Chapter 22

Distinguishing Somatic and Germline Copy Number Events in Cancer Patient DNA Hybridized to Whole-Genome SNP Genotyping Arrays

Gavin Ha and Sohrab Shah

Abstract

Chromosomal aneuploidy and segmental copy number changes are common genomic aberrations in cancer. Copy number alterations (CNAs) arise from deletions, insertions, or duplications resulting in chromosomal aberrations and aneuploidy. Genomes of normal cells also exhibit variable copy number called germline copy number variants (CNVs). CNVs in the general population tend to confound interpretation of predictions when attempting to extract relevant driver somatic events in cancer. In large studies of CNAs in cancer patients, it becomes necessary to accurately identify and separate CNAs and CNVs so as to prioritize candidate tumor suppressors and oncogenes. We have developed a probabilistic approach, HMM-Dosage, for segmenting and distinguishing CNAs and CNVs as separate, discrete events in cancer SNP genotyping array data. We outline the steps and computer code for the analysis of whole-genome cancer DNA hybridized to SNP genotyping arrays, focusing on distinguishing somatic CNA and germline CNVs, and describe the combined approach of HMM-Dosage for probabilistic inference and classification of somatic and germline copy number changes.

Key words: Copy number alterations, Germline copy number variants, Probabilistic, HMM-Dosage, Cancer SNP genotyping array data, High-density SNP genotyping arrays, Affymetrix Genome-Wide Human SNP6.0, Illumina Human1M BeadChip

1. Introduction

Human cancer contains a complex combination of aberrations in the genome (1). Chromosomal aneuploidy and segmental copy number changes are among the most common genomic aberrations in cancer (2). CNAs, which can range in size from 1 kb to entire chromosomes, arise from events such as deletions, insertions, and duplications. These events can have pathogenic implications if regions containing oncogenes and tumor suppressor

Diponkar Banerjee and Sohrab P. Shah (eds.), *Array Comparative Genomic Hybridization: Protocols and Applications*, Methods in Molecular Biology vol. 973, DOI 10.1007/978-1-62703-281-0_22, © Springer Science+Business Media, LLC 2013

genes are affected. Large-scale studies profiling multiple types of genomic aberrations, including copy number and loss of heteozygosity (LOH), in cancer genomes have recently emerged. The Cancer Genome Atlas (TCGA) analyzed copy number and LOH (among many other genomic alterations) for 206 glioblastoma (3) and 489 high-grade serous ovarian adenocarcinomas (4) hybridized to single nucleotide polymorphism (SNP) genotyping arrays. In other studies, 746 cell-lines (5) and 2,520 primary tissues (6) across a range of cancer types were analyzed for copy number and LOH profiles. Moreover, the inclusion of somatic copy number events in the analysis of tumor samples, such as breast cancer, can aid in refining existing and discovering novel subtype classifications (7, 8).

Genomes of normal cells also exhibit, natural polymorphic regions with variable copy number called germline copy number variants (CNVs) (9–13). CNVs are often prevalent near segmental duplications (14) and contribute to global variations and phenotypic differences in healthy individuals. Substantial effort has been made in sequencing large cohorts to comprehensively explore CNVs as part of the 1000 Genomes Project (15, 16). Results from a majority of these studies surveying CNVs have been archived in the Database of Genomic Variants (DGV, http://www.projects.tcag.ca/variation/) (17). While germline CNVs in the population may be associated with disease (18–21), in the case of cancer, these events tend to confound interpretation of predictions when attempting to extract relevant driver somatic events. Analogous to germline SNPs in somatic point mutation analysis, CNVs are present as a source of genetic variation whose role in tumor pathophysiology is undetermined. As more studies involve ever-increasing cohort sizes of cancer patients, it becomes necessary to routinely use methods that are capable of accurately identifying and systematically separating CNA and CNV events so as to prioritize candidate tumor suppressors and oncogenes.

1.1. Detecting Copy Number Changes in Cancer Patient DNA Hybridized to SNP Genotyping Arrays

High-density SNP genotyping arrays are effective high-throughput assays for analyzing genome-wide copy number in humans. This technology is well established and cost-effective for large cohort cancer studies and analysis for recurrent copy number events.

Affymetrix Genome-Wide Human SNP6.0 and Illumina Human1M BeadChip are popular platforms used for genotyping, copy number, and LOH profiling. A single array, consisting of more than 1 million SNP and CNV probes, measures hybridization intensity as a surrogate for the amount of DNA at each loci. Analysis of SNP arrays primarily use two quantitative metrics derived from the allelic intensities. The first is the *log ratio*, which is computed as the ratio between the total (sum of A + B alleles) intensity of the tumor sample and the total intensity of the match normal or reference sample at a particular locus. The log ratio data at each SNP and CNV probe is used for performing total copy number prediction while allele-specific or parent-specific copy number uses only the set of SNP probes. In the absence of matched normal tissue, a pooled *reference* sample can be used. Typically, the reference is generated by taking the median probe intensity across a pool of healthy samples such as from the HapMap project (22). While the reference sample can help reduce the amount of population CNVs from the tumor data, using matched normal samples will be best for removing patient-specific CNVs. The second metric is B-allele frequency (BAF; usually derived from minor allele identification in population studies), which is the proportion of B-allele intensity relative to the total intensity computed solely using SNP probes. BAF naturally allows for analysis of genotypes and LOH whereby clusters of consecutive SNP probes are heterozygous with BAF of ~0.5 in the normal and becomes homozygous with BAF equal to 0 or 1 due to somatic allelic loss.

Similar to gene expression microarrays, SNP array intensities also need to be normalized to eliminate background noise. Normalization tools are platform-specific and generally categorized based on the manufacturer. For Affymetrix, *CRMAv2* (23, 24) and *ACNE* (25) perform cross-hybridization of probes between the alleles, GC content, and restriction enzyme induced fragment-length biases. For Illumina, *crlmm* (26, 27) and *tQN* (28) are tools that normalize raw and BeadStudio-processed intensities (29), respectively. In general, copy number neutrality, gain, and loss are inferred from values near, greater than, and less than zero in log scale, respectively.

