Publication Supplement

Institute of Bioinformatics, Johannes Kepler University Linz

cn.MOPS: mixture of Poissons for discovering copy number variations in next generation sequencing data with a low false discovery rate — *Supplementary Information* —

Günter Klambauer, Karin Schwarzbauer, Andreas Mayr, Djork-Arné Clevert, Andreas Mitterecker, Ulrich Bodenhofer, and Sepp Hochreiter

Institute of Bioinformatics, Johannes Kepler University, Linz, Austria

Institute of Bioinformatics
Johannes Kepler University Linz
A-4040 Linz, Austria

Tel. +43 732 2468 8880
Fax +43 732 2468 9511
http://www.bioinf.jku.at

Contents

List of Figures

List of Tables

S1 Introduction

This report gives supplementary information to the manuscript “cn.MOPS: Mixture of Poissons for Discovering Copy Number Variations in Next Generation Sequencing Data”.

The supplementary information contain

- derivative of the algorithm, that is of the update rules of the EM algorithm and of the different I/NI calls,
- numerical tests of the approximations — especially of the signed I/NI call by the expected fold change,
- test for Poisson distribution with and without normalization of read counts,
- parameter settings for the compared CNV detection methods in the experiments,
- comparison of the likelihoods for copy number 2 if the true copy number is 3 or 1,
- additional experiments,
- additional information on the data used in the experiments.
- variants of the noise model of cn.MOPS
- investigations about the influence of the hyperparameter ϵ
- exemplary DNA locations with CNV calls of different methods

S2 The Mixture of Poissons Model

Summary. To avoid the false discoveries induced by read count variations along the chromosome or across samples, we propose a “Mixture Of PoissonS model for CNV detection” (cn.MOPS). The cn.MOPS model is not affected by read count variations along the chromosome, because at each DNA position a local model is constructed. Read count variations across samples are decomposed by the cn.MOPS model into integer copy numbers and noise by its mixture components and Poisson distributions, respectively. In contrast to existing methods, cn.MOPS model’s posterior provides integer copy numbers together with their uncertainty. Model selection in a Bayesian framework is based on maximizing the posterior given the samples by an expectation maximization (EM) algorithm. The model incorporates the linear dependency between average read counts in a DNA segment and its copy number. Most importantly, a Dirichlet prior on the mixture components prefers constant copy number 2 for all samples. The more the data drives the posterior away from the Dirichlet prior corresponding to copy number two, the more likely the data is caused by a CNV, and, the higher is the informative/non-informative (I/NI) call. cn.MOPS detects a CNV in the DNA of an individual as a segment with high I/NI calls. I/NI call based CNV detection guarantees a low false discovery rate (FDR) because false detections are less likely for high I/NI calls.

We assume that the genome is partitioned into segments in which reads are counted but which need not be of constant length throughout the genome. For each of such an segment we build a model. We consider the read counts x at a certain segment of the genome, for which we construct a model across samples. The model incorporates both read count variations due to technical or biological noise and variations stemming from copy number variations.

S2.1 The Model

In this Subsection we introduce the cn.MOPS model, which models the read counts of the samples at a certain chromosome segment by copy numbers and noise due to technical or DNA variations.

S2.1.1 The Mixture of Poissons

The cn.MOPS model assumes that the read counts x for a certain copy number i are distributed across samples according to a Poisson. Assuming different copy numbers across samples, the cn.MOPS model is a mixture of Poissons:

$$p(x) \;=\; \sum_{i=0}^{n} \alpha_i \, \mathrm{P}(x; \tfrac{i}{2}\lambda) \; . \tag{S1}$$

In model Eq. (S1) α_i are the percentages of samples with copy numbers $0 \leq i \leq n$ and λ is the mean as well as the variance of read counts for copy number 2, where n is the number of different copy numbers. For copy number i, the Poisson parameter is $\frac{i}{2}\lambda$, by which we assume that the read counts are linearly related to the number of copies. P is the density of the Poisson distribution:

$$\mathrm{P}(x;\beta) \;=\; \frac{1}{x!} \, e^{-\beta} \, \beta^{x} \; . \tag{S2}$$

For notational convenience, we did not distinguish between $i = 0$ and $i \geq 1$ in the above formula. For copy number $i = 0$, we assume a Poisson distribution with parameter $\beta = \frac{\epsilon}{2}\lambda$ which accounts for background noise stemming from wrongly or ambiguously mapped reads as well as for sample contamination by other DNA.

S2.1.2 Estimation of Integer Copy Numbers

The model Eq. (S1) allows for estimating integer copy numbers with fixed model parameters α_i and λ. The prior probability that a read count stems from copy number i is $p(i) = \alpha_i$. The likelihood that a read count x is produced by the i-th mixture component is $p(x \mid i) = \mathrm{P}(x; \frac{i}{2}\lambda)$. Then Bayes' formula can be used to compute the posterior $p(i \mid x)$, that is, the probability that read count x stems from the i-th component corresponding to copy number i. We estimate the copy number by the component that has the largest posterior probability. In Subsection S3.4.3 we found that 99.383%(± 0.001%) of the integer copy numbers were correctly assigned using our posterior integer copy number estimate.

S2.1.3 The Poisson assumption is justified after normalization

The latent variable model of cn.MOPS assumes Poisson distributed read counts across samples for a segment with a constant copy number. This assumption is only justified if sample normalization is applied. Sample normalization corrects the read counts of one sample by the number of mappable reads of the sample. We tested segments of constant size (25kbp) for being Poisson distributed with and without sample normalization. The data is from the Sanger sequencing center on HapMap phase 1 individuals (see Subsection S3.5). Without sample normalization the Poisson assumption was rejected for 92% of the genomic segments. With sample normalization the rejection rate has dropped to 2%. It is plausible, that two percent of segments were rejected by the Poisson test, if the occurrence of known CNV regions is considered. Table S1 shows the contingency table of segments within known CNV regions vs. non-CNV segments and Poisson vs. non-Poisson segments as determined by a test for Poisson suggested by Brown and Zhao (2002). Segments that were rejected by the Poisson test with sample normalization coincide significantly (p-value 2.2e-16) with segments within known CNV regions.

Table S1: Contingency table of segments within known CNV regions vs. non-CNV segments and Poisson vs. non-Poisson segments. The table gives counts of non-CNV segments (first column) and segments within known CNV regions (second column) and counts of segments not rejected by a Poisson test suggested by Brown and Zhao (2002) (first row) and segments rejected by the test (second row). Fisher's exact test for coincidence of non-Poisson segments with segments within known CNV regions is highly significant with a p-value of 2.2e-16. Thus, segments within known CNV regions coincide with segments which are not Poisson distributed.

Poisson assumption/segments	non-CNV	within known CNV region	sum
not rejected	111,876	145	112,021
rejected	2,573	123	2,697
sum	114,449	268	114,717

S2.2 Model Selection: EM Algorithm

In a Bayes framework for model selection, $\boldsymbol{\alpha}$ and λ are considered as random variables, thus, $p(x)$ in Eq. (S1) becomes a conditional probability $p(x \mid \boldsymbol{\alpha}, \lambda)$, i.e. the likelihood that read count x has been produced by the model with parameters $\boldsymbol{\alpha}$ and λ. The EM algorithm minimizes an upper bound on the negative log-posterior of the parameters. The parameter posterior of $\boldsymbol{\alpha}$ and λ is given by:

$$p(\boldsymbol{\alpha}, \lambda \mid x) \;=\; \frac{p(x \mid \boldsymbol{\alpha}, \lambda)\; p(\boldsymbol{\alpha})\; p(\lambda)}{\int p(x \mid \boldsymbol{\alpha}, \lambda)\; p(\boldsymbol{\alpha})\; p(\lambda)\; d\boldsymbol{\alpha}\; d\lambda}\,, \tag{S3}$$

where we assumed that the priors on $\boldsymbol{\alpha}$ and λ are independent of each other. This independence is justified because the copy number distribution $\boldsymbol{\alpha}$ on the samples (determined by the cohort which is investigated) is independent of the expected read count λ for copy number 2 (determined by DNA and biotechnological characteristics). We now introduce priors on $\boldsymbol{\alpha}$ and λ. The parameter posterior $p(\boldsymbol{\alpha}, \lambda \mid x)$ should not be confused with the posterior of latent variable $p(i \mid x)$, the probability that x has been drawn from the i-th mixture component after having observed x.

S2.2.1 Dirichlet Prior on Alpha

In the cn.MOPS model, the prior $p(\boldsymbol{\alpha})$ on $\boldsymbol{\alpha}$ should reflect the fact that predominantly locations with copy number 2 for all samples are present in the data set. Thus, the prior represents the null hypothesis that at a location the copy number is the same across samples, i.e. no sample has a CNV. The Dirichlet prior is well suited to express our prior assumptions on $\boldsymbol{\alpha}$. The Dirichlet prior with parameters $\boldsymbol{\gamma}$ is:

$$p(\boldsymbol{\alpha}) \;=\; \mathrm{D}(\boldsymbol{\alpha}^1;\boldsymbol{\gamma}) \;=\; b(\boldsymbol{\gamma}) \prod_{i=0}^{n} \alpha_i^{\gamma_i-1} \;, \tag{S4}$$

where $\boldsymbol{\alpha}^1$ is the n-dimensional vector $(\alpha_1,\ldots,\alpha_n)$ while α_0 is obtained via $\alpha_0 = 1 - \sum_{i=1}^{n} \alpha_i$. Each component α_i is distributed according to a beta distribution with mean

$$\mathrm{mean}(\alpha_i) \;=\; \frac{\gamma_i}{\gamma_s} \;, \tag{S5}$$

mode

$$\mathrm{mode}(\alpha_i) \;=\; \frac{\gamma_i - 1}{\gamma_s - n} \;, \tag{S6}$$

and variance

$$\mathrm{var}(\alpha_i) \;=\; \frac{\gamma_i\;(\gamma_s - \gamma_i)}{\gamma_s^2\;(\gamma_s + 1)} \;, \tag{S7}$$

where we set

$$\gamma_s \;=\; \sum_{i=0}^{n} \gamma_i \;. \tag{S8}$$

To express our prior knowledge that predominantly locations with copy number 2 for all samples are present, we set $\gamma_2 \gg \gamma_i$ for $i \neq 2$.

S2.2.2 Uniform Prior on λ

For the prior on λ we use an uniform distribution on a sufficiently large interval $(0, 1/t]$ with left endpoint 0 and right endpoint $1/t$. Thus, the density in $(0, 1/t]$ is

$$p(\lambda) \;=\; t \;. \tag{S9}$$

S2.2.3 Upper Bound on the Negative Log Posterior

According to Eq. (S3), the posterior of the model parameters is

$$\begin{aligned} p(\boldsymbol{\alpha}, \lambda \mid x) \;&=\; \frac{p(x \mid \boldsymbol{\alpha}, \lambda)\; p(\boldsymbol{\alpha})\; p(\lambda)}{\int p(x \mid \boldsymbol{\alpha}, \lambda)\; p(\boldsymbol{\alpha})\; p(\lambda)\; d\boldsymbol{\alpha}\; d\lambda} \\ &=\; \frac{p(x \mid \boldsymbol{\alpha}, \lambda)\; p(\boldsymbol{\alpha})}{\int p(x \mid \boldsymbol{\alpha}, \lambda)\; p(\boldsymbol{\alpha})\; d\boldsymbol{\alpha}\; d\lambda} \\ &=\; \frac{1}{c(x)}\; p(x \mid \boldsymbol{\alpha}, \lambda)\; p(\boldsymbol{\alpha}) \;, \end{aligned} \tag{S10}$$

where $c(x)$ is independent of the parameters $\boldsymbol{\alpha}$ and λ.

Deriving the upper bound. For deriving an upper bound on the log posterior needed by the EM algorithm, we deduce the following inequality for one sample x by introducing variables $\hat{\alpha}_i$ with $\sum_{i=1}^{n} \hat{\alpha}_i = 1$:

$$\begin{aligned} -\ \log p(\boldsymbol{\alpha}, \lambda \mid x) \ &= \ -\ \log \left(p(x \mid \boldsymbol{\alpha}, \lambda) \, p(\boldsymbol{\alpha}) \; / \; c(x) \right) \\ &= \ -\ \log \sum_{i=0}^{n} \alpha_i \ \mathrm{P}(x; \frac{i}{2} \lambda) \ - \ \log p(\boldsymbol{\alpha}) \ + \ \log(c(x)) \\ &= \ -\ \log \sum_{i=0}^{n} \frac{\hat{\alpha}_i}{\hat{\alpha}_i} \ \alpha_i \ \mathrm{P}(x; \frac{i}{2} \lambda) \ - \ \log p(\boldsymbol{\alpha}) \ + \ \log(c(x)) \\ &\leq \ -\ \sum_{i=0}^{n} \hat{\alpha}_i \ \log \frac{\alpha_i \ \mathrm{P}(x; \frac{i}{2} \lambda)}{\hat{\alpha}_i} \ - \ \log p(\boldsymbol{\alpha}) \ + \ \log(c(x)) \\ &= \ -\ \sum_{i=0}^{n} \hat{\alpha}_i \ \log \left(\alpha_i \ \mathrm{P}(x; \frac{i}{2} \lambda) \right) \ - \ \log p(\boldsymbol{\alpha}) \\ &\quad + \ \sum_{i=0}^{n} \hat{\alpha}_i \ \log \hat{\alpha}_i \ + \ \log(c(x)) \ , \end{aligned} \tag{S11}$$

where we applied Jensen's inequality. Note that $c(x)$ is independent of $\boldsymbol{\alpha}$ and that for

$$\hat{\alpha}_i \ = \ p(i \mid x, \boldsymbol{\alpha}, \lambda) \ = \ \frac{\alpha_i \ \mathrm{P}(x; \frac{i}{2} \lambda)}{p(x \mid \boldsymbol{\alpha}, \lambda)} \tag{S12}$$

we have in the fifth line of Eq. (S11)

$$\log \frac{\alpha_i \ \mathrm{P}(x; \frac{i}{2} \lambda)}{\hat{\alpha}_i} \ = \ \log p(x \mid \boldsymbol{\alpha}, \lambda) \ , \tag{S13}$$

thus the inequality Eq. (S11) becomes an equality.

The data set. We assume that the data set $\{x_1, \ldots, x_N\}$ of the read counts across the samples is given, where the read count from the k-th sample is denoted by x_k. Model selection and therefore the EM algorithm is based on these samples. The posterior that x_k is drawn from the i-th mixture component is

$$\alpha_{ik} \ = \ p(i \mid x_k, \boldsymbol{\alpha}, \lambda) \ = \ \frac{p(i) \ p(x_k \mid i, \boldsymbol{\alpha}, \lambda)}{p(x_k \mid \boldsymbol{\alpha}, \lambda)} \ = \ \frac{\alpha_i \ \mathrm{P}(x_k; \frac{i}{2} \lambda)}{p(x_k \mid \boldsymbol{\alpha}, \lambda)} \ , \tag{S14}$$

where α_i is the prior of being drawn from the i-th mixture component.

S2.2.4 E-step

In analogy to the $\hat{\alpha}_i$ in Subsection S2.2.3, we introduce for each x_k variables $\hat{\alpha}_i k$ with $\sum_{i=1}^{n} \hat{\alpha}_i k = 1$ which estimate $p(i \mid \boldsymbol{\alpha}, x_k, \lambda)$ (see Eq. (S12)), are is formally independent of the parameters $\boldsymbol{\alpha}$ and λ. For the E-step of the EM algorithm, we estimate the posterior α_{ik} by

$$\hat{\alpha}_{ik} \ = \ \frac{\alpha_i^{\mathrm{old}} \ \mathrm{P}(x_k; \frac{i}{2} \lambda^{\mathrm{old}})}{p(x_k; \boldsymbol{\alpha}^{\mathrm{old}}, \lambda^{\mathrm{old}})} \ , \tag{S15}$$

where for the estimation the actual parameters $\boldsymbol{\alpha}^{\text{old}}$ and λ^{old} are used instead of the optimal parameters $\boldsymbol{\alpha}$ and λ in the expression for the posterior in Eq. (S14).

Based on inequality Eq. (S11) but with $\hat{\alpha}_{ik}$ instead of $\hat{\alpha}_i$, we define an upper bound B on the $\frac{1}{N}$ scaled negative log-posterior as

$$\begin{aligned} B \;=\; & -\,\frac{1}{N}\,\sum_{k=1}^{N}\,\sum_{i=0}^{n}\hat{\alpha}_{ik}\,\log\left(\alpha_i\,\mathrm{P}(x;\frac{i}{2}\lambda)\right)\;-\;\frac{1}{N}\,\log p(\boldsymbol{\alpha}) \\ & +\,\frac{1}{N}\,\sum_{k=1}^{N}\,\sum_{i=0}^{n}\hat{\alpha}_{ik}\,\log\hat{\alpha}_{ik}\;+\;\frac{1}{N}\,\sum_{k=1}^{N}\,\log c(x_k)\;, \end{aligned} \tag{S16}$$

where we summed over all terms depending on x_k. Note, that according to Eq. (S12) and Eq. (S13) an exact estimate in the E-step Eq. (S15) (using the optimal parameters $\boldsymbol{\alpha}$ and λ) make inequality Eq. (S11) to an equality, thus the upper bound B would be equal to the negative log posterior. For notational convenience "$\frac{0}{2}\lambda$" stands for $\frac{\epsilon}{2}\lambda$ according to the model defined in Eq. (S1).

