3. Predicting binding sites

- basics of TFBS identification
- defining a background model
- tools
- phylogenetic footprinting
- **including "in-vivo features"**

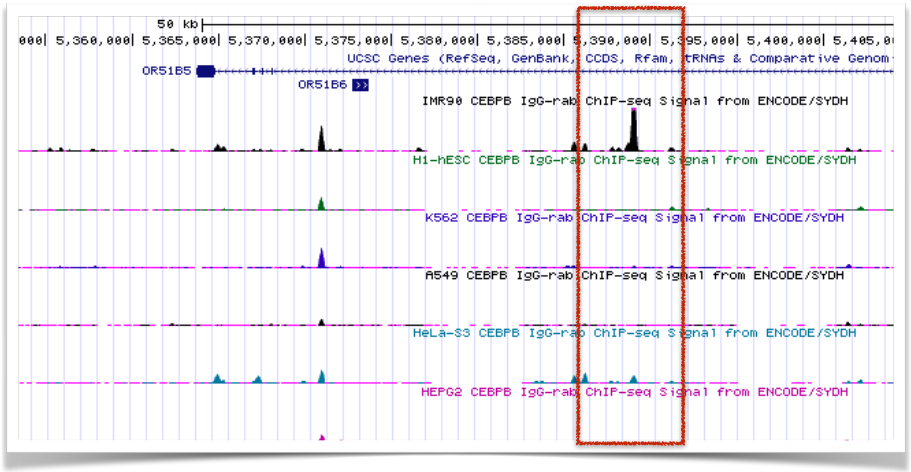
IPMB
Institut für Pharmazie und Molekulare Biotechnologie

UNIVERSITÄT HEIDELBERG
ZUKUNFT SEIT 1386

Improving TFBS predictions

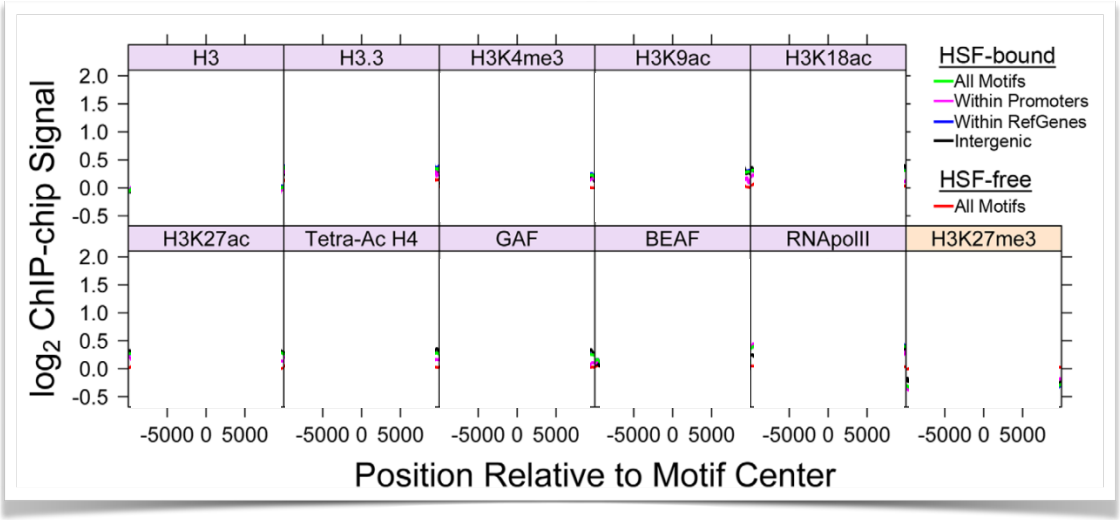
- **Limitations**
  - quality of the matrix (PWMs constructed from few sites are not discriminative, low information content !)
  - difficulty to predict **low affinity** binding events
  - correct choice of the **background** model
  - **in-vivo context** is not taken into account

Why is CEBPB binding in some cell-lines and not in others ?



Motifs are not always binding events

- Compare **in-silico** TFBS to **in-vivo** binding event using ChIP data
- some bona fide motifs are not bound in-vivo : why ?
- example in Drosophila : heat-shock factor (HSF)
  - 464 ChIP peaks containing a HSF-motif ($p < 0.001$)
  - 708 unbound motifs (with $p < 5e\text{-}6$)



Bound sites have:

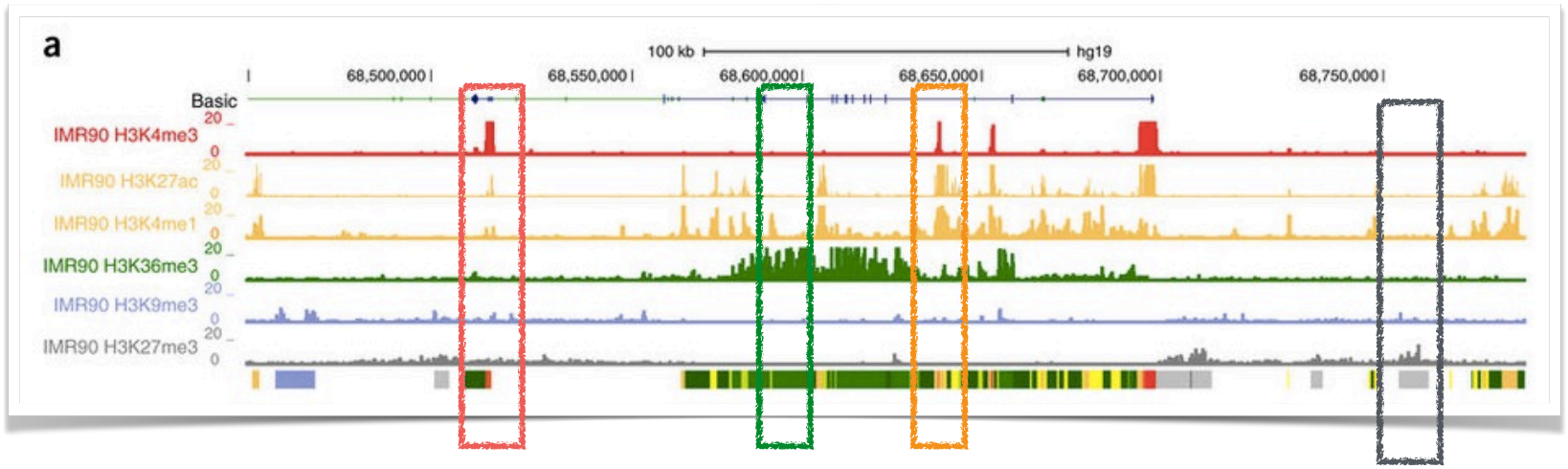
- **high levels of lysine acetylation**
- **high levels of Pol2 binding**
- **low levels of H3K27me3 (repressive mark related to polycomb repression)**

[Guertin et al., PLoS Gen. (2010)]

Histone code

Mark	Interpretation
H3K4me1	activating mark; found at promoters and enhancers
H3K4me3	mark of active and open gene promoters
H3K27ac	mark of active promoters and enhancers
H3K36me3	mark of actively transcribed genes
H3K9me3	mark of closed heterochromatin
H3K27me3	polycomb associated mark → repressed chromatin

Histone code



active marks
mostly H3K4me3

mostly
H3K36me3

strong activating marks
mostly H3K27ac

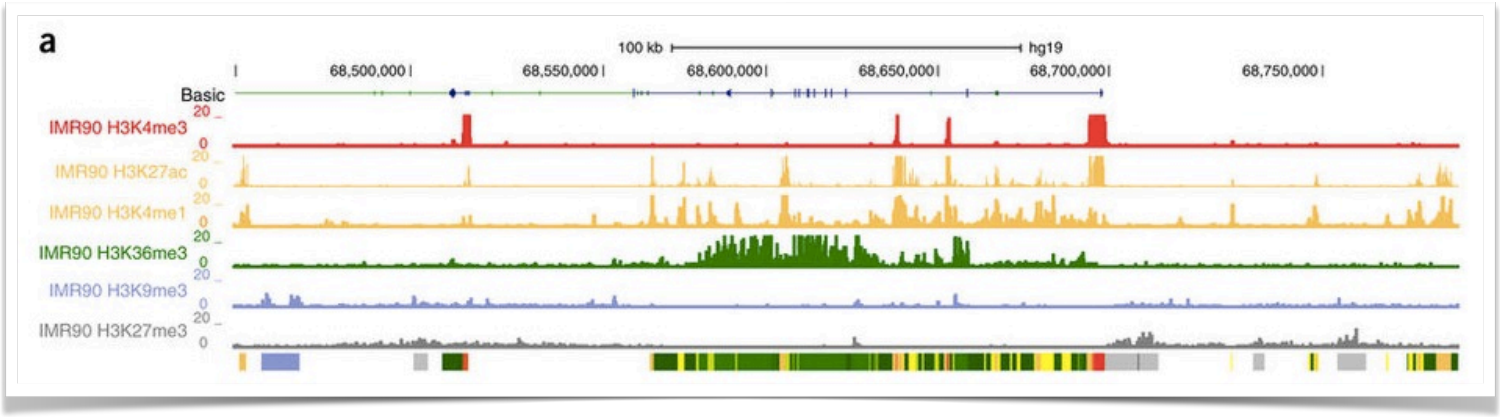
repressive marks
H3K27me3

Histone modifications appear to occur in specific combinations related to functional impact → **combinatorial chromatin states**

How can we define/annotate these **chromatin states** ?
→ **Hidden Markov model**

Hidden Markov Model of chromatin states

- There are ***n*** **(unobserved) chromatin states** (active promoter, enhancer, repressed chromatin,...)
  → **HIDDEN STATES**
- These states **emit specific chromatin features** (histone modifications, DNA methylation, accessibility)
  → **OBSERVED DATA**



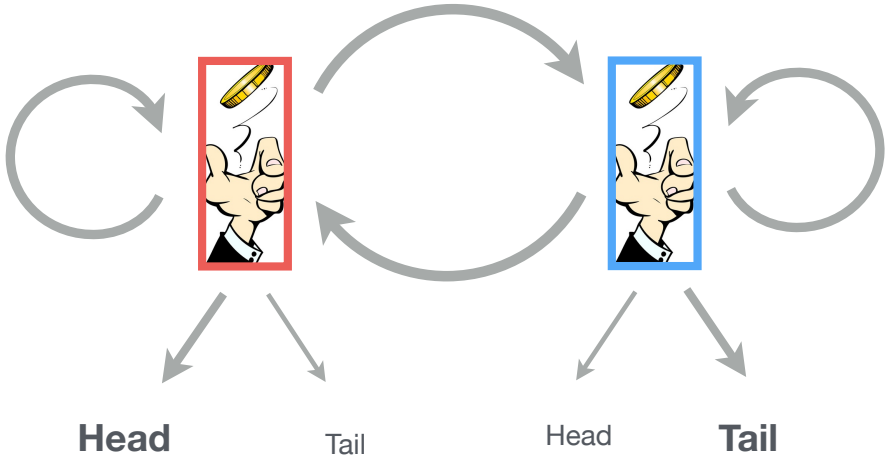
How can we resconstruct the chromatin states from the observed data?

Hidden Markov Models

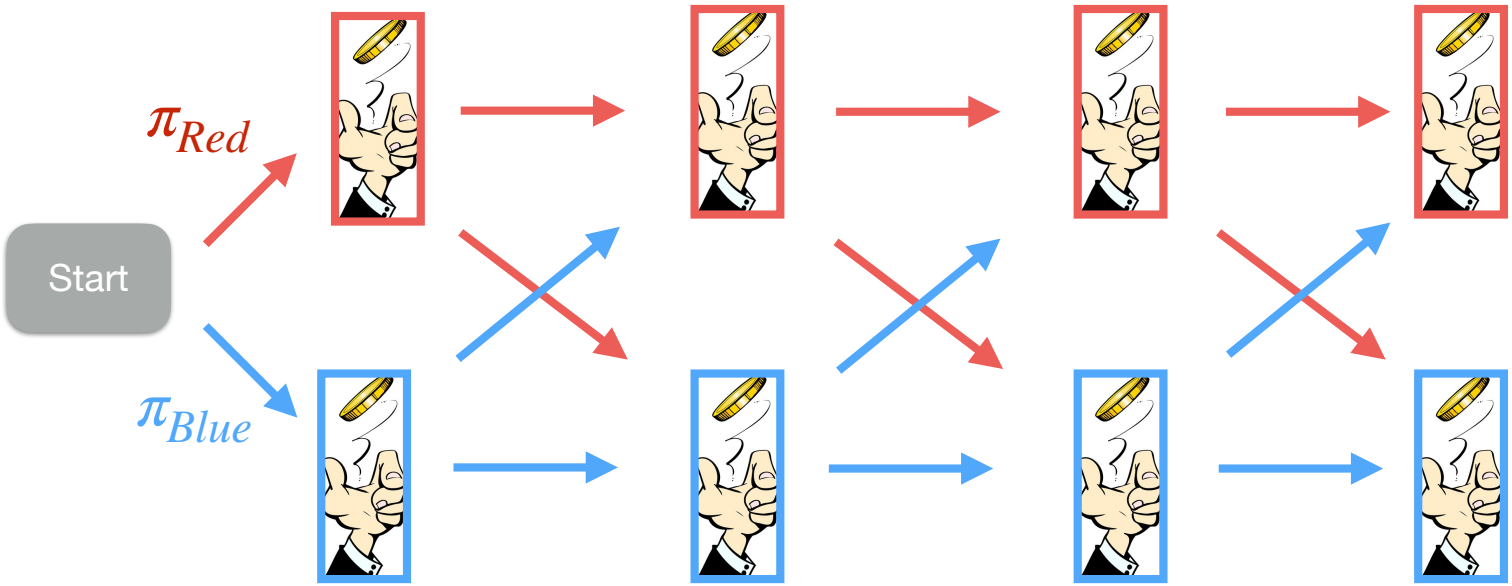
- In Hidden Markov Models, we observe a **sequence of observed events *x***, generated by **hidden underlying states *z***

$$\begin{array}{cccc} x_1 & x_2 & \dots & x_n \\ \uparrow & \uparrow & \uparrow & \uparrow \\ z_1 & z_2 & \dots & z_n \end{array}$$

- Suppose we have 2 different coins, which are both biased in a different way
- at each step, we
  - choose one of the coins
  - throw it and record the obtained number
  - choose if we
    - keep the same coin
    - take the other coin
  - start again …



