superFreq

clonality tracking with SNVs and CNAs

Christoffer Flensburg
flensburg.c@wehi.edu.au

Ian Majewski
majewski@wehi.edu.au

October 8, 2015

Abstract

SuperFreq is an R software that identifies and track SNVs and CNAs over multiple samples.

Contents

1 Introduction

This software is built to provide an automated analysis of multiple cancer exomes, identifying SNVs and copy number alterations of different cell populations.

1.1 features

- Use coverage and SNP frequency for CNA calls.
- GC and MA corrections of coverage.
- uses pool of (potentially unrelated) normal samples to estimate variance.
- Matched normals used if available, but not required. Uses SNPs from CNA calling even without matched normal.
- Uses state of the art differential expression methods for coverage: limma-voom.
- Uses Log likelihood ratio with binomials for heterozygous SNP frequencies.
- Robust to misidentified heterozygous SNPs.
- Segments genome for CNA based on both coverage and SNPs.
- Ploidy calculation taking the uncertainties of each region into account.
- Does not assume limited number of clonalities.
- Compares somatic SNVs to pool of normal samples, as well as mapping and base quality.
- Tracks clonality of both SNVs and CNAs over sample.
- Allele specific: separates not only AAB from AAA, but also AAB from ABB between samples.
- Groups mutations (SNVs and CNAs) into clones.
- Identifies tree structures, including self consistency contraints.

2 Get it running

2.1 requirements

- linux
- VEP (it runs without it, throwing some warnings, but you dont get point variant annotation)
- At least two normal samples with the same capture regions, preferably prepared in the same way.

2.2 download the example with resources

Start by downloading the working example from XXX, which contains everything you need, including public resources such as reference genome, COSMIC data, dbSNP information, and so on.

```
wget URL.edu.au
```

Then move to the pipeline subfolder and open R.

```
cd ~/superFreq/testCNV/pipeline
R
```

Then run the example.

```
source('main.R')
```

This takes around 20 minutes depending on your hardware, and should generate a range of plots in testCNV/plots, as well as the results of the analysis in R-format in testCNV/R. A log of the run is printed to testCNV/R/runtimeTracking.log. The output should show a biallelic loss of TP53 in one of the samples: two SNVs and a loss. Specially check that variant annotation has worked, for example by making sure that somaticVariants.xls in the plots directory has properly annotated the TP53 SNVs as missense variants, and identified their presence in COSMIC.

2.3 insert your (meta) data

2.3.1 fill out meta data

Fill out the meta data of your samples in metaData.txt, such as:

```
BAM   VCF   INDIVIDUAL NAME TIMEPOINT  NORMAL
bam/normal.bam vcf/normal.vcf patient1 normalSample unrelatedTissue YES
bam/cancer.bam vcf/cancer.vcf patient1 cancerSample diagnosis YES
```

Note that spaces or tabs are interpreted as separators, so names cannot contain spaces. Any other characters like ".", "_" or "," will be replaced by "." using make.names(). Each sample is entered in one row, with the required columns

- BAM: The path to the .bam file of the sample, either relative from the metaData file, or absolute path. And index file also has to be present (with ".bam" replaced by ".bai" or ".bam.bai").
- VCF: The path to the .vcf file of the sample, either relative from the metaData file, or absolute path. These variants will be further QCed and filtered, but no positions not present in the .vcfs will be included, so the variant calling in the .vcf file should be liberal. More comments and methods to generate the .vcf in section 4.
- INDIVIDUAL: which individual the sample comes from. Samples from the same individuals will share variants, so that a detected SNV in any sample will be checked for in all samples from that indiviudal. Plots are made mainly to compare samples from the same individual, and clonal evolution is tracked over all samples from the same individual.
- NAME: the name of the sample. Has to be unique, and will be used as identifier in plots and output, so try to use human-readable names that are not too long. "P4.relapse", "P2.normal" or "DOHH2.resistant" are useful names, "AGRF0534563457_ATGTCTGCT_LANE003.dedup.bams is not.
- TIMEPOINT: A label indicating the timepoint of the sample, such as "diagnosis" or "relapse". Only used at a few places in output.

