Integrated Analysis of Partial Sampling Techniques in Bioinformatics

Jiang Du
CS@Yale

Feb 2010

Background

- Machine learning and data mining studies in Bioinformatics
  - build mathematical models explaining the experimental results (samples)
    - A model $Model(\Theta)$ will be defined with parameters Θ
    - The samples S will be used to find the Θ that optimizes an objective function $Obj(Model, S, \Theta)$
- Samples in Bioinformatics
  - Multiple sampling (experimental) technologies with different cost and characteristics
  - Large-scale (whole-genome), high-throughput
  - Limited by the experimental cost and budget
  - Are usually *partial* samples when compared to the object being observed

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", can we estimate the accuracy of our Θ estimation more efficiently 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?

Outline

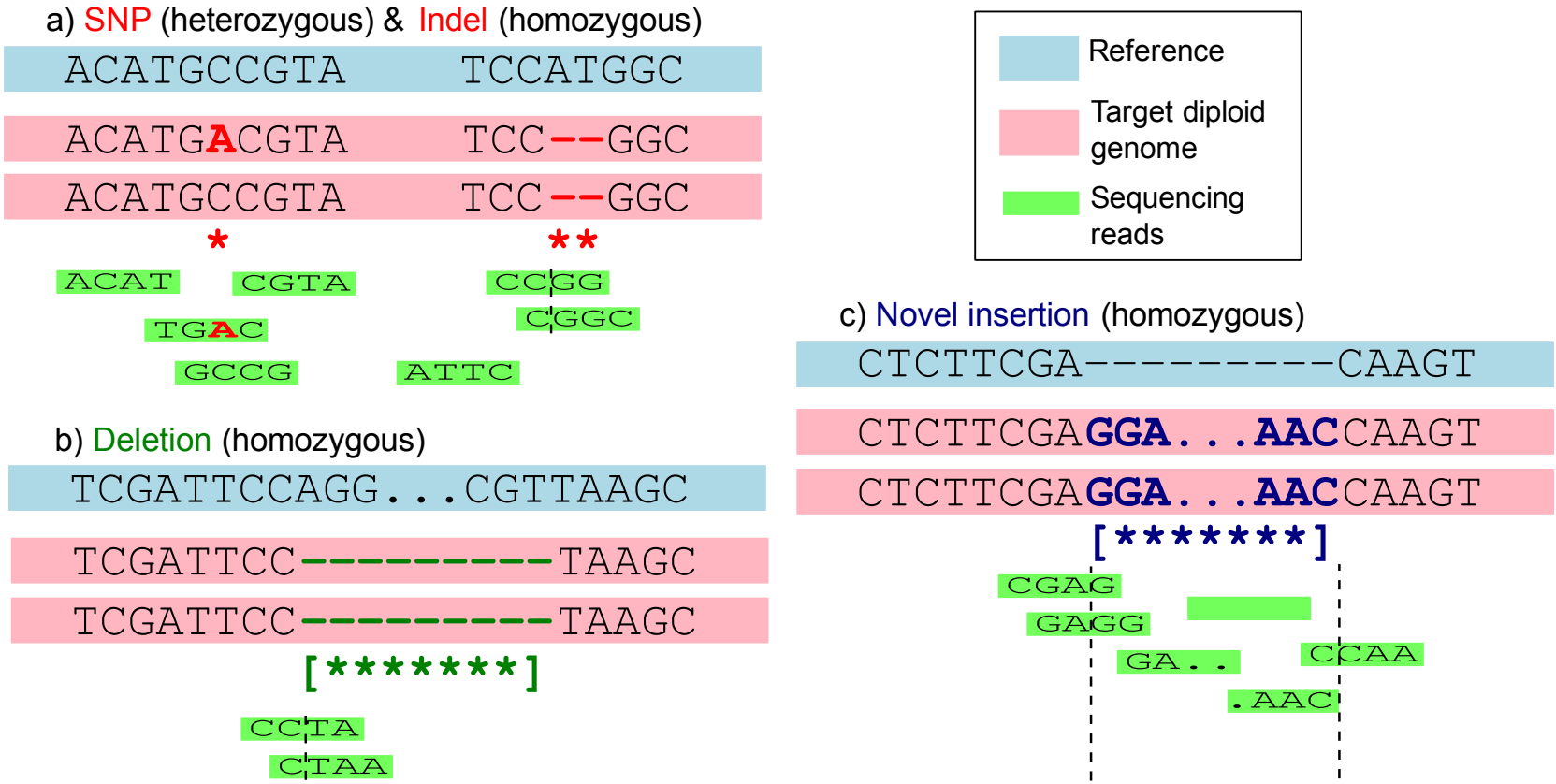
Integrated Analysis of Partial Sampling Techniques

1. Efficient Simulation of a Random Sampling Process (brief)
   - Optimal Low Cost Integration of Sampling Techniques in Re-sequencing
2. Optimal Utilization of Deterministic Sampling Techniques (brief)
   - Deterministic Sampling in A Supervised Hidden Markov Model Framework
3. Integrated Analysis of Partial Sampling Techniques
   - Distribution Estimation based on Nondeterministic Partial Samples

Genome/Gene Primer: a crude view

1. Genome
   - Long string of A, C, G, T
   - Human genome: diploid, each 3Gbases
   - Reference human genome: 3Gbases, first release in 2003
   - An individual's genome: estimated to differ from the reference by 0.05%
2. Genes
   - Regions in the genome with certain functions, e.g. coding proteins
   - Different expression levels
   - Transcribed to mRNA, then translated to protein
3. Exon (intron)
   - Regions in the gene that are (not) present in the gene transcripted mRNA

Partial Samples Example #1: individual genome re-sequencing



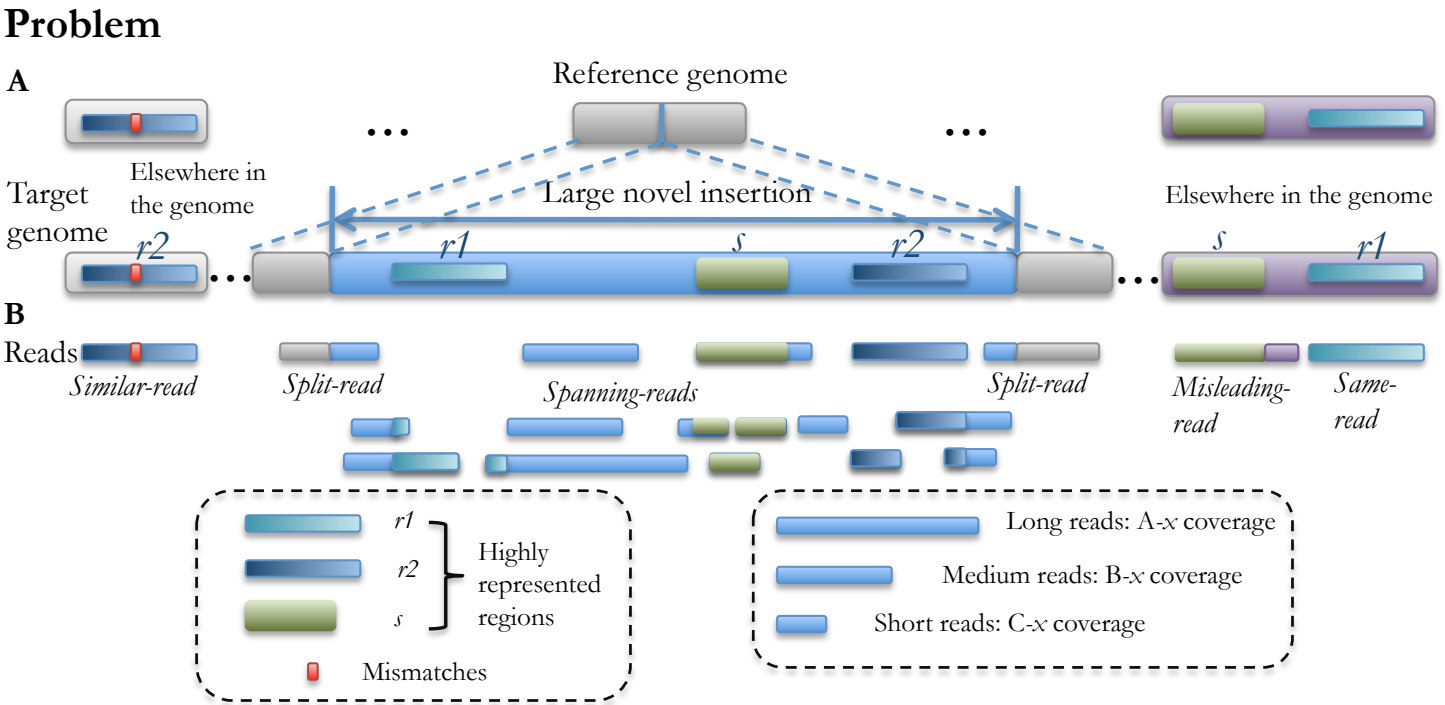
- Sequencing
  - sampling DNA fragments at random genomic locations
  - Sometimes need to be assmbled for novel insertion reconstruction

More on sequencing techniques

- Different characteristics, different costs
- Sometimes needs to be combined to obtain optimal analysis results

