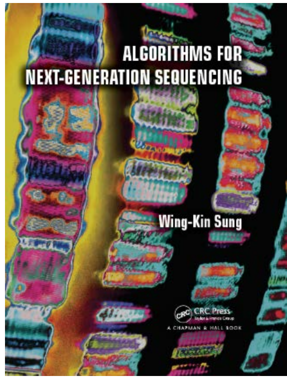
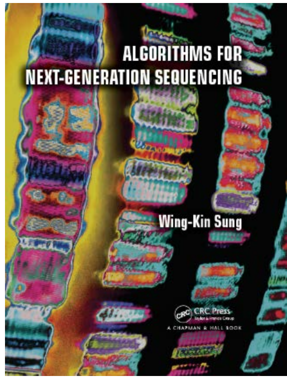


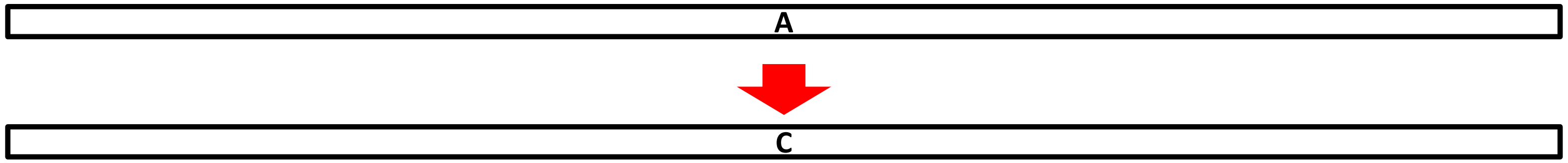
# Algorithms for Next-Generation Sequencing

# SNV calling

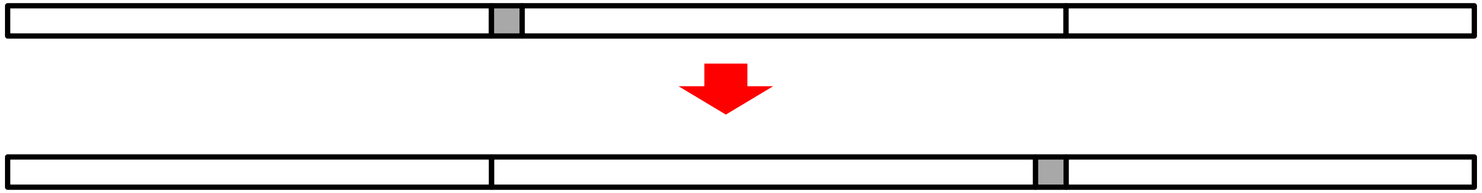
# Variations in our genome



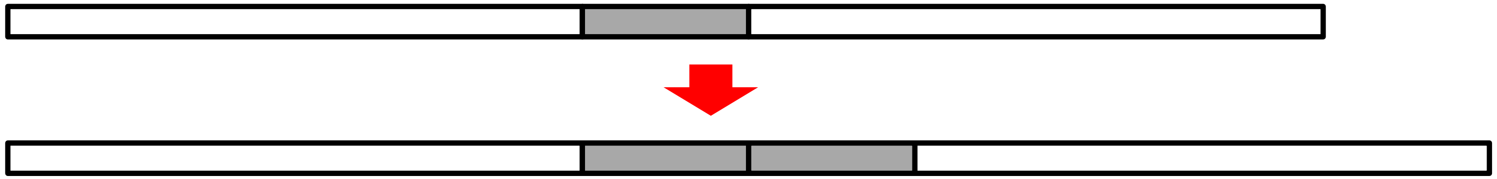
## (a) Single Nucleotide Polymorphism (SNV)



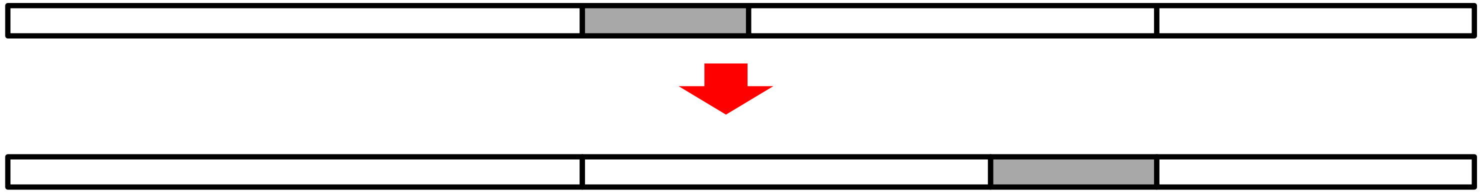
## (b) Small indel

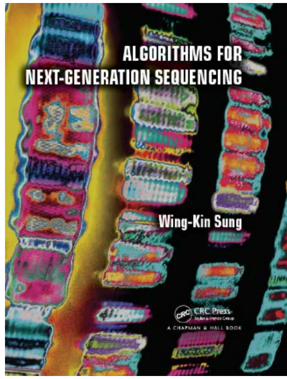


## (c) Copy Number Variation (CNV)



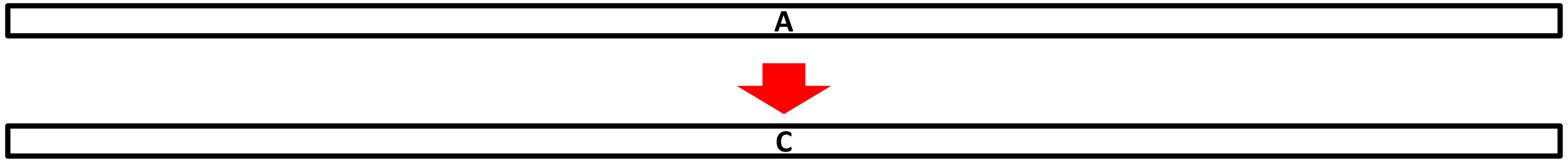
## (d) Structural Variation (SV)



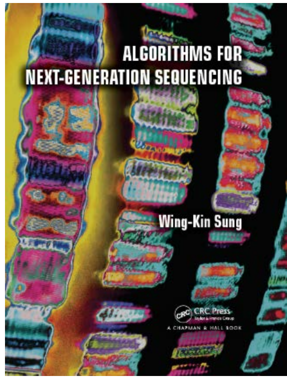


# SNV

- SNV is a point mutation.
- It is the most frequent genome variations.
- Each individual expects to have one SNV per 1000bp.



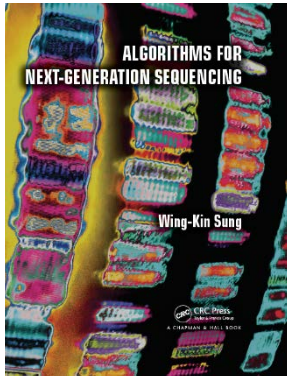
- SNV can occur in
  - protein coding region (a sequence of codons) or
  - non-coding region.



# SNV

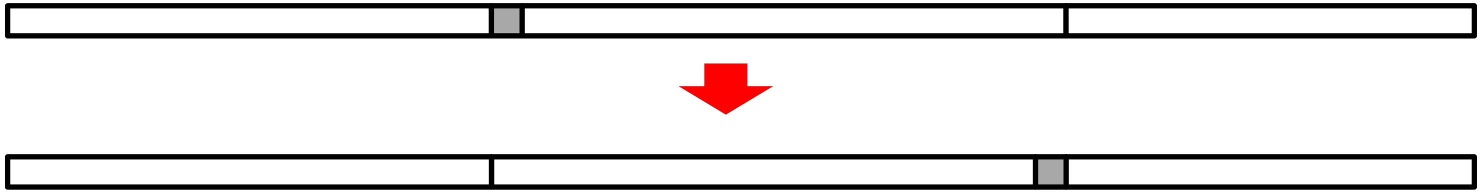
- For SNVs on protein coding regions,
  - Synonymous SNV: SNV that does not change amino acid
    - Since they do not change amino acid, they may be neutral
  - Non-synonymous SNV: SNV that changes amino acid
    - Non-sense SNV: SNV that changes amino acid to stop codon
    - Missense SNV: otherwise
    - They can severely impact the 3D structure and function of the protein

- For SNVs on non-coding regions,
  - Most of them are neutral.
  - Some occur at functional sites like transcription factor binding sites or splice junctions. They affect the gene expression.

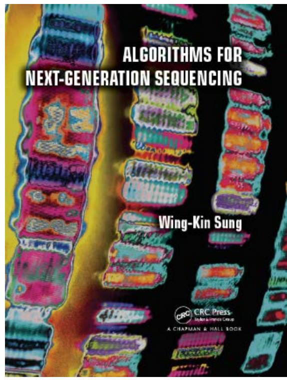


# indels

- Indels is a small insertion or deletion (of size <50bp).
- It is the 2nd most frequent genome variations.
- Each individual expects to have one indel per 3000bp.

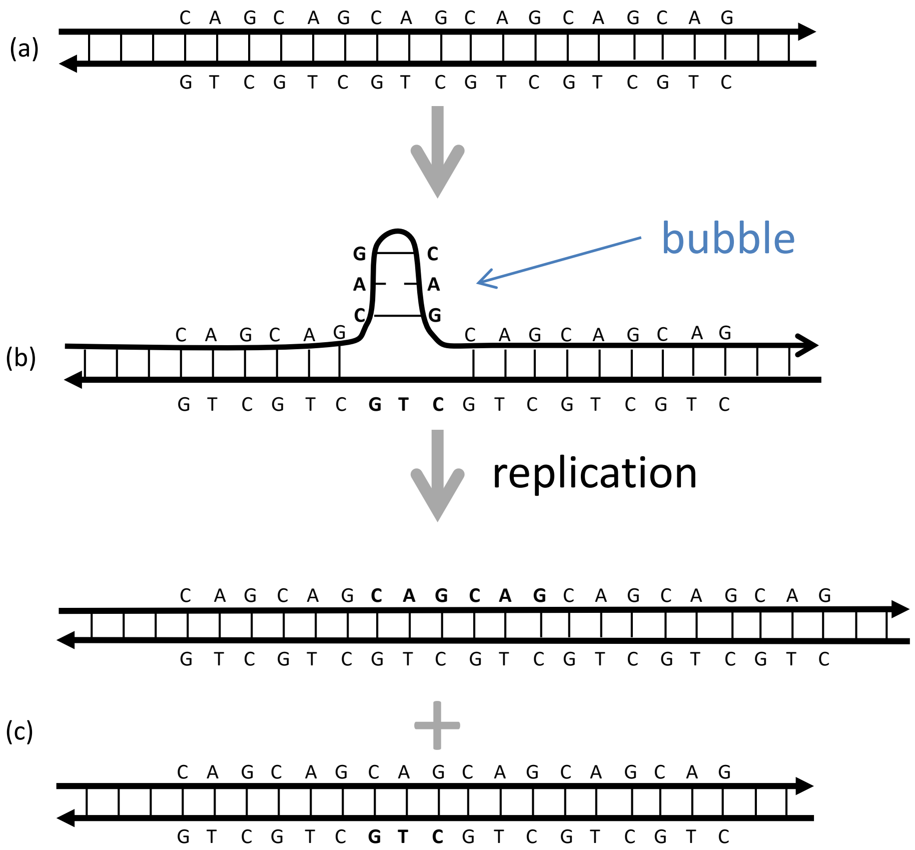


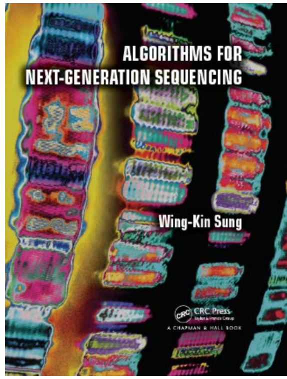
- Most indels are of size 1-20bp (98.5%)
- Most indels (43-48%) are located at 4% of the genome.



# Formation of indels

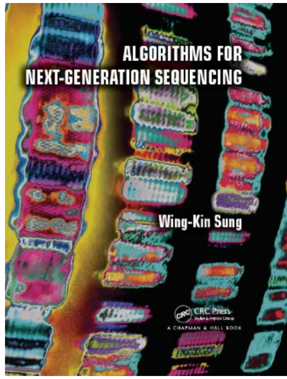
- 75% indels are caused by polymerase slippage.
- It occurs in a section with repeat patterns of bases (like CAG).





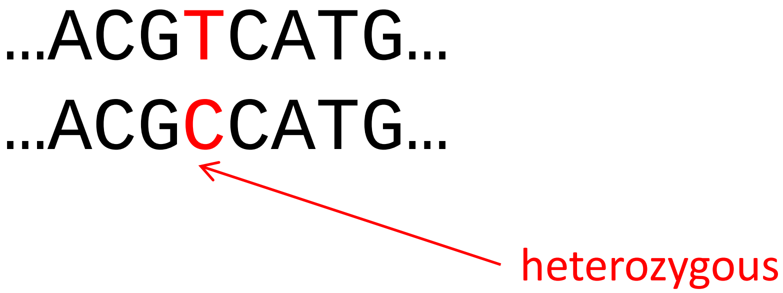
# Effect of indels

- Indels in non-coding regions
  - Mostly neutral
  - If they occur in functional sites like binding site, it may have effect.
- Indels in protein coding regions
  - It will cost frame-shift
  - If indel is multiple of 3, it will cause deletion or insertion of a few codons. It may not affect the property of the gene
  - If indels is not multiple of 3, it will destroy the whole protein.



# Homozygous and Heterozygous

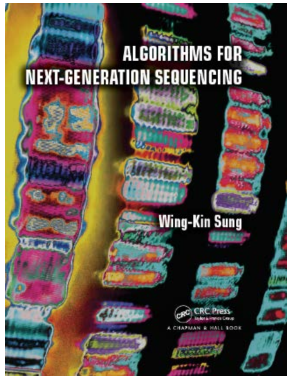
- Human genome is diploid.
- The pair of nucleotides (alleles) appear in a particular position (locus) is its **genotype**.
- If the two alleles at a locus are the same, it is a **homozygous** genotype; otherwise, it is a **heterozygous** genotype.



…ACGTCATG…

…ACGCCATG…

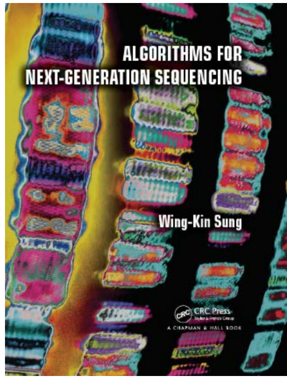
heterozygous



# SNV/indel versus phenotype

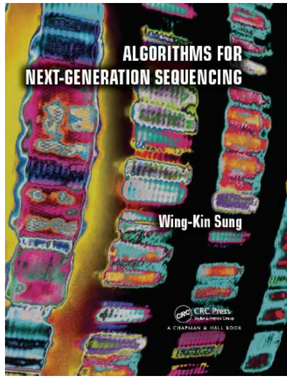
- SNV/indel are related to a number of diseases:
  - SNVs in TP53 and CTNNB1 are recurrently associated with HCC (liver cancer)
  - Indels appear in microsatellites have been linked to >40 neurological diseases
  - Deletion of intron 2 of the BIM gene is associated with the resistance to tyrosine kinase inhibitors in CML patients

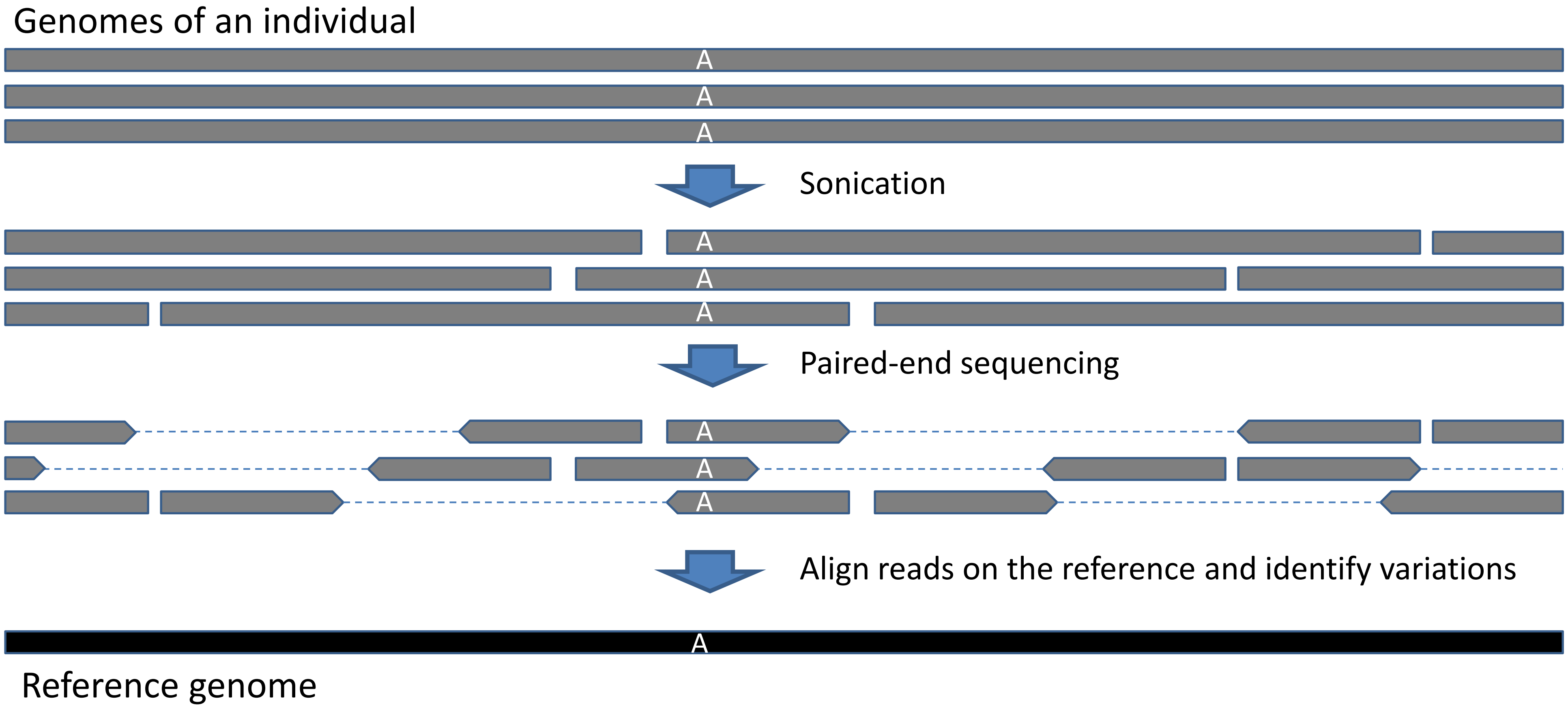
# Somatic and germline mutations

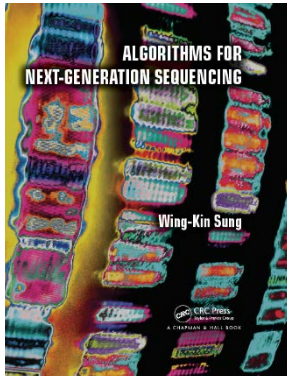


- Germline mutations
  - Mutations that are transmitted from parents to offspring.
  - These mutations present in every cell of an individual.
- Somatic mutations
  - Mutations that occur in a small group of an individual.
  - These mutations will not pass to his/her children.
  - These mutations may cause diseases like cancer.

