# SomatiCA: identifying, characterizing and quantifying somatic copy number aberrations from cancer genome sequencing

Mengjie Chen, Hongyu Zhao

17 Jan 2013

## 1 Introduction

This guide provides a tour of the Bioconductor package `SomatiCA`, a R package that is capable of identifying, characterizing and quantifying somatic CNAs from cancer whole genome sequencing. It is especially designed for somatic copy number analyses taking into account: (i) an unknown fraction of normal cells (admixture rate) that are nearly always intermixed with cancer cells; and (ii) the heterogeneity of cancer cell population owing to ongoing subclonal evolution. The package implements a pipeline for characterizing somatic copy number aberrations based on different statistical methods: segmentation on Lesser Allele Frequency (LAF), Maximum Likelihood Estimation of somatic ratio (read depth ratio of tumor/ normal), admixture rate estimation by a Bayesian finite mixture model and subclonality characterization based on hypothesis testing. It is especially suitable for studies designed to understand tumor evolution. It currently works for cancer sample with control.

This manual is composed of basic test run on each module with a simulated dataset. The main purpose is to help users familiar with basic functions and the data format.

### 1.1 User Guide

A user guide with detailed analysis on real data, is available as part of the online documentation. To reach the User's Guide, install the SomaticCNA package and load it into an R session by `library(SomatiCA)`. In R for Windows, the User's Guide will then be available from the drop-down menu called "Vignettes". In other operating systems, type

```
> library(SomatiCA)
> SomatiCAUsersGuide()
```

```
[1] "D:/biocbld/bbs-3.0-bioc/tmpdir/RtmpoLkOzh/Rinst171c6bab6f9f/SomatiCA/doc/SomatiCA
```

at the R prompt to open the User's Guide in a pdf viewer.

# 2 Overview of capabilities

## 2.1 Input

Let us simulate a simple dataset with 1450 snps on 2 chromosome.

```
> set.seed(1)
> rawLAF <- c(rnorm(300, 0.2, 0.05), rnorm(300, 0.4, 0.05),
+             rnorm(200, 0.3, 0.05), rnorm(200, 0.2, 0.05),
+             rnorm(200, 0.3, 0.05), rnorm(250, 0.4, 0.05))
> rawLAF <- ifelse(rawLAF>0.5, 1-rawLAF, rawLAF)
> germLAF <- c(rnorm(800+650, 0.4, 0.05))
> germLAF <- ifelse(germLAF>0.5, 1-germLAF, germLAF)
> reads1 <- c(rpois(300, 25), rpois(300, 50), rpois(200, 60),
+             rpois(200, 25), rpois(200, 40), rpois(250, 50))
> reads2 <- rpois(800+650,50)
> chr <- c(rep("chr1", 800), rep("chr2", 650))
> position <- c(seq(1, 16000000, by=20000), seq(1, 13000000, by=20000))
> zygo <- rep("het", 800+650)
> data <- data.frame(chr, as.integer(position), as.character(zygo), as.integer(reads
>
```

