Pairwise sequence alignments

Sequence alignment

A way to arrange DNA, RNA or protein sequences to reflect how they are related to each other

Sequence alignment

A way to arrange DNA, RNA or protein sequences to reflect how they are related to each other

Example: alignment of mouse and rat hemoglobin alpha

```
>gi|37748075|gb|AAH59150.1| Hemoglobin alpha, adult chain 1 [Rattus norvegicus]
Length=142

 Score =  248 bits (632),  Expect = 5e-65
 Identities = 120/142 (84%), Positives = 127/142 (89%), Gaps = 0/142 (0%)

Mouse Hb  1    MVLSGEDKSNIKAAWGKIGGHGAEYGAEALERMFASFPTTKTYFPHFDVSHGSAQVKGHG  60
               MVLS +DK+NIK  WGKIGGHG EYG EAL+RMFA+FPTTKTYF H DVS GSAQVK HG
Rat Hb    1    MVLSADDKTNIKNCWGKIGGHGGEYGEEALQRMFAAFPTTKTYFSHIDVSPGSAQVKAHG  60

Mouse Hb  61   KKVADALANAAGHLDDLPGALSALSDLHAHKLRVDPVNFKLLSHCLLVTLASHHPADFTP  120
               KKVADALA AA H++DLPGALS LSDLHAHKLRVDPVNFK LSHCLLVTLA HHP DFTP
Rat Hb    61   KKVADALAKAADHVEDLPGALSTLSDLHAHKLRVDPVNFKFLSHCLLVTLACHHPGDFTP  120

Mouse Hb  121  AVHASLDKFLASVSTVLTSKYR  142
               A+HASLDKFLASVSTVLTSKYR
Rat Hb    121  AMHASLDKFLASVSTVLTSKYR  142
```

Why do we make sequence alignments?

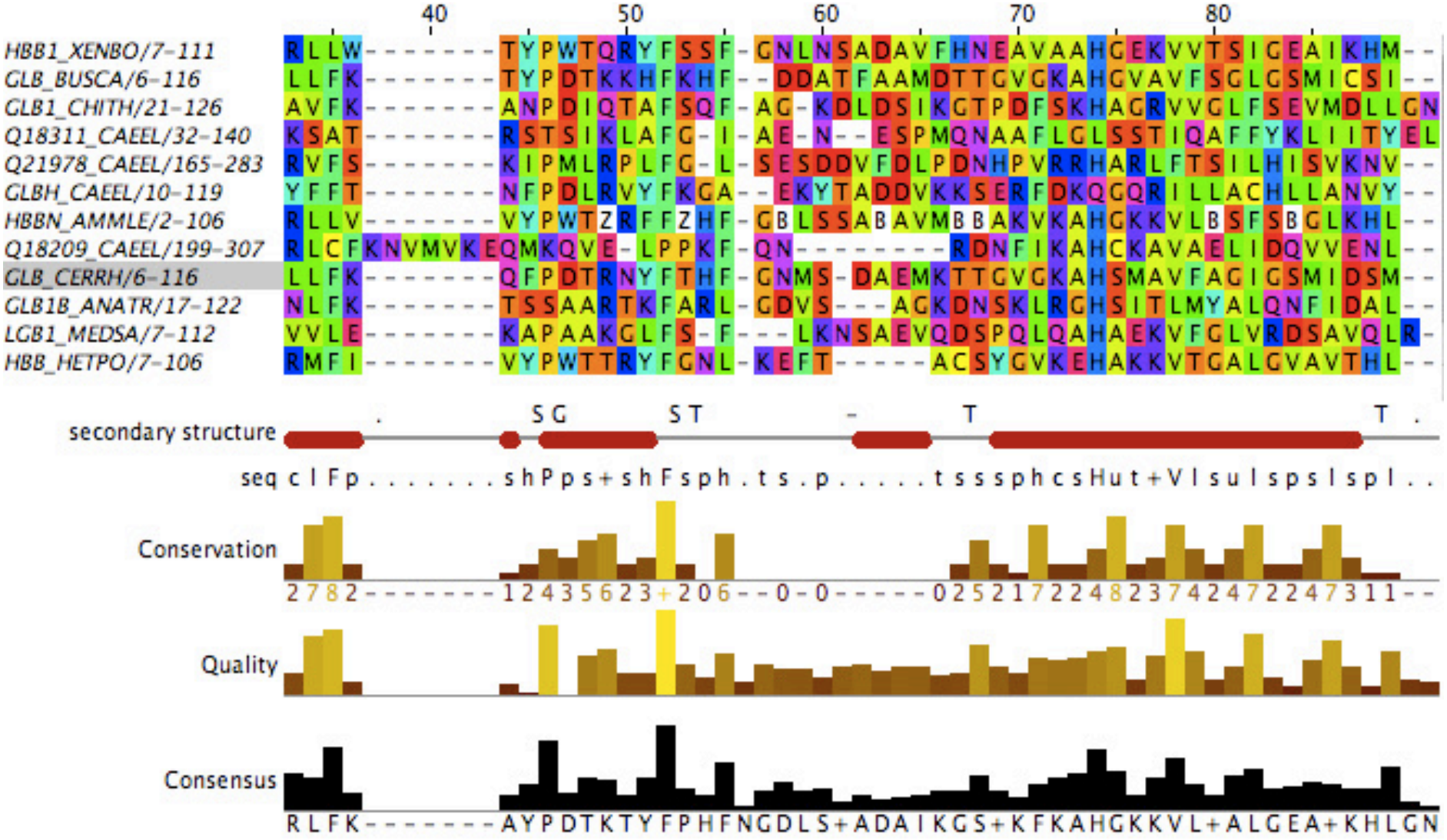
To infer evolutionary relationships between sequences,

uncover sequences that are under selective constraint,

find sequences in databases, etc.

Identification of positions under selective constraint due to their contribution to some function

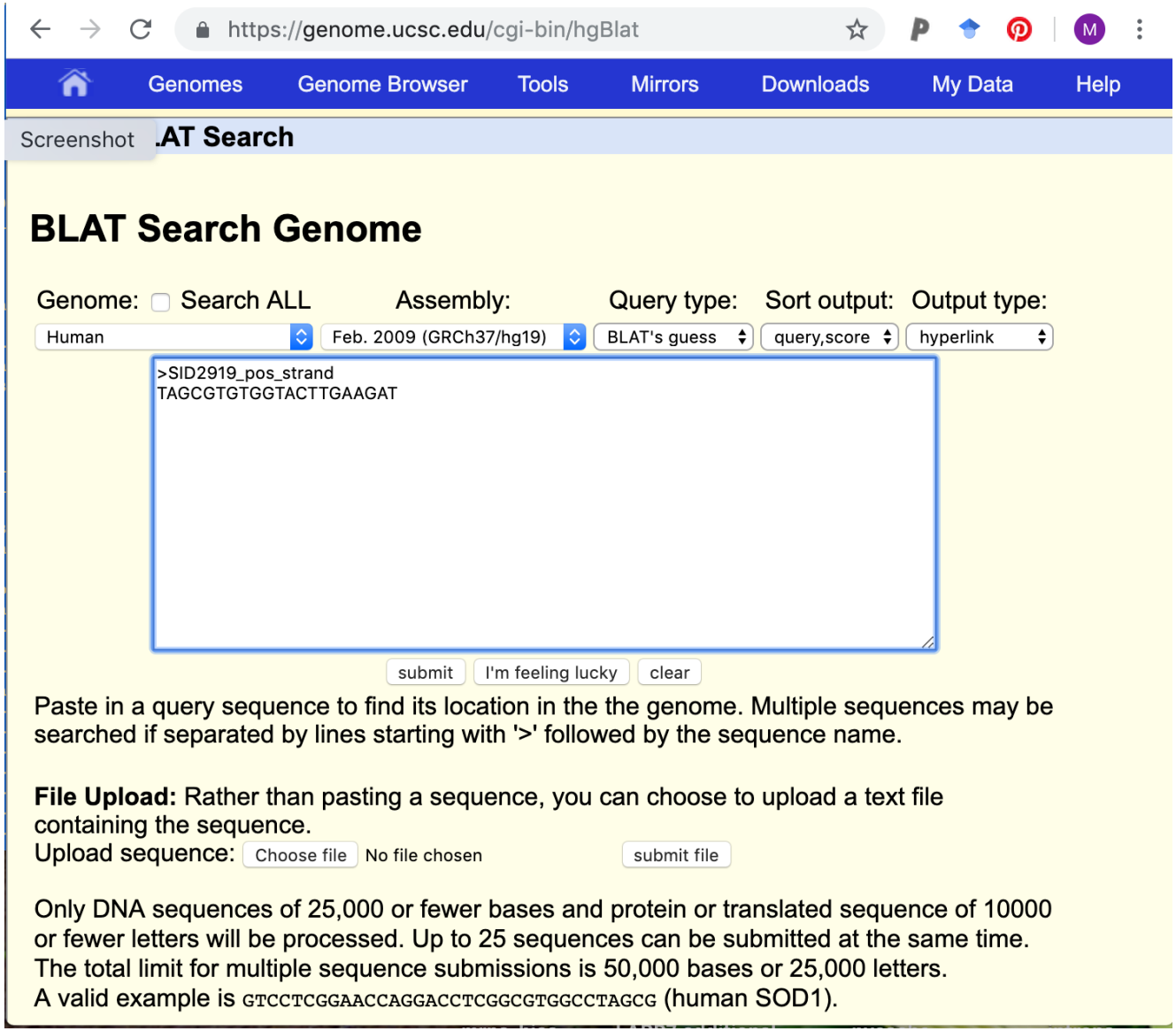
Example: Model of the globin domain.



