Paired PSCBS

A.B. Olshen, H. Bengtsson, P. Neuvial, P.T. Spellman, R.A. Olshen, V.E. Seshan

March 20, 2012

Abstract

The Paired Parent-Specific Circular Binary Segmentation (Paired PSCBS) method partitions a tumor genome into segments of constant parent-specific copy numbers (PSCNs) based on SNP DNA microrrary data from a match tumor-normal pair. The method also calls when the identified segments are in run of homozygosity (ROH), in allelic balance (AB), or loss of heterozygosity (LOH). Paired PSCBS was designed to work with data from any SNP microarray technology and generation, including Affymetrix and Illumina.

This document shows how to use the *PSCBS* package to run Paired PSCBS on a tumor-normal pair.

Keywords: copy numbers, allele specific, parent specific, genomic aberrations

This vignette is distributed as part of the PSCBS package, which is available on CRAN (`http://cran.r-project.org/`). The authors very much appreciate feedback on this document.

Contents

1 Background

We will here use a small example data set to illustrate how to setup the data in a format suitable for Paired PSCBS, how to identify segments, how to call them, and how to plot and export the segmentation results. The statistical model and the algorithm behind Paired PSCBS is explained in detail in Olshen *et al.* (2011).

2 Preparing data to be segmented

The Paired PSCBS (Olshen *et al.*, 2011) method requires tumor-normal paired parent-specific copy-number (PSCN) signals. More precisely, it requires total copy-number (TCN) estimates for the tumor relative to the matched normal (C_T), allele B fractions (BAFs) for the tumor (β_T) and BAFs for the matched normal (β_N). The genomic location of the loci in form of chromosome and physical position are also required.

2.1 Locus-level SNP copy-number signals

In this example we will use a small example data set part of the *PSCBS* package. It can be loaded as:

```
> pathname <- system.file("data-ex/PairedPSCBS,exData,chr01.Rbin",
+     package = "PSCBS")
> data <- R.utils::loadObject(pathname)
> str(data)
'data.frame': 73346 obs. of  6 variables:
 $ chromosome: int  1 1 1 1 1 1 1 1 1 1 ...
 $ x         : int  1145994 2224111 2319424 2543484 2926730 2941694 3084986 3155127..
 $ CT        : num  1.625 1.071 1.406 1.18 0.856 ...
 $ betaT     : num  0.757 0.771 0.834 0.778 0.229 ...
 $ CN        : num  2.36 2.13 2.59 1.93 1.71 ...
 $ betaN     : num  0.827 0.875 0.887 0.884 0.103 ...
```

In additional to the mandatory fields (`chromosome`, `x`, `CT`, `betaT`, and `betaN`), this data set also contains TCNs for normal (`CN`) relative to a large pool of normal samples. The latter will not be used here.

2.2 Dropping TCN outliers

There may be some outliers among the tumor TCNs. In CBS (Olshen *et al.*, 2004; Venkatraman and Olshen, 2007), the authors propose to drop those before segmentation, which can be done by:

```
> data <- dropSegmentationOutliers(data)
```

Dropping TCN outliers is optional.

3 Paired PSCBS segmentation

3.1 Skipping centromeres and other large gaps

Like the CBS method, Paired PSCBS does not take the physical locations (in units of nucleotides) of the loci in to account when segmenting the data, only their relative ordering along the genome. This means that after having ordered the loci along genome, it will treat two "neighboring" loci that are on both sides of the centromere equally as two neighboring loci that are only few hundred bases apart. This may introduce

erroneous change points that appears to be inside the centromere and biological impossible interpretation of the identified PSCN states. The same issues occur for other large gaps of the genome where there are no observed signals.

