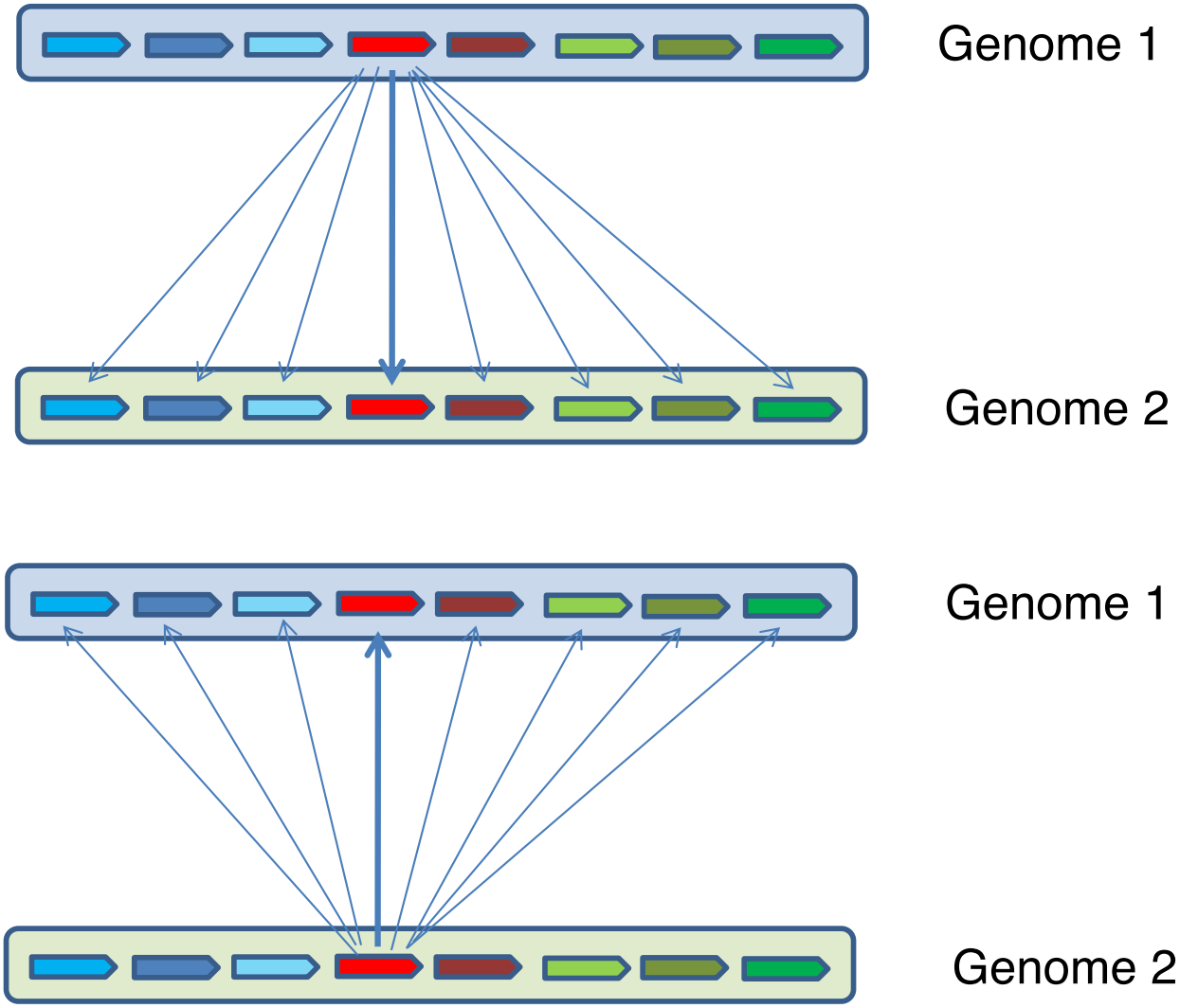
Multiple sequence alignment

Why?

Decipher evolutionary processes to learn about functional constraints

Simple ortholog detection: reciprocal best match



Both genes are a better match to each other than to any other gene in the genome.
Note: all we need for this is pairwise alignments and a way to estimate distance from the alignments.

Family of orthologous sequences

Alignment of trmE (C. *hydrogenoformans*) with mnmE (*E. coli*)

```
 GENE ID: 948222 mnmE | GTPase [Escherichia coli str. K12 substr. MG1655]
(Over 10 PubMed links)

 Score =  282 bits (721),  Expect = 5e-77, Method: Compositional matrix adjust.
 Identities = 169/464 (36%), Positives = 269/464 (57%), Gaps = 20/464 (4%)

Query  4    DTIAAISTPLGEGGIGIVRVSGPGAIEAVKNVFIPRQSKDLSKVPSFTLHYGKIVDPADG  63
            DTI A +TP G GG+GI+R+SG  A E  + V        L K+P            ADG
Sbjct  5    DTIVAQATPPGRGGVGILRISGFKAREVAETV--------LGKLPKPRYADYLPFKDADG  56

Query  64   KIVDEVLVSVMRAPKSYTGEDVVEINCHGGIVAVEKVLELILK-QGIRLAEPGEFTKRAF  122
             ++D+ +      P S+TGEDV+E+  HGG V ++ +L+ IL   G+R+A PGEF++RAF
Sbjct  57   SVLDQGIALWFPGPNSFTGEDVLELQGHGGPVILDLLLKRILTIPGLRIARPGEFSERAF  116
```

Alignment of trmE (C. *hydrogenoformans*) with trmE (*B. anthracis*)

```
 GENE ID: 7784224 trmE | tRNA modification GTPase TrmE
[Bacillus anthracis str. CDC 684]

 Score =  442 bits (1137),  Expect = 2e-125, Method: Compositional matrix adjust.
 Identities = 222/459 (48%), Positives = 328/459 (71%), Gaps = 5/459 (1%)

Query  4    DTIAAISTPLGEGGIGIVRVSGPGAIEAVKNVFIPRQSKDLSKVPSFTLHYGKIVDPADG  63
            DTIAAIST LGEG I IVRVSG  A+E V  +F   + KDL++VPS T+HYG IVD
Sbjct  4    DTIAAISTALGEGAIAIVRVSGDDAVEKVNRIF---KGKDLTEVPSHTIHYGHIVDLDTN  60

Query  64   KIVDEVLVSVMRAPKSYTGEDVVEINCHGGIVAVEKVLELILKQGIRLAEPGEFTKRAFL  123
            ++++EV+VS+MRAP+++T E++VEINCHGG+V+V KVL+LIL QG+RLAEPGEFTKRAFL
Sbjct  61   QVIEEVMVSIMRAPRTFTRENIVEINCHGGLVSVNKVLQLILAQGVRLAEPGEFTKRAFL  120
```

Alignment of trmE (B. anthracis) with mnmE (*E. coli*)

```
 GENE ID: 948222 mnmE | GTPase [Escherichia coli str. K12 substr. MG1655]
(Over 10 PubMed links)

 Score =  246 bits (627),  Expect = 4e-66, Method: Compositional matrix adjust.
 Identities = 157/464 (33%), Positives = 263/464 (56%), Gaps = 23/464 (4%)

Query  4    DTIAAISTALGEGAIAIVRVSGDDAVEKVNRIFKGKDLTEVP-SHTIHYGHIVDLDTNQV  62
            DTI A +T  G G + I+R+SG  A E    +     L ++P      Y    D D   V
Sbjct  5    DTIVAQATPPGRGGVGILRISGFKAREVAETV-----LGKLPKPRYADYLPFKDAD-GSV  58

Query  63   IEEVMVSIMRAPRTFTRENIVEINCHGGLVSVNKVLQLILA-QGVRLAEPGEFTKRAFLN  121
            +++ +      P +FT E+++E+  HGG V ++ +L+ IL   G+R+A PGEF++RAFLN
Sbjct  59   LDQGIALWFPGPNSFTGEDVLELQGHGGPVILDLLLKRILTIPGLRIARPGEFSERAFLN  118
```

Family of orthologous sequences

Alignment of trmE (C. *hydrogenoformans*) with mnmE (*E. coli*)

```
 GENE ID: 948222 mnmE | GTPase [Escherichia coli str. K12 substr. MG1655]
(Over 10 PubMed links)

 Score =  282 bits (721),  Expect = 5e-77, Method: Compositional matrix adjust.
 Identities = 169/464 (36%), Positives = 269/464 (57%), Gaps = 20/464 (4%)

Query  4    DTIAAISTPLGEGGIGIVRVSGPGAIEAVKNVFIPRQSKDLSKVPSFTLHYGKIVDPADG  63
            DTI A +TP G GG+GI+R+SG  A E  + V        L K+P           ADG
Sbjct  5    DTIVAQATPPGRGGVGILRISGFKAREVAETV--------LGKLPKPRYADYLPFKDADG  56

Query  64   KIVDEVLVSVMRAPKSYTGEDVVEINCHGGIVAVEKVLELILK-QGIRLAEPGEFTKRAF  122
             ++D+ +      P S+TGEDV+E+  HGG V ++ +L+ IL   G+R+A PGEF++RAF
Sbjct  57   SVLDQGIALWFPGPNSFTGEDVLELQGHGGPVILDLLLKRILTIPGLRIARPGEFSERAF  116
```

Alignment of trmE (C. *hydrogenoformans*) with trmE (*B. anthracis*)

```
 GENE ID: 7784224 trmE | tRNA modification GTPase TrmE
[Bacillus anthracis str. CDC 684]

 Score =  442 bits (1137),  Expect = 2e-125, Method: Compositional matrix adjust.
 Identities = 222/459 (48%), Positives = 328/459 (71%), Gaps = 5/459 (1%)

Query  4    DTIAAISTPLGEGGIGIVRVSGPGAIEAVKNVFIPRQSKDLSKVPSFTLHYGKIVDPADG  63
            DTIAAIST LGEG I IVRVSG  A+E V  +F   + KDL++VPS T+HYG IVD
Sbjct  4    DTIAAISTALGEGAIAIVRVSGDDAVEKVNRIF---KGKDLTEVPSHTIHYGHIVDLDTN  60

Query  64   KIVDEVLVSVMRAPKSYTGEDVVEINCHGGIVAVEKVLELILKQGIRLAEPGEFTKRAFL  123
            ++++EV+VS+MRAP+++T E++VEINCHGG+V+V KVL+LIL QG+RLAEPGEFTKRAFL
Sbjct  61   QVIEEVMVSIMRAPRTFTRENIVEINCHGGLVSVNKVLQLILAQGVRLAEPGEFTKRAFL  120
```

Alignment of trmE (B. anthracis) with mnmE (*E. coli*)

```
 GENE ID: 948222 mnmE | GTPase [Escherichia coli str. K12 substr. MG1655]
(Over 10 PubMed links)

 Score =  246 bits (627),  Expect = 4e-66, Method: Compositional matrix adjust.
 Identities = 157/464 (33%), Positives = 263/464 (56%), Gaps = 23/464 (4%)

Query  4    DTIAAISTALGEGAIAIVRVSGDDAVEKVNRIFKGKDLTEVP-SHTIHYGHIVDLDTNQV  62
            DTI A +T  G G + I+R+SG  A E    +     L ++P       Y    D D   V
Sbjct  5    DTIVAQATPPGRGGVGILRISGFKAREVAETV-----LGKLPKPRYADYLPFKDAD-GSV  58

Query  63   IEEVMVSIMRAPRTFTRENIVEINCHGGLVSVNKVLQLILA-QGVRLAEPGEFTKRAFLN  121
            +++ +      P +FT E+++E+  HGG V ++ +L+ IL   G+R+A PGEF++RAFLN
Sbjct  59   LDQGIALWFPGPNSFTGEDVLELQGHGGPVILDLLLKRILTIPGLRIARPGEFSERAFLN  118
```

Pairwise alignments

↓

Anchored multiple alignment

```
KNVFIPRQSKDLSK
NRIF---KGKDLTE
ETV--------LGK
```

These turn out to be mutually consistent pairwise alignments.

But in general different pairwise alignments will not be mutually consistent.

The multiple alignment problem

Given a family of sequences, reconstruct the entire evolutionary history of these sequences

This includes:

- Inferring the phylogenetic tree relating the sequences.
- Inferring the sequences at all of the internal nodes.
- Inferring for each branch of the tree how the bases in ancestor and child are related, i.e. a pairwise alignment along each branch of the tree

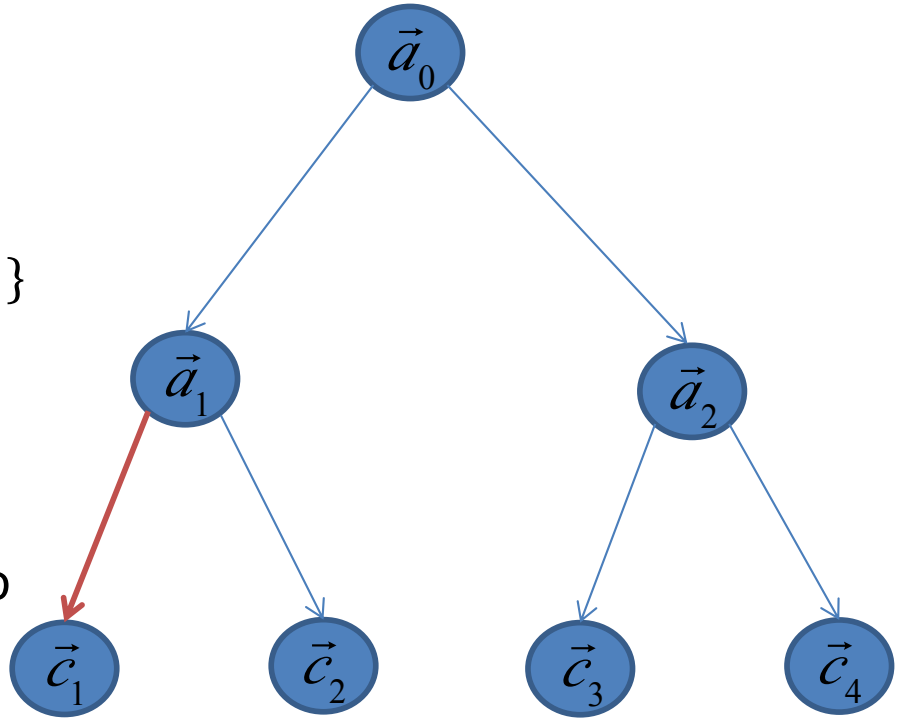
Example:

Leaf sequences: $\{\vec{c_1}, \vec{c_2}, \vec{c_3}, \vec{c_4}\}$

Unknown internal node sequences: $\{\vec{a_0}, \vec{a_1}, \vec{a_2}\}$

Unknown relationship on the branches

We can evaluate the likelihood of any hypothesized alignment of leaf $\vec{c_1}$ sequence to a hypothesized parent sequence $\vec{a_1}$



The full Bayesian solution is a probability distribution over all possible internal node sequences and all possible ways of aligning along each edge.

The multiple alignment problem full Bayesian solution

BIOINFORMATICS

Vol. 17 no. 9 2001
Pages 803–820

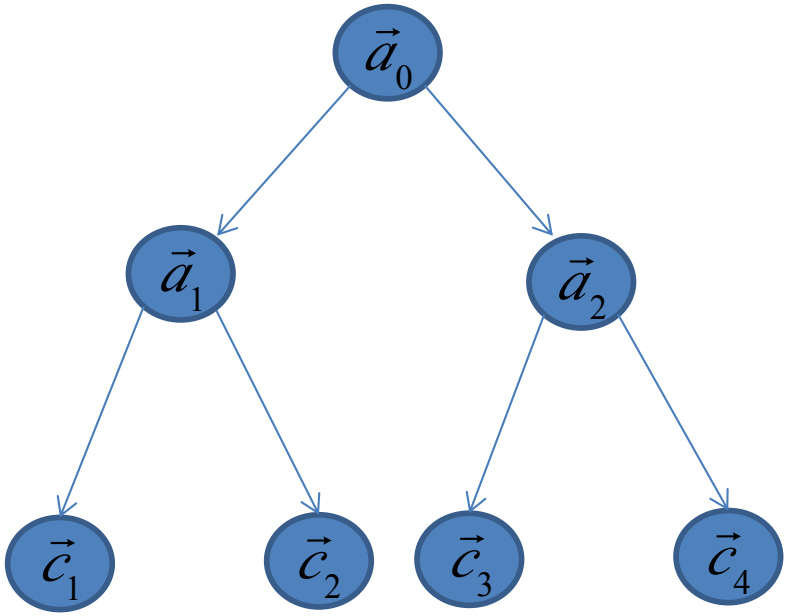
Evolutionary HMMs: a Bayesian approach to multiple alignment

Ian Holmes and William J. Bruno

Group T10, Los Alamos National Laboratory, NM 87545, USA

Received on February 21, 2001; revised and accepted on April 6, 2001

- The algorithm *samples* over all possible ancestral states and alignments at each branch
- It does so by, at each step, keeping everything fixed except for an internal sequence or the alignments along one or two branches
- It then samples a new internal sequence or pairwise alignment by using *pair HMMs* as we discussed before

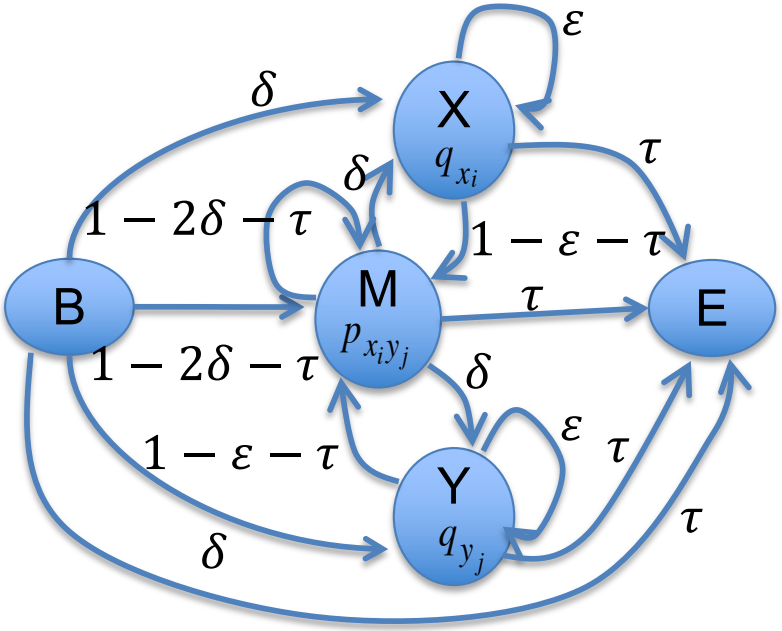


Recall

Applications: Probabilistic sampling of alignments

How to sample alignments?

We have a pair-HMM and we know how to calculate the forward sums $f^k(i,j)$, representing the probabilities of ending the alignments with characters x_i and y_j, in state k.



Etc.

Recall

How to sample alignments?

Traceback through the matrix $f^k(i,j)$ but instead of following the highest scoring move, choose probabilistically. E.g. for a match state we have

$$
\begin{aligned}
&f^M(i,j) \\
&= p_{x_i,y_j}\big[(1-2\delta-\tau)f^M(i-1,j-1) \\
&+ (1-\varepsilon-\tau)\big(f^X(i-1,j-1)+f^Y(i-1,j-1)\big)\big]
\end{aligned}
$$

$M(i-1,j-1)$ with probability $\dfrac{p_{x_i y_j}(1-2\delta-\tau)f^M(i-1,j-1)}{f^M(i,j)}$

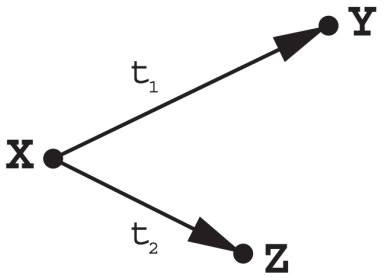
$X(i-1,j-1)$ with probability $\dfrac{p_{x_i y_j}(1-\varepsilon-\tau)f^X(i-1,j-1)}{f^M(i,j)}$

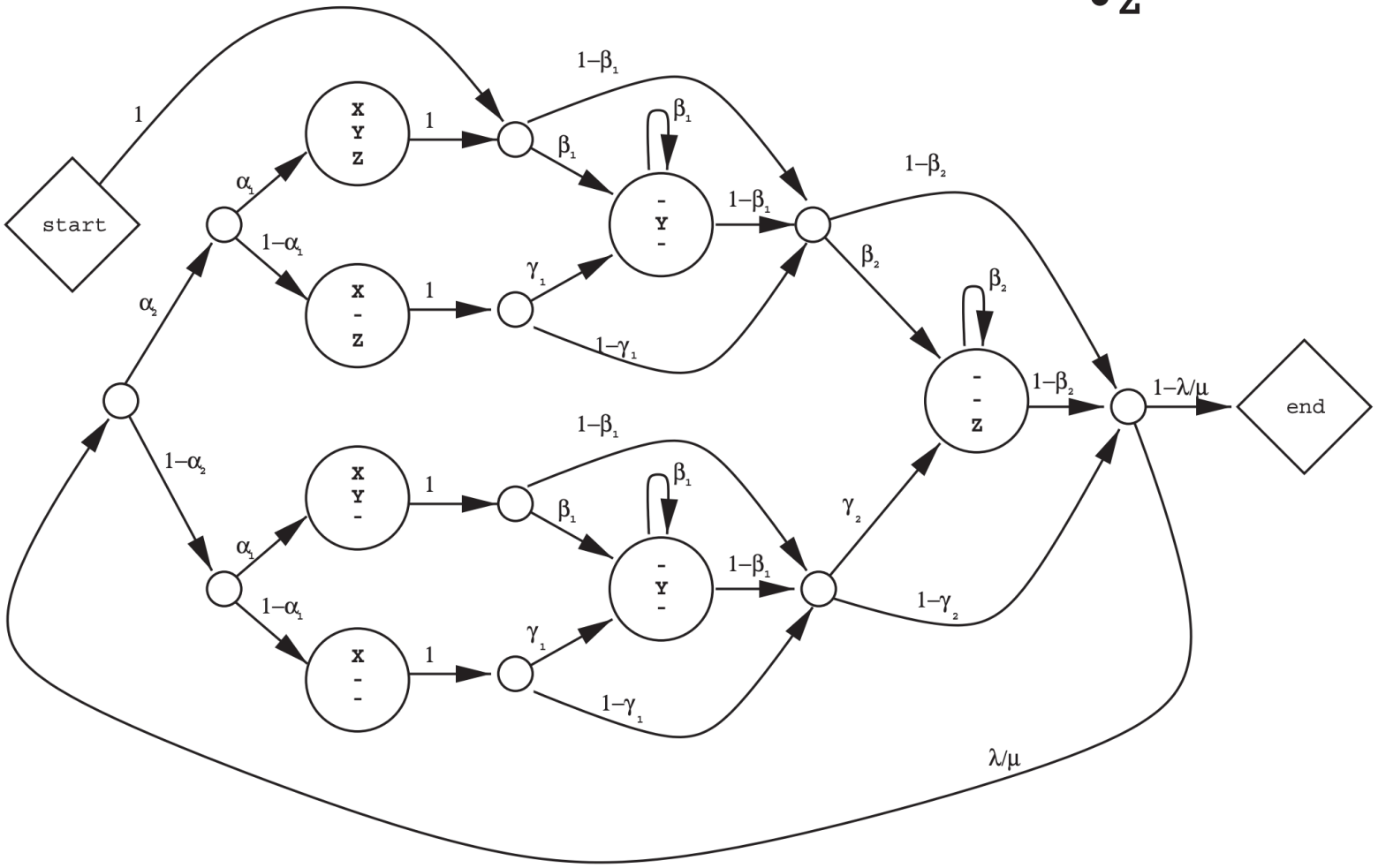
$Y(i-1,j-1)$ with probability $\dfrac{p_{x_i y_j}(1-\varepsilon-\tau)f^M(i-1,j-1)}{f^M(i,j)}$

Etc.

