mBPCR: A package for DNA copy number profile estimation

P. M. V. Rancoita[1,2,3] and M. Hutter[4]

[1]Istituto Dalle Molle di Studi sull'Intelligenza Artificiale (IDSIA), Manno-Lugano, Switzerland
[2]Laboratory of Experimental Oncology, Oncology Institute of Southern Switzerland (IOSI), Bellinzona, Switzerland
[3]Dipartimento di Matematica, Università degli Studi di Milano, Milano, Italy
[4]RSISE @ ANU and SML @ NICTA, Canberra, ACT, 0200, Australia

paola@idsia.ch

Contents

1 Introduction

The algorithm mBPCR is a tool for estimating the profile of the $\log_2$ratio of copy number data. The procedure is a Bayesian piecewise constant regres-

sion and can be applied, generally, to estimate any piecewise constant function (like the $\log_2$ratio of the copy number data). The method is described in [3] and represents a significant improvement of the original algorithm BPCR, presented in [1] and [2].

This document shows several examples of how to use the package. The data used are principally: the Affymetrix GeneChip Mapping 10K Array data of cell line REC-1 [4] and the Affymetrix GeneChip Mapping 250K Array data of chromosome 11 of cell line JEKO-1 (unpublished).

2   Example 1: profile estimation

In this example we estimate the copy number profile of sample `rec10k`.

```
> library(mBPCR)
```

First, we import the 10K Array data of cell line REC-1.

```
> data(rec10k)
```

During the computation, the algorithm needs to create a vector of size (`maxProbeNumber`+1)(`maxProbeNumber`+2)/2, where `maxProbeNumber` is the maximum number of probes of a chromosome (or arm of a chromosome, for denser Array). Hence, before the estimation, we must verify if we have enough RAM to allocate such a vector. In case of the 10K Array data, we know that all chromosomes have less than 1000 probes, thus we verify if we can set `maxProbeNumber=1000`, with the following commands,

```
> maxProbeNumber <- 1000
```

```
> A <- array(1, dim = (maxProbeNumber + 1) * (maxProbeNumber +
+     2)/2)
```

If last command does not give any error regarding the memory allocation, then we can set `maxProbeNumber=1000` and remove `A` to save space.

```
> remove(A)
```

To estimate the profile of one or more chromosomes, we need to set the parameter `chrToBeAnalyzed` with the vector of the names of the chromosomes that we want to analyze (the names allowed are: `X`, `Y` and any integer from `1` to `22`). In the following example, we estimate the profile of chromosomes 3 and 5 of sample REC-1. Instead, to estimate the profile of the whole genome, we need to set `chrToBeAnalyzed = c(1:22,"X")`.

```
> results <- estProfileWithMBPCR(rec10k$SNPname, rec10k$Chromosome,
+      rec10k$PhysicalPosition, rec10k$log2ratio, chrToBeAnalyzed = c(3,
+          5), maxProbeNumber = 1000)
```

We can nicely write the results on tab delimited files in the working directory, by using the function `writeEstProfile` (Tables 1 and 2 show the first lines of the two tables created by the command below). Setting `sampleName="rec10k"`, the name of the files will contain the name of the sample `rec10k`. If `path=NULL`, the tables will not be written on files, but only returned by the function.

```
> writeEstProfile(path = "", sampleName = "rec10k", rec10k$SNPname,
+      rec10k$Chromosome, rec10k$PhysicalPosition, rec10k$log2ratio,
+      chrToBeWritten = c(3, 5), results$estPC, results$estBoundaries)
```

SNPname	chromosome	position	rawLog2ratio	mBPCRestimate
SNP_A-1511742	3	540961	0.367371066	-0.33118451
SNP_A-1515436	3	653347	-0.051399153	-0.33118451
SNP_A-1515061	3	1100383	-0.577766999	-0.33118451
SNP_A-1510244	3	1167829	-1.377069649	-0.33118451
SNP_A-1517422	3	1167988	-1.058893689	-0.33118451
SNP_A-1515258	3	1478475	-0.184424571	-0.33118451

Table 1: Example of table containing the profile estimated with mBPCR.