S2.2.5 M-step: Alpha Optimization

In the M-step, we minimize the upper bound B on the negative log posterior with respect to α under the constraint that the α_i sum to 1. Only terms depending on $\boldsymbol{\alpha}$ are considered:

$$\begin{aligned} & \min_{\boldsymbol{\alpha}}\;\; -\,\frac{1}{N}\,\sum_{k=1}^{N}\sum_{i=0}^{n}\hat{\alpha}_{ik}\log\alpha_i\;-\;\frac{1}{N}\;\log p(\boldsymbol{\alpha}) \\ & \text{s.t.}\;\;\sum_{i=0}^{n}\alpha_i\;=\;1\;. \end{aligned} \tag{S17}$$

The Lagrangian with Lagrange parameter ρ is

$$\begin{aligned} L \;=\; & -\,\frac{1}{N}\,\sum_{k=1}^{N}\sum_{i=0}^{n}\hat{\alpha}_{ik}\log\alpha_i\;-\;\frac{1}{N}\;\log p(\boldsymbol{\alpha}) \\ & +\,\rho\,\left(\sum_{i=0}^{n}\alpha_i\;-\;1\right) \\ =\; & -\,\frac{1}{N}\,\sum_{k=1}^{N}\sum_{i=0}^{n}\hat{\alpha}_{ik}\;\log\alpha_i\;-\;\frac{1}{N}\,\sum_{i=0}^{n}(\gamma_i\;-\;1)\;\log\alpha_i \\ & +\,\rho\,\left(\sum_{i=0}^{n}\alpha_i\;-\;1\right)\;. \end{aligned} \tag{S18}$$

The solution requires that, the derivative of L with respect to α_i is zero:

$$\frac{\partial L}{\partial\alpha_i}\;=\;-\,\frac{1}{N}\,\sum_{k=1}^{N}\hat{\alpha}_{ik}\frac{1}{\alpha_i}\;-\;\frac{1}{N}\,\frac{1}{\alpha_i}\,(\gamma_i\;-\;1)\;+\;\rho\;=\;0\;. \tag{S19}$$

Multiplying this equation by α_i gives

$$-\frac{1}{N}\sum_{k=1}^{N}\hat{\alpha}_{ik} - \frac{1}{N}(\gamma_i - 1) + \rho\,\alpha_i = 0\,. \tag{S20}$$

Summation over i leads to

$$1 + \frac{1}{N}(\gamma_s - n) = \rho\,. \tag{S21}$$

Inserting this expression for ρ in Eq. (S20) results in

$$-\frac{1}{N}\sum_{k=1}^{N}\hat{\alpha}_{ik} - \frac{1}{N}(\gamma_i - 1) + \left(1 + \frac{1}{N}(\gamma_s - n)\right)\alpha_i = 0\,. \tag{S22}$$

Solving Eq. (S22) for α_i gives the update rule for α_i:

$$\alpha_i^{\mathrm{new}} = \frac{\hat{\alpha}_i + \frac{1}{N}(\gamma_i - 1)}{1 + \frac{1}{N}(\gamma_s - n)}\,, \tag{S23}$$

where we used

$$\hat{\alpha}_i = \frac{1}{N}\sum_{k=1}^{N}\hat{\alpha}_{ik}\,. \tag{S24}$$

We introduced $\hat{\alpha}_i$ which sums up the $\hat{\alpha}_{ik}$ and thereby approximates α_i. This approximation is justified because α_i can be decomposed into α_{ik}:

$$\begin{aligned}\alpha_i &= p(i) = p(i \mid \boldsymbol{\alpha}, \lambda) = \int p(i, x \mid \boldsymbol{\alpha}, \lambda)\, dx \\ &= \int p(i \mid x, \boldsymbol{\alpha}, \lambda)\, p(x \mid \boldsymbol{\alpha}, \lambda)\, dx = \mathrm{E}_{p(x\mid\boldsymbol{\alpha},\lambda)}(p(i \mid x, \boldsymbol{\alpha}, \lambda)) \\ &\approx \frac{1}{N}\sum_{k=1}^{N} p(i \mid x_k, \boldsymbol{\alpha}, \lambda) = \frac{1}{N}\sum_{k=1}^{N}\alpha_{ik}\,.\end{aligned} \tag{S25}$$

S2.2.6 M-Step: Lambda Optimization

In the M-step, B need not only be minimized with respect to $\boldsymbol{\alpha}$ but also with respect to λ (only terms depending on λ are considered):

$$\min_{\lambda}\left(-\frac{1}{N}\sum_{k=1}^{N}\sum_{i=0}^{n}\hat{\alpha}_{ik}\log \mathrm{P}(x; \frac{i}{2}\lambda)\right)\,. \tag{S26}$$

For the minimum, the derivative of the above objective with respect to λ must be zero. Using

$$\log \mathrm{P}(x_k; \frac{i}{2}\lambda) = -\log(x_k!) - \frac{i}{2}\lambda + x_k\left(\log(\lambda) + \log(i/2)\right)\,, \tag{S27}$$

this derivative is

$$- \frac{1}{N} \sum_{k=1}^{N} \sum_{i=0}^{n} \left(- \frac{i}{2} \ + \ x_k \, \frac{1}{\lambda} \right) \, \hat{\alpha}_{ik} \ . \tag{S28}$$

Multiplying Eq. (S28) by λ and solving it for λ gives the update rule:

$$\begin{aligned} \lambda^{\text{new}} \ &= \ \frac{\sum_{k=1}^{N} \sum_{i=0}^{n} x_k \, \hat{\alpha}_{ik}}{\sum_{k=1}^{N} \sum_{i=0}^{n} \frac{i}{2} \, \hat{\alpha}_{ik}} \ = \ \frac{\sum_{k=1}^{N} x_k}{\sum_{k=1}^{N} \sum_{i=0}^{n} \frac{i}{2} \, \hat{\alpha}_{ik}} \\ &= \ \frac{\frac{1}{N} \, \sum_{k=1}^{N} x_k}{\sum_{i=0}^{n} \hat{\alpha}_i \, \frac{i}{2}} \ , \end{aligned} \tag{S29}$$

where according to the model defined in Eq. (S1) for notational convenience $\frac{0}{2}$ stands for $\frac{\epsilon}{2}$.

S2.2.7 Update Rules

The update rules of previous subsections can be summarized as follows:

$$\hat{\alpha}_{ik} \ = \ \frac{\alpha_i^{\text{old}} \, \mathrm{P}(x_k; \frac{i}{2} \lambda^{\text{old}})}{p(x_k \mid \boldsymbol{\alpha}^{\text{old}}, \lambda^{\text{old}})} \ , \tag{S30}$$

$$\alpha_i^{\text{new}} \ = \ \frac{\frac{1}{N} \sum_{k=1}^{N} \hat{\alpha}_{ik} \ + \ \frac{1}{N} (\gamma_i \ - \ 1)}{1 \ + \ \frac{1}{N} (\gamma_s \ - \ n)} \ , \tag{S31}$$

$$\lambda^{\text{new}} \ = \ \frac{\frac{1}{N} \sum_{k=1}^{N} x_k}{\sum_{i=0}^{n} \left(\frac{1}{N} \, \frac{i}{2} \, \sum_{k=1}^{N} \hat{\alpha}_{ik} \right)} \ . \tag{S32}$$

Concerning the EM algorithm the update rule Eq. (S30) is the E-step, the update rule Eq. (S31) is the M-step for $\boldsymbol{\alpha}$, and the update rule Eq. (S31) is the M-step for λ.

The update rule Eq. (S31) can be obtained in an alternative way. The Dirichlet distribution is conjugate to the multinomial distribution, that is the posterior $p(\boldsymbol{\alpha} \mid \{\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_k, \ldots, \boldsymbol{\alpha}_N\})$ is a Dirichlet distribution as is the prior $p(\boldsymbol{\alpha})$ with $\boldsymbol{\alpha}_k \ = \ p(\boldsymbol{\alpha} \mid x_k)$. The Dirichlet prior $p(\boldsymbol{\alpha}) \ = \ \mathrm{D}(\boldsymbol{\alpha}^1; \boldsymbol{\gamma})$ with parameters $\boldsymbol{\gamma}$ leads to the conjugate posterior $p(\boldsymbol{\alpha} \mid \{\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_k, \ldots, \boldsymbol{\alpha}_N\})$ with parameters

$$\hat{\boldsymbol{\gamma}} \ = \ \boldsymbol{\gamma} \ + \ \sum_{k=1}^{N} \boldsymbol{\alpha}_k \ = \ \boldsymbol{\gamma} \ + \ N \, \boldsymbol{\alpha} \ , \tag{S33}$$

where we used Eq. (S25). We obtain update rule Eq. (S31) from Eq. (S33) component-wise by first replacing the unknown values α_{ik} by their estimates $\hat{\alpha}_{ik}$ and then computing the posterior's mode because we search for the maximum posterior.

S2.2.8 Parameter Setting for the Dirichlet Prior in the Update Rule Eq. (S31)

We set the parameter $\boldsymbol{\gamma}$ of the Dirichlet prior $\mathrm{D}(\boldsymbol{\alpha}^1; \boldsymbol{\gamma})$ to $\boldsymbol{\gamma} = (1, 1, 1 + G, 1, 1, ..., 1)$, where $G > 0$ is a hyperparameter that controls the prior's impact during model selection. Note that $\gamma_s - n = I$ and therefore we obtain the mode $\boldsymbol{m} = (0, 0, 1, 0, \ldots, 0)$, since $\mathrm{mode}(\alpha_i) = \frac{\gamma_i - 1}{\gamma_s - n}$. This mode corresponds to our null hypothesis that all samples have copy number 2. The mode is not affected by the choice of the hyperparameter G, however the variance decreases as we increase G:

$$\mathrm{var}(\alpha_i) \ = \ \frac{\gamma_i \ (\gamma_s \ - \ \gamma_i)}{\gamma_s^2 \ (\gamma_s \ + \ 1)} \ , \tag{S34}$$

The hyperparameter G affects the EM algorithm via the update rule and can serve to keep percentage of samples having copy number 2 above a threshold. The smaller the variance, the less likely a deviation from copy number 2. The update rule is

$$\alpha_i^{\mathrm{new}} \ = \ \frac{\frac{1}{N} \sum_{k=1}^{N} \hat{\alpha}_{ik} \ + \ \frac{1}{N} \ (\gamma_i \ - \ 1)}{1 \ + \ \frac{1}{N} (\gamma_s \ - \ n)} \ . \tag{S35}$$

Thus, the estimate for the percentage of samples having copy number 2 cannot fall below $\frac{G}{G+N}$ for $\boldsymbol{\gamma} = (1, 1, 1 + G, 1, 1, ..., 1)$ because

$$\alpha_2^{\mathrm{new}} \geq \frac{\frac{1}{N} \ (\gamma_2 \ - \ 1)}{1 \ + \ \frac{1}{N} (\gamma_s \ - \ n)} \ = \ \frac{G}{G \ + \ N} \ . \tag{S36}$$

In our experiments we ensured that the estimate for the percentage of the samples having copy number 2 is always greater or equal to 50% by setting G to N ($G = N$) which leads to

$$\alpha_2^{\mathrm{new}} \ = \ \frac{\hat{\alpha}_2 + 1}{2} \ \geq \ \frac{1}{2} \ , \tag{S37}$$

$$\alpha_i^{\mathrm{new}} \ = \ \frac{\hat{\alpha}_i}{2} \ \text{ for } i \ \neq \ 2 \ . \tag{S38}$$

S2.2.9 Three Posteriors in Our Framework

In our Bayesian framework we introduced 3 different posterior distributions: (i) in Eq. (S14) the posterior $\alpha_{ik} = p(i \mid x_k, \boldsymbol{\alpha}, \lambda)$ of the data x_k stemming from the i-th component with prior $\alpha_i = p(i)$ — this posterior is defined for fixed model parameters $(\boldsymbol{\alpha}, \lambda)$; (ii) in Eq. (S3) the parameter posterior $p(\boldsymbol{\alpha}, \lambda \mid x)$ with priors $p(\boldsymbol{\alpha})$ and $p(\lambda)$ — this posterior is the objective that we maximize during model selection; (iii) the posterior $p(\boldsymbol{\alpha} \mid \{\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_k, \ldots, \boldsymbol{\alpha}_N\})$ used in Eq. (S33) with prior $p(\boldsymbol{\alpha})$ — this posterior is used for the I/NI call (see Subsection S2.3), but in contrast to (ii) it is not the posterior for the full mixture of Poisson model but only for the multinomial distribution given by $\boldsymbol{\alpha}$ where the posteriors $\alpha_{ik} = p(i \mid x_k, \boldsymbol{\alpha}, \lambda)$ from (i) serve as data. At (i) we consider the fixed parameter mixture model which can be combined with the parameter $\boldsymbol{\alpha}$ multinomial model at (iii) to the full model at (ii) if the posterior on λ analog to the posterior on $\boldsymbol{\alpha}$ at (iii) is included.

S2.3 I/NI Call: Information Gain of Posterior over Prior

Based on cn.MOPS' Bayesian approach to model selection, we define an informative/non-informative (I/NI) call analogous to the I/NI call obtained for the FARMS algorithm which excelled in summarization and gene filtering for microarray data (Hochreiter *et al.* 2006; Talloen *et al.* 2007, 2010).

In contrast to λ, which captures noise variation, $\boldsymbol{\alpha}$ captures variation stemming from CNVs, therefore, its posterior indicates CNVs in the data. The I/NI call measures the information gain of the posterior compared to its prior distribution $p(\boldsymbol{\alpha})$ which represents the null hypothesis that all samples have copy number 2. Therefore, the I/NI call measures the tendency to reject the null hypothesis based on the observed data.

The multidimensional distribution and the Dirichlet distribution are conjugate, therefore the posterior is also a Dirichlet distribution with parameters according to Eq. (S33). The I/NI call measures the information gain of the Dirichlet posterior $p(\boldsymbol{\alpha} \mid \{\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_k, \ldots, \boldsymbol{\alpha}_N\})$ over the Dirichlet prior $p(\boldsymbol{\alpha}) = \mathrm{D}(\boldsymbol{\alpha}^1; \boldsymbol{\gamma})$. The prior $p(\boldsymbol{\alpha})$ represents the null hypothesis that all samples have copy number 2, therefore we set $\boldsymbol{\gamma} = (1, 1, 1 + G, 1, 1, ..., 1)$ which leads to the mode $\boldsymbol{m} = (0, 0, 1, 0, \ldots, 0)$. The I/NI call is the distance between the prior's and the posterior's mode. We assess how much the prior assumption to see only copy number 2 has changed after having observed the data. Note, that we do not consider the variance because it is determined by the hyperparameter G and the number of samples. The difference between the prior's mode and the posterior's mode is component-wise

$$\frac{\gamma_i \ - \ 1}{\gamma_s \ - \ n} - \frac{\gamma_i \ + \ N\,\alpha_i \ - \ 1}{\gamma_s \ + \ N \ - \ n} \ = \ \frac{N}{\gamma_s \ - \ n \ + \ N} \left(\frac{\gamma_i \ - \ 1}{\gamma_s \ - \ n} \ - \ \alpha_i \right) . \tag{S39}$$

The difference for copy number i is difference between the prior's mode $(\gamma_i - 1)/(\gamma_s - n)$ and the estimate α_i of observing copy number i in the data set, where the difference is weighted by a factor.

If we assume a Poisson distribution for read counts of each copy number, then the read count distributions for copy numbers > 4 or 0 have less overlap with the copy number 2 read count distribution than copy number 1, 3, or 4 distributions (see Subsection S3.3.3). Further, the read count distribution for copy number 1 has less overlap with the copy number 2 read count distribution than the copy number 3 distribution as shown in Subsection S3.3.3. Summarizing, the more the copy number differs from 2, the less overlap has its read count distribution with those of copy number 2. Consequently, the more a read count differs from the average copy number 2 read count, the more likely a copy number different from copy number 2 is present.

We incorporate this fact of being more sure on read counts belonging to copy numbers which differ more from 2 into the I/NI call. We weight the difference between the prior's mode $\boldsymbol{m}$ and the posterior's mode per component by its absolute log fold change relative to copy number 2. Thus components 1 and 4, which half and double the read counts of copy number 2 respectively, are weighted equally. With $\boldsymbol{m} = (0, 0, 1, 0, \ldots, 0)$, we define the I/NI call as

$$\mathrm{I/NI}(\boldsymbol{\alpha}) \ = \ \sum_{i=0}^{n} |m_i \ - \ \alpha_i| \, |\log(i/2)| \ = \ \sum_{i=0}^{n} \alpha_i \, |\log(i/2)| \, . \tag{S40}$$

The I/NI call is the expected fold change given the data set. For notational convenience, we did not distinguish between $i = 0$ and $i \geq 1$ in the above formula. "log(0/2)" must be understood as

$\log(\epsilon/2)$ — in accordance with the fact that read counts for copy number 0 are Poisson distributed with parameter $\epsilon\lambda/2$. Note that the I/NI call does not depend on the value of α_2 but it is a distance measure for vectors $\|\boldsymbol{\alpha}\|_1 = 1$ and $\alpha_i \geq 0$ from $\boldsymbol{m}$ as shown in the following.

Let $\boldsymbol{\alpha}_{-2}$ be the vector $\boldsymbol{\alpha}$ where the second component α_2 is removed. We define the distance between two vectors $\boldsymbol{\alpha}^1_{-2}$ and $\boldsymbol{\alpha}^2_{-2}$, where from both the second component is removed, as

$$\left\|\boldsymbol{L}(\boldsymbol{\alpha}^1_{-2} \ - \ \boldsymbol{\alpha}^2_{-2})\right\|_1 \tag{S41}$$

with diagonal matrix $\boldsymbol{L}$ having diagonal elements $L_{ii} = |\log(i/2)|$ for $i > 0, i \neq 2$ and $L_{00} = |\log(\epsilon/2)|$. Eq. (S41) is a valid distance measure between vectors $\boldsymbol{\alpha}^1_{-2}$ with $\boldsymbol{\alpha}^2_{-2}$ because it is the 1-norm of the difference of two vectors after component-wise scaling. For $\|\boldsymbol{\alpha}\|_1 = 1$ and $\alpha_i \geq 0$ the component α_2 can be computed from components $\alpha_i, \ i \neq 2$. Therefore Eq. (S41) is a valid distance measure for $\|\boldsymbol{\alpha}\|_1 = 1$ with $\alpha_i \geq 0$ between vectors $\boldsymbol{\alpha}^1$ and $\boldsymbol{\alpha}^2$. It follows that the I/NI call in Eq. (S40) is a valid distance measure of vectors $\|\boldsymbol{\alpha}\|_1 = 1$ with $\alpha_i \geq 0$ from $\boldsymbol{m}$. Consequently, $\mathrm{I/NI}(\boldsymbol{m}) = 0$ and $\mathrm{I/NI}(\boldsymbol{\alpha}) > 0$ for $\boldsymbol{\alpha} \neq \boldsymbol{m}$. The more copy numbers differ from 2, the higher is the I/NI call, where gains and losses are treated on the same level by the absolute value of the logarithm.

Using Eq. (S25), the I/NI call can be decomposed into contributions from each sample k:

$$\begin{aligned} \mathrm{I/NI}(\boldsymbol{\alpha}) \ &= \ \sum_{i=0}^{n} \alpha_i \ |\log(i/2)| \ = \ \sum_{i=0}^{n} \frac{1}{N} \sum_{k=1}^{N} \alpha_{ik} \ |\log(i/2)| \\ &= \frac{1}{N} \sum_{k=1}^{N} \sum_{i=0}^{n} \alpha_{ik} \ |\log(i/2)| \ = \ \frac{1}{N} \sum_{k=1}^{N} \mathrm{I/NI}(\boldsymbol{\alpha}_k) \ , \end{aligned} \tag{S42}$$

where $\boldsymbol{\alpha}_k = (\alpha_{1k}, \ldots, \alpha_{ik}, \ldots, \alpha_{nk})$. The *individual I/NI call* of sample k is $\mathrm{I/NI}(\boldsymbol{\alpha}_k) = \sum_{i=0}^{n} \alpha_{ik} \ |\log(i/2)|$ which is the contribution of sample k to the I/NI call and the expected copy number fold change of sample k.

