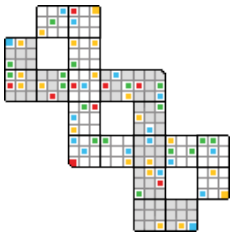

Avadis NGS v1.5
Reference Manual

October 22, 2013

Contents

Chapter 1

Next Generation Sequencing Concepts

This chapter introduces concepts and terminology that will be referred to in the rest of the manual.

1.1 Concepts and Terminology

- **Read**

  A read is a sequence of nucleotides called by the sequencer corresponding to the end of a fragment. The sequence is called out in the direction of growth of the DNA strand, i.e., from $5^{'}$ to $3^{'}$. If the template strand is the forward one, then the read created will align in the negative direction to the reference. Conversely, reverse template strands on sequencing give rise to positively aligning reads.

- **Base quality**

  Each nucleotide output by the sequencer has a corresponding number accorded to it. This is the quality value and is assigned based on the confidence with which a particular base is called by the base-calling software.

  If the true nucleotide is B and the one estimated by base caller is $B^{'}$, the base error $\epsilon = P\left[B \neq B^{'}\right]$, and the associated base quality Q_B is given by

  $$Q_B = -10 \log_{10} \epsilon, \tag{1.1}$$

- **Alignment score**

  The match of a read to a reference sequence can be represented in the form of two (possibly gapped) strings placed one over the other. The following figure shows the optimal alignment of a read, `ATATTAGCC`, with a portion of the reference with the sequence `ATTAAGGC`

  ```
  pos :  1234567890
  read:  A-TATTAGCC
  ref :  ATTA--AGGC
  ```

  The total length occupied by the alignment is 10, and it consists of:

  – 6 positions where the read and reference characters match exactly

  - a deletion of length one at position 2 of the alignment
  - an insertion of length 2 at positions 5,6, and
  - a mismatch at position 9

  The alignment score of a match is defined as the percentage of the alignment length which is occupied by exactly matching characters (60% in the above example).

- **Depth of coverage**

  The number of reads that align to a particular postion of the reference is said to be the coverage at the position. This number when averaged across all positions of the reference gives the average depth of coverage. Obviously greater sequencing depths lead to higher accuracy in later deductions.

- **Mate pairs and paired-end reads**

  Paired-end reads and **mate-pairs** refer to methodologies that give information about two reads belonging to a pair. The basic idea involves shearing DNA into random fragments of a selected size, called the **insert length**, and then sequencing both ends of each fragment.

  In addition to the usual sequence information, the physical distance between the two reads in the sample genome is also known. The extra distance information provides a dramatic increase in alignment accuracy, by providing a kind of linked scaffolding, useful in reconstructing the parent genomic sequence. This is especially helpful in identifying structural variants, and aligning across repetitive regions.

  "Paired-end sequencing" sequences both the forward and reverse template strands of the same DNA molecule. Each end is separately sequenced; the two sequences are termed paired end reads.

  The distance between paired-end reads is limited by the cluster generation technology to $\sim$ 300bp (200-600bp). In "Mate-pair sequencing" tags that are sequence belong to ends of a much larger molecule, typically between $\sim$ 2 and 10kbps.

- **Mapping quality of reads**

  Most alignment algorithms assign quality scores to a read based on how well the read aligned with the reference. These scores are relevant in SNP detection because they measure the likelihood of a read originating from the suggested position on the reference. Even if the individual bases on a read are called with high quality values, the read may align imperfectly with the reference. The mapping quality score takes into account the inserts, deletes, and substitutions necessary for alignment at a particular position. Note that unlike alignment score which considers only the sequences of the read and the reference, mapping quality can incorporate other considerations like the presence of similar sequences in the genome, the location of the mate (in case of mate-pair libraries) etc.

- **Read count distribution**

  Consider an experiment that sequences DNA fragments from a selected genome. In the absence of biases due to various biological phenomena such as copy number variation, hotspots in the chromatin structure prone to easy breakage, PCR bias etc., the reads will map to the reference genome fairly uniformly, showing no particular regional preference. The resulting read count for any region would then be expected to follow a Poisson distribution fairly closely, with distribution parameters dependent on (i) the total number of reads, (ii) the length of the region and (iii) the length of the sequenced portion of the genome.

- **Insert length distribution**

  The distribution of the insert length, i.e., the distance between mate pairs or paired-end reads is assumed to be Gaussian with a given mean and variance, which is based on the technology and library preparation protocol.

- **Duplicate reads**

  Duplicate reads are identical to each other in sequence (but possibly differing in quality values), and are common next-generation sequencing artifacts. The maximum expected number of duplicates are easily computed based on the length of the read as well as the sequencing depth. Often, duplicates are far in excess of this number, creating large, sharp spikes in the genomic read density profile. This biased representation occurs as a result of hotspots for chromosomal breakage due to the inherent chromatin structure, or perhaps due to a sequencing bias.

- **Normalization of read counts based on sample concentration**

  Different experiments often start by processing varying amounts of DNA and using various techniques for isolation and fragmentation. Thus, comparison across samples requires normalization by appropriately scaling read counts.

- **Repeat regions in the genome**

  Large mammalian genomes are often repetitive in nature. Reads from repeat regions do not align uniquely and are referred to as multiply mapping reads. Algorithms either discard reads mapping multiply to the genome, or assign them probabilistically.

- **Sequencing error**

  The reads generated by an NGS system may differ from the actual sample sequence due to a variety of errors ranging from sample contamination, PCR errors to instrument errors. The base quality associated with each base of a read is a rough measure of the error.

- **Homopolymer**

  A *homopolymer* is a repetitive sequence of a single nucleotide, e.g., AAAAAA. Some sequencers exhibit imprecise representations of homopolymers due to saturation of signal intensity while incorporating nucleotides. Their immediate neighbors are also subject to an increase in "carry forward errors" which affect the next flow of the same dNTP. Such regions need to be handled carefully by analytical algorithms.

- **Alignment errors**

  These are not strictly part of NGS technology, however, in software analysis tools such as **Avadis NGS** that directly handle aligned data, slips in alignment can lead to completely erroneous conclusions. Taking into account the read mapping qualities when provided could help reduce alignment errors.

- **Paired read models**

  In a pair of reads, one will be labelled first (mate1), and the other second (mate2). Given that each can have two directions (when aligned against the reference genome), we have 8 possibilities which are shown in the following picture.

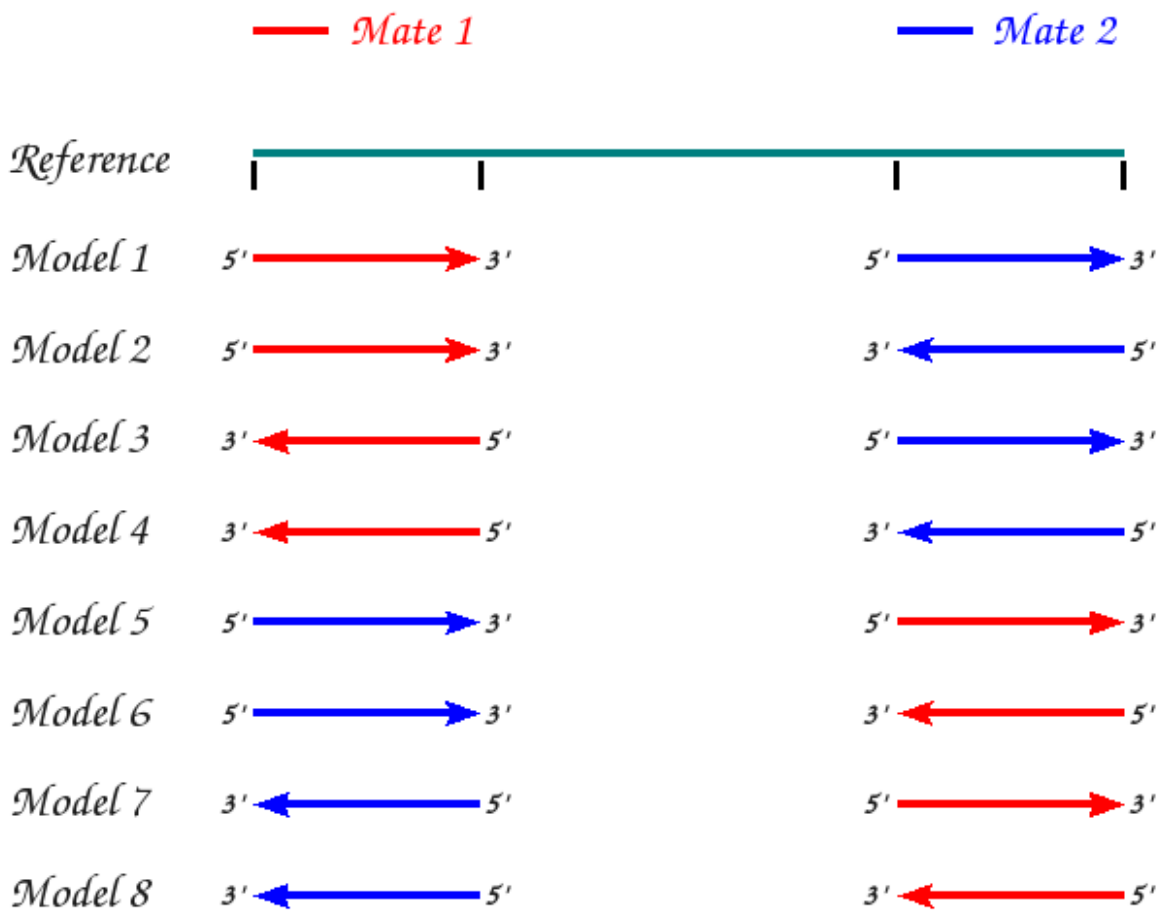

Figure 1.1: Mate-pairs and paired end reads: Relative orientations when aligned to the reference genome under normal conditions

The terms paired-end and mate-pair are sometimes used interchangeably in the sequencing community. In the context of **Avadis NGS** we make the following assumptions:

Vendor	Library	Model
454 Roche	Mate pair	Models 4,5
Illumina	Mate pair	Models 3,7
	Paired end	Models 2,6
ABI SOLiD	Mate pair	Models 4,5
	Paired end	Models 2,6
Ion Torrent	Paired end	Models 3,7

Table 1.1: Paired-end, mate-pair orientations

- **Read-lengths**

  Some next-generation technologies such as Illumina and ABI use fixed-length reads. The reads may experience some minor variation in length post quality filtering process such as read trimming, however for the most part they can roughly be said to have a certain fixed length. Thus each entire read can be considered to provide the same unit of information. On

the other hand, technologies such as 454-Roche, use variable-length reads; one read now may be ten times the length of another.

Chapter 2

ChIP-Seq Analysis

2.1 Introduction

ChIP (chromatin immunoprecipitation) is a technology that isolates a library of target DNA sites that bind to a particular protein in vivo. The ChIP process selectively enriches specific crosslinked DNA-protein complexes using an antibody against the protein of interest. ChIP-Seq combines ChIP with massively parallel sequencing technologies and is used to identify the interaction pattern of a protein with DNA and the surrounding chromatin structure. It provides a view of epigenetic factors influencing gene expression and is complementary to genotype and expression analysis.

Identification of binding sites associated with transcription factors and histone methylation are among the most important applications of ChIP-Seq.

- **Transcription factors**

  The first step in the gene expression process is **transcription**, which involves gene activation by a set of proteins, mainly *transcription factors*. Transcription factors contain one or more DNA binding sites (typically short DNA sequences, $\sim$ 4 - 30 bps in length), which attach to specific sequences of DNA adjacent to the genes that they regulate. They control transcription either by activating or repressing the recruitment of RNA polymerase to these genes.

- **Histone methylation**

  Histone proteins compress the DNA into a compact chromatin package. Modifications on these proteins affect the access of regulatory factors and complexes to the DNA and thus influence gene expression. Histone methylations in particular have been clearly shown to play a key role both in gene activation and repression [5].

Determination of the binding sites from ChIP-Seq data is described briefly in Section 2.1.1. This now raises the question of seeking patterns or motifs that might be common to them, indicating a similar sequence to which the protein of interest would attach. This is done via motif discovery described in Section 2.1.2. Brief descriptions of the associated algorithms used in **Avadis NGS** are provided in Section 2.2.

2.1.1 Binding Site Identification

In the absence of biases, read counts from a DNA sequencing experiment will roughly follow a Poisson distribution with parameters dependent on (i) the total number of reads, (ii) the length of the region and (iii) the length of the sequenced portion of the genome.

In a ChIP-Seq experiment, select regions on the genome are enriched due to proteins that bind to them. The binding sites are fairly small in size, much smaller than the read-size. Only reads to which proteins have bound are retained, all others are discarded. The number of reads vary with the factors outlined earlier; additionally they also reflect (i) the number of sites on the genome to which the studied protein is capable of binding, and (ii) the affinity or efficiency with which the protein binds to a particular site.

At a fundamental level, all algorithms infer binding sites by determining those portions on the genome with higher-than-expected read counts. Duplicate reads is a technological artifacts frequently observed in ChIP-Seq data and must be accounted for while identifying binding sites. A few additional issues unique to ChIP-Seq data are presented below:

- **Repeat regions in the genome**

  Large mammalian genomes are often repetitive in nature. Reads from repeat regions do not align uniquely. The alignment process usually discard reads mapping multiply to the genome. This could result in partially represented peaks, with a significant portion of the peak knocked off. Consequently, one peak might easily be misinterpreted as two or more.

- **Shifted read density on the positive and negative strands**

  In practice, reads from the two strands are observed to be slightly shifted from the precise binding sites. This shift results from the fact that reads represent fragment ends, instead of precise binding sites. The two reads do not typically cover the entire length of the fragment. Consequently, read densities follow a bimodal pattern, as shown in Fig. 2.1 with the mode shifted upstream on the positive strand and downstream on the negative strand. The size of the shift δ, roughly corresponds to the difference between the size of the parent DNA fragment containing the binding site and the size of the ensuing read. Binding site detection algorithms must correct for the shift ensuring that the combined read densities follow a unimodal distribution centered at the binding site.

- **Evaluation of closely-spaced multiple binding sites**

  When two binding sites are very close on the genome, the corresponding data might be clubbed together as evident of only one peak. Data must therefore be evaluated for differentiation into neighbouring peaks.

- **Regional biases in data along the genome**

  Biases in read density profiles can be estimated by comparing control vs. immuno-precipitated sample profiles. Control data provides a numerical background for comparative identification of significantly enriched regions. Using this, an enrichment score relative to the control, can be computed for each binding event. It also provides a mechanism for identification of false positives.

2.1.2 Motif Discovery

Having found a number of candidate binding regions via the peak finding algorithm, the next step is to investigate if they share an underlying sequence signature or **motif**. A motif discovery algorithm sifts through a collection of sequences and returns a set of motif predictions with associated locations on the sequences and confidence scores.

What is a motif

Very simplistically, a motif is a short string of letters $\{A, C, G, T\}$ which occurs more frequently than expected in the sequences of interest, and may be indicative of a common transcription factor binding site. Motifs are not rigidly described by a exact occurrence of specific nucleotides but more typically, by a number of similar, frequently-present words combined into a single flexible representation. Popular motif formats include:

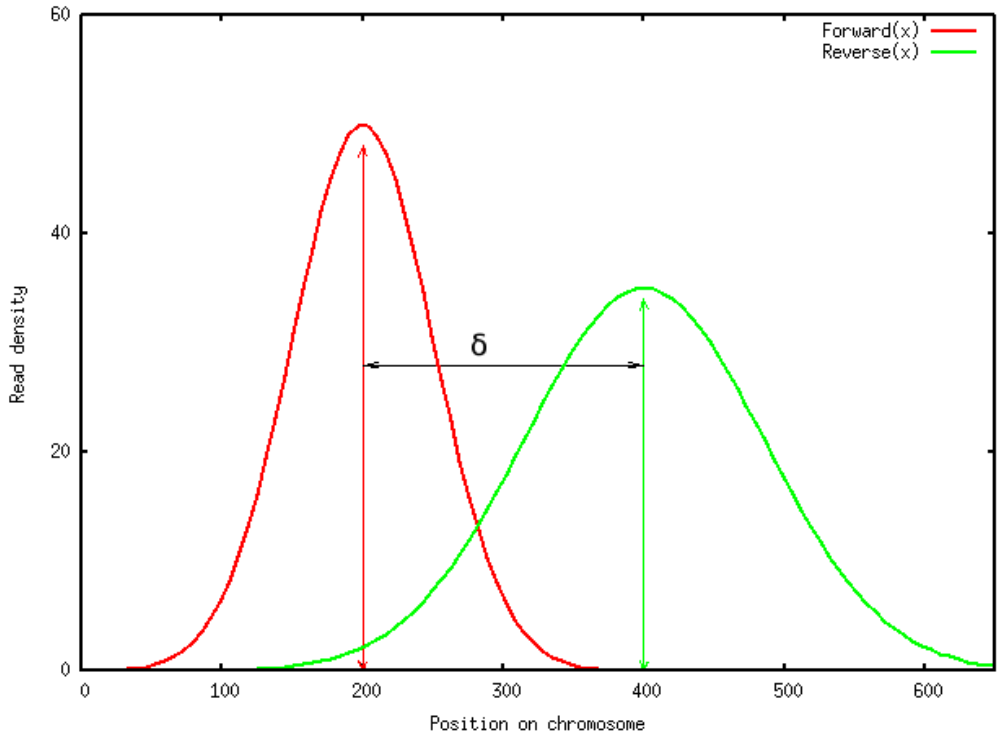

Figure 2.1: Bimodal structure of read densities at the binding site

- A single sequence based format that employs the IUPAC code convention to represent 2 or 3 nucleotide degeneracy, e.g., AYGGNNNNGR, where $Y \equiv \{C, T\}$, $R \equiv \{A, G\}$, and $N \equiv \{A, C, G, T\}$. While this representation has the advantage of being compact, it disregards a fair amount of information.

- A probabilistic approach that views a motif as a short probabilistic string and uses a **Position Weight Matrix (PWM)** for its representation. The PWM matrix gives information of the frequency of each base at each position of the motif. It assumes positional independence, in other words, it assumes that each position contributes independently to the binding site.

  Example:

  Consider a k-length sequence $S = \{S_1, S_2, \ldots, S_k\}$. Each member of the sequence takes values among $\{A, C, G, T\}$ with some associated probability, p, e.g.

$$S_1 = \begin{cases} A & p = 0.9 \\ G & p = 0.1 \end{cases}, S_2 = \begin{cases} A & p = 0.4 \\ C & p = 0.1 \\ G & p = 0.4 \\ T & p = 0.1 \end{cases}, S_3 = \begin{cases} A & p = 0.3 \\ C & p = 0.2 \\ G & p = 0.3 \\ T & p = 0.2 \end{cases}, S_4 = \begin{cases} G & p = 0.4 \\ C & p = 0.1 \\ T & p = 0.5 \end{cases},$$

  etc.

  A simple representation for a k-length probabilistic sequence is given by a Position Weight Matrix (PWM). A PWM provides position-specific letter frequencies through a matrix of size $4 \times k$, with elements $PWM[i][b] = P[S_i = b]$, where $i \in [1, k], b \in \{A, C, G, T\}$. For example, the k-length motif shown earlier could be represented as

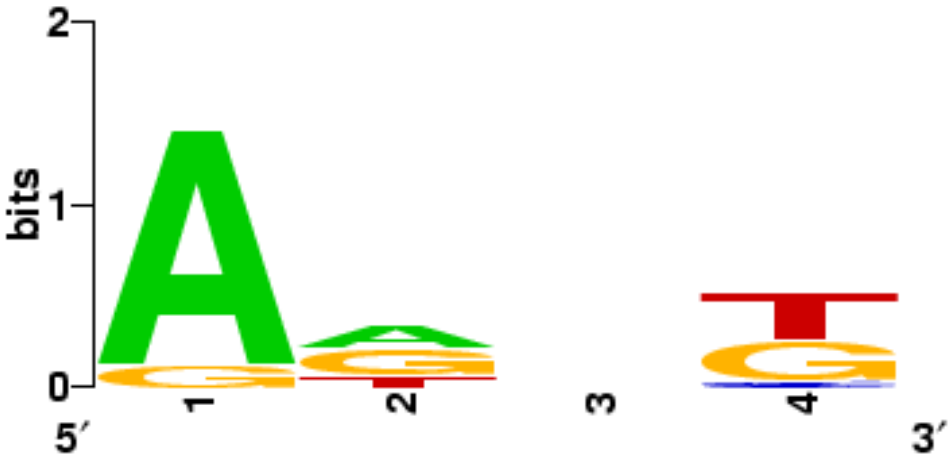

Figure 2.2: Example Motif Logo

$$PWM = \begin{bmatrix} 0.9 & 0.4 & 0.3 & 0.0 & . & . \\ 0.0 & 0.1 & 0.2 & 0.1 & . & . \\ 0.1 & 0.4 & 0.3 & 0.4 & . & . \\ 0.0 & 0.1 & 0.2 & 0.5 & . & . \end{bmatrix} \begin{matrix} A \\ C \\ G \\ T \end{matrix}$$

The PWM is conveniently visualized by a graphical sequence logo representation, where the height of letters at each position is proportional to the information content at that position. In this representation, the relative height of the letters $\{A, C, G, T\}$ indicate the relative presence of the bases at that position, while the total height reflects the sequence conservation.

The PWM illustrated in the above example is shown in Fig. 2.2. Note that the third position of the motif is blank, due to the fact that all the bases are almost uniformly present.

Very often, motifs observed across sequences are in a **spaced dyad** arrangement, consisting of two **words** or short strings, separated by a short stretch of an arbitrary number of unspecified bases called **spacers**, e.g., $W_1xxxxxW_2$ has a spacer of length 5. Spaced dyads may be represented in either of the two formats discussed earlier.

Motif discovery methods
A comprehensive motif search involves searching the entire space of dyads and is computationally immense. Even after restricting the two **words** in the dyads to strings of size between 3 and 6, and the spacers to a maximum size of d nucleotides, the computational complexity remains daunting. Popular motif discovery methods use various techniques to restrict the search and fall into two categories:

1. **Word enumeration:** This technique views motifs in the single-sequence format and counts the number of occurrences of a motif in the sequence data. A sequence is considered an instance of the motif when the number of mismatches between the sequence and the motif falls below a threshold. Motifs with the highest over-representation are selected.

2. **Local search technique:** This technique views motifs as PWMs and involves deterministic motif optimization methods such as Expectation Maximization (EM). The algorithm involves pre-selection of motifs via a number of possible methods, and then iterative refinement of the selection. The primary challenge for this approach is in restricting the complexity of the search via clever initialization.

2.2 Overview of Algorithms

Avadis NGS presents two methods for binding site identification. One is based on the PICS algorithm, [33] and the other is the MACS algorithm, [34], briefly described in Sections 2.2.2 and 2.2.3 respectively. The motif discovery algorithm used is based on GADEM, [23], and is presented in Section 2.2.4. It also includes a simple enriched region identification method that can be used to quickly identify regions with a large number of reads.

2.2.1 Enriched Region Detection

The Enriched Region Detection algorithm is an extremely fast way of identifying regions with high density of reads. It uses a sliding window approach to identify stretches with large numbers of reads. Unlike PICS and MACS it does not check the orientation of reads or the shape of the peaks formed by reads.

A window is considered to be *enriched* in a run without a control sample if the number of treatment reads that fall in the window exceeds the "Minimum reads in window" criteria. For the window to be considered *enriched* in a run with a control sample, the following additional critera needs to be met:

$$\frac{n_t}{n_c}\frac{S_c}{S_t} > e$$

where

- n_t is the number of treatment reads in the window,
- n_c is the number of control reads in the window,
- S_c is the total number of reads in the control sample,
- S_t is the total number of reads in the treatment sample, and
- e is the enrichment threshold

The equation essentially says that if the treatment sample had the same number of reads as the control sample it would still have at least e-fold more reads in this window. Consecutive enriched windows are merged into a single region and the region is saved if the region size and number of reads in it exceed the user specified thresholds.

2.2.2 Peak Detection using PICS

PICS (Probabilistic Inference for ChIP-Seq) [33] uses forward and reverse read count information to estimate peaks associated with binding sites. Closely-spaced, multiply occurring binding sites can be identified through a Bayesian approach which models the DNA fragment length distribution as prior information. PICS accommodates removal of other inherent biases by allowing comparison against a control sample. PICS also provides a **Mappability Profile** which helps remove biases due to genome repetitiveness.

PICS uses t distributions with 4 degrees of freedom to model the start positions of both forward and reverse reads. Distribution parameters include:

- μ, the binding site position,
- δ, the shift between the peaks of the forward and reverse hills, and

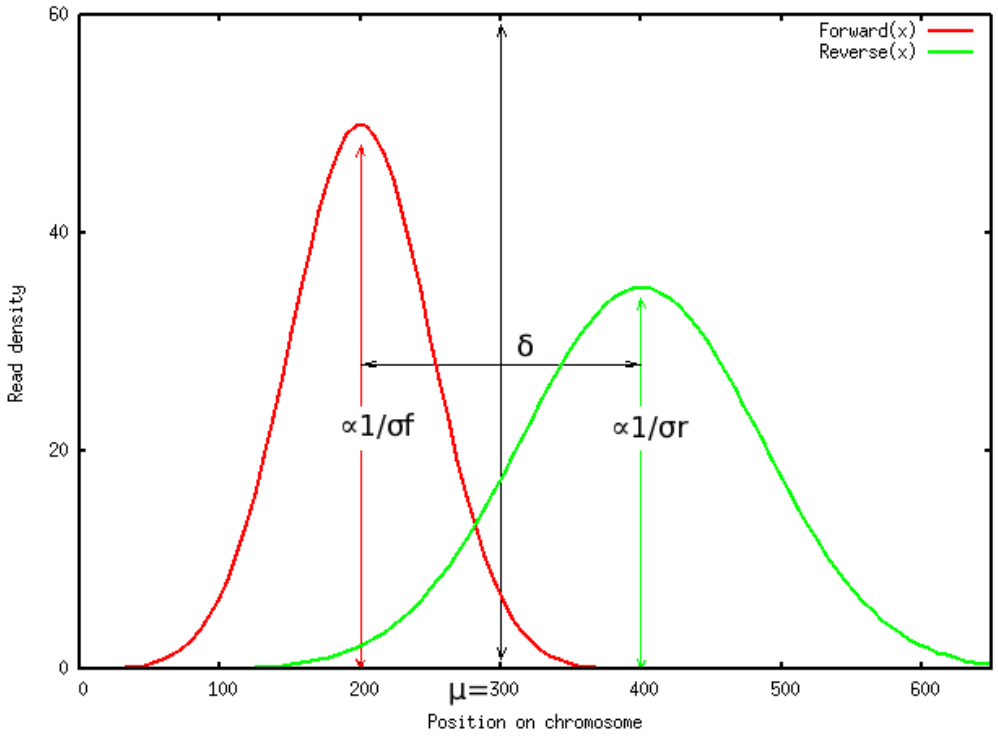

Figure 2.3: Read density at a binding site

- σ_f and σ_r, the variability in length for forward and reverse DNA fragments

depicted in Fig. 2.3. These parameters are estimated through localized measurements of read concentrations. The algorithm accommodates multiply occurring binding sites via weighted mixtures of t distributions. Prior distributions reflect underlying behavior assumptions for parameters δ, σ_f and σ_r. Details regarding the read distributions and the priors are available in Sections 2.3.1.1 and 2.3.1.2.

The PICS algorithm incorporates the following steps:

1. **Inclusion of a user-provided mappability profile:** As seen earlier in Section 2.1.1, repeat regions in a genome create reads that cannot be uniquely mapped. Such regions can be identified purely on the basis of the reference sequence and a corresponding **Mappability Profile**. If a Mappability Profile is provided by the user, PICS discards reads that align to regions demarcated by the Mappability Profile, and uses interpolative techniques to fill in the gaps. See Section 2.3.1.3 for details.

2. **Preprocessing:** PICS follows a sliding window approach, where the read counts aligning to a window are computed across the genome for both forward and reverse reads. Windows with counts falling below a minimum window count threshold (*Minimum Forward & Reverse Read Count in a Window*) are discarded. Overlapping significant windows are merged to eventually create a set of disjoint significant windows or *regions*. Regions with fewer reads than a required threshold (*Minimum Forward & Reverse Read Count in a Region*) are discarded. Each region is assumed to contain one or more binding sites and is individually analysed. Summits in each window, i.e., locations with local maximum read counts are output as predicted binding sites. Further information can be obtained in Section 2.3.1.4.

3. **EM algorithm:** The Expectation-Maximization procedure estimates the model parameters given the observed data. It begins by assuming a fixed number of binding sites k in a

region and an initial distribution for the model parameters, and arrives at a final estimate by progressively iterating two steps: The E-step, and the M-step. The EM algorithm is described in detail in Section 2.3.1.5.

4. **Peak merging:** Peaks with overlapping binding events are detected and merged and their parameters recomputed. Details are provided in Section 2.3.1.6.

5. **Peak filtering:** Binding events with atypical parameter estimates, such as atypical average DNA fragment length estimates or large fragment length variability (hard-coded to be > 200) are filtered out. The filter criteria are elaborated in Section 2.3.1.7.

6. **Computing enrichment scores:** The significance of a binding event is determined based on an enrichment score. Refer to Section 2.3.1.8 for further information.

7. **Estimating the False Discovery Rate (FDR):** When control data is present for comparison, an FDR can be computed as a function of the enrichment score. Section 2.3.1.9 describes the details.

2.2.3 Peak Detection using MACS

The MACS (Model-based Analysis of ChIP-Seq) algorithm described in [34] is a simple yet effective technique for peak detection. It empirically models the shift size of ChIP-Seq reads and attempts to capture local biases in the genome via a dynamic Poisson distribution.

Avadis NGS includes version 1.3.7.1 of the publicly available MACS implementation with portability fixes that were necessary for it run in the **Avadis** framework.

The main steps of the algorithm are outlined below:

1. **Find preliminary peaks**

   The fundamental idea of this step is to identify windows with read count more than m times the count expected from a random read distribution, where m is the enrichment factor. This is achieved by sliding a window of size $2 \times BW$ across the genome, and counting forward and reverse reads that lie in the window. This total count is compared against a threshold $R_{min} = m \times 2 \times BW \times N/G$, where N is the total number of reads and G, the total size of the sequenced genome. All windows with read count greater than the threshold R_{min} are selected.

2. **Model inter-peak shift**

   This step is based on the assumption that the inter-peak shift δ at all binding sites will be approximately the same. It therefore attempts to pool a large number of suspected binding sites and estimate the inter-peak shift from this data. Accordingly, it randomly picks a suitably large number (1000) from the selected windows. In each case, it separates forward and reverse reads and identifies the corresponding peaks. If the forward-read peak is to the left of the reverse-read peak, the reads are aligned according to the midpoint. The reads are then pooled and two peaks are identified as before. The distance between the modes of the peaks gives δ.

3. **Handling duplicates**

   MACS removes duplicate reads in excess of what is warranted by the sequencing depth. It compares the number of duplicate reads against a threshold T_D, and removes the spillover. This threshold is based on the number of duplicates that will correspond to a p-value of 10^{-5} assuming a random genome-wide read distribution. For more details refer to [34].

4. **Correcting for δ**

   The remaining reads are shifted by $\delta/2$; the forward reads towards the right and the reverse reads towards the left.

5. **Finding candidate peaks**

   Candidate peaks are identified by sliding a window of size 2δ across the genome. Significant read enrichment is concluded if the observed read count $N_{2\delta} > j^*$, where

$$j^* = \text{argmin}_j \left[\sum_{i=j}^{\infty} \frac{e^{-2\delta\lambda}(2\delta\lambda)^i}{i!} \leq 10^{-5} \right], \tag{2.1}$$

   and $\lambda = \lambda_{BG} = N/G$, the expected background density of reads.

   Overlapping candidate peaks are first merged. Then each read position is extended upto δ bases from its center, so as to make its final length 2δ. The location with the highest number of reads is the summit and is predicted as the precise binding location.

6. **Correcting for local biases**

   MACS attempts to apply local corrections to each candidate peak by using a dynamic λ_{local} instead of the λ_{BG} estimated from the whole genome.

   - In the absence of a control sample

$$\lambda_{local} = max(\lambda_{BG}, \lambda_{5k}, \lambda_{10k}), \tag{2.2}$$

     where, $L \times \lambda_L$ = number of reads in the **ChIP** sample in a window of size L centred around candidate peak location.

   - In the presence of a control sample

     The total control read count in the control sample is first linearly scaled so as to equal the total ChIP read count.

$$\lambda_{local} = max(\lambda_{BG}, \lambda_{1k}, \lambda_{5k}, \lambda_{10k}) \tag{2.3}$$

     where, $L \times \lambda_L$ = number of reads in the **control** sample in a window of size L centred around candidate peak location.

   MACS provides an option to turn the correction mechanism off. In this case, the $\lambda_{local} = \lambda_{BG}$.

7. **Peak detection**

   The p-value of each candidate peak is called using λ_{local} thus removing false positives due to local biases. Finally, candidate peaks with p-values below a user-defined threshold p-value are called. The ratio between the ChIP-Seq read count and λ_{local} is reported as the fold enrichment.

Note that a couple of options offered by the MACS code have been turned off in **Avadis NGS**. MACS has the ability to produce a saturation table by reporting, at different fold-enrichments, the proportion of sites that can still be detected using 90% to 20% of the reads. MACS also provides an option to skip the initial estimation of parameter δ, and to use a global user-provided value both for δ and λ. Both features are absent in **Avadis NGS**.

2.2.4 Motif Detection using GADEM

GADEM (**G**enetic **A**lgorithm guided formation of spaced **D**yads coupled with an **EM** algorithm for motif discovery) uses word-enumeration techniques to create initial motif models and subsequently refines these by a local search technique, an EM algorithm. An initial subset of motifs is created by randomly combining over-represented words in the data. The subset is modified over various iterations through genetic operations. The selected motifs act as *seeds* to be progressively optimized by the EM algorithm. As the EM algorithm only provides a local optimum, its efficacy depends strongly on the quality of the input motif. Thus, intelligent filtering of motifs combined with local EM optimization help identify good quality candidate motifs in datasets.

The GADEM algorithm:

Preprocessing: To save on runtime, the algorithm first creates a random subset of the initial data. All further motif searches in all subsequent runs are restricted to this subset. However, once motifs are found, their presence is verified in the entire parent data set. See Section 2.3.2.1 for details.

The general steps followed by **one run** of GADEM are outlined below:

1. **Initial population generation:** A list of frequently occurring short words in the subset is created, and ordered according to prevalence. Spaced dyads are generated by randomly choosing two words and then picking the length of the intermediate spacer. An indexed population of such spaced dyads is thus created to the desired size. Section 2.3.2.2 provides further information.

2. **Motif optimization and evaluation:** Each dyad in the indexed population is represented as a PWM. This seed PWM now experiences a number of operations:

   (a) **EM optimization:** Starting with the seed PWM, the EM algorithm toggles between estimating the positions of the motif in the E-step and updating the motif letter frequencies in the PWM in the M-step. After a fixed number of such iterations, it outputs a more refined PWM. Note that the size of the PWM is always preserved through all optimizations; in other words, the optimized motif retains the original length k of the seed spaced dyad. For the particulars of the EM algorithm and examples, refer to Section 2.3.2.4.

   (b) **Scoring and identification of binding sites:** Sequences in the entire dataset are scored using the optimized $4 \times k$ PWM. This is done by considering each k-length substring in the sequences and computing its probability of occurrence under the PWM. For each score, associated p-values are derived representing the probability that a random string of the same length has a better (lower) score. A binding site is declared when the p-value of its PWM score falls below a threshold. Further details are available in Section 2.3.2.5.

   (c) **Fitness evaluation:** The binding sites corresponding to one PWM are collected and aligned and their collective significance is represented by a fitness score. The fitness score represents the efficacy of the motif and incorporates the number of associated binding sites, how strongly the binding sites differ from the background, etc. Fitness scores are relevant in comparing and ranking motifs. Section 2.3.2.6 includes additional information.

   The three steps are repeated for each of the dyads in the population and a comprehensive set of motifs derived.

3. **Population evolution through genetic operations:** Populations evolve using genetic operations through several generations. The best ranked motifs from the current population are retained in the new generation. The rest are supplemented by mutations and crossovers,

further elaborated in Section 2.3.2.7. Each dyad in each generation undergoes the motif optimization and evaluation step.

4. **Post-processing:** At the end of the desired number of generations, motifs with fitness values less than a threshold are reported. These motifs are checked for prevalence in the data, relevance of information content, and redundancy before being finalised.
   (a) **Prevalence in the data:** The motifs are required to satisfy a minimum frequency of occurrence in the data.
   (b) **Motif trimming:** A selected PWM might be significant overall, but yet retain a small number of redundant locations. The motif trimming step attempts to trim away these locations. Extension and trimming operations are performed at each binding site corresponding to a motif.
   (c) **Motif filtering:** The set of motifs is checked for redundancy. Motifs judged close relatives are only represented once.

   Details regarding these steps are available in Section 2.3.2.8. The final set of motifs along with their binding sites are output.

At the end of one run, binding sites of all declared motifs are masked in the input data and the entire process rerun for further motif discovery. The algorithm stops when no more motifs can be found, or when a maximum number of runs has been reached. See Section 2.3.2.9 for more information.

2.3 Mathematical Details of the Algorithms

2.3.1 PICS Algorithm

As discussed in Section 2.2.2, PICS is a peak-finding algorithm that takes a number of aligned reads as input and outputs candidate binding sites. PICS uses a Bayesian formulation, with DNA fragment length distribution as prior information to model read locations across the genome. Mathematical details regarding the model and priors as well as the working of the EM algorithm used to estimate the model parameters are provided here.

2.3.1.1 Modeling Read Distributions

For each candidate region, the read distribution for forward and reverse reads is modeled. PICS uses an approach that models distributions of fragment ends, rather than directly modeling sequence counts as is usually done. Let f_i denote the start position of the i^{th} forward read, $i = 1, ..., n_f$ and r_j denote the start position of the j^{th} reverse read, $j = 1, ..., n_r$. Start positions of forward and reverse reads are each modeled in this manner independently with different parameters.

PICS uses t distributions for this modeling. A univariate t-distribution of a random variable X with parameters ν, μ, σ^2 denoted by $t_\nu(\mu, \sigma^2)$, is defined as follows:

$$(X|U = u) \sim \mathcal{N}(\mu, \frac{\sigma^2}{u})$$

where

$$U \sim \text{gamma}(\frac{\nu}{2}, \frac{\nu}{2}),$$

and its density at x denoted by $t_\nu(x; \mu, \sigma^2)$ is

$$\frac{\Gamma(\frac{\nu+1}{2})}{\sqrt{\nu\pi}\,\Gamma(\frac{\nu}{2})}(1 + \frac{(x-\mu)^2}{\nu})^{-\frac{\nu+1}{2}}.$$

For a single binding event in a region, the start positions f_i and r_j are modeled by t-distributions with 4 degrees of freedom, with

$$f_i \sim t_4\left(\mu - \delta/2, \sigma_f^2\right) \quad \text{and } r_j \sim t_4\left(\mu + \delta/2, \sigma_r^2\right)$$

where μ represents the binding site position, δ is the shift between the peaks of the forward and reverse hills (roughly equal to the average DNA fragment size), and σ_f^2 and σ_r^2 measure the variability in DNA fragment lengths.

A similar formulation represents multiple binding events in a region, but using weighted mixture models. In this case,

$$f_i \sim \sum_{k=1}^{K} w_k t_4\left(\mu_k - \delta_k/2, \sigma_{f,k}^2\right) \quad \text{and } r_j \sim \sum_{k=1}^{K} w_k t_4\left(\mu_k + \delta_k/2, \sigma_{r,k}^2\right),$$

where the subscript k represents the k^{th} binding site position in the region, and weight w_k is the corresponding relative proportion of reads and K represents the total number of binding events. The model is completely characterized by $5K-1$ parameters, $\Theta \equiv \{w_k, \mu_k, \delta_k, \sigma_{f,k}, \sigma_{r,k}\}$, $k = 1, 2, \ldots, K$ (under the constraint that the weights sum to 1).

2.3.1.2 Prior Distributions

PICS estimates the model parameters Θ outlined in Section 2.3.1.1 assuming certain underlying behavior for the DNA fragment distribution. Priors defined include

1. independent but identical inverse gamma priors for σ_f^2 and σ_r^2, viz., $\sigma_f^2, \sigma_r^2 \sim \mathcal{IG}(\alpha, \beta)$.
2. prior for the shift given by $(\delta_k | \sigma_f^2, \sigma_r^2) \sim \mathcal{N}(\xi, \rho^{-1}/\sigma_f^{-2} + \sigma_r^{-2})$, where ξ is the mean fragment length, and ρ is a scaling factor for the corresponding spread.

Setting parameters for priors: The parameters α, β, ξ, and ρ can be set through the *Tools* → *Options* menu, under the *ChIP-seq Analysis Algorithms* tab, *PICS* selection. These parameters can be tuned according to the experimentally obtained length distribution of the DNA fragments. In PICS, the prior parameters α, β, ρ are set to values which make the prior distribution non-informative; large values of the parameter β result in a fairly flat (non-informative) prior for σ_f^2, and σ_r^2. The prior parameter ξ is set based on data as described below.

For a DNA fragment starts at position f_i and ends at r_j, the length is given by $r_j - f_i + 1$. As f_i and r_j are assumed independent, the variance in the fragment length can be approximated by $\sigma_f^2 + \sigma_r^2$. The expected variance of the DNA fragment length is then $2\beta/(\alpha - 1)$.

The shift δ_k between the forward and reverse read distributions is approximated by the diference in the average length of the DNA fragment and the average read length. The prior parameter ξ has therefore been set at the value of the (average observed fragment length − average observed read length).

For example, in [33] the parameters chosen are $\alpha = 20$, $\beta = 40000$, $\xi = 175$, and $\rho = 1$. This corresponds to normally distributed DNA fragment lengths with mean = 175 bps, and (expected) standard deviation given by

$$\begin{aligned} SD &= \left(\frac{\beta}{2\alpha - 1}\right)^{1/2} \\ &= \left(\frac{40000}{39}\right)^{1/2} \\ &\approx 32 \end{aligned}$$

These values were based on DNA fragment sizes on the order of 100-250 bps.

2.3.1.3 Mappability Profile

A **Mappability Profile** helps to identify problematic regions that might cause multiply mapping reads. This is constructed by considering the set of all possible sequences of a length L arising from the genome. The number of times a sequence repeats in the genome is then counted. Regions with number of repeats greater than one are marked. The Mappability Profile depends on the length L of the sequence; longer sequences tend to be less repetitive. Further, the Mappability Profile is only dependent on the reference genome and so it may be generated irrespective of the experimental data. PICS allows the user to provide their own mappability profile.

2.3.1.4 Preprocessing

PICS creates read count profiles by summing localized read counts over disjoint candidate regions. The details of the counting procedure are as follows:

- All duplicate reads are removed.
- For each sliding window of length w the number of forward strand reads in the left half and the reverse strand reads in the right half are counted. The window is shifted by s and the process repeated.
- The read count per window is checked for both forward and reverse reads. Windows with counts falling below a minimum window count threshold are discarded. Overlapping windows are merged to obtain disjoint candidate regions. The read count per region is also checked against a minimum regional count threshold for both forward and reverse reads. Regions with fewer reads than the required threshold are discarded.

2.3.1.5 Expectation-Maximization (EM) Algorithm

EM is an iterative procedure for maximum likelihood estimation of parameters in problems where the observed data lead to a difficult maximization problem and where an augmented dataset (including missing or latent data) can make the maximization problem more tractable. The algorithm can be modified to deal with penalized likelihood maximization arising in Bayesian estimation problems like the one here. General details of the EM algorithm can be found in Appendix B.

The model used in PICS described above is a mixture model discussed in Appendix B for resolving which the EM algorithm is a suitable technique. A complete-data problem that will facilitate the application of the EM algorithm is formulated by the addition of latent data in the form of the binding site affiliation for each read, denoted by a binary vector variable $Z_{di} = (Z_{di1}, Z_{di2}, \ldots, Z_{diK})$, $d = f, r$, where the k^{th} component Z_{dik} is 1, if the read arises from site k and the other components are 0.

In this particular problem with the t-distribution, treating U defined above as an additional latent variable makes the EM algorithm application simpler making the M-step a weighted least-squares algorithm. Thus besides $\boldsymbol{Z}_{di}$, $\boldsymbol{U}_{di} = (U_{di1}, U_{di2}, \ldots, U_{diK})$ for $i = 1, 2, \ldots, K$ are the latent variables used to define the complete-data problem.

In order to write down the complete-data log-likelihood, the following distributions are used:

$$(d_i | U_{dik} = u_{dik}, z_{dik} = 1, \mu_k, \sigma^2_{dk}) \sim \mathcal{N}(\mu_{dk}, \frac{\sigma^2_{dk}}{u_{dik}}),$$

$$(U_{dik} | z_{dil} = 1) \sim \text{gamma}(2, 2),$$

$$\boldsymbol{Z}_{di} \sim \text{multinomial}(1; w_1, w_2, \ldots, w_K).$$

The penalized log-likelihood is then written in terms of these distributions as the sum of the log-likelihood based on complete-data and the logarithm of the prior distributions. In view of the multinomial nature of the $\boldsymbol{Z}$ distribution and the nature of the appearance of u in the normal distribution's variance, both the Z's and U's appear in linear forms in the penalized complete-data log-likelihood and so it becomes fairly easy to compute the conditional expected value of the penalized log-likelihood; only expected values of U's and Z's need to be evaluated.

Suggested initial estimates for model parameters to start the EM iterations are described in Section 2.3.1.2. Users have the option to reset these values.

The EM algorithm alternates between computing the expectation of the log-likelihood of the data (under the current parameter estimate) in the E-step and maximizing the log-likelihood with respect to the parameter space in the M-step.

These expected values are:

$$\tilde{z}_{dik} = \frac{w_k t_4(d_i; \mu_{dk}, \sigma^2_{dk})}{\sum_{k'=1}^{K} w_{k'} t_4(d_i; \mu_{dk'}, \sigma^2_{dk'})}$$

$$\tilde{u}_{dik} = \frac{5}{4 + \frac{(d_i - \mu_{dk})^2}{\sigma^2_{dk}}}.$$

This then is the E-step.

For the M-step, a conditional approach is used, whereby maximization is done first with respect to the w, μ, δ's conditional on (σ^2_f, σ^2_r) and then maximization is done conditional on the w, μ, δ values obtained. This is called the Expectation--Conditional-Maximization (ECM) algorithm. Please refer to [33] for formulas based on this approach for the M-step.