- NORMAL: "YES" if the sample is normal, "NO" otherwise. Normal samples will be used as matched normal for other samples from the same individual, mainly to separate germline variants from somatic mutations. The algorithm can tolerate a very low amount of cancer content (below 2% is pretty safe, 5% is on the limit) in a normal sample, as can be the case in remission samples in blood cancers. If in doubt, start by marking the same with "NO", as the pipeline still will identify that the somatic variants appear at much lower clonality in the normal, while the germline variants remain clonal in all samples. If after a first analysis you see that the aspiring normal sample indeed has a very low cancer content, you can change the metadata and rerun to get rid of the non-dbSNP germline variants called as somatic. The cancer content isn't explicitly called, but can be read out from the CNA plots, scatter plots or river plots for example.

2.3.2 add pool of normal samples

Next, add your pool of normal samples. Take any normal samples you have from the same capture regions and place (or link) them in a bam subdirectory of a new directory, such as testCNV/myPoolOfNormals/bam/myNormalSample*.bam. Note that the subdirectory must be called "bam". An index file is also required.

These bam files will be used as reference for CNA and SNV calling. They may overlap with the samples being analysed (the samples in the metaData.txt file), but it is not necessary. For example any matched normals used as reference normals should be added to the metaData.txt file as well, or the pipeline will not identify it as matched, and cannot use it to separate germline SNVs from somatic SNVs.

2.3.3 link data from main.R

Copy the file main.R in the pipeline subdirectory, and rename it to an identifier of the project, for example myProject.R. This is the file setting the parameters of the analysis. Change the R and plot directoy to new directories (they will be created as long as the parent directory exists) to avoid interference with the example run. The saved data from the run will be saved in the R directory, the output will be placed in the plot directory. Relative path from the pipeline directory, or absolute path.

```
normalDirectory = '../myPoolOfNormals'

Rdirectory = '../myProjectR'

plotDirectory = '../myProjectPlots'
```

2.3.4 link capture regions

Change the capture region path in myProject.R to a bed file of the UNPADDED capture regions of the exome. The pipeline looks for variants and reads up to 300 bp outside these regions. These capture regions are used for all samples, including the pool of normals. If you have samples done with different capture regions, it is best to split the data up into batches analysed separately. Link the capture regions relative to the pipeline directory, or as an absolute path:

```
captureRegionsFile = '../captureRegions/myCaptureRegions.bed'
```

2.4 Settings

Set the "cpus" setting in myProject.R to the number of parallel threads you want to use at most. As parallelisation is often over chromosomes, 6, 8 or 12 are good number, as it allows the 24 human chromosomes to be run in full batches.

```
cpus=12
```

The two parameters of the analysis, "systematicVariance" and "maxCov" control how much confidence to put in the coverage and SNP information respectively for the CNA calling. A low "systematicVariance" makes the pipeline more prone to segment the genome into smaller pieces based on the coverage, and larger values makes the pipeline more prone to keep larger segments. A large "maxCov" makes the pipeline more prone to segment the genome into smaller pieces based on the SNPs, and the other way around. Default values are 0.03 and 150 respectively, which gives a fairly conservative fragmentation (ie, large fragments). For increased sensitivity, at the cost of increased false calls as well, decrease "systematicVariance" towards 0 or increase "maxCov" towards infinity.

2.5 Run the analysis!

But wait! The analysis can take some time (hours per sample, including the pool of normals), so you probably want to open it in a screen if you are not running on your own computer.

go to the pipeline folder, open R and source myProject.R.

```
source('myProject.R')
```

The pipeline is fairly verbose with what is going on, and the log is stored in myProjectR/runtimeTracking.log. Diagnostic information is output during the run, both to the plots directory and to runtimeTracking.log.

3 Troubleshooting

- **I get an error related to downloading ensembl annotation.** Sometimes the ensembl server doesn't reply to download requests through R. Move the file ensemblhg19annotation.Rdata in the R directory of the example run to the R directory of your run.
- **I get warnings about VEP not running, and my SNVs aren't annotated.** The pipeline calls vep through system("vep -i [input] -o [output] –everything –force_overwrite –fork [cpus]"). If this call does not go through, the pipeline continues with unannotated SNVs. To fix, make sure VEP version 75 is callable from the command line with "vep". Alternatively, if you are brave, modify the call in pipeline/runVEP.R to match the format on your command line.

