CS681: Advanced Topics in Computational Biology

Week 5 Lectures 2-3

Can Alkan
EA224
calkan@cs.bilkent.edu.tr

http://www.cs.bilkent.edu.tr/~calkan/teaching/cs681/

Indel discovery with NGS data

- Indels: insertions and deletions < 50 bp.
  - ~0.5 million indels per person
  - Database: dbSNP http://www.ncbi.nlm.nih.gov/projects/SNP/
- Input: sequence data and reference genome
- Output: set of indels and their genotypes (homozygous/heterozygous)
- Often there are errors, filtering required
- Most indel detection methods are based on statistical analysis
- Tools: GATK, Dindel, Pindel, SAMtools, SPLITREAD, PolyScan, VarScan, etc.

Challenges (reminder)

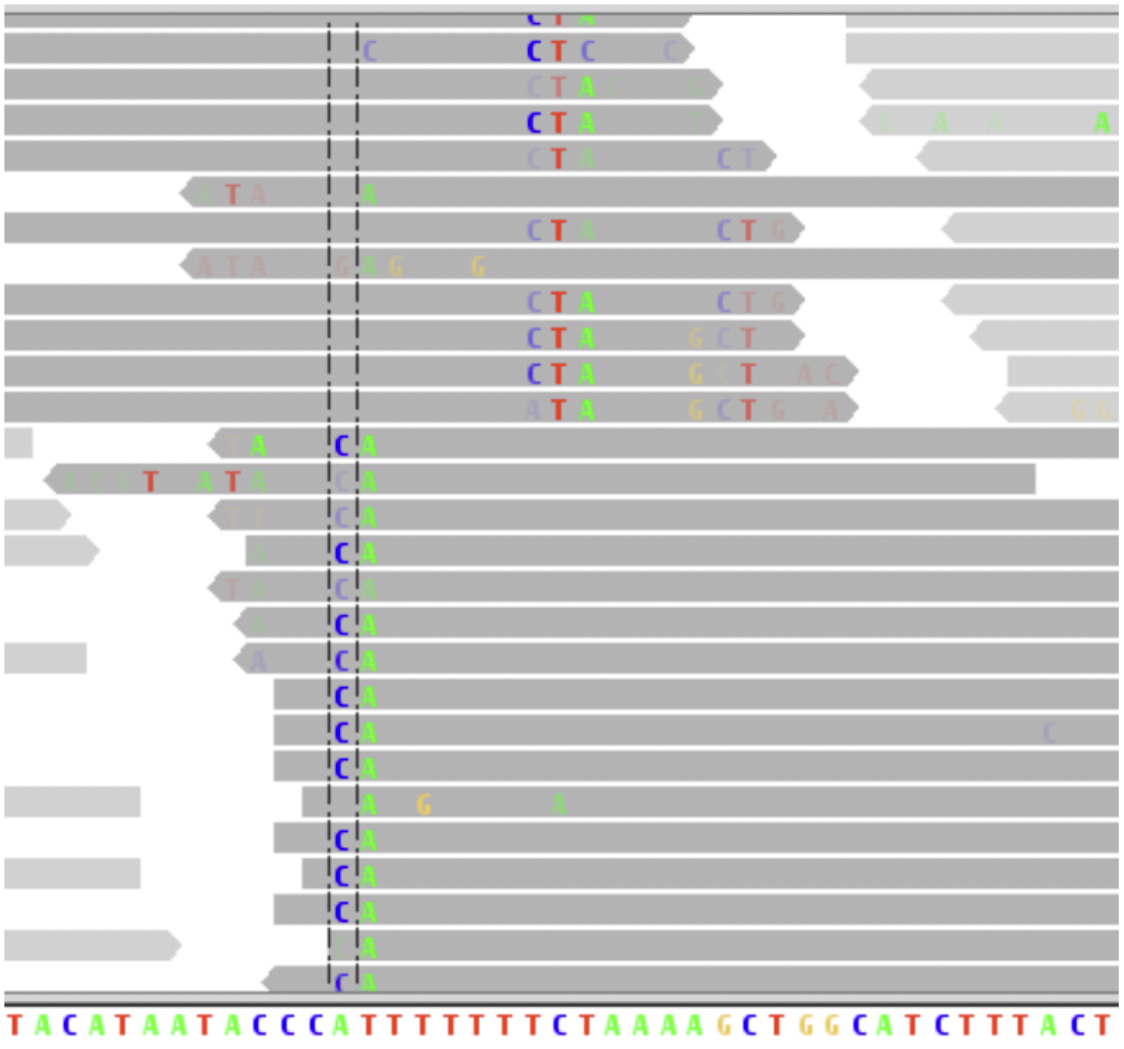
- Sequencing errors
- Paralogous sequence variants (PSVs) due to repeats and duplications
- Misalignments
  - Indels vs SNPs, there might be more than one optimal trace path in the DP table
  - Short tandem repeats
  - Need to generate multiple sequence alignments (MSA) to correct

Finding indels

- Sequence aligners are often unable to perfectly map reads containing insertions or deletions (indels)
  - Indel-containing reads can be either left **unmapped** or arranged in gapless alignments
  - Mismatches in a particular read can interfere with the gap, esp. in low-complexity regions
  - Single-read alignments are “correct” in a sense that they do provide the best guess given the limited information and constraints.

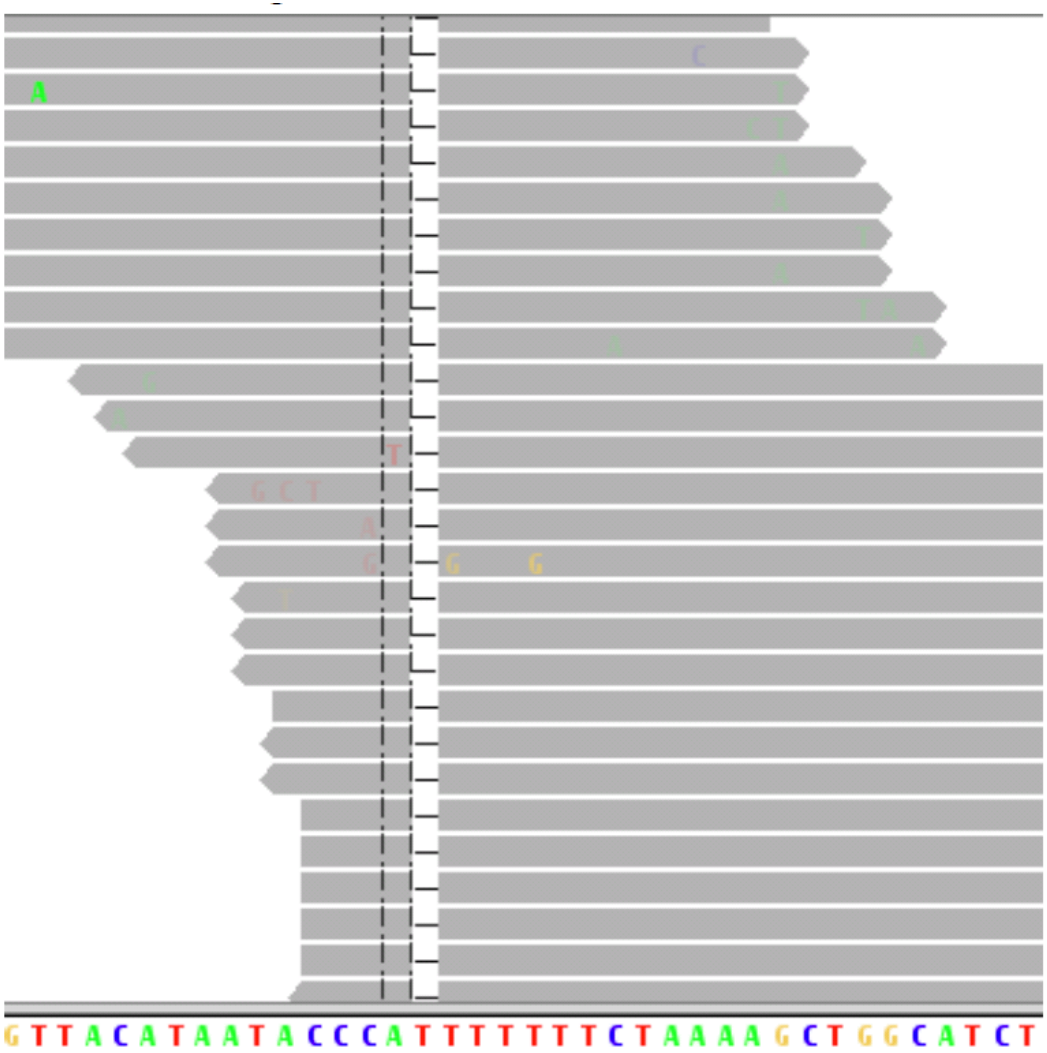
Slide from Andrey Sivachenko

Need to realign



Slide from Andrey Sivachenko

After MSA



Slide from Andrey Sivachenko

Left alignment of indels

- If there is a short repeat, there might be more than one alternative alignments of indels
  - Common practice is to select the “left aligned” version

CGTATGATCTAGCGCGCTAGCTAGCTAGC
CGTATGATCTA - - GCGCTAGCTAGCTAGC

CGTATGATCTAGCGCGCTAGCTAGCTAGC
CGTATGATCTAGC - - GCTAGCTAGCTAGC

CGTATGATCTAGCGCGCTAGCTAGCTAGC
CGTATGATCTAGCGC - -TAGCTAGCTAGC

Left aligned

GATK indel calling

$$P(G \mid D) = \frac{P(G)P(D \mid G)}{\sum_i P(G_i)P(D \mid G_i)}$$

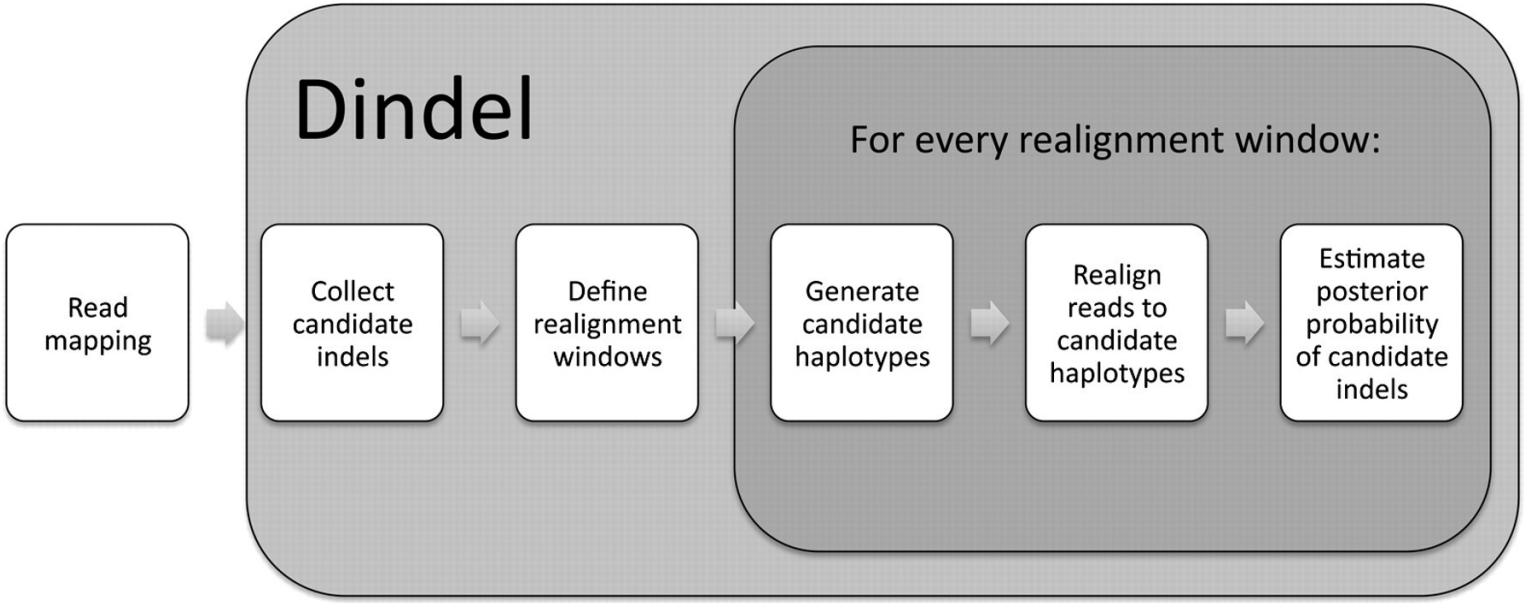
$$P(D \mid G) = \prod_j \left(\frac{P(D_j \mid H_1)}{2} + \frac{P(D_j \mid H_2)}{2} \right), where\ G = H_1 H_2$$

$$P(D_j \mid H) = \sum_{\substack{alignments \pi \\ of\ Dj\ to H}} P(D_j \mid \pi)$$

