Bioinformatics

Multiple Alignment, Patterns & Profiles

David Gilbert

Bioinformatics Research Centre

www.brc.dcs.gla.ac.uk

Department of Computing Science, University of Glasgow

Lecture summary

- Characterising *families* of sequences
- Multiple sequence alignment
- Weight matrices
- Searching for distant relatives: beyond Blast - PSI-Blast
- Patterns
- Pattern discovery
- Rating & using patterns

Multiple Sequence Alignment

- Why do MSA?
  - Help prediction of the secondary and tertiary structures of proteins of new sequences
  - Help to find motifs or signatures characteristic of protein family

```
VTISCTGSSSNIGAG-NHVKWYQQLPG
VTISCTGTSSNIGS--ITVNWYQQLPG
LRLSCSSSGFIFSS--YAMYWVRQAPG
LSLTCTVSGTSFDD--YYSTWVRQPPG
PEVTCVVVDVSHEDPQVKFNWYVDG--
ATLVCLISDFYPGA--VTVAWKADS--
AALGCLVKDYFPEP--VTVSWNSG---
VSLTCLVKGFYPSD--IAVEWWSNG--
```

MSA

```
VTISCTGSSSNIGAG-NHVKWYQQLPG
VTISCTGTSSNIGS--ITVNWYQQLPG
LRLSCSSSGFIFSS--YAMYWVRQAPG
LSLTCTVSGTSFDD--YYSTWVRQPPG
PEVTCVVVDVSHEDPQVKFNWYVDG--
ATLVCLISDFYPGA--VTVAWKADS--
AALGCLVKDYFPEP--VTVSWNSG---
VSLTCLVKGFYPSD--IAVEWWSNG--
```

- 8 fragments from immunoglobulin sequences
- alignment highlights
  - – conserved residues,
  - –conserved regions
  - –more sophisticated patterns, like the dominance of hydrophobic residues (V,L,I) at fragment positions 1 and 3.
  - — http://www.techfak.uni-bielefeld.de/bcd/Curric/MulAli

MSA

```
VTIS C TGSSSNIGAG-NHVK W YQ QLPG
VTIS C TGTSSNIGS--ITVN W YQ QLPG
LRLS C SSSGFIFSS--YAMY W VR QAPG
LSLT C TVSGTSFDD--YYST W VR QPPG
PEVT C VVVDVSHEDPQVKFN W YVDG--
ATLV C LISDFYPGA--VTVA W KADS--
AALG C LVKDYFPEP--VTVS W NSG---
VSLT C LVKGFYPSD--IAVE W WSNG--
```

- The alignment can also enable us to infer the evolutionary history of the sequences.
- It looks like the first 4 sequences and the last 4 sequences are derived from 2 different common ancestors, that in turn derived from a "root" ancestor.
- But true phylogentic analysis is more complex
- http://www.techfak.uni-bielefeld.de/bcd/Curric/MulAli

Multiple sequence aligment - methods

- Simultaneous: N-wise alignment (adapted from *pairwise* approach)
  - uses N-dimension dynamic programming matrix.
  - Complexity is for global alignment
    - $O(m_1 m_2)$ [2 sequences length m_1 & m_2]
    - $O(m^2)$ [2 sequences of length m]
    - $O(m^n)$ [n sequences of length m]
    - Ten sequences of length 1000 requires $1000^{10} = 10^{?}$
      - *Approximate age of universe in pico-seconds*
  - Combinatrial explosion!
  - Thus only good for short sequences.

- Manua1 (!)

- Heuristic…

Multiple sequence aligment - methods

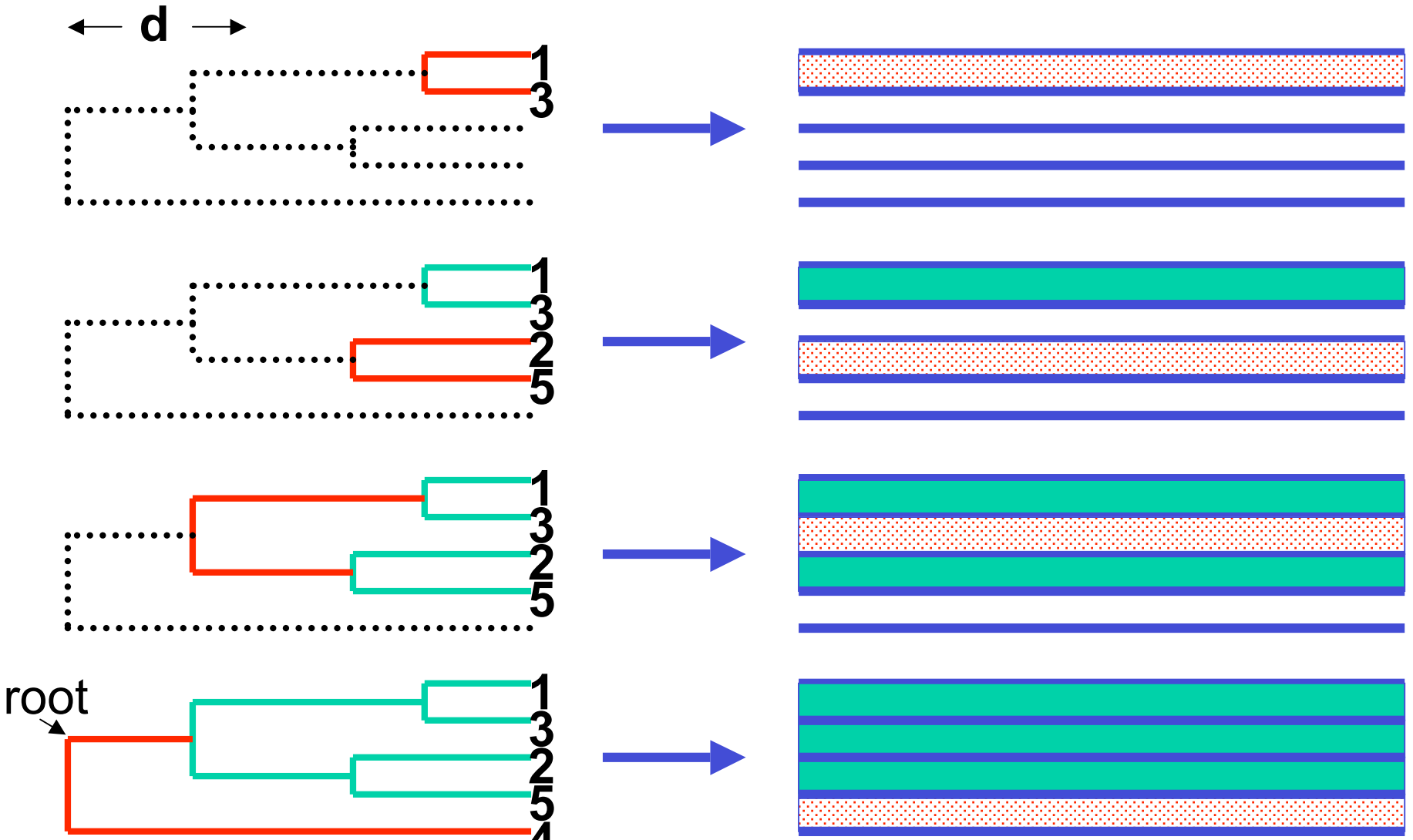
- Heuristic methods, e.g. Progessive -- ClustalW:
  - Split multiple alignment into pairwise alignments (?how?)
  - optimise locally – greedy – at each step

- Many possibilities as to how the sequence of (pairwise) alignments can be built
- Must attempt to minimise errors introduced in early alignments which will accumulate during the progressive alignment
- Can be achieved in part by aligning the MOST similar sequences in turn
- Employ a phylogenetic tree to 'guide' the progressive alignment
  - compute pairwise sequence identities
  - construct binary tree (can output phylogenetic tree)
  - align similar sequences in pairs, add distantly related ones later.
- No guarantee that the global optimum will be found
  - *But provides a computationally tractable and biologically useful algorithm*

Multiple Sequence Alignment

- Outline of CLUSTAL (Thomson et al 1994)
  - Calculate the pairwise similarity scores for the sequences
    - *Can use full dynamic programming approach*
  - Employing similarity score create a phylo tree (UPGMA)
  - From tree produce weights for each sequence
    - Based on similarities
      - High weighting to dissimilar sequences
      - Low weighting to similar sequences
    - Weighting used when combining alignments
  - Employing tree structure as a guide perform progressive pairwise alignments

