BCH 519
Introduction to Bioinformatics

Motif Discovery

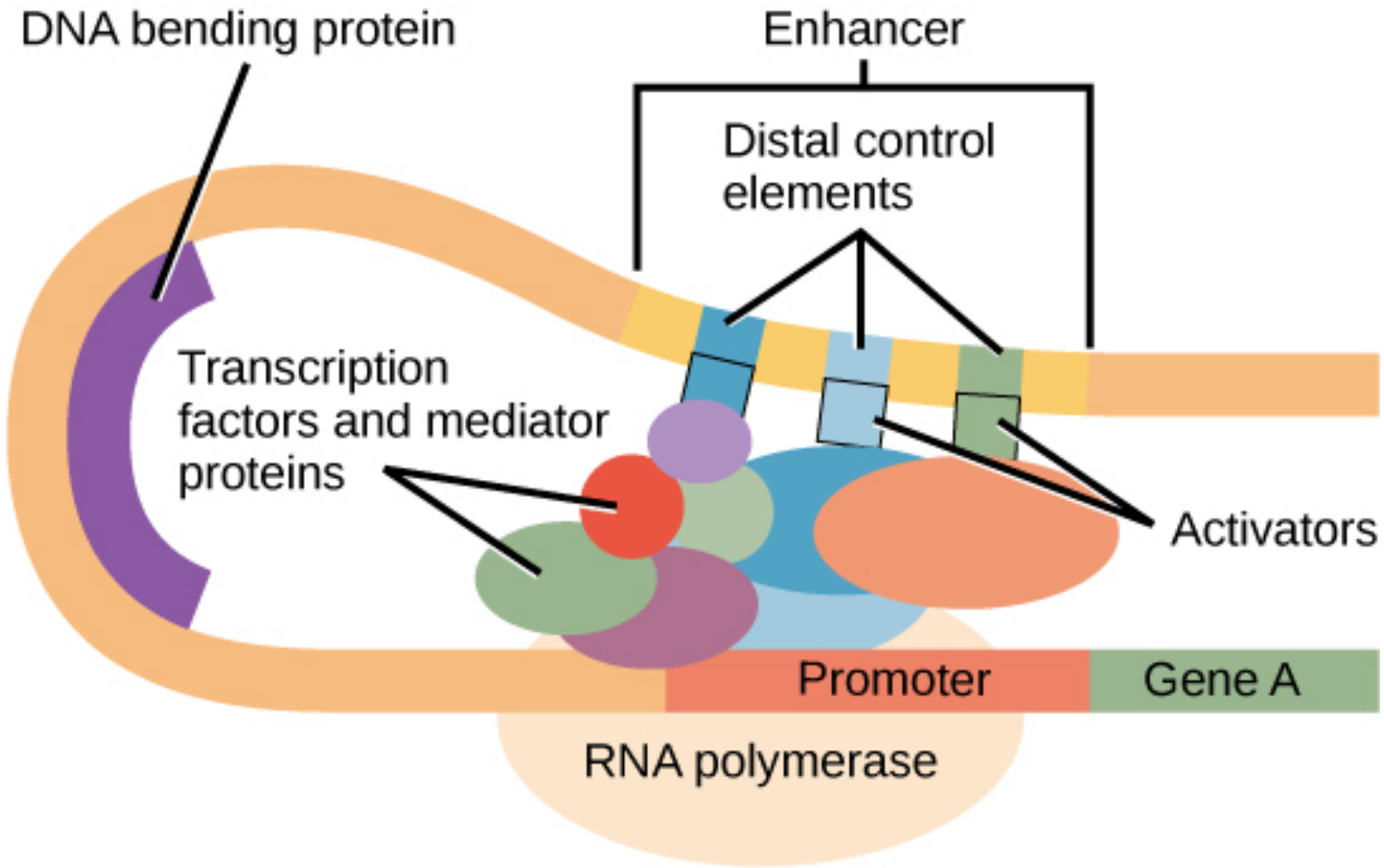
Dr. Tao Liu

March 31, 2015

Outline

- Review traditional **motif** definition by **PWM**
- Learn the experimental way to detect **transcriptional factor binding sites — ChIP-Seq**
- Focus on bioinformatics approaches to **discovery motifs:**
  - **Enumeration**
  - **Greedy search**
  - **Randomized search/Gibbs sampling**
  - **Expectation Maximization**
- Thursday: In-class exercises for homework. Practice some of the programs we discuss in class today.
- Supplemental reading if you want more information:
  - How does DNA sequence motif discovery work? (Patrik D'haeseleer) August 2006, Volume 24, No 8; pp 959
  - What is the expectation maximization algorithm? (Chuong B Do & Serafim Batzoglou) August 2008, Volume 26 No 8; pp 897

