Learning Probabilistic Splice Graphs from RNA-Seq data

Laura LeGault and Colin Dewey

December 20, 2010

Abstract

RNA-Seq technology provides the foundation for accurately measuring gene expression levels when paired with a model for mapping the produced sequencing reads to a reference genome. Because of the shorter length of RNA-Seq reads, a single read is not uniquely mapped to a single location in the genome and requires probabilistic treatment to accurately measure relative expression levels. We present a directed acyclic graph with edge weights learned using the EM algorithm from a collection of RNA-Seq where reads mapping to multiple locations are distributed according to their probability.

1 Introduction

Ever since Francis Crick proposed the central dogma of molecular biology in 1958 [2], it has been widely known and accepted that DNA and RNA are actually detailed repositories for information. Transcription transfers DNA information to RNA via RNA polymerase, and the product RNA interacts with the cell's ribosomes to translate the information into proteins. Between transcription and translation, alternative gene splicing may act to modify the pre-mRNA to alter the eventual protein produced. Once a genetic sequence of a number of introns and exons has been transcribed, the introns as well as various exons or portions of exons are removed from the mRNA based on the demands of the particular cell at that time. This allows for a single genetic sequence to code for a number of different proteins simultaneously [3].

There are several possible alternative splicing options. Two exons may always appear, with an optional exon between them, called a "cassette exon". A portion of the intronic sequence may be retained, or a portion at the beginning or end of an exon may be spliced out. There may be several alternate promoter regions or polyadenylation sites. Exons may also be mutually exclusive – if one appears in the mRNA, the other is removed. Diagramming the various splice activities of a genetic sequence can be done using what is called the *alternative splice graph*, or just splice graph [3].

Until recently, it has been prohibitively expensive to sequence RNA in a cell using the standard microarray methods or tag-based sequencing approaches. Recently, with the advent of the RNA-Seq technology [7], a population of RNA is converted to cDNA fragments which are then subjected to high-throughput sequencing to obtain short reads from each fragment (typically 30-400 base pairs). These reads can then be mapped to an existing reference genome, and are useful for determining expression levels – say, of different alternative splicings of a single gene sequence.

A particular advantage of the RNA-Seq technology is that as opposed to standard microarray methods, which give rough analog estimates of gene expression levels, the data resulting from an RNA-Seq procedure is a digital count of reads [6], much more precise than microarrays. These digital counts lend themselves much more readily to machine learning methods, as we simply take the reads and their counts directly and perform alignment against our reference genome.

After alignment of these reads, we note there are three types of reads: reads which map uniquely to a single location on the genetic sequence, reads which map to multiple locations, and reads which do not map to any location on the sequence [1]. Uniquely mapping reads are very easy to deal with, and non-mapping reads can simply be removed from analysis, but multiply-mapping reads are generally difficult to analyze, particularly in deterministic models. Some models (e.g. [4]) simply discard these reads as well as the non-mapping reads, but this clearly increases experimental bias.

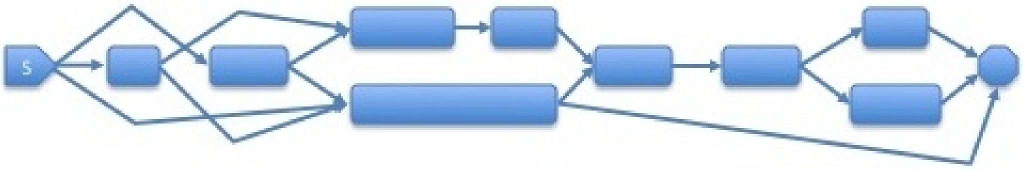


Figure 1: The probabilistic splice graph for the known isoforms of the mouse gene *Gfra4* as listed in the UCSC Genome Browser [5] (`http://genome.ucsc.edu`, mm9, UCSC Genes track). This gene's possible transcripts include alternate promoter regions, a cassette exon (represented as a choice between two shorter, gapped segments or one long segment), and optional beginning and ending sequences in the final segment.

1.1 Problem statement

Our aim is to create a representative *transcriptome*: a complete collection of the possible alternative transcripts or *isoforms* of a given gene and their relative frequencies in a cell at a given time. The work in this paper is concentrated on discovering the frequencies of various isoforms; our future work will explore the possible isoforms and constructing an accurate representation of the alternative splicing events of a particular gene given read data (see section 4.1 for a more detailed discussion).

