CEITEC

Central European Institute of Technology

BRNO | CZECH REPUBLIC

Introduction to Bioinformatics (LF:DSIB01)

Week 3 : Pattern Recognition

Sequence Patterns

We will learn:

1. How to define a pattern
2. How to identify the presence of a pattern in a sequence
3. How to count pattern occurrences
4. Calculation of overrepresentation
5. Creation of elaborate queries

Defining pattern

There are several ways to define a pattern

- Deterministic Patterns
- Patterns with mismatches
- Position Weight Matrices
- Stochastic Models

Deterministic Patterns

- Defined Alphabet Σ=[A,T,G,C]
- Simple Sequence: e.g. TATAAAA
- Ambiguous character: e.g. TAT[AT]AAA : [AT] = either A or T
- Wildcard: TAT **.** AAA : **.** = any character
- Flexible gap: TAT**. {1,3}**AAA : **.{1,3}** = one to three times any character

Patterns with mismatches

- Allow exact matching of Deterministic Pattern + a certain number of mistakes
- This category will be covered in depth over the next 2 lectures (Week 4,5)

Position Probability Matrices (PPM)

- Ambiguous symbol [AT] gives the same % on both symbols
- PPM is a table Position x Alphabet containing probability (%) scores
- Using this PPM one can score the probability that this pattern produces this sequence.

$$M = \begin{matrix} A \\ C \\ G \\ T \end{matrix} \begin{bmatrix} 0.3 & 0.6 & 0.1 & 0.0 & 0.0 & 0.6 & 0.7 & 0.2 & 0.1 \\ 0.2 & 0.2 & 0.1 & 0.0 & 0.0 & 0.2 & 0.1 & 0.1 & 0.2 \\ 0.1 & 0.1 & 0.7 & 1.0 & 0.0 & 0.1 & 0.1 & 0.5 & 0.1 \\ 0.4 & 0.1 & 0.1 & 0.0 & 1.0 & 0.1 & 0.1 & 0.2 & 0.6 \end{bmatrix} . \prod_{i=1}^{k} \frac{\mathrm{M}[x_i, i]}{f(x_i)}$$

Position Weight Matrices (PWM)

- More useful is the log-odds score (weight)
- Odds Score = log2 (frequency / background frequency)
- Sum: How different is the scored seq from a background random seq
- 0 => equal prob of being M or Background, + => M more prob, - => M less prob

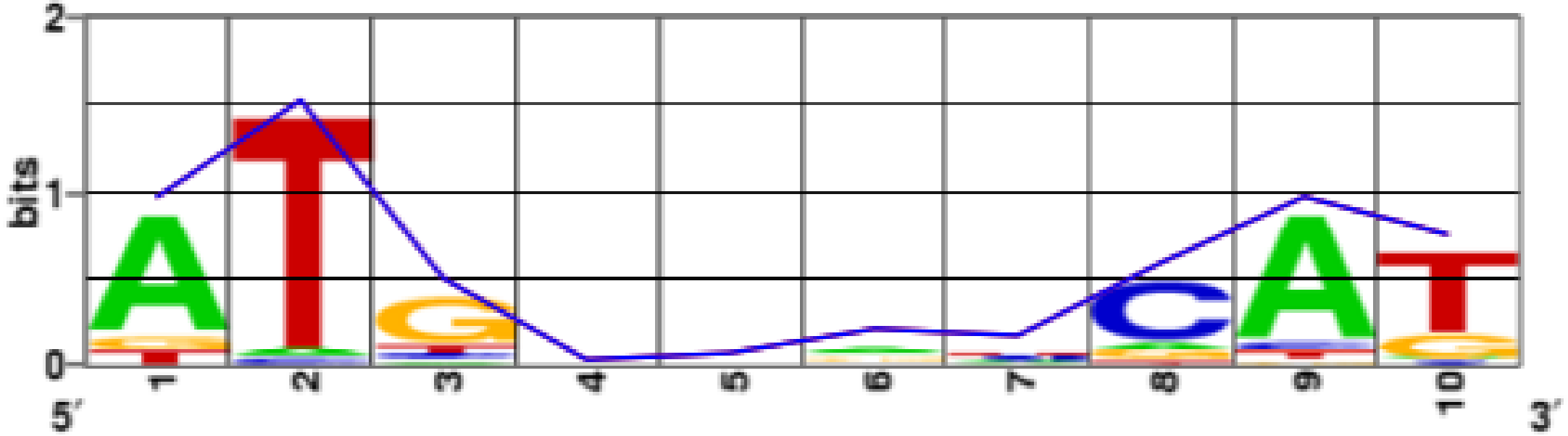
$$M = \begin{matrix} A \\ C \\ G \\ T \end{matrix} \begin{bmatrix} 0.3 & 0.6 & 0.1 & 0.0 & 0.0 & 0.6 & 0.7 & 0.2 & 0.1 \\ 0.2 & 0.2 & 0.1 & 0.0 & 0.0 & 0.2 & 0.1 & 0.1 & 0.2 \\ 0.1 & 0.1 & 0.7 & 1.0 & 0.0 & 0.1 & 0.1 & 0.5 & 0.1 \\ 0.4 & 0.1 & 0.1 & 0.0 & 1.0 & 0.1 & 0.1 & 0.2 & 0.6 \end{bmatrix}.$$

