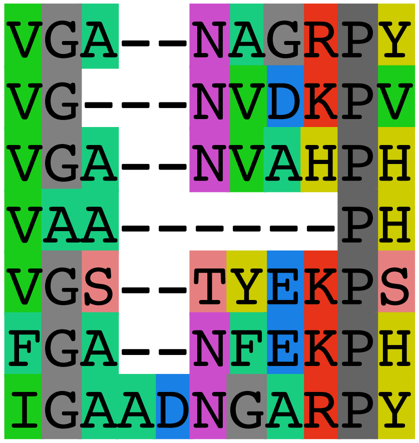
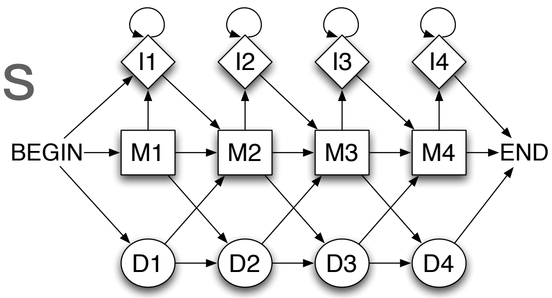
Advanced Database Searching: Sequence Patterns, Profiles & Hidden Markov Models

BI 527, Lecture #14, Fall 2011



```
VGA--NAGRPY
VG---NVDKPV
VGA--NVAHPH
VAA------PH
VGS--TYEKPS
FGA--NFEKPH
IGAADNGARPY
```

Barry Grant
2055A Palmer Commons
Tel: 647-3113
bjgrant@umich.edu
http://thegrantlab.org



Recap of the last lecture: #13

- Sequence motifs and **patterns**
  Finding functional cues from sequence conservation
  Patterns that describe a motif using a qualitative regular expression sequence
    - [LFI]-x-G-[PT]-P-G-x-G-K-[TS]-[AGSI]

  Defining and using patterns and their limitations

- Sequence **profiles**
  Profiles describe a motif using quantitative information captured in a **PSSM**
  Building log-likelihood ratio PSSMs
  The average score method for protein PSSMs
  Scoring sequences and searching with profiles

- **PSI-BLAST** algorithm
  Iterative PSSM searching to improve BLAST search sensitivity
  PSSM advantages and limitations
  The danger of PSSM corruption (triangular inequality)

- Profile software and databases

Outline of this lecture: #14

- Major **PSSM limitations**
  Do not capture positional dependencies
  Hard to recognize pattern instances that contain indels
  Do not handle boundary detection problems well

- Modeling motifs using **Markov chains**:
  Pros and cons of Markov models

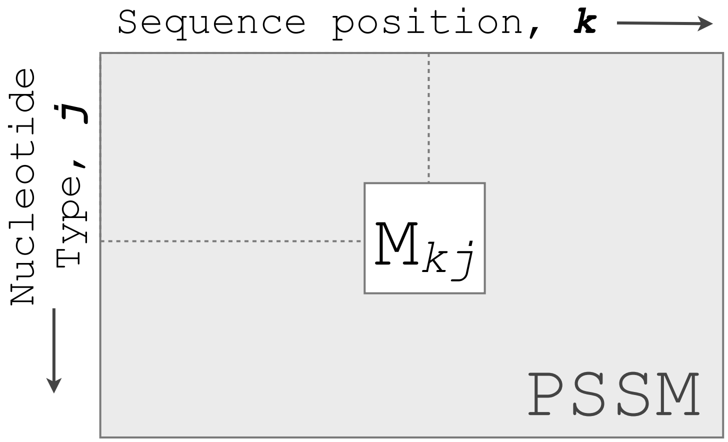
- **Hidden Markov models** (**HMMs**)
  More versatile full probabilistic model for detection of remote similarities
  Architecture and parameterization
  Boundary detection
  Key algorithms: Viterbi, Forward and Baum-Welch algorithms
  Scoring sequences and generating MSAs
  HMM limitations

- HMM software and databases
  Summary and example usage

Position Specific Scoring Matrices

A sequence profile is a **position-specific scoring matrix** (or **PSSM**) that gives a *quantitative* description of a sequence motif.

A simple PSSM is a Log odds scoring matrix that has as many columns as there are positions in the alignment, and either 4 rows (one for each DNA nucleotide) or 20 rows (one for each amino acid).



$$M_{kj} = \log\left(\frac{p_{kj}}{p_j}\right)$$

M_{kj} score for the *j*th nucleotide at position *k*

p_{kj} probability of nucleotide *j* at position *k*

p_j "background" probability of nucleotide *j*

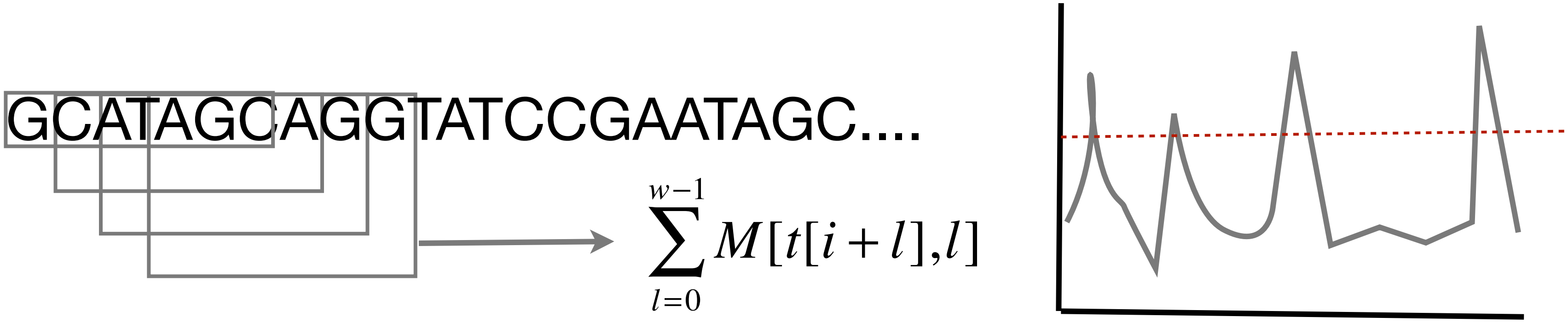
See Lecture #13

Problems with PSSMs

PSSMs work well for fixed length motifs in which the sites are more or less independent - i.e., ungapped motifs

However there are other kinds of motifs for which PSSMs are not well suited. PSSMs cannot:

1. model positional dependencies
2. recognize pattern instances containing insertions or deletions
3. model variable length patterns
4. detect boundaries



GCATAGCAGGTATCCGAATAGC....

$$\sum_{l=0}^{w-1} M[t[i+l],l]$$

Problems with PSSMs: 1. Positional dependencies

Do not capture positional dependencies

WEIRD
WEIRD
WEIQH
WEIRD
WEIQH

D					0.6
E		1.0			
H					0.4
I			1.0		
Q				0.4	
R				0.6	
W	1.0				

Note: We never see **QD** or **RH**, we only see **RD** and **QH**.
However, P(RH)=0.24, P(QD)=0.24, while P(QH)=0.16

Problems with PSSMs: 2. Insertions and deletions

Hard to recognize pattern instances that contain indels

D	0.8	0.8	0.8	0.8	2.4
E	0.6	2.9	0.6	0.6	1.6
H	2.0	2.0	2.0	2.0	3.0
I	0.8	0.8	3.1	0.8	0.8
Q	1.1	1.1	1.1	2.1	1.1
R	0.8	0.8	0.8	2.8	0.8
W	5.0	2.7	2.7	2.7	1.8

$\boxed{\texttt{WETIR}}\texttt{D}$

5.0 + 2.9 + 1.2 + 1.4 + 1.5 = 11

$\texttt{W}\boxed{\texttt{ETIRD}}$

1.2 + 1.8 + 3.1 + 3.0 + 3.4 = 12.5

$\boxed{\texttt{WE}}\texttt{T}\boxed{\texttt{IRD}}$

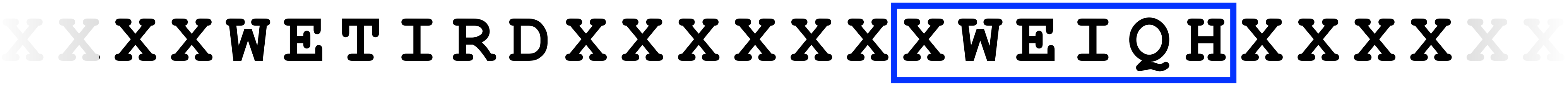
5.0 + 2.9 + 3.1 + 3.0 + 3.4= 18.4

Problems with PSSMs: 3. Variable length motifs

Cannot easily deal with variable length motifs

```
WETIRD
WE-IRD
WETIQH
WE-IRD
WETIQH
```

Gaps can be represented by expanding ∑ but what size window should be used to score new instances of the motif???


XXXXWETIRDXXXXXXXWEIQHXXXXXX

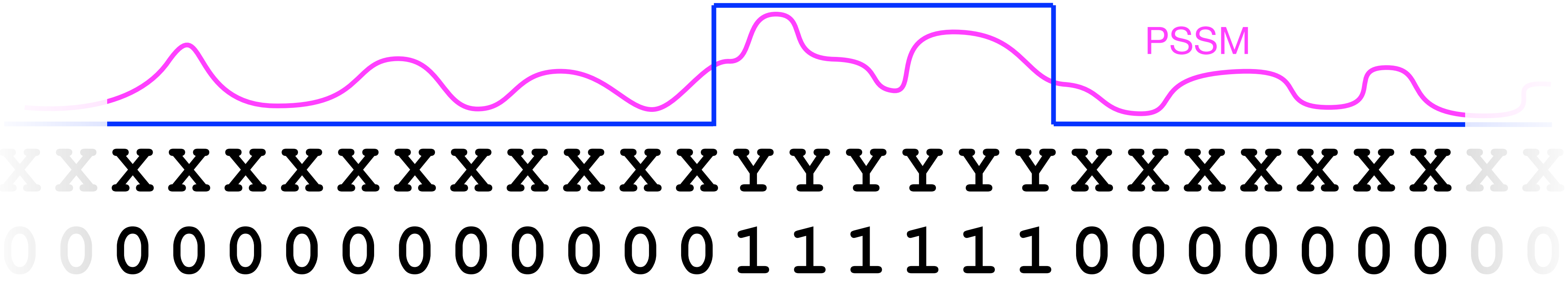
Problems with PSSMs: 4. Detecting boundaries

Do not handle boundary detection problems well

E.g. Label every element in the sequence with a 0 (not in pattern) or a 1 (in pattern)

Examples of boundary detection problems include:

- Recognition of regulatory motifs
- Recognition of protein domains
- Intron/exon boundaries
- Gene boundaries
- Transmembrane regions
- Secondary structure elements (helices and strands)

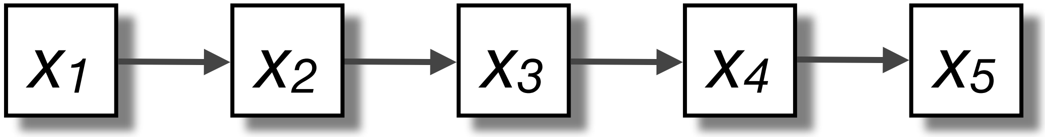


These shortcomings of PSSMs set the stage for a new kind of profile, based on **Markov chains**, called **Hidden Markov models** (**HMMs**)

- modeling positional dependencies
- recognizing pattern instances with indels
- modeling variable length patterns
- detecting boundaries

Markov chains

Markov chains are stochastic processes that undergo **transitions** between a finite series of **states** in a chainlike manner.



The system transverses states with probability

$$p(x_1, x_2, x_3, \ldots) = p(x_1)\, p(x_2 | x_1)\, p(x_3 | x_2)\, p(x_4 | x_3) \ldots$$

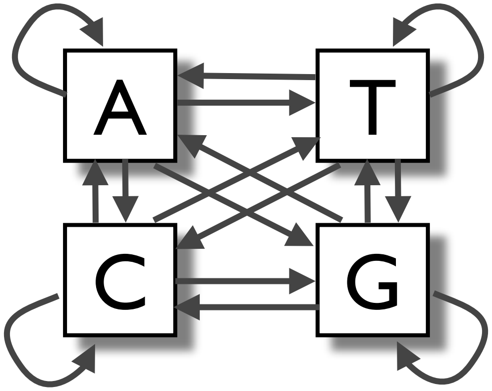
i.e. **Markov chains are memoryless**: the probability that the chain is in state x_i at time t, depends only on the state at the previous time step and not on the past history of the states visited before time $t-1$.

This specific kind of "*memorylessness*" is called the **Markov property**.

The **Markov property** states that the conditional probability distribution for the system at the next step (and in fact at all future steps) depends only on the current state of the system, and not additionally on the state of the system at previous steps.

Markov chains...

Markov chains, and their extension hidden Markov models (HMMs), are commonly represented by **state diagrams**, which consist of *states* and connecting *transitions*



E.g., A general Markov chain modeling DNA. Note that any sequence can be traced through the model by passing from one state to the next via the transitions.

A **transition probability** parameter (a_{ij}) is associated with each transition (arrow) and determines the probability of a certain state (S_j) following another state (S_i).

A Markov chain is defined by:

- a finite set of ***states***, $S_1, S_2 \ldots S_N$
- a set of ***transition probabilities:*** $a_{ij} = P(q_{t+1}=S_j|q_t=S_i)$
- *and* an ***initial state probability distribution***, $\pi_i = P(q_0=S_i)$

Simple Markov chain example for x={a,b}

Observed sequence: x = **`abaaababbaa`**

Model:

transition probabilities

Prev	Next	Prob
i	j	a_{ij}
a	a	0.7
a	b	0.3
b	a	0.5
b	b	0.5

initial state probability distribution

Start probs	π_i	a 0.5
		b 0.5

P(x) = 0.5 x 0.3 x 0.5 x 0.7 x 0.7 x 0.3 x 0.5 x 0.3 x 0.5 x 0.5 x 0.7

Q. Can you sketch the state diagram with labeled transitions for this model?

Typical questions we can ask with Markov chains include:

- What is the probability of being in a particular state at a particular time?
  (By time here we can read position in our query sequence)

- What is the probability of seeing a particular sequence of states?
  (I.e., the score for a particular query sequence given the model)

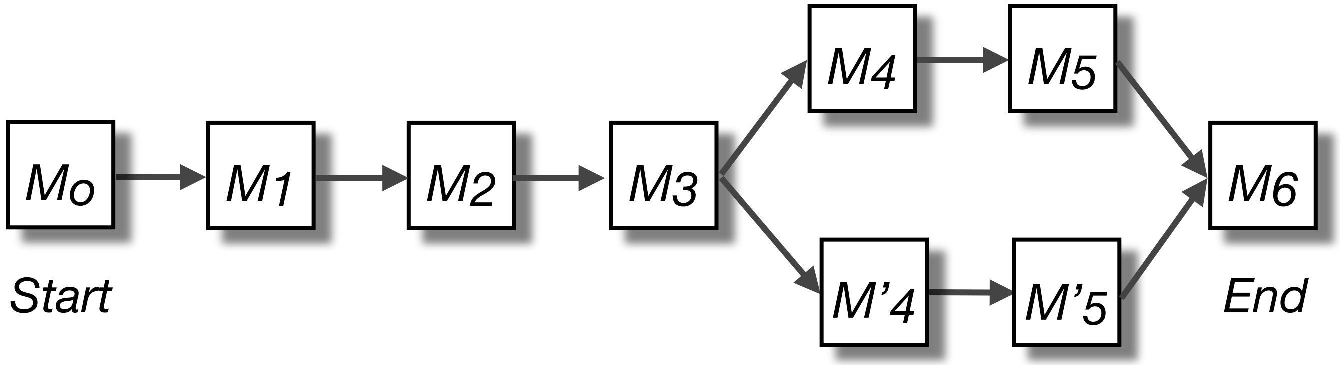
Q. What do Markov chains add over the traditional PSSM approach?
In particular how do Markov chains deal with the following PSSM weaknesses?

1. Positional dependencies
2. Pattern instances containing insertions or deletions
3. Variable length patterns, and
4. The detection boundaries (i.e. segmentation of sequences)

Markov chains: 1. Positional dependencies ✓

The connectivity or **topology** of a Markov chain can easily be designed to capture dependencies and variable length motifs.

```
WEIRD
WEIRD
WEIQH
WEIRD
WEIQH
```

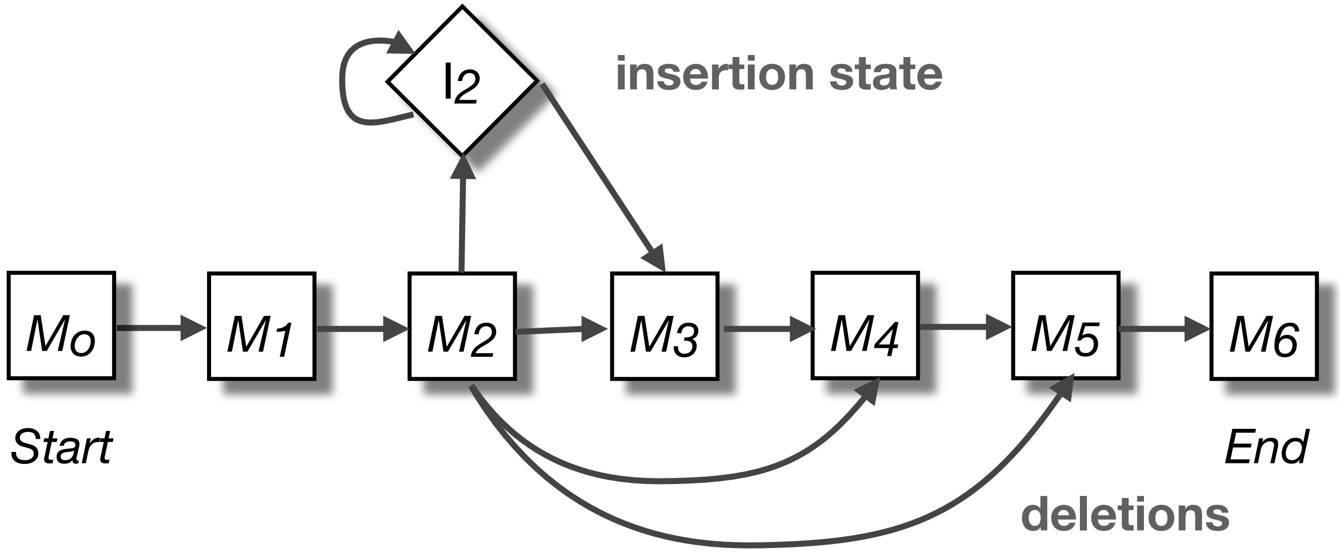


Recall that a PSSM for this motif would give the sequences **`WEIRD`** and **`WEIRH`** equally good scores even though the **`RH`** and **`QR`** combinations were not observed

Markov chains: 2. Insertions and deletions ✓

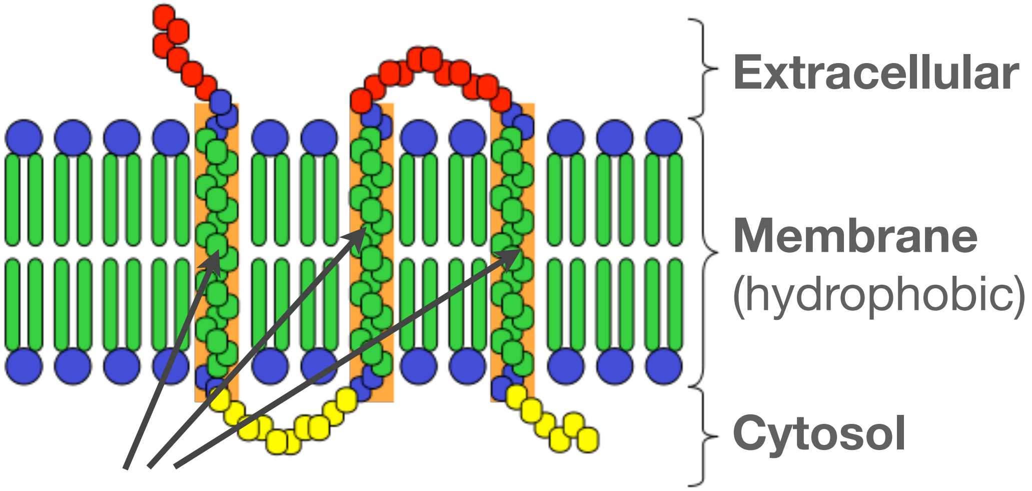
To address pattern instances with gaps and variable length motifs, we can construct a Markov chain to recognize a query sequences with insertions (via an extra insertion state) and deletions (via extra transitions (edges))

```
WETIRD
WE-IRD
WETIQH
WE-IRD
WE--QH
```



Markov chains: 3. Boundary detection ?

Giving a sequence we wish to label each symbol in the sequence according to its class (e.g. transmembrane regions or extracellular/cytosolic)



tend to be hydrophobic in composition

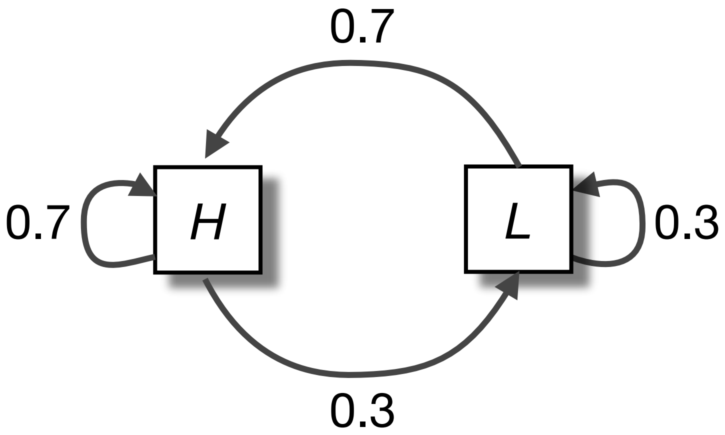
Given a training set of labeled sequences we can begin by modeling each amino acid as hydrophobic (**H**) or hydrophilic (**L**)
i.e. reduce the dimensionality of the 20 amino acids into two classes

E.g., A peptide sequence can be represented as a sequence of Hs and Ls.
e.g. HHHLLHLHHLHL...

Markov chains: boundary detection...

A simpler question: **is a given sequence a transmembrane sequence?**

A Markov chain for recognizing transmembrane sequences



- States: S_H, S_L
- $\Sigma=\{H,L\}$
- $\pi(H) = 0.6$, $\pi(L) = 0.4$

Question: Is sequence **`HHLHH`** a transmembrane protein?

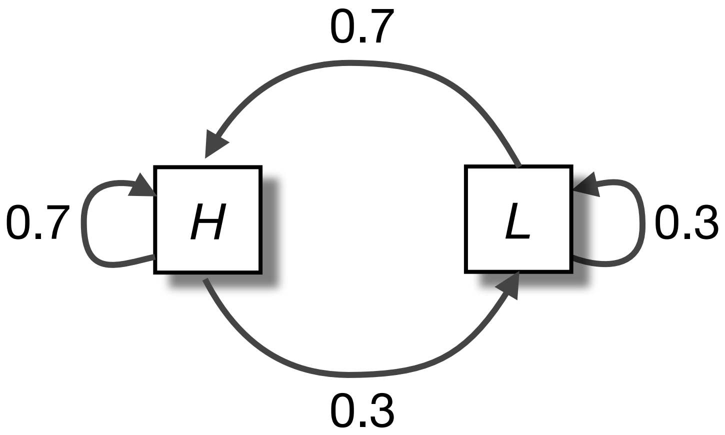
$$P(HHLHH) = 0.6 \times 0.7 \times 0.7 \times 0.3 \times 0.7 \times 0.7 = 0.043$$

Problem: need a threshold,
threshold must be length dependent

Markov chains: boundary detection

We can classify an observed sequence ($O = O_1, O_2, \ldots$) by its log odds ratio

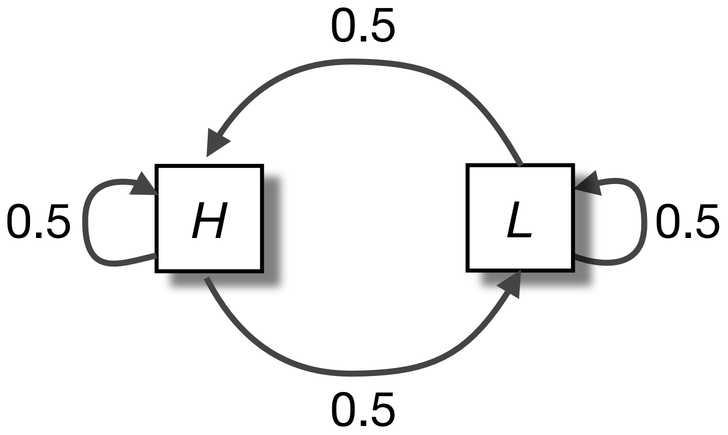
transmembrane model



Transmembrane (TM)

- $\pi(H) = 0.6$, $\pi(L) = 0.4$

null model



Extracellular/cytosolic (E/C)

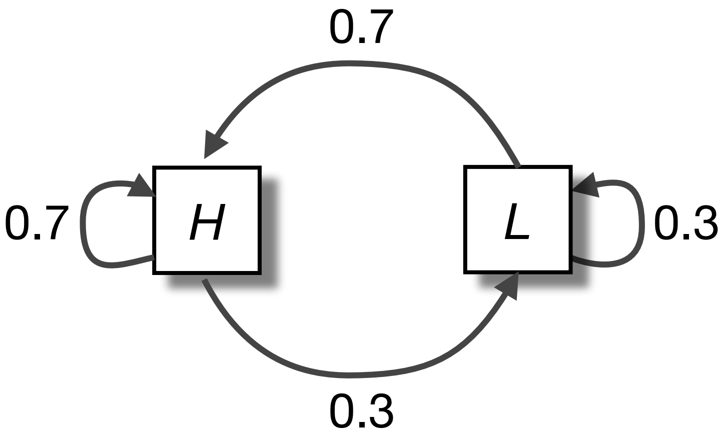
- $\pi(H) = 0.5$, $\pi(L) = 0.5$

$$\frac{P(HHLHH \mid TM)}{P(HHLHH \mid EC)} = \frac{0.6 \times 0.7 \times 0.7 \times 0.3 \times 0.7 \times 0.7}{0.5 \times 0.5 \times 0.5 \times 0.5 \times 0.5 \times 0.5} = \frac{0.043}{0.016} = 2.69$$

In other words, it is more than twice as likely that **`HHLHH`** is a transmembrane sequence. The log-odds score is: $\log_2(2.69) = 1.43$

Side note: Parameter estimation

Both initial probabilities (π(*i*))and transition probabilities (a_{ij}) are determined from known examples of transmembrane and non-transmembrane sequences.



- initial probabilities π(H), π(L)
- transition probabilities: a_{HH}, a_{HL}, a_{LH} and a_{LL}.

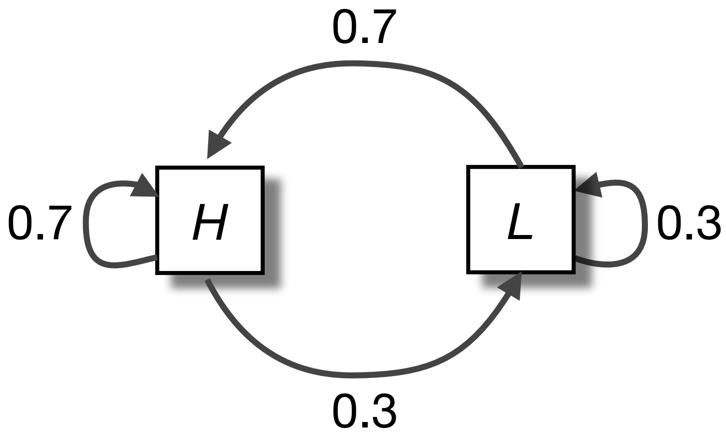
Given labeled sequences (TM and E/C), we determine the initial probabilities π(*i*) by counting the number of sequences that begin with residue *i*.

To determine transition probabilities, a_{ij}, we first determine A_{ij} (the number of transitions from state *i* to *j* in the training data, i.e. count the number of *ij* pairs in the training data). Then normalize by the number of *i** pairs.

$$a_{ij} = \frac{A_{ij}}{\sum_{j'} A_{ij'}}$$

Side note: Parameter estimation...

Both initial probabilities (π(*i*))and transition probabilities (a_{ij}) are determined from known examples of transmembrane and non-transmembrane sequences.



π(H) = # of sequences that begin with H, normalized by the total # of training sequences

- π(H) = 0.6, π(L) = 0.4

HHHLLHHHLLLHLHLLHLLLHLHHHL

HHHLHHLHLLLLLHHHHLLLHHHHHL

HH . . . (A_{HL} = 12, A_{H^*} = 40)

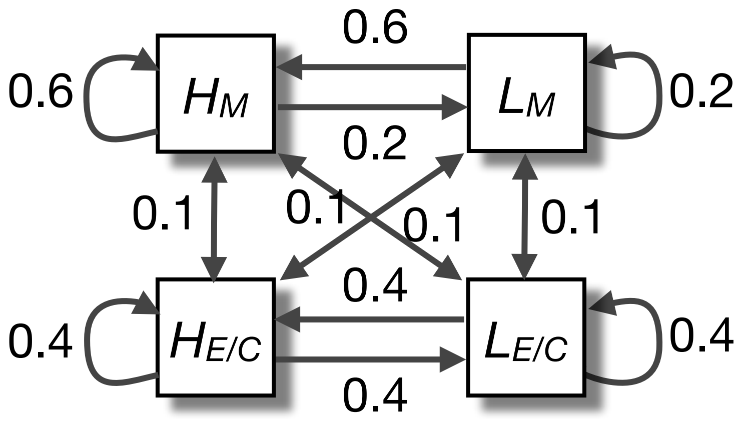
$$a_{HL} = \frac{A_{HL}}{\sum_i A_{Hi}} \qquad \frac{\text{\#HL pairs}}{\text{\# H* pairs}} \qquad \frac{12}{40}$$

Boundary detection challenge

Given sequence of Hs and Ls, find all transmembrane regions:

> Using our Markov models we would still need to score successive overlapping windows along the sequence, leading to a fuzzy boundary (just as with a PSSM).

To approach this question we can construct a new four state model by adding transitions connecting the TM and E/C models



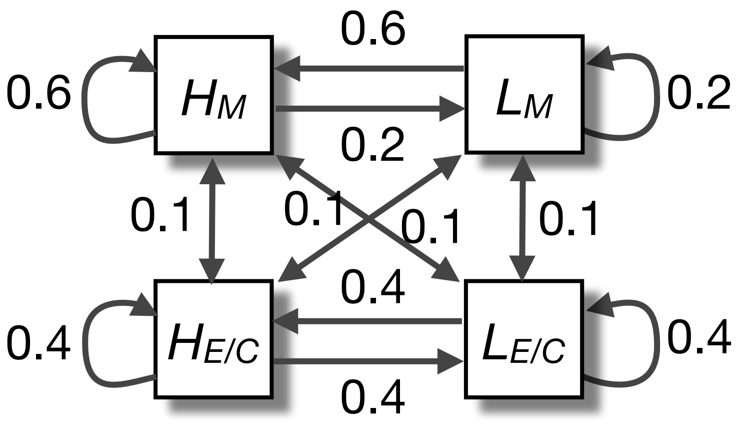
Transitions between the *M* states and the *E/C* states indicate boundaries between membrane regions and cytosolic or extracellular regions.

