View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Washington University St. Louis: Open Scholarship

Washington University in St. Louis

Washington University Open Scholarship

All Computer Science and Engineering Research

Computer Science and Engineering

Report Number: WUCSE-2004-80

2004-12-10

Basecalling for Traces Derived for Multiple Templates

Aaron Tenney

Three methods for analyzing sequencing traces derived from sequencing reactions containing two DNA templates are presented. All rely on alignment to a segment of assembled genomic sequence containing the original template sequence. Spliced alignment algorithms are used so that traces derived from processed mRNA can be analyzed. The main application of these techniques is the elucidation of alternately spliced transcripts. Several experimental verification of one of the techniques is presented including testing on a set of 48 alternately spliced targets from the human genome and 47 negative controls.

... Read complete abstract on page 2.

Follow this and additional works at: https://openscholarship.wustl.edu/cse_research

Recommended Citation

Tenney, Aaron, "Basecalling for Traces Derived for Multiple Templates" Report Number: WUCSE-2004-80 (2004). *All Computer Science and Engineering Research.*
https://openscholarship.wustl.edu/cse_research/1051

Department of Computer Science & Engineering - Washington University in St. Louis
Campus Box 1045 - St. Louis, MO - 63130 - ph: (314) 935-6160.

This technical report is available at Washington University Open Scholarship: https://openscholarship.wustl.edu/cse_research/1051

Basecalling for Traces Derived for Multiple Templates

Aaron Tenney

Complete Abstract:

Three methods for analyzing sequencing traces derived from sequencing reactions containing two DNA templates are presented. All rely on alignment to a segment of assembled genomic sequence containing the original template sequence. Spliced alignment algorithms are used so that traces derived from processed mRNA can be analyzed. The main application of these techniques is the elucidation of alternately spliced transcripts. Several experimental verification of one of the techniques is presented including testing on a set of 48 alternately spliced targets from the human genome and 47 negative controls.

Department of Computer Science & Engineering

Washington University in St.Louis
School of Engineering & Applied Science

2004-80

BASECALLING FOR TRACES DERIVED FROM MULTIPLE TEMPLATES, Master's Thesis, December 2004

Authors: Tenney, A. E

December 10, 2004

Abstract: Three methods for analyzing sequencing traces derived from sequencing reactions containing two DNA templates are presented. All rely on alignment to a segment of assembled genomic sequence containing the original template sequence. Spliced alignment algorithms are used so that traces derived from processed mRNA can be analyzed. The main application of these techniques is the elucidation of alternately spliced transcripts. Several experimental verification of one of the techniques is presented including testing on a set of 48 alternately spliced targets from the human genome and 47 negative controls.

Department of Computer Science & Engineering - Washington University in St. Louis
Campus Box 1045 - St. Louis, MO - 63130 - ph: (314) 935-6160

WASHINGTON UNIVERSITY
THE HENRY EDWIN SEVER GRADUATE SCHOOL
DEPARTMENT OF COMPUTER SCIENCE

BASECALLING FOR TRACES DERIVED FROM MULTIPLE TEMPLATES

by

Aaron Tenney

Prepared under the direction of Professor Michael Brent

A thesis presented to the Henry Edwin Sever Graduate School of
Washington University in partial fulfillment
of the requirements for the degree of
MASTER OF SCIENCE
December 2004
Saint Louis, Missouri

WASHINGTON UNIVERSITY
HENRY EDWIN SEVER GRADUATE SCHOOL
DEPARTMENT OF COMPUTER SCIENCE

ABSTRACT

BASECALLING FOR TRACES DERIVED FROM MULTIPLE TEMPATES

by Aaron Tenney

ADVISOR: Professor Michael Brent

December 2004
Saint Louis, Missouri

Three methods for analyzing sequencing traces derived from sequencing reactions containing two DNA templates are presented. All rely on alignment to a segment of assembled genomic sequence containing the original template sequence. Spliced alignment algorithms are used so that traces derived from processed mRNA can be analyzed. The main application of these techniques is the elucidation of alternately spliced transcripts. Several experimental verification of one of the techniques is presented including testing on a set of 48 alternately spliced targets from the human genome and 47 negative controls.

Contents

Contents ii
Tables iv
Figures v
Introduction 1
Background 4
- 2.1 Sanger sequencing 4
  - 2.1.1 Natural DNA replication 5
  - 2.1.2 Original Sanger method 6
  - 2.1.3 Improvements on original method 8
  - 2.1.4 Output of modern Sanger sequencing reaction 11
- 2.2 Sequence alignment 11
  - 2.2.1 Global alignment 12
  - 2.2.2 Local alignment 15
  - 2.2.3 Banded alignment 15
  - 2.2.4 Linear space alignment 16
  - 2.3.5 Spliced alignment 19
  - 2.2.6 Sub-Optimal alignment 21
  - 2.2.7 Pair Hidden Markov Models (HMMs) 21
  - 2.2.8 Continuous density HMMs 25
- 2.3 Basecalling 26
  - 2.3.1 Issues in basecalling 27
  - 2.3.2 PHRED 31
- 2.4 Biology 38
  - 2.4.1 alternate splicing 38
  - 2.4.2 Indel polymorphisms 40
  - 2.4.3 T-DNA mediated knockout libraries 41
  - 2.4.4 Mispriming 42

Trace Recalling Method 43
- 3.1 Early implementation 43
  - 3.1.1 Mixed sequence 44

    - 3.1.2 Alignment 45
    - 3.1.3 Dump PHRED state 47
    - 3.1.4 Recalling 47
    - 3.1.5 Realignment of secondary sequence 50
  - 3.2 Later refinements 50
    - 3.2.1 PHRED –d option 50
    - 3.2.2 Ambiguity alignment 51
  - 3.3 Identification of alternate splices 52
- Integrated Alignment Method 54
  - 4.1 Theory 55
  - 4.2 Complexity/Optimization Issues 58
    - 4.2.1 Banding 58
    - 4.2.2 Linear space alignment 60
  - 4.3 Implementation Details 61
    - 4.3.1 Scoring alignments between predicted and observed peaks 62
    - 4.3.2 Using only the top 3 scoring observed peaks 63
    - 4.3.3 Choice of mismatch and gap penalties 65
- Pair-HMM Method 69
  - 5.1 Alignment Sequences 69
  - 5.2 Model Topology 70
  - 5.3 State Emission Probabilities 73
  - 5.4 Implementation and Training 73
- Results 76
    - 6.1.1 Description of synthetic traces 76
    - 6.1.2 Results of trace recalling on synthetic traces 78
  - 6.2 Dual Sequencing Primer Experiments 78
    - 6.2.1 Description of dual primer set 78
    - 6.2.2 Results of trace recalling on dual primer set 79
    - 6.2.3 Results of pair-HMM method on dual primer set 82
  - 6.3 Alternate Splice Experiments 84
    - 6.3.1 Description of alternate splice test set 84
    - 6.3.2 Results of trace recalling on alternate splice test set 85
    - 6.3.2 Results of integrated alignment on alternate splice set 91
- Conclusions 93
  - 7.1 Discussion of experimental results 93
    - 7.1.1 Discussion of synthetic trace experiment 93
    - 7.1.2 Discussion of dual primer experiment 94
    - 7.1.3 Discussion of alternate splicing experiment 96
  - 7.2 Applications and future work 99
- References 101
- Vita 104

Tables

4-1. Parameters used for integrated alignment. 67
6-1. Results of dual primer experiment.. 80
6-2. Successful experiments.. 89
6-3. Unsuccessful experiments 90

Figures

1-1. Example of a trace derived from two templates (double trace)................................ 1
2-1. Sequencing gel from the original Sanger paper (Sanger et al. 1977)........................ 8
2-2. Example of a clean trace... 11
2-3. Illustration of the divide and conquer Myers-Miller algorithm (Myers and Miller 1988).. 17
2-4. Pair-HMM for global alignment (Durbin et. al 1998).. 22
2-5. Comparison of raw and processed trace data (Haan and Godsill 2002).................. 30
2-6. Locations of predicted (blue) and observed (yellow) peaks.................................. 34
2-7. Function used by PHRED to score alignments between predicted and observed peaks. Top, range over which function is applied in practice. Bottom, larger range to illustrate non-linearity of positive range of function... 37
2-8. Different types of alternate splces (Cartegni et al. 2002)...................................... 39
3-1. A double trace (top), example of original trace recalling method (middle) and trace recalling with ambiguity sequence (bottom)... 45
3-2. Flowchart describing recalling of single base 49
3-3. Example alignment of original (red) and recalled (blue) sequences to genome and indicator string used to classify alternate splice type... 53
4-1. A hypothetical multiple alignment as described in text. 57
4-2. Illustration of memory savings from banding. 60
5-1. Pair-HMM model topology for aligning a trace to genomic sequence. 71
6-1. A section of double trace with first and second stage base calls and alignments (details in text). 82
6-2. Example of a successful experiment. 87
7-1. An alignment artifact.. 98

Chapter 1

Introduction

This thesis addresses the problem of extracting usable DNA sequence data from a sequencing trace that was generated from multiple DNA templates. We encountered this problem in the context of using RT-PCR followed by direct sequencing to verify predicted gene structures (Wu, et. al, 2004). An example of a trace, which we encountered, is shown in figure 1-1.

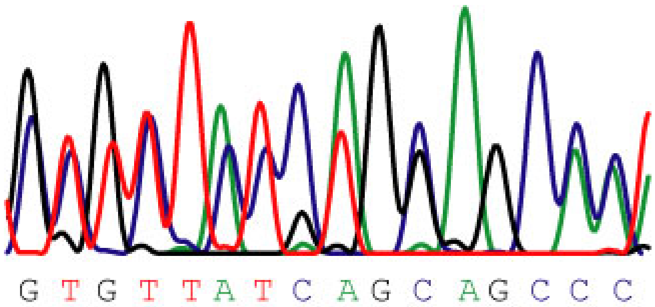


Figure 1-1. Example of a trace derived from two templates (double trace).

As can seen in figure 1-1 at most peak positions there are two well-defined peaks corresponding to two nucleotides at each position. Since the DNA template in this example was derived from reverse transcribed mRNA the cause of the multiple templates was alternate splicing of the underlying gene. It is not clear that there is a simple method for recovering the two underlying DNA template sequences by simply analyzing such a sequencing trace in isolation. One might assume that the two superimposed traces could be differentiated by peak height but in practice the peak heights for the two traces are often very similar. In many cases calling the highest peak at each position results in a sequence that alternates between the two template sequences.

We have developed 3 methods for analyzing traces such as the one pictured in figure 1-1. They combine traditional basecalling and sequence alignment techniques to align the multiple template trace to an assembled genomic sequence. All methods result in finding the sequence and genomic location of the templates that gave rise to the multiple traces.

Chapter 2 provides background needed to understand the work presented in later chapters. It starts with a review of the Sanger method of DNA sequencing followed by a review of base calling and sequence alignment algorithms and a discussion of the biological mechanisms that give rise to multiple traces. Chapter 3 describes the first method for analyzing multiple traces, *trace recalling.* Chapter 4 describes the second method, *integrated alignment*. Chapter 5 presents a pair-HMM based solution. In chapter 6 results of experimental tests of the methods are described and analyzed. Chapter 7 summarizes the work completed so far on this project and flaws and weaknesses of our methods.

Chapter 2

Background

This chapter provides the background knowledge necessary to understand the methods presented in the next three chapters. We will first review the Sanger method of DNA sequencing. The next two sections provide reviews of base calling and sequence alignment algorithms. The chapter ends with a discussion of the biological mechanisms responsible for multiple traces and how these relate to biological applications of these methods.

2.1 Sanger sequencing

In 1977 two methods for sequencing DNA templates were developed, the chemical cleavage method (Maxim and Gilbert 1977) and the chain termination method (Sanger et al. 1977). The chain termination method of Sanger has become has become the most widely used especially for high throughput sequencing projects. For this reason all data presented in this project was obtained by the Sanger method. To understand how double traces such as the one in figure 1-1 arise it is necessary to understand how Sanger sequencing works. Sanger sequencing is a modification of the natural process of DNA replication. Since current methods of DNA sequencing are much different than the original method outlined by Sanger I will next present the original core Sanger method followed by improvements to the method.

2.1.1 Natural DNA replication

DNA replication begins with an enzyme called a helicase unwinding the complementary DNA strands of the double helix. Next, a complex called the primosome manufactures a short complementary RNA primer on the exposed single stranded DNA (ssDNA) creating a short segment of double stranded DNA (dsDNA) which marks the 3’ end of the complementary DNA sequence to be generated. The next step is referred to as elongation. An enzyme called DNA polymerase attaches between the dsDNA and ssDNA regions and catalyzes a reaction that pulls a deoxynucleotide triphosphate (dNTP) complementary to the next base of the original ssDNA molecule from solution and attaches it to the new complementary strand. The dNTPs are simply the nucleotides, which make up the DNA molecule with two extra phosphate groups attached. Energy released from the cleavage of these phosphates during incorporation of the new nucleotides drives the elongation step. If allowed to run to completion the elongation step will proceed until the original template is exhausted or a region of dsDNA is encountered.

2.1.2 Original Sanger method

Each step of the original Sanger method mirrors natural DNA synthesis until the elongation step. Sequencing begins with purification of the single stranded DNA (ssDNA) template, which contains the desired sequence. This can be done in many ways including chemical or thermal denaturation of a double stranded DNA template derived from a plasmid containing the sequence of interest in bacterial culture or packaging of the DNA into viral particles which strip off the complementary strand. The original genome sequenced by Sanger was that of bacteriophage ϕX174 that is naturally single stranded so this step was not necessary. Once the ssDNA template is prepared a synthetic sequencing primer is allowed to anneal to the template at the 3' end of the sequence of interest. The sequencing primer is a short ssDNA molecule (~20 bases) complementary to the region just upstream of the region to be sequenced. Annealing refers to the process of hydrogen bonds forming between the nucleotides of the primer and the complementary region of the template so that the primer becomes fixed to the template. Next 4 separate sequencing reactions are set up, one for each nucleotide type, containing all species of dNTP, a smaller concentration of one species of ddNTP (dideoxynucleotide triphosphate) and DNA polymerase. In all reactions one of the species of dNTP is radio labeled so that the results can later be imaged on a gel. In the original publication Sanger used ^{32}P

labeled dATP. For example the A reaction is set up by adding all species of dNTP (with radio labeled dATP), ddATP and DNA polymerase. The ddNTPs differ from the dNTPs in that their 3’ hydroxyl groups are replaced by hydrogen. Synthesis of the complementary DNA strand begins as in normal DNA replication starting at the sequencing primer and proceeding in the 5’ to 3’ direction. The DNA polymerase does not differentiate between the dNTPs and ddNTPs and whenever a ddNTP is incorporated into the growing complementary strand, DNA synthesis halts. This is because the 3’ hydroxyl group, missing in the ddNTPs, acts as a socket onto which the next nucleotide is attached. This reaction results in all complementary prefixes of the template DNA ending in the base of the corresponding reaction (A, C, G or T) being generated. These reaction products are loaded into 4 separate lanes of a polyacrylimide gel spiked with a danaturating agent such as urea and separated by length. The denaturating agent is necessary to separate the newly synthesized DNA prefix strands from the parent template. Polyacrylimide (as opposed to agarose) is still routinely used in sequencing because it has sufficient resolution to separate DNA strands that differ in length by as little as one nucleotide. Once the sequencing gel has been run it is a simple task for a human operator to read the sequence off of the gel. An example gel from the original Sanger paper along with the bases called from the gel can be seen in figure 2-1.

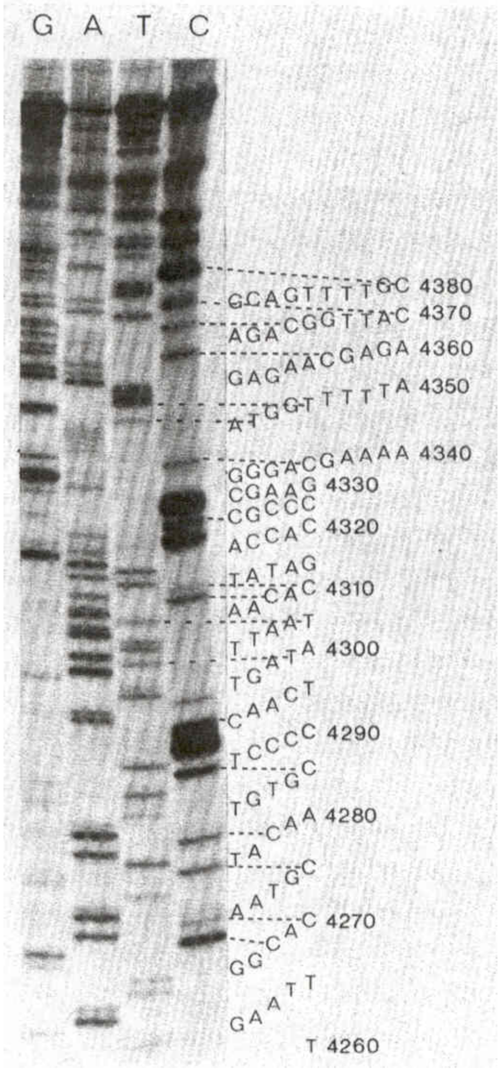


Figure 2-1. Sequencing gel from the original Sanger paper (Sanger et al. 1977).

2.1.3 Improvements on original method

Since the original Sanger chain termination method was described many improvements have been made resulting in today's standard sequencing protocol.

