A revisit of RSEM generative model and its EM algorithm for quantifying transcript abundances.

Hy Vuong¹ Thao Truong¹ Thang Tran¹ Son Pham^{1*}

Bioturing Inc.

December 21, 2018

Abstract

RSEM has been mainly known for its accuracy in transcript abundance quantification. However, its quantification time is extremely high compared to that of recent quantification tools. In this paper, we revised the RSEM's EM algorithm. In particular, we derived accurate M-step updates to eliminate incorrect heuristic updates in RSEM. We also implement some optimizations that reduce the quantification time about a hundred times while still have better accuracy compared to RSEM. In particular, we noticed that different parameters have different convergence rates, therefore we identified and removed early converged parameters to significantly reduce the model complexity in further iterations, and we also use SQUAREM method to further speed up the convergence rate. We implemented these revisions in a packaged named Hera-EM, with source code available at: <https://github.com/bioturing/hera/tree/master/hera-EM>

1 Introduction

RNA-Seq has become a standard approach for quantifying transcript abundances. Reads with multiple mapping locations (multi-mapping reads) are the main obstacle in quantifying. Many tools have been developed to resolve this problem. The simplest solution is ignoring multi-mapping reads, and just counting the number of unique reads for each transcript. This approach has limitations [6]. To take into account information from multi-mapping reads, EM solutions were developed [2, 6, 10, 11]. Using a comprehensive generative model, RSEM [6] has been known for its accuracy in transcript abundance quantification [2, 9]. However, given the large amount of RNA-Seq data generated, the running time of RSEM is a major disadvantage. This gives rise to a new class of more efficient tools e.g. Sailfish, Salmon, Kallisto [2, 10, 11]. Besides using pseudo-alignment instead of alignment. Kallisto, and Sailfish implement simpler EM solutions that group reads mapping to the same set of transcript into equivalent classes¹ which may lead to loss of mapping positions, fragment lengths, alignment score information.² In the case of multi-mapping reads, these variables are important to find the best assignments of reads to transcripts. As a result, Kallisto and Sailfish, while being much faster, cannot reach the accuracy level of RSEM [2, 9].

In the work of Li et. al., 2011 [6], while the generative RSEM model was constructed comprehensively, the authors failed to derive accurate update formulae for the models' parameters in

*Corresponding Author. Email: sonpham@bioturing.com

¹We also noticed that IsoEM [8] uses a different equivalent relation definition to build equivalent classes, which also requires the conditional probabilities of the alignments to be the same. Therefore, no information will be lost. This technique can be applied to our method as well. However, we think it is not a very significant optimization in our case as the number of equivalent classes would not be significantly reduced.

²When feeding with full alignment information, Salmon [10, 17] uses those data in the online phase and then the mapping conditional probabilities of each transcript are averaged for each equivalent class. While some information is still being lost due to averaging, it is less significant than other tools

their EM algorithm as the difficulty of the analytic work. In this manuscript, we keep the essence of RSEM’s model, and provide accurate formulae for parameter updates in the EM algorithm by finding the solutions analytically. With the accurate parameter update formulae, we achieve better results than RSEM in the SMC DREAM challenge data sets. We note that the data sets from the SMC DREAM challenge were generated using the RSEM generative model.

Besides, we observed that different transcripts have different convergence rates for their abundance estimations. Therefore, by avoiding the recalculation of early converged estimations, we can save a huge amount of computation. Specifically, if we detect a parameter unchanged for a few EM rounds, we will keep the parameter fixed in further iterations. Hence, we can skip the recalculation of the joint probability on those transcripts. This is our main contribution in speeding up the EM process. In addition, we also implement the SQUAREM method [16] to improve the convergence rate.

Hera-EM is about a hundred times faster, at the same time, gains equal or even better accuracy compared to the EM procedure in RSEM. On a data set with 60 million of reads, RSEM takes about an hour (3432 seconds) for EM step only, while Hera-EM needs half a minute (24 seconds); for an other data set with 75 million of reads, RSEM takes about 1.5 hours (5044 seconds), while Hera-EM takes 39 seconds. Hera-EM consumes about 7 GB of memory in all of these benchmarks.

2 Method

Notations

Let $R = \{r_i = (r_1^i, r_2^i)\}$ be a set of N read pairs; $T = \{t_i\}$ be a set of M transcripts. Transcript t_i has length l_i . For each read $r = (r_1, r_2)$, we denote $A_r = \{a_i\}$ as the set of alignments, where each alignment a_i is a tuple (t, f, s, o) of transcript t , fragment length f , starting position s , and orientation o . We also denote t_a, f_a, s_a, o_a as the original transcript, fragment length, starting position and orientation of an alignment a , respectively.

Generative Model

The process of generating a read is described in the following generative process.

- Draw transcript t_i from the set of all transcript T with probability θ_i .
- Choose a fragment length f_i of the read with probability depends on the the universal fragment length parameter λ adjusted by the length of transcript t_i .
- Choose a starting position s_i on transcript t_i of the fragment with a universal probability π adjusted based on the length of transcript t_i , and the fragment length f_i .
- Choose an orientation o_i of the read pair with probability ω of being forward-stranded.
- Adding sequencing errors, with each position has the probability ϵ of being erroneously sequenced.

The details of the above parameters $\theta, \lambda, \pi, \omega, \epsilon$ are explained below.