Finding sequences in databases

A short RNA fragment obtained through deep sequencing

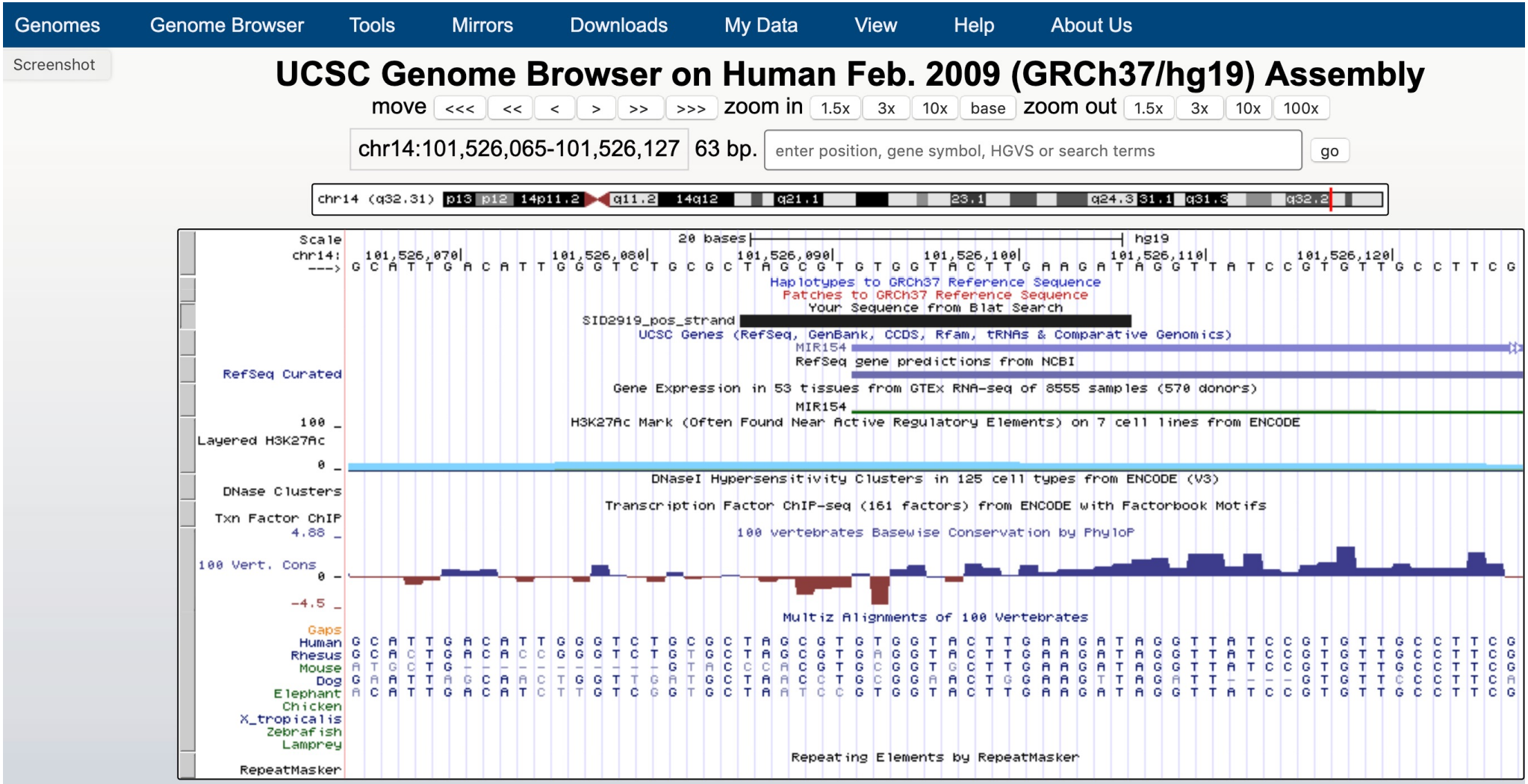
http://genome.ucsc.edu



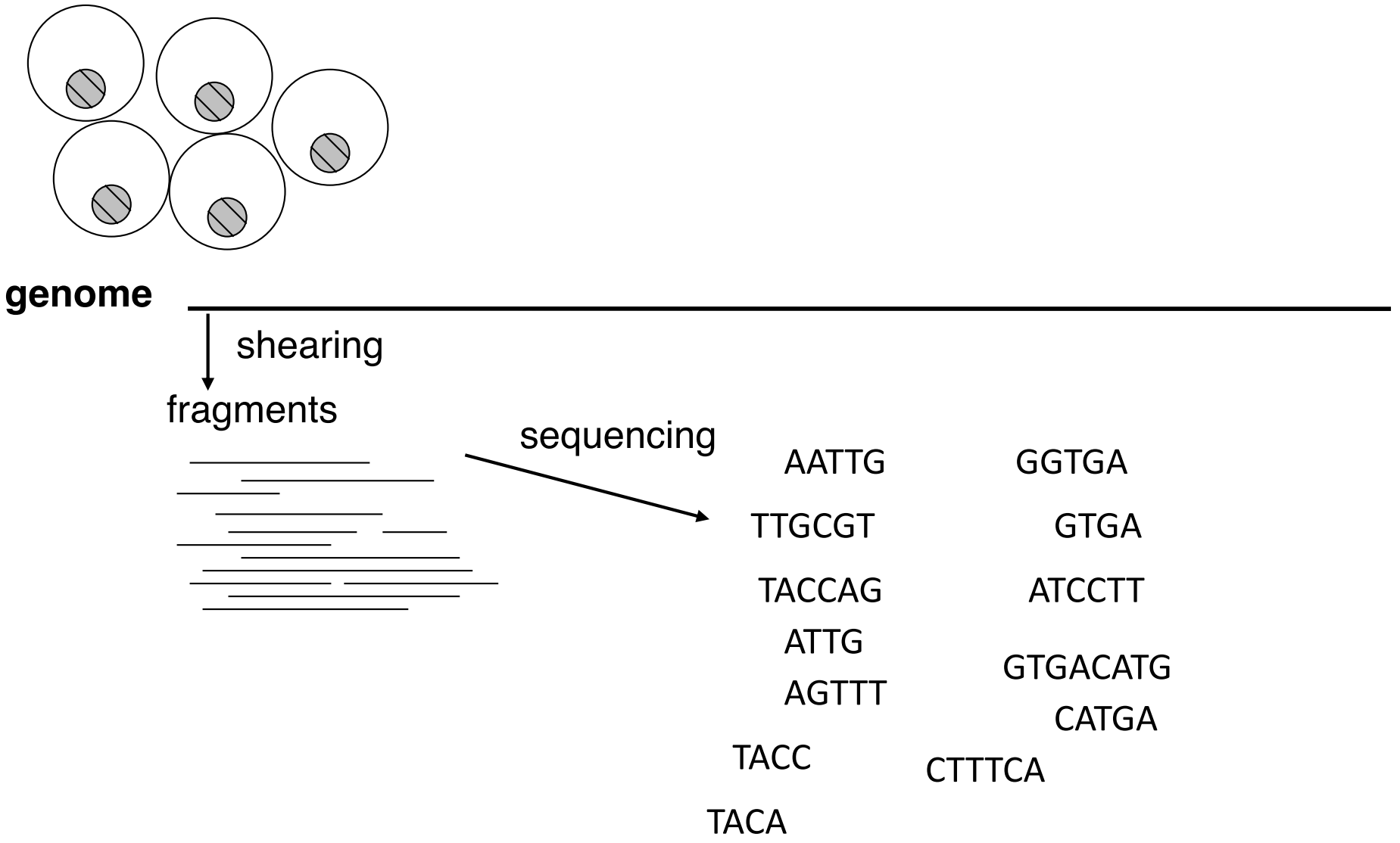
Finding sequences in databases

A short RNA fragment obtained through deep sequencing

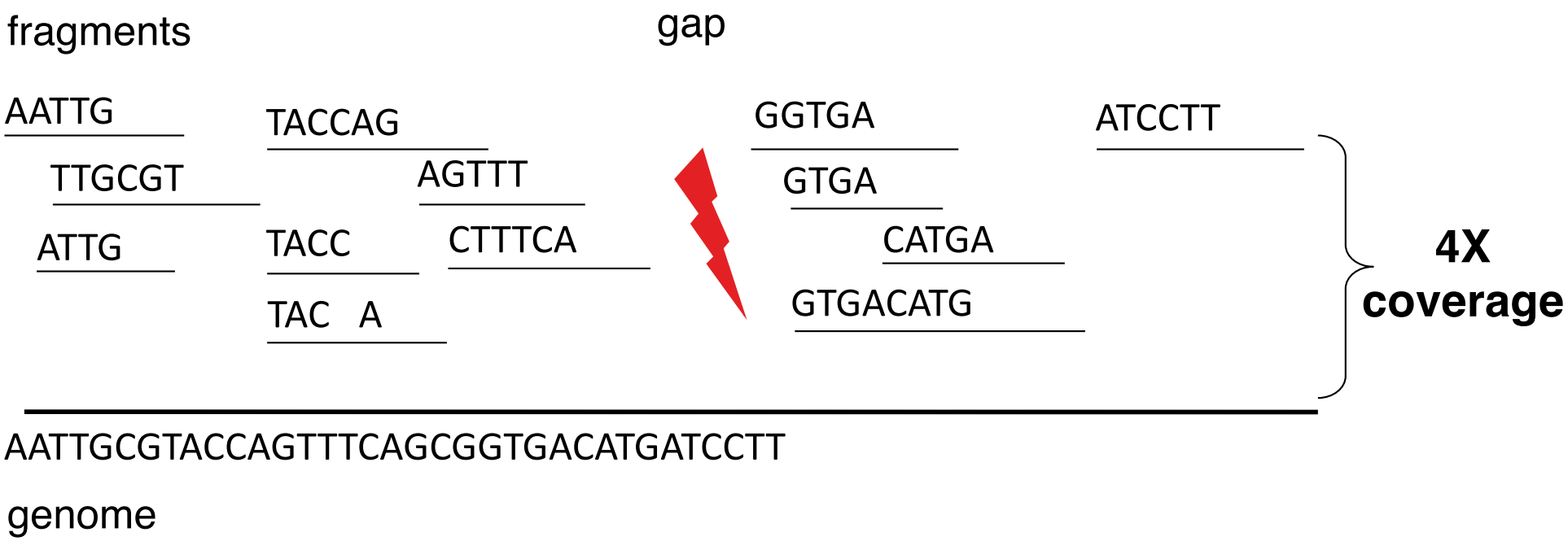
http://genome.ucsc.edu



Genome assembly



Genome assembly

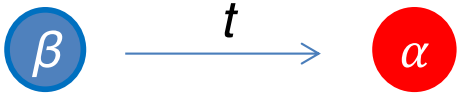


What does it take to make these alignments?

Reminder: Substitution process

Our substitution model gives the matrix of rates of substitution of base β by base α.

• For a finite amount of time t the substitution probabilities are given by:



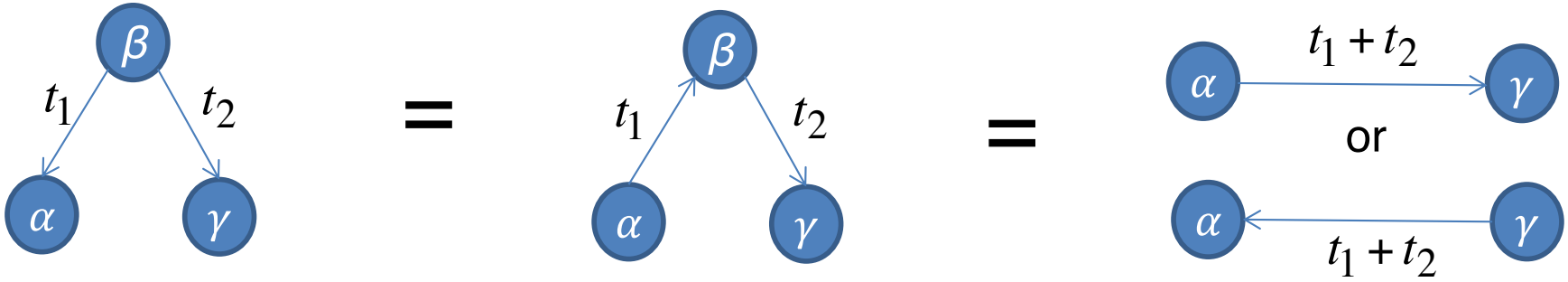
$$\frac{\partial P(\alpha|\beta,t)}{\partial t} = \sum_{\gamma} R_{\alpha\gamma}\ P(\gamma|\beta,t) \Longrightarrow P(\alpha|\beta,t) = (e^{Rt})_{\alpha\beta}$$

• In the limit of long time we reach a *limit distribution*: $\lim_{t\to\infty} P(\alpha|\beta,t) = \pi_\alpha$

• Most substitution rate matrices are *reversible* in the sense that:

$$P(\alpha|\beta,t)\pi_\beta = P(\beta|\alpha,t)\pi_\alpha$$

• For a reversible model we have: $\sum_\beta P(\alpha|\beta,t_1)P(\gamma|\beta,t_2)\pi_\beta = P(\gamma|\alpha,t_1+t_2)\pi_\alpha$



Evolutionary change

The models of molecular evolution that we discussed so far only took into account nucleotide substitutions.
These are however, not the only mutations that take place in evolution.

```
ATGGGTGCGAGAGCGTCA
```

```
TTGGGAGCGAGAGTATCA
```

```
ATGGGTGCGAGAGCGTCA
```

```
ATGGGTCGAAGGCGT
```

Which is the ‘correct’ alignment?

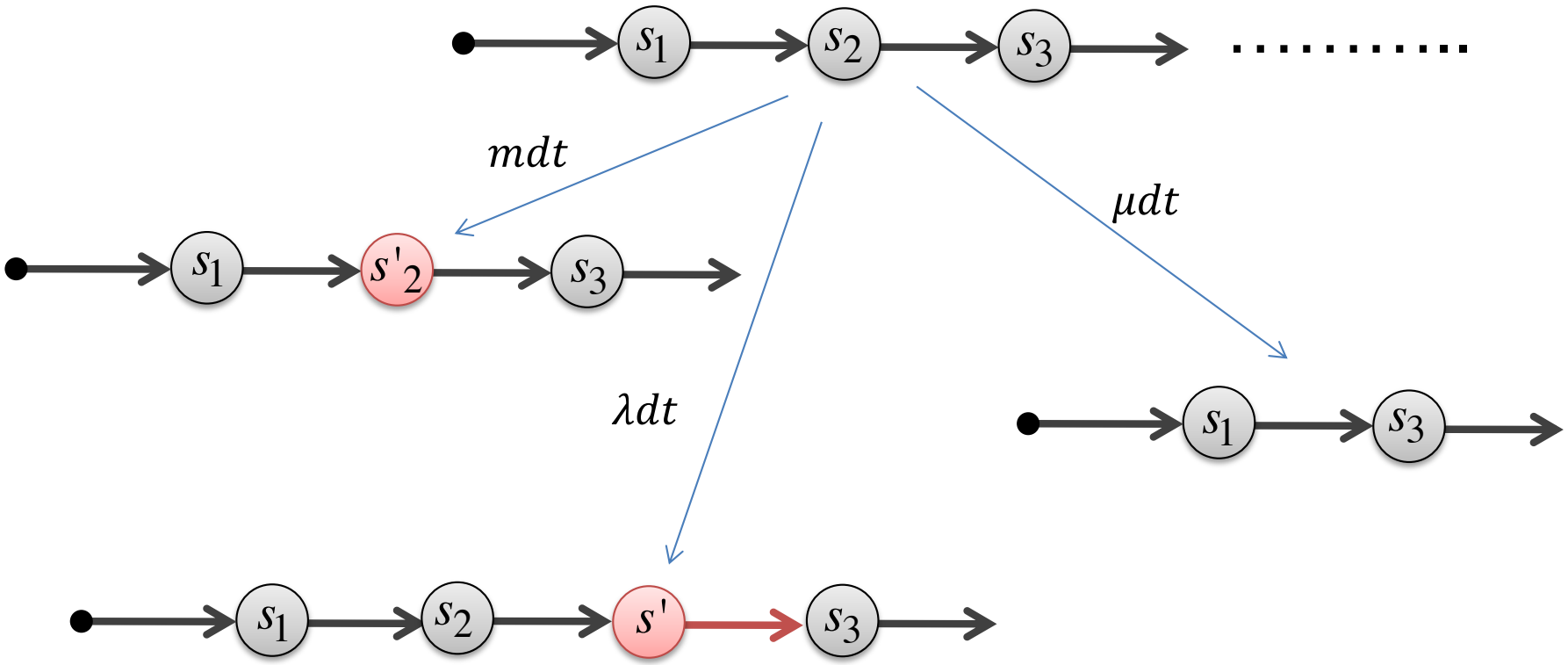
```
ATGGGTGCGAGAG-CGTCA
ATGGGT-CGA-AGGCGT--
****** *** ** ***
```

```
ATGGGTGCGAGA--GCGTCA
ATGGGT---CGAAGGCGT--
******    **  ****
```

Sequence evolution with insertions and deletions

Assume that in a very short time interval dt three types of events can happen:

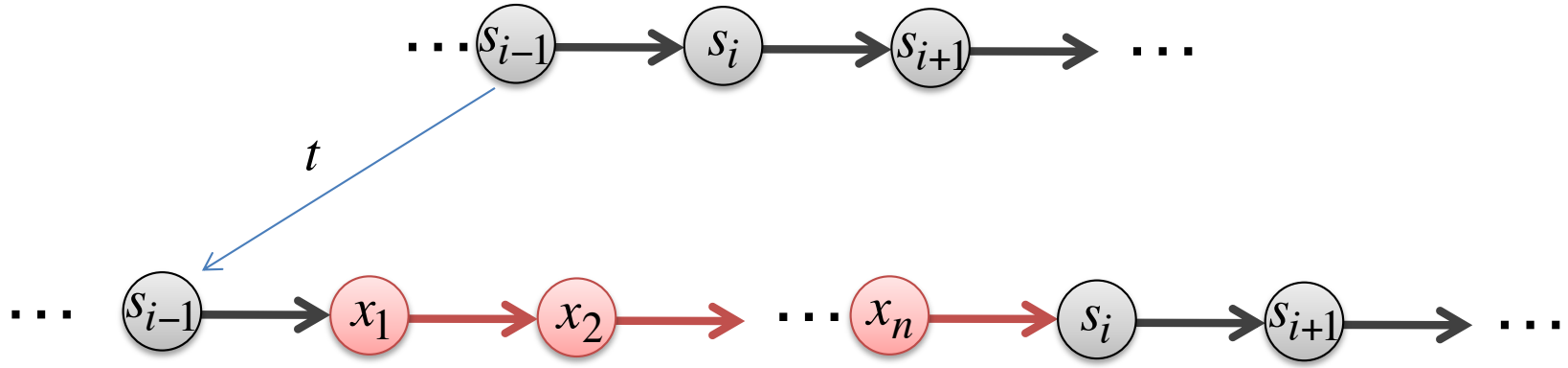
1. A base is mutated (with probability mdt per base).
2. A new base is inserted (with probability λdt after each base).
3. A base is deleted (with probability μdt per base).



Initial work:
Thorne, Kishino, and Felsenstein
Mol Evol **33** 114-124 (1991)

More recent treatment:
Holmes and Bruno
Bioinformatics **17** 803-820 (2001)

Modeling insertions and deletions: descendants of a single base



$p_n(t)$ = The probability that through the process of insertion and deletion over a time *t* a single node **survives** and leaves *n* descendants (including itself).