Fluorescent reporters. In the early days of sequencing the DNA fragments generated by the Sanger method had to be radioactively labeled to be imaged on the sequencing gel. While imaging by autoradiography provides very sharp gel images it carries with it the overhead of dealing with radiation. More importantly it meant that 4 separate sequencing reactions were necessary each requiring a separate lane on the gel. This changed with the introduction of fluorescently labeled reporter dyes (Prober et al. 1987). In this modification to the Sanger method flourescently labeled dye molecules are attached either to the sequencing primer or ddNTPs that halt chain elongation. During electrophoresis the gel is scanned by a laser which excites the dye causing it to re-emit photons on different wavelengths which are picked up by a detector. If the dye is attached to the primer (primer chemistry) the sequencing run still requires 4 gel lanes but if the dye is attached to the ddNTPs (terminator chemistry) all 4 reactions can take place in 1 solution and be run on 1 gel lane. This is because each ddNTP species is labeled with a dye that emits at a different wavelength. The output of the detector is commonly referred to as a chromatogram. An additional advantage of dye terminator chemistry is that it results in fewer compression problems caused by secondary structures forming at the 3’ end of the DNA sequence. Presumably this is because the dye molecules interfere with the base pairing which cause compressions.

Capillary sequencing. The next major advance in sequencing technology was the introduction of capillary sequencing (Swerdlow 1990). Prior to capillary sequencing the media of electrophoretic seperation was a slab of polyacrylamide gel. Slab gels have several drawbacks. Pouring the gel is labor intensive. This limited the throughput of sequencing machines. They are also subject to the phenomena of "Joule heating" whereby at high voltages a thick slab gel is unable to dissipate the heat generated by the applied electric field and melts. It is desirable to run sequencing gels at a high voltage because the speed at which the DNA fragments migrate through the gel is linearly related to the voltage applied to the gel, higher voltages translate to higher throughput. Capillary sequencing reduces the problem of Joule heating by carrying out electrophoresis in capillary tubes with an inner diameter of about 50 μm. Preloaded capillary tubes also drastically reduce the time required to prepare for electrophoretic separation and the amount of gel required. They also solve the basecalling problem of lane tracking errors in which two gel lanes merge during basecalling resulting in a chimeric read.

Automation. One of the main sub-goals of the human genome project was to develop technologies for automating DNA sequencing to increase throughput to a point where sequencing a large genome was feasible. To this end many of the physical tasks of sequencing such as setting up reactions and loading gels (or capillary tubes) are now performed by robots. Gathering the chromatogram data has been automated since the introduction of fluorescent dyes. Many base calling tools for extracting sequence from chromatograms, which will be discussed in the next chapter, have been developed as well as tools for storing and assembling individual reads into larger contigs.

2.1.4 Output of modern Sanger sequencing reaction

Before moving on to the topic of base calling it will be helpful to actually see some examples of the final product of a Sanger sequencing run, chromatograms (or traces). Figure 2-2 shows a clean trace. Notice that it consists of several large, well-defined equally spaced peaks representing flourescently labeled DNA fragments of different length migrating through the gel. This is what we expect if there is one DNA template in the sequencing reaction. Figure 1-1 shows a trace in which two different DNA templates are present in the sequencing reaction. The peaks are still evenly spaced, however, at most locations there are clearly DNA fragments representing 2 different terminal bases. Currently a trace such as this is simply discarded. It is the goal of this thesis to extract useful information from such a trace.

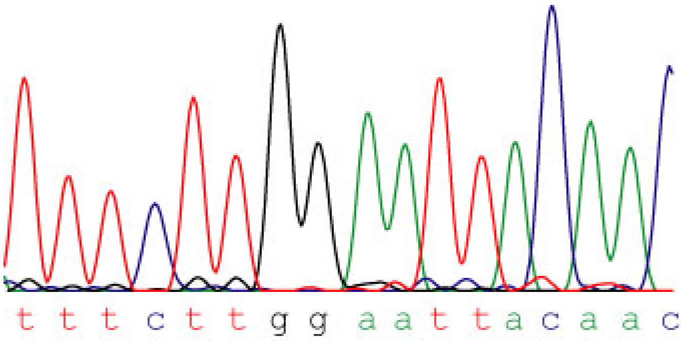


Figure 2-2. Example of a clean trace.

2.2 Sequence alignment

Alignment is used in several different ways in this work, so a review of alignment algorithms is appropriate. This section will discuss both global and local alignment followed by two methods to optimize time and memory usage of alignment algorithms, banding and linear space alignment. Spliced alignment will be discussed since many alignment steps involve aligning spliced sequences to the genome. A method for finding a specific class of sub-optimal alignments will be described since this is used in one of the methods implemented. Finally alignment using pair-HMMs and continuous density pair-HMMs will be discussed.

2.2.1 Global alignment

Sequence alignment refers to the identification of corresponding subsequences of two or more strings. In the context of biology this usually means finding homologies between related protein or nucleic acid sequences. The simplest form of sequence alignment is global alignment. Global alignment algorithms assume that the two sequences under consideration align end to end. This makes sense in the context of aligning proteins known to be homologous, which was the major application of the first global alignment algorithm. Global alignment as first described (Needleman and Wunsch 1970) required $O(M^2N)$ time and $O(MN)$ memory where M and N are the lengths of the sequences being aligned. Subsequently it was discovered that the time complexity of the algorithm can be reduced to $O(MN)$ (Gotoh 1982) and it is this solution to the problem which is commonly implemented and described in textbooks. Broadly, alignment works by defining a function from the set of all possible alignments to the real numbers (a scoring model) and selecting the alignment which scores highest under the model. For practical reasons this function takes the form of defining a score for every potential atomic unit of the alignment (match, mismatch, gap) and defining the score of the overall alignment to be the sum of the scores of these base cases. Clearly it is not practical to enumerate all possible alignments between two sequences and their scores, since the number of alignments is exponential in the sum of the lengths of the two sequences. The optimization technique of dynamic programming (Bellman 1957) is employed to achieve the time and memory complexities described above. Dynamic programming reduces the time complexity of the problem from exponential to polynomial by building up the optimal alignment from base cases. The vast majority of the computations necessary in a naïve implementation are redundant and thus can be recorded and referred to in subsequent steps. The central data structures of a dynamic programming alignment implementation are the score and traceback matrices denoted S and T. If the sequences being aligned are A and B of lengths M and N, the indices (i,j) of these matrices run from 0 to M and 0 to N. Each entry of the score matrix S(i,j) contains the score of the optimal alignment between the prefix A[1]..A[i] and B[1]..B[j], the value contained in S[M][N] is the score of the optimal alignment. The traceback matrix T is used to actually recover the optimal alignment by following the path through the matrix that gave rise to the optimal score. The alignment algorithm begins by initializing the 0th row and column of S and T as in eqn 2-1.

$$\begin{aligned} S[i][0] &= i \cdot gap \\ S[0][j] &= j \cdot gap \end{aligned}, \tag{2-1}$$

where *gap* denotes the penalty for aligning a gap to a character in the other sequence. This agrees with the definition of the S matrix since the score of aligning the prefix of either sequence entirely to gaps is simply the length of the prefix so aligned times the penalty of a single gap. The traceback pointers reflect that this is the alignment being described in the initialization. Next the rest of the S and T matrices are filled with the recurrence relation in eqn 2-2.

$$S[i][j] = \max\begin{cases} S[i-1][j-1] + s(A[i], B[j]) \\ S[i-1][j] - gap \\ S[i][j-1] - gap \end{cases} \tag{2-2}$$

s(A[i], B[j]) is the match function between characters in the sequence. In the case of nucleic acids this is usually a positive value if the characters are the same and negative otherwise. When aligning protein sequences this can be a number derived from a scoring matrix such as BLOSUM (Henikoff and Henikoff 1992) or PAM (Dayhoff et. al 1978) which measures the similarities of amino acids or the probability of seeing two specific amino acids in an alignment. The top case in the max can be understood as asserting that the characters A[i] and B[j] align to each other. This alignment can be either a match or a mismatch. The next case asserts that A[i] aligns to a gap in B and similarly the final case asserts that B[j] aligns to a gap in A. T[i][j] is filled in to indicate which choice was taken in the max. Recovery of the optimal alignment involves beginning a traceback starting at T[M][N], following the path that generated the optimal score and employing the rules described above to write out the optimal alignment. For proof that this algorithm results in an optimal alignment, see (Bellman 1957; Gusfield 1997).

2.2.2 Local alignment

Local alignment (Smith and Waterman 1981) is a modification of global alignment, which does not require the sequences to align end to end but finds the optimal subsequence alignments between the two sequences. Instead of initializing the score matrix as before, the 0th row and column are initialized with zeros. A zero is also considered in the max of the matrix fill step. Finally, the highest scoring cell is kept track of during recursion. Traceback begins from this cell instead of T[M][N], the traceback terminates when a cell with score zero is encountered. This has the effect of allowing free end gaps in both sequences of the alignment, examining only positive scoring alignments, and recovering the best subsequence alignments between A and B. Local alignment makes possible alignment of a short sequence (such as a spliced mRNA) to genomic sequence and database searching.

2.2.3 Banded alignment

There is no specific paper describing banded dynamic programming. It seems to be a technique that most people know about, but which has never been published. It has the potential to greatly reduce the amount of time and memory needed to perform a global or local alignment. The basic idea is that if the problem specification restricts which paths through the dynamic programming table represent true alignments it is not necessary to examine the regions of the table that cannot generate one of the potential desired alignments. For example, in the case of aligning two closely related proteins it is known that the correct alignment will never stray very far from the diagonal of the dynamic programming matrix. As a result, to get the optimal alignment it is sufficient to compute just a small "band" around the diagonal. The technique's name derives from this application. This optimization effectively reduces the time and space complexity of the problem from O(NM) to $O(b\sqrt{(N^2+M^2)})$ where b is the width of the band. This represents a reduction in the asymptotic complexities from quadratic to linear in both time and space. The width of the band need not be fixed; a variable width band can change from row to row. Nor must the band be constrained to follow the diagonal. An example implementation of such a system will be discussed in greater depth in chapter 4.

2.2.4 Linear space alignment

If the problem is amenable, banding can reduce both the time and memory complexities of a dynamic programming alignment algorithm from quadratic to linear. Another method for reducing the memory but not time complexity of global alignment has been described (Myers and Miller 1988) which places no search space constraints on the problem. This method relies on the fact that filling in the dynamic programming matrix can be iteratively decomposed into small sub-problems each of which can be solved using an amount of memory linear in one of the sequences. A figure from the original Myers & Miller paper fig 2-3 illustrates this technique.

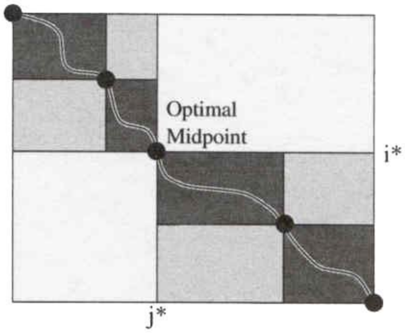


Figure 2-3. Illustration of the divide and conquer Myers-Miller algorithm (Myers and Miller 1988).

First note that the optimal alignment can be found either by filling in the matrix starting at the top left or bottom right corner and tracing back. Next note that whatever the optimal alignment turns out to be it must pass through the middle row of the matrix. Furthermore the column in which the optimal alignment crosses the middle row can be computed in linear space. This is accomplished by computing the values in the middle row by working down from the top row as usual. Next the values in the middle row are computed by working up from the bottom row. This can be done in linear space since only scores are being computed (no traceback pointers are kept) and to fill in any cell of the matrix requires storing only the current and previous rows. The corresponding values in the two middle rows thus computed are summed. Each of these values represents the score of the optimal alignment constrained to cross the middle row at the corresponding column. Thus, the middle row cell with the highest sum must lie on the optimal traceback. This creates 2 sub-problems as shown in figure 2-3, aligning the prefixes and suffixes of the sequences. Two more points on the optimal alignment can be found by solving these sub-problems in the same manner as the first. The whole optimal alignment is thus obtained by iteratively solving smaller and smaller sub-problems. When the subsequences being aligned are sufficiently small normal global alignment with traceback can be used to solve the base cases. While this algorithm does increase the running time of the alignment it can be shown that it does not increase the time complexity beyond quadratic.

A potential problem with this technique is that it works only for global alignment. This is solved, though, by noting that any local alignment can easily be converted to a global alignment between subsequences of the original problem in linear space. This works by filling in the local alignment dynamic programming matrix as usual from top to bottom keeping only the scores and the coordinates of the highest scoring cell in linear space as described above. Next fill in the matrix from bottom to top in a similar way, keeping the coordinates of the highest scoring cell. At the end of this procedure the beginning and ending coordinates of the optimal local alignment are the coordinates of the optimal scoring cell from the forward pass and the coordinates of the optimal scoring cell from the backward pass respectively. These coordinates define subsequences of the original sequences on which global alignment can be performed whose solution is the same as the solution to the local alignment problem on the original sequences.

2.3.5 Spliced alignment

The main applications of the methods developed in this thesis involve aligning traces derived from alternately spliced mRNA to genomic sequence. As a result, spliced alignment algorithms are heavily used in the algorithms developed. Spliced alignment refers to standard global or local alignment with an additional type of gap used to model long intron gaps. There are several spliced alignment programs available, but for this work EST_GENOME (Mott 1997) was used because it is a published algorithm and source code was available. In addition to the affine gaps present in global or local alignment, which in this context represent sequencing errors or polymorphisms, there is an intron gap possible in the spliced sequence but not in the genomic sequence. The score penalty of an intron gap differs from the usual affine gap penalty in two ways. First the penalty does not depend on the length of the gap, it is constant. Second the score of the gap takes into account consensus splice signals at the ends of the gap. A gap with consensus splice GT/AG gets penalized less than one without this consensus sequence. This makes it easier to pull out the correct alignment of the spliced sequence to the genome while still making it possible to get an alignment without the consensus sequence. Non-consensus splice sites must be possible because a small fraction of real splice sites do not possess the GT/AG consensus sequence.

Two modifications to the standard global or local alignment recursion and two new data structures make spliced alignment possible. As described in the EST_GENOME (Mott 1997) paper the data structures are two arrays called B and C, which are essentially extensions of the standard dynamic programming score and traceback matrices. The indices of these arrays run the length of the spliced sequence. The i^{th} entry of the B matrix holds the score of the best alignment with an intron gap at the i^{th} position of the spliced sequence. The i^{th} entry of the C matrix holds the position in the genomic sequence where this optimal intron gap begins. These values are updated in the innermost loop of the dynamic programming recursion. Once the maximum score of a cell is computed it is compared to B[i], if the max is greater B[i] takes on this value and C[i] takes the coordinate of the genomic sequence to which that this cell refers. This means that if an intron gap were to be subsequently inserted into this row of the matrix tracing back to the cell where C[i] points would yield a higher score over than any cell seen so far in this row. In the fill step an extra term is added to the max step making use of B and C to score potential intron gaps. If there is a consensus splice GT/AG at positions C[i] and j (the beginning and end of the potential intron gap) B[i] – splice_penalty is considered in the max step. If C[i] and j do not represent a consensus splice signal B[i] – intron_penalty is considered. The value of intron_penalty is higher than that of splice_penalty. The C[i] value answers the question which arises in the fill step, “if I wanted to insert an intron gap at this position (make a long range jump in the matrix) which cell in this row should I jump back to to get the best overall score”.

2.2.6 Sub-Optimal alignment

Several methods for finding sub-optimal alignments have been described in the literature. This project employs a method first described by Waterman and Eggert (Waterman and Eggert 1987). It differs from other sub-optimal alignment finding algorithms in that it finds sub-optimal alignments that differ significantly from the optimal alignment instead of differing in only one or a few positions. This method works by first finding the optimal alignment (global or local using any of the optimizations described above). The cells through which the optimal alignment passes are stored, and the optimal alignment is re-computed. This time cells that appeared in the optimal alignment are given a score of 0 regardless of the score they would have received. This results in a sub-optimal alignment that cannot overlap the optimal alignment.

2.2.7 Pair Hidden Markov Models (HMMs)

All of the alignment variations described above are based on the Needleman-Wunsch algorithm. Pair Hidden Markov Models (Pair-HMMs) provide an alternate framework in which to consider the sequence alignment problem. They are an extension of ordinary (typically discrete emission) Hidden Markov Models (HMMs). A general discussion of HMM theory is beyond the scope of this thesis, however, a general introduction to HMMs can be found in (Rabiner, 1989) and an introduction as applied to biological problems in (Durbin et. al 1998). A full development of HMM theory is presented in (Elliott et al 1995). The pair-HMM framework extends the classical sequence alignment framework in two important ways. First, it provides a probabilistic interpretation to the scores (match, mismatch, gap, etc) associated with the classical algorithms. This interpretation makes possible the automatic training of model parameters starting from unlabeled data. Second, it provides a simple and correct means to specify alignment models more complex than the classical match, mismatch, and gap model.