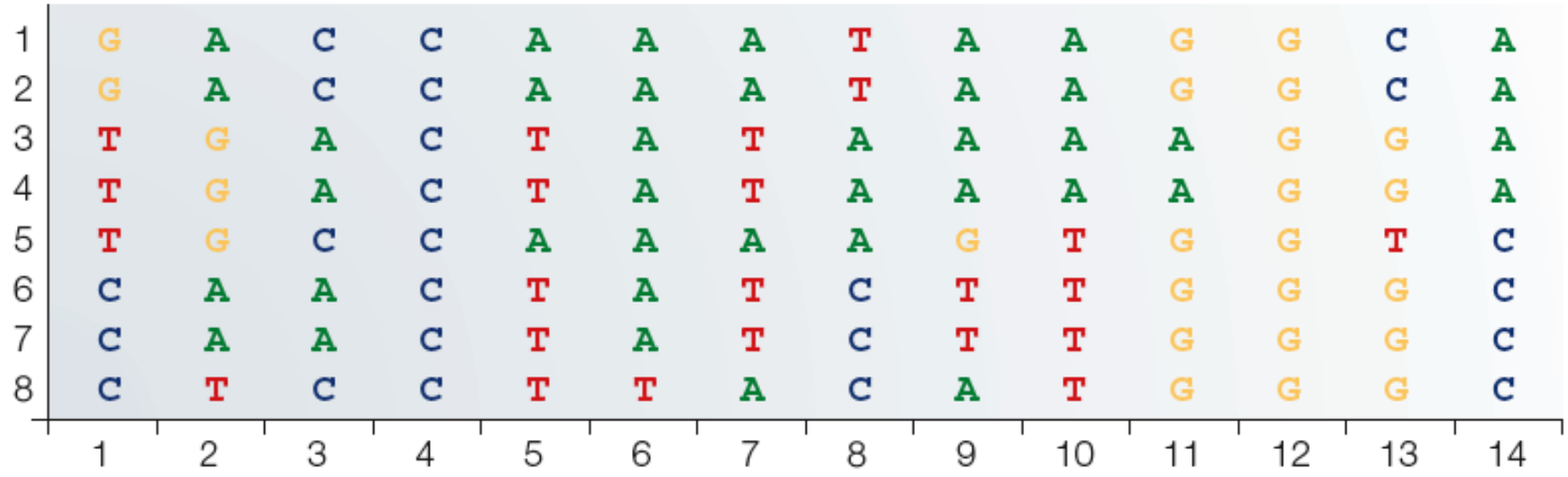
Regulatory elements are bound by transcription factors



- Special DNA sequence is bound at regulatory element.
- Note, some regulatory elements can repress gene expression.

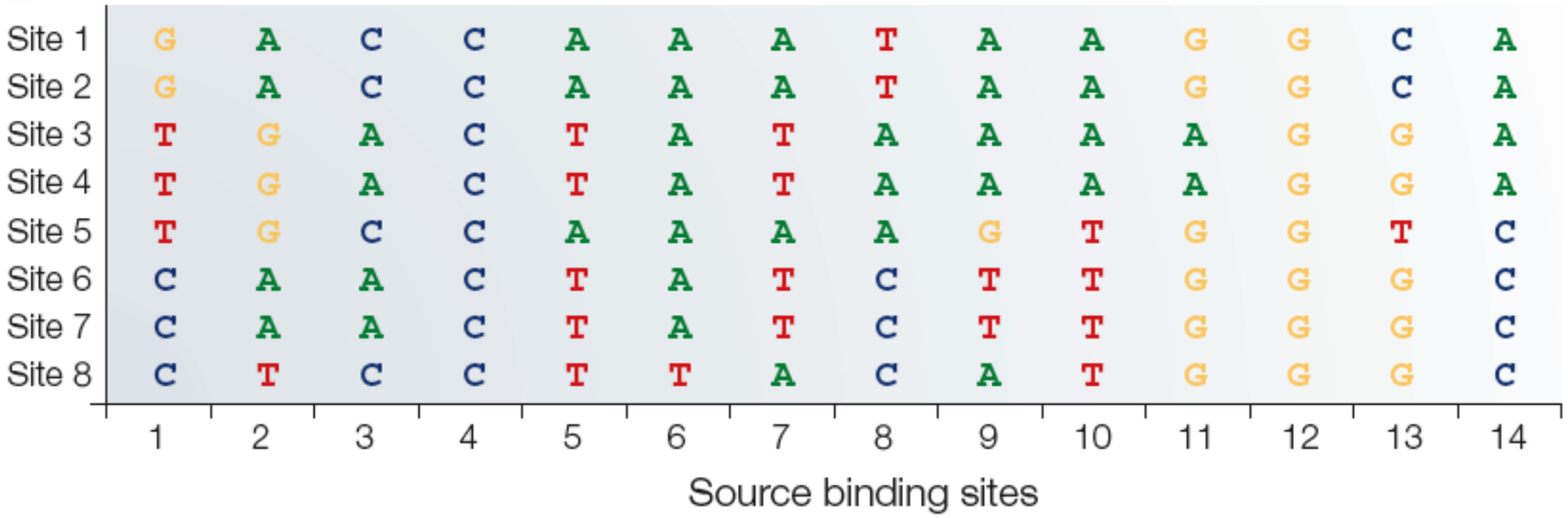
What is **motif?**

- Motif = a set of similar DNA (k-mer strings) sequences.



