# FACETS: A Package for Analyzing Allele-specific Copy Number and Clonal Heterogeneity from High-throughput Sequencing

Venkatraman E. Seshan[1] and Ronglai Shen[2]

August 9, 2016

[1]Department of Epidemiology and Biostatistics
Memorial Sloan-Kettering Cancer Center
seshanv@mskcc.org

[2]Department of Epidemiology and Biostatistics
Memorial Sloan-Kettering Cancer Center
shenr@mskcc.org

## Contents

## 1 Overview

This document presents an overview of the `FACETS` package. This package is for analyzing allele-specific DNA copy number and clonal heterogeneity from high-throughput sequencing including whole-genome, whole-exome, and some targeted cancer gene panels. The method implements a bivariate genome segmentation, followed by allele-specific copy number calls. Tumor purity, ploidy, and cellular fractions are estimated and reported from the output.

## 2 Data

The file `stomach.csv.gz` in `extdata` directory contains data from a TCGA stomach cancer sample. To generate these data, tumor and normal bam files were downloaded from CGHub `https://cghub.ucsc.edu/`. Reference and variant allele read counts were extracted from the bam files for germline polymorphic sites catalogued in the dbSNP and 1000genome database ($\sim$ 1.9 million polymorphic positions). C++ code to generate such read count files and instructions for using it are available in `extcode` directory. This program generates csv file in which the first 4 columns are: Chormosome, Position, Ref and Alt. Then for each bam file 4 columns with names File#R, File#A, File#E and File#D giving the counts of number of reads with the ref allele, alt allele, errors (neither ref nor alt) and deletions in that position. The hashmark (#) is the number in position of the file in the sequence of bam files given to the pileup code. In the included data the ref and alt allele information were removed and random noise added to the SNP positions to deidentify the case.

```
> datafile = system.file("extdata", "stomach.csv.gz", package="facets")
> head(read.csv(datafile))

  Chromosome Position File1R File1A File1E File1D File2R File2A File2E File2D
1          1    69424    170    117      0      0    158    103      0      0
2          1    69515      0     76      0      0      0     77      0      0
3          1    69536    103      0      0      0     99      0      0      0
4          1   808866     96      0      0      0    133      0      0      0
5          1   809120     66      0      0      0    105      0      0      0
6          1   809176     79      0      0      0    126      0      0      0
```

## 3 An Example

Here we perform an analysis on the stomach cancer whole-exome described above.

```
> library(facets)
```

We first perform various pre-processing steps to prepare the data for segmenation analysis. Positions with total read count below a lower depth threshold (default 35, use ndepth to change the default value) or exceed an upper threshold ($> 1000$) (excessive coverage) in the matched normal sample were removed. We scan all positions by 150-250 bp interval to space out SNP-dense regions to reduce local patterns of serial dependencies that can result in hyper-segmentation in the downstream steps. Read depth ratio between tumor and normal gives information on total copy number. The variant (non-reference) allele frequency at heterozygous loci (germline variant allele frequency greater than 0.25 or less than 0.75) contain information on allelic imbalance. This pre-processing procedure on average yields 250K SNP loci from TCGA whole-exomes that pass these quality filters, and 10% of which are heterozygous. At each position, logR is defined by the log-ratio of total read depth in the tumor versus that in the normal and logOR is defined by the log-odds ratio of the variant allele count in the tumor versus in the normal. A normalizing constant is calculated for each tumor/normal pair to corrected for total library size, and GC-bias is corrected using a loess regression of logR over GC content along 1kb windows along the genome.

```
> set.seed(1234)
> datafile = system.file("extdata", "stomach.csv.gz", package="facets")
> rcmat = readSnpMatrix(datafile)
> xx = preProcSample(rcmat)
```

A bivariate genome segmentation is performed on logR and logOR by extending the CBS algotithm (Olshen et al., 2004; Venkatraman and Olshen, 2007) to the bivariate scenario using a $T^2$ statistic for identifiying change points. If the maximal statistic is greater than a pre-determined critical value (cval), we decalre a change exists and the change points that maximize this statistic. Lower cval lead to higher sensitivity for small changes. After segmentation, a clustering process is applied to group the segments into clusters of the same underlying genotype.

```
> oo=procSample(xx,cval=150)
```

We note that logR estimates are proportional to the absolute total copy number up to a location constant. In diploid genome, logR = 0 is the normal 2-copy state. However, aneuploidy can lead to systematic shift of the normal diploid state. In order to obtain correct genotype calls for copy number, we need to identify the location of the normal diploid state. We use the logOR summary measure estimates to identify the segment clusters in allelic balances, and use these segments to determine the 2-copy state. We call the logR for the 2-copy state $logR_0$ which is output below.

```
> oo$dipLogR
```

```
[1] -0.04444348
```

Call allele-specific copy number and associated cellular fraction, estimate tumor purity and ploidy.

```
> fit=emcncf(oo)
```

Once the logR value for the diploid state is obtained we calculate the observed copy number for each cluster as $\exp(\mathrm{logR}_c - \mathrm{logR}_0)$ where $\mathrm{logR}_c$ is the logR summary for the cluster and $\mathrm{logR}_0$ is the diploid state level. Once the observed total number is obtained we obtain the allele specific copy numbers m and p and the cellular fraction $\phi$ using the logOR data. The cellular fraction is associated with the aberrant genotype. For clonal copy number alterations, $\phi$ equals tumor purity. For subclonal events, $\phi$ will be lower than the overall sample purity.