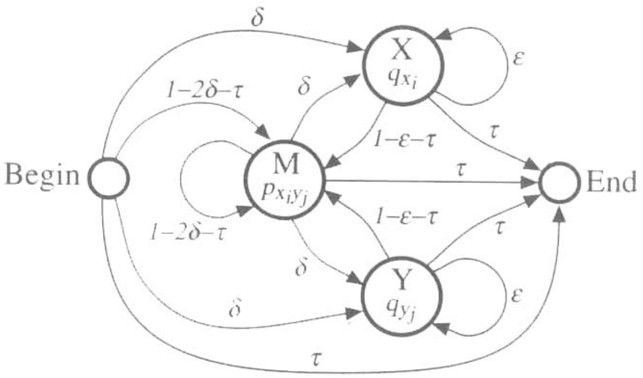


Figure 2-4. Pair-HMM for global alignment (Durbin et. al 1998)

The canonical Pair-HMM, which solves the global sequence alignment problem as described in (Durbin, et. al, 1998), is presented in figure 2-4. This model can be thought of as generating all possible alignments between two sequences of arbitrary length $X=(x_1, x_2, \dots x_n)$ and $Y=(y_1, y_2, \dots y_m)$ and assigning each one a probability. The three states in the middle of the model represent all possible atomic building blocks of a global alignment. The M state represents aligned bases (either matched or mismatched), the state labeled X represents a base in the X sequence aligned to a gap in the Y sequence. The state labeled Y represents a base in the Y sequence aligned to a gap in the X sequence. The states are thought of as emitting these atomic building blocks of the alignment and each possible emission is associated with a probability. In the case of DNA sequence, which has an alphabet of 4 letters, the M state is defined by 16 emission probabilities because there are 4 ways to emit a match and 12 ways to emit a mismatch. Similarly the X and Y states are each defined by 4 emission probabilities. States labeled begin and end are non-emitting states which model the beginning and end of the alignment. Directed arrows from one state to another denote transitions between the states. Each transition is associated with a probability, the transition probabilities out of any given state must sum to 1. The model generates an alignment in the following manner. Start at the begin state and based on the transition probabilities out of begin move to a new state without emitting anything. Depending on which state was chosen emit an aligned pair (M) or a gap (X or Y) using the new state's emission probabilities. Move on to the next state based on the new state's transition probabilities. This process is repeated until the end state is encountered which terminates the alignment. The probability of the alignment given the model generated in this manner is the product of all emission and transition probabilities encountered while traversing the graph. Since all possible alignments can be generated by the graph (with associated probabilities) the model can be used to rank the alignments by their probabilities. Usually we desire the alignment with the highest probability given the model. This can be found efficiently with the pair-HMM version of the Viterbi algorithm.

A pair-HMM is defined by the model toplolgy (states, emission alphabets of the states, and non-zero probability transitions between states) and the transition and emission probabilities. The model topology defines the set of alignments, which will be considered. For example if there is no transition between the begin and match states any alignment which begins with a match or mismatch will be implicitly assigned a probability of zero and excluded from consideration. Model topology is based on prior assumptions of what a correct alignment should look like. This may be different for different specific applications. Once the model topology is defined it is possible to obtain locally optimal transition and emission probabilities if training examples (even unlabeled training examples) are available. We obtain these values with the Baum-Welch EM estimation algorithm (Baum 1972). The objective of the Baum-Welch algorithm is to adjust the transition and emission probabilities of the pair-HMM so as to maximize the probability of the training data given the model. Initially random values are selected for the model transition and emission probabilities. Forward and backward variables are computed for each pair of sequences in the training set in a manner similar to the Viterbi algorithm. These variables provide a means to determine the expected number of times each emission and transition is used given the model and the observed sequences, the expectation is taken with respect to all possible alignments. The expected counts for each transition and emission for each training example are pooled and used to compute a new set of transition and emission probabilities. This procedure iterates until the parameters converge or a stopping criterion is met. It can be shown that the Baum-Welch algorithm is guaranteed to converge however the maximum it finds may not be the global maximum. For this reason it is desirable to either begin with parameters thought to lie near the global maximum, run the algorithm several times with different starting parameters, or both.

2.2.8 Continuous density HMMs

The overwhelming majority of HMMs used in computational biology are discrete density HMMs. This means that the emission alphabet of the states is finite. This makes sense because typically the sequences being parsed (HMMs) or aligned (pair-HMMs) are either DNA with an alphabet of size 4, or protein with an alphabet of size 20. Pair-HMMs in which the sequences being aligned have discrete alphabets are also discrete since the alphabet of the pair-HMM states is the cross product of the alphabets of the sequences plus the gap symbol. There is a large literature (Young, et. al 1995; Wheddon and Linggard 1990), mostly from computational linguistics, in which continuous density HMMs (CD-HMMs) are developed. In a CD-HMM the state emission symbols are drawn from an uncountably infinite alphabet such as the real numbers. It is still necessary, however, to assign probabilities to each emission from each state. This is accomplished by defining a continuous p.d.f. for each emitting model state. These usually take the form of a parameterized probability distribution (typically a normal or a weighted sum of normals). An approximation of the probability of an observation symbol from a specific state is obtained by evaluating the p.d.f of the state at the observation symbol. It can be shown that Baum-Welch algorithm remains valid in CD-

HMMs if the discrete emission probabilities are replaced with the parameters of the p.d.f.s at each state.

It is possible to make use of continuous valued data in a discrete HMM, however, this causes many problems. In such a model it is necessary to discretize the continuous data in order to model it with a discrete state HMM. This leads to a loss of accuracy in the resulting model. To minimize the loss of accuracy the discretization can be made very fine, but this increases the number of parameters needed making over-training a problem. These problems are elegantly solved by the CD-HMM formulation.

2.3 Base calling

Base calling refers to the process of extracting a string representing the sequence of the underlying DNA template from a trace. The nature of trace data stipulates certain issues that every base calling algorithm must address. These will be discussed in the first section. A review of base calling algorithms will be presented in the next section. Finally a detailed description of the PHRED base calling algorithm will be presented as well as the reasons this package was chosen as the starting point of this research.

2.3.1 Issues in base calling

A perfect sequencing trace would look like the trace presented in figure 2-2 across its entire length. As discussed earlier good traces possess two properties. The first is regular spacing. The peaks in the trace represent collections of DNA fragments which each differ in length by a single nucleotide. Since each species of nucleotide is the same length the distance between peaks should be the same. The second property is that the peaks are large and well defined. When sequencing a single template there is a unique base at each position. This should be reflected in the trace by the presence of a large peak in one channel and a low background in the other three at each position.

There are several reasons for deviations from the perfect trace. These can broadly be broken down into gel, dye, DNA and detector effects.

Gel effects. The speed at which DNA fragments migrate through the gel is partially influenced the density of the gel. The denser the gel the more time it takes a DNA fragment to move through it. Ideally the sequencing gel has a perfectly uniform density resulting in completely predictable migration times. This is obviously not possible in practice. Local variations in gel density can cause peaks to be shifted toward the front of the trace (faster migration) in regions of low local density and toward the back of the trace (slower migration) in regions of high local density. In practical terms this means that it is impossible to predict the precise location of peaks *de novo*. Local density variations are not a major problem if the slab gel is carefully prepared. It has become even less of a problem with the introduction of pre-loaded capillary gels. A more serious gel problem is an air bubble in the gel especially in capillary gels. An air bubble in a capillary gel tube physically separates the gel into two regions making it impossible for the DNA to migrate the length of the tube, ruining the sequencing run.

Another slab gel related problem is lane tracking errors. When a gel runs the DNA fragments should migrate parallel to the electric field applied to the gel. If there is a slight skew due to improper placement of the gel into the sequencing machine the lane tracking algorithm may run two lanes together. This problem has been completely solved by capillary gel sequencing since the “lanes” in such a system are confined to the capillary tubes that are rigidly fixed relative to the detector.

Dye effects. All modern sequencing uses fluorescent labels, usually attached to the chain terminating nucleotide. Two properties of these dyes affect the final trace; electrophoretic mobility and intensity. Dye mobility refers to the tendency of DNA fragments attached to different dye molecules to migrate at different rates. If not taken into account mobility effects cause whole channels of the trace to be shifted relative to each other resulting in non-uniform spacing of peaks. Mobility rates, however, depend only on the dye molecule and are easily determined experimentally. Hence one of the first functions applied to the trace is “mobility correction” to correct for these predictable dye effects.

The other dye effect has to do with the relative intensities of the fluorescent signal emitted by the different dyes and the ability of the detector to pick up these signals. Certain dyes fluoresce much more brightly than the other dyes at comparable concentrations and the detector is more sensitive to light emitted by certain dyes. This might cause a noise peak in some channel to appear more significant than the peak representing the correct base. Dye intensity effects like mobility effects are constant for any particular dye/detector system and can be experimentally determined prior to the sequencing run. Therefore another pre-processing step on the raw trace involves correcting for these dye intensity effects.

DNA effects. Several DNA templates related effects can also cause deviations from a perfect trace. The most notable of these results from hairpin secondary structures, which can form in the ssDNA template, called GC or CC compressions. These structures alter the electrophoretic mobility of the DNA template resulting in trace peaks that appear much closer than expected. This problem has been partially solved by the use of dye terminator chemistries. Another common problem results from long runs of the same base in the template. In the trace this can appear as one large peak making it hard to determine exactly how many bases are in the repetitive region. Another problem related to the sequencing of very G/C rich sequences is referred to as a “stop”. These are caused by very strong base pairing of the template DNA to the complementary strand or itself, which make it impossible for DNA polymerase to extend past the G/C rich region. This is reflected by an abrupt premature termination of the trace, a “stop”.

Detector effects. Finally the detector used to collect the fluorescence from the gel is not perfect. Detection of fluorescence signals is inherently less sharp than autoradiography of radio labeled DNA but the advantages of fluorescence far outweigh this drawback. It is worth noting that the traces commonly seen in textbooks and even with trace viewing software are heavily smoothed and do not represent the actual intensities recorded by the detector. A graphic example of this can be seen in figure 2-5, which depicts the raw trace data and corresponding smoothed trace.

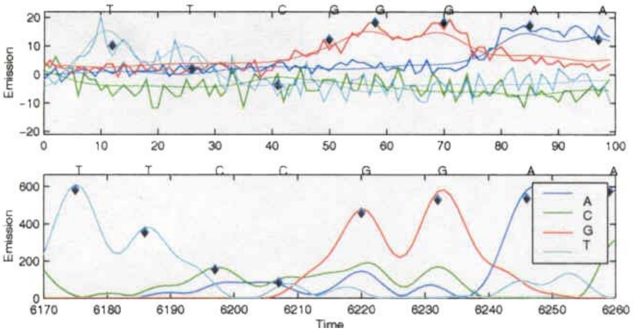


Figure 2-5. Comparison of raw and processed trace data (Haan and Godsill 2002).

2.3.2 PHRED

After reviewing several base calling programs we decided that PHRED was the best base from which to build. First, its source code is freely available for academic use. It is a successful program as shown by the fact that most base callers published in the past several years use PHRED as a benchmark. These newer algorithms usually do only modestly better than PHRED. Furthermore the internal mechanics of the PHRED algorithm are amenable to the types of modifications we had envisioned at the start of this project. Since PHRED algorithms and abstractions are at the heart of the integrated alignment method a short review of the relevant parts of the PHRED algorithm is included.

PHRED takes as input a trace file in .ABI or .SCF format. Lane tracking, smoothing and corrections for dye mobility effects and dye intensity/detector sensitivity effects are assumed have been performed on the trace. A broad outline of the PHRED algorithm can be described as follows. Locate predicted peaks, locate observed peaks, align predicted and observed peaks and pick up any good stray peaks left over. These steps will be described in greater detail below.

Find predicted peaks. PHRED predicted peaks address the first base calling issue mentioned in section 2.3.1, that of regular spacing. The method of obtaining the predicted peaks begins by looking at a skyline projection of a small section of the trace to identify the exact locations where there “should” be a peak. First a set of “detected peaks” is generated. These are defined as local maxima or midpoints between maxima in any of the 4 color channels. This set is trimmed by throwing out any detected peak that is less than 10% the height of the previous detected peak or is shorter than another detected peak at the same location. Using this set of detected peaks a synthetic trace is generated by placing a square wave centered at the location of each detected peak of height 1 and width equal to ¼ the spacing between detected peaks. This means that the heights of the detected peaks play no role in determining the locations of predicted peaks. It is interesting to note that at this stage of the algorithm all information used has to do with peak spacing. Next the entire trace is examined to find the region of most uniform peak spacing based on the detected peaks. A window of 200 trace points (in most traces peak to peak spacing ends up being about 12 trace points near the middle of the trace) is examined around each detected peak. The mean and standard deviation of detected peak spacing is calculated for each window and the region with the lowest mean/standard deviation value is selected as the most uniform region. Once this most regular region of 200 trace points is found the detected peaks in the region are damped by passing the region through a triangular filter with value 1 at the midpoint and 0 at the ends of the region. This weights the detected peaks in the region according to their distance from the center peak. Peaks nearer the center are weighted more heavily than peaks near the edges. A Fourier method is used to fit a sine wave to this damped, synthetic trace region and the location of the peak of this sine wave nearest the middle of the region is designated as a predicted peak. This defines the “first” predicted peak, which will usually be near the middle of the trace. The rest of the predicted peak locations are obtained in an iterative fashion. This works by considering a window defined by shifting the previous region to the right by the period of the sine wave, θ_R selected in the previous iteration. The same procedure as described above is carried out to find the predicted peak at the center of this region except only sine waves with periods $\{\theta_R - 0.03, \theta_R, \theta_R + 0.03\}$ are considered. This results in definition of all predicted peaks from the starting region to the end of the trace. An analogous procedure is used to find predicted peaks between the starting region and beginning of the trace. Once again the algorithm starts with the initial well spaced starting region but this time shifts the region to the left and considers potential sine wave frequencies from the set $\{\theta_R - 0.25, \theta_R - 0.20, \ldots \theta_R + 0.25\}$. Restricting the set of permissible sine wave frequencies in this way means that the average peak-to-peak spacing of the final set of predicted peaks must vary slowly while walking through the trace. This is desirable since this is expected in a well-behaved trace.

Find observed peaks. Observed peaks address the second base calling issue discussed in 2.3.1, that of finding large, well defined peaks. The definition of PHRED observed peaks is much simpler that that of predicted peaks. Observed peaks are the centers of regions of each channel trace that are concave down. An observed peak fitting this criteria is retained if its area (defined as the sum of the trace values in the region) is greater than 10% the average area of the 10 peaks to the left and 5% the area of the adjacent peak to the left. These restrictions eliminate very small observed peaks which probably represent noise. Observed peaks with a large relative area may be split into up to 4 observed peaks in later steps. Notice that definition of the observed peaks takes into account only the shape and size of the peak; no peak spacing criteria are taken into account.

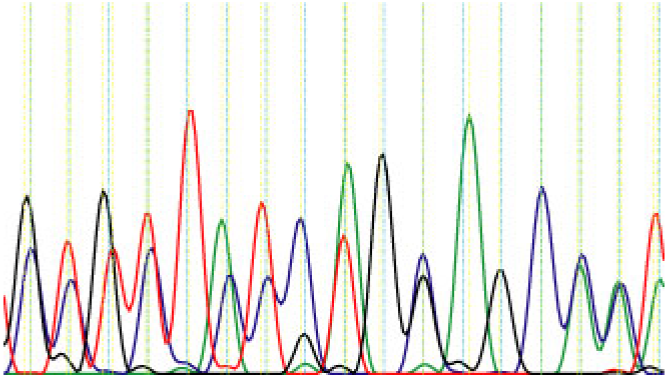


Figure 2-6. Locations of predicted (blue) and observed (yellow) peaks.

Align predicted and observed peaks. Until this point in the PHRED algorithm peak spacing and size considerations have been carefully separated into the definitions of predicted and observed peaks. Figure 2-6 shows the locations of predicted (blue) and observed (yellow) peaks identified by PHRED superimposed on the original trace. At this stage predicted and observed peaks are aligned to create most of the final base calls. Bases are called whenever a predicted peak is aligned to an observed peak. First “fixed peaks” are identified. These are runs of 4 or more predicted peaks that have observed peaks of relative area greater than 0.2 and are within 0.2 trace points of a predicted peak. These predicted peaks are aligned to the corresponding observed peak automatically. This results in long runs of fixed peaks interspersed with gaps in which it is unclear which observed peak goes with each predicted peak. There are several reasons for such gaps: large observed peaks which actually represent a short run of the same base but which are not well resolved on the gel, noise or aberrant migration of fragments in the gel due to dye, template or gel effects. A dynamic programming algorithm is used to align the predicted and observed peaks in these gap regions. If an observed peak has relative area greater than 1.6 several splittings of the peak are tested in the alignment. The observed peak is split up to 4 times into virtual peaks whose positions and areas are distributed evenly across the region of the real peak. Each splitting is tested with the alignment algorithm; the best fit to the predicted peaks in the region is kept. Once this is done the predicted and observed peaks in the region bounded by runs of fixed peaks are aligned globally. The function for the score between an aligned predicted and observed peak is given by:

$$Score = \begin{cases} 0.01 \cdot area \cdot (1 + 0.3 \cdot shift) & \text{if shift} < 0 \\ 0.01 \cdot area \cdot 0.9^{\lfloor shift \rfloor} \cdot \left(1 - 0.1 \cdot \left(shift - \lfloor shift \rfloor\right)\right) & \text{if shift} > 0 \end{cases} \qquad (2\text{-}3)$$

Where area is the area of the observed peak and shift is the distance between the observed and predicted peak. Negative shift means the observed peak precedes the predicted peak. A plot of this function can be seen in the top pane of figure 2-7 assuming an area of 1 since area only linearly scales the score. Extending the axes in the bottom pane of figure 2-7 one can see that the positive shift line is not just straight but slightly curved. Since it appears straight on the domain in which it is used in practice it seems unlikely that this curvature affects the alignments in any meaningful way and is an artifact of the implementation. This alignment matches most predicted peaks to an observed peak. Next an attempt is made to match observed peaks that were not assigned to predicted peaks in earlier steps to predicted peaks. If an observed peak is assigned to a predicted peak that is not already aligned to an observed peak it becomes so assigned and is called. Otherwise the observed peak gets assigned to a predicted/observed peak pair as the “best_uncalled_peak” for the predicted peak. These can be viewed with the –d option, which is used as input to the program polyphred as well as one of the methods that I have developed.