motif representation

a



b

B	R	M	C	W	A	W	H	R	W	G	G	B	M

Consensus sequence

c Position frequency matrix (PFM)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
A	0	4	4	0	3	7	4	3	5	4	2	0	0	4
C	3	0	4	8	0	0	0	3	0	0	0	0	2	4
G	2	3	0	0	0	0	0	0	1	0	6	8	5	0
T	3	1	0	0	5	1	4	2	2	4	0	0	1	0

d Position weight matrix (PWM)

A	–1.93	0.79	0.79	–1.93	0.45	1.50	0.79	0.45	1.07	0.79	0.00	–1.93	–1.93	0.79
C	0.45	–1.93	0.79	1.68	–1.93	–1.93	–1.93	0.45	–1.93	–1.93	–1.93	–1.93	0.00	0.79
G	0.00	0.45	–1.93	–1.93	–1.93	–1.93	–1.93	–1.93	0.66	–1.93	1.30	1.68	1.07	–1.93
T	0.15	0.66	–1.93	–1.93	1.07	0.66	0.79	0.00	0.00	0.79	–1.93	–1.93	–0.66	–1.93

Sequence Logo

b

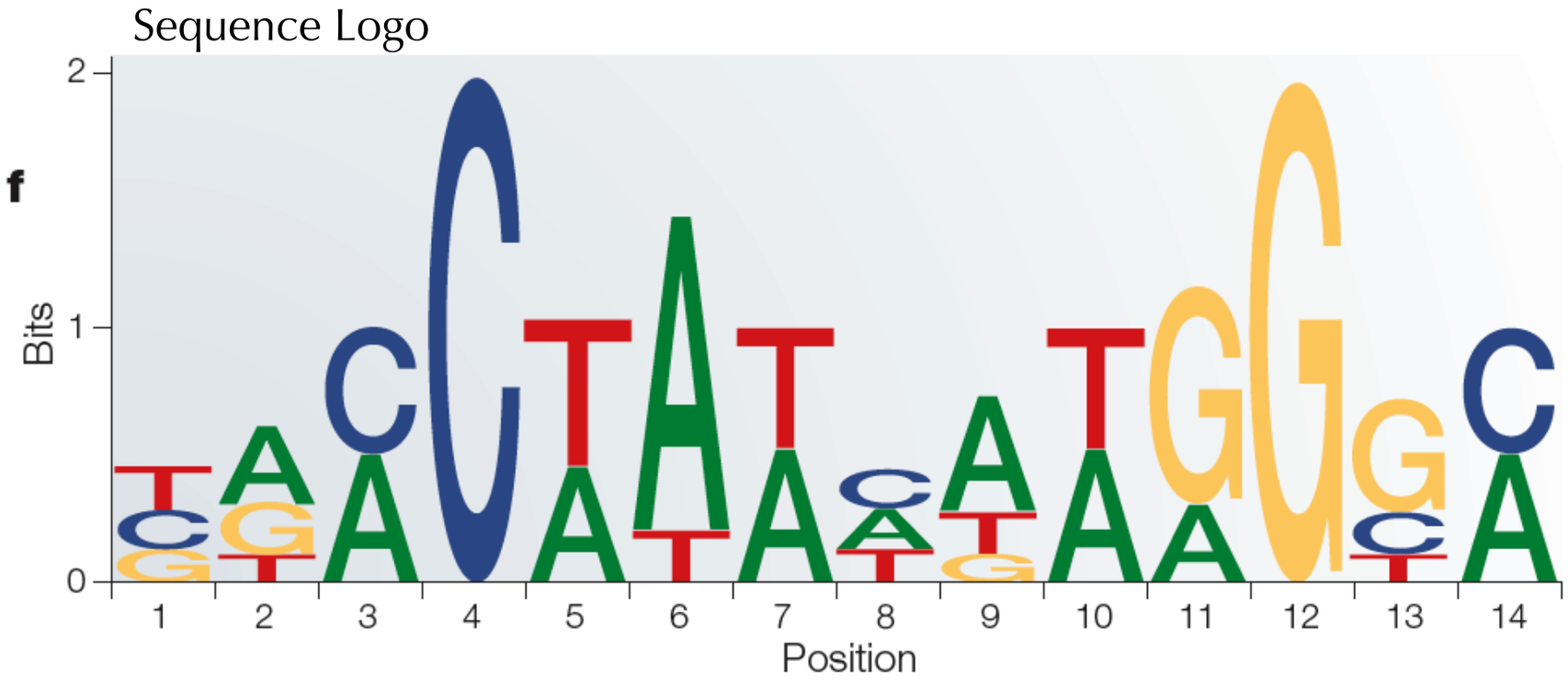
B	R	M	C	W	A	W	H	R	W	G	G	B	M

Consensus sequence

c Position frequency matrix (PFM)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
A	0	4	4	0	3	7	4	3	5	4	2	0	0	4
C	3	0	4	8	0	0	0	3	0	0	0	0	2	4
G	2	3	0	0	0	0	0	0	1	0	6	8	5	0
T	3	1	0	0	5	1	4	2	2	4	0	0	1	0

f



Question!

