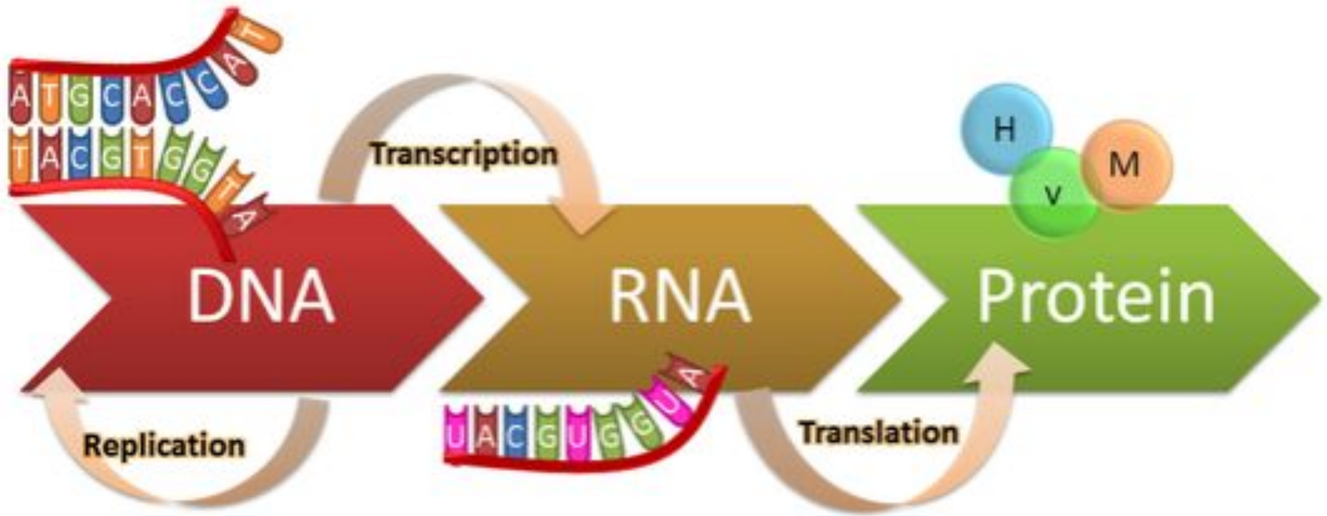
Cloud Search: A Heuristic Forward-Backward

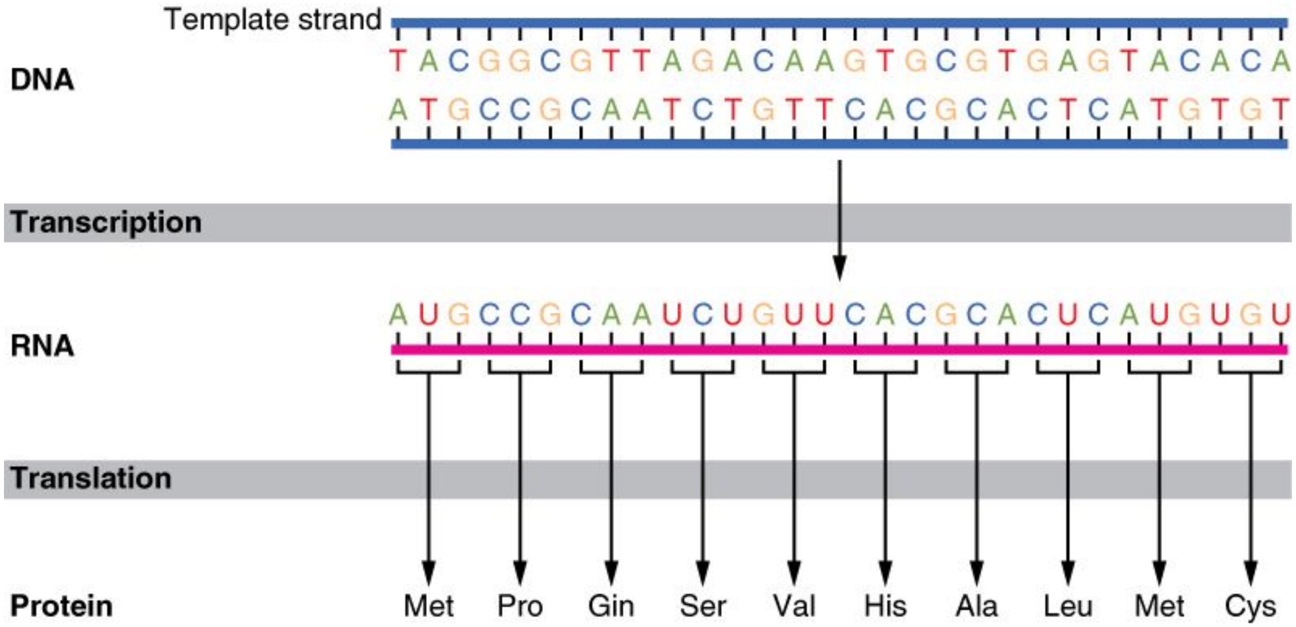
By David Rich

Biological Background

Central Dogma of Biology



Central Dogma of Biology


Template strand
DNA
TACGGCGTTAGACAAGTGCGTGAGTACACA
ATGCCGCAATCTGTTCACGCACTCATGTGT
Transcription
RNA
AUGCCGCAAUCUGUUCACGCACUCAUGUGU
Translation
Protein
Met
Pro
Gin
Ser
Val
His
Ala
Leu
Met
Cys

