Lecture Title and Date

Fast Alignment, 02/10

Objectives of the Lecture

By the end of this lecture, students should be able to:

1. Understand the computational complexity of dynamic programming alignment algorithms (e.g., Needleman-Wunsch, Smith-Waterman) and their limitations in scalability.
2. Explain the evolution of sequence alignment methods from FASTA and BLAST to Burrows-Wheeler Transform (BWT)-based tools like Bowtie.
3. Describe the role of hashing, indexing, and database preprocessing in accelerating sequence alignment.
4. Compare different sequence alignment tools (FASTA, BLAST, Smith-Waterman, HMM) and their trade-offs in speed and sensitivity.
5. Recognize modern challenges in sequence alignment, such as handling long error-prone reads and privacy concerns in personal genome alignment.

Key Concepts and Definitions

- **Computational Complexity**: the measure of algorithm performance in terms of time and memory; sequence alignment using dynamic programming can be $O(n^2)$ or optimized for faster searches.
- **FASTA**: an early alignment tool using hashing of short query words to identify matches in a database.
- **Basic Blast**: an alignment algorithm that indexes the query sequence, finds high-scoring segment pairs (HSPs), and extends them for alignment.
- **BLAT**: a faster alternative to BLAST that indexes the database instead of the query for rapid alignment, sacrificing sensitivity for speed.
- **Burrows Wheeler Transform**: a reversible permutation method for compressing and indexing genomes, enabling efficient substring searches.
- **Burrows-Wheeler Aligner**: an efficient tool using BWT for aligning short reads to a reference genome.
- **Speed vs. Sensitivity Tradeoff**: the balance between how quickly an algorithm runs and how accurately it detects sequence matches.
- **High Scoring Pairs (HSPs)**: sub-sequences with significant similarity found during the BLAST alignment process.

Main Content/Topics

Computational Complexity Challenge

Dynamic programming alignment (Smith-Waterman/Needleman-Wunsch) scales as $O(n\ m) \sim O(n^2)$ in speed and memory, becoming impractical for:

- Database searches (find interested DNA or Protein in database)
- Short-read alignment to a reference database(NGS applications)
- Multi-sequence comparisons

Heuristic Solutions

- FASTA (1980s): **query hashing**
  Uses query sequence hashing with k-tuples (e.g., 6-mer words). Linear DB scanning with hash lookups. Implements diagonal joining for extended matches.
- BLAST (1990s): **more efficient query hashing**
  Adds neighborhood words via substitution matrices (PAM/BLOSUM). Extends High Scoring Pairs(HSPs) without gaps left and right to maximal length. FInds Maximal Segment Pairs(MSOs)between query and databases. Statistical significance was assessed via extreme value distribution (EVD) for p-values.
  Extension is O(N), which takes most of the time in BLAST.
- BLAT (2000s): **hashing the DB**
  Inverts indexing to hash the database (non-overlapping words). Critical for human genome assembly. Trade-off: faster searches but massive precomputed indices.

Original Rotations	Sorted Matrix
0: acaacg$	6: $acaacg
1: caacg$a	2: aacg$ac
2: aacg$ac	0: acaacg$
3: acg$aca	3: acg$aca
4: cg$acaa	1: caacg$a
5: g$acaac	4: cg$acaa
6: $acaacg	5: g$acaac

- BWA/Bowtie (2010s): **BW transform of the DB**

Uses Burrows-Wheeler Transform (BWT) + FM-index for O(n) search complexity. Enables efficient short-read alignment via prefix trees.
Example: Burrows-Wheeler Transform for X = "acaacg$"

1. Build matrix of cyclic rotations of X
2. Sort the matrix alphabetically ($ is considered as smallest)
3. Take the last character of the column as the result of the BWT

Identical characters are more likely to be clustered together for subsequent searches.

Speed-Sensitivity Tradeoff of Different Algorithms

Algorithm	Sensitivity	Speed
BWA Blast FASTA Smith-Waterman PSI-Blast Profiles HMMs	Low High	High Low

Discussion/Comments

- Alignment algorithms scaling to keep pace with data generation. ll methods sacrifice sensitivity for speed. BLAST misses gapped alignments (addressed later in BLAST2).
- BWT Tradeoffs: Require substantial memory for reference genome indexing (~3GB for human genome).
- Challenges:
  1. Long-Read Alignment: PacBio/Oxford Nanopore data (15-100kbp reads) demand new error-tolerant algorithms.
  2. Population Genomics: Aligning personal genomes against 1M+ references with privacy constraints (homomorphic encryption?).
  3. Real-Time Clinical Use: <6hr turnaround for cancer genomics requires GPU/TPU acceleration.

