60-Minute Discussion Session Plan
Lecture 7: Searching Genomes and Genome Indexing

Course: BINF301 – Computational Biology
Instructor: Tom Michoel
Date: 4/2/2026
Created with Copilot

0-5 min – Warm-Up

Prompts:

- "When you use Ctrl+F or grep, what do you think happens computationally?"
- "Why might naïve string matching fail for gigabase-scale genomes?"
- "Which concept from the pre-read seemed most confusing: Boyer–Moore, suffix arrays, or FM-index?"

Instructor note. Goal: surface intuition and break the assumption that search is trivial. Anchor discussion in Slides 3–6, which show costly worst-case $O(mn)$ naive search.

5–20 min – Guided Concept Walkthrough

Purpose: build shared understanding before group work.

Topics to revisit:

- Why naïve pattern search is slow (Slides 4–6).
- Boyer–Moore logic: bad-character + good-suffix skipping (Slides 9–12).
- Why pattern preprocessing alone is insufficient for large-scale searching.
- Why indexing the *text* (genome) matters (Slides 17–18).

Guiding Questions:

- "How do Boyer–Moore's rules avoid redundant comparisons?"
- "Why is the pattern scanned from right to left in Boyer–Moore?"
- "Why is text indexing crucial for read mapping and multi-query workloads?"

Instructor note. Stress the shift from "searching faster" (Boyer–Moore) to "preprocessing for repeated queries" (indexing). Students should clearly connect the motivation for suffix arrays and FM-index to the impracticality of repeated naïve search.

20–38 min — Structured Small-Group Discussion (Rotating Roles)

Students form groups of 3. Roles rotate every 6–7 minutes.

Roles

- **Summarizer:** explains how k-mer tables and hash tables provide fast fixed-length searches (Slides 19–23, 28–29).
- **Questioner:** asks about tradeoffs in suffix trees, suffix arrays, and FM-index (Slides 32–52).
- **Connector:** links indexing structures to real tools (mappers, aligners, search engines).

Starter Question

Why do we need different indexing structures for fixed-length and variable-length pattern searches?

Instructor note. Expected points:

- k-mer tables excel for fixed-length lookup but fail for variable-length queries (Slides 19–21).
- Suffix trees support arbitrary-length prefix queries but require huge memory (Slides 33–35).
- Suffix arrays provide efficient lexicographic search with smaller space (Slides 36–38).
- FM-index enables compressed and fast backward search (Slides 44–52).

Encourage comparison of memory vs. time tradeoffs.

40–50 min — Mini Applied Exercise Block

See handout.

Instructor note. Prompt reasoning, not computation. Guide learners toward:

- Understanding skip heuristics conceptually.
- Seeing k-mers as fast fixed-length seeds.
- Understanding how suffix arrays support prefix search.
- Grasping that BWT is reversible and basis for FM-index.

50–60 min – Synthesis Discussion

Prompts:

- "If you needed to index the human genome on a laptop, which structure would you choose and why?"
- "How does the FM-index achieve both small size and fast lookup?"
- "What characteristics of genomes make indexing harder than indexing typical English text?"

Instructor note. Key memory comparisons from slides:

- Suffix tree: >45 GB for human genome (Slide 35).
- Suffix array: ∼12 GB (Slide 37).
- FM-index: ∼1.5 GB (Slide 52).

Push students toward articulating why compression and rank/select operations achieve speed + small footprint.