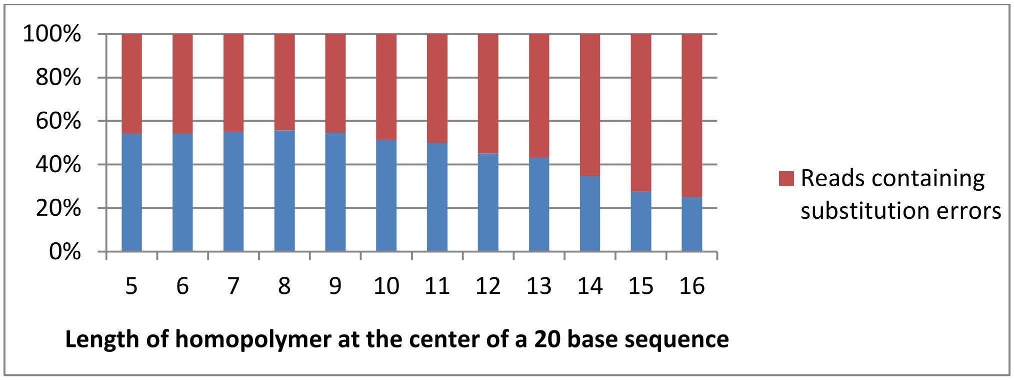
Genotyping of microsatellite loci using a Gaussian mixture model

Hongseok Tae, Dong-Yun Kim, John McCormick, Robert E. Settlage, and Harold R. Garner

Contents

Supplementary Figure S1



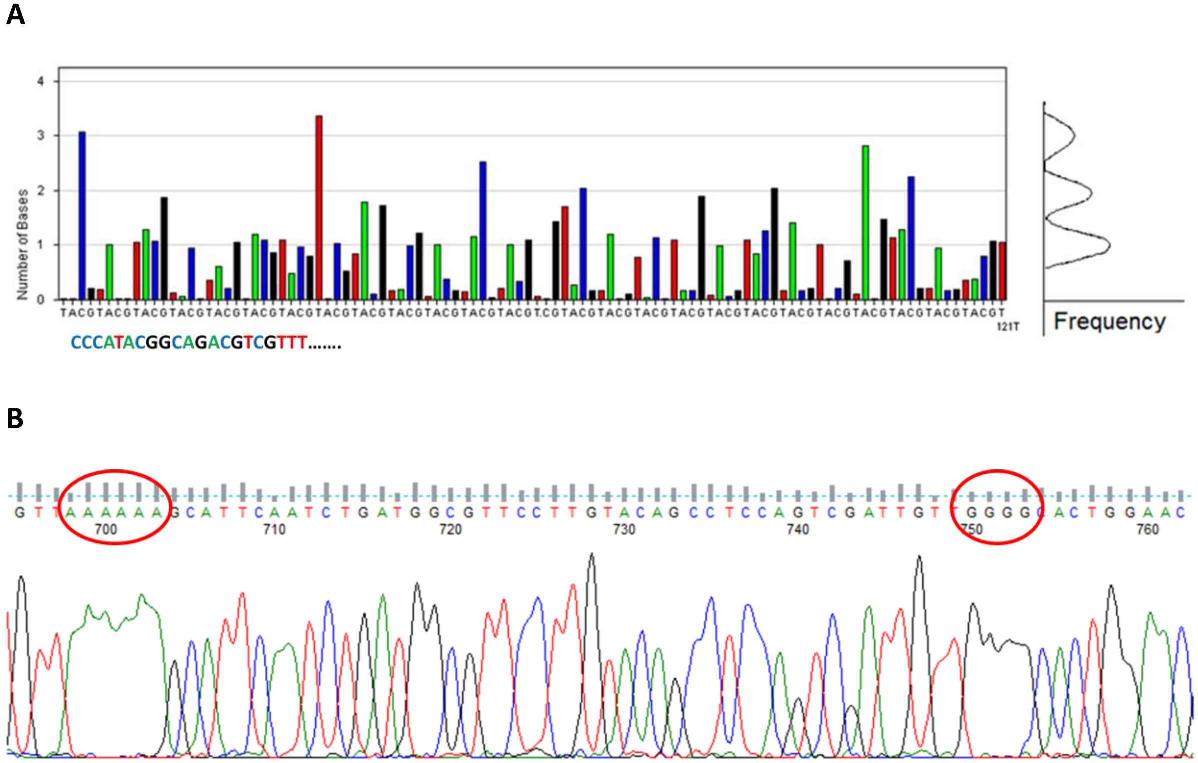
Proportion of reads containing substitution errors, which cover 20 base sequences containing different lengths of homopolymers at the center. The sequence reads were from the Illumina 36 cycle single end sequencing for a Drosophila inbred line. The proportion of reads containing substitution errors rapidly increased when the length of a homopolymer is longer than 9 bases.

Supplementary Figure S2

```
B.suis   GGCGCATTTTGCAACTGATTCTATGCC---GGGGGGGGGGAAGCGCATACGTTGGAGCGT
read1    GGCGCATTTTGCAACTGATTCTATGCCGGGGGGGGGGGGGAAGCGCATACGT--------
read2    GGCGCATTTTGCAACTGATTCTATGCCGGGGGGGGGGGGGAAGCGCATACGTTGG-----
read3    GGCGCATTTTGCAACTGATTCTATGCCGGGGGGGGGGGGGAAGCGCATACGTTGGAG---
read4    ------TTTTGCAACTGATTCTATGCCGGGGGGGGGGGGGAAGCGCATACGTTGGAGCGT
read5    GGCGCATTTTGCAACTGATTCTATGCC-GGGGGGGGGGGGAAG-----------------
read6    GGCGCATTTTGCAACTGATTCTATGCC-GGGGGGGGGGGGAAGCGC--------------
read7    GGCGCATTTTGCAACTGATTCTATGCC-GGGGGGGGGGGGAAGGGCA-------------
read8    GGCGCATTTTGCAACTGATTCTATGCC-GGGGGGGGGGGGAGGCGCATACG---------
read9    GGCGCATTTTGCAACTGATTCTATGCC-GGGGGGGGGGGGAAGCGCATACGTT-------
read10   GGCGCATTTTGCAACTGATTCTATGCC-GGGGGGGGGGGGAAGCGCATACGTTG------
read11   ---GCATTTTGCAACTGATTCTATGCC-GGGGGGGGGGGGAAGCGCATACGTTGGAGCGT
read12   ------TTTTGCAACTGATTCTATGCC-GGGGGGGGGGGGAAGCGCATACGTTGGAGCGT
read13   --------------CTGATTCTATGCC-GGGGGGGGGGGGAAGCGCATACGTTGGAGCGT
read14   GGCGCATTTTGCAACTGATTCTATGCC--GGGGGGGGGGGAAGCGCATACGTTGGAGCGT
read15   ---GCATTTTGCAACTGATTCTATGCC--GGGGGGGGGGGAAGCGCATACGTTGGAGCGT
read16   GGCGCATTTTGCAACTGATTCTATGCC---GGGGGGGGGGAAGCGCATACGT--------
read17   -----ATTTTGCAACTGATTCTATGCC---GGGGGGGGGGAAGCGCATACGTTGGAGCGT
```

