DE LA RECHERCHE À L'INDUSTRIE

www.cea.fr

C3CO - Cancer Cell Clonality from COpy number data

Un package pour l'inference de la clonalité des cellules cancéreuses à partir du nombre de copies d'ADN

Morgane Pierre-Jean

CEA/DRF/IBFJ/CNRGH

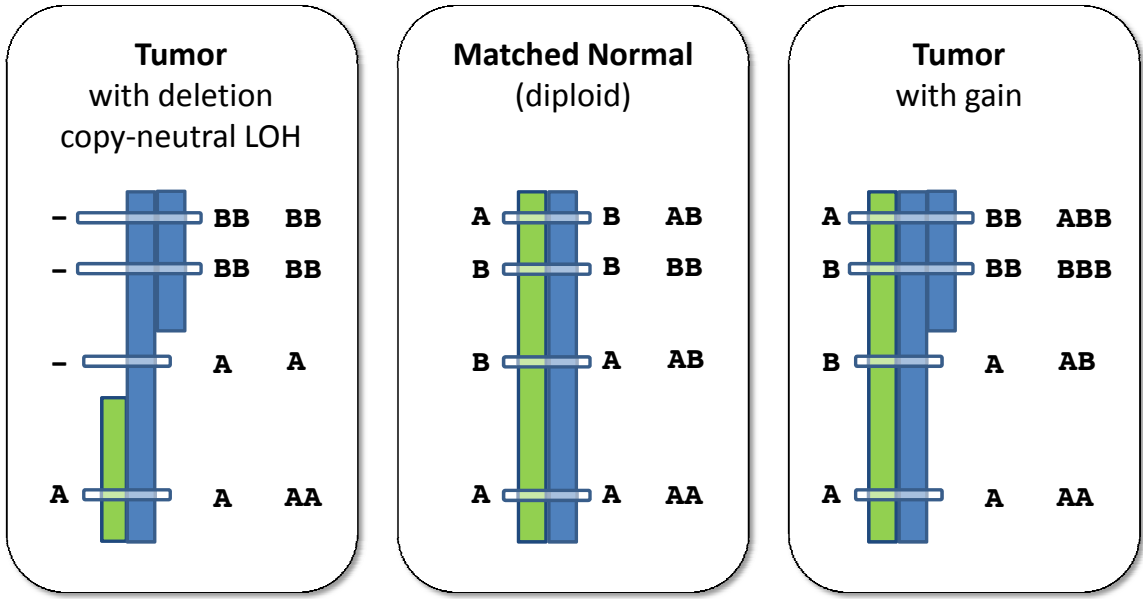
Introduction

General context

Goal of DNA copy number studies in cancerology is to identify altered regions of the genome for

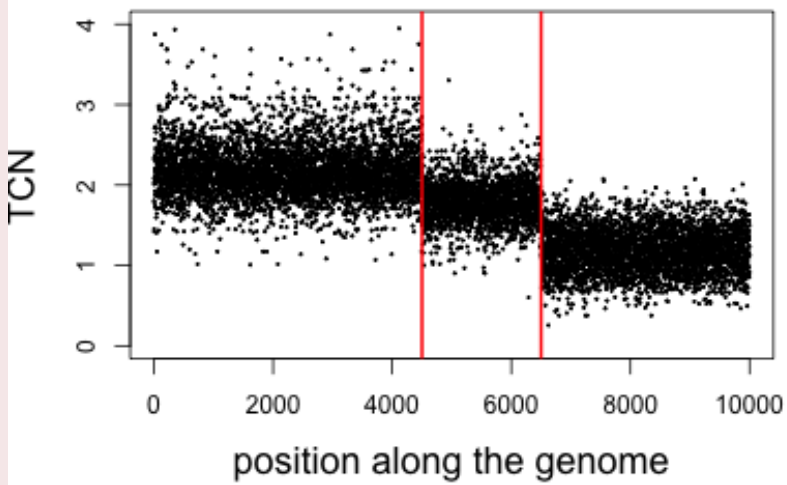
- Better understanding of tumor development
- Personalized therapies