Biological Sequence Alignment

Sequence Alignment

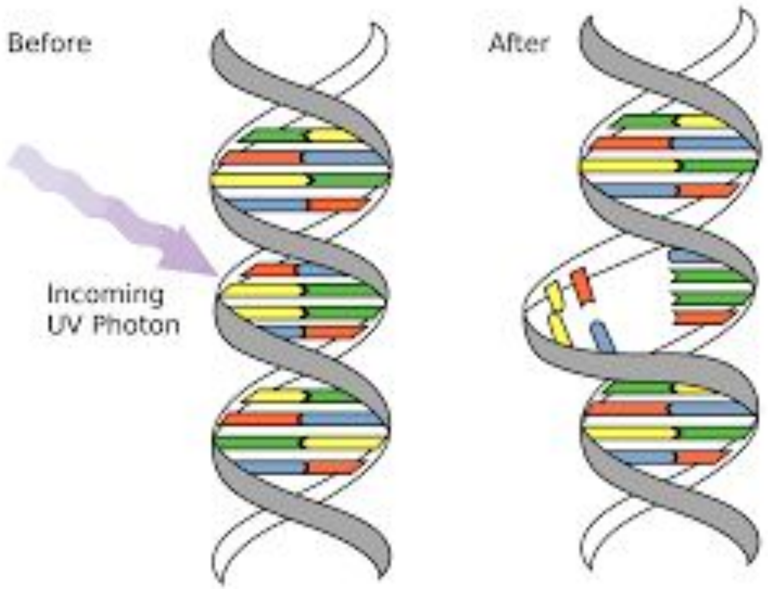
Could these two DNA sequences be related? How can we tell?

```
ATCTCGTATGAT

GTCTATCAC
```

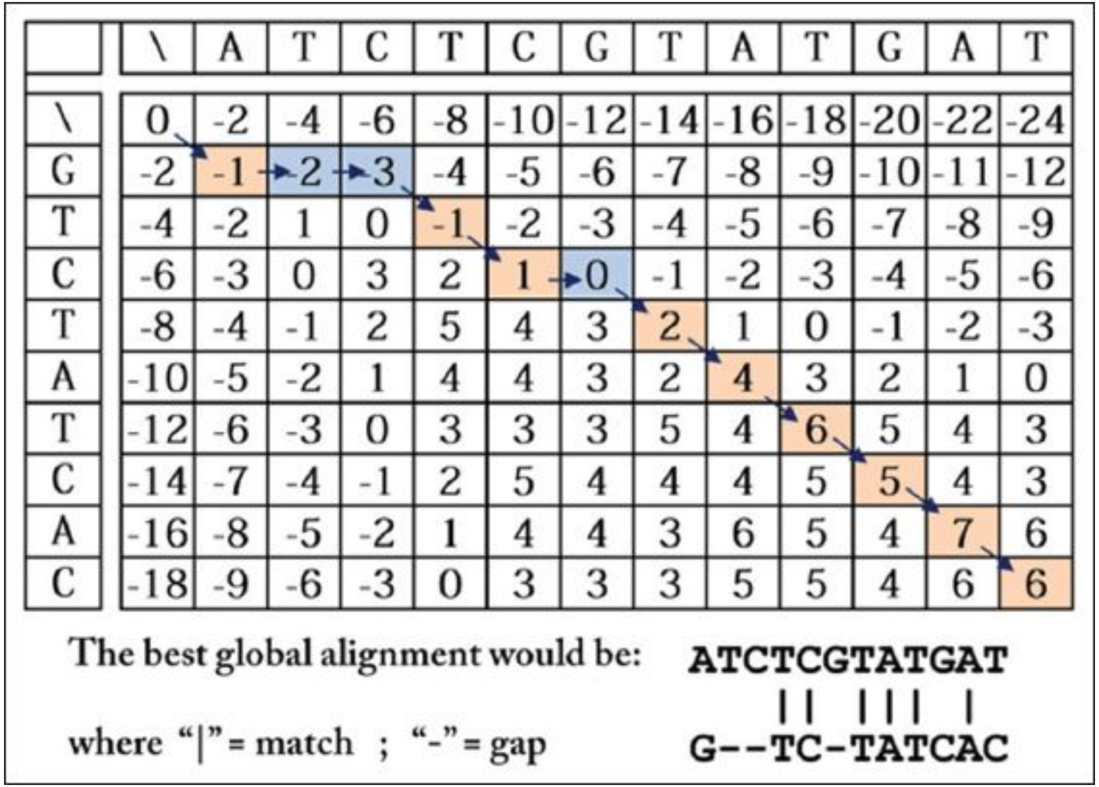
DNA Mutations

- Mutation are the changes to DNA. These occur naturally over time.
- Because mutations tend to happen at relatively stable intervals, we can use this to see how closely related two things are.
- This can also show us important parts of the genome, since changes to critical regions generally results in the inability to procreate. This causes these regions to be conserved among its offspring.