- Haplotypes are discovered from indels in the reads
- Diploid genotypes G for all haplotype H_iH_j combinations
- For each haplotype H_i, calculate likelihood of reads D_j over all possible alignments π
- Sum computed by an HMM using haplotype, bases and quality scores

Slide from Mark Depristo

Dindel



- Statistical methods that GATK indel caller is based on
- Candidate indels are collected from regions with reads with mismatches & indels

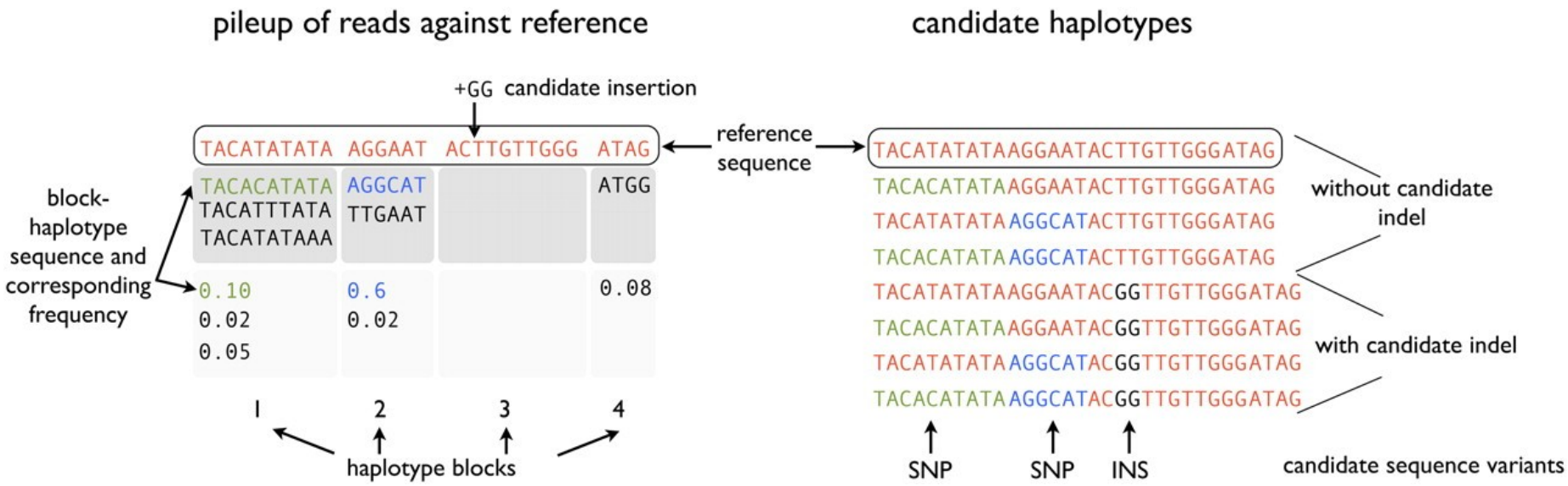
Albers et al. Genome Research, 2011

Dindel main steps

- Identify the set of reads $\{\mathbf{R}_i\}$ to be realigned.
  - Reads that overlap with 120 bp windows around the candidates
- Generate the set of candidate haplotypes $\{\mathbf{H}_j\}$.
  - Same 120 bp windows
- Compute the maximum likelihood $P_{\max}(\mathbf{R}_i \mid \mathbf{H}_j)$ and maximum-likelihood alignment of each read $\mathbf{R}_i$ given each candidate haplotype $\mathbf{H}_j$ using the probabilistic realignment model.
- Estimate haplotype frequencies from the read-haplotype likelihoods $P_{\max}(\mathbf{R}_i \mid \mathbf{H}_j)$ and the prior probability of each candidate haplotype.
- Estimate quality scores for the candidate indels and other sequence variants.

Albers et al. Genome Research, 2011

Dindel candidate haplotypes



Albers et al. Genome Research, 2011

Probabilistic realignment

$\boldsymbol{P_{max}(R_i \mid H_j)}$**,** the probability of observing the read R_i given that the true underlying haplotype sequence from which it was sequenced is given by H_j.

Aligment done using an HMM

$$P_{\max}(R_i \mid H_p) = \max_{X_i, I_i} P(R_i = \cdot_i, X_i, I_i \mid H_p, \theta$$

Albers et al. Genome Research, 2011

Dindel haplotype inference

$$l(\mathbf{H}_j, \mathbf{H}_{j'}) \equiv \prod_i \left[\frac{P_{\max}(\mathbf{R}_i \mid \mathbf{H}_j)}{2} + \frac{P_{\max}(\mathbf{R}_i \mid \mathbf{H}_{j'})}{2}\right], \quad (1)$$

$$P_{\text{post}}(\mathbf{H}_j, \mathbf{H}_{j'}) \propto l(\mathbf{H}_j, \mathbf{H}_{j'}) P(\mathbf{H}_j, \mathbf{H}_{j'}), \quad (2)$$

$$(\mathbf{H}_{\text{pat}}^{\text{MAP}}, \mathbf{H}_{\text{mat}}^{\text{MAP}}) = \underset{(\mathbf{H}_j, \mathbf{H}_{j'}):\#\text{indels}(\mathbf{H}_j, \mathbf{H}_{j'})>0}{\arg\max} P_{\text{post}}(\mathbf{H}_j, \mathbf{H}_{j'}). \quad (3)$$

$$Q\left(\text{indels} \in (\mathbf{H}_{\text{pat}}^{\text{MAP}}, \mathbf{H}_{\text{mat}}^{\text{MAP}})\right) = -10 \log_{10} \frac{\max_{(\mathbf{H}_j, \mathbf{H}_{j'}):\#\text{indels}(\mathbf{H}_j, \mathbf{H}_{j'})=0} P_{\text{post}}(\mathbf{H}_j, \mathbf{H}_{j'})}{P_{\text{post}}(\mathbf{H}_{\text{pat}}^{\text{MAP}}, \mathbf{H}_{\text{mat}}^{\text{MAP}}) + \max_{(\mathbf{H}_j, \mathbf{H}_{j'}):\#\text{indels}(\mathbf{H}_j, \mathbf{H}_{j'})=0} P_{\text{post}}(\mathbf{H}_j, \mathbf{H}_{j'})}. \quad (4)$$

Albers et al. Genome Research, 2011

SPLITREAD

Mapping Strategy

- *mrsFAST* is used for all mappings.
  - Hamming Distance
    - Substitution Only/ No Insertions and Deletions.
    - All possible mappings of the reads.
- Input: FASTQ files/ Paired-end data
- Target: Reference genome
  - If exome sequencing is analyzed, use only Coding Regions based on RefSeq and CCDS and 300bp flanking regions + Processed pseusogenes
  - Consensus repeat sequences are combined into an artificial chromosome chrN.
- Can be used for both indel and structural variation discovery
- High sequence coverage needed

Karakoc et al., Nature Methods, 2011

SPLITREAD

- Map all reads.
  - Paired-end reads are paired based on the distribution of the insert size.
- Unmapped reads for Single/One end anchored(OEA) reads for paired-end
  - Split into half reads and form paired-end reads with 0 expected insert size.
- Map the split reads.
  - All possible mappings are reported.
- Cluster the mappings based on the mapping of split reads.
  - For each perfect split region create a cluster.
  - An OEA mapping around the split region is added to a cluster if it does not contradict the perfect split.
  - Each cluster implies an INDEL event.

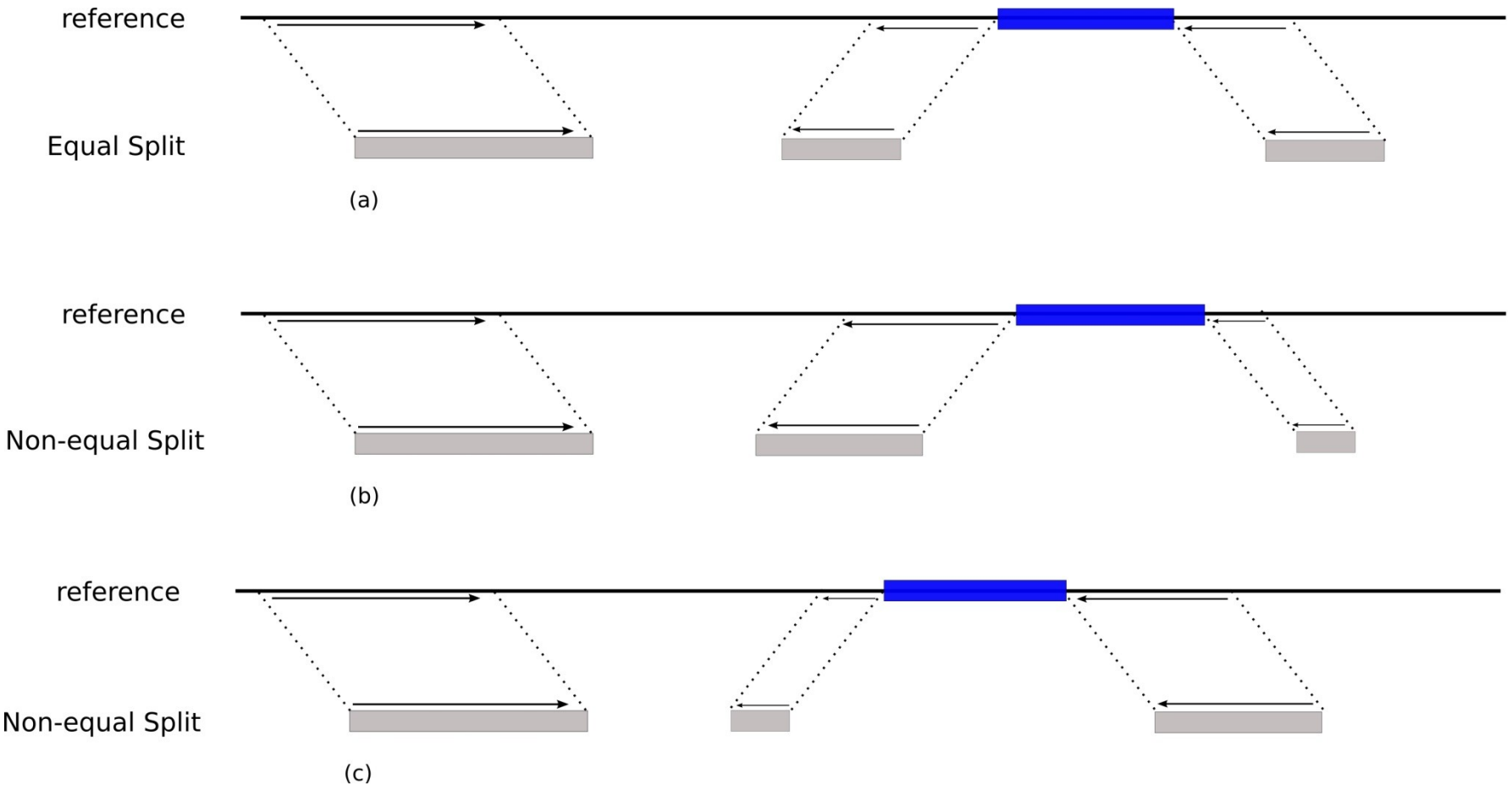
Karakoc et al., Nature Methods, 2011

SPLITREAD (cont)

- Select the approximately optimal set of events with maximum likelihood.
  - Set-cover (greedy method) is used for approximation.
  - Minimum number of events with maximum number of perfect and unbalanced events.
- Transchromosomal events -> ALU/L1/SVA insertions.
- Remaining unbalanced splits -> Large insertions.

