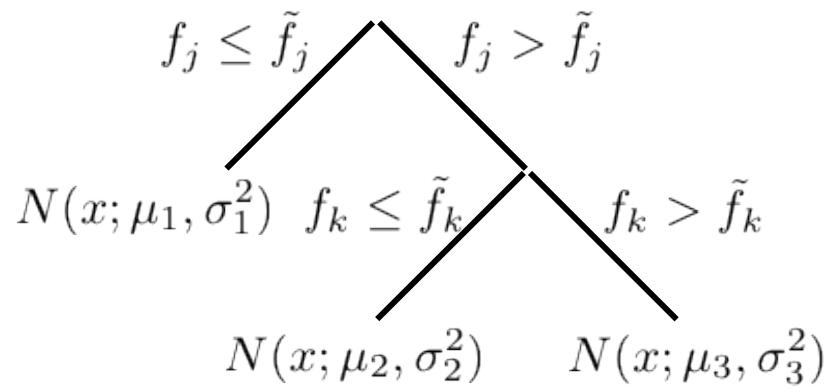
else

if  $f_k \leq \tilde{f}_k$

$x \sim N(x; \mu_2, \sigma_2^2)$

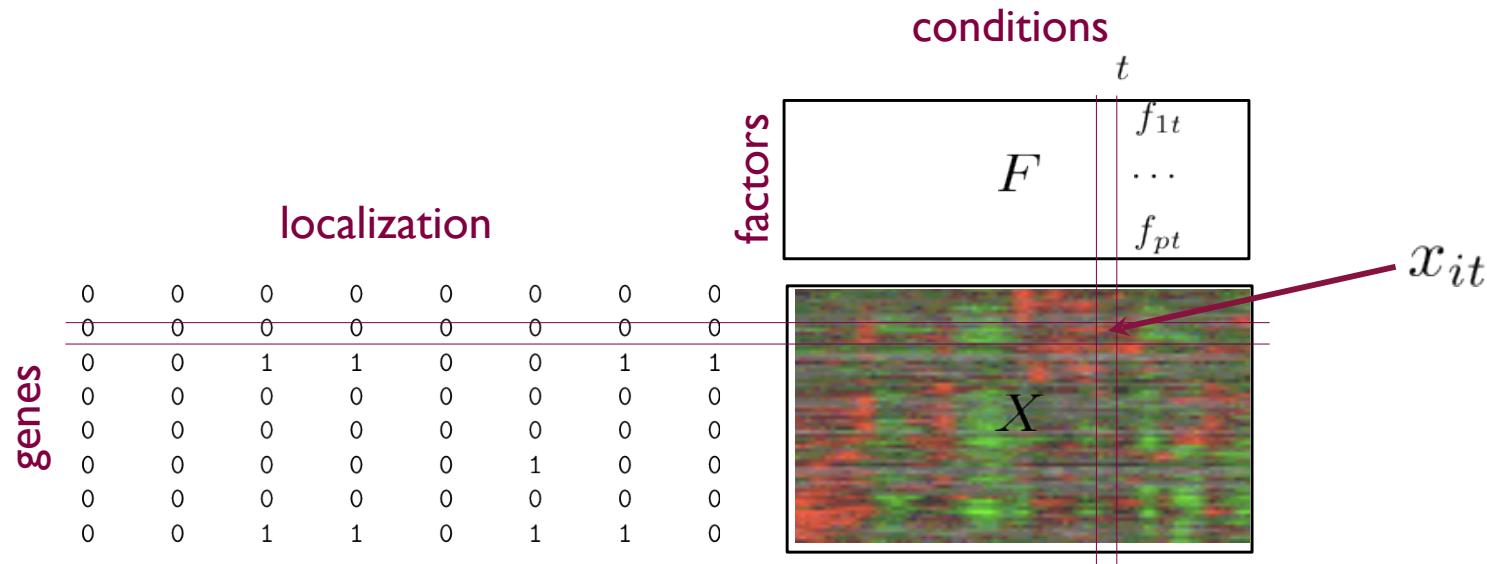
else

$x \sim N(x; \mu_3, \sigma_3^2)$



# Extensions...

- The basic regression trees formulation permits us to estimate various types of “programs”
  - e,g,TF expression & localization, etc.