Following normalization, segmentation algorithms are used to detect regions of copy number gain and loss that contiguously span a variable number of probes. Circular binary segmentation (30, 31) is a nonparametric technique for detecting significant change at breakpoints in aCGH and SNP genotyping data of a single sample. More recently, hidden Markov models (HMM) designed specifically for analyzing cancer data have been applied to segment aCGH and SNP genotyping data to handle noise and outliers by probabilistically accounting for spatial correlation (32–35) (Table 1). Furthermore, HMM-based tools are attractive due to their flexibility in modeling variable number of discrete biologically interpretable states and use statistical modeling approaches to account for polyploidy, normal contamination, and tumor heterogeneity.

1.2. Germline CNVs and Somatic CNAs in Cancer

We have previously determined the comprehensive landscapes of CNA and CNV in a cohort of 2,000 breast cancer patients whose DNA were hybridized to SNP genotyping arrays (8). We found that CNVs in tumor data have distinct statistical patterns compared

Table 1
Segmentation algorithms

Algorithm/software	Type of genomic aberration	Comments
DNAcopy (CBS) (36)	Copy number	Nonparametric. Requires post hoc step for discrete copy number classification
PICNIC (33)	Copy number and LOH	Ploidy correction, allele-specific copy number
GPHMM (35)	Copy number and LOH	Ploidy correction, GC content correction, stromal contamination
OncoSNP (34)	Copy number and LOH	Ploidy correction, GC content correction, stromal contamination, intra-tumoral heterogeneity
HMM-Dosage (8)	Copy number	Distinguish somatic CNA and germline CNV

to CNAs likely due to the polymorphic nature of germline variants and contributing normal tissue contamination in the sample. Furthermore, CNVs tend to be much shorter in length compared to CNAs based on inspection of records in Database of Genomic Variants (DGV) (17) and analysis of the breast cancer cohort. Normal contamination leads to weaker CNA event signals and shifting of log ratio amplitudes toward a diploid state. This is because CNAs are only found in the subpopulation of the sample making up the cancer cells; moreover, CNV signals are exaggerated relative to CNAs because all cells in the sample contain these germline events.

There are several strategies that can be used to separate CNAs and CNVs in cancer genome analysis. The best solution is to use an array from the corresponding matched normal DNA to compute log ratios prior to segmentation. This generates neutral ratio signals for germline variant probes, which allows for excluding CNVs inherently during somatic analysis but removes the ability to predict CNVs during segmentation.

In many instances, normal DNA from patients cannot be acquired; therefore, a reference is used (as described earlier). CNAs and CNVs can be detected and distinguished using post hoc approaches. The simplest approach is to manually "subtract" (or filter) copy number events that overlap records in a database containing germline polymorphic CNVs such as DGV. This presents a drawback; however, as there are uncertainties in region sizes due to differing resolutions from various technologies used in DGV entries. Furthermore, DGV regions collectively account for a large fraction of the genome, which can be problematic when applying the filtering approach. This can potentially exclude a large percentage of the genome from the results. To address these issues, PredictCNV (36), a regression tree classifier trained on DGV features, classifies segments of copy number changes as CNA or CNV. The distributed classifier is trained only on the lower resolution arrays in the TCGA glioblastoma data, and it is not clear if it is suitable for other cancer types or data using high-density arrays. These methods applied separately after segmentation analysis and do not probabilistically factor into improving segmentation of these events.

We have developed a probabilistic approach, HMM-Dosage, which extends ref. (32) for segmenting and distinguishing CNAs and CNVs as separate, discrete events in cancer SNP genotyping array data. This algorithm, described in ref. (8), is a combined approach that simultaneously performs segmentation and classification of CNAs and CNVs. HMM-Dosage models the different statistical patterns of CNVs and CNAs in a single, unified probabilistic framework by incorporating position-specific CNV prior knowledge.

We outline the steps and computer code for the complete analysis pipeline of whole-genome cancer DNA hybridized to SNP genotyping arrays, with the specific focus on distinguishing somatic CNA and germline CNVs. For this protocol, we will first describe the normalization of Affymetrix SNP6.0 arrays using CRMAv2. Next, we will detail the procedure for segmenting log ratio data using two different approaches in CBS and OncoSNP, and distinguishing CNAs and CNVs using the post-processing approach of the PredictCNV classifier. Finally, we will describe the combined approach of HMM-Dosage for probabilistic inference and classification of somatic and germline copy number changes.

The advantages of separating CNA and CNV predictions are twofold. First, and most importantly, filtering germline events from somatic CNAs helps prioritize regions that contain candidate driver tumor suppressors or oncogenes. Secondly, the dichotomized list of events can allow separate downstream analysis including investigating CNV-specific patterns in the cohort of cancer patients such as its cis-acting effect on gene expression.

2. Materials

2.1. Tumor Whole-Genome SNP Genotyping Array Data

1. Raw hybridization intensities of a tumor sample hybridized to an Affymetrix Genome-Wide Human SNP6.0 array. Match normal sample array or reference sample also hybridized to Affymetrix SNP6.0 (see Note 1). A precomputed reference derived from the HapMap270 dataset can be downloaded at http://www.compbio.bccrc.ca/software/hmm-dosage/.

2.2. Normalization Software Download and Installation

1. R programming software is required to run many software packages described in this protocol. R (version R-2.10.1 or higher) can be obtained at http://www.r-project.org/.
2. Allele-specific CRMAv2 normalization tool is part of the R library package *aroma.affymetrix* (v1.4). To install these packages, use these commands while in an active R session:

   ```
   >source("http://www.aroma-project.org/
   hbLite.R");
   ```

   ```
   >hbInstall("aroma.affymetrix");
   ```
3. Download required annotation data files from http://www.affymetrix.com and http://www.aroma-project.org/chipTypes/GenomeWideSNP_6 (see Note 2). Set up the directory structure as described in http://www.aroma-project.org/setup.

2.3. Copy Number Segmentation Software Download and Installation

1. CBS is part of the R package, DNAcopy, which can be installed within an R session. Method and parameter definitions are comprehensively detailed in http://www.bioconductor.org/packages/2.6/bioc/manuals/DNAcopy/man/DNAcopy.pdf.