4 preparing BAM and VCF files

The pipeline assumes aligned bam files, and a preliminary variant calling. It is important that all files (inlcuding the pool of normals) are aligned in the same way, so that alignment artifacts are present consistently and can be handled. The pipeline performs extensive quality control and filtering of the variants in the vcf files, but no variants outside those supplied will be considered, so it is important that the preliminary variant calling sensitive rather than accurate. Only drawback of a larger vcf file is increased runtime. We normally align and do preliminaary variant calling with bwa, samtools and varscan with the following settings:

```
bwa mem -M -a
samtools mpileup -d 10000 -q 1 -Q 15 -A
java -jar VarScan.v2.3.6.jar mpileup2cns mpileup --variants --strand-filter 0
    --p-value 0.01 --min-var-freq 0.01
```

but other software may work as well or better.

5 Interpreting the output

The output are mainly split up into two kinds: diagnostic output that is meant to assess the quality of the data and analysis, and analytic output that is meant to extract biological

information about the samples. Both kinds are in the plots directory assigned in main.R, with the diagnostic output in the diagnostics subdirectory.

5.1 analytic output

5.1.1 Scatters

Scatter plots of the SNV frequency for every pair of samples within each individual (as assigned in the metaData.txt file). While the x and y-axis are very straight forward, there is abundant information in the way the dots are shown.

- **Point size** is set from the coverage, so that SNVs with low coverage that are less likely to be close to their true value are given less visual weight. The (linear) size is proportional to the square root of the geometric mean of the coverage in the two samples. The default setting is to hit size 1 at coverage 100 and cap the size at 1.5 (corresponding to 225 reads coverage).

- **Point type** is set from the dbSNP status. SNVs present in dbSNP are shown as black crosses, otherwise as blue dots. SNVs that are not in dbSNP, but behave as germline SNVs (consistent with 100% clonality in all samples of the individual) are represented as blue (horizontal-vertical) crosses. If all samples of an individual have close to 100% purity, then early somatic mutations (present clonally in all samples) may be misidentified as germline and can be represented as blue crosses. Note that there is no way to distinguish germline variants from early somatic variants in this case. A matched normal or a sample with purity significantly lower than 100% avoids this issue, as it allows the analysis to separate germline from somatic SNVs.

- **Redness** is set if there is a significant difference in frequency between the samples. A fisher exact test is performed, and the effective number of hypothesis N is set to the number of non-zero SNVs. The redness is then $-\log_{10}(p)/\log_{10}(N) - redCut$ where $redCut$ is set to 0.75 by default. The redness is limited to $(0, 1)$, and added to the existing colour, so that black crosses turn red, and blue dots go purple at redness 0.5 and then completely red at 1.

- **Orange rings** are added to SNVs that are altering the protein (severity <= 11 in somaticVariants). This links to the VEP data, which is only run for SNVs that have a somatic score larger than 0 (present in somaticVariants), so the assay is not done for all SNVs. The ring is thicker for lower severity, separating for example nonsense mutations from missense mutations.

- **Green rings** are added if the mutation is in a gene in the COSMIC consensus. Again, this is based on the VEP data so will only apply to SNVs with a somatic score larger than 0.

- **Gene names** are shown on the "named" plot above the SNV if the SNV has a sufficiently significantly different frequency. The idea is to show the gene names of somatic mutations.

For two normal clean human samples, you will find a cluster at $(0.5, 0.5)$ for the heterozygous germline SNPs, and a cluster at $(1, 1)$ for the homozygous SNPs. Most samples have some low frequency noise as well, which will turn up as a cloud of mainly blue dots (and the occasional cross) in the lower left corner.

A cancer sample (on the y-axis for example) compared to a normal will have the somatic SNVs along the y-axis, being mainly red dots (or blue/purple at low coverage or frequency). Copy number changes will show up as the $(0.5, 0.5)$ cloud sending out satelite clouds to the top and bottom. These SNPs (partially) losing heterozygousity will turn red if sufficiently far from 0.5 and sufficiently high coverage to gain significance.