Smith-Waterman

- One of the first algorithms used for solving the problem of sequence alignment.
- Finds the best sequence alignment by finding maximal path through dynamic programming matrix.
- Uses constant terms for its match, insert, and delete scoring scheme.



	\	A	T	C	T	C	G	T	A	T	G	A	T
\	0	-2	-4	-6	-8	-10	-12	-14	-16	-18	-20	-22	-24
G	-2	-1	-2	-3	-4	-5	-6	-7	-8	-9	-10	-11	-12
T	-4	-2	1	0	-1	-2	-3	-4	-5	-6	-7	-8	-9
C	-6	-3	0	3	2	1	0	-1	-2	-3	-4	-5	-6
T	-8	-4	-1	2	5	4	3	2	1	0	-1	-2	-3
A	-10	-5	-2	1	4	4	3	2	4	3	2	1	0
T	-12	-6	-3	0	3	3	3	5	4	6	5	4	3
C	-14	-7	-4	-1	2	5	4	4	4	5	5	4	3
A	-16	-8	-5	-2	1	4	4	3	6	5	4	7	6
C	-18	-9	-6	-3	0	3	3	3	5	5	4	6	6

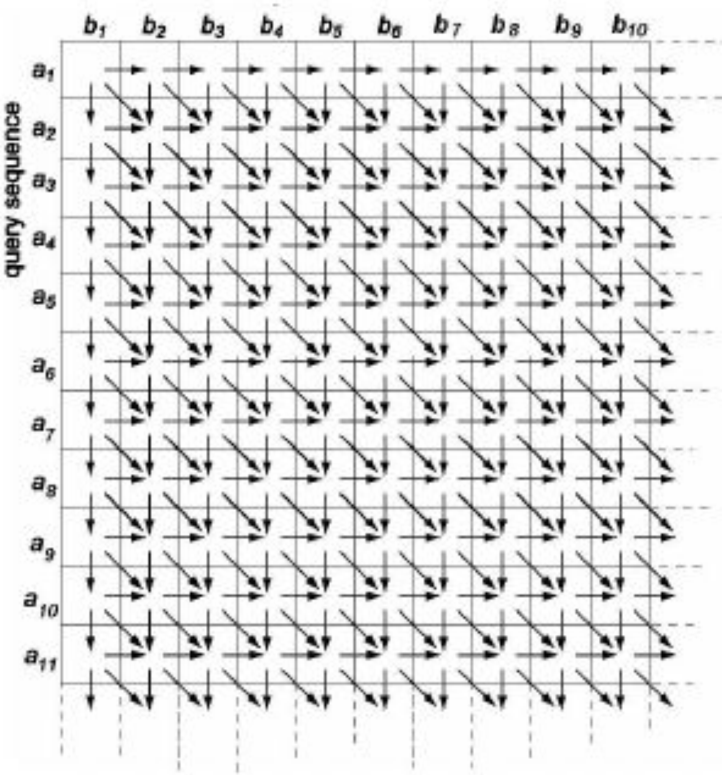
The best global alignment would be:

```
ATCTCGTATGAT
   || ||| |
G--TC-TATCAC
```

where "|" = match ; "-" = gap

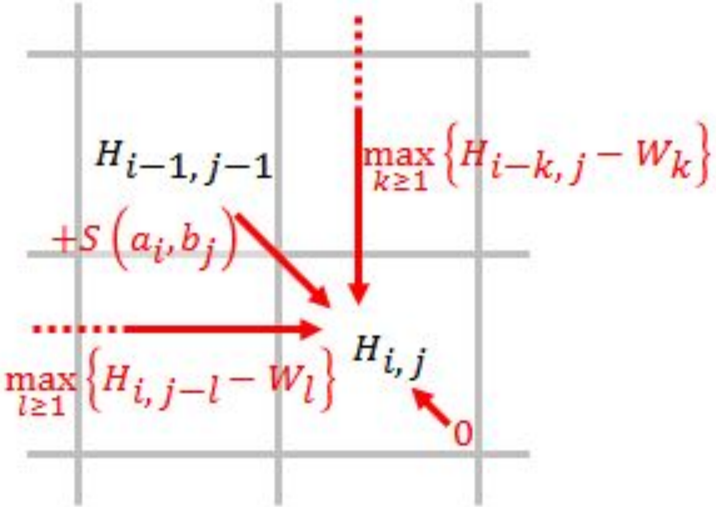
Smith-Waterman Runtime

- Every cell is conditionally dependent upon all cells to it left and right.
- By calculating these cells from top-left to bottom-right, we can isolate this relationship such that each cell is only dependent on its immediately adjacent cells.