However this is no longer a standard Markov chain!

Boundary detection challenge...

In a Markov chain, there is a one-to-one correspondence between symbols and states, which is not true of our new merged four state, two symbol model.

For example, both H_M and $H_{E/C}$ are associated with hydrophilic residues.
- This four-state transmembrane model is a **hidden Markov model**.



So whats hidden?

We will distinguish between the *observed* parts of the problem and the *hidden* parts

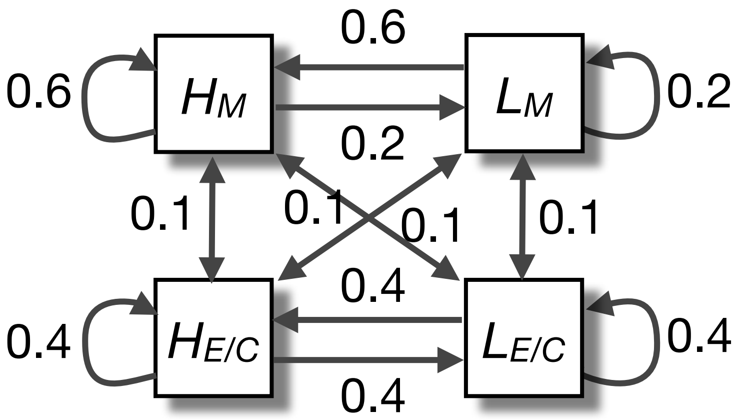
- In the Markov models we have considered previously it is clear which states account for each part of the observed sequence

  Due to the one-to-one correspondence between symbols and states

- In our new model, there are multiple states that could account for each part of the observed sequence

  i.e. we don't know which state emitted a given symbol from knowledge of the sequence and the structure of the model

  - This is the *hidden* part of the problem

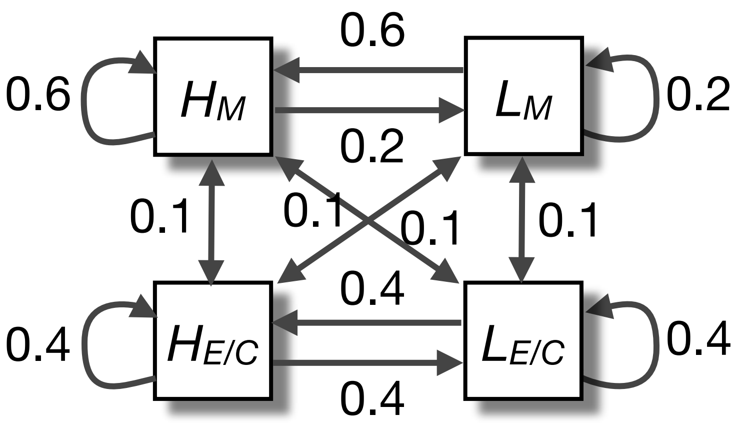


For our Markov models

- Given HLLH..., we know the exact state sequence (q_0=S_H, q_1=S_L, q2=$S_{L,}$ ...)

For our HMM

- Given HLLH..., we must infer the most probable state sequence
- This HMM state sequence will yield the boundaries between likely TM and E/C regions



```
HM, LM, LM, HM
HM, LM, LM, HE/C
HM, LM, LH/C, HM
HM, LM, LH/C, HE/C
HM, LE/C, LM, HM
HM, LE/C, LM, HE/C
HM, LE/C, LH/C, HM,
HM, LE/C, LH/C, HE/C,
HE/C, LM, LM, HM
HE/C, LM, LM, HE/C
HE/C, LM, LH/C, HM
HE/C, LM, LH/C, HE/C
HE/C, LE/C, LM, HM
HE/C, LE/C, LM, HE/C
HE/C, LE/C, LH/CM, HM
HE/C, LE/C, LH/CM, HE/C
```

Side note: HMM states as sequence emitters

It’s useful to imagine HMM states **emitting symbols** each time they are visited

In this way, transversing the model will “generate” a sequence with a certain probability (i.e. “score”).

This probability is a product of the state path taken through the model

That is, it depends on *initial probabilities*, *transition probabilities* and ***emission probabilities*** (the probability that a visited state emits a particular symbol) along the path

There may be many possible paths that can generate the same sequence

An HMM is a **full probabilistic model** – the model parameters θ and the overall sequence “scores” *P(x, S | HMM, θ)* are all probabilities. As a result, we can use standard **Bayesian probability theory** to manipulate these numbers in powerful ways, including optimizing parameters, calculating confidence in predictions, and interpreting the statistical significance of scores.

Hidden Markov models (HMMs)

Markov Chains

- States: $S_1, S_2 \ldots S_N$
- Initial probabilities: π_i
- Transition probabilities: a_{ij}

One-to-one correspondence between states and symbols

Hidden Markov Models

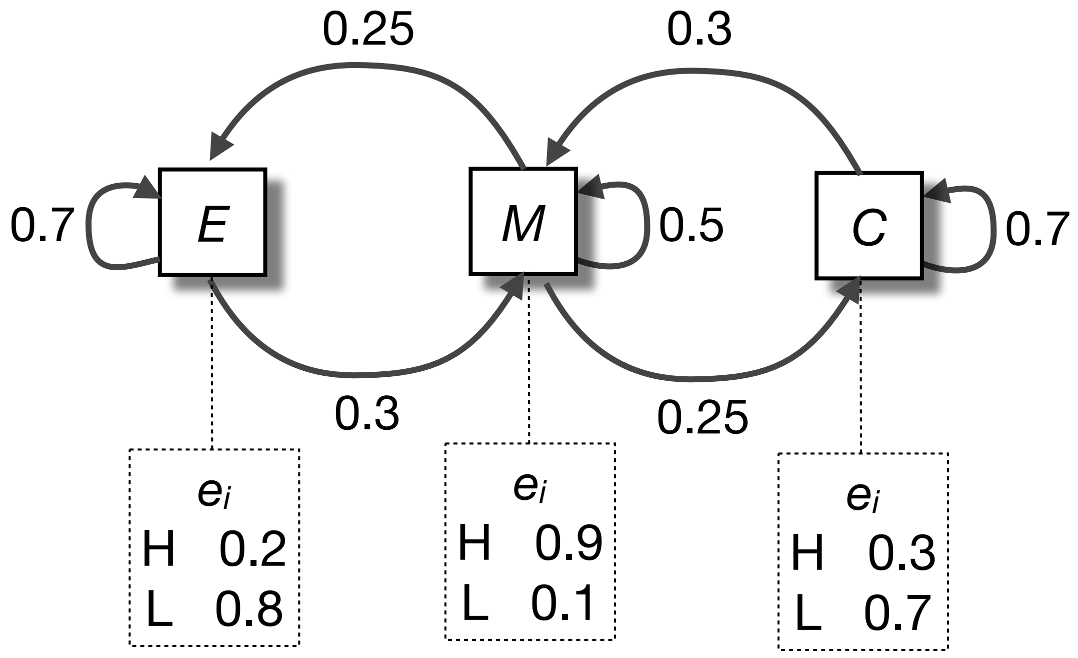
- States: $S_1, S_2 \ldots S_N$
- Initial probabilities: π_i
- Transition probabilities: a_{ij}
- **Alphabet** of emitted symbols, Σ
- **Emission probabilities**: $e_i(a)$
  probability state *i* emits symbol *a*

Symbol may be emitted by more than one state

Similarly, a state can emit more than one symbol

Example three state HMM

In this example we will use only one state for the transmembrane segment (*M*) and use emission probabilities to distinguish between H and L residues. We will also add separate *E* & *C* states with distinct emission probabilities.



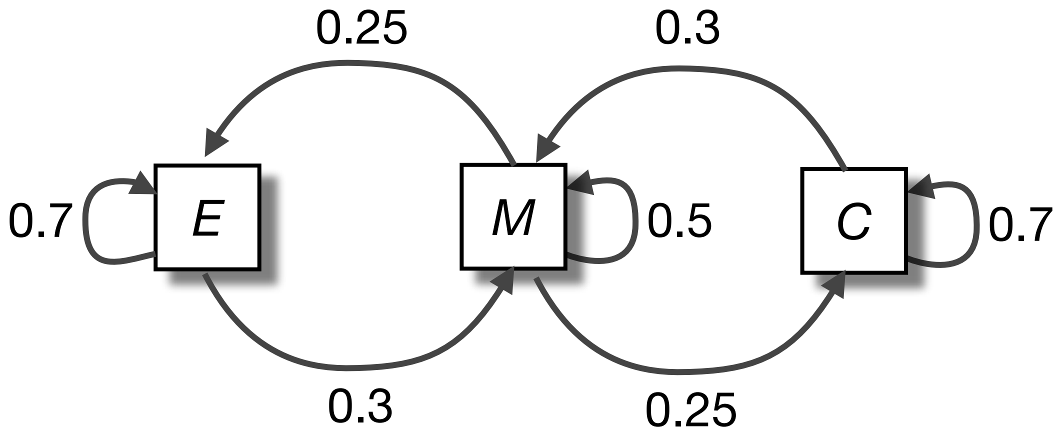
$$a_{ij} = \begin{bmatrix} 0.7 & 0.3 & 0 \\ 0.25 & 0.5 & 0.25 \\ 0 & 0.3 & 0.7 \end{bmatrix}$$

Side note: Parameter estimation

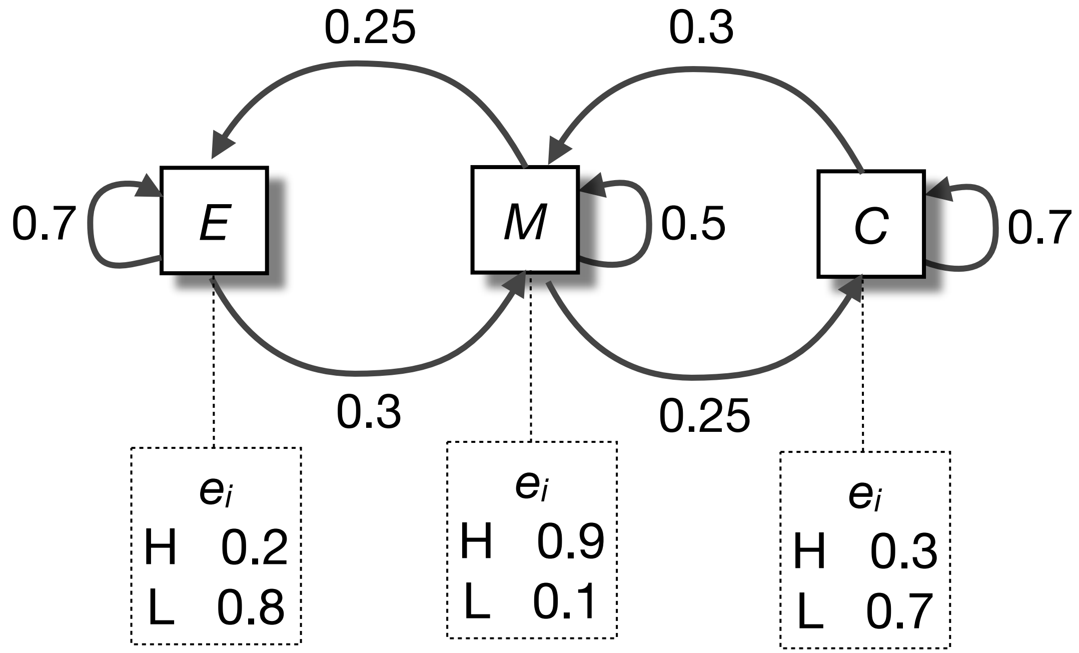
As in the case of Markov chains, the HMM parameters can be learned from labeled training data

Note that we now have to learn the initial probabilities, transition probabilities and *emission probabilities*

$$a_{ij} = \frac{A_{ij}}{\sum_{j'} A_{ij'}} \qquad e_i(x) = \frac{E_i(x)}{\sum_{x'} E_i(x')}$$



	E	*M*	*C*
π_i	0	0	1
$e_i(H)$	0.2	0.9	0.3
$e_i(L)$	0.8	0.1	0.7



e_i

H	0.2
L	0.8

e_i

H	0.9
L	0.1

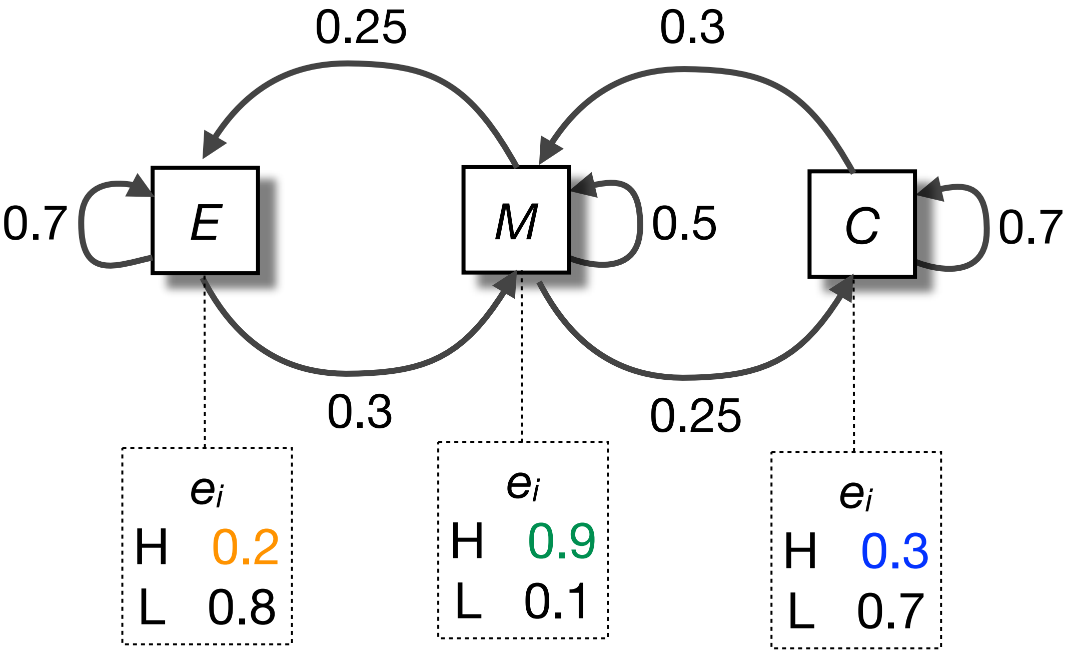
e_i

H	0.3
L	0.7

π(E) = 0
π(M) = 0
π(C) = 1

Query Sequence

States	**H**	**H**	**L**	**L**	**H**
E					
M					
C					
START					



e_i (*E*)	
H	0.2
L	0.8

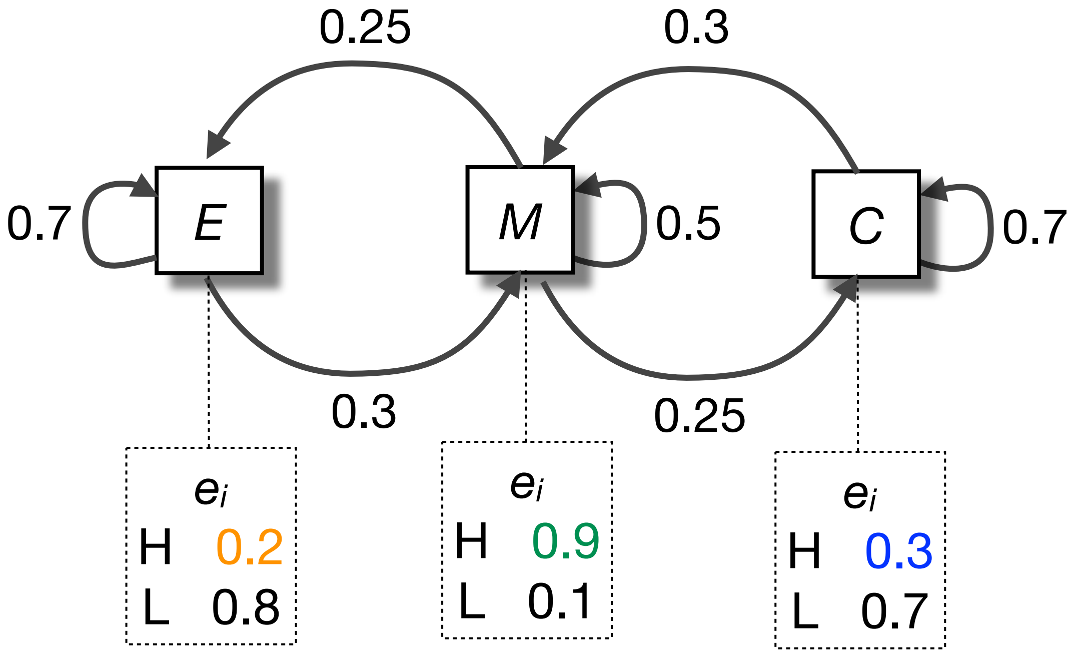
e_i (*M*)	
H	0.9
L	0.1

e_i (*C*)	
H	0.3
L	0.7

π(E) = 0
π(M) = 0
π(C) = 1

Query Sequence

States	H	H	L	L	H
E	0x0.2 =0				
M	0x0.9 =0				
C	1x0.3 =0.3				
START					

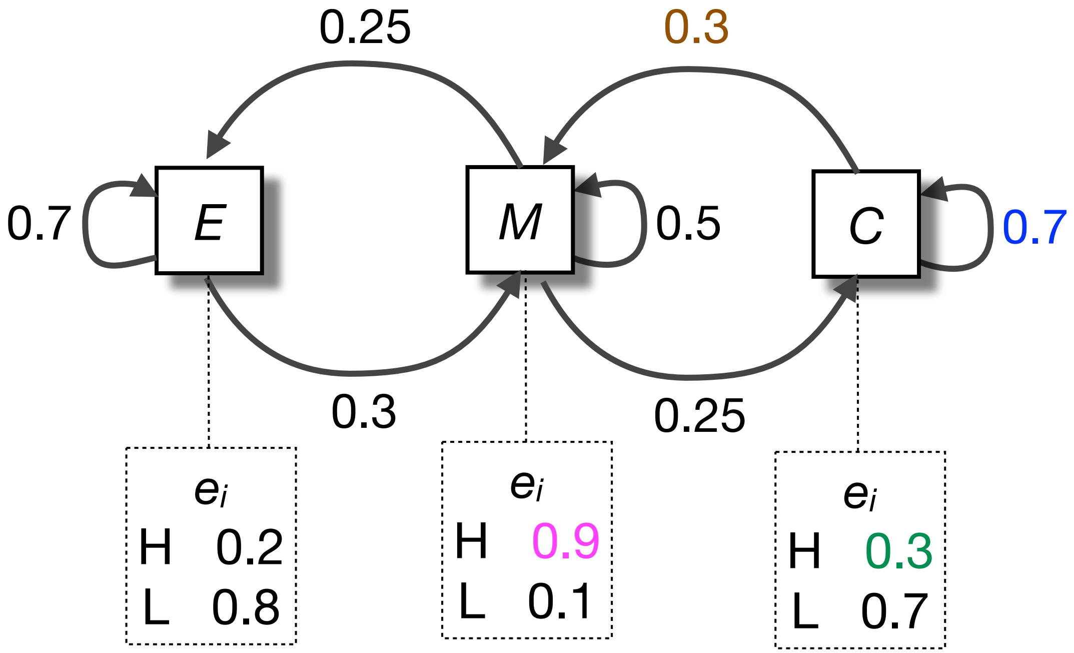


π(E) = 0
π(M) = 0
π(C) = 1

Query Sequence

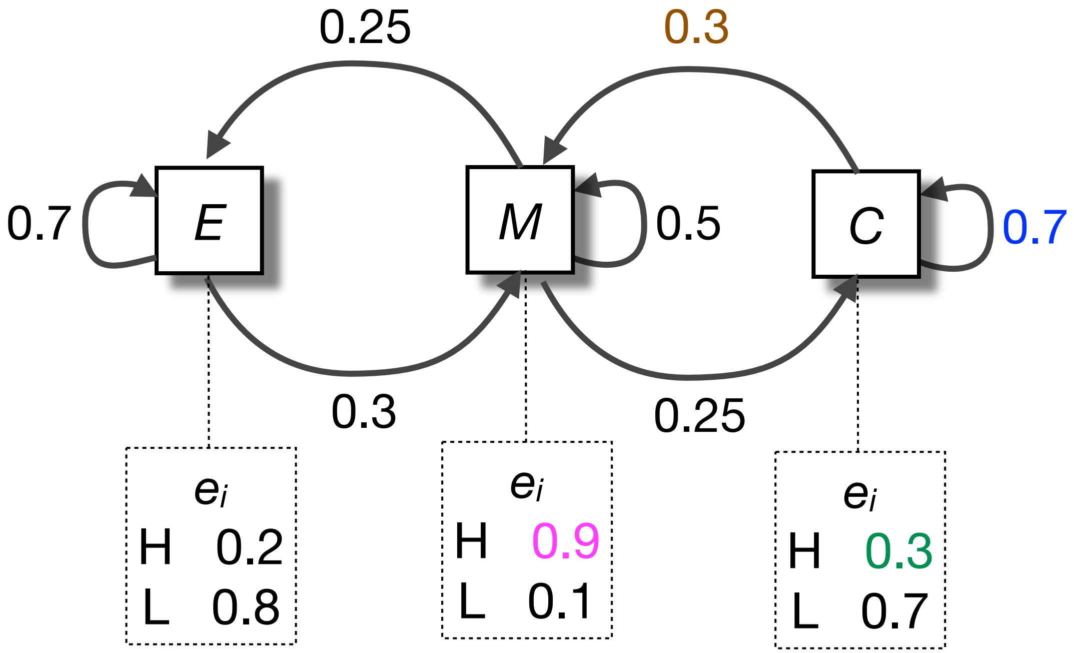
States	H	H	L	L	H
E	0x0.2 =0				
M	0x0.9 =0				
C	1x0.3 =0.3				

START



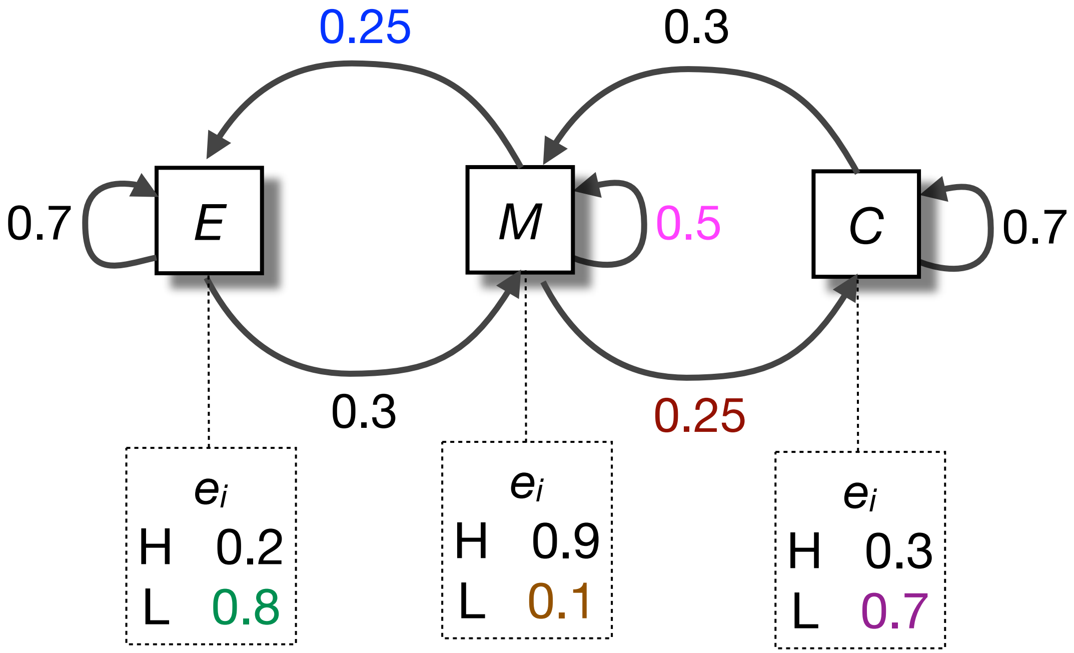
Query Sequence

States	H	H	L	L	H
E	0x0.2 =0	-			
M	0x0.9 =0	0.3x0.9x0.3 =0.081			
C	1x0.3 =0.3	0.7x0.3x0.3 =0.063			
START					



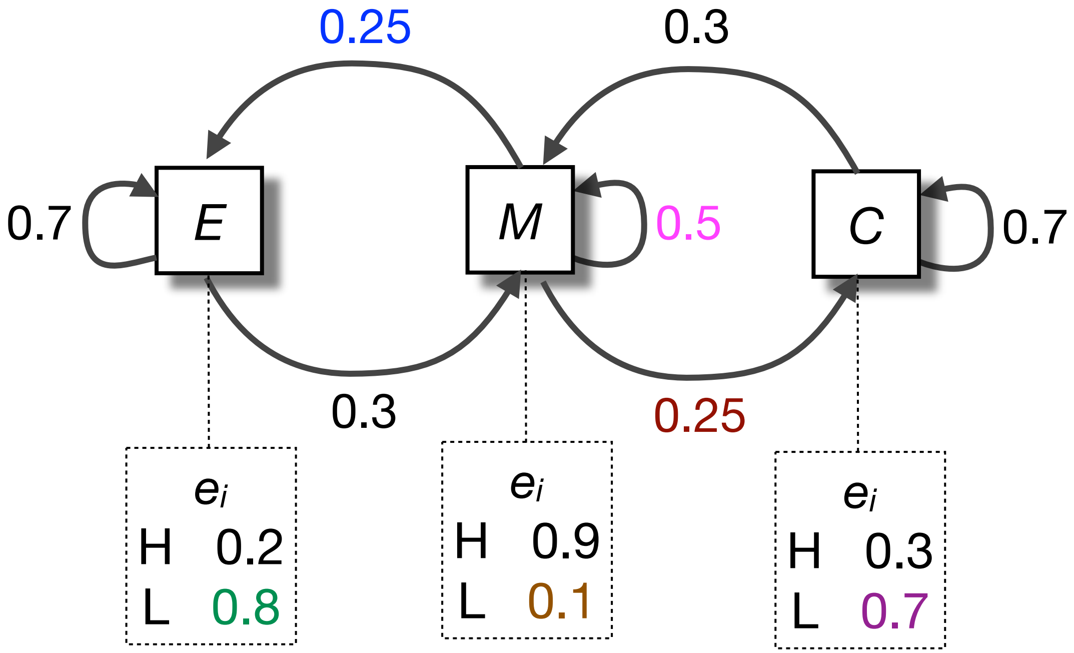
Query Sequence

States	**H**	**H**	**L**	**L**	**H**
E	0x0.2 =0	-			
M	0x0.9 =0	0.3x0.9x0.3 =0.081			
C	1x0.3 =0.3	0.7x0.3x0.3 =0.063			
START					



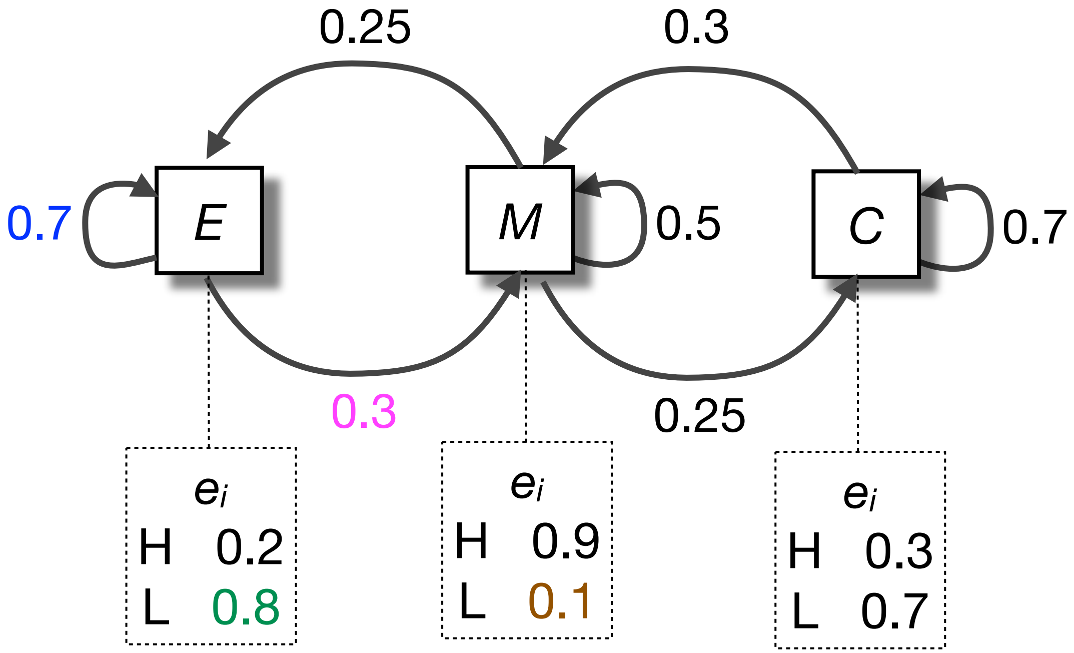
Query Sequence

States	H	H	L	L	H
E	0x0.2 =0	-	0.25x0.8x0.081 =0.016		
M	0x0.9 =0	0.3x0.9x0.3 =0.081	0.5x0.1x0.081 =0.04		
C	1x0.3 =0.3	0.7x0.3.0.3 =0.063	0.25x0.7x0.081 =0.014		
START					