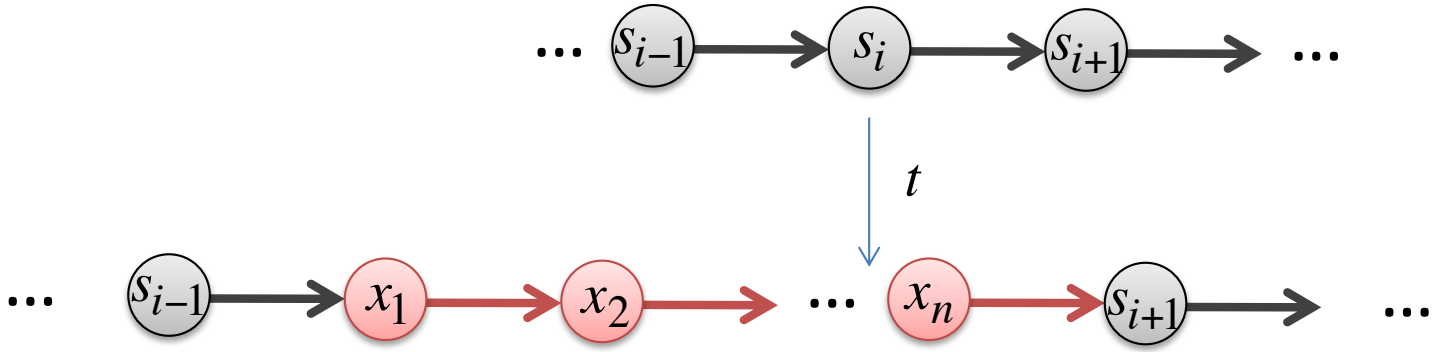
The probability $p_n(t)$ obeys the differential equation:

$$\frac{dp_n}{dt} = \lambda(n-1)p_{n-1} + \mu n p_{n+1} - (\lambda+\mu)np_n$$

which can be solved to give:

$$p_n = \alpha\beta^{n-1}(1-\beta) \text{ with } \alpha = e^{-\mu t} \text{ and } \beta = \frac{\lambda - \lambda e^{(\lambda-\mu)t}}{\mu - \lambda e^{(\lambda-\mu)t}}$$

Modeling insertions and deletions: descendants of a single base



$q_n(t)$ = The probability that through the process of insertion and deletion over a time *t* a single node ***disappears*** and leaves *n* extra nodes after it.

For $n > 0$ the probability $q_n(t)$ obeys the differential equation:

$$\frac{dq_n}{dt} = \lambda(n-1)q_{n-1} + \mu(n+1)q_{n+1} - (\lambda+\mu)nq_n + \mu p_{n+1}$$

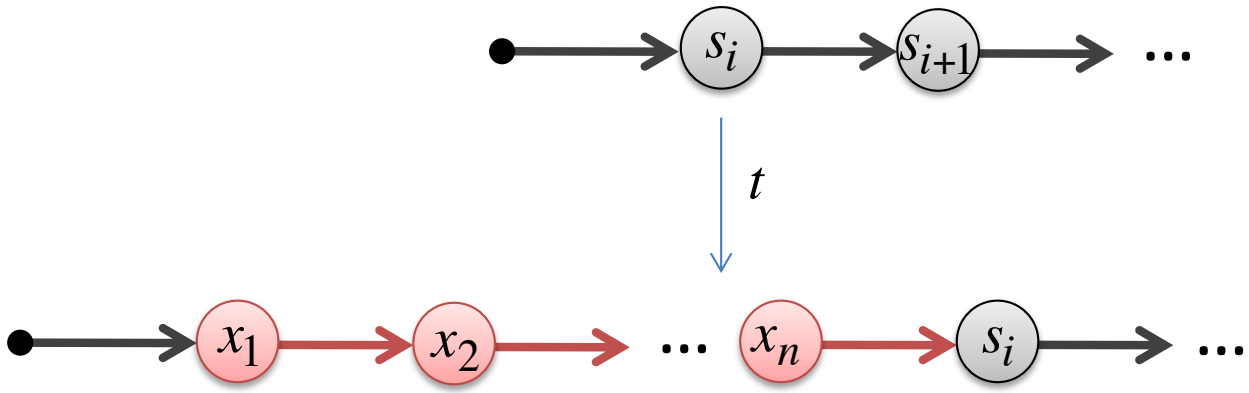
For $n = 0$:

$$\frac{dq_0}{dt} = \mu(q_1 + p_1)$$

With $\gamma = 1 - \frac{\mu\left(1-e^{(\lambda-\mu)t}\right)}{\left(1-e^{-\mu t}\right)\left(\mu-\lambda e^{(\lambda-\mu)t}\right)}$

The solution is given by $q_n = \begin{cases} (1-\alpha)(1-\gamma) & \text{for } n = 0 \\ (1-\alpha)\gamma\beta^{n-1}(1-\beta) & \text{for } n > 0 \end{cases}$

Modeling insertions and deletions: descendants of the root link



$r_n(t)$ = The probability that through the process of insertion and deletion over a time *t* the immortal link at the start of the sequence leaves *n* nodes.

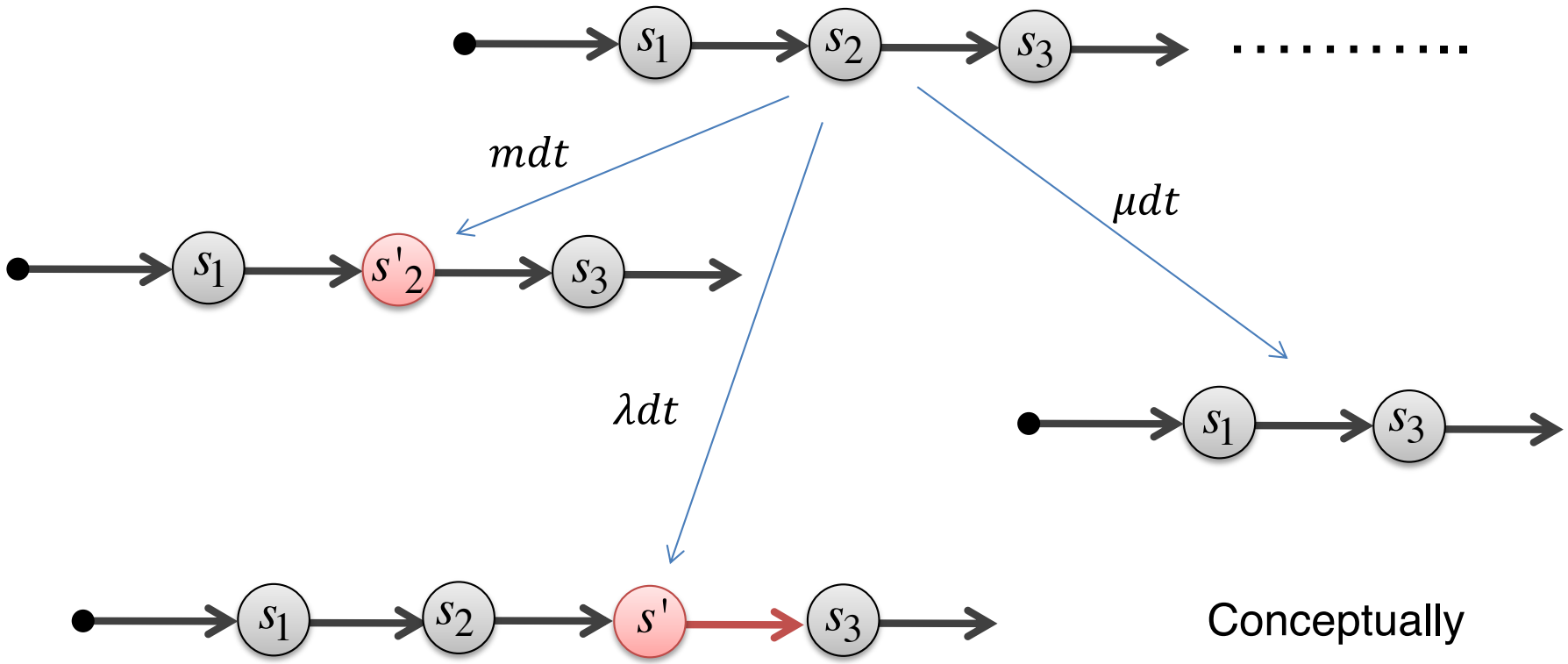
For $n > 0$ the probability $r_n(t)$ obeys the differential equation:

$$\frac{dr_n}{dt} = \lambda n r_{n-1} + \mu(n+1) r_{n+1} - \lambda(n+1) r_n - \mu n r_n$$

and for $n = 0$ $\frac{dr_0}{dt} = \mu r_1 - \lambda r_0$

The solution is given by $r_n = \beta^n(1-\beta)$

Summary of evolution with indels



$$p_n = \alpha\beta^{n-1}(1-\beta)$$

$$q_n = \begin{cases} (1-\alpha)(1-\gamma) & \text{for } n = 0 \\ (1-\alpha)\gamma\beta^{n-1}(1-\beta) & \text{for } n > 0 \end{cases}$$

$$r_n = \beta^n(1-\beta)$$

Conceptually

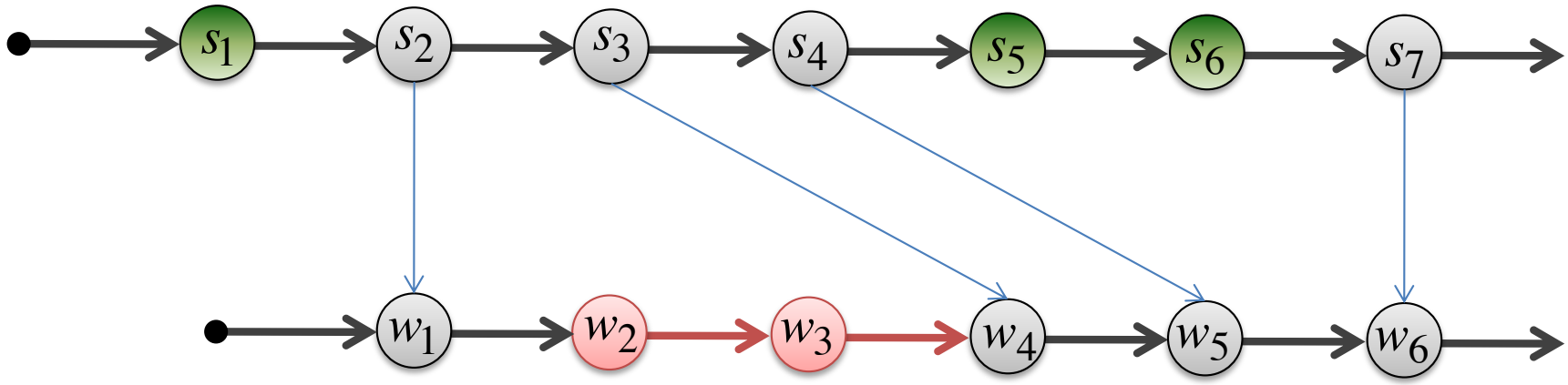
$\alpha = e^{-\mu t}$ probability that the ancestral residue survives

$\beta = \frac{\lambda - \lambda e^{(\lambda-\mu)t}}{\mu - \lambda e^{(\lambda-\mu)t}}$ probability of insertions given that the ancestral node survived

$\gamma = 1 - \frac{\mu\left(1-e^{(\lambda-\mu)t}\right)}{\left(1-e^{-\mu t}\right)\left(\mu-\lambda e^{(\lambda-\mu)t}\right)}$ probability of insertions given that the ancestral node disappeared

From evolution along a branch to alignments

Consider this example evolutionary scenario:



Which can be represented as the following alignment:

$S_1 S_2 - - S_3 S_4 S_5 S_6 S_7$

$- W_1 W_2 W_3 W_4 W_5 - - W_6$

The probability of this alignment has two components

1. The part due to insertion/deletion: $P(\sigma) = r_0 q_0 p_3 p_1 p_1 q_0 q_0 p_1$ …
2. The probability of the bases given the alignment:

$$P(\vec{s}, \vec{w}|\sigma) = \pi_{s_1} P(w_1|s_2) \pi_{s_2} \pi_{w_2} \pi_{w_3} P(w_4 |\ s_3)\ \pi_{s_3} P(w_5 |\ s_4)\ \pi_{s_4} \pi_{s_5} \pi_{s_6} P(w_6 |\ s_7)\ \pi_{s_7}$$

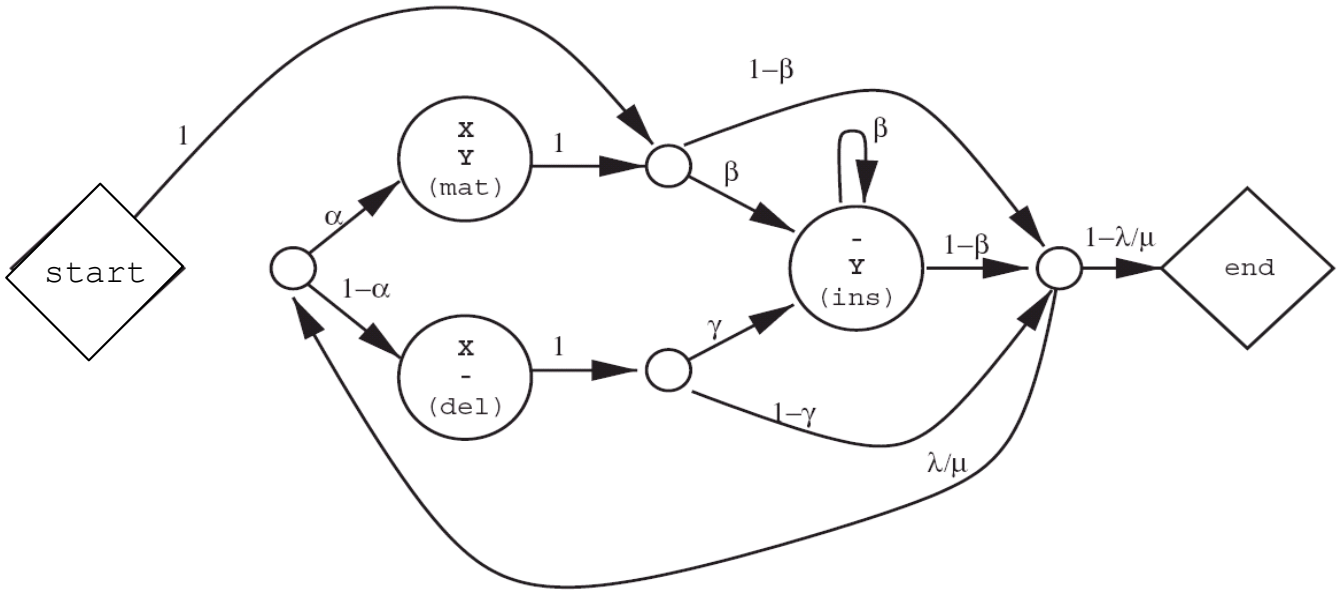
Pair-HMM representation of the evolution along a branch

The probability of any aligned pair of sequences calculated with the model

$$p_n = \alpha\beta^{n-1}(1-\beta) \quad q_n = \begin{cases} (1-\alpha)(1-\gamma) \text{ for } n = 0 \\ (1-\alpha)\gamma\beta^{n-1}(1-\beta) \text{ for } n > 0 \end{cases} \quad r_n = \beta^n(1-\beta)$$

$$\alpha = e^{-\mu t} \qquad \beta = \frac{\lambda - \lambda e^{(\lambda-\mu)t}}{\mu - \lambda e^{(\lambda-\mu)t}} \quad \gamma = 1 - \frac{\mu\left(1 - e^{(\lambda-\mu)t}\right)}{(1-e^{-\mu t})(\mu - \lambda e^{(\lambda-\mu)t})}$$

is given by the following pair-HMM



'Large circle' states correspond to emissions with associated probabilities.
'Small circle' states are not associated with emissions.

