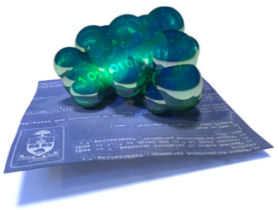
A
Bioinformatics
Course

Sequence Analysis: Comparison



Boris Steipe

department of biochemistry – department of molecular genetics
University of Toronto

Biological patterns are not merely abstract symbols ...

DNA: restriction nuclease site

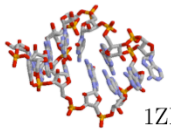
RNA: GNRA Tetraloop

Peptide: KDEL ER retention signal

Protein structure: Zn-finger

5'-R'AATTY-3' ApoI

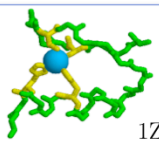
rebase.neb.com



1ZIF.PDB

PDOC00014

www.expasy.ch/prosite



1ZAA.PDB

... but representations of actual molecules.

A sequence is fundamentally different from an unordered set, since its elements provide context for each other.

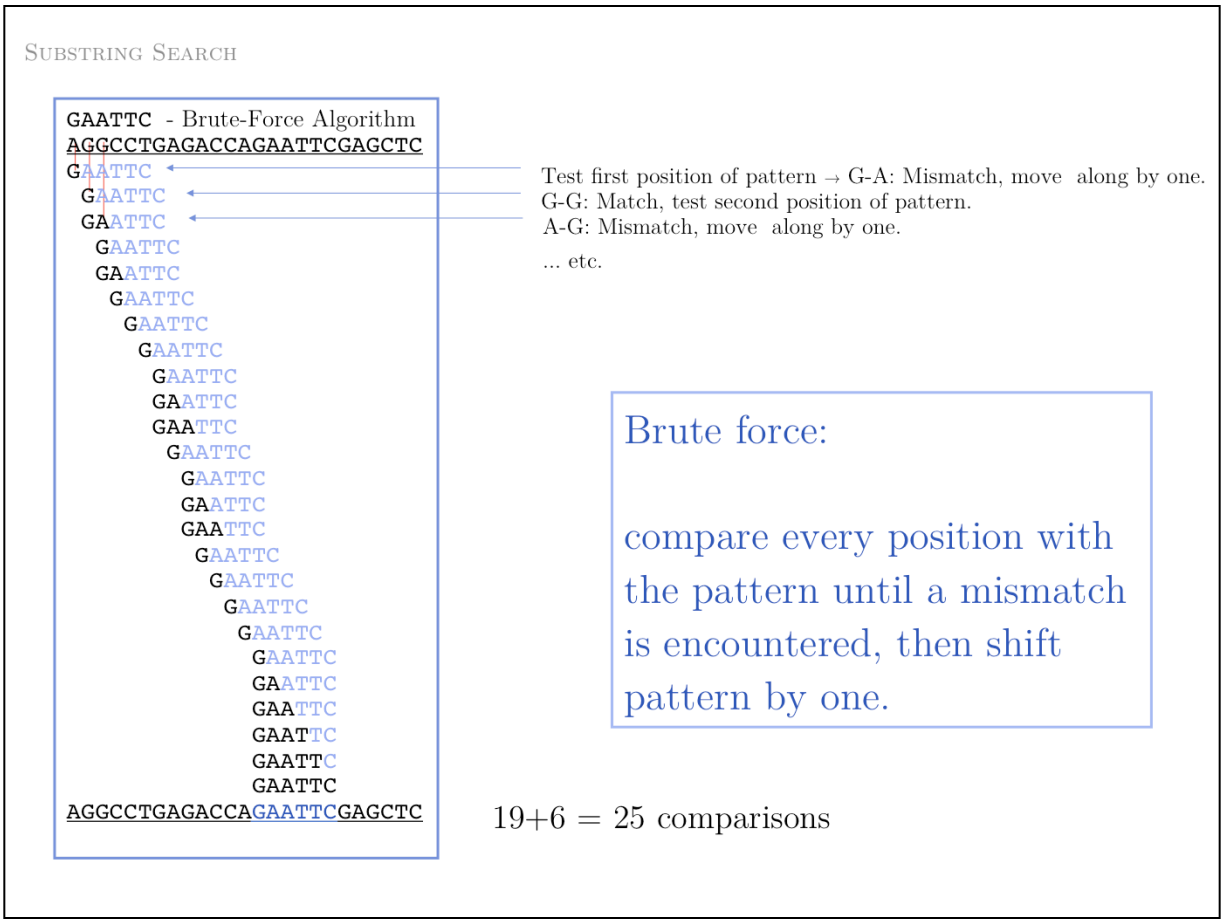
Sequence patterns are not just *signs*, they are different molecules: a pattern with a different sequence is a different pattern. Constraints on patterns can be structural or functional.

Pattern matching is a decision problem

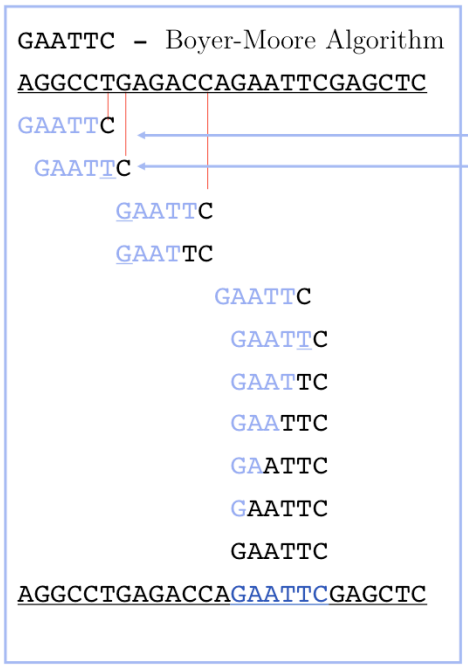
Substring matching Regular expressions	Yes or No answer: Deterministic ...
PSSMs & Profiles HMMs Neural networks Support Vector Machines Decision Trees ...	More or Less answer: Probabilistic ...

Pattern **search** (or pattern matching) means inspecting an entity and stating whether that entity is an example of a given pattern. Usually the entity is a substring of a *sequence*, but patterns in *protein structure*, biological *networks* or *morphogenesis* can also be computationally defined.

Pattern **discovery** means finding patterns that have not been defined *a priori*.



Deterministic pattern matching is a well understood field of computer science. The worst case scenario is that every position of the pattern needs to be compared with every position of the sequence to determine whether an instance of the pattern i spresent in the sequence or not. But much more elegant solutions than this “brute force” approach have been described ...



