# BC205: Algorithms for Bioinformatics. III. Analyzing Biological Motifs

Christoforos Nikolaou

March 22nd, 2017

# In previous chapters

- We saw the limitations of composition approaches
  - They can give us rough estimates of sequence properties
  - They are not precise in locating elements such as HGT, OriC etc

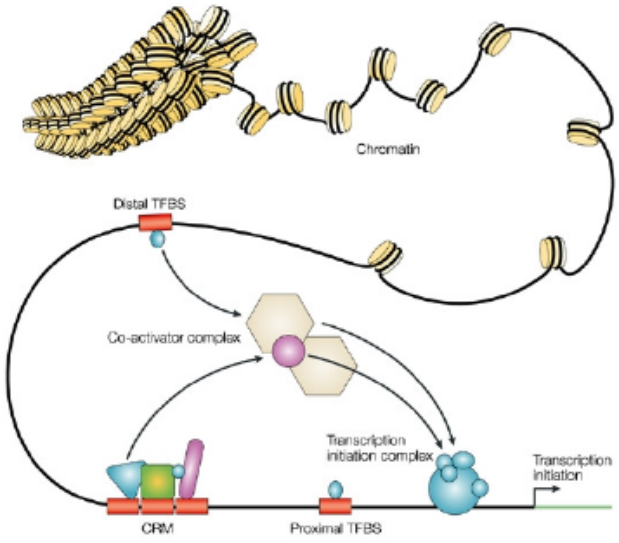
Lucky for us many problems in biology are related with much more specific signals

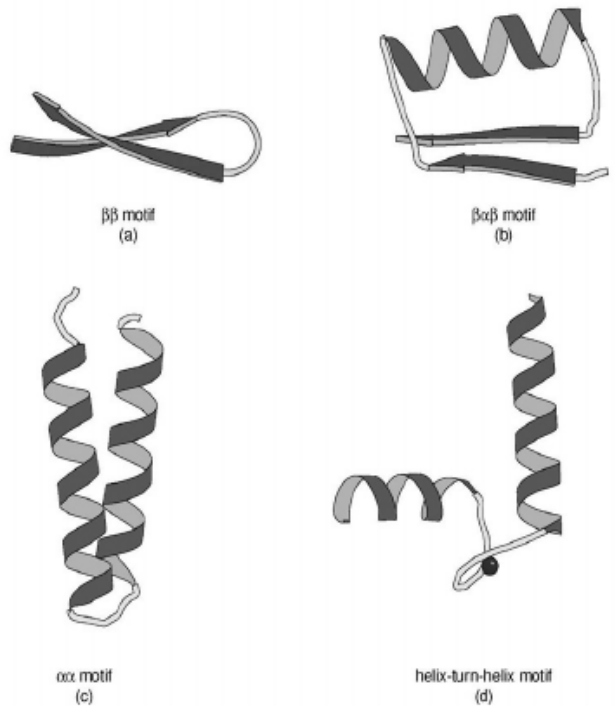
# Motifs

- Most (in not all) forms of messages employ "motifs" for:
  - **repat/emphasis** : e.g. "I have a dream", MLK, "March on Jobs and Freedom Speech" "Crawth the raven, Nevermore", Edgar Allan Poe
  - **coherence**: e.g. "Who controls the past, controls the future", George Orwell "1984"
  - **subtextualization**: e.g. "Fair is foul and foul is fair", William Shakespeare, "Macbeth" (and almost all of "Pulp Fiction")
  - **internal reference**: e.g. all "leitmotivs" in Operas

# Motifs in Biology

- Genome: Codons, Transcription factor binding sites, CpG islands,
- All areas of the genome that interact with proteins in sequence-dependent manner
- Protein: Patterns of aminoacids that are related to particular function, modules, domains etc



