BIOINFORMATICS ORIGINAL PAPER

Vol. 30 no. 24 2014, pages 3491–3498
doi:10.1093/bioinformatics/btu437

Sequence analysis

Advance Access publication July 15, 2014

Resolving complex tandem repeats with long reads

Ajay Ummat and Ali Bashir*

Department of Genetics and Genomic Science and Institute for Genomics and Multiscale Biology, Icahn School of Medicine at Mount Sinai, New York, NY, USA

Associate Editor: Michael Brudno

ABSTRACT

Motivation: Resolving tandemly repeated genomic sequences is a necessary step in improving our understanding of the human genome. Short tandem repeats (TRs), or microsatellites, are often used as molecular markers in genetics, and clinically, variation in microsatellites can lead to genetic disorders like Huntington's diseases. Accurately resolving repeats, and in particular TRs, remains a challenging task in genome alignment, assembly and variation calling. Though tools have been developed for detecting microsatellites in short-read sequencing data, these are limited in the size and types of events they can resolve. Single-molecule sequencing technologies may potentially resolve a broader spectrum of TRs given their increased length, but require new approaches given their significantly higher raw error profiles. However, due to inherent error profiles of the single-molecule technologies, these reads presents a unique challenge in terms of accurately identifying and estimating the TRs.

Results: Here we present PacmonSTR, a reference-based probabilistic approach, to identify the TR region and estimate the number of these TR elements in long DNA reads. We present a multistep approach that requires as input, a reference region and the reference TR element. Initially, the TR region is identified from the long DNA reads via a *3-stage modified Smith–Waterman* approach and then, expected number of TR elements is calculated using a *pair-Hidden Markov Models*–based method. Finally, TR-based genotype selection (or clustering: homozygous/heterozygous) is performed with *Gaussian mixture models*, using the *Akaike information criteria*, and coverage expectations.

Availability and implementation: https://github.com/alibashir/pacmonstr

Contact: ajayummat@gmail.com or ali.bashir@mssm.edu

Received on April 6, 2014; revised on June 29, 2014; accepted on July 2, 2014

1 INTRODUCTION

Tandem repeats (TRs) are contiguous regions of DNA in which the same (or highly similar) DNA sequences are repeated multiple times in order. TRs can range in size from single nucleotide repeats (homopolymers) to encompassing entire genes (Tyson *et al*., 2014). Short tandem repeats (TRs) of 1–6 bp, or microsatellites, are often used as molecular markers in genetics (Chen *et al*., 2003) or in forensic applications (Veselinović, 2006). Clinically, variation in microsatellites is observed in many genetic disorders. Triplicate CAG repeat expansions have been implicated in a number of neurological disorders, including exonic expansions in Huntington's disease (Walker, 2007) and spinocerebellar ataxia (Tsai *et al*., 2004) and non-coding expansions in fragile X syndrome (Jin, 2000). Microsatellite instability has also been used as a marker of cancer progression (Arzimanoglou *et al*., 1998). Additionally, microsatellites are implicated in regulatory roles; they have been shown to be highly conserved near transcription start sites (Sawaya *et al*., 2013) and are sometimes known to modulate gene expression (Sawaya *et al*., 2012). Despite their importance, TRs are often ignored because accurate resolution of TRs in both variation and assembly settings remains a challenging task.

With the availability of high-quality genomes, a number of tools were developed specifically for TR resolution (Lim *et al*., 2013). Tools like Tandem Repeat Finder (TRF; Benson *et al*., 1999) attempt to detect such repeats *de novo*. TRF has a detection phase that looks for statistically significant k-mer matches within sliding windows and an analysis phase, which determines period size, multiplicity and consensus repeats (Benson, 1999). Other methods have used clustering-based approaches (T-Reks) and local autocorrelation (TandemSwan), respectively, to attempt to identify more divergent and complex TRs [sometimes termed Fuzzy Tandem Repeats (Boeva *et al*., 2006; Jorda and Kajava, 2009]. This increased sensitivity comes at the cost of increased runtime and higher false-positive rates (Lim *et al*., 2013). Other methods, such as RepeatMasker, can then be used to screen entire genomes using known databases of TRs (and other repeat elements) identified by such *de novo* tools (Tarailo-Graovac and Chen, 2009).

The advent of low-cost sequencing over the past decade has made cataloguing genomic variation cost-effective across individuals. Robust tools have been created for short-read alignment (Langmead and Salzberg, 2012; Li and Durbin, 2009), single-nucleotide polymorphism (SNP) and indel detection (Garrison and Marth, 2012), as well as completely integrated end-to-end pipelines (McKenna *et al*., 2010). These methods have proven highly accurate in calling SNPs in diverse populations (Abecasis *et al*., 2012). Additionally, these technologies have been used to enumerate more complex copy number and copy neutral forms of structural variation (SV; Abyzov *et al*., 2011; Rausch *et al*., 2012; Sindi *et al*., 2009; Ye *et al*., 2009). Similarly, a number of excellent tools have become available for detecting microsatellites in short-read sequencing data (Doi *et al*., 2013; Gymrek *et al*., 2012; Highnam *et al*., 2013). Some of these methods rely on databases of known TRs (such as those generated by TRF across a genome sequence or a curated list of known variable TR intervals) and look at *spanning* reads (short reads which span

*To whom correspondence should be addressed.

© The Author 2014. Published by Oxford University Press. All rights reserved. For Permissions, please email: journals.permissions@oup.com

Downloaded from http://bioinformatics.oxfordjournals.org/ at :: on August 30, 2016

the repeat) to call TR alleles for a given individual (Gymrek *et al*., 2012; Highnam *et al*., 2013). Given the length of the reads being analyzed, these methods are limited in the multiplicity and/or period size of TRs that can be resolved even when they take into account more complex mate-pairing information. More recently, methods have arisen to detect such repeats *de novo* using short read data (Doi *et al*., 2013). Doi *et al*.'s study uses depth of coverage information on tagged TRs to estimate TR multiplicity in repeats longer than the size of a single short read.

In the past few years, 'third-generation' sequencing technologies have arisen that offer substantially improved read lengths. Single-molecule real-time (SMRT) sequencing from Pacific Biosciences (PacBio) creates continuous sequence reads ranging from several kb to tens of kb in length. These reads have been shown to allow dramatic improvements in genome assembly (Bashir *et al*., 2012; Chin *et al*., 2013; Koren *et al*., 2012; Ribeiro *et al*., 2012), haplotype phasing (Lo *et al*., 2011) and SV applications (Ritz *et al*., 2010). Although they are obvious candidates for broadening the scope of detectable TRs, the unique error profile (and higher raw error-rate) (Eid *et al*., 2009) necessitate new approaches for accurate resolution of TR multiplicities.