Hidden Markov Models



initial probabilities

$\pi_{Blue} = 0.5$

$\pi_{Red} = 0.5$

B

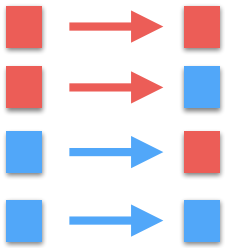
emission probability

Blue coin : 1 p=2/3 ; 2 p=1/3

Red coin: 1 p=1/3 ; 2 p=2/3

A

transition probability



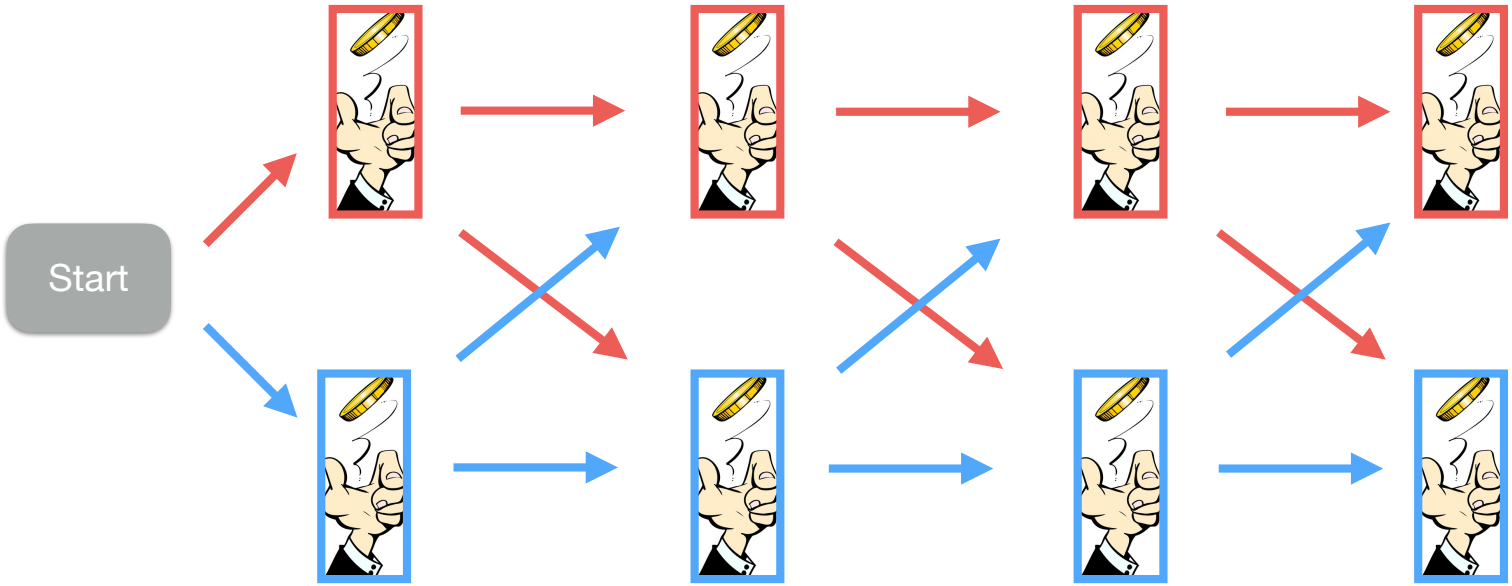
p = 0.33

p = 0.67

p = 0.45

p = 0.55

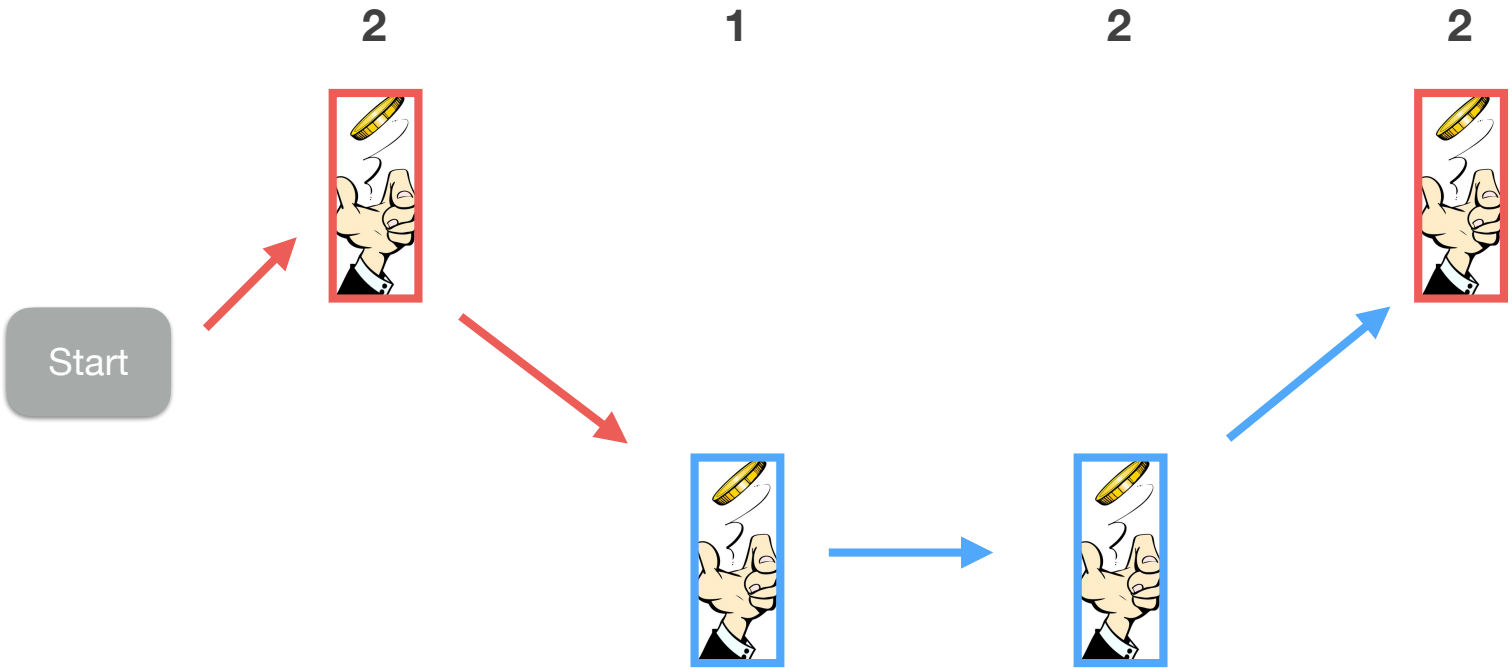
Hidden Markov Models



Markov properties :

1. ***parameters (transition A and emission B) do not change during the process !!***
2. ***Each throw is independent of the previous one***
3. ***Each transition is independent of the previous one***

Hidden Markov Models



initial probabilities

$\pi_{Blue} = 0.5$

$\pi_{Red} = 0.5$

B

emission probability

Blue coin : 1 p=2/3 ; 2 p=1/3

Red coin: 1 p=1/3 ; 2 p=2/3

A

transition probability

p = 0.33

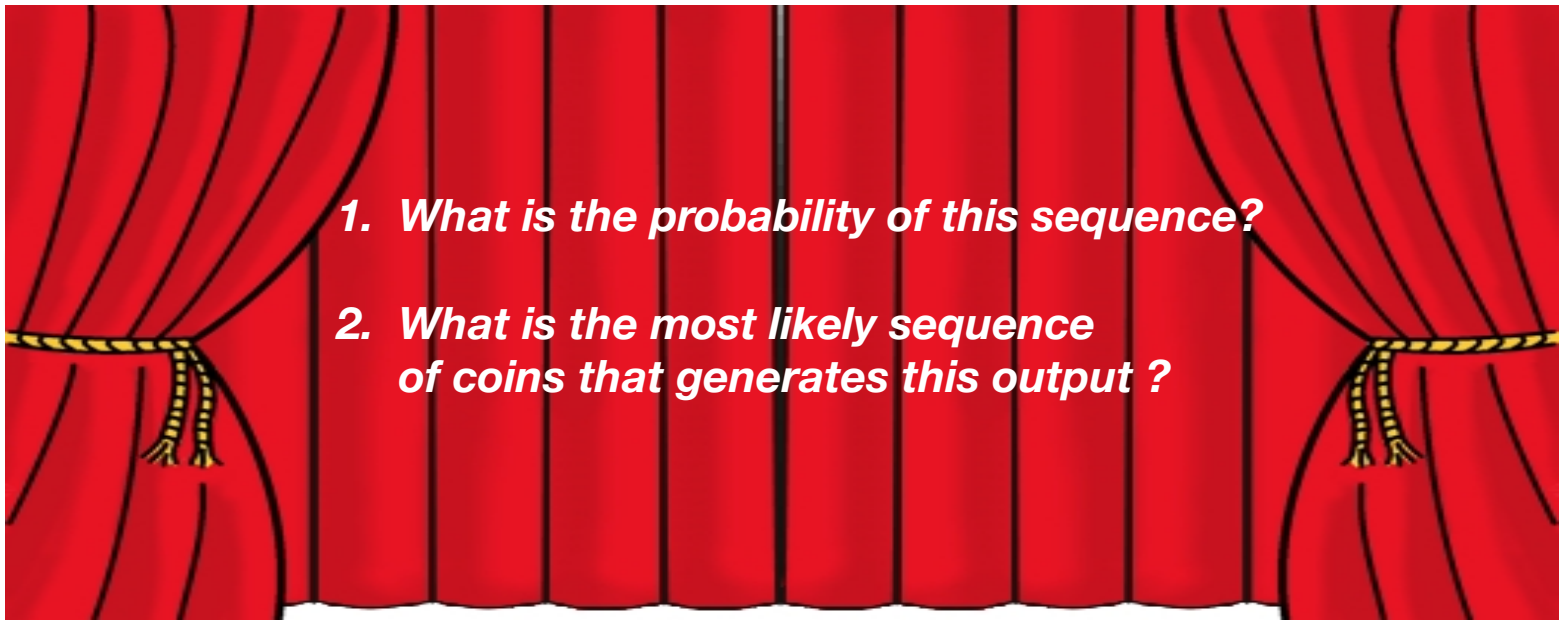
p = 0.67

p = 0.45

p = 0.55

Hidden Markov Models

2 1 2 2



initial probabilities

$\pi_{Blue} = 0.5$

$\pi_{Red} = 0.5$

B

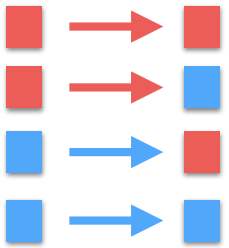
emission probability

Blue coin : 1 p=2/3 ; 2 p=1/3

Red coin: 1 p=1/3 ; 2 p=2/3

A

transition probability



p = 0.33

p = 0.67

p = 0.45

p = 0.55

Hidden Markov Models

- **Probability of an observed sequence**
  - **observed** sequence (length *T*) $\vec{x}$
  - **unobserved** sequence of states (number of states *S*): $\vec{z}$
  - Probability of sequence *x*:

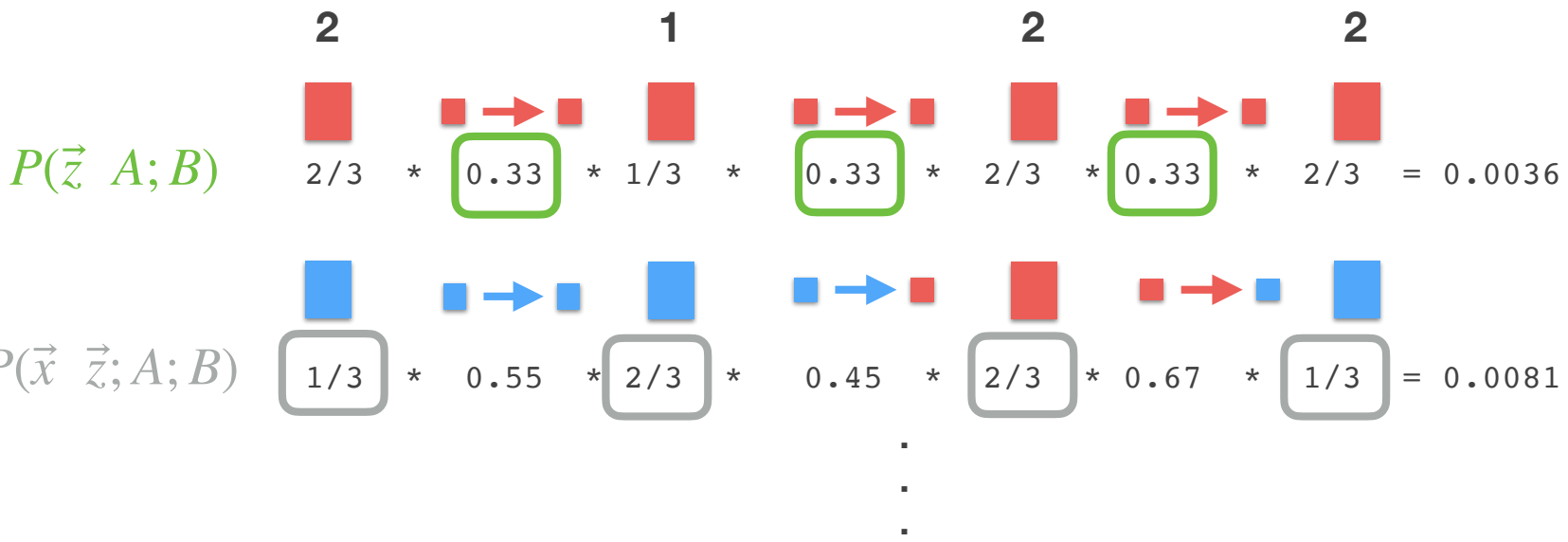
$$P(\vec{x};A;B) = \sum_{\vec{z}} P(\vec{x};\vec{z};A;B)$$
$$= \sum_{\vec{z}} P(\vec{x}|\vec{z};A;B)P(\vec{z};A;B)$$