Definition of the distributions and parameters

θ : Transcript distribution

For transcript distribution, we denote θ_t as the probability that a transcript t is chosen from all transcripts to generate a fragment.

$$P(t = t_a \mid \theta) \triangleq \theta_{t_a}$$

Figure 1: Bayesian graph for read generating process. $r = (r_1, r_2)$ is the only observed variable; t_i, f_i, s_i, o_i are the latent variables; $\theta, \lambda, \pi, \omega$, and ϵ are the parameters

λ : Fragment length distribution

Given parameter λ as the distribution of the fragment length generated from a hypothetical infinitely long transcript. In the generative model of fragments on a particular transcript, this parameter needs to be adjusted according to the length of the transcript, as fragments cannot be longer the transcript itself. We set the probability of the all fragments that are longer than the transcript to zero and normalize the total probability to 1.

Denote $l = |t|$ as the length of the chosen transcript t .

The probability of choosing a fragment length f_a :

If $0 \leq f_a \leq l$

$$P(f = f_a | t, \lambda) \triangleq \frac{\lambda_{f_a}}{\sum_{i=0}^l \lambda_i}$$

Otherwise

$$P(f = f_a | t, \lambda) \triangleq 0$$

π : Read start position distribution

As many studies have observed that reads are not generated uniformly along the transcripts [4, 6, 12], Li and Dewey [6] modeled the read start distribution by dividing each transcript into 20 bins and count the frequency of reads starting in each bin. We also model this distribution by dividing each transcript into a fixed number of segments (64 by default) and assign a common probability π_i to the i^{th} segment of every transcript. Given a hypothetical fragment of length 1, π_i represents the probability of this fragment originated from the segment i . Within each segment, the starting positions of the fragments are uniformly distributed. With fragment lengths larger than 1, not every starting position on the transcript is possible because fragments can not end outside the transcript, so we need to adjust the distribution.

Let π_i be the probability that a hypothetical fragment of length 1 is originated from the segment i . Denote b as the number of segments of each transcript, l as the length of the chosen transcript.

We define $g(k)$ as the cumulative distribution function for the hypothetical fragment of length 1.

$$g(k) = P(s \leq k | t, f = 1, \pi) = \hat{\pi}_{[h]} + \{h\} \times \pi_{[h]}$$

Where $\hat{\pi}_k = \sum_{i=0}^{k-1} \pi_i$, and $h = \frac{k}{l} \times b$.

With a fragment of length f originated from transcript t of length l , the fragment can only start from $0 \leq s_a \leq l - f$, so we have the following definition for $0 \leq s_a \leq l - f$,

$$P(s = s_a | f, t, \pi) \triangleq \frac{g\left(\frac{s_a + 1}{l}\right) - g\left(\frac{s_a}{l}\right)}{g\left(\frac{l - f + 1}{l}\right)}$$

Otherwise,

$$P(s = s_a | f, t, \pi) \triangleq 0$$

ω : Orientation distribution

We assume the distribution of the read orientation o is a Bernoulli distribution with parameter ω . $o = 1$ means the read is forward-stranded, and $o = 0$ means the read is reverse-stranded. We can set $\omega = \frac{1}{2}$ in unstranded protocol, and $\omega = 1$ in stranded protocol, or by default it is estimated from the data.

$$\begin{aligned} P(o = 1 | \omega) &\triangleq \omega \\ P(o = 0 | \omega) &\triangleq 1 - \omega \end{aligned}$$

ϵ : Error rate

Sequencing error also needs to be accounted for. We assume a simple distribution, every base has the same probability for being erroneously sequenced³, denoted as ϵ . Therefore a read r with edit distance e to the reference has the probability of $\epsilon^e \times (1 - \epsilon)^{|r| - e}$.

Given an alignment $a(t, f, s, o)$ of the read $r(r_1, r_2)$, let d be the edit distance between the read and the reference of the aligned position.

$$P(r | t, f, s, o, \epsilon) = P(e = d | t, f, s, o, \epsilon) \triangleq \epsilon^d \times (1 - \epsilon)^{|r_1| + |r_2| - d}$$

Likelihood function

Denote Ω as the set of all parameters $\theta, \lambda, \pi, \omega$, and ϵ . From Fig. 1, we derive a formula for joint probability of read r and alignment $a(t, f, s, o)$ given parameter Ω .

$$P(r, a | \Omega) = P(t | \theta) \times P(f | t, \lambda) \times P(s | f, t, \pi) \times P(o | \omega) \times P(r | t, f, s, o, \epsilon) \quad (1)$$

We also have the conditional probability,

$$P(a | r, \Omega) = \frac{P(r, a | \Omega)}{P(r | \Omega)}$$

With $P(r | \Omega) = \sum_{a \in A_r} P(r, a | \Omega)$

We define the likelihood function.

$$\mathcal{L}(\Omega) = \prod_{r \in R} \sum_{a \in A_r} P(r, a | \Omega) = \prod_{r \in R} P(r | \Omega)$$

Where A_r is the set of all alignments of a read pair r . R is the set of all read pairs.

Our goal is to find the parameter Ω that maximizes the likelihood function $\mathcal{L}$. In this work, we use an optimized EM algorithm to obtain the desired parameters. First we initialize our distributions as uniform distributions, and then applying the following EM update iteratively until the relative change of all elements in the parameter θ is below 10^{-3}

³We decided to use a "spatially-homogeneous" model for sequencing error, specifically, we only use the edit distance of the alignment and ignore the read quality and the position of those error bases, as the accuracy loss due to this simplification is minimal, while gaining a huge computational advantage.