Here, we present PacmonSTR, a tool for the detection and resolution of TRs specifically optimized for raw single-molecule sequencing data. PacmonSTR attempts to build on the ideas of previous reference-based approaches to best capture the idiosyncratic features of single-molecule sequencing data. The method provides the following:

(1) Improved boundaries of spanning TRs by performing a more sensitive TR-specific dynamic programming alignment
(2) Statistical estimates of TR multiplicity using a pair-hidden Markov models (pairHMM)
(3) Predictions of heterozygosity and homozygosity with consensus sequences when multiple spanning reads exist
(4) Predictions of boundaries for compound structural variants within or around a TR interval

Using simulated and real sequencing data compared with the hg19 build of the human genome, we compare the performance of our methods and *de novo* approaches in terms of detecting TRs and estimating multiplicity. Clustering performance is assessed using known homozygous and synthetic heterozygous TR sequences. Lastly, we simulate the introduction of SV into TR intervals to show the ability of the tool to flag non-TR intervals, and apply the SV detection approach on real data.

2 METHODS

An overview of the methods is shown in Supplementary Figure S1. We begin with alignment of a set of reads $Q = \{Q_k | k = \text{number of reads}\}$ to the reference genome G (*hg*19), using Blasr (Chaisson and Tesler, 2012). Uniquely mapped reads spanning TR intervals on the reference are selected. In the scenario that two disparate alignments are found spanning a reference TR interval (which may occur if a TR allele is highly divergent from the reference allele at that locus), multiple unique alignments may be merged to provide the initial query seed interval for downstream processing. Next we use a modified Needleman–Wunsch alignment algorithm to better identify TR boundaries and give initial TR multiplicity estimates (Section 2.1). The identified TR interval is then passed through a pairHMM to give a more rigorous estimate of the TR multiplicity (Sections 2.2 and 2.3). Reads are then grouped by locus and clustered based on their estimated TR multiplicities to determine zygosity (Section 2.4). Lastly, the probability values calculated by the forward algorithm of the pairHMM forms a basis to discover potential SVs internal to TRs (Section 2.5).

2.1 Identification of TR boundaries within reads

Let the interval $[\alpha, \beta)$ denote the occurrence of a TR in G and r denote the consensus sequence for the TR element. The initial alignment information is used to extract a subinterval from the query, Q_k, containing the TR as well as a predetermined amount of flanking reference sequence, ϵ. Given that the TR in Q_k may deviate from the expected reference size, it is likely that the initial reference alignment inaccurately represents the boundaries of the TR. To address this, we extract subinterval $[a - e, b + e)$ in Q_k corresponding to the reference coordinates $[\alpha - \epsilon, \beta + \epsilon)$, where a, b are query positions in the alignment that match reference position α, β. The spanning query interval $Q_k[a - e, b + e)$, along with r and the set of prefix and suffix sequences in G (corresponding to the intervals $[\alpha - \epsilon, \alpha)$ and $[\beta, \beta + \epsilon)$, respectively), are passed into a TR-specific dynamic programming algorithm to improve boundary resolution, as shown in Figure 1.

For a given query subinterval $Q_k[a - e, b + e)$, let p, r and s correspond to the prefix, TR element and suffix reference sequences and P, R and S correspond to their respective alignment matrices to $Q_k[a - e, b + e)$. The prefix alignment matrix P is filled out according to the Needleman–Wunsch algorithm (Needleman and Wunsch, 1970). The repeat matrix R, however, requires additional initialization and recurrence conditions:

$$R_{i,j} = \max \begin{cases} P_{i-1,|p|} + m \\ R_{i-1,j-1} + m \\ R_{i-1,j} + \rho \\ R_{i,j-1} + \rho \\ R_{i-1,|r|} + (j-1)\rho + m \end{cases}$$

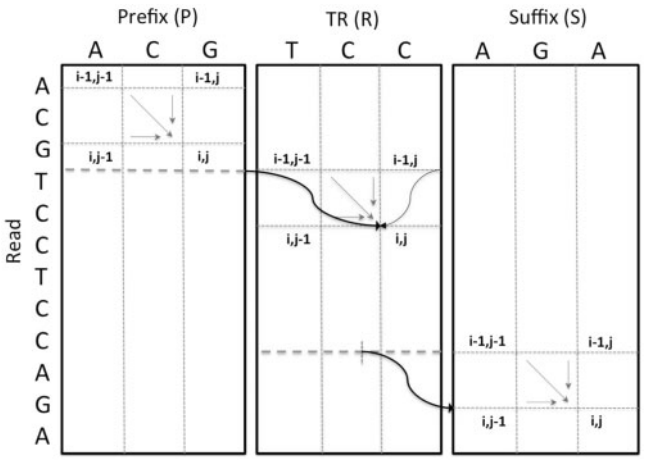

Fig. 1. Representation of the dynamic programming algorithm for unbiased identification of the TR boundaries within reads. Here, 'Read' corresponds to the query subinterval $Q_k[a - e, b + e)$ and sequence 'TCC' corresponds to the reference TR element. The thick dashed gray lines in the Prefix (P) and TR (R) matrices represent the resolved TR boundaries within the read. The arrows represent the recurrence conditions for the three matrices

Downloaded from http://bioinformatics.oxfordjournals.org/ at :: on August 30, 2016

where $i \in Q_k[\mathrm{a}-e, \mathrm{b}+e)$, $j \leq |r|$, ρ is used to denote a gap penalty, and $m[Q_k^i, r_j]$ is the score for aligning the corresponding base i in $Q_k[\mathrm{a}-e, \mathrm{b}+e)$ and j in r.