   ```
   >install.packages("DNAcopy");
   ```
2. PredictCNV is implemented in R. The files required ("*prediction_code.R*" and "*CNV_prediction_objects.RData*") to run the algorithm can be downloaded from http://www.mskcc.org/mskcc/html/72726.cfm. The authors provide a document with detailed definitions and instructions for each function. The *randomForest* R library is also required and can be downloaded here http://www.cran.r-project.org/web/packages/randomForest/index.html.
3. OncoSNP (https://sites.google.com/site/oncosnp/) is a cancer-specific copy number segmentation software that can be downloaded as compiled MATLAB executables (see Note 3).
4. HMM-Dosage (http://www.compbio.bccrc.ca/software/hmm-dosage/) is software for segmentation and prediction of copy number in tumor data (see Note 3).

In the next section, we assume that all software has been correctly installed.

3. Methods

3.1. Normalization of Array Intensities Using AS-CRMAv2

1. Within a new R session (version R-2.10.1 or higher), load the *aroma.affymetrix* library package that contains the CRMAv2 normalization tool (see Fig. 1, A1).

   ```
   > library("aroma.affymetrix")
   ```

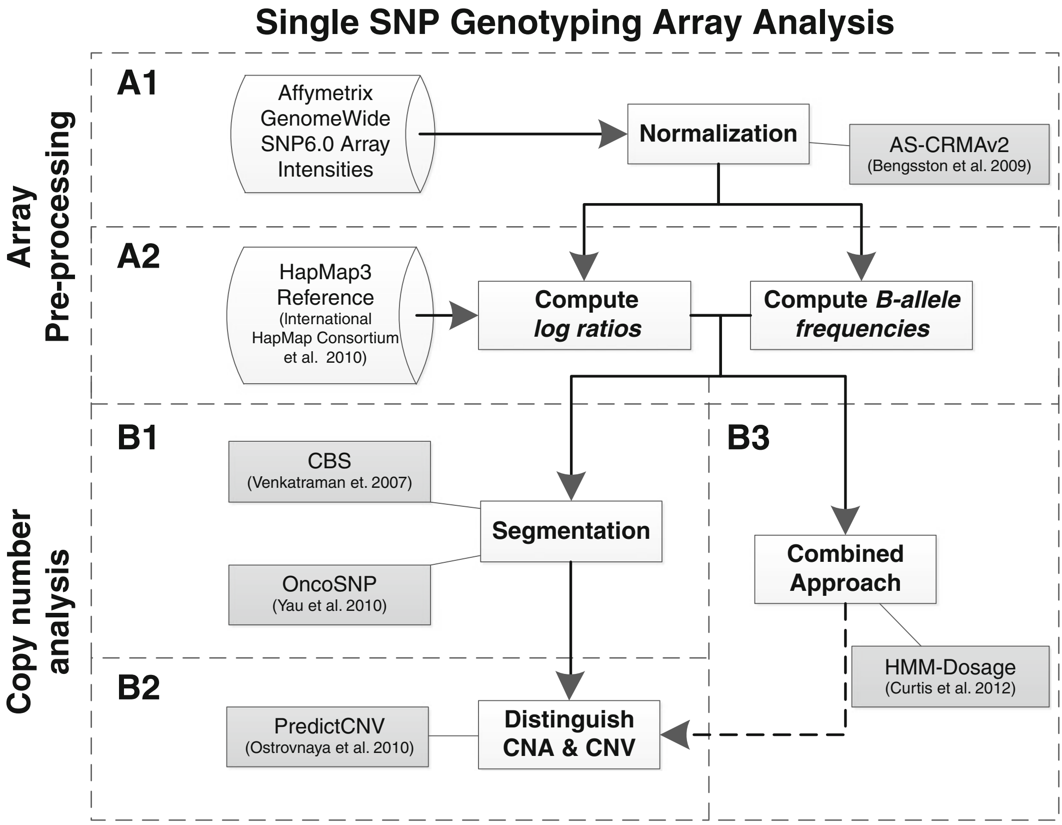
Single SNP Genotyping Array Analysis
Array Pre-processing
Copy number analysis
A1
Affymetrix GenomeWide SNP6.0 Array Intensities
Normalization
AS-CRMAv2 (Bengsston et al. 2009)
A2
HapMap3 Reference (International HapMap Consortium et al. 2010)
Compute log ratios
Compute B-allele frequencies
B1
CBS (Venkatraman et. 2007)
Segmentation
OncoSNP (Yau et al. 2010)
B3
Combined Approach
HMM-Dosage (Curtis et al. 2012)
B2
PredictCNV (Ostrovnaya et al. 2010)
Distinguish CNA & CNV

Fig. 1. Workflow diagram for single genotyping array analysis procedure of the Affymetrix Genome-Wide SNP6.0 platform. (A1) First step in the analysis consists of intensity normalization using allele-specific (AS)-CRMAv2. (A2) Computing the *log ratios* requires a reference sample that can be generated from any dataset consisting of normal (not tumor) individuals; HapMap3 dataset is described. The *B-allele frequency* (BAF) is computed as the relative fraction of the B-allele intensity with respect to the total intensity. (B1) Nonparametric (e.g., CBS) and parametric (e.g., OncoSNP) algorithms are used to segment log ratio and/or BAF data into regions of discrete copy number changes. However, most approaches do not inherently consider distinguishing germline CNV and somatic CNA events. (B2) The classification model, PredictCNV, was designed to probabilistically separate CNVs and CNAs in results generated from other segmentation algorithms. (B3) HMM-Dosage is a hidden Markov model that simultaneously accounts for CNAs and CNVs, segmenting and predicting regions of copy number changes for both somatic and germline events. As an optional post-processing step, PredictCNV can be used to filter the output of HMM-Dosage (*dashed arrow*).