EM update

For each iteration of the EM algorithm, from the old parameter $\Omega^{(i)}$, we need to find a new parameter $\Omega^{(i+1)}$ that maximizes this function.

$$\begin{aligned} Q(\Omega \mid \Omega^{(i)}) &= \mathbb{E}_{A \mid R, \Omega^{(i)}} [\log P(R, A \mid \Omega)] \\ &= \sum_{r \in R} \mathbb{E}_{a \mid r, \Omega^{(i)}} [\log P(r, a \mid \Omega)] = \sum_{r \in R} \sum_{a \in A_r} \log(P(r, a \mid \Omega)) \times P(a \mid r, \Omega^{(i)}) \end{aligned}$$

Where A is the set of alignments of R .

Taking partial derivatives of $Q(\Omega \mid \Omega^{(i)})$ by the following variables θ , λ , π , ω , ϵ and equating them to zero, we can solve for the roots of those equations with the constraints that the sum of each variable equals to 1 (See supplementary for detailed calculations). This gives us the following update formulae.

Transcript distribution

For each transcript t , we have the following update,

$$\theta_t^{(i+1)} = \frac{1}{|R|} \times \sum_{r \in R} \sum_{a \in A_r, t_a=t} P(a \mid r, \Omega^{(i)})$$

Fragment length distribution

$$\lambda_f^{(i+1)} = \frac{\frac{X_f}{Z_f}}{\sum_k \frac{X_k}{Z_k}}$$

Where $X_f = \sum_{r \in R} \sum_{a \in A_r, f_a=f} P(a \mid r, \Omega^{(i)})$, $Y_f = \sum_{r \in R} \sum_{a \in A_r, f=|t_a|} P(a \mid r, \Omega^{(i)})$, $Z_0 = 1$, and $Z_f = Z_{f-1} - \frac{Y_{f-1} \frac{X_i}{Z_i}}{\sum_{i \leq f-1} \frac{X_i}{Z_i}}$

Read start position distribution

Let $j_a = \left\lfloor \frac{|t_a| - f_a + 1}{|t_a|} \times b \right\rfloor$, and $\beta_a = \left\lfloor \frac{|t_a| - f_a + 1}{|t_a|} \times b \right\rfloor$

Let $X_k = \sum_{r \in R} \sum_{a \in A_r, q_a=k} P(a \mid r, \Omega^{(i)})$

Let $Y_k = \sum_{r \in R} \sum_{a \in A_r, \beta_a=k} \frac{1}{\hat{\pi}_{\beta_a} + j_a * \pi_{\beta_a}} \times P(a \mid r, \Omega^{(i)})$

$$\frac{X_{i+1}}{\pi_{i+1}} = \frac{X_i}{\pi_i} - Y_{i+1}$$

Let $Z_k = \frac{X_k}{\pi_k}$, $\hat{Y}_k = \sum_{i=1}^k Y_i$

$$Z_i = \frac{X_i}{\pi_i} = Z_0 - \hat{Y}_i$$

We cannot eliminate the variable π_i in $\hat{Y}_i$ so we use an approximation from the previous value of π_i . We can easily compute the estimation of Y_i alongside with the EM algorithm. However there is still an unsolved variable Z_0 , we use the Newton method [1] to solve the equation $\hat{\pi}_b = 1$ to get the value of Z_0

After finding Z_0 , we have

$$\pi_k^{(i+1)} = \frac{X_k}{Z_k}$$

Orientation distribution

$$\omega^{(i+1)} = \frac{1}{|R|} \times \sum_{r \in R} \sum_{a \in A_r, o_a=1} P(a \mid r, \Omega^{(i)})$$

Error rate

We have the following update formula,

$$\epsilon^{(i+1)} = \frac{\sum_{r \in R} \sum_{a \in A_r} d \times P(a \mid r, \Omega^{(i)})}{\sum_{r \in R} |r|}$$

Inaccuracy in RSEM parameter updates

While our generative model is the same as RSEM, the update formulae for the parameters π and λ are different. RSEM does not solve the partial derivative equations to derive the formulae for these two parameters, but rather uses intuition to come up with incorrect update formulae.

In particular, in the RSEM published source code, the parameter $\lambda^{(i+1)}$ is updated as follow, $\lambda_f^{(i+1)} = \frac{X_f}{\sum_k X_k}$, which is not the optimal result for this optimization. Similar to fragment length distribution, RSEM uses an intuitive update formula $\pi_f^{(i+1)} = \frac{X_k}{\sum_f X_k}$ for the parameter π of read start position distribution. This leads to underestimating read coverage in the 3' tails. The error in the update of *read start position* distribution affects the results significantly. In the Results section, we show that RSEM incorrectly estimates the read start distribution, and leads to lower correlations with the simulated truth.

Optimizations

As the number of reads is extremely high in each run, it yields a heavy computational cost in each naive EM iteration. Many recent methods try to reduce the complexity by grouping similar reads⁴ into equivalent classes [2, 10, 11]. This approach significantly speeds up the computation, however, with a notably loss in accuracy. In Hera-EM, our goal is to preserve the accuracy, while reducing the quantification time.

Similar to RSEM, we only run full EM iterations for a few first iterations (default of 30 in our implementation), after that we run simplified EM iterations, which only θ is being updated, other parameters are kept fixed. This saves a lot of computation as other parameters tend to converge after only a few rounds of EM. Besides, we only need to keep the edit distance for each alignment instead of the whole read sequence to calculate the joint probability. Therefore, we can store the whole alignment data on RAM with even less memory than RSEM. Hence, we don't have to reread the alignment file in every EM round. This also significantly speeds up each EM iteration.

An interesting observation is that after just a few EM rounds, only a small number of elements in parameter θ are still being changed as most of them have already been converged early. This leads us to a more sophisticated optimization. If we detect a parameter θ_i of a transcript have a relative change smaller than a threshold of 10^{-5} in a few EM rounds, the parameter will be kept fixed in further iterations. This means the joint probability of the alignments on these transcripts will not change in further iterations and we don't need to update the parameters θ_i of those transcript, therefore they can be safely removed from the future iterations. We only need to keep the total joint probability of the removed alignments for each read to calculate the conditional

⁴Reads that map to the same set of transcripts

probability of other alignments. Applying this optimization, we see a dramatic reduction of data complexity. This is our main optimization of the EM algorithm.