First, any position i in $Q_k[\mathrm{a}-e, \mathrm{b}+e)$, denoted by Q_k^i, which reaches the end of the prefix sequence is a valid entry point into the repeat matrix R at row $i+1$. We cannot assume a priori that the repeat starts at the beginning of the predicted consensus copy, and thus, at any position $R_{i,j}$ we must consider $P_{i-1,|p|} + m[Q_k^i, r_j]$. The key distinguishing feature of the repeat matrix is its ability to traverse a repeat unit multiple times. This is captured at position $i < |Q_k[\mathrm{a}-e, b+e)| = |Q_k|$, by allowing the move from position $(i-1, |r|)$ in R to any position (i, j) in the following row by $R_{i-1,j} + (j-1)\rho + m[Q_k^i, r_j]$ (the term penalizes any insertion gaps induced by such a move). This allows unbiased expansion and contraction of the repeat and substantially reduces the time complexity, especially in the case of highly expanded repeats (the most challenging types of TRs to resolve). Under the assumption that no sequence was lost at the junction, the suffix is straightforward to compute. A single special case is required in the first column of the suffix matrix S to consider entries from the repeat matrix R. This is handled by the relation $S_{i,1} = \max_{0<k<|r|} R_{i-1,k} + m[q_i, s_1]$, otherwise the standard Needleman–Wunsch recurrence is used.

After retracing the optimal path through the S, R and P matrices, in order, and returning the entry and exit points in R, the predicted TR sequence in $Q_k[\mathrm{a}-e, \mathrm{b}+e)$ is obtained. The multiplicity of the repeat is given directly by this traversal; it is roughly the number of times the last case of the repeat recurrence is used in the optimal traversal path, plus the fractions of the repeat from the prefix entry and suffix exit points. The time complexity of this algorithm is $O(|Q_k||p|) + O(|Q_k||s|) + O(|Q_k||r|)$. However, the prefix and suffix lengths are constrained by the fixed flanking criteria, ϵ, leading to time complexity $O(|Q_k|\epsilon) + O(|Q_k||r|)$. In practice, as $\epsilon \ll |Q_k|$ and $r \ll |Q_k|$, this compute is substantially faster than $|Q_k|^2$.

2.2 Probabilistic TR resolution through pairHMM

To improve estimations and allow for allele assignment, we extend the previous strategy in two key ways:

(1) Explicit modeling of error modes specific to the sequencer, such as cognate sampling (Eid *et al.*, 2009)

(2) Provide probabilities for predicted TR multiplicities in a given read.

To address this we propose a pairHMM approach that computes the probability for the sum of all alignment paths between the query and a putative repetitive TR sequence. In this paradigm, the number of TR elements is treated as a random variable and the pairHMM is used to calculate the expected value of this random variable based on an estimated discrete probability mass function. Supplementary Figure S2 details the model structure of the pairHMM, which takes as input the predicted TR interval in $Q_k[\mathrm{a}-e, \mathrm{b}+e)$ (labeled as q) and the consensus TR element sequence, r, and models the error modes with three hidden states: M = match, X = deletion and Y = insertion.

Transition and emission probabilities are generated from Blasr (Chaisson and Tesler, 2012) alignments in non-TR regions. These probabilities are computed globally over all sequenced reads and locally using the flanking sequence to account for variability between reads. In practice, global parameters for transitions are seen in Supplementary Figure S3A. To more accurately represent the known issue of 'cognate sampling' (Eid *et al.*, 2009), whereby the emitted base has a dependence on the last incorporate nucleotide, we incorporate conditional dependence in emission probabilities when calculating forward probability values (Supplementary Methods) for the insertion state (defined as X). Parameters for these conditional emission probabilities for the insertion state X are shown in Supplementary Figure S3B. Our application differs from the canonical pairHMM in that we are not comparing two true sequences, but rather comparing an error-prone sequence (the long read) to the *potential* true sequence derived from a reference TR element (r). Therefore, we use a slightly modified version of the pairHMM presented by Durbin *et al.*, to allow for an insertion of a base to be followed immediately by a deletion (or vice versa) to account for this type of sequencing pathology.

2.3 Calculating expected TR multiplicities

Let $P(\vec{O}\,|\lambda)$ correspond to the probability of an observation sequence $\vec{O}$ given the pairHMM model parameters λ. We run the forward algorithm (Durbin *et al.*, 1998; Rabiner, 1989) with q and an upper bounded synthetic TR sequence, Ω (constructed by repeating r), as the pair of sequences constituting the observation sequence $\vec{O} = (q, \Omega)$. In practice, we use $|\Omega| = 1.5|q|/|r|$. For all $j = 1, \ldots, |\Omega|$, we calculate the sum of forward probabilities $f_S(|q|, j)$ from each state $S = M, X, Y$ to obtain $P(\vec{O}\,|\lambda, j) = f_M(|q|, j) + f_X(|q|, j) + f_Y(|q|, j)$. Details on the forward calculation can be found in Supplementary Methods. These probability values are used to get relative normalized weights, $w_j = P(\vec{O}\,|\lambda, j) / \sum_{k=1}^{|\Omega|} P(\vec{O}\,|\lambda, k)$ of observing a given TR multiplicity j in the observation sequence $\vec{O}$. The relative weights are then used to calculate the expected value of the TR multiplicity, $numTR = \sum j w_j$ (Supplementary Figure S4).

We also calculate the log odds ratio with respect to a random model, Z (Durbin *et al.*, 1998). Let $j_i^* = \mathrm{argmax}_{\forall j} P(\vec{O}_i|\lambda, j)$ represent the TR multiplicity at which $P(\vec{O}_i|\lambda, j)$ is maximal. The log odds ratio is calculated as $\log\left(\max_{\forall j} P(\vec{O}_i|\lambda, j) / P(\vec{O}_i|Z, j_i^*)\right)$. The higher the value of the log odds ratio, higher the significance of the pairwise alignment between the query sequence, q, and the constructed TR sequence, Ω.

Optimization of the model parameter λ, given the observation sequence $\vec{O}$ is typically performed using Baum–Welch or expectation-maximization algorithms (Durbin *et al.*, 1998). In our construct, the true observation sequence $\vec{O}$ is unknown and hence we estimate the model parameters λ from the known pairwise alignments. We calculate $P(\vec{O}|\lambda)$ where λ is either λ_{Local} or λ_{Global}. λ_{Global} is pre-estimated from sequence alignments of the reads to the reference genome, G, and λ_{Local} is estimated at runtime from the sequence alignments upstream and downstream of q (from the prefix P and suffix S matrices, as previously defined). The model parameter λ is then selected based on the higher value for $P(\vec{O}|\lambda)$ for $\lambda = (\lambda_{Local}, \lambda_{Global})$ or $\lambda = argmax\{P(\vec{O}|\lambda_{Local}), P(\vec{O}|\lambda_{Global})\}$.

