Bioinformatics Methods for Diagnosis and Treatment of Human Diseases

Jorge Alexander Duitama Castellanos

University of Connecticut, 2010

The availability of large databases of genomic information has enabled research efforts focused on refining methods for diagnosis and treatment of human diseases. However, proper use of genomic databases can not be achieved without the development of sophisticated data analysis methods, which is by itself a challenging task due to the size and heterogeneity of the data. The focus of the research proposed in this document is on developing computational methods and software tools for diagnosis and treatment of human diseases.

We describe a primers design tool for rapid virus subtype identification, applied to Avian Influenza called PrimerHunter, which takes as input sets of both target and non-target sequences and select primers that efficiently amplify any one of the targets, and none of the non-targets. PrimerHunter ensures the desired amplification properties by using accurate estimates of melting temperature with mismatches, computed based on the nearest-neighbor model via an efficient fractional programming algorithm

We also present a bioinformatics pipeline for detection of immunogenic cancer mutations by high throughput mRNA sequencing. As part of this pipeline, we developed and integrated novel algorithms and strategies for mRNA reads mapping, SNV detection, genotyping and haplotyping. We show through validations on real data that our methods improve accuracy to identify expressed mutations over existing methods and that our haplotyping algorithm is more efficient than other solutions with comparable accuracy levels. Our pipeline predicted more than a thousand candidate epitopes for six different mouse cancer tumor cell lines, which are currently used to find stable protocols for immunotherapy.

Bioinformatics Methods for Diagnosis and Treatment of Human Diseases

Jorge Alexander Duitama Castellanos

M.Sc., Universidad de los Andes, Bogotá, Colombia, 2004

A Dissertation

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Doctor of Philosophy

at the

University of Connecticut

2010

APPROVAL PAGE

Doctor of Philosophy Dissertation

Bioinformatics Methods for Diagnosis and Treatment of Human Diseases

Presented by

Jorge Alexander Duitama Castellanos

Major Advisor: _____

Ion Măndoiu

Associate Advisor: _____

Yufeng Wu

Associate Advisor: _____

Sanguthevar Rajasekaran

University of Connecticut
2010

Acknowledgments

I will start by saying thanks to my advisor professor Ion Măndoiu for all his support and guidance during these years. Professor Măndoiu is a full time committed educator and an outstanding researcher. Working with him I have been able to acquire not just the knowledge needed to work as a bioinformatician but also the skills and discipline needed to conduct a successful research project.

I would like to thank my associate advisors Yufeng Wu and Sanguthevar Rajasekaran and also the researchers that I had the privilege to collaborate with during these years. Thanks to professors Craig Nelson, Mazhar Khan and Pramod Srivastava, to doctors Fiona Hyland and Dumitru Brinza from Life Technologies and to doctors Thomas Huebsch, Gayle McEwen, Eun-Kyung Suk and Margret Hoehe from the Max Planck Institute for Molecular Genetics. Thanks also to Justin Kennedy, James Lindsay, Bogdan Paşaniuc, Xiguang Yan, Jin Jun, Dipu Kumar and Edward Hemphill.

I will always be thankful with my friends from the group Jarana for all the good and bad things that we went through during these years. Thanks also to my new friends from all over the world for all what I have learned. Thanks to my friends back home who always supported me. Special thanks to Andres Perez, Manuel Hernandez, Juan Vivas, Hernan Duarte, and Jorge Osorio. Finally, thanks to my grandparents, uncles, aunts, cousins and everybody in the family Castellanos Forero for all their inconditional support. Special thanks to Pedro Tulio

Castellanos up in heaven.

This thesis is dedicated to my mother Martha Castellanos. Her lifetime sacrifice is the main reason why I am able to finish this work. For this, all what I am and all what I will be, gracias mamá.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Research efforts during the last two decades have provided huge amounts of genomic information for almost every form of life [8, 74, 20, 77]. The availability of this information has enabled a deeper understanding on the behavior of studied organisms. Indeed, much of this research effort is focused on refining methods for diagnosis and treatment of human diseases [22, 75]. However, proper use of genomic databases can not be achieved without the development of sophisticated data analysis methods, which is by itself a challenging task due to the size and heterogeneity of the data. The focus of the research consigned in this document is on developing computational methods and software tools for diagnosis and treatment of human diseases. In particular, we describe a primers design tool for rapid virus subtype identification, applied to Avian Influenza and a bioinformatics pipeline for detection of immunogenic cancer mutations by high throughput mRNA sequencing.

Rapid and reliable virus subtype identification is critical for accurate diagnosis of human infections, effective response to epidemic outbreaks, and global-scale surveillance of highly pathogenic viral subtypes such as avian influenza H5N1. The Polymerase Chain Reaction (PCR) has become the method of choice for virus sub-

type identification. However, designing subtype specific PCR primer pairs is a very challenging task: on one hand, selected primer pairs must result in robust amplification in the presence of a significant degree of sequence heterogeneity within subtypes, on the other, they must discriminate between the subtype of interest and closely related subtypes [32, 90].

We present a new tool, called PrimerHunter, that can be used to select highly sensitive and specific primers for virus subtyping. Our tool takes as input sets of both *target* and *non-target* sequences. Primers are selected such that they efficiently amplify any one of the target sequences, and none of the non-target sequences. PrimerHunter ensures the desired amplification properties by using accurate estimates of melting temperature with mismatches, computed based on the nearest-neighbor model via an efficient fractional programming algorithm. Validation experiments with 3 Avian influenza HA subtypes confirm that primers selected by PrimerHunter have high sensitivity and specificity for target sequences.

Genomes research has been boosted by recent advances in *high-throughput sequencing* (HTS) technologies such as the Roche 454, Illumina, ABI SOLiD, and Helicos HeliScope. This advances have led to orders of magnitude higher throughput compared to classic Sanger sequencing (see [37] for a review). Indeed, at full capacity each one of these HTS sequencers is capable of producing approximately 1Gb of read data per day. Coupled with continuously decreasing prices, HTS has profoundly transformed genomics research, enabling a host of novel HTS applications like individual genome resequencing [86] and deep sequencing of mRNA [61, 64]. This last application is of particular interest for cancer research because a deeper understanding on human transcriptomes can lead to a better differentiation between normal and cancer cells. In particular, we expect to show that mRNA sequencing can improve methods for immunotherapy, which is a promising cancer treatment approach that relies on awakening the immune system to the presence

of antigens associated with tumor cells. The success of this approach depends on the ability to reliably detect immunogenic cancer mutations, the vast majority of which are expected to be tumor-specific [76].

Standard analysis of RNA-seq data includes reads mapping to a reference genome and variants discovery. We studied both problems and contributed with novel strategies and models for these tasks. We map mRNA reads against both a reference genome [77] and a database of consensus coding sequences (CCDS) [74] and defined a set of rules to combine mapping results. We also implemented a bayesian model for Single Nucleotide Variants (SNV) detection and genotyping that chooses the genotype with highest posterior probability based on counts and base quality scores. Validation against known SNP genotypes of a Hapmap individual shows that the combined mapping strategy yields improved SNV genotyping accuracy compared to performing genome or CCDS mapping alone. We also show that our bayesian model for genotyping achieved improved accuracy for our testing dataset compared with other widely used bayesian methods as Maq [53] or SOAPsnp [54]. We finally discuss how different data curation and filtering strategies affect sensitivity and specificity.

As a tumor accumulates more and more mutations, phasing of close SNVs becomes critical to achieve an accurate prediction of translation to proteins. Since common statistical methods for SNPs phasing like fastPHASE [84] are not suitable for this settings, we studied the single individual haplotyping problem [78] and we proposed a heuristic algorithm to solve it. We defined a new problem formulation which tries to find the haplotype origin of each fragment before assembling the haplotypes and then we reduced our formulation to the well known Max-Cut problem. We performed a comprehensive set of simulation experiments to show that our algorithm is significantly faster than other heuristic methods like HapCUT [6] achieving the same accuracy.

We combined standard and novel techniques in a bioinformatics pipeline for detection of immunogenic cancer mutations from high throughput mRNA sequencing data. We map reads, discover and phase SNVs following the techniques described above and we also integrated Primer3 [42, 80] to find primers to validate mutations experimentally by performing PCR experiments on genomic DNA. Finally, after predicting how phased SNVs affect translation to proteins, we integrate two common epitope prediction tools as NetMHC [57] and SYFPEITHI [75]. We tested the whole pipeline on MethA cell lines and spontaneous prostate cancer tumor samples and we found that our pipeline is able to detect tens of immunogenic mutations for each dataset. We also were able to find cases of variants close enough to each other to make phasing relevant and in which the coocurrance of two close mutations in a single haplotype increases the translated peptides score according with the NetMHC predictions. Epitopes found for these datasets are currently under experimental validation.

This document is organized as follows: In Chapter 2 we introduce and formalize the discriminative primers selection problem, we describe the algorithm implemented in PrimerHunter, and we show validation results on experiments with Avian Influenza subtypes. In Chapter 3 we present our improved mapping and SNV genotyping strategies, and we present accuracy results under different settings from experiments with publicly available mRNA reads. In Chapter 4 we present our novel single individual haplotyping algorithm, including results of simulations under different settings. In Chapter 5 we propose an immunotherapy approach for treatment of cancer tumors which relies on detection of immunogenic cancer mutations. We combine all previous methods to present the pipeline deveolped to find such mutations from high throughput mRNA sequencing data, and we show results and validation procedures for identified mutations. Finally, we summarize the current status and present directions for future work in Chapter 6.

Chapter 2

Primer Design for Virus Subtype Identification[1]

RNA viruses such as avian influenza, hepatitis C virus, and human immunodeficiency virus are characterized by a extensive genetic heterogeneity, primarily due to the lack of proofreading mechanisms in their RNA polymerase. As a result, most RNA viruses can be subdivided into distinct taxonomic subunits referred to as genotypes or subtypes. For example, over 100 avian influenza subtypes have been identified in wild birds as the result of independent assortment of 16 subtypes of the RNA segment encoding the Haemagglutinin (HA) protein with 9 subtypes of the segment encoding for Neuraminidase (NA). Rapid virus subtype identification is critical for accurate diagnosis of human infections, effective response to epidemic outbreaks, and global-scale surveillance of highly pathogenic subtypes such as avian influenza H5N1 [22].

The Polymerase Chain Reaction (PCR) has become the method of choice for virus subtype identification, largely replacing traditional immunological assays due to its high sensitivity and specificity, fast response time, and affordable cost [90].

[1]The results presented in this chapter are based on joint work with D.M. Kumar, E. Hemphill, M. Khan, I.I. Măndoiu and C.E. Nelson [26]

However, designing subtype specific PCR primer pairs is a very challenging task [32]: on one hand, selected primer pairs must result in robust amplification in the presence of a significant degree of sequence heterogeneity within subtypes, on the other, they must discriminate between the subtype of interest and closely related subtypes.

Unfortunately, existing primer design tools are not well suited for designing PCR primers for subtype identification. Commonly used packages such as Primer3 [42, 80] seek to amplify a single known target nucleic acid sequence, and cannot guarantee amplification sensitivity in the presence of high sequence heterogeneity within a subtype. A widely-used approach to primer design for virus identification relies on first constructing a "consensus gestalt" from a multiple alignment of target virus sequences [30]. After masking regions that also appear in the genome of related viruses, remaining "unique" regions are mined for primers using standard tools such as Primer3. This approach can be quite successful at finding *species-specific* primers, since virus genomes often include highly conserved genes and non-coding regions that serve critical roles in replication, transcription, and packaging. However, the approach has limited applicability when the goal is to discriminate between virus subtypes, since most highly conserved regions are shared by all subtypes. The same limitation applies to several suffix-tree based algorithms [4, 28, 73] that search for long substrings that appear exactly or with a small number of mutations in all (or a large percentage) of the sequences of a given target set, and in none of the sequences of a given non-target set.

Another common approach to ensuring amplification of heterogenous sets of nucleic acid sequences is the use of primers with degenerate bases. Several methods have been proposed for selecting degenerate primers, including various greedy algorithms [5, 56, 87] and heuristics based on multiple alignments of nucleic acid [38] and protein sequences [99]. Unfortunately, all these methods ignore primer

specificity (i.e., preventing amplification of related virus subtypes) which prevents their use for direct viral subtyping assays.

A comparison of the main features provided by a selection of most relevant existing primer and probe selection tools [28, 31, 38, 41, 73, 79, 80, 96, 99, 106] is presented in Table 2.1. As it can be seen from the table, most existing tools miss key features that make them inappropriate for use in designing PCR primers for virus subtyping. Of the surveyed methods, only OligoSpawn [106] and SLICSel [96] were successful at finding subtype specific probes when run on a large set of avian influenza HA sequences. The other methods were either not available, could not handle multiple target/non-target sequences, or simply did not find any subtype specific primers or probes.

Design Tool	**Multiple Targets**	**Non Targets**	T_M **Model**	**Salt Correction**	**Output**
Primer3 [80]	No	Yes (DB)	NN	Yes	Multiple primer pairs
Insignia [73]	Yes (DB)	Yes (DB)	None	No	Multiple signatures
QPrimer [41]	No (DB)	No	NN	No	Multiple primers
DePict [99]	Yes (MSA)	No	None	No	Best primer
PROBEMer [28]	Yes	Yes	NN	No	Multiple probes
Greene SCPrimer [38]	Yes (MSA)	No	NN	Yes	Multiple primer pairs
OligoSpawn [106]	Yes	Yes	NN	No	Multiple probes
SLICSel [96]	Yes	Yes	NN	Yes	Multiple probes
Primaclade [31]	Yes (MSA)	No	NN	No	Multiple primers
OligoArray [79]	Yes	Yes (DB)	NN	No	Multiple probes
PrimerHunter	Yes	Yes	NN w/ mismatches	Yes	Multiple primer pairs

Table 2.1: Features comparison between primer and probe selection tools most similar to PrimerHunter. (DB: user can select targets from a pre-constructed database; MSA: input must be provided as a multiple sequence alignment; NN: nearest-neighbor model)

We present a new tool, called PrimerHunter, that can be used for selecting highly sensitive and specific primers for virus subtyping and is likely to find applications in other contexts that require discriminative probes/primers. As in [28, 73, 106], our tool takes as input sets of both *target* and *non-target* sequences. To guarantee high sensitivity, primers are selected such that they efficiently amplify any one of the target sequences representing different isolates of the subtype of interest. High specificity is ensured by requiring that none of the non-target sequences be amplified by selected primers; non-targets typically being sequences representing isolates of closely related virus subtypes. Unlike previous methods, which restrict the primer search space to the set of substrings shared by all target sequences or to highly conserved regions in a multiple alignment, PrimerHunter achieves a higher design success rate by generating an exhaustive set of candidate primers from the target sequences and using accurate melting temperature computations to ensure the desired amplification/non-amplification properties. Melting temperature computation is performed based on the state-of-the-art nearest-neighbor model of [82]. Of critical importance in selective target amplification is accurate prediction of primer-template hybridization with mismatches. Melting temperature with mismatches is efficiently computed in PrimerHunter by using the fractional programming approach of [49], modified to incorporate the salt correction model of [82].

PrimerHunter has been used to design specific primer pairs for all avian influenza HA and NA subtypes from complete sequences of North American origin in the NCBI flu database [8]. Validation experiments confirm that primers selected by PrimerHunter are both specific and robust in the PCR amplification of target sequences. The PrimerHunter web server, as well as the open source code released under the GNU General Public License, are available at `http://dna.engr.uconn.edu/software/PrimerHunter/`.

2.1 Problem Formulation

Unless stated otherwise, we assume that all sequences are over the DNA alphabet, $\{A, C, G, T\}$, and are given in 5′-3′ orientation. For a sequence s, we denote by $|s|$ its length, and by $s(l, i)$ the subsequence of length l ending at position i, i.e., $s(l, i) = s_{i-l+1} \ldots s_{i-1} s_i$. We denote by $T(p, t, i)$ the melting temperature of the duplex formed by a primer p and the Watson-Crick complement of $t(|p|, i)$. In order to ensure sensitive amplification of target sequences, we require for each selected primer p to have at least one position i within each target t such that $T(p, t, i)$ is greater than or equal to a user specified threshold T^{min}_{target}. Since mismatches at the 3′ end of the primer can significantly reduce amplification efficiency [44], we additionally require that the 3′ end of p match perfectly $t(|p|, i)$ at a set of bases specified using a 0-1 *perfect match* mask M. For example, a mask $M = 3'$-1101-$5'$ specifies that the first, second, and fourth 3′-most bases of the primer must be matched exactly. For a primer p and a target sequence t, we denote by $\mathcal{I}(p, t, M)$ the set of positions i of t at which the 3′ end of p matches $t(|p|, i)$ according to M. Thus, in order to ensure sensitive PCR amplification of target sequences, we require that a selected primer p have, for every target t, at least one position $i \in \mathcal{I}(p, t, M)$ for which $T(p, t, i) \geq T^{min}_{target}$.

To avoid non-specific amplification, we further require for each selected primer to have a melting temperature $T(p, t, i)$ below a user specified threshold $T^{max}_{nontarget}$ at every position i of every non-target sequence t. The problem of selecting target-specific forward PCR primers is therefore formulated as follows:

Discriminative Primer Selection Problem (DPSP)

Given: Sets $TARGETS$ and $NONTARGETS$ of 5′-3′ DNA sequences, perfect match mask M, melting temperature thresholds T^{min}_{target} and $T^{max}_{nontarget}$, and constraints on primer length, GC content, self-complementarity, etc.

Find: Primers p satisfying given constraints on primer length, GC content, self-complementarity, etc., such that:

- For every $t \in TARGETS$, there exists $i \in \mathcal{I}(p, t, M)$ such that $T(p, t, i) \geq T^{min}_{target}$, and
- For every $t \in NONTARGETS$, $T(p, t, i) \leq T^{max}_{nontarget}$ for every $i \in \{|p|, \ldots, |t|\}$.

2.2 Melting Temperature Calculation

PrimerHunter estimates the melting temperature of primer-target and primer-nontarget duplexes using the nearest-neighbor model of [82], which is considered to be the most accurate melting temperature model to date [71]. However, unlike most other primer design packages, which only require estimates of the melting temperature between a primer and its perfectly complementary template, PrimerHunter critically relies on accurate estimates of the melting temperature for non-complementary duplexes. This requires finding the optimum thermodynamic alignments for all evaluated duplexes, i.e., the alignments with minimum Gibbs free energy. As in [49], optimum alignments are computed using the fractional programming algorithm of [25]. In this section we describe our modification of the algorithm to incorporate SantaLucia's correction for the concentration of salt cations in the PCR mix [82]. As shown below, incorporating this correction yields significantly improved estimates compared to [49].

In SantaLucia's nearest-neighbor model [82], the melting temperature of a specific alignment x between a 5′-3′ primer p with concentration c_p and a 3′-5′ template t with concentration c_t is given by

$$T_M(x) = \frac{\Delta H(x)}{\Delta S(x) + 0.368 \times N/2 \times \ln(\mathrm{Na}^+) + R \times \ln(C)} \tag{2.1}$$

where $\Delta H(x)$ and $\Delta S(x)$ are enthalpy and entropy changes for the annealing reaction resulting in a duplex with Watson-Crick pairings given by alignment x, N is the total number of phosphates in the duplex, R is the gas constant, C is the total DNA concentration calculated as $c_p - c_t/2$ if $c_p > c_t$ and $(c_p/2)$ if $c_p = c_t$ [82], and Na^+ is the the concentration of salt cations. For a given alignment x the enthalpy and entropy changes $\Delta H(x)$ and $\Delta S(x)$ are computed by summing experimentally estimated contributions of constitutive dimer duplexes (including internal mismatches and gaps), with additional terms for duplex initiation/termination and (when applicable) symmetry correction.