The multiple alignment problem
full Bayesian solution

E.g. alignment of a parent to child sequences





The multiple alignment problem
full Bayesian solution

BIOINFORMATICS

Vol. 17 no. 9 2001
Pages 803–820

Evolutionary HMMs: a Bayesian approach to multiple alignment

Ian Holmes and William J. Bruno

Group T10, Los Alamos National Laboratory, NM 87545, USA

Received on February 21, 2001; revised and accepted on April 6, 2001

Disadvantages:

• The method is computationally very expensive and is thus limited to relatively small data sets
• It can only handle the simplest evolutionary models for insertions/deletions and these may be unrealistic
• Currently these sophisticated methods still produce alignments of less quality than other, more heuristic methods

• The most commonly used and currently most successful algorithms are *progressive alignment* and *profile HMM models*

Progressive multiple alignment

J Mol Evol. 1984;20(2):175-86.

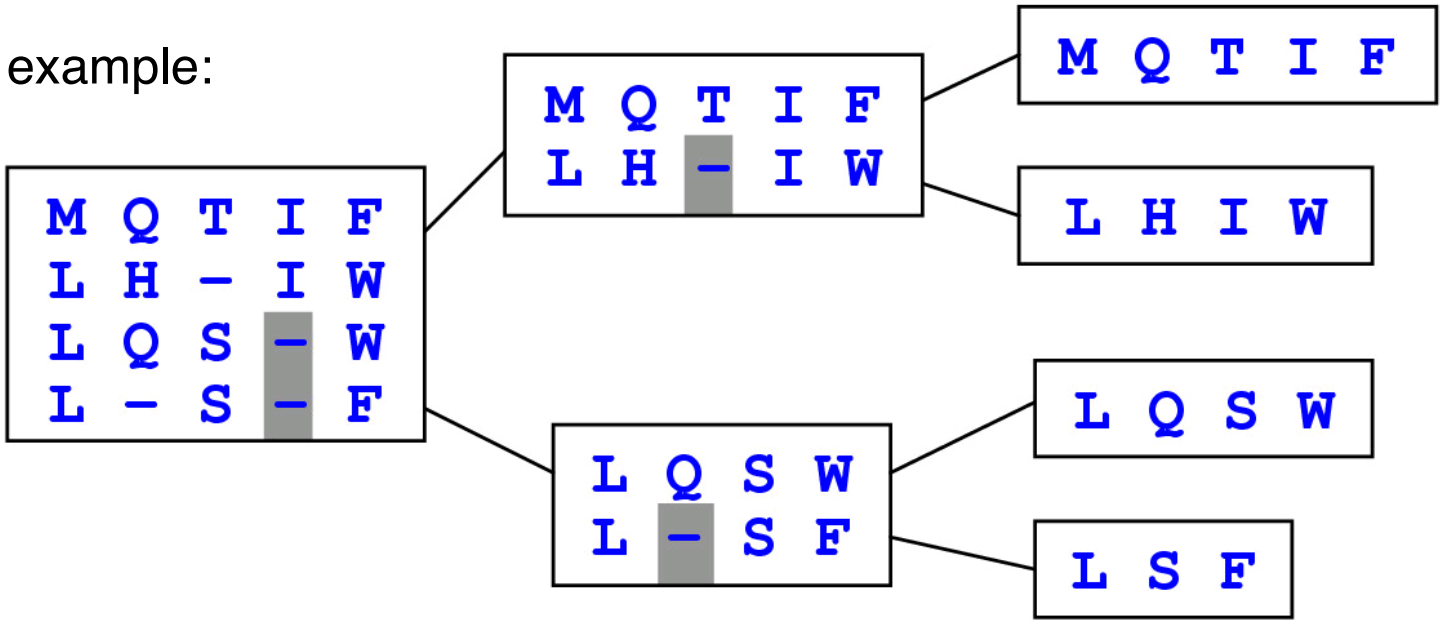
The alignment of sets of sequences and the construction of phyletic trees: an integrated method.

Hogeweg P, Hesper B.

Outline of the approach:

1. For a set of *N* sequences, perform all *N*(*N*-1)/2 pairwise alignments
2. From the pairwise distances, create a phylogenetic tree, for example through neighbor-joining
3. Start at the closest pair of leaves and align them
4. Work one's way "up the tree" by pairwise aligning the closet pair of sequences/alignments that are currently not yet aligned

Cartoon example:



Progressive multiple alignment

J Mol Evol. 1984;20(2):175-86.

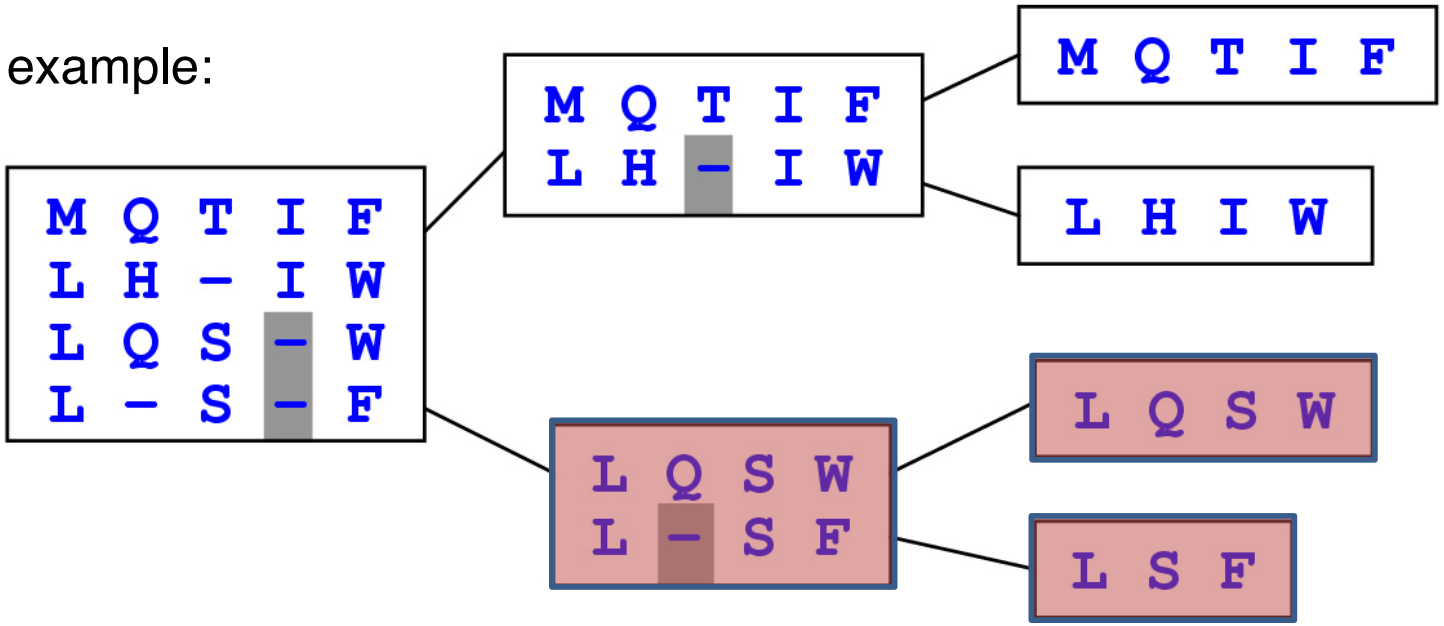
The alignment of sets of sequences and the construction of phyletic trees: an integrated method.

Hogeweg P, Hesper B.

Outline of the approach:

1. For a set of *N* sequences, perform all *N*(*N*-1)/2 pairwise alignments
2. From the pairwise distances, create a phylogenetic tree, for example through neighbor-joining
3. Start at the closest pair of leaves and align them
4. Work one's way "up the tree" by pairwise aligning the closet pair of sequences/alignments that are currently not yet aligned

Cartoon example:



Progressive multiple alignment

J Mol Evol. 1984;20(2):175-86.

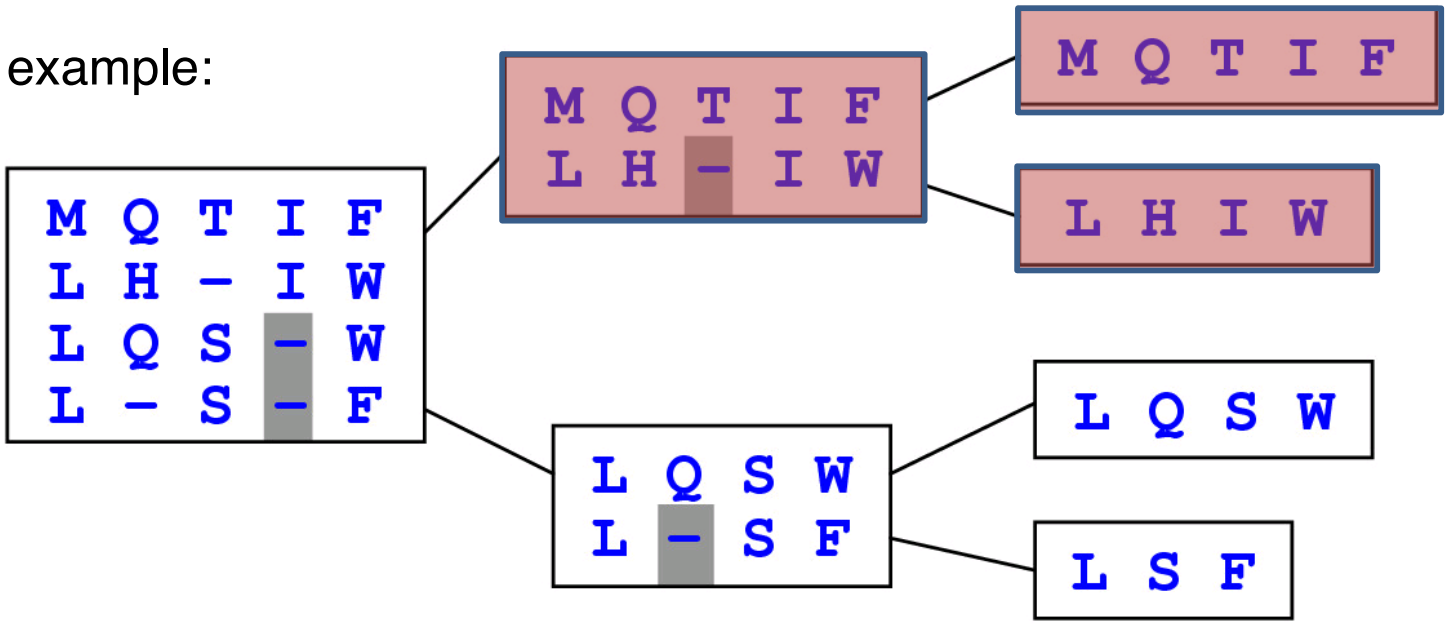
The alignment of sets of sequences and the construction of phyletic trees: an integrated method.

Hogeweg P, Hesper B.

Outline of the approach:

1. For a set of *N* sequences, perform all *N*(*N*-1)/2 pairwise alignments
2. From the pairwise distances, create a phylogenetic tree, for example through neighbor-joining
3. Start at the closest pair of leaves and align them
4. Work one's way "up the tree" by pairwise aligning the closet pair of sequences/alignments that are currently not yet aligned

Cartoon example:



Progressive multiple alignment

J Mol Evol. 1984;20(2):175-86.

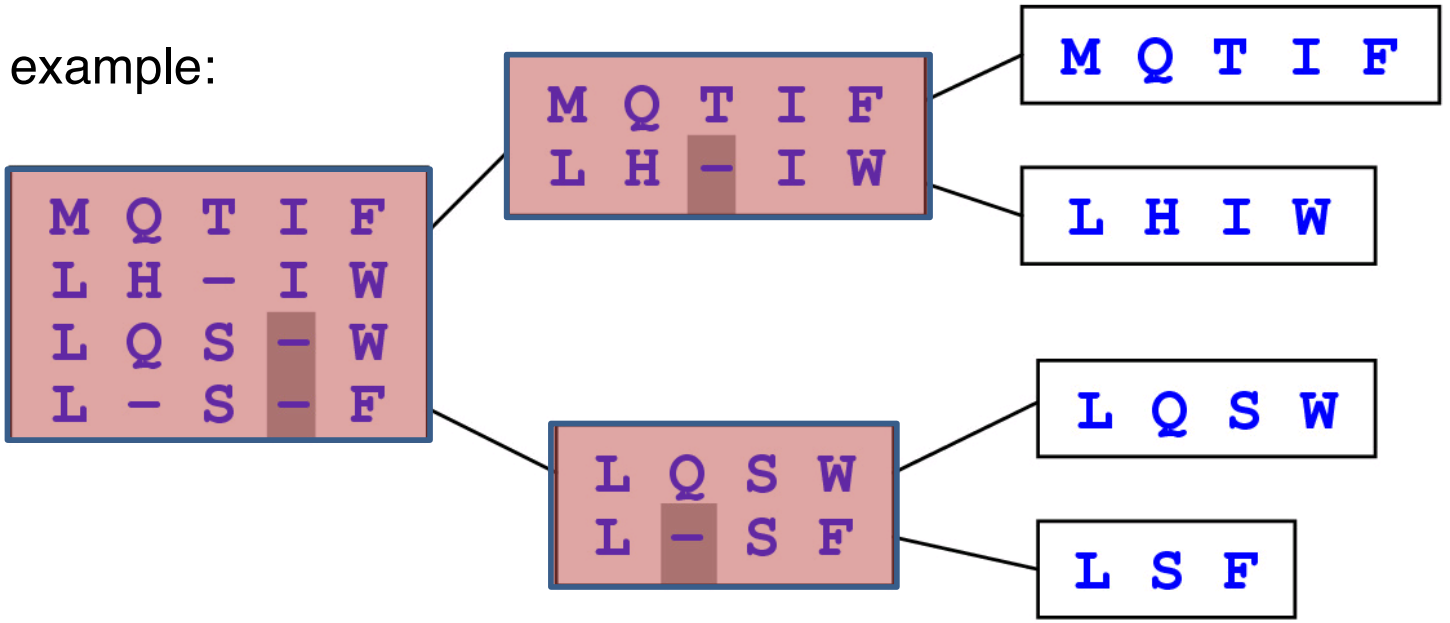
The alignment of sets of sequences and the construction of phyletic trees: an integrated method.

Hogeweg P, Hesper B.

Outline of the approach:

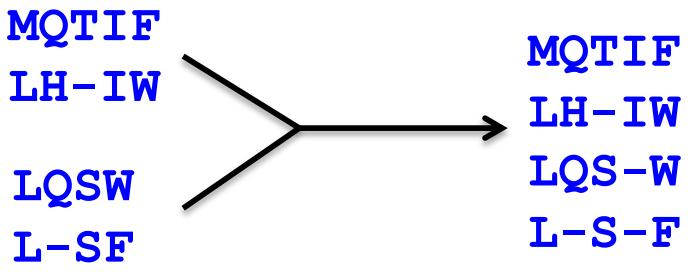
1. For a set of *N* sequences, perform all *N*(*N*-1)/2 pairwise alignments
2. From the pairwise distances, create a phylogenetic tree, for example through neighbor-joining
3. Start at the closest pair of leaves and align them
4. Work one's way "up the tree" by pairwise aligning the closet pair of sequences/alignments that are currently not yet aligned

Cartoon example:



Profile to Profile alignment

The one new ingredient needed is to determine how to optimally align two sub-alignments



Most widely used progressive alignment algorithms use a *sum of pairs* score, which scores a multiple alignment column by the sum of scores between all pairs of letters:

$$s_{\alpha\beta} = \text{ substitution score for substituting } \beta \text{ with } \alpha$$

$$s_{\alpha -} = s_{-\beta} = \text{gap score}$$

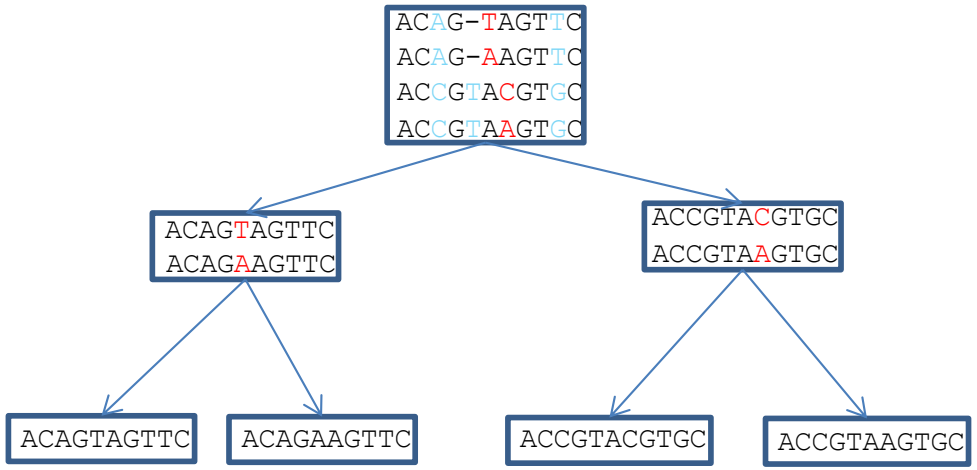
$$s_{--} = 0$$

$$S\begin{pmatrix} M & L \\ L & L \end{pmatrix} = s_{ML} + s_{ML} + s_{LL} + s_{LL}$$

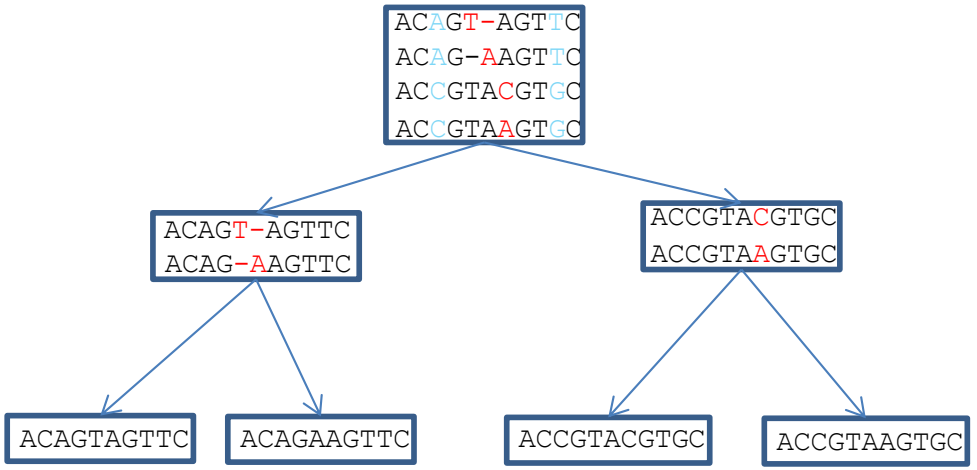
Progressive alignment

- One problem with progressive alignment is that it is *greedy*
- Once sequences are aligned they are 'glued' forever, even if later on, parts of their alignment clearly are wrong

Example: Progressive solution



Alternative solution



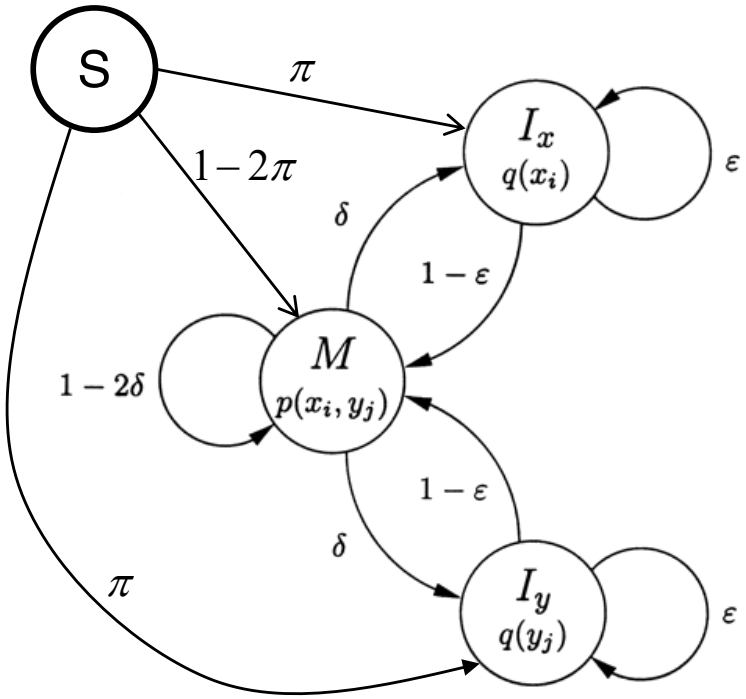
Iterative refinement procedure:

- Using the multiple alignment, re-estimate the pairwise distances and re-build the tree
- Take out one sequence and realign it to the profile formed by all other sequences
- Randomly divide the sequences into 2 subsets. Re-align the two sub-alignments made by the sequences in the two sets

Alternative: Prevent early mistakes rather than curing them - consistency-based alignment.

Consistency-based progressive alignment: ProbCons

Uses a pairHMM for sequence alignment



Consistency-based progressive alignment: ProbCons

Step 0: Train transition probabilities $(\pi, \delta, \varepsilon)$ by Expectation Maximization (Baum Welch) on the input set of sequence pairs.

Step 1: For every pair of sequences (x, y) and each pair of positions (i, j) in them, calculate the probability $P(i{:}\, j \mid x, y)$ that these two positions are aligned.

Step 2: For each pair (x, y), determine the alignment *a* with the highest *accuracy* with Viterbi-like recursion relations.

Step 3: Consistency-based updating of probabilities that two positions (i, j) in the pair of sequences (x, y) are aligned.

