VanillaICE: Hidden Markov Models for the Assessment of Chromosomal Alterations using High-throughput SNP Arrays

Robert Scharpf

July 16, 2009

Abstract

Chromosomal DNA is characterized by variation between individuals at the level of entire chromosomes (e.g. aneuploidy in which the chromosome copy number is altered), segmental changes (including insertions, deletions, inversions, and translocations), and changes to small genomic regions (including single nucleotide polymorphisms). A variety of alterations that occur in chromosomal DNA, many of which can be detected using high density single nucleotide polymorphism (SNP) microarrays, are linked to normal variation as well as disease and therefore of particular interest. These include changes in copy number (deletions and duplications) and genotype (e.g. the occurrence of regions of homozygosity). Hidden Markov models (HMM) are particularly useful for detecting such abnormalities, modeling the spatial dependence between neighboring SNPs. Here, we extend previous approaches that utilize HMM frameworks for inference in high throughput SNP arrays by integrating copy number, genotype calls, and the corresponding measures of uncertainty when available. Using simulated and experimental data, we demonstrate how confidence scores control smoothing in a probabilistic framework.

1 Overview

This vignette describes how to fit a hidden Markov model to locus-level estimates of genotype or copy number. This vignette requires that you have

- an absolute estimate of the *total* copy number organized such that rows correspond to loci and columns correspond to samples

  and / or

- a matrix of genotype calls (1=AA, 2 = AB, 3= BB): rows correspond to loci and columns correspond to samples

Additional options can improve the HMM predictions include

- a CRLMM / oligo confidence score of the genotype call
- standard errors of the copy number estimates

If using the R package *crlmm*, see the the vignette copynumber.Rnw for locus-level estimation of copy number and suggestions for fitting a hidden Markov model to allele-specific estimates of copy number. Note that several HMM implementations are now available for the joint analysis of copy number and genotype, including QuantiSNP [2] and PennCNV [5].

Citing this software. Robert B Scharpf, Giovanni Parmigiani, Jonathan Pevsner, and Ingo Ruczinski. Hidden Markov models for the assessment of chromosomal alterations using high-throughput SNP arrays. *Annals of Applied Statistics*, 2(2):687–713, 2008.

2 Simple Usage

* Before beginning, verify that it is reasonable to assume integer copy number by plotting the locus-level estimates as a function of the physical position.

Assuming that a integer copy number hidden state is reasonable, we create an object of class `oligoSnpSet` from the *simulated* data available provided in this package:

```
> library(VanillaICE)
> library(Biobase)
> library(oligoClasses)
> data(locusLevelData)
> copynumber <- locusLevelData[["copynumber"]]/100
> chromosome <- locusLevelData[["annotation"]][, "chromosome"]
> position <- locusLevelData[["annotation"]][, "position"]
> names(position) <- names(chromosome) <- rownames(locusLevelData[["annotation"]])
> locusAnnotation <- data.frame(list(chromosome = chromosome, position = position),
+     row.names = names(chromosome))
> featuredata <- new("AnnotatedDataFrame", data = locusAnnotation,
+     varMetadata = data.frame(labelDescription = colnames(locusAnnotation)))
> locusset <- new("oligoSnpSet", copyNumber = copynumber, calls = locusLevelData[["genotypes"]],
+     callsConfidence = locusLevelData[["crlmmConfidence"]], phenoData = annotatedDataFrameFrom(copynu
+         byrow = FALSE), featureData = featuredata, annotation = locusLevelData[["platform"]])
> stopifnot(all(c("chromosome", "position") %in% varLabels(featuredata)))
> locusset <- locusset[order(chromosome(locusset), position(locusset)),
+     ]
```

The following components are required to fit the HMM:

1. initial state probabilities
2. emission probabilities
3. transition probabilities

Several of the functions in the VanillaICE package are wrappers to facilitate the specification of these arguments. In the following code chunk, we assume the hidden states are hemizygous deletion (hemDel; copy number = 1, probability of a homozygous genotype call is 0.999), normal (copy number = 2, probability of a homozygous genotype calls is 0.7), regions of homozygosity (ROH: copy number = 1, probability of a homozygous genotype call is 0.999), and amplification (copy number greater than 2).

```
> joint.states <- c("hemDel", "normal", "ROH", "amplification")
> copynumber.states <- log2(c(1, 2, 2, 3))
> prob.genotype.is.homozygous <- c(0.999, 0.7, 0.999, 0.7)
> names(prob.genotype.is.homozygous) <- joint.states
> initialP <- (rep(1, length(joint.states)))/length(joint.states)
> tau <- transitionProbability(chromosome = chromosome, position = position,
+     TAUP = 1e+08)
> log.sds <- VanillaICE:::robustSds(copyNumber(locusset))
```