To further refine these initial estimates and obtain a genome-wide optimization, we apply a genotype mixture model and maximize a joint likelihood that summarizes over all SNP loci and segment clusters across the genome. An expectation-maximization (EM) algorithm is used for the estimation procedure. It can be viewed as an estimation problem with the latent copy number states as “missing” data. In the E-step of the EM procedure, Bayes theorem is used to compute the posterior probability of a segment cluster being assigned copy number state g given the parameter estimates at the kth iteration. In the M-step, given the imputed genotype, we update the model parameters by maximizing the complete-data likelihood. This procedure is iterated until convergence. For a segment, if the number of heterozygous SNPs is less than 15, only the total copy number is given. For male gender, the number of copy for chromosome X is 1 in the normal. The X chromosome copy number in the tumor is adjusted accordingly.
output file

```
> head(fit$cncf)

  chrom seg num.mark nhet cnlr.median        mafR segclust cnlr.median.clust
1     1   1     2631  158  0.50028101 0.305665373       12        0.49145998
2     1   2     4598  249 -0.03699913 0.021520755        7       -0.04444348
3     1   3     6342  408  0.49218666 0.404039235       12        0.49145998
4     2   4     7006  375 -0.03654935 0.019844538        7       -0.04444348
5     2   5       14    0 -0.85252562 0.000000000        6       -0.88771396
6     2   6     2299   96 -0.05328541 0.001226293        7       -0.04444348
  mafR.clust     start       end     cf.em tcn.em lcn.em
1 0.38807104     69424  29651873 0.8767632      3      1
2 0.01655667  31188034 144922109 1.0000000      2      1
3 0.38807104 144922463 249212378 0.8767632      3      1
4 0.01655667     41507 197673978 1.0000000      2      1
5         NA 197706054 197873655 0.8919360      1      0
6 0.01655667 197947995 243061179 1.0000000      2      1
```

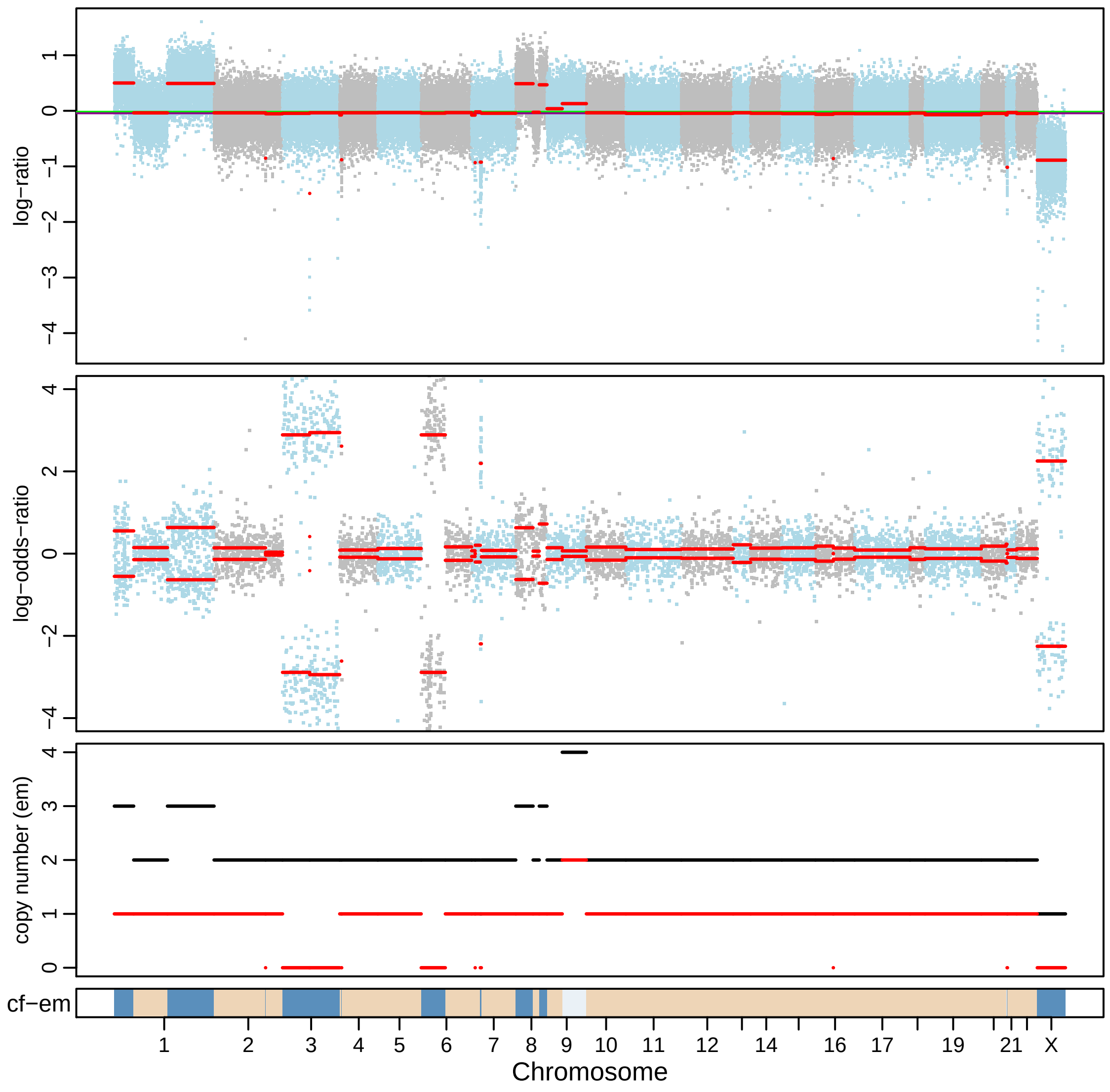
In the output, cf, tcn, lcn are the initial estimates of cellular fraction, total and minor copy number estimates, and cf.em, tcn.em, lcn.em are the estimates by the mixture model optimized using the EM-algorithm. cf is used as initial values for the EM algorithm. For diploid normal segments (total copy=2, minor copy=1), we report cellular fraction as 1 (100% normal).