Step 4: Create a *phylogenetic tree* for all sequences.

Step 5: Carry out progressive alignment.

Step 6: Carry out iterative refinement.

Consistency-based progressive alignment: ProbCons

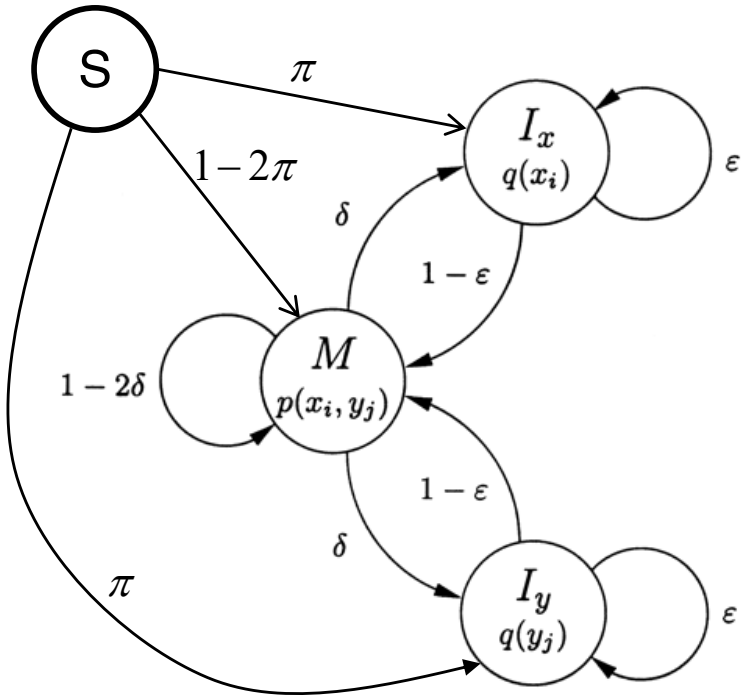
Expectation-Maximization update of pair-HMM parameters:

- Using the current values of $\{\pi, \delta, \varepsilon\}$ calculate the expected transitions: $\langle n_{II} \rangle, \langle n_{MI} \rangle$
- Use these to set: $\varepsilon^{new} = \frac{\langle n_{II} \rangle}{\langle n_{II} \rangle + \langle n_{MI} \rangle}$ $\quad \langle n_{IS} \rangle, \langle n_{MS} \rangle, \langle n_{MM} \rangle, \langle n_{IM} \rangle$
- We analogously calculate $\langle n_{IS} \rangle$, $\langle n_{MS} \rangle$, $\langle n_{MM} \rangle$, $\langle n_{IM} \rangle$ to update $\{\pi, \delta\}$
- We keep iterating these update equations until $\{\pi, \delta, \varepsilon\}$ no longer change
- One can repeat the procedure starting from different initial values of $\{\pi, \delta, \varepsilon\}$
- We keep the set of parameters that have overall maximal $P(D|\pi, \delta, \varepsilon)$

How do we get the EM equations? Reminder…

Consistency-based progressive alignment: ProbCons

Step 0: Train transition probabilities $\{\pi, \delta, \varepsilon\}$ by Expectation Maximization (Baum Welch) on the input set of pairs.



• Emission probabilities $p(x_i, y_j)$, $q(x_i)$ are based on the BLOSUM62 matrix.

• The probability of the data is a product over all pairs of sequences:

$$P(D|\pi, \delta, \varepsilon) = \prod_{x<y} P(x, y|\pi, \delta, \varepsilon)$$

• Formally, the probability of a pair is a sum over all possible alignments of the pair:

$$P(x, y|\pi, \delta, \varepsilon) = \sum_{a} P(x, y, a|\pi, \delta, \varepsilon)$$

Let us write out the probability of the sequences and the alignment in terms of the parameters of the pair-HMM

Consistency-based progressive alignment: ProbCons

Probability of a pair of sequences and their alignment: $P(x, y, a|\pi, \delta, \varepsilon)$

Let $n_{MM}(a)$ = Number of transitions from M to M in alignment a

$n_{IM}(a)$ = Number of transitions from M to insertion in alignment a

$n_{MI}(a)$ = Number of transitions from insertion to M in alignment a

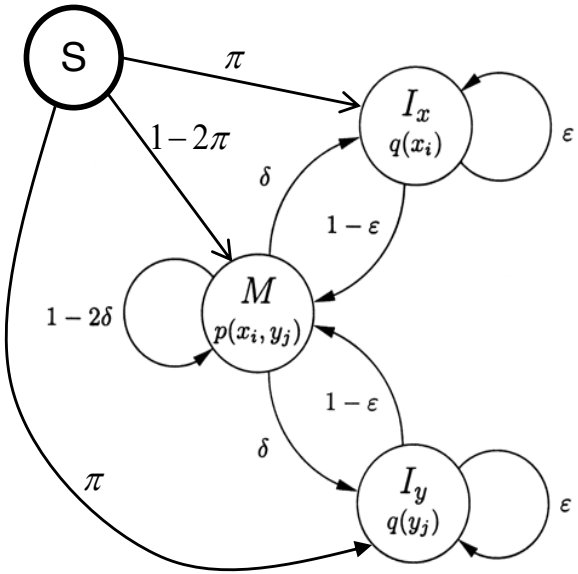
$n_{II}(a)$ = Number of transitions from insertion to itself in alignment a

$s_M(a)$ = 1 if the alignment starts in match state, 0 otherwise

M = all positions that are matched in alignment a

I_X = all positions with insertions in x in alignment a

I_Y = all positions with insertions in y in alignment a



Then we have

$$P(x, y, a|\pi, \delta, \varepsilon)$$
$$= [s_M(a)(1-2\pi) + 2\pi(1-s_M(a))]$$
$$\cdot \delta^{n_{IM}(a)}(1-2\delta)^{n_{MM}(a)}\varepsilon^{n_{II}(a)}(1-\varepsilon)^{n_{MI}(a)} \prod_{(i,j)\in M} p(x_i, y_j) \prod_{i\in I_x} q(x_i) \prod_{j\in I_y} q(y_j)$$

Consistency-based progressive alignment: ProbCons

$$P(x, y, a|\pi, \delta, \varepsilon)$$
$$= [s_M(a)(1 - 2\pi) + 2\pi(1 - s_M(a))]$$
$$\cdot \delta^{n_{IM}(a)}(1 - 2\delta)^{n_{MM}(a)}\varepsilon^{n_{II}(a)}(1 - \varepsilon)^{n_{MI}(a)} \prod_{(i,j)\in M} p(x_i, y_j) \prod_{i\in I_x} q(x_i) \prod_{j\in I_y} q(y_j)$$

To optimize the parameters we need to calculate the derivatives such as:

$$\frac{\partial log[P(D|\pi, \delta, \varepsilon)]}{\partial \varepsilon} = \sum_{x<y} \frac{\partial log[P(x, y|\pi, \delta, \varepsilon)]}{\partial \varepsilon} = \sum_{x<y} \frac{\frac{\partial}{\partial \varepsilon} P(x, y|\pi, \delta, \varepsilon)}{P(x, y|\pi, \delta, \varepsilon)} = \sum_{x<y} \frac{\frac{\partial}{\partial \varepsilon} [\sum_a P(x, y, a|\pi, \delta, \varepsilon)]}{P(x, y|\pi, \delta, \varepsilon)}$$

With respect to ε, $P(x, y, a \mid \pi, \delta, \varepsilon)$ can be written as:

$$P(x, y, a|\pi, \delta, \varepsilon) = C\varepsilon^{n_{II}(a)}(1 - \varepsilon)^{n_{MI}(a)}$$

Where C subsumes all the terms not dependent on ε. Then we have:

$$\frac{\partial}{\partial \varepsilon} P(x, y, a|\pi, \delta, \varepsilon) = C\left[n_{II}(a)\varepsilon^{n_{II}(a)-1}(1 - \varepsilon)^{n_{MI}(a)} - n_{MI}(a)\varepsilon^{n_{II}(a)}(1 - \varepsilon)^{n_{MI}(a)-1}\right]$$
$$= C\varepsilon^{n_{II}(a)}(1 - \varepsilon)^{n_{MI}(a)}\left[\frac{n_{II}(a)}{\varepsilon} - \frac{n_{MI}(a)}{1 - \varepsilon}\right] = P(x, y, a|\pi, \delta, \varepsilon)\left[\frac{n_{II}(a)}{\varepsilon} - \frac{n_{MI}(a)}{1 - \varepsilon}\right]$$

Consistency-based progressive alignment: ProbCons

Then

$$\frac{\partial log[P(D|\pi,\delta,\varepsilon)]}{\partial \varepsilon}$$

$$= \sum_{x<y} \frac{\frac{\partial}{\partial \varepsilon}[\sum_a P(x,y,a|\pi,\delta,\varepsilon)]}{P(x,y|\pi,\delta,\varepsilon)} = \sum_{x<y} \frac{\sum_a P(x,y,a|\pi,\delta,\varepsilon)\left[\frac{n_{II}(a)}{\varepsilon} - \frac{n_{MI}(a)}{1-\varepsilon}\right]}{P(x,y|\pi,\delta,\varepsilon)}$$

In other words

$$\frac{\partial log[P(D|\pi,\delta,\varepsilon)]}{\partial \varepsilon} = 0 \Rightarrow \frac{\sum_{x<y}\langle n_{II}(x,y)\rangle}{\varepsilon} = \frac{\sum_{x<y}\langle n_{MI}(x,y)\rangle}{1-\varepsilon} \Rightarrow \varepsilon = \frac{\langle n_{II}\rangle}{\langle n_{II}\rangle + \langle n_{MI}\rangle}$$

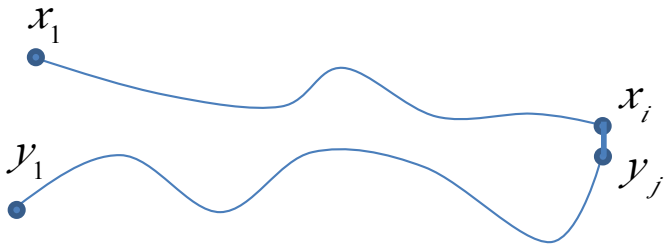
At the optimal parameter setting ε matches the expected fraction of the time one transitions from I to I (rather than to M).

Estimated numbers like $\langle n_{II}\rangle$ are obtained using the *Forward-Backward* algorithm.

Consistency-based progressive alignment: ProbCons

Calculate expected numbers of transitions

$F_s(x_i, y_j)$ = Forward sum of the probabilities of all alignments that end in state s after aligning the first i letters of x with the first j letters of y.

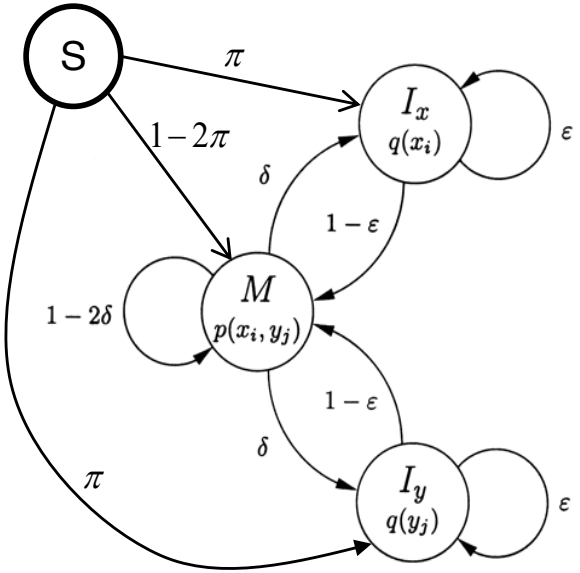


Recursion relations:

$$F_M(x_i, y_j) = p(x_i, y_j)\left[(1-2\delta)F_M\left(x_{i-1}, y_{j-1}\right) + (1-\varepsilon)F_{I_x}\left(x_{i-1}, y_{j-1}\right) + (1-\varepsilon)F_{I_y}\left(x_{i-1}, y_{j-1}\right)\right]$$

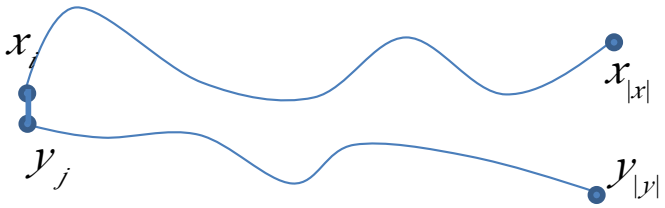
$$F_{I_x}\left(x_i, y_j\right) = q(x_i)\left[\delta F_M\left(x_{i-1}, y_j\right) + \varepsilon F_{I_x}\left(x_{i-1}, y_j\right)\right]$$

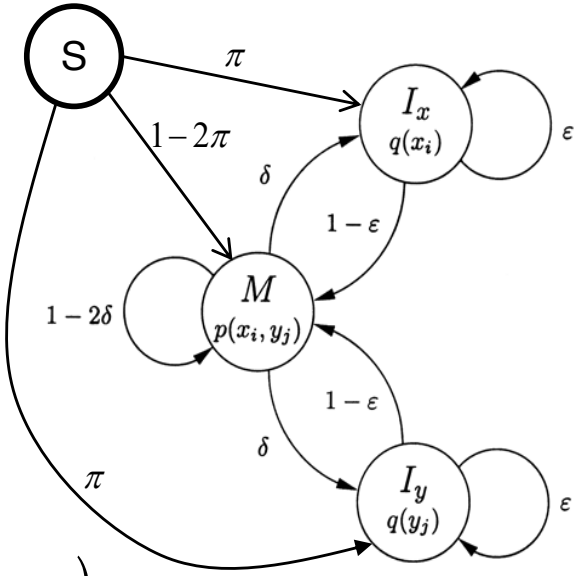
$$F_{I_y}\left(x_i, y_j\right) = q(y_j)\left[\delta F_M\left(x_i, y_{j-1}\right) + \varepsilon F_{I_x}\left(x_i, y_{j-1}\right)\right]$$



Consistency-based progressive alignment: ProbCons

$B_s(x_i, y_j)$ = Backward sum of the probabilities of all alignments starting in state s and aligning x from position i to the end with y from position j to the end.





Recursion relations:

$$B_M(x_i, y_j) = (1-2\delta)\, p(x_{i+1}, y_{j+1}) B_M(x_{i+1}, y_{j+1}) + \delta q(x_{i+1}) B_{I_x}(x_{i+1}, y_j) + \delta q(y_{j+1}) B_{I_y}(x_i, y_{j+1})$$

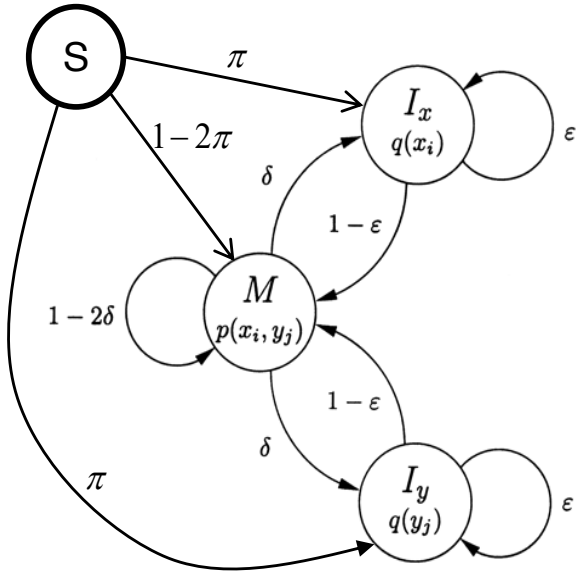
$$B_{I_x}(x_i, y_j) = \varepsilon q(x_{i+1}) B_{I_x}(x_{i+1}, y_j) + (1-\varepsilon)\, p(x_{i+1}, y_{j+1}) B_M(x_{i+1}, y_{j+1})$$

$$B_{I_y}(x_i, y_j) = \varepsilon q(y_{j+1}) B_{I_y}(x_i, y_{j+1}) + (1-\varepsilon)\, p(x_{i+1}, y_{j+1}) B_M(x_{i+1}, y_{j+1})$$

Total probability for a pair of sequences:

$$P(x, y \mid \pi, \delta, \varepsilon) = F_M(x_{|x|}, y_{|y|}) + F_{I_x}(x_{|x|}, y_{|y|}) + F_{I_y}(x_{|x|}, y_{|y|})$$

Consistency-based progressive alignment: ProbCons



Expected number of transitions:

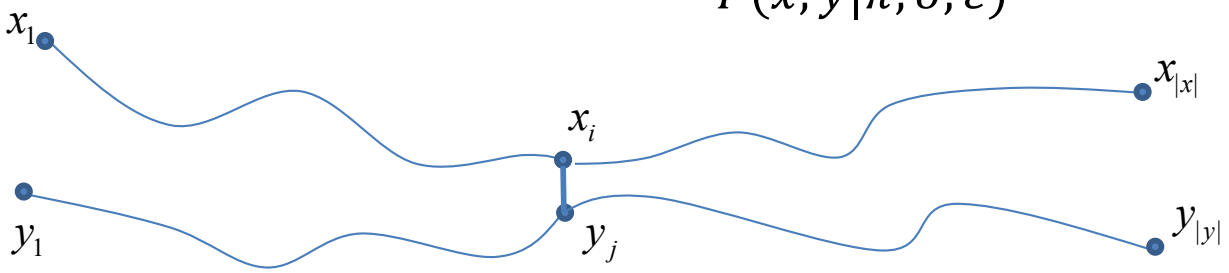
$$\langle n_{II} \rangle = \sum_{i,j} \frac{F_{I_x}(x_i, y_j)\varepsilon q(x_{i+1}) B_{I_x}(x_{i+1}, y_j) + F_{I_y}(x_i, y_j)\varepsilon q(y_{j+1}) B_{I_y}(x_i, y_{j+1})}{P(x, y|\pi, \delta, \varepsilon)}$$

$$\langle n_{MI} \rangle = \sum_{i,j} \frac{F_{I_x}(x_i, y_j)(1-\varepsilon) p(x_{i+1}, y_{j+1}) B_M(x_{i+1}, y_{j+1}) + F_{I_y}(x_i, y_j)(1-\varepsilon) p(x_{i+1}, y_{j+1}) B_M(x_{i+1}, y_{j+1})}{P(x, y|\pi, \delta, \varepsilon)}$$

Consistency-based progressive alignment: ProbCons

Step 1: For every pair of sequences (x, y) and each pair of positions (i, j) in them, calculate the probability $P(i{:}\, j|x, y)$ that these two positions are aligned:

$$P(i{:}\, j|x, y) = \frac{F_M(x_i, y_j)B_M(x_i, y_j)}{P(x, y|\pi, \delta, \varepsilon)}$$



Step 2: For each pair (x, y), determine the alignment a with the highest *accuracy*

$$\langle A(a|x, y)\rangle = \sum_{(i,j)\in a} P(i{:}\, j|x, y)$$

with Viterbi-like recursion relations:

If $S(x_i, y_j)$ = expected accuracy for the optimal alignment up to positions x_i and y_j

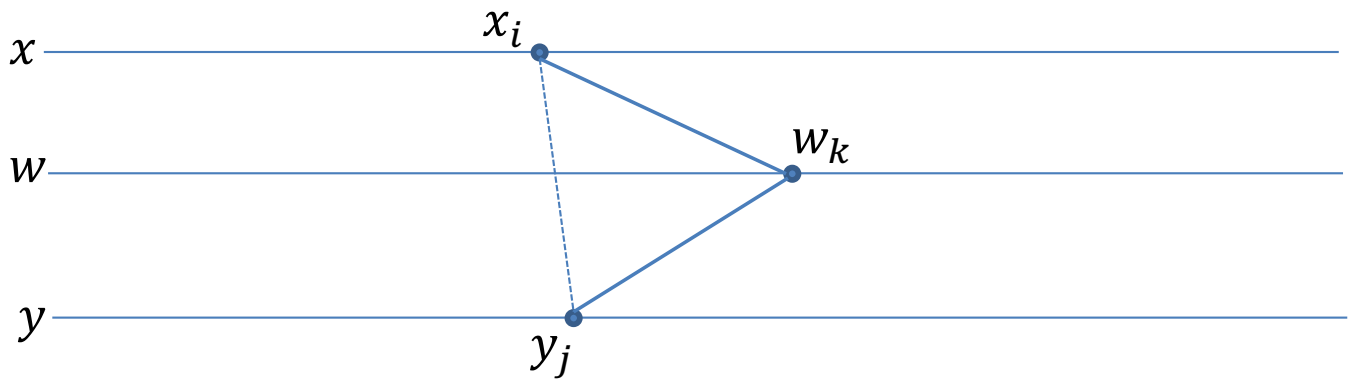
$$S(x_i, y_j) = max\left[S(x_{i-1}, y_{j-1}) + P(i{:}\, j|x, y), S(x_{i-1}, y_j), S(x_i, y_{j-1})\right]$$

Final accuracy: $\langle S(x, y)\rangle = \frac{S(x_{|x|}, y_{|y|})}{min[|x|,|y|]}$ is a measure of the closeness of (x, y).

Consistency-based progressive alignment: ProbCons

Step 3: Consistency updating

Update the probabilities $P(i:j|x,y)$ by looking at all *triplets* of sequences:



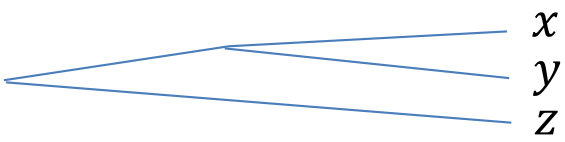
$$\tilde{P}(i:j|x,y) = \frac{1}{|S|}\sum_{w\in S}\sum_{k} P(i:k|x,w)P(k:j|w,y)$$

In this way, pairs whose alignment is consistent across triplets get increased probability, and inconsistent pairs get their probability lowered. The triplet-update can be iterated multiple times.

Consistency-based progressive alignment: ProbCons

Step 4: Creating a *phylogenetic tree* for the sequences in *S*

- For this task ProbCons use simple *hierarchical clustering*:
  One iteratively clusters the pair with the highest expected alignment accuracy.
- The expected accuracy for the merged cluster is calculated as follows:

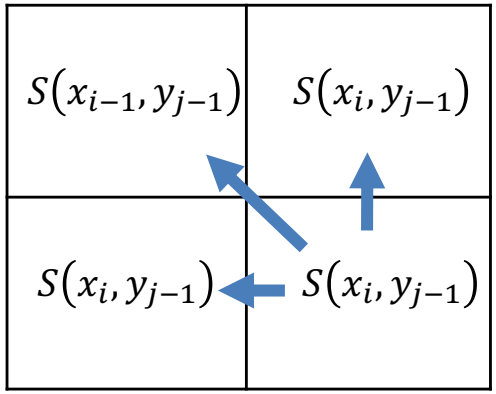


$$\langle S((x,y),z)\rangle = \langle S(x,y)\rangle \frac{(\langle S(x,z)\rangle + \langle S(y,z)\rangle)}{2}$$

Step 5: Progressive alignment

- Starting from the leaves of the phylogenetic tree, align pairs of sequences by at each step maximizing the expected accuracy of the resulting alignment.
- For two individual sequences (x, y):

$$S(x_i, y_j) = max \begin{bmatrix} S(x_{i-1}, y_{j-1}) + P(i:j|x,y), \\ S(x_{i-1}, y_j), S(x_i, y_{j-1}) \end{bmatrix}$$



- Score of the alignment is just the sum of $\tilde{P}(i:j|x,y)$ for aligned pairs.