- How can we find motifs?
  - Given a set of DNA sequences, can we identify the common k-mer?
  - Can we identify motif experimentally?
- How can we use motif to predict binding sites or regulatory elements?

TCTGAGCTTGCGTTATTTTTAGACC

GTTTGACGGGAACCCGACGCCTATA

TTTTAGATTTCCTCAGTCCACTATA

CTTACAATTTCGTTATTTATCTAAT

CAGTAGGAATAGCCACTTTGTTGTA

AAATCCATTAAGGAAAGACGACCGT

Methods based on consensus

- Given a set of DNA sequences (**sites**), e.g. **promoter sequences** of a set of genes clustered by gene expression experiment or the **possible binding sites** from ChIP-Seq.
- Find the most frequent k-mer (or the **consensus**) appeared in all the sites.
- This is a frequent word problem or word counting problem.

TCTGAGCTTGCGTTATTTTTAGACC

GTTTGACGGGAACCCGACGCCTATA

TTTTAGATTTCCTCAGTCCACTATA

CTTACAATTTCGTTATTTATCTAAT

CAGTAGGAATAGCCACTTTGTTGTA

AAATCCATTAAGGAAAGACGACCGT

ChIP-Seq to detect binding sites

- Figure out where a transcription factor bind to DNA through Chromatin Immunoprecipitation coupled with sequencing or **ChIP-Seq**

- Steps:

  - **Cross-link** proximal protein and DNA by UV or formaldehyde

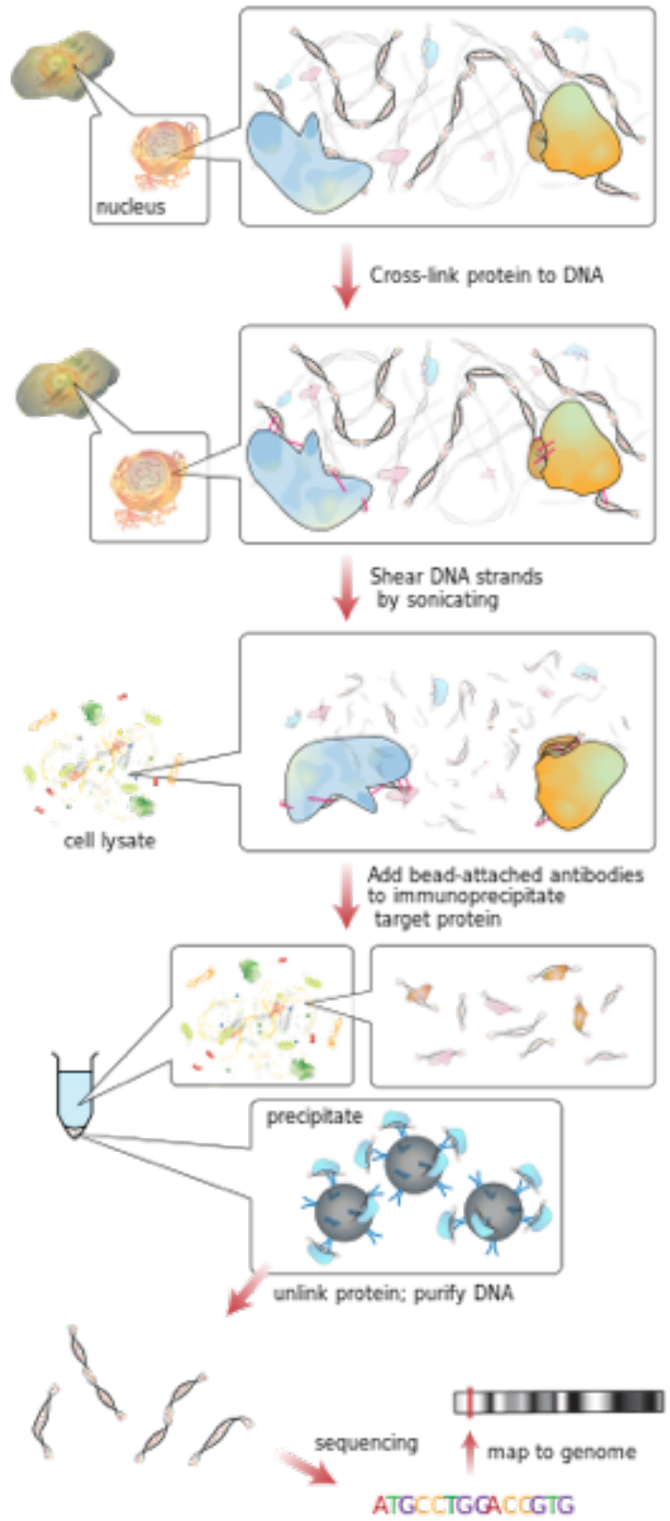
  - Make **DNA fragments** through sonication or enzyme digestion