2. Link the location of the chip definition file (*cdf*) for the *Affymetrix GenomeWideSNP_6* platform (see Note 2).

```
> cdf <- AffymetrixCdfFile$byChipType("Genom
eWideSNP_6", tags="Full")
```

3. Load a single CEL file into an R object and assign it to a set containing only the one sample (see Note 4).

```
> celFile <- AffymetrixCelFile$fromFile(celF
ilename)
>csRSingle <- AffymetrixCelSet(list(celFile))
> setCdf(csRSingle, cdf=cdf)
```

4. Perform allele-specific normalization using CRMAv2 (24) methodology (see Note 5).

```
> ces <- doASCRMAv2(csRSingle, chipType="Gen
omeWideSNP_6,Full")
```

5. If a match normal is available, normalize the match normal sample CEL file using steps 1–4. Assign the output of `doASCRMAv2()` to the variable `cesMN`. Skip to step 7. If the match normal is unavailable, skip step 5 and move on to step 6 to use the reference sample.

```
> cesMN <- doASCRMAv2(…)
```

6. Download and load the provided reference sample tab-delimited text file (see Note 1 and Subheading 2.1) if the match normal is unavailable. Load the text file and store it in `cesR`.

```
> cesR <- read.table(refFilename, sep="\t",
header=FALSE)
```

7. Load the genomic information corresponding to the probe sets given by the *cdf* annotation file (see Note 2).

```
> gi <- getGenomeInformation(cdf)
```

8. For a specified chromosome c, obtain sorted unit identifiers and genomic positions.

```
> units <- getUnitsOnChromosome(gi,
chromosome=c)
> pos <- getPositions(gi, units=units)
> probes <- getUnitNames(cdf, units=units)
> N <- length(pos)
> chrIndex <- as.data.frame(rep.int(c,N))
> sI <- order(pos)
```

9. Extract the total intensities and B-allele frequencies (BAF) for the probe units of the given chromosome c (see Note 6).

```
> theta <- extractTotalAndFreqB(ce, units=
units)
> # computed BAF
> BAF <- theta[,"freqB"]
> BAF[grepl("CN",probes)] <- 2 # use 2 for CN
probes
```

10. Compute the raw copy number for each probe by first extracting the normalized intensities of the match normal or reference (see Notes 1 and 7, and Fig. 1, A2). Here, the given example uses the reference. One can use the matched normal instead by replacing `cesR` with `cesRMN`.

```
> # extract reference intensities
> thetaR <- cesR[cesR[,1]==j,]
> # compute raw copy number
> # assume that reference positions match exactly with the sorted positions of theta (that is, pos[sI])
> C <- 2.0*theta[sI]/thetaR[,3]
```

11. Output two sets of results for raw copy number and *log2 ratios*. Each file consists of 5 columns, sorted by positions for given chromosome c using sI generated in step 8. The columns are probe name, chromosome number, genomic coordinate of probe, raw copy number or log ratio, and B-allele frequency.

```
> cnDataFrameRawCN <- cbind(probes[sI],chrIndex,pos[sI],C[sI],BAF[sI])
> cnDataFrameLogR <- cbind(probes[sI],chrIndex,pos[sI],log2(C[sI]/2),BAF[sI])
> write.table(cnDataFrameRawCN, file=fc1, append=T, quote=FALSE, sep=”\t”, eol=”\n”, na=”NaN”, dec=”.”, row.names=F, col.names=F)
> write.table(cnDataFrameLogR, file=fc2, append=T, quote=FALSE, sep=”\t”, eol=”\n”, na=”NaN”, dec=”.”, row.names=F, col.names=F)
```

12. Repeat steps 8–11 for each of the 22 autosomes and 2 sex chromosomes. To do this in a single code block, use a loop around the code commands listed in steps 8–11. The results are output to the file "*rawCN-BAF_output.txt*" and "*logR-BAF_output.txt*" Fig. 2, Panel 1 shows a plot of the normalized log ratios for chromosome 10 of a breast cancer sample.

```
> fc1 <- file(“rawCN-BAF_output.txt”,“w+”) #open file handles
> fc2 <- file(“logR-BAF_output.txt”,“w+”)
> for (j in 1:24){
## Use chr numbers 1-22, X, and Y
if (j==23){ c=’X’ }
else if (j==24){ c=’Y’ }
else { c=j }
## code from steps 8-11…
}
> close(fc1) #close the file handle
> close(fc2)
```

3.2. Segmentation Analysis of Copy Number Changes: CBS

1. Within a new R session (version R-2.10.1 or higher), load the *DNAcopy* library package that contains the circular binary segmentation (CBS) algorithm (see Fig. 1, B1).

```
> library(DNAcopy)
```