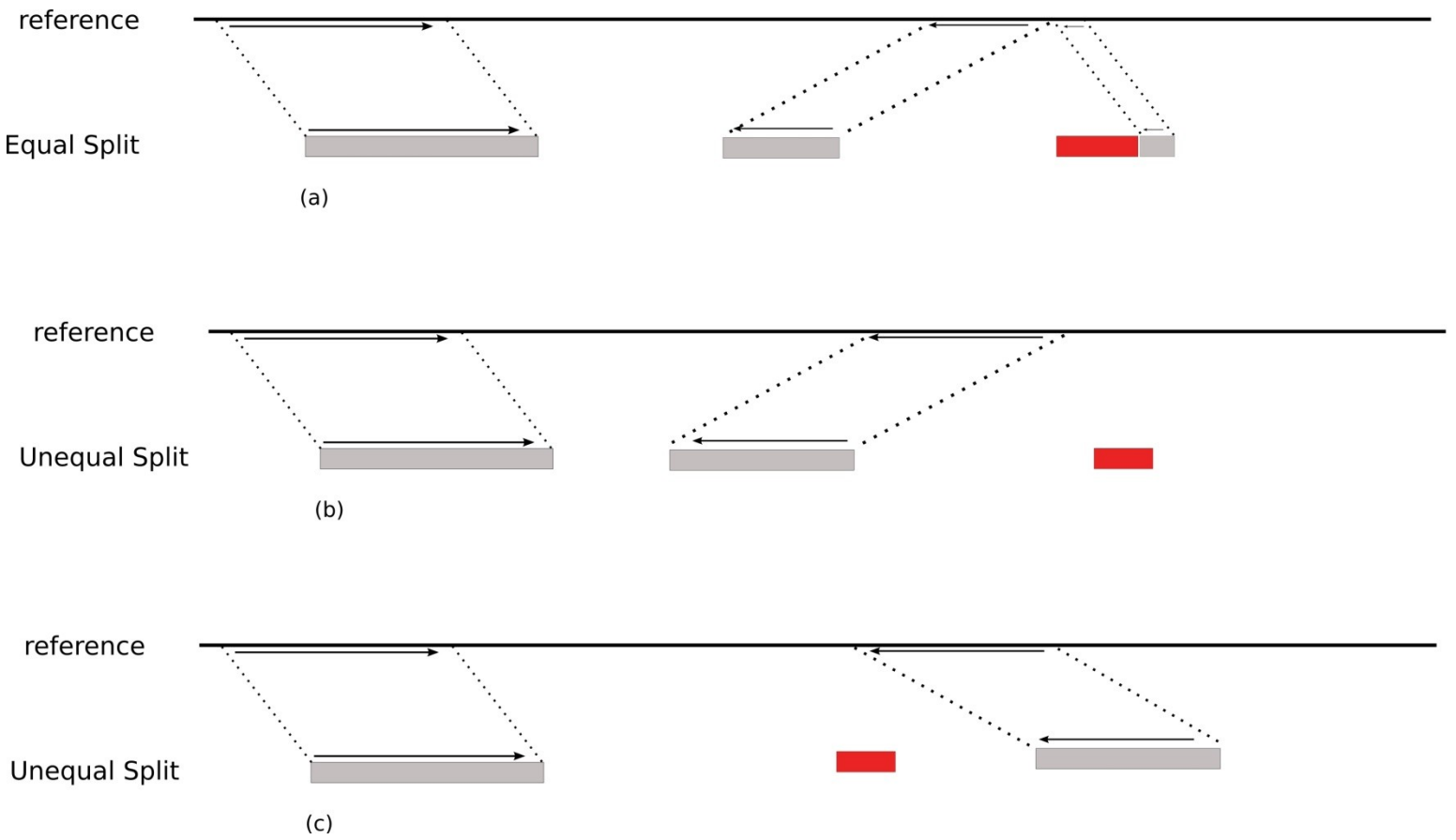
Karakoc et al., Nature Methods, 2011

Split Read - Deletion



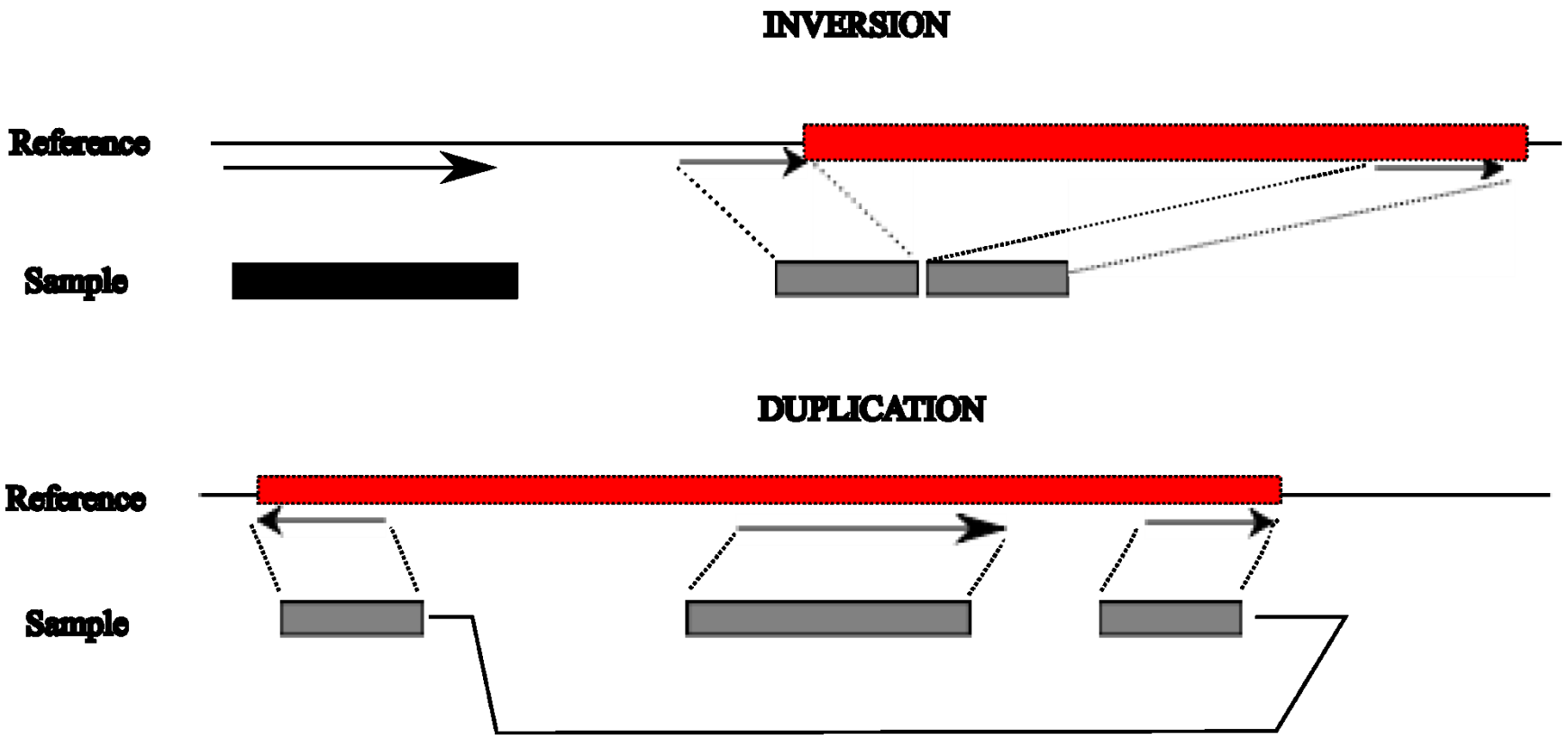
Karakoc et al., Nature Methods, 2011

Split Read - Insertion



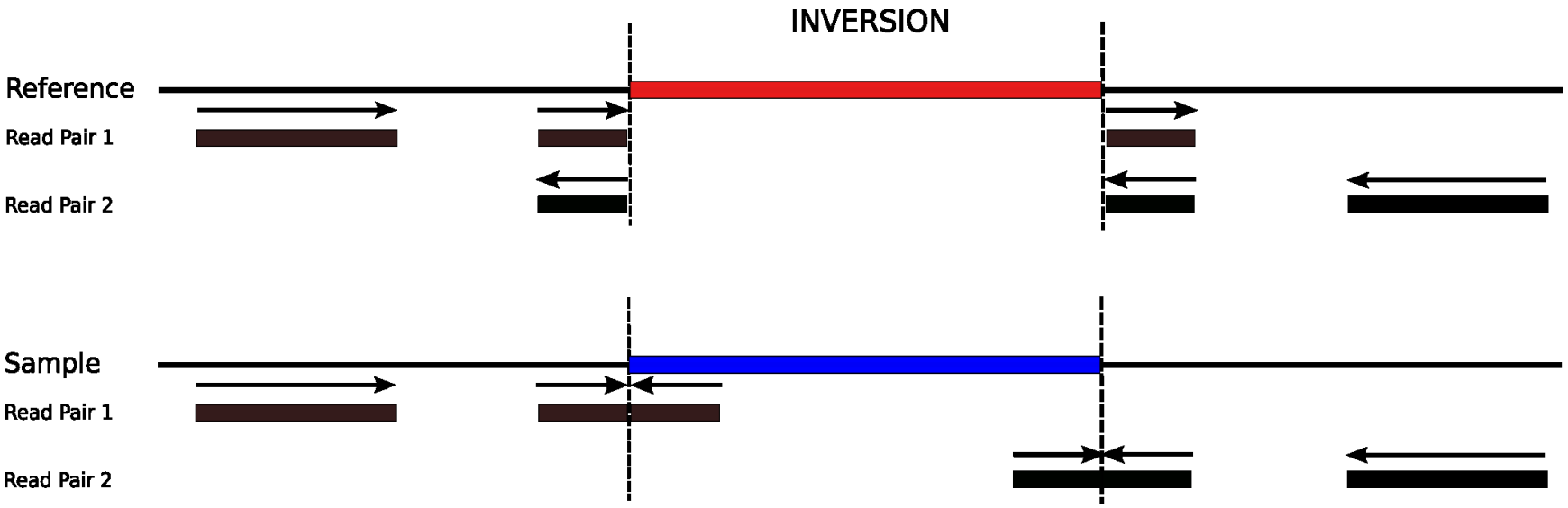
Karakoc et al., Nature Methods, 2011

Split Read – Inversion/duplication



Karakoc et al., Nature Methods, 2011

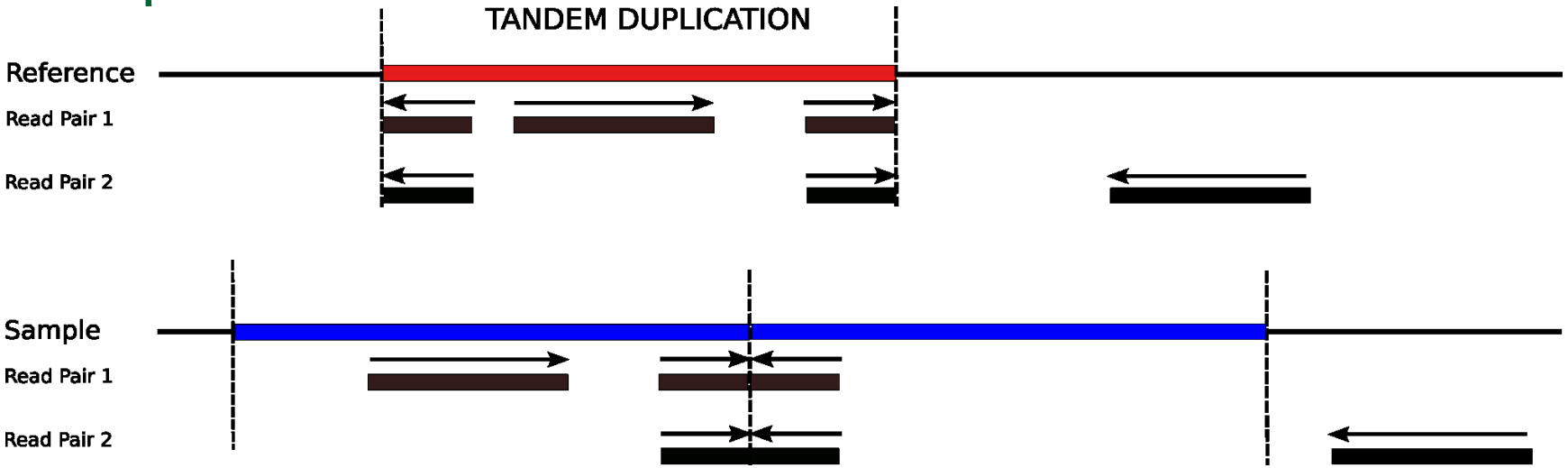
Split Reads for detecting Inversions



- Strong signature at the breakpoints of the Inversions based on directions
  - Validation from both directions.
  - Repeat content at the breakpoint defines the specificity.
  - [End of Split1 – Start of Split2] defines the inversion.

Karakoc et al., Nature Methods, 2011

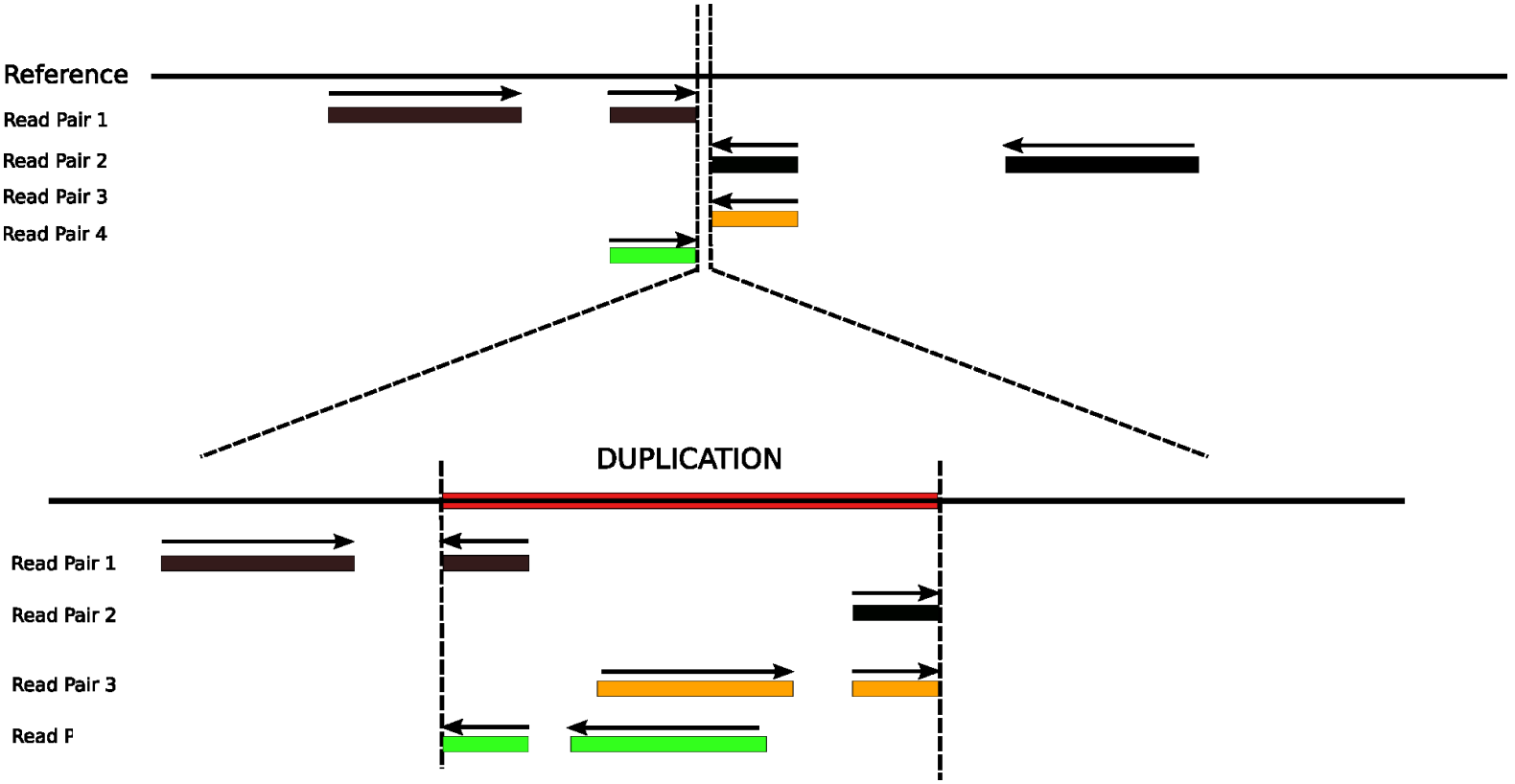
Split Reads for detecting Tandem Duplications



- Signature at the breakpoints of Tandem duplication based on direction and mapping position.
  - Validation from both directions and within the duplicated region.
  - Repeat content at the breakpoint defines the specificity.
  - Non-template duplications are not clear.
  - [End of Split1 – Start of Split2] defines the tandem duplication.

Karakoc et al., Nature Methods, 2011

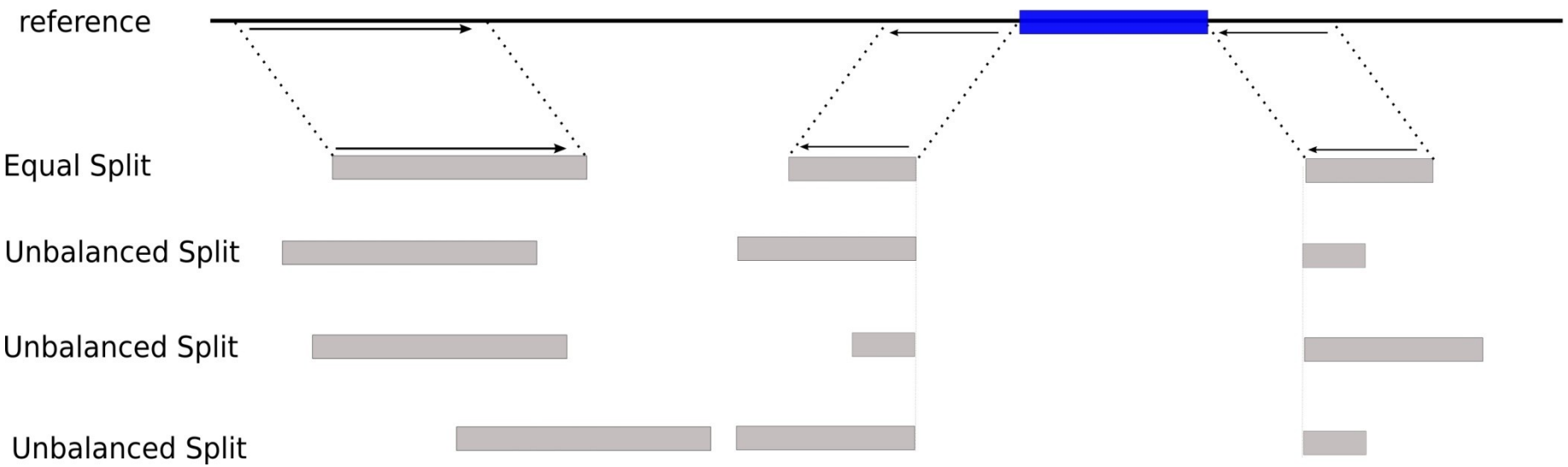
Split Read for detecting Duplications



- Validation from both directions and within the duplicated region.