2.4 TR allele calling

To determine zygosity for each TR event, we use a generalized Gaussian mixture models (GMMs)-based approach, which uses Akaike information criterion (AIC) for the initial model selection. Algorithm 1 details the process by which heterozygous or homozygous calls are made and consensus sequences are generated for each cluster.

Algorithm 1 TrAlleles (D, t_C, t_b)

```
R1 = Gmm (D, 1)   # run GMM with 1 component
R2 = Gmm (D, 2)   # run GMM with 2 components
while R2[2] < tR do
    D = R2[1]   # assume smaller component is outlier, set D to larger component
    R1 = Gmm (D, 1)
    R2 = Gmm (D, 2)
end while
if Aic(R2) < Aic(R1) ∧ Csep (σ1, σ2) < tC ∧ BinCdf (|D|, |R2[1]|) > tb then
    return POAhb(R2[1]), POAhb(R2[1])
else
    return POAhb(D)
end if
```

Algorithm 1 TrAlleles. Takes as input the set of reads spanning a TR event and returns the consensus homozygous or heterozygous allele(s)

We assume the observed TR multiplicity calls are normally distributed around the true TR length for each allele. We use the *scikit-learn* GMM

Downloaded from http://bioinformatics.oxfordjournals.org/ at :: on August 30, 2016

implementation (Pedregosa *et al.*, 2011), with the covariance type for the dataset to be diagonal (given that the one-dimensional data points have no correlations). For diploid genomes, we allow either 1 or 2 components in the model, corresponding to homozygous or heterozygous alleles at each locus. Note that outliers are inevitable in such analyses and to handle such outliers we require that each component in the mixture model contain at least two data points. If a component represents a singleton value, we eliminate the value from the initial set and iterate until a component with at least two data points is observed. For initial model selection, we compare the AIC (Akaike, 1978) for $n = 1$ and $n = 2$ components, and choose the one with a lower value (Burnham, 2004).

To call the event as heterozygous, we intersect the model selection with two different sets of criterions. The first criterion calculates the *c*-separation, C_{SEP}, between the means of the two predicted Gaussian distributions (Dasgupta, 1999), where c is a constant scaling factor. Let $C_{SEP}(\sigma_1, \sigma_2) = c * \max(\sigma_1, \sigma_2)$ where σ_1, σ_2 corresponds to the standard deviation of clusters 1 and 2, respectively. We set a threshold $t_C = |\mu_1 - \mu_2|$, where μ_1, μ_2 corresponds to cluster means. For the second criterion, we confirm whether the number of data points per cluster is possible under the assumption that the observed split of data should follow a binomial distribution. We approximate this by calculating BINCDF, 1 - binomial CDF of observing at least the number of reads in the larger component given total clustered reads, and enforce that this is above some cutoff, t_b.

If $C_{SEP}(\sigma_1, \sigma_2) < t_c$ and the probability of observing the number of predicted data points per clusters is more than t_b, the event is labeled as heterozygous. The parameters '*min_covar*' (the lower bound for the covariance values of the mixture model in *scikit-learn's* GMM implementation) and c are optimized using a homozygous and heterozygous training set (See Section 3). In the final step, a consensus sequence is generated for each cluster via 'partial order alignment', using the heaviest bundle algorithm (Lee, 2003). These consensus sequences are then used to reestimate the number of TRs via pairHMM for each allele.

2.5 SV detection and multiplicity lower bounds

The sequence in the query TR interval may not simply represent TR expansions or contractions. SVs have been shown to flank many repeat-mediated structural variants including TRs (Yalcin *et al.*, 2011). The $P(\vec{O}|\lambda)$ derived from the forward algorithm provides a probabilistic basis to estimate and classify non-repeat intervals in q. These non-repeat intervals (defined as *region*SV) could either be repeat sequences with some similarity with the reference TR element or internal SV.

Using q as the input we follow a simple procedure to identify these non-repeat intervals. For each index i of q, we calculate $\max_{\forall j} P(\vec{O_i}|\lambda, j)$ and j_i^* (refer section 2.3), where $\vec{O_i} = (q_{(1,2,\ldots,i)}, \Omega)$ is the observation sequence. Next, we calculate log-likelihood ratio, $LLR(i, j_i^*) = P(\vec{O_i}|\lambda, j_i^*) - P(\vec{O_i}|Z, j_i^*)$, for each index i of q. We then calculate the deviation of LLR for each i, defined as $\Delta LLR_{i,i-1} = LLR(i, j_i^*) - LLR(i-1, j_{i-1}^*)$. For a perfect TR sequence between $[i', i]$, $\Delta LLR_{i,i'}$ will be >0; SVs in q can then be identified as follows:

$$region\text{SV} = \{[i_h, i_{\bar{h}}] : \ \Delta LLR_{i_{\bar{h}}, i_h} < 0 \ | 0 \le i_h < i_{\bar{h}} \le |q|\}$$

The likely repeat region in q is then a set $\{(1, |q|) - region\text{SV}\}$. Thus, $region\text{SV} \subseteq q$ may refine our estimate of the expected TR multiplicity as $numTR - |region\text{SV}|$. This correction presents a more conservative bound to the expected TR multiplicity in q. A high value for $|region\text{SV}|$ suggests a significant insertion region in q with respect to the reference TR sequence. In the current implementation we have restricted ourselves to evaluating small SV insertions; a simple two-state HMM with Gaussian emissions was used to flag predicted insertion intervals within the query by looking at the ΔLLR values at each adjacent $[i', i]$, of q. The GaussianHMM implementation from *scikit-learn* (Pedregosa *et al.*) was run with two components to call SV events in q. The mean and covariance values for the model were initialized based on the assumption that the inserted intervals would have $\Delta LLR < 0$.

3 RESULTS

3.1 Raw read TR prediction analysis

To evaluate TR detection methods we created simulated reads on chromosome 1 of hg19 using pbsim (Ono *et al.*, 2013) with 'datatype' as CLR, 'depth' as 10, 'length-mean' as 7000 and using the 'model-qc' option. We focused on detection (Does the method identify a TR in a TR spanning read?) and accuracy (How consistent is the predicted TR multiplicity with the known value)? Detection is only relevant for *de novo* TR predictions; reference-based methods (such as PacmonSTR) depend on the reference as input. TRF was run on all raw read sequences using a minimum score setting of 25 (roughly twice as sensitive as default settings, leading to many false-positive calls). Figure 2A indicates the ability of TRF to detect a consensus sequence similar to the true repeat consensus.