We noticed that TIGAR2 [7] utilizes a similar optimization, which stops the re-evaluation of reads that mapped onto the transcripts with converged parameters. However, there is a notably difference between TIGAR2 and our method. In our method, we don't wait until all alignments of a read to be converged to stop re-evaluating that read. We stop the re-evaluating of an alignment immediately after that alignment is converged, independent from other alignments of the same read.

We also ignore the improbable alignments with conditional probability $P(a | r, \Omega)$ less than a defined threshold. Through experimenting, we notice that while a smaller threshold of 10^{-6} give a nearly identical result compared to no-filtered version, higher threshold of 10^{-3} is notably faster but can cost accuracy. Hence we choose a default of 10^{-6} . In our simulated data sets, about half of the alignments are redundant due to very low conditional probability, therefore, this optimization reduces about half of the quantification time.

Finally, we also implement SQUAREM [16] to speed up the convergence rate.

TPM value

Similar to RSEM [6], we defined the effective length of the transcript t as $\tilde{l}_t = \sum_f \max(0, |t| - f + 1) \times \lambda_f$.

The TPM value of a transcript t is $10^6 \times \frac{\frac{\theta_t}{\tilde{l}_t}}{\sum_{t \in T} \frac{\theta_t}{\tilde{l}_t}}$

3 Results

We benchmarked Hera-EM algorithm against RSEM using two simulated data sets generated by SMC DREAM challenge [14]. The first data set denoted as *sim4x* contains 5 samples: 41, 42, 43, 44, 45; each sample has 60 million reads. The second data set, denoted as *sim5x*, is an improved version of *sim4x*, and contains 6 samples: 51, 52, 53, 54, 55, 56; each sample has 75 million reads. Both data sets are simulated by `rsem-simulate-reads`. The main difference between *sim5x* and *sim4x* is the read start position distribution. In *sim4x*, reads are uniformly generated along the transcripts, however in *sim5x* many biases are introduced to better reflect the situations in real RNA-sequencing data [14, 15].

The benchmark was run on a computer with 40-cores Intel(R) Xeon(R) E5-2680 v2 @ 2.80GHz CPU, and 96 GiB DDR3 1866 MHz RAM. Each tool was run with its default parameters in 32 threads. Besides, we also ran RSEM with `--estimate-rspd` option to test the *read start position* update of RSEM. Hera-EM and RSEM used the same alignment files performed by bowtie2 [5] with the following parameter which is from RSEM.

```
bowtie2 -p n_thread -x index --sensitive --dpad 0 --gbar 99999999 --mp 1,1 --np 1
--score-min L,0,-0.1 --no-mixed --no-discordant -k 200 -I 1 -X 1000 -1 read1
.fq -2 read2.fq | samtools view -bS - > alignment.bam
```

Table 1 describes the running time and peak memory for EM procedures of Hera-EM and RSEM. On average, Hera-EM is about a hundred times faster than RSEM's EM. Hera-EM consumes about 6 to 8 GBs memory while RSEM EM procedure consumes about 6 to 9 GBs. This table also presents the correlation between both tools' TPM estimation and the simulated truth TPM. The estimation and the simulated truth TPM values are both rounded to 2 decimal digits before computing the correlation. Spearman's rank correlation and pearson correlation are computed using their definitions [3]. To compute the log-pearson score, each TPM value is first added an offset of 0.01 then we compute the pearson correlation on the log-transformed values of the modified TPM values (These benchmark metrics are based on the SMC DREAM challenge [13]).

In *sim4x* data sets, the results from Hera-EM and RSEM are extremely similar. However,

Table 1: Running time, peak memory and correlations with simulated truth TPM of Hera-EM and RSEM

Dataset	Time (second)		Memory (MB)		Spearman		Pearson		Log-Pearson	
	Hera-EM	RSEM	Hera-EM	RSEM	Hera-EM	RSEM	Hera-EM	RSEM	Hera-EM	RSEM
sim41	32	3311	6404	6804	0.9408	0.9417	0.9988	0.9986	0.9763	0.9766
sim42	39	2971	6388	6752	0.9375	0.9377	0.9952	0.9941	0.9766	0.9767
sim43	47	4035	6454	6811	0.9416	0.9425	0.9959	0.9959	0.9773	0.9776
sim44	42	3567	6565	7080	0.9378	0.9379	0.9995	0.9995	0.9758	0.9758
sim45	24	3432	6384	6469	0.9379	0.9385	0.9989	0.9989	0.9751	0.9755
sim51	58	4290	7675	8495	0.9421	0.9291	0.9984	0.9970	0.9761	0.9593
sim52	74	4637	7747	8614	0.9448	0.9318	0.9994	0.9974	0.9774	0.9610
sim53	39	5044	7778	8970	0.9433	0.9322	0.9957	0.9956	0.9769	0.9611
sim54	51	4514	8206	9849	0.9426	0.9293	0.9988	0.9979	0.9755	0.9590
sim55	48	4157	7632	8496	0.9389	0.9266	0.9998	0.9986	0.9756	0.9599
sim56	45	4074	8083	9245	0.9373	0.9254	0.9997	0.9890	0.9740	0.9581

Table 2: Correlations of Kallisto, Hera-EM, and RSEM with rspd estimation option turned on with simulated truth TPM. Kallisto and Hera-EM were run with default parameters.