Conditional on the underlying hidden state, we assume that the copy number is independent of the genotype and we calculate the emission probabilities of each separately.

```
> emission.cn <- copynumberEmission(copynumber = log2(copyNumber(locusset)),
+     states = joint.states, mu = copynumber.states, sds = log.sds,
+     takeLog = FALSE, verbose = TRUE)
> emission.cn[1:5, , ]
```

```
                 hemDel      normal         ROH amplification
SNP_A-1677174 -2.9411591 -0.03977016 -0.03977016     -1.032571
SNP_A-1718890 -1.7779061 -0.18796632 -0.18796632     -1.947915
SNP_A-1678466 -0.2405088 -1.61556755 -1.61556755     -5.109930
SNP_A-1676440 -0.1173296 -2.06997172 -2.06997172     -5.902198
SNP_A-1662392 -0.7823100 -0.74812620 -0.74812620     -3.418134
```

```
> emission.gt <- genotypeEmission(genotypes = calls(locusset),
+     states = joint.states, probHomCall = prob.genotype.is.homozygous)
```

As the emission probabilities returned by the above functions are on the log scale, we simply add the emission probabilties to obtain the joint emission probabilities:

```
> emission.joint <- emission.gt + emission.cn
```

The sequence of states that maximizes the likelihood is obtained from the `viterbi` algorithm:

```
> fit1 <- viterbi(initialStateProbs = log(initialP), emission = emission.joint,
+     tau = tau[, "transitionPr"], arm = tau[, "arm"], normalIndex = 2)
> (brks <- breaks(x = fit1, states = joint.states, position = tau[,
+     "position"], chromosome = tau[, "chromosome"]))
```

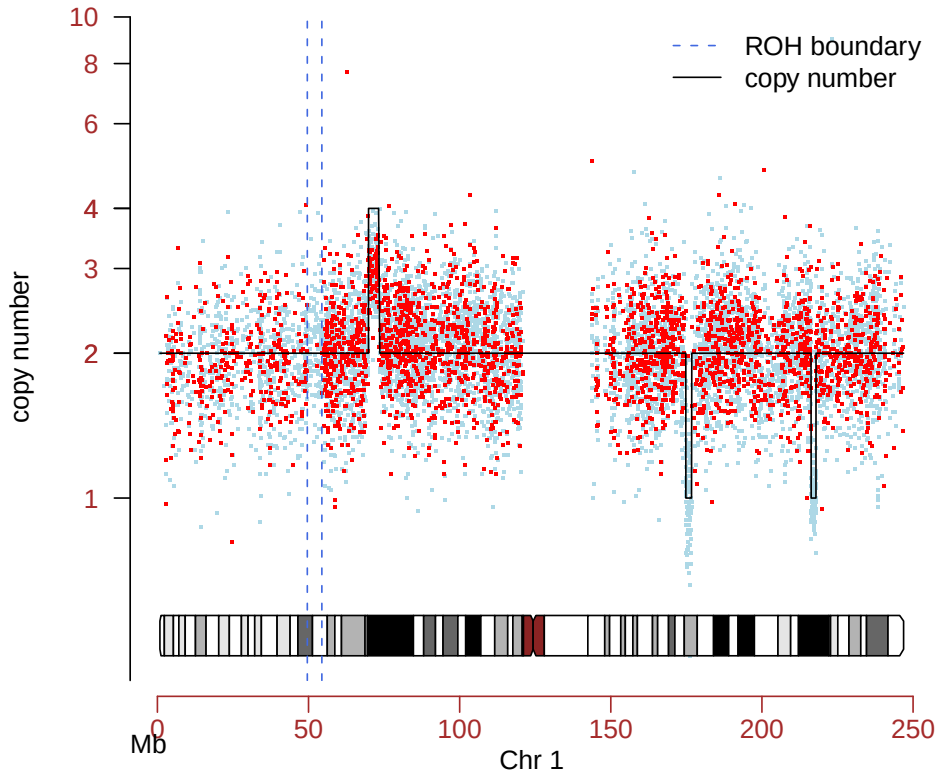
```
    sample chr     start       end   nbases nprobes         state
1  NA06993   1    836727  49545039 48708313     997        normal
2  NA06993   1  49597810  54409755  4811946     102           ROH
3  NA06993   1  54498950  69838068 15339119     902        normal
4  NA06993   1  69854466  73174389  3319924     204 amplification
5  NA06993   1  73577300 120928505 47351206    2500        normal
6  NA06993   1 143619946 174815096 31195151    1296        normal
7  NA06993   1 174828535 176704067  1875533      97        hemDel
8  NA06993   1 176800399 216239917 39439519    1902        normal
9  NA06993   1 216286002 217872810  1586809      99        hemDel
10 NA06993   1 217892177 246860994 28968818    1066        normal
11 NA06993   2    836727   7285897  6449171     100        normal
```

```
> rohBoundary <- as.integer(as.matrix(brks[brks$state == "ROH",
+     ][c("start", "end")]))
```