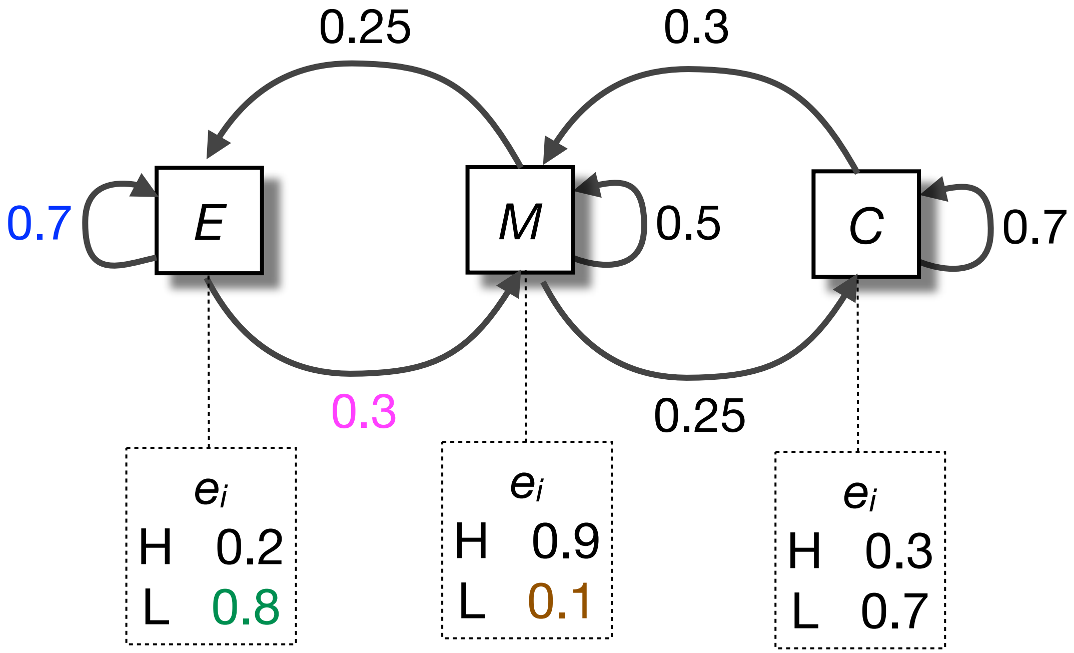
Query Sequence

States	H	H	L	L	H
E	0x0.2 =0	-	0.25x0.8x0.081 =0.016		
M	0x0.9 =0	0.3x0.9x0.3 =0.081	0.5x0.1x0.081 =0.04		
C	1x0.3 =0.3	0.7x0.3.0.3 =0.063	0.25x0.7x0.081 =0.014		
START					



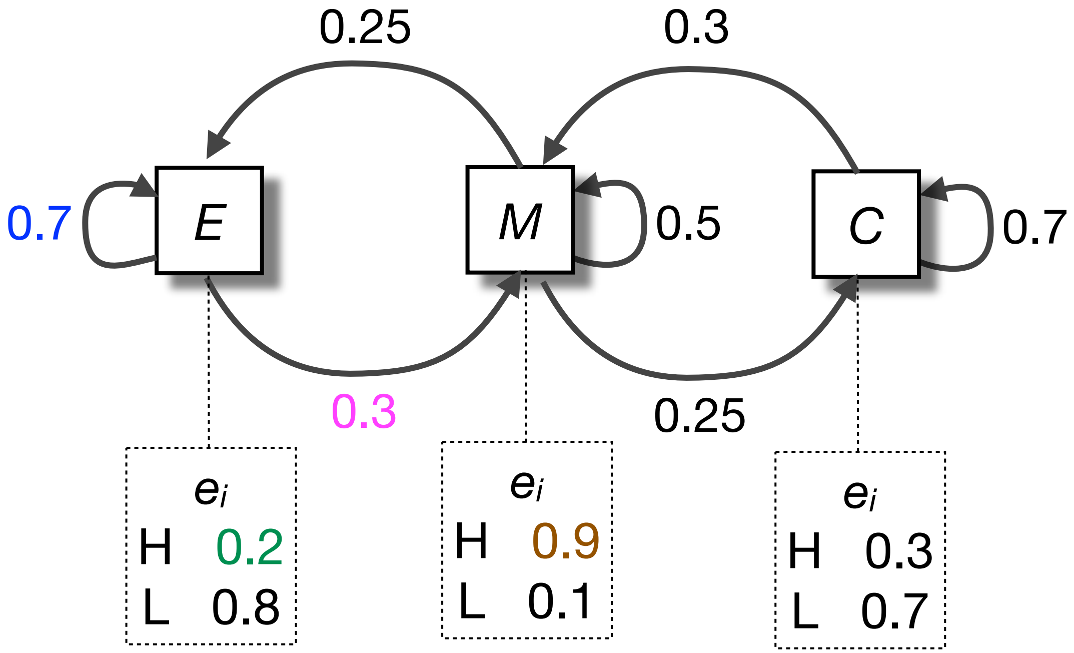
Query Sequence

States	H	H	L	L	H
E	0x0.2 =0	-	0.25x0.8x0.081 =0.016	0.7x0.8x0.016 =0.009	
M	0x0.9 =0	0.3x0.9x0.3 =0.081	0.5x0.1x0.081 =0.04	0.3x0.1x0.016 =0.0005	
C	1x0.3 =0.3	0.7x0.3.0.3 =0.063	0.25x0.7x0.081 =0.014	-	
START					



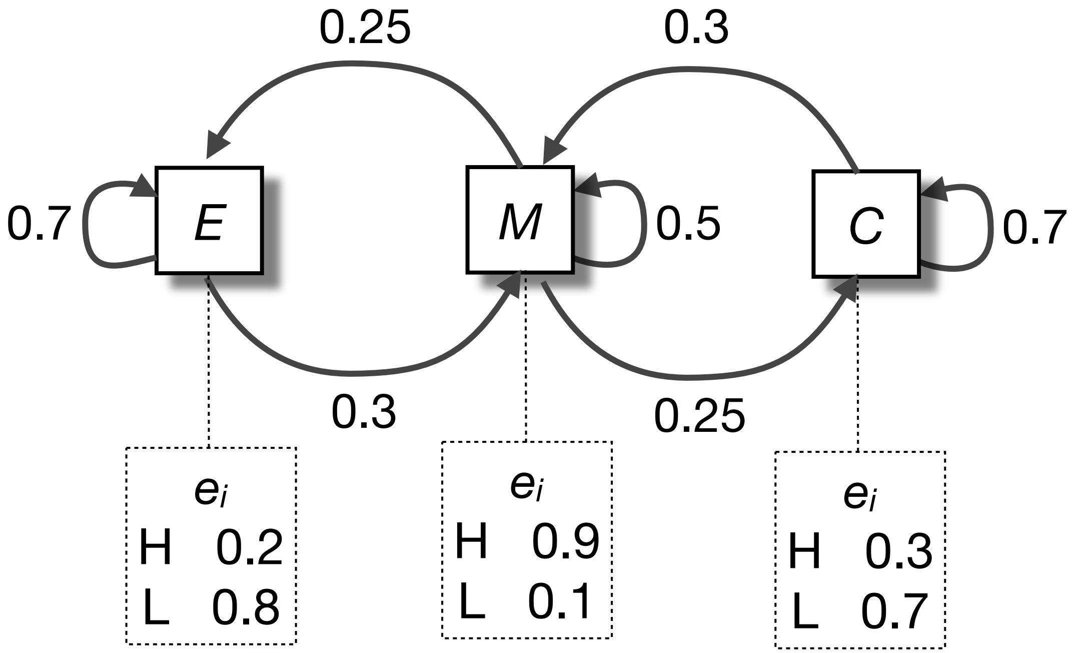
Query Sequence

States	H	H	L	L	H
E	0x0.2 =0	-	0.25x0.8x0.081 =0.016	0.7x0.8x0.016 =0.009	
M	0x0.9 =0	0.3x0.9x0.3 =0.081	0.5x0.1x0.081 =0.04	0.3x0.1x0.016 =0.0005	
C	1x0.3 =0.3	0.7x0.3.0.3 =0.063	0.25x0.7x0.081 =0.014	-	
START					



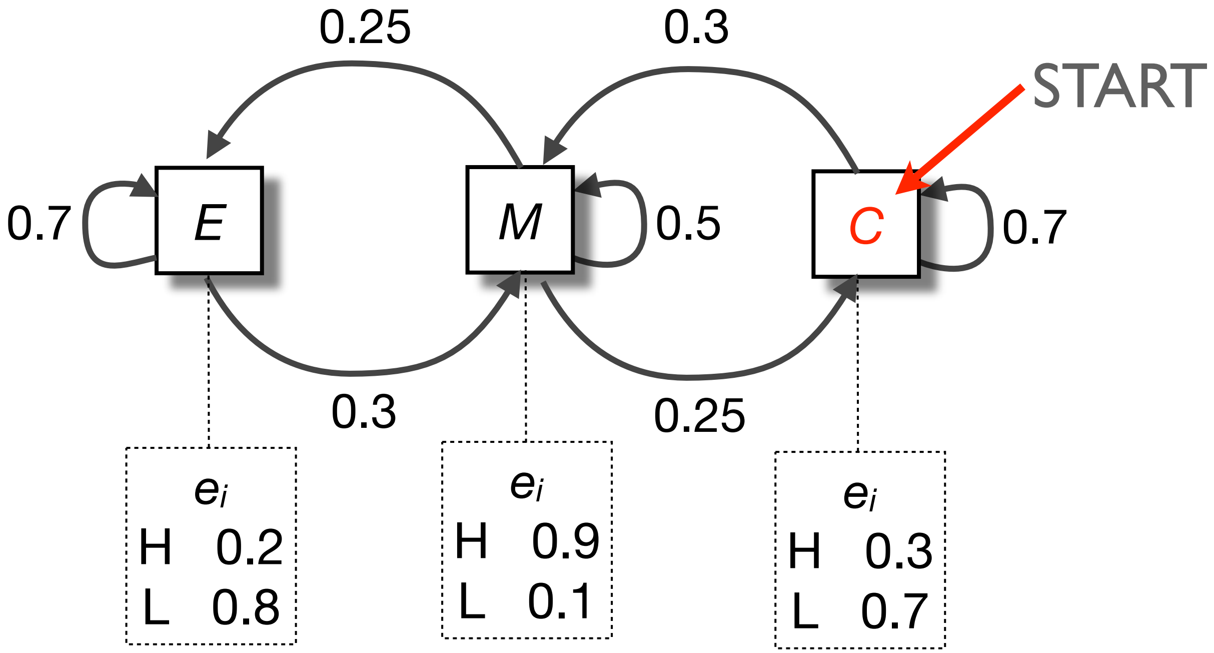
Query Sequence

States	H	H	L	L	H
E	0x0.2 =0	-	0.25x0.8x0.081 =0.016	0.7x0.8x0.016 =0.009	0.7x0.2x0.009 =0.001
M	0x0.9 =0	0.3x0.9x0.3 =0.081	0.5x0.1x0.081 =0.04	0.3x0.1x0.016 =0.0005	0.3x0.9x0.009 =0.002
C	1x0.3 =0.3	0.7x0.3.0.3 =0.063	0.25x0.7x0.081 =0.014	-	-
START					



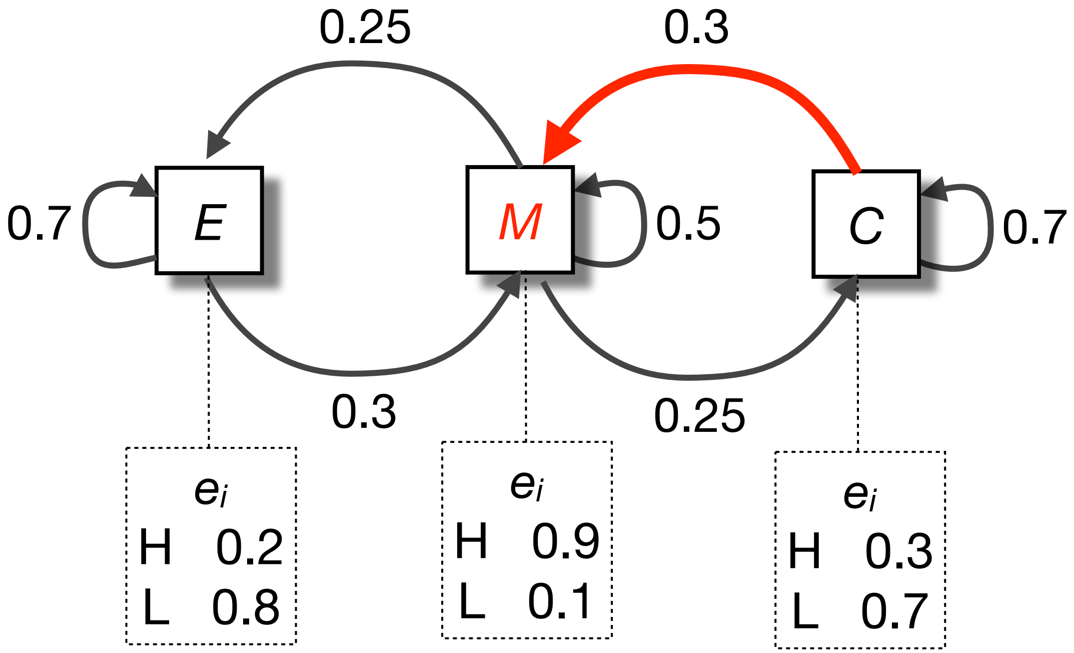
Query Sequence

States	H	H	L	L	H
E	0x0.2 =0	-	0.25x0.8x0.081 =0.016	0.7x0.8x0.016 =0.009	0.7x0.2x0.009 =0.001
M	0x0.9 =0	0.3x0.9x0.3 =0.081	0.5x0.1x0.081 =0.04	0.3x0.1x0.016 =0.0005	0.3x0.9x0.009 =0.002
C	1x0.3 =0.3	0.7x0.3.0.3 =0.063	0.25x0.7x0.081 =0.014	-	-
START					



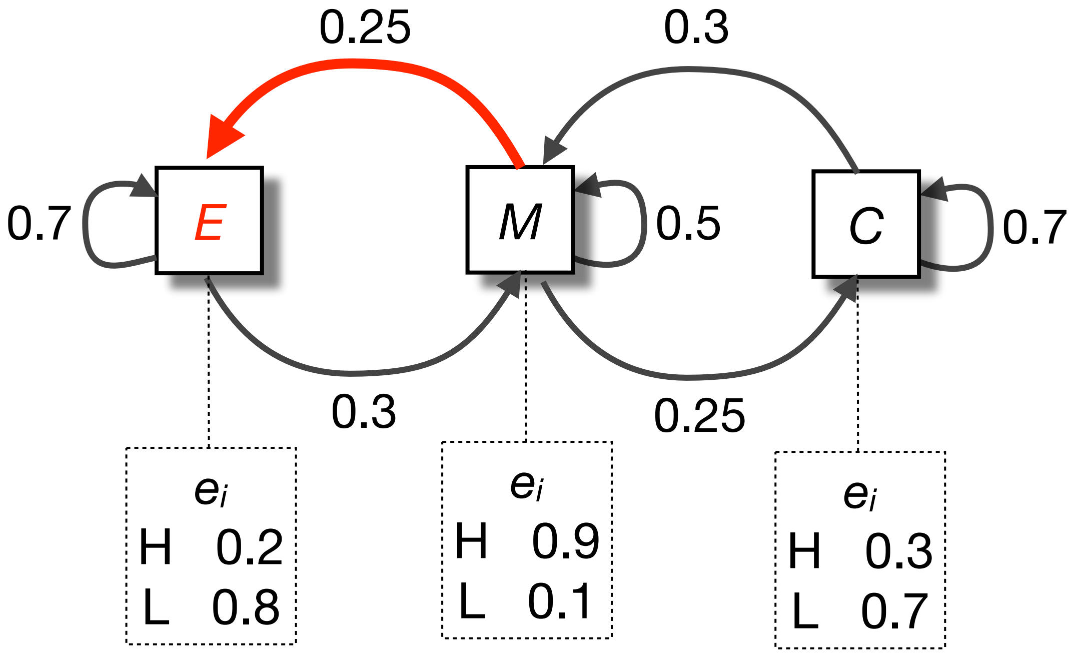
Query Sequence

States	H	H	L	L	H
E	0x0.2 =0	-	0.25x0.8x0.081 =0.016	0.7x0.8x0.016 =0.009	0.7x0.2x0.009 =0.001
M	0x0.9 =0	0.3x0.9x0.3 =0.081	0.5x0.1x0.081 =0.04	0.3x0.1x0.016 =0.0005	0.3x0.9x0.009 =0.002
C	1x0.3 =0.3	0.7x0.3.0.3 =0.063	0.25x0.7x0.081 =0.014	-	-
START	C				



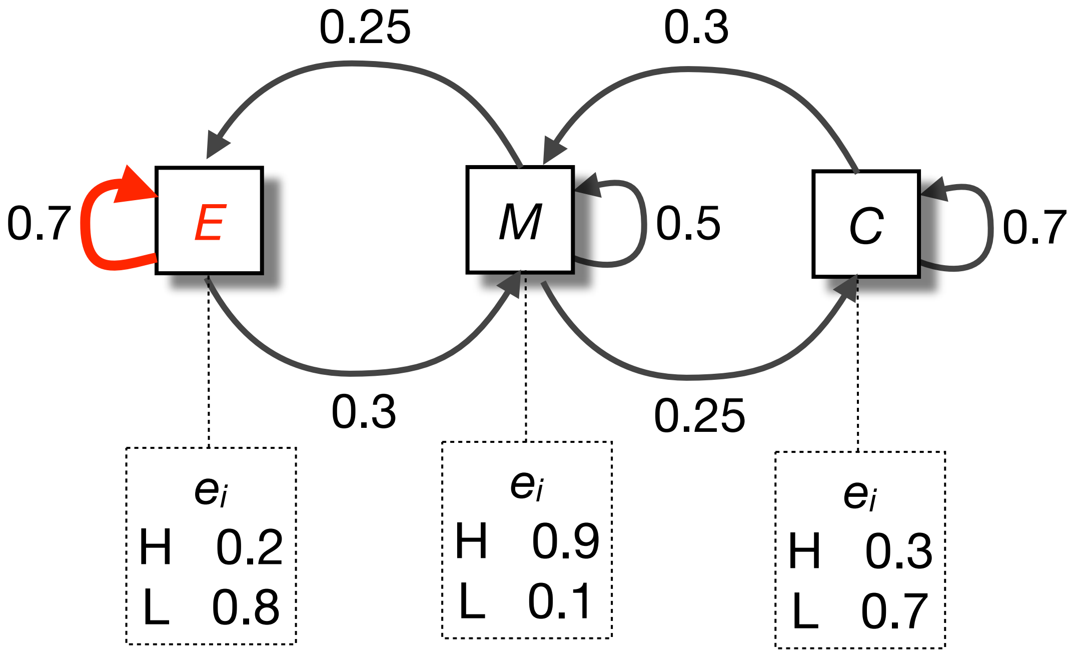
Query Sequence

States	H	H	L	L	H
E	0x0.2 =0	-	0.25x0.8x0.081 =0.016	0.7x0.8x0.016 =0.009	0.7x0.2x0.009 =0.001
M	0x0.9 =0	0.3x0.9x0.3 =0.081	0.5x0.1x0.081 =0.04	0.3x0.1x0.016 =0.0005	0.3x0.9x0.009 =0.002
C	1x0.3 =0.3	0.7x0.3.0.3 =0.063	0.25x0.7x0.081 =0.014	-	-
START	C	M			



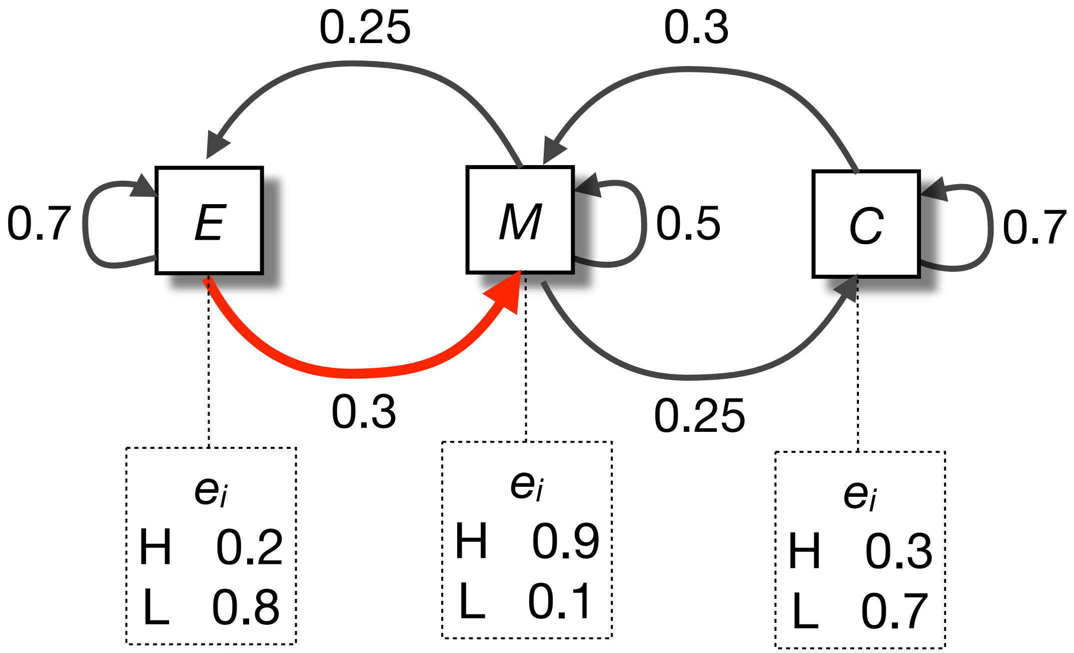
Query Sequence

States	H	H	L	L	H
E	0x0.2 =0	-	0.25x0.8x0.081 =0.016	0.7x0.8x0.016 =0.009	0.7x0.2x0.009 =0.001
M	0x0.9 =0	0.3x0.9x0.3 =0.081	0.5x0.1x0.081 =0.04	0.3x0.1x0.016 =0.0005	0.3x0.9x0.009 =0.002
C	1x0.3 =0.3	0.7x0.3.0.3 =0.063	0.25x0.7x0.081 =0.014	-	-
START	C	M	E		



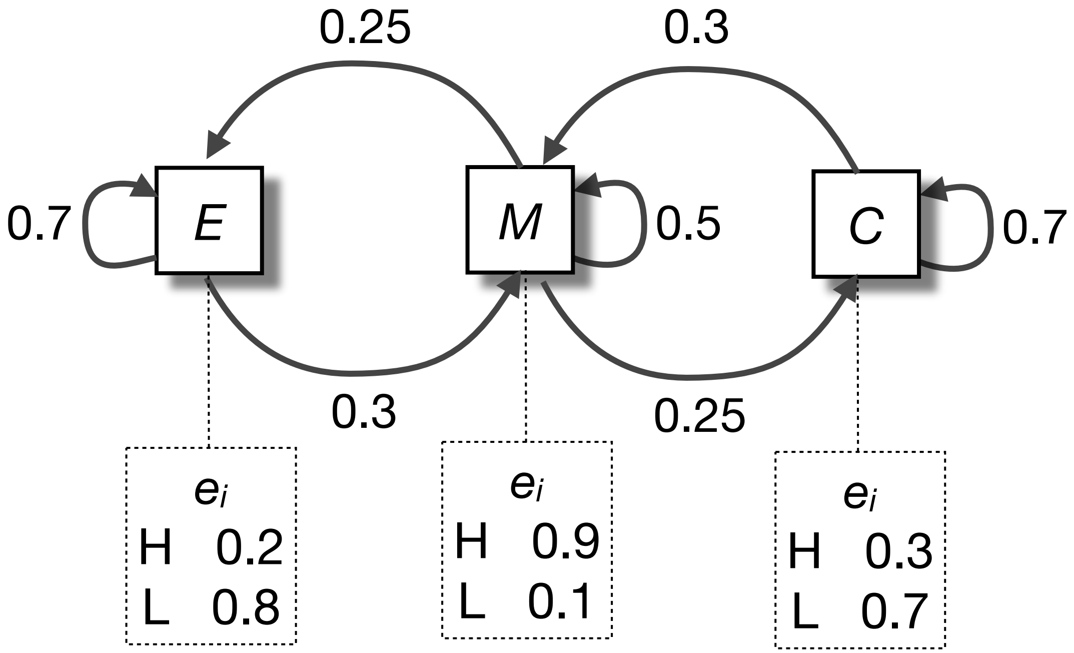
Query Sequence

States	**H**	**H**	**L**	**L**	**H**
E	0x0.2 =0	-	0.25x0.8x0.081 =0.016	0.7x0.8x0.016 =0.009	0.7x0.2x0.009 =0.001
M	0x0.9 =0	0.3x0.9x0.3 =0.081	0.5x0.1x0.081 =0.04	0.3x0.1x0.016 =0.0005	0.3x0.9x0.009 =0.002
C	1x0.3 =0.3	0.7x0.3.0.3 =0.063	0.25x0.7x0.081 =0.014	-	-
START	C	M	E	E	



Query Sequence

States	H	H	L	L	H
E	0x0.2 =0	-	0.25x0.8x0.081 =0.016	0.7x0.8x0.016 =0.009	0.7x0.2x0.009 =0.001
M	0x0.9 =0	0.3x0.9x0.3 =0.081	0.5x0.1x0.081 =0.04	0.3x0.1x0.016 =0.0005	0.3x0.9x0.009 =0.002
C	1x0.3 =0.3	0.7x0.3.0.3 =0.063	0.25x0.7x0.081 =0.014	-	-
START	C	M	E	E	M



Query Sequence

States	**H**	**H**	**L**	**L**	**H**
E	0x0.2 =0	-	0.25x0.8x0.081 =0.016	0.7x0.8x0.016 =0.009	0.7x0.2x0.009 =0.001
M	0x0.9 =0	0.3x0.9x0.3 =0.081	0.5x0.1x0.081 =0.04	0.3x0.1x0.016 =0.0005	0.3x0.9x0.009 =0.002
C	1x0.3 =0.3	0.7x0.3.0.3 =0.063	0.25x0.7x0.081 =0.014	-	-
START	C	M	E	E	M

Most Probable State Sequence

We have just used the Viterbi algorithm

The **Viterbi algorithm** finds the most probable "state path" (*S**) (i.e. sequence of hidden states) for generating a given sequence ($x = x_1, x_2, \ldots x_N$)

$$S^* = argmax\ P(x,S)$$

This process is often called **decoding** because we "decode" the sequence of symbols to determine the hidden sequence of states

HMMs were original developed in the field of speech recognition, where speech is "decoded" into words or phonemes to determine the meaning of the utterance

Note that we could have used brute force by calculating $P(x|S)$ for all paths but this quickly becomes intractable for longer sequences or HMMs with a large number of states

The Viterbi algorithm is guaranteed to find the most probable state path given a sequence and an HMM

See Durbin *et al. Biological Sequence Analysis*

Three key HMM algorithms

- **Viterbi algorithm**
  Given observed sequence *x* and an HMM *M*, composed of states *S*, calculate the most likely state sequence, *S**
  - $S^* = argmax\ P(x,S)$

- **Forward algorithm**
  Given observed sequence *x* and an HMM composed of states *S*, calculate the probability of the sequence for the HMM, *P(x|M)*
  - $P(x) = \sum_{S} P(x,S)$

- **Baum-Welch algorithm**
  Given many observed sequences, estimate the parameters of the HMM
  - heuristic expectation maximization method to optimize of a_{ij} and $e_i(a)$

The forward algorithm

Another important question is how well does a given sequence fit the HMM?

To answer this question we must sum over all possible state paths that are consistent with the sequence in question
(Because we don't know which path emitted the sequence)

The number of paths can quickly become intractable. The **forward algorithm** is a simple dynamic programing solution that makes use of the Markov property so that we don’t have to explicitly enumerate every path.

The **forward algorithm** basically replaces the maximization step of the Viterbi algorithm with sums to calculate the probability of the sequence given a HMM.

$$P(x) = \sum_{S} P(x,S)$$

See Durbin *et al. Biological Sequence Analysis*

The Baum-Welch algorithm

The **Baum-Welch algorithm** is an **heuristic optimization** algorithm for learning probabilistic models in problems that involve hidden states

If we *know* the state path for each training sequence (i.e. no hidden states with respect to the training sequences), then learning the model parameters is simple (just like it was for Markov chain models)

- count how often each transition and emission occurs
- normalize to get probabilities

If we *don't know* the path for each training sequence, we can use the **Baum-Welch algorithm**, an expectation maximization method, which estimates counts by considering every path weighted by its probability

- start from a given initial guess for the parameters
- perform a calculation which is guaranteed to improve the previous guess
- run until there is little change in parameters between iterations

For sequence profile-HMMs we train from a MSA and hence we can *estimate* our probabilities from the observed sequences

Segmentation/boundary detection

Given: A test sequence and a HMM with different sequence classes
Task: Segment the sequence into subsequences, predicting the class of each subsequence
Question: What is the most probable “path” (sequence of hidden states) for generating a given sequence from the HMM?
Solution: Use the **Viterbi algorithm**

Classification/sequence scoring

Given: A test sequence and a set of HMMs representing different sequence classes
Task: Determine which HMM/class best explains the sequence
Question: How likely is a given sequence given a HMM?
Solution: Use the **Forward algorithm**

Learning/parameterization

Given: A model, a set of training sequences
Task: Find model parameters that explain the training sequences
Question: Can we find a high probability model for sequence characterization
Solution: Use the **Forward backward algorithm**

Segmentation/boundary detection

Question: What is the most probable "path" (sequence of hidden states) for generating a given sequence from the HMM?

HMMER: **hmmalign** - align sequences to our HMM

Classification/sequence scoring

Question: How likely is a given sequence given a HMM?

HMMER: **hmmsearch** - find sequences that match our HMM

Learning/parameterisation

Question: Can we find a high probability model for sequence characterization

HMMER: **hmmbuild** - setup our HMM parameters

Half time break...

Questions:
For what kinds of motifs are PSSMs not well suited?
What is the Markov property?
In what important ways do HMMs differ Markov chains?
What is the Viterbi algorithm used for?
How does the Forward algorithm differ from the Viterbi algorithm?

For what kinds of motifs are PSSMs not well suited?

PSSMs are not well suited to pattern instances containing insertions or deletions, variable length patterns and those with positional dependencies.

What is the Markov property?

The Markov property states that the conditional probability distribution for the system at the next step (and in fact at all future steps) depends only on the current state of the system, and not additionally on the state of the system at previous steps.

In what important ways do HMMs differ Markov chains?

HMMs differ from Markov chains in a number of ways:

- In HMMs, the sequence of states visited is hidden. Unlike Markov Chains, there is no longer a one-to-one correspondence between states and output symbols.
- In a HMM the same symbol may be emitted by more than one state.
- In a HMM a state can emit more than one symbol.

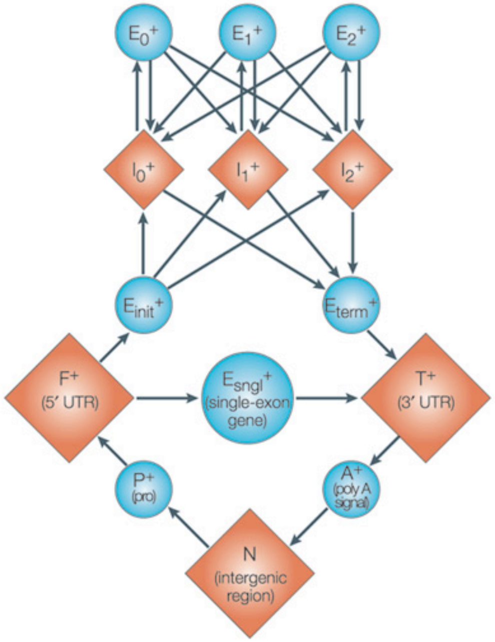
What is the Viterbi algorithm used for?