Dataset	Kallisto			Hera-EM			RSEM with rspd estimation		
	Spearman	Pearson	Log-Pearson	Spearman	Pearson	Log-Pearson	Spearman	Pearson	Log-Pearson
sim41	0.9057	0.9769	0.9624	0.9408	0.9988	0.9763	0.9332	0.9977	0.9679
sim42	0.9073	0.9887	0.9647	0.9375	0.9952	0.9766	0.9295	0.9939	0.9679
sim43	0.9072	0.9928	0.9647	0.9416	0.9959	0.9773	0.9316	0.9958	0.9665
sim44	0.9050	0.9805	0.9631	0.9378	0.9995	0.9758	0.9309	0.9993	0.9677
sim45	0.9043	0.9916	0.9621	0.9379	0.9989	0.9751	0.9299	0.9971	0.9670
sim51	0.8901	0.9881	0.9454	0.9421	0.9984	0.9761	0.9304	0.9973	0.9617
sim52	0.8922	0.9878	0.9477	0.9448	0.9994	0.9774	0.9310	0.9984	0.9620
sim53	0.8939	0.9762	0.9479	0.9433	0.9957	0.9769	0.9305	0.9958	0.9609
sim54	0.8908	0.9908	0.9460	0.9426	0.9988	0.9755	0.9277	0.9979	0.9586
sim55	0.8938	0.9948	0.9480	0.9389	0.9998	0.9756	0.9243	0.9991	0.9588
sim56	0.8938	0.9845	0.9458	0.9373	0.9997	0.9740	0.9248	0.9972	0.9567

in *sim5x* data sets, Hera-EM is more accurate than RSEM in all metrics (Spearman, Pearson, Log-pearson correlations with the ground simulated truth). A scatter plot between the estimation of both tools and simulated truth of the sim 51 is shown in Fig. 2. This difference is because the read start distribution in *sim5x* is not uniformed, while RSEM assumes it to be uniformed by default.

We further run RSEM with `--estimate-rspd` option to enable the estimation of read start position distribution. As the update formula for this parameter in RSEM is not correct, RSEM's results deteriorate (Table 2 and Fig.2).

We also benchmarked the accuracy of Kallisto (Table 2) on these data sets. Kallisto has the lowest scores in all metrics. Note that as Kallisto doesn't provide user the ability to input alignment files, the low accuracy can also be the result of spurious pseudo-alignment process.

We also noticed that given the same bowtie2 alignment, Salmon's quantification results exceed both RSEM and Hera-EM in the *sim4x*, and stay between Hera-EM (better) and RSEM in *sim5x*. We believe this may stem from the differences between variational bayesian and EM algorithms.

(a) Hera-EM transcripts abundance against truth from simulated data

(b) RSEM transcripts abundance against truth from simulated data

Figure 2: Scatter plots of estimated abundances and truth TPM in sim51 of both tools. Hera-EM shows a better correlation with the simulated truth TPM. All tools were run with default parameters.

4 Supplementary

As stated, our goal is to find the parameter Ω to maximize the likelihood $\mathcal{L}$ using EM algorithm

$$\begin{aligned} Q(\Omega \mid \Omega^{(i)}) &= \mathbb{E}_{\alpha_r \mid r, \Omega^{(i)}} [\log P(r, \alpha_r \mid \Omega)] \\ &= \sum_{r \in R} \mathbb{E}_{\alpha_r \mid r, \Omega^{(i)}} [\log P(r, \alpha_r \mid \Omega)] = \sum_{r \in R} \sum_{a \in A_r} \log(P(r, a \mid \Omega)) \times P(a \mid r, \Omega^{(i)}) \end{aligned}$$

We have to maximize this function with respect to Ω

4.1 Transcript distribution

Our goal is to find the maximum value of $Q(\Omega \mid \Omega^{(i)})$ respect to θ with the condition $\sum_i \theta_i = 1$. From the equation 1, we have,

$$P(r, a \mid \Omega) = P(t_a \mid \Omega) \times P(r, a \mid t_a, \Omega) = \theta_{t_a} \times P(r, a \mid t_a, \Omega)$$

Whereas the probability $P(r, a \mid t_a, \Omega)$ is independent from θ . Hence we have,

$$\begin{aligned} \frac{\partial Q(\Omega \mid \Omega^{(i)})}{\partial \theta_t} &= \sum_{r \in R} \sum_{a \in A_r} \frac{\partial \log(\theta_{t_a} \times P(r, a \mid t_a, \Omega))}{\partial \theta_t} \times P(a \mid r, \Omega^{(i)}) \\ &= \sum_{r \in R} \sum_{a \in A_r} \frac{\partial \log(\theta_{t_a})}{\partial \theta_t} \times P(a \mid r, \Omega^{(i)}) \\ &= \sum_{r \in R} \sum_{a \in A_r, t_a=t} \frac{1}{\theta_t} \times P(a \mid r, \Omega^{(i)}) \end{aligned}$$

Using Lagrange multiplier, we have the following system of equations,

$$\begin{aligned} \sum_{r \in R} \sum_{a \in A_r, t_a=t} \frac{1}{\theta_t} \times P(a \mid r, \Omega^{(i)}) &= \alpha \\ \sum_i \theta_i &= 1 \end{aligned}$$

With α is the Lagrange multiplier, hence,

$$\theta_t = \frac{1}{\alpha} \times \sum_{r \in R} \sum_{a \in A_r, t_a=t} P(a \mid r, \Omega^{(i)})$$