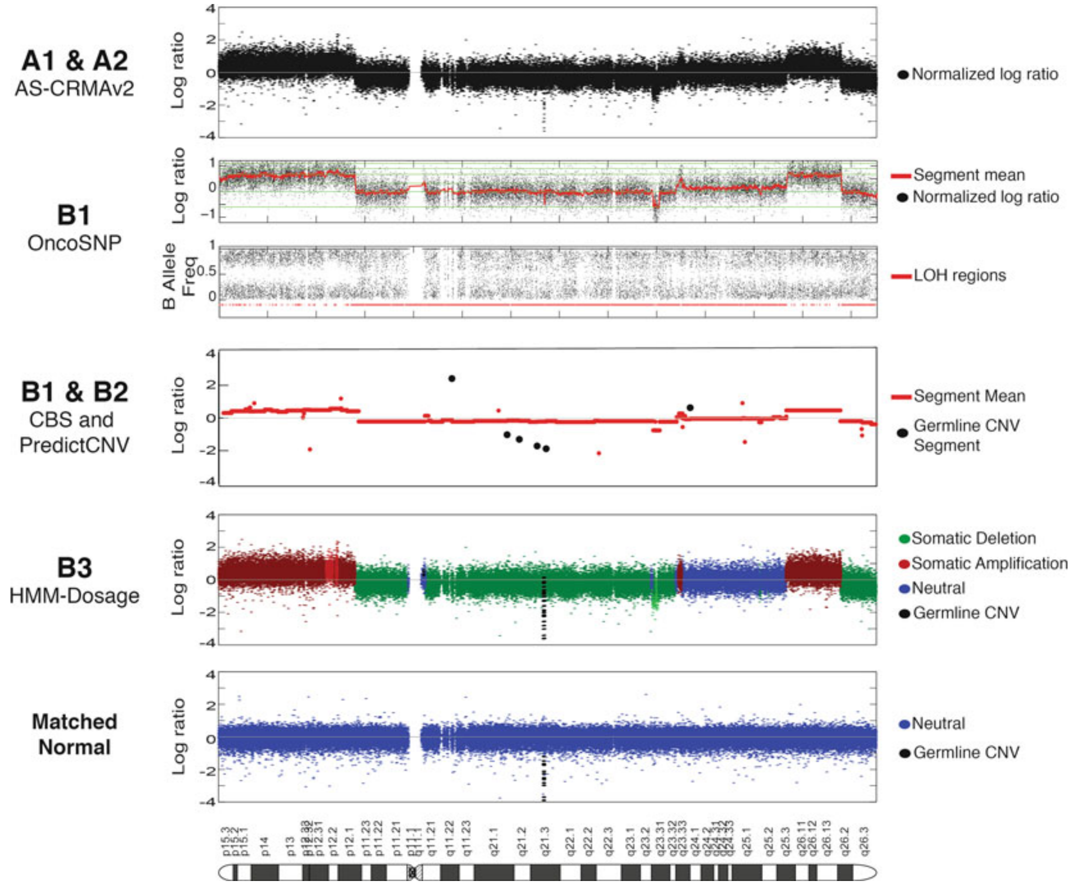

Fig. 2. Copy number results for chromosome 10 of a triple negative breast cancer sample (38) hybridized to the Affymetrix SNP6.0 platform. Panel 1 displays the results for AS-CRMAv2 (24) normalization and computation of log ratios without copy number state classification (Steps A1 and A2, see Fig. 1). Panels 2 and 3, which are plots generated directly by OncoSNP (34), show the predicted segment means and loss of heterozygosity (LOH). Panel 4 shows the predicted segments using CBS and classification of germline CNVs using PredictCNV (36) (Steps B1 and B2). Panel 5 displays copy number predictions generated by HMM-Dosage (8), which distinguishes germline CNVs and somatic CNAs (Step B3). An example of a CNV predicted by HMM-Dosage can also be found in the matched normal sample of the same patient (Panel 6).

2. Load in the log ratio data file "*logR-BAF_output.txt*" for a single sample generated from the output of AS-CRMAv2 generated in Subheading 3.1 (see Note 8).

```
> sampleRawCN <- read.table(file="logR-BAF_
output.txt", header=F, stringsAsFactors=F)
```

3. Prepare a "copy number array" object that will be passed into the main function for CBS analysis (see Note 9).

```
> sampleID = "test1"
sampleLogR <- sampleRawCN[,4]
> cna.obj <- CNA(sampleLogR,
chrom=sampleRawCN[,2], maploc=sampleRawCN[,3],
data.type="logratio", sampleid=c(sampleID))
```

4. (Optional) Smoothing can be applied to the CNA object prior to segmentation (see Note 10).

```
> cna.obj <- smooth.CNA(cna.obj)
```

5. Perform segmentation using CBS (see Note 11).

```
> segment.cna.obj <- segment(cna.obj)
```

6. Prepare the results in a new variable and save it as an image for downstream analysis (see Note 12).

```
> dataseg$data  <-  cbind(sampleRawCN[,2:3],
sampleLogR)
> names(dataseg$data) <-
c("chrom","maploc",sampleID)
> dataseg$output <- segment.cna.obj$output
> dataseg$data[is.nan(dataseg$data[,3]),3] <- 0
> save(list=dataseg, file="cbs_results.RData")
#save the variable
```

3.3. Segmentation Analysis of Copy Number Changes: OncoSNP

1. Ensure that the current operating system is Linux x64 architecture and that OncoSNP is properly installed (see author website listed in Subheading 2.3, step 2, Note 3, and Fig. 1, B1).

2. Prepare a 3-column text file that specifies the sample ID, locations to the input tumor *log ratio* and *BAF* data, and location of normal sample *log ratio/BAF* data (see Note 13). For this example, let this file be named "*singleSampleInputForOncoSNP.txt*." The tumor *log ratio/BAF* file for this example is "*logR-BAF_output.txt*," which was generated in Subheading 3.1.

3. Download and install the GC content files for NCBI genome build 37 (hg19) into a folder named "quantisnp."

```
OncoSNP_v1.1> mkdir quantisnp; cd quantisnp;
wget  ftp://ftp.stats.ox.ac.uk/pub/yau/quantisnp2/download/b37.tar.gz
tar xvfz b37.tar.gz
```

4. Create an output directory for storing the results.

```
OncoSNP_v1.1> mkdir results/
```

5. Run OncoSNP from the Linux terminal command line while in the installation directory (see Note 14). Fig. 2, Panel 2–3 shows a plot of the copy number and loss of heterozygosity (LOH) predictions.

```
OncoSNP_v1.1> run_oncosnp.sh \
MATLAB_Compiler_Runtime/v714/ \
--batch-file singleSampleInputForOncoSNP.txt \
--output-dir results/ \
--gcdir quantisnp/b37/ \
--paramsfile configuration/hyperparameters-affy.
    dat \
--levelsfile configuration/levels-affy.dat \
--trainingstatesfile configuration/trainingStates.
    dat \
--tumourstatesfile configuration/tumourStates8.
    dat \
--subsample 30 --emiters 15 --headerlines 0 \
--stromal --intratumour --female --plot –fulloutput
    --headerlines 0
```

3.4. Distinguishing Somatic CNAs and Germline CNVs: PredictCNV Classifier

1. Within a new R session (version R-2.10.1 of higher), load the library and scripts necessary to run PredictCNV (see Subheading 2.3, step 2 and Fig. 1, B2). Ensure that the required files of "*prediction_code.R*" and "*CNV_prediction_objects.RData*" are in the current working directory, otherwise, specify the full path.

   ```
   > library(randomForest)
   > source('prediction_code.R')
   > load('CNV_prediction_objects.RData')
   ```

2. Load in the segmentation data generated from Subheading 3.2 (see Note 15) using the load command in R. Ensure that the saved "*.RData*" file is in the current working directory, otherwise, specify the full path.