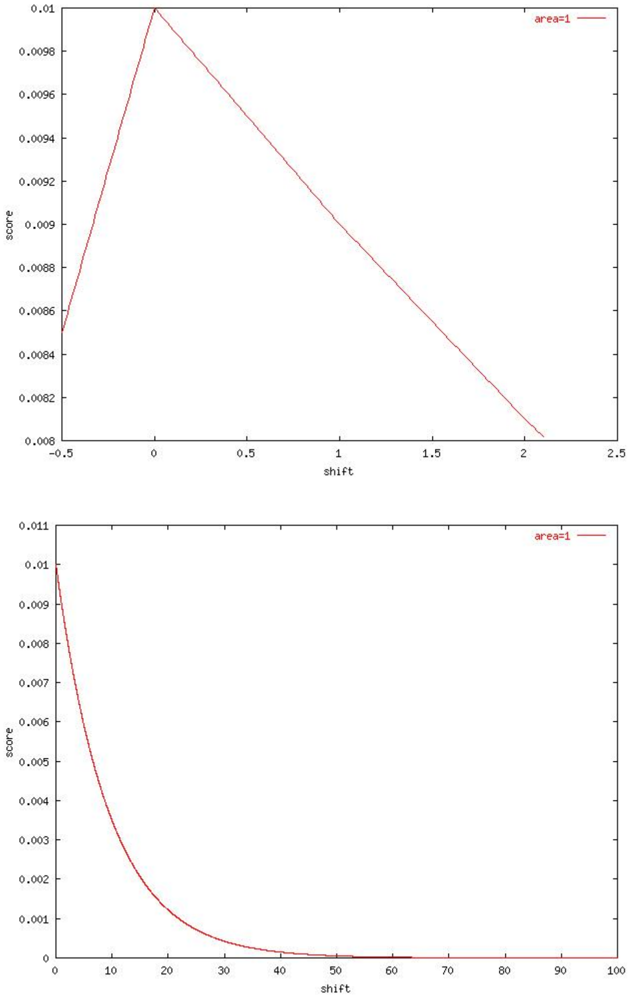


Figure 2-7. Function used by PHRED to score alignments between predicted and observed peaks. Top, range over which function is applied in practice. Bottom, larger range to illustrate non-linearity of positive range of function.

Post-processing. PHRED has a post-processing step that attempts to call peaks that were missed due to “severe compression, extensive noise, or a lane processing aberration” (Ewing et al. 1998). Since the work described here does not attempt to address such situations and modern chemistries and capillary systems greatly lessen these problems this part of the algorithm will not be described in detail.

2.4 Biology

When considering the analysis of traces derived from multiple templates (multi-traces) it is useful to consider some of the biological and biotechnological mechanisms responsible for the creation such traces.

2.4.1 alternate splicing

The central dogma. Any proper discussion of eukaryotic gene expression must begin with a review of the central dogma. The central dogma says that the flow of genetic information proceeds from DNA to RNA to protein. An RNA copy of a gene including exons, introns and untranslated regions is made from genomic DNA through the process of transcription producing a pre-mRNA or transcript. During transcription non-coding introns are removed from the transcript by splicing resulting in the mature mRNA. Finally the mRNA is exported from the nucleus where it is processed by ribosomes that generate one or more proteins based on the mRNA sequence.

Alternate splicing types. Alternate splicing refers to the phenomena that the splice sites of many transcripts are not unique *in vivo*. Individual transcripts often give rise to several mature mRNA molecules (isoforms) and subsequent proteins. There are several modes of alternate splicing as shown in figure 2-8. Among these are the inclusion or exclusion of entire exons as well as single alternate splice sites and retained introns. A common technique for obtaining the sequences of mRNAs is reverse transcriptase PCR (RT-PCR). In this PCR method the mRNA is reverse transcribed to a complementary DNA (cDNA) by the action of the enzyme reverse transcriptase. The cDNA is subsequently sequenced by standard methods as described in 2.1. If only one form of the mRNA is present a clean trace is obtained. However if multiple isofoms are present multiple cDNA templates are sequenced resulting in a multi-trace.

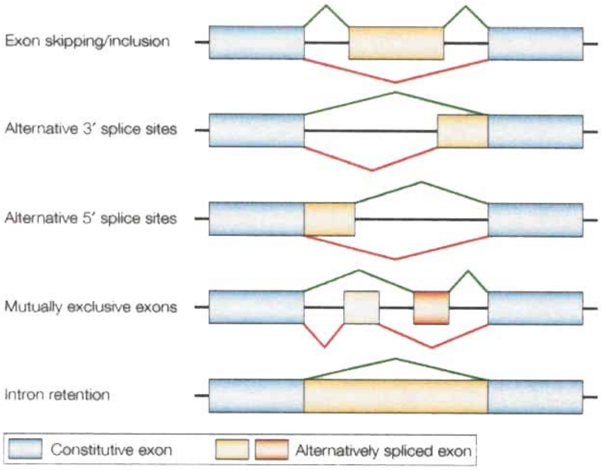

Exon skipping/inclusion
Alternative 3′ splice sites
Alternative 5′ splice sites
Mutually exclusive exons
Intron retention
Constitutive exon
Alternatively spliced exon

Figure 2-8. Different types of alternate splces (Cartegni et al. 2002).

Prevalence of alternate splicing. Alternate splicing appears to be very prevalent in the genomes of human and other higher eukaryotes. Published reports indicate that between half (Mordrek et. al. 2001) and three quarters (Garcia-Blanco et. al. 2003) of genes in the human genome may be alternately spliced. The phenomena of alternate splicing can greatly increase the proteome complexity of a genome with a modest number of gene loci. Detection and characterization of alternate spliced transcripts is an interesting and important problem in high-throughput biology today.

2.4.2 Indel polymorphisms

Sequencing of polymorphic loci provides another opportunity to generate multi-traces. If a segment of DNA is sequenced from an individual who is heterozygous for a polymorphism some type of double trace is generated depending on the exact nature of the polymorphism. If the polymorphism is a SNP in which a base at one position is substituted for another a double peak is generated only at the polymorphic site but the rest of the trace is unaffected. This is an important problem and has been addressed by many programs including polyphred. Polymorphisms in which the two polymorphic forms differ in length (indel polymorphisms) are more problematic. Sequencing of an individual heterozygous for an indel polymorphism results in a double trace in which one train of peaks is offset from the other by the difference in length between the two forms. According to representatives of ABI (personal communication) the ABI KBbasecaller can deconvolve such traces if the indel is less than 10 base pairs. There are cases, however, such as polymorphisms arising from short tandem repeats in which the difference in length between the two forms can be much longer than this.

2.4.3 T-DNA mediated knockout libraries

Another interesting procedure in which multi-traces are encountered is the creation of T-DNA mediated knockout libraries. A knockout library is a collection of of cell lines or organisms each of which has one or more genes inactivated. The function of a gene can be inferred by studying the loss of function phenotype of an organism in the library that lacks the gene. Knockout libraries have been created for several organisms including *S. cerevisiae*, *C. elegans* and *A. thaliana*. There is not one universal method used to create knockout libraries in every organism. The *S. cerevisiae* knockout library is based on that organism's efficient system of homologous recombination, in *C. elegans* RNAi is used. An *A. thaliana* knockout library using *Agrobacterium* T-DNA has recently been generated (Alonso et al 2003). In nature the bacteria *Agrobacterium tumefaciens* possessing the Ti (tumor inducing) plasmid can infect plants causing crown root disease. Infection occurs because a segment of the Ti plasmid carrying the disease genes is capable of integrating itself randomly into the plant genome. If the disease genes are removed and replaced with a selectable marker integration into the genome still occurs but recombinant plants can be identified. If the T-DNA segment integrates at the locus of a gene the function of that gene is disrupted. For the knockout library to be useful, though, it is necessary to identify the exact location of the integration to know which gene has been knocked out. This is accomplished by adapter ligation PCR (Siebert et. al. 1995). This method begins by digesting the whole genome containing inserted T-DNA segments with a blunt cutting restriction enzyme such as EcoRV. Adapters are next non-specifically ligated to all restriction fragments. The fragment containing the inserted T-

DNA segment now contains two known sequences (the T-DNA and adaptor sequences) flanking an unknown genomic sequence at the insertion site. PCR primers are designed to amplify the region between these known sequences and the PCR product is sequenced. This produces a short tag sequence that can be used to determine the genomic location of the T-DNA insertion site. If there is only one insertion site in the plant the sequence is unambiguous. In this case it is a simple matter to recover the insertion site from the PCR product sequence. If, however, there is more than one insertion event a multi trace is generated in the sequencing step.

2.4.4 Mispriming

Misprimimg can also be a source of double traces. This refers to the amplification of more than one PCR product from one set of primers, usually unintentionally. The result is a mixture of template DNA that, if sequenced, will generate a double trace. Unintentional mispriming can occur for several reasons. If the goal is to amplify all or part of a gene from genomic DNA or a pool of mRNA and the gene of interest is part of a gene family other family members could be amplified by the same primer set. Normally it is possible to check for mispriming if the whole genome sequence is available. If this is not the case unintentional mispriming is more common.

Chapter 3

Trace Recalling Method

Three methods were developed to base call the DNA template sequences represented in a double trace. All rely on the existence of an assembled version of the genomic sequence from which the double trace was derived and to which it can be aligned. We refer to the first method implemented as trace recalling. This chapter will describe the trace recalling method. The next method, integrated alignment will be described in chapter 4. Finally a pair-HMM based approach will be described in chapter 5.

3.1 Early implementation

The trace recalling method was developed in two stages. The early implementation differed significantly from the later refinements. Both are presented here as well as a running example of the methods applied to the short segment of double trace shown in figure 3-1.

3.1.1 Mixed sequence

The idea behind trace recalling, as the name implies, is to make a first base calling of the trace and then recall the trace to get the sequence of the second template. Standard base callers such as PHRED were not designed to call double traces. PHRED identifies the predicted and observed peaks and calls bases by aligning them as described in chapter 2. Since many of the predicted peaks lie near two observed peaks in a double trace PHRED usually aligns the observed peak that is larger or nearer the predicted peak. The other peak is considered noise. The base that this larger or closer peak represents then is the base that is ultimately called. If one template is much more abundant than the other in the sequencing reaction this means that the called sequence will almost completely match the more abundant template. If, however, the template abundances are nearly equal the called sequence will appear to randomly switch between the two template sequences. This is due to dye and gel effects, which perturb the size and spacing of the peaks. We call such a called sequence a mixed sequence since it represents a mixture between the sequences of the two templates. A key characteristic of mixed sequences is that the bases are called in the correct relative order but randomly from the two templates. It is as if the mixed sequence were generated by laying the two template sequences on top of each other and at each position randomly picking a base from one template or the other as the next base in the mixed sequence. Mixed sequences also have terrible quality values, since the uncalled peak from the second template is interpreted as a noise peak.

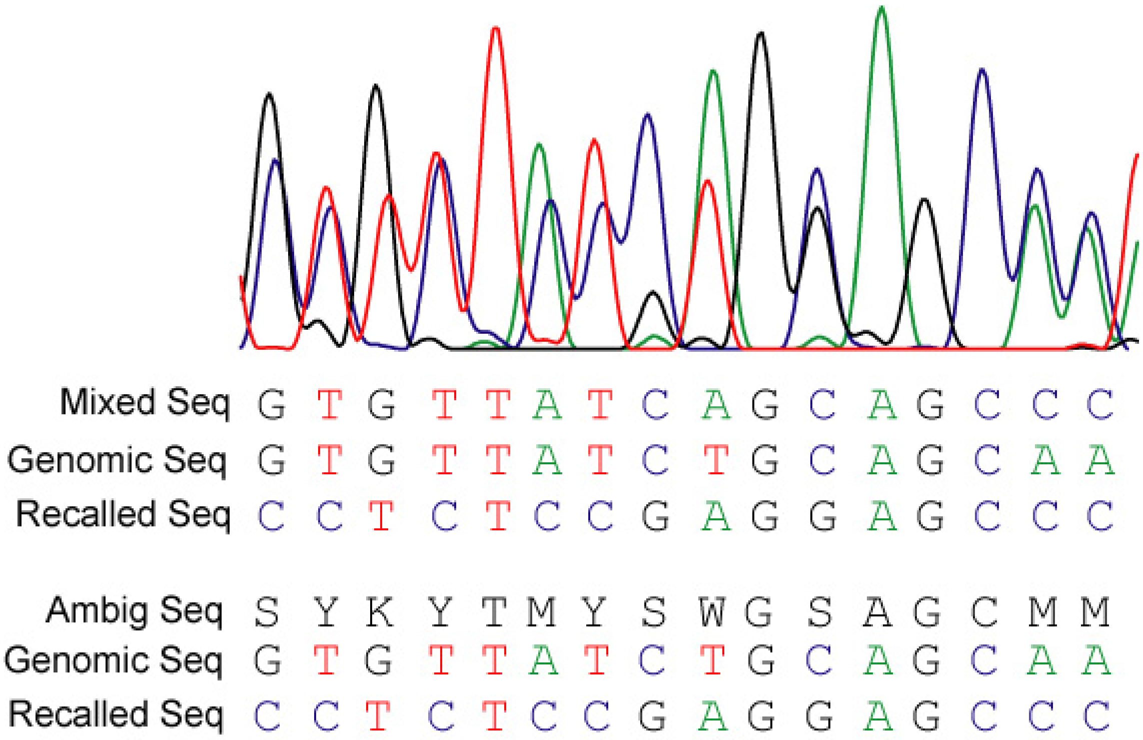


Figure 3-1. A double trace (top), example of original trace recalling method (middle) and trace recalling with ambiguity sequence (bottom).

3.1.2 Alignment

The properties of mixed sequences suggest the first step in the trace recalling method. Alignment of a mixed sequence to a genomic sequence containing one or both template sequences should yield a match between 62.5 and 100 percent identity. The closer the abundances of the two templates the closer to 62.5 percent identity the match should be.

The expected percent identity would not drop to 50 percent since by chance sometimes the same base occurs at the same position in both templates. In random sequence this would happen in 1 out of 4 positions. These positions align as matches with the remaining positions having a 50/50 chance of matching. This means that the lower bound of the alignment would be expected to be 100 * (1/2 + 1/2 x 1/4) or 62.5%. Thus to find sequence of one template represented in the double trace the mixed sequence is aligned to the genomic sequence.

At first BLAST was used for the alignment step. This works if the trace is known to derive from unspliced sequence. The main application of this technique though, is finding alternate splice forms from RT-PCR experiments which are spliced sequences. Thus we decided to use a spliced alignment program for this step, EST_GENOME (Mott 1997). If the alignment is correct, it immediately identifies the sequence of one of the templates. It is the genome sequence to which the mixed sequence aligned. The best alignment represents either the more abundant or longer DNA template sequence so we refer to it as the dominant sequence. The alignment between the mixed and genomic sequences is illustrated in the running example of 3-1 as the first two lines in the middle pane.

3.1.3 Dump PHRED state

Since the PHRED source code was available we were able to modify it to dump out relevant portions of its internal state. This includes the positions of all predicted peaks; the positions, sizes and channels of all observed peaks as well as the predicted peaks corresponding to all called peaks. Recall that PHRED defines an observed peak as a local maxima or midpoint between maxima with some size limitations. This means that sufficiently tall secondary peaks, even if they aren't called, are tracked by PHRED. A subset of these peaks represent the sequence of the secondary template.

3.1.4 Recalling

The recalling step uses the initial alignment described in 3.1.2 and PHRED's internal state to identify the sequence of the non-dominant or secondary template. First, for each predicted peak in the trace a list of nearby observed peaks is compiled. Typical peak-to-peak separation in a good trace is 12 trace points (each trace point represents one pass of the laser over the gel or capillary tubes) so an observed peak is considered near a predicted peak if it is within 5 trace points of the predicted peak. For example in the example in figure 3-1 the observed peak list at the first position is {G,C} and at the second position is {T, G, C}. Next the predicted peak corresponding to each called base is determined from PHRED's internal state. Each position in the called trace/alignment is considered in turn to determine the secondary sequence. The observed peak list corresponding to the predicted peak of each base in the called sequence is compared to the genomic base to which it aligned. In a double trace we expect exactly two observed peaks near each predicted peak, one matching the aligned genomic base and another from the secondary template sequence. If this is the case the recalled sequence at this position is the observed peak not matching the genomic base. The reasoning is that the genomic base explains one peak so the other peak must have come from the secondary template. If a peak matching the genomic base is present but there is more than one other observed peak near the predicted peak the largest peak (in terms of area) not matching the genomic base is called in the secondary template sequence. The smaller peaks not matching the genomic base are most likely noise in the trace. In case the only nearby peak is the observed peak matching the genomic base the same base is recalled in the secondary template sequence. In this case it could be that the same base is present in both templates at this position. If there is no observed peak matching the genomic base at this location or if a gap in the genome is aligned to a letter in the trace, an ‘N’ is called in the secondary template sequence. This is done because, while there is clearly a base at this location, we are not sure what it should be. The last possibility is that a gap in the trace is aligned to a base in the genome. In this case nothing is called since this might mean there is a difference between the sequence of the template and genomic sequence such that a base is actually missing from the template sequence. A flowchart of the decision process for recalling a single base as implemented is presented in figure 3-2. Recalling of a short segment of double trace can be seen more clearly in the middle pane of figure 3-1.