A plot of the locus-level data with predicted states overlaid:

```
> require(SNPchip)
> gp <- plot(locusset[chromosome(locusset) == 1, ])
> cns <- fit1
> cns[cns == 3] <- 2
> gp$abline.v <- TRUE
> gp$col.predict[3] <- "white"
> gp$hmm.ycoords <- c(0.7, 0.9)
> show(gp)
> lines(position(locusset)[chromosome(locusset) == 1], cns[chromosome(locusset) ==
```

```
+     1, ])
> abline(v = rohBoundary, col = "royalblue", lty = 2)
> legend("topright", lty = c(2, 1), col = c("royalblue", "black"),
+     legend = c("ROH boundary", "copy number"), bty = "n")
```



3 Additional options

3.1 CRLMM confidence scores for the genotypes

The HMM in Section 2 ignores the CRLMM confidence estimates that are available for the genotype calls. The following platforms are currently supported:

```
> path <- system.file("extdata", package = "VanillaICE")
> supportedPlatforms <- list.files(path)[grep("Conf", list.files(path))]
> as.character(sapply(supportedPlatforms, function(x) strsplit(x,
+     "Conf")[[1]][[1]]))
```

```
[1] "genomewidesnp6"     "pd.mapping250k.nsp" "pd.mapping250k.sty"
```

The emission probabilities for the genotypes are computed by calling the `genotypeEmissionCrlmm` function. In the simulated dataset used for this vignette, two heterozygous genotype calls were located in the middle of the ROH region.

```
> hetIndices <- which(calls(locusset) == 2 & fit1 == 3)
> confs <- round(1 - exp(-callsConfidence(locusset)[hetIndices]/1000),
+     5)
```

The confidence scores for these two SNPs are 0.99738 and 0.99979. Because these two heterozygous SNPs were genotyped with high confidence, the HMM will be less likely to call the region homozygous as evidenced by the higher emission probability for the normal state at the above loci:

```
> emission.crlmm <- genotypeEmissionCrlmm(genotypes = calls(locusset),
+     conf = callsConfidence(locusset), annotation = annotation(locusset),
+     pHetCalledHom = 0.001, pHetCalledHet = 0.995, pHomInNormal = 0.8,
+     pHomInRoh = 0.999)
> emission.crlmm[hetIndices, , ]

                   norm       ROH
SNP_A-1711289 3.635091 -4.670975
SNP_A-1752263 5.837202 -4.717411
```

We recompute the HMM predictions as follows:

```
> joint.emission2 <- emission.cn
> joint.emission2[, , c("normal", "amplification")] <- joint.emission2[,
+     , c("normal", "amplification")] + emission.crlmm[, , "norm"]
> joint.emission2[, , c("hemDel", "ROH")] <- joint.emission2[,
+     , c("hemDel", "ROH")] + emission.crlmm[, , "ROH"]
> fit2 <- viterbi(initialStateProbs = log(initialP), emission = joint.emission2,
+     tau = tau[, "transitionPr"], arm = tau[, "arm"], normalIndex = 2)
> table(fit2)

fit2
   1    2    4
 196 8865  204

> breaks(x = fit2, states = joint.states, position = tau[, "position"],
+     chromosome = tau[, "chromosome"])

   sample chr     start       end   nbases nprobes         state
1 NA06993   1    836727  69838068 69001342    2001        normal
2 NA06993   1  69854466  73174389  3319924     204 amplification
3 NA06993   1  73577300 120928505 47351206    2500        normal
4 NA06993   1 143619946 174815096 31195151    1296        normal
5 NA06993   1 174828535 176704067  1875533      97        hemDel
6 NA06993   1 176800399 216239917 39439519    1902        normal
7 NA06993   1 216286002 217872810  1586809      99        hemDel
8 NA06993   1 217892177 246860994 28968818    1066        normal
9 NA06993   2    836727   7285897  6449171     100        normal
```

