The Cufflinks transcript assembly algorithm

Cole Trapnell, Ali Mortazavi, and Lior Pachter

April 22, 2009

1 Introduction

The Cufflinks algorithm takes as input a reference genome and a set of single or paired end alignments of RNA-Seq reads and reports as output a set of messenger RNA (transcript) sequences along with estimates of their relative abundances in the input sample. The algorithm is a direct extension of an algorithm that performs haplotype frequency estimation in viral populations [1]. Much of what appears here is copied verbatim from that paper to make the presentation of the extensions more clear. Both algorithms report a parsimonious set of assembled sequences that are consistent with the alignments, and assign reads to the assembled sequences using a maximum likelihood step. When the algorithm terminates, a maximal number of reads are assigned to assembled sequences, and every assigned read is consistent with the sequence to which it is assigned.

The Cufflinks algorithm extends the ShoRHA algorithm in [1] in several ways. First, ShoRAH assumes that the reads are unpaired, and align to the reference without gaps. Cufflinks handles reads that aligned over splice junctions, and reads that are paired. Second, ShoRHA assumes that the haplotypes are the same length and are syntenic. Cufflinks assembles a diverse set of sequences, corresponding to transcripts that are generally not the same length. Third, Cufflinks can integrate annotations seamlessly to produce its assemblies, increasing its effectiveness in regions of relatively low read coverage.

2 Algorithm

We describe the algorithm to comparatively assemble paired-end read alignments into transcripts in this section. Discussion of how to assemble single-end alignments, or mixtures of single- and paired-ends is omitted, but is believed to be straightforward with minor extensions to this algorithm. The first step in the algorithm computes a minimal set of transcript sequences needed to explain the reads. The second step estimates the relative abundances of those transcripts in the input via a maximum-likelihood calculation.

Let M be the paired-end reads alignments or *mate alignments* provided as input to the algorithm. Each mate alignment consists of two *read alignments*, which each consist of an offset into the reference and a sequence of CIGAR operations. We restrict ourselves to the 'match' and 'skip' operations. Match operations simply denote a contiguous interval in the alignment (i.e. a region of alignments that contains exact base matches and substitutions, but no gaps). Skip operations denote a (often intron-sized) gap in the alignment. For a mate alignment m, we denote the read alignment with the lower offset into the reference as the "left" read alignment, m_L, and the larger offset alignment the "right" read alignment, m_R. Also, we write the lowest-offset base in the reference covered by m as m_l, and the highest-offset base as m_r.

Definition 1. Two mate alignments x and y are said to overlap if the intervals $[x_l, x_r]$ and $[y_l, y_r]$ intersect.

An overlapping pair of mate alignments is *consistent* if they do not "disagree" about the locations and lengths of their implied introns (if any) and could have come from the same transcript. A pair of overlapping *read* alignments "intron-agree" on introns only if the computed offsets of the CIGAR skip operations are identical for both alignments. A pair of read alignments that do *not* overlap are said to vacuously agree.

The RNA-Seq protocol usually specifies randomly fragmenting cDNA and then size-selecting fragments. Each fragment or *insert* is sequenced for a fixed number of cycles from both ends, resulting in a pair of reads separated in the transcriptome coordinate space by a length following a generally well behaved (and presumed to be normal) distribution. Consider a pair of mates x and y that overlap such that part of a read alignment from y falls within the genomic

interval in between the read alignments of x. Call this part of y's read alignment $y*$, and denote its length l_{y*}. If x and y are from the same transcript, then the (unknown) part of the transcript between x's read alignments must contain $y*$ as a substring. Thus the distance between x's read alignments in the transcriptome coordinate space must be at least l_{y*}. If the cumulative distribution function of the insert length distribution is $F(d)$, then the probability that the transcriptomic inner distance x is at least l_{y*} is $1 - F(l_{y*})$. For some probability threshold t, we say x "distance-agrees" with y if:

$$1 - F(l_{y*}) \geq t$$

Note that the binary relation of distance-agreement is not generally symmetric

Definition 2. A pair of overlapping mate alignments x and y are consistent only if the following are both true:

1. x_L intron-agrees with y_L and x_R intron-agrees with y_R.
2. x distance-agrees with y and y distance-agrees with x with some probability threshold t.