The melting temperature between p and t is given by the most stable alignment x, i.e., it is taken to be the maximum $T_M(x)$ over all possible alignments x. This maximum can be found using Dinkelbach's fractional programming algorithm [25], which relies on a simple iterative procedure to maximize the ratio between two functions when linear combinations of the two functions can be maximized efficiently. More specifically, given a finite set S and two functions $f, g : S \to \mathbb{R}$ with $g > 0$, the maximum ratio $t^* = \max_{x \in S} \frac{f(x)}{g(x)}$ can be approximated arbitrarily close via the following algorithm:

1. Choose $t_1 \leq t^*$; $i \leftarrow 1$
2. Find $x_i \in S$ maximizing $F(x) := f(x) - t_i g(x)$
3. If $F(x_i) \leq \varepsilon$ for some tolerance $\varepsilon > 0$, output t_i
4. Else, set $t_{i+1} \leftarrow f(x_i)/g(x_i)$ and $i \leftarrow i + 1$, and then go to step 2

As shown by Dinkelbach, this algorithm produces values $t_1 < t_2 < t_3 < \ldots$ converging to t^*. When using Dinkelbach's algorithm to maximize (2.1) over the set of alignments x, the function to be maximized in Step 2 is $-\Delta G(x) = t_i[\Delta S(x) + (0.368) \times N/2 \times \ln(Na^+) + R \times \ln(C)] - \Delta H(x)$. Since $-\Delta G(x)$ is

additively decomposable, the alignment x maximizing it can be found efficiently by a standard dynamic programming algorithm, similar to [49]. As shown in [49], the algorithm typically converges in a small number of iterations.

2.3 Algorithm

PrimerHunter works in two stages: in the first stage forward and reverse primers are selected according to the problem formulation given above, while in the second stage feasible primer pairs are formed using the primers selected in first stage.

The first stage starts with a preprocessing step that builds a hash table storing all occurrences in the target sequences of "seed" nucleotide patterns consistent with the given mask M. This is done by aligning the mask M at every position i of every target sequence t, and storing in the hash table an occurrence of the seed pattern created by extracting from $t(|M|, i)$ the nucleotides that appear at positions aligned with the 1's of M. For example, if $M = 3'$-1101-$5'$ and $t(4, i) = 5'$-$GATC$-$3'$, we store in the hash table an occurrence of seed GTC at position i of t.

Once the hash table is constructed, candidate primers are generated by taking substrings with lengths within a user-specified interval $[l_m, l_M]$ from one or more of the target sequences. Similar to the Primer3 package [80], PrimerHunter filters the list of primer candidates by enforcing user-specified bounds on GC Content, $3'$-end GC clamp, maximum number of consecutive mononucleotide repeats, and self-complementarity. For each surviving candidate p, PrimerHunter uses the hash table to recover for each target t the list $\mathcal{I}(p, t, M)$ of positions at which p matches t according to M. It then computes the melting temperature of p with the Watson-Crick complement of t at each of these positions, retaining p only if $\max_{i \in \mathcal{I}(p,t,M)} T(p, t, i) \geq T^{min}_{target}$. Finally, PrimerHunter computes the maximum

melting temperature between p and the Watson-Crick complements of non-target sequences, retaining p only if $\max_{i\in\{|p|,\ldots,|t|\}} T(p,t,i) \leq T^{max}_{nontarget}$ for every non-target sequence t.

The above process is repeated on the reverse complements of target and non target sequences to generate reverse primers. Then, in the second stage of the algorithm, the lists of selected forward and reverse primers are used to create feasible primer pairs by enforcing the following constraints:

1. Product length: for each target sequence the total product length must fall between user specified bounds.

2. Melting temperature similarity: for every target sequence, the difference between the maximum and the minimum melting temperature of the two primers must not exceed a user defined value.

3. Primer dimers: a criteria similar to that used for preventing primer self-complementarity is used to avoid hybridization between the two primers of the pair; the test is identical to that implemented in Primer3 [80].

2.4 Algorithm Extensions

Since degenerate bases at specific primer positions yield perfect matches at these positions regardless of target variability, the use of degenerate primers is an effective technique for ensuring robust amplification of heterogenous targets. However, degenerate primer design is a difficult problem due to the large space from which degenerate primers can be selected [5, 56, 87]. To overcome this difficulty, we adopted a simple *pattern-based* approach to degenerate primer design, based on the observation that most of a virus' sequence is coding for proteins and that the vast majority of sequence heterogeneity is observed at synonymous positions.

PrimerHunter uses a user-specified *degeneracy mask*, specifying the positions at which fully degenerate nucleotides should be incorporated in candidate primers. Formally, the degeneracy mask is a vector D of integers 1 or 4 in $3'$ to $5'$ orientation. In each position i where $D_i = 4$, a degenerate base N will be included in every primer. For example, if $D = 3'$-114114-$5'$, every primer will end with the pattern $5'$-NxxNxx-$3'$. A degeneracy mask may be used in conjunction with a complementary perfect match mask ($M = 3'$-110110-$5'$ for the above D), although this is not required. The only required change to the primer selection algorithm is in the computation of melting temperatures: the range of melting temperatures fBioinformatics Methods for Diagnosis and Treatment of Human Diseasesor a degenerate primer is obtained by computing the melting temperatures against the given template for all compatible non-degenerate primers.

For target sets exhibiting a very high degrees of heterogeneity, or for overly stringent design constraints, it may be impossible to find specific primer pairs that amplify all targets. When detecting this situation, PrimerHunter automatically seeks and reports a small set of primer pairs that collectively amplify all targets. The set of pairs is constructed using the classic greedy set cover algorithm [16, 39], where the elements to be covered are target sequences and the sets correspond to pairs of compatible primers that amplify at least one of the target sequences and none of the non-targets. From the well-known approximation guarantee in [16, 39] it follows that the greedy algorithm yields a number of primer pairs within a factor of $1 + \ln m_t$ of optimum for m_t target sequences.

When multiple primer pairs are needed to cover all targets, the number of primer pairs can be further reduced by relaxing the constraint that forward and reverse primer candidates must amplify all targets. As in [28], this is achieved in PrimerHunter by specifying a minimum percentage of target sequences to which selected primers must hybridize. Similarly, the non-targets filtering can be re-

laxed, allowing selected primers to hybridize to a small percentage of non-targets. However, to maintain specificity, primer pairs that feasibly amplify one of the non-target sequences are discarded before running the greedy set cover algorithm.

2.5 Results

2.5.1 Accuracy of Melting Temperature Predictions

We compared the accuracy of estimates obtained based on (2.1) to those obtained as in [49] by using a simplified formula that does not include the salt correction term $0.368 \times N/2 \times \ln(\mathrm{Na}^+)$ in the denominator. Figure 2.1 shows the mean and standard deviation of the difference between the melting temperature determined experimentally and that predicted by the two models for a set of 812 duplexes of perfectly complementary oligonucleotides with lengths between 9 and 30 base pairs, GC content between 8% and 80%, and salt concentrations between $0.069M$ and $1.02M$ [68, 71]. The data has been stratified in four categories of salt concentration, with ranges given in Table 2.2. Table 2.2 also includes the Mean Squared Error (MSE) for each model and each salt concentration category. The results show that predictions given by (2.1) have much lower MSE values for all salt concentration categories except $1-1.02M$. Although the two models result in identical predictions at 1M concentration, for salt concentrations larger than $1M$ applying the salt correction produces slightly worse estimates. The difference between the two models is statistically significant: within each salt concentration category the null hypothesis that prediction errors of the two models have the same mean is rejected by the Wilcoxon signed-rank test with a p-value smaller than 10^{-16}.

Since duplexes involving primers with atypical length or GC content could potentially skew the results, we repeated the above comparison by considering only duplexes consisting of primers with length between 20 and 25 base pairs and GC

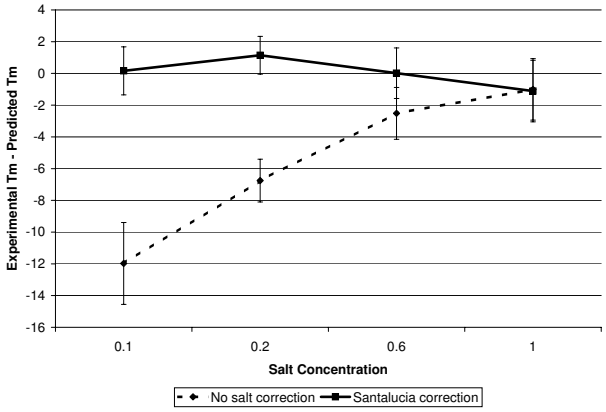


Figure 2.1: Average and standard deviation of the difference (in degrees Celsius) between experimental melting temperatures and predictions obtained by fractional programming without salt correction [49] and with salt corrections performed using the SantaLucia model (2.1) for 812 duplexes of perfectly complementary oligonucleotides with lengths between 9 and 30 base pairs and GC content between 8% and 80%

	Primer length $9-30$ GC content $8\%-80\%$			Primer length $20-25$ GC content $25\%-75\%$		
Salt Conc. (M)	# duplexes	MSE w/o salt correction	MSE with salt correction	# duplexes	MSE w/o salt correction	MSE with salt correction
$0.069-0.15$	351	150.03	2.30	158	148.91	2.25
0.22	152	47.44	2.71	72	43.14	3.18
$0.62-0.621$	152	8.98	2.52	72	6.90	1.38
$1-1.02$	157	4.75	4.97	74	2.61	2.76

Table 2.2: Mean Squared Error (MSE) for residuals calculated as the difference (in degrees Celsius) between experimental melting temperatures and predictions obtained by fractional programming without salt correction [49] and with salt corrections performed using the SantaLucia model (2.1).

content between 25% and 75%, which are typical values used in primer design and the default ranges for PrimerHunter. The results shown in Supplementary Figure 1 and Table 2.2 show that the predictions given by (2.1) remain more accurate than predictions based on [49] for salt concentrations below $1M$ even when disregarding primers with extreme GC-content or length. In all categories, the null hypothesis that prediction errors of the two models have the same mean is still rejected by the Wilcoxon signed-rank test, with a p-value smaller than 10^{-14}.

Unfortunately, experimental data on melting temperature of duplexes with mismatches is much more limited. We could collect only 110 duplexes with one mismatch and 28 duplexes with two mismatches from [1, 2, 3, 72]. Duplexes with one mismatch have lengths between 9 and 16 base pairs and GC content between 21% and 78%, while duplexes with two mismatches have lengths between 12 and 14 base pairs and GC content between 50% and 75%. Except for twelve duplexes with one mismatch, the melting temperature of all these duplexes was experimentally calculated at $1M$ of salt concentration. Since both prediction models produce exactly the same answer for a salt concentration of $1M$, we did not have enough information to compare them for duplexes with mismatches. Table 2.3 gives the mean and standard deviation for the prediction errors made by the SantaLucia model (2.1). The results suggest that, although less accurate than in the case of perfectly complementary duplexes, melting temperature estimates for duplexes with mismatches still provide good approximations. (We have also implemented the salt correction model of [68], but found the SantaLucia model to be slightly more accurate.).

2.5.2 Design Success Rate

Primer Hunter has been implemented in C++ on a standard Linux platform. We designed primer pairs for 14 HA subtypes using the complete Avian influenza HA

# mismatches	Length range	GC content range	# duplexes	Average difference	Standard deviation
1	$9-16$	$21\%-78\%$	110	0.56	2.06
2	$12-14$	$50\%-75\%$	28	-1.25	2.70

Table 2.3: Average and standard deviation for the difference (in degrees Celsius) between experimental melting temperature and predictions made by the SantaLucia model (2.1) on duplexes with one and two mismatches.

sequences from North America available in the NCBI flu database [8] as of March 2008 (a total of 574 HA sequences). Figure 2.2 shows the unrooted phylogenetic tree generated using the TREEVIEW program [69] from a multiple alignment of a subset of these sequences constructed using ClustalW [47].

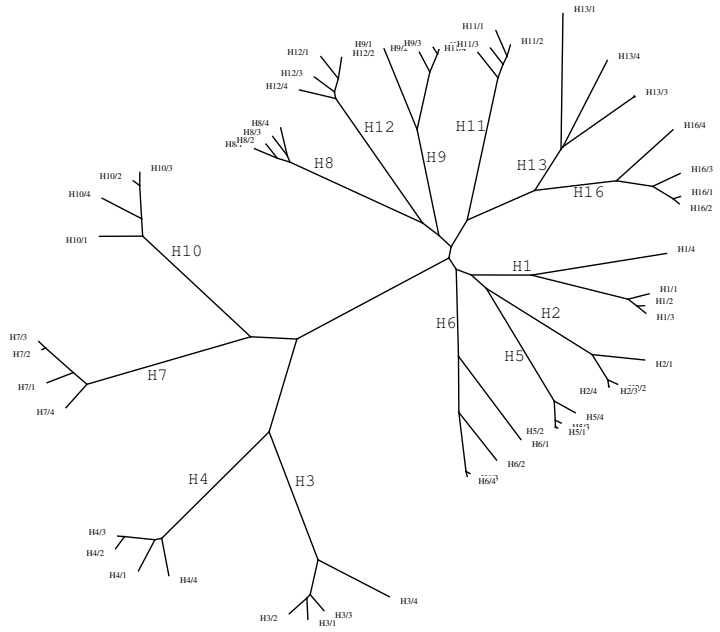


Figure 2.2: Phylogenetic tree of avian influenza HA sequences of North American origin from the NCBI flu database (5 complete sequences selected at random for each subtype).

When designing primers for each subtype H_i we used all available HA sequences classified as H_i as targets, and all NCBI HA sequences labeled with different subtypes as non-targets. Primer selection was performed using the following parameters:

1. Primer length between 20 and 25
2. Amplicon length between 75 and 200
3. GC content between 25% and 75%
4. Maximum mononucleotide repeat of 5
5. $3'$-end perfect match mask $M = 11$
6. No required $3'$ GC clamp
7. Primer concentration of $0.8\mu M$
8. Salt concentration of $50mM$
9. $T_{target}^{min} = T_{nontarget}^{max} = 40°C$

We also attempted to design primer pairs for the 9 known NA subtypes based on the 668 avian Influenza NA sequences available in [8], using the same set of parameters as for HA subtypes. An initial PrimerHunter run resulted in primer pairs selected for all subtypes except N4 and N1. Upon inspection of the phylogenetic tree (see Supplementary Figure 2) we detected an N1 sequence (GI:115278096) that was mis-labeled as N4. After correcting the label of the sequence, PrimerHunter was able to select discriminative primer pairs for all NA subtypes (see Supplementary Table 1).

The numbers of identified primer pairs using these parameters are summarized in Table 2.4. For comparison, we also include in Table 2.4 the number of probes reported by OligoSpawn [106] and SLICSel [96]. These were the only methods among those listed in Table 2.1 that were available and could run successfully run on the HA dataset. OligoSpawn and SLICSel were run using similar settings as PrimerHunter for the common parameters. Using these settings, all three methods were

able to identify discriminative primers/probes for each subtype represented in the NCBI flu database. The number of discriminative primers found by PrimerHunter is consistently larger than the number of probes found by OligoSpawn and SLIC-Sel. PrimerHunter identified at least a few tens of forward and reverse primers for each subtype. With an amplicon length constrained to be between 75 and 200 base pairs, PrimerHunter was able to always identify feasible primer pairs, i.e., pairs of primers predicted to amplify *all* target sequences and *none* of the non-target sequences when using an annealing temperature of $40°C$ in the PCR reaction. Identified primers typically have minimum primer-target melting temperature is significantly higher than $40°C$, and maximum primer-non-target melting temperature is significantly lower than $40°C$ (see supplementary material). The large number of feasible primers enables further optimizations such as selecting most discriminative primers (based on the difference between minimum primer-target T_M and maximum primer-non-target T_M) and T_M matching the primers within selected primer pairs.

Subtype	# Targets	# Non-Targets	Avg. % Diss.	# FP	# RP	# PP	# Probes SlicSel	# Probes OligoSpawn
H1	48	526	8.4	51	52	70	20	2
H2	41	533	9.1	42	43	187	14	2
H3	72	502	11.1	41	61	135	7	1
H4	67	507	7.4	265	225	3724	18	2
H5	69	505	9.1	68	66	160	17	1
H6	100	474	15.4	36	27	3	4	3
H7	55	519	8.9	77	81	260	2	1
H8	9	565	6.3	489	482	14415	100	1
H9	23	551	8.7	140	152	1222	58	1
H10	16	558	6.8	243	302	3712	35	1
H11	45	529	5.9	267	262	4117	32	1
H12	15	559	7.1	472	494	12895	52	1
H13	10	564	14.4	41	33	98	1	2
H16	4	570	9.5	367	352	7629	68	1

Table 2.4: Primers found for each subtype of Avian influenza HA and comparison with number of probes generated by related tools. The dissimilarity within a subtype is calculated as the average pairwise Hamming distance in the multiple sequence alignment expressed as percentage of the average sequence length. (FP: Forward Primers; RP: Reverse Primers; PP: Primer Pairs)

The number of discriminative primers and primer pairs found for a subtype is positively correlated with the amount of variability within the subtype and negatively correlated with the average similarity to closely related subtypes. Indeed, for pairs of subtypes such as (H3,H4), (H7,H10), (H8,H12), and (H13,H16) which are nearest neighbors in the NA phylogenetic tree in Figure 2.2, the subtype with lower within-subtype dissimilarity (included in Table 2.4) always yields a larger number of primer pairs. For our design parameters the number of suitable primer pairs varies from 3 for the highly variable H6 subtype, which has an average within-subtype dissimilarity of 15.4%, to 14,415 for the H8 subtype, which has an average within-subtype dissimilarity of 6.3%.

2.5.3 Primer Validation

A total of 9 randomly selected primer pairs specific to H3, H5 and H7 subtypes (3 pairs per subtype, see the supplementary material) were ordered from Integrated DNA Technologies (IDT). In a first experiment, triplicate Q-PCR reactions were performed for each primer pair with $1 : 10^3$ dilutions of each of the 3 plasmid types as template. Triplicate reactions with no template (*no template controls*, or NTC) were also performed. Figure 2.3 gives the amplification curves for a typical experiment where 3 on-target and 6 off-target Q-PCR reactions were performed with one of the H5-specific primer pairs. For each reaction, the *threshold cycle* Ct is defined as the PCR cycle in which the fluorescent signal intensity passes the self-calibrated detection threshold. When no detectable fluorescent signal is present (e.g., in a NTC reaction), Ct is set to 40.

For each reaction, ΔCt is computed as the difference between the respective threshold cycle and the average threshold cycle of the 3 NTC reactions. The minimum, maximum, and average ΔCt values for all 9 primer pairs and both on- and off-target templates are given in Figure 2.4. The results show a large difference

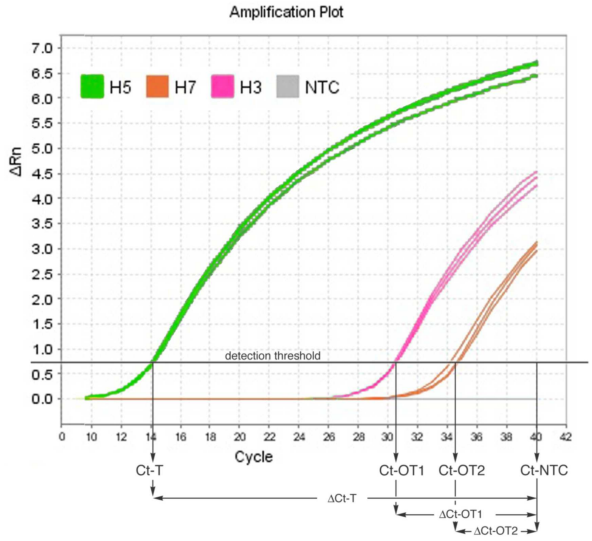


Figure 2.3: Amplification curves using an H5-specific primer pair and H3, H5, H7 plasmids or no template (3 replicates each).

(15 cycles or more) between the average on-target and off-target ΔCt values.

To assess the discriminative power over a range of template concentrations, 3 primer pairs (one specific to each of the 3 cloned subtypes) were used in triplicate Q-PCR reactions performed using each of the on- and off-target plasmids at 10 different dilutions. As can be seen from these graphs, PrimerHunter primer pairs showed template specific amplification over 5 to 7 orders of magnitude. Figure 2.5 shows ΔCt values of these reactions plotted against approximate plasmid copy numbers.