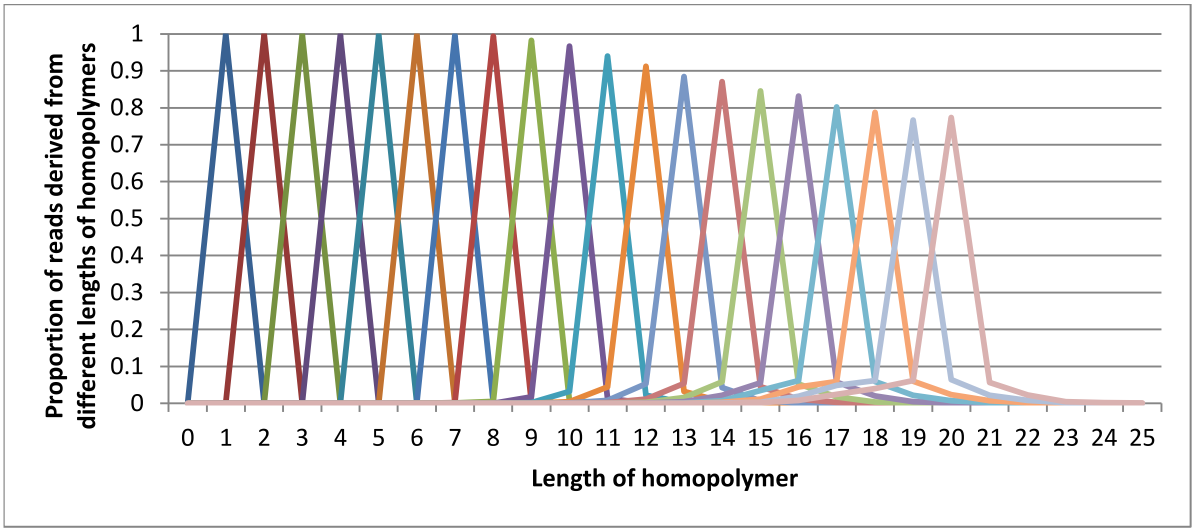
Recurrence of INDEL errors in sequence reads. The reference is a genomic sequence of *Brucella suis* 1330 which is a prokaryote with a haploid genome, and the locus is a unique region in the genome. The sequence reads were generated from the Illumina 101 cycle paired-end sequencing technology. Insertions or deletion errors by sequencing errors or other sources, such as PCR amplification error or individual cell mutation, results in various lengths of G homopolymers in sequence reads.

Supplementary Figure S3



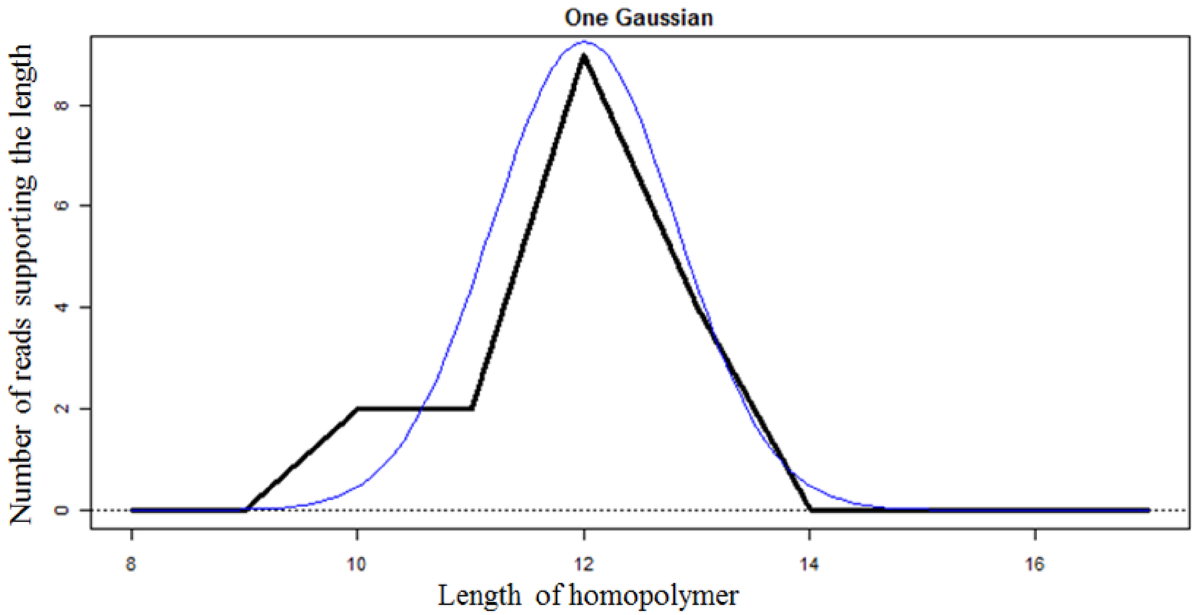
Signal intensities at the repeat sequences. (A) pyrosequencing. The number of homopolymer bases is decided by the intensity of a signal. The method often generates homopolymer errors. (B) Sanger sequencing. When homopolymer bases locate near to the beginning or ending position of sequencing, the boundaries of the homopolymer bases often become ambiguous and base-calling programs can generate homopolymer errors.

Supplementary Figure S4



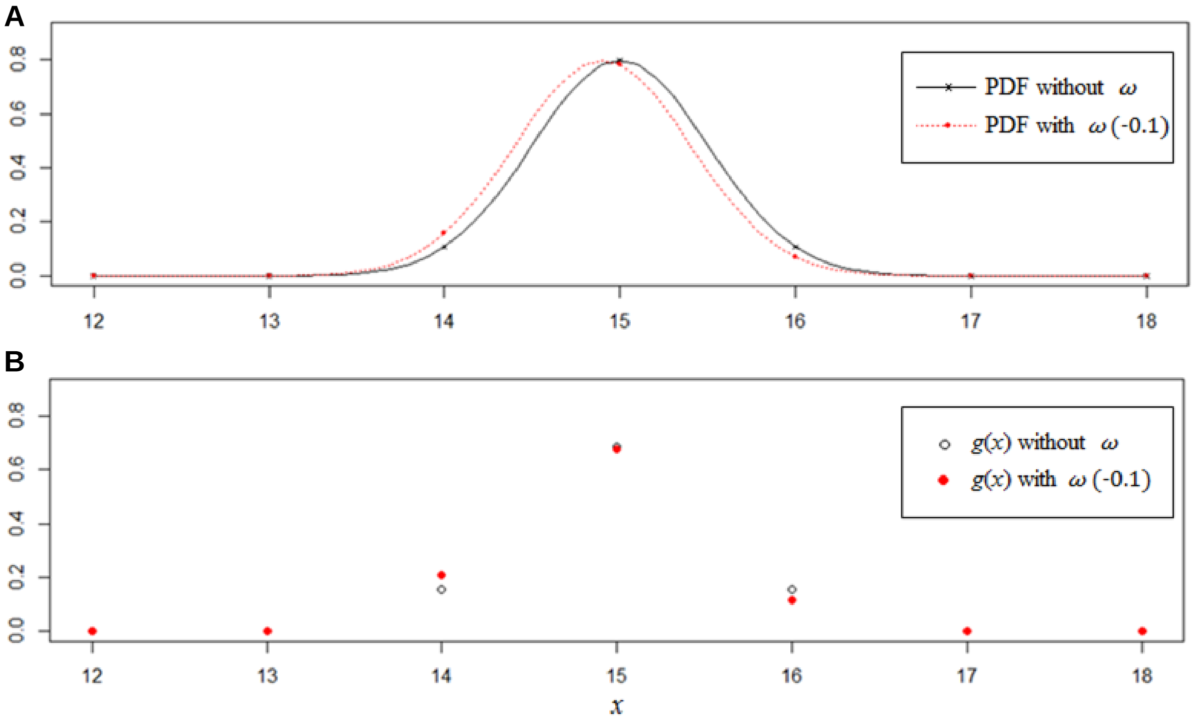
Different proportions of reads containing INDEL errors derived from the different lengths of homopolymers in the Illumina sequencing data. Each line represents the proportion of reads containing length *x* homopolymers derived from the called homopolymer alleles (the highest peak of a line, called by GenoTan). The graph was created with the Illumina sequencing data of the human genome HG01974 from the 1000 genome project.

