CS681: Advanced Topics in Computational Biology

Week 2, Lectures 2-3

Can Alkan

EA224

calkan@cs.bilkent.edu.tr

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

Microarrays (refresher)

- Targeted approach for:
  - SNP / indel detection/genotyping
    - Screen for mutations that cause disease
  - Gene expression profiling
    - Which genes are expressed in which tissue?
    - Which genes are expressed “together”
    - Gene regulation (chromatin immunoprecipitation)
  - Fusion gene profiling
  - Alternative splicing
  - CNV discovery & genotyping
  - ….
- 50K to 4.3M probes per chip

Gene clustering (revisit)

- Clustering genes with respect to their expression status:
  - Not the signal clustering on microarray
  - Clustering the information gained by microarray
- Assume you did 5 experiments in t_1 to t_5
  - Measure expression 5 times (different conditions / cell types, etc.)

Gene clustering (revisit)

Experiment	1	2	3	4	5
Genes	g1, g5	g2, g3	g1,g3, g4, g5	g2, g3, g4	g1, g4, g5

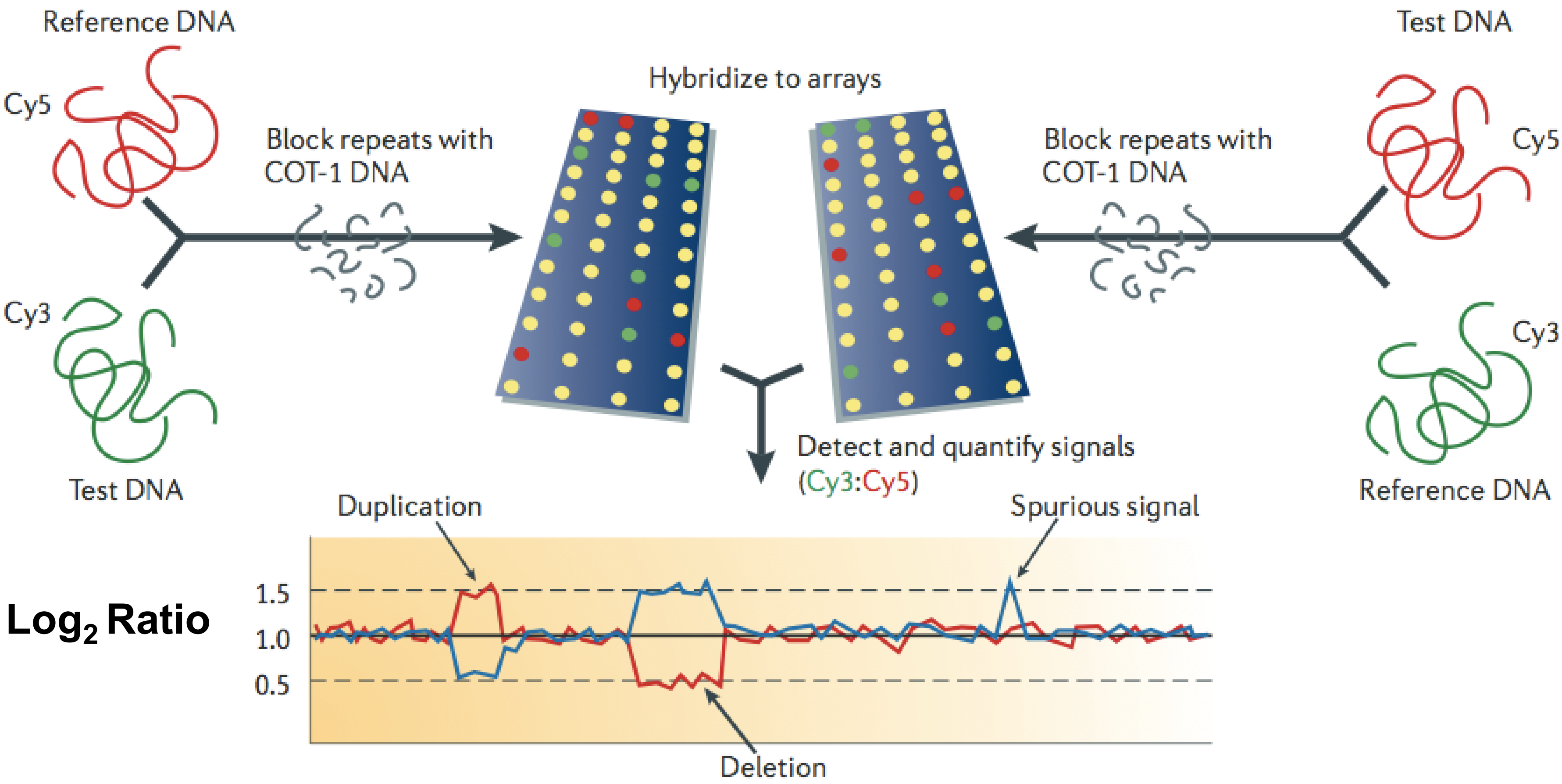
Genes	1	2	3	4	5
1	-				
2	0	-			
3	1	2	-		
4	1	1	1	-	
5	3	0	1	2	-

(g1,g5), g4) and (g2, g3)

CNV Genotyping vs Discovery

- Discovery is done per-sample, genome-wide, and without assumptions about breakpoints
  - consequently, sensitivity is compromised to facilitate tolerable FDR
- Genotyping is targeted to known loci and applies to all samples simultaneously
  - good sensitivity **and** specificity are required
  - knowledge that a CNV is likely to exist and borrowing information across samples reduces the number of probes needed

Array CGH



Array comparative genomic hybridization

Feuk et al, Nat Rev. Genet. 2006

CNV detection with Array CGH

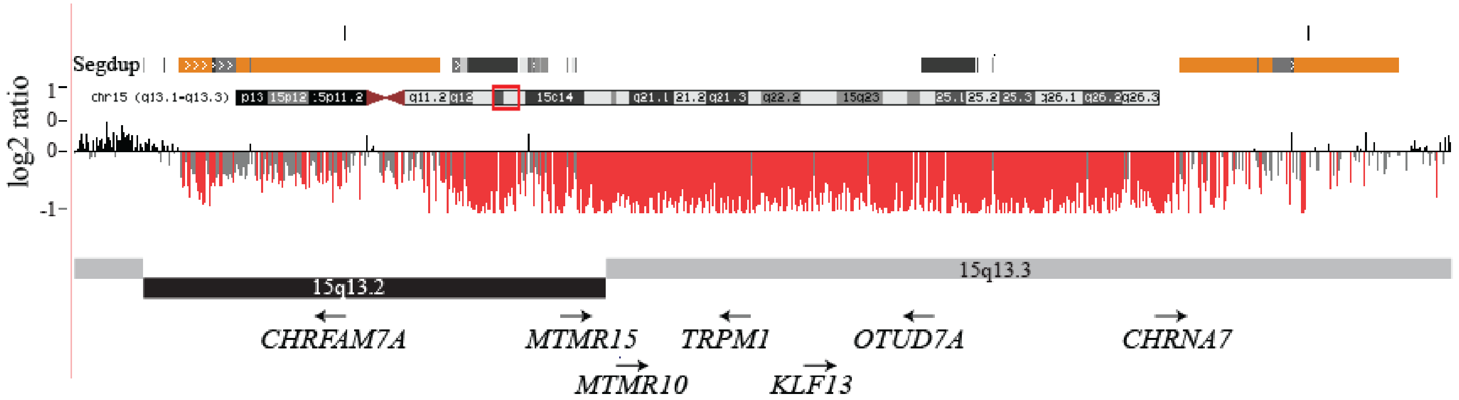
- Signal intensity log_2 ratio:
  - No difference: $log_2(2/2) = 0$
  - Hemizygous deletion in test: $log_2(1/2) = -1$
  - Duplication (1 extra copy) in test: $log_2(3/2) = 0.59$
  - Homozygous duplication (2 extra copies) in test: $log_2(4/2) = 1$
- HMM-based segmentation algorithms to call CNVs
  - HMMSeg: Day et al, Bioinformatics 2007