  - Add **bead-attached antibodies to the target protein**

  - **Precipitate**

  - **Reverse-cross-link** to **purify** DNA fragments attached to the target protein

  - **Sequence** them and **map** them to the genome



Word counting

- Count "**words**" or "**k-mers**". k = 5 or 6 is common as many binding site core sequences are about this size.
- Find the most frequent word or compare the frequency to random background (e.g. randomly generated sequences with the same length of input)

```
example:

seq: CGGAATCACCACTGGATG k=5
CGGAA
 GGAAT
  GAATCA
   AATCAC
    ATCACC
     TCACCA
      CACCAC
       ACCACT
        …
```

Pros and Cons of word counting

- Pros:
  - Enumerate all possible word and will ultimately find the global optimal solution.
  - Can scale well to large genome. An O(n) algorithm.
- Cons:
  - Is it the real biology?
  - Transcription factor binding site or actual DNA motif is probabilistic!

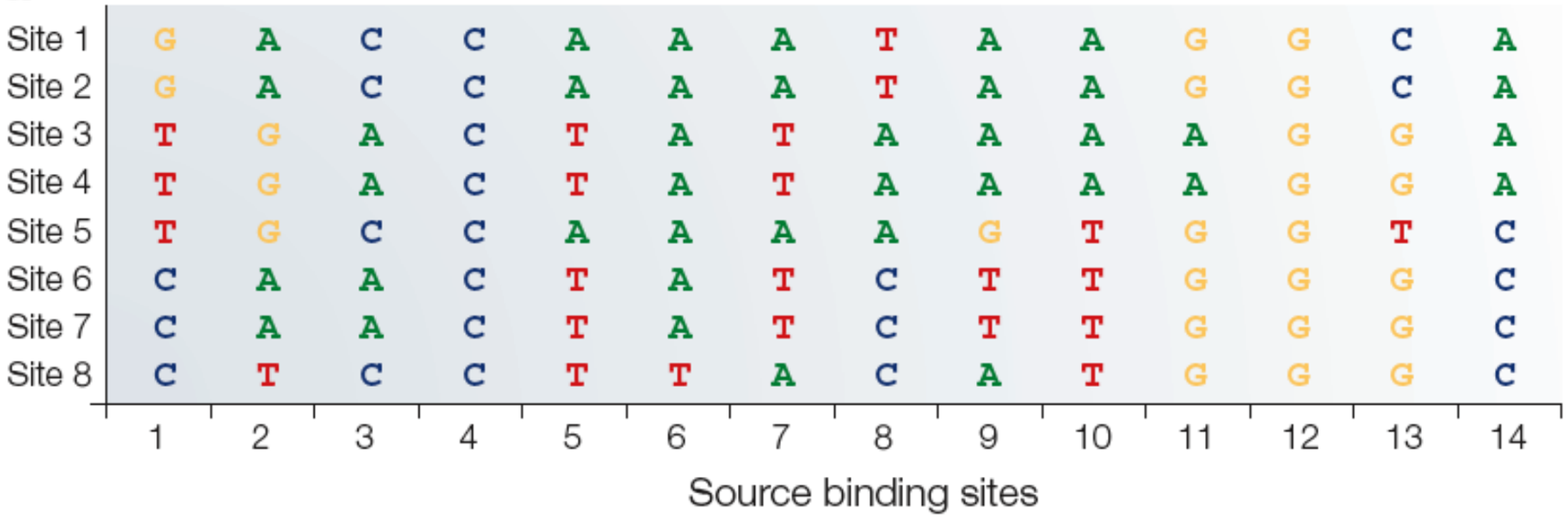
A better version of word counting

- Allow **mismatches…**
- e.g. if we allow 2 mismatches from 5-mer, then we need to
  - first, get all exact 5-mers from the input DNA sequence.
  - next, pick two positions then substitute them with ‘other’ bases.
    - Think how many possible 5-mers? For each observed 5-mers, there are C(5,2) ways to pick two positions, then multiply with 4*4.
  - Finally, find the most frequent 5-mer from the all possible 5-mers set.
- Still, not the perfect way since binding is a **probabilistic** event — certain positions in the motif may have more information content!

```
example:

exact 5-mer:
CGGAA

   |   |

CAGAT
CTGAT
…
CAGAA
…
```

The profile-based algorithm

a



b

B	R	M	C	W	A	W	H	R	W	G	G	B	M

Consensus sequence

c Position frequency matrix (PFM)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
A	0	4	4	0	3	7	4	3	5	4	2	0	0	4
C	3	0	4	8	0	0	0	3	0	0	0	0	2	4
G	2	3	0	0	0	0	0	0	1	0	6	8	5	0
T	3	1	0	0	5	1	4	2	2	4	0	0	1	0

d Position weight matrix (PWM)

