

A set reduction and pattern matching problem motivated by Allele Specific Primer Design

Christopher T Lewis
University of Saskatchewan
Dept of Computer Science
Saskatoon, Saskatchewan
ctl271@mail.usask.ca

ABSTRACT

Allele specific primers allow researchers to rapidly test for the presence of an allele in a sample. Manually designing these primers for a polyploid organism is a laborious task, and an automated method would expedite tests for desirable alleles. This paper formally defines the problem of allele specific primer design in terms of set reduction and pattern matching, and examines the applicability of suffix trees by reducing each step to a related problem solved with suffix trees.

General Terms

Allele, Polyploid, PCR Primer Design, Bioinformatics, Suffix Trees, Sets, Pattern Matching

1. INTRODUCTION

1.1 PCR and Alleles

PCR is a method to amplify specific DNA sequences using a pair of primers [12] (referred to herein as the primer, or primer pair). This pair is designed from a set of template sequences so they flank the desired region. When the primers bind to the DNA they enable amplification of the flanked region by a DNA polymerase which extends the primer sequence and creates two copies of the flanked region. This process is repeated many times causing exponential amplification of the sequence and ensuring it is present in detectable levels (Fig. 1).

Allelic variants—different sets of alleles in an individual—introduce diversity into the species and account for characteristics such as blood type in mammals or levels of disease resistance in plants. An allele is a variant of a specific gene located at a single chromosomal location (a locus) [14].¹ Additional genomic diversity is introduced when genes duplicated during evolution acquire a new function. Gene duplication is common in higher organisms and whole genome duplication (polyploidy) is a phenomenon that is particularly common in plants. Homologous genes within the same species (paralogues) are a result of gene duplication, while homologues in different species (orthologues) are a result of a speciation event. Homologous genes have common regions that can be used to classify them as members of the same gene family.

¹In this discussion a locus is assumed to contain only a single gene.

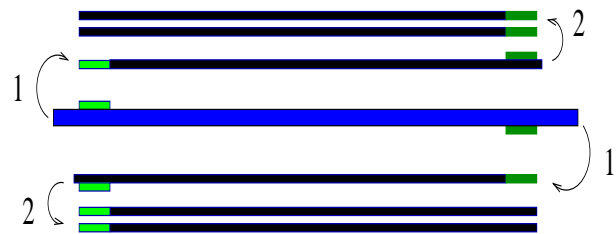


Figure 1: The PCR primers (small green boxes) bind to single strands of the DNA sequence (blue box) allowing extension of the primers by DNA polymerase. 1) After one iteration the process has created two copies of the original sequence (navy boxes). One copy begins with the left primer, and the other begins with the right primer. 2) A second iteration doubles the number of sequences by duplicating the original sequence and each of the copies. Repeating this process many times causes exponential replication of the target sequence.

Alleles at a single locus may be very similar and a valid primer pair will exploit small differences between the sequences (as few as 1 in 20 bases) to amplify only the desired allele. Designing the primer from a restricted set of sequences increases the likelihood of finding a valid primer pair because it reduces the sequence diversity of the set. A pair is valid only in the set of sequences from which it was designed; if a pair is applied to a larger or different set of sequences the primer might match and amplify undesirable sequences. Thus each step in the following process restricts the resulting pool of sequences so that amplification produces only a specific subset of sequences. This process results in a final pool of sequences for allele specific primer design that is as narrow as possible and ensures that primer pairs will be found for target alleles.

To successfully develop allele specific primer pairs for a polyploid organism, sequences must be identified that represent individual gene loci and allelic differences must be found for each locus. This process requires three steps:

1. Design a primer pair that will selectively amplify the gene family containing the target gene from a pool of genomic sequences.

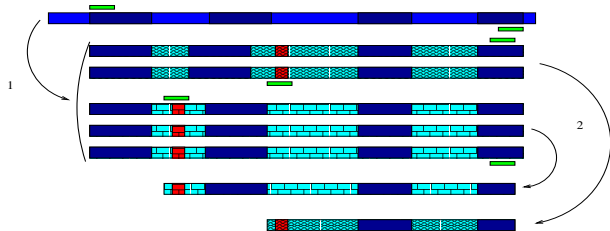


Figure 2: 1) Primers (green) have been designed for the conserved regions (dark blue areas) of a family specific consensus sequence and used to amplify members of the gene family. 2) Analysis of the amplified family members identifies regions specific to two subsets of the family (shown in red). These regions allows the amplified regions to be divided into two clusters corresponding to two loci of the gene family. Each locus has a specific region (shown in red) which can be used to design primers to amplify the individual loci, which results in the final pair of shorter sequences.