Note that after incorporating the confidence scores, the previously called ROH region is now called *normal*.

The following two sections describe how to fit the HMM to copy number-only or diallelic genotype-only data.

3.2 Copy number only

```
> states <- 0:5
> mus <- log2(c(0.05, 1:5))
> emission.cn <- copynumberEmission(copynumber = log2(copyNumber(locusset)),
+     states = states, mu = mus, sds = log.sds, takeLog = FALSE,
+     verbose = TRUE)
> initialP <- log(rep(1/length(states), length(states)))
> fit3 <- viterbi(initialStateProbs = initialP, emission = emission.cn,
+     tau = tau[, "transitionPr"], arm = tau[, "arm"], normalIndex = 3)
> breaks(x = fit3, states = states, position = tau[, "position"],
+     chromosome = tau[, "chromosome"])
```

```
   sample chr     start       end   nbases nprobes state
1 NA06993   1    836727  69838068 69001342    2001     2
2 NA06993   1  69854466  73174389  3319924     204     3
3 NA06993   1  73577300 120928505 47351206    2500     2
4 NA06993   1 143619946 174815096 31195151    1296     2
5 NA06993   1 174828535 176800844  1972310     100     1
6 NA06993   1 176926556 216239917 39313362    1899     2
7 NA06993   1 216286002 217872810  1586809      99     1
8 NA06993   1 217892177 246860994 28968818    1066     2
9 NA06993   2    836727   7285897  6449171     100     2
```

For homozygous deletion, I just specified a small number. An empirical estimate of 'zero copy number' could be obtained from chromosome Y for females.

3.3 Genotypes only

Note the hemizygous deletions are called 'ROH' when copy number is ignored:

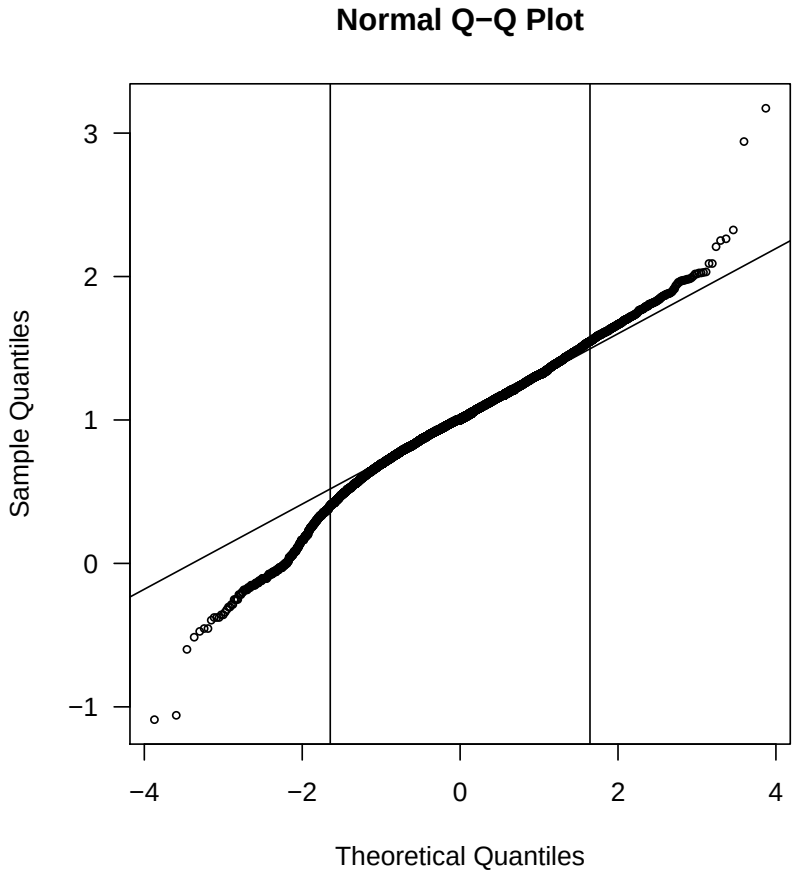
```
> states <- c("normal", "ROH")
> prob.genotype.is.homozygous <- c(0.7, 0.999)
> emission.gt <- genotypeEmission(genotypes = calls(locusset),
+     states = states, probHomCall = prob.genotype.is.homozygous)
> initialP <- log(rep(1/length(states), length(states)))
> fit4 <- viterbi(initialStateProbs = initialP, emission = emission.gt,
+     tau = tau[, "transitionPr"], arm = tau[, "arm"], normalIndex = 1)
> breaks(x = fit4, states = states, position = tau[, "position"],
+     chromosome = tau[, "chromosome"])
```

```
   sample chr     start       end   nbases nprobes  state
1 NA06993   1    836727  49545039 48708313     997 normal
2 NA06993   1  49597810  54409755  4811946     102    ROH
3 NA06993   1  54498950 120928505 66429556    3606 normal
4 NA06993   1 143619946 174309008 30689063    1284 normal
5 NA06993   1 174418117 176704067  2285951     109    ROH
6 NA06993   1 176800399 216239917 39439519    1902 normal
7 NA06993   1 216286002 217771828  1485827      97    ROH
8 NA06993   1 217872013 246860994 28988982    1068 normal
9 NA06993   2    836727   7285897  6449171     100 normal
```