The input of *SomatiCA* is formatted as a GRanges instance as defined in *GenomicRanges*.

```
> colnames(data) <- c("seqnames", "start", "zygosity", "tCount",
+                     "LAF", "tCountN", "germLAF")
> input <- SomatiCAFormat(data, missing = F, verbose = T)
> input

GRanges object with 1450 ranges and 5 metadata columns:
         seqnames               ranges strand   |
            <Rle>            <IRanges>  <Rle>   |
     [1]     chr1       [    1,     1]      *   |
     [2]     chr1       [20001, 20001]      *   |
     [3]     chr1       [40001, 40001]      *   |
     [4]     chr1       [60001, 60001]      *   |
     [5]     chr1       [80001, 80001]      *   |
     ...      ...                  ...    ... ...
  [1446]     chr2 [12900001, 12900001]      *   |
  [1447]     chr2 [12920001, 12920001]      *   |
  [1448]     chr2 [12940001, 12940001]      *   |
  [1449]     chr2 [12960001, 12960001]      *   |
  [1450]     chr2 [12980001, 12980001]      *   |
```

```
         zygosity    tCount       LAF   tCountN
       <character> <integer> <numeric> <integer>
   [1]         het        28 0.1686773        51
   [2]         het        24 0.2091822        40
   [3]         het        25 0.1582186        53
   [4]         het        18 0.2797640        58
   [5]         het        33 0.2164754        49
   ...         ...       ...       ...       ...
[1446]         het        44 0.3354413        50
[1447]         het        45 0.4453190        46
[1448]         het        49 0.2602019        43
[1449]         het        48 0.4116780        51
[1450]         het        44 0.3865482        40
         germLAF
       <numeric>
   [1] 0.3670727
   [2] 0.4031673
   [3] 0.4027238
   [4] 0.4123505
   [5] 0.4062018
   ...       ...
[1446] 0.3966479
[1447] 0.4220027
[1448] 0.3239019
[1449] 0.4054811
[1450] 0.4094522
-------
seqinfo: 2 sequences from an unspecified genome; no seqlengths
```

## 2.2 Segmentation

Given input in the format of a GRanges object, `larsCBSsegment()` segments each chromosome with LAF of heterozygous sites on that chromosomes. `larsCBSsegment()` extracts hetergozygous sites by 'grep' any site with 'het' in the column of zygosity. For Complete Genomic data, sites with zygosity of 'het-ref' and 'het-alt' will be extracted. If genotype calling results are obtained from other platforms, a transformation is needed. For example, the '0/1/2' coding from VCF file will be needed to transform to 'het'/'hom' coding. Users can name 'zygosity' in their own way but keep in mind that only names containing 'het' will be used as heterozygous sites for segmentations in SomatiCA, such as 'het', 'het-ref', 'heter', 'heter1' etc.

`larsCBSsegment()` firstly calls a function `denoise()` to smooth the outliers. Then it segment each chromosome with CBS followed by a model selection procedure. The

default model selection criteria is Bayesian Information Criterion (BIC). Users can set `rss=T` to apply BIC plus a minimum cut-off for change in residue sum of squares (RSS) between neighboring change points. `collapse.k` is an option to average LAF on each k SNPs.

```
> seg <- larsCBSsegment(input, collapse.k = 0, ncores = 1, verbose = T, rss = FALSE)
```

This is a toy example without much noise. Consider to use rss=T to select change points from sequencing data. Output of `larsCBSsegment()` includes two part: segmentation results and heterozygous sites used for segmentation (denoised).

```
> seg
```

```
$segment
GRanges object with 6 ranges and 2 metadata columns:
      seqnames               ranges strand |    medLAF
         <Rle>            <IRanges>  <Rle> | <numeric>
  [1]     chr1 [       0,  5900001]      * |    0.1981
  [2]     chr1 [ 5900001, 11900001]      * |    0.3976
  [3]     chr1 [11900001, 15980001]      * |    0.2961
  [4]     chr2 [       0,  3900001]      * |    0.2068
  [5]     chr2 [ 3900001,  7920001]      * |    0.2936
  [6]     chr2 [ 7920001, 12980001]      * |    0.3977
       medgLAF
      <numeric>
  [1]    0.4009
  [2]    0.3964
  [3]    0.3908
  [4]    0.4037
  [5]    0.3977
  [6]    0.4036
  -------
  seqinfo: 2 sequences from an unspecified genome; no seqlengths

$hetsites
GRanges object with 1450 ranges and 5 metadata columns:
         seqnames               ranges strand   |
            <Rle>            <IRanges>  <Rle>   |
     [1]     chr1       [    1,     1]      *   |
     [2]     chr1       [20001, 20001]      *   |
     [3]     chr1       [40001, 40001]      *   |
     [4]     chr1       [60001, 60001]      *   |
     [5]     chr1       [80001, 80001]      *   |
     ...      ...                   ...    ... ...
```

```
  [1446]     chr2 [12900001, 12900001]      *   |
  [1447]     chr2 [12920001, 12920001]      *   |
  [1448]     chr2 [12940001, 12940001]      *   |
  [1449]     chr2 [12960001, 12960001]      *   |
  [1450]     chr2 [12980001, 12980001]      *   |
           zygosity    tCount       LAF   tCountN
         <character> <integer> <numeric> <integer>
     [1]         het        28 0.1686773        51
     [2]         het        24 0.2091822        40
     [3]         het        25 0.1582186        53
     [4]         het        18 0.2797640        58
     [5]         het        33 0.2164754        49
     ...         ...       ...       ...       ...
  [1446]         het        44 0.3354413        50
  [1447]         het        45 0.4453190        46
  [1448]         het        49 0.2602019        43
  [1449]         het        48 0.4116780        51
  [1450]         het        44 0.3865482        40
          germLAF
         <numeric>
     [1] 0.3670727
     [2] 0.4031673
     [3] 0.4027238
     [4] 0.4123505
     [5] 0.4062018
     ...       ...
  [1446] 0.3966479
  [1447] 0.4220027
  [1448] 0.3239019
  [1449] 0.4054811
  [1450] 0.4094522
  -------
  seqinfo: 2 sequences from an unspecified genome; no seqlengths
```