Interpolating missing reads: Reads from repeat regions on the genome are identified by the marked regions in the Mappability Profile and are discarded. The E-step of the EM algorithm now breaks up the log-likelihood over the marked sections of the candidate region and recalculates using appropriately interpolated read counts.

The algorithm computes parameter estimates $\hat{\Theta}_k$ for each value of $k, k = 1, 2, \ldots k_{MAX}$. The maximum value k_{MAX} can be modified by changing the *Max Number of Peaks per Region* under the *ChIP-seq Analysis Algorithms* tab, *PICS* selection from the *Tools* → *Options* menu. The default value is 5 peaks, and the algorithm requires the values to lie between 1 and 15.

The appropriate number k of binding sites is chosen by computing the Bayes Information Criterion (BIC) for each value of k from 1 to k_{MAX} and selecting k as that value for which the criterion is the largest. The BIC criterion is 2×the conditional expected value of the log-likelihood at the converged parameter values $-(5k-1)log(n_{f0} + n_{r0})$, where n_{f0}, n_{r0} are the number of forward and reverse reads respectively. For more information on BIC, see Appendix A.

For more details of the PICS algorithm, see [33].

2.3.1.6 Peak Merging

Peaks with overlapping binding events are detected and merged and their parameters recomputed. Regions with multiple binding events, $K > 1$, are rechecked for overlapping binding events as follows: Each pair of adjacent peaks with means μ_i and μ_{i+1}, $(\mu_i < \mu_{i+1})$, and variances σ^2_i and σ^2_{i+1} meeting the condition $\mu_i + \sigma_i \leq \mu_{i+1} - \sigma_{i+1}$ are merged. The parameters of merged binding events are computed by solving moment matching equations given in [33].

2.3.1.7 Peak Filtering

Binding events with atypical parameter estimates, such as atypical average DNA fragment length estimates or large fragment length variability are filtered out. Events satisfying either of the two conditions listed below are removed:

1. $\delta_{\min} < \delta < \delta_{\max}$ (where $\delta_{\min}$ and $\delta_{\max}$ are user-given or default values); or
2. $\sigma_f, \sigma_r < 200$

2.3.1.8 Scoring

To decide the statistical significance of a binding event, it is accorded an enrichment score.

- **Single sample case:** The EM algorithm delineates the regions where the peak has been detected. The region around the mean μ_f, expected to contain 90% of the forward reads, corresponding to $\mu_f \pm 2.13\sigma_f$ is considered. Forward read counts F_{ChIP} are computed for this region. Similarly, the reverse read counts R_{ChIP} are obtained from the reverse peak. The minimum of the forward and reverse counts gives the binding site enrichment score.
- **Two-sample case:** Read counts are computed for the ChIP sample as outlined in the single sample case. For the control sample, the corresponding reads are counted, using the regions defined in the ChIP sample. Ratios of the forward read counts (ChIP/Control) are computed, so also ratios of the reverse read counts. The enrichment score is the minimum of these ratios multiplied by a normalization factor, which accounts for the difference in the total number of reads in each case. Thus the relative enrichment score is given by

$$\frac{N_{control}}{N_{ChIP}} \cdot \min\left\{\frac{F_{ChIP}+1}{F_{Control}+1}, \frac{R_{ChIP}+1}{R_{Control}+1}\right\}$$

2.3.1.9 Estimating the False Discovery Rate (FDR)

The FDR is computed as a function of the enrichment score, and only in the presence of control data. The computation involves flipping the ChIP and Control samples and redoing the entire analysis. For a particular score x, the number of events scoring x or higher are counted in both cases. The count in the flipped case n_0, divided by the corresponding original count n gives the q-value.

2.3.2 GADEM Algorithm

The motif-searching tool GADEM used in **Avadis NGS** was introduced in Section 2.2.4. This section contains details of the algorithm including mathematical representations, dependence on configurable parameters and a few illustrative examples. It is advisable to read the introductory material in Section 2.2.4 before proceeding further. Further information is available in [23].

2.3.2.1 Preprocessing

Motifs are embedded in larger background sequences which are more random in nature. As motifs are not yet known, a **Background Probability Distribution** is derived using the entire set of sequences. This is used throughout in evaluating whether a sequence was part of the background or part of the motif.

GADEM spends a considerable amount of time in the EM procedure; this increases significantly with the number of candidate regions to be analysed. Therefore, the EM procedure is restricted to a random subset of the input sequences, thus limiting run-time, while still yielding most motifs of interest. The fraction of the entire data set that is considered during the motif search can be configured by setting the variable *Fraction of Sequences to Sample* under the *Motif-finding* selection of the *ChIP-seq Analysis Algorithms* tab. This may be accessed from the *Configuration Dialog* box under the *Tools* → *Options* menu. By default 50% of the data is selected; however, the sampled fraction can lie anywhere between 50% to 100% of the entire data.

Note that the subset initially selected, remains common to all the subsequent runs.

2.3.2.2 Creation of a Population of Spaced Dyads

To create the initial population of motifs as seeds for the EM, flexible combinations of words and spacer length are used.

1. All possible non-overlapping k-mers (short DNA words) of length $k : k_{MIN}$ to k_{MAX} present in the sampled subset are counted. For each k-mer S, an associated z-score is computed according to

$$z(S) = \frac{N(S) - E(S)}{\sigma(S)}$$

   where
   $N(S) \equiv$ number of occurrences of the k-mer S,
   $E(S) \equiv$ expected number of occurrences of S, and
   $\sigma(S) \equiv$ estimate of the standard deviation of occurrences of S, based on the background distribution estimated from the entire subset, assuming independence between positions.

2. The k-mers are rank-ordered by their z-scores; a high z-score is suggestive of participation in motif(s). Of the remaining, only N_k top-ranked k-mers are selected, with $N_3 = 20$, $N_4 = 40$, $N_5 = 60$, and $N_6 = 100$.

3. A spaced dyad is generated by independently creating two words and a spacer. A word is created by first randomly selecting its length k from the set $\{k_{MIN}, \ldots, k_{MAX}\}$. Next, a word from the set of N_k k-mers is chosen with probability proportional to the z-score. The width of the spacer is randomly chosen between d_{MIN} and d_{MAX}.

4. An indexed population of such spaced dyads is thus generated to the desired size R.

2.3.2.3 Conversion of Spaced Dyad to PWM

Having created a desired population of spaced dyads, a k-length spaced dyad is selected from the population. As described earlier, it can be represented by a $k \times 4$ dimensional PWM. Positions in the PWM corresponding to the words take value 0 for absent nucleotides and 1 for present nucleotides. The positions corresponding to the spacer all take equal values of 0.25. Thus each column of the PWM corresponds to one position on the spaced dyad, and sums to 1.0.

Example: Consider a spaced dyad constructed with two words $W_1 = CCAG$, a 4-mer and $W_2 = GTG$, a 3-mer, with a spacer of length 2 separating the two. The corresponding 9-base string $CCAGxxGTG$ can be represented by the PWM

$$PWM = \begin{array}{c} \begin{matrix} S_1 & S_2 & S_3 & S_4 & S_5 & S_6 & S_7 & S_8 & S_9 \end{matrix} \\ \begin{bmatrix} 0 & 0 & 1.0 & 0 & 0.25 & 0.25 & 0 & 0 & 0 \\ 1.0 & 1.0 & 0 & 0 & 0.25 & 0.25 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.0 & 0.25 & 0.25 & 1.0 & 0 & 1.0 \\ 0 & 0 & 0 & 0 & 0.25 & 0.25 & 0 & 1.0 & 0 \end{bmatrix} \end{array} \begin{matrix} \\ A \\ C \\ G \\ T \end{matrix}$$

2.3.2.4 Expectation Maximization (EM) Algorithm

A brief description of the general EM algorithm is given in Appendix B. In the context of motif discovery, the observed data are the DNA sequences identified by PICS, the parameters to be estimated are the motifs represented by the PWM, and the latent data are the start positions of the motif in the data. Note that the motif identification would be much simpler given the start positions, and vice versa. Starting with the seed PWM, the EM algorithm alternates between estimating the start positions of the motif in the E-step and updating the motif letter frequencies in the PWM in the M-step, and outputs a more refined PWM after a fixed number of iterations. This number can be set by modifying the *Number of EM Steps* parameter under the *ChIP-seq Analysis Algorithms* tab, *Motif-finding* selection, from the *Tools* → *Options* menu, and takes default value 50. It is constrained to lie between 50 and 100.

E-step:
The EM algorithm considers each instance of a motif present in an input data sequence as a k-length substring or a k-mer, where k is the length of the seed spaced dyad. The probability of the target substring being a motif is calculated in two parts: The first calculates the probability of the substring having been sampled from a position-specific distribution over letter, in other words the PWM, and the second computes the probability of the remaining letters in the sequence as being derived from a fixed background distribution. The probability so calculated is normalized by dividing by the sum of such probabilities computed for each and every k-length substring in the parent data sequence.

M-step:
The k-mers with their corresponding weights are combined to generate a new PWM.

Mathematical details of the EM algorithm as applied in GADEM are given in Section 2.3.2.11.

Example:
We perform one iteration to illustrate the details of the EM algorithm. Only two DNA sequences are considered in the interests of brevity.

$$\begin{array}{cc} & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \end{matrix} \\ S_1 = & \begin{matrix} G & C & C & A & A & C & T & A & C & G & T \end{matrix} \\ S_2 = & \begin{matrix} A & G & T & G & T & G & C & C & G & A \end{matrix} \end{array}$$

The background probabilities are assumed to be given by $BP = \begin{matrix} \text{A} & \text{C} & \text{G} & \text{T} \end{matrix}$
$\begin{bmatrix} 0.23 & 0.26 & 0.26 & 0.25 \end{bmatrix}$

The spaced dyad $CCAGxxGTG$, presented earlier in the earlier example, is used to seed the EM algorithm to find 9-mer motifs in the 2 sequences. The PWM is altered slightly and all 0 probabilities replaced by 0.01. Thus the seed PWM is

$$\text{PWM} = \begin{array}{c} \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{matrix} \\ \begin{bmatrix} 0.01 & 0.01 & 0.97 & 0.01 & 0.25 & 0.25 & 0.01 & 0.01 & 0.01 \\ 0.97 & 0.97 & 0.01 & 0.01 & 0.25 & 0.25 & 0.01 & 0.01 & 0.01 \\ 0.01 & 0.01 & 0.01 & 0.97 & 0.25 & 0.25 & 0.97 & 0.01 & 0.97 \\ 0.01 & 0.01 & 0.01 & 0.01 & 0.25 & 0.25 & 0.01 & 0.97 & 0.01 \end{bmatrix} \end{array} \begin{matrix} \\ A \\ C \\ G \\ T \end{matrix}$$

The first step lists all 9-mers present in the 2 data sequences. The weights for each of the sequences are calculated using the PWM and the background probabilities.

For example, the probability that the first substring GCCAACTAC from the first sequence is sampled from the given PWM is given by

$$\begin{aligned} P &= PWM[1][G] \cdot PWM[2][C] \cdot PWM[3][C] \cdot \ldots \cdot PWM[8][A] \cdot PWM[9][C] \\ &= 0.97 \cdot (0.01)^6 \cdot (0.25)^2 \\ &= 6.0625 \times 10^{-13} \end{aligned}$$

The probability of the remainder of the first sequence S_1 under the background probability distribution is $BP[G] \cdot BP[T] = 0.065$. Thus the probability that the motif starts from position 1 of the first sequence is given by the product of the two terms and is 3.94×10^{-14}. This probability normalized by the probabilities for all substrings in the parent sequence gives the final weight.

Thus, for the substring GCCAACTAC, the weight is given by

$$\begin{aligned} W &= \frac{3.94 \times 10^{-14}}{3.94 \times 10^{-14} + 5.75 \times 10^{-8} + 6.36 \times 10^{-12}} \\ &= 6.85 \times 10^{-15} \end{aligned}$$

where the probabilities are as shown in the table.

#	Start postion	Substring									Probability motif	Weight
	in sequence, i	1	2	3	4	5	6	7	8	9	starts at position i	
1	1 in S_1	G	C	C	A	A	C	T	A	C	3.94×10^{-14}	7×10^{-15}
2	2 in S_1	C	C	A	A	C	T	A	C	G	5.75×10^{-8}	0.999
3	3 in S_1	C	A	A	C	T	A	C	G	T	6.36×10^{-12}	10^{-12}
4	1 in S_2	A	G	T	G	T	G	C	C	G	2.16×10^{-11}	0.999
5	2 in S_2	G	T	G	T	G	C	C	G	A	2.3×10^{-15}	10^{-4}

The weights of all 5 substrings can now be used in the M-step of the algorithm to update the PWM. The development of the PWM is fairly intuitive, we provide the computation of the first column as an illustration.

Looking at the nucleotides in the first column, we see that G is represented twice by substrings of weight 7×10^{-15} and 10^{-4}. Thus the probability of the first position in the motif having nucleotide G is proportional to the sum of the weights $7 \times 10^{-15} + 10^{-4}$. The first column is therefore given by

$$\begin{aligned} PWM[1] &= \begin{bmatrix} 0.999 \\ 0.999 + 10^{-12} \\ 7 \times 10^{-15} + 10^{-4} \\ 0 \end{bmatrix} \begin{matrix} A \\ C \\ G \\ T \end{matrix} \\ &= \begin{bmatrix} 0.999 \\ 0.999 \\ 10^{-4} \\ 0 \end{bmatrix} \begin{matrix} A \\ C \\ G \\ T \end{matrix} \end{aligned}$$

The entire PWM can be thus populated and is given by

$$PWM = \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{matrix}$$
$$\begin{bmatrix} 0.999 & 10^{-12} & 0.999 & 0.999 & 7 \times 10^{-15} & 10^{-12} & 0.999 & 7 \times 10^{-15} & 10^{-4} \\ 0.999 & 0.999 & 7 \times 10^{-15} & 10^{-12} & 0.999 & 10^{-4} & 0.999 & 1.998 & 7 \times 10^{-15} \\ 10^{-4} & 0.999 & 10^{-4} & 0.999 & 10^{-4} & 0.999 & 0 & 10^{-4} & 1.998 \\ 0 & 10^{-4} & 0.999 & 10^{-4} & 0.999 & 0.999 & 7 \times 10^{-15} & 0 & 10^{-12} \end{bmatrix} \begin{matrix} A \\ C \\ G \\ T \end{matrix}$$

Normalizing the matrix gives the final PWM:

$$PWM = \begin{bmatrix} 0.499 & 5 \cdot 10^{-13} & 0.499 & 0.499 & 3.5 \cdot 10^{-15} & 5 \cdot 10^{-13} & 0.499 & 3.5 \cdot 10^{-15} & 5 \cdot 10^{-5} \\ 0.499 & 0.499 & 3.5 \cdot 10^{-15} & 5 \cdot 10^{-13} & 0.499 & 5 \cdot 10^{-5} & 0.499 & 0.999 & 3.5 \cdot 10^{-15} \\ 5 \cdot 10^{-5} & 0.499 & 5 \cdot 10^{-5} & 0.499 & 5 \cdot 10^{-5} & 0.499 & 0 & 5 \cdot 10^{-5} & 0.999 \\ 0 & 5 \cdot 10^{-5} & 0.499 & 5 \cdot 10^{-5} & 0.499 & 0.499 & 3.5 \cdot 10^{-15} & 0 & 5 \cdot 10^{-13} \end{bmatrix}$$

This PWM is used in the next iteration to recalculate probabilities of the 5 substrings and the associated weights.

2.3.2.5 Scoring Motifs: Assigning p-values

The EM-optimized PWM represents a k-length candidate motif. Associated binding sites are now deduced by scanning each input data sequence for the occurrence of this motif. This is done by considering each k-length substring in the sequences and computing its associated probability of occurrence under the PWM. For a substring $s_1, s_2, \ldots, s_k$ the probability of occurrence is given by

$$P[s] = \prod_{u=1}^{k} PWM[u][s_u].$$

Scores are assigned to each substring based on the absolute values of the log probabilities as follows:

$$Score(s) = \sum_{u=1}^{k} SM[u][s_u]$$

where elements of the Score Matrix $SM[u][s_u]$, for $u = 1, 2, \ldots, k$ are simply scaled, log-transformed versions of the optimized PWM, rounded off to the nearest integer, and are given by

$$SM[u][s_u] = \text{integer value}\,[200 \cdot |\log PWM[u][s_u]|]$$

The higher the probability of occurrence of a substring, the lower the score.

The significance of a subsequence with score z can be realized through its p-value, i.e., the probability that a random string of the same length k has score **at most** z. A subsequence in the data is declared a binding site, when the p-value of the observed score falls below a threshold of 0.05. Details regarding computation of p-values can be found in Section 2.3.2.12.

Note that the sequences checked for the presence of the motif are not restricted to the fractional subset sampled for the EM procedure, but include the entire set of candidate regions presented by the peak finding algorithm.

2.3.2.6 Fitness Evaluation

A list of corresponding binding sites is now available for each EM-optimized PWM. The *goodness* of this motif depends on a various issues, such as, the number of associated binding sites, how strongly the binding sites match the motif, how different the motif is from the background, etc. Motifs are evaluated and a **Fitness Score** assigned to each, by which they may be ranked and compared.

This is done by first aligning all binding sites corresponding to a motif and then computing a Log Likelihood Ratio (LLR) score computed, based on the relative entropy score of the alignment:

$$LLR = M \sum_{l=1}^{k} \underbrace{\sum_{b=\{A,C,G,T\}} f_{l,b} \cdot \log\left(\frac{f_{l,b}}{p_b}\right)}_{\text{relative entropy } \mathcal{D}(f_l||p)}$$

where
M is the number of binding sites in the alignment,
k is the size of the PWM
$f_{l,b}$ is the frequency of base b at position l of the alignment and
p_b is the background frequency of base b, estimated from the entire data.

Relative entropy, $\mathcal{D}$, also known as the Kullback-Leibler divergence, is commonly used as a measure of the difference between two probability distributions (in this case $f_{l,b}$ and p_b). It is always non-negative, and takes value 0 when the two distributions are identical. Therefore, the LLR is a non-decreasing function of k. LLR also increases directly with M.

Example: Consider the following list of $M = 10$ binding sites associated with a given motif of length $k = 9$ bases.

		1	2	3	4	5	6	7	8	9
BS_1	=	G	C	T	A	A	C	A	G	T
BS_2	=	A	G	T	A	T	G	C	G	G
BS_3	=	G	C	C	A	G	A	T	A	T
BS_4	=	G	G	G	G	T	A	G	G	T
BS_5	=	G	C	C	G	C	C	T	G	C
BS_6	=	A	C	C	G	C	G	C	C	G
BS_7	=	C	C	T	A	A	C	G	G	T
BS_8	=	A	G	T	G	T	G	C	G	T
BS_9	=	C	C	C	A	G	T	C	C	G
BS_{10}	=	A	C	C	T	A	G	A	G	G

The background probabilities are given by $BP = \begin{array}{c} \text{A} \quad \text{C} \quad \text{G} \quad \text{T} \\ [\ 0.23 \quad 0.26 \quad 0.26 \quad 0.25\ ] \end{array}$

The position-specific frequencies $f_{l,b}$ are calculated by counting the occurences of base b in the lth position of the alignment. For example, for the 1st position, A and G occur 4 times each, and C occurs twice. Thus $f_{1,A} = 0.4$, $f_{1,C} = 0.2$, $f_{1,G} = 0.4$ and $f_{1,T} = 0$. The Kullback-Leibler divergence at the 1st position is given by

$$\begin{aligned}
\mathcal{D}(f_1||p) &= \sum_{b=\{A,C,G,T\}} f_{l,b} \cdot \log\left(\frac{f_{l,b}}{p_b}\right) \\
&= 0.4 \cdot \log\left(\frac{0.4}{0.23}\right) + 0.2 \cdot \log\left(\frac{0.2}{0.26}\right) + 0.4 \cdot \log\left(\frac{0.4}{0.26}\right) + 0 \\
&= 0.221 - 0.052 + 0.172 \\
&= 0.341
\end{aligned}$$

Similarly calculating $\mathcal{D}(f_l||p), l = 2, \ldots, 9$, we can compute $LLR = 10 \cdot \sum_{l=1}^{9} \mathcal{D}(f_l||p)$.

As with the fitness scores, the significance of an LLR score, can be assessed through its associated p-value, i.e., the probability of observing an LLR score or higher under the (null) hypothesis that the distribution of the letters in each column follows an independent multinomial distribution. The p-value computation is based on the approach used by MEME; details can be found in [3].

Accordingly, each spaced dyad in the population is assigned a fitness score FS, given by

$FS = \log(\text{p-value of LLR} \times \text{number of data sequences})$

2.3.2.7 Genetic Operations

Once the entire population of spaced dyads has been evaluated and motifs derived, a new generation of candidate spaced dyads is evolved via two genetic operations: **Mutation** and **Crossover**. The

motifs from the current generation are ranked according to fitness score, and the top 10% are retained in the next generation. The remaining candidates are generated via a series of operations:

1. The operation, mutation or crossover is randomly selected.
2. Candidate(s) are selected from the entire current generation with probability specified by the fitness score.
3. The new spaced dyad(s) are created via the genetic operation.

Mutation: A candidate spaced dyad can undergo mutation by replacing either of its first word, number of spacers or second word. One of these are chosen at random for replacement. The replacement is performed according to the word/spacer selection rules outlined under item 4 of Section 2.3.2.2.

Crossover: Two spaced dyads undergo crossover by exchanging one of the three components of the spaced dyad, i.e., the first word, number of spacers or second word, thus creating two new candidates.

As most unique motifs remain unchanged from generation to generation, the number of GA cycles is limited to a maximum of 5. This can be programmed through the *Configuration Dialog* box accessible through the *Tools* → *Options* menu. The *Number of Generations* parameter under the *ChIP-seq Analysis Algorithms* tab, *Motif-finding* selection is constrained to a maximum of 5 generations, which is also the default value.

2.3.2.8 Post-processing

At the end of all GA cycles in a run, a set of motifs with fitness scores below a certain threshold are obtained. These motifs are checked for prevalence in the data, relevance of information content, and redundancy before being finalised.

1. **Prevalence in the data**: The fraction of the data covered by the motifs is compared against a threshold to ensure some minimum presence of the motif in the data:

   $$\frac{M \cdot k}{N \cdot L_A} \geq \phi^{-1}$$

   where
   M = Number of binding sites associated with the motif,
   k = Length of the motif in bases,
   N = Total number of sequences in the input data,
   L_A = Average length of input sequences in bases and,
   ϕ is a threshold parameter, which can be configured by setting the parameter *Minimum Frequency of Motif in the Data* under *Motif-finding* in the *ChIP-seq Analysis Algorithms* tab, *Tools* → *Options* menu. ϕ takes values between 5000 (default value) and 10000.

2. **Motif trimming:** A few locations within a motif may be insignificant, being fairly uniform (or close to background probability) in distribution. Motif trimming removes such locations.

   In the fitness evaluation step, motifs are evaluated for low information content through the LLR. If a large number of locations in a PWM have low information content (e.g., a column in the matrix with uniform probability for each base $\{A, C, G, T\}$), it will have a low LLR and consequently an undesirably high fitness score. However, there could be PWMs with a small number of redundant locations. The motif trimming step attempts to trim away these locations.

The information content at a position l is quantified by the following formula:

$$\mathcal{I}(l) = 2 + \sum_{b=\{A,C,G,T\}} f_{l,b} \cdot \log_2 (f_{l,b})$$

where, $f(l, b)$ is the frequency of occurrence of a base b at location l.

To examine whether a motif can be extended, all binding sites corresponding to a derived motif are extended by upto a maximum of 10 bases on either side. The extended strings are then trimmed one base at a time from both sides until any one of the following conditions is met:

(a) $\mathcal{I}(l) \geq 0.5$ at any single position.

(b) $\mathcal{I}(l) \geq 0.3$ at any two consecutive positions.

(c) $\mathcal{I}(l) \geq 0.2$ at any three consecutive positions.

3. **Motif filtering:** Motifs are compared against each other for the presence of repeats, subsets, reverse complements, etc. Details are given below.

The EM-optimized motifs at the end of all genetically altered generations may be similar to one another. For instance, PWMs resulting from a 10-base motif with consensus ATTTGCCATT and an 8-base motif ATGGCAAA may both be listed; the latter is simply a substring in the reverse complement of the former. Motif filtering sieves out such overlapping motifs by comparing PWMs using a q-sized sliding window for both the plus and reverse complementary orientations, with value $q = 6$. All PWMs that are closer than a critical value are represented only once, by the one with the highest fitness scores. Details of the algorithm are given below.

The PWMs are first rank-ordered according the fitness scores of the parent spaced dyads. Two motif lists are created: The first lists **unique motifs**, the second, **candidate motifs**. The motif with the highest fitness score is added to the unique motifs list; all the remaining form the candidate motifs list. For each candidate motif, the q-Distance is computed against each entity in the unique motifs list, and then compared against a column distance threshold, $CD\_Threshold$. Only if the condition q-Distance $\geq q \times CD\_Threshold$, is satisfied against each unique motif, the candidate gets added to the unique motif list.

Computing q-Distance

(a) **Column Distance** between two columns of a matrix, c_i, c_j, is given by

$$CD(c_i, c_j) \equiv \sum_{b:\{A,C,G,T\}} |c_i[b] - c_j[b]|$$

(b) **Window Distance** between two PWMs, PWM_1 and PWM_2 of the same (window) size q, is the sum of the pair-wise column distances

$$WD(PWM_1, PWM_2) = \sum_{i=1}^{q} CD\,(PWM_1[i], PWM_2[i])$$

(c) **q-Oriented Distance** between two PWMs, PWM_1 of size n, and PWM_2 of size m, is given by

$$OD_q(PWM_1, PWM_2) = \min_{i\in[1,\ldots,n-q],j\in[1,\ldots,m-q]} WD(PWM_1[i,\ldots,i{+}q{-}1], PWM_2[j,\ldots,j{+}q{-}1])$$

(d) **q-Distance** between two PWMs, PWM_1 of size n, and PWM_2 of size m, is given by

$$q\text{-Distance} = \min\left(OD_q(PWM_1, PWM2), OD_q(PWM_1, (PWM_2)^c)\right)$$

where $(PWM)^c$ is the complementary matrix to PWM.

Computing $CD\_Threshold$:

(a) Randomly generate 100,000 pairs of 4-element PWM-like vectors.

(b) Compute the distance between them and keep the distances in a sorted array.

(c) Set the critical value $CD\_Threshold$ at the distance representing 30% percentile of differences i.e. by chance 30% of random columns will be at distance less than this value.

2.3.2.9 Multiple Runs of GADEM

At the end of one run, binding sites of all output motifs are masked in the input data. More motifs are produced by running several GADEM cycles. The algorithm stops when one of the following conditions are satisfied:

1. No motifs can be found that satisfy the fitness criterion.
2. All motifs satisfying the fitness criterion get trimmed to size 0.
3. Binding sites corresponding to the motifs occur in less than 10% of the sequences.
4. Maximum number of runs have been reached. This can be set by modifying the *Number of runs* parameter under the *ChIP-seq Analysis Algorithms* tab, *Motif-finding* selection from the *Tools* $\rightarrow$ *Options* menu. The value is constrained to a maximum of 5 runs; the default value is 3.

2.3.2.10 Position Weight Matrix (PWM) Computation

Consider a collection of n sequences, $T = \{T_1, T_2, \ldots, T_n\}$, typically of varying length $\ell(T_i), i = 1, 2, \ldots, n$. Let $T_i^k(j)$ denote a k-length substring starting from the j^{th} position to the $(j+k-1)^{\text{th}}$ position of the i^{th} sequence. Let $t_i^k[j][m], m = 1, 2, \ldots, k$ denote the m^{th} nucleotide in the subsequence $T_i^k(j)$.

1. **Assigning weights to substrings:** For each substring $T_i^k(j)$,
   (a) compute $P\left[T_i^k(j)\right]$, the probability that this substring is generated from the motif. This involves computing probabilities of
      i. $T_i^k(j)$ under the distribution defined by the PWM; and
      ii. the remaining part of the sequence T_i, as per background probability assumptions.
   (b) compute the associated weight, $W\left(T_i^k(j)\right)$, given by

$$W\left(T_i^k(j)\right) = \frac{P\left[T_i^k(j)\right]}{\sum_{j=1}^{\ell(T_i)-k+1} P\left[T_i^k(j)\right]}.$$

2. **Updating the Profile Matrix:** Consider all k-mers for all $T_i, i = 1, 2, \ldots, n$. For $u \in \{1, 2, \ldots, k\}$ and $b \in \{A, C, G, T\}$, compute

$$PWM[u][b] = \sum_{i=1}^{n} \sum_{j=1}^{\ell(T_i)-k+1} W\left(T_i^k(j)\right) \underbrace{\mathbb{1}\left(t_i^k[j][u] = b\right)}_{\substack{\text{idenfities } u^{th} \text{ column} \\ \text{of the PWM,}}}$$

where $\mathbb{1}$ is the indicator function, taking value 1 when the argument is true and 0 otherwise. Normalize the matrix, so that $\sum_{\text{over all b}} PWM[i][b] = 1$.

2.3.2.11 Details of the EM algorithm in GADEM

GADEM an acronym for ''Genetic Algorithm guided Expectation Maximization'' algorithm for motif disovery uses an EM algoirthm (see Appendix B). This application of the EM algorithm has been developed by Li [22].

The EM algorithm used here uses inputs as the sequence data and the initial Position Weight Matrix (PWM). The GADEM algorithm is described in Section 2.3.2.12 and here we only describe the EM algorithm part of GADEM.

The assumptions made are that

1. each element of a sequence follows a multinomial (4-nomial) probability distribution;
2. the bases in various positions are independent;
3. the sequences have the same length L.

Let w be the length of the PWM. Each subsequence of length w is scored as the sum of matching cells of standardized log-transformed PWM. With both the positive and reverse complimentary starnds there are $2 \times (L - w + 1)$ subsequences for each sequence.

Notations:

$S_i, i = 1, 2, \ldots, N$ are the sequences in the data set;
s_{ij}: nucleotide in psotion j of sequence i;

$$Y_{i.j} = \begin{cases} 1 & \text{if protein binding site starts at position } \ j \ \text{ in sequence } \ i \\ 0 & \text{otherwise.} \end{cases}$$

$P_{ij} = P(Y_{i.j} = 1)$.
$\boldsymbol{\theta}$: $w \times 4$ matrix of letter probabilities, with elements $\theta_{\ell,b}$, $\ell = 1, 2, \ldots, w$.
$b = \{A, C, G, T\}$.
These are the parameters of interest to be estimated.
The data that we have are only the sequences $S_i, i = 1, 2, \ldots, N$. The probability of a sequence $S_i = (s_{i.1}, s_{i.2}, \ldots, s_{i.L})$ depends on the parameters through the site (motif) start position $i.j, j = 1, 2, \ldots, 2(L - w + 1)$ in the sequence S_i. If this position is known, say j (say, $Y_{i.j} = 1$) then

$$P(S_i | Y_{i.j} = 1) = \prod_{\ell=1}^{w} \theta_{\ell, s_{i.j}+\ell-1}.$$

It is easy to see that if we know $Y_{i.j}$ then the MLE of $\theta_{\ell,b}$ is

$$\hat{\theta}_{\ell,b} = \frac{n_{\ell,b}}{N},$$

where $n_{\ell,b}$ is the number of nucleotides b counted at position ℓ of the motif.

Thus the incompete-data problem is the problem of estimating $\theta_{\ell,b}$ from only the observations S_i, the distribution model for S_i being the mixture

$$P(S_i) = \sum_{j=1}^{2(L-w+1)} P(S_i | Y_{i.j} = 1) P(Y_{i,j} = 1).$$

A complete-data problem suitable for this situation is the problem with data $(S_i, Y_{i.j})$. The complete-data model is described by introducing the joint distribution of $Y_{i.j}, j = 1, 2, \ldots, 2(L - w = 1)$ as an equal probability multinomial

$$\prod_{j=1}^{L-w+1} (\frac{1}{L-w+1})^{Y_{i.j}}.$$

With this the complete-data likelihood based on $(S_i, Y_{i.j})$ is

$$\prod_{\ell=1}^{w} \theta_{\ell,s_{i.j}+\ell-1} \prod_{j=1}^{L-w+1} (\frac{1}{L-w+1})^{Y_{i.j}}$$

E-Step:

With this the E-Step is the calculation of

$$P(Y_{i.j} = 1) = N \frac{P(S_i|Y_{i.j} = 1, \boldsymbol{\theta})}{\sum_{i=1}^{N} \sum_{k=1}^{2(L-w+1)} P(S_i|Y_{i.j} = 1, \boldsymbol{\theta})}.$$

Then by defining $I\{s_{i,j+\ell-1} = b\} = 1$ if the $(j+\ell-1)^{\text{th}}$ nucleotide in sequence i matches nucleotide b (b = A, C, T, G) and 0 otherwise, the expected count of base b at position ℓ is

$$c_{\ell,b} = \sum_{i=1}^{N} \sum_{j=1}^{2(L-w+1)} P(Y_{i.j} = 1|S_i, \boldsymbol{\theta}) I\{s_{i,j+\ell-1} = b\}, \ell = 1, 2, \ldots, w; b = A, C, G, T.$$

M-Step:

The M-Step consists in computing updates

$$\hat{\theta}_{\ell,b} = \frac{c_{\ell,b}}{N}.$$

To avoid zeros sometimes a pseudo-count δ is added to make

$$\hat{\theta}_{\ell,b} = \frac{c_{\ell,b} + \delta}{N + 4\delta}.$$

A value like $\delta = 0.01$ is often used.

2.3.2.12 p-values for Motif Scores

p-values are computed by first deriving the associated probability mass function, i.e., PMF. This is the probability that a random string of a given length k has score *equal to* z.

Let $PMF^v[z]$ be the probability that a random string of length $v(\leq k)$ has score z. We require that a string of length 0, have score 0. In probabilistic terms,

$$PMF^0[z] = \begin{cases} 1 & z = 0 \\ 0 & z > 0. \end{cases}$$

The PMF can be recursively computed using the relation

$$PMF^i[z] = \sum_{b \in \{A,C,G,T\}} BP(b) \cdot PMF^{i-1}[z - SM[i][b]]$$

For details refer to [28].

1. Each column of the PFM matrix is considered a realization of a multinomial (4-nomial) frequency vector.
2. The p-value of column $i, i = 1, 2 \ldots, w$ denoted by p_i is computed for the hypothesis that the column is a realization of a random multinomial vector with given probabilities for $\{A, C, G, T\}$.
3. let $p = \prod_{i=1}^{w}$.
4. The p-value of the PFM matrix is calculates as ([4], Theorem 1) as

$$p \sum_{i=0}^{w-1} \frac{(-lnp)^i}{i!}.$$

2.3.2.13 E-values for Motif Scores

At the end of the EM algorithm, a motif is represented by a PWM matrix, which can be summarized by a quantity called the information content. A p-value is associated with this information content, which indicates the probability that a PWM matrix randomly drawn has information equal to or more than what is observed.

Then using the number of all possible alignments and this p-value a fitness score called E-value is computed as follows:

1. Let N: total number of sequences.
2. Let Q_i: length of the i^{th} sequence.
3. Let n: number of words in the alignments.
4. Compute $N_T = \sum_{i=1}^{n} (Q_i - w + 1)$, the total number of starting positions.
5. Compute
$$A(n) = \frac{N_T!}{n!\,(N_T - n)!},$$
the total number of possible alignments.
6. Calculate E-value as ln(p-value $\times$ $A(n)$). ([15]).

Chapter 3

RNA-Seq Analysis

3.1 Introduction

3.1.1 Gene Expression

Gene expression is the process that facilitates translation of a blueprint DNA into functional RNA and protein molecules in a cell. The first step in this process is **transcription** and involves gene activation by a set of proteins, mainly *transcription factors*. The resultant RNA transcript is created from selected exonic regions of DNA within the activated gene. As shown in Fig. 3.1, an activated gene is capable of producing more than one type of RNA transcript or isoform, based on the combination of exons spliced together. These alternatively spliced transcripts could be expressed in different tissues, or in the same tissue under different conditions.

Of the different species of RNA created, messenger RNA (mRNA) are particularly significant, as they directly translate into proteins which then participate in various biological processes further downstream. Different genes are activated in various tissue types at different times leading to different amounts of mRNA created. As malfunctions in gene expression, or aberrant splice variants, could lead to disruptions in the downstream biological pathways, gene and isoform expression profiles are of tremendous interest.

3.1.2 RNA-Seq: Challenges

RNA-Seq applies next-generation sequencing technology to sequence mRNA in the sample of interest. The mRNA sample is first converted into cDNA which then undergoes the next-generation sequencing process.

If an mRNA transcript is abundantly present, it reflects in the corresponding quantity of cDNA created and consequently in the number of reads that align to the parent gene. Gene expression is inferred by counting reads at gene and exon levels using genome annotations. Differential expression allows the comparison of expression levels across samples, under different conditions. Isoform or transcript quantification via RNA-Seq involves estimating expression of individual transcripts of a gene. Novel exon and gene discovery is also possible via RNA-Seq, when substantial read representation occurs from inter-genic or exonic regions of a chromosome.

Algorithms that process RNA-Seq data must handle multiple issues; some, such as duplicate or multiply mapping reads are inherent to next-generation sequencing, and have been discussed in Chapter 1, while others are particular to the RNA-Seq technology.

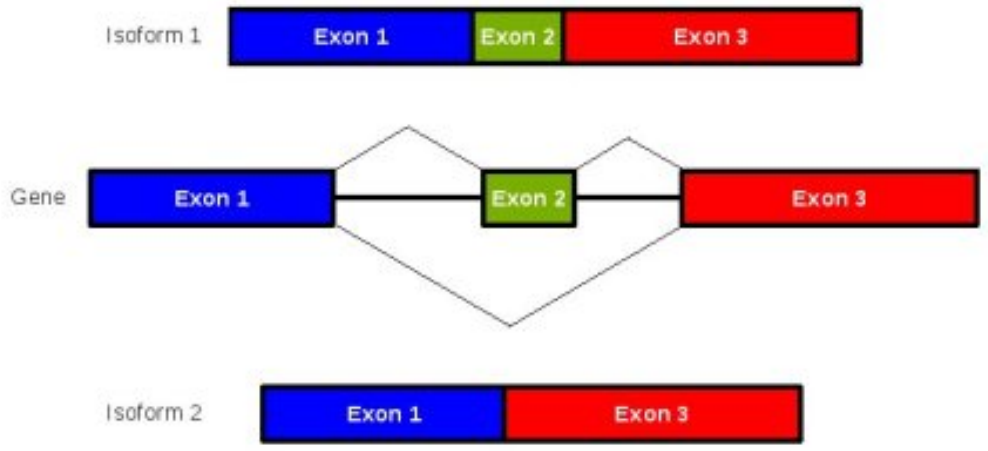

Figure 3.1: Different combinations of exons lead to alternatively spliced isoforms

- **Normalization of read counts based on exon length**

  Chapter 1 discussed why comparison across samples requires normalization by scaling of associated read counts. A larger number of reads map to longer exons or transcripts than to their shorter counterparts. Thus comparison across exons within a sample also requires normalization with respect to exon (or transcript) lengths.

  A commonly used measure to represent read counts over a region of interest is **R**eads **P**er **K**ilo base per **M**illion reads (**RPKM**) [24], which is computed as follows:

  $$RPKM \equiv \frac{R}{L \cdot N} \cdot 10^9 \tag{3.1}$$

  where
  $R \equiv$ number of reads mapping to the region of interest,
  $N \equiv$ total number of reads from the sample, and
  $L \equiv$ length of the region of interest.

  RPKM provides a normalized measure to allow comparison across samples for a particular gene, exon or transcript, as well as across genes, exons and transcripts within a sample.

- **Spliced reads**

  Spliced reads are those that straddle exon boundaries, and thus map partially to two known exons on the reference genome. Read R_1 shown in Fig. 3.2 directly maps to the reference as it is entirely contained in an exon. The spliced read R_2 on the other hand aligns partially with two disjoint portions of the reference. These reads are particular to RNA-sequencing, as intronic regions are entirely skipped and only exons spliced to form RNA transcripts. As read sizes increase with improvements in technology, we expect to see more spliced reads, especially for shorter exons.

- **Fractional reads**

  During the read counting process, we often come across reads that do not lie entirely within an exon. Not all of these can be explained away as known exon splices; some map partially onto known exons, and partly onto intronic regions. These reads are useful in novel isoform and exon discovery. Algorithms differ in their approach, as to whether such reads should contribute wholly or fractionally to the exon read count.

- **Overlapping exons**

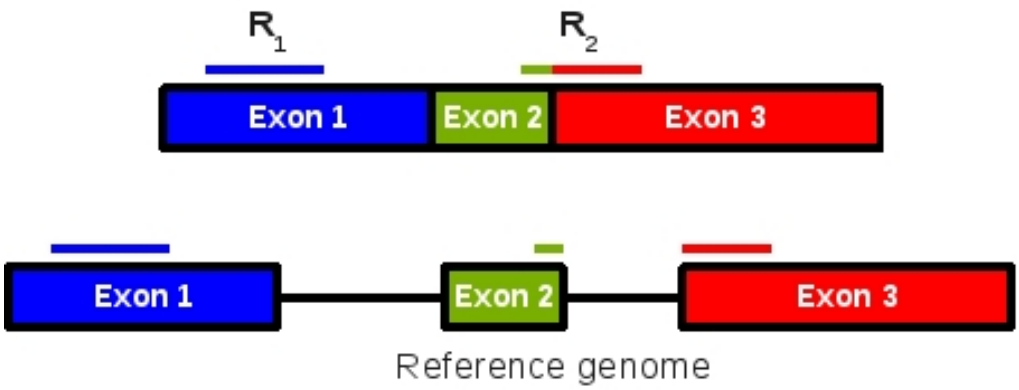

Figure 3.2: Spliced reads

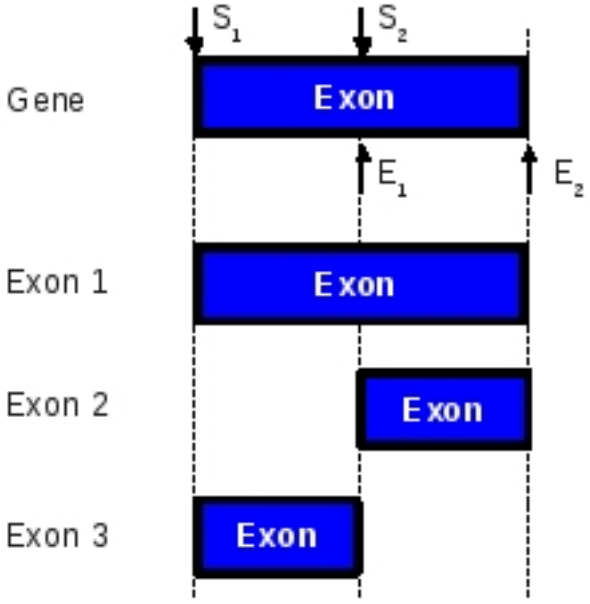

Figure 3.3: Overlapping exons on a gene

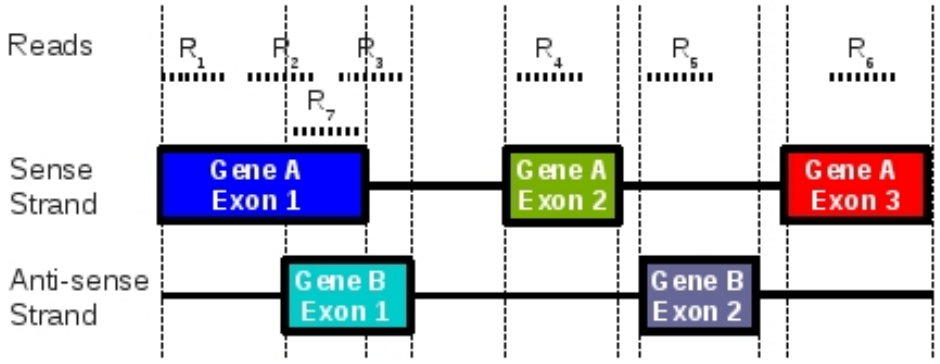

Figure 3.4: Overlapping exons on positive and negative strands

- **On the same gene, same strand**

  Several genes contain exons that have different start and end points but display a clear overlap. Assigning reads that fall in the overlap to any one of the exons poses a problem. In Fig. 3.3, exon 1 has two start points S_1 and S_2 and two stop points E_1 and E_2; thus three exonic representations are possible.

- **On a different gene on the opposite strand**

  The problem is a little more complex in case the overlapping genes and exons lie on opposite strands. Currently not all next-generation sequencing technologies discriminate between the sense and antisense strands. Therefore, reads from overlapping exons on opposite strands, cannot be easily assigned. Note that there is no confusion in aligning the reads to the reference genome, the difficulty is only in assigning the read to the appropriate strand. For example, all reads shown in Fig. 3.4 can be easily assigned to the parent gene and exon, with the exception of R_7. This read may be assigned to Gene A, Exon 1 or Gene B, Exon 1.

  Of course, in case a technology can distinguish between reads emerging from positive and negative strands, this issue poses no problem.

- **On a different gene on the same strand**

  In a few cases, there is an overlap between genes on the same strand, which may also result in overlapping exons. Resolution in such cases is via a probabilistic approach, depending on the expressions of the two genes.

- **3′ bias**

  During the library preparation process, cDNA is created from mRNA beginning at the 3′ end and growing towards the 5′ end. mRNAs longer than 3 kb do not always extend completely to the 5′ end, which results in an over-representation of the 3′ end in comparison to the 5′ end of the transcript.

- **Lack of poly-A tails in some mRNA**

  Most mRNA are characterized by polyadenylated (poly-A) tails at the 3′ end which protect them from degradation and also play a role in transcription termination. The poly-A tails also make it convenient for selection of mRNA during RNA sequencing. However, there are mRNAs without poly-A tails, such as histone mRNAs which consequently remain undetected by the RNA-seq process.