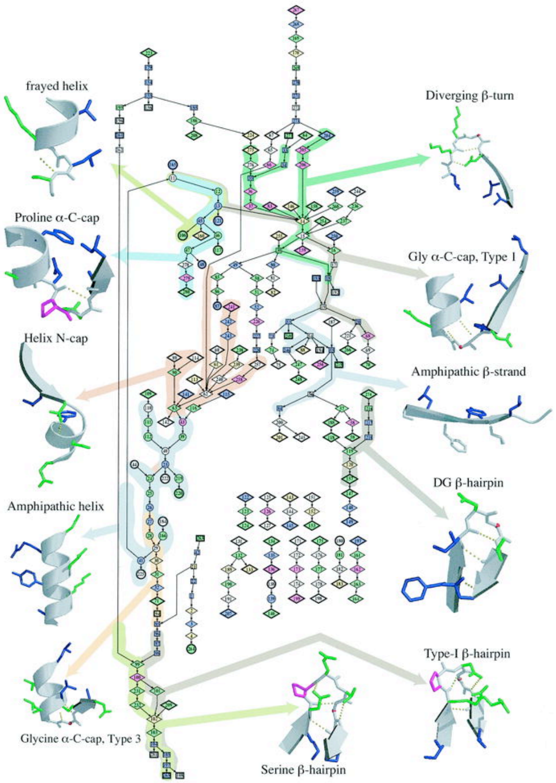
The Viterbi algorithm is used to find the most probable state path given a sequence and an HMM

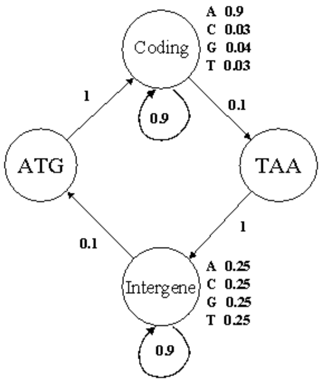
HMM network structure is hand tailored to the problem

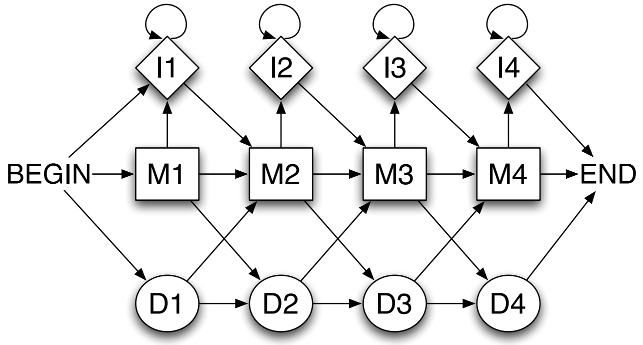
No algorithm for the prediction of optimal HMM network structure and probabilities has yet been able to beat simple hand-built topologies

These topologies are tailored to the problem at hand - exon/intron detection, transmembrane regions, secondary structure elements, protein families...

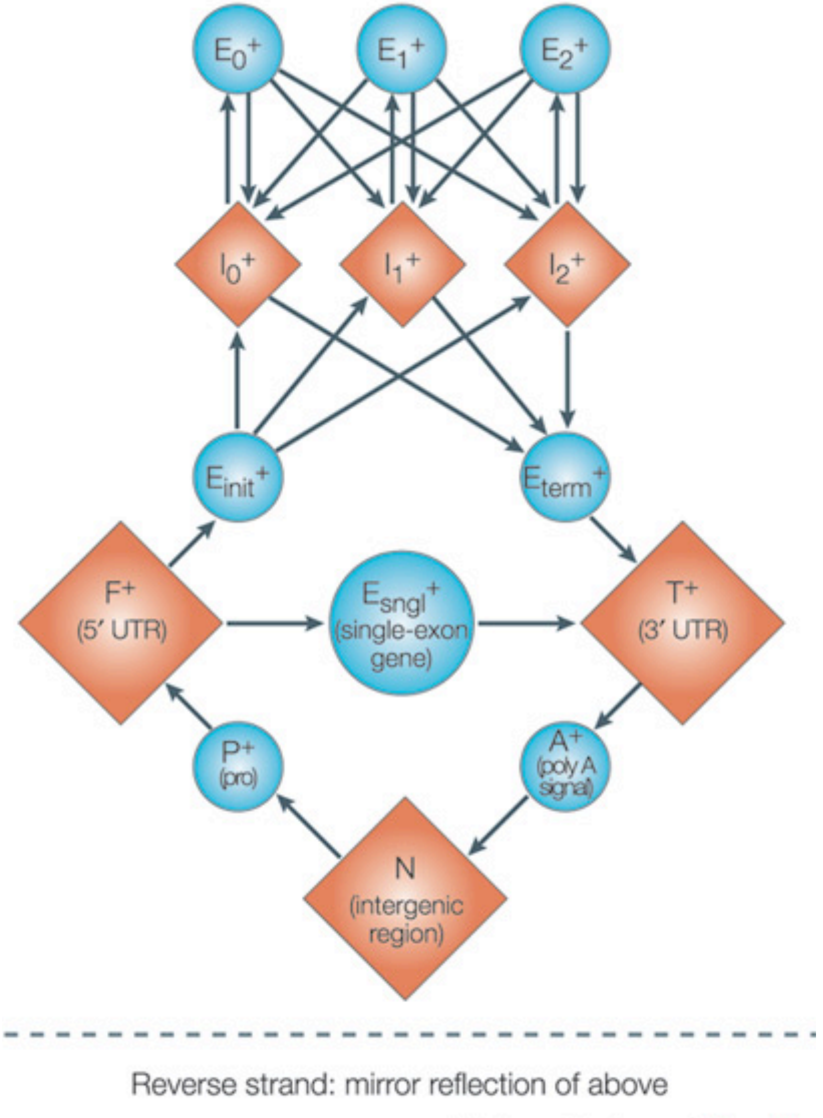








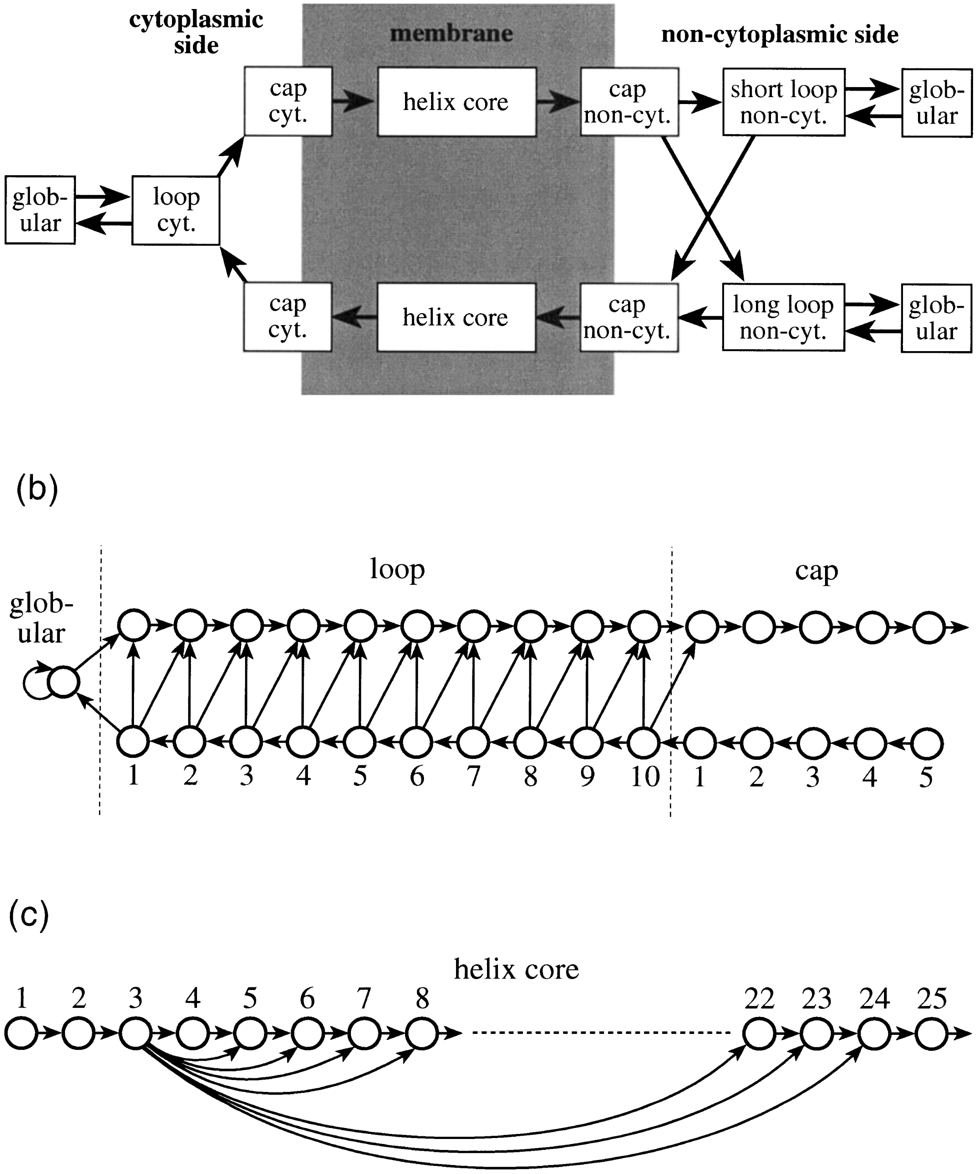
GenScan - gene-prediction HMM



Here, each circle or square represents a functional unit (a state) of a gene on its forward strand (for example, E_{init} is the 5' coding sequence (CDS) and E_{term} is the 3' CDS, and the arrows represent the transition probability from one state to another. The GenScan HMM is trained by pre-computing the transition probabilities from a set of known gene structures.

See: Zhang et al. (2002) Nature Reviews Genetics 3, 698-709

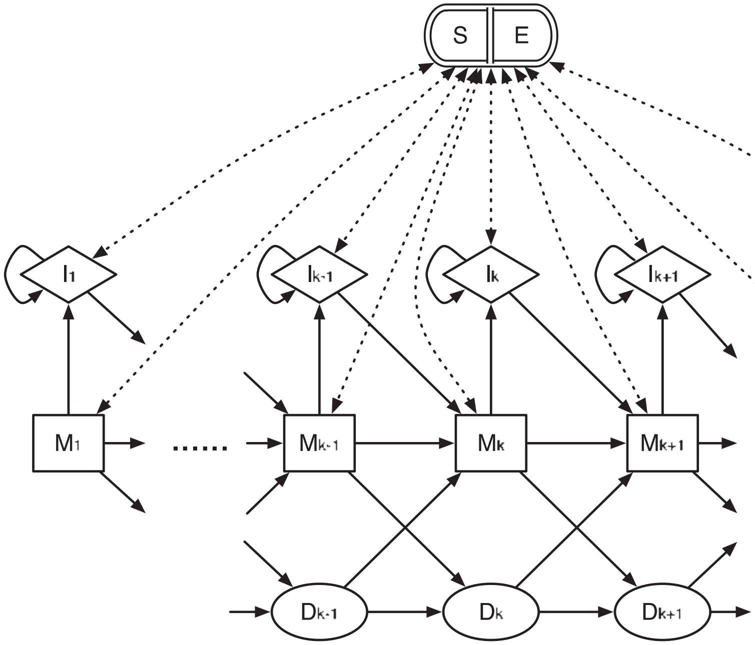
TMHMM - transmembrane protein topology prediction



Each box corresponds to one or more states in the HMM. Cyt. represents the cytoplasmic side of the membrane and non-cyt. the other side. (b) The detailed structure of the inside and outside loop models and helix cap models. (c) The structure of the model for the helix core modeling lengths between 5 and 25, which translates to helices between 15 and 35 when the caps are included.

See: Krogh et al. (2001) JMB 305, 567-580

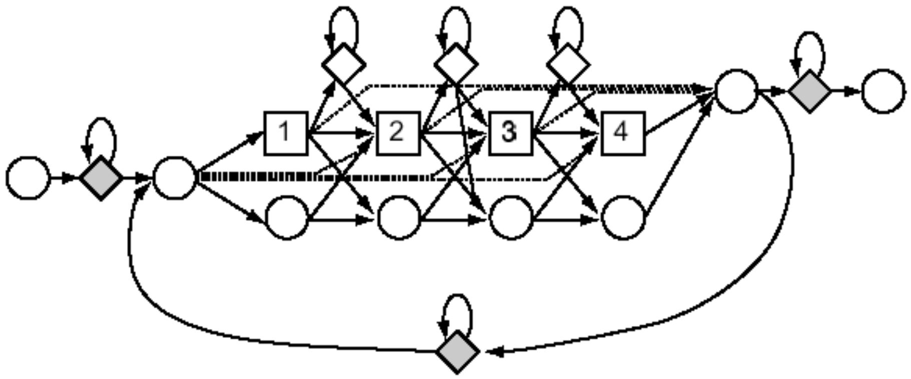
SAMTOOLS - SNP calling in NextGen sequencing data



Application of HMMs in the area of SNP discovery from NextGen sequencing data, to greatly reduce false SNP calls caused by misalignments around insertions and deletions (indels). The central concept is per-Base Alignment Quality, which accurately measures the probability of a read base being wrongly aligned.

See: Li et al. (2011) Bioinformatics 27, 1157–1158

HMMER - protein homology detection and alignment



Profile HMM architecture used in HMMER2, SAM and PFTOOLS protein homology detection and alignment packages. *Match states* carry position-specific emission probabilities for scoring residues at each consensus position. *Insert states* emit residues with emission probabilities identical to a background distribution. We will describe this in more detail shortly...

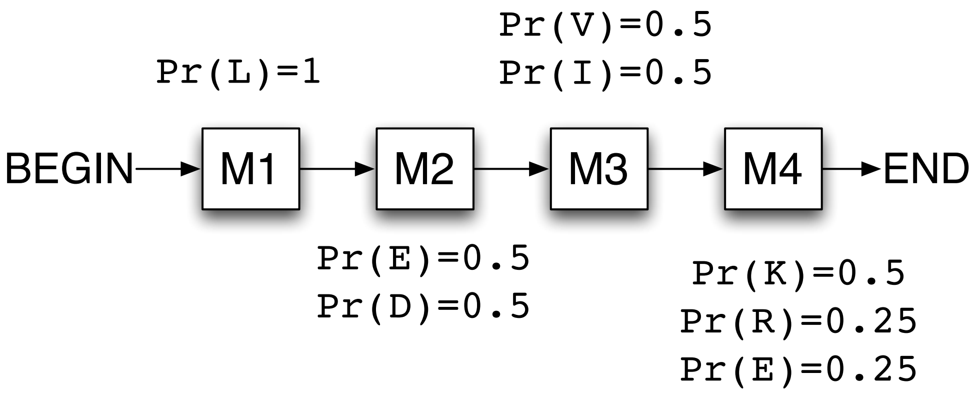
See: Eddy (1998) Bioinformatics 14, 755–763

Building sequence profile-HMMs: Match states

How do the above HMMs relate to profiles? Let’s see how we can use the HMM framework to build **profile HMMs** that describe families of related sequences.

In the last lecture, we built a profile for the alignment:

```
s1    LEVK
s2    LDIR
s3    LEIK
s4    LDVE
```



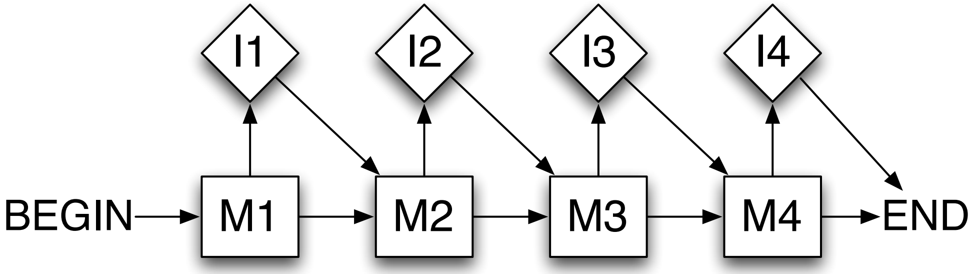
Ignoring the “background” frequencies for now, a profile for this alignment can be viewed as a simple HMM with one “match” state for each column, where consecutive match states are separated by transitions of probability 1.

Q. Why is this not a Markov chain?

Building profile-HMMs: Insert states

Introduce **insert states** (*Ij*), which will model inserts after the jth column in our alignment.

```
s1       LE-VK
s2       LD-IR
s3       LE-IK
s4       LD-VE
query1   LDAVK
```

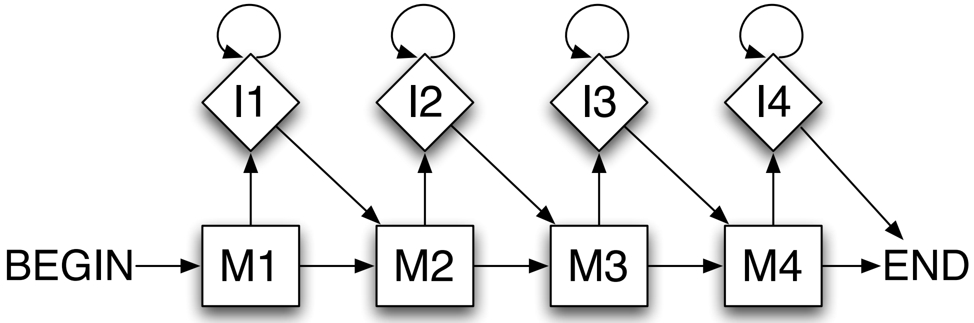


Typically, the output probabilities for insert states are set equal to the background probabilities. Note that we can have different probabilities for entering different insert states, and this models the fact that insertions may be less well-tolerated in certain portions of the alignment.

Building profile-HMMs: Insert states + *affine* gaps

For any particular insert state, we may have different transition probabilities for entering it for the first time *vs*. staying in the insert state; this models *affine* gap penalties.

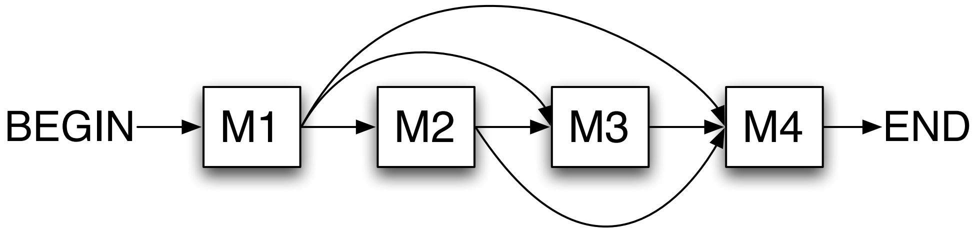
```
s1      LE---VK
s2      LD---IR
s3      LE---IK
s4      LD---VE
query1  LDA--VK
query2  LDAAAVK
```

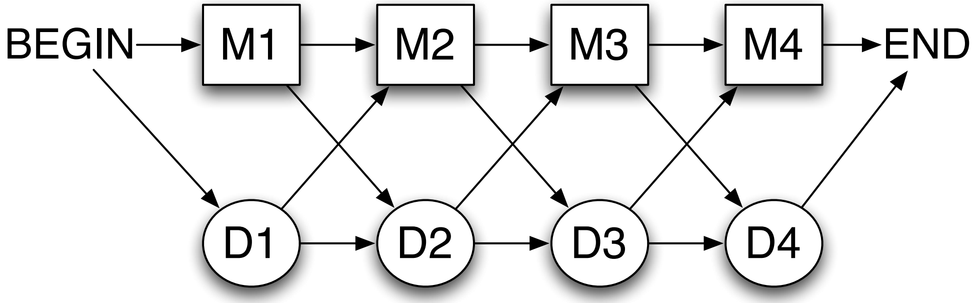


Building profile-HMMs: Delete states

One could model deletions with additional transitions between match states. However, arbitrarily long gaps would introduce lots of transitions in the model. Instead, we will introduce **delete states** that do not emit any symbols

```
s1       LEVK
s2       LDIR
s3       LEIK
s4       LDVE
query3   L-VK
```

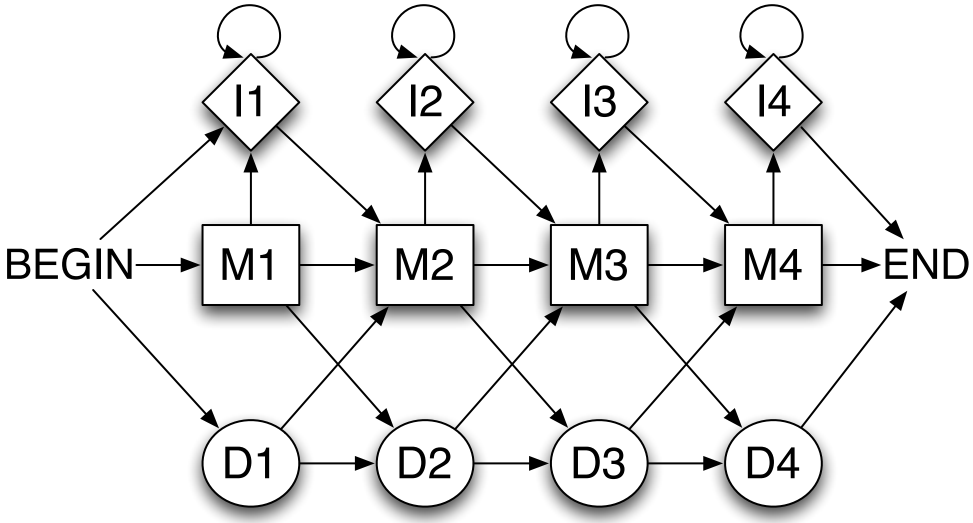




Building profile-HMMs

Putting it all together we get a complete profile HMM topology with match, insert and delete states.

```
s1        LE---VK
s2        LD---IR
s3        LE---IK
s4        LD---VE
query1    LDA--VK
query2    LDAAAVK
query3    L----VK
```



However we still need to decide how many states our HMM has, what the transition probabilities are, etc.

Example profile-HMM building

- How do we pick the length of the HMM?
  Common heuristic is to include only those columns that have > 50% occupancy

- How do we pick emission probabilities for match states?

$$b_{m1}(V) = 5/7$$
$$b_{m1}(F) = 1/7$$
$$b_{m1}(I) = 1/7$$

```
s1   VGA--NAGRPY
s2   VG---NVDKPV
s3   VGA--NVAHPH
s4   VAA------PH
s5   VGS--TYEKPS
s6   FGA--NFEKPH
s7   IGAADNGARPY
```

How do we pick transition probabilities?

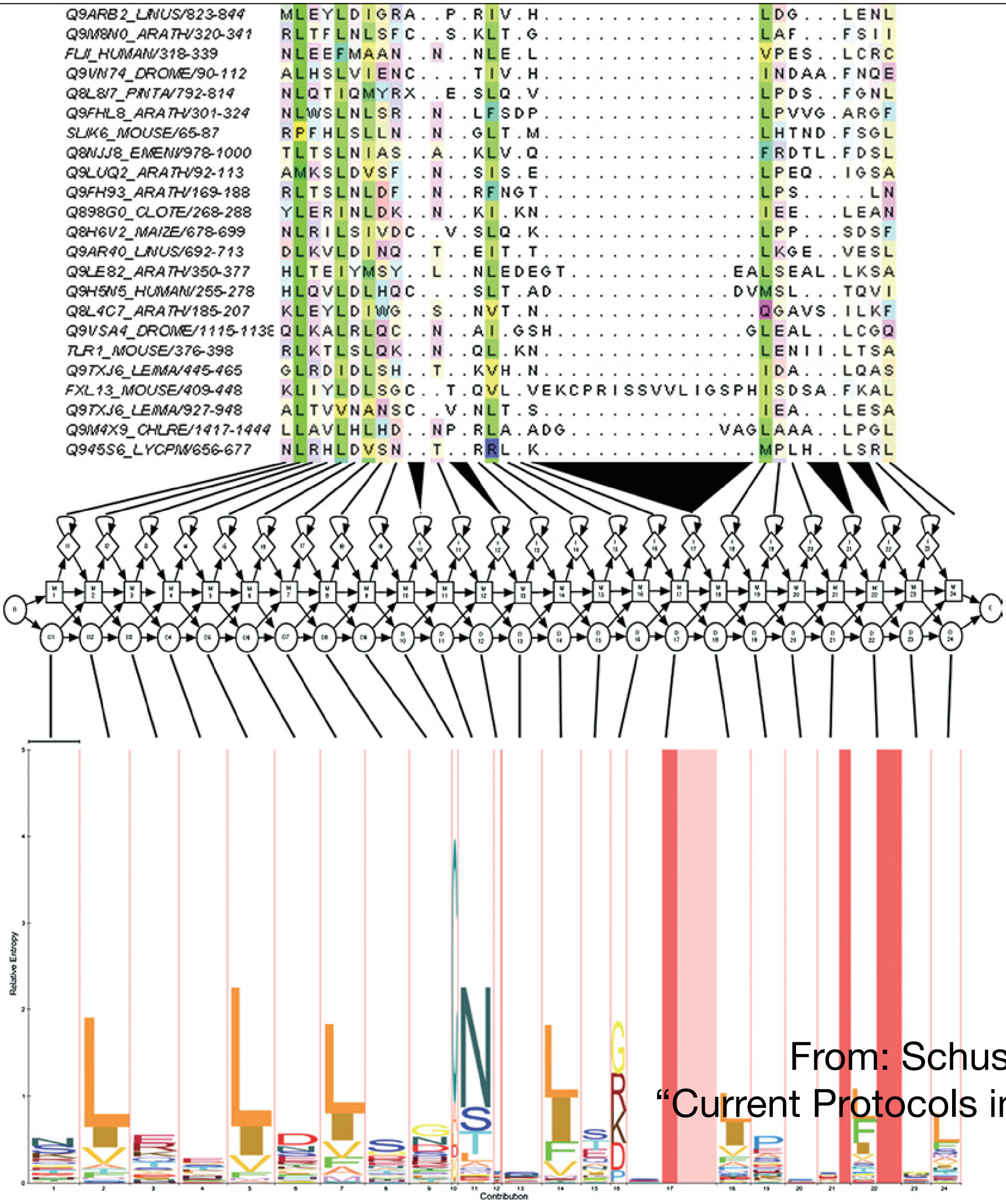
- We let the transition probability of going from state *i* to state *j*, a_{ij} be equal to:

$$\frac{\text{No. of transitions from state } i \text{ to state } j}{\text{No. of transitions from state i to any other state}}$$

$a_{M2M3}(V) = 6/7$ No. of matches (=6)

$a_{M2D3}(F) = 1/7$ No. of gaps (=1)

$a_{M2I2}(I) = 0/7$ No. of insertions (=0)



```
Q9ARB2_LINUS/823-844       MLEYLDIGRA..P.RIV.H................LDG...LENL
Q9M8N0_ARATH/320-341       RLTFLNLSFC..S.KLT.G................LAF...FSII
FLII_HUMAN/318-339         NLEEFMAAN..N..NLE.L................VPES..LCRC
Q9VN74_DROME/90-112        ALHSLVIENC....TIV.H................INDAA.FNQE
Q8L8I7_PINTA/792-814       NLQTIQMYRX..E.SLQ.V................LPDS..FGNL
Q9FHL8_ARATH/301-324       NLWSLNLSR..N..LFSDP................LPVVG.ARGF
SLIK6_MOUSE/65-87          RPFHLSLLN..N..GLT.M................LHTND.FSGL
Q8NJJ8_EMENI/978-1000      TLTSLNIAS..A..KLV.Q................FRDTL.FDSL
Q9LUQ2_ARATH/92-113        AMKSLDVSF..N..SIS.E................LPEQ..IGSA
Q9FH93_ARATH/169-188       RLTSLNLDF..N..RFNGT................LPS.....LN
Q898G0_CLOTE/268-288       YLERINLDK..N..KI.KN................IEE...LEAN
Q8H6V2_MAIZE/678-699       NLRILSIVDC..V.SLQ.K................LPP...SDSF
Q9AR40_LINUS/692-713       DLKVLDINQ..T..EIT.T................LKGE..VESL
Q9LE82_ARATH/350-377       HLTEIYMSY..L..NLEDEGT............EALSEAL.LKSA
Q9H5N5_HUMAN/255-278       HLQVLDLHQC....SLT.AD.............DVMSL...TQVI
Q8L4C7_ARATH/185-207       KLEYLDIWG..S..NVT.N................QGAVS.ILKF
Q9VSA4_DROME/1115-113E     QLKALRLQC..N..AI.GSH..............GLEAL..LCGQ
TLR1_MOUSE/376-398         RLKTLSLQK..N..QL.KN................LENII.LTSA
Q9TXJ6_LEIMA/445-465       GLRDIDLSH..T..KVH.N................IDA...LQAS
FXL13_MOUSE/409-448        KLIYLDLSGC..T.QVL.VEKCPRISSVVLIGSPHISDSA.FKAL
Q9TXJ6_LEIMA/927-948       ALTVVNANSC..V.NLT.S................IEA...LESA
Q9M4X9_CHLRE/1417-1444     LLAVLHLHD..NP.RLA.ADG...........VAGLAAA..LPGL
Q945S6_LYCPM/656-677       NLRHLDVSN..T..RRL.K................MPLH..LSRL
```

From: Schuster-Bockler *et al.* “Current Protocols in Bioinformatics” Supplement 18.

Side note: Weighting the training sequences

If there is a high degree of **redundancy** in our initial MSA (i.e. it contains a large group of very closely related sequences and a small number of more distantly related sequences) the resulting HMM will over represent the similar sequences and adversely effect our ability to detect distantly related sequences when searching databases

Sequences weighting attempts to compensate for this *sequence sampling bias* by differentially weighting sequences to reduce redundancy prior to model building

By default HMMER uses a sequence clustering tree as a guide to weight each sequence by its distance to other sequences. This approach will effectively down-weight the influence of redundant sequences.

A number of other approaches have been developed (Voronoi algorithm, maximum entropy, etc.)

See: Karchin *et al.* (1998) Bioinformatics 14, 772-778

Side note: Pseudocounts and Dirichlet distributions

Unfortunately, for alignments containing a small number of sequences the observed counts may not be representative of the family as a whole.

In such cases we must adjust the probabilities to account for our under-sampling (i.e. unobserved residues)

One common approach is to add **pseudocounts** to the observed counts so that no zero probabilities can occur.