- Markov assumption:
  parameters
  A and B remain constant

$$P(\vec{x};A;B) = \sum_{\vec{z}} P(\vec{x}|\vec{z};A;B)P(\vec{z};A;B)$$
$$= \sum_{\vec{z}} \left(\prod_{t=1}^{T} P(x_t|z_t;B)\right)\left(\prod_{t=1}^{T} P(z_t|z_{t-1};A)\right)$$
$$= \sum_{\vec{z}} \left(\prod_{t=1}^{T} B_{z_t,x_t}\right)\left(\prod_{t=1}^{T} A_{z_{t-1}z_t}\right)$$

emission probabilities **transition probabilities**

Hidden Markov Models



initial probabilities

$\pi_{Blue} = 0.5$

$\pi_{Red} = 0.5$

B

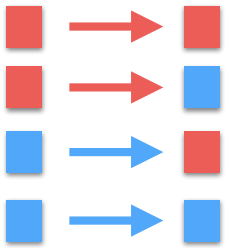
emission probability

Blue coin : 1 p=2/3 ; 2 p=1/3

Red coin: 1 p=1/3 ; 2 p=2/3

A

transition probability



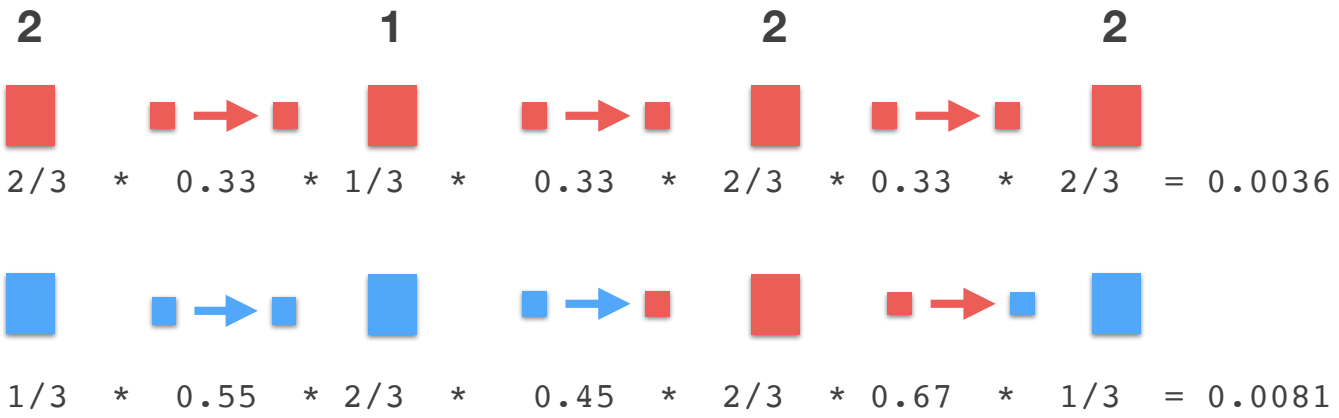
p = 0.33

p = 0.67

p = 0.45

p = 0.55

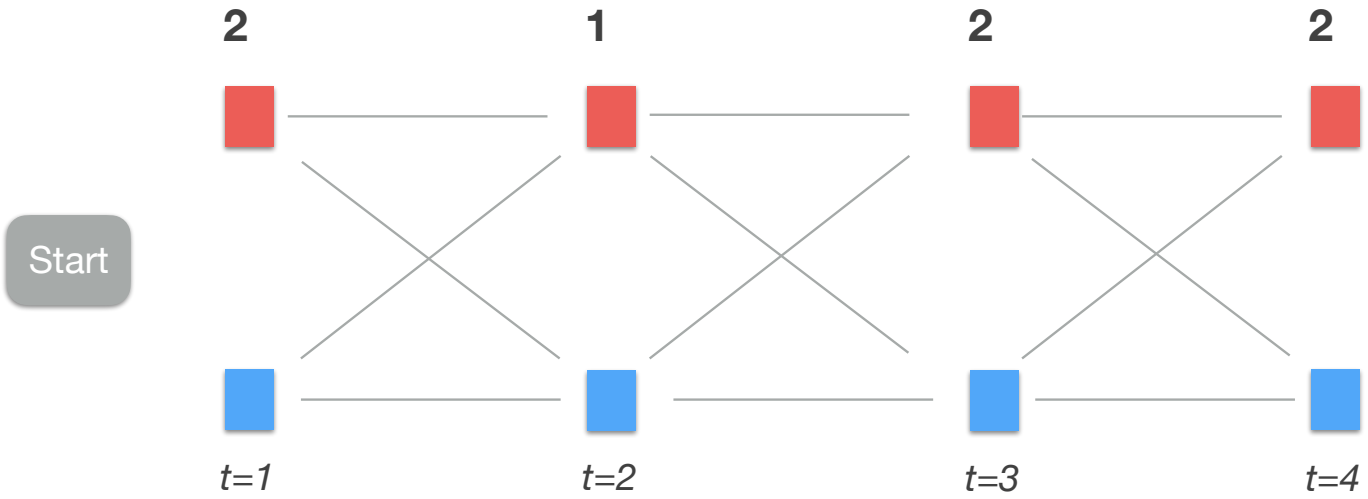
Hidden Markov Models



How many possible sequences ? → $2^4 = 16$
In general : → $o(|S|^T)$ 😩

(Chromatin states : S = 18 ; T = 10^7 …)

Hidden Markov Models



To compute the probability of an observed sequence, we use the **Forward algorithm** (dynamical programming)

For each time point t_0 and each possible state at $t=t_0$, compute the total probability for the observed sequence until t_0

P(2-1-2, ■) = probability of observing the sequence 2-1-2 and being in the red state at *t=3*

Hidden Markov models

- **Forward algorithm** :
  - probability of observing sequences until *t*, and being in state *i* at *t*

$$\alpha_i(t) = P(x_1, x_2, \ldots, x_t, z_t = s_i; A; B)$$

  - Total probability:

$$P(\vec{x}; A; B) = \sum_{i=1}^{|S|} \alpha_i(T)$$

  - Dynamic programming:

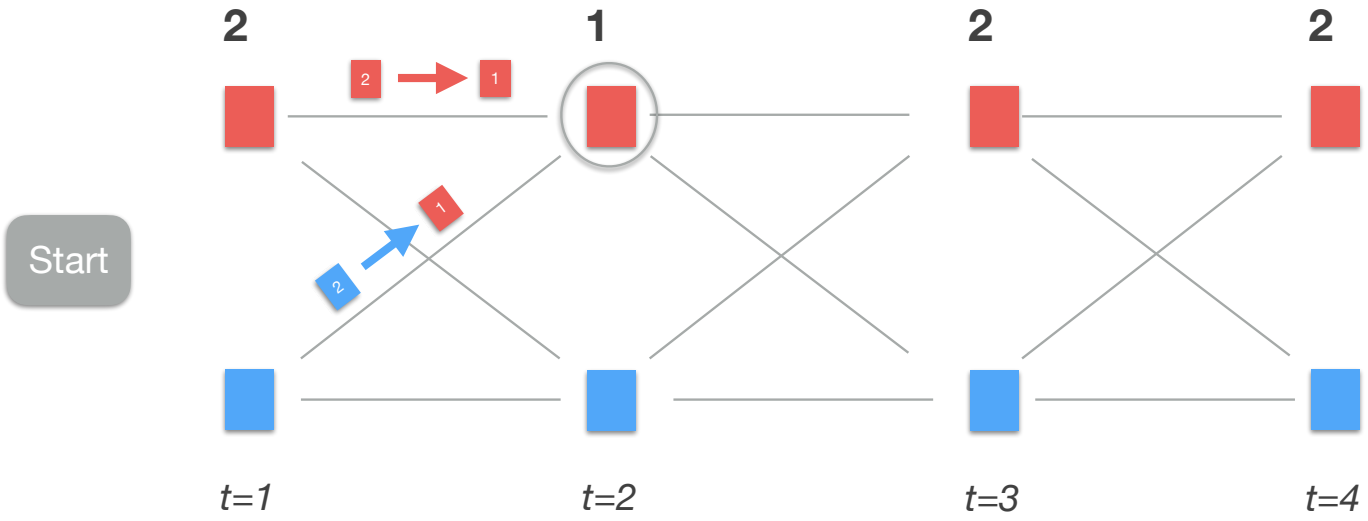
Algorithm 1 Forward Procedure for computing $\alpha_i(t)$

1. Base case: $\alpha_i(0) = A_{0\,i}$, $i = 1..|S|$
2. Recursion: $\alpha_j(t) = \sum_{i=1}^{|S|} \alpha_i(t-1) A_{ij} B_{j\,x_t}$, $j = 1..|S|$, $t = 1..T$

  - Complexity: ***o(|S|² T)***

😃

Hidden Markov Models

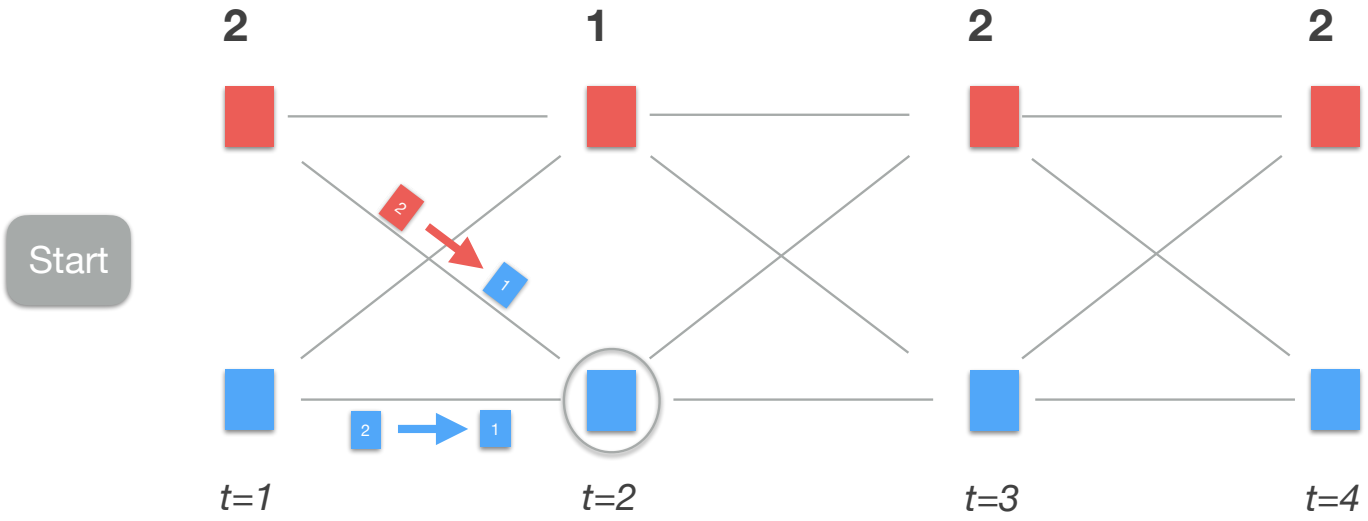


Probability red=2 * probability red→ red * probability red=1
= 2/3 * 1/3 * 1/3 = 0.074

Probability blue=2 * probability blue→ red * probability red=1
= 1/3 * 0.45 * 1/3 = 0.05

Probability(2-1,) = 0.074 + 0.05 = 0.124

Hidden Markov Models

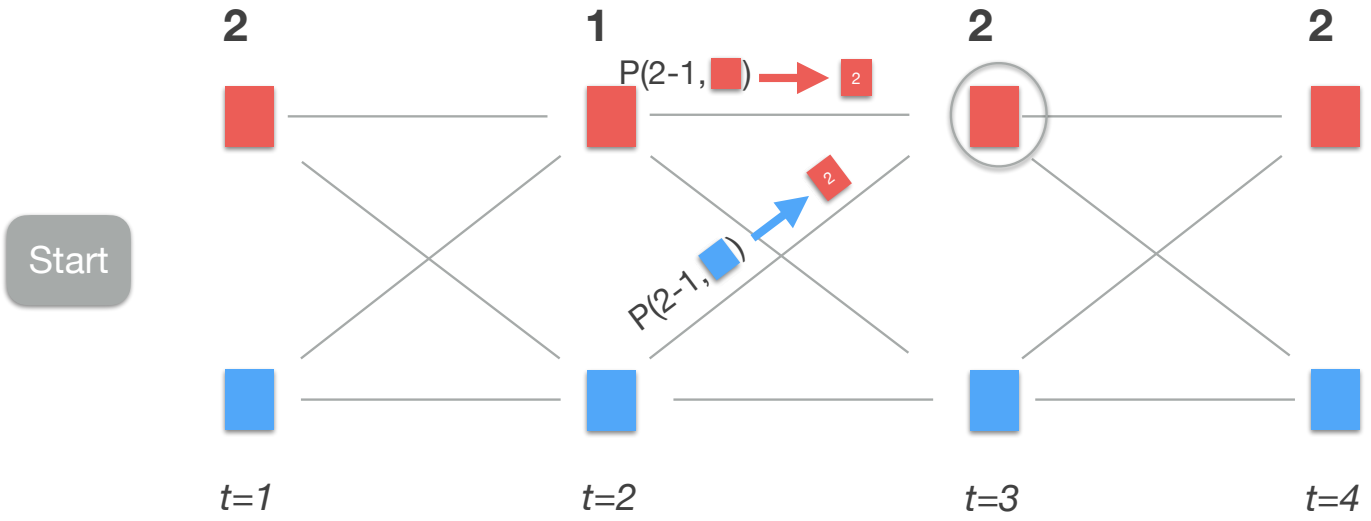


Probability blue=2 * probability blue→ blue * probability blue=1
= 1/3 * 0.55 * 2/3 = 0.12

Probability red=2 * probability red→ blue * probability blue=1
= 2/3 * 2/3 * 2/3 = 0.296

Probability(2-1,) = 0.12 + 0.296 = 0.416

Hidden Markov Models



P(2-1,) → 2

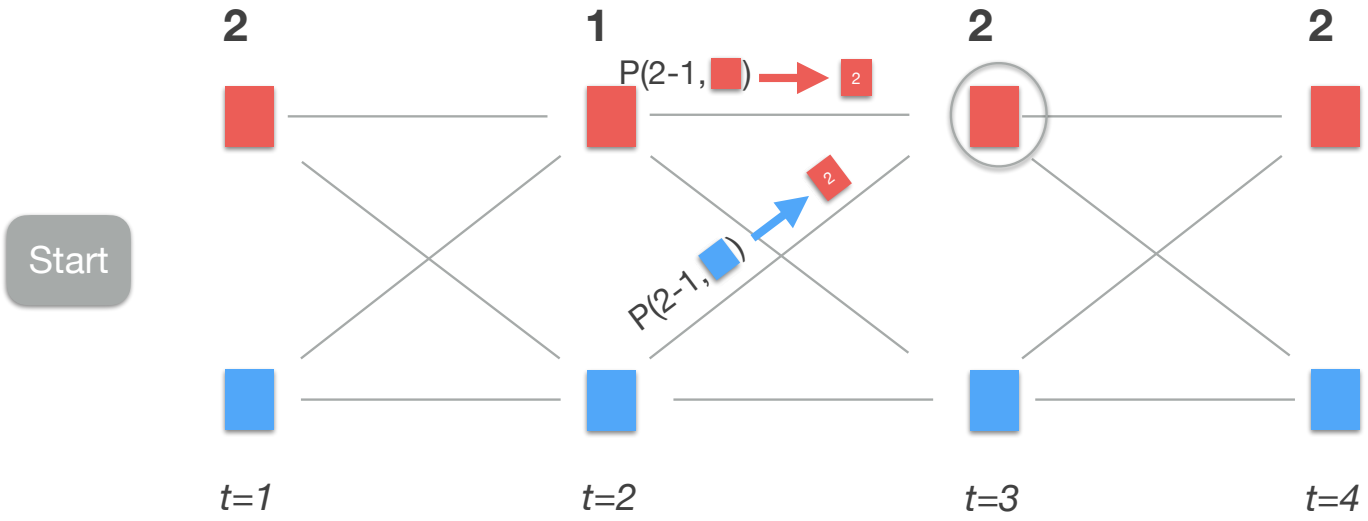
P(2-1) * probability red→ red * probability red=2
= 0.124* 1/3 * 2/3 = 0.0276