   ```
   > load("cbs_results.RData")
   ```

3. Run the prediction software using the PredictCNV classifier (see Notes 16 and 17).

   ```
   > newSegs <- predict.CNVs(dataseg, use.
   cohort=FALSE, smoothed=TRUE, glad=FALSE,
   gainloss.defined=FALSE, nmad=1)
   ```

4. Output the results into a file consisting of tab-delimited columns (see Note 18). Figure 2, Panel 4 shows examples of classified CNV segments.

   ```
   > write.table(newSegs, file=outFile, col.
   names=T, row.names=F, sep="\t", quote=F)
   ```

3.5. HMM-Dosage: A Combined Approach for Segmenting and Predicting CNAs and CNVs

1. Ensure that the current operating system is Linux x64 architecture and that HMM-Dosage is properly installed (see author website listed in Subheading 2.3, step 4, Note 3, and Fig. 1, B3).

2. Prepare the input raw copy number file by processing the file "*rawCN-BAF_output.txt*," which was generated in Subheading 3.1. In the command line, extract the columns 2–4, which contain chromosome number, genomic position, and log ratio, and substitute the chromosome X and Y symbols with 23 and 24, respectively. This is assigned to the new file "*rawCN-BAF_output_processed.txt*."

```
HMMK11_0.1.0> cut -f 2-4 rawCN-BAF_output.txt
| sed s/X/23/g | sed s/Y/24/g > rawCN-BAF_out-
put_processed.txt
```

3. Download the CNV frequency file from the website http://www.compbio.bccrc.ca/software/hmm-dosage/ (see Subheading 2.3, step 4 and Note 19). For this example, let's assume the desired frequencies are derived from the HapMap3 (22) dataset.

```
HMMK11_0.1.0> wget http://www.compbio.bccrc.
ca/wp-content/uploads/2010/10/
HapMap3CNV1258_probeFreq_AffySNP6.txt.gz;
gunzip   HapMap3CNV1258_probeFreq_AffySNP6.
   txt.gz
```

4. Create an output directory to store the results.

```
HMMK11_0.1.0> mkdir results/
```

5. Run HMM-Dosage from the Linux command line (see Note 20). Figure 2, Panel 5 shows an example plot of a predicted CNV, distinguished from somatic events.

```
HMMK11_0.1.0> cd bin/;
./run_hmmK11LogR.sh \
../MATLAB_Component_Runtime/v77/ \
../rawCN-BAF_output_processed.txt \
../HapMap3CNV1258_probeFreq_AffySNP6.txt \
../test/initialParams.mat \
../results/hmmdosage_results_cna.txt \
../results/hmmdosage_results_segs.txt \
../results/hmmdosage_results.mat \
-1
```

4. Notes

1. When a matched normal is unavailable, the reference sample can be generated from an external dataset of normal individuals such as from HapMap270 (37) or HapMap3 (22) cohorts. The dataset is summarized by computing the median across all individuals for each probe *i*, θ_i^R . This value is used as the denominator in the log ratio. The choice of the reference dataset should take into consideration gender, which can affect analysis of chromosome X, and ethnic origin, which can affect population specific germline CNVs.
2. Ensure that the chip definition file (cdf), "*GenomeWideSNP_6, Full.cdf*," is in the location, "*/annotationData/chipTypes/GenomeWideSNP_6/*." This file contains the genomic coordinates

and probe sequences corresponding to the probe sets found on the array. Furthermore, ensure the presence of a file for each of these three extensions: *.acs*, *.ufl*, *.ugp*; these files should all correspond to the same genome build (e.g., *hg18*).

3. These MATLAB executables were originally compiled on Linux x64 architecture operating systems; thus, they can only be run on machines with the this architecture.

4. Scripting the analysis to run one sample, independently of others, allows for easy parallelization. CRMAv2 was designed as a single-sample normalization technique.

5. The function `doASCRMAv2()` can take up to a few hours. It performs allelic cross-talk calibration, base-pair normalization, probe-level summarization, and fragment-length normalization (23, 24).

6. The output to function `extractTotalandFreqB()` consists of 2 columns: (1) Total intensity and (2) Fraction of B-allele intensity (BAF) with respect to total intensity. Total intensity θ_i is computed as the sum between the normalized intensities of the major allele θ_i^A and minor allele θ_i^B for each SNP probe *i*, and the measured intensity of each CNV probes. BAF is computed as $\theta_i^B / \theta_i^A + \theta_i^B$. CNV probes are non-polymorphic and will only contain one intensity value for each probe; hence, BAF values will not be computed for these.

7. Raw copy number is computed as the ratio between the normalized tumor intensity and normalized reference (or match normal) intensity of probe *i* given by equation: $C_i = 2(\theta_i / \theta_i^R)$. The factor of 2 is simply for interpretability of a neutral ratio being "two" copies. When computing raw copy number, you must ensure that the positions in the reference file correspond exactly with the positions in the tumor sample.

8. This demonstration of CBS only involves segmentation of total copy number; hence, the BAF data (5th column) in the input data file (output from AS-CRMAv2) is not used. It is possible to segment the BAF data to find segments of allelic imbalance for LOH analysis. For this example, the log ratio data is used; therefore, the file "*logR-BAF_output.txt*" is used.

9. The `Copy Number Array` object requires several arguments. `chrom` specifies the chromosome for each probe; this is given by the 2nd column in the input raw copy number file. `maploc` specifies the genomic coordinate of the probes; this is given by the 3rd column. `data.type` is set to "log ratio." `sampleid` is user specified.

10. Smoothing is applied to the `Copy Number Array` object prior to segmentation. Default settings for this function using sliding window of 5 data points for detecting outliers with the 3rd value being the data point of interest. Smoothing of an

outlier involves shifting the data point to 2 standard deviations from the median of the sliding window.

11. The `segment()` function does *not* generate the discrete copy number call; it only performs segmentation. The $output list of the return object for the `segment()` function consists of 6 columns: (1) sample ID, (2) chromosome, (3) genomic start coordinate of segment, (4) genomic end coordinate of segment, (5) number of markers for segment, and (6) average log ratio across all probes in the segment.