2. Design a locus specific primer pair to amplify the locus containing the desired allele from the sequences produced by amplification with the family specific primer (Fig. 2).
3. Design a primer pair to amplify the desired allele from the pool of sequences produced using the locus specific primer (Fig. 3).

The three steps seem quite similar due to their common purpose: reduction of the existing set. However, they are distinct as the primers must be designed to leverage different regions and levels of conservation within the sequences at each stage. Thus different techniques are required to identify appropriate regions for primer design. For instance, the pool of family specific sequences contains representatives from different loci, and each locus is expected to contain regions specific to itself. To design a primer in these locus specific regions they must first be identified. This requires a technique to 1) to cluster the sequences corresponding to each loci, and 2) to identify the locus specific regions within the loci. Designing a primer pair within one of these locus specific regions ensures that only the target locus is amplified by the primer. On the other hand, the pool of allelic sequences will be very similar and will only contain a small number of differences (Fig. 3). As such there is no need to cluster sequences in this step of the problem because redundant copies are expected to be identical. Instead the techniques applied to this step must a) discard identical sequences, and b) identify allele specific differences in the remaining sequences. Designing the primer pair so each member contains one or more of these differences allows it to amplify a specific allele.

1.2 Suffix Trees

Suffix trees allow efficient string operations and can be constructed in linear time and space [5]. A Generalized Suffix Tree (GST) contains more than one string but maintains the linear time and space bounds. There are two principal types of suffix trees: explicit suffix trees, where each suffix

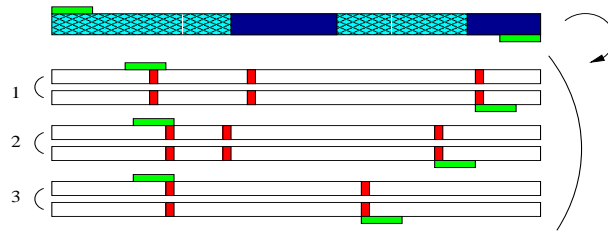


Figure 3: Closer examination of the sequences amplified by the locus specific primer (white boxes) reveals small Single Nucleotide Polymorphisms (SNPs) (denoted in red) present in the sequences. In this case there were three alleles amplified by the locus specific primer (1,2,3). The number of alleles can be determined by clustering the sequences based on shared SNP patterns. The SNPs can then be used to design primers to amplify individual alleles.

is guaranteed to end at a leaf node; and implicit suffix trees, where a suffix may end at an internal node or edge. Suffix trees have been used for primer design to store the sequences under analysis [6], and should be applicable to each part of the allele specific primer design problem:

- Step one requires a method to select regions common to all members of the set.
- Step two requires a method to cluster a group of sequences and select regions that are specific to individual clusters.
- Step three requires a method to select subsequences that are specific to individual sequences. At this stage there should be no need to cluster sequences because redundant copies are expected to be identical; instead identical copies are removed.

GSTs have been used to identify all conserved regions within a set of genomes when performing whole genome alignment [8, 4, 3], and to select gene specific subsequences for microarray design [9]. Similar techniques are applicable to selecting appropriate regions for primer design. Probabilistic suffix trees (PSTs) have been used to generate profiles of conserved patterns within proteins and to cluster a family of proteins into its constituent members [1, 2], and similar techniques are applicable to clustering locus specific sequences.

1.3 Overview

This project examines the applicability of suffix trees to each step of allele specific primer design by showing how it relates to a similar problem solved using suffix trees. The first step requires a method to select all common sequences in the set of sequences: suffix trees were used for this purpose in whole genome alignment [8, 4, 3]. The second step requires a method to cluster sequences by locus based on a profile of conserved sequences: suffix trees were used in a similar fashion for protein classification [1, 2]. The third step requires a method to select sequences which uniquely identify individual sequences in the set: suffix trees were used for

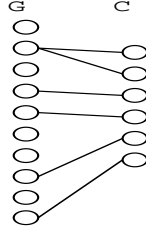


Figure 4: The relationship between cDNA (C) and genomic (G) sequences. Each cDNA is produced from one genomic sequence, while each genomic sequence may produce zero or more cDNA.

this purpose to design oligonucleotides for microarray experiments [9]. The remainder of this paper makes each of these points much more explicit and precise.