P(2-1,) → 2

P(2-1) * probability blue→ red * probability red=2
= 0.416 * 0.45 *2/3 = 0.1249

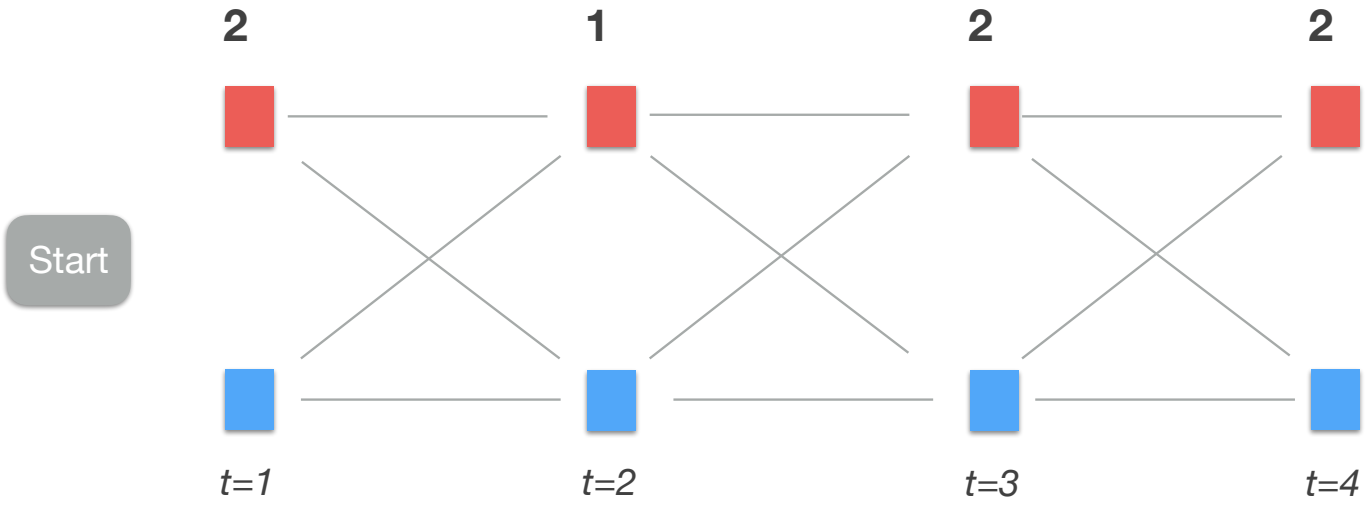
Probability(2-1-2,) = 0.0276 + 0.1249 = 0.1525

Hidden Markov Models



- at each time point, we perform *S***S* operations (S = number of possible states)
- all together: S*S*T operations → complexity $o(S^2 \cdot T)$

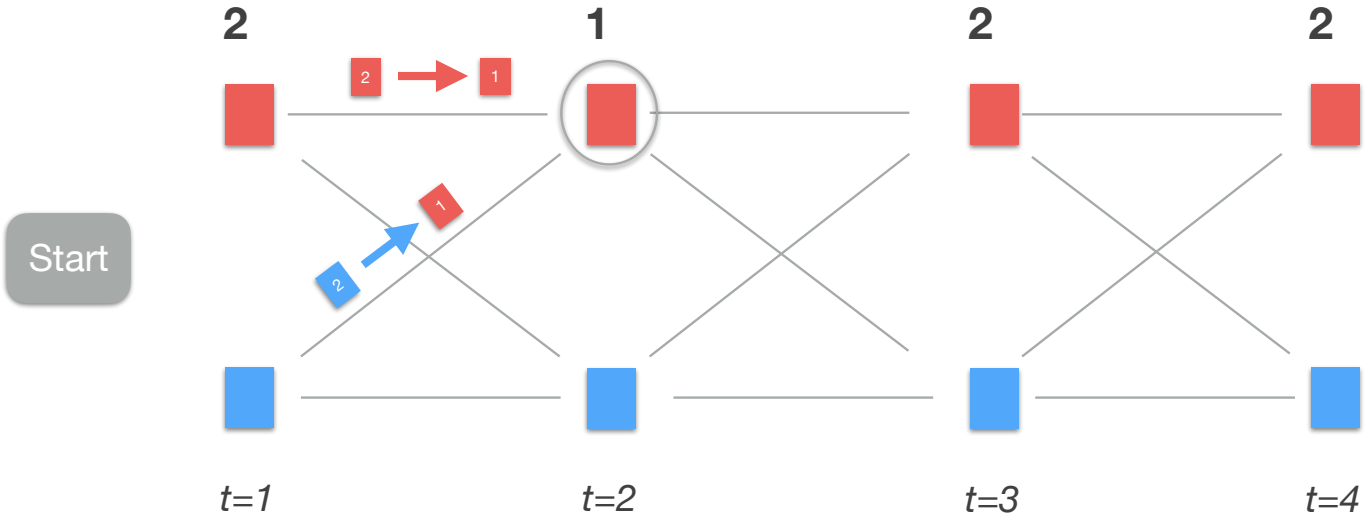
Viterbi algorithm

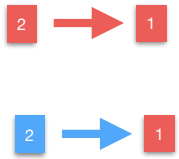


What is the most likely sequence of states leading to the observed sequence ? → **Viterbi algorithm**

For each time point and each possible state, **record the most probable path leading to this state** at this time point

Viterbi algorithm



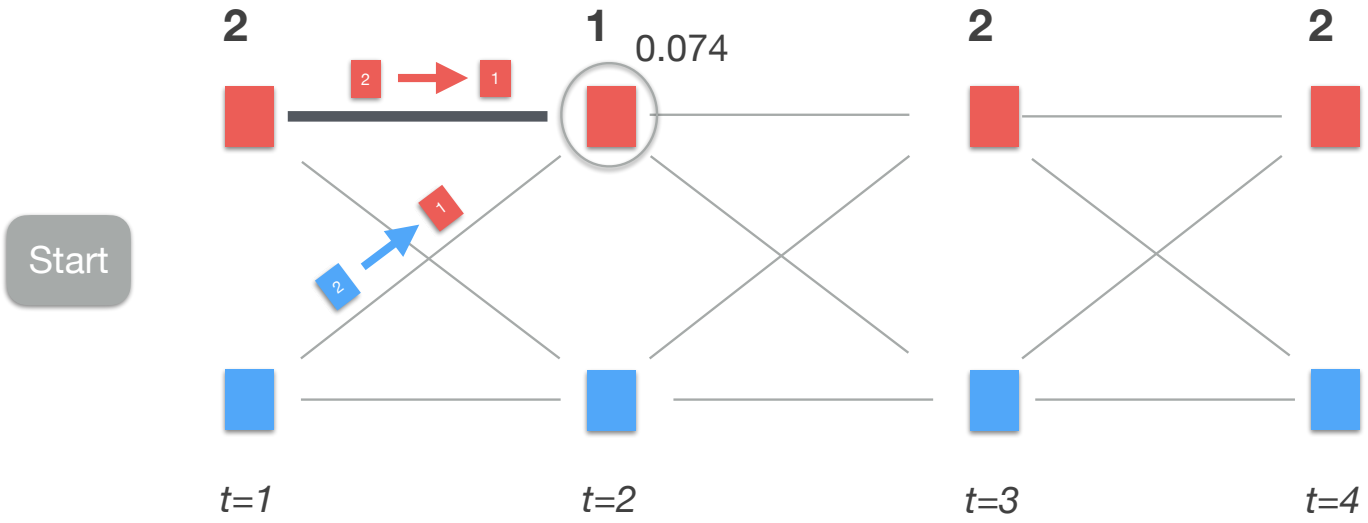


Probability red=2 * probability red→ red * probability red=1
= 2/3 * 1/3 * 1/3 = 0.074

Probability blue=2 * probability blue→ red * probability red=1
= 1/3 * 0.45 * 1/3 = 0.05

Path **red→ red** (0.074) is more likely than **blue → red** (0.05): we thus assign **the highest of the 2 probabilities** to the red node

Viterbi algorithm

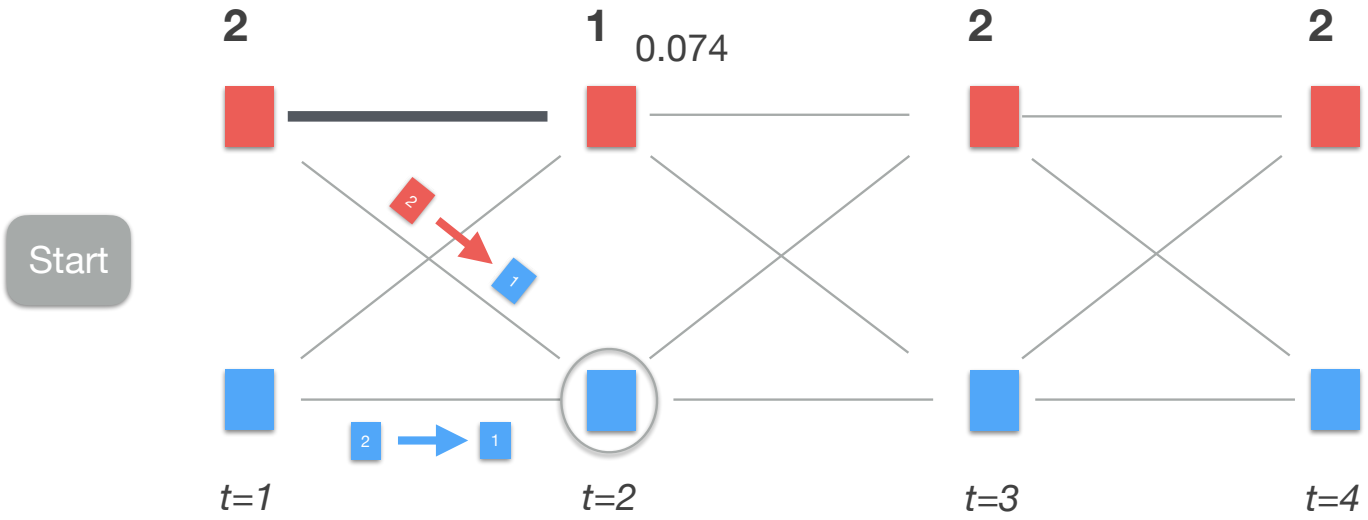


Probability red=2 * probability red→ red * probability red=1
= 2/3 * 1/3 * 1/3 = 0.074

Probability blue=2 * probability blue→ red * probability red=1
= 1/3 * 0.45 * 1/3 = 0.05

*This is **unlike the Forward algorithm**, where we would assign the sum of the 2 probabilities (0.074+0.05 = 0.124) to the red node!! (compare with slide 129)*

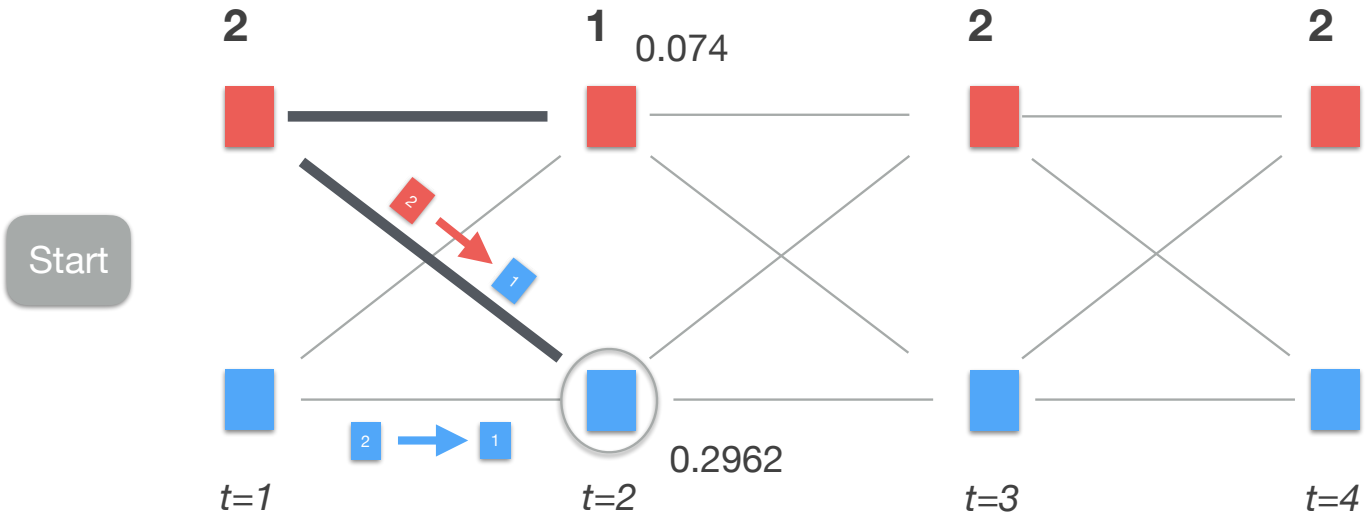
Viterbi algorithm



Probability red=2 * probability red→ blue * probability blue=1
= 2/3 * 2/3 * 2/3 = 0.2962

Probability blue=2 * probability blue→ blue * probability blue=1
= 1/3 * 0.55 * 2/3 = 0.1222

