# Low-Level
# Copy Number Analysis

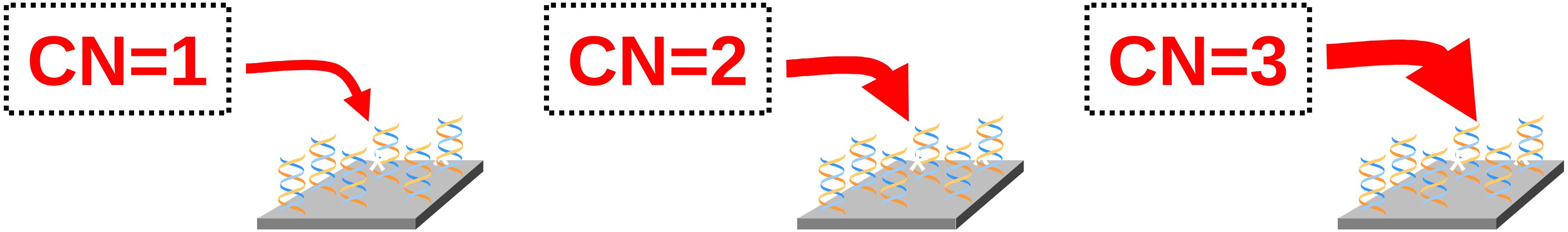
## CRMA v2 preprocessing

**Henrik Bengtsson**
Post doc, Department of Statistics,
University of California, Berkeley, USA

CEIT Workshop on SNP arrays,
Dec 15-17, 2008, San Sebastian

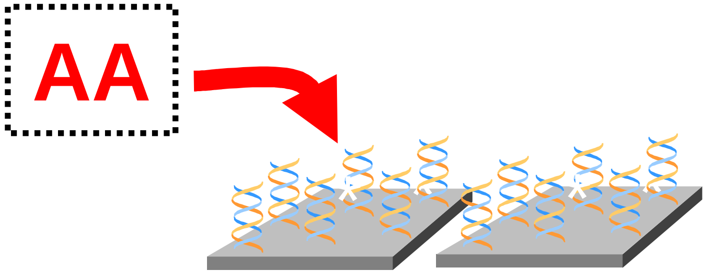
# Copy-number probes are used to quantify the amount of DNA at known loci

**CN locus:** ***...CGTAGCCATCGGTAAGTACTCAATGATAG...***
**PM:** **ATCGGTAGCCATTCATGAGTTACTA**



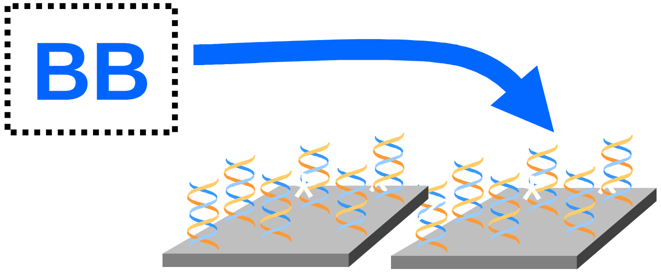
**PM = c** **PM = 2c** **PM = 3c**

# SNP probes can also be used to estimate total copy numbers



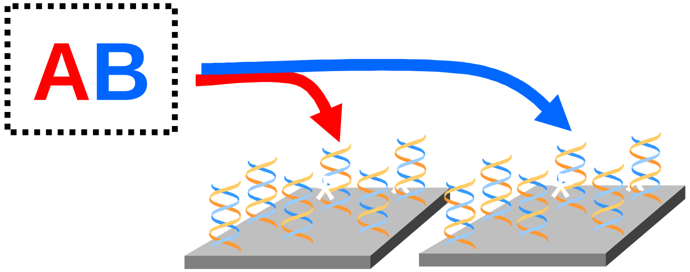
AA

$PM = PM_A + PM_B = 2c$



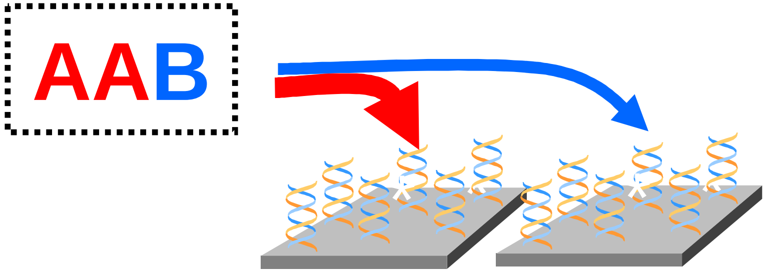
BB

$PM = PM_A + PM_B = 2c$



AB

$PM = PM_A + PM_B = 2c$



AAB

$PM = PM_A + PM_B = 3c$

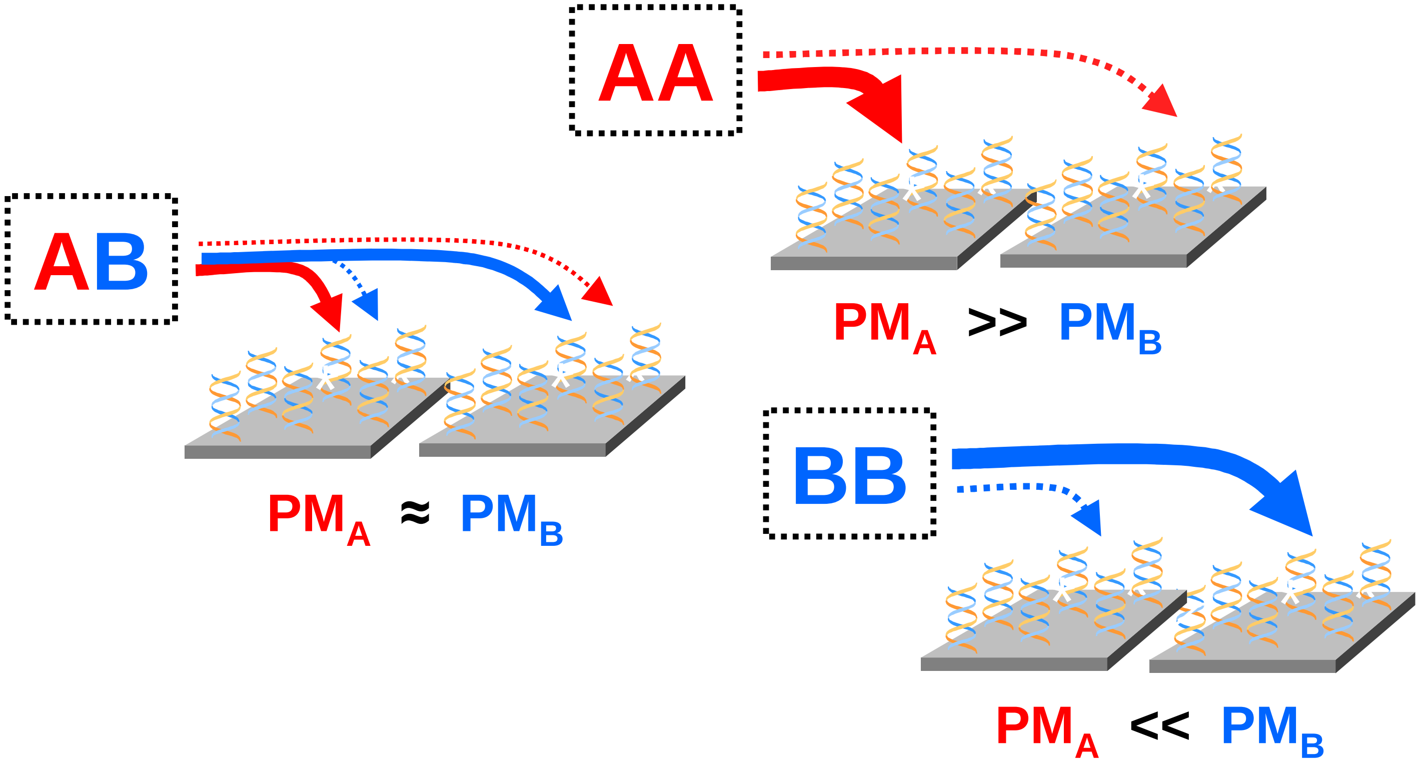
|  | **CRMA v2** |
|---|---|
| ***Preprocessing***<br>*(probe signals)* | 1. Allelic crosstalk calibration<br>2. Probe-sequence normalization |
| ***Summarization*** | Robust averaging:<br>CN probes: $\theta_{ij} = PM_{ij}$<br>SNPs: $\theta_{ijA} = \text{median}_k(PM_{ijkA})$<br>$\theta_{ijB} = \text{median}_k(PM_{ijkB})$<br>array *i*, loci *j*, probe *k.* |
| ***Post-processing*** | PCR fragment-length normalization |
| ***Transform*** | $(\theta_{ijA},\theta_{ijB}) \Rightarrow (\theta_{ij}, \beta_{ij})$<br>$\theta_{ij} = \theta_{ijA}+\theta_{ijB}$, $\beta_{ij} = \theta_{ijB} / \theta_{ij}$ |
| ***Allele-specific & total CNs*** | $C_{ijA} = 2*(\theta_{ijA} / \theta_{Rj})$ and $C_{ijB} = 2*(\theta_{ijA} / \theta_{Rj})$<br>$C_{ij} = 2*(\theta_{ij} / \theta_{Rj})$ reference *R* |

# Allelic crosstalk calibration

# Crosstalk between alleles
# - *adds significant artifacts to signals*

Cross-hybridization:

*Allele A*: ***TCGGTAAGTACTC***
*Allele B*: ***TCGGTATGTACTC***



AA

$PM_A >> PM_B$

AB

$PM_A \approx PM_B$

BB

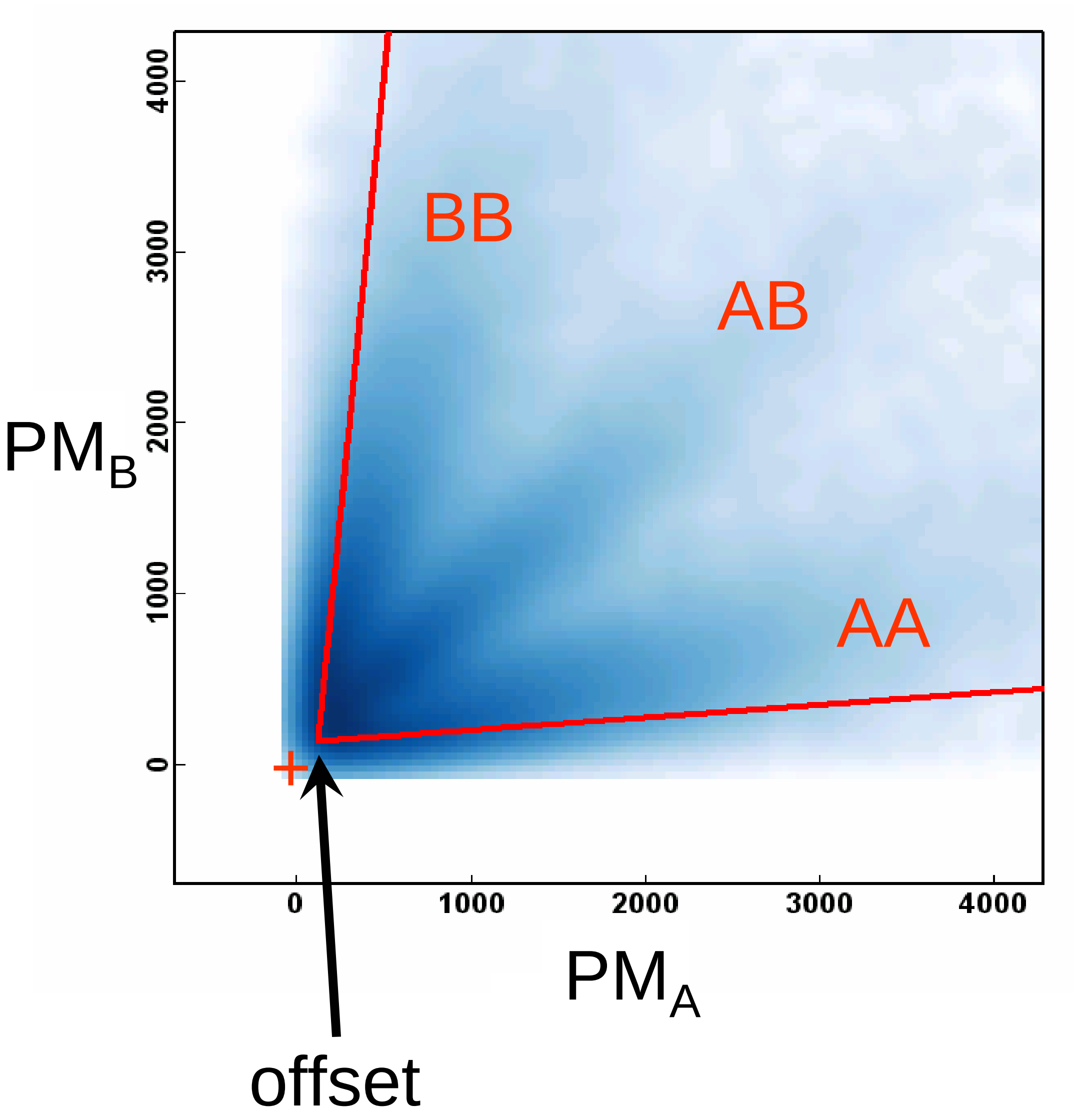
$PM_A << PM_B$

# There are six possible allele pairs

- Nucleotides: {A, C, G, T}
- Ordered pairs:
  - (A,C), (A,G), (A,T), (C,G), (C,T), (G,C)

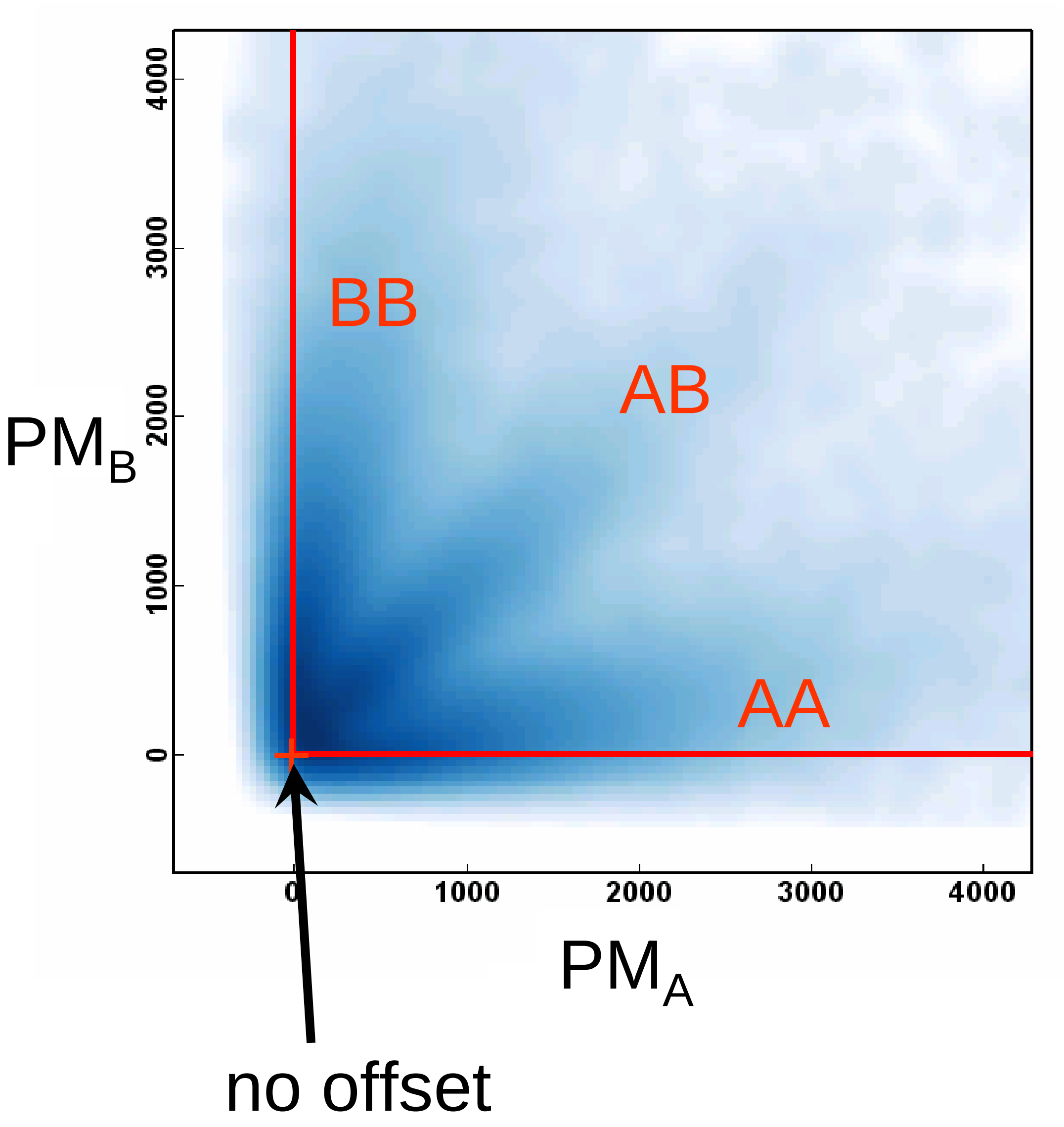
- Because of different nucleotides bind differently, the crosstalk from A to C might be very different from A to T.

# Crosstalk between alleles is easy to spot



Example:
Data from one array.
Probe pairs ($PM_A$, $PM_B$)
for nucleotide pair (A,T).

# Crosstalk between alleles can be estimated and corrected for



What is done:

1. **Offset is removed** from SNPs and CN units.

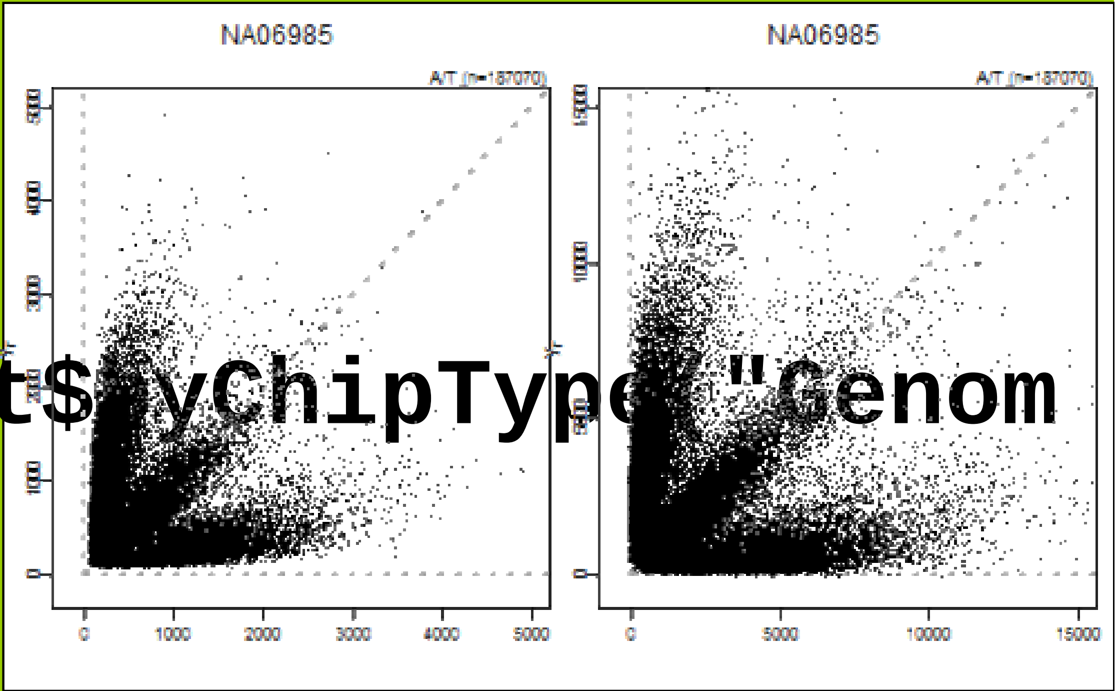
2. **Crosstalk is removed** from SNPs.

aroma.affymetrix

## You will need:

- Affymetrix CDF, e.g. GenomeWideSNP_6.cdf
- Probe sequences*, e.g. GenomeWideSNP_6.acs

## Calibrate CEL files:

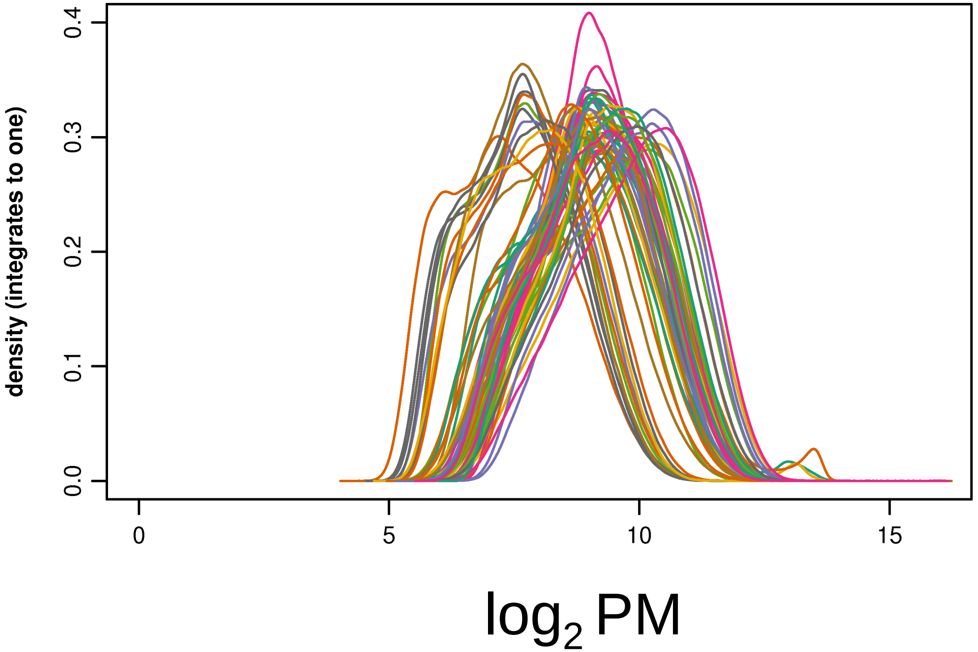


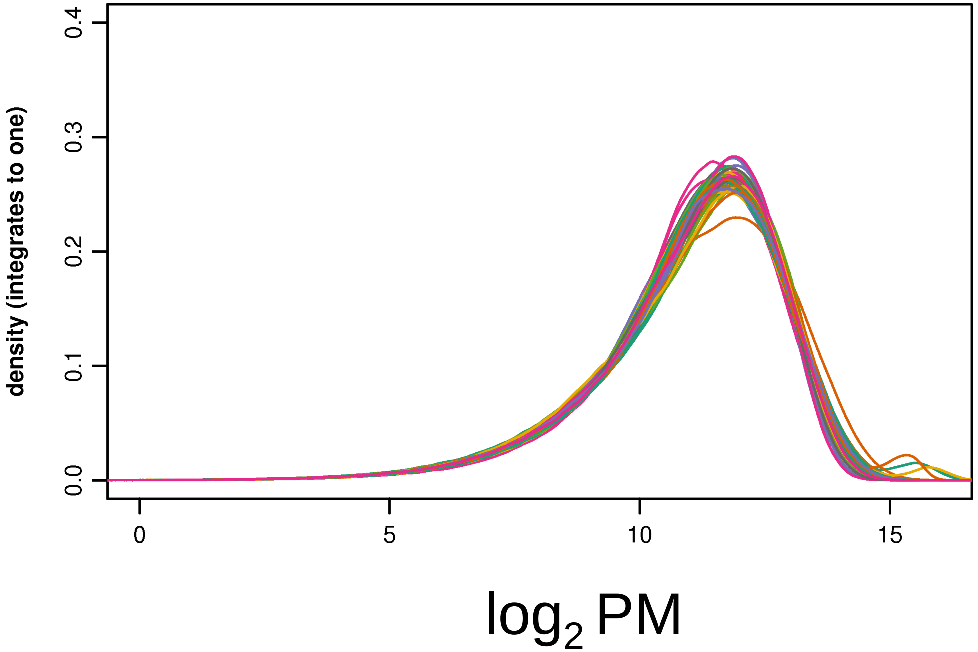
```
cdf <-
  AffymetrixCdfSet$byChipType("GenomeWideSNP_6")
csR <-
```

# Crosstalk calibration corrects for differences in distributions too

Before removing crosstalk the arrays differ significantly...

...when removing offset & crosstalk differences goes away.



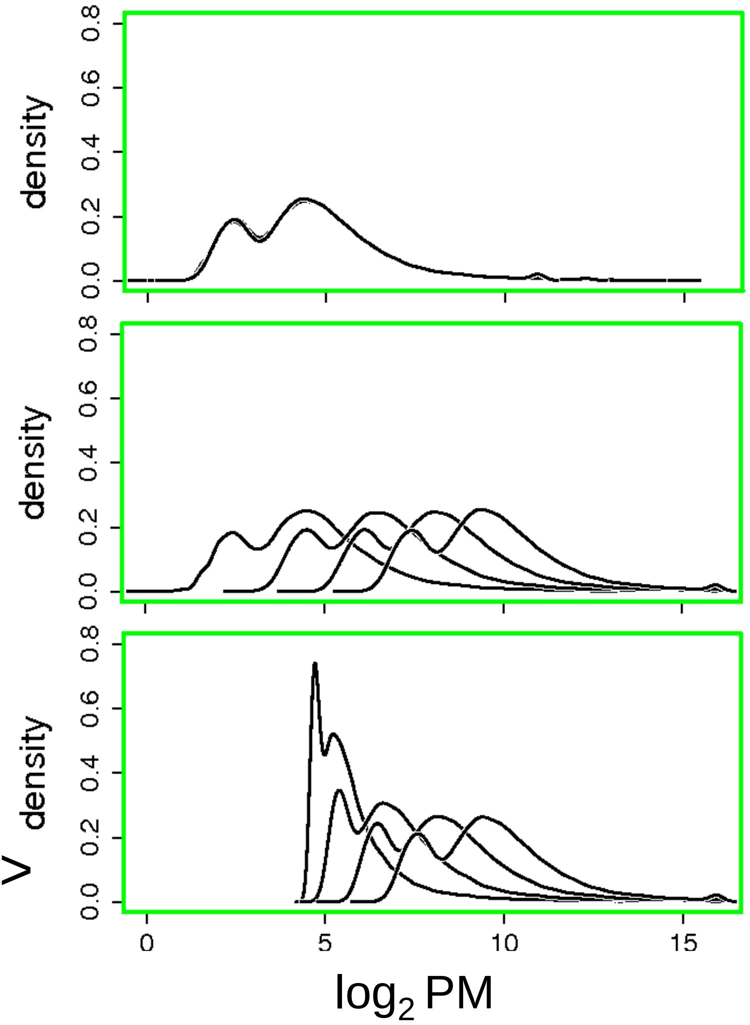


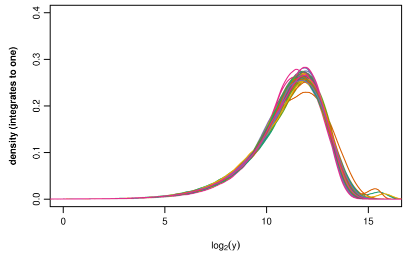
# How can a translation and a rescaling make such a big difference?