Supplementary Figure S5



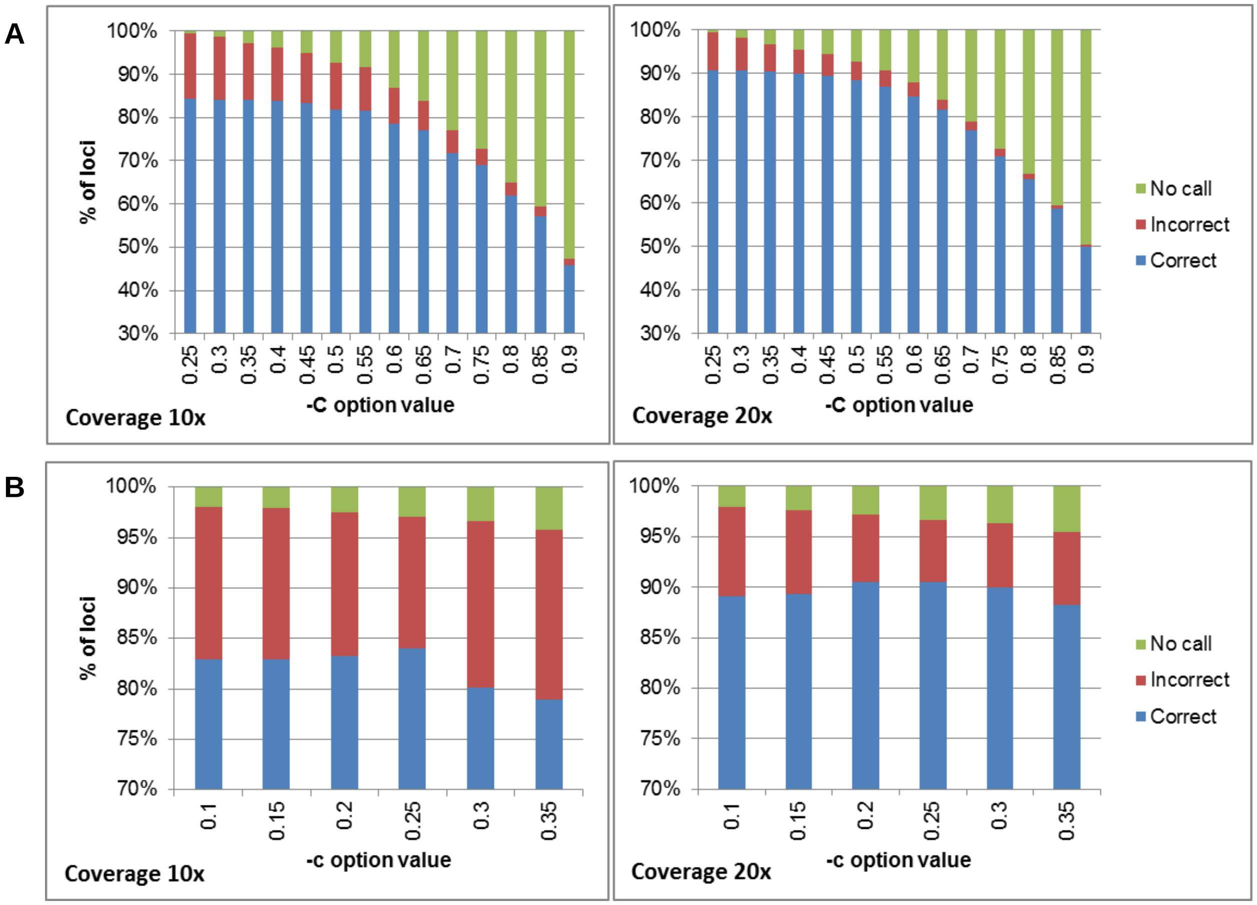
Gaussian distribution to express the bidirectional INDEL errors in sequence reads. The graph shows a Gaussian distribution for the locus at the supplementary figure 2. The chances of insertion errors and deletion errors are proportional.

Supplementary Figure S6



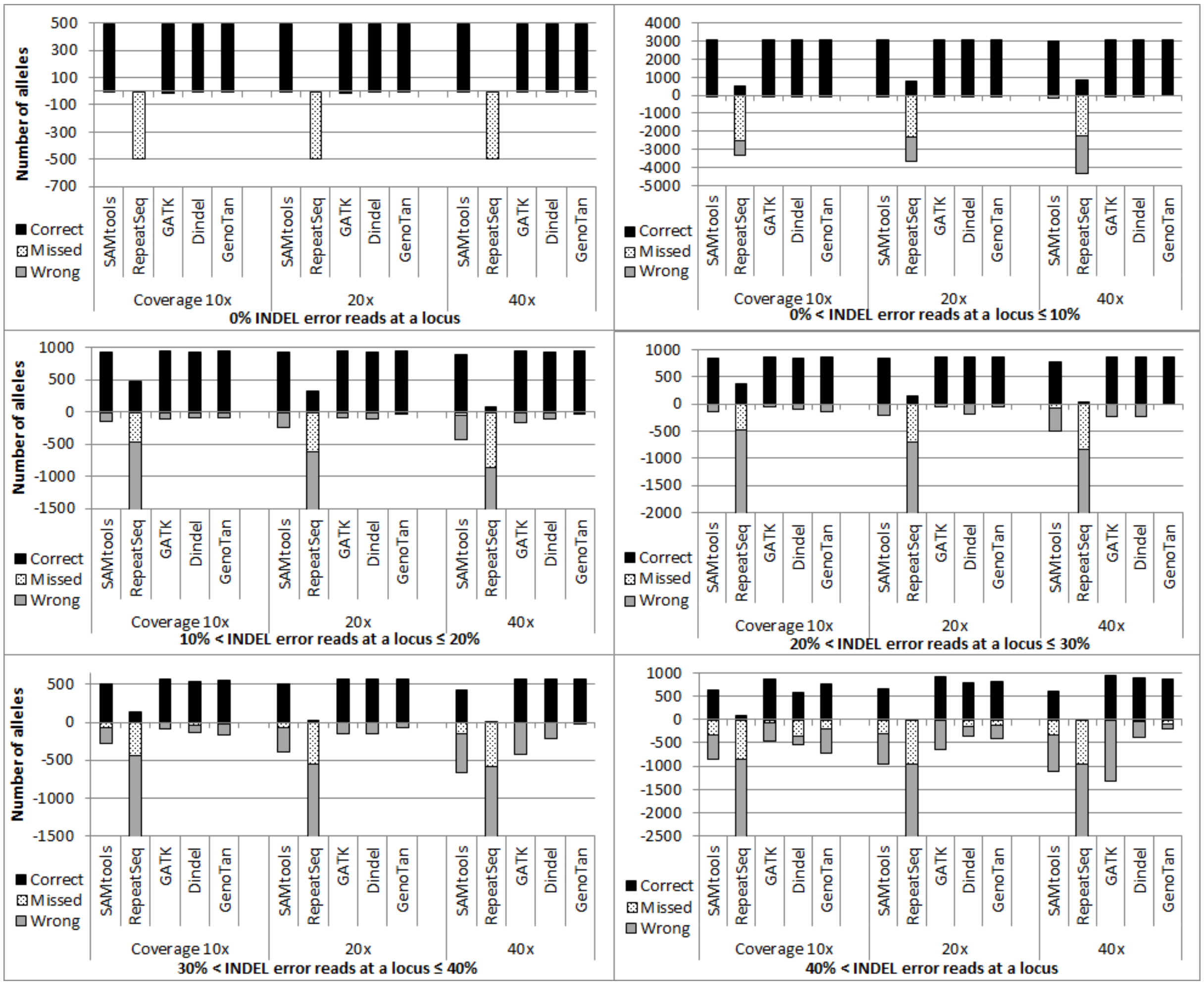
Application of error bias toward to deletion at a 15 base repeat sequence locus. (A) This graph shows outputs of the probability density functions (PDFs) of the Gaussian distributions $N(15, 0.5)$ and $N(15+\omega, 0.5)$, where the estimate of ω is -0.1. The parameter ω represents bias of INDEL errors at the repeat sequences in sequence reads derived from a repeat locus containing a 15 nucleotide allele. Because ω is a negative value, deletion is more likely to occur than insertion during sequencing. (B) The outputs of PDFs are discretized by the $p(x)$ function as in equation 1 where μ=15+ ω.