Consistency-based progressive alignment: ProbCons

Scoring for profile-to-profile alignment:

• The score for aligning two profiles (columns of already aligned sequences) is simply the *sum of pairwise* scores:

$$S\begin{pmatrix} x_i & w_m \\ y_j, & u_n \\ z_k & v_r \end{pmatrix} = \tilde{P}(i{:}\,m|x,w) + \tilde{P}(i{:}\,n|x,u) + \tilde{P}(i{:}\,r|x,v) + \ldots + \tilde{P}(k{:}\,r|z,v)$$

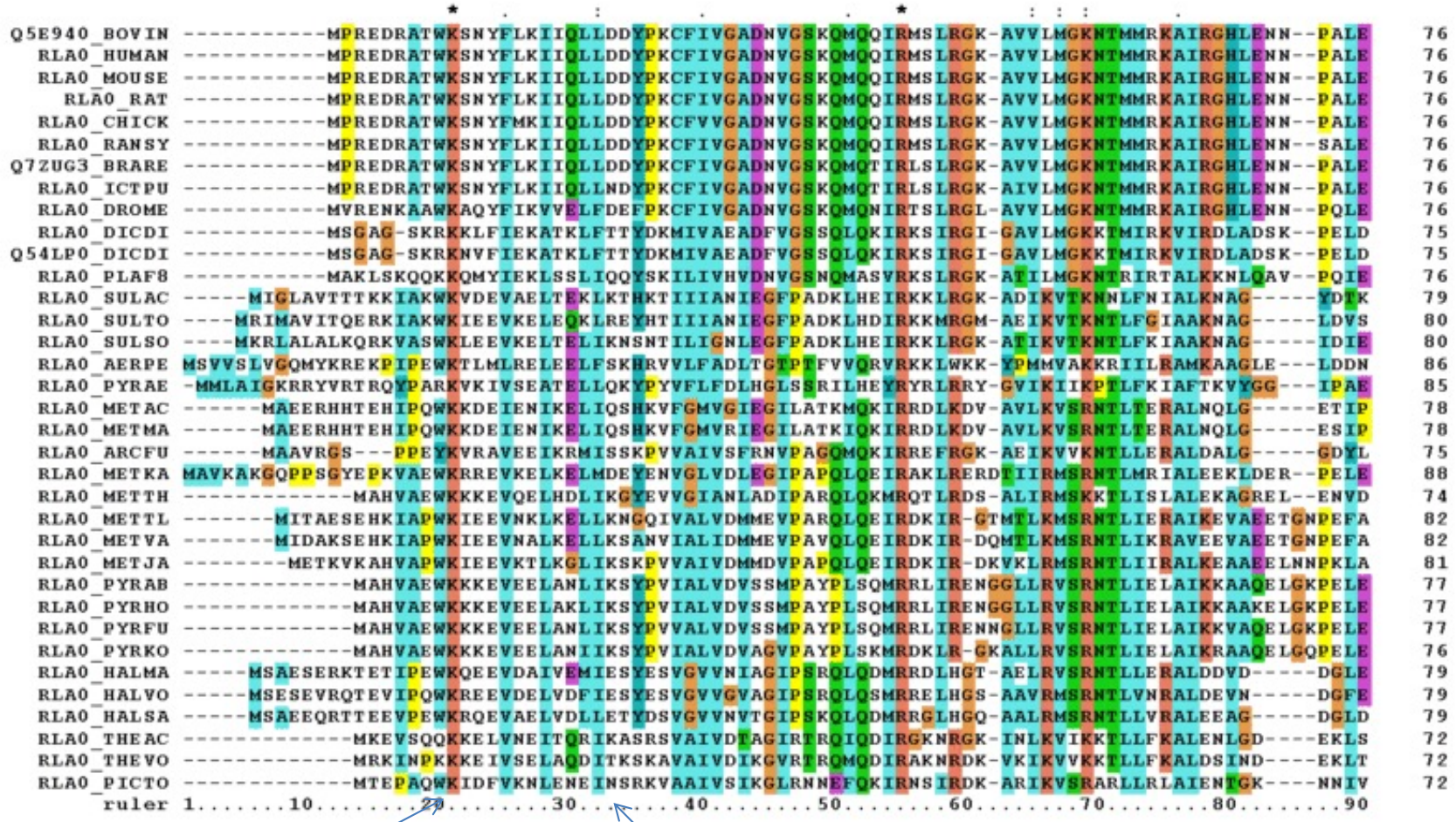
Final step: Iterative refinement

• Randomly divide the set of sequences *S* into two groups.

• Realign the sub-alignments of these two groups with each other.

Alternative approach to multiple alignment

Aim to learn the position-specific constraints of the genes in the family
(they reflect different *functional constraints)*



Very conserved column

Less conserved column

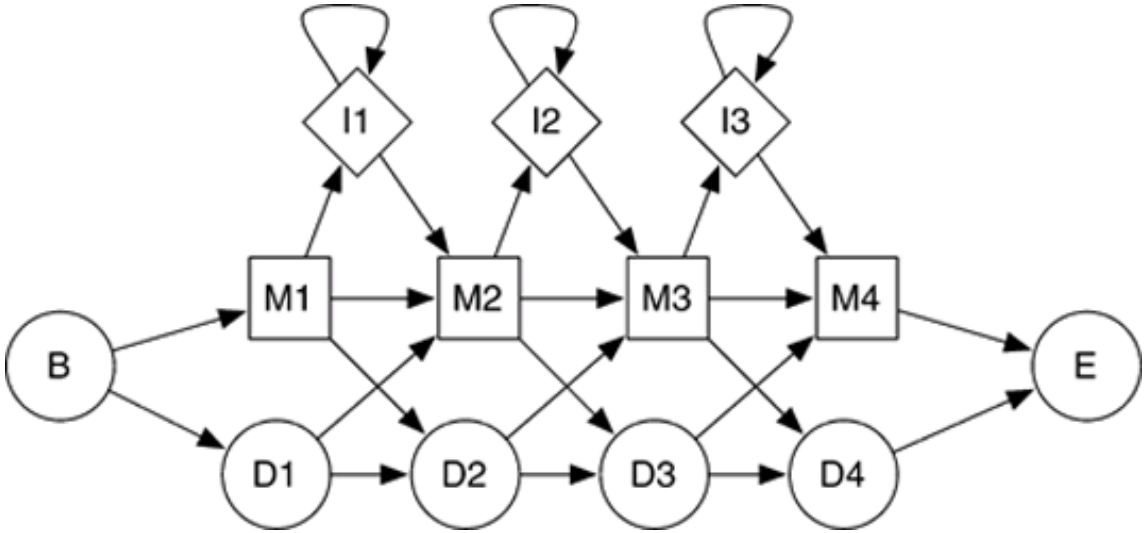
Orthologous acidic ribosomal proteins from multiple species

Profile HMMs for multiple alignment

• Instead of aligning the sequences to *each other* they get all aligned to a *model* that reflects the selective constraints that are acting on the sequences in the family

• Rather than reconstructing the evolutionary history of a set of closely-related sequences, we focus on the limit where all sequences in the model are separated by long evolutionary times and they are considered *independent* samples from an underlying distribution

• Profile-HMMs are models with the following general form:

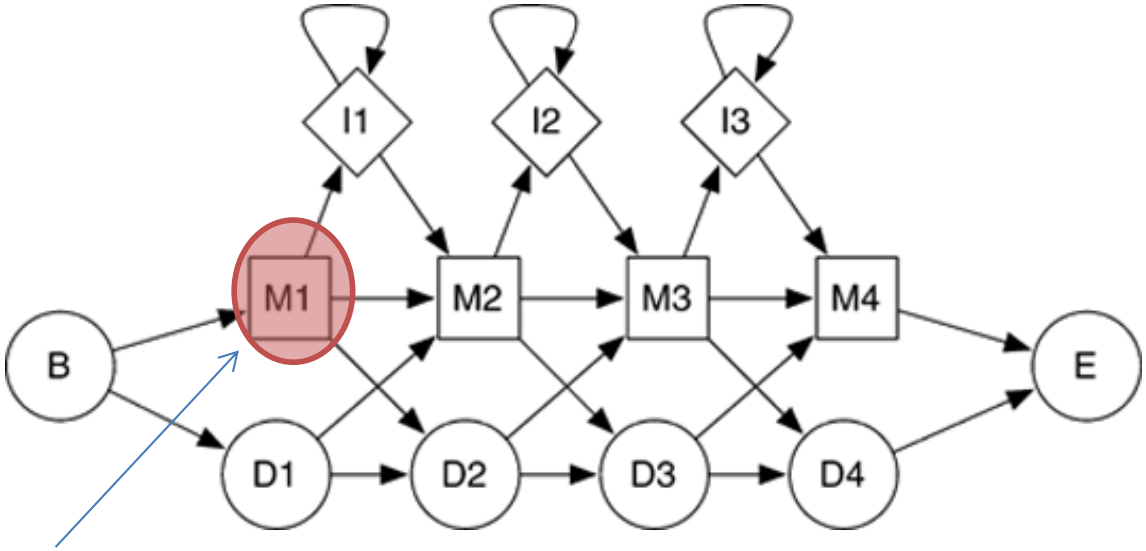


Profile HMMs for multiple alignment

• Instead of aligning the sequences to *each other* they get all aligned to a *model* that models the selective constraints that are acting on the sequences in the family

• Rather than reconstructing the evolutionary history of a set of closely-related sequences, we focus on the limit where all sequences in the model are separated by long evolutionary times and they are considered *independent* samples from an underlying distribution

• Profile-HMMs are models with the following general form:



Match states corresponding to different functional positions in the sequences.

Each match state has its own emission probabilities: $E(s|M_i)$

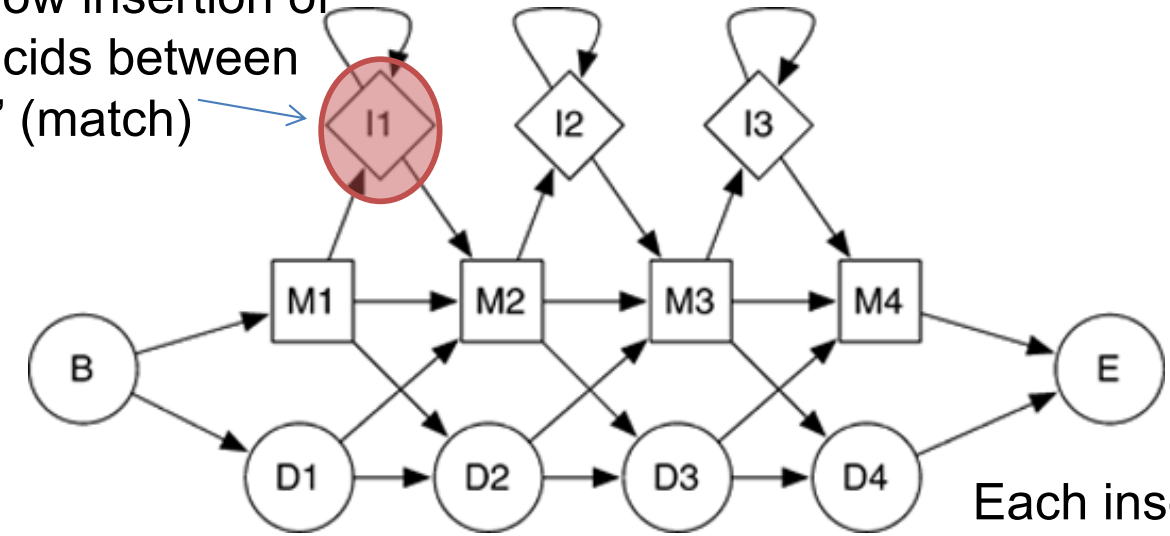
Profile HMMs for multiple alignment

- Instead of aligning the sequences to *each other* they get all aligned to a *model* that models the selective constraints that are acting on the sequences in the family

- Rather than reconstructing the evolutionary history of a set of closely-related sequences, we focus on the limit where all sequences in the model are separated by long evolutionary times and they are considered *independent* samples from an underlying distribution

- Profile-HMMs are models with the following general form:

Insert states allow insertion of "extra" amino acids between two "functional" (match) positions



Transitions to each insert state may have different probabilities: $T(I_i|M_i), T(I_i|I_i), T(M_{i+1}|I_i)$

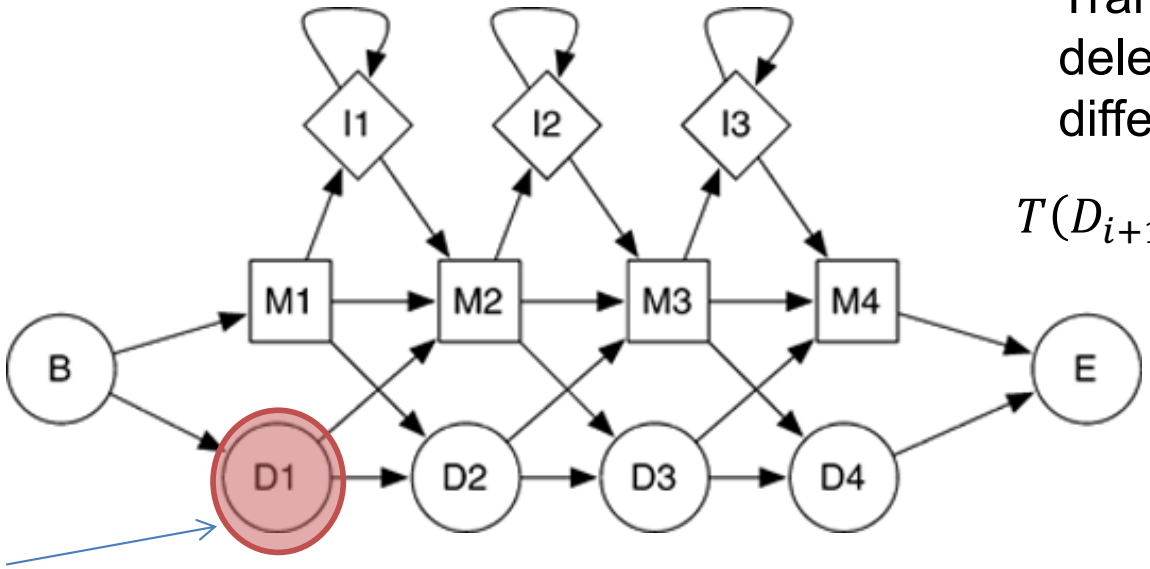
Each insert state can have its own emission probabilities: $E(s|I_i)$

Profile HMMs for multiple alignment

• Instead of aligning the sequences to *each other* they get all aligned to a *model* that models the selective constraints that are acting on the sequences in the family

• Rather than reconstructing the evolutionary history of a set of closely-related sequences, we focus on the limit where all sequences in the model are separated by long evolutionary times and they are considered *independent* samples from an underlying distribution

• Profile-HMMs are models with the following general form:



Transitions to and from each delete state may have different probabilities:

$$T(D_{i+1}|M_i), T(M_{i+1}|D_i), T(D_{i+1}|D_i)$$

Deletion states allow a family member to skip certain functional positions

There are no emissions in delete states

Profile HMMs for multiple alignment

Outline of the procedure:

• Make an initial *reasonable* guess for the HMM's transition and emission probabilities
• Often one could start by using another algorithm to make a multiple alignment of the family
• Given an alignment, create a match state for each column with less than 50% gaps in it. Number them from left to right
• For each match state M_i count the number of occurrences $n_i(s)$ of each amino acid s and set the emission probabilities:

$$E(s|M_i) = \frac{n_i(s) + 20\pi_s}{n_i + 20}$$

n_i = Number of amino acids at position i

π_s = Background frequency of amino acid s

• To set transition probabilities $T(S'|S)$ simply determine the fraction of sequences that move from state S to S' in the multiple alignment

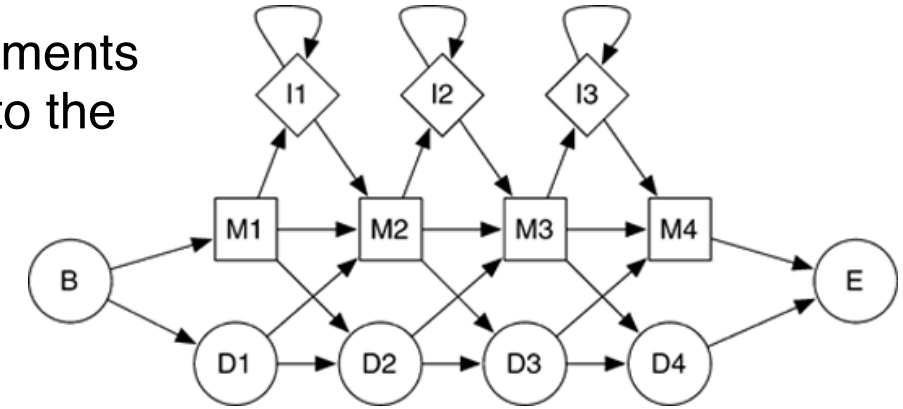
• This procedure sets the initial profile-HMM model

• We refine this model through *Baum-Welch* training, iterating:
    • For each sequence, run the forward-backward algorithm
    • Calculate the expected number of transitions and emissions for each sequence
    • From these update the transition and emission probabilities

Profile HMMs for multiple alignment

Forward-backward algorithm for the profile-HMM:

$F_S(n) =$ Sum of the probabilities of all alignments of the first n letters of sequence x to the HMM ending in state S



Recursion relations:

$$F_{M_i}(n) = E(x_n|M_i)\left[T(M_i|M_{i-1})F_{M_{i-1}}(n-1) + T(M_i|I_{i-1})F_{I_{i-1}}(n-1) + T(M_i|D_{i-1})F_{D_{i-1}}(n-1)\right]$$

$$F_{I_i}(n) = E(x_n|I_i)\left[T(I_i|M_i)F_{M_i}(n-1) + T(I_i|I_i)F_{I_i}(n-1)\right]$$

$$F_{D_i}(n) = T(D_i|M_{i-1})F_{M_{i-1}}(n) + T(D_i|D_{i-1})F_{D_i}(n)$$

$B_S(n) =$ Sum of the probabilities of all alignments of letters n through the end of sequence x, starting in state S

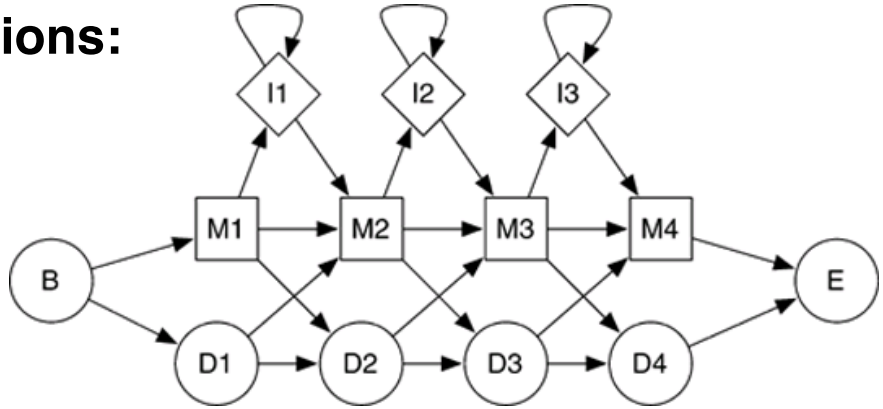
The recursion relations for the backward probabilities are entirely analogous.

Profile HMMs for multiple alignment

Expected numbers of transitions and emissions:

$\langle nr(s|M_i)\rangle =$ Expected number of times amino acid s is emitted in state M_i

$\langle nr(S'|S)\rangle =$ Expected number of transitions from state S to state S'.