3.4 Modeling assumptions

The emission probabilities are calculated under the assumption that the estimates, or a suitable transformation, are approximately Gaussian. Hence, in the simulated data we check that the middle 90% is approximately Gaussian.

```
> par(pty = "s", las = 1)
> qqnorm(log2(copyNumber(locusset)), cex = 0.6)
> qqline(log2(copyNumber(locusset)))
> abline(v = c(-1.645, 1.645))
```



3.5 Copy number outliers

When true uncertainty estimates for the copy number are not available, copy number outliers are more likely to result in extreme emission probabilities than can influence the HMM segmentation. There are several approaches to reducing the influence of outliers on the HMM segmentation: (i) improved uncertainty estimates (preferred– see the *crlmm* package), (ii) increase `TAUP` for the transition probabilities, (iii) threshold the emission probabilities, and (iv) threshold extreme values for the copy number estimates. For example of (iii), one could do

```
> emission.cn[emission.cn[, , "normal"] < -10, , "normal"] <- -10
```

`copynumberEmission` estimates the scale parameter for the Normal distribution from the supplied data using the median absolute deviation (MAD). However, different standard deviations can be supplied by the user with the argument `sds`. The supplied `sds` must be of the same dimension as the copy number matrix. The following discussion elaborates briefly on the default procedure used to estimate the standard deviations.

In the example dataset, we have only one sample and no estimates of the copy number uncertainty. Therfore, we obtain a robust estimate of the copy number standard deviation across SNPs and use this as a rough estimate of the uncertainty. As the log-transformed copy number estimates are more nearly Gaussian, we calculate a robust estimate of the standard deviation using the median absolute deviation (MAD) on the log scale:

```
> cn.sds <- VanillaICE:::robustSds(copyNumber(locusset), takeLog = TRUE)
> dim(cn.sds)
```

```
[1] 9265    1
```

When multiple samples are available (e.g., 10 or more), SNP-specific estimates of the copy number uncertainty can be obtained by scaling an estimate of the variability across samples by a sample-specific estimate of noise. In the following code chunk, we simulate a matrix of copy number for 200 samples and then compute a robust SNP-specific estimate of the standard deviation.

```
> CT <- matrix(sample(copyNumber(locusset), 100 * 200, replace = TRUE),
+     100, 200)
> sds <- VanillaICE:::robustSds(CT, takeLog = TRUE)
```

The `robustSds` function returns a SNP-specific matrix of standard deviations provided that the copy number matrix has at least 3 samples. The *preferred* approach when the sample size is small (say, less than 10), is to develop SNP-specific estimates of the variance on a larger reference set, such as HapMap, using the same software, and then scale these estimates by a measure of the sample variance.