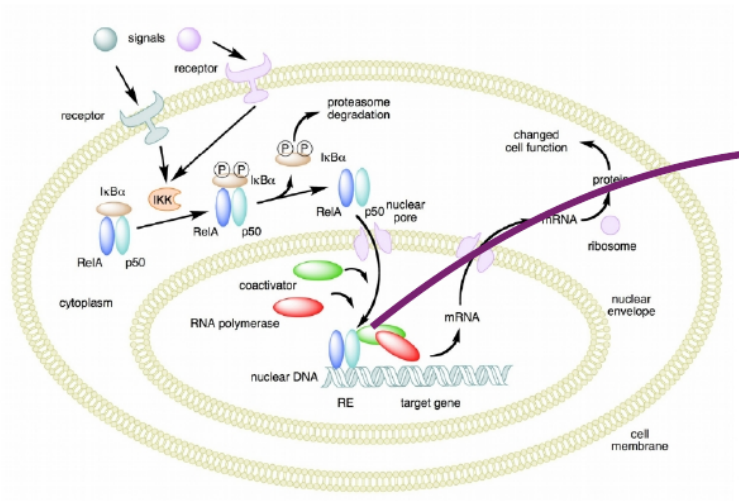



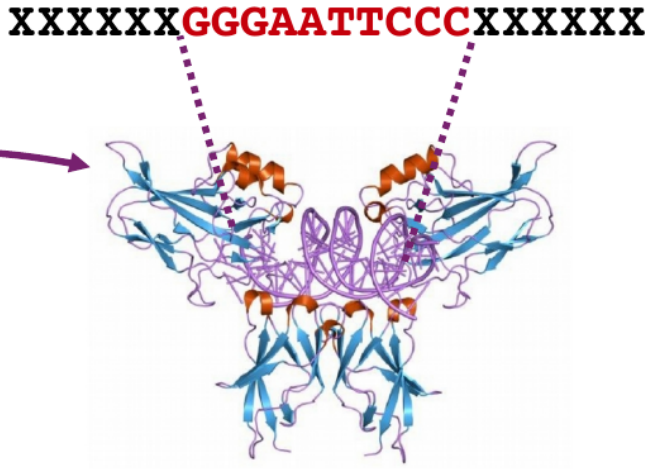
# Motif-related biological problems

- How to define a motif?
- How to locate a *known* motif?
- How to evaluate the motif?
- How to discover *unknown* motifs in a sequence?

# Problem #1: What is a motif?

- How do we define a biological motif?
- What do we need as input?
- What will the output be?



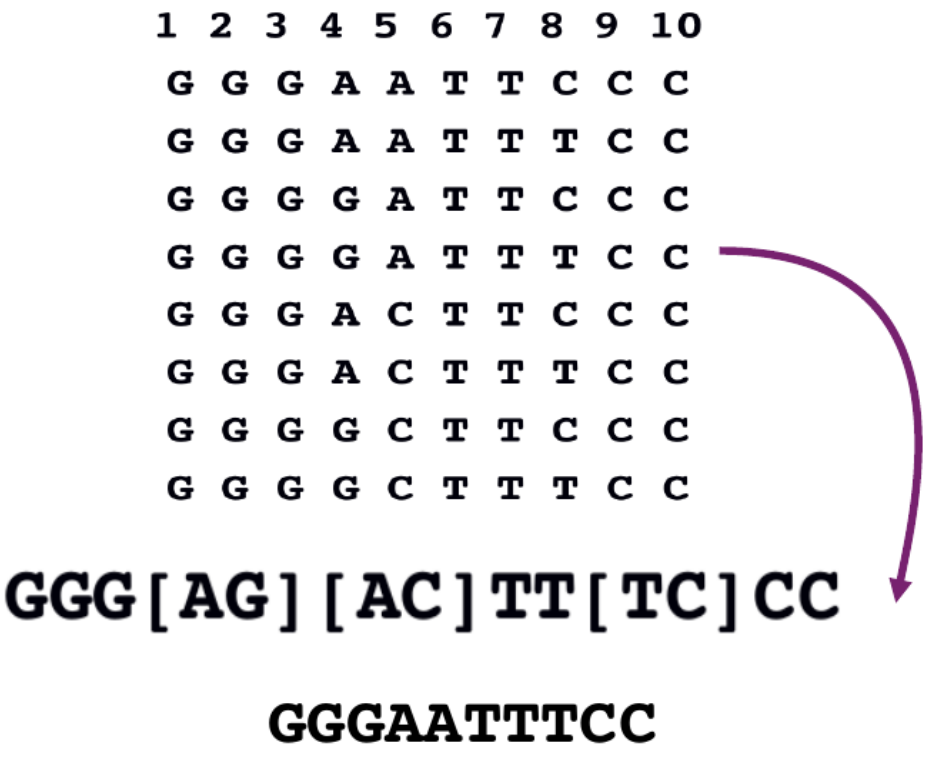
# Problem #1: Input

- Given a set of oligonucleotides that fulfil a certain function:
  - Sequence have variability, so we should:
  - Define the motif as a coherent entity that describes all *instances* of the sequence

```
G G G A A T T C C C
G G G A A T T T C C
G G G G A T T C C C
G G G G A T T T C C
G G G A C T T C C C
G G G A C T T T C C
G G G G C T T C C C
G G G G C T T T C C
```

# Problem #1: Consensus Sequence

- We may define as "consensus" either the most common sequence variant or
- A set of rules (in the form of a "regular expression") that describes all instances of the motif



# Problem #1: The problem with the Consensus approach

- As the instances we collect grows bigger, the variants increase
- Regular Expressions don't work