- Mobile element insertions/transchromosomal events are classified as duplications
- The size of the insertions can be detected unlike large novel insertions.

Karakoc et al., Nature Methods, 2011

Clustering



- Each perfect split defined a cluster region.
- Unbalanced splits around the cluster are inserted to the cluster.
- Split reads can map to other regions of the genome.
  - Perfect/Unbalanced splits can be a member of multiple clusters.
  - Redundancy and unreliable support value.
- Each cluster can be represented as a set with a number of members.
  - 1 perfect split / 3 unbalanced split / 4 total splits

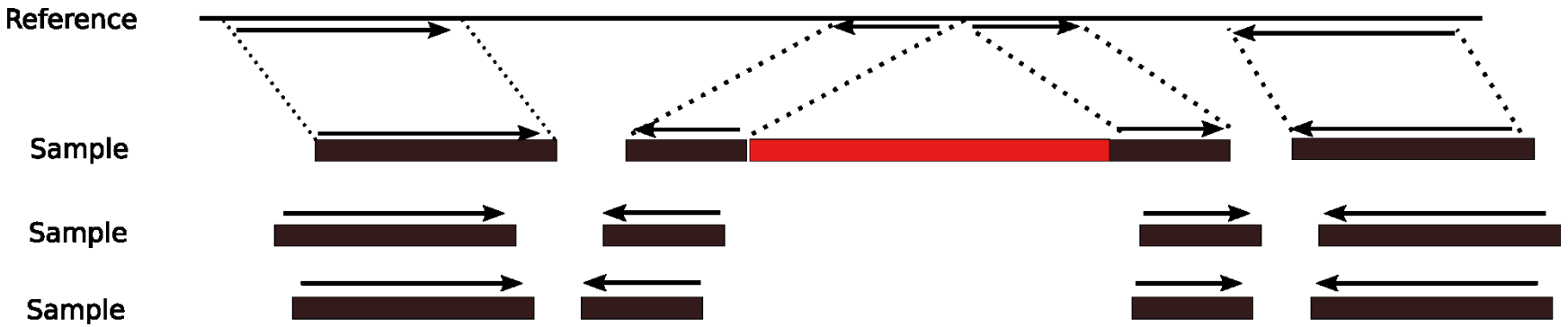
Karakoc et al., Nature Methods, 2011

Detecting correct clusters

- Problem can be represented as set cover problem.
  - Find the minimum number of clusters such that union of them will represent all splits.
- Greedy approach
  - Select the cluster with the maximum elements and report it as an event.
  - Remove all splits that are a member of this cluster from the remaining clusters.
  - Repeat the above procedure until all splits are removed.
  - Logarithmic approximation to optimal.
- Cluster remaining unbalanced splits that does not belong to any cluster in a similar fashion.
  - They can indicate large insertions and deletions without perfect split support.