To avoid this, although not mandatory, we will locate all gaps of the genome where there are no observered loci. As a threshold we will consider a region to be a "gap" if the distance between the two closest loci is greater than 1Mb.

```
> gaps <- findLargeGaps(data, minLength = 1e+06)
> gaps
  chromosome     start       end   length
1          1 120992604 141510002 20517398
```

which shows that there is a 20.5Mb long gap between 121.0Mb and 141.5Mb on Chromosome 1. This is the centromere of Chromosome 1. Gaps cannot be specified directly. Instead they need to be give as part of a set of "known" segments, which is done as:

```
> knownSegments <- gapsToSegments(gaps)
> knownSegments
  chromosome    start      end   length
1          1     -Inf 1.21e+08      Inf
2          1 1.21e+08 1.42e+08 20517398
3          1 1.42e+08      Inf      Inf
```

Below, we will use this to tell Paired PSCBS to segment Chromosome 1 in three independent segments, where the first segments is from the beginning of the chromosomes (hence '-Inf') to 120.1Mb, the second from 120.1-141.5Mb (the above gap), and the third is from 141.5Mb to the end of the chromosome (hence '+Inf'). Just as Paired PSCBS segments chromosomes independently of each other, it also segments priorly known segments independently of each other. Specifying known segments is optional.

3.2 Identifying PSCN segments

We are now ready to segment the locus-level PSCN signals. This is done by[1]:

```
> fit <- segmentByPairedPSCBS(data, knownSegments = knownSegments,
+     seed = 48879, verbose = -10)
```

Note that this may take several minutes when applied to whole-genome data. The above call will also normalize the tumor BAFs using the TumorBoost normalization method (Bengtsson *et al.*, 2010). If this has already been done or the tumor signals have been normalized by other means, the TumorBoost step can be skipped by setting argument `tbn=FALSE`.

The result of Paired PSCBS segmentation is a set of segments identified to have the same underlying PSCN levels. In this particular case, 12 PSCN segments were found:

```
> getSegments(fit, simplify = TRUE)
   chromosome tcnId dhId     start       end tcnNbrOfLoci tcnMean tcnNbrOfSNPs
1           1     1    1    554484  33414619         9413    1.38         9413
2           1     1    2  33414619  86993745        17433    1.38        17433
3           1     2    1  86993745  87005243            2    3.19            2
4           1     3    1  87005243 119796080        10404    1.39        10404
5           1     3    2 119796080 119932126           72    1.47           72
6           1     3    3 119932126 120992603          171    1.44          171
```

[1]We fix the random seed in order for the results of this vignette to be exactly reproducible.

```
7           1     4    1 120992604 141510002           0      NA            0
8           1     5    1 141510003 185527989       13434    2.07        13444
9           1     6    1 185527989 199122065        4018    2.71         4028
10          1     7    1 199122065 206512702        2755    2.59         2756
11          1     8    1 206512702 206521352          14    3.87           14
12          1     9    1 206521352 247165315       15581    2.64        15607
   tcnNbrOfHets dhNbrOfLoci dhMean c1Mean c2Mean
1          2765        2766  0.642  0.247   1.13
2          4544        4544  0.684  0.218   1.16
3             0           0     NA     NA     NA
4          2777        2778  0.686  0.218   1.17
5             8           8  0.101  0.661   0.81
6            52          52  0.676  0.233   1.21
7             0          NA     NA     NA     NA
8          3771        3771  0.124  0.904   1.16
9          1276        1276  0.338  0.896   1.81
10          784         784  0.290  0.918   1.67
11            9           9  0.365  1.229   2.64
12         4499        4499  0.302  0.920   1.72
```

Note how Segment #7 has no mean-level estimates. It is because it corresponds to the centromere (the gap) that was identified above. Paired PSCBS did indeed try to segment it, but since there are no data points, all estimates are missing values. Similarly, for Segment #3 the DH and minor and major CNs mean estimates are all missing values. This is because, Paired PSCBS identified that segment by first segmenting the TCN signals by themselves, and thereafter it tried segmenting the DH signals within that segment. Since there are no heterozygous SNPs in the segment, there exist no DH signals, and hence no DH mean estimate.