CNV detection with Array CGH

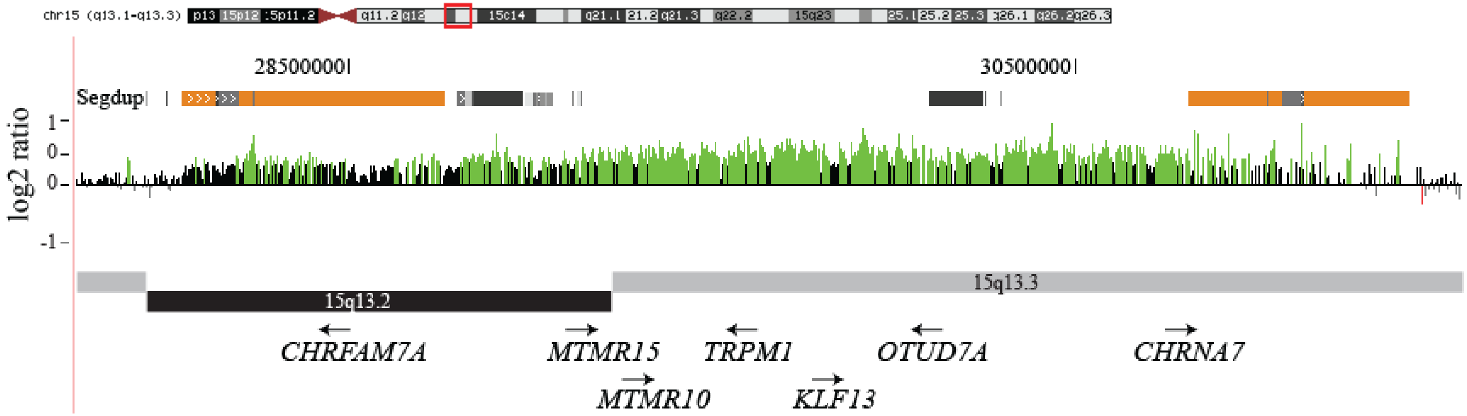
- Advantages:
  - Low cost, high throughput screening of deletions, insertions (when content is known), and copy-number polymorphism
  - Robust in CNV detection in unique DNA
- Disadvantages:
  - Targeted regions only, needs redesign for “new” genome segments of interest
  - Unreliable and noisy in high-copy duplications
  - *Reference effect:* All calls are made against a “reference sample”
  - Inversions, and translocations are not detectable

Array CGH Data

Deletion

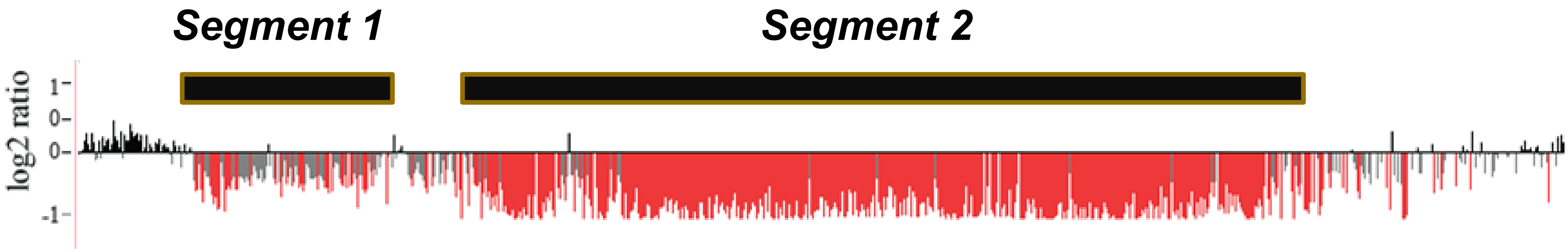


Duplication



Analyzing Array CGH: Segmentation

- "Summarization"
- Partitioning a continuous information into discrete sets: *segments*
- Hidden Markov Models



Hidden Markov Model (HMM)

- Can be viewed as an abstract machine with *k hidden* states that emits symbols from an alphabet Σ.
- Each state has its own probability distribution, and the machine switches between states according to this probability distribution.
- While in a certain state, the machine makes 2 decisions:
  - What state should I move to next?
  - What symbol - from the alphabet Σ - should I emit?

Why “Hidden”?

- Observers can see the emitted symbols of an HMM but have *no ability to know which state the HMM is currently in*.
- Thus, the goal is to infer the most likely hidden states of an HMM based on the given sequence of emitted symbols.

HMM Parameters

Σ: set of emission characters.

Ex.: Σ = {H, T} for coin tossing

Σ = {1, 2, 3, 4, 5, 6} for dice tossing

Q: set of hidden states, each emitting symbols from Σ.

Q={F,B} for coin tossing

HMM Parameters (cont'd)

$A = (a_{kl})$: a |Q| x |Q| matrix of probability of changing from state *k* to state *l*.

$$a_{FF} = 0.9 \qquad a_{FB} = 0.1$$

$$a_{BF} = 0.1 \qquad a_{BB} = 0.9$$

$E = (e_k(b))$: a |Q| x |Σ| matrix of probability of emitting symbol *b* while being in state *k*.

$$e_F(0) = \tfrac{1}{2} \qquad e_F(1) = \tfrac{1}{2}$$

$$e_B(0) = \tfrac{1}{4} \qquad e_B(1) = \tfrac{3}{4}$$

Fair Bet Casino Problem

- The game is to flip coins, which results in only two possible outcomes: **H**ead or **T**ail.
- The **F**air coin will give **H**eads and **T**ails with same probability ½.
- The **B**iased coin will give **H**eads with prob. ¾.

The "Fair Bet Casino" (cont'd)

- Thus, we define the probabilities:
  - $P(H|F) = P(T|F) = \frac{1}{2}$
  - $P(H|B) = \frac{3}{4}$, $P(T|B) = \frac{1}{4}$
  - The dealer/cheater changes between Fair and Biased coins with probability 10%

The Fair Bet Casino Problem

- **Input:** A sequence $x = x_1x_2x_3\ldots x_n$ of coin tosses made by two possible coins (***F*** or ***B***).

- **Output:** A sequence $\pi = \pi_1\ \pi_2\ \pi_3\ldots\ \pi_n$, with each π_i being either *F* or *B* indicating that x_i is the result of tossing the Fair or Biased coin respectively.

HMM for Fair Bet Casino

- The *Fair Bet Casino* in *HMM* terms:

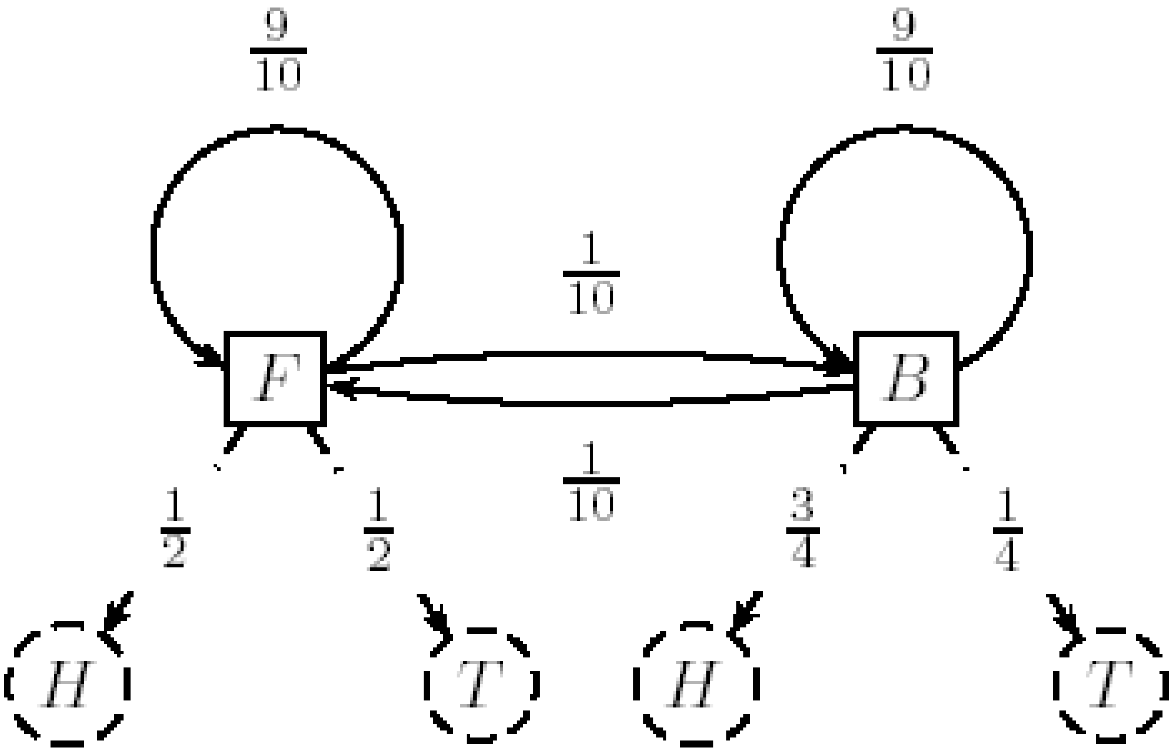
  Σ = {0, 1} (0 for ***T***ails and 1 ***H***eads)

