Assignment: Investigating the EM Algorithm in kallisto

MDGE 610: Foundations of Bioinformatics

Due: Two weeks from assignment date

Overview

In this assignment, you will investigate the Expectation-Maximization (EM) algorithm as implemented in kallisto, a widely-used tool for RNA-seq transcript quantification. You will read about the EM algorithm, examine how kallisto applies it, and conduct empirical experiments to evaluate the convergence criteria used in the software.

This assignment has both conceptual and practical components. You will need to:

- Read and understand the EM algorithm
- Examine kallisto's source code
- Modify, compile, and run kallisto with different settings
- Analyze results and draw conclusions

Background

The Transcript Quantification Problem

RNA sequencing produces millions of short reads that originate from transcribed RNA molecules. A fundamental challenge is estimating how many RNA molecules came from each transcript (gene isoform) in the original sample. This is complicated by the fact that many reads are *multi-mapping*—they are compatible with multiple transcripts due to sequence similarity among gene family members, splice isoforms, and repetitive elements.

kallisto's Approach

kallisto [**?**] uses *pseudoalignment* to rapidly determine which transcripts each read could have originated from, then applies the EM algorithm to estimate transcript abundances. The key insight is that reads can be grouped into *equivalence classes* based on which transcripts they are compatible with, and the EM algorithm can work with these equivalence class counts rather than individual reads.

The EM Algorithm

The Expectation-Maximization algorithm [**?**] is a general method for maximum likelihood estimation when some data are missing or unobserved. In transcript quantification, we observe which transcripts each read *could* have come from, but not which transcript it *actually* came from. The EM algorithm iteratively:

1. **E-step**: Estimates the probability that each read came from each compatible transcript, given current abundance estimates
2. **M-step**: Updates abundance estimates based on these probabilistic assignments

The algorithm is guaranteed to increase (or maintain) the likelihood at each iteration, eventually converging to a local maximum.

Required Reading

- Dempster, A.P., Laird, N.M., and Rubin, D.B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B*, 39(1):1–38. [**?**]

This classic paper introduces the EM algorithm and establishes its theoretical properties. Focus on understanding the general framework (Sections 1–3) rather than all the specific applications.

Approaching the DLR Paper

The Dempster-Laird-Rubin paper is a foundational work in statistics, but its mathematical notation and generality can be challenging. You are encouraged to use modern AI tools to help you navigate the material. A recommended approach:

1. Upload the DLR paper to a tool like Google's NotebookLM to generate an initial summary and enable interactive Q&A with the document.
2. Use a frontier LLM (e.g., Claude, ChatGPT, Gemini) to help "translate" the mathematical notation in Section 2 (General Properties) into plain English.
3. **Critical step:** Verify the LLM's explanations by cross-referencing specific equations in the paper. Do not blindly trust the output—use it to unlock the primary text, not replace it.

The goal is understanding, not just getting answers. If an LLM explains something, make sure you can trace that explanation back to the paper itself.

Provided Data

You are provided with simulated RNA-seq data from the human protein-coding transcriptome. Using simulated data allows us to know the "ground truth" abundances and evaluate estimation accuracy.

File	Description
`gencode.v44.kidx`	Pre-built kallisto index (GENCODE v44 protein-coding transcripts)
`sim_reads_1.fastq.gz`	Simulated paired-end reads (read 1)
`sim_reads_2.fastq.gz`	Simulated paired-end reads (read 2)
`sim_true_counts.txt`	Ground truth transcript counts

Table 1: Provided data files for the assignment.

Files

Data Characteristics

- **Reference**: GENCODE v44 human protein-coding transcriptome (110,962 transcripts)
- **Read pairs**: 1,000,000
- **Read length**: 75 bp paired-end
- **Fragment length**: Mean 250 bp, SD 50 bp
- **Sequencing error**: 1% per-base

The ground truth file (`sim_true_counts.txt`) is tab-delimited with columns `transcript_id` and `true_counts`.

Assignment Questions

Prepare a written report addressing the following questions. Your report should be clear, well-organized, and include evidence (figures, tables, or specific observations) supporting your conclusions.

Part 1: Conceptual Understanding

(1) The EM Algorithm. What is the EM algorithm? Describe how it works in your own words:

- What objective function does it maximize?
- What are the E-step and M-step, conceptually?
- What constitutes the "missing data" in kallisto's application of EM?

(2) kallisto's Objective Function. Examine the kallisto paper and/or source code to understand the specific likelihood function being maximized.

- What is the mathematical form of the objective function?
- How does kallisto's EM implementation maximize it?
- What convergence criteria does kallisto use to decide when to stop iterating? Where are these specified in the code?
- How might these criteria be justified?

(3) Local vs. Global Maxima. The EM algorithm is guaranteed to find a local maximum, but not necessarily the global maximum. Different starting points could, in principle, lead to different solutions.

- Examine how kallisto initializes its abundance estimates.
- Do you think initialization matters for this problem? Why or why not?
- *Hint*: Consider what fraction of reads map to a single transcript (uniquely) versus multiple transcripts (ambiguously).

Part 2: Empirical Investigation

(4) Testing Convergence Criteria. You will empirically test how kallisto's EM convergence criteria affect estimation accuracy.

Setup:

- Obtain the kallisto source code from `https://github.com/pachterlab/kallisto`
- Create a git repository for your work. Make an initial commit containing the original, unmodified source code.
- Compile kallisto following the instructions in the repository.

Experiments:

- Modify the kallisto source code to vary the EM convergence behavior (e.g., number of iterations, convergence threshold, or other relevant parameters).
- For each modification, commit your changes to git with a descriptive commit message.
- Run kallisto on the provided simulated data and compare the estimated abundances to the ground truth.
- You may choose how to assess accuracy (e.g., correlation, relative error, or other metrics you find appropriate).

Write-up:

- Describe what modifications you made and why.
- Present your results clearly (tables and/or figures).
- Interpret your findings: Are the convergence criteria used in the kallisto paper adequate? What evidence supports your conclusion?

Deliverables

1. **Written report** (PDF) addressing all four questions. There is no strict page limit, but aim for clarity and concision. Include relevant figures and tables.
2. **Git repository** containing:
   - Initial commit with original kallisto source
   - Subsequent commits documenting each modification you tested
   - Any analysis scripts you wrote

   You may submit this as a link to a GitHub/GitLab repository or as a zipped archive of the repository (including the `.git` directory).

Evaluation Criteria

Your report will be evaluated on:

- **Understanding**: Do you demonstrate a clear understanding of the EM algorithm and its application in kallisto?
- **Rigor**: Are your experiments well-designed? Do you test a reasonable range of conditions?
- **Analysis**: Do you interpret your results thoughtfully? Do you consider alternative explanations?
- **Reproducibility**: Can your experiments be reproduced from your git repository and description?
- **Communication**: Is your report clear, well-organized, and appropriately concise?

Tips

- Start early. Compiling software from source and troubleshooting build issues takes time.
- Use `git diff` to review your changes before committing.
- If you get stuck on the code, focus on understanding what you *can* find and document your process.
- There is no single "right answer" to the final question. A well-reasoned conclusion supported by evidence is what matters.

Office Hours and Q&A

Class time will be allocated for questions and working on the assignment. Come prepared with specific questions or issues you've encountered.

References

[1] Bray, N.L., Pimentel, H., Melsted, P., and Pachter, L. (2016). Near-optimal probabilistic RNA-seq quantification. *Nature Biotechnology*, 34(5):525–527.

[2] Dempster, A.P., Laird, N.M., and Rubin, D.B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–38.