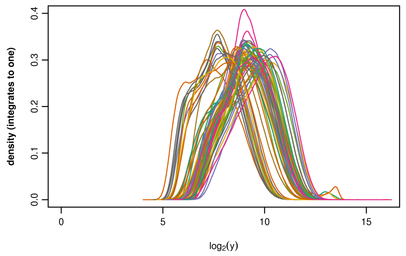
4 measurements
of the **same thing**:

With **different scales**:
*log(b*PM) = log(b)+log(PM)*

With **different scales**
and **some offset**:
*log(a+b*PM) = <non-linear>*







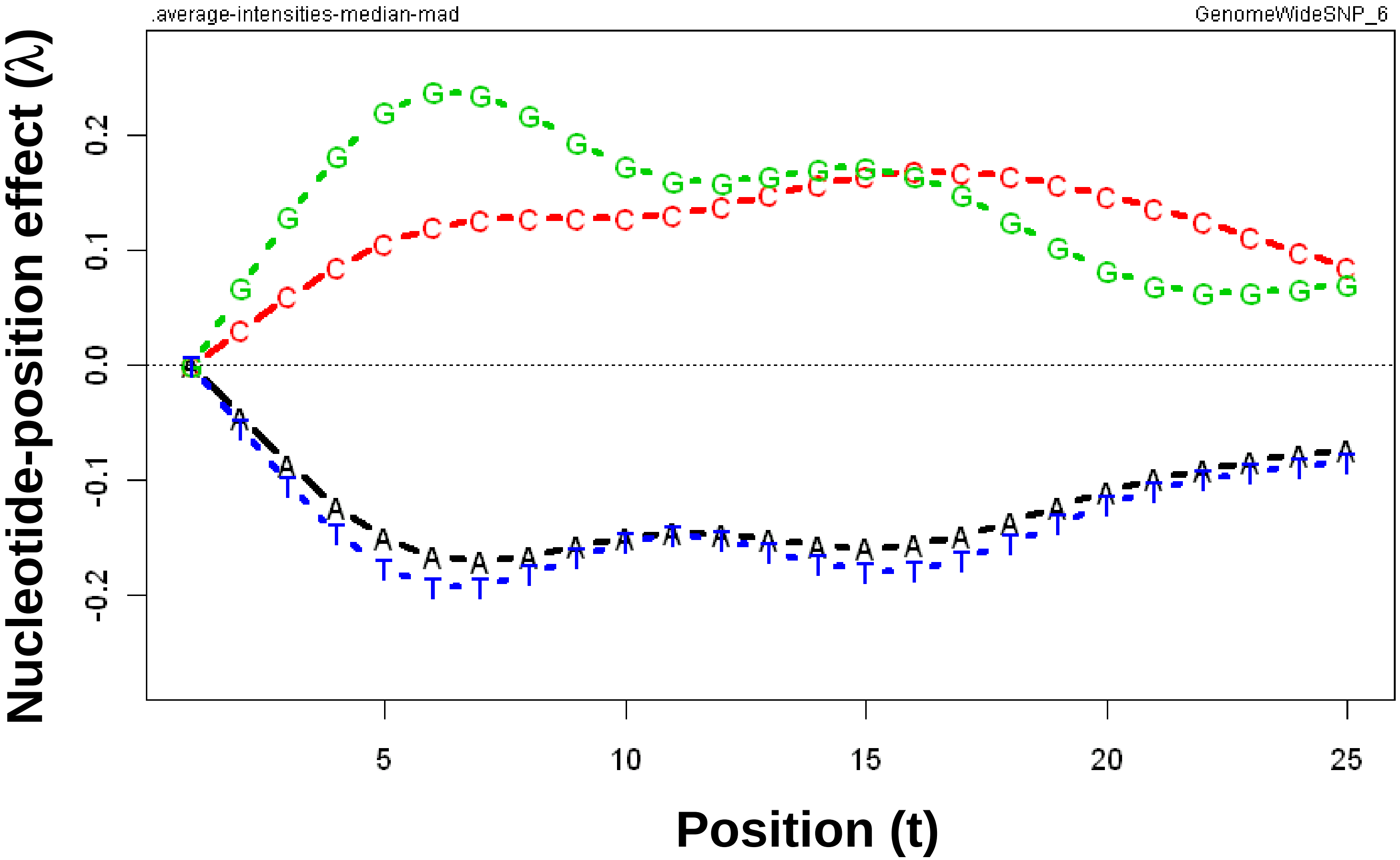
# Take home message

Allelic crosstalk calibration controls for:

1) offset in signals
2) crosstalk between allele A and allele B.

# Probe sequence normalization

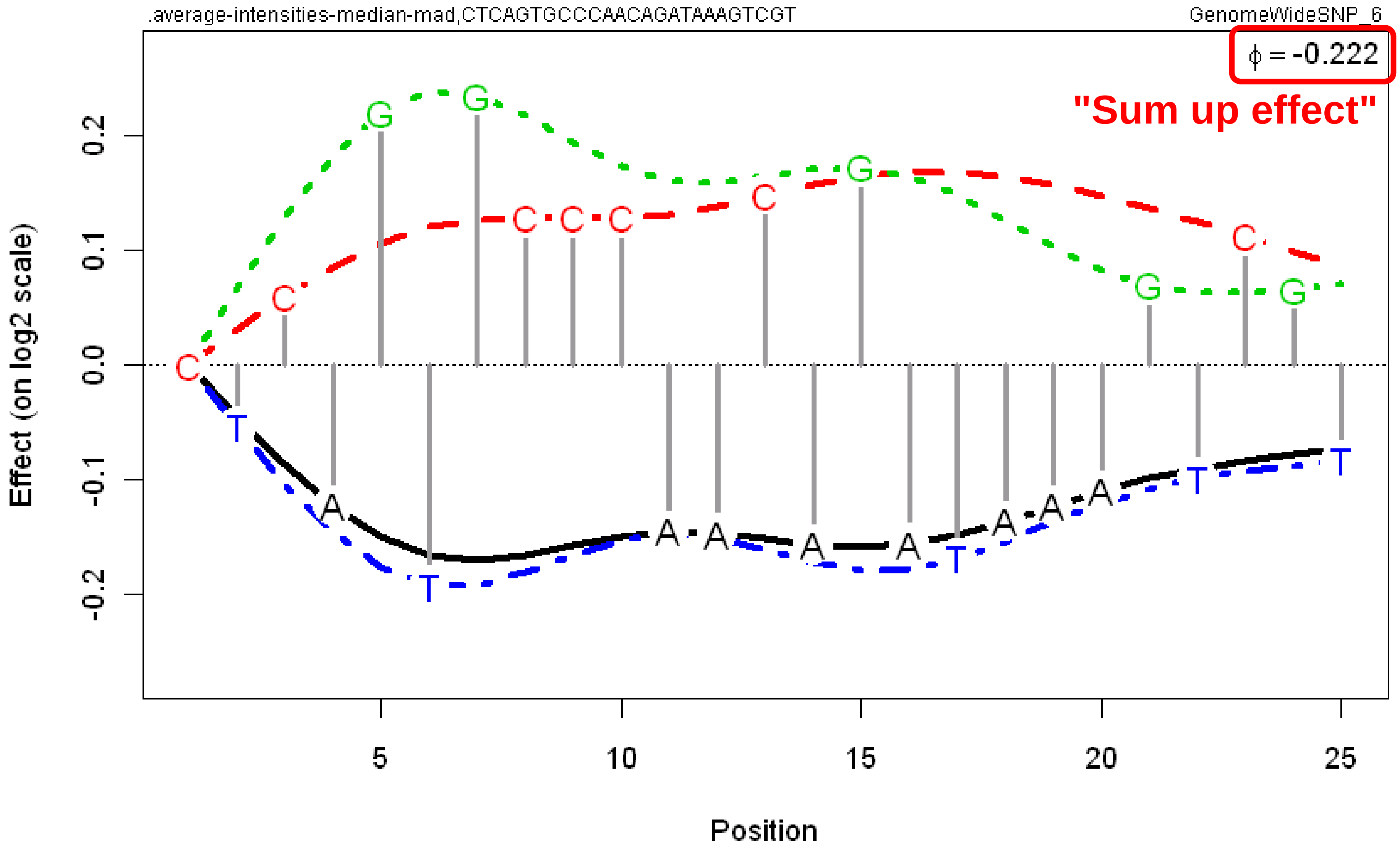
# Nucleotide-Position Model



Probe-position ($\log_2$) affinity for probe k:

$$\phi_k = \phi((b_{k,1},b_{k,2},\ldots,b_{k,25})) = \sum_{t=1..25} \sum_{b=\{ACGT\}} I(b_{k,t}=b)\lambda_{b,t}$$

# Example: Probe-position affinity for **CTCAGTGCCCAACAGATAAAGTCGT**



"Sum up effect"

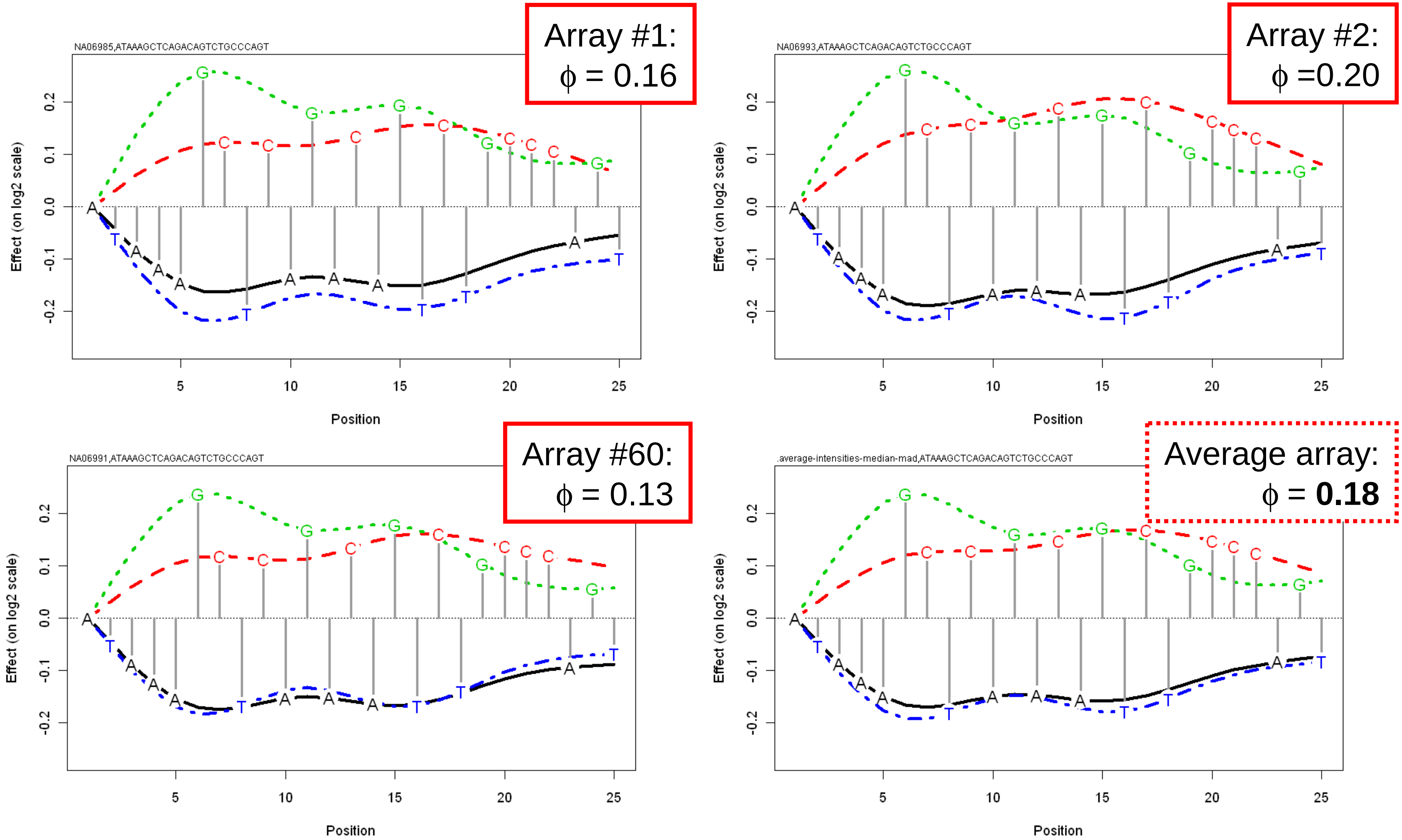
# Probe-sequence normalization helps

1. The effects differ slightly across arrays:
   - adds extra across-array variances
   - *will be removed*

2. The effects differ between $PM_A$ and $PM_B$:
   - introduces genotypic imbalances such that $PM_A$+$PM_B$ will differ for AA, AB & BB.
   - *will be removed*

# 1. BPN controls for across array variability

# The nucleotide-position effect differ between arrays



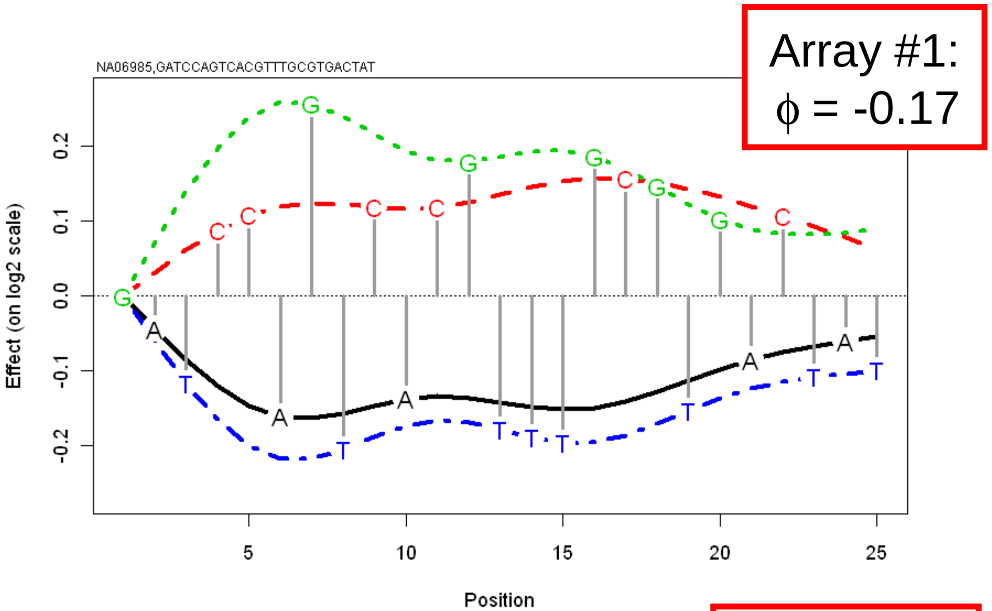
Array #1: ϕ = 0.16

Array #2: ϕ =0.20

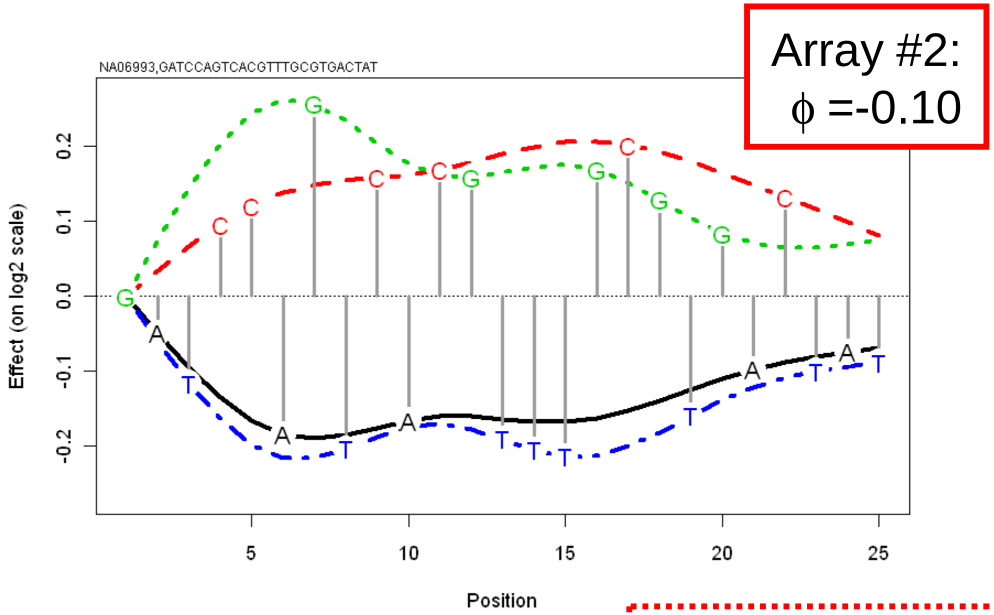
Array #60: ϕ = 0.13

Average array: ϕ = **0.18**

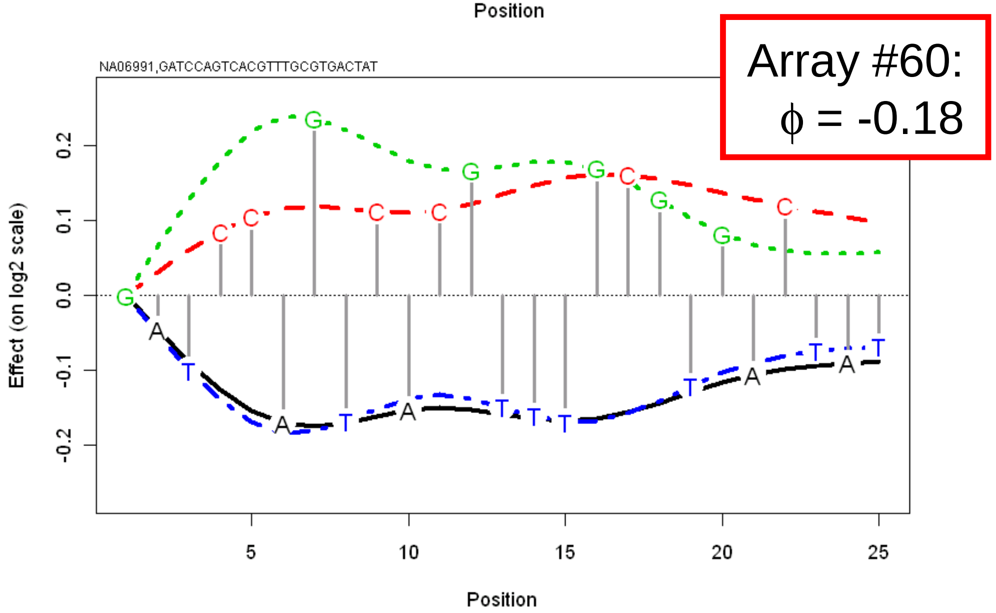
# The impact of these effects varies with probe sequence



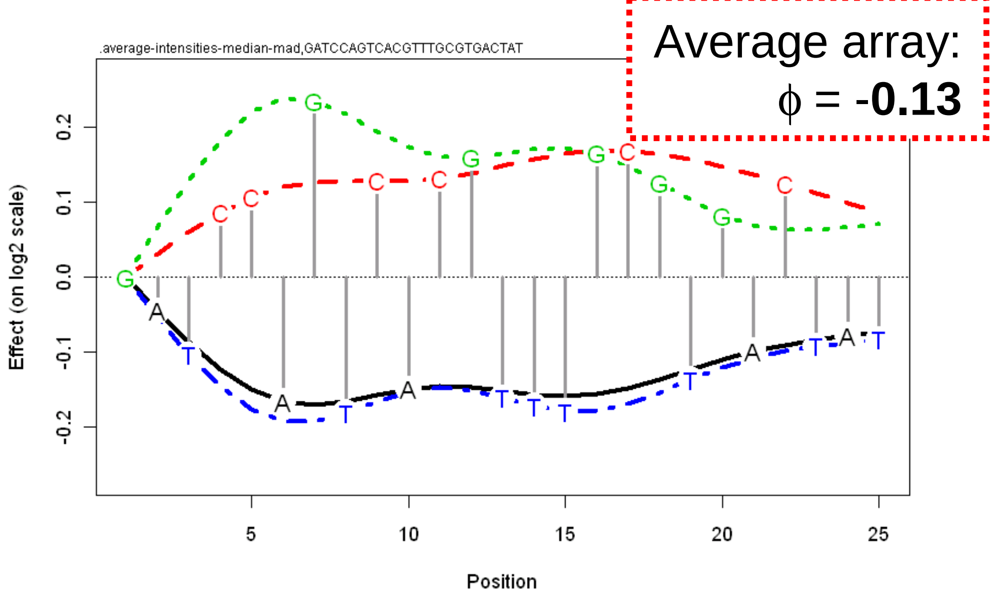
Array #1:
$\phi$ = -0.17



Array #2:
$\phi$ =-0.10

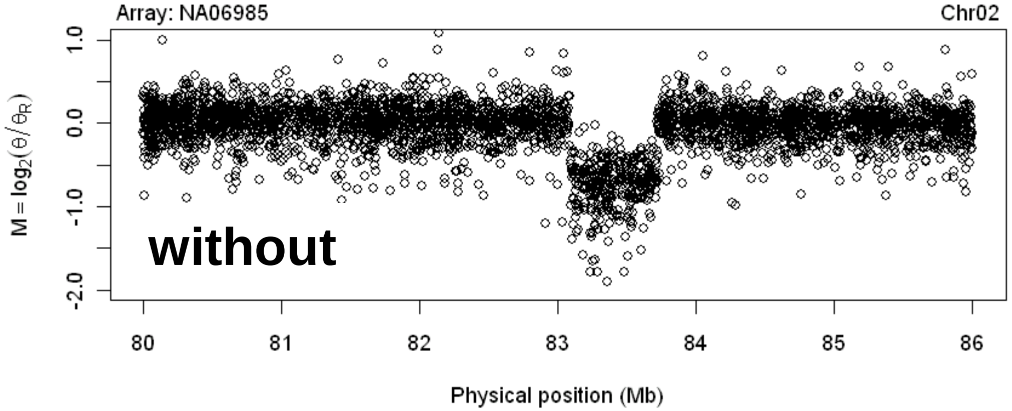


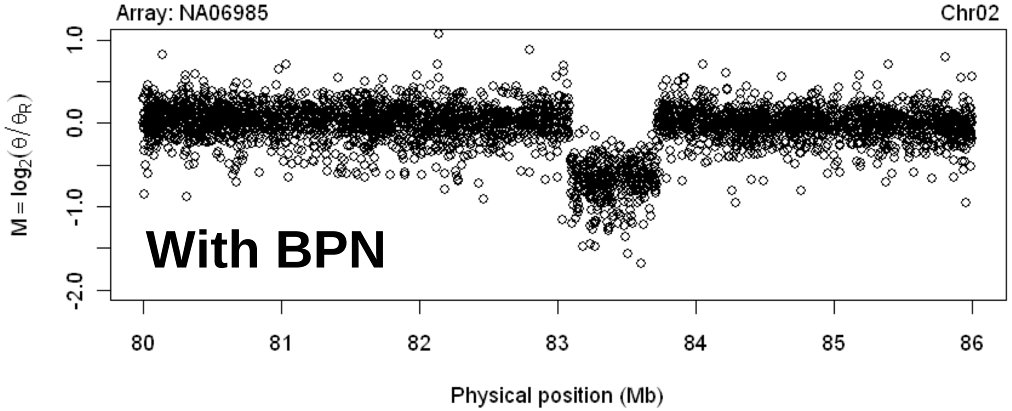
Array #60:
$\phi$ = -0.18

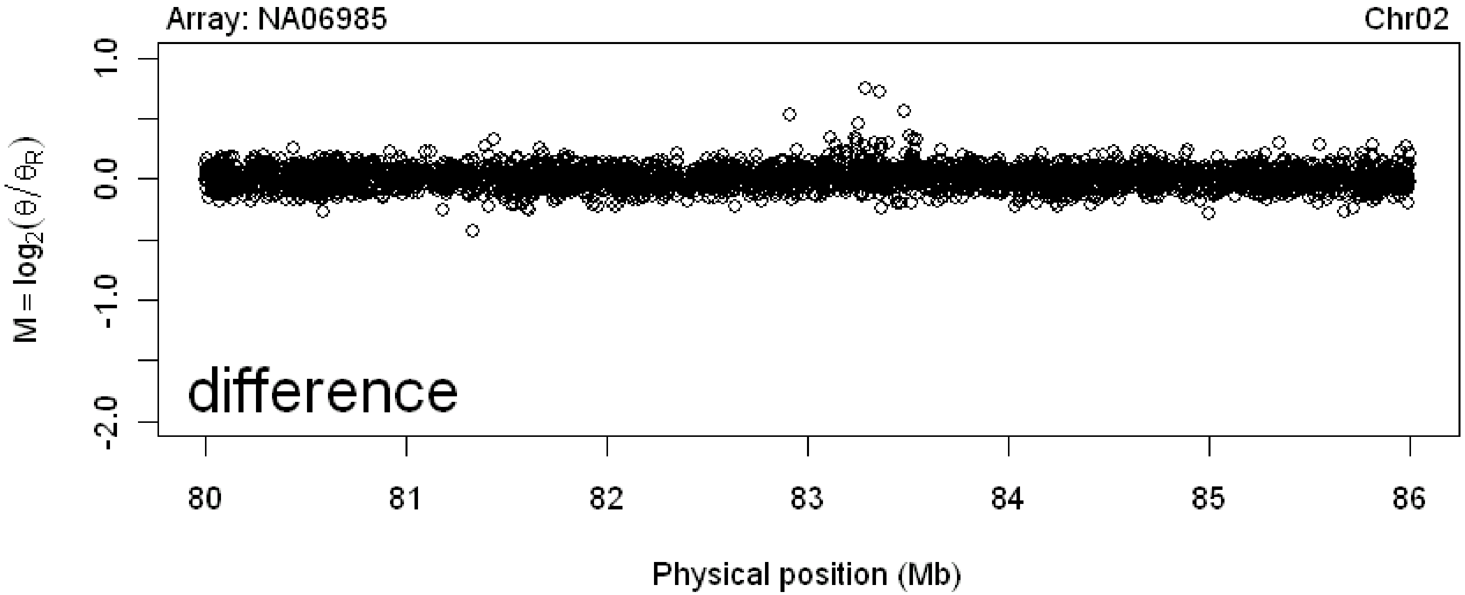


Average array:
$\phi$ = **-0.13**

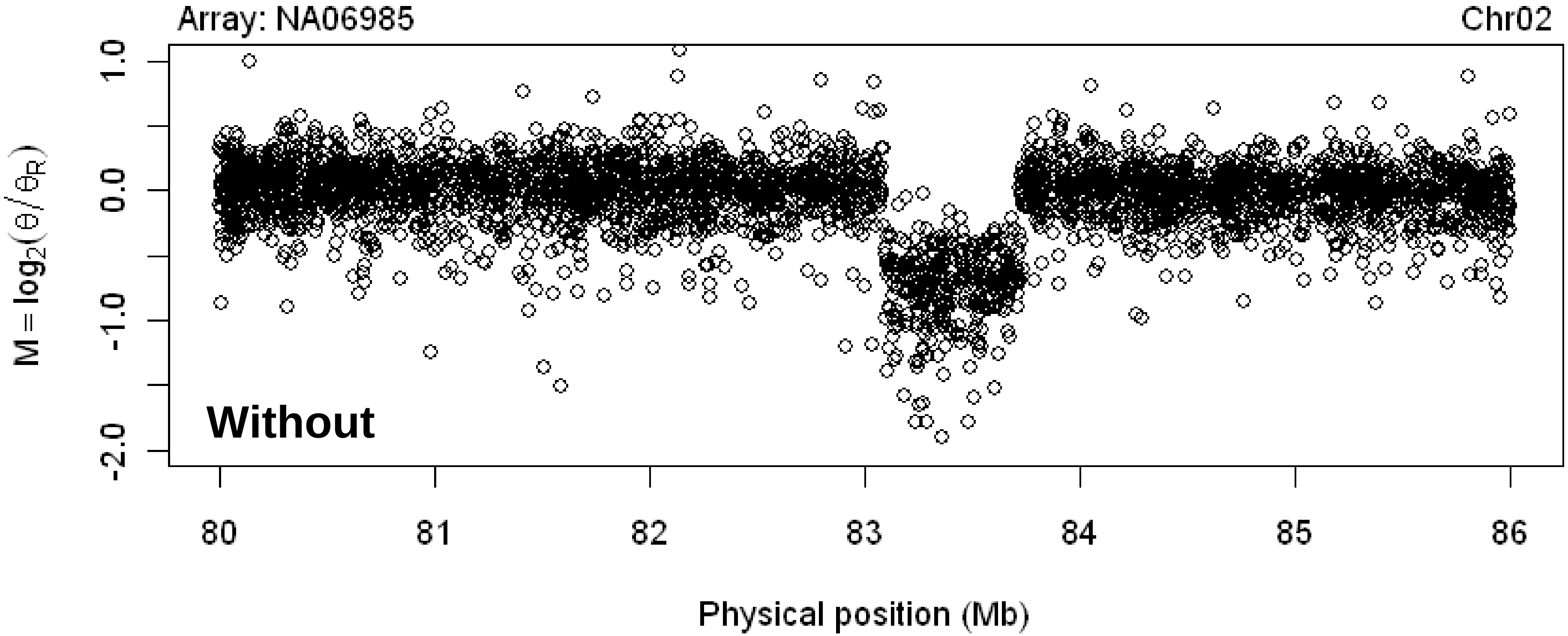
# There is a noticeable difference in raw CNs before and after normalization