S2.4 Segmentation and CNV Call

S2.4.1 Segmentation

CNVs are detected by segmenting the chromosomes of individuals based on their individual I/NI calls, where genomic adjacent I/NI calls that show the same copy numbers are joined. Note, however, that the individual I/NI call defined Eq. (S42) does not allow for distinguishing losses and gains with the same fold change.

The signed individual I/NI call. To avoid joining losses and gains, we define the *signed individual I/NI call* as the expected log fold change:

$$\begin{aligned} \mathrm{sI/NI}(\boldsymbol{\alpha}_k) &= \sum_{i=0}^{n} \alpha_{ik} \ \log(i/2) \qquad \text{(S43)} \\ &\approx \ \mathrm{sgn} \left(\sum_{i=0}^{n} \alpha_{ik} \log(i/2) \right) \ \sum_{i=0}^{n} \alpha_{ik} \ |\log(i/2)| \\ &= \ \mathrm{sgn} \left(\sum_{i=0}^{n} \alpha_{ik} \log(i/2) \right) \ \mathrm{I/NI}(\boldsymbol{\alpha}_k) \end{aligned}$$

The absolute value of the signed I/NI call $|\mathrm{sI/NI}(\boldsymbol{\alpha}_k)|$ is not exactly the individual I/NI call $\mathrm{I/NI}(\boldsymbol{\alpha}_k)$, but the two values are always very close. They are close because for one sample the summands with largest α_{ik} are either ≥ 0 or ≤ 0, that is the model assumes for one sample at one location either a loss ($i/2 \leq 1$) or a gain ($i/2 \geq 1$) if deviating from the prior of constant copy number 2.

Numerical investigation of the difference between sI/NI and expected log fold change. We investigated the numerical difference between the signed I/NI call and the expected log fold change, that is the quality of the approximation in Eq. (S43). Based on data from the Sanger sequencing center on HapMap phase 1 individuals, we calculated the difference of these values for more than 2 million data points and found that the median difference was zero, the third quartile was $6.2e - 17$, the maximum was $1.4e - 02$. Note, that for copy number 3 $\mathrm{sI/NI}$ values are in the range of $0.6 \approx \log(3/2)$ and for copy number 1 in the range of $-1 = \log_2(1/2)$, therefore, the difference between the signed I/NI call and the expected log fold change is negligible.

Segmentation algorithm. The circular binary segmentation algorithm (DNAcopy; Venkatraman and Olshen 2007) is applied to $\mathrm{sI/NI}(\boldsymbol{\alpha}_k)$ along the chromosome. DNAcopy joins consecutive segments with large or small expected fold changes to a candidate segment. Note, that other segmentation algorithms led to similar results as DNAcopy on the experimental data. The segments obtained by the segmentation algorithm are candidate segments as they show a variation along the chromosome indicated by the signed individual I/NI call.

S2.4.2 CNV Call of cn.MOPS

A candidate segment is called a CNV if the median of the signed individual I/NI call $\mathrm{sI/NI}(\boldsymbol{\alpha}_k)$ over the segment is at least 0.6 for gains and at most -1 for losses. Thus, also a variation across samples is needed to call a CNV. The CNV call combines two calls: (1) an I/NI call across samples and (2) a segment call along the chromosome. Only if consecutive segments obtain an I/NI call, they are joined by the segmentation algorithm (see second bar and third sample in Fig. S1). This idea of calling a CNV by two calls, where one call is supplied by a model across samples, has already led to improvements of CNV detection based on DNA microarray data via the cn.FARMS method (Clevert *et al.* 2011).

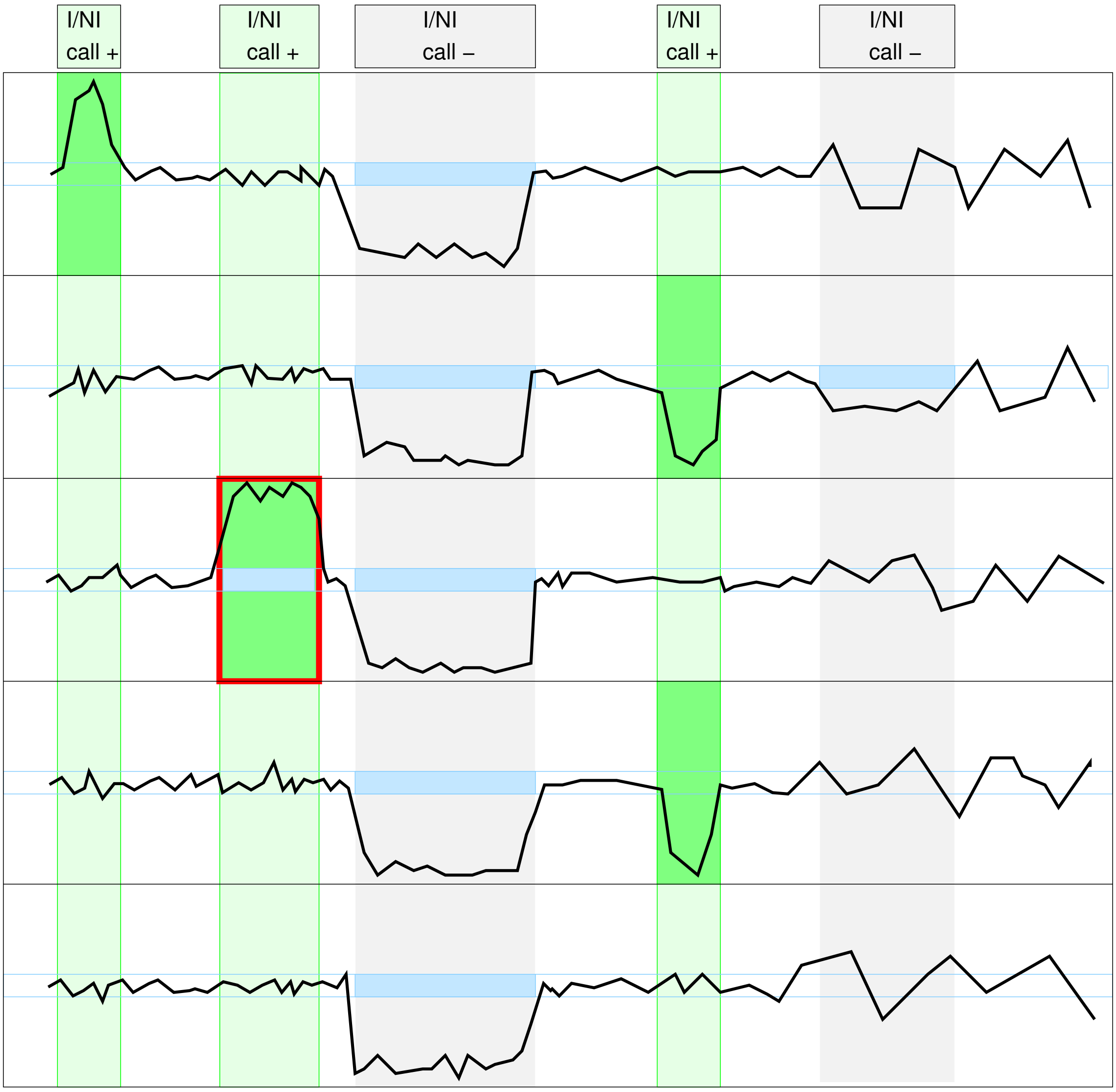


Figure S1: Illustration of the basic concept of cn.MOPS: a CNV call incorporates the detection of variation across samples (I/NI call) and the detection of variation along a chromosome (segmentation). Curves depict read counts along one chromosome for five samples. I/NI calls (green) detect variation across samples (green vertical boxes). A CNV (red box) is called, if consecutive segments have high I/NI calls. Blue boxes mark segments that a segmentation algorithm of class (a) would combine into a CNV. *First vertical bar (from the left) and first sample:* the I/NI call indicates variation across samples ("I/NI call +"). However, too few adjacent segments show high I/NI calls. *Second bar and third sample:* The I/NI call indicates variation across samples ("I/NI call +") and sufficient adjacent segments show high I/NI calls, which leads to a CNV call (red box). *Third bar:* the read counts drop consistently, thus would be detected by a segmentation algorithm of class (a) methods (blue boxes). However, the samples' read counts do not vary, which does not lead to an I/NI call ("I/NI call -"). A CNV is not detected, which is correct as the copy number does not vary across samples. *Fourth bar and samples no. two and four:* I/NI call indicates variation across samples ("I/NI call +"). As in the first bar, too few adjacent segments show high I/NI calls. *Fifth bar and second sample:* a segmentation algorithm of class (a) methods would combine adjacent read counts that are consistently small (blue box) into a CNV. However, the read counts are within the variation of the constant copy number at this location. Therefore the I/NI call does not indicate variation across samples ("I/NI call -").

S2.5 Noise Model Variants

In the following, we introduce some variants of cn.MOPS with different noise assumptions. These variants have in common that not only copy number 0, but also other copy number regions may have additional reads stemming from wrong mappings or sample contamination.

The main problem of these noise assumptions is that only an increase of read counts by noise is modeled, but no decrease. For copy number 0, this is correct, but it is hard to justify for other copy number regions. To allow negative ϵ, that is, a loss of reads, leads to numerical problems at copy number 0 regions. This is the main reason why we included a noise term only for copy number 0 in cn.MOPS.

In the following two subsections, we consider two variants of cn.MOPS with alternative noise models. In a third subsection, we investigate another variant of cn.MOPS where the noise level ϵ is considered as a parameter which is optimized by the EM algorithm. As it will turn out later, the main problem of this approach is that the increased model complexity potentially leads to overfitting.

S2.5.1 Variant (a): Additive Poisson Noise for Each Segment

First, we introduce a variant where, in each segment, additional reads are modeled via additive Poisson-distributed reads with parameter ϵ.

The objective Eq. (S26) for optimizing the average read count λ now becomes:

$$\min_{\lambda} \quad - \frac{1}{N} \sum_{k=1}^{N} \sum_{i=0}^{n} \hat{\alpha}_{ik} \ \log \mathrm{P}(x ; \frac{i}{2} \ \lambda \ + \ \frac{\epsilon}{2}) \ . \tag{S44}$$

Using

$$\log \mathrm{P}(x_k \ ; \ \frac{i}{2} \ \lambda \ + \ \frac{\epsilon}{2}) \ = \ - \ \log(x_k!) \ - \ \frac{i}{2} \ \lambda \ - \ \frac{\epsilon}{2} \ + \ x_k \ (\log(\frac{i}{2} \ \lambda \ + \ \frac{\epsilon}{2})) \ , \tag{S45}$$

the derivative of the objective is

$$- \ \frac{1}{N} \sum_{k=1}^{N} \sum_{i=0}^{n} \left(- \ \frac{i}{2} \ + \ x_k \ \frac{\frac{i}{2}}{\frac{i}{2} \ \lambda \ + \ \frac{\epsilon}{2}} \right) \ \hat{\alpha}_{ik} \ . \tag{S46}$$

Setting this derivative to zero and solving the resulting equation with respect to λ leads to the following alternative update rule:

$$\lambda^{\mathrm{new}} \ = \ \frac{\frac{1}{N} \ \sum_{k=1}^{N} \ x_k \ - \ \frac{\epsilon}{2}}{\sum_{i=0}^{n} \ \hat{\alpha}_i \ \frac{i}{2}} \ . \tag{S47}$$

We will refer to this approach as Variant (a) in the following. Recall that the λ update rule of cn.MOPS is

$$\lambda^{\mathrm{new}} \ = \ \frac{\frac{1}{N} \ \sum_{k=1}^{N} x_k}{\sum_{i=0}^{n} \ \hat{\alpha}_i \ \frac{i}{2}} \ = \ \frac{\frac{1}{N} \ \sum_{k=1}^{N} \ x_k}{\sum_{i=1}^{n} \ \hat{\alpha}_i \ \frac{i}{2} \ + \ \hat{\alpha}_0 \ \frac{\epsilon}{2}} \ . \tag{S48}$$

S2.5.2 Variant (b): Additive Poisson Noise Scales with Average Read Count

Secondly, we introduce a variant where, in each segment, the additional noise reads scale with the average number of reads in this segment.

The objective Eq. (S26) for optimizing the average read count λ now becomes:

$$\min_{\lambda} \quad - \frac{1}{N} \sum_{k=1}^{N} \sum_{i=0}^{n} \hat{\alpha}_{ik} \ \log \mathrm{P}(x \ ; \ \frac{(i \ + \ \epsilon)}{2} \ \lambda) \ . \tag{S49}$$

Using

$$\log \mathrm{P}(x_k \ ; \ \frac{(i \ + \ \epsilon)}{2} \ \lambda) \ = \ - \ \log(x_k!) \ - \ \frac{(i \ + \ \epsilon)}{2} \ \lambda \ + \ x_k \ (\log(\lambda) \ + \ \log(\frac{i \ + \ \epsilon}{2})) \ , \tag{S50}$$

the derivative of the objective is

$$- \ \frac{1}{N} \sum_{k=1}^{N} \sum_{i=0}^{n} \left(- \ \frac{i \ + \ \epsilon}{2} \ + \ x_k \ \frac{1}{\lambda} \right) \ \hat{\alpha}_{ik} \ . \tag{S51}$$

We will refer to this approach as Variant (b) in the following. Setting this derivative to zero and solving the resulting equation with respect to λ leads to the following alternative update rule:

$$\lambda^{\text{new}} \ = \ \frac{\frac{1}{N} \ \sum_{k=1}^{N} \ x_k}{\sum_{i=1}^{n} \ \hat{\alpha}_i \ \frac{i}{2} \ + \ \frac{\epsilon}{2}} \ . \tag{S52}$$

S2.5.3 Variant (c): Poisson Noise Level as Model Parameter

As a third variant, we consider the noise level ϵ as a model parameter which is adjusted by the EM algorithm.

The objective for optimizing ϵ can be derived analogously to Eq. (S26), where the terms containing ϵ are:

$$\min_{\epsilon} \quad - \ \frac{1}{N} \sum_{k=1}^{N} \hat{\alpha}_{0k} \ \log \mathrm{P}(x \ ; \ \frac{\epsilon}{2} \ \lambda) \ . \tag{S53}$$

Using

$$\log \mathrm{P}(x_k \ ; \ \frac{\epsilon}{2} \ \lambda) \ = \ - \ \log(x_k!) \ - \ \frac{\epsilon}{2} \ \lambda \ + \ x_k \ (\log(\lambda) \ + \ \log(\frac{\epsilon}{2})) \ , \tag{S54}$$

the derivative of the objective is

$$- \ \frac{1}{N} \sum_{k=1}^{N} \hat{\alpha}_{0k} \ \left(- \ \frac{1}{2} \ \lambda \ + \ \frac{x_k}{\epsilon} \right) \tag{S55}$$

We will refer to this approach as Variant (c) in the following. Setting this derivative to zero and solving the resulting equation with respect to ϵ leads to the following update rule for ϵ:

$$\epsilon^{\text{new}} \ = \ \frac{2 \ \frac{1}{N} \ \sum_{k=1}^{N} \ \hat{\alpha}_{0k} \ x_k}{\lambda \ \frac{1}{N} \ \sum_{k=1}^{N} \ \hat{\alpha}_{0k}} \ . \tag{S56}$$

The problem with variant (c) is that the model complexity increases by introducing a second parameter; therefore, it is prone to overfitting.

In experiments (see below), we observed that ϵ codes for the average read counts of copy number 2 for data with large copy numbers, while the copy number 2 component is used for large copy numbers. In this case, the model can model the large fold changes in the data, although the assignment of integer copy numbers can be incorrect.

These two drawbacks of adjusting the noise via a model parameter are the main reasons why we decided to consider ϵ as a hyperparameter in the cn.MOPS model.

S2.5.4 cn.MOPS Variants Tested on Simulated Data

First, we compared cn.MOPS with variants (a)–(c) on the simulated data used in the experiments described in Section "Simulated Data with Constructed CNVs" of the main manuscript. Table S2 shows the results. The low performance of variant (c) is caused by overfitting via an overly complex model class. Variant (b) performs best for gains, while cn.MOPS performs best for losses. The differences, however, are only marginal.

Table S2: Performance of cn.MOPS and variants on simulated data. "PR AUC" gives the area under the precision-recall curve. "Recall" reports the recall at a precision of 0.95. Variant (b) performs best for gains and cn.MOPS performs best for losses, but the differences are only marginal.

	Gains		Losses	
Variant	PR AUC	Recall	PR AUC	Recall
cn.MOPS	0.935	0.883	0.963	0.961
(a) additive ϵ	0.931	0.860	0.961	0.959
(b) additive scaled ϵ	0.941	0.889	0.962	0.959
(c) ϵ as parameter	0.904	0.821	0.938	0.933

S2.5.5 cn.MOPS Variants Tested on Real Sequencing Data with Implanted CNVs from the X Chromosome

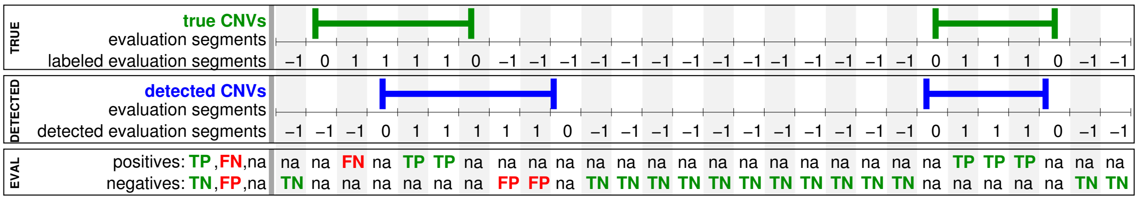
Next, we compared cn.MOPS with variants (a)–(c) on the data used in the experiments described in Section "Real Sequencing Data with Implanted CNVs from the X Chromosome" of the main manuscript. Table S3 shows the results. The low performance of variant (c) is again caused by overfitting as in the previous experiment. Variant (b) performs worse in this experiment, while cn.MOPS performs generally best.

Table S3: Performance of cn.MOPS and variants on real sequencing data with implanted CNVs from the X chromosome. "PR AUC" gives the area under the precision-recall curve. "Recall" reports the recall at a precision of 0.95. For gains, Variant (b) performs best in terms of PR AUC, but only marginally. For losses and in terms of recall for gains, cn.MOPS performs best.