Boyer-Moore algorithm

Test last position of pattern → C-T: Mismatch, realign pattern to where "T" could have matched (+1)

Test last position of pattern → C-G: Mismatch, realign pattern to where "G" could have matched (+6)

etc. ...

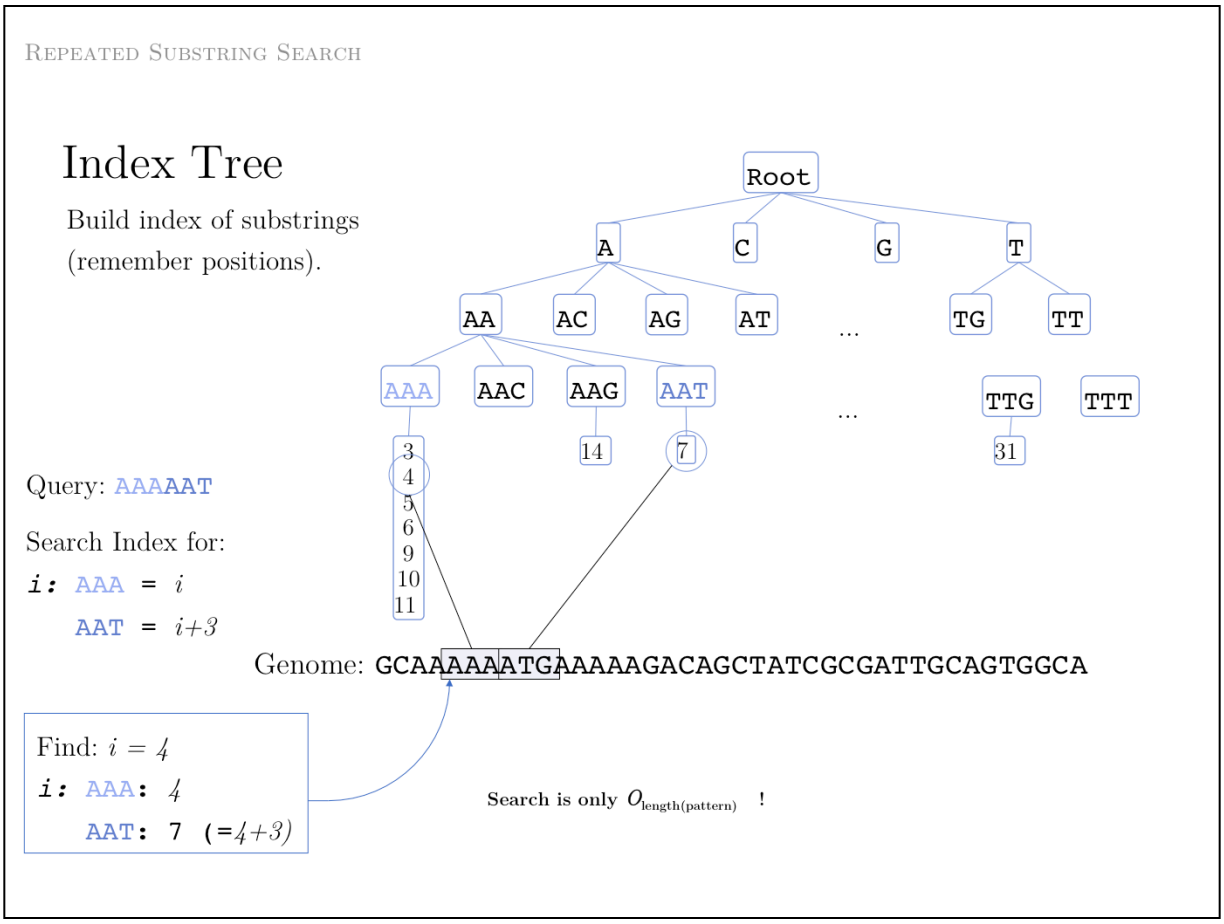
Boyer-Moore:

Move pattern to skip over positions that can't possibly be part of a pattern-match.

5 + 6 = 11 comparisons,
75% more efficient

Check out J. S. Moore's own step-by-step explanation ...
`http://www.cs.utexas.edu/users/moore/best-ideas/string-searching/fstrpos-example.html`

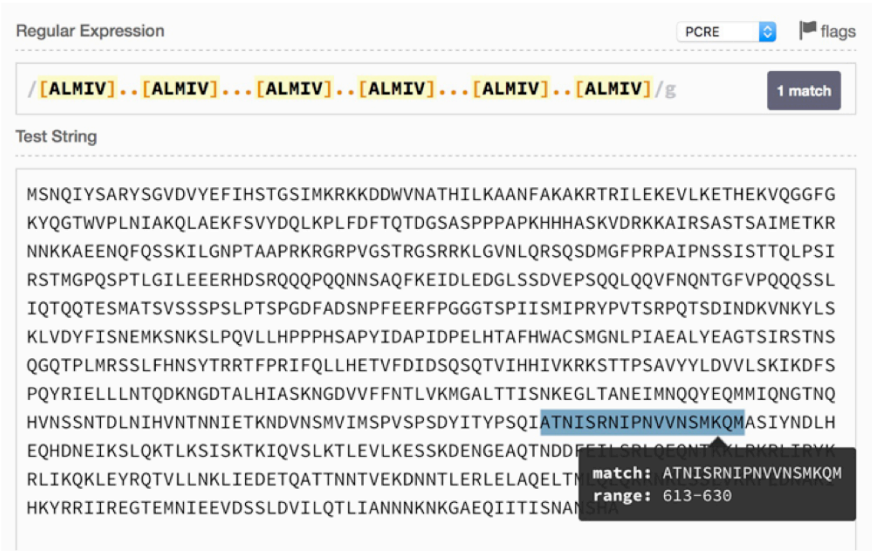
... such as the **Boyer-Moore** algorithm. For a step-by-step explanation see http://www.cs.utexas.edu/users/moore/best-ideas/string-searching/fstrpos-example.html



If searches are to be repeated, pre-computed **index trees** are much faster than examining the entire sequence. In an index tree, simply look up where a pattern could be. Time (and storage space) invested in constructing the index pays off manyfold for every lookup.

Tree-based pattern searches use "suffix trees" to find matches in time proportional to the length of the pattern, not the size of the database!

