An introduction to Patterns, Profiles, HMMs and PSI-BLAST

Marco Pagni and Lorenzo Cerutti
Swiss Institute of Bioinformatics

Course, 2003

Outline

- Introduction
  - Multiple alignments and their information content.
- Models for multiple alignments
  - Consensus sequences
  - Patterns and regular expressions
  - Position Specific Scoring Matrices (PSSMs)
  - Generalized Profiles
  - Hidden Markov Models (HMMs)
- PSI-BLAST and protein domain hunting
- Databases of protein motifs, domains, and families

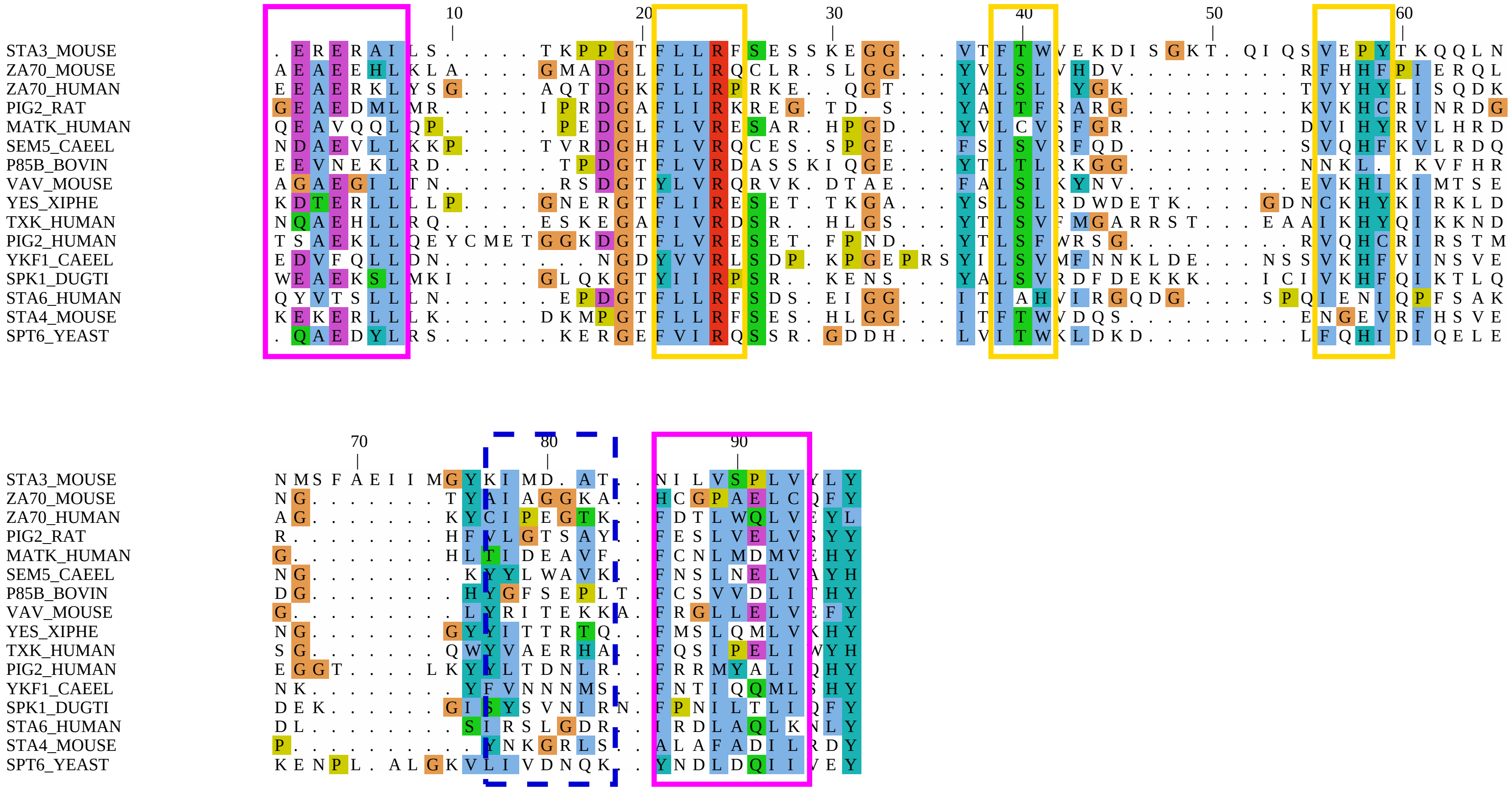
Color code: *Keywords*, *Databases*, *Software*

Multiple alignments

Multiple sequence alignment (MSA)

- The alignment of multiple sequences is a method of choice to detect *conserved regions* in protein or DNA sequences. These particular regions are usually associated with:
  - Signals (promoters, signatures for phosphorylation, cellular location, ...);
  - Structure (correct folding, protein-protein interactions...);
  - Chemical reactivity (catalytic sites,...).
- The information represented by these conserved regions can be used to align sequences, search similar sequences in the databases or annotate new sequences.
- Different methods exist to build *models* of these conserved regions:
  - Consensus sequences;
  - Patterns;
  - Position Specific Score Matrices (PSSMs);
  - Profiles;
  - Hidden Markov Models (HMMs),
  - ... and a few others.

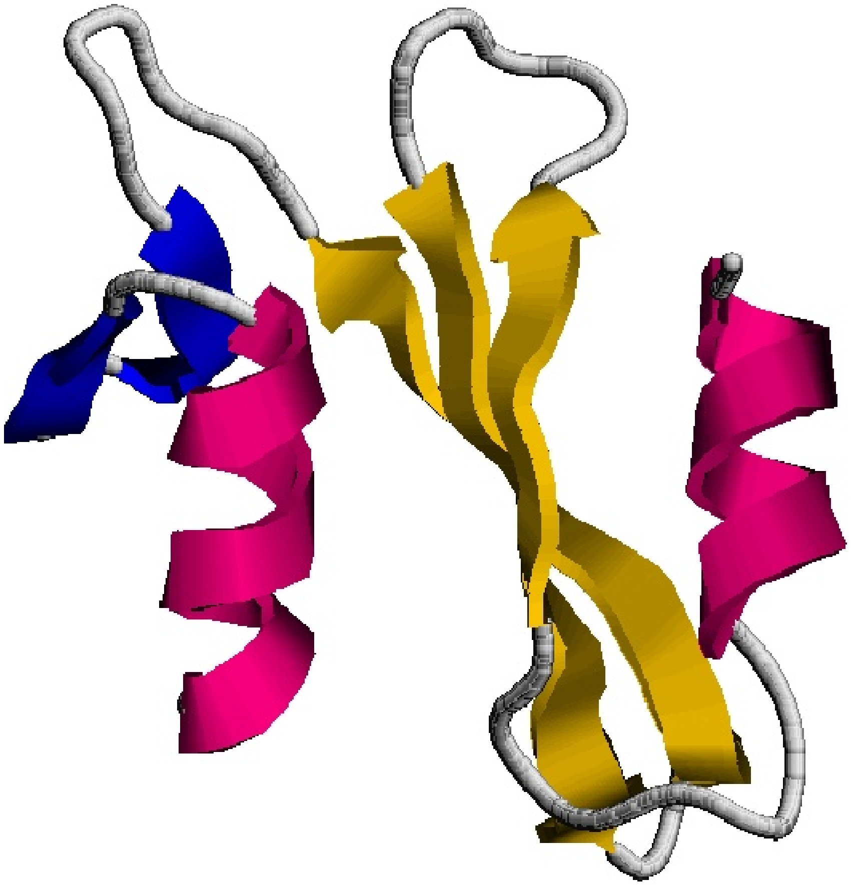
Example: Multiple alignments reflect secondary structures



```
                               10                  20                  30                  40                  50                  60
                               |                   |                   |                   |                   |                   |
STA3_MOUSE   . E R E R A I L S . . . . . T K P P G T F L L R F S E S S K E G G . . . V T F T W V E K D I S G K T . Q I Q S V E P Y T K Q Q L N
ZA70_MOUSE   A E A E E H L K L A . . . . G M A D G L F L L R Q C L R . S L G G . . . Y V L S L V H D V . . . . . . . . . R F H H F P I E R Q L
ZA70_HUMAN   E E A E R K L Y S G . . . . A Q T D G K F L L R P R K E . . Q G T . . . Y A L S L I Y G K . . . . . . . . . T V Y H Y L I S Q D K
PIG2_RAT     G E A E D M L M R . . . . . I P R D G A F L I R K R E G . T D . S . . . Y A I T F R A R G . . . . . . . . . K V K H C R I N R D G
MATK_HUMAN   Q E A V Q Q L Q P . . . . . . P E D G L F L V R E S A R . H P G D . . . Y V L C V S F G R . . . . . . . . . D V I H Y R V L H R D
SEM5_CAEEL   N D A E V L L K K P . . . . T V R D G H F L V R Q C E S . S P G E . . . F S I S V R F Q D . . . . . . . . . S V Q H F K V L R D Q
P85B_BOVIN   E E V N E K L R D . . . . . . T P D G T F L V R D A S S K I Q G E . . . Y T L T L R K G G . . . . . . . . . N N K L . I K V F H R
VAV_MOUSE    A G A E G I L T N . . . . . . R S D G T Y L V R Q R V K . D T A E . . . F A I S I K Y N V . . . . . . . . . E V K H I K I M T S E
YES_XIPHE    K D T E R L L L L P . . . . G N E R G T F L I R E S E T . T K G A . . . Y S L S L R D W D E T K . . . . G D N C K H Y K I R K L D
TXK_HUMAN    N Q A E H L L R Q . . . . . E S K E G A F I V R D S R . . H L G S . . . Y T I S V F M G A R R S T . . . E A A I K H Y Q I K K N D
PIG2_HUMAN   T S A E K L L Q E Y C M E T G G K D G T F L V R E S E T . F P N D . . . Y T L S F W R S G . . . . . . . . . R V Q H C R I R S T M
YKF1_CAEEL   E D V F Q L L D N . . . . . . . . N G D Y V V R L S D P . K P G E P R S Y I L S V M F N N K L D E . . . N S S V K H F V I N S V E
SPK1_DUGTI   W E A E K S L M K I . . . . G L Q K G T Y I I R P S R . . K E N S . . . Y A L S V R D F D E K K K . . . I C I V K H F Q I K T L Q
STA6_HUMAN   Q Y V T S L L L N . . . . . . E P D G T F L L R F S D S . E I G G . . . I T I A H V I R G Q D G . . . . S P Q I E N I Q P F S A K
STA4_MOUSE   K E K E R L L L K . . . . . D K M P G T F L L R F S E S . H L G G . . . I T F T W V D Q S . . . . . . . . . E N G E V R F H S V E
SPT6_YEAST   . Q A E D Y L R S . . . . . . K E R G E F V I R Q S S R . G D D H . . . L V I T W K L D K D . . . . . . . . L F Q H I D I Q E L E

                     70                  80                  90
                     |                   |                   |
STA3_MOUSE   N M S F A E I I M G Y K I M D . A T . . N I L V S P L V Y L Y
ZA70_MOUSE   N G . . . . . . . T Y A I A G G K A . . H C G P A E L C Q F Y
ZA70_HUMAN   A G . . . . . . . K Y C I P E G T K . . F D T L W Q L V E Y L
PIG2_RAT     R . . . . . . . . H F V L G T S A Y . . F E S L V E L V S Y Y
MATK_HUMAN   G . . . . . . . . H L T I D E A V F . . F C N L M D M V E H Y
SEM5_CAEEL   N G . . . . . . . . K Y Y L W A V K . . F N S L N E L V A Y H
P85B_BOVIN   D G . . . . . . . . H Y G F S E P L T . F C S V V D L I T H Y
VAV_MOUSE    G . . . . . . . . . L Y R I T E K K A . F R G L L E L V E F Y
YES_XIPHE    N G . . . . . . . G Y Y I T T R T Q . . F M S L Q M L V K H Y
TXK_HUMAN    S G . . . . . . . Q W Y V A E R H A . . F Q S I P E L I W Y H
PIG2_HUMAN   E G G T . . . . L K Y Y L T D N L R . . F R R M Y A L I Q H Y
YKF1_CAEEL   N K . . . . . . . . Y F V N N N M S . . F N T I Q Q M L S H Y
SPK1_DUGTI   D E K . . . . . . G I S Y S V N I R N . F P N I L T L I Q F Y
STA6_HUMAN   D L . . . . . . . . S I R S L G D R . . I R D L A Q L K N L Y
STA4_MOUSE   P . . . . . . . . . . Y N K G R L S . . A L A F A D I L R D Y
SPT6_YEAST   K E N P L . A L G K V L I V D N Q K . . Y N D L D Q I I V E Y
```

Example: Multiple alignments reflect secondary structures

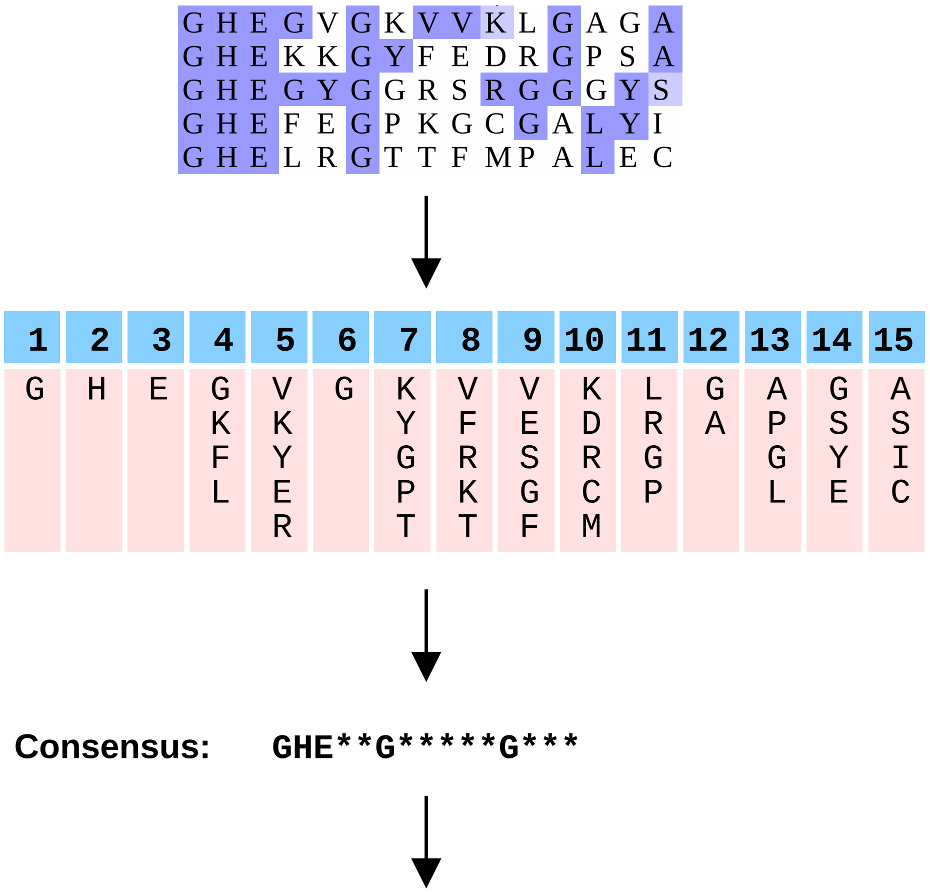


Consensus sequences

Consensus sequences

- The *consensus sequence* method is the simplest method to build a model from a multiple sequence alignment.
- The consensus sequence is built using the following rules:
  - Majority wins.
  - Skip too much variation.

How to build consensus sequences



```
G H E G V G K V V K L G A G A
G H E K K G Y F E D R G P S A
G H E G Y G G R S R G G G Y S
G H E F E G P K G C G A L Y I
G H E L R G T T F M P A L E C
```

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
G	H	E	G	V	G	K	V	V	K	L	G	A	G	A
			K	K		Y	F	E	D	R	A	P	S	S
			F	Y		G	R	S	R	G		G	Y	I
			L	E		P	K	G	C	P		L	E	C
				R		T	T	F	M					

Consensus: `GHE**G*****G***`

Search databases

Consensus sequences

- Advantages:
  - This method is very fast and easy to implement.
- Limitations:
  - Models have no information about variations in the columns.
  - Very dependent on the training set.
  - No scoring, only binary result (YES/NO).
- When I use it?
  - Useful to find highly conserved signatures, as for example enzyme restriction sites for DNA.