Multiple Sequence Alignment



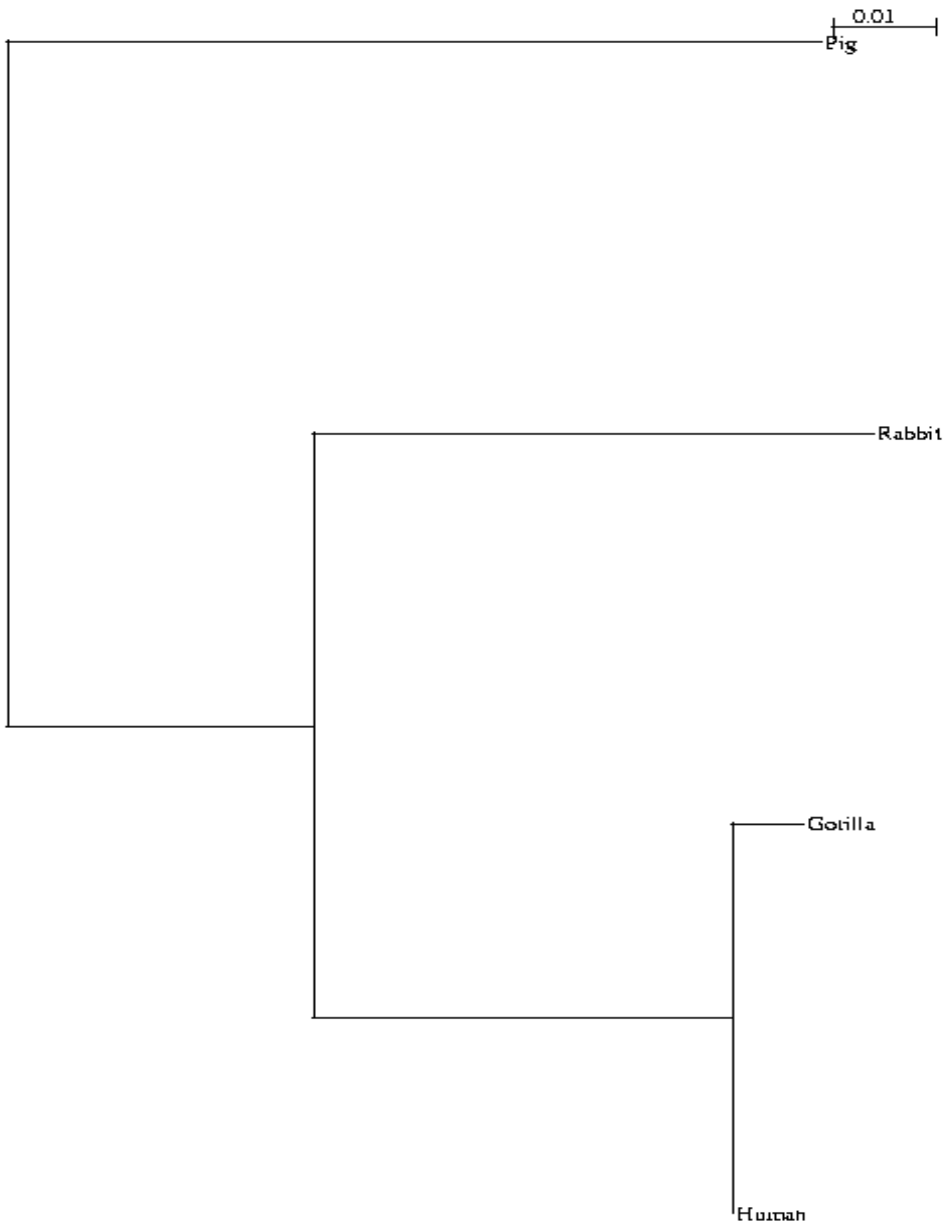
Multiple sequence alignment (globins)

```
CLUSTAL W (1.81) multiple sequence alignment


Human     VHLTPEEKSAVTALWGKVNVDEVGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMGNPKV 60
Gorilla   VHLTPEEKSAVTALWGKVNVDEVGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMGNPKV 60
Rabbit    VHLSSEEKSAVTALWGKVNVEEVGGEALGRLLVVYPWTQRFFESFGDLSSANAVMNNPKV 60
Pig       VHLSAEEKEAVLGLWGKVNVDEVGGEALGRLLVVYPWTQRFFESFGDLSNADAVMGNPKV 60
          ***:.***.** .*******:*****************************..:***.****

Human     KAHGKKVLGAFSDGLAHLDNLKGTFATLSELHCDKLHVDPENFRLLGNVLVCVLAHHFGK 120
Gorilla   KAHGKKVLGAFSDGLAHLDNLKGTFATLSELHCDKLHVDPENFKLLGNVLVCVLAHHFGK 120
Rabbit    KAHGKKVLAAFSEGLSHLDNLKGTFAKLSELHCDKLHVDPENFRLLGNVLVIVLSHHFGK 120
Pig       KAHGKKVLQSFSDGLKHLDNLKGTFAKLSELHCDQLHVDPENFRLLGNVIVVVLARRLGH 120
          ******** :**:** **********.*******:********:*****:* **::::*:

Human     EFTPPVQAAYQKVVAGVANALAHKYH 146
Gorilla   EFTPPVQAAYQKVVAGVANALAHKYH 146
Rabbit    EFTPQVQAAYQKVVAGVANALAHKYH 146
Pig       DFNPNVQAAFQKVVAGVANALAHKYH 146
          :*.* ****:*****************
```

Multiple sequence alignments & phylogenetic trees

Pair	Score
Human-Gorilla	99
Human-Rabbit	90
Gorilla-Rabbit	89
Human-Pig	84
Gorilla-Pig	84
Rabbit-Pig	83

```
((Human:0.00000,
Gorilla:0.00685) :0.04110,
Rabbit:0.05479,
Pig:0.10959);
```



Multiple alignments

- Analyse *gene families*
  - reveal (subtle) conserved family characteristics

		characters								
		1	2	3	4	5	6	7	8	9	10
sequences	S1	Y	D	G	G	A	V	-	E	A	L
	S2	Y	D	G	G	-	-	-	E	A	L
	S3	F	E	G	G	I	L	V	E	A	L
	S4	F	D	-	G	I	L	V	Q	A	V
	S5	Y	E	G	G	A	V	V	Q	A	L
consensus		**y**	**d**	**G**	**G**	**AI**	**VL**	**V**	**e**	**A**	**l**

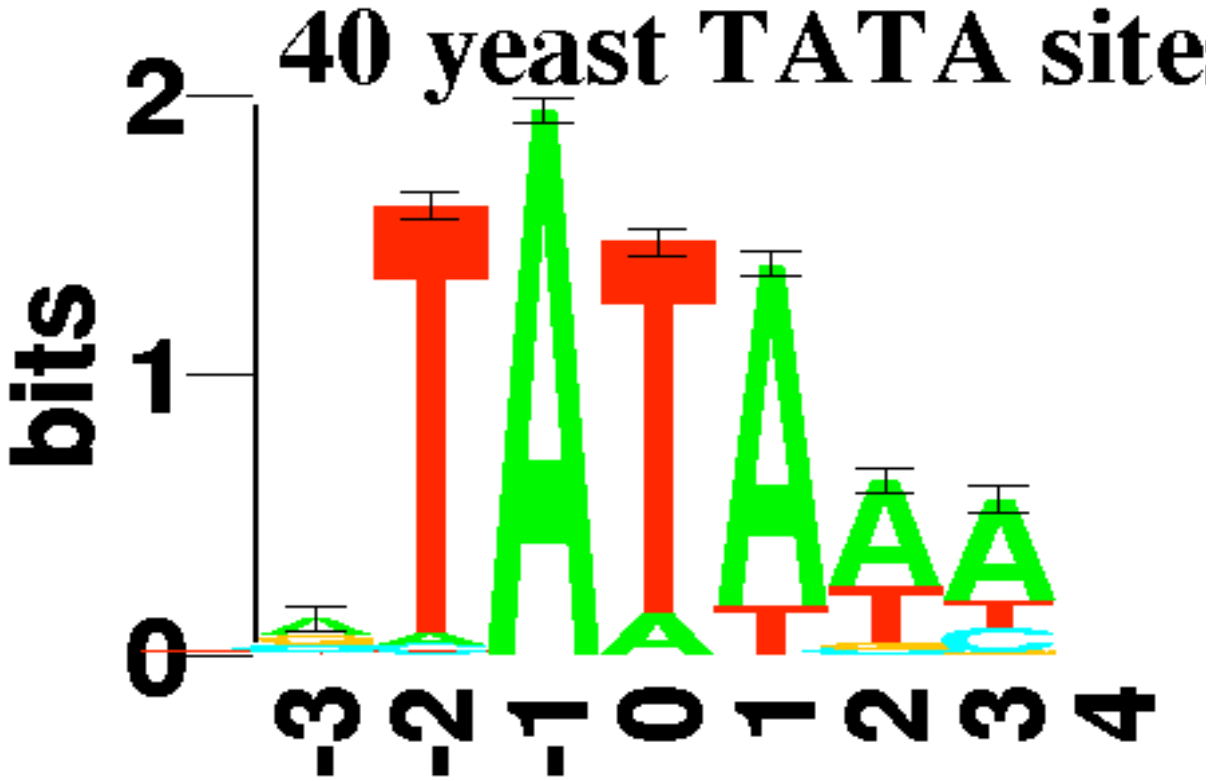
Profile (frequency matrix)

characters

sequences	1	2	3	4	5	6	7	8	9	10
S1	Y	D	G	G	A	V	-	E	A	L
S2	Y	D	G	G	-	-	-	E	A	L
S3	F	E	G	G	I	L	V	E	A	L
S4	F	D	-	G	I	L	V	Q	A	V
S5	Y	E	G	G	A	V	V	Q	A	L
	y	**d**	**G**	**G**	**AI**	**VL**	**V**	**e**	**A**	**l**
	Y=.6	**D=.6**	**G=1**	**G=1**	**A=.5**	**V=.5**	**V=1**	**E=.6**	**A=1**	**L=.8**
	F=.4	**D=.4**			**I=.5**	**L=.5**		**Q=.4**		**V=.2**

(Can further weight the profile using PAM or BLOSUM matrices)

Sequence logos



A graphic representation of an aligned set of binding sites. A logo displays the frequencies of bases at each position, as the relative heights of letters, along with the degree of sequence conservation as the total height of a stack of letters, measured in bits of information. Subtle frequencies are not lost in the final product as they would be in a consensus sequence