2.6 Discussion

We demonstrate the performance of PrimerHunter by designing thousands of primer pairs specific to fourteen HA and nine NA Avian Influenza subtypes. For the HA subtypes, the number of primers found by PrimerHunter is consistently larger than the number of probes found by two probe design tools with closely related func-

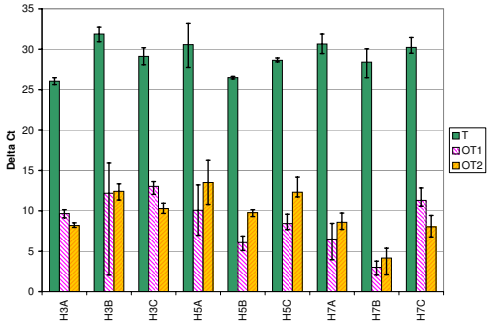


Figure 2.4: Average ΔCt for on-target (T), off-target (OT1 and OT2), and no template control (NTC) Q-PCR amplification with 9 primer pairs (3 subtype-specific pairs for each of H3, H5, and H7; error bars indicate minimum/maximum values).

tionality [106, 96]. The number of discriminative primers and primer pairs found for a subtype is positively correlated with the amount of variability within the subtype and negatively correlated with the average similarity to closely related subtypes. Indeed, for pairs of subtypes such as (H3,H4), (H7,H10), (H8,H12), and (H13,H16) which are nearest neighbors in the HA phylogenetic tree in Figure 2.2, the subtype with lower within-subtype dissimilarity (included in Table 2.4) always yields a larger number of primer pairs. For our design parameters the number of suitable primer pairs varies from 3 for the highly variable H6 subtype, which has an average within-subtype dissimilarity of 15.4%, to 14,415 for the H8 subtype, which has an average within-subtype dissimilarity of 6.3%. Degenerate primers were not needed by PrimerHunter when designing primer pairs based on Avian Influenza originating from North America. We expect that degenerate primers will become useful when designing discriminative primer pairs based on world-wide subtype isolates, and we plan to experiment with degenerate primers in the future.

In order to assess the specificity of these primers we tested 3 primer-pairs de-

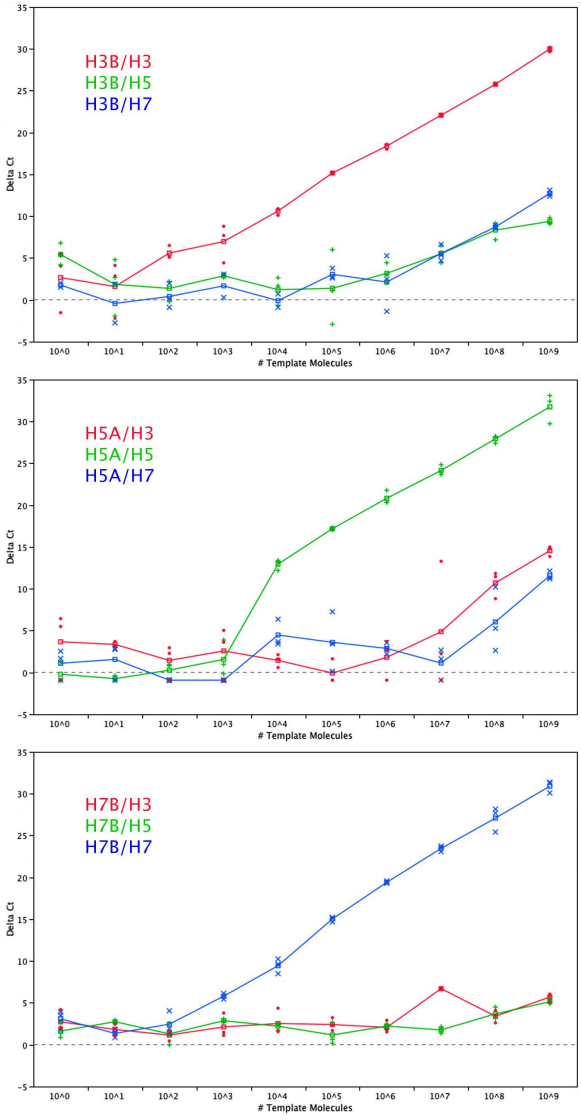


Figure 2.5: ΔCt for triplicate Q-PCR reactions performed with H3-, H5-, and H7-specific primer pairs at ten different dilutions of on- and off-target templates. Lines connect triplicate means at each dilution. The legend in each graph indicates the color for the primer (numerator) and target (denominator) combination.

signed to amplify HA fragments from H3, H5, and H7. To avoid the possibility of contaminated or non-clonal primary viral samples, fragments of the HA gene from one isolate of each subtype were cloned into a plasmid vector. This allowed us to test both the specificity of the PrimerHunter primers on defined on- and off-target sequences, and to assess the performance of the primers over a very large range of template concentrations. We found that in each of these experiments, PrimerHunter primers selectively amplified the targeted HA subtype over 5-7 orders of magnitude of target concentrations and that the target sequence was first detected at 10^4-10^6 fold lower concentrations than non-target templates. When template concentrations of both targets are raised to detectable levels, the target is typically amplified to concentrations $> 2^{15}$ fold greater than the off-target sequence.

In a typical field or clinical assay, target and off-target nucleic acid sequences are likely to be present at low concentrations. In the case of retroviruses such as influenza, the target nucleic acid will be viral RNA and any PCR assay will perforce be preceded by a reverse transcription (RT) step resulting in a linear DNA template. While the sensitivity of such an assay will be heavily dependent upon the efficiency of the RT step, we have shown that PrimerHunter primers are functional and specific under a wide range of template concentrations and thus are likely to be robust under a variety of experimental conditions including viral subtyping by RT-PCR in the clinic and in the field [14, 88, 101, 103].

By default, PrimerHunter seeks to select primer pairs predicted to amplify *all* target sequences and *none* of the non-target sequences under specified reaction conditions. When targets exhibit extremely large dissimilarity and such primer pairs cannot be found, PrimerHunter automatically seeks and reports a small set of primer pairs that collectively amplify all targets and none of the non-targets. If the number of primer pairs required to cover all targets is large, the pairs may need to be portioned into multiple multiplex PCR reactions due to limitations on

the number of primers that can be used in a single reaction.

Complete classification of unknown viral samples into subtypes can be achieved by using PrimerHunter to design a specific primer pair (or set of primers) for each subtype, then running n parallel PCR reactions where n is the number of subtypes. The number of PCR reactions can be further reduced by designing primer pairs specific to sets of subtypes (e.g., superclades in the phylogenetic tree). By employing such non-specific primer pairs and group testing methods similar to those in [24] the number of reactions can potentially be reduced to $\log n$, and we plan to explore such methods in future work.

Chapter 3

Towards accurate expressed variants detection and genotyping on whole transcriptome sequencing[1]

Second generation sequencing technologies have significantly increased our knowledge about the structure and amount of variation within and among human genomes. After publication of the first two human genome sequences [19, 51] and the establishment of the human reference genome [45], sequencing technologies based on production of massive amounts of short reads have enabled completion of a growing number of individual genomes within the last two years, including two cancer samples (See [86] for a review of current achievements and challenges). Drops in sequencing costs will allow research groups like the thousand genomes project (`http://www.1000genomes.org/`) to establish a database of thousands of individual genomes from different human populations in the near future.

[1]The results presented in this chapter are based on joint work with I. I. Mandoiu and P. Srivastava

As DNA sequencing has been used to get near to complete catalog of variants and individual genotypes, sequencing of mRNA transcripts (RNA-Seq) is becoming the method of choice to understand the functional implications of genetic variability [60, 91, 94] and to study highly variable genomes like cancer tumors [18, 63, 92, 95]. RNA sequencing promises to be an important source of information to understand the effect of genetic variation on regulation mechanisms and to establish causal relationships between mutations and diseases. For cancer research, comparisons between different RNA-Seq experiments on normal tissues and tumors can provide the information needed to discover driver mutations or to find new targets for therapies [58].

RNA-seq introduces novel data analysis problems [21]. First, the DNA to mRNA translation mechanism includes the junction of separate regions called exons delimited by splicing sites. This makes the default usage of tools for mapping of DNA reads to the reference genome like Maq [53] or Bowtie [46] not suitable for finding the right location of reads spanning splicing sites. Different methods based on spliced alignments have been proposed to identify splicing sites and try to assemble full transcripts, at the cost of computational resources [11, 61, 64, 94, 104]. But even if this problem is solved, differences in transcription levels introduce unequal coverage depths in the sequencing data making it difficult to identify variants in regions with low expression levels. One way to overcome this difficulty is to sequence both genomic DNA and mRNA and identify the variants from the genomic DNA reads using standard methods. However, if the interest is in expressed variants, it is more cost effective to identify them directly from the mRNA reads using bioinformatics approaches [17].

In this chapter we present a strategy for accurate and efficient mapping of mRNA reads, and a new method for single nucleotide variants (SNV) detection and genotyping. To improve the success rate and accuracy of read mapping, we map

mRNA reads against both the reference genome used by the UCSC browser [77] and the consensus coding sequences (CCDS) database [74] and introduce a rule set to combine mapping results. We also present a bayesian model for SNV detection and genotyping, that calculates for each locus conditional probabilities for each of the ten possible genotypes given the allele calls and their quality scores and then chooses the genotype with highest posterior probability based on the Bayes rule. Unlike other bayesian methods [53, 54], we keep calls for the four possible alleles and we do not apply a separate model to test heterozygosity. We instead assumme independence among the reads and try to fully exploit the information included in the base quality scores.

We performed validation of new and existing methods by reanalyzing publicly available Illumina mRNA reads taken from blood cell tissue of an individual in the CEU population of the International Hapmap project [20]. We used as gold standard more than three million validated SNP genotypes available in the Hapmap database. Our results indicate that the combined mapping strategy yields improved genotype calling accuracy compared to performing genome or CCDS mapping alone and that our SNV detection and genotyping method is more sensitive than existing methods for equal levels of specificity. We also show how sensitivity and specificity are affected by commonly used data curation strategies like reads trimming, filtering of copies to correct for variable transcription levels and PCR artifacts and allele coverage thresholds.

3.1 Materials and Methods

3.1.1 Mapping strategy for mRNA reads

Mapping mRNA reads against the reference genome using standard mapping programs such as Bowtie [46] or Maq [53] does not require gene annotations but leaves

reads spanning exon junctions unmapped. Spliced alignment methods such as [11] could theoretically overcome this difficulty but in practice they are computationally intensive and not well suited for very small read lengths. On the other hand, mapping against a reference transcript library like the Consensus Coding Sequences Database (CCDS) [74] allows to recover reads spanning known splicing junctions but fails to recover reads coming from not annotated genes.

We decided to map reads both against the reference genome and CCDS transcripts and to implement a custom rule set for merging the two resulting datasets. We implemented two approaches that we called hard merging and soft merging. For hard merging, we require unique alignments against both reference sequences and agreement between them while in soft merging we relaxed the uniqueness constraint by requiring a unique alignment to at least one reference and keeping that alignment. For both approaches we keep reads that map uniquely to one reference and do not map to the other one. Table 3.1 summarizes the decision rules applied to each read by each approach, depending on how the read mapped on each reference and on the concordance between the two alignments. For transcripts mapping, multiple alignments can be reported for some reads not because there exist different genomic locations where the read could come from but because the same genomic location is shared by many transcripts. Our merging approach checks for each read with multiple alignments if that is the case, and if so, we keep the best alignment as unique.

3.1.2 SNV detection and genotyping

To discover expressed SNVs in the sample we experimented with SOAPsnp [54] and Maq [53], which are two widely used bayesian methods implemented in the SAMtools package [52]. We also tried the SNV detection method for mRNA reads called PMA[15], which is based in careful filtering of alignmed reads and a binomial

Genome Mapping	CCDS Mapping	Agree?	Hard Merge	Soft Merge
Unique	Unique	Yes	Keep	Keep
Unique	Unique	No	Throw	Throw
Unique	Multiple	No	Throw	Keep
Unique	Not Mapped	No	Keep	Keep
Multiple	Unique	No	Throw	Keep
Multiple	Multiple	No	Throw	Throw
Multiple	Not Mapped	No	Throw	Throw
Not Mapped	Unique	No	Keep	Keep
Not Mapped	Multiple	No	Throw	Throw
Not Mapped	Not Mapped	No	Throw	Throw

Table 3.1: Decision rules for mapped reads merging. Unique, Multiple and Not Mapped mean that the read was respectively mapped uniquely, mapped in multiple places or not mapped at all to the corresponding reference.

test equivalent to set up a minimum coverage threshold to make a varian call relative to the total locus coverage. The trade between sensitivity and specificity of this method is controled by the maximum p-value required to discard the null hypothesis of absence of a variant allele. In terms of outcome, both SOAPsnp and Maq have the apriori advantage of not just point out the loci with alleles different than the reference but also infer which is the most likely genotype of each locus. The bayesian methods are also able to calculate for each locus both the probabilities of having an allele different than the reference and of the genotype itself.

We also implemented a custom bayesian method named *SNVQ* which calculates for each locus the posterior probability of each of the ten possible genotypes given the reads. Given a locus i, let R_i be the set of mapped reads spanning this locus. In all bayesian methods, the posterior probability of each genotype is calculated from its prior and conditional probabilities by using the bayes rule:

$$P(G_i|R_i) = \frac{P(R_i|G_i)P(G_i)}{P(R_i)}$$

The main difference between models lies in the way conditional probabilities are calculated [23]. Both Maq and SOAPsnp use a different model to calculate probabilities of homozygous and heterozygous genotypes. Maq uses a binomial distribution on the alleles having the two highest counts while SOAPsnp uses a rank test to determine heterozygosity. SOAPsnp also assumes as prior information that the homozygous reference genotype is the most likely one and calculates conditional probabilities based on Illumina specific knowledge about the reads [54]. We decided instead to use the same assumptions to calculate conditional probabilities for homozygous and heterozygous genotypes. Assuming independence between the reads the conditional probability of G_i can be expressed as a product of read contributions as follows:

$$P(R_i|G_i) = \prod_{r \in R_i} P(r|G_i)$$

For a mapped read $r \in R_i$ let $r(i)$ be the base spanning locus i and $\varepsilon_{r(i)}$ be the probability of error sequencing the base $r(i)$, which we estimated from the quality score $q(i)$ calculated during primary analysis using the Phred formula $\varepsilon_{r(i)} = 10^{-q(i)/10}$ [29]. We discarded allele calls with quality scores zero and one. Let H_i and H_i' be the two real alleles in the locus i, or in other words, let $G_i = H_i H_i'$. The observed base $r(i)$ could be read from either H_i or H_i'. If there is an error in this read, we assume that the error can produce any of the other three possible bases to be observed with the same probability, so the probability of observing

a base $r(i)$ given than the real base is different is $\varepsilon_{r(i)}/3$ while the probability of observing $r(i)$ without error is $1 - \varepsilon_{r(i)}$.

Assumming a heterozygous genotype $G_i = H_i H'_i$, $H_i \neq H'_i$ if the observed allele $r(i)$ is equal to H_i (H'_i) it could be due to two possible events. Either $r(i)$ was sampled without error from the haplotype H (H') or $r(i)$ was sampled from the haplotype H' (H) but an error turned it to be equal to H_i (H'_i). Assuming that both haplotypes are sampled with equal probability, the first event happens with probability $(1 - \varepsilon_{r(i)})/2$ while the second happens with probability $\varepsilon_{r(i)}/6$. Since for homozygous loci, the probability of observing each possible base does not depend on the haplotype from which the reads were sampled, the following equation summarizes how to calculate the probability of r for each possible genotype:

$$P(r|G_i = H_i H'_i) = \begin{cases} 1 - \varepsilon_{r(i)} & , \text{ if } H_i = H'_i = r(i) \\ \frac{\varepsilon_{r(i)}}{3} & , \text{ if } H_i \neq r(i) \wedge H'_i \neq r(i) \\ \frac{1}{2} - \frac{\varepsilon_{r(i)}}{3} & , \text{ otherwise} \end{cases}$$

We complete the model by calculating prior probabilities based on the expected heterozygosity rate h in the following way:

$$P(G_i = H_i H'_i) = \begin{cases} \frac{1-h}{4} & , \text{ if } H_i = H'_i \\ \frac{h}{6} & , \text{ otherwise} \end{cases}$$

In our experiments, we assumed a heterozygosity rate $h = 0.001$. Finally, a variant is called if the genotype with highest posterior probability is different than homozygous reference. In the next section we show a comparison of results among these methods by reanalyzing a publicly available dataset.

3.1.3 Software and performance issues

We implemented the merging methods and SNVQ in Java 1.6 and we packed both programs with a few additional utilities in a single jar file. In order to enable integration with other analysis tools we chose the SAM format [52] both as input and as output format for the code implementing the mapping strategies. We also sorted alignments by chromosome and absolute position to enable efficient processing in the subsequent modules and fast merging with results from different lanes if available. SAM files produced by the merging module can be used directly as input for the SAMtools package [52] to produce run statistics, pileup information, and for variants detection. We recommend to run the merging process lane by lane because it needs to load all unique alignments in memory in order to sort them at the end of the process. We used space efficient structures that allow us to process more than ten million reads in a few minutes, using up to 16Gb of memory. More reads can be processed at the expense of memory. The code implementing SNVQ is able to receive as input either aligments in SAM format or pileup information in the format described in the SAMtools package. The pileup format is recommended because it enables faster processing and reduces the memory requirements. Our experiments indicate that SNVQ is able to process a whole transcriptome pileup file in about 20 minutes using a single processor and up to 4Gb of memory. The open source code released under the GNU General Public License, is available at `http://dna.engr.uconn.edu/software/NGSTools/`.

3.2 Results

3.2.1 Methods validation with mRNA reads

We tested the performance of the mapping calling strategies and SNV detection methods on 33 bp long publicly available Illumina mRNA reads generated from blood cell tissue of the Hapmap individual NA12878 [20] (NCBI SRA database accession numbers SRX000565 and SRX000566). We used Bowtie [46] to map the reads against both the hg19 reference genome available at the UCSC genome database [77] and the CCDS database [74]. Table 3.2 shows results in terms of reads uniquely mapped using each considered method. We used as gold standard for comparison purposes $3,371,552$ Hapmap genotype calls for NA12878 on known SNPs of the CEU population. We classified them by their genotype as $2,008,415$ homozygous reference, $802,472$ heterozygous and $560,665$ homozygous non reference.

Sample Id	Raw Reads	Transcripts Mapping	Genome Mapping	Hard Merge	Soft Merge
SRR002052	12.6	2.9	4.3	4.5	4.7
SRR002054	12.9	3.9	5.7	5.9	6.2
SRR002060	25.7	4.4	6.7	7.0	7.3
SRR002055	11.4	3.7	5.5	5.6	5.9
SRR002063	23.0	3.5	5.6	5.8	6.0
SRR005091	13.9	3.3	4.9	5.0	5.2
SRR005096	14.4	0.6	1.0	1.1	1.1
Total	113.9	22.4	33.8	34.9	36.4

Table 3.2: Number of initial reads (Millions), and reads after mapping to the reference genome and to the CCDS transcripts and selecting unique alignments either separately or using the merging strategies presented in the methods section

We measured accuracy of genotype calling for each method in the following way: We defined as true positive a correctly called heterozygous or homozygous non reference SNP and as false positive an incorrectly called homozygous SNP. We did not consider as error a heterozygous SNP called homozygous or not called

because this can be due to lack of expression of one or the two alleles. We consider that one method is more accurate than other when it is able to detect more true positives for the same amount of false positives, or conversely if it is able to detect the same amount of true positives with less false positives.

To assess the accuracy of our mapping strategies, we ran SNVQ on the datasets of unique reads after transcripts mapping, after genome mapping and after the two merging strategies explained in the methods section. Since after transcripts mapping it is only possible to detect SNVs in transcripts included in the CCDS database, we filtered out for this comparison the Hapmap SNPs in regions different than the exons in CCDS. Figure 3.1 shows that our merging strategies produced more accurate results than just genome or transcripts mapping for this dataset. Although in this comparison it seems like genome mapping could be more sensitive for some specificity levels, we confirmed by performing the same comparison on the full dataset of Hapmap SNPs and also calling variants using SOAPsnp and Maq that merging methods dominate for all levels of specificity (See supplementary material). Since we could not find a clear sensitivity advantage for the soft merge method over hard merging, we decided to keep hard merge as our method of choice for further experiments.