  Q = {*F*,*B*} – *F* for Fair & *B* for Biased coin.
- Transition Probabilities *A* *** Emission Probabilities *E*

	Fair	Biased
Fair	$a_{FF} = 0.9$	$a_{FB} = 0.1$
Biased	$a_{BF} = 0.1$	$a_{BB} = 0.9$

	Tails(0)	Heads(1)
Fair	$e_F(0) = ½$	$e_F(1) = ½$
Biased	$e_B(0) = ¼$	$e_B(1) = ¾$

HMM for Fair Bet Casino (cont'd)



HMM model for the *Fair Bet Casino* Problem

Hidden Paths

- A *path* $\pi = \pi_1 \ldots \pi_n$ in the HMM is defined as a sequence of states.
- Consider path π = FFFBBBBBFFF and sequence x = 01011101001

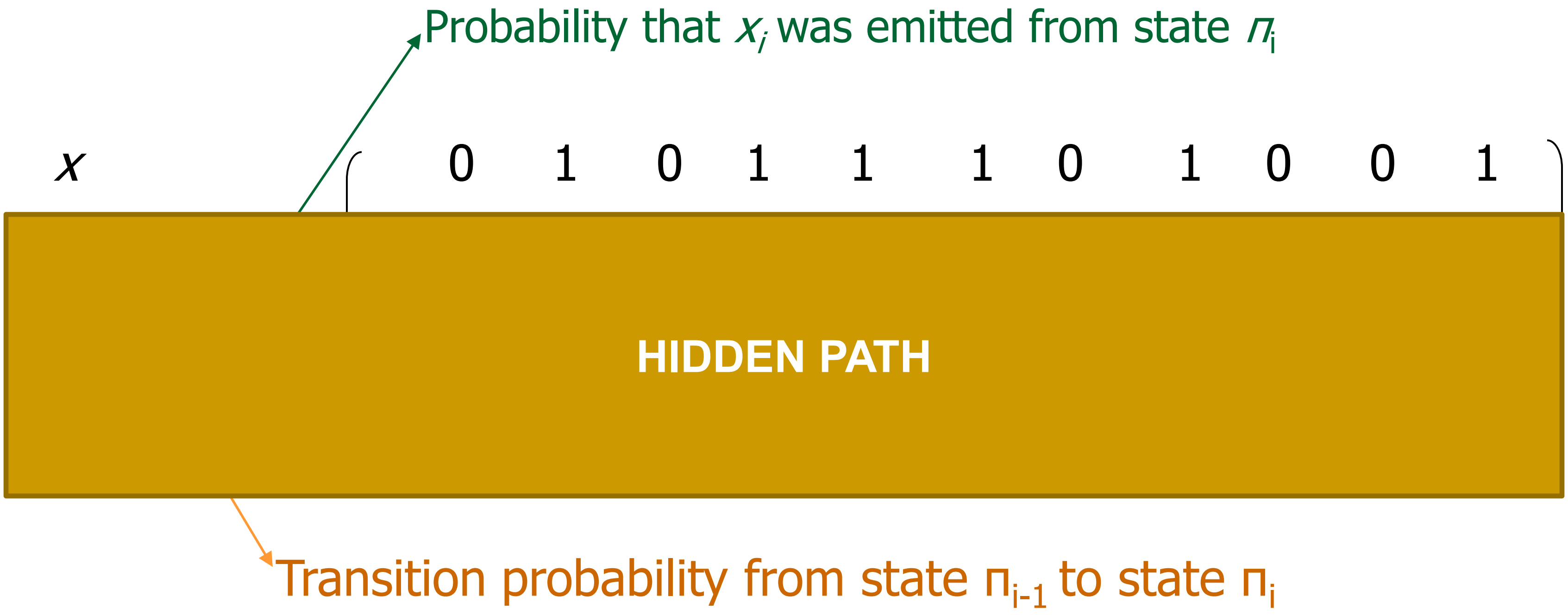
Probability that x_i was emitted from state π_i

x	0	1	0	1	1	1	0	1	0	0	1
π =	F	F	F	B	B	B	B	B	F	F	F
$P(x_i \mid \pi_i)$	½	½	½	¾	¾	¾	¼	¾	½	½	½
$P(\pi_{i-1} \rightarrow \pi_i)$	½	$^9/_{10}$	$^9/_{10}$	$^1/_{10}$	$^9/_{10}$	$^9/_{10}$	$^9/_{10}$	$^9/_{10}$	$^1/_{10}$	$^9/_{10}$	$^9/_{10}$

Transition probability from state π_{i-1} to state π_i

Hidden Paths

- A *path* $\pi = \pi_1 \ldots \pi_n$ in the HMM is defined as a sequence of states.
- Consider path π = FFFBBBBBFFF and sequence x = 01011101001



Probability that x_i was emitted from state π_i

x 0 1 0 1 1 1 0 1 0 0 1

HIDDEN PATH

Transition probability from state π_{i-1} to state π_i

$P(x|\pi)$ Calculation

- $P(x|\pi)$: Probability that sequence *x* was generated by the path π:

$$P(x|\pi) = P(\pi_0 \to \pi_1) \cdot \prod_{i=1}^{n} P(x_i|\pi_i) \cdot P(\pi_i \to \pi_{i+1})$$

$$= a_{\pi_0, \pi_1} \cdot \prod e_{\pi_i}(x_i) \cdot a_{\pi_i, \pi_{i+1}}$$

P(*x*|π) Calculation

- P(*x*|π): Probability that sequence *x* was generated by the path *π:*

$$P(x|\pi) = P(\pi_0 \rightarrow \pi_1) \cdot \prod_{i=1}^{n} P(x_i | \pi_i) \cdot P(\pi_i \rightarrow \pi_{i+1})$$