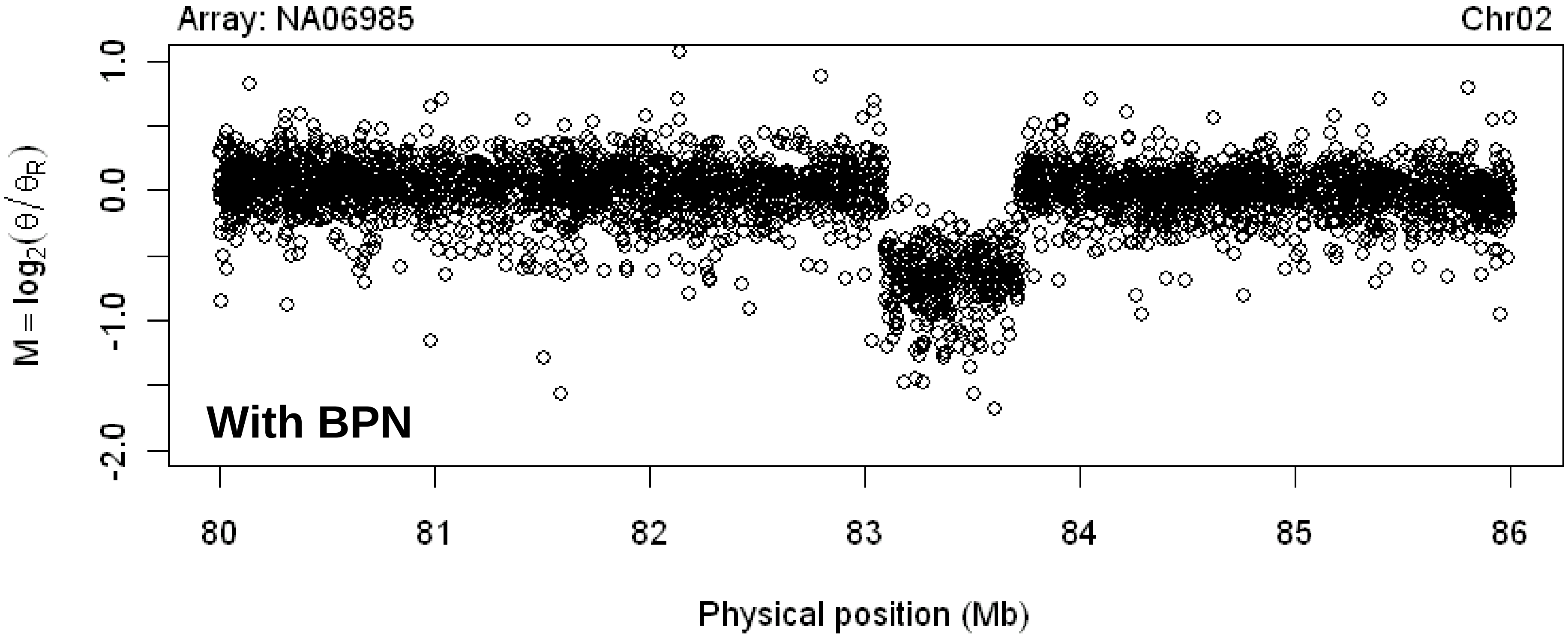




# There is a noticeable difference in raw CNs before and after normalization

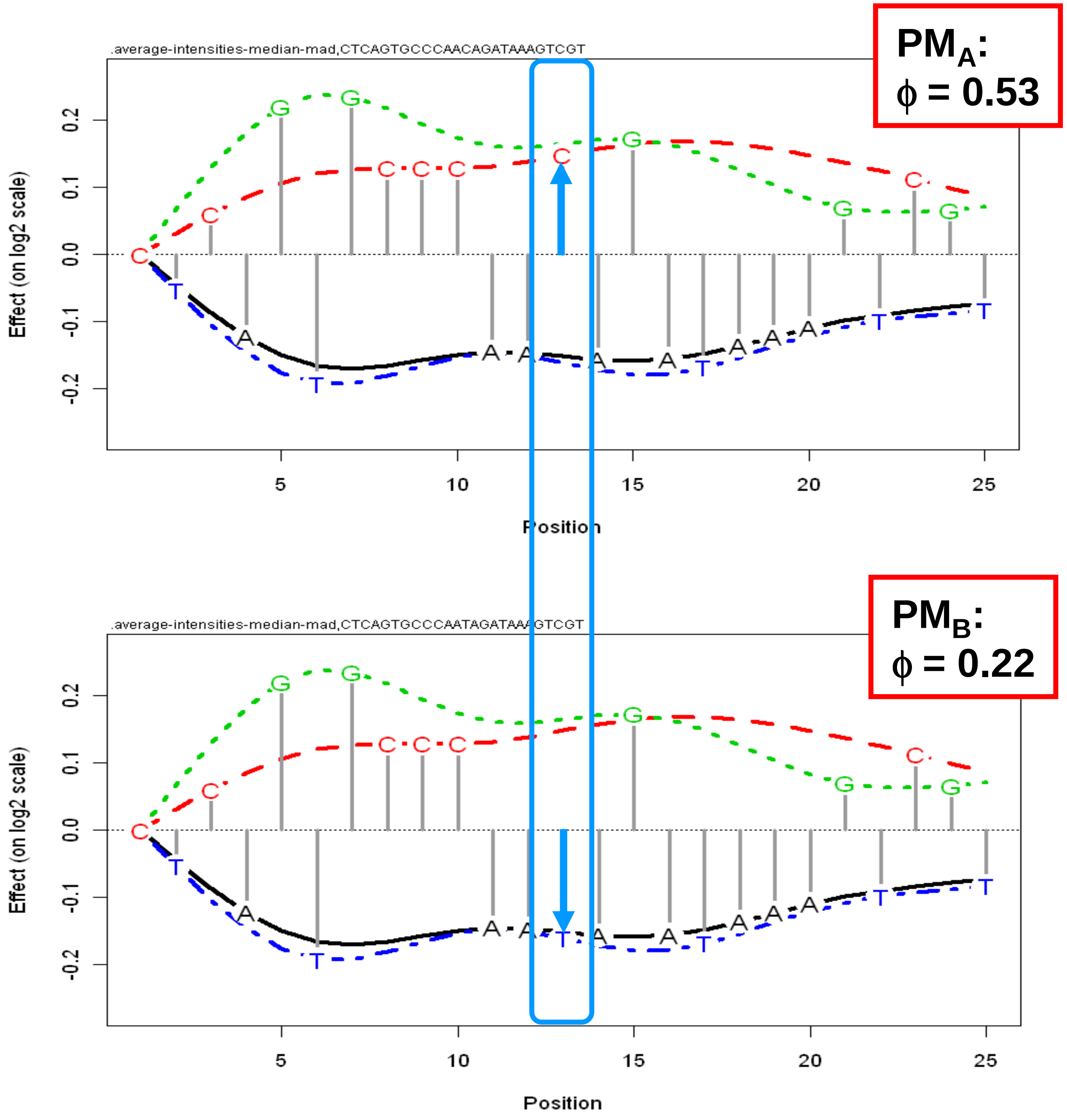


# There is a noticeable difference in raw CNs before and after normalization



# 2. BPN controls for allele A and allele B imbalances

# Nucleotide-position normalization controls for imbalances between allele A & allele B



**$PM_A$:**
**$\phi$ = 0.53**

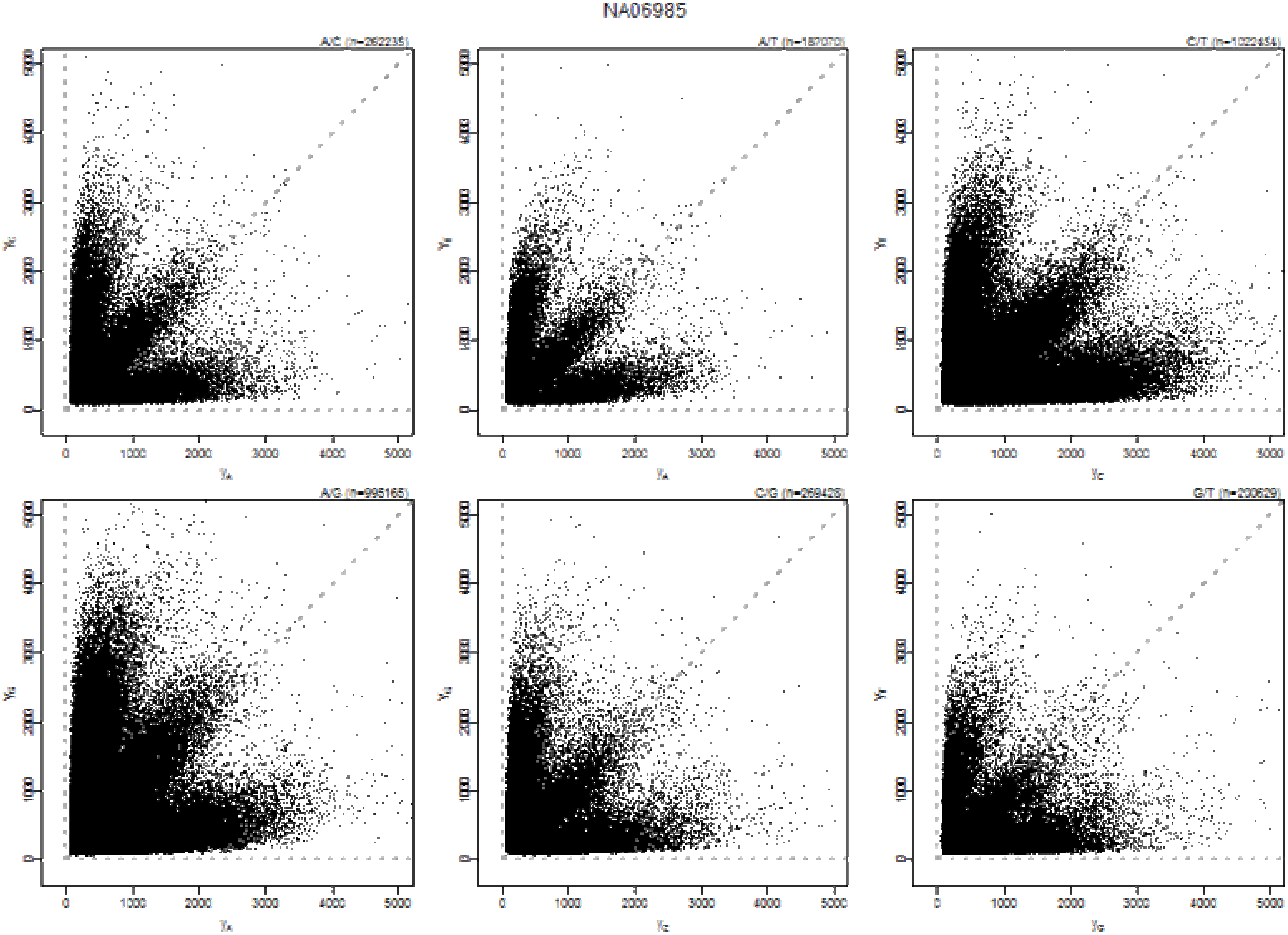
**$PM_B$:**
**$\phi$ = 0.22**

Genotypic imbalances:

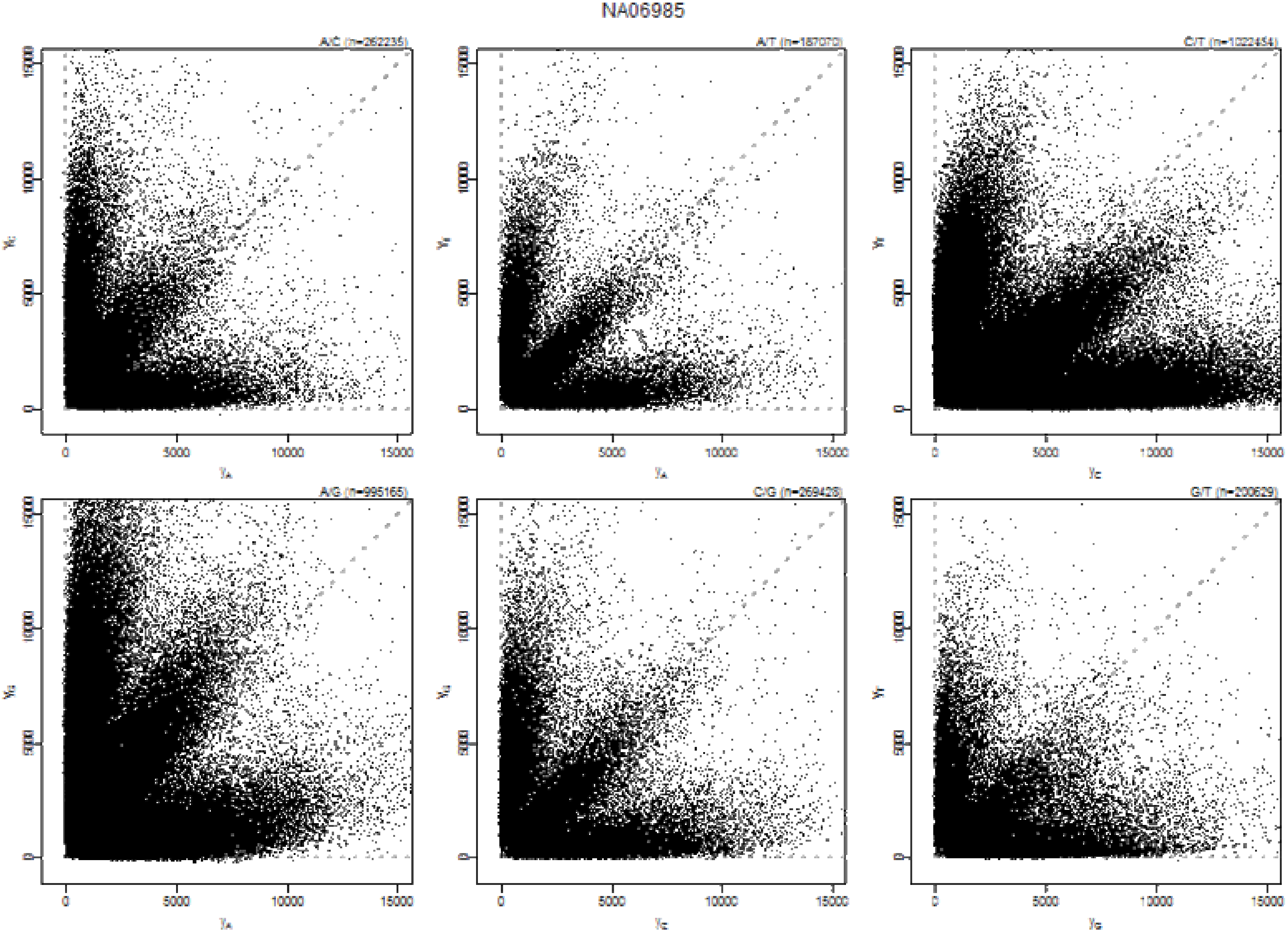
$PM=PM_A+PM_B$:
AA: 0.53+0.53 = 1.06
AB: 0.53+0.22 = 0.75
BB: 0.22+0.22 = 0.44

**Thus, AA signals are 2^(1.06-0.44) = 2^0.62 = 1.54 times stronger than BB signals.**

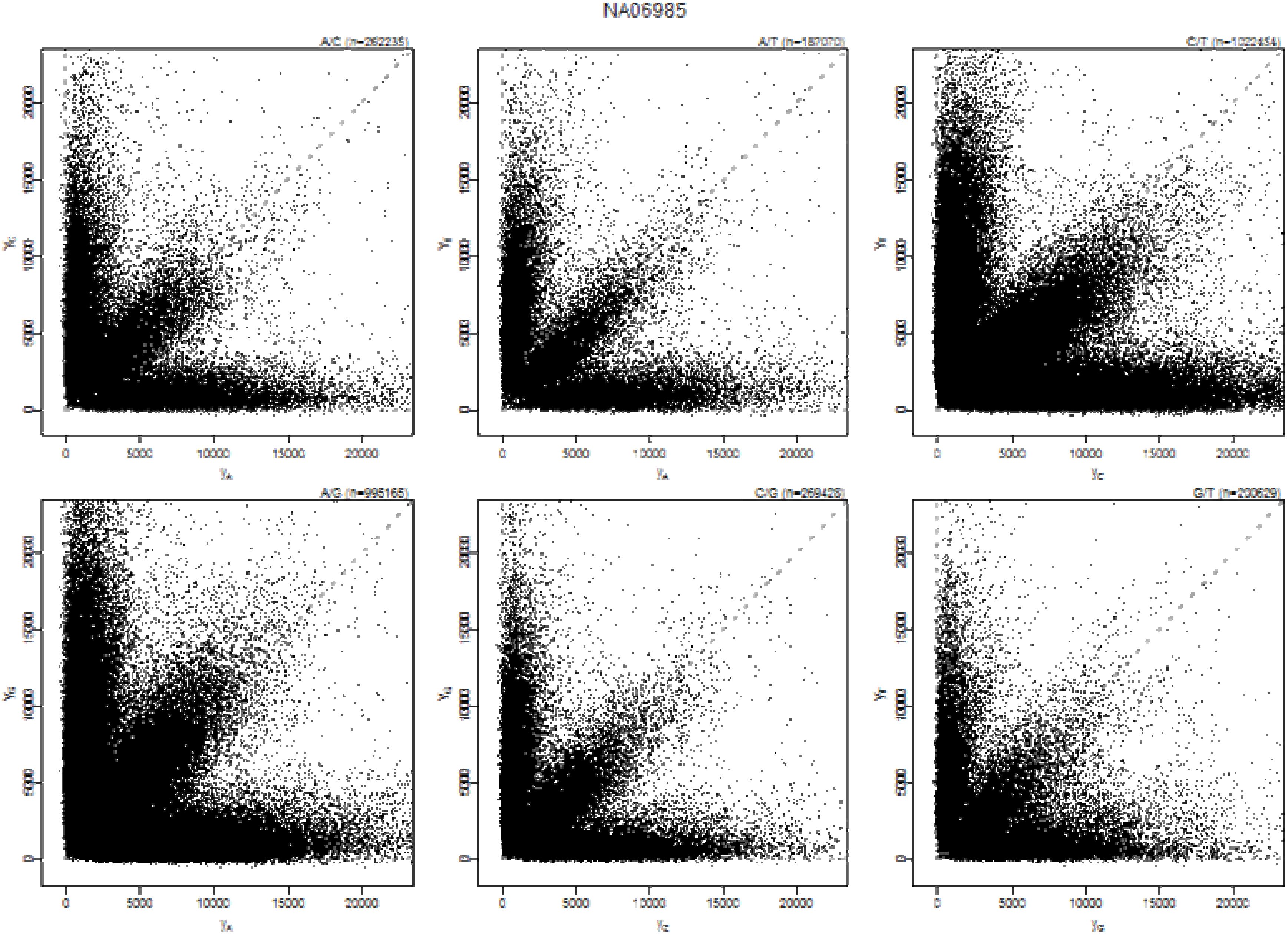
# (i) Before calibration there is crosstalk
## *- pairs AC, AG, AT, CG, CT & GT*



# (ii) After calibration the homozygote arms are more orthogonal (note heterozygote arm!)



# (iii) After sequence normalization the heterozygote arms are more balanced



# aroma.affymetrix

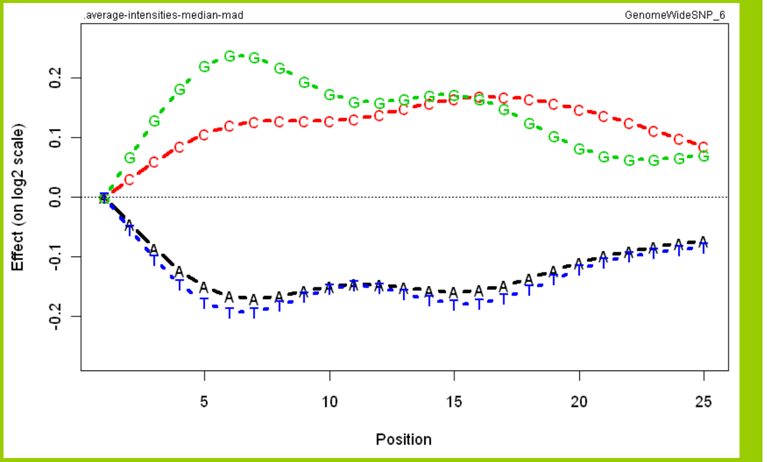
You will need:

- Affymetrix CDF, e.g. GenomeWideSNP_6.cdf
- Probe sequences*, e.g. GenomeWideSNP_6.acs

Normalize CEL files:

```
bpn <- BasePositionNormalization(csC, target="zero")
csN <- process(bpn)
```

Works with any chip type, e.g. resequencing, exon, expression, SNP.



To plot:

```
fit <- getFit(bpn, array=1)
plot(fit)
```

# Probe summarization

# Probe summarization
(on the new arrays)

- CN units: All single-probe units:
  - Chip-effect estimate: $\theta_{ij} = PM_{ij}$

- SNPs: Identically replicated probe pairs:
  - Probe pairs: $(PM_{ijkA}, PM_{ijkB})$; $k=1,2,3$
  - Allele-specific estimates:
    - $\theta_{ijA} = \text{median}_k\{PM_{ijkA}\}$
    - $\theta_{ijB} = \text{median}_k\{PM_{ijkB}\}$

# aroma.affymetrix

You will need:

- Affymetrix CDF, e.g. GenomeWideSNP_6.cdf

Summarizing probe signals:

```
plm <- AvgCnPlm(csN, combineAlleles=FALSE)
fit(plm)

ces <- getChipEffectSet(plm)
theta <- extractTheta(ces)
```

# Probe-level summarization (10K-500K)

***- (if) replicated probes respond differently***

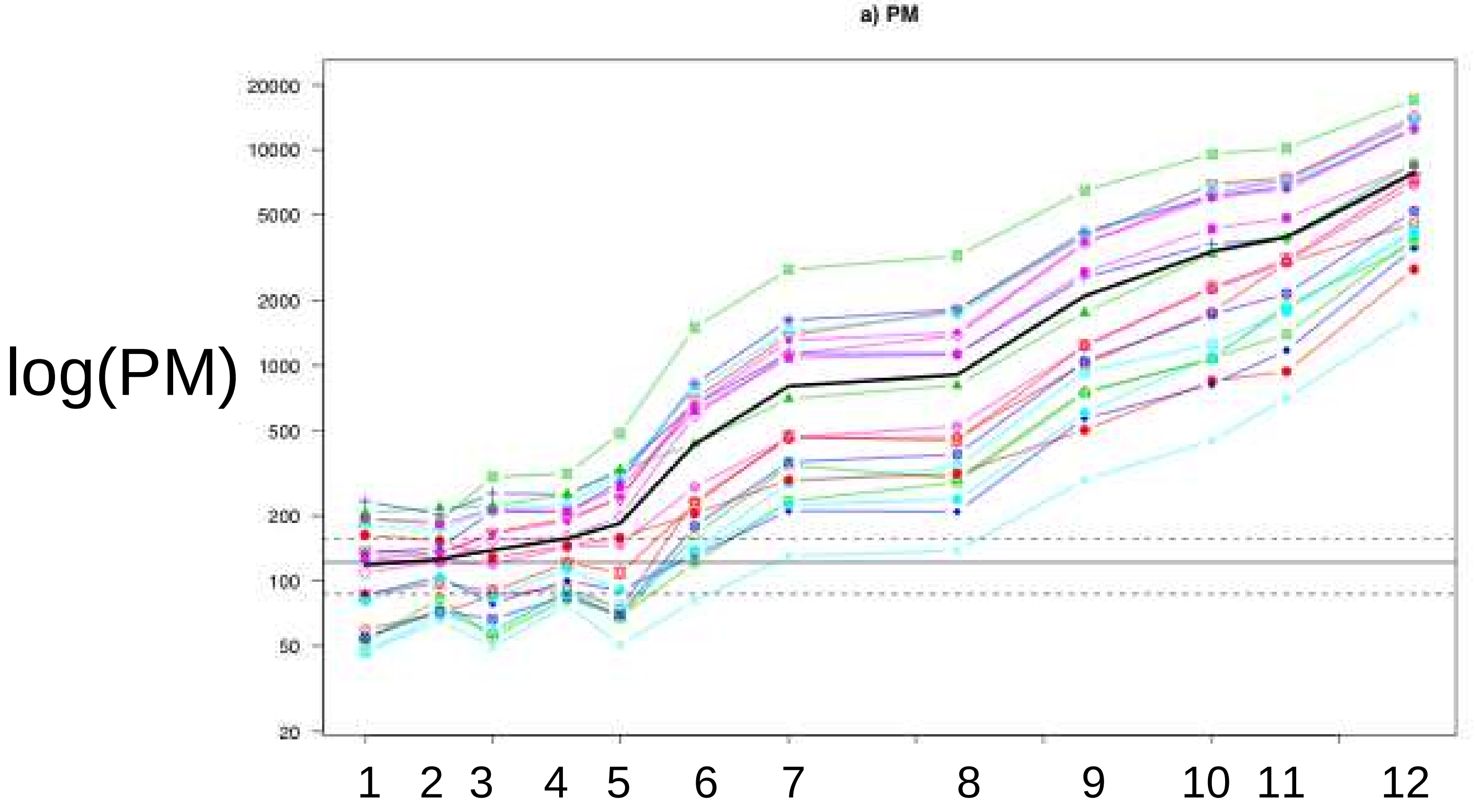
For a particular SNP we now have K added signals:

$$(PM_1, PM_2, ..., PM_K)$$

which are measures of the same thing - the CN.  However, they have slightly different sequences, so their hybridization efficiency might differ.