Karakoc et al., Nature Methods, 2011

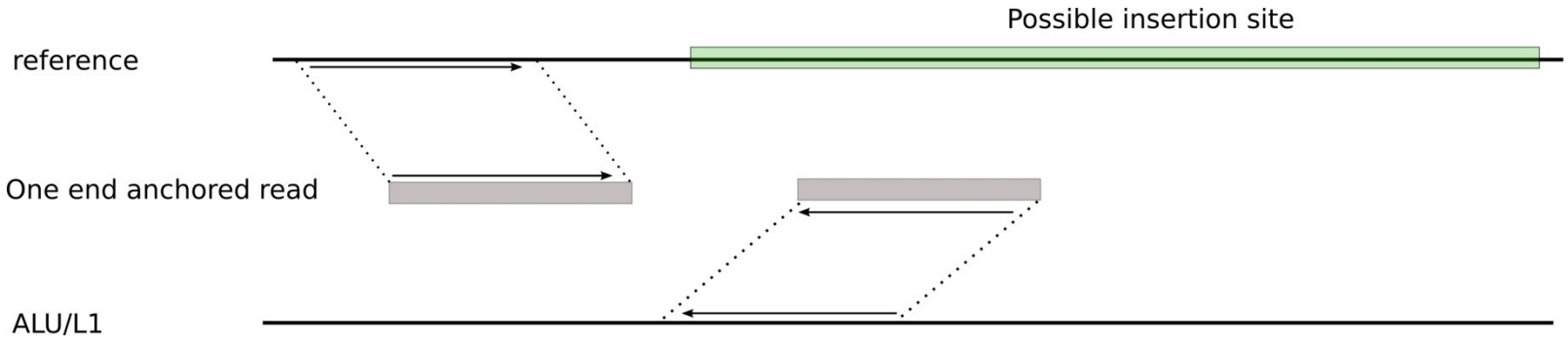
Large Insertions



- There are no perfect splits for large insertions.
  - The other end of the split is in insertion.
- Unbalanced splits around the insertion site.
- After the initial INDEL/SV selection using balanced splits
  - Cluster the remaining unbalanced splits. (within 15bp)
- The content of the Large Insertion can not be identified without assembly.

Karakoc et al., Nature Methods, 2011

Alu/L1 Insertions



- "Transchromosomal" events since the repeat consensus sequences are treated as separate chromosomes
- Possible Alu/L1/SVA insertions
- One end anchored reads
- Novel insertions
- Deletions/Insertions with no perfect split support.

Karakoc et al., Nature Methods, 2011

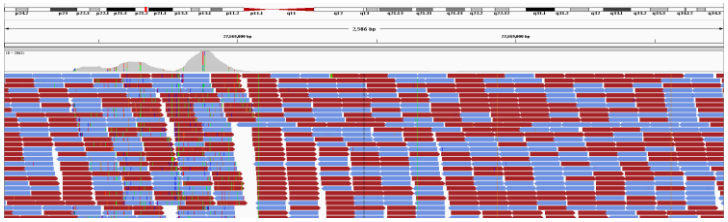
Overview of SPLITREAD

FASTQ Files

All Reads / OEA +INDEL Reads

BAM Files

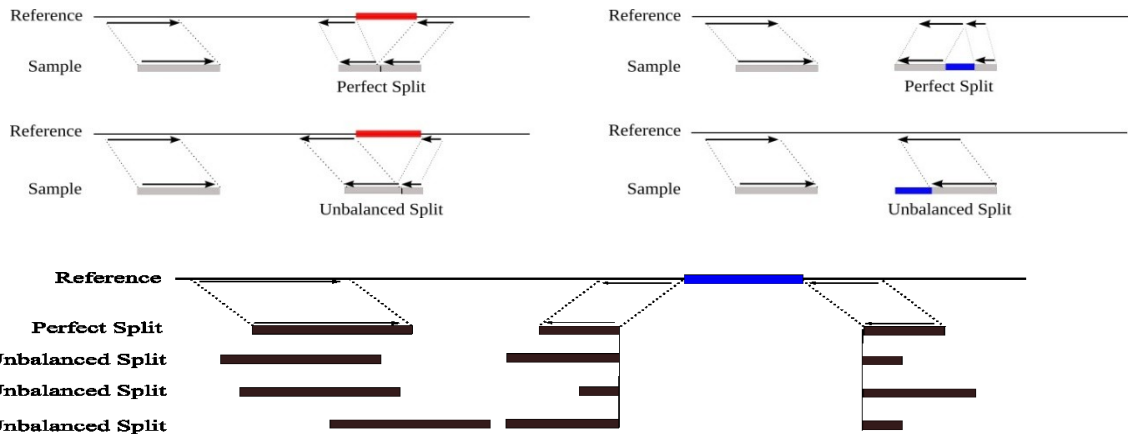
Mapping



mrsFAST
No insertions/deletions
All possible mappings

One End Anchored Reads

Split Read Mapping



Clustering

Maximum Parsimony

Minimum number of events (Deletions + small insertions) with maximum total support

Alu/L1/SVA Insertions + Large Insertions

Using remaining unbalanced reads

Karakoc et al., Nature Methods, 2011

SPLITREAD

- SPLITREAD detects
  - All deletions ranging from 1bp up to 10Mbp.
  - Small insertions that are less than read length.
  - Large insertion sites for insertions larger than read length.
  - Polymorphic Processed Pseudogenes.
  - Mobile element insertions/deletions.
- SPLITREAD can detect
  - Inversions.
  - Tandem duplications.
  - Translocations:
    - Interspersed duplications.

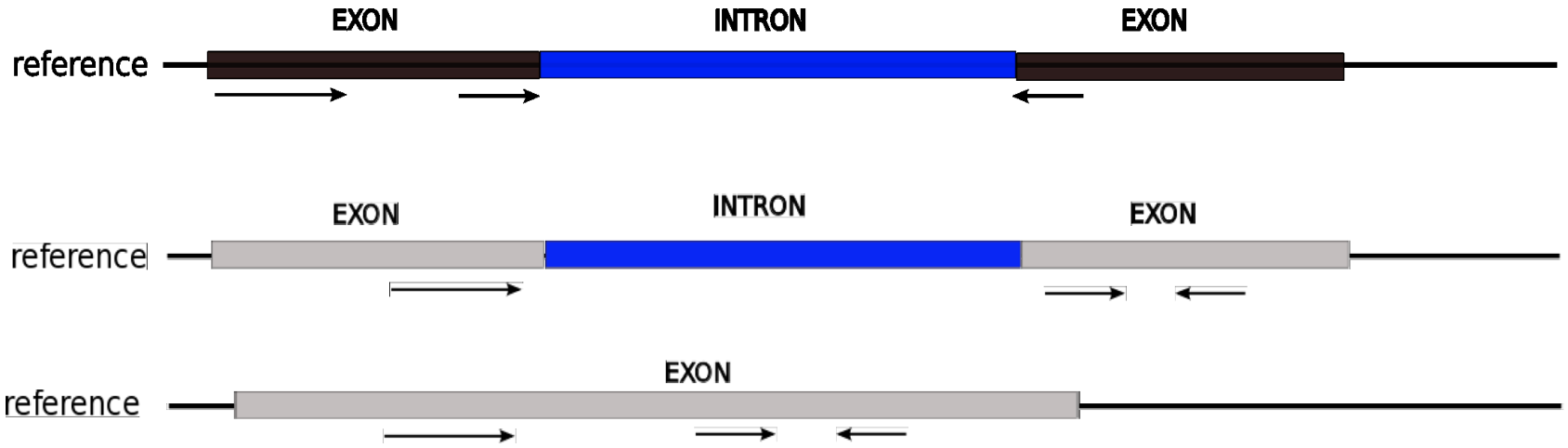
Karakoc et al., Nature Methods, 2011

SPLITREAD

- Better for exome sequencing.
  - 40 CPU - 25-50min per exome.
  - Slow for the whole genome data.
- Using coding regions + Processed pseudogenes in the reference as reference
  - Faster mappings.
  - Reduced specificity for paralogous regions.
- Unmasked reference
  - Large output files. (50GB per sample for exome seq.)
  - Unpredictable memory usage.

Karakoc et al., Nature Methods, 2011

Processed Pseudogenes



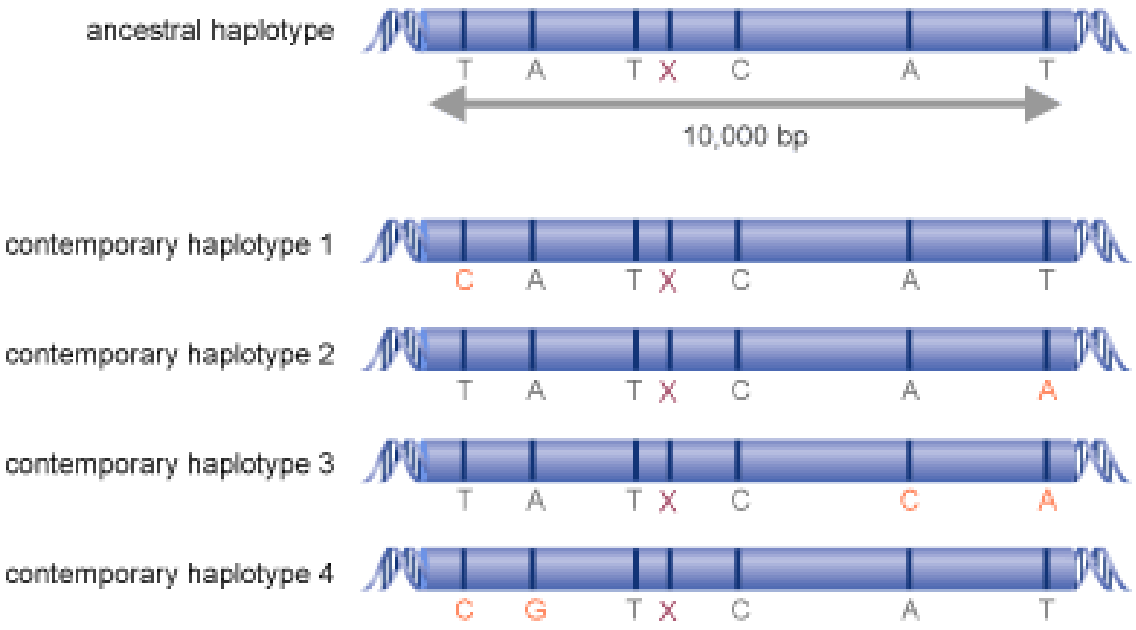
- Processed pseudogenes look like intron deletions with precise breakpoints