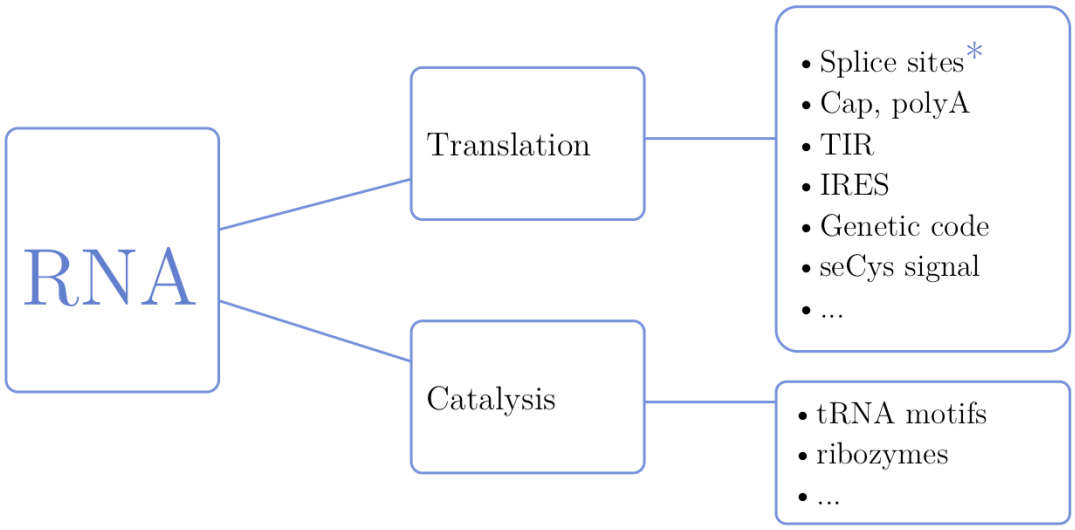
Example: amphipathic helix search in Mbp1



An amphipathic helix has a regular pattern of hydrophobic amino acids which are adjacent in the folded helix and thus form a hydrophobic face.

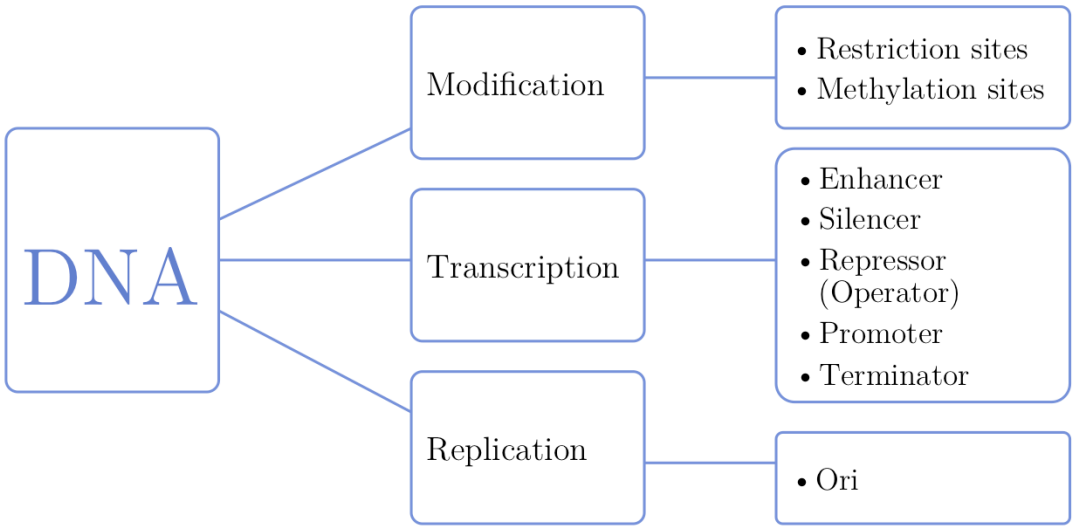
To be able search for patterns we need a convention to define them. In particular, we would like to be able to find degenerate patterns: patterns in which we allow a number of alternative choices for particular positions. Such patterns are commonly written as *Regular Expressions*.

Biological Pattern Examples



*Splice sites: specific bases are required, but context is of key importance.

Biological Pattern Examples



http://rebase.neb.com/

BspMI

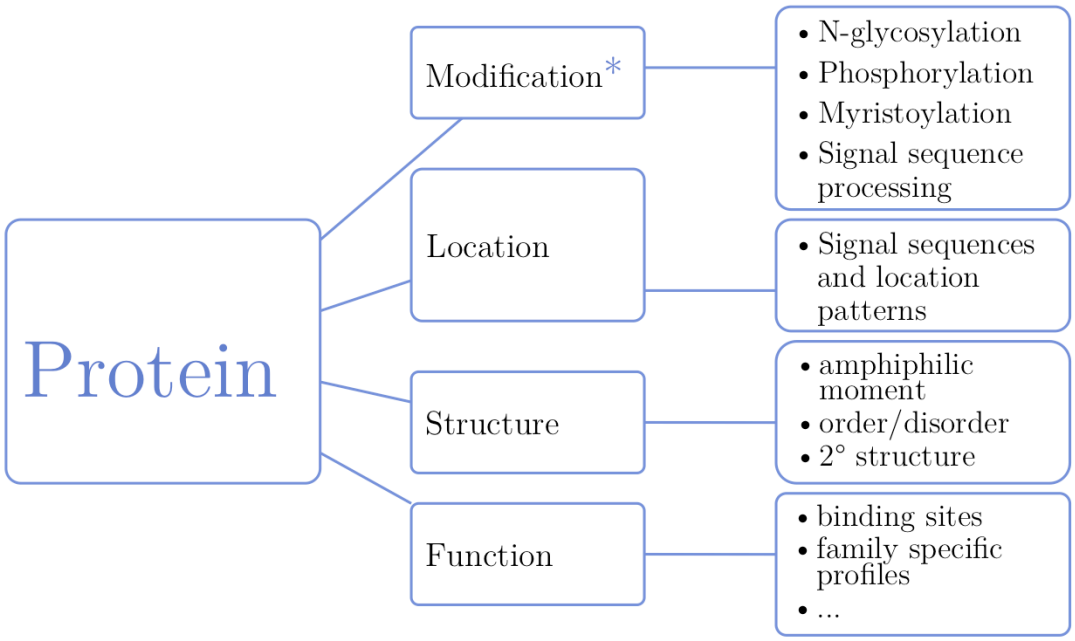
Type IIs restriction enzyme

```
Recognition Sequence: ACCTGC (4/8)


5' .. A C C T G C N N N N↓N N N N 3' ..
3' .. T G G A C G N N N N N N N N↑5' ..

REBASE enzyme #: 527
Prototype: BspMI
Org #: 411
Organism: Bacillus species M
Organism source: NEB 356
Growth Temperature: 37 °
Exhibits star activity
Enzyme gene cloned.
Enzyme gene sequenced.
Entered: Oct 3 1985 ... Modified: Jan 24 2000


Related Records:        Related References...
M.BspMIA M.BspMIB

Commercially            Isoschizomers...
Available...            (commercially available)
```

Restriction endonucleases are the quintessential pattern recognition molecules. They bind strongly the specific conformation of DNA that is associated with a particular DNA sequence. Even though the structural differences between DNA strands of similar sequence is small, evolutionary pressure has resulted in enzymes that are highly specific for their cognate sequence. An excellent site for endonuclease information is Rebase: http://rebase.neb.com/

These patterns are examples of patterns that may be slightly variable in practice, since the cleavage properties of the restriction endonuclease are determined by the free energy of the complex, and different nucleotides may be admissible with reduced catalytic rate – but in practice the enzymes are so discriminatory that a dtereministic pattern matching approach describes the biologically relevant patterns well enough.