# Probe-level summarization

## *- different probes respond differently*



12 arrays with different expression levels

18 probes for the same probe set

Example:

$\log_2(PM_1) = \log_2(PM_2) + a_1$

=>

$PM_1 = \phi_1 * PM_2$

$(\phi_1 = 2^{a_1})$

# Probe-level summarization

## *- probe affinity model*

For a particular SNP, the total CN signal for sample *i=1,2,...,I* is: $\theta_i$

Which we observe via K probe signals: $(PM_{i1}, PM_{i2}, ..., PM_{iK})$

rescaled by probe affinities: $(\phi_1, \phi_2, ..., \phi_K)$

A **multiplicative model for the observed PM signals** is then:

$$PM_{ik} = \phi_k * \theta_i + \xi_{ik}$$

where $\xi_{ik}$ is noise.

# Probe-level summarization

## *- the log-additive model*

For one SNP, the model is:

$$PM_{ik} = \phi_k * \theta_i + \xi_{ik}$$

Take the logarithm on both sides:

$$\log_2(PM_{ik}) = \log_2(\phi_k * \theta_i + \xi_{ik})$$

¼ $\log_2(\phi_k * \theta_i) + \varepsilon_{ik}$

$$= \log_2\phi_k + \log_2\theta_i + \varepsilon_{ik}$$

Sample *i=1,2,...,I*, and probe *k=1,2,...,K*.

# Probe-level summarization

## *- the log-additive model*

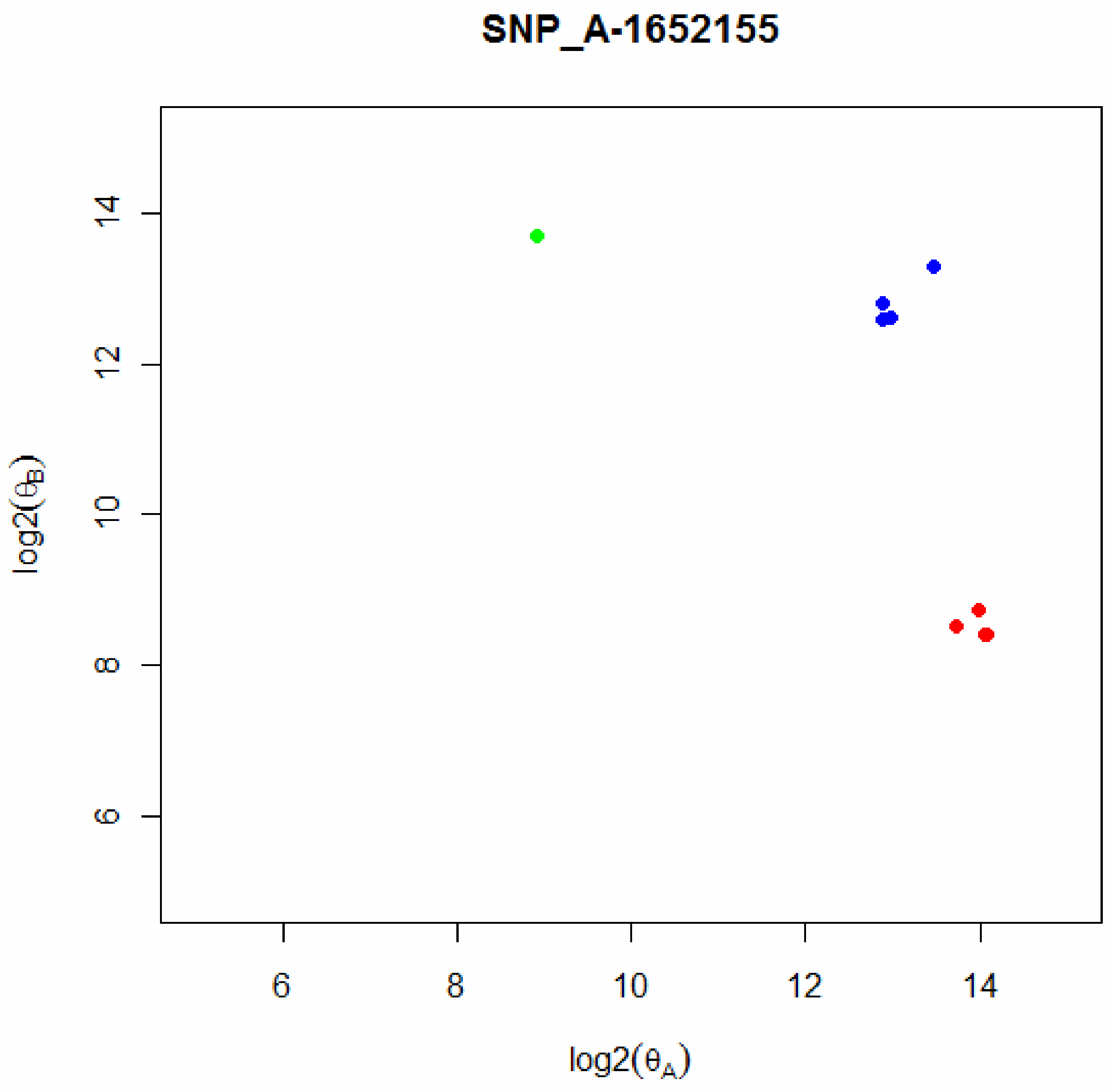
With multiple arrays i=1,2,...,I, we can estimate the probe-affinity parameters $\{\phi_k\}$ and therefore also the "chip effects" $\{\theta_i\}$ in the model:

$$\log_2(PM_{ik}) = \log_2\phi_k + \log_2\theta_i + \varepsilon_{ik}$$

**Conclusion:** We have summarized signals $(PM_{Ak}, PM_{Bk})$ for probes k=1,2,...,K into **one signal $\theta_i$ per sample**.

# Very brief on existing genotyping algorithms

# Allele-specific estimates ($\theta_{ijA}$,$\theta_{ijB}$)



# Idea of RLMM, BRLMM, CRLMM

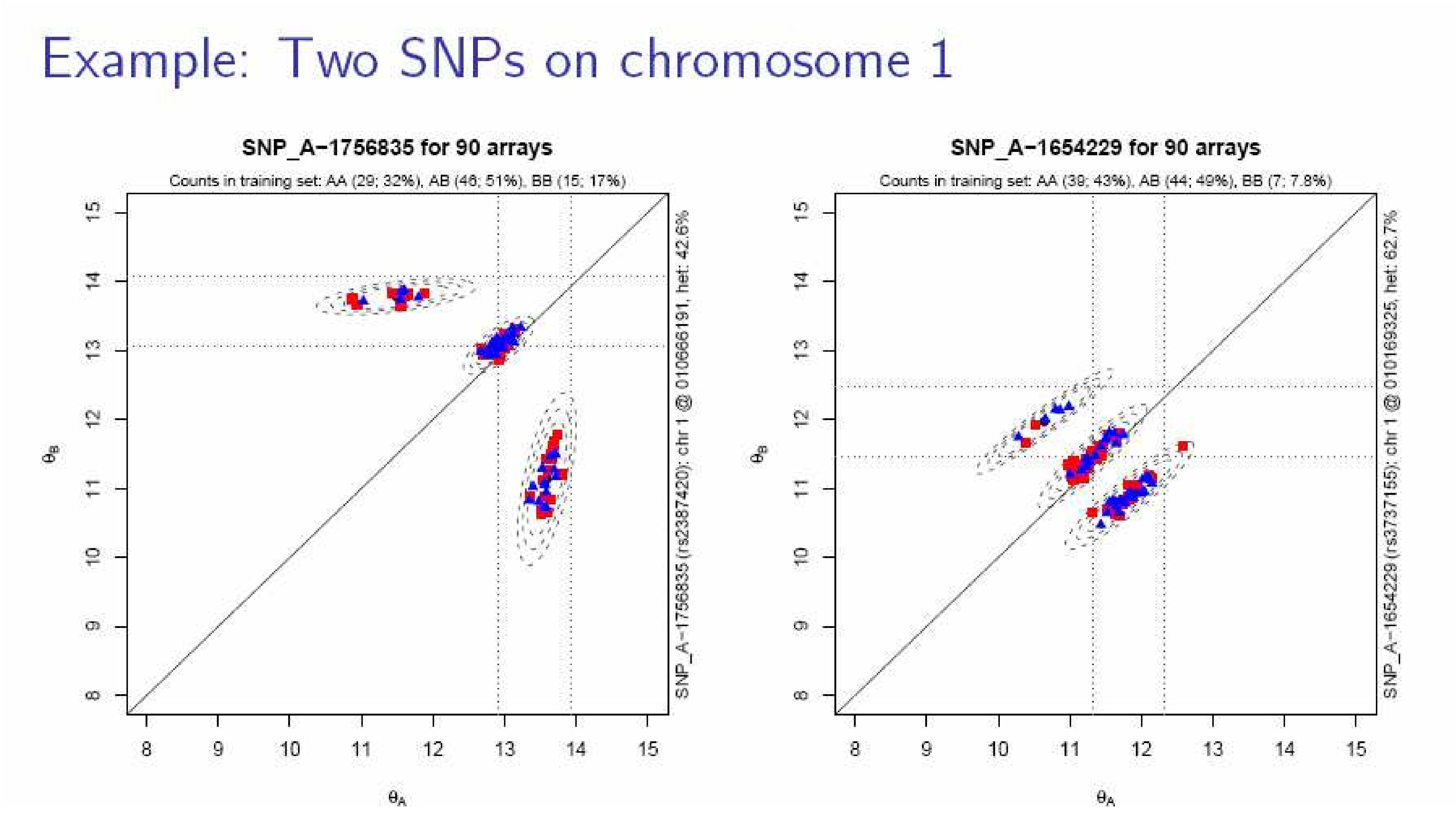
**Find genotype regions for each SNP:**

- Pick a high-quality training data set for which we know the true genotypes, e.g. the 270 HapMap samples.
- Estimate ($\theta_{ijA}$,$\theta_{ijB}$) for all samples and SNPs.
- For each SNP, find the regions for all samples with AA, then with AB, and the with BB.
  - The regions will differ slightly between SNPs.
- (Bayesian modelling of prior SNP regions)

**For a new sample:**

- For each SNP, identify the trained genotype region that is closest to its ($\theta_{ijA}$,$\theta_{ijB}$). That will be the genotype.

# Calling genotyping in ($\theta_{ijA}$,$\theta_{ijB}$)



## Example: Two SNPs on chromosome 1

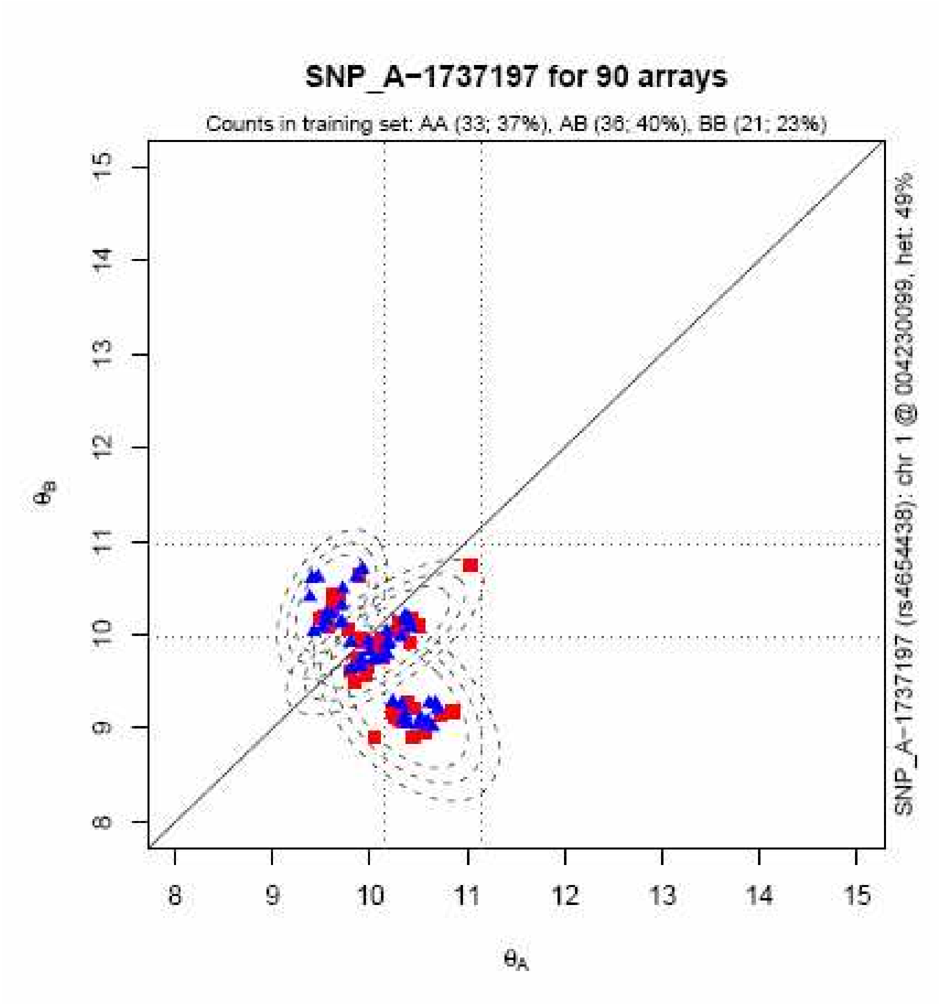
**SNP_A−1756835 for 90 arrays**

Counts in training set: AA (29; 32%), AB (46; 51%), BB (15; 17%)

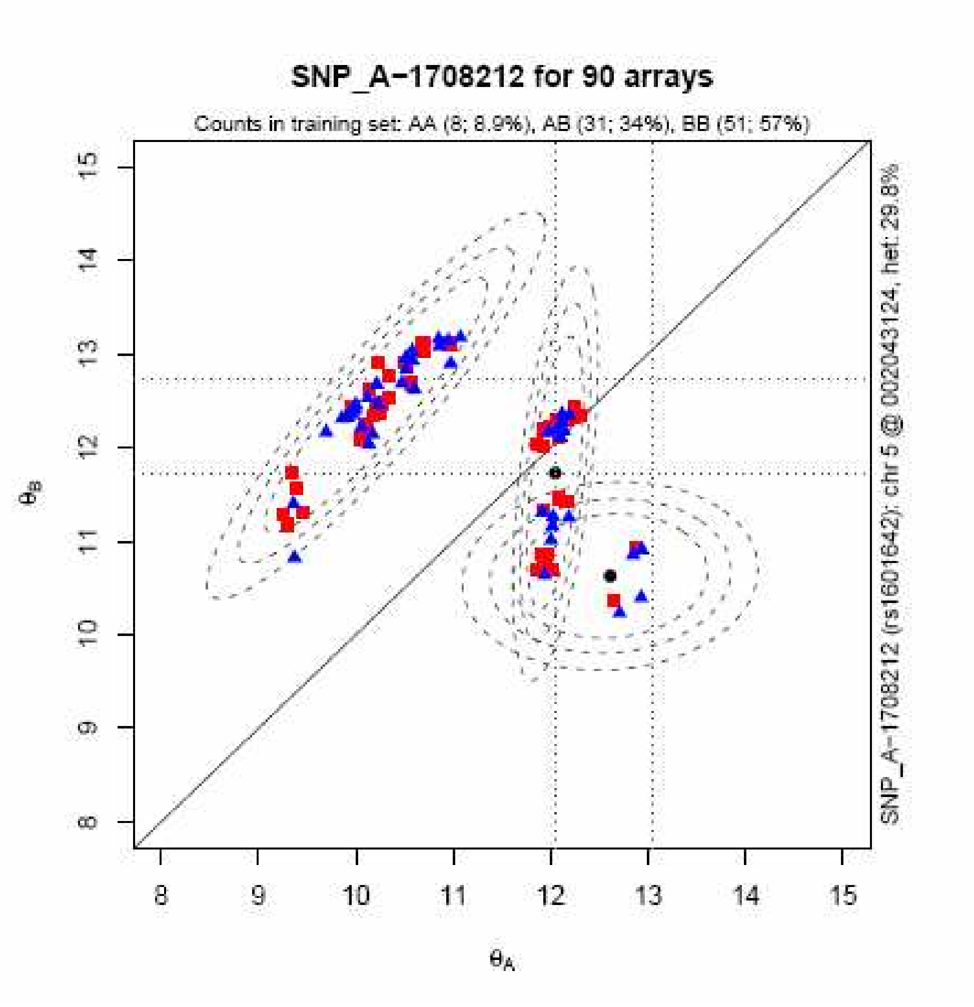
**SNP_A−1654229 for 90 arrays**

Counts in training set: AA (39; 43%), AB (44; 49%), BB (7; 7.8%)

# For some SNPs it is harder to distinguish the genotype groups

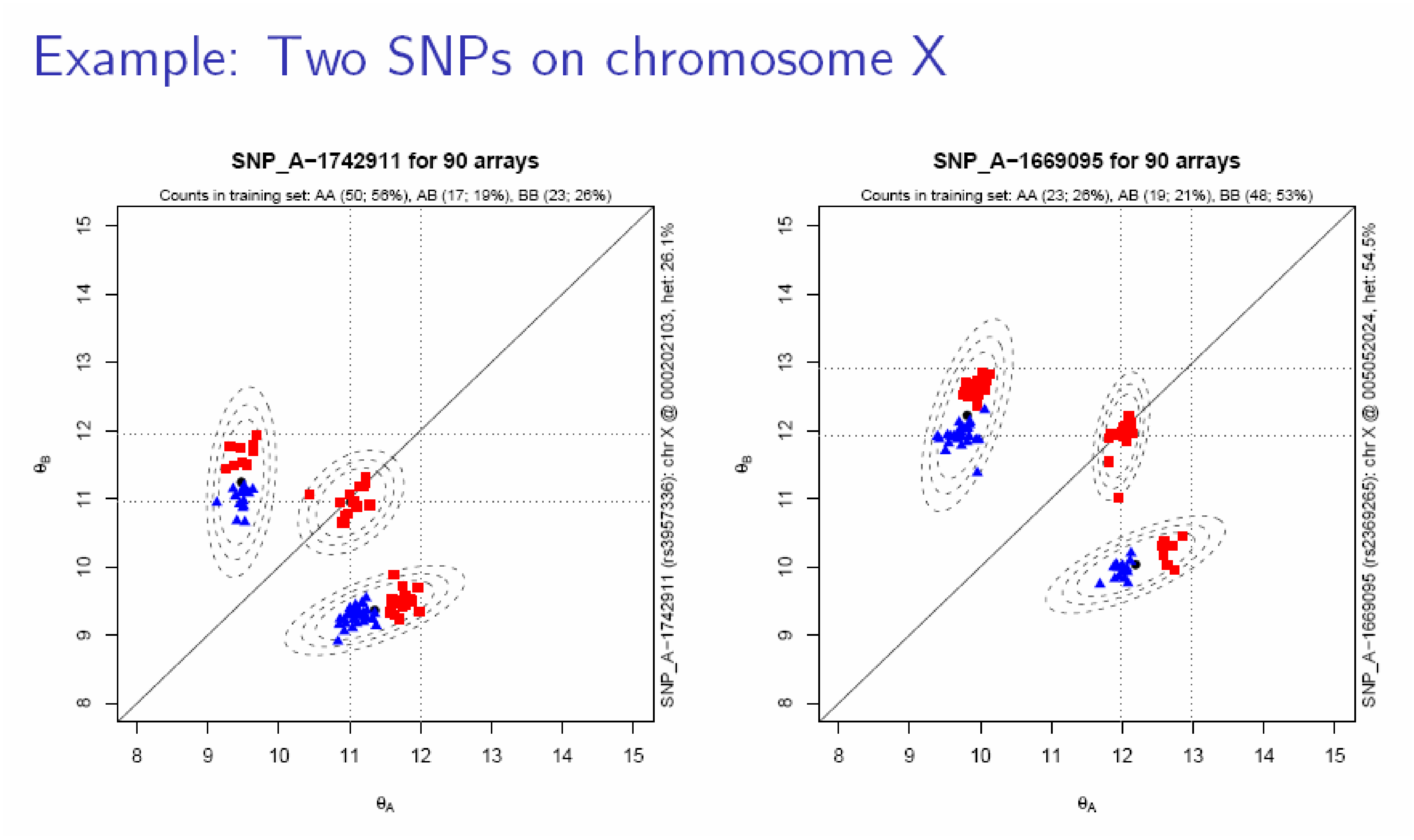

SNP_A−1737197 for 90 arrays

Counts in training set: AA (33; 37%), AB (36; 40%), BB (21; 23%)


SNP_A−1708212 for 90 arrays

Counts in training set: AA (8; 8.9%), AB (31; 34%), BB (51; 57%)

# Careful: Genotyping algorithms often assume diploid states, not CN aberrations



## Example: Two SNPs on chromosome X

SNP_A−1742911 for 90 arrays

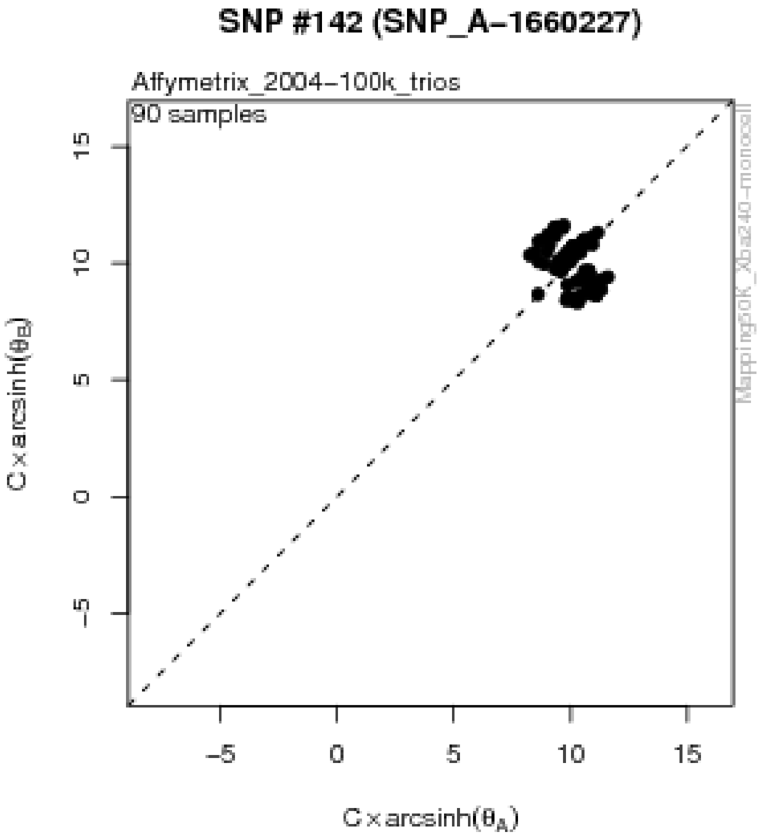
Counts in training set: AA (50; 56%), AB (17; 19%), BB (23; 26%)

SNP_A−1669095 for 90 arrays

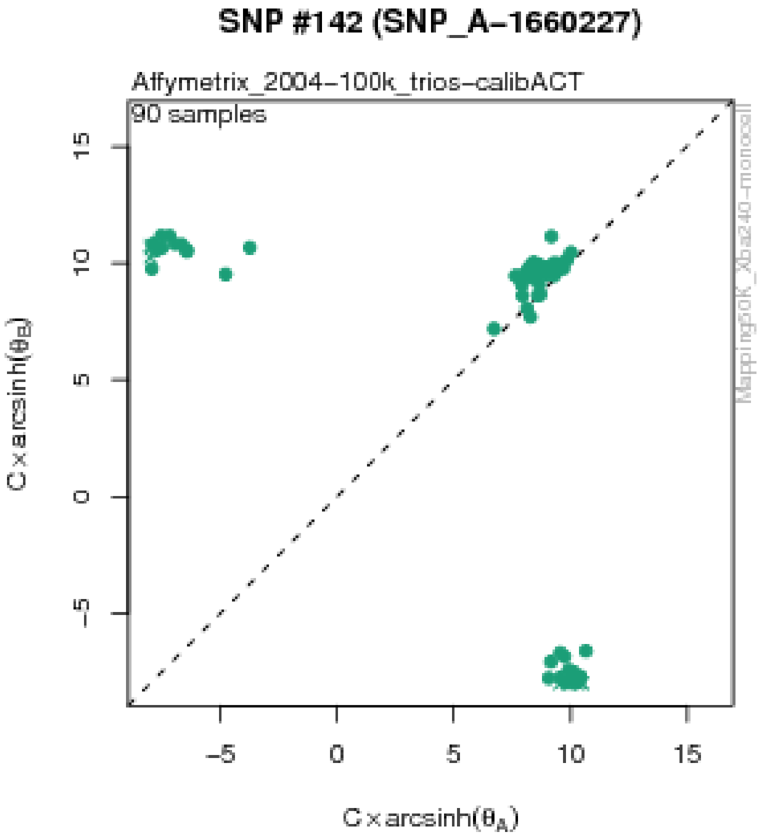
Counts in training set: AA (23; 26%), AB (19; 21%), BB (48; 53%)