To provide a conservative comparison, we tried to be as permissive as possible in our definition of 'detection'. Each predicted consensus repeat from TRF was compared with the known TR, by performing Smith–Waterman alignment of the predicted repeat against a 2× TR of the true consensus. This ensured that any cyclic variations of the true repeat would be captured. Using a match score of 5, mismatch of –5 and indel penalty of –5, a true repeat was considered 'detected' if any of the alignment scores to TRF repeat predictions in the interval exceeded $0.35L (= 0.8 * 5 * L - 0.1 * 5 * L)$, where L is the length of the true TR consensus. In general, the higher the multiplicity and the shorter the repeat interval the more likely TRF was able to accurately detect the element. The multiplicity result is straightforward: the more instances of a TR, the more likely the method is able to capture it. Shorter repeats are likely easier to detect owing to error rates—the longer repeats are often too highly diverged to satisfy TRF's k-tuple requirements unless extremely high copy numbers are observed.

Reads that had detectable events by TRF were used to assess the accuracy of TR multiplicity predictions. For TRF, the best candidate repeat (closest to the true consensus) was used for multiplicity estimation. Figure 2B shows the performance of TRF versus our modified recurrence ('Naïve DP') and pairHMM algorithms over all simulated reads of hg19 chromosome 1. As expected, TRF underpredicts missing TR instances or over-fragmenting a continuous TR in multiple disjoint elements. At all frequencies and period lengths the pairHMM accurately predicts TR multiplicity, outperforming both the naïve dynamic program and the TRF. The pairHMM and naïve dynamic programming converge at large period sizes (>16 bp), as it is unlikely that slight indels or substitutions would disrupt the optimal path of the query through a predicted repeat instance.

3.2 Evalutation of clustering performance

3.2.1 Dataset construction To assess the performance of the clustering approach, we used a roughly publicly available sequence reads (roughly 9× coverage) from the CHM1hTERT cell line (Pacific Biosciences, 2013), a haploid complete

Downloaded from http://bioinformatics.oxfordjournals.org/ at :: on August 30, 2016

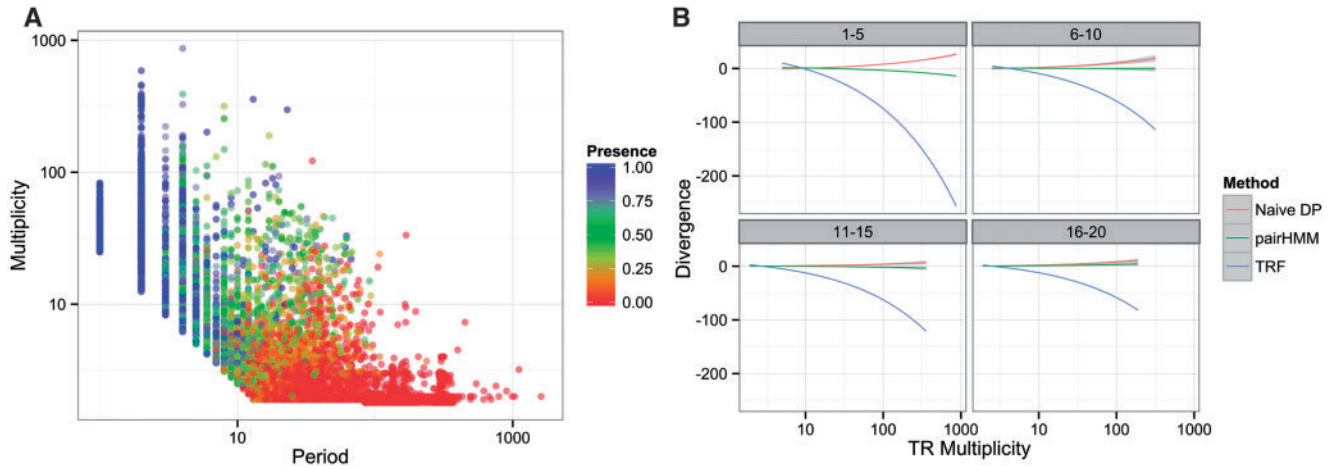

Fig. 2. (**A**) TR detectability of TRF shown by 'Presence' value (1.00 = TRF detects TR element in all reads spanning TR event and 0.0 = TRF detects no TR elements in any read spanning TR event). Both the axes are in $\log_{10}$ for clarity. (**B**) Accuracy of the multiplicity prediction by different methods. The TR Multiplicity axis is in $\log_{10}$ for clarity

hydatiform mole (CHM) (Jacobs *et al.*, 1980). All TR zygosities for this genome should therefore be homozygous.

Evaluating heterozygous calls is more challenging, as it is difficult to establish 'ground truth' for individual reads derived from a diploid reference. To address this ambiguity we constructed a synthetic heterozygous test set using the same haploid CHM dataset. First, we identified disparate regions in the reference genome with identical consensus TR sequences. Next, for each group of regions with a shared TR sequence we performed a pairwise merger of the constituent read sets. Each disparate pair yielded a synthetic heterozygous event. These events were then grouped by the difference in their TR multiplicities. We then evaluated the performance of the clustering approach for the TR multiplicity differences of 1, 3, 5 and 8.

To select the parameters for our clustering approach, we trained our model on a training set and selected the model parameters with the highest accuracy score on the homozygous and synthetic heterozygous test sets. '*min_covar*' of 0.0175 and t_C of 2.0 were used for clustering of CHM data.

3.2.2 Clustering performance PacmonSTR's ability to call homozygous TRs is shown in Figure 3A. The approach correctly predicts >90% of events at medium to large period sizes. Small periods were more difficult to accurately resolve, as small variances in size can lead to large deviations in predicted multiplicity (especially for singletons or small repeats that contain homopolymeric sequences).