A plot between two cancer samples can have clusters of blue/red dots anywhere in the lower left quadrant depending on clonalities of the cell population with the mutations, and can have clusters of dbSNP variants along the diagonal for shared CNAs, or along $x = 0.5$ or $y = 0.5$ for CNAs unique to one sample. If a different alleles are gained/lost in the two samples, you can see off-diagonal dnSNP clusters.

Each pair of sample have the default plot (called all.png), a version that also plots the SNVs flagged as low quality in the background as grey dots (flagged.png) and a version that shown gene names over the significantly different SNVs (named.png). SNVs are also plotted by chromosome, where each SNVs is linked to the position on the chromosome by colour as shown on the top of the plot.

5.1.2 SNV heatmap

SNV frequencies are also compared over all samples of each individual. Here the frequency is shown in a heatmap, where the samples are on the x-axis and the SNVs are on the y-axis. The samples and SNVs are clustered by similarity using the default R heatmap clustering, which helps show groups of SNVs that behave similarly. If a gene has two or more SNVs, it is assigned a colour which is shown in the left-side barcode (unassigned SNVs in genes appearing only once get a grey bar in the barcode). The colours are linked to the genes in a legend on the right side.

Note that the colourscale (shown on the left) has a sharp contrast from grey to black in the last few percent. The purpose of this is to easier separate SNVs that are present at very low frequency from SNVs that are not present at all. Missing data (no coverage over the position) is displayed as a white.

The same information is also shown as line plots on the next page of the .pdf, where each SNV is plotted as a line as function of samples.

The plot is repeated for three subsets of SNVs: first for dbSNPs (page 1-2), then for somatic SNVs that change significantly between samples (page 3-4) and last for all coding SNVs called in any sample (page 5-6).

5.1.3 CNA plots

The copy number calls are shown by sample, genomwide and by chromosome.

The log fold change (LFC) of the coverage of the sample compared to the (sex-corrected) pool of normal samples is shown in the top panel. The LFC is shown by gene as dimmed dots, where the size of the dot represents the accuracy (inversely proportional to the width of the moderated t-distribution from limma-voom). The identified segments are shown as opaque dots with horizontal lines representing the extension of the segment, and a horizontal error bar representing the uncertainty of the consensus LFC within the segment. Values falling outside of the plotted region are marked by an arrow with the value of the LFC printed next to it. For better readability, LFCs larger than 0 are coloured increasingly red, and LFC below 0 are coloured increasingly blue.

The second panel shows the frequency of the germline heterozygous SNPs, mirrored down to minor allele frequencies (MAF). The dimmed dots each represent the frequency within a gene (mean weighted by coverage if a gene has multiple heterozygous SNPs), and as above, the opaque dots represent the consensus within called CNA segments. Note that many individual genes will have a MAF lower than 0.5 due to fluctuations even if the region is a normal AB. For the segments however, a statistical test is made to check if all the SNPs in the segment are consistent with a 0.5 MAF, in which case the segment is plotted at $MAF = 0.5$. Due to this, a normal region will have the dots from the individual genes hovering around 0.45 at typical coverage, with the opaque line segment set to exactly 0.5.

If applicable, the CNA calls within the segments are shown at the bottom of the second panel. AB is normal, A means loss of an allele, AAB is gain, AA is copy number neutral loss of heterozygousity, etc. The third panel (if present) shows the clonality of the CNA (regions without CNA call, AB regions, are assigned a clonality of 1 ± 0.

5.1.4 river plots

The river plot (page 1 of the .pdf) represents the phylogenetic relationship of the clones, their clonalities over samples and the somatic mutations identifying the clone.