| | | | | | | |
|---|---|---|---|---|---|---|
| GGGGCATTCC | GGGATATCCC | GGGAATTCCC | GGGAATGTCC | GGGATATTTC | GGGGCCTCCC | GGGAATTTCC | GGGACTGCCC |
| GGGAAATTCC | GGGAAATCCC | GGGAATTCCC | GGGACTTACC | GGGGATTTCC | GGGAATTTCC | GGGACATTCC | GGGAATTTCC |
| GGAAATTTCC | GGGAATTCCC | GGGGATTTCC | GGGGTTTCAC | GGGAAGGTCC | GGGGCTTCCC | GGGGCTTTCC | GGGAAATTCC |
| GGGGCTTTCC | GGGACTTTCC | GGGACATTCC | GGGAATTTCC | GGGACATTCT | GGGACAGCCC | GGGGCTTTAC | GGGACTTCCC |
| GGGAATTCAC | GGGAAATCCC | GGAGCTTTCC | GGGACTTTCC | GGGAAACCCC | GGGGCTTCCC | GGGAATTTCC | GGGAAATTCC |
| GGGACTTCCC | GGGAATTTCT | GGGAATTCCC | GGGACTTCCC | GGGACTTTCC | GGGGATTTCC | GGGACATCCC | GGGAAATCCC |
| GGGATGTTCC | GGGGTCTCCC | GGGACTGTCC | GGGAATTCCC | GGGACTTTAC | GGGAATTTCC | GGGACTTTCC | GGGGCGTCCC |
| GGGGTTTCCC | GGGAATTTCC | GGGAATTTCC | GGGGATTTCC | GGGAATGCCC | GGGGATTTCC | GGGAATTTCC | GGGATTTTCC |
| GGGGAATTCC | GGGACTTCCC | GGGATTTTCC | GGGAAGTCCC | GGGAAATTCC | GGGAATTTCC | GGGAATTTAC | GGGAAATTCC |
| GGGGGTTTAC | GGGACTTTCC | GGGAATTTCC | GGGAATTTCC | GGGACATCCC | GGGAATTCAC | GGGACTTCCC | GGGACTTTCC |
| GGGAATTTCC | GGGACTTTCC | GGGGACTTCC | GGGACTTTAC | GGGACTTTCC | GGGATACTCC | GGGGATGTAC | GGGATATCCC |
| GGGAATTCCC | GGGACTTCCC | GGGACTTCAC | GGGGTTACCC | GGGAATCTCC | GGGAATTTCC | GGGACATCTC | GGAAATTCCC |
| GGGAAACTCT | GGGGTTTCCC | GGGATTTTCC | GGGGCGTTCC | GGGAAACTCT | GGGGTTTCCC | GGGATTTTCC | GGGGCGTTCC |

***Πίνακας 3.1**: 104 σημεία πρόσδεσης του NF-κB από το γονιδίωμα του ποντικιού (Mus musculus)*

**GG[AG][AG][AGCT][AGCT][AGCT][ACT][ACT][CT]**

How do we describe all the variants without losing in specificity?

# Algorithmic Interlude: Edit Distances

- We need a measure to describe differences between variants
- E.g.:
  - How different from the most common instance GGGAATTCCC is AAAAATTCCC?
  - How different from the most common instance GGGAATTCCC is GGGTTTACCC?

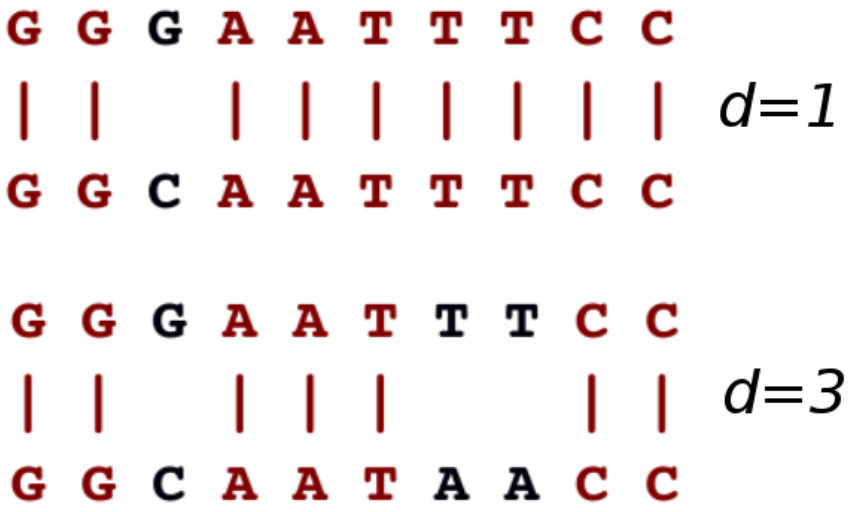
# Edit Distances

- Levenshtein Distance. Allows Insertions/Deletions/Substitutions
- **Hamming Distance. Allows substitutions only**
- Longest Common Subsequence (LCS). Allows Insertions/Deletions only
- Damerau-Levenshtein Distance. Allows Insertions/Deletions/Substitutions *and* Transpositions
- Jaro Distance. Allows Transpositions only.

The ***Hamming Distance*** is the one that best fits our goal for now, but we'll revisit some of the above in the future.