A given transcript is called *completely consistent* with the input set of mate alignments M if its sequence can be constructed from a subset of M, where any mates in the subset that overlap are consistent. Let C_M be the set of all transcripts constructible from subsequences (by concatenation) of the reference and that are completely consistent with M. What follows are methods for constructing and sampling C_M necessary for computing a minimal set of transcripts necessary to explain the mates in M.

Definition 3. The *mate graph* G_M associated with M is a directed, acyclic graph with vertices $\{M_{irr}, s, t\}$ consisting of a source s, a sink t, and a vertex for each irredundant mate alignment. A mate alignment is x redudant if it is overlapped by another mate alignment y, x_L intron-agrees with and is contained by y_L and x_R intron-agrees with and is contained by y_R. The edges of G_M are defined by including an edge from x to y when

1. $x_l < y_l$
2. x and y overlap consistently
3. there would not be a path from x to y in G_M without this edge.

Finally, edges are added from the source vertex s to any vertex in M_{irr} that lacks a "left overlap", i.e. a vertex for mate x where there is no overlapping, consistent mate y with $y_l < x_l$. Edges are similarly added from mates without right overlaps to the sink vertex t.

A path through the mate graph corresponds to a transcript that is completely consistent with M and whose sequence can be constructed from overlaps implied by the edges on the path. We say that a set of transcript sequences T is an explaining set for M if every mate alignment $m \in M$, can be obtained as a pair of substrings from a transcript in T separated by a distance d that is less than two standard deviations from the mean of the inner distance length distribution.

We want to compute a minimal explaining set of transcripts that is completely consistent with our mate alignments. The proposition on page four of [1] implies that an explaining set of completely consistent transcripts is precisely a set of paths from the source to the sink, such that each vertex of the mate graph is covered by at least one path. This amounts to "explaining" each read by including it in a larger assembly. Such a set of paths is called a *cover*, and can be computed efficiently by the following theorem:

Dilworth's Theorem. *Given a mate graph:*

1. *Every minimal cover of the mate graph has the same cardinality, namely the size of the largest set Q of vertices such that there are no paths between elements of Q.*
2. *A minimal cover of the mate graph can be computed by solving a maximum matching problem in an associated bipartite graph. This maching problem can be solved in time at worst cubic in the number of irredundant reads.*

A minimal cover obtained from the maximum matching algorithm is in general not unique. It provides a minimal *chain decomposition* of the graph. A *chain* in a DAG is a set of vertices that all lie on at least one common path from the source to the sink, and can generally be extended to a number of different paths. Put another way, a chain is a set of reads that are all comparable to each other according to the partial ordering implied by the mate graph. While this chain decomposition is also in general not unique, the cardinality of the minimal cover is well-defined, and is an important invariant of the set of alignments. The cardinality of the minimal cover is a

lower bound on the number of transcripts needed to explain the mate alignments.

2.1 Computing the minimal set of explaining transcripts

The algorithm to compute a minimal set of explaining transcripts has four steps:

1. Construct the mate graph G_M associate with M
2. Compute a minimal chain decomposition of G_M.
3. Extend chains in the decomposition to paths from the source to the sink in G_M.
4. Output the transcript sequences corresponding to the paths.

Step one is straightforward, and consists of sorting the mate alignments by reference position, checking for overlaps, and then checking for consistency between any overlapping pairs. It has worst case complexity $O(|M|^2)$, but is likely to be fast for typical inputs.

Step two begins by taking G_M and building the associated bipartite graph with a vertex for each mate alignment, and an edge between mates x and y if there is a path from x to y in G_M. This amounts to a transitive closure calculation on G_M, which has time complexity $O(|V'||E'|)$, where here $|V'| = 2|M|$ and $|E'| = |M|^2$. Thus this step is worst-case cubic.

After building the bipartite graph from the transitive closure of G_M, the algorithm computes a minimal chain decomposition on G_M. Dilworth's Theorem (http://en.wikipedia.org/wiki/Dilworth's_theorem) states that we can construct a minimal chain decomposition from a maximum cardinality matching H on the bipartite graph. A constructive proof, which follows from König's Theorem, works by building an *antichain* on G_M. An antichain here is a set of mate alignments from G_M where there is no path between any two elements. After building the bipartite graph as above, König's Theorem says that there is a matching H and a set of vertices C in the bipartite graph, such that each edge in the graph contains at least one vertex in C and such that H and C have the same cardinality m. Let A be the set of mate alignments that do not correspond to any vertex in C; then A has at least $|M| - |H|$ elements. Now let P be a family of chains formed by including the mate alignment for x and the mate alignment for y in the same chain whenever there is an edge from a vertex representing x to a vertex representing y in the matching. P has $|H| - |H|$ chains. Therefore, we have constructed an antichain and a partition into chains with the same cardinality.