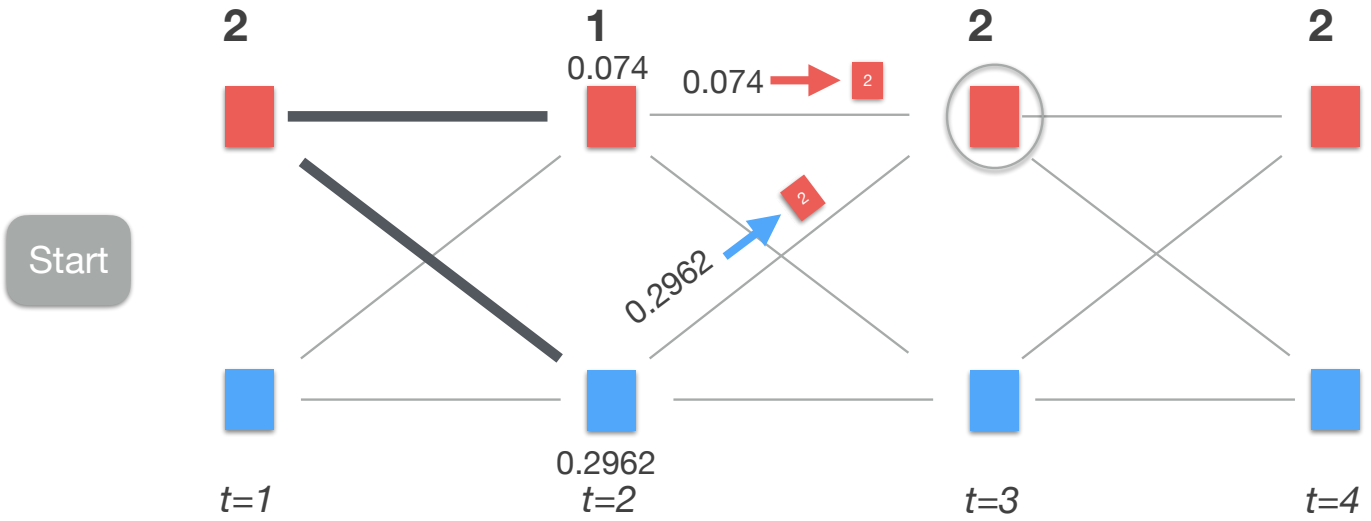
Viterbi algorithm



Probability red=2 * probability red→ blue * probability blue=1
= 2/3 * 2/3 * 2/3 = 0.2962

Probability blue=2 * probability blue→ blue * probability blue=1
= 1/3 * 0.55 * 2/3 = 0.1222

Viterbi algorithm



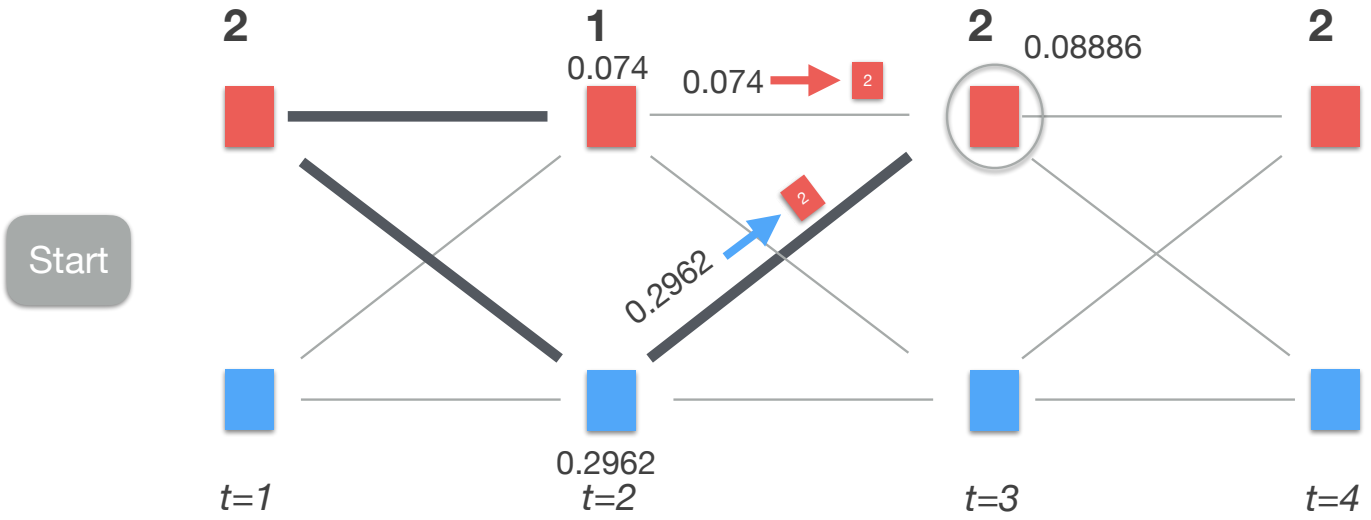
0.074 → 2

0.074 * probability red → red * probability red=2
= 0.074 * 1/3 * 2/3 = 0.01644

0.2962 → 2

0.2962 * probability blue → red * probability red=2
=0.2962 * 0.45 * 2/3 = 0.08886

Viterbi algorithm



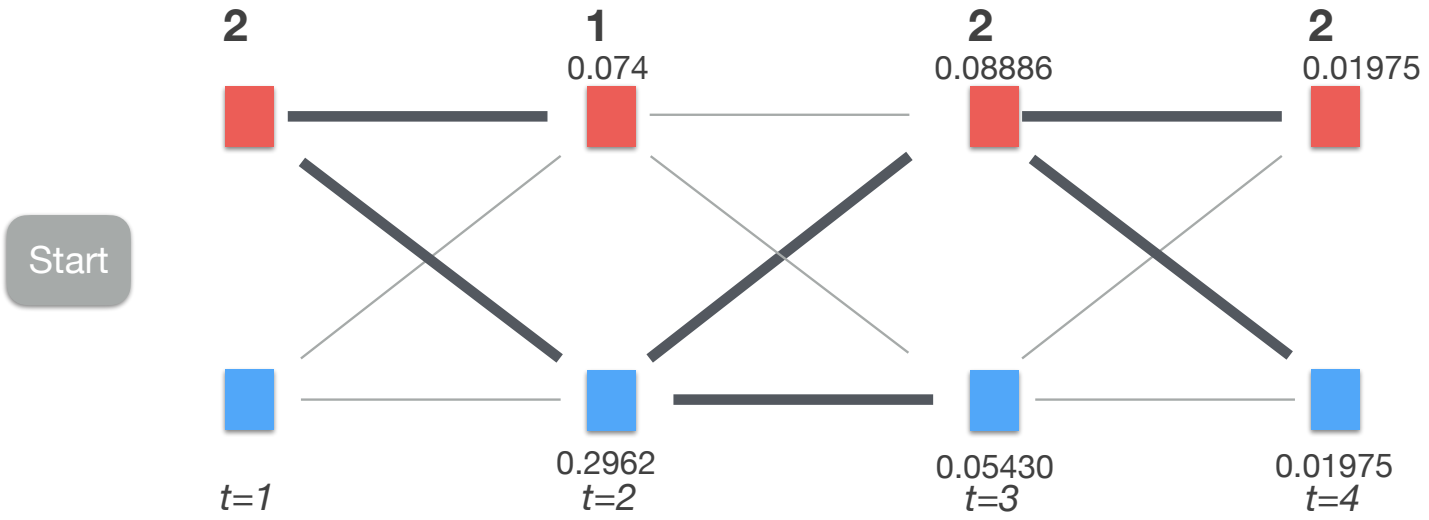
0.074 → 2

0.074 * probability red → red * probability red=2
= 0.074 * 1/3 * 2/3 = 0.01644

0.2962 → 2

0.2962 * probability blue → red * probability red=2
=0.2962 * 0.45 * 2/3 = 0.08886

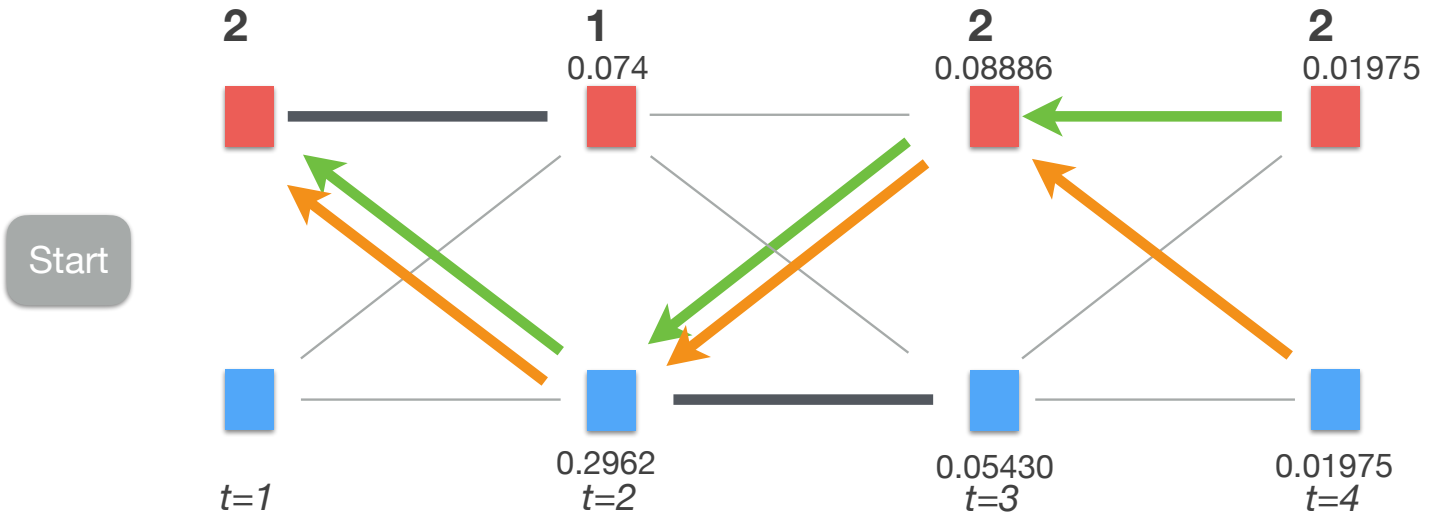
Viterbi algorithm



We have two end nodes with equal probability!
Hence, we have 2 optimal path obtained by backtracking from the 2 possible optimal final states!

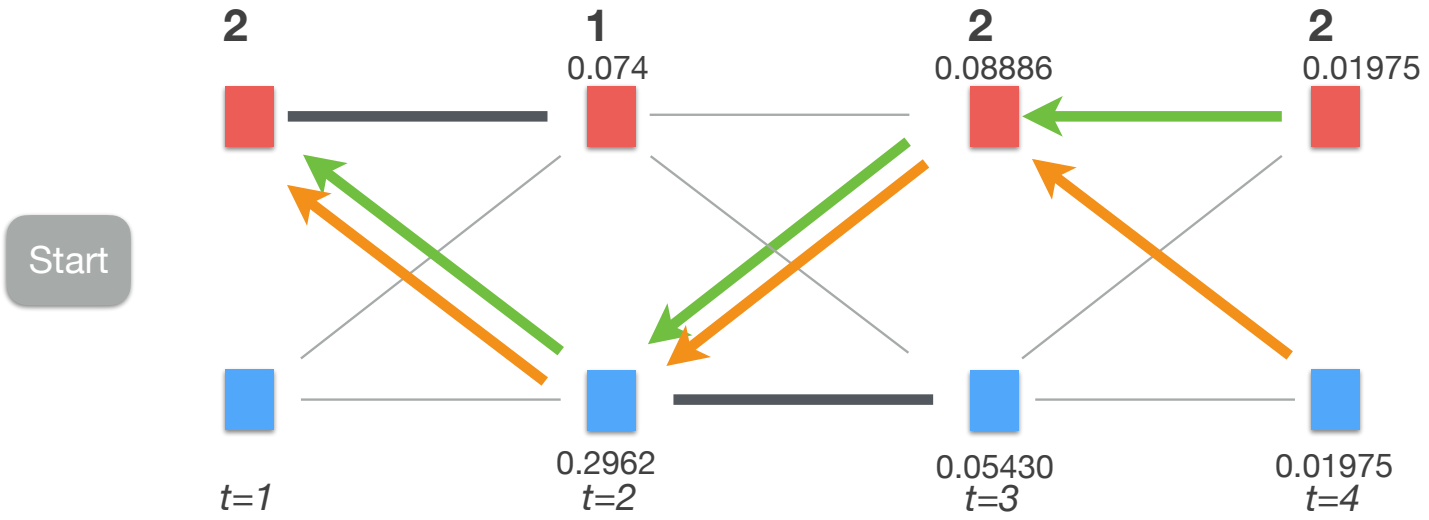
Generally, we would have only one end node with highest probability hence a single backtracking path!

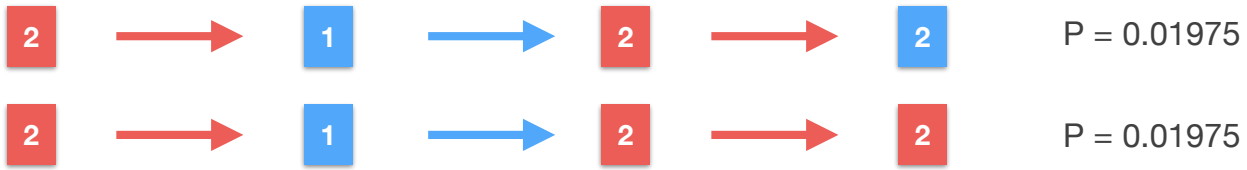
Viterbi algorithm



We have **two final states with equal probability**!
Hence, we have 2 optimal path (**green** and **orange**)
obtained by backtracking from the 2 possible optimal final states!