	Gains		Losses	
Variant	PR AUC	Recall	PR AUC	Recall
cn.MOPS	0.703	0.652	0.888	0.878
(a) additive ϵ	0.705	0.640	0.886	0.876
(b) additive scaled ϵ	0.712	0.544	0.844	0.831
(c) ϵ as parameter	0.680	0.590	0.864	0.853

S2.5.6 Conclusion

In summary, other variants of cn.MOPS show inferior performance. For variants (a) and (b) with additive noise for all copy numbers, only additional reads are explained, but potentially missing reads are not modeled. The value of λ is therefore systematically underestimated, which leads to a decrease in performance. Variant (c) considers ϵ as a model parameter and thus uses a model of higher complexity, which leads to overfitting and, consequently, to a decrease in performance. We conclude that cn.MOPS is the best choice compared to the variants considered here.



labeled evaluation segments −1 0 1 1 1 1 0 −1 −1 −1 −1 −1 −1 −1 −1 −1 −1 −1 −1 −1 −1 0 1 1 1 0 −1 −1

detected evaluation segments −1 −1 −1 0 1 1 1 1 1 0 −1 −1 −1 −1 −1 −1 −1 −1 −1 −1 −1 0 1 1 1 0 −1 −1

positives: TP,FN,na na na FN na TP TP na na na na na na na na na na na na na na na na TP TP TP na na na

negatives: TN,FP,na TN na na na na na na FP FP na TN TN TN TN TN TN TN TN TN TN TN na na na na na TN TN

Figure S2: Definitions for the evaluation of copy number detection methods. A genome is split into equally sized evaluation segments of length shorter than the shortest CNV. *Top panel*: Knowing the true CNV regions (green), the evaluation segments are labeled as class 1 (CNV segment) or class -1 (non-CNV segment). *Middle panel*: A CNV detection method classifies each evaluation segment into CNV segments (blue, class 1) and non-CNV segments (class -1). *Bottom panel*: In the first line, positives (known CNV regions) are divided into true positives (TP, green) and false negatives (FN, red). In the second line, negatives (no overlap with known CNV regions) are divided into true negatives (TN, green) and false positives (FP, red). Segments partly overlapping with known or predicted CNV regions are not considered (“na”).

S3 Experiments

S3.1 Evaluation of CNV Detection Results

In order to compare methods which detect copy number variations in next generation sequencing data, we need an evaluation criterion. We assume that the true CNVs are known and to be rediscovered. Each chromosome is split into equally large evaluation segments the size of which is chosen to accommodate the shortest known CNV. An evaluation segment is called a *true positive* (TP) if it is entirely contained both in a true CNV and in a detected CNV segment. It is called a *false negative* (FN) if it is entirely contained in a true CNV but does not overlap with any predicted CNV segment. An evaluation segment is called a *false positive* (FP) if it is entirely detected as a CNV segment but does not overlap with any true CNV. Finally, it is called a *true negative* (TN) if it overlaps neither with a true CNV nor with a detected CNV segment. These definitions imply that all evaluation segments that partly overlap with true CNVs or detected CNV segments remain ignored, as the copy numbers in these segments are ambiguous. Figure S2 illustrates the definitions of the four categories of evaluation segments. The two measures we employ hereafter are *recall* $\big(\#\mathrm{TP}/(\#\mathrm{TP}+\#\mathrm{FN})\big)$ and *precision* $\big(\#\mathrm{TP}/(\#\mathrm{TP}+\#\mathrm{FP})\big)$. Note that precision is one minus the false discovery rate, in which we are especially interested. A CNV calling threshold governs the trade-off between recall and precision or, in other words, the trade-off between FNs and FPs, because more detected CNVs lead to more FPs but fewer FNs, and vice versa. To assess the performance of methods at different CNV calling thresholds, we use *precision-recall curves*. Precision-recall curves are independent of the number of TNs, which makes them an ideal tool for our evaluation, as the majority of samples are negatives (non-CNVs).

We also considered using “receiver-operator characteristic (ROC) curves” or the Matthews correlation coefficient as evaluating criterion for the performance of different methods. However, we decided to use the area under the precision recall curve as evaluation criterion because it is independent of the number of true negatives. Our data sets contain many negatives (segments with constant copy number 2) and most methods classify the majority of them correctly which leads to a large number of true negatives. By using precision-recall curves we avoid that methods which tend to classify most segments as negatives, that are methods with a low discovery power (low recall), systematically obtain higher performance values.

S3.2 Compared CNV Detection Methods

We compared following methods:

1. cn.MOPS our new model and pipeline,
2. MOFDOC according to the variant described in (Alkan *et al.* 2009),
3. EWT (Yoon *et al.* 2009) event-wise testing,
4. JointSLM (Magi *et al.* 2011),
5. CNV-Seq (Xie and Tammi 2009),
6. FREEC (Boeva *et al.* 2011).

For a fair comparison, the parameters of the methods were optimized on simulated data sets similar to the one we used in our first experiment in Section S3.3.

S3.2.1 cn.MOPS

For cn.MOPS model we initialized the parameter λ by the median read count of the segment across samples $\bar{\lambda}$. ϵ from Eq. (S1) is set to $\epsilon = 0.05$, which is our estimate for the percentage of wrongly mapped reads. We set $n = 8$ which leads to nine possible copy numbers $0 \leq i \leq 8$. The parameter $\boldsymbol{\alpha}$ should be initialized close to the location of the prior's mode $(0, 0, 1, 0, \ldots, 0)$, which are the optimal parameters if all samples have copy number 2. However, initializing $\boldsymbol{\alpha}$ by $(0, 0, 1, 0, \ldots, 0)$ would clamp all $\hat{\alpha}_{ik}$ and α_i^{new} to zero according to Eq. (S31) and Eq. (S32). Therefore we initialized $\boldsymbol{\alpha}$ by $\boldsymbol{\alpha} = (0.05, 0.05, 0.6, 0.05, \ldots, 0.05)$.

S3.2.2 Class (a) Methods: MOFDOC, EWT , and JointSLM

The methods MOFDOC ("model free depth of coverage") according to Alkan *et al.* (2009), EWT ("event-wise testing") according to Yoon *et al.* (2009), and JointSLM (Magi *et al.* 2011) are all based on detecting deviations of read counts from an average read count which can be measured by z-scores or log z-scores, i.e. the multiple in standard deviations the read count differs from the mean.

MOFDOC: We implemented MOFDOC using the CNV calling criterion from Alkan *et al.* (2009). A CNV region is called if a out of b consecutive segments show a read count with a z-score beyond a threshold (abnormal large or small read counts) to call a segment (default $a = 6$ and $b = 7$). We generalized this "a-b-smoother" to a smoothing algorithm, that is not only able to smooth logical, but also real values stemming from CNV calls (see Section S3.5 and Fig. S5), which improved MOFDOC's results.

All parameters are optimized on artificial test data sets similar to one used in Section S3.3, which resulted in following parameter settings:

- a=4

- `b=4`
- `GCcorrection=TRUE`

The parameter `WL` was set to `25000/2500/500` for the low coverage, medium coverage and high coverage data set, respectively.

EWT: We reimplemented EWT as described in Yoon *et al.* (2009) but improved the GC correction by using all samples for estimating the GC effect. Further we restricted the parameter "event size" to an upper bound `maximumEventSize` and a lower bound `minimumEventSize`. "event size" is a parameter that prevents EWT from testing too short CNVs. We generalized EWT to variable segment sizes like 10kbp, 25kbp and 50kbp that are apt for low coverage CNV detection. Our modifications of EWT improved its results. EWT adjusts a threshold on the p-values of joined segments (called "false positive rate") assuming independent Gaussian read counts per segment. This threshold governs the number of detections. To compute the precision-recall curve we considered all possible thresholds of the log "false positive rate". All parameters are optimized on artificial test data sets similar to one used in Section S3.3, which resulted in following parameter settings:

- `minimumEventSize = 4`
- `maximumEventSize = 8`
- `GCcorrection = TRUE`

The parameter `WL` was set to `25000/2500/500` for the low, medium and high coverage data set, respectively.

JointSLM: We applied the R -package (version 0.1) to 25kbp/2500bp/500bp (low/medium/high coverage) segments, for which the GC content was computed. In contrast to the original implementation, we did not round the scores per segment to integer copy numbers to allow for thresholding and to compute the area under precision-recall curve. All parameters are optimized on artificial test data sets similar to one used in Section S3.3, which resulted in following parameter settings:

- `omega = 0.1`
- `eta = 1e-06`
- `K0 = 20`
- `baseCopy = 2`

S3.2.3 Class (b) Methods: SeqSeg, CNV-Seq, and FREEC

The methods SeqSeg (Chiang *et al.* 2008), CNV-Seq (Xie and Tammi 2009), and FREEC (Boeva *et al.* 2011) require a reference genome for copy number detection. Most of these methods are designed for and applied to studies with tumor samples and matched normal samples, which are often blood cells of the same individual. For CNV detection, matched normals are in general not available. Therefore, reference read counts per segment are build as the median of read counts over all samples. Additionally, the median is more robust than using a matched sample, because both the reference and the analyzed genome are subject to random read count variations (note, the read count variation is not estimated across samples). For the method SeqSeg, which requires read positions on the genome, we generated a reference genome in the following way: we pooled the read positions of all samples and then sorted them according to their genomic position. We then used the median of n (number of samples) consecutive reads as read position for the reference genome.

SeqSeg: We did not include SeqSeg (Chiang *et al.* 2008) in the comparisons because we were not able to find suitable parameters, not even after an extensive search. The problem was that SeqSeg either did not detect any breakpoints or the thresholds for the p-values were not determined. However, the performance of SeqSeg can be estimated via CNV-Seq which is very similar to SeqSeg.

CNASeg: We also omitted CNAseg (Ivakhno *et al.* 2010) from the comparison, as its developers state that this method is specifically tailored to CNA detection in tumor samples.

CNV-Seq: We used the authors' implementation[1], where the median of the samples' read counts served as reference read count. All parameters are optimized on artificial test data sets similar to one used in Section S3.3, which resulted in following parameter settings:

- `pValue = 0.001`
- `log2Threshold = 0.6`

All other parameters were set to their default values. The parameter `windowLength` was set to `25000/2500/500` for the low, medium and high coverage data set, respectively.

FREEC: We used Version 3.2[2], where analogously to CNV-Seq, the median of the samples' read counts was used as reference. For computing the precision-recall curves, all possible thresholds for the returned median ratio per segment are used. All parameters are optimized on artificial test data sets similar to one used in Section S3.3, which resulted in following parameter settings:

- `breakPointThreshold = -0.001`
- `ploidy = 2`
- `minCNAlength = 4`
- `step = 10000`
- `mode = reference`

All other parameters were set to their default values. The parameter `window` was set to `25000/2500/500` for the low, medium and high coverage data set, respectively.

[1] http://tiger.dbs.nus.edu.sg/cnv-seq/ (version as of 2010/07/16)

[2] http://bioinfo-out.curie.fr/projects/freec/ (version as of 2011/04/04)

S3.3 Simulated Sequencing Data with Constructed CNVs

We constructed 100 artificial benchmark data sets. We assume an artificial genome to consist of a single chromosome of 125Mbps length which is divided into 5,000 segments of length 25kbp. We created 40 samples by sampling read counts for all segments and all samples according to a Poisson process. The overall number of evaluation segments was, therefore, $40 \times 5,000 = 200,000$.

S3.3.1 Distribution of CNV Types and Copy Numbers for Data Generation

We determined characteristics of CNV regions and how copy numbers are distributed from the HapMap individuals (The International HapMap 3 Consortium 2010). CNVs of different individuals cluster at certain regions of the DNA, the "CNV regions", of which many contain only losses or only gains. CNV regions can be divided into 3 types: CNV regions of the type "loss region" contain only losses, of type "gain region" contain only gains, and of type "mixed region" contain both losses and gains. As depicted in Fig. S3, the CNV region type "loss region" was observed in 80%, "gain region" in 15%, and "mixed region" in 5%.

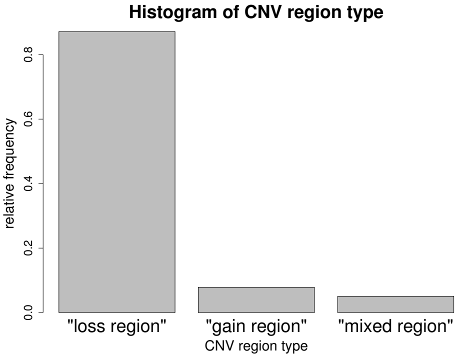


Figure S3: Histogram of CNV region types "loss region", "gain region", and "mixed region" according to The International HapMap 3 Consortium (2010).

We implanted 20 CNV regions into each of these benchmark chromosomes. The CNV regions' lengths were chosen randomly from the interval 75–200kbp, which is the range of accurate detection for the given coverage according to Xie and Tammi (2009). The 20 starting points of the CNV regions are randomly chosen along the chromosome. After having determined the 20 CNV regions, we have to decide how CNVs are implanted into the single samples. According to the HapMap individuals, we assign CNV region types such that 80% are "loss region" (contain only losses), 15% "gain region" (contain only gains), and 5% "mixed region" (contain both losses and gains). Then the actual copy number for each sample is drawn according to the copy numbers observed for HapMap individuals (The International HapMap 3 Consortium 2010): For a loss region, a sample has probabilities of 0.8, 0.15, and 0.05 of having copy numbers 2, 1, and 0, respectively. For a gain region, a sample has probabilities of 0.85, 0.08, 0.06, and 0.01 of having copy numbers 2, 3, 4 and 5, respectively. For a mixed region, a sample has probabilities 0.04, 0.16, 0.67, 0.11, and 0.02 of having copy numbers 0, 1, 2, 3 and 4, respectively. Of the 200,000 evaluation segments, on average 101(±56) are gains and 612 (±104) are losses. The CNVs' lengths range from 75,006bp to 199,848bp with an average of 136,921bp.

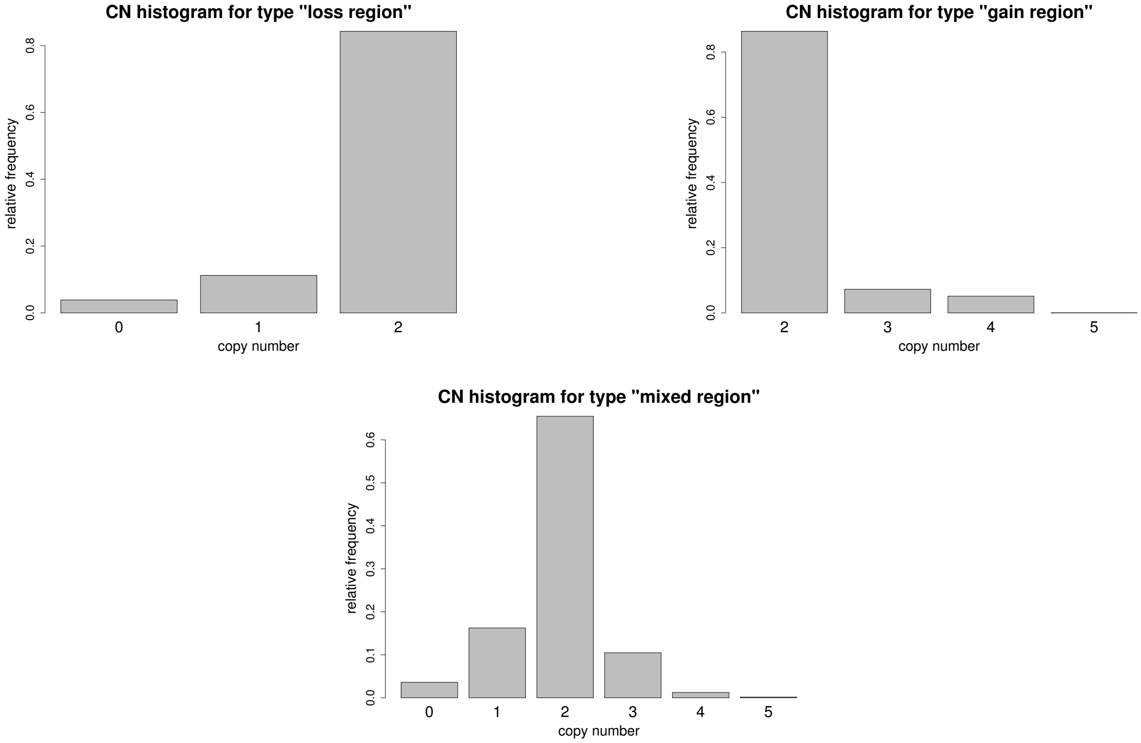


Figure S4: Histograms of integer copy numbers for CNV region types “loss region”, “gain region”, and “mixed region” according to The International HapMap 3 Consortium (2010).

S3.3.2 Results

Table S4 reports the performance of the compared copy number detection methods separately for gains and losses. As evaluation measures, we use the area under the precision-recall curve and the recall for an FDR fixed to 0.05.

Table S4: Performance of the compared copy number detection methods on the artificial benchmark data set. "PR AUC" gives the average area under the precision-recall curve of 100 experiments. The second column "p-value" reports the p-value of a Wilcoxon signed-rank test (over the 100 experiments) with null hypothesis that cn.MOPS and another method have the same area under the curve. "Recall" reports the recall at a precision of 0.95, that is, an FDR of 0.05. The last column "p-value" gives the p-value of an analogous Wilcoxon test for the recall with an FDR of 0.05. cn.MOPS had significantly higher performance than all other methods.

Gains				
	PR AUC	p-value	Recall	p-value
cn.MOPS	**0.94**	—	**0.88**	—
MOFDOC	0.81	1.14e-13	0.76	9.75e-12
EWT	0.79	5.95e-14	0.74	1.34e-12
JointSLM	0.25	4.23e-18	0.22	2.80e-17
CNV-Seq	0.35	4.23e-18	0.35	3.98e-17
FREEC	0.65	1.95e-17	0.53	3.42e-14

Losses				
	PR AUC	p-value	Recall	p-value
cn.MOPS	**0.96**	—	**0.96**	—
MOFDOC	0.92	3.50e-17	0.90	9.22e-17
EWT	0.91	3.20e-18	0.90	8.44e-17
JointSLM	0.34	1.98e-18	0.28	1.98e-18
CNV-Seq	0.81	1.98e-18	0.81	3.84e-17
FREEC	0.73	1.98e-18	0.72	3.32e-17

S3.3.3 Different Performance on Gains and Losses

Table S4 shows that all methods perform better at detecting losses. The superior performance at losses can be explained by the fact that copy number 3 can be more likely be confused with copy number 2 than copy number 1. We show that under a Poisson assumption, a typical read for copy number 2, that is λ, is more likely to come from copy number 3 than from copy number 1.