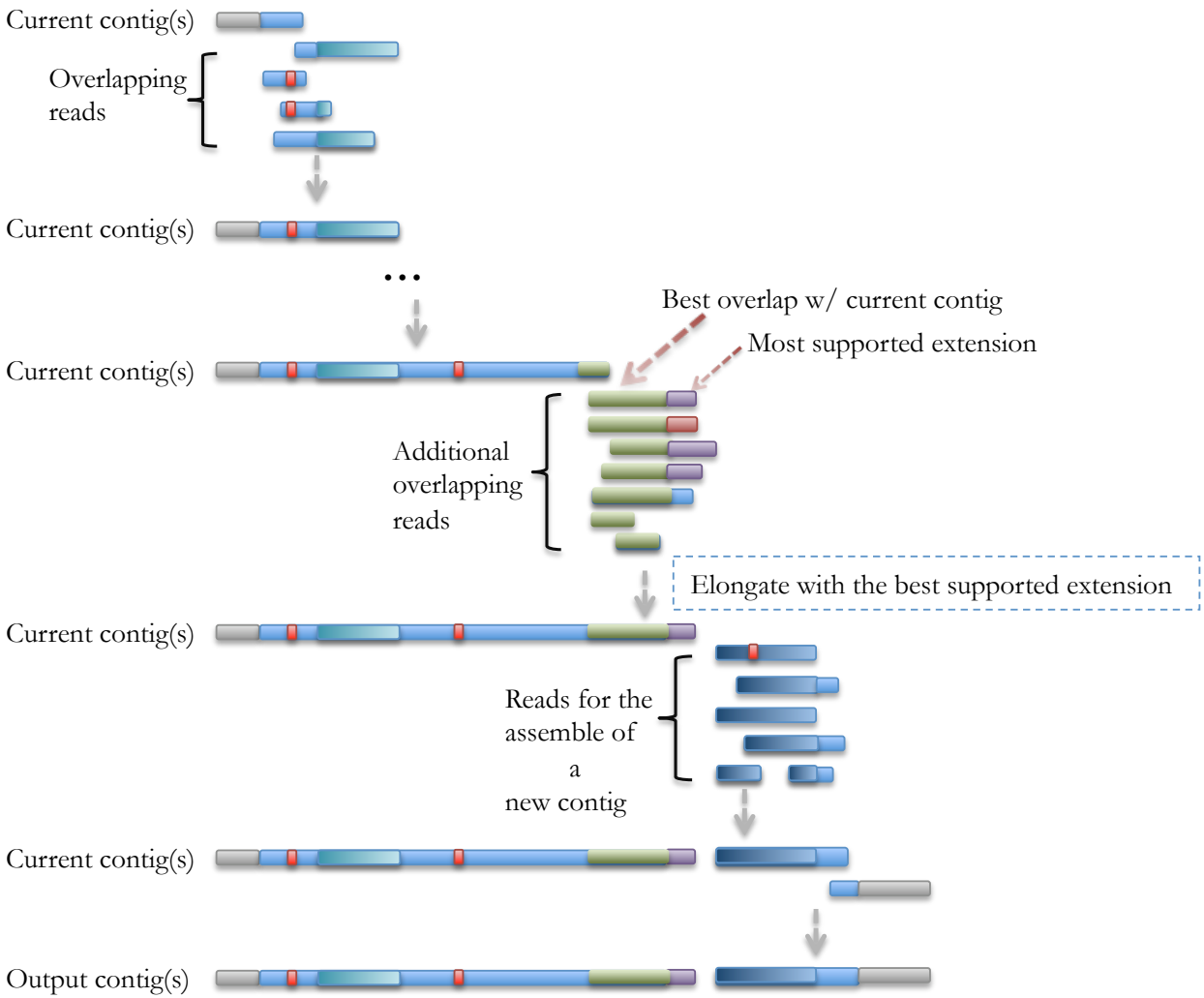
	Long Sequencing	**Medium Sequencing**	**Short Sequencing**
Read length (bases)	~ 800	~ 250	~ 30
Approximate cost per base ($)	$\sim 1E-3$	$\sim 7E-5$	$\sim 7E-6$
Error rate per base	$0.001-0.002\%$	$0.3-0.5\%$	$0.2-0.6\%$
Major error type	Substitution errors	Insertion / deletion errors (usually caused by homo-polymers)	All error types

Optimal Low Cost Integration of Sampling Techniques in Re-sequencing: reconstructing large novel insertions

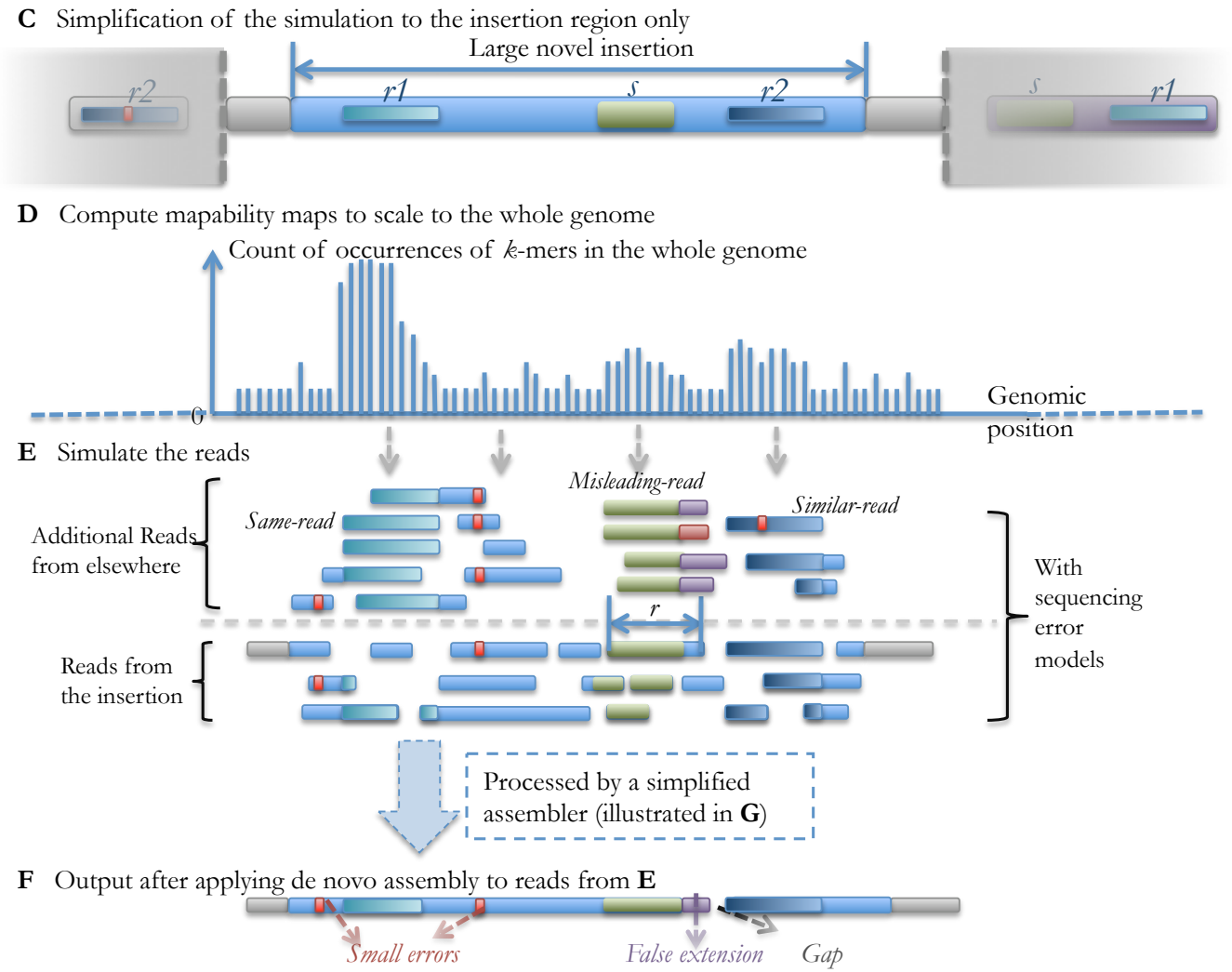


Given a fixed budget, what are the sequencing coverage A, B and C that can achieve the maximum reconstruction rate (on average/worst-case)?

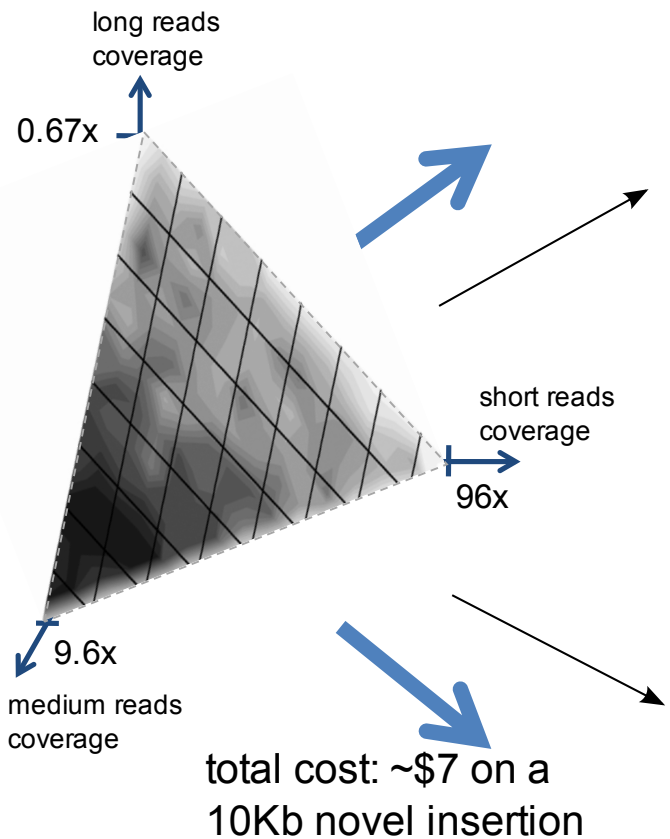
Optimal Low Cost Integration of Sampling Techniques in Re-sequencing: simple assembly algorithms in reality

