Abstract

Integrated Analysis of Partial Sampling Techniques in Bioinformatics

by

Jiang Du

2010

With the development of microarray and the more recent next-generation sequencing technologies, researchers in genomics have been able to conduct large-scale and high-throughput experiments on the DNA level in order to investigate the abundance of different gene transcripts in the cell, and also to identify structural variants in individual genomes. The biological data from such experiments are usually signal intensities or sequence contents of DNA fragments, which can be viewed as partially observed samples from a pool of complete objects (e.g. short DNA fragments from a mixture of full-length transcript sequences). What is more, these partial samples can be obtained via different technologies, each with its own characteristic error rate, sampling bias and per-sample cost. This thesis describes methods for integrated analysis of such samples in different problems, where computational frameworks and solutions are established to quantitatively parameterize statistical models and efficient algorithms are designed to estimate the variance of the method's accuracy. Both simulation and analytical methods are developed to find the optimal low-cost integration of different sampling techniques in each experiment design. The specific problems being considered include 1) systematically selecting unlabeled DNA regions for validation to train a predictive model, 2) integrated analysis of fragmented DNA sequences to estimate

the distribution of full-length gene transcripts, and 3) conducting efficient simulations to model the local de novo assembly process in individual genome re-sequencing. A key aspect of some of the above problems is establishing fast algorithms to compute a corresponding Fisher information based measurement for performance estimation.

Integrated Analysis of Partial Sampling Techniques in Bioinformatics

A Dissertation

Presented to the Faculty of the Graduate School

of

Yale University

in Candidacy for the Degree of

Doctor of Philosophy

by

Jiang Du

Dissertation Directors:

Mark Gerstein and Drew McDermott

May 2010

Declaration

I declare that this thesis represents my own work, except where due acknowledgement is made, and that it has not been previously included in a thesis, dissertation or report submitted to this university or to any other institution for a degree, diploma or other qualifications.

. . . .

Jiang Du

May 2010

Acknowledgments

I am in great debt to many who have supported and encouraged me during my past five years in New Haven. Without them, it would not have been possible for me to reach this final stage of doctoral study.

First of all, I thank my family, for their being supportive to let me study abroad, and for their continuous support in all aspects. I thank my beloved Cecile for traveling this long journey with me.

I would like to express my sincere gratitude to my adviser Mark Gerstein, who always does everything he could in helping my research. He has devoted an uncountable number of hours to detailed discussions about my research, study, and many other interesting topics.

I thank my adviser Drew McDermott for his great efforts in helping me progress through the many stages of my study. The discussions with him have always been intriguing and thought-inspiring.

I thank my thesis committee members Martin Schultz for his continuous support on my research since my first year at Yale, and Roderic Guigo for his advice on my work.

I would like to thank my fellow colleagues in the Gerstein lab, especially Joel Rozowsky, Kevin Yip, Zhengdong Zhang, Hugo Lam, Jing Leng, Lukas Habegger, Andrea Sboner and everyone in the Genome Technology/Assembly groups. I have enjoyed very much working with people in the lab with diverse backgrounds and characters, and have learned a lot from each and every one of them, academically and non-academically.

I would also like to thank Michael Snyder and Jiaqian Wu at Stanford University, Robert Bjornson and Nicholas Carriero in the Yale University Biomedical High Performance Computing Center, Yong Kong in W.M. Keck Foundation Biotechnology Resource Laboratory at Yale, and the anonymous reviewers of my research articles. I would like to acknowledge the AL Williams Professorship funds, and the grants from NIH that have directly and indirectly supported my research.

Last but not least, I would like to thank my many friends in the Yale Chinese community, the Computer Science department, and other groups and individuals, who have greatly enriched my life throughout these years.

Contents

Chapter 1

Introduction

Many machine learning and data mining studies focus on building and parameterizing mathematical models explaining the samples that are obtained. Usually a model $Model(\Theta)$ will be defined with parameters Θ, and the samples S will be used to find the Θ that optimizes a certain objective function $Obj(Model, S, \Theta)$. In applications of these techniques to bioinformatics problems, results of large-scale biological experiments are the samples, and researcher attempt to build quantitative models based on these samples to answer questions with biological importance. The type and number of samples that can be obtained are not only limited by the capability of existing experimental (sampling) technologies, but also affected by the experimental cost and budget. What is more, there are often multiple sampling technologies generating similar type of experimental data, each with its own cost and characteristics, and a researcher will need to decide which one to choose or even how to carry out an experiment using a combination of such sampling techniques, in order to derive the most accurate answer from the resulting experimental data. One typical example is in individual re-sequencing projects [45, 81, 79, 64, 6, 53, 2, 39, 17, 46, 50, 59, 60], where there exist at least three major categories of sequencing methods:

- Long, expensive and slow: generates long sequencing reads (samples) of $\sim$ 800bp.
- Medium, cheaper, and high-throughput: generates medium-size samples of $\sim$ 250bp, much cheaper.
- Short, cheapest and high-throughput: generates short samples of $\sim 30 - 50$bp, $\sim$ 10 times cheaper than the medium sequencing technology.

The researchers will not only have to find a method that can infer an individual's diploid genome sequence ($\sim$ 6 billion bases long) based on the reads (samples) generated by these sequencing methods, but also need to design ahead of time a way to carry out the experiment using all the sequencing technologies available with a relatively low budget (with an infinite budget in time and money, the longer sequencing method will be a better choice, since it can intrinsically distinguish different repetitive regions in the human genome, which is one of the major causes of mis-assembly of the genome sequence).

Also, the samples obtained in such experiments are usually partial samples when compared to the object being observed. The sequencing technologies mentioned above can only extract the content of a DNA sequence fragment of limited length, and information from individual reads will have to be combined in order to infer either the full DNA sequence of an individual's genome, or to estimate the composition of full-length transcript isoforms of a certain gene of interest.

In this thesis, we focus on the integrated analysis of such partial samples obtained from different sampling techniques, and seek answers and insights to the following questions:

1. How to construct different $Model(\Theta)$s and formulate $Obj(Model, S, \Theta)$s for some important biological experiments, where S is a set of partial samples from different sampling methods?
2. How to find Θ that optimizes $Obj(Model, S, \Theta)$?
3. How to estimate the accuracy of our Θ estimation?
    - Even in simulations where the true Θ is known, is there a more efficient way to estimate the accuracy of our Θ estimation than the brute-force method?
4. Given a fixed total budget, how to find a low-cost integration of different sampling methods to get the best outcome in estimating Θ?

In Chapter 2, we investigate a combination of two different partial sampling methods on the genome, which contains an enormous number of small regions $\{r_1, r_2, \cdots, r_N\}$: one is the tiling array technique, which generates measurements for the abundance of transcriptional activity $\{X_1, \cdots, X_N\}$ for all the small regions; the other is experimental validation of a selected set of consecutive regions $\{r_i, \cdots, r_j\}$, which can accurately identify their corresponding transcriptional activities $\{Y_i, \cdots, Y_j\}$. The former technology is cheap and can be performed in a high-throughput fashion, though its measurement contains an considerable level of noise. The latter one is much more expensive and time-consuming, and is able to provide very precise results. Usually a sequential data generation model $Model(Y \rightarrow X, \Theta)$ is built to infer the actual transcriptional activity Y based on the resulting measurements X from the first technique, and the results $\{Y_{i_1}, \cdots, Y_{j_1}\}, \cdots, \{Y_{i_M}, \cdots, Y_{j_M}\}$ from the second sampling technique on a selected set of DNA regions are used to refine the parameters Θ in the model to more accurately predict Y from X. Besides proposing an effective $Model(Y \rightarrow X, \Theta)$ and the corresponding supervised model training method, we are also interested in how to optimally select the $\{r_{i_1}, \cdots, r_{j_1}\}, \cdots, \{r_{i_M}, \cdots, r_{j_M}\}$ as input to the second sampling method to obtain a set of $\{Y_{i_1}, \cdots, Y_{j_1}\}, \cdots, \{Y_{i_M}, \cdots, Y_{j_M}\}$ that can best train the model. We consider these selected partial samples as "deterministic" since we can decide ahead of time the exact position to sample from.

We thus present in Chapter 2 an efficient HMM framework which systematically incorporates *validated biological knowledge* into tiling array data analysis. This framework, which consists of a *MaxEntropy* sample selection algorithm and HMM learning and decoding approaches, is proposed based on *HTPIO*, an idealized definition of the tiling array analysis problem. Empirical results of our methods in the framework on a simulated dataset, a transcriptional dataset and a ChIP-chip dataset show that our framework effectively handles large datasets, even with a relatively noisy training set. Our work differs from previous studies in tiling array data analysis by specifically taking *validated biological knowledge* into consideration and systematically incorporating it using an empirically tested *MaxEntropy* sample selection scheme for optimal analysis. These features ensure the good performance of our framework with even a relatively small gold standard training set, which has not been specifically considered by previous methods. In this way our framework can consistently analyze tiling array data across a number of experiments, and can process different types of array data automatically, without the need to manually set additional parameters.

In Chapter 3, we discuss the problem of integrating different sequencing techniques to quantify the relative abundance of different isoform transcripts, which can be generalized to the problem of estimating the distribution based on partial samples from different sampling techniques. We first introduce a statistical framework to model the generative process of the partial samples, using a “pluggable” function G to allow flexible incorporation of different sampling characteristics, and then present the original problem as a maximum likelihood estimation (MLE) problem, with an iterative solution based on expectation maximization, which guarantees a local optimal answer. This provides a solution to the question of estimating a distribution based on partial samples. The partial samples being considered here are obviously “non-deterministic” since the sequencing process is in general random.

In order to further investigate the problem involving partial samples, we also introduce in Chapter 3 a Fisher information matrix (FIM) based heuristic to estimate the variance of the previously presented MLE solution. Also, in order to accelerate the computation of this measurement, we introduce the concept of equivalent partial samples and develop a fast algorithm, Algorithm 3, to accurately calculate FIM, achieving a speedup of ~ 500 times compared to the brute-force method. Simulation results on both hypothetical and real gene models also show that our FIM-based heuristic gives good approximation to the value of $Average\left(var(\hat{\theta}_k)\right)$, and accurately predicts the numeric order of this value under different conditions. With this metric, we are also able to demonstrate examples of how to efficiently find low-cost combinations of different sampling techniques to best estimate the isoform compositions in RNA-seq experiments. Although we are only using individual genes as examples, once we have good assumptions of expression levels of different genes, this procedure can be generalized to all the genes for the low-cost design of actual whole genome RNA-seq experiments.

In Chapter 4, we still focus on the partial samples from sequencing experiments, which are used to perform individual genome re-sequencing. Recently, there has been great excitement about the proliferation of new sequencing (sampling) technologies (e.g. medium and short read sequencing from companies such as 454 and SOLiD, and high-density oligo-arrays from Affymetrix and NimbelGen), with even more expected to appear. The costs and sensitivities of these technologies differ considerably from each other. As an important goal of personal genomics is to reduce the cost of re-sequencing to an affordable point, it is worthwhile to consider optimally integrating these sequencing technologies. Here, we build a simulation toolbox that will help us optimally combine different technologies for genome re-sequencing, especially in reconstructing large structural variants (SVs). SV re-construction is considered the most challenging step in human genome re-sequencing. (It is sometimes even harder than de novo assembly of small genomes because of the duplications and repetitive sequences in the human genome.)

On one hand, the re-sequencing process is a complex procedure that can hardly be accurately described in closed-form; on the other, executing real assembly algorithms on large sets of reads will make large-scale simulation computationally intractable. To this end, we formulate canonical problems that are representative of issues in reconstruction and are of small enough scale to be computationally tractable and simulatable. Using semi-realistic simulations, we show how we can combine different technologies to optimally solve the assembly at low cost. With mapability maps, our simulations efficiently handle the inhomogeneous repeat-containing structure of the human genome and the computational complexity of practical assembly algorithms. They quantitatively show how combining different read lengths is more cost-effective than using one length, how an optimal mixed sequencing strategy for reconstructing large novel SVs usually also gives accurate detection of SNPs/indels, how paired-end reads can improve reconstruction efficiency, and how adding in arrays is more efficient than just sequencing for disentangling some complex SVs. Our strategy should facilitate the sequencing of human genomes at maximum accuracy and low cost.

We conclude the thesis in Chapter 5 and point out potential future directions.

Part I

Optimal Utilization of Deterministic Sampling Techniques

Chapter 2

Optimal Deterministic Sampling in A Supervised Hidden Markov Model Framework

2.1 Introduction

2.1.1 Motivation

Tiling arrays are used to survey genomic transcriptional activity [8, 13, 37, 67, 71] and transcription factor binding sites [9, 11, 33] at high resolution. The raw/preprocessed data from tiling array experiments are first processed by certain analysis methods, which produce a list of predicted genomic "active regions". These are either transcriptionally active regions (TARs)/transcribed fragments (transfrags) [8, 13, 36, 67] or transcription factor binding sites. Usually a subset of these regions is further studied by experimental validation, which answers the question of whether these regions are actually active or not.

With the beginning of projects such as ENCODE [21], which aims to annotate the genome sequence with the function of specific elements (e.g. whether they are regulatory sites, exons or introns), the large scale tiling array experiments that are carried out present a number of new challenges. One of these is how to build up an existing *knowledge base of validated biological information* about genomic elements such as the location of exons and introns or of transcription factor binding sites, and how to use this knowledge base in combination with the tiling array data on a limited region of the genome to construct a predictive model that we can extrapolate to the rest of the genome in order to best segment it into functional elements.

We also have the related problem of how to grow this knowledge base of validated biological information systematically so as to do the extrapolation most efficiently. We envision that it will not be possible to validate every single ChIP-chip experiment (which determines the binding sites of transcription factors) result, or every single exon in the human genome using RT-PCR (which can amplify and simultaneously quantify a targeted DNA molecule). However, we can imagine that following the large scale tiling array experiments there will be medium-scale validation experiments done on thousands of predicted binding sites and gene structures to try to verify them. The question is: how should these binding sites and gene structures for validation be picked? They could, of course, be selected in terms of having the best scores, but one would like to pick them so as to derive a model that would be best able to analyze the remainder of the data accurately.

Here we tackle both of these challenges by proposing a hidden Markov model (HMM) [65] framework which integrates the existing *validated biological knowledge* about gene structures and transcription factor binding sites, and then uses this encapsulated biological knowledge to segment tiling array data. In particular, we also show how one can systematically pick un-annotated or unlabeled regions from the tiling array data for further validation to grow the validated biological knowledge base of labeled examples in order to get a maximally predictive model [19].

We do our analysis side by side on both transcriptional data and ChIP-chip binding site data. We have two reasons for this. First of all, it shows the general utility of the approach that we can apply the same formalism to tiling array data from both types of experiments. Second, since data from the two different experiments have different levels of validated biological knowledge, it allows us to see how our formalism performs in two areas with different amounts of knowledge. Finally, because we can get a better handle of how things work on the better studied transcriptional data, we can have great confidence that we are applying a correct approach when segmenting the ChIP-chip data.

2.1.2 Previous work

In tiling array data analysis, the goal is to identify genomic active regions with high signal intensities. This procedure can not be implemented in a naïve fashion, due to the noise in the background and the possible low signal intensities in some active regions [30, 68]. Different statistical algorithms have been developed to process the tiling array data. Earlier examples include pseudo-median threshold with maxgap/minrun [38], p-value cutoff with maxgap/minrun [8], sliding-window PCA with MD [71], and variance stabilization [24]. More recently, several HMM approaches and HMM variants have been developed [34, 47, 52, 85, 41]. Flicek *et al.* (personal communication) have also applied an HMM to ChIP-chip data resulting from tiling array signals characteristic of histone modifications.

Some of these existing methods, such as maxgap/minrun [8, 36], involve parameters that have to be decided manually. HMM approaches, formerly introduced in the field of sequence analysis [20, 38, 42], have the advantage of not using any additional parameters other than the model itself. [47] proposed the construction of a two-state HMM for ChIP-chip data partially based on the results of Affymetrix SNP arrays [48]. [34], more recently, proposed a more general Unbalanced Mixture Subtraction (UMS) approach to recover different emission distributions in a HMM from a mixture distribution. However, in some cases, there may exist neither corresponding experimental results that can be utilized to build the HMM, nor validated biological knowledge comprehensive enough for an unbiased evaluation in the UMS analysis.

On the other hand, the use of partially validated knowledge about the array data, such as gene annotation or experimental validation results on small genomic regions, has not been specifically considered by existing methods; and there does not exist a systematic framework to optimally obtain and utilize this kind of knowledge in tiling array analysis. Such a framework will have the potential to better assist the analysis of tiling array data, as the related validated knowledge becomes more abundant and accurate via experimental validations.

2.1.3 Methodology

In this paper we propose a new supervised scoring framework based on HMM that will consistently score different types of tiling array data by incorporating validated biological knowledge. As our framework will be based on both transcriptional and regulatory data, we can demonstrate its efficiency on the better described transcriptional data so that we have greater confidence when applying it to the ChIP-chip data.

An integral part of our strategy is developing a scheme to intelligently select sub-regions for validation, in order to better build up gold standard sets to incorporate into our statistical model. We investigate the performances of different sample selection schemes described in section 2 on a simulated dataset in section 4, and propose to employ the *MaxEntropy* scheme as a measure for sample selection: we want to select sub-regions that have the highest entropies for experimental validation first, so as to effectively build up the validated biological knowledge for our HMM approach.

After the sample sub-regions are selected and their corresponding state sequences are obtained via further validation experiments or according to existing validated biological knowledge, a frequency-based supervised learning algorithm is applied to build the HMM and then the Viterbi algorithm is utilized to compute the most likely state sequence for the whole sequence of array signals. Since current experimental validation data are insufficient to apply our *MaxEntropy* sampling scheme, we also propose alternative methods for choosing sample sub-regions. As described in section 3, for transcriptional tiling array experiments, a four-state HMM can be constructed by learning from the sequences of probes which fall into regions of the corresponding gene annotation. For ChIP-chip data, the knowledge of gene annotation is again relevant to the identification of binding sites, because transcription factor binding sites (TFBS) are usually considered to be enriched in upstream regions of genes and unlikely to occur in inner regions of genes. By incorporating this knowledge, a two-state HMM can be constructed for further analysis. Empirical results in section 4 show that our methods effectively handle large datasets, even with relatively noisy training data.

2.2 Methods

2.2.1 Idealized definitions of the problem

In this section we give two idealized definitions of the tiling-array analysis problem, which will form the basis of our core algorithms on both sample sub-region selection and HMM analysis based on the selected samples.

Definition 1 (Idealized HMM Tiling Problem (*HTP*))**.** An idealized HMM tiling problem is a tuple $\langle D, C_{sample}, O \rangle$, where D is the emission sequence corresponding to a hidden state sequence S generated by an unknown HMM M, C_{sample} is the constraint on how sample sub-regions can be selected in D (e.g. the maximum length of each sample sub-sequence), and O is a labeling oracle (an imaginary black box which is able to answer certain questions) that can discover the corresponding hidden state sequence of any sample sub-region in D. A solution to the problem first selects a set of sample sub-regions in D according to the constraint C_{sample}, asks the labeling oracle O about the corresponding state sequences of these sample sub-regions, then efficiently computes a model M' for D and outputs the corresponding state sequence S' for D.

As shown in Figure 2.1A, S and D are generated by an HMM M, and correspond to the biological state (for instance, transcribed or not transcribed) sequence and signal intensity sequence of the probes in the array, preferably after necessary preprocessing such as normalization. The length of the sequence, L, corresponds to the size of the tiling array. The solution to the problem, which is also the framework we propose, first selects m sample sub-regions $\{U_1, U_2, ..., U_m\}$ in D according to the sampling constraint C_{sample}, and passes them to the labeling oracle O, which corresponds to an experimenter who refers to *validated*

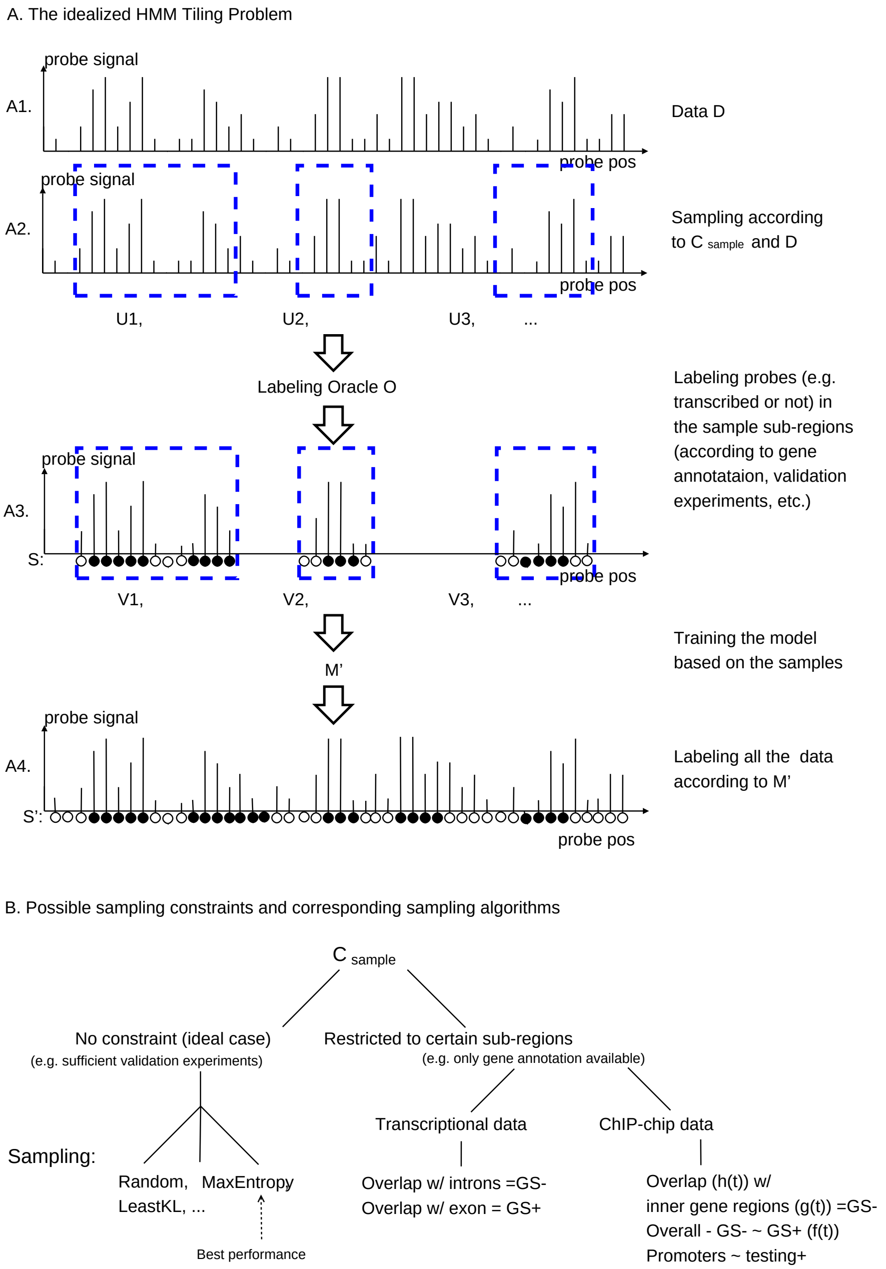

A. The idealized HMM Tiling Problem
A1.
probe signal
probe pos
Data D
A2.
probe signal
probe pos
Sampling according to C sample and D
U1,
U2,
U3,
...
Labeling Oracle O
Labeling probes (e.g. transcribed or not) in the sample sub-regions (according to gene annotataion, validation experiments, etc.)
A3.
probe signal
S:
probe pos
V1,
V2,
V3,
...
M'
Training the model based on the samples
A4.
probe signal
S':
probe pos
Labeling all the data according to M'
B. Possible sampling constraints and corresponding sampling algorithms
C sample
No constraint (ideal case)
(e.g. sufficient validation experiments)
Restricted to certain sub-regions
(e.g. only gene annotation available)
Transcriptional data
ChIP-chip data
Sampling:
Random, MaxEntropy
LeastKL, ...
Best performance
Overlap w/ introns =GS-
Overlap w/ exon = GS+
Overlap (h(t)) w/
inner gene regions (g(t)) =GS-
Overall - GS- ~ GS+ (f(t))
Promoters ~ testing+

Figure 2.1. Idealized HMM tiling-array analysis problem. (A) Idealized HMM tiling problem. (B) Sampling constraints and corresponding strategies.

biological knowledge (existing annotation, validation experiments, etc.) and then discovers the hidden state (label) sequences $\{V_1, V_2, ..., V_m\}$ for these small subsets of neighboring probes in the array. These sub-sequences of U_is and V_is form the samples/training set of our analysis methods. A model M' is then learned based on this training set, and processed by a decoding algorithm on D, which outputs the predicted corresponding state sequence S' for D.

The sampling constraint C_{sample} corresponds to the possible limitations in selecting sample sub-regions in real tiling array problems. As shown in Figure 2.1B, when experimental validations can be done on any set of genomic sub-regions, there will be no constraint on sampling at all and C_{sample} will be equal to null/empty. In the other extreme, if no further validation experiments can be done and the only available validated knowledge is the gene annotation related to the transcriptional tiling experiment, C_{sample} will only allow those sub-regions inside the gene annotation to be selected (otherwise the labeling oracle will fail to label all the sample sub-regions). One can imagine intermediate situations between these extremes.

HTP differs from the real problem of tiling array data analysis in two main aspects. On one hand, the actual state sequence S of the array data is not necessarily generated by a certain HMM. Such an HMM assumption is stated in *HTP* not only because that it is a reasonable approximation to the real problem, whose data fits the sequential nature of a HMM, but also because it is necessary for further performance analysis of the solutions to this problem. On the other hand, the labeling oracle O (e.g. experimental validation) in real problems is not always perfect and can make mistakes, from which we can give a generalization of *HTP* in the following definition:

Definition 2 (Idealized HMM Tiling Problem with an Imperfect Oracle (*HTPIO*))**.** An idealized HMM tiling problem with an imperfect labeling oracle is a tuple $\langle D, C_{sample}, O^I \rangle$, which has the same definition as *HTP*, except that the labeling oracle O^I is not perfect and may make mistakes when discovering the underlying state sequences $\{V_1, V_2, ..., V_m\}$ for sample sub-sequences $\{U_1, U_2, ..., U_m\}$. Obviously, *HTPIO* is a generalization of *HTP*.

Here we also define an intuitive metric for the solution S' to both problems:

Definition 3. Error rate of a solution S' for *HTPIO* ($Error(S', S)$).

$$Error(S', S) = \frac{Difference(S', S)}{L} \tag{2.1}$$

where the difference of two state sequences is computed as the number of corresponding elements that do not agree with each other, and S' and S are of the same length L.

The smaller the error rate, the better is the solution. However, in real problems it is hard to apply this metric, since the actual hidden sequence is unknown. This definition only serves as a performance measurement in section 4 about results on simulated datasets. Other possible performance measures for real experimental datasets are also discussed in section 4.

A similar problem to *HTPIO* has been studied by [1] in the context of Probabilistic Automata (PA). Our work differs from theirs in several aspects. First of all, we investigate the problem of sample sub-region selection whereas they do not. Second, we take errors in the labeling oracle into consideration. Third, we introduce a more intuitive measurement of error, compared to the *Kullback-Leibler divergence* of different PAs in their paper. Last but not least, we seek a time-efficient solution, whereas their work focuses on obtaining sample complexity bounds for learning the model while ignoring computational efficiency.

As described above, *HTPIO* asks for efficient solutions to two different kinds of sub-problems simultaneously: finding an effective sub-region sampling scheme and finding a good approximation of S. These two solutions form our HMM framework, which systematically incorporates validated knowledge into tiling array data analysis. In the following two sub-sections, we present efficient solutions to both sub-problems separately.

2.2.2 Selection of sample sub-regions

When deciding which sample sub-regions in D should be selected as inputs to the labeling oracle, we investigate a set of sample selection schemes besides random selection. To simplify discussion, we assume that C_{sample} is equal to null/empty and that we are selecting m non-overlapping sample sub-sequences $\{U_1, U_2, ..., U_m\}$, each of length k.

Some of these sampling schemes employ entropy as a measure. The first one of these, *MaxEntropy*, selects m non-overlapping sub-regions with the highest entropies. The second one, *UnbiasedEntropy*, divides all the sub-regions into m groups according to their entropy values, and randomly selects one sub-region out of each group. The third one, *MaxMinEntropy*, selects $m/2$ sub-regions with the highest entropies and $m/2$ sub-regions with the lowest entropies. *MaxEntropy* tends to pick up those sub-regions that contain both active and inactive probes in the same region (e.g. the transcribed gene regions in transcriptional tiling arrays), while the other two methods will pick up totally inactive sub-regions as well.

Another sampling scheme, *LeastKL*, employs a well-known measure in information theory called *"Kullback-Leibler divergence"* [43], between D of length L and its sub-sequence U_i of length k.

Definition 4 (Kullback-Leibler Divergence (*K-L divergence*))**.** Let D and U_i be probability distributions over a countable domain Z. The *Kullback-Leibler Divergence* of D with respect to U_i, $d_{KL}(D, U_i)$ is defined as follows:

$$d_{KL}(D, U_i) = \sum_{z \in Z} P_D(z) \log_2 \frac{P_D(z)}{P_{U_i}(z)} \tag{2.2}$$

By convention we let $0 \log 0 = 0$, and $0/0 = 1$.

Normally we think the smaller $d_{KL}(D, U_i)$, the more similar U_i is to D in terms of their probability distributions over Z. When selecting sample sub-sequences for *HTPIO* using *LeastKL*, we want to select m sub-sequences U_i with the smallest $d_{KL}(D, U_i)$ values. The underlying idea is to obtain information from those most representative regions for future learning algorithms.

For tiling array data, D is usually a sequence of uncountable real numbers, so the elements in D need to be discretized to integers (either by direct rounding, or rounding after log transformation, depending on the nature of the data), which requires $O(L)$ operations. When m, k are constants and $m, k << L$, an approximate result of the m non-overlapping sub-sequences can be obtained in $O(L)$ for all these schemes.

Empirical results in section 4 show that when the labeling oracle is perfect, the *MaxEntropy* and *LeastKL* sample selection algorithm are superior to other schemes; when the oracle makes relatively small mistakes, *MaxEntropy* always outperforms other schemes.

2.2.3 An efficient HMM approach for *HTPIO*

After the sample sub-sequences and their corresponding state sequences have been obtained, a frequency-based supervised learning algorithm is applied to build the HMM and then a Viterbi algorithm [65, 78] is utilized to compute the most likely state sequence S' for the whole sequence D, which is an approximate answer to *HTPIO*. The forward-backward algorithm [65] can also be used to generate detailed scores for each element in D, although it will be more time consuming than the Viterbi algorithm.

The supervised learning algorithm takes as input the sample sub-sequences $\{U_1, U_2, ..., Um\}$ and corresponding state sequences $\{V_1, V_2, ..., V_m\}$, each of length k, and outputs the following matrices:

$$A_{ij} = \frac{\sum_{V \in \{V_1, V_2, ..., V_m\}} \xi_V^S(i,j)}{\sum_{V \in \{V_1, V_2, ..., V_m\}} \gamma_V^S(i)} \tag{2.3}$$

$$B_{ik} = \frac{\sum_{(V,U) \in \{(V_1,U_1),(V_2,U_2),...,(V_m,U_m)\}} \xi_{V,U}^O(i,k)}{\sum_{V \in \{V_1, V_2, ..., V_m\}} \gamma_V^S(i)} \tag{2.4}$$

where $\xi_V^S(i,j)$ is the number of transitions from state i to j in state sequence V, $\gamma_V^S(i)$ is the number of occurrences of state i in V, $\xi_{V,U}^O(i,k)$ is the number of times state i in V emits k in U. We can then build a discrete HMM with A as the transition matrix, and B as the emission matrix. We set the initial state distribution of the HMM to uniform to avoid biased estimation for this parameter. As long as the initial state distribution is set to a reasonable distribution, it should not have a great impact on the final result when L is sufficiently large. When the sample size is relatively small, the discrete emission matrix B may be ill-formed if estimated directly, in which case we build a continuous HMM and use kernel density estimation [55] to construct smoother emission distributions for different states: if $x_1, x_2, ..., x_N$ are the observed emissions for a certain state, then its corresponding emission distribution is computed as $P(x) = \frac{1}{N}\sum_{i=1}^{N} W(x - x_i)$, where in this case W is a Gaussian function with mean 0 and predefined variance σ^2.