Viterbi algorithm

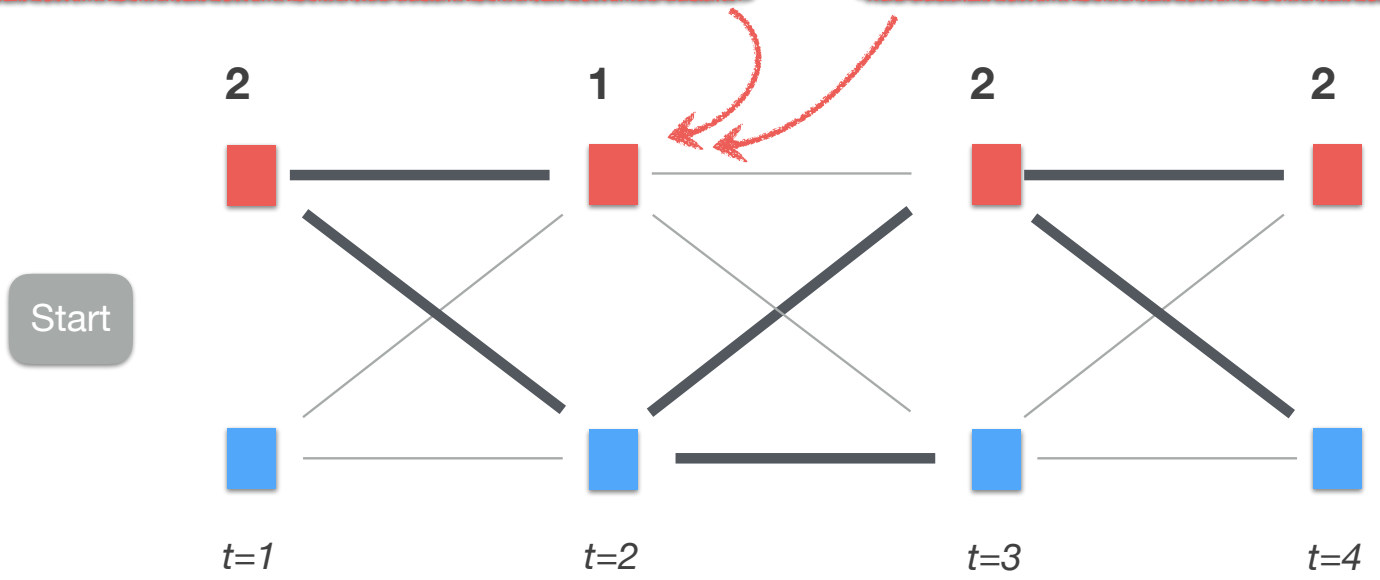




Summary: difference Forward / Viterbi

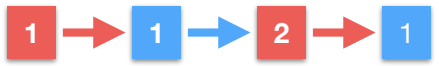
Forward: assign to the node at *t* the sum of the probabilities of **all path leading to that node** from *t-1*

Viterbi: assign to the node at *t* the probability of **the most likely path leading to that node** from *t-1* + backtracking

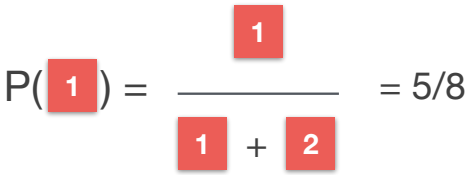


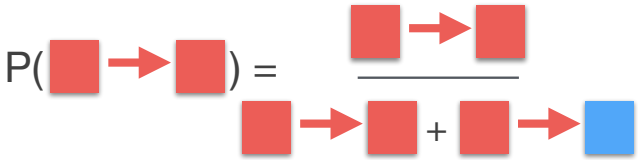
HMM - learning parameters

- So far, we assumed that the parameters A and B are given…
  How can we determine them ?
  → use a set of **training sequences** with **known states**



etc …

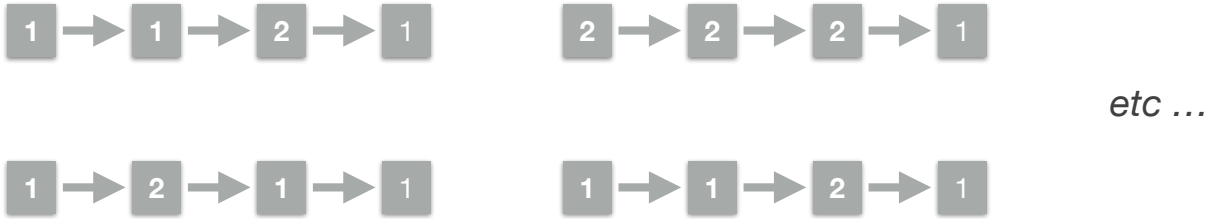




etc …

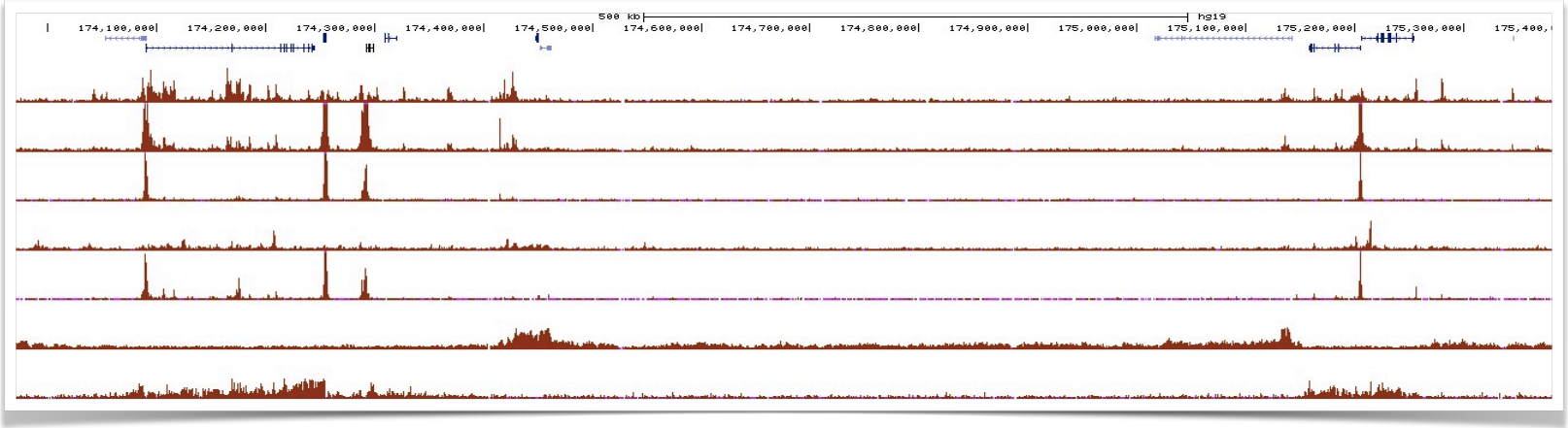
HMM - learning parameters

- So far, we assumed that the parameters A and B are given…
  How can we determine them ?
  → use a set of **training sequences** with **unknown states**



- Initialize randomly parameters; update these parameters iteratively until convergence (*Baum-Welch* algorithm, based on **Expectation Maximization**)

HMM and Chromatin states



n histone marks;
Binarize the data in windows of 200bp (using appropriate thresholds)

```
H3K4me3   :1000010010101000000000000000000000000000001111011010100 01
H3K4me1   :0010010001000100010000010000000010000000001001010111111 1000
H3K27ac   :0000100010001000100000111111000011111000001101000001011 1010
H3K36me3  :0010000010001010101001010001111100001111000011010101010 1
H3K27me3  :0010001111110001010101001010000011111000100101001010011
H3K9me3   :0000101011101000011110000010010101001101010100010100000
```

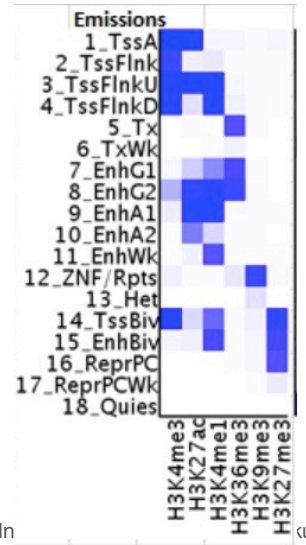
[ChromHMM]

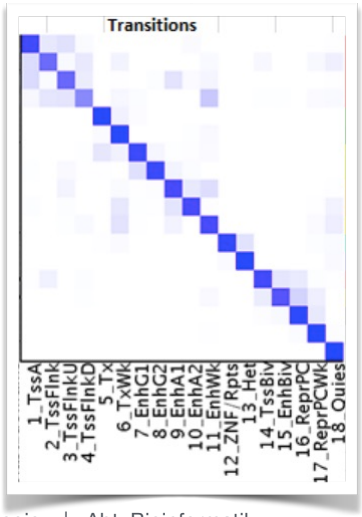
Each observation is a binary vector of length n in a genomic window of 200bp

Chromatin states

We postulate the existence of *k* (= 15 or 18) chromatin states
Each state has different **emission probability** for each histone mark …

Emission and transition probabilities are learned from training data:





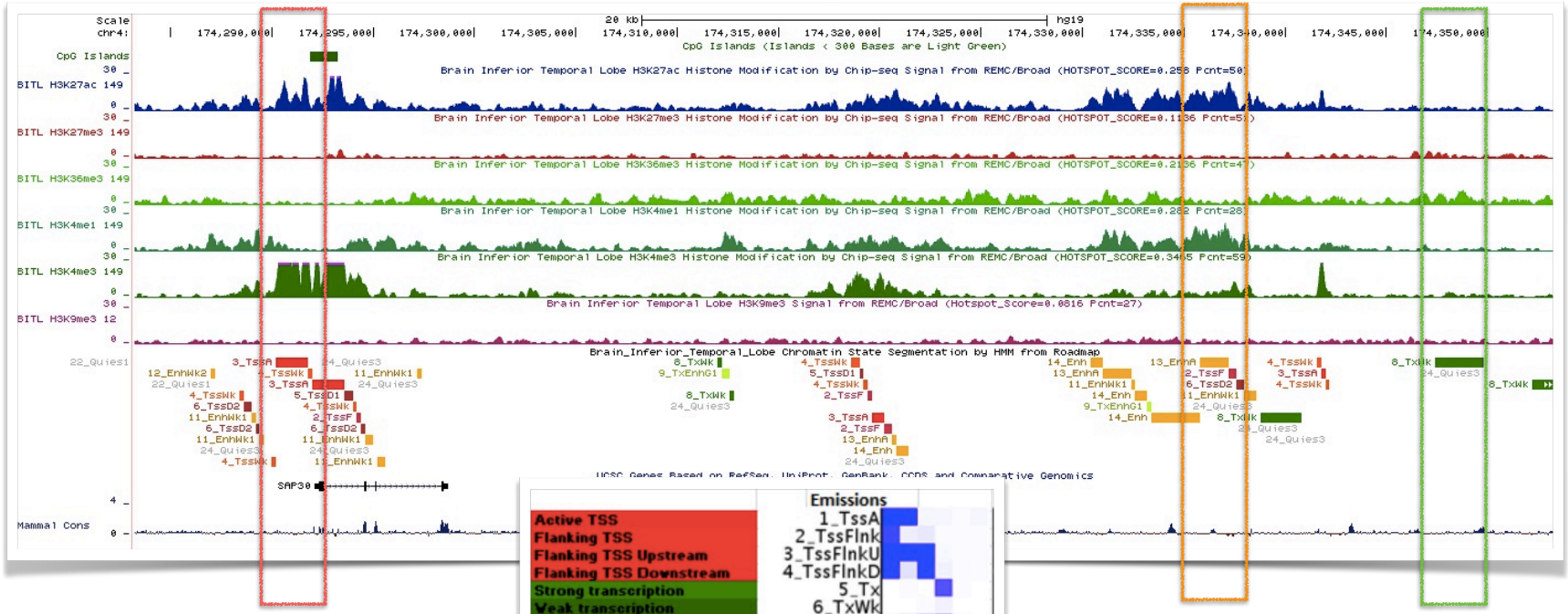
Chromatin states

- Given that we observe

```
H3K4me3   :1000010010101000000000000000000000000000000011101101010001
H3K4me1   :0010010001000100000010000000010000000000010010101111111000
H3K27ac   :0000100010001000001111110000111111000011010000101111010
H3K36me3  :0010000010001010101001010001111100001111000011010101011
H3K27me3  :0010001111100010101010010100000111110001001010010101001 1
H3K9me3   :0000101011101000011110000100101010011010101000101000000
```

… what is the most likely sequence of chromatin states leading to these observed values ? → ***Viterbi algorithm***