12. Processing of the `segment()` function results into the `dataseg` variable as shown is necessary for running PredictCNV in a downstream analysis step.

13. OncoSNP version 1.1 accepts a 3-column file that specifies the sample IDs, locations of tumor sample *log ratio/BAF* data files, and locations of normal sample *log ratio/BAF* data files (if available). For the purposes of parallelization, a single sample is specified: the input tumor *log ratio* and *BAF* data is the file "*logR-BAF_output.txt*." Note that "*logR-BAF_output.txt*" is already in the correct format.

14. The full details of the parameter arguments for OncoSNP version 1.1 are described in the author website listed in Subheading 2.3, step 3. `--batch-file` specifies the file created in Subheading 3.3, step 2. `--gcdir` specifies the directory of the GC content files discussed in Subheading 3.3, step 3; this is used for correcting local GC content bias during total copy number analysis of *log ratios*. `--paramsfile` contains the software settings and hyperparameter of the model. `--levelsfile` contains the mean (log ratio) for each distribution corresponding to a copy number state. `--trainingstatesfile` contains the settings used for model training via expectation maximization. `--tumourstatesfile` specifies the list of copy number and allelic imbalance states used in the model. `--subsample` specifies the proportion of randomly sampled data points to use for model training. `--emiters` specifies number of iterations for model training. `--stromal` is a flag that, when used, will inform the model to account for normal contamination, which by default, will be estimated as between 0 and 90% (at intervals of 10%). `--intratumour` is a flag that will inform the model to estimate intra-tumoral heterogeneity. Use `--female` for including chromosome X in the analysis whereas `--male` includes chromosome Y. `--fulloutput` flag indicates for the software to also output the probe-level results in addition to the segments. The description of output files and plots are provided in the author website listed in Subheading 2.3, step 3.

15. This example will only demonstrate the application of the PredictCNV classifier using CBS segmentation output generated in Subheading 3.2. The classifier can use segmentation results generated from any method as described in ref. (36). We leave it to the reader to experiment with segments produced from other algorithms such as OncoSNP or HMM-Dosage. To be compatible with PredictCNV, the outputs from these algorithms will need to be adjusted to the acceptable format.

16. PredictCNV is a supervised approach that uses a random forest decision tree classification model. The model in the distributed code was trained using TCGA glioblastoma data hybridized to Aglient 244k arrays (3). This algorithm is designed to produce more accurate results when a cohort of tumor patients is used. Therefore, the segments from each patient can be concatenated into a single object for input into the classifier. For this guide, we use the single sample analysis approach to allow for convenient parallelization of large number of samples.

17. The default input parameters to the `predict.CNVs()` function are used with the exception of `use.cohort=FALSE`. The `gain.loss.defined` parameter specifies whether a column with header name "*state*" and values of {"*Loss*," "*Normal*," "*Gain*"} is included in the object `dataseg$output`. OncoSNP and HMM-Dosage provide discrete copy number predictions that can be used to populate the "*state*" column after proper formatting (see Note 15). These results can be analyzed with `predict.CNVs()` while setting `gain.loss.defined=TRUE`. For CBS, the copy numberpredictionisnotprovided;thus,`gain.loss.defined=FALSE` and `nmad` parameters are used. `nmad` specifies the "threshold for the number of median absolute deviations of the array residuals" for calling "*Gain*" or "*Loss*" (see Documentation provided by the authors, Subheading 2.3, step 2).

18. The output of the `predict.CNVs()` the same as `dataset$output` except with an additional column, "*predict.CNV*," where TRUE denotes the segment as a predicted CNV and FALSE denotes a predicted CNA segment.

19. The CNV frequency file consists of probe-level frequencies of the germline copy number gains and losses across a normal (healthy) dataset such as HapMap (37). The frequency file contains 6 columns, each corresponding to the CNV frequencies of homozygous deletion, hemizygous deletion, gain, amplification, and high-level amplification. Inclusion of these frequencies, which is part of a key feature in HMM-Dosage, provides prior knowledge of a particular locus being a germline CNV event in the population. This allows HMM-Dosage to simultaneously segment and predict copy number, returning calls from an increased state-space that includes both somatic CNAs and germline CNVs.

20. The full details of the parameter arguments for HMM-Dosage (version 0.1.0) are described in the author website listed in Subheading 2.3, step 4. The first argument is the `infile`, which in the example was generated at Subheading 3.5, step 2, contains 3 columns for chromosome number, genomic coordinate of probe, and raw copy number (see Note 7). `freqfile` specifies the file downloaded as described in Subheading 3.5, step 3 (see Note 19). `paramSetFile` contains the software settings, hyperparameter, means (*log ratio*) for each distribution corresponding to the copy number state, and settings used for model training via expectation maximization. `outfile` specifies the probe-level output file which will contain 5 columns for chromosome number, start position of probe, end position of probe, log ratio, and state predictions. Somatic CNA states are correspond to 1 (homozygous deletion), 2 (hemizygous deletion), 4 (gain), 5 (amplification), and 6 (high-level amplification). For germline CNVs, the same corresponding discrete state definitions are assigned to 7–11, respectively. State 3 is the neutral genotype. `paramfile` specifies the output file that stores the converged training parameters for the model. This information is useful for advanced users who wish to inspect parameter changes throughout expectation maximization estimation. The file is in MATLAB binary format, thus, MATLAB installation is required. `chr` use −1 for this argument.