The supervised learning algorithm runs in $O(mk)$ time, and the Viterbi algorithm requires $O(n^2L)$ time, where n is the number of states (which is 2 or 4 in examples in section 3) in the HMM and L is the length of D. Since $mk < L$, the total time cost of our solution (sampling, learning, and decoding) to *HTPIO* is thus $O(n^2L)$, which is comparable to most of the existing tiling array analysis methods. Results in section 4 show that our methods handle large datasets effectively.

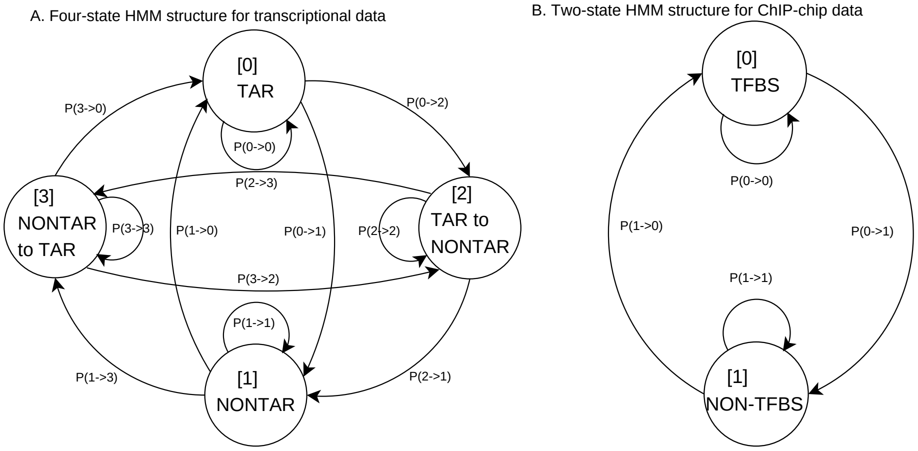
2.3 Implementations

In this section, we will show that even though at present there may exist too little experimentally validated data to be incorporated in our HMM approach described above, other kinds of validated knowledge such as gene annotation already provide a good basis for our methods in both transcriptional and ChIP-chip data analysis.

2.3.1 Incorporating gene annotation in transcriptional data analysis

In transcriptional tiling array experiments, TARs or transfrags form the subject of interest. Here the gene annotation of the organism under study is obviously the validated biological knowledge we should consider incorporating into our HMM approach.

Despite its inaccuracy, the knowledge of gene annotation usually involves a large



A. Four-state HMM structure for transcriptional data

B. Two-state HMM structure for ChIP-chip data

Figure 2.2. HMM structures for transcriptional and CHIP-chip data. (A) Four-state HMM structure for transcriptional data. The four states in the HMM are TAR and NONTAR states, and two corresponding transition states. The corresponding sequence starts from each state with equal probabilities. The reason for the existence of non-zero transition probabilities between states 2 and 3 is that neighboring probes in a tiling array may overlap with each other. (B) Two-state HMM structure for ChIP-chip data, which contains a TFBS state 0 and a non-TFBS state 1.

amount of information. This allows the construction of a four-state HMM instead of a two-state HMM. The structure of the HMM is illustrated in Figure 2.2A. Each probe in the tiling array can be in one of the four HMM states (TAR, NONTAR, and two other intermediate transition states), emitting the assigned intensity/score. As shown in Figure 2.1B, the parameters of the HMM can be estimated by learning from both positive and negative samples in the sequences of probes which fall into regions with known transcription characteristics, in this case, the knowledge of corresponding gene annotation.

What is more, the choice of annotated genes as the training set conforms to our Max-Entropy sample selection scheme, since these regions usually contain both high and low signals, thus having relatively high entropy values.

2.3.2 Incorporating gene annotation in ChIP-chip data analysis

For ChIP-chip data, we should first identify the possible knowledge to incorporate into our HMM approach, since this is not as obvious as for transcriptional data, where gene annotation is an intuitive choice. One option is the dataset of those experimentally verified regions, which at present is usually limited in size and cannot form a valid training set for HMM construction. On the other hand, the knowledge of gene annotations is somewhat related to the identification of binding sites, since transcription factor binding sites (TFBS) are usually considered to be enriched in upstream regions of genes, and unlikely to occur in inner regions of genes. By incorporating this knowledge, a two-state HMM can be constructed in the following way:

As shown in Figure 2.2B, the HMM contains a TFBS state 0 and a non-TFBS state 1. The overall emission distribution $h(t)$ is computed based on the ChIP-chip data. As shown in Figure 2.1B, the emission distribution of the non-TFBS state, $g(t)$, according to the above discussion, can be estimated based on the knowledge of inner regions in genes. The emission distribution of the TFBS state, $f(t)$, can then be obtained by subtracting $g(t)$ from $h(t)$, using canonical FDR procedures. The transition parameters of the HMM can be estimated based on empirical knowledge. Actually, if $f(t)$ and $g(t)$ are significantly different from each other, a small variance in transition parameters should not affect the result of the HMM approach very much.

However, the HMM constructed in this way may not be as effective as in the case of transcriptional data, since the knowledge involved in the construction does not relate to the TFBS very closely. Further scoring on the initial analysis results can be done by computing the posterior probabilities $P(S_i = k|D)$ for the predicted states on probes, where S_i is the state of the ith probe, k is the predicted state, and D is the emitted sequences of the probes involved. These scores indicate the confidence in every single prediction and can be used to refine the prediction results obtained by HMM analysis. The identified active probes can then be ranked according to the overall confidence levels in their regions and a threshold confidence level may either be set manually or be learned automatically to refine the original results.

2.3.3 Incorporating other validated knowledge in tiling array data analysis

Since our HMM framework defined in section 2 provides a general interface for incorporating validated knowledge about the dataset in question, virtually any such knowledge can be used by this approach. For example, our framework can take the data from a tiling array experiment, and select a medium-sized set of sub-regions by using some appropriate analysis method (e.g. the *MaxEntropy* sampling scheme in section 2.2). These sub-regions can be further studied by experimental validation, which identifies the underlying state (e.g. transcribed or not, in a transcriptional tiling array experiment) of every single probe inside these sub-regions. These outcomes form a well-established training set and can then be incorporated into our HMM approach in the framework, which will lead to more accurate analysis results than that obtained using only information from the array data. Since all these can be done systematically within our framework, it actually provides a way to consistently analyze tiling array data across a number of experiments and also across different types of experiments.

2.4 Results

2.4.1 Performance measurement

We use $Error(S', S)$ defined in section 2.1 as an intuitive measure to analyze the results on a simulated dataset, where we have access to the actual hidden state sequence S. We also investigate some key issues in our HMM approach, including sample selection, size of the training set, and error in the training data.

When we analyze the results on real experimental data, it is hard to get a good estimation of S, which makes it difficult to compute the overall error rate. One the other hand, for a rigorous performance evaluation like cross-validation, a gold-standard dataset with exact information is required. Unfortunately, in many cases no such dataset exists, especially over large genomic regions. In the absence of such a gold standard, we evaluate the performance of different methods by comparing their results against the imperfect training set used in the approach, and also against previous segmentation results of other non-HMM methods on the same dataset. Furthermore, we investigate how the size and noise of the training set affects the performance of our HMM approach.

2.4.2 Results on simulated dataset

A simulation on our framework of the solution to *HTPIO* proposed in section 2 was done to investigate its performance. We performed ~ 17000 trials, each of which solved a randomly generated *HTPIO* of $\langle D, C_{sample}, O^I \rangle$, where the length L of D is 1M, constraint C_{sample} specifies that $m = 2^i$ $(i = 1, 2, ..., 8)$ sub-regions, each of length $k = 50$, should be selected as samples, and O^I makes mistakes randomly with probability $e = 0, 0.05, 0.1$; $Error(S', S)$ was computed in each trial for different sample selection schemes described in section 2.2. The results in Figure 2.3 confirm that *MaxEntropy* and *K-L divergence* based sample selections are superior to other selection schemes when the labeling oracle O^I is perfect. When O^I makes mistakes with a relatively low probability, *MaxEntropy* outperforms all other sampling schemes. We also observe that as the sample size mk increases, the overall performances of all methods improve, and become stable when the sample size is larger than $\sim$ 13K. This observation leads to a hypothesis that an intelligently selected medium-sized training set is sufficient for our HMM approach on real experimental datasets, which is supported by the results in section 4.3 as well.

2.4.3 Results on transcriptional dataset

We tested our method on a transcriptional tiling array dataset which has 25mer oligonucleotide probes tiled approximately every 21bp covering all the non-repetitive DNA sequence

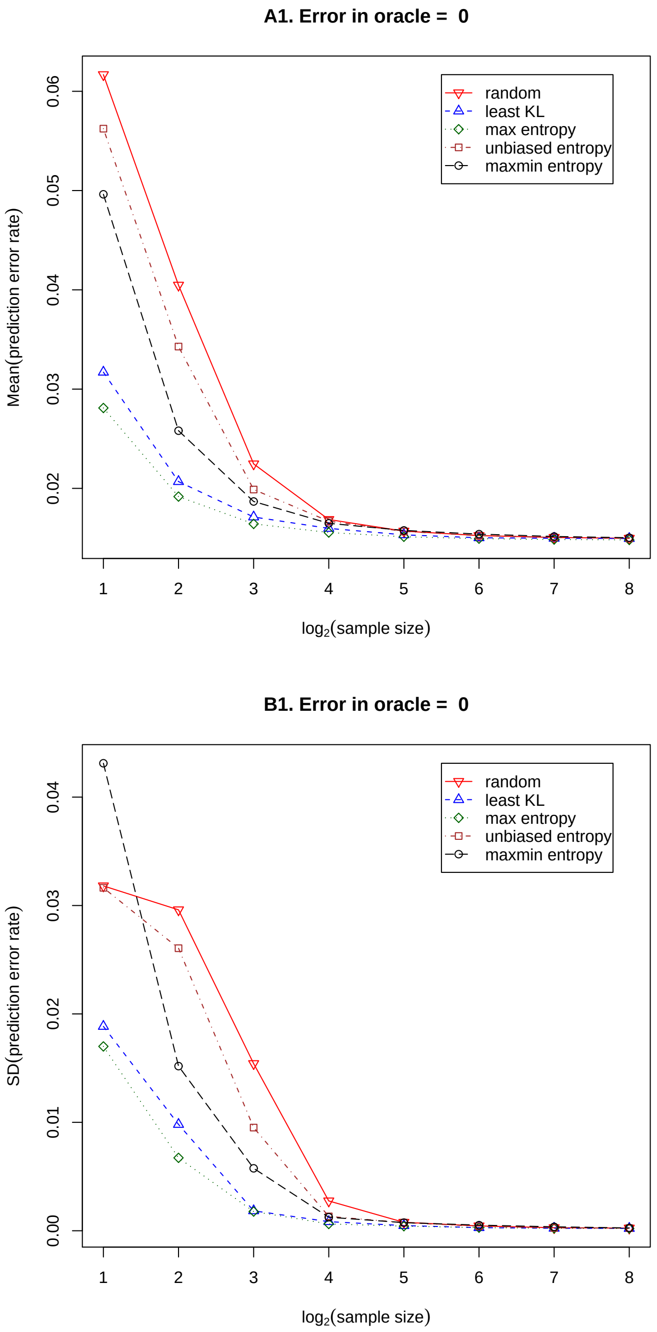


Figure 2.3. Results on simulated dataset. “Error in Oracle" is the probability with which O^I makes mistakes. (A) Mean of the prediction error rates. (B) Standard deviation of prediction error rates.

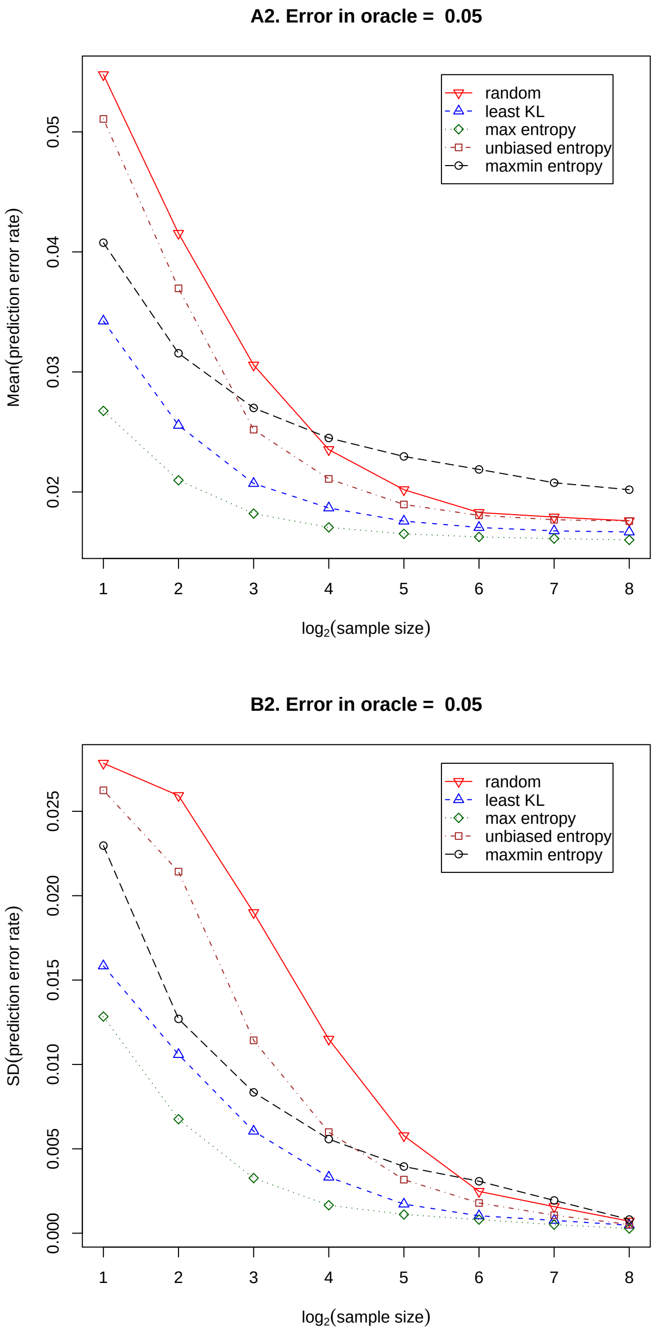
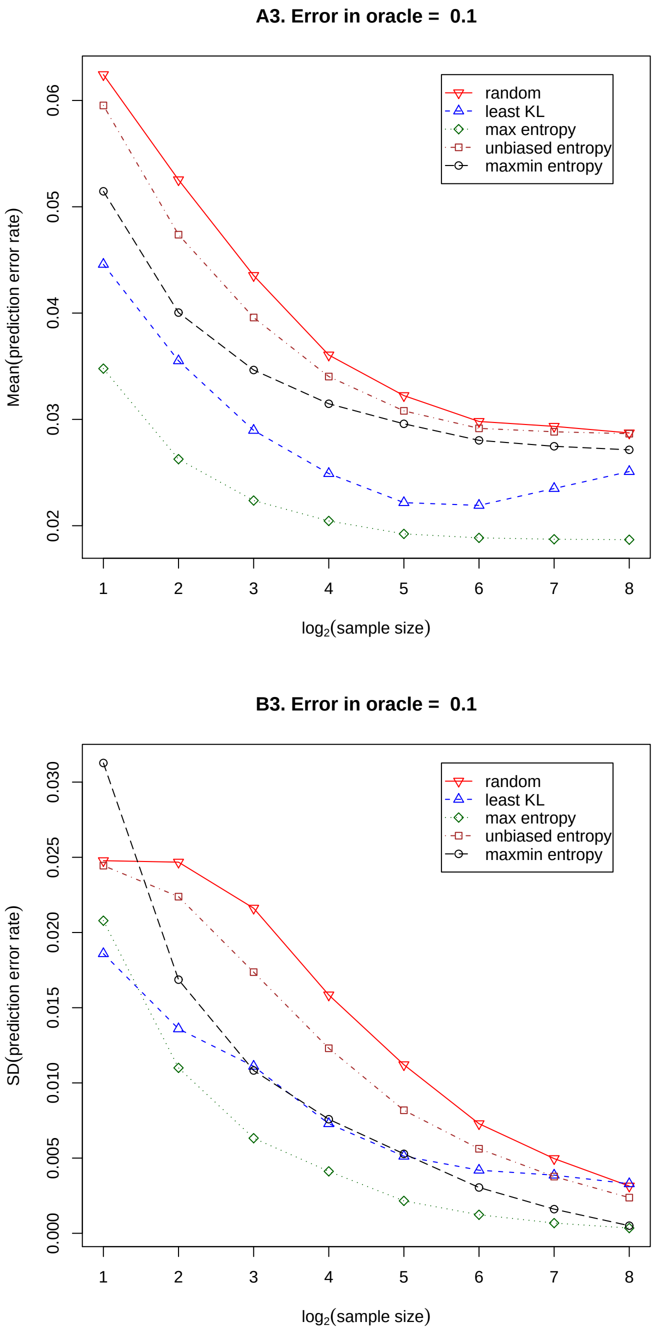


Figure 2.3 (continued)


A3. Error in oracle = 0.1

B3. Error in oracle = 0.1

Figure 2.3 (continued)

of the ENCODE regions ($\sim$ 30Mbases) [21]. This dataset is sufficiently large for our performance test, and the corresponding prediction result of a minrun/maxgap method [8] is available as well, which provides a good estimation of the TARs.

We formed the training set ($\sim$ 7.5Mb) from the normalized dataset by using the method in section 3.1 with the RefSeq annotation [63]. In order to investigate the performances of our methods with different-sized training sets, we also randomly selected a certain portion of the whole training set, and then built a basic discrete four-state HMM (Figure 2.2A) and a continuous HMM (by using kernel density estimation) based on that portion. The portions we selected were 1/2, 1/4, 1/8 and 1/16 of the whole training set, and every selection was repeated 16 times so that the variance of the corresponding performances could be estimated empirically. We also built a generalized HMM (GHMM) [54, 65] based on the whole training set to test the possible gain of using a more sophisticated model which captures length characteristics.

Figure 2.4 uses *Youden's J* [83], which is $Sensitivity + 1 - Specificity$, as a measure of the overall performances of different methods with different-sized training sets. The sensitivity and specificity of the HMM prediction results are computed based on both the whole training set and the previous prediction results of maxgap/minrun. Figure 2.4 shows that even when 1/4 ($\sim$ 1.9Mb) of the whole training set is used, our HMM approach gives a performance comparable to or better than existing methods, with either gene annotation or previous prediction results as performance criteria. Another important fact shown in Figure 2.4 is that the continuous HMM has much more stable performance than the discrete model, especially when the training set is small (less than 1/4 of the whole training set). This is because the continuous HMM has smoother emission distribution estimations than the discrete one, and its performance is thus less likely to be affected by a small set of biased samples. We can also observe that GHMM does not seem to give significantly better performance than simpler models.

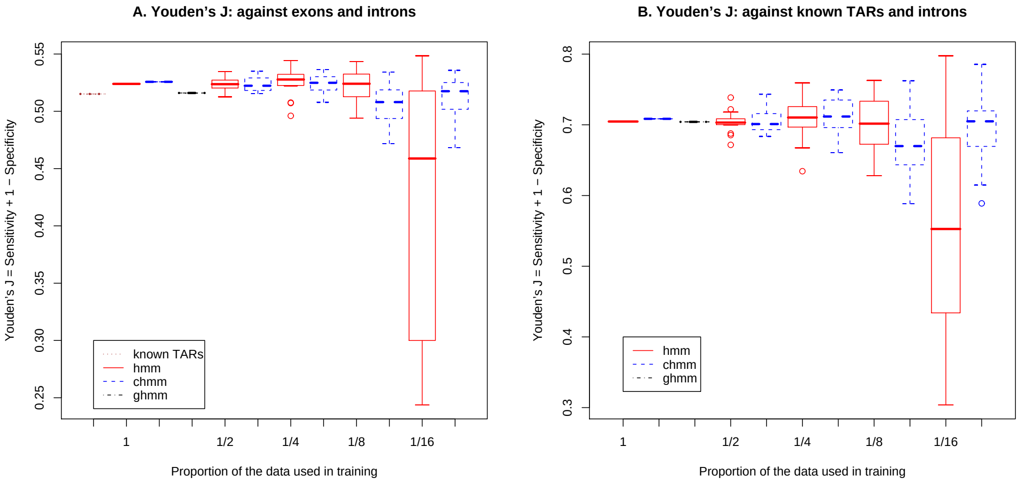


Figure 2.4. Results on transcriptional dataset: Youden's J. Different types of HMMs are built based on samples drawn from the whole training set with different proportions. Each trial with proportions less than 1 is repeated 16 times. Youden's J ($Sensitivity + 1 - Specificity$) is computed for the prediction result of each trial, and a boxplot of the calculated values for different models with different-sized training sets is generated. "known TARs" stands for previously predicted TARs by maxgap/minrun, "hmm" stands for discrete HMM, "chmm" stands for continuous HMM, and "ghmm" for generalized HMM. (A) RefSeq gene annotation is used as a criterion, where exon regions are used as positives, and intron regions are used as negatives. (B) Known TARs predicted by maxgap/minrun method are used as positives, and RefSeq intron regions are used as negatives.

We further computed the posterior probabilities for the predicted states on probes, and set different thresholds to identify TARs. Figure 2.5 shows the ROC curves of different models with different training sets. Again the continuous HMM outperforms the discrete one, and has good performance even with a relatively small ($\sim$ 1.9Mb) training set. The similarity of A and B diagrams in Figure 2.4 and Figure 2.5 also shows that gene annotation is a good criterion for performance measurement, if we do not have any existing prediction results to utilize.

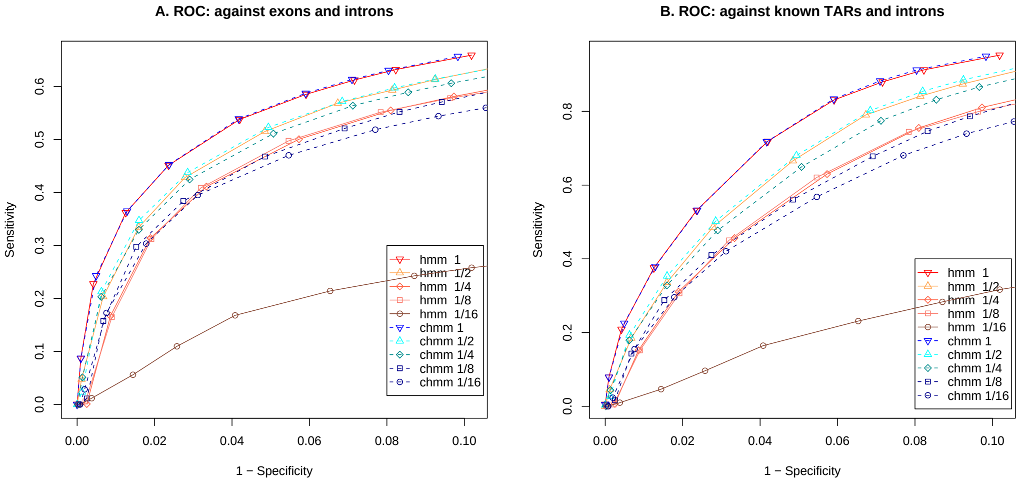


Figure 2.5. Results on transcriptional dataset: ROC curves. Based on the models described in section 3.1, we compute posterior probabilities for the predicted states on probes, and set different thresholds to identify TARs. ROC curves of the worst-case performance of these different models are then generated. "hmm 1/2" stands for the discrete HMM built with $1/2$ of the whole training set, and so on. (A) RefSeq gene annotation is used as a criterion, where exon regions are used as positives, and intron regions are used as negatives. (B) Known TARs predicted by the maxgap/minrun method are used as positives, and RefSeq intron regions are used as negatives.

The minimum training set guaranteeing good performance for our approach on this dataset is $\sim$ 1.9Mbases, which includes $\sim$ 0.1M probes. Since the size of the training set needed for satisfying performance of our method does not increase with the size of the dataset, it seems that if $\sim$ 0.1M probes in this type of tiling array experiment can be labeled and put into the training set, our method becomes immediately applicable to identify TARs for the whole dataset. We also want to point out that the labeling process does not have to be perfect: in this case, Figure 2.4A shows that less than 60% of the training set is actually correct, while Figure 2.4B shows that our method has satisfying performance with this training set.

2.4.4 Results on ChIP-chip dataset

We tested our method on a STAT1 ChIP-chip tiling array dataset which has 50mer oligonucleotide probes tiled approximately every 38bp covering most of the non-repetitive DNA sequence of the ENCODE regions ($\sim$ 30Mb). This dataset, as in the case of section 4.3, is sufficiently large for our performance test, and the corresponding prediction result of a maxgap/minrun method is available as well, which provides a good estimation of the TFBSs.

Due to the lack of available validated biological knowledge, we built a simple two-state continuous HMM (Figure 2.2B) based on the negative training set ($\sim$ 8Kb) from the normalized dataset by using the method described in section 3.1 with RefSeq annotation, computed the posterior probabilities for the probes being in NON-TFBS state, and set different thresholds to get different sets of TFBSs. Figure 2.6 shows the ROC curves of predictions by using our HMM approach and a p-value cutoff method. The inner gene regions are used as negatives, while both previously predicted TFBSs and the promoter regions in the array are used as positives. We can observe that the HMM approach has better performance than the p-value cutoff approach in both criteria.

The near-linear ROC curves in Figure 2.6B also show that the promoter regions may not be as good a criterion as the previous TFBS results. Analogous to the case with transcriptional data, when experimental validation results become sufficient to form a medium-sized

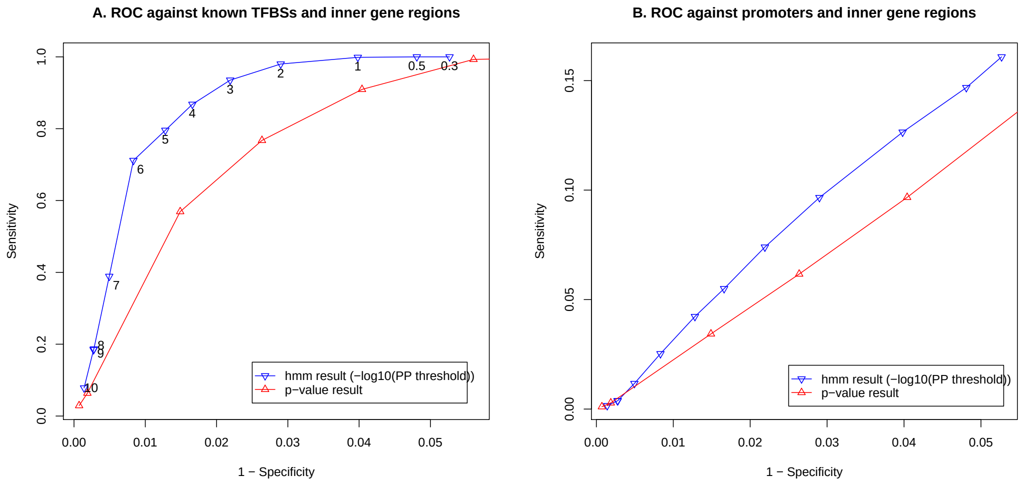


Figure 2.6. Results on ChIP-chip dataset: ROC curves. A continuous HMM is built based on the normalized ChIP-chip dataset by using the method described in section 3.2. ROC curves of the performance of the HMM approach and another p-value cutoff method are then generated. (A) Previously predicted TFBSs are used as positives, and the inner gene regions are used as negatives. The numbers along the ROC curve of HMM result are the $-\log_{10}(PP\,threshold)$, where PP is the posterior probability of a probe being in NON-TFBS state. (B) The promoter regions in the array are used as positives, and the inner gene regions are used as negatives.

(covering $\sim$ 0.1M probes) knowledgebase about the dataset in question, this knowledgebase can be utilized as a performance measure as well as the training set for our HMM approach.

2.5 Discussions

We present an efficient HMM framework which systematically incorporates *validated biological knowledge* (e.g. known gene annotation, experimental validation results) into tiling array data analysis. This framework, which consists of a *MaxEntropy* sample selection algorithm and HMM learning and decoding approaches, is proposed based on *HTPIO*, an idealized definition of the tiling array analysis problem. Empirical results of our methods in the framework on a simulated dataset, a transcriptional dataset and a ChIP-chip dataset show that our framework effectively handles large datasets, even with a relatively noisy training set.

Our work differs from previous studies in tiling array data analysis by specifically taking *validated biological knowledge* into consideration and systematically incorporating it using an empirically tested *MaxEntropy* sample selection scheme for optimal analysis. These features ensure the good performance of our framework with even a relatively small gold standard training set, which has not been specifically considered by previous methods. In this way our framework can consistently analyze tiling-array data across a number of experiments, and can process different types of array data automatically, without the need to manually set additional parameters. This feature will become an advantage for analyzing very large datasets (e.g. for the $\sim$ 3Gb human genome): when sufficient experimental validations are done afterwards, a *medium-sized* (covering $\sim$ 0.1M probes, according to the empirical results in section 4.3) *validated biological knowledgebase* can be formed for the array data in question. Our framework can then improve its performance with the guidance of this medium-sized knowledgebase, and its refined analysis results can in turn assist further experimental studies. What is more, in section 4.3 our framework gives good performance by incorporating some relatively inaccurate biological knowledge (with approximately 60% correctness), and the sub-regions in the training set are not specifically chosen according to our proposed sampling scheme. We can expect that for real problems which use validated biological knowledge from highly accurate experimental validations, the necessary minimum size of the biological knowledgebase could be even smaller than $\sim$ 0.1M probes for our framework to achieve satisfying performance.

Another feature of our method is that given a set of regions with similar signal intensities, it can identify all the regions in the whole dataset with similar signal distributions. This feature is potentially useful for identifying regions with different transcription levels. For instance, our HMM method can take as the training set all the known highly expressed genes in the tissue, and then identify all the regions in the corresponding transcriptional tiling array that have the similar transcription level.

Part II

Integrated Analysis of Nondeterministic Sampling Techniques

Chapter 3

Integrated Distribution Estimation based on Nondeterministic Partial Samples

3.1 Introduction

The concept of the "gene" has evolved and become more complex [23]: the discovery of splicing [7, 14, 22] revealed that the gene was a series of exons, coding for, in some cases, discrete protein domains, and separated by long noncoding stretches called introns. With alternative splicing, one genetic locus could code for multiple different mRNA transcripts (isoform transcripts). This discovery complicated the concept of the gene radically. As of 2007, the GENCODE annotation [26] contained on average 5.4 transcripts per locus.

With the recent development of high-throughput RNA sequencing (RNA-Seq) technology, it possible for biologists to measure transcription at an unprecedented precision. The problem of isoform quantification tries to reconstruct the abundances of similar isoforms based on a set of RNA-seq reads. Various methods have been developed to solve this problem. In previous work, researchers proposed different statistical frameworks [82, 35] for using maximum likelihood estimation to solve the problem, others [44] studied the conditions under which the problem can be solved, revealing that although neither single nor paired-end sequencing guarantee a unique solution, paired-end reads may be sufficient to solve the vast majority of the transcript variants in practice.