# Determine SNVs/indels by resequencing



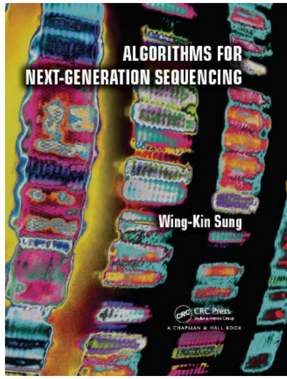




# Target sequencing

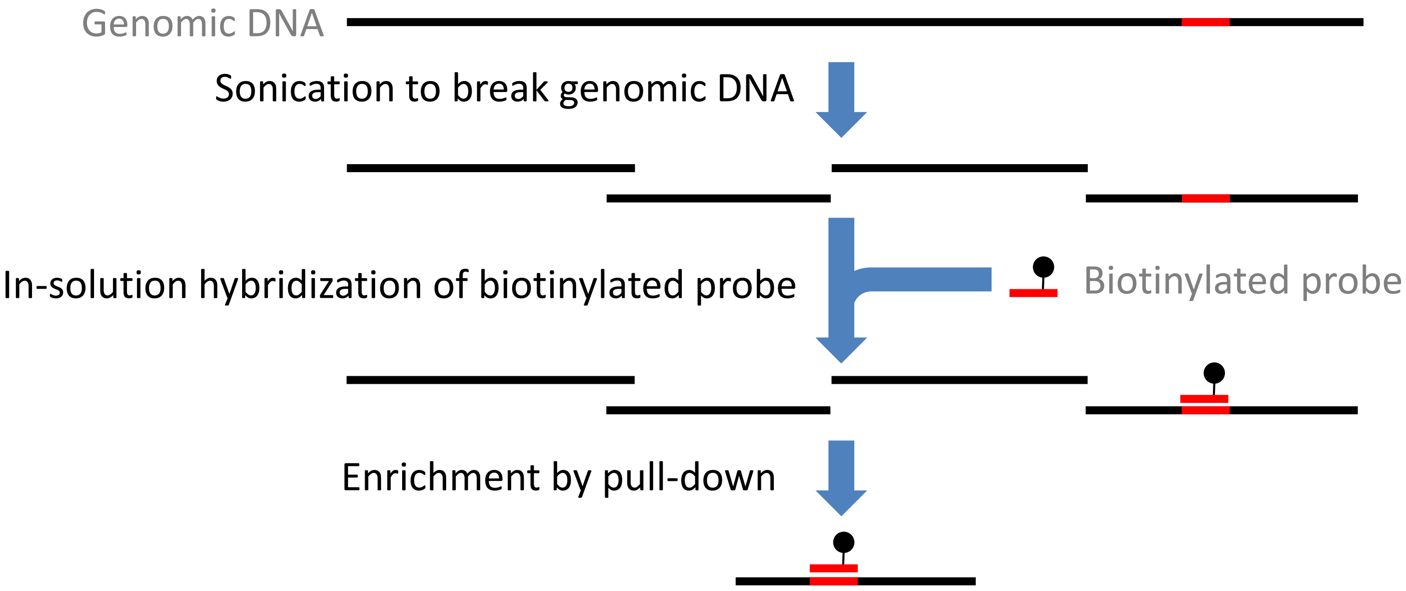
- Most disease related variants are located in protein coding regions (or exons).
- Exons represent <2% of the human genome.

- To reduce cost, we can perform target sequencing:
  - The most popular one is Whole Exome Sequencing (WES)
  - It is cheaper than Whole Genome Sequencing (WGS)



# Target enrichment workflow

- This workflow tries to pull down targeted DNA fragments.



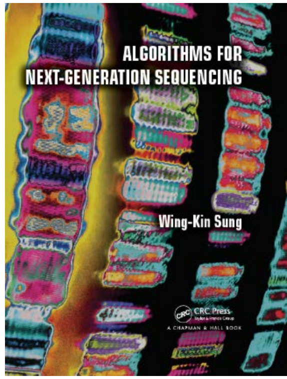
Genomic DNA

Sonication to break genomic DNA

In-solution hybridization of biotinylated probe

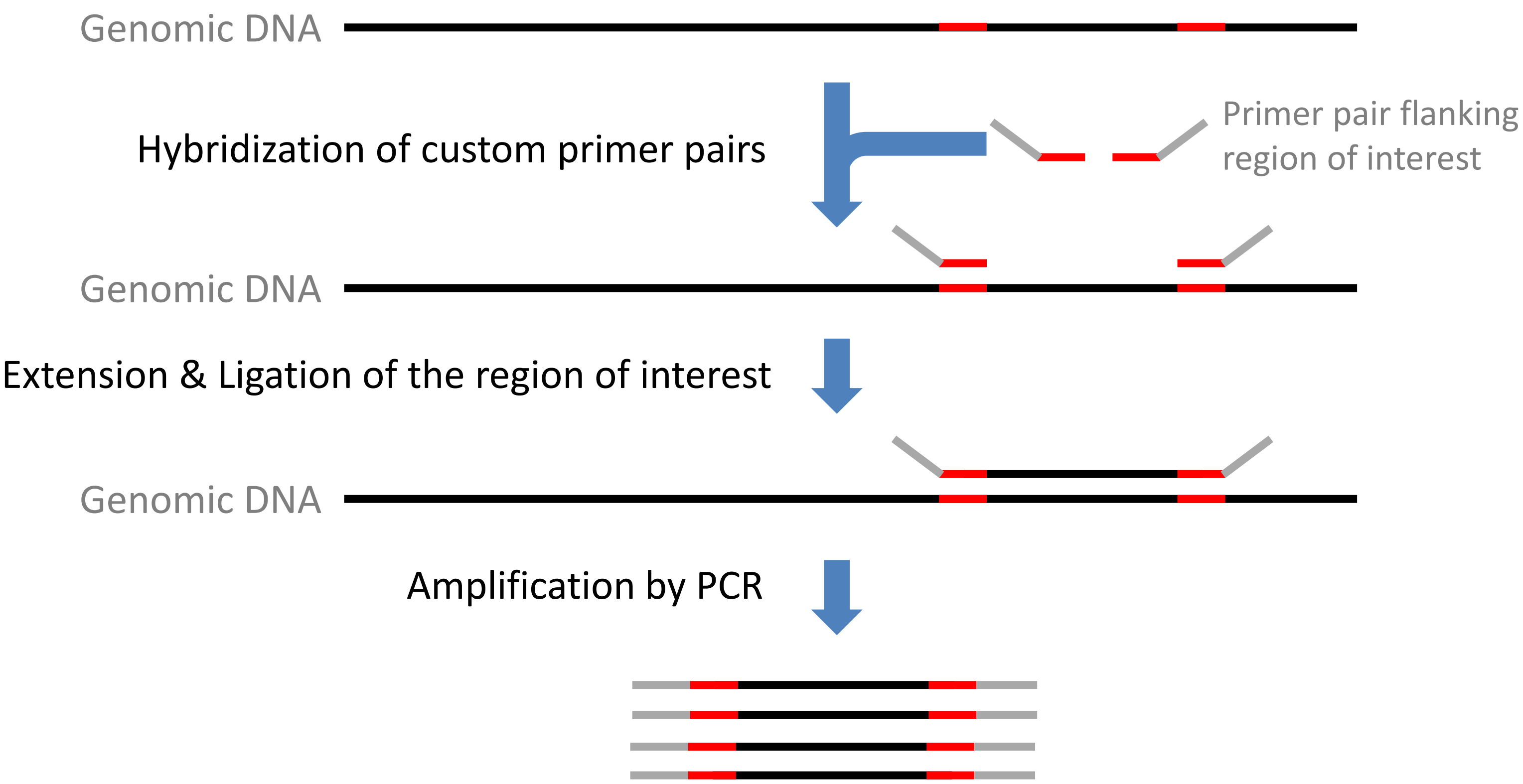
Biotinylated probe

Enrichment by pull-down

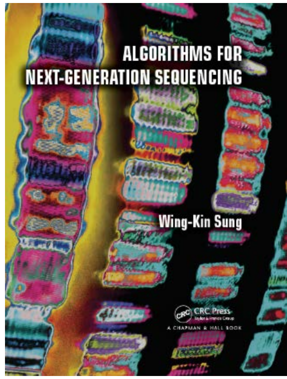


# Amplicon generation workflow

- This workflow amplifies targeted regions.


Genomic DNA
Hybridization of custom primer pairs
Primer pair flanking region of interest
Genomic DNA
Extension & Ligation of the region of interest
Genomic DNA
Amplification by PCR

# VCF format



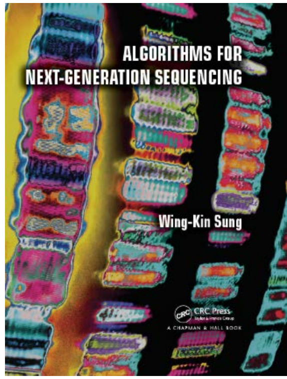
(a)

```
Chr1               1111111111222222  22223333333333444444444455555555556666666666777777
          1234567890123456789012345  67890123456789012345678901234567890123456789012345
 REF:     ACGTACAGACAGACTTAGGACAGAT--CGTCACACTCGGACTGACCGTCACAACGGTCATCACCGGACTTACAATCG

 Sample1:   GTACACACAGAC      CAGATAACGTCAC   CGGACTGACCGTCA AACGGT--------------CAATCG
              ACACACAGACTT
               CACACAGACTTA

 Sample2: ACGTACAGACAG      GACAGATAACGTC    TCGGACT---CG  ACAACGGT--------------CAAT
           CGTACAGACAGA    GGACAGATT-CGT                    CAACGGT--------------CAATC
                          AGGACAGATT-CGT
```

(b)

```
##fileformat=VCFv4.2
##fileDate=20110705
##source=VCFtools
##reference=NCBI36
##ALT=<ID=DEL,Description="Deletion">
##FILTER=<ID=q10,Description="Quality below 10">
##INFO=<ID=SVTYPE,umber=1,Type=String,Description="Type of structural variant">
##INFO=<ID=END,Number=1,Type=Integer,Description="End position of the variant">
##FORMAT=<ID=GQ,Number=1,Type=Integer,Description="Genotype Quality (phred score)">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
##FORMAT=<ID=DP,Number=1,Type=Integer,Description="Read Depth">
#CHROM POS ID REF  ALT    QUAL FILTER INFO              FORMAT Sample1 Sample2
1      8   .  G    C      .    PASS   .                 GT:DP  1/1:3   0/0:2
1      25  .  T    TAA,TT .    q10    .                 GT:DP  1/1:1   1/2:3
1      40  .  TGAC T      .    PASS   .                 GT:GQ  1/1:50  0/0:70
1      55  .  T    <DEL>  .    PASS   SVTYPE=DEL;END=69 GT     1/1     1/1
```

VCF



# Basic SNV calling

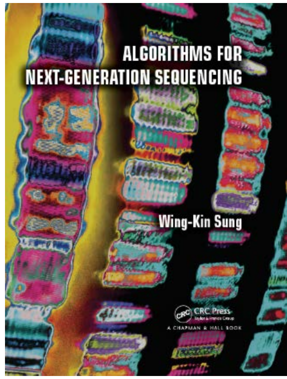
- 1. Align reads 2. Identify a column with variants. 3. Call SNVs

Aligned reads

```
CACGACAC--------------------------------
CACGTCACATAG----------------------------
CACGACACATAGACACCA----------------------
CACGACACATAGACACCATTGAAC----------------
--CGACACATAGACACCATTGAACAC--------------
----ACACATAGACACCATTGAACACGT------------
-----CACATAGACACCATTGAACACGTG-----------
---------TAGACACCATGGAACACGGGGTC--------
-----------GACACCATTGAACACGTGGGTCAC-----
---------------CCATTGAACACGGGGTCACCATA-
-----------------ATTGACCACGTGGGTCACCATAT
---------------------AACACGTGGGTCACCATAT
---------------------------TGGGTCACCATAT
-----------------------------GGTCACCATAT
```

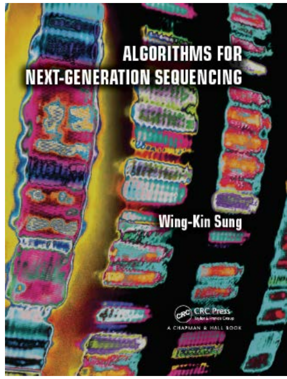
Reference

```
CACGTCACATAGACACCATTGAACACGTGGGTCACCATAT
```



# Different methods for calling SNVs

- SNV calling
  - Counting alleles
  - Binomial distribution
  - Poisson-Binomial model
  - Bayesian approach
  - Posterior odds ratio
- Somatics SNV calling
  - Fisher exact test
  - Probabilistic binomial mixture

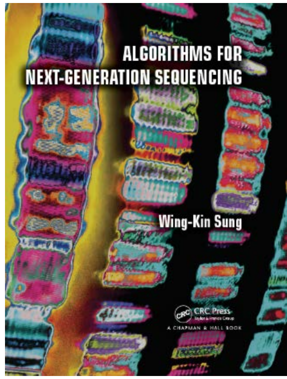


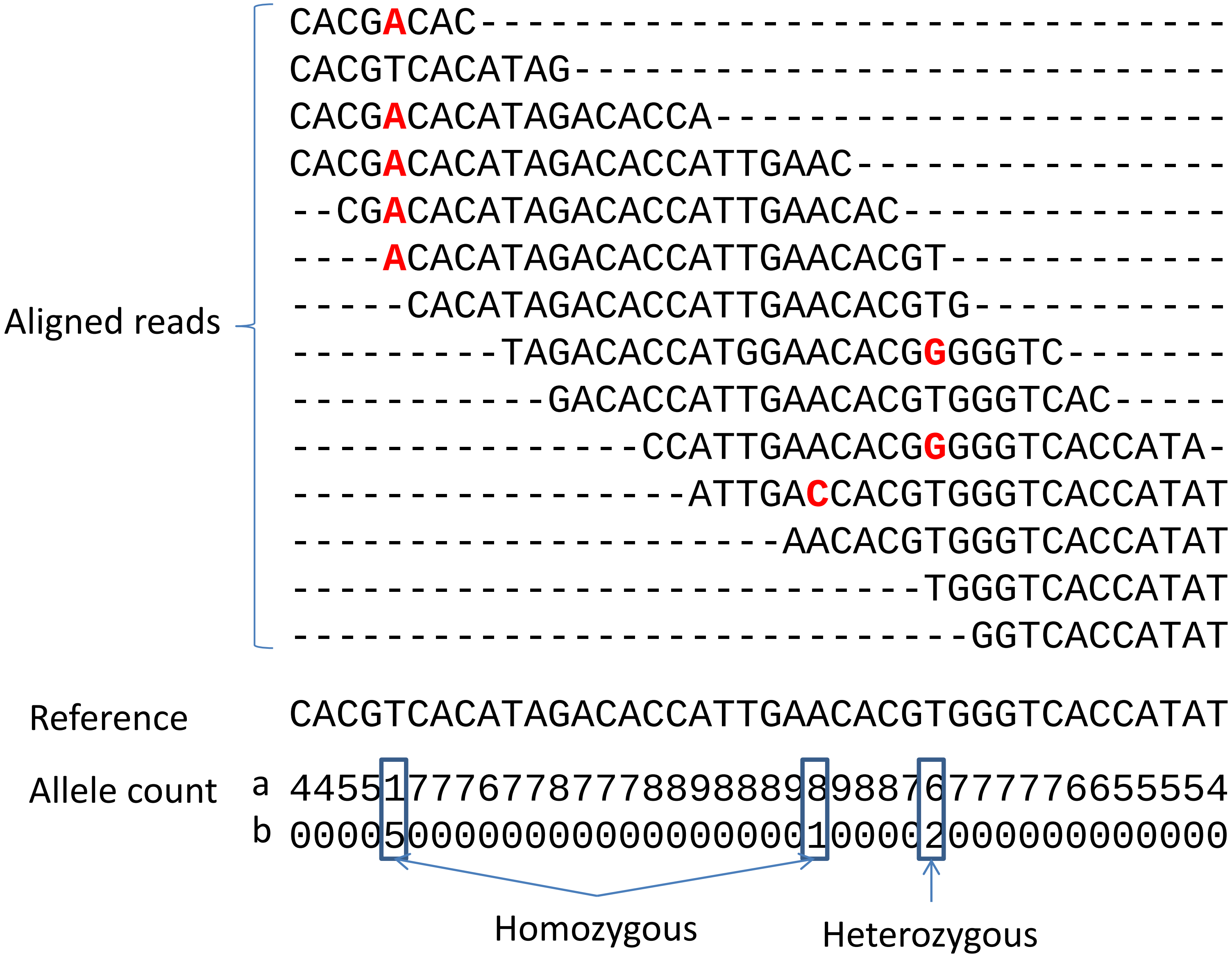
# SNP calling based on counting alleles

1. Keep high-confident bases
   – Usually, keep bases with phred score ≥ 20
2. For each loci, counts the number of occurrences of each allele
3. If the proportion of the non-reference allele is between 20% and 80%, it is called a heterozygous genotype; otherwise, it is called a homozygous genotype.

- This method is used in a number of commercial software including Roche’s GSMapper, the CLC Genomic Workbench and the DNSTAR Lasergene.