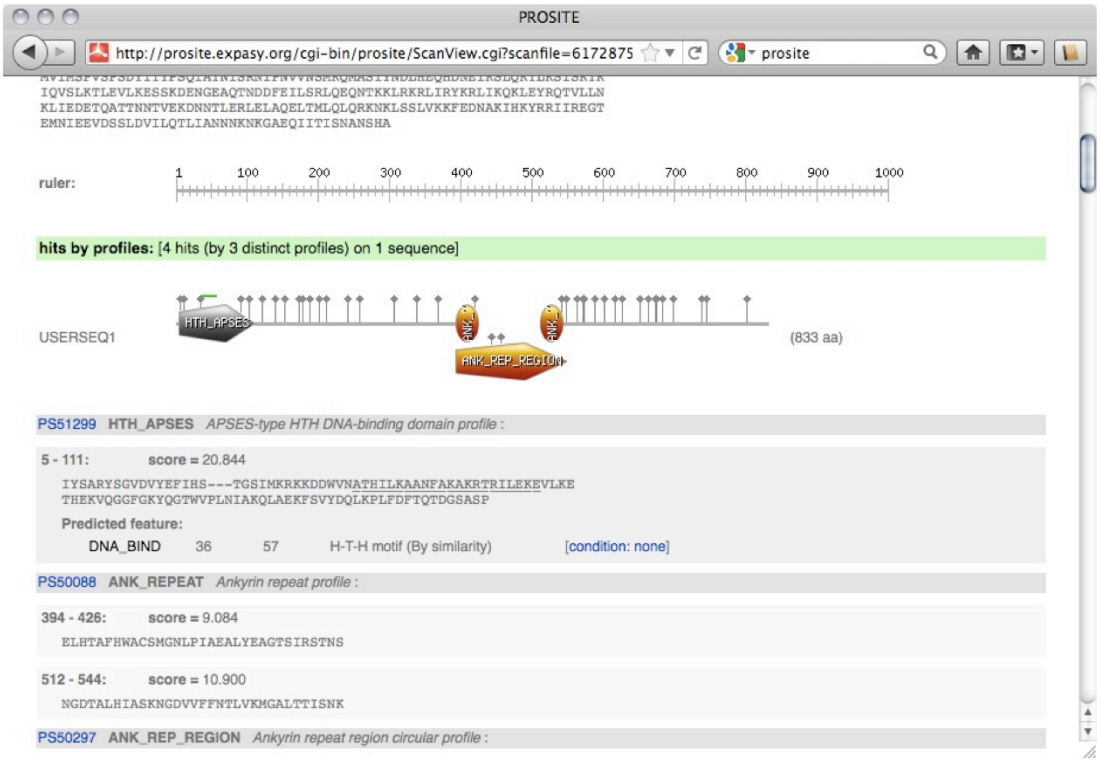
Biological Pattern Examples



* For post translational modification sites, specific residues are required, but structural context usually determines whether any particular site will be modified or not.

A wide variety of protein functions and properties are mediated by simple sequence patterns.

Prosite



The Prosite server (http://prosite.expasy.org) provides a tool that scans sequence for biological patterns – domains, and post-translational modification sites. It also supports scanning of user-defined patterns.

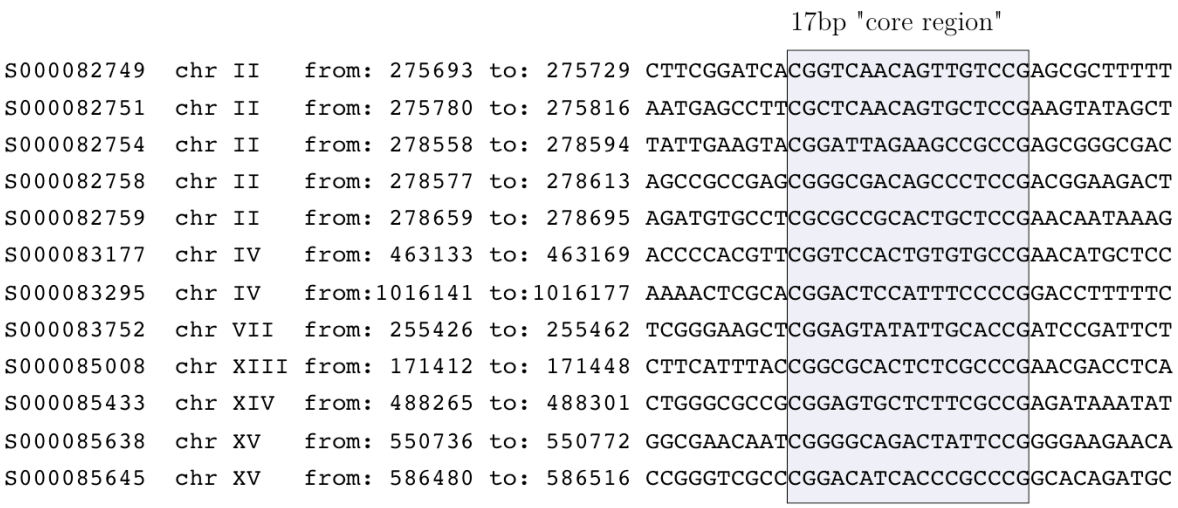
In probabilistic pattern matching, we ask for the probability that a specific sequence is an instance of a generalized pattern.

This is motivated by thermodynamics: there are no "impossible" reactions, since that would imply infinite energy.

$$\Delta G = -RT \ln K$$

Since biomolecular interactions depend on the probabilities of events – captured in the equilibrium constant K, probabilistic descriptions describe biological reality better than deterministic descriptions.

Sequences with a common property: annotated Gal4 binding sites



To generate this collection of sequences, the feature "Gal4-binding-site" was searched in the SGD – Saccharomyces Genome Database; the actual sequences were retrieved by specifying the genome coordinates in the appropriate search form of the database. I have added ten bases upstream and downstream of the core binding region.