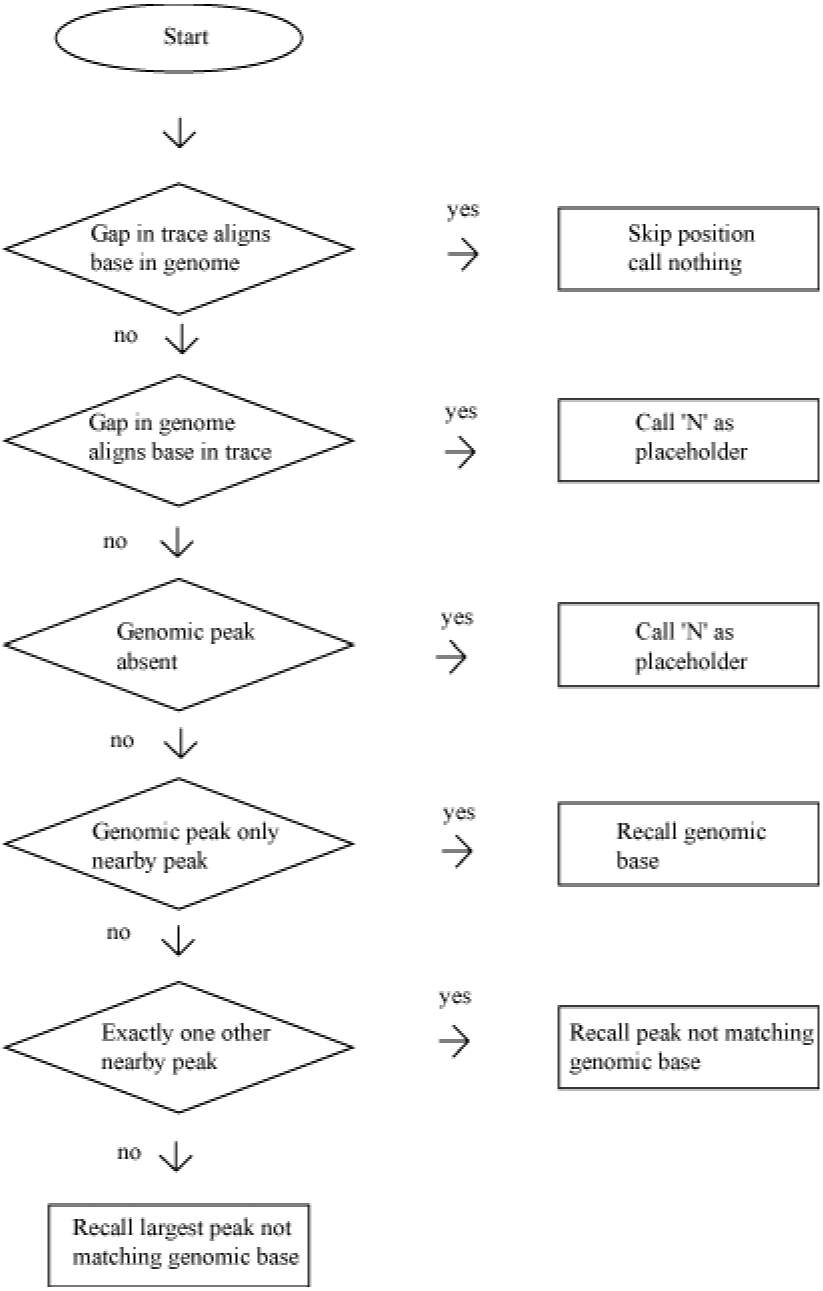

Start
Gap in trace aligns base in genome
yes
Skip position call nothing
no
Gap in genome aligns base in trace
yes
Call 'N' as placeholder
no
Genomic peak absent
yes
Call 'N' as placeholder
no
Genomic peak only nearby peak
yes
Recall genomic base
no
Exactly one other nearby peak
yes
Recall peak not matching genomic base
no
Recall largest peak not matching genomic base

Figure 3-2. Flowchart describing recalling of single base

3.1.5 Realignment of secondary sequence

Once the secondary sequence is recalled as described above it is aligned to the genomic sequence. The success of the procedure can be evaluated at this point. If the secondary trace actually represents a second DNA template the secondary sequence should align well somewhere else in the genome. If the trace is simply noisy the secondary sequence should be essentially random sequence and not align well.

3.2 Later refinements

Since the initial implementation of trace recalling there have been two major refinements of the core procedure.

3.2.1 PHRED –d option

Trace recalling can be greatly simplified by using the –d option to PHRED. With this option PHRED outputs the two best bases at each position. The second best base represents the second best observed peak for each predicted peak. If the trace was derived from exactly two templates this is ideal since probable noise peaks are automatically filtered out. Knowledge of the best and second best observed peak at each position allows one to replace the mixed sequence with an ambiguity sequence. That is a sequence of DNA ambiguity codes which represent at each position, both bases present. This leads directly to the next refinement.

3.2.2 Ambiguity alignment

Once the ambiguity sequence is available the natural next step is to align this sequence directly to the genome instead of the mixed sequence. The code, which determines matches in EST_GENOME, was modified to recognize ambiguity matches as well as single base matches. Similarly BLAST matrices were created which recognize ambiguity matches. This was a major improvement since often the alignment of the mixed sequence to the genome was incorrect near splice boundaries. When aligning ambiguity sequence, however, one expects a near perfect alignment since at each position both template bases are represented. Aligning ambiguity sequences to large portions of the genome, however, can cause problems. Since the ambiguity sequence contains less information than a DNA sequence of equal length (each position has 1 bit instead of 2) spurious alignments are more likely. Unlike the usual application of finding homologies by alignment short gaps are not to be expected when aligning ambiguity sequences derived from double traces back to the genome. Gap penalties thus can be increased to reduce spurious alignments. Mismatches should also be very unlikely so those penalties can be increased which reduces spurious alignments further. Increasing gap and mismatch penalties greatly reduce spurious alignments. An example of trace recalling using ambiguity sequence can be seen in the bottom pane of figure 3-1.

3.3 Identification of alternate splices

As stated previously, the main application of trace recalling is to elucidate alternate splice forms by RT-PCR sequencing. Once the trace recalling procedure has been applied to a trace it is desirable to have an automated system for determining if an alternate splice has been detected and if so to classify it. Initially this was done by manually viewing the initial and secondary alignments of traces to the genome. An example of such an alignment as viewed in ACEDB is shown in figure 3-3 (blue and red genes only). Automation of this task is complicated by the fact that alignments near the edges of the trace are unreliable. To solve this problem a string summarizing each pair of alignments with one character for each position of the genomic sequence is generated. The characters at each position represent the number of alignments covering that position. For example if both alignments contained a certain base the character in the string representing that base would be a 2. If only one and not the other alignment contained that base a 1 would represent it. This is shown in Figure 3-3. Matching regular expressions to this string can recognize specific kinds of alternate splice forms. For example an alternate exon can be represented by $2^+1^*0^+1^+0^+1^*2^+$. This regular expression specifies that between two common regions in the alignment there is an exon in one alignment (the middle 1^+) of non-zero length flanked by regions that do not align (the 0^+s) with some slop possible around the edges of the common regions (the 1*s). Similar regular expressions have been used to find mutually exclusive exons, alternate 3’ and 5’ splice sites and retained introns. Results of applying trace recalling to several data sets will be presented in chapter 6.

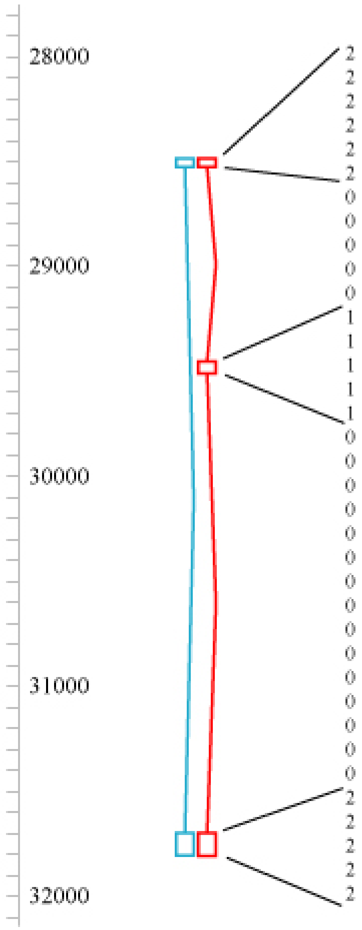


Figure 3-3. Example alignment of original (red) and recalled (blue) sequences to genome and indicator string used to classify alternate splice type.

Chapter 4

Integrated Alignment Method

The trace recalling method works by the sequential application of several modified off the shelf applications. This consists of an initial alignment followed by a recalling step and finally a second alignment to verify the recalled sequence. It is a heuristic, which works very well in practice. It is nonetheless a heuristic. Trace recalling works very well despite throwing away much information. This included the size and exact location of observed peaks relative to predicted peaks. It also excluded all but two potential observed peaks for each predicted peak. To incorporate these additional lines of evidence we developed a new approach called the integrated alignment method.

4.1 Theory

The integrated alignment method relies on the fact that the core of every step in the trace recalling method uses a Smith-Waterman type dynamic programming alignment. Clearly this is true of the initial and secondary alignments. Recall from section 2.3.3 also that the base calling step of PHRED is actually a dynamic programming alignment between the predicted and observed peaks. As a result base calling and both alignment steps can be combined into one algorithm. The pairwise alignments between the predicted and observed peaks, the ambiguity sequence and genome, and the recalled sequence and genome are replaced with one multiple alignment between the predicted peaks, the observed peaks and the genome. Thus the alignment steps are integrated into the base caller.

Initialization of the alignment matrix is handled like the initialization of a Smith-Waterman pairwise alignment matrix. The boundary values (planes in this case) are set to zero. A local alignment model is desirable because knowledge of the location of the template DNA sequence in the genomic sequence is not assumed. Thus free end gaps are required. Equation 4-1 shows the dynamic programming recurrence used to fill in the multiple alignment matrix. The i indices refer to predicted peaks, j indices refer to observed peaks and k indices refers to genomic sequence. $Score_{Phred}(i, j)$ refers to the score returned by the PHRED scoring function for aligning predicted peak i to observed peak j. The gap penalties and match bonus are arbitrary constants.

$$S[i][j][k] = \max \begin{cases} S[i-1][j-1][k-1] + Score_{Phred}(i, j) + bonus(j,k) \\ S[i-1][j-1][k] + Score_{Phred}(i, j) - gap_1 \\ S[i][j-1][k-1] - gap_2 \\ S[i][j-1][k] - gap_3 \\ S[i-1][j][k-1] - gap_4 \\ S[i-1][j][k] - gap_5 \\ S[i][j][k-1] - gap_6 \\ 0 \end{cases} \tag{4-1}$$

Where,

$$bonus(j,k) = \begin{cases} \text{match bous} & \text{if genomic base k} = \text{channel of observed peak j} \\ \text{- match bonus} & \text{otherwise} \end{cases} \tag{4-2}$$

The cases which align a predicted and observed peak (and thus call a base) use a score similar to the PHRED score for aligning predicted and observed peaks plus an extra constant factor. This term reflects whether or not the channel of the observed peak matches the type of genomic base to which it is aligned. If they match a bonus is added, otherwise a penalty is subtracted. In the cases where a predicted and an observed peak are not aligned the score from the originating cell is simply propagated less an arbitrary but constant penalty. This scoring scheme is meant to pull out an alignment that calls bases that match the genome while localizing sections of the trace to specific genomic regions. An example of the type of alignment obtained from aligning a double trace to the genome with this method is displayed in figure 4-1. In this figure o_i stand for observed peaks, p_j stand for predicted peaks and g_k stand for genomic bases.

$$\begin{array}{ccccccccccccc} - & g_1 & - & g_2 & - & g_3 & - & g_4 & - & g_5 & - & g_6 & - \\ o_1 & o_2 & o_3 & o_4 & o_5 & o_6 & o_7 & o_8 & o_9 & o_{10} & o_{11} & o_{12} & o_{13} \\ - & p_1 & - & p_2 & - & p_3 & - & p_4 & - & p_5 & - & p_6 & - \end{array}$$

Figure 4-1. A hypothetical multiple alignment as described in text.

The dynamic programming alignment is applied to the trace and genomic sequence twice. First, as described above, yielding the stronger of the two templates. On the second application a sub-optimal alignment is found by the Waterman-Eggert method as described in section 2.2.6. This gives an alignment between the genome and the portion of the secondary template sequence, which is represented by double trace in the trace. Single trace regions of the trace are not found again by this method since all cells corresponding to single trace segments of the dynamic programming matrix on the optimal path have been zeroed out. To identify this common region the traceback step is modified in the sub-optimal alignment. Traceback proceeds until a zero score is found. If this is due to contact with the previous optimal path, the traceback jumps to the old optimal traceback and proceeds. Continuing the traceback from this intersection with the optimal path at the predicted peak immediately preceding the last predicted peak on the sub-optimal path recovers the single trace segment.

4.2 Complexity/Optimization Issues

A simple calculation shows that filling in the full dynamic programming matrix as described in section 4.1 is not practical. Assume a normal trace contains about 1000 predicted peaks. If this is a double trace it could easily contain 2000 observed peaks not counting noise peaks. This means that one plane through the full matrix requires 2 million cells. If we assume a tight implementation in which each score is represented by a float (4 bytes) and each traceback pointer is represented by a char (1 byte), each such plane takes 10 megabytes of memory. There is one plane for each base in the genomic sequence. Assume we have access to a machine with 4 gigabytes of memory, the length of the genomic sequence that we can align such a trace to is only 400 base pairs, less than the length of the trace! As described so far the problem is $O(l_{pred} l_{obs} l_{genome})$ in both time and memory where the these lengths represent the number of predicted peaks, observed peaks and genomic bases. Clearly, major optimizations are needed.

4.2.1 Banding

The first optimization is a banding scheme. As described above every predicted peak is compared to every observed peak, which is not necessary. A correct alignment should not align a predicted and observed peak more than a few trace points away from each other. Recall that the typical separation between peaks is about 12 trace points.   A processing step is used which identifies observed peaks within a fixed window of each predicted peak. This defines a variable width band through the dynamic programming matrix that is not necessarily constrained by the diagonal. The only matrix cells filled in correspond to potential alignments between predicted and observed peaks within this band and therefore close enough that they could reasonably align. Consider again the hypothetical trace presented earlier with 1000 predicted and 2000 observed peaks. Say the trace is well behaved and a band width of 40 trace points is used. Now only about 7 observed peaks are considered for each predicted peak. The number of cells required to align each genomic base drops from 2 million to 7000. On the same 4 gigabyte machine the trace can now be aligned to over 114 kilobases of genomic sequence with a corresponding drop in runtime. Nothing is sacrificed from this optimization since the cells not examined could not represent real base calls. Figure 4-2 graphically illustrates the memory and time savings of the banding scheme. In practice the number of observed peaks within a fixed with band is bounded. Thus, banding effectively reduces the time and memory complexity of the problem to $O(l_{pred}l_{genome})$.

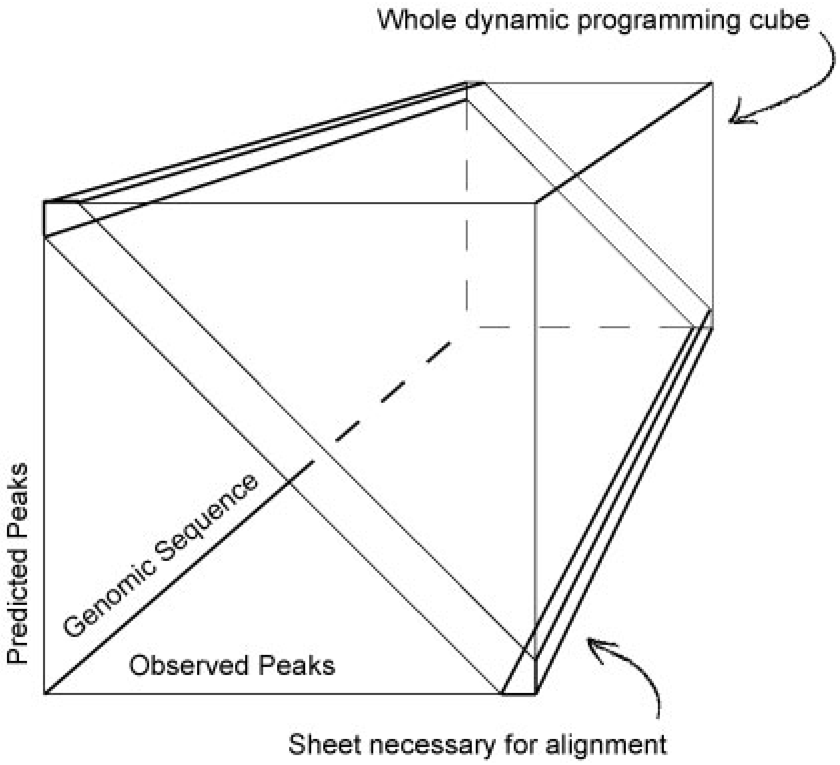


Figure 4-2. Illustration of memory savings from banding.