# Example





Aligned reads

```
CACGACAC--------------------------------
CACGTCACATAG----------------------------
CACGACACATAGACACCA----------------------
CACGACACATAGACACCATTGAAC----------------
--CGACACATAGACACCATTGAACAC--------------
----ACACATAGACACCATTGAACACGT------------
-----CACATAGACACCATTGAACACGTG-----------
---------TAGACACCATGGAACACGGGGTC-------
-----------GACACCATTGAACACGTGGGTCAC-----
---------------CCATTGAACACGGGGGTCACCATA-
-----------------ATTGACCACGTGGGTCACCATAT
---------------------AACACGTGGGTCACCATAT
---------------------------TGGGTCACCATAT
-----------------------------GGTCACCATAT
```

Reference

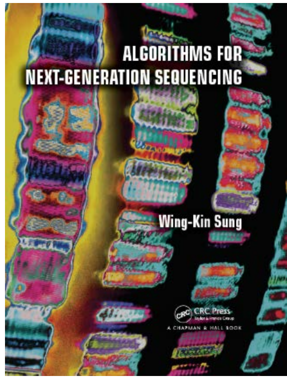
```
CACGTCACATAGACACCATTGAACACGTGGGTCACCATAT
```

Allele count

```
a 4455177767787778898889898876777776655554
b 0000500000000000000000100002000000000000
```

Homozygous

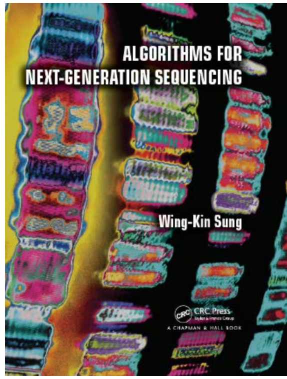
Heterozygous



# Goodness of the simple approach

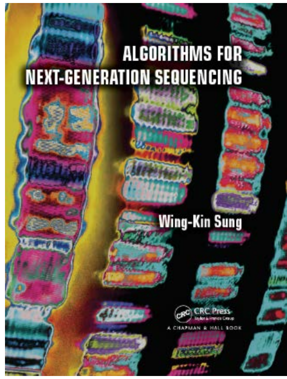
- This method works fairly well when the sequencing depth is high (> 20x).

- Limitations:
  - Simple quality score cutoff may lead to loss of information.
  - This approach cannot provide measures of uncertainty.
  - It may under-call heterozygous genotypes.



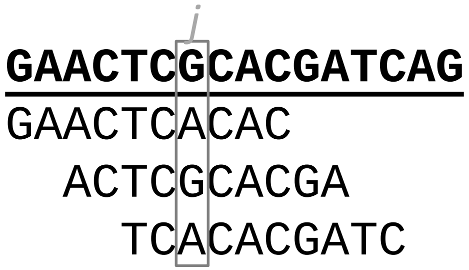
# Binomial distribution

- Simple counting does not give p-value.
- To determine uncertainty, we can use binomial distribution.
- Let D={$b_1$, …, $b_n$} be the set of bases covering a particular locus.
- $H_0$ (null model): All non-reference bases are generated by sequencing error.
  - (Assume p (say 0.01) is the chance of sequencing error)
- $H_1$: The non-reference bases are real variant.

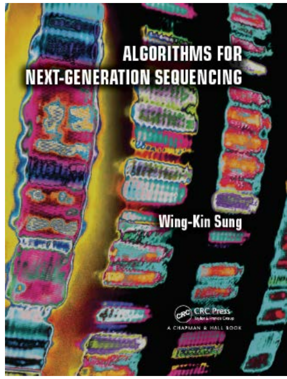


# Binomial distribution

- Null model: There is no SNV. (Assume p is the sequencing error rate.)
- Denote $\Pr_n(X=k)$ be the probability of observing k non-reference variant among n bases under null model. Under binomial distribution, we have: $Pr_n(X = k) = \binom{n}{k} p^k (1-p)^{n-k}$
- In the example below, D={A, G, A}. A is the non-reference variant which occurs twice.
- Suppose the sequencing error rate is p=0.01.
- The chance of observing two non-reference variant is
  - $Pr_n(X \geq 2) = \binom{3}{2}(0.01)^2(1-0.01)^1 + \binom{3}{3}(0.01)^3 = 0.000298.$
- With p-value threshold 0.05, we reject the null model.

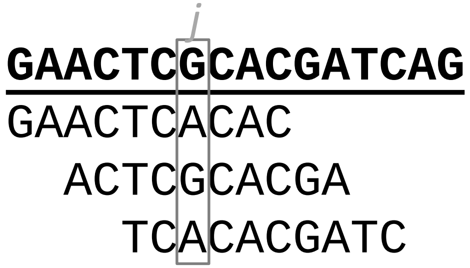


```
GAACTCGCACGATCAG
GAACTCACAC
   ACTCGCACGA
      TCACACGATC
```



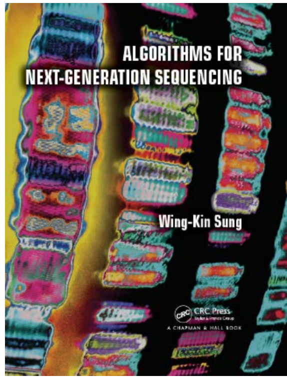
# Poisson-binomial model

- Previous solution assume sequencing error is the same for every called base.
- We can estimate the sequencing error using the quality score per base.
- Consider a base $b_i$ with quality score $q_i$.
- The error probability $e_i = 10^{-\frac{q_i}{10}}$.



```
GAACTCGCACGATCAG
GAACTCACAC
  ACTCGCACGA
    TCACACGATC
```

| Base $b_i$ | Qscore $q_i$ | Err prob $e_i$ |
|---|---|---|
| A | 20 | $10^{-2}$ |
| G | 10 | $10^{-1}$ |
| A | 50 | $10^{-5}$ |

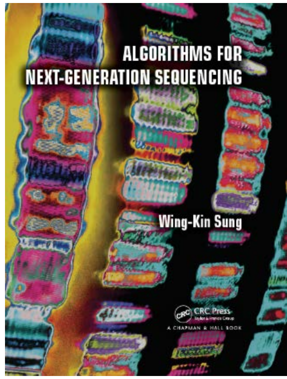


# Poisson-binomial model

- With the error probability {$e_1$, …, $e_n$}, we can compute $Pr_n(X = k)$ as follows.

$$Pr_n(X = k) = \sum_{b_1 \ldots b_n} \left\{ \left( \prod_{b_i = r} (1 - e_i) \right) \left( \prod_{b_i \neq r} e_i \right) \mid \text{the number of } (b_i \neq r) \text{ is } k \right\}$$

- If $Pr_n(X \geq k)$ is smaller than the p-value threshold, we reject the null model.



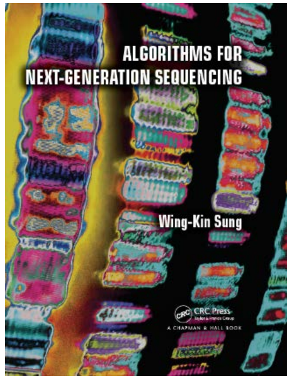
# Poisson-binomial model

- For the previous example, $Pr_n(X \geq 2) = 0.00100108$. We reject the null model.

$$
\begin{aligned}
Pr_3(X=0) &= (1-e_1)(1-e_2)(1-e_3) = 0.89099109 \\
Pr_3(X=1) &= (e_1)(1-e_2)(1-e_3) + (1-e_1)(e_2)(1-e_3) + (1-e_1)(1-e_2)(e_3) \\
&= 0.10800783 \\
Pr_3(X=2) &= (1-e_1)(e_2)(e_3) + (e_1)(1-e_2)(e_3) + (e_1)(e_2)(1-e_3) \\
&= 0.00100107 \\
Pr_3(X=3) &= (e_1)(e_2)(e_3) = 0.00000001
\end{aligned}
$$

```
      i
GAACTCGCACGATCAG
GAACTCACAC
  ACTCGCACGA
    TCACACGATC
```

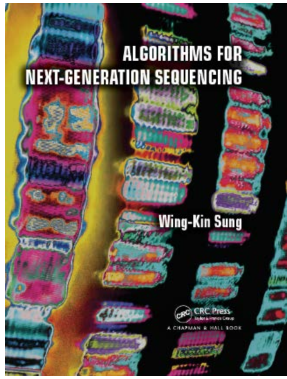
| Base $b_i$ | Qscore $q_i$ | Err prob $e_i$ |
|---|---|---|
| A | 20 | $10^{-2}$ |
| G | 10 | $10^{-1}$ |
| A | 50 | $10^{-5}$ |



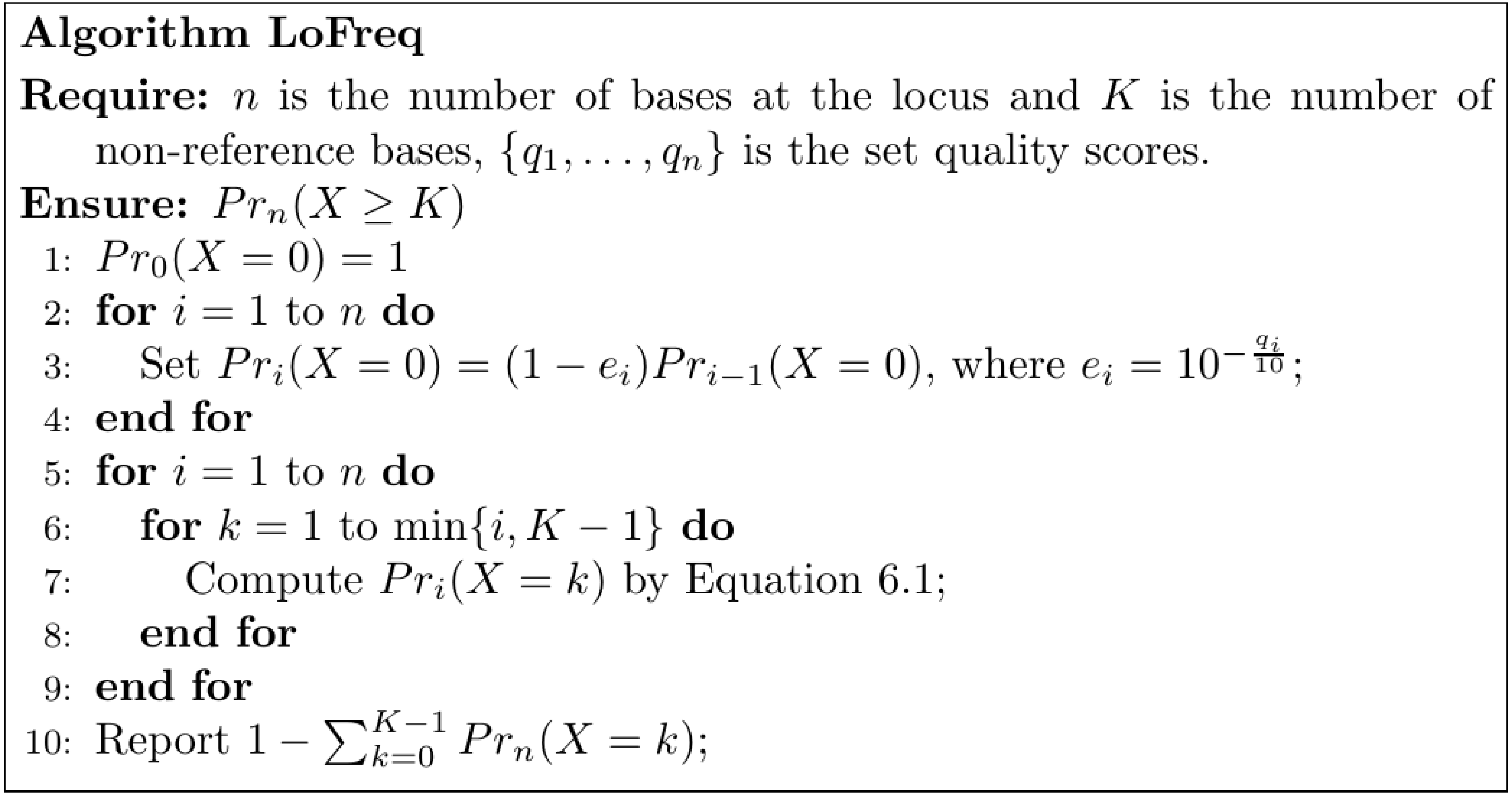
# How to compute $Pr_n(X \geq K)$?

- LoFreq proposed a dynamic programming solution.
- When k=0, n=0 (base case),
  - $Pr_n(X = 0) = 1$
- When k=0, n>0 (recursive case),
  - $Pr_n(X = 0) = (1 - e_n)Pr_{n-1}(X = 0)$
- When k>0 (recursive case),
  - $Pr_n(X = k) = (1 - e_n)Pr_{n-1}(X = k) + e_n Pr_n(X = k - 1)$

- By the above recursive equation, we have an O(Kn) time algorithm for computing $Pr_n(X \geq K)$.



# Algorithm for computing $Pr_n(X \geq K)$

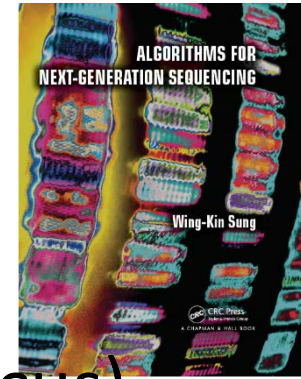


**Algorithm LoFreq**

**Require:** $n$ is the number of bases at the locus and $K$ is the number of non-reference bases, $\{q_1, \ldots, q_n\}$ is the set quality scores.

**Ensure:** $Pr_n(X \geq K)$

```
$Pr_0(X = 0) = 1$
for $i = 1$ to $n$ do
    Set $Pr_i(X = 0) = (1 - e_i)Pr_{i-1}(X = 0)$, where $e_i = 10^{-\frac{q_i}{10}}$;
end for
for $i = 1$ to $n$ do
    for $k = 1$ to $\min\{i, K - 1\}$ do
        Compute $Pr_i(X = k)$ by Equation 6.1;
    end for
end for
Report $1 - \sum_{k=0}^{K-1} Pr_n(X = k)$;
```

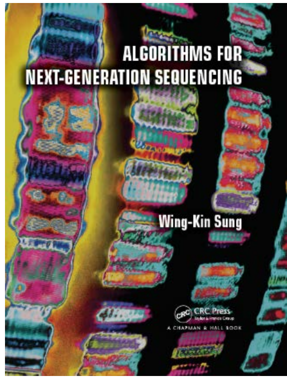


# Bayesian approach

- D represents the observed data (i.e. the bases at a particular locus)
- G represents the genotype at the locus.
  - (There are 10 possible genotypes: AA, CC, GG, TT, AC, AG, AT, CG, CT, GT)

- Let D={$b_1$, …, $b_d$} and G be a genotype $A_1A_2$.
- Our aim to compute Pr(G|D).
- Then, we report the genotype G that maximizes Pr(G|D).

- By Bayesian, Pr(G|D) $\propto$ Pr(G) Pr(D|G).



# Posterior Probability Pr(D|G)

- Since the read bases pileup at the reference position are independent,

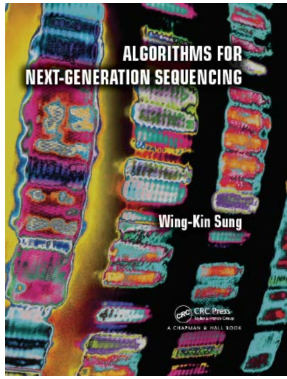
$$\Pr(D \mid G) = \prod_{b_i \in D} \Pr(b_i \mid G)$$

- Assume G=$A_1A_2$, Pr($b_i$|G) can be computed as follows.

$$\Pr(b_i \mid G) = \Pr(b_i \mid A_1A_2) = \tfrac{1}{2}\left(\Pr(b_i \mid A_1) + \Pr(b_i \mid A_2)\right)$$

$$\Pr(b_i \mid A_j) = \begin{cases} 1 - e_i & \text{if } b_i = A_j \\ e_i / 3 & \text{otherwise} \end{cases}$$

where $e_i = 10^{-\frac{q_i}{10}}$ is the error probability and $q_i$ is the Phred score of the base $b_i$.



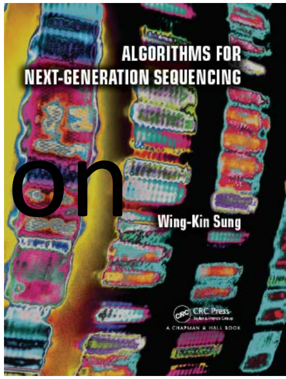
# Prior probability Pr(G)

- There are 10 possible genotypes G.
- The prior probability Pr(G) is influenced by its identity as a homozygous reference, heterozygous, or homozygous non-reference genotype.

- Let r be the reference and s be the alternative allele.
  - Typically, we set
    - Homozygous SNP rate = altHOM = 0.0005
    - Heterozygous SNP rate = altHET = 0.001
- (For example, r=G and s=A.)

| | A | C | G | T |
|---|---|---|---|---|
| A | 0.0005 | 0 | 0.001 | 0 |
| C | | 0 | 0 | 0 |
| G | | | 0.9985 | 0 |
| T | | | | 0 |

# Estimate prior with extra biological information

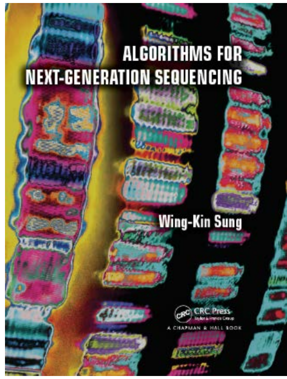


- Many methods use extra biological information to improve the estimation of Pr(G).

- For example, we can use Ti/Tv ratio and dbSNP.

- Transition (Ti):
  - purine<->purine (A <-> G)
  - pyrimidine pyrimidine<-> pyrimidine pyrimidine (C <-> T)
- Transversion (Tv):
  - purine <-> pyrimidine (A <->C, A<->T, G<->C and G<->T)
- Transition is more frequent than transversion

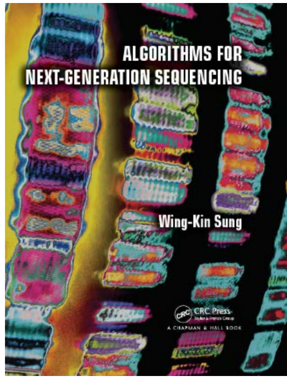
- dbSNP is the a database of known SNVs.