| SNPname(start)	SNPname(end)	chromosome	position(start)	position(end)	nProbes	mBPCRestimate
SNP_A-1511742	SNP_A-1517209	3	540961	3814711	23	-0.33118451
SNP_A-1512404	SNP_A-1508199	3	3887946	6473283	28	0.26189600
SNP_A-1519522	SNP_A-1509746	3	6482290	141372655	528	-0.05735106
SNP_A-1516670	SNP_A-1518807	3	141372855	141373169	4	0.04208212
SNP_A-1511225	SNP_A-1516851	3	141494264	141925969	3	0.11634699
SNP_A-1517017	SNP_A-1517017	3	142426479	142426479	1	-0.07473727

Table 2: Example of table containing a summary of the breakpoints estimated with mBPCR.

We can also estimate the profile with a Bayesian regression curve [3]. For example, with the following command we estimate the profile of chromosome 3 using both mBPCR and the Bayesian Regression Curve with $\hat{K}_2$.

3

```
> results <- estProfileWithMBPCR(rec10k$SNPname, rec10k$Chromosome,
+     rec10k$PhysicalPosition, rec10k$log2ratio, chrToBeAnalyzed = 3,
+     regr = "BRC", maxProbeNumber = 1000)
```

After the estimation, we can plot the profiles using the function `plotEstPro-file`. For example, the following command plots the profile of chromosome 3 estimated with both methods.

```
> plotEstProfile(sampleName = "rec10k", rec10k$Chromosome, rec10k$PhysicalPosition,
+     rec10k$log2ratio, chrToBePlotted = 3, results$estPC, maxProbeNumber = 2000,
+     regrCurve = results$regrCurve, regr = "BRC")
```

rec10k
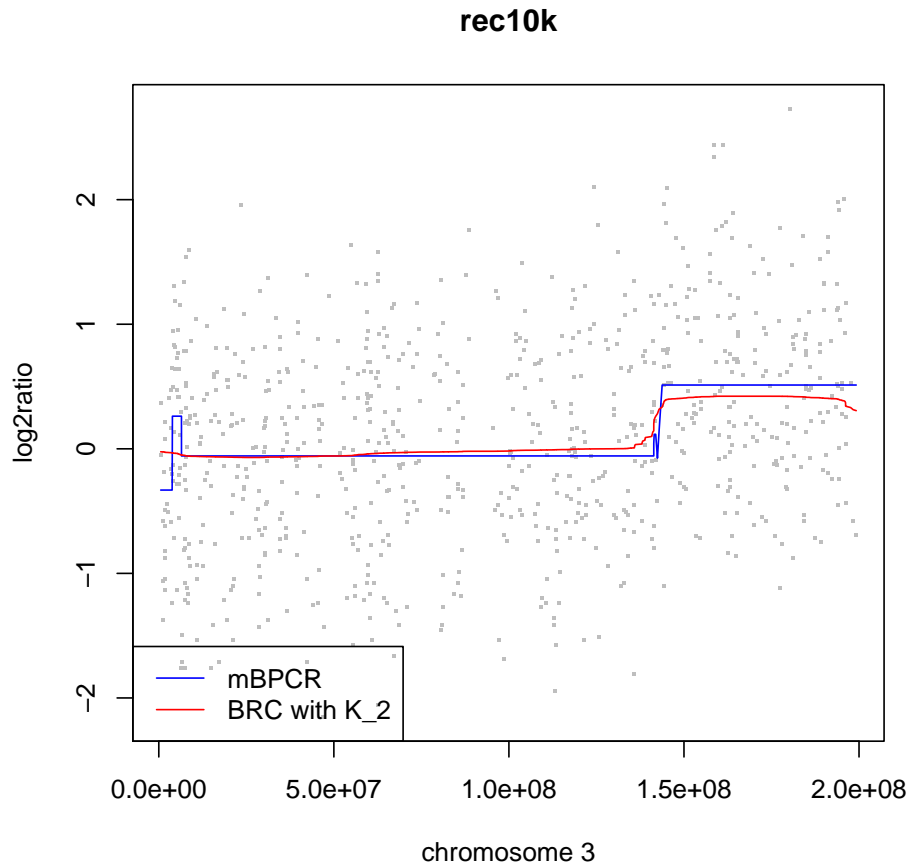
chromosome 3

As second example, we estimate the profile of chromosome 11 of sample JEKO-1. Notice that we need to set `maxProbeNumber <- 9000` (because