# Hamming Distance in motifs

Simply calculate the number of nucleotides that need to be changed from $S_1$ to become $S_2$, assuming two sequences of equal size



# Problem #1: Calculate the Hamming Distance of two strings

```
seq1=str(raw_input("Give the 1st sequence to compare: "))
seq2=str(raw_input("Give the 1st sequence to compare: "))

distance=0

if len(seq1) == len(seq2):
    for i in range(len(seq1)-1):
        if seq1[i]!=seq2[i]:
            distance=distance+1
    print "Hamming Distance is equal to: ",distance

if len(seq1) != len(seq2):
    print "Cannot Calculate Hamming Distance"
```

# Problem #1: The problem with Hamming/Edit Distances

- Assuming the motif is $GGG[AG][AG]TT[TC]CC$ how good a motif is:

1. AAAAATTCCC?
2. compared to GGGTTTACCC?

- We actually have two problems:
  - We cannot compare with a "consensus" regexp
  - Even if we did compare with most common sequence variant, the results would be misleading

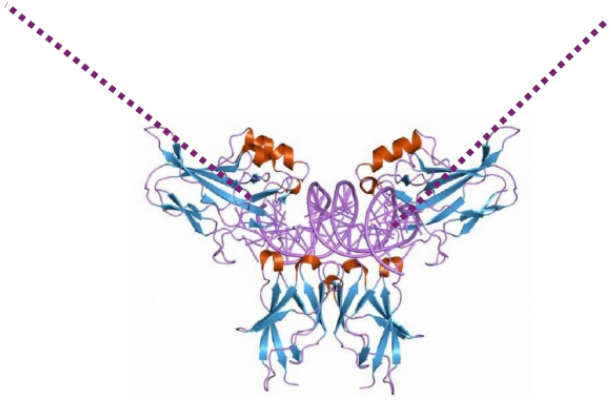
Why?

# Not all positions in the motif are equal

- Comparison of instances with the Hamming Distance disregards the local tendencies in the motifs position
- We need to account for the fact that some positions are more "fixed" and other more "flexible"
- We need a *probabilistic* description of the motif

```
AAAAATTCCC => d=3
GGGGGTTTCC => d=3
GGGAACCCCC => d=2
```

```
XXXXXXGGGAATTCCCXXXXXX
```



# Problem #1: Defining a motif with PWM

- Given a number of sequence of equal size
- Calculate the probabilities of occurrence of *each nucleotide* for *each position* in the sequences
- Create a table of the probabilities

```
GGGGCATTCC  GGGATATCCC  GGGAATTCCC  GGGAATGTCC  GGGATATTTC GGGGCCTCCC GGGAATTTCC GGGACTGCCC
GGGAAATTCC  GGGAAATCCC  GGGAATTCCC  GGGACTTACC  GGGGATTTCC GGGAATTTCC GGGACATTCC GGGAATTTCC
GGAAATTTCC  GGGAATTCCC  GGGGATTTCC  GGGGTTTCAC  GGGAAGGTCC GGGGCTTCCC GGGGCTTTCC GGGAAATTCC
GGGGCTTTCC  GGGACTTTCC  GGGACATTCC  GGGAATTTCC  GGGACATTCT GGGACAGCCC GGGGCTTTAC GGGACTTCCC
GGGAATTCAC  GGGAAATCCC  GGAGCTTTCC  GGGACTTTCC  GGGAAACCCC GGGGCTTCCC GGGAATTTCC GGGAAATTCC
GGGACTTCCC  GGGAATTTCT  GGGAATTCCC  GGGACTTCCC  GGGACTTTCC GGGGATTTCC GGGACATCCC GGGAAATCCC
GGGATGTTCC  GGGGTCTCCC  GGGACTGTCC  GGGAATTCCC  GGGACTTTAC GGGAATTTCC GGGACTTTCC GGGGCGTCCC
GGGGTTTCCC  GGGAATTTCC  GGGAATTTCC  GGGGATTTCC  GGGAATGCCC GGGGATTTCC GGGAATTTCC GGGATTTTCC
GGGGAATTCC  GGGACTTCCC  GGGATTTTCC  GGGAAGTCCC  GGGAAATTCC GGGAATTTCC GGGAATTTAC GGGAAATTCC
GGGGGTTTAC  GGGACTTTCC  GGGAATTTCC  GGGAATTTCC  GGGACATCCC GGGAATTCAC GGGACTTCCC GGGACTTTCC
GGGAATTTCC  GGGACTTTCC  GGGGACTTCC  GGGACTTTAC  GGGACTTTCC GGGATACTCC GGGGATGTAC GGGATATCCC
GGGAATTCCC  GGGACTTCCC  GGGACTTCAC  GGGGTTACCC  GGGAATCTCC GGGAATTTCC GGGACATCTC GGAAATTCCC
GGGAAACTCT  GGGGTTTCCC  GGGATTTTCC  GGGGCGTTCC  GGGAAACTCT GGGGTTTCCC GGGATTTTCC GGGGCGTTCC
```

| Νουκλεοτίδιο | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| A | 0.00 | 0.00 | 0.03 | 0.76 | 0.49 | 0.23 | 0.01 | 0.01 | 0.10 | 0.00 |
| C | 0.00 | 0.00 | 0.00 | 0.00 | 0.37 | 0.03 | 0.04 | 0.38 | 0.88 | 0.97 |
| G | 1.00 | 1.00 | 0.97 | 0.24 | 0.01 | 0.05 | 0.07 | 0.00 | 0.00 | 0.00 |
| T | 0.00 | 0.00 | 0.00 | 0.00 | 0.13 | 0.69 | 0.88 | 0.61 | 0.02 | 0.03 |