$$= a_{\pi_0, \pi_1} \cdot \prod e_{\pi_i}(x_i) \cdot a_{\pi_i, \pi_{i+1}}$$

$$= \prod e_{\pi_{i+1}}(x_{i+1}) \cdot a_{\pi_i, \pi_{i+1}}$$

if we count from *i=0* instead of *i=1*

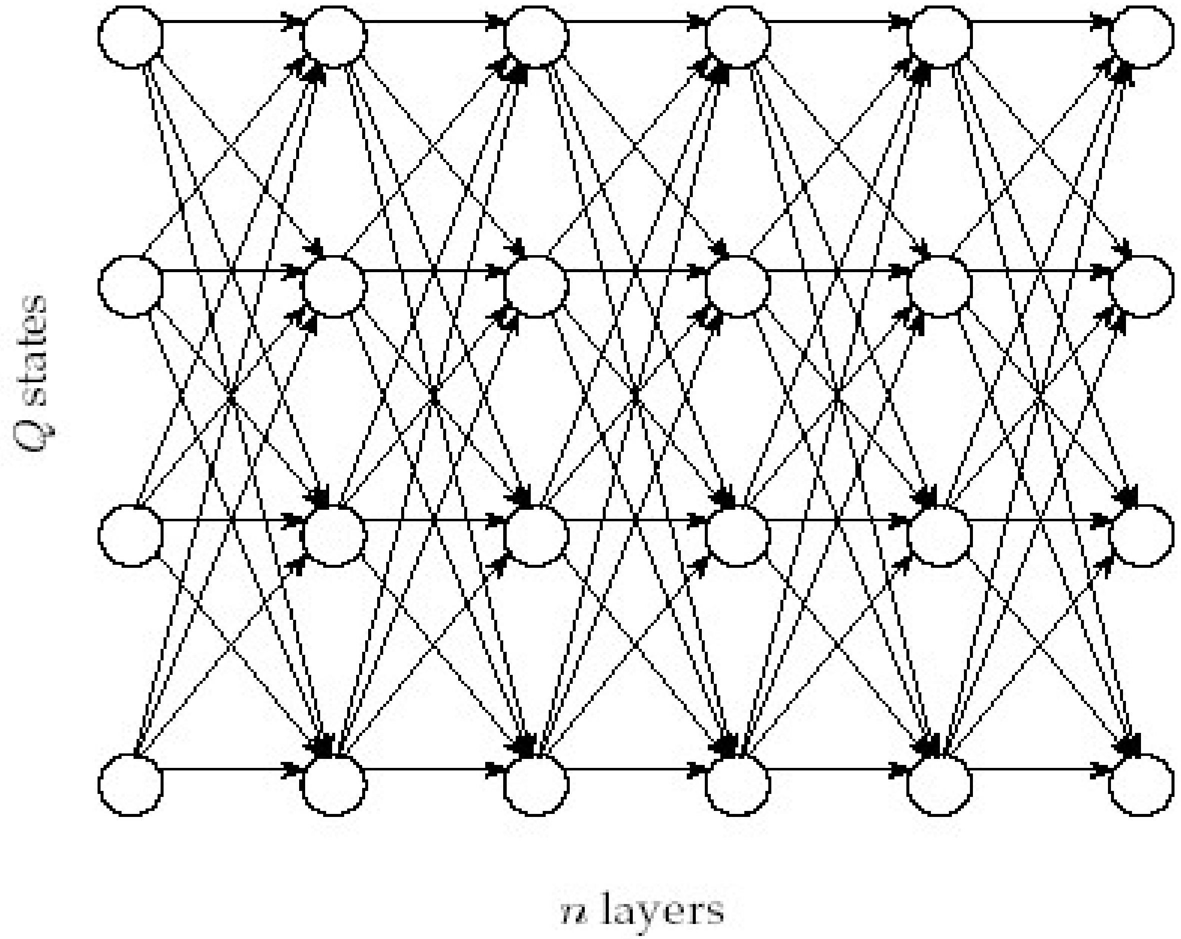
Decoding Problem

- **Goal:** Find an optimal hidden path of states given observations.
- **Input:** Sequence of observations $x = x_1 \ldots x_n$ generated by an HMM $M(\Sigma, Q, A, E)$
- **Output:** A path that maximizes $P(x|\pi)$ over all possible paths π.

Manhattan grid for Decoding Problem

- Andrew Viterbi used the Manhattan grid model to solve the *Decoding Problem*.
- Every choice of $\pi = \pi_1 \ldots \pi_n$ corresponds to a path in the graph.
- The only valid direction in the graph is *eastward.*
- This graph has $|Q|^2(n-1)$ edges.
    - |Q|=number of possible states; n=path length

Edit Graph for Decoding Problem



Decoding Problem as Finding a Longest Path in a DAG

- The *Decoding Problem* is reduced to finding a longest path in the *directed acyclic graph (DAG)* above.

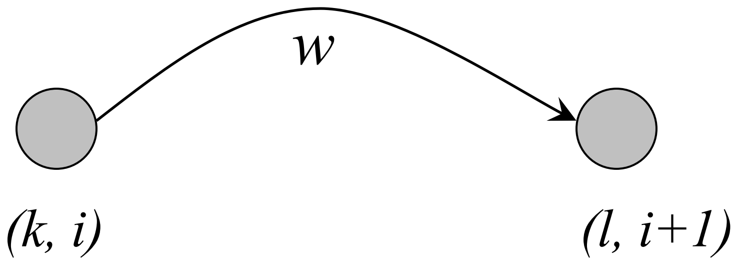
- **Notes:** the length of the path is defined as the *product* of its edges' weights, not the *sum.*

Decoding Problem (cont'd)

- Every path in the graph has the probability $P(x|\pi)$.
- The Viterbi algorithm finds the path that maximizes $P(x|\pi)$ among all possible paths.
- The Viterbi algorithm runs in $\boldsymbol{O(n|Q|^2)}$ time.

Decoding Problem: weights of edges

i-th term = $e_{\pi_i}(x_i) \,.\, a_{\pi_i, \pi_{i+1}}$ = $\mathbf{e_l(x_{i+1}).\ a_{kl}}$ *for* $\pi_i = k,\ \pi_{i+1} = l$



The weight $\mathbf{w = e_l(x_{i+1}).\ a_{kl}}$

Decoding Problem (cont'd)

- Initialization:
  - $s_{begin,0} = 1$
  - $s_{k,0}$ = 0 for $k \neq begin$.
- Let π^* be the optimal path. Then,

$$\mathrm{P}(x|\pi^*) = \max_{k \in Q} \{s_{k,n} \cdot a_{k,end}\}$$

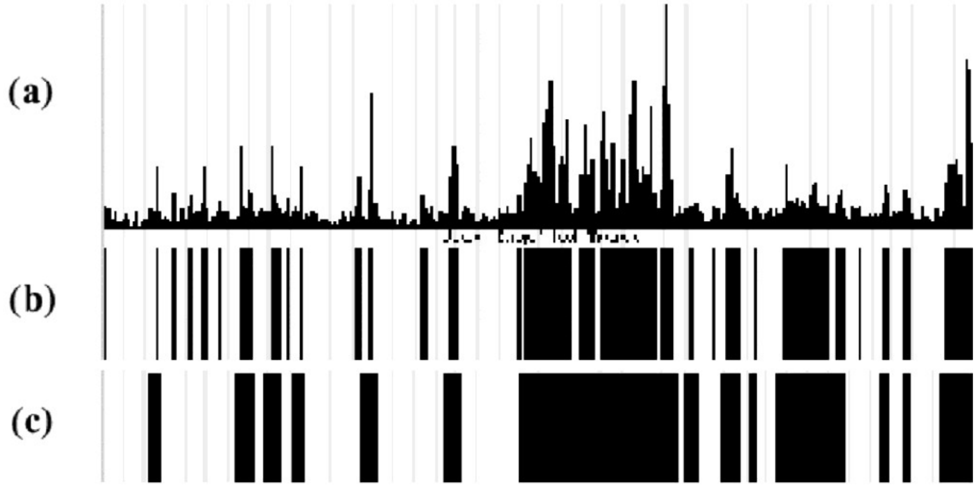
Viterbi Algorithm

- The value of the product can become extremely small, which leads to overflowing.
- To avoid overflowing, use log value instead.

$$s_{k,i+1} = \log e_l(x_{i+1}) + \max_{k \in Q} \{s_{k,i} + \log(a_{kl})\}$$

HMM for segmentation

- HMMSeg (Day et al., Bioinformatics, 2007)
  - general-purpose
- Two states: up/down
- Viterbi decoding
- Wavelet smoothing (Percival & Walden, 2000)



Raw

2-state segmentation

Wavelet smoothing

Multi datatype functional domains

DNA replication timing

RNA transcription

Histone modification (-)

Histone modification (+)