4

both arms of chromosome 11 contain less than 9000 probes) and, if this is possible on your machine, the computation can be long. Moreover, for the estimation, we use the estimates of the parameters computed on the whole genome to achieve a better profile (for the estimation of the global parameters, see the use of function `estGlobParam` in Section 3).

```
> data(jekoChr11Array250Knsp)

> maxProbeNumber <- 9000

> A <- array(1, dim = (maxProbeNumber + 1) * (maxProbeNumber +
+     2)/2)

> remove(A)

> results <- estProfileWithMBPCR(jekoChr11Array250Knsp$SNPname,
+     jekoChr11Array250Knsp$Chromosome, jekoChr11Array250Knsp$PhysicalPosition,
+     jekoChr11Array250Knsp$log2ratio, chrToBeAnalyzed = 11, maxProbeNumber = 9000,
+     rhoSquare = 0.0479, nu = -3.012772e-10, sigmaSquare = 0.0699)

> plotEstProfile(sampleName = "jeko250Knsp", jekoChr11Array250Knsp$Chromosome,
+     jekoChr11Array250Knsp$PhysicalPosition, jekoChr11Array250Knsp$log2ratio,
+     chrToBePlotted = 11, results$estPC, maxProbeNumber = 9000)
```

3 Example 2: use of function `estGlobParam`

In general, even if we are not interested in the analysis of the whole genome, the global parameters should be estimated on the entire sample, using the function `estGlobParam`. Here, we estimate the global parameters of sample REC-1 (in the following, the variance of the segment ρ^2 is estimated with $\hat{\rho}_1^2$).

```
> data(rec10k)

> estGlobParam(rec10k$log2ratio)

$nu
[1] -0.02403854

$rhoSquare
[1] 0.0889637

$sigmaSquare
[1] 0.5971426
```

4   Example 3: use of function `computeMBPCR`

If we are interested in estimating only a part of a chromosome or a simulated sample, we should not use the function `estProfileWithMBPCR`, but use the function `computeMBPCR` which estimates the profile directly. In the following example, we estimates the profile of a part of chromosome 11 of sample JEKO-1.

```
> data(jekoChr11Array250Knsp)
```

We select a part of chromosome 11.

```
> y <- jekoChr11Array250Knsp$log2ratio[10600:11200]
```

```
> p <- jekoChr11Array250Knsp$PhysicalPosition[10600:11200]
```

We estimate the profile with mBPCR and BRC with $\hat{K}_2$, using the global parameters estimated on the whole genome.

```
> results <- computeMBPCR(y, nu = -3.012772e-10, rhoSquare = 0.0479,
+      sigmaSquare = 0.0699, regr = "BRC")
```