Pattern matching

Pattern syntax

- A *pattern* describes a set of alternative sequences, using a single expression. In computer science, patterns are known as *regular expressions*.
- The *Prosite* syntax for patterns:
  - uses the standard IUPAC one-letter codes for amino acids (G=Gly, P=Pro, ...),
  - each element in a pattern is separated from its neighbor by a '-',
  - the symbol 'X' is used where any amino acid is accepted,
  - ambiguities are indicated by square parentheses '[ ]' ([AG] means Ala or Gly),
  - amino acids that are not accepted at a given position are listed between a pair of curly brackets '{ }' ({AG} means any amino acid except Ala and Gly),
  - repetitions are indicated between parentheses '( )' ([AG](2,4) means Ala or Gly between 2 and 4 times, X(2) means any amino acid twice),
  - a pattern is anchored to the N-term and/or C-term by the symbols '<' and '>' respectively.

Pattern syntax: an example

- The following pattern

  $<$A-x-[ST](2)-x(0,1)-{V}

  means:

  - an Ala in the N-term,
  - followed by any amino acid,
  - followed by a Ser or Thr twice,
  - followed or not by any residue,
  - followed by any amino acid except Val.

How to build a pattern

```
G H E G V G K V V K L G A G A
G H E K K G Y F E D R G P S A
G H E G Y G G R S R G G G Y S
G H E F E G P K G C G A L Y I
G H E L R G T T F M P A L E C
```

↓

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
G	H	E	G K F L	V K Y E R	G	K Y G P T	V F R K T	V E S G F	K D R C M	L R G P	G A	A P G L	G S Y E	A S I C

↓

Pattern: **G−H−E−X(2)−G−X(5)−[GA]−X(3)**

↓

Search databases

Pattern examples

- Example of short signatures:
  - Post-translational signatures:
    - Protein splicing signature:
      [DNEG]-x-[LIVFA]-[LIVMY]-[LVAST]-H-N-[STC]
    - Tyrosine kinase phosphorylation site:
      [RK]-x(2)-[DE]-x(3)-Y or [RK]-x(3)-[DE]-x(2)-Y
  - DNA-RNA interaction signatures:
    - Histone H4 signature:
      G-A-K-R-H
    - p53 signature:
      M-C-N-S-S-C-[MV]-G-G-M-N-R-R
  - Enzymes:
    - L-lactate dehydrogenase active site:
      [LIVMA]-G-[EQ]-H-G-[DN]-[ST]
    - Ubiquitin-activating enzyme signature:
      P-[LIVM]-C-T-[LIVM]-[KRH]-x-[FT]-P

Patterns: Conclusion

- Patterns and PSSMs are appropriate to build models of short sequence signatures.
- Advantages:
  - Pattern matching is fast and easy to implement.
  - Models are easy to design for anyone with some training in biochemistry.
  - Models are easy to understand for anyone with some training in biochemistry.
- Limitations:
  - Poor model for insertions/deletions (indels).
  - Small patterns find a lot of false positives. Long patterns are very difficult to design.
  - Poor predictors that tend to recognize only the sequence of the training set.
  - No scoring system, only binary response (YES/NO).
- When I use patterns?
  - To search for small signatures or active sites.
  - To communicate with other biologists.

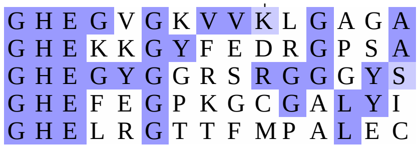
Patterns: beyond the conclusion

- Patterns can be automatically extracted (discovered) from a set of unaligned sequences by specialized programs.
- *Pratt*, *Splash* and *Teiresas* are three of these specialized programs.
- Today *machine learning* is a very active research field
- Such automatic patterns are usually distinct from those designed by an expert with some knowledge of the biochemical literature.

Position Specific Scoring Matrice (PSSM)

How to build a PSSM

- A *PSSM* is based on the *frequencies* of each residue in a specific position of a multiple alignment.



```
G H E G V G K V V K L G A G A
G H E K K G Y F E D R G P S A
G H E G Y G G R S R G G G Y S
G H E F E G P K G C G A L Y I
G H E L R G T T F M P A L E C
```

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A	0	0	0	0	0	0	0	0	0	0	0	2	1	0	2
C	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1
D	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E	0	0	5	0	1	0	0	0	1	0	0	0	0	1	0
F	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0
G	5	0	0	2	0	5	1	0	1	0	2	3	1	1	0
H	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
K	0	0	0	1	1	0	1	1	0	1	0	0	0	0	0
L	0	0	0	1	0	0	0	0	0	0	1	0	2	0	0
M	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
N	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0
Q	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	0	0	0	0	1	0	0	1	0	1	1	0	0	0	0
S	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0
T	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0
V	0	0	0	0	1	0	0	1	1	0	0	0	0	0	0
W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Y	0	0	0	0	1	0	1	0	0	0	0	0	0	1	0

- Column 1: $f_{A,1} = \frac{0}{5} = 0,\ f_{G,1} = \frac{5}{5} = 1,\ \ldots$
- Column 2: $f_{A,2} = \frac{0}{5} = 0,\ f_{H,2} = \frac{5}{5} = 1,\ \ldots$
- ...
- Column 15: $f_{A,15} = \frac{2}{5} = 0.4,\ f_{C,15} = \frac{1}{5} = 0.2,\ \ldots$

Pseudo-counts

- Some observed frequencies usually equal 0. This is a consequence of the limited number of sequences that is present in a MSA.
- Unfortunately, an observed frequency of 0 might imply the exclusion of the corresponding residue at this position (this was the case with patterns).
- One possible trick is to add a small number to all observed frequencies. These small non-observed frequencies are referred to as *pseudo-counts*.
- From the previous example with a pseudo-counts of 1:
    - Column 1: $f'_{A,1} = \frac{0+1}{5+20} = 0.04$, $f'_{G,1} = \frac{5+1}{5+20} = 0.24$, ...
    - Column 2: $f'_{A,2} = \frac{0+1}{5+20} = 0.04$, $f'_{H,2} = \frac{5+1}{5+20} = 0.24$, ...
    - ...
    - Column 15: $f'_{A,15} = \frac{2+1}{5+20} = 0.12$, $f'_{C,15} = \frac{1+1}{5+20} = 0.08$, ...
- There exist more sophisticated methods to produce more “realistic” pseudo-counts, and which are based on *substitution matrix* or *Dirichlet mixtures*.

Computing a PSSM

- The frequency of every residue determined at every position has to be compared with the frequency at which any residue can be expected in a *random sequence*.
- For example, let's postulate that each amino acid is observed with an identical frequency in a random sequence. This is a quite simplistic *null model*.
- The *score* is derived from the ratio of the observed to the expected frequencies. More precisely, the logarithm of this ratio is taken and refereed to as the *log-likelihood ratio*:

$$Score_{ij} = log(\frac{f'_{ij}}{q_i})$$

where $Score_{ij}$ is the score for residue i at position j, f'_{ij} is the relative frequency for a residue i at position j (corrected with pseudo-counts) and q_i is the expected relative frequency of residue i in a random sequence.

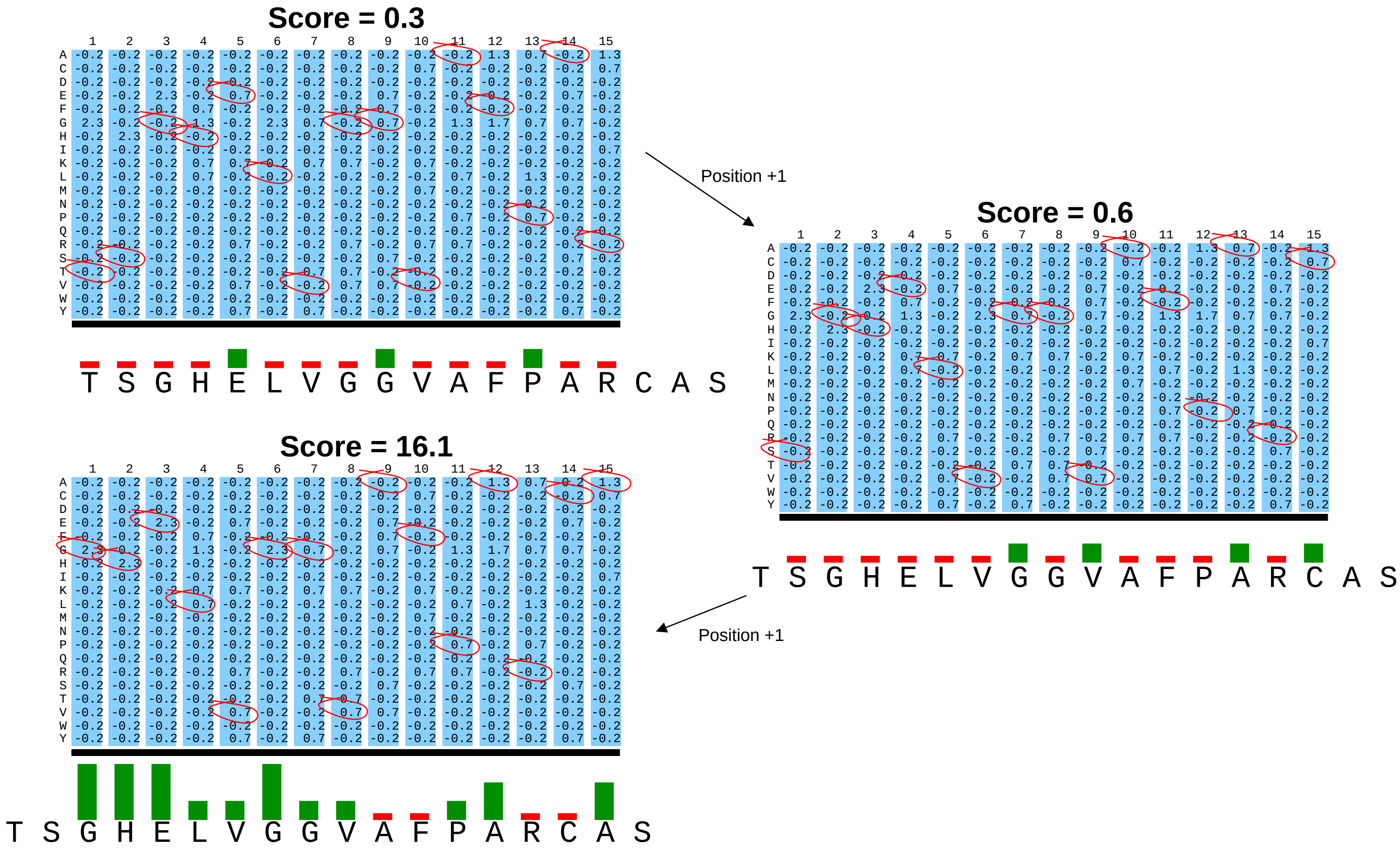
Example

- The complete position specific scoring matrix calculated from the previous example:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	1.3	0.7	-0.2	1.3
C	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	0.7	-0.2	-0.2	-0.2	-0.2	0.7
D	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2
E	-0.2	-0.2	2.3	-0.2	0.7	-0.2	-0.2	-0.2	0.7	-0.2	-0.2	-0.2	-0.2	0.7	-0.2
F	-0.2	-0.2	-0.2	0.7	-0.2	-0.2	-0.2	-0.2	0.7	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2
G	2.3	-0.2	-0.2	1.3	-0.2	2.3	0.7	-0.2	0.7	-0.2	1.3	1.7	0.7	0.7	-0.2
H	-0.2	2.3	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2
I	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	0.7
K	-0.2	-0.2	-0.2	0.7	0.7	-0.2	0.7	0.7	-0.2	0.7	-0.2	-0.2	-0.2	-0.2	-0.2
L	-0.2	-0.2	-0.2	0.7	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	0.7	-0.2	1.3	-0.2	-0.2
M	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	0.7	-0.2	-0.2	-0.2	-0.2	-0.2
N	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2
P	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	0.7	-0.2	0.7	-0.2	-0.2
Q	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2
R	-0.2	-0.2	-0.2	-0.2	0.7	-0.2	-0.2	0.7	-0.2	0.7	0.7	-0.2	-0.2	-0.2	-0.2
S	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	0.7	-0.2	-0.2	-0.2	-0.2	0.7	-0.2
T	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	0.7	0.7	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2
V	-0.2	-0.2	-0.2	-0.2	0.7	-0.2	-0.2	0.7	0.7	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2
W	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2
Y	-0.2	-0.2	-0.2	-0.2	0.7	-0.2	0.7	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	0.7	-0.2

How to use PSSMs

- The PSSM is applied as a *sliding window* along the subject sequence:
  - At every position, a PSSM score is calculated by summing the scores of all columns;
  - The highest scoring position is reported.

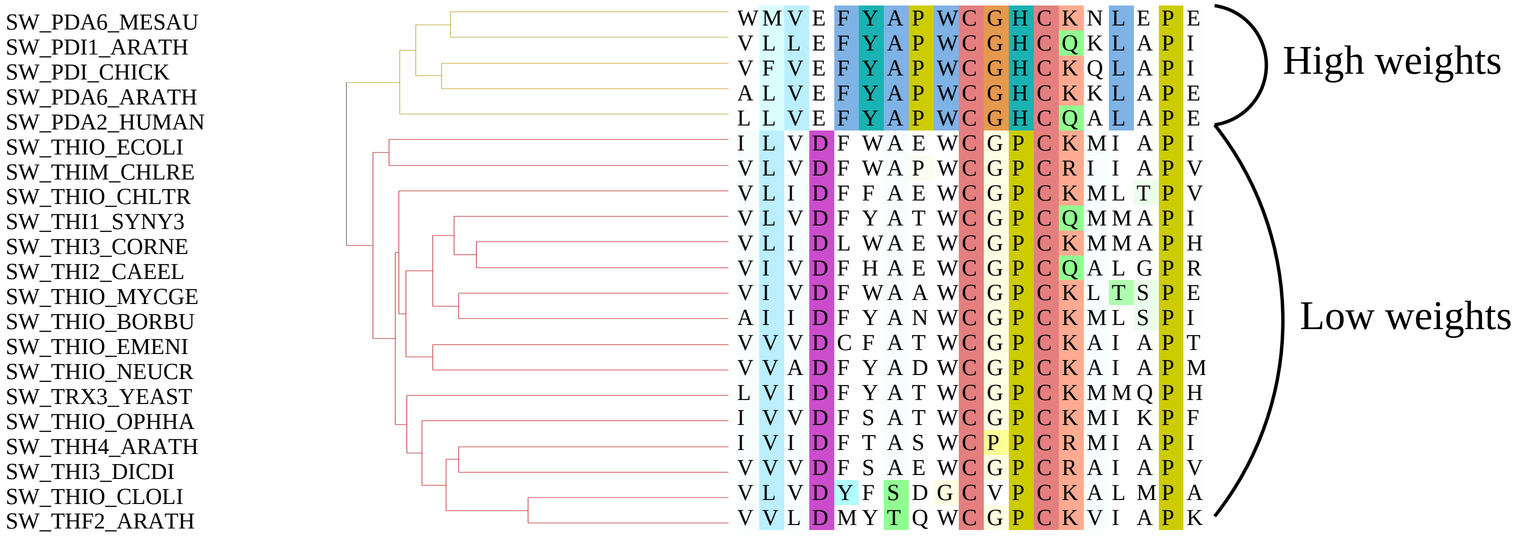


	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	1.3	0.7	-0.2	1.3
C	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	0.7	-0.2	-0.2	-0.2	-0.2	0.7
D	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2
E	-0.2	-0.2	2.3	-0.2	0.7	-0.2	-0.2	-0.2	0.7	-0.2	-0.2	-0.2	-0.2	0.7	-0.2
F	-0.2	-0.2	-0.2	0.7	-0.2	-0.2	-0.2	-0.2	0.7	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2
G	2.3	-0.2	-0.2	1.3	-0.2	2.3	0.7	-0.2	0.7	-0.2	1.3	1.7	0.7	0.7	-0.2
H	-0.2	2.3	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2
I	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	0.7
K	-0.2	-0.2	-0.2	0.7	0.7	-0.2	0.7	0.7	-0.2	0.7	-0.2	-0.2	-0.2	-0.2	-0.2
L	-0.2	-0.2	-0.2	0.7	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	0.7	-0.2	1.3	-0.2	-0.2
M	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	0.7	-0.2	-0.2	-0.2	-0.2	-0.2
N	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2
P	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	0.7	-0.2	0.7	-0.2	-0.2
Q	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2
R	-0.2	-0.2	-0.2	-0.2	0.7	-0.2	-0.2	0.7	-0.2	0.7	0.7	-0.2	-0.2	-0.2	-0.2
S	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	0.7	-0.2	-0.2	-0.2	-0.2	0.7	-0.2
T	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	0.7	0.7	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2
V	-0.2	-0.2	-0.2	-0.2	0.7	-0.2	-0.2	0.7	0.7	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2
W	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2
Y	-0.2	-0.2	-0.2	-0.2	0.7	-0.2	0.7	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	0.7	-0.2

Sequence weighting

- An MSA is often made of a few distinct sets of related sequences, or sub-families. It is not unusual that these sub-families are very differently populated, thus influencing observed residue frequencies.