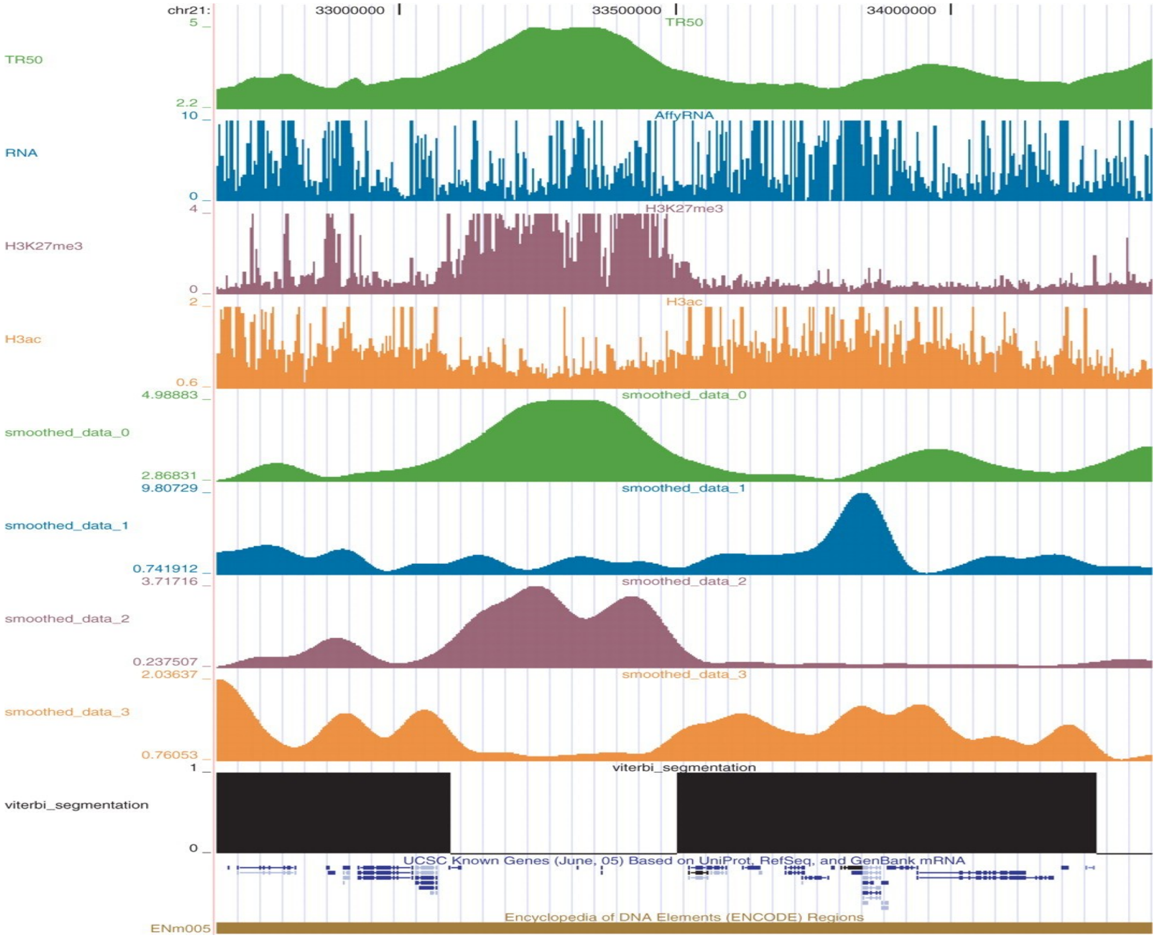
DNA replication timing

RNA transcription

Histone modification (-)

Histone modification (+)

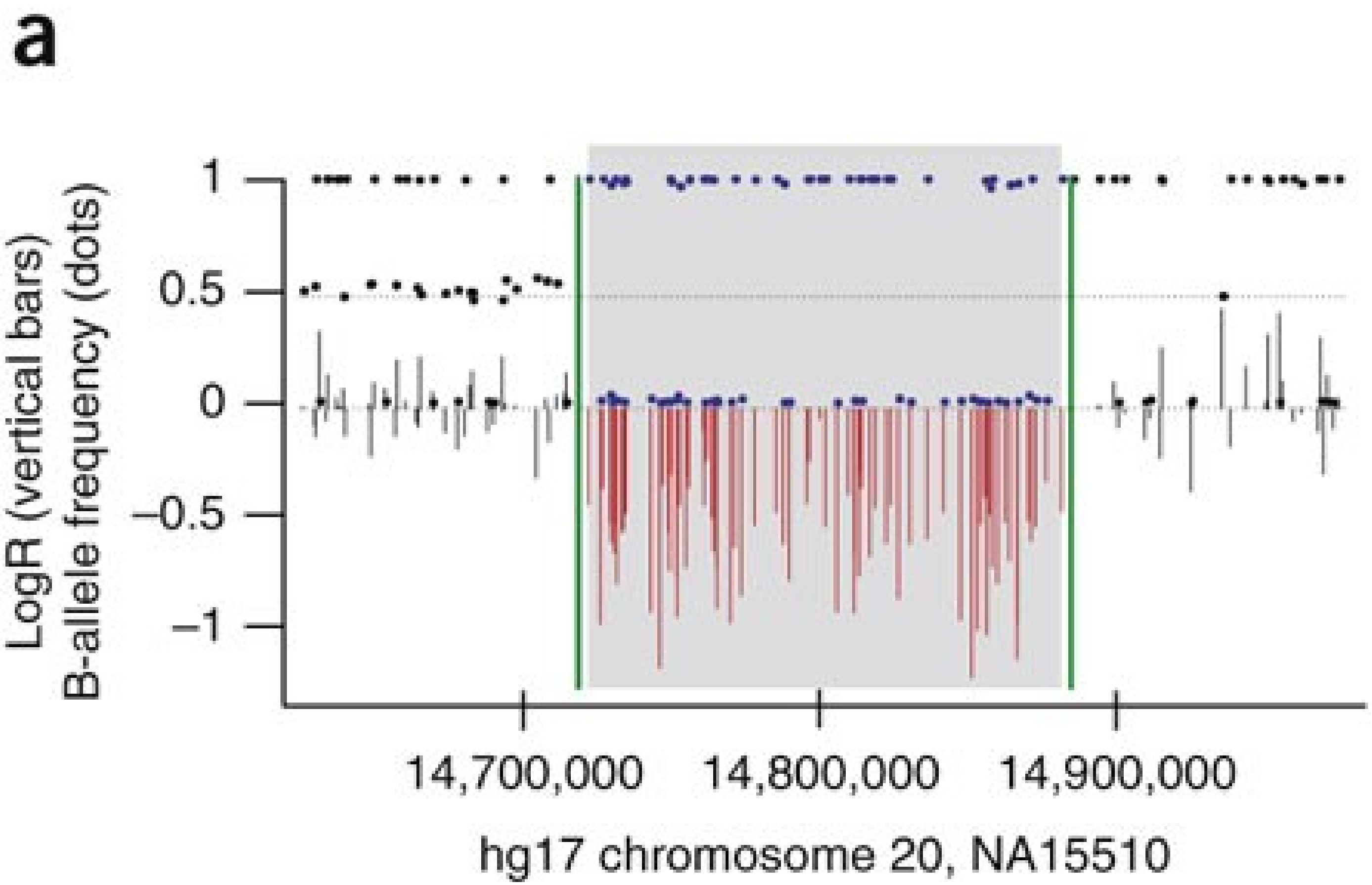
Viterbi segmentation



CNVs using SNP microarrays

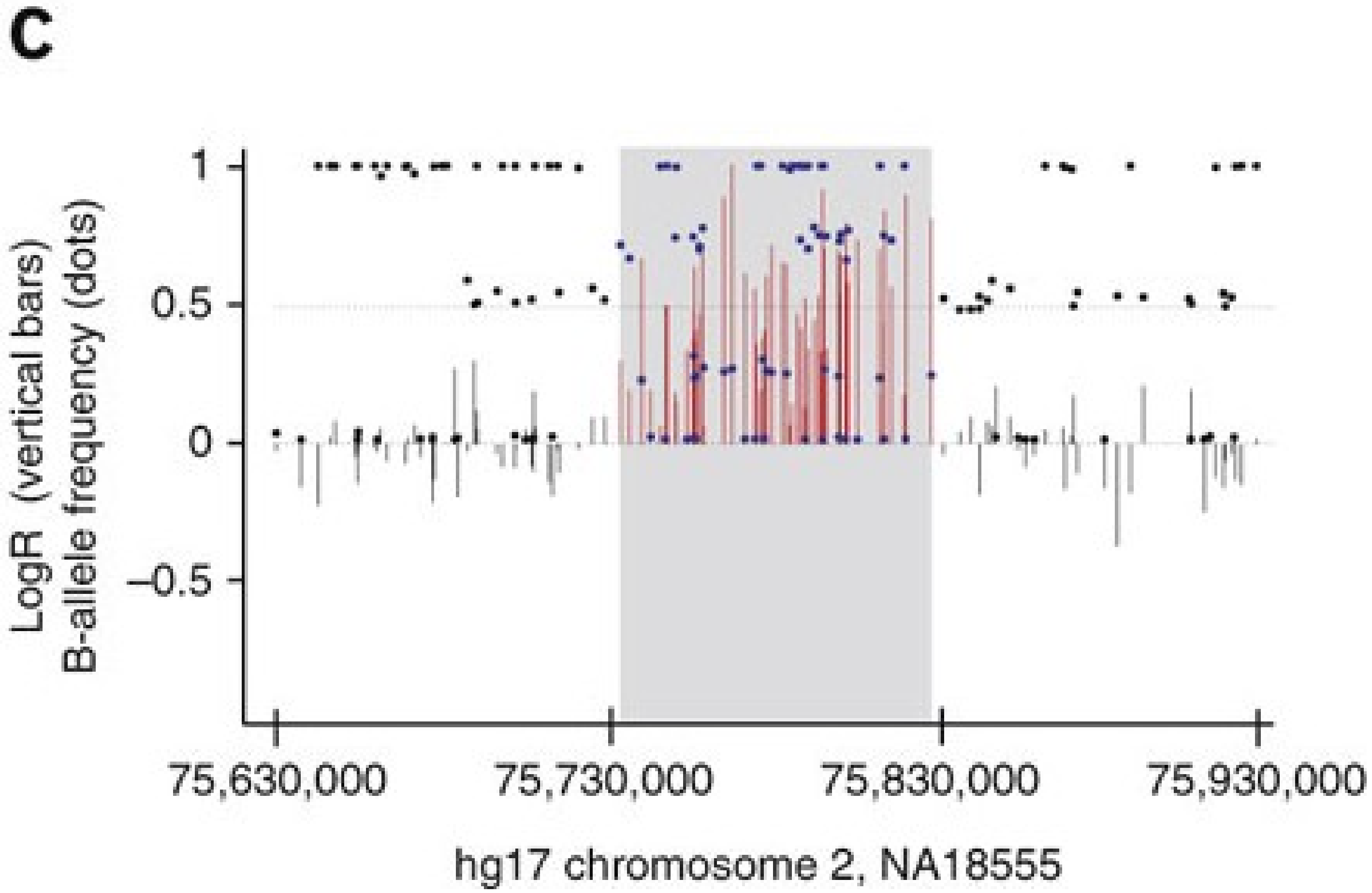
- Input: set of SNPs from a microarray experiment
- Assume there are 2 possible bases for a location: C and T
  - A-allele: Possibility #1 (usually the reference base)
  - B-allele: Possibility #2 (alternative allele)
  - LogR ratio: normalized signal intensity