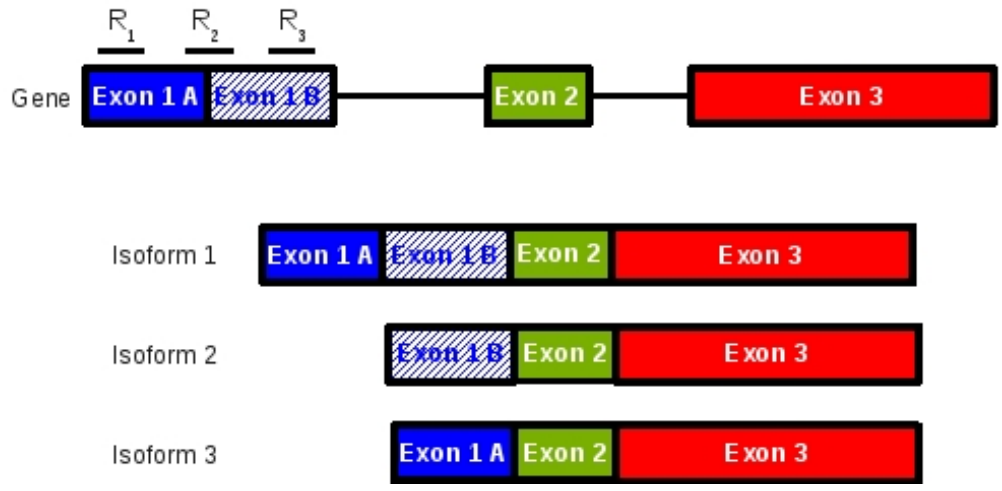

Figure 3.5: Exon partitions

3.2 Overview of Algorithms

3.2.1 Partitioning Exons

As mentioned earlier in Section 3.1.2, exon overlaps are often observed within the same gene. Fig. 3.5 elaborates on the three exonic representations introduced earlier in Fig. 3.3. Read R_2 is easily assigned to one isoform, but R_1 and R_3 each map to two. This can be handled by partitioning the exon into non-overlapping sub-units and performing read counting for these partitioned units. Thus, the reads R_1 and R_3 now map uniquely to the partitions Exon 1A and 1B, while the read R_2 now becomes a spliced read.

While doing analysis at the gene level, we consider all the transcript structures and derive the partition set of a gene. The partition set for a gene is defined as the smallest set of disjoint regions that cover all exons of the gene. This computation depends only on the annotation used and is performed at the outset upon creation of the experiment.

3.2.2 Quantification

3.2.2.1 Quantification of Known Exons and Genes

Quantification is done by counting the number of reads that map to the gene and exon of interest. This computation is done on a per sample basis. For greater speeds, the algorithm is parallelized on different processors; each available processor runs the algorithm for one chromosome. The user is allowed the option of discarding reads that map partially to exon partitions. The algorithm described in this section assumes that both partial and multiple matches are present in the read list.

In any case, the algorithm first handles reads that map uniquely to the reference genome. If there is no ambiguity introduced by overlapping exons from different genes, it directly assigns them to the appropriate exonic partition and updates read counts as follows:

- **Read count of exon partition:** The count of the partition is increased by one for each **entire read** appropriated to it. Reads that map partially to an exonic partition contribute fractionally to its read count.

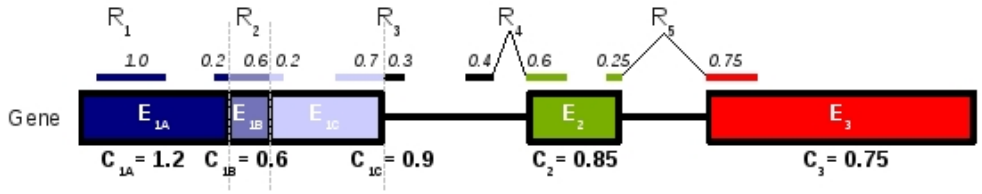

Figure 3.6: Counting unique reads unambiguously assigned to exon partitions

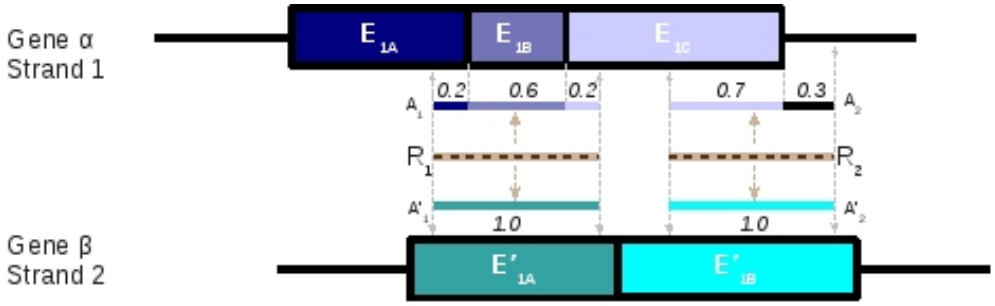

Figure 3.7: Counting unique reads in case of ambiguous assignment to exon partitions

- **Read count for a gene:** A read contributes one unit to the total read count for a gene if it lies completely within the union of the exon partitions of the gene. This includes both reads lying entirely within an exon partition as well as spliced reads. Reads that partially map over intronic or intergenic regions contribute less than one unit. This approach is consistent for all technologies using fixed-length reads. Thus, currently **Avadis NGS** correctly accounts for Illumina and ABI reads of fixed length. Variable length reads, such as those created by the 454 technology are at present treated as having a fixed length and contribute uniformly irrespective of length.

Fig. 3.6 depicts an example: All reads in the figure are assigned by the alignment algorithms to unique positions on the reference. Read R_1 lies entirely within exon partition E_{1A} as shown, and so increases the count of E_{1A} by one. Read R_2 is multiply spliced; 20% of it lies within E_{1A}, 60% in E_{1B} and the rest in E_{1C}. Read R_3 lies partly in E_{1C}, the rest lies in an intronic region. Read R_4 is a spliced read connecting an intronic region with E_2. Read R_5 is a spliced read lying across two exons E_2 and E_3. The read counts C for each partition are as shown in the figure. Even though there are 5 reads that align to the gene the total read count for the gene is computed as the sum of the individual partition counts and is $C_g = 4.3$. This is easily accounted for by the fact that 30% of read R_3 and 40% of read R_4 lie in intronic regions and are therefore not included.

Based on the gene read counts computed from uniquely assigned reads, RPKMs are computed for each gene, as defined in equation 3.1 (specifically the read count over all exonic partitions of the gene normalized by the sum of the lengths of the partitions and total number of reads in the sample).

The algorithm is modified for a read that lies in a region corresponding to overlapping exons on different genes. In such a situation, even though the position of the read with respect to the reference is certain and unique, it can be assigned to exons on either strand. Such reads are fractionally assigned after the appropriation of all other unique reads; the assigned fraction is proportional to the RPKM of the gene.

In Fig. 3.7 reads R_1 and R_2 are aligned to the reference in a way that suggests either gene α or β

as their origin. Assume the RPKMs of genes α and β to be 4.0 and 1.0 respectively. In this case, read counts for each of the exons partitions shown in the figure will be

$$C^{\alpha}_{1A} = 0.2 \times 0.8 = 0.16$$
$$C^{\alpha}_{1B} = 0.6 \times 0.8 = 0.48$$
$$C^{\alpha}_{1C} = (0.2 + 0.7) \times 0.8 = 0.72$$
$$C^{\beta}_{1A} = 1.0 \times 0.2 = 0.2$$
$$C^{\beta}_{1B} = 1.0 \times 0.2 = 0.2$$

and adding counts over each gene we have $C_{\alpha} = 1.36$ and $C_{\beta} = 0.4$. As in the previous example (Fig. 3.6), the total contribution to the read count is 1.76, which falls short of the 2 reads due to the portion of read R_2 that falls in the intronic region.

After **all** the unique reads have been assigned, both ambiguous and unambiguous, RPKMs are recomputed for all genes. If detection of novel genes (described in next section) is chosen to be performed, it is done at this stage and the RPKMs are recomputed for all genes. Next the algorithm assigns multiple matches using an approach similar to that for ambiguous, unique reads, i.e., based on their new, relative RPKM ratios. Final RPKM ratios are calculated considering all desired reads; these are output along with the raw read counts for each gene.

Note that the algorithm does not take into account read alignment scores or individual base qualities on the reads in the quantification. Also, mate-pairs are considered equivalent to two individual reads, taking no notice of their paired property.

3.2.2.2 Novel Genes/Exons

The novel transcribed region detection algorithm in **Avadis NGS** discovers regions that are similar in structure and in expression to the known transcribed regions present in the Gene and Transcript Annotation files, G_A, T_A, that were used to create the experiment. The algorithm is configurable by a number of parameters that can change the ratio of false positives to false negatives. The default parameters are used in the following description of the algorithm. These default values can be changed via **Tools** → **Options** → **RNA-Seq** → **Novel genes and exons**. The algorithm proceeds in four phases:

Learning parameters :

- The exon, intron and gene length distributions are computed from T_A. The minimum novel exon length, l_e, minimum novel intron length, l_i, and the minimum novel gene length, l_g are set as the 10th percentile of the exon, intron, and gene length distributions respectively. The maximum intron length, l_I is set as the 90th percentile of the intron length distribution.
- The distribution of RPKMs of all known partitions with a non-zero count is computed and the 50th percentile of this distribution is set as the minimum RPKM threshold, m_r, for new exons. The distribution of RPKMs of all known genes with a non-zero count is computed and the 50th percentile of this distribution is set as the minimum RPKM threshold, m_g, for new genes.

Novel partition creation : The transcript annotation, T_A, is used to identify maximal regions, R, that are do not overlap any known exon (these roughly correspond to inter-genic and intronic regions). For each region of R, the following steps are carried out:

- the region is tiled with consecutive windows of length l_e,
- windows with RPKM greater than m_r are merged into a novel exon if they are less than l_i apart,

- the novel exon is retained if it's RPKM is greater than m_r and it has at least 10 reads

At the end of this step, the novel exons all satisfy the minimum length, RPKM criteria and the distance between novel exons is greater than l_i.

Assignment to known genes : The rules for assigning a novel partition to a gene are as follows:

- Novel partitions are allocated to known genes if they share spliced reads with those genes.
- Novel partitions which are not allocated to any of the known genes in the above step are allocated to known genes if they share paired reads with those genes. Paired reads spanning across genes are ignored while assigning a partition to a known gene.
- Any novel partition remaining unallocated in the above two steps, that is within a distance of l_I from known genes is assigned to the closest such gene.
- A novel partition that gets allocated to multiple genes in the above steps is assigned the status ''Ambiguous".
- A novel partition that gets allocated to a single known gene is assigned to that gene (but not to any specific transcript of the gene), if the RPKM of the novel partition is greater than a specified fraction (can be set from *Tools* $\longrightarrow$*Options* $\longrightarrow$*RNA-Seq* $\longrightarrow$*Novel Genes and Exons*, default value is 75%) of the gene's RPKM in some sample. Novel partitions that fail this criterion because of insufficient RPKM are marked as such.

Creation of new genes : Remaining novel partitions that share spliced reads / paired reads, or the ones that are within a distance of l_I from each other are clustered into novel genes and retained if their length, RPKM exceed m_g, m_r respectively. But in the Novel Detection report we show the discarded novel genes also with their status showing the reason for discarding (Small Length Gene, or Low RPKM Gene)

If the sample is directional (ABI RNA-Seq data for instance), then two passes are made - one for each strand. In this case, the novel genes are assigned a strand. Novel genes that are created from non-ABI data cannot be assigned a strand. Novel partitions are assigned a ''Location" -- one of Internal, Upstream, Downstream -- based on their position with respect to the gene. Novel partitions are also assigned a ''Nature" -- one of Standalone, Extension -- based on their proximity to known exons.

3.2.2.3 Quantification of Isoforms

The object of this analysis is to use reads data to estimate the relative expression levels of various known isoforms, using the annotations available (assumed complete and accurate). The relative expression of an isoform is defined as the fraction of reads that are derived from it among all the reads. Under certain reasonable assumptions, this is equivalent to the proportion of nucleotides in the isoform.

As pointed out earlier, RNA-Seq reads do not span entire transcripts; transcripts from which reads are derived are not uniquely determined either. Various factors contribute to read-mapping uncertainty, such as gene and isoform multi-reads (alignments to multiple positions in reference genome or transcriptome).

We implement the method of Li et al. [22] which is based on a probabilistic model for read distributions with relative isoform expression levels as parameters. The maximum likelihood method is used for estimation of these parameters. In the model formulated by Li et al., in principle, any known distribution can be incorporated for sampling of reads (in respect of their start positions). Moreover, in the absence of a more realistic model for sampling of reads we have assumed uniform distribution of start positions of reads over its length.

The model also incorporates a *noise* isoform, which generates reads not mapping to any known isoform. It takes into account a user-specified model for sequencing errors. The model has two latent variables---isoform expression and start position of the read.

The parameters in the model are $\theta_i, i = 0, 1, 2, \ldots, M$ the relative proportions of the M isoforms including the noise isoform indexed by subscript 0, so that $\sum_{i=0}^{M} \theta_i = 1$. The θ_i's represent fractions of nucleotides in the transcriptome made up by isoform i. It is assumed that start positions of reads are uniformly distributed over the length ℓ_i of the i^{th} isoform to which it is assigned; start positions could be even ℓ_i because of the presence of poly(A) tails. Under this assumption

- as the number of reads $N \to \infty$ the fraction of reads assigned to transcripts tends to the fraction of nucleotides making up the isoform.
- $10^9 \frac{\theta_i}{\ell_i}$is an estimate of RPKM.

3.2.2.4 Isoform Quantification Algorithm

The N reads (sequences) are the observations (data) $R_n, n = 1, 2, \ldots, N$, each of length L. Associated with each read R_n is a random variable S_n which represents its start position along the length ℓ_i of the transcript i. S_n is regarded as a latent variable which, conditional on the isoform being i, is assumed to be uniform over $\{1, 2, \ldots, \ell_i\}$ as mentioned above. In the Li et al. model a position-specific substitution matrix for nucleotide bases (of base a in read for base b in reference) can be used to handle base-call errors, but in our implementation we do not use it; that is, our substitution matrix is the 4×4 identity matrix at every position. For the noise isoform an equal-probability distribution over $\{A, T, G, C\}$ is used.

We treat paired-end reads as separate reads if they do not align perfectly and entirely within one or more isoforms. Thus we consider only the case of perfect alignment of paired-end reads.

For each read, the Expectation-Maximization (EM) algorithm [10] is used to estimate the parameters and the relative expressions of the $M + 1$ isoforms and $\sum_{i=0}^{M} \theta_i = 1$.

The inputs to the algorithm are:

- reads;
- isoform annotations;
- iteration control parameters:
  - maximum number of iterations (default: 1000) ; or
  - convergence criterion to be satisfied by each isoform (absolute value of difference between consecutive iteration values) (default: 0.001);

The output consists of estimates of relative isoform expression levels (including the noise isoform) for each sample. If isoform expression levels are required for pooled samples, data are to be presented in terms of pooled samples.

A discussion of the EM algorithm is given in Appendix B and a description of the specific algorithm used here is given in Section 3.3.1.

3.2.2.5 Differential Isoform Expression

If isoform quantification is carried out (for the same gene/exon) as above for several conditions and for several samples within conditions, then differential isoform expression between conditions

is quantiified by providing an index d_m for each isoform including the noise isoform. The index is between 0 and 1. The larger the d_m are, the more is the differential expression, and the isoforms m with large d_m are responsible for the differential expression. The largest of the d_m's is output as the splicing index for the gene/exon. Details of the algorithm used and the definition of d_m are given in Section 3.3.2.

3.2.3 Normalization of Read Counts

The expression values obtained by quantifying the association of reads to genes and exons are called 'raw counts', and can be normalized using one of multiple normalization options available: DESeq, TMM, Quantile, or Normalize Total Sample Read Count. The raw counts after applying the normalization on them are called the 'normalized counts'. The normalized counts are log-transformed and baselined (if the baselining option is chosen) to result in what are called 'normalized signal values'.

The details of the normalization and baselining options are given in Appendix C.

3.2.4 Gene Fusion

Transcripts with regions from multiple genes can be created at various stages from transcription to creation of proteins. A classic example is the BCR-ABL inter-chromosomal gene fusion in Leukemia. Such transcripts stemming from the fusion of 2 genes can be identified via spliced reads or paired reads spanning genes. The algorithm for detection of gene fusions is briefly described below:

- **Check for paired reads aligning to different genes:** Mate pair reads occuring in different genes are indicative of transcripts formed by fused genes.
- **Check for spliced reads partially mapping to different genes:** Spliced reads occurring in different genes arise from reads that occur at the splice sites of the two genes.
- **Discard read-pairs with incorrect configuration or orientation:** Read-pairs showing anomalous behavior are attributed to errors and discarded. Illumina paired end reads with forward-forward or reverse-reverse orientations when both genes lie on the same strand are an example of erroneous configuration.

  Valid configurations for Illumina paired-end reads are listed in Table 3.1 and shown in Fig. 3.8. The transcript used for illustrative purposes has representations from two genes, Gene 1, present at the $5^{'}$ end of the transcript, and Gene 2 at its $3^{'}$ end. Reads 1 and 2 map to Gene 1 and 2 respectively. A_0 to A_3, and B_0 to B_3, represent all possible valid alignments to the reference. Cases A_0 and B_0 depict the simplest configuration, when both Gene 1 and 2 are on the positive or negative strand, and the reads map to the reference in the expected order. A_1 illustrates alignment when A_0 undergoes an intra-chromosomal translocation. A_2 and A_3 depict configurations when one of the genes in A_0 experiences an inversion event. The same arguments hold for B_1 to B_3.

  For spliced reads in the forward orientation, the first gene is the $5^{'}$ end and the second one is the $3^{'}$ end of the transcript. For spliced reads in the reverse orientation, the gene that encountered first lies at the $3^{'}$ end.
- **Deviant threshold:** If the number of remaining read-pairs mapping to two specific genes exceeds the minimum threshold defined by the user, declare a fused gene.
- **Detect read through transcripts:** Transcripts created by genes adjacent to each other on the chromosome are called read through transcripts. These could arise from the same or different strands.

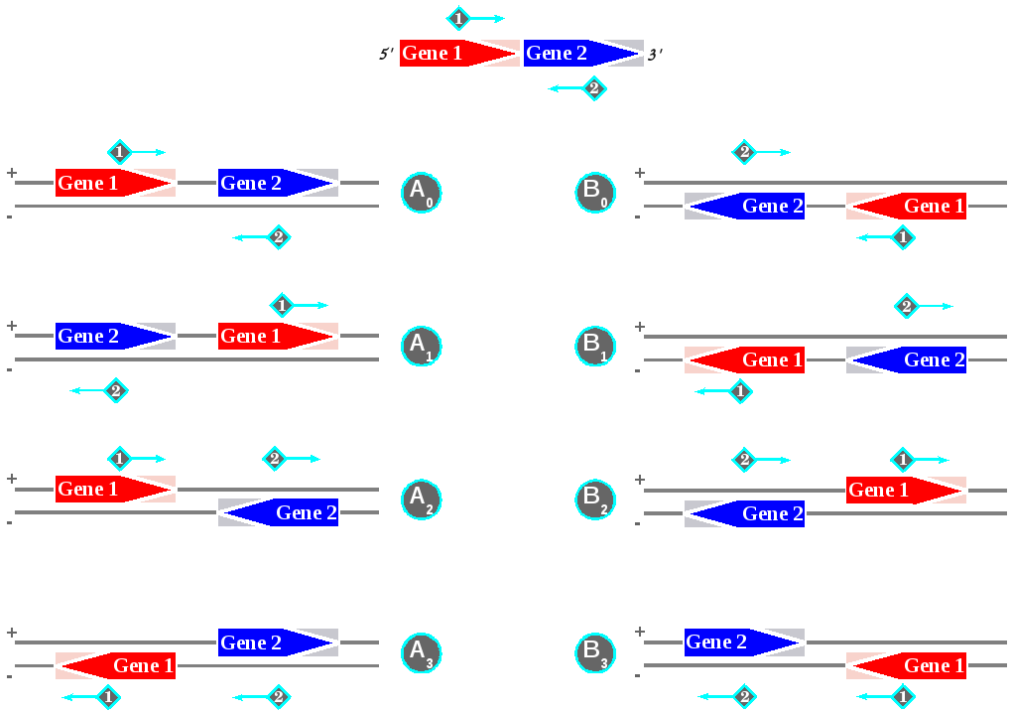

Figure 3.8: Gene fusion events: Possible alignments to a reference

Case in Fig. 3.8	Gene 1 Orientation	Gene 2 Orientation	Possible Orientation	Gene 1 ($5'$) Mapping
A_0	+	+	F-R	F read
A_1	+	+	R-F	F read
A_2 and B_2	+	-	F-F	read aligning to + strand
A_3 and B_3	-	+	R-R	read aligning to - strand
B_0	-	-	F-R	R read
B_1	-	-	R-F	R read

Table 3.1: Gene Fusion Orientations

Inter-Chromosomal Fusions: Note that **Avadis NGS** is capable of identifying inter-chromosomal fusions on account for paired reads but cannot handle splicing across chromosomes. This will be handled in a later release.

3.2.5 Differential Gene/Exon Expression

Two types of tests are provided for differential expression.

1. **Replicate Analysis:** These tests are to be used when replicates are availabe for each condtion. Both parametric and non-parametric tests are avaialble. Parametric tests assume that the normalized signal values are normally distributed. For more details on the statistical tests, see Section 3.3.3.

2. **Pooled Analysis:** We provide two tests under this category: the AC Test and the Z-test. The AC-test pools together raw counts across replicates within each condition and tests the differences between conditions based on the Poisson assumption on distribution of counts. The Z-test uses the square-root transformation but uses a 'fixed' variance as dictated by theory rather than estimating variance from replicates. Both these tests are applicable only to compare two conditions. Details of these tests are given in Section 3.3.4.

These tests when done for multiple entities are followed by multiple testing correction. See Appendix D for details of the various options available in **Avadis NGS** for multiple testing correction.

In addition to these tests, **Avadis NGS** also provides DESeq, a statistical test for differential expression of sequence count data from the literature, as an R script. This script calls the R package DESeq, of Simon Anders, et. al. to do differential expression analysis. The statistical method behind DESeq is explained in [12].

3.3 Mathematical Details of the Algorithms

3.3.1 Isoform Quantification

3.3.1.1 EM algorithm

This is an implementation of the algorithm in [22] and [11] with some changes; the changes have already been mentioned in 3.2.2.3 and 3.2.2.4. We continue to use the same descriptions and notations given in these sections. This method is a more general version of the method used in ERANGE. Also, a well-known model due to Jiang and Wong ([20]) is similar to a special case of this Li et al. model. This model is similar to one used by Beissbarth et al. ([6]) for sequencing errors in SAGE libraries.

The algorithm is an EM algorithm (see Appendix B) for maximum likelihood estimation (MLE) of the isoform expression proportions $\theta_m, m = 0, 1, 2, \ldots, M$, the subscript 0 standing for the "noise" isoform. We use the notation $\boldsymbol{\theta}$ for $(\theta_0, \theta_1, \theta_2, \ldots, \theta_M)$. These proportions are estimated for each sample in a condition within each gene, depending upon how the data are presented.

The latent variables (in the sense described in Appendix B) for each read R_n (with observed value r_n) are G_n the actual isoform from which the read originated and S_n the start position of the read on the isoform. As mentioned in 3.2.2.3, another latent variable O_n that is used in [22] is not invoked here; this is the strand information. We assume that every read has the same orientation as its parent isoform.

3.3.1.2 Incomplete-data Problem

The incomplete data that we have are N reads $R_n, n = 1, 2, \ldots, N$ independently distributed with a probability distribution depending on the parameters $\theta_m, m = 0, 1, 2, \ldots, M$ and the values of the latent variables s_n, g_n. The distribution of R_n then is of the form

$$P(R_n = r_n) = \sum_{s_n, g_n} P(R_n = r_n | g_n, s_n) P(s_n, g_n | \boldsymbol{\theta}),$$

and we justifiably write $P(s_n, g_n|\boldsymbol{\theta}) = P(s_n|g_n)P(g_n|\boldsymbol{\theta})$.

Let us denote by Z_{nij} the binary $\{0, 1\}$ variable with $Prob(Z_{nij} = 1) = P(S_n = j|G_n = i)P(G_n = i) = P(S_n = j|G_n = i)\theta_i$, for $i \neq 0$. For $i = 0$, we use an equal probability distribution over $\{A, T, G, C\}$ independent of S_n.

The distribution of S_n depends on the isoform and so we need to model $P(S_n|G_n)$. The random variable $(S_n|G_n = i)$ takes values in $\{1, 2, \ldots, \ell_i\}$, where ℓ_i is the length of the isoform i. We assume this distribution to be uniform over this set. Then $P(Z_{nij} = 1) = P(S_n = j|G_n = i)\theta_i = \frac{\theta_i}{\ell_i}$.

Let $\rho = (\rho_1, \rho_2, \ldots, \rho_L)$ be a L-length sequence of bases. Then

$$P(R_n = \rho | Z_{nij} = 1) = 1 \quad \text{and} \quad P(R_n = \rho | Z_{nij} = 0) = 0 \quad \text{for} \quad i \neq 0.$$

And $P(R_n = \rho|Z_{n0j} = 1) = (\frac{1}{4})^L$ and $P(R_n = \rho|Z_{n0j} = 0) = 0$.

Thus the model for the distribution of a read R_n is

$$\sum_{i=0}^{M} \sum_{j=1}^{\ell_i} \frac{\theta_i}{\ell_i} P(R_n = \rho | Z_{nij} = 1).$$

This is a mixture model of the kind described in Appendix B. The data we have is **incomplete-data** in this sense. The incomplete-data likelihood function is

$$\prod_{n=1}^{N} \sum_{i=0}^{M} \sum_{j=1}^{\ell_i} \frac{\theta_i}{\ell_i} P(R_n = \rho | Z_{nij} = 1).$$

3.3.1.3 Complete-data Problem

The complete-data problem is one where Z_{nij} is known whereupon the MLEs of the parameters θ_i are simply the sample proportions where $Z_{nij} =$ for a given i, that is

$$\hat{\theta}_i = \frac{\sum_{n=1}^{N} \sum_{j=1}^{\ell_i} Z_{nij}}{N}.$$

Then we apply the standard EM algorithm for the resolution of a mixture model.

The E-step is

$$E_{Z|r_n, \boldsymbol{\theta}^{(t)}}(Z_{nij}^{(t)}) = \frac{\frac{\theta_i^{(t)}}{\ell_i} P(r_n | Z_{nij} = 1)}{\sum_{i'=0}^{M} \sum_{j'=1}^{\ell_{i'}} \frac{\theta_{i'}^{(t)}}{\ell_{i'}} P(r_n | Z_{ni'j'} = 1)}.$$

The M-Step then is

$$\theta_i^{(t+1)} = \frac{E_{Z|r_n, \boldsymbol{\theta}^{(t)}}(Z_{nij}^{(t)})}{N}.$$

The EM algorithm is run gene by gene with the collection of isoforms for the gene taken from the annotations.

3.3.1.4 Modification for Partially Aligned Reads

Let f_n be the fraction of the n^{th} read that we wish to allocate among the isoforms. Of course, $f_N = 1$ if it is a standard read and < 1 if it is a partial read. Then the EM algorithm is suitably modified as follows: The E-step is

$$E_{Z|r_n,\boldsymbol{\theta}^{(t)}}(Z_{nij}^{(t)}) = f_n \frac{\frac{\theta_i^{(t)}}{\ell_i} P(r_n|Z_{nij}=1)}{\sum_{i'=0}^{M}\sum_{j'=1}^{\ell_{i'}} \frac{\theta_{i'}^{(t)}}{\ell_{i'}} P(r_n|Z_{ni'j'}=1)}.$$

The M-Step then is

$$\theta_i^{(t+1)} = \frac{E_{Z|r_n,\boldsymbol{\theta}^{(t)}}(Z_{nij}^{(t)})}{\sum_{n=1}^{N} f_n}.$$

3.3.1.5 Some Implementation Details

Reads which are not well aligned are already filtered out before reads are analyzed by this algorithm.

Start values for the parameters are computed as follows:
An arbitrary value in $(0, 1)$ is assigned to θ_0. Then values for $\theta_i, i = 1, 2, \ldots, M$ are assigned in proportion to the length ℓ_i of isoform i so that $\sum_{i=0}^{M} \theta_i = 1$. Thus

$$\frac{\frac{\theta_i}{(1-\theta_0)\ell_i}}{\sum_{j=1}^{M} \frac{\theta_j}{(1-\theta_0)\ell_j}} = \frac{1}{M},$$

making

$$\theta_i = \frac{\ell_i}{\sum_{j=1}^{M} \ell_j}(1-\theta_0).$$

3.3.2 Differential Isoform Expression

If isoform quantification is carried out (for the same gene/exon) as above for several conditions and for several samples within conditions, then differential isoform expression between conditions is quantiified as explained below. Note that for each sample (within each condition) there are $M + 1$ proportions adding up to 1, representing the relative expression levels of the isoforms including the noise isoform. Let these be $\theta_{ijm}, i = 1, 2, \ldots, k, j = 1, 2, \ldots, n_i, m = 0, 1, 2, \ldots, M$, i denoting the condition, j (within i) denoting the sample, and m denoting the isoform (0 denoting the noise isoform). The steps in the computation are:

1. Summarize these proportions for each condition by the average of the proportions over the samples; that is, compute

$$\theta_{im} = \frac{1}{n_i}\sum_{j=1}^{n_i} \theta_{ijm}, \quad i = 1, 2, \ldots, k; m = 0, 1, 2, \ldots, M.$$

2. Compute
$$\tilde{\theta}_m = \min_{i=1 \text{ to } k} \theta_{im}; \quad \breve{\theta}_m = \max_{i=1 \text{ to } k} \theta_{im}.$$

3. Compute $d_m = \breve{\theta}_m - \tilde{\theta}_m$.

4. Output (m, d_m) by sorting d_m in decreasing order.

The values d_m quantify differential expression between conditions. d_m take values between 0 and 1. Large values of d_m evidently are indicators of differential expression and small values indicate no differential expression. When there is differential expression, the isoform(s) responsible for it appear at the top of the table of sorted d_m. This may well include the noise isoform. These quantities are displayed in the gene view.

3.3.3 Statistical Tests for Replicate Analysis

3.3.3.1 Unpaired t-Test for Two Groups

The standard test that is performed in such situations is the so called t-test, which measures the following t-statistic for each gene g (see, e.g., [19]):

$t_g = \frac{m_1 - m_2}{s_{m_1 - m_2}}$

where $s_{m_1-m_2} = \sqrt{\frac{(n_1-1)s_1^2+(n_2-1)s_2^2}{n_1+n_2-2}(\frac{1}{n_1} + \frac{1}{n_2})}$ is the unbiased pooled variance estimate.

Here, m_1, m_2 are the mean expression values for gene g within groups 1 and 2, respectively, s_1, s_2 are the corresponding standard deviations, and n_1, n_2 are the number of experiments in the two groups. Qualitatively, this t-statistic has a high absolute value for a gene if the means within the two sets of replicates are very different and if each set of replicates has small standard deviation. Thus, the higher the t-statistic is in absolute value, the greater the confidence with which this gene can be declared as being differentially expressed. Note that this is a more sophisticated measure than the commonly used fold-change measure (which would just be $m_1 - m_2$ on the log-scale) in that it looks for a large fold-change in conjunction with small variances in each group, The power of this statistic in differentiating between true differential expression and differential expression due to random effects increases as the numbers n_1 and n_2 increase.

3.3.3.2 t-Test against 0 for a Single Group

This is performed on one group using the formula

$$t_g = \frac{m_1}{\sqrt{s_1^2/n_1}}$$

3.3.3.3 Paired t-Test for Two Groups

The paired t-test is done in two steps. Let $a_1 \ldots a_n$ be the values for gene g in the first group and $b_1 \ldots b_n$ be the values for gene g in the second group.

- First, the paired items in the two groups are subtracted, i.e., $a_i - b_i$ is computed for all i.
- A t-test against 0 is performed on this single group of $a_i - b_i$ values.

3.3.3.4 Unpaired Unequal Variance t-Test (Welch t-test) for Two Groups

The standard t-test assumes that the variance of the two groups under comparison. Welch t-test is applicable when the variance are significantly different. Welch's t-test defines the statistic t by the following formula:

$$t_g = \frac{m_1 - m_2}{\sqrt{s_1^2/n_1 + s_2^2/n_2}}$$

Here, m_1, m_2 are the mean expression values for gene g within groups 1 and 2, respectively, s_1, s_2 are the corresponding standard deviations, and n_1, n_2 are the number of experiments in the two groups. The degrees of freedom associated with this variance estimate is approximated using the Welch-Satterthwaite equation:

$$df = \frac{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}{\frac{s_1^4}{n_1^2 - df_1} + \frac{s_2^4}{n_2^2 - df_2}}$$

3.3.3.5 Unpaired Mann-Whitney Test

The t-Test assumes that the gene expression values within groups 1 and 2 are independently and randomly drawn from the source population **and** obey a normal distribution. If the latter assumption may not be reasonably supposed, the preferred test is the non-parametric Mann-Whitney test , sometimes referred to as the Wilcoxon Rank-Sum test. It only assumes that the data within a sample are obtained from the same distribution but requires no knowledge of that distribution. The test combines the raw data from the two samples of size n_1 and n_2 respectively into a single sample of size $n = n_1 + n_2$. It then sorts the data and provides ranks based on the sorted values. Ties are resolved by giving averaged values for ranks. The data thus ranked is returned to the original sample group 1 or 2. All further manipulations of data are now performed on the rank values rather than the raw data values. The probability of erroneously concluding differential expression is dictated by the distribution of T_i, the sum of ranks for group i, $i = 1, 2$. This distribution can be shown to be normal mean $m_i = n_i(\frac{n+1}{2})$ and standard deviation $\sigma_1 = \sigma_2 = \sigma$, where σ is the standard deviation of the combined sample set.

3.3.3.6 Paired Mann-Whitney Test

The samples being paired, the test requires that the sample size of groups 1 and 2 be equal, i.e., $n_1 = n_2$. The absolute value of the difference between the paired samples is computed and then ranked in increasing order, apportioning tied ranks when necessary. The statistic T, representing the sum of the ranks of the absolute differences taking non-zero values obeys a normal distribution with mean $m = \frac{1}{2}(n_1\frac{(n_1+1)}{2}) - S_0)$, where S_0 is the sum of the ranks of the differences taking value 0, and variance given by one-fourth the sum of the squares of the ranks.

The Mann-Whitney and t-test described previously address the analysis of two groups of data; in case of three or more groups, the following tests may be used.

3.3.3.7 Moderated t-Test

The moderated t-test is a modification of the unpaired t-test. It is tuned to handle the following corner cases in a better way:

- Two conditions having small differences between their means and also having very low variance within each condition which results in the unpaired t-test calling this differentially expressed which may not be correct.

- Two conditions having large difference between their means and also having very high variance within each condition which might result in them not passing the unpaired t-test.

The moderated t-test attempts at estimating the variance, v_{global} and a degree of freedom, d_{global} of the assumed normal distribution across genes. The details of the process can be obtained from [7].

Following this, the moderated variance for each gene is computed by combining the variance for that gene across the given condition and the 'global variance' calculated. The variance across the condition is calculated as

$s_{m_1-m_2} = \sqrt{\frac{(n_1-1)s_1^2+(n_2-1)s_2^2}{d_{f(m_1-m_2)}}(\frac{1}{n_1}+\frac{1}{n_2})}$

$d_{f(m_1-m_2)} = n_1 + n_2 - 2$

Here, m_1, m_2 are the mean expression values for gene g within groups 1 and 2, respectively, s_1, s_2 are the corresponding standard deviations, and n_1, n_2 are the number of experiments in the two groups.

The moderated variance is then calculated as follows:

$s_{mod} = \frac{s_{m_1-m_2} * d_{f(m_1-m_2)} + s_{global} d_{global}}{d_{f(m_1-m_2)} + d_{global}}$

Once the moderated variance has been calculated, it is used for calculating the corresponding T-Stat values as

$t_{gmod} = \frac{m_1-m_2}{s_{mod}}$

It is to be noted that this test works with the premise of unpaired samples and equal variances for both the groups involved. The test has not been made available against zero because the zero condition test essentially is like a paired t-test and the literature doesn't say much about using moderated t-test for the paired case.

3.3.3.8 One-Way ANOVA

When comparing data across three or more groups, the obvious option of considering data one pair at a time presents itself. The problem with this approach is that it does not allow one to draw any conclusions about the dataset as a whole. While the probability that each individual pair yields significant results by mere chance is small, the probability that any one pair of the entire dataset does so is substantially larger. The One-Way ANOVA takes a comprehensive approach in analyzing data and attempts to extend the logic of t-tests to handle three or more groups concurrently. It uses the mean of the sum of squared deviates (SSD) as an aggregate measure of variability between and within groups. NOTE: For a sample of n observations $X_1, X_2, ...X_n$, the sum of squared deviates is given by

$$SSD = \sum_{i=1}^{n} X_i^2 - \frac{(\sum_{i=1}^{n} X_i)^2}{n}$$

The numerator in the t-statistic is representative of the difference in the mean **between** the two groups under scrutiny, while the denominator is a measure of the random variance **within** each group. For a dataset with k groups of size $n_1, n_2, ...n_k$, and mean values $M_1, M_2, ..., M_k$ respectively, One-Way ANOVA employs the SSD between groups, SSD_{bg}, as a measure of variability in group mean values, and the SSD within groups, SSD_{wg} as representative of the randomness of values within groups. Here,

$$SSD_{bg} \equiv \sum_{i=1}^{k} n_i (M_i - M)^2$$

and

$$SSD_{wg} \equiv \sum_{i=1}^{k} SSD_i$$

with M being the average value over the entire dataset and SSD_i the SSD within group i. (Of course it follows that sum $SSD_{bg} + SSD_{wg}$ is exactly the total variability of the entire data).

Again drawing a parallel to the t-test, computation of the variance is associated with the number of degrees of freedom (df) within the sample, which as seen earlier is $n-1$ in the case of an n-sized sample. One might then reasonably suppose that SSD_{bg} has $df_{bg} = k - 1$ degrees of freedom and SSD_{wg}, $df_{wg} = \sum_{i=1}^{k} n_i - 1$. The mean of the squared deviates (MSD) in each case provides a measure of the variance between and within groups respectively and is given by $MSD_{bg} = \frac{SSD_{bg}}{df_{bg}}$ and $MSD_{wg} = \frac{SSD_{wg}}{df_{wg}}$.

If the null hypothesis is false, then one would expect the variability between groups to be substantial in comparison to that within groups. Thus MSD_{bg} may be thought of in some sense as $MSD_{hypothesis}$ and MSD_{wg} as MSD_{random}. This evaluation is formalized through computation of the

$$F - ratio = \frac{MSD_{bg}/df_{bg}}{MSD_{wg}/df_{wg}}$$

It can be shown that the F-ratio obeys the F-distribution with degrees of freedom df_{bg}, df_{wg}; thus p-values may be easily assigned.

The One-Way ANOVA assumes independent and random samples drawn from a normally distributed source. Additionally, it also assumes that the groups have approximately equal variances, which can be practically enforced by requiring the ratio of the largest to the smallest group variance to fall below a factor of 1.5. These assumptions are especially important in case of unequal group-sizes. When group-sizes **are** equal, the test is amazingly robust, and holds well even when the underlying source distribution is not normal, as long as the samples are independent and random. In the unfortunate circumstance that the assumptions stated above do not hold and the group sizes are perversely unequal, we turn to the Welch ANOVA for unequal variance case or Kruskal-Wallis test when the normality assumption breaks down.

3.3.3.9 Post hoc Testing of ANOVA results

The significant ANOVA result suggests rejecting the null hypothesis H0 = ''means are the same''. It does not tell which means are significantly different. For a given gene, if any of the group pair is significantly different, then in ANOVA test the null hypothesis will be rejected. Post hoc tests are multiple comparison procedures commonly used on only those genes that are significant in ANOVA F-test. If the F-value for a factor turns out non significant, one cannot go further with the analysis. This 'protects' the post hoc test from being (ab)used too liberally. They are designed to keep the experiment wise error rate to acceptable levels.

The most common post hoc test is **Tukey's** Honestly Significant Difference or HSD test . Tukey's test calculates a new critical value that can be used to evaluate whether differences between any

two pairs of means are significant. One simply calculates one critical value and then the difference between all possible pairs of means. Each difference is then compared to the Tukey critical value. If the difference is larger than the Tukey value, the comparison is significant. The formula for the critical value is:

$HSD = q\sqrt{\frac{MS_{error}}{n}}$, where q is the studentized range statistic (similar to the t-critical values, but different). MS_{error} is the mean square error from the overall F-test, and n is the sample size for each group. Error df is the df used in the ANOVA test.

SNK test is a less stringent test compared to Tukey HSD. $SNK = q_r\sqrt{\frac{MS_{error}}{n}}$ Different cells have different critical values. The r value is obtained by taking the difference in the number of steps between cells and q_r is obtained from standard table. In Tukey HSD the q value is identical to the lowest q from the Newman-Keuls.

3.3.3.10 Unequal Variance (Welch) ANOVA

ANOVA assumes that the populations from which the data came all have the same variance, regardless of whether or not their means are equal. Heterogeneity in variance among different groups can be tested using Levine's test (not available in **Avadis NGS**). If the user suspect that the variance may not be equal and the number of samples in each group is not same, then Welch ANOVA should be done.

In Welch ANOVA, each group is weighted by the ratio of the number of samples and the variance of that group. If the variance of a group equals zero, the weight of that group is replaced by a large number. When all groups have zero variance and equal mean, the null hypothesis is accepted, otherwise for unequal means the null hypothesis is rejected.

3.3.3.11 Kruskal-Wallis Test

The Kruskal-Wallis (KW) test is the non-parametric alternative to the One-Way independent samples ANOVA, and is in fact often considered to be performing ''ANOVA by rank''. The preliminaries for the KW test follow the Mann-Whitney procedure almost verbatim. Data from the k groups to be analyzed are combined into a single set, sorted, ranked and then returned to the original group. All further analysis is performed on the returned ranks rather than the raw data. Now, departing from the Mann-Whitney algorithm, the KW test computes the **mean** (instead of simply the sum) of the ranks for each group, as well as over the entire dataset. As in One-Way ANOVA, the sum of squared deviates between groups, SSD_{bg}, is used as a metric for the degree to which group means differ. As before, the understanding is that the groups means will not differ substantially in case of the null hypothesis. For a dataset with k groups of sizes $n_1, n_2, ..., n_k$ each, $n = \sum_{i=1}^{k} n_i$ ranks will be accorded. Generally speaking, apportioning these n ranks amongst the k groups is simply a problem in combinatorics. Of course SSD_{bg} will assume a different value for each permutation/assignment of ranks. It can be shown that the mean value for SSD_{bg} over all permutations is $(k-1)\frac{n(n-1)}{12}$. Normalizing the observed SSD_{bg} with this mean value gives us the H-ratio, and a rigorous method for assessment of associated p-values: The distribution of the

$$H - ratio = \frac{SSD_{bg}}{\frac{n(n+1)}{12}}$$

may be neatly approximated by the chi-squared distribution with $k-1$ degrees of freedom.

3.3.3.12 Repeated Measures ANOVA

Two groups of data with inherent correlations may be analyzed via the paired t-Test and Mann-Whitney. For three or more groups, the Repeated Measures ANOVA (RMA) test is used. The RMA test is a close cousin of the basic, simple One-Way independent samples ANOVA, in that it treads the same path, using the sum of squared deviates as a measure of variability between and within groups. However, it also takes additional steps to effectively remove extraneous sources of variability, that originate in pre-existing individual differences. This manifests in a third sum of squared deviates that is computed for each individual set or row of observations. In a dataset with k groups, each of size n,

$$SSD_{ind} = \sum_{i=1}^{n} k(A_i - M)^2$$

where M is the sample mean, averaged over the entire dataset and A_i is the mean of the $k$values taken by individual/row i. The computation of SSD_{ind} is similar to that of SSD_{bg}, except that values are averaged over individuals or rows rather than groups. The SSD_{ind} thus reflects the difference in mean per individual from the collective mean, and has $df_{ind} = n - 1$ degrees of freedom. This component is removed from the variability seen within groups, leaving behind fluctuations due to "true" random variance. The F-ratio, is still defined as $\frac{MSD_{hypothesis}}{MSD_{random}}$, but while $MSD_{hypothesis} = MSD_{bg} = \frac{SSD_{bg}}{df_{bg}}$ as in the garden-variety ANOVA.

$$MSD_{random} = \frac{SSD_{wg} - SSD_{ind}}{df_{wg} - df_{ind}}$$

Computation of p-values follows as before, from the F-distribution, with degrees of freedom $df_{bg}, df_{wg} - df_{ind}$.

3.3.3.13 Repeated Measures Friedman Test

As has been mentioned before, ANOVA is a robust technique and may be used under fairly general conditions, provided that the groups being assessed are of the same size. The non-parametric Kruskal Wallis test is used to analyst independent data when group-sizes are unequal. In case of correlated data however, group-sizes are necessarily equal. What then is the relevance of the Friedman test and when is it applicable? The Friedman test may be employed when the data is collection of ranks or ratings, or alternately, when it is measured on a non-linear scale.

To begin with, data is sorted and ranked **for each individual or row** unlike in the Mann Whitney and Kruskal Wallis tests, where the entire dataset is bundled, sorted and then ranked. The remaining steps for the most part, mirror those in the Kruskal Wallis procedure. The sum of squared deviates between groups is calculated and converted into a measure quite like the H measure; the difference however, lies in the details of this operation. The numerator continues to be SSD_{bg}, but the denominator changes to $\frac{k(k+1)}{12}$, reflecting ranks accorded to each individual or row.

3.3.3.14 N-way ANOVA

The N-Way ANOVA is used to determine the effect due to N parameters concurrently. It assesses the individual influence of each parameter, as well as their net interactive effect.

Avadis NGS uses type-III sum of square (SS) in N-way ANOVA [16, 17]. This is equivalent to the method of weighted squares of means or complete least square method of Overall and Spiegel

[18]. The type-III ss is defined as follows :
Let A and B be the factors, each having several levels. The complete effects model for these two factors is
$y_{ijk} = \mu + a_i + b_j + t_{ij} + e_{ijk}$,
where y_{ijk} is the k-th observation in ij-th treatment group, μ is the grand mean, $a_i(b_j)$ is additive combination and t_{ij} is the interaction term and e_{ijk} is the error term, which takes into account of the variation in y that cannot be accounted for by the other four terms on the right hand side of the equation. The difference in residual sum of square (RSS) of the models
$y_{ijk} = \mu + a_i + t_{ij} + e_{ijk}$,
and
$y_{ijk} = \mu + a_i + b_j + t_{ij} + e_{ijk}$, is the SS corresponding to factor B. Similarly, for other factors we take the difference of RSS of the model excluding that factor and the full model.