# Try it yourselves

- Get the sequences of the GATA binding protein from here (`https://tinyurl.com/ms6rm24`)
- Write a program that will create a PWM

# Calculating a PWM from a sequence dataset

```
"your code here"
```

# Problem #1: PSSM: PWMs without the background

- PWM are sensititve to background nucleotide composition
- This means that sequences rich is some nucleotides will tend to "load" motifs with those nucleotides
- By now we know how to control for that by dividing over a background model
- PSSM (Position-Specific Scoring Matrices) are motifs derived like this:

Μοτίβο NF-κB ***(P)***

| Νουκλεοτίδιο | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| A | 0.00 | 0.00 | 0.03 | 0.76 | 0.49 | 0.23 | 0.01 | 0.01 | 0.10 | 0.00 |
| C | 0.00 | 0.00 | 0.00 | 0.00 | 0.37 | 0.03 | 0.04 | 0.38 | 0.88 | 0.97 |
| G | 1.00 | 1.00 | 0.97 | 0.24 | 0.01 | 0.05 | 0.07 | 0.00 | 0.00 | 0.00 |
| T | 0.00 | 0.00 | 0.00 | 0.00 | 0.13 | 0.69 | 0.88 | 0.61 | 0.02 | 0.03 |

Πίνακας Υποβάθρου ***(Q)***

| Νουκλεοτίδιο | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| A | 0.16 | 0.16 | 0.16 | 0.16 | 0.16 | 0.16 | 0.16 | 0.16 | 0.16 | 0.16 |
| C | 0.27 | 0.27 | 0.27 | 0.27 | 0.27 | 0.27 | 0.27 | 0.27 | 0.27 | 0.27 |
| G | 0.33 | 0.33 | 0.33 | 0.33 | 0.33 | 0.33 | 0.33 | 0.33 | 0.33 | 0.33 |
| T | 0.23 | 0.23 | 0.23 | 0.23 | 0.23 | 0.23 | 0.23 | 0.23 | 0.23 | 0.23 |

$$R = \log_2\left(P_{i,j} / Q_{i,j}\right)$$

| Νουκλεοτίδιο | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| A | -7.3 | -7.3 | -2.4 | 2.2 | 1.6 | 0.5 | -3.9 | -3.9 | -0.7 | -7.3 |
| C | -8.1 | -8.1 | -8.1 | -8.1 | 0.5 | -3.1 | -2.7 | 0.5 | 1.7 | 1.8 |
| G | 1.6 | 1.6 | 1.6 | -0.5 | -4.9 | -2.7 | -2.2 | -8.4 | -8.4 | -8.4 |
| T | -7.8 | -7.8 | -7.8 | -7.8 | -0.8 | 1.6 | 1.9 | 1.4 | -3.5 | -2.9 |

***Position-Specific Scoring Matrix, PSSM***

# Problem #2: Finding a motif in a sequence with PWM/PSSM

- Calculate the PWM scores of AAAAATTCCC and GGGTTTACCC. How does this compare with their Hamming Distances
- Now think of how you can use the PWM to scan a longer sequence

# Problem #2: Finding a motif: PSSM vs PWM

1. Given a PWM, can we calculate the probability of a given pattern to match the motif?
2. What should we be careful about the probability calculations? [Hint: Products are sensitive to 0s]
   - 2.1 We should be careful to add "pseudocounts" to PWMs or
   - 2.2 Work with sums instead