Simplest approach is to just add one to all counts. More accurate adjustments consider prior knowledge about the behavior of sequence families adjusting counts according to pre-tabulated **Dirichlet distributions** - which are rather like protein comparison matrixes used in profile methods

Such information is often called **prior information**, indicating that it is known before any sequence data is seen

See: Durbin et al. “Biological Sequence Analysis”

Generating multiple sequence alignments

Large MSAs can be generated very quickly by using the Viterbi algorithm to find the most likely path through the HMM for a set of unaligned sequences

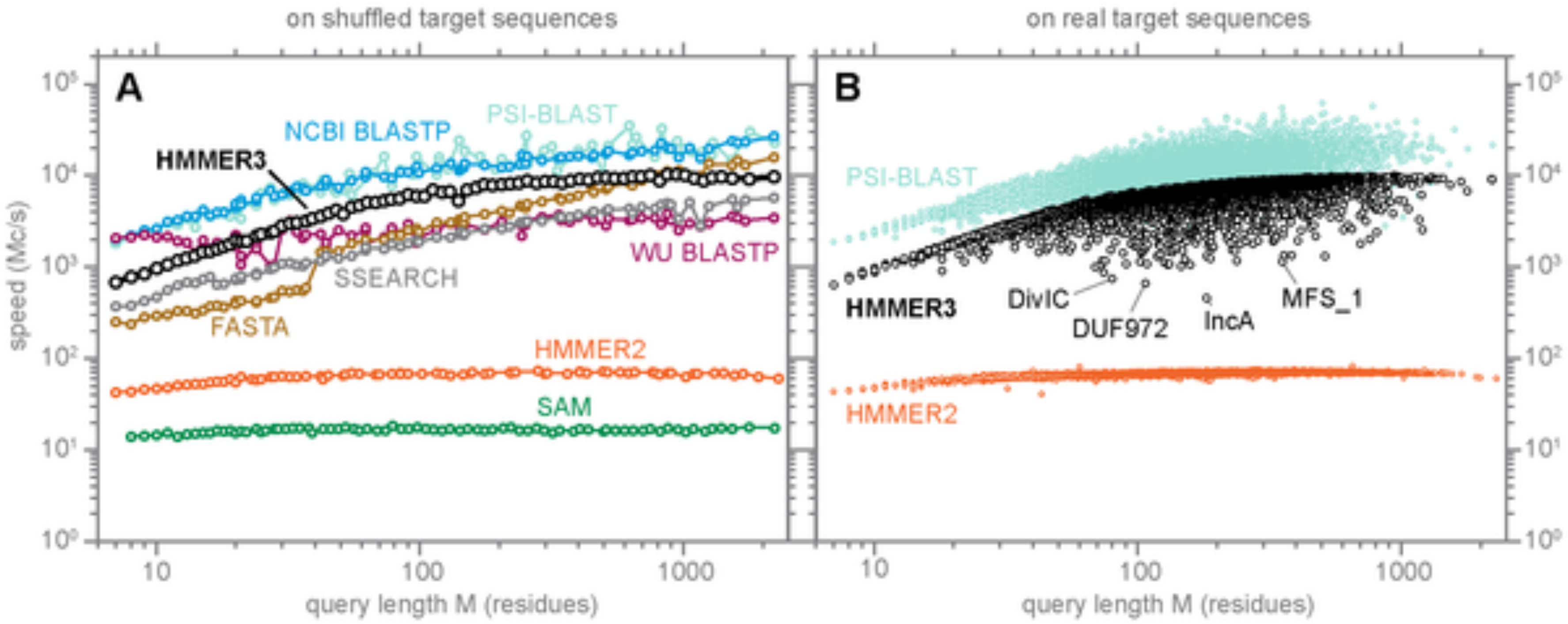
This is the basis of the PFAM database which uses the HMMER software package
Namely, HAMMER's *hmmalign* from the results of *hmmsearch*

MSA produced by HMMs are not true MSAs in the way that those produced by ClustalW are. ClustalW compares every sequence to every other sequence, whereas HMM aligning compares every sequence to the model independently so that the alignment between sequences is by proxy. Adding new sequences to the ClustalW alignment will add new information which may alter the alignment of existing sequences; adding new sequences to the HMM alignment never changes the alignment of any sequences relative to each other.

As an alternative to HMMER, you can use the *Sequence Alignment and Modeling Software System* (SAM)

http://compbio.soe.ucsc.edu/sam.html

HMM sequence searching performance

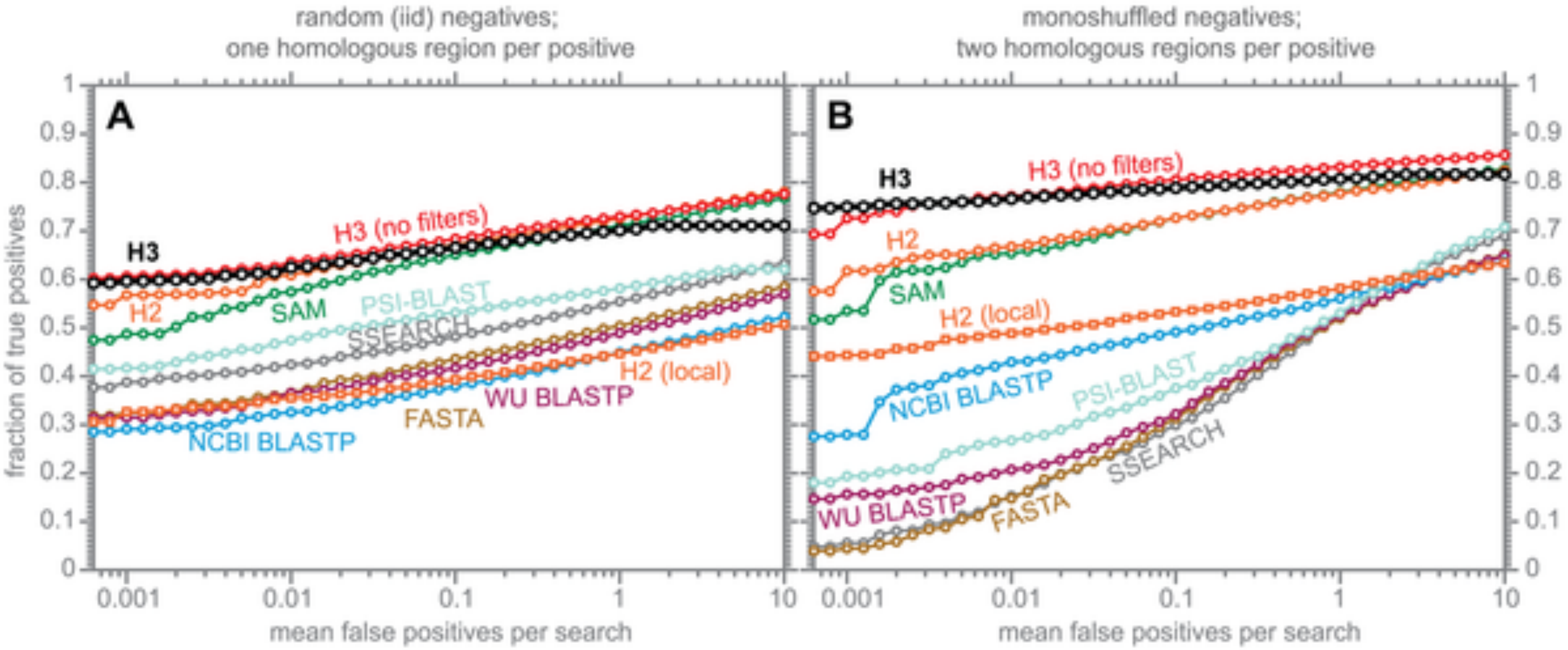


Recent speed benchmarks indicate that HMMER3 is approaching BLAST speed

Each point represents a speed measurement for one search with one query against target sequences. Both axes are logarithmic, for speed in millions of dynamic programming cells per second (Mc/s) on the y-axis and query length on the x-axis.

See: Eddy (2011) PLoS Comp Biol 7(10): e1002195

HMM sequence searching performance...



However HMMER3 has a much higher search sensitivity and specificity

In each benchmark, true positive subsequences have been selected to be no more than 25% identical to any sequence in the query alignment ... (see paper for details).

See: Eddy (2011) PLoS Comp Biol 7(10): e1002195

HMM limitations

HMMs are linear models and are thus **unable to capture higher order correlations** among positions (e.g. distant cysteins in a disulfide bridge, RNA secondary structure pairs, etc).

Another flaw of HMMs lies at the very heart of the mathematical theory behind these models. Namely, that the probability of a sequence can be found from the product of the probabilities of its individual residues.

This claim is only valid if the probability of a residue is independent of the probabilities of its neighbors. In biology, there are frequently **strong dependencies between these probabilities** (e.g. hydrophobic residues clustering at the core of protein domains).

These biological realities have motivated research into new kinds of statistical models. These include hybrids of HMMs and neural nets, dynamic Bayesian nets, factorial HMMs, Boltzmann trees and stochastic context-free grammars.

See: Durbin et al. “Biological Sequence Analysis”

PFAM: Protein Family Database of Profile HMMs

Comprehensive compilation of both multiple sequence alignments and profile HMMs of protein families.

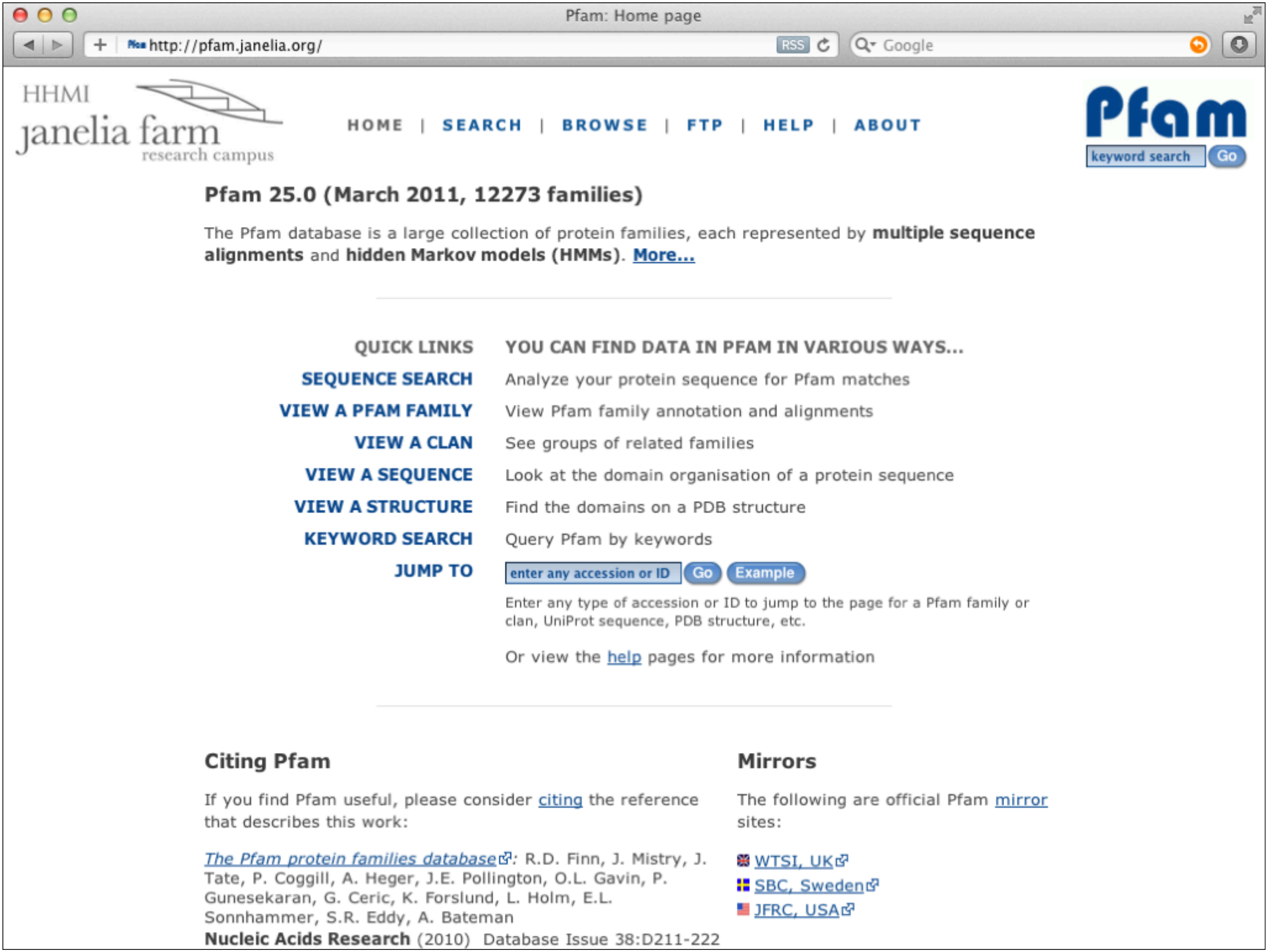
http://pfam.sanger.ac.uk/

PFAM consists of two databases:

- **Pfam-A** is a manually curated collection of protein families in the form of multiple sequence alignments and profile HMMs. HMMER software is used to perform searches.

- **Pfam-B** contains additional protein sequences that are automatically aligned. Pfam-B serves as a useful supplement that makes the database more comprehensive.

- Pfam-A also contains higher-level groupings of related families, known as **clans**



Pfam 25.0 (March 2011, 12273 families)

The Pfam database is a large collection of protein families, each represented by **multiple sequence alignments** and **hidden Markov models (HMMs)**. More...

QUICK LINKS	YOU CAN FIND DATA IN PFAM IN VARIOUS WAYS...
SEQUENCE SEARCH	Analyze your protein sequence for Pfam matches
VIEW A PFAM FAMILY	View Pfam family annotation and alignments
VIEW A CLAN	See groups of related families
VIEW A SEQUENCE	Look at the domain organisation of a protein sequence
VIEW A STRUCTURE	Find the domains on a PDB structure
KEYWORD SEARCH	Query Pfam by keywords
JUMP TO	enter any accession or ID Go Example

Enter any type of accession or ID to jump to the page for a Pfam family or clan, UniProt sequence, PDB structure, etc.

Or view the help pages for more information

Citing Pfam

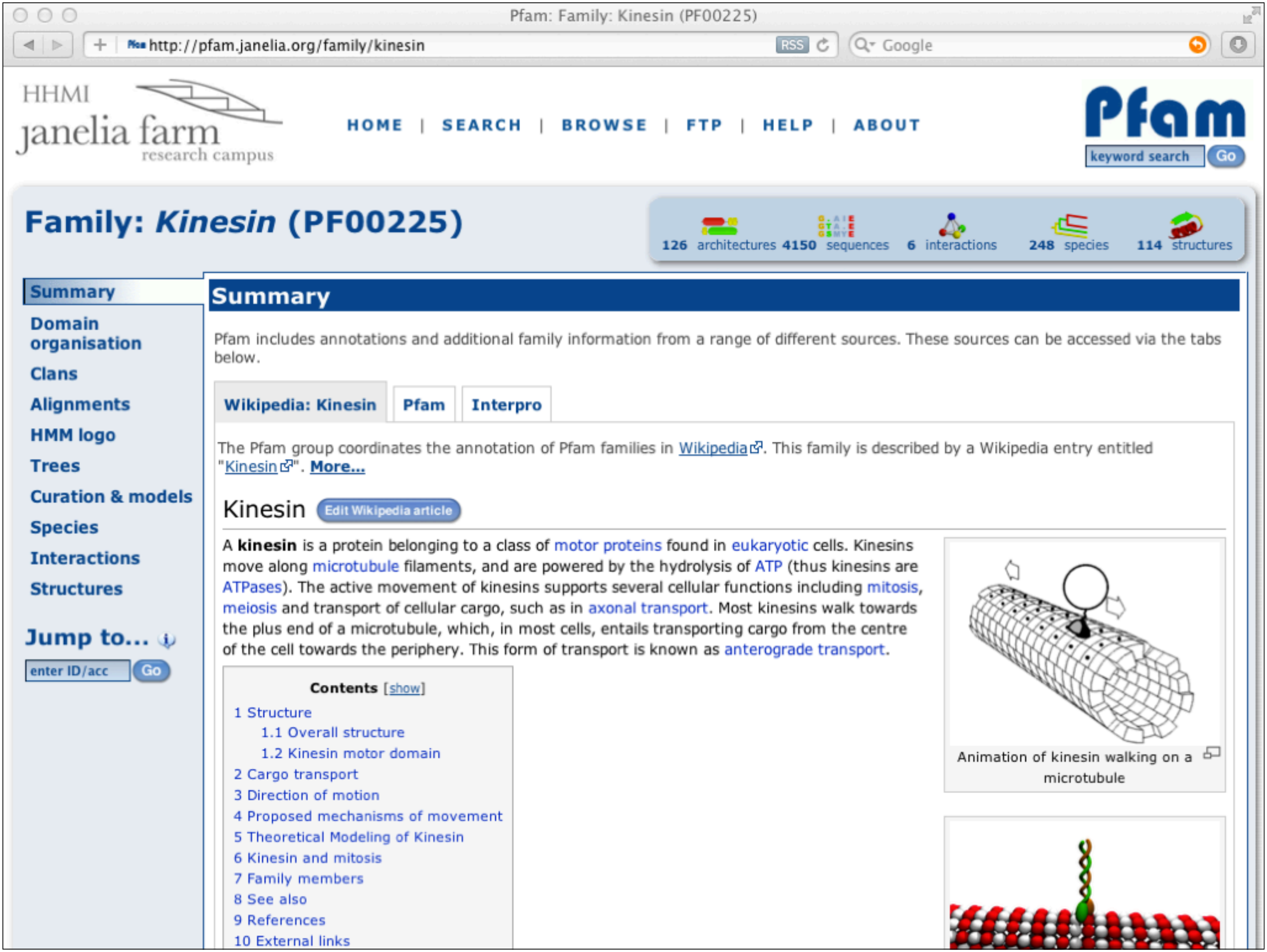
If you find Pfam useful, please consider citing the reference that describes this work:

The Pfam protein families database: R.D. Finn, J. Mistry, J. Tate, P. Coggill, A. Heger, J.E. Pollington, O.L. Gavin, P. Gunesekaran, G. Ceric, K. Forslund, L. Holm, E.L. Sonnhammer, S.R. Eddy, A. Bateman **Nucleic Acids Research** (2010) Database Issue 38:D211-222

Mirrors

The following are official Pfam mirror sites:

- WTSI, UK
- SBC, Sweden
- JFRC, USA



Family: *Kinesin* (PF00225)

Summary

Pfam includes annotations and additional family information from a range of different sources. These sources can be accessed via the tabs below.

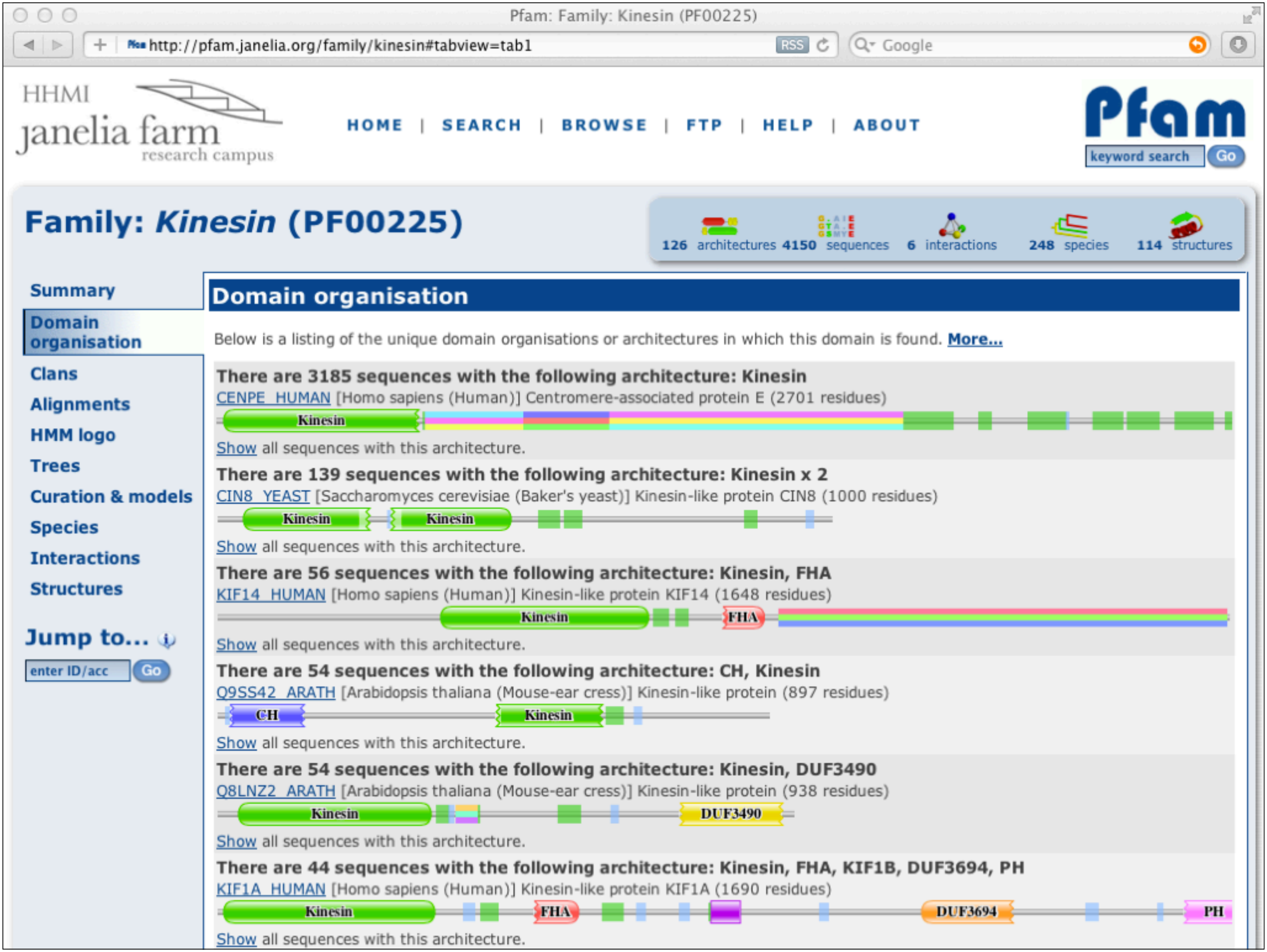
The Pfam group coordinates the annotation of Pfam families in Wikipedia. This family is described by a Wikipedia entry entitled "Kinesin". More...

Kinesin

A **kinesin** is a protein belonging to a class of motor proteins found in eukaryotic cells. Kinesins move along microtubule filaments, and are powered by the hydrolysis of ATP (thus kinesins are ATPases). The active movement of kinesins supports several cellular functions including mitosis, meiosis and transport of cellular cargo, such as in axonal transport. Most kinesins walk towards the plus end of a microtubule, which, in most cells, entails transporting cargo from the centre of the cell towards the periphery. This form of transport is known as anterograde transport.

Animation of kinesin walking on a microtubule

Contents [show]



Pfam: Family: Kinesin (PF00225)

http://pfam.janelia.org/family/kinesin#tabview=tab1

HOME | SEARCH | BROWSE | FTP | HELP | ABOUT

Family: *Kinesin* (PF00225)

126 architectures 4150 sequences 6 interactions 248 species 114 structures

- Summary
- Domain organisation
- Clans
- Alignments
- HMM logo
- Trees
- Curation & models
- Species
- Interactions
- Structures

Jump to...

Domain organisation

Below is a listing of the unique domain organisations or architectures in which this domain is found. More...

There are 3185 sequences with the following architecture: Kinesin

CENPE_HUMAN [Homo sapiens (Human)] Centromere-associated protein E (2701 residues)

Show all sequences with this architecture.

There are 139 sequences with the following architecture: Kinesin x 2

CIN8_YEAST [Saccharomyces cerevisiae (Baker's yeast)] Kinesin-like protein CIN8 (1000 residues)

Show all sequences with this architecture.

There are 56 sequences with the following architecture: Kinesin, FHA

KIF14_HUMAN [Homo sapiens (Human)] Kinesin-like protein KIF14 (1648 residues)

Show all sequences with this architecture.

There are 54 sequences with the following architecture: CH, Kinesin

Q9SS42_ARATH [Arabidopsis thaliana (Mouse-ear cress)] Kinesin-like protein (897 residues)

Show all sequences with this architecture.

There are 54 sequences with the following architecture: Kinesin, DUF3490

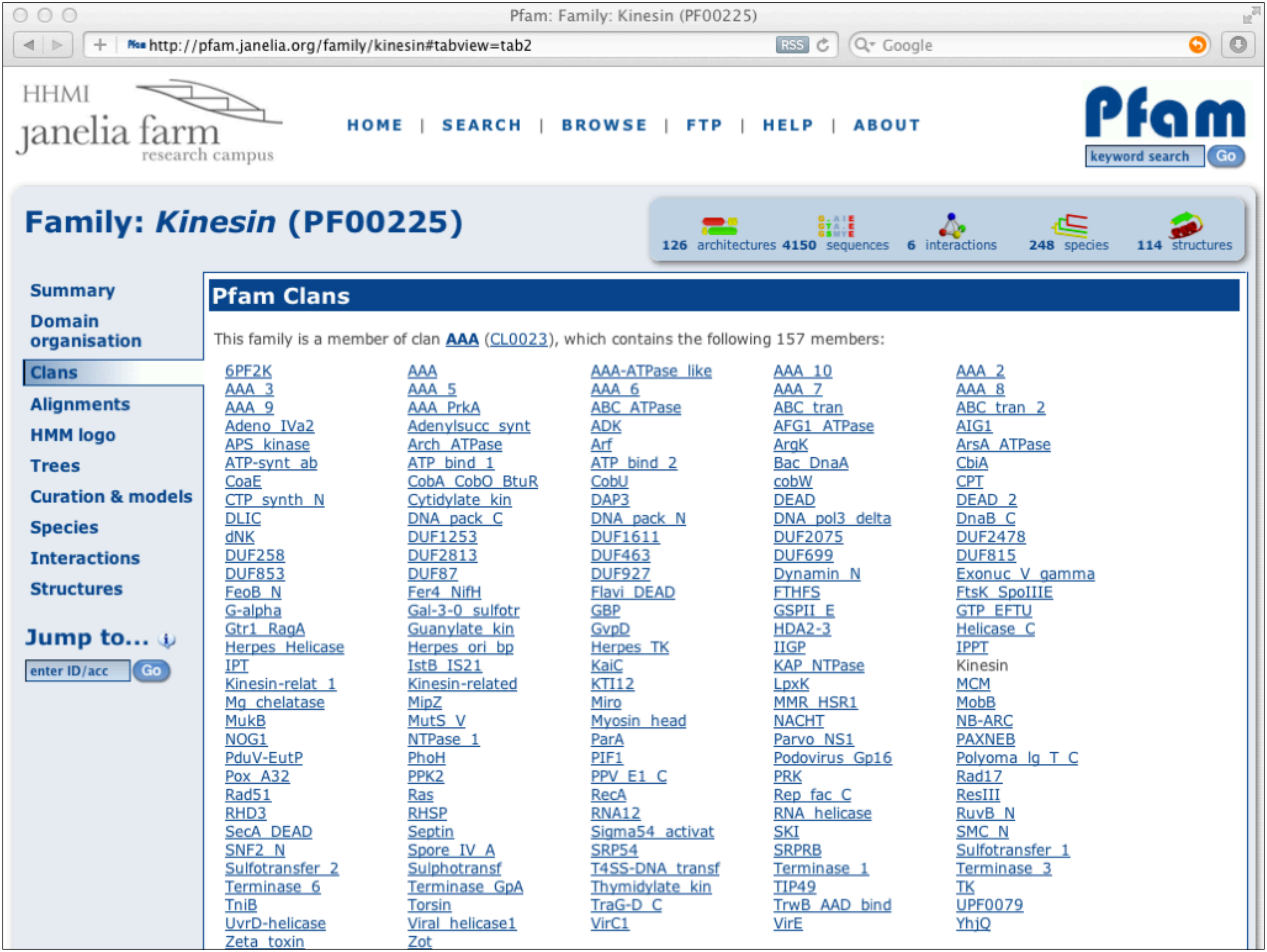
Q8LNZ2_ARATH [Arabidopsis thaliana (Mouse-ear cress)] Kinesin-like protein (938 residues)

Show all sequences with this architecture.

There are 44 sequences with the following architecture: Kinesin, FHA, KIF1B, DUF3694, PH

KIF1A_HUMAN [Homo sapiens (Human)] Kinesin-like protein KIF1A (1690 residues)

Show all sequences with this architecture.