A	–1.93	0.79	0.79	–1.93	0.45	1.50	0.79	0.45	1.07	0.79	0.00	–1.93	–1.93	0.79
C	0.45	–1.93	0.79	1.68	–1.93	–1.93	–1.93	0.45	–1.93	–1.93	–1.93	–1.93	0.00	0.79
G	0.00	0.45	–1.93	–1.93	–1.93	–1.93	–1.93	–1.93	0.66	–1.93	1.30	1.68	1.07	–1.93
T	0.15	0.66	–1.93	–1.93	1.07	0.66	0.79	0.00	0.00	0.79	–1.93	–1.93	–0.66	–1.93

How to use the profile

GTGG
AAGC
CCGA
GCGA

	1	2	3	4
A	1	1	0	2
T	0	1	0	0
C	1	2	0	1
G	2	0	4	1

	1	2	3	4
A	1/4	1/4	0	1/2
T	0	1/4	0	0
C	1/4	1/2	0	1/4
G	1/2	0	1	1/4

- When you have a sequence, e.g. CAGT, the probability that it fits the profile is:

  Pr(CAGA|profile) = 1/4 * 1/4 * 1 * 1/2 = 1/32 = 0.03125

- We don't use 'mismatch' as for consensus-based method.

- Q: "CAGT" absolutely never fits the profile?

Pseudocount

- zero-frequency problem: rare events can't have a possibility of zero. So we have to adjust it.
- Laplace's Rule of Succession: simply add ***1*** to all observed frequencies.

GTGG
AAGC
CCGA
GCGA

	1	2	3	4
A	1+1	1+1	0+1	2+1
T	0+1	1+1	0+1	0+1
C	1+1	2+1	0+1	1+1
G	2+1	0+1	4+1	1+1

	1	2	3	4
A	1/4	1/4	1/8	3/8
T	1/8	1/4	1/8	1/8
C	1/4	3/8	1/8	1/4
G	3/8	1/8	5/8	1/4

Pr(CAGA|profile) = 1/4 * 1/4 * 5/8 * 3/8 ~= 0.0146

Pr(CAGT|profile) = 1/4 * 1/4 * 5/8 * 1/8 ~= 0.0049

The motif finding problem is a multiple alignment problem

```
TCTGAGCTTGCGTTATTTTTAGACC
GTTTGACGGGAACCCGACGCCTATA
TTTTAGATTTCCTCAGTCCACTATA
CTTACAATTTCGTTATTTATCTAAT
CAGTAGGAATAGCCACTTTGTTGTA
AAATCCATTAAGGAAAGACGACCGT
```

The Greedy algorithm

- Heuristic process to find local optimum at each step in the hope to find the global optimum (**may fail!**)
- Start from the first DNA sequence of the input, select a k-mer to form a motif profile,
- then for the next sequence, find the **most probable k-mer** that fit the profile. Then add this k-mer to the previous set of k-mers to form a new profile
- Iterate till the last sequence.
- Maximize the score of the final motif by choosing different **starting** point on the first sequence.
    - Entropy or the total information content. For each position, the information content is 2+sum(p*log_2 p). Then add them up!

Randomized method

- The ultimate solution is to iterate all the possible ways to align all the input DNA sequences, then to figure out the best motif profile with the largest motif score. —- it's almost impossible.
- Instead of using greedy algorithm and starting from k-mers of the first sequence, **we generate 'random' motif profile from random k-mers from input sequences**; then in the next step, **use the profile to identify the probable k-mers from the sequences**; then **update the profile**; then **repeat until the motif score can't increase**.

Gibbs sampling

- Similar to, but different with previous randomized method, Gibbs sampling only **replaces a single k-mer at each step**.

  1. Randomly align sequences or randomly choose k-mer for each sequence.

  2. Randomly remove a sequence, then construct motif profile

  3. Use the profile to find the most probable k-mer in the removed sequence, after identified the position, align it back with other sequences.

  4. Go to step 2 and repeat, until given number of iterations or until motif score can't be improved.

- Note: there are many variations of Gibbs sampler, but you just need to know the idea.