# Problem #2: PSSM search

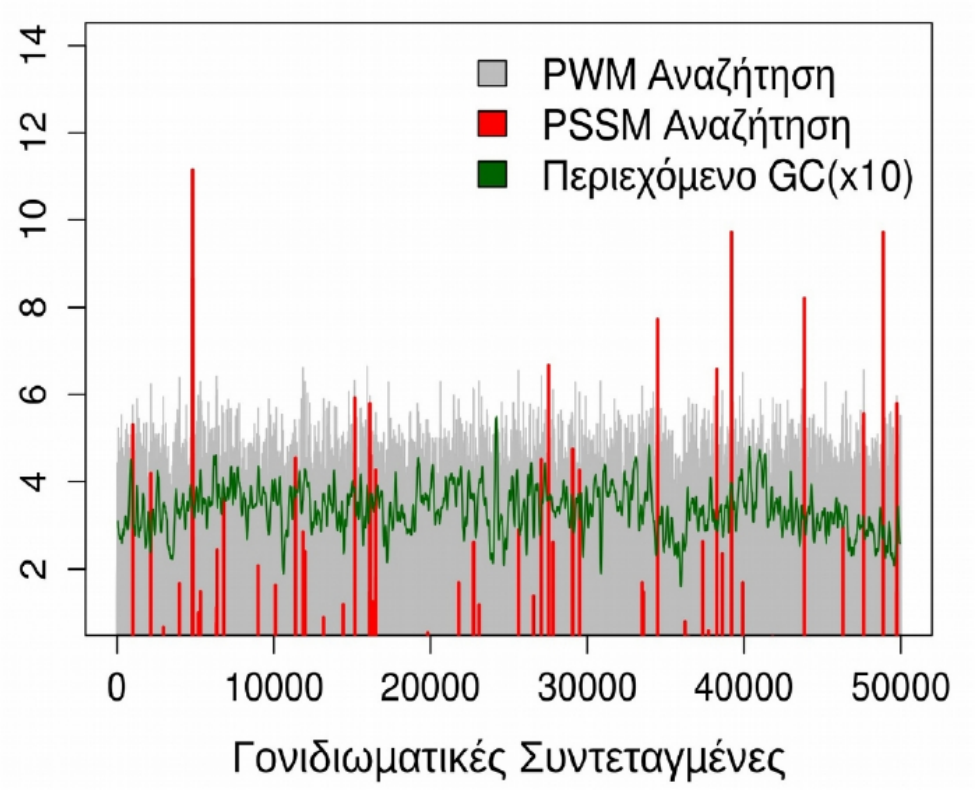
```
import numpy as np

pssm=np.genfromtxt('pssm.tsv', names=True,
+delimiter='\t', dtype=None)
size=len(pssm)

score=[0 for x in range(len(seq)-size)];


for i in range(len(seq)-size+1):
  pattern=""
    for j in range(size):
      pattern=pattern+seq[i+j]
        score[i]=score[i]+pssm[seq[i+j]][j]
    print pattern,"\t",score[i]
```

# Problem #2: Finding a motif: PSSM vs PWM




1. See how noisy the PWM output is. Why?
2. What makes the PSSM more specific?

# Problem #3: Evaluating a motif instance

We saw how every motif can be described as a PWM. But:

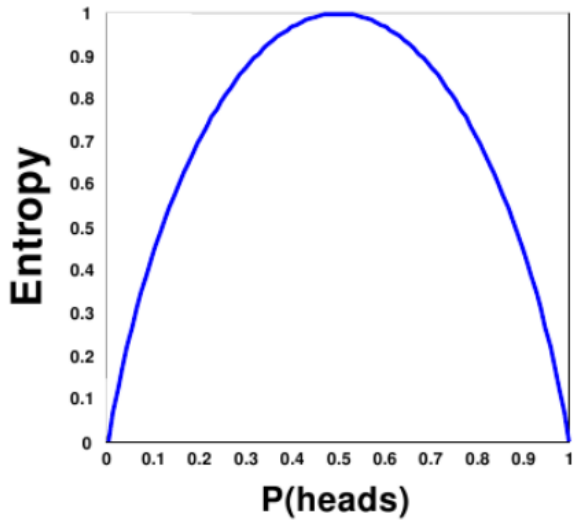
1. How are different PWM describing patterns?
2. How strong is the motif given its PWM?

# Mathematics Interlude: Information as Entropy

- In 1948 Claude Shannon's pioneering work on message transmission introduce a fundamental concept and gave rise to a whole field of Science called "Information Theory"
- The basis of information theory is the concept of Entropy which is defined as:
  - Given the set $S$ of $n$ probable outcomes of a "source", each of which has probability $P[i]$
  - The "Shannon" Entropy of this source is equal to the negative sum of the products of those probabilities and their logarithms, such as: $H(S) = -\sum_{i=1}^{n} P[i] log(P[i])$

# Mathematics Interlude: Information as Entropy

- It derives from Shannon's formula that Entropy maximizes when all possible outcomes have equal probability
- This is directly related to the notion of Entropy as you know it from Physics. Can you see how?




$$H(X) = -\sum_i p_i \log_2 p_i$$

# Stop and think: How is this related to motifs?

- A motif where all positions are equiprobable for all nucleotides has maximum Entropy
- It also conveys the least possible information. There isn't absolutely anything it can tell us about where the sequence has embedded a message
- According to Information Theory, Information can be measured as the change in the Entropy before and after a message has been transmitted: $I(S) = H(S)_{before} - H(S)_{after}$

# Problem #3: Evaluating a motif with Information (I)

- What is the maximum entropy for any given position in a motif?
  $H(S)_{before} = -\sum_{i=1}^{4} P[0.25] log(P[0.25]) = 2$
  we will call this the before Entropy
- What is the entropy once the message has been transmitted? We will denote as "after" the entropy we can calculate from the PWM:
  $H(S)_{after} = -\sum_{i=1}^{4} P[i] log(P[i]) = H$ and thus
  $I(S) = 2 - H(S)_{after}$
- The key is that the smaller the $H(S)_{after}$ the more we have gained as information, since we are **reducing the uncertaintly** of the message

# Problem #3: Calculating the Information Content of a motif

- Each position in the motif gets a score $I(p)$
- Each nucleotide in each position gets a weight equal to $P * log(P)$