Avadis NGS ANOVA can handle both balanced and unbalanced design, though only full factorial design is allowed. For more than three factors, terms only up to 3-way interaction is calculated, due to computational complexity. Moreover, **Avadis NGS** calculates maximum 1000 levels, i.e., if the total number of levels for 3-way interaction model is more than 1000 (main + doublet + triplet), then **Avadis NGS** calculates only up to 2-way interactions. Still if the number of levels is more than 1000 **Avadis NGS** calculates only the main effects.

Full factorial designs with no replicate excludes the highest level interaction (with previous constraints) to avoid over fitting.

Missing values are handled in **Avadis NGS** ANOVA. If for a condition, if more than one sample has values, then ANOVA handles them. But, if all the samples have missing values, then those values (entities) are excluded for p-value computation and a separate list titled 'Excluded Entities' is output at the end. See Fig. 3.9.

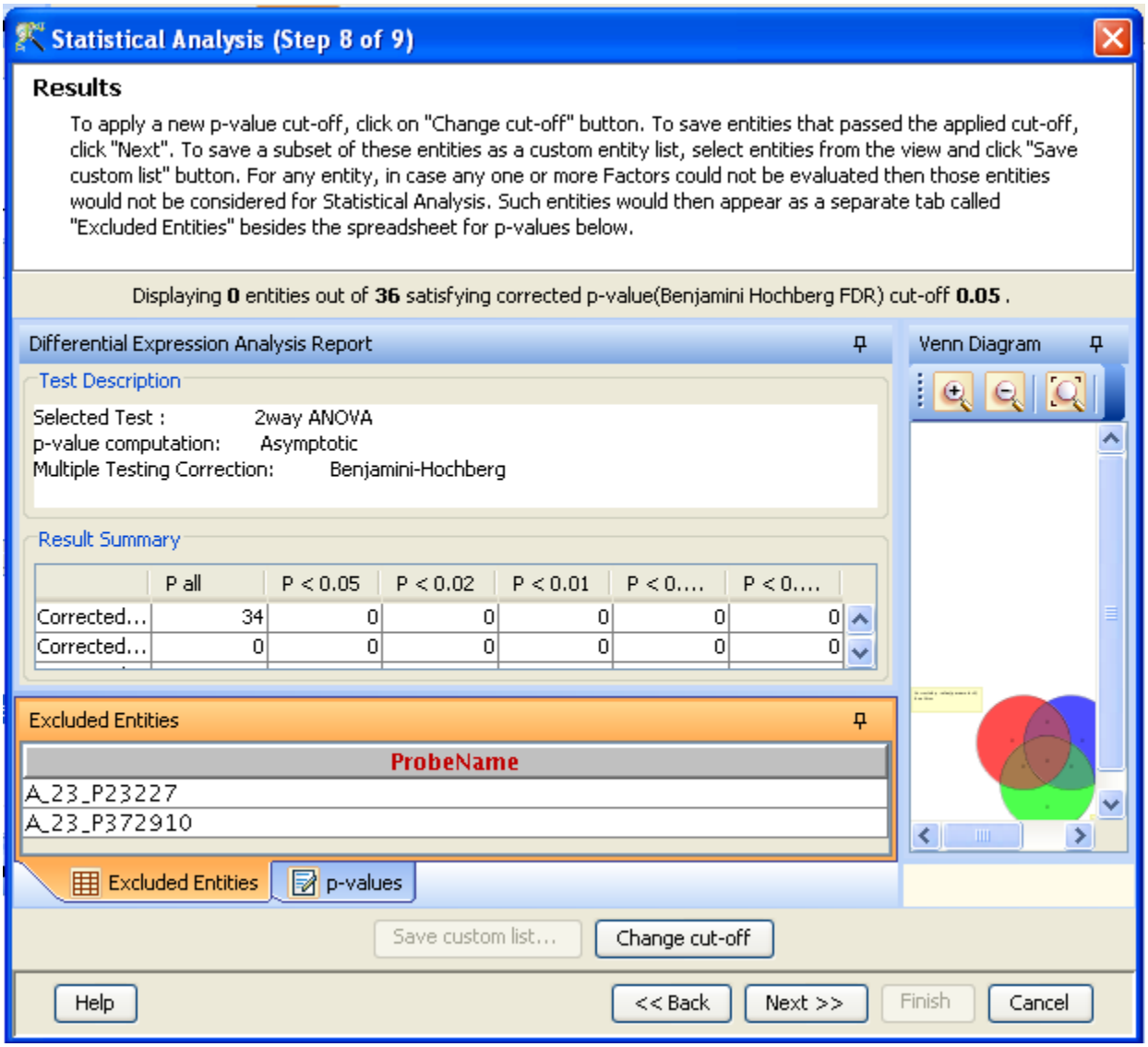

Figure 3.9: Anova result showing 'Excluded Entities' because of missing values

3.3.4 Statistical Tests for Pooled Samples

3.3.4.1 Z-Test

Reads are normalized by the total number of mapped reads before they are put through this test. If there are several samples under a condition, they are pooled. Assuming read counts to be distributed as Poisson, they are subjected to a square root transformation in an effort to make them nearly normally distributed with a constant ($\frac{1}{4}$) variance. Then a hypothesis test of differential expression is carried by a Z-test (normal distribution test) for two conditions with known variance.

This test is based on transformed Poisson variables to normality. It is well known that if $X \sim \text{Poisson}(\lambda)$ then $\sqrt{X}$ has an approximate normal distribution with mean $\sqrt{\lambda}$ and constant variance of $\frac{1}{4}$.

Algorithm:

1. Let G_1 and G_2 be the two experimental groups.
2. Let q_1 and q_2 be the number of samples from G_1 and G_2 respectively.
3. Let $\{x_{11}, x_{12}, \ldots, x_{1q_1}\}$ and $\{x_{21}, x_{22}, \ldots, x_{2q_2}\}$ be the read counts from groups G_1 and G_2 respectively. Each observation is associated with a total mapped read count, so we have read counts $\{N_{11}, N_{12}, \ldots, N_{1q_1}, N_{21}, N_{22}, \ldots, N_{2q_2}\}$.
4. Let $x_1 = \sum_{i=1}^{q_1} x_{1i}$, $x_2 = \sum_{i=1}^{q_2} x_{2i}$, $N_1 = \sum_{i=1}^{q_1} N_{1i}$ and $N_2 = \sum_{i=1}^{q_1} N_{2i}$.
5. The test statistic is given by, $Z = \frac{\sqrt{\frac{x_1}{N_1}} - \sqrt{\frac{x_2}{N_2}}}{\sqrt{\frac{1}{4N_1} + \frac{1}{4N_2}}}$, which has a standard normal distribution under the hypothesis of no differential expression.
6. p-value is calculated as follows.

$$\text{If } Z < 0,\ p = 2 * \Phi(Z)$$
$$\text{else } p = 2 * (1 - \Phi(Z)).$$

where $\Phi(Z)$ is the Standard Normal CDF and Z is the test statistic computed in the previous step.

3.3.4.2 AC-Test (Audic-Claverie Test)

Originally formulated as a method of finding differentially expressed genes in a SAGE framework, this technique examines if tag counts or read counts in two libraries or groups or conditions have the same mean under the assumption that the counts or reads have a Poisson distribution ([2]; [31]). We use this method for testing differential expression between two conditions by consolidating observations under each condition into a single number. The test is based on the conditional distribution of one of the counts given the other as a negative binomial distribution; this derivation is done in a Bayesian framework. Depending on how small or large the observations are, we compute exact negative binomial probabilities, or approximate them numerically using the incomplete beta integral, or approximate them statistically using an asymptotic normal distribution of with the same mean and variance as the negative binomial variable.

The original Audic-Claverie test is a one-sided test. We give both one-sided and two-sided p-values.

Algorithm:

1. Let A and B be the two groups with more than one observation. Group A has observations (read counts) $\{x_{A1}, x_{A2}, \ldots, x_{Ap}\}$ and B has observations (read counts) $\{x_{B1}, x_{B2}, \ldots, x_{Bq}\}$ and we assume that these are all identically distributed under each condition. Each observation is associated with a total mapped read count;

$$\{N_{A1}, N_{A2}, \ldots, N_{Ap}; N_{B1}, N_{B2}, \ldots, N_{Bq}\}$$

are the mapped read counts corresponding to the observations.

2. Let $x_A = \sum_{i=1}^{p} x_{Ai}$, $x_B = \sum_{j=1}^{q} x_{Bj}$.

3. Let $N_A = \sum_{i=1}^{p} N_{Ai}$ and $N_B = \sum_{j=1}^{q} N_{Bj}$. N_A and N_B may not be the same.

4. Experiment A could be run first, followed by B. Then the probability of observing a read count x_B in the region of interest in the second run, given that x_A was already observed in the first run, has been shown by Audic-Claverie under their model to be the negative binomial probability

$$p(x_B|x_A) = (\frac{N_B}{N_A})^{x_B} \frac{(x_A+x_B)!}{(x_A)!(x_B)!(1+\frac{N_B}{N_A})^{x_A+x_B+1}}$$

5. For large values of x_A, x_B, it is hard to compute this exactly since it involves factorials. So we use the incomplete beta integral approximation of the negative binomial for moderate values of x_A, x_B and transformed normal approximation for large values of x_A, x_B as follows:

6. Using *Incomplete Beta approximation* to negative binomial, we calculate probability as follows:

   (a) Compute $p_1 = \text{Ibeta}(x_A + 1, x_B + 1, p)$, where x_A, x_B are given read counts, p_1 is the probability, and Ibeta is the incomplete beta function, which can be computed using N_A and N_B.

   (b) Compute $p_2 = 1 - p_1$.

7. For large values, the negative binomial is approximated by a normal distribution with the same mean and variance as the negative binomial.

8. One-sided p-value is computed as,

$$p\text{-value} = \min\{p1, p2\}.$$

9. Two-sided p-value is twice the above p-value.

Chapter 4

Small Variant (SNP/MNP) Analysis

4.1 Introduction

Most organisms within a particular species differ very little in their genomic structure. A single nucleotide variation between two complementary DNA sequences corresponding to a specific locus is referred to as a SNP or Single Nucleotide Polymorphism and is the most common variation observed within a species. This variation could be due to insertion, deletion or substitution of a nucleotide. The variations are referred to as allele changes at the specific chromosomal co-ordinates. Typically, SNPs commonly observed in a population exhibit two alleles---a major allele, which is more prevalent, and a relatively rarely occurring minor allele. Such variations, if they occur in genic regions, can result in changed phenotypes and may even cause disease.

Sometimes, SNPs occur together. A set of adjacent SNPs is termed a Multiple Nucleotide Polymorphism or MNP. Thus MNP alleles have multiple nucleotides.

Microarrays for SNP detection use probes with specific targets while searching for SNPs. Next-generation sequencing (NGS) allows SNP identification without prior target information. The high coverage possible in NGS also facilitates discovery of rare alleles within population studies.

SNP detection algorithms compare the nucleotides present on aligned reads against the reference, at each position. Based on the distribution of As, Ts, Gs, and Cs at that position, and the likelihood of error, a judgement is made as to the existence of a SNP.

Some issues that must be handled by SNP detection algorithms are mentioned below:

- **Quality of base-calls**

  A sequencer outputs a sequence of nucleotides corresponding to each read. It also assigns a quality value based on the confidence with which a particular base is called. Clearly, SNP detection algorithms must put greater weight on bases called with higher confidence.

- **Mapping quality of reads**

  Most alignment algorithms assign quality scores to a read based on how well the read aligned with the reference. These scores are relevant in SNP detection because they measure the likelihood of a read originating from the suggested position on the reference. Even if the individual bases on a read are called with high quality values, the read may align imperfectly with the reference. The mapping quality score takes into account the inserts, deletes, and substitutions necessary for alignment at a particular position.

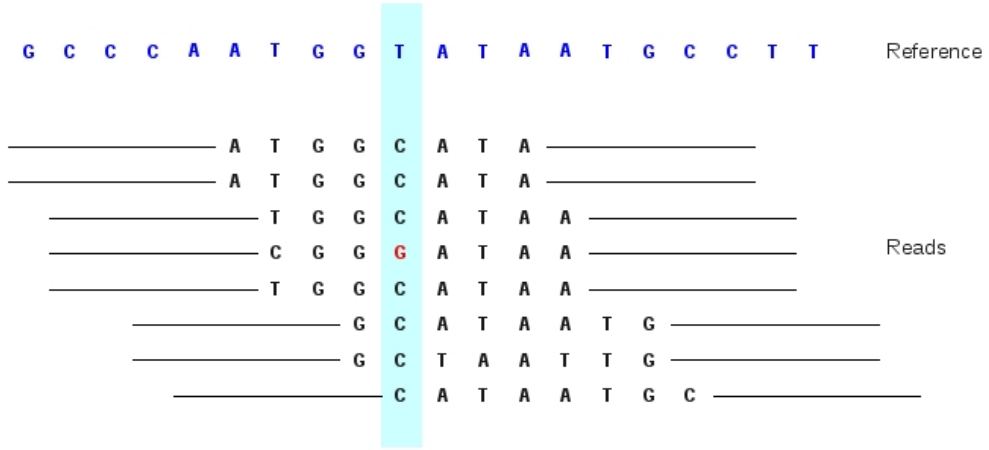

Figure 4.1: Homozygous SNP

- **Depth of coverage**

  The number of reads ‘‘covering’’ a position also determine how confidently a SNP can be called. Obviously greater sequencing depths lead to higher SNP calling accuracy.

- **Homopolymer**

  A *homopolymer* is a repetitive sequence of a single nucleotide, e.g., AAAAAA. Sequencers often exhibit inaccurate representations of homopolymers and their immediate neighbors due to limitations in the next-generation sequencing technology. Such regions need to be handled carefully by SNP detection algorithms.

- **Ploidy**

  Ploidy is the number of sets of chromosomes in a cell. Haploid organisms have one set of chromosomes per cell, while diploid organisms like humans have two. Polyploidy, the state where all cells have multiple (but more than two) sets of chromosomes is chiefly observed in plants. SNP detection must take into account the ploidy while calling SNPs. For example, in a sample of a haploid individual organism, each position can correspond to only one nucleotide. Reads must ideally agree at each position, and any deviation is easily detected as a sequencing error. Diploid organisms inherit one set of chromosomes from each parent and so reads can show two nucleotides at each position, one from each set.

  Currently, SNP analysis in **Avadis NGS** assumes that the ploidy is two. In this case a SNP can be called under two circumstances:

  1. **Homozygous SNP**
     One when the consensus reached from the sample reads shows a single nucleotide at the location and this consensus differs from the reference nucleotide. In Fig. 4.1, the highlighted position shows a **T** in the reference genome but a consensus nucleotide of **C** among the reads. The nucleotide **G** in one of the reads is attributed to sequencing error.
  2. **Heterozygous SNP**
     Alternatively, the sample reads may show considerable presence of two different nucleotides; typically one of the two agrees with the reference nucleotide. The two alleles should ideally show close to 50% presence each. In Fig. 4.2, the reference genome shows a **T** at the locus of interest, whereas the reads show a combination of **T** and **C** bases.

In the following we describe the two main steps in the SNP analysis: i) SNP Detection, and ii) SNP Effect Analysis. SNP Detection identifies the SNPs by comparing the given sample’s genome with

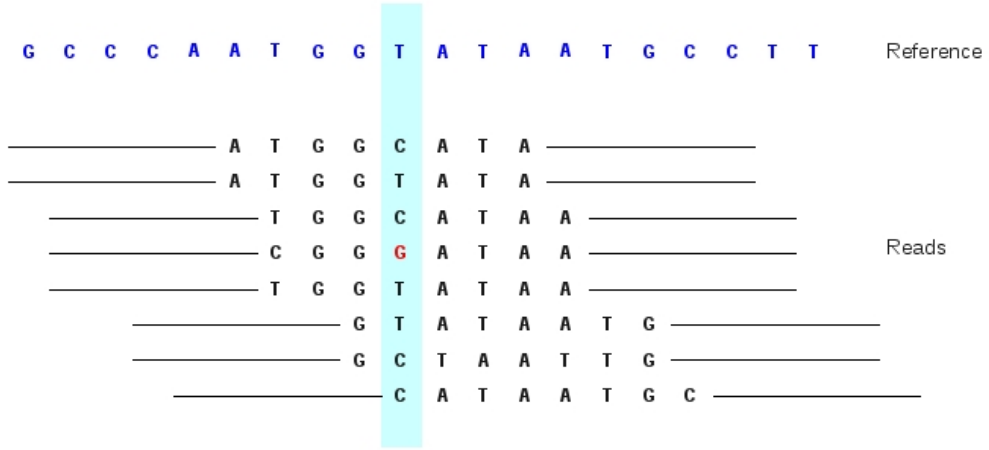

Figure 4.2: Heterozygous SNP

the reference genome. SNP Effect Analysis takes the list of SNPs detected in the first step and generates a report indicating the effect that these SNPs have on the genes in a given annotation.

4.2 Local Realignment

4.2.1 Background and Motivation

Since alignment algorithms map each read independently to the reference genome, it may result in some alignment artefacts. This means that SNPs or indels can be improperly placed with respect to their true location. This could be because of 1) sequencing errors, 2) several equally likely positions of the variant, and 3) adjacent variants or near by errors [37]. In particular, insertions and deletions especially towards the ends of the reads are difficult to anchor and resolve without the use of multiple reads.

These misalignments can lead to false positive SNP calls as well as skewed recalibrated base quality scores and there by have critical impact on the downstream results. Therefore the most important objective behind realigning reads around indels is to reduce the number of false positive variant calls.

Consider for example the case shown in Fig. 4.3, which demonstrates the issues of alignment artefacts. If we shift the 6 reads having C,T as mismatch bases to the left by having C as insertion, then the T bases cease to be mismatches and the inserted sequence matches other reads which already had an insertion at that locus.

As another example, consider the case shown in Fig. 4.4. In the reads that show mismatches, if we shift TG to the right to allow for 3 deletions, there will be no mismatches and the percentage of supporting reads for deletion would be increased.

4.2.2 Local Realignment Approach

Some of the popular approaches for local realignment include *GATK* [39], *Dindel* [35] and *SRMA* [37]. In fact both GATK and Dindel, barring few differences use same method conceptually and we have adopted ideas from both of these approaches in **Avadis NGS**. *SRMA* on the other hand, follows a different approach, which uses short read alignment coupled with assembly inspired approach and builds a variant graph for each region from the initial alignments. Once the variant

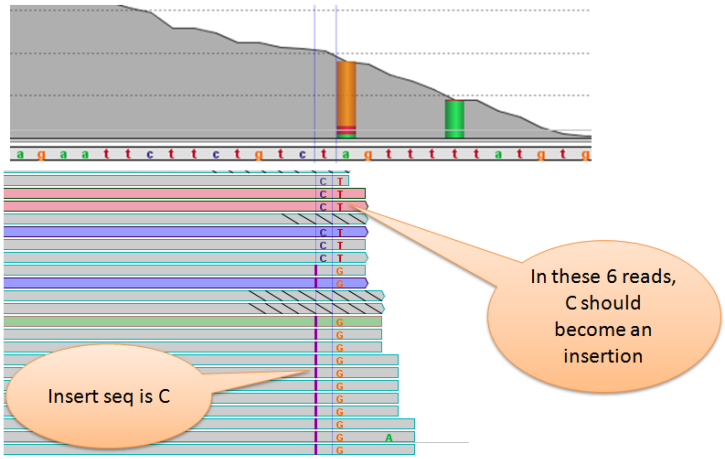

Figure 4.3: Example showing alignment artefacts in a potential genomic region for local realignment.

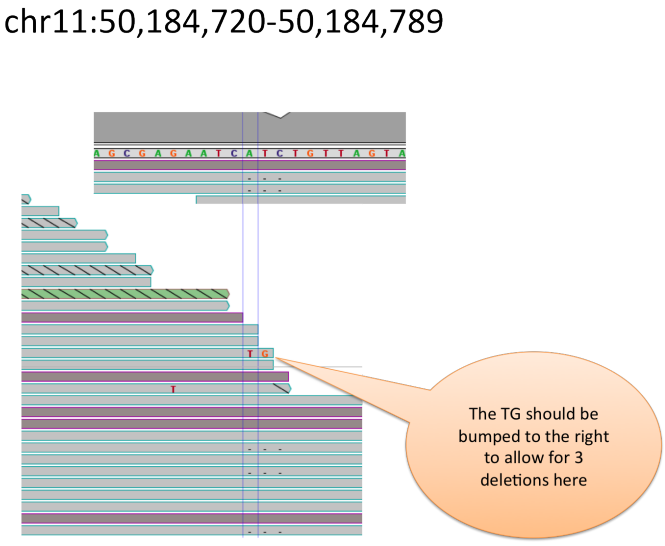

Figure 4.4: Example showing alignment artefacts in a potential genomic region for local realignment.

graph is built, all reads are realigned according to the variant graph. We will not discuss the $SRMA$ approach here as we have mostly adopted the ideas from $GATK$ and $Dindel$.

Below are the four basic steps that are used in our implementation:

- Identify regions of reads to be realigned
- Generate candidate haplotypes in each identified region
- Infer most likely haplotypes
- Realign reads to most likely haplotypes

In the following we will discuss each of the above four steps in more detail:

4.2.2.1 Identification of candidate regions for local realignment

The genome is first divided into windows of non-overlapping reads and then each window is considered separately to identify candidate regions for local realignment. In each window, we look at all the indels and evaluate local region around each one of them. Specifically, we check the following three things for each indel:

- *Minimum indel coverage:* Unlike the approach adopted by $GATK$, which considers all indels that are supported by at least one read, we require every indel to be supported by at least 5 reads. Indels that do not satisfy this criteria are discarded.
- *Mismatches in neighbourhood of indel:* For every indel that satisfies the 'minimum coverage', we check if it has at least one mismatch in its neighbourhood ($+/-20bp$).
- *Possibility of merging near-by indels:* Indels that satisfy the 'minimum coverage' requirement and have mismatches in it's neighbourhood, are further evaluated to assess the possibility of merging them into a single candidate region. If two indels are within a fixed distance l_m (called indel merging window width) apart, they will be considered in a single common region for realignment.

To output the final candidate regions for local realignment, we use a parameter called 'Indel effect boundary' window of length l_e, and determine left effect boundary and right effect boundary for each region (a region may comprise either single indel or multiple closeby indels) identified in the previous step. In the case of single indel region, length of the candidate region for local realignment is equal to l_e, with indel occurring in the middle of the effect boundary window. In the case of merged indel region, left effect boundary is governed by the leftmost indel (basically a distance of $l_e/2$ from the left indel) and right effect boundary is governed by the rightmost indel (a distance of $l_e/2$ from the right indel).

To further illustrate all the steps involved in the identification of candidate regions for local realignment, let us consider an example as shown in Fig. 4.5.

This figure shows 5 indels (indicated by red color) in the region. First, for each indel we check 1) minimum indel coverage; 2) mismatches in the local neighborhood; and 3) possibility of merging indels. Only those indels are chosen that satisfy both the criteria 1) and 2). Out of the five indels considered, 4^{th} indel does not satisfy the min coverage criteria, and hence discarded, 5^{th} indel had no mismatches in the local neighborhood, and hence discarded too. Of the remaining three indels, we can check the merging criteria. As shown in the figure, based on the window size used, first two indels can be merged together, while third one is by itself. Finally to obtain the candidate regions, for each region (whether merged or single), left and right effect boundaries are determined using the 'effect boundary' window as shown in the figure.

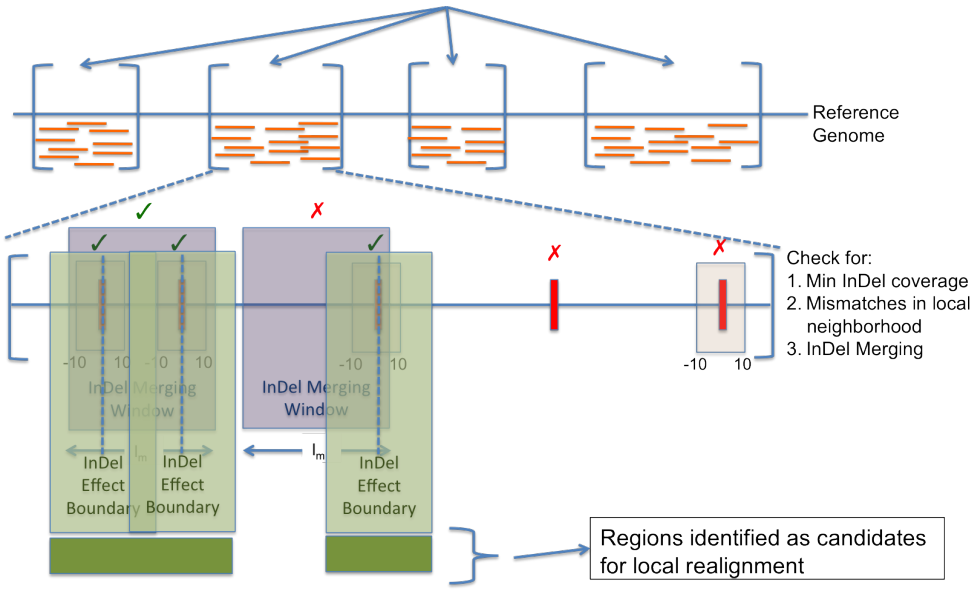

Figure 4.5: Identification of candidate regions for local realignment.

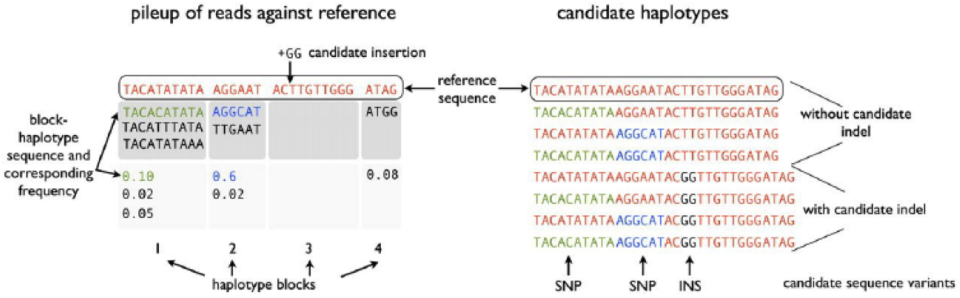

Figure 4.6: Candidate Haplotype Generation (Figure taken from Albers et. al., Genome Research 2010).

4.2.2.2 Generation of candidate haplotypes

Once we have the candidate regions for local realignment as identified in the previous step, we generate candidate haplotypes in each of the candidate regions. The steps are depicted in Fig. 4.6. First a partition set is created using information from start and end positions of all the reads that are aligned to this region. Then in each partition or sub-block, we count all possible haplotypes and record their frequency (this step is similar to Dindel approach [35]). This is repeated for all partitions and then eventually n block haplotypes with the highest empirical frequency are obtained. Using these n block haplotypes, we have at most 2^n candidate haplotypes. In addition, we allow known variants to be provided from outside and consider at most k variants per candidate region. In that case also, we consider all possible combinations and hence we have at most $2^k * 2^n$ candidate haplotypes that should be evaluated using the read data.

4.2.2.3 Inferring most likely haplotypes

Once we have obtained the candidate haplotypes, they are evaluated using all the reads. To do this, each read is aligned to each haplotype in a gapless manner and a score is calculated as shown in Eqn. 4.1. Assume R_j is the j^{th} read, H_i is the i^{th} haplotype, we define score, L as the following:

$$L(R_j|H_i) = \prod_k L(R_{j,k}|H_{i,k+s_{ji}}) \tag{4.1}$$

$$L(R_{j,k}|H_{i,k+s_{ji}}) = \begin{cases} 1-\epsilon_{j,k}, & R_{j,k} = H_{i,k+s_{ji}} \\ \epsilon_{j,k}, & R_{j,k} \neq H_{i,k+s_{ji}} \end{cases} \tag{4.2}$$

Here, $R_{j,k}$ is the k^{th} base of read R_j, $\epsilon_{j,k}$ is the error rate corresponding to the declared quality score for that base, and s_{ji} is the offset in the gapless alignment of R_j to H_i.

The L-scores are combined across all the reads to get the score for each haplotype as shown in Eqn. 4.3. This is similar to the approach followed in $GATK$. However, $GATK$ chooses only one alternate haplotype that yields the maximum score. In our approach, we evaluate all the haplotypes including the reference haplotype but choose the best two based on their scores. These selected two may or may not have the reference haplotype.

$$L(H_i) = \prod_j L(R_j|H_i) \tag{4.3}$$

4.2.2.4 Realigning reads using most likely haplotypes

Finally to realign the reads, we select the two best haplotypes H_{i_1}, and H_{i_2} (as mentioned earlier, unlike GATK, this may or may not include the reference haplotype) based on their scores. However reads are aligned to the two haplotype model instead of a single reference haplotype model only if the log odds ratio of the two-haplotype model is better than single haplotype model by at least 5 units, as shown in Eqn. 4.4

$$\frac{L(H_{i_1}, H_{i_2})}{L(H_o)} = \frac{\prod_j max(L(R_j|H_{i_1}), L(R_j|H_{i_2}))}{\prod_j L(R_j|H_o)} \tag{4.4}$$

4.3 Base Quality Recalibration

4.3.1 Background and Motivation

Base quality score is the confidence of the sequencer in calling a base. In other words, base quality reflects the probability of a sequencer making an error and calling incorrect base. Typically, base quality score is reported on a Phred scale as defined below:

$$\text{Base Quality Score} = -10 * log_{10}(p) \tag{4.5}$$

For example, a base quality score of 20 on a Phred scale means an error probability of 1 in 100, a score of 30 would mean an error probability of 1 in 1000, and so on. These sequencing errors may either result due to random sequencer noise or systematic biases. The systematic biases could be machine-, run-, or sample-specific. Bayesian variant caller, like the one used in Avadis NGS, uses the base qualities in calling out variants. Therefore inaccurate or biased base qualities may result in spurious variant calls. If base qualities are over-estimated (under-estimated), it may result in false positive (false negative) variant calls. This problem may not be severe in case of high coverage samples however for low coverage samples, recalibrating the base quality scores may help reduce spurious variant calls.

4.3.2 Types of Approaches

There are two types of approaches for base quality score recalibration.

- *Data driven empirical approach:* This is presented in the paper by DePristo et. al. [39] and implemented in the software suite GATK. In this approach, a recalibration table is built which can then be used to recalibrate base quality scores. To populate the table, this approach uses all the data. In fact, if multiple samples are available, it is recommended to pool the data.

- *Machine learning based logistic regression approach:* This is presented in the paper by Cabanski et. al. [38] and implemented in a R package called ''ReQON''. In this approach, a logistic regression model is learned and used to recalibrate the quality scores. For model training, 'ReQON' can train on a smaller set of data. In fact, for large training data, logistic regression model can be built on separate data chunks and then median regression coefficients can be used.

Based on the above, it is apparent that ReQON model has relatively solid theoretical underpinnings and can be used with smaller training data. However, the ease of implementation, usability and popularity of GATK's empirical approach, motivated us to adopt the empirical model in **Avadis NGS** for base quality score recalibration. Below we describe the empirical approach in more detail.

4.3.3 Empirical Approach for Base Quality Score Recalibration

Empirical method for base quality score recalibration as proposed in [39] identified the following four covariates on which base quality score depends upon.

- *Machine Cycle:* Based on the sequencer, machine cycle can either be represented as flow cycle (for e.g. 454 and Ion Torrent) or discrete cycle (for e.g. Illumina and PacBio). In flow cycle, each flow grabs all the TACG's in order in a single cycle. So consider for e.g. the sequence AAACCCCGAAATTTTTACTC, flow cycle would be 1 for the A's, C's and G's since they are in order of TACG. Next set of A's would be one cycle by itself. The subsequent T's, A, and C would be captured in the 3rd cycle, followed by TC being captured in the 4th cycle. In discrete cycle, machine cycle is simply the position in the read, which should be counted in the reverse direction for a negative strand read.

- *Nucleotide Context:* Another covariate of interest is nucleotide context, which looks into the sequence chemistry effects. The fundamental question here is what is the likelihood of mis-calling a base A, if it is preceded by A or T or G or C? In other words, is the error probability independent of the preceding base(s) or dependent upon it?. It turns out that preceding base makes a difference in the sequencing error of the called base. Although this effect can be studied for one or more preceding bases, we only consider di-nucleotide context i.e. the effect of one preceding base on the likelihood of the sequencing error of the called base. However, for homopolymer stretches, we use longer context, up to 8 bases. So we have a total of 40 combinations (16 for dinucleotides and 24 for homopolymer stretches).

- *Read Group:* Another covariate that can affect the base quality scores is read group. Sample ID is typically taken as the read group. For illumina, which has a lane concept, (sample ID + lane ID) can be used as read group.

- *Original Base Quality:* Finally reported original base quality is used as another covariate since the recalibrated base quality score, in addition to being dependent upon other covariates as discussed above, would of course depend upon this as well.

In **Avadis NGS** , by default the three covariates, namely machine cycle, nucleotide context, and the original base quality are used. Once the covariates are identified the empirical base quality score recalibration method involves two steps:

- *Empirical model training (i.e. Generating recalibration table):* In order to generate the recalibration table using the data, the basic assumption is to consider all mismatches as sequencing errors, except the mismatches at the locations that are known to vary, for e.g. in dbSNP. This is done to avoid true variations being considered as sequencing errors. For the remaining locations, we look at every base and categorise it in one of the bins according to the covariates described above i.e. original reported base quality, machine cycle, and nucleotide context. So, for three covariates, if we try to visualise these bins, it will result in a cube where each cell of the cube represent one bin, and with each bin consisting of several data points (base locations in this case). Finally to generate and populate the recalibration table, using these data points, for each cell we compute the number of mismatching bases and total number of bases. One can now easily appreciate why the availability of more data is useful for generating more accurate recalibration table.

$$mismatches(R, C, D) = \sum_{r \in R} \sum_{c \in C} \sum_{d \in D} b_{r,c,d} \neq b_{ref} \tag{4.6}$$

$$bases(R, C, D) = \sum_{r \in R} \sum_{c \in C} \sum_{d \in D} |b_{r,c,d}| \tag{4.7}$$

  Here capital letters R, C, and D denote the set of all reported base quality scores, machine cycle, and nucleotide context respectively. On the other hand, small letters r, c, and d denote specific instances or values of reported base quality scores, machine cycle, and nucleotide context respectively. b is used to denote the base call, and ref is used to denote the reference base.

  Once we have determined the number of mismatches and total bases in each cell, we can compute the empirical base quality scores, which is nothing but the ratio of mismatches and total bases, but corrected using Yates correction. Yates correction is done to avoid divide by 0 cases. So, we start with two bases in each bin and assume that one was a mismatch and other was not, therefore 1 would be added to the mismatch count and 2 would be added to the count of total bases.

$$Qempirical(R, C, D) = \frac{mismatches(R, C, D) + 1}{bases(R, C, D) + 2} \tag{4.8}$$

- *Recalibrate base quality scores using the generated recalibration table:* After generating a recalibration table as explained above, quality score of each base in every read can be recalibrated using the equation below:

$$\begin{aligned} recal(r, c, d) = &(Q_e(r, C, D)) + \\ &(Q_e(r, c, D) - Q_e(r, C, D)) + \\ &(Q_e(r, C, d) - Q_e(r, C, D)) \end{aligned} \tag{4.9}$$

  Here $recal(r, c, d)$ denotes the recalibrated base quality score given a specific value of original reported quality r, machine cycle c, and nucleotide context d. The right hand side terms are empirical qualities that were obtained in the first step as shown in Eqn. 4.8. These terms are nothing but the projections of the cube in 2 or 1-dimensional space. Capital letters indicate summation or marginalization and small letters indicate specific values of the variable.

  It is also interesting to note from this equation, especially the way it is represented that it is fairly easy to incorporate additional covariates.

4.4 SNP Detection

The SNP detection algorithm in **Avadis NGS** is capable of detecting three types of variants, namely substitution (or mutation), deletion and insertion. While reporting, these variants are categorized into four events namely substitution, insertion, deletion and complex. Substitution consists of one or more substitutions occurring in consecutive locations; similarly deletions and insertions comprise of events where one or more deletions or insertions, as the case may be, occur together. A deletion event is modeled as a mutation from a nucleotide value {A, T, G, C} to "gap". Complex events are reported when there is a mixture of multiple types of variants occurring together in consecutive locations.

The SNP detection algorithm works on a per-sample basis. For each sample, data for each chromosome is analyzed, with a capability for parallel computation per chromosome. The computation is optimized for memory usage by segregating the data into windows based on maximum read size.

4.4.1 Pre-processing: Selection of Potential SNP Sites

SNP identification is only performed at sites determined by pre-processing to be likely SNP locations. Locations are only considered potential SNP sites if they satisfy all the following requirements:

- **Read coverage threshold:** The number of reads covering the location must exceed a user-defined threshold.
- **Variant coverage threshold:** The number of reads covering the location and differing from the reference at the location, must also exceed a user-defined threshold.
- **Homopolymer effects:** In some technologies, reads covering homopolymers and adjacent nucleotides are more prone to errors in sequencing and alignment.
  - **Homopolymer threshold:** Homopolymers are determined by sequences repeating a single nucleotide value more than a threshold number of times; this is set by the user-defined homopolymer threshold. Currently, SNPs are not called at locations within a homopolymer stretch. In Fig. 4.7, the column shaded in pink lies at the end of a homopolymer of length 7 bases. At this position, both the Read Coverage (11) and Variant coverage (3 Ts and 2 gaps = 5 variants) satisfy the related default thresholds of 10 and 2 respectively. However, as the position lies within a homopolymer it will not be considered for SNP calls.
  - **Spillover threshold:** If the length of the homopolymer exceeds a certain user-defined spillover threshold, bases at locations immediately adjacent to the homopolymer and which are same as that present in the homopolymer are discarded. The examples given in Figures 4.7 and 4.8 illustrate this point. The column shaded blue in each case is adjacent to a homopolymer of length 7. This position is not within a homopolymer, and has sufficient read coverage as well. However, in Fig. 4.7, the variant coverage is only 1, contributed by the gap. The two reads with nucleotide G do not contribute to the variant count, as they are considered due to the spill-over effect of the adjacent homopolymer with 7 consecutive Gs. On the other hand, in Fig. 4.8, the variant coverage is 5, caused by 1 gap and 4 Cs.

4.4.2 Bayesian SNP Calling Algorithm

At every location declared significant in the pre-processing step, the Bayesian SNP calling algorithm is applied to identify the genotype at that location, and report the SNP if detected. The algorithm considers the nucleotide or the base taken by each read covering the location, as well as its

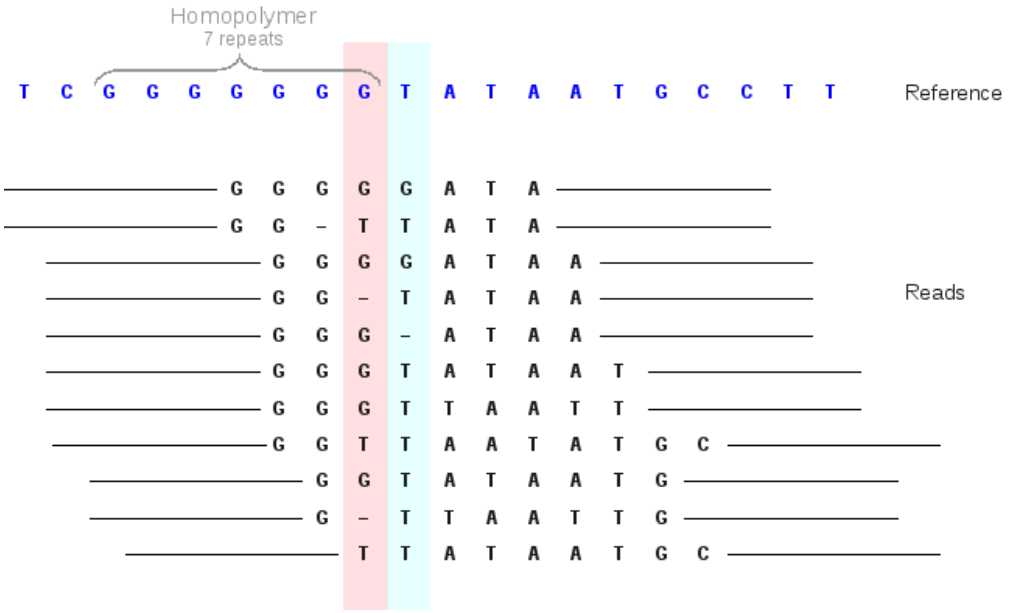

Figure 4.7: SNP calling close to a homopolymer

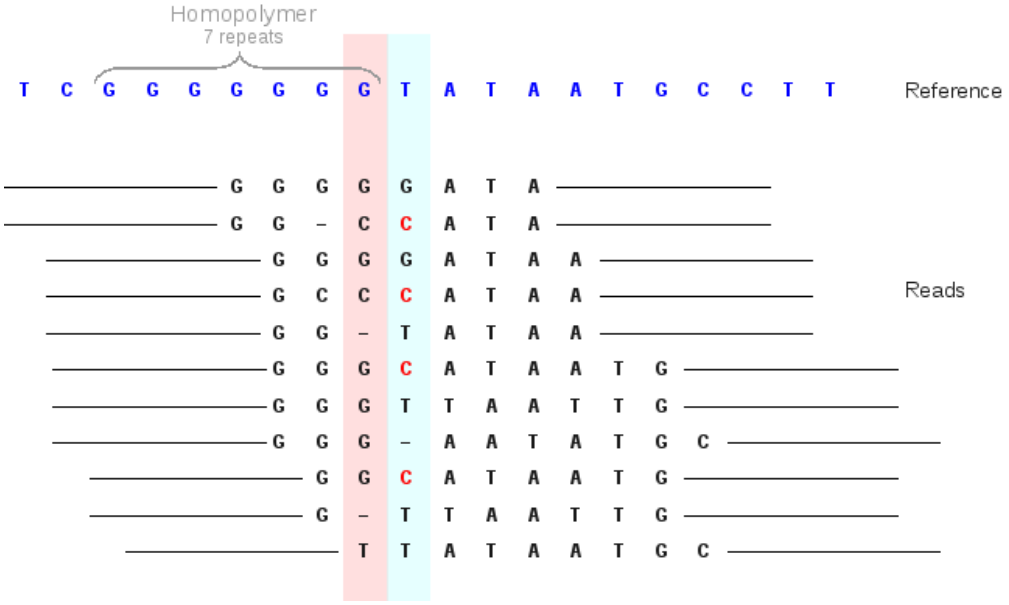

Figure 4.8: Heterozygous SNP adjacent to a homopolymer

associated base quality and finds the consensus genotype. The consensus genotype is the one most likely to have caused the observed data and is computed using Bayes principle.

If the observed data is collectively denoted by $\mathcal{D}$, then the task is to infer the consensus genotype from $\mathcal{D}$. SNPs are then detected by comparison to the reference sequence. The ploidy of the sample determines the number of nucleotide inferences necessary to infer the underlying genotype $\hat{\mathcal{G}}$. As **Avadis NGS** currently only handles diploid organisms, the number of nucleotides to be concluded (the length of vector $\hat{\mathcal{G}}$) is two.

The algorithm selects the genotype $\hat{\mathcal{G}}$ that maximizes the posterior probability, $P(\mathcal{G}|\mathcal{D})$, defined by the probability of a genotype $\mathcal{G}$ given the observed data. In other words, the genotype concluded is that most likely to have caused the particular observations

$$\hat{\mathcal{G}} = \text{argmax}_{\mathcal{G}} P(\mathcal{G}|\mathcal{D}).$$

This is easily computed under Bayes principle as

$$P(\mathcal{G}|\mathcal{D}) = \frac{P(\mathcal{D}|\mathcal{G}) \cdot \mathcal{P}(\mathcal{G})}{P(\mathcal{D})}$$

where $P(\mathcal{G})$ is the prior probability of genotype $\mathcal{G}$ and $P(\mathcal{D})$ is the likelihood of data, calculated over all possible genotypes,

$$P(\mathcal{D}) = \sum_{\forall \mathcal{G}_i} P(\mathcal{D}|\mathcal{G}_i) \cdot P(\mathcal{G}_i).$$

The conditional probability $P(\mathcal{D}|\mathcal{G})$ represents the probability of observing the data $\mathcal{D}$ under the given genotype $\mathcal{G}$. (See below for the details of the calculation of the prior and the conditional probabilities.)

Combining the above-mentioned equations, we arrive at the consensus genotype, given by

$$\hat{\mathcal{G}} = \text{argmax}_{\mathcal{G}} \frac{P(\mathcal{D}|\mathcal{G}) \cdot \mathcal{P}(\mathcal{G})}{\sum_{\forall \mathcal{G}_i} P(\mathcal{D}|\mathcal{G}_i) \cdot P(\mathcal{G}_i)}$$

Avadis NGS specifies a score to represent the confidence with which a SNP is called at a locus. This score is a function of the posterior probability of the locus being the reference genotype RR given the observed data and is defined as

$$Score = -10 log_{10}(P(\mathcal{G} = RR|\mathcal{D})).$$

If the consensus genotype is not the reference genotype, and if the above score is greater than the user-specified cut-off, then the location is reported as a variant location and the alleles corresponding to the consensus genotype which are different from the reference are reported as the variant alleles.

Calculation of the prior probabilities $P(\mathcal{G})$:
We calculate the prior probabilities of different genotypes at a given location by taking into account the following parameters. The approach is similar to that in [40].

- The reference base at the location
- Heterozygosity: This is representative of the prior expectation of a position being heterozygous in the organism. The default value is set to 0.001 which correlates with the population SNP rate for the human genome.
- Homozygous to heterozygous ratio: This is the prior expectation of the ratio of the number of homozygous mutations to the number of heterozygous mutations. It has been observed that heterozygous variants are twice as frequent as homozygous variants in any genome. Hence the default value is set to 0.5.

- Indel to substitution ratio: This is the prior expectation of the ratio of the number of indels to the number of substitutions. Based on the information from the literature, the default value is set to 0.125.