Supplementary Figure S7



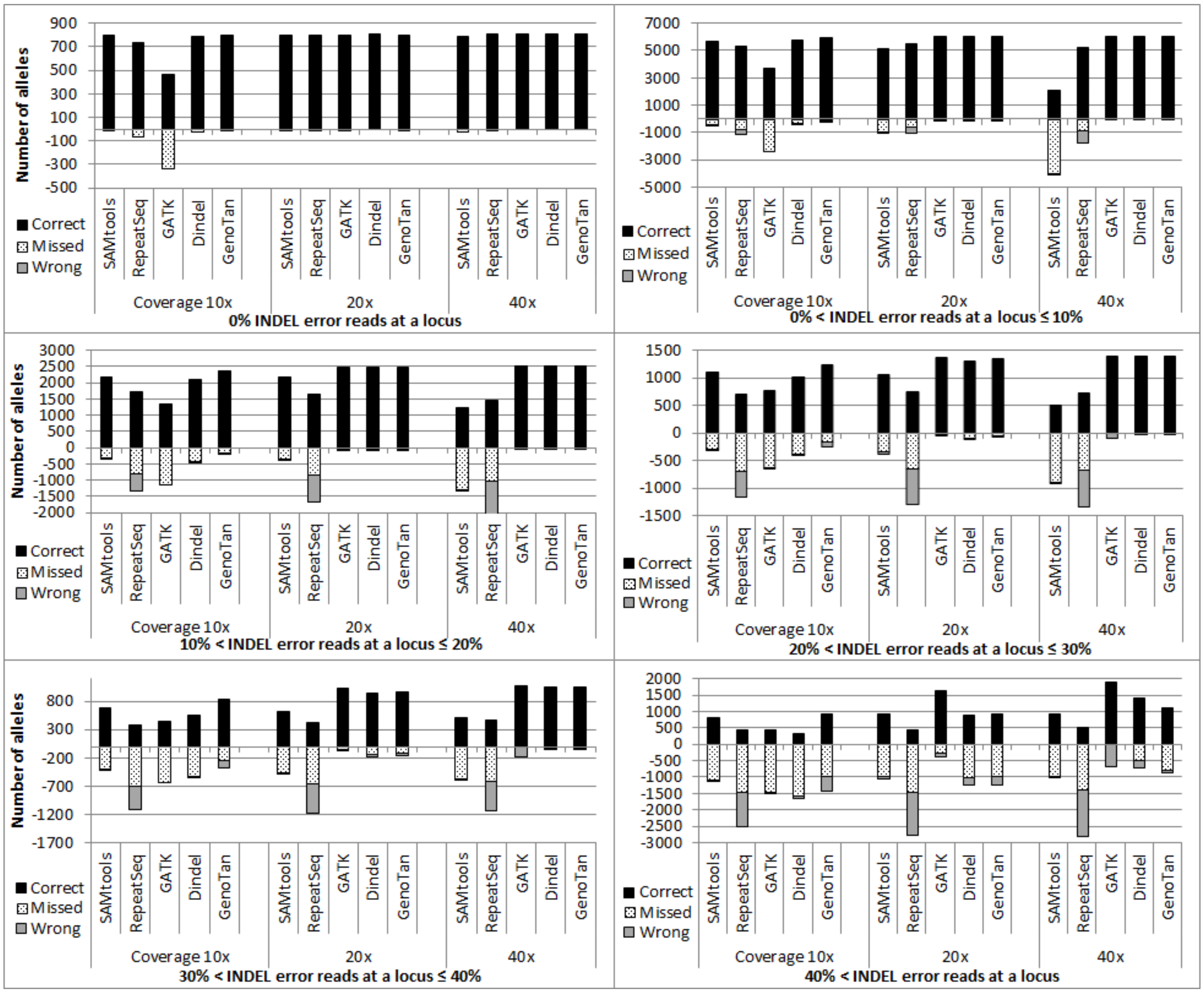
Performances of GenoTan with different "-C" and "-c" option values for the simulated data. (A) Different "-C" option values with a fixed "-c" option (0.25) of GenoTan were tested for the simulated data. The default value for the option "-C" is 0.35. (B) Different "-c" option values with a fixed "-C" option (0.35) of GenoTan were tested for the simulated data. The default value for the option "-c" is 0.25.

Supplementary Figure S8



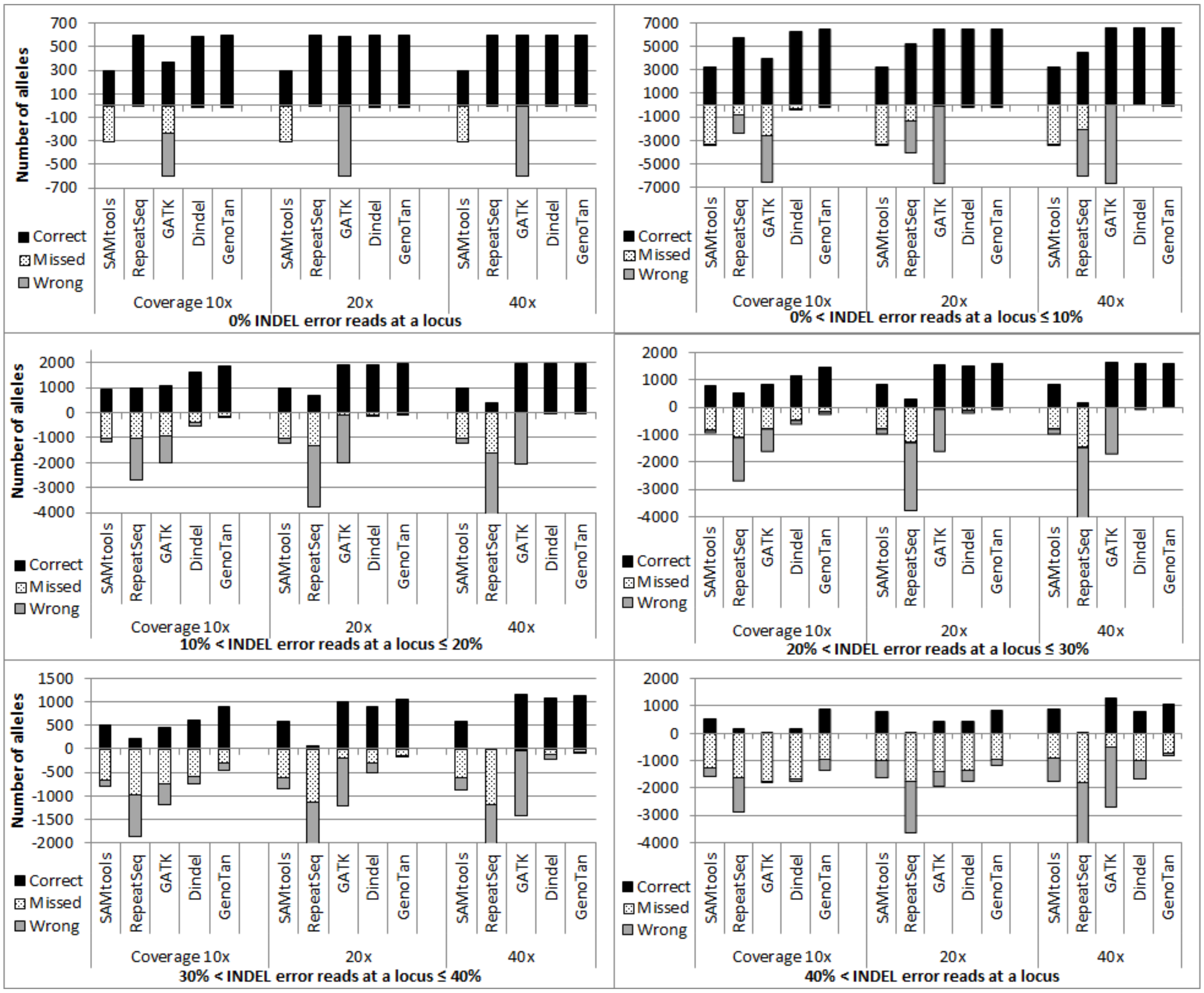
Number of correct, missed and wrong allele calls for homozygous loci of the simulated data in different error reads rates. The numbers of missed and wrong allele calls are shown as negative numbers. Genotan showed slightly higher rate of wrong allele calls than Dindel when the coverage is 10x and the proportion of INDEL error reads is more than 40%, but have very low rates of missed/wrong allele calls at most categories.