TCTGAGCTTGCGTTATTTTTAGACC

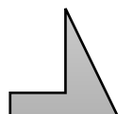
GTTTGACGGGAACCCGACGCCTATA

TTTTAGATTTCCTCAGTCCACTATA

CTTACAATTTCGTTATTTATCTAAT

CAGTAGGAATAGCCACTTTGTTGTA

AAATCCATTAAGGAAAGACGACCGT



TCTGAGC**TTGC**GTTATTTTTAGACC

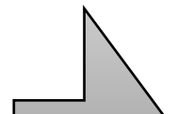
GTTTGACGGG**AACC**CGACGCCTATA

TTTTAGATT**TCCT**CAGTCCACTATA

CTTAC**AATT**TCGTTATTTATCTAAT

CAGTAGGA**ATAG**CCACTTTGTTGTA

AA**ATCC**ATTAAGGAAAGACGACCGT



TCTGAGC**TTGC**GTTATTTTTAGACC

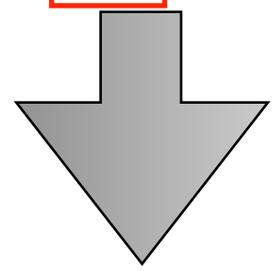
GTTTGACGGG**AACC**CGACGCCTATA

TTTTAGATT**TCCT**CAGTCCACTATA

~~CTTAC**AATT**TCGTTATTTATCTAAT~~

CAGTAGGA**ATAG**CCACTTTGTTGTA

AA**ATCC**ATTAAGGAAAGACGACCGT



TTGC

AACC

TCCT

ATAG

ATCC

	1	2	3	4
A	3+1	1+1	1+1	0+1
T	2+1	3+1	0+1	1+1
C	0+1	1+1	3+1	3+1
G	0+1	0+1	1+1	1+1

	1	2	3	4
A	4/9	2/9	2/9	1/9
T	1/3	4/9	1/9	2/9
C	1/9	2/9	4/9	4/9
G	1/9	1/9	1/9	2/9

TCTGAGC**TTGC**GTTATTTTTAGACC

GTTTGACGGG**AACC**CGACGCCTATA

TTTTAGATT**TCCT**CAGTCCACTATA

CAGTAGGA**ATAG**CCACTTTGTTGTA

AA**ATCC**ATTAAGGAAAGACGACCGT

	1	2	3	4
A	4/9	2/9	2/9	1/9
T	1/3	4/9	1/9	2/9
C	1/9	2/9	4/9	4/9
G	1/9	1/9	1/9	2/9

CTTACAATTTCGTTATTT**ATCT**AAT

Repeat

TCTGAGC**TTGC**GTTATTTTTAGACC

GTTTGACGGG**AACC**CGACGCCTATA

TTTTAGATT**TCCT**CAGTCCACTATA

CAGTAGGA**ATAG**CCACTTTGTTGTA

AA**ATCC**ATTAAGGAAAGACGACCGT

CTTACAATTTCGTTATTT**ATCT**AAT

Pros and Cons

- Pros:
  - Similar to randomized approach, won't be easily trapped in local optimal
  - Guaranteed to converge
- Cons:
  - Similar to randomized approach, can be very slow
  - Tricky to define a criterion to stop the iterations.

EM algorithm

- EM = **Expectation Maximization**
- A **probabilistic method** used when there are **incomplete** data that prevent parameters from being properly estimated.
    - Remember our dishonest gambler? We used a **HMM** to figure out when he was cheating. BUT—we knew the probability that he would switch dice, and the probabilities for the result of each roll.
    - Expectation Maximization (EM) can be used when some of the parameters are **unknown.**
- The EM algorithm consists of two steps, which are repeated consecutively.
    - In step 1, the expectation (“E”) step, a guess is made about the missing parameters and the resulting probabilities of the results are calculated.
    - In Step 2, the maximization (“M”) step, the missing parameters are re-estimated based on the probabilities determined in the E step.
    - These steps are reiterated until ultimately the algorithm converges.

EM for motif finding

- EM can be used for motif finding to identify conserved areas in unaligned DNA and proteins
- Make a guess as to the motif site (with length k) in a single sequence and calculate the weight matrix for the site (using all of the sequences) (the **initialization** step)
- Next, for each k-mer in each of the sequences, calculate the likelihood that it is part of the motif, rather than background (the **expectation** step)
- Then take a weighted average across these likelihoods and use this to refine the motif model (the **maximization** step)
- Repeat until convergence

Expectation step

- Step 0: Guess, then construct the profile (probability matrix)
- Step 1: Expectation

  Use these probabilities to find the likely position of the motif in <u>each</u> sequence
- Note: different with previous matrix, now we need to add a 'background' probability column for each base. Because we want to calculate the likelihood that the motif is found at a particular position.

Initial guess

```
OOOOOOOOXXXXOOOOOOOO
OOOOOOOOXXXXOOOOOOOO
o o o o o o o o o o o o o o o o o o o o o o o o
OOOOOOOOXXXXOOOOOOOO
OOOOOOOOXXXXOOOOOOOO
         IIII

IIIIIIII     IIIIIII
```

background **background**

Bases	Background	Site column 1	Site column 2	…
G	0.27	0.4	0.1	…
C	0.25	0.4	0.1	…
A	0.25	0.2	0.1	…
T	0.23	0.2	0.7	…
Total	1.00	1.00	1.00	…

Expectation step (cont.)

- **For each position on each sequence**, calculate the likelihood that a motif can be found there. Note: multiply with background as well.
- Calculate the **weights** for each position on each sequence as P/total(P) for all positions of that sequence.

Bases	Background	Site column 1	Site column 2	…
G	0.27	0.4	0.1	…
C	0.25	0.4	0.1	…
A	0.25	0.2	0.1	…
T	0.23	0.2	0.7	…
Total	1.00	1.00	1.00	…

AACGTGCT

P = 0.25*0.25*0.4*0.1*0.23*0.27*0.25*0.23

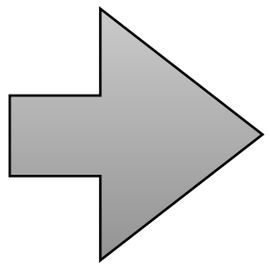
Then sum(P) over all positions should be normalized to 1.