Sequences with a common property: annotated Gal4 binding sites

17bp "core region"

```
S000082749  chr II   from: 275693 to: 275729 CTTCGGATCACGGTCAACAGTTGTCCGAGCGCTTTTT
S000082751  chr II   from: 275780 to: 275816 AATGAGCCTTCGCTCAACAGTGCTCCGAAGTATAGCT
S000082754  chr II   from: 278558 to: 278594 TATTGAAGTACGGATTAGAAGCCGCCGAGCGGGCGAC
S0                                                                        GAAGACT
S0                                                                        AATAAAG
S0                                                                        ATGCTCC
S0                                                                        CTTTTTC
S0                                                                        CGATTCT
S0                                                                        GACCTCA
S0                                                                        TAAATAT
S000085638  chr XV   from: 550736 to: 550772 GGCGAACAATCGGGGCAGACTATTCCGGGGAAGAACA
S000085645  chr XV   from: 586480 to: 586516 CCGGGTCGCCCGGACATCACCCGCCCGGCACAGATGC
```

Multiple, non-identical instances of functional sequence fragments represent information about the underlying biological process:

How can we represent this information?

How can we use it for making inferences about an unknown sequence?

The "sequence profile" of Gal4 binding sites can be represented by a consensus sequence.

```
S000082749    CTTCGGATCACGGTCAACAGTTGTCCGAGCGCTTTTT
S000082751    AATGAGCCTTCGCTCAACAGTGCTCCGAAGTATAGCT
S000082754    TATTGAAGTACGGATTAGAAGCCGCCGAGCGGGCGAC
S000082758    AGCCGCCGAGCGGGCGACAGCCCTCCGACGGAAGACT
S000082759    AGATGTGCCTCGCGCCGCACTGCTCCGAACAATAAAG
S000083177    ACCCCACGTTCGGTCCACTGTGTGCCGAACATGCTCC
S000083295    AAAACTCGCACGGACTCCATTTCCCCGGACCTTTTTC
S000083752    TCGGGAAGCTCGGAGTATATTGCACCGATCCGATTCT
S000085008    CTTCATTTACCGGCGCACTCTCGCCCGAACGACCTCA
S000085433    CTGGGCGCCGCGGAGTGCTCTTCGCCGAGATAAATAT
S000085638    GGCGAACAATCGGGGCAGACTATTCCGGGGAAGAACA
S000085645    CCGGGTCGCCCGGACATCACCCGCCCGGCACAGATGC
--------------------------------------------------
Consensus     AATGGACGCTCGGACCACACTGCTCCGAACGAGATCT
```

Pro: The consensus sequence best represents the whole alignment.

Con: No information about how constrained a position is.

A consensus sequence simply lists the most frequent amino acid or nucleotide at each position, or a random one if there is more than one with the highest frequency. The consensus sequence is the one that you would chemically synthesize to make an idealized representative of the set. It is likely to bind more tightly or to be more stable than each of the individual sequences in the alignment.

The "sequence profile" of Gal4 binding sites can be represented by a degenerate consensus sequence.

```
S000082749    CTTCGGATCACGGTCAACAGTTGTCCGAGCGCTTTTT
S000082751    AATGAGCCTTCGCTCAACAGTGCTCCGAAGTATAGCT
S000082754    TATTGAAGTACGGATTAGAAGCCGCCGAGCGGGCGAC
S000082758    AGCCGCCGAGCGGGCGACAGCCCTCCGACGGAAGACT
S000082759    AGATGTGCCTCGCGCCGCACTGCTCCGAACAATAAAG
S000083177    ACCCCACGTTCGGTCCACTGTGTGCCGAACATGCTCC
S000083295    AAAACTCGCACGGACTCCATTTCCCCGGACCTTTTTC
S000083752    TCGGGAAGCTCGGAGTATATTGCACCGATCCGATTCT
S000085008    CTTCATTTACCGGCGCACTCTCGCCCGAACGACCTCA
S000085433    CTGGGCGCCGCGGAGTGCTCTTCGCCGAGATAAATAT
S000085638    GGCGAACAATCGGGGCAGACTATTCCGGGGAAGAACA
S000085645    CCGGGTCGCCCGGACATCACCCGCCCGGCACAGATGC
--------------------------------------------------
Consensus     AATGGACGCTCGGACCACACTGCTCCGAACGAGATCT
Degenerate    AnTGGwCGCTCGGACrACACTsCTCCGAACGAkATCT
```

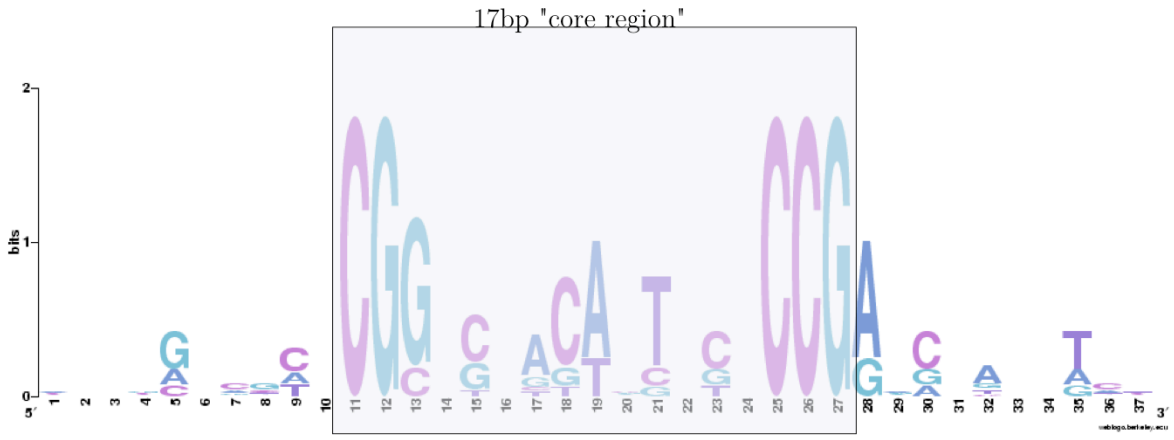
Pro: Better indication of ambiguity.

Con: Refinements become increasingly arbitrary. Doesn't work for amino acids.

The degenerate consensus sequence uses ambiguity codes to capture variability and type of variation better than a specific representative nucleotide could.

Defining Probabilistic Patterns: Sequence Logo

The "sequence profile" of Gal4 binding sites can be represented by a "sequence logo".



Pro: Much better indication of conservation/propensity of characters.

Con: Small samples have artificially high information scores.

Sequence logo of Gal4 binding sites with 10 nucleotides flanking bases. Created with **WebLogo** (http://weblogo.berkeley.edu/).