Solve for α

$$\sum_t \theta_t = \frac{1}{\alpha} \times \sum_{r \in R} \sum_{a \in A_r} P(a \mid r, \Omega^{(i)}) = \frac{1}{\alpha} \times \sum_{r \in R} 1 = \frac{|R|}{\alpha} = 1$$

$$\alpha = |R|$$

Therefore,

$$\theta_t = \frac{1}{|R|} \times \sum_{r \in R} \sum_{a \in A_r, t_a=t} P(a \mid r, \Omega^{(i)}) \quad (2)$$

4.2 Fragment length distribution

Similarly to transcript distribution, we have,

$$P(r, a \mid \Omega) = P(f_a \mid t_a, \Omega) \times C = \frac{\lambda_{f_a}}{\sum_{i \leq |t_a|} \lambda_i} \times C$$

With C is some function that is independent from λ Hence,

$$\begin{aligned} \frac{\partial Q(\Omega \mid \Omega^{(i)})}{\partial \lambda_f} &= \sum_{r \in R} \sum_{a \in A_r, f \leq |t_a|} \frac{\partial \log \left(\frac{\lambda_{f_a}}{\sum_{i \leq |t_a|} \lambda_i} \times C \right)}{\partial \lambda_f} \times P(a \mid r, \Omega^{(i)}) \\ &= \sum_{r \in R} \sum_{a \in A_r, f \leq |t_a|} \left(\frac{\partial \log(\lambda_{f_a})}{\partial \lambda_f} - \frac{\partial \log \left(\sum_{i \leq |t_a|} \lambda_i \right)}{\partial \lambda_f} \right) \times P(a \mid r, \Omega^{(i)}) \\ &= \sum_{r \in R} \sum_{a \in A_r, f_a=f} \frac{1}{\lambda_f} \times P(a \mid r, \Omega^{(i)}) \\ &\quad - \sum_{r \in R} \sum_{a \in A_r, f \leq |t_a|} \frac{1}{\sum_{i \leq |t_a|} \lambda_i} \times P(a \mid r, \Omega^{(i)}) \end{aligned}$$

Using Lagrange multiplier, we have the following system of equations,

$$\begin{aligned} \sum_{r \in R} \sum_{a \in A_r, f_a=f} \frac{1}{\lambda_f} \times P(a \mid r, \Omega^{(i)}) - \sum_{r \in R} \sum_{a \in A_r, f \leq |t_a|} \frac{1}{\sum_{i \leq |t_a|} \lambda_i} \times P(a \mid r, \Omega^{(i)}) &= \alpha \\ \sum_i \lambda_i &= 1 \end{aligned}$$

We calculate the difference between the equation for λ_f and λ_{f-1}

$$\begin{aligned} \sum_{r \in R} \sum_{a \in A_r, f_a=f} \frac{1}{\lambda_f} \times P(a \mid r, \Omega^{(i)}) - \sum_{r \in R} \sum_{a \in A_r, f_a=f-1} \frac{1}{\lambda_{f-1}} \times P(a \mid r, \Omega^{(i)}) \\ + \sum_{r \in R} \sum_{a \in A_r, f-1=|t_a|} \frac{1}{\sum_{i \leq f-1} \lambda_i} \times P(a \mid r, \Omega^{(i)}) = 0 \end{aligned}$$

Let $X_f = \sum_{r \in R} \sum_{a \in A_r, f_a=f} P(a \mid r, \Omega^{(i)})$

Let $Y_f = \sum_{r \in R} \sum_{a \in A_r, f=|t_a|} P(a \mid r, \Omega^{(i)})$

Let $\hat{\lambda}_f = \sum_{i=0}^f \lambda_i$

$$\begin{aligned} \frac{X_f}{\lambda_f} - \frac{X_{f-1}}{\lambda_{f-1}} + \frac{Y_{f-1}}{\hat{\lambda}_{f-1}} &= 0 \\ \frac{X_f}{\lambda_f} &= \frac{X_{f-1}}{\lambda_{f-1}} - \frac{Y_{f-1}}{\hat{\lambda}_{f-1}} \end{aligned}$$

Assume that $\frac{X_i}{\lambda_i} = Z_i \times c$ with $Z_0 = 1$,

$$Z_f \times c = Z_{f-1} \times c - \frac{Y_{f-1}}{\sum_{i \leq f-1} \frac{X_i}{Z_i \times c}}$$

Hence,

$$Z_f = Z_{f-1} - \frac{Y_{f-1}}{\sum_{i \leq f-1} \frac{X_i}{Z_i}}$$

After computing Z_f we can compute $\lambda_i \times c$, and with the constraint $\sum_i \lambda_i = 1$ to get the value of c

4.3 Read start position distribution

We have,

$$P(r, a \mid \Omega) = P(s_a \mid t_a, f_a, \Omega) \times C = \frac{g(s_a + 1) - g(s_a)}{g(|t_a| - f_a + 1)} \times C$$

With C is some function that independent from f , hence,

$$\begin{aligned} Q(\Omega \mid \Omega^{(i)}) &= \sum_{r \in R} \sum_{a \in A_r} \log \left(\frac{g(s_a + 1) - g(s_a)}{g(|t_a| - f_a + 1)} \times C \right) \times P(a \mid r, \Omega^{(i)}) \\ &= \sum_{r \in R} \sum_{a \in A_r} \log \left(\frac{g(s_a + 1) - g(s_a)}{g(|t_a| - f_a + 1)} \right) \times P(a \mid r, \Omega^{(i)}) + C_2 \end{aligned}$$

With C_2 is some function independent from Ω

For simplification we assume that $\left\lfloor \frac{s_a}{|t_a|} \times b \right\rfloor = \left\lfloor \frac{s_a + 1}{|t_a|} \times b \right\rfloor = q$ for all cases. Because transcripts are usually much longer than the number of segments, only a small gradient comes from other cases.