The remainder of this paper is organized as follows: (2) Each step of the allele specific primer design problem is formally defined in terms of strings and sets. This formalism ensures the problem is properly understood and will guide the solution. (3) Related problems solved using suffix trees are introduced and parallels are drawn between pairs of corresponding problems. These similarities provide support for the applicability of suffix trees to the problem. (4) Initial results are discussed for the first step of the primer design process, and (5) conclusions and thoughts on future work are provided.

2. FORMAL DEFINITIONS

2.1 Notation and Necessary Functions

Sets are denoted by an uppercase letter. For instance, F might denote a set of family specific sequences. The cardinality of a set is given by $|F|$.

f refers to a string in a set F . f^1, f^2 are used to refer to distinct strings from the same set. A substring of f is denoted as f_j , while f_j^i is a substring j of string f^i within set F . Adjacency indicates concatenation; for instance s_1s_2 is the concatenation of two substrings and fs is the concatenation of string f and string s . The length of string s is given by $|s|$, and strings start at position 0.

Define a function $pos(p, t)$ which returns the position of the first character of substring p in string t and -1 when p is not found in t .

2.2 Concepts

cDNA and Genes

Given two sets of strings over the alphabet $\{A, T, G, C\}$:

- G - a set of genomic sequences;
- C - a set of cDNA sequences;

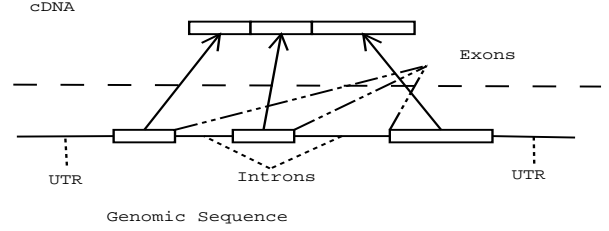


Figure 5: The relationship between genomic sequence and cDNA. The exonic regions of each genomic sequence are flanked by unexpressed regions (introns in the middle and UTR on extreme ends). The exonic regions are copied and then concatenated with their neighbours to form a shorter cDNA/mRNA sequence.

there exists a relation² $TSC \subseteq G \times C$ s.t.

$$\forall c \exists! g \text{ s.t. } (g, c) \in TSC.$$

This allows definition of the function

$$TSC' : C \rightarrow G,$$

which is neither injective—multiple cDNA may be produced from a single gene, nor surjective—not every genomic sequence produces a cDNA (Fig. 4).

The relationship TSC from genomic sequence to cDNA cannot be defined as a function because each genomic sequence may produce 0 or more cDNA. However this biological fact can be captured by defining a family of subsets C' of TSC :

$$\forall g \in G, \exists C'_g = \{(x, y) \in TSC \mid g = x\},$$

Note that C'_g could be $\{\}$ for some g .

Furthermore, $\forall (g, c) \in TSC$, g is a superstring of c such that c and g can be split into sets of ordered maximal substrings $\{c_1, \dots, c_k\}$ and $\{g_1, \dots, g_m\}$ where each region c_i has a corresponding region of genomic sequence g_j from which it was expressed (Fig. 5):

$$\forall c_i \exists g_j \text{ s.t. } (c_i = g_j)$$

and each expressed region of genomic sequence is flanked by an unexpressed region of genomic sequence. This latter statement can be formalized as follows:

$$\forall c_i \exists g_j \text{ s.t. } pos(g_{j-1}, g) < pos(c_i, g) < pos(g_{j+1}, g),$$

$$g_{i-1} \text{ and } g_{i+1} \notin \{c_1, \dots, c_k\}$$

Recall that $c_i = g_j$, so $pos(c_i, g) \equiv pos(g_j, g)$. Genomic regions g_1 and g_m correspond to the untranslated region (UTR) of the gene, and regions $\{g_2, \dots, g_{m-1}\} \notin \{c_1, \dots, c_k\}$ correspond to introns.

²Note that cDNA exists only in the laboratory; in reality the genomic sequence produces mRNA, and the mRNA is turned into cDNA in the lab because it is easier to work with. Thus the relation TCS represents an indirect relationship where the cDNA represents the mRNA.

Exons and Introns

Consider some $(g, c) \in TSC$. Expressed regions of g , substrings $\{g_j | c_i = g_j\}$ for some c_i , are called exons. The unexpressed regions, substrings $\{g_j | c_i \neq g_j\}$ for all c_i s.t. $g_i \neq g_1$ and $g_i \neq g_m$, are called introns. g_1 and g_m are referred to as UTRs.