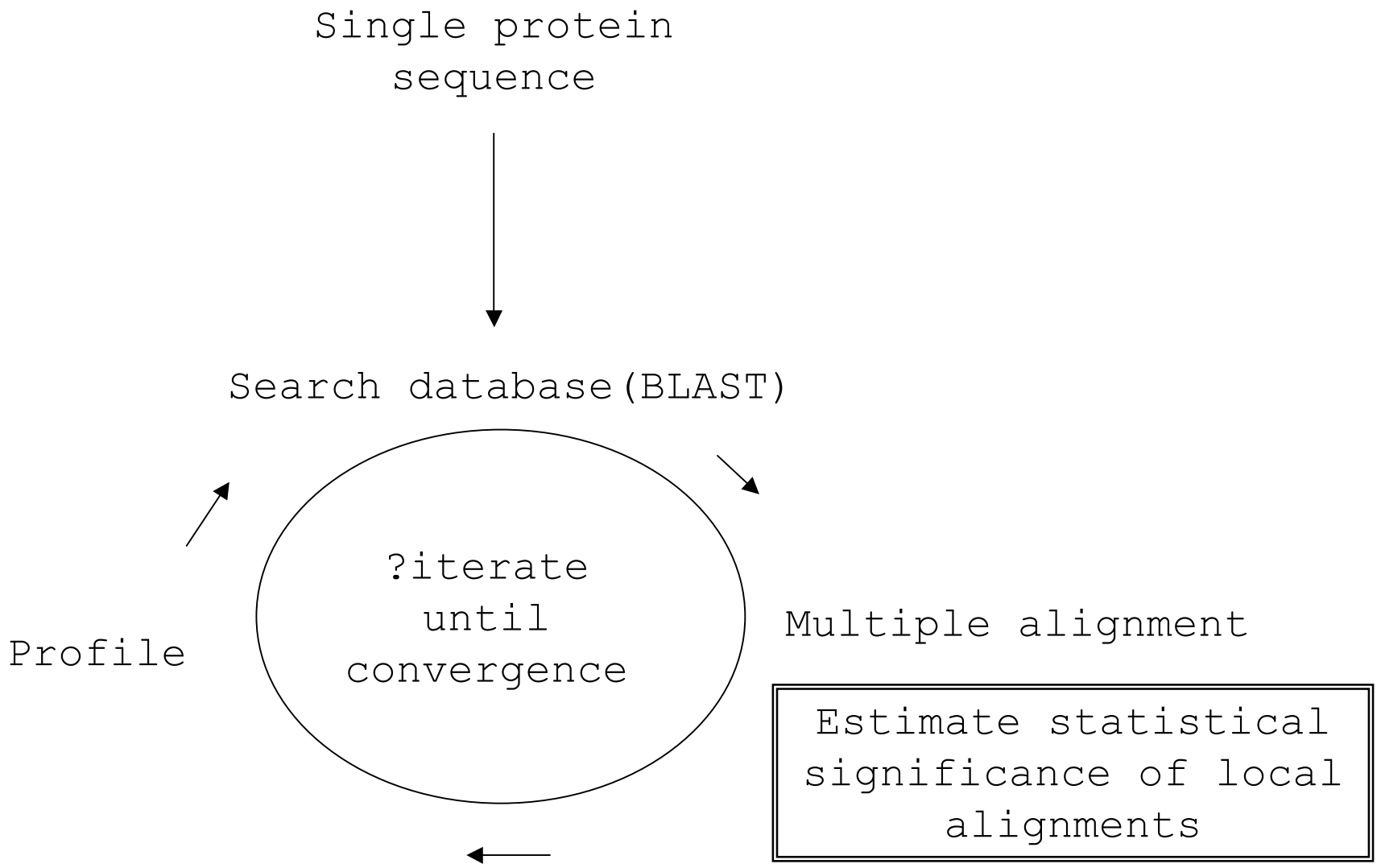
What can we do with multiple alignments?

- Create (databases of) profiles derived from multiple alignments for protein families
  - profile = multiple alignment + observed character frequencies at each position

- Search with a sequence against a database of profiles
  (e.g. **PROSITE** database)
  - faster than sequence against sequence
  - gives a more general result (“the input sequence matches globin profile”)

- Search with a profile against a database of sequences
  - PSI-BLAST : can identify more distant relationships than by normal BLAST search

PSI-BLAST (position specific iterated BLAST)



PSI-BLAST (Altschul et al 1997)

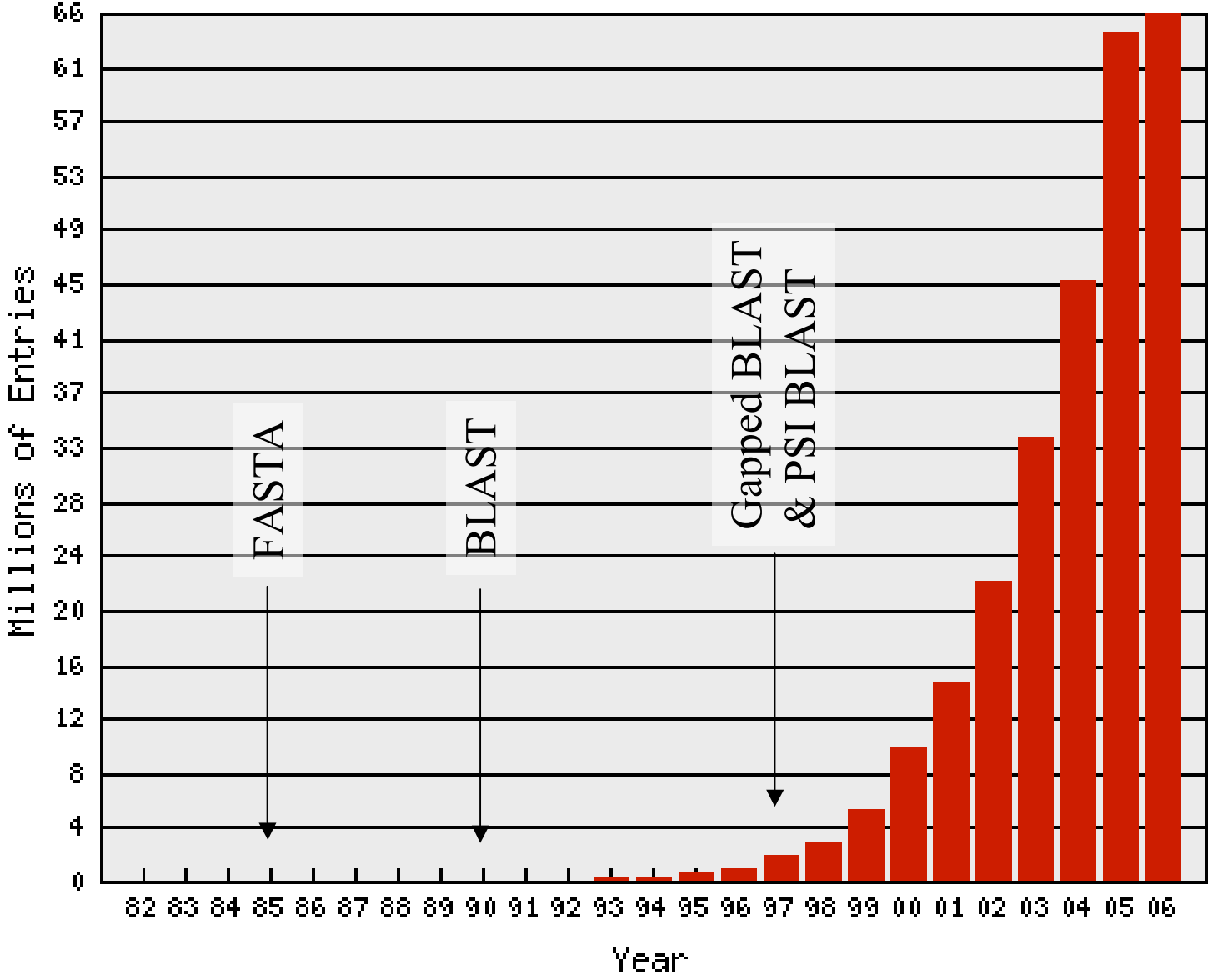
(1) Start with 1 sequence (or profile) = ‘probe’

(2) Search with BLAST and select top hits manually or automatically

(3) Make multiple alignment & profile

(4) Estimate statistical significance of local alignments.
If significance ok & you want to continue, then go to (1) using the profile, else **exit**

Dates & programs



Patterns and alternative representations

- Patterns
  - unions of patterns
  - decision trees
  - exact/approximate matching
- Alignments, weight matrices, profiles, HMMs, Neural networks, SCFGA, ...

Brazma et al, Approaches to the automatic discovery of patterns in biosequences, Journal of Computational Biology, 5(2):277-303, 1998

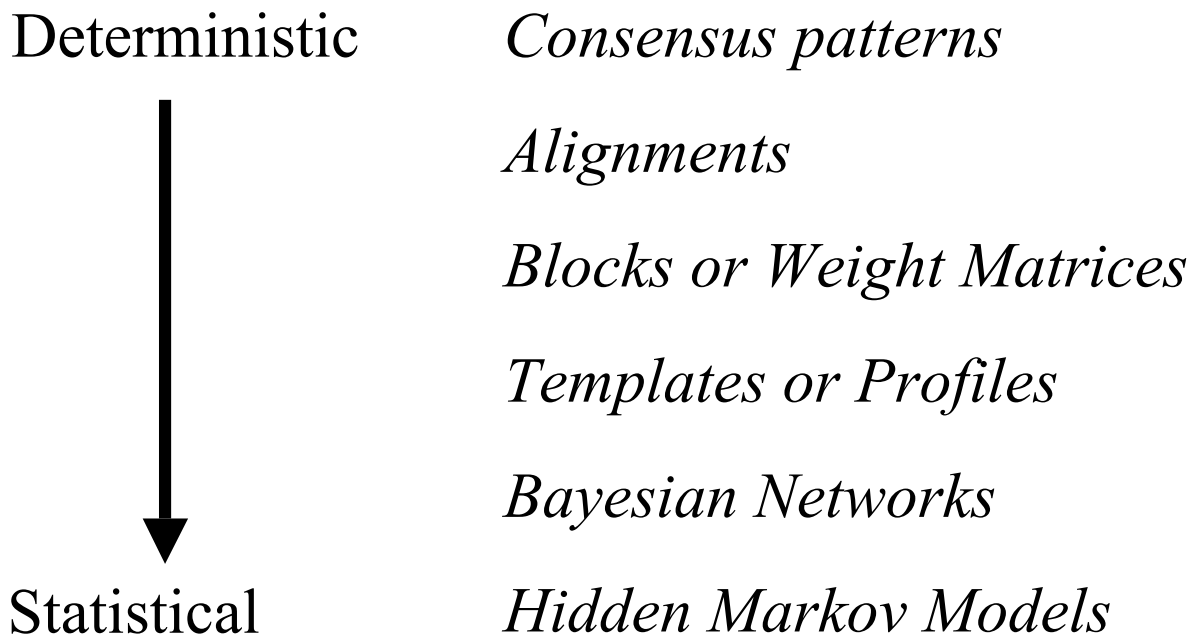
Some terminology

Common similarities between sequences/structures:

- pattern, motif, fingerprint, template, fragment, core, site, alignment, weight matrix, profile…
- "Pattern": description of structure properties
  - (*Deterministic*) Decide if a protein matches it or not
  - (*Probabilistic*) Assign a value to the match
- **"Motif" - pattern with biological meaning**

Adapted from: Eidhammer, Jonassen & Taylor, "Structure Comparison and Structure Patterns", JCB, 7:5 pp 685-716, 2000.