- Ti/Tv ratio: This is the prior expectation of the ratio of the number of transitions to the number of transversions. A transition is a point mutation that changes a purine nucleotide (A or G) to another purine, or a pyrimidine nucleotide (T or C) to another pyrimidine, whereas a transversion refers to the substitution of a purine for a pyrimidine or vice versa. Although there are twice as many possible transversions, because of the molecular mechanisms by which they are generated, transition mutations are generated at higher frequency than transversions. Expected value of this ratio for genome wide variations is in the range 2.0-2.1, whereas it is in the range $3.0 - 3.3$ for the exonic variations. The default value for this parameter is set to 2.6 to make it suitable for both whole genome and exome data analysis.

The tables below give the prior probablities computed for the default values of the parameters mentioned above. Table 4.1 gives the probabilities assuming the reference base is A, whereas Table 4.2 gives the same when the reference base is a gap (this corresponds to an insertion in the read). In these tables, the rows and the columns correspond to the two different alleles of the genotype. Thus, for example, the prior probability for the genotype AT is given in row A and column T. Of course, you will find the same value in row T and column A also.

	A	C	G	T	-
A	9.984×10^{-1}	2.174×10^{-4}	5.652×10^{-4}	2.174×10^{-4}	6.250×10^{-5}
C	2.174×10^{-4}	1.087×10^{-4}	1.229×10^{-7}	4.726×10^{-8}	1.359×10^{-8}
G	5.652×10^{-4}	1.229×10^{-7}	2.826×10^{-4}	1.229×10^{-7}	3.533×10^{-8}
T	2.174×10^{-4}	4.726×10^{-8}	1.229×10^{-7}	1.087×10^{-4}	1.359×10^{-8}
-	6.250×10^{-5}	1.359×10^{-8}	3.533×10^{-8}	1.359×10^{-8}	3.125×10^{-5}

Table 4.1: Prior probability of genotypes of a diploid genome when Ref = A

	A	C	G	T	-
A	7.813×10^{-6}	2.441×10^{-10}	2.441×10^{-10}	2.441×10^{-10}	1.563×10^{-5}
C	2.441×10^{-10}	7.813×10^{-6}	2.441×10^{-10}	2.441×10^{-10}	1.563×10^{-5}
G	2.441×10^{-10}	2.441×10^{-10}	7.813×10^{-6}	2.441×10^{-10}	1.563×10^{-5}
T	2.441×10^{-10}	2.441×10^{-10}	2.441×10^{-10}	7.813×10^{-6}	1.563×10^{-5}
-	1.563×10^{-5}	1.563×10^{-5}	1.563×10^{-5}	1.563×10^{-5}	9.999×10^{-1}

Table 4.2: Prior probability of genotypes of a diploid genome when Ref = -

Computation of the conditional probabilities $P(\mathcal{D}|\mathcal{G})$:
The computation of $P(\mathcal{D}|\mathcal{G})$ takes into account the probability of errors observed in the data under each genotype $\mathcal{G}$. It is assumed that the error in the observation of any base is independent of observation error in any other base. In most cases, error rates are derived from the associated base qualities as follows.

$$\epsilon = 10^{-(\text{base quality})/10}$$

The error rates may be adjusted by taking the mapping quality into consideration. A mis-aligned read creates unreliable results irrespective of the base-qualities. Therefore, **Avadis NGS** has an option to take into account the mapping qualities whenever provided by the aligner and uses the minimum of the read mapping quality and individual base quality in place of base quality for its computations.

In the absence of error rates, hard-coded error rates (that are technology-specific) are used. If error rates for a particular technology are not present in **Avadis NGS**, it assumes a quality score of 30 for the base, which is interpreted as an error rate of 0.001.

If the numbers of As, Ts, Cs, Gs, and $-$s in the reads at the locus being considred are n_a, n_t, n_c, n_g, and n_-, respectively, then the conditional probability $P(\mathcal{D}|\mathcal{G})$ can be broken down as follows.

$$\begin{aligned} &P(\mathcal{D}|\mathcal{G}) \\ &= k\left[P(B=A|\mathcal{G})\right]^{n_a}\left[P(B=T|\mathcal{G})\right]^{n_t}\left[P(B=C|\mathcal{G})\right]^{n_c}\left[P(B=G|\mathcal{G})\right]^{n_g}\left[P(B=-|\mathcal{G})\right]^{n_-} \end{aligned}$$

where B is the read base at the locus under consideration, and k is the number of distinct permutations of the different read bases at that locus, and is equal to $\frac{(n_a+n_t+n_c+n_g+n_-)!}{(n_a!)(n_t!)(n_c!)(n_g!)(n_-!)}$. Note that this factor will be present in both the numerator and the denominator in $P(\mathcal{G}|\mathcal{D})$. So the final result doesn't require this to be calculated.

If $\mathcal{G}$ is homozygous, the probability of observing each of the bases can be calculated as follows.

$$P(B=b|\mathcal{G}=<g,g>) = \begin{cases} 1-\epsilon, & \text{if } b \text{ is the same base as } g \\ \frac{\epsilon}{3}, & \text{if } b \text{ is not present in } \mathcal{G} \end{cases}$$

where b is the observed base.

For heterozygous genotypes $\mathcal{G}=<g_1,g_2>$, the probability is calculated as follows.

$$\begin{aligned} &P(B=b|\mathcal{G}=<g_1,g_2>) \\ &= \frac{1}{2}\times P(B=b|\text{Read came from chromosome 1}) + \frac{1}{2}\times P(B=b|\text{Read came from chromosome 2}) \\ &= \begin{cases} \frac{1}{2}(\frac{\epsilon}{3})+\frac{1}{2}(1-\epsilon), & \text{if } b \text{ is one of the bases in } \mathcal{G} \\ \frac{1}{2}(\frac{\epsilon}{3})+\frac{1}{2}(\frac{\epsilon}{3}), & \text{if } b \text{ is not present in } \mathcal{G} \end{cases} \end{aligned}$$

4.4.3 Further Notes on SNP Detection

- **Calling out Multi Base Variants** MNP identification currently does not use a different algorithm; adjacent SNPs are grouped together and called MNPs. Note that the associated score is the minimum of the scores for all individual SNPs comprising the MNP.
- **Calling Single-deletions** Single-deletions are also output as part of the SNP algorithm. This is due to the fact that aligners introduce "gaps", representing deletions into reads for better mapping. Thus at each position a read may take one of 5 values: A, T, G, C, - (gap). The gap is treated exactly as a nucleotide value and in case of a preponderence of gaps at a location, a deletion is called. Since gaps do not have a quality score associated with them, the average of the surrounding base qualities is used for the computation.
- **Calling Insertions** The problem with calling out insertions is that insertions can be of any length and there might be more than four different types of variants at a single locus (which is always between two reference bases) unlike in the case of substitutions. Hence the normal method of variant detection cannot be applied in this case.

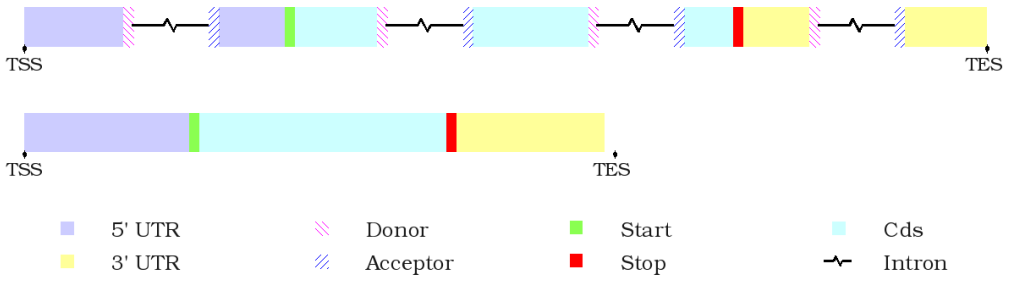

Figure 4.9: Transcript structure

Here is what is done: All the insert sequences are grouped into a single type and the probability that the location is a variant is calculated. If the score computed from this probability is above the specified cut-off, a variant is declared. Then the top two insert sequences are taken and the final genotype is the combination of these insert sequences that gives the best score.

4.5 SNP Effect Analysis

SNP Effect Analysis investigates a list of SNPs and generates a report indicating the effect that these SNPs have on the genes in a given annotation.

A transcript model has physical units called exons that are separated by introns. The transcript also has logical units called $5^{'}$ UTR, Coding region (Cd), $3^{'}$ UTR. The final protein is created by the coding regions and show no contribution from the UTRs (untranslated regions). The first base of the $5^{'}$ UTR is called the Transcription Start Site (TSS). The last base of the $3^{'}$ UTR is called the Transcription End Site (TES).

The first two and last two bases of each intron are very important for splicing. These sites are called Donor and Acceptor sites. Fig. 4.9 shows a transcript structure with the important components.

The coding region is sequence of codons -- each of which is 3 nucleotides long. The length of the coding region is therefore a multiple of 3. The first codon in the coding region is called the start codon and the last is called the stop codon. Each codon is translated into an amino acid.

Classifying mutations according to annotation
Mutations could occur anywhere in the chromosome. As their functional impact depends on how they affect protein formation, they may be classified accordingly.

- INTERGENIC Mutation: A mutation that does not fall within the neighborhood of any gene in the annotation. (A neighborhood is defined based on a chosen value of δ.)
- UPSTREAM Mutation: Mutation occuring upstream of the transcript, i.e., within co-ordinates $[TSS - \delta, TSS]$ (for +ve strand) or $[TSS, TSS + \delta]$ (for negative strand)
- DOWNSTREAM Mutation: Mutation occuring downstream of the transcript: say within genomic co-ordinates $[TES, TES + \delta]$ (for +ve strand) or $[TES - \delta, TES]$ (for negative strand)
- INTRONIC Mutation: Mutations occuring in intronic regions.
  - ESSENTIAL SPLICE SITE Mutation: Mutations to the donor and acceptor sites of the intron (i.e. two bases into the intron from either side)

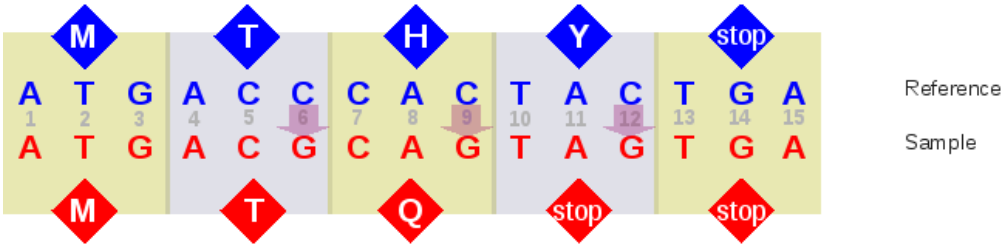

Figure 4.10: Mutations in a coding region

  - SPLICE SITE Mutation: Mutations to locations of splicing signals (i.e. 3-8 bases into the intron from either side, 1-3 bases into neighbouring exon).

- 5 PRIME UTR Mutation: Mutation in the $5^{'}$ region.
- 3 PRIME UTR Mutation: Mutation in the $3^{'}$ region.
- Coding Region Mutation: Mutations in the coding region are of two types.
  - SYNONYMOUS CODING Mutations: Mutations that have no effect on the final amino acid sequence (e.g. change of TTT codon to TTC codon still results in the Phenylalanine(F) amino acid). These are called synonymous changes.
  - Mutations that are non-synonymous have an effect on the amino acid sequence and are classified accordingly.

Classifying non-synonymous mutations
Fig. 4.10 shows a sequence from the coding region of a transcript and its comparison against a reference. The sequence has been split into codons for ease of explanation. Above the reference sequence codons are written the corresponding amino acids from the codon translation table. Similarly amino acids corresponding to the sample sequence codons are written directly below it.

The SNP detection algorithm correctly identifies SNPs at positions 6, 9 and 12. All involve a change from C to G. One can see that each of the C $\rightarrow$ G SNPs has a different effect on the amino acid sequence. The first change at position 6 causes no change in the amino acid sequence and so is termed SYNONYMOUS CODING. The second mutation is termed NON SYNONYMOUS CODING since amino acid changes from H to Q. The final change is termed STOP GAINED since a stop codon has been introduced before the original stop codon in the reference.

Similarly, a change that mutated the reference stop/start codon would be classified as STOP LOST/START LOST.

4.5.1 Challenges in SNP Effect Analysis

In this section we discuss a few of the challenges that present themselves during SNP Effect Analysis.

- **Sequential processing of SNP reports:** Effect Analysis typically processes SNPs sequentially, one by one. As long as the SNPs are single base pair mutations that do not occur too close to each other, the effect of each SNP can be individually considered. However, insertions, deletions, and multiple changes within a codon complicate the analysis.
  - **Multiple SNPs close to each other:** This can cause erroneous classification of mutations. For example, two SNPs in a codon taken together may result in a NON SYNONYMOUS CODING classification. However, when processed one at a time, each SNP may be considered as being a SYNONYMOUS CODING mutation.

Mutation Location	Classification
Coding region	START LOST STOP GAINED STOP LOST FRAMESHIFT CODING NON SYNONYMOUS CODING SYNONYMOUS CODING SPLICE SITE
Intronic	ESSENTIAL SPLICE SITE INTRONIC SPLICE SITE
Genic regions	5PRIME UTR 3PRIME UTR SPLICE SITE
Non-genic regions	UPSTREAM DOWNSTREAM INTERGENIC
Miscellaneous	EXONIC (Coding region information absent) NEAR GENE (Strand information for a gene absent) GENIC (Transcript information for a gene absent) COMPLEX VARIATION (MNP overlaps a boundary region OR two or more variations occur concurrently in an MNP)

Table 4.3: Mutation classification based on region

  - **Indels:** Any indel which is not a multiple of 3 are marked FRAMESHIFT CODING as they cause a cascading change in the amino acid sequence. Indels that are multiples of 3 are considered NON SYNONYMOUS CODING mutations. However, if they appear as two separate entries in the SNP report, sequential processing could cause them to erroneously be reported as FRAMESHIFT CODING mutations.
  - **Several types of variation occuring in a single MNP:** If an MNP simultaneously exhibits more than one type of mutation, for example, an insertion and two SNPs, effect analysis becomes complicated. In this case, the effect is simply termed COMPLEX VARIATION.
  - **Boundary region issues:** If the locations involved in an MNP overlap boundary regions, so that part of the MNP would be classified in one way and part in another, the effect is again called COMPLEX VARIATION.

- **Ploidy:** Further complications arise from the fact that a single chromosome is not being sequenced. It is possible therefore to have two SNPs one from each copy of the chromosome. Considering the combined effect of the two SNPs would be erroneous in this case.

- **Missing information:** If transcript annotations do not specify the coding region, the $5'$ UTR, $3'$ UTR, and Cds are not known. The only calls that can be made are - EXONIC (if it falls in an exon), INTRONIC, ESSENTIAL SPLICE SITE, SPLICE SITE, UPSTREAM, DOWNSTREAM, INTERGENIC.

  If strand information is not available for a particular gene, then UPSTREAM/DOWNSTREAM calls also cannot be made. Instead, these calls are termed NEAR GENE mutations.

  In case a gene is identified by the annotation but all transcript information for the gene is missing, intronic and exonic regions are not known. In this case, the call is GENIC.

Table 4.3 gives a list of mutation classifications based on the location of the mutation.

Chapter 5

Large Structural Variants Analysis

5.1 Introduction

Structural Variations (SVs) are differences in the DNA structures of individuals over base pairs numbering from a few hundred to a million. These could be inclusions, exclusions or rearrangements of portions of DNA on individual chromosomes. Such determinations are always made in comparison to a given reference DNA structure. At present we consider the following kinds of structural variations. *Note that the figures and discussions included in this chapter use Illumina paired-end (Forward-Reverse) reads as illustrations. The same arguments apply to other next-generation sequencing technologies as well.*

1. **INDEL**: This term is an amalgam of INsertions and DELetions in a DNA sequence when compared to a reference.

   - **Insertion**: Presence of additional base pairs in a DNA sequence.
   - **Deletion**: Absence of a sequence of base pairs in a DNA sequence.

   Fig. 5.1 illustrates single and multiple insertion and deletion events. Insertions are highlighted in blue and deletions in beige. (The figure also depicts substitution events highlighted in pink. See Chapter 4 for more details about these.)

2. **Inversion**: Reversal of a part of a DNA sequence, with the positions of the previously positive and negative strands interchanged and reversed as in Fig. 5.2. Note that the inverted segment re-attaches in a way that preserves the $5'$ -- $3'$ orientation.

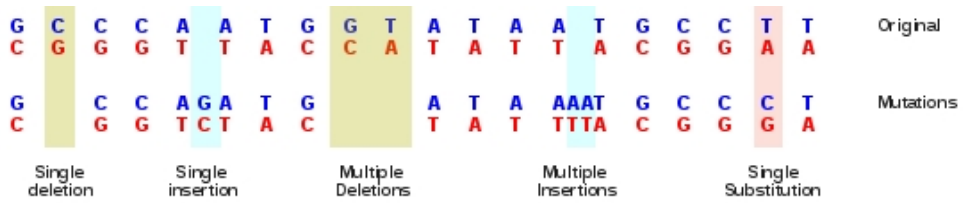

Figure 5.1: Insertions and deletions in DNA

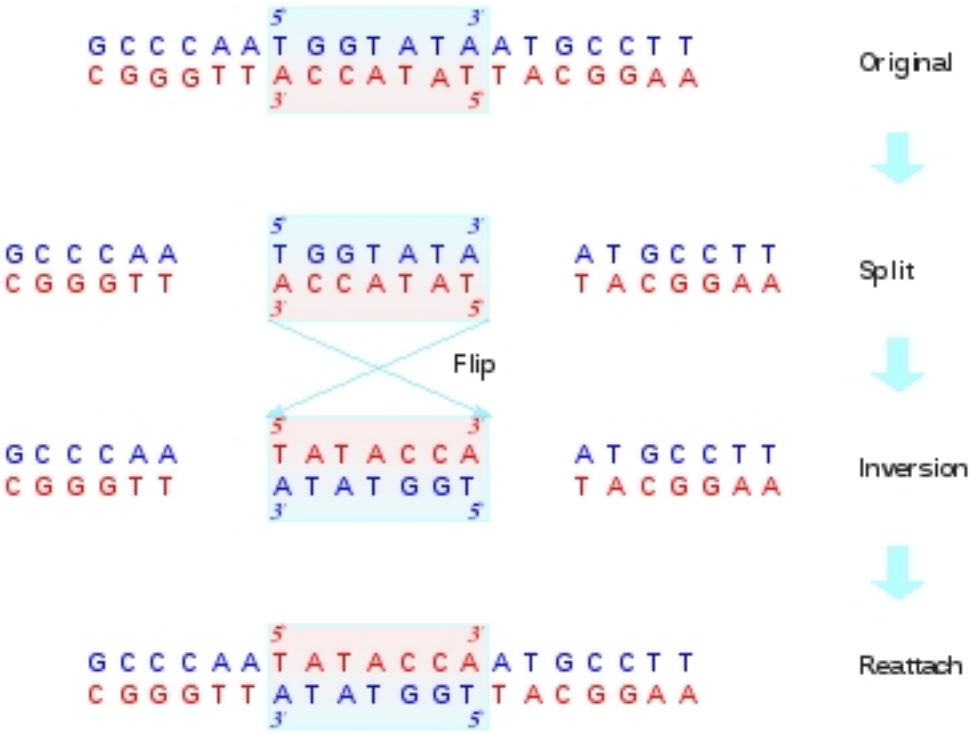

Figure 5.2: Inversion events in DNA

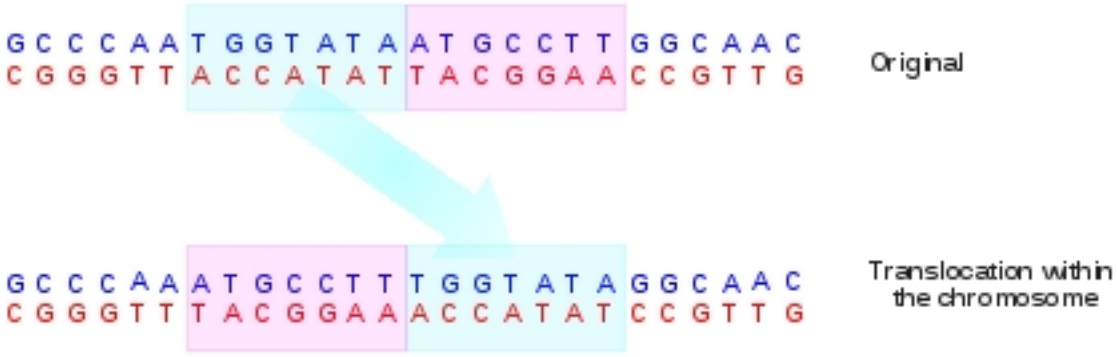

Figure 5.3: Intra-Chromosomal Translocation

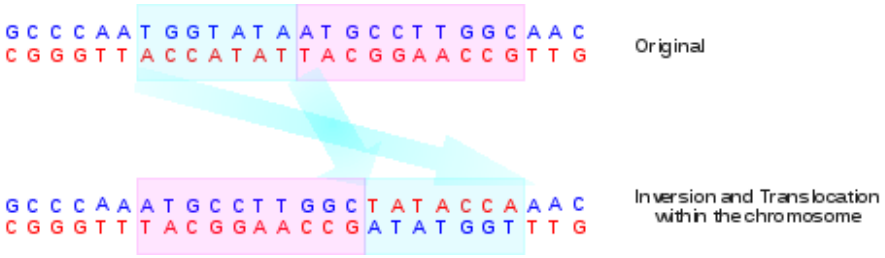

Figure 5.4: Intra-Chromosomal Inverted Translocation

3. **Translocation**: Deletion of a portion of DNA at a location and its insertion elsewhere. This could happen within a chromosome or even across chromosomes. Fig. 5.3 illustrates a translocation phenomenon.

4. **Inverted Translocation**: When a DNA sequence detaches itself from a chromosome, inverts itself and re-attaches to another portion of the same chromosome or to a different chromosome, the resulting structural variant is called an inverted translocation. Fig. 5.4 illustrates such a phenomenon.

Identification of SVs is fairly complex and can only be achieved via paired-end or mate-pair reads. See Section 1 for more detail. By varying the fragment lengths, we gain different insights into the genome sequence, especially with respect to SV identification. As the library fragment length distribution (assumed to be Gaussian with a given mean and variance) as well as the relative orientation and order of the mates is known a priori, deviant behavior is indicative of an underlying SV.

Each technology employs a different protocol during mate-pair and paired-read generation. The protocols are briefly described in Table 1.1 for completeness.

The primary challenges in SV prediction involve identification and clustering of deviant reads and then analysis and interpretation of the underlying cause of deviance. Ploidy issues introduce additional complexity, as each copy of the chromosome can differ in structure. Thus some fraction of reads from a region may indicate one kind of variant, while the rest conform with the reference.

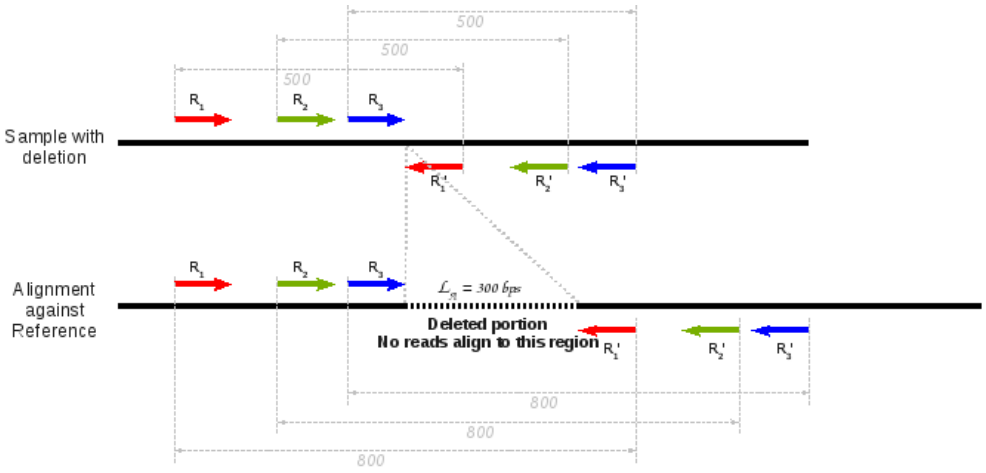

Figure 5.5: Read behavior in the presence of a deletion

5.1.1 Read characteristics in the presence of underlying structural variation

In this section, we will explore deviant read behavior caused by SVs. Each SV introduces trademark peculiarities in the pair orientations and inter-mate distance. These may be used to typify the SV and deduce its location.

5.1.1.1 Deletions

If a portion of DNA is missing from the sample, reads in the adjoining regions will show a longer inter-mate distance than was originally introduced during library preparation. Fig. 5.5 clearly illustrates this effect. The dotted portion of size 300bps in the reference is absent in the sample, representing a deletion. Reads immediately to the left of the deletion are mapped to the reference as expected, however, their mates align to the reference with a shift of 300 bps to the right. Thus the increase in the inter-mate distance is clearly linked to the length of the deletion. Furthermore, no reads align to the deleted portion on the reference in homozygous condition.

Note that if the deleted region is very small (on the order of a standard deviation of the original library fragment length), the increase in inter-mate distance may not be noticeable and the deletion may go undetected.

5.1.1.2 Insertions

If a portion of DNA is inserted into the sample in comparison to the reference, reads in the adjoining regions will show a shorter inter-mate distance than expected. As seen in Fig. 5.6, 400 extra base pairs are present in the sample DNA. Reads to the near left of the insertion, such as R_1,and R_6, map to the reference as expected, but their mates $R_1^{'}$, and $R_6^{'}$ align 400 bps earlier than anticipated. The decrease in the inter-mate distance for such reads relates directly to the length of the insertion. Furthermore reads such as R_2, R_4, $R_3^{'}$ and $R_5^{'}$ that directly lie within the inserted region, will not map to the reference at all; thus their mates $R_2^{'}$, $R_4^{'}$, R_3 and R_5 show up as unmatched reads.

Again in case of a very small insertion (on the order of a standard deviation of the original library fragment length), the decrease in inter-mate distance may be insignificant and the insertion may be ignored. In case of large insertions, specifically, where the length of the insertion exceeds the

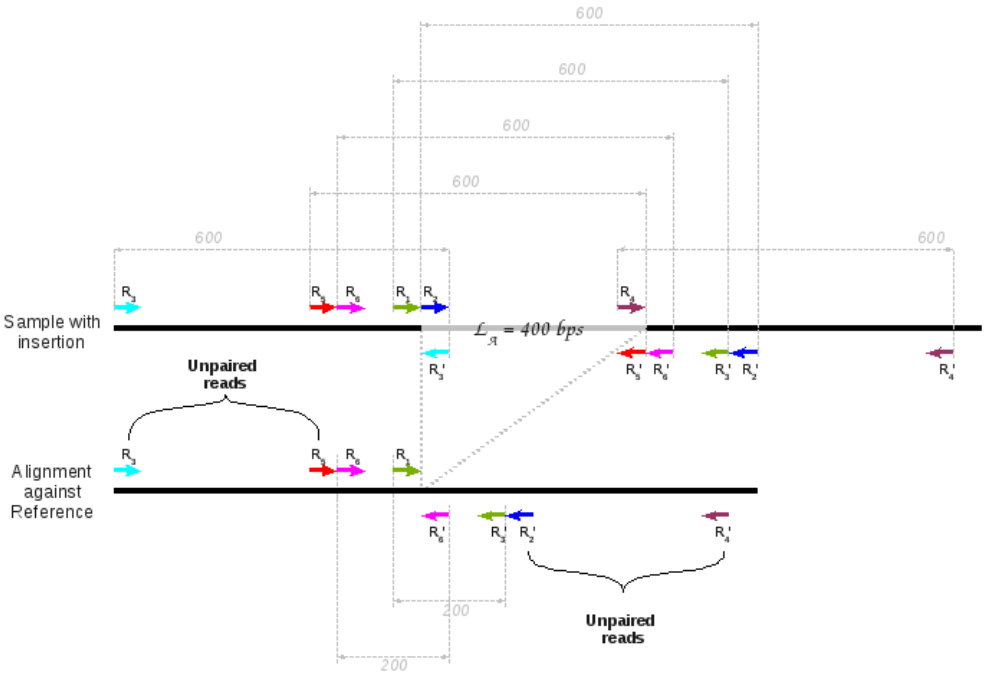

Figure 5.6: Read behavior in the presence of an insertion

inter-mate distance, all reads to the near left or near right of the insertion have mates falling within the insertion. As a result, all of these end up unmatched upon alignment to the reference. Reads R_1, R_3, $R_5^{'}$ and $R_6^{'}$ in Fig. 5.7 illustrate this point. Of course read-pairs such as R_2-$R_2^{'}$, and R_4-$R_4^{'}$ are discarded as unmapped reads, since they lie within the inserted region.

5.1.1.3 Inversions

When a portion of DNA detaches from the chromosome and re-attaches with reversed orientation, the mutation is termed an inversion. Fig. 5.8 illustrates such an instance. Note that the negative strand of the inverted portion now attaches to the positive strand of the chromosome and vice versa. Thus all reads lying in the inverted region change orientation. This effect is illustrated in Fig. 5.8. Reads R_1, R_2, R_4 and R_6 have one mate lying within the inversion, and the other outside. When mapped to the reference, the orientation of the mates is no longer Forward-Reverse as per the Illumina protocol. See Table 1.1. It now changes to forward-forward or reverse-reverse. Note that the inter-mate distance for these reads changes as well; it may increase (as in the case of R_2 and R_4) or decrease (as for R_1 and R_6), depending on the position of the read in the inverted area. For reads such as R_3 and R_5, where both mates lie in the inversion, the orientation is now switched, but the inter-mate distance remains unchanged. As the orientation is still compatible with Illumina formats (see Chapter 1 for details), these will not be identified as deviant reads.

5.1.1.4 Translocations

Read behavior in the presence of translocations depends on whether the segment translocates within the same chromosome or onto a different chromosome. Inter-chromosomal translocations are simpler to detect as the mates of some of the reads close to the translocated region map to a different chromosome. On one chromosome, the loss of DNA appears as a deletion; on the other, the introduction of the segment shows up as an insertion. In Fig. 5.9 a segment from

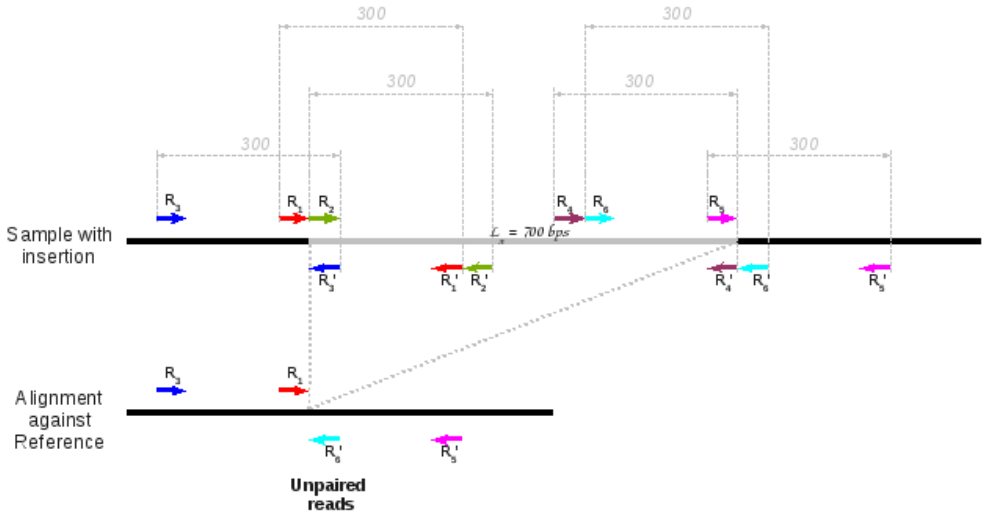

Figure 5.7: Read behavior in the presence of a large insertion

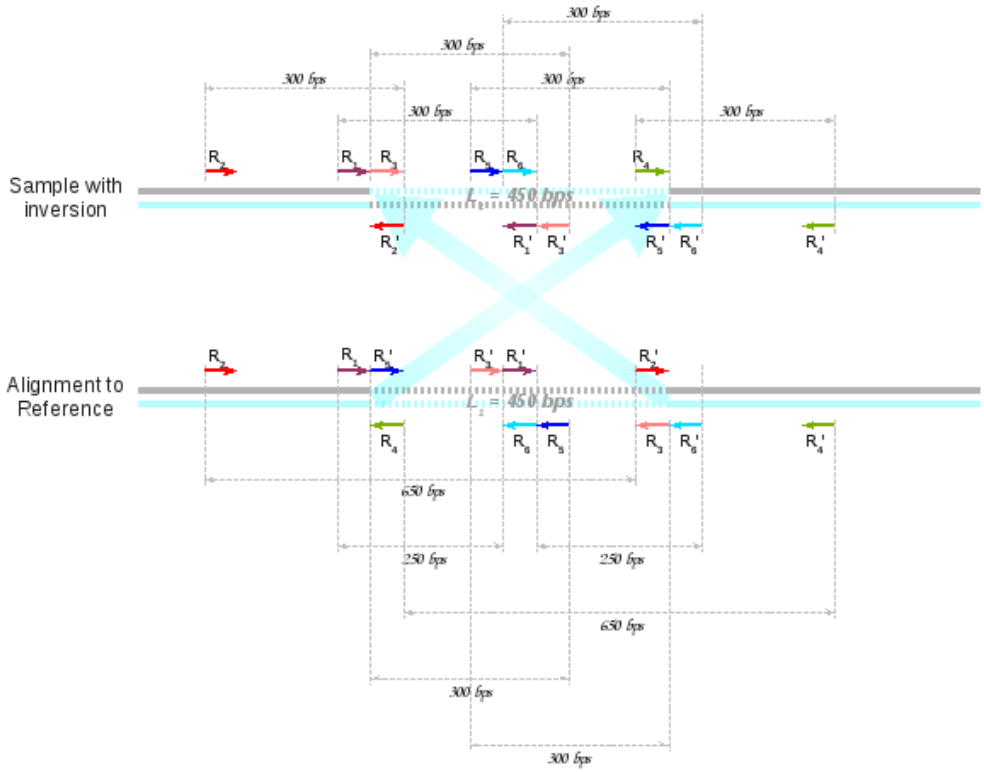

Figure 5.8: Read behavior in the presence of an inversion

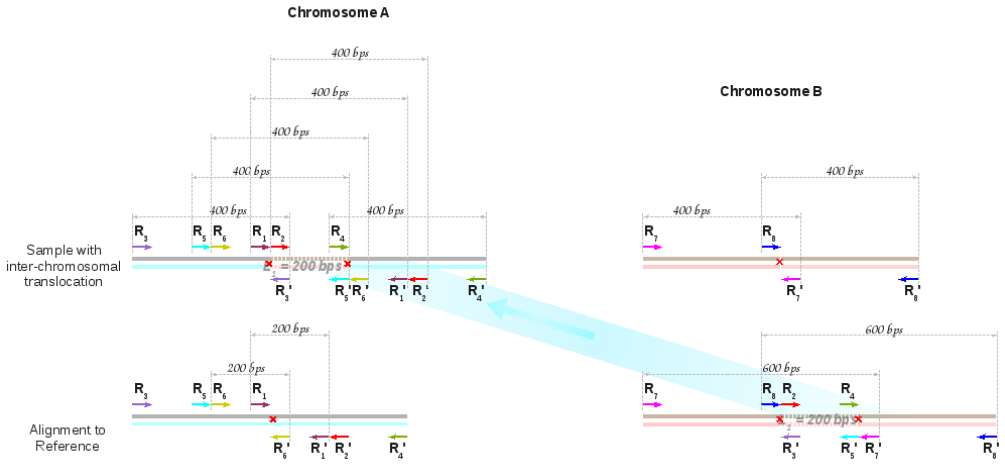

Figure 5.9: Read behavior in the presence of an inter-chromosomal translocation

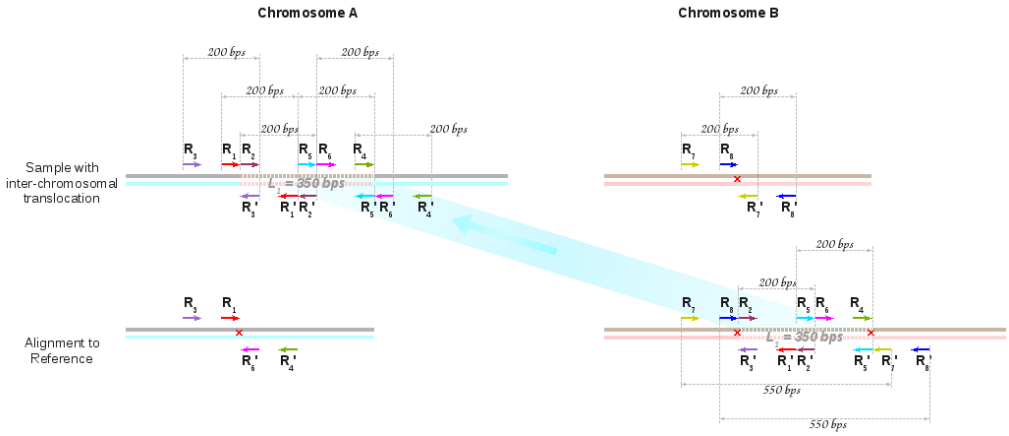

Figure 5.10: Read behavior in the presence of a large inter-chromosomal translocation

Chromosome B gets detached and re-attaches itself to Chromosome A. Reads R_2 and R_4 get mapped to Chromosome B while their mates align to Chromosome A. Similarly, reads R_3 and R_5 map to Chromosome A and their mates to Chromosome B. Reads R_1 and R_6 map to Chromosome A with a decreased inter-mate distance of 200 bps in place of the original library insert length of 400 bps, indicating an insertion on Chromosome A. On the other hand reads R_7 and R_8 map with an increased inter-mate distance of 600 bps in place of the original library insert length of 400 bps, suggesting a deletion on Chromosome B. If the translocated region is large, as in Fig. 5.10, the insertion on Chromosome A manifests in the form of several unmapped reads in the vicinity of the translocation (such as R_1, R_3, $R_4^{'}$ and $R_6^{'}$), instead of mates with decreased inter-mate distance.

Note that translocations of very small regions however, might show up as two unrelated events: An insertion on one chromosome and a deletion on another.

Intra-chromosomal translocations involve a shift along the same chromosome as shown in Fig. 5.11. Note that a translocation along the same chromosome can be viewed as a lateral shift of two regions (both shown by dotted lines), one to the right and the other to the left. Paired-end reads that lie entirely within or entirely outside these regions, such as reads R_2 - $R_2^{'}$, and R_5 - $R_5^{'}$ conform to the expected read behavior. Reads in the left-shifting region with a mate in the right-shifting

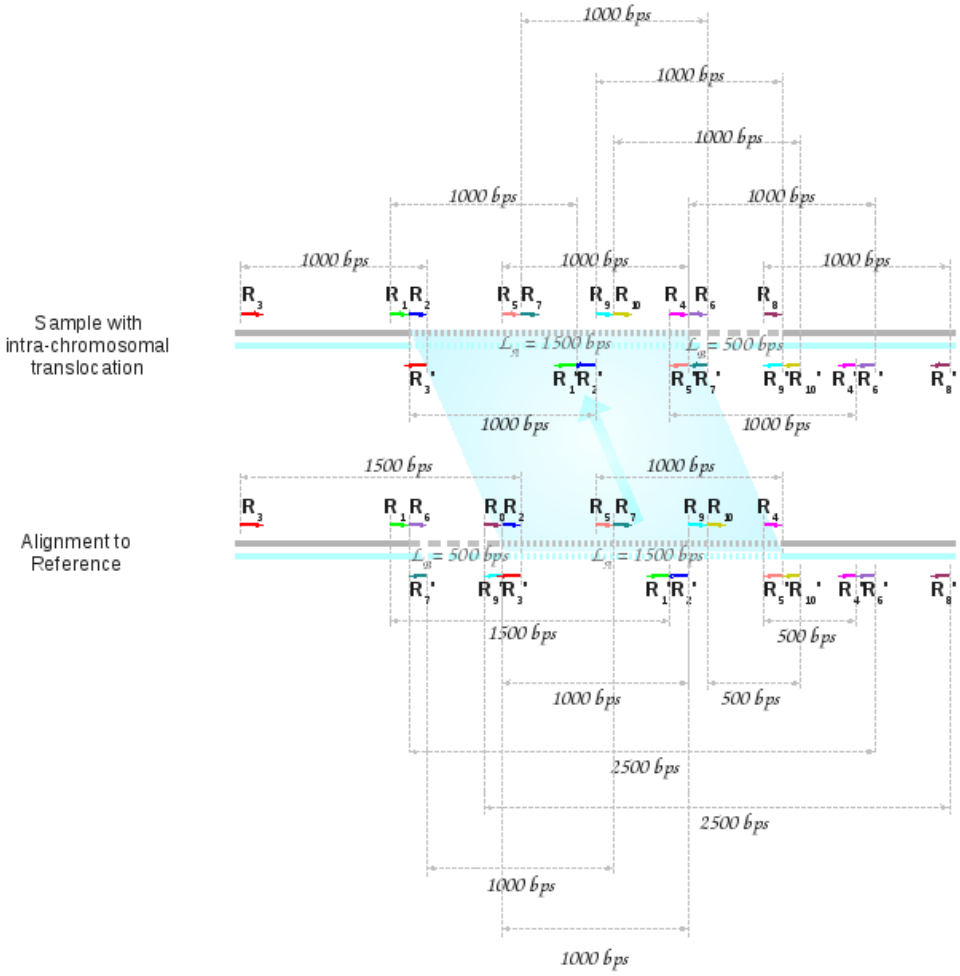

Figure 5.11: Read behavior in the presence of an intra-chromosomal translocation: Small shift

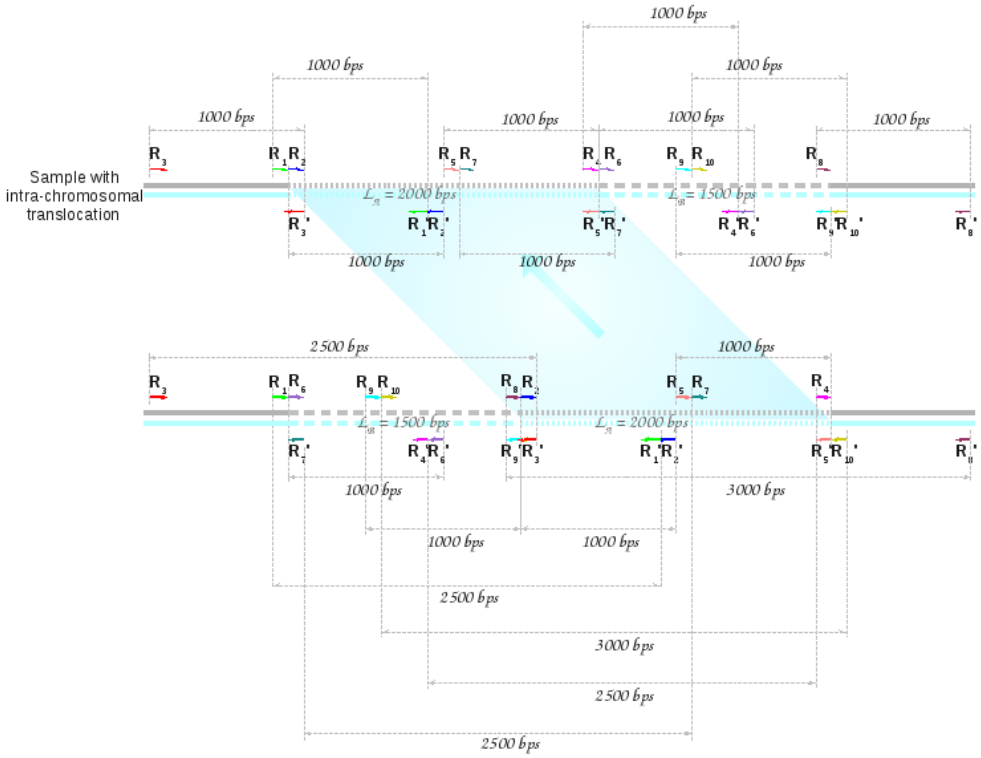

Figure 5.12: Read behavior in the presence of an intra-chromosomal translocation: Large shift

Relation between L_A, L_B and D	Reads Observed			
	Normal	Insertion	Deletion	Translocation
$D \leq L_A, L_B$	✔	✘	✔	✔
$L_A \leq D \leq L_B$ or $L_B \leq D \leq L_A$	✔	✔	✔	✔
$L_A, L_B \leq D \leq L_A + L_B$	✘	✔	✔	✔
$L_A + L_B \leq D$	✔	✔	✔	✘

Table 5.1: Intra-chromosomal translocation: Read patterns observed

region, such as R_7 - $R_7^{'}$ and R_9 - $R_9^{'}$ could deviate both in orientation and inter-mate distance. Reads with one mate in a translocating region and one in a region that does not experience a shift, conform in orientation, but may exhibit a larger or smaller inter-mate distance than expected, commensurate with a deletion or insertion respectively. In Fig. 5.11, reads R_1 - $R_1^{'}$, R_3 - $R_3^{'}$, R_6 - $R_6^{'}$ and R_8 - $R_8^{'}$ imply deletions, while R_4 - $R_4^{'}$ and R_{10} - $R_{10}^{'}$ are suggestive of an insertion.

Note that while reads denoting deletions are always observed, those indicating insertions are not always present. Fig. 5.12 demonstrate such a case. Here deletions may be inferred at both ends of the region experiencing shift, but there is clearly no read suggesting an insertion. In fact, the read patterns observed depend on the relative sizes of the translocated region L_A and its shift L_B in comparison to the original library insert length D, between mates. Table 5.1 lists the reads observed in each situation. We follow a convention where the region on the right is considered the translocated region, and the region on the left, the shift. The same arguments hold in the reverse situation as well.