Exons and introns have the following properties:

1. The exons are expressed—they produce a region of corresponding cDNA, while introns are unexpressed—they do not produce a region of the corresponding cDNA.
2. The exons and introns alternate within the gene, and every genomic sequence begins and ends with an unexpressed region (Fig. 5).
3. The exons of a gene exhibit greater conservation than the introns when comparing two related genomic sequences.

Define a similarity function, sim , which measures the similarity of two strings:

$$sim : S \times S \rightarrow [0..1]$$

where 1 indicates that the strings are similar, and 0 indicates that the strings are dissimilar. This function is symmetric; $sim(s^1, s^2) = sim(s^2, s^1)$.

If the similarity function is applied to a general set of related sequences R , we have:

$$\forall r^j, \forall r^k \in R, sim(r^j, r^k) \text{ is close to } 1.$$

A value is *close to 1* if it is less than 1 by a small defined threshold, e.g. 0.25.

Consider members f^j and f^k of a gene family $F \subseteq G$. Let f_g^j and f_h^k be substrings of f^j and f^k respectively.

Define the exons of f^j and f^k as:

$$exons(f^j) = \{f_g^j | f_g^j \text{ is an exon}\}$$

$$exons(f^k) = \{f_h^k | f_h^k \text{ is an exon}\}$$

and introns as:

$$introns(f^j) = \{f_g^j\}_{\forall g} - exons(f^j)$$

$$introns(f^k) = \{f_h^k\}_{\forall h} - exons(f^k).$$

By property 3 of introns and exons, each exon in f^j is expected to have a pair in f^k where the similarity is close to 1:

$$\forall e^j \in exons(f^j) \exists e^k \in exons(f^k) \text{ s.t.}$$

$$sim(e^j, e^k) \text{ is close to } 1,$$

and each of these pairs (e^j, e^k) will be more similar to each other than their flanking intronic regions are to each other:³

$$\exists i^j \in introns(f^j) \text{ and } i^k \in introns(f^k) \text{ s.t.}$$

³Note that each $e^j = f_i^j$ for some i and $e^k = f_i^k$ for some i .

$$pos(i^j, f^j) < pos(e^j, f^j) \text{ or } pos(i^j, f^j) > pos(e^j, f^j)$$

$$pos(i^k, f^k) < pos(e^k, f^k) \text{ or } pos(i^k, f^k) > pos(e^k, f^k), \text{ and}$$

$$sim(e^j, e^k) > sim(i^j, i^k)$$

PCR

PCR is a process to create many copies of a desired string or set of strings. It requires two substrings p_l, p_r (the primer pair) that exactly match the sequence flanking the region to be copied. The amplification of one set of strings (A - the amplification products) from another (T - the template set) can be described as⁴:

$$amp(T, p_l, p_r) \rightarrow A, A \subseteq T.$$

After the PCR is complete, each sequence in A will begin with p_l and end with p_r ,

$$\forall a \in A, pos(p_l, a) = 0, pos(p_r, a) = |a| - |p_r|,$$

and there should be no other detectable sequences in A.

2.3 Amplification of a gene family

In order to amplify a gene family (F), a primer pair (p_l and p_r) must be designed to flank each f^k (or a large portion of each f^k).

$$\forall f^k \in F, pos(p_l, f^k) \neq -1 \wedge pos(p_r, f^k) \neq -1.$$

Designing primers within the conserved regions of the family ensures they are present in each member. The search space can be reduced by considering only the exonic regions because they exhibit the highest levels of conservation within the gene family. This is easily accomplished by searching for primer pairs in the set of cDNA corresponding to the target gene family rather than the set of genomic sequence. For a given f^k , if $(f^k, c^i) \in TSC$, search c^i rather than f^k .

Maximal Common Substring

The only regions that can contain family specific primers are common to each member of the gene family. These regions correspond to the set of maximal common substrings [7] for the gene family. A string a_{max} is a maximal common substring of a set of strings F if a_{max} is present in each $f \in F$ and a_{max} satisfies the properties of right and left maximality: the characters to the left and right of a_{max} are not common across the entire set, i.e. a_{max} cannot be extended and remain a common substring.

Maximal common substrings are used because an appropriate primer sequence contained in a non-maximal substring is also contained in the maximal substring. The set of maximal common substrings is the set of all non-overlapping substrings in the set of sequences because overlap means the substrings can be joined and are therefore not maximal.

⁴Note that $|A| \leq |B|$ because A contains fewer unique sequences than B. The number of strings produced by PCR amplification may be greater than the number of strings in the initial set, however many of these strings will be duplicates.