The isoform quantification problem represents a special class of sampling process: partial samples are drawn from a pool of similar objects with multiple nondeterministic sampling techniques, and the only control an experimenter has is on the total cost of these samples and how to assign the cost to the different sampling methods. As we will discuss in detail in the following sections, each partial sample can be compatible with multiple objects, and a traditional "counting" solution is no longer able to recover the relative abundances of the objects in the pool (i.e. the distribution). Here we present a generalized statistical solution, which differs from previous ones in the following aspects:

1. With a generalized G function, we provide a flexible way to incorporate characteristics of different sequencing technologies.
2. This framework integrates the analysis of different sample sets generated from different sampling technologies.
3. We developed a fast algorithm for estimating the expected performance of our expectation maximization based solution.
4. Given the estimated isoform abundances, we also propose to use an information the-

oretical method to measure the transcriptome complexity.

In this chapter, we will first introduce a mathematical definition of the generalized partial sampling and distribution estimation problem, provide a expectation maximization based iterative solution, discuss in detail on how to estimate the performance of this solution using Fisher information based heuristics, and present fast algorithms that implement compute these heuristics. We will also show results of applying our methods to both simulated and real-world data, illustrating scenarios where such integrated analysis can be the most informative.

3.2 Problem Definition

We start by defining the generalized process of batch partial sampling, and the relationships between partial samples and the objects being sampled.

Definition 5 (Batch Partial Sampling)**.** Let $I = \{I_1, ..., I_K\}$ be all the possible isoforms, with relative abundances $\Theta = (\theta_1, ..., \theta_K)^T$, where $\sum_{k=1}^{K} \theta_k = 1$. We assume that there are M different partial sampling methods: $Samp_1, ..., Samp_M$, and let S denote all the samples: $S = \{s \text{ from } Samp_m | m = 1, ..., M\}$. We also define $\delta_{s,k} = Ind(\text{partial sample } s \text{ is compatible with } I_k)$, where Ind is the indicator function. There are in total $N = \sum_{m=1}^{M} N_m$ samples, where N_m is the total number of partial samples from $Samp_m$.

Here we assume a two-step sampling process: First, a sampling method $Samp_m$ chooses an isoform instance I_k according to Θ. Second, the sampling method generates a partial sample s according to a local partial sample generation model $G_{s,k}^{(m)} = Pr(\text{generating } s | I_k, Samp_m)$.

Definition 6 (Distribution Estimation based on Batch Partial Samples)**.** Given I, and S as defined in Definition 5, estimate Θ.

As shown in Figure 3.1, I are the isoforms with different relative abundances Θ, and S are the single- and paired-end reads whose sequences align with part of this gene region. Some of these reads (e.g. read2, 3 and 5) are compatible with multiple isoforms. The ultimate problem is to estimate Θ based on I and S, i.e., reconstructing a distribution based on partial observations.

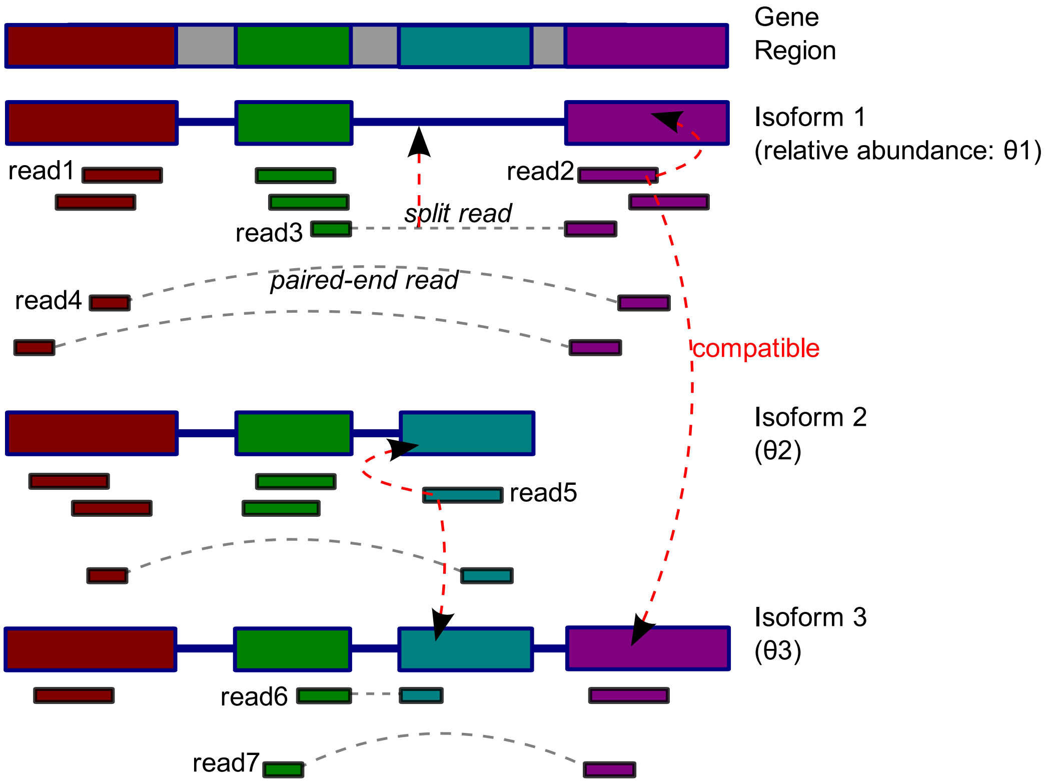


Figure 3.1. Partial samples in the isoform quantification problem.

In the remaining part of this chapter, we will use two notations to describe a partial sample s: $s_{m,i}$ is the ith sample from $Samp_m$; and $s^{(k)}_{[a,b)}$ stands for a partial sample from I_k, starting (inclusive) from position a and ending (exclusive) at b in that isoform. We also define exons as those nodes in the splicing graph of a gene, so that there are no exons that overlap with each other.

3.3 Maximum Likelihood Estimation (MLE)

Definition 6 does not give an explicit criterion for a "good" estimation of Θ. Since the problem is defined in a statistical sampling framework, it is natural to consider using Maximum Likelihood as such a criterion.

Definition 7 (Maximum-Likelihood Distribution Estimation based on Batch Partial Samples)**.** Given I, and S as defined in Definition 5, find $\hat{\Theta}$ such that:

$$\hat{\Theta} = argmax_{\Theta}\log(Pr(S|\Theta)) \tag{3.1}$$

By plugging in the partial samples $s_{m,i}$s and $G^{(m)}_{s,k}$s, we can rewrite the formula above as follows:

$$\begin{aligned}
\hat{\Theta} &= argmax_{\Theta}\log(Pr(S|\Theta)) && (3.2)\\
&= argmax_{\Theta}\log\prod_{m=1}^{M}\prod_{i=1} N_m Pr(s_{m,i}|\Theta, Samp_m) && (3.3)\\
&= argmax_{\Theta}\log\prod_{m=1}^{M}\prod_{s=s_{m,*}} Pr(s|\Theta, Samp_m) && (3.4)\\
&= argmax_{\Theta}\log\prod_{m=1}^{M}\prod_{s=s_{m,*}}\sum_{k=1}^{K}\delta_{s,k}\theta_k Pr(s|\Theta, Samp_m, I_k) && (3.5)\\
&= argmax_{\Theta}\log\prod_{m=1}^{M}\prod_{s=s_{m,*}}\sum_{k=1}^{K}\delta_{s,k}\theta_k G_{s,k}^{(m)} && (3.6)\\
&= argmax_{\Theta}\sum_{m=1}^{M}\sum_{s=s_{m,*}}\log\sum_{k=1}^{K}\delta_{s,k}\theta_k G_{s,k}^{(m)} && (3.7)
\end{aligned}$$

In the next section, we demonstrate how this problem can be solved by introducing a hidden variable $Z_{s,k}$ and using the technique of Expectation Maximization [15].

3.4 Applying the Expectation Maximization Method

We define $Z_{s,k} = Ind(s \text{ is from } I_k)$, which are the hidden variables in this problem. Since Expectation Maximization gives an iterative solution, we denote the estimation for Θ in the nth step as $\Theta^{(n)}$, and further define $\zeta_{s,k}^{(n)} = \mathbf{E}_{Z|S,\Theta^{(n)}}[Z_{s,k}]$.

$$
\begin{aligned}
\zeta_{s,k}^{(n)} &= \mathbf{E}_{Z|S,\Theta^{(n)}}\left[Z_{s,k}\right] && (3.8)\\
&= \mathbf{E}\left[Z_{s,k}|S,\Theta^{(n)}\right] && (3.9)\\
&= Pr\left(Z_{s,k}=1|s,\Theta^{(n)}\right) && (3.10)\\
&= \frac{Pr\left(Z_{s,k}=1,s|\Theta^{(n)}\right)}{Pr\left(s|\Theta^{(n)}\right)} && (3.11)\\
&= \frac{\delta_{s,k}\theta_k^{(n)}G_{s,k}^{(m)}}{\sum_{k'=1}^{K}\delta_{s,k'}\theta_{k'}^{(n)}G_{s,k'}} && (3.12)
\end{aligned}
$$

3.4.1 E step

$$
\begin{aligned}
Q^{(n)}(\Theta) &= \mathbf{E}_{Z|S,\Theta^{(n)}}\left[\log(Pr(Z,S|\Theta))\right] && (3.13)\\
&= \mathbf{E}_{Z|S,\Theta^{(n)}}\left[\sum_{m=1}^{M}\sum_{s=s_{m,*}}\log\sum_{k=1}^{K}Z_{s,k}\theta_k G_{s,k}^{(m)}\right] && (3.14)\\
&= \mathbf{E}_{Z|S,\Theta^{(n)}}\left[\sum_{m=1}^{M}\sum_{s=s_{m,*}}\sum_{k=1}^{K}Z_{s,k}\log\theta_k G_{s,k}^{(m)}\right] && (3.15)\\
&\quad \text{(for all } Z_{s,*}\text{, one and only one can have a value of 1)} && (3.16)\\
&= \sum_{m=1}^{M}\sum_{s=s_{m,*}}\sum_{k=1}^{K}\mathbf{E}_{Z|S,\Theta^{(n)}}\left(Z_{s,k}\right)\log\theta_k G_{s,k}^{(m)} && (3.17)\\
&= \sum_{m=1}^{M}\sum_{s=s_{m,*}}\sum_{k=1}^{K}\zeta_{s,k}^{(n)}(\log\theta_k+\log G_{s,k}^{(m)}) && (3.18)\\
&= \sum_{m=1}^{M}\sum_{s=s_{m,*}}\sum_{k=1}^{K}\zeta_{s,k}^{(n)}\log\theta_k + C && (3.19)
\end{aligned}
$$

3.4.2 M step

We want to maximize

$$Q^{(n)}(\Theta) \tag{3.20}$$

with constraint:

$$\sum_{k=1}^{K} \theta_k = 1 \tag{3.21}$$

We introduce a Lagrange multiplier λ and rewrite the problem as maximizing:

$$T^{(n)}(\Theta, \lambda) = Q^{(n)}(\Theta) + \lambda \left(\sum_{k=1}^{K} \theta_k - 1 \right) \tag{3.22}$$

By computing the partial derivatives, we have:

$$\begin{aligned} \frac{\partial T^{(n)}(\Theta, \lambda)}{\partial \theta_k} &= 0 && (3.23) \\ \sum_{m=1}^{M} \sum_{i=1}^{N_m} \frac{\zeta_{s,k}^{(n)}}{\theta_k} + \lambda &= 0 && (3.24) \\ \theta_k &= -\frac{\sum_{m=1}^{M} \sum_{i=1}^{N_m} \zeta_{s,k}^{(n)}}{\lambda} && (3.25) \end{aligned}$$

Inserting the result above into the constraint, we have:

$$
\begin{aligned}
-\sum_{k=1}^{K}\sum_{m=1}^{M}\sum_{s=s_{m,*}} \zeta_{s,k}^{(n)}\frac{1}{\lambda} &= 1 && (3.26)\\
-\sum_{m=1}^{M}\sum_{s=s_{m,*}} \frac{\sum_{k=1}^{K}\delta_{s,k}\theta_k^{(n)}G_{s,k}^{(m)}}{\sum_{k=1}^{K}\delta_{s,k}\theta_k^{(n)}G_{s,k}^{(m)}}\frac{1}{\lambda} &= 1 && (3.27)\\
\lambda &= -\sum_{m=1}^{M}\sum_{s=s_{m,*}} 1 && (3.28)\\
\lambda &= -\sum_{m=1}^{M} N_m && (3.29)\\
&= -N && (3.30)
\end{aligned}
$$

Inserting the calculated value of λ back into the estimation for θ_k's, we have:

$$
\begin{aligned}
\theta_k^{(n+1)} &= \frac{\sum_{m=1}^{M}\sum_{s=s_{m,*}}\zeta_{s,k}^{(n)}}{N} && (3.31)\\
&= \frac{\sum_{m=1}^{M}\sum_{s=s_{m,*}}\frac{\delta_{s,k}\theta_k^{(n)}G_{s,k}^{(m)}}{\sum_{k'=1}^{K}\delta_{s,k'}\theta_{k'}^{(n)}G_{s,k'}}}{N} && (3.32)
\end{aligned}
$$

as the new estimation for Θ.

The iterative estimation in Equation 3.32 is intuitively consistent with the case of estimating a distribution based on full samples: consider the scenario in which for each s, there is only one $k \in 1, ..., K$ satisfying $\delta_{s,k} > 0$, the right hand side of Equation 3.32 thus becomes $\frac{\sum_{m=1}^{M}\sum_{s=s_{m,*}}\delta_{s,k}}{N}$, which is exactly how the distribution estimation problem with traditional full samples can be solved. In the case of partial samples, our solution provides a way to adjust the "weight" each sample s contributes to the θ_ks of different objects.

3.5 Application of the Maximum Likelihood Estimation Solution

Figure 3.2 presents an example of applying our method above to a human RNA-Seq dataset. In this example, 4 different sampling methods have been used in the experiment, each bearing its unique sampling characteristics (e.g. read length, uniqueness of mapping) and generating a set of partial samples. The $\hat{\Theta}$ estimates we get from applying Equation 3.32 are also shown in the figure.

As mentioned in the previous sections, the $\hat{\Theta}$ estimation is optimal only in the sense of Maximum Likelihood, and one important question that needs to be addressed is to have an estimation on how close this estimation is to the true value of Θ. For example, biologists would want to know how much error should be expected when looking at the isoform abundances inferred by various estimation algorithms. Some previous studies [35] use simulations (by using the $\hat{\Theta}$ as the true Θ and performing many trials of the random sampling and Θ estimation progress) to provide an answer, which may become time-consuming for large scale problems (consider repeatedly running such simulation on the whole genome for many experiments). In the following sections, we will focus on developing a heuristic for estimating the error in $\hat{\Theta}$ and also efficient algorithms to compute it.

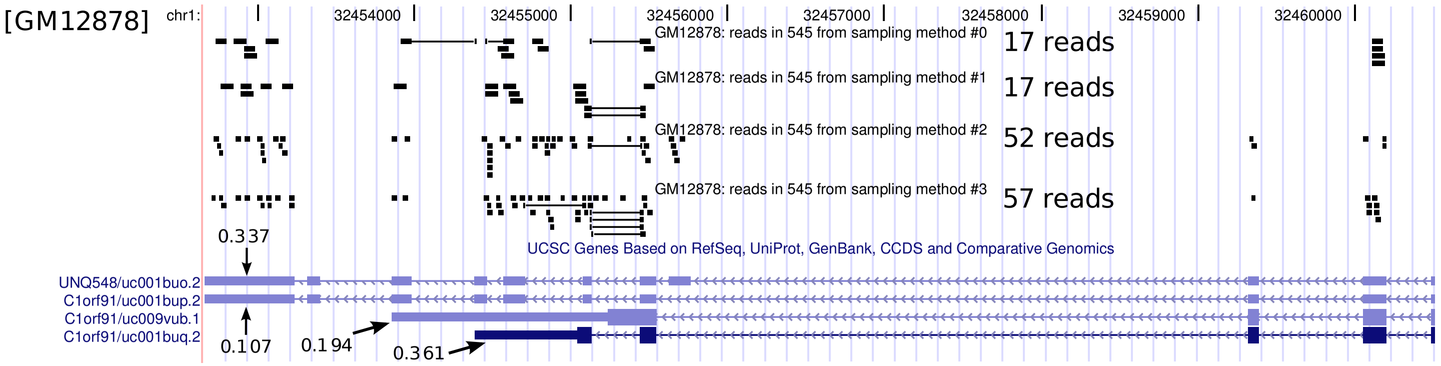


Figure 3.2. Application to human RNA-seq Data.

The numbers beside the isoforms are the estimated Θ based on the four sets of reads in this gene region.

3.6 Analyzing the Performance of Estimation

Given $\hat{\Theta}$ obtained from the MLE solution presented in the previous section, we would like to understand how much this estimate will deviate from the "true" Θ on average. Here we focus on the variance of the $\hat{\Theta}$, which describes how stable the MLE result is over many different partial sample sets drawn from the same isoform set:

$$Average\left(var(\hat{\theta}_k)\right) = \frac{\sum_{k=1}^{K-1} var(\hat{\theta}_k)}{K-1} \tag{3.33}$$

As we will show later, although brute-force simulation can be performed to obtain a relatively accurate estimation of this measurement, it is may become computationally intractable when there are too many reads and genes to be considered. We thus propose to use a Fisher information based heuristic for estimating $Average\left(var(\hat{\theta}_k)\right)$, and present a fast algorithm to compute the exact value of this heuristic.

We first introduce the Fisher information matrix [72, 76] as a basis for further discussion. The Fisher information is a way of measuring the amount of information that the random samples S carries about the unknown parameter Θ upon which the likelihood function of Θ, $Pr(S|\Theta)$, depends. An important use of the Fisher information matrix in statistical analyses is its contribution to the calculation of the covariance matrices of estimates of parameters fitted by maximium likelihood.

Let $\theta_1, ..., \theta_{K-1}$ be the free parameters, and $\theta_K = 1 - \sum_{k=1}^{K-1} \theta_k$.

Definition 8 (Observed Fisher information matrix)**.**

$$\begin{aligned}
\mathfrak{I}(\Theta)_{p,q} &= -\frac{\partial^2 \log(Pr(S|\Theta))}{\partial\theta_p\partial\theta_q}, \text{ where } p,q=1,...,K-1 && (3.34)\\
&= \sum_{m=1}^{M}\sum_{s=s_{m,*}} -\frac{\partial^2 \log\sum_{k=1}^{K}\delta_{s,k}\theta_k G_{s,k}^{(m)}}{\partial\theta_p\partial\theta_q} && (3.35)\\
&= \sum_{m=1}^{M}\sum_{s=s_{m,*}} -\frac{\partial^2\left[\log\left(\sum_{k=1}^{K-1}\delta_{s,k}\theta_k G_{s,k}^{(m)} + \delta_{s,K}G_{s,K}^{(m)}(1-\sum_{l=1}^{K-1}\theta_l)\right)\right]}{\partial\theta_p\partial\theta_q} && (3.36)\\
&= \sum_{m=1}^{M}\sum_{s=s_{m,*}} -\frac{\partial\left[\frac{1}{\sum_{k=1}^{K}\delta_{s,k}\theta_k G_{s,k}^{(m)}}\left(\delta_{s,p}G_{s,p}^{(m)} - \delta_{s,K}G_{s,K}^{(m)}\right)\right]}{\partial\theta_q} && (3.37)\\
&= \sum_{m=1}^{M}\sum_{s=s_{m,*}} \frac{\left(\delta_{s,p}G_{s,p}^{(m)} - \delta_{s,K}G_{s,K}^{(m)}\right)\left(\delta_{s,q}G_{s,q}^{(m)} - \delta_{s,K}G_{s,K}^{(m)}\right)}{\left[\sum_{k=1}^{K}\delta_{s,k}\theta_k G_{s,k}^{(m)}\right]^2} && (3.38)
\end{aligned}$$

Definition 9 (Expected Fisher information matrix)**.**

$$\mathcal{I}(\Theta)_{p,q} = \mathbf{E}\left[\mathfrak{I}(\Theta)_{p,q}\right] \qquad (3.39)$$

3.6.1 Covariance matrix of the maximum likelihood estimator

Let $T(S) = (\hat{\theta}_1, ..., \hat{\theta}_{K-1}, 1-\sum_{k=1}^{K-1}\hat{\theta}_k)^T$, and $\psi(\Theta) = \mathbf{E}\left[T(S)\right]$. The Cramér-Rao bound [76] states that:

$$cov_\Theta\left(T(S)\right) \geq \frac{\partial\psi(\Theta)}{\partial\Theta}\left[\mathcal{I}(\Theta)\right]^{-1}\left(\frac{\partial\psi(\Theta)}{\partial\Theta}\right)^T \qquad (3.40)$$

, where $\left[\partial\psi(\Theta)/\partial\Theta\right]_{u,v} = \partial\psi_u(\Theta)/\partial\theta_v$, $u=1,...,K$; $v=1,...,K-1$.

We then estimate $\psi(\Theta)$ by Θ, and use the bound above to estimate the covariance matrix:

$$\begin{aligned}\frac{\partial \psi_u(\Theta)}{\partial \theta_v} &\approx \frac{\partial \theta_u}{\partial \theta_v} \qquad (3.41)\\ &= \begin{cases} 1 & \text{if } u = v \text{ and } u, v < K; \\ -1 & \text{if } u = K; \\ 0 & \text{otherwise} \end{cases} \qquad (3.42)\end{aligned}$$

$$cov_{\Theta}\left(T(S)\right) \approx \frac{\partial\psi(\Theta)}{\partial\Theta}\left[\mathcal{I}(\Theta)\right]^{-1}\left(\frac{\partial\psi(\Theta)}{\partial\Theta}\right)^{T} \tag{3.43}$$

$$= \begin{bmatrix} & \mathbf{I}_{(K-1)\times(K-1)} & \\ -1 & \cdots & -1 \end{bmatrix}_{K\times(K-1)} \times \left(\left[\mathcal{I}(\Theta)\right]^{-1}\right)_{(K-1)\times(K-1)} \tag{3.44}$$

$$\times \begin{bmatrix} & -1 \\ \mathbf{I}_{(K-1)\times(K-1)} & \vdots \\ & -1 \end{bmatrix}_{(K-1)\times K} \text{, } \mathbf{I} \text{ is the identity matrix} \tag{3.45}$$

$$= \begin{bmatrix} & \mathcal{I}^{-1}_{(K-1)\times(K-1)} & \\ -\sum_{k=1}^{K-1}\mathcal{I}^{-1}_{k,1} & \cdots & -\sum_{k=1}^{K-1}\mathcal{I}^{-1}_{k,K-1} \end{bmatrix}_{K\times(K-1)} \tag{3.46}$$

$$\times \begin{bmatrix} & -1 \\ \mathbf{I}_{(K-1)\times(K-1)} & \vdots \\ & -1 \end{bmatrix}_{(K-1)\times K} \tag{3.47}$$

$$= \begin{bmatrix} & & & -\sum_{k=1}^{K-1}\mathcal{I}^{-1}_{1,k} \\ & \mathcal{I}^{-1}_{(K-1)\times(K-1)} & & \vdots \\ & & & -\sum_{k=1}^{K-1}\mathcal{I}^{-1}_{K-1,k} \\ -\sum_{k=1}^{K-1}\mathcal{I}^{-1}_{k,1} & \cdots & -\sum_{k=1}^{K-1}\mathcal{I}^{-1}_{k,K-1} & \sum_{i=1}^{K-1}\sum_{j=1}^{K-1}\mathcal{I}^{-1}_{i,j} \end{bmatrix}_{K\times K} \tag{3.48}$$

3.6.2 Heuristic for MLE performance estimation

In order to provide a single value measure for the expected performance of Maximum Likelihood estimation, we propose to use the following heuristic to estimate the average variance of $\hat{\Theta}$:

$$Average\left(var(\hat{\theta}_k)\right) \approx \frac{\sum_{k=1}^{K-1} \frac{1}{\mathcal{I}(\Theta)_{k,k}}}{K-1} \tag{3.49}$$

This heuristic avoids the potential computational intensive and numerically unstable computation of the inverse of $\mathcal{I}$, and is consistent with the theoretical result on the lower-bound of $var(\hat{\theta})$ in one dimensional case:

$$var(\hat{\theta}) \geq \frac{1}{\mathcal{I}(\theta)} \tag{3.50}$$

which is a specialization of the result in the previous subsection. In other words, the precision to which we can estimate Θ is fundamentally limited by the Fisher information.

In order to compute this heuristic, all we need is $\mathcal{I}(\Theta)$ itself. However, the brute-force computation (according to Definition 8 and 9) of this matrix will be time-consuming since its time complexity is proportional to the total number of possible sample sets (which in turn grows exponentially with the number of samples). In the next section, we will present algorithms that can compute this matrix in a more efficient fashion.

3.7 Efficient Computation of $\mathcal{I}(\Theta)$

First of all, we can decompose $\mathcal{I}(\Theta)$ in the following way:

$$\begin{aligned}
\mathcal{I}(\Theta)_{p,q} &= \mathbf{E}\left[\mathfrak{I}(\Theta)_{p,q}\right] && (3.51)\\
&= \sum_{m=1}^{M}\sum_{s=s_{m,*}} \mathbf{E}\left[-\frac{\partial^2 \log \sum_{k=1}^{K} \delta_{s,k}\theta_k G_{s,k}^{(m)}}{\partial\theta_p \partial\theta_q}\right] && (3.52)\\
&= \sum_{m=1}^{M} N_m \mathcal{I}^{(m)}(\Theta)_{p,q} && (3.53)
\end{aligned}$$

where

$$\mathcal{I}^{(m)}(\Theta)_{p,q} = \mathbf{E}_{s\sim Samp_m}\left[-\frac{\partial^2 \log \sum_{k=1}^{K} \delta_{s,k}\theta_k G_{s,k}^{(m)}}{\partial\theta_p \partial\theta_q}\right] \qquad (3.54)$$

is the expected Fisher information matrix of a single partial sample based on $Samp_m$. Thus we need to be able to compute $\mathcal{I}^{(m)}(\Theta)$ in order to obtain $\mathcal{I}(\Theta)$.

3.7.1 Further decomposing $\mathcal{I}^{(m)}(\Theta)$

$$\begin{aligned}
\mathcal{I}^{(m)}(\Theta)_{p,q} &= \sum_{k=1}^{K}\theta_k \left\{ \sum_{s=s_{[a,b)}^{(k)};\forall[a,b)\in I_k} -G_{s,k}^{(m)} \frac{\partial^2 \log \sum_{k'=1}^{K} \delta_{s,k'}\theta_{k'} G_{s,k'}}{\partial\theta_p \partial\theta_q} \right\} && (3.55)\\
&= \sum_{k=1}^{K}\theta_k \sum_{s=s_{[a,b)}^{(k)};\forall[a,b)\in I_k} G_{s,k}^{(m)} \mathfrak{I}^{(m)}_{s=s_{[a,b)}^{(k)}}(\Theta)_{p,q} && (3.56)
\end{aligned}$$

where

$$\mathfrak{I}^{(m)}_{s=s^{(k)}_{[a,b)}}(\Theta)_{p,q} = -\frac{\partial^2 \log \sum_{k'=1}^{K} \delta_{s,k'} \theta_{k'} G^{(m)}_{s,k'}}{\partial\theta_p \partial\theta_q} \tag{3.57}$$

$$= \frac{\left(\delta_{s,p} G^{(m)}_{s,p} - \delta_{s,K} G^{(m)}_{s,K}\right)\left(\delta_{s,q} G^{(m)}_{s,q} - \delta_{s,K} G^{(m)}_{s,K}\right)}{\left[\sum_{k'=1}^{K} \delta_{s,k'} \theta_{k'} G^{(m)}_{s,k'}\right]^2} \tag{3.58}$$

is the Fisher information matrix of a partial sample s from $Samp_m$ at $[a,b)$ in I_k.

A brute-force algorithm for computing $\mathfrak{I}^{(m)}_{s=s^{(k)}_{[a,b)}}(\Theta)$ can thus be described as follows:

Algorithm 1 BruteForceFIM$(I, \Theta, Samp_m, p, q)$

```
GIVEN: Possible isoforms $I = \{I_1, I_2, ..., I_K\}$;
Relative abundances $\Theta = (\theta_1, \theta_2, ..., \theta_K)$;
Sampling method $Samp_m$ Integer $p, q \in \{1, 2, ..., K-1\}$.
COMPUTE: The value of $\mathcal{I}^{(m)}(\Theta)_{p,q}$.
$\mathcal{I} \leftarrow 0$
for all $I_k \in I$ do
    $\mathcal{I}_k \leftarrow 0$
    for all $[a,b) \in I_k$ do
        $s \leftarrow s^k_{[a,b)}$
        $\mathcal{I}_k \leftarrow \mathcal{I}_k + G^{(m)}_{s,k} \mathfrak{I}^{(m)}_s(\Theta)_{p,q}$
    end for
    $\mathcal{I} \leftarrow \mathcal{I} + \theta_k \mathcal{I}_k$
end for
return $\mathcal{I}$
```

In Algorithm 1, if `length` is the length of a given sequence I_k, then the whole algorithm consists of $\sim \sum_{k=1}^{K}$ `length`(I_k) computations of $\mathfrak{I}^{(m)}_s(\Theta)$.

3.7.2 Equivalent partial samples

In order to continue our discussion on faster algorithms to compute $\mathfrak{I}^{(m)}_{s=s^{(k)}_{[a,b)}}(\Theta)$, we introduce the concept of equivalent partial samples below:

Definition 10. Two partial samples s_1 and s_2 are equivalent w.r.t. $Samp_m$ if and only if $\mathfrak{I}^{(m)}_{s_1}(\Theta) = \mathfrak{I}^{(m)}_{s_2}(\Theta)$.

Lemma 1. *If $\forall I_k \in I$, $\delta_{s_1,k}G^{(m)}_{s_1,k} = \delta_{s_2,k}G^{(m)}_{s_2,k}$, then s_1 and s_2 are equivalent w.r.t. $Samp_m$.*

Proof. According to Equation 3.58, we have: $\mathfrak{I}^{(m)}_{s_1}(\Theta) = \mathfrak{I}^{(m)}_{s_2}(\Theta)$. Thus the two partial samples are equivalent w.r.t. $Samp_m$ according to Definition 10. □

Definition 11. A set of partial samples S is an equivalent sample set w.r.t. $Samp_m$ if and only if $\forall s_1, s_2 \in S$, s_1 and s_2 are equivalent w.r.t. $Samp_m$.

Lemma 2. *Given an isoform I_k and a sampling method $Samp_m$, if we divide all its possible partial samples into n non-overlapping equivalent sample sets $S_1, S_2, ..., S_n$, then:*

$$\mathcal{I}^{(m)}(\Theta)_{p,q} \quad = \quad \sum_{k=1}^{K} \theta_k \sum_{i=1}^{n} |S_i| G^{(m)}_{s_i,k} \mathfrak{I}^{(m)}_{s_i}(\Theta)_{p,q}, \textit{ for any } s_i \in S_i \tag{3.59}$$

Proof. We can rewrite the $\sum_{s=s^{(k)}_{[a,b)};\forall[a,b)\in I_k} G^{(m)}_{s,k}\mathfrak{I}^{(m)}_{s=s^{(k)}_{[a,b)}}(\Theta)_{p,q}$ part in Equation 3.56 by dividing all the possible $s^{(k,m)}_{a,b}$s into equivalent sample sets $S_1, S_2, ..., S_n$, and then obtain the equation above. □

3.7.3 Results from a simple shotgun read generation model

In this section, we consider a set of simplified partial sample generation models:

Definition 12. A simple shotgun sampling method $Samp_m$ generates samples with fixed read length r_m. When sampling from an isoform I_k with length l_k, there are in total $l_k - r_m + 1$ different samples $s^{(k)}_{[a,b)}$, where $a = 0, 1, 2, ..., (l_k - r_m)$; and $b = a + r_m$. Each of these samples has equal probability of being generated from I_k: $G^{(m)}_{s,k} = 1/(l_k - r_m + 1)$.

Figure 3.3 illustrates simple shotgun sampling process and its corresponding per-base coverage on the isoform being sampled.