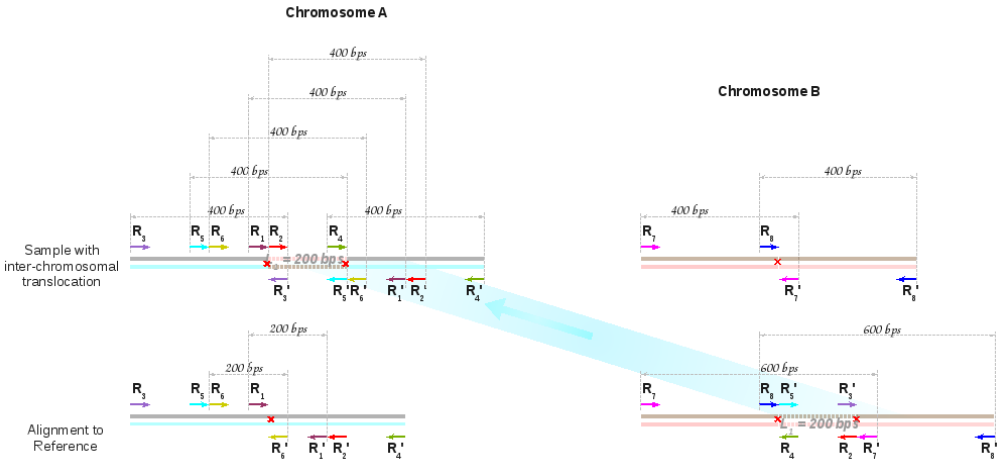

Figure 5.13: Read behavior under an inverted inter-chromosomal translocation

5.1.1.5 Inverted Translocations

As in the previous case, inverted translocations are easier to detect when the shift occurs onto another chromosome. The read behavior in this case, is very similar to a simple inter-chromosomal translocation. Here too, one chromosome experiences loss of DNA which appears as a deletion, while the other exhibits an insertion.The one difference in the current situation, is that read-pairs with a single mate lying in the translocated region get mapped either in a forward-forward or reverse-reverse direction.

In Fig. 5.13 a segment from Chromosome B gets detached, inverted and attached to Chromosome A. Reads R_1 and R_6 map to Chromosome A with a decreased inter-mate distance of 200 bps in place of the original library insert length of 400 bps, indicating an insertion on Chromosome A. On the other hand reads R_7 and R_8 map with an increased inter-mate distance of 600 bps in place of the original library insert length of 400 bps, suggesting a deletion on Chromosome B. Reads R_2 and R_4 get mapped to Chromosome B while their mates align to Chromosome A. Note that for these read-pairs, both mates map in the reverse direction. Similarly, reads R_3 and R_5 map to Chromosome A and their mates to Chromosome B, all in the forward direction.

If the translocated region is large, the insertion manifests in the form of several unmapped reads in the vicinity of the translocation instead of mates with decreased inter-mate distance.

Note that an intra-chromosomal translocation whether inverted or non-inverted can be viewed as a lateral shift of two regions, one to the right and the other to the left. In the absence of inversion, it is not possible to distinguish whether the left segment detached and re-attached or the right. However, in case of inverted translocations along the same chromosome, this distinction is now possible as the inverted portion is identified as the shifted segment.

Fig. 5.14 illustrates an inverted intra-chromosomal translocation, where a segment with length L_A has shifted left by a distance L_B and re-attached itself in an inverted direction.

Paired-end reads that lie entirely within or entirely outside the two translocated regions (namely, the shifted and inverted segment of length L_B, and the shift of length L_A), such as reads R_6 - $R_6^{'}$, and R_9 - $R_9^{'}$ conform to the expected read behavior. Read-pairs with one mate in the inverted region and the other outside, such as R_2 - $R_2^{'}$, R_3 - $R_3^{'}$, R_4 - $R_4^{'}$ and R_5 - $R_5^{'}$ could deviate both in orientation and inter-mate distance. Read-pairs with one mate in the non-inverted region of length L_B and one in a region that does not experience a shift, conform in orientation, but may exhibit a larger or smaller inter-mate distance than expected, commensurate with a deletion or

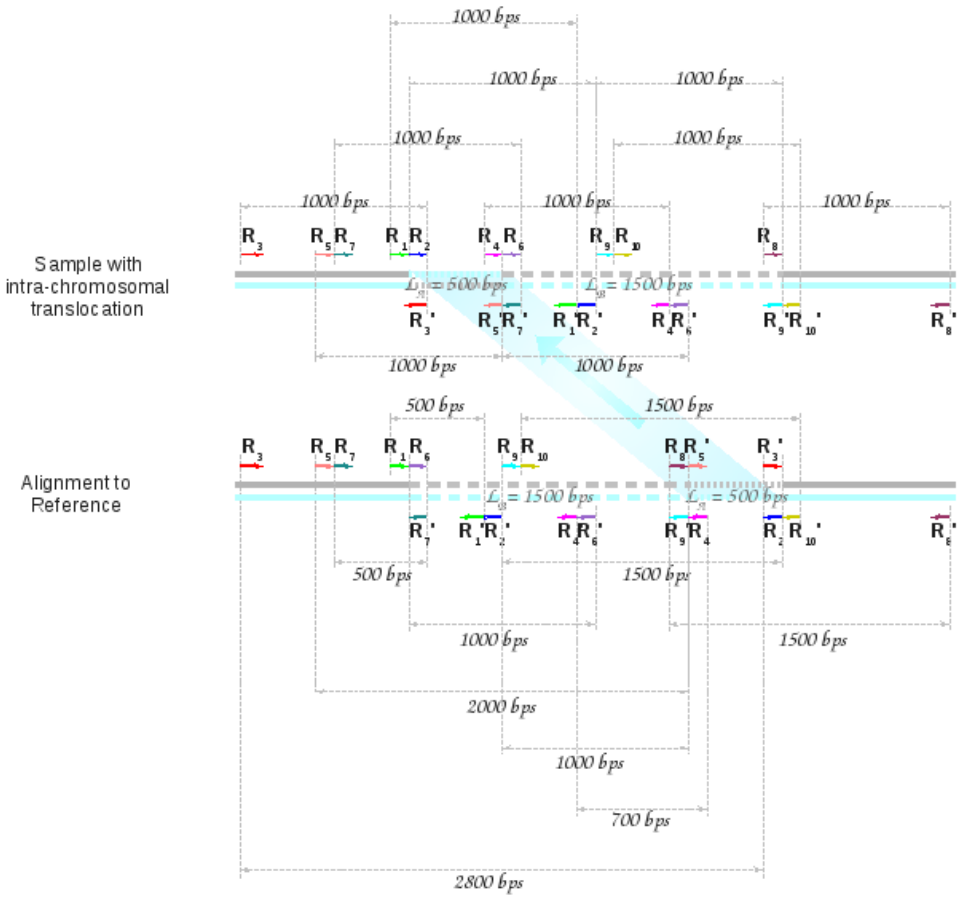

Figure 5.14: Read behavior under an inverted intra-chromosomal translocation: Small shift

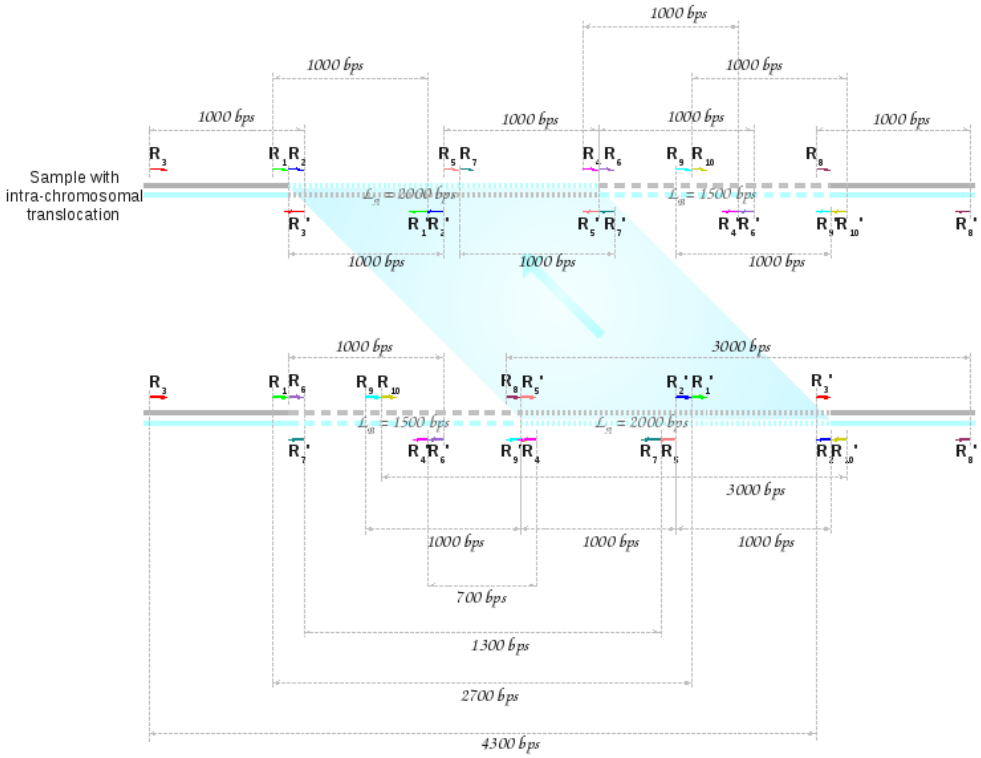

Figure 5.15: Read behavior in the presence of an intra-chromosomal translocation: Large shift

insertion respectively. In Fig. 5.14, reads R_8 - $R_8^{'}$, R_{10} - $R_{10}^{'}$, imply deletions, while R_1 - $R_1^{'}$ and R_7 - $R_7^{'}$ are suggestive of an insertion.

As with intra-chromosomal translocations, in this case too, reads denoting insertions are sometimes absent, though deletions are always observed. Fig. 5.15 illustrates this point. Here a deletion may be inferred at one end of the region experiencing shift, but there is clearly no read suggesting an insertion. As discussed previously, the read patterns observed, depend on the relative sizes of the inverted region L_A and its shift L_B in comparison to the original library insert length D, between mates. Table 5.1 lists the reads observed in each situation.

5.1.2 Key definitions and concepts

In this section we introduce some fundamental concepts used by the SV identification algorithm. We first define *paired read coverage* at a location and then introduce the notion of *ought-to-span reads*. The latter are read-pairs with only one mate mapping to a particular chromosome and are handled in a special way by **Avadis NGS**.

5.1.2.1 Paired Read Coverage

Paired read coverage at a point is defined as the number of read-pairs spanning the point. In other words, it is the number of reads starting at the point or to its left, whose mates end to the right of the point. (Here the term start denotes the left end of the read and the term end refers to its right end.)

Fig. 5.16 illustrates how the paired read coverage is computed at various points across a section of

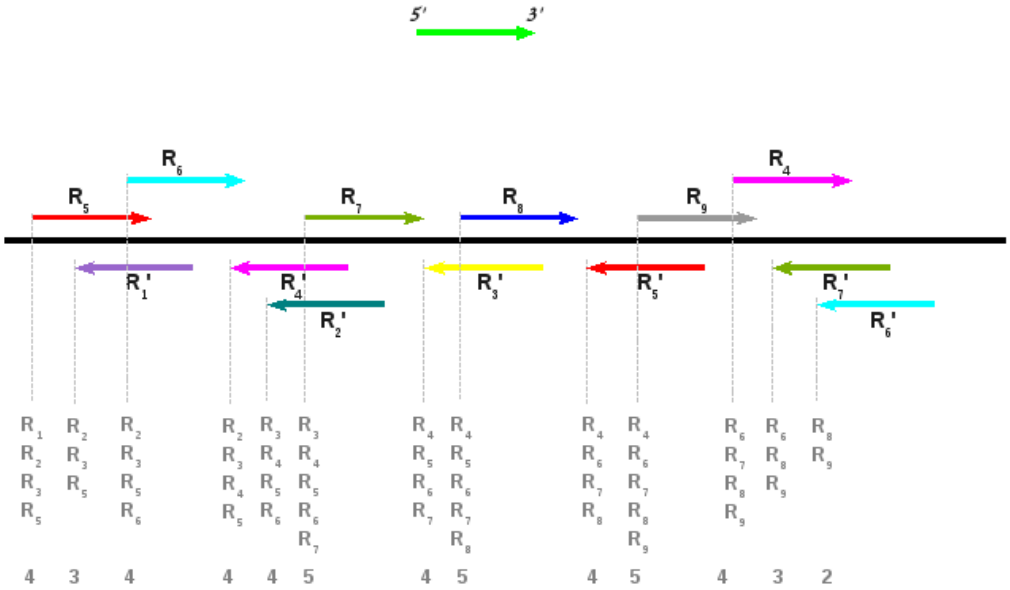

Figure 5.16: Paired Read Coverage

the reference. The reads included at each considered point are listed in grey.

While computing the paired read coverage at a point, **Avadis NGS** only considers forward and reverse reads lying within a specific window. Reads whose mates have been previously processed are discarded, and the paired read coverage is computed from the remaining reads. The longer the size of the window, the greater the storage requirements for the algorithm.

5.1.2.2 Ought-to-span reads

Read-pairs with only one mate mapping to a particular chromosome are termed ought-to-span reads. This typically happens under the following situation:

- The two mates map onto different chromosomes, possibly indicating an underlying translocation or inverted translocation.
- The mate is completely missing, possibly indicating a large insertion.

For such *ought-to-span* reads, the algorithm introduces an imaginary mate, a mean insert length away from the read. This enables the clustering algorithm (discussed later in Section 5.3.2) to make decisions on whether such reads should be grouped together as representative of the same underlying phenomenon.

5.2 Overview of Algorithms

Avadis NGS performs a search for SVs on a per-sample, per-chromosome basis. For each sample, the SV search is parallelized over chromosomes. The algorithm requires a minimum presence of total and deviant read pairs as necessary evidence to call an SV. Previous discussions suggest three different types of deviant read behavior corresponding to different kinds of SV as follows:

1. **Reads deviate in terms of their inter-mate distance, but not in orientation:** This is suggestive of insertions and deletions. A read is considered deviant if its inter-mate distance $L_{Inter-mate}$ differs from the expected inter-mate distance $\hat{\mu}$ by more than a user-determined minimum L. This number L takes default value $2\hat{\sigma}$, where $\hat{\sigma}$ is the estimated standard deviation of the inter-mate distance.

2. **Reads deviate in orientation:** Based on the type of deviation, the underlying SV could be an inversion or an intra-chromosomal translocation.

3. **Reads have mates that are missing or do not map to the same chromosome:** This implies long insertions or inter-chromosomal translocations.

End-to-end algorithm:

1. **Computation/Estimation of sample distribution parameters:** To mark a read as deviant, it is necessary to investigate whether the deviation of the observed inter-mate distance from the library mean is more than twice the standard deviation. The expected mean and standard deviation are part of the experimental process and can be presented as a field in the input SAM/BAM file. However, being optional fields, **Avadis NGS** offers an algorithm that approximates these by a computed mean, $\hat{\mu}$, and standard deviation, $\hat{\sigma}$ of the "normal" (non-deviant) reads. Details of estimation of these parameters are provided in Sections 5.3.1.1 and 5.3.1.2. The μ and σ attributes of the sample can be modified by the user from the sample inspector.

   In addition to these, **Avadis NGS** also pre-computes the average paired read coverage. This involves computing the paired read coverage (defined earlier in Section 5.1.2.1), at the read start position for each valid forward read and the read end position for each valid reverse read and then taking the average over all such positions. While computing the paired read coverage at a point, **Avadis NGS** only considers forward and reverse reads lying within a window of $2\hat{\mu} + 4\hat{\sigma}$ before the point.

   This quantity is useful in providing a threshold on the minimum presence of deviant reads C required at a location, for it to be considered a candidate for further investigation.

2. **Location-wise processing:** Reads mapping to a particular chromosome are ordered according to the locations of their start positions. For each start position in sequence along the chromosome, a window of length $2\hat{\mu} - 4\hat{\sigma}$ preceding the start position is considered. The algorithm proceeds with the following steps in the given window:

   - **Unmated reads**

     The number of spanning pairs with one mate missing, $N_{unmated}$ are counted. The corresponding fraction $f_{unmated}$, of such read pairs to the total spanning pairs, N_{total}, is computed.

   - **Paired reads with skewed orientation**

     The number of spanning pairs with skewed orientation, $N_{skewed\ orientation}$, are counted. The corresponding fraction $f_{skewed\ orientation}$, of such read pairs to the total spanning pairs, N_{total}, is computed.

   - **Paired read with expected orientation**

     The number of spanning pairs with expected orientation but deviant inter-mate distance, $N_{deviant\ IMD}$, are counted. The corresponding fraction $f_{deviant\ IMD}$, of deviant to total spanning pairs N_{total}, is computed.

   If any of the fractions $f_{unmated}$, $f_{skewed\ orientation}$, or $f_{deviant\ IMD}$ exceeds a user-specified threshold f (*minimum local deviant read fraction*), and the corresponding count ($N_{unmated}$, $N_{skewed\ orientation}$, or $N_{deviant\ IMD}$) exceeds C then

- Deviant spanning read pairs are grouped into clusters.
  - In case of unmated reads and read pairs with skewed orientation, the clustering algorithm used is the one presented in Section 5.3.2.3.
  - In the case of reads with normal orientation but deviant inter-mate distance, the clustering algorithm used could be either the PEMer or the k-BIC algorithm, both discussed in Section 5.3.2.2.
- The cluster with the largest number of spanning pairs is selected. If the fraction of cluster member pairs to total spanning pairs falls below deviant fraction threshold f, the cluster is disregarded. Otherwise, it is merged with a previously created SV via the merging algorithm in Section 5.3.2.3. If the cluster cannot be merged with any existing clusters, a new SV is created.

3. **Post-processing:**

- Clusters are suitably combined to deduce the underlying SV and its boundaries and size are determined. Details are presented in Section 5.3.3.
- The SV is determined to be homozygous or heterozygous based on the ratio of deviant reads in the dominant cluster to the total number of reads (with the exception of INDELs detected by the k-BIC algorithm). Details are available in Section 5.3.3.2.

5.3 Mathematical Details of the SV Detection Algorithm

5.3.1 Estimation of Inter-mate Distance Distribution Parameters

Assuming that the inter-mate distance follows a normal distribution, this section discusses how the associated mean $\hat{\mu}$ and standard deviation $\hat{\sigma}$ are computed in **Avadis NGS**.

5.3.1.1 Estimation of Mean Inter-mate Distance

The aim of this calculation is to divine the original inter-mate insert length introduced during library preparation. A mean taken across all possible mate-pairs would provide a decidedly distorted result, as it would include deviations relating to structurally variant reads, which exhibit both short and long extreme values and hence are outliers in the data set. This makes it imperative to separate the ‘‘normal” from the deviant reads before computing the mean. The algorithm uses a Trimmed Mean, obtained by discarding a pre-decided percentage (hard-coded to 25% in the **Avadis NGS** algorithm) of data in both the left and the right tails of the distribution of the inter-mate distance for all read pairs. The details of the algorithm are as follows:

1. Bin inter-mate distance for all mate-pairs. Each bin is of width one. A maximum is set for the number of bins (hard-coded to 50 million in **Avadis NGS**) (same as insert length). A special bin is formed for distances exceeding 50 million (in the expectation that there would be very few elements in this bin).

2. Beginning from the left of the distribution, the number of reads in each bin are summed until they reach 25% of the total number of read-pairs; they are then removed. A similar approach is applied from the right-side of the distribution.

3. The mean $\hat{\mu}$ of the remaining observations is calculated.

No.	Cluster Types	Reads included		
		Mate alignment on the same chromosome	Orientation (F- Forward/ (R - Reverse)	Inter-mate distance
1.	Deletion	Yes	F-R	Larger than expected
2.	Insertion	Yes	F-R	Smaller than expected
3.	Large Insertion	Unmapped mates	-	-
4.	Inversion	Yes	R-R AND F-F	Smaller AND larger than expected
5.	Inter-chromosomal translocation	Different chromosome	-	-
6.	Intra-chromosomal translocation	Yes	R-F	Smaller OR larger than expected

Table 5.2: Cluster Types

5.3.1.2 Estimation of the Standard Deviation of the Inter-mate Distances

The approach used in Section 5.3.1.1 cannot be used for computing standard deviation: While this would certainly remove most deviant reads, it might also discard normal reads with large standard deviation. Thus the standard deviation estimate from the remainder of the population might be smaller than originally introduced. In view of this, we use a more robust measure of dispersion called the **Median Absolute Deviation** or **MAD**, on which outliers have very little effect. The standard deviation is then approximated from the calculated MAD.

The details of the algorithm are as follows:

1. Calculate the median of inter-mate distances over all read pairs.
2. For each read pair, compute the absolute deviation of the inter-mate distance from the previously calculated median.
3. Now compute the median of the absolute deviations. This is the MAD.
4. As the inter-mate distance follows a normal distribution, the standard deviation $\hat{\sigma} = 1.4826 \times MAD$.

5.3.2 Clustering Algorithm Used for SV Identification

Deviant reads in a region collectively indicate presence or absence of an underlying SV. **Avadis NGS** uses a clustering algorithm to group reads in a region that exhibit similarity in deviant behavior.

A cluster may either contain only read-pairs or only unmated reads. If a read belongs to a cluster, its mate (if present on the same chromosome) must also belong to the same cluster.

5.3.2.1 Cluster Properties

- **SV Tag**

  Each cluster has an SV tag declaring it as an insertion, large insertion, deletion, inversion, inter-chromosomal translocation or intra-chromosomal translocation cluster. Table 5.2 describes the characteristics of member reads for each cluster type. Note that the inter-mate distance for reads under inversion have a wide spread. Under intra-chromosomal translocation,

Cluster Types	Predictive SV region	
	Start position	End position
Deletion/Insertion	Right-most position of right-most F read	Left-most position of left-most R read
Large Insertion	Right-most position of right-most F read (excluding imaginary reads)	Left-most position of left-most R read (excluding imaginary reads)
Inversion	Left-most position of left-most R read	Left-most position of right-most F read
Intra-chromosomal translocation	Right-most position of right-most R read	Left-most position of left-most F read
Inter-chromosomal translocation	Right-most position of right-most F read (excluding imaginary reads)	Left-most position of left-most R read (excluding imaginary reads)

Table 5.3: Predictive SV region for each cluster type

the inter-mate distance has standard deviation commensurate with that present in the data, namely $\hat{\sigma}$.

Figure 5.17 illustrates each of the six types.

- **Cluster region**

  For a cluster of paired-reads, the cluster region extends from the left-most position of the left-most read in the cluster to the left-most position of the right-most read in the cluster.

  In case of unmated or ought-to-span reads, their imaginary mates created in the preprocessing step (discussed earlier in Section 5.1.2.2) are taken into account while defining the cluster region.

  Figure 5.17 provides a clear picture of the cluster region for each cluster type.

- **Estimated inter-mate distance of the cluster**

  The estimated inter-mate distance of a cluster is defined by the average of inter-mate distances for all reads within the cluster. A cluster of ought-to-span reads has inter-mate distance set to the mean expected inter-mate distance $\hat{\mu}$ by design.

- **Predictive SV region**

  The predictive SV region for a cluster, or in other words, the region where the SV is expected to lie based on the reads present in the cluster, is defined differently for each cluster. Table 5.3 provides the definitions, while Figure 5.17 gives a pictorial description.

Note that in the case of inter-chromosomal translocation, two chromosomes are involved: Chromosome A, experiencing a deletion, and Chromosome B, experiencing an insertion. The cluster is formed only on Chromosome A and ought-to-span reads are also included. Mates of reads in this cluster all lie on Chromosome B and are automatically grouped together. However, they do not form a cluster in the sense used here, and so are not associated with any of the cluster properties. This is depicted in Figure 5.17 by denoting the group by an ellipse instead of a rectangle.

5.3.2.2 Cluster Algorithms Used for Creating INDEL Clusters

INDELs are determined based on deviations in their inter-mate distance from a norm. **Avadis NGS** provides two algorithms to detect these SVs: One is the **P**aired-**E**nd-**M**ap**p**er (PEMer)

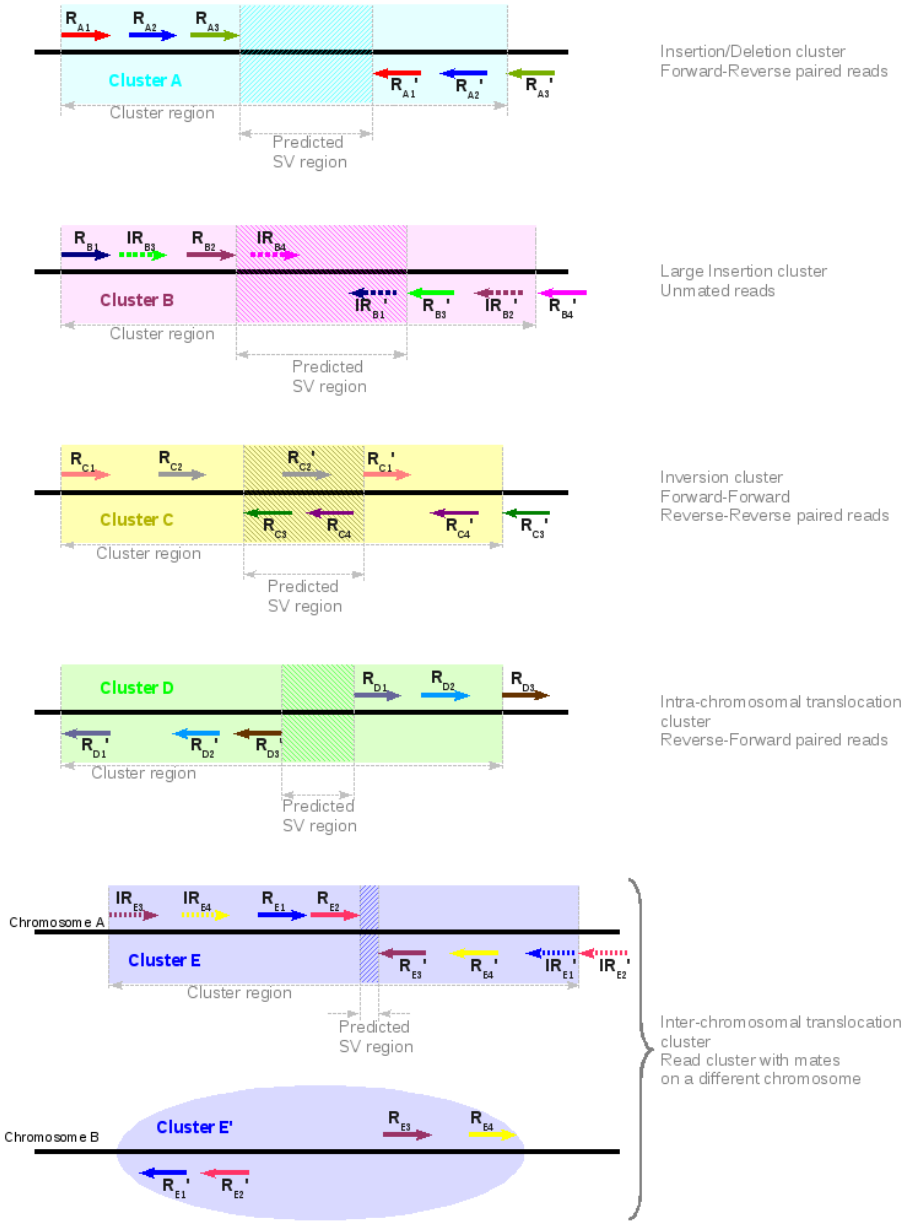

Figure 5.17: Cluster region

algorithm [21], which compares the inter-mate distance with a pre-determined cutoff and draws conclusions as to the presence/absence of an insertion or deletion. The other, the k-BIC method, uses cluster analysis of insert lengths to classify the reads into an appropriate number of groups, each group being called a particular type of SV.

The PEMer Algorithm

1. Let the mean library insert length be μ. Let the standard deviation of original library insert lengths be σ. If these quantities are given by the user, then they are used for subsequent calculations. If these are not given, then these quantities are estimated from the data. In either case, these quantities are respectively denoted by $\hat{\mu}$ and $\hat{\sigma}$, as explained in Sections 5.3.1.1 and 5.3.1.2 respectively. For subsequent analysis these estimates are used.
2. Find all the paired reads that overlap a particular point in the reference.
3. These points are the start of all the reads.
4. Carry out the following steps for each of the points:
   (a) Select read-pairs having normal orientation. Discard those which do not have normal orientation.
   (b) Find the deviant read pairs based on Table 5.5, given in Section 5.3.5, which gives cut-off values C. This table is adapted from Korbel et al. (2009) [21]. Only those reads with insert lengths greater than mean + $C\times$ the standard deviation, are used for subsequent analyses.
   (c) Run the clustering algorithm explained earlier in Section 5.3.2 on all the reads obtained in (2) above.
   (d) Remove clusters which contain less than a user-specified number d of deviant read-pairs.
   (e) Tag each of the remaining clusters as an insertion cluster or a deletion cluster based on the average inter-mate distance of the cluster---a short one if it is less than $\hat{\mu} - 2\hat{\sigma}$ and a long one if it is greater than $\hat{\mu} + 2\hat{\sigma}$. Discard those that are neither short nor long.
   (f) Consider the possibility of merging clusters by applying the method described in Section 5.3.2.
   (g) Consider the cluster with the largest number of reads as the SV cluster.
   (h) Detect the SV region as explained later in Section 5.3.2.

The k-BIC Algorithm

This is an algorithm to cluster the reads into an optimal number of clusters by their insert lengths. The algorithm is a combination of the k-means algorithm, which clusters reads into a given number k of groups, followed by the application of a criterion, called the Bayesian Information Criterion (BIC) to choose the most suitable value of k in a given range. The k-means algorithm is an iterative algorithm which groups the given observations into k groups in such a way that the sum of squares of distances between objects within the groups is minimized in an effort to make the groups homogenoeus. If k is not fixed then making a large number (say, as many as n the number of observations) of groups will make the total within group sum of squares small; in the limit if $n = k$, then this can be made 0. However, such a solution will over-fit the data and make its predictive ability poor. Hence a compromise between the minimization criterion of sum of squares and the number of groups is necessary and this is achieved by introducing a penalty term added to the criterion which will discourage large k. One such combination of the k-means criterion and a penalty term is the BIC. Thus in our approach we find clusters for various values of $k = 2, 3, 4, 5$ and choose clusters based on the value of k, which is best (least) according to BIC. Details of this method are given in Section 5.3.4.

Detection of Smaller Indels Using k-BIC Algorithm

1. Determine the optimal number of clusters of read pairs using the k-BIC algorithm.
2. If this does not exceed P, the user-specified ploidy, retain the clusters, otherwise discard them.
3. Determine the deviant clusters with the help of the confidence interval, explained below:
   (a) Compute $(\hat{\mu}_i - \frac{1.96\hat{\sigma}}{\sqrt{n_i}}, \hat{\mu}_i + \frac{1.96\hat{\sigma}}{\sqrt{n_i}})$, where $\hat{\mu_i}$ is the mean of insert lengths of reads in this cluster; $\hat{\sigma}$ is standard deviation of all the reads computed as explained in 5.3.1.2.
   (b) If this interval contains $\hat{\mu}$, the given mean of insert lengths or mean of all the reads computed as explained in 5.3.1.1, then we conclude that the cluster thus formed is not a deviant cluster and hence is not an INDEL. Else it is considered to be a valid INDEL.
4. Remove any cluster in which the number of deviant reads is less than the user-specified cutoff, d; this step is the same as in PEMer.
5. Pick the best such cluster (this is defined as the cluster containing the largest number of deviant reads). This cluster is considered to be the SV cluster; this step is the same as in PEMer.
6. If the number of clusters formed is more than one (it does not matter whether they are deviant or non-deviant), mark the SV as heterozygous, otherwise homozygous.
7. Detect the SV region as explained later in Section 5.3.2. This step is same as in PEMer.
8. Consider the possibility of merging clusters by applying the method described in Section 5.3.2; this step is the same as in PEMer.

Compute $\Delta_i, i = 1, 2, \ldots, k$, the INDEL lengths associated with clusters i as $\Delta_i = \hat{\mu}_i - \hat{\mu}$.

5.3.2.3 Creating and Merging Clusters

The previous section addressed algorithms used for creating insertion and deletion clusters. This section addresses the algorithms used to merge previously created clusters of each kind, including insertion/deletion clusters.

The same algorithm is also used to create and merge the remaining types of clusters given in Table 5.2. The approach taken is to consider each new read pair as a independent, small cluster of two reads. The new cluster is included into an existing cluster if it satisfies the conditions of the merging algorithm.

The merging algorithm incorporates a set of criteria listed below:

1. **SV tag criterion**

   The two clusters must have the same SV tag.

2. **Similar estimated inter-mate distance criterion**

   The two clusters A and B must have *similar* estimated inter-mate distance (EID), where similarity in this case requires any ONE of the following conditions to be satisfied:

   (a) $0.9 \leq \frac{EID(A)}{EID(B)} \leq 1.0$ with A being the cluster with the smaller EID)

   (b) In case of inversions: $|EID(A) - EID(B)| \leq 2\hat{\mu} + 4\hat{\sigma}$.

   (c) In all other cases: $|EID(A) - EID(B)| \leq 4\hat{\sigma}$.

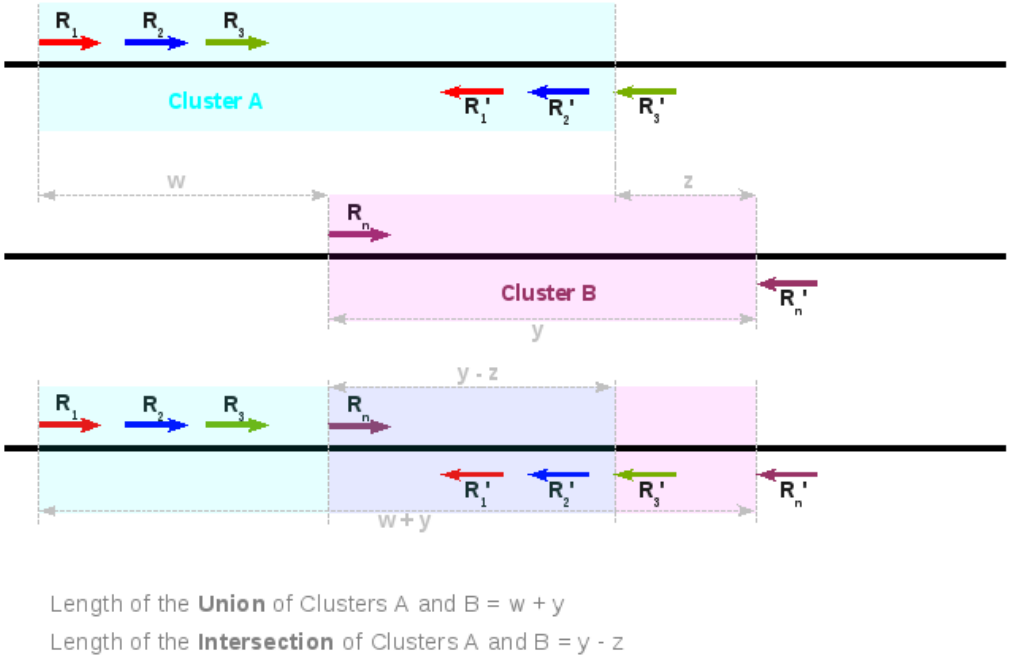

Figure 5.18: Merging Clusters

where $\hat{\mu}$ is the mean and $\hat{\sigma}$, the standard deviation of the inter-mate distance, by design during library preparation or as estimated by Sections 5.3.1.1 and 5.3.1.2

For deletion clusters, there is an additional requirement that the span of the predicted SV region should exceed 0.7 times the EID.

3. **Distance between clusters criterion**

   This ensures that reads that are too far apart are not included in the same cluster. It checks the distance between the start positions of the two clusters and also the end positions of the two clusters, shown by w and z respectively in Figure 5.18. Both distances must lie within a certain threshold; this threshold is $2\hat{\mu} + 4\hat{\sigma}$ for inversion clusters, $\hat{\mu} + 2\hat{\sigma}$ for all other clusters.

4. **Cluster region overlap criterion**

   The fractional overlap (defined by the ratio of the length of the intersection of the two clusters to that of their union) must exceed a lower threshold of 0.5. Figure 5.18 shows an example where a new read pair R_n - $R_n^{'}$ is examined for inclusion in an existing cluster A. The new read pair defines a cluster B. The length of the union of the two clusters as shown in the figure is $w + y$, and that of the intersection, $y - z$. The cluster is a valid candidate for inclusion if it meets the condition $\frac{y-z}{w+y} > 0.5$.

5. **Predictive SV region overlap criterion**

   The formulation also considers the predictive SV region, i.e. the region where the SV is expected to lie. The previous example shown in Figure 5.18, is further elaborated in Figure 5.19. Two clusters may only be merged if the ratio of the length of the intersection of their predicted SV regions to that of their union exceeds 0.4. In Figure 5.18, the length of the intersection of the predictive SV regions of clusters A and B is v, and that of their union is $x + y - u$. The condition $\frac{v}{x+y-u} > 0.4$ must be met for cluster B to be merged with A.

Note that all criteria need not be satisfied for merging clusters. Table 5.4 displays conditions under which clusters of various types may be merged.

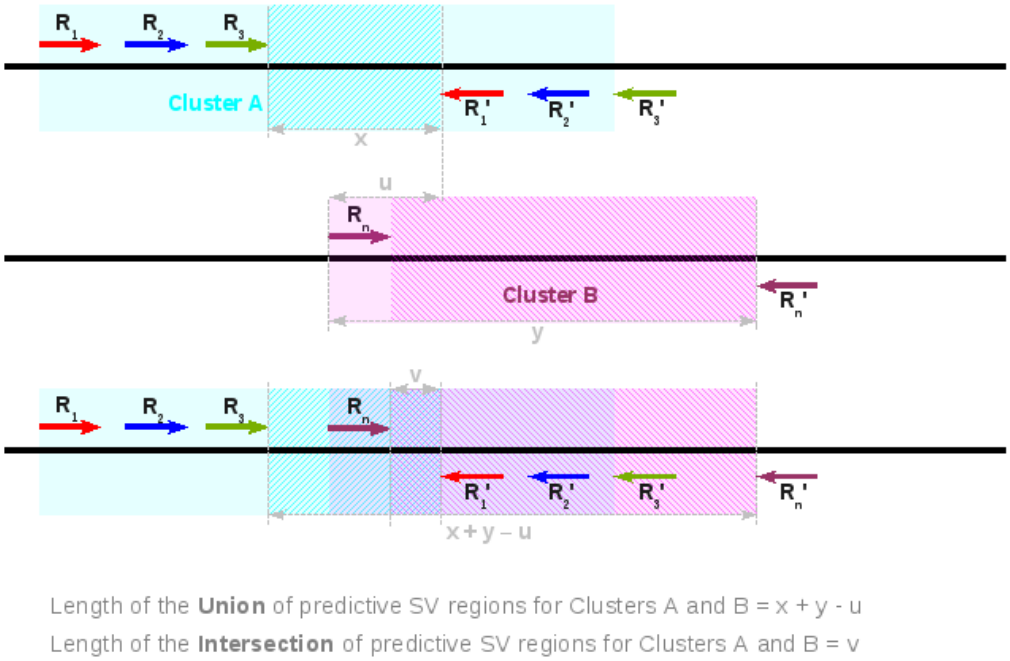

Figure 5.19: Predictive SV regions in merging clusters

Cluster Types	Merge criteria				
	SV tag	Similar estimated inter-mate distance	Distance between clusters	Cluster region overlap	Predictive SV region overlap
Deletion	✔	✔	✔	✔	✔
Insertion	✔	✔	✔	✔	✘
Large Insertion	✔	✘	✔	✔	✘
Inversion	✔	✔	✔	✔	✘
Inter-chromosomal translocation	✔	✘	✔	✔	✘
Intra-chromosomal translocation	✔	✔	✔	✔	✔

Table 5.4: Cluster Merge Criteria

As noted before, for an inter-chromosomal translocation, a cluster is formed on one chromosome only, specifically the one experiencing a deletion. The criteria for merging are checked on this cluster only. If a read is merged into the cluster on this chromosome, its mate automatically gets grouped with the mates of other cluster members on the insertion chromosome. The algorithm does however check that all reads included in the cluster on the deletion chromosome have mates on the same chromosome.

5.3.3 SV Post-processing Issues

As seen earlier in Section 5.1.1, some SVs manifest in the form of different read clusters. For example, intra-chromosomal translocations are typically represented by a single translocation cluster flanked by two deletion clusters. Sometimes, insertion clusters are also evident. Post-processing challenges include

- Converging the relevant clusters into a single SV call
- Identifying the boundaries and the size of the SV region. This is typically done using the predicted SV regions defined in Table 5.3, the average or expected inter-mate distance, $\hat{\mu}$, described in Section 5.3.1.1 and the estimated inter-mate distance of the SV cluster, EID, as defined in Section 5.3.2.3.
- Classifying a call as heterozygous or homozygous.

5.3.3.1 SV Boundary Determination

Deletions

An SV deletion manifests as a single deletion cluster. The boundaries of the deletion are given by the boundaries of the corresponding predicted SV region. The size of the SV, SV_{sz}, is given by

$$SV_{sz} = min\left(PR_{span}, EID - \hat{\mu}\right),$$

where PR_{span} is the span of the predictive SV region.

Insertions

In case of insertions, the clusters evident depend on the size of the inserted region.

If the inserted region is larger than the library inter-mate distance, only a large insertion cluster is visible. For large insertions, the point of insertion can be estimated through the predicted SV region in the corresponding cluster. The size of the insertion, however, cannot be estimated.

For smaller insertions, both a large insertion cluster and an insertion cluster may be present. The large insertion cluster should be discarded. The region of insertion is denoted by the predicted SV region for the insertion cluster; its size SV_{sz}, is given by $SV_{sz} = \hat{\mu} - EID$.

Inversions

Inversions result in single inversion clusters; the corresponding predicted SV region define the SV boundaries, and the size of the SV is the span of this region, i.e., $SV_{sz} = PR_{span}$.

Inter-chromosomal Translocation

In this case, inter-chromosomal translocation clusters are seen on both chromosomes along with a deletion cluster on one chromosome. If the translocated region is small, an insertion cluster is also present on the other chromosome.

The boundaries of the translocated region on the deletion chromosome can be found proceeding exactly as for a deletion, described in Section 5.3.3.1. On the insertion chromosome, the point of

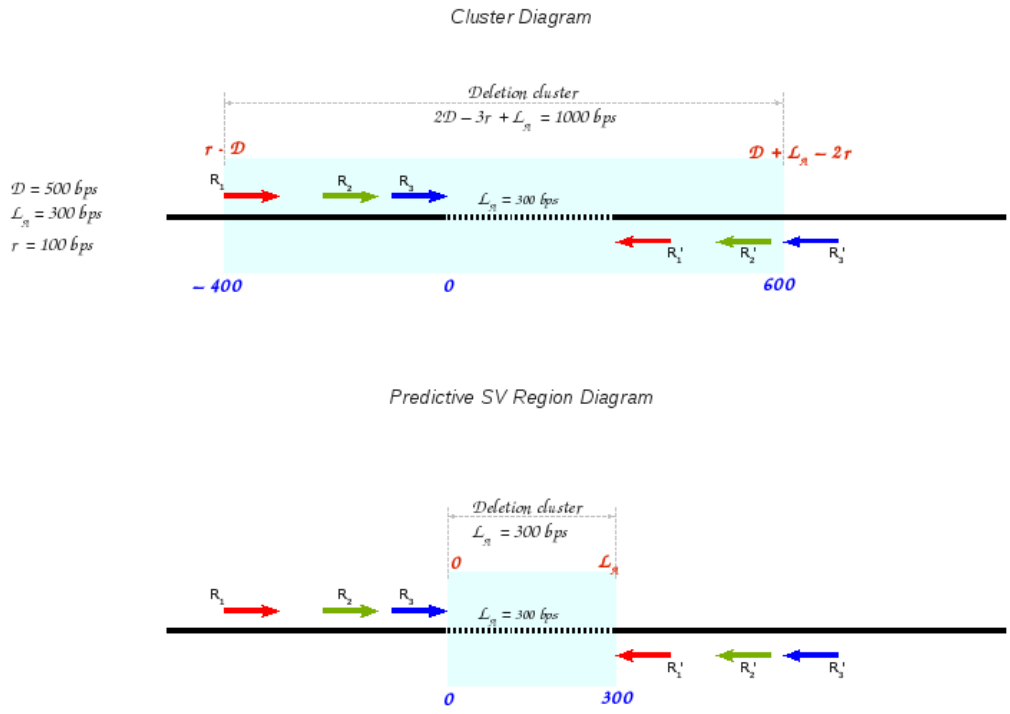

Figure 5.20: Boundary determination for deletions

insertion is found from the predictive region of the translocation cluster. It may also be deduced from the predictive region of the insertion cluster when the latter is present, i.e. in case of small translocations.

The size of the SV region can be derived from the deletion cluster and if present also from the insertion cluster following the approach outlined in Sections 5.3.3.1 and 5.3.3.1.

Inter-chromosomal Inverted Translocation

Inter-chromosomal translocations with and without inversion are very similar in physiognomy. One chromosome experiences a deletion, while another an insertion. A deletion cluster is therefore visible in both inverted and non-inverted translocations. An insertion cluster is also present in case of small translocations. Inter-chromosomal translocation clusters appear on both chromosomes in both cases.

The one significant difference between an inverted versus a non-inverted translocation is in the orientation of the reads in the translocation cluster. In the former case, the directions of the two mates (on the two different chromosomes) would be forward-forward or reverse-reverse, while the latter will show the regular forward-reverse direction.

Thus the detection and post-processing for both an inverted and a non-inverted translocation is identical up to the very end, when the direction of reads in the translocation cluster determines the difference in calls. Determination of size and boundaries in both cases is identical.

Intra-chromosomal Translocation

Intra-chromosomal translocations are the most complex of the SVs to identify. They manifest in the form of a single intra-chromosomal translocation cluster typically flanked by two deletion clusters. If the size of the translocated region, or its shift is small, insertion clusters could also appear beside the translocation cluster.