**FIN - Thank You**

# Part 1 - Model Capacity

# Sufficient training data are necessary to generalize well

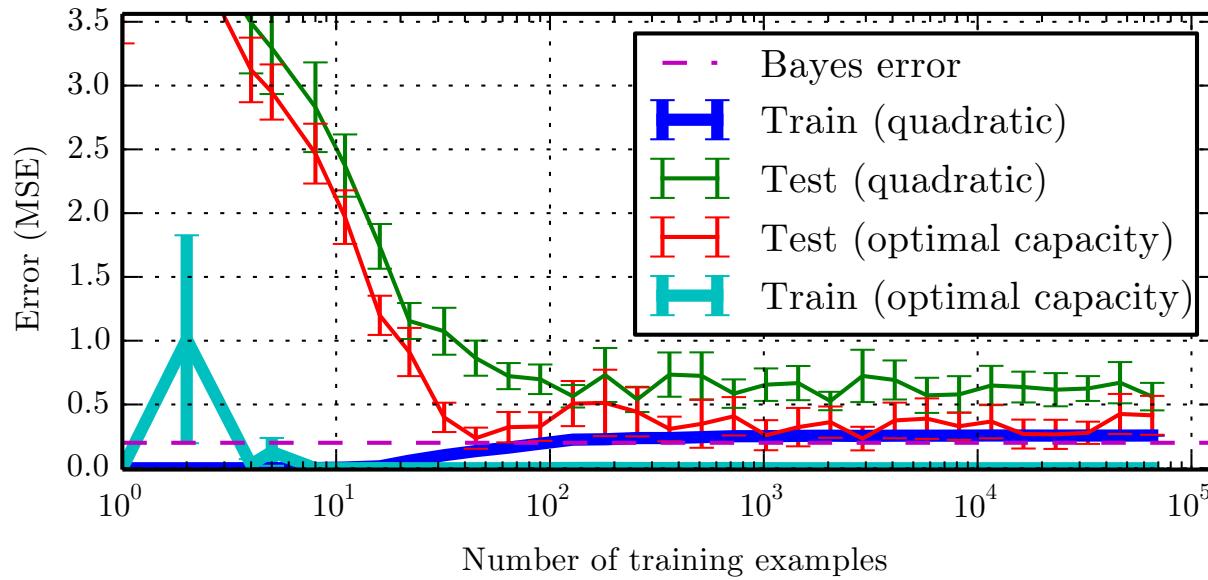
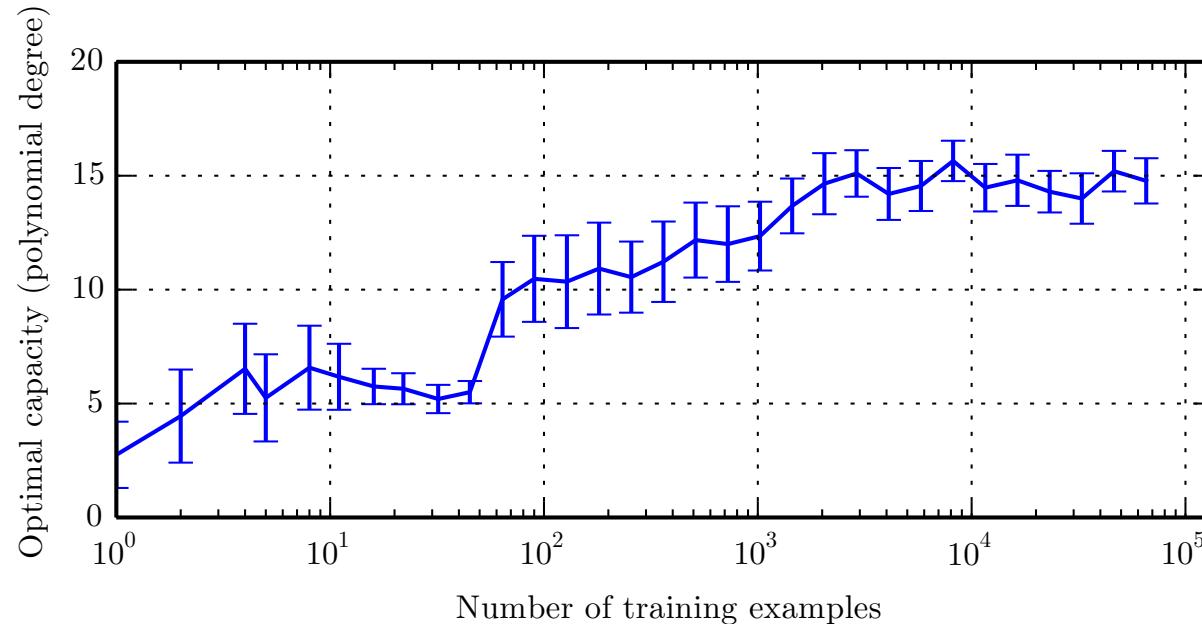