Chromatin states

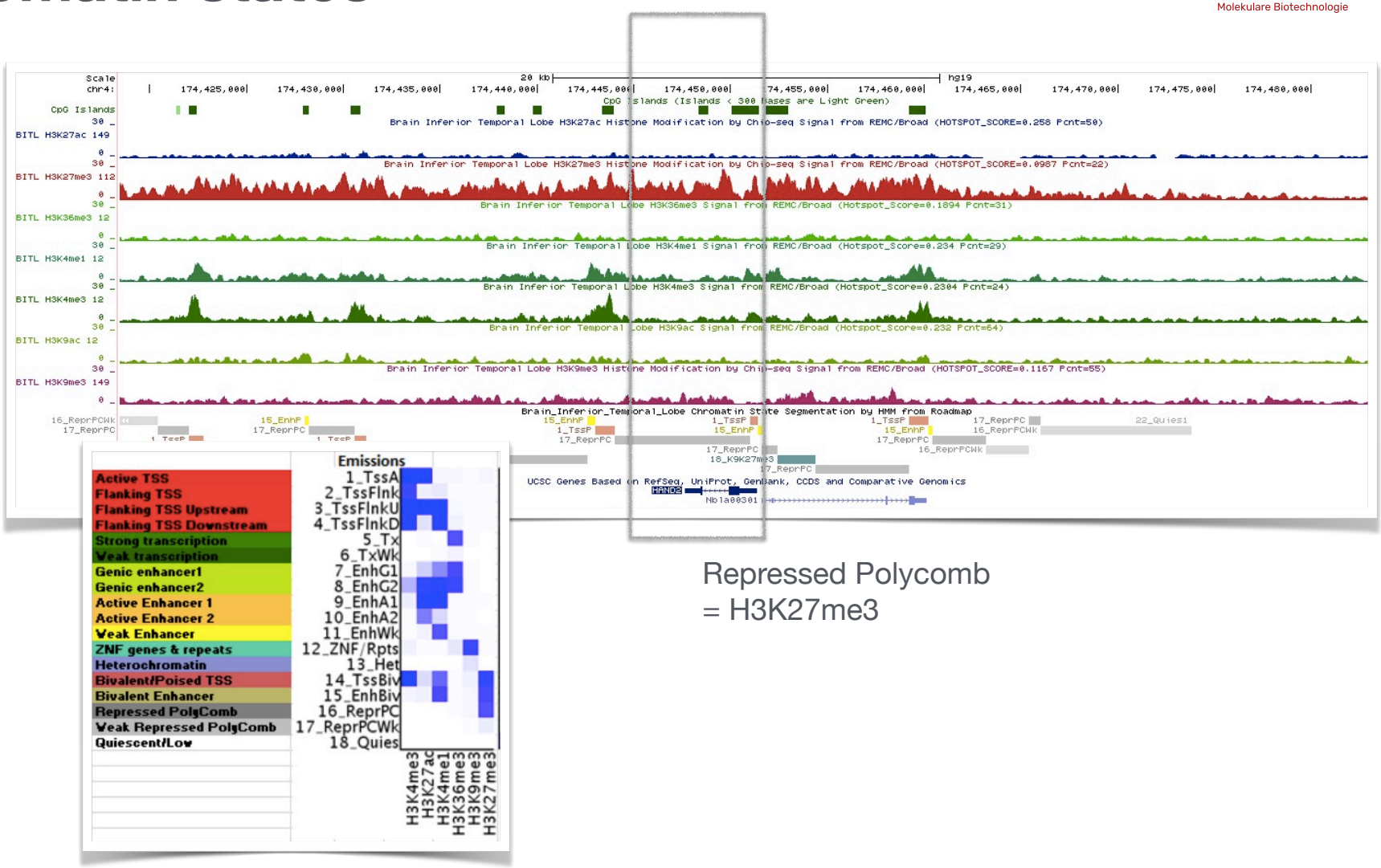


Active Transcription Start Site
= H3K4me3 + H3K27ac

Active Enhancer
= H3K4me1 + H3K27ac

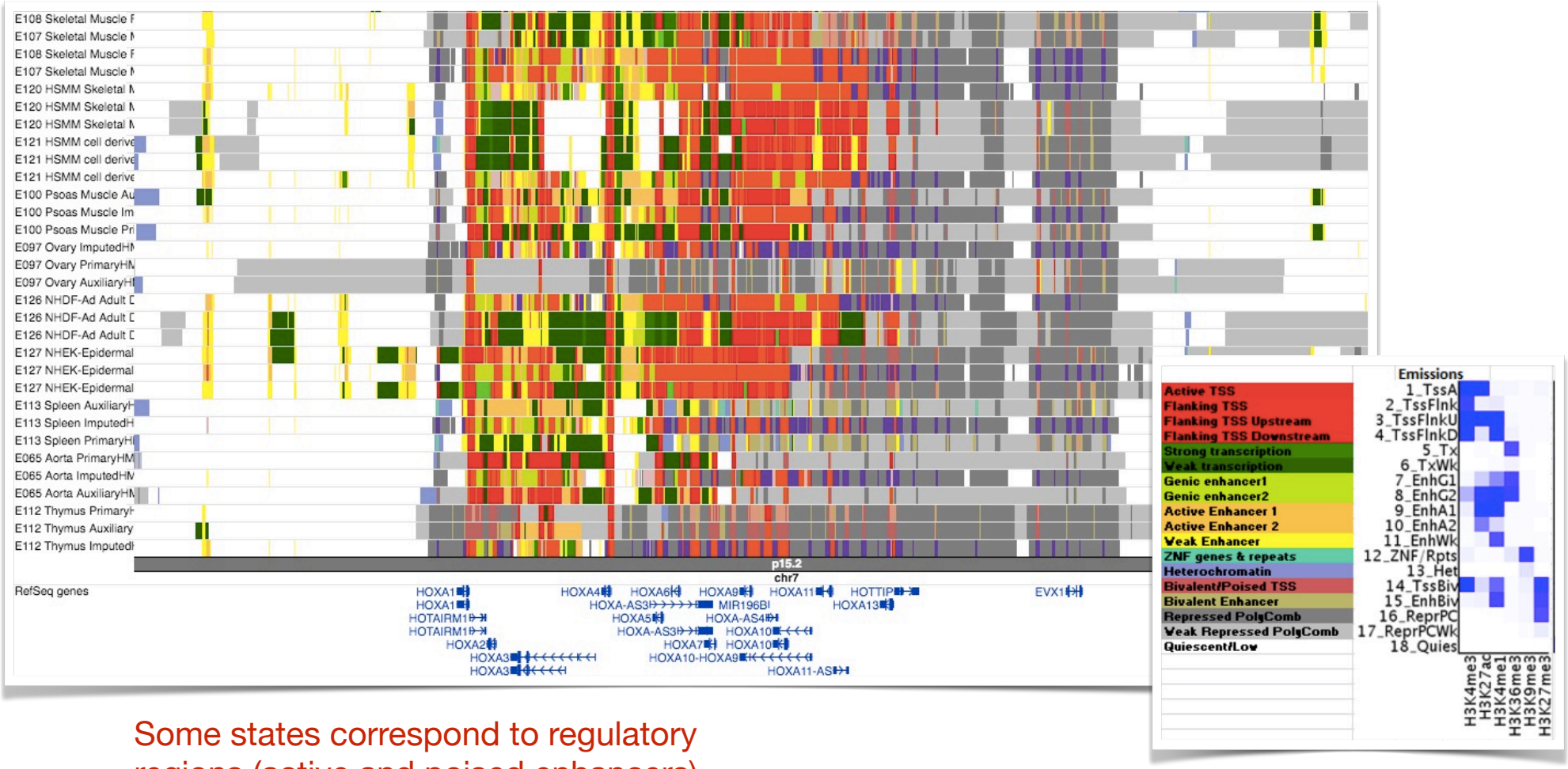
Transcribed region
=H3K36me3

Chromatin states



Repressed Polycomb
= H3K27me3

Roadmap chromatin segmentation in different human adult tissues

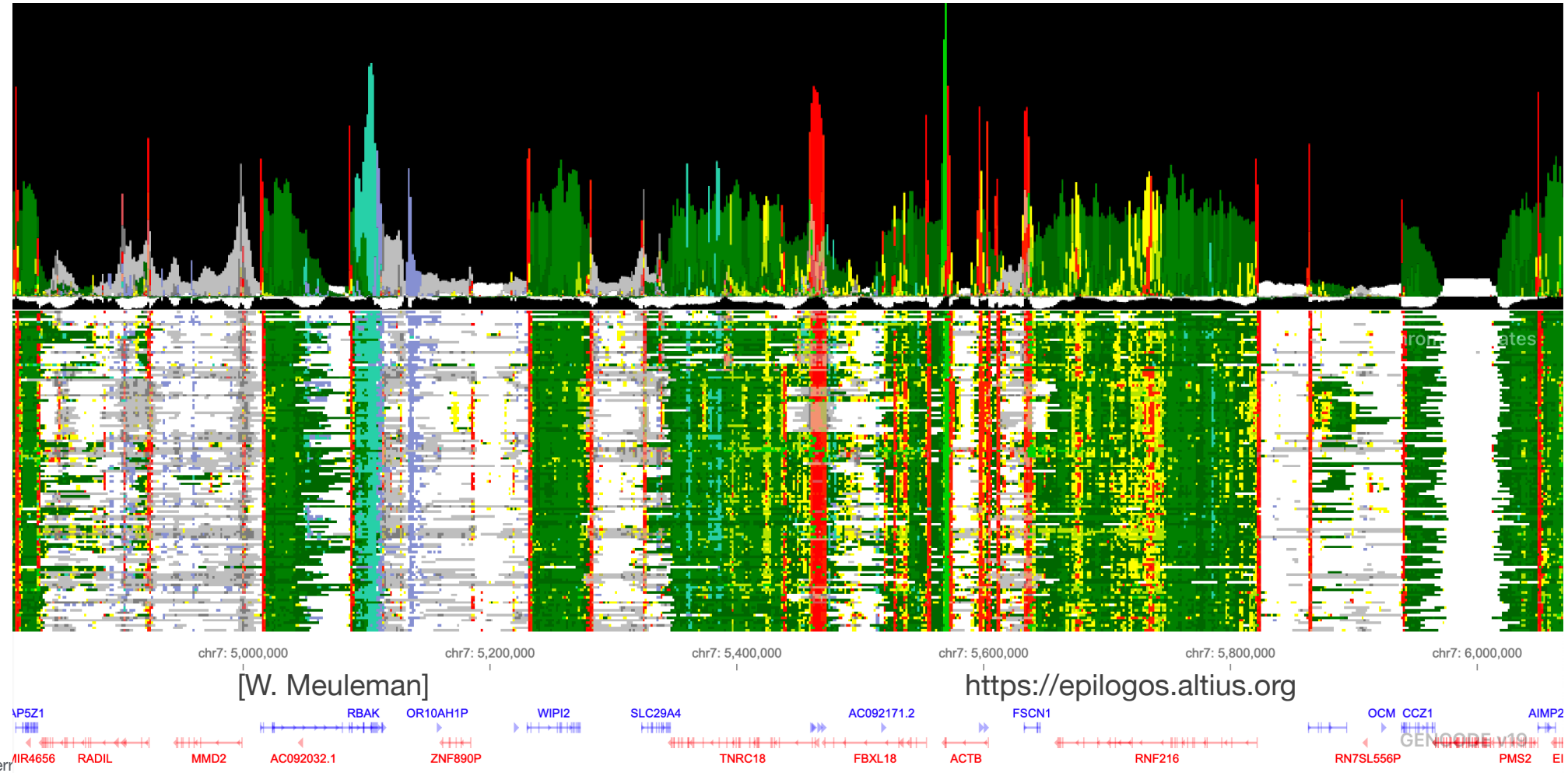


Some states correspond to regulatory regions (active and poised enhancers)

→ **motif search can be restricted to these regions**

http://epigenomegateway.wustl.edu

EpiLogos



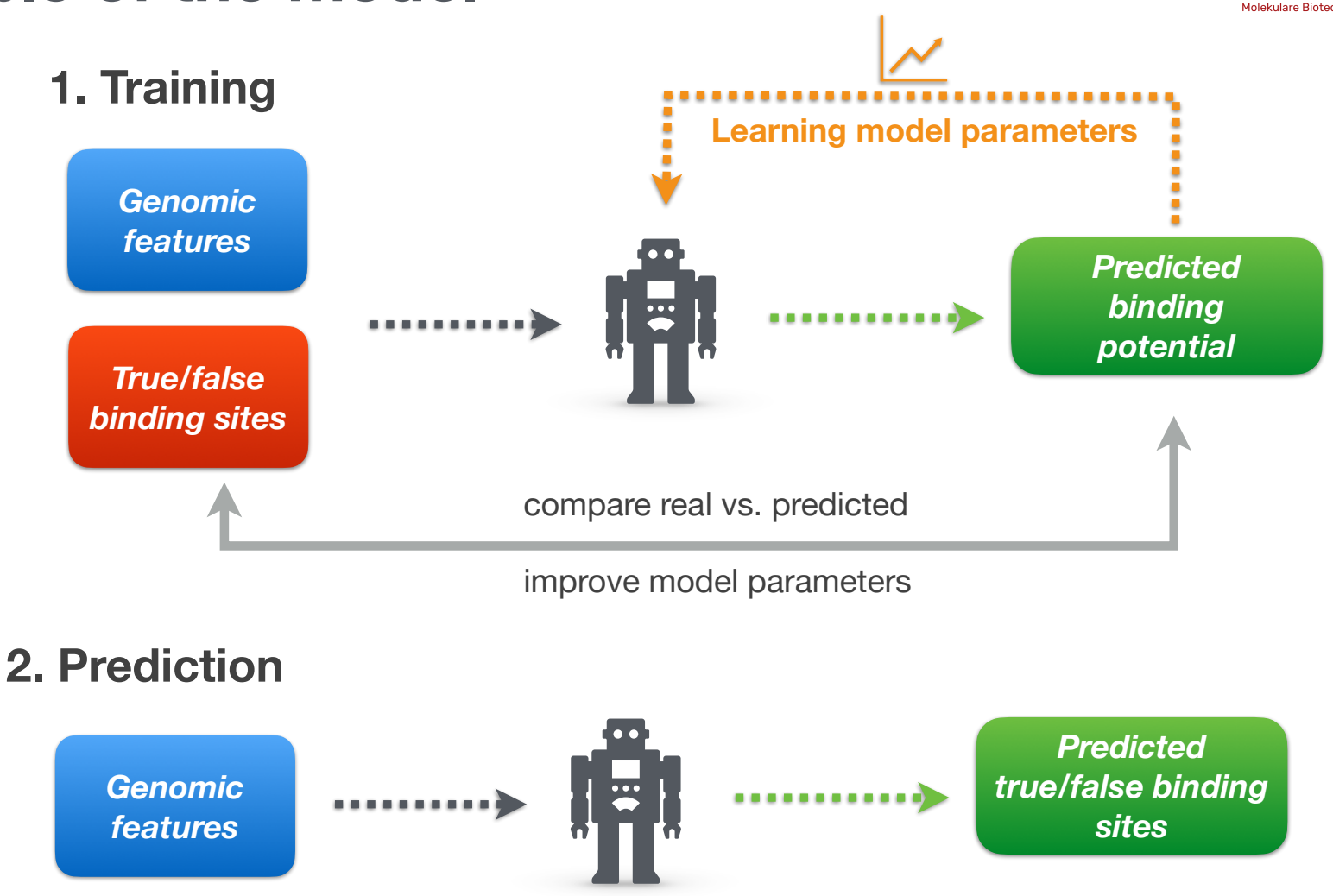
Carl Herr

Machine-learning approach to TFBS prediction

- Define a **General Binding Potential** (GBP) for each base in the genome using a **Logistic Regression Approach** (LRA) based on a set of features
  → ***What features discriminate best bound/unbound bases ?***
- Training set
  - **positive** : center of ChIP peaks for 14 TF datasets is various cell lines
  - **negative** : randomly sampled bases (50x more negative sequences)
- Features
  - sequence features : evolutionary conservation
  - distance to TSS
  - density of ChIP-seq reads for various histone marks and Pol2
  - presence of DNAse I hypersensitive region
  - …