Finally, we plot the results.

```
> plot(p, y)
> points(p, results$estPC, type = "l", col = "red")
> points(p, results$regrCurve, type = "l", col = "green")
> legend(x = "bottomleft", legend = c("mBPCR", "BRC with K_2"),
+      lty = c(1, 1), col = c(4, 2))
```
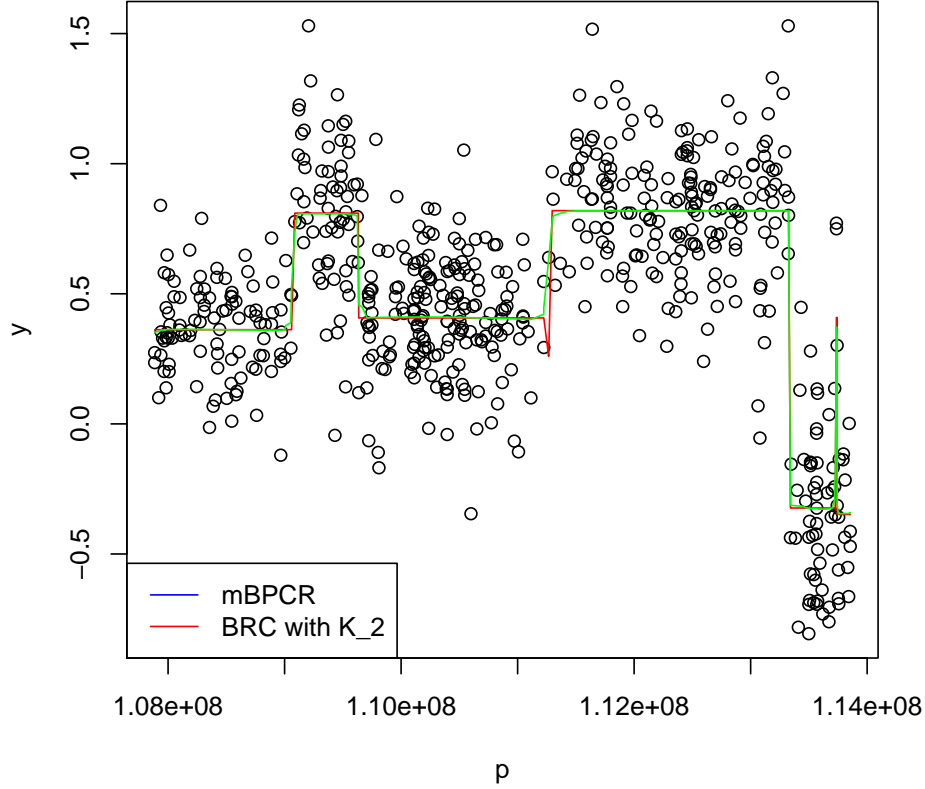
5 Example 4: `importCNData`, an easy function to import data

There is also the possibility to easily import external data, by using **im-portCNData**. The data should be in a tab delimited file and the data table should have at least four columns representing, respectively, the probe names, the chromosome to which each probe belongs, the physical positions of the probes inside the chromosome and the copy number data (an example of table can be found in Table 3). The allowed names of the chromosomes are: X, Y and any integer from 1 to 22). In the following example, we import data of sample REC-1.

As first step, we need to set a variable with the path of the file containing

7

SNPname	Chromosome	PhysicalPosition	log2ratio
SNP_A-1509443	1	2882121	-0.184424571
SNP_A-1518557	1	3985402	0.097610797
SNP_A-1517286	1	4804829	0.443606651
SNP_A-1516024	1	4982250	-1.089267338
SNP_A-1514538	1	5468765	-0.862496476
SNP_A-1516403	1	5596686	1.097610797
⋮	⋮	⋮	⋮

Table 3: Example of data table.

the data. To import our data, we set `path` as the path of REC-1 data in the folder of package `mBPCR`,

```
> path <- system.file("extdata", "rec10k.txt", package = "mBPCR")
```

Then, we use function `importCNData`. The parameter `NRowSkip` denotes how many rows there are before the table (notice that the name of the columns must be skipped). If the copy number data are not in $\log_2$ratio scale, the parameter `ifLogRatio` should be put as zero.

```
> rec10k <- importCNData(path, NRowSkip = 1)
```