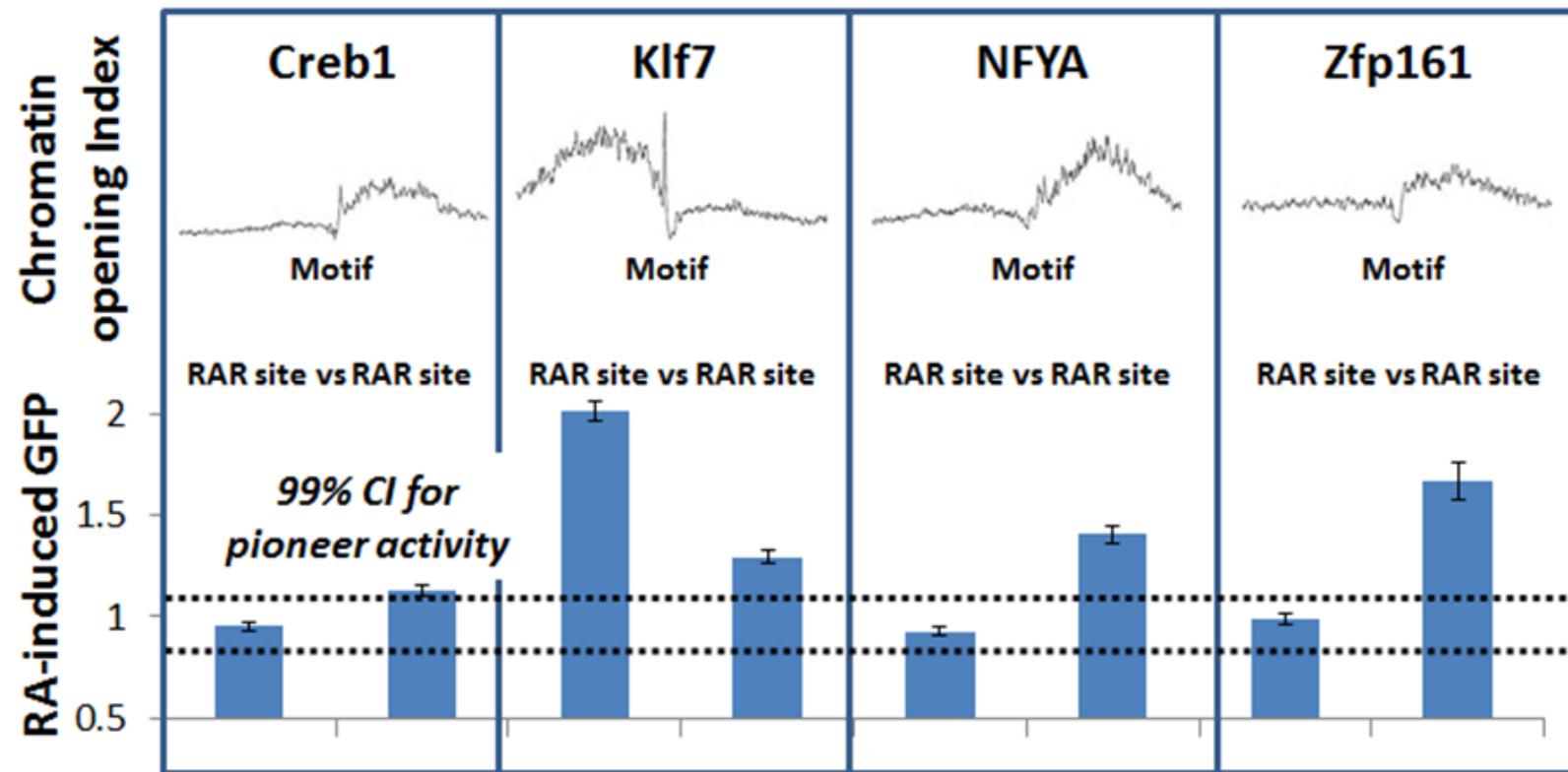
Figure 5.4



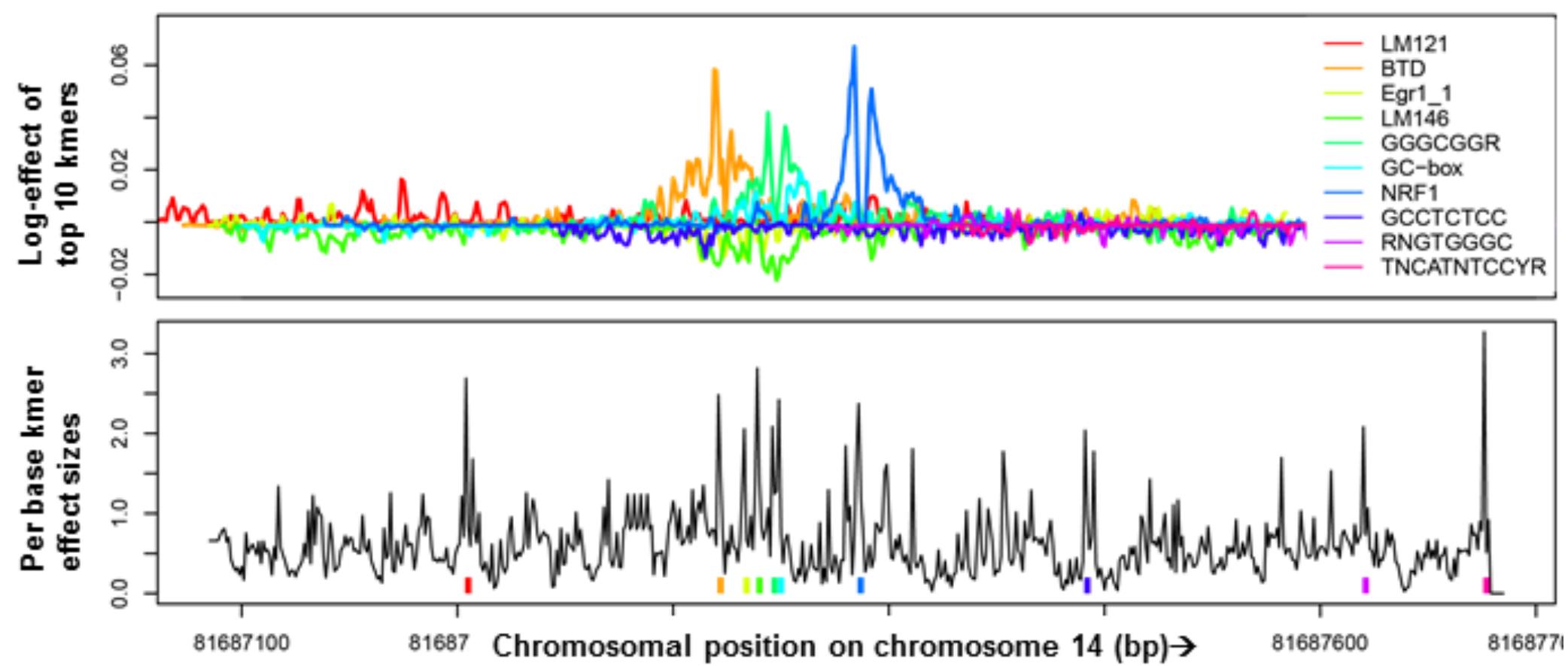
# Part 2 - Modeling Genome Accessibility

# Certain pioneer TFs are directional

Orienting the motif direction in the reporter recapitulates expected directional behaviors.