A Sequence Logo is a graphical representation of aligned sequences where at each position the height of a column is proportional to the (Shannon) information of that position and the relative size of each character is proportional to its frequency in the column. Sequence Logos were pioneered by Tom Schneider who maintains an informative Website about their use and theoretical foundations.

http://www.lecb.ncifcrf.gov/~toms/sequencelogo.html

A Position Frequency Matrix (PFM) – sometimes also called Sequence Profile – records the number of observations of every character in every position of a multiple alignment.

e.g. TRANSFAC, for transcription factors
http://www.gene-regulation.com/

```
...
AC   M00049
XX
ID   F$GAL4_01
XX
DT   13.04.1995 (created); ewi.
DT   16.10.1995 (updated); dbo.
CO   Copyright (C), Biobase GmbH.
XX
NA   GAL4
XX
DE   GAL4
XX
BF   T00302 GAL4; Species: yeast, Saccharomyces cerevisiae.
XX
PO       A       C       G       T
01       1       5       3       2       N
02       5       2       1       3       N
03       3       2       1       5       N
04       1      10       0       0       C
05       0       0      10       1       G
06       0       1      10       0       G
07       4       3       3       1       N
08       1       3       4       3       N
09       2       4       4       1       N
10       7       0       2       2       A
11       1       8       2       0       C
12       4       1       0       6       W
13       1       3       5       2       N
14       0       2       1       8       T
15       1       6       2       2       C
16       1       5       4       1       S
17       2       1       1       7       T
18       0      10       1       0       C
19       0      11       0       0       C
20       0       0      11       0       G
21       8       0       0       3       A
22       7       0       4       0       R
23       2       6       3       0       S
XX
BA   11 genomic binding sites from 6 genes
XX
...
```

Defining Probabilistic Patterns: Position Specific Scoring Matrix

A Position Specific Scoring Matrix (PSSM) expresses observed frequencies as a *score*, e.g. a log-odds score for each observed character, or an information based score.

When a log-odds score is used, the probability of observing a sequence can simply be calculated from the sum of scores *(assuming independence of positions)*.

Pos	A	C	G	T
01	-5.92	-1.13	-5.92	-5.92
02	-5.92	-5.92	-1.13	-5.92
03	-5.92	-2.88	-1.31	-5.92
04	-1.99	-3.53	-2.49	-2.49
05	-5.92	-1.66	-2.21	-3.53
06	-2.49	-2.21	-3.53	-2.21
07	-1.53	-3.53	-2.88	-3.53
08	-5.92	-1.41	-2.88	-3.53
09	-1.41	-5.92	-5.92	-2.49
10	-3.53	-1.99	-2.21	-2.88
11	-5.92	-2.88	-3.53	-1.41
12	-3.53	-2.21	-2.21	-2.49
13	-5.92	-1.66	-2.49	-2.88
14	-3.53	-2.49	-2.49	-1.99
15	-5.92	-1.13	-5.92	-5.92
16	-5.92	-1.13	-5.92	-5.92
17	-5.92	-5.92	-1.13	-5.92

Pro: Captures all information log(p) for match is simply the sum of weights.

Con: Not very readable. (Arbitrary) corrections have to be applied for unobserved states.

Since log(0) is not defined, we have to introduce an arbitrary correction for unobserved characters. In this example I have added 0.1 to each character frequency before calculating log odds.

Scanning a sequence ...

Experimentally annotated Gal4 binding site:	Sequence	
Y$GAL1_03	CGGATTAGAAGCCGCCG	
Y$GAL1_04	CGGGTGACAGCCCTCCGA	
Y$GAL1_05	AGGAAGACTCTCCTCCG	
Y$GAL1_06	CGCGCCGCACTGCTCCGAACAAT	
Consensus:	CGGNNNACWNTCSTCCGARS	
		"score":
chrXIII:171415,171441	TACCGGCGCACTCTCGCCCGAACGA	(4)
chrXIII:171416,171442	TACCGGCGCACTCTCGCCCGAACGA	(13)
chrXIII:171417,171443	TACCGGCGCACTCTCGCCCGAACGA	(6)

But: at which score will we assume that a match is biologically meaningful ?

In this informal example, I have simply counted matches with the consensus sequence (excluding "N"). We can slide the PSSM over the entire chromosome, and calculate scores for each position. Only the middle sequence is an annotated binding site. Whatever method we use for probabilistic pattern matching, we will **always** get a score. It is then **our** problem to decide what the score means.

If the PSSM has been created like we mentioned above, the score can be interpreted as a probability. Then we can apply a common level of significance to determine whether a match should be considered better than random. At least in principle, that's what we would do. In practice, biological sequences are **notorious** for violating assumptions about the independence of positions, upon which the probability/significance argument is based.

Machine learning: generalized representations of patterns

PSSMs are limited, especially to represent patterns that have variable length gaps ...

"Machine Learning" has developed alternative ways to represent high-dimensional information and to classify it. Examples are Markov Models, Neural Networks, Support Vector Machines ... and many more techniques

... but the principle of representing probabilities rather than discrete events is similar.

Machine learning succeeds wherever flexible, general patterns are needed for decision problems and cannot be generated from first principles, and where training sets exist.

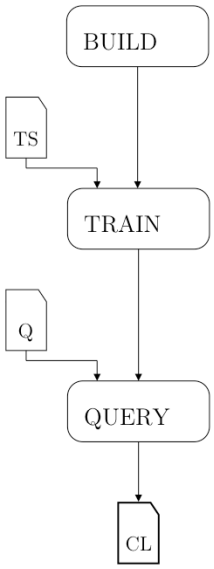
"Data rich and theory-poor."

Example applications

- Signal peptide recognition
- Gene finding
- Splice sites, Exons and Introns
- Protein domain boundaries
- ...

... however: machine learning will find correlations, not causalities. It cannot replace your biological insight to distinguish a statistical anomaly from a biologically meaningful result!

Machine Learning Strategy



BUILD: Define an architecture appropriate to the problem

TS: Training set, containing true positives (TP) and true negatives (TN)

TRAIN: Determine decision parameters that optimally discriminate TP/TN

Q: Unclassified example

QUERY: Analyze features of Q, apply decision parameters, classify

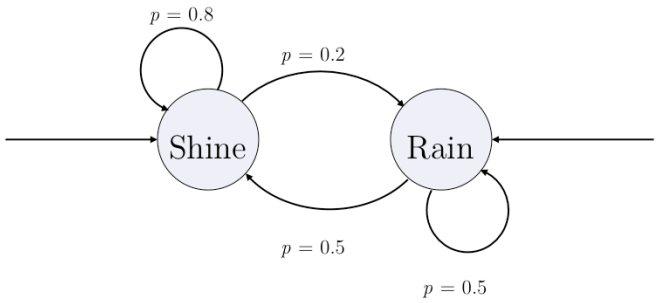
CL: (Classification) result

Machine learning methods must first be trained. Typically we use "supervised" learning approaches for which we define examples of True Positives and True Negatives, for the algorithm to generalize from.