The segmentation result can be examined by `plotSegment()`. If multiple chromosomes are segmented, use `k` to plot segmentation on each chromosome respectively by their orders in the input.

```
> plotSegment(seg$segment, input, k = 1, smooth = F)
```

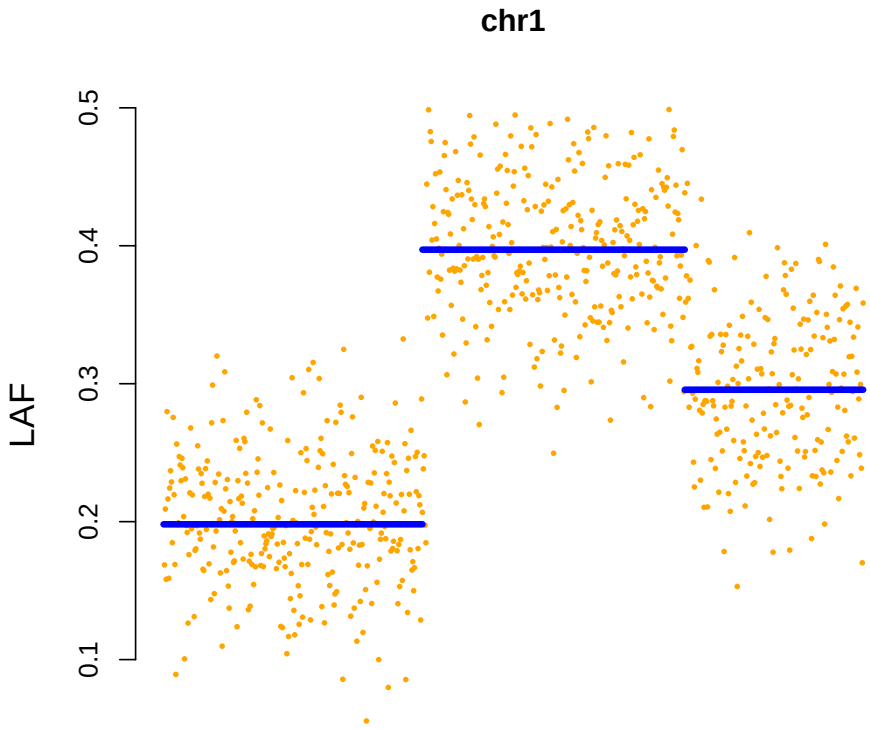


Figure 1: Segmentation on chr1 based on LAF.