# Prior probability in SOAPsnp

- For example, in SOAPsnp, for non-dbSNP position
- Assume
  - heterozygous SNP rate 0.001, homozygous SNP rate 0.0005
  - Reference allele:G
  - Transition/transversion ratio 4
- Note: A is transition of G; C and T are transversion of G

| | A | C | G | T |
|---|---|---|---|---|
| A | 3.33E-04 | 1.11E-07 | 6.67E-04 | 1.11E-07 |
| C | | 8.33E-05 | 1.67E-04 | 2.78E-08 |
| G | | | 0.9985 | 1.67E-04 |
| T | | | | 8.33E-05 |

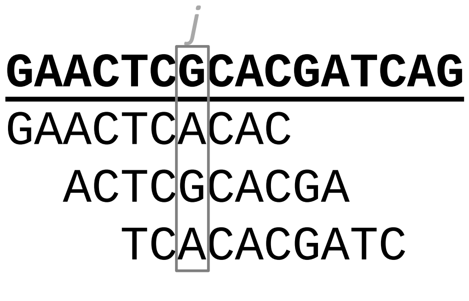


# Example

- $Pr(b_1=A|AG)=1/2(Pr(b_1=A|A)+Pr(b_1=A|G))=1/2((1-10^{-2})+10^{-2}/3)=0.49667$
- $Pr(b_2=G|AG)=1/2(Pr(b_2=G|A)+Pr(b_2=G|G))=1/2(10^{-1}/3+(1-10^{-1}))=0.466667$
- $Pr(b_3=A|AG)=1/2(Pr(b_3=A|A)+Pr(b_3=A|G))=1/2((1-10^{-5})+10^{-5}/3)=0.499997$
- $Pr(D|AG) = 0.49667*0.466667*0.499997 = 0.115888$
- $Pr(AG|D) = Pr(D|AG)*Pr(AG)=0.000116$

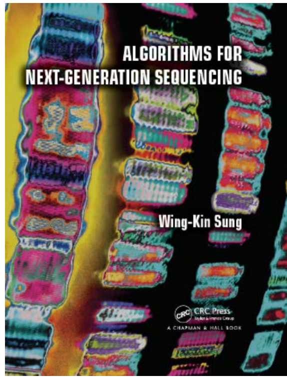
- Hence, we predict the genotype is AG.

| Base $b_i$ | Qscore $q_i$ | Err prob $e_i$ |
|---|---|---|
| A | 20 | $10^{-2}$ |
| G | 10 | $10^{-1}$ |
| A | 50 | $10^{-5}$ |



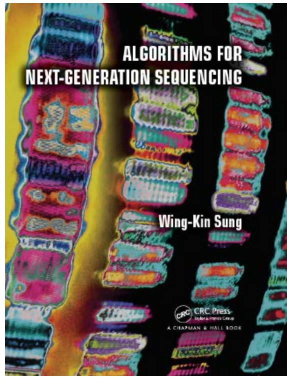
```
GAACTCGCACGATCAG
GAACTCACAC
   ACTCGCACGA
      TCACACGATC
```

| $b_i$ | AG | AA | GG | other |
|---|---|---|---|---|
| A | 0.496667 | 0.99 | 0.003333333 | 0.003333 |
| G | 0.466667 | 0.033333 | 0.9 | 0.033333 |
| A | 0.499997 | 0.99999 | 0.333333 | $3.33x10^{-6}$ |
| Pr(D\|G) | 0.115888 | 0.033 | 0.00000001 | $3.7x10^{-10}$ |
| Pr(G\|D) | 0.000116 | 1.65E-05 | 9.985E-09 | 0 |



# Somatic and germline SNVs detection

- Given the tumor and normal tissue of the same patients.
- Somatic SNVs are SNVs that appear in tumor but not normal.
- Germline SNVs are SNVs that appear in both tumor and non-tumor while they are different from reference.

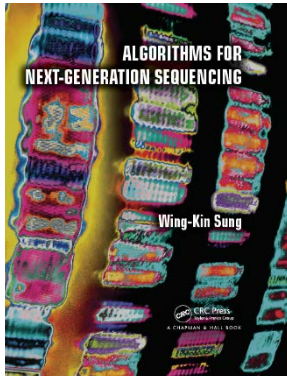


# Somatic SNV detection

- Input: sequencing data from Tumor and Normal
- Output: Somatic SNVs

- Simple method:
  - Identify SNVs from tumor sample
  - Identify SNVs from normal sample
  - Report SNVs appear in tumor but not normal.

- Better methods: MuTect, VarScan2

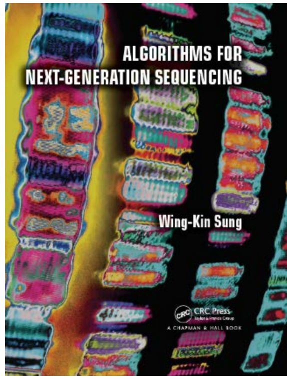


# VarScan2: Fisher exact test

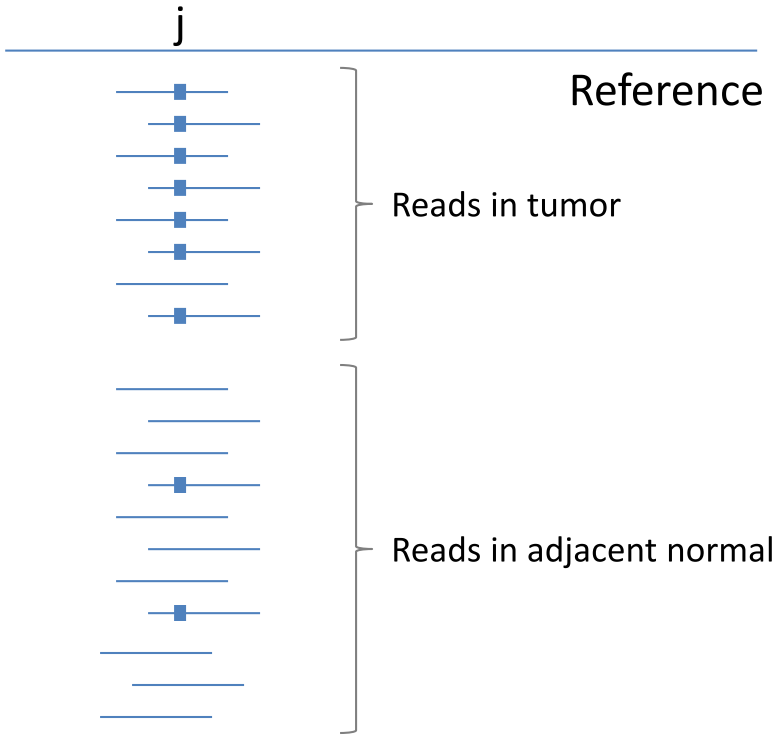
- Use fisher exact test in the following 2-by-2 table.
- If p-value < 0.1 (default),
  - The variant is called somatic (if normal match reference)
  - It is called LOH (if the normal is heterozygous)
- Otherwise, it is a germline variant.

| | # of variant supporting reads | # of reference supporting reads |
|---|---|---|
| **tumor** | a | b |
| **normal** | c | d |

# Somatic SNV calling by Fisher exact test

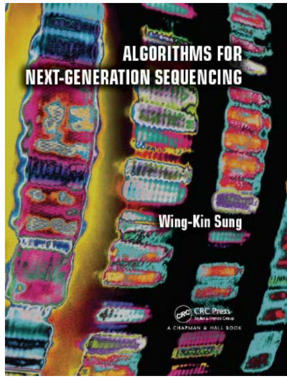


- To test if the SNV appear more in tumor, we can use Fisher exact test.
- If p-value $= \sum_{i=0}^{c_{t,r}} \frac{\binom{c_t}{x}\binom{c_n}{c_r-x}}{\binom{c_t+c_n}{c_r}} < \theta$, reference allele is under-represented in tumor.
- If locus j in normal is a homozygous reference, then it is a somatic SNV.
- If locus j in normal is heterozygous, then it is an LOH (Loss Of Heterozygosity).
- Otherwise, locus j is a germline variant.



- For the example, p-value = 0.0049. It is a somatic SNV.

| | REF allele | ALT allele | Total |
|---|---|---|---|
| Tumor | $c_{t,r}$=1 | 7 | $c_t$=8 |
| Normal | 9 | 2 | $c_n$=11 |
| Total | $c_r$=10 | $c_m$=9 | 19 |

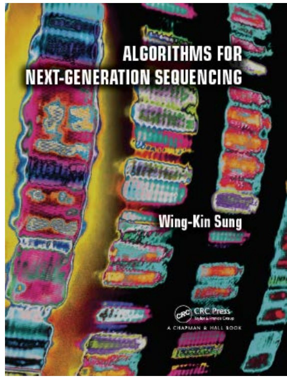


# Somatic variant caller --- MuTect

- Consider a locus j whose reference base is r.
- Input: $D_T = \{b_1, \ldots, b_n\}$ and $D_N = \{b'_1, \ldots, b'_{n'}\}$.

- Two steps:
- 1. Check if locus j is a SNV in tumor.
- 2. Verify if locus j is somatic SNV.

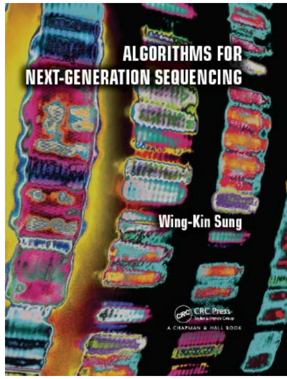
# Step 1: Call SNV in Tumor sample



- Input: $D_T = \{b_1, \ldots, b_n\}$
- We explain the data using two models.
  - $M_0$: There is no variant at this locus. The observed non-reference bases are due to random sequencing errors.
  - $M_f^m$: A variant m exists; the frequency of m is f.
- Note: $M_0 = M_0^m$.
- $L(M_f^m | D_T) = \prod_{i=1}^{n} \Pr(b_i | M_f^m) = \prod_{i=1}^{n} \Pr(b_i | e_i, r, m, f)$

$$\Pr(b_i \mid e_i, r, m, f) = \begin{cases} f\dfrac{e_i}{3} + (1-f)(1-e_i) & \text{if } b_i = r \\ f(1-e_i) + (1-f)\dfrac{e_i}{3} & \text{if } b_i = m \\ \dfrac{e_i}{3} & \text{if } b_i \neq r, m \end{cases}$$

# Step 1: Call SNV in Tumor sample



- Variant is detected by their ratio.
- We declare m as a candidate variant if

$$\max_f \frac{P(m,f)L\left(M_f^m|D_T\right)}{\left(1-P(m,f)\right)L\left(M_0|D_T\right)} \geq \delta$$

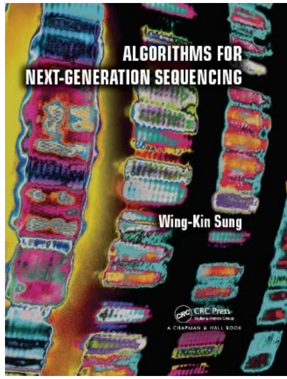
- where $\delta$ is set to be 2.

- P(m,f)

| | | |
|---|---|---|
| P(m,f) | =P(m)P(f) | [assume they are independent] |
| | =P(m) | [assume P(f) is uniformly distributed] |
| | $=\frac{1}{3}$E(mutation frequency) | |
| | $=10^{-6}$ | [somatic mutation frequency $\approx 3\text{x}10^{-6}$] |

- Hence, we declare m as a candidate variant if

$$\max_f LOD(m,f) = \max_f \left(\frac{L(M_f^m|D_T)}{L(M_0|D_T)}\right) \geq \log_{10}\left(\frac{1-10^{-6}}{10^{-6}}\delta_T\right) \approx 6.3$$



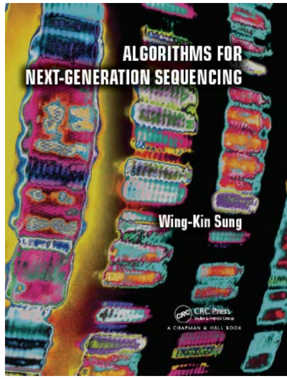
# What is f?

- It is time consuming to find f that maximize LOD(m,f).

- In MuTect, it estimates f to be

$$\hat{f} = \frac{\text{number of mutant reads}}{\text{total number of reads}}$$

# Example



| | $f = \frac{2}{3}$ | $f = 0$ |
|---|---|---|
| $\Pr(b_1 = A \mid e_1 = 10^{-2}, r = G, m = A, f)$ | 0.661111 | 0.003333 |
| $\Pr(b_2 = G \mid e_2 = 10^{-1}, r = G, m = A, f)$ | 0.322222 | 0.9 |
| $\Pr(b_3 = A \mid e_3 = 10^{-5}, r = G, m = A, f)$ | 0.666661 | 3.33x$10^{-6}$ |

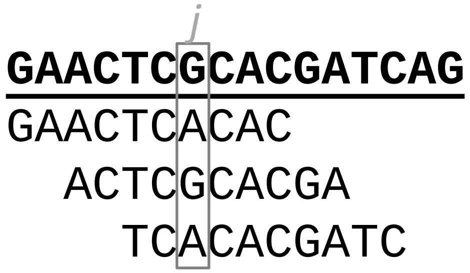
- We set $f = \frac{2}{3}$.

$$Pr(b_1 = \mathtt{A} \mid e_1 = 10^{-2}, r = \mathtt{G}, m = \mathtt{A}, f = \frac{2}{3}) = \frac{2}{3}(1 - 10^{-2}) + (1 - \frac{2}{3})\frac{10^{-2}}{3} = 0.661111$$

$$Pr(b_2 = \mathtt{G} \mid e_2 = 10^{-1}, r = \mathtt{G}, m = \mathtt{A}, f = \frac{2}{3}) = \frac{2}{3}\frac{10^{-1}}{3} + (1 - \frac{2}{3})(1 - 10^{-1}) = 0.322222$$

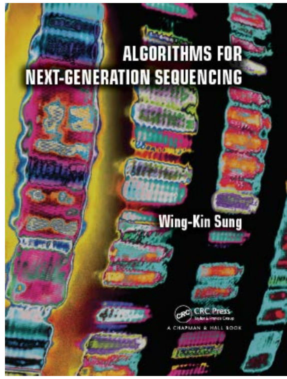
$$Pr(b_3 = \mathtt{A} \mid e_3 = 10^{-5}, r = \mathtt{G}, m = \mathtt{A}, f = \frac{2}{3}) = \frac{2}{3}(1 - 10^{-5}) + (1 - \frac{2}{3})\frac{10^{-5}}{3} = 0.666661$$

- We have
  - $\Pr(D \mid M_f^A)$ = 0.661111*0.32222*0.666661=0.142015
  - $\Pr(D \mid M_0)$ = 0.003333*0.9*3.33x$10^{-6}$ = 1x$10^{-8}$



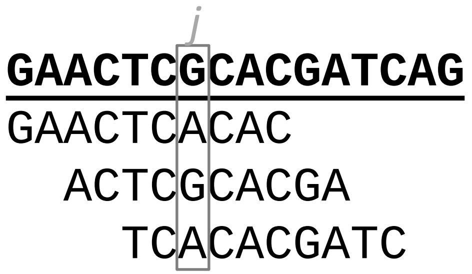
```
GAACTCGCACGATCAG
GAACTCACAC
  ACTCGCACGA
    TCACACGATC
```

| Base $b_i$ | Qscore $q_i$ | Err prob $e_i$ |
|---|---|---|
| A | 20 | $10^{-2}$ |
| G | 10 | $10^{-1}$ |
| A | 50 | $10^{-5}$ |



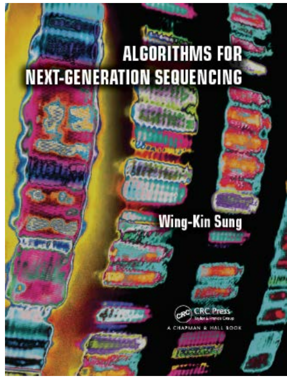
# Example

- We have
  - $\Pr\left(D\middle|M_f^A\right)$ = 0.661111*0.32222*0.666661=0.142015
  - $\Pr(D|M_0)$ = 0.003333*0.9*3.33x$10^{-6}$ = 1x$10^{-8}$
- Then, $LOD\left(m = A, f = \frac{2}{3}\right) = \log_{10}\frac{\Pr\left(D\middle|M_f^A\right)}{\Pr(D|M_0)} = 7.15.$
- Since 7.15 > 6.3, we predict this locus is a SNV.



```
      j
GAACTCGCACGATCAG
GAACTCACAC
    ACTCGCACGA
        TCACACGATC
```

| Base $b_i$ | Qscore $q_i$ | Err prob $e_i$ |
|---|---|---|
| A | 20 | $10^{-2}$ |
| G | 10 | $10^{-1}$ |
| A | 50 | $10^{-5}$ |



# Step 2: Verify if the locus is somatic

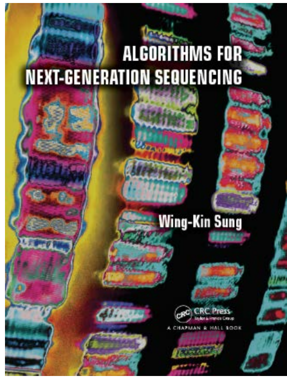
- Given a candidate somatic SNV at locus j, we said it is a somatic SNV if

$$\frac{Pr(\text{locus } j \text{ is reference}|\mathcal{D}_N)}{Pr(\text{locus } j \text{ is mutated}|\mathcal{D}_N)} = \frac{Pr(\text{somatic})L(M_0|\mathcal{D}_N)}{Pr(\text{germline})L(M^m_{0.5}|\mathcal{D}_N)} \geq \delta_N$$

- Otherwise, it is a germline SNV.

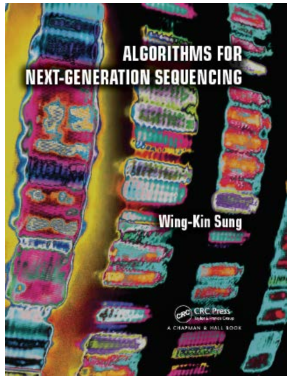
- Since we expect 3 somatic SNVs out of 1 million bases, we set $\Pr(somatic) = 3 \times 10^{-6}$.
- Fact:
  - There are $30 \times 10^6$ dbSNPs.
  - We expect $3 \times 10^6$ SNVs per individual
  - We expect 95% SNVs are in dbSNP position.
- For non-dbSNP, we set $\Pr(germline) = \frac{0.05 \times 3 \times 10^6}{3 \times 10^9} = 5 * 10^{-5}$
- For dbSNP, we set $\Pr(germline) = \frac{0.95 \times 3 \times 10^6}{30 \times 10^6} = 0.095$.

# Step 2: Verify if the locus is somatic



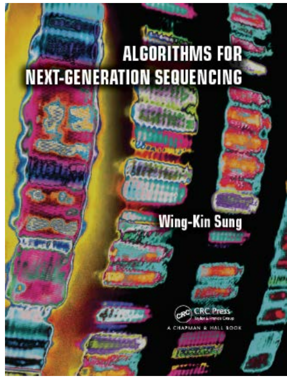
- We set $\delta_N = 10$.
- Let $LOD_N = \frac{L(M_0|D_N)}{L(M_{0.5}^m|D_N)}$.
- Rule:
  - For non-dbSNP, locus j is a somatic SNV if $LOD_N \geq 2.2$.
  - For dbSNP, locus j is a somatic SNV if $LOD_N \geq 5.5$.