Assuming $\lambda > 0$, we obtain:

$$\begin{aligned} & 0 \; < \; \log(3) \; - \; 1 \\ \Longleftrightarrow \; & -\frac{1}{2} \; < \; \log(3) \; - \; \frac{3}{2} \\ \Longleftrightarrow \; & \log\left(\frac{1}{2}\right) \; - \; \frac{1}{2} \; < \; \log\left(\frac{3}{2}\right) \; - \; \frac{3}{2} \\ \Longleftrightarrow \; & \lambda \, \log\left(\frac{1}{2}\right) \; + \; \lambda \, \log(\lambda) \; - \; \frac{1}{2}\,\lambda \; < \; \lambda \, \log\left(\frac{3}{2}\right) \; + \; \lambda \, \log(\lambda) \; - \; \frac{3}{2}\,\lambda \\ \Longleftrightarrow \; & \left(\frac{1}{2}\,\lambda\right)^{\lambda} e^{-\frac{1}{2}\lambda} \; < \; \left(\frac{3}{2}\,\lambda\right)^{\lambda} e^{-\frac{3}{2}\lambda} \\ \Longleftrightarrow \; & \mathrm{P}\left(\lambda\, ;\, \frac{1}{2}\lambda\right) \; < \; \mathrm{P}\left(\lambda\, ;\, \frac{3}{2}\lambda\right) , \end{aligned} \tag{S57}$$

where $\mathrm{P}(x;\beta)$ is the Poisson distribution with parameter β evaluated at x. The inequality shows that the average read count λ for copy number 2 has higher probability to be drawn from a copy number 3 than from a copy number 1 distribution.

S3.4 Real Sequencing Data with Implanted CNVs From the X Chromosome

In contrast to the previous benchmark for which the read counts were simulated, we now consider real reads stemming from sequencing of a single male HapMap individual (NA20755). This man's genome was sequenced 17 times by the Solexa Genome Analyzer II at the Sanger Sequencing Center (see Table S5). These 17 samples ensure a constant copy number, as they stem from the same individual. The reads were mapped by Bowtie (Langmead *et al.* 2009) for paired reads. We allowed for two mismatches. The numbers of reads range from 12,069,758 to 18,810,212 of which between 10,419,510 and 16,041,464 could be mapped, which corresponds to coverages between 0.13 and 0.21.

We created 110 benchmark data sets by choosing each human chromosome 1–22 five times, where in each chromosome data set 20 random CNV regions were implanted. The lengths of these implanted CNV regions were chosen to be 75kbp, 100kbp, 150kbp and 200kbp (5 each), and for each of the regions a random segment on the X chromosome was selected which supplied reads for the region. CNV region types and individual copy numbers were determined according to the same procedure and distributions as described in Subsection S3.3 except that we only consider CNV copy numbers 1 and 3, since they are most difficult to distinguish from copy number 2. We chose 80% of the CNV regions as loss regions, 15% as gain regions, and 5% as mixed regions. For a loss region, a sample has probabilities 0.8 and 0.2 of having copy number 2 and 1, respectively, for a gain region, 0.85 and 0.15 of having copy numbers 2 and 3, respectively, and for a mixed region, 0.2, 0.67, and 0.13 of having copy numbers 1, 2, and 3, respectively. Finally, the read counts of the 17 samples are computed in the following way: outside CNVs, for constant copy numbers, the original reads counts are used; within CNVs we added as many read counts as there are copies from the corresponding segment on the X chromosome, where read counts are obtained from the considered sample and other random samples.

The CNV detection results were evaluated as described in Subsection S3.1. The number of evaluation segments ranges from around 32,000 for chromosome 21 to around 168,000 for chromosome 1. On average, 0.1% of the evaluation segments are gains and 0.4% are losses.

S3.4.1 Data and Mapping

The sequencing reads of the male sample NA20755 were obtained from the 1000 genomes project (The 1000 Genomes Project Consortium 2010) web page (http://www.1000genomes.org). TableS5 lists the unique names of the sequencing read files. We applied the Bowtie software (Langmead *et al.* 2009) to map the reads against the human reference genome 18 (build 36). The Bowtie parameters were set as follows:

- `-q` $\Longrightarrow$ Input files are fastq files.
- `-v 2` $\Longrightarrow$ Two mismatches are allowed.
- `-M 1` $\Longrightarrow$ M-alignment mode. Reports at most one valid alignment. If more than one best mapping position is available then the read is randomly assigned to one of them.
- `--best` $\Longrightarrow$ The alignment is the best matching position.
- `--sam` $\Longrightarrow$ Output format is SAM.

Table S5 further lists the number of sequenced reads, the number of mapped reads, the number of used reads, and the ratio of mapped reads.

Mapping reads to only unique positions by Bowtie parameter `-m 1` leads to many segments with low read counts as the histogram in Fig. S14 in Subsection S4.1 shows. Compared to the histogram in Fig. S13 in Subsection S4.1 for non-uniquely mapped reads, we observe a shift of the density toward lower read counts (left) because some segments systematically loose reads due to ambiguous mappings. Because there are too many low read counts, class (a) methods like MOFDOC, EWT, and JointSLM are not suited for this kind of read mapping. However, methods based on ratios perform well on data from unique position mapping, since they use reference read counts which have similar read counts as the samples for the same copy number. cn.MOPS is also suited to handle data from unique position mapping as it builds a local model, which regards read count characteristics.

S3.4.2 Results

Table S6 reports the performance of the compared copy number detection methods separately for gains and losses. As before, we use area under the precision-recall curve and the recall for an FDR fixed to 0.05.

S3.4.3 Estimation of Integer Copy Numbers

For this experiment we were able to evaluated the model's performance of assigning an integer copy number to each sample in each genomic location. Evaluation segments that contained a CNV breakpoint were excluded from the further analysis since they have no unique copy number. Inside CNVs 92.293% (± 0.026%) and on the whole genome, including constant copy number two, 99.383%(± 0.001%) of the integer copy numbers were correctly assigned.

Table S5: Summary of the sequencing data for the implanted CNVs benchmark data set. Column “individual” reports the ID number of the HapMap sample. Column “base name” shows the base name of the files containing the sequence reads. Mate 1 has the filename “base name_1.filt.fastq.gz” and mate 2 “base name_2.filt.fastq.gz”. The following columns “sequenced reads”, “mapped reads” and “used reads” report the number of totally sequenced reads, the number of reads that were mapped to the reference genome, and the number of reads used for the analysis (after removing potential PCR duplicates). The last column “ratio mapped” gives the proportion of mapped reads to sequenced reads.

			sequenced	mapped	used	ratio
	individual	base name	reads	reads	reads	mapped
1	NA20755	ERR003683	17,303,620	14,890,416	14,846,540	0.86
2	NA20755	ERR003775	16,453,240	14,017,334	13,974,864	0.85
3	NA20755	ERR003776	17,471,474	15,051,916	14,998,386	0.86
4	NA20755	ERR003777	18,275,368	15,763,846	15,702,220	0.86
5	NA20755	ERR003778	18,651,270	15,892,764	15,831,810	0.85
6	NA20755	ERR003779	18,566,172	15,881,196	15,816,916	0.86
7	NA20755	ERR003780	18,611,728	15,840,592	15,774,308	0.85
8	NA20755	ERR003781	17,713,468	14,974,160	14,918,680	0.85
9	NA20755	ERR003782	17,649,380	15,039,086	14,995,282	0.85
10	NA20755	ERR003783	18,810,212	16,041,464	15,991,994	0.85
11	NA20755	ERR003784	18,650,906	16,020,806	15,971,248	0.86
12	NA20755	ERR003785	16,960,214	14,052,008	14,012,730	0.83
13	NA20755	ERR003786	14,799,104	11,741,258	11,711,502	0.79
14	NA20755	ERR003787	12,888,982	10,419,510	10,395,492	0.81
15	NA20755	ERR003855	15,003,086	13,131,902	13,067,934	0.88
16	NA20755	ERR003867	17,359,258	15,090,776	15,044,648	0.87
17	NA20755	ERR003878	12,069,758	10,652,836	10,603,636	0.88

S3.5 Rediscovering of Known CNVs in HapMap Sequencing Data

Finally, we compare how well the methods are able to rediscover known CNVs of HapMap individuals whose DNA was sequenced by the Solexa Genome Analyzer II at the Sanger Sequencing Center. We focused on 18 individuals for each of which the reads were produced on one lane (one sequencing run contains 7 lanes). The reads were mapped by Bowtie (Langmead *et al.* 2009) for paired reads, again allowing three mismatches. The numbers of reads range from 12,442,124 to 31,977,690 of which 7,498,420 to 22,217,020 could be mapped, leading to a coverage between 0.20 and 0.60 (see Supplement for details on read mapping and the number of reads).

The CNVs of these 18 individuals have been determined previously using microarrays (The International HapMap 3 Consortium 2010) which we consider as the true CNVs in the following. These true CNVs were detected by the Affymetrix Human SNP array 6.0 and reconfirmed with the Illumina Human1M-single beadchip. After filtering for CNVs larger than 75kbp, we obtained 170 CNVs, of which 66 are gains and 104 are losses, with lengths ranging from 76kbp to 457kbp. The CNV detection results are evaluated as described in Subsection S3.1 with evaluation segments of length 25kbp. In total, we have 2,064,906 evaluation segments of which 450 are labeled as losses as they lie within one of the 104 loss CNVs and 469 are labeled as gains as they lie within one of

Table S6: Performance of the compared copy number detection methods on real sequencing data with implanted CNVs from the X chromosome. “PR AUC” gives the average area under the precision-recall curve of 100 experiments. The second column “p-value” reports the p-value of a Wilcoxon signed-rank test (over the 100 experiments) with null hypothesis that cn.MOPS and another method have the same area under the curve. “Recall” reports the recall at a precision of 0.95, that is, an FDR of 0.05. The last column “p-value” gives the p-value of an analogous Wilcoxon test for the recall with an FDR of 0.05. cn.MOPS significantly outperformed all other methods.

Gains

	PR AUC	p-value	Recall	p-value
cn.MOPS	**0.70**	—	**0.65**	—
MOFDOC	0.20	1.12e-17	0.10	2.31e-17
EWT	0.22	1.95e-16	0.13	8.70e-17
JointSLM	0.06	1.94e-19	0.03	7.00e-18
CNV-Seq	0.13	1.74e-19	0.13	5.75e-18
FREEC	0.49	1.22e-12	0.30	4.41e-15

Losses

	PR AUC	p-value	Recall	p-value
cn.MOPS	**0.89**	—	**0.88**	—
MOFDOC	0.57	3.78e-15	0.21	2.48e-18
EWT	0.62	1.77e-12	0.34	2.02e-17
JointSLM	0.17	4.43e-20	0.08	4.43e-20
CNV-Seq	0.50	4.43e-20	0.50	4.43e-20
FREEC	0.52	7.05e-17	0.36	4.56e-20

the 66 gain CNVs.

S3.5.1 Data and Mapping

The sequence reads of 18 different HapMap samples were obtained from the 1000 genomes project (The 1000 Genomes Project Consortium 2010) web page (http://www.1000genomes.org). Table S7 lists the unique names of the sequence read files. We used the Bowtie software (Langmead *et al.* 2009) to map the reads against the human reference genome 18 (build 36). The Bowtie parameters were set as follows:

- `-q` ⟹ Input files are fastq files.
- `-v 3` ⟹ Three mismatches are allowed.
- `-M 1` ⟹ M-alignment mode. Reports at most one valid alignment. If more than one best mapping position is available then the read is randomly assigned to one of them.
- `--best` ⟹ The alignment is the best matching position.
- `--sam` ⟹ Output format is SAM.

Table S7 further lists the number of sequenced reads, the number of mapped reads, the number of used reads, and the ratio of mapped reads.

Table S7: Summary of the sequencing data for the HapMap CNV reconfirmation benchmark data set. Column "individual" reports the ID number of the HapMap sample. Column "base name" displays the base name of the files containing the sequence reads. Mate 1 has the filename "base name_1.filt.fastq.gz" and mate 2 "base name_2.filt.fastq.gz". The following columns "sequenced reads", "mapped reads" and "used reads" report the number of totally sequenced reads, the number of reads that were mapped to the reference genome, and the number of reads used for the analysis (after removing potential PCR duplicates). The last column "ratio mapped" gives the proportion of mapped reads to sequenced reads.

			sequenced	mapped	used	ratio
	individual	base name	reads	reads	reads	mapped
1	NA11832	SRR023299	31,977,690	22,217,020	22,056,918	0.69
2	NA11920	SRR024102	24,558,838	16,315,660	16,177,166	0.66
3	NA12003	SRR020472	19,089,406	15,043,846	14,956,114	0.79
4	NA12045	SRR020475	22,117,122	15,081,430	15,023,326	0.68
5	NA12154	SRR023306	17,922,674	14,292,494	14,236,294	0.80
6	NA07051	SRR023301	12,442,124	7,498,420	7,472,714	0.60
7	NA07347	SRR029852	26,281,042	10,691,556	10,649,784	0.41
8	NA11831	SRR027529	25,276,660	19,211,670	19,067,842	0.76
9	NA18486	SRR027528	20,145,972	11,397,190	11,367,976	0.57
10	NA18499	SRR011011	24,605,896	16,997,702	16,839,102	0.69
11	NA18510	SRR024103	25,813,190	18,631,922	18,541,544	0.72
12	NA18516	SRR020479	22,861,892	17,427,526	17,398,112	0.76
13	NA18519	SRR018113	17,574,956	10,444,800	10,420,156	0.59
14	NA18871	SRR020470	19,613,172	15,084,744	15,022,780	0.77
15	NA18959	SRR023789	27,674,990	9,137,412	9,123,166	0.33
16	NA18960	SRR029849	16,607,250	9,384,384	9,245,626	0.57
17	NA18964	SRR022591	23,445,028	18,035,194	17,979,096	0.77
18	NA19102	SRR023300	25,236,608	18,016,936	17,987,502	0.71

S3.5.2 Results

Table S8 shows the performance of the six compared methods in rediscovering known CNVs for the 18 HapMap individuals, where the average area under the precision-recall curve is used as evaluation criterion. All methods perform better at detecting losses as already seen in previous experiments. cn.MOPS yields a significantly higher performance than its competitors both in terms of the AUC as well as in terms of the recall for FDR set to 0.05, except that FREEC performs equally well for gains.

Table S8: Performance of the compared copy number detection methods on HapMap individuals, where known CNVs should be rediscovered. "PR AUC" gives the average area under the precision-recall curve of 18 samples. "p-value" reports the p-value of a Wilcoxon signed-rank test (over the 18 samples) with null hypothesis that cn.MOPS and another method have the same area under the curve. "Recall" reports the recall at a precision of 0.95, that is, an FDR of 0.05. The last column "p-value" gives the p-value of an analogous Wilcoxon test for the recall with an FDR of 0.05. cn.MOPS could most reliably reconfirm known CNVs. Only for gains, FREEC and cn.MOPS have similar performance, whereas cn.MOPS has significantly higher performance than its competitors at losses.

Gains

	PR AUC	p-value	Recall	p-value
cn.MOPS	**0.35**	—	**0.24**	—
MOFDOC	0.13	1.17e-03	0.06	1.95e-03
EWT	0.16	5.34e-04	0.10	1.86e-02
JointSLM	0.08	3.81e-05	0.05	7.81e-03
CNV-Seq	0.22	1.74e-02	**0.21**	3.61e-01
FREEC	**0.35**	8.68e-01	**0.17**	2.38e-01

Losses

	PR AUC	p-value	Recall	p-value
cn.MOPS	**0.53**	—	**0.45**	—
MOFDOC	0.40	2.67e-04	0.33	3.42e-03
EWT	0.36	7.63e-06	0.23	6.10e-05
JointSLM	0.15	3.81e-06	0.06	1.53e-05
CNV-Seq	0.32	7.63e-05	0.27	3.66e-04
FREEC	0.42	2.37e-03	0.26	1.01e-03

S3.5.3 Evaluation based on a different criterion

In an additional assessment we determine how many known CNVs are rediscovered in NGS data, but now we do not regard the precision of the detection in terms of CNV length and position. In previous experiments we used evaluation segments and assessed not only whether known CNVs are rediscovered but also how precisely. However, array techniques were not able to precisely determine the known CNVs' breakpoints, thus the ground truth is not reliable concerning length and position of known CNVs.

We are interested in the recall, the true positive rate $\big(\#\mathrm{TP}/(\#\mathrm{TP} + \#\mathrm{FN})\big)$, where the positives are the 170 known CNVs (66 gains and 104 losses). As we do not regard CNV length and position, we redefine true positives: a known CNV is a *true positive* of a method's result if at least one of its detected CNVs overlaps with a known CNV. A method should not be able to improve its performance by calling more CNVs, because the increased recall comes at the cost of more false positives and hence a reduced precision $\big(\#\mathrm{TP}/(\#\mathrm{TP} + \#\mathrm{FP})\big)$. To trade true positives off against false positives, we limit the number of detections $\big(\#\mathrm{TP} + \#\mathrm{FP}\big)$ for each method. As detections we select the 66 top ranked gain segments and the 104 top ranked loss segments in accordance with the known CNVs. Note that there is bias toward methods that detect longer segments, because they are more likely to overlap with known CNVs (we avoided this bias with the precision-recall curves used above).

Table S9 shows the recall results without regarding the precision in terms of CNV length and position. cn.MOPS had significantly (McNemar's test) larger recall values.

Table S9: Recall of known copy number regions by detection methods on HapMap individuals without regarding the precision in terms of CNV length and position. "recall" is the recall (true positive rate) and "p-value" gives the p-values of McNemar's test which indicates whether cn.MOPS has a larger recall than its competitors. Recall values in boldface indicate methods that have significantly larger recall than all other methods. cn.MOPS has significantly larger recall values than other methods.

	Gains		Losses	
	Recall	p-value	Recall	p-value
cn.MOPS	**0.58**	—	**0.75**	—
MOFDOC	0.00	1.95e-09	0.26	2.53e-12
EWT	0.12	1.19e-07	0.16	1.56e-14
JointSLM	0.00	1.95e-09	0.18	4.32e-14
CNV-Seq	0.45	1.33e-02	**0.68**	1.46e-01
FREEC	0.08	2.54e-08	0.51	1.59e-06

S3.5.4 CNV Calls of Different Methods

So far we have considered CNV detection as a classification task whose goal was to detect CNVs in individual samples. Next we assess the quality of the CNV calling across HapMap samples for detecting CNV regions. In contrast to the previous task, we consider a CNV call for a genomic segment across samples but not individual CNV calls. The task is to classify the 114,717 evaluation segments from Subsection S3.1 into segments within a CNV region or non-CNV segments.

The CNV calls have to be defined depending on the method. For cn.MOPS we can readily use the I/NI call. The hyperparameter G of cn.MOPS model was set to $100 \cdot N$. For class (a) methods, namely MOFDOC, EWT, and JointSLM, we use the mean of the z-score on the evaluation segment. For the class (b) methods, CNV-Seq and FREEC, we use the mean log-ratios of the evaluation segments. Log-ratios per segment were computed as the log of the read count divided by the segment's median read count. Note, that the calls shown in the following plots are not the final calls, since all methods suggest a segmentation algorithm that joins initial segments to larger segments for the final CNV call.