Example: Deletion



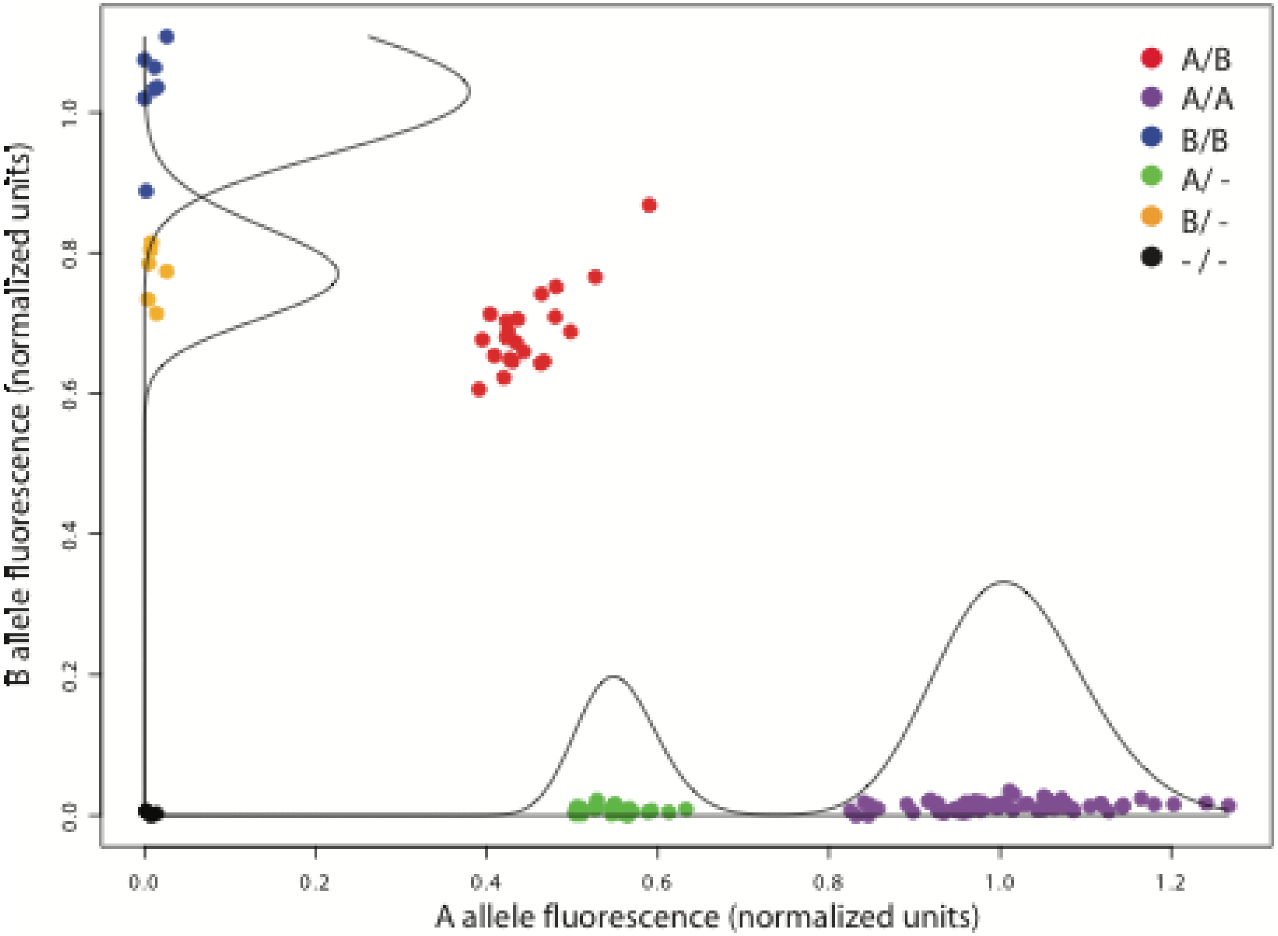
Cooper et al., Nat Genet, 2008

Example: Duplication



Cooper et al., Nat Genet, 2008

SNP-based Common CNV Genotyping

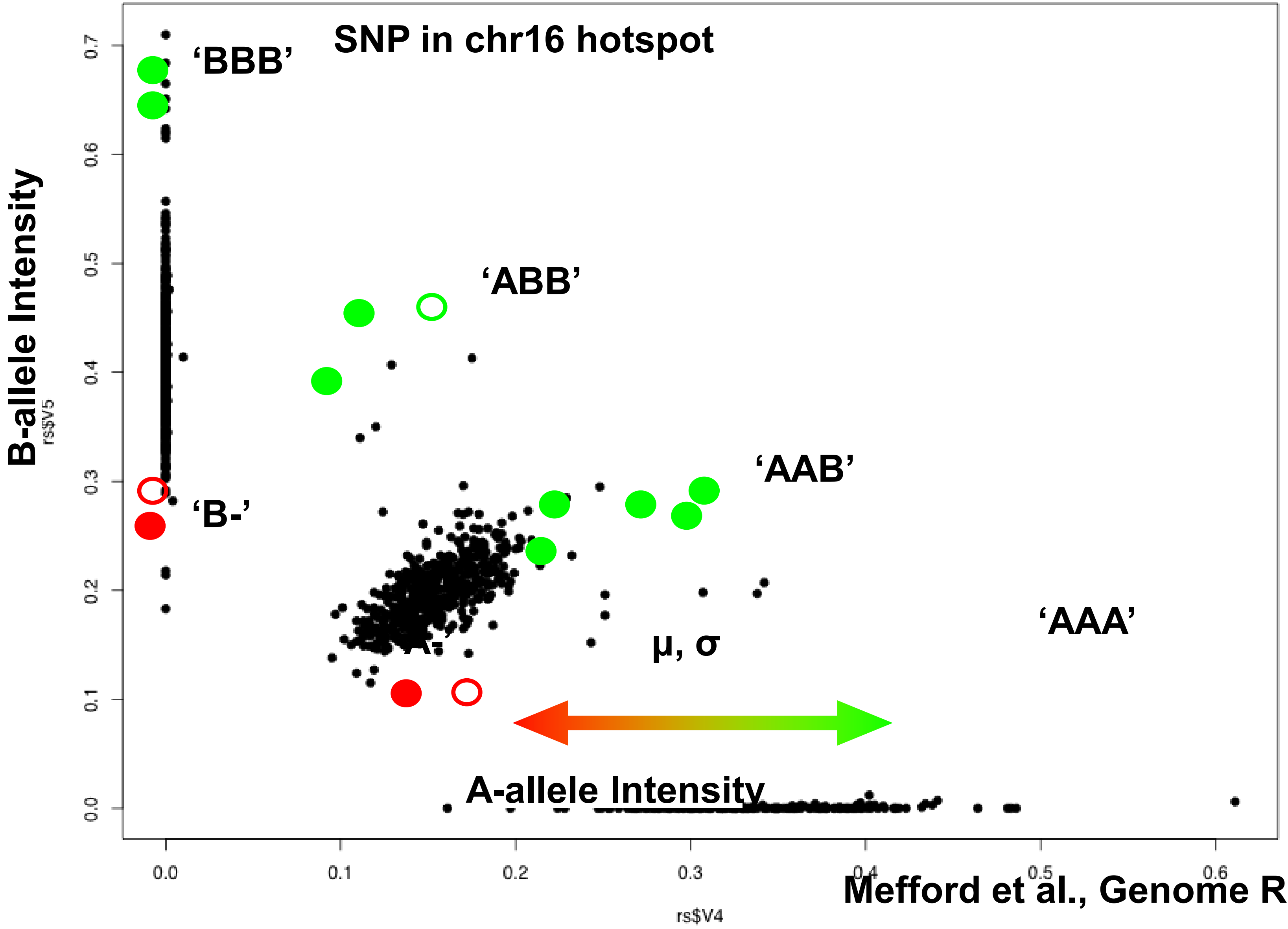


SNP-Conditional Mixture Modeling (SCIMM) for Deletion Genotyping

Uses the EM algorithm
to define copy number
(0, 1, 2) for each sample

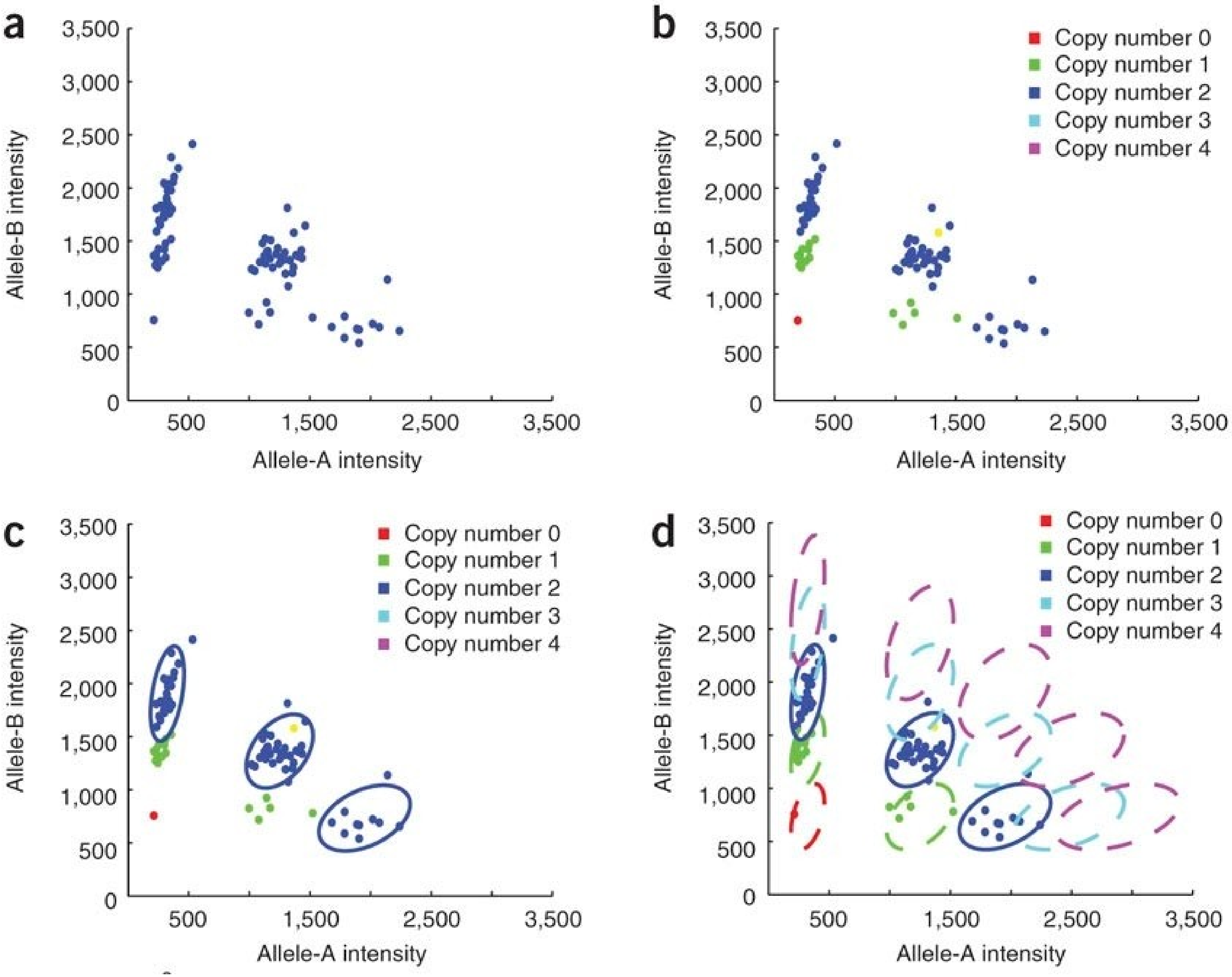
Cooper et al., Nat Genet, 2008

Snp-Conditional OUTlier (SCOUT) Detection



Mefford et al., Genome Res, 2009

Birdsuite



Korn et al., Nature Genet, 2008

SV detection with SNP arrays

- HMM based approaches that make use of:
  - the allele frequency of SNPs,
  - the distance between neighboring SNPs,
  - the signal intensities