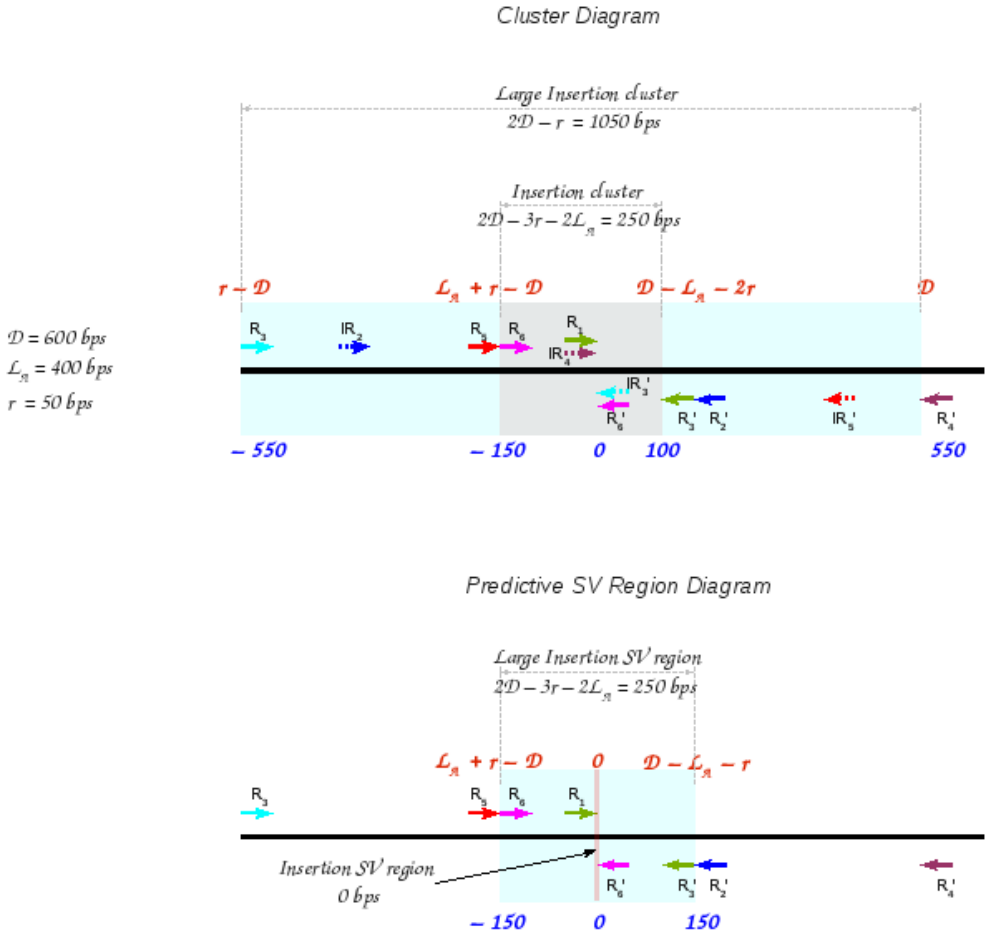

Figure 5.21: Boundary determination for small insertions

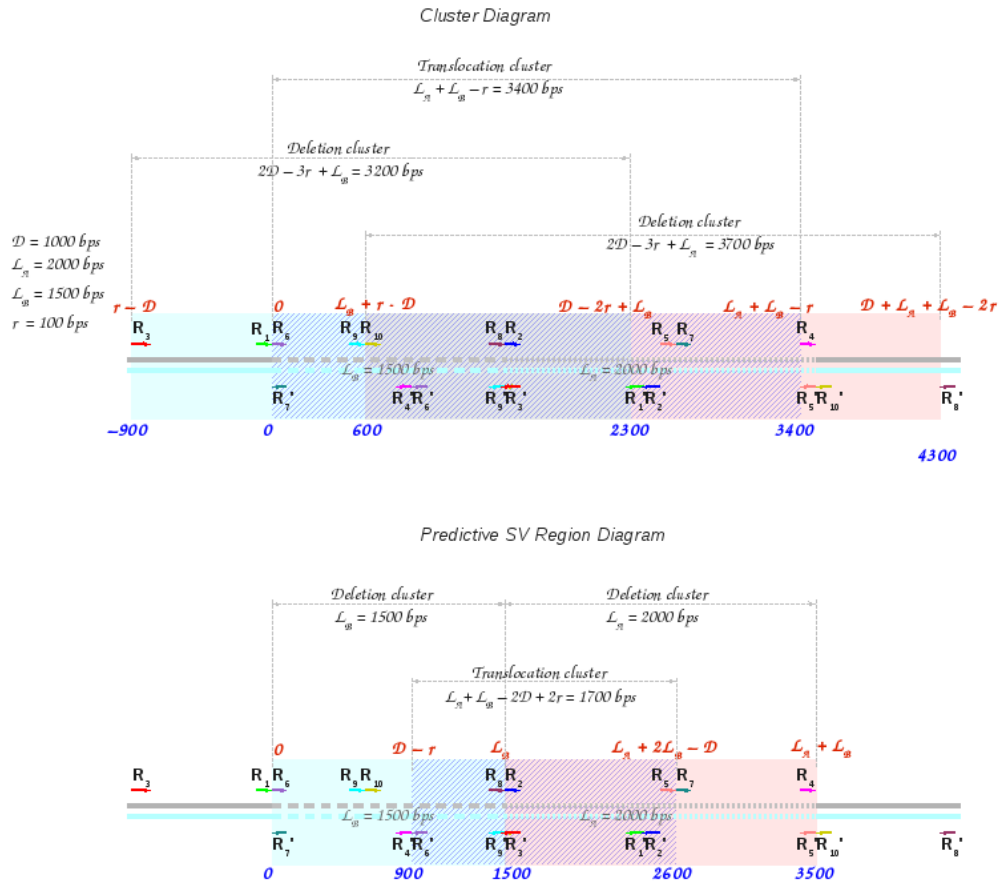

Figure 5.22: Boundary determination for intra-chromosomal translocation

Avadis NGS must identify three points to entirely describe an intra-chromosomal translocation, namely, the boundaries of the region that got detached and the point at which it got inserted. Let us denote the insertion point by X, the boundary point further away from X by Z, and the boundary point near X by Y.

In the absence of insertion clusters: This case is illustrated in Figure 5.22. X and Z can be estimated from the end points of the translocation cluster, as well as from one end point each, of the predictive SV regions of the deletion clusters. The other end point of the predictive SV regions of the two deletion clusters indicates the position of Y. An additional check for Y is that it must lie within the predictive SV region of the translocation cluster.

In the presence of 1 insertion cluster, at end Z: This case is illustrated in Figure 5.23. X can be estimated from one end point of the translocation cluster and one end point of one deletion cluster (at the end where the insertion cluster is absent), as well as from one end of the predictive SV region of the other deletion cluster. Z can be estimated from the predictive region of the insertion cluster. Additionally, it must lie between one end of the translocation cluster and one end of the predictive SV region of the one deletion cluster (at the end where the insertion cluster is present).

One end point each of the predictive SV regions of the two deletion clusters indicates the position of Y. An additional check for Y is that it lies at one end point of the predictive SV region of the translocation cluster.

In the presence of 2 insertion clusters: This case is illustrated in Figure 5.24. X and Z can

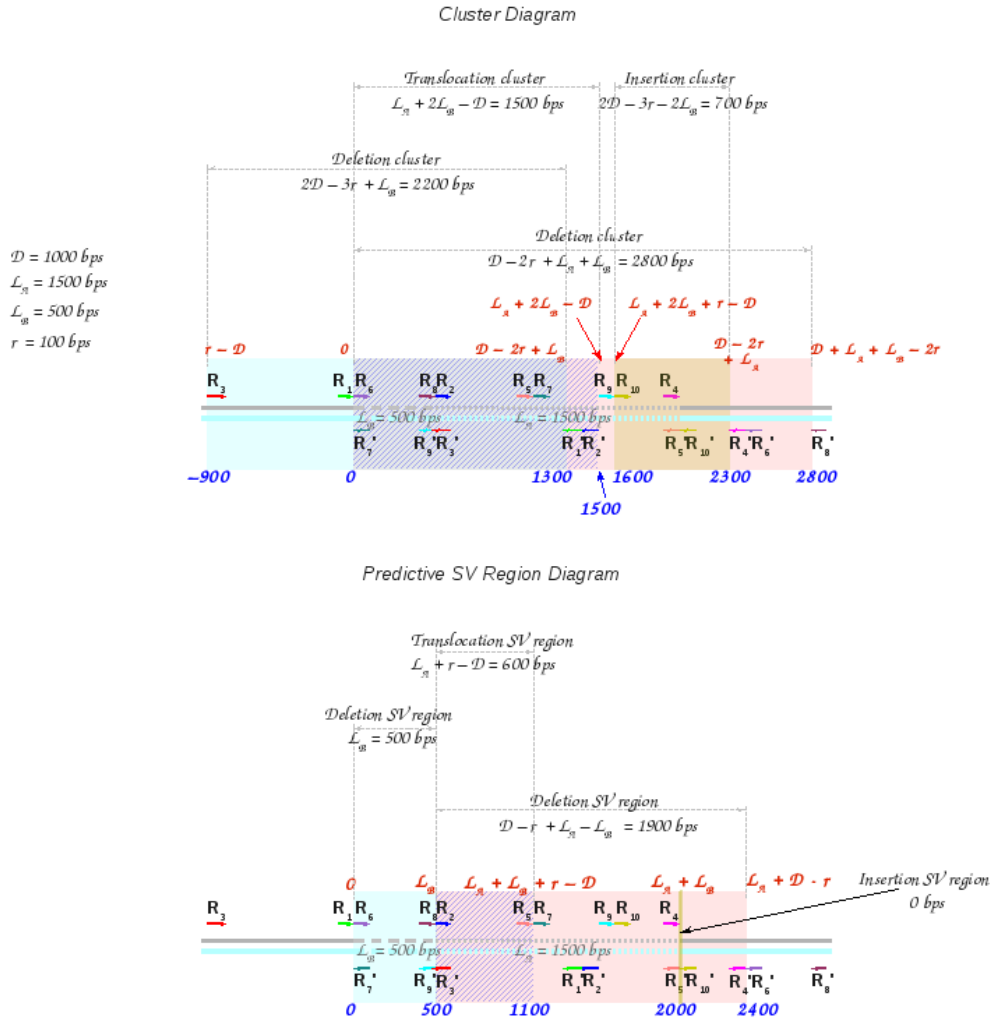

Figure 5.23: Boundary determination for intra-chromosomal translocation: Small shift

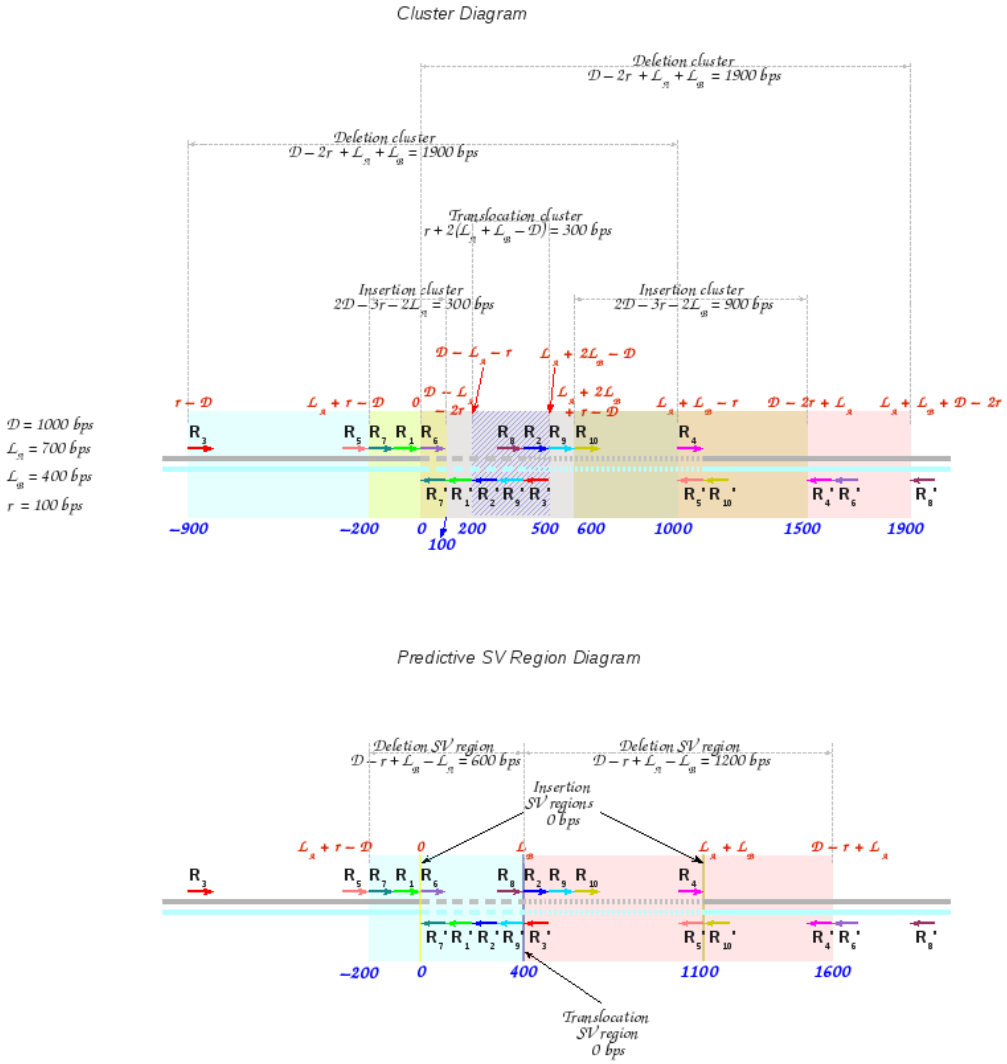

Figure 5.24: Boundary determination for intra-chromosomal translocation: Small region, small shift

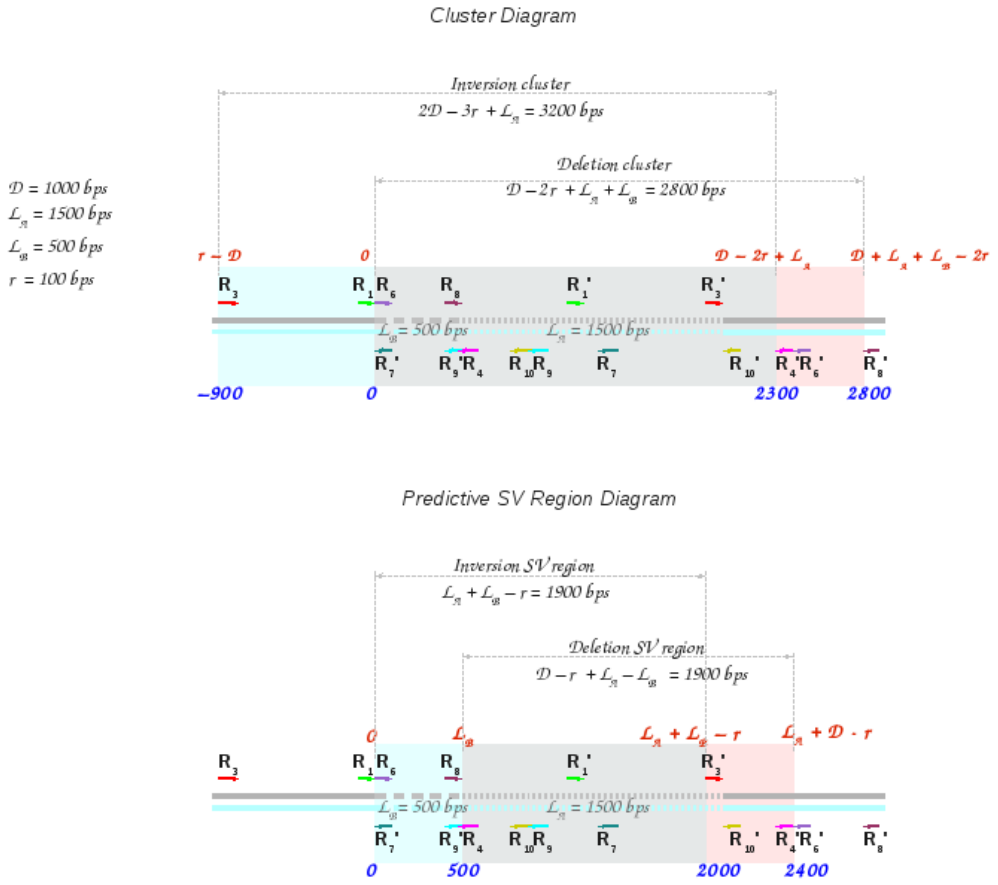

Figure 5.25: Boundary determination for intra-chromosomal translocation

be estimated from one end point each of the deletion cluster, as well as from the predictive SV regions of the two insertion clusters. One end point each of the predictive SV regions of the two deletion clusters indicates the position of Y. An additional check for Y is that it must lie within the predictive SV region of the translocation cluster.

In all cases, the $EIDs$ of the various clusters also provide good estimates as to the translocated regions.

Intra-chromosomal Inverted Translocation

Inverted intra-chromosomal translocations manifest in the form of a single inversion cluster typically accompanied by a deletion cluster. If the size of the translocated region is small, a single insertion cluster would also be present.

As for intra-chromosomal translocation, three points describe an inverted intra-chromosomal translocation, namely, Y and Z, the boundaries of the region that got detached and inverted, and the point X at which it got inserted.

In the absence of insertion clusters: This case is illustrated in Figure 5.25. The boundary Y of the translocated region can be estimated from one end point of the predictive SV region of the deletion cluster. The end points of the predictive SV region of the inversion cluster give the points X and Z.

The EID of the deletion cluster also gives an estimate of the size of the inverted translocated region.

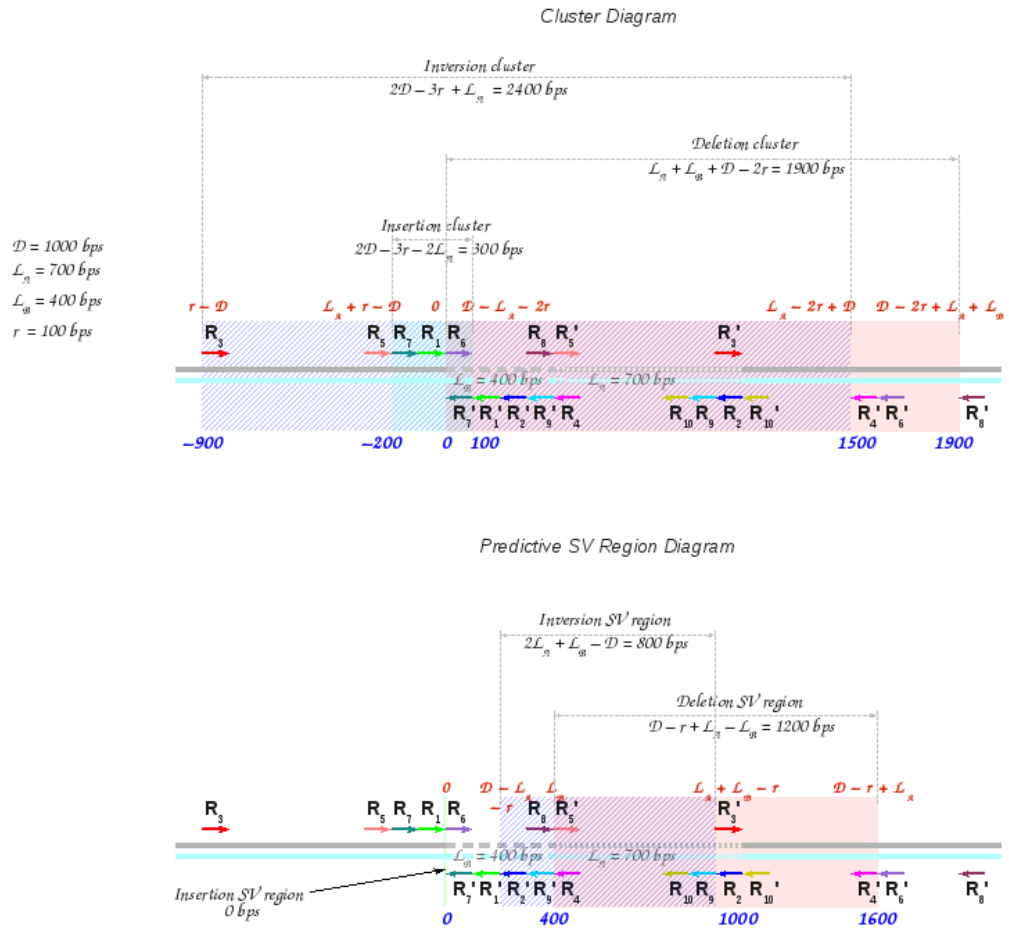

Figure 5.26: Boundary determination for intra-chromosomal translocation: Small shift

In the presence of 1 insertion cluster, at end Z: This case is illustrated in Figure 5.26. X can be estimated from the predictive SV region of the insertion cluster. Z can be estimated from one end of the predictive region of the inversion cluster. One end point of the predictive SV regions of the deletion cluster indicates the position of Y.

5.3.3.2 Zygosity Call for SVs

Avadis NGS decides between heterozygosity and homozygosity based on the relative proportion of deviant reads, specifically, the ratio of deviant reads supporting the corresponding SV to that of all the other reads present in the region. One thing to be noted here is that the reads participating in the computation of this ratio are only the first mates of the paired reads. The second mates do not feature in the ratio. If this ratio exceeds a threshold r_{Th}, a homozygous call is made; else the variant is deemed heterozygous. This threshold r_{Th} is set to 0.65, and is the same for all types of SVs. But the region which is taken into consideration, and the types of reads which participate in the zygosity call change across different types of SVs. These differences are explained in the following subsections.

This approach holds throughout with the one exception being detection of indels via the k-BIC algorithm.

Deletion

- Deviant reads supporting the SV: 'Mate far' reads supporting the deletion
- Reads supporting the heterozygous nature of the SV: All reads in the region other than the deviant reads supporting the SV. Even 'mate fars' not supporting the SV are considered here, as there might be two heterozygous deletions in the same region.
- Region from which the reads are taken: (start of the deletion region - mean insert length of the sample, start of the deletion region)

  The reason for conisdering this region is that it is within this region that if the SV was homozygous all the reads would have gone across to the other side of the deletion. Thus in case of an ideal homozygous condition, the ratio would have been 1. There will also be reads present outside this region which participate in the SV, but their number would be far lesser and since we are taking the ratio it is unlikely to affect the final zygosity call.

Large Insertion

- Deviant reads supporting SV: 'Mate missing' reads supporting the large insertion
- Reads supporting the heterozygous nature of the SV: All reads in the region other than the deviant reads supporting the SV, except 'mate nears' that might also be present supporting the same SV. This might happen when the insert length is close to the actual insertion sequence size in the sample.
- Region from which the reads are taken: (start of the large insertion region - mean insert length of the sample, start of the large insertion region)

  The reasoning behind considering this region is same as that in the case of deletion.

Insertion

- Deviant reads supporting SV: 'Mate near' reads supporting the insertion
- Reads supporting the heterozygous nature of the SV: All reads in the region other than the deviant reads supporting the SV, except 'mate missings', as they might be supporting the same SV. This is because in any kind of insertion there will always be some 'mate missing' reads present which support the that insertion. This phenomenon happens as the mates of those reads fall within the insertion region in the sample, and when they are mapped back to the reference, these mates are not able to align to the reference thus resulting in mate missing reads.
- Region from which the reads are taken: Bigger of the following two regions, R1: (start of the first read supporting the insertion + 2 times standard deviation, start of the insertion region), R2: (start of the insertion region - average read length, start of the insertion region).

  In this case all the supporting reads lie in the region from the start of the first read supporting the insertion to the start of the insertion region. Now very few deviant reads will lie between the 'start of this region' and 'two times standard deviation of the insert length from the start', because of the gaussian nature of the insert length. Thus the reads considered for the ratio are taken from the end of the this region to the start of the insertion region. This is R1 mentioned above. In cases where the insertion size is very close to the insert length of the reads, and the starndard deviation is more than the read length, the size of the region R1 might become very small and in some cases might even become negative. To circumvent this problem, the second region R2 also is considered whose size will never be close to zero.

Inversion

- Deviant reads supporting SV: 'One mate flip' reads supporting the inversion

- Reads supporting the heterozygous nature of the SV: All reads in the region other than the deviant reads supporting the SV, except 'mate far' reads indicating deletions because in case of inverted intra-chromosomal translocation mate far reads are produced as a side-effect of it. At the time of the detection of inversion the algorithm is not able to differentiate between inverted intra-chromosomal translocation and normal inversion and hence the mate far reads are not considered for the zygosity calculation.
- Region from which the reads are taken: Bigger of the following two regions, R1: (start of the first forward read + two times standard deviation, start of the last forward read), R2: (start of the last forward read - average read length, start of the last forward read).

  The reasoning behind considering this region is the same as that of the case of insertion. One thing to be noted though is that here the second region comes into effect when the size of the inversion is smaller than two times the standard deviation of the insert length of the sample.

Intra-chromosomal translocation

- Deviant reads supporting the SV: 'Both mate flip' reads supporting the intra-chromosomal tranlocation
- Reads supporting the heterozygous nature of the SV: All reads in the region other than the deviant reads supporting the SV, except reads indicating INDELs. This is because translocations always produce 'mate far' and 'mate near' reads as a side-effect, and considering them in the calculation would not be appropriate.
- Region from which the reads are taken: Bigger of the following two regions, R1: (start of the first read indicating translocation, start of the first read indicating translocation + average read length), R2: (start of the first read indicating translocation, start of the last read indicating translocation - two times the standard deviation).

Inter-chromosomal translocation

- Deviant reads supporting the SV: 'Translocated' reads supporting the translocation
- Reads supporting the heterozygous nature of the SV: All reads in the region other than the deviant reads supporting the SV, except INDEL reads as they can be produced as a side-effect of the translocation itself.
- Region from which the reads are taken: Bigger of the following two regions, R1: (start of the first forward read supporting the translocation + two times standard deviation, start of the last forward read supporting the translocation), R2: (start of the last forward read supporting the translocation - average read length, start of the last forward read supporting the translocation)

Inverted Intra-chromosomal translocation

For this case the zygosity calculation has already been made during the inversion call and it is not calculated separately once again.

Inverted Inter-chromosomal translocation

Here the call is made similar to the case of inter-chromosmomal translocation.

5.3.4 k-BIC

See Appendix A for a description of k-BIC.

Span	Cluster size										
coverage⇓	2	3	4	5	6	7	8	9	10	11	12
5x	3.5	3.1	3.0	2.7	2.4	2.3	2.1	2.1*	2.1*	2.1*	2.1*
8x	3.6	3.2	3.0	3.0	2.6	2.5	2.4	2.3	2.2	2.2*	2.2*
10x	3.9	3.2	3.0	2.8	2.7	2.6	2.5	2.3	2.2	2.1	2.1*
15x	N/A	3.8	3.4	3.1	2.8	2.6	2.5	2.4	2.4	2.4	2.3
20x	N/A	N/A	3.4	3.1	2.8	2.7	2.7	2.5	2.5	2.5	2.4
25x	N/A	N/A	N/A	3.4	3.1	2.9	2.8	2.6	2.6	2.5	2.4

Table 5.5: Cutoffs used for given cluster size at given span-coverage

5.3.5 PEMer Algorithm

PEMer (Paired-End Mapper) (Korbel et al. [21]) is an algorithm for detecting structural variants of large insert lengths using paired-end sequencing data. The following are the details:

1. **Pre-processing**: Let the mean library insert length be μ. Let the standard deviation of original library insert lengths be σ. If these quantities are given by the user, then they are used for subsequent calculations. If these are not given, then these quantities are estimated from the data, as explained in Sections 5.3.1.1 and 5.3.1.2 respectively.. In either case, these quantities are respectively denoted by $\hat{\mu}$ and $\hat{\sigma}$, For subsequent analysis these estimates are used.

2. **Read-alignment**: Find all the paired reads that overlap a particular point in the reference. These points are the start of all the reads.

3. **Selection of read-pairs**: Select read-pairs having normal orientation. Discard those which do not have normal orientation.

4. **Identification of deviant read-pairs**: Find the deviant read pairs based on Table 5.5, which gives cut-off values C. Only those reads with insert lengths greater than (mean + $C\times$ the standard deviation), are used for subsequent analyses. This table is adapted from Korbel et al. ([21]) and gives cutoff values for defining deviant read-pairs given in terms of standard deviations (SDs) from the median (we have used mean instead of median) of the expected distribution of insert lengths. For cells marked N/A no cutoffs were reported as the number of observed false-positive calls was extensive. Numbers marked with an * correspond to blank cells in [21] which concern paired-ends with a span deviating from the mean insert size by less than two standard deviations and are not considered in their work. However, in our algorithm, we have used the last value without an * in that span coverage row.

5. **Clustering of deviant read-pairs**: Run the clustering algorithm explained earlier in Section 5.3.2 on all the reads obtained in (2) above. Remove clusters which contain less than a user-specified number d of deviant read-pairs.

6. **SV type identification**: Tag each of the remaining clusters as an insertion cluster or a deletion cluster based on the average inter-mate distance of the cluster---a short one if it is less than $\hat{\mu} - 2\hat{\sigma}$ and a long one if it is greater than $\hat{\mu} + 2\hat{\sigma}$. Discard those that are neither short nor long.

7. **Merging clusters**: Consider the possibility of merging clusters by applying the method described in Section 5.3.2. Consider the cluster with the largest number of reads as the SV cluster.

8. **Detection of SV region**: Detect the SV region as explained in Section 5.3.2.

This approach can detect INDELs of insert length greater than 2σ, since we are discarding those with shorter insert length.

Chapter 6

Copy Number Analysis

6.1 Background and Motivation

Copy number variations (CNVs) are a form of structural variations (SVs) and are defined as large scale amplifications or deletions (typically larger than 1 kbp). Contrary to the earlier belief, CNVs are present in human populations with high frequency and potentially explain more variation than SNPs [41]. In particular, often there are large scale variations with possible deletions of tumor suppressor genes and/or amplifications of oncogenes, in the cancer genome. An example karyotype that is shown in Fig. 6.1 clearly reflects genome-wide large scale changes in cancer.

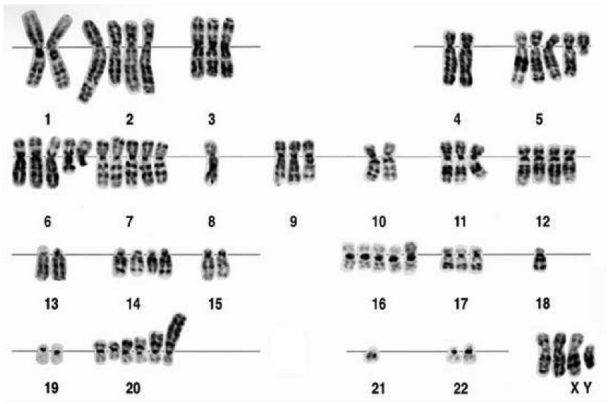

Figure 6.1: Example karyotype showing large scale variations in the cancer genome.

Many studies have already shown the association of CNVs to complex diseases such as cancer, Alzheimer, Autism, etc [42, 43, 44, 45]. Therefore detection and understanding of the implications of CNVs become critical for diagnosis and treatment of various genetic diseases.

Traditionally, fluorescence in situ hybridisation (FISH), array comparative genomic hybridisation (aCGH), and SNP arrays are employed to detect CNV regions but they have their own set of issues and it is generally difficult to detect short CNV regions as the resolution is low, particularly for FISH and aCGH technology [46]. But the recent progress in next generation sequencing (NGS) has enabled us to detect copy number variations (CNVs) at an unprecedented resolution. In this chapter we describe the approach used in **Avadis NGS** for detecting CNVs using the NGS data. The primary objective of this analysis is to detect the CNV regions in individual samples (CNVs in tumor genome for example, with respect to the normal genome) and assign absolute copy number

to each region.

However, several technical and biological challenges inhibit the detection of true CNV regions and assignment of absolute copy numbers. These include:

- Noise/random errors in the data
- Technical bias of the NGS technology due to which the number of reads may not truly indicate DNA content
- Aneuploidy of tumor cells (tumor genomes are often not diploid)
- Tumor sample contamination by normal and stromal cells
- Heterogeneity in tumor cells i.e. there may be polyclonal tumors within a tumor with each clone having different variations and therefore different CNV regions

6.2 CNV detection approach

In this section, we will describe our approach for the detection of CNV regions while addressing some of the issues mentioned above. Most of the approaches for detecting CNVs using NGS data are based on 1) read depth or depth of coverage; 2) distance/orientation of read pairs; and 3) split reads. Since depth of coverage based methods are more commonly employed to detect CNV regions, we have also adopted a similar approach.

In **Avadis NGS** , CNV Detection can be run on whole genome data as well as on targeted reseqeuncing data. In case of whole genome data, the genome is divided into small windows, and read counts in tumor and normal samples are obtained for each of these windows. These read counts are normalized for GC-bias and the respective total sample read counts, and the read count ratios between the tumor and the normal samples are computed. The normalized ratios are then corrected for ploidy and normal cell contamination of the tumor sample. Finally, segmentation is done to detect the CNV regions - in this process adjacent windows are merged if they have similar ratios.

Note: In case of targeted resequencing data, the ratios are computed for each target region without further subdividing. Also, segmentation is not performed.

The details of the main processing steps are given below:

6.2.1 Determining window size

The 'depth of coverage' based methods choose non-overlapping windows or bins representing non-overlapping genomic regions and count the number of reads in each window. The 'window size' is an important factor that determines the length of the shortest CNV region that can be detected. A smaller window size may allow detection of shorter CNVs as well, however, a very small-sized window may be affected by noise and may not have enough reads to generate a robust signal. So, a balance is necessary while determining the window size. A minimum of the two window sizes (computed and user-provided), as defined below, is eventually used. We used the following equation to compute the window size [47]:

$$w_{computed} = \frac{L}{T * CV^2} \tag{6.1}$$

Here L is the total length of the genomic region considered, T is the total number of reads, CV is the coefficient of variation. Recommended range of values for this parameter is $0.05 - 0.10$.

User-provided window size is given by the following equation:

$$w_{user} = \frac{\text{'Min CNV Length'}}{k} \tag{6.2}$$

Here the parameters 'Min CNV Length', and k, provided by the user, are the length of the shortest CNV region the user is interested in detecting (specified in the wizard), and the minimum segment size (specified in Tools Options), respectively.

As mentioned, minimum of the two window sizes is used.

$$w = min(w_{computed}, w_{user}) \tag{6.3}$$

Finally, we impose a lower bound of $200bp$ and an upper bound of $100Kbp$ on the window size, w. Also, as we need one window size for both the tumor and the normal samples, we take the maximum of the two window sizes computed and use it in all the subsequent steps.

6.2.2 Calculating read count for tumor and normal samples

Once we have determined the window size, we count the number of reads that fall in each window. The reads that span across the window boundary are fractionally counted to the windows, based on the percentage of overlap.

6.2.3 GC bias correction and count normalization

It is known that GC content can influence the read counts. Typically, read depths are lower for both high and low GC content, as depicted in Fig. 6.2.

It is important to remove the GC content bias before doing any downstream analysis. There are generally two ways to do it: 1) Mean/Median based approach; and 2) Regression based approach. We use a mean based approach [48] that make use of GC content bins and correct the read count in each window according to the following equation:

$$R_i^c = R_i^o - (m_{g_i} - m) \tag{6.4}$$

Here R_i^o, and R_i^c are the original and the GC-bias corrected read counts respectively, for the i^{th} window. g_i is the GC-percent for the i^{th} window. m is the mean of the read counts across all the windows, and m_{g_i} is the mean of the read counts for all the windows with GC-percent g_i. This is quite intuitive and simple to understand. The formula above increases the read counts of the windows which have either low or high GC content, whereas for windows in which GC content is neither too high nor too low, the read counts will be decreased.

6.2.4 Calculating tumor-normal ratios

Once we have removed the GC content bias from the read counts in each window for both the tumor and normal samples, we normalize the counts by the total read count in the respective sample. This is done because the samples could have been sequenced at different depths and therefore have different average read coverage. We then compute, for each window, the ratio of the normalized corrected read count in the tumor to that in the normal sample. Although this ratio would only make more sense when both the tumor and the normal sample genomes have the same size (or same average ploidy), which in fact is not the case since tumor genomes are often

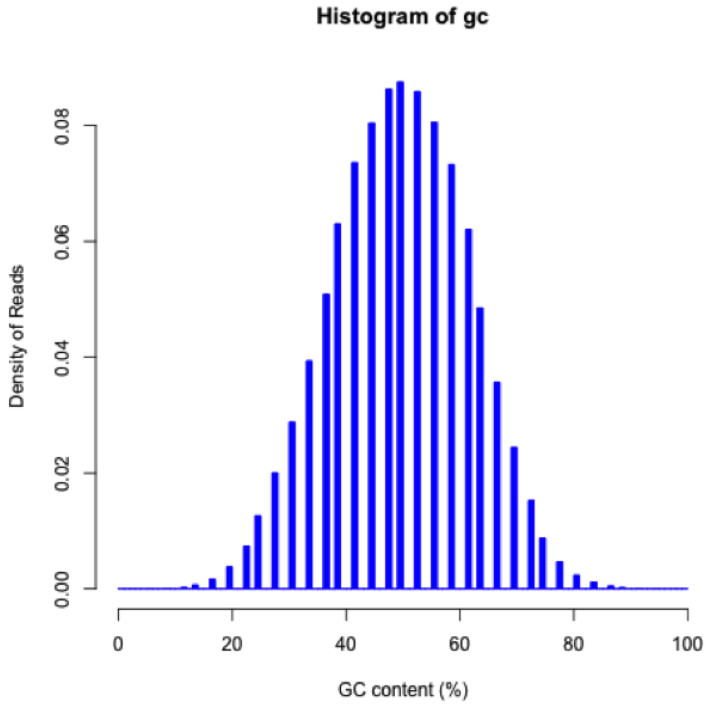

Figure 6.2: An example showing GC bias - Y-axis represent the read depth and X-axis represent the GC content.

aneuploid, this normalization is still a good approximation of the ratio signal, especially when we eventually correct for the average sample ploidy (more on this in the next section).

$$ratio(r_i) = \frac{R_{i,t}/T_t}{R_{i,n}/T_n} \tag{6.5}$$

Here $[R_{i,t}, T_t]$, and $[R_{i,n}, T_n]$ denote the [read count in i^{th} window, total read count] for tumor and normal samples respectively. r_i denotes the tumor-normal ratio for the i^{th} window.

If there were no aneuploidy and no normal cell contamination in the tumor sample, the tumor-normal ratios would take values from the set R : {0, 0.5, 1.0, 1.5, 2.0, ..., and so on}, corresponding to tumor copy numbers CN : {0, 1, 2, 3, 4, ..., and so on}. However, in practice due to aneuploidy of tumor cells and normal cell contamination, ratios deviate from the set R [49]. As a result, copy number (CN) can also deviate from the integer values shown in set CN. There is typically an interplay between average ploidy and contamination (will be explored in the next section), but generally speaking, contamination in the tumor sample moves all the ratios towards 1 (with 50% contamination, a $4X$ amplification may look like a $3X$ amplification; or a $0X$ deletion may look like a $1X$ deletion); and ploidy scales all the ratios in either upward direction (when average sample ploidy is less than 2) or in downward direction (when average sample ploidy is more than 2).

Note: We assign a 'NaN' value to the ratio for windows in which $R_{i,n}$ is less than a specified threshold. This is done to avoid non-robust and erroneous ratio values. Further, we use lower and upper bounds on the calculated tumor-normal ratios - all the ratios greater than 10.0 are clamped to 10.0 and on the lower side they are bounded by 1.0×10^{-10}.

6.2.5 Effect of aneuploidy and normal cell contamination

Aneuploidy shifts the copy number ratio signal in the downward direction if the average tumor sample ploidy is more than 2. Similarly, it shifts the signal in the upward direction if the average tumor sample ploidy is less than 2. The observed copy number ratio signal is given by:

$$r_i^{observed} = \frac{2}{P} r_i^{actual}, \tag{6.6}$$

where P is the average ploidy of the tumor sample, r_i^{actual} is the actual tumor-normal ratio, and $r_i^{observed}$ is the observed ratio. Since the overall signal is scaled in either upward or downward direction, aneuploidy alone does not make the process of segmentation difficult, however accurate assignment of copy numbers to each segment becomes more difficult.

Normal cell contamination shifts the ratio towards the normal ratio, which is 1. Due to this, contrast between adjacent copy number states will reduce and therefore both segmentation and assignment of copy numbers become difficult. Assuming C to be the fraction of normal cells in the tumor sample, observed ratio can be written as:

$$r_i^{observed} = (1 - C) * r_i^{actual} + C * 1 \tag{6.7}$$

Combined effect of aneuploidy and normal cell contamination is given by:

$$r_i^{observed} = \frac{2}{P} \left[(1 - C) * r_i^{actual} \right] + C * 1 \tag{6.8}$$

6.2.6 Estimation of average sample ploidy and normal cell contamination

As mentioned earlier, tumor samples often exhibit aneuploidy and are contaminated by normal and stromal cells. Both these sample-specific biological parameters affect the ratio signal (shrinking towards 1 or scaling or both) and therefore affect the accuracy by which one can detect CNV regions and assign copy numbers. Tumor-normal ratios typically show multi-modal distribution depicting different copy number states. We use the Gaussian mixture model to model this data [49]. However, in case of contamination or aneuploidy, the challenge is that separation in the modes would become smaller as they all shrink towards ratio 1. The interplay amongst sample ploidy, normal cell contamination, and the distance between adjacent Gaussian components is given by the following equation:

$$C = 1 - d * P \tag{6.9}$$

where C is the normal cell contamination (given as a fraction between 0 and 1), P is the average sample ploidy, d is the distance between adjacent Gaussian components in the mixture model. This can be easily derived from Eqn. 6.8 by computing d as the difference between the observed ratios corresponding to two adjacent copy number states.

In fitting a Gaussian mixture model, one needs to estimate the number of components, mean and standard deviation of each component, and mixing proportions (weights) of the components. We do that using the standard EM algorithm. We use an important fact to reduce the search space in the EM algorithm: for any given value of ploidy and contamination, the ratios corresponding to adjacent copy number states lie a fixed distance apart. To illustrate further, let us assume there are three modes in the ratio distribution: first corresponds to deletion (CN = 1), second corresponds to normal region (CN = 2), and third corresponds to amplification (CN = 3). If

average sample ploidy is 2 and the sample has no contamination, then $d = 0.5$, which means if Gaussian component corresponding to $CN = 2$ is at 1.0, $CN = 1$ will be at 0.5, $CN = 3$ will be at 1.5 and so on. Keeping the ploidy same at 2, if we have a 20% contamination, then $d = 0.4$ which will change the position of all the Gaussian components by the same amount. Similarly for 40% contamination, we get $d = 0.30$. Note that if average sample ploidy is more than 2, d will decrease further. In summary, no matter what the value of d is, it is constant between the peaks for a given set of ploidy and contamination values. From equation 6.9 above, we get minimum value of 'd' when C and P are maximum. In a nutshell, we can fix few values of 'd' and therefore Gaussian component means and use the EM algorithm to update other unknown parameters in each iteration. Theoretically, one could consider a full range of contamination (0% - 100%) and a larger range of ploidy, however, at higher contamination and higher ploidy, it become practically very difficult to segregate the multiple modes (or copy number states) in the data. Therefore, we limit the values of detectable contamination and ploidy to 40% and 3 respectively. This boils down to a value of $d = 0.20$, and this is the minimum distance between the adjacent Gaussian components that we consider in our approach while modelling the ratio data.

Steps in model fitting: As explained above, we follow a fixed mean Gaussian mixture modelling approach by first anchoring at the highest ratio peak and then placing a series of components on either side with uniform spacing. To do this, we bin the ratio data and set the anchor at the bin with highest density. Then for each $d \in \{0.20, 0.21, 0.22, ..., 0.60\}$, we fit a Gaussian mixture model and using the EM algorithm, estimate the variance and mixture weight of each component. Number of Gaussian components are determined by anchor point and 'd' itself. Once we have fitted the Gaussian mixture models with this fixed mean approach for all values of d, we can compute BIC (Bayesian Information Criterion) to select the best model. All the Gaussian components that have estimated mixture weight of less than 0.1% are discarded as noise, and are ignored while computing the BIC. The BIC criterion is very similar to popular AIC (Akaike's Information Criterion), but has a larger penalty term. These model selection approaches become very useful because it is often possible to increase the likelihood of the data by adding more parameters to the model but doing so may result in over fitting. This is when AIC and BIC criteria try to strike a balance. We experimentally evaluated both AIC and BIC and found BIC criterion to be more suitable for this problem. Once the best Gaussian mixture model is determined and the corresponding value of d is available, we use Eqn. 6.9 and follow the steps below to estimate the contamination and ploidy.

- *Case 1: User has not provided the expected values of either contamination or ploidy* - In this case we assume that $CN = 2$ is the most predominant copy number state. So we assign a copy number of 2 to the component with the highest weight, and assign the copy numbers to the components on either side accordingly. We then compute the average ploidy as the weighted average of the copy numbers assigned to the various components, as given in the following equation:

$$P_{estimated} = \sum w_i * c_i, \tag{6.10}$$

  where w_i, and c_i are the weight and the assigned copy number of the i^{th} Gaussian component. Once ploidy P is estimated, we can use Eqn. 6.9 to estimate the normal cell contamination C.

- *Case 2: User has provided the expected value of contamination but not of ploidy* - In this case, we use a different approach for assigning copy number values to the estimated Gaussian components. After discarding the components with mixture weight less than 0.1%, the first component (minimum mean) is considered for copy number value of 0, 1, or 2. For each of these possibilities we assign copy numbers to other components, and estimate the ploidy and contamination as in *Case 1.* This way, we get 3 values of the contamination, out of which

we choose the one that is closest to the expected contamination provided by the user. The corresponding ploidy is chosen as well.

- *Case 3: User has provided the expected value of ploidy but not contamination* - The approach followed in this case is similar to the one followed in *Case 2* above. Amongst the 3 possible ploidy estimates, we choose the one which is closest to the expected ploidy provided by the user. Again, once we have the final estimate of Ploidy P, we use Eqn. 6.9 to estimate the contamination C.

6.2.7 Correcting ratio track

Now that we have estimated the average sample ploidy (P) and normal cell contamination (C), we can correct the observed ratios to obtain the estimates for the actual ratios before doing the segmentation. Solving the equation 6.8, we get

$$r_i^{actual} = \left(\frac{r_i^{observed} - C}{1 - C} \right) \frac{P}{2}, \tag{6.11}$$

where C, and P are the normal cell contamination and the average ploidy of the tumor sample estimated as described in the previous section.