Smith-Waterman Runtime

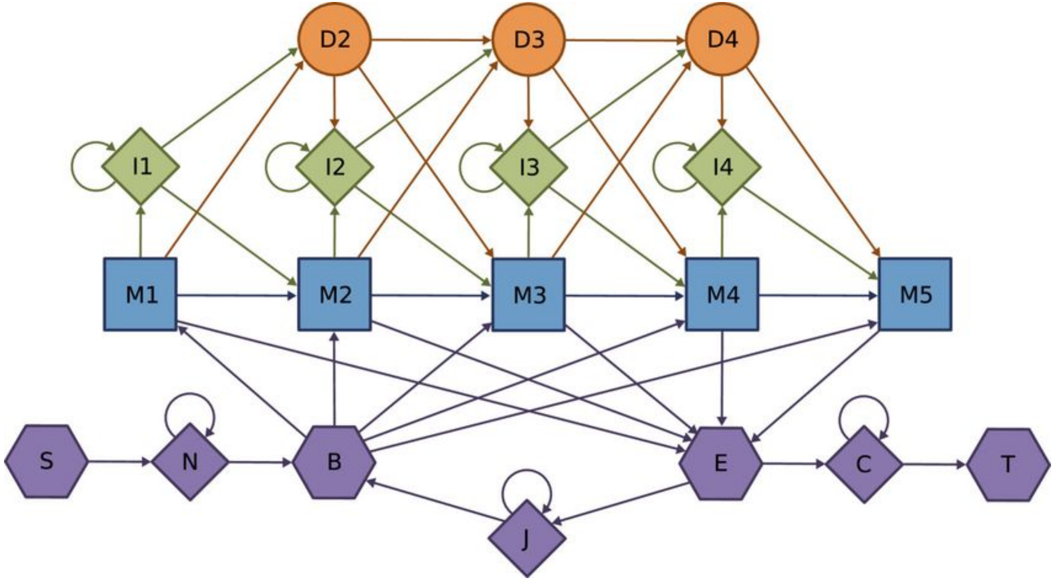
- If both sequences are of length ~N, then there will ~N^2 cells in the dynamic programming matrix (H).
- If we move through the matrix row-by-row, at every point H(i,j), we will have the values of its dependent cells.
- Therefore, even though there are far more than N^2 possible alignments, we only need to compute each cell once.


$H_{i-1,j-1}$
$\max_{k\geq1}\{H_{i-k,j}-W_k\}$
$+S(a_i,b_j)$
$\max_{l\geq1}\{H_{i,j-l}-W_l\}$
$H_{i,j}$
0

Hidden Markov Models, Viterbi & Forward-Backward

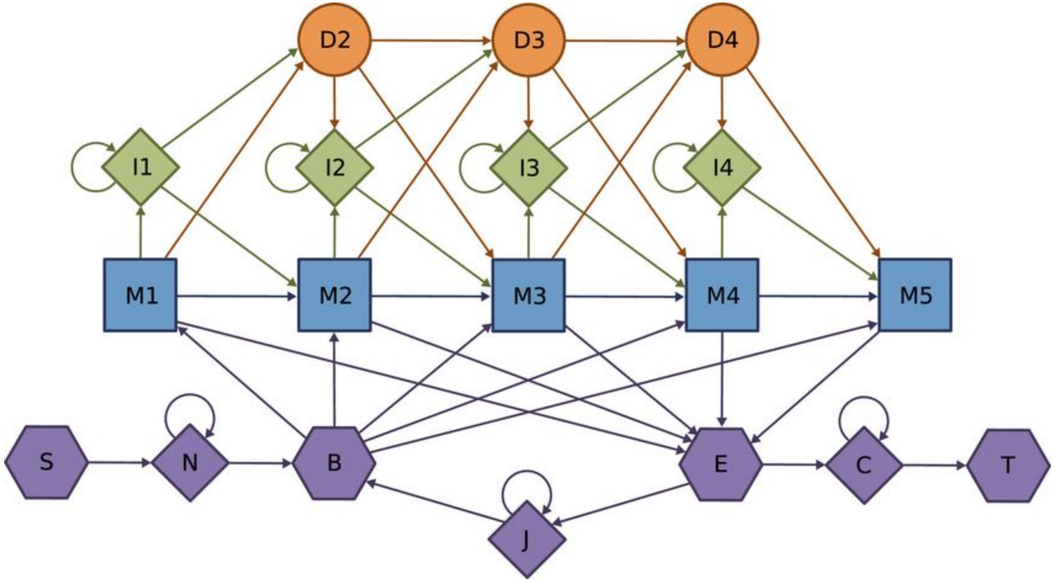
Hidden Markov Model

- Each edge in the graph has a given probability for determining the likelihood of reaching that point in the graph (given the sequences are related).
- This more allows us a much more robust model of the sequences (or sequence families). We can have unique match, insert, and delete probabilities based on position, as well as account for background frequencies of matches.

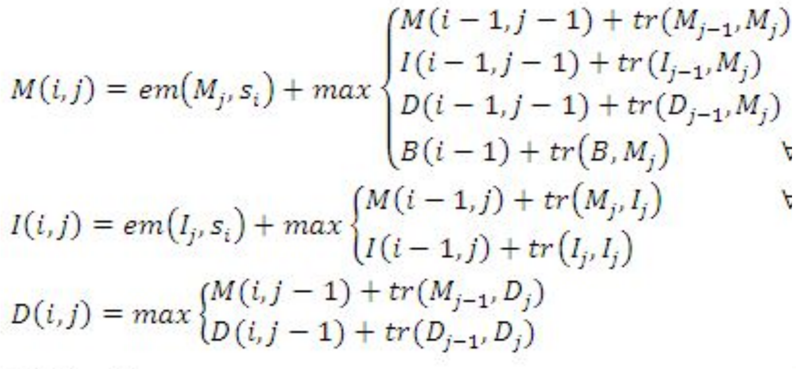


Viterbi Algorithm

- This algorithm closely follows Smith-Waterman. At each state, we can select the maximal probability score of its previous state plus its incoming edge (transition probability).