4.2.2 Linear space alignment

Another optimization is possible to make the algorithm run in linear memory. The Myers and Miller linear space pairwise alignment algorithm was discussed in section 2.2.4. The basis of the algorithm is that the optimal pairwise alignment traceback must pass through the middle row of the dynamic programming matrix. Thus the middle row can be computed two ways, from the top of the matrix down and from the bottom of the matrix up. Corresponding cells are summed and the cell with the largest sum indicates where the optimal traceback path crosses the middle row. Since only scores are computed and the score in each cell is based on only the scores of cells in the current and previous rows only 2 rows are required to be in memory at any time. Analogously in the multiple alignment case, the optimal alignment must pass through the middle plane of the dynamic programming matrix. Thus one can compute the middle plane in both directions and sum the corresponding cell scores to determine where the optimal alignment crosses that plane. This breaks the problem into two sub-problems that can be solved in the same way. The score of each cell depends only on scores of cells in the current and previous plane so only 2 planes are required to be in memory at any time. Returning to our hypothetical trace the amount of memory required is $2*7*5*l_{genome}$ where l_{genome} is the length of the genomic sequence. This means on a 4 gigabyte machine the trace can be aligned to 57 megabases. Linear space alignment was not implemented in the final algorithm due to other limitations of the integrated alignment approach.

4.3 Implementation Details

During the course of implementation and testing of the integrated alignment method several departures from the theory described above were necessary.

4.3.1 Scoring alignments between predicted and observed peaks

The original idea for scoring alignments between predicted and observed peaks was simply to use the scoring function that PHRED uses (eqn 2-3) and modulate the score depending on whether or not the observed peak matches the genomic base (see eqn 4-1). This was not possible since the PHRED alignment score depends directly on the absolute area of the observed peak, making it unbounded. This is a problem because it is necessary to balance the scores obtained from aligning the predicted and observed peaks in the base calling part of the algorithm against mismatch and gap penalties in the sequence alignment part. Having unbounded scores in the alignment between predicted and observed peaks is analogous to using a BLAST matrix in which match scores are unbounded. In this situation it is impossible to come up with reasonable mismatch and gap penalties that can pull out the desired alignments.

The solution to this problem was to design a simple, bounded scoring system for aligning predicted and observed peaks. The basic criteria for the scoring system was that it should give high scores to large observed peaks near predicted peaks and lower scores as the observed peak got smaller and farther away from the predicted peak. In accord with this criterion a two part score, separately considering shift and area, was used. The shift part considered the distance between the predicted and observed peak in a similar way to the PHRED score. If the predicted and observed peak coincided exactly the shift score got a fixed maximum value (20 was used in the final implementation). The shift score decreased linearly with the distance between the predicted and observed peak until it reached zero at a distance of 6 trace points. Six trace points was used since on average peaks in real traces are 12 trace points apart. When the shift exceeds 6 trace points the observed peak starts getting closer to another predicted peak. The area part considers the relative areas of observed peaks in the vicinity of the predicted peak. This works by computing a scaled area for each observed peak in a 6 trace point window around the predicted peak. The scaled area is simply the area of the observed peak divided by the area of the largest observed peak in the window. The area part of the score is obtained by multiplying the scaled area by a fixed constant (again 20 was used as this constant in the final implementation). The alignment score for the predicted and observed peak is simply the sum of the shift and area scores. Changing the constants used in computing the alignment score can control the relative weights of position vs. area, however, equal weighting seems to work well in practice. This solves the bounded score problem since the highest alignment score possible for any predicted/observed peak pair is the sum of the constants (40 in the final implementation).

4.3.2 Using only the top 3 scoring observed peaks

Originally all observed peaks within the 6 trace point window around each predicted peak were considered. In well-behaved regions of a double trace this works well since only 2 or 3 observed peaks could get significant scores for any given predicted peak. However, in very noisy regions of the trace or in traces that are all noise often there are regions of the trace in which every predicted peak is within 6 trace points of observed peaks in each channel. To compound the problem in these noise regions the observed peaks often have similar areas resulting in scores that seem significant. This is a problem since proper functioning of the integrated alignment method relies on the sequence represented in the trace to match some subsequence of the genomic sequence. When viewed from a sequence alignment perspective the situation in noisy traces is similar to aligning a sequence of Ns to a genomic sequence using a matrix that gives positive scores to alignment between N and any base. The noise region can align anywhere in the genomic sequence with a score high enough to swamp out any real signal present.

This problem was solved by a pre-processing step in which alignment scores for all predicted/observed peak pairs are computed. Only the 3 highest scoring observed peaks for each predicted peak are retained and used in subsequent steps. It is still likely that the 2 desired observed peaks per predicted peak are retained. In noisy regions of the trace, however, nuisance alignments to noisy regions are broken up since several mismatches and gaps in such alignments become necessary. This usually ensures that they score poorly enough to not show up at all. Empirically this had the effect of all but eliminating alignments to noisy traces while not impacting performance on real double traces.

4.3.3 Choice of mismatch and gap penalties

The mismatch and gap penalties in eqn 4-1 were chosen based on the composition of alignments desired. As can be seen in eqn 4-1, alignments are determined by 10 parameters in addition to the alignment score between the predicted and observed peaks. These include the match bonus (type1 match), mismatch penalty (type1 mismatch), 6 types of single base gaps (type2, type3, type4, type5, type6 & type7) and 2 intron penalties, one for canonical GT/AG introns (type8 splice) and one for non-canonical introns (type8 intron). To understand the rationale behind these alignment parameters it is first necessary to understand what each alignment type means in the context of a full alignment. The type1 and type8 alignments are relatively straightforward. Type1 refers to the case where a predicted peak, observed peak and genomic base are all aligned. The match or mismatch depends on whether or not the observed peak channel matches or mismatches the genomic base. The type8 alignment refers to the case of a long range jump in the traceback when an intron is encountered in the genomic sequence. The penalty for such a jump is smaller when the canonical GT/AG splice signal is present at the ends of the intron (type8 splice) and larger otherwise (type8 intron). The other alignment types refer to the rest of the possible alignments. For example a type2 alignment means that a predicted and observed peak are aligned to each other but to a gap in the genomic sequence. This can be represented graphically as –po. The graphical representations of the other alignment types are type3 (g-p), type4 (--p), type5 (go-), type6 (-o-) and type7 (g--). Three of these gap types (type3, type4 & type5) are nonsensical and so are given scores of negative infinity. Consider, for example, the type3 alignment. It makes no sense to align a genomic base and predicted peak to a gap in the observed peak sequence so it is excluded. Similar arguments can be made for type4 & type5 alignments. The remaining alignments, however, do have reasonable interpretations. A type2 alignment corresponds to an insertion in the sequenced template or a deletion in the genomic sequence. Similarly a type7 alignment corresponds to a deletion in the sequenced template or insertion in the genomic sequence. The type6 alignment at first appears to be another nonsensical case but it turns out to be important. Type6 alignments are used to “skip over” observed peaks which could be either noise peaks or peaks corresponding to the other sequenced template. To understand this, consider a perfect double trace in which for each predicted peak there are exactly 2 observed peaks corresponding to different subsequences of the genomic sequence. Both the optimal and sub-optimal alignments will be composed of alternating type1 match and type6 alignments. This is because for each type1 match alignment the observed peak corresponding to the other trace at that predicted peak location must be “skipped over” with a type6 alignment.

Based on the considerations above, penalties for the different types of mismatches and gaps can be set based on the kinds of alignments expected. The basic score unit is the average score of a properly aligned predicted and observed peak. As implemented, this turns out to be about 35 for the optimal and 25 for the sub-optimal alignment. This difference in average scores makes sense because the optimal alignment, by design, pulls out the stronger of the two signals present in the trace. The match bonus is set to zero. Once this is done the mismatch and other gap scores can be set based on the desired composition of the alignments. The nominal parameter set used in the experiments is given in table 4-1.

Table 4-1. Parameters used for integrated alignment.

Penalty type	**Value**
Type1 match	0
Type1 mismatch	6 * average peak alignment score
Type2	6 * average peak alignment score
Type3	Infinity
Type4	Infinity
Type5	Infinity
Type6	0.3 * average peak alignment score
Type7	6 * average peak alignment score
Type8 splice	8 * average peak alignment score
Type8 intron	24 * average peak alignment score

These scores can be understood as “in a good alignment how many type X alignments will be tolerated per properly aligned type1 match before the alignment is terminated or excluded”. For example the type2 penalty says that for each insertion in the trace or deletion in the genomic sequence there must be at least 6 type1 match alignments, otherwise the score for the segment becomes negative and the alignment is excluded. This is similar to the way in which match and mismatch scores are chosen in a BLAST matrix to pull out alignments with specific desired percent identities. These parameters work well for many but not all traces which will be discussed later in the results.

Chapter 5

Pair-HMM Method

An alignment algorithm based on pair-HMMs was developed to address some of the shortcomings of the integrated alignment approach.

5.1 Alignment Sequences

As in the integrated alignment approach we wish to align the sequencing trace to an assembled genomic sequence. Instead of a three-way alignment of predicted peaks, observed peaks and genomic sequence the pair-HMM developed here aligns the observed peaks directly to the genomic sequence. We still wish to include information concerning the regularity of spacing of the observed peaks and their size in the scoring function. This is accomplished by appending two values to each observed peak in addition to its channel. The first value is the distance from the observed peak to the nearest predicted peak in units of trace points. This value can be positive or negative depending on the relative order of the predicted and observed peak. The second value is a relative area of the observed peak obtained by dividing it's area by the average area of the ten neighboring observed peaks. Thus the sequence aligned to the genomic sequence by the pair-HMM is actually a sequence of vectors in which each observed peak is defined by a channel (A, C, G, or T) a distance, and a relative area. The last two attributes of the observed peak sequence are real valued.

5.2 Model Topology

Several model topologies were used, the largest of which is depicted in figure 5-1.

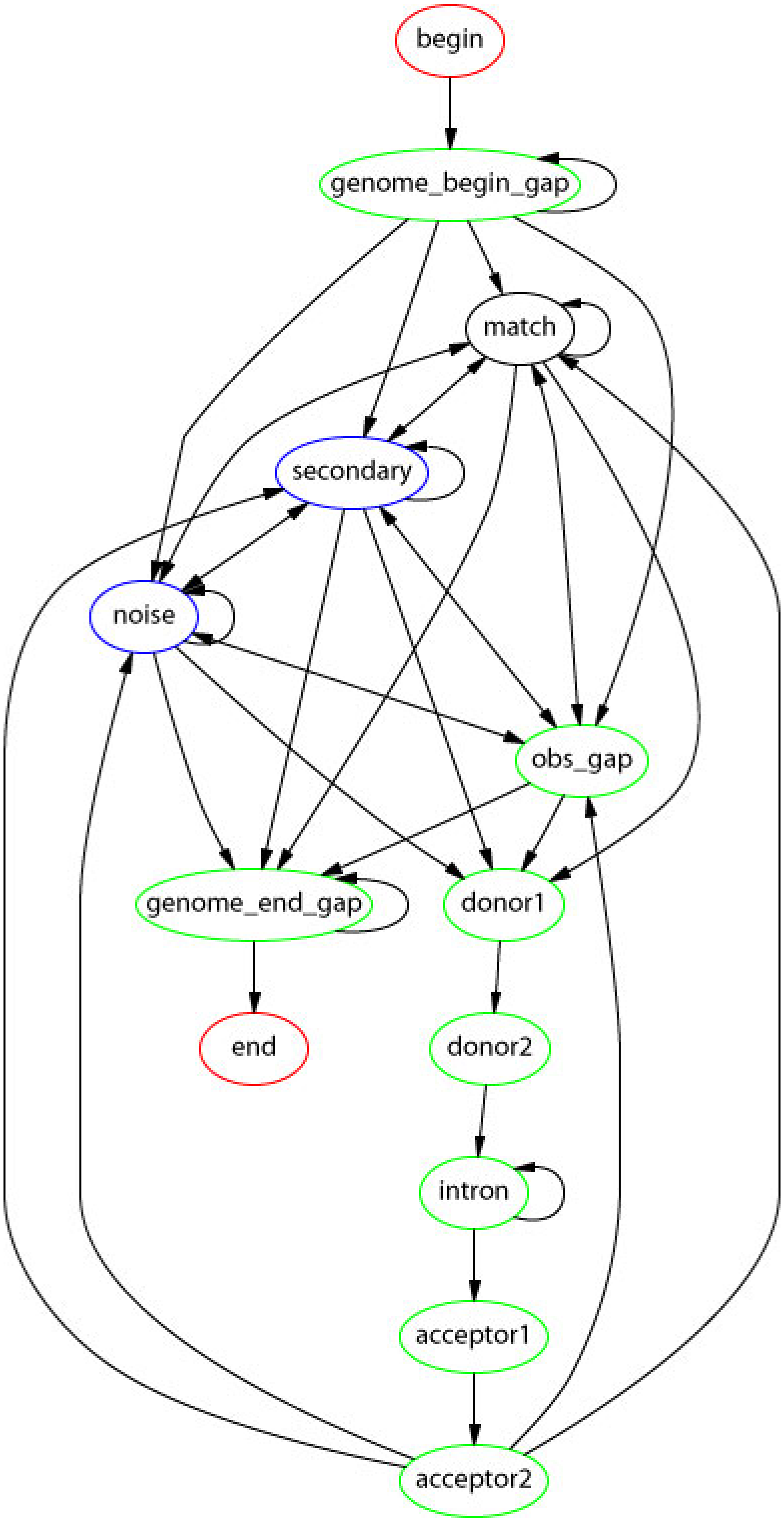


Figure 5-1. Pair-HMM model topology for aligning a trace to genomic sequence.

This model allows a spliced trace to align locally to a genomic sequence while enforcing consensus GT/AG splice signals in the genomic sequence. Models with subsets of these states were tested which exclude the states used to model introns as well as the state which allows a gap in the observed peak sequence. The states are interpreted as follows. Emission from the match state signifies that an observed peak aligns to a genomic base. The channel of the observed peak must match the type of the genomic base. The secondary and noise states represent observed peaks aligned to gaps in the genomic sequence. They differ in that secondary peaks are assumed to align elsewhere in the genome while noise peaks are not assumed to align anywhere. Secondary and match states emit observed peaks from the same probability function (tied parameters) while noise peaks emit from a different distribution. This will be described in more detail in 5.3. The obs_gap state models a genomic base aligning to a gap in the observed peak sequence. This should be a rare occurrence since it requires the assembled genomic sequence to be different from the sequence of the molecule which gave rise to the trace. The donor, acceptor, and intron states are used to model alignments between a spliced trace and genome sequence. They all emit genome bases aligned to gaps in the observed peak sequence. The donor and acceptor states emit bases necessary to form a consensus GT/AG splice signal with probability 1 and the intron state emits A, C, G or T with equal probability. The genome_begin_gap and genome_end_gap states allow the trace to align anywhere within the genome sequence (semi-local alignment) by aligning long stretches of the genomic sequence before and after the alignment to gaps in the observed peak sequence. Finally the silent begin and end states formalize the beginning and end of the alignment. The states in figure 5-1 are color coded to indicate the types of alignments emitted. Red means non-emitting, blue an observed peak against a gap in the genome, green a genomic base against a gap in the observed peak sequence and black an observed peak aligned to a genomic base.

5.3 State Emission Probabilities

The probability distribution of states which emit an observed peak (match, secondary and noise) are defined by a bivariate normal distribution. One dimension of the distribution models the relative distance between the observed peak and its nearest predicted peak. The mean of this distribution is fixed at zero and the standard deviation is estimated in the training step. The second dimension of the bivariate normal models the area of the peak relative to other nearby peaks. Both the mean and standard deviation of this distribution are estimated in the training step. For simplicity it assumed that the normals are uncorrelated ($\rho = 0$) however we would like to explore whether or not this assumption is true.

5.4 Implementation and Training

Pair-HMM versions of the standard HMM algorithms were implemented for the continuous density pair-HMM described above. These include the Viterbi algorithm, forward and backward algorithms, and Baum-Welch parameter estimation. Parameters were trained for a sub-model of the one shown in figure 5-1 which excluded the states necessary to model introns. Training data was derived from a set of 96 double primed sequencing runs in which forward and reverse primers were used to sequence a pGEM plasimd vector lacking an insert. This results in a double trace in which a section from each strand of the plasmid is superimposed. Clean segments from 69 of the traces were identified by visual inspection. There are a total of 16,830 observed peaks in the training set. Of these 6500 peaks are expected to align to each strand of the pGEM plasmid sequence. This means 3830 noise peaks are present in the training set. These segments and the pGEM plasmid sequence were used to train the model parameters. Initial model parameters were selected which were thought to be close to the correct values. Self-transition probabilities for the genome_begin_gap and genome_end_gap were picked which were thought to approximately model the location of the trace within the plasmid sequence. Transition probabilities between the match, secondary, noise, and obs_gap states were set to be approximately equal with the exception that the transitions to the obs_gap state were set to be very small since it is assumed that this state is only rarely entered. Parameters to the bivariate normal distributions used to emit observed peaks were set based on informal observation of double trace data and expectation of their relationships (noise peaks spread over a larger distance than match or secondary peaks, noise peaks have smaller average area than match or secondary peaks, etc…). Baum-Welch parameter estimation was performed for 15 rounds from these initial values. On each iteration, expected counts were derived from each observed sequence and one strand of the pGEM sequence using Baum-Welch. After 15 rounds the model parameters had converged.