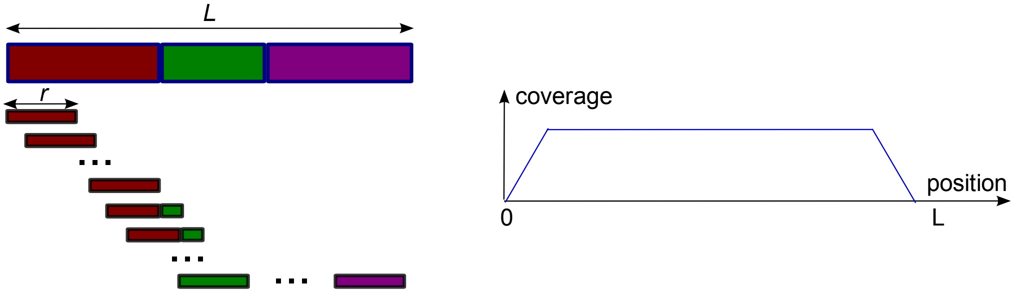


Figure 3.3. A simple shotgun read generation model.

Lemma 3. *Given the sample generation model $Samp_m$ above, if two samples s_1 and s_2 generated by this method are compatible with the same set of isoforms, i.e. $\delta_{s_1,k} = \delta_{s_2,k}, \forall I_k \in I$, then s_1 and s_2 are equivalent w.r.t. $Samp_m$.*

Proof. If $\delta_{s_1,k} = \delta_{s_2,k} = 0$, then obviously $\delta_{s_1,k}G^{(m)}_{s_1,k} = \delta_{s_2,k}G^{(m)}_{s_2,k} = 0$. Otherwise, if $\delta_{s_1,k} = \delta_{s_2,k} = 1$, then both s_1 and s_2 are partial samples that may be generated by $Samp_m$ from I_k. In this case, according to Definition 12, $G^{(m)}_{s_1,k} = G^{(m)}_{s_2,k} = 1/(l_k - r_m + 1)$. Thus we always have: $\delta_{s_1,k}G^{(m)}_{s_1,k} = \delta_{s_2,k}G^{(m)}_{s_2,k}$, $\forall I_k \in I$. According to Lemma 1, s_1 and s_2 are equivalent w.r.t. $Samp_m$. □

Theorem 1. *Given the sample generation model $Samp_m$ above, if two samples s_1 and s_2*

generated by this method overlap with all the junctions in the same set of connected exons $e_{k_1} \to e_{k_2} \to ... \to e_{k_n}$*, then* s_1 *and* s_2 *are equivalent w.r.t.* $Samp_m$*.*

Proof. We first prove by contradiction that $\forall I_k \in I$, $\delta_{s_1,k} = \delta_{s_2,k}$:

If $\delta_{s_1,k} \neq \delta_{s_2,k}$, without loss of generality, we assume that $\delta_{s_1,k} = 1$ and $\delta_{s_2,k} = 0$. Then there must exist an exon junction $e_{k_i} \to e_{k_{i+1}}$, $i \in \{1, 2, ..., n-1\}$, such that $e_{k_i} \to e_{k_{i+1}}$ is not compatible with I_k (otherwise, as a part of $e_{k_1} \to e_{k_2} \to ... \to e_{k_n}$, s_2 will be compatible with I_k). Since s_1 overlaps with the junction of $e_{k_i} \to e_{k_{i+1}}$, s_1 is not compatible with I_k either, which will lead to a contradiction to the previous assumption that $\delta_{s_1,k} = 1$. Thus the original statement $\delta_{s_1,k} = \delta_{s_2,k}$ must be true.

Then according to Lemma 3, s_1 and s_2 are equivalent w.r.t. $Samp_m$. □

For example, in Figure 3.4, where the reads are generated from a simple shotgun sampling process, the equivalent partial samples are {read1, read2, read9}, {read10, read11}. Also, if we consider a paired-end read as a long shotgun read with its gap filled, the samples read5 and read6 are also (approximately) equivalent, if their insert sizes are close to each other. However, read12 is not equivalent to these reads, since its shotgun version overlaps with a different exon junction set (with an addition exon).

3.7.4 Algorithms for efficiently computing $\mathcal{I}^{(m)}(\Theta)$

Based on Definition 12 and Theorem 1, we can design the following algorithm for efficiently computing $\mathcal{I}^{(m)}(\Theta)$ by combining the computation of this value for equivalent partial samples from each isoform.

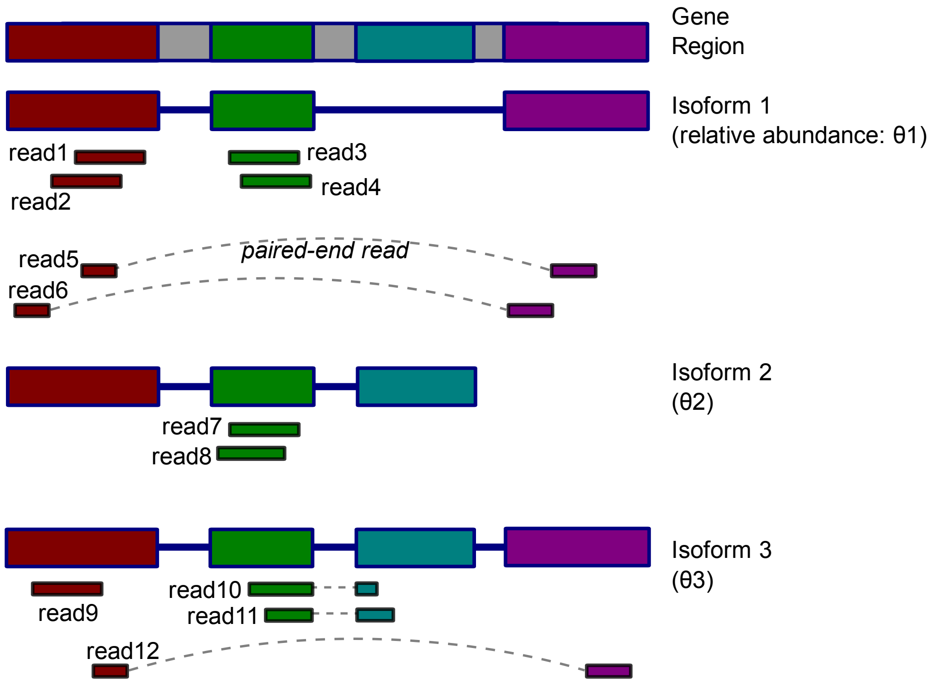


Figure 3.4. Equivalent samples in a simple shotgun read generation model.

Algorithm 2 $\textsc{FastShotgun}\text{FIM}(I, \Theta, Samp_m, p, q)$

```
GIVEN: Possible isoforms $I = \{I_1, I_2, ..., I_K\}$;
    Relative abundances $\Theta = (\theta_1, \theta_2, ..., \theta_K)$;
    Sampling method $Samp_m$ as in Definition 12;
    Integer $p, q \in \{1, 2, ..., K-1\}$.
COMPUTE: The value of $\mathcal{I}^{(m)}(\Theta)_{p,q}$.

$\mathcal{I} \leftarrow 0$
for all $I_k \in I$ do
    $\mathcal{I}_k \leftarrow 0$
    $a \leftarrow 0$
    while $a \leq \texttt{length}(I_k) - r_m$ do
        $b \leftarrow a + r_m$;
        $s \leftarrow s^k_{[a,b)}$
        $(e_{k_1} \rightarrow e_{k_2} \rightarrow ... \rightarrow e_{k_n}) \leftarrow \texttt{overlappingExons}(s, I_k)$
        $N_{EqSamples} \leftarrow \min\left(\sum_{e_{k'} \in I_k; k' <= k_1} \texttt{length}(e_{k'}) - a, \sum_{e_{k'} \in I_k; k' <= k_n} \texttt{length}(e_{k'}) - b + 1\right)$
            {Get the number of equivalent samples}
        $\mathcal{I}_k \leftarrow \mathcal{I}_k + N_{EqSamples} G^{(m)}_{s,k} \mathfrak{I}^{(m)}_s(\Theta)_{p,q}$
        $a \leftarrow a + N_{EqSamples}$ {Move $a$ to the beginning of the next equivalent sample set}
    end while
    $\mathcal{I} \leftarrow \mathcal{I} + \theta_k \mathcal{I}_k$
end for
return $\mathcal{I}$
```

In Algorithm 2, $\texttt{overlappingExons}(s, I_k)$ identifies the connected exons set in I_k that overlaps with a given partial sample s, and can be implemented with $O(\log NumExons_k)$ time complexity by pre-computing an exon-position index table for the isoforms.

We can further reduce the number of times of computing $\mathfrak{I}^{(m)}_s(\Theta)$ by combining equivalent partial samples from across isoforms: when an equivalent sample set from an isoform has been identified, all the same samples from other isoforms can be recorded in lists of intervals to avoid redundant computation of their $\mathfrak{I}^{(m)}_s(\Theta)$s. The algorithm is shown below:

Algorithm 3 FASTERSHOTGUNFIM$(I, \Theta, Samp_m, p, q)$

```
Require: Possible isoforms $I = \{I_1, I_2, ..., I_K\}$;
    Relative abundances $\Theta = (\theta_1, \theta_2, ..., \theta_K)$;
    Sampling method $Samp_m$ as in Definition 12;
    Integer $p, q \in \{1, 2, ..., K-1\}$.
Ensure: The value of $\mathcal{I}^{(m)}(\Theta)_{p,q}$.
$\mathcal{I} \leftarrow 0$
for all $I_k \in I$ do
  $CoveredSampleStarts_k \leftarrow$ empty interval list
end for
for all $I_k \in I$ do
  $a \leftarrow \texttt{minNotCoveredStart}(CoveredSampleStarts_k, Samp_m)$
  while $a \leq \texttt{length}(I_k) - r_m$ do
    $b \leftarrow a + r_m$;
    $s \leftarrow s^k_{[a,b)}$
    $(e_{k_1} \rightarrow e_{k_2} \rightarrow ... \rightarrow e_{k_n}) \leftarrow \texttt{overlappingExons}(s, I_k)$
    $N_{EqSamples} \leftarrow \min\left(\sum_{e_{k'} \in I_k; k' <= k_1} \texttt{length}(e_{k'}) - a, \sum_{e_{k'} \in I_k; k' <= k_n} \texttt{length}(e_{k'}) - b + 1\right)$
    $\mathcal{I} \leftarrow \mathcal{I} + \theta_k N_{EqSamples} G^{(m)}_{s,k} \mathfrak{I}^{(m)}_s(\Theta)_{p,q}$
    $CoveredSampleStarts_k \leftarrow CoveredSampleStarts_k + [a, a + N_{EqSamples})$
    for all $I_{k'} \neq I_k$ do
      if $I_{k'}$ contains $(e_{k_1} \rightarrow e_{k_2} \rightarrow ... \rightarrow e_{k_n})$ then
        $s' \leftarrow s^{k'}_{[a',b')} \leftarrow \texttt{firstSample}(I_{k'}, Samp_m, (e_{k_1} \rightarrow e_{k_2} \rightarrow ... \rightarrow e_{k_n}))$
        $\mathcal{I} \leftarrow \mathcal{I} + \theta_{k'} N_{EqSamples} G^{(m)}_{s',k'} \mathfrak{I}^{(m)}_s(\Theta)_{p,q}$ {Use previously computed $\mathfrak{I}^{(m)}_s(\Theta)_{p,q}$}
        $CoveredSampleStarts_{k'} \leftarrow CoveredSampleStarts_{k'} + [a', a' + N_{EqSamples})$
      end if
    end for
    $a \leftarrow \texttt{minNotCoveredStart}(CoveredSampleStarts_k, Samp_m)$
  end while
end for
return $\mathcal{I}$
```

In Algorithm 3, $\texttt{minNotCoveredStart}(CoveredSampleStarts_k, Samp_m)$ finds the minimum position $a \in \{0, 1, ..., \texttt{length}(I_k) - r_m + 1\}$ that is outside a given interval list $CoveredSampleStarts_k$; $\texttt{firstSample}(I_k, Samp_m, ConnectedExonSet)$ returns the partial sample $s^k_{[a,b)}$ from I_k covering all the exon junctions in $ConnectedExonSet$ with a minimum a, and can be implemented with a worst-case $O(\log NumExons_k + |ConnectedExonSet|)$ time complexity by using a pre-computed exon position index table for the isoforms.

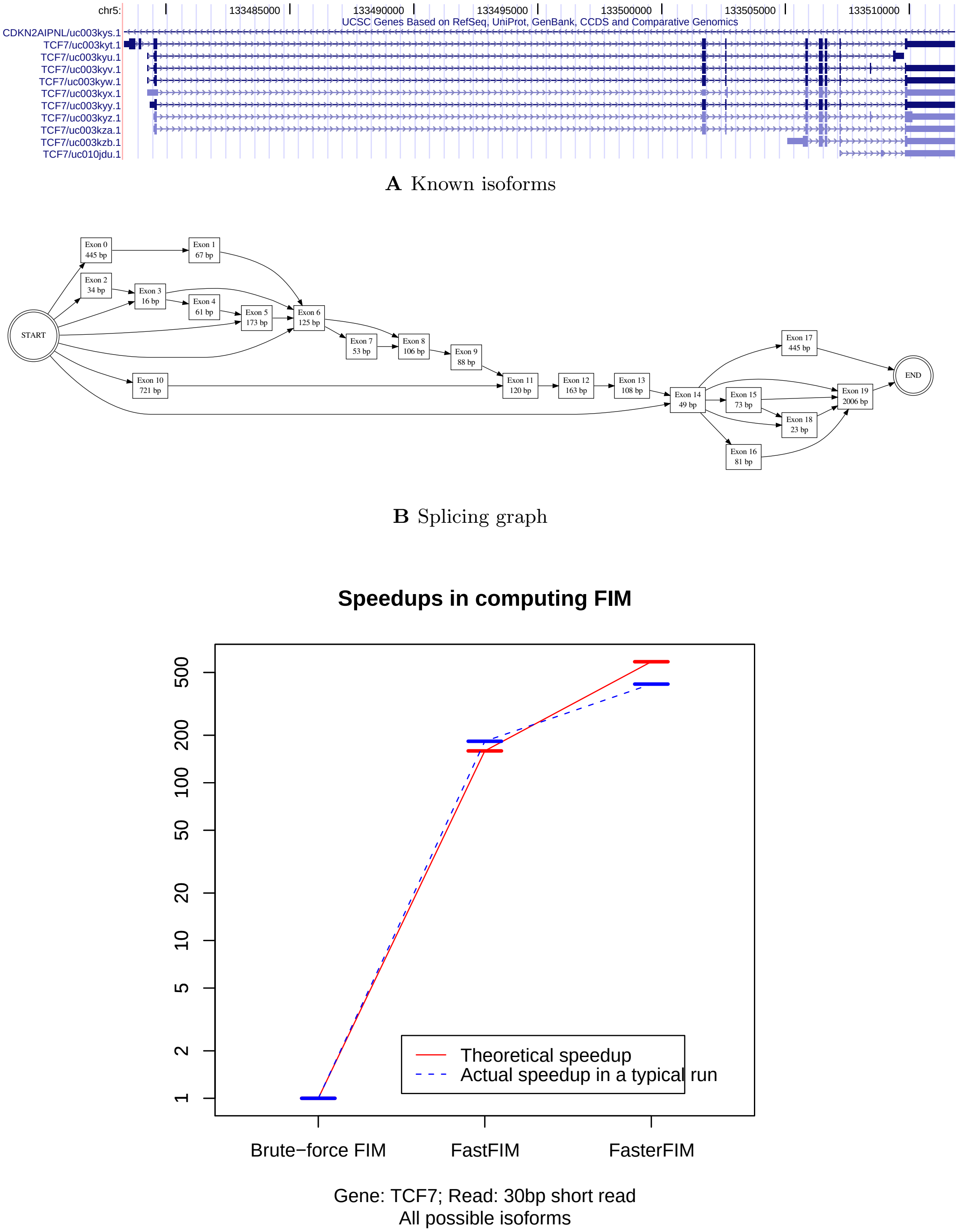
3.7.5 Complexity analysis

Given a set of K possible isoforms $I = \{I_1, I_2, ..., I_K\}$, with lengths $l_1, l_2, ..., l_K$, respectively, and a shotgun sampling method $Samp_m$ with sample length r_m as described in Definition 12, Algorithm 1 requires $\sum_{k=1}^{K} l_k$ steps of computing $\mathfrak{I}_s^{(m)}(\Theta)_{p,q}$. Thus computing $\mathcal{I}^{(m)}(\Theta)$ using this brute-force algorithm requires $(K-1)^2 \cdot \sum_{k=1}^{K} l_k$ operations of calculating $\mathfrak{I}_s^{(m)}(\Theta)_{p,q}$. If we assume that the average length of an isoform is $l_{AvgIsoform}$, this corresponds to $\sim K^3 \cdot l_{AvgIsoform}$ computations of $\mathfrak{I}_s^{(m)}(\Theta)_{p,q}$.

Suppose that on average an isoform can be divided into $N_{EqSampleSets}$ equivalent sample sets by Algorithm 2, this algorithm will then require $\sim K^3 \cdot N_{EqSampleSets}$ steps of computing $\mathfrak{I}_s^{(m)}(\Theta)_{p,q}$ to obtain the Fisher information matrix $\mathcal{I}^{(m)}(\Theta)$ for the given sampling method, thus being more efficient than Algorithm 1 by a ratio of $l_{AvgIsoform}/N_{EqSampleSets}$. Algorithm 3 will obviously be even more efficient by avoiding the redundant computation of some of the equivalent sample sets in Algorithm 2.

3.7.6 Application on a typical gene

We consider the gene $TCF7$, which has 10 known isoforms shown in Figure 3.5A. Figure 3.5B shows its corresponding splicing graph [27, 82], with 19 exon blocks, and 96 possible isoforms, which are all the possible paths from node “START” to node “END” in the splicing graph.



A Known isoforms

B Splicing graph

C Speedup

Figure 3.5. Speedup in FIM computation for gene TCF7.

When computing $\mathcal{I}_s^{(m)}(\Theta)$, Algorithm 1 requires 26902 computations of $\mathfrak{I}_s^{(m)}(\Theta)$, while Algorithm 2 involves 169 such computations, and the number for Algorithm 3 is 46, achieving a $\sim$ 585 times speedup compared to the brute-force method. A summary of the speedups is shown in Figure 3.5C.

3.8 Simulation Results

3.8.1 Simulation on simplified genes

Due to the complexity of real gene structure, we apply our methods to three artificially constructed genes with simplified isoform structures, so as to better illustrate how different characteristics of the gene structures can affect the outcome of the isoform quantification analysis.

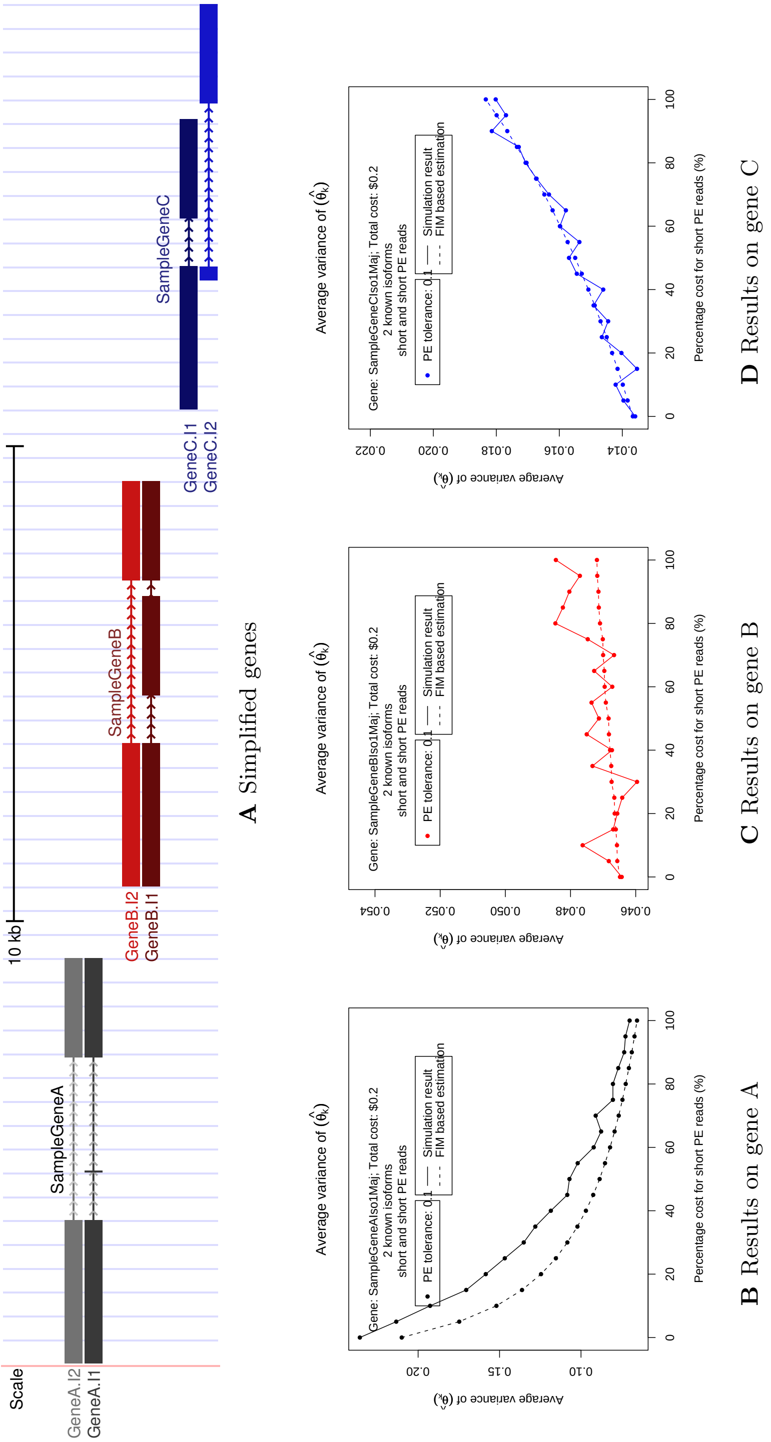
As shown in Figure 3.6A, each of these genes has two different isoforms, with the more abundant one shown in a darker color. Two sampling techniques, short single and short paired-end (PE), are used to generate reads from them, with a fixed total cost of \$0.20. The per-base costs of these sampling techniques are based on Table 4.1. Different cost combinations, e.g. different percentage of the total cost being assigned to a certain sampling technique, are illustrated by the x-axis in Figure 3.6B-D. For each of these cost combinations, we randomly generate 1000 read sets, and use the MLE solution to estimate $\hat{\Theta}$, based on which $Average\left(var(\hat{\theta}_k)\right)$ are computed (solid lines in Figure 3.6B-D). We also use Algorithm 3 to estimate the same quantity, and plot the estimations using dashed lines in the same figure for comparison. The results show that the FIM estimation of $Average\left(var(\hat{\theta}_k)\right)$ are close to the direct simulation results, and also correctly predicts the

Table 3.1. Total time used by brute-force simulation vs. FIM based heuristic to estimate $Average\left(var(\hat{\theta}_k)\right)$ in simplified genes

Total trials for one gene	Number of trials $\times$ Number of sampling method combinations $= 1000 \times 21$
Total FIM computation for one gene	Number of sampling methods$= 2$
Total CPU time used by brute-force simulation	~ 52 minutes
Total CPU time used by FIM based heuristic	< 1 second

trend in how this value changes with different cost combinations. Also, different gene structures have noticeable impact on the MLE accuracy, mostly due to the ability of sampling techniques to distinguish isoforms from each other with different gene structures.

Not only can the FIM based heuristic correctly approximate how the performance of MLE changes with regard to different sampling technique combinations, it is also able to dramatically shorten the time of computation, as shown in Table 3.1. This is mainly because while the computation of brute-force simulation depends heavily on the number of reads being generated and the number of trials needed to obtain a relatively stable estimation of $Average\left(var(\hat{\theta}_k)\right)$, the core computation taken by the FIM based heuristic is the evaluation of individual FIMs for the sampling techniques involved, which can be efficiently computed using Algorithm 3, and then combined based on Equation 3.53 to estimate $Average\left(var(\hat{\theta}_k)\right)$ under different cost combinations.



A Simplified genes

B Results on gene A

C Results on gene B

D Results on gene C

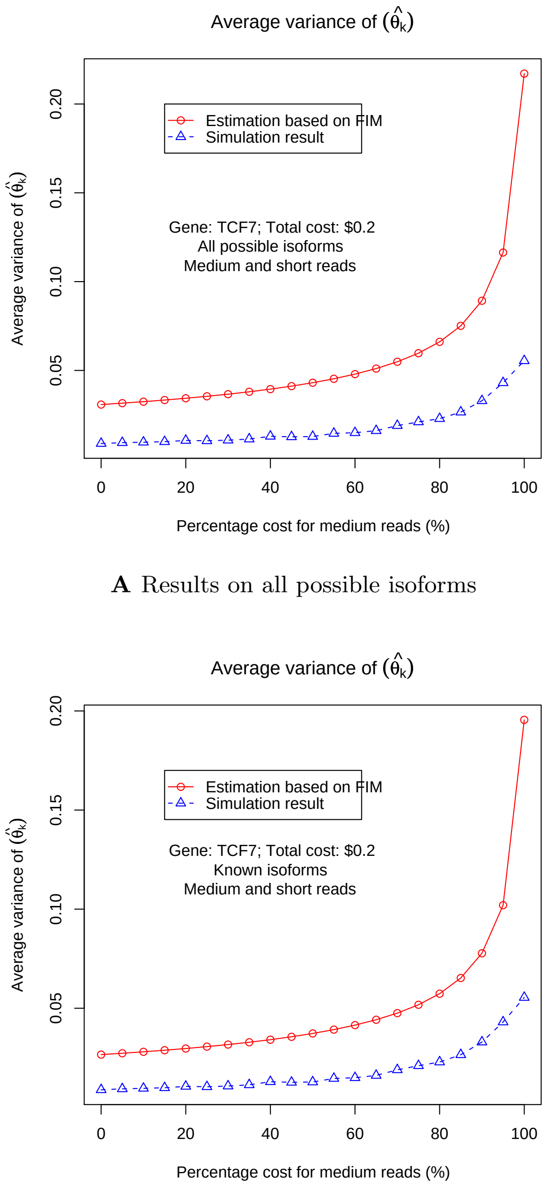
Figure 3.6. Results on simplified genes.

3.8.2 Simulation on real genes

We present in this section the application of the FIM based heuristic on a real gene, and compared the results to the ones obtained from direct simulations. We pick TCF7 again as a typical example gene with multiple isoforms. Similarly to our procedure in the previous section, two sampling techniques, medium and short shotgun sequencing, are used to generate reads from them, with a fixed total cost of $0.2, with 200 trials being conducted for each cost combination. Two different sets of results are shown in Figure 3.7, one using all the 96 possible isoforms deduced from its splicing graph, and the other just using its 10 known isoforms. As in the previous section, the results here show that the FIM estimation of $Average\left(var(\hat{\theta}_k)\right)$ are close to the direct simulation results, and also correctly predicts the trend in how this value changes with different cost combinations.

Figure 3.8 presents a more detailed simulation focusing on short paired-end reads. The *tolerance* value reflects the expectation of the variance in insert size for such experiments: a 0 value means that all the paired-end reads are expected to have exactly the same insert size; the higher the *tolerance* is, the more relaxed are we on the insert size variation. As we can see from this figure, the higher the *tolerance*, the larger $Average\left(var(\hat{\theta}_k)\right)$ becomes, corresponding to a worse expected performance of MLE. This can be explained by the fact that a higher *tolerance* makes the sampling method less capable of distinguishing highly similar isoforms from each other based on a single paired-end read (e.g. GeneA in Figure 3.6A). The FIM based heuristic is again able to correctly depict the different trends of MLE performance under different cost combinations and *tolerance* settings.

We also show the computation time used by brute-force simulation and FIM based heuristic in Table 3.2. Note that the brute-force simulation is even more computational



A Results on all possible isoforms

B Results on known isoforms

Figure 3.7. Simulation results on TCF7.

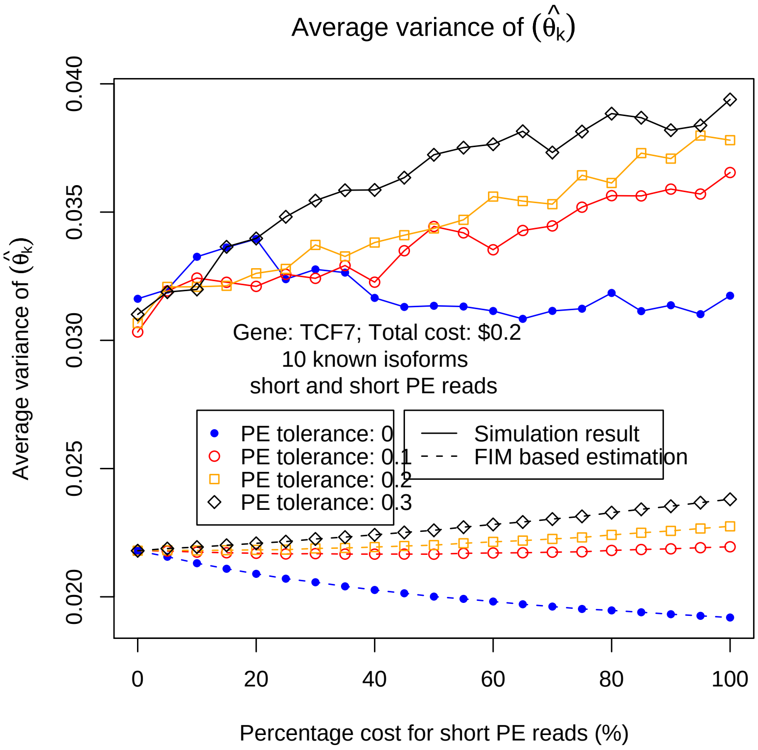


Figure 3.8. Simulation results on TCF7 with paired-end reads.

Table 3.2. Total time used by brute-force simulation vs. FIM based heuristic to estimate $Average\left(var(\hat{\theta}_k)\right)$ **in TCF7**

Total trials for one gene	Number of trials × Number of sampling method combinations = 200 × 21
Total FIM computation for one gene	Number of sampling methods= 2
Total CPU time used by brute-force simulation	$\sim$ 10.6 hours
Total CPU time used by FIM based heuristic	$<$ 1 second

consuming, mainly because more isoforms are involved in the MLE process. Given the fact that there exist more than 20000 genes in the human genome and that the simulation has to be rerun for every new experiment to adjust its read counts, using the FIM based heuristic instead for the purpose of estimating isoform quantification accuracy is obviously a more computationally tractable choice.

3.9 Application to a worm dataset

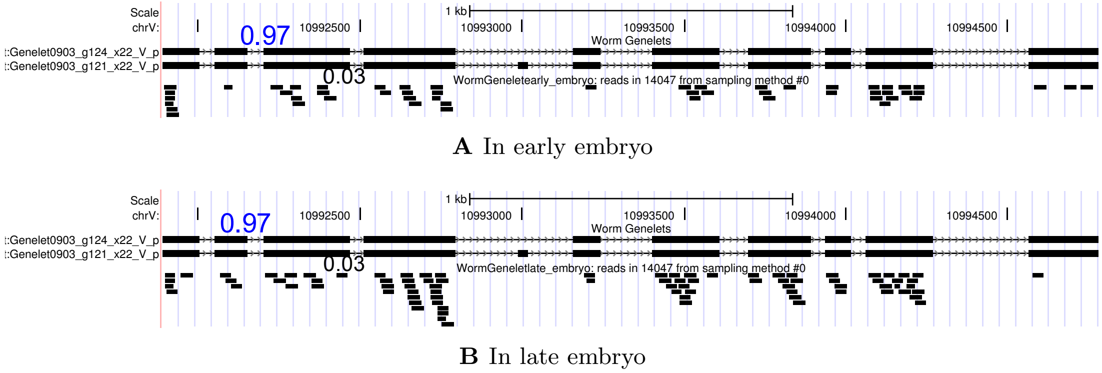
We further apply our MLE solution to a worm dataset [29], which includes multiple developmental stages, so as to compare the results on a same set of isoforms under different conditions. The worm genome contains $\sim$ 20K genes, and the transcripts from each stage are sequenced with $\sim$ 50M short Solexa reads.