Pfam: Family: Kinesin (PF00225)

http://pfam.janelia.org/family/kinesin#tabview=tab2

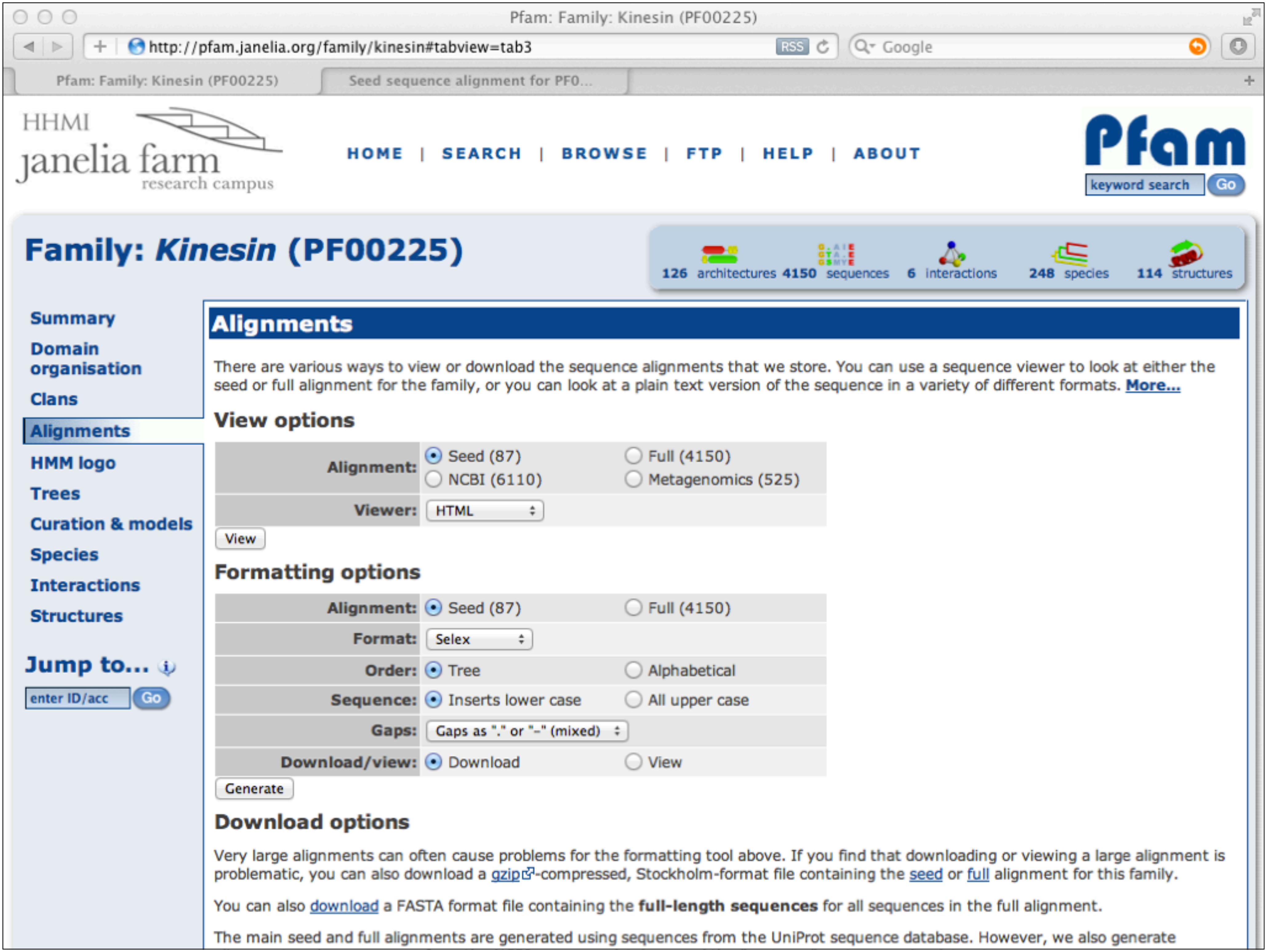
Family: *Kinesin* (PF00225)

126 architectures 4150 sequences 6 interactions 248 species 114 structures

Pfam Clans

This family is a member of clan **AAA** (CL0023), which contains the following 157 members:

6PF2K	AAA	AAA-ATPase_like	AAA_10	AAA_2
AAA_3	AAA_5	AAA_6	AAA_7	AAA_8
AAA_9	AAA_PrkA	ABC_ATPase	ABC_tran	ABC_tran_2
Adeno_IVa2	Adenylsucc_synt	ADK	AFG1_ATPase	AIG1
APS_kinase	Arch_ATPase	Arf	ArgK	ArsA_ATPase
ATP-synt_ab	ATP_bind_1	ATP_bind_2	Bac_DnaA	CbiA
CoaE	CobA_CobO_BtuR	CobU	cobW	CPT
CTP_synth_N	Cytidylate_kin	DAP3	DEAD	DEAD_2
DLIC	DNA_pack_C	DNA_pack_N	DNA_pol3_delta	DnaB_C
dNK	DUF1253	DUF1611	DUF2075	DUF2478
DUF258	DUF2813	DUF463	DUF699	DUF815
DUF853	DUF87	DUF927	Dynamin_N	Exonuc_V_gamma
FeoB_N	Fer4_NifH	Flavi_DEAD	FTHFS	FtsK_SpoIIIE
G-alpha	Gal-3-0_sulfotr	GBP	GSPII_E	GTP_EFTU
Gtr1_RagA	Guanylate_kin	GvpD	HDA2-3	Helicase_C
Herpes_Helicase	Herpes_ori_bp	Herpes_TK	IIGP	IPPT
IPT	IstB_IS21	KaiC	KAP_NTPase	Kinesin
Kinesin-relat_1	Kinesin-related	KTI12	LpxK	MCM
Mg_chelatase	MipZ	Miro	MMR_HSR1	MobB
MukB	MutS_V	Myosin_head	NACHT	NB-ARC
NOG1	NTPase_1	ParA	Parvo_NS1	PAXNEB
PduV-EutP	PhoH	PIF1	Podovirus_Gp16	Polyoma_lg_T_C
Pox_A32	PPK2	PPV_E1_C	PRK	Rad17
Rad51	Ras	RecA	Rep_fac_C	ResIII
RHD3	RHSP	RNA12	RNA_helicase	RuvB_N
SecA_DEAD	Septin	Sigma54_activat	SKI	SMC_N
SNF2_N	Spore_IV_A	SRP54	SRPRB	Sulfotransfer_1
Sulfotransfer_2	Sulphotransf	T4SS-DNA_transf	Terminase_1	Terminase_3
Terminase_6	Terminase_GpA	Thymidylate_kin	TIP49	TK
TniB	Torsin	TraG-D_C	TrwB_AAD_bind	UPF0079
UvrD-helicase	Viral_helicase1	VirC1	VirE	YhjQ
Zeta_toxin	Zot			



Pfam: Family: Kinesin (PF00225)
http://pfam.janelia.org/family/kinesin#tabview=tab3
RSS
Google
Pfam: Family: Kinesin (PF00225)
Seed sequence alignment for PF0...
HHMI
janelia farm
research campus
HOME | SEARCH | BROWSE | FTP | HELP | ABOUT
Pfam
keyword search
Go

Family: *Kinesin* (PF00225)

126 architectures 4150 sequences 6 interactions 248 species 114 structures
Summary
Domain organisation
Clans
Alignments
HMM logo
Trees
Curation & models
Species
Interactions
Structures
Jump to...
enter ID/acc
Go

Alignments

There are various ways to view or download the sequence alignments that we store. You can use a sequence viewer to look at either the seed or full alignment for the family, or you can look at a plain text version of the sequence in a variety of different formats. More...

View options

Alignment:	● Seed (87) ○ NCBI (6110)	○ Full (4150) ○ Metagenomics (525)
Viewer:	HTML	

View

Formatting options

Alignment:	● Seed (87)	○ Full (4150)
Format:	Selex	
Order:	● Tree	○ Alphabetical
Sequence:	● Inserts lower case	○ All upper case
Gaps:	Gaps as "." or "-" (mixed)	
Download/view:	● Download	○ View

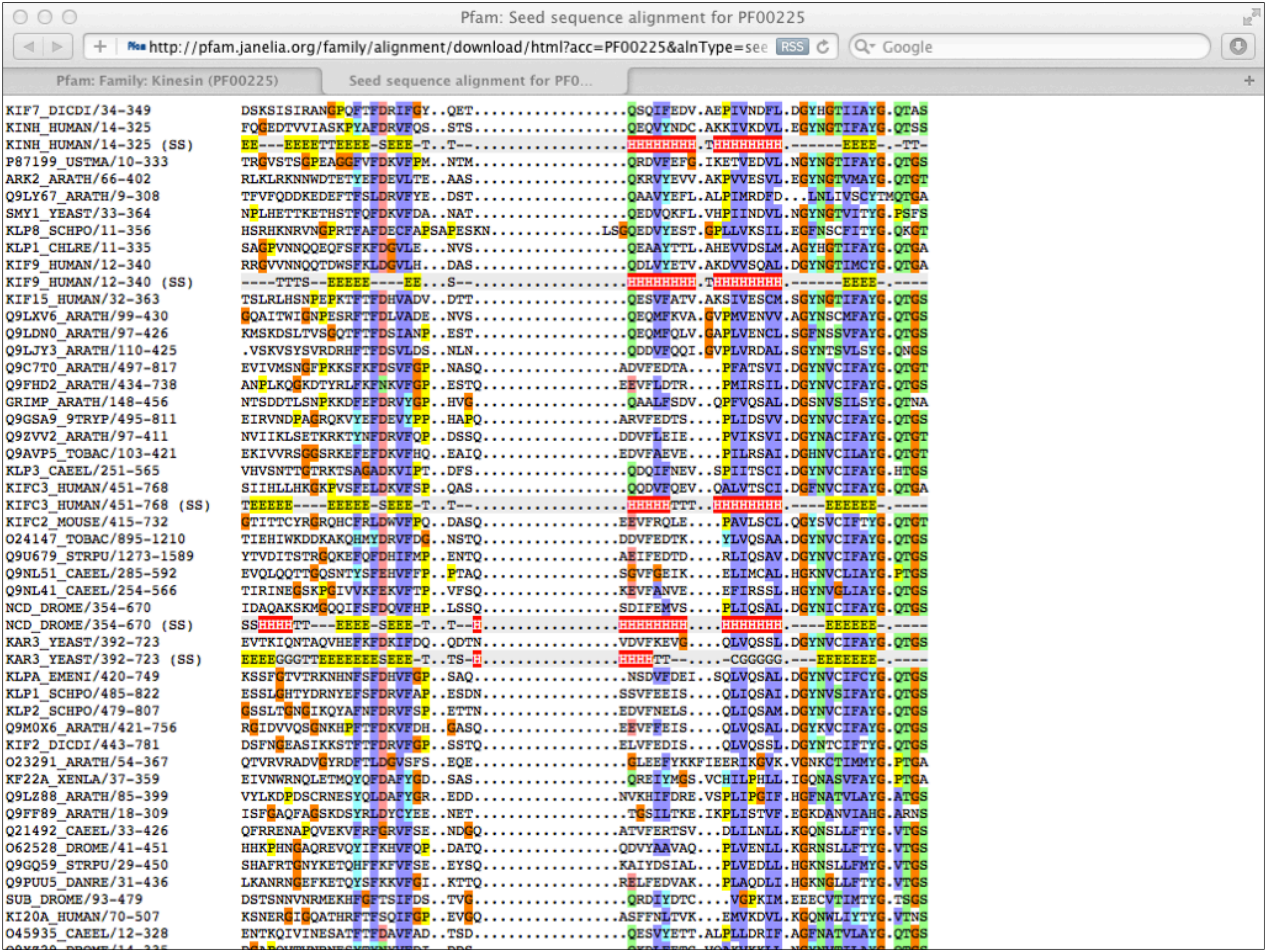
Generate

Download options

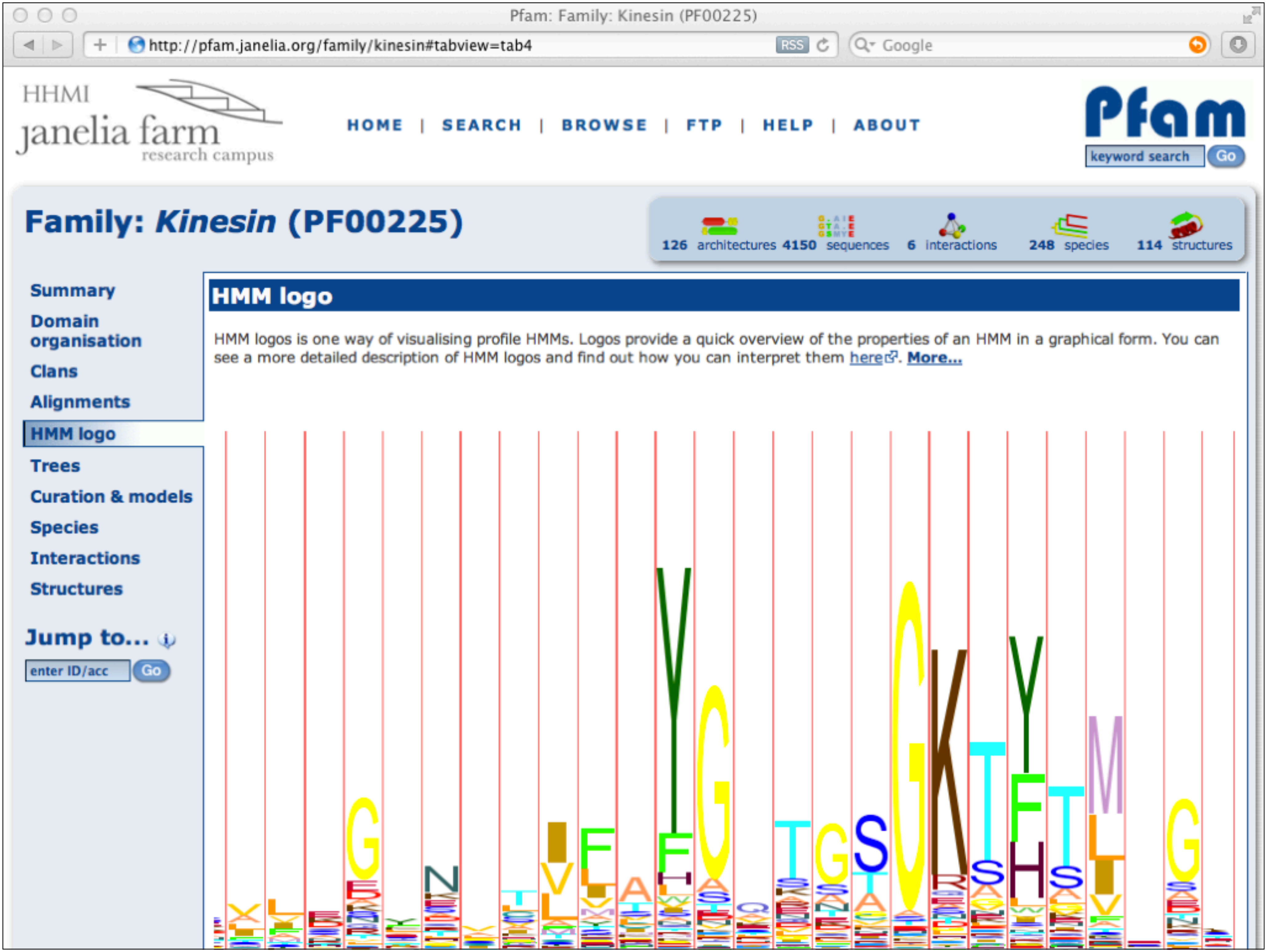
Very large alignments can often cause problems for the formatting tool above. If you find that downloading or viewing a large alignment is problematic, you can also download a gzip-compressed, Stockholm-format file containing the seed or full alignment for this family.

You can also download a FASTA format file containing the **full-length sequences** for all sequences in the full alignment.

The main seed and full alignments are generated using sequences from the UniProt sequence database. However, we also generate


Pfam: Seed sequence alignment for PF00225
http://pfam.janelia.org/family/alignment/download/html?acc=PF00225&alnType=see
RSS
Google
Pfam: Family: Kinesin (PF00225)
Seed sequence alignment for PF0...

```
KIF7_DICDI/34-349          DSKSISIRANGPQFTFDRIFGY..QET..................QSQIFEDV.AEPIVNDFL.DGYHGTIIAYG.QTAS
KINH_HUMAN/14-325          FQGEDTVVIASKPYAFDRVFQS..STS..................QEQVYNDC.AKKIVKDVL.EGYNGTIFAYG.QTSS
KINH_HUMAN/14-325 (SS)     EE---EEEETTEEEE-SEEE-T..T--..................HHHHHHHH.THHHHHHHH.------EEEE-.-TT-
P87199_USTMA/10-333        TRGVSTSGPEAGGFVFDKVFPM..NTM..................QRDVFEFG.IKETVEDVL.NGYNGTIFAYG.QTGS
ARK2_ARATH/66-402          RLKLRKNNWDTETYEFDEVLTE..AAS..................QKRVYEVV.AKPVVESVL.EGYNGTVMAYG.QTGT
Q9LY67_ARATH/9-308         TFVFQDDKEDEFTFSLDRVFYE..DST..................QAAVYEFL.ALPIMRDFD...LNLIVSCYTMQTGA
SMY1_YEAST/33-364          NPLHETTKETHSTFQFDKVFDA..NAT..................QEDVQKFL.VHPIINDVL.NGYNGTVITYG.PSFS
KLP8_SCHPO/11-356          HSRHKNRVNGPRTFAFDECFAPSAPESKN.............LSGQEDVYEST.GPLLVKSIL.EGFNSCFITYG.QKGT
KLP1_CHLRE/11-335          SAGPVNNQQEQFSFKFDGVLE...NVS..................QEAAYTTL.AHEVVDSLM.AGYHGTIFAYG.QTGA
KIF9_HUMAN/12-340          RRGVVNNQQTDWSFKLDGVLH...DAS..................QDLVYETV.AKDVVSQAL.DGYNGTIMCYG.QTGA
KIF9_HUMAN/12-340 (SS)     ----TTTS--EEEEE----EE...S--..................HHHHHHHH.THHHHHHHH.------EEEE-.----
KIF15_HUMAN/32-363         TSLRLHSNPEPKTFTFDHVADV..DTT..................QESVFATV.AKSIVESCM.SGYNGTIFAYG.QTGS
Q9LXV6_ARATH/99-430        GQAITWIGNPESRFTFDLVADE..NVS..................QEQMFKVA.GVPMVENVV.AGYNSCMFAYG.QTGS
Q9LDN0_ARATH/97-426        KMSKDSLTVSGQTFTFDSIANP..EST..................QEQMFQLV.GAPLVENCL.SGFNSSVFAYG.QTGS
Q9LJY3_ARATH/110-425       .VSKVSYSVRDRHFTFDSVLDS..NLN..................QDDVFQQI.GVPLVRDAL.SGYNTSVLSYG.QNGS
Q9C7T0_ARATH/497-817       EVIVMSNGFPKKSFKFDSVFGP..NASQ................ADVFEDTA....PFATSVI.DGYNVCIFAYG.QTGT
Q9FHD2_ARATH/434-738       ANPLKQGKDTYRLFKFNKVFGP..ESTQ................EEVFLDTR....PMIRSIL.DGYNVCIFAYG.QTGS
GRIMP_ARATH/148-456        NTSDDTLSNPKKDFEFDRVYGP..HVG..................QAALFSDV..QPFVQSAL.DGSNVSILSYG.QTNA
Q9GSA9_9TRYP/495-811       EIRVNDPAGRQKVYEFDEVYPP..HAPQ................ARVFEDTS....PLIDSVV.DGYNVCIFAYG.QTGS
Q9ZVV2_ARATH/97-411        NVIIKLSETKRKTYNFDRVFQP..DSSQ................DDVFLEIE....PVIKSVI.DGYNACIFAYG.QTGT
Q9AVP5_TOBAC/103-421       EKIVVRSGGSRKEFEFDKVFHQ..EAIQ................EDVFAEVE....PILRSAI.DGHNVCILAYG.QTGT
KLP3_CAEEL/251-565         VHVSNTTGTRKTSAGADKVIPT..DFS..................QDQIFNEV..SPIITSCI.DGYNVCIFAYG.HTGS
KIFC3_HUMAN/451-768        SIIHLLHKGKPVSFELDKVFSP..QAS..................QQDVFQEV..QALVTSCI.DGFNVCIFAYG.QTGA
KIFC3_HUMAN/451-768 (SS)   TEEEEE----EEEEE-SEEE-T..T--..................HHHHHTTT..HHHHHHHH.----EEEEEE-.----
KIFC2_MOUSE/415-732        GTITTCYRGRQHCFRLDWVFPQ..DASQ................EEVFRQLE....PAVLSCL.QGYSVCIFTYG.QTGT
O24147_TOBAC/895-1210      TIEHIWKDDKAKQHMYDRVFDG..NSTQ................DDVFEDTK....YLVQSAA.DGYNVCIFAYG.QTGS
Q9U679_STRPU/1273-1589     YTVDITSTRGQKEFQFDHIFMP..ENTQ................AEIFEDTD....RLIQSAV.DGYNVCIFAYG.QTGS
Q9NL51_CAEEL/285-592       EVQLQQTTGQSNTYSFEHVFFP..PTAQ................SGVFGEIK....ELIMCAL.HGKNVCLIAYG.PTGS
Q9NL41_CAEEL/254-566       TIRINEGSKPGIVVKFEKVFTP..VFSQ................KEVFANVE....EFIRSSL.HGYNVGLIAYG.QTGS
NCD_DROME/354-670          IDAQAKSKMGQQIFSFDQVFHP..LSSQ................SDIFEMVS....PLIQSAL.DGYNICIFAYG.QTGS
NCD_DROME/354-670 (SS)     SSHHHHTT---EEEE-SEEE-T..T--H................HHHHHHHHH...HHHHHHH.----EEEEEE-.----
KAR3_YEAST/392-723         EVTKIQNTAQVHEFKFDKIFDQ..QDTN................VDVFKEVG....QLVQSSL.DGYNVCIFAYG.QTGS
KAR3_YEAST/392-723 (SS)    EEEEGGGTTEEEEEEESEEE-T..TS-H................HHHHTT--....-CGGGGG.---EEEEEEE-.----
KLPA_EMENI/420-749         KSSFGTVTRKNHNFSFDHVFGP..SAQ..................NSDVFDEI..SQLVQSAL.DGYNVCIFCYG.QTGS
KLP1_SCHPO/485-822         ESSLGHTYDRNYEFSFDRVFAP..ESDN................SSVFEEIS....QLIQSAI.DGYNVSIFAYG.QTGS
KLP2_SCHPO/479-807         GSSLTGNGIKQYAFNFDRVFSP..ETTN................EDVFNELS....QLIQSAM.DGYNVCIFAYG.QTGS
Q9M0X6_ARATH/421-756       RGIDVVQSGNKHPFTFDKVFDH..GASQ................EEVFFEIS....QLVQSAL.DGYKVCIFAYG.QTGS
KIF2_DICDI/443-781         DSFNGEASIKKSTFTFDRVFGP..SSTQ................ELVFEDIS....QLVQSSL.DGYNTCIFTYG.QTGS
O23291_ARATH/54-367        QTVRVRADVGYRDFTLDGVSFS..EQE..................GLEEFYKKFIEERIKGVK.VGNKCTIMMYG.PTGA
KF22A_XENLA/37-359         EIVNWRNQLETMQYQFDAFYGD..SAS..................QREIYMGS.VCHILPHLL.IGQNASVFAYG.PTGA
Q9LZ88_ARATH/85-399        VYLKDPDSCRNESYQLDAFYGR..EDD.................NVKHIFDRE.VSPLIPGIF.HGFNATVLAYG.ATGS
Q9FF89_ARATH/18-309        ISFGAQFAGSKDSYRLDYCYEE..NET..................TGSILTKE.IKPLISTVF.EGKDANVIAHG.ARNS
Q21492_CAEEL/33-426        QFRRENAPQVEKVFRFGRVFSE..NDGQ................ATVFERTSV...DLILNLL.KGQNSLLFTYG.VTGS
O62528_DROME/41-451        HHKPHNGAQREVQYIFKHVFQP..DATQ................QDVYAAVAQ...PLVENLL.KGRNSLLFTYG.VTGS
Q9GQ59_STRPU/29-450        SHAFRTGNYKETQHFFKFVFSE..EYSQ................KAIYDSIAL...PLVEDLL.HGKNSLLFMYG.VTGS
Q9PUU5_DANRE/31-436        LKANRNGEFKETQYSFKKVFGI..KTTQ................RELFEDVAK...PLAQDLI.HGKNGLLFTYG.VTGS
SUB_DROME/93-479           DSTSNNVNRMEKHFGFTSIFDS..TVG..................QRDIYDTC....VGPKIM.EEECVTIMTYG.TSGS
KI20A_HUMAN/70-507         KSNERGIGQATHRFTFSQIFGP..EVGQ................ASFFNLTVK...EMVKDVL.KGQNWLIYTYG.VTNS
O45935_CAEEL/12-328        ENTKQIVINESATFTFDAVFAD..TSD..................QESVYETT.ALPLLDRIF.AGFNATVLAYG.QTGS
```



Pfam: Family: Kinesin (PF00225)

http://pfam.janelia.org/family/kinesin#tabview=tab4

HOME | SEARCH | BROWSE | FTP | HELP | ABOUT

Family: *Kinesin* (PF00225)

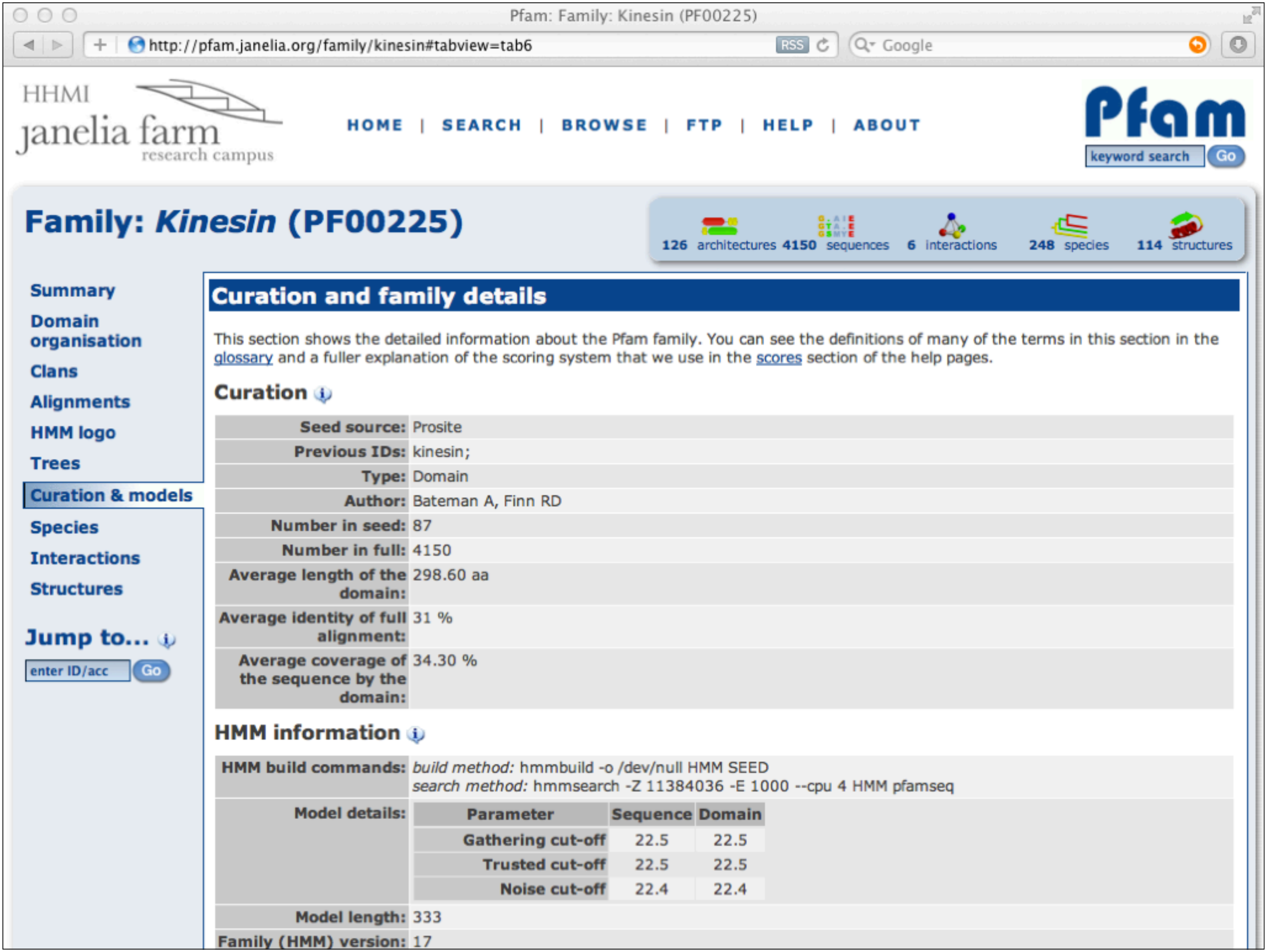
126 architectures 4150 sequences 6 interactions 248 species 114 structures

- Summary
- Domain organisation
- Clans
- Alignments
- HMM logo
- Trees
- Curation & models
- Species
- Interactions
- Structures

Jump to...

HMM logo

HMM logos is one way of visualising profile HMMs. Logos provide a quick overview of the properties of an HMM in a graphical form. You can see a more detailed description of HMM logos and find out how you can interpret them here. More...



Family: *Kinesin* (PF00225)

Curation and family details

This section shows the detailed information about the Pfam family. You can see the definitions of many of the terms in this section in the glossary and a fuller explanation of the scoring system that we use in the scores section of the help pages.

Curation

Seed source:	Prosite
Previous IDs:	kinesin;
Type:	Domain
Author:	Bateman A, Finn RD
Number in seed:	87
Number in full:	4150
Average length of the domain:	298.60 aa
Average identity of full alignment:	31 %
Average coverage of the sequence by the domain:	34.30 %

HMM information

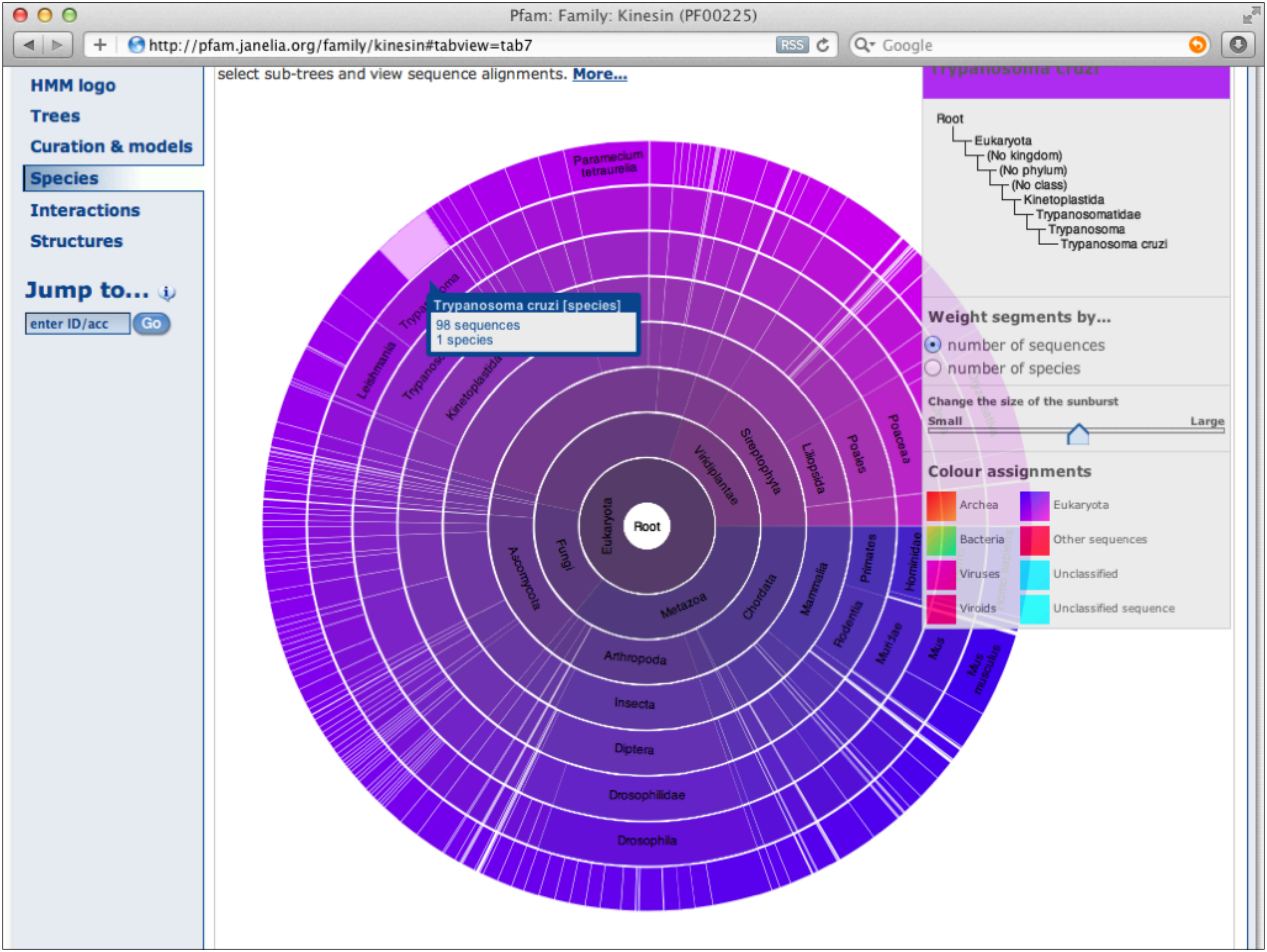
HMM build commands: *build method:* hmmbuild -o /dev/null HMM SEED
search method: hmmsearch -Z 11384036 -E 1000 --cpu 4 HMM pfamseq

Model details:

Parameter	Sequence	Domain
Gathering cut-off	22.5	22.5
Trusted cut-off	22.5	22.5
Noise cut-off	22.4	22.4

Model length: 333

Family (HMM) version: 17



Pfam: Family: Kinesin (PF00225)

http://pfam.janelia.org/family/kinesin#tabview=tab7

select sub-trees and view sequence alignments. More...