## 2.3 Estimation of somatic ratio

Somatic ratio is defined as the ratio of read depths between a tumor and its paired normal sample for a given segment. SomatiCA implements different methods to estimate somatic ratio. For the "mle" method somatic ratio is estimated by a maximum likelihood approach. For the "mean" method, somatic ratio is estimated by the ratio between mean of tumor sample and normal sample. For the "geometric", somatic ratio is estimated by geometric mean of somatic ratios of all sites in a given segment. To estimate somatic ratio, both segmentation and input with read depths are required.

```
> segmentwithRatio <- somaticRatio(seg$segment, input, method = "mle")
> segmentwithRatio

GRanges object with 6 ranges and 3 metadata columns:
      seqnames               ranges strand |    medLAF
         <Rle>            <IRanges>  <Rle> | <numeric>
  [1]     chr1 [       0,  5900001]      * |     0.198
  [2]     chr1 [ 5900001, 11900001]      * |     0.397
  [3]     chr1 [11900001, 15980001]      * |    0.2956
  [4]     chr2 [       0,  3900001]      * |    0.2065
  [5]     chr2 [ 3900001,  7920001]      * |     0.293
  [6]     chr2 [ 7920001, 12980001]      * |    0.3986
       medgLAF     ratio
     <numeric> <numeric>
  [1]    0.4009     0.509
  [2]    0.3964     0.979
  [3]    0.3908     1.191
  [4]    0.4037     0.501
  [5]    0.3977     0.805
  [6]    0.4036     0.997
  -------
  seqinfo: 2 sequences from an unspecified genome; no seqlengths
```

## 2.4 Refine segments

Two adjacent segments are merged if the difference in the somatic ratios is less than "threshold2", which is tunable in the implementation with its default value being 0.05, equivalent to 5% change in somatic copy-number without normal contamination. The MLEs of the somatic ratio for the refined segments are recalculated. This refinement procedure is applied repeatedly until no adjacent segments have somatic ratio difference less than T. "threshold1" is the threshold used to merge the segments based on median LAF.

```
> refined <- refineSegment(segmentwithRatio, input, method = "mle",
+                          adjust = TRUE, threshold1 = 0 , threshold2 = 0.05)
> refined

GRanges object with 6 ranges and 3 metadata columns:
      seqnames               ranges strand |    medLAF
         <Rle>            <IRanges>  <Rle> | <numeric>
  [1]     chr1 [       0,  5900001]      * |     0.198
  [2]     chr1 [ 5900001, 11900001]      * |     0.397
  [3]     chr1 [11900001, 15980001]      * |    0.2956
  [4]     chr2 [       0,  3900001]      * |    0.2065
  [5]     chr2 [ 3900001,  7920001]      * |     0.293
  [6]     chr2 [ 7920001, 12980001]      * |    0.3986
       medgLAF     ratio
     <numeric> <numeric>
  [1]    0.4009     0.509
  [2]    0.3964     0.979
  [3]    0.3908     1.191
  [4]    0.4037     0.501
  [5]    0.3977     0.805
  [6]    0.4036     0.997
  -------
  seqinfo: 2 sequences from an unspecified genome; no seqlengths
```

## 2.5 Estimation of admixture rate

The estimation of the admixture rate is accomplished by fitting the somatic copy number (somatic ratio*2) of all segments with a Bayesian finite mixture model, with components centered at the discrete levels. Each segment was assigned with a discrete level based on corresponding posterior probability. Segments with ambiguous assignments will be classified as candidate subclonal events and excluded from admixture rate inference. The admixture rate will be estimated by an optimal solution contributed by explanation of tumor copy number with all remaining segments as integer level. *SomatiCA* implements a Markov Chain Monte Carlo with Metropolis Hasting algorithm to estimate posterior probabilities of the Bayesian finite mixture model. Option `mcmc` set the number of MCMC iteration, `burnin` set the number of MCMC iteration for burnin and `p` set the cutoff of posterior probability for ambiguous integer copy number assignments.

```
> ll <- admixtureRate(refined, mcmc = 5000, burnin = 1000, p = 0.01)
> admix <- ll$admix
```