Supplementary Figure S9



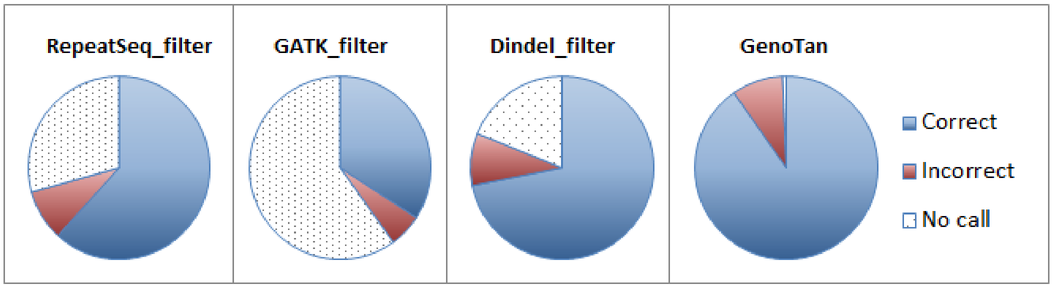
Number of correct, missed and wrong allele calls for reference/non-reference heterozygous loci of the simulated data in different error reads rates. The numbers of missed and wrong allele calls are shown as negative numbers.

Supplementary Figure S10



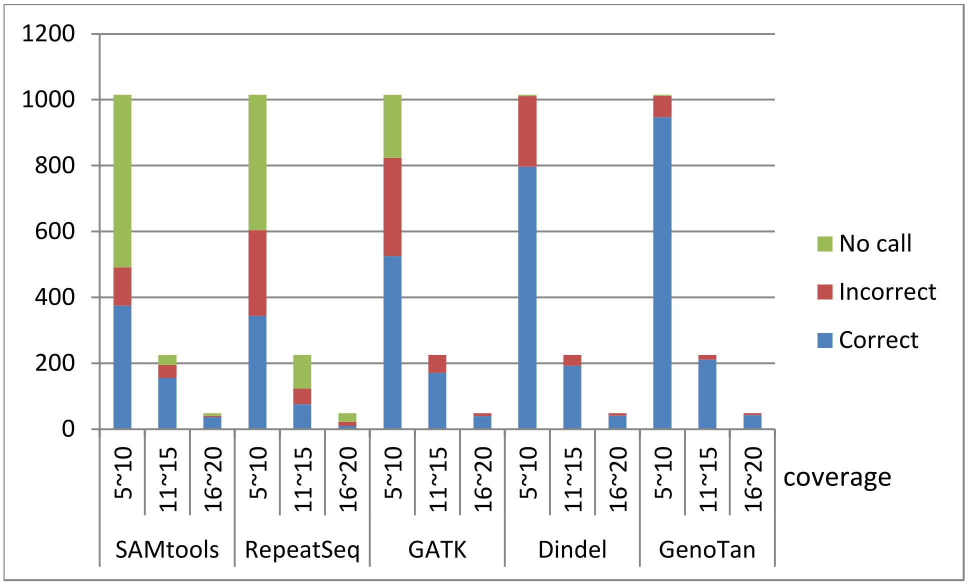
Number of correct, missed and wrong allele calls for non-reference/non-reference heterozygous loci of the simulated data in different error reads rates. The numbers of missed and wrong allele calls are shown as negative numbers. SamTools and GATK call two reference/non-reference heterozygous genotypes or one reference/non-reference heterozygous with additional non-reference homozygous genotypes for a non-reference/non-reference heterozygous locus. RepeatSeq frequently calls more than two alleles for a locus when a rate of error reads is high. Only Dindel and GenoTan call a non-reference/non-reference heterozygous genotype but Dindel also called many wrong allele calls as reference alleles since it favored calling reference alleles.

Supplementary Figure S11



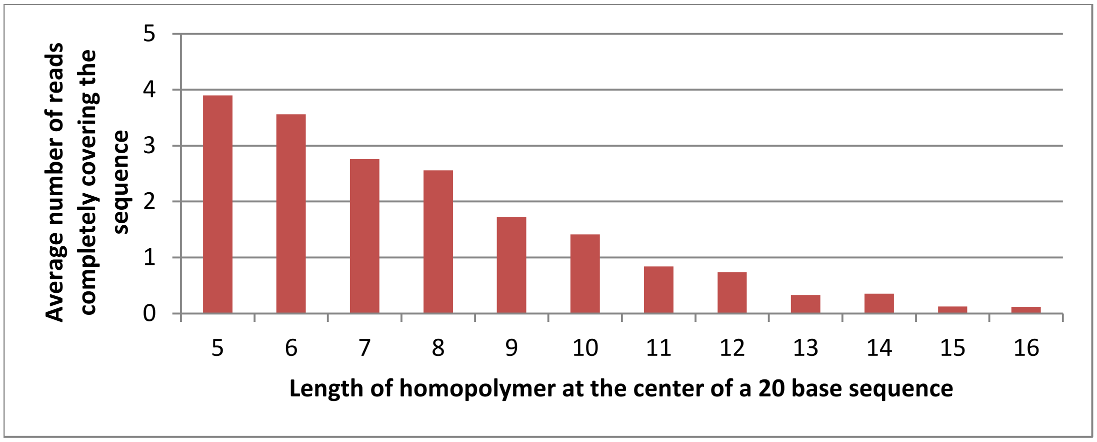
Proportions of no calls, incorrect calls and correct calls of the programs after filtering out reads not completely covering the repeat sequences. RepeatSeq, GATK and Dindel did not show significant improvement for the filtered data.

Supplementary Figure S12



Performance comparison of genotyping programs for sequence data of a single Drosophila inbred line (RAL-301)

Supplementary Figure S13



An average number of reads completely covering the 20 base sequence containing a homopolymer at the center. We compared the average number of reads completely covering 20 base sequences with different lengths of homopolymers at their center. The number of reads dropped significantly when the length of a homopolymer is long.

Supplementary Table S1

Estimated percentages of INDEL error reads at a locus	Homozygous	Heterozygous
0%	499	704
0< ~5%	2383	5000
5~10%	693	1340
10~15%	493	1103
15~20%	458	1158
20~25%	463	891
25~30%	385	610
30~35%	414	747
35~40%	159	395
40~45%	499	704
45% ~	953	1852
Total	6900	13800

Number of loci in the simulated data

The simulated data was generated from the Binomial random function (p=0.5) and two Gaussian random functions with $\mu_1=l_{L1}$ and $\mu_2=l_{L2}$ (where L_1 and L_2 were artificially generated from 1~8mer motif sets), and σ_1^2 and σ_2^2 for the functions were calculated from β_a(=0.00099) and β_b(=0.0153) which were estimated by a homopolymer decomposition method from the human genome NA19138. The percentage of INDEL error reads at a locus is estimated from the $g(x;\, l_{L1},\, l_{L2},\, \sigma_1^2,\, \sigma_2^2,\, \theta = 0.5)$ function for the given alleles, L_1 and L_2, at the locus. The average error rates of the simulated data may be much higher than real sequencing data since it includes many long homopolymer loci.

Supplementary Table S2

	Coverage 10x					Coverage 20x					Coverage 40x				
Overall	SAMtools	RepeatSeq	GATK	Dindel	GenoTan	SAMtools	RepeatSeq	GATK	Dindel	GenoTan	SAMtools	RepeatSeq	GATK	Dindel	GenoTan
Correct	53.9	34	47.4	76.2	84.1	51.1	29.5	61.7	86.3	90.6	37.6	23.6	56.5	89.5	94.9
Incorrect	36.7	38.5	30.8	11.3	13.2	40.2	52.1	35.2	8.7	5.7	42.8	62.5	43.4	9.1	1.8
No call	9.4	27.5	21.9	12.5	2.7	8.7	18.4	3.1	5	3.7	19.5	13.9	0.1	1.4	3.4
Homozygous loci (non-reference)	SAMtools	RepeatSeq	GATK	Dindel	GenoTan	SAMtools	RepeatSeq	GATK	Dindel	GenoTan	SAMtools	RepeatSeq	GATK	Dindel	GenoTan
Correct	80.7	0	90	89.1	87.4	76.7	0	89.4	89.5	93.8	69.5	48.5	81.3	87.1	97.4
Incorrect	14.8	45.8	9.6	6.5	11.9	19.9	58.1	10.6	8.9	5.3	27	51.5	18.7	12.4	1.9
No call	4.6	54.2	0.4	4.4	0.7	3.4	41.9	0	1.6	0.9	3.5	0	0	0.4	0.8
Heterozygous loci (ref/non-ref)	SAMtools	RepeatSeq	GATK	Dindel	GenoTan	SAMtools	RepeatSeq	GATK	Dindel	GenoTan	SAMtools	RepeatSeq	GATK	Dindel	GenoTan
Correct	81	53.9	52	75.1	82.5	76.7	50.8	95.6	88.9	88.9	43.4	41.5	88.2	94.6	93.5
Incorrect	2.1	30.7	0.3	4.4	13.9	2.3	42.6	1.7	4.3	6.3	1.7	54.5	11.8	3.9	1.6
No call	16.9	15.4	47.6	20.5	3.6	21.1	6.7	2.7	6.8	4.9	54.9	3.9	0	1.4	4.9
Heterozygous loci (non-ref/non-ref)	SAMtools	RepeatSeq	GATK	Dindel	GenoTan	SAMtools	RepeatSeq	GATK	Dindel	GenoTan	SAMtools	RepeatSeq	GATK	Dindel	GenoTan
Correct	0	48.1	0	64.3	82.5	0	37.7	0	80.6	89.2	0	29.4	0	86.8	93.7
Incorrect	93.2	39.1	82.4	23.1	13.8	98.5	55.6	93.3	13	5.6	99.8	66.7	99.8	10.9	1.8
No call	6.8	12.8	17.6	12.6	3.7	1.5	6.7	6.7	6.5	5.2	0.2	3.9	0.2	2.3	4.5