Chapter 6

Results

The only double traces initially available for testing were a few which were accidentally generated in the course of testing Twinscan predicted genes in rat. These resulted from alternately spliced transcripts and suggested that this system could be used to detect alternate splices. The first set of traces generated for testing were synthetic traces. Another set of experiments was performed to test the ability of the system to deconvolve traces that result from sequencing off of both ends of a template simultaneously. Next, a set of real traces was generated from alternately spliced transcripts to recreate the types of traces seen in rat but this time under controlled conditions.

6.1.1 Description of synthetic traces

The synthetic traces represent the ideal input to the algorithms developed here. They were mainly used to test the code during debugging. A genomic sequence and double trace pair are generated which simulate an alternate splicing event. Double traces were created by first generating a random “genomic” DNA sequence, a string whose characters are randomly drawn from the set {A,C,G,T} with uniform probability. Short portions of the “genomic” sequence are designated as exons. The sequence before the first and after the last exon is designated intergenic sequence and the rest is designated as intronic sequence. Bases at exon/intron boundaries are removed and replaced with consensus the GT/AG splice signal. This defines one gene isoform. An alternate form is defined by altering the initial gene, for example by removing an exon or altering the position of an individual splice site. The sequences corresponding to the spliced products of the two isoforms is extracted and used to build a synthetic double trace. This is done by walking through each sequence, starting at the first index, generating a guassian corresponding to the base in each sequence, and shifting to the right by 12 trace points. Spacing between peaks and parameters of the guassian which model the peaks were chosen to approximate peaks seen in known double trace examples. A program was written to feed the resulting trace values into PHRED, which packages them in .scf format. This synthetic trace .scf file can subsequently be treated as a normal trace file. Thus a variety of clean double traces for which the “correct” answer is known were generated. The “genes” modeled in the synthetic traces consisted of 5 exons each containing between 50 and 75 bases. The exons are separated by introns between 100 and 200 bases long. The “genes” are flanked by between 400 and 500 bases of intergenic sequence.

6.1.2 Results of trace recalling on synthetic traces

A test set of synthetic traces was generated as described above. The set consist of 40 randomly generated genomic sequences and double traces. Ten alternate splices of each alternate splice class; skipped exon, mutually exclusive exons, alternate 3 prime splice site and 5 prime splice site were created (see figure 2-7). Trace Recalling was applied to the synthetic double trace and genomic sequence with EST_GENOME alignment parameters, match = 1, mismatch = -1, gap = -3. In 39 of the 40 test cases the two forms of the gene were called exactly right. The case that didn't work was a mutually exclusive exon and it worked if the gap penalty in the first alignment was changed from –3 to –4. This seems to be the result of an odd self-similarity in the randomly generated genomic sequence. The mean and median percent identities of the first stage alignments (aligning ambiguity sequence to the genome) were 99.9% and 100%, respectively. For the second stage alignment (alignment of recalled sequence to the genome) the mean and median were 99.7 and 100%.

6.2 Dual Sequencing Primer Experiments

6.2.1 Description of dual primer set

As noted in 2.4.4 another way to generate double traces is mispriming. This section describes a controlled mispriming experiment in which traces were generated by sequencing off both primers flanking the insertion site of the pGEM vector in a single reaction. No insert was present so the traces generated represent the plasmid sequence. Four different reactions were used to generate the traces representing different ratios of the forward primer concentration to reverse primer concentration, 1:1, 2:1, 3:1 and 4:1. Each ratio was tested with 24 individual double-ended sequencing reactions.

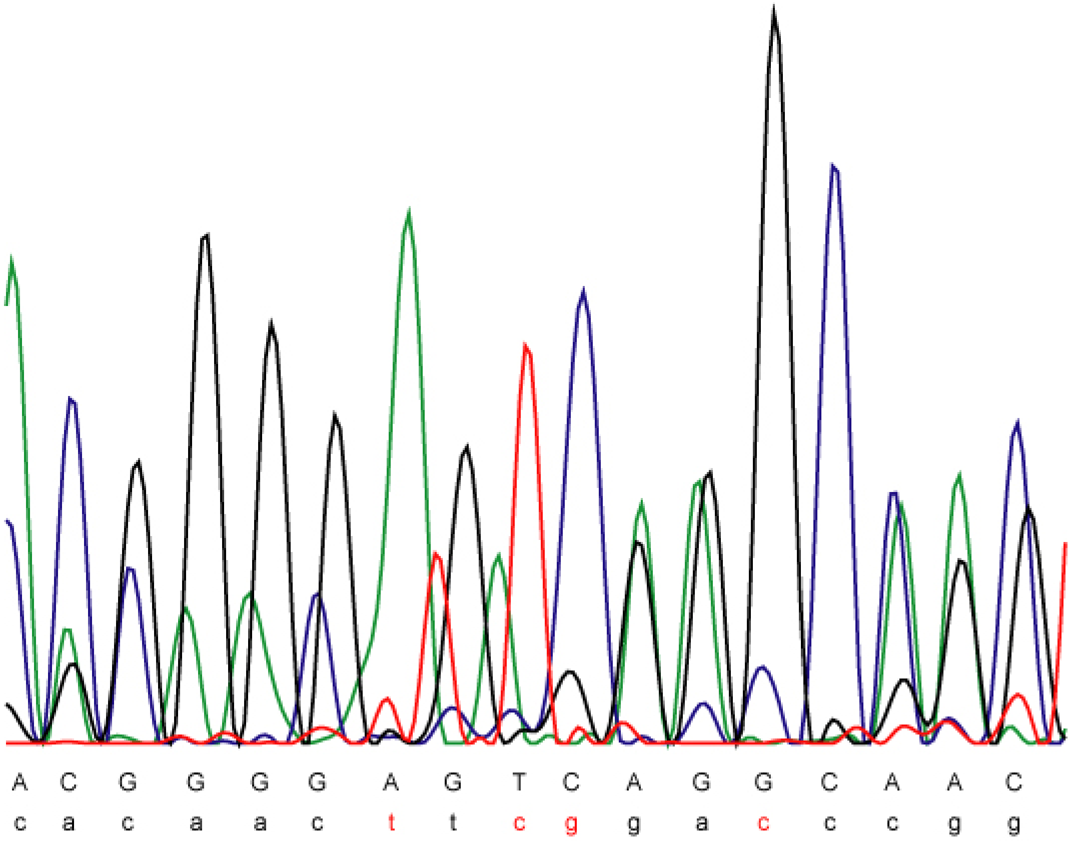
6.2.2 Results of trace recalling on dual primer set

Trace recalling was applied to each of the double traces and the sequence of the pGEM plasmid. The expected outcome of this experiment is a perfect ungapped alignment in the first alignment stage across the whole trace followed by a perfect alignment of the recalled sequence also across the entire length of the trace on the reverse strand. Statistics about the first and second stage alignments are presented in table 5-1, broken out by both for the entire set of 96 experiments and each set of 24 experiments with different ratio of primer concentrations. Notice that in general the second stage alignments are slightly shorter than the first stage alignments and have a lower percent identity. Note also that while the length of the alignment appears not to be correlated to the ratio of concentrations of the sequencing primers the percent identity seems to increase for both alignment stages as the ratio of primers approaches 1:1.

Table 6-1. Results of dual primer experiment.

Set	**Length**			**% Identity**		
	Mean	Median	Std dev	Mean	Median	Std dev
Stage 1 all	736	785	179	95	96	4
Stage 2 all	686	725	125	70	71	4
Stage 1 4:1	684	792	231	93	94	5
Stage 1 3:1	753	827	203	94	94	4
Stage 1 2:1	802	812	50	96	96	3
Stage 1 1:1	704	740	142	96	98	4
Stage 2 4:1	644	673	140	68	70	4
Stage 2 3:1	762	784	70	71	72	4
Stage 2 2:1	656	718	158	71	72	3
Stage 2 1:1	677	716	65	72	72	3

We had hoped to see similar percent identities the first and second stage alignments. As seen in table 6-1 the percent identities in the second stage alignments were lower than in the first stage. A major reason for this is illustrated in figure 6-1. The figure shows a section of double trace. The colors in the trace are: green for A, blue for C, black for G and red for T. At the bottom of the figure are the first and second stage alignments (using ambiguity sequence in the first stage) for the displayed trace section. Below the trace in capitol letters are the peaks aligned in the first stage. Below these in lower case letters are the peaks aligned in the second stage. Letters in red correspond to mismatched bases in the second stage alignment. The first, third and fourth mismatches appear to be the result of noise peaks masquerading as peaks from the secondary template. For example look at the first mismatch (red t). The correct call at that location was an A and there is a large A peak present, however, the small T peak was called in the recalling step. Similar arguments hold for the third and fourth mismatches. The second mismatch appears to be due to bad spacing. In this case (first red c) the correct call would have been an A. There is clearly an A peak nearby but the C was called because it was much nearer the predicted peak though it was also much smaller.



```
         First alignment                       Second alignment

genome:     ACGGGGAGTCAGGCAAC         genome:     CACAACATACGAGCCGG
            |||||||||||||||||                     |||||| |  || ||||
ambig:      MMSRRSWKYSRRSCMRS         recalled:   CACAACTTCGGACCCGG
```

Figure 6-1. A section of double trace with first and second stage base calls and alignments (details in text).

6.2.3 Results of pair-HMM method on dual primer set

The pair-HMM method was tested on a subset of the dual primer set. The training set described in section 5.4 was used. Each of the 69 clean segments of observed peaks were aligned to both strands of the pGEM sequence this resulted in 138 alignments. When each of the clean segments was extracted the number of bases present in the segment was recorded. This number was compared to the number of bases predicted by the pair-HMM method. All pair-HMM derived sequences were within 5 bases of the expected length with only one exception, which was 15 bases shorter. BLASTing each one against the pGEM vector sequence tested the quality of the pair-HMM derived sequence. In all cases this resulted in a 100% match. From this I conclude that the pair-HMM described in chapter 5 can be used to align an experimentally derived double trace with minimal noise to the genome. More work is required to determine how the pair-HMM alignment method behaves in nosier regions of the trace.

While it is possible that this test could result in an over-training problem, I do not believe this is the case. First, The training set consisted of 16, 830 observed peaks. This should be more than enough data to train the small model used. Second, while training only used one strand of the pGEM vector sequence testing was performed on both strands. If over-training were taking place the results on the strand for training would appear to be much better than those on the strand not used in training. This was not observed.

6.3 Alternate Splice Experiments

6.3.1 Description of alternate splice test set

A set of 96 targets in the human genome was selected to test the methods developed in this thesis. I refer to targets instead of genes because some genes that containing multiple alternate splices are represented more than once in the test set, once for each alternately spliced target. The goal was to differentiate between targets with and without alternate isoforms and to determine the two different isoforms that existed when present. This set also provided quality examples of double traces that were used to develop the algorithms. The set consisted of 49 targets thought to be alternately spliced and 46 thought not to be alternately spliced that served as negative controls. One target was thrown out of the analysis due to improper primer placement. Each target was classified as alternately spliced or not by viewing alignments of RefSeq genes and ESTs to the human genome on the UCSC genome browser. If two RefSeq or EST isoforms were present the target was categorized as alternately spliced. If only one form was present supported by two or more ESTs the site was marked as not alternately spliced. The composition of the alternately spliced set was 34 skipped exon, 8 alternate splice site, 5 retained intron and 2 mutually exclusive exon targets.

Once the test set was finalized PCR primers were designed flanking the alternately spliced targets. For the non-alternately spliced targets primers were designed to amplify regions of similar length to the alternately splice targets. The selected regions were amplified and sequenced three times. Most traces from the first round of sequencing were not usable so that set of traces was not discarded. The other two sequencing rounds produced better traces and were used in subsequent analysis. Thus there are 4 repetitions of each experiment. Forward and backward reads from two different sequencing runs. Since it was known which genes were being tested, the genomic sequence of these genes was extracted and used in the alignment steps instead of aligning to the whole genome.

A follow up cloning experiment was performed. The rationale of this experiment was to determine exactly which isoforms were present in the PCR mixture that was used to generate the double traces. The PCR products were randomly inserted into plasmid vectors. The plasmids were introduced to a bacterial culture. The bacteria were plated and allowed to form colonies, each of which should have possessed an insert containing plasmid with a single isoform present in the mixture of PCR products. Twelve colonies were selected for each of the 96 experiments and the plasmid inserts were sequenced.

6.3.2 Results of trace recalling on alternate splice test set

Results are based on visual inspection of first and second stage alignments to genomic sequence. These alignments are compared to alignments of all refseq genes, ESTs and clone sequences to the same genomic sequence. Examples of such alignments for one example are presented in figure 5-1. Red and green genes represent the first and second stage alignments to genomic sequence for 2 of the 4 repetitions of the experiment that worked. The leftmost red gene represents the first stage alignment and the rightmost red gene represents the second stage alignment, similarly for the green genes. Blue genes represent alignments of sequence derived from cloning experiments and brown genes represent pooled refseq and EST alignments from the UCSC genome browser. Green boxes represent the positions of the original PCR primers used to select the region to amplify. The black bar on the right hand side of the figure gives the coordinates of the genomic sequence to which all alignments are made. An experiment is considered a success if in at least one of the repetitions the first and second stage alignments correctly identify each of the two forms of the gene thought to be present based on refseq and EST evidence. By this criterion 23 of the 48 genes thought *a priori* to be alternately spliced were successes. In this example (figure 6-2) there is clear evidence for the skipping of the exon near coordinate 70,000 from the refseqs and ESTs. Evidence also exists from the cloning experiments that the transcript is alternately spliced in our mRNA pool. The two experiments shown exactly highlight the skipped exon.

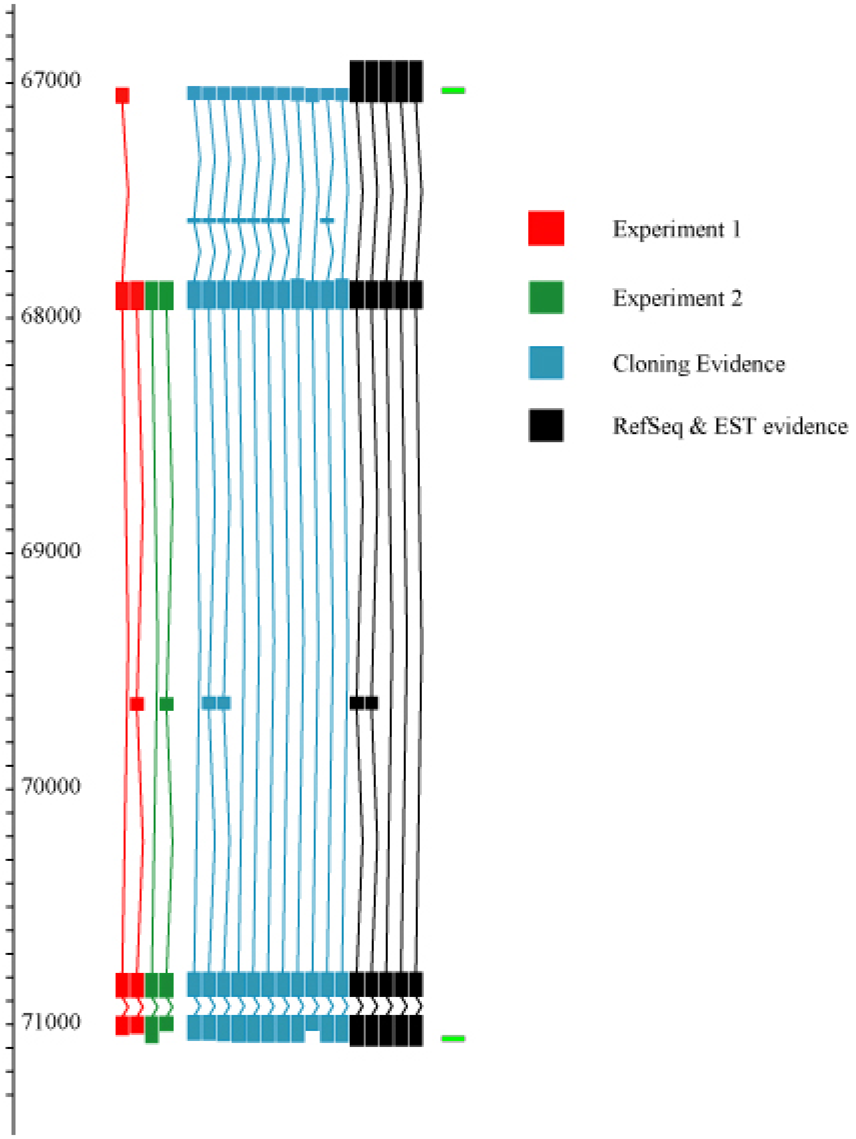


Figure 6-2. Example of a successful experiment.