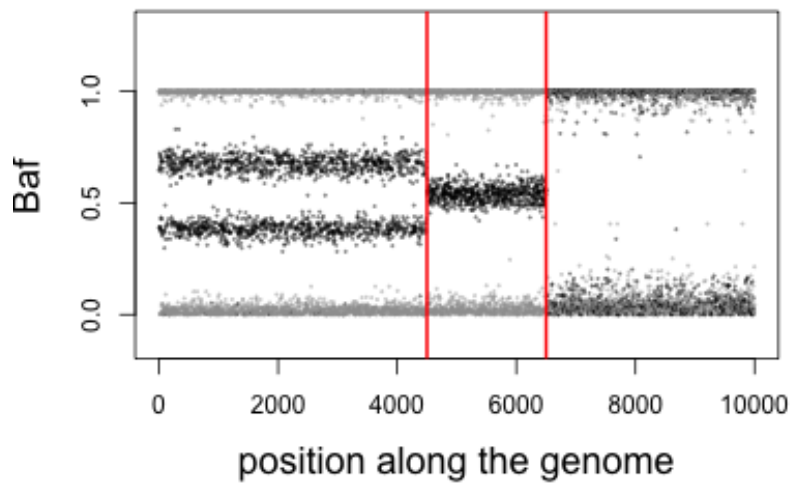
DNA copy number



- Identify breakpoints
  - jointSeg available since January 2013 on Github.
- Tumoral heterogeneity
  - c3co available since January 2017 on Github.

Joint segmentation of TCN and BAF





Notion of heterogeneity in cancers

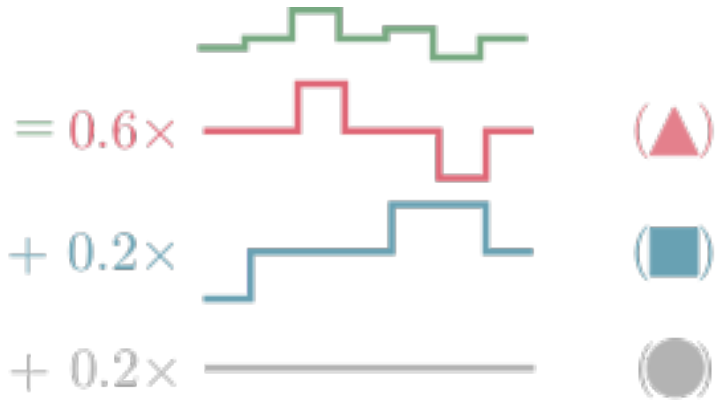
Heterogeneity

- Differences between tumors of the same disease in different patients (inter-tumor heterogeneity)
- Differences between cancer cells within a single tumor of one patient (intra-tumor heterogeneity).

Illustration

(a) Tumor sample

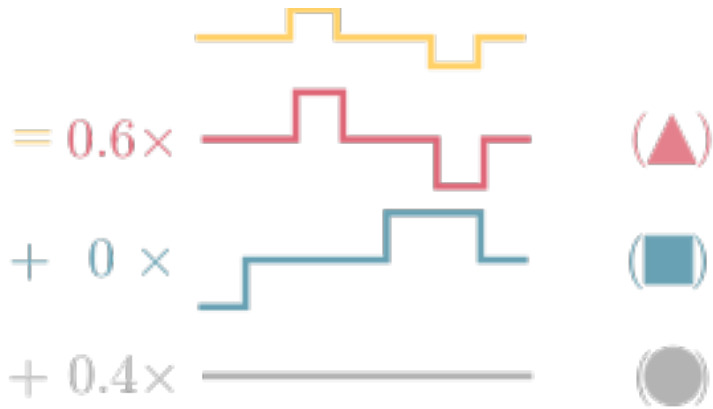
(b) Copy-number profile



Illustration

(a) Tumor sample

(b) Copy-number profile



Model

Modelization

- $y_{1\bullet} \in \mathbb{R}^J$ and $y_{2\bullet} \in \mathbb{R}^J$ DNA copy number observed.