Classification of functions



Discrete patterns

- **Advantages**
  - simple and easily interpretable objects
  - easier to discover from scratch (i.e., if no additional information to sequences are given), particularly in noisy data
- **Disadvantages**
  - limited descriptive power (no weights can be attributed to alternatives)

Regular expressions

- **Symbol**: for each symbol **a** in the alphabet of the language, the regular expression **a** denotes the language containing just the string `a`
- **Alternation**: Given 2 regular expressions M and N then M | N is a new regex. A string is in lang(M|N) if it is lang(M) or lang(N). The lang(**a|b**) = {a,b} contains the 2 strings `a` and `b`.
- **Concatenation**: Given 2 regexes M and N then M•N is a new regex. A string is in lang(M•N) if it is the concatenation of 2 strings α and β s.t. α in lang(M) and β in lang(N). Thus regex **(a|b)•a** = {aa,ba} defines the language containing the 2 strings `aa` and `ba`

Regular expression notation

a	ordinary character, stands for itself
ε	the empty string
	another way to write the empty string!
M | *N*	alternation
M • *N*	concatenation
*M**	repetition (zero or more times)
M+	repetition (one or more times)
M?	Optional, zero or one occurrence of M
[a-zA-Z]	Character set alternation
.	Period stands for any single character except newline
`"a.+*"`	quotation, string stands for itself

Biosequences - general

- Basic alphabet
  Σ = { a, t/u, c, g} (DNA/RNA)
  Σ = {A, C, .., Y} (Protein sequence)
- Character group alphabet $\Pi = \{g_1 \ldots g_n\}$
  (e.g. amino-acid class)
- Wild card X = $\{ x(n_1, n_2) \mid n_1 < n_2 \in N\}$
- V(x(c_1,c_2)) set of all words over Σ of length between c_1 and c_2
- Pattern P = $p_1 \ldots p_n$, $p_i \in \Sigma \cup \Pi \cup X$

→ character & position constraints ←

Pattern notation and matching

- Separate the pattern alphabet characters by a dash "-"
- Pattern

$$P = \text{A-x(2,6)-[LI]-x(0,}\infty\text{)}$$

matches string

$$S = \text{ACDEFLGHJKL}$$

because

$$\text{S} = \text{A} \bullet \text{CDEF} \bullet \text{L} \bullet \text{GHJKL}$$

($\bullet$ meaning concatenation) and

$$\text{A} \in \text{V(A)},\ \text{CDEF} \in \text{V(x(2,6))},\ \text{L} \in \text{V([LI])},\ \text{GHJKL} \in \text{V(x(0,}\infty\text{))}$$

PROSITE patterns

- Database of protein families and domains
- Consists of biologically significant sites, patterns and profiles that help to reliably identify to which known protein family (if any) a new sequence belongs

- `**x**' any amino acid
- Ambiguities :

  [ALT] =Ala or Leu or Thr

  {AM} any amino acid except Ala and Met.
- `-' separator, `<` N-terminal, `>` C-terminal
- `.` end of pattern
- Repetition: x(3) = x-x-x
- x(2,4) = x-x *or* x-x-x *or* x-x-x-x.

PROSITE examples

- [AC]-x-V-x(4)-{ED}.
  - [Ala or Cys]-x-Val-x-x-x-x-{any but Glu or Asp}

- <A-x-[ST](2)-x(0,1)-V.
  - Start at N-terminal of the sequence
  - Ala-x-[Ser or Thr]-[Ser or Thr]-(x or none)-Val

How to obtain these patterns?

Example property

A given sequence belongs to the chromo-domain family if it matches either the pattern:

E-x(0,1)-E-E-[FY]-x-V-E-K-[IV]-[IL]-D-[KR]-R-x(3,4)-G-x-V-x-Y-x-L-K-W-K-G-[FY]-x-[ED]-x-[HED]-N-T-W-E-P-x(2)-N-x-[ED]-C-x-[ED]-L-[IL]

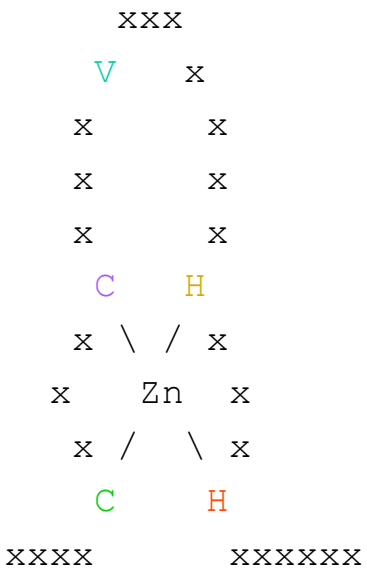
or the pattern:

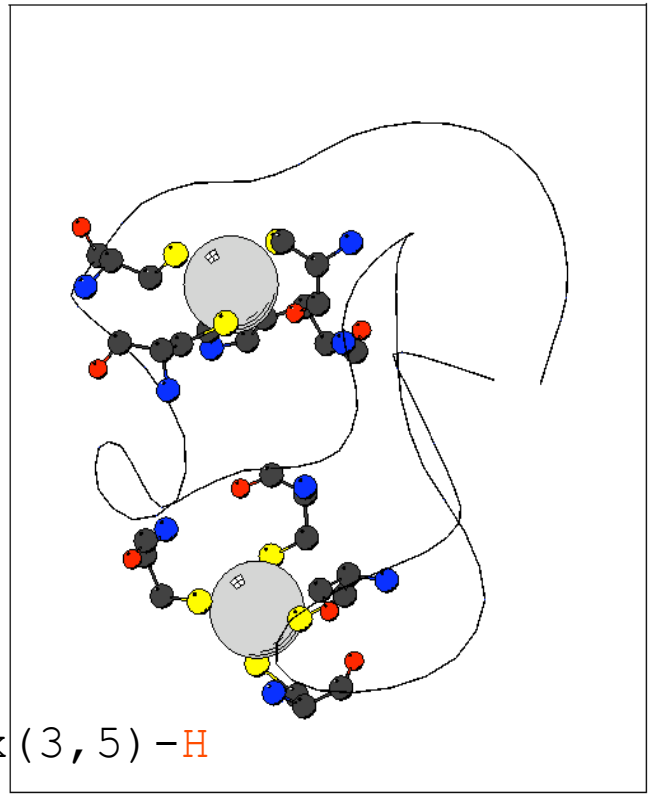
L-x(2,3)-E-[KR]-I-[IL]-G-A-[TS]-D-[TSN]-x-G-[EDR]-L-x-F-L-x(2)-[FW]-[KE]-x(2)-D-x-A-[ED]-x-V-x-[AS]-x(2)-A-x(2)-K-x-P-x(2)-[IV]-I-x-F-Y-E

or the pattern:

Y-x(0,2)-L-[IV]-K-W-x(6)-[HE]-x-[TS]-W-E-x(4)-[IL]

Example family (zinc finger c2h2)





C-x(2,4)-C-x(3)-[ILVMFYWC]-x(8)-H-x(3,5)-H

RNA structural patterns

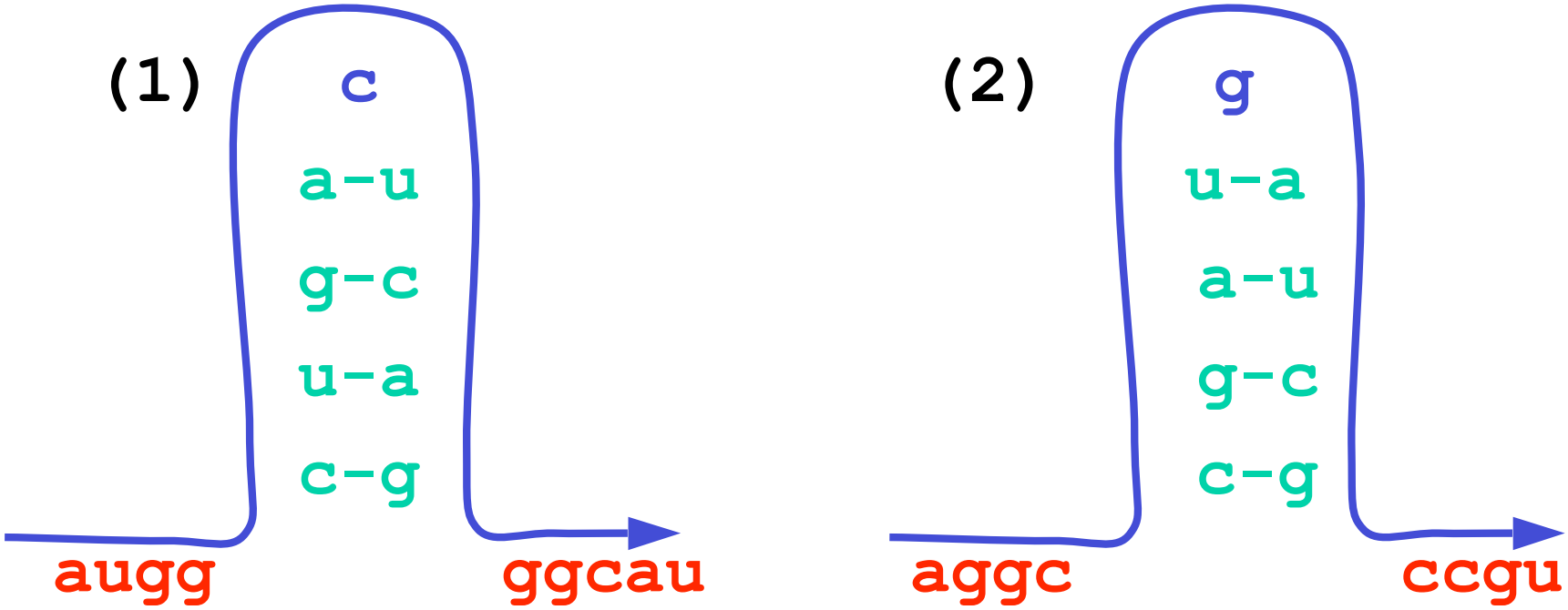
Eidhammer, Jonassen, Grinhang, Gilbert & Ratnayake,
A contraint-based structure description language for biosequences,
Journal of Constraints 6:2/3, 2001

- Constraints:
  - string length
  - inter-string distance
  - character contents
  - matching positions
  - correlation (identical, reverse, complement).
- Complements **`a-u g-c, g-u`** (weaker)
- Structures: Stem-loops, Pseudo-knots, Clover leafs
- Context free grammar