The performance comparison for the simulated data. The values in the table are percentages. This table is corresponding to the Figure 2A-D at the main text.

Supplementary Table S3

	Num. of target loci	Num. of reads on target loci	**lobSTR**	**RepeatSeq**	**GATK**	**Dindel**	**GenoTan**
Simulated data, depth 40	20,700	827,999	11M 05S	15S	57M 40S	4H 53M 4S	9M 20S
Mixed Drosophila sequence data	3,300	73,063	45S	6S	1M 3S	4M 8S	32S
SRR345592, chr1	58,246	3,215,705	36M 21S	32S	59M 5S	50H 24M 3S	10M 42S

Comparison of computational speeds. The programs were tested in a LINUX environment with Intel 2.67GHz CPUs and 32Gbyte memory. To test the computational speeds of genotyping programs for only the target loci, the reads mapped to target loci were used. The bitwise FLAG in the SAM format for each paired-end read was converted to a FLAG for single-end, after mapping and filtering (Dindel does not use information of mapped reads of which paired reads were not mapped in paired-end mapping, when we filter out the reads not mapped to the target loci).

Supplementary Table S4

Data	Coverage	Method	Correct	Incorrect	No call
simulated data	10X	Simple decision without Gaussian	58.3	36.0	5.7
		After the first step	81.1	16.3	2.6
		With all steps	84.1	13.2	2.7
	20X	Simple decision without Gaussian	64.8	30.3	4.9
		After the first step	90.1	6.6	3.3
		With all steps	90.6	5.7	3.7
	40X	Simple decision without Gaussian	72.4	19.9	7.8
		After the first step	94.9	1.8	3.3
		With all steps	94.9	1.7	3.4
Merged Drosophila inbred samples		Simple decision without Gaussian	63.1	36.9	0.0
		After the first step	89.8	9.4	0.8
		With all steps	90.2	9	0.8
pIRS simulated data		Simple decision without Gaussian	65.1	32.4	2.5
		After the first step	91.8	5.7	2.5
		With all steps	91.8	5.7	2.5

Comparison of genotyping results from different steps in the GenoTan's process. For the simple decision, a similar rules to the Final decision of GenoTan,

$C_{high} = N_{highest} / T$ and $C_{low} = N_{2nd} / N_{highest} \times (N_{2nd} / T)$

,where $N_{highest}$= is the read count supporting an allele candidate with the highest read frequency, N_{2nd}= is the read count supporting an allele candidate with the second highest read frequency and T is the total read count for the locus, were calculated with cutoff values 0.35 and 0.25, which are default values of GenoTan to decide the allele calls. (Note1. C_{low} has been used to filter out allele candidates supported by reads from mismapping, individual cell mutation or PCR amplification artifacts (which are difficult to identify with statistic approaches), when a ratio of N_{2nd} to $N_{highest}$ is too low) (Note2. $p_{L_{low}}(l_{L_{low}})$ in the Final decision of GenoTan can be higher than $p_{L_{high}}(l_{L_{high}})$ because of the Gaussian mixture model, while N_{2nd}/T is always equal to or lower than $N_{hightest}/T$. Other simple rules may work better than this simple decision)

The whole genotyping process of GenoTan is composed of two step calculations using the Gaussian distribution. The genotyping results after only the first step were also compared. Since the pIRS simulator didn't account the homopolymer errors in generating simulated data, the three methods didn't show significantly different results. (Note3. Since pIRS simulator did not account homopolymer errors, the two step calculation didn't improve the accuracy for its data from the first step calculation.)

Supplementary Methods

Rescaling the Gaussian cumulative distribution function

The approximate probability of that a length *x* repeat sequence in a read derived from the length *L* allele is *x* bases can be estimated by $\int_{x-0.5}^{x+0.5} f_L(t)dt$. Since *x* is a natural number, we need to rescale to make the distribution a proper probability mass function. The scale factor ς is computed from the following.

$$1 = \varsigma\left(\int_{1-0.5}^{1+0.5} f_L(t)dt + \int_{2-0.5}^{2+0.5} f_L(t)dt + \ldots + \int_{X-0.5}^{X+0.5} f_L(t)dt + \ldots\right) = \varsigma \sum_{x=1}^{\infty} \int_{x-0.5}^{x+0.5} f_L(t)dt$$

$$= \varsigma\left(\int f_L(t)dt - \left(\int f_L(t)dt - \left(\sum_{x=1}^{\infty} \int_{x-0.5}^{x+0.5} f_L(t)dt\right)\right)\right) = \varsigma\left(\int f_L(t)dt - \left(\int f_L(t)dt - \int_{0.5}^{\infty} f_L(t)dt\right)\right)$$

$$= \varsigma\left(\int f_L(t)dt - \int_{-\infty}^{0.5} f_L(t)dt\right) = \varsigma\left(1 - F(0.5; L, \sigma_L^2)\right)$$

$$\varsigma = \frac{1}{\left(1 - F(0.5; L\ , \sigma_L^2)\right)}$$

Note that for a large *L*, ς is close to 1.

Pseudo code applying the Nonlinear Least-Squares regression

We used the Nonlinear Least-Squares (NLS) regression provided by the GNU Scientific Library (GSL, http://www.gnu.org/software/gsl). The NLS regression is an extension model of the linear least squares regression to estimate unknown parameters fitting to a given data generated from a non-linear model. The NLS function in the GSL requires user defined functions coordinated to the non-linear model and takes a data vector containing training data.

This is a pseudo code to use the NLS regression with the mixture form of two cumulative distribution functions ($g(x; L_1,L_2)$).

```
Set minDiff to 0
FOR each genotype candidate ($L_1$,$L_2$)
    MIN = min. observed length – $k$
    MAX = max. observed length + $k$
    Initialize vector[ from MIN to MAX] to 0
    $\mu_{L1} = l_{L1}$
    $\mu_{L2} = l_{L2}$
    IF it is the second regression step
        $\mu_{L1}$ += $\omega_{L1}$
        $\mu_{L2}$ += $\omega_{L2}$
        FOR $x$ = MIN to MAX
            vector[$x$] = $g(x; \mu_{L1}, \mu_{L2}, \sigma_1^2 = \upsilon_{L1}, \sigma_2^2 = \upsilon_{L2}, \theta = 0.5)$
        END FOR
    END IF
    FOR $x$ = MIN to MAX
        vector[$x$] += total mapping probability of reads with length $x$ repeat sequence (instead **observed number**)
    END FOR
    total = sum(vector)
    FOR $x$ = MIN to MAX
        vector[$x$] = vector[$x$]/total;
    END FOR
    ($\sigma_{L1}, \sigma_{L2}, \theta$) = estimation_with_NLS($\mu_{L1}, \mu_{L2}$, vector)
    diff = 0
    FOR $x$ = MIN to MAX
        diff += (vector[$x$]- $g(x; \mu_{L1}, \mu_{L2}, \sigma_1^2 = \sigma_{L1}^2, \sigma_2^2 = \sigma_{L2}^2, \theta))^2$;
    END FOR
    IF minDiff > diff
        minDiff = diff
    END IF
END FOR
RETURN a genotype candidate with the smallest diff
```