Viterbi Algorithm

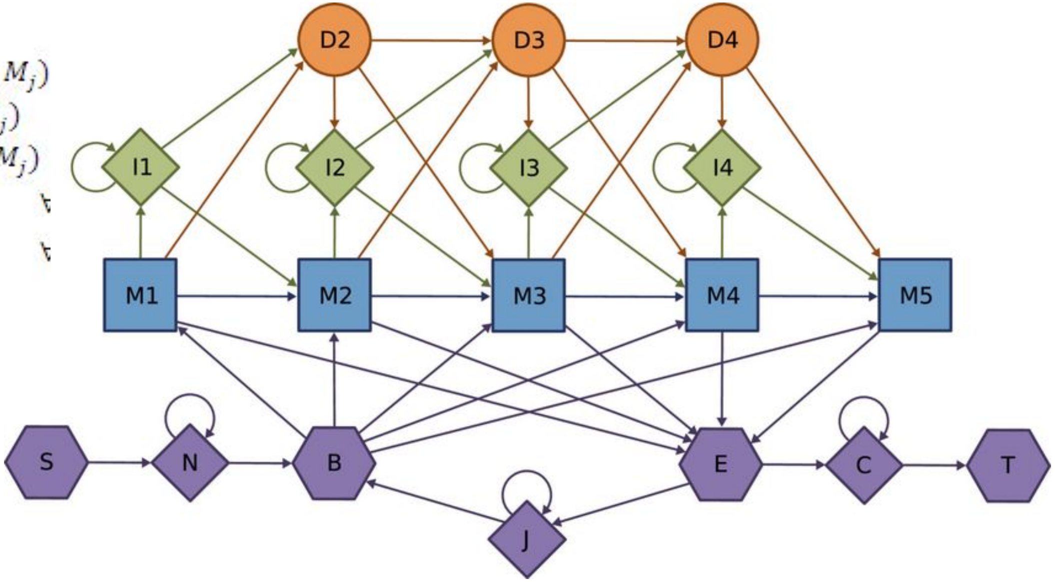


$$M(i,j) = em(M_j, s_i) + max\begin{cases} M(i-1,j-1) + tr(M_{j-1}, M_j) \\ I(i-1,j-1) + tr(I_{j-1}, M_j) \\ D(i-1,j-1) + tr(D_{j-1}, M_j) \\ B(i-1) + tr(B, M_j) \end{cases}$$

$$I(i,j) = em(I_j, s_i) + max\begin{cases} M(i-1,j) + tr(M_j, I_j) \\ I(i-1,j) + tr(I_j, I_j) \end{cases}$$

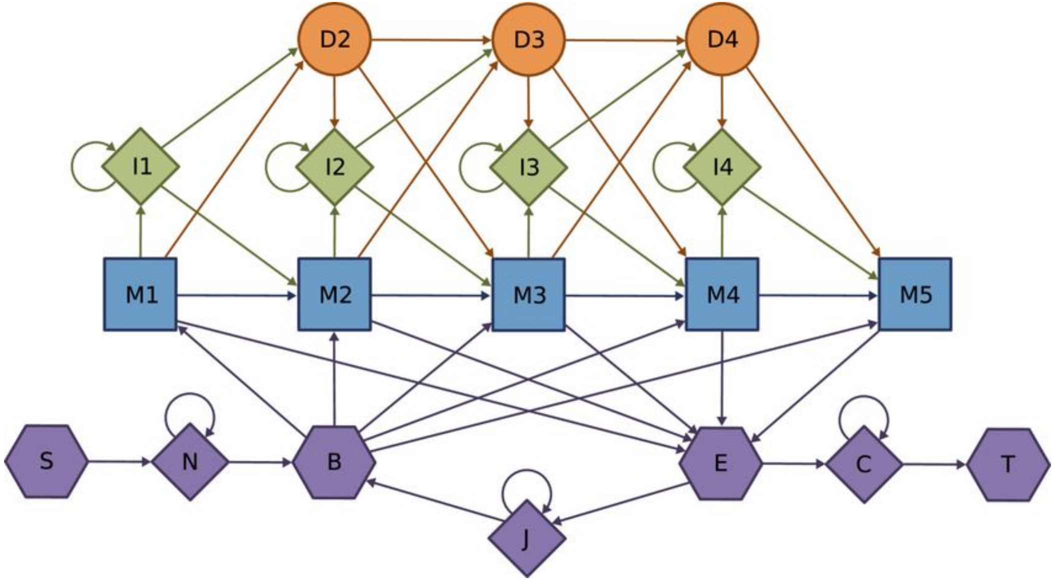
$$D(i,j) = max\begin{cases} M(i,j-1) + tr(M_{j-1}, D_j) \\ D(i,j-1) + tr(D_{j-1}, D_j) \end{cases}$$

- Probabilities are stored in log-space, so multiplication of probabilities becomes addition.