Only 1 when amino acid s equals x_n

$$\langle nr(s|M_i)\rangle = \sum_x \sum_n \delta_{sx_n} \frac{F_{M_i}(n)B_{M_i}(n)}{F_E(|x|)}$$

Probability of the whole sequence

Other examples:

$$\langle nr(M_i|M_{i-1})\rangle = \sum_x \sum_n \frac{F_{M_{i-1}}(n-1)E(x_n|M_i)T(M_i|M_{i-1})B_{M_i}(n)}{F_E(|x|)}$$

$$\langle nr(M_i|D_{i-1})\rangle = \sum_x \sum_n \frac{F_{D_{i-1}}(n-1)E(x_n|M_i)T(M_i|D_{i-1})B_{M_i}(n)}{F_E(|x|)}$$

And similar equations for all the other transitions and emissions

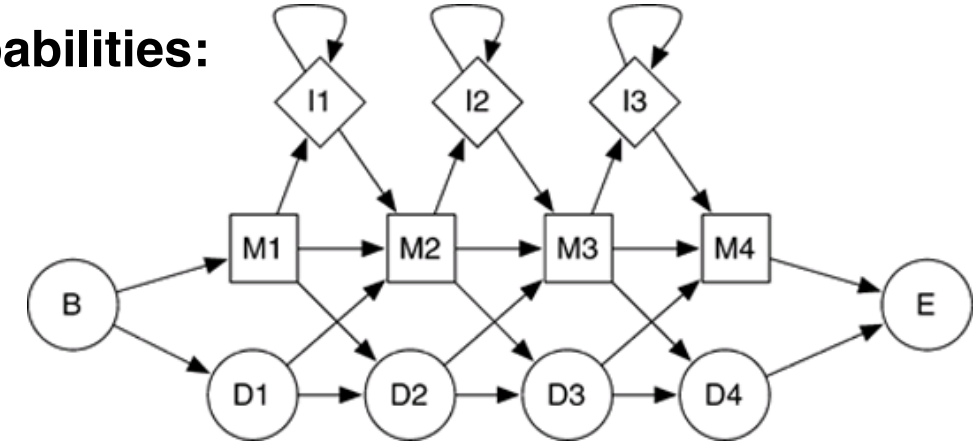
Profile HMMs for multiple alignment

Update of emission and transitions probabilities:

$$E^{new}(s|M_i) = \frac{\langle nr(s|M_i)\rangle}{\sum_{s'}\langle nr(s'|M_i)\rangle}$$

$$E^{new}(s|I_i) = \frac{\langle nr(s|I_i)\rangle}{\sum_{s'}\langle nr(s'|I_i)\rangle}$$

$$T^{new}(S'|S) = \frac{\langle nr(S'|S)\rangle}{\sum_{S''}\langle nr(S''|S)\rangle}$$



- We iterate the Baum-Welch update until the probabilities no longer change
- Most implementations use additional schemes to avoid getting stuck on local optima

Final Alignment:

- Once we have determined all emission and transition probabilities for the HMM, we determine the *optimal* alignment to the HMM for each of the sequences x using the Viterbi algorithm

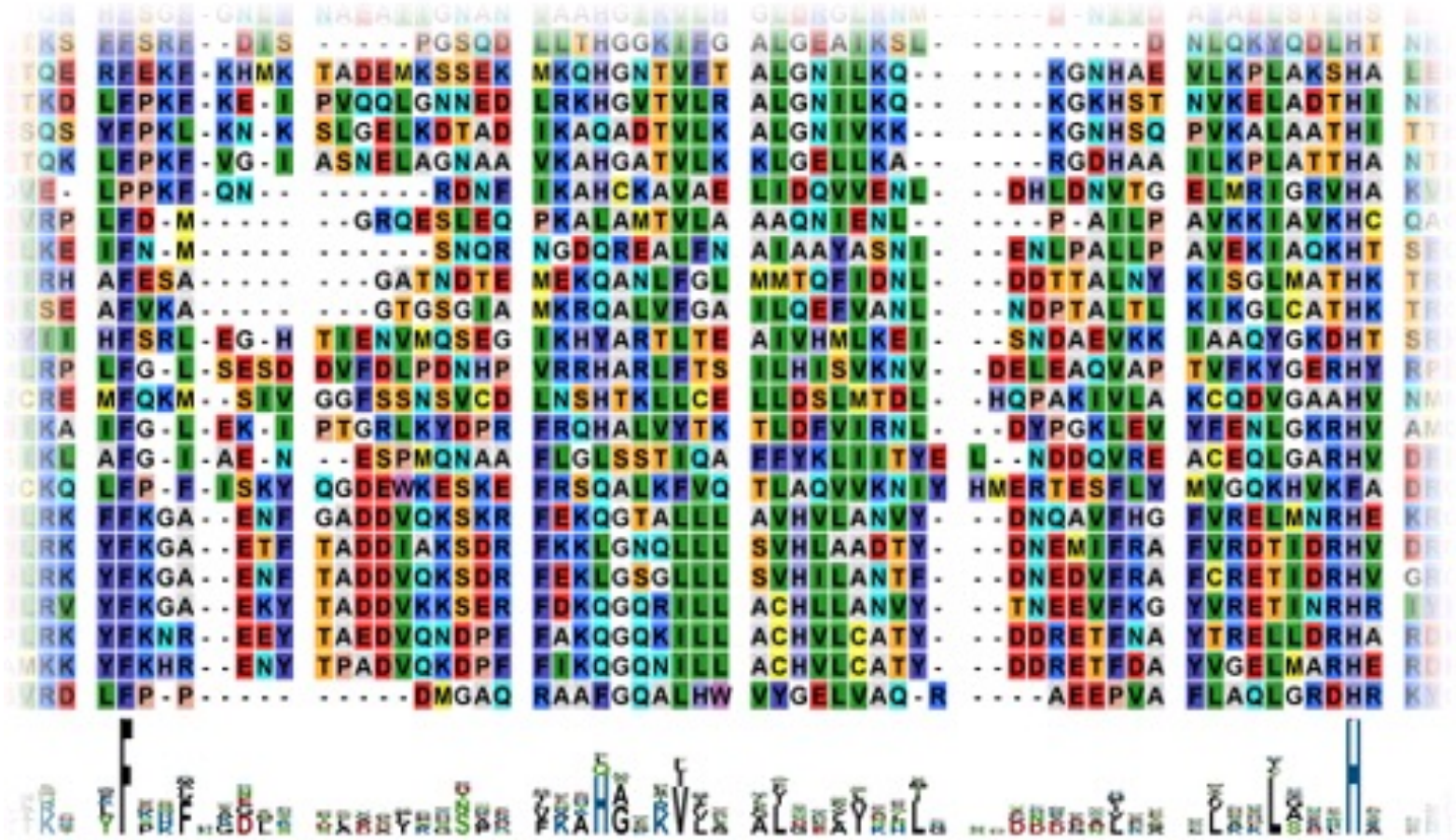
$$P_{M_i}(n) = E(x_n|M_i)\left[T(M_i|M_{i-1})P_{M_{i-1}}(n-1) + T(M_i|I_{i-1})P_{I_{i-1}}(n-1) + T(M_i|D_{i-1})P_{D_{i-1}}(n-1)\right]$$

$$P_{I_i}(n) = E(x_n|I_i)\left[T(I_i|M_i)P_{M_i}(n-1) + T(I_i|I_i)P_{I_i}(n-1)\right]$$

$$P_{D_i}(n) = T(D_i|M_{i-1})P_{M_{i-1}}(n) + T(D_i|D_{i-1})P_{D_i}(n)$$

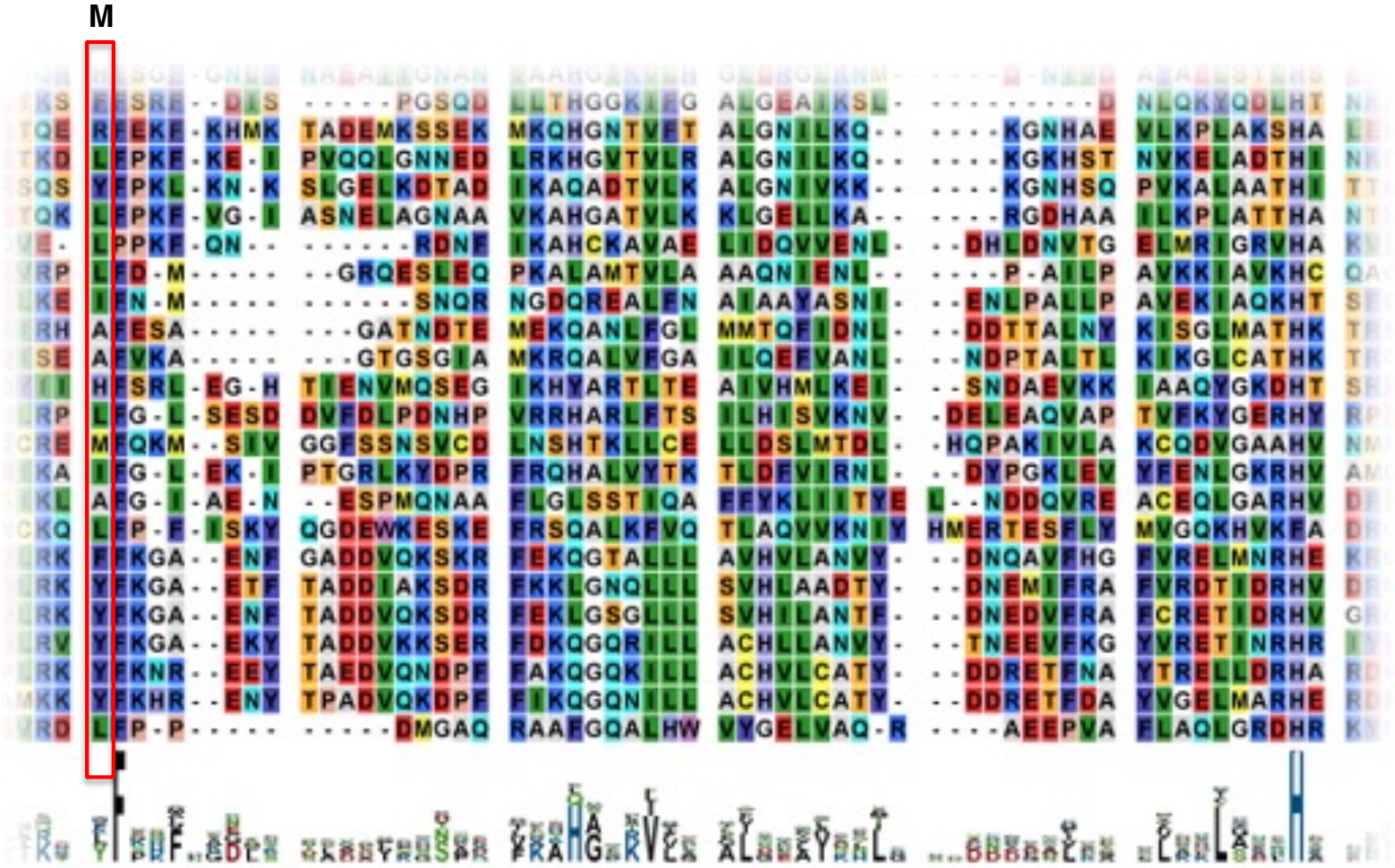
Profile HMMs for multiple alignment

Part of the profile-HMM alignment of the globin family of proteins.



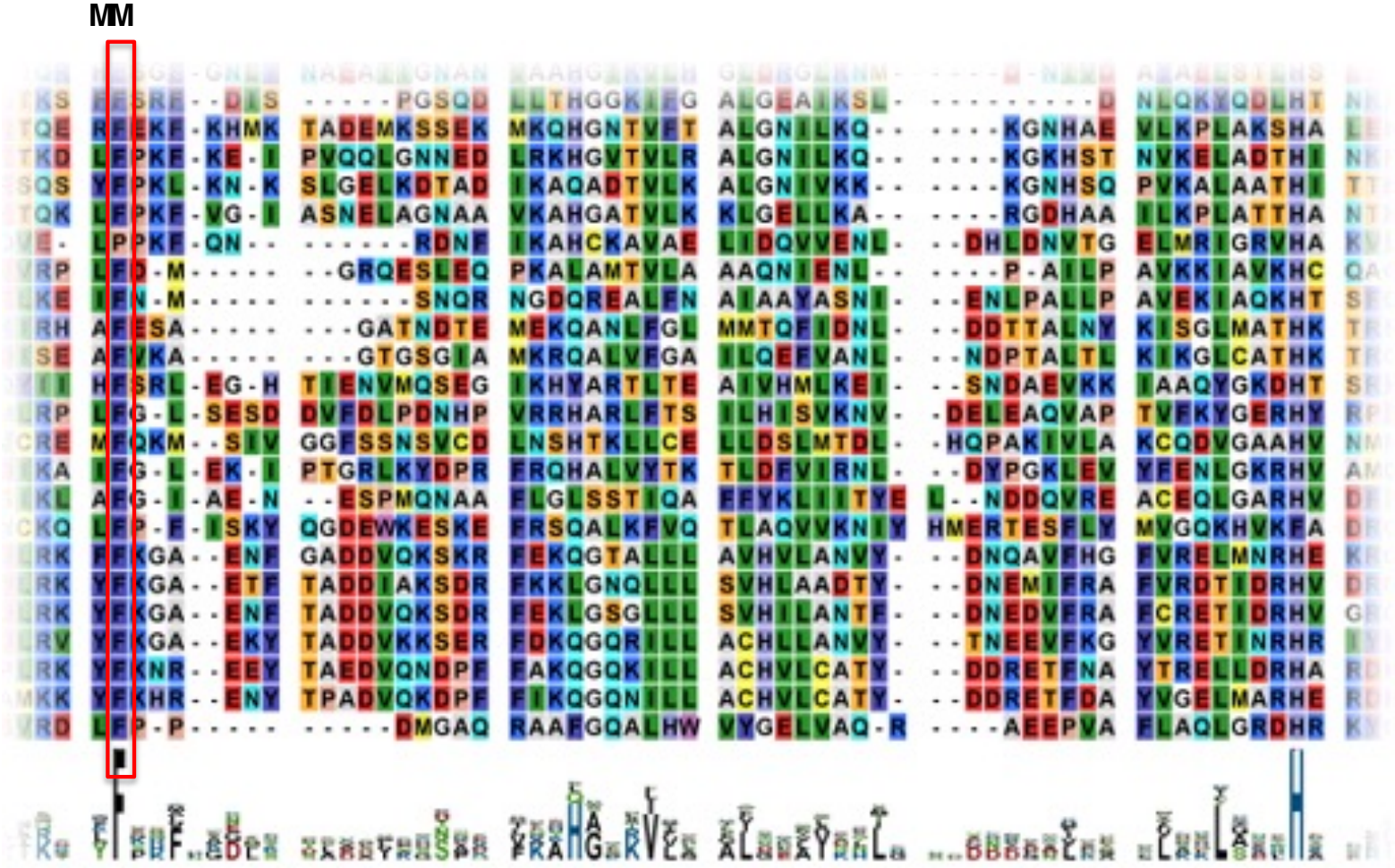
Position-dependent emission probabilities

Setting up the initial parameter values



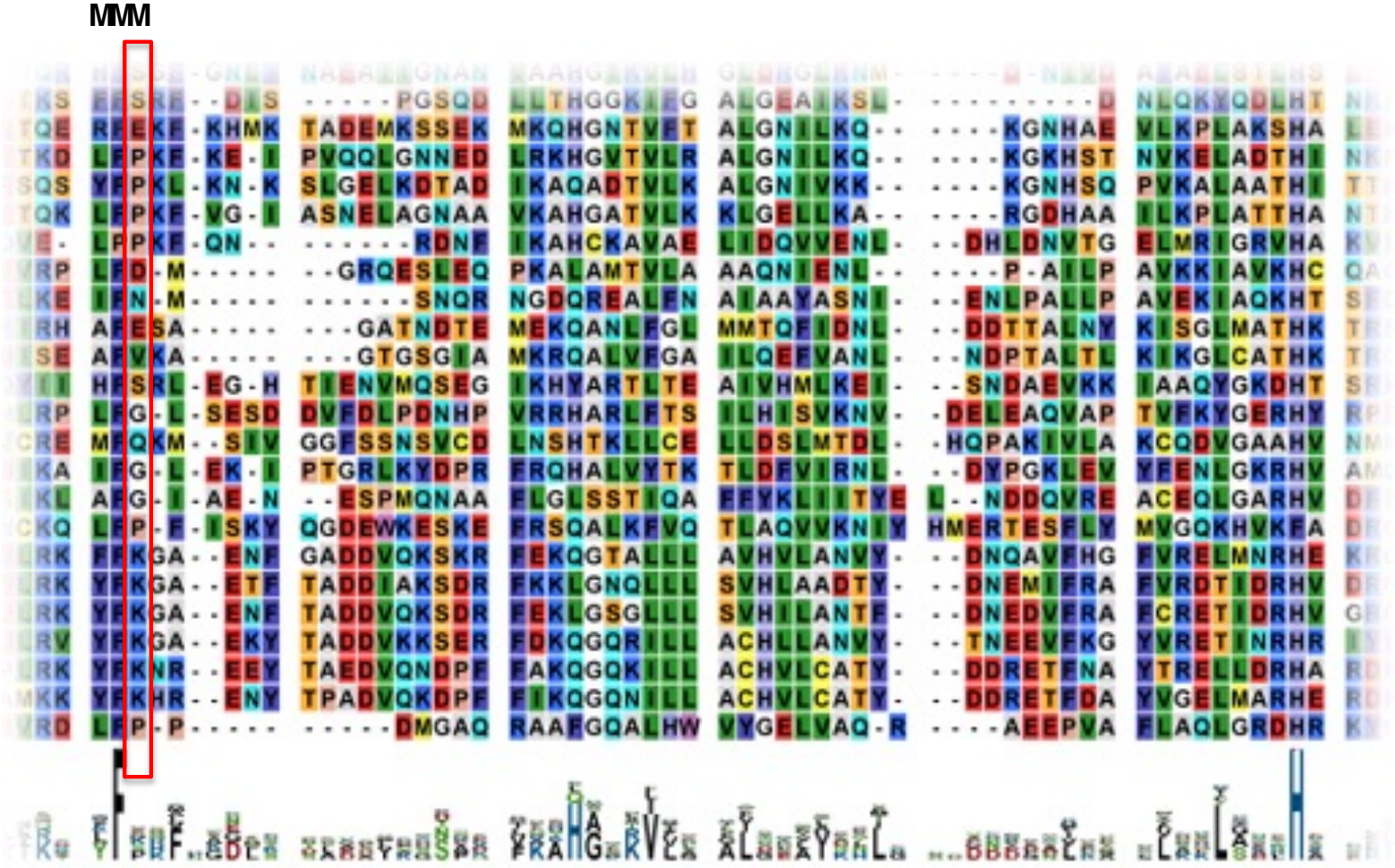
Part of the profile-HMM alignment of the globin family of proteins

Setting up the initial parameter values



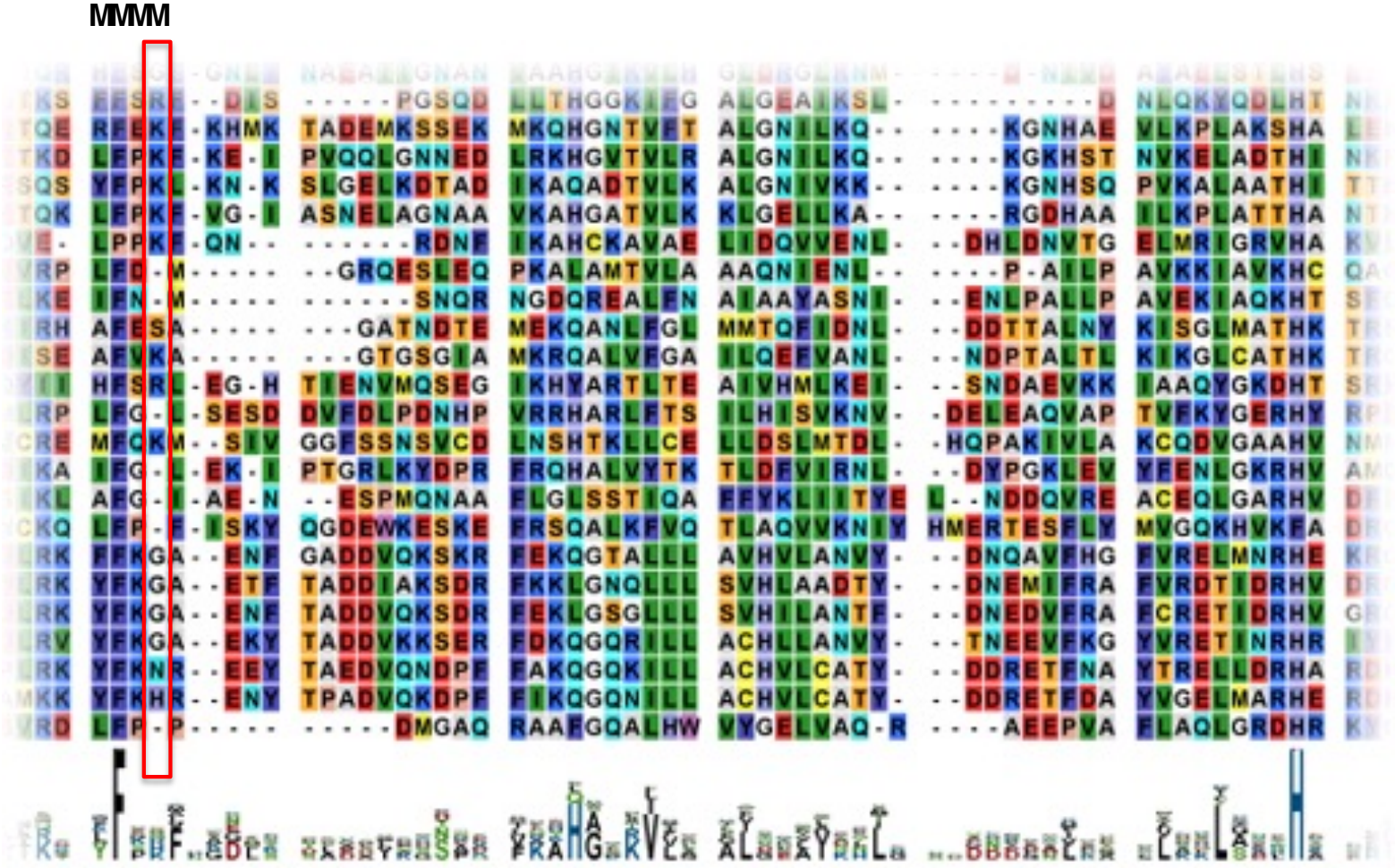
Part of the profile-HMM alignment of the globin family of proteins

Setting up the initial parameter values



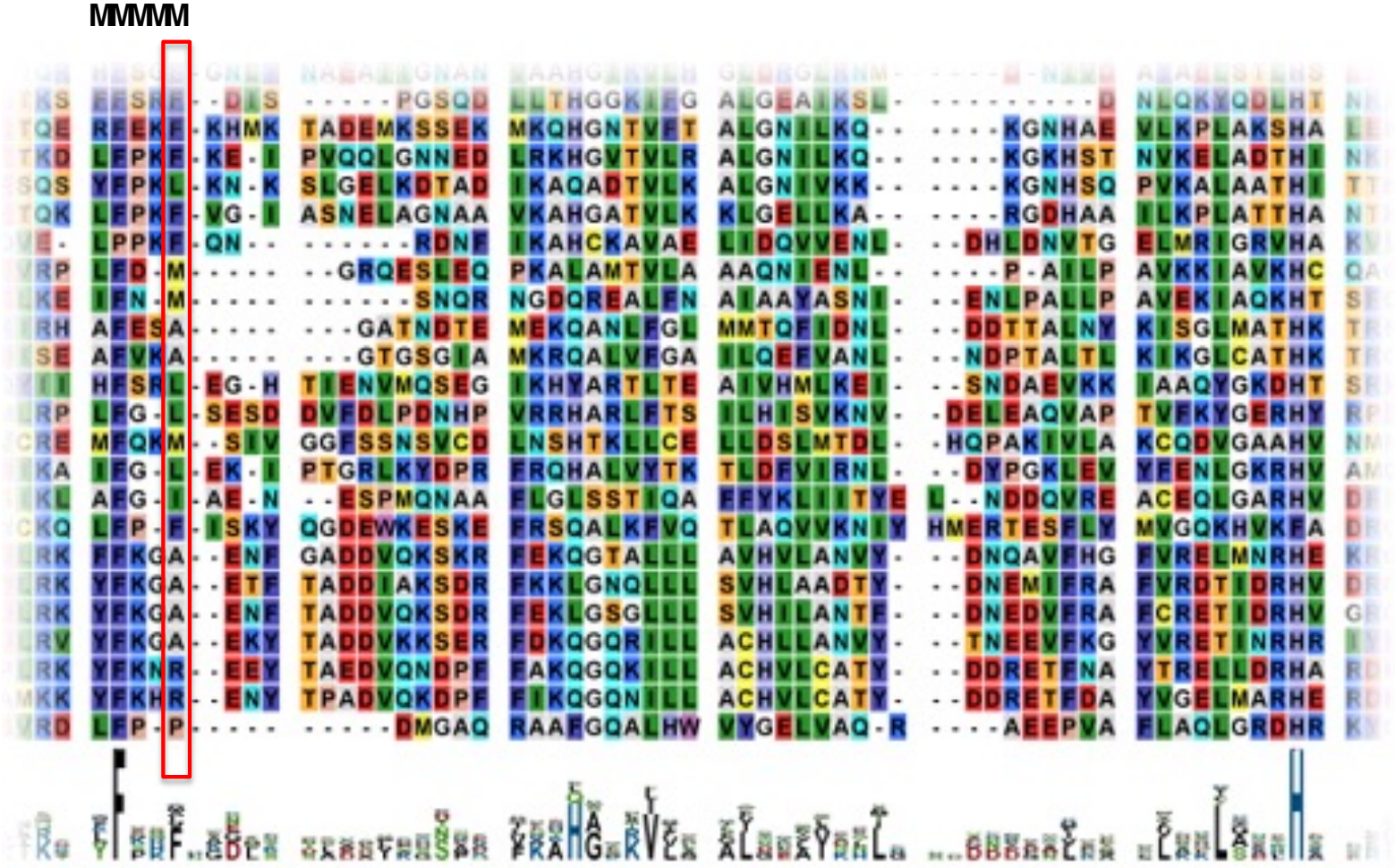
Part of the profile-HMM alignment of the globin family of proteins