"Unsupervised" approaches exist for special cases and potentially allow *discovering* categories or populations in datasets. The result is commonly a classification probability: the probability that query Q is a member of a category the algorithm was trained on.

Markov Process

In a first order Markov-process, a system moves through a sequence of states, governed by the transition probabilities that are associated with its current state. A Markov model depicts this in a graph of states (nodes) and transitions (edges). Transitions are annotated by their probability.



Example: A Markov chain model for Toronto weather prediction. Note that this not only models the number of rainy days, but also the length of rainy periods.
Estimate the ratio of rainy days to sunshine and the probability for two and three days of rain in a row. How would you adjust the model for Vancouver weather (where it can rain for weeks)?

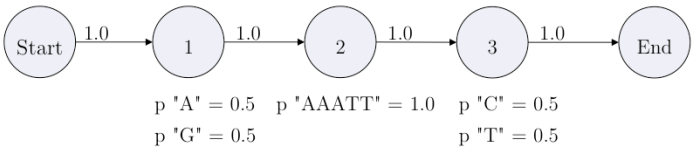
This *first order Markov model* depends only on the current state. Higher-order models take increasing lengths of "history" into account, *i.e.* how the system arrived in its current state.

Note that the exit probabilities for a state always have to sum to 1.0. The so called "stationary probability" over a long period of time for p(rain) is 0.32 - this is determined by the combined effects of all individual transition probabilities. The stationary probabilities for two- or three consecutive rainy days are 8.4% and 4.2%, respectively. This is a very simple model, but it reflects approximately our experience (the average actual number of rainy days in Toronto is 114 per year: 31%).

Here is a site with an online Markov Model simulator where you can play with models and probabilities: http://markov.yoriz.co.uk/

Markov Model

A Markov Model is a stochastic generative model, i.e. a computational device that generates sequences of events. Applied to biological sequences, the "event" is the observation of a particular nucleotide or amino acid. The model has a number of states S_i, and each state has an emission probability matrix E_{ik} that defines the probability with which S emits one character k from an alphabet of symbols, and a transition probability matrix T_{ij} that defines the probability with which the process goes to state j from state i.



A simple Markov model that generates "RAATTY" sequences. Here, transition probabilities T are always 1.0.

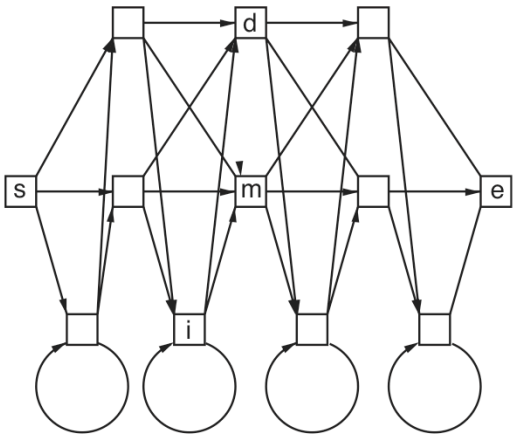
Markov models can be used as general descriptions of patterns. Substrings, PSSMs and profiles can be seen as special cases of Markov Models.

In a "Hidden Markov Model" (HMM), only the emitted symbols are visible, not the state that emitted them, nor the transitions between the states.

Architecture of a general Markov model for sequence patterns

This is the most general model to represent a set of aligned sequences (a profile). Each arrow is a possible transition, with an associated probability that depends on the current state in the node it originates from.

- s: start
- m: matching state - produce a character according to a table of probabilities
- d: delete state - skip a match
- i: insert state - output a new character according to a table of probabilities
- e: end



Build

1. Construct architecture: number of states
2. Initialize with some transition probabilities and emission probabilites according to desired global amino acid composition

Train

3. Examine all possible paths for generating each training sequence
4. Count number of times a specific transition is used to generate the corresponding sequence position
5. Update HMM with improved parameters
6. Repeat, until parameters are stable

Use

7. Query the probability of an observed sequence to have been generated by the parametrized HMM. If p is high: the sequence shares characteristic features with the training set: we may then ascribe some biological significance to the similarity.

HMM advantages:

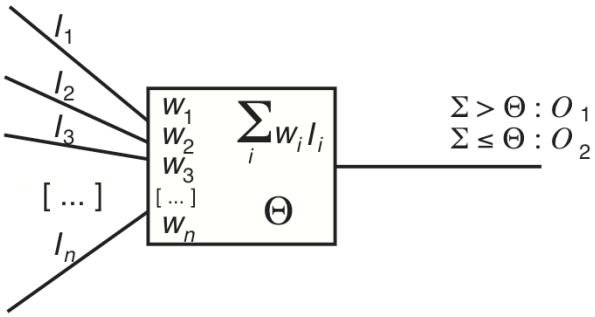
1. Solid statistical foundation
2. Efficient learning algorithms, learning can proceed from raw data
3. Unsupervised
4. Flexible

HMM limitations:

1. Large number of unstructured parameters
2. First order Markov models do not capture pairwise or higher order correlations

Neural Networks:
Universal functions, constructed from "neurons" that accept some real-valued input signal I, multiply input by some scaling factor w, sum over all scaled inputs, and produce a real-valued output O, according to whether the sum exceeds a threshold Q or not given an "activation function".

Scaling factors and thresholds are adjustable parameters. Neurons can be connected into multi-layered networks ...

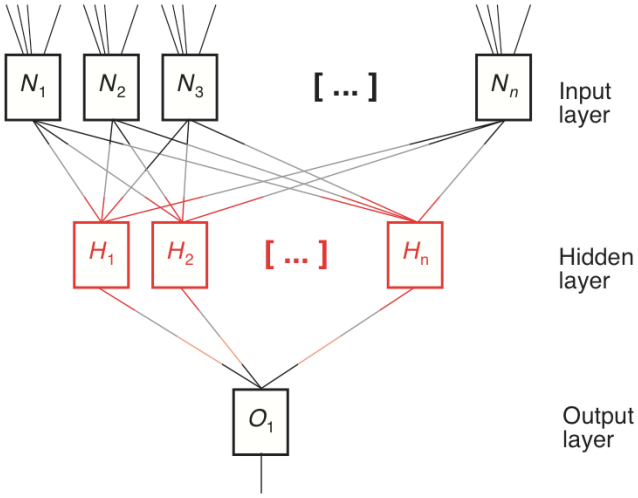


Each "neuron" contains:

- a set of inputs;
- a set of weights, one for each input which are optimized during training;
- an activation function;
- a threshold, also optimized during training.