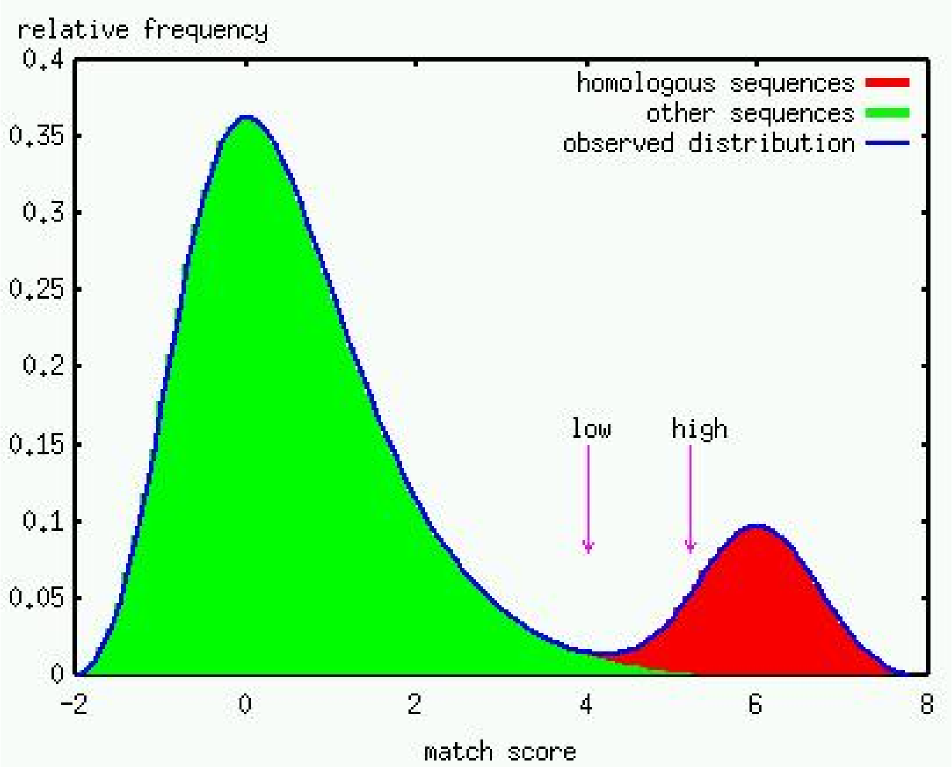
- *Sequences weighting algorithms* attempt to compensate this sequence sampling bias.



```
SW_PDA6_MESAU   WMVEFYAPWCGHCKNLEPE
SW_PDI1_ARATH   VLLEFYAPWCGHCQKLAPI
SW_PDI_CHICK    VFVEFYAPWCGHCKQLAPI
SW_PDA6_ARATH   ALVEFYAPWCGHCKKLAPE
SW_PDA2_HUMAN   LLVEFYAPWCGHCQALAPE
SW_THIO_ECOLI   ILVDFWAEWCGPCKMIAPI
SW_THIM_CHLRE   VLVDFWAPWCGPCRIIAPV
SW_THIO_CHLTR   VLIDFFAEWCGPCKMLTPV
SW_THI1_SYNY3   VLVDFYATWCGPCQMMAPI
SW_THI3_CORNE   VLIDLWAEWCGPCKMMAPH
SW_THI2_CAEEL   VIVDFHAEWCGPCQALGPR
SW_THIO_MYCGE   VIVDFWAAWCGPCKLTSPE
SW_THIO_BORBU   AIIDFYANWCGPCKMLSPI
SW_THIO_EMENI   VVVDCFATWCGPCKAIAPT
SW_THIO_NEUCR   VVADFYADWCGPCKAIAPM
SW_TRX3_YEAST   LVIDFYATWCGPCKMMQPH
SW_THIO_OPHHA   IVVDFSATWCGPCKMIKPF
SW_THH4_ARATH   IVIDFTASWCPPCRMIAPI
SW_THI3_DICDI   VVVDFSAEWCGPCRAIAPV
SW_THIO_CLOLI   VLVDYFSDGCVPCKALMPA
SW_THF2_ARATH   VVLDMYTQWCGPCKVIAPK
```

PSSM Score Interpretation

- The *E-value* is the number of matches with a score equal to or greater than the observed score that are expected to occur *by chance*.
- The E-value depends on the size of the searched database, as the number of false positives expected above a given score threshold increases proportionately with the size of the database.



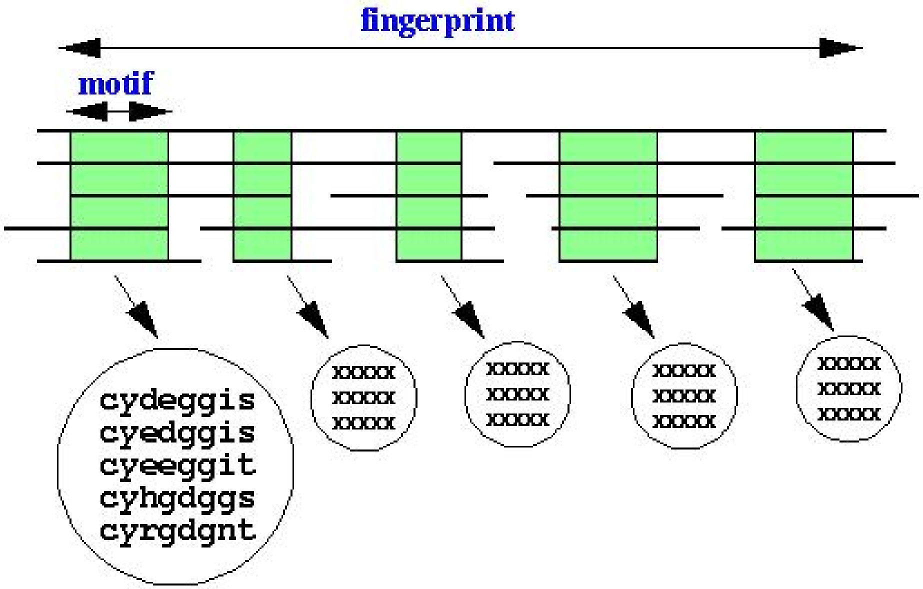
PSSM: Conclusion

- Advantages:
  - Good for short, conserved regions.
  - Relatively fast and simple to implement.
  - Produce match scores that can be interpreted based on statistical theory.
- Limitations:
  - Insertions and deletions are strictly forbidden.
  - Relatively long sequence regions can therefore not be described with this method.
- When I use it?
  - To model small regions with high variability but constant length.

PSSM: beyond the conclusion

- PSSMs can be automatically extracted (discovered) from a set of unaligned sequences by specialized programs. The program *MEME* is such a tool which is based on the *expectation-maximization algorithm* http://meme.sdsc.edu/meme/website/.

- A couple of PSSMs can be used to describe the conserved regions of a large MSA. A database of such diagnostic PSSMs and search tools dedicated for that purpose is available (*Prints*).



Generalized profiles

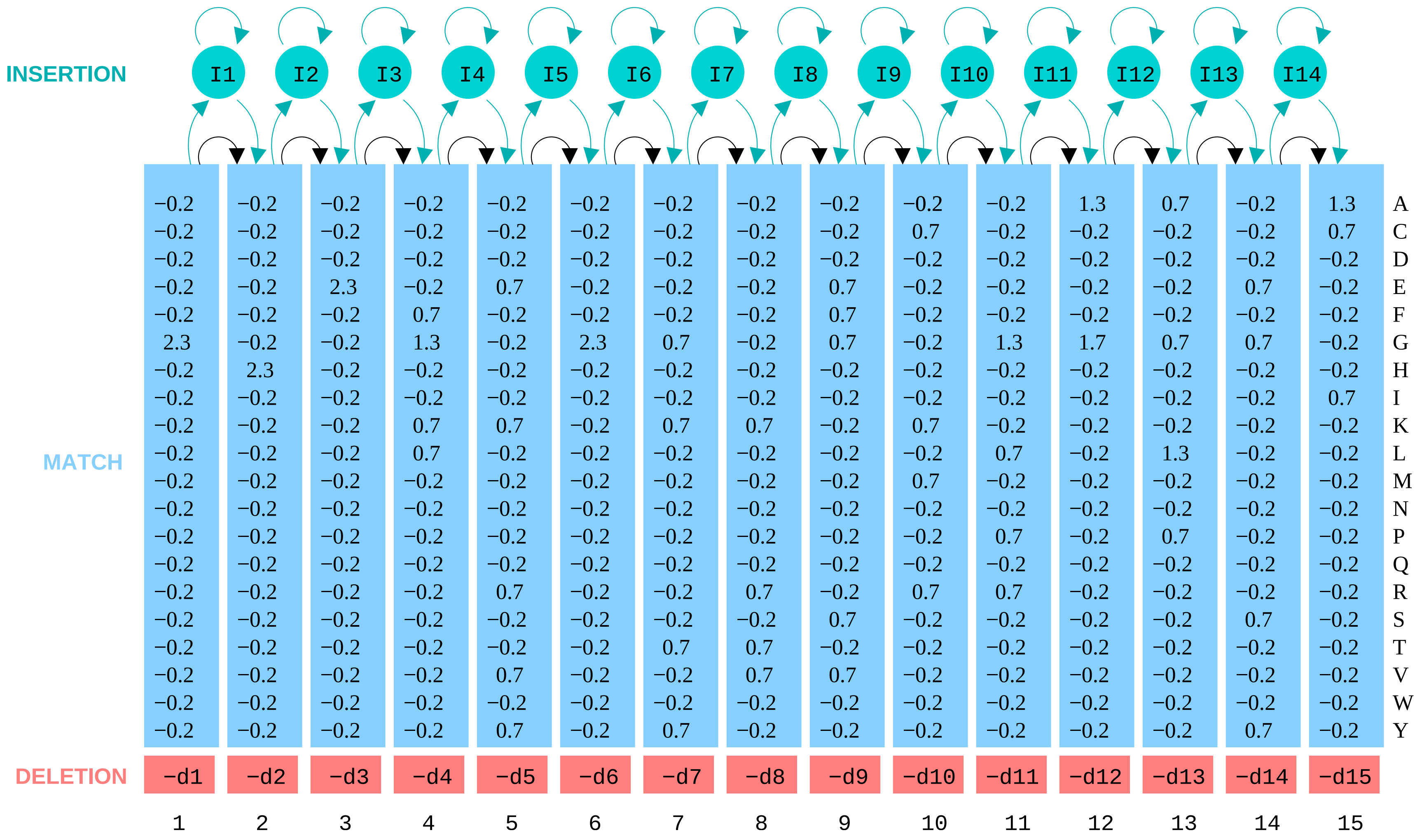
The idea behind generalized profiles

- One would like to generalize PSSMs to allow for insertions and deletions. However this raises the difficult problems of defining and computing an optimal alignment with gaps.
- Let us recycle the principle of *dynamic programing*, as it was introduced to define and compute the optimal alignments between a pair of sequences e.g. by the Smith-Waterman algorithm, and generalize it by the introduction of:
  - position-dependent match scores,
  - position-dependent gap penalties.

Generalized profiles as an extension of PSSMs

- The following information is stored in any generalized profile:
    - each position is called a *match state*. A score for every residue is defined at every match states, just as in the PSSM.
    - each match state can be omitted in the alignment, by what is called a *deletion state* and that receives a position-dependent penalty.
    - insertions of variable length are possible between any two adjacent match (or deletion) states. These *insertion states* are given a position-dependent penalty that might also depend upon the inserted residues.
    - every possible *transition* between any two states (match, delete or insert) receives a position-dependent penalty. This is primarily to model the cost of opening and closing a gap.
    - a couple of additional parameters permit to finely tune the behavior of the extremities of the alignment, which can forced to be 'local' or 'global' at either ends of the profile and of the sequence.

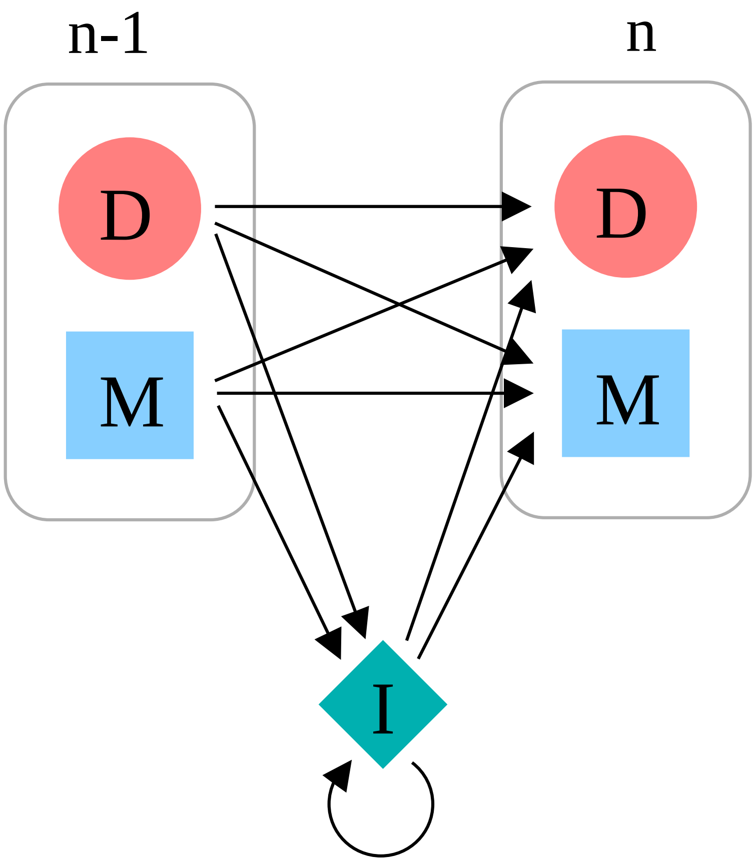
Generalized profiles as an extension of PSSMs



	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	1.3	0.7	−0.2	1.3	A
	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	0.7	−0.2	−0.2	−0.2	−0.2	0.7	C
	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	D
	−0.2	−0.2	2.3	−0.2	0.7	−0.2	−0.2	−0.2	0.7	−0.2	−0.2	−0.2	−0.2	0.7	−0.2	E
	−0.2	−0.2	−0.2	0.7	−0.2	−0.2	−0.2	−0.2	0.7	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	F
	2.3	−0.2	−0.2	1.3	−0.2	2.3	0.7	−0.2	0.7	−0.2	1.3	1.7	0.7	0.7	−0.2	G
	−0.2	2.3	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	H
	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	0.7	I
	−0.2	−0.2	−0.2	0.7	0.7	−0.2	0.7	0.7	−0.2	0.7	−0.2	−0.2	−0.2	−0.2	−0.2	K
MATCH	−0.2	−0.2	−0.2	0.7	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	0.7	−0.2	1.3	−0.2	−0.2	L
	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	0.7	−0.2	−0.2	−0.2	−0.2	−0.2	M
	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	N
	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	0.7	−0.2	0.7	−0.2	−0.2	P
	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	Q
	−0.2	−0.2	−0.2	−0.2	0.7	−0.2	−0.2	0.7	−0.2	0.7	0.7	−0.2	−0.2	−0.2	−0.2	R
	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	0.7	−0.2	−0.2	−0.2	−0.2	0.7	−0.2	S
	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	0.7	0.7	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	T
	−0.2	−0.2	−0.2	−0.2	0.7	−0.2	−0.2	0.7	0.7	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	V
	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	W
	−0.2	−0.2	−0.2	−0.2	0.7	−0.2	0.7	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	0.7	−0.2	Y
DELETION	−d1	−d2	−d3	−d4	−d5	−d6	−d7	−d8	−d9	−d10	−d11	−d12	−d13	−d14	−d15	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	

Generalized profiles are an extension of PSSMs

- Generalized profiles can be represented by a *finite state automata*:



Excerpt of a generalized profile

```
ID   THIOREDOXIN_2; MATRIX.
AC   PS50223;
DT         ? (CREATED); MAY-1999 (DATA UPDATE);        ? (INFO UPDATE).
DE   Thioredoxin-domain (does not find all).
MA   /GENERAL_SPEC: ALPHABET=’ABCDEFGHIKLMNPQRSTVWYZ’; LENGTH=103;
MA   /DISJOINT: DEFINITION=PROTECT; N1=6; N2=98;
MA   /NORMALIZATION: MODE=1; FUNCTION=LINEAR; R1=1.9370; R2=0.01816483; TEXT=’-LogE’;
MA   /CUT_OFF: LEVEL=0; SCORE=361; N_SCORE=8.5; MODE=1; TEXT=’!’;
MA   /DEFAULT: D=-20; I=-20; B1=-100; E1=-100; MM=1; MI=-105; MD=-105; IM=-105; DM=-105; M0=-6;
MA   /I: B1=0; BI=-105; BD=-105;
```

... many lines deleted ...

```
MA   /M: SY=’K’; M=-8,0,-25,1,8,-24,-14,-9,-22,19,-20,-11,0,-9,5,13,-3,-4,-16,-24,-13,6; D=-3;
MA   /I: I=-3; DM=-16;
MA   /M: SY=’P’; M=-6,-13,-26,-12,-9,-12,-19,-14,-5,-11,-5,-4,-12,8,-11,-13,-9,-6,-6,-25,-11,-12;
MA   /M: SY=’V’; M=-4,-22,-19,-24,-20,-2,-25,-21,11,-15,2,3,-20,-23,-17,-14,-9,-1,19,-11,-4,-19;
MA   /M: SY=’A’; M=28,-7,-15,-13,-6,-20,-2,-15,-15,-6,-14,-11,-5,-12,-6,-11,9,1,-6,-21,-17,-6;
MA   /M: SY=’P’; M=-6,-3,-27,2,2,-22,-14,-11,-20,-6,-24,-17,-5,25,-4,-11,3,1,-19,-29,-17,-3;
MA   /M: SY=’W’; M=-16,-27,-41,-28,-21,2,-13,-20,-20,-16,-19,-17,-26,-25,-15,-15,-26,-20,-26,93,19,-15;
MA   /M: SY=’C’; M=-9,-17,106,-26,-27,-20,-27,-28,-29,-28,-20,-20,-17,-37,-28,-28,-8,-9,-10,-48,-29,-27;
MA   /M: SY=’G’; M=-4,-12,-31,-9,-9,-27,24,-18,-27,-13,-25,-17,-7,14,-13,-17,-3,-13,-24,-24,-26,-13;
MA   /M: SY=’H’; M=-12,-10,-30,-8,-4,-14,-18,18,-17,-10,-18,-8,-7,16,-5,-11,-8,-10,-20,-22,-1,-8;
MA   /M: SY=’C’; M=-9,-19,111,-28,-28,-20,-29,-29,-28,-29,-20,-19,-18,-38,-28,-29,-8,-8,-9,-49,-29,-28;
MA   /M: SY=’R’; M=-12,-4,-27,-4,3,-22,-20,-2,-21,22,-19,-6,-2,-13,9,23,-9,-8,-16,-20,-6,4;
```

... many lines deleted ...