References

1. Stratton MR, Campbell PJ, Futreal PA (2009) The cancer genome. Nature 458:719–724
2. Negrini S, Gorgoulis VG, Halazonetis TD (2010) Genomic Instability—an evolving hallmark of cancer. Nat Rev Mol Cell Biol 11:220–228
3. Cancer Genome Atlas Research Network (2008) Comprehensive genomic characterization defines human glioblastoma genes and core pathways. Nature 455:1061–1068
4. Cancer Genome Atlas Research Network (2011) Integrated genomic analyses of ovarian carcinoma. Nature 474:609–615
5. Bignell GR, Greenman CD, Davies H et al (2010) Signatures of mutation and selection in the cancer genome. Nature 463:893–898
6. Beroukhim R, Mermel CH, Porter D et al (2010) The landscape of somatic copy-number alteration across human cancers. Nature 463:899–905
7. Chin SF, Teschendorff AE, Marioni JC et al (2007) High-resolution aCGH and expression profiling identifies a novel genomic subtype of ER negative breast cancer. Genome Biol 8:R215
8. Curtis C, Shah SP, Chin S et al (2012) The genomic and transcriptomic architecture of 2,000 breast tumours reveals novel subgroups. Nature 486:346–352
9. Sebat J, Lakshmi B, Troge J et al (2004) Large-scale copy number polymorphism in the human genome. Science 305:525–528
10. Tuzun E, Sharp AJ, Bailey JA et al (2005) Fine-scale structural variation of the human genome. Nat Genet 37:727–732
11. Redon R, Ishikawa S, Fitch KR et al (2006) Global variation in copy number in the human genome. Nature 444:444–454
12. Kidd JM, Cooper GM, Donahue WF et al (2008) Mapping and sequencing of structural variation from eight human genomes. Nature 453:56–64
13. Conrad DF, Pinto D, Redon R et al (2010) Origins and functional impact of copy number variation in the human genome. Nature 464:704–712

14. Sharp AJ, Locke DP, McGrath SD et al (2005) Segmental duplications and copy-number variation in the human genome. Am J Hum Genet 77:78–88
15. 1000 Genomes Project Consortium, Durbin RM, Abecasis GCR et al (2010) A map of human genome variation from population-scale sequencing. Nature 467:1061–1073
16. Mills RE, Walter K, Stewart C et al (2011) Mapping copy number variation by population-scale genome sequencing. Nature 470:59–65
17. Iafrate AJ, Feuk L, Rivera MN et al (2004) Detection of large-scale variation in the human genome. Nat Genet 36:949–951
18. Friedman JM, Baross A, Delaney AD et al (2006) Oligonucleotide microarray analysis of genomic imbalance in children with mental retardation. Am J Hum Genet 79:500–513
19. Sebat J, Lakshmi B, Malhotra D et al (2007) Strong association of de novo copy number mutations with autism. Science 316:445–449
20. Lee C, Iafrate AJ, Brothman AR (2007) Copy number variations and clinical cytogenetic diagnosis of constitutional disorders. Nat Genet 39:S48–S54
21. Sharp AJ, Mefford HC, Li K et al (2008) A recurrent 15q13.3 microdeletion syndrome associated with mental retardation and seizures. Nat Genet 40:322–328
22. The International HapMap 3 Consortium and Principal investigators and Altshuler, David M and Gibbs, Richard A and Peltonen, Leena and Project coordination leaders and Altshuler, David M and Gibbs, Richard A and Peltonen, Leena and Dermitzakis, Emmanouil and Manuscript writing group and Schaffner, Stephen F and Yu, Fuli and Peltonen, Leena and Dermitzakis, Emmanouil and Bonnen, Penelope E and Altshuler, David M and Gibbs, Richard A and Genotyping, QC, de Bakker, Co-Leader, Paul IW et al (2010) Integrating common and rare genetic variation in diverse human populations. Nature 467:52–58
23. Bengtsson H, Irizarry R, Carvalho B et al (2008) Estimation and assessment of raw copy numbers at the single locus level. Bioinformatics 24:759–767
24. Bengtsson H, Wirapati P, Speed TP (2009) A single-array preprocessing method for estimating full-resolution raw copy numbers from all affymetrixgenotypingarraysincludingGenomeWideSNP 5 & 6. Bioinformatics 25:2149–2156
25. Ortiz-Estevez M, Bengtsson H, Rubio A (2010) ACNE: a summarization method to estimate allele-specific copy numbers for Affymetrix SNP arrays. Bioinformatics 26:1827–1833
26. Scharpf RB, Ruczinski I, Carvalho B et al (2011) A multilevel model to address batch effects in copy number estimation using SNP arrays. Biostatistics 12:33–50
27. Ritchie ME, Carvalho BS, Hetrick KN et al (2009) R/Bioconductor software for Illumina's Infinium whole-genome genotyping BeadChips. Bioinformatics 25:2621–2623
28. Staaf J, Vallon-Christersson J, Lindgren D et al (2008) Normalization of Illumina Infinium whole-genome SNP data improves copy number estimates and allelic intensity ratios. BMC Bioinformatics 9:409–409
29. Peiffer DA, Le JM, Steemers FJ et al (2006) High-resolution genomic profiling of chromosomal aberrations using Infinium whole-genome genotyping. Genome Res 16:1136–1148
30. Venkatraman ES, Olshen AB (2007) A faster circular binary segmentation algorithm for the analysis of array CGH data. Bioinformatics 23:657–663
31. Olshen AB, Venkatraman ES, Lucito R et al (2004) Circular binary segmentation for the analysis of array-based DNA copy number data. Biostatistics 5:557–572
32. Shah SP, Xuan X, DeLeeuw RJ et al (2006) Integrating copy number polymorphisms into array CGH analysis using a robust HMM. Bioinformatics 22:e431–e439
33. Greenman CD, Bignell G, Butler A et al (2010) PICNIC: an algorithm to predict absolute allelic copy number variation with microarray cancer data. Biostatistics 11:164–175
34. Yau C, Mouradov D, Jorissen RN et al (2010) A statistical approach for detecting genomic aberrations in heterogeneous tumor samples from single nucleotide polymorphism genotyping data. Genome Biol 11:R92
35. Li A, Liu Z, Lezon-Geyda K et al (2011) GPHMM: an integrated hidden Markov model for identification of copy number alteration and loss of heterozygosity in complex tumor samples using whole genome SNP arrays. Nucleic Acids Res 39:4928–4941
36. Ostrovnaya I, Nanjangud G, Olshen AB (2010) A classification model for distinguishing copy number variants from cancer-related alterations. BMC Bioinformatics 11:297–297
37. International HapMap Consortium, Frazer KA, Ballinger DG et al (2007) A second generation human haplotype map of over 3.1 million SNPs. Nature 449:851–861
38. Shah SP, Roth A, Goya R et al (2012) The clonal and mutational evolution spectrum of primary triple-negative breast cancers. Nature 486:395–399