Let's use an example to verify this

```
X-----XX
YYYYYYYY
```

Under our generative model

$$p_n = \alpha\beta^{n-1}(1-\beta) \quad q_n = \begin{cases} (1-\alpha)(1-\gamma) \text{ for } n = 0 \\ (1-\alpha)\gamma\beta^{n-1}(1-\beta) \text{ for } n > 0 \end{cases} \quad r_n = \beta^n(1-\beta)$$

$$\alpha = e^{-\mu t} \qquad \beta = \frac{\lambda - \lambda e^{(\lambda-\mu)t}}{\mu - \lambda e^{(\lambda-\mu)t}} \quad \gamma = 1 - \frac{\mu\left(1 - e^{(\lambda-\mu)t}\right)}{(1-e^{-\mu t})(\mu - \lambda e^{(\lambda-\mu)t})}$$

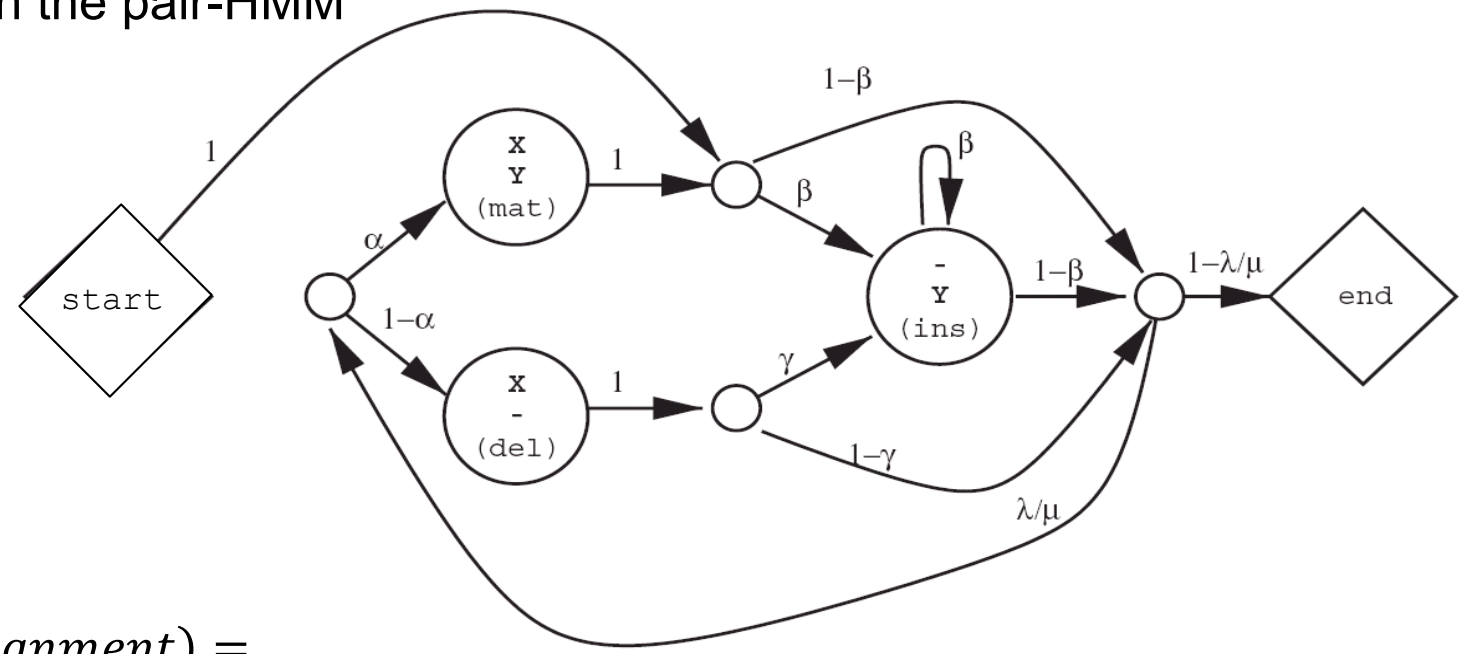
the likelihood of getting sequence Y from sequence X (ignoring emissions) is:

$$P(Y|X) = r_0 p_6 p_1 p_1 = (1-\beta)\alpha\beta^5(1-\beta)\alpha(1-\beta)\alpha(1-\beta) = \alpha^3\beta^5(1-\beta)^4$$

Let's use an example to verify this

```
X-----XX
YYYYYYYY
```

With the pair-HMM



$$P(X-Y\ alignment) =$$

$$\left(1(1-\beta)\frac{\lambda}{\mu}\right)(\alpha\cdot 1\cdot\beta)\left(\beta^4(1-\beta)\frac{\lambda}{\mu}\right)\left(\alpha\cdot 1\cdot(1-\beta)\frac{\lambda}{\mu}\right)\left(\alpha\cdot 1\cdot(1-\beta)\left(1-\frac{\lambda}{\mu}\right)\right)$$

$$=\alpha^3\beta^5(1-\beta)^4\left(\frac{\lambda}{\mu}\right)^3\left(1-\frac{\lambda}{\mu}\right)$$

Pair-HMM representation of the evolution along a branch

```
X-----XX
YYYYYYYY
```

Under our generative model, the likelihood of getting sequence Y from sequence X (ignoring emissions) would be:

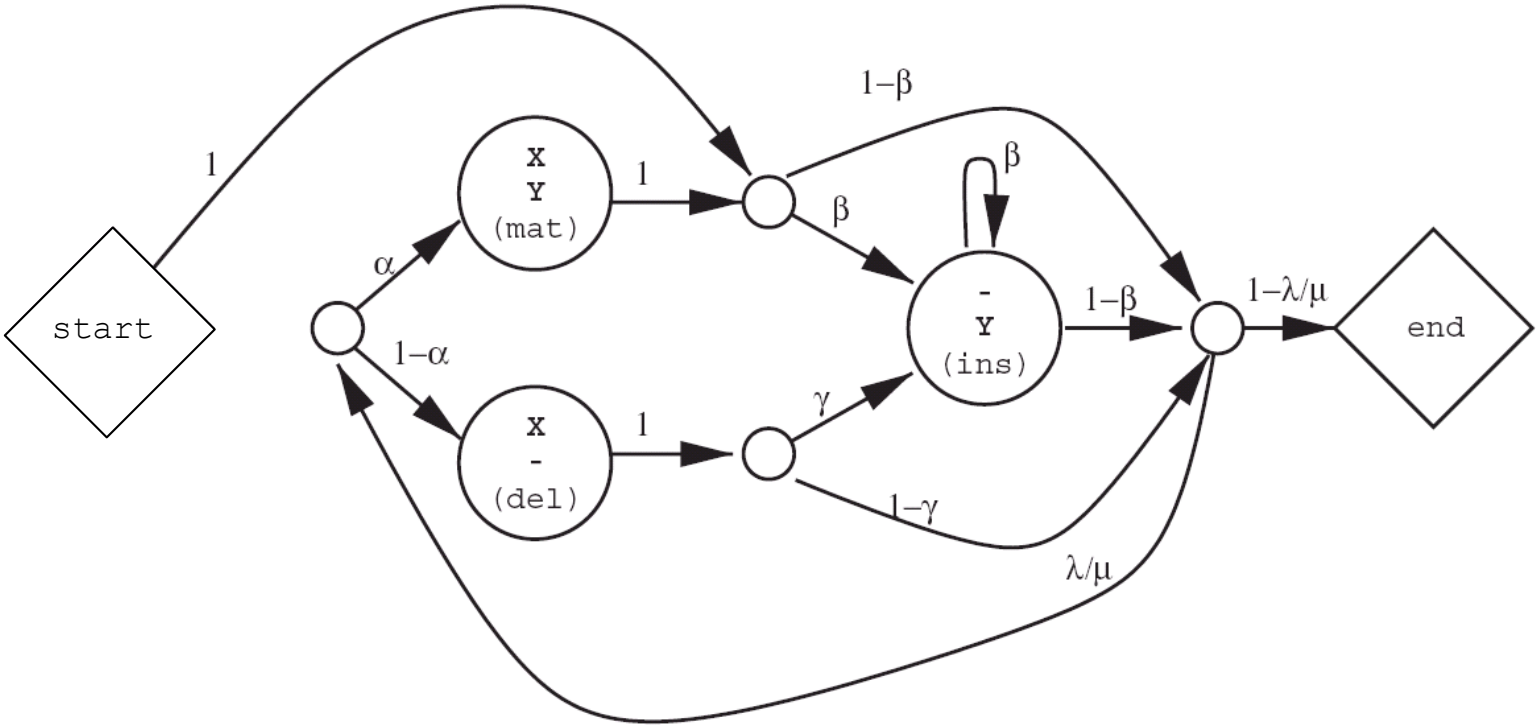
$$P(Y|X) = r_0 p_6 p_1 p_1 = (1-\beta)\alpha\beta^5(1-\beta)\alpha(1-\beta)\alpha(1-\beta) = \alpha^3\beta^5(1-\beta)^4$$

$$P(X-Y\ alignment) = \alpha^3\beta^5(1-\beta)^4\left(\frac{\lambda}{\mu}\right)^3\left(1-\frac{\lambda}{\mu}\right)$$

Why the difference?

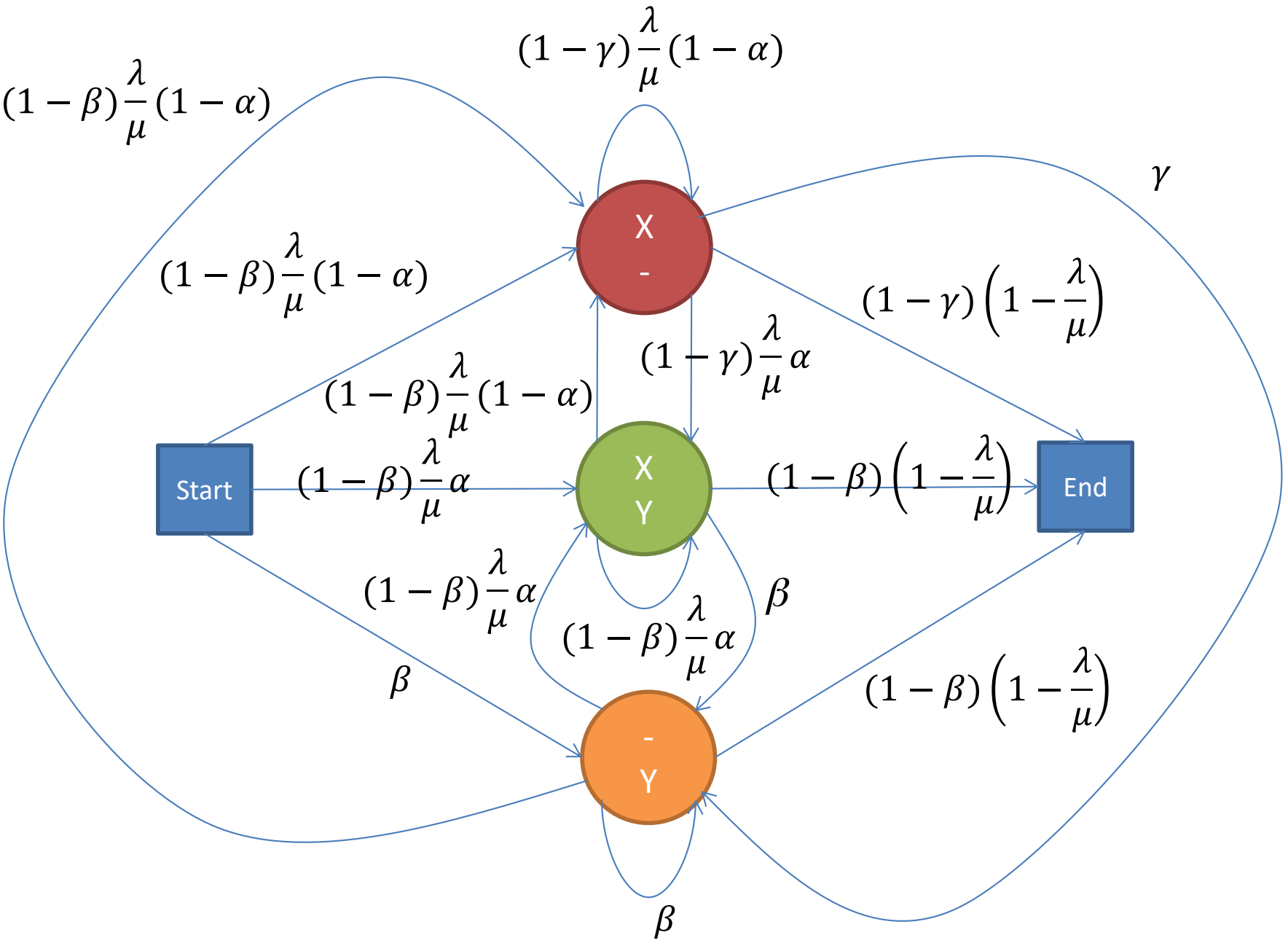
Because in the pair-HMM we also model explicitly the length of the alignment.

Pair-HMM representation of the evolution along a branch

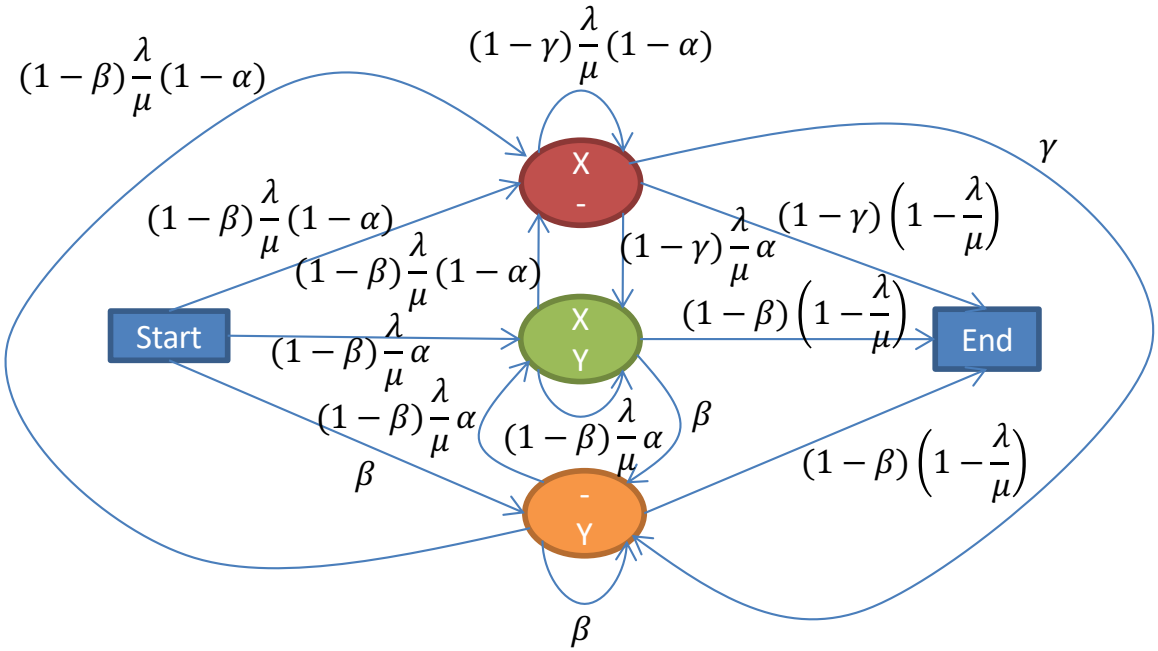


We can further collapse the small circle states by putting their effects in the link probabilities.