We use two measures of relative frequency in our work: the relative frequency of transcripts t of a given isoform n, denoted $P(t_n)$, and the relative frequency z_{ij} of moving from one possible set of nucleotides (which we refer to as "segments", denoted s_i) to any of the possible subsequent segments s_j. The primary assumption from which we derive our relative frequencies is that the number of reads produced by RNA-Seq and corresponding to isoform n is directly proportional to the relative frequency of that isoform in the RNA population. We also assume that all potential isoforms are represented in our reference transcript.

2 Methods

2.1 Model description

The model which we use to represent the relative frequencies of transcript isoforms for a given gene is called a *probabilistic splice graph*, or PSG. A PSG is a directed acyclic graph with M nodes, each of which represents a segment of nucleotides of some given length. Each of the nodes is reachable from the source node s_1 via one or more paths along the edges of the graph, and there is also at least one path from each node to the sink node s_M. Each edge has exactly one source node and exactly one target node, and has some weight $0 \leq \alpha_{ij} \leq 1$.

A subpath s in the PSG is a sequence of vertices $s = \{s_1, \ldots, s_m\}$; a transcript t or *path* is therefore a subpath with $s_1 = 1$ and $s_m = M$. The probability of a subpath s is the product of the weights of its edges:

$$w(s) = \prod_{i=1}^{m-1} \alpha_{s_i s_{i+1}}$$

The probability that a transcript containing segment s_i also contains a subpath from s_i to s_j can be computed as the sum of the weights of all paths between the two segments, which can be computed recursively:

$$f(i,j) = \sum_{s:s_1=i, s_m=j} w(s) = \begin{cases} 1 & i = j \\ \sum_k \alpha_{kj} f(i,k) & i \neq j \end{cases} \tag{1}$$

The length of a subpath s is simply the sum of the lengths of its segments:

$$l(s) = \sum_{i=1}^{m} \ell_{s_i}$$

Any given read produced by RNA-Seq is dependent upon both the frequency of the transcript from which it derives and the length of that transcript: a longer transcript may produce more reads, but each given read has a lower probability because of the large number of nucleotides from which a read might begin in that transcript. Because of this dependence on length, it is also useful to have definitions for the expected prefix $(s : s_1 = 0, s_m = i)$, and suffix $(s : s_1 = i, s_m = M)$ lengths for segment s_i, denoted $d_p(i)$ and $d_s(i)$ respectively, which we can compute recursively.

$$\begin{aligned} d_p(i) &= \ell_i + \frac{1}{f(0,i)} \sum_j f(j)\alpha_{ji} d_p(j) && (2) \\ d_q(i) &= \ell_i + \sum_j \alpha_{ij} d_s j && (3) \end{aligned}$$

Using these equations, we can compute the expected length of a transcript generated from our PSG by $d_q(1) = d_p(M)$.

2.2 Learning with EM

The goal of our maximization is quite simple: create a model that fits our data as well as possible. Given a model structure and a set of N reads generated by RNA-Seq, we wish to learn the parameters of our PSG. In this situation, we only wish to learn the appropriate weights for the edges, maximizing the likelihood of the reads R given the model under current parameters α:

$$\begin{aligned} P(R|\alpha) &= \prod_{n=1}^{N} \sum_t \sum_{s,b} P(r_n|s,b)P(s,b|t)P(t) \\ &= \prod_{n=1}^{N} \sum_{t,s,b} \frac{w(t)}{\sum_i \ell_i f(0,i)} \\ &= \left(\sum_i \ell_i f(0,i) \right)^{-N} \prod_{n=1}^{N} \sum_{s,t,b} w(t) \end{aligned}$$

where $f(0,i)$ is as defined in (1), s are the possible subpaths, t are the possible transcripts, and b are the possible locations where a read may start within a segment. Since $f(i,j)$ is a function of the parameters α, this function is quite difficult to maximize directly. We choose to employ the Expectation Maximization (EM) algorithm to maximize the likelihood of the data given the model incrementally: the model is randomly initialized, the data is mapped to the model given the probabilities, and the parameters are updated to more accurately reflect the data mapping. The data is then re-mapped given these new parameters, and the steps are repeated until some convergence criteria are met.