Fig. S5 visualizes the results of this task by whole genome CNV calling plots along all evaluation segments. cn.MOPS separates segments within true CNV regions (indicated by red dots) from normal segments (blue dots) better than the other methods. Furthermore, cn.MOPS has lower FDRs for different calling thresholds, as can be seen from the lower variance of the blue dots at the bottom. cn.MOPS's superior performance at CNV calling across samples is the reason that cn.MOPS has outperformed the other methods in previous experiments.

In Fig. S6 the CNV call is based on variances. For cn.MOPS the variance of the individual I/NI call is used. For both the z-score and the log-ratio based CNV calls their variances are used. Also for the variance-based criterion cn.MOPS separates segments within true CNV regions better from non-CNV segments than the other methods.

Finally, in Fig. S7 the CNV call is based on maximal values across samples. For cn.MOPS the maximum of the individual I/NI call is used. For both the z-score and the log-ratio based CNV calls their maxima are used. Also for this maximum criterion cn.MOPS separates segments within true CNV regions better from non-CNV segments than the other methods.

The superiority of the I/NI call over z-score or log-ratio based methods can not only be deduced from the visualizations in Fig. S5, Fig. S6, and Fig. S7 but also from the area under the precision-recall curve (PR AUC). We compared the performance of the mean, variance and maximum of the I/NI call, z-scores and log-ratios. Table S10 reports the area under the precision-recall curve of different methods, where the task was to classify a genomic segment into segments within CNV regions or non-CNV segments. The classification thresholds were the mean, variance and maximum of the individual I/NI call, z-scores and log-ratios. Note that there was no segmentation algorithm applied.

Table S10: Performance of different approaches for CNV calling. The task was to classify genomic segments into segments within CNV regions and non-CNV segments. The mean, variance and maximum of the individual I/NI call, z-scores and log-ratios served as classification criteria and allowed to compute the area under the precision-recall curve (PR AUC). cn.MOPS outperformed the other methods in all three CNV calling approaches.

	PR AUC		
	mean	variance	maximum
individual I/NI call	**0.18**	**0.21**	**0.21**
z-Score	0.02	0.03	0.03
log-ratio	0.14	0.14	0.13

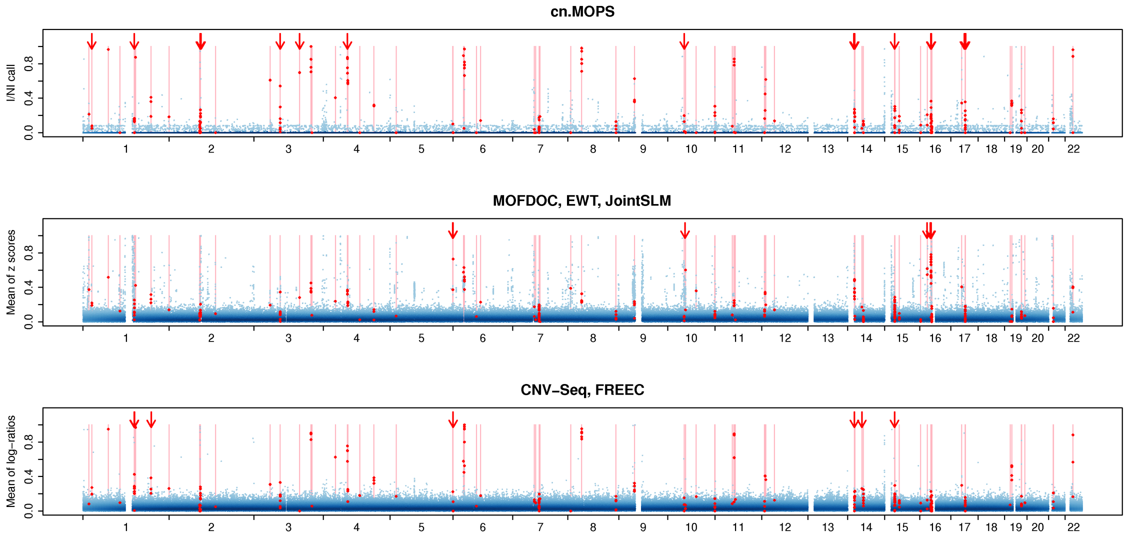


Figure S5: Whole genome CNV calling plots that visualize the performance of cn.MOPS, MOFDOC, EWT, JointSLM , CNV-Seq, and FREEC in rediscovering known CNVs of HapMap individuals. The plots visualize CNV calling values (vertical axis) along chromosomes 1–22 of the human genome without segmentation. The first panel shows the I/NI call of cn.MOPS. The second panel provides mean z-scores used by MOFDOC, EWT, JointSLM, while the last panel depicts mean log-ratios used by CNV-Seq and FREEC. We called the largest 0.5% of the CNV calling values (blue dots) and scaled them to maximum one. Darker shades of blue indicate a high density of calling values. True CNV regions are displayed as light red bars, and the corresponding CNV calls are indicated by red dots. Segments without calling values (white segments) correspond to assembly gaps in the reference genome. A perfect calling method would call all segments in true CNV regions (red dots) at maximum 1 and would call others (blue dots) at minimum 0. Arrows indicate segments in true CNV regions that are called by one method group, but not by the other method groups. cn.MOPS separates segments in true CNV regions from non-CNV segments better than the other methods, as indicated by the lower variance of I/NI values (see blue area at the bottom of the first panel). The better separation by cn.MOPS results in lower FDRs than those of other methods, regardless of the calling thresholds.

S3.6 High Coverage Real World Data Set

This subsection supplies additional information on the data used in the experiments described in Section “High Coverage Real World Data Set” of the main manuscript. The sequencing files were downloaded on October 25, 2011, from the 1000 Genomes Project Web page.[3] Table S11 provides information on filenames, mapped and used reads of the data.

[3] http://www.1000genomes.org

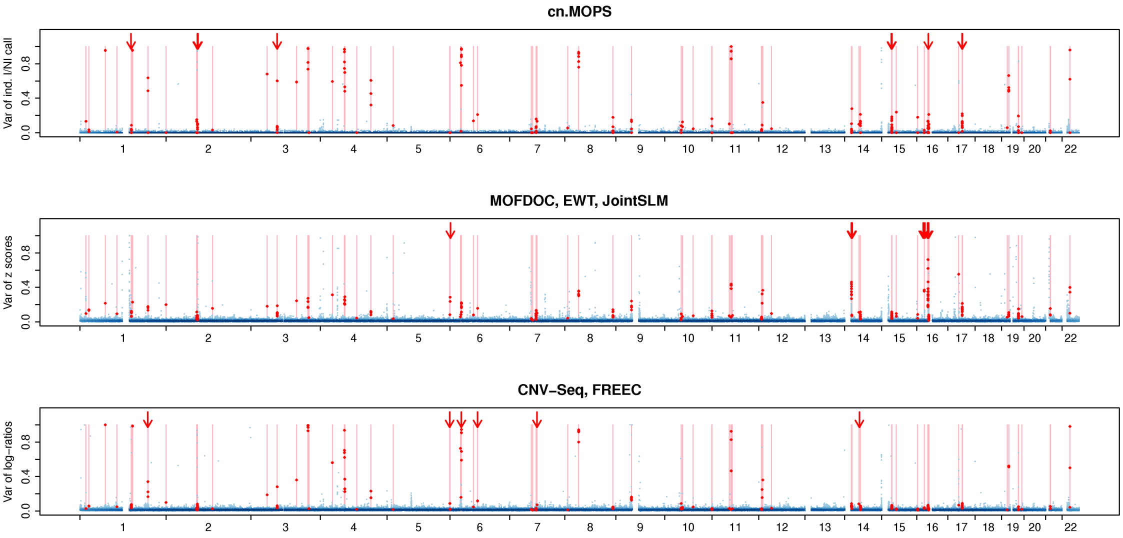


Figure S6: Whole genome CNV calling plots that visualize the performance of cn.MOPS, MOFDOC, EWT, JointSLM , CNV-Seq, and FREEC in rediscovering known CNVs of HapMap individuals. The plots visualize CNV calling values (vertical axis) along chromosomes 1–22 of the human genome without segmentation. The first panel shows the variance of the signed I/NI call of cn.MOPS. The second panel provides variance of the z-scores used by MOFDOC, EWT, JointSLM, while the last panel depicts variance of the log-ratios used by CNV-Seq and FREEC. We called the largest 0.5% of the CNV calling values (blue dots) and scaled them to maximum one. Darker shades of blue indicate a high density of calling values. True CNV regions are displayed as light red bars, and the corresponding CNV calls are indicated by red dots. Segments without calling values (white segments) correspond to assembly gaps in the reference genome. A perfect calling method would call all segments in true CNV regions (red dots) at maximum 1 and would call others (blue dots) at minimum 0. Arrows indicate segments in true CNV regions that are called by one method group, but not by the other method groups. cn.MOPS separates segments in true CNV regions from non-CNV segments better than the other methods, as indicated by the lower variance of I/NI values (see blue area at the bottom of the first panel). The better separation by cn.MOPS results in lower FDRs than those of other methods, regardless of the calling thresholds.

Table S11: Information on the high coverage data set from the 1000 Genomes Project. Column “individual” provides the individual’s identifier, “mapped reads” the number of mapped reads, “used reads” the number of reads that were used, and “filename” the reads’ file name. The numbers in the second and third column differ because the sequence library files contain both single and paired end reads.

individual	mapped reads	used reads	filename
NA12878	258,212,016	178,971,764	`NA12878.chrom1.ILLUMINA.bwa.CEU.high_coverage.20100311.bam`
NA12891	190,462,551	120,566,718	`NA12891.chrom1.ILLUMINA.bwa.CEU.high_coverage.20100517.bam`
NA12892	165,437,943	86,444,082	`NA12892.chrom1.ILLUMINA.bwa.CEU.high_coverage.20100517.bam`
NA19238	127,491,937	72,871,012	`NA19238.chrom1.ILLUMINA.bwa.YRI.high_coverage.20100311.bam`
NA19239	173,589,216	123,354,710	`NA19239.chrom1.ILLUMINA.bwa.YRI.high_coverage.20100311.bam`
NA19240	216,431,838	173,422,058	`NA19240.chrom1.ILLUMINA.bwa.YRI.high_coverage.20100311.bam`

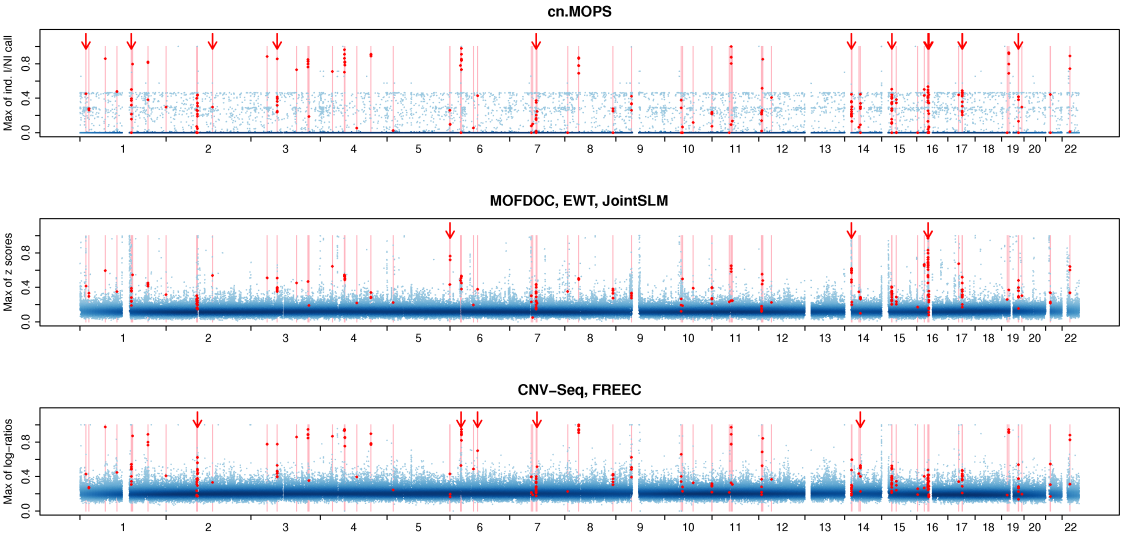


Figure S7: Whole genome CNV calling plots that visualize the performance of cn.MOPS, MOFDOC, EWT, JointSLM , CNV-Seq, and FREEC in rediscovering known CNVs of HapMap individuals. The plots visualize CNV calling values (vertical axis) along chromosomes 1–22 of the human genome without segmentation. The first panel shows the maximum of the absolute value of the signed I/NI call of cn.MOPS. The second panel provides maximum absolute value of the z-scores used by MOFDOC, EWT, JointSLM, while the last panel depicts maximum of the absolute value of the log-ratios used by CNV-Seq and FREEC. We called the largest 0.5% of the CNV calling values (blue dots) and scaled them to maximum one. Darker shades of blue indicate a high density of calling values. True CNV regions are displayed as light red bars, and the corresponding CNV calls are indicated by red dots. Segments without calling values (white segments) correspond to assembly gaps in the reference genome. A perfect calling method would call all segments in true CNV regions (red dots) at maximum 1 and would call others (blue dots) at minimum 0. Arrows indicate segments in true CNV regions that are called by one method group, but not by the other method groups. The discrete nature of the cn.MOPS model which is caused by calls of copy number 0, 1, 3 or larger is revealed in this plot. cn.MOPS separates segments in true CNV regions from non-CNV segments better than the other methods, as indicated by the lower variance of I/NI values (see blue area at the bottom of the first panel). The better separation by cn.MOPS results in lower FDRs than those of other methods, regardless of the calling thresholds.

S3.7 Medium Coverage Data Set

In an additional experiment, we investigated the performance of cn.MOPS for medium coverage data. The data set consists of 58 samples of the 1000 Genomes Project that were sequenced at coverages ranging from 2.5X to 8X. Table S13 provides information on filenames, mapped and used reads of the data.

Fixing a segment length of 2,500bp resulted in 25,211 segments on chromosome 20. The International HapMap 3 Consortium identified two CNVs of type “loss” and 21 of type “gain” after filtering for CNVs longer than 10kbp The International HapMap 3 Consortium (2010). This 10kbp range is the limit for accurate detection for the given coverage according to Xie and Tammi Xie and Tammi (2009). The final data set consisted of 1,462,238 evaluation segments of which 18 are losses and 294 are gains.

We decided to analyze gains only, as there were too few losses for a reliable evaluation. Table S12 presents the results. The recall is low since a lot of newly detected CNVs are ranked higher than the confirmed CNVs. Note, however, that the newly detected CNVs highly overlap between CNVSeq, FREEC and cn.MOPS. CNV-Seq and cn.MOPS performed best for gains. The resulting copy number table is available as a separate file (Supplementary Table S17).

Table S12: Performance of the compared copy number detection methods on the medium coverage data (gains only). "PR AUC" gives the area under the precision-recall curve. "Recall" reports the recall at a precision of 0.95. cn.MOPS and CNV-Seq perform equally well in terms of PR AUC, while cn.MOPS performs best in terms of recall. The performance, however, is generally low.

	PR AUC	Recall
cn.MOPS	**0.48**	**0.07**
MOFDOC	0.00	0.00
EWT	0.00	0.00
JointSLM	0.00	0.00
CNV-Seq	**0.48**	0.03
FREEC	0.41	0.00

S3.8 Influence of the Hyperparameter ϵ on cn.MOPS Results

Using two data sets, we investigated the influence of the choice of ϵ on the performance of cn.MOPS.

S3.8.1 Influence of the Hyperparameter ϵ tested on Simulated Data

The first data set again consists of the simulated data described in Section "Simulated Data with Constructed CNVs" of the main manuscript. Noisy reads for copy number 0 were generated via a Poisson distribution with parameter $\epsilon = 0.05$. Table S14 shows results obtained by cn.MOPS for different choices of ϵ. Different ϵ values only lead to minor changes of the performance, thus the cn.MOPS results are robust against the choice of the hyperparameter ϵ.

Table S14: Performance of cn.MOPS on simulated data with different choices of the hyperparameter ϵ. Different ϵ values only lead to minor changes of the performance, thus the cn.MOPS results are robust against the choice of the hyperparameter ϵ.

	Gains		Losses	
ϵ	PR AUC	Recall	PR AUC	Recall
0.00001	0.94	0.89	0.96	0.96
0.001	0.94	0.89	0.98	0.97
0.01	0.94	0.89	0.97	0.97
0.02	0.94	0.89	0.97	0.97
0.05	0.94	0.88	0.96	0.96
0.1	0.93	0.88	0.96	0.95
0.2	0.93	0.87	0.96	0.95

S3.8.2 Influence of the Hyperparameter ϵ tested on Real Sequencing Data with Implanted CNVs

As a second benchmark, we use the data set described in section "Real Sequencing Data with Implanted CNVs from the X Chromosome" of the main manuscript. Table S15 shows results obtained by cn.MOPS for different choices of ϵ. In this case, the performance does not even depend on ϵ which can be explained easily by the fact that there are no copy number 0 segments in this data set. So we again confirmed that the results of cn.MOPS are robust against the choice of the hyperparameter ϵ.

Table S15: Performance of cn.MOPS on real world benchmarking data with different choices of the hyperparameter ϵ. In this case, ϵ does not influence the results of cn.MOPS at all.

	Gains		Losses	
ϵ	PR AUC	Recall	PR AUC	Recall
0.001	0.70	0.65	0.89	0.88
0.01	0.70	0.65	0.89	0.88
0.02	0.70	0.65	0.89	0.88
0.05	0.70	0.65	0.89	0.88
0.1	0.70	0.65	0.89	0.88
0.2	0.70	0.65	0.89	0.88

S3.9 Number of Samples vs. Performance for cn.MOPS

In order to study the influence of the number of samples on the performance of cn.MOPS, we again generated simulated data as described in the section S3.3, but with varying numbers of samples. Table S16 shows the performance of cn.MOPS for different numbers of samples. At least 6 samples seem to be necessary to ensure sufficient performance for detecting gains, while losses are also detected with fewer samples. For sample numbers larger than 15, the performance saturates.