The collapsed pair-HMM



The collapsed pair-HMM



X-----XX
YYYYYYYY

Likelihood under the initial pair-HMM

$$\left(1(1-\beta)\frac{\lambda}{\mu}\right)(\alpha\cdot 1\cdot\beta)\left(\beta^4(1-\beta)\frac{\lambda}{\mu}\right)\left(\alpha\cdot 1\cdot(1-\beta)\frac{\lambda}{\mu}\right)\left(\alpha\cdot 1\cdot(1-\beta)\left(1-\frac{\lambda}{\mu}\right)\right)=$$

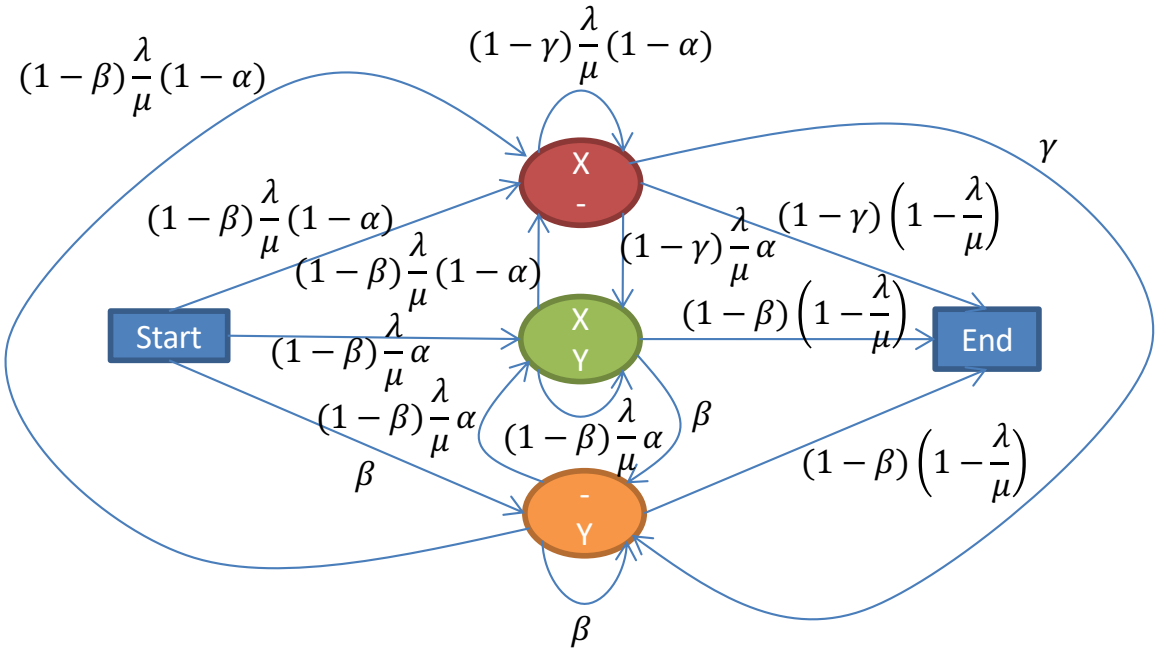
$$\alpha^3\beta^5(1-\beta)^4\left(\frac{\lambda}{\mu}\right)^3\left(1-\frac{\lambda}{\mu}\right)$$

Likelihood under the collapsed pair-HMM

$$\left((1-\beta)\frac{\lambda}{\mu}\alpha\right)\beta^5\left((1-\beta)\frac{\lambda}{\mu}\alpha\right)\left((1-\beta)\frac{\lambda}{\mu}\alpha\right)\left((1-\beta)\left(1-\frac{\lambda}{\mu}\right)\right)=$$

$$\alpha^3\beta^5(1-\beta)^4\left(\frac{\lambda}{\mu}\right)^3\left(1-\frac{\lambda}{\mu}\right)$$

The collapsed pair-HMM



Note:

• All transition probabilities depend only on the two parameters λ and μ, and the time t

• The ratio $\frac{\lambda}{\mu}$ controls the expected length of the sequence and the absolute value μt the amount of insertion/deletion

• In a more general model we can introduce more parameters to independently control:

- The number of insertion/deletions
- The average length of insertions/deletions
- The total sequence length.

Pair-HMM to represent sequence alignments

What we have seen so far is that we can use HMMs to represent alignments of sequences that are derived through an evolutionary process in which mutations, insertions and deletions occur.

The transition rates between HMM states can be derived from the rates of mutation, insertion and deletion of the evolutionary model.

What can we do with this pair-HMM?

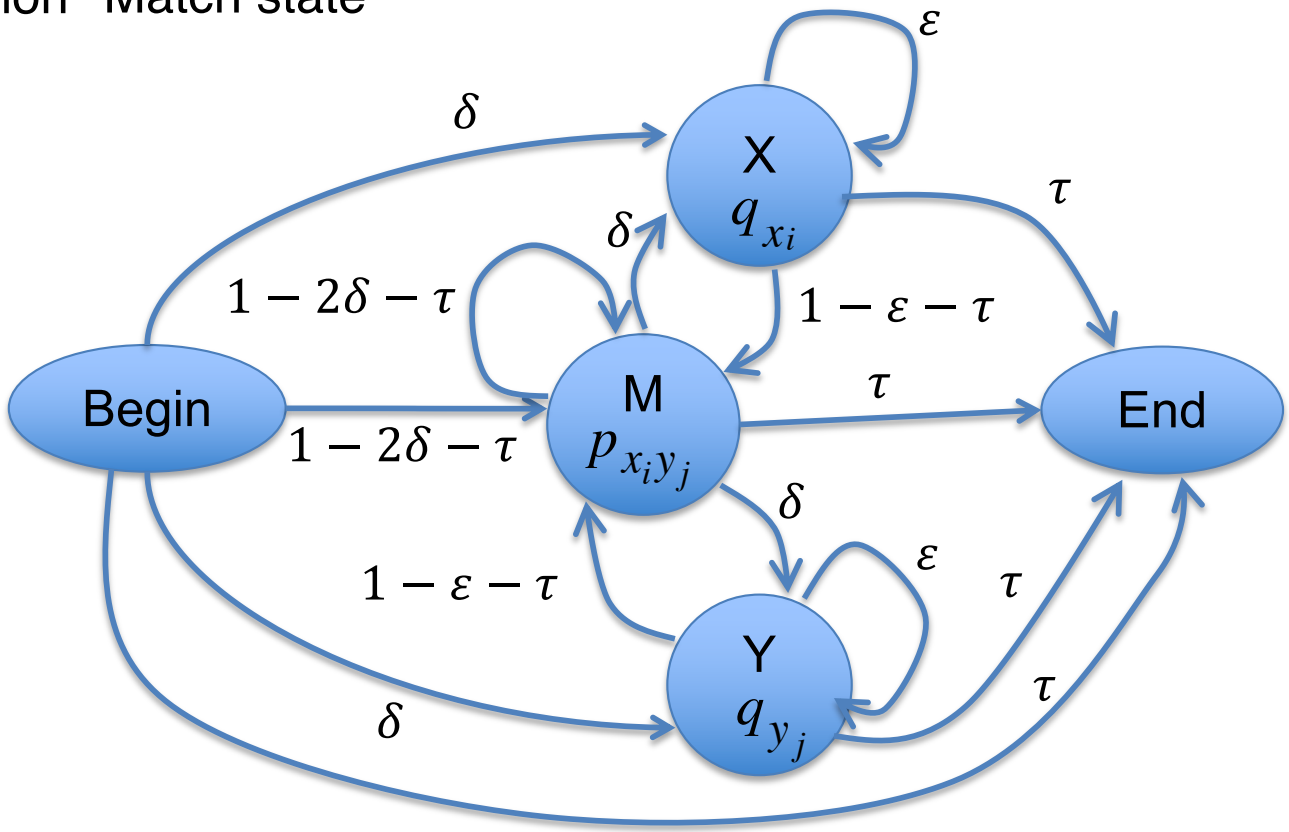
Pair-HMM to represent sequence alignments

Deletion

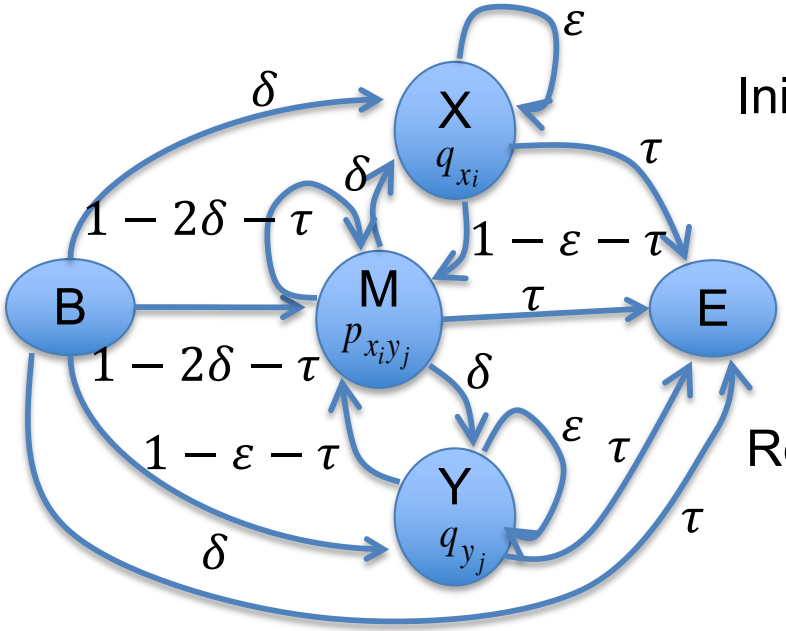
```
ATGGGTGCGAGA--GCGTCA
ATGGGT---CGAAGGCGT--
```

Insertion Match state

δ – open insertion/deletion block
ε – extend insertion/deletion block
τ – terminate alignment



Viterbi algorithm for most probable path



Initialization:

$v^M(0,0) = 1$ assuming starting from match state

$v^*(i,0) = v^*(0,j) = 0$ for all other $v^*(i,j)$

Recursion:

$$v^M(i,j) = p_{x_i,y_j} max \begin{cases} (1-2\delta-\tau)v^M(i-1,j-1) \\ (1-\varepsilon-\tau)v^X(i-1,j-1) \\ (1-\varepsilon-\tau)v^Y(i-1,j-1) \end{cases}$$

$$v^X(i,j) = q_{x_i} max \begin{cases} \delta v^M(i-1,j) \\ \varepsilon v^X(i-1,j) \end{cases}$$

$$v^Y(i,j) = q_{y_j} max \begin{cases} \delta v^M(i,j-1) \\ \varepsilon v^Y(i,j-1) \end{cases}$$

This score will depend on the length of the sequences.

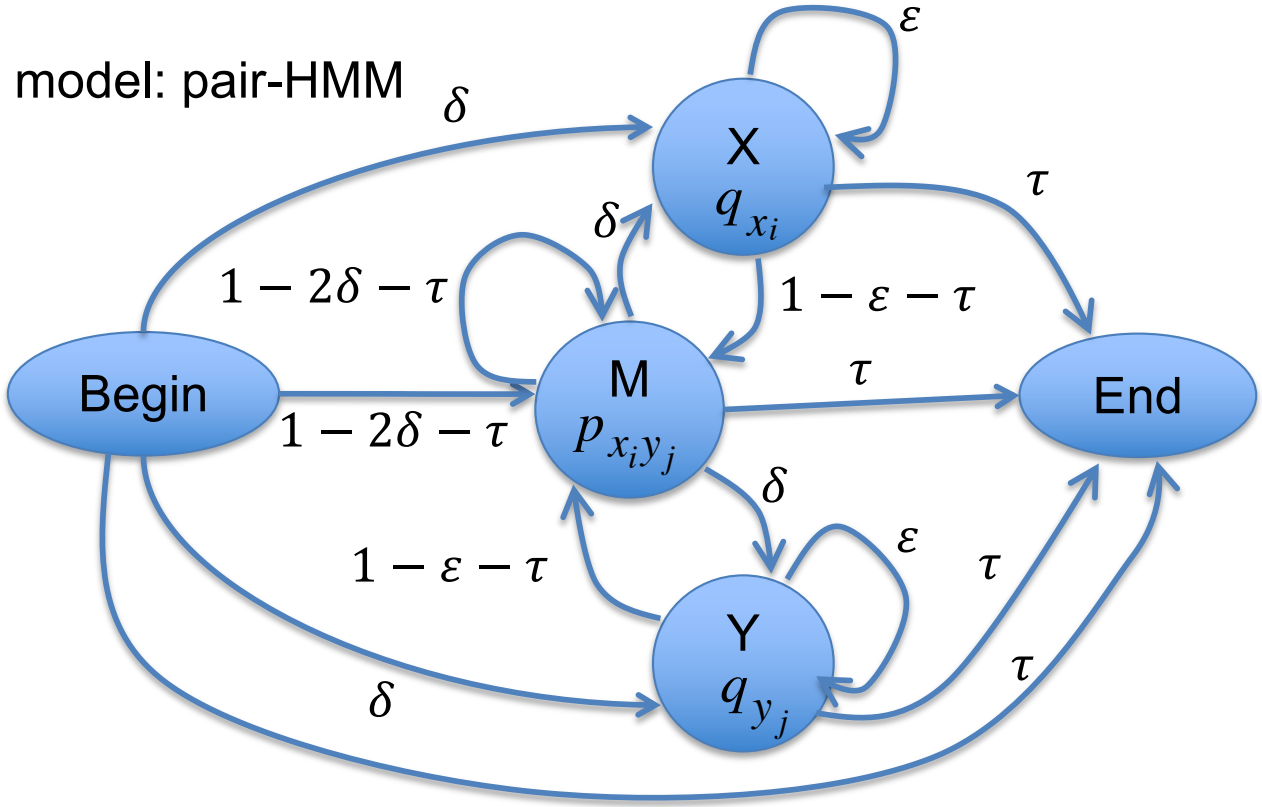
What if we want to compare sequences of different lengths?