$$y_{1\bullet} = w_{11} z_{1\bullet} + w_{12} z_{2\bullet} + w_{13} z_{3\bullet}$$

$$y_{2\bullet} = w_{21} z_{1\bullet} + w_{22} z_{2\bullet} + w_{23} z_{3\bullet}$$

Goals

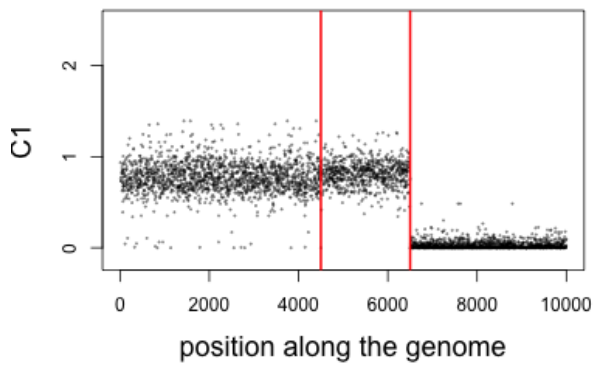
- Tumor composition w
- Alterations in subclones z

Integrating BAF through Parental copy numbers

What is parental copy number ?
$d_j = 2|b_j - 1/2|$ for AB SNPs

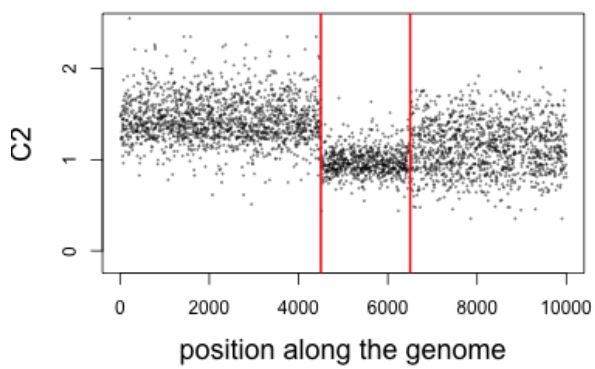
Minor copy number

$$c_j^1 = c_j(1 - d_j)/2$$



Major copy number

$$c_j^2 = c_j(1 + d_j)/2$$



First step joint segmentation

First step : dimension reduction

- Double joint segmentation
  - On TCN and DH
  - On all samples simultaneously
- Use jointSeg package

Optimization problem

$$\min_{W,Z^1,Z^2} \|Y^1 - WZ^1\|_F^2 + \lambda_1 \sum_{k=1}^{p} \sum_{s=1}^{S-1} |z^1_{k,s+1} - z^1_{k,s}| \qquad (1)$$

$$\|Y^2 - WZ^2\|_F^2 + \lambda_2 \sum_{k=1}^{p} \sum_{s=1}^{S-1} |z^2_{k,s+1} - z^2_{k,s}|$$

s.t $w_{i\bullet} \in \Delta_p$ where
$\Delta_p = \{w \in \mathbb{R}^p \quad s.t. \quad w \geq 0 \quad and \quad \sum_{k=1}^{p} w_k = 1\}$