Table S13: Overview of the medium coverage data set. Column "individual" gives the identifier (the file names are `[identifier].chrom20.ILLUMINA.bwa.CEU.low_coverage.20101123.bam`), "mapped reads" the number of mapped reads contained in the BAM file, and "used reads" the number of reads that were used. Mapped and used reads differ because the sequence library files contained both single and paired end reads.

individual	mapped reads	used reads	individual	mapped reads	used reads
NA06984	6,964,852	5,684,482	NA11993	5,981,863	5,162,066
NA06986	8,039,090	6,883,008	NA11994	4,098,534	3,782,234
NA06989	4,091,782	3,268,000	NA11995	3,757,432	3,416,176
NA06994	4,874,326	4,520,672	NA12003	3,474,069	3,180,788
NA07000	8,645,550	8,086,286	NA12004	5,908,548	4,956,778
NA07037	4,139,953	3,385,920	NA12006	6,922,378	5,092,864
NA07048	10,981,715	8,253,422	NA12043	6,060,516	5,164,746
NA07051	4,964,792	4,337,588	NA12044	5,395,490	4,863,338
NA07056	6,155,442	5,194,792	NA12045	7,799,642	6,532,574
NA07346	6,100,820	4,857,210	NA12046	4,193,666	4,042,864
NA07347	7,261,061	6,804,926	NA12058	3,392,356	3,269,980
NA07357	10,972,910	10,244,264	NA1214	4,214,321	3,899,976
NA10847	5,766,593	4,709,486	NA12154	6,558,385	5,993,336
NA10851	6,302,624	4,650,842	NA12155	11,425,911	10,337,646
NA11829	6,942,857	5,142,718	NA12249	5,607,636	4,764,802
NA11830	4,812,141	4,137,246	NA12272	5,277,117	4,954,532
NA11831	5,337,376	4,737,914	NA12273	4,904,914	4,168,754
NA11843	4,695,305	4,098,448	NA12275	5,248,711	3,861,836
NA11892	4,711,946	4,240,890	NA12282	5,843,315	4,318,018
NA11893	6,593,431	5,118,210	NA12283	6,361,286	5,411,708
NA11894	6,341,119	3,997,266	NA12286	3,616,759	3,314,524
NA11918	6,346,989	5,514,434	NA12287	5,880,103	4,890,098
NA11919	8,961,941	8,327,522	NA12340	4,029,639	3,543,810
NA11920	4,243,883	3,905,198	NA12341	6,851,344	3,569,532
NA11930	6,415,281	5,654,630	NA12342	7,817,761	4,174,086
NA11931	5,243,455	4,030,984	NA12347	5,184,621	4,591,614
NA11932	4,303,807	3,959,018	NA12348	6,175,829	5,053,302
NA11933	11,876,139	7,682,740	NA12383	5,182,815	4,307,720
NA11992	4,517,788	4,141,492	NA12399	9,099,699	5,821,360

Table S16: Number of samples vs. performance for cn.MOPS. At least 6 samples seem to be necessary to ensure sufficient performance for detecting gains, while losses are also detected with fewer samples. For sample numbers larger than 15, the performance saturates.

	Gains		Losses	
samples	PR AUC	Recall	PR AUC	Recall
3	0.38	0.08	0.69	0.53
4	0.59	0.21	0.78	0.66
6	0.74	0.35	0.87	0.76
8	0.85	0.48	0.92	0.84
15	0.90	0.72	0.95	0.94
25	0.93	0.84	0.96	0.96
40	0.94	0.88	0.96	0.96

S3.10 Exemplary DNA Locations With CNV Calls

In this subsection, we exemplify CNV calls of different methods. Each compared method supplies a CNV call value (signed I/NI call, z-scores, or log-ratios) at each evaluation segment. We visualize these CNV calls for different methods at exemplary DNA locations.

S3.10.1 CNV Calls at Exemplary DNA Locations With CNVs

First we visualize CNV calls at exemplary DNA locations with CNVs that were previously found and confirmed by the International HapMap 3 Consortium The International HapMap 3 Consortium (2010). Figures S8, S9, and S10 show CNV calls along with GC-corrected read counts. Each line represents the read counts or CNV calls of a sample across consecutive genomic segments, where green lines indicate losses and red lines indicate gains. For cn.MOPS, the mean signed I/NI call per segment is plotted; for MOFDOC, the return values of the segmentation algorithm are plotted; for EWT, the scaled and signed (positive for gains, negative for losses) log-p-values (transformed z-scores) are plotted; for FREEC and CNV-Seq, the log-ratios per segment are plotted; for JointSLM, the median normalized read counts per segment are plotted. Note that even if methods use the same approach (z-scores or log-ratios), the calling values can be different, because of the different segmentation algorithms they apply.

Figure S8 shows a CNV region that is detected by all methods. Figure S9 shows a CNV region in which only one sample has a deletion that is detected by all methods except JointSLM. The reason for this is that JointSLM only detects variations that appear consistently in the majority of samples. Figure S10 shows a CNV region only detected by cn.MOPS, MOFDOC, and EWT. Note that MOFDOC and EWT would have a high false discovery rate if the detection threshold was chosen low enough to detect the gain. To conclude, the figures show that cn.MOPS produces the most robust and reliable CNV calls.

S3.10.2 CNV Calls at Exemplary DNA Locations Without CNVs

Next, we visualize CNV calls at exemplary DNA locations in which no CNVs have been reported. Figures S11 and S12 again show read counts and CNV calls as in previous figures (see description above). In contrast to previous figures, line colors now represent individual samples.

Figure S11 shows that the class (a) methods MOFDOC, EWT, and JointSLM falsely detect a CNV at this genomic region. The detection is caused by technical or genomic biases in the center of the region shown, in which read counts are consistently larger. FREEC and CNV-Seq are based on ratios, where the bias is removed by normalization using a reference read count. Therefore, they correctly do not detect a CNV. cn.MOPS does not detect a CNV either at this region because the variation across the samples is too low.

The class (b) methods FREEC and CNV-Seq are prone to false detections in regions of low coverage, which is exemplified by Figure S12. cn.MOPS avoids the low coverage problem by fitting a Poisson distribution across samples.

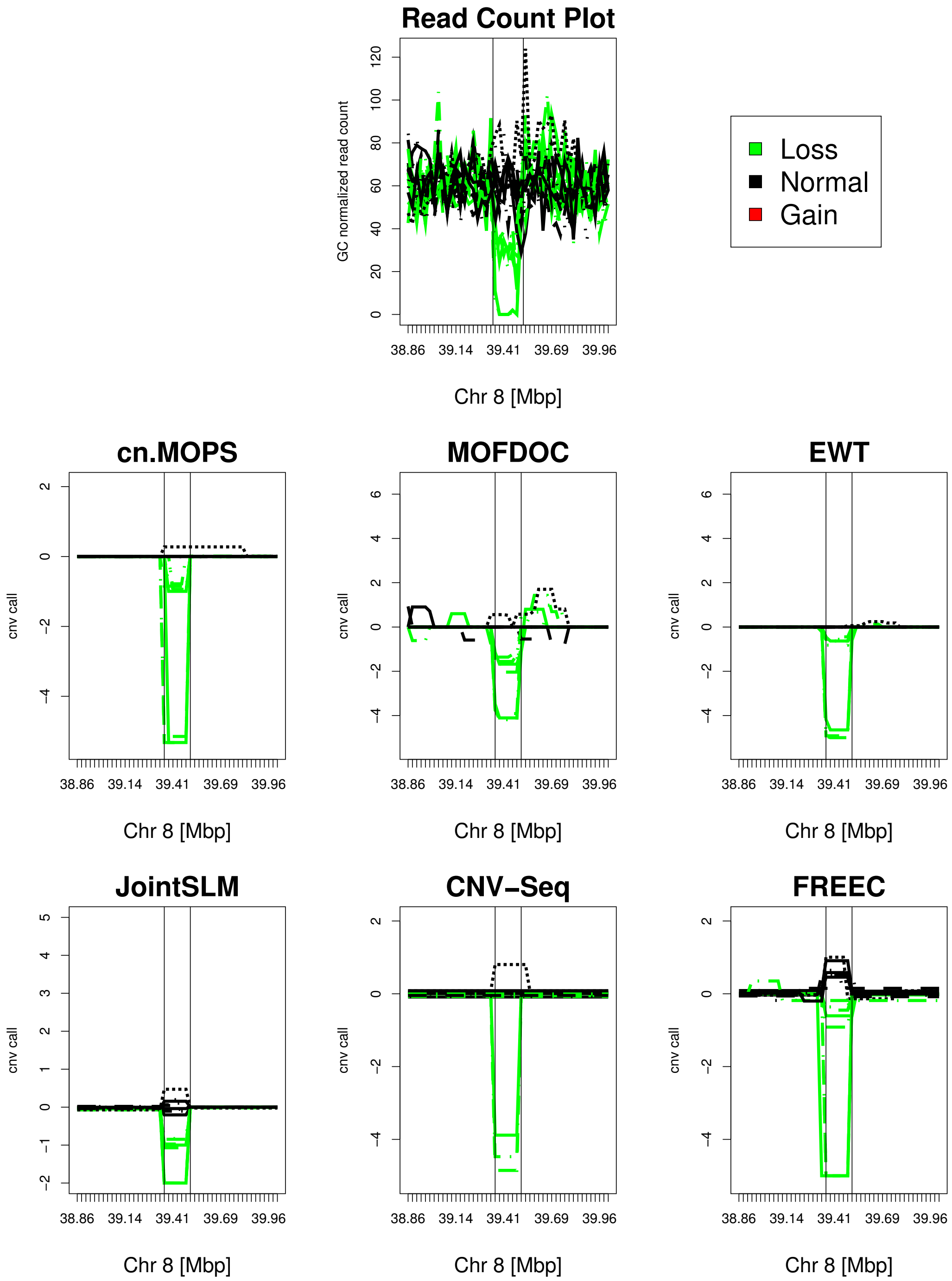


Figure S8: Copy number call plot for CNV region HM3_CNP_463. *Top middle:* read counts of each sample around the CNV region (vertical lines); *middle left:* cn.MOPS' mean signed I/NI call.; *middle:* MOFDOC's smoothed z-scores; *middle right:* EWT's scaled and signed log-p-values (transformed z-scores); *lower left:* JointSLM's median normalized read count; *lower middle:* CNV-Seq's median log-ratio; *lower right:* FREEC's median log-ratio. Each line represents read counts or CNV calls of a sample across consecutive genomic segments; green lines indicate a loss and red lines a gain. All methods detected this loss region.

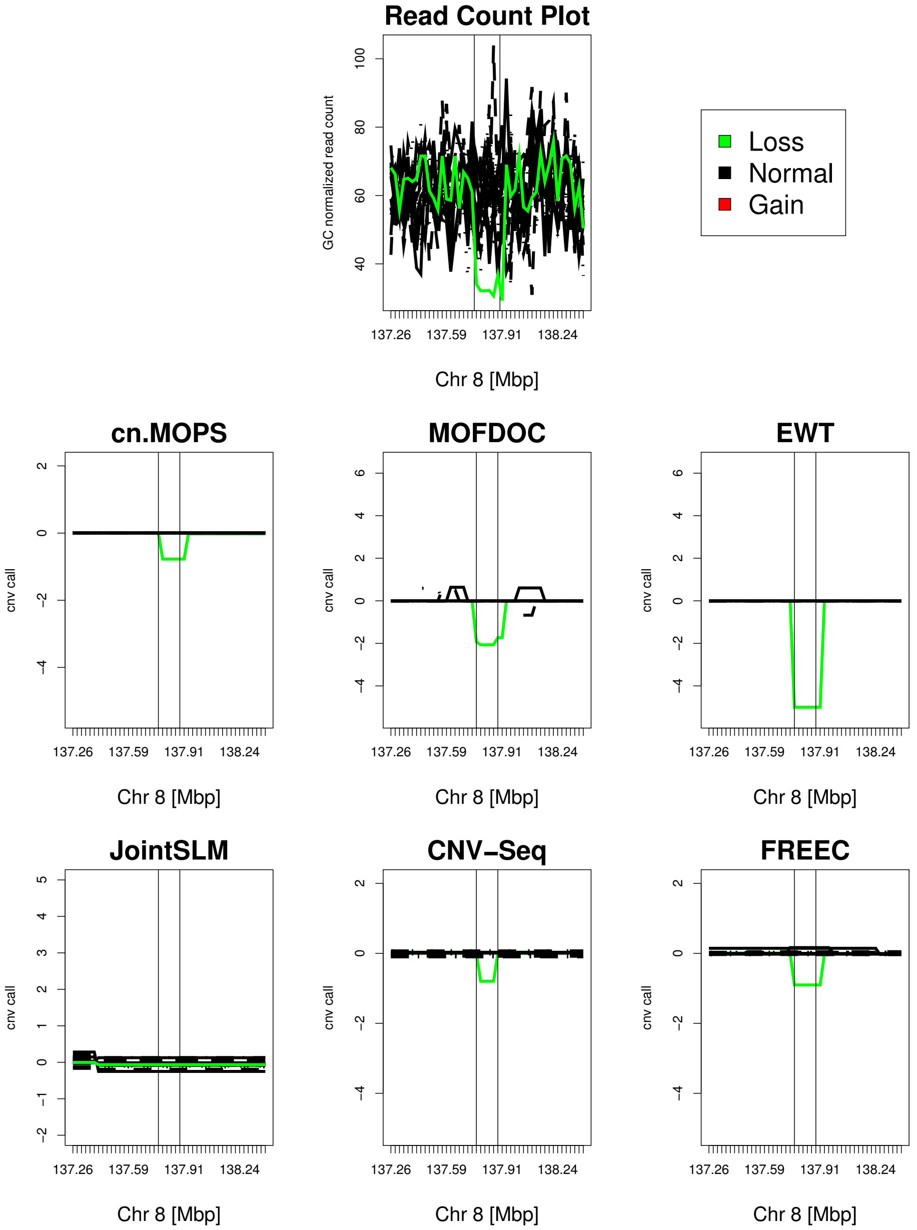


Figure S9: Copy number call plot for CNV region HM3_CNP_494. *Top middle:* read counts of each sample around the CNV region (vertical lines); *middle left:* cn.MOPS' mean signed I/NI call.; *middle:* MOFDOC's smoothed z-scores; *middle right:* EWT's scaled and signed log-p-values (transformed z-scores); *lower left:* JointSLM's median normalized read count; *lower middle:* CNV-Seq's median log-ratio; *lower right:* FREEC's median log-ratio. Each line represents read counts or CNV calls of a sample across consecutive genomic segments; green lines indicate a loss and red lines a gain. JointSLM did not detect the CNV segment, as it is only able to detect only variations that appear consistently in the majority of samples.

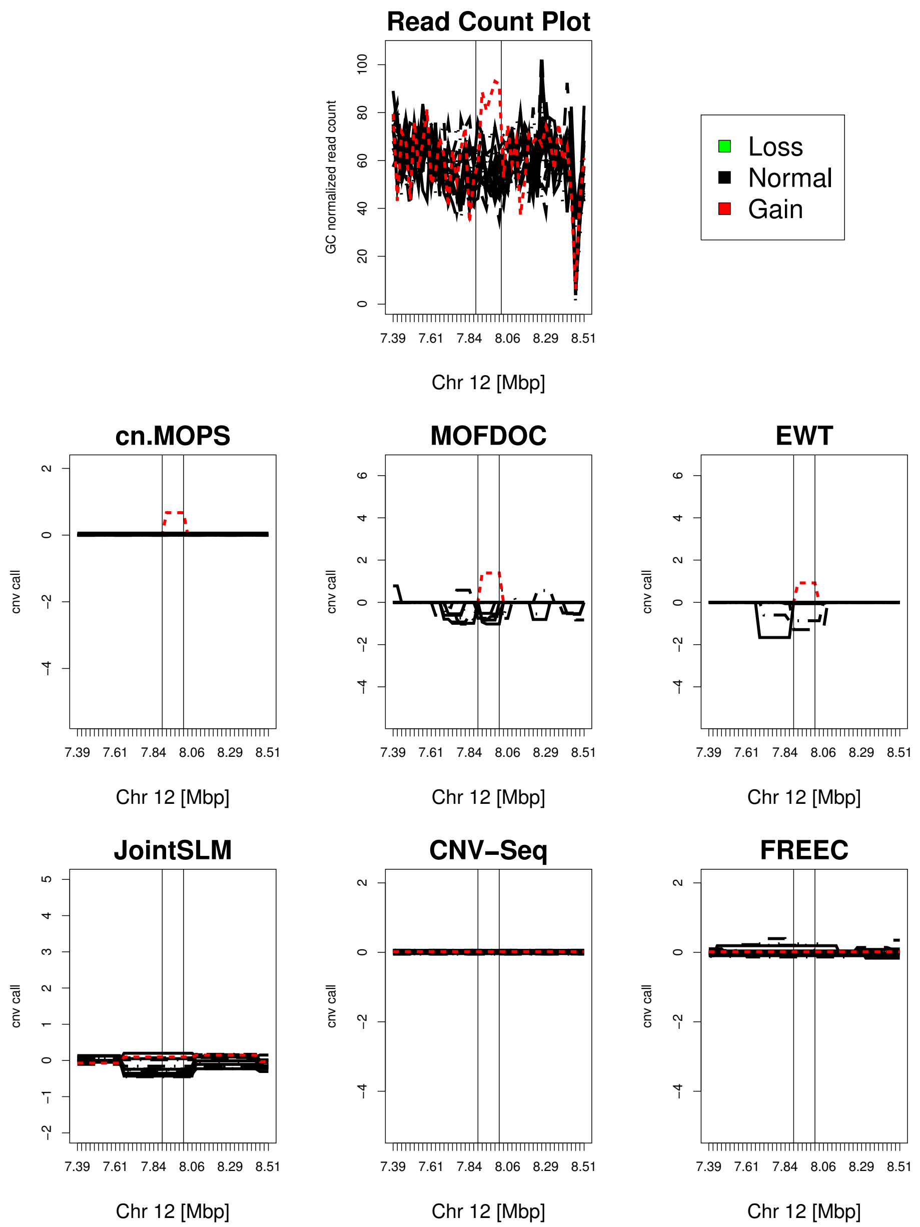


Figure S10: Copy number call plot for CNV region HM3_CNP_618. *Top middle:* read counts of each sample around the CNV region (vertical lines); *middle left:* cn.MOPS' mean signed I/NI call.; *middle:* MOFDOC's smoothed z-scores; *middle right:* EWT's scaled and signed log-p-values (transformed z-scores); *lower left:* JointSLM's median normalized read count; *lower middle:* CNV-Seq's median log-ratio; *lower right:* FREEC's median log-ratio. Each line represents read counts or CNV calls of a sample across consecutive genomic segments; green lines indicate a loss and red lines a gain. Only cn.MOPS and maybe MOFDOC detect this gain in one sample. Note that MOFDOC would have a high false discovery rate if the detection threshold was chosen low enough to detect the gain.