$$\prod_{i=1}^{k} \frac{\mathrm{M}[x_i, i]}{f(x_i)}$$

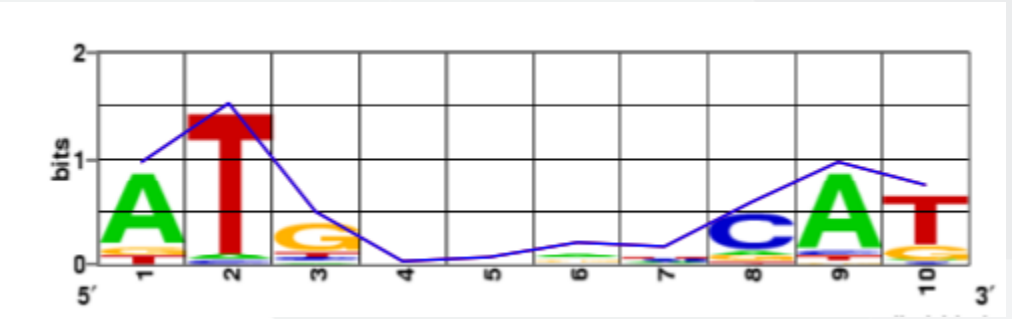
$$M = \begin{matrix} A \\ C \\ G \\ T \end{matrix} \begin{bmatrix} 0.26 & 1.26 & -1.32 & -\infty & -\infty & 1.26 & 1.49 & -0.32 & -1.32 \\ -0.32 & -0.32 & -1.32 & -\infty & -\infty & -0.32 & -1.32 & -1.32 & -0.32 \\ -1.32 & -1.32 & 1.49 & 2.0 & -\infty & -1.32 & -1.32 & 1.0 & -1.32 \\ 0.68 & -1.32 & -1.32 & -\infty & 2.0 & -1.32 & -1.32 & -0.32 & 1.26 \end{bmatrix}.$$

$$\sum_{i=1}^{k} \mathrm{M}'[x_i, i].$$

Information Content



Information Content



Height = Information content (bits)

Information = degree of decrease in uncertainty

Hartley 1928: I(N) = log(N)