6.2.8 Segmentation

After we have corrected the tumor-normal ratios using estimated ploidy and normal cell contamination, we can use any simple segmentation approach to detect CNV regions and assign absolute copy numbers to the detected regions. For this purpose, we use a moving average based approach. The steps are described below:

- Iterate over the windows until some non-normal window is found. A tolerance factor of 0.3 is used to classify windows as normal or non-normal. For example, all windows with ratio value between $0.7 - 1.3$ are considered normal (CN = 2); windows with ratio value between $1.2 - 1.8$ are considered to have a CN of 3 and so on. There is an overlap region in these ranges. So, if the window has a ratio value of 1.25, it can either be considered as normal with $CN = 2$, or as non-normal with $CN = 3$, depending on the context.
- Starting at a non-normal window, we evaluate the 'genomic segment' corresponding to k windows at a time (k is the minimum segment size specified in Tools Options) and compute the tumor-normal ratio of the region. Please note we do not take the average of already computed window ratios. Instead, we sum the (GC-bias corrected) reads in k windows separately for tumor and normal samples, normalize them using total read count in respective sample, and use these sums to compute the ratio. This way we get more robust estimates as compared to taking the mean of the ratios across the windows.
- We then shift the 'genomic segment' (which consists of k windows) by one window at a time and we keep going as long as the ratio of the 'moving segment' corresponds to the copy number of the first segment. If it differs, we stop.
- Finally, we backtrack and discard individual windows that by themselves have ratio value outside the tolerance band.

Merging of regions: Once all the CNV regions are detected using the above moving average based method, we merge adjacent regions based on two criteria: 1) gap between the regions; and

2) difference in copy number values between the two regions. Both of the following two conditions should be satisfied before the CNV regions are merged:

$$\frac{gap}{\text{Total size of merged region}} \leq a\%$$

'Total size of merged region' is calculated from start of the first region up to end of the second region, and the default value of a in the tool is 10%.

$$\frac{CN_1 - CN_2}{(CN_1 + CN_2)/2} \leq b\%$$

Default value of b is 20%. Due to the second condition, there is a higher tendency to merge regions with larger copy numbers compared to smaller copy numbers. For example, let us consider two sets of two regions each that are candidates for merging. The two regions in the first set have $CN = 3$ & $CN = 4$; and in the second set have $CN = 6$ & $CN = 7$. Let us also assume that regions in both sets satisfy the gap criterion. However, second condition will only be satisfied for the second set and not the first set. Due to precisely this reason, second criteria may not be appropriate for dealing with deletion regions. Therefore, we change the second criteria for deletions to require that closest assigned integer copy numbers for the two regions considered for merging, should be same. The first criteria to check gap remains unchanged. Please note that due to aneuploidy and normal cell contamination, copy number values may not be integer values.

Filtering regions based on minimum CNV length: At the end, before we output the results, regions whose lengths are less than the 'Minimum CNV length' are discarded.

Appendices

Appendix A

k-BIC

A.1 Introduction

The k-BIC algorithm is an application of the k-means algorithm to make k clusters from n objects, for various values of k from $k = 1$ to $k = K$, a given number, followed by the application of the Bayes Information Criterion (BIC) to determine the optimal number of clusters in the range 1 to K. This algorithm is used to determine the appropriate number of groups in a cluster analysis situation. In cluster analysis, the aim is to make clusters so that there is homogeneity within clusters, generally measured by the sum of squared distances of all pairs within a cluster. Given n observations, the larger the number of groups (k), the more will the homogeneity be; but this will overfit the cluster model. To guard against this a cluster analysis criterion is modified by introducing a penalty term to the criterion.

The BIC criterion is applied in several contexts in **Avadis NGS**. For instance in the PICS algorithm the likelihood criterion is modified by the BIC criterion. In Structural Variation detection where the appropriate number of clusters are determined, the sum of squares of distances criterion is offset by a penalty term as per the BIC criterion.

In what follows, we discuss the details of this approach.

A.2 k-means Algorithm

The k-means algorithm is a method of forming k disjoint clusters of n data points for a chosen number k. A measure of distance between points (metric) relevant to the nature of the data points is chosen and the object of the algorithm is to form k groups in such a way that each observation is assigned to the cluster with the nearest (in the sense of the chosen metric) mean. The criterion for forming clusters is the minimization of the sum of squares of distances (again, in the sense of the chosen metric) between pairs of points within clusters, summed over clusters. From the clusters finally arrived at, parameters such as cluster proportions, cluster means, etc. are estimated, if required.

Details of the k-means Algorithm

1. Choose k data points as initial centroids either based on heuristics or at random.
2. Repeat steps 3 to 5 below until the stopping/convergence criterion stated below is met.
3. For each data point x_i:

(a) compute the distance of x_i to each (current) centroid.

(b) identify x_i with the closest centroid.

4. Form k clusters based on the above assignments.

5. Recompute new cluster centroids as means of the data points assigned to that cluster.

The stopping/convergence criteria generally used are:

1. Minimum number of re-assignments of data points to different clusters between successive iterations.

2. Minimum number of changes of centroids between successive iterations.

3. Minimum decrease in the within-cluster sum of squares of distances---sum of squares of distances between all pairs of data points within a cluster, summed up over all clusters (SSE).

A.3 Model and Likelihood

The above method is not based on any model for the data. But subsequent analysis in Structural Variation application is based on the univariate normal mixture model as follows. A similar approach is valid for multivariate normal and other mixtures. A normal mixture model for a univariate random variable X describes the probability density of X as

$$f(x;\Theta) = \sum_{j=1}^{k} p_j \phi(x;\mu_j,\sigma_j), \tag{A.1}$$

where $\Theta = (\mu_1,\mu_2,\ldots,\mu_k;\sigma_1,\sigma_2,\ldots,\sigma_k;p_1,p_2,\ldots,p_k)$ with $\sum_{j=1}^{k} p_j = 1$; $\phi(x;\mu,\sigma)$ is the probability density of the normal distribution with mean μ and standard deviation σ. Here we assume that σ_j are all equal and equal to σ.

Let $x_i, i = 1,2,\ldots,n$ be the observations, assumed to be a random sample from $f(x;\Theta)$ in (A.1). The likelihood based on the mixture density (A.1) is

$$\prod_{i=1}^{n} f(x_i;\Theta) = \prod_{i=1}^{n}\sum_{j=1}^{k} p_j \phi(x;\mu_i,\sigma), \tag{A.2}$$

Upon convergence of the k-means algorithm, parameter estimates are computed as follows: Let n_j be the number of data points in the j^{th} cluster, $j = 1,2,\ldots,k$.
Let x_{ij} be the data points in cluster j, $i = 1,2,\ldots,n_j, j = 1,2,\ldots,k$.

- $\hat{p}_j = \frac{n_j}{n}$.
- $\hat{\mu}_j = \bar{x}_j = \frac{1}{n_j}\sum_{i=1}^{n_j} x_{ij}$.
- $\hat{\sigma}^2 = \frac{1}{n-k}\sum_{j=1}^{k}\sum_{i=1}^{n_j}(x_{ij}-\bar{x}_j)^2$.

Let

$$\hat{\Theta} = (\hat{\mu}_1, \hat{\mu}_2, \ldots, \hat{\mu}_k; \hat{\sigma}, \hat{p}_1, \hat{p}_2, \ldots, \hat{p}_k)$$

denote the estimates of the parameters obtained as above. Then the observed likelihood is

$$L_O = \prod_{i=1}^{n} f(x_i; \hat{\Theta}) = \prod_{i=1}^{n} \sum_{j=1}^{k} \hat{p}_j \phi(x_i; \hat{\mu}_j, \hat{\sigma}), \tag{A.3}$$

The algorithm is implemented for a range of values of k, and the best model (optimal value of k) is selected using a criterion discussed below.

A.4 Model Selection Criteria

Model building and model comparison criteria are based on the principle of *parsimony*, whereby one wishes to develop a model with as few parameters as possible, at the same time fitting the data reasonably well. Thus one aims a trade-off between a good fit and the number of parameters. This is the well-known problem of bias vs variance. In our particular example, by increasing the number of clusters and hence the number of parameters, the model can be made to fit very well, with the likelihood increasing with the number of clusters and number of parameters. However, a model with a good fit with a large number of parameters does not necessarily generalize due to biases introduced by over-fit. Hence one adjusts the likelihood (or equivalently log-likelihood) by a *penalty* term. Many of the often-used such penalized likelihood criteria are obtained using concepts and methods of information theory. They are called *information criteria* (IC). The general form of these criteria for p parameters is:

$$IC(n,p) = -2\log(\text{likelihood})(\boldsymbol{x}|\hat{\boldsymbol{\theta}}_p) + \alpha(n,p),$$

where $\boldsymbol{x}$ represents the observations and $\hat{\boldsymbol{\theta}}_p$ represents the maximum likelihood estimates of the p parameters, and $\alpha(n,p)$ represents the penalty function, a function which increases with n and p. Model choices (choice of p) are generally made such that IC(n,p) is minimized. Of course, n is taken as given.

The most well-known and often-used criteria are **Akaike Information Criterion (AIC)** ([1]) and **Schwarz's Bayesian Information Criterion (BIC)** ([27]). AIC uses

$$\alpha(n,p) = \alpha(p) = AIC(p) = 2p,$$

and is based on its asymptotic (with respect to n) behavior and has been often found to under-penalize the log-likelihood and consequently overparameterize the model.. The BIC criterion depends also on the sample size n. It has been derived from two different points of view. One of them, by Schwarz ([27]), is based on Bayesian considerations and the predictive power of the model; however, the criterion turns out to be independent of the prior. The other derivation of the same criterion is by the use of a coding technique for parameterized density using the notion of minimum description length(MDL) by Rissanen ([26]). The criterion is

$$\alpha(n,p) = BIC(n,p) = p\log n,$$

where n is the number of observations. Many other forms have been suggested in the literature (for instance, Olivier et al. ([25]) for the penalty term in the form $c(n)\alpha(p)$ with various functions $c(n)$ and $\alpha(p)$, especially for $c(n)$ growing at different rates with n like $\log n$, $\log\log n$, and functions with growth rates in-between. We found the BIC to yield most satisfactory results.

In our problem the number of parameters is $p = 2k$, where k is the number of clusters; the parameters are the k normal means $\mu_j, j = 1, 2, \ldots, k$, the $k-1$ proportions $p_j, j = 1, 2, \ldots, k$ (since the k proportions add up to 1), and the common standard deviation σ.

A.5 BIC (Bayes Information Criterion) Algorithm

1. Choose an integer K.
2. For each value of $k = 2, 3, \ldots, K$, compute the following:
3. Compute k clusters using the k-means algorithm:
4. Let $x_{ij}, j = 1, 2, \ldots, n_i, i = 1, 2, \ldots, k,$ (with $\sum_{i=1}^{k} n_i = n$) be the n insert lengths rearranged with x_{ij} being the elements in the i^{th} cluster.
5. Compute means $\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_k$ in the k clusters. $\bar{x}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} x_{ij}$.
6. Compute $p_i = \frac{n_i}{n}, i = 1, 2, \ldots, k$, the proportions of elements in the k clusters.
7. Compute a common standard deviation s where

$$s^2 = \frac{1}{n-k} \sum_{i=1}^{k} \sum_{j=1}^{n_i} (x_{ij} - \bar{x}_i)^2.$$

8. Let $f_i(x; \bar{x}_i, s)$ denote the density at x of a normal (Gaussian) distribution with mean $\bar{x}_i$ and standard deviation s.
   Compute log-likelihood as

$$\ell(k) = \sum_{m=1}^{n} log[\sum_{i=1}^{k} p_i f_i(x_m; \bar{x}_i, s)].$$

9. Compute BIC(k) as

$$\text{BIC}(k) = -2\ell + 2k \log(n).$$

10. The optimal number of clusters is given by

$$\hat{k} = \arg\min \text{BIC}(k).$$

11. Equivalently compute BIC(k) as

$$\text{BIC}(k) = 2\ell - 2k \log(n).$$

    and the optimal number of clusters is given by

$$\hat{k} = \arg\max \text{BIC}(k).$$

12. Output $\hat{k}$ as the number of components in the mixture.

Appendix B

EM Algorithm

B.1 Introduction

In many features of **Avadis NGS**, the Expectation--Maximization algorithm (EM Algorithm) is being used to compute maximum likelihood estimates (MLEs) of parameters involved in the model. These features include PICS, GADEM, and isoform quantification. In all these features, the basic problem is one of resolving a mixture of distributions, which is one of the most significant applications of the EM algorithm. In this Appendix we give an introduction to the basic ideas of the EM algorithm, followed by a description of the algorithm as it applies to the mixture resolution problem. In individual chapters describing the features or in appendices to these chapters, more details of the algorithm specific to the feature are given.

B.2 Incomplete- and Complete-Data Problems

The EM algorithm is a general-purpose technique for computing MLEs when the likelihood function is complicated due to

1. the nature of the statistical model from which the data arise such as mixture of distributions, truncated distributions, and the like;
2. such data-related complications as missing data, latent data, censored data, and the like;
3. both of the above;
4. other reasons.

For convenience, these problems are dubbed **incomplete-data problems**, even if there is no data missing in terms of the experiment or study design; the reason for the terminology is their connection with **complete-data problems** as explained below. In such situations, standard techniques of MLE computation such as Newton-Raphson method or its variant in the form of Fisher's scoring method are far too complicated to apply. In many such situations, the EM algorithm can provide a simpler and more elegant way of computing the MLE.

The EM algorithm relies on the ability of the analyst to find a related complete-data problem, the

1. data model for which contains all the parameters of the incomplete-data problem; and

2. the sample space of which can be mapped to that of the incomplete-data problem---often the difference between the sample spaces is caused by **missing data** or **latent data**, the availability of which will make the incomplete-data problem a complete-data problem.

For instance, a cluster analysis problem (an unsupervised learning problem) where the number of groups is known but the group memberships of data points are unknown, can be regarded as an incomplete-data problem; a related problem where group memberships of data points are also known (a supervised learning problem) can be regarded as a corresponding complete-data problem. Here the latent or missing piece of information is the group to which a data object belongs. Generally, the parameters involved are estimated in a much-easier way if the latent data are available.

B.3 E- and M-steps of the EM algorithm

1. In many cases a complete-data problem may be a naturally occurring problem in the context.

2. Anyhow, the complete-data problem should have a MLE which is simpler to compute.

3. Since the logarithm is an increasing function, maximizing the likelihood $L(\mathbf{\Theta})$ with respect to parameters $\mathbf{\Theta}$ is equivalent to maximizing $\ell(\mathbf{\Theta}) = \log(L(\mathbf{\Theta}))$ with respect to $\mathbf{\Theta}$.

4. The EM algorithm is an iterative algorithm where each cycle of the algorithm has two steps, the E-Step or the expectation Step and the M-Step or the maximization step.

5. The algorithm starts with some arbitrarily, randomly or intelligently chosen initial values of parameters, say $\mathbf{\Theta}^{(0)}$, applies the E-Step as explained below.

6. It then carries out the M-Step as explained below.

7. At the end of the M-step, an update $\mathbf{\Theta}^{(1)}$ is computed,

8. which is put through another cycle of E-and M-steps to produce $\mathbf{\Theta}^{(2)}$ and so on, until the sequence converges to a value $\hat{\mathbf{\Theta}}$ which will be the MLE of $\mathbf{\Theta}$.

Under fairly liberal conditions, the algorithm is known to converge to the MLE of the incomplete-data problem.

B.3.1 E-step

Given current update $\mathbf{\Theta}^{(t)}$, the E-step in a sense imputes the missing values, using which a surrogate for the complete-data log-likeihood is computed. More precisely, the E-step computes the conditional expected value $\ell(\mathbf{\Theta}^{(t)})$ of the complete-data log-likelihood using $\mathbf{\Theta}^{(t)}$ given the actually observed data, the expected value being computed with respect to the distribution of the missing or latent variables.

B.3.2 M-step

Using this conditional expected value $\ell(\mathbf{\Theta}^{(t)})$ as a surrogate for the complete-data log-likelihood, and exploiting the simpler MLE computation for the complete-data case, the M-step produces an update $\mathbf{\Theta}^{(t+1)}$.

B.4 Resolution of Normal Mixtures

As an example of the EM algorithm, we discuss a problem of resolution of normal mixtures. A normal mixture model for a univariate random variable X describes the probability density of X as

$$f(x;\Theta) = \sum_{j=1}^{k} p_j \phi(x;\mu_j,\sigma_j), \tag{B.1}$$

with known k and where $\Theta = (\mu_1,\mu_2,\ldots,\mu_k;\sigma_1,\sigma_2,\ldots,\sigma_k;p_1,p_2,\ldots,p_k)$ with $\sum_{j=1}^{k} p_j = 1$; $\phi(x;\mu,\sigma)$ is the probability density of the normal distribution with mean μ and standard deviation σ. Here we assume that σ_j are all equal and equal to σ.

The mixture resolution problem is an unsupervised learning problem of estimating the parameters Θ from a random sample on X, the mixture variable. It is well known that when the components of the mixture are normal (Gaussian) then it is, in principle, possible to uniquely resolve the mixture into its components, including the mixture proportions. There are many standard algorithms for this kind of mixture resolution when the number of components is known, the most often-used one being the EM algorithm.

Let $x_i, i = 1,2,\ldots,n$ be the observations, assumed to be a random sample from $f(x;\Theta)$ in (B.1). The likelihood based on the mixture density (B.1), that is, the incomplete-data likelihood is

$$\prod_{i=1}^{n} f(x_i;\Theta) = \prod_{i=1}^{n}\sum_{j=1}^{k} p_j \phi(x;\mu_i,\sigma), \tag{B.2}$$

and is rather difficult to maximize with respect to Θ directly.

B.4.1 EM Algorithm

B.4.1.1 Missing Data Variable

We discuss the case where $k = 2$. The procedure is quite the same for a general known k and even for a multidimensional variable X.

The incomplete-data likelihood function is

$$L = \prod_{i=1}^{n} f(x_i;\Theta) = \prod_{i=1}^{n}\sum_{j=1}^{2} p_j \phi(x_i;\mu_j,\sigma), \tag{B.3}$$

where $\phi(x_i;\mu_j,\sigma)$ is the density at x_i of $\mathcal{N}(\mu_j,\sigma)$. This is rather difficult to maximize, since $\ell = \log L$ is the sum of log of sums.

Notice that $p_1 + p_2 = 1$ and so we denote p_1 by p and p_2 by $1-p$.

Let Z_i be the variable denoting the group to which a data point x_i belongs. Let $Z_i = 1$ if the data point x_i belongs to group 1; else 0. This is the supervisor's data (group membership data), which is missing or latent in the mixture resolution incomplete-data problem. Z_i is a Bernoulli variable with $Prob(Z_i = 1) = p$ for all i and are independent. Note that $E(Z_i) = p$. If we have complete data (that is data z_i on Z_i also), then the data from the two groups are separated and the MLEs of μ_1, μ_2 are

$$\hat{\mu}_1 = \bar{x}_1, \hat{\mu}_2 = \bar{x}_2,$$

the respective group means.
The common variance is estimated by

$$\hat{\sigma}^2 = \frac{1}{n_1 + n_2}[\sum_{Z_i=1}(x_i - \bar{x}_1)^2 + \sum_{Z_i=0}(x_i - \bar{x}_2)^2].$$

$$\hat{p} = \frac{1}{n}\sum_{i=1}^{n} z_i.$$

B.4.2 Complete-Data Likelihood

Let us look at the EM algorithm with the complete-data problem mentioned here where Z_i is the missing information of group membership. The complete-data likelihood is

$$L_c = \prod_{i=1}^{n}[pf_1(x_i)]^{z_i}[(1-p)f_2(x_i)]^{1-z_i}.$$

The complete-data log-likelihood is

$$\ell_c = \sum_{i=1}^{n}\{[z_i \log p + (1-z_i)\log(1-p)] + [z_i \log f_1(x_i) + (1-z_i)\log f_2(x_i)].$$

B.4.3 E-Step for the Mixture Problem

Let us denote the data by $\boldsymbol{x}$ and the parameters p, μ_1, μ_2, σ collectively by $\boldsymbol{\theta}$.

- The E-Step consists in working out $E(\ell_c|\boldsymbol{x}, \boldsymbol{\theta})$.
- This involves the working out of $E(Z_i), i = 1, 2, \ldots, n$.
- Z_i depends only on x_i not on all $\boldsymbol{x}$, and so $E(Z_i|\boldsymbol{x}, \boldsymbol{\theta}) = E(Z_i|x_i, \boldsymbol{\theta})$.
- Since Z_i is a 0-1 variable, $E(Z_i|x_i, \boldsymbol{\theta}) = P(Z_i = 1|x_i, \boldsymbol{\theta}) = w_i$, say.
- This is just the **posterior probability** of $Z_i = 1$ given data calculated at current iteration value of the parameters.
- An application of **Bayes Theorem** on conditional probability gives

$$w_i^{(t)} = E(Z_i|\boldsymbol{x}_i, \boldsymbol{\theta}) = \frac{p_1^{(t)} f_1(x_i; \mu_1^{(t)}, \sigma^{(t)})}{p^{(t)} f_1(x_i; \mu_1^{(t)}, \sigma^{(t)}) + (1-p)^{(t)} f_2(x_i; \mu_2^{(t)}, \sigma^{(t)})}$$

B.4.4 M-Step for the Mixture Problem

- The M-Step is the familiar MLE for the supervised complete-data problem, but with a difference.
- In the supervised problem Z_i's are either 1 or 0, but in the surrogate log-likelihood $E(\ell_c|\boldsymbol{x}, \boldsymbol{\theta})$, observation x_i belongs to group 1 with probability $w_i^{(t)}$ and to group 2 with probability $1 - w_i^{(t)}$ at the t^{th} iteration.
- Thus in the M-Step MLE, the observations are involved with these weights.
- The update formulas from iteration t to iteration $t+1$ are given below.
- The EM algorithm then is an application of these E-and M-steps in an iterative procedure until convergence.

B.4.5 M-Step Formulas for the Mixture Problem

$$p^{(t+1)} = \frac{\sum_{i=1}^{n} w_i^{(t)}}{n}$$

$$\mu_1^{(t+1)} = \frac{\sum_{i=1}^{n} w_i^{(t)} x_i}{\sum_{i=1}^{n} w_i^{(t)}} \qquad \mu_2^{(t+1)} = \frac{\sum_{i=1}^{n} (1 - w_i^{(t)}) x_i}{\sum_{i=1}^{n} (1 - w_i^{(t)})}$$

$$\sigma^{2^{(t+1)}} = \frac{1}{n} \sum_{i=1}^{n} [w_i^{(t)} (x_i - \mu_1^{(t+1)})^2 + (1 - w_i^{(t)})(x_i - \mu_2^{(t+1)})^2]$$

B.4.6 Assignment to Groups

If assignment of each data point to a group or if a new data point x is to be assigned to one of the two groups, then once the parameter estimates have been computed,

- **hard assignment**: each data point x can be assigned to that group for which the $p_j f(x_i; \mu_j, \sigma)$ is the largest, $p_1 = p, p_2 = (1 - p)$;
- **soft assignment**: for each data point a probability distribution over the groups can be computed;
- both the above assignments can be performed with the output from the last iteration results of the EM algorithm, using the results of the E-Step.
- the observed likelihood L_O can be computed plugging in the MLEs in L for the parameters, rather $\ell_O = \log L_O$, the observed log-likelihood can be computed, which is useful for further work of model selection, as explained in Appendix A.

B.4.7 General Mixture Resolution Application

The variations of this problem are:

- The number of groups may be > 2;
- The variable may be multidimensional, say d-dimensional.
- The multidimensional analog of this problem is where the d-dimnensional observational variable $\boldsymbol{X}$ has a multivariate normal distribution with different mean vectors $\boldsymbol{\mu}_j, j = 1, 2, \ldots, k$ in the k groups. with covariance matrices Σ_j or a common covariance matrix Σ.
- More generally the k distributions may belong to a more general family $\mathcal{F}$ of distributions than multivariate normal.
- Only certain families of distributions admit solutions to the mixture resolution problem, the family of multivariate normal distributions being one such.
- When the family admits a resolution, the approach outlined above will generally work, the analytical work required being
  - the derivation of the conditional expectation involved in the E-step;

  - the derivation of the MLEs of the parameters in the complete-data problem.

- In some problems instead of the likelihood function, penalized likelihood functions are used.
- In Bayesian problems (e.g., the PICS algorithm explained in Appendix 2.3.1, instead of MLEs, maximum *a posteriori* probability (MAP) estimates are computed by a very similar technique.

Like most optimization algorithms, the EM algorithm finds only the local maximum with respect to the initital values used to start the iteration process. In practise, it is advisable to start the algorithm at diverse points in the parameter space and locate the maximum of the local maxima in an effort to find the global maximum.

The convergence criteria to be used is a matter that needs careful attention. The rate of convergence and the number of ierations needed for convergence of the EM algorithm depends on the difference between the complete and incomplete problems in terms of the amount of latent or missing information. It is not easy *a priori* to specify the number of iterations run for the EM algorithm. It is generally safer, if one can afford the required computing resources, to choose a fairly large value for the number of iterations. A better option is to run the algorithm until successive values of each parameter estimate or successive values of the log-likelihood is reduced by a pre-determined rate.

Appendix C

Read Count Normalization

The quantification step in RNA-Seq and Small RNA analyses experiments computes 'normalized counts' of reads for every gene for each sample from the raw read counts. This process is called 'Normalization', and **Avadis NGS** provides various options to do this normalization. The different normalization methods and the subsequent baselining operation are explained below.

C.1 Normalization using DESeq Approach

This method is based on the paper 'Differential expression analysis for sequence count data' by Simon Anders and Wolfgang Huber [12]. In this, the normalized counts are computed as described below. Refer to the paper for complete details.

- Let r_{ij} be the raw count of gene g_i in sample S_j.
- For each gene g_i, calculate the geometric mean of the raw counts of all the samples for that gene. Let it be m_i.
- For each sample S_j, calculate the normalization factor N_j as the median of the values r'_{ij}, where $r'_{ij} = \frac{r_{ij}}{m_i}$. Note that while computing this median, the genes with $m_i = 0$ are ignored.
- Finally compute the normalized counts n_{ij} for gene g_i in sample S_j as $\frac{r_{ij}}{N_j}$.

C.2 Normalization using the TMM approach

This method is based on the paper 'A scaling normalization method for differential expression analysis of RNA-seq data' by Mark Robinson and Alicia Oshlack [13]. Computation of the normalization factors is quite involved in this approach. Refer to the paper for details of the steps involved.

C.3 Quantile Normalization

This approach tries to make the distribution of all the samples the same. The following are the steps involved:

- For every sample, rank of all the entities is calculated.

- A 'mean' array is calculated using the entities of same rank across all the samples, i.e., k^{th} entry in this array will be the mean of raw counts of entities of rank k in all the samples.
- Then for every sample, normalized count for an entity with rank k is set to the k^{th} entry in the 'mean' array computed above.

C.4 Normalize Total Sample Read Count

Let t_i be the total read count for sample S_i, and let m be the median of these total read counts for all the samples. Then the normalized counts for the sample S_i are obtained by dividing the raw counts by $\frac{t_i}{m}$. With this approach, the total sample read count will become the same for all samples after the normalization.

C.5 Baselining

Baselining essentially subtracts the mean or meadian (as per the option chosen) of the log-transformed normalized counts of the chosen set of samples (all samples or a subset of samples chosen as control set) from the log-transformed normalized count of every sample.

Appendix D

Multiple Testing Correction

D.1 Adjusting for Multiple Comparisons

Statistical tests are often carried out simultaneously on entities running into several thousands and even tens of thousands in number. This leads to the following problem: Suppose there are m entities. Suppose, for instance, the tests are for differential expression of the entities under a certain set of conditions, the null hypotheses in each case being no differential expression. Suppose a p-value for each entity has been computed and all entities with a p-value of less than 0.01 are considered for rejecting the corresponding null hypothesis. Let k be the number of such entities. Each of these entities has a less than 1 in 100 chance of appearing to be differentially expressed by chance if they are not really differentially expressed (that is, if the null hypotheses are true). However, the chance that *at least* one of these k entities appears differentially expressed by chance is much higher than 1 in 100, when none of these entities is differentially expressed. In fact, this could be as high as $1-(1-0.01)^k$ approximately equal to $k \times 0.01$ (if the test statistics for these entities are assumed to be independently distributed). As an analogy, consider fair coin tosses, each toss producing a head with a probability of 0.5; the chance of getting at least one head in a hundred tosses is much higher than 0.5; it is about $1-(0.5)^{100}$ which is very close to 1. Thus if many tests are carried out simultaneously each at a level of significance of 0.01, then taking all tests together the level of significance is no longer 0.01 but much higher. Thus if the overall level of significance called Family-Wise Error Rate (FWER) is to be controlled, some adjustment or correction for the level of significance of each test (called Comparison-Wise Error Rate---CWER) is to be made so that the overall level of significance and hence the proportion of false positives is controlled to a desired extent. The FWER controls the proportion of hypotheses declared significant among all the hypotheses that are actually null (false positives).

Another approach taken in this context takes a different point of view. It is the control of False Discovery Rate (FDR) which is the proportion of hypotheses that are actually null among those that are declared significant.

Avadis NGS offers the following six types of multiple testing corrections, the first three controlling Family-Wise Error Rate (FWER). and the remaining three controlling False Discovery Rate (FDR).

1. Bonferroni
2. Bonferroni Step-down (Holm)
3. Westfall-Young
4. Benjamini-Yekutieli

5. Benjamini-Hochberg
6. Storey q-value

See Dudoit et al. [14] and Glantz [19] for detailed descriptions of various algorithms for adjusting p-values and Storey [29], [30] for q-value and FDR.

The methods are listed in order of their stringency, with the Bonferroni being the most stringent, and the Storey q-value being the least stringent. The more stringent a multiple testing correction, the less proportion of false positive entities results. The trade-off of increasing stringency of a multiple testing correction is that the rate of false negatives (entities that are called non-significant when they are significant) increases.

We present below a few examples. In the examples, an error rate of 0.05 and a entity list of 1000 entities are assumed.

D.2 Bonferroni

Bonferroni method is a single step procedure, where each p-value is corrected independently of others. The p-value of each entity is multiplied by the number of entities in the entity list. If the corrected p-value is still below the chosen significance level, the entity is declared significant. Corrected p-value= p-value $\times n$ (number of entities in test). As a consequence, if testing 1000 entities at a time at the same p-value, the highest accepted individual p-value for declaring significance is 0.00005, making the correction very stringent. With a FWER of 0.05 (i.e., the probability of at least one error in the family), the expected number of false positives will be 0.05.

D.3 Bonferroni Step-Down

Holm's test is a stepwise method, also called a sequential rejection method, because it examines each hypothesis in an ordered sequence, and the decision to accept or reject a null hypothesis depends on the results of the previous hypothesis tests.

Entities are sorted in increasing order of p-values. The p-value of the j^{th} entity in this order is now multiplied by $(m - j + 1)$ to get the adjusted p-value. Because it is a little less corrective as the p-value increases, this correction is less conservative.

Example:

Gene Name	p-value before correction	Rank	Correction	Is entity significant after correction
A	0.00002	1	$0.00002 \times 1000 = 0.02$	$0.02 < 0.05 \rightarrow Yes$
B	0.00004	2	$0.00004 \times 999 = 0.039$	$0.039 < 0.05 \rightarrow Yes$
C	0.00009	3	$0.00009 \times 998 = 0.0898$	$0.0898 > 0.05 \rightarrow No$

D.4 Westfall-Young

The Westfall-Young [32] permutation method takes advantage of the dependence structure between entities, by permuting data in entities between groups (conditions) in all the entities at the same time. Suppose the tests are two-sample t-tests with n_1, n_2 samples in groups (conditions) 1 and 2 respectively. Then the $n_1 + n_2$ sample units are permuted by allotting at random n_1, n_2 of the units to the two conditions. The same permutation of units will apply to all entities. Thus in all the entities the data in the two groups will also be permuted likewise. The same test is conducted

as before in each of the entities. Thus each permutation will give rise to a different set of p-values for the entities. A similar approach is used for any other test.

The Westfall-Young procedure is a permutation procedure in which entities are first sorted by increasing t or the statistic used for testing obtained on unpermuted data. Then, for each permutation, the test statistics obtained for the various entities in this permutation are artificially adjusted so that the following property holds: if entity i has a higher original test statistic than entity j, then entity i has a higher adjusted test statistic for this permutation than entity j. The overall corrected p-value for an entity is now defined as the fraction of permutations in which the adjusted test statistic exceeds the test statistic computed on the unpermuted data. Finally, an artificial adjustment is performed on the p-values so that an entity with a higher unpermuted test statistic has a lower p-value than an entity with a lower unpermuted test statistic; this adjustment simply increases the p-value of the latter entity, if necessary, to make it equal to the former. Though not explicitly stated, a similar adjustment is usually performed with all other algorithms described here as well.

Because of the permutations, the method is very slow.

D.5 Benjamini-Hochberg

This method [8] assumes independence of p-values across entities. However, Benjamini and Yekuteili showed that the technical condition under which the test holds is that of positive regression dependency on each test statistics corresponding the true null hypothesis. In particular, the condition is satisfied by positively correlated normally distributed one sided test statistics and their studentized t-tests. Furthermore, since up-regulation and down-regulation are about equally likely to occur, the property of FDR control can be extended to two-sided tests.

This procedure makes use of the ordered p-values $p_{(1)} \leq p_{(2)} \ldots \leq p_{(m)}$. Denote the corresponding null hypotheses by $H_{(1)}, H_{(2)}, \ldots, H_{(m)}$. For a desired FDR level q, the ordered p-value $p_{(i)}$ is compared to the critical value $q.\frac{i}{m}$. Let $k = \max_{p_{(i)} \leq q\frac{i}{m}} i$. Then reject $H_{(1)}, \ldots, H_{(k)}$, if such a k exists. Else, do not reject any hypothesis.

D.6 Benjamini-Yekutieli

For more general cases, in which positive dependency conditions do not apply, Benjamini and Yekutieli [9] showed that replacing q by $\frac{q}{m \sum_{i=1}^{m} (\frac{1}{i})}$ will provide control of the FDR. This control is typically applied in GO analysis, since the GO terms have both positive and negative regression dependency.

D.7 Storey's q-value

The stepup Benjamini-Hochberg procedure estimates the rejection region so that on average, FDR $< q$. Here q is the desired level of significance. Alternatively, Storey considers fixing the critical region and then estimating the FDR. This FDR for given cut-off is called the q-value.

p-value of 5% means 5% of all (null) tests result in false positives. q-value of 5% means 5% of all tests declared significant tests result in false positives (truly null). FDR = P(hypothesis i is truly null | test i is declared significant). The FDR is estimated taking into account the distribution

of p-values. Let π_0 be the prior probability that a hypothesis is true (null). It is assumed that p-values of null hypotheses are uniformly distributed on (0,1). Overall the p-value distribution is a mixture of a uniform (0,1) distribution with a proportion π_0 and a distribution concentrated over small values (close to 0) with a proportion $1 - \pi_0$. The proportion of tests in the significant results which come from the uniform distribution is the estimate of FDR. Storey's approach estimates this proportion using the bootstrap method. Other methods for estimating π_0 such as smoothing methods and mixture resolution methods are also current.

D.7.0.1 q-value Calculation

Let

- Let m be the total number of tests performed.
- Let π_0 be the probability that a hypothesis is null. This is estimated from the distribution of p-values.
- Let δ be the p-value below which a hypothesis is rejected.
- Let k be the number of entities having p-values less than δ.

Then the q-value is

$$q(\delta) = \frac{\pi_0 \times m \times \delta}{k}.$$

The denominator is the number of hypotheses rejected. The numerator is the (estimated) number of true null hypotheses ($m \times \pi_0$) multiplied by the chosen p-value (δ) giving the estimated number of true null hypotheses rejected.

Bibliography

[1] K.Akaike (1973): Information theory and an extension of the maximum likelihood principle. In B.N.Petrov and F.Cáski (Editors). *Second International Symposium on Information Theory*, pp. 267--281. Budapest: Akademiai Kaidó.

[2] S.Audic and J.-M.Claverie (1997): The significance of digital gene expression profiles. *Genome Research*, **7**, 986--995.

[3] T.L.Bailey and M.Gribskov (1998): Methods and statistics for combining motif match scores. *Journal of Computational Biology*, **5**, 211--221.

[4] T.L.Bailey and M.Gribskov (1998): Combining evidence using p-values: application to sequence homology searches. *Bioinformatics*, **14**, 48--54.

[5] A.Barski, S.Cuddapah, K.Cui, T-Y.Roh, D.E.Schones, Z.Wang, G.Wei, I.Chepelev, and K.Zhao (2007): High-Resolution Profiling of Histone Methylations in the Human Genome. *Cell* **129**, 823--837.

[6] T.Beissbarth, L.Hyde, G.K.Smyth, C.Job, W.M.Boon, S.-S.Tan, H.S.Scott, and T.P.Speed (2004): Statistical modeling of sequencing errors in SAGE libraries. *Bioinformatics*, **20** (suppl 1), 31--39.

[7] Gordon K. Smyth (2004): Linear Models and Empirical Bayes Methods for Assessing Differential Expression in Microarray Experiments.

[8] Y.Benjamini and Y.Hochberg (1995): Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society, Series B (Methodological)*, **57**, 289--300.

[9] Y.Benjamini and D.Yekutieli (2001): The control of the false discovery rate in multiple testing under dependency. *The Annals of Statistics*, **29**, 1165--1188.

[10] A.P.Dempster, N.M.Laird, and D.B.Rubin (1977): Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, **B 39**, 185--197.

[11] C.N.Dewey and B.Li (2009): Transcriptome analysis methods for RNA-Seq data *CAMDA 2009*, Chicago, IL, Oct 5-6, 2009. Biomedical Informatics Center, Northwestern University.

[12] Simon Anders and Wolfgang Huber (2010): Differential expression analysis for sequence count data, *Genome Biology*, **11:R106**.

[13] Mark D Robinson and Alicia Oshlack (2010): A scaling normalization method for differential expression analysis of RNA-seq data, *Genome Biology*, **11:R25**.

[14] S.Dudoit, H.Yang, M.J.Callow, and T.P.Speed (2000): Statistical Methods for identifying genes with differential expression in replicated cDNA experiments *Statistica Sinica*. **12**, 111--139.

[15] G.Z.Hertz and G.D.Stormo (1999): Identifying DNA and protein patterns with statistically significant alignments of multiple sequences. *Bioinformatics*, **15**, 563--577.

[16] F.M.Speed, R.R.Hocking, and O.P.Hackney (1978): Methods of Analysis of Linear Models with Unbalanced Data. *J. Am Stat Assoc*, **73**, 361, (105-112).

[17] R.G.Shaw, and T.M.Olds (1993): ANOVA for Unbalanced Data: An overview. *Ecology*, **74**, 6, (1638- 1645).

[18] J.E.Overall, and D.K.Spiegel (1969): Concerning least squares analysis of experimental data. *Psychological Bulletin*, **72**, 311--322.

[19] S.Glantz (2005): *Primer of Biostatistics*. Sixth Edition. New York:McGraw-Hill.

[20] H.Jiang and W.H.Wong (2009): Statistical inferences for isoform expression in RNA-Seq. *Bioinformatics*, **25**, 1026--1032.

[21] J.O.Korbel A.Abyzov, X.J.Mu, N.Carriero, P.Cayting, Z.Zhang, M.Snyder, and M.B.Gerstein (2009): PEMer: a computational framework with simulation-based error models for inferring genomic structural variants from massive paired-end sequencing data. *Genome Biology*, **10**, issue 2. oi:10.1186/gb-2009-10-2-r23.

[22] B.Li, V.Ruott, R.M.Stewart, J.A.Thomson, and C.N.Dewey (2009): RNA-Seq gene expression estimation with read mapping uncertainty. *Bioinformatics*, **26**, 493--500.

[23] L.Li (2009): GADEM: A genetic algorithm guided formation of spaced dyads coupled with an EM algorithm for motif discovery. *Journal of Computational Biology*, **16**, 317--329.

[24] A.Mortazavi, B.Williams, K.McCue, L.Schaeffer, and B.Wold (2008): Mapping and quantifying mammalian transcriptomes by RNA-seq. *Nature Methods*, **5**, 621-628.

[25] C.Olivier, F.Joluzel, and A.El Matouat (1999): Choice of the number of component clusters in mixture models by information criteria. *Vision Interface '99, Trois-Rivières, Canada*, 19-21 May 1999, 74--81.

[26] J.Rissanen (1986): Stochastic complexity and modeling. *The Annals of Statistics*, **14**, 1080--1100.

[27] G.Schwarz (1978): Estimating the dimension of a model. *The Annals of Statistics*, **6**, 461--464.

[28] R.Staden (1989): Methods for calculating the probabilities of finding patterns in sequences. *Computational and Applied Biosciences*, **5**, 89--96.

[29] J.D.Storey (2002): A direct approach to false discovery rates. *Journal of the Royal Statistical Society, Series B (Methodological)*, **64**, 479--498.

[30] J.D.Storey (2003): The positive false discovery rate: A Bayesian interpretation and the q-value. *The Annals of Statistics*, **31**, 2013--2035.

[31] P.Tiňo (2009): Basic properties and information theory of Audic-Claverie statistic for analyzing cDNA arrays: *Bioinformatics*, **10**:310 doi:10.1186/1471-2105-10-310.

[32] P.H.Westfall and S.S.Young (1993): *Resampling based multiple testing*. New York: John Wiley and Sons.

[33] X.Zhang, G.Robertson, M.Krzywinski, K.Ning, A.Droit, S.Jones, and R.Gottardo (2010): PICS: Probabilistic Inference for ChIP-seq, *Biometrics*, DOI: 10.1111/j.1541-0420.2010.01441.x

[34] Y.Zhang, T.Liu, C.A.Meyer, J.Eeckhoute, D.S.Johnson, B.E.Bernstein, C.Nusbaum, R.M.Myers, M.Brown, W.Li, and X.S.Liu (2008): Model-based Analysis of ChIP-Seq (MACS). *Genome Biology*, **9**, 2008, 9:R137 doi:10.1186/gb-2008-9-9-r137

[35] Albers, Cornelis A, Lunter, Gerton and MacArthur, Daniel G and McVean, Gilean and Ouwehand, Willem H and Durbin, Richard (2011): Dindel: accurate indel calls from short-read data, *Genome Research*, **21(6)**, 961-973.

[36] Mark A. DePristo, et. al. (2011): A framework for variation discovery and genotyping using next-generation DNA sequencing data, *Nature Genetics*, **43(5)**, 491-498.

[37] Homer, Nils and Nelson, Stanley F and others (2010): Improved variant discovery through local re-alignment of short-read next-generation sequencing data using SRMA, *Genome Biology*, **11(10)**, R99.

[38] Cabanski, Christopher and Cavin, Keary and Bizon, Chris and Wilkerson, Matthew and Parker, Joel and Wilhelmsen, Kirk and Perou, Charles and Marron, JS and Hayes, D (2012): ReQON: a Bioconductor package for recalibrating quality scores from next-generation sequencing data, *BMC bioinformatics*, **13(1)**, 221.

[39] DePristo, Mark A and Banks, Eric and Poplin, Ryan and Garimella, Kiran V and Maguire, Jared R and Hartl, Christopher and Philippakis, Anthony A and del Angel, Guillermo and Rivas, Manuel A and Hanna, Matt and others (2011): A framework for variation discovery and genotyping using next-generation DNA sequencing data, *Nature genetics*, **43(5)**, 491-498.

[40] Ruiqiang Li, Yingrui Li, Xiaodong Fang, et al. (2009): SNP detection for massively parallel whole-genome resequencing, *Genome Research*, **19(6)**, 1124-1132.

[41] Pang, Andy W and MacDonald, Jeffrey R and Pinto, Dalila and Wei, John and Rafiq, Muhammad A and Conrad, Donald F and Park, Hansoo and Hurles, Matthew E and Lee, Charles and Venter, J Craig and others (2010): Research Towards a comprehensive structural variation map of an individual human genome.

[42] Sebat, Jonathan and Lakshmi, B and Malhotra, Dheeraj and Troge, Jennifer and Lese-Martin, Christa and Walsh, Tom and Yamrom, Boris and Yoon, Seungtai and Krasnitz, Alex and Kendall, Jude and others (2007): Strong association of de novo copy number mutations with autism, *Science*, **316(5823)**, 445-449.

[43] Stefansson, Hreinn and Rujescu, Dan and Cichon, Sven and Pietiläinen, Olli PH and Ingason, Andres and Steinberg, Stacy and Fossdal, Ragnheidur and Sigurdsson, Engilbert and Sigmundsson, Thordur and Buizer-Voskamp, Jacobine E and others (2008): Large recurrent microdeletions associated with schizophrenia, *Nature*, **455(7210)**, 232-236.

[44] Rovelet-Lecrux, Anne and Hannequin, Didier and Raux, Gregory and Le Meur, Nathalie and Laquerrière, Annie and Vital, Anne and Dumanchin, Cécile and Feuillette, Sébastien and Brice, Alexis and Vercelletto, Martine and others (2005): APP locus duplication causes autosomal dominant early-onset Alzheimer disease with cerebral amyloid angiopathy, *Nature genetics*, **38(1)**, 24-26.

[45] Campbell, Peter J and Stephens, Philip J and Pleasance, Erin D and O'Meara, Sarah and Li, Heng and Santarius, Thomas and Stebbings, Lucy A and Leroy, Catherine and Edkins, Sarah and Hardy, Claire and others (2008): Identification of somatically acquired rearrangements in cancer using genome-wide massively parallel paired-end sequencing, *Nature genetics*, **40(6)**, 722-729.

[46] Duan, Junbo and Zhang, Ji-Gang and Deng, Hong-Wen and Wang, Yu-Ping (2013): Comparative studies of copy number variation detection methods for next-generation sequencing technologies, *PloS one*, **8(3)**, e59128.

[47] Boeva, Valentina and Zinovyev, Andrei and Bleakley, Kevin and Vert, Jean-Philippe and Janoueix-Lerosey, Isabelle and Delattre, Olivier and Barillot, Emmanuel (2011): Control-free calling of copy number alterations in deep-sequencing data using GC-content normalization, *Bioinformatics*, **27(2)**, 268-269.

[48] Alkan C, Kidd JM, Marques-Bonet T, Aksay G, Antonacci F, Hormozdiari F, Kitzman JO, Baker C, Malig M, Mutlu O, Sahinalp SC, Gibbs RA, Eichler EE (2009): Personalized copy number and segmental duplication maps using next-generation sequencing, *Nature Genetics*, **41(10)**, 1061-1067.

[49] Gusnanto A, Wood HM, Pawitan Y, Rabbitts P, and Berri S. (2012): Correcting for cancer genome size and tumour cell content enables better estimation of copy number alterations from next-generation sequence data, *Bioinformatics*, **28(1)**, 40-47.