Setting up the initial parameter values



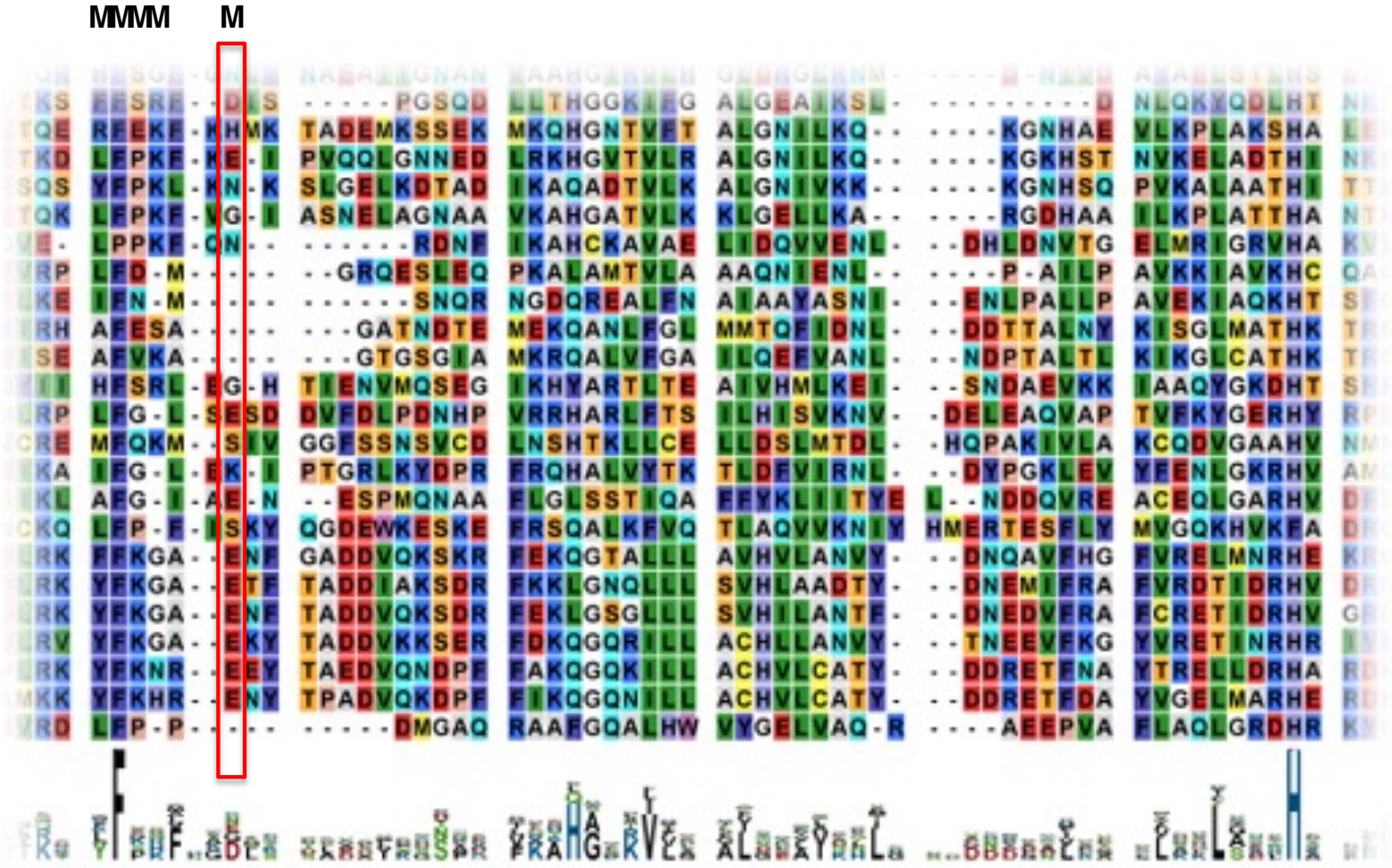
Part of the profile-HMM alignment of the globin family of proteins

Setting up the initial parameter values



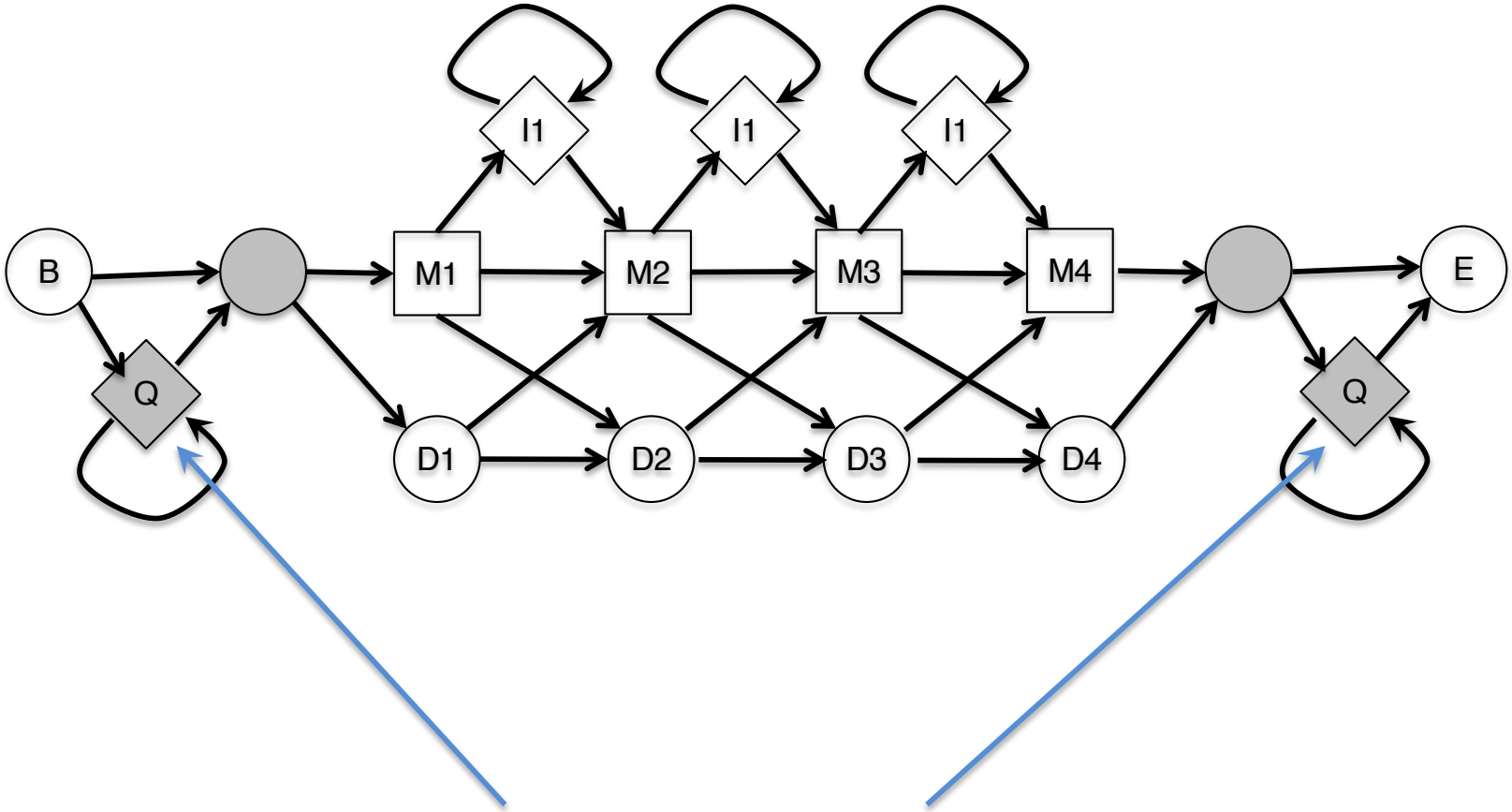
Part of the profile-HMM alignment of the globin family of proteins

Setting up the initial parameter values



Part of the profile-HMM alignment of the globin family of proteins

Profile HMM for finding local matches



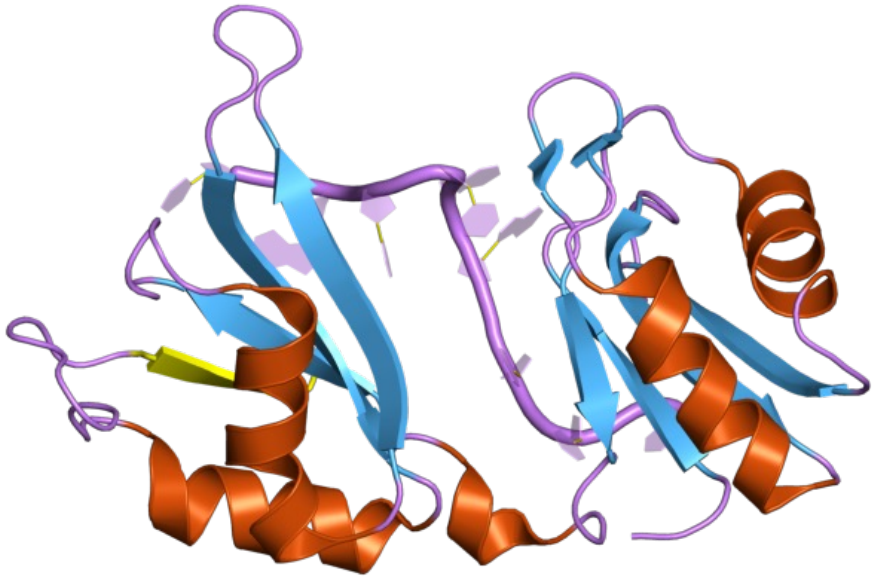
Introduce extra states to model regions flanking the model

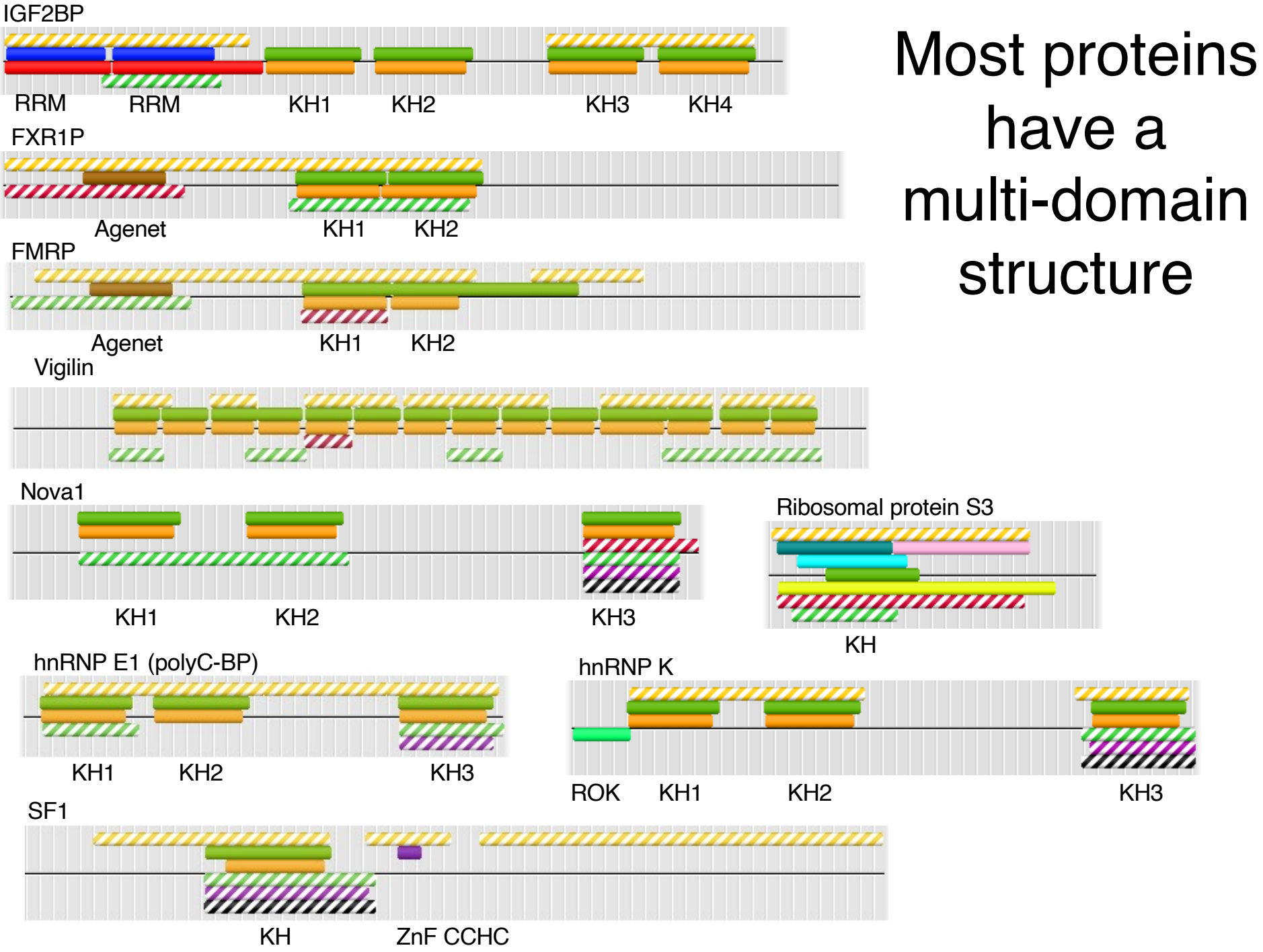
Protein domains

Protein domain: part of protein that can fold, function and evolve independently of the rest of the protein.

One domain typically associated with a specific function

- if we can recognize a domain in a protein sequence we can predict the function of the protein.





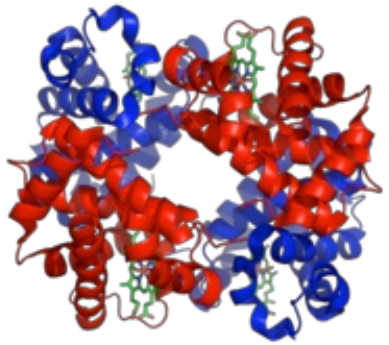
Most proteins have a multi-domain structure

Methods to identify protein domains

Structural classification of proteins (SCOP): database of protein structural domains based on similarity of their amino acid sequences and their three-dimensional structures. It is largely built through visual inspection of protein structures.

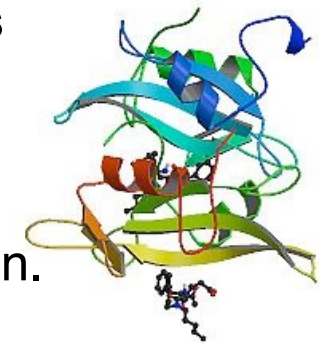
Methods to identify protein domains

Structural classes in SCOP
α-helical domains
β-sheet domains
α/β domains consisting of ”β-α-β" structural units (motifs) that form mainly parallel β-sheets
α+β domains formed by independent α-helices and mainly antiparallel β-sheets
multi-domain proteins
membrane and cell surface proteins and peptides
"small" proteins
coiled-coil proteins
low-resolution protein structures
peptides and fragments
designed proteins of non-natural sequence



α-helices in the hemoglobin protein

Automated methods to infer protein domains based on structure
General idea: structural domains are compact, globular sub-structures with more interactions within themselves than with the rest of the protein.



β-sheets in the avian src kinase protein

Methods to identify protein domains

Automated methods to infer protein domains based on sequence conservation
PSI-BLAST (position-specific iterated BLAST)

1. Use BLAST (algorithm for local alignment) to find good matches to a query protein of interest
2. From the matches with a maximum E-value construct a position-specific scoring matrix (PSSM)
3. Use the PSSM to find additional matches in the database and re-iterate

PSI-BLAST

Input: HuD, search against the protein DB at NCBI
(http://blast.ncbi.nlm.nih.gov/Blast.cgi)

Sequences producing significant alignments with E-value BETTER than threshold

Accession	Description	Max score	Total score	Query coverage	E value	Links
NEW NP_068771.2	ELAV-like protein 4 isoform 1 [Homo sapiens] >ref|XP_001110490.1| PRED	788	788	100%	0.0	U G
NEW NP_001033787.1	ELAV-like protein 4 isoform b [Mus musculus] >ref|XP_002715150.1| PRED	786	786	100%	0.0	U G
NEW AAA58396.1	brain protein [Homo sapiens] >gb|AAK57541.1| HUDPRO1 [Homo sapiens]	786	786	100%	0.0	G
NEW XP_001492578.1	PREDICTED: similar to ELAV-like 4 isoform 4 [Equus caballus]	785	785	100%	0.0	U G
NEW XP_001135120.1	PREDICTED: ELAV-like 4 isoform 4 [Pan troglodytes] >emb|CAI14638.1| El	784	784	100%	0.0	U G
NEW NP_034618.2	ELAV-like protein 4 isoform a [Mus musculus] >sp|Q61701.1|ELAV4_MOUS	783	783	100%	0.0	U G
NEW XP_860272.1	PREDICTED: similar to ELAV (embryonic lethal, abnormal vision, Drosophila	782	782	100%	0.0	U G
NEW EFB22722.1	hypothetical protein PANDA_010481 [Ailuropoda melanoleuca]	780	780	99%	0.0	
NEW XP_849969.1	PREDICTED: similar to ELAV-like protein 4 (Paraneoplastic encephalomyeliti	778	778	100%	0.0	U G
NEW NP_001071119.1	ELAV-like protein 4 [Rattus norvegicus] >sp|O09032.1|ELAV4_RAT RecNam	773	773	98%	0.0	U G
NEW AAH48159.1	Elavl4 protein [Mus musculus]	769	769	100%	0.0	G
NEW NP_001138246.1	ELAV-like protein 4 isoform 2 [Homo sapiens] >ref|XP_001110714.1| PRED	749	749	100%	0.0	U G
NEW NP_001075075.1	ELAV-like 4 [Bos taurus] >ref|XP_859918.1| PREDICTED: similar to ELAV (	747	747	100%	0.0	U G
NEW XP_001135046.1	PREDICTED: similar to RNA binding protein Elavl4 isoform 3 [Pan troglodyte	746	746	100%	0.0	U G
NEW XP_001110568.1	PREDICTED: similar to ELAV-like 4 isoform 3 [Macaca mulatta] >ref|XP_00	746	746	99%	0.0	U G
NEW XP_001492503.1	PREDICTED: similar to ELAV-like 4 isoform 1 [Equus caballus]	745	745	100%	0.0	U G
NEW NP_001138247.1	ELAV-like protein 4 isoform 3 [Homo sapiens] >ref|XP_001110671.1| PRED	745	745	99%	0.0	U G
NEW XP_001110750.1	PREDICTED: similar to ELAV-like protein 4 (Paraneoplastic encephalomyeliti	744	744	100%	0.0	U G
NEW AAH36071.1	ELAVL4 protein [Homo sapiens]	744	744	100%	0.0	G
NEW NP_001138248.1	ELAV-like protein 4 isoform 4 [Homo sapiens] >ref|XP_001110634.1| PRED	744	744	100%	0.0	U G
NEW EAX06841.1	ELAV (embryonic lethal, abnormal vision, Drosophila)-like 4 (Hu antigen D)	743	743	99%	0.0	G
NEW XP_859949.1	PREDICTED: similar to ELAV-like protein 4 (Paraneoplastic encephalomyeliti	743	743	99%	0.0	U G
NEW CAM19665.1	ELAV (embryonic lethal, abnormal vision, Drosophila)-like 4 (Hu antigen D)	743	743	99%	0.0	
NEW XP_860065.1	PREDICTED: similar to ELAV (embryonic lethal, abnormal vision, Drosophila	743	743	100%	0.0	G
NEW NP_001138249.1	ELAV-like protein 4 isoform 5 [Homo sapiens] >ref|XP_001135525.1| PRED	743	743	99%	0.0	G
NEW NP_001156871.1	ELAV-like protein 4 isoform d [Mus musculus] >ref|XP_860413.1| PREDICT	743	743	100%	0.0	U G
NEW XP_532585.2	PREDICTED: similar to ELAV-like protein 4 (Paraneoplastic encephalomyeliti	742	742	100%	0.0	U G
NEW XP_859879.1	PREDICTED: similar to ELAV-like protein 4 (Paraneoplastic encephalomyeliti	742	742	99%	0.0	U G
NEW XP_002715153.1	PREDICTED: ELAV-like 4-like isoform 4 [Oryctolagus cuniculus]	742	742	99%	0.0	G
NEW XP_001492529.2	PREDICTED: similar to ELAV-like 4 isoform 2 [Equus caballus]	741	741	99%	0.0	U G