Shannon 1948: $I(a_i) = \log\left(\frac{1}{P(a_i)}\right) = \log(1) - \log(P(a_i)) = -\log(P(a_i)), \quad P(a_i) \in [0,1]$

$$I(a_i) = -\log\left(\frac{1}{n}\right) = -[\log(1) - \log(n)] = \log(n), \quad P(a_i) = \frac{1}{n}$$

$$I(a_1) = I(a_2) = -\log\left(\frac{1}{2}\right) = \log(2)$$ < Definition of a bit

https://erilllab.umbc.edu/files/2016/04/Introduction_Information_Theory.pdf

Entropy

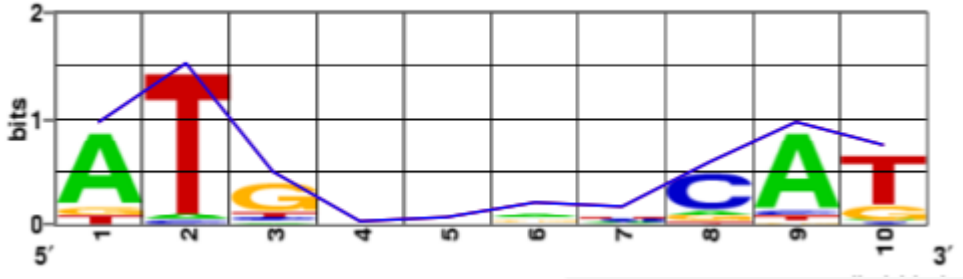
Entropy is a measure of the *unpredictability* of a state (*average information content)*

$$\mathrm{H}(X) = \mathrm{E}[\mathrm{I}(X)] = \mathrm{E}[-\log(\mathrm{P}(X))].$$

	1	2	3	4	5	6	7	8	9	10
A	0.76	0.04	0.08	0.28	0.12	0.44	0.24	0.12	0.80	0.04
C	0.00	0.04	0.12	0.32	0.28	0.12	0.28	0.68	0.08	0.04
T	0.12	0.92	0.16	0.16	0.28	0.12	0.40	0.08	0.08	0.68
G	0.12	0.00	0.64	0.24	0.32	0.32	0.08	0.12	0.04	0.24