Possible patterns

- Tandem repeat $\alpha-\alpha$ acg acg
- Simple repeat $\alpha-\beta-\alpha$ acgaaaacg
- Multiple repeat $\alpha-\beta-\alpha-\delta-\alpha$
  acgaaacguuacg
- Palindrome $\alpha-\alpha^{r}$ acg gca
- Stem loop $\alpha-\beta-\alpha^{rc}$ acgaacgu
- Pseudoknot $\alpha-\gamma_1-\beta-\gamma_2-\alpha^{rc}-\gamma_3-\beta^{rc}$
  auggcugaaggccgaucucagggcauaucgccgu

Stem loops

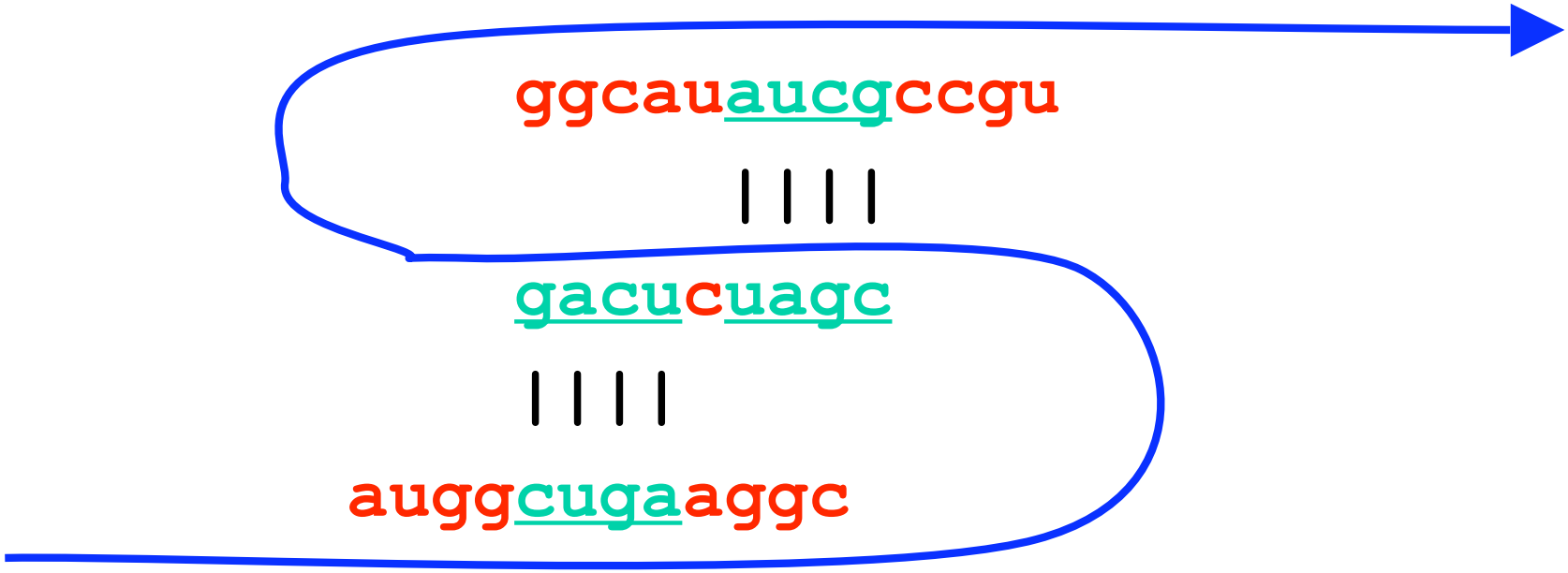


(1) auggcugacucagggcau

(2) aggccgaugaucgccgu

α β α^{rc}

Pseudo-knot



String:

auggcugaaggccgaucucagggcauaucgccgu

$\alpha \quad \gamma_1 \quad \beta \quad \gamma_2 \; \alpha^{rc} \quad \gamma_3 \quad \beta^{rc}$

Various ways of using pattern matching for family characterization

A sequence belongs to the family if

1. it matches the given sequence *pattern*;
2. if it is within a certain *distance* from a string that matches a the pattern
   (distance between strings can be defined either as a number of mismatches, or as an edit-distance, or based on similarity matrices or some other way) ;
3. if it matches one of a given set of patterns (i.e.,if it matches a union of patterns);
4. if a decision-tree over the matching patterns returns “yes”

Learning

- Automatically find pattern (given a training set)
- Characterisation: (positive examples only) patterns describing "interesting" properties of a family
- Classification: (positive **and** negative examples) pattern distinguishing S+ and S- .. Which may overlap...

  - Formal language for descriptions
  - Scoring function to rate descriptions
  - Algorithm

Pattern discovery in biosequences

- Motivation:
  - gene functional class prediction
  - RNA splicing
  - protein structure & function
  - gene regulation (transcription factor binding site prediction)
  - detection of repeats

- Prediction of structure/function from sequence:
  - sequence database similarity search
  - compare to family descriptions
  - structure prediction programs

[Alvis Brazma & Inge Jonnassen]

Pattern discovery in biosequences

- Group together sequences thought to have common biological (structural, functional) properties -> families
  (biological - semantic level)
- Study the purely syntactic properties common to these sequences ignoring their biological (semantic) properties -> patterns, clusters
  (mathematical - syntactic level)
- Test whether the discovered patterns make sense (back to semantic level)

Protein family analysis

- Collect sequences (structures) in family
- Analyze
  - local multiple alignment
  - global multiple alignment
  - pattern discovery
- Make family description
- Pick up more family members?
  - Analyze extended set

Pattern discovery (machine learning)

- Languages & associated discovery mechanisms
- Strings - much work
- Finding gene expression sites in DNA may require context sensitive patterns.
- Structures

Approaches to pattern discovery

- **Pattern driven:**

  enumerate all (or some) patterns up to certain complexity (length), for each calculate the score, and report the best

- **Sequence driven:**

  look for patterns by aligning the given sequences

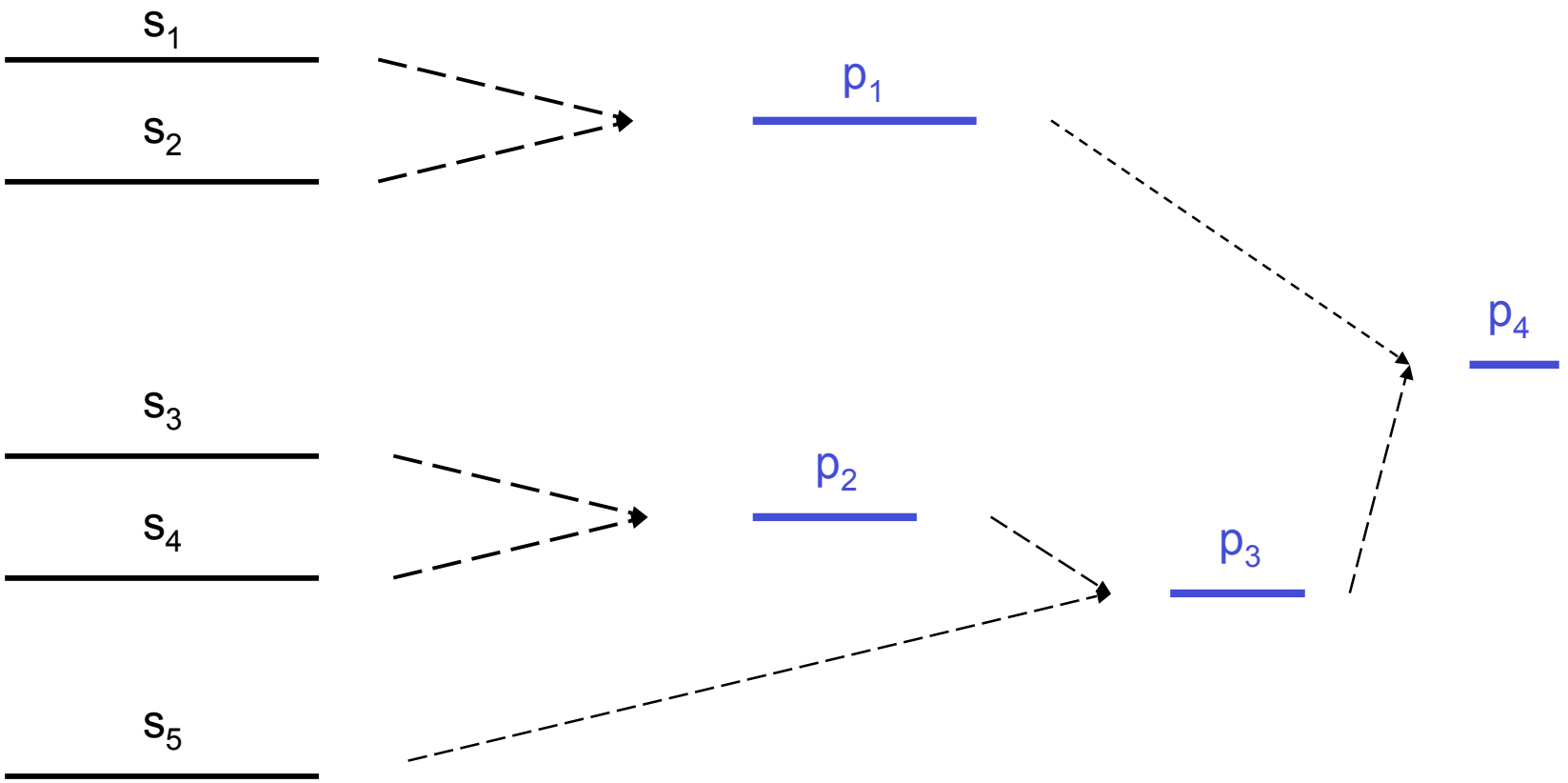
Pattern driven algorithms

- Brute force - enumerate all patterns (for instance, all substrings) up to a given length (complexity)
- Evaluate their fitness with respect to the input sequences and output the best
- Unrealistic for patterns of even modest size even for substring patterns (e.g., for substring patterns of length 10 over the amino acid alphabet, there are more than 10^{13} different substrings to enumerate in this way)

Sequence driven algorithms

- Group similar sequences together (e.g., in pairs);
- For each group find a common pattern (e.g., by dynamic programming);
- Group similar patterns together and repeat the previous step until there is only one group left

Sequence driven approach



Algorithm for string pattern discovery

- Design (a naive) algorithm for a simple language ***s*** where s $\in \Sigma^*$ and * is a wild card of arbitrary length, i.e. x(0,inf)