In step three, the algorithm extends the chains in the graph to paths from the source to the sink. Note that while there is not necessarily a unique extension from a given chain to such a path, any set of extensions of the chains to paths will yield a minimal explaining set of transcripts for the reads. As described in below, a path through G_M corresponds to a sequence of (possibly overlapping) read alignments of the mate alignments intermixed with regions where the transcript sequence is unknown, but where the length of the unknown sequence in the transcriptome coordinate space can be estimated. Since any set of path extensions will produce a minimal explaining set, the algorithm choosing the one which minimizes the amount unknown sequence in each path. Since the paths form a cover of the mate alignments, rather than a partition, this is essentially a trivial optimization problem which can be solved optimally with greedy, local choices during the path extension step.

Step four takes paths generated by step three and outputs the sequences for the transcripts constructible from them. Each path corresponds to a set of consistent mate alignments, and each mate alignment consists of a pair of read alignments, each of which has a CIGAR string. The algorithm produces a single CIGAR string for the entire path, along with a sequence for the path on which (along with the reference) the CIGAR string operates.

2.2 Transcript abundance estimation

We view an input mRNA sample as a probability distribution on a set of transcripts. We want to estimate this distribution from a set of observed paired-end reads. Let $\mathbf{T}$ be the set of candidate transcripts. Ideally, we would take $\mathbf{T}$ as the set of all possible transcripts, but we must limit ourselves to a small explaining set of transcripts. Using the set of transcripts produced by the algorithm above will make the abundance estimate feasible to calculate. Let $\mathbf{M}$ be the set of possible mate alignments that are compatible with transcripts in $\mathbf{T}$. Then we can write the mate alignment observation data as a vector u, where u_m is the number of times we observed mate alignment m.

The inference process is based on a statistical model for the generation of paired-end reads from

an mRNA sample. We assume that read pairs are generated as follows. First, a transcript t is drawn at random from the unknown probability distribution $p = (p_t)_{t \in \mathbf{T}}$. Next, a new mate m is generated from t by first picking a left read alignment starting at m_l at random from all positions in t. Then, a the end of the right read alignment m_r is generated by picking a length from the distribution of insert lengths and adding that length to m_l. If the length would exceed the end of the transcript, then a new length is picked from the mate pair length distribution until a valid mate pair is obtained. Estimating the structure of the population is the problem of estimating p from u under this generative model.

Let T be the hidden random variable with values in $\mathbf{T}$ that describes the transcript and M the observed random variable over $\mathbf{M}$ for the mate pair. Then the probability of observing mate pair m under this model is

$$Pr(M = m) = \sum_{t \in \mathbf{T}} p_t Pr(M = m | T = t)$$

Where the conditional probability is taken as zero if m is not consistent with t, and otherwise defined as $Pr(M = m||T = t) = (1/K)p_l(m_r - m_l)$, where K is the length of t and p_l is the mate pair length distribution, which we assume to normal and with a given mean and variance.

The algorithm then estimates p by iteratively estimating the missing data u_{mt}, the number of times mate alignment m originated from transcript t, by maximizing the log-likelihood function of the hidden model

$$\mathcal{L}_{hid}(p_1, ..., p_{|T|}) = \sum_{m \in \mathbf{M}} \sum_{t \in \mathbf{T}} u_{mt} log(Pr(M = m | T = t))$$

In the E step, the expected values of the missing data is computed as

$$u_{mt} = u_m \frac{p_t Pr(M = m | T = t)}{P(M = m)}$$

In the M step, maximization of $\mathcal{L}_{hid}$ yields

$$\hat{p}_t = \frac{\sum_{m \in \mathbf{M}} u_{mt}}{\sum_{m \in \mathbf{M}} u_m}$$

References

[1] Nicholas Eriksson, Lior Pachter, Yumi Mitsuya, Soo-Yon Rhee, Chunlin Wang, Baback Gharizadeh, Mostafa Ronaghi, Robert W Shafer, Niko Beerenwinkel, and Glenn Tesler. Viral population estimation using pyrosequencing. *PLoS Computational Biology*, 4(5), May 2008.