The pipeline performs the following steps:

- **Sanity checks input.**
- **Associates capture regions with genes**, using ensembl annotation.
- **Counts GC** and dinucleotide content in the capture regions.
- **Counts reads over the capture regions** of each sample (including pool of normals) using featureCounts.
- **Corrects the counts** for GC (dinucleotide) and MA bias. Sex is determined and properly taken into account.
- **Differential coverage is performed with limma-voom** both over capture regions and genes for each sample compared to the pool of normals.
- The genomic positions from the vcf files are shared within individuals, and the **positions are examined in the bam files**. Variants are flagged based on mapping quality, base quality, strand bias and stuttering.
- Positions present in any individual are examined in the pool of normal samples. The sample **variants are flagged if suspicious in the pool of normals**.
- **A somatic score is assigned to variants**, based on matched normal sample if present, or dbSNP otherwise.
- **Heterozygous germline SNPs are identified** in each sample.
- For each sample, **differential coverage and SNPs are summarised for each gene** over genome.
- Neighbouring genes with sufficiently similar differential coverage and SNP frequencies are clustered recursively, until a **segmentation of the genome is achieved** for each sample.
- **Consensus LFC and SNP frequency** is summarised for each segment.
- **LFC is renormalised** based on consistent CNAs for all regions, taking uncertainties in LFC and SNP frequency into account.
- **CNAs and clonality are called** in each segment, based on LFC and SNP frequency.
- The CNA calls undergo **post-analysis**, checking for neighbouring regions with similar CNA and clonality and other artifacts. Recurs to normalisation and CNA calling if segments are changed.
- With CNAs and clonalities determined, the **clonalities of somatic SNVs** are calculated, based on local CNA.

- **The clonalities of SNVs and CNAs are tracked over samples** from the same individual. CNAs are checked for direction in SNP frequency deviation, determining if the same or different alleles are gained/lost between samples.
- **Mutations (SNVs or CNAs) with similar clonalities in all samples of the individual are clustered into clones**. The germline clone (clonality 1 in all samples) is added to absorb germline mutations misidentified as somatic mutations.
- **The clones are sorted into a tree structure**, with smaller clones being assigned as subclones if possible. Checks for self consistency: the sum of the clonalities of disjoint subclones cannot be larger than the clonality of the containing clone.
- As afterburner, the **somatic SNVs are annotated using VEP** and comparing to COSMIC data. Plots and output are updated with this information.

The following is a more detailed description of the algorithm of the major steps. Although the important features are included, there will always be details and special cases that require special handling which may not always be mentioned in this description.

7.1 variant filtering

The variants important from the supplied .vcf files are quality assessed by importing the reads from the assocaited bam files. Any variants more than 300bp away from a capture region are discarded. The quality features checked and asociated flags are listed below:

- "Bq", **Base Quality**. If the variant reads have a significantly lower base quality than the reference reads (p ¡ 0.01, Mann-Whitney U-test) and the mean base quality is at least 10 lower. This flag is also used if the overall mean base quality is below 20, or strictly less than 10% of the variant reads achieve a base quality of 30.
- "Mq", **Mapping Quality**. If the variant reads have a significantly lower mapping quality than the reference reads (p ¡ 0.01, Mann-Whitney U-test) and the mean mapping quality is at least 10 lower. This flag is also used if the overall mean mapping quality is below 20, or strictly less than 10% of the variant reads achieve a mapping quality of 30.
- "Sb", **Strand Bias**. If the variant reads have a significantly different strand ratio than the reference reads (p ¡ 0.001, Fisher's exact test).
- "St" **STuttering**. If the variant is equivalent to an elongation or shortening of a stretch of at least 20 repeated base pairs.

The variants are also compared to the pool of normal samples, and a set of new flags are assigned to variants that behave suspiciously in the normals:

- "Nnc" and "Nnn", **Normal Noise Consistent or Non-consistent**. If the variant is present at more than 10% in any of the normal samples. Consistency is determined based on whether all normal samples are consistent with the same background frequency (fisher's exact test, p ¿ 0.01). Variants in dbSNP are allowed to have frequencies consistent with 0.5 or 1 as well without being flagged.
- "Mc", **Many Copies**. Variants that have more than 10 times the median coverage, summed over all normal samples. This is often associated with regions that are present multiple times in the human genome, but only present once in the reference, leading to inflated coverage and heterozygous germline variants deviating from 50%.