In order to perform comparisons between bayesian methods and PMA, which just performs SNV detection, we redefined our accuracy measures as follows: We defined as true positive a detected heterozygous or homozygous non reference SNP, no matter which is the actual genotype call, and as false positive a homozygous reference SNP marked as having a variant. The main difference between the two measures is that calling heterozygous a homozygous not reference SNP is a true positive for SNV detection, because the variant was detected, but a false positive for genotype calling because an inexistent reference allele is called. Figure 3.2 shows that bayesian methods performed better for SNV detection purposes than

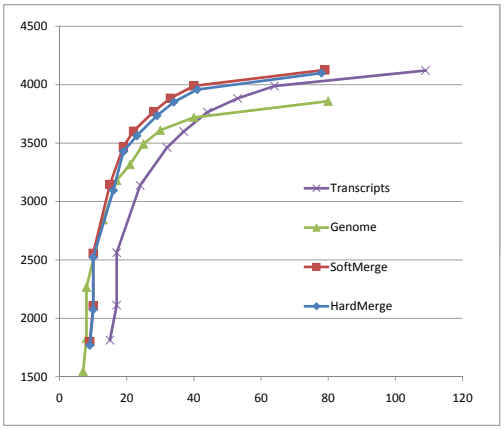


Figure 3.1: Accuracy comparison among the datasets of reads aligned uniquely to the reference genome, reads aligned uniquely to the CCDS transcripts, hard merged alignments, and soft merged alignments. This comparison was done over 41,961 Hapmap SNPs in CCDS exons using SNVQ for SNV detection and genotyping

PMA and that SNVQ was more accurate than SOAPsnp and Maq for different thresholds on the probability of having an allele different than the reference. We also compared the genotyping accuracy among the three bayesian methods. Figure 3.3 shows that SNVQ also achieved better accuracy than both SOAPsnp and Maq on this dataset for different specificity levels obtained by varying the threshold on the genotype probability reported by each method. We confirmed this behavior on the dataset of reads mapped uniquely to the genome reference and even after applying some of the data curation mechanisms presented below (See supplementary material).

3.2.2 Comparison of strategies for data curation

In practice SNV detection is the problem of separating allele calls that are different from the reference because of sequencing errors from calls that are different from the reference because they were sampled from a variant locus. With the current sequencing error rates, if sequencing errors were randomly distributed, it is not

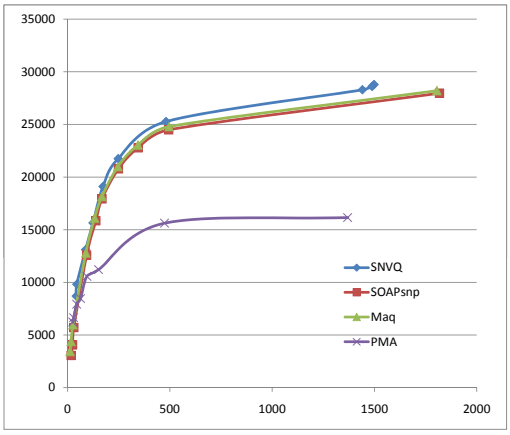


Figure 3.2: Accuracy comparison among four different SNV detection methods on the Hard Merged dataset. A total of 3,371,552 Hapmap SNPs with known genotypes for the individual NA12878 were used as gold standard for comparison. The tradeoff between sensitivity and specificity is controlled in the Bayesian Methods (SNVQ, SOAPsnp, and Maq) by varying the minimum probability of having a genotype different than the reference, while in PMA it is controlled by varying the maximum p-value required to discard the null hypothesis of absence of variants

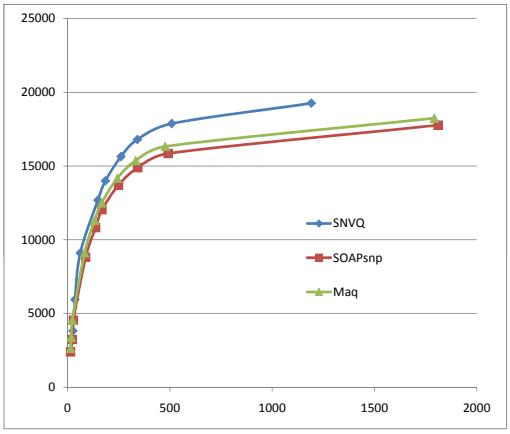


Figure 3.3: Accuracy comparison among three different bayesian methods for genotyping on the Hard Merged dataset. A total of 3,371,552 Hapmap SNPs with known genotypes for the individual NA12878 were used as gold standard for comparison

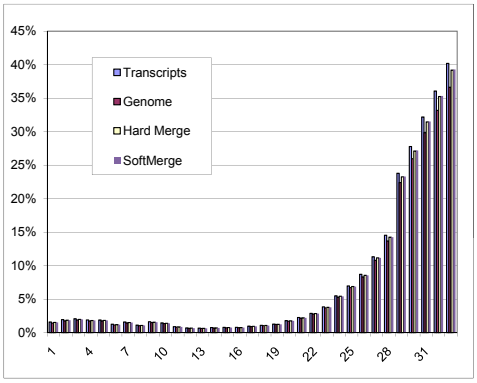


Figure 3.4: Percentage of aligned reads with a mismatch with the reference genome per read position from 5' end to 3' end

difficult to show that any of the discussed methods would have high accuracy. Unfortunately, each sequencing technology has different error patterns which can break the randomness assumption. In this section we describe three common issues related with Illumina sequencing and we show how common ways to solve these issues performed in our testing data.

One well known consequence of the way Illumina reads are produced is that base calling errors tend not to be totally random but they tend to accumulate toward the 3' end of the reads [10]. To test if this effect was happening in our dataset, we developed a module which calculates for a set of aligned reads the mismatches distribution per read position from the 5' to the 3' end. In absence of biases, this distribution should be close to uniform. However, as shown in Figure 3.4, the proportion of mismatches increases towards the 3' end of the reads.

After observing this pattern in the mismatches rate, we decided to apply a filter on the aligned reads by disregarding the first base and the last 10 bases of each aligned read for SNV detection. Although this trimming strategy is equivalent to throw away one third of the aligned bases for this dataset, figure 3.5 shows that this correction improves the specificity of the final calls without loosing sensitivity. Trimming aligned reads instead of raw reads is better because the bases sampled

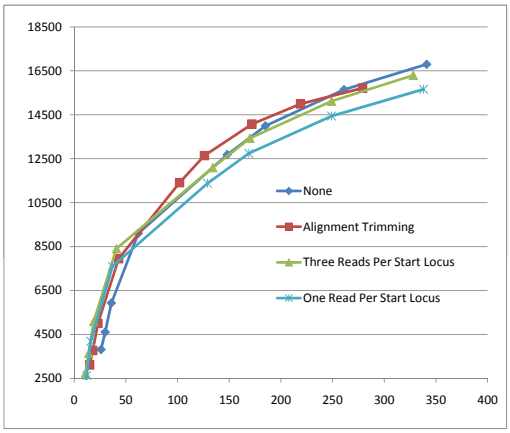


Figure 3.5: Accuracy comparison among three different strategies to filter information for SNV detection and genotyping on the Hard Merged dataset. Results for the dataset without filters are included as reference

correctly in the trimmed region are useful to locate uniquely the place where the read must be aligned.

Another common source of false positive results is PCR amplification artifacts which produce large copies of the same read introducing biases for variants detection [43]. One usual way to deal with this issue is to allow just one read to start at each possible locus. This filter eliminates artificial high coverages on every locus, which is apriori desirable not just for disregarding reads coming from PCR artifacts but also in mRNA sequencing to normalize for biases produced by variable transcription levels. The main drawback of this strategy is that it can throw away a lot of information affecting sensitivity. An intermediate approach consists on allowing a small number x of different reads per start locus as it is described in [15]. We implemented the two filtering strategies and we compared the results. Figure 3.5 shows that selecting just one read per start locus is indeed too restrictive for this dataset but the three reads filter of [15] did not affect sensitivity and even improved it for high specificity levels. Although the improvement is not as consistent as the one obtained by trimming aligned reads, we consider that this

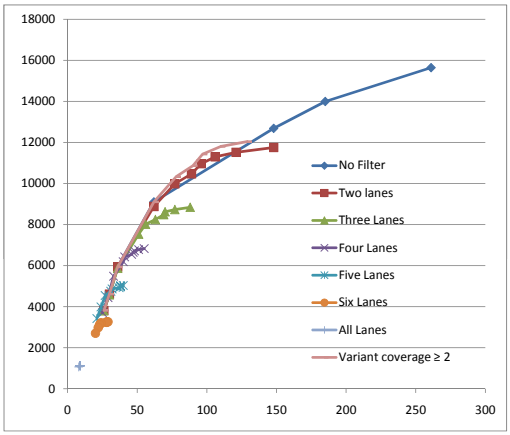


Figure 3.6: Accuracy comparison among different thresholds on the number of lanes where each alternative allele must be seen. Results for the SNPs without filtering and with a simple coverage minimum threshold of at least two reads per alternative allele are also included as reference

filter is worth to be used in general to disregard coverage biases without loosing sensitivity.

Finally, to control for potential lane specific biases, since the same sample is normally sequenced in many lanes to increase coverage, one potentially useful rule to increase specificity is to require each called variant allele to be seen in at least x different lanes. We analyzed reads coming from seven different lanes and hence we were able to observe how sensitivity and specificity are affected by this kind of filter. Figure 3.6 shows that after requiring a minimum of three lanes out of seven, the loss of sensitivity is larger than the improvement in specificity. We compared also the simple rule of keeping variants passing the threshold of observing at least two times the non reference allele with the more restringent rule of observing the non reference allele in at least two different lanes. We found that the first filter produced slightly better accuracy for this dataset.

3.2.3 Accuracy for different expression levels

One of the most important facts to take into account while sequencing messenger RNA is that transcripts are sampled according with their relative expression levels. Hence, it is expected to have better genotyping accuracy for variants in transcripts with higher expression levels. To check if that was the case for our dataset we calculated RPKM (Reads per Kilobase per Million Reads) values for every exon in the CCDS database according with the hard merged alignments. For each exon, we counted the number of reads that span it partially or totally and then we normalized the count, to account for different exon lengths, dividing it by the size of the exon in kilobases. We finally divided that number by the total number of reads in millions (34.9 for the hard merged dataset). We defined the RPKM of each heterozygous and homozygous non reference Hapmap SNP in a known exon as equal to the RPKM of the exon where it belongs. We finally took bins on the RPKM values and grouped variants according with their bin membership to check if there was any relation between SNV detection accuracy and the RPKM values. Figure 3.7 shows that, as expected, the sensitivity is very low for variants with low RPKMs. The average RPKMs for the second and the second to last bins are 2.6 and 70.23 respectively, so, a direct estimation based on these results suggests that the sample should be sequenced approximately 27 more times to be able to detect more than 80% of the variants for all exons with RPKMs greater than 1. The total number of mapped bases after hard merging is 1.15Gb, so this estimate corresponds to 31Gb in mapped bases, which is less than one third of the number of mapped bases needed to achieve similar percentages for genomic DNA [98]. Larger read lengths increase the percentage of mapped bases for the same initial coverage lowering down this estimate.

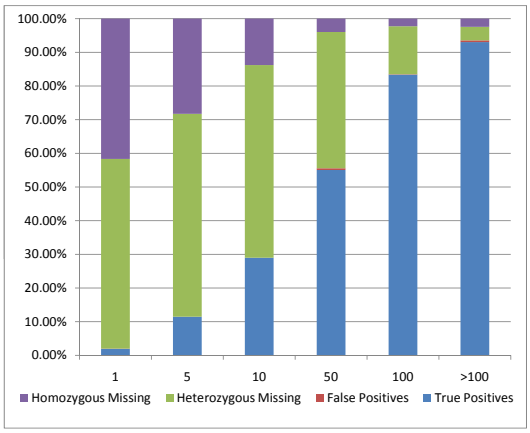


Figure 3.7: Percentage of true positives, false positives and not seen alleles for different bins on the RPKM values of the CCDS exons

3.3 Conclusion

Second generation sequencing of messenger RNA is becoming the method of choice to investigate the behavior of human cells and to reveal the functional effects of variation. In this paper we performed a comparison among different commonly used mapping, SNV detection and genotyping strategies with the aim to select the most effective methods to identify, as accurately as possible, the expressed single nucleotide variants in a sample from mRNA sequencing data. We also introduced a mapping strategy for mRNA reads which fully utilizes the information contained in the reference sequence and in a database of known transcripts like CCDS. Finally, we presented a bayesian model for SNV detection and gentyping called SNVQ that seeks to improve genotype calls by fully exploiting the information contained in base quality scores.

The availability of the Hapmap genotypes database and the short reads archive (SRA) allowed us to make direct accuracy assessments of different SNV detection methods and mapping and filtering strategies. Our experiments indicate that the double reference mapping and merging strategy yields to improved accuracy on

SNV calls compared with methods based on mapping to a single reference file. We also observed that SNVQ achieved improved accuracy over existing models. We tested different data curation and filtering strategies and we found that reads trimming after mapping improved the accuracy for our testing dataset and that controling for highly covered regions as described in [15] does not significantly affect sensitivity.

As future work, we plan to make further improvements in genotype calling by adapting our model to differential allelic expression events [95]. We also plan to use our methods on mRNA from cancer tumors to generate the variants information needed to look for new targets for different kinds of treatments like immunotherapy.

Chapter 4

ReFHap: A Reliable and Fast Algorithm for Single Individual Haplotyping[1]

DNA sequencing is at the cornerstone of current advances in genetics, enabling breakthroughs in medical and biological research [10]. The first complete human genome sequence was in fact a "representative" genome sequence based on the DNA of several individuals [45]. Advances in sequencing technology and computational methods are resulting in increasingly cost-effective, high-throughput sequencing, making the sequencing of genomes from individuals possible [10, 51, 62, 85, 98, 100]. This allows the identification of common patterns of human genetic variation between individual genomes that may affect health, disease, and individual responses to medications.

Human somatic cells are diploid, containing two sets of chromosomes, one set derived from each parent. Differences between the two copies of each chromosome

[1]The results presented in this chapter are based on joint work with T. Huebsch, G. McEwen, E. Suk and M.R. Hoehe [27]. See the original version at `http://doi.acm.org/10.1145/1854776.1854802`

are called heterozygous variants and for each variant the sequences that differ are called alleles. Most of the variation comes in form of single nucleotide variants (SNV) where the alleles are base pairs that differ between the two copies. However, alleles can also be identified in other types of variations, for example, structural variations and insertion and deletion events (indels).

The process of grouping alleles that are present together on the same chromosome copy of an individual is called haplotyping [35, 78]; reconstruction the two sequences of each chromosome is the most advanced type of haplotyping. Besides getting the full structure of the genome, complete haplotype sequences enable improved predictions of changes in protein structure produced by mutations in coding regions and increase power for genome-wide association studies [83]. It will also allow insights into the complex interplay of alleles of genes and their regulatory elements.

At present, published sequences may be considered "mixed diploid" [36]; because they actually represent a composite of the two underlying haploid sequences. Although each study presents preliminary haplotyping results, complete construction of the "true" molecular sequences for each of the chromosome pairs remains a challenge towards full genome completion [51, 62]. This is mainly due to the fact that current sequencing technologies do not provide enough information to reliably separate alleles originating from each of the two copies of a chromosome unless parental or population information is available [6, 97, 102]. However, this situation is likely to change in the near future with improvements in second-generation sequencing methods such as longer read lengths, mate-pairs and increased throughput, and also the development of new experimental approaches. This work is part of a currently productive approach towards molecular haplotype determination which relies upon fosmid-based sequencing [13, 40]; Details about resources and approaches we have developed towards this aim are available at

http://www.molgen.mpg.de/~genetic-variation/Projects.html.

Algorithms for haplotyping can be grouped in three categories depending on the type of source information: (i) population information, (ii) parental and individual genotype information and (iii) evidence of co-occurrence of alleles. Population information takes genotype information for a group of individuals known to come from the same population and uses an evolutionary model to phase all genotypes at the same time [34, 84, 89] whereas parental and individual genotype information enables alleles to be grouped into loci where either parent is homozygous and hence there is no doubt on deciding which allele comes from which parent [12, 59]. A combination of these approaches is used by the International HapMap Project [20] to generate haplotypes for each population. These two categories have the advantage that they require genotype information which is easy to acquire for certain loci. However, acquiring parental or population information may not feasible for all heterozygous variants of the individual of interest. In algorithms based on evidence of the allele co-occurrence, this information has so far come from reads or mate-pairs spanning at least two heterozygous variant loci but in general the evidence can come from any source. DNA sequences showing co-occurrence of two or more alleles are usually called fragments. The computational problem of haplotyping based on this kind of information is called single individual haplotyping and many formulations and algorithms have been proposed to solve it [78, 70]. Although absence of real data has been always an issue, the problem has been carefully studied from both theoretical and practical points of view and simulated data has been used to make comparisons between different approaches [6, 97, 102]. The algorithm presented here is a contribution to this approach.

Previous studies on single individual haplotyping have established different problem formulations seeking for different optimization objectives. Computational properties of these formulations have been studied and it has been shown that most

of them are NP-hard [78, 70]. Proposed algorithms can be divided into exact, genetic, probabilistic and heuristic. Due to the NP-hardness of the formulations, exact algorithms require an exponential dependency on at least one input parameter so they do not scale well as the size of the input gets large [97]. Genetic and probabilistic algorithms have the advantage of searching within a large set of possible haplotypes at the expense of time [7, 97]. Heuristic algorithms try to find efficiently a haplotype as close to the optimum as possible according to a specific optimization criterion. One of the most accurate of these algorithms is HapCUT [6], which starting from a random solution, builds a graph and uses max-cut to find loci that should be flipped to improve the current haplotype based on the input data. Our experiments indicate that although the algorithm is reliable, its running time is too large for whole genome haplotyping.

We present a novel problem formulation for single individual haplotyping and a heuristic algorithm called ReFHap. Like in [6] our formulation allows us to reduce the problem to max-cut but here we design a graph that enforces separation of fragments rather than variant loci. Our approach initially attempts to find the cut that groups together fragments coming from the same copy of each chromosome to subsequently build haplotypes consistent with the best cut found by our heuristic algorithm. We show through extensive simulation experiments that ReFHap represents an improvement in running time compared to previous algorithms without loss of accuracy. Simulations indicate that ReFHap is more accurate and scalable than the model of [102] and that it has comparable accuracy and higher efficiency than HapCUT [6]. Moreover, we tested both ReFHap and HapCUT with preliminary real data from fosmid-based sequencing. Results indicate that ReFHap is able to efficiently perform whole chromosome haplotyping with good accuracy.

4.1 Methods

4.1.1 Problem Formulation

Informally, the objective of single individual haplotyping is to reconstruct the two haplotypes of an individual from a set of partial readings called fragments. Each fragment provides evidence of coocurrance of two or more alleles of different SNPs in the same haplotype. The usual strategy to predict the true haplotypes is to define a real function on the input data and an arbitrary pair of haplotypes, hoping that the real haplotypes will correspond with the result of an optimization objective on this function.

As in previous works [6, 70], we represent the input of the problem as a matrix M of size mxn where m is the number of fragments and n is the number of variant loci. Each row of M encodes the information for one fragment as a string on the alphabet $\{0, 1, -\}$. Here, $M[i, j] \neq -$ means that fragment i calls allele $M[i, j]$ in locus j while $M[i, j] = -$ means that fragment i does not cover locus j. This problem definition inmposes a restriction of at most two alleles per locus, which is sufficient for most of the variation in many diploid organisms. There is no restriction on the type of variations considered as long as they can be mapped to a specific locus and the two alleles can be identified. Usually, the reference allele is encoded as a zero and the alternative allele as a one but this is not required by ReFHap. As in [6] ReFHap assumes that all input loci are heterozygous. If this is not the case, a preprocessing step like the one in [70] can be implemented to call genotypes and remove homozygous loci and fragments covering at most one heterozygous locus.