The NLS function provided by the GSL requires initial values of the parameters to be trained and incorrect initial values can cause local maxima. To reduce this problem, GenoTan tests initial values of σ_{L1}, σ_{L2} from 0.15 to 1.15 and chooses the output parameters (σ_{L1}, σ_{L2}, θ) of the NLS with the initial values, which produce the least square errors.

```
function estimation_with_NLS($\mu_{L1}$, $\mu_{L2}$,vector)
    Set minDiff to 0
    Set min_$\sigma_{L1}$ to 0
    Set min_$\sigma_{L2}$ to 0
    Set min_$\theta$ to 0
    FOR ($\sigma_{L1}$, $\sigma_{L2}$) = 0.15 to 1.15, by 0.05
        $\theta$ = 0.5
         ($\sigma_{L1}$, $\sigma_{L2}$, $\theta$) = NLS_with_ $g$_function($\mu_{L1}$, $\mu_{L2}$, $\sigma_{L1}$, $\sigma_{L2}$, $\theta$, vector)
        diff = 0
        FOR $x$ = MIN to MAX
            diff += (vector[$x$]- $g(x; \mu_{L1}, \mu_{L2}, \sigma_1^2 = \sigma_{L1}^2, \sigma_2^2 = \sigma_{L2}^2, \theta))^2$;
        END FOR
        IF minDiff > diff
            minDiff = diff
            min_$\sigma_{L1}$ = $\sigma_{L1}$
            min_$\sigma_{L2}$ = $\sigma_{L2}$
            min_$\theta$ = $\theta$
        END IF
    END FOR
    RETURN (min_$\sigma_{L1}$, min_$\sigma_{L2}$, min_$\theta$)
```

Homopolymer decomposition

The homopolymer decomposition method is a process to decompose sequences into a set of homopolymers to estimate parameters ω_L and υ_L. For example, the 'TAAACAAATAAA' sequence is composed of three 'AAA', two 'T' and one 'C' ('T' and 'C' are monomers but we treat them as homopolymers). To make the problem tractable, we assume the following:

A1) Insertion and deletion error events in each homopolymer are independent from those in the neighborhood homopolymers.

A2) Each error at a base is independent from the errors at neighborhood bases.

A3) Only one of the insertion or deletion error events in the repeat sequence of a read is considered. This means we only consider the observed event. For example, we only consider 1 base deletion error for {1 base insertion + 2 base deletion}, {2 base insertion + 3 base deletion} and so on.

A4) All of the insertion errors are derived only from the existing neighborhood nucleotides. If a sequence read has 'TGAAATAAATAAA' sequence and the second base 'G' is identified as an insertion error, we assume the first homopolymer 'T' or the second homopolymer 'AAA' caused the insertion error.

A5) Probabilities of insertion and deletion errors are affected only by the lengths of homopolymers. We ignore the other factors including high error rates at the end bases of sequence reads, GC-content biases during library amplification/sequencing and effects of specific sequences such as 'GGC' inducing sequencing errors which are known to occur in the Solexa next generation sequencing platform (Nakamura, et al., 2011).

As an example, suppose that 15 and 1 reads containing 'TAAATAAA' and 'TAATAAA' respectively, have been mapped to a locus *A*. We would conclude that the inherited allele is 'TAAATAAA' and 'TAATAAA' is derived from 'TAAATAAA' by a 1-base deletion error. Then an estimated average length of the sequence in a read which is derived from the 'TAAATAAA' allele is 7.93 bases (15/16×8 + 1/16×7). For another example, suppose that 14, 2 and 1 reads containing 'GTTTGTTT', 'GTTGTTT', and 'GTTTTCGTTT' respectively, have been mapped to another locus *B*. We would conclude that the inherited allele is 'GTTTGTTT', and 'GTTGTTT' and 'GTTTTCGTTT' have a 1-base deletion error and a 2-base insertion error respectively. Then an estimated average length of the sequence in a read which is derived from the 'GTTTGTTT' allele is 7.99 bases (14/17×8 + 2/17×7 + 1/17×10). Based on the assumption A5, the alleles of locus *A* and *B* can be treated as the same sequence in an abstract form, {1N3N1N3N}, and the average length of the sequence can be calculated together. Then the estimated average length of the sequence in a read derived from {1N3N1N3N} is 7.97 (=29/33×8 + 3/33×7 + 1/33×10). By simply subtracting 7.97 from 8, we can estimate ω, which represents the error bias toward deletion or insertion at the microsatellite sequence in a read derived from the {1N3N1N3N} allele. While the positive result of the subtraction represents bias toward insertion, the negative result represents bias toward deletion in sequence reads derived from the allele.

If we collect more reads derived from all loci containing the {1N3N1N3N} alleles, we can estimate a more accurate average length of repeat sequences in reads derived from the alleles. But some alleles (e.g. {40N10N}) may not be covered by enough reads to be used as the training set to estimate the accurate average length, so we apply the homopolymer decomposition method. The average length of the sequences in the previous example is 7.97 and the abstract form of the allele is {1N3N1N3N}. This form can be decomposed into '2·{1N} + 2·{3N}'. Since each {iN} can be regarded as an individual variable, we can define them as $\{N_1, N_2, N_3, N_4 \ldots\}$, and the example can be described by '$7.97 = 2 \cdot N_1 + 2 \cdot N_3$'. Then we can write an equation summarizing all possible allele sequences as follows

$$Y = n_1 \cdot N_1 + n_2 \cdot N_2 + n_3 \cdot N_3 + ... = \sum_i^I n_i \cdot N_i \tag{S1}$$

where Y is the average length of repeat sequences in reads derived from a single abstracted allele. Due to the limitation of the current sequencing technology, the maximum length, I, of a sequence, we can obtain, is not infinite. Y and n_i for an allele are simply calculated from the training data, and $\{N_1, N_2, N_3, N_4 \ldots\}$ can be estimated by a linear regression method. Moreover, because of the correlation between N_i and $N_{\mathrm{i+1}}$, we define N_i with two additional cofactors α_a and α_b as

$$N_i = i + \alpha_a \cdot i + \alpha_b \tag{S2}$$

where α_b and α_b represent a bias gradient and an initial bias respectively. Then we can write the equation S1 as

$$Y = \sum_i^I n_i (i + \alpha_a \cdot i + \alpha_b) \tag{S3}$$

Because the variables i and n_i represent the length and the number of each homopolymer at a given abstracted allele respectively, the equation S3 can be modified as follows

$$Y - (\text{allele length}) = \sum_i^I n_i (\alpha_a \cdot i + \alpha_b) \tag{S4}$$

The cofactors α_a and α_b are estimated by a nonlinear regression method from the genotyping results of the first genotyping regression step and are used to calculate the parameters ω_L for a given allele candidate L in the second genotyping regression step from the following function

$$\omega_L = \text{get\_mean\_bias(consensus sequence of allele } L, \alpha_a, \alpha_b) = \sum_i^I n_i (\alpha_a \cdot i + \alpha_b) \tag{S5}$$

since we can simply count the number of each length i homopolymer from the consensus sequence of the given allele candidate L.

Based on the assumption A1 and A2, the parameter υ_L can be estimated in the same way with ω_L. For a given abstracted allele {1N3N1N3N}, the variance is calculated by the NLS regression function. And the abstracted form is decomposed into ‘$2 \cdot M_1 + 2 \cdot M_3$’, where M_i is a corresponding variable to N_i in the previous paragraph. Then we can write an equation summarizing all possible allele sequences as follows

$$Z = \sum_i^I n_i \cdot M_i \tag{S6}$$

where Z is an estimated variance of lengths of microsatellite sequences in reads derived from a given abstracted allele. We define M_{i} with two additional cofactors β_a and β_b as

$$M_i = i^2 \cdot \beta_a \cdot e^{i \cdot \beta_b} \tag{S7}$$

$$Z = \beta_a \cdot \left(\sum_i^I n_i \cdot i^2 \cdot e^{i \cdot \beta_b} \right) \tag{S8}$$

which describes rapid change of variances according to the length of homopolymers. They are also estimated by a nonlinear regression, and are used to estimate the parameters υ_L for a given allele candidate L in the second genotyping regression step from the following function

$$\upsilon_L = \text{get\_var\_prior(consensus sequence of allele } L, \beta_a, \beta_b) = \beta_b \left(\sum_i^I n_i \cdot i^2 \cdot e^{i \cdot \beta_b}) \right) + \varphi \tag{S9}$$

where φ with default value 0.5, is added to υ_L to reduce the probability of allele candidates supported by a small number of reads.