3.3 Displaying genomic PSCN profiles

To plot the PSCN segmentation results, do:

```
plotTracks(fit)
```

which by default displays three panels containing TCN, decrease of heterozygosity (DH), and minor and major CNs as in Figure 1. To only plot one panel with TCN and minor and major CNs and zoom in on a partical region, do:

```
plotTracks(fit, tracks="tcn,c1,c2", xlim=c(120,244)*1e6)
```

4 Calling segments

The calling algorithms for allelic balance (AB) and loss of heterozygosity (LOH) are based on quantile estimates of the different mean levels. These estimates are obtained from using non-parametric bootstrap techniques. For more details, see Olshen *et al.* (2011). After the Paired PSCBS method was published, we have also added a method for calling run of homozygosity (ROH).

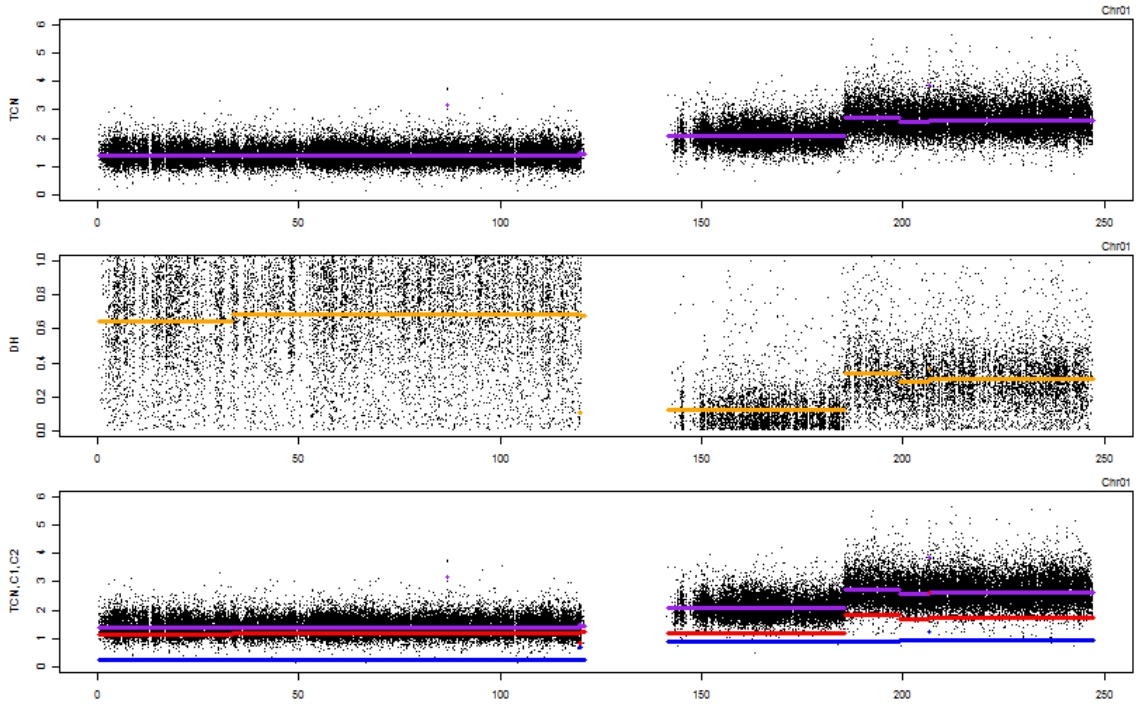

Figure 1: PSCN segments identified by Paired PSCBS. **Top**: The TCN signals (black dots) with the TCN mean levels (purple). **Middle**: The DH signals (black dots) with the DH mean levels (orange). **Bottom**: The TCN signals (black dots) with the minor CN (C_1; blue), the major CN (C_2; red) and the TCN ($C = C_1 + C_2$; purple) mean levels.

4.1 Calling segments with run of homozygosity (ROH)

A region with run of homozygosity (ROH) is a region where all SNPs are homozygous (in the normal). Since such a region has no heterozyous SNPs, its decrease of heterozygosity (DH) is undefined. Likewise, the minor and major copy numbers are unknown. However, if there are genotyping errors within an ROH region, we will obtain a non-missing DH mean level and hence also finite minor and major CNs. In order to adjust for these faulty estimates, we test if the identified segments are in ROH or not by:

```
> fit <- callROH(fit, verbose = -10)
```

This will also set the corresponding DH and minor and major CN mean levels to NA. The total CN mean levels are not affected by the ROH call.