Given two strings f_1, f_2 of length n, we say that $f_1 = f_2$ if and only if for every $1 \leq j \leq n$, $f_1[j] = -$, $f_2[j] = -$ or $f_1[j] = f_2[j]$. In absence of errors, the problem reduces to find two haplotypes $h, \bar{h}$ such that every fragment f_i is equal to either

h or $\bar{h}$ according with the definition of equality given above. This problem can be solved just by separating the fragments in two groups such that for any pair of fragments f_1, f_2 inside a group $f_1 = f_2$ and then building the haplotypes by taking the consensus allele for each locus inside each group. This is equivalent to solve max-cut on a graph $G = (V, E)$ where V is the set of fragments and $e = \langle f_1, f_2 \rangle$ if and only if $f_1 \neq f_2$. In absence of errors, G is bipartite and hence max-cut can be solved in polynomial time for G. However, the solution is not unique if the graph is not connected. The connected components of G are called in this setting haplotype blocks. The number of these blocks affects the quality of the output haplotypes because in absence of additional inputs, there is no information to decide how to connect two consecutive blocks and hence the probability of joining them consistently is $0,5$.

If fragments contain errors, G may not be bipartite anymore and conflics are created between fragments. A simple example of a conflict happens when there are two loci j_1, j_2 covered by two fragments f_1, f_2 for which $f_1[j_1] = f_2[j_1]$ and $f_1[j_2] \neq f_2[j_2]$. Clearly one of the entries must be wrong if both loci are heterozygous. Different strategies to remove conflicts lead to optimization objectives studied in previous works like finding the minimum number of fragments to remove (MFR), the minimum number of loci to remove (MSR) or the minimum number of allele calls to correct (MEC). Computational properties of these problems have been analyzed by [78, 70] and several algorithms have been proposed for MEC [6, 33, 97, 101]. If weights are available for each allele call on each freagment, the model called (WMLF) described by [102] tries to minimize the sum of weights of corrected alleles.

Our approach is to reduce the problem to max-cut as in the case without errors but adding weights to the edges of G in such a way that the cut that maximizes the sum of the weights of crossing edges resembles as accurate as possible the actual

origin of each fragment. Weights are calculated based on the following scoring scheme. Given two allele calls a_1, a_2, the score $s(a_1, a_2)$ is given by:

$$s(a_1, a_2) = \begin{cases} -1 & \text{if } a_1 \neq - \wedge a_2 \neq - \wedge a_1 = a_2, \\ 1 & \text{if } a_1 \neq - \wedge a_2 \neq - \wedge a_1 \neq a_2, \\ 0 & \text{otherwise.} \end{cases}$$

Now, given two rows i_1, i_2 of M, the score $s(M, i_1, i_2)$ is just the sum of the contributions for each pair of alleles at each locus:

$$s(M, i_1, i_2) = \sum_{j=1}^{n} s(M[i_1, j], M[i_2, j])$$

Note that if two fragments do not cover any common locus then their score is zero but the opposite is not necessarily true. Given a fragments matrix M we can define a cut of fragments as a set of rows $I \subseteq \{1, \ldots m\}$. Given a matrix M, the score of the cut I is given by:

$$s(M, I) = \sum_{i \in I} \sum_{k \notin I} s(M, i, k)$$

We use this scoring function to state the following problem definition:

Maximum Fragments Cut (MFC): Given a mxn matrix M of m fragments covering n loci, find a cut I such that $s(M, I)$ is maximized.

Theorem 1 *Maximum Fragments Cut is NP-Hard.*

Proof. Max-Cut can be reduced to MFC in the following way. Given an instance $G = (V, E)$ for max-cut, build M by creating a row for each element of V and for each edge $\langle i_1, i_2 \rangle \in E$ make a column j and set $M[i_1, j] = 0$, $M[i_2, j] = 1$ and $M[k, j] = -$ for every row l different than i_1 or i_2. The score $s(M, i_1, i_2)$ for any

pair of rows is equal to 1 if and only if $\langle i_1, i_2 \rangle \in E$ otherwise, it will be zero because the two rows cover at most one common locus. Now, given a cut on G represented by a subset $V' \subseteq V$, the weight of this cut will be equal to the score $s(M, I)$ of the cut I made by selecting the rows corresponding with the vertices in V'. Hence, any algorithm that can calculate the maximum of the function $s(M, I)$ will calculate also the max-cut value for G and conversely, any algorithm that can calculate the max-cut value for G will also calculate the maximum of the function $s(M, I)$. □

4.1.2 Algorithm

To solve MFC we build a graph $G = (V, E, w)$, where $V = \{1, \ldots, m\}$, $\langle i_1, i_2 \rangle \in E$ if and only if $s(M, i_1, i_2) \neq 0$ and for all $e = \langle i_1, i_2 \rangle \in E$, $w(e) = s(M, i_1, i_2)$. Then, we solve the weighted version of Max-Cut on G. We implemented a heuristic algorithm similar to the one used by [6] that iterates over the edges and for each one builds an initial greedy cut and then tries to improve it through local optimization. The main steps are the following:

ReFHap(M, k)

1. Build $G = (V, E, w)$ from M
2. Sort E from largest to smallest weight
3. Init I with a random subset of V
4. For each e in the first k edges of E
   (a) $I' \leftarrow$ GreedyInit(G, e)
   (b) $I' \leftarrow$ GreedyImprovement(G, I')
   (c) If $s(M, I) < s(M, I')$ then $I \leftarrow I'$

The procedure GreedyInit finds a cut I' in which e crosses from I' to $V \setminus I'$ and the procedure GreedyImprovement tries to improve I' by local optimization. The parameter k controls how many edges are considered for the initialization step and hence allows to make a compromise between accuracy and speed. Unlike the algorithm in [6] in which initial edges are chosen at random, we decided to sort the edges because edges with large weight are more likely to be part of the best cut. The maximum value that k can take is $|E|$ but we can achieve good accuracy in many cases with a small value of k. In our current implementation $k = \sqrt{|E|}$.

To show how we implemented the greedy procedures we need to expand the score function to subgraphs. Given a graph $G = (V, E, w)$ and two disjoint subsets I_1, I_2 of V we define:

$$s(G, I_1, I_2) = \sum_{i \in I_1} \sum_{k \in I_2} w(\langle i, k \rangle)$$

and for each vertex $v \in V \setminus (I_1 \cup I_2)$ we can define:

$$s(G, I_1, I_2, v) = \max(s(G, I_1 \cup \{v\}, I_2), s(G, I_1, I_2 \cup \{v\}))$$

As in [6], to avoid high negative edges crossing the cut, we build a cut from an input edge $\langle i_1, i_2 \rangle$ by initializing I_1 with i_1 and I_2 with i_2 and then adding either to I_1 or I_2 the edge that locally maximizes $s(G, I_1, I_2, v)$:

GreedyInit$(G, \langle i_1, i_2 \rangle)$

1. Init $I_1 \leftarrow \{i_1\}$ and $I_2 \leftarrow \{i_2\}$
2. While $I_1 \cup I_2 \neq V$
   (a) Find $v \in V \setminus (I_1 \cup I_2)$ maximizing $s(G, I_1, I_2, v)$
   (b) If $s(G, I_1 \cup \{v\}, I_2) > s(G, I_1, I_2 \cup \{v\})$ add v to I_1 else add v to I_2

3. return I_1

For local optimization we implemented the classical greedy algorithm of [81], which calculates for each vertex $v \in I'$ the score $s(G, I' \setminus \{v\}, V \setminus (I' \setminus \{v\}))$ and for each vertex $w \notin I'$ the score $s(G, I' \cup \{w\}, V \setminus (I' \cup \{w\}))$ and flips the vertex with maximum score if it is larger than the current score $s(G, I', V \setminus I')$. We also implemented a local optimization step flipping edges rather than vertices, which is equivalent to check the improvement after flipping every possible pair of vertices at the same time. We iterate these two optimizations until neither of them can achieve any improvement.

After finding the cut I, the algorithm uses the input matrix to find the haplotype h that minimizes the number of entries to be corrected assuming that rows in I belong to h and rows in $V \setminus I$ belong to $\bar{h}$. Since we assume that all loci are heterozygous, the output of ReFHap is just one haplotype h and $\bar{h}$ is just the haplotype obtained by flipping every allele call in h. The haplotype h can be inferred by making a single traversal of M as follows:

1. For each column j

   (a) $I_{j,0} \leftarrow \{i : (i \in I \wedge M[i,j] = 0) \vee (i \notin I \wedge M[i,j] = 1)\}$

   (b) $I_{j,1} \leftarrow \{i : (i \in I \wedge M[i,j] = 1) \vee (i \notin I \wedge M[i,j] = 0)\}$

   (c) If $|I_{j,0}| \geq |I_{j,1}|$ then $h_j \leftarrow 0$ otherwise $h_j \leftarrow 1$

2. output h

The complexity of this algorithm depends on the number of fragments m, the number of different starting edges k, the maximum number of loci covered by a single fragment and the maximum number of fragments covering a single locus, which as in [102] we call respectively k_1 and k_2. First note that the maximum degree of a vertex in G is bounded by $k_1(k_2 - 1)$. To prove this note that one

fragment f calls alleles for at most k_1 loci. Each of these loci is covered by at most $k_2 - 1$ fragments different than f. Therefore, each locus contributes with at most $k_2 - 1$ edges to the vertex associated with f. In the worst case, there are no shared edges between loci and then the total number of edges is $k_1(k_2 - 1)$.

Since $k_1 k_2$ is typically much smaller than m, the total number of edges of G is $O(mk_1k_2)$. The sorting step takes then $O(mk_1k_2 \log(mk_1k_2))$ and, as shown below, it is dominated by the iterations step. For fixed G, I_1, I_2 and v, $s(G, I_1, I_2, v)$ can be calculated just by inspecting the edges of v, so, for each of the k edges considered, GreedyInit takes $O(m^2k_1k_2)$. Although the local optimization in theory could take a time equivalent to the sum of the positive edge weights which would be exponential on the input size, in practice the number of iterations is small enough to be considered a constant, so the time needed for each local optimization is $O(mk_1k_2 + mk_1^2k_2^2) = O(mk_1^2k_2^2)$. The total time is $O(k(m^2k_1k_2 + mk_1^2k_2^2))$. Comparing this result with the complexity of the algorithm designed by [102], which includes an exponential dependency on k_2, and the algorithms designed by [6] and [33], which need at least $O(mn^2)$ time, we can predict that ReFHap will perform faster especially as the number of loci covered by each fragment increases and hence less fragments are needed to achieve the same coverage. In the next section we will show simulation experiments confirming this hypothesis.

4.2 Results

4.2.1 Experimental Setup

We performed several simulation experiments to test the behavior of ReFHap under a wide range of circumstances. We generated instances varying over five different criteria: number of loci n, number of fragments f, mean fragment length l, error rate e and gap rate g. For each instance, we created a random haplotype h of

size n and then we created f fragments. For each fragment we selected its length l_i drawing from a normal distribution centered at l and with standard deviation equal to 1. We then selected its starting position j as a random integer in the range from 1 to $n - l_i + 1$. The fragment f_i is then the substring of h starting at position j with length l_i. We flipped the whole fragment with probability 0.5, assuming that real fragments are equally likely to come from either of the two haplotypes. Finally, we introduce errors by flipping each allele call of f_i with probability e and we also introduced gaps by deleting each allele call of f_i, except for the first and the last position, with probability g.

We performed one experiment for each combination of selected values of the simulation parameters. For each experiment, we generated 100 random instances following the procedure described above. We implemented ReFHap in Java 1.6 and we compared it with the public available implementation of HapCUT [6] and with the implementation of the WMLF model kindly provided by the authors of [102]. Both HapCUT and WMLF were implemented in C. We ran all experiments on a RedHat Linux 64 bit server.

Before checking the performance of ReFHap we investigated how the number of haplotype blocks changes for different number of fragments and fragment lengths. As shown in section 4.1, the number of blocks heavily influences the quality of the haplotype no matter which model is used to solve the problem. Figure 4.1 shows how as the fragment length increases, less coverage is needed to connect every locus with each other. Although the simulation assumes that variants are evenly distributed in the genome (in real organisms that is not always the case) this result means that for the same coverage, few large fragments produce fewer blocks than many short fragments.

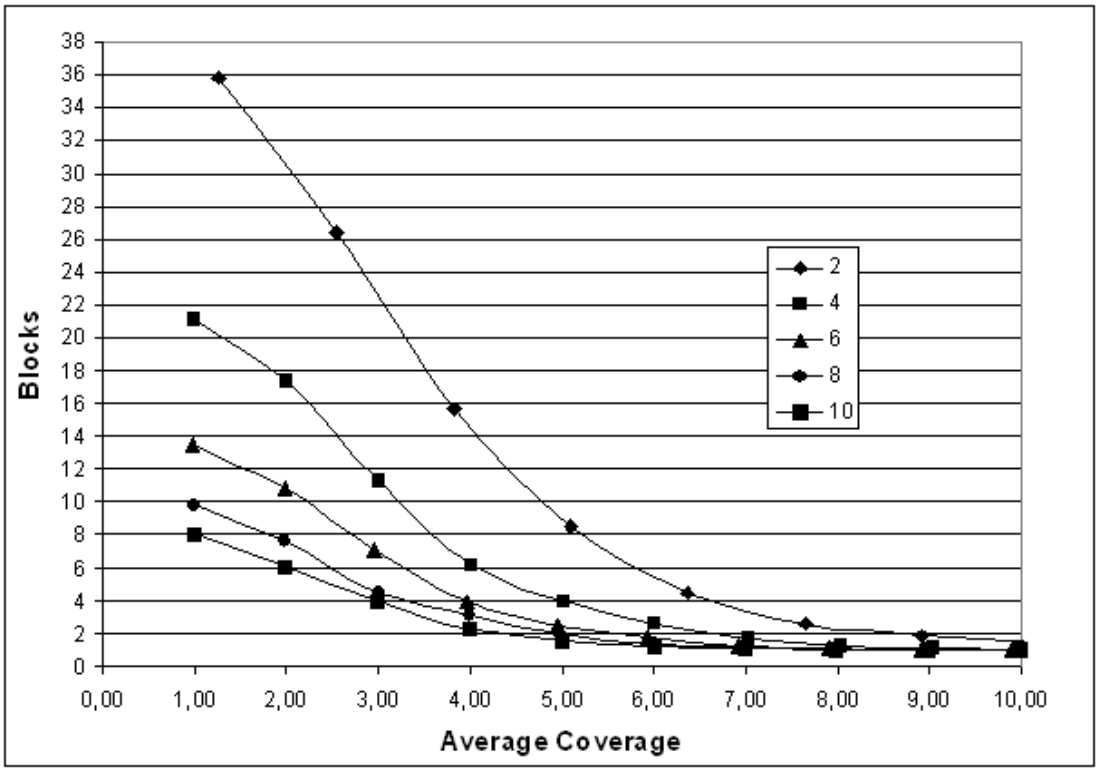


Figure 4.1: Number of blocks as a function of coverage for different fragment lengths.

4.2.2 Simulation Results

For each experiment we calculated means for three measures. The first one is the Minimum Error Correction (MEC), which is the minimum number of changes within the matrix to make it consistent with the answer haplotypes. This measure divided by the total number of allele calls in the input matrix is a good estimator of the allele calling error rate. The algorithm implemented in HapCUT assumes that the true haplotype is the one that minimizes this measure. The second measure is the switch error (SE) which is calculated by traversing the resulting haplotypes from left to right and computing the number of times needed to jump from one haplotype to its complement to reconstruct the real haplotype. Assuming absence of genotyping errors, this is the true measure of quality for any solution. However, this measure can not be calculated for real instances unless a gold standard haplotype is known. For our simulations we can calculate the number of switch errors because we know the true haplotype for each instance. The third measure is the running time of the algorithm measured by running it on a single processor.

The upper panel of figure 4.2 shows the distribution of differences between the WMLF model and ReFHap in MEC, switch errors and running time for experiments varying coverage by increasing the number of reads and fixing the number of loci to 200, the fragment length to 6 and with no errors or gaps. ReFHap consistently produces lower MEC and switch errors. WMLF has a better runtime for small instances but that changes when coverage increases and even after $10x$, the limit on the maximum coverage for one locus of 23 is often achieved. This limit is required by WMLF because the algorithm has an exponential dependency on this parameter.

The lower panel of figure 4.2 shows the distribution of differences between HapCUT and ReFHap for the same criteria as above for experiments with the

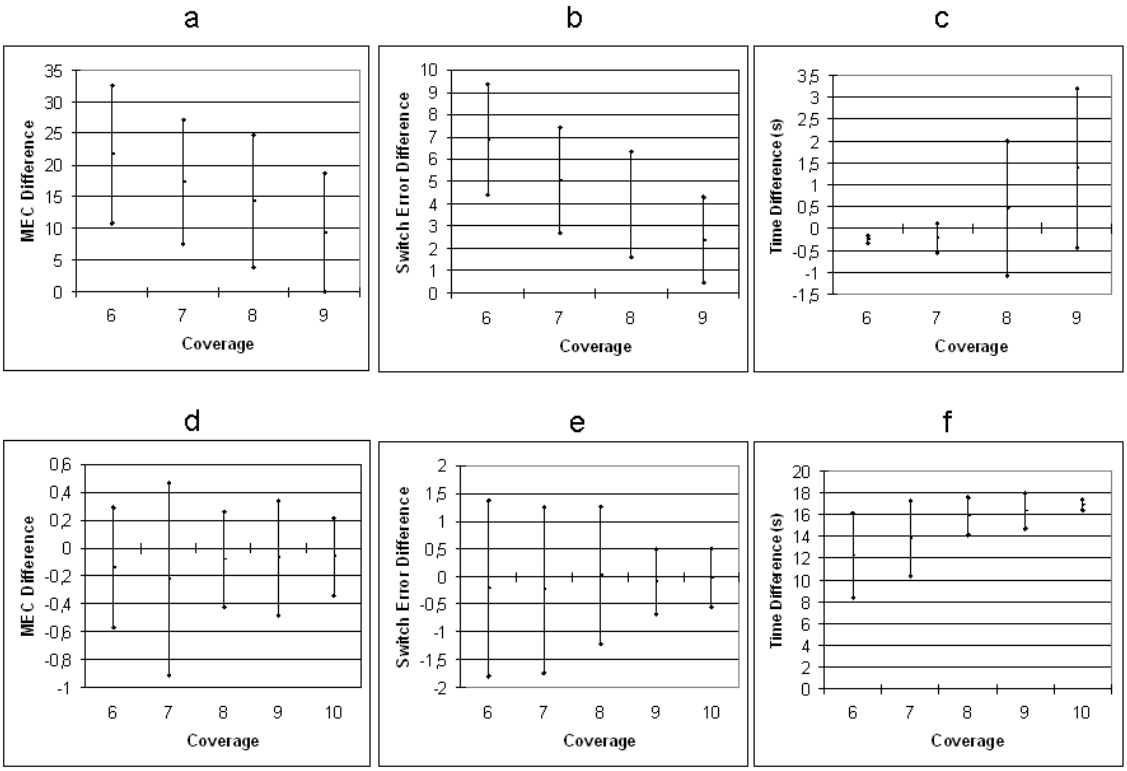


Figure 4.2: Distribution of differences between values reported by WMLF and HapCUT and values reported by ReFHap for experiments varying coverage by increasing the number of fragments. Markers above and below the mean correspond to the mean plus and minus one standard deviaton respectively. The upper panel shows the differences between WMLF and ReFHap in (a) MEC , (b) switch errors and (c) running time. The lower panel shows the differences between HapCUT and ReFHap in (d) MEC , (e) switch errors, and (f) running time

same number of loci and fragment length but adding an error rate of 5% and a gap rate of 10%. While the difference in switch errors between HapCUT and ReFHap is almost zero on average, ReFHap performs consistently faster than HapCUT. We performed a statistical test for each experiment to see if the differences are on average significanly different from zero and we found that this is the case in general for the time differences, in favor of ReFHap and for the MEC differences in favor of HapCUT. HapCUT provides lower MEC values because that is its optimization objective. However, switch errors are the true measure of quality, not MEC. Table 4.1 shows that we could not find evidence of a significant difference between HapCUT and ReFHap in switch errors for most of the experiments carried on, which means that the reliability of the two methods is similar. This table also shows that results in Figure 4.2 are replicated consistently by experiments increasing the mean fragments length up to 12 and the number of loci up to 1000.