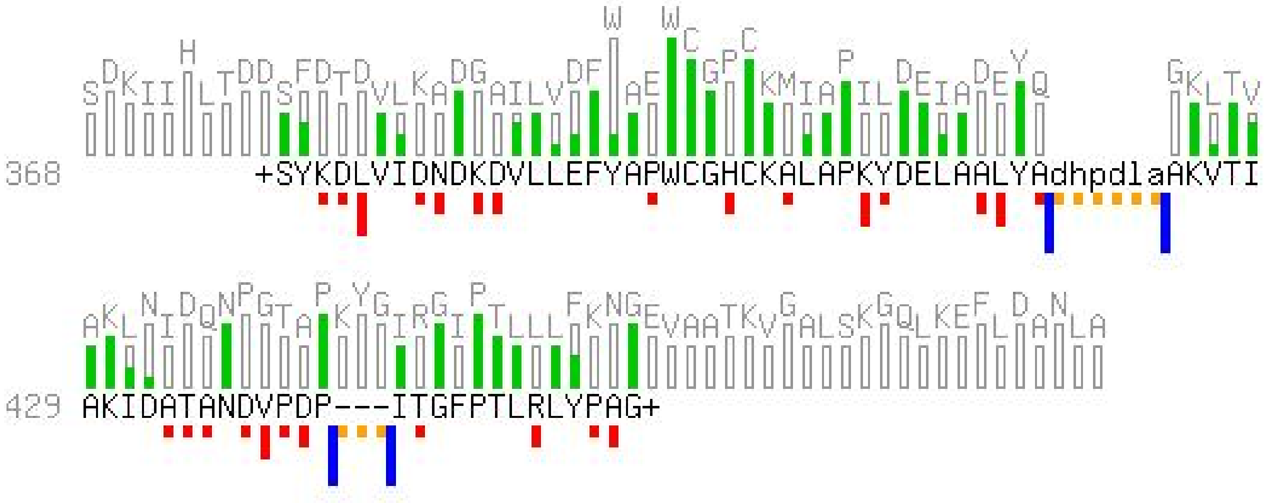
```
//
```

Details of the scores along an alignment I

- Smith-Waterman alignment of two thioredoxin domains:

```
THIO_ECOLI SFDTDVLKADGAILVDFWAEWCGPCKMIAPILDEIADEYQ------GKLTVAKLNIDQNP
           :.   :.  :  .:..:.: ::: :: .:: ::.:  :       .:.:.::..   :
PDI_ASPNG  SYKDLVIDNDKDVLLEFYAPWCGHCKALAPKYDELAALYADHPDLAAKVTIAKIDATAND

THIO_ECOLI GTAPKYGIRGIPTLLLFKNG
              :   : :.::: :.  :
PDI_ASPNG  VPDP---ITGFPTLRLYPAG
```

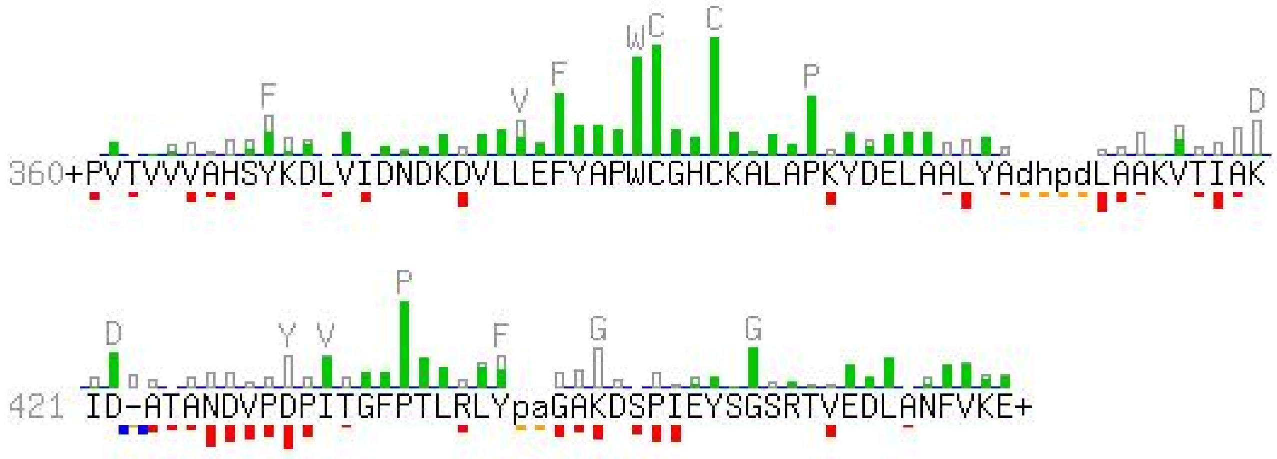


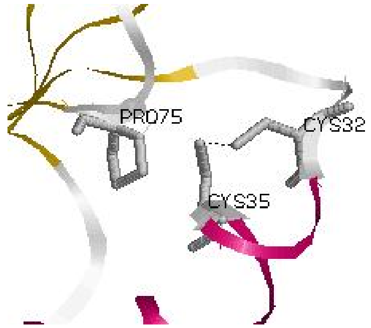
Details of the scores along an alignment II

- Alignment of a sequence of a thioredoxin domain on a profile built from a MSA of thioredoxins:

```
consensus   1 XVXVLSDENFDEXVXDSDKPVLVDFYAPWCGHCRALAPVFEELAEEYK----DBVKFVKV  -48
               : :         : : :  ::  : : ::::: : :    : :  :        :
PDI_ASPNG 360 PVTVVVAHSYKDLVIDNDKDVLLEFYAPWCGHCKALAPKYDELAALYAdhpdLAAKVTIA  -97

consensus  57 DVDENXELAEEYGVRGFPTIMFF--KBGEXVERYSGARBKEDLXEFIEK             -1
                :            : ::              : : :  : :  :
PDI_ASPNG 420 KID-ATANDVPDPITGFPTLRLYpaGAKDSPIEYSGSRTVEDLANFVKE             -49
```





Generalized profiles: Software

- *Pftools* is a package to build and use generalized profiles, which was developed by Philipp Bucher (http://www.isrec.isb-sib.ch/ftp-server/pftools/).

- The package contains (among other programs):
  - *pfmake* for building a profile starting from multiple alignments.
  - *pfcalibrate* to calibrate the profile model.
  - *pfsearch* to search a protein database with a profile.
  - *pfscan* to search a profile database with a protein.

Generalized profiles: Conclusions

- Advantage:
  - Possible to specify where deletions and insertions occur.
  - Very sensitive to detect homology below the twilight zone.
  - Good scoring system.
  - Automatic building of the profiles.
- Limitations:
  - Require more sophisticated software.
  - Very CPU expensive.
  - Require some expertise to use proficiently.

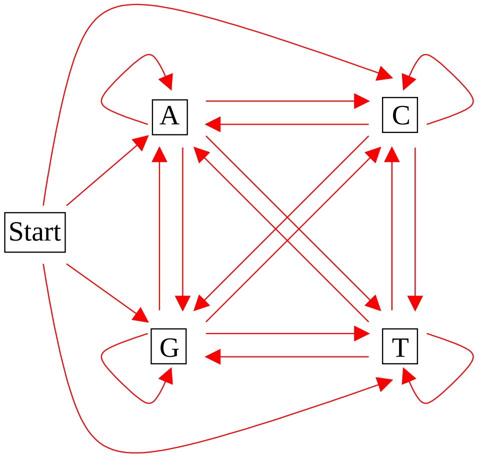
Hidden Markov Models (HMMs): probabilistic models

HMMs derive from Markov Chains

- *Hidden Markov Models (HMMs)* are an extension of the Markov Chains theory, which is part of the theory of probabilities.

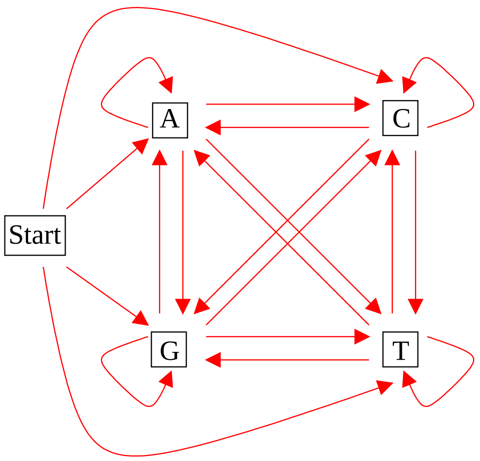
- A *Markov Chain* is a succession of *states* S_i $(i = 0, 1, ...)$ connected by *transitions*. Transitions from state S_i to state S_j has a probability of P_{ij}.

- An example of Markov Chain:
  - Transition probabilities:
    $P(A|G) = 0.18$, $P(C|G) = 0.38$, $P(G|G) = 0.32$, $P(T|G) = 0.12$
    $P(A|C) = 0.15$, $P(C|C) = 0.35$, $P(G|C) = 0.34$, $P(T|C) = 0.15$



How to calculate the probability of a Markov Chain

- Given a Markov Chain M where all transition probabilities are known:

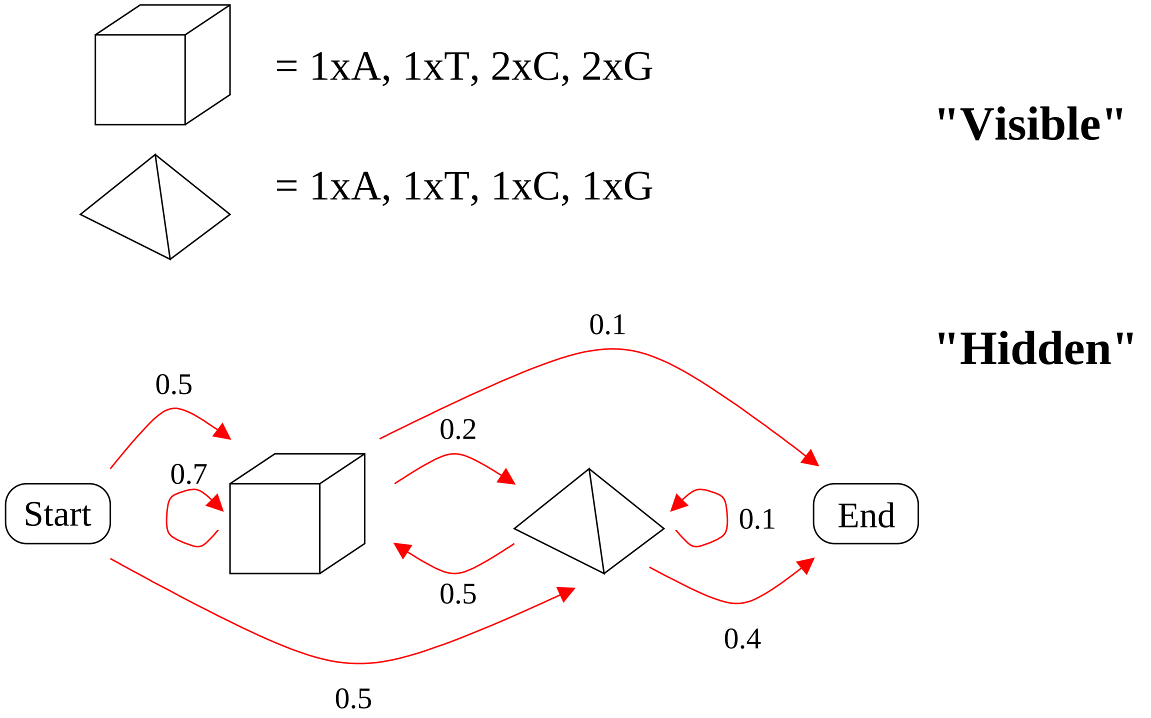


The probability of sequence $x = GCCT$ is:

$$P(GCCT) = P(T|C)P(C|C)P(C|G)P(G)$$

HMMs are an extension of Markov Chains

- HMMs are like Markov Chains: a finite number of *states* connected by *transitions*.
- But the major difference between the two is that the states of a HMM are not a symbol but a *distribution* of symbols. Each state can *emit* a symbol with a probability given by the distribution.

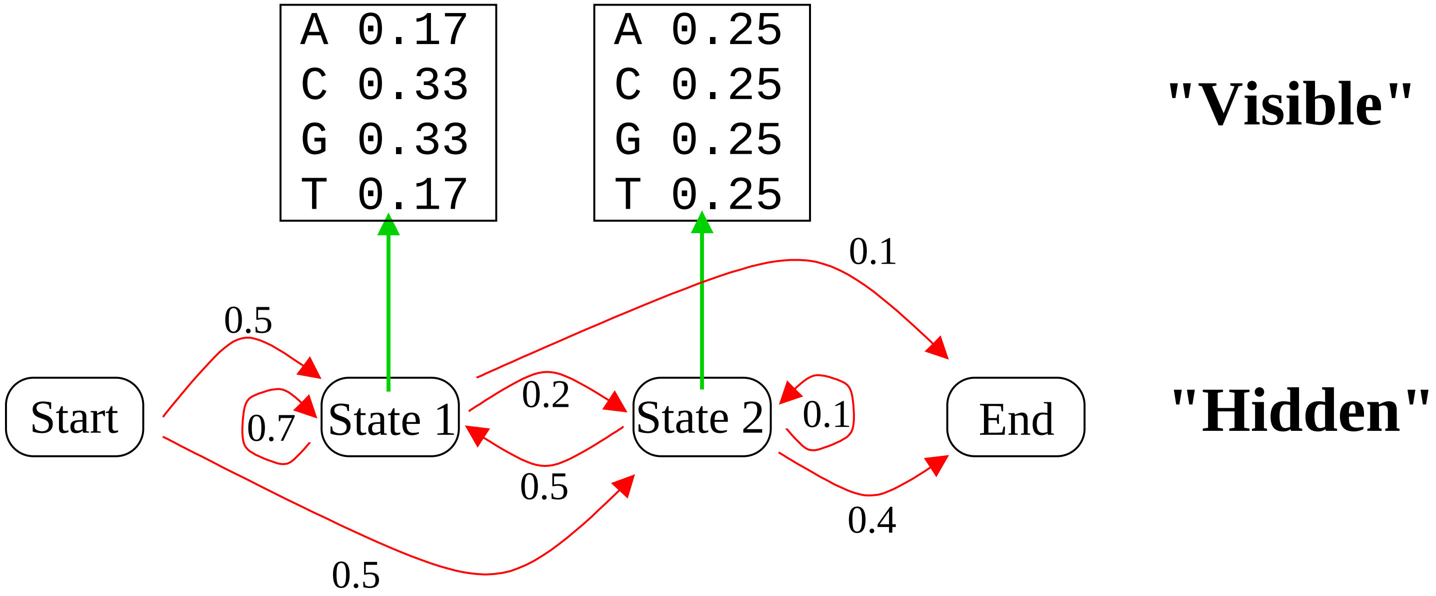


= 1xA, 1xT, 2xC, 2xG

= 1xA, 1xT, 1xC, 1xG

Example of a simple HMM

- Example of a simple HMM, generating GC rich DNA sequences:



A	0.17
C	0.33
G	0.33
T	0.17

A	0.25
C	0.25
G	0.25
T	0.25

START	1	1	1	1	2	2	1	1	1	2	END
	G	C	A	G	C	T	G	G	C	T	

HMM parameters

- The parameters describing HMMs:
  - *Emission probabilities*. The probability of emitting a symbol x from an alphabet α being in state q.