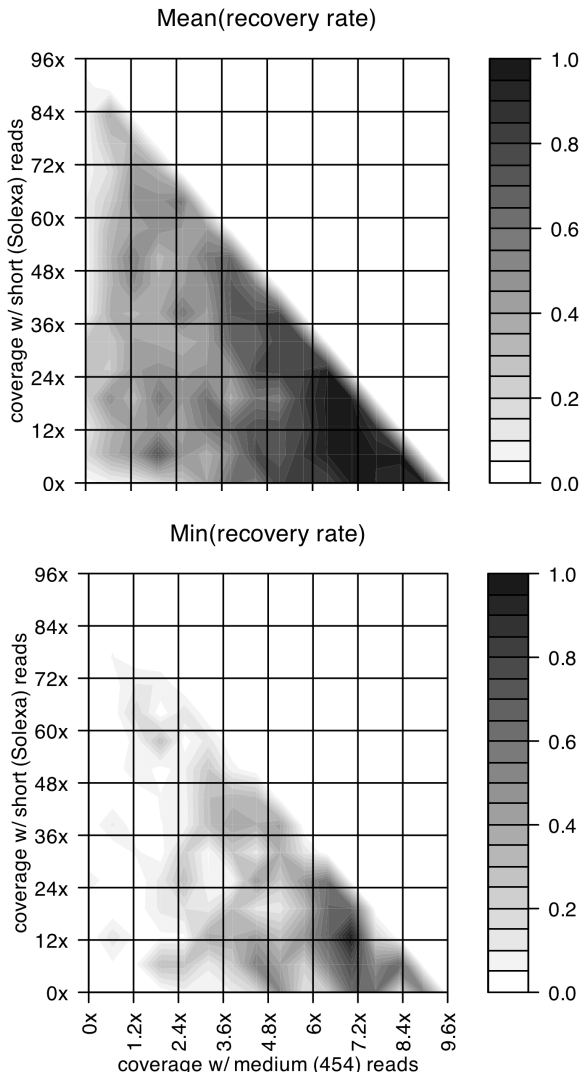


Optimal Low Cost Integration of Sampling Techniques in Re-sequencing: simulation based method

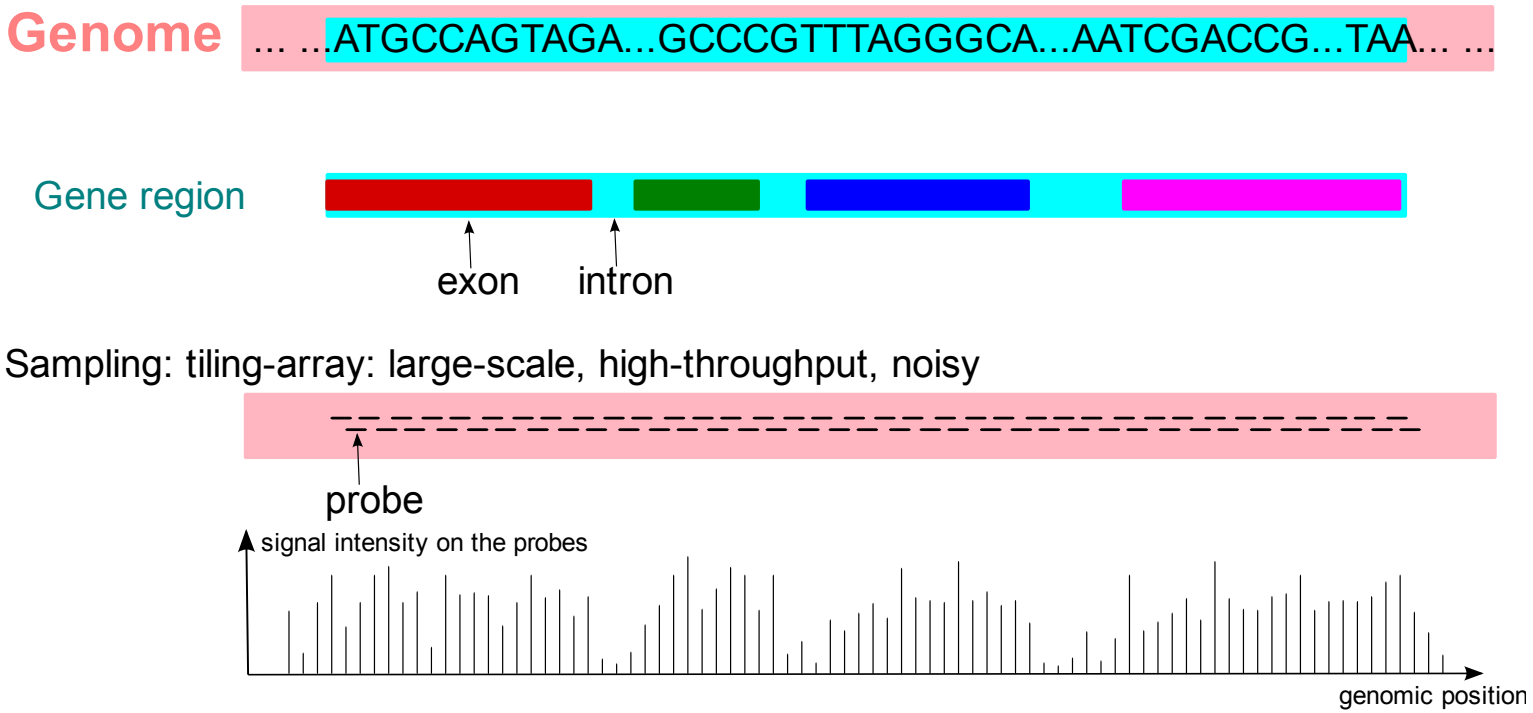


Optimal Low Cost Integration of Sampling Techniques in Re-sequencing: results





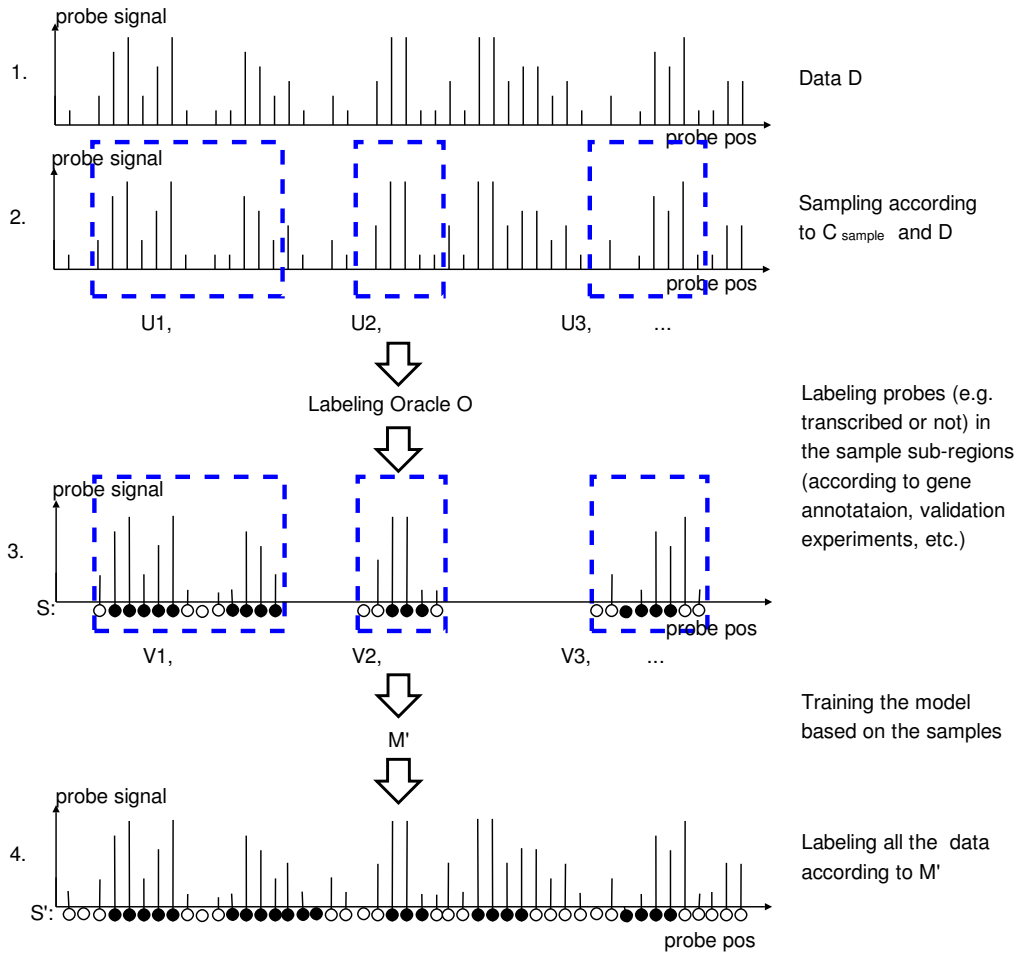
Partial Sample Example #2: identifying transcriptional activity



Transcriptional tiling array and experimental validation:

- Sampling at deterministic genomic locations

Optimal Sampling in Supervised Hidden Markov Model

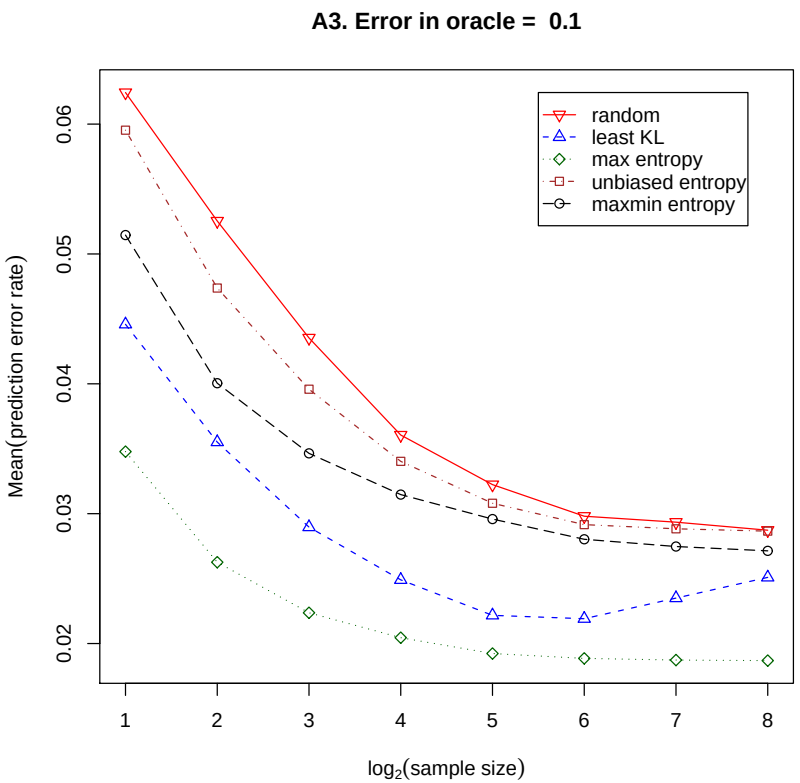


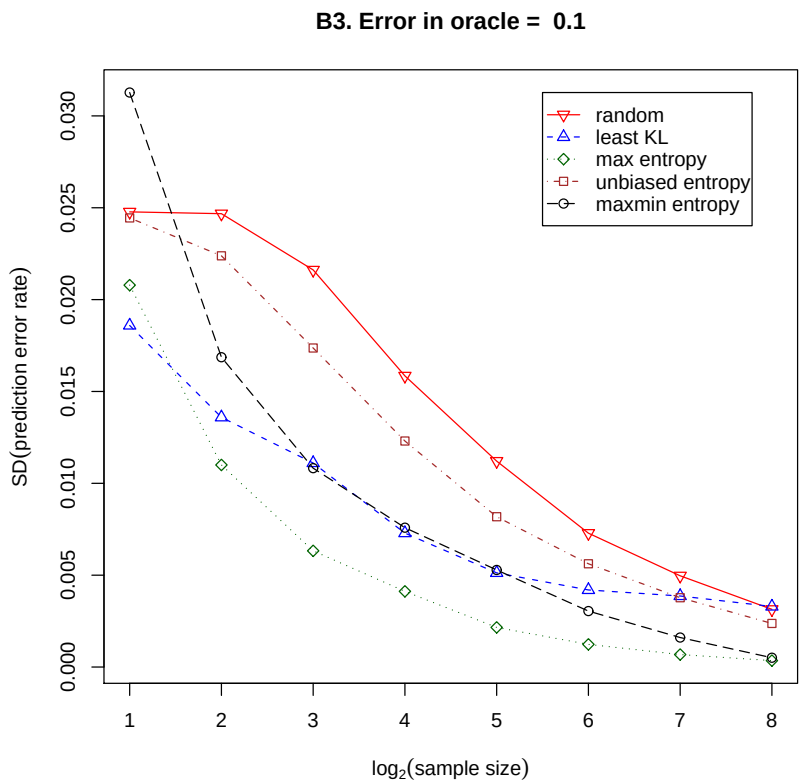
How to choose samples to best train the model?

Optimal Sampling in Supervised Hidden Markov Model

When M is a Hidden Markov Model

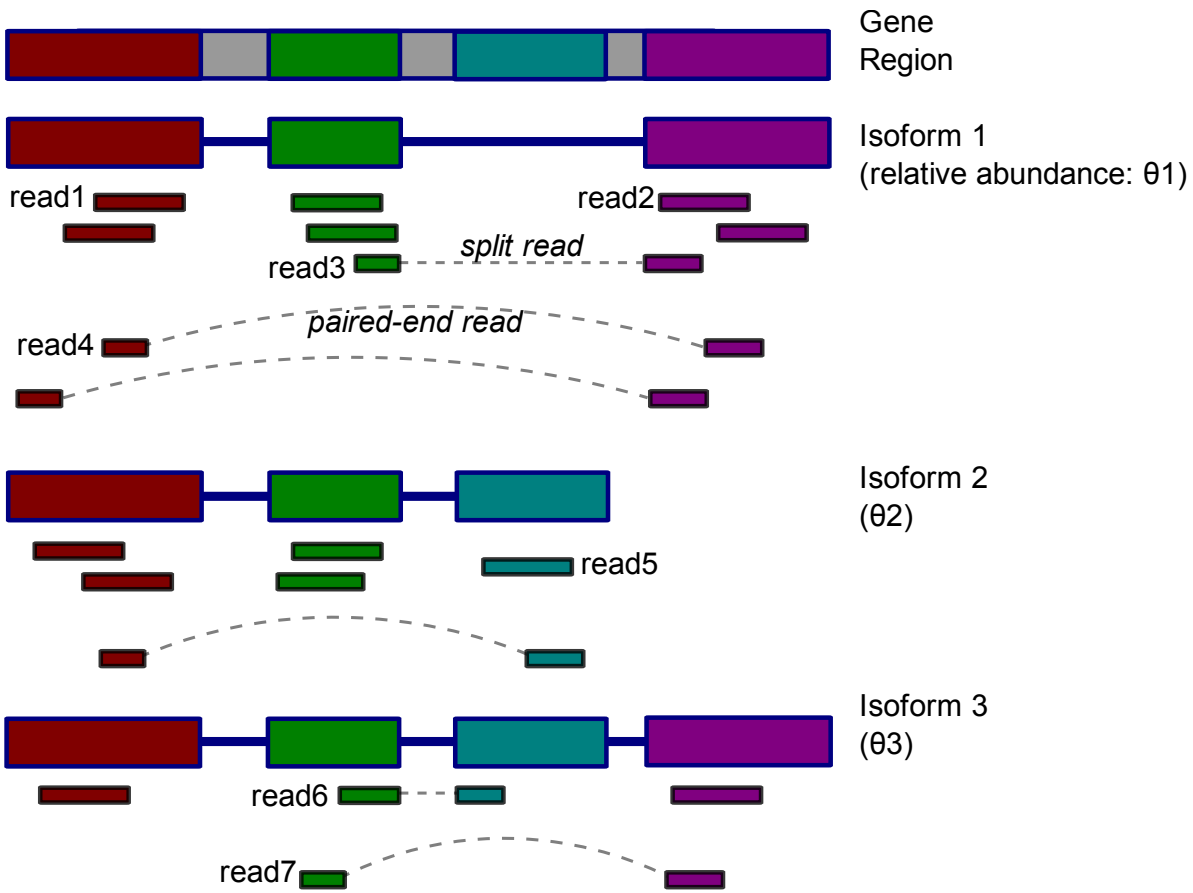
- *MaxEntropy*: selects m non-overlapping sub-regions with the highest entropies.





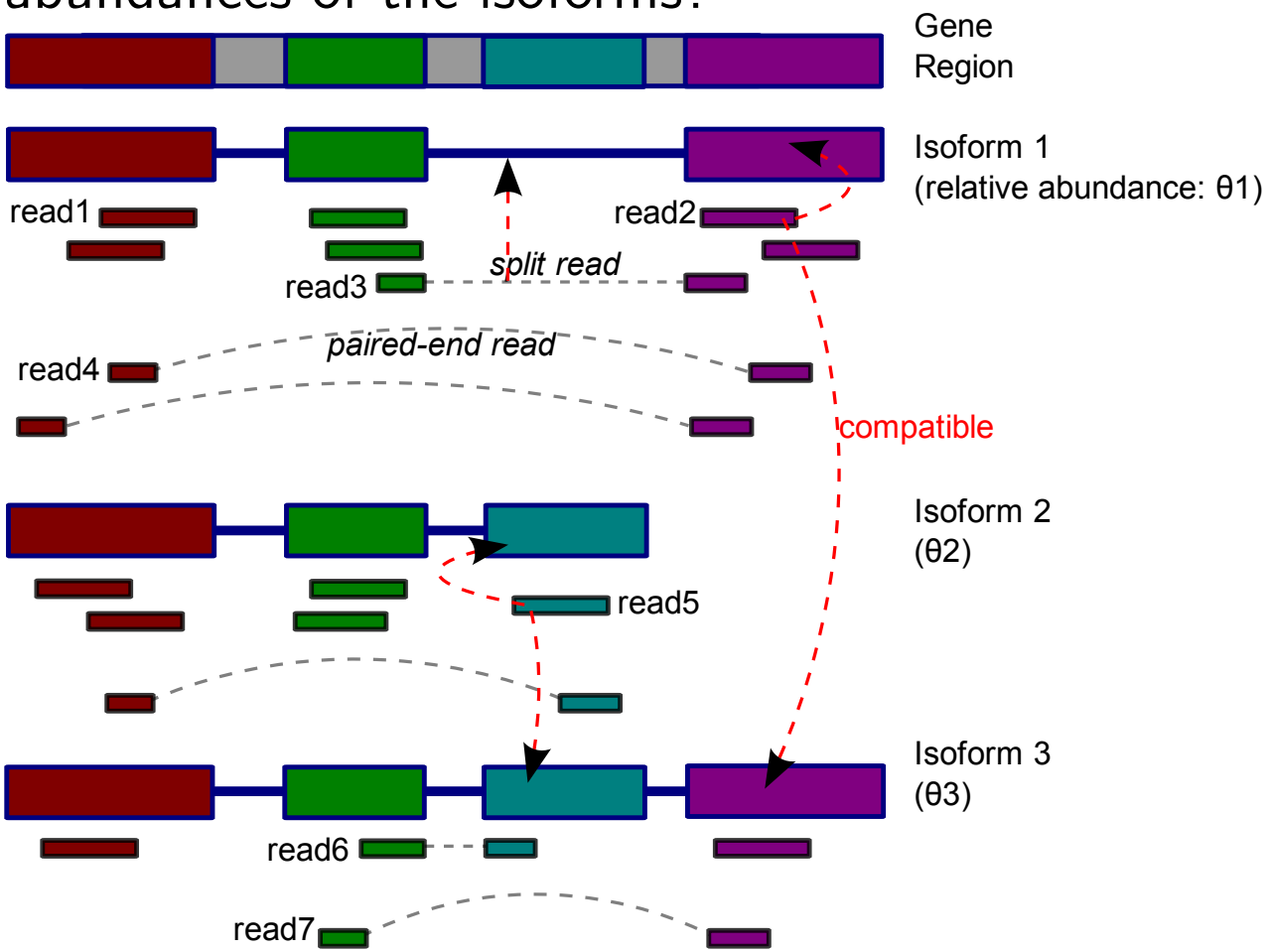
Partial Samples Example #3: transcript isoform quantification

- Transcript isoforms: exon skipping
- RNA Sequencing:
  - Sampling at random genomic locations
  - in a pool of different transcript isoforms



Isoform Quantification based on Partial Samples

Given the isoform structures and the reads, what are the relative abundances of the isoforms?