Karakoc et al., Nature Methods, 2011

HAPLOTYPE PHASING

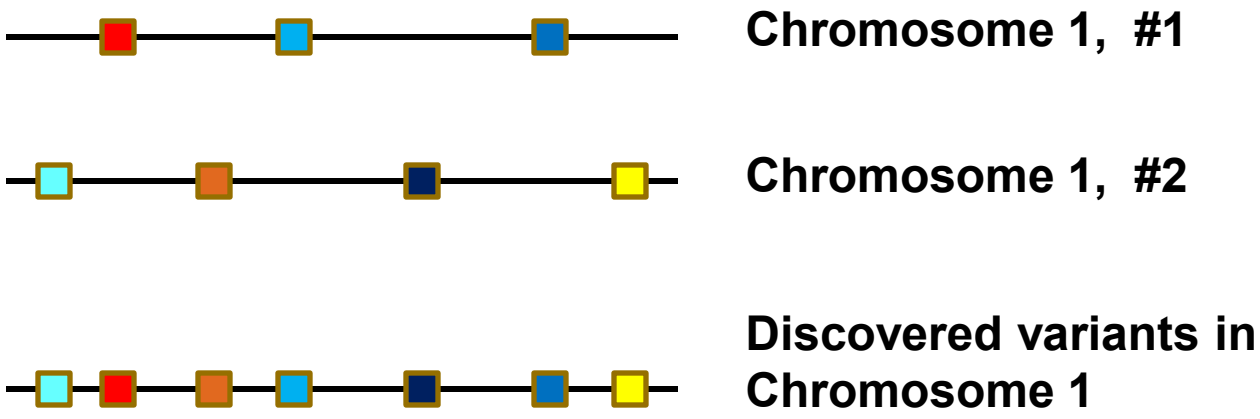
Haplotype

- "Haploid Genotype": *a combination of alleles at multiple loci that are transmitted together on the same chromosome*



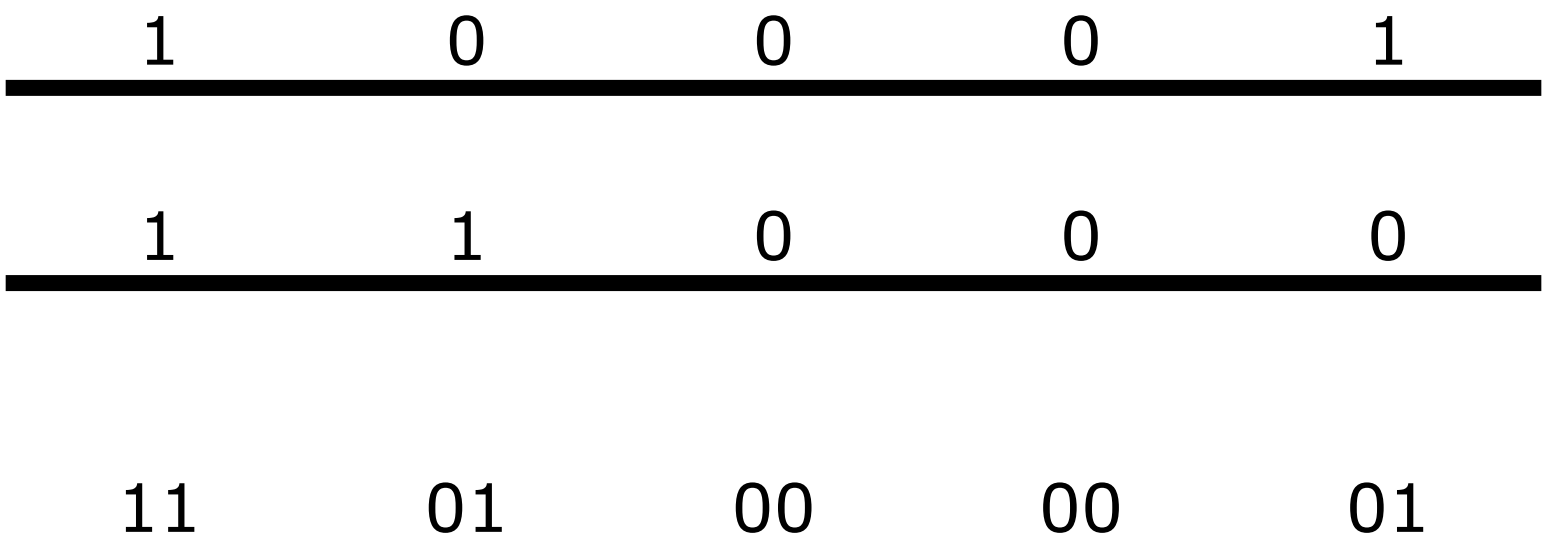
Haplotype resolution

- Variation discovery methods do not directly tell which copy of a chromosome a variant is located
- For heterozygous variants, it gets messy:



Haplotype resolution or haplotype phasing: finding which groups of variants "go together"

Haplotypes and genotypes (1)



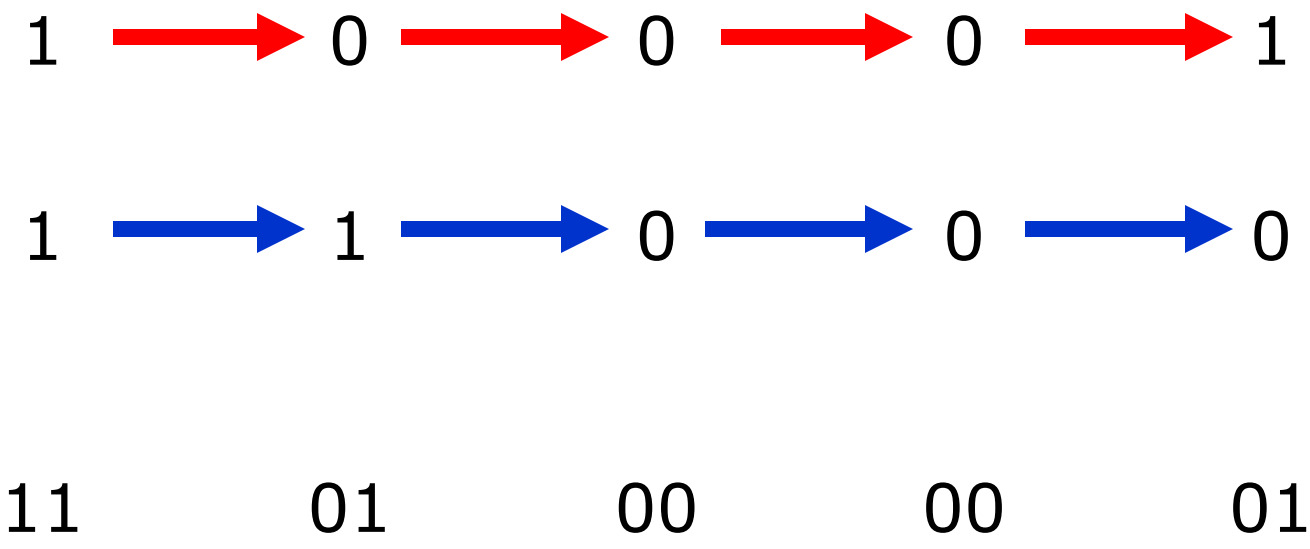
Slide from Andrew Morris

Haplotypes and genotypes (1)

1	0	0	0	1
1	1	0	0	0
11	01	00	00	01

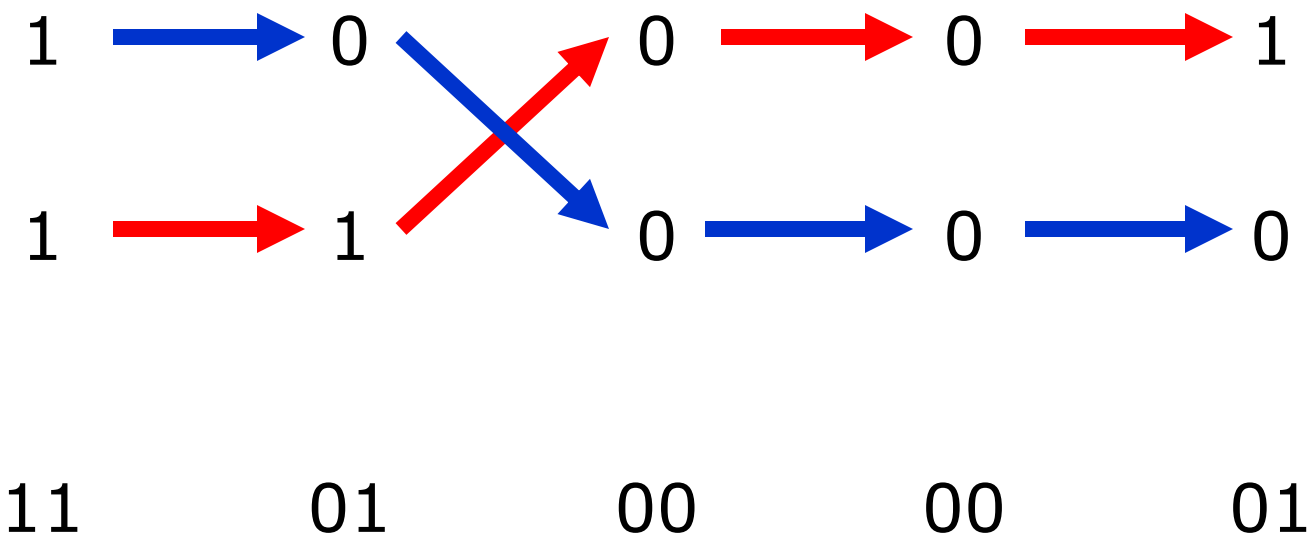
Slide from Andrew Morris

Haplotypes and genotypes (1)



Slide from Andrew Morris

Haplotypes and genotypes (1)



Slide from Andrew Morris

Haplotypes and genotypes (2)

- Individuals that are homozygous at every locus, or heterozygous at just one locus can be **resolved**.
- Individuals that are heterozygous at k loci are consistent with 2^{k-1} configurations of haplotypes.

Slide from Andrew Morris

Why do we need haplotypes?

- Correlation between alleles at closely linked locations
- Fine-scale mapping studies.
- Association studies with multiple markers in candidate genes.
- Investigating patterns of linkage disequilibrium (LD) across genomic regions.
- Inferring population histories.

Slide from Andrew Morris

Simplex family data (1)

00 01 00 11 (M) x 01 11 01 01 (F)

00 01 01 01

Slide from Andrew Morris

Simplex family data (1)

00 01 00 11 x 01 11 01 01

(M) (F)

00 01 01 01

Slide from Andrew Morris

Simplex family data (1)

00 01 00 11 x 01 11 01 01

(M) (F)

00 01 01 01

Inferred haplotypes: 0001 / 0110

Slide from Andrew Morris

Simplex family data (2)

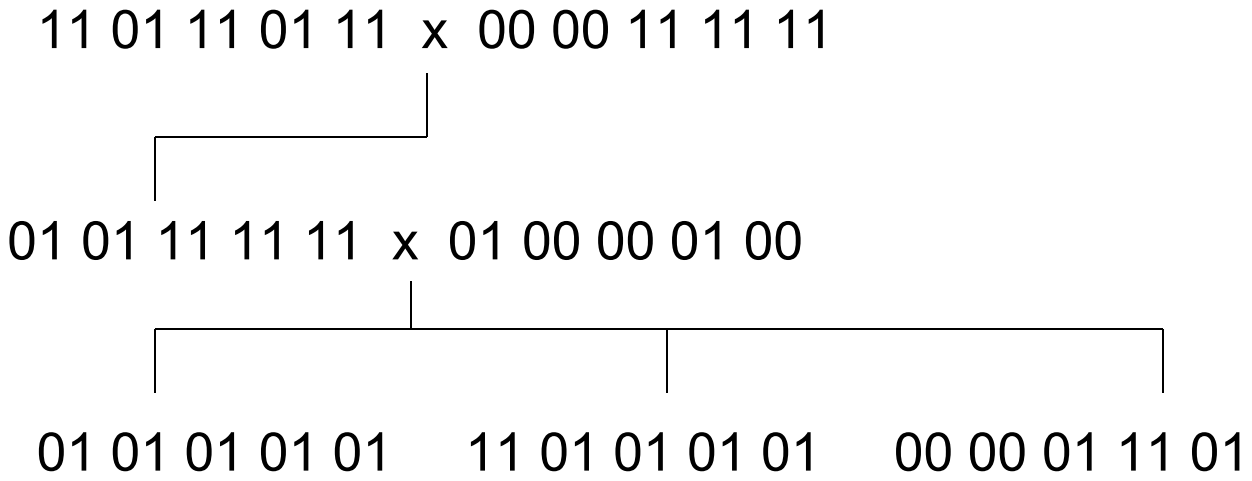
00 01 00 01 (M) x 01 01 00 01 (F)