4.2 Calling segments in allelic balance (AB)

```
> fit <- callAB(fit, verbose = -10)
```

Because this utilizes bootstrapping techniques, calling AB may take some time if there is a large number of segments.

4.3 Calling segments with loss of heterozygosity (LOH)

```
> fit <- callLOH(fit, verbose = -10)
```

Note that in order to call LOH, one has to call allelic balance first. Since the bootstrapping was already done in the AB caller, it is not repeated here, which is why calling LOH is faster than calling AB.

4.4 Results from calling ROH, AB and LOH

All calls are appended to the segmentation results as logical columns:

```
> getSegments(fit, simplify = TRUE)
   chromosome tcnId dhId     start       end tcnNbrOfLoci tcnMean tcnNbrOfSNPs
1           1     1    1    554484  33414619         9413    1.38         9413
2           1     1    2  33414619  86993745        17433    1.38        17433
3           1     2    1  86993745  87005243            2    3.19            2
4           1     3    1  87005243 119796080        10404    1.39        10404
5           1     3    2 119796080 119932126           72    1.47           72
6           1     3    3 119932126 120992603          171    1.44          171
7           1     4    1 120992604 141510002            0      NA            0
8           1     5    1 141510003 185527989        13434    2.07        13444
9           1     6    1 185527989 199122065         4018    2.71         4028
10          1     7    1 199122065 206512702         2755    2.59         2756
11          1     8    1 206512702 206521352           14    3.87           14
12          1     9    1 206521352 247165315        15581    2.64        15607
   tcnNbrOfHets dhNbrOfLoci dhMean c1Mean c2Mean rohCall abCall lohCall
1          2765        2766  0.642  0.247   1.13   FALSE  FALSE    TRUE
2          4544        4544  0.684  0.218   1.16   FALSE  FALSE    TRUE
3             0           0     NA     NA     NA    TRUE     NA      NA
4          2777        2778  0.686  0.218   1.17   FALSE  FALSE    TRUE
5             8           8     NA     NA     NA    TRUE     NA      NA
6            52          52  0.676  0.233   1.21   FALSE  FALSE    TRUE
7             0          NA     NA     NA     NA      NA     NA      NA
8          3771        3771  0.124  0.904   1.16   FALSE   TRUE   FALSE
9          1276        1276  0.338  0.896   1.81   FALSE  FALSE   FALSE
10          784         784  0.290  0.918   1.67   FALSE  FALSE   FALSE
11            9           9  0.365  1.229   2.64   FALSE  FALSE   FALSE
12         4499        4499  0.302  0.920   1.72   FALSE  FALSE   FALSE
```

4.5 Writing segments to a tab-delimited text file

To write the PSCN segmentation results to file, do:

```
writeSegments(fit, name="MySample", simplify=TRUE)
```

5 Ongoing/Future work

In this section we illustrate some of the ongoing and future work of the PSCBS package. Please be aware that these methods are very much under construction, possibly incomplete and in worst case even incorrect.

5.1 Pruning segmentation profile

By using hierarchical cluster of the segment means it is possible to prune the PSCN profile such that change points with very small absolute changes are dropped. If change points are dropped this way, this results in a smaller number of segments, which are hence longer.

```
> fitP <- pruneByHClust(fit, h = 0.25, verbose = -10)
```