$$H(X) = H_{before}(l) = -\sum_{S\in\Omega}[f(S)\cdot(\log_2(f(S))]$$

$$H(X \mid Y) = H_{after}(l) = -\sum_{S_l\in\Omega}(p(S_l)\cdot\log_2(p(S_l)))$$

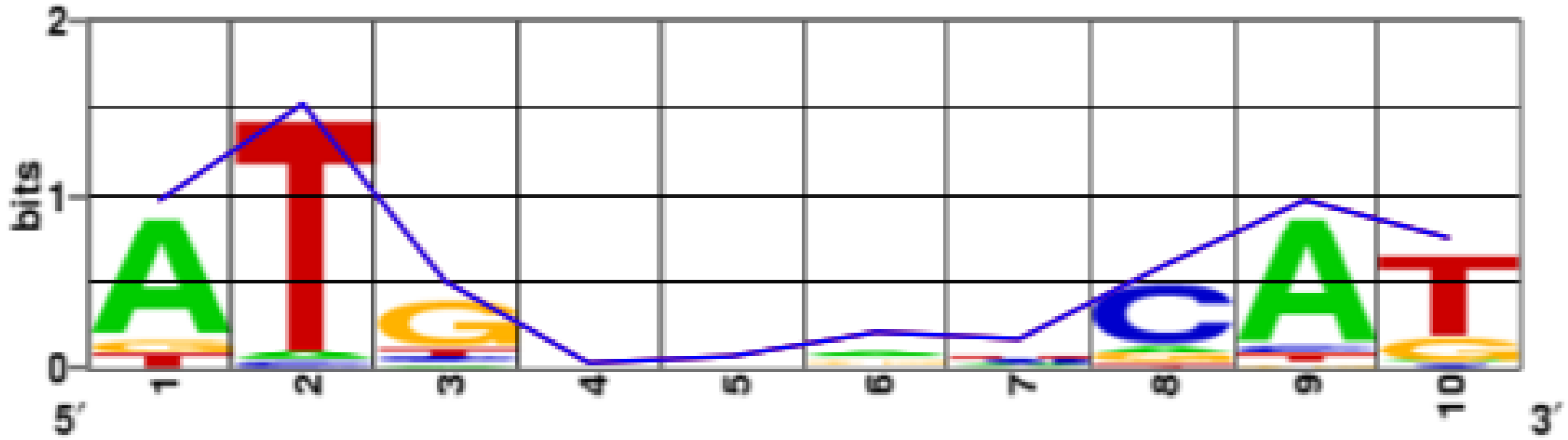


$$I(l) = H_{before}(l) - H_{after}(l) = \left[-\sum_{S\in\Omega}(f(S)\cdot(\log_2(f(S)))\right] - \left[-\sum_{S_l\in\Omega}(p(S_l)\cdot\log_2(p(S_l)))\right]$$

$R_{sequence}(l)$

https://erilllab.umbc.edu/files/2016/04/Introduction_Information_Theory.pdf

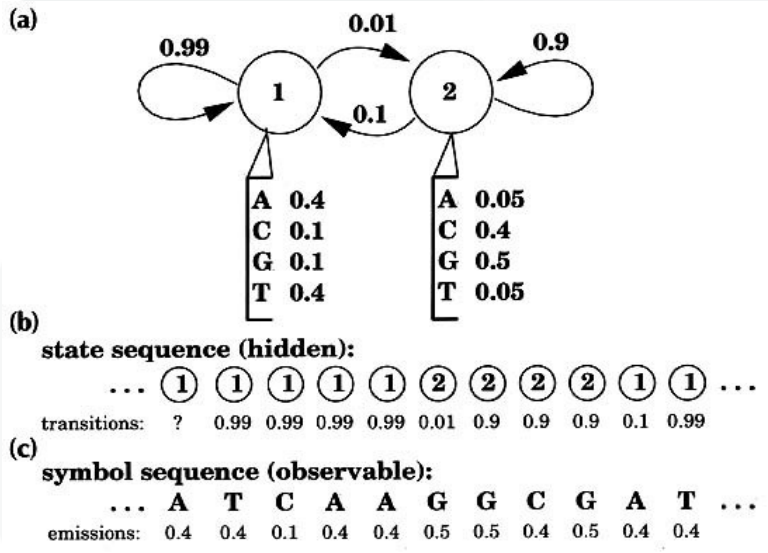
How to read a Seq Logo



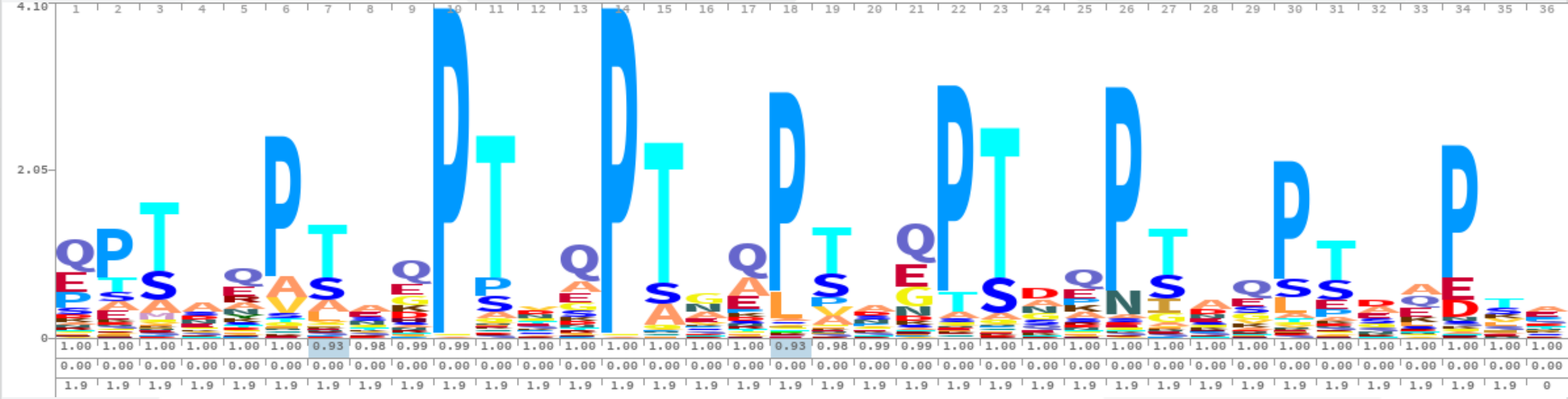
Stack height indicates the Information Content per position (Rseq(I))
Letter height indicates the Base Frequency per position