`copynumberCorrected` could take a segmentation profile and an admixture rate to calculate the integer somatic copy number in tumor sample then characterize somatic

events based on corrected somatic ratio. Each segment will be annotated with "=", "Loss", "Gain", "LOH", "neutral LOH" or "double deletion". Note that somatic copy number here is calculated from the ratio with the assumption that the control sample is diploid. This result will likely be modified in next step with the calculation of ploidy of each segment in the control sample.

```
> y <- copynumberCorrected(segmentwithRatio, admix)
> y
```

```
GRanges object with 6 ranges and 5 metadata columns:
    seqnames               ranges strand |    medLAF
       <Rle>            <IRanges>  <Rle> | <numeric>
  1     chr1 [       0,  5900001]      * |    0.1980
  2     chr1 [ 5900001, 11900001]      * |    0.3970
  3     chr1 [11900001, 15980001]      * |    0.2956
  4     chr2 [       0,  3900001]      * |    0.2065
  5     chr2 [ 3900001,  7920001]      * |    0.2930
  6     chr2 [ 7920001, 12980001]      * |    0.3986
      medgLAF     ratio  somaCN       event
    <numeric> <numeric> <array> <character>
  1    0.4009     0.509       1         LOH
  2    0.3964     0.979       2           =
  3    0.3908     1.191       2           =
  4    0.4037     0.501       1        Loss
  5    0.3977     0.805       2           =
  6    0.4036     0.997       2           =
  -------
  seqinfo: 2 sequences from an unspecified genome; no seqlengths
```

## 2.6 Subclonality characterization

*SomatiCA* estimates subclonality for each somatic copy number abberation. To do this, it first calculates allelic copy number nB and nA (segment level allelic copy number is estimated by median in that segment) in a control sample based on GC corrected read counts. SomatiCA tests whether copy number change in corresponding tumor sample can result in a change of exactly one copy of one allele. If the somatic ratio (corrected by admixture rate) in the corresponding tumor sample is greater than 1, SomatiCA tests for one copy gain, otherwise it tests for one copy loss. With null hypothesis that clonal copy number ratio follows a normal distribution, p-value is calculated for each segment as the probability of obtaining a copy number ratio at least as extreme as the one that was actually observed. Segments with p-value less than 0.05 are classified as subclonal.

```
> data(GCcontent)
> segmentGCcorrected <- segmentGCbiasRemoval(y, input, GCcontent)
```