Example: s_1 = **TAWCEFGOPA**
s_2 = **FGOPAAWCES**
s_3 = **WUVTAWCESAW**

Try discovering patterns using pattern-driven & sequence-driven approaches
Sequence-driven: P(s) == set of patterns for s
P(s1) = {s1}, P(s2) = {s2}, P(s3) = {s3}
P(s1,s2) = {...}, P(s1,s2,s3) = {...}

Amino acid residue groups

Residue property	Residue groups	
Small	Ala, Gly	A,G
Small hydroxyl	Ser, Thr	S,T
Basic	Lys, Arg	K,R
Aromatic	Phe, Tyr, Trp	F,Y,W
Basic+	His, Lys, Arg	H,K,R
Small hydrophobic	Val, Leu, Ile	V,L,I
Medium hydrophobic	Val, Leu, Ile, Met	V,L,I,M
Acidic/amide	Asp, Glu, Asn, Gln	D,E,N,Q
Small/polar	Ala, Gly, Ser, Thr, Pro	A,G,S,T,P

Deriving regular expressions

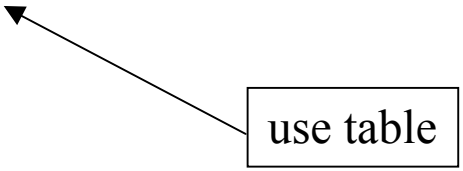
s_1 = ALDGAVFALCDRYFQ

s_2 = SDVGPRSCFCERFYQ

s_3 = ADLGRTQNRCDRYYQ

s_4 = ADIGQPHSLCERYFQ

Make a regular expression & a ‘fuzzy’ regular expression!



Rating patterns

- **Size** (e.g. number of characters…).
  - Hence **Information content:** e.g. length of the pattern (& perhaps penalties for wild cards)

- **Compression**
  - measure of how much of each of the items in the learning set is described

- **Sensitivity**, **Specificity** etc
  - requires evaluation against learning [training] & test sets

Compression - see updated slides

Send the pattern once, and then for each item, send the unmatched parts

(1) Raw Compression (chars k):

$$C_{raw} = (\sum_{i \in 1..n} N(k_i)) - (n-1)*N(k_p)$$

sum of chars in the examples minus (No_examples - 1) * chars_in_pattern

Varies from ? to ?

(2) Normalised compression:

$$C_{norm} = 1 - ((\sum_{i \in 1..n} N(k_i)) - C_{raw}) / ((\sum_{i \in 1..n} N(k_i)) - \min(N(k_i)))$$

This is a goodness of compression measure (0=good to 1=bad).

Compression

Send the pattern once, and then for each item, send the unmatched parts

(1) *Raw Compression*:

$$C_{raw} = \left(\sum_{i=1}^{n} |S_i| \right) - (n-1) * |P|$$

i.e. SumOfElementsInExamples - (NumberOfExamples - 1) * elements in pattern

(2) *Normalised compression*:

$$C_{norm} = \frac{\left(\sum_{i=1}^{n} |S_i| \right) - C_{raw}}{\left(\sum_{i=1}^{n} |S_i| \right) - \min_{i=1}^{n} \left(|S_i| \right)}$$

This is a goodness measure (1=good, 0=bad).

More compression

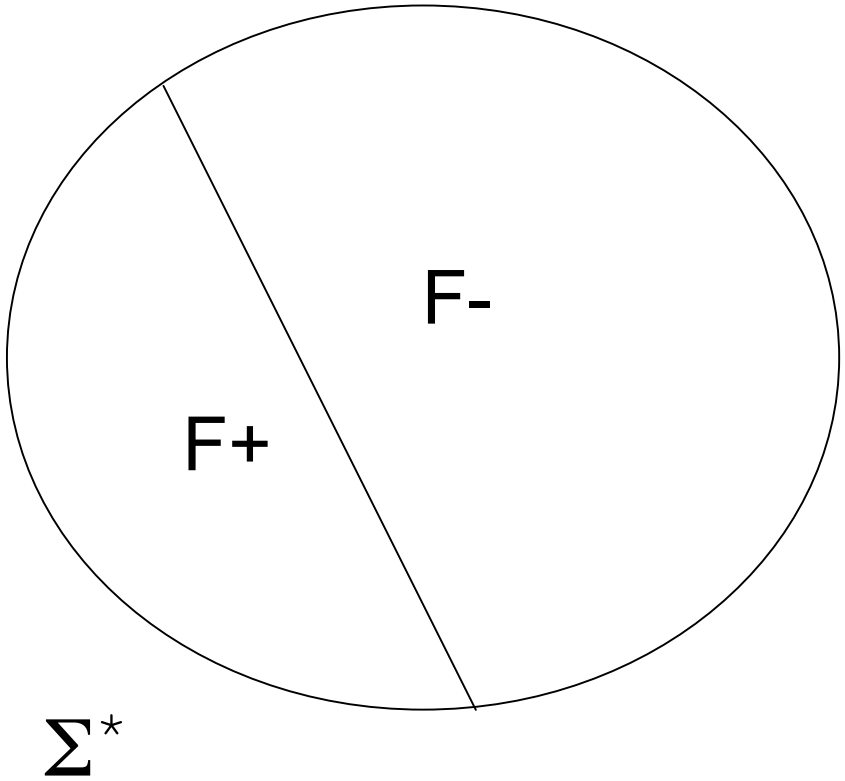
(3) Substituting (1) into (2):

$$C_{norm} = \frac{(n-1) * |P|}{\left(\sum_{i=1}^{n} |S_i|\right) - \min_{i=1}^{n}(|S_i|)}$$

(4) Pairwise comparison via compression:

$$Comp(S_1, S_2) = \frac{|P|}{\max(|S_1|, |S_2|)}$$

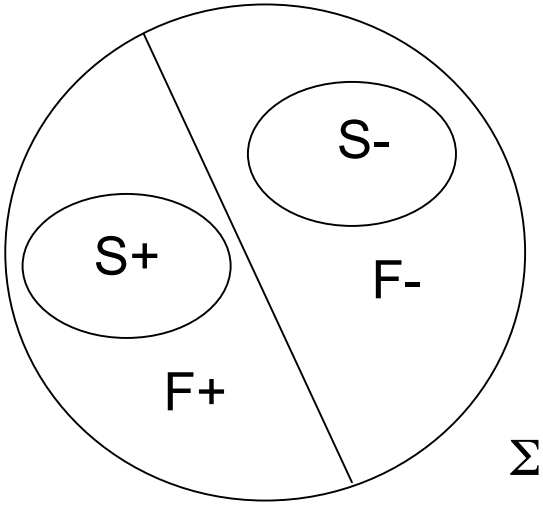
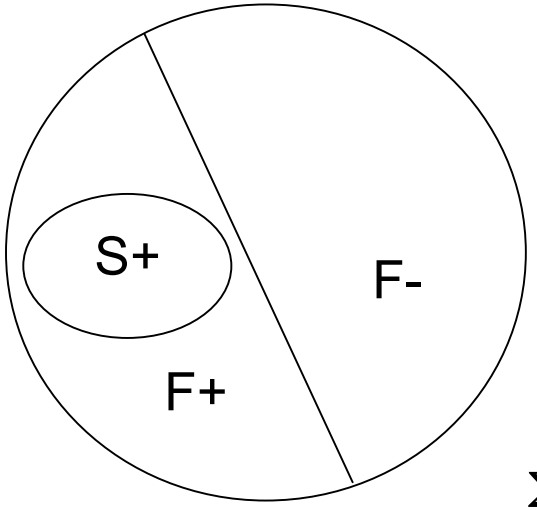
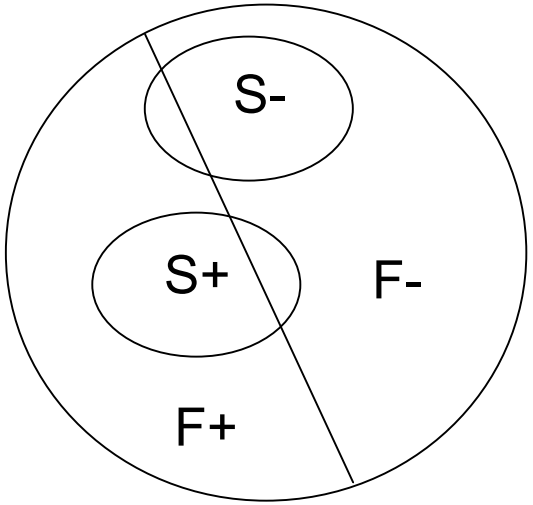
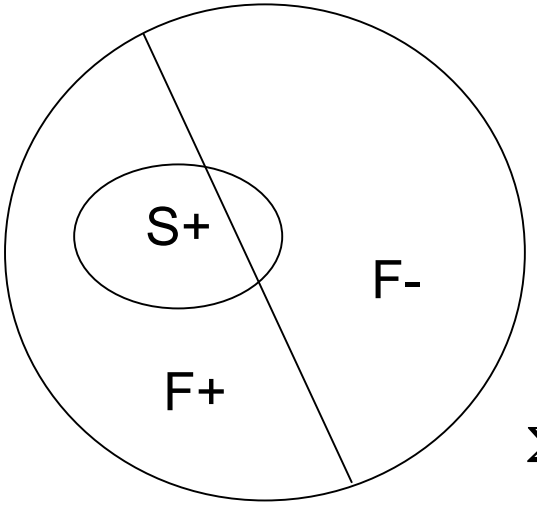
Characteristic string function for family F+



function $g : \Sigma^* \rightarrow \{FALSE, TRUE\}$

$$g(s) = \begin{cases} TRUE \text{ if } s \in F+ \\ FALSE \text{ if } s \in F- \end{cases}$$

Classification & conservation problems

Classification: + and - examples	Characterisation: + examples only
clean training data 	 clean training data
noisy training data 	 noisy training data

Classification problem C1

- Given a set S+ of sequences believed to be members of family F+, and a set S- of sequences believed not to be members, i.e.