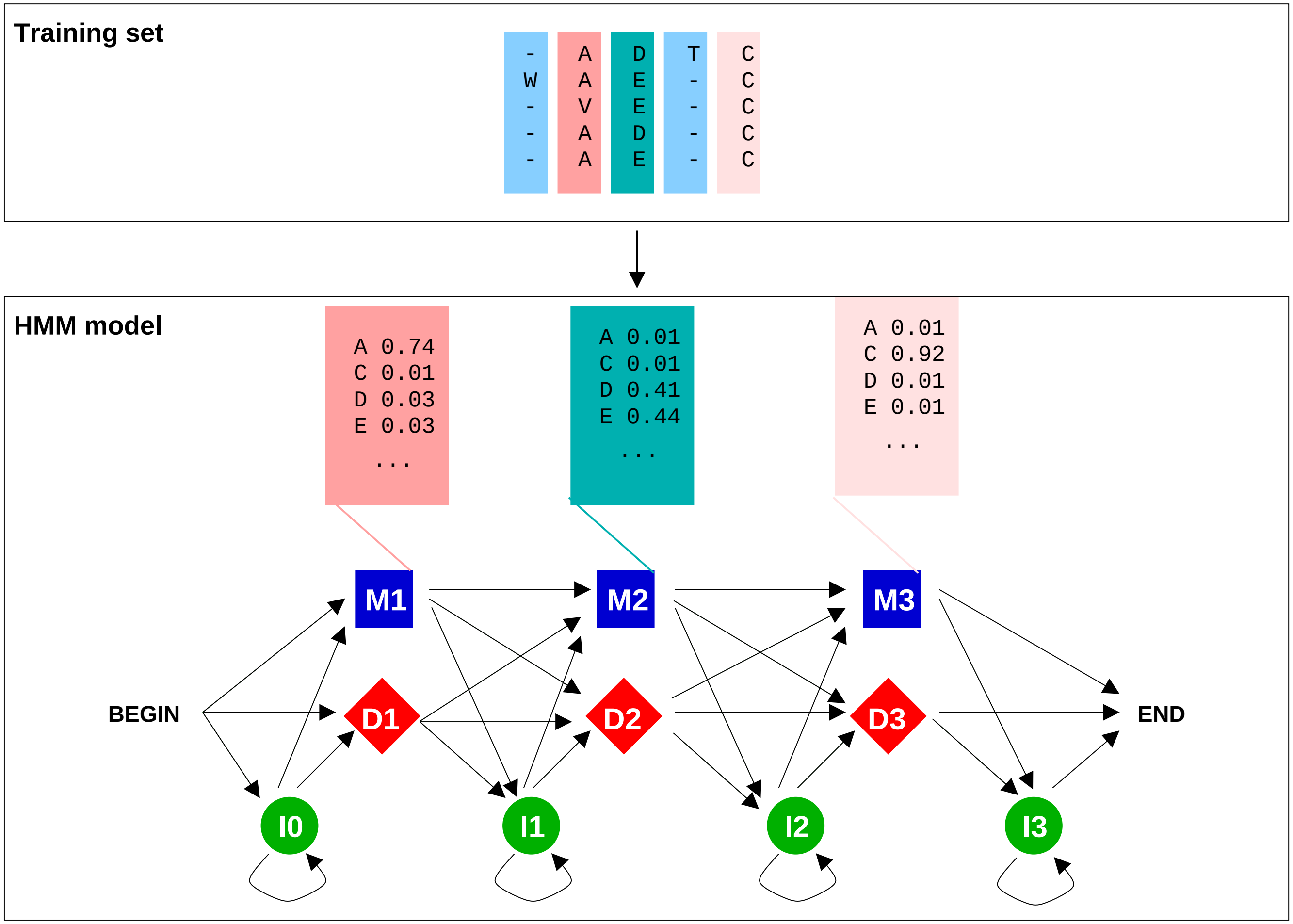
    $$E(x|q)$$

    - Residue emission probabilities are evaluated from the observed frequencies as for PSSMs.
    - Pseudo-counts are added to avoid emission probabilities equal to 0.
  - *Transition probabilities*. The probability of a transition to state r being in state q.

    $$T(r|q)$$

    - Transition probabilities are evaluated from observed transition frequencies.
- Emission and transition probabilities can also be evaluated using the *Baum-Welch training algorithm*.

HMMs are trained from a multiple alignment



Training set

```
- A D T C
W A E - C
- V E - C
- A D - C
- A E - C
```

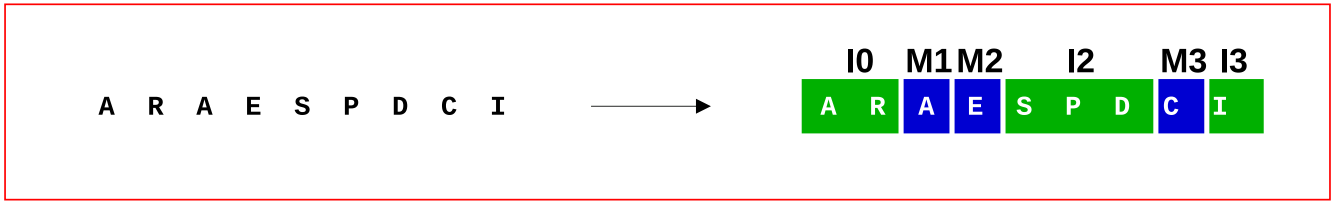
HMM model

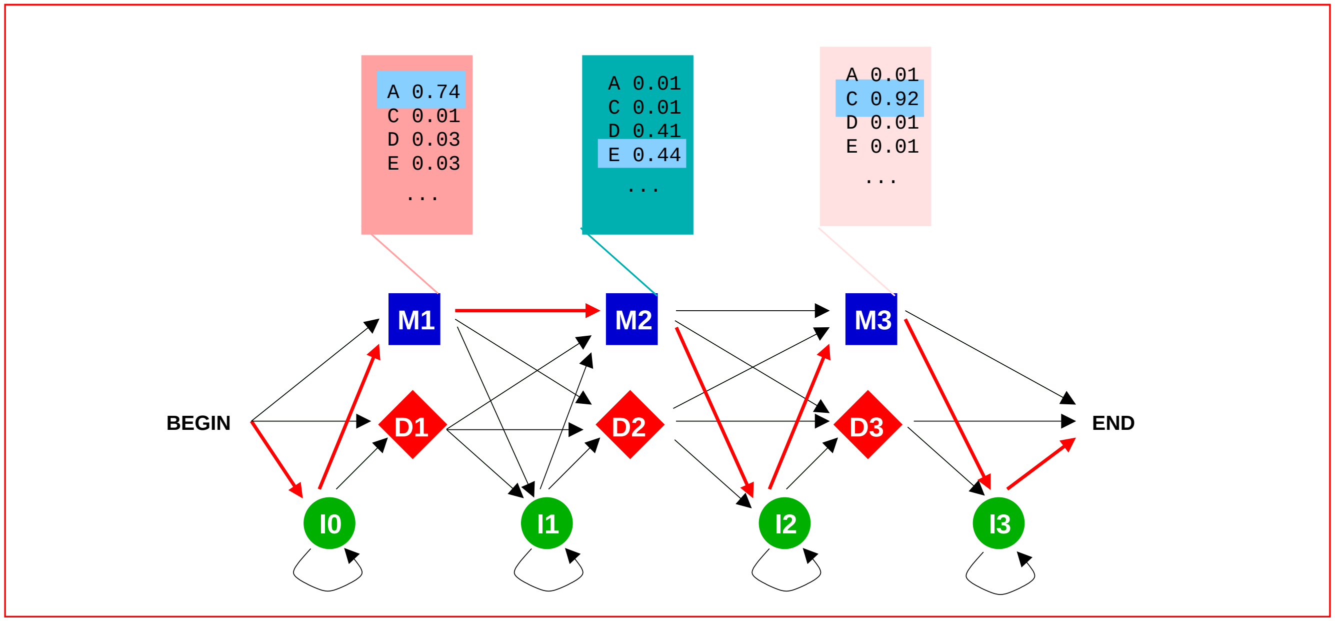
```
A 0.74
C 0.01
D 0.03
E 0.03
...
```

```
A 0.01
C 0.01
D 0.41
E 0.44
...
```

```
A 0.01
C 0.92
D 0.01
E 0.01
...
```

Match a sequence to a model: find the best path





Algorithms associated with HMMs

- Three important questions can be answered by three algorithms.
  - How likely is a given sequence under a given model?
    - This is the scoring problem and it can be solved using the *Forward algorithm*.
  - What is the most probable path between states of a model given a sequence?
    - This is the alignment problem and it is solved by the *Viterbi algorithm*.
  - How can we learn the HMM parameters given a set of sequences?
    - This is the training problem and is solved using the *Forward-backward algorithm* and the *Baum-Welch expectation maximization*.

- For details about these algorithms see:

  Durbin, Eddy, Mitchison, Krog.
  Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids.
  Cambridge University Press, 1998.

HMMs: Softwares

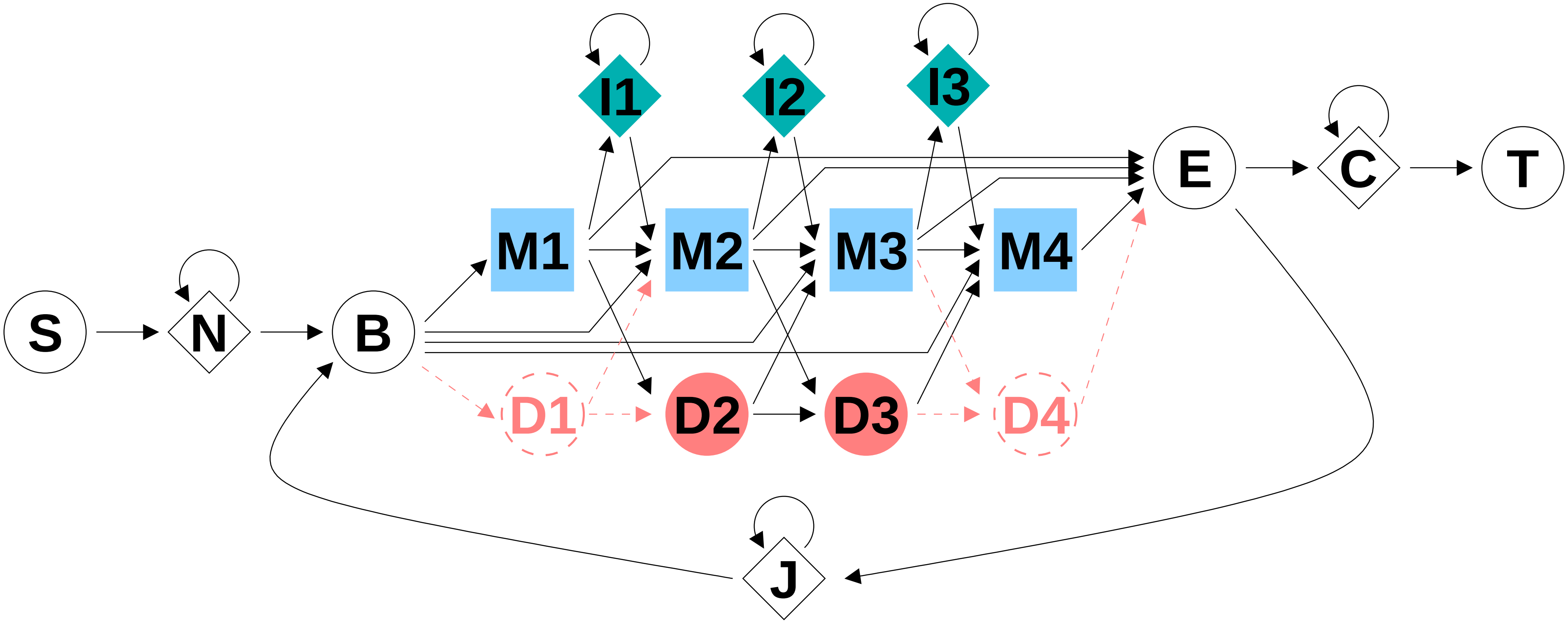
- *HMMER2* is a package to build and use HMMs developed by Sean Eddy (http://hmmer.wustl.edu/).

- Software available in HMMER2:

  - *hmmbuild* to build an HMM model from a multiple alignment;
  - *hmmalign* to align sequences to an HMM model;
  - *hmmcalibrate* to calibrate an HMM model;
  - *hmmemit* to create sequences from an HMM model;
  - *hmmsearch* to search a sequence database with an HMM model;
  - *hmmpfam* to scan a sequence with a database of HMM models;
  - ...

- *SAM* is a similar package developed by Richard Hughey, Kevin Karplus and Anders Krogh (http://www.cse.ucsc.edu/research/compbio/sam.html).

The "Plan 7" architecture of HMMER2



HMMs: Conclusions

- Solid theoretical basis in the theory of probabilities.
- Other advantages and limitations just like generalized profiles.

Generalized profiles and HMMs I

- Generalized profiles are *equivalent* to the 'linear' HMMs like those of SAM or HMMER2 (they are not equivalent to other HMMs of more complicated architecture).

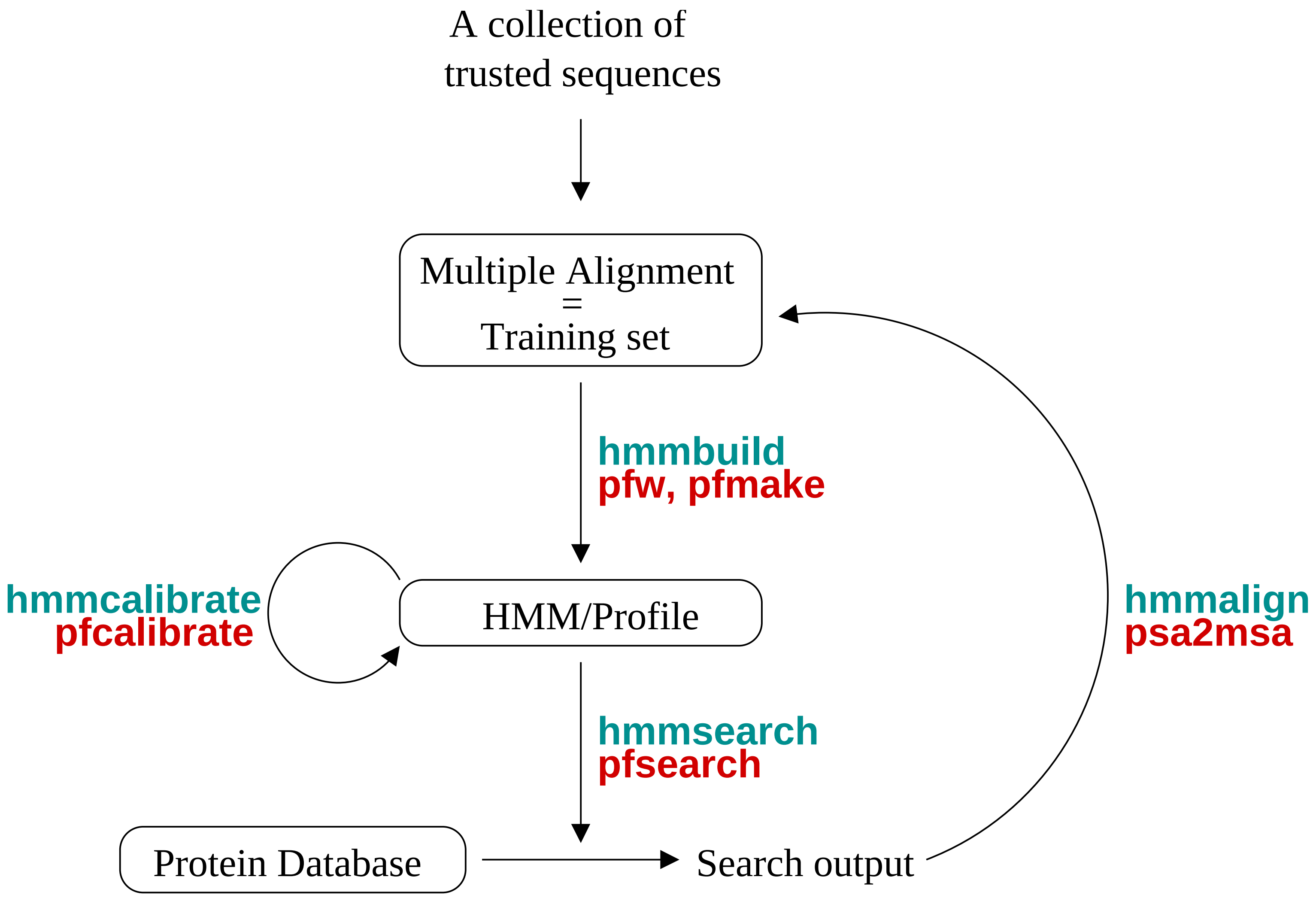
- The optimal alignment produced by dynamical programming is *equivalent* to the Viterbi path on a HMM.