Pre-processing

As an implementation detail, there are three steps that we take before we begin our algorithm. First we perform string alignment on the RNA-Seq reads – if the reads are guaranteed exact (as we assume for this implementation) we merely use substring matching; if not, a more complex alignment method may be employed allowing for uncertainty in the reads. As an important detail, we allow reads to match multiple segments as well as multiple times within a segment. Our other two steps initialize the model for algorithmic processing: we reset all outgoing edges from a segment to be equal and sum to 1; we also pre-calculate the current values of the $f(i,j)$ function, though this must be repeated for every update to the model parameters.

E-step

The expectation step in our algorithm comprises the computation of the expected values of the Z_{ij} variables for each edge e_{ij}, which can be interpreted as using the current parameters to calculate how much of read n's transcript might have been explained by a given edge. $Z_{ij} = 1$ if edge e_{ij} is used in a read's transcript; $Z_{ij} = 0$ otherwise. Because we allow a read to map to multiple locations – potentially mutually exclusive locations – our $E(Z_{ij})$ values for each read will be some probability $0 \le E(Z_{ij}) \le 1$.

$$\begin{aligned} E(Z_{n,ij}) &= P(Z_{n,ij} = 1|r_n) \\ &= \frac{P(r_n, Z_{n,ij} = 1)}{P(r_n)} \\ &= \frac{\sum_{s,b} \sum_t P(r_n, Z_{n,ij} = 1, t, s, b)}{\sum_{s,b} \sum_t P(r_n, t, s, b)} \\ &= \frac{\sum_{s,b} \sum_{t:(i,j)\in t, s\in t} (\sum_i \ell_i f(0,i))^{-1} w(t)}{\sum_{s,b} \sum_{t:s\in t} (\sum_i \ell_i f(0,i))^{-1} w(t)} \\ &= \frac{\sum_{s,b} \sum_{t:(i,j)\in t, s\in t} w(t)}{\sum_{s,b} \sum_{t:s\in t} w(t)} \\ &= \frac{\sum_{s,b} g(s,i,j)}{\sum_{s,b} g(s)} \end{aligned}$$

where we define the g functions as

$$\begin{aligned} g(s) &= f(0,s_1)w(s) \\ g(s,i,j) &= \begin{cases} f(s_1)w(s) & e_{ij} \in s \\ w(s)\alpha_{ij} f(0,i) f(j,s_1) & else \end{cases} \end{aligned}$$

if we assume that e_{ij} precedes s in the PSG; otherwise

$$g(s,i,j) = \begin{cases} f(s_1)w(s) & e_{ij} \in s \\ w(s)\alpha_{ij} f(0,s_1) f(s_m,i) & else \end{cases}$$

Once the $E(Z_{ij}) = \sum_n E(Z_{n,ij})$ has been calculated for all reads n, we determine the current maximum *a posteriori* likelihood of the data given the model, given by the product over all reads of the probability of taking the transcript path suggested by that read, divided by all possible transcript probabilities times their lengths:

$$P(r,z|\alpha) = \frac{\prod_{n=1}^{N} w(t_n)}{\left(\sum_t w(t)\ell_t\right)^{-N}}$$

If we calculate the log value of the complete data likelihood, however, we reduce this to

$$\begin{aligned} E(\log P(r,z|\alpha) &= \log \frac{\prod_{n=1}^{N} w(t_n)}{(\sum_t w(t)\ell_t)^{-N}} \\ &= \frac{1}{-N} \sum_{n=1}^{N} \frac{w(t_n)}{\sum_t w(t)\ell_t} \end{aligned}$$

If the edge weights obey the constraint $\sum_j \alpha_{ij} = 1$, then we observe that $\sum_t w(t)\ell_t$ is the mean length of the expressed transcripts, which we write more concisely as $d_s(0)$, as defined in (3). Because we observe only the segments to which our reads r_n map rather than their entire transcripts t_n, we then replace $w(t_n)$ with the sum of the weights over the possible paths to the mapped segments, denoted $f(0,s)w(s)$. We also discard the constant $1/N$ since it is not a function of α and provides us no additional information. This results in a log likelihood equation of

$$E(\log P(r,z|\alpha) = \sum_{n=1}^{N} \log \frac{\sum_{s_n} g(s_n)}{d_q(0)}.$$

Once this value is maximized (that is, when the magnitude of its increase over the previous probability is below a certain threshold), we consider the algorithm complete.