Therefore $g(s_a + 1) - g(s_a) = \left(\left\lfloor \frac{s_a + 1}{|t_a|} \times b \right\rfloor - \left\lfloor \frac{s_a}{|t_a|} \times b \right\rfloor \right) \times \pi_q = \left\lfloor \frac{b}{|t_a|} \right\rfloor \times \pi_q$

$$\begin{aligned} &\frac{\partial \log(g(s_a + 1) - g(s_a))}{\partial \pi_i} \\ &= \frac{1}{g(s_a + 1) - g(s_a)} \times \frac{\partial (g(s_a + 1) - g(s_a))}{\partial \pi_i} \\ &= \frac{1}{\left\lfloor \frac{b}{|t_a|} \right\rfloor \times \pi_q} \times \left\lfloor \frac{b}{|t_a|} \right\rfloor \times [q = i] \\ &= \frac{[q = i]}{\pi_i} \end{aligned}$$

Similarly, let $j = \left\lfloor \frac{|t_a| - f_a + 1}{|t_a|} \times b \right\rfloor$, and $\beta = \left\lfloor \frac{|t_a| - f_a + 1}{|t_a|} \times b \right\rfloor$

We have, $g(|t_a| - f_a + 1) = \hat{\pi}_k + j \times \pi_k$,

$$\begin{aligned} &\frac{\partial \log(g(|t_a| - f_a + 1))}{\partial \pi_i} \\ &= \frac{1}{g(|t_a| - f_a + 1)} \times \frac{\partial g(|t_a| - f_a + 1)}{\partial \pi_i} \end{aligned}$$

$$\begin{aligned}
 &= \frac{1}{\hat{\pi}_\beta + j * \pi_\beta} \times \frac{\partial (\hat{\pi}_\beta + j * \pi_\beta)}{\partial \pi_i} \\
 &= \frac{1}{\hat{\pi}_\beta + j * \pi_\beta} \times ([\beta > i] + j \times [\beta = i])
 \end{aligned}$$

System of equation, with α is the Lagrange multiplier.

$$\begin{aligned}
 \sum_{r \in R} \sum_{a \in A_r} \left(\frac{[\beta = i]}{\pi_i} - \frac{1}{\hat{\pi}_\beta + j * \pi_\beta} \times ([\beta > i] + j \times [\beta = i]) \right) \times P(a | r, \Omega^{(i)}) &= \alpha \\
 \hat{\pi}_b &= 1
 \end{aligned}$$

Let $X_k = \sum_{r \in R} \sum_{a \in A_r, k=i} P(a | r, \Omega^{(i)})$

Take the difference of the equations of π_{i+1} and π_i , we have,

$$\frac{1}{\pi_{i+1}} \times X_{i+1} - \frac{1}{\pi_i} \times X_i + \left(\sum_{r \in R} \sum_{a \in A_r} \left(\frac{1}{\hat{\pi}_\beta + j * \pi_\beta} \times ([\beta > i]) - \frac{1}{\hat{\pi}_\beta + j * \pi_\beta} \times ([\beta > i + 1]) \right) \times P(a | r, \Omega^{(i)}) \right)$$

Hence,

$$\frac{1}{\pi_{i+1}} \times X_{i+1} - \frac{1}{\pi_i} \times X_i + \left(\sum_{r \in R} \sum_{a \in A_r} \left(\frac{1}{\hat{\pi}_\beta + j * \pi_\beta} \times ([\beta = i + 1]) \right) \times P(a | r, \Omega^{(i)}) \right) = 0$$

We can't solve this equation explicitly so, we use an iterative method.

$$\begin{aligned}
 \text{Let } Y_k &= \sum_{r \in R} \sum_{a \in A_r, \beta=k} \frac{1}{\hat{\pi}_\beta + j * \pi_\beta} \times P(a | r, \Omega^{(i)}) \\
 \frac{X_{i+1}}{\pi_{i+1}} &= \frac{X_i}{\pi_i} - Y_{i+1}
 \end{aligned}$$

$$\text{Let } Z_k = \frac{X_k}{\pi_k}, \hat{Y}_k = \sum_{i=1}^k Y_i$$

$$\frac{X_i}{\pi_i} = Z_0 - \hat{Y}_i$$

We can't eliminate the variable π_i in $\hat{Y}_i$ so we use an approximation from the previous value of π_i . We can easily compute the estimation of Y_i alongside with the EM algorithm. However there is still an unsolved variable Z_0 , we can use Newton method to solve the equation $\hat{\pi}_b = 1$ to get the value of Z_0 . We have,

$$\pi_i = \frac{X_i}{Z_0 - \hat{Y}_i}$$

Hence,

$$\frac{d\pi_i}{dZ_0} = - \frac{X_i}{(Z_0 - \hat{Y}_i)^2}$$

Therefore, we have the update to get Z_0 ,

$$Z_0 \leftarrow Z_0 - \frac{\sum \pi_i}{\sum \frac{d\pi_i}{dZ_0}}$$

4.4 Orientation distribution

$$\begin{aligned}
 P(r, a | \Omega) &= P(o_a | \omega) \times C \\
 \frac{\partial Q(\Omega | \Omega^{(i)})}{\partial \omega_i} &= \sum_{r \in R} \sum_{a \in A_r, o_a=i} \frac{\partial \log(\omega_i \times C)}{\partial \omega_i} \times P(a | r, \Omega^{(i)}) \\
 &= \sum_{r \in R} \sum_{a \in A_r, o_a=i} \frac{1}{\omega_i} \times P(a | r, \Omega^{(i)})
 \end{aligned}$$