Supplementary Notes

Limitation of the Bayesian approach considering only sequencing errors for INDEL genotyping

Most genotyping programs, including Dindel, employing the Bayesian approach calculate a likelihood outcome for a locus in a diploid genome from the following equation;

$$L(H_pH_m|R_i) = \prod_i p(R_i|H_pH_m) = \prod_i [\frac{p(R_i|H_p)}{2} + \frac{p(R_i|H_m)}{2}]$$

where R_i is the i^{th} observed read at the locus, and H_p and H_m are paternal and maternal hapolotype candidates respectively. Since this approach assumes that all reads are generated from one of two chromosomes, genotyping programs incorporate the mapping quality scores generated by mapping programs into the approach to control incorrectly mapped reads. However, relying on mapping quality scores could result in false positive prediction for INDEL genotyping, since mapping programs cannot calculate all possible alignments and often generate incorrect quality scores. Suppose that we have 5 reads, 6 reads and 1 read, which are mapped to a same locus, containing five base deletions, four base deletions and one base insertion respectively, and mapping quality scores of all reads are same. We would expect that the heterozygous genotype [allele with two base deletion, allele with three base deletion], which is presented by G(-4, -6), produces the highest probability among all possible genotype candidates, but the likelihood outcome of G(-4, 1) could be the highest because $p(R_{(1\ base\ insertion)}|$ {-4, -5}) by sequencing errors is close to 0. Thus, an effective method to handle noise due to incorrectly mapped reads (the noise could also be the result of individual cell mutation or PCR amplification artifacts) is necessary, and we used a regression approach which could filter out such noise efficiently.

Shell scripts to run genotyping programs

GATK realignment

```
cat test.repeat.lst | perl -ne '@arr=split/\t/; print "$arr[0]:$arr[1]-".($arr[1]+$arr[2]-1)."\n";' > test.repeat.intervals
perl getReadsOnMicrosatellites.pl -m test.repeat.lst -s  $bam -o test.ms.sam    ### gets reads overlapping 2 bases down/upstream of targets.
samtools view -S test.ms.sam -bo test.ms.bam
samtools index test.ms.bam

java -jar ~/opt/GATK/GenomeAnalysisTK.jar -T IndelRealigner -R $ref -targetIntervals test.repeat.intervals -I test.ms.bam -o
test.GATK.bam
```

DIndel

```
mkdir dindel
/bin/rm -f dindel/*dindel*
~/dindel/dindel --analysis getCIGARindels --bamFile $bam --outputFile dindel/dindel --ref $ref
~/dindel/makeWindows.py --inputVarFile dindel/dindel.variants.txt --windowFilePrefix dindel/dindel.realign_windows --numWindowsPerFile
1000
perl -e '
```

```
  for($i = 1; $i < 10000; $i++){
    last if(not -e "dindel/dindel.realign_windows.$i.txt");
    system("~/dindel/dindel --analysis indels --doDiploid --bamFile $bam --ref $ref --varFn dindel/dindel.realign_windows.$i.txt --libFile dindel/dindel.libraries.txt --outputFile dindel/dindel.stage2.windows.$i");
  }
'
perl -e '
  for($i = 1; $i < 10000; $i++){
    last if(not -e "dindel/dindel.stage2.windows.$i.glf.txt");
    system("echo dindel/dindel.stage2.windows.$i.glf.txt >> dindel/dindel.stage2.files.txt");
  }
'

~/dindel/mergeOutputDiploid.py --inputFiles dindel/dindel.stage2.files.txt --outputFile dindel/dindel.vcf --ref $ref

rm dindel/*.txt -f
```

GATK

```
java -Xmx1g -jar GATK_dir/GenomeAnalysisTK.jar -T UnifiedGenotyper -R $ref -glm INDEL -I $bam -o GATK.vcf
##  “-allowPotentiallyMisencodedQuals”  is required for GATK2
```

SamTools

```
samtools mpileup -uf $ref $bam | bcftools/bcftools view -Abvcg - > samtools.raw.bcf
bcftools/bcftools view samtools.raw.bcf | bcftools/vcfutils.pl varFilter -W 50 > samtools.vcf
```

lobSTR

```
python ~/lobSTR/scripts/GetSTRInfo.py $bed $ref > test.str.tab

mkdir lobSTR_index

python ~/lobSTR/scripts/lobstr_index.py --str $bed  --ref $ref --out_dir lobSTR_index
~/lobSTR/bin/lobSTR  --fastq --p1 $fq_1 --p2 $fq_2 --index-prefix lobSTR_index/lobSTR_ -o test.lobSTR

allelotype --command simple --bam  test.lobSTR.aligned.bam  --strinfo test.str.tab  --out test.lobSTR  --haploid no
python ~/lobSTR/scripts/lobstr_to_vcf.py --gen test.lobSTR.genotypes.tab --sample test --out test.lobSTR
```

RepeatSeq

```
repeatseq  $bam  $ref  $regions
```

Supplementary Results

Performance test with two different mapping programs for simulated data generated by pIRS from the *Drosophila* reference

The performance of genotyping programs were compared for mapping results generated by two different mapping programs, BWA (0.6.1) and Novoalign (2.08.02) (default options). To create a simulated sequence reads, we selected the first 10,000 microsatellite loci in the *Drosophila* reference sequence and created two different chromosome sets by inserting INDELs to the reference sequence for the loci using a random function (1~8 repeat units variation, {non-ref Homozygous, ref/non-ref Heterozygous, non-ref/non-ref Heterozygous}). From the two chromosome sets, simulated sequence reads were generated by a pIRS simulator resulting in 20 read coverage depth. Then, BWA and Novoalign were used to map reads to the reference sequence, and genotyping programs, GATK, Dindel and GenoTan (with "-c 0.15" option because of no homopolymer or PCR amplification errors), were used to call genotypes for the target loci. The profiling results of lobSTR (2.0.2, default options, "allelotype --command simple") were also compared.

Mapping program	BWA				Novoalign				
	GATK	**Dindel**	**GenoTan**	**RepeatSeq**	**GATK**	**Dindel**	**GenoTan**	**RepeatSeq**	**lobSTR**
Correct	79.8	92.4	91.8	53.7	84.3	95.6	95.4	55.0	2.8
Incorrect	9.5	2.7	5.7	3.8	6.9	1.5	2.7	1.3	24.2
No call	10.7	4.9	2.5	42.5	8.7	2.9	2.0	43.7	73.0 {<2, >6} mer motif : 49%

The numbers are percentages (%).

GenoTan is slightly more sensitive to the mapping results than Dindel. Since the pIRS simulator didn't account the INDEL errors and substitution errors induced by long homopolymer runs (Supplementary Figures S1 and S2) and the errors due to individual cell mutation or PCR amplification artifacts, for which our method has strength, the results only partially represent real data and might be biased to the local alignment-based approach used by Dindel. Considering the computational speeds, GenoTan is highly competitive with respect to other genotyping programs.

Comparison of GATK ver. 1 and ver. 2

Since the new version of GATK was released during our experiment, we compared the results of GATK ver. 1.6-9 and ver. 2.3-9 to test the difference in performance.

		Simulated data			**Mixed data of two Drosophila inbred**	**pIRS simulated data**
	Coverage	**10x**	**20x**	**40x**		
Version 1.6-9	Correct	47.4	61.7	56.5	45.9	79.8
	Incorrect	30.8	35.2	43.4	8.7	9.5
	No call	21.9	3.1	0.1	45.4	10.7
Version 2.3-9	Correct	45.1	58.5	59.5	44.9	79.7
	Incorrect	33.3	38.8	39.3	7.6	9.6
	No call	21.6	2.7	1.2	47.5	10.7

Two versions of GATK showed very similar results.