Constructing the PSSM

Part of the alignment returned in the first iteration of PSI-BLAST

	Position	i i+1 i+2 i+3 i+4 i+5 i+6 i+7 i+8 . . .	
XP_001091917	373	FADGGQKKRQNPNKYIPNGRPWHREGE--------------AGMTLTYDPTTAAIQNGFYPSPYSIATNRMITQTSITP	437
NP_001030438	223	FADGGQKKRQNPNKYIPNGRPWHREGE--------------AGMTLTYDPTTAAIQNGFYPSPYSIATNRMITQTSITP	287
XP_001493156	190	FADGGQKKRQNPNKYIPNGRPWHREGE---VRL---------AGMTLTYDPTTAAIQNGFYPSPYSIATNRMITQTSITP	257
NP_001163350	187	FADGGPKKKNLFKTPDPNARAWRDVSA--------------EGIPVAYDPTMQ--QNG--------------VSVNVGT	235
XP_001894673		----------	
XP_002196753	219	FADGGQKKRQNQNKYIQNGRAWHREGE------------VRLAGMTLTYDPTTAALQNGFYPSPYSITANRMITQTSITP	286
NP_990355	190	FADGGQKKRQNQNKYIQNGRAWHREGE------------VRLAGMTLTYDPTTAALQNGFYPSPYSITANRMITQTSITP	257
CAA54628	190	FADGGQKKRQNPNKYIPNGRPWHREGE---VRL---------AGMTLTYDPTTAAIQNGFYPSPYSIATNRMITQTSITP	257
NP_001135403	221	FADGGQKKRQNPNKYIPNGRPWPRDGE--------------AGMTLTYDPTTAALHNGFYPSPYSIATNRMITQTSLTP	285
NP_001012184	223	FADGGQKKRQNPNKYIPNGRPWPREGE--------------AGMTLTYDPTTAALHNGFYPSPYSIATNRMITQTSLTP	287
NP_001135404	221	FADGGQKKRQNPNKYIPNGRPWPRDGE--------------AGMTLTYDPTTAALHNGFYPSPYSIATNRMITQTSLTP	285
NP_002888	223	FADGGQKKRQNPNKYIPNGRPWHREGE--------------AGMTLTYDPTTAAIQNGFYPSPYSIATNRMITQTSITP	287
AAH57866	190	FADGGQKKRQNPNKYIPNGRPWPRDGE--------------AGMTLTYDPTTAALHNGFYPSPYSIATNRMITQTSLTP	254
NP_058520	223	FADGGQKKRQNPNKYIPNGRPWHREGEVRL------------AGMTLTYDPTTAAIQNGFYPSPYSIATNRMITQTSITP	290
NP_064692	223	FADGGQKKRQNPNKYIPNGRPWPRDGE--------------AGMTLTYDPTTAALHNGFYPSPYSIATNRMITQTSLTP	287
XP_001511073	190	FADGGQKKRQNQSKYIQNGRAWQREGE--------------AGMTLTYDPTTAALQNGFYPSPYSIATNRMITQTSITP	254
XP_001510957	223	FADGGQKKRQNQSKYIQNGRAWQREGE--------------AGMTLTYDPTTAALQNGFYPSPYSIATNRMITQTSITP	287
XP_001366603	190	FADGGQKKRQNQNKYIQNGRAWHREGE--------------AGMTLTYDPTTAALQNGFYPSPYSIATNRMITQTSITP	254
XP_001366555	223	FADGGQKKRQNQNKYIQNGRAWHREGE--------------AGMTLTYDPTTAALQNGFYPSPYSIATNRMITQTSITP	287
XP_418760	187	FADGGQKKRQNQSKYTQNGRPWPREGE--------------AGMALTYDP-TAAIQNGFYSSPYSIPTNRMIPQTSITP	250
NP_001070184	183	FADGGQKKRQSQSKYPQNGRPWPREGE--------------SGMALTYDP-T-AMQNGFYSSPYSISTNRMIAQTSITP	245
XP_001366652	190	FADGGQKKRQNQNKYIQNGRAWHREGE---------VRL---AGMTLTYDPTTAALQNGFYPSPYSIATNRMITQTSITP	257
NP_729057	176	FADGGPKKKNLFKTPDPNARAWRDVSA--------------EGIPVAYDPTMQ--QNG--------------VSVNVGT	224
XP_001370004	223	FADGGQKKRQNQSKYTQNGRPWPREGE--------------AGMALTYDP-TAAIQNGFYSSPYSIATNRMIPQTSITP	286
AAK30205		----------	
NP_729054	386	FADGGPKKKNLFKTPDPNARAWRDVSA--------------EGIPVAYDPTMQ--QNG--------------VSVNVGT	434
XP_002035429	386	FADGGPKKKNLFKTPDPNARAWRDVSA--------------EGIPVAYDPTMQ--QNG--------------VSVNVGT	434
XP_001370033	223	FADGGQKKRQNQSKYTQNGRPWPREGE--------------AGMALTYDP-TAAIQNGFYSSPYSIATNRMIPQTSITP	286
XP_002093780	386	FADGGPKKKNLFKTPDPNARAWRDVSA--------------EGIPVAYDPTMQ--QNG--------------VSVNVGT	434

Domain boundaries will be defined implicitly by the evolutionary constraints

Databases for profile HMMs

- Pfam http://www.sanger.ac.uk/Pfam focus on divergent domains
- Prosite http://www.expasy.ch/prosite focus on functional sites
- Smart http://smart.embl-heidelberg.de as Pfam
- ProDom http://prodom.toulouse.inra.fr/prodom/doc/prodoc.html clusters of sequences
- PRINTS http://www.bioinf.manchester.ac.uk/dbbrowser/PRINTS focus on defining protein families

Protein domain, pattern and motif databases are bundled together in Interpro http://www.ebi.ac.uk/interpro

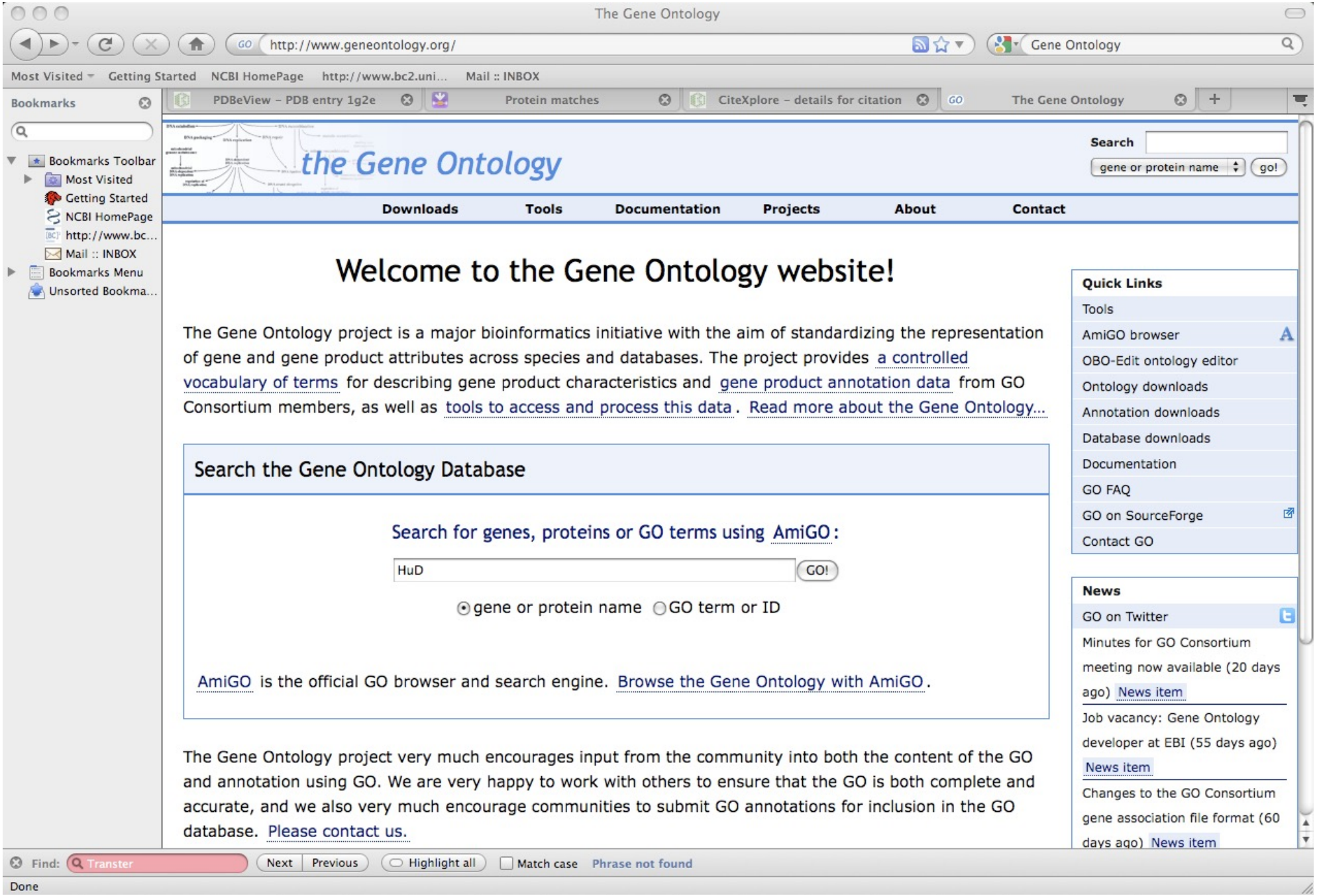
Functional annotation

What is functional annotation?

- Assignment of molecular functions to proteins encoded in a genome
- Inference of membership in metabolic and regulatory networks.

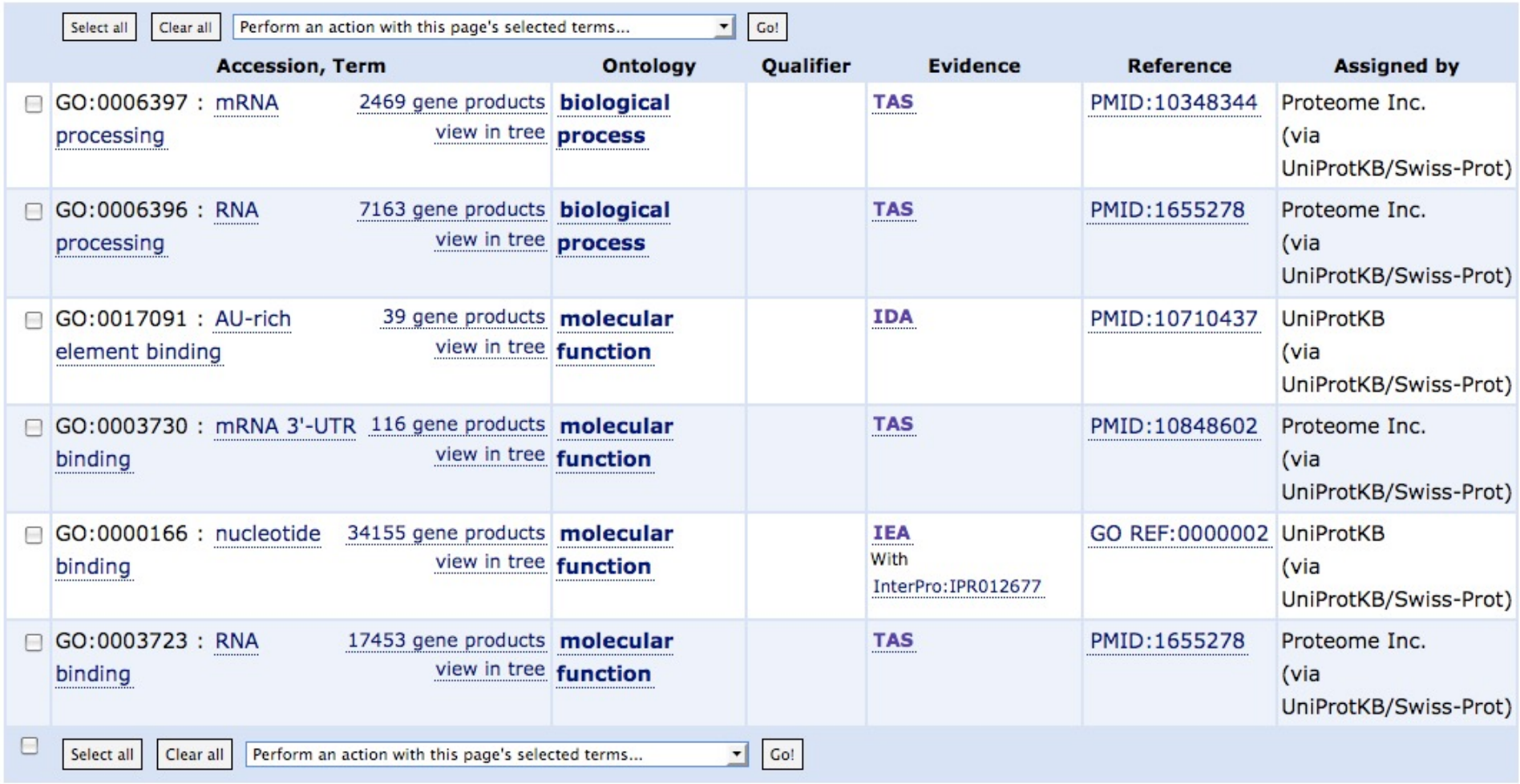
General idea: if specific protein domains are associated with specific functions, by identifying the domains encoded in a protein we will be able to infer what the protein “does”.

Gene Ontology www.geneontology.org



Gene Ontology www.geneontology.org

Gene Ontology www.geneontology.org



Select all | Clear all | Perform an action with this page's selected terms... | Go!

Accession, Term	Ontology	Qualifier	Evidence	Reference	Assigned by
GO:0006397 : mRNA processing — 2469 gene products — view in tree	biological process		TAS	PMID:10348344	Proteome Inc. (via UniProtKB/Swiss-Prot)
GO:0006396 : RNA processing — 7163 gene products — view in tree	biological process		TAS	PMID:1655278	Proteome Inc. (via UniProtKB/Swiss-Prot)
GO:0017091 : AU-rich element binding — 39 gene products — view in tree	molecular function		IDA	PMID:10710437	UniProtKB (via UniProtKB/Swiss-Prot)
GO:0003730 : mRNA 3'-UTR binding — 116 gene products — view in tree	molecular function		TAS	PMID:10848602	Proteome Inc. (via UniProtKB/Swiss-Prot)
GO:0000166 : nucleotide binding — 34155 gene products — view in tree	molecular function		IEA With InterPro:IPR012677	GO REF:0000002	UniProtKB (via UniProtKB/Swiss-Prot)
GO:0003723 : RNA binding — 17453 gene products — view in tree	molecular function		TAS	PMID:1655278	Proteome Inc. (via UniProtKB/Swiss-Prot)

Select all | Clear all | Perform an action with this page's selected terms... | Go!

Gene Ontology www.geneontology.org

Biological process

A biological process is series of events accomplished by one or more ordered assemblies of molecular functions. Examples of broad biological process terms are cellular physiological process or signal transduction. Examples of more specific terms are pyrimidine metabolic process or alpha-glucoside transport. It can be difficult to distinguish between a biological process and a molecular function, but the general rule is that a process must have more than one distinct steps.

A biological process is not equivalent to a pathway; at present, GO does not try to represent the dynamics or dependencies that would be required to fully describe a pathway.

Further information can be found in the process ontology documentation.

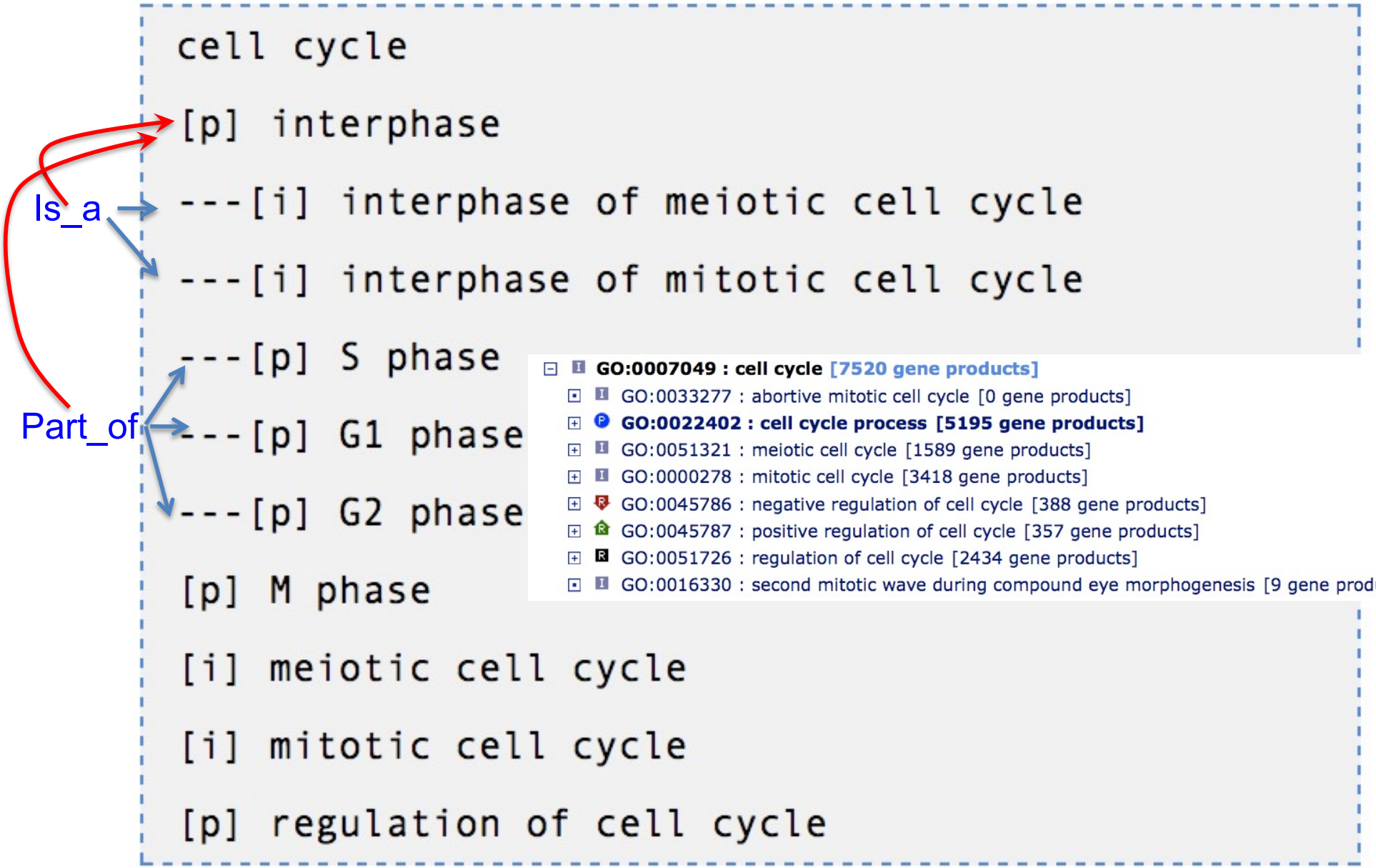
Molecular function

Molecular function describes activities, such as catalytic or binding activities, that occur at the molecular level. GO molecular function terms represent activities rather than the entities (molecules or complexes) that perform the actions, and do not specify where or when, or in what context, the action takes place. Molecular functions generally correspond to activities that can be performed by individual gene products, but some activities are performed by assembled complexes of gene products. Examples of broad functional terms are catalytic activity, transporter activity, or binding; examples of narrower functional terms are adenylate cyclase activity or Toll receptor binding.

It is easy to confuse a gene product name with its molecular function, and for that reason many GO molecular functions are appended with the word "activity". The documentation on the function ontology explains more about GO functions and the rules governing them.

Back to top

Biological process ontology: cell cycle



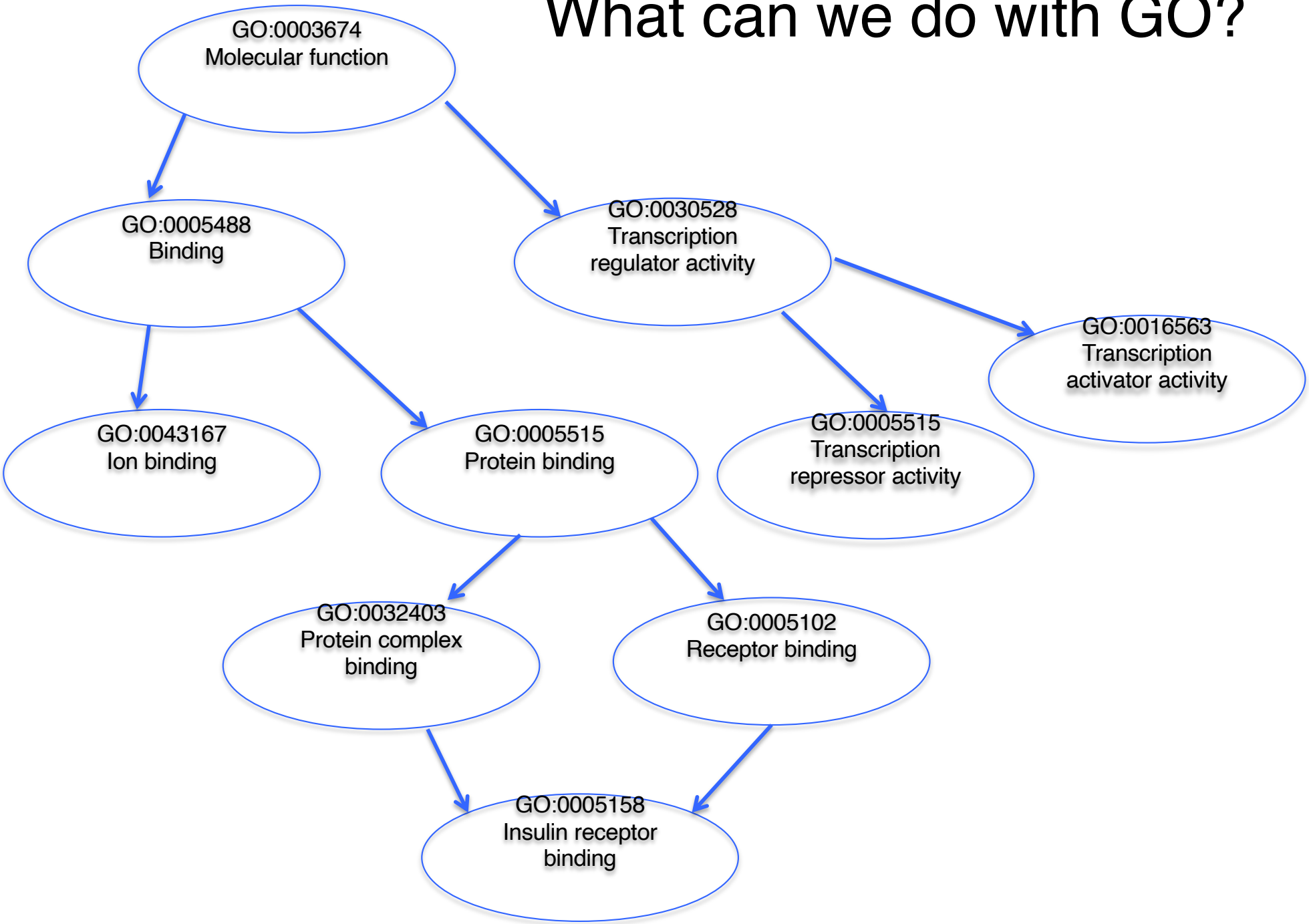
P21359 NF1_HUMAN Neurofibromin
Q13627 DYR1A_HUMAN Dual specificity tyrosine-phosphorylation-regulated kinase 1A
Q10571 MN1_HUMAN Probable tumor suppressor protein MN1
P11182 ODB2_HUMAN Lipoamide acyltransferase component of branched-chain alpha-keto acid dehydrogenase complex, mitochondrial
Q8TAP4 LMO3_HUMAN LIM domain only protein 3
O60315 ZEB2_HUMAN Zinc finger E-box-binding homeobox 2
Q9H2E6 SEM6A_HUMAN Semaphorin-6A
Q86YT6 MIB1_HUMAN E3 ubiquitin-protein ligase MIB1
Q5T0B9 ZN362_HUMAN Zinc finger protein 362
Q9Y5I0 PCDAD_HUMAN Protocadherin alpha-13
Q02156 KPCE_HUMAN Protein kinase C epsilon type
Q86VS8 HOOK3_HUMAN Protein Hook homolog 3
Q3L8U1 CHD9_HUMAN Chromodomain-helicase-DNA-binding protein 9
Q8N5G2 MACOI_HUMAN Macoilin
Q99743 NPAS2_HUMAN Neuronal PAS domain-containing protein 2
Q8WY54 PPM1E_HUMAN Protein phosphatase 1E
Q13635 PTC1_HUMAN Protein patched homolog 1
P49116 NR2C2_HUMAN Nuclear receptor subfamily 2 group C member 2
Q09013 DMPK_HUMAN Myotonin-protein kinase
Q9NZI8 IF2B1_HUMAN Insulin-like growth factor 2 mRNA-binding protein 1
Q0VAM2 RGF1B_HUMAN Ras-GEF domain-containing family member 1B
Q13591 SEM5A_HUMAN Semaphorin-5A
Q9C0K0 BC11B_HUMAN B-cell lymphoma/leukemia 11B
O94933 SLIK3_HUMAN SLIT and NTRK-like protein 3
Q9Y5H6 PCDA8_HUMAN Protocadherin alpha-8
Q99592 ZN238_HUMAN Zinc finger protein 238
Q5PSV4 BRM1L_HUMAN Breast cancer metastasis-suppressor 1-like protein
Q12926 ELAV2_HUMAN ELAV-like protein 2
P40424 PBX1_HUMAN Pre-B-cell leukemia transcription factor 1
O75525 KHDR3_HUMAN KH domain-containing, RNA-binding, signal transduction-associated protein 3
Q96QR8 PURB_HUMAN Transcriptional activator protein Pur-beta
Q99583 MNT_HUMAN Max-binding protein MNT
9H2X6 HIPK2_HUMAN Homeodomain-interacting protein kinase 2
O95948 ONEC2_HUMAN One cut domain family member 2
Q96M96 FGD4_HUMAN FYVE, RhoGEF and PH domain-containing protein 4
Q7RTV3 ZN367_HUMAN Zinc finger protein 367
P25105 PTAFR_HUMAN Platelet-activating factor receptor
Q96QB1 RHG07_HUMAN Rho GTPase-activating protein 7
Q96P70 IPO9_HUMAN Importin-9
Q2KHR2 RFX7_HUMAN DNA-binding protein RFX7
Q86V48 LUZP1_HUMAN Leucine zipper protein 1
Q9BZQ8 NIBAN_HUMAN Protein Niban
Q8IVU1 IGDC3_HUMAN Immunoglobulin superfamily DCC subclass member 3