Stochastic models

- A set of rules or a machine learning method
- Must be able to discriminate / classify / score sequence
- Commonly used: Hidden Markov Models
- We will not talk about stochastic models in this course



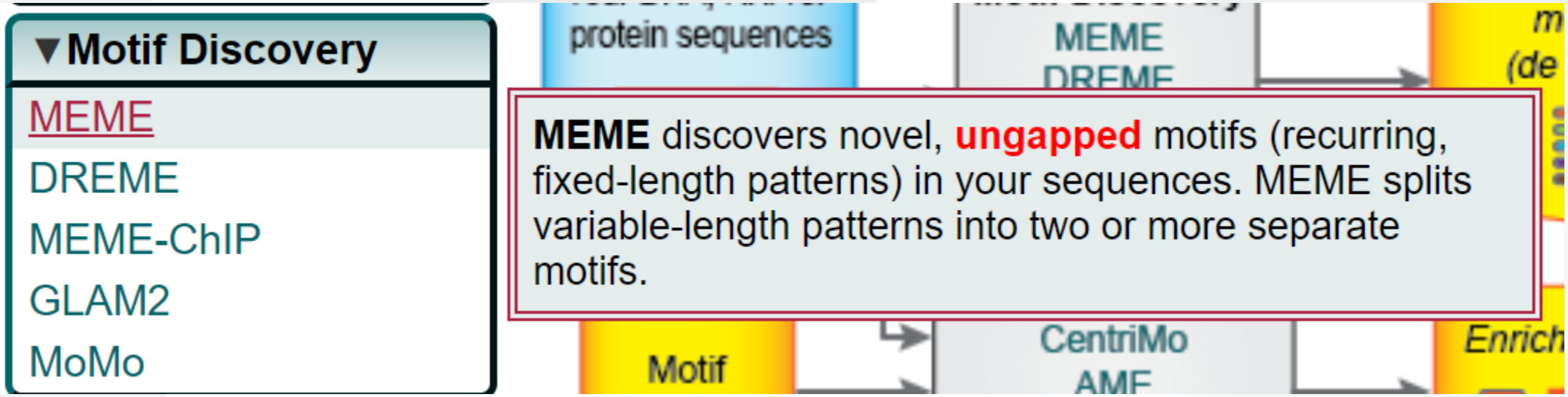
HMM Logo (PFAM)