3.9.1 Comparison of isoform composition between stages

We first present the isoform quantification results on individual genes in two different stages, early embryo (EE) and late embryo (LE), to briefly illustrate the fact that different genes have different isoform composition differences between stages. Here we use the following formula to measure the difference in isoform composition of the same gene in two different stages:

$$Diff_{gene_i}(\Theta^{(Stage1)}, \Theta^{(Stage2)}) = \frac{\sum_{k=1}^{K} (\theta_k^{(Stage1)} - \theta_k^{(Stage2)})^2}{K} \tag{3.60}$$

where K is the total number of isoforms in gene $gene_i$.

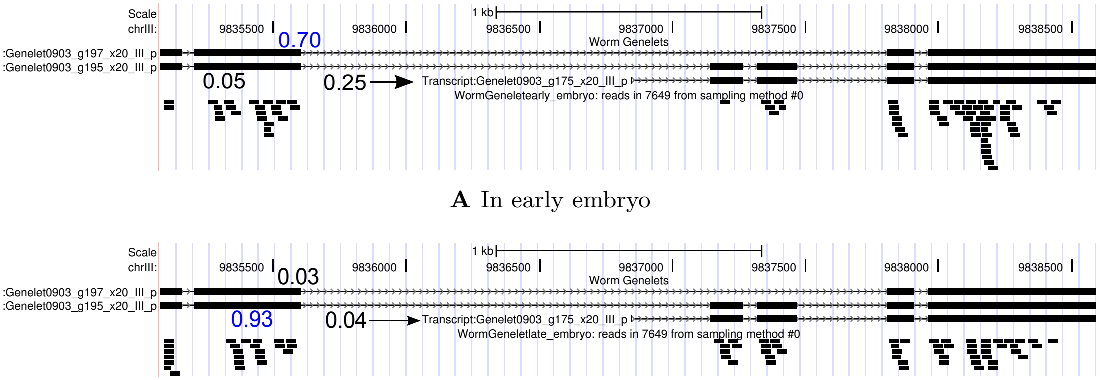

Scale
chrV:
1 kb
10992500
10993000
10993500
10994000
10994500
Worm Genelets
::Genelet0903_g124_x22_V_p
::Genelet0903_g121_x22_V_p
0.97
0.03
WormGeneletearly_embryo: reads in 14047 from sampling method #0
WormGeneletlate_embryo: reads in 14047 from sampling method #0

A In early embryo

B In late embryo

Figure 3.9. Gene 14047 in two stages ($Diff = 0$).

Figure 3.9 and Figure 3.10 show two examples of zero and non-zero *Diff* values. The reads are plotted below the isoforms, and the numbers associated with the isoforms are their estimated relative abundances based on MLE. Furthermore, if we compute such values for all the genes in these two stages, we can get a histogram of isoform composition differences as illustrated in Figure 3.11, which characterizes the general isoform composition difference


Scale
chrIII:
1 kb
9835500
9836000
9836500
9837000
9837500
9838000
9838500
Worm Genelets
:Genelet0903_g197_x20_III_p
:Genelet0903_g195_x20_III_p
0.70
0.05
0.25
Transcript:Genelet0903_g175_x20_III_p
WormGeneletearly_embryo: reads in 7649 from sampling method #0
0.03
0.93
0.04
WormGeneletlate_embryo: reads in 7649 from sampling method #0

A In early embryo

B In late embryo

Figure 3.10. Gene 7649 in two stages ($\mathit{Diff} = 0.42$).

between stages.

3.9.2 The effect of different isoform sets on MLE result

We also investigate how different isoform sets (e.g. with a major/minor isoform missing, with an additional “dummy” isoform) will affect the MLE result, especially in terms of the maximized likelihood value. We pick gene 7649 as a base isoform set, using the same set of reads and the per-read average maximized likelihood value LL to measure the goodness of fitting:

$$LL_{gene_i} = \sum_{r \in R} \log \sum_{k=1}^{K} \delta_{r,k} \theta_k G_{r,k} \tag{3.61}$$

As we can see from Figure 3.12, Figure 3.13 and Figure 3.14, the LL value always decreases when we modify the “true” isoform set in an unfavorable fashion. This shows that the likelihood score is an effective metric for ranking isoform sets for a particular gene.

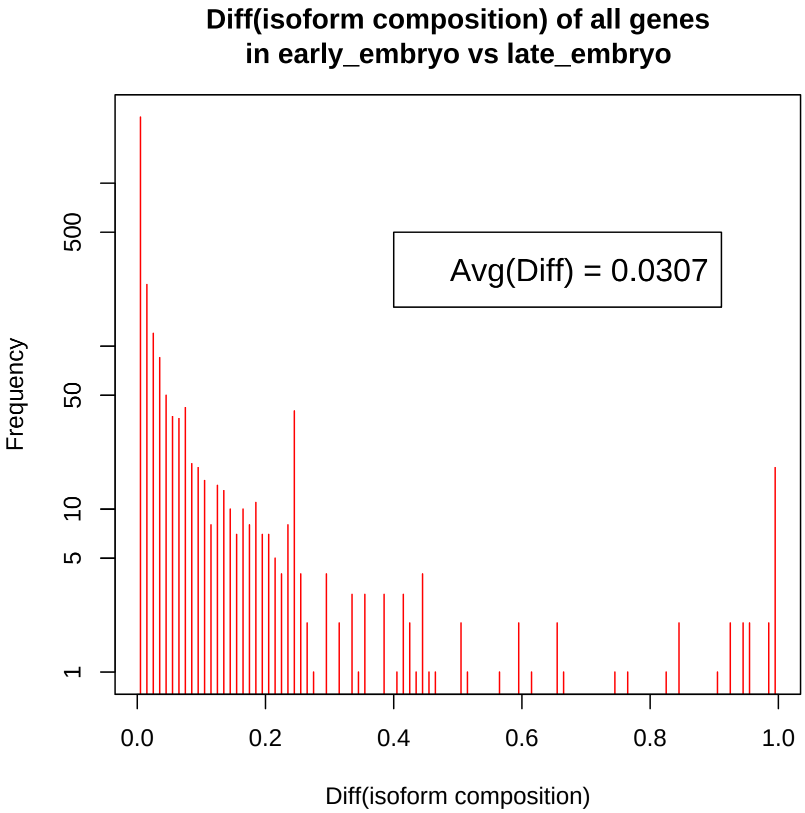
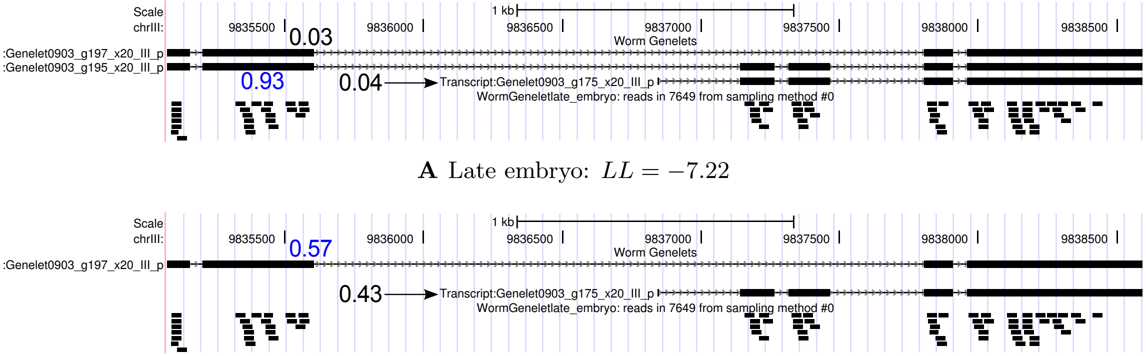


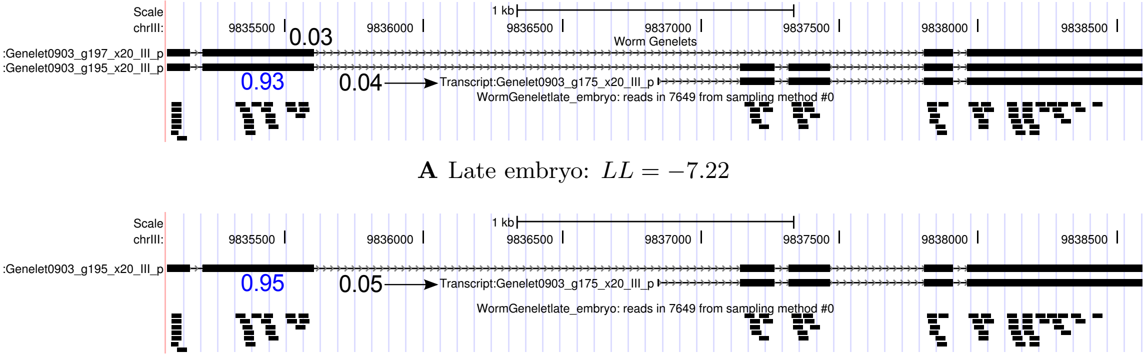
Figure 3.11. $Diff$ **of all genes in two stages.**


Scale
chrIII:
1 kb
9835500
9836000
9836500
9837000
9837500
9838000
9838500
Worm Genelets
:Genelet0903_g197_x20_III_p
:Genelet0903_g195_x20_III_p
0.03
0.93
0.04
Transcript:Genelet0903_g175_x20_III_p
WormGeneletlate_embryo: reads in 7649 from sampling method #0
0.57
0.43

A Late embryo: $LL = -7.22$

B Late embryo: leave out the dominant isoform: $LL = -7.35$

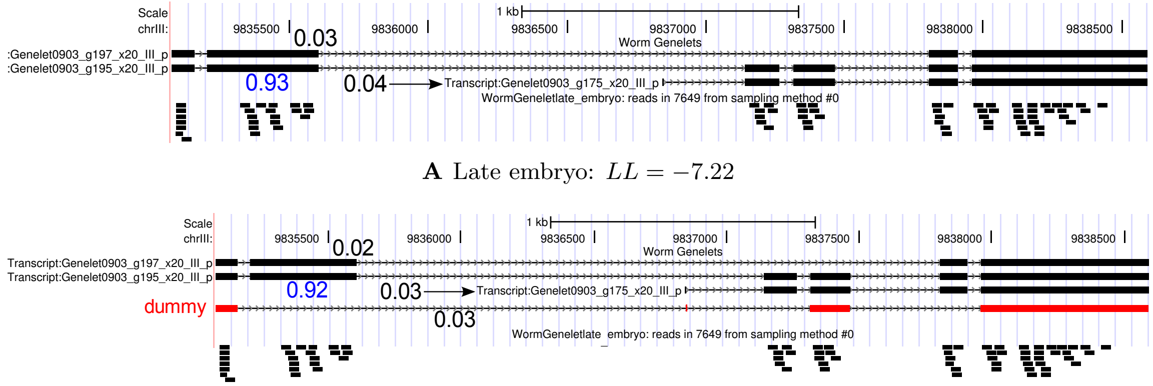
Figure 3.12. Gene 7649: Leave out the dominant isoform.


Scale
chrIII:
1 kb
9835500
9836000
9836500
9837000
9837500
9838000
9838500
Worm Genelets
:Genelet0903_g197_x20_III_p
:Genelet0903_g195_x20_III_p
0.03
0.93
0.04
Transcript:Genelet0903_g175_x20_III_p
WormGeneletlate_embryo: reads in 7649 from sampling method #0
0.95
0.05

A Late embryo: $LL = -7.22$

B Late embryo: leave out a non-dominant isoform: $LL = -7.29$

Figure 3.13. Gene 7649: Leave out a non-dominant isoform.



A Late embryo: $LL = -7.22$

B Late embryo: add a "dummy" isoform: $LL = -7.29$

Figure 3.14. Gene 7649: add a "dummy" isoform.

3.10 Conclusion and Discussion

In this chapter we explore the problem of integrating different sequencing techniques to quantify the relative abundance of different isoform transcripts, which can be generalized to the problem of estimating the distribution based on partial samples from different sampling techniques. We first introduce a statistical framework to model the generative process of the partial samples, using a "pluggable" function G to allow flexible incorporation of different sampling characteristics, and then present the original problem as a maximum likelihood estimation (MLE) problem, with an iterative solution based on expectation maximization, which guarantees a local optimal answer. This provides a solution to the question of estimating a distribution based on partial samples.

In order to further investigate the problem involving partial samples, we introduce a heuristic based on the Fisher information matrix (FIM) to estimate the variance of the previously presented MLE solution. Also, in order to accelerate the computation of this measurement, we introduce the concept of equivalent partial samples and develop a fast algorithm, Algorithm 3, to accurately calculate FIM, achieving a speedup of ~ 500 times compared to the brute-force method. Simulation results on both hypothetical and real gene models also show that our FIM-based heuristic gives a good approximation to the value of $Average\left(var(\hat{\theta}_k)\right)$, and accurately predicts the numeric order of this value under different conditions. With this metric, we are also able to demonstrate examples of how to efficiently find low-cost combinations of different sampling techniques to best estimate the isoform compositions in RNA-seq experiments. Although we are only using individual genes as examples, once we have good assumptions of expression levels of different genes, this procedure can be generalized to all the genes for the low-cost design of actual whole genome RNA-seq experiments.

What is more, by applying the MLE method to a worm RNA-seq dataset, we illustrate how we can compare the differential isoform composition between different developmental stages, and how different isoform sets (e.g. with a major/minor isoform missing, with an additional ‘dummy” isoform) will affect the MLE result, especially in terms of the maximized likelihood value, showing that the likelihood score is an effective tool for ranking the “fitness” of isoform sets for a particular gene.

3.10.1 Using more complex G function in Algorithm 3

The sequencing technology being used in an RNA-seq experiment is usually more complicated than the simplified G function described in Definition 12, which assumes equal sample-length and uniform generative probability. In reality, a typical G usually involves reads with different lengths within a certain range, and also biased sample generation probability at different locations of a full-length isoform. Although once such a G is defined, our MLE solution can treat it in the same way as it does for simplified versions, Algorithm 3 no longer works "out of the box" due to its dependency on Definition 12 to find equivalent partial samples. We discuss briefly in this subsection on how to handle more complex features.

When the assumption of uniform sample generation still holds, it is straightforward to handle samples with different lengths in FIM computation. We can treat one sampling method as a combination of multiple simplified methods as in Definition 12, with different sample lengths $\{l_1, \cdots, l_L\}$:

$$\mathcal{I}^{(m)}(\Theta)_{p,q} = \sum_{l=l_1}^{l_L} Pr_m\{length(s) = l\} \mathcal{I}^{(m_l)}(\Theta)_{p,q} \tag{3.62}$$

$$= \sum_{l=l_1}^{l_L} Pr_m\{length(s) = l\} \left[-\frac{\partial^2 \log \sum_{k=1}^{K} \delta_{s_l,k} \theta_k G_{s_l,k}^{(m,l)}}{\partial\theta_p \partial\theta_q} \right] \tag{3.63}$$

where $Pr_m\{length(s) = l\}$ represents the probability of generating a sample with length l in sampling method $Samp_m$, s_l is a sample with length l, and $G_{s_l,k}^{(m,l)}$ is the simplified sample generation probability as in Definition 12, with sample length l.

In the case of non-uniform sample generation along the isoform, if $G_{s,k}^{(m)}$ is a step function (piece-wise constant function) for sample s along isoform I_k, we will still be able to find equivalent sample sets as described in Definition 11, based on both the isoform structures and the intervals in G. If, however, very few such constant components exist in G, we will need to relax the definition of equivalent partial samples to satisfying $\delta_{s_1,k} = \delta_{s_2,k}$ only. With this relaxed definition, we can find samples S_{eq} with equivalent structural similarities to all the isoforms. In this case, if the isoforms contain regions where any s_1 and s_2 from it satisfy $G_{s_1,k}^{(m)} = c_{s_1,s_2} \cdot G_{s_2,k}^{(m)}$ with a constant c_{s_1,s_2} for all k, we still have

$\mathfrak{I}_{s_1}^{(m)}(\Theta)_{p,q} = \mathfrak{I}_{s_2}^{(m)}(\Theta)_{p,q}$ according to Equation 3.58, and the $\mathcal{I}^{(m)}$ can thus be efficiently computed using a variant of Algorithm 3 by combining the compution for such equivalent partial samples. For more complex G functions, however, approximation algorithms may have to be introduced for fast computation of $\mathcal{I}^{(m)}$.

Part III

Efficient Simulation of Nondeterministic Sampling Process

Chapter 4

Optimal Low Cost Integration of Sampling Techniques in Re-sequencing

4.1 Introduction

The human genome is comprised of approximately 6 billion nucleotides on two pairs of 23 chromosomes. Variations between individuals are comprised of $\sim$ 6 million single nucleotide polymorphisms (SNPs) and $\sim$ 1000 relatively large structural variants (SVs) of $\sim$ 3kb or larger and many more smaller SVs are responsible for the phenotypic variation among individuals [28, 40]. Most of these large SVs are due to genomic rearrangements (e.g. duplication and deletion), and a few others contain novel sequences that are not present in the reference genome [45]. The goal of personal genomics is to determine all these genetic differences between individuals and to understand how these contribute to phenotypic differences in individuals.

Making personal genomics almost a reality over the past decade, the development of high throughput sequencing technologies has enabled the sequencing of individual genomes [45, 81]. In 2007, Levy et al. reported the sequencing of an individuals genome based on Sanger [70] whole-genome shotgun sequencing, followed by de novo assembly strategies. [81] presented another individuals genome sequence constructed from 454 sequencing reads [51] and comparative genome assembly methods. In the mean time, other new sequencing technologies such as Solexa/Illumina sequencing [5] have become available for individual genome sequencing with corresponding, specially-designed sequence assembly algorithm designed [16, 80, 10, 84, 62].

These projects and algorithms, however, mostly relied on a single sequencing technology to perform individual re-sequencing and thus did not take full advantage of all the existing experimental technologies. Table 4.1 gives a summary of the characteristics of several technologies in comparative individual genome sequencing. At one extreme, performing long Sanger sequencing with a very deep coverage will lead to excellent results at high cost. In another, performing only the inexpensive and short Illumina sequencing may generate good and cost-efficient results in SNP detection, but will not be able to either unambiguously locate some of the SVs in repetitive genomic regions or fully reconstruct many of the large SVs. Moreover, array technologies such as the SNP array [28] and the CGH array at different resolutions [58, 74, 75, 56] can also be utilized to identify the SVs: the SNP arrays can detect SNPs directly, and the CGH array is able to detect kilobase-(kb) to megabase-(mb) sized copy number variants (CNV) [66], which can be integrated into the sequencing-based SV analysis. It is thus advantageous to consider optimally combining all these experimental techniques into the individual genome re-sequencing framework and to design experiment protocols and computational algorithms accordingly.

Due to the existence of reference genome assemblies [32, 77] and the high similarity between an individuals genome and the reference [45], the identification of small SVs is relatively straightforward in comparative re-sequencing with the analysis of single split-reads covering small SVs. Meanwhile, although there exist algorithms to detect large SVs with paired-end reads [40], the complete reconstruction of a large SV requires the integration of reads spanning a wide region, often involving misleading reads from other locations of the genome. If there were no repeats or duplications in the human genome, the reconstruction of such large SVs would be trivially accomplished by the de novo assembly with a high coverage of inexpensive short reads around these regions. With the existence of repeats and duplications in the human genome, however, a set of longer reads will be required to accurately locate some of these SVs in repetitive regions, and a hybrid re-sequencing strategy with both comparative and de novo approaches will be necessary to identify genomic rearrangement events such as deletions and translocations, and also to reconstruct large novel insertions in individuals. Such steps are thus much harder than the others, and will be the main focus of this paper.

Here we present a toolbox and some representative case studies on how to optimally combine the different experimental technologies in the individual genome re-sequencing project, especially in reconstructing large SVs, so as to achieve accurate and economical sequencing [18]. An ‘optimal experimental design should be an intelligent combination of the long, medium, and short sequencing technologies and also some array technologies such as CGH. Some of the previous genome sequencing projects [12, 25] have already incorporated such hybrid approaches using both long and medium reads, although the general problem

Table 4.1. Characteristics of different sequencing/array technologies in comparative individual genome sequencing

		Long Sequencing	**Medium Sequencing**	**Short Sequencing**	**CGH array**
Read length (bases)		~ 800	~ 250	~ 30	Tiling step size: ~ 85bp
Approximate cost per base (\$)		$\sim 1E-3$	$\sim 7E-5$	$\sim 7E-6$	$\sim 3E{-}7$ per array
Error rate per base		$0.001 - 0.002\%$	$0.3 - 0.5\%$	$0.2 - 0.6\%$	N/A (detecting signals rather than sequences)
Major error type		Substitution errors	Insertion / deletion errors (usually caused by homopolymers)	All error types	Array-specific errors (cross-hybridization effects)
Characteristics in comparative individual genome sequencing	**Single reads**	Identify small / medium SVs; localize SVs close to highly represented genomic regions	Identify small SVs; localize SVs in highly represented ~ 100mers	Identify SNPs; localize SNPs in lowly represented genomic regions	Detect large CNVs with relatively low resolution; relatively cheaper than current sequencing technologies
	Paired-end reads	Detect large Indels with relatively low resolution; provide extra information to localize SVs	Detect large Indels with relatively low resolution; provide extra information to localize SVs	Link distant SNPs for haplotype phasing	

of optimal experimental design has not yet been systematically studied. While it is obvious that combining technologies is advantageous, we want to quantitatively show the potential savings based on different integration strategies. Also, since the technologies are constantly developing, it will be useful to have a general and flexible approach to predict the outcome of integrating different technologies, including the new ones coming in the future.

In the following sections, we will first briefly describe a schematic comparative genome re-sequencing framework, focusing on the intrinsically most challenging steps of reconstructing large SVs, and then use a set of semi-realistic simulations of these representative steps to optimize the integrated experimental design. Since full simulations are computationally intractable for such steps in the large parameter space of combinations of different technologies, the simulations are carried out in a framework that can combine the real genomic data with analytical approximations of the sequencing and assembly process. Also, this simulation framework is capable of incorporating new technologies as well as adjusting the parameters for existing ones, and can provide informative guidelines to optimal re-sequencing strategies as the characteristics and cost-structures of such technologies evolve, when combining them becomes a more important concern. The simulation framework is downloadable as a general toolbox to guide optimal re-sequencing as technology constantly advances.

4.2 Results

We first briefly describe in the following subsection a systematic genome assembly strategy for the different types of sequencing reads and array signals, which is an integration of different sequence assembly and tiling array data analysis algorithms. With the most difficult steps in the assembly strategy, i.e. the reconstructions of large SVs, discussed in detail and the performance metric for such large SV reconstruction defined, we then present a semi-realistic sequencing simulation framework, which can guide the optimal experimental design, and show the results of simulations in the reconstruction of two types of large SVs.

4.2.1 Schematic genome assembly strategy

The hybrid genome assembly strategy incorporates both comparative [61] and de novo methods. On one hand, most of the assembly can be done against the reference, and it will be unnecessary to perform a computationally intensive whole genome de novo assembly. Comparative approaches will be capable of identifying small SVs and large rearrangement events. On the other hand, de novo assembly will sometimes still be useful in reconstructing regions with large and novel SVs.

Figure 4.1 shows the schematic steps of SV reconstruction in the context of the genome sequencing/assembly process. The data from different sequencing/array experiments can be processed in the following way: As shown in Figure 4.1A-B, with errors corrected [57] and short reads combined into ‘unipaths [10], all the reads (long/medium/short) from the individuals genome can be mapped back to the reference genome. In Figure 4.1C, the SNPs can then be identified immediately based on the reads with single best matches, and the boundaries of deletions or small insertions will be detected by such reads as well (allowing gaps in alignment). Meanwhile, haplotype islands can also be extracted based on the paired-end information [45, 49, 3] and the prior knowledge of the population haplotype patterns revealed by previous work [31].

Further analysis of the single/paired-end reads are required to reconstruct the large SVs

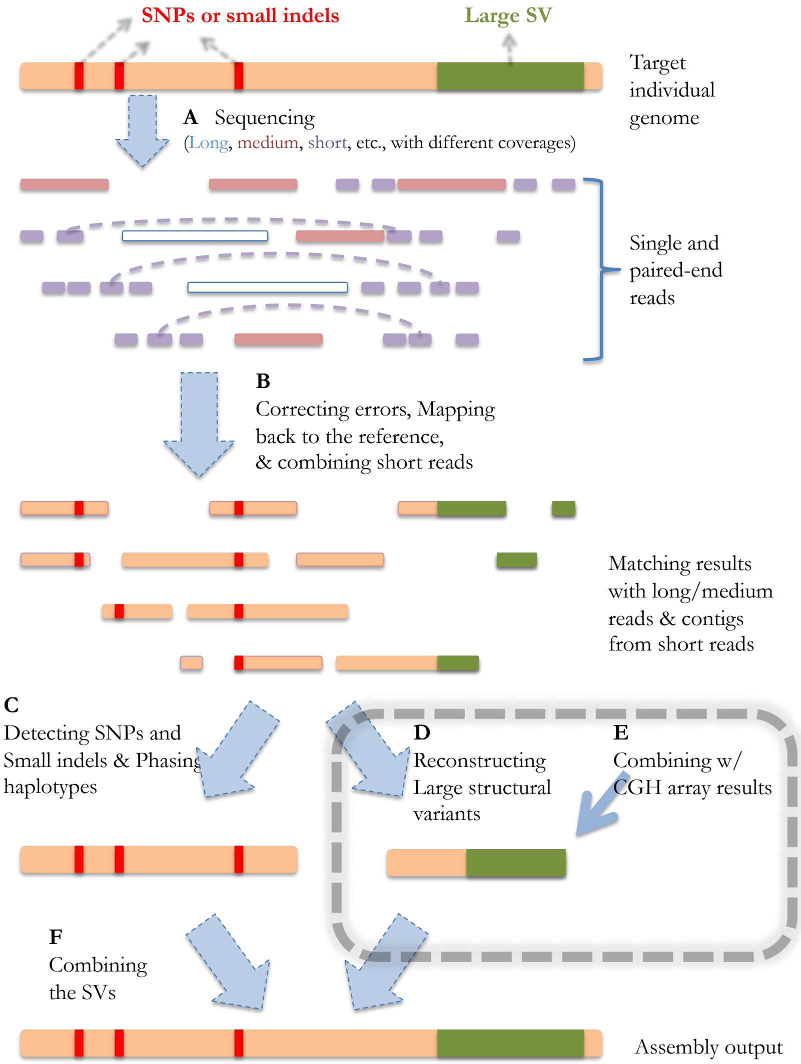

SNPs or small indels
Large SV
Target individual genome
A Sequencing
(Long, medium, short, etc., with different coverages)
Single and paired-end reads
B
Correcting errors, Mapping back to the reference, & combining short reads
Matching results with long/medium reads & contigs from short reads
C
Detecting SNPs and Small indels & Phasing haplotypes
D
Reconstructing Large structural variants
E
Combining w/ CGH array results
F
Combining the SVs
Assembly output

Figure 4.1. Schematic strategy of genome sequencing/assembly.
The orange line represents the target individual genome, the red bars stand for the SNPs and small SVs compared to the reference, and the green region represents a large SV. A) Generated reads can be viewed as various partial observations of the target genome sequence. B) The red and green regions stand for the mismatches/gaps in the mapping results. C) The SNPs and small SVs can be inferred directly from the mapping results, and haplotype phasing can also be performed after this step. D-E) Large SVs can be detected and reconstructed based on the reads without consistent matches in the reference genome. F) The final assembly is generated after all the small and large SVs are identified.

(Figure 4.1D-E), which are by nature more complicated than identifying small SVs. First of all, locations of such SV events need to be detected by analyzing the split-reads (shown in Figure 4.2A-B) that cover their boundaries. Second, two distinct types of SVs need to be handled separately: de novo assembly is required to reconstruct large novel insertions, and comparative algorithms should be utilized to identify genomic rearrangement events (e.g. segmental duplication/deletion). The homozygosity/heterozygosity of such SVs can be determined based on the existence of the reads that map back to the corresponding reference sequences.

Figure 4.2A-C show the overall process of de novo assembly for large novel insertions. While the reconstruction of such regions mostly depends on the spanning-reads from the new inserted sequence, misleading-reads from elsewhere in the genome can often hinder the full reconstruction process. These reads usually comes from the highly represented regions in the genome, which also exist in the insertion. In such cases, reads longer than such regions and appropriate assembly strategies are needed to ensure the unambiguous and correct assembly output. Paired-end reads with an appropriate gap size can also help the unambiguous mapping of the reads inside novel insertions [40].

4.2.2 Defining a performance metric for large SV reconstructions

It is important for us to define a reasonable performance metric so that the re-sequencing approach can be designed in such a way that its outcome will be optimized according to that metric. For large SVs, the metric can be defined based on the alignment result of the actual variant sequence and the inferred variant sequence. For a large SV due to genomic rearrangements (e.g. deletion, duplication), it is natural to define its recovery rate as either 1 (detected) or 0 (missed). For a large novel insertion, on the other hand, we may want

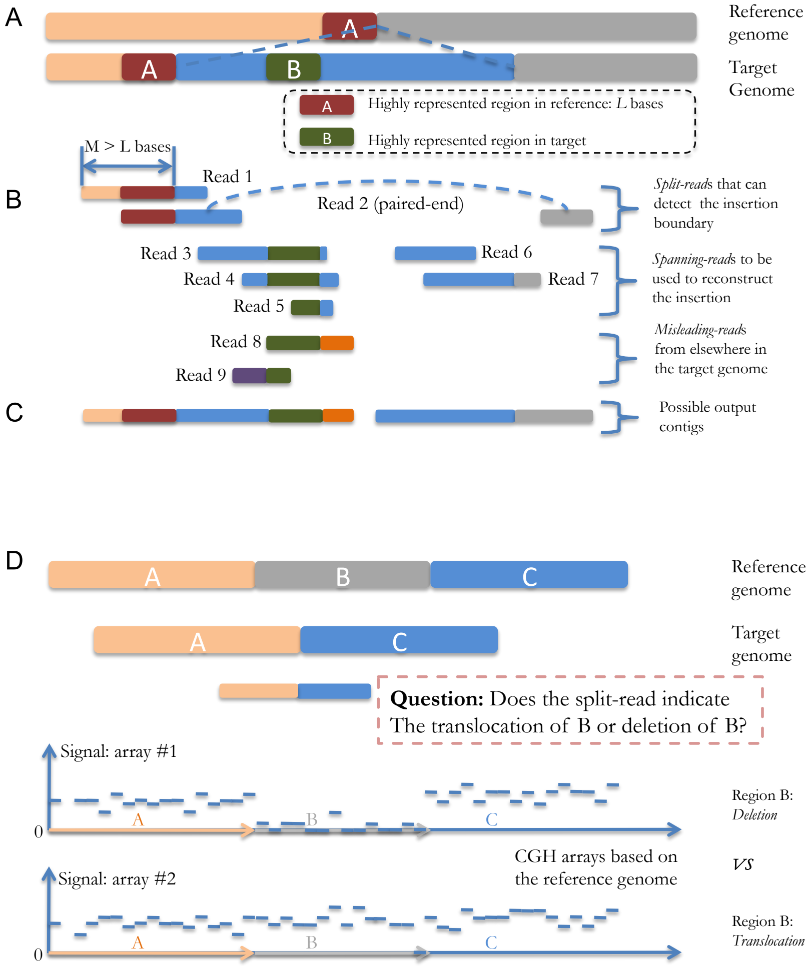

A
A
Reference genome
A
B
Target Genome
A Highly represented region in reference: L bases
B Highly represented region in target
M > L bases
B
Read 1
Read 2 (paired-end)
Split-reads that can detect the insertion boundary
Read 3
Read 4
Read 5
Read 6
Read 7
Spanning-reads to be used to reconstruct the insertion
Read 8
Read 9
Misleading-reads from elsewhere in the target genome
C
Possible output contigs
D
A
B
C
Reference genome
A
C
Target genome
Question: Does the split-read indicate The translocation of B or deletion of B?
Signal: array #1
0
A
B
C
Region B: *Deletion*
CGH arrays based on the reference genome
VS
Signal: array #2
0
A
B
C
Region B: *Translocation*

Figure 4.2. Schematic of the reconstruction of a novel insertion and rearrangement analysis.