Trypanosoma cruzi [species]

98 sequences
1 species

Trypanosoma cruzi

```
Root
└Eukaryota
  └(No kingdom)
    └(No phylum)
      └(No class)
        └Kinetoplastida
          └Trypanosomatidae
            └Trypanosoma
              └Trypanosoma cruzi
```

Weight segments by...

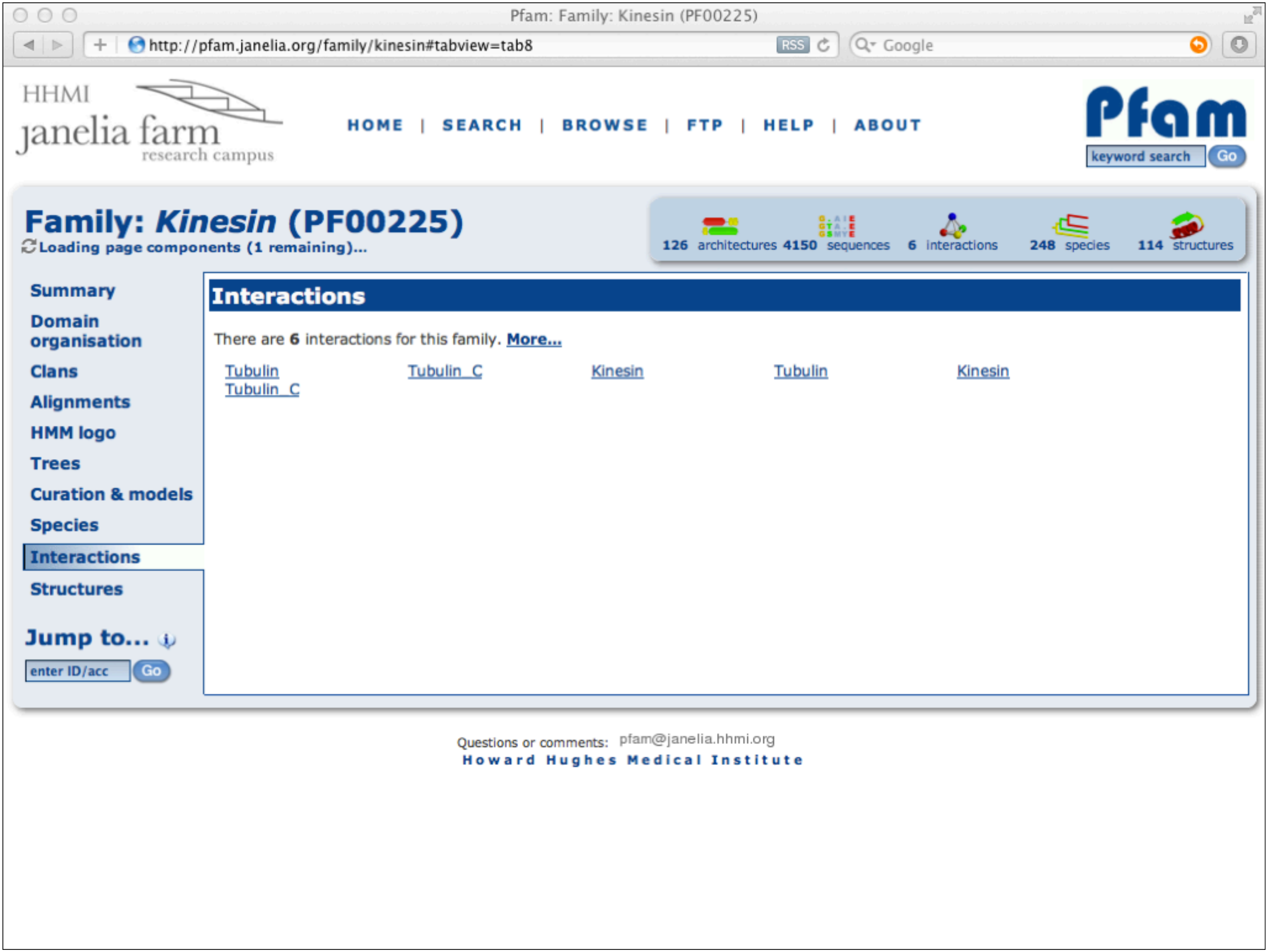
● number of sequences
○ number of species

Change the size of the sunburst

Small Large

Colour assignments

🟥	Archea	🟪	Eukaryota
🟩	Bacteria	🟥	Other sequences
🟪	Viruses	🟦	Unclassified
🟥	Viroids	🟦	Unclassified sequence


Pfam: Family: Kinesin (PF00225)
http://pfam.janelia.org/family/kinesin#tabview=tab8
RSS
Google
HHMI
janelia farm
research campus
HOME | SEARCH | BROWSE | FTP | HELP | ABOUT
Pfam
keyword search
Go

Family: *Kinesin* (PF00225)

Loading page components (1 remaining)...

126 architectures 4150 sequences 6 interactions 248 species 114 structures

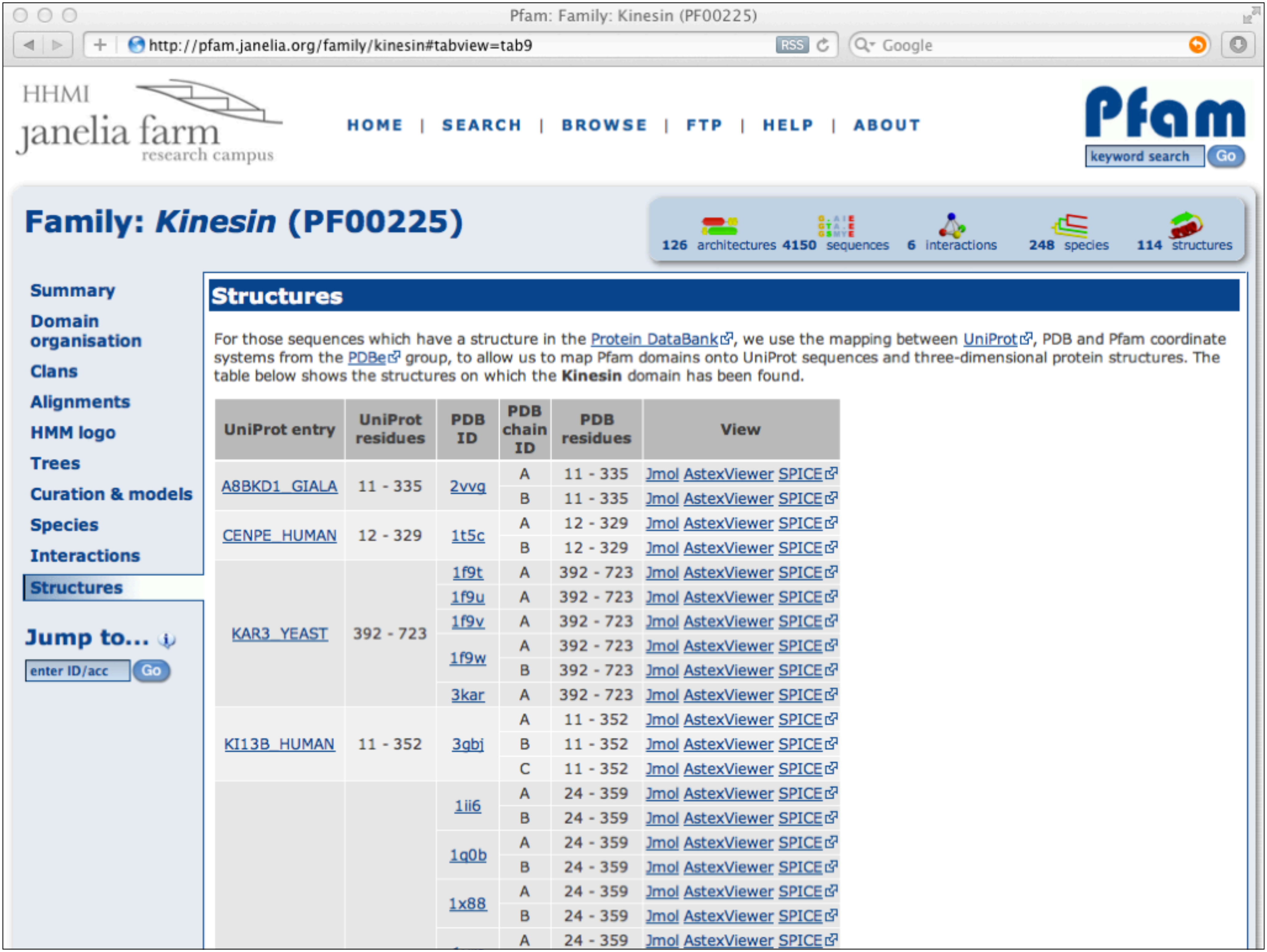
Interactions

There are **6** interactions for this family. **More...**

Tubulin Tubulin_C Kinesin Tubulin Kinesin
Tubulin_C

Questions or comments: pfam@janelia.hhmi.org

Howard Hughes Medical Institute



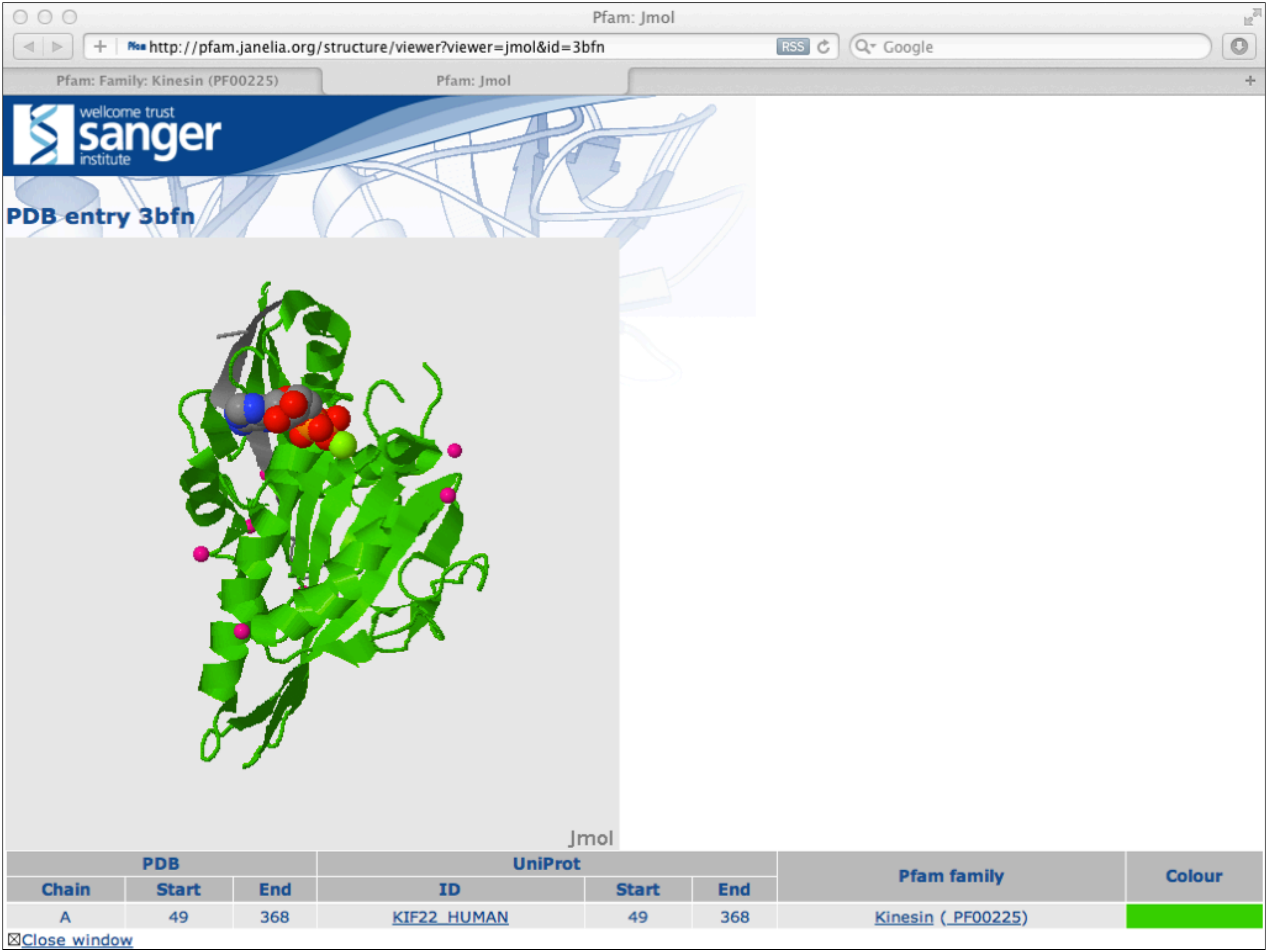
Pfam: Family: Kinesin (PF00225)
http://pfam.janelia.org/family/kinesin#tabview=tab9
RSS
Google
HHMI janelia farm research campus
HOME | SEARCH | BROWSE | FTP | HELP | ABOUT
Pfam
keyword search
Go

Family: *Kinesin* (PF00225)

Structures

For those sequences which have a structure in the Protein DataBank, we use the mapping between UniProt, PDB and Pfam coordinate systems from the PDBe group, to allow us to map Pfam domains onto UniProt sequences and three-dimensional protein structures. The table below shows the structures on which the **Kinesin** domain has been found.

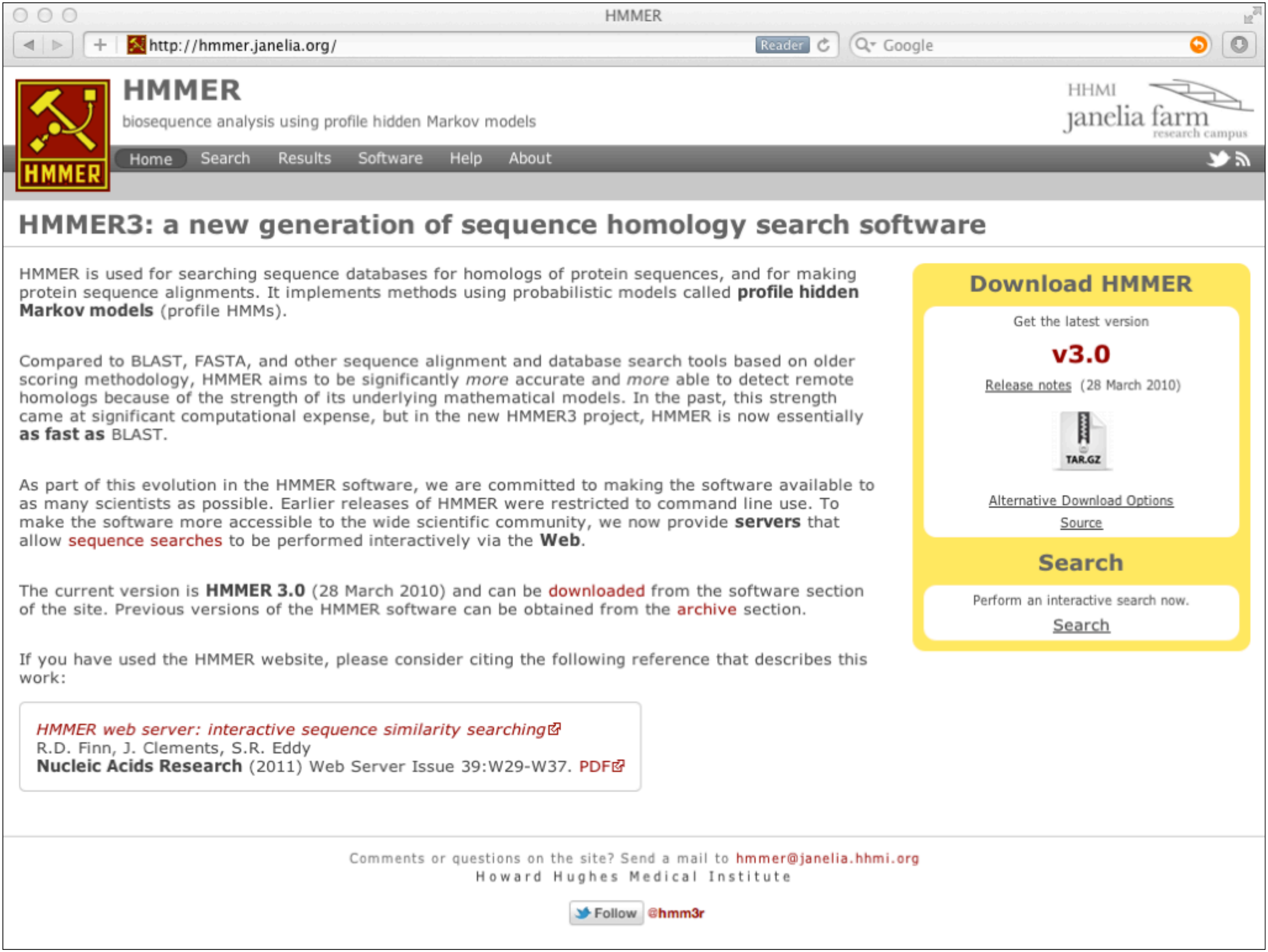
UniProt entry	UniProt residues	PDB ID	PDB chain ID	PDB residues	View
A8BKD1_GIALA	11 - 335	2vvg	A	11 - 335	Jmol AstexViewer SPICE
			B	11 - 335	Jmol AstexViewer SPICE
CENPE_HUMAN	12 - 329	1t5c	A	12 - 329	Jmol AstexViewer SPICE
			B	12 - 329	Jmol AstexViewer SPICE
KAR3_YEAST	392 - 723	1f9t	A	392 - 723	Jmol AstexViewer SPICE
		1f9u	A	392 - 723	Jmol AstexViewer SPICE
		1f9v	A	392 - 723	Jmol AstexViewer SPICE
		1f9w	A	392 - 723	Jmol AstexViewer SPICE
			B	392 - 723	Jmol AstexViewer SPICE
		3kar	A	392 - 723	Jmol AstexViewer SPICE
KI13B_HUMAN	11 - 352	3gbj	A	11 - 352	Jmol AstexViewer SPICE
			B	11 - 352	Jmol AstexViewer SPICE
			C	11 - 352	Jmol AstexViewer SPICE
		1ii6	A	24 - 359	Jmol AstexViewer SPICE
			B	24 - 359	Jmol AstexViewer SPICE
		1q0b	A	24 - 359	Jmol AstexViewer SPICE
			B	24 - 359	Jmol AstexViewer SPICE
		1x88	A	24 - 359	Jmol AstexViewer SPICE
			B	24 - 359	Jmol AstexViewer SPICE
			A	24 - 359	Jmol AstexViewer SPICE


Pfam: Jmol
http://pfam.janelia.org/structure/viewer?viewer=jmol&id=3bfn
RSS
Google
Pfam: Family: Kinesin (PF00225)
Pfam: Jmol
wellcome trust
sanger
institute
Jmol

PDB entry 3bfn

PDB			UniProt			Pfam family	Colour
Chain	Start	End	ID	Start	End		
A	49	368	KIF22_HUMAN	49	368	Kinesin (PF00225)	🟩

☒Close window



HMMER

biosequence analysis using profile hidden Markov models

HMMER3: a new generation of sequence homology search software

HMMER is used for searching sequence databases for homologs of protein sequences, and for making protein sequence alignments. It implements methods using probabilistic models called **profile hidden Markov models** (profile HMMs).

Compared to BLAST, FASTA, and other sequence alignment and database search tools based on older scoring methodology, HMMER aims to be significantly *more* accurate and *more* able to detect remote homologs because of the strength of its underlying mathematical models. In the past, this strength came at significant computational expense, but in the new HMMER3 project, HMMER is now essentially **as fast as** BLAST.

As part of this evolution in the HMMER software, we are committed to making the software available to as many scientists as possible. Earlier releases of HMMER were restricted to command line use. To make the software more accessible to the wide scientific community, we now provide **servers** that allow sequence searches to be performed interactively via the **Web**.

The current version is **HMMER 3.0** (28 March 2010) and can be downloaded from the software section of the site. Previous versions of the HMMER software can be obtained from the archive section.

If you have used the HMMER website, please consider citing the following reference that describes this work:

HMMER web server: interactive sequence similarity searching
R.D. Finn, J. Clements, S.R. Eddy
Nucleic Acids Research (2011) Web Server Issue 39:W29-W37. PDF

Download HMMER

Get the latest version

v3.0

Release notes (28 March 2010)

Alternative Download Options

Source

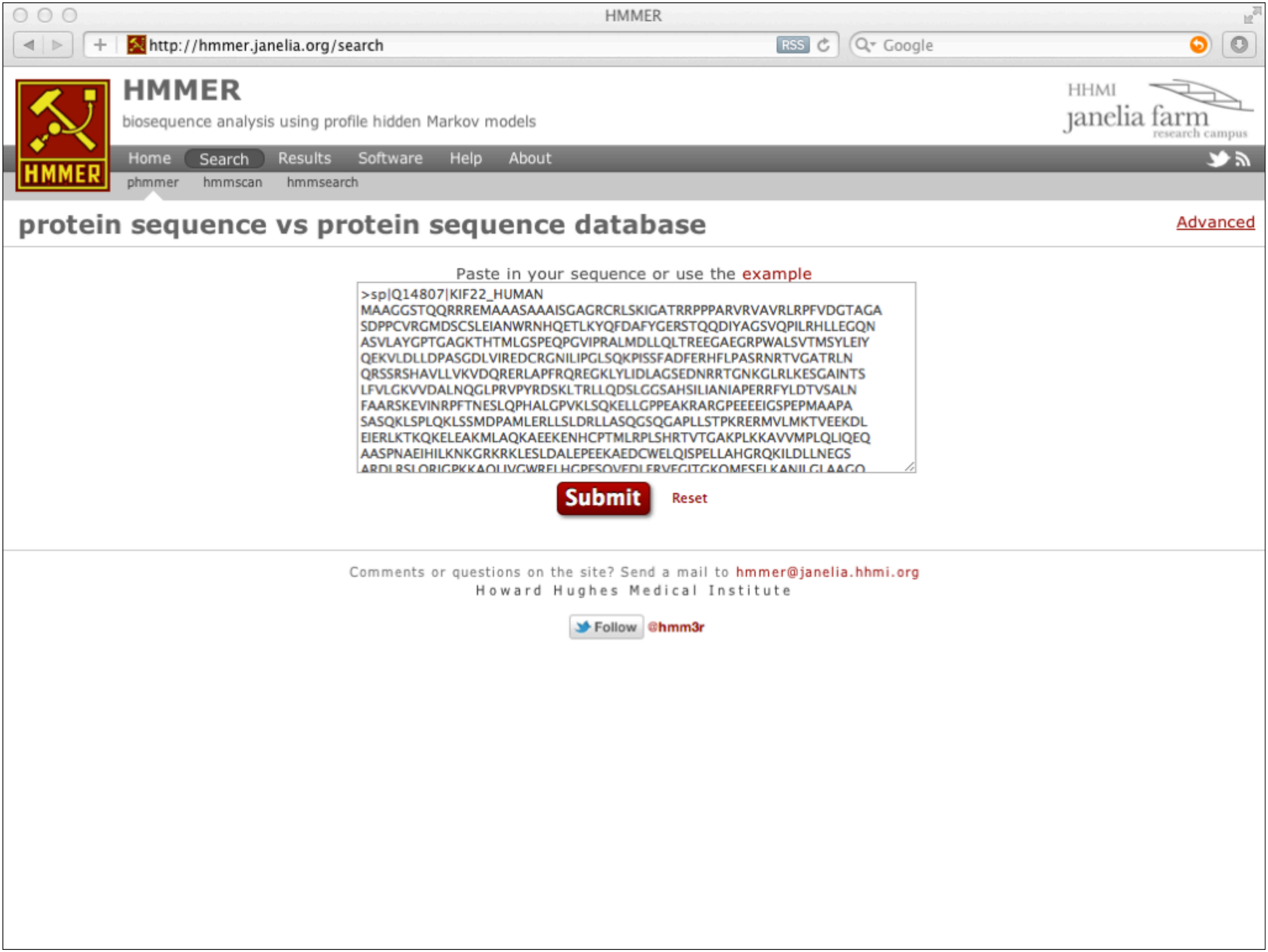
Search

Perform an interactive search now.

Search

Comments or questions on the site? Send a mail to hmmer@janelia.hhmi.org

Howard Hughes Medical Institute



HMMER

biosequence analysis using profile hidden Markov models

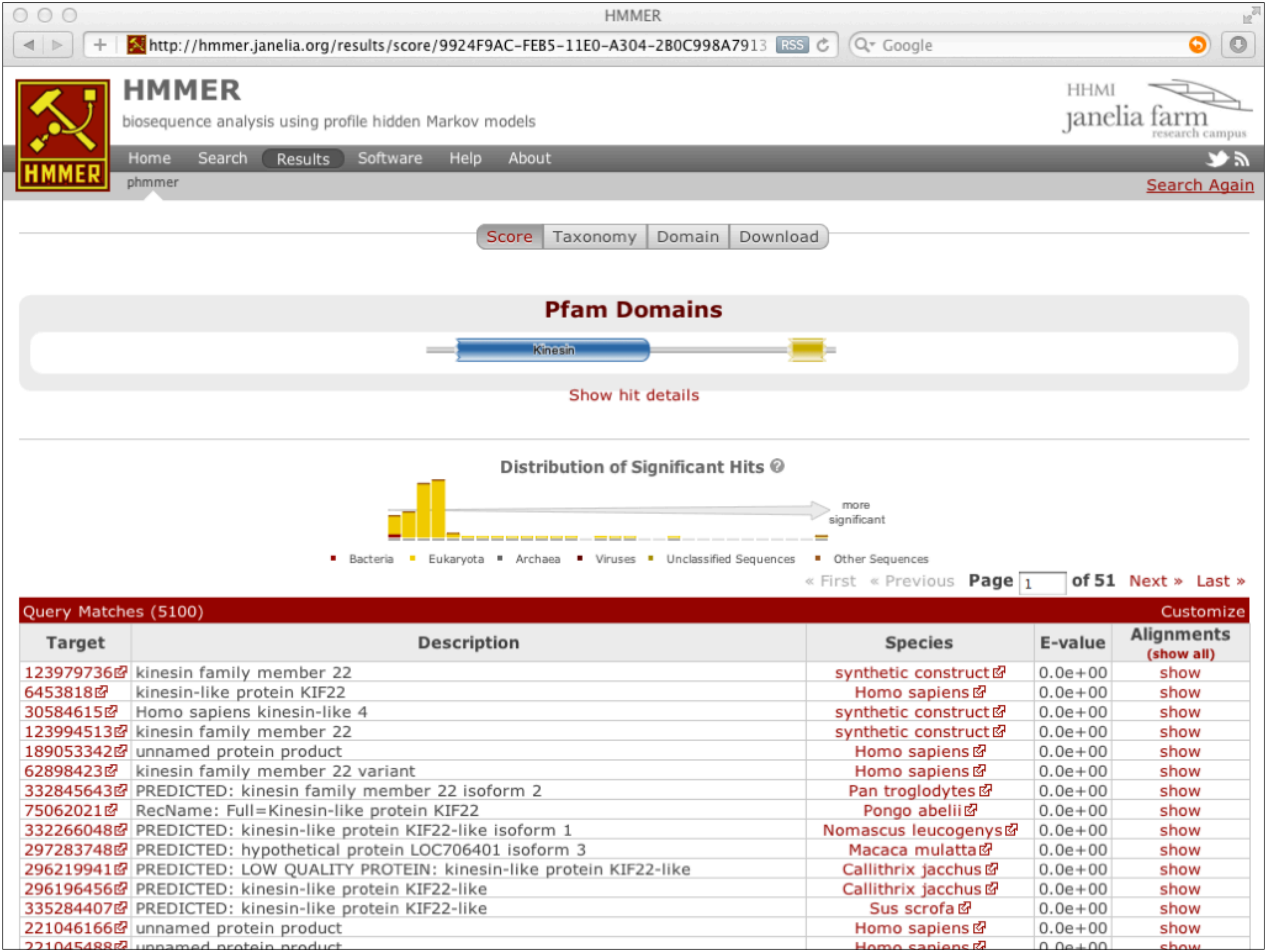
protein sequence vs protein sequence database

Paste in your sequence or use the example

```
>sp|Q14807|KIF22_HUMAN
MAAGGSTQQRRREMAAASAAAISGAGRCRLSKIGATRRPPPARVRVAVRLRPFVDGTAGA
SDPPCVRGMDSCSLEIANWRNHQETLKYQFDAFYGERSTQQDIYAGSVQPILRHLLEGQN
ASVLAYGPTGAGKTHTMLGSPEQPGVIPRALMDLLQLTREEGAEGRPWALSVTMSYLEIY
QEKVLDLLDPASGDLVIREDCRGNILIPGLSQKPISSFADFERHFLPASRNRTVGATRLN
QRSSRSHAVLLVKVDQRERLAPFRQREGKLYLIDLAGSEDNRRTGNKGLRLKESGAINTS
LFVLGKVVDALNQGLPRVPYRDSKLTRLLQDSLGGSAHSILIANIAPERRFYLDTVSALN
FAARSKEVINRPFTNESLQPHALGPVKLSQKELLGPPEAKRARGPEEEEIGSPEPMAAPA
SASQKLSPLQKLSSMDPAMLERLLSLDRLLASQGSQGAPLLSTPKRERMVLMKTVEEKDL
EIERLKTKQKELEAKMLAQKAEEKENHCPTMLRPLSHRTVTGAKPLKKAVVMPLQLIQEQ
AASPNAEIHILKNKGRKRKLESLDALEPEEKAEDCWELQISPELLAHGRQKILDLLNEGS
ARDLRSLQRIGPKKAQLIVGWRELHGPFSQVEDLERVEGITGKQMESFLKANILGLAAGQ
```