7.2 Coverage analysis

The coverage of each sample is compared to the coverage of the pool of normal samples, using limma-voom. First, fragments are counted over each of the padded (300bp on each side) capture regions for all samples, including the pool of normals. The counts are then corrected for:

- GC content by capture region. A loess curve is fitted to $\log(N_i/L)$ as function of GC content for each sample i, where N_i is the number of reads over a capture region in sample i, and L is the length of the capture region. The loess fit is weighted by $\sqrt{\langle N_i \rangle_i}$ with i running over all samples. The counts are then divided by the value of the weighted loess fit, maintaining total read count.

- MA-bias by capture region. The log fold change (M) of the counts of each sample comapred to the sum of the refence samples are plotted against half the logarithm of the product of the counts (A). A loess fit is made to the curve, and the counts are corrected to flatten the curve while maintaining total read count.

- The pool of normal samples are sex corrected to two copies of every chromosome, meaning that male samples have their X and Y chromosome counts doubled while female sample maintain their X chromosome counts but have their Y chromsome counts removed from the analysis.

The corrected counts are pooled by gene and analysed for differential coverage using limma-voom.

8 Examples

8.1 Known monosomy 7

In figure 1 we show the CNA calls of an AML patient with monosomy 7, as known from karyotyping (right?). The monosomy is accurately called, supported by both the coverage and SNP frequencies. Chromosome Y is completely lost but chromosome X is maintained at two copies, as is expected from a female. A matched normal was available, and 6 unrelated normal samples were used in the pool of normals. A region around the centromere has slightly higher coverage and is separated into a separate segment. A single gene with a much lower coverage is called as complete loss with similar clonality as the monosomy. This is just a very simple proof of concept, showing that the pipeline is capable of identifying almost clonal CNAs covering an entire chromosomes.

8.2 Normal sample: false positive rate

We here call CNAs on a normal sample, using another 5 normal samples as pool of normals, but no matched normal. The results in figure 2 show that only a clonal loss of chromosome X is called, indicating that the sample is from a male individual. The pool of normal samples contain only samples identified as females, so CNA calling could not be performed on chromosome Y and the pipeline excluded those regions in the CNA calling.

The complete absence of false calls, not even small regions, separates superFreq from many other CNA callers, that often call small CNAs very liberally. This property allows superFreq to include all CNA calls in the downstream clonal tracking without fitlering out small CNAs. If a loss or amplification of a single gene is called, it has a decent probablity of being real.

The amount of false calls of course depends on the sample, but superFreq has mechanisms in place to measure the noise level of samples and increases variance estimates if needed. This is done by comparing neighbouring genes that in a majority of cases will share the true LFC, and thus allows an empirical estimate of the variance within a single sample, as well as the between-sample estimates used within the pool of normals.

8.3 cancer sample with complicated behaviour

Take some cancer sample with a lot of CNAs and show that segmentation, calls and normalisation works as intended