Termination:

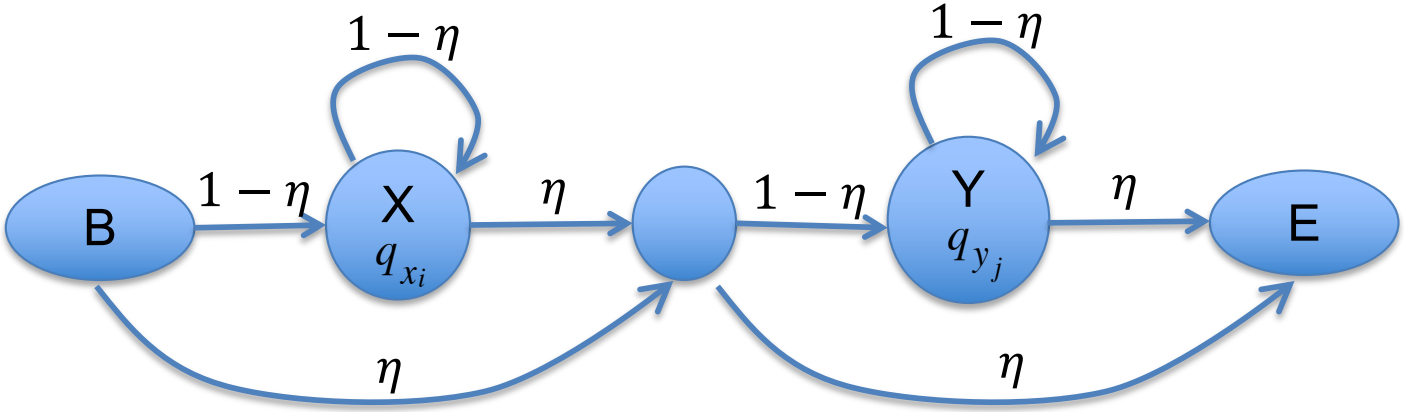
$$v^E = \tau\, max\big(v^M(n,m), v^X(n,m), v^Y(n,m)\big)$$

Most probable path with log-odds scores

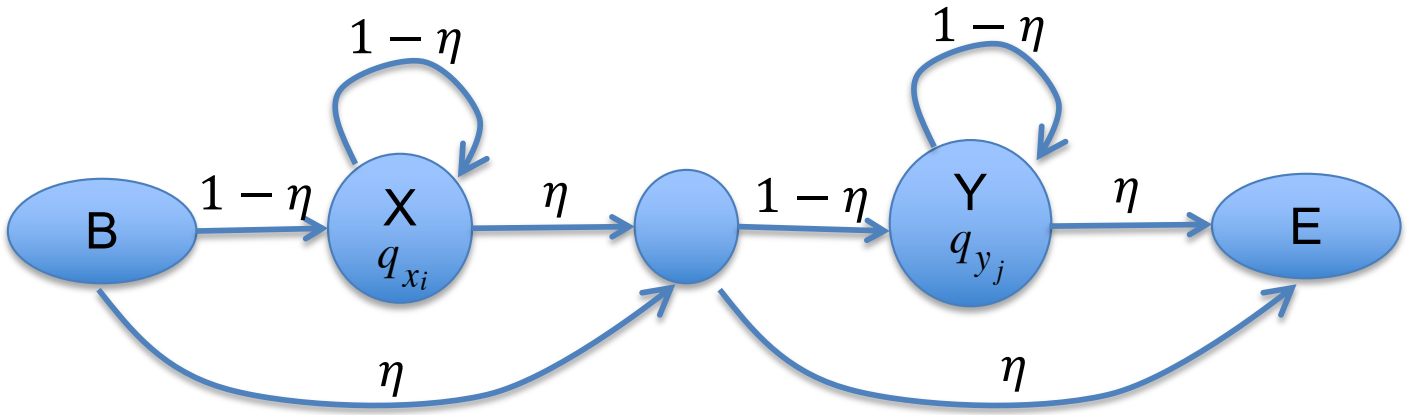
Foreground model: pair-HMM



Background model: sequences emitted independently of each other

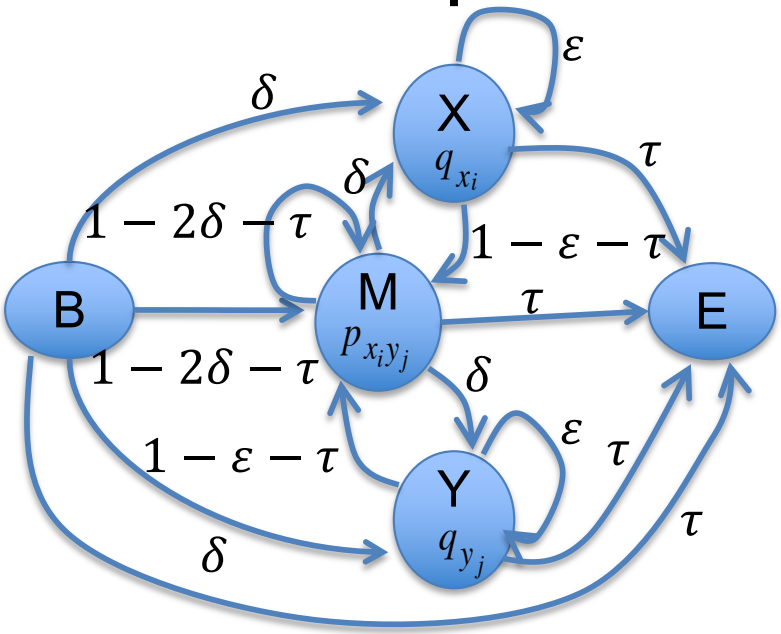


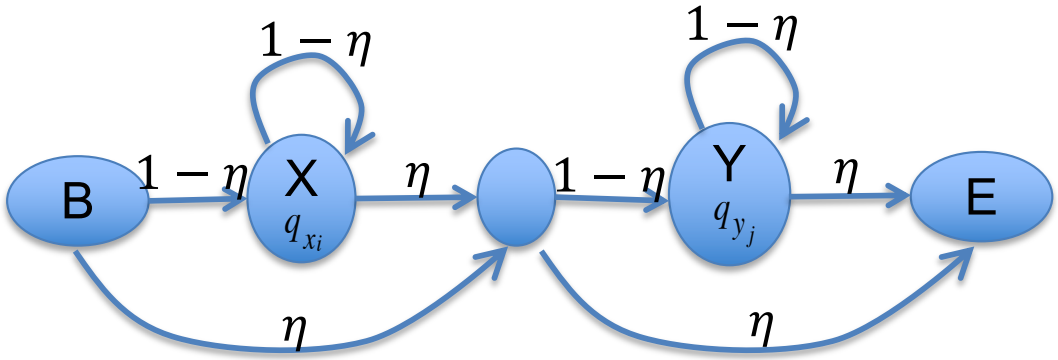
Most probable path with log-odds scores



$$P(x,y|R) = (1-\eta)^n \eta \prod_{i=1}^{n} q_{x_i} (1-\eta)^m \eta \prod_{j=1}^{m} q_{y_j} = \eta^2 (1-\eta)^{m+n} \prod_{i=1}^{n} q_{x_i} \prod_{j=1}^{m} q_{y_j}$$

Most probable path with odds scores





Recursion:

$$v^X(i,j) = \frac{q_{x_i}}{q_{x_i}} max \begin{cases} \frac{\delta}{(1-\eta)} v^M(i-1,j) \\ \frac{\varepsilon}{(1-\eta)} v^X(i-1,j) \end{cases}$$

$$v^M(i,j) = \frac{p_{x_i,y_j}}{q_{x_i} q_{y_j}} max \begin{cases} \frac{(1-2\delta-\tau)}{(1-\eta)^2} v^M(i-1,j-1) \\ \frac{(1-\varepsilon-\tau)}{(1-\eta)^2} v^X(i-1,j-1) \\ \frac{(1-\varepsilon-\tau)}{(1-\eta)^2} v^Y(i-1,j-1) \end{cases}$$

$$v^Y(i,j) = \frac{q_{y_j}}{q_{y_j}} max \begin{cases} \frac{\delta}{(1-\eta)} v^M(i,j-1) \\ \frac{\varepsilon}{(1-\eta)} v^Y(i,j-1) \end{cases}$$

The more common formulation

Recursion:

$$V^M(i,j) = s(x_i, y_j) + max \begin{cases} V^M(i-1, j-1) \\ V^X(i-1, j-1) \\ V^Y(i-1, j-1) \end{cases}$$

$$V^X(i,j) = max \begin{cases} V^M(i-1, j) - d \\ V^X(i-1, j) - e \end{cases}$$

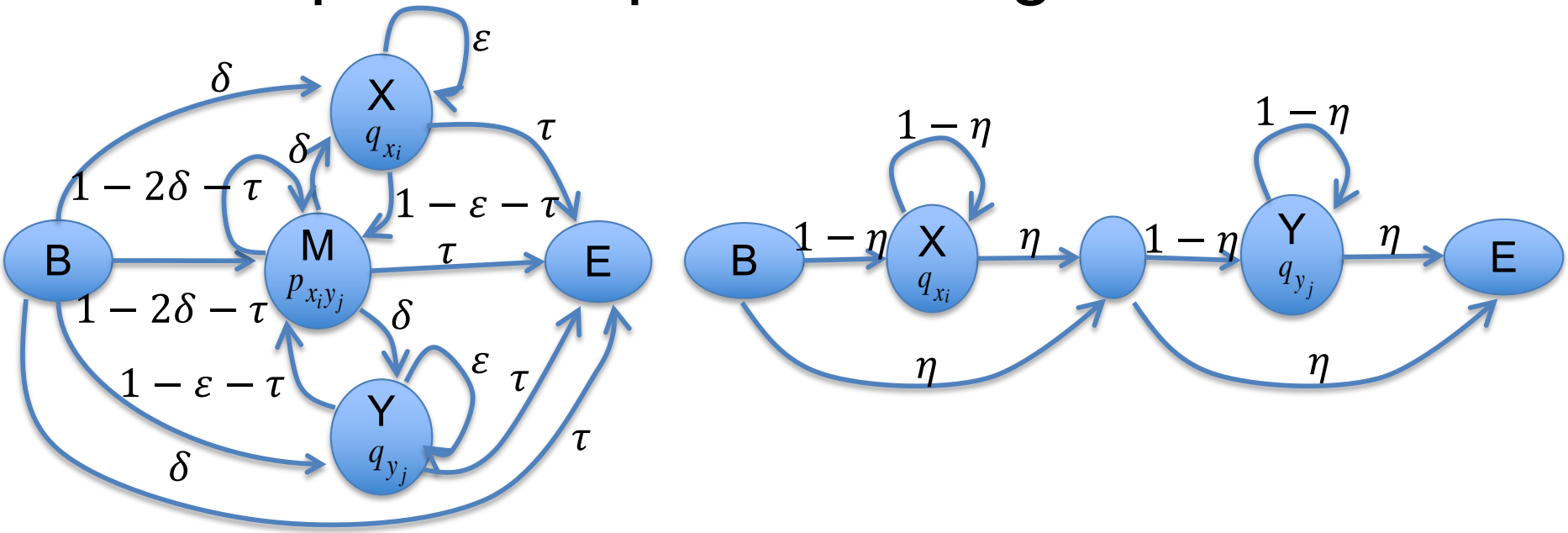
$$V^Y(i,j) = max \begin{cases} V^M(i, j-1) - d \\ V^Y(i, j-1) - e \end{cases}$$

Where the only parameters are:

- Scores for character-to-character alignment $s(x_i, y_j)$
- Gap opening penalty d
- Gap extension penalty e

Is there a mapping between the HMM with log-odds scores and these parameters?

Most probable path with log-odds scores



Likelihood ratio for match **after deletion/insertion**

$$\frac{p_{x_i y_j}(1-\varepsilon-\tau)}{q_{x_i} q_{y_j}(1-\eta)^2}$$

Likelihood ratio for match **after match**

$$\frac{p_{x_i y_j}(1-2\delta-\tau)}{q_{x_i} q_{y_j}(1-\eta)^2}$$

Likelihood ratio for deletion/insertion **after match**

$$\frac{q_{x_i/y_j}\delta}{q_{x_i/y_j}(1-\eta)} = \frac{\delta}{1-\eta}$$

Likelihood ratio for deletion/insertion **after deletion/insertion**

$$\frac{q_{x_i/y_j}\varepsilon}{q_{x_i/y_j}(1-\eta)} = \frac{\varepsilon}{1-\eta}$$

Most probable path with log-odds scores

Likelihood ratio for match **after deletion/insertion** $\frac{p_{x_i y_j}(1-\varepsilon-\tau)}{q_{x_i} q_{y_j}(1-\eta)^2}$

Likelihood ratio for match **after match** $\frac{p_{x_i y_j}(1-2\delta-\tau)}{q_{x_i} q_{y_j}(1-\eta)^2}$

Likelihood ratio for deletion/insertion **after match** $\frac{q_{x_i/y_j}\delta}{q_{x_i/y_j}(1-\eta)} = \frac{\delta}{1-\eta}$

Likelihood ratio for deletion/insertion **after deletion/insertion** $\frac{q_{x_i/y_j}\varepsilon}{q_{x_i/y_j}(1-\eta)} = \frac{\varepsilon}{1-\eta}$

Likelihood ratio for match->gap->match

Gap 'close' Gap 'open'

$$\frac{\delta}{1-\eta}\frac{p_{x_i y_j}(1-\varepsilon-\tau)}{q_{x_i} q_{y_j}(1-\eta)^2} = \frac{p_{x_i y_j}(1-2\delta-\tau)}{q_{x_i} q_{y_j}(1-\eta)^2}\frac{(1-\varepsilon-\tau)}{(1-2\delta-\tau)}\frac{\delta}{(1-\eta)}$$

Likelihood of match after match

Gap contribution

Most probable path with log-odds scores

Likelihood ratio for match **after deletion/insertion** $\frac{p_{x_i y_j}(1-\varepsilon-\tau)}{q_{x_i} q_{y_j}(1-\eta)^2}$

Likelihood ratio for match **after match** $\frac{p_{x_i y_j}(1-2\delta-\tau)}{q_{x_i} q_{y_j}(1-\eta)^2}$

Likelihood ratio for deletion/insertion **after match** $\frac{q_{x_i/y_j}\delta}{q_{x_i/y_j}(1-\eta)} = \frac{\delta}{1-\eta}$

Likelihood ratio for deletion/insertion **after deletion/insertion** $\frac{q_{x_i/y_j}\varepsilon}{q_{x_i/y_j}(1-\eta)} = \frac{\varepsilon}{1-\eta}$

Likelihood ratio for match->gap->match

$$\frac{\delta}{1-\eta}\frac{p_{x_i y_j}(1-\varepsilon-\tau)}{q_{x_i} q_{y_j}(1-\eta)^2} = \frac{p_{x_i y_j}(1-2\delta-\tau)}{q_{x_i} q_{y_j}(1-\eta)^2}\frac{(1-\varepsilon-\tau)}{(1-2\delta-\tau)}\frac{\delta}{(1-\eta)}\frac{\varepsilon}{1-\eta}$$

"Match score" "Gap opening score" "Gap extension score"

Most probable path with log-odds scores

Initialization:

$$V^M(0,0) = -2\log(\eta) \quad V^*(i,0) = V^*(0,j) = -\infty, \forall i,j$$

Recursion: $i = 1,\dots,n, j = 1,\dots,m$

$$V^M(i,j) = s(x_i, y_j) + max\begin{cases} V^M(i-1,j-1) \\ V^X(i-1,j-1) \\ V^Y(i-1,j-1) \end{cases}$$

$$V^X(i,j) = max\begin{cases} V^M(i-1,j) - d \\ V^X(i-1,j) - e \end{cases}$$

$$V^Y(i,j) = max\begin{cases} V^M(i,j-1) - d \\ V^Y(i,j-1) - e \end{cases}$$

When the alignment ended with a gap, we do not have the compensation from match after gap

$$c = log\left(\frac{(1-\varepsilon-\tau)}{(1-2\delta-\tau)}\right)$$

Termination: $V = max(V^M(n,m), V^X(n,m) - c, V^Y(n,m) - c)$

Defining score parameters for sequence alignments

For DNA: parameters derived from mechanistic models
(JC, Kimura, etc.)