  - detection: PennCNV (Wang *et al.* 2007), CBS (Olshen *et al.* 2004), CNVFinder (Fiegler *et al.* 2006) , cnvPartition (Illumina), QuantiSNP (Colella *et al.* 2007), SCOUT (Mefford *et al*. 2009)
- Genotyping in large cohorts: SCIMM (Cooper *et al*. 2008), BirdsEye (Korn *et al*. 2008), ÇOKGEN (Yavaş *et al.* 2010)
- Limited to deletions and insertions (Copy number variants – CNVs)

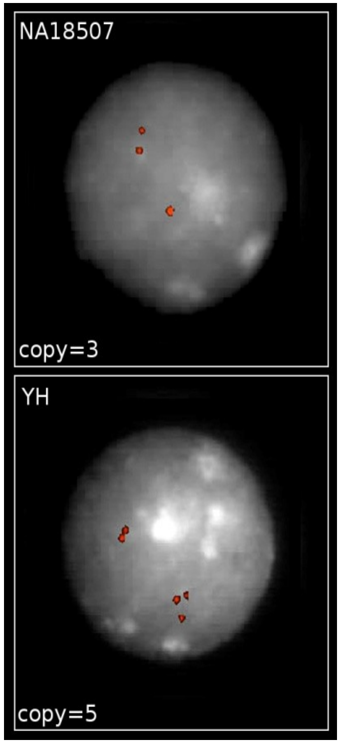
Microarrays (summary)

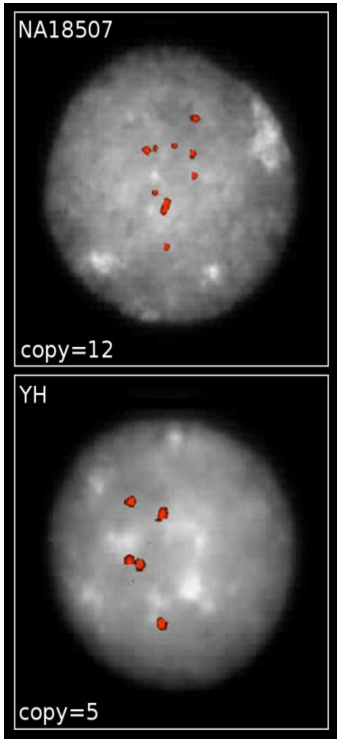
- Advantages:
  - Cheap
  - Fast
  - Good for genotyping thousands of individuals
- Disadvantages:
  - Resolution (finding exact breakpoints)
  - Targeted – i.e. no probes -> no detection
  - Relies on reference genome
  - No balanced events (inversion, translocation)
  - No transposon insertions
  - No novel sequence insertions
  - No high-copy segmental duplications -> Signal saturation