# Simple SNV caller gives many false positives



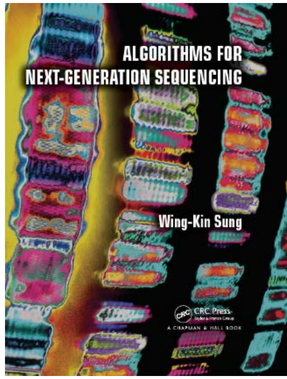
- Reasons:
  - Systematic errors in base calling.
  - Read mapping error.

- A number of techniques are proposed:
  - Base quality score recalibration
    - Used by SOAPsnp, GATK, MuTect
  - Local realignment
    - Used by GATK, MuTect
  - Rule-based filter
    - Used by MAQ, SamTool, GATK, MuTect



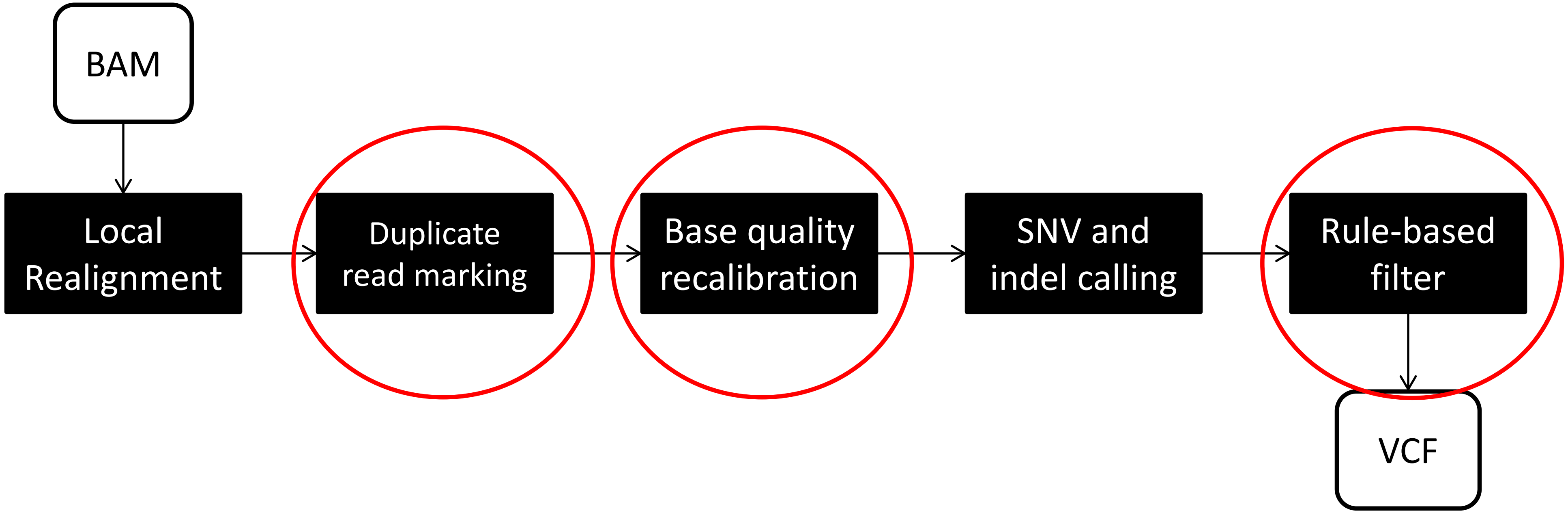
# A more advance SNV caller

- 1. Align the reads
- 2. Realign the reads
- 3. Base quality Recalibration
- 4. SNV calling
- 5. SNV filtering



# A more advance SNV caller

- Input: the alignment file (BAM file)
- Output: a list of SNVs/indels (VCF file)


BAM

Local Realignment

Duplicate read marking

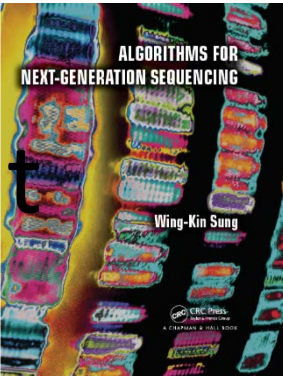
Base quality recalibration

SNV and indel calling

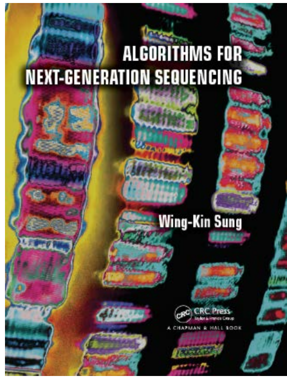
Rule-based filter

VCF

# SNV calling is heavily affected by read alignment

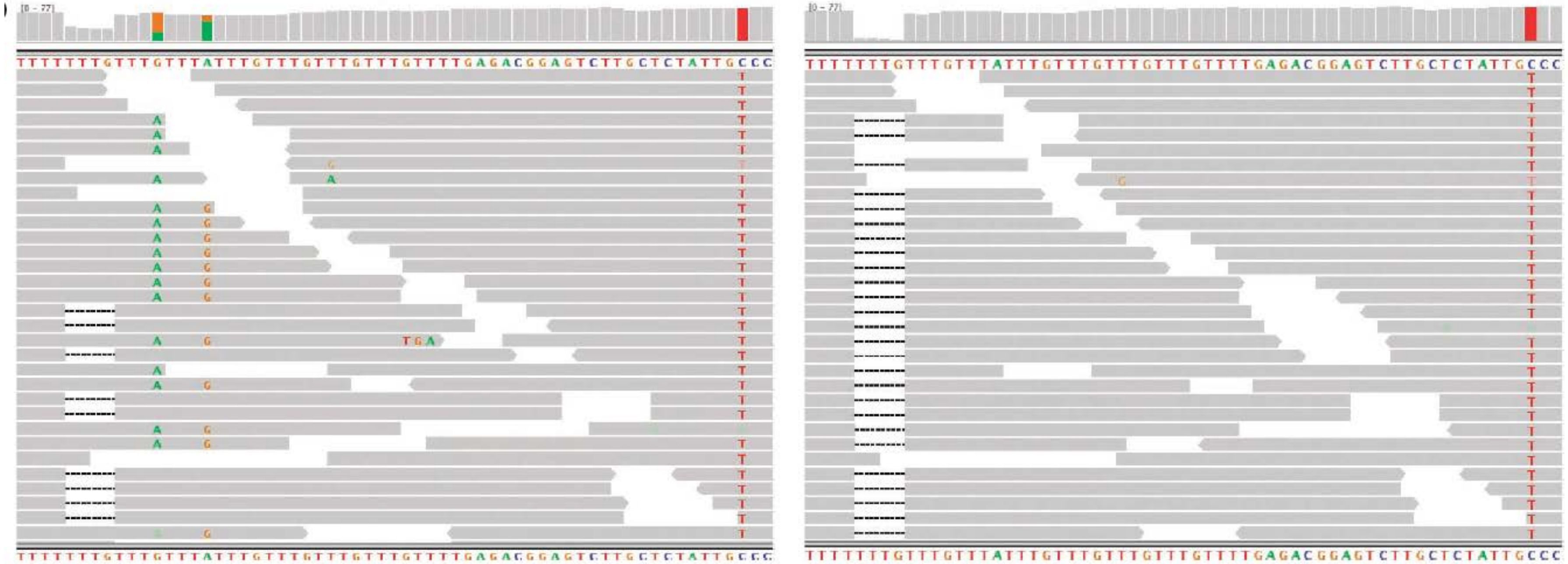


- Read alignment is difficult.
- DePristo et al.(Nature Genetics, 2011) found that nearly two thirds of the differences in SNV calling can be attributed to different read mappings between BWA and MAQ (for HiSeq and exome call sets).



# Local realignment

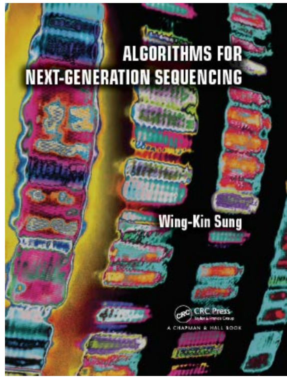
- Read mapping near indels is difficult.
- On the left, there are three SNVs A, G and T.
- After realignment, only SNV C->T is remained.


[0 - 77]
TTTTTTTGTTTGTTTATTTGTTTGTTTGTTTGTTTTGAGACGGAGTCTTGCTCTATTGCCC
A
G
T
TGA
[0 - 77]
TTTTTTTGTTTGTTTATTTGTTTGTTTGTTTGTTTTGAGACGGAGTCTTGCTCTATTGCCC

HiSeq data, raw BWA alignments

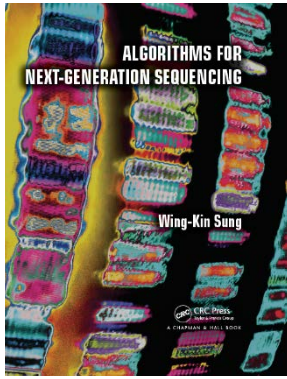
HiSeq data, after MSA

DePristo et al. Nature Genetics, 2011.



# GATK local realignment algorithm (I)

- 1. Find regions that
  - contain at least one read with indel;
  - contain a cluster of mismatch bases; or
  - contain some known indel (e.g. from dbSNP)
- 2. For each region, construct haplotypes
  - from reference sequence and known indel
  - from indels in reads spanning the site
  - from Smith-Waterman alignment of reads that do not perfectly match the reference genome.



# Example

Reference $H_0$: `GTCATCAGCTCACATAGACACCATTGAACACGTGGGTCACCATAT`

Aligned reads

$R_1$: `-TCATCAGCTCACATAGACACTG----------------------`

$R_2$: `-----CAGCTCACATAGACAC___TGAACACG-------------`

$R_3$: `-------GCT__CATAGACACCATTGAACACGT------------`

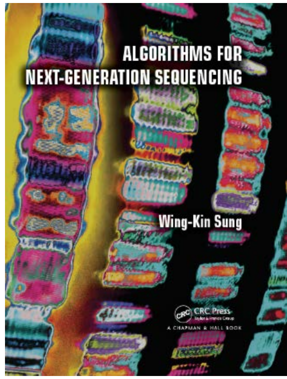
$R_4$: `--------------TAGACAC___TGAACACGTGGGTCACC----`

$R_5$: `--------------------ACACTGAACACGTGGGTCACCATA-`

$R_1$ has a cluster of mismatches while $R_2$ and $R_4$ have a indel.
The set of reads overlap with the indel and the cluster of mismatches is $\{R_1, R_2, R_3, R_4, R_5\}$.

From these reads, we observe two possible deletions (delete CAT and delete CA).
We generate two possible haplotypes:
$H_1$ = TCATCAGCTCACATAGACAC | TGAACACGTGGGTCACCATA.
$H_2$ = TCATCAGCT | CATAGACACCATTGAACACGTGGGTCACCATA.

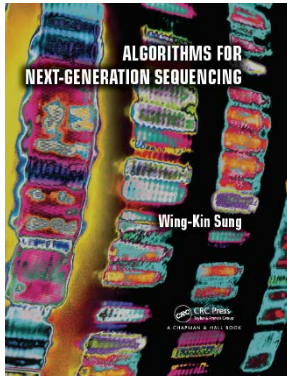
# GATK local realignment algorithm (II)



- 3. For each haplotype $H_i$,
  - Align reads without gaps to $H_i$
  - Suppose $R_1$, …, $R_m$ are aligned to $H_i$.
  - Compute the score $L(H_i)$

- Let $R_j$ be the $j^{th}$ read. Let $R_{j,k}$ be the $k^{th}$ base of read $R_j$.
- Let $\varepsilon_{j,k}$ be the error probability determined from the quality score of the $k^{th}$ base of the read $R_j$.
- $L(R_j|H_i)=\prod_{k=1..|Rj|} L(R_{j,k}|H_{j,i})$
- $L(R_{j,k}|H_{j,i})= (1-\varepsilon_{j,k})$ if $R_{j,k}=H_{j,i}$; and $\varepsilon_{j,k}$ if $R_{j,k}=H_{j,i}$.
- $L(H_i) = \prod_{j=1..m} L(R_j|H_i)$

- 4. Identify the haplotype $H_i$ that maximizes $L(H_i)$



# Example

Assume every base is Q20.

$H_1$ = `TCATCAGCTCACATAGACACTGAACACGTGGGTCACCATA`

Ungapped alignment

$R_1$: `TCATCAGCTCACATAGACACTG------------------` $L(R_1|H_1)=1$
$R_2$: `----CAGCTCACATAGACACTGAACACG------------` $L(R_2|H_1)=1$
$R_3$: `--------GCTCATAGACACCATTGAACACGT--------` $L(R_3|H_1)=10^{-22}$
$R_4$: `-------------TAGACACTGAACACGTGGGTCACC---` $L(R_4|H_1)=1$
$R_5$: `----------------ACACTGAACACGTGGGTCACCATA` $L(R_5|H_1)=1$

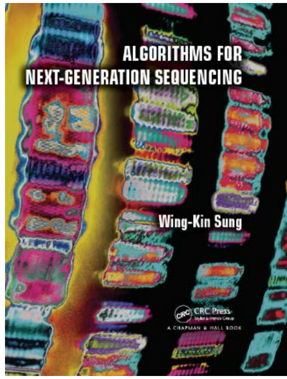
$H_2$ = `TCATCAGCTCATAGACACCATTGAACACGTGGGTCACCATA`

Ungapped alignment

$R_1$: `TCATCAGCTCACATAGACACTG---------------------` $L(R_1|H_2)=10^{-2}$
$R_2$: `----CAGCTCACATAGACACTGAACACG---------------` $L(R_2|H_2)=10^{-22}$
$R_3$: `------GCTCATAGACACCATTGAACACGT-------------` $L(R_3|H_2)=1$
$R_4$: `--------------TAGACACTGAACACGTGGGTCACC-----` $L(R_4|H_2)=10^{-10}$
$R_5$: `-----------------ACACTGAACACGTGGGTCACCATA--` $L(R_5|H_2)=10^{-4}$

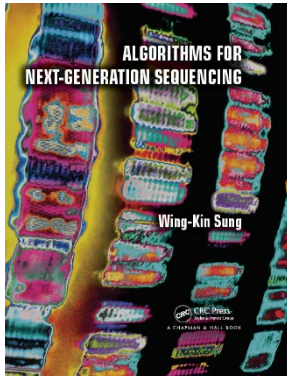
$L(H_1)=L(R_1|H_1)L(R_2|H_1)L(R_3|H_1)L(R_4|H_1)L(R_5|H_1)=10^{-22}$.
$L(H_2)=L(R_1|H_2)L(R_2|H_2)L(R_3|H_2)L(R_4|H_2)L(R_5|H_2)=10^{-38}$.

Since $L(H_1) > L(H_2)$, we select $H_1$.



# GATK local realignment algorithm (III)

- Denote $L(H_0, H_i) = \prod_{j=1..m} \max\{L(R_j \mid H_i), L(R_j \mid H_0)\}$
- 5. Accept $H_i$ if log $(L(H_0,H_i)/L(H_0)) > 5$.



# Example

$H_1$ = `TCATCAGCTCACATAGACACTGAACACGTGGGTCACCATA`

Ungapped alignment

$R_1$: `TCATCAGCTCACATAGACACTG------------------` $L(R_1|H_1)=1$

$R_2$: `----CAGCTCACATAGACACTGAACACG------------` $L(R_2|H_1)=1$

$R_3$: `--------GCTCATAGACACCATTGAACACGT--------` $L(R_3|H_1)=10^{-22}$

$R_4$: `-------------TAGACACTGAACACGTGGGTCACC---` $L(R_4|H_1)=1$

$R_5$: `----------------ACACTGAACACGTGGGTCACCATA` $L(R_5|H_1)=1$

$H_0$ = `TCATCAGCTCACATAGACACCATTGAACACGTGGGTCACCATA`

Ungapped alignment

$R_1$: `TCATCAGCTCACATAGACACTG---------------------` $L(R_1|H_0)=10^{-2}$

$R_2$: `----CAGCTCACATAGACACTGAACACG---------------` $L(R_2|H_0)=10^{-14}$

$R_3$: `--------GCTCATAGACACCATTGAACACGT-----------` $L(R_3|H_0)=10^{-4}$

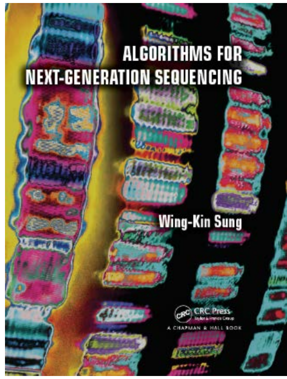
$R_4$: `----------------TAGACACTGAACACGTGGGTCACC---` $L(R_4|H_0)=10^{-10}$

$R_5$: `-------------------ACACTGAACACGTGGGTCACCATA` $L(R_5|H_0)=10^{-4}$

$L(H_0)=L(R_1|H_0)L(R_2|H_0)L(R_3|H_0)L(R_4|H_0)L(R_5|H_0)=10^{-34}$.
$L(H_0,H_1)=L(R_1|H_1)L(R_2|H_1)L(R_3|H_0)L(R_4|H_1)L(R_5|H_1)=10^{-4}$.

Since $\log (L(H_0,H_1)/L(H_0))=30>5$, we accept $H_1$.

# GATK local realignment algorithm (IV)



- 6. Realign every read $R_j$ to $H_i$ if $L(R_j|H_i)>L(R_j|H_0)$.

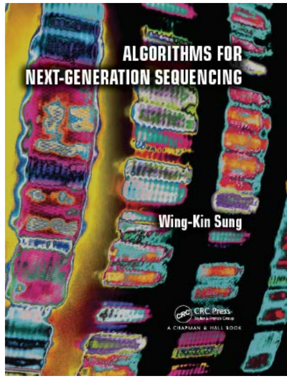
Reference $H_0$:

Aligned reads

```
Reference H0:  GTCATCAGCTCACATAGACACCATTGAACACGTGGGTCACCATAT
         R1 :  -TCATCAGCTCACATAGACACTG----------------------
         R2 :  -----CAGCTCACATAGACAC___TGAACACG-------------
         R3 :  -------GCT__CATAGACACCATTGAACACGT------------
         R4 :  --------------TAGACAC___TGAACACGTGGGTCACC----
         R5 :  --------------------ACACTGAACACGTGGGTCACCATA-
```

Realigned reads

```
Reference H0:  GTCATCAGCTCACATAGACACCATTGAACACGTGGGTCACCATAT
         R1 :  -TCATCAGCTCACATAGACAC___TG-------------------
         R2 :  -----CAGCTCACATAGACAC___TGAACACG-------------
         R3 :  ---------GCTCATAGACACCATTGAACACGT------------
         R4 :  --------------TAGACAC___TGAACACGTGGGTCACC----
         R5 :  -----------------ACAC___TGAACACGTGGGTCACCATA-
```

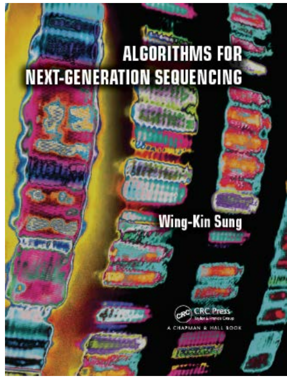


# Marking of duplicate reads

- Due to the PCR amplification step during the NGS library preparation, duplicate reads may generated.
- Those duplicate reads may bias the SNP calling.
- Hence, we need to mark them.

