Comparative Genomics Practical 01
Basic Genome Analysis

Group 6: Tianlin He Xueqing Wang

SUMMARY

In this practical we familiarized ourselves with several nucleotide alignment algorithms, such as BLAST, HMMER, Needleman-Wunsch and Smith-Waterman algorithms. We performed BLAST on our five query genomes assigned in group 6. On the website we chose “nucleotide BLAST”, searched against “nucleotide collection(nr/nt)”, and optimized for “Highly similar sequences”. We didn’t identify the organism or query sequence as we know nothing about them. After running BLAST we chose the sequence with the highest identity to characterize the genomes. We explored their size, kingdom as well as number of genes on NCBI and compared among them to find the uniqueness. Fortunately, the sequence identity of all our 5 query genomes turned out to be high (from 77% to 100%), implying that running BLAST is enough. After these operations we further extended our understandings about the algorithms and solved the questions.

ACTIVITIES

1. Familiarize yourself with BLAST and HMMER algorithms

2. Compare BLAST to HMMER, where do the methods overlap and where are they unique. What are the advantages of each?
Overlap:
- Both of them are database searching method.
- They can make use of a query sequence, while a sequence profile is not compulsory.
- They align the query sequence with sequences in database. [1]
Unique:
- BLAST is a statistical method which evaluates the E value of each result.
- HMMER makes use of the probability model of hidden markov chain to detect distant homologs.
Advantages:
- BLAST strikes a good balance between sensitivity and speed, because it
1) creates a short-hit scoring word list from query, and identify matches of these “words” from database.
2) makes local alignment.
3) evaluates the entire database using the same E value.
- HMMER is very sensitive in detecting distant homologs. [2]

3. Characterize your genomes with BLAST using NCBI website
   a. What organisms genome belongs to?
   - 3: Bacteroides thetaiotaomicron strain 7330 (CP012937)
   - 8: Dictyoglomus turgidum DSM 6724 (CP001251.1)
   - 9: Escherichia coli (HF572917)
   - 18: Synechocystis sp. (PCC6714)
   - 30: Saccharomyces cerevisiae S288C (BK006943)

b. How do the genomes differ (size, kingdom, number of genes)?

Genome	Size / bp	Kingdom	Number of genes
B. thetaiotaomicron	6487685	Bacteria	5145
D. turgidum	1855560	Bacteria	1865
E. coli	5277676	Bacteria	5500
Synechocystis sp.	3485441	Bacteria	3504
S. cerevisiae	12.1571 M	Fungi	7445

c. Does anything interesting standing out?
- S. cerevisiae is the only eukaryotic genome in the list. It has the largest genome size and highest number of genes. However, the ratio of gene : genome size is the lowest, where the rest have the constant number of around 0.001.
- Besides nuclear DNA, some bacteria possess circular plasmid DNA which can be transferred between clones. For example, E.coli genome (no. 9) contains one chromosome DNA and 3 plasmid DNAs.

d. Why aren't we asking you to run HMMER in parallel?
The purpose of this practical is to find out the identity of five unknown genomes, therefore only the result with highest sequence identity and overlap will be considered. Indeed, the sequence identity of all 5 query genomes turns out to be very high (from 77% to 100%). It would be unnecessary to run HMMER, as it looks for all the distant homologs.

CHECK-POINT

1. What is the difference between Needleman-Wunsch and Smith-Waterman algorithms? Why? What are advantages of each?

Firstly, Needleman-Wunsch is the first established, general algorithm for the comparison of the sequence, focusing on the global sequence alignment. Smith-Waterman, on the other hand, is a local sequence alignment algorithm. In this algorithm, instead of focusing on the entire sequence, it compares different segments with different possible lengths and optimize the similarity measurement. Secondly, in Needleman-Wunsch algorithm, for every pair of residues the alignment score has to be positive, at least equal to zero. In Smith-Waterman algorithm the alignment score can be either positive or negative. This is because that in Needleman-Wunsch no gap penalty is used, while in Smith-Waterman a penalty need to be added every time you have a gap.
Thirdly, usually in Needleman-Wunsch the scores don't decrease between two cells along the pathway, while in Smith-Waterman the score can either increase, decrease or just stay the same as that in the former cell. This is also due to the fact that Needleman-Wunsch algorithm makes use of gap penalty. As a result, when tracing back, the cell with the highest score is always at the end (lower-right) of the table while in Smith-Waterman algorithm, the highest-score cell can be in the middle of the table. [3]