M-step

Given the Z_{ij} values as calculated in the expectation step, we now adjust our model parameters to their maximum likelihood estimation. Our intuition suggests that the ML estimate for α_{ij} is directly proportional to the number of times e_{ij} is used and inversely proportional to the average length of a transcript including e_{ij}. To show this, we consider again the complete data likelihood

$$P(r,z|\alpha) = d_q(0)^{-N} \prod_{n=1}^{N} \prod_{i,j} \alpha_{ij}^{z_{n,ij}}$$

We take the log of this function to produce our Q function for maximization over α using the current values $\alpha^{(t)}$ and the z_{ij} values from the expectation step:

$$\begin{aligned} Q(\alpha|\alpha^{(t)}) &= E(\log P(r,z|\alpha)) \\ &= -N \log d_q(0) + \sum_{i,j} z_{ij} \log \alpha_{ij} \end{aligned}$$

Since we must constrain $\sum_j \alpha_{ij} = 1$ for all i, for maximization purposes we introduce the Lagrangian

$$\begin{aligned} \Lambda(\alpha,\lambda) &= -N \log d_q(0) + \sum_{i,j} z_{ij} \log \alpha_{ij} \\ &\quad + \sum_i \lambda_i \left(\sum_j \alpha_{ij} - 1 \right) \end{aligned}$$

To maximize, we take the derivative with respect to α_{ij} and set the result to zero:

$$0 = \frac{\partial \Lambda}{\partial \alpha_{ij}} = -\frac{N}{d_q(0)} \frac{\partial d_q}{\partial \alpha_{ij}} + \frac{z_{ij}}{\alpha_{ij}} + \lambda_i$$

The derivative of the $d_q(0)$ factor can be written as

$$\begin{aligned} \frac{\partial d_q}{\partial \alpha_{ij}} &= \frac{\partial}{\partial \alpha_{ij}} \left(\ell_i + \sum_j \alpha_{ij} d_q(j) \right) \\ &= f(0,i)(d_p(i) + d_q(j)) \end{aligned}$$

And so to maximize our $Q(\alpha|\alpha^{(t)}$ function, we must find a solution to

$$0 = -\frac{N}{d_q(0)} f(0,i)(d_p(i) + d_q(j)) + \frac{z_{ij}}{\alpha_{ij}} + \lambda_i$$

Since $d_p(i)$ and $d_q(j)$ are not functions of $\alpha_{ik} \forall k$, we can treat all parameters $\alpha_{i'j}, i' \neq i$ as fixed and observe that

$$\begin{aligned} \lambda_i &= 0 \\ \alpha_{ij} &= \frac{\frac{z_{ij}}{(d_p(i)+d_q(j))}}{\sum_k \frac{z_{ik}}{(d_p(i)+d_q(k))}} \end{aligned}$$

The denominator is a sum over all outgoing edges from segment i, so we ensure that $\sum_j \alpha_{ij} = 1$.

We also consider at this point a termination condition wherein if the largest magnitude of the difference between the previous edge weights $\alpha^{(t)}$ and the new weights α is below a certain threshold, we consider the algorithm complete.

3 Results

3.1 Simulating RNA-Seq data

At this point in the investigation we are simply developing the methods for analysis rather than performing actual analysis on RNA-Seq data, so it is necessary to simulate our own given PSGs with preset weights. It is extremely important to generate these data accurately, since the EM model is extremely sensitive to systematic probability errors and will fail to learn correct probabilities if the reads generated do not adhere to our previously stated assumptions of proportionality to frequency of expression and transcript length.

In our generated data, we allow reads to span multiple segments[1] and overflow the 3' end of the gene, resulting in a poly-A tail, so as to more closely mimic potential RNA-Seq data. We do not introduce read errors or SNPs, so as to simplify the alignment step.

Modeling the reads requires four random variables:

- R_n: the sequence of read n.
- T_n: the full transcript path from which read n arises.
- S_n: the subpath in the PSG from which read n arises. $S_n = \{S_{n,1}, \ldots, S_{n,m}\}$ where m is the number of vertices in the subpath, and $S_n \subset T_n$.
- B_n: the position in $S_{n,1}$ at which read n begins.