00 01 00 01

- Cannot be fully resolved…

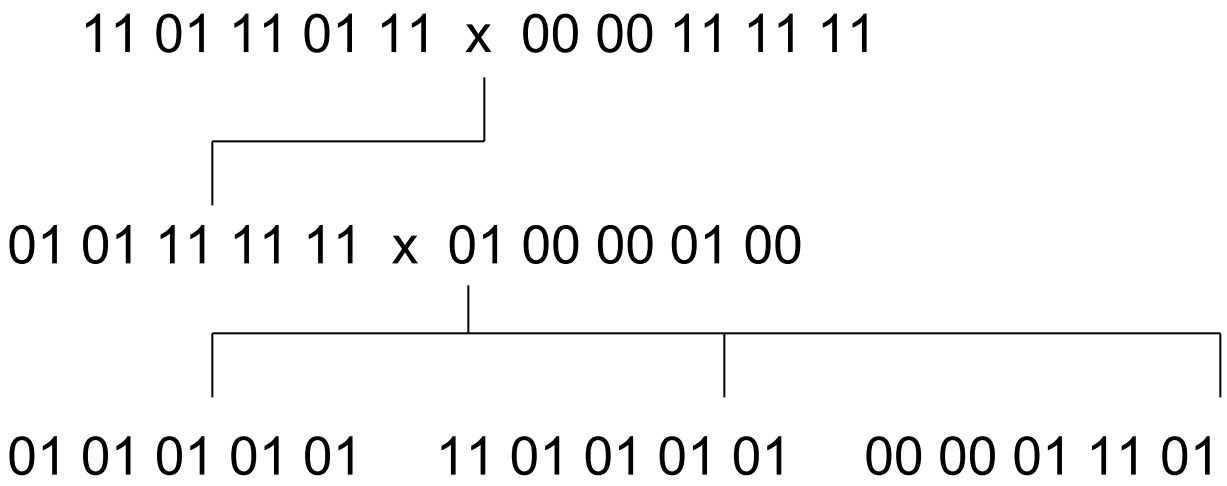
Slide from Andrew Morris

Pedigree data (1)



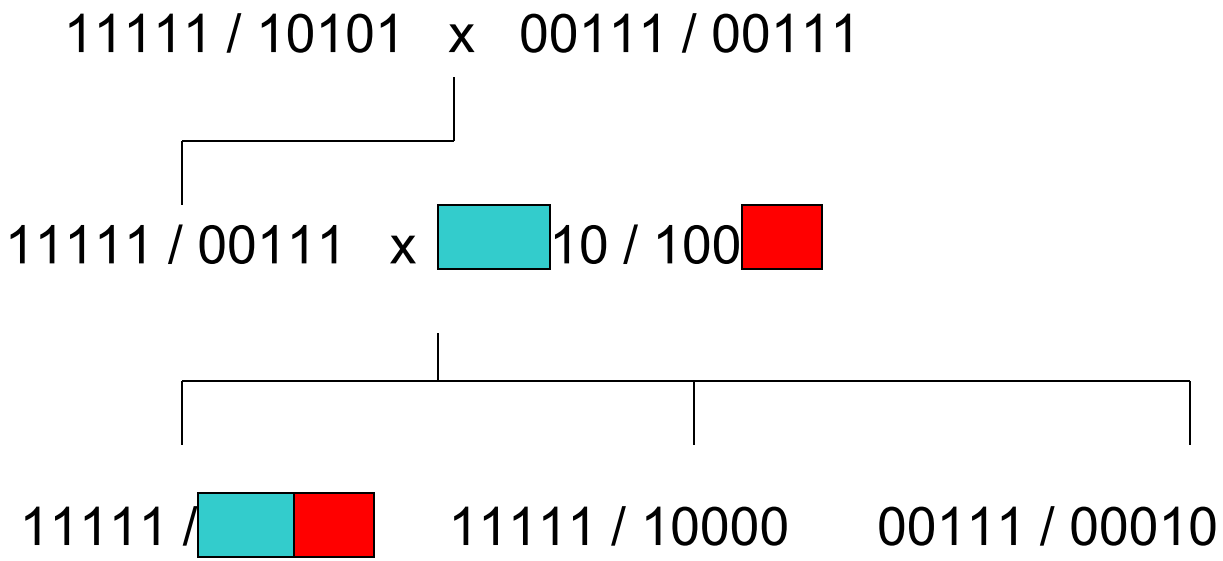
Slide from Andrew Morris

Pedigree data (1)



Slide from Andrew Morris

Pedigree data (1)



Slide from Andrew Morris

Pedigree data (2)

- Many combinations of haplotypes may be consistent with pedigree genotype data.
- Complex computational problem.
- Need to make assumptions about recombination.
- SIMWALK and MERLIN.

Slide from Andrew Morris

Statistical approaches to reconstruct haplotypes in unrelated individuals

- **Parsimony** methods: Clark's algorithm.
- **Likelihood** methods: E-M algorithm.
- **Bayesian** methods: PHASE algorithm.

- **Aims**: **reconstruct** haplotypes and/or **estimate population frequencies**.

Slide from Andrew Morris

Clark's algorithm (1)

- Reconstruct haplotypes in unresolved individuals via parsimony.
- Minimise number of haplotypes observed in sample.
- Microsatellite or SNP genotypes.

Slide from Andrew Morris

Clark's algorithm (2)

1. Search for **resolved** individuals, and record all recovered haplotypes.
2. Compare each **unresolved** individual with list of recovered haplotypes.
3. If a recovered haplotype is identified, individual is resolved.
4. Complimentary haplotype added to list of recovered haplotypes.
5. Repeat 2-4 until all individuals are resolved or no more haplotypes can be recovered.

Slide from Andrew Morris

Example

(A)	00 01 01 00
(B)	00 00 00 00
(C)	00 01 00 00
(D)	01 11 01 11
(E)	00 11 01 01
(F)	01 11 11 00
(G)	00 01 11 01
(H)	00 01 01 11
(I)	00 00 00 00
(J)	00 00 00 11

Slide from Andrew Morris

Example

(A) 00 01 01 00

(B) 00 00 00 00

(C) 00 01 00 00

(D) 01 11 01 11

(E) 00 11 01 01

(F) 01 11 11 00

(G) 00 01 11 01

(H) 00 01 01 11

(I) 00 00 00 00

(J) 00 00 00 11

Slide from Andrew Morris

Example

(A)	00 01 01 00
(B)	**0000 / 0000**
(C)	**0000 / 0100**
(D)	01 11 01 11
(E)	00 11 01 01
(F)	**0110 / 1110**
(G)	00 01 11 01
(H)	00 01 01 11
(I)	**0000 / 0000**
(J)	**0001 / 0001**

- Recovered haplotypes:

  0000

  0100

  0110

  1110

  0001

Slide from Andrew Morris

Example

(A) 00 01 01 00
(B) **0000 / 0000**
(C) **0000 / 0100**
(D) 01 11 01 11
(E) 00 11 01 01
(F) **0110 / 1110**
(G) 00 01 11 01
(H) 00 01 01 11
(I) **0000 / 0000**
(J) **0001 / 0001**

- Recovered haplotypes:

0000
0100
0110
1110
0001

Slide from Andrew Morris

Example

(A) **0000 / 0110**
(B) **0000 / 0000**
(C) **0000 / 0100**
(D) 01 11 01 11
(E) 00 11 01 01
(F) **0110 / 1110**
(G) 00 01 11 01
(H) 00 01 01 11
(I) **0000 / 0000**
(J) **0001 / 0001**

- Recovered haplotypes:

0000 0111
0100
0110
1110
0001

Slide from Andrew Morris

Example

(A) **0000 / 0110**
(B) **0000 / 0000**
(C) **0000 / 0100**
(D) 01 11 01 11
(E) **0100 / 0111**
(F) **0110 / 1110**
(G) 00 01 11 01
(H) 00 01 01 11
(I) **0000 / 0000**
(J) **0001 / 0001**

- Recovered haplotypes:

0000 0111
0100 0011
0110
1110
0001

Slide from Andrew Morris

Example

(A) **0000 / 0110**
(B) **0000 / 0000**
(C) **0000 / 0100**
(D) **0111 / 1101**
(E) **0100 / 0111**
(F) **0110 / 1110**
(G) **0110 / 0011**
(H) **0001 / 0111**
(I) **0000 / 0000**
(J) **0001 / 0001**

- Recovered haplotypes:

0000	0111
0100	0011
0110	1101
1110	
0001	

Slide from Andrew Morris

Example: problem…

(A) **0000 / 0110**
(B) **0000 / 0000**
(C) **0000 / 0100**
(D) 01 11 01 11
(E) **0100 / 0111**
(F) **0110 / 1110**
(G) 00 01 11 01
(H) 00 01 01 11
(I) **0000 / 0000**
(J) **0001 / 0001**

- Recovered haplotypes:

0000 0111
0100 0011
0110
1110
0001

Slide from Andrew Morris

Example: problem…

(A) **0000 / 0110**
(B) **0000 / 0000**
(C) **0000 / 0100**
(D) 01 11 01 11
(E) **0100 / 0111**
(F) **0110 / 1110**
(G) 00 01 11 01
(H) 00 01 01 11
(I) **0000 / 0000**
(J) **0001 / 0001**

- Recovered haplotypes:

0000 0111
0100 0010
0110
1110
0001

Slide from Andrew Morris

Clark's algorithm: problems

- Multiple solutions: try many different orderings of individuals.
- No starting point for algorithm.
- Algorithm may leave many unresolved individuals.
- How to deal with missing data?

Slide from Andrew Morris

E-M algorithm (1)

- **Maximum likelihood** method for population haplotype frequency estimation.
- Allows for the fact that unresolved genotypes could be constructed from many different haplotype configurations.
- Microsatellite or SNP genotypes.

Slide from Andrew Morris

E-M algorithm (2)

- Observed sample of N individuals with genotypes, **G**.
- Unobserved population haplotype frequencies, **h**.
- Unobserved configurations, **H**, consisting of a complimentary haplotype pairs $H_i = \{H_{i1}, H_{i2}\}$.

Slide from Andrew Morris

E-M algorithm (3)

- Likelihood:

$$f(\mathbf{G}|\mathbf{h}) = \prod_k f(G_k|\mathbf{h})$$
$$= \prod_k \sum_i f(G_k|H_i)\, f(H_i|\mathbf{h})$$

where $f(H_i|\mathbf{h}) = f(H_{i1}|\mathbf{h})\, f(H_{i2}|\mathbf{h})$ under Hardy-Weinberg equilibrium.

Slide from Andrew Morris

E-M algorithm (4)

- **Numerical algorithm** used to obtain maximum likelihood estimates of **h**.
- Initial set of haplotype frequencies $\mathbf{h}^{(0)}$.
- Haplotype frequencies $\mathbf{h}^{(t)}$ at iteration t updated from frequencies at iteration t-1 using **Expectation** and **Maximisation** steps.
- Continue until $\mathbf{h}^{(t)}$ has converged.

Slide from Andrew Morris

E-M algorithm: comments

- Can handle missing data.
- For many loci, the number of possible haplotypes is large, so population frequencies are difficult to estimate: re-parameterisation.
- Does not provide reconstructed haplotype configuration for unresolved individuals: can use “maximum likelihood” configuration.