# The Synergistic Chromatin Model (SCM) is a K-mer model



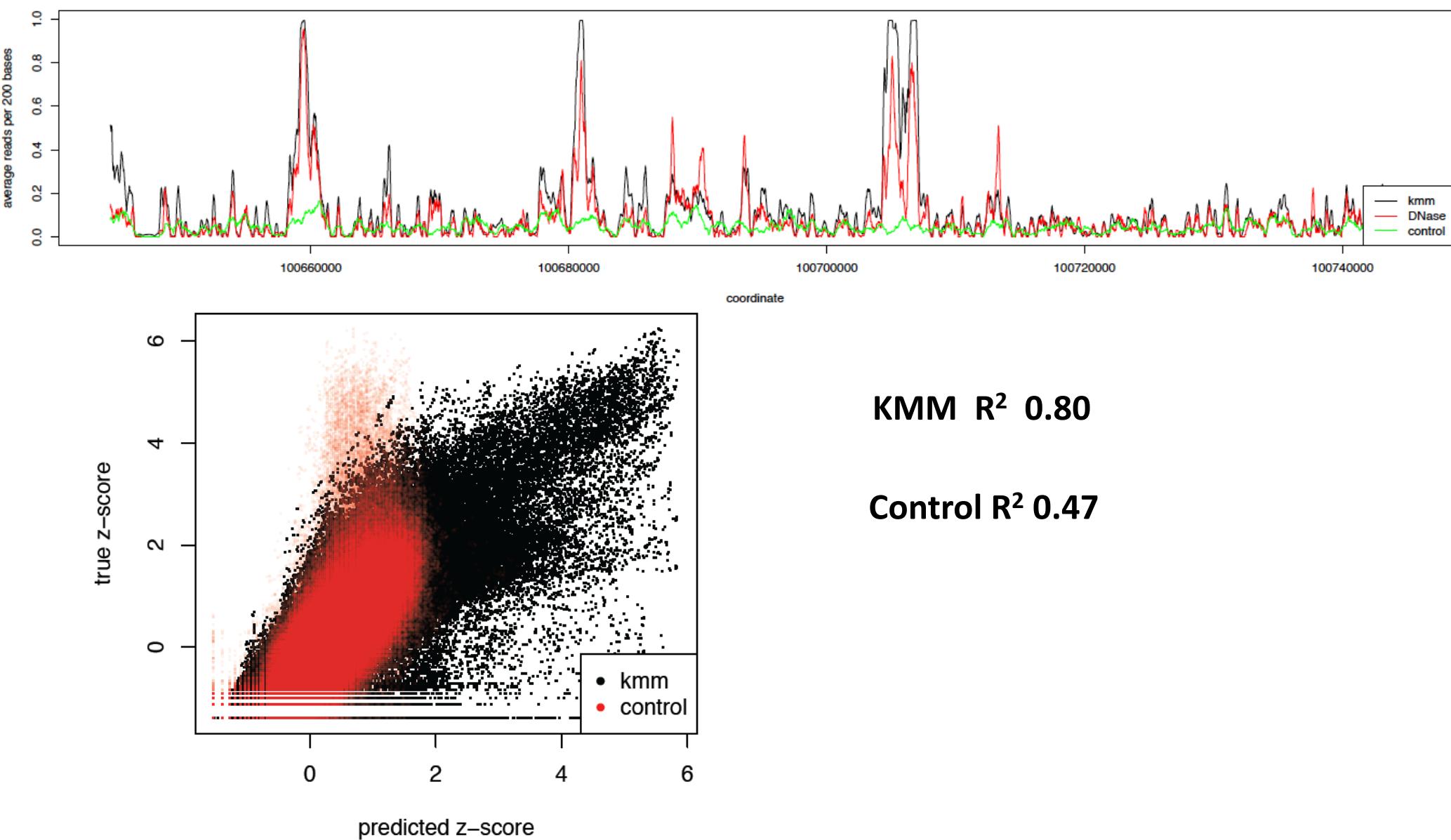
~40,000 K-mers in model

~5,000,000 parameters