- Method:
  - Merge all lanes.
  - Identify all paired-end reads where the outer ends map to the same position on the genome.
  - Those paired-end reads may be generated by PCR amplification.
    - They may result in false SNP calls.
  - We mark all these reads as duplicates.

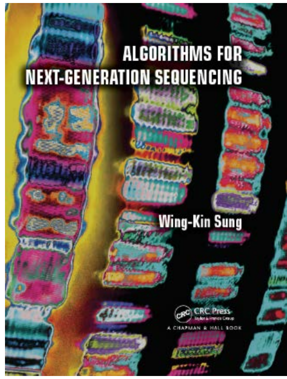
# Quality score



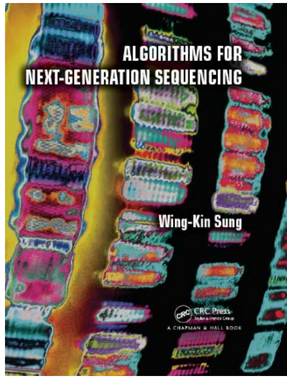
- By definition, if a base with error probability p, its quality score is -10 $\log_{10}$p.
- In previous discussion, we use this score to improve SNV calling.

- However, the inaccuracy and covariation patterns differ strikingly between sequencing technologies.
- We need to recalibrate the quality score.

# Three factors that affect the quality score



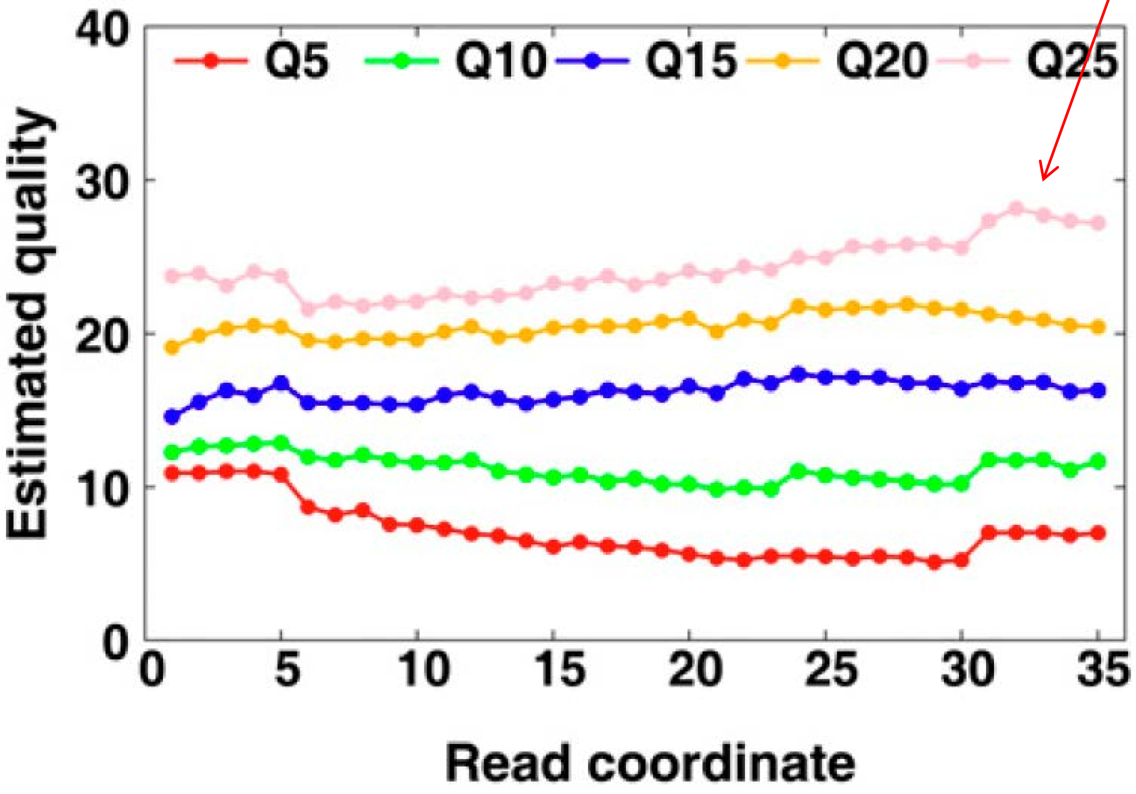
- Position of the base
  - The error rate is different at different position
- Substitution bias
  - Some substitution mismatches (like T→G) are under-represented
- Dinucleotide context
  - G is a likely base before an error



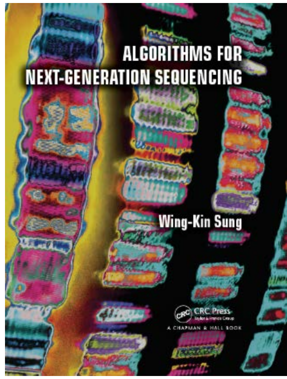
# Cycle effect

Quality is estimated by
- 10 $\log_{10}$(mismatch rate)

Although all are Q25,
the estimate qualities are
different at different positions

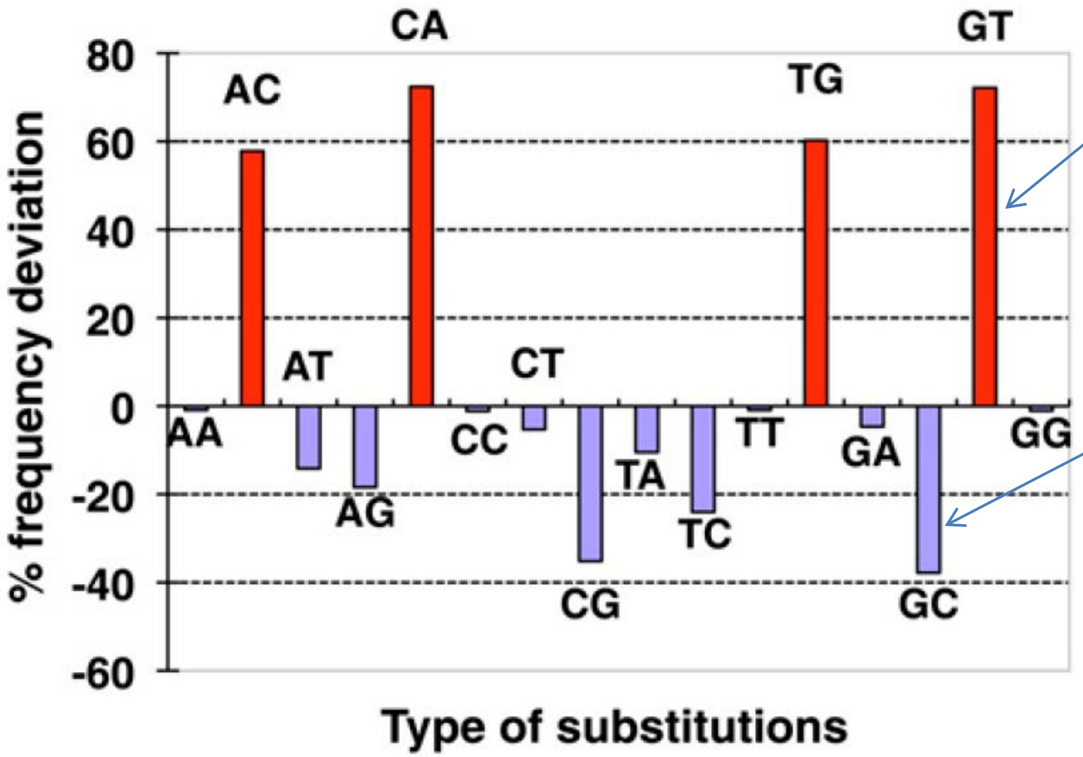


R Li et al(2009) Genome Research 19:1124-132



# Substitution bias

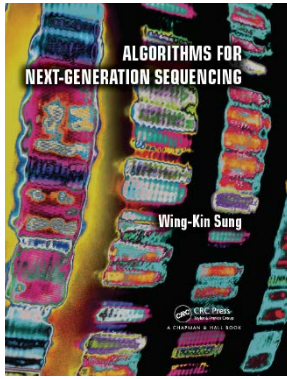
- (O-R)/R= (mismatch rate - Qscore error rate)/(Qscore error rate)



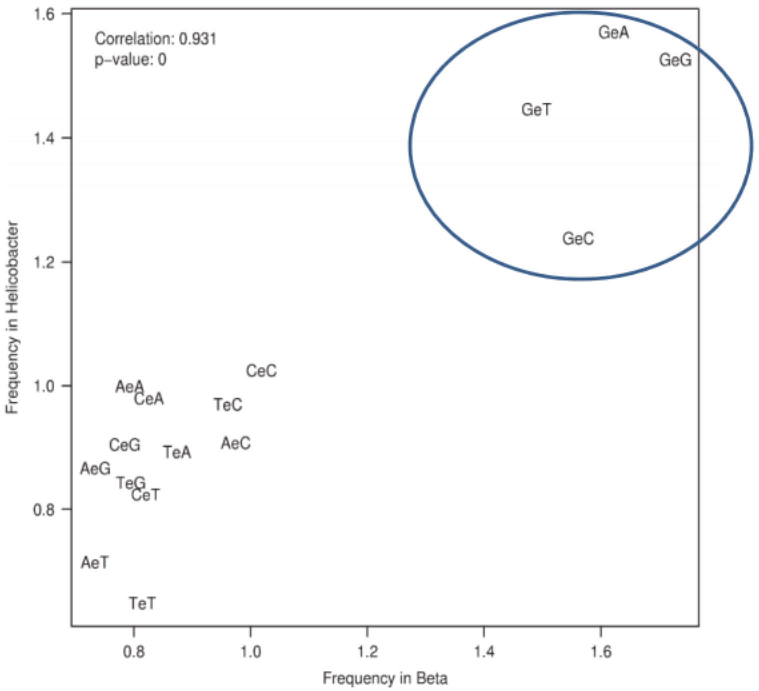
Read=G and Ref=T
Qscore under-estimate mismatch rate

Read=G and Ref=C
Qscore over-estimate mismatch rate

R Li et al(2009) Genome Research 19:1124-132

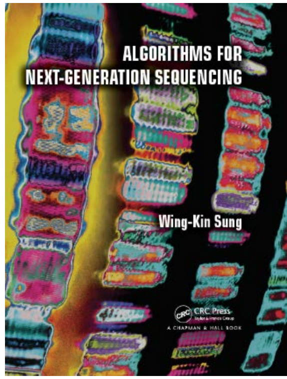


# Dinucleotide context



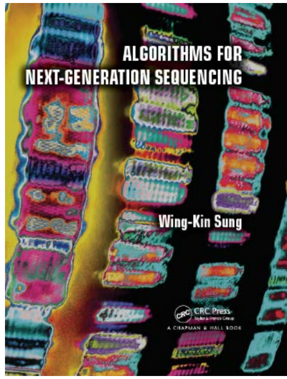
G is a likely base before an error

Dohmet al (2008) NAR. 36(16):e105



- A number of methods are proposed to recalibrate base quality score:
  - SOAPsnp
  - GATK
  - ReQON

- ReQON uses logistic regression model on



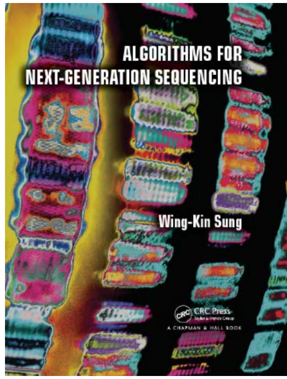
# Simple recalibration of quality score

- Let $b_1, \ldots, b_n$ be all bases.
- Let $q_1, \ldots, q_n$ be the corresponding quality scores.

- Let $e_i = 10^{-\frac{q_i}{10}}$.
- Let $q_{global} = -10 \log_{10}\left(\frac{1}{n}\sum_{i=1}^{n} e_i\right)$.
- Let x be the number of true errors.
- Let $\epsilon = -10 \log_{10} \frac{x}{n}$.

- Then, the recalibrated score is $q_i+(\varepsilon-q_{global})$.

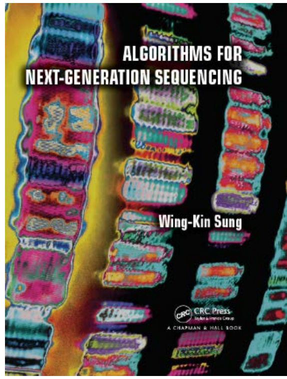
# Illustration the simple recalibration of quality score



- In practice, $\varepsilon$ and x is unknown.
- We assume any SNV that is not in dbSNP128 is an error; otherwise, it is not an error.

- **Example**: Suppose we have 1000 reads, each of length 100bp.
- So, we sequenced 100k bases.
- Assume 100 of them are different from the reference bases.
- Out of these 100 bases, suppose 95 of them are dbSNP128.
- Then, x=100-95.
- $\epsilon = -10\log_{10}\frac{100-95}{100000} = 43.$

- Suppose $q_{global}$=45.
- If $q_i$=30, the recalibrated score is 30 + 43 – 45 = 28.

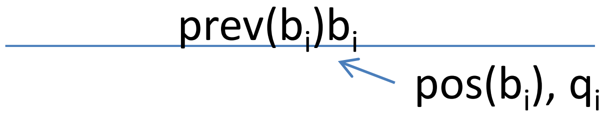


# Recalibration table

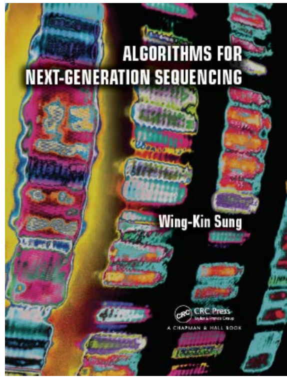
- Align subsample of reads from a lane to human reference
  - Exclude all known dbSNP sites
  - Assume all other mismatches are sequencing errors

- $\varepsilon(prev(b_i)b_i, pos(b_i), q_i) = -10\log_{10}\frac{error(prev(b_i)b_i, pos(b_i), q_i)+1}{count(prev(b_i)b_i, pos(b_i), q_i)+1}$

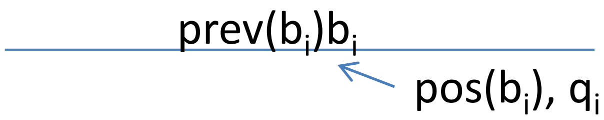
- The recalibration score of $b_i$ is $\left(\varepsilon - q_{global}\right) + \left(\varepsilon(prev(b_i)b_i, pos(b_i), q_i) - q_i\right)$



# Example

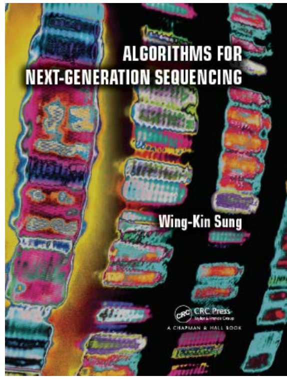


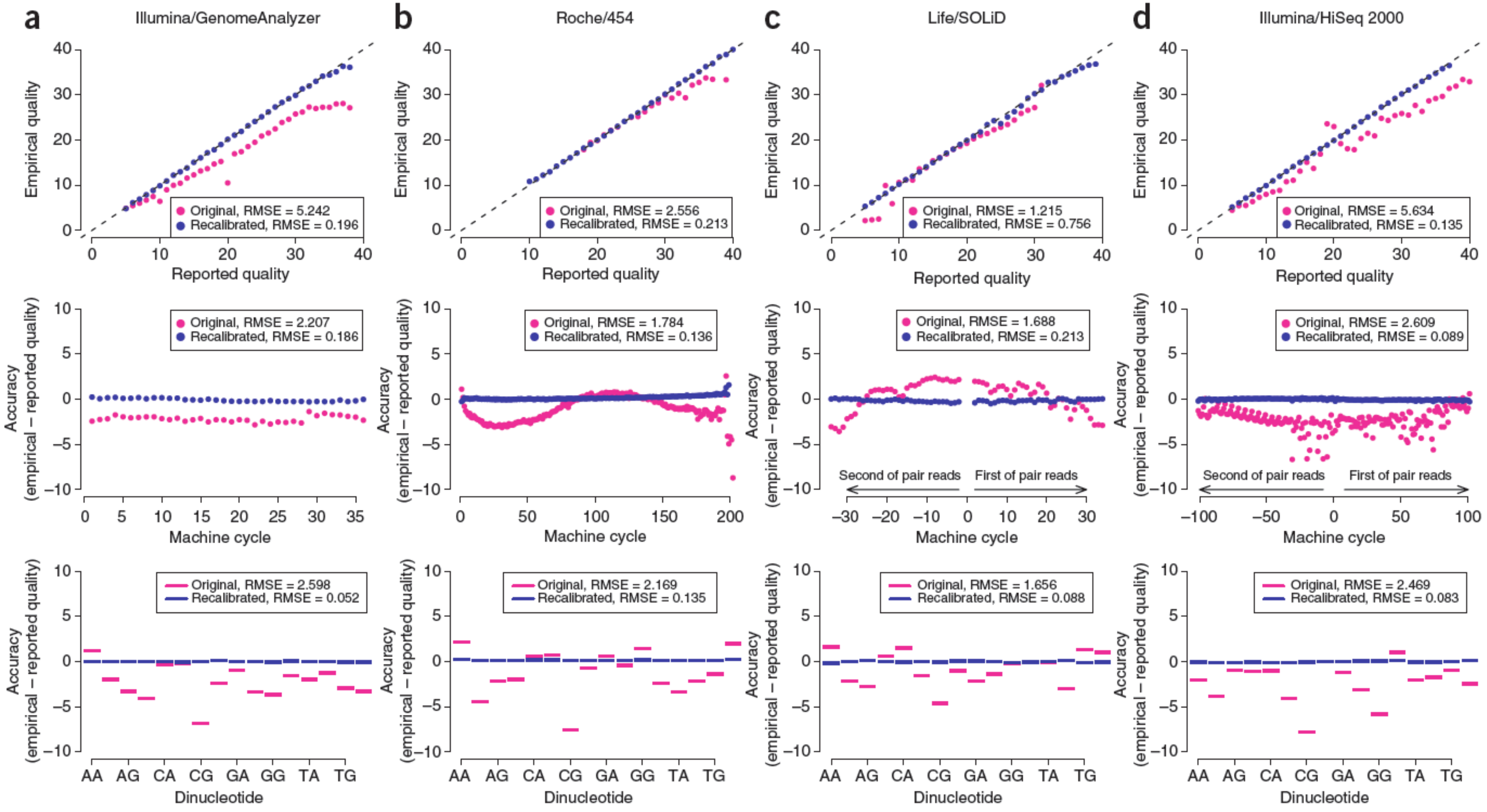
- Example:
  - error(AA, 2, 40) = 8
  - count(AA, 2, 40) = 3239
  - $\varepsilon(AA, 2, 40) = -10\log_{10}\frac{8+1}{3239+1} = 25.56$



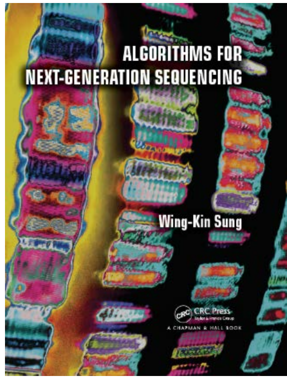
| Positions 1 & 2 | Count | Diff from ref | ε |
|---|---|---|---|
| AA | 3239 | 8 | $-10\log_{10}\frac{8+1}{3239+1}$ |
| CA | 4223 | 5 | $-10\log_{10}\frac{8+1}{3239+1}$ |
| GA | 3518 | 2 | $-10\log_{10}\frac{8+1}{3239+1}$ |
| TA | 4032 | 20 | $-10\log_{10}\frac{8+1}{3239+1}$ |
| … | | | |
| TT | | | |

# Quality score before and after recalibration





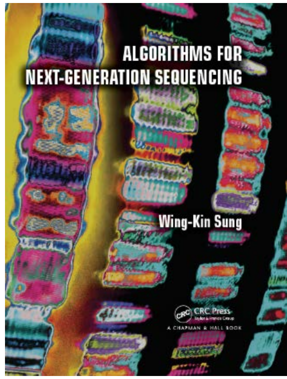
DePristo et al. Nature Genetics, 2011.