https://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-15-7

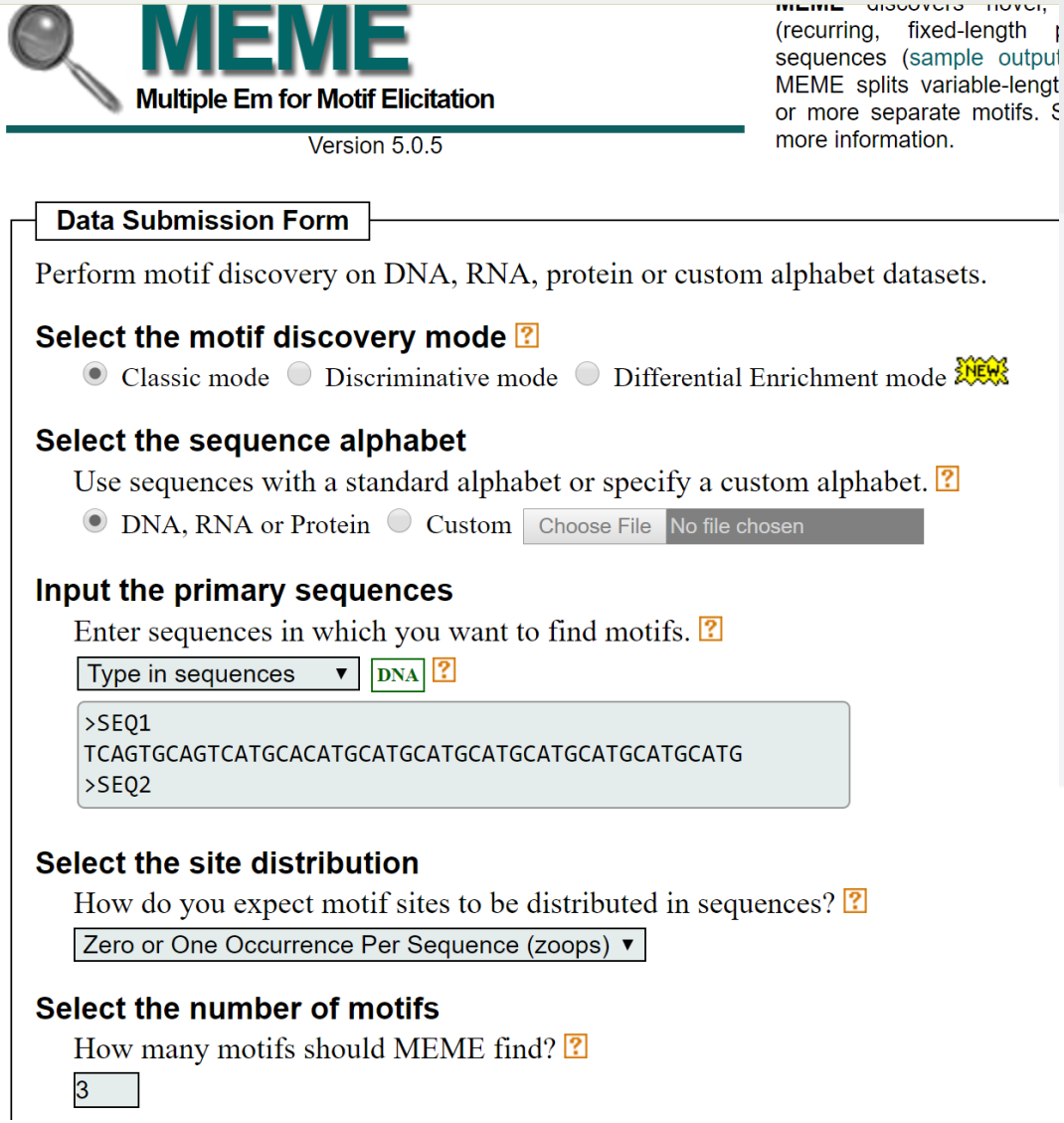

Hunting for Models

Motif Discovery with MEME



Motif Discovery with MEME

```
>SEQ1
TCAGTGCAGTCATGCACATGCATGCATGCATGCATGCATGCATGCATG
>SEQ2
TGTGCTGACTGCATGACTCTATCTGCATGACTGTTTCTGCGCGGC
>SEQ3
TGCATGCATGCACTGAAAAAAAAAATGCATGCATGCACTGACATGCTGACTGA
```