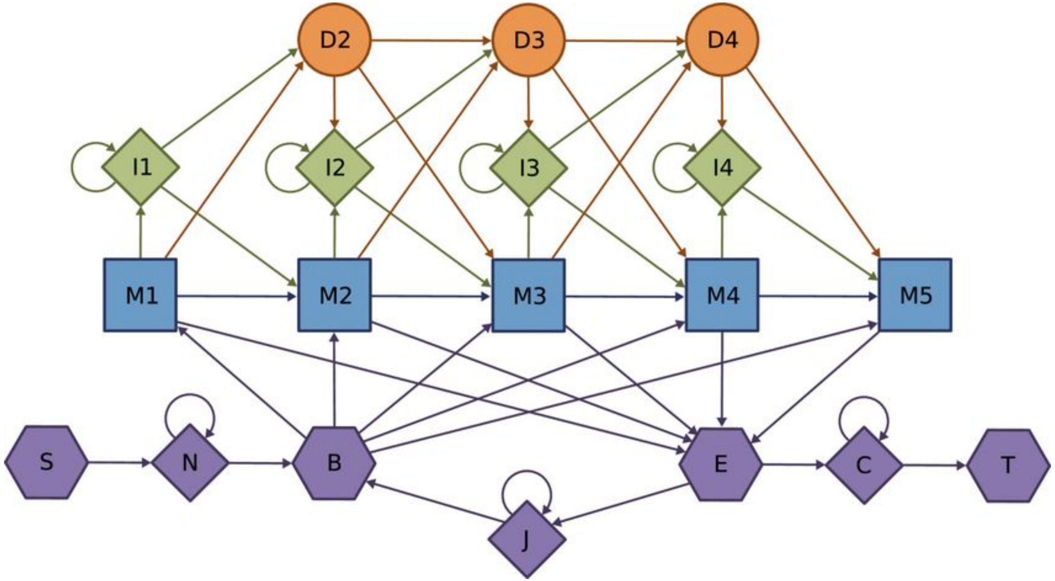
Viterbi Runtime

- For its implementation, Viterbi uses 3 dynamic programming matrices (for Match, Insert, and Delete states).
- It has a runtime only slightly longer than traditional Smith-Waterman.



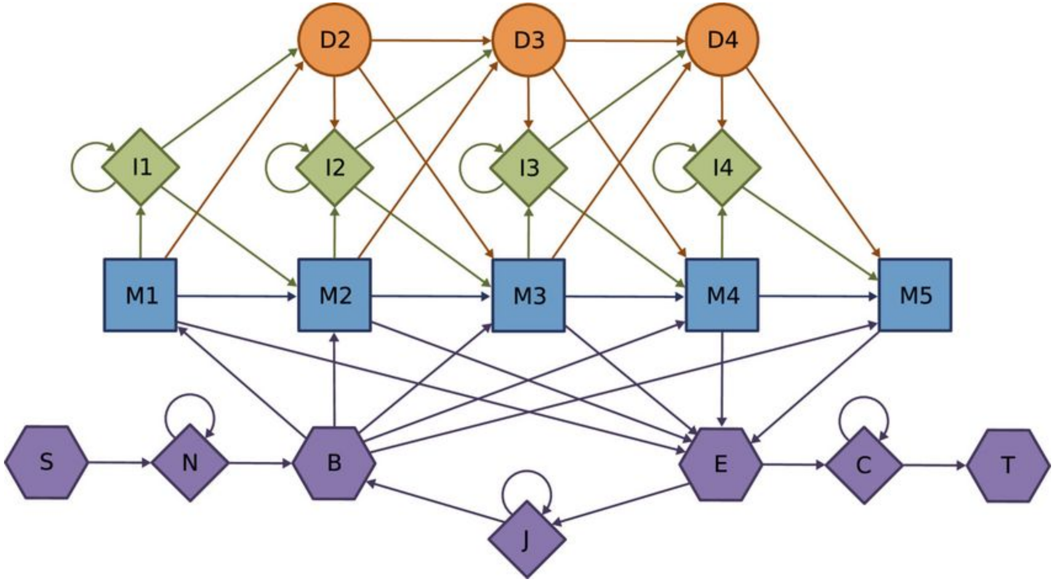
Forward-Backward

- One weakness of Viterbi is it doesn't account for multiple possible strong alignments, since it only accounts for a single maximal alignment.
- This lost data can obscure possible similarities between sequences.