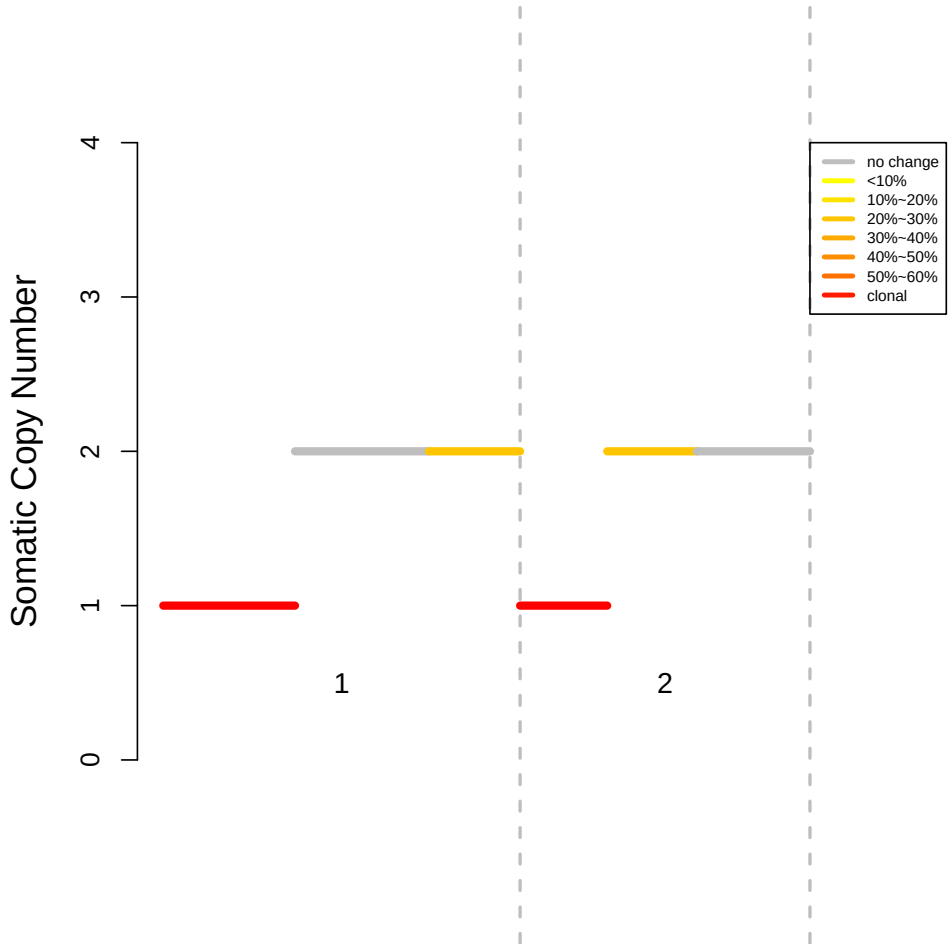


Figure 2: An example of subclonality.

```
> segmentClonality <- subclonality(segmentGCcorrected, admix)
```

Users can further merge neighboring segments with same somatic copy number and events together using `MergeSegment` if necessary.

```
> merged <- MergeSegment(segmentClonality)
```

The subclonality and somatic copy number for a sample can be visualized by `plotSubclonality`.

```
> plotSubclonality(segmentClonality)
```

Then you can output all the clonal events or the subclonal events with high proportion of aberration.

```
> merged[elementMetadata(merged)[, "clonality"]=="clonal", ]

GRanges object with 2 ranges and 6 metadata columns:
      seqnames       ranges strand |      somaCN
         <Rle>    <IRanges>  <Rle> | <character>
  [1]     chr1 [0, 5900001]      * |           1
  [2]     chr2 [0, 3900001]      * |           1
            event   clonality    germCN subclonalCN
      <character> <character> <integer>   <integer>
  [1]         LOH      clonal         2           1
  [2]        Loss      clonal         2           1
      subpercent
       <numeric>
  [1]          1
  [2]          1
  -------
  seqinfo: 2 sequences from an unspecified genome; no seqlengths
```

# 3 SomatiCA pipeline

Get familiar with how `SomatiCA` works? If so, call `SomatiCApipe` directly to run all the steps described above automatically. Multithread computing is still supported through `ncores`. We recommend to use `verbose=T` to print working messages and track the working progress.

```
> data(GCcontent)
> res <- SomatiCApipe(input, ncores = 1, collapse.k = 0, method = "mle",
+                     mcmc = 50000, burnin = 10000, p = 0.001, GC = GCcontent)
> merged <- MergeSegment(res$segment)
```

# 4 Others

## 4.1 GC content

In case the precalculated GC content will be out of date or users may want to use smaller window size, we provide a function first downloading the .fa.gz of a given chromosome from UCSC genome browser and then calculating the GC content for a given window size (10,000 bp in the following example).

```
> chr <- paste("chr", c(1:22, "X"), sep="")
> url <- "http://hgdownload.soe.ucsc.edu/goldenPath/hg19/chromosomes/"
> GC <- foreach(i=chr, .combine=rbind)%dopar%{
```

```
+       return(GCcount(i, 10000, url))
+ }
```