The horizontal positions of the reads indicate the mapping locations, and the colors refer to sequences from different genomic regions. A) The region A (L bases) has multiple copies in the reference genome, and the region B has multiple copies in the target genome. The novel sequence is inserted right after a copy of region A and contains a copy of region B. B) Split-reads such as read 1 or 2 will be needed to detect the left boundary of the insertion. ; spanning-reads 3-7 are the reads from the novel insertion region; misleading-reads 8-9 are the reads from elsewhere in the target genome containing the same sequence contents of region B. C) A possible set of resulting contigs after the reconstruction process. The gap is due to the false extension of the first contig caused by the misleading read 8. D) An example of rearrangement analysis.

to take into account cases where the insertion is detected but its sequence content is not reconstructed with full accuracy. Hence, we define the recovery rate of such a large novel insertion as follows based on its reconstruction percentage:

$$RecontructionRate_{insertion} = 1 - \frac{mismatch(wflanking(SV), wflanking(SV_{inf}))}{size(SV)} \tag{4.1}$$

, in which SV is the actual insertion (in simulations, it is already known; in reality, it will need to be identified in a validation step), SV_{inf} is the insertion sequence inferred by the genome re-sequencing approach, $mismatch$ returns the number of mismatches of two aligned sequences, $wflanking$ returns a sequence with its flanking sequences on both ends, and $size$ returns the size of a sequence. The purpose of introducing flanking sequences is to take into account the accuracy of the predicted location of the SV.

4.2.3 Simulations of genome re-sequencing for optimal experimental design

Based on the schematic assembly strategy and the performance measure defined in the previous sections, we can simulate the sequence assembly process in order to obtain an optimal set of parameters for the design of the sequencing experiments (e.g. the amount of long (Sanger), medium (454) and short (Illumina) reads, the amount of single and paired-end reads) and the array experiments (e.g. the incorporation of CGH arrays) to achieve the desired performance with a relatively low cost in the individual genome re-sequencing project.

Here we present the results of a set of simulation case studies on reconstructing large SVs, which are in general much more challenging problems compared to the detection of small SVs. In order to fully reconstruct a long novel insertion, for instance, one needs to not only detect the insertion boundaries based on the split-reads, but also assemble the insertion sequence from the spanning- and misleading-reads. For the identification of genomic rearrangements such as deletion/translocations, one may also want to incorporate array data to increase the confidence level of such analysis. The simulations described in this section are based on large ($\sim$ 10kb, $\sim$ 5Kb and $\sim$ 2Kb) novel insertions and deletions discovered [45], and they perform semi-realistic whole genome assembly representative using the sequence characteristics of both the NCBI reference genome [32] and the target HuRef genome [45]. The sequencing/array technologies considered in these simulations are long, medium and short sequencing methods and CGH arrays, as shown in Table 4.1. Paired-end reads are also included in these simulations.

One major challenge in implementing these simulations is to design them in a computationally realistic way. Brute-force full simulations of whole-genome assembly in this case would be unrealistic: thousands of possible combinations of different technologies will need to be tested, and for each of these combinations hundreds of genome assembly simulations need to be carried out to obtain the statistical distributions of their performance. Since a full simulation of one round of whole-genome assembly will probably take hundreds of CPU hours to finish, the full simulation to explore the full space of technology combinations will then require hundreds of millions ($\sim$ 108) of CPU hours, equivalent to $\sim$ 10 years with 1000 CPUs. We designed the simulations using analytical approximations of the whole-genome assembly process in order for them to be both time and space efficient, and the gain in efficiency is summarized in Table 4.2 and will be described in details later in the Materials

Table 4.2. Time and space complexity of different simulation strategies on the reconstruction of a large novel insertion

Variable	**Description**	**Representative value**
G	Size of the genome	$3E9$bp
c	Sequencing coverage	10x
I	Size of the large novel insertion of interest	$1E4$bp
r	Average read length	50bp
m	Average mapability values of the sub-sequences in the novel insertion	3
Simulation strategy	**Number of reads generated for the reconstruction of a novel insertion**	**Time to compute read overlaps**
Whole genome sequencing + hybrid (comparative + de novo) assembly	$O(G \times c/r)$ (Need to first generate all the reads from the whole genome and then perform selection)	$O((Icm)2)$ (can be improved by hashing the k-mers in the reads)
Simulation utilizing pre-computed mapability maps	$O(Icm/r)$ (simulating the reads based on the insertion region and the mapability maps)	$O(Icm/r)$ (loss of accuracy due to the simulated misleading reads)
Approximate reduction in complexity (fold)	$\sim 1E5$	$\sim 1.5E7$

and Methods section. We have also made this simulation framework publicly available as a toolbox that can incorporate technology advancements as well as other SV regions.

4.2.3.1 Case study: large novel insertion reconstruction with shotgun reads of different lengths

Figure 4.3 show the simulation results of the reconstruction of a large (10Kb) novel insertion in the target individuals genome. Bear in mind that the numbers obtained are dependent on specific parameter settings of the sequencing technologies, which are summarized in Table 4.1. Since these technologies are evolving very rapidly (with new advancements coming out every month), these settings do not represent the current state of the art in these technologies, but are sufficient for the purpose to illustrate how our simulation approach can be used in experiment design and in combining technologies. Also, we are focusing on the full reconstruction of large novel insertions, which would in general require a higher sequencing coverage, thus a higher cost than the detection of small SVs or discovering SNPs. In these figures, the performance measures are obtained by using different combinations of long, medium and short single sequencings reads with a total cost of $\sim$ \$7 on this novel insertion (i.e. the reads covering this region cost $\sim$ \$7). The total re-sequencing budget is $\sim$ \$2.1M if we scale the cost on this region to the whole genome with the same sequencing depth. Please note again that this \$2.1M is for illustrative purposes and does not represent the practical current "street price". The results show that the actual performance, both average and worst-case, is heavily dependent on the coverage combination of the different technologies. The optimal performance (both average and worst-case) of sequencing/assembly is achievable when the long reads have $\sim$ 0.05x coverage, medium reads have $\sim$ 7x coverage, and short reads have $\sim$ 12x coverage (as Figure 4.3C shows, the worst-case performance will decrease, i.e. the color becomes lighter, around the optimal point). A different set of simulations (results not shown) with a total budget of $\sim$ \$600K indicate that the full reconstruction of this SV is still achievable in the optimal configuration, with an average reconstruction rate of $\sim$ 0.61.

Our simulation here is focusing on the reconstruction of large novel SVs, and thus depending on the actual characteristics of different sequencing technologies, the optimal combination of these technologies obtained in this simulation may have a trade-off in the accuracy of detecting SNPs and small indels, i.e., the optimal mixed sequencing strategy for the reconstruction of large novel SVs could lead to a low detection rate of smaller SV events. In this particular example, however, our optimal combination would also guarantee a high recovery rate of SNPs and small indels in the genome, according to the results of an individual genome re-sequencing project described in [81], where $\sim$ 7.4x medium reads were used to detect 3.3 million SNPs and 0.22 million indels. That is, if we focus on the optimal output of large novel SV reconstruction when designing a mixed sequencing strategy, this strategy will give us satisfying result in SNP and indel detection as well. It is also worth noting that the long reads are statistically still useful in these simulations. In general, the long reads are useful in two ways: 1) Long split-reads spanning the insertion boundary have a better chance of being correctly mapped back to the reference, thus detecting the insertion. 2) Long spanning-reads will be especially useful during novel insertion reconstruction when they cover highly repetitive regions that are longer than single medium/short reads. Figure 4.4 shows some typical worst-case simulation results with and without low-coverage long reads using a same total budget. In these examples, mis-assembly around highly repetitive regions is more likely to take place without the long reads.

4.2.3.2 Case study: large novel insertion reconstruction with shotgun and paired-end reads

Similarly to Figure 4.3, Figure 4.5 shows the simulation results on the same insertion as well as a $\sim$ 5Kb and a $\sim$ 2Kb novel insertion using a combination of single and paired-end reads (medium paired-end reads with 3Kb inserts) with a total budget of $\sim$ \$600K (corresponding to $\sim$ \$2 on the 10Kb novel insertion, $\sim$ \$1 on the 5Kb insertion, and $\sim$ \$0.4 on the 2Kb insertion). The optimal performance in reconstruction the $\sim$ 10Kb insertion, in

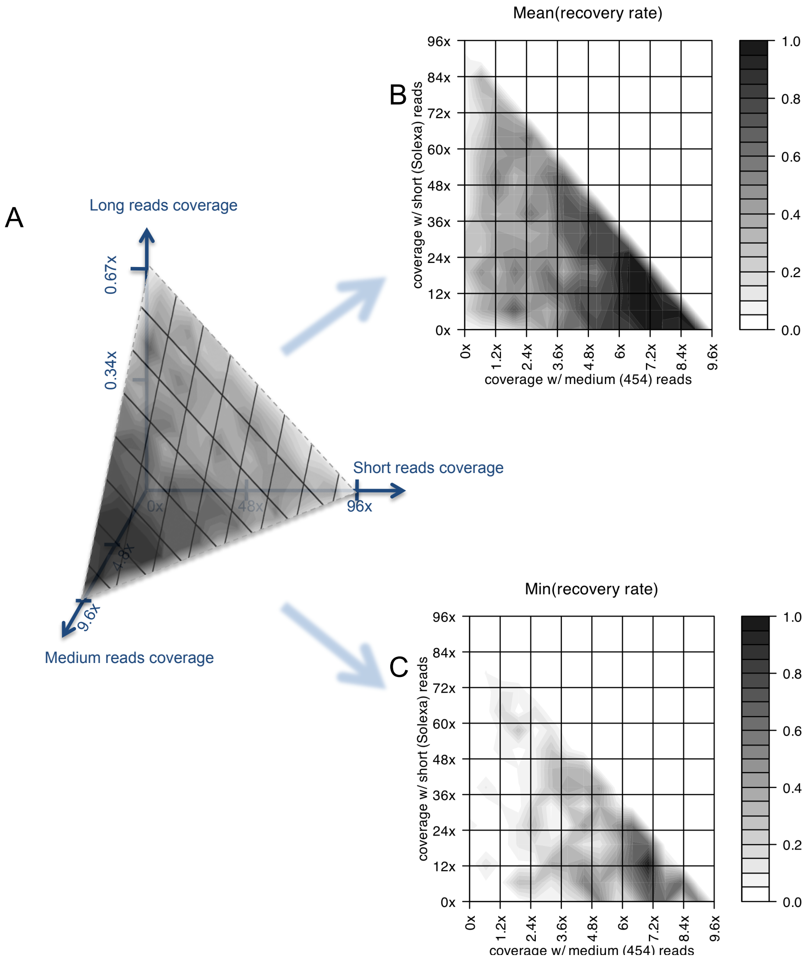


Figure 4.3. Simulation results on the reconstruction of a large novel insertion. The simulation results of the recovery rates of novel insertions when we combine long, medium and short sequencing technologies with a fixed total cost and reconstruct a ∼ 10Kb novel insertion region previously identified in the HuRef genome compared to the NCBI reference genome. The total cost is ∼ \$7 on this novel insertion (i.e. the reads covering this region cost ∼ \$7), and the total re-sequencing budget is ∼ \$2.1M if we scale the cost on this region to the whole genome with the same sequencing depth. A) The triangle plane corresponds to all the sequencing combinations whose total costs are fixed. The colors on the plane indicate the average recovery rates of the novel insertion with different sequencing combinations, averaged over multiple trials of simulations. B) The same triangle region as in Figure 4.3A, projected to the 2D space with two axes representing the coverage of medium and short reads. The coverage of long reads is not explicitly shown and changes with the values of the two other two, forming a same fixed total cost as in A. C) The same type of figure as in Figure 4.3A, showing the worst-case recovery rates on the insertion region with a fixed total sequencing cost.

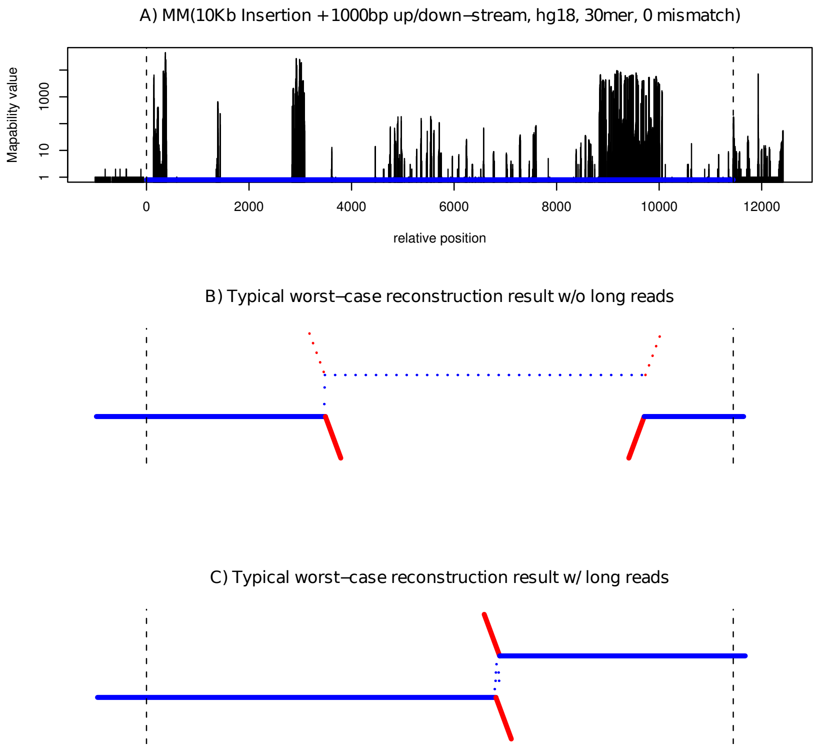


Figure 4.4. MM **values and worst case reconstruction examples of a 10Kb novel insertion.** A) Mapability values for all the 30mers of a $\sim$ 10Kb novel insertion (Variant ID in Huref: 1104685256488, with 1000 flanking sequences): $MM(flanking_{1000bp}(Ins), G_{hg18}, 30, 0)$. The insertion region is shown in blue. B) and C) show the simulation results in reconstructing this region with a same total budget of $\sim$ \$7. The solid blue lines are the assembled contigs that can be localized back to this insertion, with solid red lines for the parts that do not match due to mis-assembly. The dotted blue lines are the contigs that cannot be localized back to this insertion, with the dotted red lines representing the parts that do not match. B) Typical worst-case reconstruction result with $\sim$ 0x long reads, $\sim$ 7x medium reads, and $\sim$ 17.5x short reads. C) Typical worst-case reconstruction result with $\sim$ 0.05x long reads, $\sim$ 7x medium reads, and $\sim$ 10x short reads.

this case, is achieved when medium paired-end reads have $\sim$ 2.4x coverage, medium reads have $\sim$ 0.24x coverage, and short read have $\sim$ 2.4x coverage, with an average reconstruction rate of $\sim$ 0.8, which is significantly better than the results using the single reads only with the same total budget. The reconstructions on the $\sim$ 5Kb and $\sim$ 2Kb insertions also reach their optimal performance with a similar configuration, although their overall mean and worst-case performance differ from each other, due to the different sizes and sequence characteristics of these large novel insertions.

4.2.3.3 Case study: large novel insertion reconstruction with paired-end reads using different insert sizes

We also carried out simulations on reconstructing these novel insertion regions ($\sim$ 10Kb, $\sim$ 5Kb, $\sim$ 2Kb) using paired-end reads with different insert sizes (10Kb and 3Kb inserts for medium paired-end reads, and 150b insert for short paired-end reads). Figure 4.6 shows the simulation results using different combinations of these technologies. In general, the results indicate that a low sequencing coverage of medium paired-end reads (which takes up a large fraction of the total budget due to its relatively high per-base cost) with large inserts (10Kb in this case) and a high coverage of short paired-end reads with small inserts would be optimal for the best reconstruction performance of such novel insertions.

4.2.3.4 Case study on CNV analysis

The second simulation focuses on the identification of genomic rearrangement events, such as deletions and translocations. CNV analysis can be used for this purpose and in this section we simulate its results based on the read-depth and signal intensity analysis of

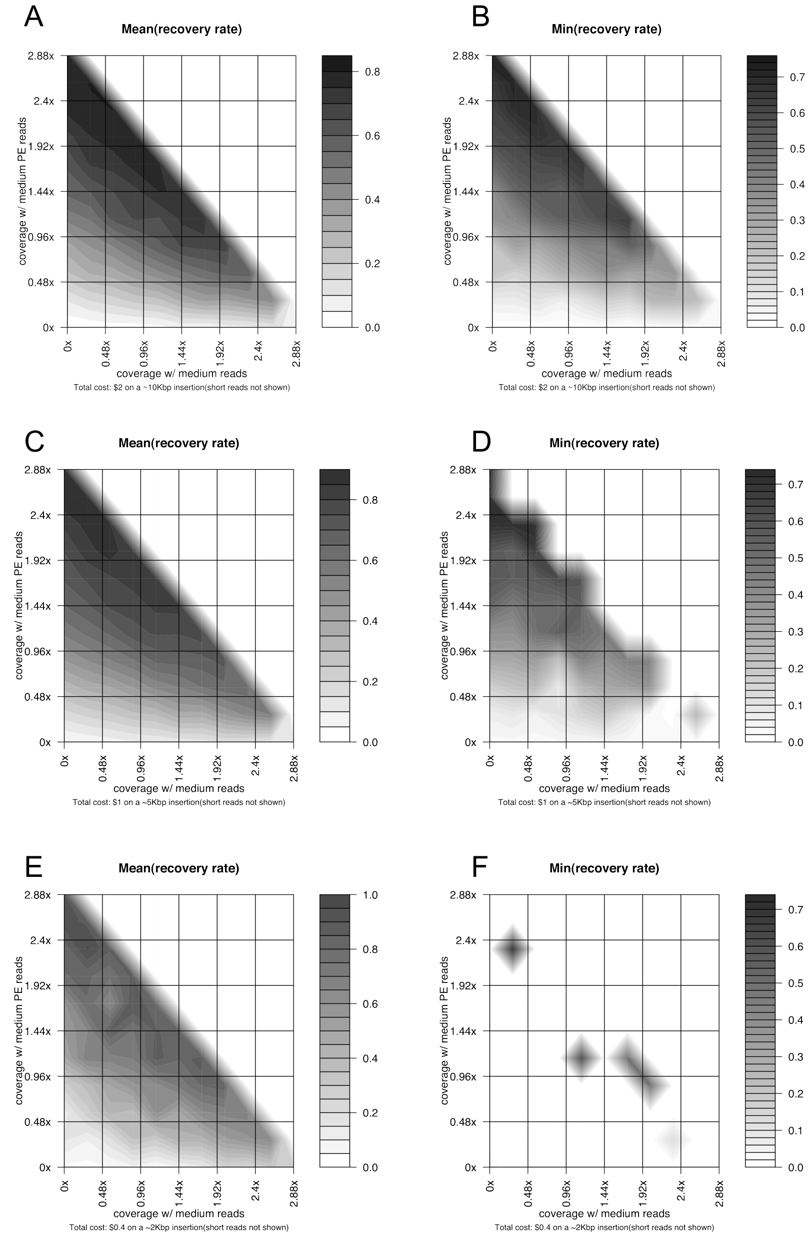


Figure 4.5. Simulation results on the reconstruction of large novel insertions using paired-end reads.

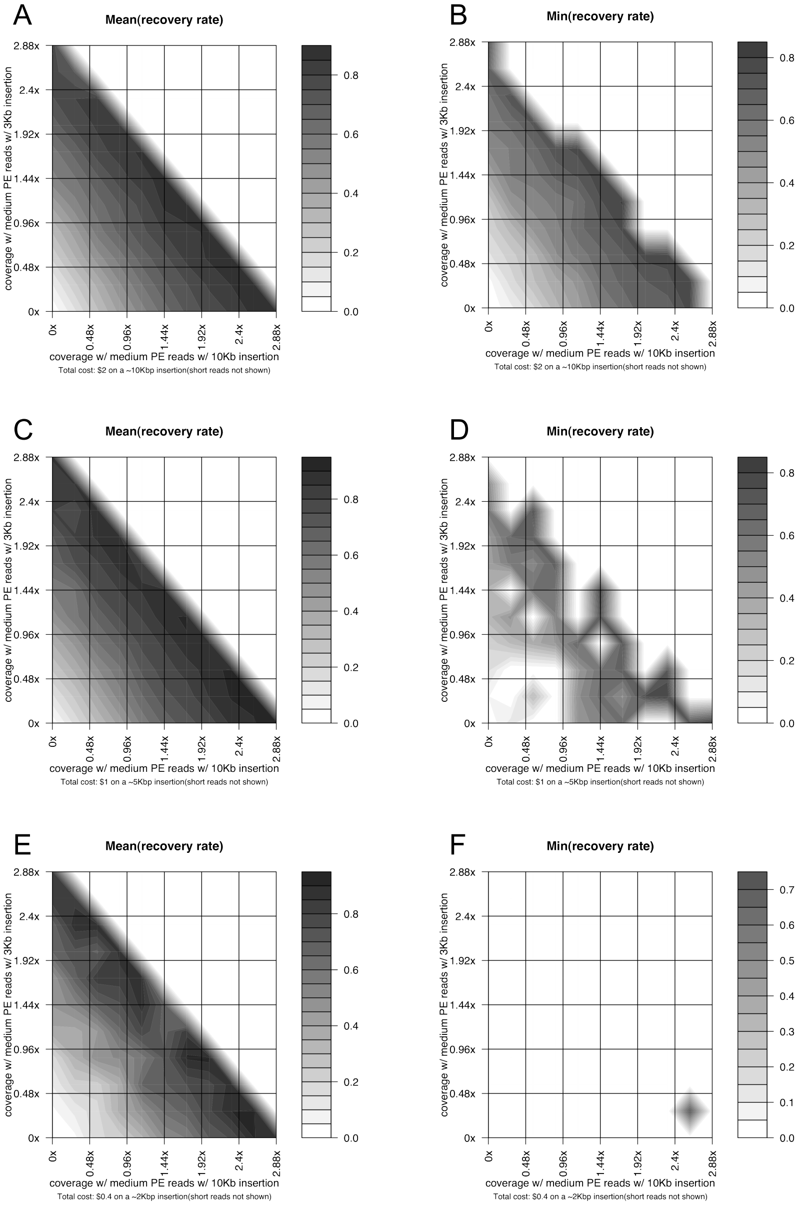


Figure 4.6. Simulation results on the reconstruction of large novel insertions using paired-end reads with different insert sizes.

sequencing and CGH array data. Figure 4.7 shows the simulation results of such analysis on a large (~ 18Kb) deletion in the target individuals genome. The analysis is based on simulated short sequence reads at different coverage, and also on simulated CGH array data with different noise levels. The log-ratio of the posterior probabilities of the deletion (as opposed to translocation) event is computed for each dataset, and used as an indicator of the confidence in determining the deletion event based on that dataset. As shown in the boxplot in Fig. 6, the confidence offered by the CGH arrays is comparable to that offered by the sequencing data with ~ 16x coverage. While ~ 16x coverage of short read sequencing costs ~ $0.3M, using CGH data in this case has the advantage of achieving satisfying performance (as shown in this simulation and [45, 41] in a much more inexpensive way (~ $1000 per array).

4.2.4 Implementation and Availability

In order to be adaptive to the fast development of the experimental technologies in personal genomics, our simulation framework is modularized in such a way that it is capable of incorporating new technologies as well as adjusting the parameters for the existing ones. Also, this approach relies on the general concept of mapability data, and can be easily applied to any representative SV for similar analysis. We envision that in the future, more experimental technologies can be incorporated into this sequencing/assembly simulation and the results of such simulations can provide informative guidelines for the actual experimental design to achieve optimal assembly performance at relatively low costs. With this purpose, we have made this simulation framework downloadable at http://archive.gersteinlab.org/proj/ReSeqSim/ as a general toolbox that can be either used directly or extended easily.

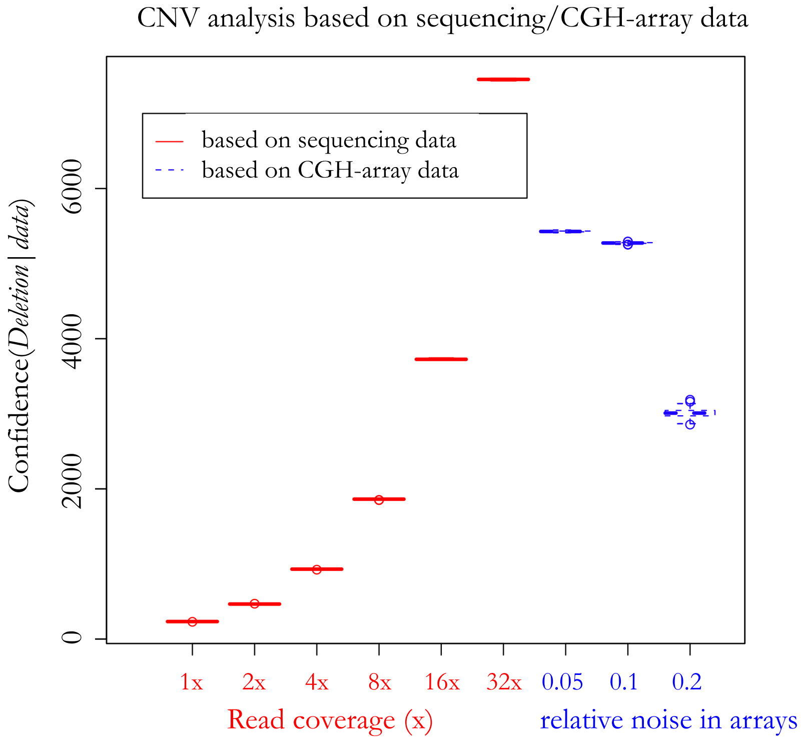


Figure 4.7. Simulation results on rearrangement and CNV analysis.
Boxplot of the CNV analysis simulation results of a large (~ 18Kb) deletion in the target individuals genome. The values on the x-axis correspond to different sequencing coverage and relative noise level in the CGH arrays. The value on the y-axis indicates the confidence of using different datasets to determine that a deletion event takes place instead of a translocation event.

4.3 Methods

4.3.1 The data and parameters used in the simulation

The NCBI assembly v36 [32] and the HuRef assembly [45] were used as reference and target genomes, respectively. Three sequencing technologies, long (Sanger), medium (454), and short (Illumina) sequencing, were considered with the characteristics summarized in Table 4.1. We also assumed that the per-base sequencing error rate increases linearly from the start to the end of a read similar to ReadSim [73], and assigned error types (insertion, deletion or substitution) randomly according to the characteristics of the sequencing technique used [51, 5, 73]. The novel SVs used in the novel insertion reconstruction simulation are $\sim$ 10Kb, $\sim$ 5Kb and $\sim$ 2Kb insertion sequences in the HuRef genome [45] with variant IDs 1104685256488, 1104685222085 and 1104685613186, respectively. The deletion used in the CNV analysis simulation is a $\sim$ 18Kb sequence in the HuRef genome with variant ID 1104685125828.

4.3.2 The simulation of the sequencing/assembly of large novel insertions

Since we would be testing thousands of possible combinations of the long, medium and short sequencing technologies, it would be unrealistic (both time and space consuming) to generate for each combination all the reads from the whole target genome and then apply any existing assembler to these reads. We decided to semi-realistically simulate the assembly process of large novel insertions to achieve relatively accurate estimates in an affordable amount of time. Several difficulties need to be addressed by such a simulation: 1) One of the most time-consuming step in a real assembler is the read overlap-layout step.

2) The whole-genome sequencing experiment introduces large numbers of misleading reads that are partially similar to the reads from the targeted genomic region, which would require a huge storage space in a real assembly process.

4.3.2.1 The mapability data

In order to both accelerate the simulation of the overlap-layout step and simulate the whole-genome sequencing setting in a space-efficient manner, we pre-computed the mapability [69] values of all the possible sub-sequences in the reads from the inserted region. The mapability value of a sequence is the number of times this particular sequence (allowing the specified number of mismatches) appears in a genome, defined below:

Definition 13 (Mapability)**.** For a given genome G and a given sequence s, the mapability function $M(s, G, m)$ is defined as the total number of occurrences of the elements in S in G, where $S = \{s'|mismatch(s, s') \leq m\}$. For simplicity, we let $M(s, G) = M(s, G, 0)$, which is the extract occurrences of s in G.

The following lemmas are obvious:

Lemma 4. *Given a genome G and two sequences s and s', if s contains s', then $M(s, G) \leq M(s', G)$. $M(s, G) = M(s', G)$ if and only if all the occurrences of s' in G are within sequence s. An intuitive interpretation of this lemma is that if a sequence s contains s', then s must occur at most the same number of times as s' in a genome.*

Lemma 5. *Given a genome G, a sequence s, and two non-negative integers m, m', if $m > m'$, then $M(s, G, m) \geq M(s, G, m')$. This lemma states that for any given sequence, its mapability value in a genome never decreases with an increasing mismatch threshold.*

Definition 14 (Mapability Map)**.** For a given genome G and a given sequence s, the k-mapability map $MM(s, G, k, m)$ of s with respect to G is a vector sequentially containing the mapability values of all the k-mers in s with a tolerance of m mismatches: $MM(s, G, k, m) = [M(sub(s, 0, k), G, m), M(sub(s, 1, k + 1), G, m), \cdots]$, where $sub(s, a, b)$ returns the sub-sequence of s from a to $b - 1$ (0-based index). For simplicity, we let $MM(s, G, k) = MM(s, G, k, 0)$, which counts exact occurrence only.

According to the above definition, $MM(s, G, k, m)$ can be viewed as a set of mapability values of all the length-k sub-sequences in s allowing no more than m mismatches.

4.3.2.2 Generation of the split-/spanning-reads and computation of the mapability maps

First, all the reads from the target insertion region are generated (Figure 4.8E) based on the same setting of the long, medium and short sequencing coverages as in the problem being simulated (Figure 4.8A-B). Second, as shown in Figure 4.8D, in order to take into account the effects of the same/similar/misleading-reads from elsewhere in the genome in a whole-genome sequencing experiment, we computed the mapability maps $MM(s, G, k, m)$ of the insertion region s (the $\sim$ 10Kb insertion sequence with its 1Kb up/down-stream flanking sequences), where G = NCBI reference genome, HuRef target genome; $k = 25, 26, \cdots, 800$; $m = 0, 2$. For computational efficiency, the “mismatch” function is currently implemented to take into account only the nucleotide mismatches of two sequences with the same length. On one hand, it would be more realistic to include indel mismatches as well to represent such sequencing errors. On the other hand, we would expect that in practice most of such sequencing errors will be corrected in a preprocessing step [57].