Comments or questions on the site? Send a mail to hmmer@janelia.hhmi.org

Howard Hughes Medical Institute



HMMER

biosequence analysis using profile hidden Markov models

Pfam Domains

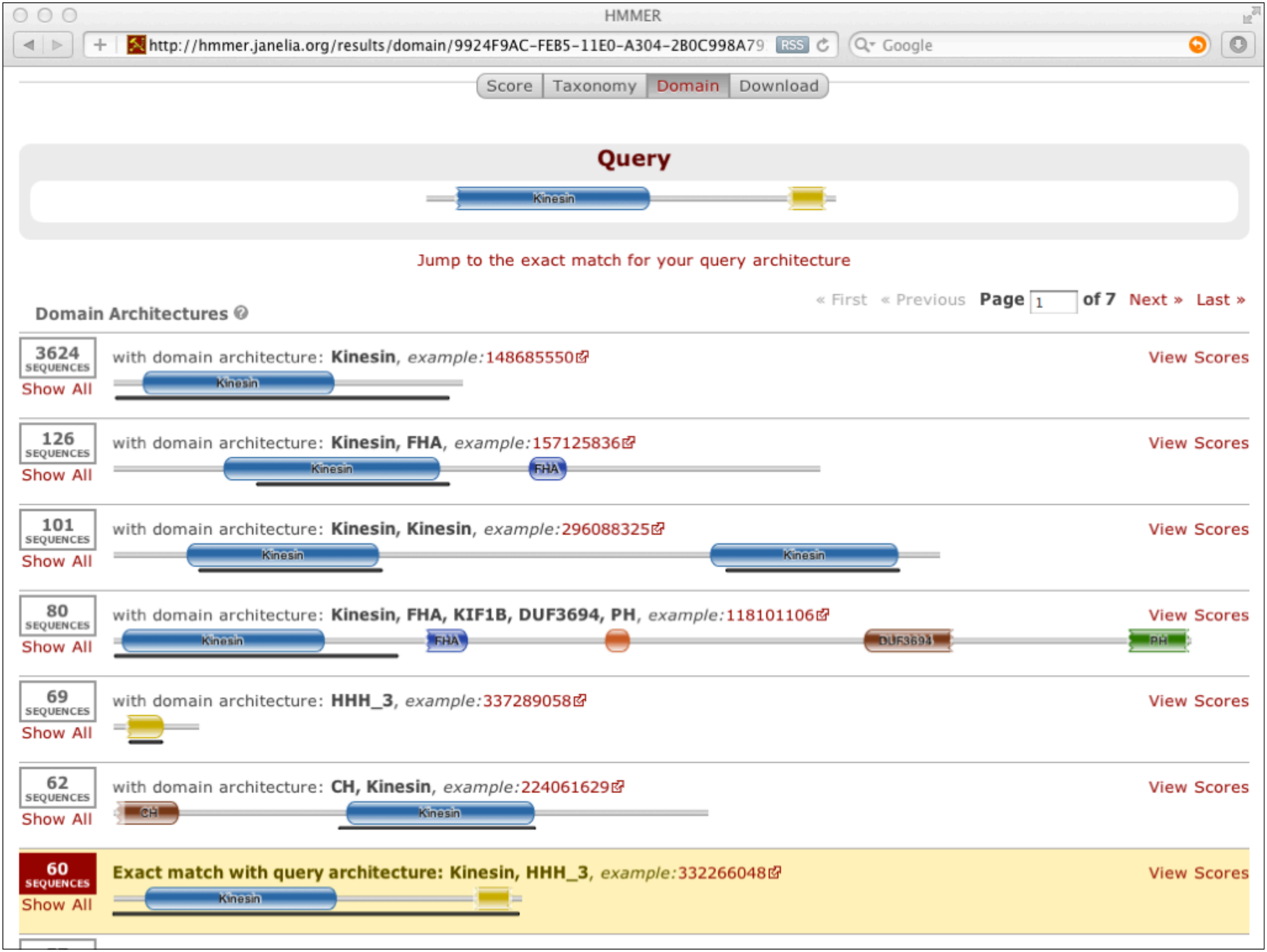
Show hit details

Distribution of Significant Hits

Page 1 of 51

Query Matches (5100)

Target	Description	Species	E-value	Alignments (show all)
123979736	kinesin family member 22	synthetic construct	0.0e+00	show
6453818	kinesin-like protein KIF22	Homo sapiens	0.0e+00	show
30584615	Homo sapiens kinesin-like 4	synthetic construct	0.0e+00	show
123994513	kinesin family member 22	synthetic construct	0.0e+00	show
189053342	unnamed protein product	Homo sapiens	0.0e+00	show
62898423	kinesin family member 22 variant	Homo sapiens	0.0e+00	show
332845643	PREDICTED: kinesin family member 22 isoform 2	Pan troglodytes	0.0e+00	show
75062021	RecName: Full=Kinesin-like protein KIF22	Pongo abelii	0.0e+00	show
332266048	PREDICTED: kinesin-like protein KIF22-like isoform 1	Nomascus leucogenys	0.0e+00	show
297283748	PREDICTED: hypothetical protein LOC706401 isoform 3	Macaca mulatta	0.0e+00	show
296219941	PREDICTED: LOW QUALITY PROTEIN: kinesin-like protein KIF22-like	Callithrix jacchus	0.0e+00	show
296196456	PREDICTED: kinesin-like protein KIF22-like	Callithrix jacchus	0.0e+00	show
335284407	PREDICTED: kinesin-like protein KIF22-like	Sus scrofa	0.0e+00	show
221046166	unnamed protein product	Homo sapiens	0.0e+00	show
221045488	unnamed protein product	Homo sapiens	0.0e+00	show



Query

Jump to the exact match for your query architecture

Domain Architectures

« First « Previous Page 1 of 7 Next » Last »

3624 SEQUENCES Show All — with domain architecture: **Kinesin**, *example:*148685550 — View Scores

126 SEQUENCES Show All — with domain architecture: **Kinesin, FHA**, *example:*157125836 — View Scores

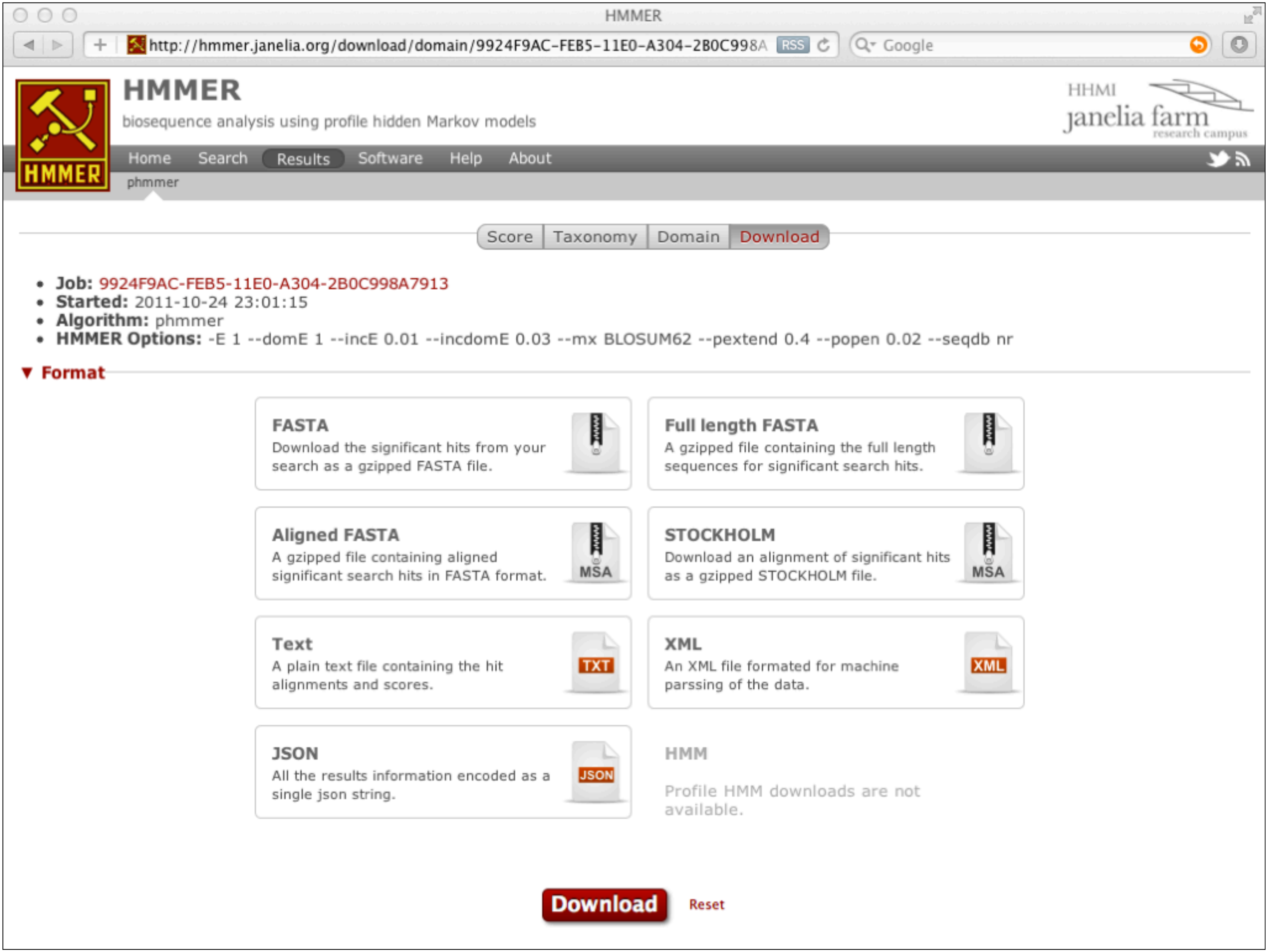
101 SEQUENCES Show All — with domain architecture: **Kinesin, Kinesin**, *example:*296088325 — View Scores

80 SEQUENCES Show All — with domain architecture: **Kinesin, FHA, KIF1B, DUF3694, PH**, *example:*118101106 — View Scores

69 SEQUENCES Show All — with domain architecture: **HHH_3**, *example:*337289058 — View Scores

62 SEQUENCES Show All — with domain architecture: **CH, Kinesin**, *example:*224061629 — View Scores

60 SEQUENCES Show All — **Exact match with query architecture: Kinesin, HHH_3**, *example:*332266048 — View Scores



HMMER

biosequence analysis using profile hidden Markov models

- **Job:** 9924F9AC-FEB5-11E0-A304-2B0C998A7913
- **Started:** 2011-10-24 23:01:15
- **Algorithm:** phmmer
- **HMMER Options:** -E 1 --domE 1 --incE 0.01 --incdomE 0.03 --mx BLOSUM62 --pextend 0.4 --popen 0.02 --seqdb nr

▼ Format

FASTA
Download the significant hits from your search as a gzipped FASTA file.

Full length FASTA
A gzipped file containing the full length sequences for significant search hits.

Aligned FASTA
A gzipped file containing aligned significant search hits in FASTA format.

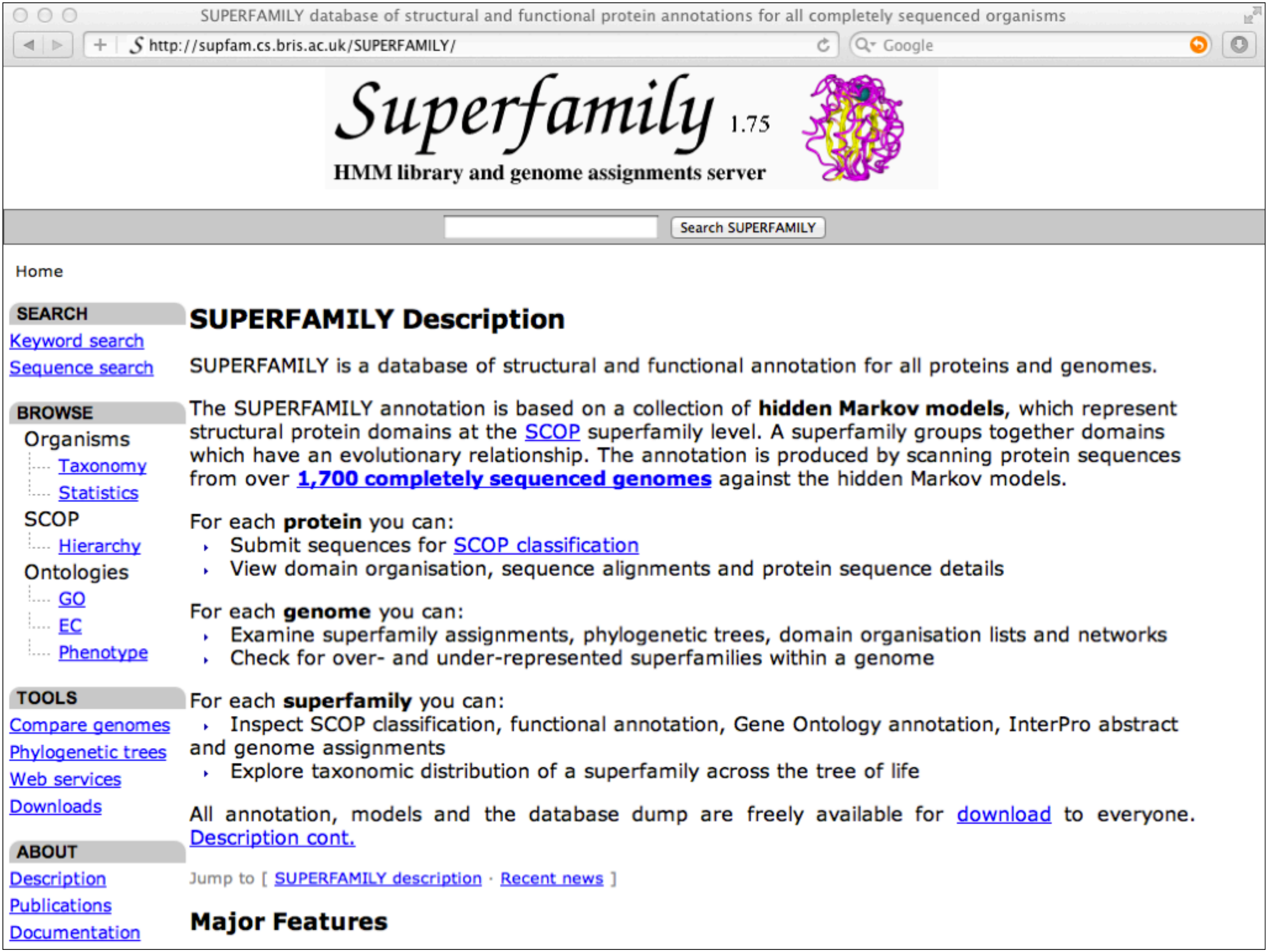
STOCKHOLM
Download an alignment of significant hits as a gzipped STOCKHOLM file.

Text
A plain text file containing the hit alignments and scores.

XML
An XML file formated for machine parssing of the data.

JSON
All the results information encoded as a single json string.

HMM
Profile HMM downloads are not available.



SUPERFAMILY Description

SUPERFAMILY is a database of structural and functional annotation for all proteins and genomes.

The SUPERFAMILY annotation is based on a collection of **hidden Markov models**, which represent structural protein domains at the SCOP superfamily level. A superfamily groups together domains which have an evolutionary relationship. The annotation is produced by scanning protein sequences from over **1,700 completely sequenced genomes** against the hidden Markov models.

For each **protein** you can:

- Submit sequences for SCOP classification
- View domain organisation, sequence alignments and protein sequence details

For each **genome** you can:

- Examine superfamily assignments, phylogenetic trees, domain organisation lists and networks
- Check for over- and under-represented superfamilies within a genome

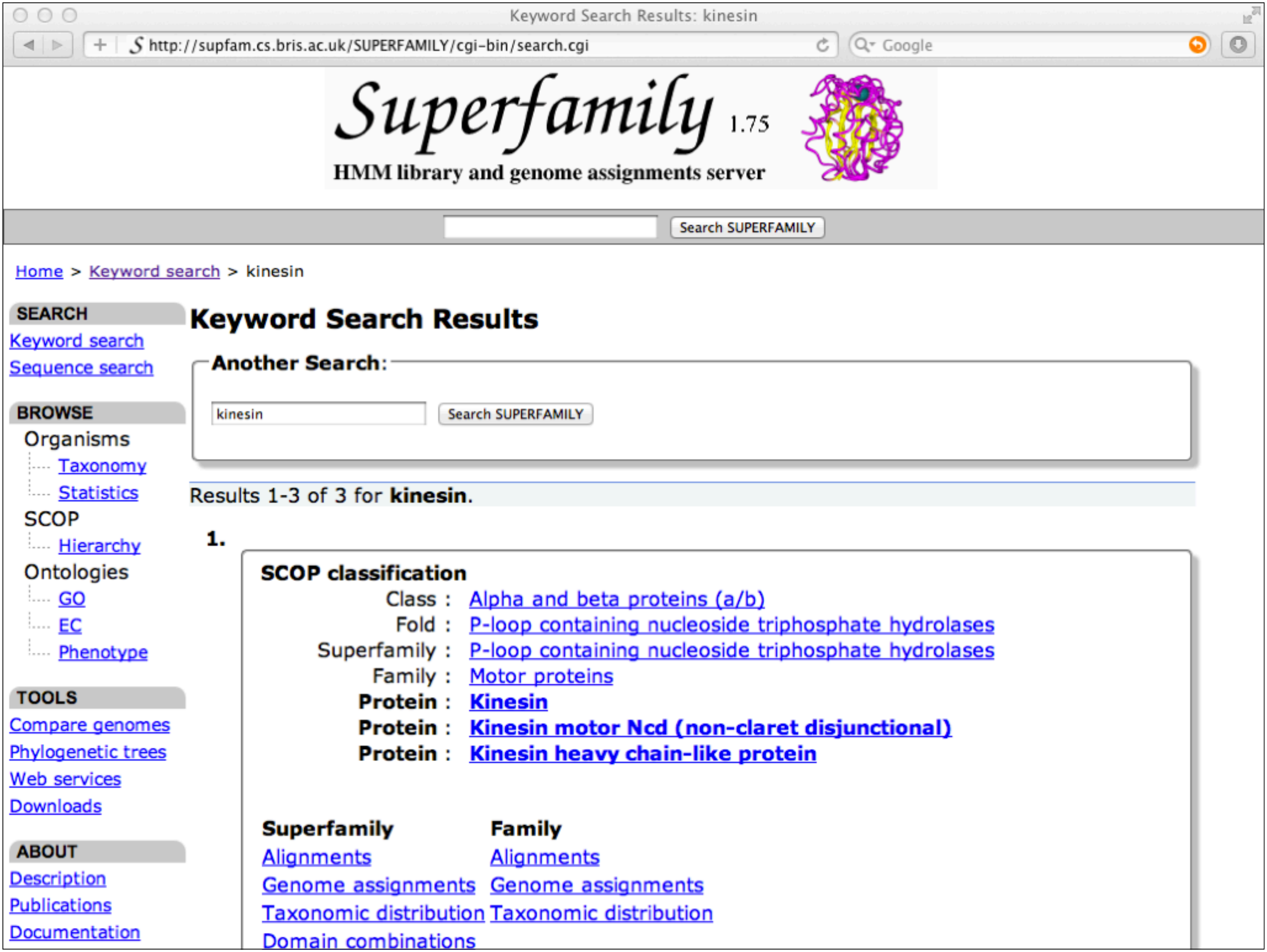
For each **superfamily** you can:

- Inspect SCOP classification, functional annotation, Gene Ontology annotation, InterPro abstract and genome assignments
- Explore taxonomic distribution of a superfamily across the tree of life

All annotation, models and the database dump are freely available for download to everyone. Description cont.

Jump to [SUPERFAMILY description · Recent news]

Major Features


Keyword Search Results: kinesin
http://supfam.cs.bris.ac.uk/SUPERFAMILY/cgi-bin/search.cgi
Google
Superfamily 1.75
HMM library and genome assignments server
Search SUPERFAMILY

Home > Keyword search > kinesin

Keyword Search Results

Another Search:

kinesin Search SUPERFAMILY

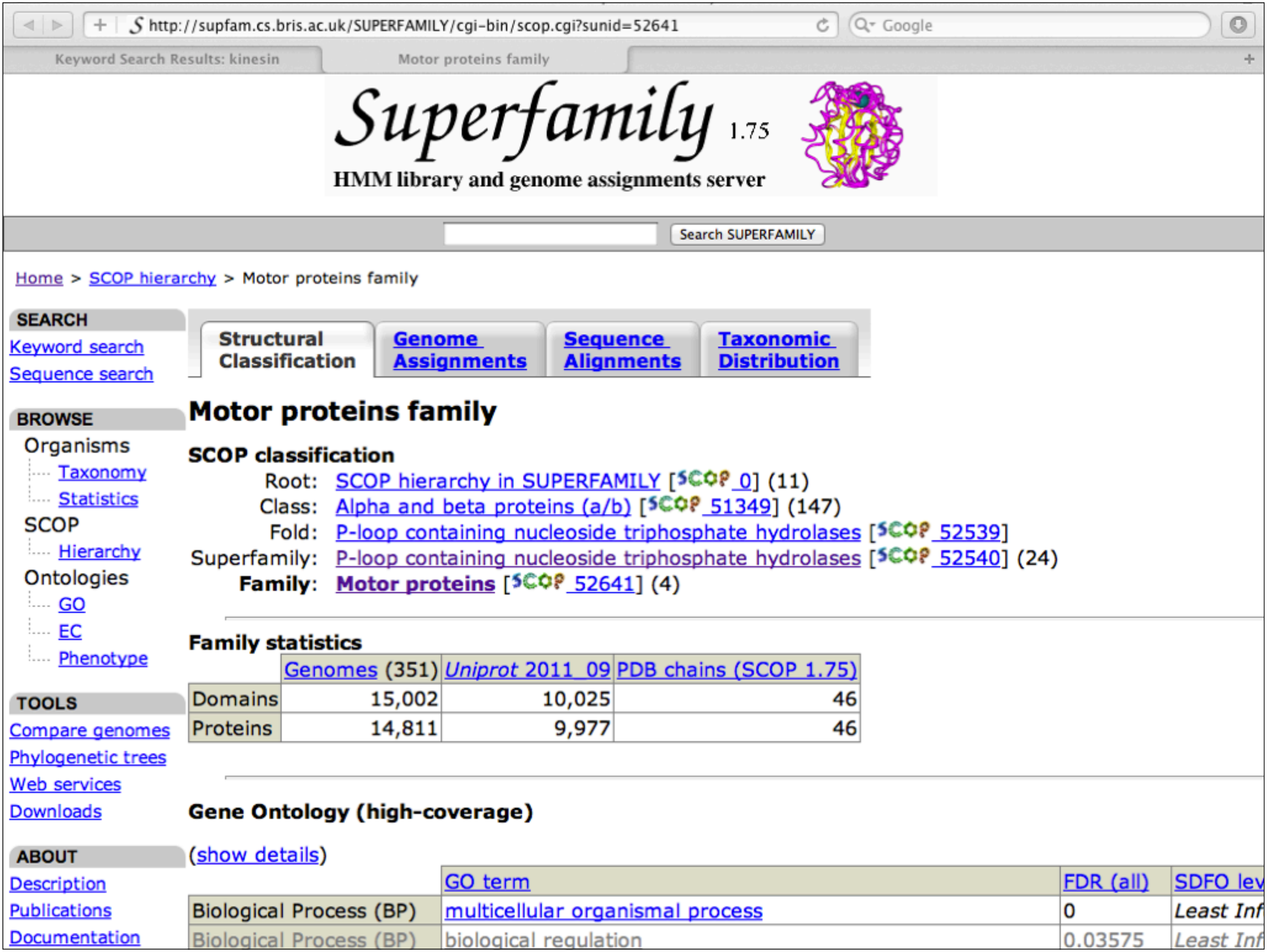
Results 1-3 of 3 for **kinesin**.

1.

SCOP classification

Class :	Alpha and beta proteins (a/b)
Fold :	P-loop containing nucleoside triphosphate hydrolases
Superfamily :	P-loop containing nucleoside triphosphate hydrolases
Family :	Motor proteins
Protein :	**Kinesin**
Protein :	**Kinesin motor Ncd (non-claret disjunctional)**
Protein :	**Kinesin heavy chain-like protein**

Superfamily	**Family**
Alignments	Alignments
Genome assignments	Genome assignments
Taxonomic distribution	Taxonomic distribution
Domain combinations	


http://supfam.cs.bris.ac.uk/SUPERFAMILY/cgi-bin/scop.cgi?sunid=52641
Google
Keyword Search Results: kinesin
Motor proteins family
Superfamily 1.75
HMM library and genome assignments server
Search SUPERFAMILY

Home > SCOP hierarchy > Motor proteins family

Motor proteins family

SCOP classification

Root: SCOP hierarchy in SUPERFAMILY [SCOP 0] (11)
Class: Alpha and beta proteins (a/b) [SCOP 51349] (147)
Fold: P-loop containing nucleoside triphosphate hydrolases [SCOP 52539]
Superfamily: P-loop containing nucleoside triphosphate hydrolases [SCOP 52540] (24)
Family: Motor proteins [SCOP 52641] (4)

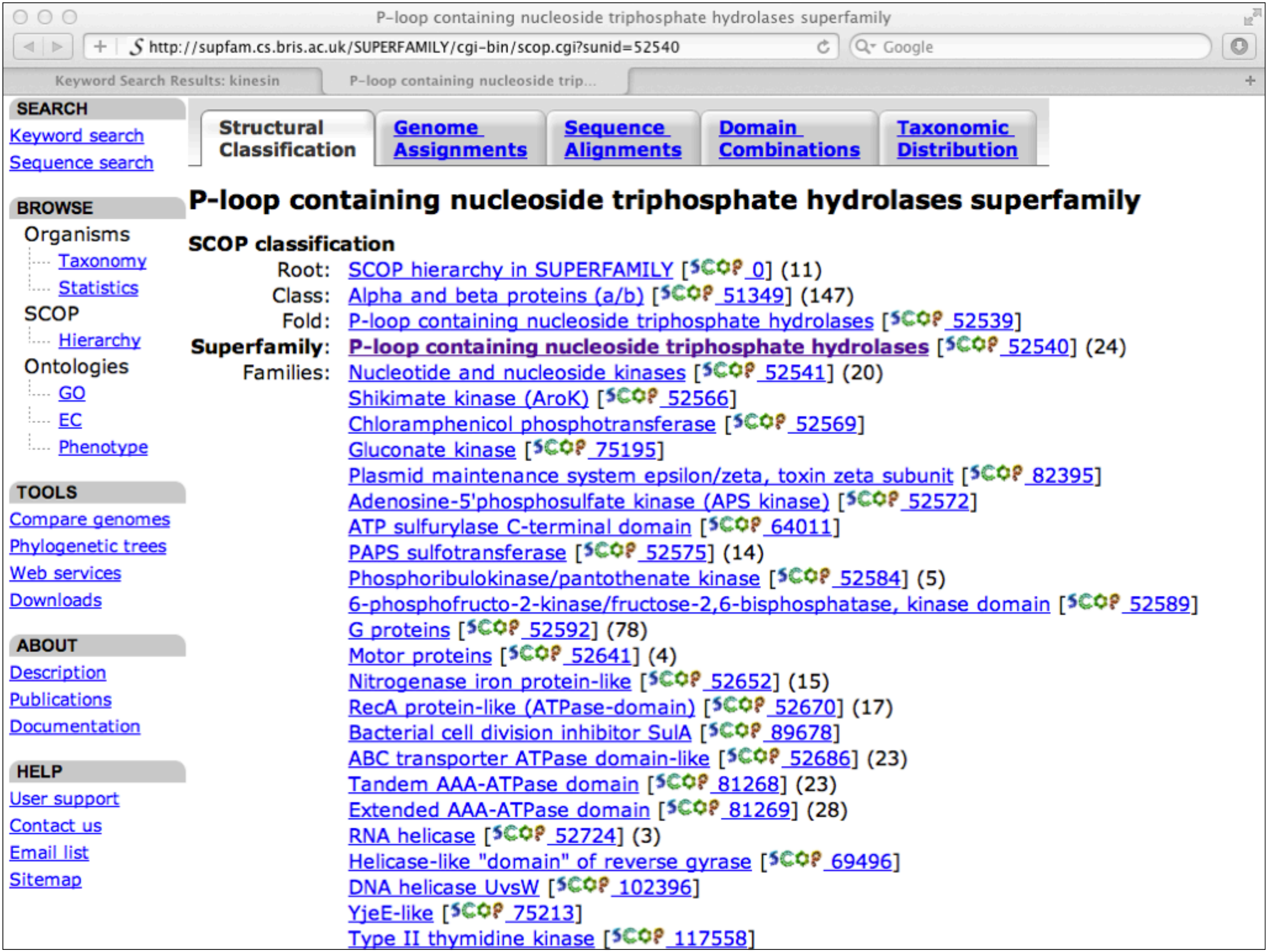
Family statistics

	Genomes (351)	*Uniprot 2011_09*	PDB chains (SCOP 1.75)
Domains	15,002	10,025	46
Proteins	14,811	9,977	46

Gene Ontology (high-coverage)

(show details)

	GO term	FDR (all)	SDFO lev
Biological Process (BP)	multicellular organismal process	0	*Least Inf*
Biological Process (BP)	biological regulation	0.03575	*Least Inf*



P-loop containing nucleoside triphosphate hydrolases superfamily

SCOP classification

Root: SCOP hierarchy in SUPERFAMILY [SCOP 0] (11)
Class: Alpha and beta proteins (a/b) [SCOP 51349] (147)
Fold: P-loop containing nucleoside triphosphate hydrolases [SCOP 52539]
Superfamily: P-loop containing nucleoside triphosphate hydrolases [SCOP 52540] (24)
Families: Nucleotide and nucleoside kinases [SCOP 52541] (20)
Shikimate kinase (AroK) [SCOP 52566]
Chloramphenicol phosphotransferase [SCOP 52569]
Gluconate kinase [SCOP 75195]
Plasmid maintenance system epsilon/zeta, toxin zeta subunit [SCOP 82395]
Adenosine-5'phosphosulfate kinase (APS kinase) [SCOP 52572]
ATP sulfurylase C-terminal domain [SCOP 64011]
PAPS sulfotransferase [SCOP 52575] (14)
Phosphoribulokinase/pantothenate kinase [SCOP 52584] (5)
6-phosphofructo-2-kinase/fructose-2,6-bisphosphatase, kinase domain [SCOP 52589]
G proteins [SCOP 52592] (78)
Motor proteins [SCOP 52641] (4)
Nitrogenase iron protein-like [SCOP 52652] (15)
RecA protein-like (ATPase-domain) [SCOP 52670] (17)
Bacterial cell division inhibitor SulA [SCOP 89678]
ABC transporter ATPase domain-like [SCOP 52686] (23)
Tandem AAA-ATPase domain [SCOP 81268] (23)
Extended AAA-ATPase domain [SCOP 81269] (28)
RNA helicase [SCOP 52724] (3)
Helicase-like "domain" of reverse gyrase [SCOP 69496]
DNA helicase UvsW [SCOP 102396]
YjeE-like [SCOP 75213]
Type II thymidine kinase [SCOP 117558]

That’s it!