What can we do with GO?

High-throughput experiments typically yield lists of “potentially interesting genes”

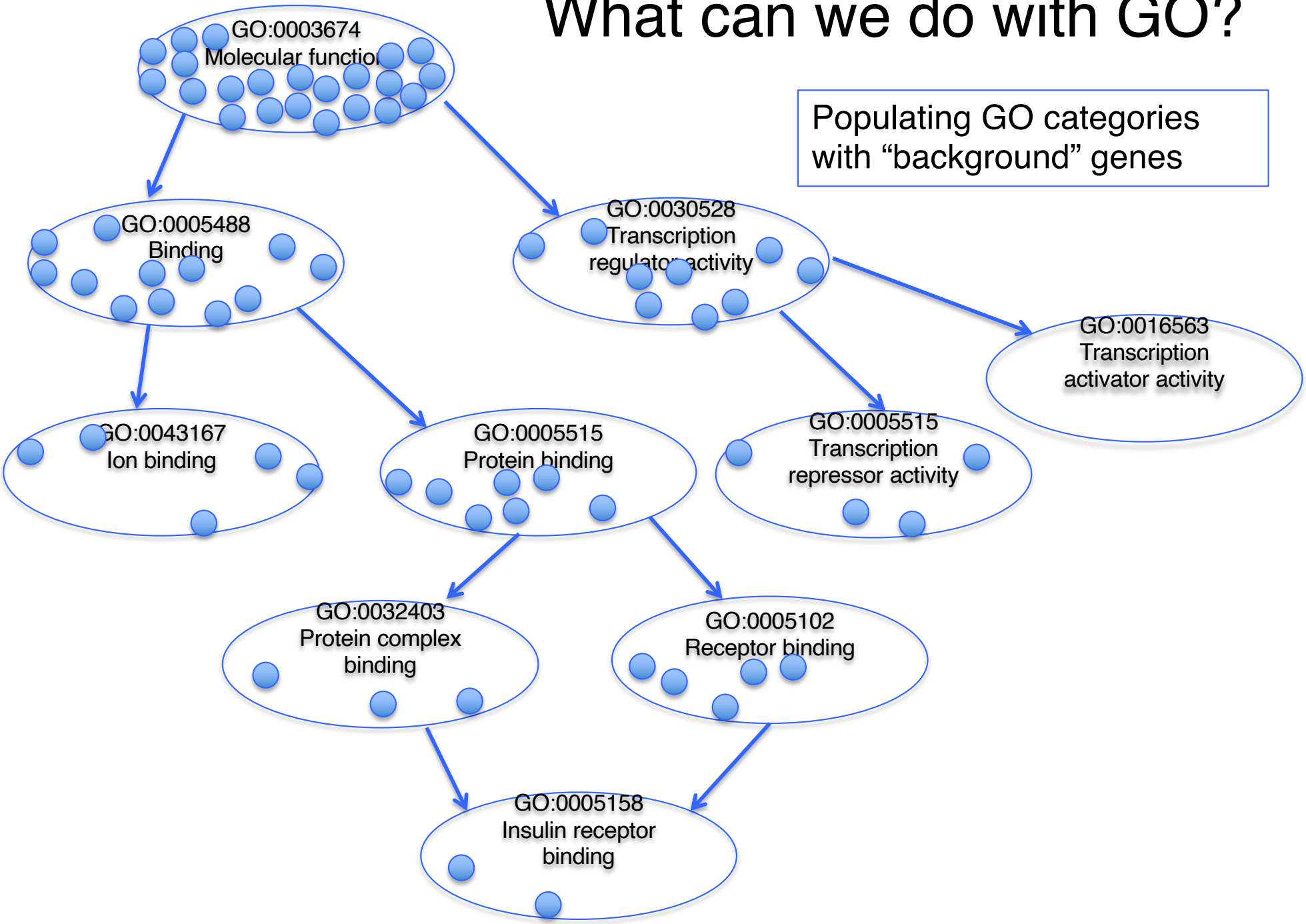
Can we zoom onto the processes that were most perturbed in the experiment?

What can we do with GO?

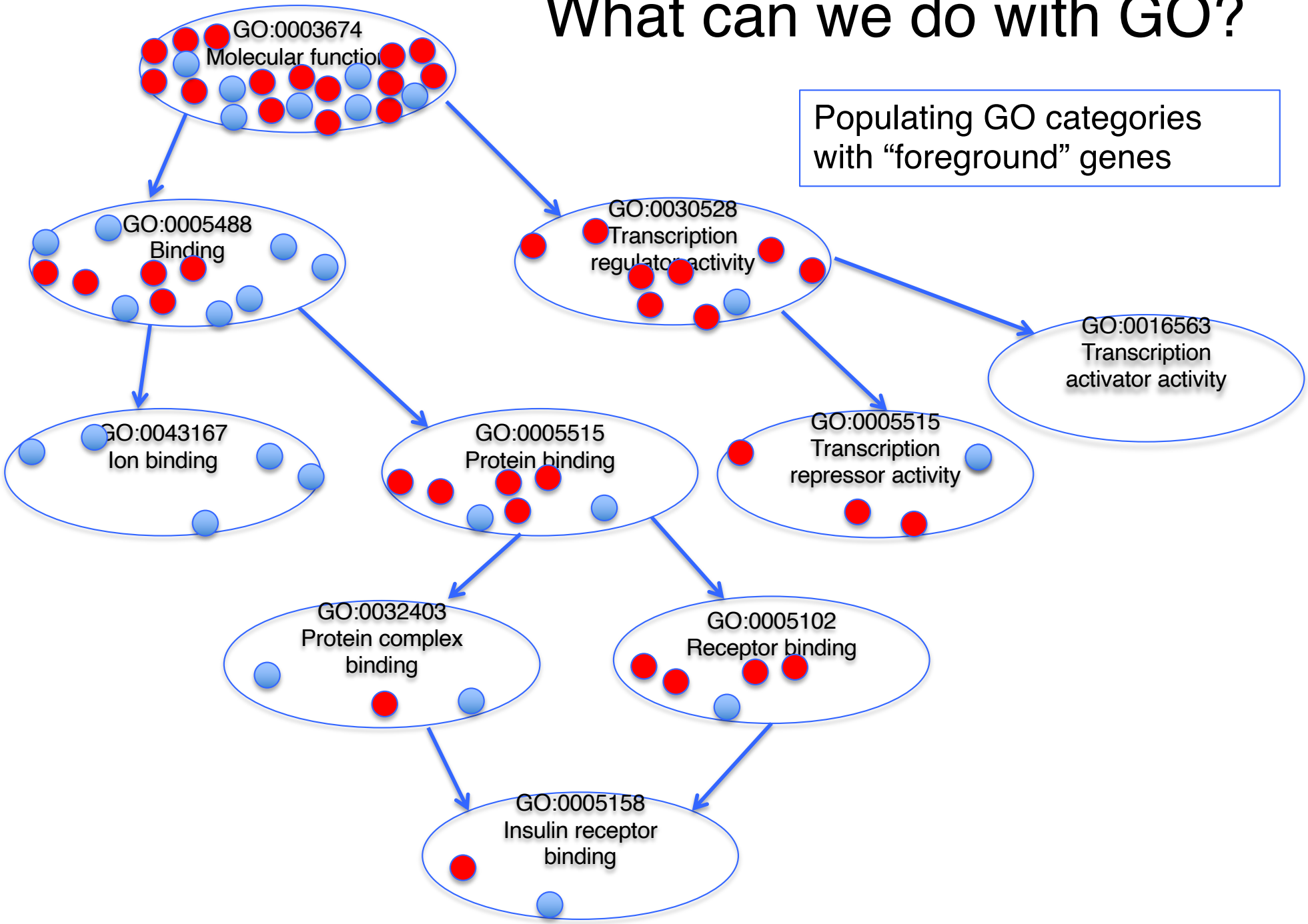


What can we do with GO?

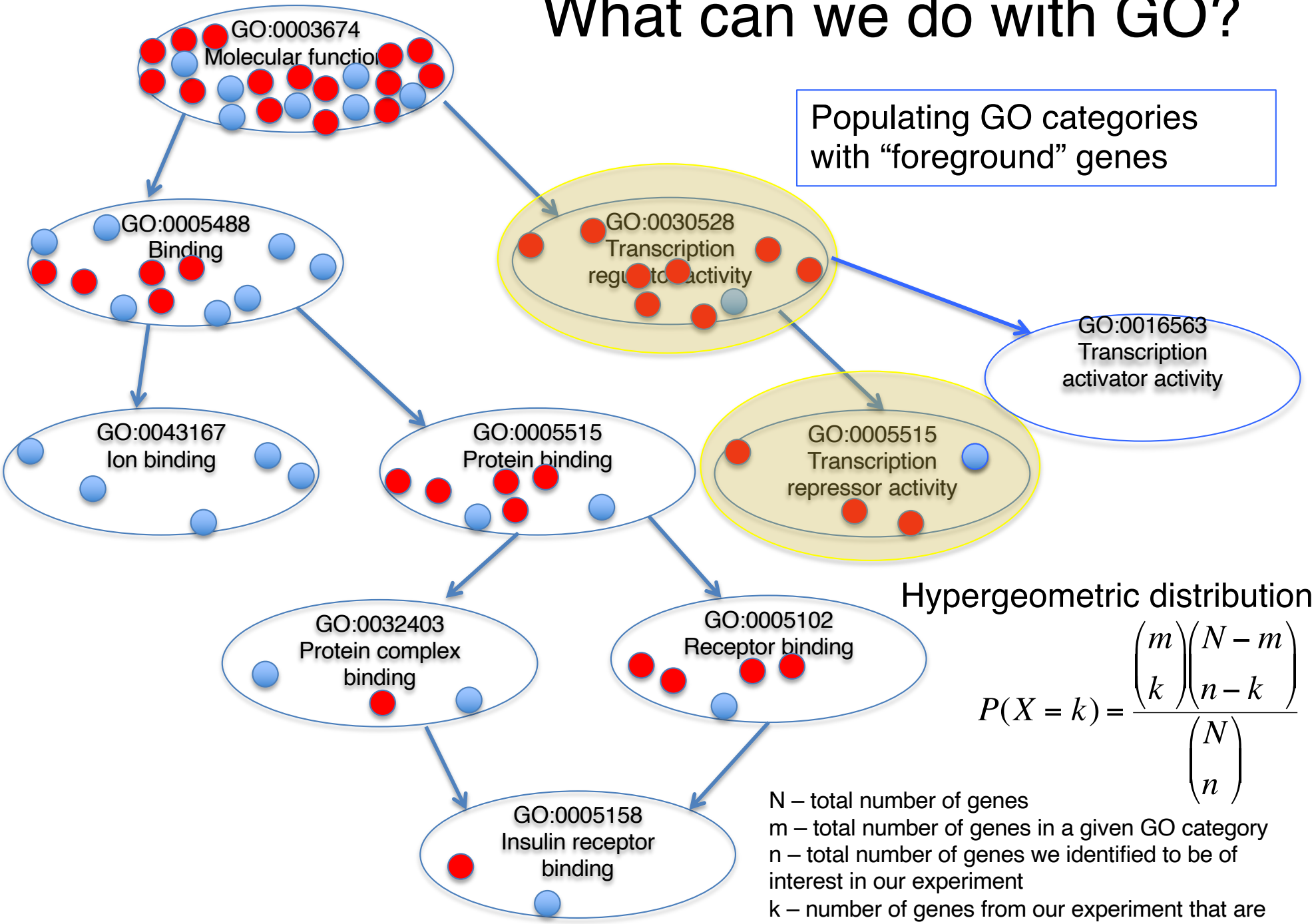
Populating GO categories with “background” genes



What can we do with GO?



What can we do with GO?



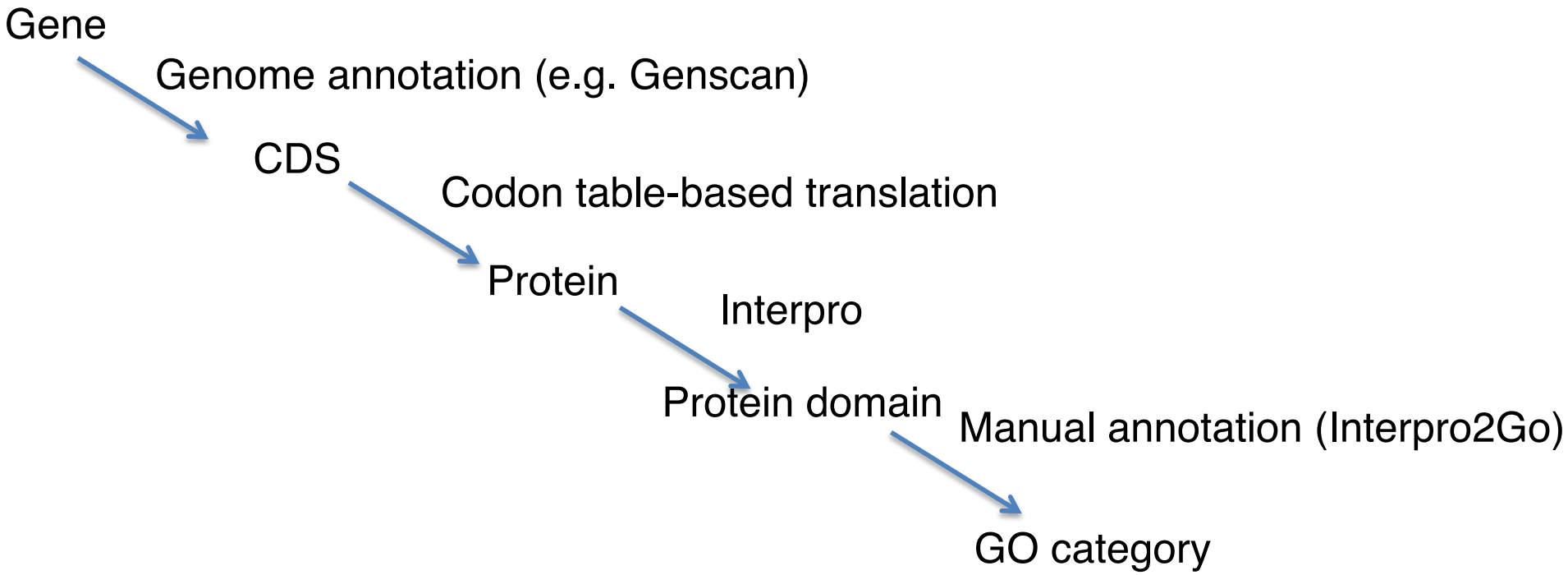
Populating GO categories with "foreground" genes

Hypergeometric distribution

$$P(X = k) = \frac{\binom{m}{k}\binom{N-m}{n-k}}{\binom{N}{n}}$$

N – total number of genes
m – total number of genes in a given GO category
n – total number of genes we identified to be of interest in our experiment
k – number of genes from our experiment that are assigned to the specific GO category

How do we go from genes to GO?



How do we go from genes to GO?

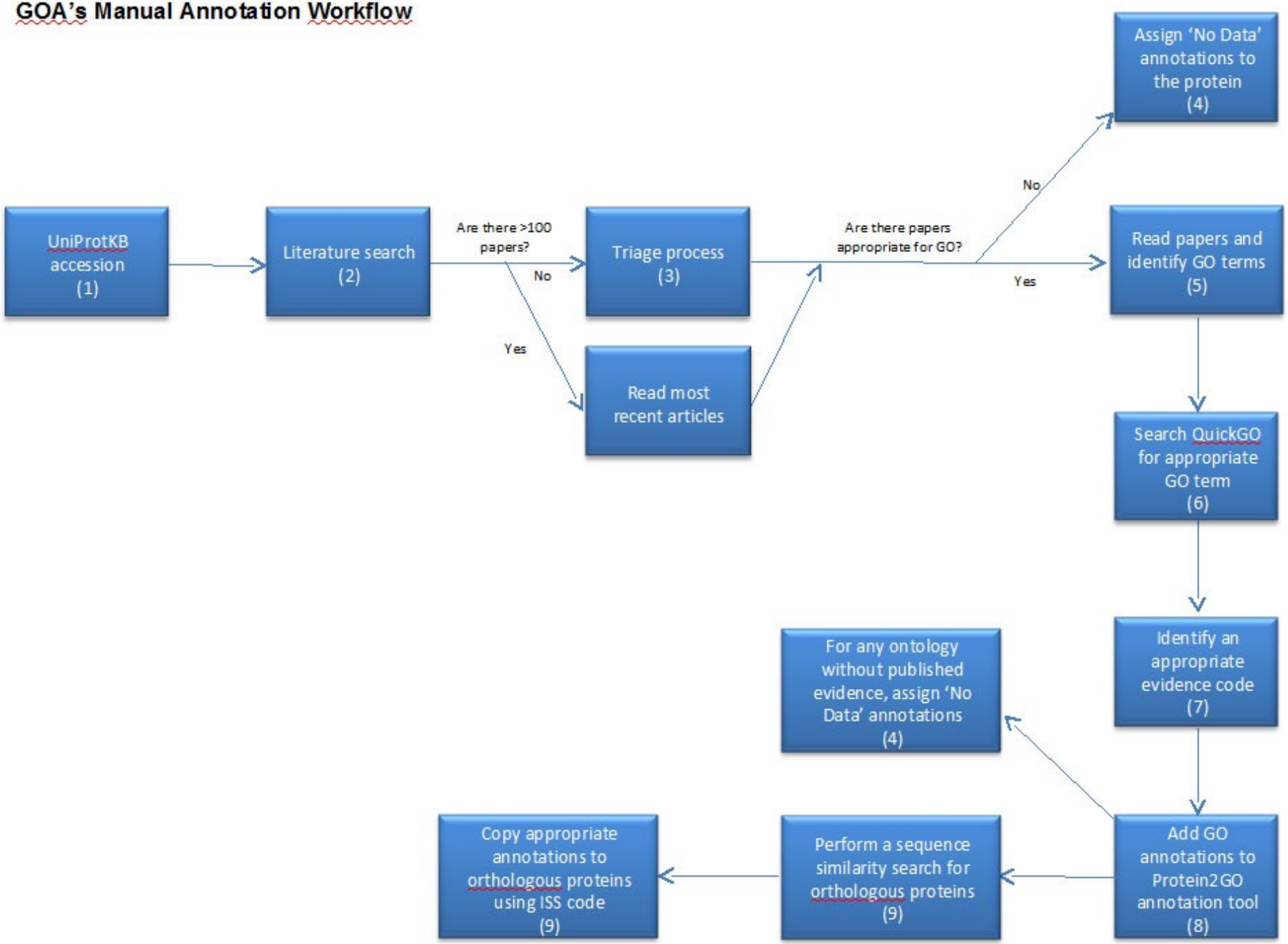
This InterPro2GO file is generated manually by the InterPro team at the EBI. To generate this table, curators compare InterPro and protein entries and for matching entries they;

- Look at the statistics on DE lines, keywords and comments
- Check how conserved the common annotation is
- Look for an appropriate GO term at the most specific level to be relevant to all proteins in that family

The mapping file is then used to assign annotations to UniProtKB proteins at each GOA release. GO annotations using this technique receive the evidence code Inferred from Electronic Annotation (IEA).

This method has been evaluated at 91-100% accurate (Camon *et. al.* 2005).

GOA's Manual Annotation Workflow



What can we do with GO?

For our gene list…