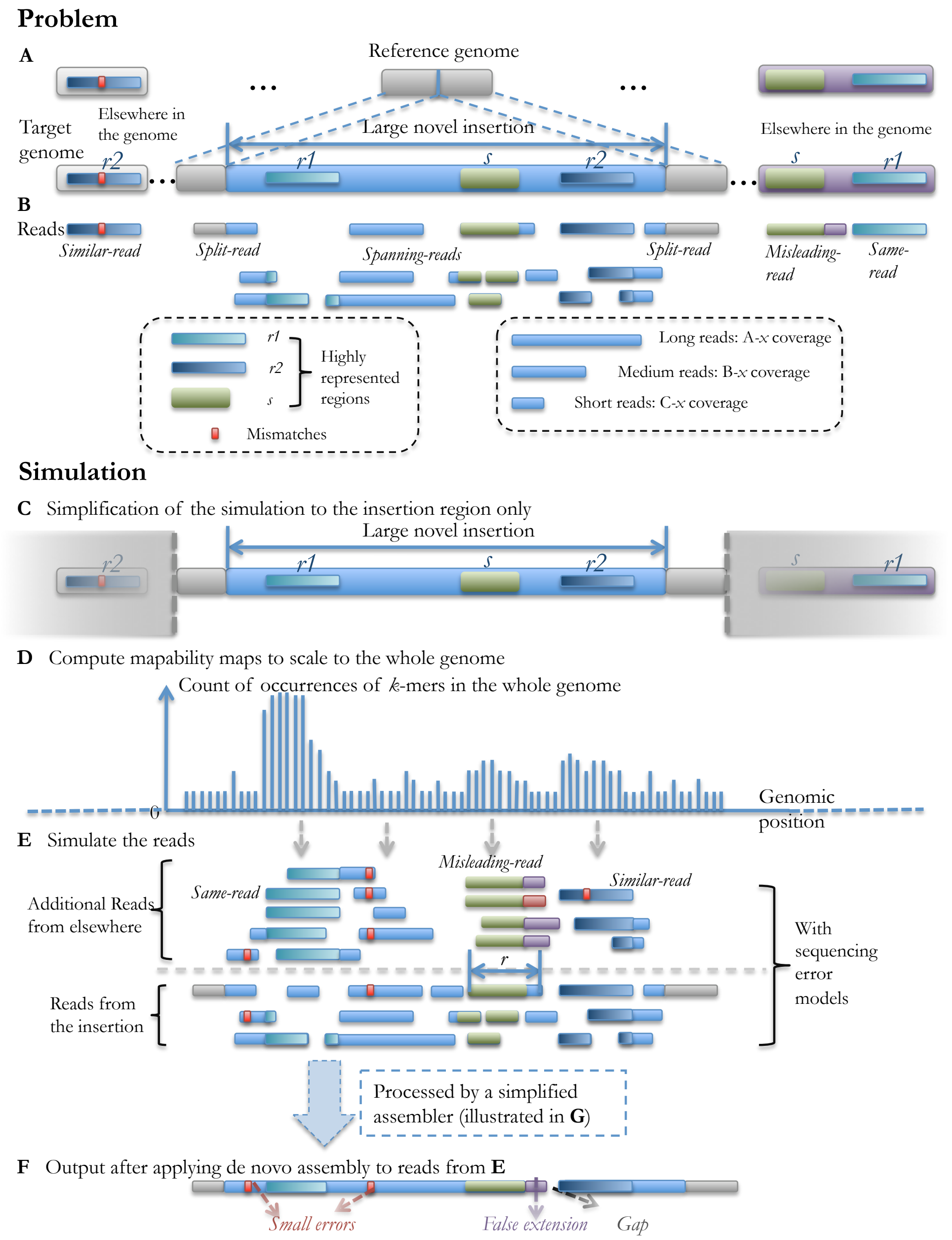

Problem
A
Reference genome
Target genome
Elsewhere in the genome
Large novel insertion
Elsewhere in the genome
r2
r1
s
r2
s
r1
B
Reads
Similar-read
Split-read
Spanning-reads
Split-read
Misleading-read
Same-read
r1
r2
s
Highly represented regions
Mismatches
Long reads: A-x coverage
Medium reads: B-x coverage
Short reads: C-x coverage
Simulation
C Simplification of the simulation to the insertion region only
Large novel insertion
r2
r1
s
r2
s
r1
D Compute mapability maps to scale to the whole genome
Count of occurrences of k-mers in the whole genome
0
Genomic position
E Simulate the reads
Additional Reads from elsewhere
Same-read
Misleading-read
Similar-read
r
Reads from the insertion
With sequencing error models
Processed by a simplified assembler (illustrated in G)
F Output after applying de novo assembly to reads from E
Small errors
False extension
Gap

Figure 4.8. The simulation of novel insertion reconstruction.

The generated reads that align to the same genomic starting locations are grouped together and the per-position error statistics are computed, resulting in a set of read-groups that starts from different locations with their position-specific error statistics computed. These read-groups are then further combined in the de novo reconstruction process describe below.

4.3.2.3 Simulation of same/similar/misleading-reads in de novo reconstruction

Additional reads (same, similar and misleading) are introduced (Figure 4.8E) to simulate the effects of the whole genome sequencing in Figure 4.8A-B. The reads originating from the insertion region and the additional reads are then combined into contigs based on a heuristic read extension algorithm. This is a partial simulation of the overlap-layout-consensus/read-extension/unipath-finding step in the de novo assembly process [16, 10, 77, 4], where the current contig is extended based on the information of the reads that overlap with its end. The extension is only performed when there is either an unambiguous extension supported by all the overlapping reads, or when there is a sufficiently large set of reads with the longest overlap that supports the same extension.

In order to simulate such a process in a whole-genome sequencing setting, the mapability data are again utilized, as illustrated in Figure 4.8D-E. For a highly represented region r in the insertion, its corresponding same/similar reads from elsewhere in the target genome are generated based on the pre-computed $M(r, TargetG, 0)$ and $M(r, TargetG, 2) - M(r, TargetG, 0)$ (the maximal allowed mismatch of 2 corresponds to $\sim 6\%$ or lower difference between two short reads). The number of such reads are randomly generated based on the mapability values of r, the sequencing coverage, and the distribution $P(n, r)$ of the number of reads (n) exactly covering a region with the same size of r, which can be either empirically generated based on the previously simulated reads from the inserted region, or constructed based on a theoretical Poisson distribution representing a uniform sequencing process.

The misleading-reads are generated in the following way: for a contig c and a read r that overlaps it, denote the overlapping sequence with s, then according to Lemma 4, the unambiguous extension of c based on read r is guaranteed if and only if $M(s, TargetG) = M(r, TargetG)$, which means that the sequence s is always within sequence r in the target genome. When $M(s, TargetG) > M(r, TargetG)$, we introduce the misleading reads based on $M(s, TargetG) - M(r, TargetG)$ and $P(n, r)$.

For computational efficiency, we also developed a simplified assembler module to assemble all the generated reads. As illustrated in Figure 4.9, this assembler estimates the overlaps between different reads based on their locations and the corresponding mapability values. It extends a contig by the best overlapping reads with the most supported extension, and simulates the effect of the misleading-reads in the following way: If r is from a paired-end read whose other end r' satisfies $M(r', RefG) = 1$, we assign high confidence to r and always extend with its sequence. Otherwise, if the estimated number of misleading reads are significantly lower than the number reads supporting read r (e.g. by 2-fold), the correct extension is selected. Otherwise, if the misleading reads are over-represented, the misleading extension is chosen and the extended sequence will be different from what is in the actual inserted sequence. The longest common extension supported by all the reads is appended to c if neither type of reads significantly out-numbers the other. The sequencing error statistics at each position are updated accordingly in this procedure.

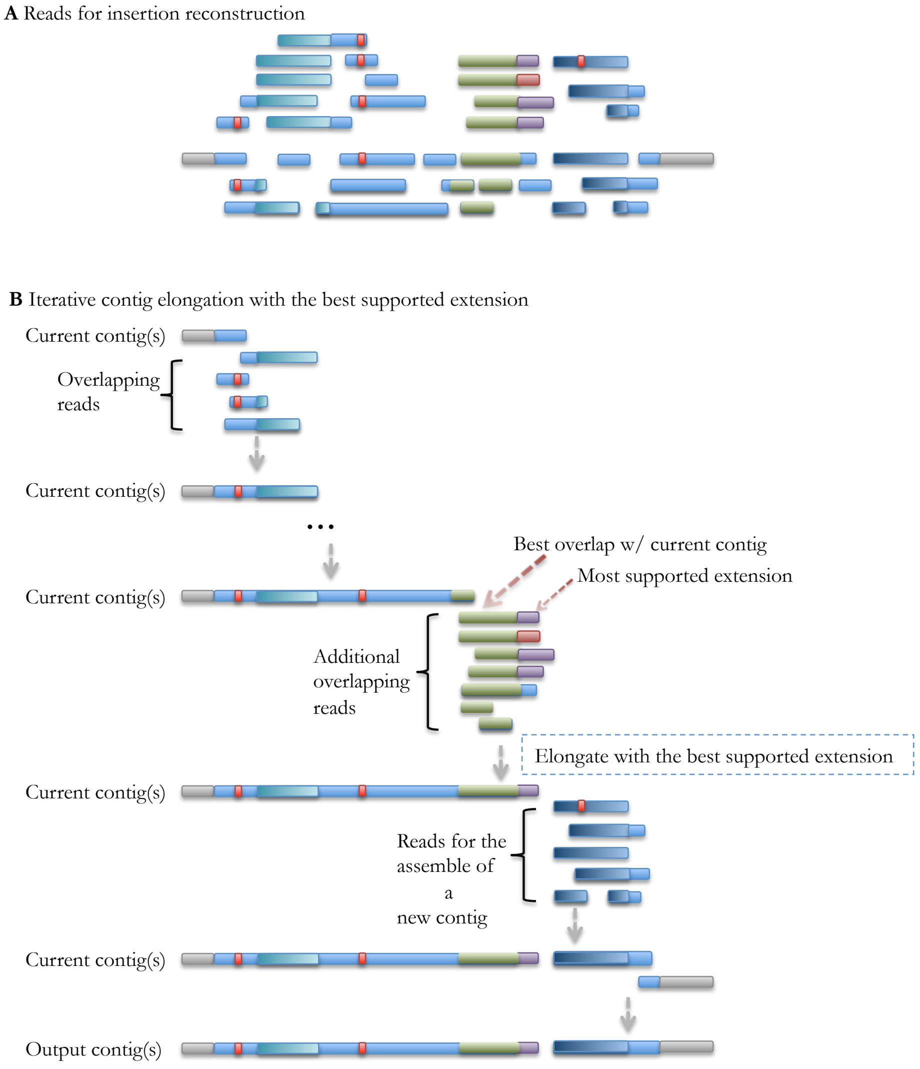

A Reads for insertion reconstruction
B Iterative contig elongation with the best supported extension
Current contig(s)
Overlapping reads
Current contig(s)
...
Best overlap w/ current contig
Most supported extension
Current contig(s)
Additional overlapping reads
Elongate with the best supported extension
Current contig(s)
Reads for the assemble of a new contig
Current contig(s)
Output contig(s)

Figure 4.9. A simplified assembler module to assemble all the generated reads.

4.3.2.4 Computing the reconstruction rate of the combined result

The de novo extensions are performed by the simplified assembler described above from both ends of the insertion region, and the combined results are then compared to the actual insertion to obtain the reconstruction rate of the target region, based on the metric described in the Results section. The flanking sequences are taken into account to measure the accuracy of boundary detection. If the de novo reconstruction result does not cover the insertion boundaries, the reconstructed sequence cannot be localized in the reference genome and the reconstruction rate is set to 0. Figure 4.8F shows example output contigs, which contain small sequencing errors, a false extension error due to the misleading-reads introduced by a highly represented region inside the insertion, and a gap due to both the false extension and the low-coverage of sequencing in that particular region.

4.3.3 The simulation of CNV analysis

In this simulation, we assume that the boundaries of a large deletion event have already been identified by sequence reads, and we are simulating the process of determining whether this is a deletion or translocation event, based on the short reads alone or on the idealized CGH data. The reads are generated in a similar fashion as described in the previous section, without considering sequencing errors for simplicity. The idealized CGH signal of a corresponding region r is defined as Gaussian variable with mean $M(r, TargetG)$, and noise/standard deviation $= 0.05, 0.1, 0.2$. For each dataset, the log-ratio of the posterior probability of the deletion event is computed to represent the confidence level provided by each dataset for determining that deletion. These confidence levels are computed according the following formulas:

$$\begin{aligned}
R_i &= sub(SV, i, i+l) && (4.2)\\
N_{signals} &= |\frac{size(SV)}{l}| && (4.3)\\
C &= \log_{10} \frac{Pr(Deletion)}{Pr(NotDeletion)} && (4.4)\\
Confidence_{seq} &= \log_{10} \frac{Pr(Deletion|reads)}{Pr(NotDeletion|reads)} - C && (4.5)\\
&= \log_{10} Pr(reads|Deletion) - \log_{10} Pr(reads|NotDeletion) && (4.6)\\
&= \sum_{i=1}^{size(SV)} \log_{10} PDF && (4.7)\\
&\quad \left\{Poisson\left((M(R_i, RefG) - 1)\frac{cov_{reads}}{l}\right), obs(R_i)\right\} && (4.8)\\
&\quad - \sum_{i=1}^{size(SV)} \log_{10} PDF && (4.9)\\
&\quad \left\{Poisson\left(M(R_i, RefG)\frac{cov_{reads}}{l}\right), obs(R_i)\right\} && (4.10)\\
Confidence_{array} &= \log_{10} \frac{Pr(Deletion|signals)}{Pr(NotDeletion|signals)} - C && (4.11)\\
&= \sum_{i=1}^{N_{signals}} \log_{10} Pr\left\{sig(R_i l)|Deletion\right\} && (4.12)\\
&\quad - \sum_{i=1}^{N_{signals}} \log_{10} Pr\left\{sig(R_i l)|NotDeletion\right\} && (4.13)\\
&= \sum_{i=1}^{N_{signals}} \log_{10} PDF && (4.14)\\
&\quad \left\{Normal\left(M(R_i, RefG) - 1, noise\right), sig(R_{(i-1)l+1})\right\} && (4.15)\\
&\quad - \sum_{i=1}^{N_{signals}} \log_{10} PDF && (4.16)\\
&\quad \left\{Normal\left(M(R_i, RefG), noise\right), sig(R_{(i-1)l+1})\right\} && (4.17)
\end{aligned}$$

, where $sub(s, a, b)$ returns the sub-sequence of s from a to $b - 1$ (1-based index), l is the length of the short read, SV stands for the deleted region, cov_{reads} is the sequencing coverage, $obs(r)$ is the number of observed reads that are the same as r, $sig(r)$ is the normalized CGH-array signal of probe r, $PDF\{D, v\}$ is the probability density/mass function of the distribution D at value v, and $RefG/TargetG$ refers to the reference/target genome.

4.4 Discussion

The simulation results in the previous sections are based on three sequencing technologies and an idealized array technology, and assume a specific parameterization of their characteristics and costs. Thus, the particular optimal solutions found may not be immediately applicable to a real individual genome re-sequencing project. However, these results illustrate quantitatively how we can design and run simulations to obtain guidelines for optimal experimental design in such projects.

Since our simulation approach is based on the general concept of mapability map and comparative SV reconstruction instead of on a specific organism, it can also be adapted to the comparative sequencing of a non-human genome with regard to a closely related reference. In such a study, we can first construct an artificial target genome based on estimations of its divergence from the reference, and then compute the mapability maps of those representative SVs as input to the simulation framework to find the optimal combination of technologies. Obviously, the closer the two genomes are, the more informative the simulation result would be. In cases where it is hard to estimate the divergence of the target genome from the reference, a two-step approach can be conducted: First, combined sequencing experiments will be carried out using an optimal configuration obtained from the simulation based on the “best guess”, such as another closely related genome. Second, by using the target genome constructed in the previous step, a new set of simulations can be executed and their results can guide a second round of combined sequencing which can provide a finer re-sequencing outcome when combined with the previous sequencing data. Meanwhile, our simulation framework specifically focuses on the effects of misleading reads in the SV reconstruction process, and it will be the most helpful in cases where the target and reference genome both have complex repetitive/duplicative sequence characteristics which will introduce such reads.

In this chapter, we propose to optimally incorporate different experimental technologies in the design of an individual genome-sequencing project, especially for the full reconstruction of large SVs, to achieve accurate output with relatively low costs. We first describe a hybrid genome re-sequencing strategy for detecting SVs in the target genome, and then propose how we can design the optimal combination of experiments for reconstructing large SVs based on the results of semi-realistic simulations with different single and paired-end reads. We also present several examples of such simulations, focusing on the reconstruction of large novel insertions and confirmation of large deletions based on CNV analysis, which are the most challenging steps in individual re-sequencing. The simulations for actual sequencing experimental design can integrate more technologies with different characteristics, and also test the sequencing/assembly performance at different SV levels. By doing so, a set of experiments based on various technologies can be integrated to best achieve the ultimate goal of an individual genome re-sequencing project: accurately detecting all the nucleotide and structural variants in the individuals genome in a cost-efficient way. Such information will ultimately prove beneficial in understanding the genetic basis of phenotypic differences in humans.

Chapter 5

Conclusion

In this thesis, we demonstrate that integrated analysis on partial samples from different sampling techniques can improve the training and parameterization of statistical learning models in bioinformatics, and also present methods and algorithms to efficiently find optimal integration of different sampling techniques to get the best model training outcome with a fixed total budget.

In the study of integrating tiling array and experimental validation data to solve a sequential labeling problem along the genome, we showed how we can use a supervised method to systematically train a hidden Markov model based on these two types of deterministic (in terms of sampling locations on the genome) partial samples, and use to model to provide the corresponding transcriptional state sequence for the genome. We also investigated how to select deterministic samples to be labeled in order to best improve the trained model, and proposed to employ a *MaxEntropy* scheme as a measure for optimal sample selection. Using this scheme, we can identify ahead of time all the locations on the genome to be sampled by the validation experiments, which is particularly desirable for time-consuming validation experiments.

We then studied the scenario where the partial samples are no longer selected from deterministic positions, but are generated randomly by the sampling technique (i.e. various sequencing technologies). We generalized the problem of transcript isoform quantification in RNA-seq experiments to a distribution estimation problem based on a set of different types of partial samples, and presented an expectation maximization based solution to the corresponding maximum likelihood estimation problem. Furthermore, we proposed a Fisher information based heuristic to estimate the performance of our MLE solution, and also introduced the concept of equivalent sample sets to develop a fast algorithm to compute this value efficiently, achieving a speedup of ~ 500 times compared to the brute-force method. We also used both simulated and real data to demonstrate how such a heuristic can be used to find optimal low-cost combinations of sampling techniques as well as to estimate MLE performance.

Last but not least, we investigated the problem of individual genome re-sequencing using sequencing reads as random partial samples, with complicated genome sequence characteristics and sequence assembly algorithms which are unlikely to be accurately modeled analytically, while also computationally intractable for large-scale simulations. We formulated canonical problems that are representative of issues in this process, and introduced the concept of mapability maps to develop a simulation toolbox that can efficiently handle the inhomogeneous repeat-containing structure of the human genome and the computational complexity of practical assembly algorithms. This simulation framework is capable of incorporating new technologies as well as adjusting the parameters for existing ones, and can provide informative guidelines to optimal re-sequencing strategies as the characteristics and cost-structures of such technologies evolve, when combining them becomes a more important concern.

We have identified a number of directions for future research on the integrated analysis of partial samples, especially in the field of bioinformatics:

Incorporation of more accurate partial sampling models. With a better understanding of the sampling techniques based on the study of existing data, more accurate models of the sampling process can be developed. Some of the partial sample integration methods discussed in this thesis are developed with a generalized "pluggable" sampling model, and can incorporate these new sampling models seamlessly. However, as discussed in section 3, the algorithms for computing the performance heuristic will need to be revised to utilize a relaxed definition of equivalent samples. What is more, the realistic modeling of the partial sampling itself is a non-trivial task, and would require extensive data analysis to distinguish general sampling characteristics from individual experiment artifacts.

Incorporation of domain-specific knowledge The partial samples are usually obtained for a certain purpose. For example, the sequencing reads from RNA-seq experiments are obtained because researchers want to use them to study the characteristics of genes and their transcript isoforms. It is important for the corresponding statistical learning model to take the relevant biological knowledge into account, particularly in a quantitative fashion. Such knowledge can sometime be treated as prior knowledge and taken into account by a Bayesian framework.

Integration of different types of sampling methods We have been mostly focusing on the integration of the same type of sampling techniques (e.g. sequencing) with different characteristics. However, as we have briefly mentioned in the case of combining tiling array and experimental validation data in Chapter 2, and also of combining sequencing and CGH array data at the end of Chapter 4, it is sometimes necessary to consider integrating different types of samples to answer certain biological questions, and training models based on such heterogeneous samples is obviously a interesting problem worth investigating.

Bibliography

[1] N. Abe and M.K. Warmuth. On the computational complexity of approximating distributions by probabilistic automata. *Machine Learning*, 9:205–260, 1992.

[2] Sung-Min Ahn, Tae-Hyung Kim, Sunghoon Lee, Deokhoon Kim, Ho Ghang, Dae-Soo Kim, Byoung-Chul Kim, Sang-Yoon Kim, Woo-Yeon Kim, Chulhong Kim, Daeui Park, Yong Seok Lee, Sangsoo Kim, Rohit Reja, Sungwoong Jho, Chang Geun Kim, Ji-Young Cha, Kyung-Hee Kim, Bonghee Lee, Jong Bhak, and Seong-Jin Kim. The first korean genome sequence and analysis: full genome sequencing for a socio-ethnic group. *Genome Res*, 19(9):1622–1629, Sep 2009.

[3] Vikas Bansal, Aaron L Halpern, Nelson Axelrod, and Vineet Bafna. An mcmc algorithm for haplotype assembly from whole-genome sequence data. *Genome Res*, 18(8):1336–1346, Aug 2008.

[4] Serafim Batzoglou, David B Jaffe, Ken Stanley, Jonathan Butler, Sante Gnerre, Evan Mauceli, Bonnie Berger, Jill P Mesirov, and Eric S Lander. Arachne: a whole-genome shotgun assembler. *Genome Res*, 12(1):177–189, Jan 2002.

[5] David R Bentley. Whole-genome re-sequencing. *Curr Opin Genet Dev*, 16(6):545–552, Dec 2006.

[6] David R Bentley, Shankar Balasubramanian, Harold P Swerdlow, Geoffrey P Smith, John Milton, Clive G Brown, Kevin P Hall, Dirk J Evers, Colin L Barnes, Helen R Bignell, Jonathan M Boutell, Jason Bryant, Richard J Carter, R. Keira Cheetham, Anthony J Cox, Darren J Ellis, Michael R Flatbush, Niall A Gormley, Sean J Humphray, Leslie J Irving, Mirian S Karbelashvili, Scott M Kirk, Heng Li, Xiaohai Liu, Klaus S Maisinger, Lisa J Murray, Bojan Obradovic, Tobias Ost, Michael L Parkinson, Mark R Pratt, Isabelle M J Rasolonjatovo, Mark T Reed, Roberto Rigatti, Chiara Rodighiero, Mark T Ross, Andrea Sabot, Subramanian V Sankar, Aylwyn Scally, Gary P Schroth, Mark E Smith, Vincent P Smith, Anastassia Spiridou, Peta E Torrance, Svilen S Tzonev, Eric H Vermaas, Klaudia Walter, Xiaolin Wu, Lu Zhang, Mohammed D Alam, Carole Anastasi, Ify C Aniebo, David M D Bailey, Iain R Bancarz, Saibal Banerjee, Selena G Barbour, Primo A Baybayan, Vincent A Benoit, Kevin F Benson, Claire Bevis, Phillip J Black, Asha Boodhun, Joe S Brennan, John A Bridgham, Rob C Brown, Andrew A Brown, Dale H Buermann, Abass A Bundu, James C Burrows, Nigel P Carter, Nestor Castillo, Maria Chiara E Catenazzi, Simon Chang, R. Neil Cooley, Natasha R Crake, Olubunmi O Dada, Konstantinos D Diakoumakos, Belen Dominguez-Fernandez, David J Earnshaw, Ugonna C Egbujor, David W Elmore, Sergey S Etchin, Mark R Ewan, Milan Fedurco, Louise J Fraser, Karin V Fuentes Fajardo, W. Scott Furey, David George, Kimberley J Gietzen, Colin P Goddard, George S Golda, Philip A Granieri, David E Green, David L Gustafson, Nancy F Hansen, Kevin Harnish, Christian D Haudenschild, Narinder I Heyer, Matthew M Hims, Johnny T Ho, Adrian M Horgan, Katya Hoschler, Steve Hurwitz, Denis V Ivanov, Maria Q Johnson, Terena James, T. A. Huw

Jones, Gyoung-Dong Kang, Tzvetana H Kerelska, Alan D Kersey, Irina Khrebtukova, Alex P Kindwall, Zoya Kingsbury, Paula I Kokko-Gonzales, Anil Kumar, Marc A Laurent, Cynthia T Lawley, Sarah E Lee, Xavier Lee, Arnold K Liao, Jennifer A Loch, Mitch Lok, Shujun Luo, Radhika M Mammen, John W Martin, Patrick G McCauley, Paul McNitt, Parul Mehta, Keith W Moon, Joe W Mullens, Taksina Newington, Zemin Ning, Bee Ling Ng, Sonia M Novo, Michael J O'Neill, Mark A Osborne, Andrew Osnowski, Omead Ostadan, Lambros L Paraschos, Lea Pickering, Andrew C Pike, Alger C Pike, D. Chris Pinkard, Daniel P Pliskin, Joe Podhasky, Victor J Quijano, Come Raczy, Vicki H Rae, Stephen R Rawlings, Ana Chiva Rodriguez, Phyllida M Roe, John Rogers, Maria C Rogert Bacigalupo, Nikolai Romanov, Anthony Romieu, Rithy K Roth, Natalie J Rourke, Silke T Ruediger, Eli Rusman, Raquel M Sanches-Kuiper, Martin R Schenker, Josefina M Seoane, Richard J Shaw, Mitch K Shiver, Steven W Short, Ning L Sizto, Johannes P Sluis, Melanie A Smith, Jean Ernest Sohna Sohna, Eric J Spence, Kim Stevens, Neil Sutton, Lukasz Szajkowski, Carolyn L Tregidgo, Gerardo Turcatti, Stephanie Vandevondele, Yuli Verhovsky, Selene M Virk, Suzanne Wakelin, Gregory C Walcott, Jingwen Wang, Graham J Worsley, Juying Yan, Ling Yau, Mike Zuerlein, Jane Rogers, James C Mullikin, Matthew E Hurles, Nick J McCooke, John S West, Frank L Oaks, Peter L Lundberg, David Klenerman, Richard Durbin, and Anthony J Smith. Accurate whole human genome sequencing using reversible terminator chemistry. *Nature*, 456(7218):53–59, Nov 2008.

[7] S. M. Berget, C. Moore, and P. A. Sharp. Spliced segments at the 5' terminus of adenovirus 2 late mrna. *Proc Natl Acad Sci U S A*, 74(8):3171–3175, Aug 1977.

[8] Paul Bertone, Viktor Stolc, Thomas E Royce, Joel S Rozowsky, Alexander E Urban, Xiaowei Zhu, John L Rinn, Waraporn Tongprasit, Manoj Samanta, Sherman Weiss-

man, Mark Gerstein, and Michael Snyder. Global identification of human transcribed sequences with genome tiling arrays. *Science*, 306(5705):2242–2246, Dec 2004.

[9] Michael J Buck and Jason D Lieb. Chip-chip: considerations for the design, analysis, and application of genome-wide chromatin immunoprecipitation experiments. *Genomics*, 83(3):349–360, Mar 2004.

[10] Jonathan Butler, Iain Maccallum, Michael Kleber, Ilya A Shlyakhter, Matthew K Belmonte, Eric S Lander, Chad Nusbaum, and David B Jaffe. Allpaths: De novo assembly of whole-genome shotgun microreads. *Genome Res*, 18(5):810–820, May 2008.

[11] Simon Cawley, Stefan Bekiranov, Huck H Ng, Philipp Kapranov, Edward A Sekinger, Dione Kampa, Antonio Piccolboni, Victor Sementchenko, Jill Cheng, Alan J Williams, Raymond Wheeler, Brant Wong, Jorg Drenkow, Mark Yamanaka, Sandeep Patel, Shane Brubaker, Hari Tammana, Gregg Helt, Kevin Struhl, and Thomas R Gingeras. Unbiased mapping of transcription factor binding sites along human chromosomes 21 and 22 points to widespread regulation of noncoding rnas. *Cell*, 116(4):499–509, Feb 2004.

[12] Mark J Chaisson and Pavel A Pevzner. Short read fragment assembly of bacterial genomes. *Genome Res*, 18(2):324–330, Feb 2008.

[13] Jill Cheng, Philipp Kapranov, Jorg Drenkow, Sujit Dike, Shane Brubaker, Sandeep Patel, Jeffrey Long, David Stern, Hari Tammana, Gregg Helt, Victor Sementchenko, Antonio Piccolboni, Stefan Bekiranov, Dione K Bailey, Madhavan Ganesh, Srinka Ghosh, Ian Bell, Daniela S Gerhard, and Thomas R Gingeras. Transcriptional maps of 10 human chromosomes at 5-nucleotide resolution. *Science*, 308(5725):1149–1154, May 2005.

[14] L. T. Chow, R. E. Gelinas, T. R. Broker, and R. J. Roberts. An amazing sequence arrangement at the 5' ends of adenovirus 2 messenger rna. *Cell*, 12(1):1–8, Sep 1977.

[15] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

[16] Juliane C Dohm, Claudio Lottaz, Tatiana Borodina, and Heinz Himmelbauer. Sharcgs, a fast and highly accurate short-read assembly algorithm for de novo genomic sequencing. *Genome Res*, 17(11):1697–1706, Nov 2007.

[17] Radoje Drmanac, Andrew B Sparks, Matthew J Callow, Aaron L Halpern, Norman L Burns, Bahram G Kermani, Paolo Carnevali, Igor Nazarenko, Geoffrey B Nilsen, George Yeung, Fredrik Dahl, Andres Fernandez, Bryan Staker, Krishna P Pant, Jonathan Baccash, Adam P Borcherding, Anushka Brownley, Ryan Cedeno, Linsu Chen, Dan Chernikoff, Alex Cheung, Razvan Chirita, Benjamin Curson, Jessica C Ebert, Coleen R Hacker, Robert Hartlage, Brian Hauser, Steve Huang, Yuan Jiang, Vitali Karpinchyk, Mark Koenig, Calvin Kong, Tom Landers, Catherine Le, Jia Liu, Celeste E McBride, Matt Morenzoni, Robert E Morey, Karl Mutch, Helena Perazich, Kimberly Perry, Brock A Peters, Joe Peterson, Charit L Pethiyagoda, Kaliprasad Pothuraju, Claudia Richter, Abraham M Rosenbaum, Shaunak Roy, Jay Shafto, Uladzislau Sharanhovich, Karen W Shannon, Conrad G Sheppy, Michel Sun, Joseph V Thakuria, Anne Tran, Dylan Vu, Alexander Wait Zaranek, Xiaodi Wu, Snezana Drmanac, Arnold R Oliphant, William C Banyai, Bruce Martin, Dennis G Ballinger, George M Church, and Clifford A Reid. Human genome sequencing using unchained base reads on self-assembling dna nanoarrays. *Science*, 327(5961):78–81, Jan 2010.

[18] Jiang Du, Robert D Bjornson, Zhengdong D Zhang, Yong Kong, Michael Snyder, and Mark B Gerstein. Integrating sequencing technologies in personal genomics: optimal low cost reconstruction of structural variants. *PLoS Comput Biol*, 5(7):e1000432, Jul 2009.

[19] Jiang Du, Joel S Rozowsky, Jan O Korbel, Zhengdong D Zhang, Thomas E Royce, Martin H Schultz, Michael Snyder, and Mark Gerstein. A supervised hidden markov model framework for efficiently segmenting tiling array data in transcriptional and chip-chip experiments: systematically incorporating validated biological knowledge. *Bioinformatics*, 22(24):3016–3024, Dec 2006.

[20] S. R. Eddy. Profile hidden markov models. *Bioinformatics*, 14(9):755–763, 1998.

[21] ENCODE Project Consortium. The encode (encyclopedia of dna elements) project. *Science*, 306(5696):636–640, Oct 2004.