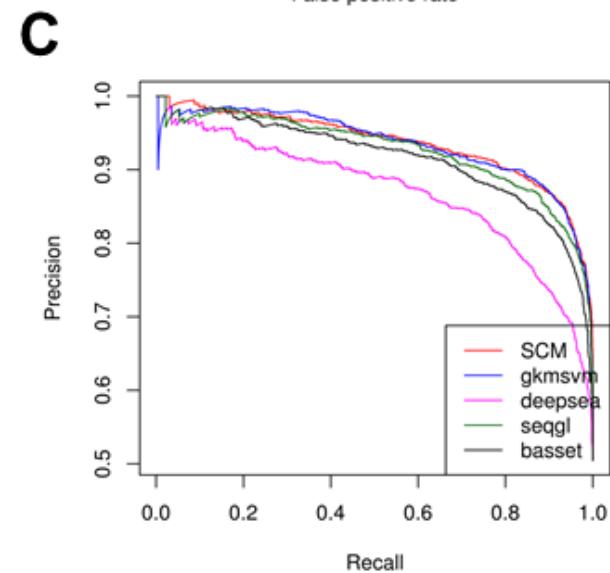
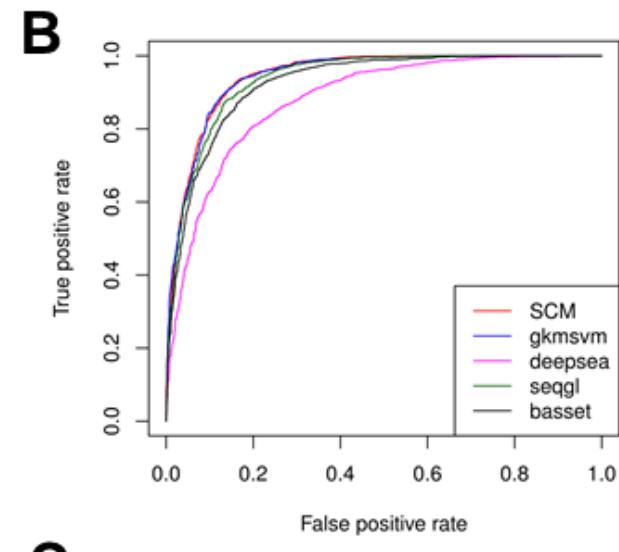
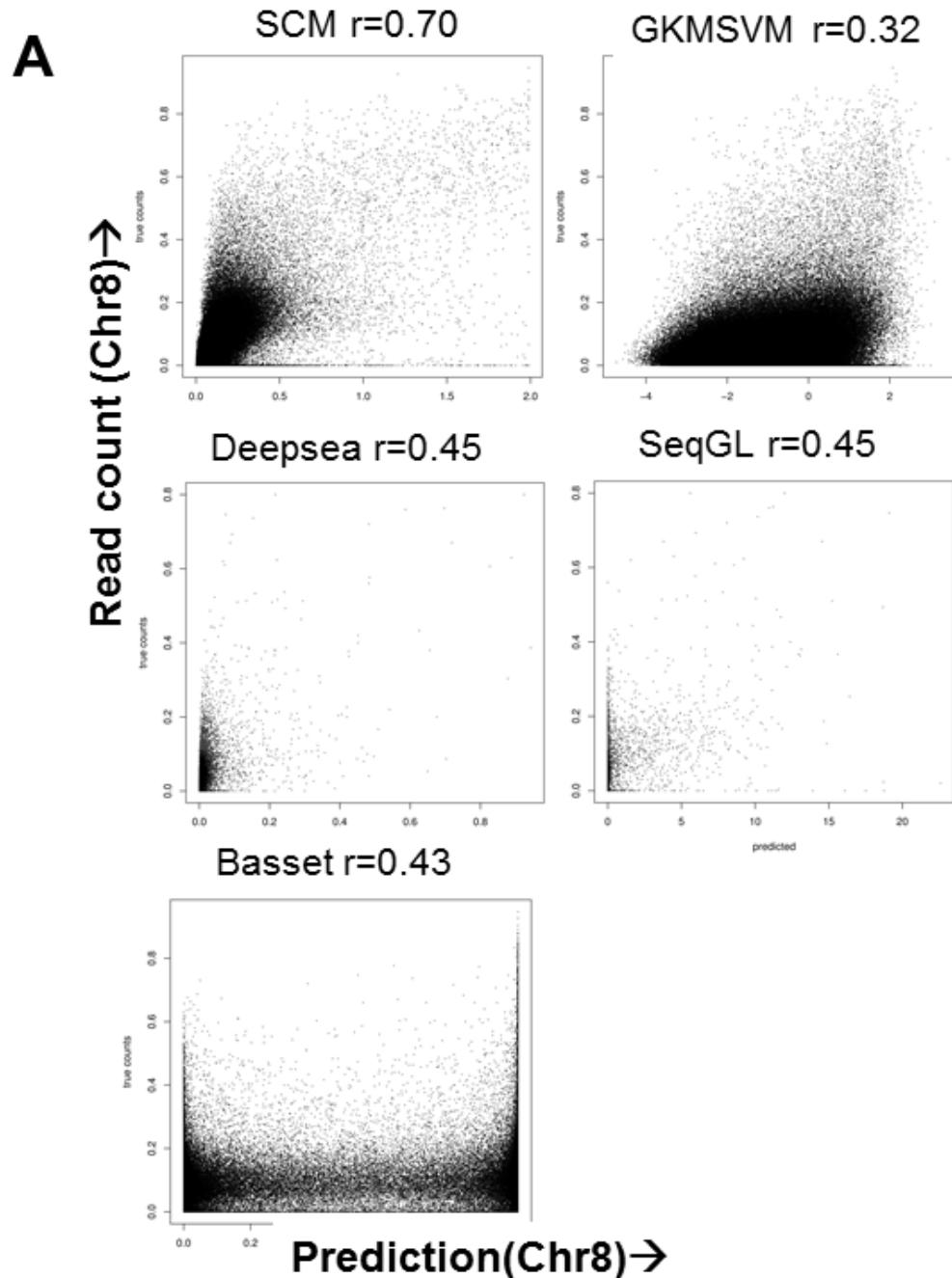
543 iterations \* 360 seconds / iteration \* 40 cores