3.3 Small insertion detection in TRs

As expected, heterozygous calling performance improves as the difference in the TR multiplicity increases (Fig. 3B). However, even at a multiplicity difference of 3 the method is still able to identify >75% of regions accurately. As in the case of homozygous events, single base TR elements were problematic at all difference levels. For heterozygous events we also evaluated the accuracy of assigning reads to a given cluster. Figure 3C shows the fraction of reads that are misassigned based on the period

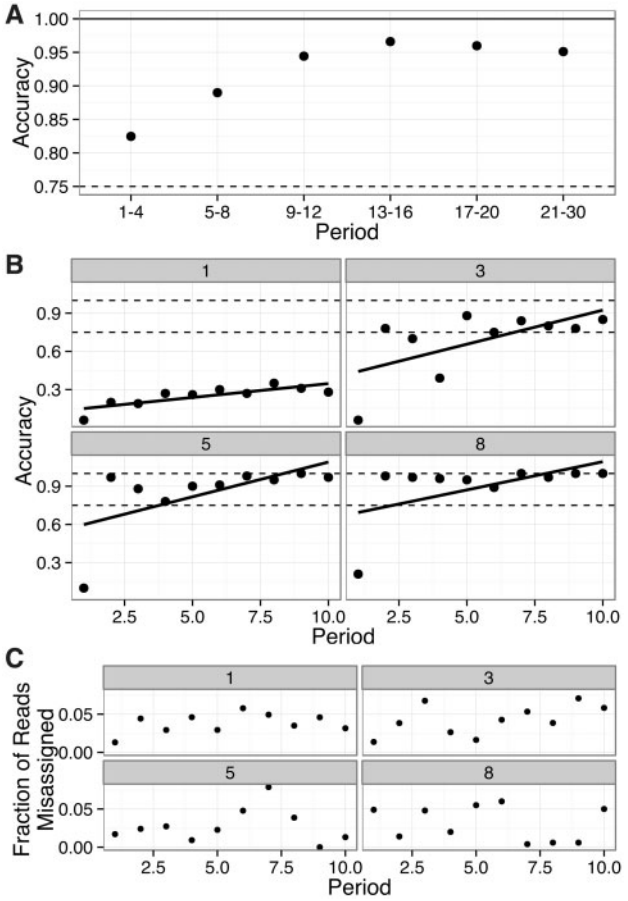

Fig. 3. (**A**) Clustering Performance for a homozygous test set from CHM1hTERT cell line. (**B**) Clustering performance for a constructed heterozygous test case. The horizontal dashed line represents the accuracy of 1 and 0.75. (**C**) Fraction of reads that are misassigned from the heterozygous calls. For all the figures the x-axis is the period of the TR event, (B) and (C) are faceted by the difference in predicted cluster means of the homozygous test cases

Downloaded from http://bioinformatics.oxfordjournals.org/ at :: on August 30, 2016

size and TR multiplicity differences. In general, clustering is extremely accurate except for small repeats or low differences in the TR multiplicities.

To evaluate the pairHMM's ability to identify insertions (or structural variations, SV) in the TR interval, we simulated random sequences and added those to the simulated repeats from chromosome 1 of the CHM cell line. Fifty replicates were performed at each period, frequency and SV size bin. We then assessed the ability of the method to accurately call the insertion and estimate its boundaries in two different cases: (i) insertions internal to TR regions (Fig. 4A and B) and (ii) insertions at the ends of TR regions (Supplementary Figure S5A and B).

Figure 4A and B highlights the predicted number of events under different period, multiplicities and insertion sizes. As expected, the smallest insertion (10 bp) was consistently the hardest event to detect. Notably, larger SV sizes are rarely missed and no over-calling occurs at any event size (Fig. 4A). To assess the accuracy of the boundaries, we examined reads that predicted a single insertion; Figure 4B shows that the algorithm is roughly centered at 0 bp divergence between the expected and observed start sites. Insertion sizes do not seem to substantially change the precision of boundary predictions. Though smaller periods give the best results, the boundary predictions are surprisingly tight across simulated values.

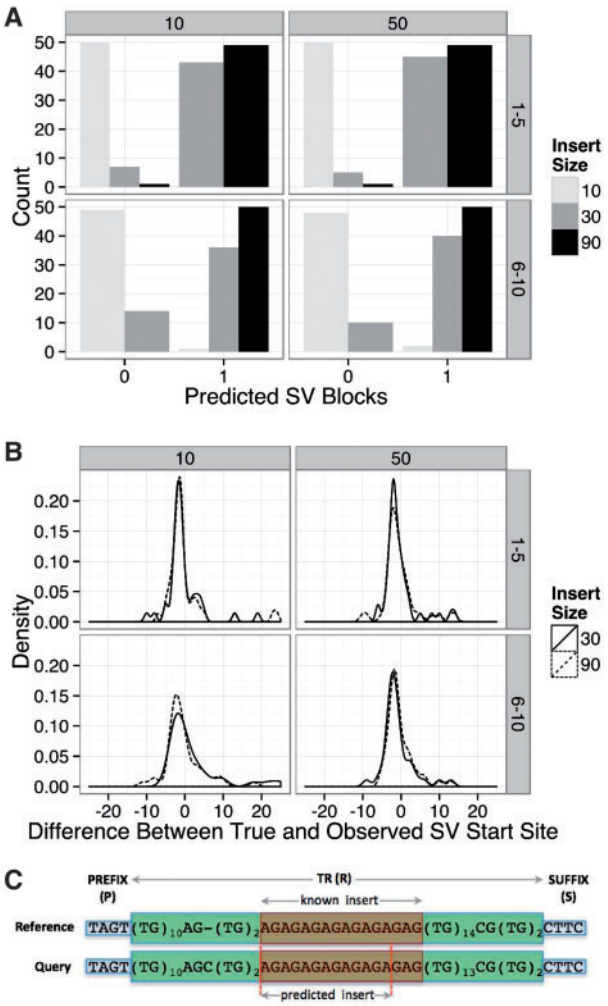

Fig. 4. (**A**) Detection count. The x-axis is the predicted states from HMM and y-axis is the number of counts. (**B**) Density plot of the accuracy of insert boundaries when the insertion is in the middle of the TR region as predicted by the model. The x-axis is the difference between the true and predicted insertion start site in the TR region. Insert sizes of 10, 30 and 90 base pairs are plotted. The figure is facetted by TR multiplicity and TR period sizes. (**C**) Example insertion in CHM consensus 'TG' repeat. Comparison with hg19, chr1:152779513 -152779589, verifies an insertion internal to event

Using these parameters, we probed for insertions within CHM chromosome 1. Nine regions on chromosome 1 showed up as positive for insertions. Figure 4C highlights an example insertion that we were able to confirm via comparison with hg19. The consensus repeat 'TG' is split by an internal 'AG' repeat; this is detected without knowledge of the full reference sequence (only the 'TG' element). The observed consensus differs from hg19 in 'TG' multiplicity downstream of the event insertion site.