3.6 Missing genotype calls

If any of the genotype calls are missing and missingness is not independent of the underlying hidden state, one may specify the probability of a missing genotype calls for each hidden state (`probMissing`). By default, the HMM will assume that missing genotype calls are independent of the underlying hidden state. In particular, this assumption may not be reasonable for homozygous deletions. Again, the emphpreferred approach is to use the confidence scores provided by crlmm and the function `genotypeEmissionCrlmm`.

3.7 Transition probabilities

The sequence of states that maximizes the likelihood is obtained by the Viterbi algorithm. Note that the argument `arm` to this function is a factor indicating the chromosomal arms – the Viterbi algorithm is computed for independently for each chromosomal arm. We may scale the probability of transitioning between states by setting the arguments `normal2altered`, `altered2normal`, and `altered2altered`. For example, to facilitate comparisons to the Birdseye HMM [3] one may pass the following arguments to `viterbi`:

```
> fit <- viterbi(initialStateProbs = log(initialP), emission = emission,
+     tau = tau[, "transitionPr"], arm = tau[, "arm"], normalIndex = 2,
+     normal2altered = 0.005, altered2normal = 0.5, altered2altered = 0.0025)
```

The `TAUP` argument to the function `transitionProbability` together with the above arguments to the `viterbi` function can be used to control the 'smoothness' of the resulting predictions. In particular, higher values of `TAUP` encourages fewer jumps.

4 Session Information

The version number of R and packages loaded for generating the vignette were:

- R version 2.10.0 Under development (unstable) (2009-06-20 r48812), `i386-apple-darwin9.7.0`
- Locale: `en_US.UTF-8/en_US.UTF-8/C/C/en_US.UTF-8/en_US.UTF-8`
- Base packages: base, datasets, graphics, grDevices, methods, stats, utils
- Other packages: Biobase 2.5.3, crlmm 1.3.12, genomewidesnp6Crlmm 1.0.4, oligoClasses 1.7.5, SNPchip 1.9.1, VanillaICE 1.7.7
- Loaded via a namespace (and not attached): affyio 1.13.3, annotate 1.23.1, AnnotationDbi 1.7.5, Biostrings 2.13.21, DBI 0.2-4, ellipse 0.3-5, genefilter 1.25.5, IRanges 1.3.26, mvtnorm 0.9-7, preprocessCore 1.7.4, RSQLite 0.7-1, splines 2.10.0, survival 2.35-4, tools 2.10.0, xtable 1.5-5

References

[1] Benilton Carvalho, Henrik Bengtsson, Terence P Speed, and Rafael A Irizarry. Exploration, normalization, and genotype calls of high-density oligonucleotide SNP array data. *Biostatistics*, 8(2):485–499, Apr 2007.

[2] Stefano Colella, Christopher Yau, Jennifer M Taylor, Ghazala Mirza, Helen Butler, Penny Clouston, Anne S Bassett, Anneke Seller, Christopher C Holmes, and Jiannis Ragoussis. QuantiSNP: an Objective Bayes Hidden-Markov Model to detect and accurately map copy number variation using SNP genotyping data. *Nucleic Acids Res*, 35(6):2013–2025, 2007.

[3] Joshua M Korn, Finny G Kuruvilla, Steven A McCarroll, Alec Wysoker, James Nemesh, Simon Cawley, Earl Hubbell, Jim Veitch, Patrick J Collins, Katayoon Darvishi, Charles Lee, Marcia M Nizzari, Stacey B Gabriel, Shaun Purcell, Mark J Daly, and David Altshuler. Integrated genotype calling and association analysis of snps, common copy number polymorphisms and rare cnvs. *Nat Genet*, 40(10):1253–1260, Oct 2008.

[4] Robert B Scharpf, Giovanni Parmigiani, Jonathan Pevsner, and Ingo Ruczinski. Hidden Markov models for the assessment of chromosomal alterations using high-throughput SNP arrays. *Annals of Applied Statistics*, 2(2):687–713, 2008.

[5] Kai Wang, Mingyao Li, Dexter Hadley, Rui Liu, Joseph Glessner, Struan F A Grant, Hakon Hakonarson, and Maja Bucan. Penncnv: an integrated hidden markov model designed for high-resolution copy number variation detection in whole-genome snp genotyping data. *Genome Res*, 17(11):1665–1674, Nov 2007.