If there are appropriate primer pairs for the gene family they will exist within the set of maximal common substrings (conserved regions). A pseudo-sequence p can be generated from the set of maximal common substrings M ,

$$\forall m^i \in M \text{ let } p = pm^i, \text{ s.t.}$$

$$\forall c^j \text{ pos}(m^i, c^j) < \text{pos}(m^{i+1}, c^j)$$

and used as input to a standard primer design tool.⁵ A discussion of algorithmic techniques for primer design is available [10]. The primers produced by the tool must amplify the whole family because the sequence from which primers were designed is present in each member of the set:

$$\text{amp}(S, p_l, p_r) \rightarrow F.$$

2.4 Amplification of a single locus

It is expected that there will be large detectable differences between only the intronic regions of locus specific sequences because:

1. Exonic regions are highly conserved within a gene family, so there should be no large differences in the exonic regions of related sequences.
2. Alleles are highly conserved across their entire sequence—because genes at the same locus experience similar rates of evolution—and should contain no large differences.
3. Intronic regions evolve independently from locus to locus because changes in these regions are unlikely to affect the gene product and distinct loci are subject to different environmental stresses causing different rates of evolution.

Based on the characteristic levels of conservation within regions of the sequences it should be possible to partition a family of related genes F into k sets L^1, \dots, L^k s.t. $\bigcup_i L^i = F$, where each L^i contains the alleles of one locus in this gene family. Sequences in a given set L^i should contain only small allelic differences while sequences from different sets will contain large differences due to the specific regions in each set.

Given a family of sequences F , \exists a partition $F' \subseteq F^*$ of sequences where for every set $F^j \in F' \exists$ a set of subsequences that is present for all $f \in F^j$. Let this set be denoted \bar{F}^j . That is

$$\forall f \in F^j, \forall f_i \in \bar{F}^j, \text{pos}(f_i, f) \neq -1.$$

Each subsequence in \bar{F}^j is unique to \bar{F}^j , i.e. each subsequence in \bar{F}^j is not present in any other string in F :

$$\forall f \in (F - F^j), \forall f_i \in \bar{F}^j, \text{pos}(f_i, f) = -1.$$

Furthermore the sequences within a set F^j will have higher similarity to each other than to sequences from any other string in $(F - F^j)$:

$$\forall f^i, f^k \in F^j, \forall f^m \in (F - F^j)$$

⁵Techniques for primer design are well known, and there are standard tools available to handle this task.

$$\text{sim}(f^i, f^k) > \text{sim}(f^i, f^m) \text{ and } \text{sim}(f^i, f^k) > \text{sim}(f^j, f^m)$$

which makes intuitive sense because of the regions of locus specific sequence in each partition.

The locus specific subsequences used to partition the gene family can also be used to create locus specific primers. Primer pairs designed within or overlapping the locus specific regions will amplify only the desired locus because this sequence is not present in any of the other loci. Regions of interest include the specific subsequences for the locus, each $f_i \in F^j$, and a small amount of flanking sequence, s_l and s_r , of length $\min(|p_l|, |p_r|) - 1$ for each f_i :

$$R = \{r_k | r_k = s_l f_i s_r\}_{\forall f_i}$$

Generating a pseudo-sequence p from the regions of interest:

$$\forall r_k \in R \text{ let } p = pr_k, \text{ s.t.}$$

$$\forall f \in F^j \text{ pos}(r^k, f) < \text{pos}(r^{k+1}, f)$$

will force a primer design tool to create primer pairs that will amplify this locus:

$$\text{amp}(F, p_l, p_r) \rightarrow F^j.$$

2.5 Amplification of a single allele

Alleles are expected to be highly conserved across their entire sequence because they are subject to the same environmental stresses—by virtue of being at the same chromosomal location—and should experience similar rates of evolution. However, they will generally contain small differences in the form of Single Nucleotide Polymorphisms (SNPs).

Consider two allelic sequences a^1 and a^2 . If the sequences contain SNPs they must be located between two maximal common substrings.

$$a^1 = a_i a_1 a_j,$$

$$a^2 = a_i a_j$$

where a_i, a_j are maximal common substrings, and a_1 is present only in a^1 . If $|a_1| = 1$, then a_1 is a SNP due to the insertion of a base in a^1 or the deletion of a base in a^2 .

Consider further:

$$a^1 = a_i a_1 a_j,$$

$$a^3 = a_i a_3 a_j$$

where $a_1 \neq a_3$ (this must be the case or a_i and a_j are not maximal). If $|a_1| = 1$ and $|a_3| = 1$ then a_1 and a_3 mark a substitution difference between a^1 and a^3 . If $|a_1| > 1$ and/or $|a_3| > 1$ all that can be said for certain is that a_1 and a_3 mark regions specific to a^1 and a^3 . Such regions should be rare in homologous alleles, and might signal the presence of alleles from another locus that were not filtered by the previous amplification.