- There are programs to translate generalized profiles from and into HMMs:
  - *htop*: HMM to profile.
  - *ptoh*: profile to HMM.

- Possible manual tuning of Generalized profiles (by a well trained expert). This is very difficult with HMMs.

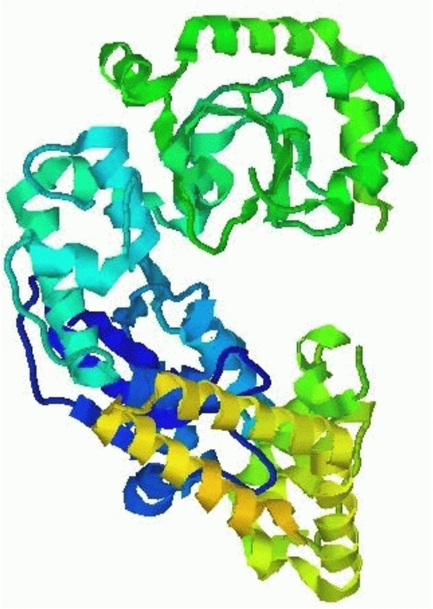
Generalized profiles and HMMs II

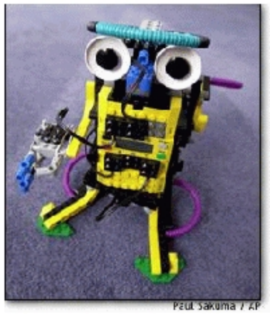
- Iterative model training with the PFTOOLS or HMMER2:



Generalized profiles and HMMs III

- HMMs and generalized profiles are very appropriate for the modeling of protein domains.
- What are protein domains:
  - Domains are discrete structural units (25-500 aa).
  - Short domains (25-50 aa) are present in multiple copies for structural stability.
  - Domains are functional units.





Paul Sakuma / AP

Position Specific Iterative BLAST (PSI-BLAST)

PSI-BLAST principle

- PSSM could have simply been improved by the introduction of a position-independent affine gap cost model. This is less sophistication than the generalized profiles, but it is just this principle that is behind *PSI-BLAST*.

- PSI-BLAST principle:

  1 A standard BLAST search is performed against a database using a substitution matrix (e.g. BLOSUM62).

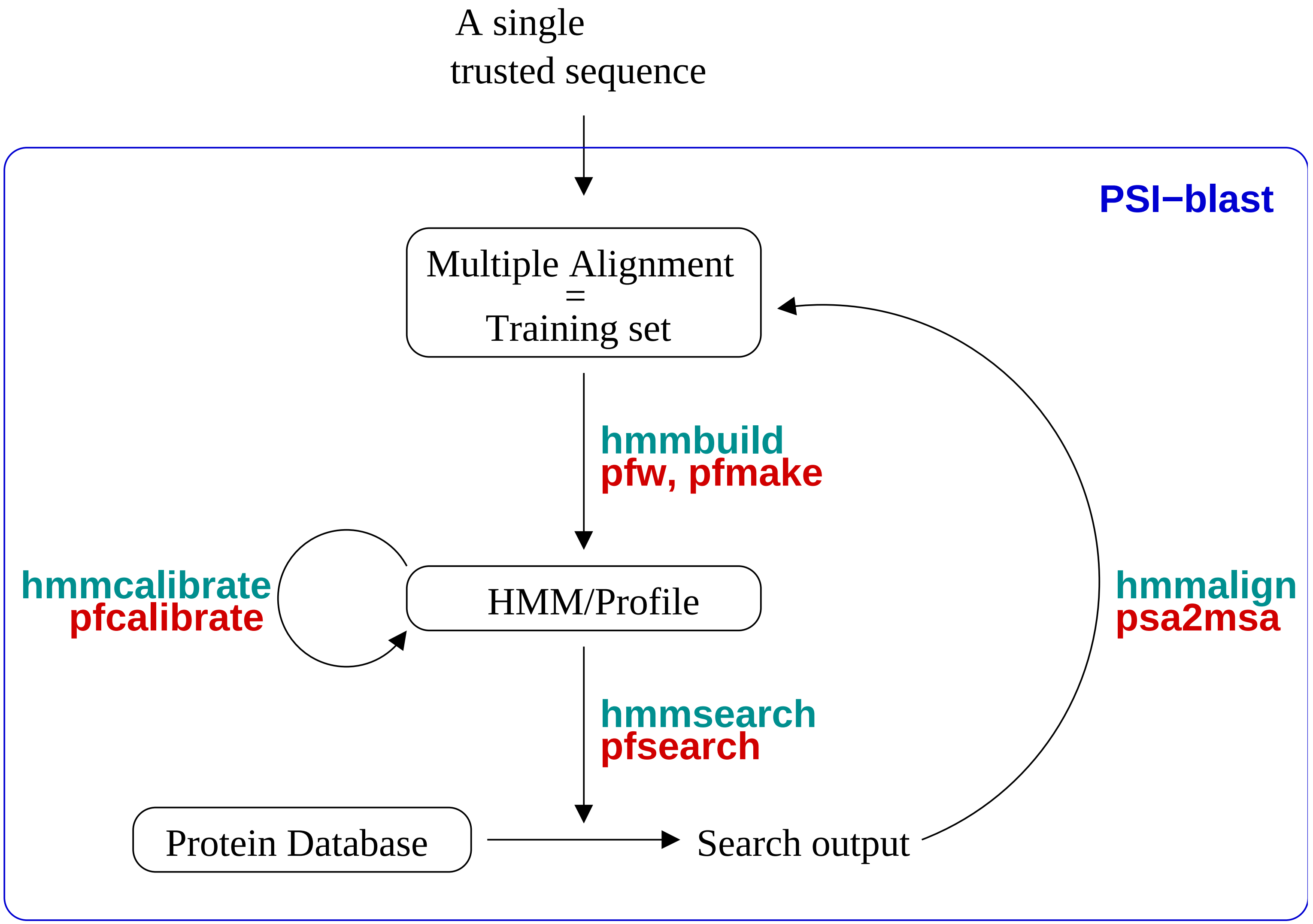
  2 A PSSM (*checkpoint*) is constructed automatically from a multiple alignment of the highest scoring hits of the initial BLAST search. High conserved positions receive high scores and weakly conserved positions receive low scores.

  3 The PSSM replaces the initial matrix (e.g. BLOSUM62) to perform a second BLAST search.

  4 Steps 3 and 4 can be repeated and the new found sequences included to build a new PSSM.

  5 We say that the PSI-BLAST has *converged* if no new sequences are included in the last cycle.

PSI-BLAST, Generalized profiles, and HMMs



PSI-BLAST vs BLAST

- Because of its cycling nature, PSI-BLAST allow to find more *distant homologous* than a simple BLAST search.

- PSI-BLAST uses two E-values:

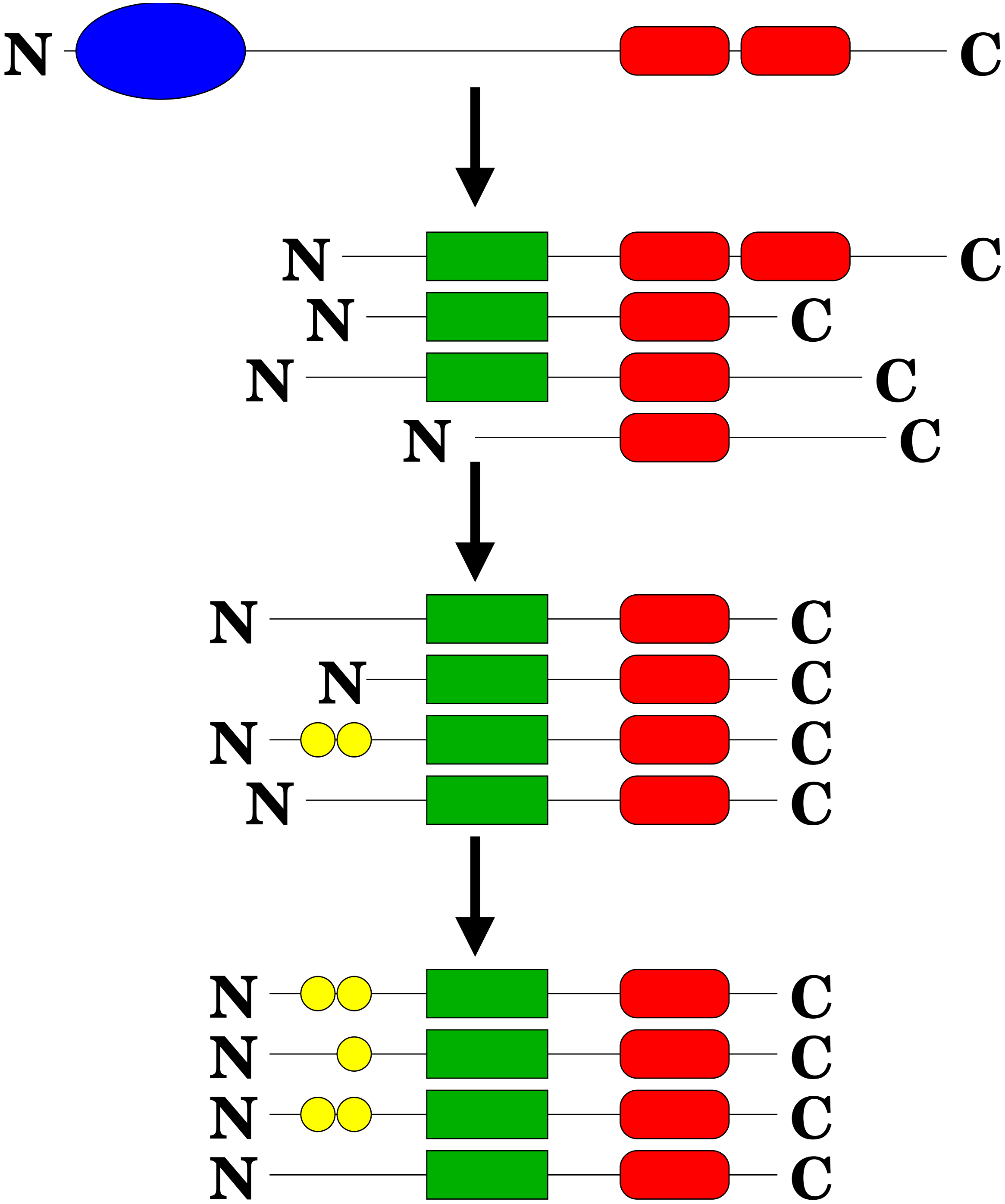
  - the *threshold* E-value for the initial BLAST (-e option). The default is 10 as in the standard BLAST;
  - the *inclusion* E-value to accept sequences (-h option) in the PSSM construction (default is 0.001).

PSI-BLAST advantages

- Fast because of the BLAST *heuristic*.
- Allows PSSMs searches on large databases.
- A particularly efficient algorithm for sequence weighting.
- A very sophisticated statistical treatment of the match scores.
- Single software.
- User friendly interface.

PSI-BLAST danger

- Avoid too close sequences ⇒ overfit!
- Can include false homologous! Therefore check the matches carefully: include or exclude sequences based on biological knowledge.
- The E-value reflects the significance of the match to the previous training set not to the original sequence!
- Choose carefully your query sequence.
- Try reverse experiment to certify.



WRONG ANNOTATION!

Databases

Patterns database: *Prosite*

- *Prosite* is a database containing patterns and profiles:
  - WEB access: http://www.expasy.ch/prosite/.
  - Well documented.
  - Easy to test new patterns.
  - Patterns length typically around 10-20 aa.
- Patterns in Prosite contain a number of useful information:
  - A *quality* estimation by counting the number of true positives (TP), false negatives (FN), and false positives (FP) in SWISS-PROT.
  - Taxonomic range:

A	Archaea
B	Bacteriophages
E	Eukaryota
P	Procaryota
V	Viruses

  - A SWISS-PROT match-list. This list is absent if the profile is too short or too degenerated to return significant results (SKIP_FLAG = TRUE).

Patterns database: *Prosite*

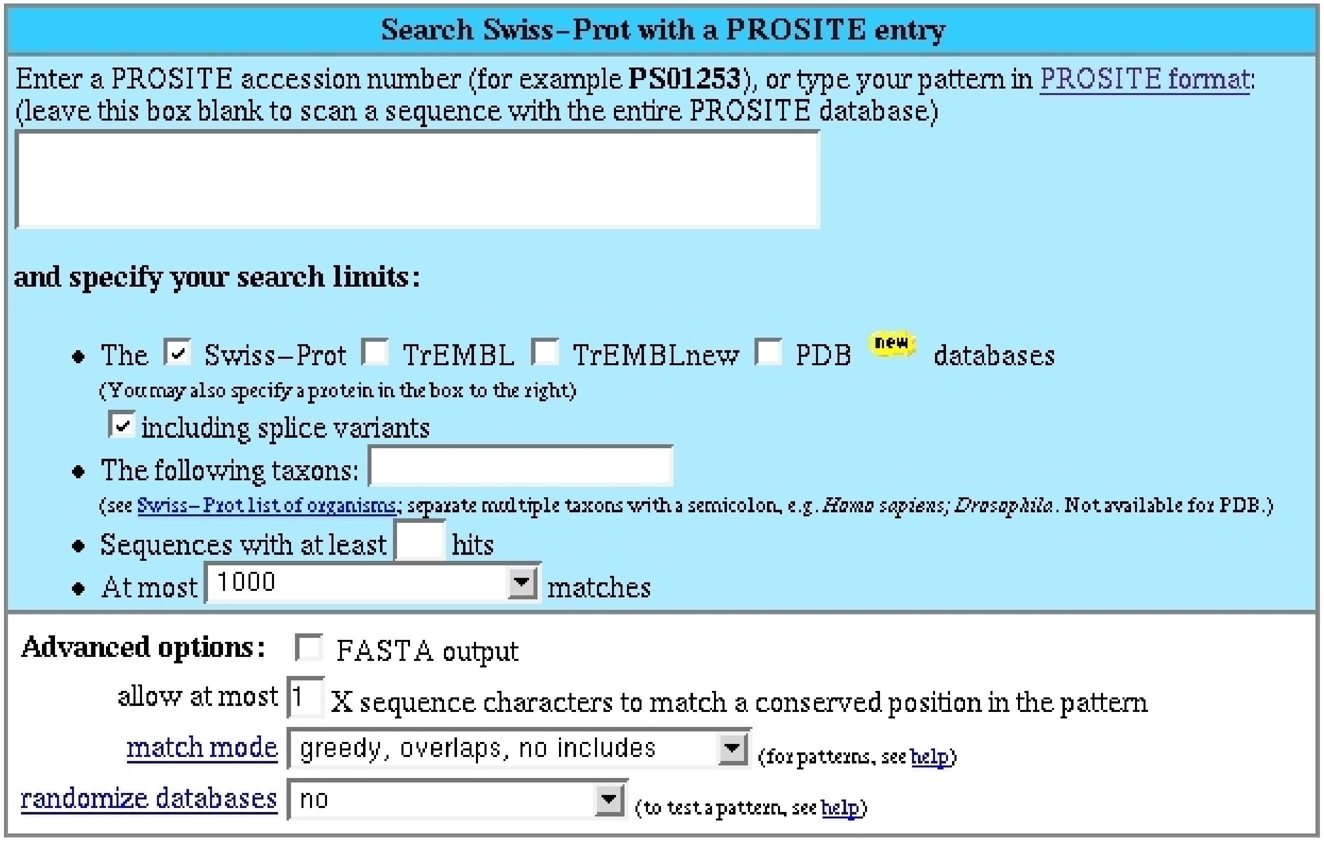
```
ID   UCH_2_1; PATTERN.
AC   PS00972;
DT   JUN-1994 (CREATED); DEC-2001 (DATA UPDATE); DEC-2001 (INFO UPDATE).
DE   Ubiquitin carboxyl-terminal hydrolases family 2 signature 1.
PA   G-[LIVMFY]-x(1,3)-[AGC]-[NASM]-x-C-[FYW]-[LIVMFC]-[NST]-[SACV]-x-[LIVMS]-
PA   Q.
NR   /RELEASE=40.7,103373;
NR   /TOTAL=58(58); /POSITIVE=58(58); /UNKNOWN=0(0); /FALSE_POS=0(0);
NR   /FALSE_NEG=5; /PARTIAL=1;
CC   /TAXO-RANGE=??E??; /MAX-REPEAT=1;
CC   /SITE=7,active_site(?);
DR   P55824, FAF_DROME , T; Q93008, FAFX_HUMAN, T; P70398, FAFX_MOUSE, T;
DR   O00507, FAFY_HUMAN, T; P54578, TGT_HUMAN , T; P40826, TGT_RABIT , T;
(...)
DR   Q99MX1, UBPQ_MOUSE, T; Q61068, UBPW_MOUSE, T; P34547, UBPX_CAEEL, T;
DR   Q09931, UBPY_CAEEL, T;
DR   Q01988, UBPB_CANFA, P;
DR   P53874, UBPA_YEAST, N; Q9UMW8, UBPI_HUMAN, N; Q9WTV6, UBPI_MOUSE, N;
DR   Q9UPU5, UBPO_HUMAN, N; Q17361, UBPT_CAEEL, N;
DO   PDOC00750;
//
```