= ~ 90 days

Training on K562 DNase-seq data from chromosomes 1 – 13  
predicts chromosome 14 (black line)



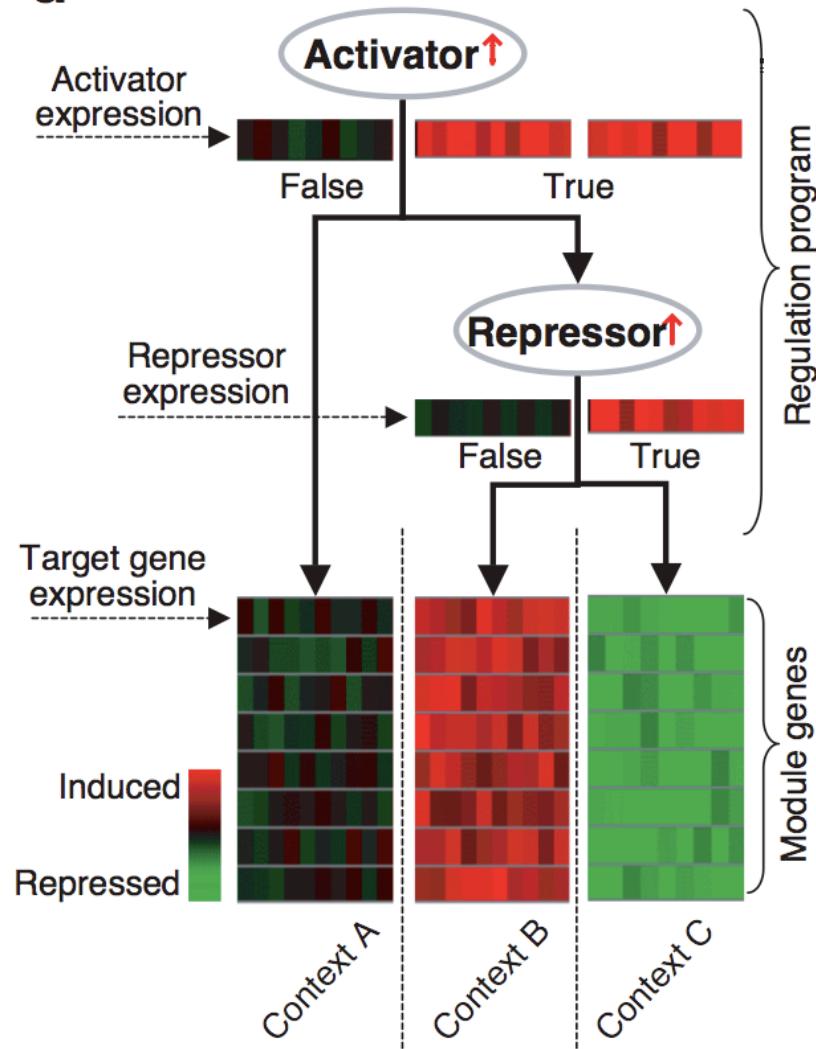
# SCM outperforms contemporary models at predicting chromatin accessibility from sequence (K562)



# Part 3 - Modeling Gene Expression

# An ideal regression tree

d



(Segal et al. 2003)

# Expression programs

- An example expression program found in yeast