We finally examined how the haplotype quality decreases as the error rate increases. Figure 4.3 shows that the number of switch errors increases at the same pace for both HapCUT and ReFHap as the error rate increases to an extreme value of 50%. These experiments were performed on 200 loci with 296 fragments of length 6 and a gap rate of 0.1, achieving a mean coverage of about $8x$ and a mean number of blocks of 1.06. These parameters were set up to ensure that most of the switch errors are produced by the error rate and by the behavior of the algorithms and not by the number of blocks.

4.2.3 Results with Real Data

We tested ReFHap and HapCUT on data resulting from the experimental fosmid based sequencing approach introduced by [13] and that we are currently developing (See `http://www.molgen.mpg.de/~genetic-variation/Projects.html`). We built a test case by sequencing and aligning fosmids generated from chromo-

Input			HapCUT			ReFHap			p-values	
l	n	f	%MEC	%SE	Time	%MEC	%SE	Time	p-value MEC	p-value SE
200	200	2	3.57	12.35	0.28	3.62	12.45	0.22	$6 * 10^{-6}$	0.32
200	200	4	4.62	5.29	1.81	4.69	5.95	0.26	$6 * 10^{-11}$	0.0001
200	200	6	4.89	1.46	10.54	4.9	1.55	0.39	$9 * 10^{-4}$	0.16
200	200	8	4.94	0.75	19.06	4.94	0.74	0.58	0.02	0.4
200	200	10	4.91	0.34	24.55	4.91	0.29	0.68	0.16	0.12
200	200	12	5.0	0.21	29.27	5.0	0.2	0.73	0.5	0.29
200	222	6	4.94	1.17	12.72	4.95	1.27	0.51	$5 * 10^{-4}$	0.09
200	259	6	5.08	0.7	14.49	5.1	0.82	0.67	$7 * 10^{-4}$	0.05
200	296	6	4.97	0.43	16.74	4.97	0.42	0.91	$9 * 10^{-3}$	0.44
200	333	6	4.93	0.22	17.58	4.93	0.27	1.29	0.04	0.06
200	370	6	4.91	0.14	18.65	4.91	0.15	1.79	0.02	0.35
200	700	2	4.75	1.75	10.72	4.8	1.87	3.58	$2 * 10^{-13}$	0.09
400	700	5	4.89	0.46	79.12	4.91	0.58	4.97	$2 * 10^{-6}$	0.002
600	700	7	4.98	0.48	300.63	4.99	0.51	4.70	$4 * 10^{-4}$	0.15
800	700	10	5.01	0.35	1064.62	5.01	0.35	5.47	0.05	0.48
1000	700	12	4.98	0.38	2279.25	4.98	0.45	5.89	0.16	0.002
200	296	6	4.95	0.4	16.72	4.95	0.48	0.88	0.01	0.05
400	592	6	4.98	0.38	94.95	4.98	0.395	4.34	$3 * 10^{-3}$	0.33
600	888	6	5.02	0.4	273.87	5.02	0.46	8.05	$5 * 10^{-5}$	0.03
800	1185	6	4.97	0.4	595.47	4.98	0.43	13.4	$6 * 10^{-4}$	0.10
1000	1481	6	4.98	0.35	1019.78	4.98	0.36	20.8	$3 * 10^{-3}$	0.30

Table 4.1: Minimum Error Correction (MEC), Switch errors (SE) and running time for HapCUT and ReFHap for simulation experiments varying haplotype length (l), number of fragments (n) and mean fragment length (f). Each reported p-value is the probability that HapCUT and ReFHap report on average the same value for MEC and SE on each set of input conditions. Time p-values were also calculated but, except for the first row, they are always less than 10^{-32}

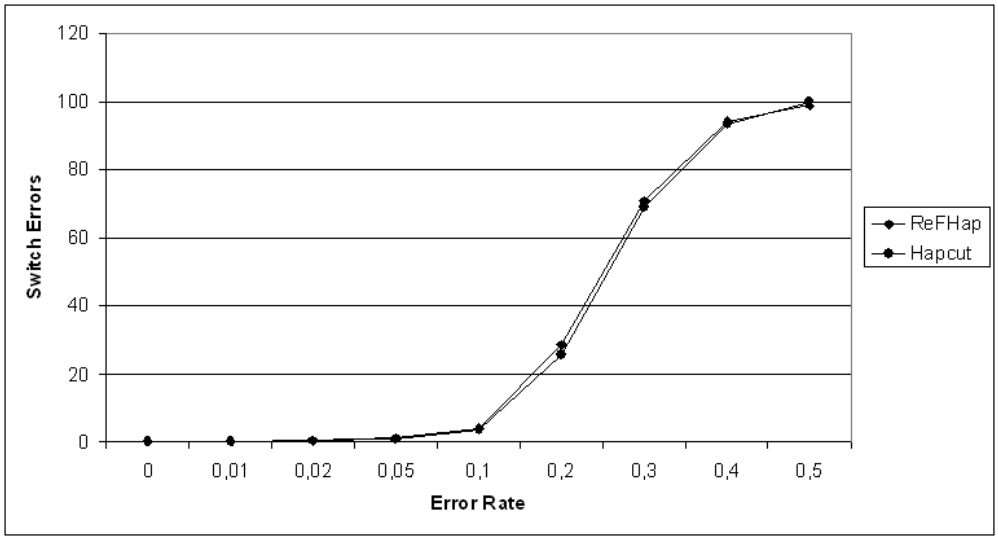


Figure 4.3: Switch error rate for HapCUT and ReFHap for experiments varying error rate

	ReFHap	HapCUT (1 It)	HapCUT (50 It)
%MEC	6.32%	6.26%	6.24%
Time	73.04 Sec	0.99 Hours	50.4 Hours

Table 4.2: MEC percentage and running time of ReFHap and HapCUT for a real instance with 32347 SNPs and 13905 fragments in chromosome 22

some 22 of a caucasian individual. The input for this test case is a matrix of 32347 SNPs covered by 13905 fragments. The total number of allele calls is 178191. Hence, each SNP is covered on average 5.51 times and each fragment covers on average 12.81 SNPs. The total number of haplotype blocks is 102. Table 4.2 shows MEC percentage and running time values for both ReFHap and HapCUT. We included HapCUT results for its first iteration and after 50 iterations. ReFHap clearly performed faster than HapCUT by solving this test case in about one minute while even one single iteration of the heuristic implemented in HapCUT takes about one hour.

Unfortunately for this data we do not have the true haplotype so we can not calculate the exact switch error rate. However, we did several quality control steps to verify the accuracy of the assembled haplotypes. First, we estimated the switch

error rate by running a simulation experiment with the same number of variants and fragments as in the real instance ($l = 32347$ and $n = 13905$), mean fragment length $f = 13$, gap probability $g = 0.1$ and error rate $e = 0.063$. Even though the total number of allele calls is on average 178004, which is less than the total for the real data, the mean switch error rate is 1.86%.

We also had access to Affymetrix 1000k chip genotypes for one hundred individuals coming from the same population as our indvidual. We ran fastPHASE [84] on these genotypes to obtain a phasing of 3158 SNPs on chromosome 22 for the same individual. This haplotype can not be considered a gold standard but we can compare it with other haplotypes by defining a measure of concordance. Given two haplotypes, we calculate the switch error rate of the first as if the second were the gold standard and then we call percentage of concordance the result of one minus this rate. We achieved a 92.89% concordance between ReFHap and fastPHASE on 2941 SNPs shared between the two haplotypes. The percentage of concordance between HapCUT and fastPHASE was 93.30%.

Finally, we used the same measure of concordance to compare ReFHap, HapCUT and fastPHASE haplotypes with 89 haplotypes from individuals in the the CEU population of the HapMap project [20] assembled by trio phasing. We selected these haplotypes for comparison because the information provided by trios makes them more reliable and because the true haplotype of our individual should be similar to CEU haplotypes. Figure 4.4 shows the proportion of CEU haplotypes for different percentages of concordance, or in different words, the distribution of percentages of concordance for each assembled haplotype with the CEU haplotypes. The average percentages of concordance for HapCUT and ReFHap are 87.55% and 87.21% respectively, while the average percentage of concordance for fastPHASE is 84.67%. This comparison was done over an average of 2065 common SNPs for ReFHap and HapCUT and an average of 989 common SNPs for

fastPHASE.

4.3 Discussion

Current advances in sequencing technologies will increase the amount and types of variation discovered for individual genomes and will provide the information needed to assemble the true pair of haplotype sequences underlying each human chromosome through single individual haplotyping [40]. However, the increase in throughput, accuracy and completeness of sequencing technologies will not be reflected in improved haplotype construction if algorithms are not suitable to handle efficiently genome-wide scale data. Full haploid sequences are the ultimate goal to achieve a complete understanding of the structure of the human genome.

We have contributed to this field by introducing a novel problem formulation for single individual haplotyping and a heuristic algorithm to solve it. Our approach tries first to predict the actual separation of the input fragments into two groups, one for each chromosome copy. To this aim, we introduced a scoring scheme that allows us to build a graph of fragments and assign a weight to the relation between each pair of overlapping fragments based on their calls for common loci. After solving max-cut on this graph, we build the consensus haplotypes based on the best cut found. Since this approach resembles the well known max-cut problem, a heuristic algorithm for this problem is used as the base for our algorithm.

We compared our algorithm with HapCUT[6], which is the most accurate heuristic algorithm that we found available and with the WMLF model [102] which is the only solution including error probabilities for the input alleles. We have shown through extensive simulations that ReFHap computes haplotypes faster than these solutions without loosing accuracy. We also used experimental data to show that ReFHap scales better than other solutions for chromosome wide input,

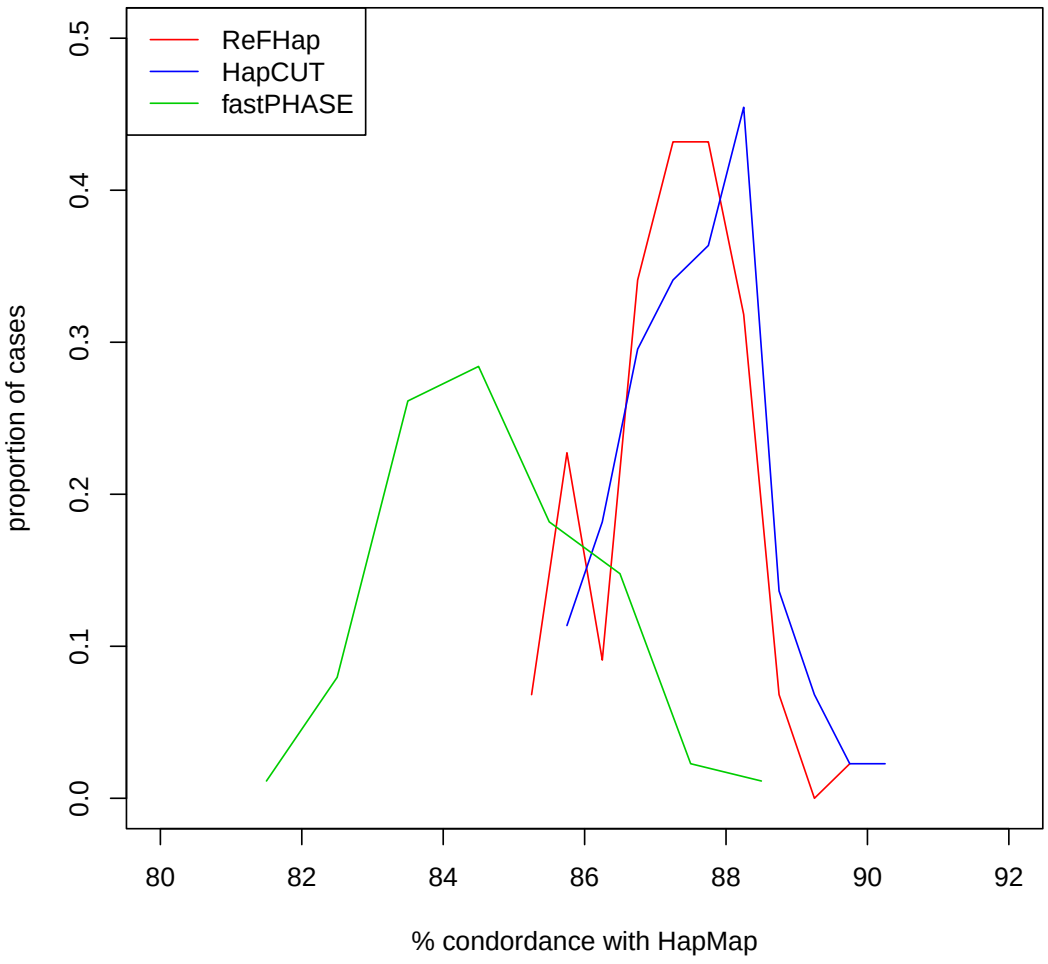


Figure 4.4: Distribution of percentages of concordance between assembled haplotypes for a caucasian individual and CEU HapMap haplotypes assembled from trio phasing

for which ReFHap finds reliable haplotypes within seconds while HapCUT takes one hour to make one iteration. We also performed comparisons with a statistical phasing approach and with high quality haplotypes from the CEU HapMap population [20].

It is difficult to establish a fair comparison between statistical phasing and phasing based on evidence of coocurance of alleles because the input information for both methods is too different to be comparable. However, we have shown that haplotypes assembled with single individual haplotyping can be more accurate than haplotypes inferred by statistical phasing if the region to assemble has enough coverage. Our simulations also help to get an idea of the coverage needed to achieve different levels of confidence.

In general, the biggest disadvantage of heuristic algorithms is that unlike exact algorithms, they do not provide the best solution for every instance. However, for this particular problem, formulations seek to optimize objective functions that are not fully correlated with the switch error rate. In that sense, even an exact algorithm cannot claim to provide the true haplotype sequences in every instance. In this scenario, an efficient heuristic algorithm with low error rates will be a better option from a practical point of view than an exact algorithm that can not ensure to have zero switch error rate.

In the near future, we intend to make further accuracy improvements by taking into account quality scores of fragment allele calls and by including other types of information like parental or population information within a single framework.

Chapter 5

Bioinformatics pipeline for detection of immunogenic cancer mutations by high throughput mRNA sequencing[1]

Immunotherapy is a promising cancer treatment approach that relies on awakening the immune system to the presence of antigens associated with tumor cells. In most of the cells, the proteasome breaks complex proteins into small peptides and some of these peptides are able to bind to MHC molecules. The main function of MHC molecules is to present attached peptides on the surface of the cell. Killer T-cells generated in the bone marrow are trained by negative selection in the thymus to ignore self peptides and attach just to MHC molecules presenting foreing peptides, usually called epitopes. When a T-cell binds to a foreign peptide presented by a host cell, it releases cytotoxins that induce apoptosis in the host cell. Figure 5.1 shows an schematic of this process [105].

[1]The results presented in this chapter are based on joint work with P. Srivastava and I. I.

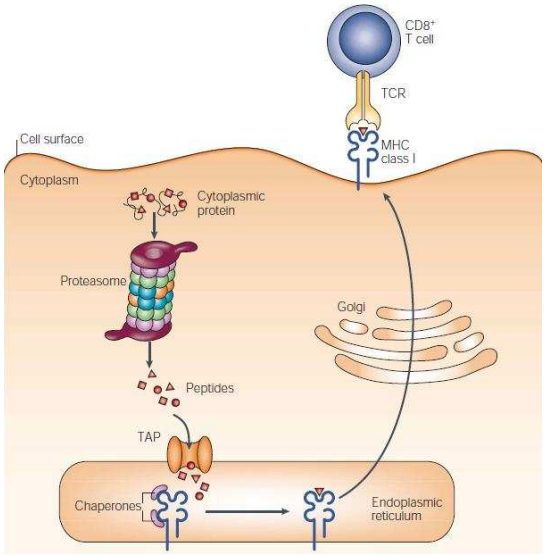


Figure 5.1: **MHC class I antigen presentation: the basics.** Cytosolic and nuclear proteins are degraded by the proteasome into peptides. The transporter for antigen processing (TAP) then translocates peptides into the lumen of the endoplasmic reticulum (ER) while consuming ATP. MHC class I heterodimers wait in the ER for the third subunit, a peptide. Peptide binding is required for correct folding of MHC class I molecules and release from the ER and transport to the plasma membrane, where the peptide is presented to the immune system. TCR, T-cell receptor. [105]

Provided that cancer cells accumulate mutations and some of these mutations can be translated to foreign peptides, killer T-Cells should be able to recognize and kill cancer cells. Although there is evidence that this actually happens for some tumors[65], sometimes the tumor grows fast enough to create barriers that block the immune system. The main assumption for an immunotherapy solution involving killer T-Cells to deal with these cases is that killer T-Cells can be "trained" to go over the tumor barriers and look for specific epitopes. Several therapy approaches have been proposed in previous works, but most of them look for self peptides overexpressed in cancer cells, which are easier to find than foreign peptides. The main risk of immunizing with self peptides is that they can produce autoimmunity disease on the individual.

Our approach is to combine next generation sequencing technologies with epitope prediction tools to reconstruct the tumor transcripts from mRNA reads and find a large amount of tumor specific epitopes. Then, we immunize the individual with a combination of such epitopes to induce clonal amplification on specific killer T-Cells that hopefully will find and attach to cancer cells presenting those epitopes, achieving tumor remission. Figure 5.2 shows an schematic of this cancer immunotherapy that we are currently evaluating on a mouse model before applying it to human tumors.

The success of this approach depends on the ability to reliably detect immunogenic cancer mutations, the vast majority of which are expected to be tumor-specific [75].

Mandoiu

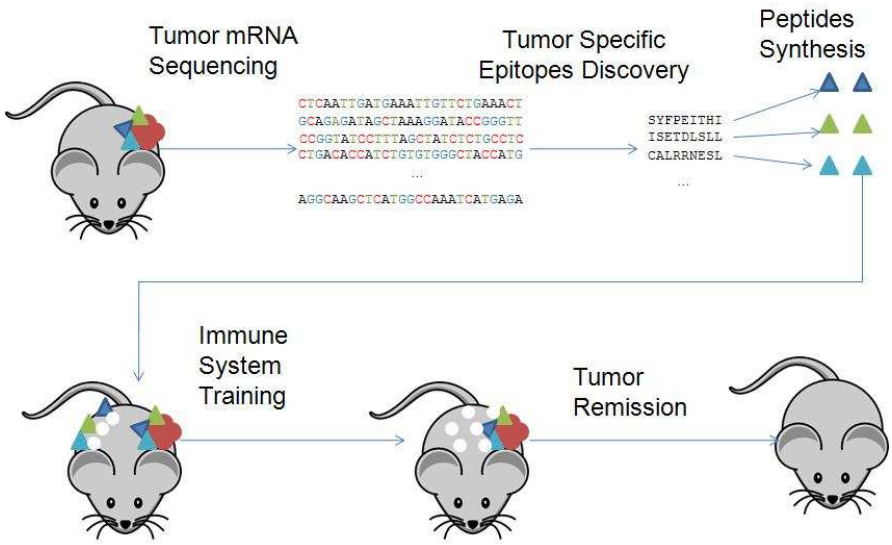


Figure 5.2: Cancer immunotherapy applied to a mouse model. MRNA reads are taken from tumor cells and are sequenced to find epitopes presented by the tumor cells. This epitopes are synthesized and injected in a normal tissue to awake the immune system. Killer T-cells clonally amplify and look for the same epitopes in other tissues and induce tumor remission

5.1 Analysis pipeline

We present a bioinformatics pipeline for detection of tumor specific epitopes from high throughput mRNA sequencing data. A schematic representation of our analysis pipeline is given in Figure 5.3. The pipeline consists of four main stages. First, sequencing reads are mapped separately against a reference transcript library (CCDS) and the reference genome using Bowtie [54]. Second, reads mapped by the two methods are merged as explained in chapter 3. Third, merged reads are used to call single nucleotide variants (SNV) and to predict haplotype configurations for detected close SNVs. In the last stage, discovered mutations are translated into mutated peptides, which are tested for immunogenic response.

