Final paper

Annie Dai[1†], Emma Shroyer[1†], Nathalie Bonin[1†]

[1]Computer Science Department, University of Maryland, College Park, Maryland, USA.

Contributing authors: anniedai@umd.com; eshroyer@gmail.com; nbonin@gmail.com;
[†]These authors contributed equally to this work.

Abstract

The abstract serves both as a general introduction to the topic and as a brief, non-technical summary of the main results and their implications. Authors are advised to check the author instructions for the journal they are submitting to for word limits and if structural elements like subheadings, citations, or equations are permitted.

Keywords: Salmon, Tuna, Onefish, Twofish, Redfish, Bluefish

1 Background

Transcriptome introduction: splicing, gene expression, SVs, etc

Quick explanation of RNA-Seq and its role in quantifying transcriptome

Problem: The existence of different isoforms makes the problem of transcript quantification difficult. While we also want to map a read to a gene, we also want to get an estimate of the abundances of each isoform given a set of reads. .

1.1 Salmon

Salmon utilizes raw reads or transcriptome aligned reads in a quasi-mapping phase.[1] It then moves to an online inference phase where it uses the EM algorithm to estimate the 'foreground' bias models and compute the equivalence class weights. The execution then moves to offline inference where the user can specify the inference algorithm of EM or VBEM. The program gives the initial abundances and fragment equivalence classes to these algorithms. Optionally, using the converged abundances and fragment

equivalence classes, Salmon draws and saves estimates from the posterior distribution using Gibbs or bootstrap sampling. Salmon utilizes relatively little memory to quantify abundance estimates from RNA sequence reads. Given a known transcript and a set of sequenced fragments, Salmon aims to quantify the relative abundance of each transcript in the sample. Salmon takes into account sample-specific biases to improve accuracy. When these biases are not accounted for, calculations like the false discovery rate cannot be controlled for. Using multiple inference steps, Salmon estimates abundances using either the VBEM or EM inference algorithms. However, both these methods have drawbacks. They both return point estimates of abundances, but we are uncertain about the returned estimates without ground truth. Further, the choice of Bayesian priors used in the VBEM algorithm also has considerations. A small prior leads to sparser results than EM. However, a larger prior may result in more estimated non-zero abundance than EM. Prior simulated tests [**?**] show than VBEM with a small prior can lead to more accurate estimates. Given the trade offs of the two algorithms, we sought to know if the averaged prediction for EM and VBEM inference is better than one algorithm or another. Further, if this ensemble estimate is better, could it be improved by varying the Bayesian priors.

1.2 EM & VBEM

The Expectation Maximization (EM) algorithm optimizes the likelihood of the parameters given the data. This algorithm returns a point estimate of the abundances. This algorithm is the default of Salmon.
The Variational Bayesian Expectation Maximization (VBEM) algorithm accounts for the sparsity of the data. It takes a Bayesian nucleotide prior that controls for the sparsity of the data. The default prior for Salmon was 1×10^{-2}.

Each transcriptome $\mathcal{T}$ is composed of transcripts t. Each transcript is a nucleotide sequence which can be described through its length l, effective length $\tilde{l}$, and its count c—which is the number of times that t occurs in a given sample. There are M transcripts in a given transcriptome.

The probability that a given fragment originates from a transcript t depends on the length of that transcript relative to all other transcripts in the transcriptome. We define this nucleotide fraction η for a given transcript t as

$$\eta_i = \frac{c_i \cdot \tilde{l}_i}{\sum_{\boldsymbol{j}=1}^{M} c_j \cdot \tilde{l}_j}$$

We obtain a transcript fraction τ for a given transcript t by normalizing its nucleotide fraction against the effective length of all transcripts.

$$\eta_i = \frac{c_i \cdot \tilde{l}_i}{\sum_{\boldsymbol{j}=1}^{M} c_j \cdot \tilde{l}_j}$$

Let's say that the true nucleotide fraction for a transcript t is $\boldsymbol{\eta}$. We can describe the probability of observing a set of sequenced fragments $\mathcal{F}$ as a

$$\Pr\left(\mathcal{F}|\boldsymbol{\eta}, Z, \mathcal{T}\right) = \prod_{j=1}^{N} \Pr\left(f_j|\boldsymbol{\eta}, Z, \mathcal{T}\right) = \prod_{j=1}^{N}\sum_{i=1}^{M} \Pr\left(t_i|\boldsymbol{\eta} \cdot \Pr\left(f_j|t_i, z_{ij}=1\right)\right)$$

where N is the number of fragments in $\mathcal{F}$, Z is a relationship matrix where $z_{ij} = 1$ when fragment f_j is derived from t_i.