Solve the system of Lagrange equations,

$$\sum_{r \in R} \sum_{a \in A_r, o_a = i} \frac{1}{\omega_i} \times P(a | r, \Omega^{(i)}) = \alpha$$

$$\omega_0 + \omega_1 = 1$$

To get result,

$$\omega = \frac{1}{|R|} \times \sum_{r \in R} \sum_{a \in A_r, o_a = 1} P(a | r, \Omega^{(i)})$$

4.5 Alignment distribution

$$P(r, a | \Omega) = P(d | t_a, f_a, s_a, o_a) \times C = \epsilon^d \times (1 - \epsilon)^{|r_1| + |r_2| - d} \times C$$

Let $|r| = |r_1| + |r_2|$

$$\begin{aligned} \frac{\partial Q(\Omega | \Omega^{(i)})}{\partial \epsilon} &= \sum_{r \in R} \sum_{a \in A_r} \frac{\partial \log(\epsilon^d \times (1 - \epsilon)^{|R| - d} \times C)}{\partial \epsilon} \times P(a | r, \Omega^{(i)}) \\ &= \sum_{r \in R} \sum_{a \in A_r} \left(\frac{d}{\epsilon} - \frac{|r| - d}{1 - \epsilon} \right) \times P(a | r, \Omega^{(i)}) = 0 \end{aligned}$$

Solve the equation to get the result,

$$\epsilon = \frac{\sum_{r \in R} \sum_{a \in A_r} d \times P(a | r, \Omega^{(i)})}{\sum_{r \in R} |r|}$$

5 Future work

As many works have shown that variational inference is arguably a better solution for transcript abundance quantification compared to the maximum likelihood estimation. We are considering to implement a similar Hera-VB algorithm for the RSEM generative model.

6 Acknowledgements

This work was supported by BioTuring Inc. We would like to thank the organizer of SMC DREAM challenge for providing the simulated data sets. We also would like to thank the reviewers for reading this paper and giving us critical comments. We are in debt to Rob Patro for many insightful comments, and for very interesting conversations, so much so that Son Pham missed the flight while standing at the boarding gate.

References

- [1] Kendall E Atkinson. *An introduction to numerical analysis*. John Wiley & Sons, 2008.
- [2] Nicolas L Bray, Harold Pimentel, Páll Melsted, and Lior Pachter. Near-optimal probabilistic rna-seq quantification. *Nature biotechnology*, 34(5):525, 2016.
- [3] Gregory W Corder and Dale I Foreman. *Nonparametric statistics: A step-by-step approach*. John Wiley & Sons, 2014.
- [4] Jonathan Houseley and David Tollervey. The many pathways of rna degradation. *Cell*, 136(4):763–776, 2009.

- [5] Ben Langmead and Steven L Salzberg. Fast gapped-read alignment with bowtie 2. *Nature methods*, 9(4):357, 2012.
- [6] Bo Li and Colin N Dewey. Rsem: accurate transcript quantification from rna-seq data with or without a reference genome. *BMC bioinformatics*, 12(1):323, 2011.
- [7] Naoki Nariai, Kaname Kojima, Takahiro Mimori, Yukuto Sato, Yosuke Kawai, Yumi Yamaguchi-Kabata, and Masao Nagasaki. Tigar2: sensitive and accurate estimation of transcript isoform expression with longer rna-seq reads. *BMC genomics*, 15(10):S5, 2014.
- [8] Marius Nicolae, Serghei Mangul, Ion I Măndoiu, and Alex Zelikovsky. Estimation of alternative splicing isoform frequencies from rna-seq data. *Algorithms for molecular biology*, 6(1):9, 2011.
- [9] Lior Pachter. Near-optimal rna-seq quantification with kallisto. <https://liorpachter.wordpress.com/2015/05/10/near-optimal-rna-seq-quantification-with-kallisto/>. Accessed: 2018-11-04.
- [10] Rob Patro, Geet Duggal, Michael I Love, Rafael A Irizarry, and Carl Kingsford. Salmon provides fast and bias-aware quantification of transcript expression. *Nature methods*, 14(4):417, 2017.
- [11] Rob Patro, Stephen M Mount, and Carl Kingsford. Sailfish enables alignment-free isoform quantification from rna-seq reads using lightweight algorithms. *Nature biotechnology*, 32(5):462, 2014.
- [12] Irene Gallego Romero, Athma A Pai, Jenny Tung, and Yoav Gilad. Rna-seq: impact of rna degradation on transcript quantification. *BMC biology*, 12(1):42, 2014.
- [13] SMC-RNA. Github - sage-bionetworks/smc-rna-challenge. <https://github.com/Sage-Bionetworks/SMC-RNA-Challenge>. Accessed: 2018-11-04.
- [14] SMC-RNA. The icgc-tcga dream somatic mutation calling - rna challenge. <https://www.synapse.org/#!Synapse:syn2813589/wiki/401442>. Accessed: 2018-11-04.
- [15] SMC-RNA. The icgc-tcga dream somatic mutation calling - rna challenge. <https://github.com/Sage-Bionetworks/rnaseqSim>. Accessed: 2018-11-04.
- [16] Ravi Varadhan and Christophe Roland. Simple and globally convergent methods for accelerating the convergence of any em algorithm. *Scandinavian Journal of Statistics*, 35(2):335–353, 2008.
- [17] Mohsen Zakeri, Avi Srivastava, Fatemeh Almodaresi, and Rob Patro. Improved data-driven likelihood factorizations for transcript abundance estimation. *Bioinformatics*, 33(14):i142–i151, 2017.