Slide from Andrew Morris

PHASE algorithm (1)

- Treats haplotype configuration for each unresolved individual as an unobserved random quantity.
- Evaluate the conditional distribution, given a sample of unresolved genotype data.
- Microsatellite or SNP genotypes.
- Reconstruction and population haplotype frequency estimation.

Slide from Andrew Morris

PHASE algorithm (2)

- **Bayesian** framework: goal is to approximate **posterior distribution** of haplotype configurations f(**H**|**G**).
- Implements **Markov chain Monte Carlo** (MCMC) methods to sample from f(**H**|**G**): **Gibbs sampling**.
- Start at random configuration.
- Repeatedly select unresolved individuals at random, and sample from their possible haplotype configurations, **assuming all other individuals to be correctly resolved**.

Slide from Andrew Morris

PHASE algorithm: comments

- Allows for uncertainty in haplotype reconstruction in Bayesian framework.
- Can handle missing data.
- Coalescent process does not explicitly allow for recombination, but performs well even when cross-over events occur (up to ~0.1cM).
- Up to 50% more efficient than Clark's algorithm or the E-M algorithm.

Slide from Andrew Morris

PHASE algorithm: output

- "Best" reconstruction output for each individual.
- Uncertainty in reconstruction indicated by system of brackets:
  - [] inferred missing genotype uncertain with posterior probability less than specified threshold;
  - () inferred phase assignment uncertain with posterior probability less than specified threshold.

[0] [(1)] 0 0 1 (0)
[0] [(0)] 1 0 1 (1)

Slide from Andrew Morris

PHASE algorithm: interpretation

- “Best” reconstruction not necessarily correct.
- Uncertain haplotype configurations should be investigated further.
- Effective targeting of additional genotyping costs.

Slide from Andrew Morris

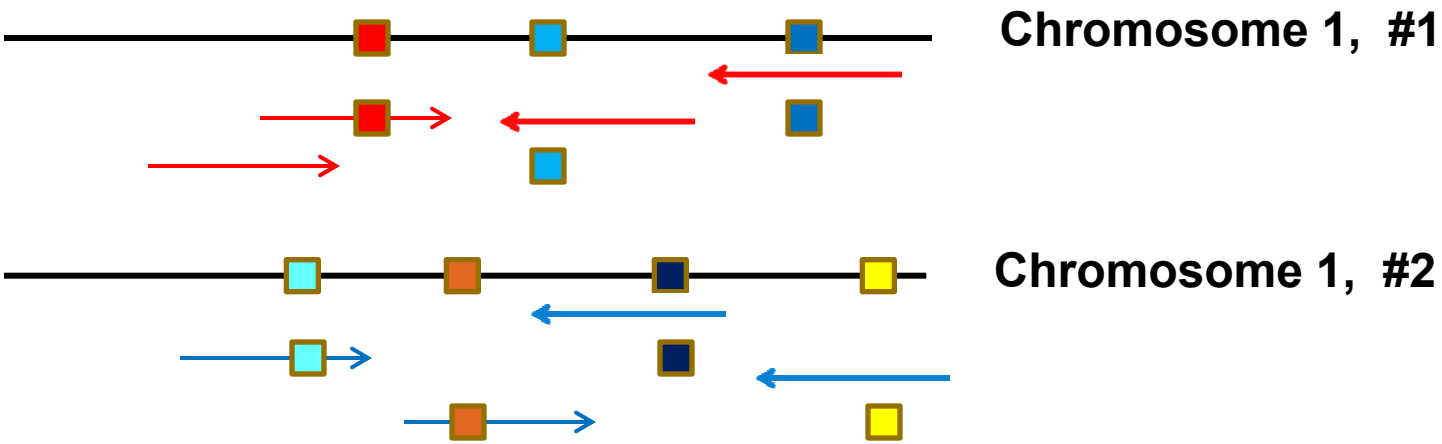
Other Bayesian MCMC algorithms

- HAPLOTYPER
  - Prior model for haplotype frequencies given by Dirichelet distribution.
  - Deals with large number of SNPs by partition ligation.
  - Outputs “best” reconstruction with uncertainty measured by posterior probability.
- HAPMCMC
  - Log-linear prior model for haplotype frequencies incorporating interactions corresponding to first order LD between SNPs.
  - Designed specifically for investigating LD across small genomic regions.

Slide from Andrew Morris

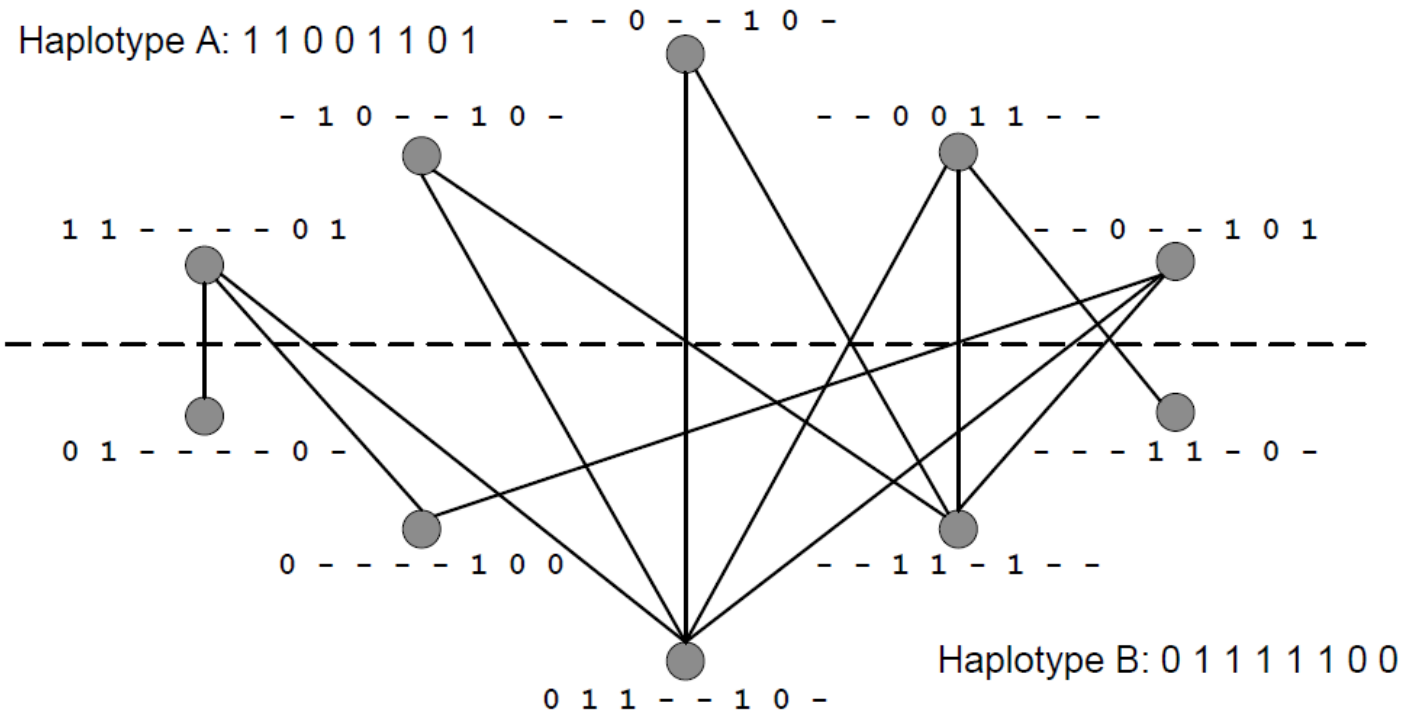
Haplotype phasing with PE sequences

PE sequences are from the same molecule, thus same haplotype



- Build initial shared haplotypes from PE reads
- Assemble shared haplotypes to get larger phased blocks

Fragment conflict graph



Two fragments conflict if they cover a common SNP with different alleles

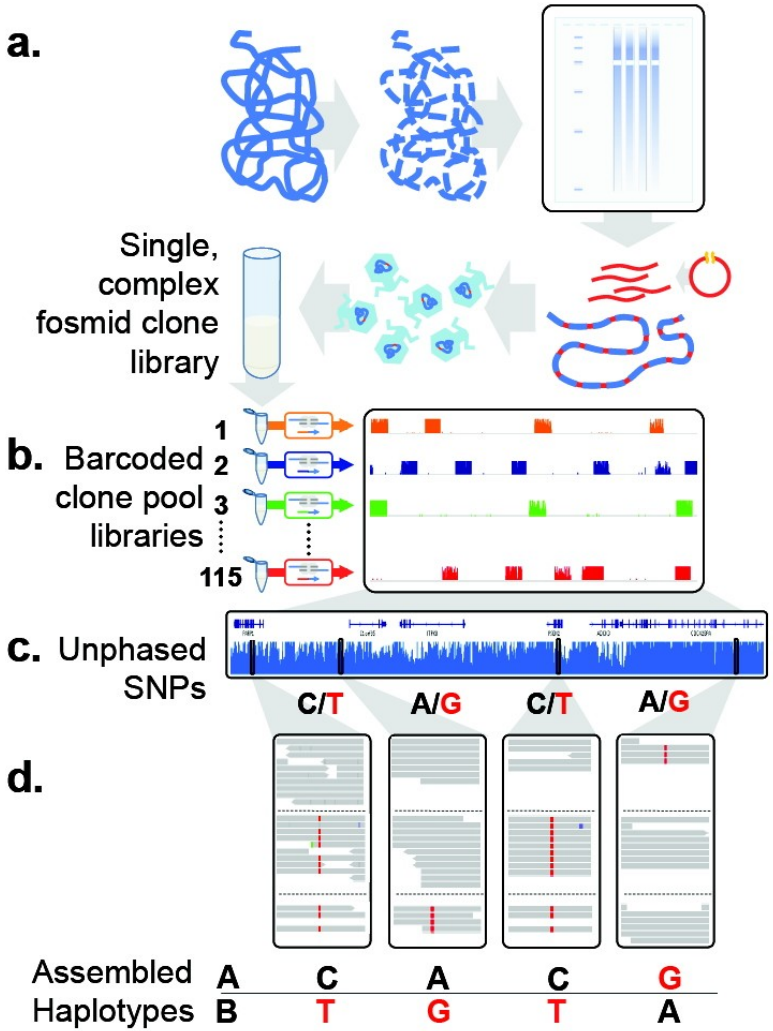
Halldorsson et al., PSB 2011

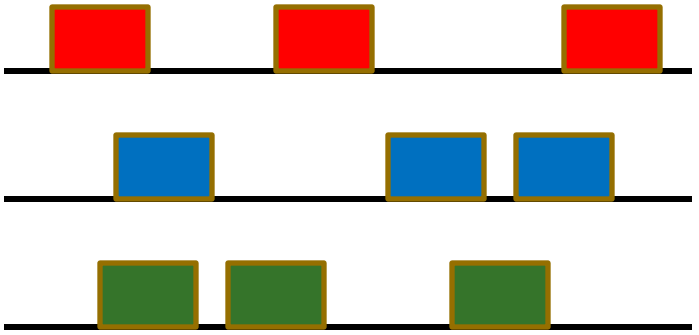
Pooled clone sequencing

- Instead of short paired-ends, use fosmids (40 kb)
  - Build fosmid library
  - Dilute the concentration of the library to cover the genome ~5X
  - Merge ~5000 fosmids in a pool
    - Total 114 pools
  - Sequence pools & separate fosmids *in silico*

Kitzman et al., Nature Biotechnology, 2011

Pooled clone sequencing





- Each fosmid represents one haplotype
- Resolve in ~40 kb blocks
- Extend blocks by overlapping fosmids in different pools