Now, the SNP name are in the variable `SNPname`, the chromosomes of the probes are in `chr`, the physical positions in `position` and the raw $\log_2$ratio data in `logratio`. Here, we plot the raw data of chromosome 3.

```
> plot(rec10k$position[rec10k$chr == 3], rec10k$logratio[rec10k$chr ==
+     3], xlab = "Chromosome 3", ylab = "log2ratio")
```
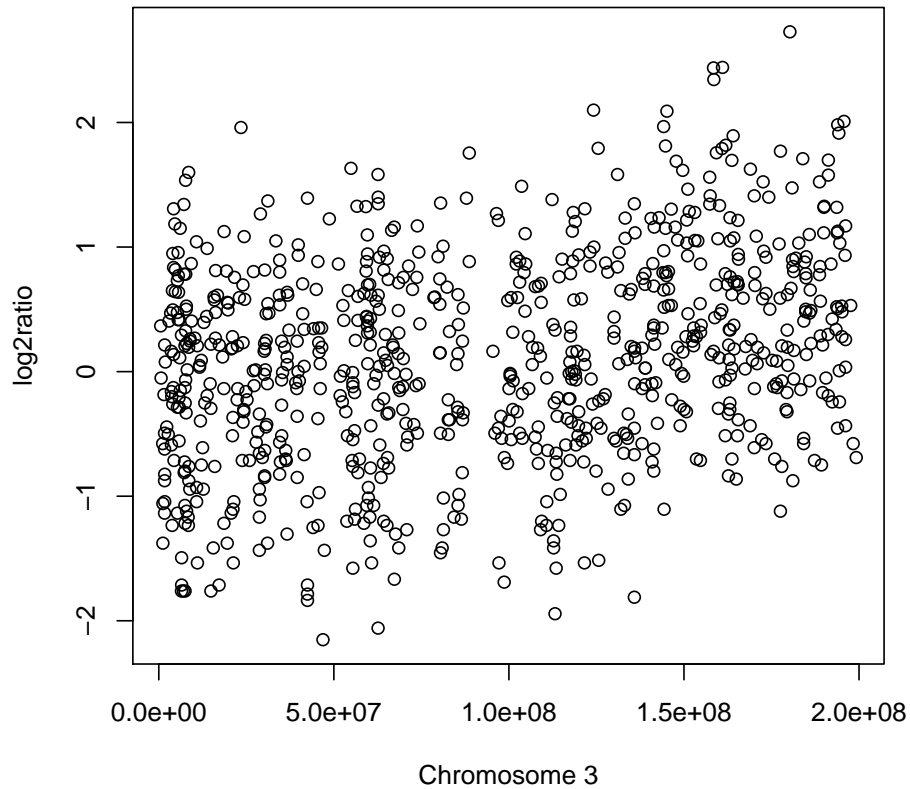
6 Example 5: estimation of samples in an oligoSnpSet object

In this example, we estimate the profile of chromosome 8 of two samples that are contained in an `oligoSnpSet` object. We load the data that are contained in the package `SNPchip`.

```
> library(SNPchip)
```

```
> data(sample.snpset)
```

The object `sample.snpset` contains the data of five HapMap samples and we want to analyze the third and the fourth samples. Since the samples

should not have many copy number changes, we use `rhoSquare` equal to the one of the sample REC-1. Moreover, the data are not in $\log_2$ratio scale, thus we set `ifLogRatio=0`.

```
> library(mBPCR)

> r <- estProfileWithMBPCRforOligoSnpSet(sample.snpset, sampleToBeAnalyzed = 3:4,
+     chrToBeAnalyzed = 8, maxProbeNumber = 1000, ifLogRatio = 0,
+     rhoSquare = 0.0889637)
```

After the estimation, we can plot the profiles using a function of the package `SNPchip`.

```
> cc <- r$estPC

> cc1 <- cc[chromosome(cc) == 8, 3:4]

> plotSnp(cc1)
```

7   Suggestions

For an optimal use of mBPCR, especially in case of samples coming from patients, we suggest to take care to the following issues:

- even if the goal is to estimate the profile of only a part of the genome, the global parameters should be estimated on the whole genome;

- if the goal is to estimate the profile of one or more patients, it is better to estimate the variance of the segment levels (ρ^2) on a cell line, or on a sample with many aberrations, and use this value in the profile estimation of all patients. In fact, we need many aberrations to estimate well ρ^2.

References

[1] M. Hutter. Exact Bayesian regression of piecewise constant functions. *Bayesian Analysis*, 2(4): 635–664, 2007.

[2] M. Hutter. Bayesian Regression of Piecewise Constant Functions. In J.M. Bernardo, M.J. Bayarri, J.O. Berger, A.P. David, D. Heckerman, A.F.M. Smith, and M. West, editors, *Bayesian Statistics: Proceedings of the Eighth Valencia International Meeting*. Universitat de València and International Society for Bayesian Analysis, 2007.

[3] P.M.V. Rancoita, M. Hutter, F. Bertoni, and I. Kwee. Bayesian DNA copy number analysis. *BMC Bioinformatics*, 10(10), 2009.

[4] A. Rinaldi, I. Kwee, M. Taborelli, C. Largo, S. Uccella, V. Martin, G. Poretti, G. Gaidano, G. Calabrese, G. Martinelli, *et al.*. Genomic and expression profiling identifies the B-cell associated tyrosine kinase Syk as a possible therapeutic target in mantle cell lymphoma. *British Journal of Haematology*, 132: 303–316, 2006.