Combining Metrics to Improve RNA-Seq Quantification

1st Sathvik Ravi
CMSC701 Spring 2024
University of Maryland
College Park, MD
sathrav5@umd.edu

2nd Bilal Mohammed
CMSC701 Spring 2024
University of Maryland
College Park, MD
bmohamm7@umd.edu

3rd Pranav Sivaraman
CMSC701 Spring 2024
University of Maryland
College Park, MD
psivaram@umd.edu

Abstract*—Transcript quantification from RNA sequencing (RNA-seq) data plays a pivotal role in understanding gene expression dynamics, particularly in complex processes like alternative splicing. This paper explores novel strategies to improve RNA-seq quantification by intelligently combining Maximum Likelihood Estimators (MLEs), specifically Expectation-Maximization (EM) and Variational Bayesian Expectation-Maximization (VBEM). We begin by reviewing existing quantification methods such as Sailfish, kallisto, and Salmon, which employ distinct algorithms for estimating transcript abundances. While EM and VBEM offer robust approaches for likelihood optimization, their standalone applications present limitations in accuracy and convergence. Our investigation introduces a weighted average scheme, integrating EM and VBEM likelihoods to achieve refined abundance estimates. To determine optimal weights, we devise an iterative algorithm that minimizes the Mean Absolute Relative Difference (MARD) between estimated counts and ground truth data. The proposed method demonstrates superior accuracy compared to standalone EM and VBEM approaches, as evidenced by reduced MARD values. Furthermore, we present modifications to the Salmon algorithm to incorporate our weighted combining strategy seamlessly. Experimental results validate the efficacy of our approach, showcasing its potential to enhance RNA-seq quantification accuracy and reliability.

I. Introduction

Alternative splicing is a process by which different exons from the same gene are joined in various combinations, leading to the production of different but related mRNA transcripts. Transcript quantification, which determines the steady-state abundance of these alternative transcripts within a sample, has numerous valuable applications. It can be used to detect biomarkers for diseased and normal tissue, understand how transcript expression levels change during organismal development, and track the progression of cancer. RNA-seq, a high-throughput sequencing technique, is employed to analyze the transcriptome of a cell or organism by generating millions of short sequence reads, facilitating these critical insights. Mapping the short-sequence reads from RNA-seq back to the transcriptome is an ongoing challenge. Various generative models have been proposed for the read generation process, and by optimizing their likelihood, we obtain abundance estimates of the transcripts. This work explores ways to improve existing quantification methods by intelligently combining different maximum likelihood estimates to potentially improve the final abundance estimates.

II. Related Work

A. Existing Quantification Methods

The Sailfish method [1] significantly increases the speed of abundance estimation by avoiding read mapping altogether and instead using counts of k-mers to estimate transcript coverage. The kallisto method [2], which is even faster, calculates the probability that each read originates from a given transcript through pseudoalignment, achieved via fast hashing of k-mers and the transcriptome de Bruijn graph (T-DBG). The Salmon method [3] employs a dual-phase inference procedure, combining information about the position and orientation of mapped fragments with abundances from online inference to compute per-fragment conditional probabilities. These probabilities are then used to estimate auxiliary models and bias terms, and update

abundance estimates using iterative methods such as Expectation-Maximization (EM) and Variational Bayesian Expectation-Maximization (VBEM).

B. Maximum Likelihood Estimators

The EM algorithm optimizes the likelihood for the estimated counts of fragments derived from each transcript, given the set of equivalence classes of fragments. It approximates this likelihood by collapsing fragments into equivalence classes and maximizing the product of transcript abundances and the affinity of each transcript for each equivalence class. This is done by updating the estimated read count for each transcript, normalizing it by the estimated number of reads across all transcripts, weighted by the equivalence class affinity for the current transcript, and taking the weighted sum across all equivalence classes.

The VBEM algorithm, on the other hand, aims to infer the posterior distribution of nucleotide fragments given the transcriptome and observed fragments. It finds the approximate posterior that best matches the true posterior by minimizing the Kullback-Leibler (KL) divergence between them. VBEM updates are performed by taking the weighted sum across equivalence classes of the affinity for each transcript, normalized by the sum of affinity values across the specified equivalence class. Once the parameters converge, an estimate for the expected value of the posterior nucleotide fractions is obtained. VBEM leverages a prior for each transcript, which represents the number of reads per nucleotide. With a larger prior, VBEM estimates more non-zero abundances than EM, and fewer non-zero abundances with a lower prior. VBEM also tends to converge after fewer iterations than EM.

In our investigation, we compare the number of iterations and the mean absolute relative difference of EM, VBEM, a simple averaging scheme of EM and VBEM, and L-BFGS weighted optimization EM and VBEM. This comparison helps in understanding the trade-offs between these methods when computing the posterior nucleotide distribution.

III. Methods

A. Ground Truth from RSEM-Simulations

In order to ensure that we intelligently choose the weights for our VBEM and EM updates, we need to ensure that we have ground truth to train on. To obtain ground truth, we use the RSEM-Simulations [4] described in the original Salmon [3] paper. We believe these are a good choice for comparison with the original paper, as well as the fact that the RSEM simulations are accurate. For each result file, we use the MARD metric and optimize it accordingly.