After identifying the SNPs in a set of allelic sequences A , primers can be designed such that each primer in the pair is specific to the target sequence. For instance, primer pairs for sequences a^1, a^2, a^3 would include the primers p_l^1, p_l^2, p_l^3 s.t.:

$$\text{pos}(p_l^1, a^1) > \text{pos}(a_1, a^1) - |p_l^1| \text{ and}$$

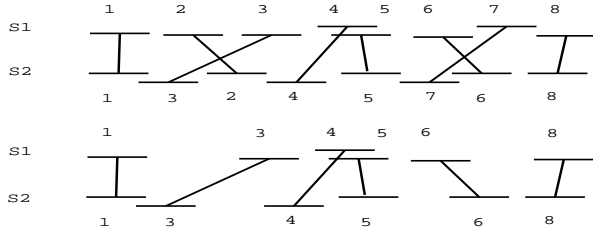


Figure 7: The MUMs in the top sequence have been numbered from 1 to 8, and the corresponding MUMs in the bottom sequence numbered accordingly. The LIS is the longest sequence of increasing numbers in sequence 2. The longest increasing sequence of MUMs in the above diagram is 1,3,4,5,6,8.

manageable pieces for Smith-Waterman alignment. This anchor based approach to alignment is similar to the approach used in the well known BLAST and FASTA alignment algorithms [13].

MGA

The second application of suffix trees to whole genome alignment comes in the form of Multiple Genome Aligner (MGA). This technique improves on MUMmer in three key ways (relevant to the current discussion):

- MGA is used for aligning multiple genomes. This is important as primer design will almost always be based on more than two sequences.
- MGA uses Maximal Exact Matches (MEMs) rather than Maximal Unique Matches (MUMs) ⁷ thereby creating additional anchor points as MUMs prevent repeated sequences from being used as anchors.
- MGA uses an $O(m \log m)$ time algorithm to return the ordered list of MEMs, whereas MUMmer uses a simpler dynamic programming algorithm requiring $O(m^2)$ time where $O(m)$ is the number of MEMs or MUMs.

MGA is significant as it demonstrates: the ability of suffix trees to extract conserved region from multiple sequences, the use of the LIS method to produce an ordered series of substrings from multiple sequences, a more efficient algorithm for producing the LIS; and it extracts additional conserved regions to be searched for valid primer pairs.

3.2 Prediction of Protein Families

Probabilistic Suffix Trees (PSTs) have been used to model protein families [2]. A PST is used to detect statistically significant, or non-random appearances of short segments common to many of the input sequences. These statistically significant segments reflect properties of the corresponding protein family, which can be used to classify new proteins. The PST is essentially a variable length Markov Model with performance similar to that of a carefully trained Hidden Markov Model (HMM).

⁷Recall that the uniqueness requirement has been relaxed in MUMmer2

The PST over an alphabet is a non-empty tree whose nodes vary in degree from 0 for a leaf to the size of the alphabet. Each edge in the tree is labelled by a single letter of the alphabet. Each node is assigned a probability vector, which is the probability of seeing character $\Sigma[i]$ after the label given by the node. PSTs differ from the classical suffix tree in the following way: In a suffix tree, the parent of a node "bra" is "br", whereas in a PST the parent of "bra" is "ra". In a suffix tree the parent is always the label minus the last character, and in a PST the parent is always the label minus the first character. The PST for a given input string is a subtree of the suffix tree associated with the reverse of the string.

Having used a PST to classify protein families, it should also be possible to use the PST to partition a super-family of proteins into the constituent families. This is enabled using a Variable Memory Markov Model (VMM) built using a PST for each group of proteins [1]. The technique involves letting the PSTs compete over the significant segments, and a deterministic annealing framework infers the number of underlying PST models corresponding to the subsets. This is directly applicable to locus specific primer design because the family of sequences must be first partitioned into locus specific subsets.