Patterns database: *Prosite*

```
{PDOC00750}
{PS00972; UCH_2_1}
{PS00973; UCH_2_2}
{PS50235; UCH_2_3}
{BEGIN}
**********************************************************************
* Ubiquitin carboxyl-terminal hydrolases family 2 signatures/profile *
**********************************************************************
Ubiquitin  carboxyl-terminal  hydrolases (EC 3.1.2.15) (UCH) (deubiquitinating
enzymes) [1,2] are thiol proteases  that recognize  and  hydrolyze the peptide
bond at the C-terminal glycine of ubiquitin. These enzymes are involved in the
processing of  poly-ubiquitin  precursors  as  well  as  that  of  ubiquinated
proteins. There are  two  distinct  families  of  UCH. The second class consist of large
proteins (800 to 2000 residues) and is currently represented by:  - Yeast UBP1, UBP2, UBP3, UBP4 (or DOA4/SSV7), UBP5, U
   UBP11, UBP12, UBP13, UBP14, UBP15 and UBP16.
 - Human tre-2.
 - Human isopeptidase T.
 - Human isopeptidase T-3.
 - Mammalian Ode-1.
 - Mammalian Unp.
 - Mouse Dub-1.
 - Drosophila fat facets protein (gene faf).
 - Mammalian faf homolog.
 - Drosophila D-Ubp-64E.
 - Caenorhabditis elegans hypothetical protein R10E11.3.
 - Caenorhabditis elegans hypothetical protein K02C4.3.
These proteins only share two regions of similarity. The first region contains
a conserved cysteine which is probably implicated  in the catalytic mechanism.
The second region contains two conserved histidines residues,  one of which is
(...)
```

Patterns database: *Prosite*

- *ScanProsite* is a tool to scan a database with Prosite or user-build patterns (http://www.expasy.org/tools/scanprosite/):



Search Swiss–Prot with a PROSITE entry

Enter a PROSITE accession number (for example **PS01253**), or type your pattern in PROSITE format:
(leave this box blank to scan a sequence with the entire PROSITE database)

and specify your search limits:

- The ☑ Swiss–Prot □ TrEMBL □ TrEMBLnew □ PDB new databases
  (You may also specify a protein in the box to the right)
  ☑ including splice variants
- The following taxons:
  (see Swiss–Prot list of organisms; separate multiple taxons with a semicolon, e.g. *Homo sapiens; Drosophila*. Not available for PDB.)
- Sequences with at least hits
- At most 1000 matches

Advanced options: □ FASTA output

allow at most 1 X sequence characters to match a conserved position in the pattern

match mode greedy, overlaps, no includes (for patterns, see help)

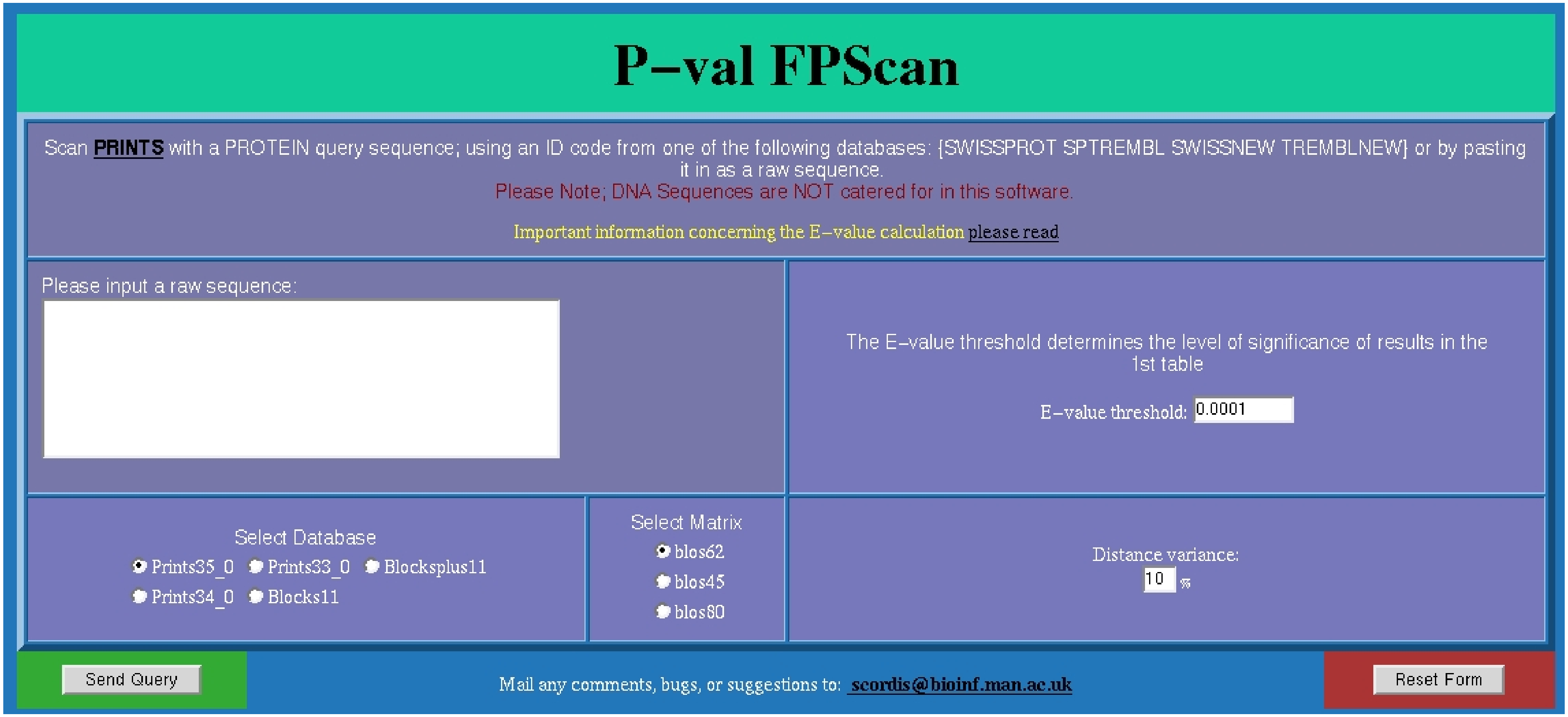
randomize databases no (to test a pattern, see help)

PSSM databases: *PRINTS*

- Collection of conserved motifs used to characterize a protein.
- Uses fingerprints (conserved motif groups).
- Very good to describe sub-families.
- Release 35.0 of PRINTS contains 1750 entries, encoding 10626 individual motifs.
- http://bioinf.man.ac.uk/dbbrowser/PRINTS.
- *BLOCKS* is another PSSMs database similar to prints (http://www.blocks.fhcrc.org/).

PSSM databases: *PRINTS*

- Example: the *PRINTS* database search page (http://bioinf.man.ac.uk/dbbrowser/PRINTS):



P–val FPScan

Scan **PRINTS** with a PROTEIN query sequence; using an ID code from one of the following databases: {SWISSPROT SPTREMBL SWISSNEW TREMBLNEW} or by pasting it in as a raw sequence.

Please Note; DNA Sequences are NOT catered for in this software.

Important information concerning the E–value calculation please read

Please input a raw sequence:

The E–value threshold determines the level of significance of results in the 1st table

E–value threshold: 0.0001

Select Database

● Prints35_0 ○ Prints33_0 ○ Blocksplus11

○ Prints34_0 ○ Blocks11

Select Matrix

● blos62

○ blos45

○ blos80

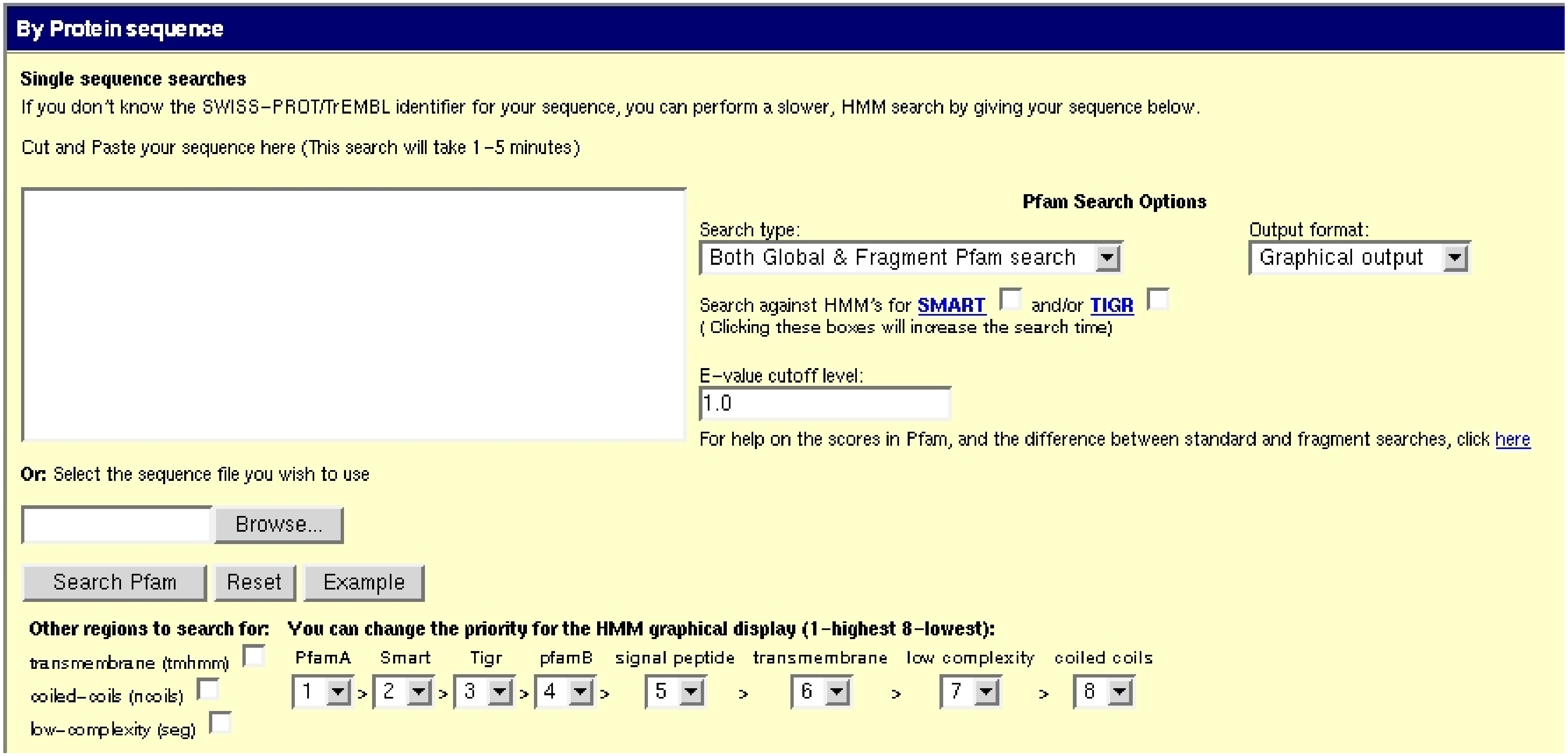
Distance variance: 10 %

Mail any comments, bugs, or suggestions to: scordis@bioinf.man.ac.uk

Protein domain databases: *Pfam*

- Collection of protein domains and families (5049 entries in Pfam release 7.8).
- Uses HMMs (HMMER2).
- Good links to structure, taxonomy.
- http://www.sanger.ac.uk/Pfam.

Protein domain databases: *Pfam*



By Protein sequence

Single sequence searches

If you don't know the SWISS-PROT/TrEMBL identifier for your sequence, you can perform a slower, HMM search by giving your sequence below.

Cut and Paste your sequence here (This search will take 1-5 minutes)

Pfam Search Options

Search type: Both Global & Fragment Pfam search

Output format: Graphical output

Search against HMM's for **SMART** □ and/or **TIGR** □
(Clicking these boxes will increase the search time)

E-value cutoff level: 1.0

For help on the scores in Pfam, and the difference between standard and fragment searches, click here

Or: Select the sequence file you wish to use

Other regions to search for:

transmembrane (tmhmm) □
coiled-coils (ncoils) □
low-complexity (seg) □

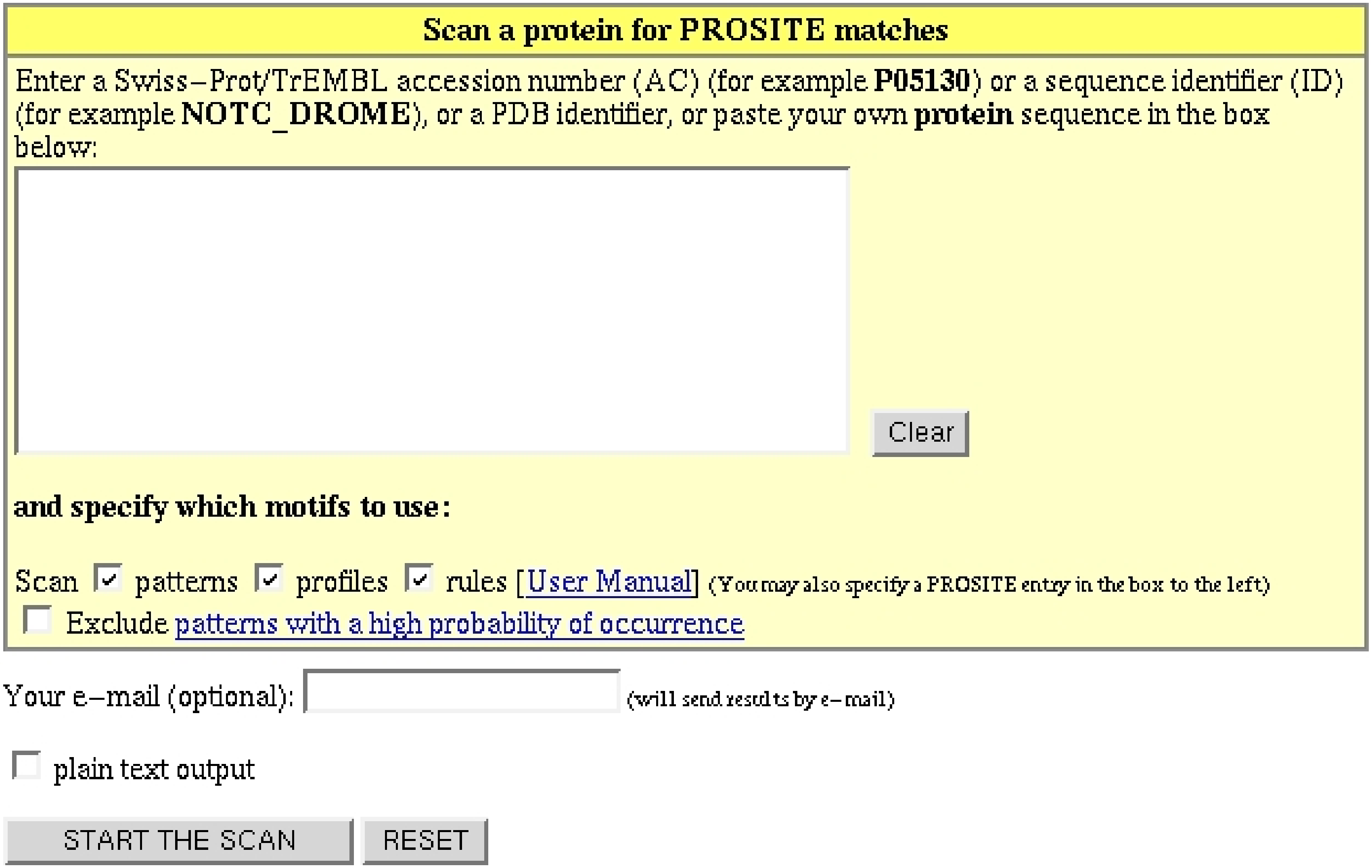
You can change the priority for the HMM graphical display (1-highest 8-lowest):

PfamA		Smart		Tigr		pfamB		signal peptide		transmembrane		low complexity		coiled coils
1	>	2	>	3	>	4	>	5	>	6	>	7	>	8

Protein domain databases: *Prosite*

- Collection of motifs, protein domains, and families (1594 patterns, rules and profiles/matrices in Prosite release 17.34).
- Uses generalized profiles (Pftools) and patterns.
- High quality documentation.
- http://www.expasy.ch/prosite.