Maximization step

- Use the P to update motif profile.
- Use the probabilities calculated in E Step to weight (update) the original table of probabilities

k=2

A	A	C	G	T	G	A
0.1	0.5	0.1	0.1	0.1	0.1	
C	C	T	T	A	A	A
0.4	0.1	0.1	0.1	0.2	0.1	
G	G	C	C	T	A	G
0.2	0.1	0.1	0.1	0.1	0.4	



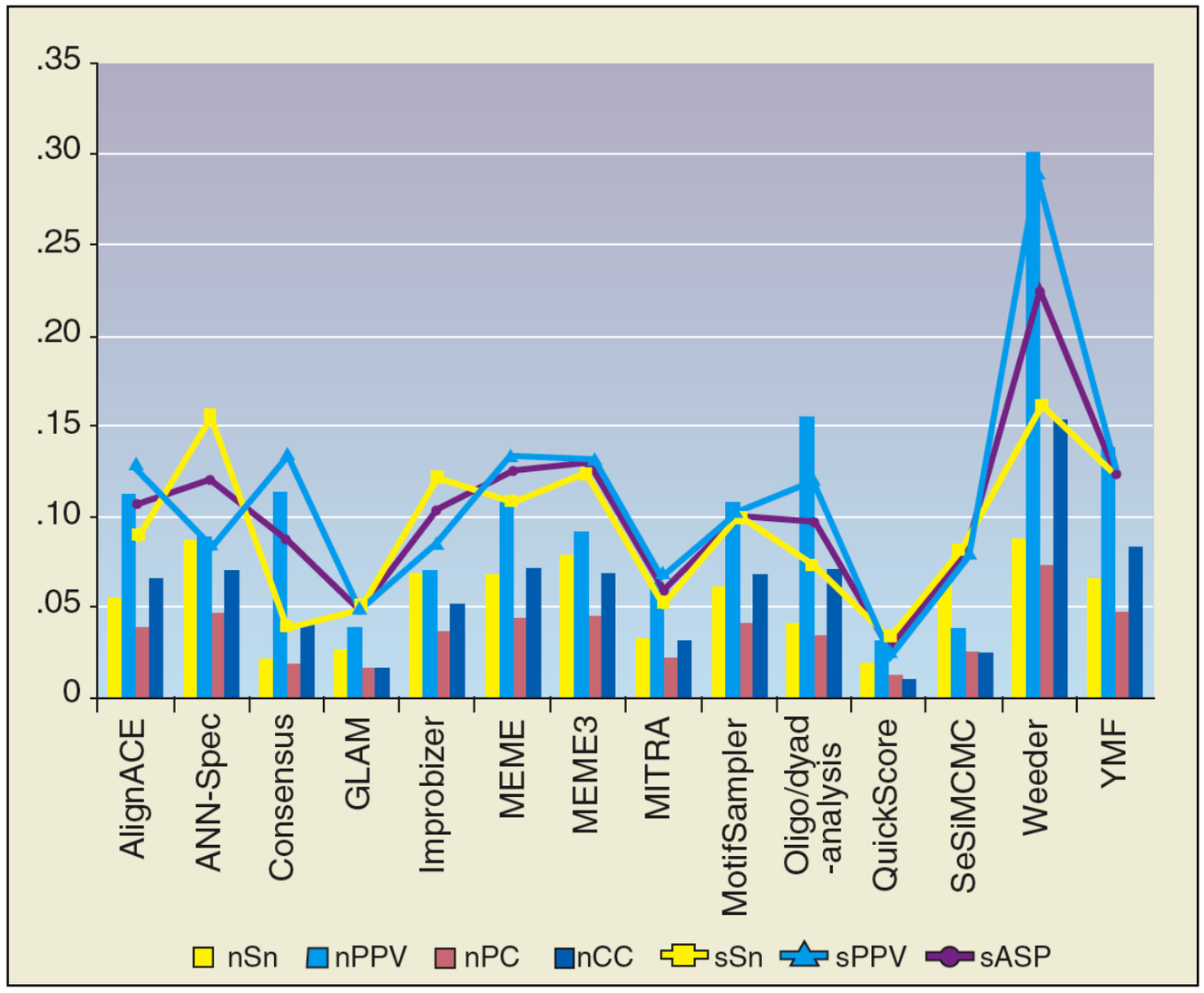
	B	1	2
A	1.1	1.3	0.6
T	2.2	0.4	0.4
C	1.1	0.8	1.1
G	1.7	0.5	0.8

Pros and Cons

- Pros:
  - Guaranteed to converge
  - And converge fast
- Cons:
  - May find local optimal
  - Sensitive to the initial guess
- The popular one is MEME which we will practice on Thursday.

Summary

- Should try different methods to reach consistent results
- How do we decide if an identified motif is significant?
  - We can use a number of measures, including:
    - Motif score — information content
    - Specificity in the genome
    - Positional bias
      e.g., with respect to transcription start site
    - Palindromicity
    - Comparison to known motifs



Tompa, M., et al. 2005. *Nat Biotechnol* **23:** 137-144.