Generalized Question #1: Distribution Estimation based on Partial Samples

- Isoforms ($I = \{I_1, ..., I_K\}$): Objects that may be similar to each other
  - Different object has different abundances ($\Theta = (\theta_1, ..., \theta_K)$)
- Reads: Partial samples generated based on I and Θ
  - One partial sample may be compatible to multiple isoforms
  - Can be generated by different sampling (sequencing) techniques ($Samp_1, ..., Samp_M$)
  - Each sampling technology has its own mechanism for generating a sample (e.g. read length, sequencing bias)
- Question #1: Given I and $S = \{s$ from $Samp_m | m = 1, ..., M\}$, how to estimate Θ?

Probabilistic Solution to Q1: Maximum Likelihood Estimation

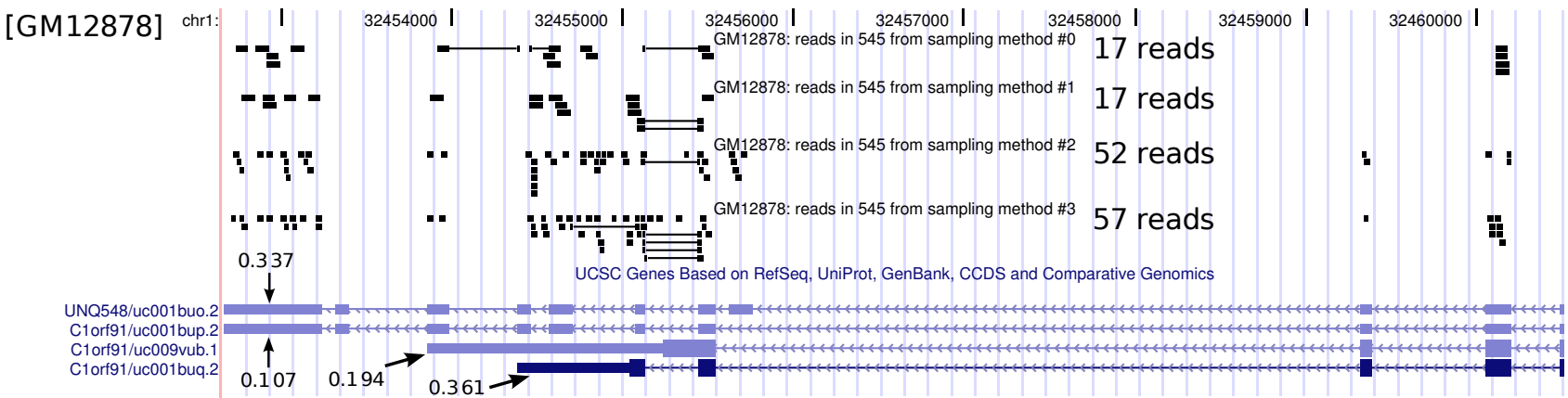
- For each possible isoform I_k, assign a probability (abundance) value θ_k
- Given all the sequencing data S, Find the Θ that maximizes $Pr(S|\Theta)$
  - Consider integrating different types ($Samp_m$) of sequencing data
  - For each sample s:
    - $\delta_{s,k}$: indicator of whether s is compatible with I_k
    - take into account the local model of sequences being generated
    - $G_{s,k}^{(m)} = Pr(\text{generating } s|I_k, Samp_m)$
    - simplified G: reads with fixed length and uniformly random starts along I_k; always uniquely mappable back to the genome
- $\hat{\Theta} = argmax_{\Theta} \sum_{m=1}^{M} \sum_{s=s_{m,*}} \log \sum_{k=1}^{K} \delta_{s,k} \theta_k G_{s,k}^{(m)}$

Solving MLE with Expectation Maximization

- Introduce hidden variable Z
  - $Z_{s,k} = Ind(s \text{ is from } I_k)$
  - Also define: $\zeta_{s,k}^{(n)} = \mathbf{E}_{Z|S,\Theta^{(n)}}[Z_{s,k}]$
  - $\zeta_{s,k}^{(n)} = \frac{\delta_{s,k}\theta_k^{(n)}G_{s,k}^{(m)}}{\sum_{k'=1}^{K}\delta_{s,k'}\theta_{k'}^{(n)}G_{s,k'}}$
- E step: $Q^{(n)}(\Theta) = \mathbf{E}_{Z|S,\Theta^{(n)}}[\log(Pr(Z,S|\Theta))]$
- M step: Maximize $Q^{(n)}(\Theta)$ with constraint $\sum_{k=1}^{K}\theta_k = 1$
- $\theta_k^{(n+1)} = \frac{\sum_{m=1}^{M}\sum_{s=s_{m,*}}\zeta_{s,k}^{(n)}}{N} = \frac{\sum_{m=1}^{M}\sum_{s=s_{m,*}}\frac{\delta_{s,k}\theta_k^{(n)}G_{s,k}^{(m)}}{\sum_{k'=1}^{K}\delta_{s,k'}\theta_{k'}^{(n)}G_{s,k'}}}{N}$

Application to human RNA-seq Data

- 4 known isoforms from UCSC known genes
- 4 sequencing technologies: 454/Solexa single/paired-end reads



Revisiting the Questions

- Question #1: Given I and $S = \{s$ from $Samp_m | m = 1, ..., M\}$, how to estimate Θ?

New questions:

- Question #2: How good is the estimation?
  - Average estimation variance: $\frac{\sum_{k=1}^{K-1} var(\theta_k)}{K-1}$
  - Using Fisher information to estimate MLE variance
- Question #3: Suppose different sampling techniques have different costs, given a fixed budget, what is a most cost-efficient way to combine these sampling methods?
  - Brute-force simulation using MLE
  - FIM based estimation

Answering Q2: Fisher Information Matrix

- Θ: isoform probabilities, w/ degree of freedom: $K-1$
- Observed and Expected FIM
  - $\mathfrak{I}(\Theta)_{p,q} = -\frac{\partial^2 \log(Pr(S|\Theta))}{\partial\theta_p \partial\theta_q}$, where $p, q = 1, ..., K-1$
  - $\mathcal{I}(\Theta)_{p,q} = \mathbf{E}\left[\mathfrak{I}(\Theta)_{p,q}\right]$
- Why is $\mathcal{I}(\Theta)$ important?
  - In one dimensional case

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

  - $\sum_{k=1}^{K-1} \frac{1}{\mathcal{I}(\Theta)_{k,k}} \sim \sum_{k=1}^{K-1} var(\theta_k)$?
- Computing $\mathcal{I}(\Theta)$
  - can be decomposed into individual samples
  - $\mathcal{I}(\Theta) = \sum_{m=1}^{M} N_m \mathcal{I}^{(m)}(\Theta)_{p,q}$
  - $\mathcal{I}^{(m)}(\Theta)_{p,q} = \mathbf{E}_{s\sim Samp_m}\left[-\frac{\partial^2 \log \sum_{k=1}^{K} \delta_{s,k}\theta_k G_{s,k}^{(m)}}{\partial\theta_p \partial\theta_q}\right]$

Fast computation of $\mathcal{I}^{(m)}(\Theta)$: Definitions

Based on the concept of equivalent samples.

Definition 1

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

Lemma 2

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

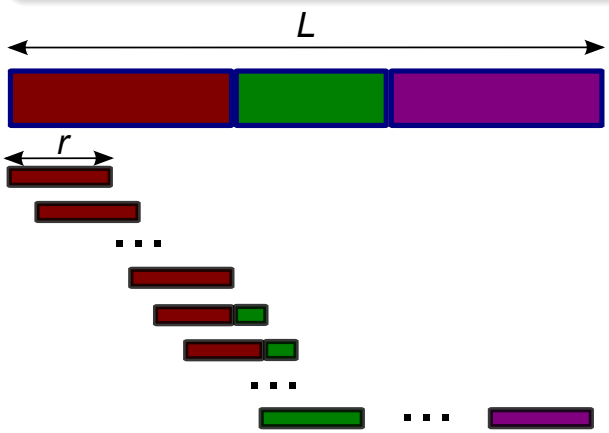
Definition 3

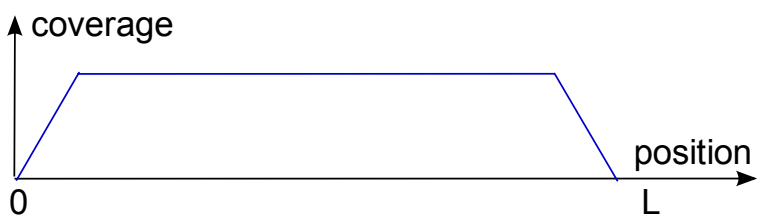
A set of partial samples S is an equivalent sample set w.r.t. $Samp_m$ if and only if $\forall s_1, s_2 \in S$, s_1 and s_2 are equivalent w.r.t. $Samp_m$.