Final algorithm

Algorithm 1 Find weights and latent profiles

Parameters : λ_1, λ_2 and p

INIT : Matrices $Y \in \mathbb{R}^{n \times S}$, $Y^1 \in \mathbb{R}^{n \times S}$ and $Y^2 \in \mathbb{R}^{n \times S}$ and matrix Z_0^1 and $Z_0^2 \in \mathbb{R}^{p \times S}$, and

for $l = 0, 1, 2, \ldots$ **do**

    Minimize in W with Z_l^1 and Z_l^2 fixed

    Minimize in Z^1 with W_l fixed

    Minimize in Z^2 with W_l fixed

    W_l, Z_l^1 and Z_l^2 are updated

    Check if $\|W_{l-1} - W_l\|_2^2 < \epsilon$ or max_{it} is reached

end for

Final algorithm

Algorithm 2 Find weights and latent profiles

Parameters : λ_1, λ_2 and p

INIT : Matrices $Y \in \mathbb{R}^{n \times S}$, $Y^1 \in \mathbb{R}^{n \times S}$ and $Y^2 \in \mathbb{R}^{n \times S}$ and matrix Z_0^1 and $Z_0^2 \in \mathbb{R}^{p \times S}$, and

for $l = 0, 1, 2, \ldots$ **do**

    Minimize in W with Z_l^1 and Z_l^2 fixed

    Minimize in Z^1 with W_l fixed

    Minimize in Z^2 with W_l fixed

    W_l, Z_l^1 and Z_l^2 are updated

    Check if $\|W_{l-1} - W_l\|_2^2 < \epsilon$ or max_{it} is reached

end for

Final algorithm

Solving 4 : Inference of W

* Weights of each patient can be treated independently
* Solve n least-squares problems with equality constraint plus inequality constraints for the non-negativity of the coefficient
* linear inverse problem that can be solved in `R` with the package **limSolve**.

Solving 5 and 6 : Inference of latent profiles

* for a fixed W cut into two independent LASSO problems in (Z_1, Z_2)
* Use matrix algebra and properties of the vectorization operator
* Obtain LASSO problem that can be solved in `R` with the package **glmnet**.

Code

Parameters of simulated data

```
library(c3co)
set.seed(19)

len <- 500*10
nbClones <- 3
bkps <- list(c(100,250)*10,
             c(150,400)*10,
             c(150,400)*10)
regions <-list(c("(0,1)", "(0,2)","(1,2)"),
               c("(1,1)", "(0,1)","(1,1)"),
               c("(0,2)", "(0,1)","(1,1)"))
```

Simulation of Subclones

```
dataAnnotTP <- acnr::loadCnRegionData(
                  dataSet="GSE13372_HCC1143",
                  tumorFraction=1)
dataAnnotN <- acnr::loadCnRegionData(
                  dataSet="GSE13372_HCC1143",
                  tumorFraction=0)
datSubClone <- buildSubclones(len, dataAnnotTP,
                                  dataAnnotN,
                                  nbClones, bkps, regions)
```

Simulation of weights

```
W = rSparseWeightMatrix(nb.samp=10,
                        nb.arch=3,
                        sparse.coeff = 0.7)
datList <- mixSubclones(subClones=datSubClone, W)
str(datList[[1]])
```

```
## 'data.frame':    5000 obs. of  7 variables:
##  $ c1       : num  NA NA NA NA NA ...
##  $ c2       : num  NA NA NA NA NA ...
##  $ tcn      : num  2.05 2.06 1.86 2.04 2.04 ...
##  $ dh       : num  NA NA NA NA NA ...
##  $ genotype: num  0 1 1 0 1 0 0 1 0.5 0.5 ...
##  $ chr      : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ pos      : int  1 2 3 4 5 6 7 8 9 10 ...
```