[22] R. E. Gelinas and R. J. Roberts. One predominant 5'-undecanucleotide in adenovirus 2 late messenger rnas. *Cell*, 11(3):533–544, Jul 1977.

[23] Mark B Gerstein, Can Bruce, Joel S Rozowsky, Deyou Zheng, Jiang Du, Jan O Korbel, Olof Emanuelsson, Zhengdong D Zhang, Sherman Weissman, and Michael Snyder. What is a gene, post-encode? history and updated definition. *Genome Res*, 17(6):669–681, Jun 2007.

[24] Francis D Gibbons, Markus Proft, Kevin Struhl, and Frederick P Roth. Chipper: discovering transcription-factor targets from chromatin immunoprecipitation microarrays using variance stabilization. *Genome Biol*, 6(11):R96, 2005.

[25] Susanne M D Goldberg, Justin Johnson, Dana Busam, Tamara Feldblyum, Steve Ferriera, Robert Friedman, Aaron Halpern, Hoda Khouri, Saul A Kravitz, Federico M Lauro, Kelvin Li, Yu-Hui Rogers, Robert Strausberg, Granger Sutton, Luke Tallon, Torsten Thomas, Eli Venter, Marvin Frazier, and J. Craig Venter. A sanger/pyrosequencing hybrid approach for the generation of high-quality draft assemblies of marine microbial genomes. *Proc Natl Acad Sci U S A*, 103(30):11240–11245, Jul 2006.

[26] Jennifer Harrow, France Denoeud, Adam Frankish, Alexandre Reymond, Chao-Kung Chen, Jacqueline Chrast, Julien Lagarde, James G R Gilbert, Roy Storey, David Swarbreck, Colette Rossier, Catherine Ucla, Tim Hubbard, Stylianos E Antonarakis, and Roderic Guigo. Gencode: producing a reference annotation for encode. *Genome Biol*, 7 Suppl 1:S4.1–S4.9, 2006.

[27] Steffen Heber, Max Alekseyev, Sing-Hoi Sze, Haixu Tang, and Pavel A Pevzner. Splicing graphs and est assembly problem. *Bioinformatics*, 18 Suppl 1:S181–S188, 2002.

[28] Jayne Y Hehir-Kwa, Michael Egmont-Petersen, Irene M Janssen, Dominique Smeets, Ad Geurts van Kessel, and Joris A Veltman. Genome-wide copy number profiling on high-density bacterial artificial chromosomes, single-nucleotide polymorphisms, and oligonucleotide microarrays: a platform comparison based on statistical power analysis. *DNA Res*, 14(1):1–11, Feb 2007.

[29] Ladeana W Hillier, Valerie Reinke, Philip Green, Martin Hirst, Marco A Marra, and Robert H Waterston. Massively parallel sequencing of the polyadenylated transcriptome of c. elegans. *Genome Res*, 19(4):657–666, Apr 2009.

[30] David C Hoyle, Magnus Rattray, Ray Jupp, and Andrew Brass. Making sense of microarray data distributions. *Bioinformatics*, 18(4):576–584, Apr 2002.

[31] International HapMap Consortium. A haplotype map of the human genome. *Nature*, 437(7063):1299–1320, Oct 2005.

[32] International Human Genome Sequencing Consortium. Finishing the euchromatic sequence of the human genome. *Nature*, 431(7011):931–945, Oct 2004.

[33] V. R. Iyer, C. E. Horak, C. S. Scafe, D. Botstein, M. Snyder, and P. O. Brown. Genomic binding sites of the yeast cell-cycle transcription factors SBF and MBF. *Nature*, 409(6819):533–538, Jan 2001.

[34] Hongkai Ji and Wing Hung Wong. TileMap: create chromosomal map of tiling array hybridizations. *Bioinformatics*, 21(18):3629–3636, Sep 2005.

[35] Hui Jiang and Wing Hung Wong. Statistical inferences for isoform expression in rnaseq. *Bioinformatics*, 25(8):1026–1032, Apr 2009.

[36] Dione Kampa, Jill Cheng, Philipp Kapranov, Mark Yamanaka, Shane Brubaker, Simon Cawley, Jorg Drenkow, Antonio Piccolboni, Stefan Bekiranov, Gregg Helt, Hari Tammana, and Thomas R Gingeras. Novel RNAs identified from an in-depth analysis of the transcriptome of human chromosomes 21 and 22. *Genome Res*, 14(3):331–342, Mar 2004.

[37] Philipp Kapranov, Simon E Cawley, Jorg Drenkow, Stefan Bekiranov, Robert L Strausberg, Stephen P A Fodor, and Thomas R Gingeras. Large-scale transcriptional activity in chromosomes 21 and 22. *Science*, 296(5569):916–919, May 2002.

[38] K. Karplus, C. Barrett, M. Cline, M. Diekhans, L. Grate, and R. Hughey. Predicting protein structure using only sequence information. *Proteins*, Suppl 3:121–125, 1999.

[39] Jong-Il Kim, Young Seok Ju, Hansoo Park, Sheehyun Kim, Seonwook Lee, Jae-Hyuk Yi, Joann Mudge, Neil A Miller, Dongwan Hong, Callum J Bell, Hye-Sun Kim, In-Soon Chung, Woo-Chung Lee, Ji-Sun Lee, Seung-Hyun Seo, Ji-Young Yun, Hyun Nyun Woo, Heewook Lee, Dongwhan Suh, Seungbok Lee, Hyun-Jin Kim, Maryam Yavartanoo, Minhye Kwak, Ying Zheng, Mi Kyeong Lee, Hyunjun Park, Jeong Yeon Kim, Omer Gokcumen, Ryan E Mills, Alexander Wait Zaranek, Joseph Thakuria, Xiaodi Wu, Ryan W Kim, Jim J Huntley, Shujun Luo, Gary P Schroth, Thomas D Wu, HyeRan Kim, Kap-Seok Yang, Woong-Yang Park, Hyungtae Kim, George M Church, Charles Lee, Stephen F Kingsmore, and Jeong-Sun Seo. A highly annotated whole-genome sequence of a korean individual. *Nature*, 460(7258):1011–1015, Aug 2009.

[40] Jan O Korbel, Alexander Eckehart Urban, Jason P Affourtit, Brian Godwin, Fabian Grubert, Jan Fredrik Simons, Philip M Kim, Dean Palejev, Nicholas J Carriero, Lei Du, Bruce E Taillon, Zhoutao Chen, Andrea Tanzer, A. C Eugenia Saunders, Jianxiang Chi, Fengtang Yang, Nigel P Carter, Matthew E Hurles, Sherman M Weissman, Timothy T Harkins, Mark B Gerstein, Michael Egholm, and Michael Snyder. Paired-end mapping reveals extensive structural variation in the human genome. *Science*, 318(5849):420–426, Oct 2007.

[41] Jan O Korbel, Alexander Eckehart Urban, Fabian Grubert, Jiang Du, Thomas E Royce, Peter Starr, Guoneng Zhong, Beverly S Emanuel, Sherman M Weissman, Michael Snyder, and Mark B Gerstein. Systematic prediction and validation of breakpoints associated with copy-number variants in the human genome. *Proc Natl Acad Sci U S A*, 104(24):10110–10115, Jun 2007.

[42] A. Krogh, M. Brown, I. S. Mian, K. Sjlander, and D. Haussler. Hidden Markov models in computational biology. Applications to protein modeling. *J Mol Biol*, 235(5):1501–1531, Feb 1994.

[43] S. Kullback and R.A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22(1):7986, 1951.

[44] Vincent Lacroix, Michael Sammeth, Roderic Guigo, and Anne Bergeron. Exact transcriptome reconstruction from short sequence reads. In *WABI '08: Proceedings of the 8th international workshop on Algorithms in Bioinformatics*, pages 50–63, Berlin, Heidelberg, 2008. Springer-Verlag.

[45] Samuel Levy, Granger Sutton, Pauline C Ng, Lars Feuk, Aaron L Halpern, Brian P Walenz, Nelson Axelrod, Jiaqi Huang, Ewen F Kirkness, Gennady Denisov, Yuan Lin, Jeffrey R MacDonald, Andy Wing Chun Pang, Mary Shago, Timothy B Stockwell, Alexia Tsiamouri, Vineet Bafna, Vikas Bansal, Saul A Kravitz, Dana A Busam, Karen Y Beeson, Tina C McIntosh, Karin A Remington, Josep F Abril, John Gill, Jon Borman, Yu-Hui Rogers, Marvin E Frazier, Stephen W Scherer, Robert L Strausberg, and J. Craig Venter. The diploid genome sequence of an individual human. *PLoS Biol*, 5(10):e254, Sep 2007.

[46] Timothy J Ley, Elaine R Mardis, Li Ding, Bob Fulton, Michael D McLellan, Ken Chen, David Dooling, Brian H Dunford-Shore, Sean McGrath, Matthew Hickenbotham, Lisa Cook, Rachel Abbott, David E Larson, Dan C Koboldt, Craig Pohl, Scott Smith, Amy Hawkins, Scott Abbott, Devin Locke, Ladeana W Hillier, Tracie Miner, Lucinda Fulton, Vincent Magrini, Todd Wylie, Jarret Glasscock, Joshua Conyers, Nathan Sander, Xiaoqi Shi, John R Osborne, Patrick Minx, David Gordon, Asif Chinwalla, Yu Zhao,

Rhonda E Ries, Jacqueline E Payton, Peter Westervelt, Michael H Tomasson, Mark Watson, Jack Baty, Jennifer Ivanovich, Sharon Heath, William D Shannon, Rakesh Nagarajan, Matthew J Walter, Daniel C Link, Timothy A Graubert, John F DiPersio, and Richard K Wilson. Dna sequencing of a cytogenetically normal acute myeloid leukaemia genome. *Nature*, 456(7218):66–72, Nov 2008.

[47] Wei Li, Clifford A Meyer, and X. Shirley Liu. A hidden Markov model for analyzing ChIP-chip experiments on genome tiling arrays and its application to p53 binding sequences. *Bioinformatics*, 21 Suppl 1:i274–i282, Jun 2005.

[48] Marshall E Lieberfarb, Ming Lin, Mirna Lechpammer, Cheng Li, David M Tanenbaum, Phillip G Febbo, Rene L Wright, Judy Shim, Philip W Kantoff, Massimo Loda, Matthew Meyerson, and William R Sellers. Genome-wide loss of heterozygosity analysis from laser capture microdissected prostate cancer using single nucleotide polymorphic allele (SNP) arrays and a novel bioinformatics platform dChipSNP. *Cancer Res*, 63(16):4781–4785, Aug 2003.

[49] Ross Lippert, Russell Schwartz, Giuseppe Lancia, and Sorin Istrail. Algorithmic strategies for the single nucleotide polymorphism haplotype assembly problem. *Brief Bioinform*, 3(1):23–31, Mar 2002.

[50] Elaine R Mardis, Li Ding, David J Dooling, David E Larson, Michael D McLellan, Ken Chen, Daniel C Koboldt, Robert S Fulton, Kim D Delehaunty, Sean D McGrath, Lucinda A Fulton, Devin P Locke, Vincent J Magrini, Rachel M Abbott, Tammi L Vickery, Jerry S Reed, Jody S Robinson, Todd Wylie, Scott M Smith, Lynn Carmichael, James M Eldred, Christopher C Harris, Jason Walker, Joshua B Peck, Feiyu Du, Adam F Dukes, Gabriel E Sanderson, Anthony M Brummett, Eric Clark, Joshua F

McMichael, Rick J Meyer, Jonathan K Schindler, Craig S Pohl, John W Wallis, Xiaoqi Shi, Ling Lin, Heather Schmidt, Yuzhu Tang, Carrie Haipek, Madeline E Wiechert, Jolynda V Ivy, Joelle Kalicki, Glendoria Elliott, Rhonda E Ries, Jacqueline E Payton, Peter Westervelt, Michael H Tomasson, Mark A Watson, Jack Baty, Sharon Heath, William D Shannon, Rakesh Nagarajan, Daniel C Link, Matthew J Walter, Timothy A Graubert, John F DiPersio, Richard K Wilson, and Timothy J Ley. Recurring mutations found by sequencing an acute myeloid leukemia genome. *N Engl J Med*, 361(11):1058–1066, Sep 2009.

[51] Marcel Margulies, Michael Egholm, William E Altman, Said Attiya, Joel S Bader, Lisa A Bemben, Jan Berka, Michael S Braverman, Yi-Ju Chen, Zhoutao Chen, Scott B Dewell, Lei Du, Joseph M Fierro, Xavier V Gomes, Brian C Godwin, Wen He, Scott Helgesen, Chun Heen Ho, Chun He Ho, Gerard P Irzyk, Szilveszter C Jando, Maria L I Alenquer, Thomas P Jarvie, Kshama B Jirage, Jong-Bum Kim, James R Knight, Janna R Lanza, John H Leamon, Steven M Lefkowitz, Ming Lei, Jing Li, Kenton L Lohman, Hong Lu, Vinod B Makhijani, Keith E McDade, Michael P McKenna, Eugene W Myers, Elizabeth Nickerson, John R Nobile, Ramona Plant, Bernard P Puc, Michael T Ronan, George T Roth, Gary J Sarkis, Jan Fredrik Simons, John W Simpson, Maithreyan Srinivasan, Karrie R Tartaro, Alexander Tomasz, Kari A Vogt, Greg A Volkmer, Shally H Wang, Yong Wang, Michael P Weiner, Pengguang Yu, Richard F Begley, and Jonathan M Rothberg. Genome sequencing in microfabricated high-density picolitre reactors. *Nature*, 437(7057):376–380, Sep 2005.

[52] J. C. Marioni, N. P. Thorne, and S. Tavar. BioHMM: a heterogeneous hidden Markov model for segmenting array CGH data. *Bioinformatics*, 22(9):1144–1146, May 2006.

[53] Kevin Judd McKernan, Heather E Peckham, Gina L Costa, Stephen F McLaughlin,

Yutao Fu, Eric F Tsung, Christopher R Clouser, Cisyla Duncan, Jeffrey K Ichikawa, Clarence C Lee, Zheng Zhang, Swati S Ranade, Eileen T Dimalanta, Fiona C Hyland, Tanya D Sokolsky, Lei Zhang, Andrew Sheridan, Haoning Fu, Cynthia L Hendrickson, Bin Li, Lev Kotler, Jeremy R Stuart, Joel A Malek, Jonathan M Manning, Alena A Antipova, Damon S Perez, Michael P Moore, Kathleen C Hayashibara, Michael R Lyons, Robert E Beaudoin, Brittany E Coleman, Michael W Laptewicz, Adam E Sannicandro, Michael D Rhodes, Rajesh K Gottimukkala, Shan Yang, Vineet Bafna, Ali Bashir, Andrew MacBride, Can Alkan, Jeffrey M Kidd, Evan E Eichler, Martin G Reese, Francisco M De La Vega, and Alan P Blanchard. Sequence and structural variation in a human genome uncovered by short-read, massively parallel ligation sequencing using two-base encoding. *Genome Res*, 19(9):1527–1541, Sep 2009.

[54] M.A. Mohamed and P. Gader. Generalized hidden markov models part i: Theoretical frameworks. *IEEE Transcations on Fuzzy Systems*, 8:67–81, 2000.

[55] Emanuel Parzen. On estimation of a probability density function and mode. *Ann. Math. Stat.*, 33:1065–1076, 1962.

[56] George H Perry, Amir Ben-Dor, Anya Tsalenko, Nick Sampas, Laia Rodriguez-Revenga, Charles W Tran, Alicia Scheffer, Israel Steinfeld, Peter Tsang, N. Alice Yamada, Han Soo Park, Jong-Il Kim, Jeong-Sun Seo, Zohar Yakhini, Stephen Laderman, Laurakay Bruhn, and Charles Lee. The fine-scale and complex architecture of human copy-number variation. *Am J Hum Genet*, 82(3):685–695, Mar 2008.

[57] P. A. Pevzner, H. Tang, and M. S. Waterman. An eulerian path approach to dna fragment assembly. *Proc Natl Acad Sci U S A*, 98(17):9748–9753, Aug 2001.

[58] D. Pinkel, R. Segraves, D. Sudar, S. Clark, I. Poole, D. Kowbel, C. Collins, W. L. Kuo, C. Chen, Y. Zhai, S. H. Dairkee, B. M. Ljung, J. W. Gray, and D. G. Albertson. High resolution analysis of dna copy number variation using comparative genomic hybridization to microarrays. *Nat Genet*, 20(2):207–211, Oct 1998.

[59] Erin D Pleasance, R. Keira Cheetham, Philip J Stephens, David J McBride, Sean J Humphray, Chris D Greenman, Ignacio Varela, Meng-Lay Lin, Gonzalo R Ordez, Graham R Bignell, Kai Ye, Julie Alipaz, Markus J Bauer, David Beare, Adam Butler, Richard J Carter, Lina Chen, Anthony J Cox, Sarah Edkins, Paula I Kokko-Gonzales, Niall A Gormley, Russell J Grocock, Christian D Haudenschild, Matthew M Hims, Terena James, Mingming Jia, Zoya Kingsbury, Catherine Leroy, John Marshall, Andrew Menzies, Laura J Mudie, Zemin Ning, Tom Royce, Ole B Schulz-Trieglaff, Anastassia Spiridou, Lucy A Stebbings, Lukasz Szajkowski, Jon Teague, David Williamson, Lynda Chin, Mark T Ross, Peter J Campbell, David R Bentley, P. Andrew Futreal, and Michael R Stratton. A comprehensive catalogue of somatic mutations from a human cancer genome. *Nature*, 463(7278):191–196, Jan 2010.

[60] Erin D Pleasance, Philip J Stephens, Sarah O'Meara, David J McBride, Alison Meynert, David Jones, Meng-Lay Lin, David Beare, King Wai Lau, Chris Greenman, Ignacio Varela, Serena Nik-Zainal, Helen R Davies, Gonzalo R Ordoez, Laura J Mudie, Calli Latimer, Sarah Edkins, Lucy Stebbings, Lina Chen, Mingming Jia, Catherine Leroy, John Marshall, Andrew Menzies, Adam Butler, Jon W Teague, Jonathon Mangion, Yongming A Sun, Stephen F McLaughlin, Heather E Peckham, Eric F Tsung, Gina L Costa, Clarence C Lee, John D Minna, Adi Gazdar, Ewan Birney, Michael D Rhodes, Kevin J McKernan, Michael R Stratton, P. Andrew Futreal, and Peter J Campbell. A small-cell lung cancer genome with complex signatures of tobacco exposure. *Nature*,

463(7278):184–190, Jan 2010.

[61] Mihai Pop, Adam Phillippy, Arthur L Delcher, and Steven L Salzberg. Comparative genome assembly. *Brief Bioinform*, 5(3):237–248, Sep 2004.

[62] Mihai Pop and Steven L Salzberg. Bioinformatics challenges of new sequencing technology. *Trends Genet*, 24(3):142–149, Mar 2008.

[63] Kim D Pruitt, Tatiana Tatusova, and Donna R Maglott. NCBI Reference Sequence (RefSeq): a curated non-redundant sequence database of genomes, transcripts and proteins. *Nucleic Acids Res*, 33(Database issue):D501–D504, Jan 2005.

[64] Dmitry Pushkarev, Norma F Neff, and Stephen R Quake. Single-molecule sequencing of an individual human genome. *Nat Biotechnol*, 27(9):847–852, Sep 2009.

[65] L Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proc. IEEE*, 77:257–286, 1989.

[66] Richard Redon, Shumpei Ishikawa, Karen R Fitch, Lars Feuk, George H Perry, T. Daniel Andrews, Heike Fiegler, Michael H Shapero, Andrew R Carson, Wenwei Chen, Eun Kyung Cho, Stephanie Dallaire, Jennifer L Freeman, Juan R Gonzlez, Mnica Gratacs, Jing Huang, Dimitrios Kalaitzopoulos, Daisuke Komura, Jeffrey R MacDonald, Christian R Marshall, Rui Mei, Lyndal Montgomery, Kunihiro Nishimura, Kohji Okamura, Fan Shen, Martin J Somerville, Joelle Tchinda, Armand Valsesia, Cara Woodwark, Fengtang Yang, Junjun Zhang, Tatiana Zerjal, Jane Zhang, Lluis Armengol, Donald F Conrad, Xavier Estivill, Chris Tyler-Smith, Nigel P Carter, Hiroyuki Aburatani, Charles Lee, Keith W Jones, Stephen W Scherer, and Matthew E Hurles. Global variation in copy number in the human genome. *Nature*, 444(7118):444–454, Nov 2006.

[67] John L Rinn, Ghia Euskirchen, Paul Bertone, Rebecca Martone, Nicholas M Luscombe, Stephen Hartman, Paul M Harrison, F. Kenneth Nelson, Perry Miller, Mark Gerstein, Sherman Weissman, and Michael Snyder. The transcriptional activity of human Chromosome 22. *Genes Dev*, 17(4):529–540, Feb 2003.

[68] Thomas E Royce, Joel S Rozowsky, Paul Bertone, Manoj Samanta, Viktor Stolc, Sherman Weissman, Michael Snyder, and Mark Gerstein. Issues in the analysis of oligonucleotide tiling microarrays for transcript mapping. *Trends Genet*, 21(8):466–475, Aug 2005.

[69] Joel Rozowsky, Ghia Euskirchen, Raymond K Auerbach, Zhengdong D Zhang, Theodore Gibson, Robert Bjornson, Nicholas Carriero, Michael Snyder, and Mark B Gerstein. Peakseq enables systematic scoring of chip-seq experiments relative to controls. *Nat Biotechnol*, 27(1):66–75, Jan 2009.

[70] F. Sanger, S. Nicklen, and A. R. Coulson. Dna sequencing with chain-terminating inhibitors. *Proc Natl Acad Sci U S A*, 74(12):5463–5467, Dec 1977.

[71] Eric E Schadt, Stephen W Edwards, Debraj GuhaThakurta, Dan Holder, Lisa Ying, Vladimir Svetnik, Amy Leonardson, Kyle W Hart, Archie Russell, Guoya Li, Guy Cavet, John Castle, Paul McDonagh, Zhengyan Kan, Ronghua Chen, Andrew Kasarskis, Mihai Margarint, Ramon M Caceres, Jason M Johnson, Christopher D Armour, Philip W Garrett-Engele, Nicholas F Tsinoremas, and Daniel D Shoemaker. A comprehensive transcript index of the human genome generated using microarrays and computational approaches. *Genome Biol*, 5(10):R73, 2004.

[72] Mark J. Schervish. *Theory of Statistics*. Springer, 1995.

[73] R. Schmid, S.C. Schuster, M.A. Steel, and D.H. Huson. Readsim - a simulator for sanger and 454 sequencing. 2006.

[74] Rebecca R Selzer, Todd A Richmond, Nathan J Pofahl, Roland D Green, Peggy S Eis, Prakash Nair, Arthur R Brothman, and Raymond L Stallings. Analysis of chromosome breakpoints in neuroblastoma at sub-kilobase resolution using fine-tiling oligonucleotide array cgh. *Genes Chromosomes Cancer*, 44(3):305–319, Nov 2005.

[75] Alexander Eckehart Urban, Jan O Korbel, Rebecca Selzer, Todd Richmond, April Hacker, George V Popescu, Joseph F Cubells, Roland Green, Beverly S Emanuel, Mark B Gerstein, Sherman M Weissman, and Michael Snyder. High-resolution mapping of dna copy alterations in human chromosome 22 using high-density tiling oligonucleotide arrays. *Proc Natl Acad Sci U S A*, 103(12):4534–4539, Mar 2006.

[76] A. W. Van der Vaart. *Asymptotic Statistics.* Cambridge University Press, 1998.

[77] J. C. Venter, M. D. Adams, E. W. Myers, P. W. Li, R. J. Mural, G. G. Sutton, H. O. Smith, M. Yandell, C. A. Evans, R. A. Holt, J. D. Gocayne, P. Amanatides, R. M. Ballew, D. H. Huson, J. R. Wortman, Q. Zhang, C. D. Kodira, X. H. Zheng, L. Chen, M. Skupski, G. Subramanian, P. D. Thomas, J. Zhang, G. L. Gabor Miklos, C. Nelson, S. Broder, A. G. Clark, J. Nadeau, V. A. McKusick, N. Zinder, A. J. Levine, R. J. Roberts, M. Simon, C. Slayman, M. Hunkapiller, R. Bolanos, A. Delcher, I. Dew, D. Fasulo, M. Flanigan, L. Florea, A. Halpern, S. Hannenhalli, S. Kravitz, S. Levy, C. Mobarry, K. Reinert, K. Remington, J. Abu-Threideh, E. Beasley, K. Biddick, V. Bonazzi, R. Brandon, M. Cargill, I. Chandramouliswaran, R. Charlab, K. Chaturvedi, Z. Deng, V. Di Francesco, P. Dunn, K. Eilbeck, C. Evangelista, A. E. Gabrielian, W. Gan, W. Ge, F. Gong, Z. Gu, P. Guan, T. J. Heiman, M. E. Higgins, R. R. Ji, Z. Ke, K. A.

Ketchum, Z. Lai, Y. Lei, Z. Li, J. Li, Y. Liang, X. Lin, F. Lu, G. V. Merkulov, N. Milshina, H. M. Moore, A. K. Naik, V. A. Narayan, B. Neelam, D. Nusskern, D. B. Rusch, S. Salzberg, W. Shao, B. Shue, J. Sun, Z. Wang, A. Wang, X. Wang, J. Wang, M. Wei, R. Wides, C. Xiao, C. Yan, A. Yao, J. Ye, M. Zhan, W. Zhang, H. Zhang, Q. Zhao, L. Zheng, F. Zhong, W. Zhong, S. Zhu, S. Zhao, D. Gilbert, S. Baumhueter, G. Spier, C. Carter, A. Cravchik, T. Woodage, F. Ali, H. An, A. Awe, D. Baldwin, H. Baden, M. Barnstead, I. Barrow, K. Beeson, D. Busam, A. Carver, A. Center, M. L. Cheng, L. Curry, S. Danaher, L. Davenport, R. Desilets, S. Dietz, K. Dodson, L. Doup, S. Ferriera, N. Garg, A. Gluecksmann, B. Hart, J. Haynes, C. Haynes, C. Heiner, S. Hladun, D. Hostin, J. Houck, T. Howland, C. Ibegwam, J. Johnson, F. Kalush, L. Kline, S. Koduru, A. Love, F. Mann, D. May, S. McCawley, T. McIntosh, I. McMullen, M. Moy, L. Moy, B. Murphy, K. Nelson, C. Pfannkoch, E. Pratts, V. Puri, H. Qureshi, M. Reardon, R. Rodriguez, Y. H. Rogers, D. Romblad, B. Ruhfel, R. Scott, C. Sitter, M. Smallwood, E. Stewart, R. Strong, E. Suh, R. Thomas, N. N. Tint, S. Tse, C. Vech, G. Wang, J. Wetter, S. Williams, M. Williams, S. Windsor, E. Winn-Deen, K. Wolfe, J. Zaveri, K. Zaveri, J. F. Abril, R. Guig, M. J. Campbell, K. V. Sjolander, B. Karlak, A. Kejariwal, H. Mi, B. Lazareva, T. Hatton, A. Narechania, K. Diemer, A. Muruganujan, N. Guo, S. Sato, V. Bafna, S. Istrail, R. Lippert, R. Schwartz, B. Walenz, S. Yooseph, D. Allen, A. Basu, J. Baxendale, L. Blick, M. Caminha, J. Carnes-Stine, P. Caulk, Y. H. Chiang, M. Coyne, C. Dahlke, A. Mays, M. Dombroski, M. Donnelly, D. Ely, S. Esparham, C. Fosler, H. Gire, S. Glanowski, K. Glasser, A. Glodek, M. Gorokhov, K. Graham, B. Gropman, M. Harris, J. Heil, S. Henderson, J. Hoover, D. Jennings, C. Jordan, J. Jordan, J. Kasha, L. Kagan, C. Kraft, A. Levitsky, M. Lewis, X. Liu, J. Lopez, D. Ma, W. Majoros, J. McDaniel, S. Murphy, M. Newman, T. Nguyen, N. Nguyen, M. Nodell, S. Pan, J. Peck, M. Peterson, W. Rowe, R. Sanders, J. Scott,

M. Simpson, T. Smith, A. Sprague, T. Stockwell, R. Turner, E. Venter, M. Wang, M. Wen, D. Wu, M. Wu, A. Xia, A. Zandieh, and X. Zhu. The sequence of the human genome. *Science*, 291(5507):1304–1351, Feb 2001.

[78] A.J. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–267, 1967.

[79] Jun Wang, Wei Wang, Ruiqiang Li, Yingrui Li, Geng Tian, Laurie Goodman, Wei Fan, Junqing Zhang, Jun Li, Juanbin Zhang, Yiran Guo, Binxiao Feng, Heng Li, Yao Lu, Xiaodong Fang, Huiqing Liang, Zhenglin Du, Dong Li, Yiqing Zhao, Yujie Hu, Zhenzhen Yang, Hancheng Zheng, Ines Hellmann, Michael Inouye, John Pool, Xin Yi, Jing Zhao, Jinjie Duan, Yan Zhou, Junjie Qin, Lijia Ma, Guoqing Li, Zhentao Yang, Guojie Zhang, Bin Yang, Chang Yu, Fang Liang, Wenjie Li, Shaochuan Li, Dawei Li, Peixiang Ni, Jue Ruan, Qibin Li, Hongmei Zhu, Dongyuan Liu, Zhike Lu, Ning Li, Guangwu Guo, Jianguo Zhang, Jia Ye, Lin Fang, Qin Hao, Quan Chen, Yu Liang, Yeyang Su, A. San, Cuo Ping, Shuang Yang, Fang Chen, Li Li, Ke Zhou, Hongkun Zheng, Yuanyuan Ren, Ling Yang, Yang Gao, Guohua Yang, Zhuo Li, Xiaoli Feng, Karsten Kristiansen, Gane Ka-Shu Wong, Rasmus Nielsen, Richard Durbin, Lars Bolund, Xiuqing Zhang, Songgang Li, Huanming Yang, and Jian Wang. The diploid genome sequence of an asian individual. *Nature*, 456(7218):60–65, Nov 2008.

[80] Ren L Warren, Granger G Sutton, Steven J M Jones, and Robert A Holt. Assembling millions of short dna sequences using ssake. *Bioinformatics*, 23(4):500–501, Feb 2007.

[81] David A Wheeler, Maithreyan Srinivasan, Michael Egholm, Yufeng Shen, Lei Chen, Amy McGuire, Wen He, Yi-Ju Chen, Vinod Makhijani, G. Thomas Roth, Xavier Gomes, Karrie Tartaro, Faheem Niazi, Cynthia L Turcotte, Gerard P Irzyk, James R

Lupski, Craig Chinault, Xing zhi Song, Yue Liu, Ye Yuan, Lynne Nazareth, Xiang Qin, Donna M Muzny, Marcel Margulies, George M Weinstock, Richard A Gibbs, and Jonathan M Rothberg. The complete genome of an individual by massively parallel dna sequencing. *Nature*, 452(7189):872–876, Apr 2008.

[82] Yi Xing, Tianwei Yu, Ying Nian Wu, Meenakshi Roy, Joseph Kim, and Christopher Lee. An expectation-maximization algorithm for probabilistic reconstructions of full-length isoforms from splice graphs. *Nucleic Acids Res*, 34(10):3150–3160, 2006.

[83] W. J. Youden. Index for rating diagnostic tests. *Cancer*, 3(1):32–35, Jan 1950.

[84] Daniel R Zerbino and Ewan Birney. Velvet: Algorithms for de novo short read assembly using de bruijn graphs. *Genome Res*, 18(5):821–829, May 2008.

[85] Zhengdong Zhang, Joel Rozowsky, Hugo Lam, Jiang Du, Michael Snyder, and Mark Gerstein. Tilescope: online analysis pipeline for high-density tiling microarray data. *Genome Biol*, 8(5):R81, May 2007.