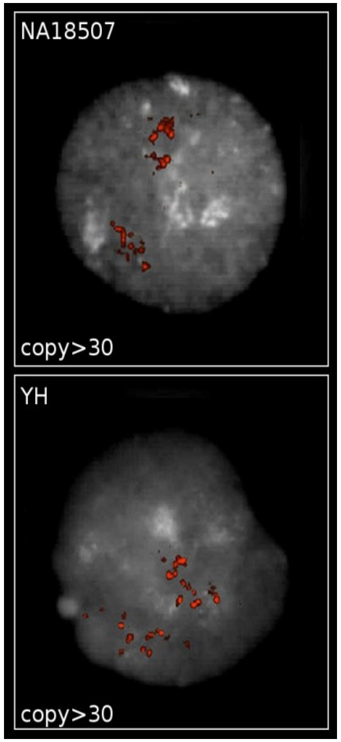
Other methods

- PCR: polymerase chain reaction
  - Run on a gel, sort by length, compare with known length (indels)
  - Follow up with sequencing (SNPs)
- qRT-PCR: quantitative real time PCR
  - Count molecules (CNV)
- FISH: Fluorescent *in situ* hybridization (large events)

CNP with FISH

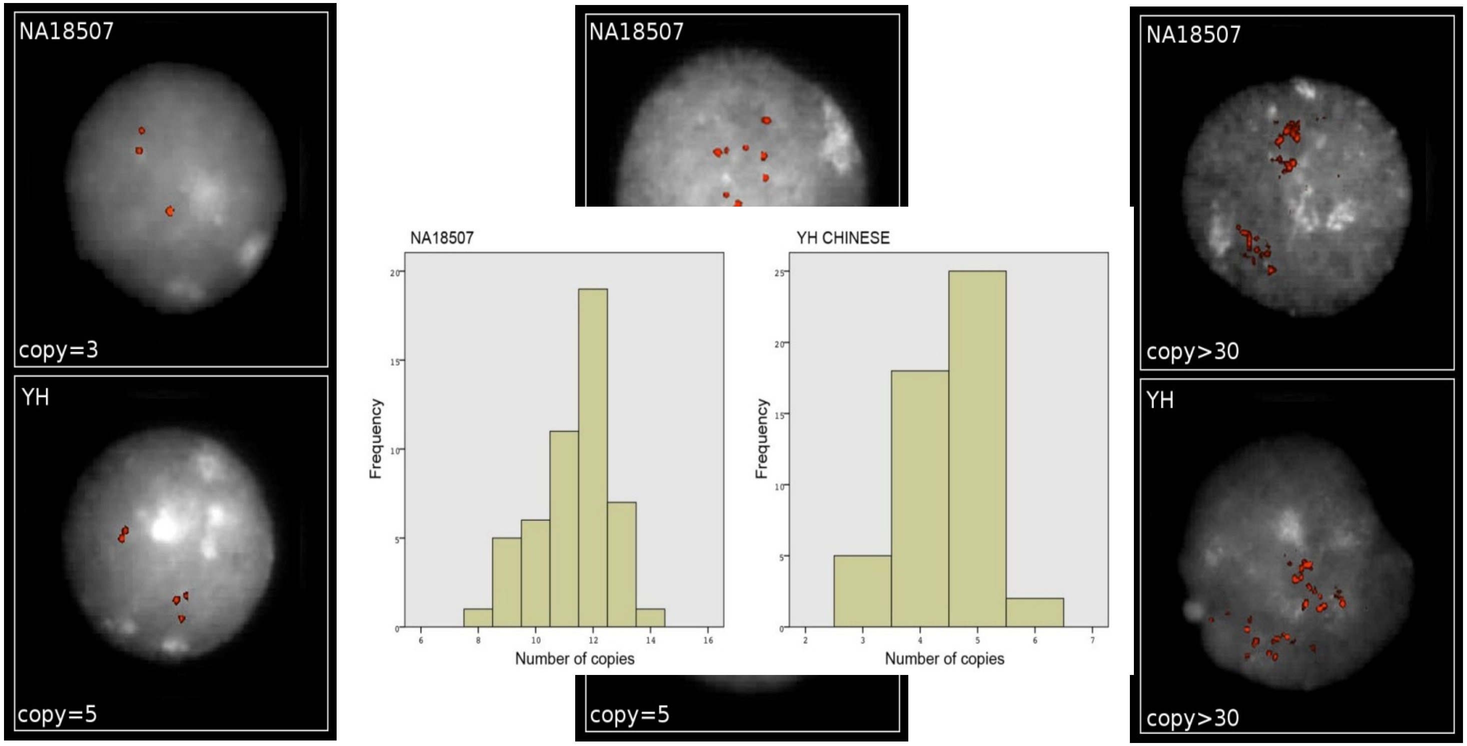






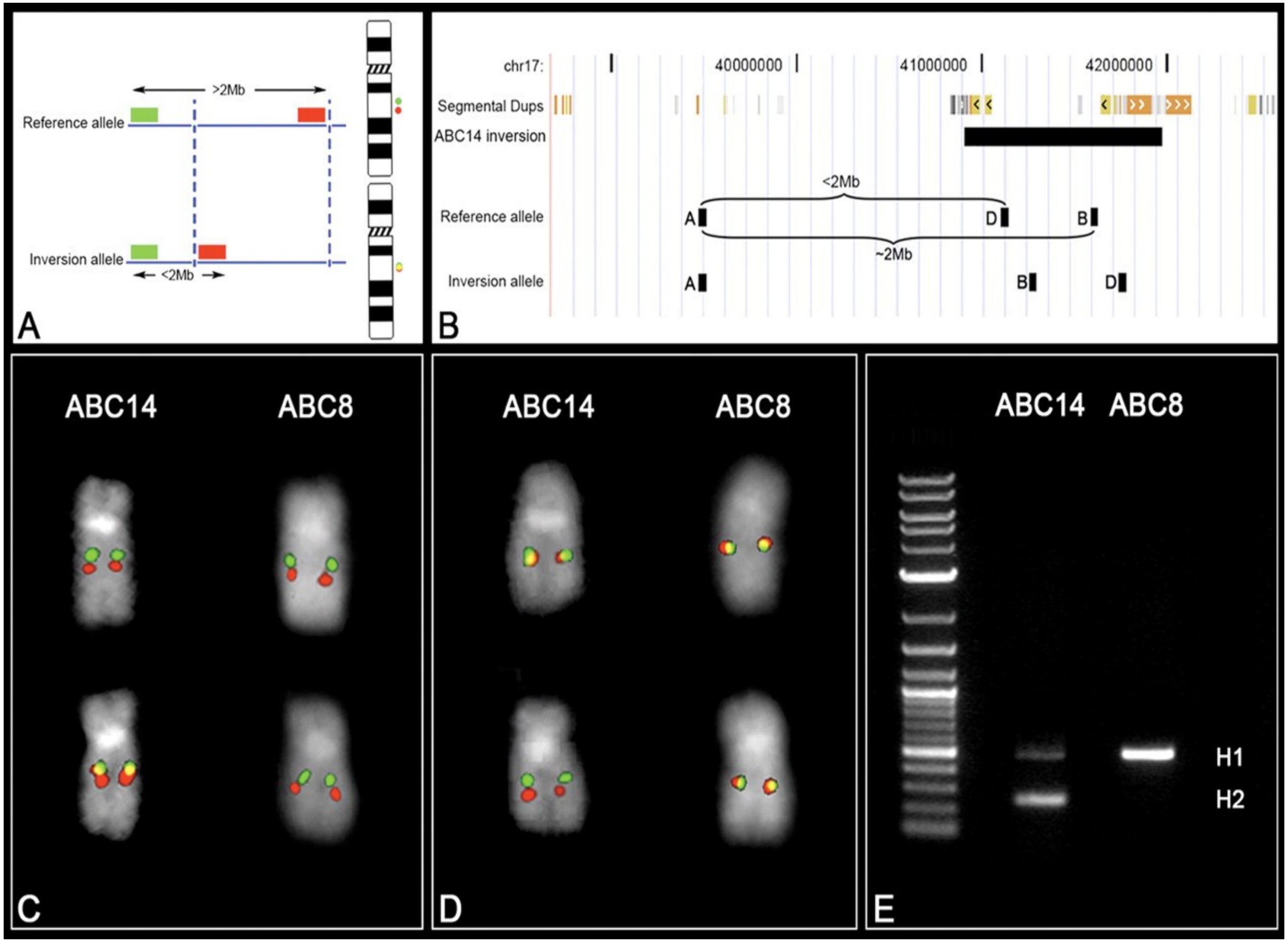
- Accurate in low-copy number
- Unreliable for >10 copies
- Noisy in high-copy number

CNP with FISH



- Accurate in low-copy number
- Unreliable for >10 copies
- Noisy in high-copy number

Inversions with FISH



Next week forward:

HIGH THROUGHPUT SEQUENCING