  $S+ \subset F+$ and $S- \subset F-$

  $F+ \cap F- = \varnothing$ and $F+ \cup F- = \Sigma^*$
- Find *compact* string functions that return
  - TRUE for all $s \in S+$ and FALSE for all $s \in S-$, and
  - have a high likelihood for returning TRUE for $s \in F+$ and FALSE for $s \in F-$
- C1a: find compact "explanations" of known sequences
- C1b: try to predict the family relationship of yet unknown sequences
- N1: suppose $F+ \cap F- = \varnothing$ and $F+ \cup F- = \Sigma^*$, and $S+ \cap F-$ and $S- \cap F+$ are small, find *compact* string functions that return
  - TRUE for *most* $s \in S+$ and FALSE for *most* $s \in S-$, and
  - have a high likelihood for returning TRUE for $s \in F+$ and FALSE for $s \in F-$

Characterisation: conservation problem C2

- Given a set S+ of sequences believed to be members of family F+, i.e. $S+ \subset F+$
- Find *interesting* string functions that return
  - TRUE for all $s \in S+$
  - have a high likelihood for returning TRUE for $s \in F+$

- N2: suppose $F+ \subset \Sigma^*$, and given $S+ \subset \Sigma^*$, such that $S+ \cap (F+)^-$ is small, find *interesting* string functions that return
  - TRUE for *most* $s \in S+$, and
  - have a high likelihood for returning TRUE for $s \in F+$
- *Interesting*: have a low probability for returning TRUE for random sequences

Training and test sets

- *training* set of

  S+ positive examples from F+, and

  optionally a set S- of negative examples from F-

- *test* set

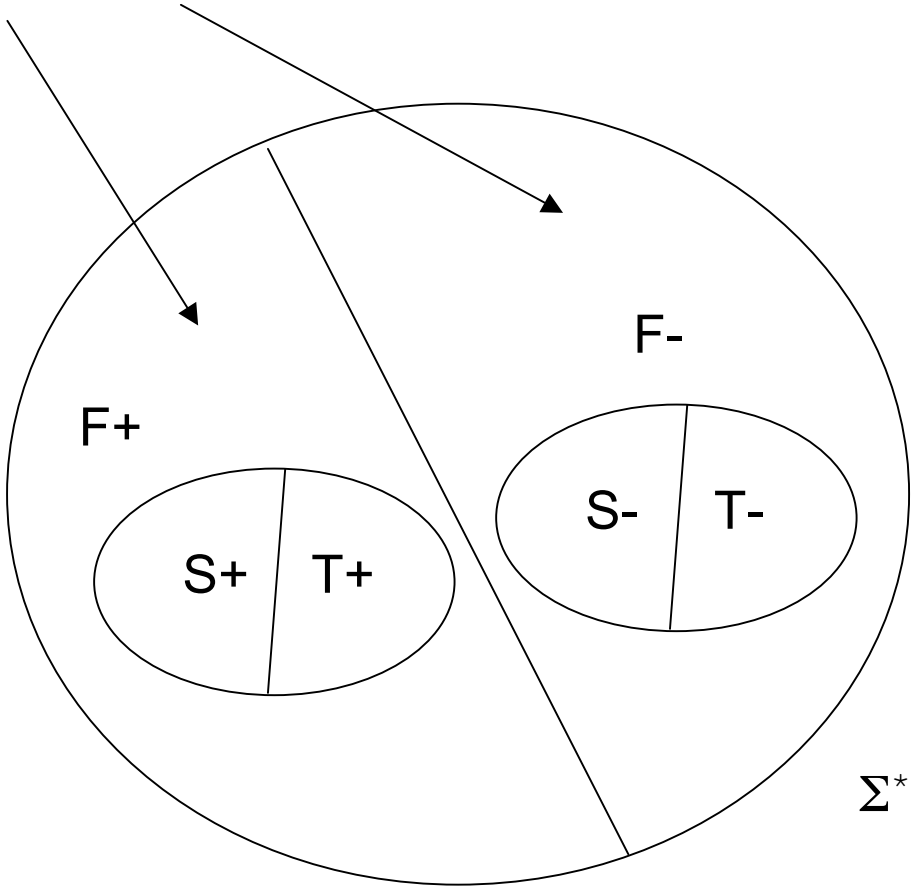
  T+ from F+ where $T+ \cap S+ = \varnothing$, and

  optionally T- from F- where $T- \cap S- = \varnothing$

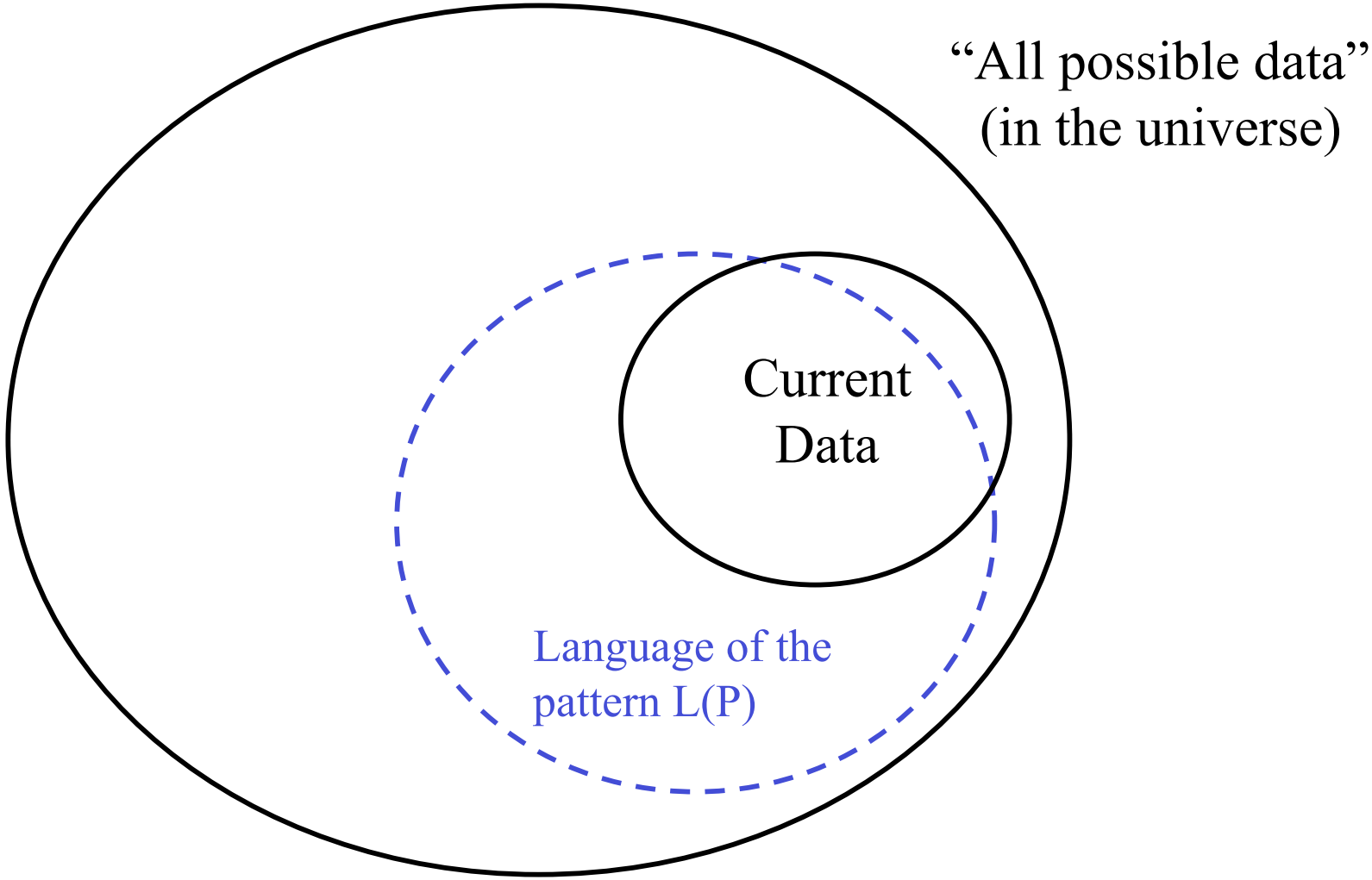
- In practice, we may not know all members of F+ and F-
  - Thus to construct training & test sets, we can randomly divide an initial set of positive examples into a training set S+ and a test set T+ , similarly for S- and T-
  - The goal is to accurately describe "new" members of F+ and F- when we come across them