Motif Discovery with MEME

Discovered Motifs

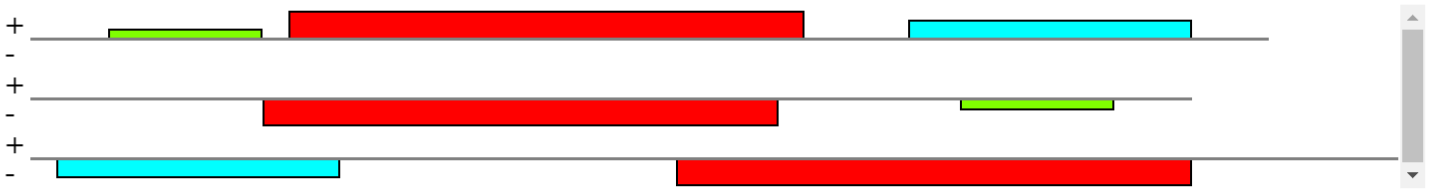
	Logo	E-value	Sites	Width	More	Submit/Download
1.		1.7e+000	3	20		
2.		2.0e+002	2	11		
3.		1.5e+003	2	6		

Stopped because requested number of motifs (3) found.

Motif Locations

Only Motif Sites | Motif Sites+Scanned Sites | All Sequences

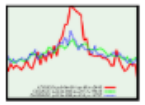
	Name	*p*-value	Motif Location
1.	SEQ1	4.78e-14	
2.	SEQ2	4.79e-9	
3.	SEQ3	4.26e-11	



Motif Discovery with MEME

MEME
Improved

Multiple Em for Motif Elicitation



CentriMo
Local Motif Enrichment Analysis

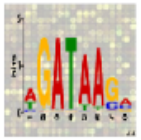
FIMO
Find Individual Motif Occurrences

DREME
Discriminative Regular Expression Motif Elicitation

AME
Improved

Analysis of Motif Enrichment

MAST
Motif Alignment & Search Tool



MEME-ChIP
Improved

Motif Analysis of Large Nucleotide Datasets

SpaMo
Spaced Motif Analysis Tool

MCAST
Motif Cluster Alignment and Search Tool

GLAM2
Gapped Local Alignment of Motifs

GOMo
Gene Ontology for Motifs

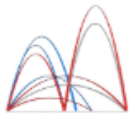
GLAM2Scan
Scanning with Gapped Motifs

MoMo
Improved

Modification Motifs

Tomtom
Motif Comparison Tool

GT-Scan
Identifying Unique Genomic Targets



CisMapper
New

Predicting Regulatory Links

http://meme-suite.org/

Regular Expressions

- For Deterministic Patterns
- Regex is a commonly used language for definition of Deterministic Patterns
- Extremely powerful syntax – but ATTENTION for unintended results

1. Meta-characters
2. Special Characters
3. Sets

https://www.w3schools.com/python/python_regex.asp

Meta-characters

Character	Description	Example
[]	A set of characters	"[a-m]"
\	Signals a special sequence (can also be used to escape special characters)	"\d"
.	Any character (except newline character)	"he..o"
^	Starts with	"^hello"
$	Ends with	"world$"
*	Zero or more occurrences	"aix*"
+	One or more occurrences	"aix+"
{}	Exactly the specified number of occurrences	"al{2}"
|	Either or	"falls|stays"
()	Capture and group	

```
import re

str = "hello world"

#Check if the string starts with 'hello':

x = re.findall("^hello", str)
if (x):
  print("Yes, the string starts with 'hello'")
else:
  print("No match")
```


https://www.w3schools.com/python/python_regex.asp

Special Characters

\	Signals a special sequence (can also be used to escape special characters)

```
import re

str = "The rain in Spain"

#Check if the string starts with "The":

x = re.findall("\AThe", str)

print(x)

if (x):
  print("Yes, there is a match!")
else:
  print("No match")
```