The measure of relatedness between a PST and a sequence segment is the probability that the PST predicts the segment. Initially the sequences are assigned to models depending on how well significant segments within them match the models. After the sequences are assigned each model is re-trained relative to the sequences it has been assigned. This process of set assignment and retraining is repeated until convergence (the strings assigned to each model remain constant). The final solution will depend on the number of models used, how the models were initialized, and how closely the sequences were required to match a model. To remove this dependency on the initial parameters, the algorithm applies an iterative approach to partitioning the sequences: An initial model over the whole set of proteins is trained until convergence at which point the model is split into two identical copies. These copies are randomly permuted so that they focus on slightly different subsets of the statistical segments, and then they are retrained to convergence. This process is repeated until new models do not contain enough meaningful information to identify any sequences in the set. At this point there should be one model corresponding to each statistically related group of sequences. Then a resolution parameter is introduced and the models are re-trained. The resolution parameter forces sequences to match the model more closely, and as the value is increased it creates a partition of the sequences such that each sequence is assigned to a unique model which identifies it. The use of a resolution parameter is intended to overcome the problem of local minima [11].

The VMM is found to outperform similarly trained HMMs, though it has problems with highly similar statistical sources and subsets specified by very short regions. This technique should be applicable to clustering locus specific regions as it is expected that there will be statistically significant differences between loci.

Given that suffix-tree techniques have been identified which can partition a family of sequences, it is now necessary to extract local specific regions from each of these partitions for use in primer design; a GST can be used for this. Nodes which cover all members of a locus will mark the end of a prefix common to all members of the locus. For instance, if node c has each member of locus L^1 as a descendant, then the label from the root to node c is common to all members of L^1 . If this region is maximal relative to L^1 , then it can be used to design a primer pair for L^1 . Node c common to all sequences from a given locus can be found using constant-time Lowest Common Ancestor (LCA) methods [5].

3.3 Selection of Signature Oligonucleotides

Signature oligonucleotides are short genomic (nucleotide) sequences which uniquely identify a single sequence in a collection. Selection of oligonucleotide probes requires an efficient method to locate unique subsequences. The PROBESEL application [9] performs this task using suffix trees and has been successfully applied to oligonucleotide selection in microarray design. Signature sequences are of interest for allele specific primer design because a signature region must contain a number of SNPs, and an allele specific primer pair must include one or more SNPs in each primer. PROBESEL has been applied to sequences up to 95% similar—1 SNP in 20 bases, which is the same degree of specificity required for allele specific primer design.

PROBESEL uses the Nearest Neighbour (NN) Thermodynamic Model to calculate the strength of interaction between each probe and each target in the collection. The NN model must know which bases are going to pair in the duplex, so each probe must be aligned with each target. Alignment is a computationally expensive operation so probes are pre-screened to minimize the number of alignments. Candidate probes are screened based on probe length, probe uniqueness (the ability of the probe to identify a single sequence), and probe melting temperature. Pre-screening is performed by pruning ineligible branches from the generalized suffix tree; for instance a branch which identifies more than one sequence is not specific and can be removed from the tree as can a branch which is shorter than the required probe length.

If primers are designed directly from the suffix tree rather than passing a pseudo-sequence to the standard primer design tools, pre-screening techniques become quite relevant as a way to reduce unnecessary computation. However, potentially more important than pre-screening would be the use of suffix trees to speed computation of the Dynamic Programming matrix for sequence alignment. All repetitive and common regions of the sequences can be located in linear time within the suffix tree, and portions of alignments which share prefixes with another previously computed alignment need not be recomputed. Use of the suffix tree to aid alignment has been shown to speed the computation by a factor of five [9]. It will be interesting to see whether a similar speedup is achieved when designing primers directly from the suffix tree, and whether there is any benefit (in terms of computation speed) to designing primers from the suffix tree rather than passing a sequence to the standard tools.

4. RESULTS

Our preliminary experiments with selecting regions for family specific primer design using GSTs have shown encouraging results as the technique was successfully applied, using a prototype based on the GST source from PROBESEL, to primer design for three different gene families. A GST was created for each of the three families, conserved regions were selected using the technique described in MGA to select MEMs, and known primers for each of the families were found within the MEMs. However several difficulties were encountered:

1. Some of the cDNA were not full length—they didn't span the entire length of the genomic sequence. There are many reasons the sequencing reaction will fail to produce the full cDNA sequence, and this is an expected occurrence. Unfortunately it made it impossible to determine all the maximal common substrings for the entire set directly from the raw cDNA.
2. The cDNA sequences are not always in the same orientation. This resulted in conserved regions between two sequences going undetected because one sequence is the reverse or reverse-complement of the other.