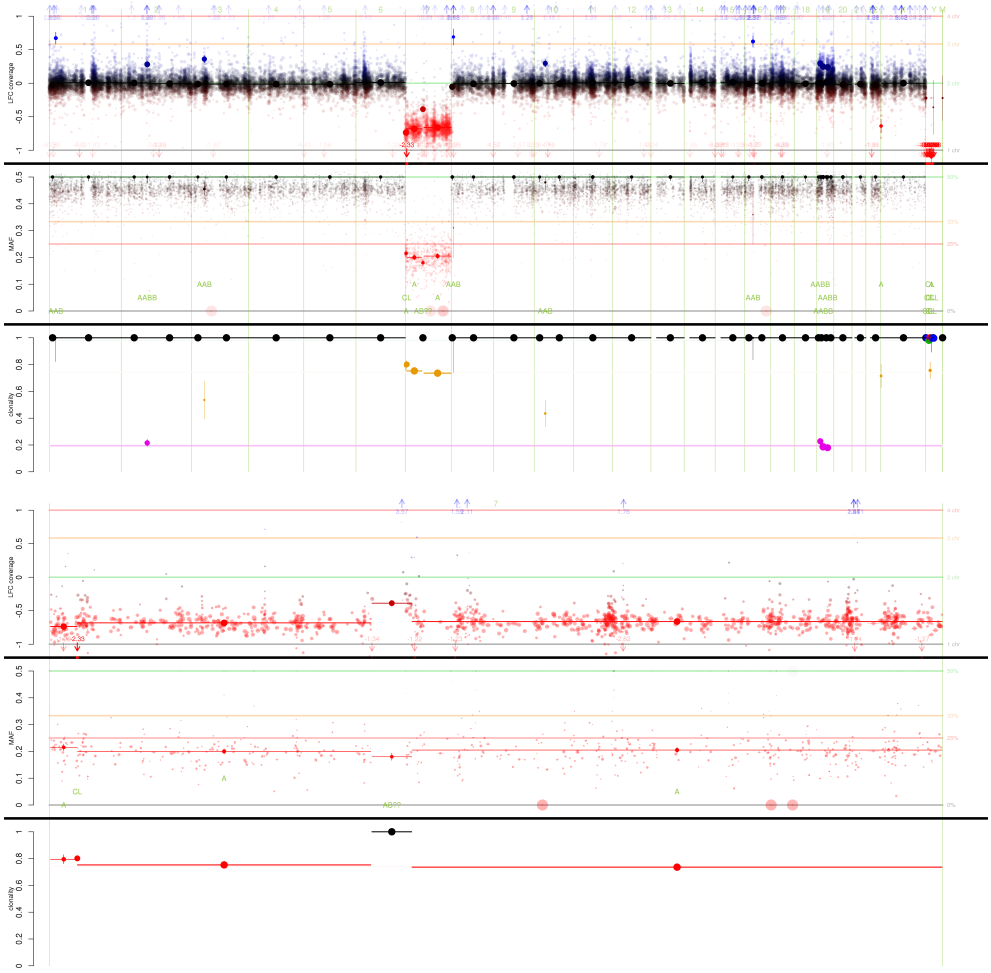

Figure 1: The CNA calls from an AML patient with known monosomy 7. Top figure shows the entire genome, bottom figure only chromosome 7. The top panels of each figure show the LFC compared to the pool of normal samples. The middle panels show the minor allele frequency of the germline heterozygous SNPs with the called CNA of each segment. The bottom panels show the called clonality.

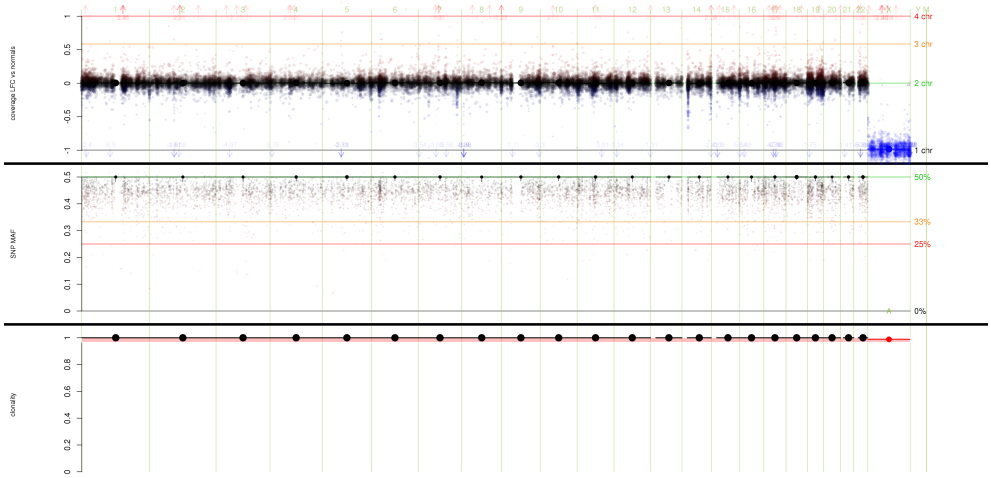

Figure 2: The CNA calls from a normal sample. The top panel of each figure shows the LFC compared to the pool of normal samples. The middle panel shows the minor allele frequency of the germline heterozygous SNPs with the called CNA of each segment. The bottom panel shows the called clonality.