In the sketch above, the activation function is linear, i.e. an "active" output depends on whether the weighted sum of inputs exceeds a hard threshold. Alternative activition functions can implement "soft" thresholds, e.g. logistic functions.

Neural networks can have "hidden" layers. Hidden layers collate and compare information from the input layers.



Build

1. Construct architecture
2. Define an encoding of input data that maps a property of the input into a real-valued function†

Train

3. Initialize neurons with random input weights and thresholds
4. Run training set and compare classification results
5. Use back propagation (compensation of output error) to update weights
6. Repeat until no further improvement is possible

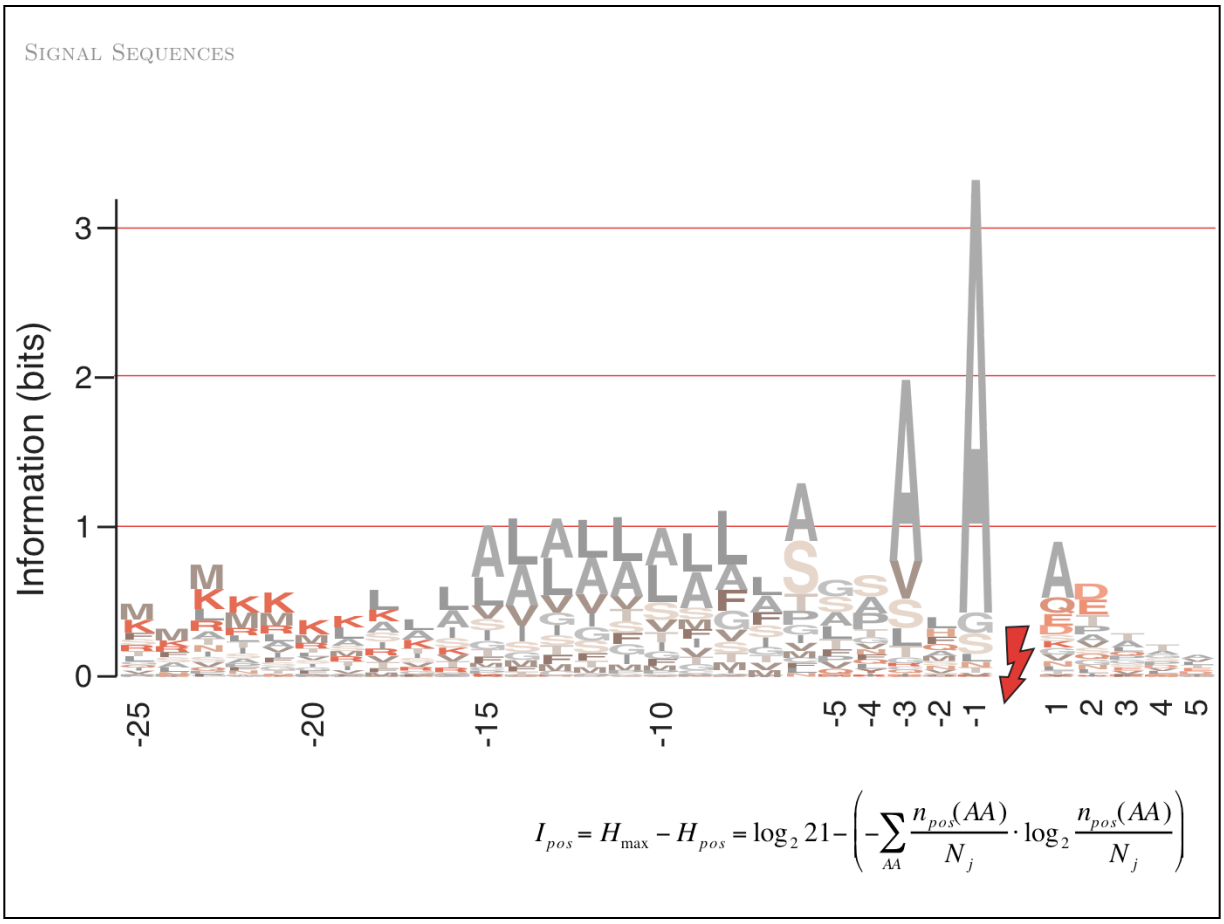
Use

7. Input observed sequence and record value of output: above/below threshold?

† encoding can be iteratively optimized as well as weights and thresholds!

- Disorder
- Signal peptides
- Secondary structure
- Transmembrane helices
- Domains
- Protein localization
- Phosphorylation sites

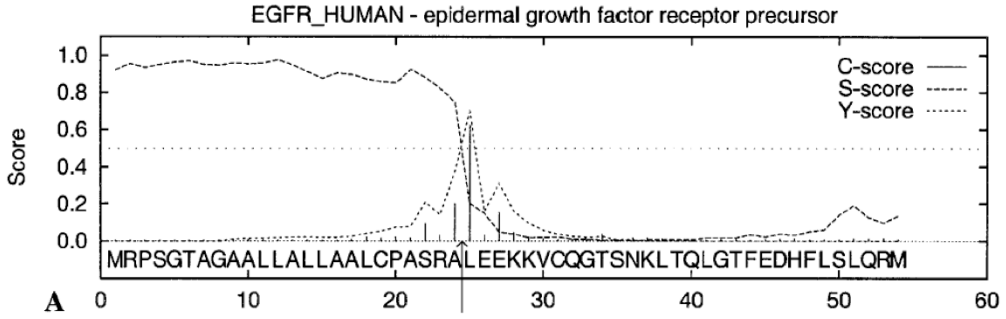
Other examples: Cystine knots, Zn-finger ...



Example for recognition of sequence features with HMMs or NNs: common features in gram-negative signal-peptide sequences are shown in a Sequence Logo.

Sequences were aligned on the signal-peptidase cleavage site. Their common features include a positively charged N-terminus, a hydrophobic helical stretch and a small residue that precedes the actual cleavage site.

Signal peptide detection is a successful application for both HMMs and NNs ...



http://www.cbs.dtu.dk/services/SignalP/

Nielsen H, Engelbrecht J, Brunak S & von Heijne G (1997) Identification of prokaryotic and eukaryotic signal peptides and prediction of their cleavage sites. *Prot Eng* 10:1–6

Two NNs: one for "signal sequence" (S-score), one for "cleavage site" (C-score).

Currently (V. 3.0) > 90% accurracy

SignalP is the premier Web server to detect signal sequences.

http://steipe.biochemistry.utoronto.ca/abc

B O R I S . S T E I P E @ U T O R O N T O . C A

DEPARTMENT OF BIOCHEMISTRY & DEPARTMENT OF MOLECULAR GENETICS
UNIVERSITY OF TORONTO, CANADA