# Crosstalk calibration (incl. the removal of the offset) gives better separation of AA, AB, BB.

**Without calibration:**

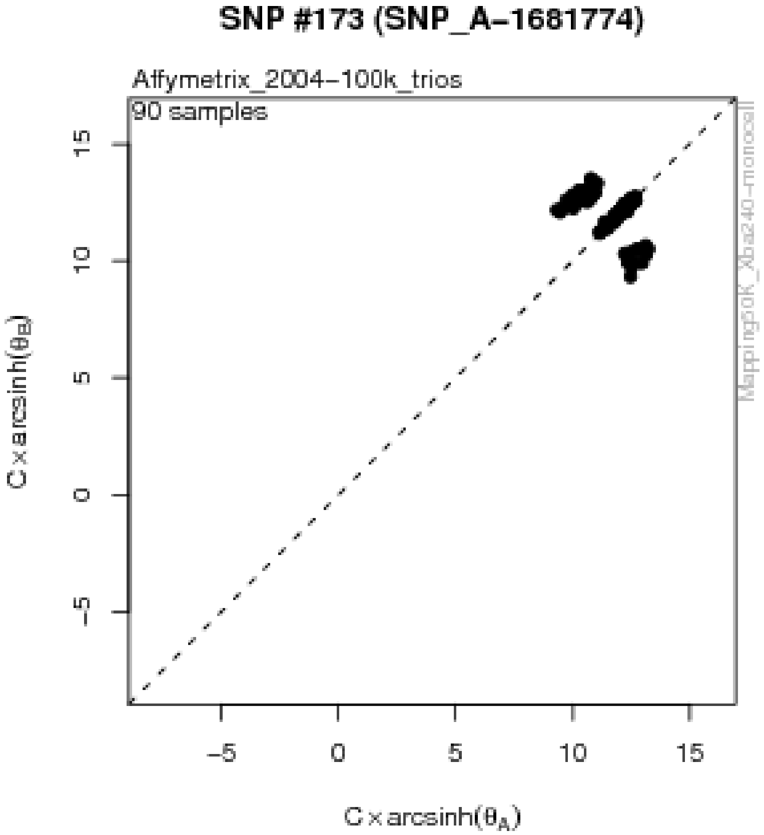


**With calibration:**

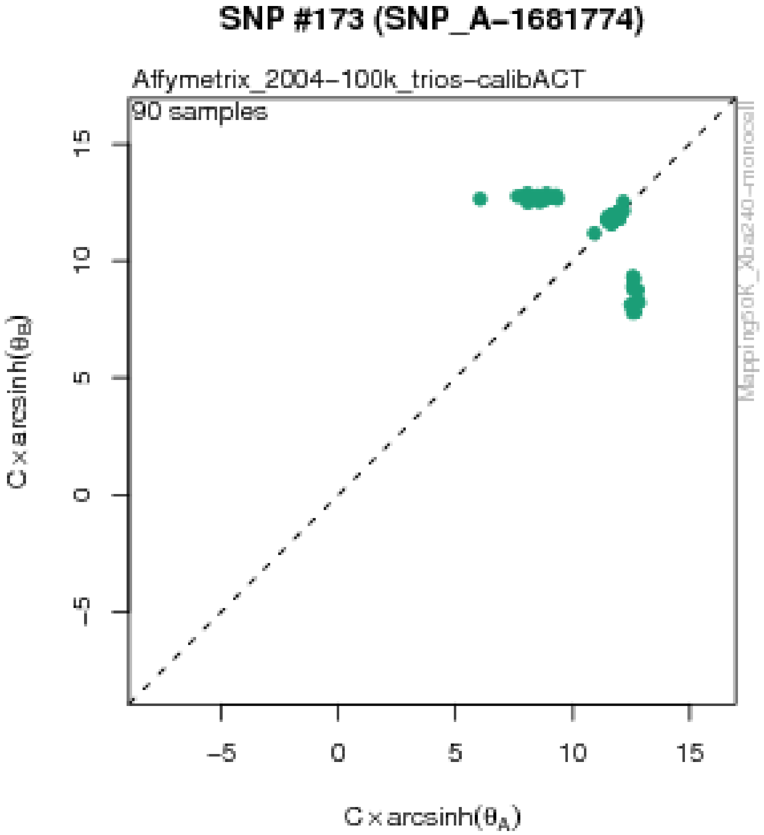


# A more suttle example

## Without calibration:

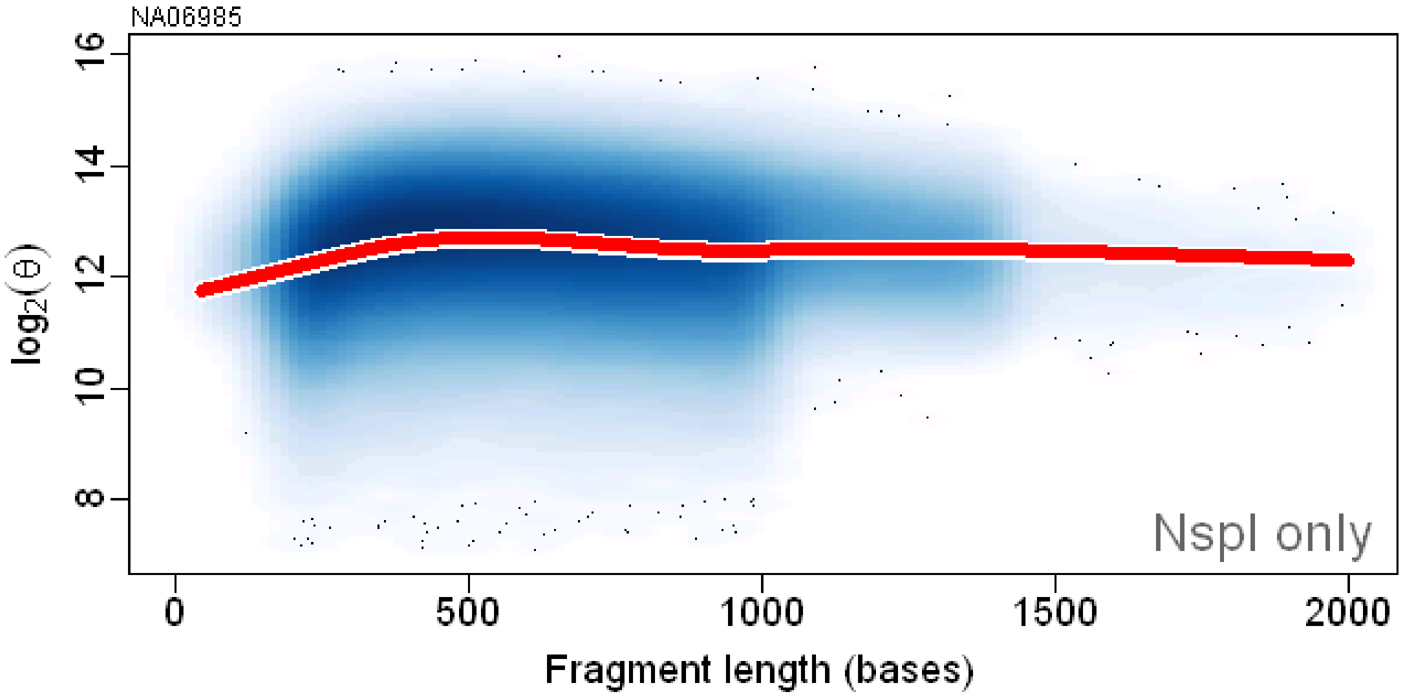


## With calibration:



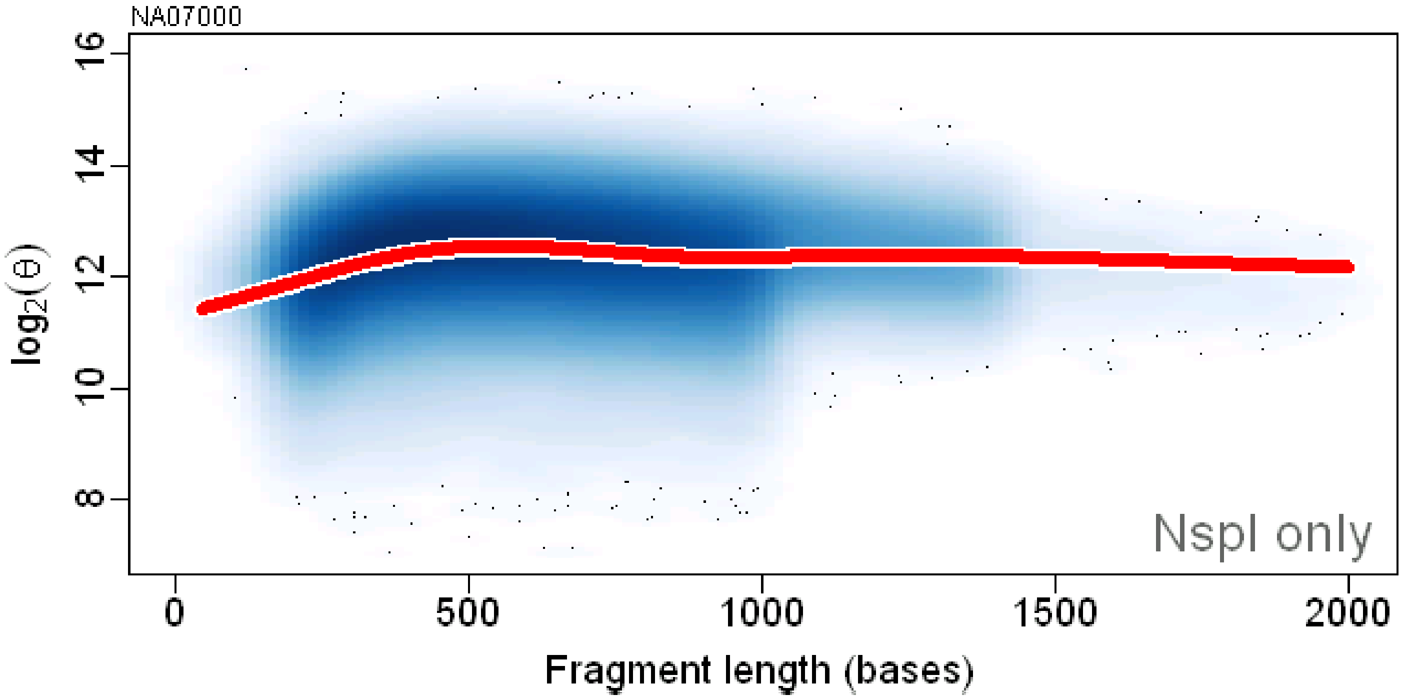
# Fragment length normalization

# Longer fragments are amplified less by PCR
# Observed as weaker θ signals



Note, here we study the effect on non-polymorphic signals, that is, for SNPs we first do $\theta_{ij} = \theta_{ijA} + \theta_{ijB}$.

# Slightly different effects between arrays adds extra variation



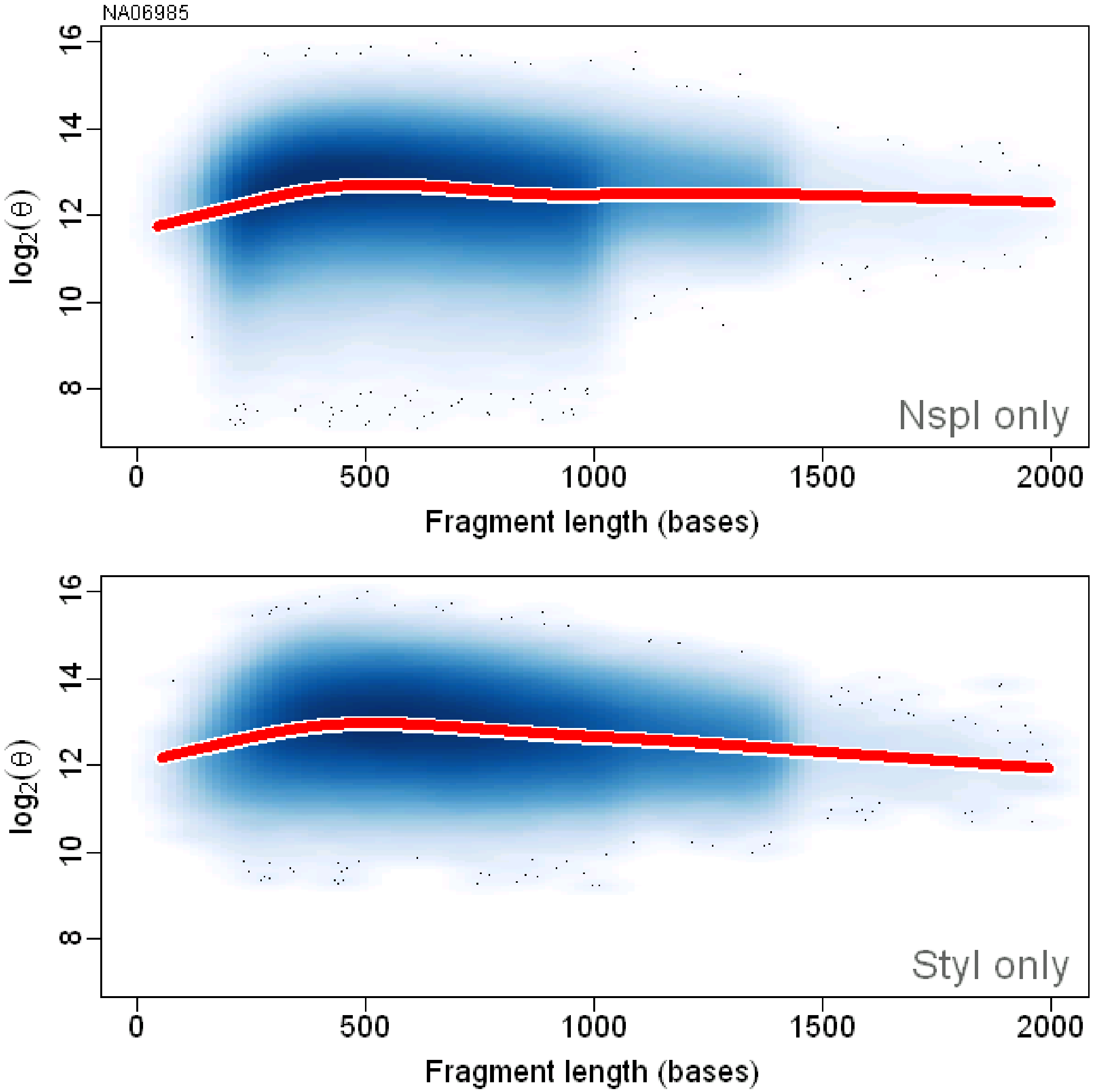
# Fragment-length normalization for multi-enzyme hybridizations

- For **GWS5 and GWS6**, the DNA is fragmented using two enzymes.

- For all CN probes, all targets originate from *Nsp*I digestion.

- For SNP probes, some targets originate exclusively from *Nsp*I, exclusively from *Sty*I, or from **both *Nsp*I and *Sty*I**.

# Fragment-length effects for co-hybridized enzymes are assumed to be additive



# Fragment-length normalization for co-hybridized enzymes

Multi-enzyme normalization model:

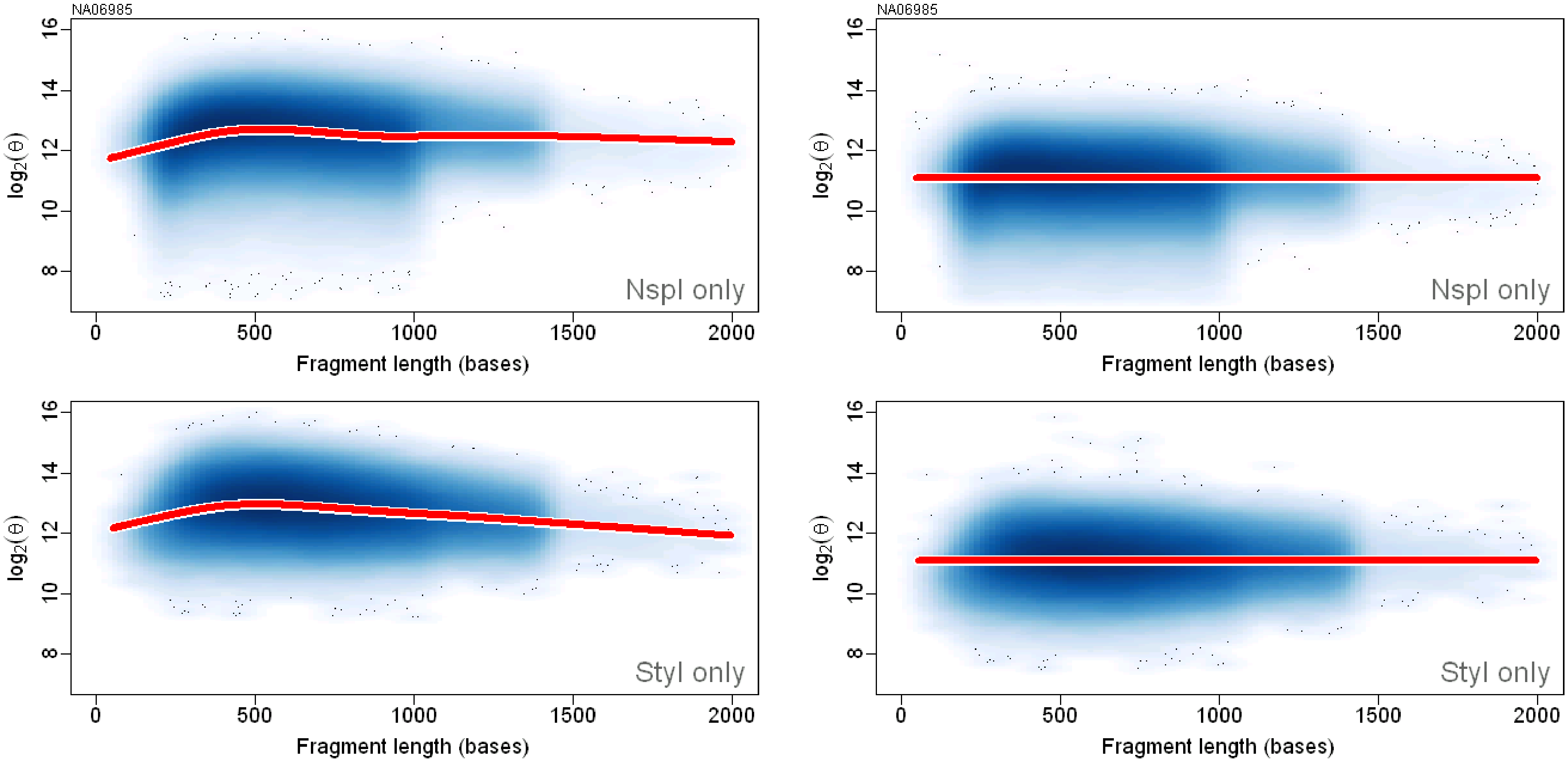
$$\log_2\theta_j^* \leftarrow \log_2\theta_j - \delta^*$$

$$\delta^* = \delta(\lambda_{Nsp,j}, \lambda_{Sty,j}) = \text{correction}$$

$\lambda_{Nsp}, \lambda_{Sty}$ = fragment lengths in *Nsp*I and *Sty*I.