Profiles databases: *Prosite*



Scan a protein for PROSITE matches

Enter a Swiss–Prot/TrEMBL accession number (AC) (for example **P05130**) or a sequence identifier (ID) (for example **NOTC_DROME**), or a PDB identifier, or paste your own **protein** sequence in the box below:

and specify which motifs to use:

Scan ☑ patterns ☑ profiles ☑ rules [User Manual] (You may also specify a PROSITE entry in the box to the left)

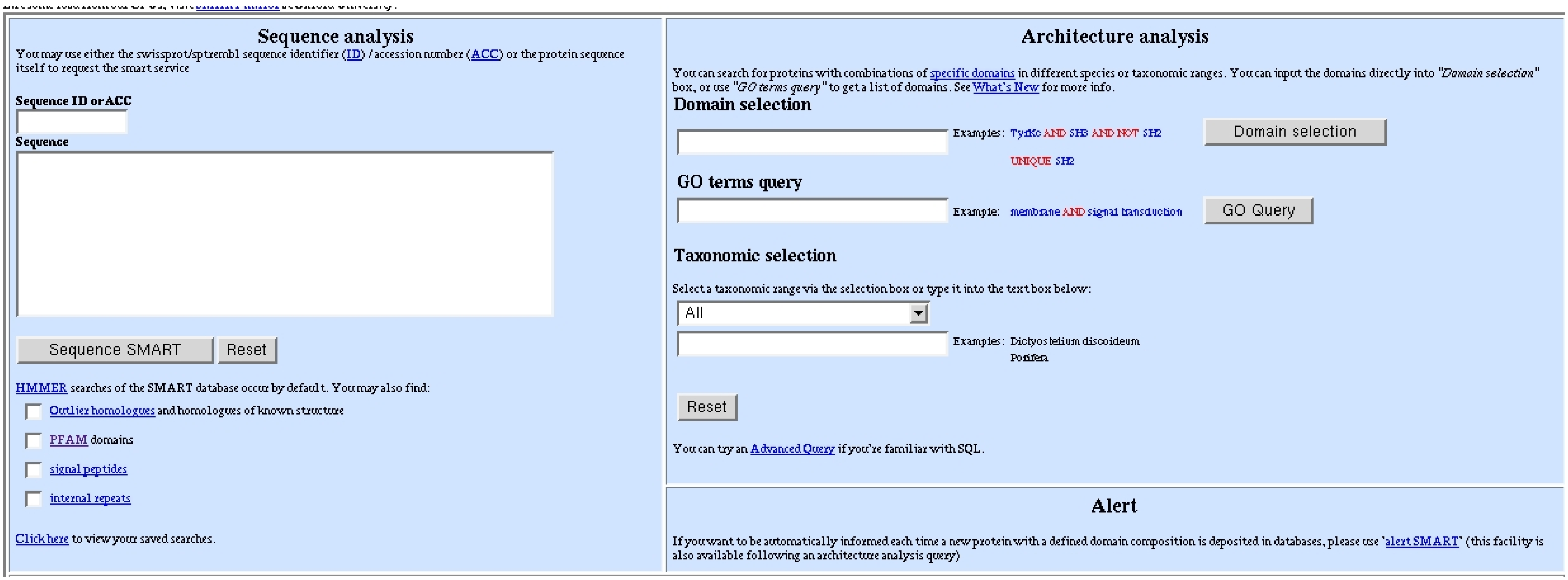
□ Exclude patterns with a high probability of occurrence

Your e–mail (optional): (will send results by e–mail)

□ plain text output

Protein domain databases: *Smart*

- Collection of protein domains (652 domains in version 3.4).
- Uses HMMs and HMMER2.
- Excellent graphic interface.
- Excellent taxonomic information.
- Easy to search meta-motifs.
- http://smart.embl-heidelberg.de



Sequence analysis

You may use either the swissprot/sptrembl sequence identifier (ID) / accession number (ACC) or the protein sequence itself to request the smart service

Sequence ID or ACC

Sequence

HMMER searches of the SMART database occur by default. You may also find:

□ Outlier homologues and homologues of known structure

□ PFAM domains

□ signal peptides

□ internal repeats

Click here to view your saved searches.

Architecture analysis

You can search for proteins with combinations of specific domains in different species or taxonomic ranges. You can input the domains directly into *"Domain selection"* box, or use *"GO terms query"* to get a list of domains. See What's New for more info.

Domain selection

Examples: TyrKc AND SH3 AND NOT SH2

UNIQUE SH2

GO terms query

Example: membrane AND signal transduction

Taxonomic selection

Select a taxonomic range via the selection box or type it into the text box below:

All

Examples: Dictyostelium discoideum

Porifera

You can try an Advanced Query if you're familiar with SQL.

Alert

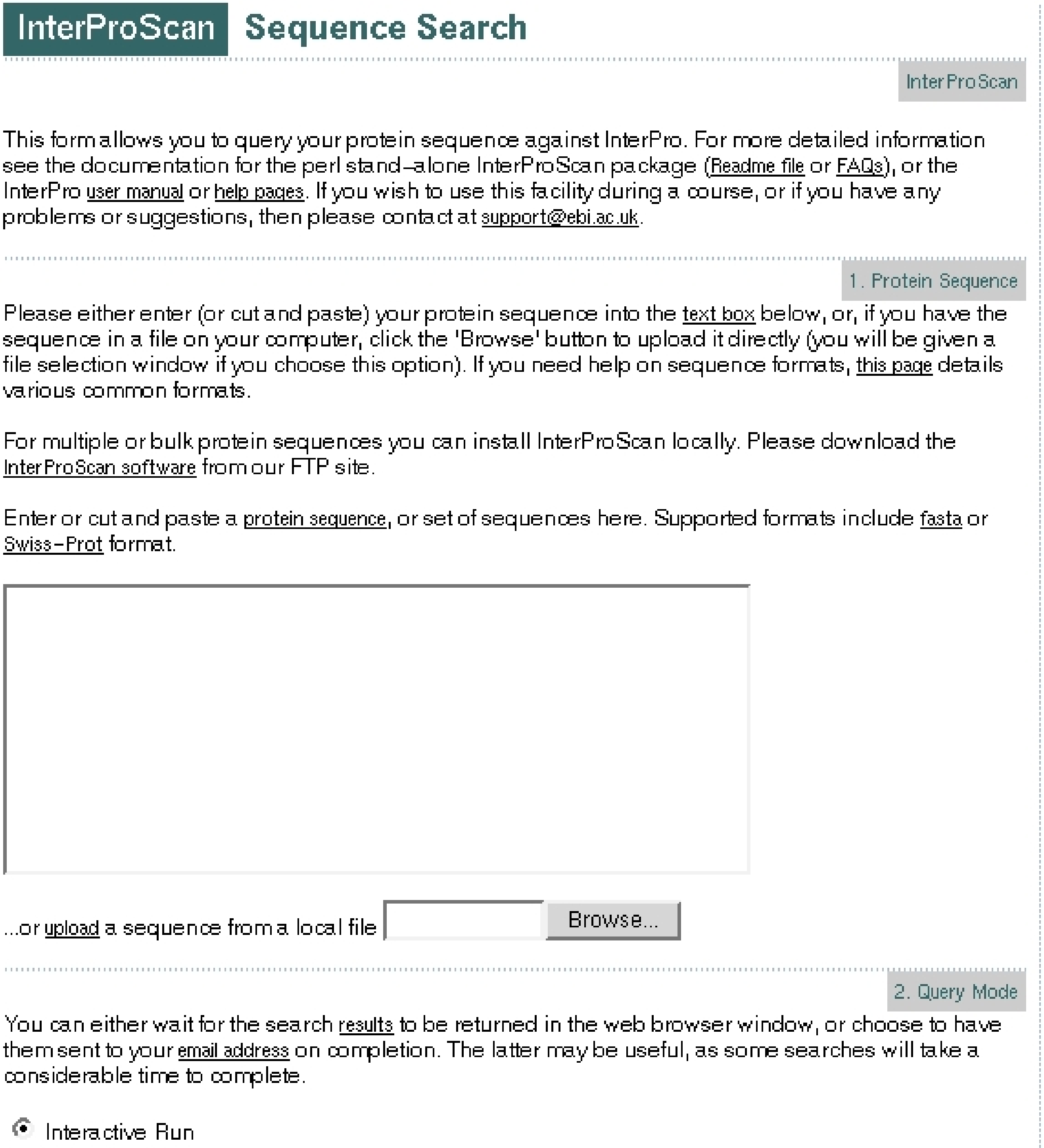
If you want to be automatically informed each time a new protein with a defined domain composition is deposited in databases, please use 'alert SMART' (this facility is also available following an architecture analysis query)

Protein domain databases: *ProDom*

- http://prodes.toulouse.inra.fr/prodom/doc/prodom.html.
- Collection of protein motifs obtained automatically using PSI-BLAST.
- Very high throughput ... but no annotation.
- ProDom release 2001.3 contains 108076 families (at least 2 sequences per family).

Protein domain databases: *InterPro*

- InterPro is an attempt to group a number of protein domain databases:
  - Pfam
  - PROSITE
  - PRINTS
  - ProDom
  - SMART
  - TIGRFAMs
- InterPro tries to have and maintain a high quality annotation.
- Very good accession to examples.
- InterPro web site: http://www.ebi.ac.uk/interpro.
- The database and a stand-alone package (*iprscan*) are available for UNIX platforms to locally run a complete Interpro analysis: ftp://ftp.ebi.ac.uk/pub/databases/interpro.



InterProScan Sequence Search

InterProScan

This form allows you to query your protein sequence against InterPro. For more detailed information see the documentation for the perl stand-alone InterProScan package (Readme file or FAQs), or the InterPro user manual or help pages. If you wish to use this facility during a course, or if you have any problems or suggestions, then please contact at support@ebi.ac.uk.

1. Protein Sequence

Please either enter (or cut and paste) your protein sequence into the text box below, or, if you have the sequence in a file on your computer, click the 'Browse' button to upload it directly (you will be given a file selection window if you choose this option). If you need help on sequence formats, this page details various common formats.

For multiple or bulk protein sequences you can install InterProScan locally. Please download the InterProScan software from our FTP site.

Enter or cut and paste a protein sequence, or set of sequences here. Supported formats include fasta or Swiss-Prot format.

...or upload a sequence from a local file

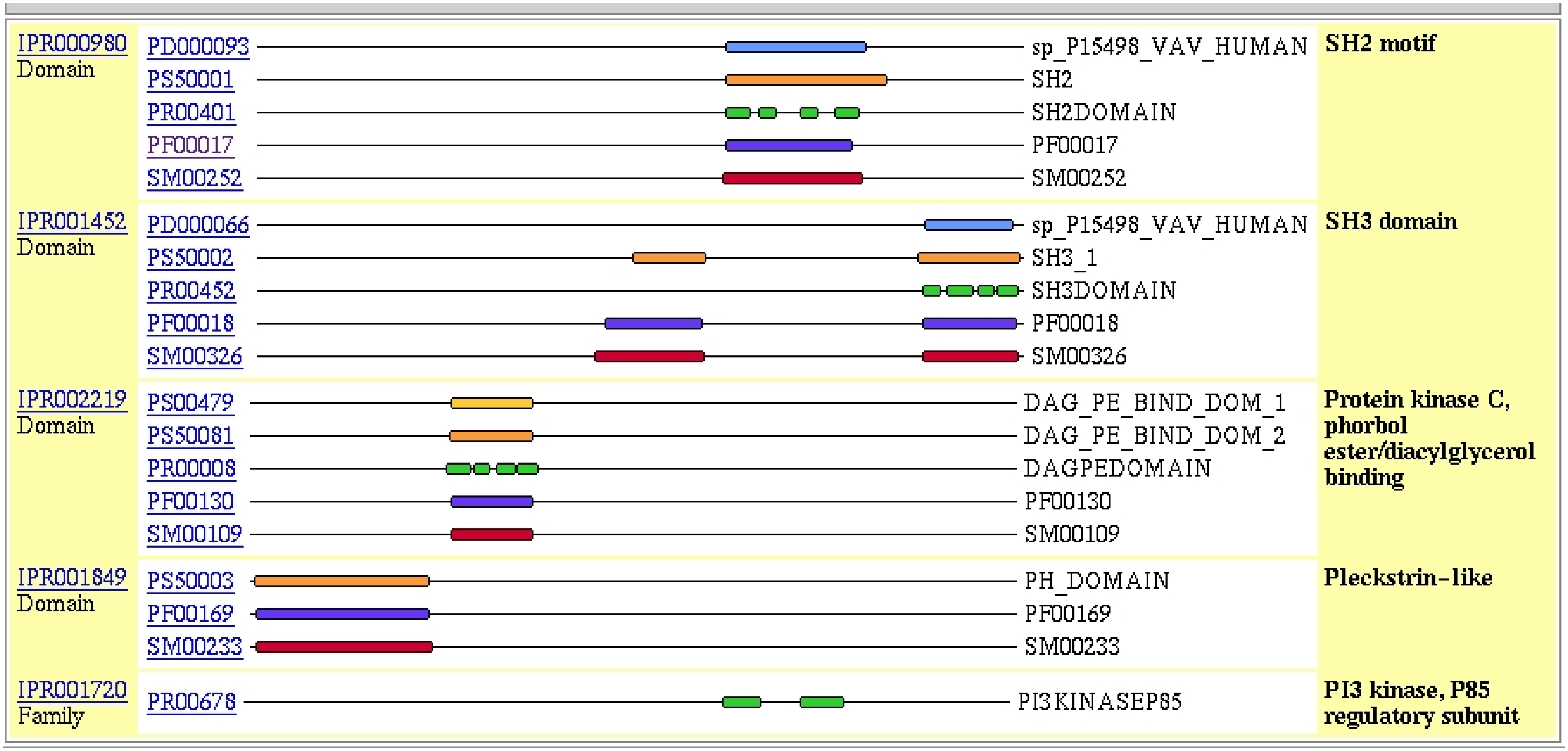
2. Query Mode

You can either wait for the search results to be returned in the web browser window, or choose to have them sent to your email address on completion. The latter may be useful, as some searches will take a considerable time to complete.

● Interactive Run

Protein domain databases: *InterPro*

- Example of a graphical output:



IPR000980 Domain	PD000093	sp_P15498_VAV_HUMAN	SH2 motif
	PS50001	SH2	
	PR00401	SH2DOMAIN	
	PF00017	PF00017	
	SM00252	SM00252	
IPR001452 Domain	PD000066	sp_P15498_VAV_HUMAN	SH3 domain
	PS50002	SH3_1	
	PR00452	SH3DOMAIN	
	PF00018	PF00018	
	SM00326	SM00326	
IPR002219 Domain	PS00479	DAG_PE_BIND_DOM_1	Protein kinase C, phorbol ester/diacylglycerol binding
	PS50081	DAG_PE_BIND_DOM_2	
	PR00008	DAGPEDOMAIN	
	PF00130	PF00130	
	SM00109	SM00109	
IPR001849 Domain	PS50003	PH_DOMAIN	Pleckstrin-like
	PF00169	PF00169	
	SM00233	SM00233	
IPR001720 Family	PR00678	PI3KINASEP85	PI3 kinase, P85 regulatory subunit

The end