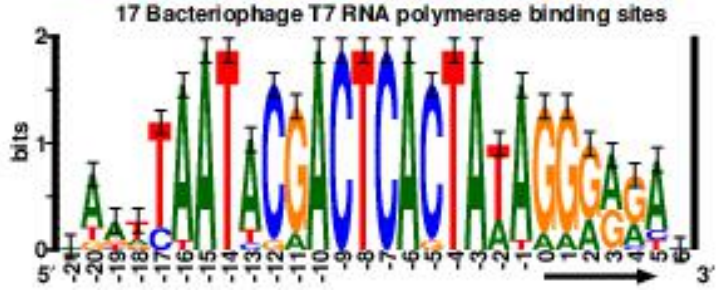
| Νουκλεοτίδιο | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| A | 0.00 | 0.00 | 0.03 | 0.76 | 0.49 | 0.23 | 0.01 | 0.01 | 0.10 | 0.00 |
| C | 0.00 | 0.00 | 0.00 | 0.00 | 0.37 | 0.03 | 0.04 | 0.38 | 0.88 | 0.97 |
| G | 1.00 | 1.00 | 0.97 | 0.24 | 0.01 | 0.05 | 0.07 | 0.00 | 0.00 | 0.00 |
| T | 0.00 | 0.00 | 0.00 | 0.00 | 0.13 | 0.69 | 0.88 | 0.61 | 0.02 | 0.03 |

$$H(X) = -\sum_i p_i \log_2 p_i \qquad I(X) = H_{\pi\rho\iota\nu} - H_{\mu\varepsilon\tau\acute{\alpha}}$$

| Θέση | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| A | 0.00 | 0.00 | 0.05 | 0.92 | 0.25 | 0.18 | 0.01 | 0.01 | 0.14 | 0.00 |
| C | 0.00 | 0.00 | 0.00 | 0.00 | 0.19 | 0.02 | 0.05 | 0.37 | 1.23 | 1.75 |
| G | 2.00 | 2.00 | 1.75 | 0.29 | 0.01 | 0.04 | 0.09 | 0.00 | 0.00 | 0.00 |
| T | 0.00 | 0.00 | 0.00 | 0.00 | 0.07 | 0.53 | 1.16 | 0.59 | 0.03 | 0.05 |
| I(θέσης) | 2.00 | 2.00 | 1.81 | 1.20 | 0.52 | 0.78 | 1.32 | 0.97 | 1.39 | 1.81 |

# Problem #3: Plotting Information as Sequence Logo

| | | | | | Bits |
|---|---|---|---|---|---|
| V01146 | 405 | + | 1 | ttattaatacaactcactataaggagag | 33.3 |
| V01146 | 5848 | + | 2 | aaatcaatacgactcactatagagggac | 37.4 |
| V01146 | 5923 | + | 3 | cggttaatacgactcactataggagaac | 34.4 |
| V01146 | 6409 | + | 4 | gaagtaatacgactcagtatagggacaa | 33.1 |
| V01146 | 7778 | + | 5 | ctggtaatacgactcactaaaggaggta | 30.7 |
| V01146 | 7895 | + | 6 | cgcttaatacgactcactaaaggagaca | 29.1 |
| V01146 | 9107 | + | 7 | gaagtaatacgactcactattagggaag | 31.8 |
| V01146 | 11180 | + | 8 | taattaattgaactcactaaagggagac | 30.1 |
| V01146 | 12671 | + | 9 | gagacaatccgactcactaaagagagag | 28.4 |
| V01146 | 13341 | + | 10 | attctaatacgactcactaaaggagaca | 29.4 |
| V01146 | 13915 | + | 11 | aatactattcgactcactataggagata | 25.2 |
| V01146 | 18545 | + | 12 | aaattaatacgactcactatagggagat | 40.7 |
| V01146 | 21865 | + | 13 | aatttaatacgactcactatagggagac | 41.3 |
| V01146 | 22904 | + | 14 | aaattaatacgactcactatagggagac | 43.1 |
| V01146 | 27274 | + | 15 | aaattaatacgactcactatagggagaa | 43.3 |
| V01146 | 34566 | + | 16 | gaaataatacgactcactatagggagag | 40.3 |
| V01146 | 39229 | + | 17 | aaattaatacgactcactatagggagag | 43.1 |