For proteins: empirical parameters derived from protein sequence alignments.

Deriving score parameters from alignment data

Intuitive approach: compute character-character alignment, gap initiation and gap extension parameters from confirmed alignments

Difficulties with this approach:

1. Confirmed alignments are hard to come by

   -> use alignments of very closely related sequences, for which we can assume that a very small number of evolutionary changes occurred

2. The overall frequency of various events depends on the evolutionary distance

   -> use alignments generated from sequences that are separated by roughly the same evolutionary distance as the sequences that we will later want to align

Dayhoff matrices

Dayhoff PAM (point accepted mutations) matrices – introduced by Margaret Dayhoff in 1970's from 1572 observed mutations in 71 families of proteins, and updated in the 1990's.

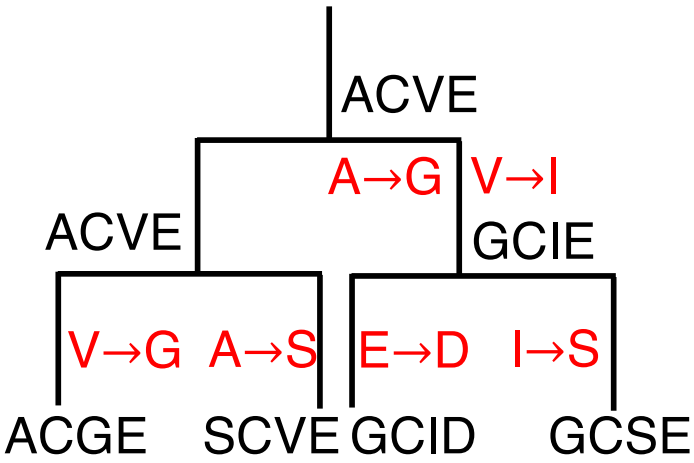
Proteins that in pairwise comparisons did not differ by more than 15% were used to construct maximum parsimony phylogenetic trees and to infer the mutations that occurred along the tree.

The number of substitutions from one amino acid a to another b, A_{ab}, and the number of occurrences of each amino acid that could have undergone mutations (depending on the amino acid frequency and the number of mutations in each branch) were counted. Assuming reversibility, changes were counted symmetrically.

The entries in the matrix were scaled so as to obtain 1 substitution in 100 amino acids. This 1 PAM matrix was defined as the substitution matrix that corresponds to an evolutionary time that yields an expected 1% of amino acids to undergo substitution.

Deriving the Dayhoff matrices

Let's assume a toy example, the following tree of sequences with the inferred substitutions that occurred along the branches:



Then we have the following parent-child sequence alignments:

```
ACVE  ACVE  ACVE  ACVE  GCIE  GCIE
ACVE  GCIE  ACGE  SCVE  GCID  GCSE
```

From which we can tabulate the number of occurrences of each amino acid and the number of mutations. We assume reversibility, so changes from *a* to *b* are also counted as changes from *b* to *a*:

Amino Acid	A	C	D	E	G	I	S	V
Occurrences	6	12	1	11	6	4	2	6
Changes	2	0	1	1	2	2	2	2
Mutability	0.33	0	1	0.09	0.33	0.5	1	0.33

Deriving the Dayhoff matrices

For the matrix to correspond to 1 substitution in 100 amino acids, we have to have $\sum_{i=1}^{20} p_i \lambda m_i = \frac{1}{100}$, where p_i is the frequency of amino acid i and m_i is its mutability.

From this we infer λ, and then the mutability matrix in which $M_{ii} = 1 - \lambda m_i$ is the proportional to the probability of amino acid i to stay unchanged, and $M_{ij} = \frac{\lambda m_i A_{ij}}{\sum_j A_{ij}}$ is proportional to the probability of amino acid i being substituted by j.

From the PAM 1 matrix (let's call it B) we can obtain the PAM matrix corresponding to an arbitrary number of evolutionary units by computing B^n

Finally, scores for the PAM$_n$ are derived as log likelihoods, q_b being the limit frequency of amino acid *b*

Deriving score parameters from alignment data

Henikoff BLOSUM matrices: derived from ungapped alignment regions of proteins that have a higher level of divergence.

Proteins are initially clustered whenever their percentage of identical residues exceeds some level L% and then only a representative is used per cluster.

Frequencies A_{ab}, representing the number of times residue a is paired with residue b are calculated, taking into account cluster size.

Then the probabilities of individual residues and pairs of residues are calculated as:

$$q_a = \frac{\sum_b A_{ab}}{\sum_{c,d} A_{cd}} \qquad p_{ab} = \frac{A_{ab}}{\sum_{c,d} A_{cd}}$$ and the score: $s(a,b) = log\left(\frac{p_{ab}}{q_a q_b}\right)$

Let’s put this into practice

Find the best global alignment of sequences S1 and S2

S1: HEAGAWGHEE
S2: PAWHEAE

assuming the BLOSUM50 substitution matrix and the gap cost per unaligned residue of -8 (identical score for gap initiation and gap extension).

Computation of the dynamic programming table

V(i,j)	i		H	E	A	G	A	W	G	H	E	E
j		0	1	2	3	4	5	6	7	8	9	10
	0											
P	1											
A	2											
W	3											
H	4											
E	5											
A	6											
E	7											

Computation of the dynamic programming table

V(i,j)	i		H	E	A	G	A	W	G	H	E	E
j		0	1	2	3	4	5	6	7	8	9	10
	0	0 ←	-8 ←	-16 ←	-24 ←	-32 ←	-40 ←	-48 ←	-56 ←	-64 ←	-72 ←	-80
P	1	-8 ↑										
A	2	-16 ↑										
W	3	-24 ↑										
H	4	-32 ↑										
E	5	-40 ↑										
A	6	-48 ↑										
E	7	-56 ↑										

Computation of the dynamic programming table

V(i,j)	i		H	E	A	G	A	W	G	H	E	E
j		0	1	2	3	4	5	6	7	8	9	10
	0	0	-8	-16	-24	-32	-40	-48	-56	-64	-72	-80
P	1	-8	-16									
A	2	-16										
W	3	-24										
H	4	-32										
E	5	-40										
A	6	-48										
E	7	-56										

Computation of the dynamic programming table

V(i,j)	i		H	E	A	G	A	W	G	H	E	E
j		0	1	2	3	4	5	6	7	8	9	10
	0	0	-8	-16	-24	-32	-40	-48	-56	-64	-72	-80
P	1	-8	-16									
A	2	-16										
W	3	-24										
H	4	-32										
E	5	-40										
A	6	-48										
E	7	-56										

Computation of the dynamic programming table

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V	B	Z	X	*
A	5	-2	-1	-2	-1	-1	-1	0	-2	-1	-2	-1	-1	-3	-1	1	0	-3	-2	0	-2	-1	-1	-5
R	-2	7	-1	-2	-4	1	0	-3	0	-4	-3	3	-2	-3	-3	-1	-1	-3	-1	-3	-1	0	-1	-5
N	-1	-1	7	2	-2	0	0	0	1	-3	-4	0	-2	-4	-2	1	0	-4	-2	-3	4	0	-1	-5
D	-2	-2	2	8	-4	0	2	-1	-1	-4	-4	-1	-4	-5	-1	0	-1	-5	-3	-4	5	1	-1	-5
C	-1	-4	-2	-4	13	-3	-3	-3	-3	-2	-2	-3	-2	-2	-4	-1	-1	-5	-3	-1	-3	-3	-2	-5
Q	-1	1	0	0	-3	7	2	-2	1	-3	-2	2	0	-4	-1	0	-1	-1	-1	-3	0	4	-1	-5
E	-1	0	0	2	-3	2	6	-3	0	-4	-3	1	-2	-3	-1	-1	-1	-3	-2	-3	1	5	-1	-5
G	0	-3	0	-1	-3	-2	-3	8	-2	-4	-4	-2	-3	-4	-2	0	-2	-3	-3	-4	-1	-2	-2	-5
H	-2	0	1	-1	-3	1	0	-2	10	-4	-3	0	-1	-1	-2	-1	-2	-3	2	-4	0	0	-1	-5
I	-1	-4	-3	-4	-2	-3	-4	-4	-4	5	2	-3	2	0	-3	-3	-1	-3	-1	4	-4	-3	-1	-5
L	-2	-3	-4	-4	-2	-2	-3	-4	-3	2	5	-3	3	1	-4	-3	-1	-2	-1	1	-4	-3	-1	-5
K	-1	3	0	-1	-3	2	1	-2	0	-3	-3	6	-2	-4	-1	0	-1	-3	-2	-3	0	1	-1	-5
M	-1	-2	-2	-4	-2	0	-2	-3	-1	2	3	-2	7	0	-3	-2	-1	-1	0	1	-3	-1	-1	-5
F	-3	-3	-4	-5	-2	-4	-3	-4	-1	0	1	-4	0	8	-4	-3	-2	1	4	-1	-4	-4	-2	-5
P	-1	-3	-2	-1	-4	-1	-1	-2	-2	-3	-4	-1	-3	-4	10	-1	-1	-4	-3	-3	-2	-1	-2	-5
S	1	-1	1	0	-1	0	-1	0	-1	-3	-3	0	-2	-3	-1	5	2	-4	-2	-2	0	0	-1	-5
T	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	2	5	-3	-2	0	0	-1	0	-5
W	-3	-3	-4	-5	-5	-1	-3	-3	-3	-3	-2	-3	-1	1	-4	-4	-3	15	2	-3	-5	-2	-3	-5
Y	-2	-1	-2	-3	-3	-1	-2	-3	2	-1	-1	-2	0	4	-3	-2	-2	2	8	-1	-3	-2	-1	-5
V	0	-3	-3	-4	-1	-3	-3	-4	-4	4	1	-3	1	-1	-3	-2	0	-3	-1	5	-4	-3	-1	-5
B	-2	-1	4	5	-3	0	1	-1	0	-4	-4	0	-3	-4	-2	0	0	-5	-3	-4	5	2	-1	-5
Z	-1	0	0	1	-3	4	5	-2	0	-3	-3	1	-1	-4	-1	0	-1	-2	-2	-3	2	5	-1	-5
X	-1	-1	-1	-1	-2	-1	-1	-2	-1	-1	-1	-1	-1	-2	-2	-1	0	-3	-1	-1	-1	-1	-1	-5
*	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	1

Computation of the dynamic programming table

V(i,j)	i		H	E	A	G	A	W	G	H	E	E
j		0	1	2	3	4	5	6	7	8	9	10
	0	0	-8	-16	-24	-32	-40	-48	-56	-64	-72	-80
P	1	-8	-2									
A	2	-16										
W	3	-24										
H	4	-32										
E	5	-40										
A	6	-48										
E	7	-56										

Computation of the dynamic programming table

V(i,j)	i		H	E	A	G	A	W	G	H	E	E
j		0	1	2	3	4	5	6	7	8	9	10
	0	0	-8	-16	-24	-32	-40	-48	-56	-64	-72	-80
P	1	-8	-2	-24								
A	2	-16										
W	3	-24										
H	4	-32										
E	5	-40										
A	6	-48										
E	7	-56										

Computation of the dynamic programming table

V(i,j)	i		H	E	A	G	A	W	G	H	E	E
j		0	1	2	3	4	5	6	7	8	9	10
	0	0	-8	-16	-24	-32	-40	-48	-56	-64	-72	-80
P	1	-8	-2	-10								
A	2	-16										
W	3	-24										
H	4	-32										
E	5	-40										
A	6	-48										
E	7	-56										

Computation of the dynamic programming table

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V	B	Z	X	*
A	5	-2	-1	-2	-1	-1	-1	0	-2	-1	-2	-1	-1	-3	-1	1	0	-3	-2	0	-2	-1	-1	-5
R	-2	7	-1	-2	-4	1	0	-3	0	-4	-3	3	-2	-3	-3	-1	-1	-3	-1	-3	-1	0	-1	-5
N	-1	-1	7	2	-2	0	0	0	1	-3	-4	0	-2	-4	-2	1	0	-4	-2	-3	4	0	-1	-5
D	-2	-2	2	8	-4	0	2	-1	-1	-4	-4	-1	-4	-5	-1	0	-1	-5	-3	-4	5	1	-1	-5
C	-1	-4	-2	-4	13	-3	-3	-3	-3	-2	-2	-3	-2	-2	-4	-1	-1	-5	-3	-1	-3	-3	-2	-5
Q	-1	1	0	0	-3	7	2	-2	1	-3	-2	2	0	-4	-1	0	-1	-1	-1	-3	0	4	-1	-5
E	-1	0	0	2	-3	2	6	-3	0	-4	-3	1	-2	-3	-1	-1	-1	-3	-2	-3	1	5	-1	-5
G	0	-3	0	-1	-3	-2	-3	8	-2	-4	-4	-2	-3	-4	-2	0	-2	-3	-3	-4	-1	-2	-2	-5
H	-2	0	1	-1	-3	1	0	-2	10	-4	-3	0	-1	-1	-2	-1	-2	-3	2	-4	0	0	-1	-5
I	-1	-4	-3	-4	-2	-3	-4	-4	-4	5	2	-3	2	0	-3	-3	-1	-3	-1	4	-4	-3	-1	-5
L	-2	-3	-4	-4	-2	-2	-3	-4	-3	2	5	-3	3	1	-4	-3	-1	-2	-1	1	-4	-3	-1	-5
K	-1	3	0	-1	-3	2	1	-2	0	-3	-3	6	-2	-4	-1	0	-1	-3	-2	-3	0	1	-1	-5
M	-1	-2	-2	-4	-2	0	-2	-3	-1	2	3	-2	7	0	-3	-2	-1	-1	0	1	-3	-1	-1	-5
F	-3	-3	-4	-5	-2	-4	-3	-4	-1	0	1	-4	0	8	-4	-3	-2	1	4	-1	-4	-4	-2	-5
P	-1	-3	-2	-1	-4	-1	-1	-2	-2	-3	-4	-1	-3	-4	10	-1	-1	-4	-3	-3	-2	-1	-2	-5
S	1	-1	1	0	-1	0	-1	0	-1	-3	-3	0	-2	-3	-1	5	2	-4	-2	-2	0	0	-1	-5
T	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	2	5	-3	-2	0	0	-1	0	-5
W	-3	-3	-4	-5	-5	-1	-3	-3	-3	-3	-2	-3	-1	1	-4	-4	-3	15	2	-3	-5	-2	-3	-5
Y	-2	-1	-2	-3	-3	-1	-2	-3	2	-1	-1	-2	0	4	-3	-2	-2	2	8	-1	-3	-2	-1	-5
V	0	-3	-3	-4	-1	-3	-3	-4	-4	4	1	-3	1	-1	-3	-2	0	-3	-1	5	-4	-3	-1	-5
B	-2	-1	4	5	-3	0	1	-1	0	-4	-4	0	-3	-4	-2	0	0	-5	-3	-4	5	2	-1	-5
Z	-1	0	0	1	-3	4	5	-2	0	-3	-3	1	-1	-4	-1	0	-1	-2	-2	-3	2	5	-1	-5
X	-1	-1	-1	-1	-2	-1	-1	-2	-1	-1	-1	-1	-1	-2	-2	-1	0	-3	-1	-1	-1	-1	-1	-5
*	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	1