# Rule-based filtering

- It is used to filter false positives resulting from correlated sequencing artifacts.

- Samtools (or MAQ) rule-base filter:
  - Discard SNPs near to indels (within 3bp flanking region of a potential indel).
  - Discard SNPs with low coverage (covered by 3 or fewer reads).
  - Discard SNPs covered by reads with poor mapping only (mapping quality lower than 60 for all covered reads).
  - Discard SNPs in SNP dense regions (within a 10bp region containing 3 or more SNPs).
  - Discard SNPs with consensus quality smaller than 10.
- MuTect rule-base filter:
  - Discard SNPs near to indels (false positives caused by misaligned small indel events).
  - Discard SNPs covered by reads with poor mapping
  - Discard SNPs on triallelic sites
  - Discard SNPs covered by reads with strand bias
  - Discard SNPs covered by reads mapped to similar location
  - Discard SNPs in tumor if some reads in normal also contain the SNPs
- Discard SNPs also appear in a panel of normal samples (since they are not expected to cause disease).



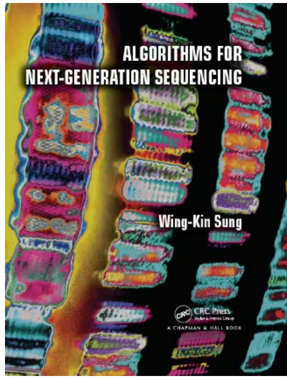
# Iterative mapping-based method for SNV calling

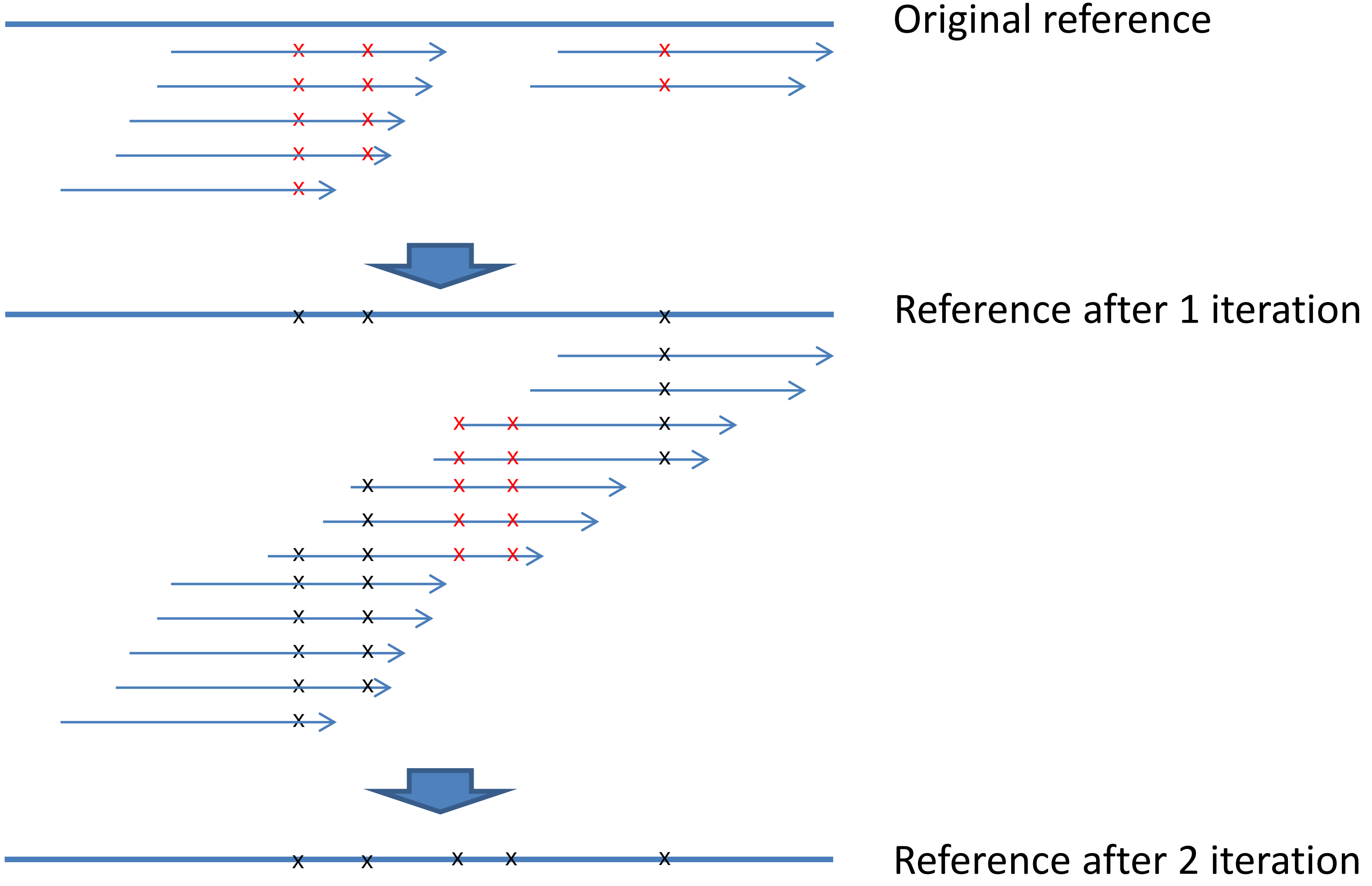
- Previous methods assume SNVs are sparse.
- When there are SNV hotspot (2 or more SNVs cluster together), previous methods fail to identify SNVs.
- Such scenerio happens in bacteria.

- Solution: Iterative mapping
  - iCORN (Otto et al. 2010)
  - ComB (Souaiaia et al. 2011)

# Idea

- Mapping allow at most 2 mismatches.
- SNV is called if there exists 2 supporting reads.



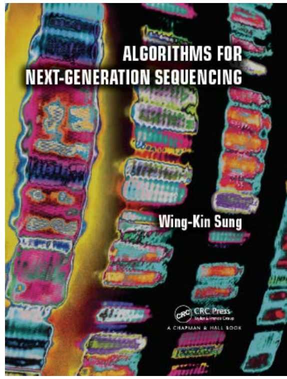


Original reference

Reference after 1 iteration

Reference after 2 iteration

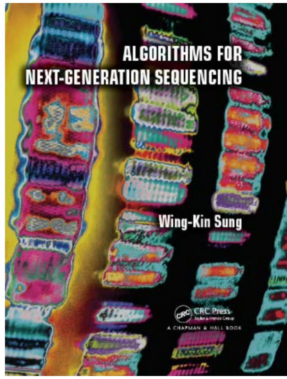
# Iterative Correction of Reference Nucleotides (iCORN)



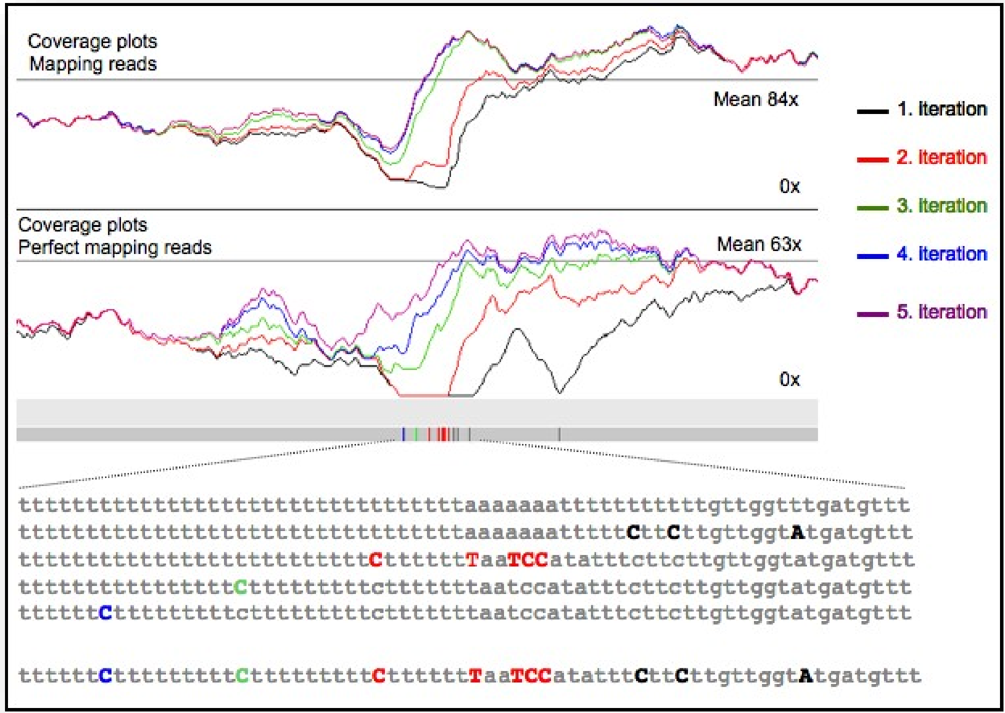
Repeat

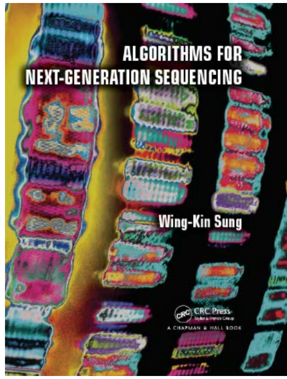
1. Map reads to reference using SSAHA.
2. Call SNVs and indels
3. Correct the reference using the called SNVs/indels
4. Remap the reads and measure the coverage (using SNPoMatic)
5. If the coverage decreased, then undo correction.

Until no new SNVs/indels can be found.

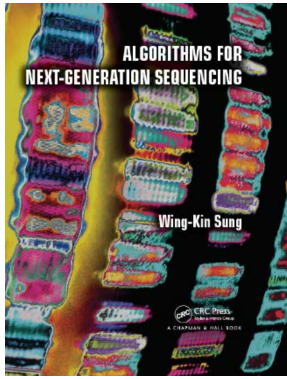


- iCORN can improve the sensitivity of a SNV caller.
- However, it also increases the number of false positives.





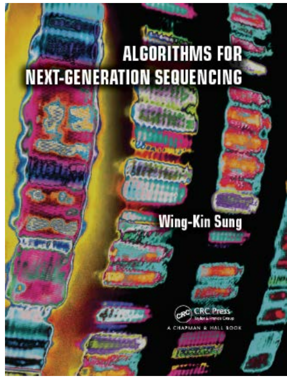
# Indel calling



# Indel calling

- Increasing evidence of indels being involved in a number of diseases (Yang et al. 2010).

1. Realignment based approach (discussed!)
   - E.g. GATK, Dindel
2. Split-read approach
   - E.g. Pindel, microindels, Splitread
3. Span distribution-based clustering approach
   - E.g. MoDIL
4. Local assembly approach
   - E.g. SOAPindel

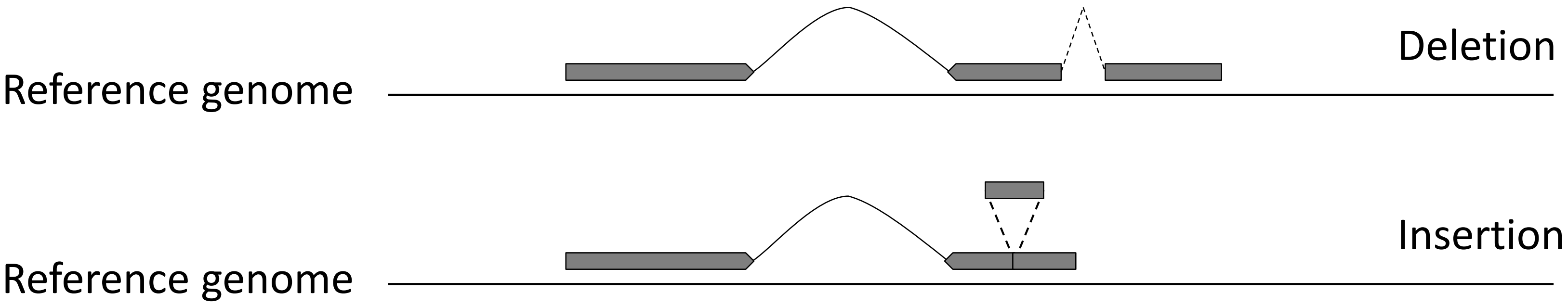


# Split-read approach

- E.g. Pindel

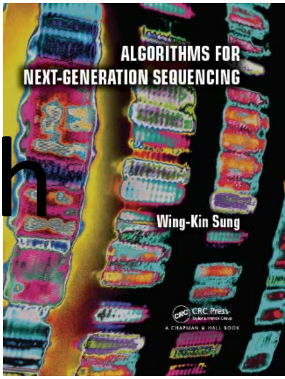
1. Enumerate all paired-end reads where only one read is fully aligned.
2. Check if the non-fully aligned read can map near to its mate after allowing a short indel
3. An indel event is reported if such candidate indel is supported by at least two paired-end reads.

- Pindel can detect the exact breakpoints of an indel.

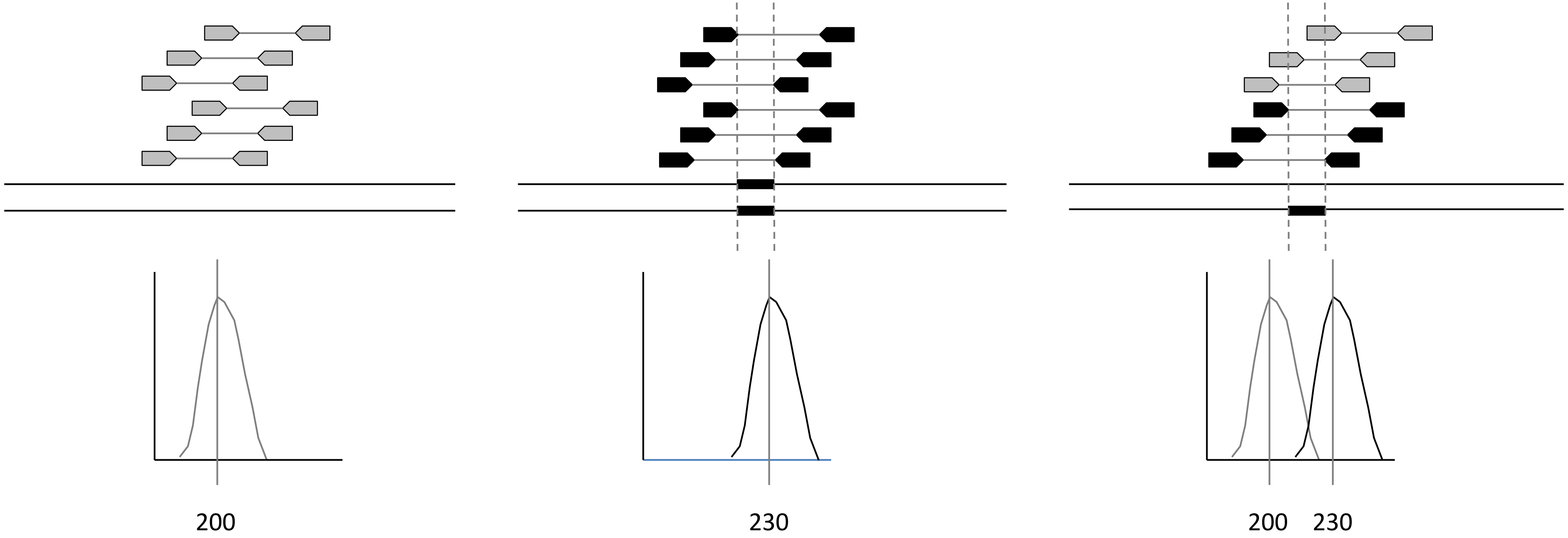


This is known as the 'anchored split mapping' signature.

# Span distribution-based clustering approach



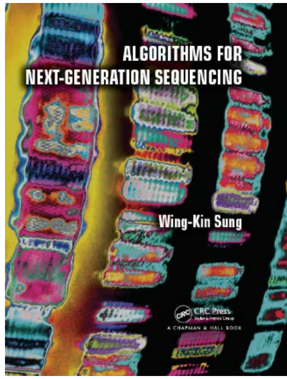
- MoDIL. The first method to use the span distribution-based clustering approach, allowing the detection of smaller indels, and explicitly modelling heterozygosity.