The advantage of Needleman-Wunsch algorithm can be interpreted as that, since no gap penalty is added and the global pathway just line up the two sequences and follow the order of the residues on them gradually, is eases the calculation procedure and can get the global alignment within m*n steps. It

fits well with the command when the sequences are of approximately the same length. However, it may not give the alignment with the best score, as it only focus on the global alignment. On the other hand, in Smith-Waterman algorithm, with the use of gap penalty, it takes insertion, deletion and gaps into consideration, thus it is possible to get rid of the length restriction and match the parts which are most similar. It requires a bit more time to run than the Needleman-Wunsch algorithm. It is more suitable when the sequences are similar along some of their lengths but not in others, have different lengths, or share a conserved domain.

2. Is BLAST different from Smith-Waterman algorithm? In what ways?

BLAST is different from Smith-Waterman algorithm in the way that it is a heuristic sequence alignment method while the Smith-Waterman algorithm is dynamic programming. Dynamic programming is sensitive and accurate since it uses all the informations in the two sequences, but it also includes many unnecessary messages in the unaligned areas and thus wastes a lot calculation power and time. On the other hand, BLAST focuses on the highly aligned regions and extends in both directions. This saves the running time as well as reserves quite a lot of sensitivity, although not as accurate as dynamic programming. [4]
Smith-Waterman algorithm is used for the alignment between two sequences (or one sequence and one alignment). It can also be used when aligning several sequences by aligning two sequences at the first time and then aligning the alignment with one other sequence, and repeating the second step over and over until all the sequences are aligned. Meanwhile, BLAST is always used when we want to compare a query sequence to a database of different sequences.
Moreover, the Smith-Waterman algorithm is a manual way and thus is more accurate, but when the number of sequences is huge then BLAST is more efficient, saving much time and can be done on the website.

3. What is the Viterbi algorithm?

The Viterbi algorithm is an algorithm to find the most likely hidden state sequence. It results in a sequence of observed events. [5]
To implement this algorithm, Viterbi use the variable score(k,i), the maximum product weight among all pathways from source to node(k,i). The possible states of this node are considered as all the predecessors of it. The score(k,i), thus, equals to the maximum through all state scores in previous columns multiplied by the weight of the edge(l,k,i-1). [6]It can be specified as the following picture shows:

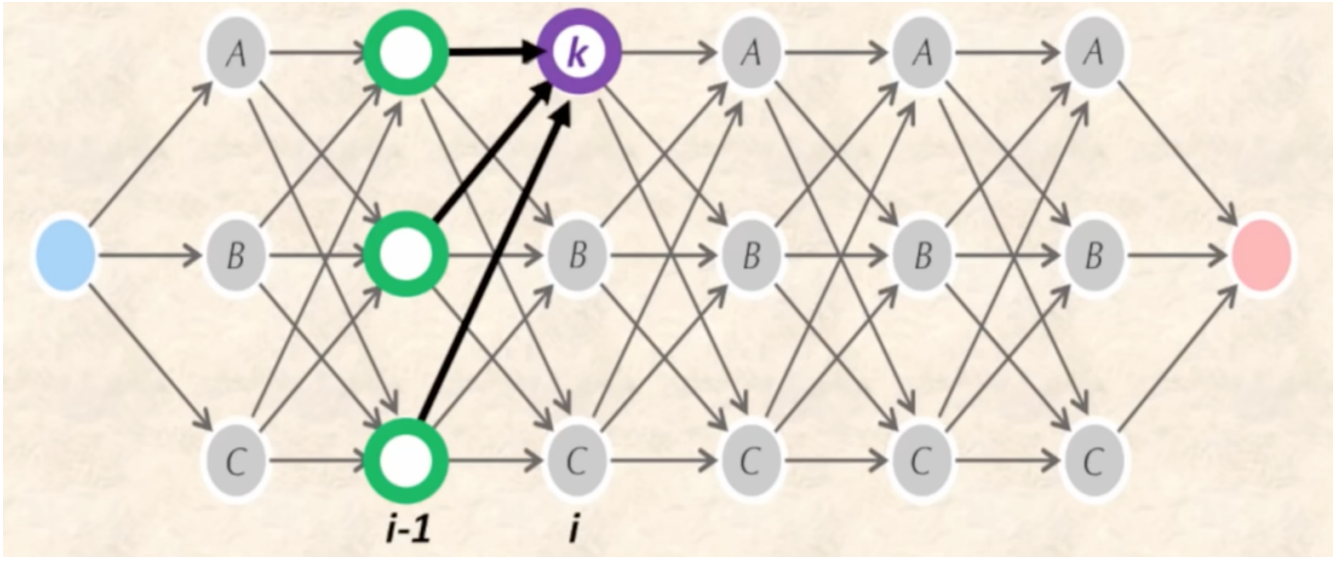

Fig 1-1 Viterbi algorithm [6]

The recurrence of Viterbi algorithm should be the maximum of all states, in other words, the maximum of all the scores multiplied by its weight. The initiation value is 1. The maximum product weight from all source to sink pathways is calculated at the node sink. The score of that sink should be the maximum through all states of the scores in the last Manhatten column. [6]

4. How do these methods compare in speed?

In terms of calculation speed, Viterbi algorithm would be super slow, because that it needs to search all paths to find the most possible path and calculate about the hidden layer. Roughly a full research on a database of 400,000 sequences can take 15 hours. [7]

Needleman-Wunsch and Smith-Waterman algorithms are also extremely slow in the case that they take all the areas of the two query sequences into consideration and thus include many useless regions. They are accurate but waste a lot time. Within them the Smith-Waterman algorithm is even more slower than the Needleman-Wunsch algorithm, since it calculates the local alignment and thus need to consider the insertion, deletion and gap penalty to find the best alignment, while the Needleman-Wunsch algorithm calculates the global algorithm and considers nothing about the gap penalty.

When comparing Needleman-Wunsch to Viterbi algorithm, we can consider the two just as slow, since that running Viterbi algorithm on a pair HMM can be considered as equivalent to Needleman-Wunsch dynamic programming. [8]

And finally, BLAST would be the fastest method among them, since that it lines up all the similar fractions of sequences together and then evaluates the results statistically, avoiding aligning the useless areas in dynamic programming. It is a smart and fast way to evaluate the entire database with the same threshold based on statistics, eliminating noises and reduces the running time.

DISCUSSION

In this practical we operated a series of procedures of nucleotide BLAST, getting a deeper understanding of several important conceptions such as different kinds of databases and optimization methods. We also learned about how to characterize the query genome with the one of the highest identity, as well as how to find the other informations related to the genome. Furthermore, we explored other algorithms and got a rough perception about their conceptions and operation procedures, as well as advantages and disadvantages.

REFERENCE

[1] Zvelebil, Baum. (2008) “Understanding Bioinfomatics”. Garland Science. 342-346.
[2] Zvelebil, Baum. (2008) “Understanding Bioinfomatics”. Garland Science. 430-433.
[3] C.GAYATHRI. Needleman-Wunsch and Smith-Waterman Algorithm. Available from: https://www.scribd.com/doc/17601679/Needleman-Wunsch-and-Smith-Waterman-Algorithm.
[4] Arne Elofsson. (2017) “BLAST”. YouTube. Available from: https://www.youtube.com/watch?v=SUfyUaczbgE&feature=youtu.be.
[5] Feldman J, Abou-Faycal I, Frigo M (2002). "A Fast Maximum-Likelihood Decoder for Convolutional Codes". Vehicular Technology Conference. **1**: 371–375.
[6] Bioinformatics Algorithms: An Active Learning Approach. (2015) “The Viterbi Algorithm”. YouTube. Available from: https://www.youtube.com/watch?v=0dVUfYF8ko0&t=330s.
[7] Arne Elofsson. (2017) “Hidden Markov Models”. YouTube. Available from: https://www.youtube.com/watch?v=ojCzwxgTsEA&feature=youtu.be.
[8] Serafim Batzoglou. “Lecture 7 Sequence Similarity”. CS273: Algorithms for Structure and Motion in Biology. Available from: https://web.stanford.edu/class/cs273/scribing/scribe7.pdf.