29 features
no PWM data

Principle of the model



Machine-learning approach to TFBS prediction

IPMB Institut für Pharmazie und Molekulare Biotechnologie

UNIVERSITÄT HEIDELBERG ZUKUNFT SEIT 1386

evolutionary features

sequence properties

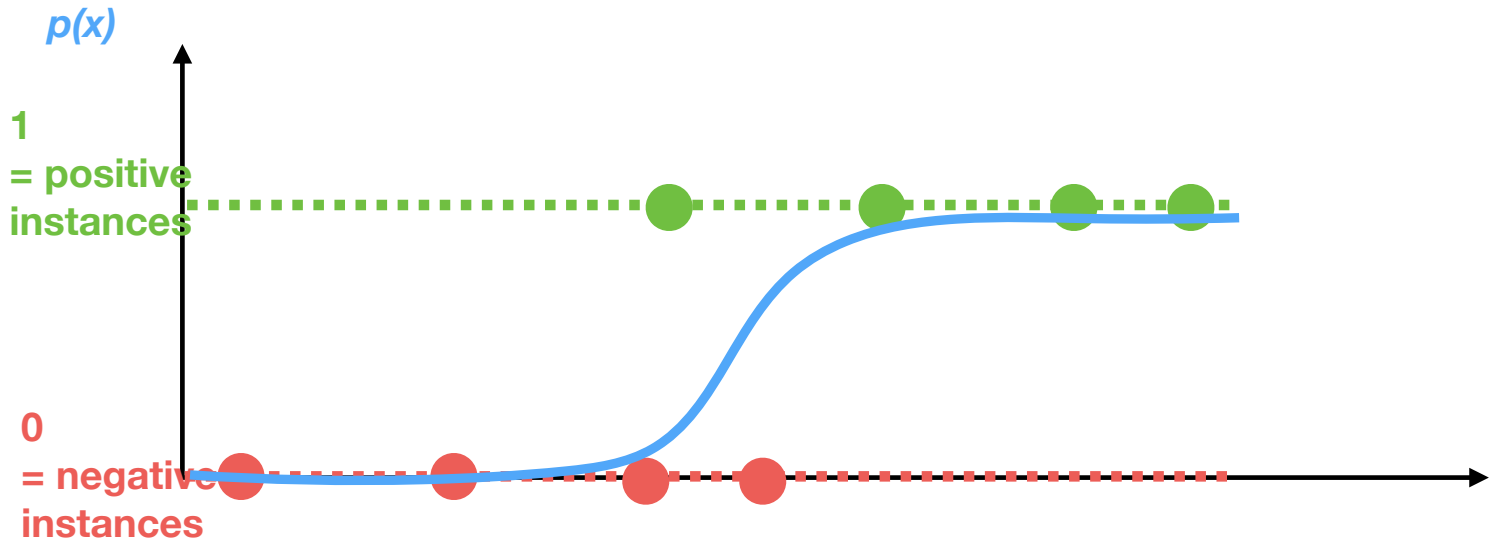
chromatin properties

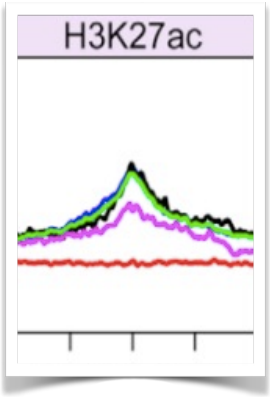
Table 1. The 29 features that were used to compute a GBP probability of transcription factor binding at specific locations

Feature no.	Feature description	Reference
1	PhastCons score for 28-way vertebrate alignment; 0 if not available	Siepel et al. 2005; Miller et al. 2007
2	PhastCons score for placental mammal subset (18 species); 0 if not available	Siepel et al. 2005; Miller et al. 2007
3	1 if PhastCons vertebrate score is available and the score is 0; 0 otherwise	Siepel et al. 2005; Miller et al. 2007
4	1 if PhastCons placental mammal score is available and the score is 0; 0 otherwise	Siepel et al. 2005; Miller et al. 2007
5	1 if PhastCons score is not available; 0 otherwise	Siepel et al. 2005; Miller et al. 2007
6	1 if part of PhastCons highly conserved vertebrate element; 0 otherwise	Siepel et al. 2005; Miller et al. 2007
7	1 if part of PhastCons highly conserved placental mammal element; 0 otherwise	Siepel et al. 2005; Miller et al. 2007
8	1 if part of a conserved indel region; 0 otherwise	Lunter et al. 2006
9	$\ln(x + 5)$ where x is distance in base pairs to nearest base of a vertebrate PhastCons element (x is 0 if base is in a highly conserved element)	Siepel et al. 2005; Miller et al. 2007
10	$\ln(x + 5)$ where x is distance in base pairs to nearest base of a placental mammal PhastCons element (x is 0 if base is in a highly conserved element)	Siepel et al. 2005; Miller et al. 2007
11	$\ln(x + 5)$ where x is distance in base pairs to nearest of a conserved indel region (x is 0 if base is in a highly conserved element)	Lunter et al. 2006
12	The estimated melting temperature at the base	Liu et al. 2007
13	Percentage of G or C base pairs of all bases within 50 bases in either direction	Karolchik et al. 2008
14	1 if base is in a UCSC Genome Browser table of CpG islands; 0 otherwise	Karolchik et al. 2008
15	1 if base is part of a repeat element based on RepeatMasker and Tandem Repeats Finder as provided by UCSC Genome Browser	http://www.repeatmasker.org; Benson 1999; Kent et al. 2002
16	1 if base is part of a transcribed region of a RefSeq gene; 0 otherwise	Pruitt et al. 2007; Karolchik et al. 2008
17	1 if base is between the start and end of the coding region of a RefSeq gene; 0 otherwise	Pruitt et al. 2007; Karolchik et al. 2008
18	1 if base is part of a RefSeq exon; 0 otherwise	Pruitt et al. 2007; Karolchik et al. 2008
19	1 if base is part of a RefSeq exon and within the coding region of the gene; 0 otherwise	Pruitt et al. 2007; Karolchik et al. 2008
20	1 if base is part of a RefSeq intron; 0 otherwise	Pruitt et al. 2007; Karolchik et al. 2008
21	1 if in a RefSeq 3′ UTR; 0 otherwise	Karolchik et al. 2008; Pruitt et al. 2007
22	1 if in a RefSeq 5′ UTR; 0 otherwise	Pruitt et al. 2007; Karolchik et al. 2008
23	$\ln(x + 5)$, where x is the absolute number of base pairs to nearest RefSeq transcription start site	Pruitt et al. 2007; Karolchik et al. 2008
24	1 if base is in a reported DNase I hypersensitive region in; 0 otherwise	Boyle et al. 2008
25	$\ln(x + 1)$, where x is the number of sequence reads for the interval of the base in the summary file for CTCF	Barski et al. 2007
26	$\ln(x + 1)$, where x is the number of sequence reads for the interval of the base in the summary file for histone variant H2A.Z	Barski et al. 2007
27	$\ln(x + 1)$, where x is the sum of the number of sequence reads for the interval of the base in the summary files for the 20 histone methylation modifications.	Barski et al. 2007
28	The sum over $\ln(x_i + 1)$ for $i = 1, \ldots, 20$, where the x_i's are the number of sequence reads for the interval of the base for the 20 histone methylation modifications.	Barski et al. 2007
29	$\ln(x + 1)$, where x is the number of sequence reads in the interval of the base in the summary file for RNA polymerase II	Barski et al. 2007

Logistic regression

Phase 1 (training phase) : learn the model parameters from training set of positive and negative instances



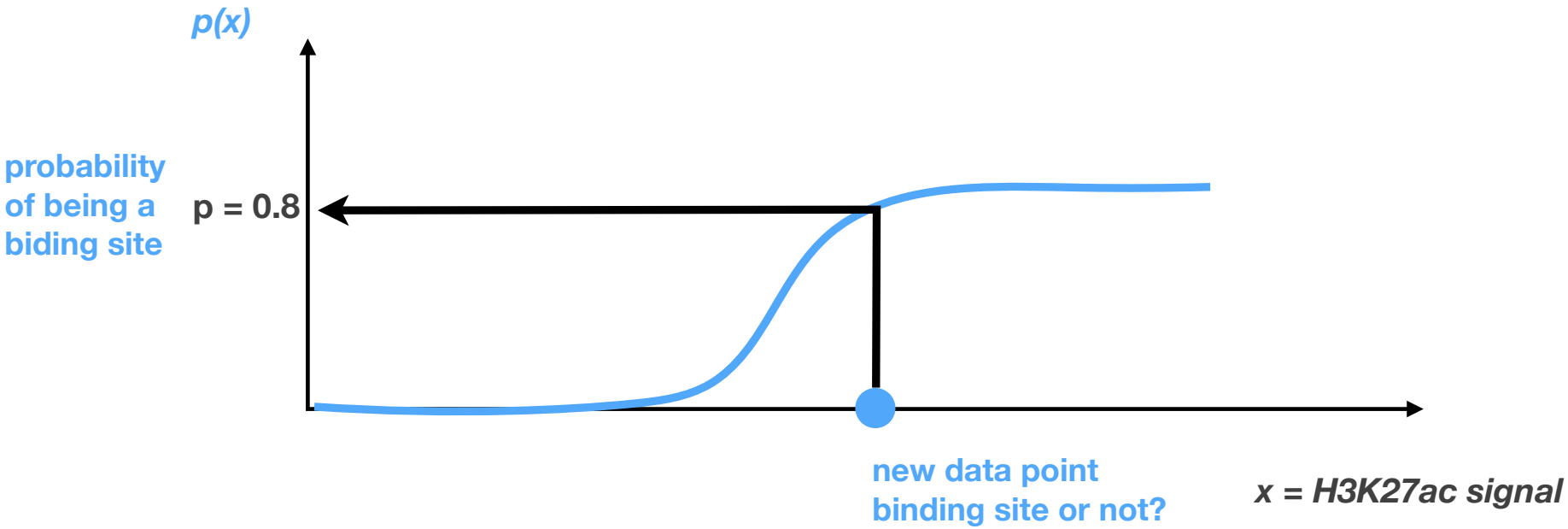


$$z = \beta_0 + \beta_1 x \qquad p(x) = \frac{1}{1 + e^{-z}}$$

p(x) is the **logit** function

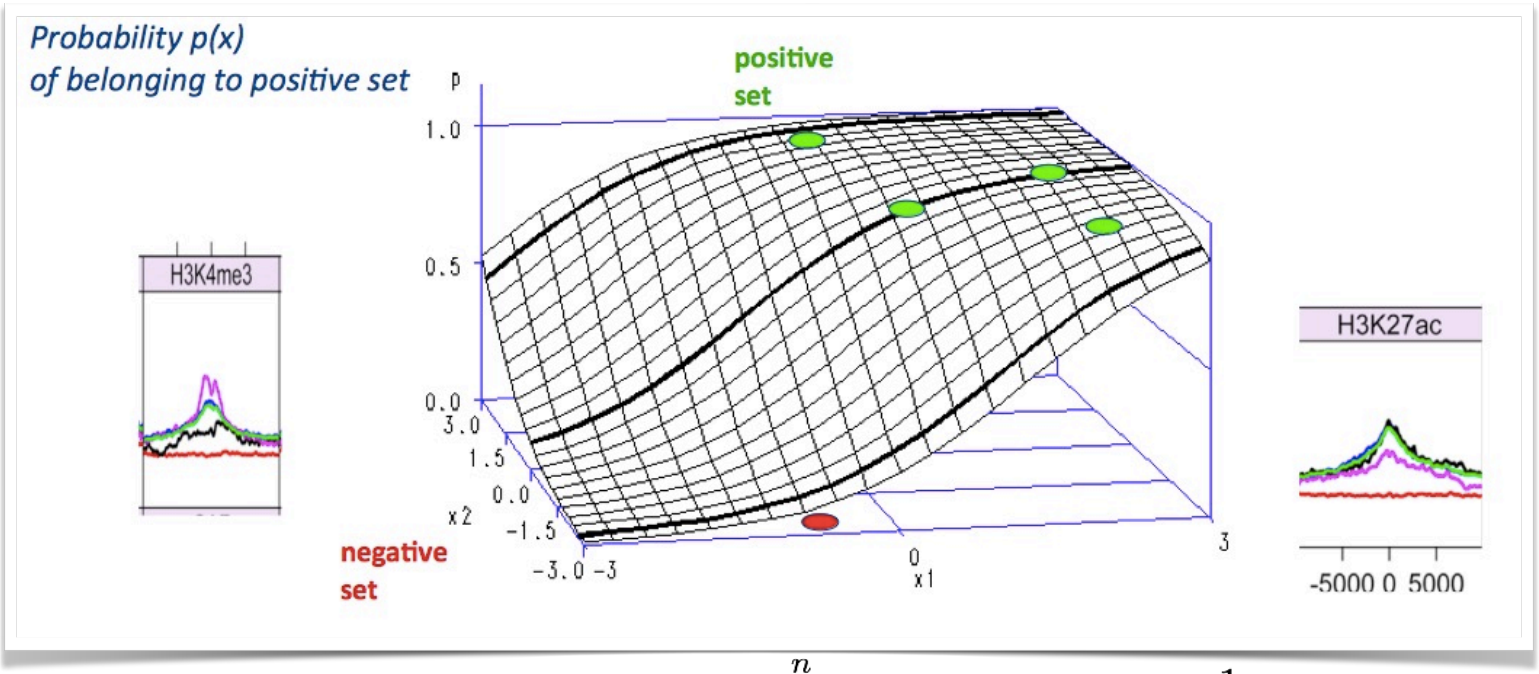
Logistic regression

Phase 2 (prediction) : predict binding probability of a new instance



Logistic regression

Using two variables (H3K4me3 and H3K27ac)



For *n* features :

$$z(x) = \beta_0 + \sum_{i=1}^{n} \beta_i x_i \qquad p(x) = \frac{1}{1 + e^{-z(x)}}$$

For each nucleotide *b*, determine the feature vector $f = \{x_i\}$
→ Nucleotide level GBP : $\boldsymbol{p(b|f)}$ $f = \{x_i\}$

General binding potential

Phase 2 (test phase) :
apply the model
to novel site

PWM matching
score (LLR)

feature vector
at each position *i*

f_1 f_2 f_3 ....

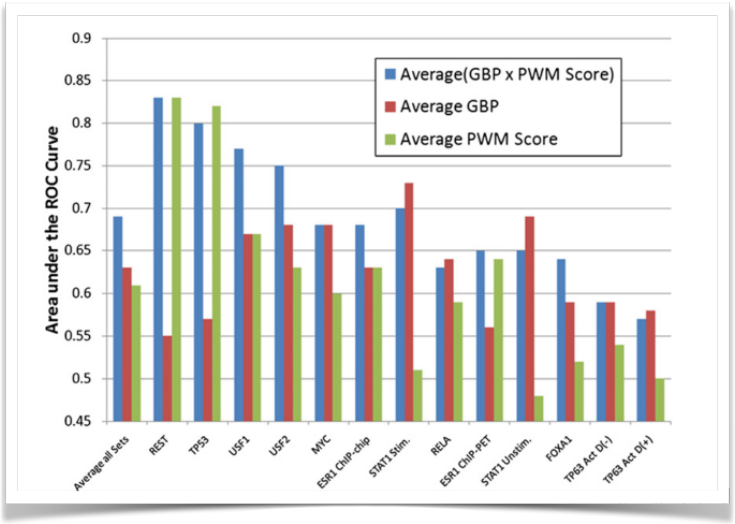
general binding potential
at each position *i*

p_1 p_2 p_3 ....

$$p = \frac{1}{w}\sum_{i=1}^{w} p_i = \frac{1}{w}\sum_{i=1}^{w} p(b_i|f_i)$$

motif only **reg. features only**

$$S = LLR \times \frac{1}{w}\sum_{i=1}^{w} p(b_i|f_i)$$



Integrating general binding potential to PWM score achieves better predictive power for most motifs !