B. Objective Function

To see if combining EM and VBEM values yields more accurate gene counts, we wanted to compute the average between the two likelihoods. There are two ways to complete this task: one is by finding the average of the two values, giving equal importance to both EM and VBEM in the final value. The other is by calculating the weighted average of the two likelihoods, with the weights determined by how effectively they minimize the difference between the ground truth counts and the estimates. Equation 1 shows the equation for finding the weighted average for set with N values. Prior to any work being done, we believed that the weighted average would produce the best counts because of its ability to leverage the more important feature and increase its influence in the final output.

C. Iterative Algorithm

The main problem that needs to be solved relating to the weighted mean between EM and VBEM is finding weight values associated with both probability outputs such that the weighted mean minimizes the error of a loss function. We had two methods we tried to solve for these weights. One of these methods was to create a nested iterative update task that simultaneously finds optimal values for EM, VBEM, and their respective weights. Each iteration step is as followed:

1) Initialize α values and α' to the default set by the original Salmon algorithm.
2) Run VBEM and EM update rules to get α' values for their respective algorithms.simultaneously finds optimal values for EM, VBEM, and their respective weights.
3) Complete L-BFGS minimization optimization algorithm to obtain new weights for iterative step that minimizes MARD.
4) Compute the new α' value by calculating the weighted mean as shown in Equation 1.

TABLE I: Accuracy of different weighting methods. (Lower is Better)

Method	MARD
EM	0.126
VBEM	0.106
Simple Mean	0.124
Weighted Mean (alpha)	0.48
Simple Mean Final Iteration (EM + VBEM)	0.126

5) Check to see if cutoff values met to end iteration. If so, return the VBEM and EM values alongside their weights. Otherwise, repeat steps 2-5.

Each iteration is only done if certain conditions are met. In our case these conditions are related to whether the weighted average is able to converge to a real number. If this criteria is not met, then another iteration is completed. Within each iteration, there is the VBEM and EM update is done as described in the original Salmon [3] paper, where α is used to update α' for their respective probability metrics. However, unlike the original work, we are running both EM's and VBEM's updates in the same iteration loop and adjusting each value based on their respective weights as well as a jointly computed α'. This new value α' will be calculated as the new weighted mean as shown in Figure 1.

In order to figure out the weights to be used to find α', which will ultimately help us find the final weights and probability values, we implemented another iterative optimization function that minimizes the Mean Absolute Relative Difference (MARD) as described by Patro et. al (2017) [3] of the ground truth data and the following function:

$$Y = (\omega_0 EM + \omega_1 VBEM)/(\omega_0 + \omega_1) \quad (1)$$

where ω_0 and ω_1 are the calculated weights. This was done using L-BFGS optimization. L-BFGS is an optimization method used to minimize an objective function over unconstrained values for a real-vector x. It employs the first derivative of the objective function to identify the direction of steepest descent and uses the second derivative to compute the inverse Hessian, capturing the curvature information about the objective function. L-BFGS utilizes a two-loop recursion procedure to approximate the inverse

Algorithm 7.5 (L-BFGS).
Choose starting point x_0, integer $m > 0$;
$k \leftarrow 0$;
repeat
  Choose H_k^0 (for example, by using (7.20));
  Compute $p_k \leftarrow -H_k \nabla f_k$ from Algorithm 7.4;
  Compute $x_{k+1} \leftarrow x_k + \alpha_k p_k$, where α_k is chosen to satisfy the Wolfe conditions;
  if $k > m$
    Discard the vector pair $\{s_{k-m}, y_{k-m}\}$ from storage;
  Compute and save $s_k \leftarrow x_{k+1} - x_k$, $y_k = \nabla f_{k+1} - \nabla f_k$;
  $k \leftarrow k + 1$;
until convergence.

Fig. 1: L-BFGS Algorithm

Hessian matrix and determine the direction and step size for updating the current solution toward the minimum of the objective function. Figure 1 shows pseudocode for the algorithm and its iterative nature [5]. By finding weights for the weighted average such that we minimize MARD, we wanted to see if we can get a weighted combination of the two likelihoods that will outperform EM and VBEM individually as well as a simple mean approach.

Another approach that we explored was to optimize the parameters for EM and VBEM before completing the minimization task for the weights. To do this, we computed the outputs for EM and VBEM as done by Patro et. al (2017) [3]. Once those values converged, we then completed the L-BFGS optimization algorithm to find ω_1 and ω_2 given the EM and VBEM abundances for each gene in the simulated data. This was done as an alternative to the method disucssed above to see if leveraging the current iterative adjustments where EM and VBEM are computed separately for more accurate results when they are combined after converging.