# Multi-enzyme fragment-length normalization removes the effects

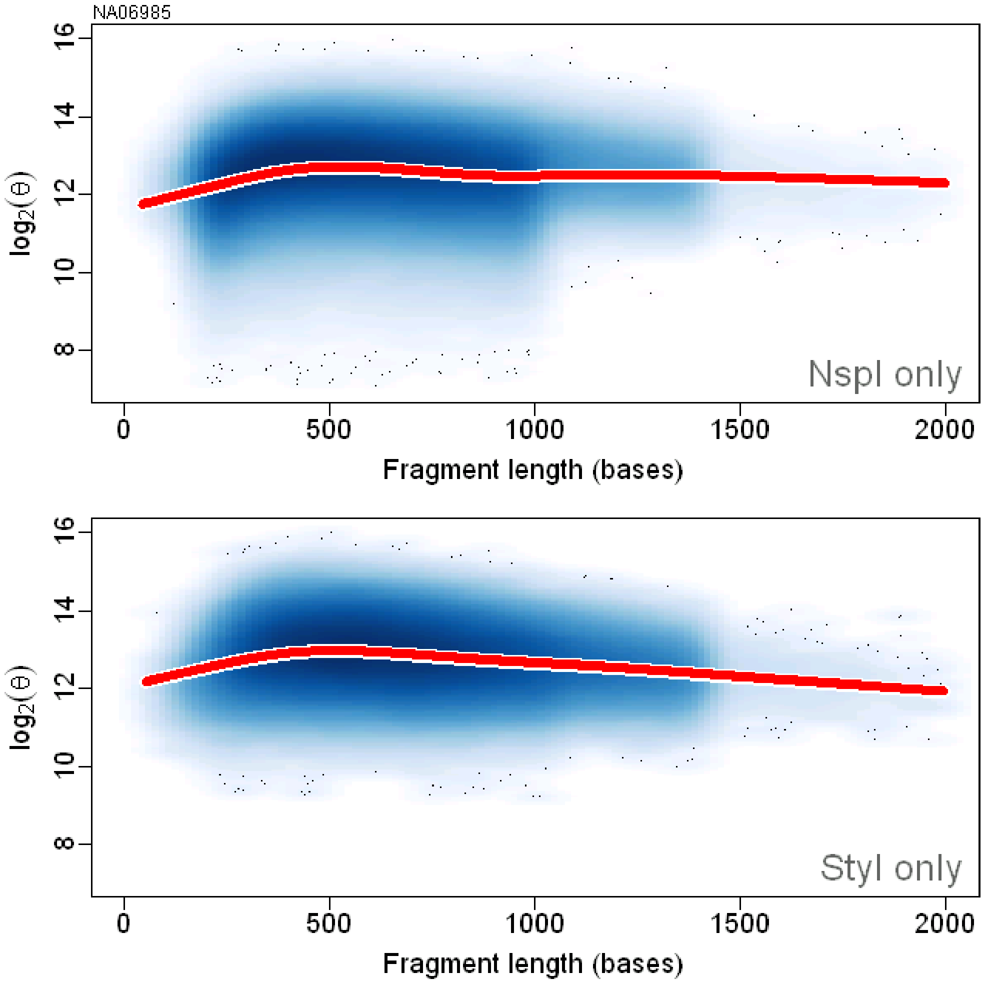
**Array #1 before** **Array #1 after**



# Multi-enzyme fragment-length normalization removes the effects

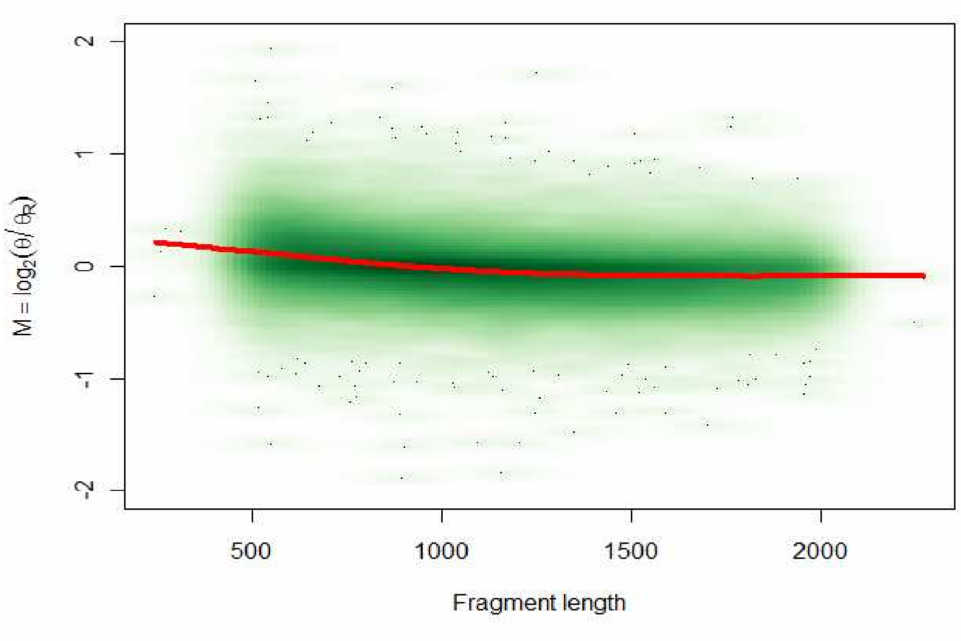
Array #1 before

Array #1 after

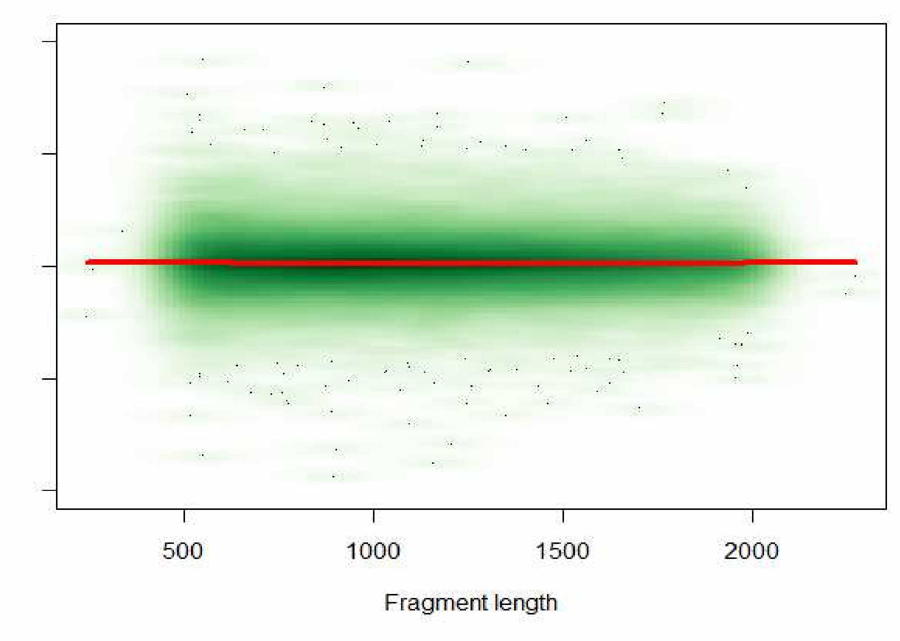


# Removing the effect on the chip effects, will also remove the effect on CN log ratios

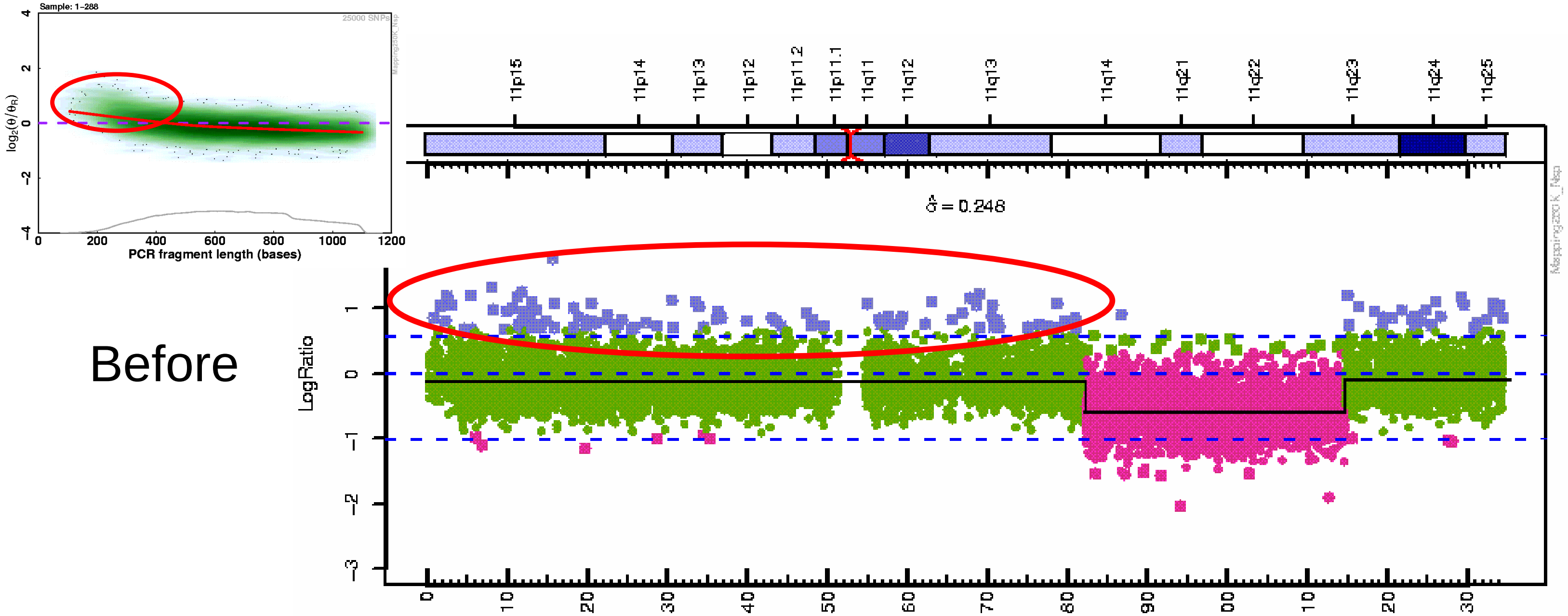
**Before:**



**After:**

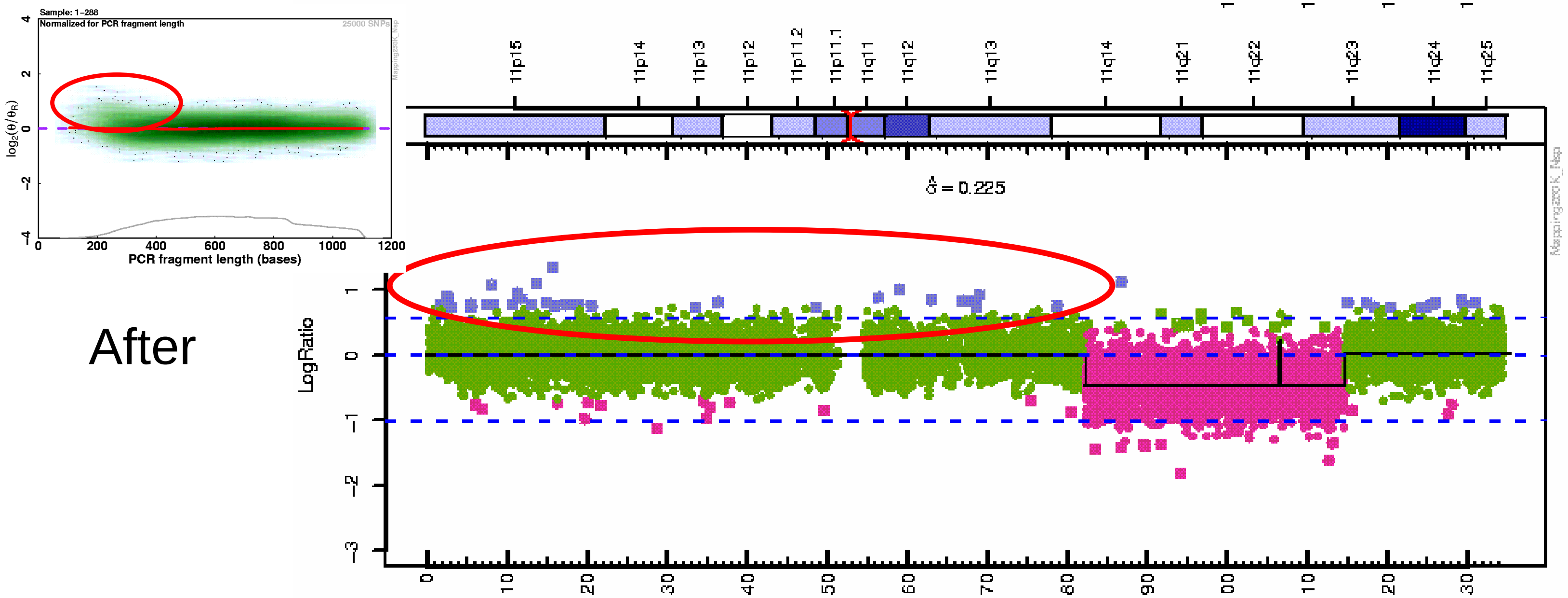


Before



σ = 0.246

After



σ = 0.225

# aroma.affymetrix

You will need:

- Affymetrix CDF, e.g. GenomeWideSNP_6.cdf
- A Unit Fragment Length file, e.g. GenomeWideSNP_6.ufl

```
fln <- FragmentLengthNormalization(ces, target="zero")
cesN <- process(fln)
```

# Finally,
# a convenient transform

# Bijective transform of $(\theta_{ijA},\theta_{ijB})$ in to $(\theta_{ij}, \beta_{ij})$.

Transform $(\theta_{ijA},\theta_{ijB})$ to $(\theta_{ij}, \beta_{ij})$ by:

Non-polymorphic SNP signal: $\theta_{ij} = \theta_{ijA} + \theta_{ijB}$

Allele B frequency signal: $\beta_{ij} = \theta_{ijB} / \theta_{ij}$

A CN probe does not have a $\beta_{ij}$. However, both CN probes and SNPs have a non-polymorphic signal $\theta_{ij}$.

We expect the following:

Genotype BB: $\theta_{ijB} >> \theta_{ijA}$ => $\beta_{ij} \approx 1$

Genotype AA: $\theta_{ijB} << \theta_{ijA}$ => $\beta_{ij} \approx 0$

Genotype AB: $\theta_{ijB} \approx \theta_{ijA}$ => $\beta_{ij} \approx ½$

Thus, $\theta_{ij}$ carry information on CN and $\beta_{ij}$ on genotype.

# Copy numbers are estimated relative to a reference

Relative copy numbers:

$$C_{ij} = 2*(\theta_{ij} / \theta_{Rj})$$

Alternatively, log-ratios:

$$M_{ij} = \log_2(\theta_{ij} / \theta_{Rj})$$

Note: $C_{ij}$ is defined also when $\theta <= 0$, but $M_{ij}$ is not.

Array i=1,2,...,I.  Locus j=1,2,...,J.

# Allele-specific copy numbers

Allele-specific copy numbers ($C_{ijA}$,$C_{ijB}$):

$$\mathbf{C_{ijA} = 2*(\theta_{ijA} / \theta_{Rj})}$$
$$\mathbf{C_{ijB} = 2*(\theta_{ijB} / \theta_{Rj})}$$

Note that,

1. $C_{ij} = C_{ijA}+C_{ijB} = 2*(\theta_{ijA}+ \theta_{Rj}) / \theta_{Rj} = 2*(\theta_{ij} / \theta_{Rj})$

2. $C_{ijB}/C_{ij} = [2*(\theta_{ijB} / \theta_{Rj})] / [2*(\theta_{ij} / \theta_{Rj})] = \theta_{ijB} / \theta_{ij} = \beta_{ij}$

3. $C_{ijB} = 2*(\theta_{ijB} / \theta_{ij}) * (\theta_{ij}/ \theta_{Rj}) = \beta_{ij} * C_{ij}$

# aroma.affymetrix

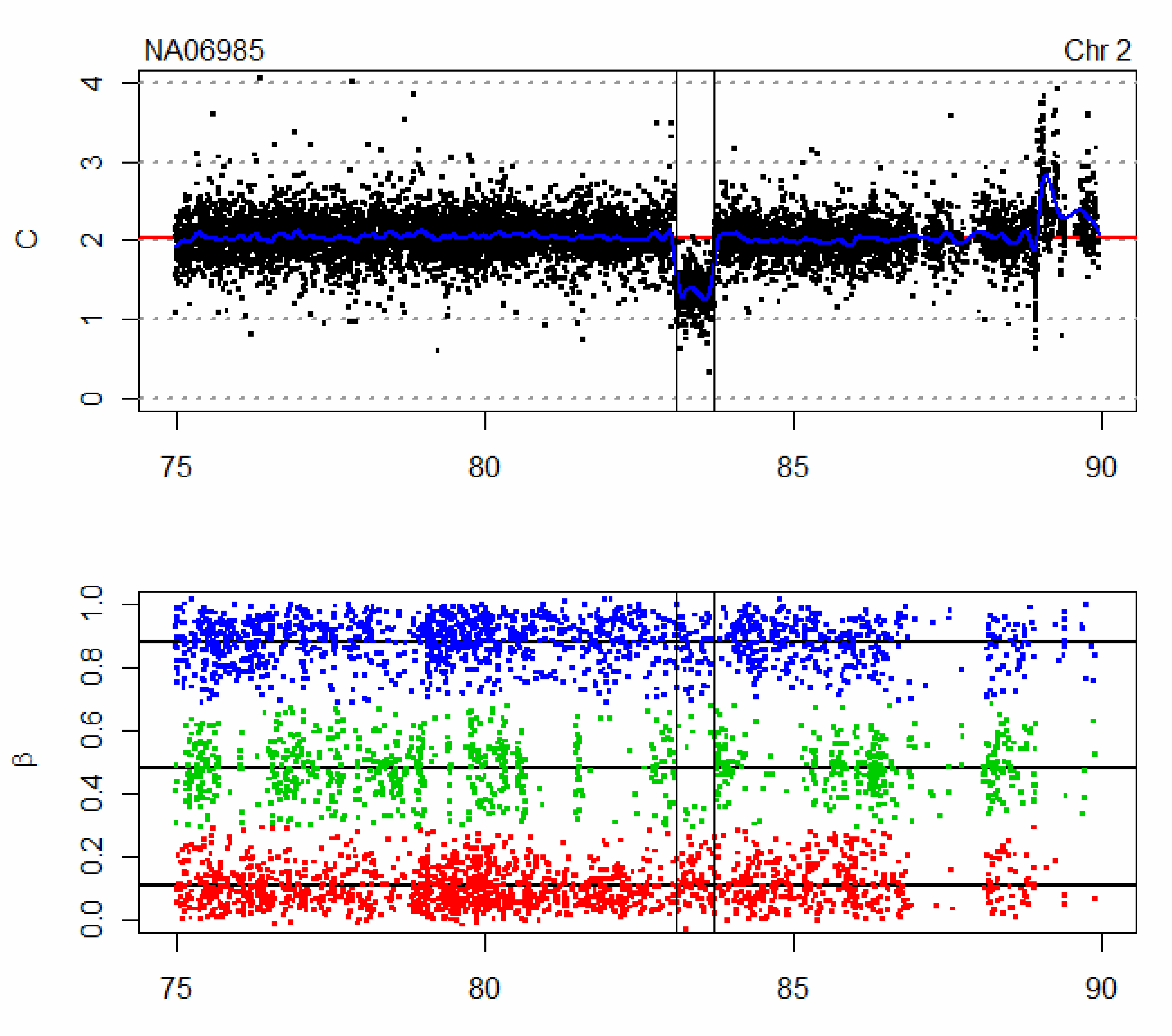
You will need:

- Affymetrix CDF, e.g. GenomeWideSNP_6.cdf
- A Unit Genome Position file, e.g. GenomeWideSNP_6.ugp

```
data <- extractTotalAndFreqB(cesN)
theta <- data[,"total",]
freqB <- data[,“freqB",]
```

```
Plot Array 3 along chromosome 2
gi <- getGenomeInformation(cdf)
units <- getUnitsOnChromosome(gi, 2)
pos <- getPositions(gi, units)
plot(pos, theta[units,3])
plot(pos, freqB[units,3])
```

# CN and freqB - (C,β) - along genome



# Selecting reference samples

# The choice of reference sample(s) is important
# - *A real example from my postdoc projects*

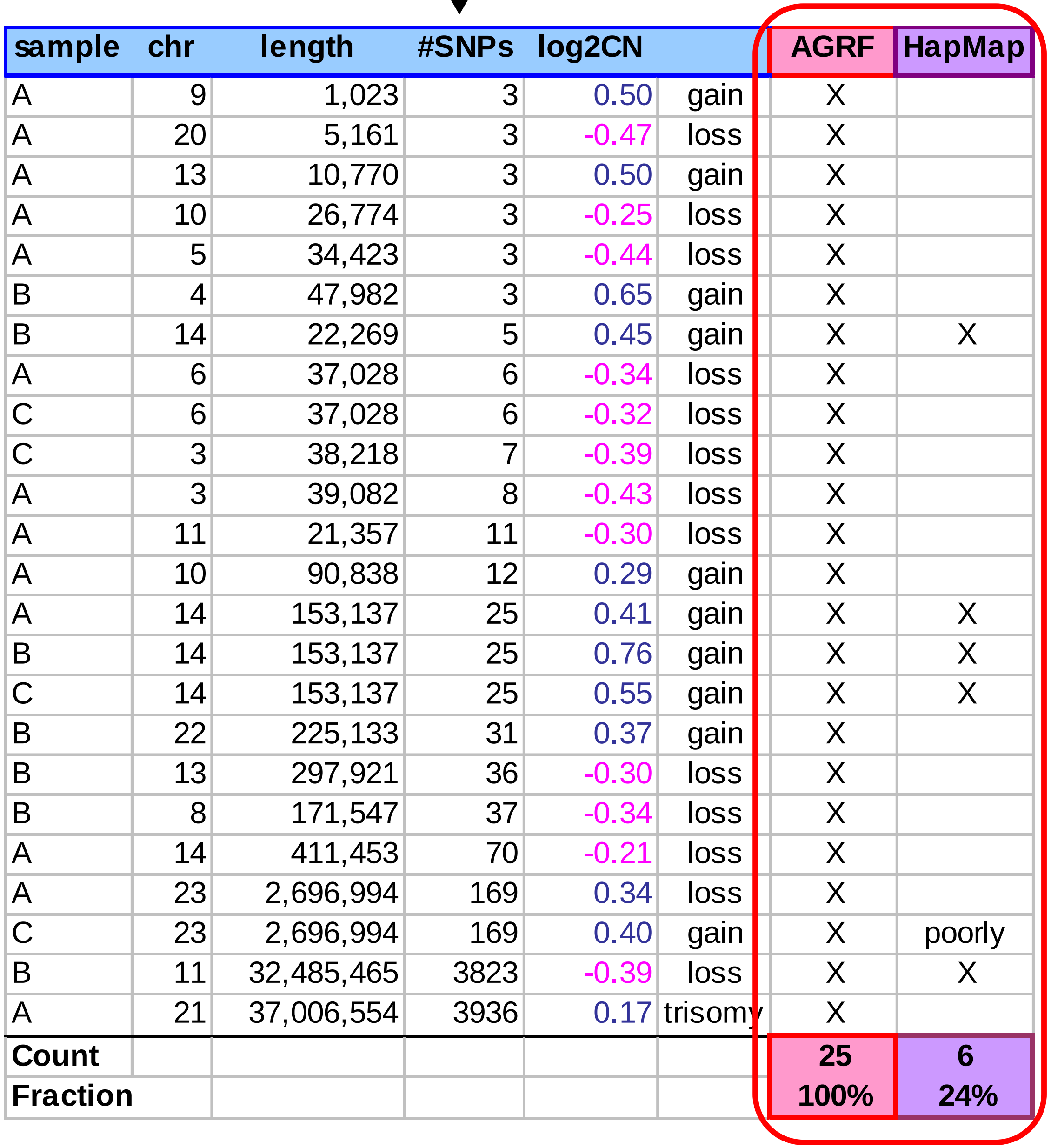
Data set:

- 3 Affymetrix 250K Nsp arrays.
- Processed at the AGRF / WEHI, Melbourne, Australia.

Reference sets:

- Public: 270 normal HapMap arrays (“gold standard”).
- In-house: 11 anonymous/unknown(!) AGRF arrays.

Segmentation regions found with reference set:
(i) 11 in-house samples and (i) 270 HapMap samples



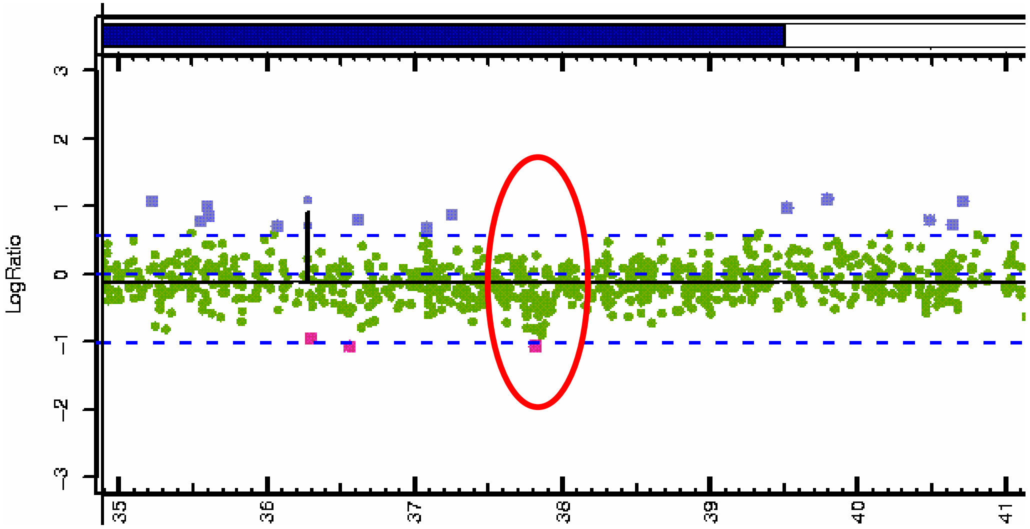
| sample | chr | length | #SNPs | log2CN | | AGRF | HapMap |
|---|---|---|---|---|---|---|---|
| A | 9 | 1,023 | 3 | 0.50 | gain | X | |
| A | 20 | 5,161 | 3 | -0.47 | loss | X | |
| A | 13 | 10,770 | 3 | 0.50 | gain | X | |
| A | 10 | 26,774 | 3 | -0.25 | loss | X | |
| A | 5 | 34,423 | 3 | -0.44 | loss | X | |
| B | 4 | 47,982 | 3 | 0.65 | gain | X | |
| B | 14 | 22,269 | 5 | 0.45 | gain | X | X |
| A | 6 | 37,028 | 6 | -0.34 | loss | X | |
| C | 6 | 37,028 | 6 | -0.32 | loss | X | |
| C | 3 | 38,218 | 7 | -0.39 | loss | X | |
| A | 3 | 39,082 | 8 | -0.43 | loss | X | |
| A | 11 | 21,357 | 11 | -0.30 | loss | X | |
| A | 10 | 90,838 | 12 | 0.29 | gain | X | |
| A | 14 | 153,137 | 25 | 0.41 | gain | X | X |
| B | 14 | 153,137 | 25 | 0.76 | gain | X | X |
| C | 14 | 153,137 | 25 | 0.55 | gain | X | X |
| B | 22 | 225,133 | 31 | 0.37 | gain | X | |
| B | 13 | 297,921 | 36 | -0.30 | loss | X | |
| B | 8 | 171,547 | 37 | -0.34 | loss | X | |
| A | 14 | 411,453 | 70 | -0.21 | loss | X | |
| A | 23 | 2,696,994 | 169 | 0.34 | loss | X | |
| C | 23 | 2,696,994 | 169 | 0.40 | gain | X | poorly |
| B | 11 | 32,485,465 | 3823 | -0.39 | loss | X | X |
| A | 21 | 37,006,554 | 3936 | 0.17 | trisomy | X | |
| **Count** | | | | | | **25** | **6** |
| **Fraction** | | | | | | **100%** | **24%** |

# Stronger signal with in-house reference set

*Example: A 37 SNP deletion on chr 8*

**HapMap**

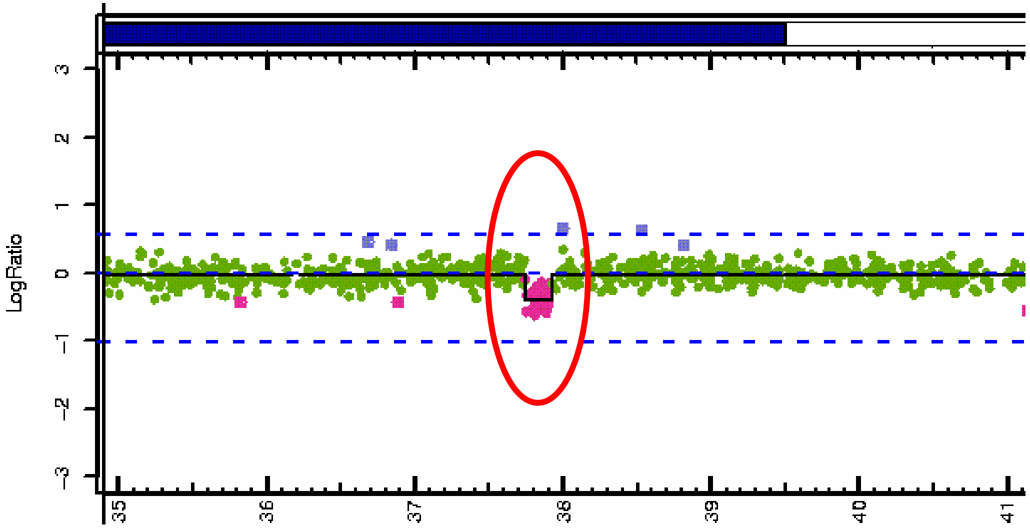
270 samples



σ = 0.237

**AGRF**

11 anonymous samples



σ = 0.126

# Conclusion

It is better to use a small,
even unknown, reference set
from the same microarray lab
than an external reference set.

# Summary of CRMA v2

| | **CRMA v2** |
|---|---|
| ***Preprocessing***<br>*(probe signals)* | 1. Allelic crosstalk calibration<br>2. Probe-sequence normalization |
| ***Summarization*** | Robust averaging:<br>CN probes: $\theta_{ij} = PM_{ij}$<br>SNPs: $\theta_{ijA} = \mathrm{median}_k(PM_{ijkA})$<br>$\theta_{ijB} = \mathrm{median}_k(PM_{ijkB})$<br>array *i*, loci *j*, probe *k.* |
| ***Post-processing*** | PCR fragment-length normalization |
| ***Transform*** | $(\theta_{ijA},\theta_{ijB}) \Rightarrow (\theta_{ij}, \beta_{ij})$<br>$\theta_{ij} = \theta_{ijA}+\theta_{ijB}$, $\beta_{ij} = \theta_{ijB} / \theta_{ij}$ |
| ***Allele-specific &<br>total CNs*** | $C_{ijA} = 2*(\theta_{ijA} / \theta_{Rj})$ and $C_{ijB} = 2*(\theta_{ijA} / \theta_{Rj})$<br>$C_{ij} = 2*(\theta_{ij} / \theta_{Rj})$ reference *R* |

# Single array method

# CRMA v2 is a single-array preprocessing method

- CRMA v2 estimates chip effects of one array independently of other arrays.
  - It does not use prior parameter estimates etc.
  - A reference signals is only needed when calculating relative CNs, i.e. $C_i = 2*(\theta_i/\theta_R)$.

- Implications:
  - Tumor/normal studies can be done with only two hybrizations.
  - No need to rerun analysis when new arrays are added.
  - Large data sets can be processed on multiple machines.

# Evaluation

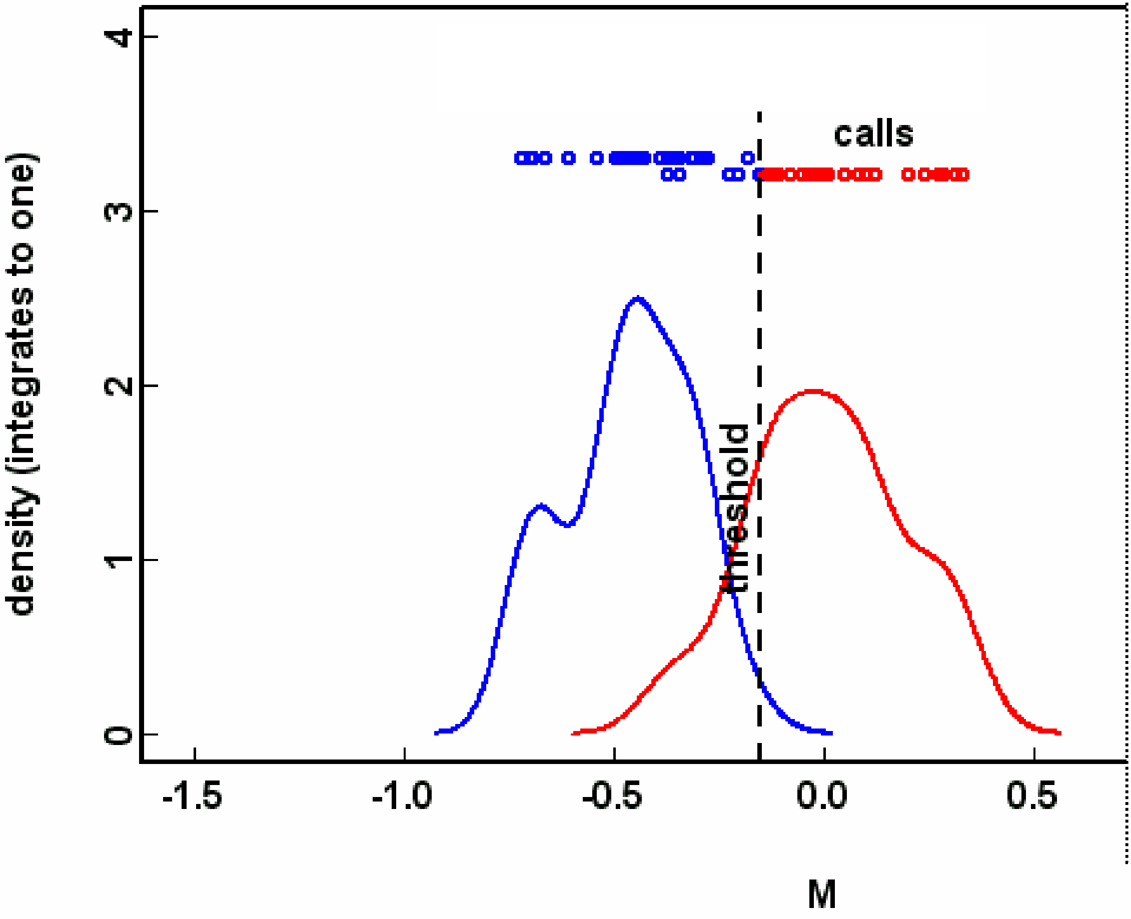
# Other methods

| | **single-array** | **multi-array** | **multi-array** |
|---|---|---|---|
| | ***CRMA v2*** | ***dChip***<br>**(Li & Wong 2001)** | ***CN5***<br>**(Affymetrix 2006)** |
| ***Preprocessing***<br>*(probe signals)* | allelic crosstalk.<br>probe-seq norm. | invariant-set | quantile |
| ***Summarization***<br>*(SNP signals $\theta$)*<br>***and total CNs*** | i) Robust avg.<br>ii) $\theta=\theta_A+\theta_B$ | i) $PM=PM_A+PM_B$<br>ii) multiplicative | i) log-additive<br>ii) $\theta=\theta_A+\theta_B$ |
| **Post-processing** | fragment-length.<br>(GC-content) | - | fragment-length.<br>GC-content.<br>Enzyme seq normalization.<br>Genome "wave" normalization |
| **Raw total CNs** | $M_{ij} = \log_2(\theta_{ij}/\theta_{Rj})$<br>$[\ C_{ij} = 2*(\theta_{ij}/\theta_{Rj})\ ]$ | $M_{ij} = \log_2(\theta_{ij}/\theta_{Rj})$ | $M_{ij} = \log_2(\theta_{ij}/\theta_{Rj})$ |

# How well can detect CN changes compare with other methods?

- Other methods:
  - Affymetrix ("CN5") estimates (software GTC v3).
  - dChip estimates (software dChip 2008).
- Data set:
  - 59 GWS6 HapMap samples (29 females & 30 males).
- Evaluation:
  - How well can we detect:
    - CN=1 among CN=2 (ChrX), and
    - CN=0 among CN=1 (ChrY)?
  - At full resolution and various amounts of smoothing.