A simple shotgun read generation model

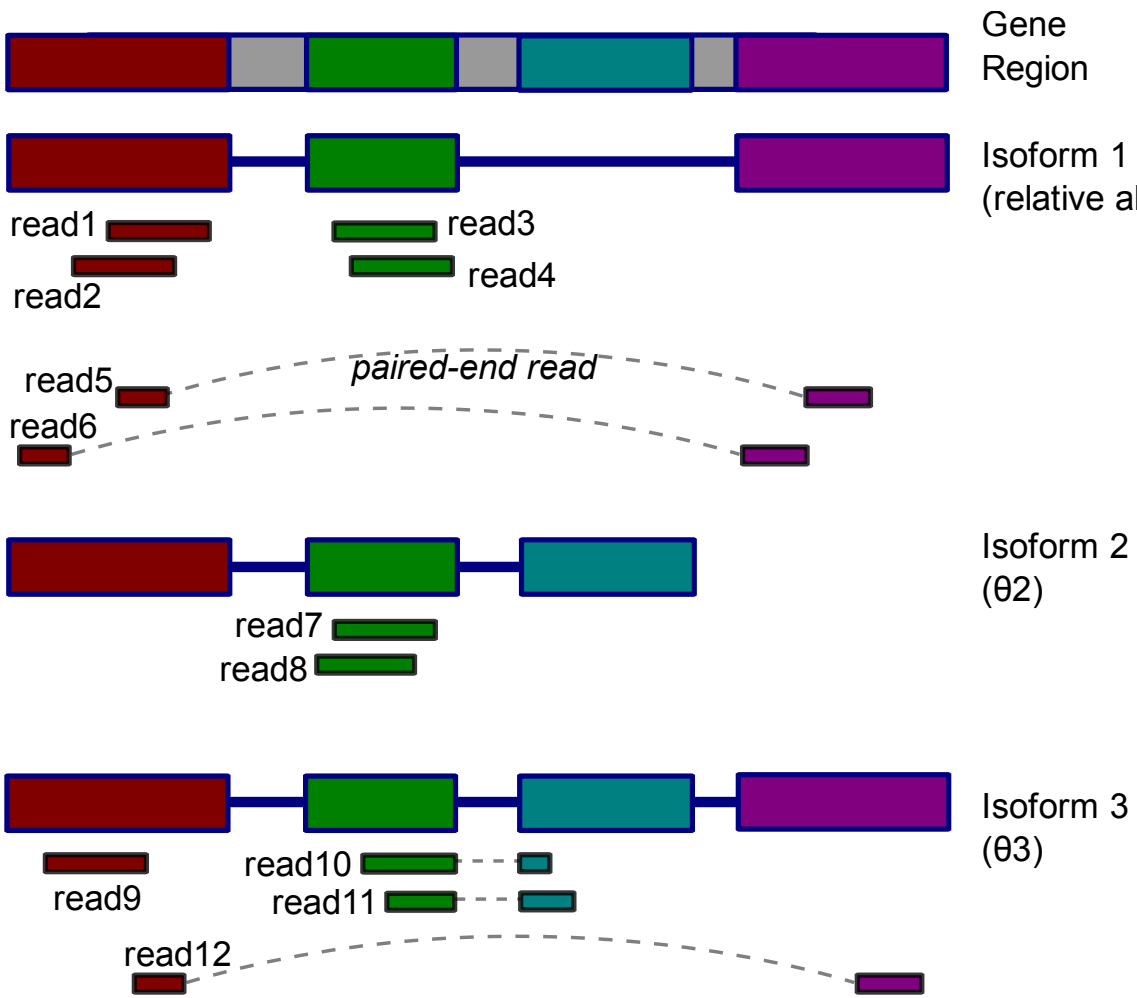
Definition 4

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: $G^{(m)}_{s,k} = 1/(l_k - r_m + 1)$.





Examples of Equivalent Samples



Fast computation of $\mathcal{I}^{(m)}(\Theta)$

Lemma 5

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:

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

Theorem 6

Given the sample generation model $Samp_m$ in Definition 4, if two samples s_1 and s_2 generated by this method overlap with all the junctions in a same set of connected exons $e_{k_1} \longrightarrow e_{k_2} \longrightarrow ... \longrightarrow e_{k_n}$, then s_1 and s_2 are equivalent w.r.t. $Samp_m$.

Algorithms for computing FIM: Brute-force

Enumerate all possible samples.

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

REQUIRE: Possible isoforms $I = \{I_1, I_2, \ldots, I_K\}$;
Relative abundances $\Theta = (\theta_1, \theta_2, \ldots, \theta_K)$;
Sampling method $Samp_m$ Integer $p, q \in \{1, 2, \ldots, K-1\}$.
ENSURE: 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 $\mathcal{I}$

Algorithms for computing FIM: Fast

Combine equivalent samples within isoforms.

Algorithm 2 FastShotgunFIM($I, \Theta, Samp_m, p, q$)

...

$\mathcal{I} \leftarrow 0$
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**
    $b \leftarrow a + r_m$;
    $s \leftarrow s^k_{[a,b)}$
    $(\mathbf{e_{k_1}} \rightarrow \mathbf{e_{k_2}} \rightarrow \ldots \rightarrow \mathbf{e_{k_n}}) \leftarrow \texttt{overlappingExons}(\mathbf{s}, \mathbf{I_k})|$
    $\mathbf{N_{EqSamples}} \leftarrow \min\left(\sum_{\mathbf{e_{k'}} \in \mathbf{I_k}; \mathbf{k'} <= \mathbf{k_1}} \texttt{length}(\mathbf{e_{k'}}) - \mathbf{a}, \sum_{\mathbf{e_{k'}} \in \mathbf{I_k}; \mathbf{k'} <= \mathbf{k_n}} \texttt{length}(\mathbf{e_{k'}}) - \mathbf{b} + \mathbf{1}\right)$
    $\mathcal{I}_k \leftarrow \mathcal{I}_k + N_{EqSamples} G^{(m)}_{s,k} \mathfrak{I}^{(m)}_s(\Theta)_{p,q}$
    $a \leftarrow a + N_{EqSamples}$
  end while
  $\mathcal{I} \leftarrow \mathcal{I} + \theta_k \mathcal{I}_k$
end for
return $\mathcal{I}$

Algorithms for computing FIM: Faster

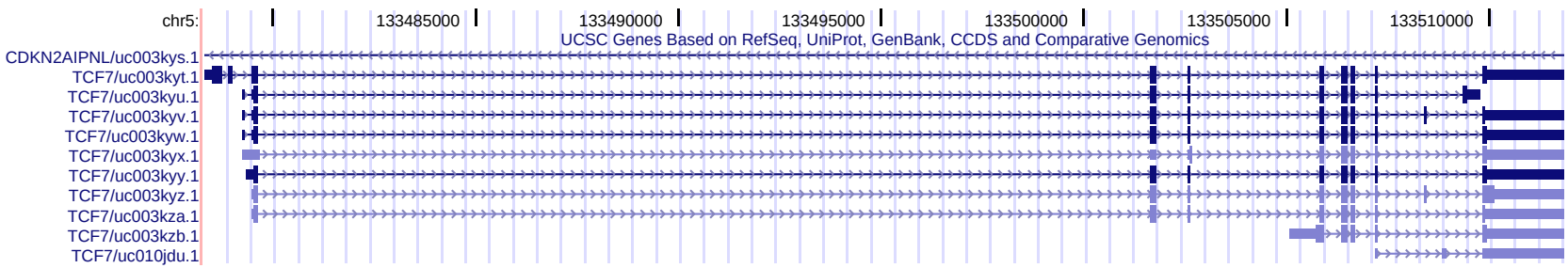
Combine equivalent samples within and across isoforms.

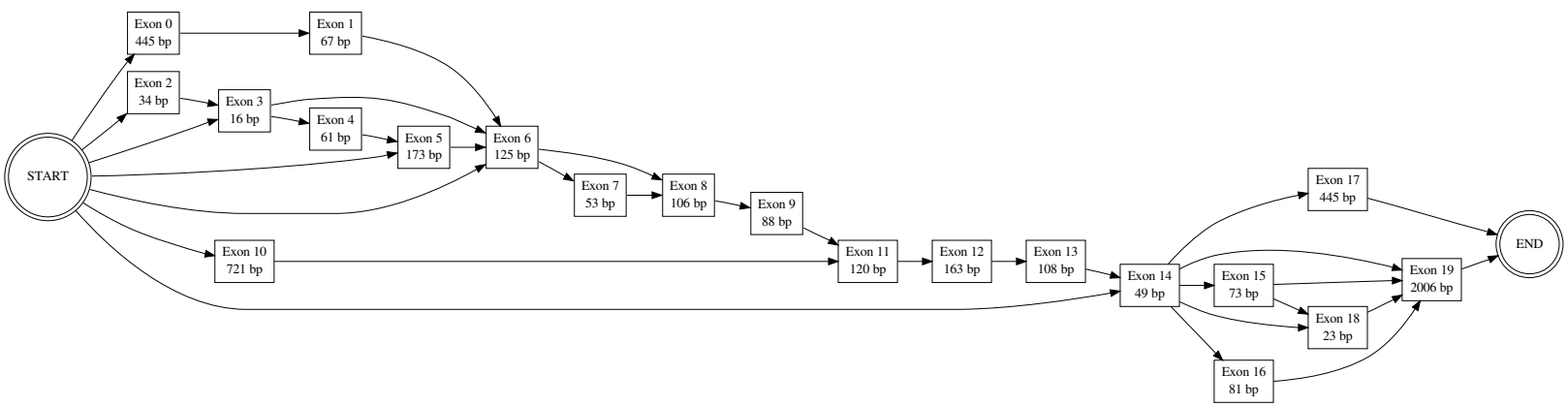
Algorithm 3 FasterShotgunFIM($I, \Theta, Samp_m, p, q$)

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**
    ...
    $N_{EqSamples} \leftarrow \min\left(\sum_{e_{k'} \in I_k; k' <= k_1} \texttt{length}(e_{k'}) - a, \sum_{e_{k'} \in I_k; k' <= k_n} \texttt{length}(e_{k'}) - b + 1\right)$
    $\mathcal{I} \leftarrow \mathcal{I} + \theta_k N_{EqSamples} G_{s,k}^{(m)} \mathfrak{I}_s^{(m)}(\Theta)_{p,q}$
    $CoveredSampleStarts_k \leftarrow CoveredSampleStarts_k + [a, a + N_{EqSamples})$
    for all $I_{k'} \neq I_k$ **do**
      if $I_{k'}$ contains $(e_{k_1} \rightarrow e_{k_2} \rightarrow \ldots \rightarrow e_{k_n})$ **then**
        ...
        $CoveredSampleStarts_{k'} \leftarrow CoveredSampleStarts_{k'} + [a', a' + N_{EqSamples})$
      end if
    end for
    $a \leftarrow \texttt{minNotCoveredStart}(CoveredSampleStarts_k, Samp_m)$
  end while
end for
return $\mathcal{I}$