We want to obtain $\boldsymbol{\alpha}$, which is the estimated number of reads from each transcript. We describe the maximum likelihood estimates as:

$$\mathcal{L}\left(\boldsymbol{\alpha}|\mathcal{F}, \boldsymbol{Z}, \mathcal{T}\right) = \prod_{j=1}^{N}\sum_{i=1}^{M} \hat{\eta}_i \Pr\left(f_j|t_i\right)$$

Written in terms of equivalence classes $\boldsymbol{C}$

$$\mathcal{L}\left(\boldsymbol{\alpha}|\mathcal{F}, \boldsymbol{Z}, \mathcal{T}\right) = \prod_{\mathcal{C}^j \in \boldsymbol{C}} \left(\sum_{t_i \in \boldsymbol{t}^j} \hat{\eta}_i w_t^j\right)^{d^j}$$

The abundances $\hat{\boldsymbol{\eta}}$ are computed directly from $\boldsymbol{\alpha}$

$$\hat{\eta}_i = \frac{\alpha_i}{\sum_j \alpha_j}$$

Then we apply an update function

$$\alpha_i^{u+1} = \sum_{\mathcal{C}^j \in \boldsymbol{\mathcal{C}}} d^j \left(\frac{\alpha_i^u w_i^j}{\sum_{t_k \in \boldsymbol{t}}} j\alpha_k^u w_k^j\right)$$

Until the maximum relative difference in $\boldsymbol{\alpha}$ is

$$\Delta\left(\alpha^u, \alpha^{u+1}\right) = \max \frac{\left|a_i^u - \alpha_i^{u+1}\right|}{\alpha_i^{u+1}} < 1 \times 10^{-2}$$

for all $\alpha_i^{u+1} > 1 \times 10^{-8}$, at which point we derive the estimated nucleotide fraction

$$\hat{\eta}_i = \frac{\alpha_i'}{\sum_j \alpha_j'}$$

Variational Bayes Optimization

Optionally, the we can apply variational bayeseian optimization where the update function is

$$\alpha_i^{u+1} = \sum_{\mathcal{C}^j \in \boldsymbol{\mathcal{C}}} d^j \left(\frac{e^{\gamma_i^u} e_i^j}{\sum_{t_k \in \boldsymbol{t}} j e^{\gamma_k^u} w_k^j}\right)$$

where

$$\gamma_i^u = \Psi\left(\sum_{k=1}^{M} \alpha_k^0 + \alpha^u{+}_k\right)$$

where Ψ is the digamma function and the expected value of nucleotide fractions can be expressed as

$$\mathbb{E}(\eta_i) = \frac{\alpha_i^0 + \alpha_i'}{\sum_j \alpha_j^0 + a_j'} = \frac{a_i^0 + \alpha_i'}{\hat{\alpha}^0 + N}$$

$$\hat{\alpha}^0 = \sum_{i=1}^{M} \alpha_i^0$$

1.3 Datasets

We used synthetic datasets used in Salmon. We generated our synthetic data using polyester 1.16.0 and alpine version 1.6.0. [citation] There was ground truth available for this dataset.

We additionally used real human datasets. We used a dataset consisting of NK cells, T cells, and tumor from matched soft tissue sarcoma and peripheral blood.

We also used the human transcriptome reference: Homo sapiens GRCh38 cDNA sequence. These were gene models built from alignments of the human proteome and alignments of human cDNAs.

2 Methods

3 Experiment

We began by editing Salmon's source code. We wrote a bash script that downloaded our datasets. We also had a script that ran our samples using EM, VBEM, and a combination of the VBEM and EM algorithms. First we ran unaltered Salmon using EM and VBEM. For each pass that used VBEM, we specified a prior 10-5 through 101. We then ran the same datasets using the average of both the EM and VBEM algorithms. We ran the EM and VBEM algorithms on the same data, producing an array. FOr each index in this array, the data was averaged and inserted into a new array that was used the final calculations. We again used priors of 10-5 through 101 when considering the VBEM algorithm. We ran this on [Nathalie's laptop] Finally, we compared the results of these passes using the ground truth of the synthetic data. We calculate the MARD of our results. We also calculated the Spearman Correlation and a two-sided Mann-Whitney U test.

4 Results

Initially, we ran EM and VBEM with varying priors by themselves without any alterations to Salmon. We found the following results:

5 Evaluation

$$\mathrm{ARD}_i = \begin{cases} 0 & \text{if} \quad x_i = y_i = 0 \\ \dfrac{|x_i - y_i|}{x_i - y_i} & \text{otherwise} \end{cases}$$

$$\mathrm{MARD} = \frac{1}{M} \sum_{i=1}^{M} \mathrm{ARD}_i$$

5.1 Spearman Correlation

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)}$$

5.2 Mann–Whitney U Test

6 Discussion

We were unable to expand on our project with the time given but in future work, we would like to expand upon ways to best combine the results of the EM and VBEM algorithms. Currently we are only taking the mean of the two results. However, we would also like to experiment with the weighted average of these two algorithms also taking into account various prior sizes. We would like to see if this also improves the estimates of Salmon.

We would also like to verify the time it takes to run our combined approach. We would like to verify that running this approach will not greatly impact the runtime of Salmon. While we expect the runtime to increase; however, we would like to verify that it is not detrimental to the use of Salmon.

In future work, it will be beneficial to get a measure of uncertainty when running our trials. We were unable to run this during our initial experiments. However, we would like to know if there is a level of uncertainty with our results when we use the average of the EM and VBEM algorithms. We would like to know if this uncertainty increases or decreases, and if it does so significantly.

References

[1] Patro, R., Duggal, G., Love, M.I., Irizarry, R.A., Kingsford, C.: Salmon provides fast and bias-aware quantification of transcript expression. Nature Methods **14**(4), 417–419 (2017) https://doi.org/10.1038/nmeth.4197 . Accessed 2024-05-15