4 DISCUSSION

We have presented an approach for reference-based TR detection that is specifically adapted for long reads with high insertion/deletion error rates. The method substantially outperforms *de novo* approaches both in detection and resolution of events. Additionally, the probabilistic pairHMM not only provides an even tighter prediction of TR multiplicities on raw reads it can also be used to infer structural variants within, or immediately flanking, a TR region. Despite meeting these goals, PacmonSTR and the simulations used to evaluate it have a number of limitations and areas where further improvements can be made.

Validation of clustering accuracy is an issue that may need to be addressed on a genome by genome (or repeat by repeat) level (Abecasis *et al.*, 2012). We attempted to model this by using real haploid genome sequences and creating synthetic heterozygous TRs from disparate genomic regions of the genome. In real diploid data, however, one does not know if a region is truly heterozygous. An alternative approach for evaluating calls is to identify SNVs, indels and SVs upstream and downstream of a called TR. Once TR multiplicities are predicted for each read, and clustering assignment has occurred, one can evaluate concordance of flanking variant alleles with the TR clustering assignment of raw reads to determine whether they form a consistent haplotype. The high-error profile of SMRT sequencing on a per-read basis could make accurate SNP assignment challenging. However, as the read lengths continue to expand, it has been shown that multiple SNPs and SVs captured within a single long read could create a stronger statistical signal to validate phasing (Lo *et al.*, 2011). Ideally, this haplotype information can be incorporated into the clustering process, in addition to the estimated TR multiplicity, to more robustly separate TR alleles with small periods or with small multiplicity differences.

Given the quadratic runtime for the pairHMM, if read lengths continue to improve into the hundreds of thousands of base pairs, the runtime of the algorithm may become unwieldy in real-world scenarios. However, the pairHMM is not required to accurately resolve multiplicity for most TRs. For large-

Downloaded from http://bioinformatics.oxfordjournals.org/ at :: on August 30, 2016

period TRs, the modified prefix-repeat-suffix dynamic program provides nearly identical results at substantially faster speeds. Although the DP runtime grows linearly as the period size increases (and may also become prohibitively large), faster banded heuristics become viable with large periods. Many small-period TRs can be resolved in the clustering phase without requiring the pairHMM. If two alleles are highly dissimilar, even if read-to-read variation is high, it is likely that they still will form distinct clusters. Once clusters are identified and consensus sequences are generated, TRs can be identified directly from the consensus. Thus, the full pairHMM can be restricted for scenarios where accurate resolution is required on short period repeats for which clear clustering cannot be performed.

The proposed and related approaches are reliant on predetermined locations of TRs in the reference as input. This limits the generalizability of the method, but can be alleviated by preprocessing the data with other *de novo* and structural variant tools. Figure 2A showed that TRF was often able to detect repeats from the raw reads. In practice, the algorithm can be run on all raw data to seed potential repetitive positions in the queries. These positions can be projected back onto the reference to identify candidate intervals for novel TR events. Reads spanning these positions of interest can then be interrogated by running the full pairHMM on the set of TRs predicted in the interval. As multiple consensus repeats will likely be identified, the one yielding highest probability can be selected for clustering analysis. Additionally, some SV tools, like Delly, allow for detection of novel TRs using reference coordinates that can similarly be passed into the method (Rausch *et al*., 2012).

Just as we can inform *de novo* TR calling by integrating other tools, we can similarly use our approach to help inform compound structural events involving TRs. In Figure 4 we demonstrated the method's ability to reliably identify inserted sequence within a TR element. SV elements observed on chromosome 1 were often TR expansions (such as in Fig. 4C); these sometimes existed at TR boundaries, suggesting potential refinements may be needed to the initial DP algorithm to better distinguish TR intervals from prefix and suffix sequences. Additionally, SVs may not always represent a distinct sequence block, the predicted consensus periods for a TR within an individual have been shown to diverge from the expected TR consensus, especially in the case of compound TRs where smaller repetitive units are grouped within a larger repeating unit (Doi *et al*., 2013). Although these can be more easily seen once a consensus is generated, local fluctuations in pairHMM probabilities can inform when such slight deviations in TR consensus are occurring periodically across a TR interval.

As alternative sequencing platforms become available, algorithms are needed that are able to harness the unique features of each technology. For any technology, even as the cost per base decreases, one must always make a choice between higher sequencing depth and larger numbers of samples. This decision is tied to the experimental goals of the project as well as the relative precision of a technology at different sequencing depths. In the context of TRs, our method provides informative guidelines for TR resolution given the type and size of the desired repeat. By allowing single read calls (and providing a probabilistic framework for evaluating these calls), the method is immediately applicable even at low depth, making it a useful option given the platform's comparatively higher cost, while also enabling potential applications in heterogeneous cell populations, as observed in cancer. Integrating raw read TR resolution with robust consensus calling, will continue to make the approach applicable for the type of high-depth studies that will soon be feasible as these technologies mature and become used by the broader research community.

ACKNOWLEDGEMENTS

We thank Sean Whalen and Matthew Pendleton for helpful discussions on the algorithmic approach and implementation. This work was also supported in part through the computational resources and staff expertise provided by the Department of Scientific Computing at the Icahn School of Medicine at Mount Sinai.

Funding: This research was funded by the Icahn School of Medicine at Mount Sinai through seed funding to A.B.

Conflict of Interest: A.B. has previously been employed at Pacific Biosciences (2009-2011).

REFERENCES

Abecasis,G.R. *et al.* (2012) An integrated map of genetic variation from 1,092 human genomes. *Nature*, **491**, 56–65.

Abyzov,A. *et al.* (2011) CNVnator: an approach to discover, genotype, and characterize typical and atypical CNVs from family and population genome sequencing. *Genome Res.*, **21**, 974–984.

Akaike,H. (1978) A Bayesian analysis of the minimum AIC procedure. *Ann. Inst. Stat. Math.*, **30**, 9–14.

Arzimanoglou,I.I. *et al.* (1998) Microsatellite instability in human solid tumors. *Cancer*, **82**, 1808–1820.

Bashir,A. *et al.* (2012) A hybrid approach for the automated finishing of bacterial genomes. *Nat. Biotechnol.*, **30**, 701–707.

Benson,G. (1999) Tandem repeats finder: a program to analyze DNA sequences. *Nucleic Acids Res.*, **27**, 573–580.