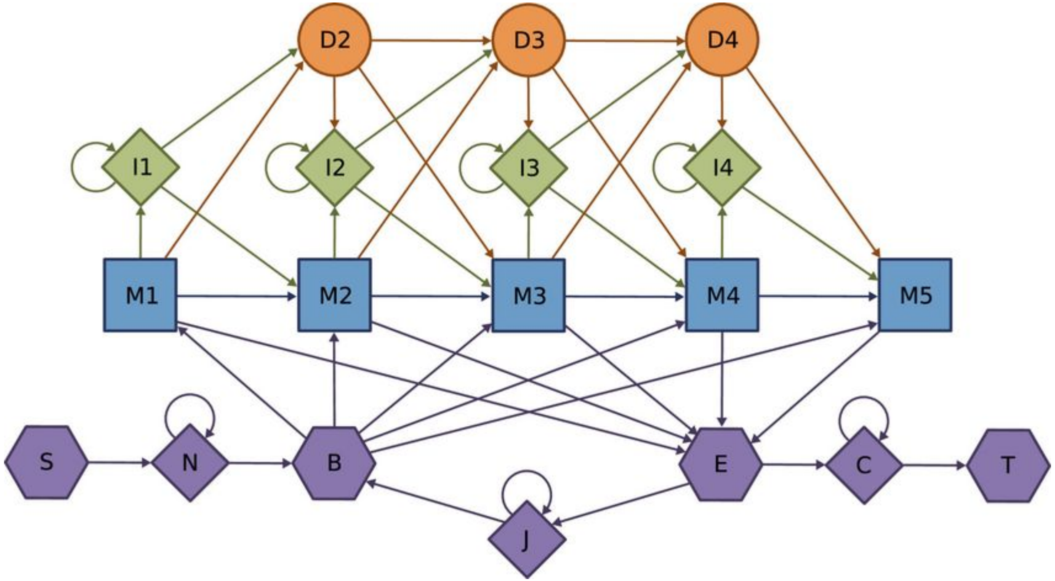
Forward-Backward

- This problem is solved by the Forward-Backward algorithm.
- The difference between Viterbi and Forward-Backward is quite simple: rather than taking the maximal alignment score, it accumulates all possible alignments by summing them.



Forward-Backward Runtime

- There is a major increase in the runtime of Forward-Backward.
- Probabilities are stored in log-space, as multiplication of many low probabilities quickly causes underflow even in high precision data-types.
- Because there is no addition under log-space, this means an expensive process of casting and scaling probabilities.

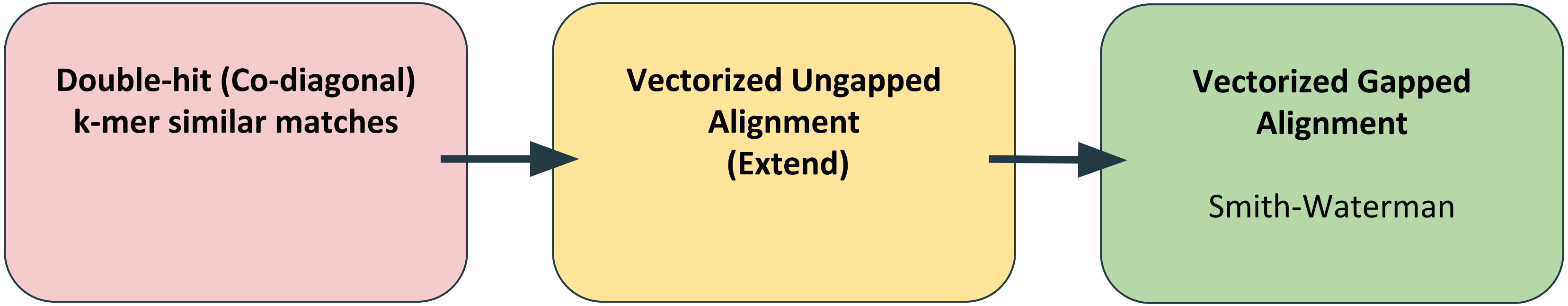


HMMER
& MMSEQS

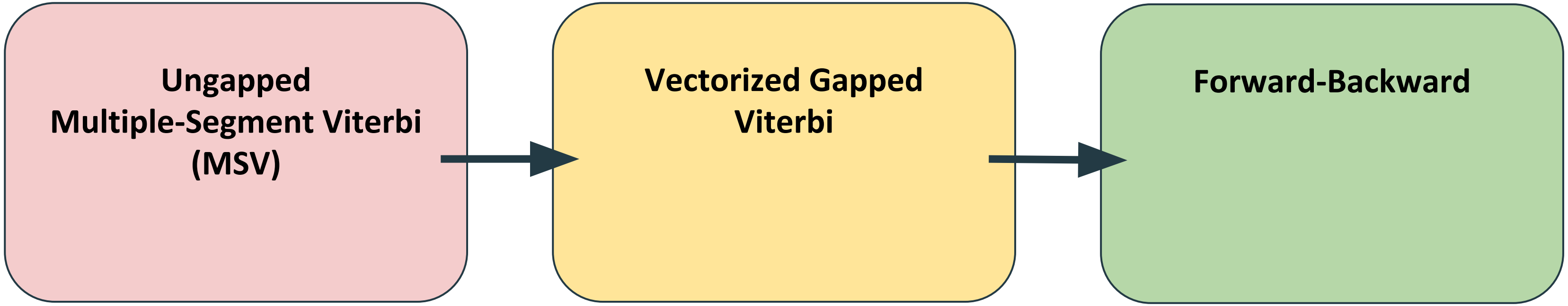
HMMER vs MMSEQS

- HMMER and MMSEQS are two of the more popular sequence alignment software suites.
- For sequence homology search, both suites use a similar approach: Using a pipeline of increasingly sensitive algorithms to filter down the search space.
- In tests, HMMER outperforms MMSEQS in accuraccy, while MMSEQS is generally faster.