For N reads, our data likelihood is a joint probability over reads, transcripts, subpaths and starting indices:

$$P(r,t,s,b) = \prod_{n=1}^{N} P(r_n|s_n,b_n)P(s_n,b_n|t_n)P(t_n)$$

Since we require at this time that reads are exactly generated from the model, we have

$$P(r_n|s_n,b_n) = \begin{cases} 1 & r_n \in s_n \text{ beginning at } b_n \\ 0 & else \end{cases}$$

We assume that the position in which a read begins in a given transcript t_n is simply uniformly distributed across the length of the transcript[2], and so we simply have

$$P(s_n,b_n|t_n) = \begin{cases} \frac{1}{l_{t_n}} & s_n \in t_n, b_n \in [1, \ell_{s_{n,1}}] \\ 0 & else \end{cases}$$

Finally we assume the probability of generating a read from a transcript is proportional to the frequency of t_n and its length:

$$P(t_n) = \frac{w(t_n)\ell_{t_n}}{\sum_t w(t)\ell_t}$$

Combining all of these probabilities and assuming that the reads r, transcripts t, subpaths s and start positions b are compatible, we observe that the final likelihood simplifies to

$$P(r,t,s,b) = \frac{\prod_n w(t_n)}{\sum_t w(t)\ell_t}$$

To generate our data, we first calculate the probability that read n begins in segment i:

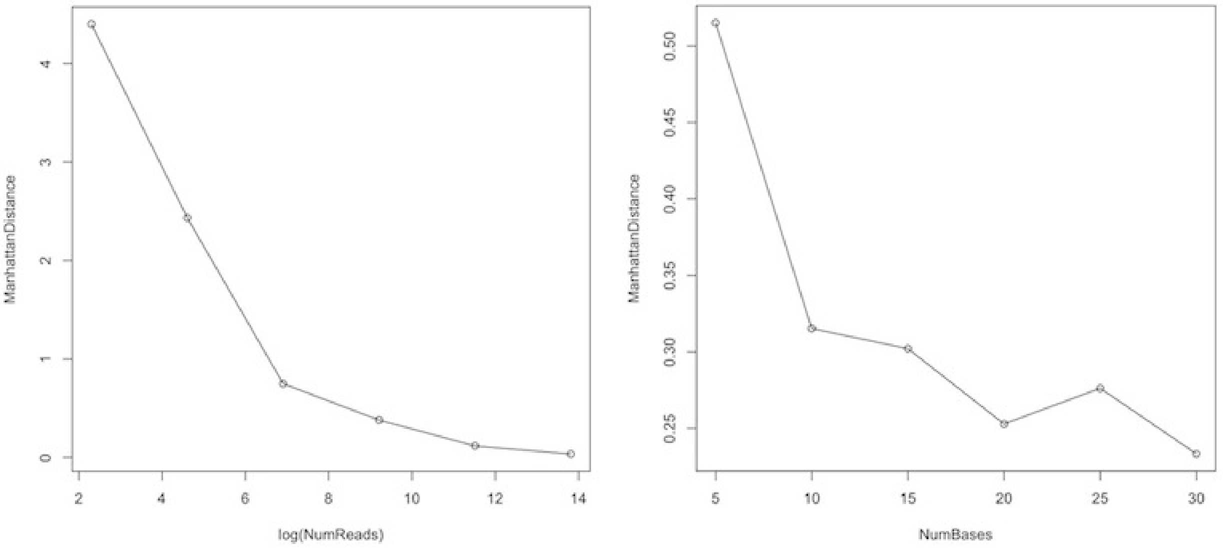
$$\begin{aligned} P(S_{n,1} = i) &= \sum_t \sum_{j<\ell_i} P(S_{n,1} = i, B_n = j, T_n = t) \\ &= \sum_t \sum_{j<\ell_i} P(s_i, b_j|t)P(t) \\ &= \sum_{t:i\in t} \ell_i \frac{1}{\ell_t} \frac{w(t)\ell_t}{\sum_i \ell_i f(0,i)} \\ &= \ell_i \frac{\sum_{t:i\in t} w(t)}{\sum_i \ell_i f(0,i)} = \frac{\ell_i f(0,i)}{\sum_i \ell_i f(0,i)} \end{aligned}$$

We sample our $S_{n,1}$ segment according to these probabilities and select a starting position b uniformly within $S_{n,1}$. If the specified read length ℓ_r does not fit fully within the segment, we select a subsequent segment with probability α_{ij}, repeated as necessary. If the subsequent vertex is the sink node, we add a poly-A tail of the required length to the end of the read.

Again we caution against carelessness with the implementation of this sampling algorithm – any systematic error in probability violating the assumptions of relationship to length and frequency of the transcript will result in the EM algorithm converging to incorrect edge weights.

[1] We note that our currently-implemented alignment strategy only robustly allows for a two-segment span.