# Calling samples for SNP_A-1920774



# **males**: 30
# **females**: 29

Call rule:
If $M_i$ < threshold, a **male**

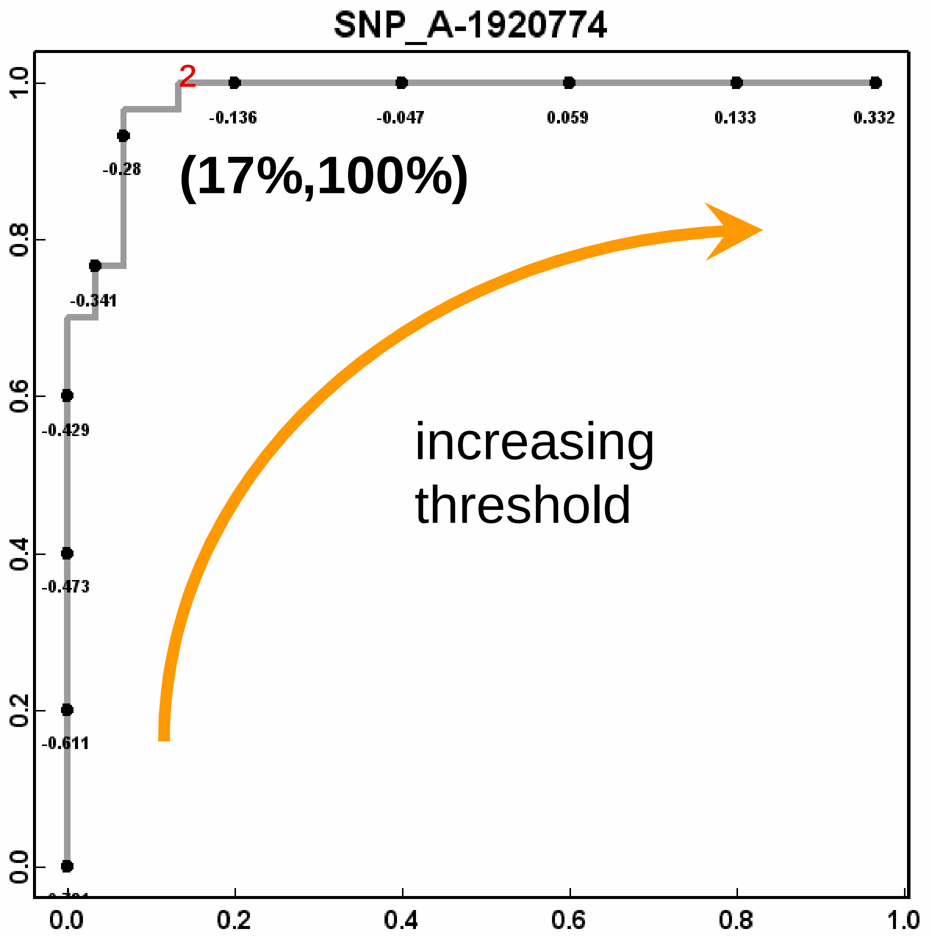
Calling a male male:
#True-positives: 30
**TP rate: 30/30 = 100%**

Calling a female male:
#False-positive : 5
**FP rate: 5/29 = 17%**

# Receiver Operator Characteristic (ROC)

**TP rate**

(correctly calling
a males male)



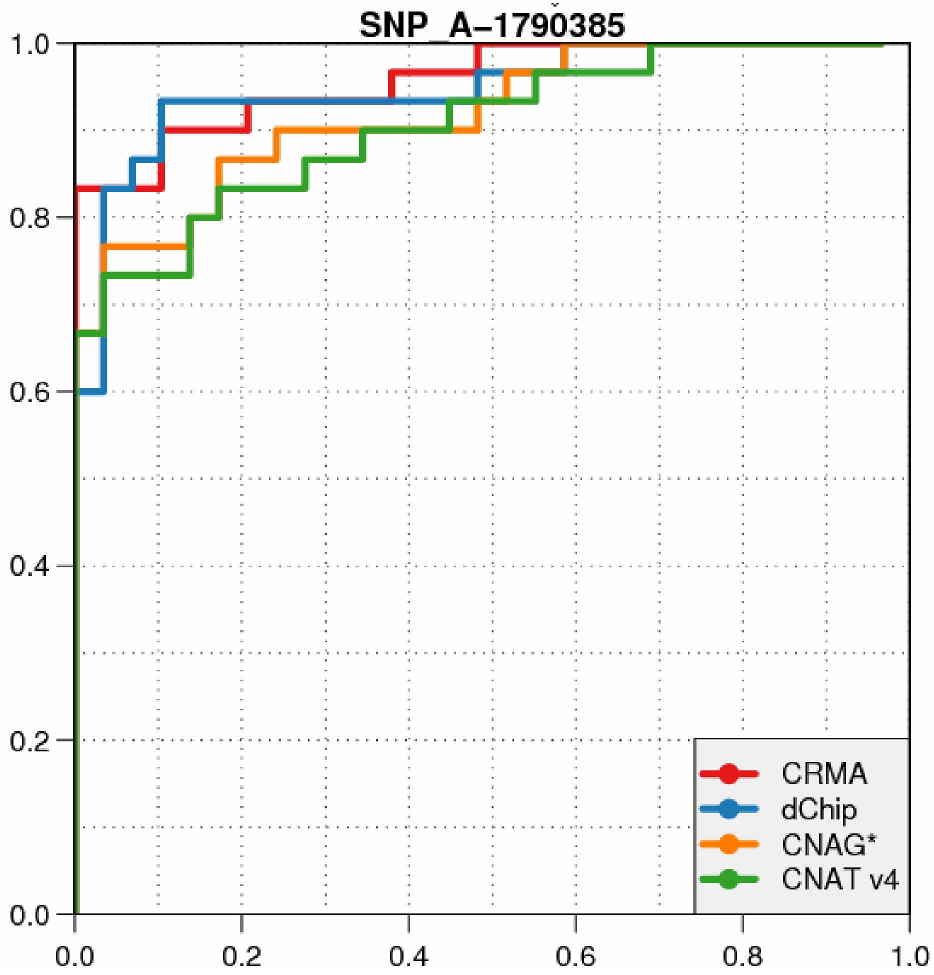
**(17%,100%)**

increasing
threshold

**FP rate**

(incorrectly calling females male)

# Single-SNP comparison
*A random SNP*



**TP rate**

(correctly calling a males male)

**FP rate**

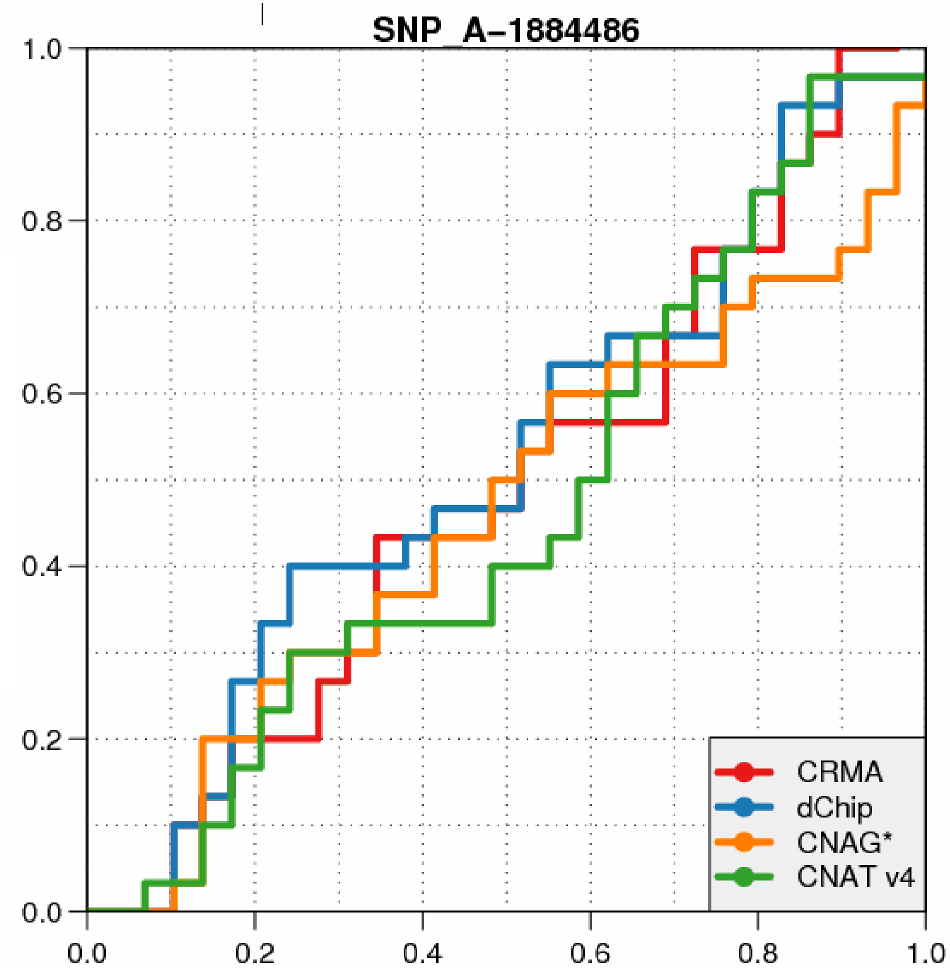
(incorrectly calling females male)

# Single-SNP comparison

*A non-differentiating SNP*

**TP rate**

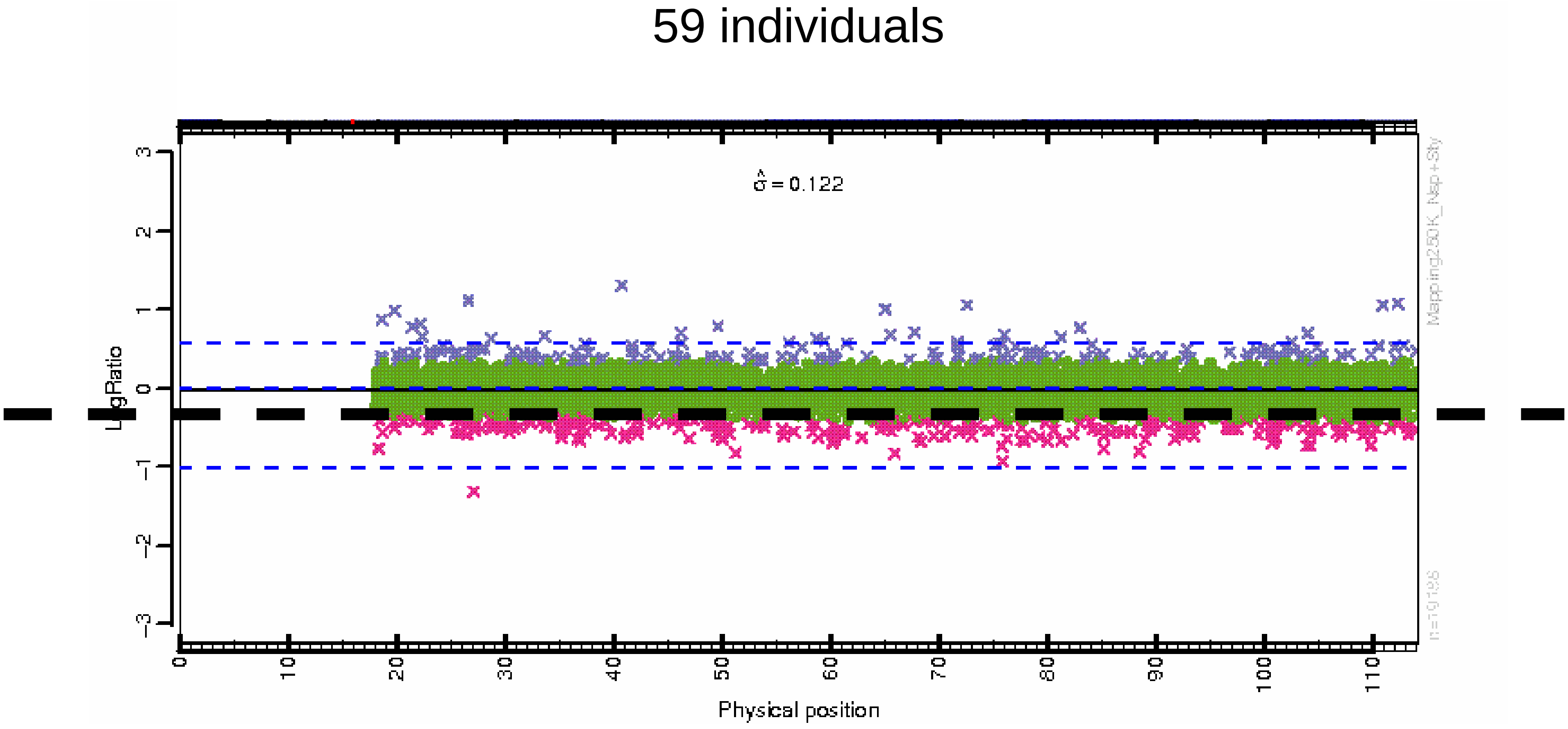
(correctly calling a males male)



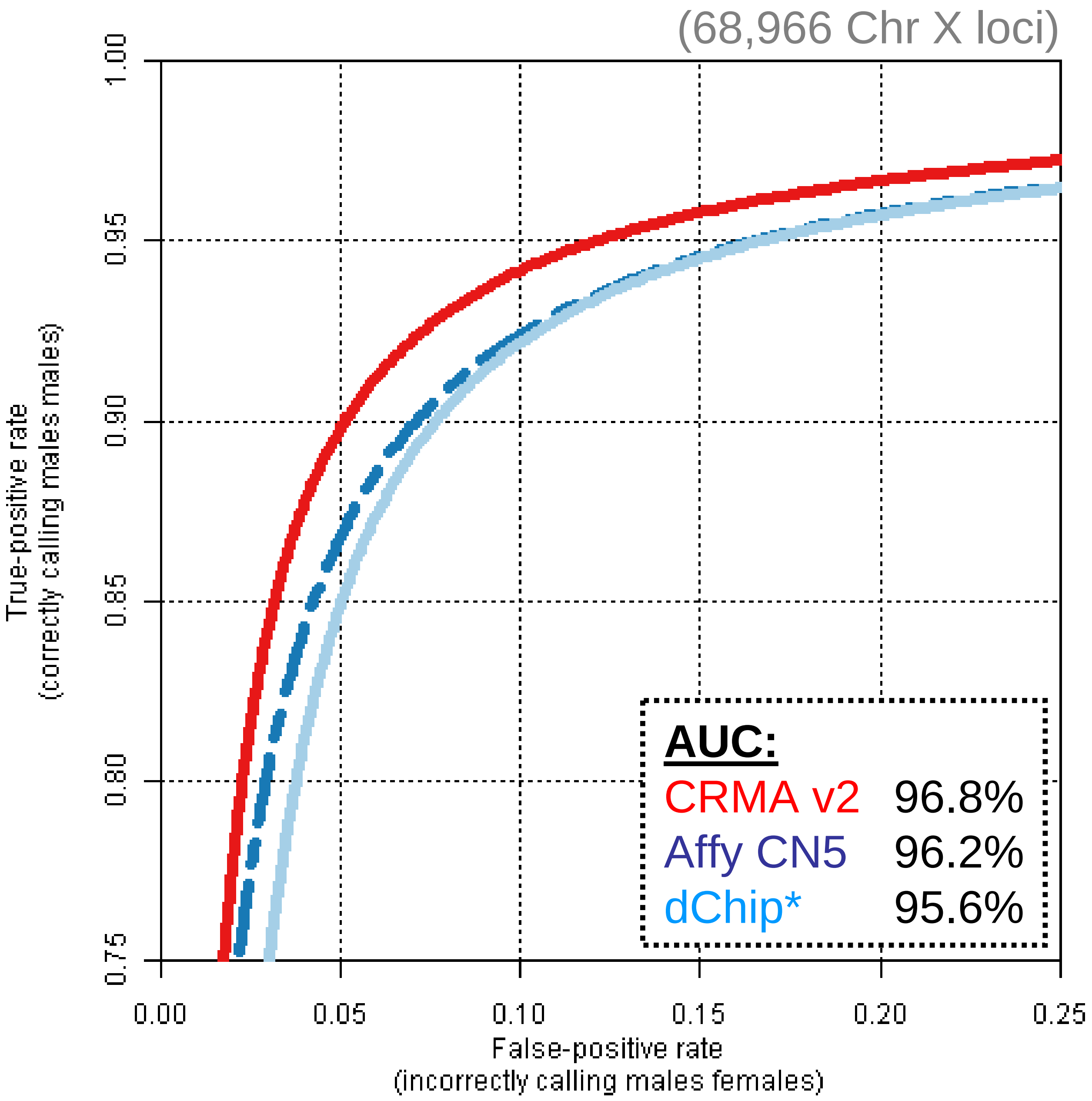
**FP rate**

(incorrectly calling females male)

# Performance of an average SNP with a common threshold



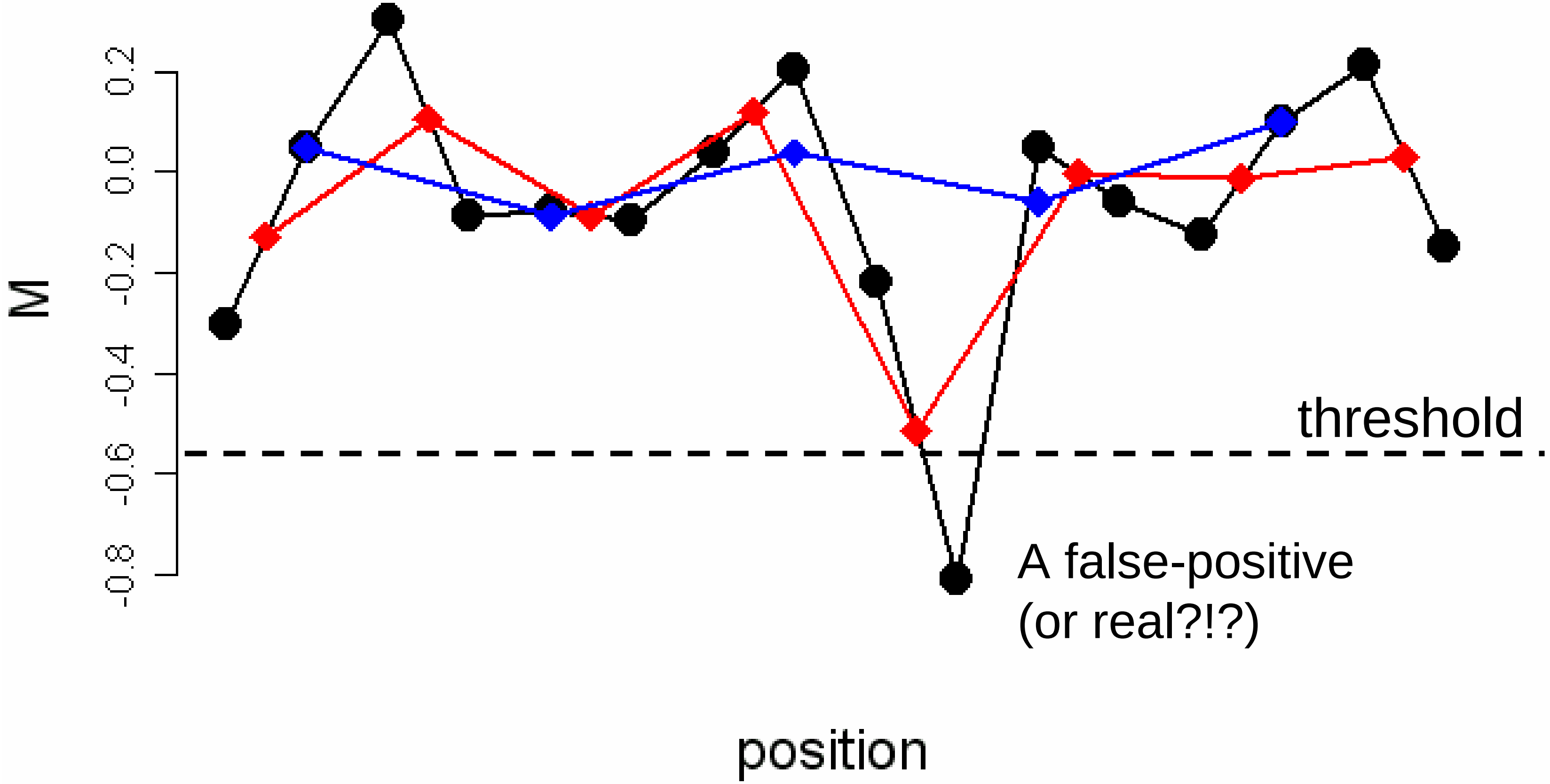
# Better detection of CN=1 among CN=2 using CRMA v2



# Comparing at different resolutions

# Average across SNPs
## *non-overlapping windows*

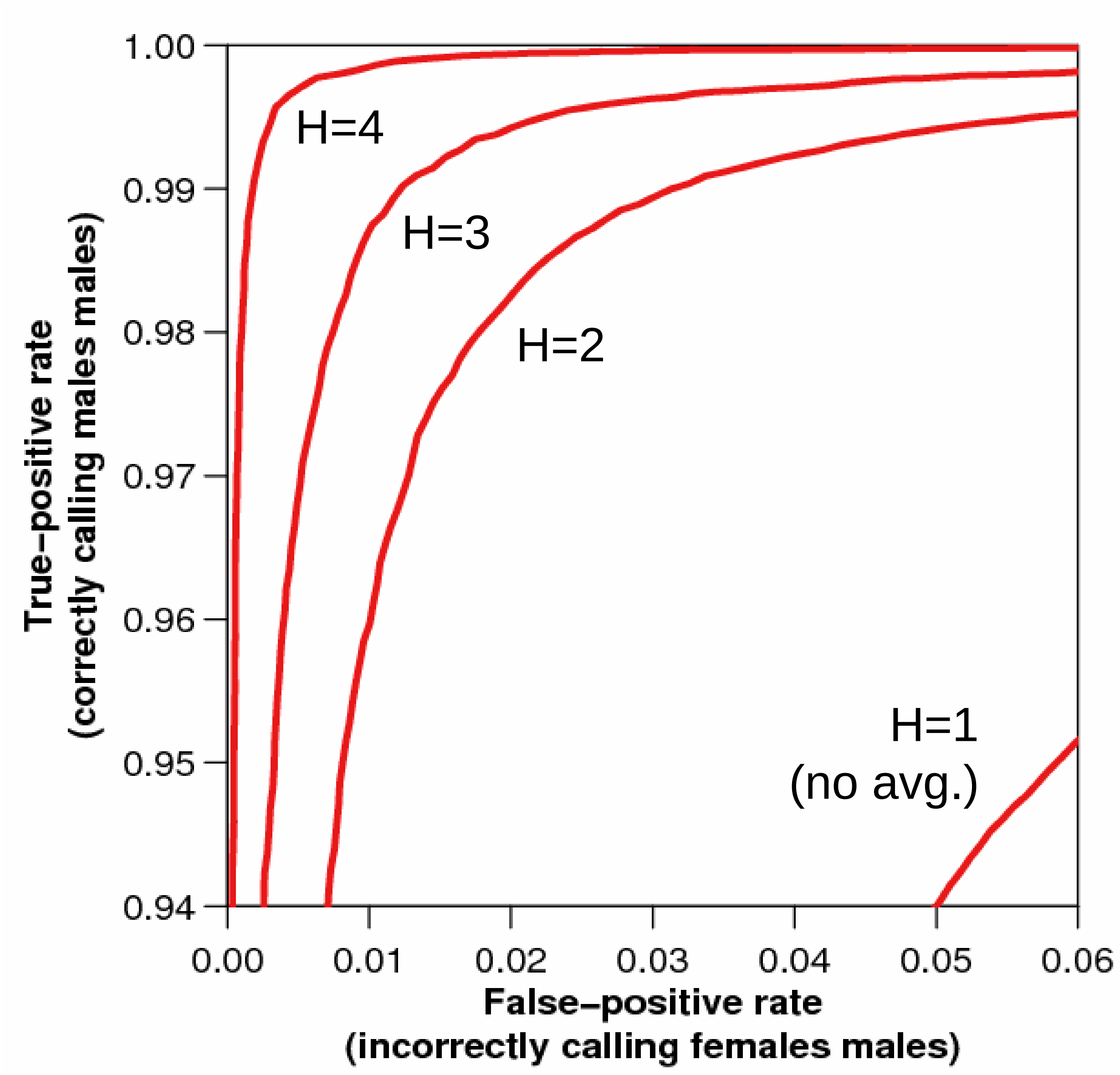
Averaging three and three (H=3)



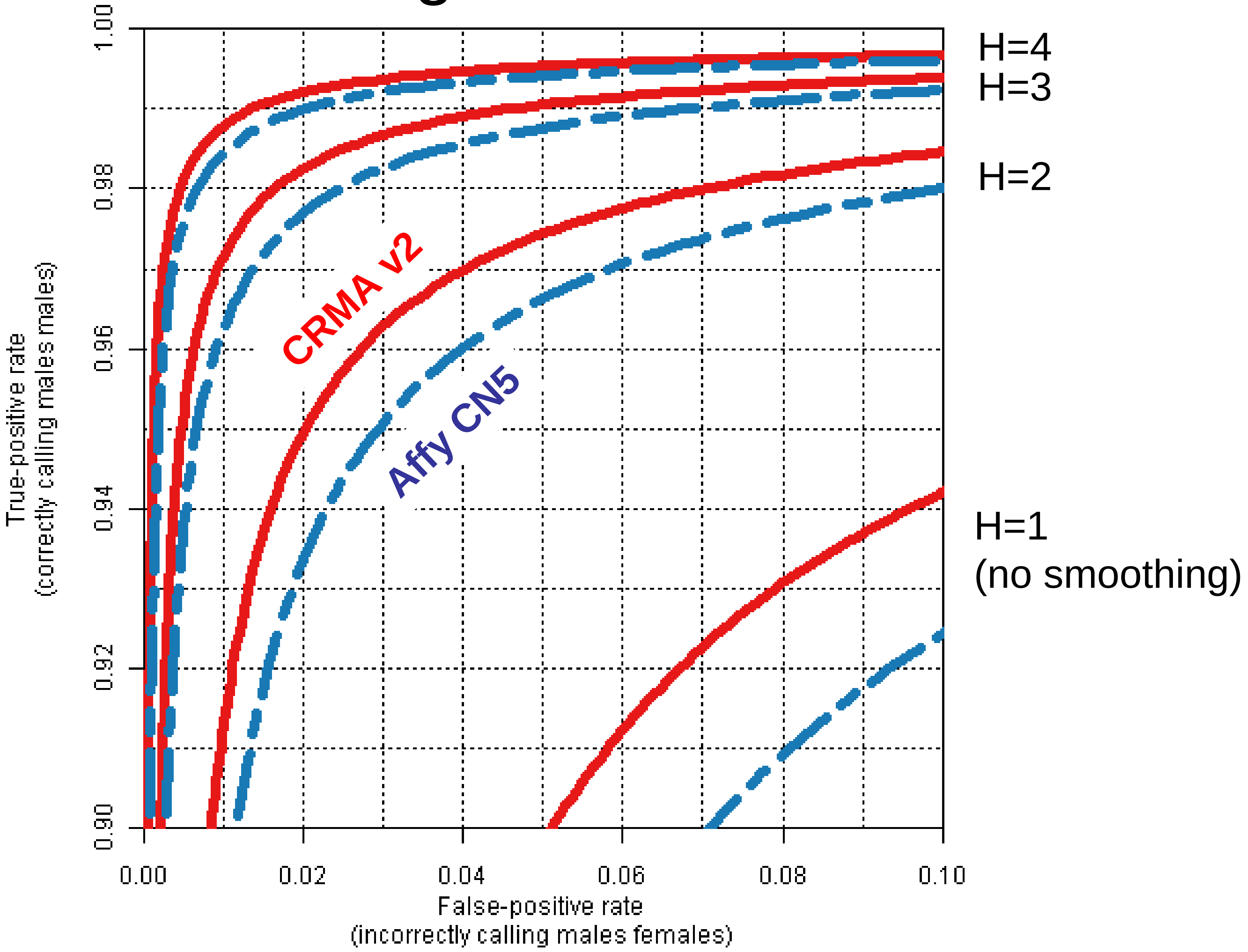
threshold

A false-positive
(or real?!?)