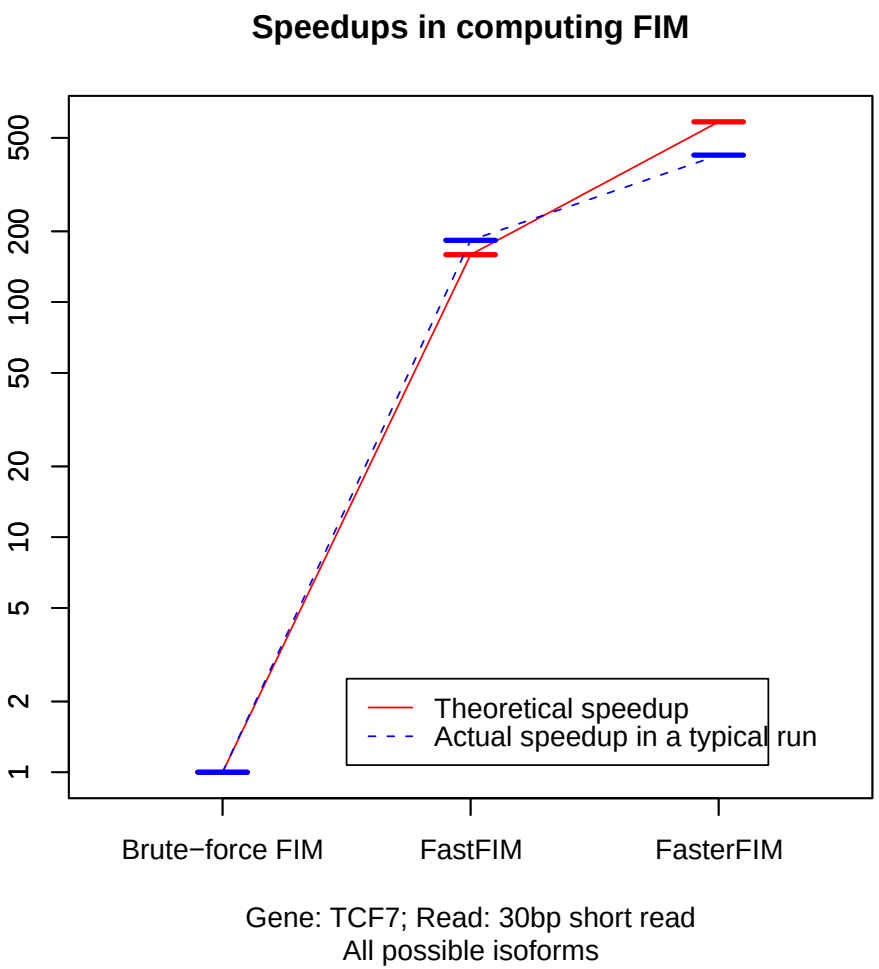
Example: TCF7

- TCF7 in UCSC knownGenes
  - 10 known isoforms
  - 96 possible paths (isoforms) in the splicing graph
- Assumptions
  - The known isoforms are the actual isoforms (the "true" isoforms)





Example: Speedups in computing FIM

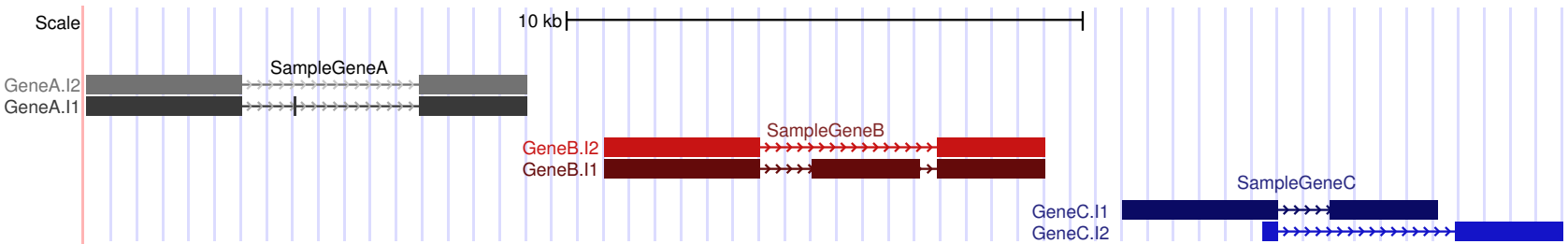


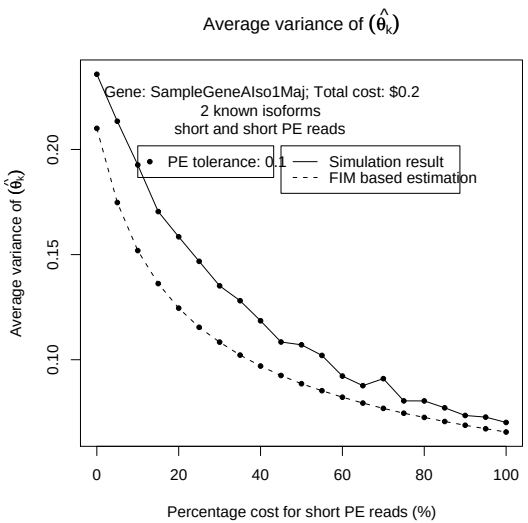
Gene: TCF7; Read: 30bp short read
All possible isoforms

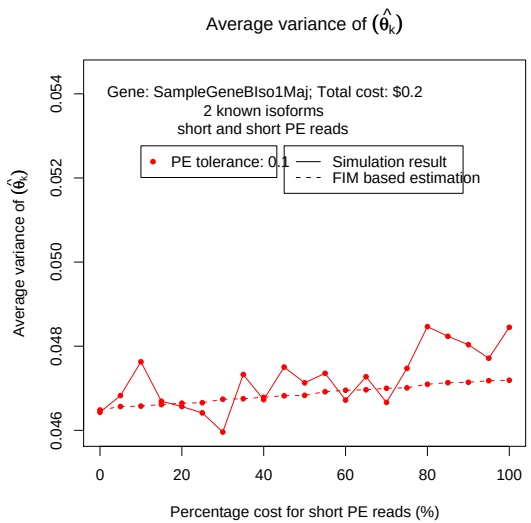
Answering Q3: Simulations on Simplified Gene Models

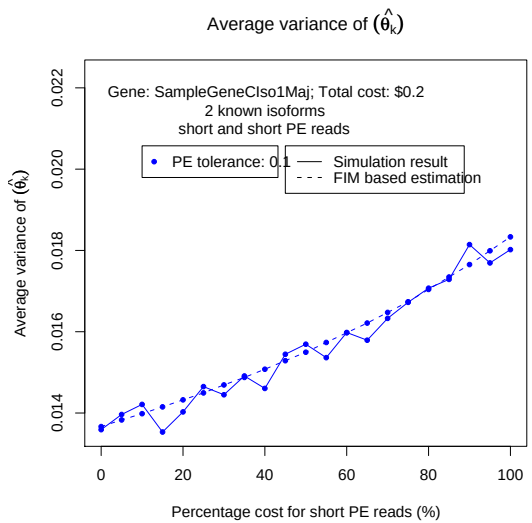
Simulation results

- 3 simplified gene models
- Short single and paired-end reads w/ fixed total cost
- 1000 trials for each cost configuration









Answering Q3: Simulations on Simplified Gene Models

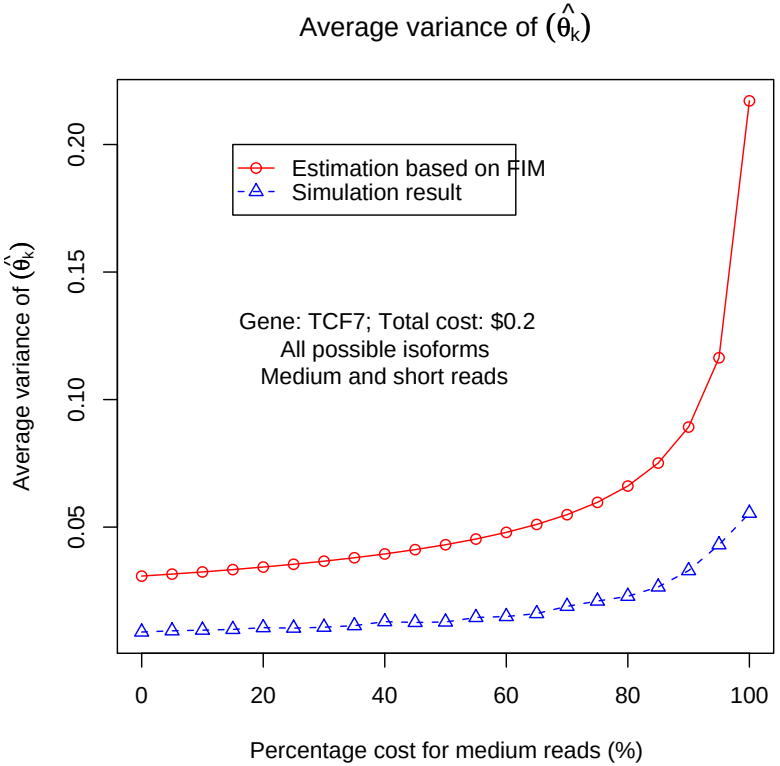
Total trials for one gene	Number of trials $\times$ Number of sampling method combinations $= 1000 \times 21$
Total FIM computation for one gene	Number of sampling methods$= 2$
Total CPU time used by brute-force simulation	~ 52 minutes
Total CPU time used by FIM based heuristic	< 1 second