[2] This is a simplifying assumption, as the cDNA fragmentation producing RNA-Seq data is strongly biased toward the 3' ends of transcripts [7].



(a) Number of reads

(b) Length of reads

Figure 2: Using the *Gfra4* PSG model with 16 edges (11 of which have weight $\neq 1$), the average Manhattan distance of the learned edge weights from the true edge weights calculated over five runs. (a) plots the distance over the log of increasing numbers of 10-base reads, from 10 reads to 1 million reads. (b) plots the distance over the length of a read from 5 bases to 30 bases as calculated for 10,000 reads.

Simulation procedure

For the purposes of our trials, we employ the above mathematical procedure to the mouse gene *Gfra4* as listed in the UCSC Genome Browser [5] (`http://genome.ucsc.edu`, mm9, UCSC Genes track). As presented in the UCSC Genome Browser, there are six known isoforms of this gene (though the model we employ in our analysis, pictured in figure 1, can produce 13 distinct isoforms). However the Genome Browser does not provide information on relative expression levels, so for our trials we assume that each of the six isoforms is equally likely. We then collapse these relative frequencies onto the appropriate edges in our model, and use these as our initial α values for read generation as detailed above. These are therefore also our target values to learn in the EM procedure, and the distances mentioned below are measured from these weights.

3.2 Sufficient numbers of reads

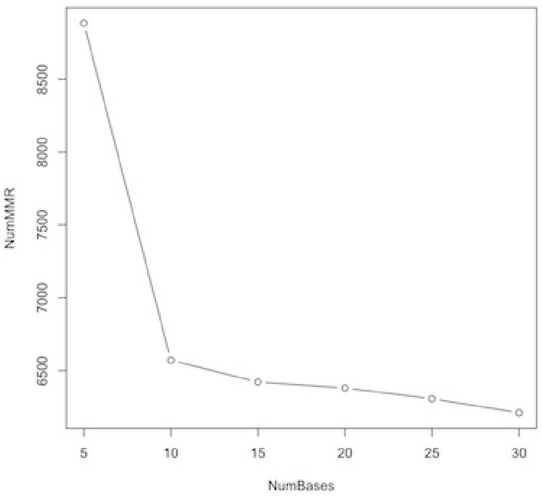
One of the critical pieces of data we wish to determine in this stage of development is how many RNA-Seq reads are required to sufficiently approximate the correct weights of a PSG. We chose to run the EM algorithm on our *Gfra4* model five times for each quantity of 10-base reads, ranging from 10 reads to 1 million reads, and determined the average Manhattan distance ($d_M(\hat{\alpha}, \alpha) = \sum_{ij} |\hat{\alpha_{ij}} - \alpha_{ij}|$) over all five runs of the learned edge weights from the true edge weights used to generate the data.

We have confirmed that with an increasing number of reads as shown in Figure 2a the average total distance between the true weights and learned weights in the PSG over five runs of the EM algorithm is a decreasing function of the number of reads with a linear regression equation of $\log(y) = -.421 \log(x) + 2.67$, where y is the expected Manhattan distance and x is the number of reads in our dataset. The slope is significantly different from zero with $p < .001$, so it is very safe to say that increases in numbers of reads is logarithmically effective at decreasing the distance between learned weights and true weights.

The *Gfra4* model has 16 edges, 11 of which have weights different from 1.0 and must be learned with the EM algorithm. We find that to achieve an average learned weight error (defined as total error divided by number of edges with learnable weights) of less than .01, approximately 100,000 reads of length 10 are required. However, we also observe that average error is only .07 with 1,000 reads and .03 with 10,000 reads, providing a relatively accurate approximation of the true frequencies even when data is somewhat limited.

3.3 Optimal read length

Based on the balance of error versus time efficiency, we chose to use datasets of 10,000 reads of various lengths from 5 to 30 bases, increasing in increments of five bases. This represents the lower bound on an RNA-Seq read – recall that RNA-Seq reads are generally between 30 and 400 nucleotides long [7]. As read length increases, the number of multiply-mapped reads (MMR) decreases as shown in the figure below, since it becomes easier with a longer read sequence to uniquely determine the position in the genetic sequence which gave rise to a given read.



Though the average Manhattan distance between the learned reads and the true reads as shown in Figure 2b is not strictly decreasing, we find that the points are fit with a linear regression equation of $\log(y) = -0.026x - .735$, where y is the expected Manhattan distance and x is the number of nucleotides in a read. The slope is significantly different from zero in a two-tailed t-test with $p < .05$, so we say that the overall trend of increasing read length is a decreasing error in learned weight.