The results of the cloning experiment on the targets in the alternate splice set thought to be alternately spliced are presented in tables 6-2 and 6-3. Table 6-2 shows data for the successful cases and table 6-3 shows data for the cases that did not work. The first column of these tables gives the ID number of the experiment. The second column gives a conclusion as to whether or not the cloning experiment supports the fact that the sequence is actually alternately spliced, a minus sign indicates that it is not, a plus sign indicates that it is and a question mark indicates that there were no alignments to the region between the PCR primers. The third column gives the evidence for the conclusion in the previous column, if there is a minus in the second column the third column gives the number of times the same form was observed. If there is a plus in the second column the third column gives the number of times each alternate form was observed. Otherwise the third column is blank since nothing was observed. The third column can be used to gauge the reliability of the conclusion in the second column. For example since a single form was seen 7 times as in the case of experiment 88 one can be fairly confident that there is no alternate splicing at the abundance level our system is designed to detect. On the other hand since only 1 alignment is seen as in experiment 96 we cannot conclude that there is no alternate splicing. Finally, the fourth column indicates in how many of the repetitions of the experiment there was a first stage alignment that matched one of the refseqs or ESTs. There are 16 targets for which the cloning experiments indicated alternate splicing in our test sample. Of these 12 were identified as alternately spliced by trace recalling, 4 were not.

Table 6-2. Successful experiments

Experiment ID	Conclusion	Evidence	Alignment
04	+	9/2	2
12	?		4
23	+	3/1	4
24	?		4
26	+	6/1	3
27	?		4
30	+	1/1	4
31	+	2/1	4
33	-	1	1
37	+	6/2	4
40	-	3	1
58	?		3
59	?		4
61	+	2/1/1	4
65	+	7/3	4
69	?		4
73	+	3/2	4
74	-	4	4
76	+	3/1	2
79	+	7/3	4
87	?		4
92	-	4	4
94	+	3/1	4

Table 6-3. Unsuccessful experiments

Experiment ID	Conclusion	Evidence	Alignment
01	?		2
06	-	2	4
07	?		4
09	?		0
10	+	7/1	1
12	?		4
16	+	8/2	4
20	-	1	0
22	-	4	4
25	?		0
28	?		3
43	+	7/1	4
48	+	1/1	2
51	?		0
52	-	7	3
55	?		0
56	-	1	0
60	?		1
62	-	1	2
67	-	1	3
68	?		0
80	?		1
82	-	2	4
88	-	7	2
91	?		2
96	-	1	2

As outlined in 3.3 a system was developed for automatically identifying and classifying alternate splices based on the first and second stage alignments. This system was applied to the alternate splicing set. The output of this procedure can be evaluated both on the locus and individual trace levels. On the locus level all 23 alternate splices found by visual inspection were identified. In addition to these it flagged 5 other experiments identified visually as alternately spliced. Four of these were incorrectly called alternate splices in targets thought to be alternately spliced and one was a predicted alternate splice in a target thought not to be alternately spliced. We consider these false positives. On the individual trace level 62 traces were flagged as alternately spliced, only 6 of these were false positives.

6.3.2 Results of integrated alignment on alternate splice set

The integrated alignment method was applied to the alternate splicing set. The output of the integrated alignment procedure is the same as the output of the trace recalling procedure (two alignments to the genomic sequence) so the results could be analyzed in the same way. Essentially the same set of alternately spliced targets was found by integrated alignment as was found by trace recalling. Three targets (12, 40 & 59) that were correctly identified as alternately spliced by trace recalling were missed by integrated alignment. However two (20 & 96) that were missed by trace recalling were correctly identified by integrated alignment. In another case (16) an alternate splice form not corresponding to refseq or EST evidence was identified in two experiments, suggesting that it may be a novel alternate splice form. At the individual trace level, 67 traces were flagged as containing alternate splices and it appears that 5 of these are false positives. However 16 of the 62 true positive examples required different alignment parameters than those described in section 4.3.3 whereas all experiments with the integrated alignment method used the same model parameters. This represents a major drawback to the integrated alignment method.

Chapter 7

Conclusions

7.1 Discussion of experimental results

7.1.1 Discussion of synthetic trace experiment

As mentioned previously the synthetic traces are mainly a debugging tool. Applying any of the methods developed to these test cases should result in alignments that exactly match the specified isoforms with 100 percent identity. This is the case when trace recalling is applied to the set of 40 synthetic traces. From this two conclusions can be drawn. First the theoretical basis of trace recalling is sound. Second as implemented the original trace recalling code and modifications to existing code (EST_GENOME) are working as expected.

7.1.2 Discussion of dual primer experiment

This experiment tests the ability of the methods developed to handle the simple task of deconvolving a long, ungaped double trace with a short genomic segment. It is more difficult than the synthetic trace experiment in that the traces were actually generated in the lab and therefore were subject to the various kinds of noise described in section 2.3.1. In addition it represents the first attempt to quantify the effect of altering sequencing reaction chemistry to optimize double trace analysis.

It is encouraging that the percent identity of the first stage alignment was generally so high, 94.6% average when all 96 experiments were pooled. It is also encouraging that the first stage alignments are so long, with a mean of 736 bp and median of 785 bp. The high standard deviation in the alignment lengths of 179 bp is suspicious but can be partially explained by the large difference between the mean and median. Upon closer inspection it was noted that this high standard deviation is the result of a handful of outliers with very short lengths. These most likely represent failed sequencing runs. The first stage alignment is especially important because if high percent identity alignments are not obtained at the start the rest of the procedure cannot function properly.

The second stage alignments had a mean and median length of 686 and 725 bp and a mean percent identity of about 70%. That the lengths of the alignments of the recalled sequence are similar to the lengths of the original alignments is promising. While the percent identify is quite high it is somewhat disturbing since we expected that in this problem the recalled sequence would be of very high quality. This result can partially be explained by the observation at the end of section 6.2.2. Sometimes when the same base is present in both templates at the same position, a noise peak is mistaken for a peak originating in the other template. This can be corrected by examining the ratio of the areas between the two peaks. If the smaller peak is much smaller than the larger peak it can be ignored. This may cause problems, however, in double traces where the secondary template trace is much weaker than the primary template trace.

There is no apparent relationship between relative concentration of primers and length of either first or second stage alignments. On the other hand, there does appear to be a trend relating relative concentration of primers to percent identity of the alignments. As the ratio of the concentrations between the primers approaches 1:1 the percent identity of both stages increases. This makes sense because the ratio of dye labeled DNA fragments used in the sequencing reaction is related to the relative concentration of the primers. In turn this affects the relative heights of the peaks in the double trace. If the peaks are nearly the same height there is less chance of confusing a smaller, but real, peak for a noise peak. The resulting trace also “looks” more like the ideal synthetic double traces if the peaks are nearly the same height. The increase, thought is slight, and due to the high standard deviation in percent identity all measurements are within each other’s 95% confidence intervals. Thus, no statistically significant conclusions can be drawn.

Another similar study with more repetitions of each concentration is needed to validate this finding.

The test of the pair-HMM using the dual primer test set should be considered a very preliminary result. It demonstrates the validity of using the pair-HMM and the correctness of the implementation. This method is still under development. At present it cannot distinguish between two isoforms on the same strand. We plan to implement Waterman-Eggert sub-optimal alignment to do this.

7.1.3 Discussion of alternate splicing experiment

When viewing the alignments resulting form trace recalling on the alternate splicing experiments such as the one in figure 5-1 the most striking observation is the binary nature of the outcomes. One would expect there to be a few experiments in which the two refseq/EST isoforms were detected exactly, several for which the general isoforms could be discerned but the exact boundaries were incorrect, and others which totally failed. This, however, was not the case. In almost all experiments either the two correct isoforms were produced or the experiment totally failed. Most likely this is due to the fact that a spliced alignment program was used to align the sequences rather than an alignment program such as BLAST. In some cases the alignments, especially the secondary alignments, have a low percent identity. Despite this, the lower splice penalty for a consensus GT/AG intron as compared to a non-consensus splice signal seems to often “lock in” the correct alignment.

Almost half (47%) of the targets thought to be alternately spliced were identified. There are 5 cases (09, 25, 51, 55, 68) in table 5-2 for which there appears to have been no transcript present at all. In these experiments no usable first stage alignments were recovered and there was no cloning evidence for a transcript. In 2 cases there is convincing evidence that there was a transcript present but it had only a single isoform (22, 88). In these cases 4 or more cloning sequences were recovered but all were of the same isoform. In only 4 experiments (10, 16, 43, 48) did cloning provide convincing evidence for the presence of an alternately spliced transcript. Not much can be said about the remaining 15 experiments since they had little cloning evidence but also had at least 1 convincing first stage alignment suggesting that at least one transcript isoform was present.

The automated detection procedure outlined in 3.3 performed very well. It identified all 23 of the alternate splices deduced from visual inspection. In addition to this it predicted only 5 false positives, 2 of which might be real altsplices for which there is simply no refseq or EST evidence yet. The other 3 false positives appear to be alignment artifacts. An example of such an artifact is presented in figure 7-1. This problem arises when there is a sequence similarity between an exon and the intron region upstream of the next exon.

As in this example, the first exon can be shifted and attached to the next exon creating the appearance of a skipped exon followed by an alternate splice site. Further refinement of the algorithm may solve this problem. Automated detection is important if this system is to be used in a high-throughput manner since visual inspection of the alignments is at present the most time consuming part of the procedure.

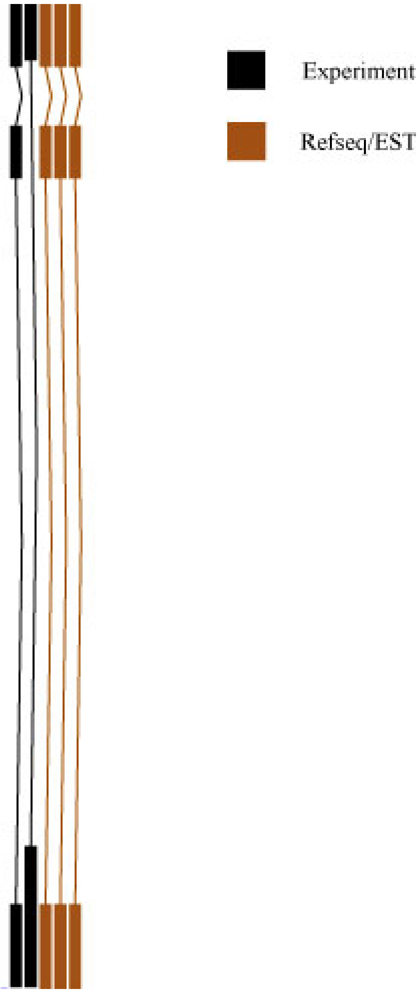


Figure 7-1. An alignment artifact.

Generally the trace recalling and integrated alignment procedures have similar accuracies. The trace recalling results presented were obtained with a single set of alignment parameters. On the other hand 15 of the 62 true positives found by integrated required manual tuning of the parameters. This represents a major drawback to the integrated alignment approach.

7.2 Applications and future work

The methods described in this work have been shown to be effective at deconvolving double traces by aligning them to assembled genomic sequence. At present trace recalling appears to be the most robust technique. Integrated alignment seemed the natural next step but met with limited success. We believe there were two main reasons for this outcome. One is lack of expressiveness inherent in the alignment model and inability to automatically determine optimal parameters. We believe the pair-HMM model of alignment addresses these problems and have promising preliminary evidence that this is so. The focus of future work on this project will be to refine the pair-HMM model. This will include testing it on nosier traces, adding in the splicing states, and implementing the Waterman-Eggert algorithm to find sub-optimal alignments on the same strand.

There are many exciting applications of this technique. Since we require the existence of assembled reference sequence they all involve resequencing. Section 2.4 outlines numerous circumstances in which double traces are generated and an assembled genomic sequence is available. The main application so far has been the elucidation of alternate splice forms following RT-PCR and direct sequencing. It has also been demonstrated that this technique can be used to deconvolve double traces generated by intentionally sequencing a template with two different sequencing primers which anneal to different strands. So far this is simply the sequence of a plasmid with no insert but we have data from a similar experiment with an insert containing plasmid. Another interesting application is the T-DNA mediated knockout libraries. In this application we would like to use short double trace tags to determine the integration sites of both T-DNA segments in the cases where there are exactly two. This poses a new set of problems since the genomic sequence in this application is the whole *A. thaliana* genome, not just a short segment. Further down the road we are considering a system that aligns double traces directly to each other rather than to a genomic sequence. In principle this could be used to assemble double traces much the same way sequences derived from single traces are assembled at present. If it works, this could significantly reduce sequencing costs.

References

Alonso, J. M. et al. 2003. Genome-wide insertional mutagenesis of *Arabidopsis thaliana. Science* 310:653-657.

Baum, L. E. 1972. An equality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes. *Inequalities*. 3:1-8.

Bellman, R. E. 1957. Dynamic Programming. New York: Dover Publications Inc.

Cartegni, L, Chew, S. L., and Krainer, A. R. 2002. Listening to silence and understanding nonsense: Exonic mutations that affect splicing. *Nature reviews genetics* 3:285-298.

Dayhoff, M. O., et. al. 1978. A model of evolutionary change in proteins. *Atlas of protein sequence and structure*. 5(3): 345-352.

Durbin R., et. al. 1998. Biological sequence analysis. Cambridge: Cambridge University Press.

Elliott, R. J., Aggoun, L. and Moore, J. B. 1995. Hidden markov models: estimation and control. New York: Springer-Verlag.

Ewing, B., Hiller, L., Wendl, M. C., and Green, P. 1998. Base-calling of automated sequencer traces using Phred I: Accuracy assessment. Genome Research. 8: 175-185.

Garcia-Blanco, M. A., Baraniak, A. P. and Lasda, E. L. 2004. Alternatice splicing in disease and therapy. *Nature Biotechnology*. 22:535-546.

Gotoh, O. 1982. An improved algorithm for matching biological sequences. *Journal of Molecular Biology* 162:705-708.

Gusfield, D. 1997. Algorithms on strings, trees, and sequences. Cambridge: Cambridge University Press.

Haan, N. M. and Godsill, S. J. 2002. Baysean models for DNA sequencing. 2002. *Proc. IEEE conference on acoustics, speech and signal processing.* IV:4020-4023.

Henikoff, S. and Henikoff, J. G. 1992. Amino acid substution matrices from protein blocks. *Proc. Natl. Acad. Sci*. USA. 89: 10915-10919.

Maxam, A. M. and Gilbert, W. 1977. A new method for sequencing DNA. *Proc. Natl. Acad. Sci. USA.* 74:560-564.

Meyers, E. W. and Miller, W. 1988. Optimal alignments in linear space. *Computer Applications in the Biosciences* 4:11-17.

Modrek, B. et al. Genome-wide detection of alternative splicing in expressed sequences of human genes. 2001. *Nucleic acids research.* 29(13):2850-2859.

Mott, R. EST_GENOME: a program to align spliced DNA sequences to unspliced genomic DNA. 1997. *Computer applications in the biosciences.* 13(4):477-478.

Needleman, S. B. and Wunsch, C. D. 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology* 48:443-453.

Prober, J. M. et. al. 1987. A system for Rapid DNA sequencing with fluorescent chain-terminating dideoxynucleotides. 1987. *Science*. 238:336-341.

Rabiner, L. R. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *IEEE ASSP Magazine*. 3:4-16.

Sanger, F., Nicklen, S. and Coulson, A. R. 1977. DNA sequencing with chain-terminating inhibitors. *Proc. Natl. Acad. Sci USA.* 74(12):5463-5467.

Siebert, P. D. et. al. 1995. An improved PCR method for walking in uncloned genomic DNA. *Nucleic Acids Research*. 23(6)1087-1088.

Smith, T. F. and Waterman, M. S. 1981. Identification of common molecular subsequences. *Journal of Molecular Biology* 147:195-197.

Swerdlow, H. and Gesterland, R. 1990. Capillary gel electrophoresis for rapid, high resolution DNA sequencing. *Nucleic Acids Research*. 18(6):1415-1419.

Waterman, M. S. and Eggert, M. 1987. A new algorithm for best subsequence alignments with application to tRNA-rRNA comparisons. *J. Mol. Biol*. 197:723-728.

Wheddon C. and Linggard R. 1990. Speech and language processing. London: Chapman and Hall.

Wu, J. K., et. al. 2004. Identification of rat genes by TWINSCAN gene prediction, RT-PCR, and direct sequencing. *Genome Research*. 14(4): 665-671.

Young, S. et. al. 1995. The HTK Book. Redmond WA: Microsoft Corporation.

Vita

Aaron Tenney

Date of Birth	September 17, 1977
Place of Birth	St. Louis, Missouri
Degrees	B.S. Computer Science University of Missouri, St. Louis Summa cume laude, May 2001 B.S. Applied Mathematics University of Missouri, St. Louis Summa cume laude, May 2001 B.A. Physics University of Missouri, St. Louis Summa cume laude, May 2001
Publications	Aaron Tenney, Randall H. Brown, Charles Vaske, Jennifer K. Lodge Tamara L. Doering, and Michael Brent. “Gene prediction and verification in a compact genome with numerous small introns”, *Genome Research*. 14(11):2330-2335.
Posters & Presentations	Randall H Brown, Aaron Tenney, Charles Vaske, Jennifer Lodge, Tamara Doering and Michael Brent. “Gene prediction and

annotation in Cryptococcus neoformans”, *Proceedings of ISHAM 2003*. May 2003.

Aaron Tenney, Jaiquan Wu, Diana Kolbe, and Michael Brent. “Base calling traces derived from multiple templates”, *Proceedings of the AGBT/AMS conference*. February 2004.

Aaron Tenney, Jaiquan Wu, Diana Kolbe, and Michael Brent. “Base calling traces derived from multiple templates”, *Proceedings of the Cold Spring Harbor biology of genomes conference*. May 2004.

December 2004

Short Title: Basecalling Multitraces Tenney, M.S. 2004