Computation of the dynamic programming table

V(i,j)	i		H	E	A	G	A	W	G	H	E	E
j		0	1	2	3	4	5	6	7	8	9	10
	0	0	-8	-16	-24	-32	-40	-48	-56	-64	-72	-80
P	1	-8	-2	-9								
A	2	-16										
W	3	-24										
H	4	-32										
E	5	-40										
A	6	-48										
E	7	-56										

Computation of the dynamic programming table

V(i,j)	i		H	E	A	G	A	W	G	H	E	E
j		0	1	2	3	4	5	6	7	8	9	10
	0	0	-8	-16	-24	-32	-40	-48	-56	-64	-72	-80
P	1	-8	-2	-9	-17	-25	-33	-42	-49	-57	-65	-73
A	2	-16	-10	-3	-4	-12	-20	-28	-36	-44	-52	-60
W	3	-24	-18	-11	-6	-7	-15	-5	-13	-21	-29	-37
H	4	-32	-14	-18	-13	-8	-9	-13	-7	-3	-11	-19
E	5	-40	-22	-8	-16	-16	-9	-12	-15	-7	3	-5
A	6	-48	-30	-16	-3	-11	-11	-12	-12	-15	-5	2
E	7	-56	-38	-24	-11	-6	-12	-14	-15	-12	-9	1

Computation of the dynamic programming table

V(i,j)	i		H	E	A	G	A	W	G	H	E	E
j		0	1	2	3	4	5	6	7	8	9	10
	0	0	-8	-16	-24	-32	-40	-48	-56	-64	-72	-80
P	1	-8	-2	-9	-17	-25	-33	-42	-49	-57	-65	-73
A	2	-16	-10	-3	-4	-12	-20	-28	-36	-44	-52	-60
W	3	-24	-18	-11	-6	-7	-15	-5	-13	-21	-29	-37
H	4	-32	-14	-18	-13	-8	-9	-13	-7	-3	-11	-19
E	5	-40	-22	-8	-16	-16	-9	-12	-15	-7	3	-5
A	6	-48	-30	-16	-3	-11	-11	-12	-12	-15	-5	2
E	7	-56	-38	-24	-11	-6	-12	-14	-15	-12	-9	1

Computation of the dynamic programming table

V(i,j)	i		H	E	A	G	A	W	G	H	E	E
j		0	1	2	3	4	5	6	7	8	9	10
	0	0	-8	-16	-24	-32	-40	-48	-56	-64	-72	-80
P	1	-8	-2	-9	-17	-25	-33	-42	-49	-57	-65	-73
A	2	-16	-10	-3	-4	-12	-20	-28	-36	-44	-52	-60
W	3	-24	-18	-11	-6	-7	-15	-5	-13	-21	-29	-37
H	4	-32	-14	-18	-13	-8	-9	-13	-7	-3	-11	-19
E	5	-40	-22	-8	-16	-16	-9	-12	-15	-7	3	-5
A	6	-48	-30	-16	-3	-11	-11	-12	-12	-15	-5	2
E	7	-56	-38	-24	-11	-6	-12	-14	-15	-12	-9	1

Computation of the dynamic programming table

V(i,j)	i		H	E	A	G	A	W	G	H	E	E
j		0	1	2	3	4	5	6	7	8	9	10
	0	0	-8	-16	-24	-32	-40	-48	-56	-64	-72	-80
P	1	-8	-2	-9	-17	-25	-33	-42	-49	-57	-65	-73
A	2	-16	-10	-3	-4	-12	-20	-28	-36	-44	-52	-60
W	3	-24	-18	-11	-6	-7	-15	-5	-13	-21	-29	-37
H	4	-32	-14	-18	-13	-8	-9	-13	-7	-3	-11	-19
E	5	-40	-22	-8	-16	-16	-9	-12	-15	-7	3	-5
A	6	-48	-30	-16	-3	-11	-11	-12	-12	-15	-5	2
E	7	-56	-38	-24	-11	-6	-12	-14	-15	-12	-9	1

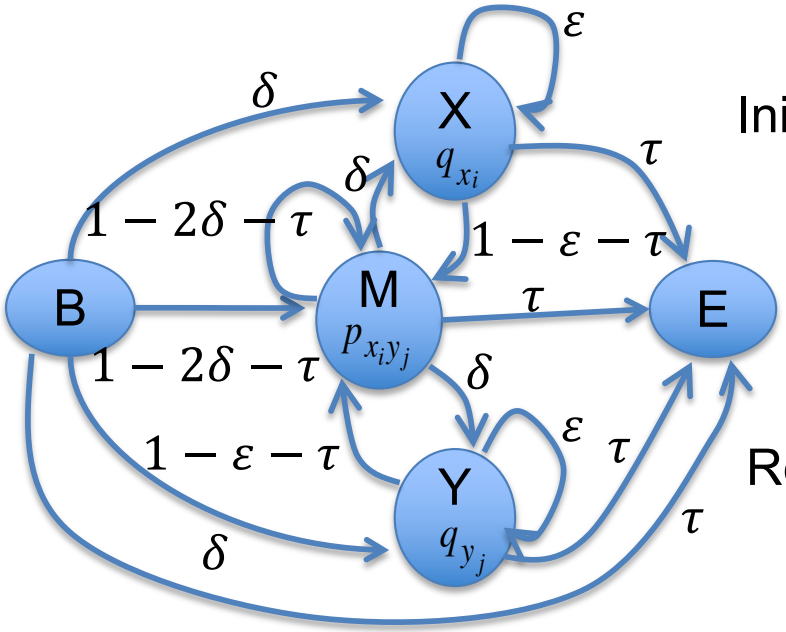
Computation of the dynamic programming table

V(i,j)	i		H	E	A	G	A	W	G	H	E	E
j		0	1	2	3	4	5	6	7	8	9	10
	0	0	-8	-16	-24	-32	-40	-48	-56	-64	-72	-80
P	1	-8	-2	-9	-17	-25	-33	-42	-49	-57	-65	-73
A	2	-16	-10	-3	-4	-12	-20	-28	-36	-44	-52	-60
W	3	-24	-18	-11	-6	-7	-15	-5	-13	-21	-29	-37
H	4	-32	-14	-18	-13	-8	-9	-13	-7	-3	-11	-19
E	5	-40	-22	-8	-16	-16	-9	-12	-15	-7	3	-5
A	6	-48	-30	-16	-3	-11	-11	-12	-12	-15	-5	2
E	7	-56	-38	-24	-11	-6	-12	-14	-15	-12	-9	1

HEAGAWGHE-E

--P-AW-HEAE

Summing over paths. Probability of x and y



Initialization:

$$f^M(0,0) = 1, f^X(0,0) = 0, f^Y(0,0) = 0$$

$$f^*(i,0) = f^*(0,j) = 0 \text{ for all other } v^*(i,j)$$

Recursion: $i = 0, \ldots, n$ $j = 0, \ldots, m$ except (0,0):

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

$$f^X(i,j) = q_{x_i}[\delta f^M(i-1,j) + \varepsilon f^X(i-1,j)]$$

$$f^Y(i,j) = q_{y_j}[\delta f^M(i,j-1) + \varepsilon f^Y(i,j-1)]$$

Termination:

$$f^E = \tau\,[f^M(n,m) + f^X(n,m) + f^Y(n,m)]$$

Summing over paths. Probability of x and y

Applications: Probabilistic sampling of alignments

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

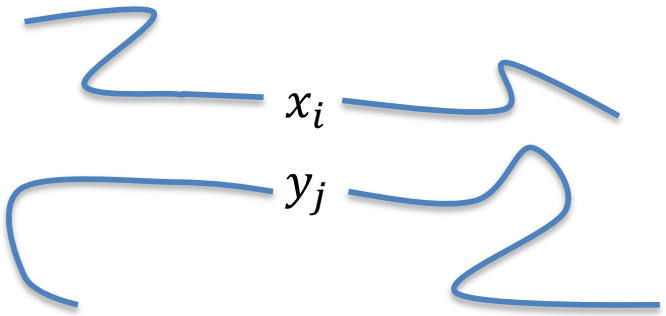
Etc.

Summing over paths. Probability of x and y

Applications: Defining posterior distributions over alignments given sequences x and y

$$P(\pi|x,y) = \frac{P(x,y,\pi)}{P(x,y)}$$

We can compute posterior probabilities of specific characters being aligned with each other by summing over alignments that share this character-to-character alignment vs. all alignments, without any constraint.



Pairwise alignment variations: local alignment

Smith-Waterman alignment

$$V(i,0) = V(0,j) = 0 \; \forall i,j$$

$$V(i,j) = max \begin{cases} 0 \\ V(i-1,j)+\sigma(x_i,-) \\ V(i,j-1)+\sigma(-,y_j) \\ V(i-1,j-1)+\sigma(x_i,y_j) \end{cases}$$

Computation of the dynamic programming table

V(i,j)	i		H	E	A	G	A	W	G	H	E	E
j		0	1	2	3	4	5	6	7	8	9	10
	0											
P	1											
A	2											
W	3											
H	4											
E	5											
A	6											
E	7											

Computation of the dynamic programming table

V(i,j)	i		H	E	A	G	A	W	G	H	E	E
j		0	1	2	3	4	5	6	7	8	9	10
	0	0	0	0	0	0	0	0	0	0	0	0
P	1	0										
A	2	0										
W	3	0										
H	4	0										
E	5	0										
A	6	0										
E	7	0										

Computation of the dynamic programming table

V(i,j)	i		H	E	A	G	A	W	G	H	E	E
j		0	1	2	3	4	5	6	7	8	9	10
	0	0	0	0	0	0	0	0	0	0	0	0
P	1	0	-8									
A	2	0										
W	3	0										
H	4	0										
E	5	0										
A	6	0										
E	7	0										

Computation of the dynamic programming table

V(i,j)	i		H	E	A	G	A	W	G	H	E	E
j		0	1	2	3	4	5	6	7	8	9	10
	0	0	0	0	0	0	0	0	0	0	0	0
P	1	0 ←	-8									
A	2	0										
W	3	0										
H	4	0										
E	5	0										
A	6	0										
E	7	0										

Computation of the dynamic programming table

V(i,j)	i		H	E	A	G	A	W	G	H	E	E
j		0	1	2	3	4	5	6	7	8	9	10
	0	0	0	0	0	0	0	0	0	0	0	0
P	1	0	-2									
A	2	0										
W	3	0										
H	4	0										
E	5	0										
A	6	0										
E	7	0										

Computation of the dynamic programming table

V(i,j)	i		H	E	A	G	A	W	G	H	E	E
j		0	1	2	3	4	5	6	7	8	9	10
	0	0	0	0	0	0	0	0	0	0	0	0
P	1	0	0	0	0	0	0	0	0	0	0	0
A	2	0	0	0	5	0	5	0	0	0	0	0
W	3	0	0	0	0	0	0	20	12	0	0	0
H	4	0	10	0	0	0	0	12	18	22	14	6
E	5	0	0	16	8	0	0	4	9	18	28	20
A	6	0	0	8	21	13	5	0	4	7	20	27
E	7	0	0	6	13	18	12	4	0	4	13	26

best local alignment:

```
AWGHE
AW-HE
```

Computation of the dynamic programming table

V(i,j)	i		H	E	A	G	A	W	G	H	E	G
j		0	1	2	3	4	5	6	7	8	9	10
	0	0	0	0	0	0	0	0	0	0	0	0
P	1	0	0	0	0	0	0	0	0	0	0	0
A	2	0	0	0	5	0	5	0	0	0	0	0
W	3	0	0	0	0	0	0	20	12	0	0	0
H	4	0	10	0	0	0	0	12	18	22	14	6
E	5	0	0	16	8	0	0	4	9	18	28	20
A	6	0	0	8	21	13	5	0	4	7	20	28
E	7	0	0	6	13	18	12	4	0	4	13	20

Computation of the dynamic programming table

V(i,j)	i		H	E	A	G	A	W	G	H	E	G
j		0	1	2	3	4	5	6	7	8	9	10
	0	0	0	0	0	0	0	0	0	0	0	0
P	1	0	0	0	0	0	0	0	0	0	0	0
A	2	0	0	0	5	0	5	0	0	0	0	0
W	3	0	0	0	0	0	0	20	12	0	0	0
H	4	0	10	0	0	0	0	12	18	22	14	6
E	5	0	0	16	8	0	0	4	9	18	28	20
A	6	0	0	8	21	13	5	0	4	7	20	28
E	7	0	0	6	13	18	12	4	0	4	13	20

Multiple equivalent alignments

V(i,j)	i		H	E	A	G	A	W	G	H	E	G
j		0	1	2	3	4	5	6	7	8	9	10
	0	0	0	0	0	0	0	0	0	0	0	0
P	1	0	0	0	0	0	0	0	0	0	0	0
A	2	0	0	0	5	0	5	0	0	0	0	0
W	3	0	0	0	0	0	0	20	12	0	0	0
H	4	0	10	0	0	0	0	12	18	22	14	6
E	5	0	0	16	8	0	0	4	9	18	28	20
A	6	0	0	8	21	13	5	0	4	7	20	28
E	7	0	0	6	13	18	12	4	0	4	13	20

AWGHE
AW-HE

AWGHEG
AW-HEA