Required Readings

- (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 11, No. 6, 2020 A Categorization of Relevant Sequence Alignment Algorithms with Respect to Data Structures https://pdfs.semanticscholar.org/ce61/04863eedcfaecad98ea2bd31d4c9 435d2b9b.pdf

A Categorization of Relevant Sequence Alignment Algorithms with Respect to Data Structures outlines existing sequence alignment algorithms and the steps required by each. It also describes the computational complexity associated with and best applications for dynamic programming, FASTA and BLAST as heuristic methods, and Clustal families. This paper was easy to understand and was presented in a straightforward manner amenable to students' understanding.

- Altschul, S. F., Gish, W., Miller, W., Myers, E. W., & Lipman, D. J. (1990). Journal of Molecular Biology, 215(3), 403–410. Basic local alignment search tool. https://doi.org/10.1016/s0022-2836(05)80360-2 (http://www.gersteinlab.org/courses/452/10-spring/pdf/Altschul.pdf)

The Methods section of this paper provides an overview of maximal segment pairs and how they are determined in a real-world experiment (i.e. leveraging BLAST to identify sequences with a MSP score above a certain threshold). Interestingly, the authors described how their

methods varied depending on whether their input sequences were of DNA or protein and the resulting impact on sensitivity and selectivity.

References ISL/ESL (if any)

ISL:

- *Chapter 5 on Resampling Methods* discusses techniques such as cross-validation and bootstrapping, which are useful when evaluating the accuracy and computational efficiency of alignment algorithms like BLAST and BLAT. These methods help assess the trade-off between speed and sensitivity in sequence alignment.
- *Chapter 6 on Linear Model Selection and Regularization* covers shrinkage methods such as Ridge Regression and Lasso, which conceptually relate to optimization strategies in sequence alignment algorithms, particularly in balancing computational complexity and accuracy.

ESL:

- *Chapter 7 on Model Assessment and Selection* is relevant to evaluating sequence alignment performance. The trade-offs between sensitivity and computational efficiency in different alignment methods (FASTA, BLAST, BWT) parallel the challenges of bias-variance trade-offs in statistical models.

Other suggested references for many of the key concepts

- Pearson, W. R., & Lipman, D. J. (1988). *Improved tools for biological sequence comparison.* Proceedings of the National Academy of Sciences, 85(8), 2444-2448.
    - This paper introduces **FASTA**, an early heuristic method for fast sequence alignment.
- Kent, W. J. (2002). *BLAT—the BLAST-like alignment tool.* Genome Research, 12(4), 656-664.
    - This paper describes **BLAT**, an alternative to BLAST, which is optimized for genome-wide sequence alignment.
- Li, H., & Durbin, R. (2009). *Fast and accurate short read alignment with Burrows–Wheeler transform.* **Bioinformatics**, 25(14), 1754-1760.
    - This paper introduces **BWA**, which uses the Burrows-Wheeler Transform (BWT) for efficient genome alignment.
- Burrows, M., & Wheeler, D. J. (1994). *A block-sorting lossless data compression algorithm.* Digital Systems Research Center Report, 124.
    - Original paper introducing the **Burrows-Wheeler Transform (BWT)**, later adapted for sequence alignment.

- Pearson W. R. (2014). BLAST and FASTA similarity searching for multiple sequence alignment. *Methods in molecular biology (Clifton, N.J.)*, *1079*, 75–101. https://doi.org/10.1007/978-1-62703-646-7_5
    - This paper describes the basic differences between fast alignment algorithms (particularly between **BLAST** and **FASTA**) and provides practical application guidance for package users.
- Alser, M., Rotman, J., Deshpande, D. *et al.* Technology dictates algorithms: recent developments in read alignment. *Genome Biol* 22, 249 (2021). https://doi.org/10.1186/s13059-021-02443-7
    - Interesting paper about the drivers and necessity of **hashing** and the above-mentioned fast alignment algorithms.
- https://medium.com/@mr-easy/burrows-wheeler-alignment-part-1-eb93057bfff5
    - A great walkthrough of how to apply the **Burrows-Wheeler Transform (BWT)**.
- Pearson W. R. (2016). Finding Protein and Nucleotide Similarities with FASTA. *Current protocols in bioinformatics*, *53*, 3.9.1–3.9.25. https://doi.org/10.1002/0471250953.bi0309s53
    - Practical paper of how to apply **FASTA** and recommended considerations.