Character	Description	Example
\A	Returns a match if the specified characters are at the beginning of the string	"\AThe"
\b	Returns a match where the specified characters are at the beginning or at the end of a word	r"\bain" r"ain\b"
\B	Returns a match where the specified characters are present, but NOT at the beginning (or at the end) of a word	r"\Bain" r"ain\B"
\d	Returns a match where the string contains digits (numbers from 0-9)	"\d"
\D	Returns a match where the string DOES NOT contain digits	"\D"
\s	Returns a match where the string contains a white space character	"\s"
\S	Returns a match where the string DOES NOT contain a white space character	"\S"
\w	Returns a match where the string contains any word characters (characters from a to Z, digits from 0-9, and the underscore _ character)	"\w"
\W	Returns a match where the string DOES NOT contain any word characters	"\W"
\Z	Returns a match if the specified characters are at the end of the string	"Spain\Z"

\A vs ^ : ^ matches start of LINE while \A start of string

https://www.w3schools.com/python/python_regex.asp

Sets

Set	Description
[arn]	Returns a match where one of the specified characters (a , r , or n) are present
[a-n]	Returns a match for any lower case character, alphabetically between a and n
[^arn]	Returns a match for any character EXCEPT a , r , and n
[0123]	Returns a match where any of the specified digits (0 , 1 , 2 , or 3) are present
[0-9]	Returns a match for any digit between 0 and 9
[0-5][0-9]	Returns a match for any two-digit numbers from 00 and 59
[a-zA-Z]	Returns a match for any character alphabetically between a and z , lower case OR upper case
[+]	In sets, + , * , . , | , () , $, {} has no special meaning, so [+] means: return a match for any + character in the string

https://www.w3schools.com/python/python_regex.asp

Python regex

Function	Description
findall	Returns a list containing all matches
search	Returns a Match object if there is a match anywhere in the string
split	Returns a list where the string has been split at each match
sub	Replaces one or many matches with a string

Python Regex Cheatsheet

Quick Filter

Regular Expression Basics	
.	Any character except newline
a	The character a
ab	The string ab
a|b	a or b
a*	0 or more a's
\	Escapes a special character

Regular Expression Quantifiers	
*	0 or more
+	1 or more
?	0 or 1
{2}	Exactly 2
{2, 5}	Between 2 and 5
{2,}	2 or more
{,5}	Up to 5

Default is greedy. Append ? for reluctant.

Regular Expression Groups	
(...)	Capturing group
(?P<Y>...)	Capturing group named Y
(?:...)	Non-capturing group
\Y	Match the Y'th captured group
(?P=Y)	Match the named group Y
(?#...)	Comment

Regular Expression Character Classes	
[ab-d]	One character of: a, b, c, d
[^ab-d]	One character except: a, b, c, d
[\b]	Backspace character
\d	One digit
\D	One non-digit
\s	One whitespace
\S	One non-whitespace
\w	One word character
\W	One non-word character

Regular Expression Assertions	
^	Start of string
\A	Start of string, ignores m flag
$	End of string
\Z	End of string, ignores m flag
\b	Word boundary
\B	Non-word boundary
(?=...)	Positive lookahead
(?!...)	Negative lookahead
(?<=...)	Positive lookbehind
(?<!...)	Negative lookbehind
(?()|)	Conditional

Regular Expression Flags	
i	Ignore case
m	^ and $ match start and end of line
s	. matches newline as well
x	Allow spaces and comments
L	Locale character classes
u	Unicode character classes
(?iLmsux)	Set flags within regex

Regular Expression Special Characters	
\n	Newline
\r	Carriage return
\t	Tab
\YYY	Octal character YYY
\xYY	Hexadecimal character YY

Regular Expression Replacement	
\g<0>	Insert entire match
\g<Y>	Insert match Y (name or number)
\Y	Insert group numbered Y

New to Debuggex? Check out the regex tester!

CEITEC

@CEITEC_Brno

Thank you for your attention!
60 minutes lunch break.

www.ceitec.eu