Boeva,V. *et al.* (2006) Short fuzzy tandem repeats in genomic sequences, identification, and possible role in regulation of gene expression. *Bioinformatics*, **22**, 676–684.

Burnham,K.P. (2004) Multimodel inference: understanding AIC and BIC in model selection. *Sociol. Methods Res.*, **33**, 261–304.

Chaisson,M.J. and Tesler,G. (2012) Mapping single molecule sequencing reads using Basic Local Alignment with Successive Refinement (BLASR): theory and application. *BMC Bioinformatics*, **13**, 238.

Chen,S. *et al.* (2003) Distribution and characterization of over 1000 T-DNA tags in rice genome. *Plant J.*, **36**, 105–113.

Chin,C.S. *et al.* (2013) Nonhybrid, finished microbial genome assemblies from long-read SMRT sequencing data. *Nat. Methods*, **10**, 563–569.

Dasgupta,S. (1999) Learning mixtures of Gaussians. In: 40th Annual Symposium on Foundations of Computer Science(Cat. No.99CB37039), Vol. 1, pp. 634–644.

Doi,K. *et al.* (2013) Rapid detection of expanded short tandem repeats in personal genomics using hybrid sequencing. *Bioinformatics*, **30**, 815–822.

Durbin,R. *et al.* (1998) *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. 1st edn. Cambridge University Press, Cambridge, UK.

Eid,J. *et al.* (2009) Real-time DNA sequencing from single polymerase molecules. *Science*, **323**, 133–138.

Garrison,E. and Marth,G. (2012) Haplotype-based variant detection from short-read sequencing. *arXiv*, 1207.3907, 9.

Gymrek,M. *et al.* (2012) lobSTR: a short tandem repeat profiler for personal genomes. *Genome Res.*, **22**, 1154–1162.

Highnam,G. *et al.* (2013) Accurate human microsatellite genotypes from high-throughput resequencing data using informed error profiles. *Nucleic Acids Res.*, **41**, e32.

Downloaded from http://bioinformatics.oxfordjournals.org/ at :: on August 30, 2016

Jacobs,P.A. *et al.* (1980) Mechanism of origin of complete hydatidiform moles. *Nature*, **286**, 714–716.
Jin,P. (2000) Understanding the molecular basis of fragile X syndrome. *Hum. Mol. Genet.*, **9**, 901–908.
Jorda,J. and Kajava,A.V. (2009) T-REKS: identification of Tandem REpeats in sequences with a K-meanS based algorithm. *Bioinformatics*, **25**, 2632–2638.
Koren,S. *et al.* (2012) Hybrid error correction and de novo assembly of single-molecule sequencing reads. *Nat. Biotechnol.*, **30**, 693–700.
Langmead,B. and Salzberg,S.L. (2012) Fast gapped-read alignment with Bowtie 2. *Nat. Methods*, **9**, 357–359.
Lee,C. (2003) Generating consensus sequences from partial order multiple sequence alignment graphs. *Bioinformatics*, **19**, 999–1008.
Li,H. and Durbin,R. (2009) Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics*, **25**, 1754–1760.
Lim,K.G. *et al.* (2013) Review of tandem repeat search tools: a systematic approach to evaluating algorithmic performance. *Brief. Bioinform.*, **14**, 67–81.
Lo,C. *et al.* (2011) Strobe sequence design for haplotype assembly. *BMC Bioinformatics*, **12** (Suppl. 1), S24.
McKenna,A. *et al.* (2010) The genome analysis toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data. *Genome Res.*, **20**, 1297–303.
Needleman,S.B. and Wunsch,C.D. (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.*, **48**, 443–453.
Ono,Y. *et al.* (2013) PBSIM: PacBio reads simulator–toward accurate genome assembly. *Bioinformatics*, **29**, 119–121.
Pacific Biosciences. (2013). http://blog.pacificbiosciences.com/2013/10/data-release-long-read-shotgun.html (31 November 2014, date last accessed).
Pedregosa,F. *et al.* (2011) Scikit-learn: machine learning in Python. *J. Mach. Learn. Res.*, **12**, 2825–2830.
Rabiner,L.R. (1989) A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE*, **77**, 257–286.
Rausch,T. *et al.* (2012) DELLY: structural variant discovery by integrated paired-end and split-read analysis. *Bioinformatics*, **28**, i333–i339.
Ribeiro,F.J. *et al.* (2012) Finished bacterial genomes from shotgun sequence data. *Genome Res.*, **22**, 2270–2277.
Ritz,A. *et al.* (2010) Structural variation analysis with strobe reads. *Bioinformatics*, **26**, 1291–1298.
Sawaya,S. *et al.* (2013) Microsatellite tandem repeats are abundant in human promoters and are associated with regulatory elements. *PLoS One*, **8**, e54710.
Sawaya,S.M. *et al.* (2012) Promoter microsatellites as modulators of human gene expression. *Adv. Exp. Med. Biol.*, **769**, 41–54.
Sindi,S. *et al.* (2009) A geometric approach for classification and comparison of structural variants. *Bioinformatics*, **25**, i222–i230.
Tarailo-Graovac,M. and Chen,N. (2009) Using RepeatMasker to identify repetitive elements in genomic sequences. *Curr. Protoc. Bioinformatics*, Chapter 4, Unit 4.10.
Tsai,H.-F. *et al.* (2004) Analysis of trinucleotide repeats in different SCA loci in spinocerebellar ataxia patients and in normal population of Taiwan. *Acta Neurol. Scand.*, **109**, 355–360.
Tyson,C. *et al.* (2014) Expansion of a 12-kb VNTR containing the REXO1L1 gene cluster underlies the microscopically visible euchromatic variant of 8q21.2. *Eur. J. Hum. Genet.*, **22**, 458–463.
Veselinović,I. (2006) Microsatellite DNA analysis as a tool for forensic paternity testing (DNA paternity testing). *Med. Pregl.*, **59**, 241–243.
Walker,F.O. (2007) Huntington's disease. *Lancet*, **369**, 218–228.
Yalcin,B. *et al.* (2011) Sequence-based characterization of structural variation in the mouse genome. *Nature*, **477**, 326–329.
Ye,K. *et al.* (2009) Pindel: a pattern growth approach to detect break points of large deletions and medium sized insertions from paired-end short reads. *Bioinformatics*, **25**, 2865–2871.

Downloaded from http://bioinformatics.oxfordjournals.org/ at :: on August 30, 2016