As our tests were performed only on a single model at this time, we hesitate to make recommendations as to the optimal balance of read length and the expense of acquiring longer reads. This optimal length is determined by both the length of the gene in question and the number and length of subsequences repeated within that gene.

4 Discussion

In this paper, we have discussed a probabilistic model for representing relative frequencies of gene isoform expression that incorporates methods of dealing with read mapping uncertainty with the goal of producing a more accurate representation of true relative expression levels. We used a simulation of RNA-Seq to produce reads from a given PSG, and then used these reads to train a neutrally-weighted PSG to learn the true weights from the original graph. We explored the differences created by various read lengths and dataset sizes, confirming that increases in read length and dataset size result in a decrease in the error of the learned weights.

Our focus for this project was on learning the relative probabilities of a single gene's expression, though in practice it would be quite unusual for only a single gene's isoforms to be present within the RNA library used for RNA-Seq. It does not affect the mathematics of the EM algorithm to generalize the current model to multiple genetic isoforms; the alignment step will merely need to be modified, dividing the read dataset into multiple subsets which align to each given gene, and an overall expression probability assigned to each model in the set of PSGs of expressed genes.

Of particular interest in the implementation stage of this project was the sensitivity of the model to different parameters (as well as implementation errors) in the generation stage. A single change in the generation step is quite powerful to influence the learned parameters, which has very optimistic implications for the sensitivity of the model to true variations in expression levels in a given RNA population. Previous work (e.g. [7]) has noted RNA-Seq's large dynamic range of expression level detection, and it is encouraging to note that our model is also sensitive to these changes.

Although the present work was done exclusively using simulated RNA-Seq data, we believe that the simulations are consistent with the true characteristics of RNA-Seq on an RNA population. Therefore we are confident that the results of our simulated runs in this situation are generalizable to true RNA-Seq data, and we hope to have the opportunity to make that generalization soon.

4.1 Future work

The next step in our research is to use RNA-Seq data to construct the PSG structure, given the exonic segments in a gene. While it is possible to model all six possible isoforms of the *Gfra4* gene using our model in Figure 1, the model also suggests an additional seven isoforms which are not known to exist, and it is our goal to create a model that contains only relative frequencies for the possible isoforms.

We are beginning with simulated reads as in the present EM weight-learning case, and combining these with a fully-connected linear DAG with only one copy of each possible segment. Using a greedy search, we then modify the graph by duplicating segments with multiple incoming edges so that each copy of the segment has only one incoming edge and the same number of outgoing edges as the original copy. We then run the EM algorithm on this graph to determine its log likelihood, and combine with a Bayesian Information Criterion (BIC) to enforce the minimum description length principle, and accept the modification with the highest BIC value and repeat our search until no improvements are made. We hope to complete the theoretical portion of this project shortly and refine our implementation.

References

[1] Valerio Costa, Claudia Angelini, Italia De Feis, and Alfredo Ciccodicola. Uncovering the complexity of transcriptomes with RNA-Seq. *Journal of Biomedicine and Biotechnology*, 2010, 2010.

[2] F.H.C. Crick. On protein synthesis. *Symp. Soc. Exp. Biol. XII*, pages 139–163, 1958.

[3] Paul Jenkins, Rune Lyngsø, and Jotun Hein. How many transcripts does it take to reconstruct the splice graph? *Lecture Notes in Computer Science*, 4175/2006:103–114, 2006.

[4] H. Jiang and W.H. Wong. Statistical inferences for isoform expression in RNA-Seq. *Bioinformatics*, 25:1026–1032, 2009.

[5] W.J Kent, C.W. Sugnet, T.S. Furey, K.M. Roskin, T.H. Pringle, A.M. Zahler, and D. Haussler. The human genome browser at UCSC. *Genome Res.*, 12(6):996–1006, 2002.

[6] Bo Li, Victor Ruotti, Ron M. Stewart, James A. Thomson, and Colin N. Dewey. RNA-Seq gene expression estimation with read mapping uncertainty. *Bioinformatics*, 26(4):493–500, 2010.

[7] Zhong Wang, Mark Gerstein, and Michael Snyder. RNA-Seq: a revolutionary tool for transcriptomics. *Genetics*, 10:57–63, January 2009.