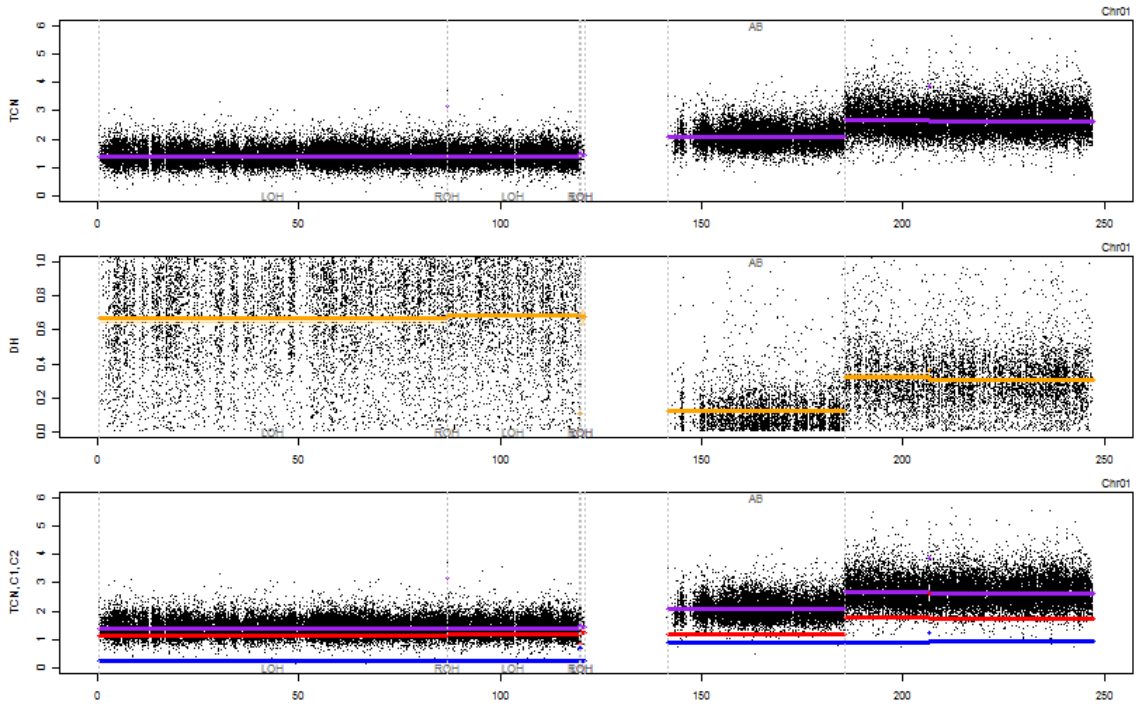

Figure 2: Pruned PSCN segments plotted as in Figure 1.

References

Bengtsson, H., Neuvial, P., and Speed, T. P. (2010). TumorBoost: Normalization of allele-specific tumor copy numbers from a single pair of tumor-normal genotyping microarrays. *BMC Bioinformatics*, **11**(1), 245.

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**(4), 557–572.

Olshen, A. B., Bengtsson, H., Neuvial, P., Spellman, P., Olshen, R. A., and Seshan, V. E. (2011). Parent-specific copy number in paired tumor-normal studies using circular binary segmentation. *Bioinformatics*, **27**(15), 2038–2046.

Venkatraman, E. S. and Olshen, A. B. (2007). A faster circular binary segmentation algorithm for the analysis of array CGH data. *Bioinformatics*, **23**(6), 657–663.

Appendix

Session information

- R version 2.14.2 Patched (2012-02-29 r58590), `x86_64-pc-mingw32`
- Locale: `LC_COLLATE=C`, `LC_CTYPE=English_United States.1252`, `LC_MONETARY=English_United States.1252`, `LC_NUMERIC=C`, `LC_TIME=English_United States.1252`
- Base packages: base, datasets, grDevices, graphics, methods, splines, stats, utils
- Other packages: DNAcopy 1.29.1, Hmisc 3.9-2, PSCBS 0.23.0, R.cache 0.6.1, R.methodsS3 1.2.3, R.oo 1.9.3, R.rsp 0.7.5, R.utils 1.12.0, aroma.light 1.23.1, digest 0.5.1, matrixStats 0.4.5, survival 2.36-12
- Loaded via a namespace (and not attached): cluster 1.14.2, grid 2.14.2, lattice 0.20-6

This report was automatically generated using `rsp()` of the R.rsp package. Total processing time after RSP-to-R translation was 42.99 secs.