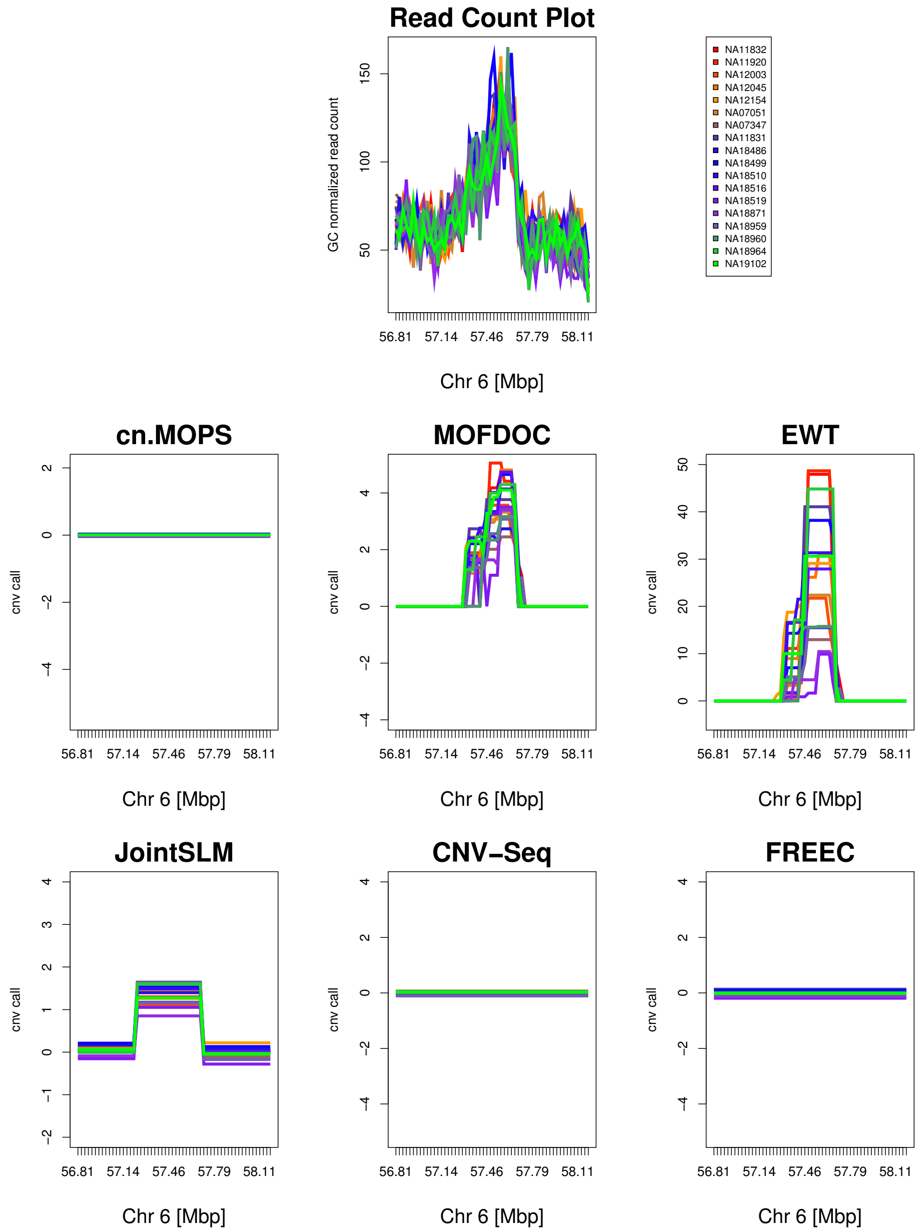


Figure S11: Copy number call plot for the region 56.81Mbp–58.01Mbp on chromosome 6 of the human reference genome 18 (build 36). Different colors represent different samples. *Top middle:* read counts of each sample around the CNV region (vertical lines); *middle left:* cn.MOPS' mean signed I/NI call.; *middle:* MOFDOC's smoothed z-scores; *middle right:* EWT's scaled and signed log-p-values (transformed z-scores); *lower left:* JointSLM's median normalized read count; *lower middle:* CNV-Seq's median log-ratio; *lower right:* FREEC's median log-ratio. The class (a) methods MOFDOC, EWT, and JointSLM falsely detect a CNV in this genomic region. cn.MOPS and the class (b) methods FREEC and CNV-Seq do not detect a CNV in this region.

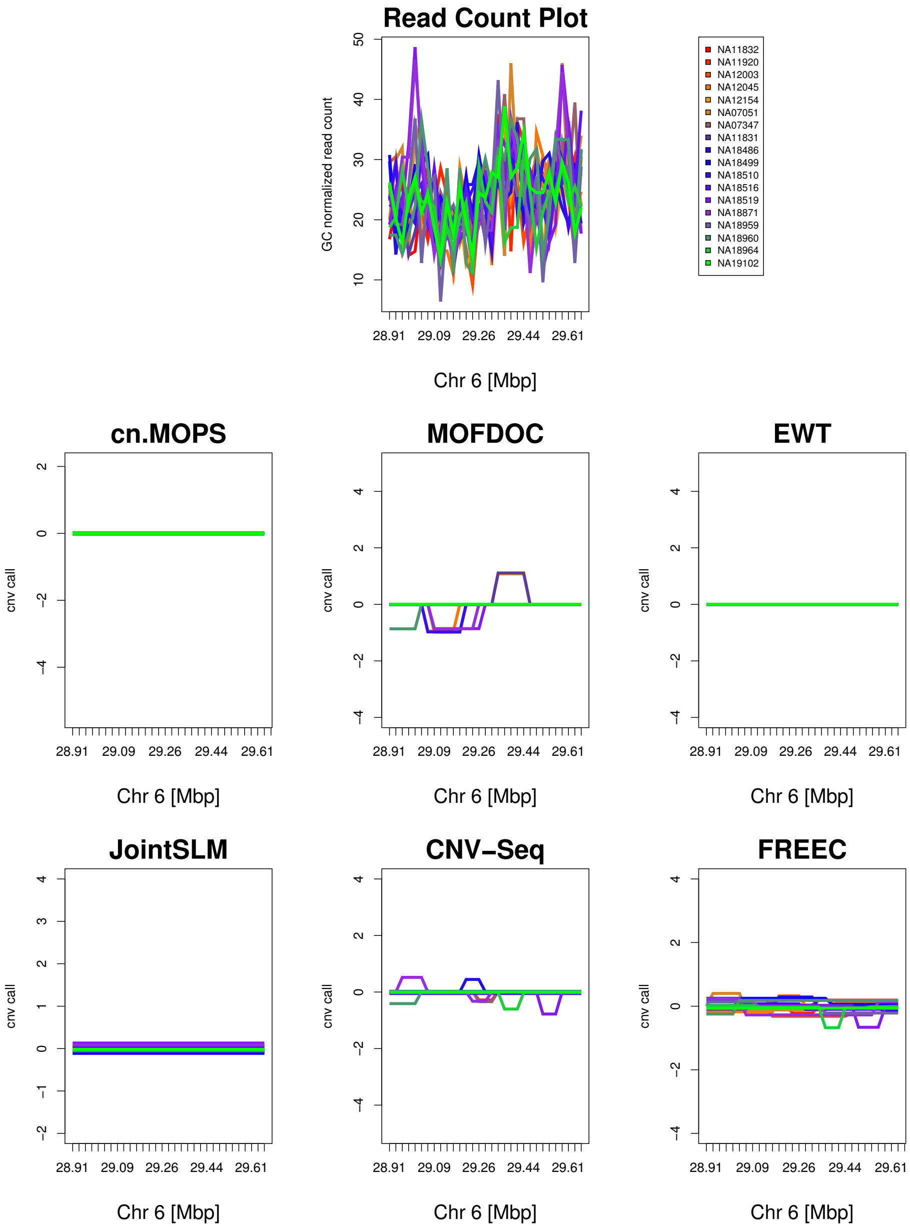


Figure S12: Copy number call plot for the region 28.91Mbp–9.95Mbp on chromosome 6 of the human reference genome 18 (build 36). Different colors represent different samples. *Top middle:* read counts of each sample around the CNV region (vertical lines); *middle left:* cn.MOPS' mean signed I/NI call.; *middle:* MOFDOC's smoothed z-scores; *middle right:* EWT's scaled and signed log-p-values (transformed z-scores); *lower left:* JointSLM's median normalized read count; *lower middle:* CNV-Seq's median log-ratio; *lower right:* FREEC's median log-ratio. The class (b) methods FREEC and CNV-Seq falsely detect CNVs in this CNV-free region. cn.MOPS avoids the low coverage problem by fitting a Poisson distribution across samples.

S4 Additional Information

This section is divided into two subsections. The first subsection investigates the read count distribution along a chromosome. The second subsection gives information on how read counts are summarized in a data structure for further processing by CNV detection methods.

S4.1 Distribution of Read Counts Along the Chromosome

The distribution of read counts of equally sized segments along the chromosome is not Poisson distributed even upon GC correction (Dohm *et al.* 2008). We confirmed the result in (Dohm *et al.* 2008).

We found that for segments with a length of 10kbp, 25kbp and 50kbp, the GC corrected read counts have a variance-to-mean ratio larger than 1. For example, on data from the Sanger sequencing center on HapMap phase 1 individuals for segments of 25kbp the variance-to-mean ratio of GC corrected reads was 2.11, after removing sequencing gaps and outliers along the chromosome (read counts larger than two times the median read count). Note, that outliers would even increase the ratio. This ratio larger than 1 contradicts the assumption of a Poisson distribution which would lead to a ratio of 1. Actually, the read counts approximately follow a Gaussian distribution.

The histogram in Fig. S13 for non-uniquely read mapping (see Subsection S3.4.1) shows that the GC corrected read counts are not Poisson distributed. Thus, effects other than the GC bias lead to different average read counts at different genomic segments. The biases cannot be avoided by different mapping strategies like mapping only unique positions (see Subsection S3.4.1) as shown in Fig. S14. Compared to the histogram in Fig. S13, we observe a shift of the density toward lower read counts (left) in the histogram in Fig. S14, because some segments systematically loose reads due to ambiguous mapping.

S4.2 Data Structure of Read Counts

Next generation sequencing (NGS) data for copy number detection or estimation is in most cases represented as a read count matrix $\boldsymbol{Z} \in \mathbb{N}^{L \times N}$, where the genome is partitioned into L segments of not necessarily equal length for N samples. Therefore $z_{lk} \in \mathbb{N}$ represents the number of reads of sample k that are mapped to the l-th segment. Note, that in previous sections we considered only one segment l with read counts $x_k = z_{lk}$. Copy number detection methods applied to such a read count matrix $\boldsymbol{Z}$ are often called "depth of coverage"-based methods. $\boldsymbol{z}^k$ is the k-th column of $\boldsymbol{Z}$, which is the read count vector of sample k. $\boldsymbol{z}_l$ is l-th row of the read count matrix, which is the vector containing read counts for the l-th genomic segment for all samples k with $1 \leq k \leq N$. Note, that here we see a substantial novel approach of the cn.MOPS model which uses $\boldsymbol{z}_l = (z_{l1}, \ldots, z_{lk}, \ldots, z_{lN}) = (x_1, \ldots, x_k, \ldots, x_N)$ for modeling across samples, while other methods use $\boldsymbol{z}^k$ to find variations along the chromosome. Fig. S1 depicts entries of the matrix $\boldsymbol{Z}$ by connected by lines and shows modeling along the chromosome and modeling across samples (vertical green boxes).

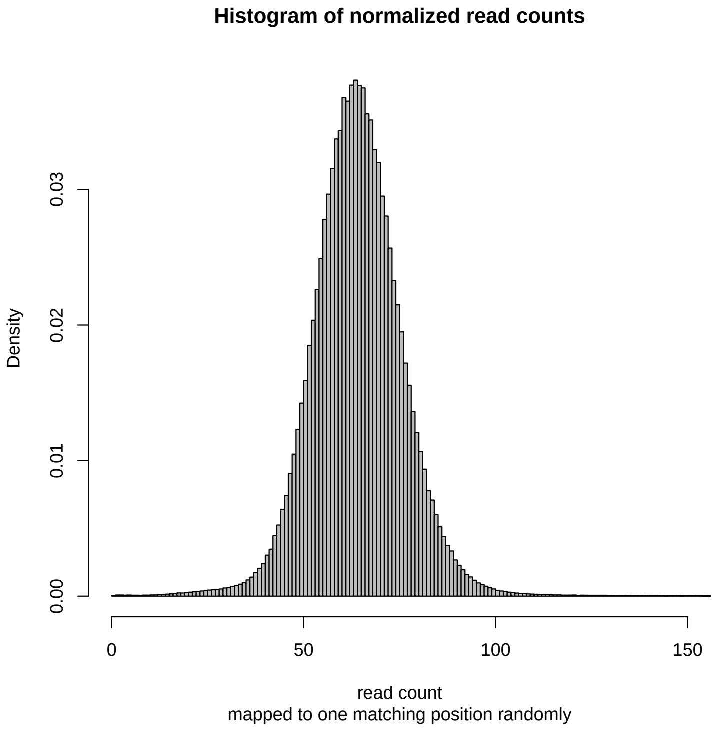


Figure S13: Histogram for non-uniquely mapped reads of GC corrected read counts from 18 HapMap samples sequenced at the Sanger sequencing center (see Subsection S3.5). Reads are mapped to a random position if more than one best matching position is available.

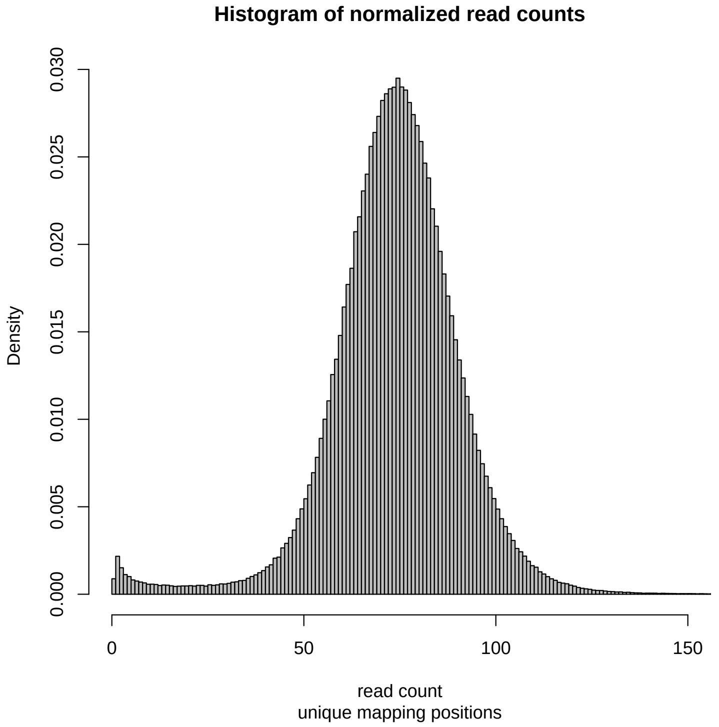


Figure S14: Histogram for uniquely mapped reads (see Subsection S3.4.1) of the GC corrected read counts from 18 HapMap samples sequenced at the Sanger sequencing center (see Subsection S3.5). Reads with multiple maps to the genome are not regarded. Compared to the histogram in Fig. S13, we observe a shift of the density toward lower read counts (left) because some segments systematically lose reads due to ambiguous mapping.

References

Alkan, C., Kidd, J. M., Marques-Bonet, T., Aksay, G., Antonacci, F., Hormozdiari, F., Kitzman, J. O., Baker, C., Malig, M., Mutlu, O., Sahinalp, S. C., Gibbs, R. A., and Eichler, E. E. (2009). Personalized copy number and segmental duplication maps using next-generation sequencing. *Nat. Genet.*, **41**(10), 1061–1067.

Boeva, V., Zinovyev, A., Bleakley, K., Vert, J.-P., Janoueix-Lerosey, I., Delattre, O., and Barillot, E. (2011). Control-free calling of copy number alterations in deep-sequencing data using GC-content normalization. *Bioinformatics*, **27**(2), 268–269.

Brown, L. and Zhao, L. (2002). A new test for the Poisson distribution. *Sankhya Ser. A*, **64**, 611–625.

Chiang, D. Y., Getz, G., Jaffe, D. B., Zhao, X., Carter, S. L., Russ, C., Nusbaum, C., Meyerson, M., and Lander, E. S. (2008). High-resolution mapping of copy-number alterations with massively parallel sequencing. *Nat. Methods*, **6**, 99–103.

Clevert, D.-A., Mitterecker, A., Mayr, A., Klambauer, G., Tuefferd, M., Bondt, A. D., Talloen, W., Göhlmann, H., and Hochreiter, S. (2011). cn.FARMS: a latent variable model to detect copy number variations in microarray data with a low false discovery rate. *Nucleic Acids Res.*, **39**(12), e79.

Dohm, J. C., Lottaz, C., Borodina, T., and Himmelbauer, H. (2008). Substantial biases in ultra-short read data sets from high-throughput dna sequencing. *Nucleic Acids Res.*, **36**(16), e105.

Hochreiter, S., Clevert, D.-A., and Obermayer, K. (2006). A new summarization method for Affymetrix probe level data. *Bioinformatics*, **22**(8), 943–949.

Ivakhno, S., Royce, T., Cox, A. J., Evers, D. J., Cheetham, R. K., and TavarÃl', S. (2010). CNAseg–a novel framework for identification of copy number changes in cancer from second-generation sequencing data. *Bioinformatics*, **26**(24), 3051–3058.

Langmead, B., Trapnell, C., Pop, M., and Salzberg, S. (2009). Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biol.*, **10**(3), R25.

Magi, A., Benelli, M., Yoon, S., Roviello, F., and Torricelli, F. (2011). Detecting common copy number variants in high-throughput sequencing data by using JointSLM algorithm. *Nucleic Acids Res.*, **39**(10), e65.

Talloen, W., Clevert, D.-A., Hochreiter, S., Amaratunga, D., Bijnens, L., Kass, S., and Göhlmann, H. (2007). I/NI-calls for the exclusion of non-informative genes: a highly effective filtering tool for microarray data. *Bioinformatics*, **23**(21), 2897–2902.

Talloen, W., Hochreiter, S., Bijnens, L., Kasim, A., Shkedy, Z., and Amaratunga, D. (2010). Filtering data from high-throughput experiments based on measurement reliability. *Proc. Natl. Acad. Sci. U.S.A.*, **107**(46), 173–174.

The 1000 Genomes Project Consortium (2010). A map of human genome variation from population-scale sequencing. *Nature*, **467**(7319), 1061–1073.

The International HapMap 3 Consortium (2010). Integrating common and rare genetic variation in diverse human populations. *Nature*, **467**(7311), 52–58.

Venkatraman, E. S. and Olshen, A. B. (2007). A faster circular binary segmentation algorithm for the analysis of array CGH data. *Bioinformatics*, **23**(6), 657–663.

Xie, C. and Tammi, M. T. (2009). CNV-Seq, a new method to detect copy number variation using high-throughput sequencing. *BMC Bioinformatics*, **10**, 80.

Yoon, S., Xuan, Z., Makarov, V., Ye, K., and Sebat, J. (2009). Sensitive and accurate detection of copy number variants using read depth of coverage. *Genome Res.*, **19**(9), 1586–1592.