Fortunately these two problems were easily overcome:

1. The first problem was overcome by forming contigs from the cDNA sequences in each family. A contig is formed by assembling sequences with overlapping subsequences to form a larger consensus sequence. However, there is a danger that creating contigs will allow allele or locus specific regions to be included in the pseudo-sequence used to design the primers because contig-ing may mask differences in the final sequence or replace ambiguous bases with the most frequently occurring character. We are currently examining the parameters of the contig-ing algorithm to minimize this possibility.
2. The second problem was overcome by comparing each sequence to a reference sequence and reorienting it as necessary. This is relatively efficient because the family was initially determined by comparing the set of cDNA to the reference sequence, and the sequences need not be reoriented as it is enough to reverse-complement the sequence when adding it to the GST.

Tests related to locus specific and allele specific primer design are currently in the planning stages.

5. CONCLUSIONS AND FUTURE WORK

A formalism based on strings and sets has been developed to precisely describe allele specific primer design. This formalism ensured that the biological problem was properly understood, and allowed parallels to be drawn between the primer design problem and related bioinformatics problems facilitated by suffix trees. The problems of whole genome alignment, prediction of protein families and selection of signature oligonucleotides each include important aspects of the primer design problem: selection of conserved regions

from a set of sequences, division of a superset, and the selection of sequence specific subsequences.

Similarities between the alleles specific primer design problem and each of the related problems support the hypothesis that suffix trees will facilitate an efficient solution to the primer design problem. Initially suffix trees will be used to select appropriate regions for primer design and to cluster locus specific sequences. This approach has been successfully applied to family specific primer design, and the formal specifications of locus specific and allele specific primer design will guide implementation of these subproblems.

Important considerations for future work include: 1) Can contigs be formed from cDNA and genomic sequence in such a way that regions conferring specificity are not lost in the consensus? 2) Will the loci be sufficiently divergent for the VMM/PST method to properly classify them? 3) Can primers be designed more efficiently within the suffix tree than using the standard tools?

6. ACKNOWLEDGEMENTS

Thanks to my supervisors, Tony Kusalik and Isobel Parkin for editing and guidance during the preparation of this document. Thanks to Chad Stratilo and Andrew Sharpe for comments, feedback and discussion of the allele specific primer design problem. Thanks to Lars Kaderali who provided the suffix tree source from PROBESEL for experiments.

7. REFERENCES

- [1] G. Bejerano, Y. Seldin, H. Margalit, and N. Tishby. Markovian domain fingerprinting: statistical segmentation of protein sequences. *Bioinformatics*, 17(10):927–934, 2001.
- [2] G. Bejerano and G. Yona. Variations on probabilistic suffix trees: statistical modeling and prediction of protein families. *Bioinformatics*, 17(1):23–43, 2001.
- [3] A. Delcher, S. Kasif, R. Fleischmann, J. Peterson, O. White, and S. Salzberg. Alignment of whole genomes. *Nucl. Acids. Res.*, 27(11):2369–2376, 1999.
- [4] A. L. Delcher, A. Phillippy, J. Carlton, and S. L. Salzberg. Fast algorithms for large-scale genome alignment and comparison. *Nucl. Acids. Res.*, 30(11):2478–2483, 2002.
- [5] D. Gusfield. *Algorithms on Strings, Trees, and Subsequences*. Cambridge University Press, University of California, Davis, 1997.
- [6] S. Haas, M. Vingron, A. Poustka, and S. Wiemann. Primer design for large scale sequencing. *Nucl. Acids. Res.*, 26(12):3006–3012, 1998.
- [7] D. S. Hirschberg. A linear space algorithm for computing maximal common subsequences. *Communications of the ACM*, 18(6):341–343, 1975.
- [8] M. Hohl, S. Kurtz, and E. Ohlebusch. Efficient multiple genome alignment. *Bioinformatics*, 18(90001):312S–320, 2002.
- [9] L. Kaderali and A. Schliep. Selecting signature oligonucleotides to identify organisms using DNA arrays. *Bioinformatics*, 18(10):1340–1349, 2002.
- [10] T. Kampke, M. Kieninger, and M. Mecklenburg. Efficient primer design algorithms. *Bioinformatics*, 17(3):214–225, 2001.
- [11] S. Z. Li. *Markov Random Field Modeling in Computer Vision*. Springer Verlag, 1995.
- [12] M. J. McPherson, P. Quirke, and G. R. Taylor, editors. *PCR 1 A Practical Approach*. Information Press Limited, Oxford, England, 1991.
- [13] D. W. Mount. *Bioinformatics Sequence and Genome Analysis*. Cold Spring Harbor Laboratory Press, 2001.
- [14] P. H. Raven and G. B. Johnson. *Understanding Biology (third edition)*. Wm. C. Brown Publishers, 1995.