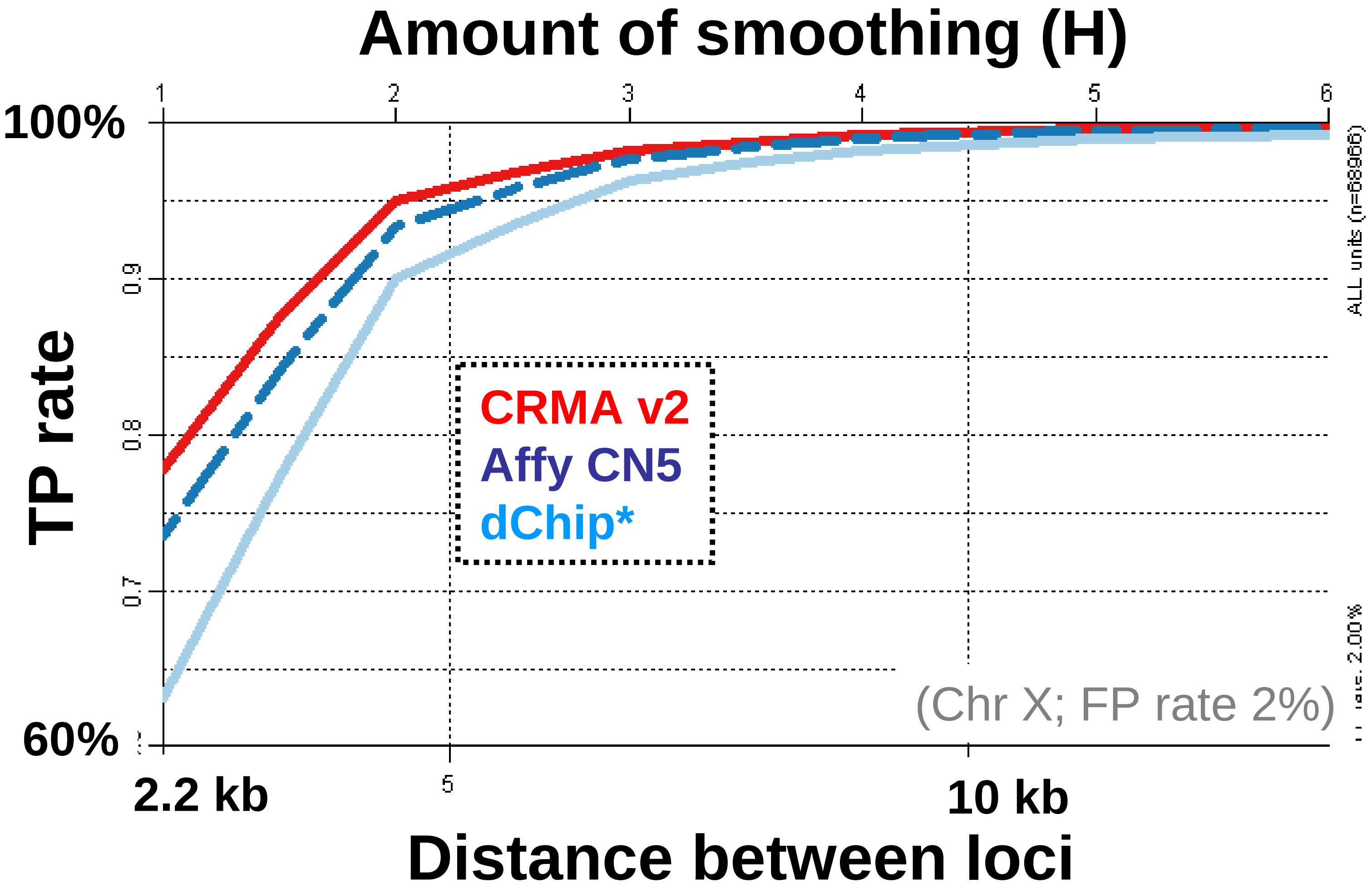
# Better detection rate when averaging *(with risk of missing short regions)*



# CRMA v2 does better also when smoothing



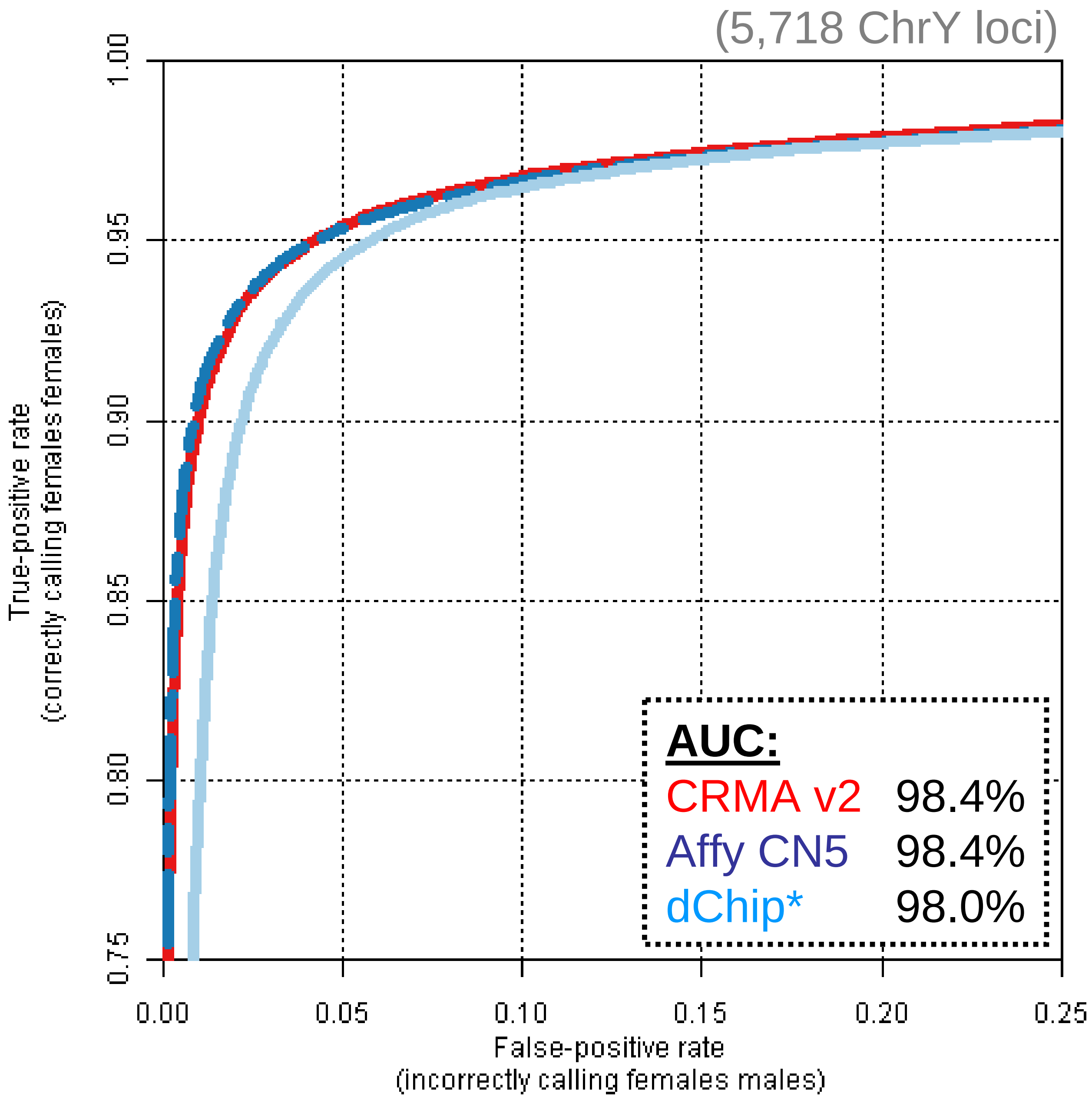
# CRMA v2 detects CN=1 among CN=2 better than other at all resolutions



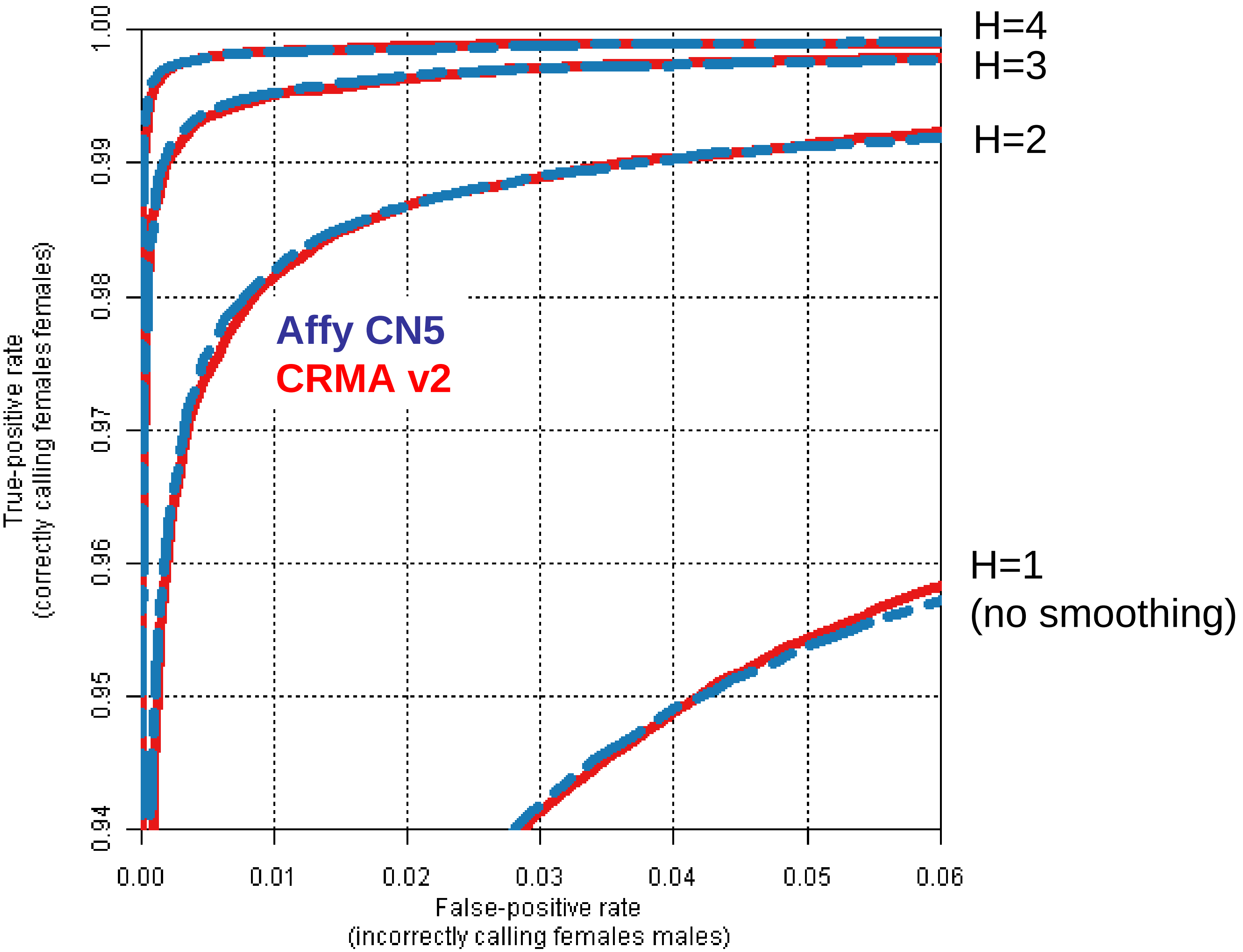
# Performance on ChrY

It is easier to detect
CN=0 among CN=1 (ChrY), than
CN=1 among CN=2 (ChrX).

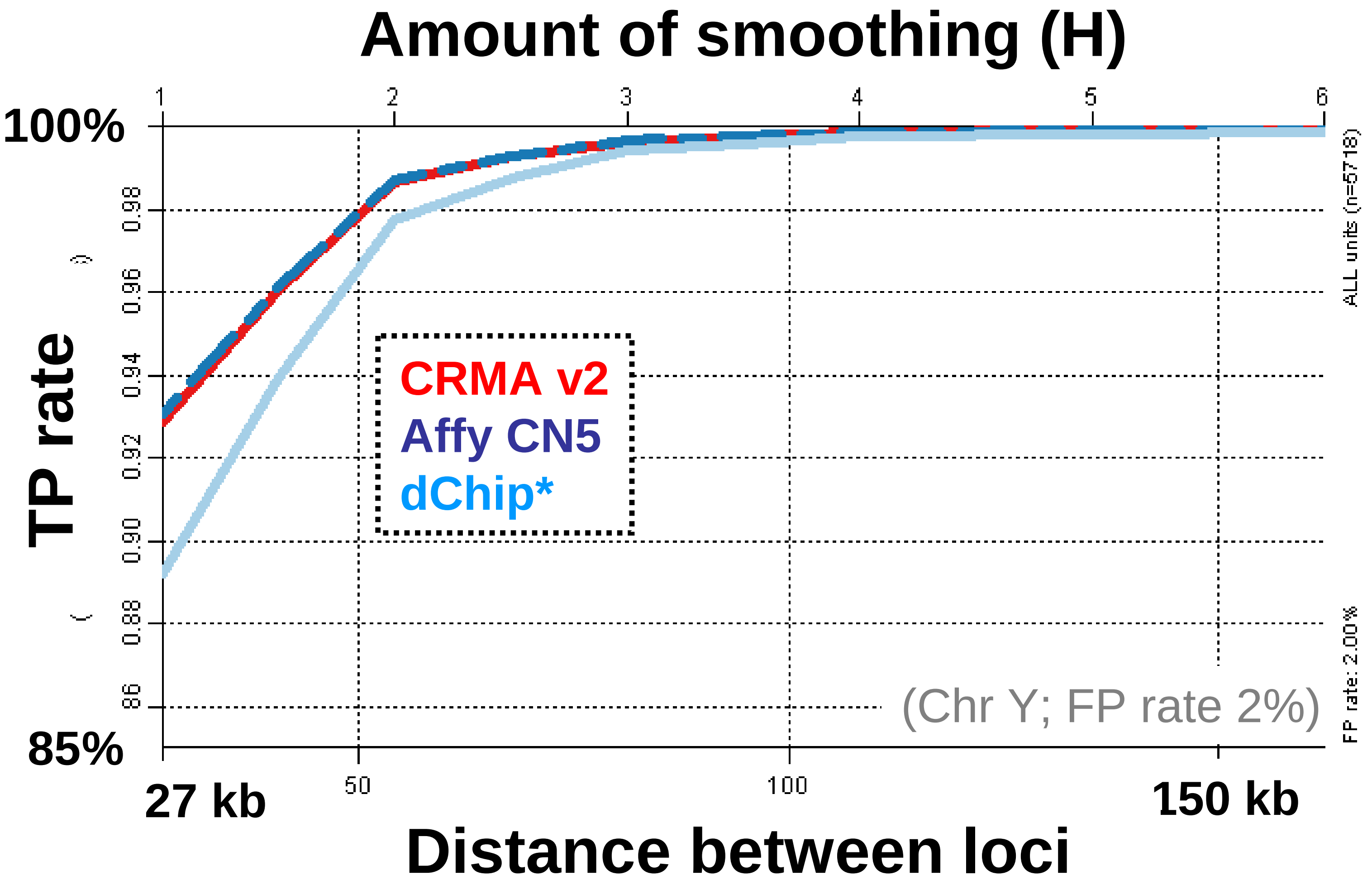
# Better detection of CN=0 among CN=1 using CRMA v2/CN5



# Similar also when smoothing

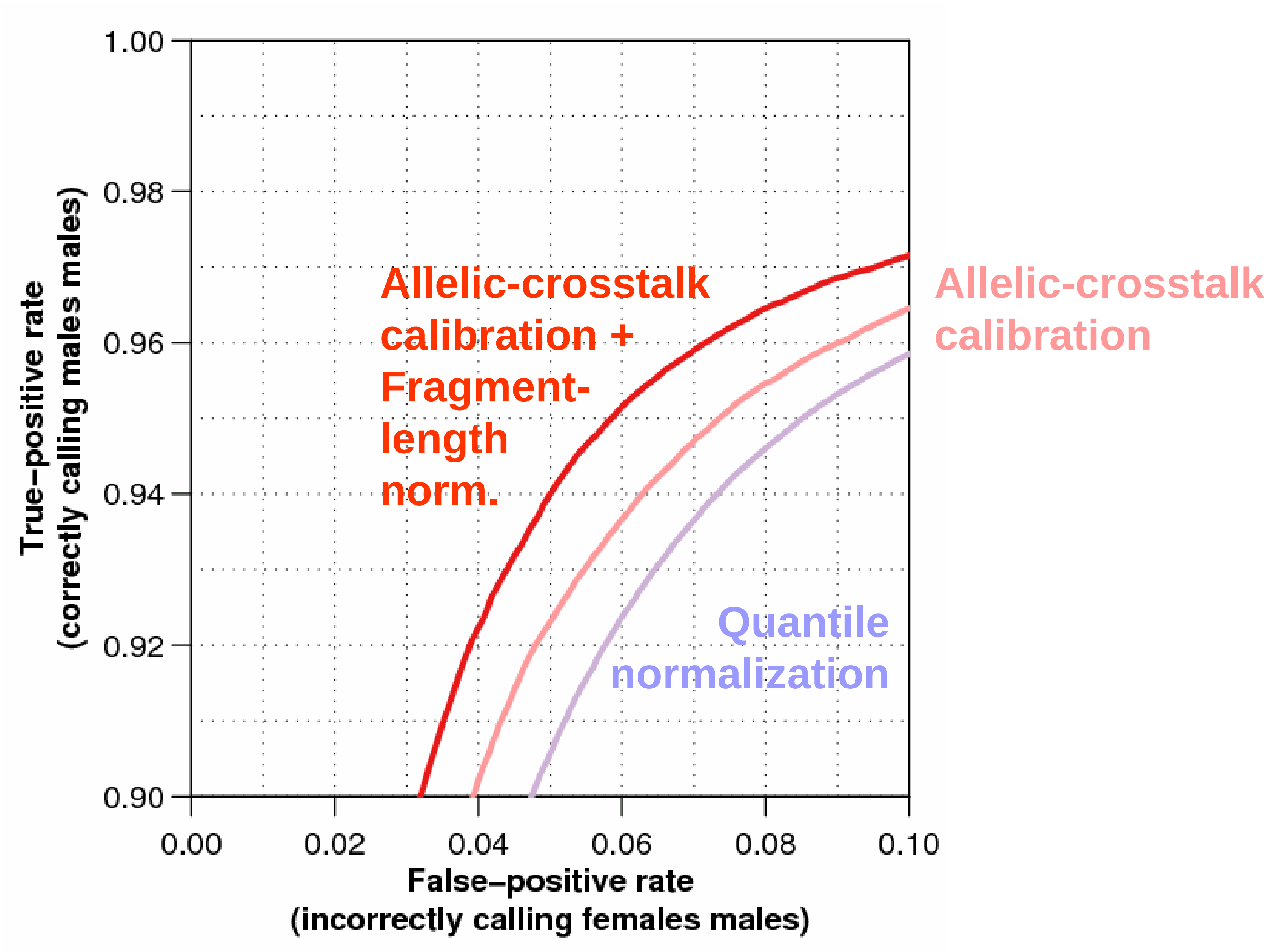


# CRMA v2 & CN5 detects CN=0 among CN=1 equally well at different resolutions

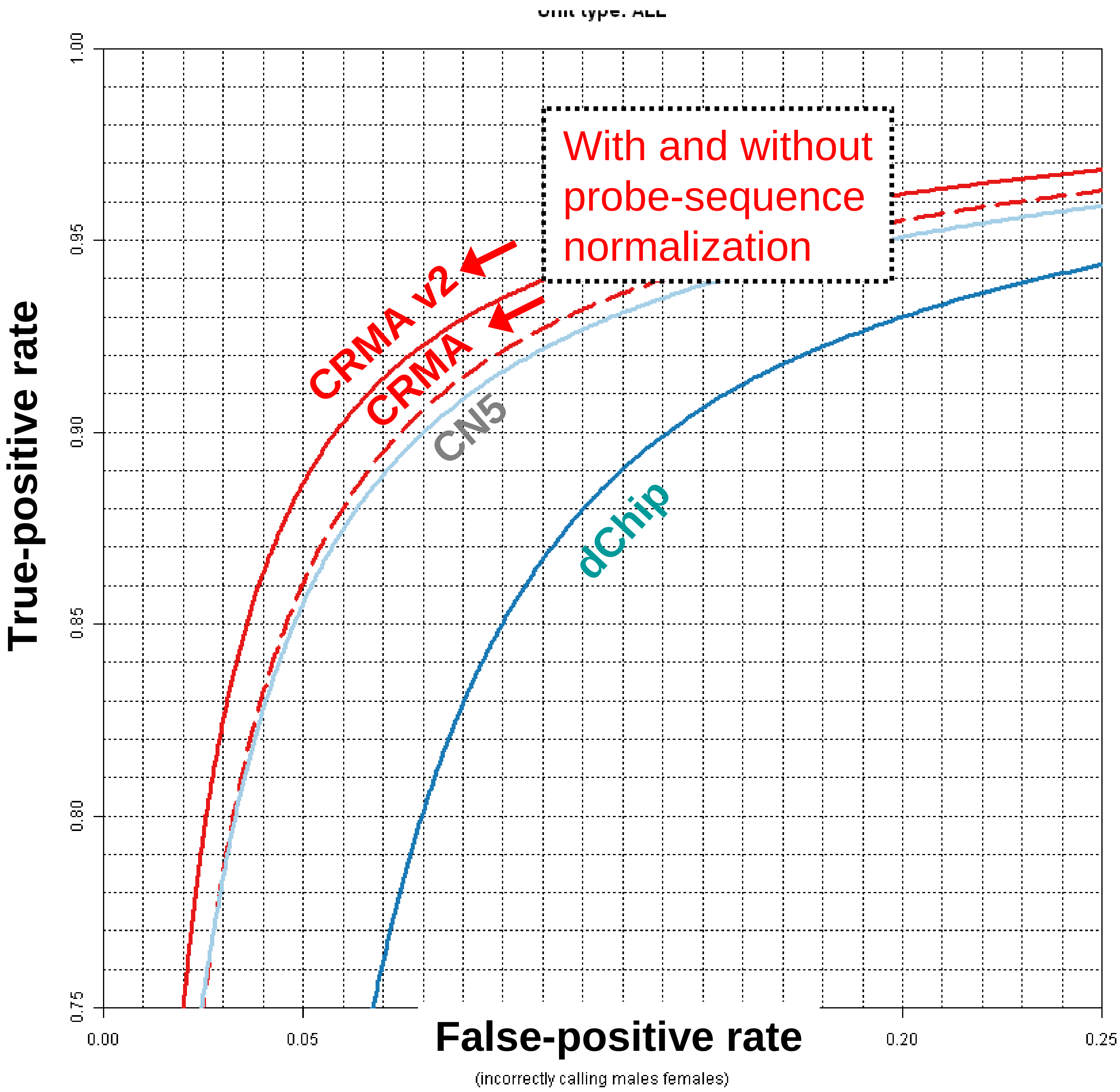


# A final revisit of the pre-processing steps

# Allelic-crosstalk calibration and PCR fragment length normalization improves the detection rate



# Nucleotide-position normalization really helps



With and without probe-sequence normalization

# Conclusions

# Pre-processing helps

- Allelic crosstalk calibration corrects for offset and provides better separation between genotype groups.
- Nucleotide-position normalization corrects for variation across arrays but also heterozygote imbalances.
- PCR fragment-length normalization remove additional variation.
- Using a in-house reference is better than an external one.

# Reason for using CRMA v2

- CRMA v2 can differentiate CN=1 from CN=2 better than other methods.
- CRMA v2 & Affymetrix CN5 differentiate CN=0 from CN=1 equally well.
- CRMA v2 applies to all Affymetrix chip types.
- CRMA v2 is a single-array estimator.
- CRMA v2 can be applied immediately after scanning the array.
- There might be a CRMA v3 later ;)

# Appendix