Training and test sets

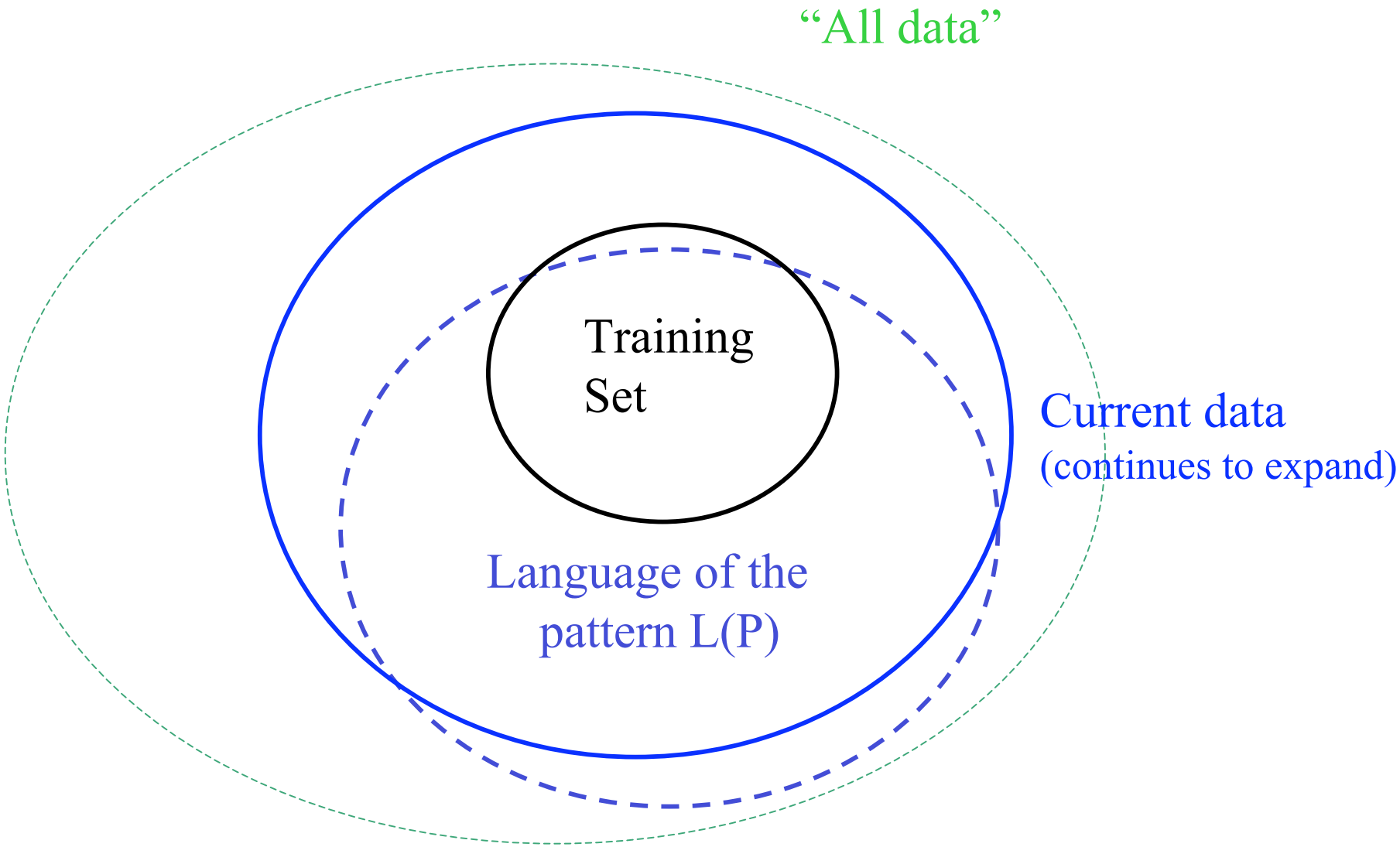
As yet not met sequences



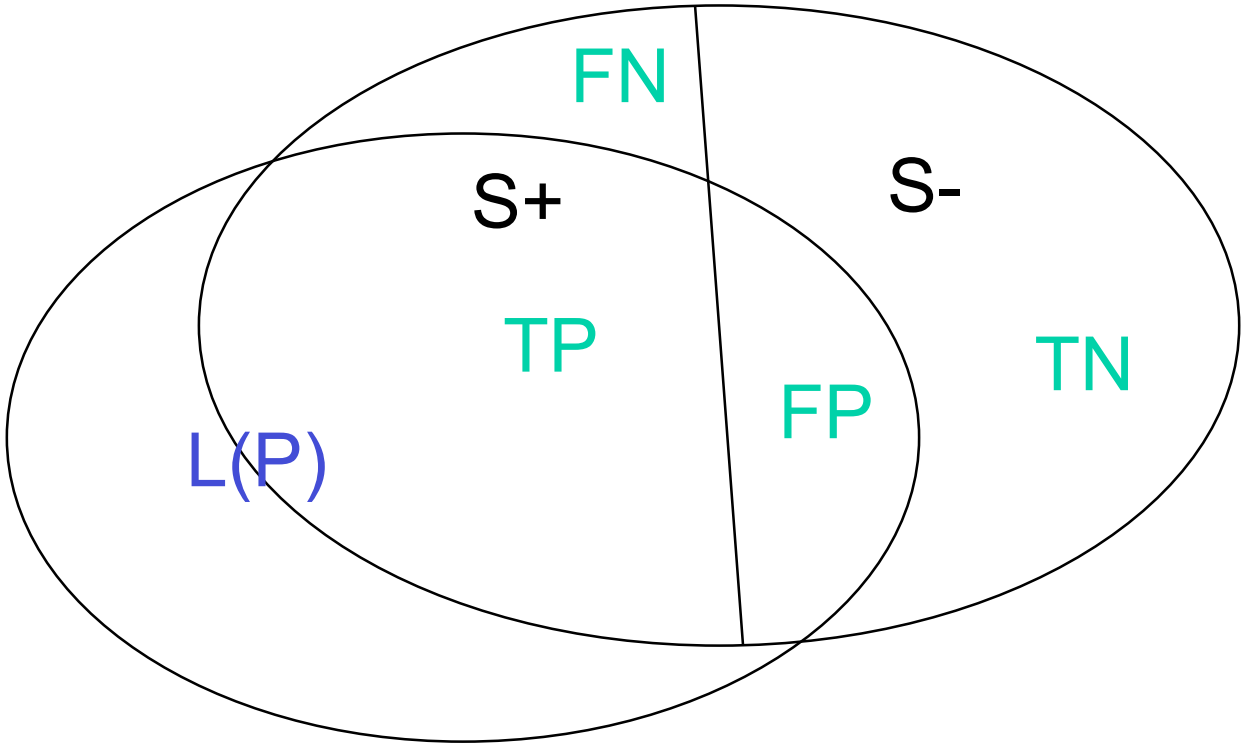
Goal


"All possible data"
(in the universe)
Current
Data
Language of the
pattern L(P)

The challenge of increasing data



True positives, true negatives, false positives, false negatives



TP - true pos
TN - true neg
FP - false pos
FN - false neg

$TP = L(P) \cap S+$

$TN = \neg L(P) \cap S-$

$FP = L(P) \cap S-$

$FN = \neg L(P) \cap S+$

L(P) - the set of sequences matched by the pattern P

Statistical Evaluation

TP - true pos TN - true neg
FP - false pos FN - false neg

Sensitivity (*Recall)*

$$Sn = \frac{TP}{TP + FN}$$

$0 \le Sn \le 1$

Specificity

$$Sp = \frac{TN}{TN + FP}$$

$0 \le Sp \le 1$

Positive Predictive Value (*Precision)*

$$PPV = \frac{TP}{TP + FP}$$

$0 \le PPV \le 1$

Correlation Coefficient

[Brazma et.al., 1998]

$$cc = \frac{(TP * TN - FP * FN)}{\sqrt{(TP + FP) * (FP + TN) * (TN + FN) * (FN + TP)}}$$

$-1 \le cc \le 1$

cc
- 1.0 no FP or FN
- 0.0 when *f* is random with respect to S+ and S-
- -1.0 only FP and FN

F-measure = 2 * (Precision * Recall) / (Precision + Recall)

F-measure

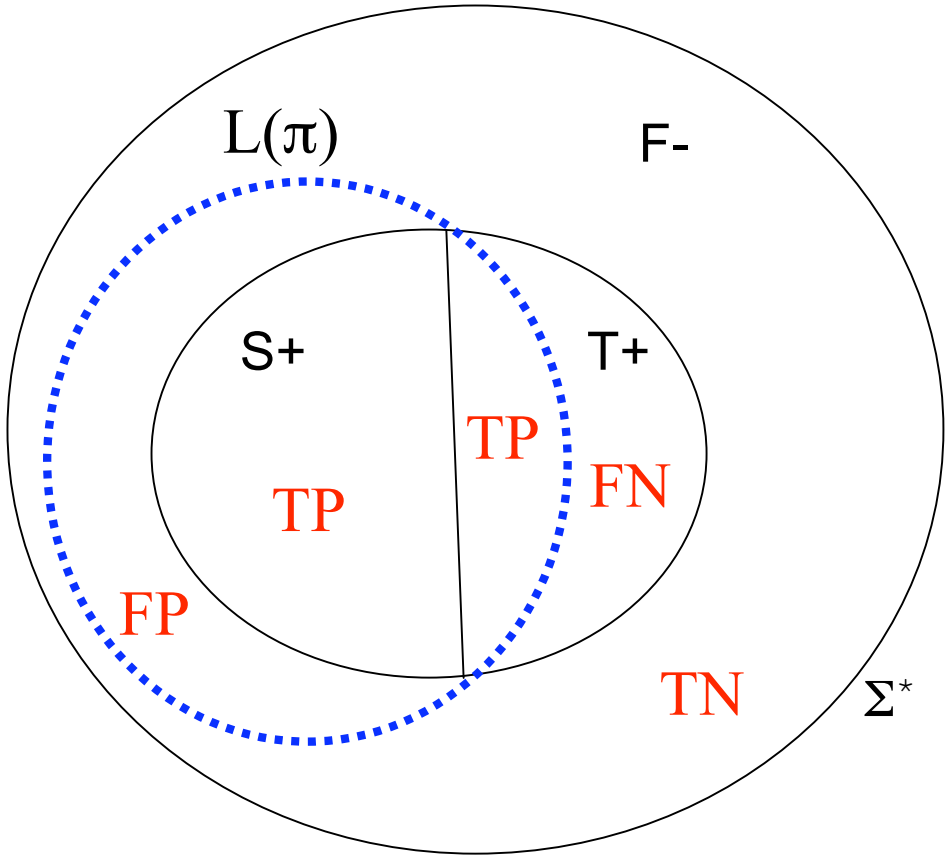
F_1-measure = 2 * (Precision * Recall) / (Precision + Recall)

General
F-measure = $(1+\alpha)$ * (Precision * Recall) / (α*Precision + Recall)

Training and test sets (positive examples only)

Training set	S+
Test set	T+
Pattern	π
Language of the pattern	$L(\pi)$

Assume that
$S+ \cup T+ = F+$
$(S+ \cup T+) \cap F- = \varnothing$



Methodology

- *Solution space / hypothesis space / target class*: find a good class of string functions from which the approximating function *f* is chosen for a real-world problem
- *Fitness measure*: define a ranking of the solution space, evaluating how good each function is for the training set (how likely *f* is to approximate *g*
- Develop an *algorithm* returning those classifier functions from the given solution space that rate high enough according to the fitness measure

Defining string functions via patterns

Given a string s and a pattern π which defines a language $L(\pi)$, define a classification (conservation) function f by

$$f(s)= \begin{cases} \text{TRUE} & \text{if } s \in L(\pi) \\ \text{FALSE} & \text{otherwise} \end{cases}$$

$$f(s)= \begin{cases} \text{TRUE} & \text{if } Dist(\pi,s) \le const \\ \text{FALSE} & \text{otherwise} \end{cases}$$

Where $Dist(\pi,s) = \min_{s' \in L(\pi)} dist(s', s)$

e.g. string comparison distance

Clean / Noisy Data

- Clean data: the training set is assumed to be "correct"
- Noisy data: training set
  - sequences may contain errors
  - sequences may have been assigned to the wrong family

PROSITE profiles

- Uses Hidden Markov Model - can characterise an entire family of sequences.