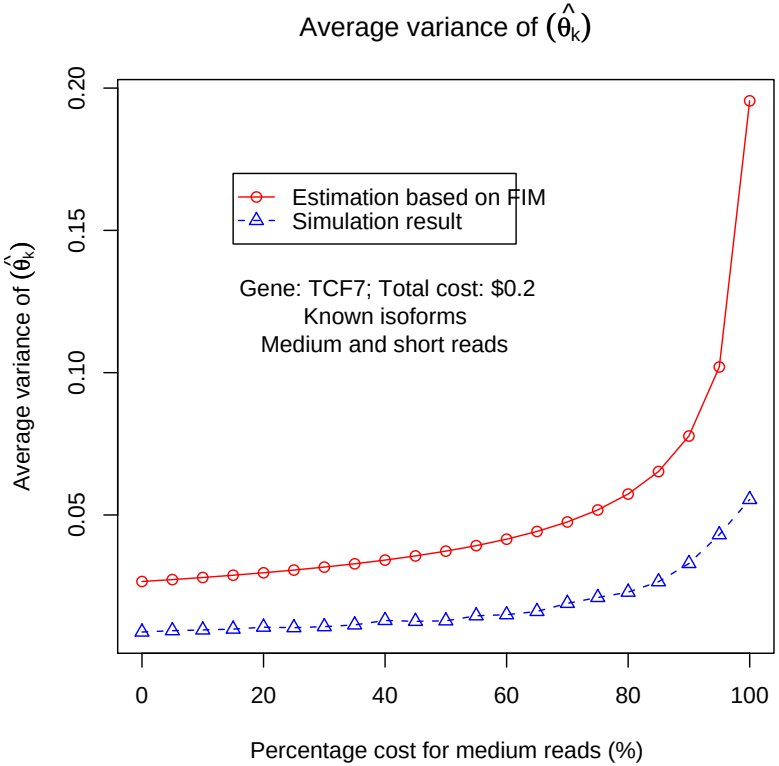
Answering Q3: Simulations on TCF7

- Average variance of MLE estimation $\hat{\theta}_k$:
  - $\frac{\sum_{k=1}^{K-1} var(\theta_k)}{K-1}$
- Estimation based on FIM
  - $\frac{\sum_{k=1}^{K-1} \frac{1}{\mathcal{I}(\Theta)_{k,k}}}{K-1}$
- Simulation:
  - gene: TCF7, equal probabilities for its known isoforms
  - medium reads: 250bp, \$7 $\times 10^{-5}$ per bp
  - short reads: 30bp, \$7 $\times 10^{-6}$ per bp
  - total budget: \$0.2
  - at each cost configuration (e.g. \$0.1 for medium reads, \$0.1 for short reads)
    - 200 trials
    - compute $\frac{\sum_{k=1}^{K-1} var(\theta_k)}{K-1}$

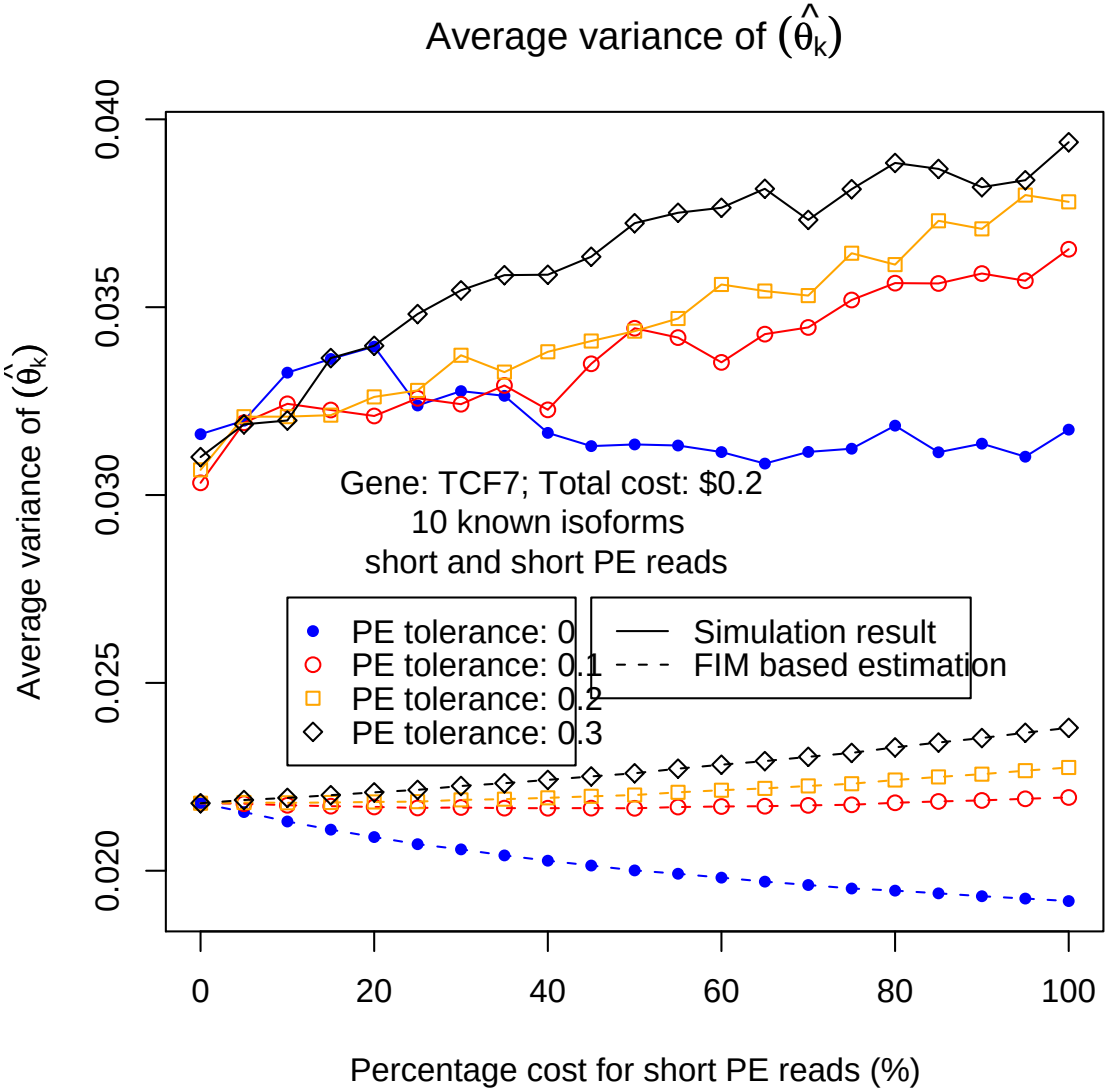
Answering Q3: Simulations on TCF7 contd.

Simulation results vs. Estimation based on $\mathcal{I}(\Theta)$





Answering Q3: Simulations on TCF7 contd.



Answering Q3: Simulations on TCF7 contd.

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

Revisiting the Questions

- Question #1: Given I and $S = \{s$ from $Samp_m | m = 1, ..., M\}$, how to estimate Θ?
  - MLE
- Question #2: How good is the estimation?
  - Efficient algorithm to compute the Fisher information matrix
  - Using FIM to estimate MLE variance
- Question #3: Suppose different sampling techniques have different costs, given a fixed budget, what is a most cost-efficient way to combine these sampling methods?
  - Brute-force simulation using MLE
  - FIM based estimation

Conclusion & Discussions

- Recap
  - Integrated analysis of partial samples
  - Fast algorithms to estimate analysis performance
  - Optimal integration
  - Efficient simulation
- Further Discussions
  - Incorporation of more accurate partial sampling models.
    - e.g. more realistic modeling of the sequencing process
    - tradeoff between model accuracy and computational efficiency
  - Incorporation of domain-specific knowledge
    - utilizing relevant biological knowledge
    - e.g. characteristics of genes and splicing
  - Integration of different types of sampling methods
    - e.g. combining sequencing and array data

Acknowledgements

- Mark Gerstein @Gerstein lab@Yale
- Drew McDermott, Martin Schultz @CS@Yale
- Roderic Guigo @CRG
- Michael Snyder @Stanford
- Joel R, KevinY, Lukas H, Zhengdong Z, Jing L, Hugo L, Andrea S, ..., TECH, ASSEM @Gerstein lab
- Jiaqian W @Snyder lab, Nicholas C, Robert B @HPC@Yale, Yong K @Keck@Yale, Joseph C, Andrew B @Stat@Yale
- Family (Beiliang D, Weiwei J, Cecile L)
- Roommates (Hong G, Bing W, Yinli X, Jianye L)
- Friends (Chen X, Songhua X, Hongzhi W, Yinghua W, Peishen Q, Haiyong X, Zheng M, Hao W, Jinqiang H, ...)
- EE

Details: Optimal Sampling in HMM: a bit of formalism

Definition 7 (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.

Details: Mapability

Definition 8 (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 also denote that $M(s, G) = M(s, G, 0)$, which is the extract occurrence of s in G.

Lemma 9

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.

Lemma 10

Given a genome G, a sequence s, and two non-negative integers m, m', if $m > m'$, then $M(s, G, m) \geq M(s, G, m')$.

Details: Computing $\zeta_{s,k}^{(n)}$

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

Details: E step

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

Details: M step

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

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

$$\frac{\partial T^{(n)}(\Theta, \lambda)}{\partial \theta_k} = 0$$

$$\sum_{m=1}^{M} \sum_{i=1}^{N_m} \frac{\zeta_{s,k}^{(n)}}{\theta_k} + \lambda = 0$$

$$\theta_k = -\frac{\sum_{m=1}^{M} \sum_{i=1}^{N_m} \zeta_{s,k}^{(n)}}{\lambda}$$

Details: M step contd.

Inserting the result above into the constraint, we have:

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

Details: M step contd.

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

Guaranteed convergence to a local maximum.