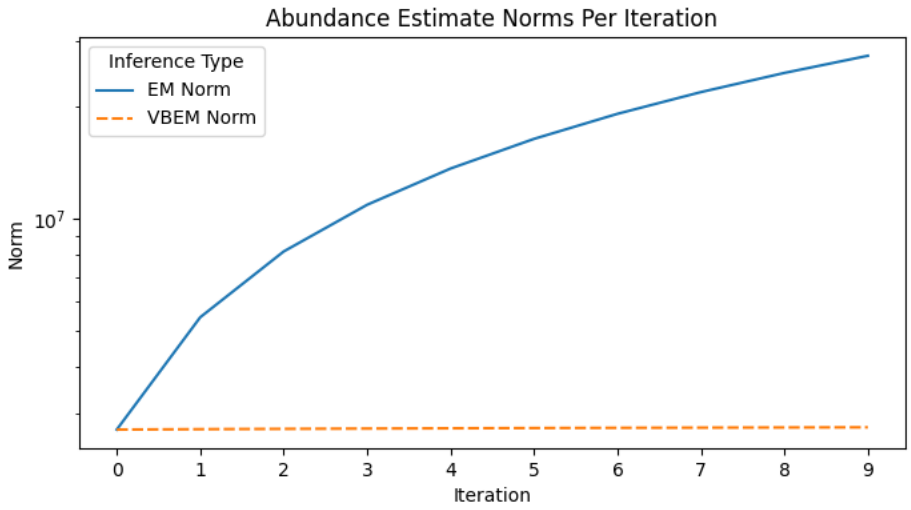

Fig. 2: Norm of VBEM and EM Abundance Estimates

IV. Results

Figure I displays the relationships between different weighting methods. The baseline methods include VBEM and EM, while our new results are derived from various weighted combinations of these two methods.

Figure 2 shows the magnitude of the EM and VBEM abundance estimates over 10 iterations. From this, we can see that the gradient of the objective function with respect to EM is much larger than the gradient with respect to VBEM. The L-BFGS minimization refines the sequence of estimates by using the first-order gradient, which explains why the updated inverse Hessian optimizes the function by moving in the direction of EM and away from VBEM. Furthermore, the Wolfe line search [6], which chooses a search direction and step size to minimize the objective function, ensures that the curvature condition is satisfied and that the L-BFGS updating is stable. However, in our optimization problem, the MARD function is not second-order differentiable, causing instability in the updates to the weights of the objective function.

V. Conclusion

Although we could show that intelligently averaging VBEM and EM drastically impacts abundance estimates, it is still worth investigating how different weighting systems can maximize the accuracy of these estimates. With an appropriate method for calculating weights for VBEM and EM, we can potentially uncover relationships in gene abundances across datasets. Exploring optimization approaches that do not heavily rely on the data's gradient would allow us to compute weights that are not biased towards EM, which typically has a much larger gradient compared to VBEM. Additionally, experimenting with other functions to minimize, such as cosine similarity, Spearman correlation, and mean-squared error, could reveal more distinctions between EM and VBEM, aiding in the development of weights that better maximize abundance counts.

Once better weights are computed, it is crucial to test their generalizability to other datasets. If the same proportions apply across different datasets, it suggests consistent underlying conditions between EM and VBEM, helping to identify similarities between datasets and relationships between individual genes and their counts. By diving into the specifics of the VBEM and EM algorithms, we might leverage properties that enhance the accuracy of abundance counts, potentially leading to a novel maximum likelihood estimator optimized for quantifying RNA sequences.

Overall, our project aims to improve the understanding of the relationship between VBEM and EM and explore how they can be used together to enhance effectiveness. This will deepen our comprehension of gene abundances within and across datasets. The work from our project can be applied

to future methods investigating the relationships between that aim to improve quantification methods and better understand why certain genes are present at the abundance that they do.

REFERENCES

[1] R. Patro, S. M. Mount, and C. Kingsford, "Sailfish enables alignment-free isoform quantification from rna-seq reads using lightweight algorithms," *Nature biotechnology*, vol. 32, no. 5, pp. 462–464, 2014.

[2] N. L. Bray, H. Pimentel, P. Melsted, and L. Pachter, "Near-optimal probabilistic rna-seq quantification," *Nature biotechnology*, vol. 34, no. 5, pp. 525–527, 2016.

[3] R. Patro, G. Duggal, M. I. Love, R. A. Irizarry, and C. Kingsford, "Salmon provides fast and bias-aware quantification of transcript expression," *Nature methods*, vol. 14, no. 4, pp. 417–419, 2017.

[4] B. Li and C. N. Dewey, "Rsem: accurate transcript quantification from rna-seq data with or without a reference genome," *BMC bioinformatics*, vol. 12, pp. 1–16, 2011.

[5] D. C. Liu and J. Nocedal, "On the limited memory bfgs method for large scale optimization," *Mathematical programming*, vol. 45, no. 1, pp. 503–528, 1989.

[6] Y.-H. Dai and C.-X. Kou, "A nonlinear conjugate gradient algorithm with an optimal property and an improved wolfe line search," *SIAM Journal on Optimization*, vol. 23, no. 1, pp. 296–320, 2013. [Online]. Available: https://doi.org/10.1137/100813026