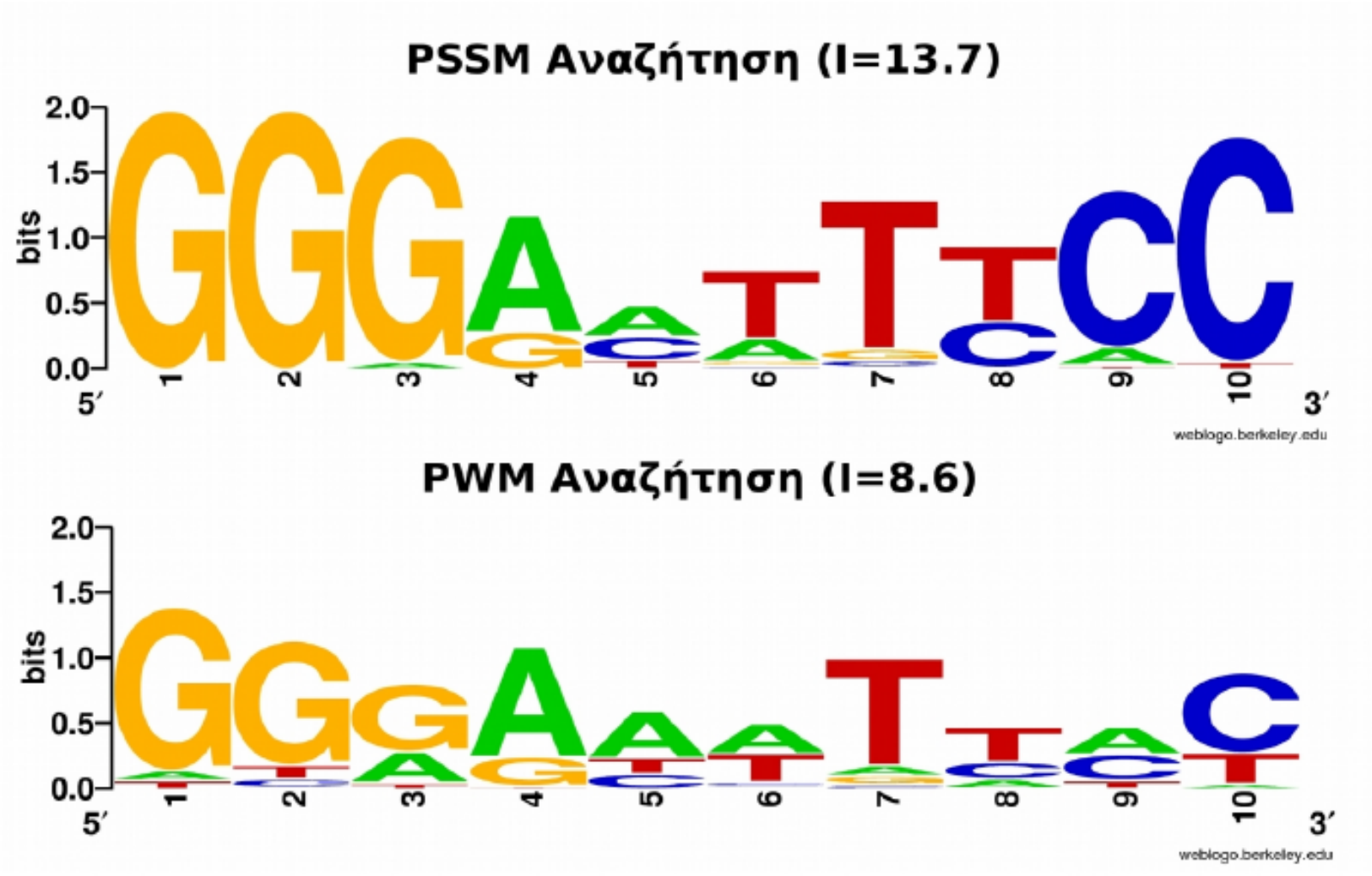


# Problem #3: Create a logo

1. Download a set of motif instances from the GATA binding factor here (`https://tinyurl.com/ms6rm24`)
2. Go to the Webpage of Weblogo, an implementation of the Sequence Logo concept here: (`http://weblogo.berkeley.edu/logo.cgi`)
3. Paste in sequences
4. Obtain Logo

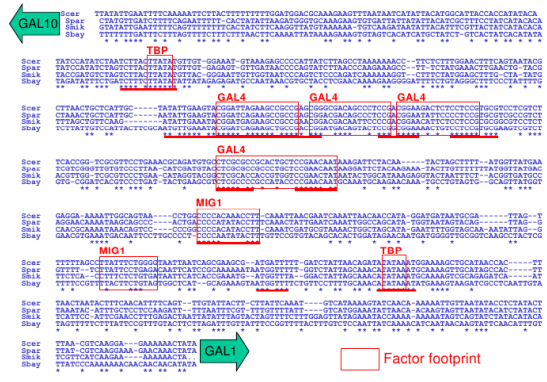
# Problem #3: Evaluation of motifs

Compare the top5% scores of our PSSM and PWM search



# Problem #4: The hard one

- Given a set of sequences, can you locate sequence instances that will represent a motif?
  These should fulfill the following:
  1. They should be more common than other (how much more common?)
  2. They should occur in close vicinity to each other (but how close?)
  3. They are probably going to be conserved in evolution (but how are we going to see this?)




- Next time: How do we **discover** motifs in sequences

# Exercises: To think about

1. Write a program to scan a sequence of DNA with a given pattern with length $L$ and extract all substrings with Hamming distance of $d <= 2/L$. *Key: Think of ways to make it faster*
2. Write a program that will take the GATA sequences and input and will produce a PWM
3. Take the above program and combine it with an analysis of genomic sequence composition (see previous chapter) to:
   - 3.1 Create a background composition model
   - 3.2 Create a PSSM based on the BC model and the PWM
4. Write the code that given a set of sequences of equal size N, will produce the Entropies and total Information per position that you can use to create a logo