MMSEQS Pipeline



HMMER Pipeline



Ungapped Multiple-Segment Viterbi (MSV)

Vectorized Gapped Viterbi

Forward-Backward

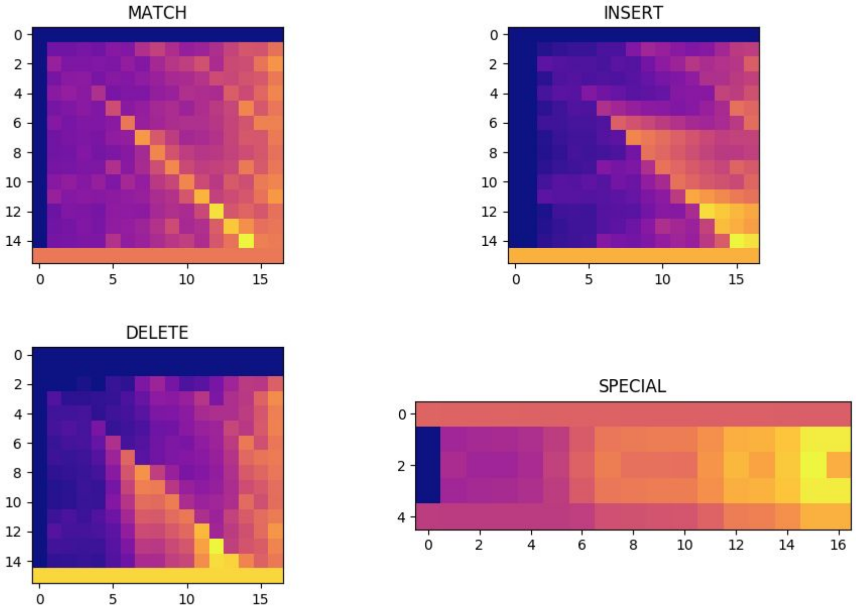
HMMER vs MMSEQS

- Their pipelines are very similar (both use gapped and ungapped Viterbi), apart from a notable difference: HMMER’s use of the Forward-Backward algorithm.
- If we are able to speed up the Forward-Backward algorithm, we may be able to capture the speed of MMSEQS and the accuraccy of HMMER.

Cloud Search:
Pruned Forward-Backward

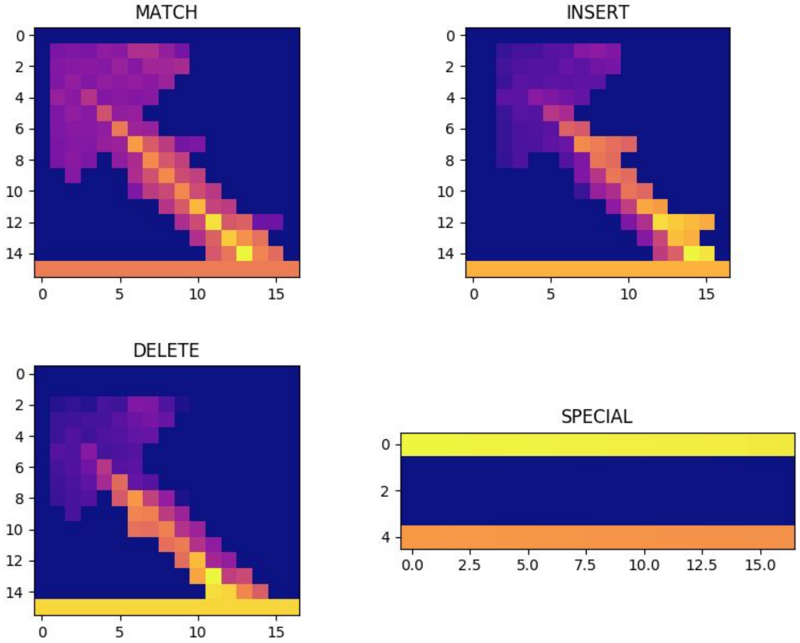
Pruned Forward-Backward

- For our Pruned Forward-Backward, the goal is reduce the search space. Of the N^2 cells to be computed, many of them do not contribute meaningfully to the final score.



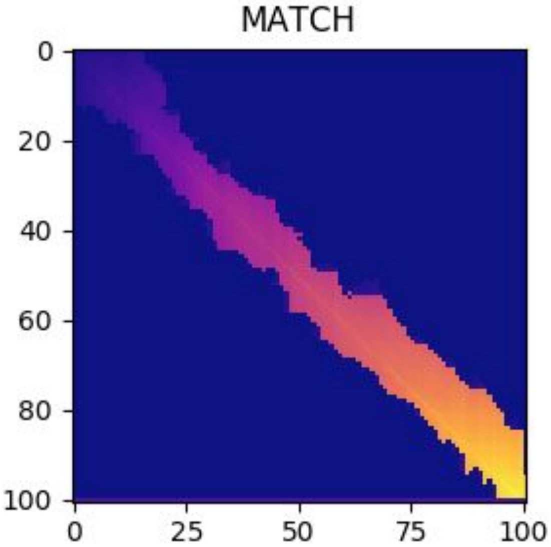
Pruned Forward-Backward

- Rather than computing the matrix row-by-row, we proceed anti-diagonally (every cell on a anti-diagonal is an equal number of edges into the HMM model).
- At each anti-diagonal, we look at the maximum score seen thus far (M). Then we use a tuning parameter k, 0<k<1, and prune all cells that fall beneath kM. The cells dependent to this cell will no longer be computed.



Pruned Forward-Backward

- If we find the proper k value, we should be able to prune a huge portion of the search space (down from quadratic to approximately linear) with minimal cost to accuracy.
- This savings will grow with the size of the window.


MATCH