We integrated the reads mapping and merging strategy presented in chapter 3 to increase the number of correctly and uniquely mapped reads. Although aligned reads are generated in SAM format and then any SNV detection and genotyping

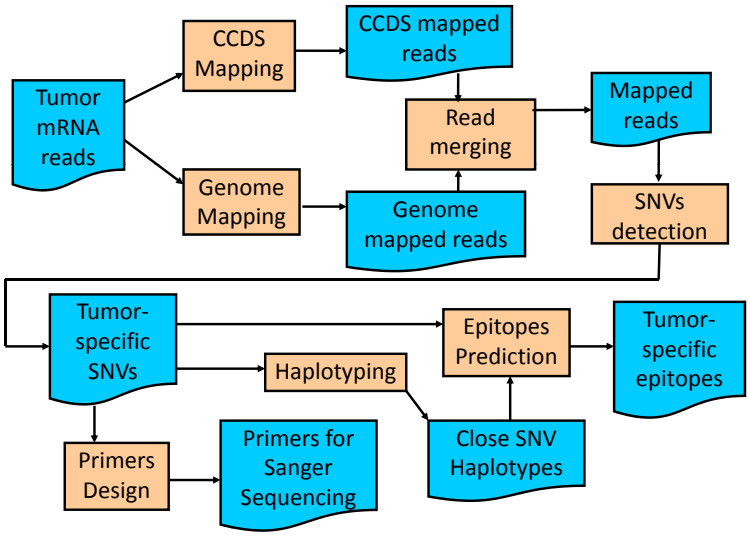


Figure 5.3: Analysis pipeline to identify antigenic mutations from mRNA sequencing reads

algorithm can be applied after this step, we chose to use SNVQ as our method to identify expressed SNVs taking into account the results comparison presented in chapter 3. Since peptides binding to killer T-Cells are usually up to 15 aminoacids long, two non synomymous SNVs must be at most 45 bp apart in a transcript and their alternative alleles must be in phase to be able to produce a double mutated epitope. This is a rare event for most of the genes in normal tissue but in cancer tissues good epitope candidates can result out of tumor specific close variants. The proximity requirement allows to use mapped reads as source information to predict the right phasing in most of these events. We integrated ReFHap (See Chapter 4) by building an input matrix for each chromosome with as many columns as heterozygous SNVs and as many rows as reads spanning at least two heterozygous SNVs. We sort the SNVs with the same order criteria used to sort the reads to build the matrix in a single parallel traversal linear to both the number of alignments and the number of heterozygous SNVs. From this input matrix, ReFHap finds blocks of close SNVs and provides an accurate phasing prediction for each one based on

the reads information. The full list of SNVs along with this phasing information is used as input for epitope predictions.

For each identified non-synonymous SNV or block of phased non synonymous SNVs, reference and alternative aminoacid sequences are generated using CCDS transcript annotations. Both peptides are then tested by either querying the NetMHC-3.0 program [57, 67] or SYFPEITHI database [75]. For each MHC allele, both approaches base their predictions on a Profile Weight Matrix (PWM), calculated from peptides that are known to bind to the allele. This matrix can be applied as a function that for each possible peptide produces a score that is supposed to be correlated with the strength of the binding between the MHC molecule and the peptide. The main assumption is that binding is mainly determined by a few highly conserved anchor aminoacids normally located toward the ends of the peptide. The final result of the analysis is the list of epitopes for which either the reference or the mutated peptide exceeds a user defined threshold.

Finally, we implemented a module to design primers for SNVs validation from genomic DNA using Primer3 [42]. This module locates each variant in the reference genome and produces the input needed to design primers for two PCR experiments. The objective of the first experiment is to amplify 500 bp around the SNV locus, so Primer3 is asked to find primers in the 150 bp before and after the selected region with a desired product length between 500 and 800 bp. The objective of the second experiment is to perform Sanger sequencing on the amplified region, so we instruct Primer3 to design primers in the first and the last 150 bp of the amplified region with a desired product length between 200 and 500 bp. This design enables to sequence at least the 200 bp surrounding the target SNV and hence improving the chances of running a succesful PCR validation. We use the lowercase masked versions of the reference genomes provided by UCSC [77] to know in advance which experiments are likely to fail due to repeated regions and also to filter out SNVs

in regions marked to be repetitive.

5.2 Results

We have run our pipeline with RNA Sequencing data from six different mouse cancer samples: a MethA cell line, a CMS5 cell line and four spontaneous prostate tumors. For each sample we built a modified reference genome and transcriptome depending on the original strain from where the cancer evolved. While MethA and CMS5 tumors evolved from a BALBC strain, spontaneous prostate tumors evolved from a C57BL strain. To build each reference we took the UCSC assembly [77] and the CCDS CDNA sequences [74] and we modified them according with the homozygous non reference SNPs released by the Sanger Mouse Genomes Project (`http://www.sanger.ac.uk/resources/mouse/genomes/`). We built a custom module that takes a reference and the set of homozygous non reference SNPs, for each SNP it goes to the reference and sets the alternative base of the SNP in the corresponding locus, and finally prints the modified reference. This module is available as part of the NGSTools package. The objective of modifying the references from the beginning is to make them as close as possible to the corresponding normal tissue for each sample to find the mutations aquired during the cancer development.

Table 5.1 shows statistics on reads analyzed for each dataset and number of variants found in each sample. To find these SNVs we ran Bowtie both against the assembly and the transcriptome, we performed hard merging, we filtered out two bases from the 5' end and ten bases from the 3' end, we ran SNVQ and we finally filtered out SNVs with genotype quality score less than 20 and less than 3 reads supporting the alternative allele.

We designed primers to perform PCR validation on each discovered SNV as

Sample	Initial Reads (M)	Aligned and Hard Merged Reads (M)	Total Aligned Bases (Gb)	Total SNVs	Het SNVs	Total Epitope Hits
MethA	105.79	67.15	4.03	29945	18063	32656
CMS5	42.31	19.52	0.94	8122	3789	9830
Prostate1	78.93	30.30	1.45	15992	6406	9756
Prostate2	38.36	10.60	0.51	3147	769	707
Prostate3	37.44	17.50	0.84	11281	3800	7980
Prostate4	21.49	12.11	0.58	9256	3549	6704

Table 5.1: Mapping and merging statistics, number of total and heterozygous SNVs discovered and total number of epitope hits for six different mouse cancer samples

described in the previous section. We validated 20 variants identified in the MethA dataset and we were able to confirm 18 of these 20 mutations. Moreover, we were able to confirm that some of these mutations were absent from normal tissue and hence were product of the cancer development. Figure 5.4 shows an example of a mutation shown to be absent in liver tissue and present in the MethA tissue.

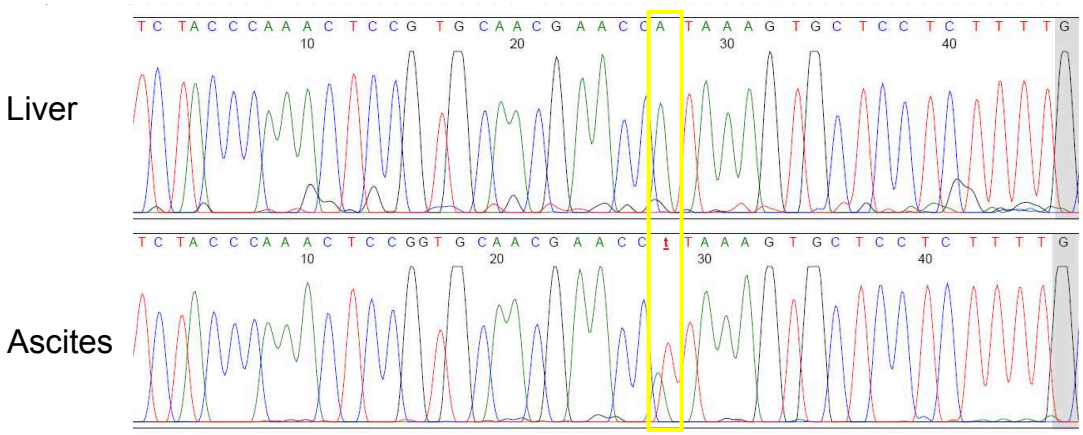
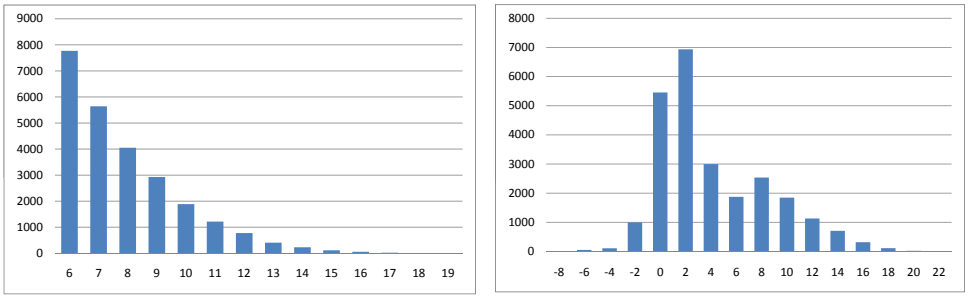


Figure 5.4: Validation using DNA Sanger sequencing shows no mutation on normal (liver) tissue and a predicted heterozygous mutation on the MethA cancer cell line

For each heterozygous variant in each dataset we predicted the protein translation of both the haplotype with the reference allele and the one with the alternative allele by modifying the reference according with the phasing information if there are SNVs close enough to the SNV under consideration and then performing cDNA to

protein translation according with the frameshift annotated in the CCDS database. We then performed class I epitope predictions on the two peptides associated with each SNV and we reported hits with a NetMHC score higher or equal than 5 for the alleles H2-Dd, H2-Kd and H2-Ld. Table 5.1 shows the total number of hits for each dataset. We also calculated for the MethA sample the distribution of scores and the distribution of differences between the peptide with the reference allele and the peptide with the alternative allele. Figure 5.5(a) shows that, as expected, the number of cases decreases as the score threshold increases but that there are still more than a hundred cases with scores higher than 13, which according with the NetMHC predictions corresponds to a high binding probability. Figure 5.5(b) shows that there are more than a thousand cases where the peptide corresponding with the alternative allele scored 10 or more points higher than the peptide with the reference allele. We found also that phased mutations help to find peptides with larger positive differences. While the average difference for cases with one aminoacid change was 2.65, the same average for cases with multiple aminoacid changes was 5.22.



(a) Scores

(b) Score Differences

Figure 5.5: Distribution of number of predicted epitopes per NetMHC score (a) and NetMHC score difference bins (b) for predicted epitopes from the MethA mouse cancer tumor cell line.

Chapter 6

Conclusion

Continuous development and improvement of methods to analyze genetic samples has boosted a dramatic increase in the understanding of the mechanisms governing the behavior of living organisms within just a few years. This detailed knowledge is fundamental for medical research to create new vaccines, diagnoses and treatments for different kinds of diseases such as viral or bacterial infections, brain disorders, and even cancer tumors. Bioinformatics methods for analysis of genomic data have become a cornerstone within this research due to the large amount of new data produced everyday and to the complexity of the questions to be answered.

We have contributed to this effort by developing PrimerHunter, a primers design tool for identification of virus subtypes from PCR experiments. Primer Hunter has been designed to deal with the complexities related with the high rates of variability within and among subtypes produced by high mutation rates in the reproductive cycle of a virus strain. Compared to existing tools based on exact matches or multiple sequence alignment, PrimerHunter achieves a higher design success rate by relying on accurate melting temperature computations allowing for mismatches based on the nearest-neighbor model of [82] and the fractional programming approach of [49]. Using this approach, PrimerHunter can

design primers that will selectively amplify target sequences from a complex background of related targets. The PrimerHunter web server, as well as the open source code released under the GNU General Public License, are available at `http://dna.engr.uconn.edu/software/PrimerHunter/`.

We plan to explore the potential application of PrimerHunter to design PCR assays for identification and subtyping of pathogens other than influenza, including bacteria, parasites and fungi. Another potential application for PrimerHunter is designing specific probes for gene expression and genome enrichment microarrays. For large eukaryotic genomes these applications would require very large numbers of melting temperature computations which can be feasibly performed by parallelizing the testing of candidate primers.

Second generation sequencing of messenger RNA (RNA-Seq) is becoming the method of choice to understand the functional effects of genetic variability and establish causal relationships between genetic variants and diseases. For cancer research, RNA-Seq provides most of the data needed to understand how the tumor evolves and discover new targets for treatment strategies such as immunotherapy. We presented a bioinformatics pipeline for detection of immunogenic cancer mutations from high throughput mRNA sequencing data. Within our pipeline we contributed to the research of methods for analysis of RNA-Seq data by developing a reads mapping strategy that seeks to fully exploit the information included in the genome reference sequence and the CCDS transcripts database to deal efficiently with the difficulties of mapping reads spanning exon junctions. We also proposed a bayesian model for SNV discovery and genotyping based on quality scores, which we called SNVQ. To assess the performance of our methods, we reanalyzed data from seven RNA-Seq lanes taken from blood cell tissue of a Hapmap individual to show that presented methods increase accuracy of the results under a wide veriety of circunstances. We released open source code implementing these techniques un-

der the GNU General Public License, as part of our NGSTools package available at `http://dna.engr.uconn.edu/software/NGSTools/`.

As accurate prediction of peptides presented by the MHC molecules requires phasing of close variants, we investigated the single individual haplotyping problem looking for an accurate and efficient alternative to obtain the phasing configuration for close SNVs. We presented a novel problem formulation for single individual haplotyping and a heuristic algorithm for this formulation. We started by assigning a score to each pair of fragments based on their common allele calls and then we used these score to formulate the problem as the cut of fragments that maximize an objective function, similar to the well known max-cut problem. Our algorithm initially finds the best cut based on a heuristic algorithm for max-cut and then builds haplotypes consistent with that cut. We have compared both accuracy and running time of ReFHap with other heuristic methods on both simulated and real data and found that ReFHap performs significantly faster than previous methods without loss of accuracy.

We finally integrated in our pipeline Primer3 to design primers for validation of predicted mutations through Sanger sequencing and NetMHC to predict immunogenic response for the peptides related with heterozygous mutations. We sequenced six different tumor samples at different coverage levels to test the accuracy of our predictions on real data. We confirmed through Sanger sequencing validation that we can identify single nucleotide variants with high accuracy even in cancer samples. We generated hundreds of epitope candidates for each sample which are currently under experimental validation. We finally verified that phasing of close variants allows to generate promising epitope candidates with larger score differences between the wild type and the mutated peptides.

As future work, we plan to extend the pipeline for detection of more kinds of transcriptome variations like indels in coding regions. Small indels make a promis-

ing source of epitopes because they may change the reading frame, producing peptides with larger differences from the wild type than those produced by SNVs. We also plan to integrate isoform reconstruction algorithms to find tumor specific transcripts [66].

Recent approaches to gene fusion detection in cancer [58] use mixed technology transcriptome sequencing, combining Illumina mRNA reads with longer (200-500bp) mRNA reads generated by 454 sequencing. However, sensitivity of this approach is limited by the relatively low sequencing depth afforded by the 454 technology. To overcome this limitation, we plan to use the Illumina platform to perform deep pair end sequencing of the cancer transcriptome, and identify gene fusion events using techniques similar to those developed for detecting large structural variation in cancer genomes [9, 50], which rely on detecting unexpected distances and/or orientations between pairs of mapped reads.

For prediction of CD8+ CTL epitopes presented by the MHC class I pathway we will use methods that integrate prediction of MHC I binding, TAP transport efficiency, and proteasomal cleavage [48, 93], which have been shown to be more accurate than prediction of MHC I binding alone. We will also predict tumor-specific CD4+ T cell epitopes, which have been shown to play a critical role in regulating immune responses, by employing the latest MHC II peptide binding prediction methods (reviewed in [55]). Since we are also performing mass spectrometry on the proteins presented on the surface of the cell, we plan to estimate the mass and charge of our predicted epitopes to match them with the data generated by mass spectrometry. This will increase confidence on existence and presentation of our predicted epitopes. Finally, we are adjusting the protocols to perform immunization experiments with promising peptides.

Bibliography

[1] H.T. Allawi and J. SantaLucia. Nearest neighbor thermodynamic parameters for internal G-A mismatches in DNA. *Biochemistry*, 37(8):2170–2179, 1998.

[2] H.T. Allawi and J. SantaLucia. Nearest-neighbor thermodynamics of internal A-C mismatches in DNA: Sequence dependence and pH effects. *Biochemistry*, 37(26):9435–9444, 1998.

[3] H.T. Allawi and J. SantaLucia. Thermodynamics of internal C-T mismatches in DNA. *Nucleic Acids Research*, 26(11):2694–2701, 1998.

[4] S. Angelov, B. Harb, S. Kannan, S. Khanna, and J. Kim. Efficient enumeration of phylogenetically informative substrings. *Journal of Computational Biology*, 14(6):701–723, 2007.

[5] S. Balla, S. Rajasekaran, and I.I. Măndoiu. Efficient algorithms for degenerate primer search. *International Journal of Foundations of Computer Science*, 18(4):899–910, 2007.

[6] Vikas Bansal and Vineet Bafna. HapCUT: an efficient and accurate algorithm for the haplotype assembly problem. *Bioinformatics*, 24(16):i153–i159, August 2008.

[7] Vikas Bansal, Aaron L. Halpern, Nelson Axelrod, and Vineet Bafna. An MCMC algorithm for haplotype assembly from whole-genome sequence data. *Genome Research*, 18:1336–1346, August 2008.

[8] Y. Bao, P. Bolotov, D. Dernovoy, B. Kiryutin, L. Zaslavsky, T. Tatusova, J. Ostell, and D. Lipman. The influenza virus resource at the national center for biotechnology information. *Journal of Virology*, 82(2):596–601, 2008.

[9] A. Bashir, S. Volik, C. Collins, V. Bafna, and B. J. Raphael. Evaluation of paired-end sequencing strategies for detection of genome rearrangements in cancer. *PLoS Computational Biology*, 4(4):e1000051, 2008.

[10] D.R. Bentley *et al.* Accurate whole human genome sequencing using reversible terminator chemistry. *Nature*, 456:53–59, 2008.

[11] F.D. Bona, S. Ossowski, K. Schneeberger, and G. Rätsch. Optimal spliced alignments of short sequence reads. *Bioinformatics*, 24(16):174–180, 2008.

[12] Dumitru Brinza and Alexander Zelikovsky. 2SNP: scalable phasing method for trios and unrelated individuals. *IEEE/ACM Transaction on Computational Biology and Bioinformatics*, 5(2):313–318, April-June 2008.

[13] Carola Burgtorf, Pamela Kepper, Margret R. Hoehe, Carsten Schmitt, Richard Reinhardt, Hans Lehrach, and Sascha Sauer. Clone-based systematic haplotyping (CSH): A procedure for physical haplotyping of whole genomes. *Genome Research*, 13:2717–2724, September 2003.

[14] H.K. Chang, J.H. Park, M.S. Song, T.K. Oh, S.Y. Kim, C.J. Kim, H. Kim, M.H. Sung, H.S. Han, Y.S. Hahn, and Y.K. Choi. Development of multiplex rt-PCR assays for rapid detection and subtyping of influenza type A

viruses from clinical specimens. *Journal of Microbiology and Biotechnology*, 18(6):1164–1169, 2008.

[15] I. Chepelev, G. Wei, Q. Tang, and K. Zhao. Detection of single nucleotide variations in expressed exons of the human genome using RNA-Seq. *Nucleic Acids Research*, 37(16):e106, 2009.

[16] V. Chvátal. A greedy heuristic for the set covering problem. *Mathematics of Operations Research*, 4:233–235, 1979.

[17] E.T. Cirulli, A. Singh, K.V. Shianna, D. Ge, J.P. Smith, J.M. Maia, E. L. Heinzen, J.J. Goedert, and D.B. Goldstein *et al.* Screening the human exome: a comparison of whole genome and whole transcriptome sequencing. *Genome Biology*, 11(5):R57, 2010.

[18] N. Cloonan, A.R. Forrest, G. Kolle, B.B. Gardiner, and G.J. Faulkner *et al.* Stem cell transcriptome profiling via massive-scale mRNA sequencing. *Nature Methods*, 5(7):613–619, 2008.