Same as reference genome

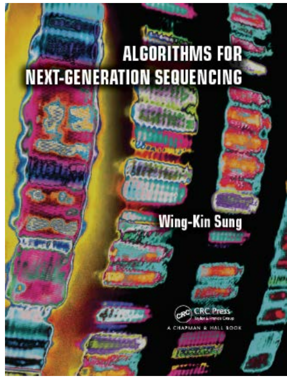
Homozygous deletion

Heterozygous deletion



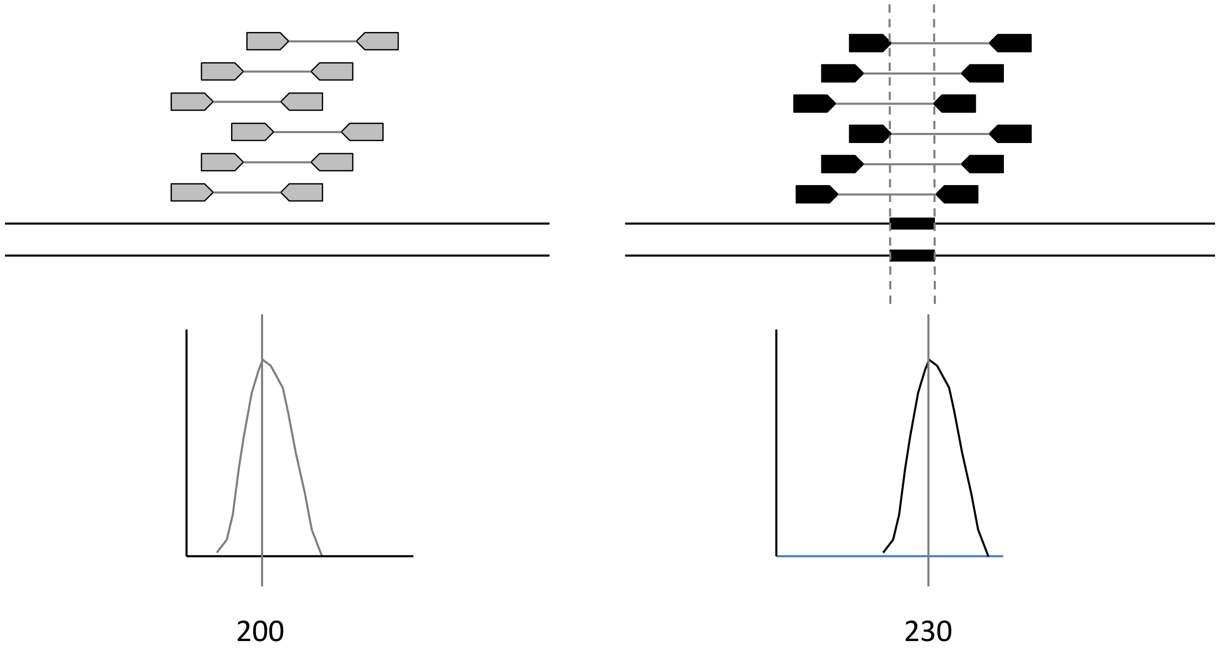
# Algorithm

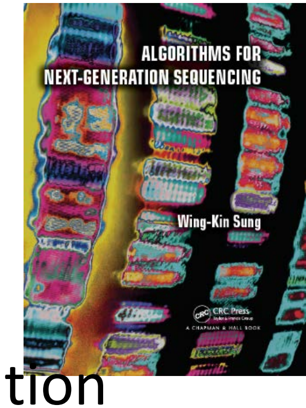
1. Align the paired-end reads to the reference genome.
2. Identify the span distribution Y of the paired-end reads.
3. Identify paired-end reads with abnormal insert size.
4. Find clusters of paired-end reads that overlap with the abnormal insert size.
5. For each cluster,
   - Check if it is:
     - (1) The same as the distribution Y
     - (2) A mixture of two distribution X1 and X2
     - (3) A distribution X
   - EM algorithm is used to model the cluster as a mixture of two distributions. (KS test is used to evaluate the goodness of the mixture.)



# Idea

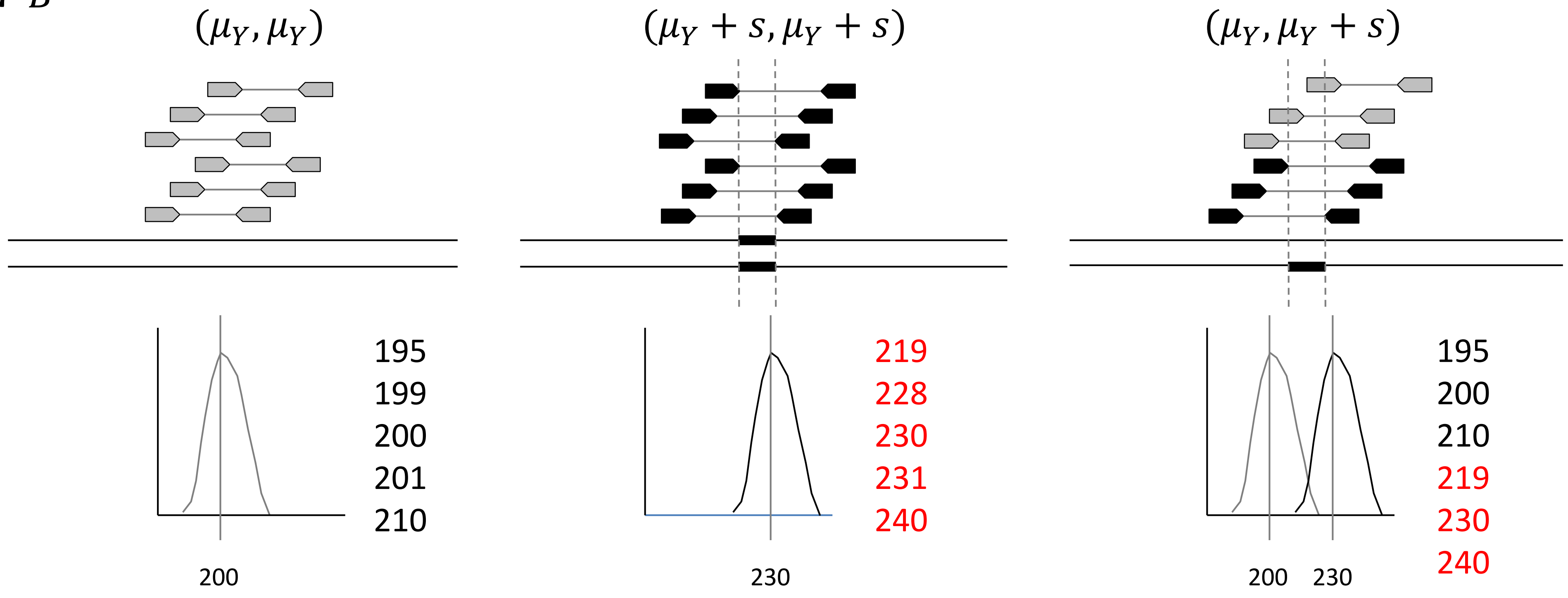
- Let Y be the insert size distribution of the whole library. Let $\mu$ be the mean insert size.
- Let X be all paired-end reads around an indel of size s, we expect their insert size distribution has the same shape as Y, but the mean is shifted to $\mu + s$.
- Precisely, let X = {$X_1$, …, $X_n$} be the insert sizes of a cluster of paired-end reads with mean $\mu_X$. We expect $\Pr(X_i|\mu_X) = \Pr(Y = X_i - \mu_X + \mu_Y)$.





# Idea

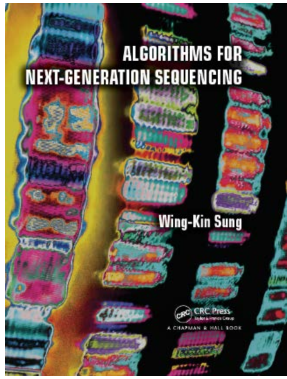
- If X contains two set of paired-end reads $X_A$ and $X_B$, then the insert size distribution is a mixture of two distributions of the same shape as Y, where their means are $\mu_A$ and $\mu_B$.



Same as reference genome

Homozygous deletion

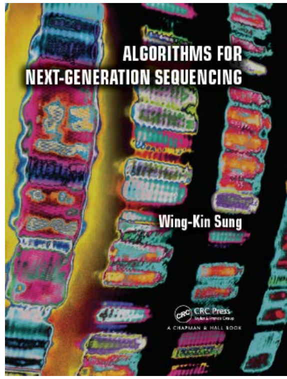
Heterozygous deletion



# Aim

- Input:
  - Y = {$Y_1$, …, $Y_{|Y|}$} is the insert sizes of the full library. Let $\mu_Y$ be its mean and $\sigma_Y$ be its standard derivation
  - X = {$X_1$, …, $X_{|X|}$} is a mixture of two set of insert sizes $X^A$ and $X^B$ , extracted from the two haplotypes.
- Aim: We aim to find mean insert sizes $\mu_A$ and $\mu_B$ of the two sets $X^A$ and $X^B$, respectively.

# Check if a distribution X have the same shape as Y

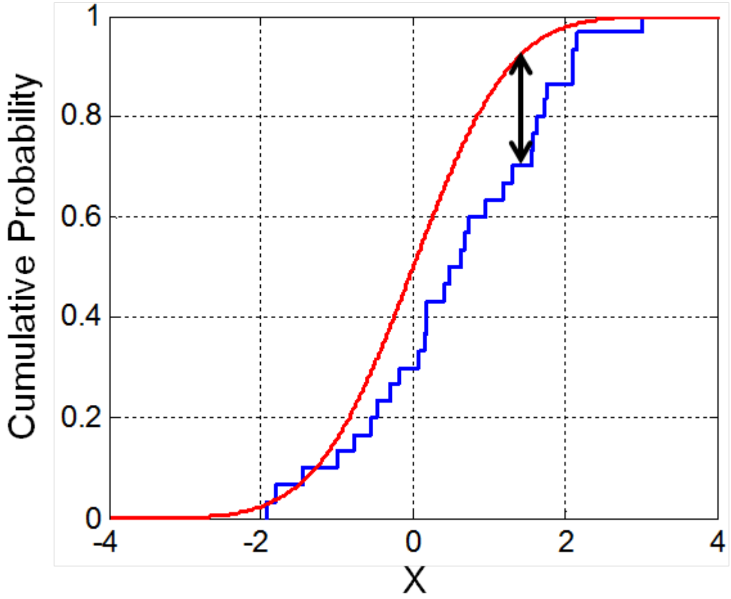


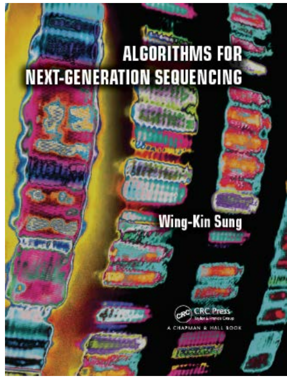
- Y = {$Y_1$, …, $Y_{|Y|}$} is the insert sizes of the full library
  - Let $\mu_Y$ be its mean and $\sigma_Y$ be its standard derivation
- X = {$X_1$, …, $X_{|X|}$} is the insert sizes of the cluster
  - Let $\mu_X$ be its mean and $\sigma_X$ be its standard derivation

- To check if X fits the distribution of Y, we can use KS test.
- Let $f_Z(v) = \frac{\sum_{Z_j \in Z} I(Z_j - \mu_Z \le v)}{|Z|}$, where

$$I(Z_j - \mu_Z \le v) = \begin{cases} 1 & \text{if } Z_j - \mu_Z \le v \\ 0 & otherwise \end{cases}.$$

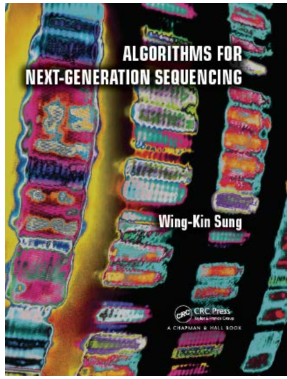
- The KS statistics is
  - $D_X = \max_v |f_X(v) - f_Y(v)|$.
- If $D_X$ is significantly small,
  Y and X have the same shape.





# Check if X fits the mixture model

- Input:
  - Y = {$Y_1$, …, $Y_{|Y|}$} is the insert sizes of the full library,
  - X is a mixture of two set of insert sizes $X^A$ and $X^B$
- To test if the distributions of both $X^A$ and $X^B$ have the same shape as that of Y, we use the following statistics:
  - $\frac{|X^A|}{|X|} D_{X^A} + \frac{|X^B|}{|X|} D_{X^B}$, where $D_X = \max_v |f_X(v) - f_Y(v)|$
- If the statistics is significantly small, we accept that X is a mixture of distributions having the same shape as Y

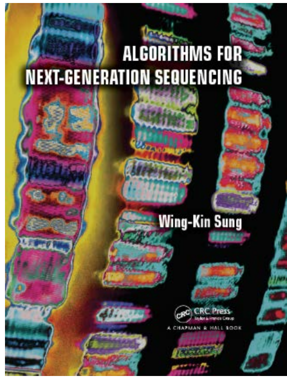


# Learn the mixture model of X

- We perform EM algorithm.
- We aim to learn $\mu_A, \mu_B$ and $\lambda$, where $\lambda_A = \frac{|X^A|}{|X|}$

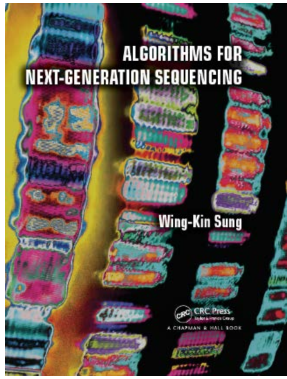
- Input: X={X$_1$, …, X$_n$}
- 1. Initialization of $\mu_A, \mu_B$ and$\lambda_A$
- 2. E-step: Compute $\gamma_{jt} = \Pr\left(X_j \in X^t \middle| \lambda_A, \mu_A, \mu_B\right)$
- 3. M-step: Determine $\lambda_A, \mu_A$ and $\mu_B$.
  - $\lambda_A = \frac{1}{n}\sum_{j=1}^{n} \gamma_{jA}, \lambda_B = \frac{1}{n}\sum_{j=1}^{n} \gamma_{jB} = 1 - \lambda_A$
  - $\mu_A$ and $\mu_B$ are set to be the value that minimizes

$$\lambda_A \max_v \left| \frac{\sum_{X_j - \mu_A \le v} \gamma_{jA}}{\lambda_A} - f_Y(v) \right| + \lambda_B \max_v \left| \frac{\sum_{X_j - \mu_B \le v} \gamma_{jB}}{\lambda_B} - f_Y(v) \right|$$



# E-step

- Compute $\gamma_{jt} = \Pr\left(X_j \in X^t \middle| \lambda_A, \mu_A, \mu_B\right)$
- $\gamma_{jt} = \dfrac{\lambda_t \Pr\left(X_j \middle| \mu_t\right)}{\lambda_A \Pr\left(X_j \middle| \mu_A\right) + \lambda_B \Pr\left(X_j \middle| \mu_B\right)}$
  - where t=A,B, j=1,…,n.

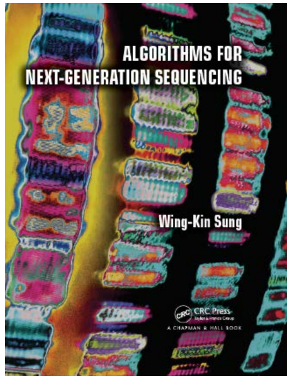


# M-step

- Given $\gamma_{jt} = \Pr\left(X_j \in X^t \middle| \lambda_A, \mu_A, \mu_B\right)$, find $\lambda_A$, $\mu_A$ and $\mu_B$ that minimizes $\lambda_A D_{X^A} + \lambda_B D_{X^B}$.
- We have: $\lambda_A = \frac{1}{n}\sum_{j=1}^{n} \gamma_{jA}$, $\lambda_B = \frac{1}{n}\sum_{j=1}^{n} \gamma_{jB} = 1 - \lambda_A$
- $\mu_A$ and $\mu_B$ are set to be the value that minimizes

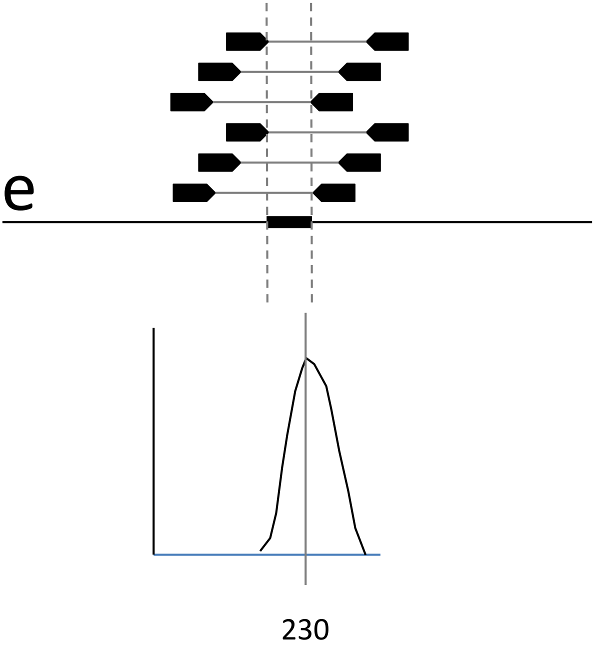
$$\lambda_A \max_v |f_A(v) - f_Y(v)| + \lambda_B \max_v |f_B(v) - f_Y(v)|$$

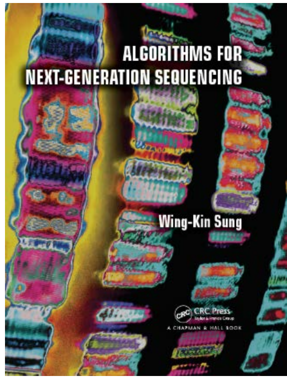
  - where $f_t(v) = \frac{\sum_{X_j - \mu_t \le v} \gamma_{jt}}{\sum_{j=1}^{n} \gamma_{jt}} = \frac{\sum_{X_j - \mu_t \le v} \gamma_{jt}}{n\lambda_t}$.



# Determine the indel size

- Let $\mu_Y$ be its mean and $\sigma_Y$ be its standard derivation of the full library Y
- Let X = {$X_1$, …, $X_n$} be the insert sizes of a cluster whose distribution is the same as Y.
- Let $\mu_X$ be the mean insert size of X
- Then, the indel size follows a Guassian distribution N($\mu_X - \mu_Y, \frac{\sigma_Y}{\sqrt{n}}$).





# Local assembly approach

- This approach is used by SOAPindel and Scalpel.
- Method for SOAPindel:
- 1. Identify a set of reads whose mates do not map on the reference genome.
- 2. Find the expected positions of the unmapped reads (given the insert size). These reads are called virtual reads.
- 3. Identify cluster of virtual reads. Then, for each cluster, contigs are generated by de novo assembly.
- 4. Align contigs on the reference genome to identify potential indels.