Estimated tumor sample purity and ploidy are reported:

```
> fit$purity

[1] 0.891936

> fit$ploidy

[1] 2.070151
```

We provide a plot function to visualize the genome-wide profile and FACETS output.

```
> plotSample(x=oo,emfit=fit)
```



The top panel of the figure displays logR with chromosomes alternating in blue and gray. The green line indicates the median logR in the sample. The purple line indicates the logR of the diploid state. The second panel displays logOR. Segment means are ploted in red lines. The third panel plots the total (black) and minor (red) copy number for each segment. The bottom bar shows the associated cellular fraction (cf). Dark blue indicates high cf. Light blue indicates low cf. Beige indicates a normal segment (total=2,minor=1).

The emcncf algorithm allows a separate cellular fraction estimate for each segment cluster. In order to impose a clonally constrained structure on $\phi$, we included a sequential approach where the modified algorithm starts with a single clonal cluster (k=1) with cellular fraction parameter $\phi_1$. We then identify segment clusters for which segment cluster-specific estimates is non-trivially lower (at least by 0.05 although this can be changed by setting desired value for difcf) from the clonally constrained estimates that result in a suboptimal fit under k=1. These segment clusters with discordant cellular fraction estimates then form a candidate subclonal cluster of events at a lower cellular fraction $\phi_2$, and a model is fitted with the joint likelihood optimized under k=2. This procedure is iterated until no additional discordance in cellular fraction estimates are found, or a specified maximum k is reached. In the default parameter setting, a maximum k=5 is allowed although user can change it to a higher number if greater intratumor heterogeneity is expected. In the output, the estimated $\phi_1$ is the cellular fraction for the clonal events and also the tumor purity by definition, and the estimated $\phi_k, k > 1$ for any subclonal clusters identified in the tumor sample.

```
> fit2=emcncf2(oo)
```

```
fitting 1 clonal cluster ...
```

output file

```
> head(fit2$cncf)
```

```
  chrom seg num.mark nhet cnlr.median        mafR segclust cnlr.median.clust
1     1   1     2631  158  0.50028101 0.305665373       12        0.49145998
2     1   2     4598  249 -0.03699913 0.021520755        7       -0.04444348
3     1   3     6342  408  0.49218666 0.404039235       12        0.49145998
4     2   4     7006  375 -0.03654935 0.019844538        7       -0.04444348
5     2   5       14    0 -0.85252562 0.000000000        6       -0.88771396
6     2   6     2299   96 -0.05328541 0.001226293        7       -0.04444348
  mafR.clust     start       end    cf.em tcn.em lcn.em clonal.cluster
1 0.38807104     69424  29651873 0.891936      3      1              1
2 0.01655667  31188034 144922109 1.000000      2      1              1
3 0.38807104 144922463 249212378 0.891936      3      1              1
4 0.01655667     41507 197673978 1.000000      2      1              1
5         NA 197706054 197873655 0.891936      1      0              1
6 0.01655667 197947995 243061179 1.000000      2      1              1
```

## References

Olshen, A. B., Venkatraman, E. S., Lucito, R., and Wigler, M. (2004). Circular binary segmentation for the analysis of array-based dna copy number data. *Biostatistics*, 5:557–72.

Venkatraman, E. S. and Olshen, A. B. (2007). A faster circular binary segmentation algorithm for the analysis of array cgh data. *Bioinformatics*, 23:657–63.