[19] F.S. Collins, E.D. Green, A.E. Guttmacher, and M.S. Guyer. A vision for the future of genomics research. *Nature*, 422:835–847, 2003.

[20] The International HapMap Consortium. A second generation human haplotype map of over 3.1 million SNPs. *Nature*, 449(18):851–861, 2007.

[21] V. Costa, C. Angelini, I. DeFeis, and A. Ciccodicola. Uncovering the complexity of transcriptomes with RNA-Seq. *Journal of Biomedicine and Biotechnology*, 2010:853916, 2010.

[22] M.D. Curran, J.S. Ellis, T.G. Wreghitt, and M.C. Zambon. Establishment of a UK national influenza H5 laboratory network. *Journal of Medical Microbiology*, 56:1263–1267, 2007.

[23] A. V. Dalca and M. Brudno. Genome variation discovery with high-throughput sequencing data. *Briefings in Bioinformatics*, 11(1):3–14, 2010.

[24] B. DasGupta, K.M. Konwar, I.I. Mandoiu, and A.A. Shvartsman. DNA-BAR: Distinguisher selection for DNA barcoding. *Bioinformatics*, 21(16):3424–3426, 2005.

[25] W. Dinkelbach. On nonlinear fractional programming. *Management Science*, 13:492–498, 1967.

[26] J. Duitama, D.M. Kumar, E Hemphill, M. Khan, I.I. Măndoiu, and C.E. Nelson. PrimerHunter: a primer design tool for PCR-based virus subtype identification. *Nucleic Acids Research*, 37(8):2483–2492, 2009.

[27] Jorge Duitama, Thomas Huebsch, Gayle McEwen, Eun-Kyung Suk, and Margret R. Hoehe. ReFHap: a reliable and fast algorithm for single individual haplotyping. In *BCB '10: Proceedings of the First ACM International Conference on Bioinformatics and Computational Biology*, pages 160–169, New York, NY, USA., August 2010. ©ACM, Inc. `http://doi.acm.org/10.1145/1854776.1854802`.

[28] S. Emrich, M. Lowe, and A. Delcher. PROBEmer: a web-based software tool for selecting optimal DNA oligos. *Nucleic Acids Research*, 31(13):3746–3750, 2003.

[29] B. Ewing and P. Green. Base-Calling of automated sequencer traces using phred. II. error probabilities. *Genome Research*, 8:186–194, 1998.

[30] J.P. Fitch, S.N. Gardner, T.A. Kuczmarski, S. Kurtz, R. Myers, L.L. Ott, T.R. Slezak, E.A. Vitalis, A.T. Zemla, and P.M. McCready. Rapid develop-

ment of nucleic acid diagnostics. *Proceedings of the IEEE*, 90(11):1708–1721, 2002.

[31] M. Gadberry, S. Malcomber, A. Doust, and E. Kellogg. Primaclade–a flexible tool to find conserved PCR primers across multiple species. *Bioinformatics*, 21(7):1263–1264, 2005.

[32] S.N. Gardner, T.A. Kuczmarski, E.A. Vitalis, and T.R. Slezak. Limitations of TaqMan PCR for detecting divergent viral pathogens illustrated by hepatitis A, B, C, and E viruses and Human Immunodeficiency Virus. *Journal of Clinical Microbiology*, 41(6):2417–2427, 2003.

[33] Loredana M. Genovese, Filippo Geraci, and Marco Pellegrini. SpeedHap: a fast and accurate heuristic for the single individual SNP haplotyping problem with many gaps, high reading error rate and low coverage. *IEEE/ACM Transaction on Computational Biology and Bioinformatics*, 5(4):492–502, October-December 2008.

[34] Alexander Gusev, Ion I. Măndoiu, and Bogdan Paşaniuc. Highly scalable genotype phasing by entropy minimization. *IEEE/ACM Transaction on Computational Biology and Bioinformatics*, 5(2):252–261, April-June 2008.

[35] Margret R. Hoehe. Haplotypes and the systematic analysis of genetic variation in genes and genomes. *Pharmacogenomics*, 4(5):547–570, September 2003.

[36] Margret R. Hoehe, Karla Köpke, Birgit Wendel, Klaus Rohde, Christina Flachmeier, Kenneth K. Kidd, Wade H. Berrettini, and George M. Church. Sequence variability and candidate gene analysis in complex disease: association of μ opioid receptor gene variation with substance dependence. *Human Molecular Genetics*, 9(19):2895–2908, September 2000.

[37] R.A. Holt and S.J.M. Jones. The new paradigm of flow cell sequencing. *Genome Research*, 18(6):839–846, 2008.

[38] O.J. Jabado, G. Palacios, V. Kapoor, J. Hui, N. Renwick, J. Zhai, T. Briese, and W.I. Lipkin. Greene SCPrimer: a rapid comprehensive tool for designing degenerate primers from multiple sequence alignments. *Nucleic Acids Research*, 34(22):6605–6611, 2006.

[39] D.S. Johnson. Approximation algorithms for combinatorial problems. *Journal of Computer and System Sciences*, 9:256–278, 1974.

[40] Jeffrey M. Kidd, Ze Cheng, Tina Graves, Bob Fulton, Richard K. Wilson, and Evan E. Eichler. Haplotype sorting using human fosmid clone end-sequence pairs. *Genome Research*, 18:2016–2023, October 2008.

[41] N. Kim and C. Lee. QPRIMER: a quick web-based application for designing conserved PCR primers from multigenome alignments. *Bioinformatics*, 23(17):2331–2333, 2007.

[42] T. Koressaar and M. Remm. Enhancements and modifications of primer design program Primer3. *Bioinformatics*, 23(10):1289–1291, 2007.

[43] I. Kozarewa, Z. Ning, M.A. Quail, M.J. Sanders, M. Berriman, and D.J. Turner. Amplification-free illumina sequencing-library preparation facilitates improved mapping and assembly of (G+C)-biased genomes. *Nature Methods*, 6(4):291–295, 2009.

[44] S. Kwok, S.Y. Chang, J.J. Sninsky, and A. Wong. A guide to the design and use of mismatched and degenerate primers. *Genome Research*, 3(10):S539–S547, 1994.

[45] E. Lander *et al.* Initial sequencing and analysis of the human genome. *Nature*, 409:860–921, 2001.

[46] B. Langmead, C. Trapnell, M. Pop, and S.L. Salzberg. Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biology*, 10:R25, 2009.

[47] M.A. Larkin, G. Blackshields, N.P. Brown, R. Chenna, P.A. McGettigan, H. McWilliam, F. Valentin, I.M. Wallace, A. Wilm, R. Lopez, J.D. Thompson, T.J. Gibson, and D.G. Higgins. ClustalW2 and ClustalX version 2. *Bioinformatics*, 23(21):2947–2948, 2007.

[48] M.V. Larsen, C. Lundegaard, K. Lamberth, S. Buus, S. Brunak, O. Lund, and M. Nielsen. An integrative approach to CTL epitope prediction: A combined algorithm integrating MHC class I binding, TAP transport efficiency, and proteasomal cleavage predictions. *European Journal of Immunology*, 35(8):2295–2303, 2005.

[49] M. Leber, L. Kaderali, A. Schonhuth, and R. Schrader. A fractional programming approach to efficient DNA melting temperature calculation. *Bioinformatics*, 21(10):2375–2382, 2005.

[50] S. Lee, E. Cheran, and M. Brudno. A robust framework for detecting structural variations in a genome. *Bioinformatics*, 24(13):i59–i67, 2008.

[51] S. Levy *et al.* The diploid genome sequence of an individual human. *PLoS Biology*, 5(10):e254+, 2007.

[52] H. Li, B. Handsaker, A. Wysoker, T. Fennell, and J. Ruan *et al.* The sequence Alignment/Map format and SAMtools. *Bioinformatics*, 25(16):2078–2079, 2009.

[53] H. Li, J. Ruan, and R. Durbin. Mapping short DNA sequencing reads and calling variants using mapping quality scores. *Genome Research*, 18(1):1851–1858, 2008.

[54] R. Li, Y. Li, X. Fang, H. Yang, J. Wang, K. Kristiansen, and J. Wang. SNP detection for massively parallel whole-genome resequencing. *Genome Research*, 19:1124–1132, 2009.

[55] H.H. Lin, G. L. Zhang, S. Tongchusak, E. L. Reinherz, and V. Brusic. Evaluation of MHC-II peptide binding prediction servers: applications for vaccine research. *BMC Bioinformatics*, 9(Suppl 12):S22, 2008.

[56] C. Linhart and R. Shamir. The degenerate primer design problem. *Bioinformatics*, 18(90001):S172–S181, 2002.

[57] C. Lundegaard, K. Lamberth, M. Harndahl, S. Buus, O. Lund, and M. Nielsen. NetMHC-3.0: accurate web accessible predictions of human, mouse and monkey MHC class I affinities for peptides of length 8-11. *Nucleic Acids Research*, 36(Web Server issue):W509–W512, 2008.

[58] C.A. Maher, C. Kumar-Sinha, X. Cao, S. Kalyana-Sundaram, B. Han, X. Jing, L. Sam, T. Barrette, N. Palanisamy, and A.M. Chinnaiyan. Transcriptome sequencing to detect gene fusions in cancer. *Nature*, 458(5):97–101, 2009.

[59] Jonathan Marchini *et al.* A comparison of phasing algorithms for trios and unrelated individuals. *American Journal of Human Genetics*, 78(3):437–450, January 2006.

[60] S. Marguerat and J. Bähler. RNA-seq: from technology to biology. *Cellular and Molecular Life Sciences*, 67(4):569–579, 2009.

[61] J.C. Marioni, C.E. Mason, S.M. Mane, M. Stephens, and Y. Gilad. RNA-seq: An assessment of technical reproducibility and comparison with gene expression arrays. *Genome Research*, 18:1509–1517, 2008.

[62] Kevin J. McKernan *et al.* Sequence and structural variation in a human genome uncovered by short-read, massively parallel ligation sequencing using two-base encoding. *Genome Research*, 19:1527–1541, 2009.

[63] R. Morin, M. Bainbridge, A. Fejes, M. Hirst, M. Krzywinski, T. Pugh, H. McDonald, R. Varhol, S. Jones, and M. Marra. Profiling the HeLa S3 transcriptome using randomly primed cDNA and massively parallel short-read sequencing. *Biotechniques*, 45(1):81–94, 2008.

[64] A. Mortazavi, B.A. Williams, K. McCue, L. Schaeffer, and B. Wold. Mapping and quantifying mammalian transcriptomes by RNA-Seq. *Nature methods*, 5:621–628, 2008.

[65] Y. Nakamura, Y. Noguchi, E. Satoh, A. Uenaka, S. Sato, T. Kitazaki, T. Kanda, H. Soda, E. Nakayama, and S. Kohno. Spontaneous remission of a non-small cell lung cancer possibly caused by anti-NY-ESO-1 immunity. *Lung Cancer*, 65(1):119–122, 2009.

[66] M. Nicolae, S. Mangul, I.I. Mandoiu, and A. Zelikovsky. Estimation of alternative splicing isoform frequencies from rna-seq data. In M. Singh and V. Moulton, editors, *Proc. 10th Workshop on Algorithms in Bioinformatics*, Lecture Notes in Computer Science, pages 202–214, 2010.

[67] M. Nielsen, C. Lundegaard, P. Worning, C.S. Hvid, K. Lamberth, S. Buus, S. Brunak, and O. Lund. Improved prediction of mhc class i and class ii epitopes using a novel gibbs sampling approach. *Bioinformatics*, 20(9):1388–1397, 2004.

[68] R. Owczarzy, Y. You, B.G. Moreira, J.A. Manthey, L. Huang, M.A. Behlke, and J.A. Walder. Effects of sodium ions on DNA duplex oligomers: Improved predictions of melting temperatures. *Biochemistry*, 43(12):3537–3554, 2004.

[69] R.D. Page. TreeView: An application to display phylogenetic trees on personal computers. *Computer Applications in the Biosciences*, 12(4):357–358, 1996.

[70] Alessandro Panconesi and Mauro Sozio. Fast Hare: a fast heuristic for single individual SNP haplotype reconstruction. *In: Jonassen, I., Kim, J. (eds.) WABI 2004. LNCS (LNBI)*, 3240:266–277, September 2004.

[71] A. Panjkovich and F. Melo. Comparison of different melting temperature calculation methods for short DNA sequences. *Bioinformatics*, 21(6):711–722, 2005.

[72] N. Peyret, P.A. Seneviratne, H.T. Allawi, and J. SantaLucia. Nearest-Neighbor thermodynamics and NMR of DNA sequences with internal A-A, C-C, G-G, and T-T mismatches. *Biochemistry*, 38(12):3468–3477, 1999.

[73] A. Phillippy, J. Mason, K. Ayanbule, D. Sommer, E. Taviani, A. Huq, R. Colwell, I. Knight, and S. Salzberg. Comprehensive DNA signature discovery and validation. *PLOS Computational Biology*, 3(5):887–894, 2007.

[74] K. Pruitt, J. Harrow, and R.A. Harte*et al.* The consensus coding sequence (CCDS) project: Identifying a common protein-coding gene set for the human and mouse genomes. *Genome Research*, 19:1316–1323, 2009.

[75] H. Rammensee, J. Bachmann, N.P. Emmerich, O.A. Bachor, and S. Stevanovic. SYFPEITHI: database for mhc ligands and peptide motifs. *Immunogenetics*, 50(3-4):213–219, 1999.

[76] H. Rammensee, T. Weinschenk, C. Gouttefangeas, and S. Stevanovic. Towards patient-specific tumor antigen selection for vaccination. *Immunological Reviews*, 188(1):164–176, 2002.

[77] B. Rhead, D. Karolchik, R.M. Kuhn, A.S. Hinrichs, and A.S. Zweig *et al.* The UCSC genome browser database: update 2010. *Nucleic Acids Resarch*, 38(suppl 1):D613–D619, 2010.

[78] Romeo Rizzi, Vineet Bafna, Sorin Istrail, and Giuseppe Lancia. Practical algorithms and fixed-parameter tractability for the single individual SNP haplotyping problem. *In Proceedings of the Second International Workshop on Algorithms in Bioinformatics*, 2452:29–43, September 2002.

[79] J. Rouillard, M. Zuker, and E. Gulari. OligoArray 2.0: design of oligonucleotide probes for DNA microarrays using a thermodynamic approach. *Nucleic Acids Research*, 31(12):3057–3062, 2003.

[80] S. Rozen and H.J. Skaletsky. Primer3 on the WWW for general users and for biologist programmers. In S. Krawetz and S. Misener, editors, *Bioinformatics Methods and Protocols: Methods in Molecular Biology*, pages 365–386. Humana Press, Totowa, NJ, 2000.

[81] Sartaj Sahni and Teofilo Gonzales. P-complete problems and approximate solutions. *In Proceedings of the 15th Annual Symposium on Switching and Automata Theory. IEEE*, pages 14–16, October 1974.

[82] J. SantaLucia and D. Hicks. The thermodynamics of DNA strucutural motifs. *Annual Review of Biophysics and Biomolecular Structure*, 33:415–440, 2004.

[83] Daniel J. Schaid. Evaluating associations of haplotypes with traits. *Genetic Epidemiology*, 27:348–364, November 2004.

[84] Paul Scheet and Matthew Stephens. A fast and flexible statistical model for large-scale population genotype data: Applications to inferring missing genotypes and haplotypic phase. *American Journal of Human Genetics*, 78(4):629–644, February 2006.

[85] Stephan C. Schuster *et al.* Complete khoisan and bantu genomes from southern africa. *Nature*, 463:943–947, 2010.

[86] M. Snyder, J. Du, and M. Gerstein. Personal genome sequencing: current approaches and challenges. *Genes & Development*, 24:423–431, 2010.

[87] R. Souvenir, J. Buhler, G. Stormo, and W. Zhang. Selecting degenerate multiplex PCR primers. In *Proc. 3rd Intl. Workshop on Algorithms in Bioinformatics (WABI)*, pages 512–526. Springer Berlin / Heidelberg, 2003.

[88] E. Spackman, D.A. Senne, L.L. Bulaga, S. Trock, and D.L. Suarez. Development of multiplex real-time RT-PCR as a diagnostic tool for avian influenza. *Avian Diseases*, 47(s3):1087–1090, September 2003.

[89] Matthew Stephens and Peter Donnelly. A comparison of bayesian methods for haplotype reconstruction from population genotype data. *American Journal of Human Genetics*, 73(5):1162–1169, October 2003.

[90] D.L Suarez, A. Das, and E.H. Ellis. Review of rapid molecular diagnostic tools for avian influenza. *Avian Diseases*, 51:201–208, 2007.

[91] M. Sultan *et al.* A global view of gene activity and alternative splicing by deep sequencing of the human transcriptome. *Science*, 321(5891):956–960, 2008.

[92] F. Tang, C. Barbacioru, Y. Wang, E. Nordman, and C. Lee *et al.* mRNA-Seq whole-transcriptome analysis of a single cell. *Nature Methods*, 6(5):377–382, 2009.

[93] S. Tenzer, B. Peters, O. Schoor, C. Lemmel, M.M. Schatz, P.M. Kloetzel, H.G. Rammensee, H Schild, and H.G. Holzhütter. Modeling the MHC class I pathway by combining predictions of proteasomal cleavage, TAP transport and MHC class I binding. *Cellular and Molecular Life Sciences*, 62(9):1025–1037, 2005.

[94] C. Trapnell, B.A. Williams, G. Pertea, A. Mortazavi, G. Kwan, M.J. Baren, S.L. Salzberg, B.J. Wold, and L. Pachter. Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nature biotechnology*, 28(5):511–515, 2010.

[95] B.B. Tuch, R.R. Laborde, X. Xu, J. Gu, and C.B. Chung *et al.* Tumor transcriptome sequencing reveals allelic expression imbalances associated with copy number alterations. *Plos One*, 5(2):e9317, 2010.

[96] University of Tartu, Department of Bioinformatics. SLICSel 1.1, http://bioinfo.ut.ee/slicsel/.

[97] Jianxin Wang, Minzhu Xie, and Jianer Chen. A practical exact algorithm for the individual haplotyping problem MEC/GI. *Algorithmica*, 56(3):283–296, March 2010.

[98] J Wang *et al.* The diploid genome sequence of an Asian individual. *Nature*, 456:60–65, 2008.

[99] X. Wei, D. Khun, and G. Narasimhan. Degenerate primer design via clustering. In *Proceedings of the IEEE Computer Society Bioinformatics Conference*, pages 75–83, 2003.

[100] D.A. Wheeler *et al.* The complete genome of an individual by massively parallel DNA sequencing. *Nature*, 452(7189):872–876, 2008.

[101] Jingli Wu, Jianxin Wang, and Jianer Chen. A parthenogenetic algorithm for single individual SNP haplotyping. *Engineering Applications of Artificial Intelligence*, 22(3):401–406, April 2009.

[102] Minzhu Xie, Jianxin Wang, and Jianer Chen. A model of higher accuracy for the individual haplotyping problem based on weighted SNP fragments and genotype with errors. *Bioinformatics*, 24(13):i105–i113, July 2008.

[103] Z. Xie, Y.S. Pang, J. Liu, X. Deng, X. Tang, J. Sun, and M.I. Khan. A multiplex RT-PCR for detection of type A influenza virus and differentiation of avian H5, H7, and H9 hemagglutinin subtypes. *Molecular and Cellular Probes*, 20(3-4):245–249, 2006.

[104] Y. Xing, T. Yu, Y. N. Wu, M. Roy, J. Kim, and C. Lee. An expectation-maximization algorithm for probabilistic reconstructions of full-length isoforms from splice graphs. *Nucleic Acids Research*, 34(10):3150–3160, 2006.

[105] J. W. Yewdell, E. Reitz, and J. Neefjes. Making sense of mass destruction: quantitating MHC class I antigen presentation. *Nature Reviews Immunology*, 3:952–961, 2003.

[106] J. Zheng, T. Svensson, K. Madishetty, T. Close, T. Jiang, and S. Lonardi. OligoSpawn: a software tool for the design of overgo probes from large unigene datasets. *BMC Bioinformatics*, 7(7), 2006.