Run algorithm

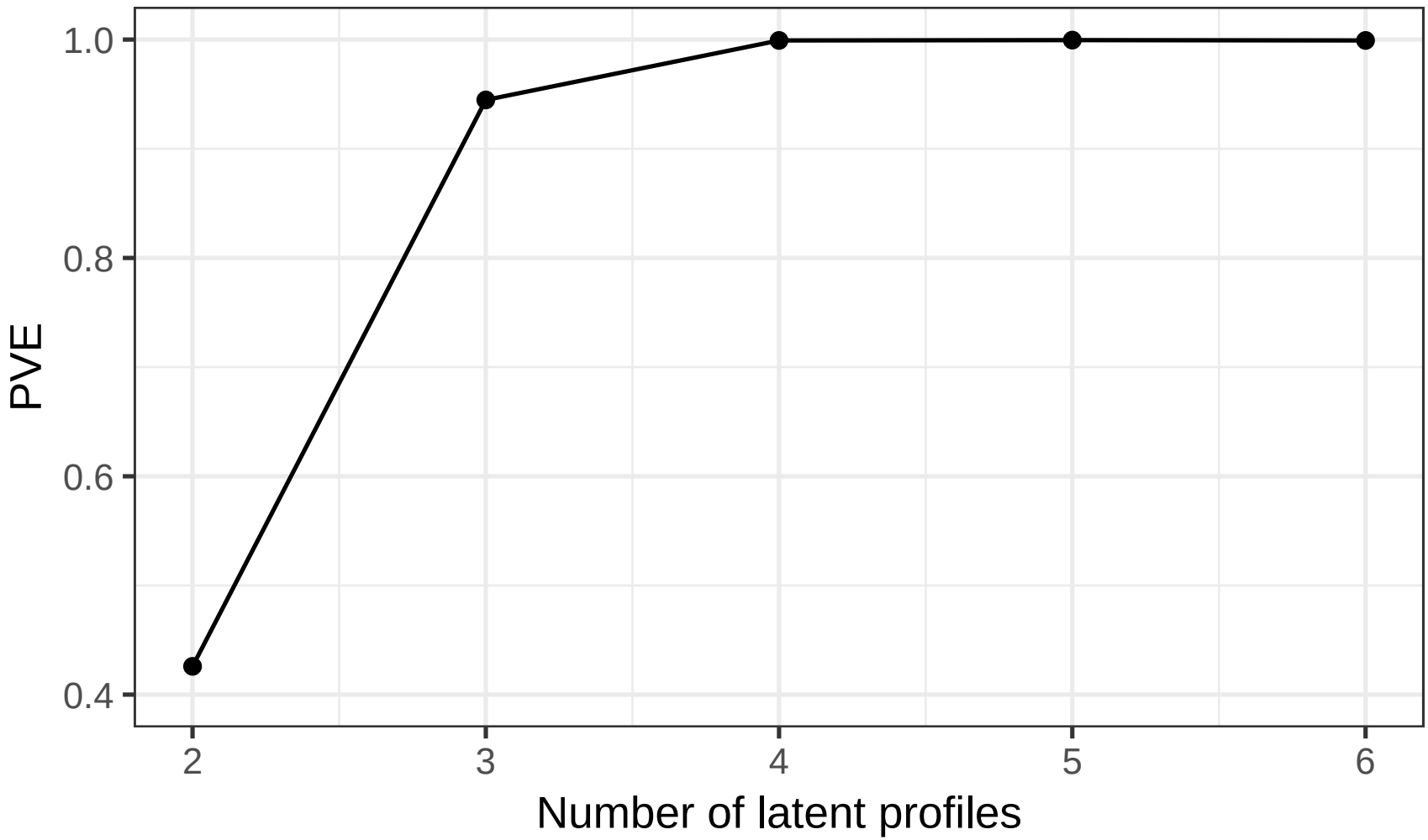
```
l1 <- seq(from = 10^-7, to = 10^-6, length = 2 )
nb.arch <- 2:6
parameters.grid <- list(lambda=l1, nb.arch=nb.arch)

res <- c3co(datList, parameters.grid, warn=FALSE)
```

```
## Note: method with signature 'dsparseMatrix#dsparseMatrix
##  target signature 'dgTMatrix#dgCMatrix'.
##  "TsparseMatrix#sparseMatrix" would also be valid
```

```
resC <- c3co(datList, parameters.grid, stat="TCN",warn=FALS
```

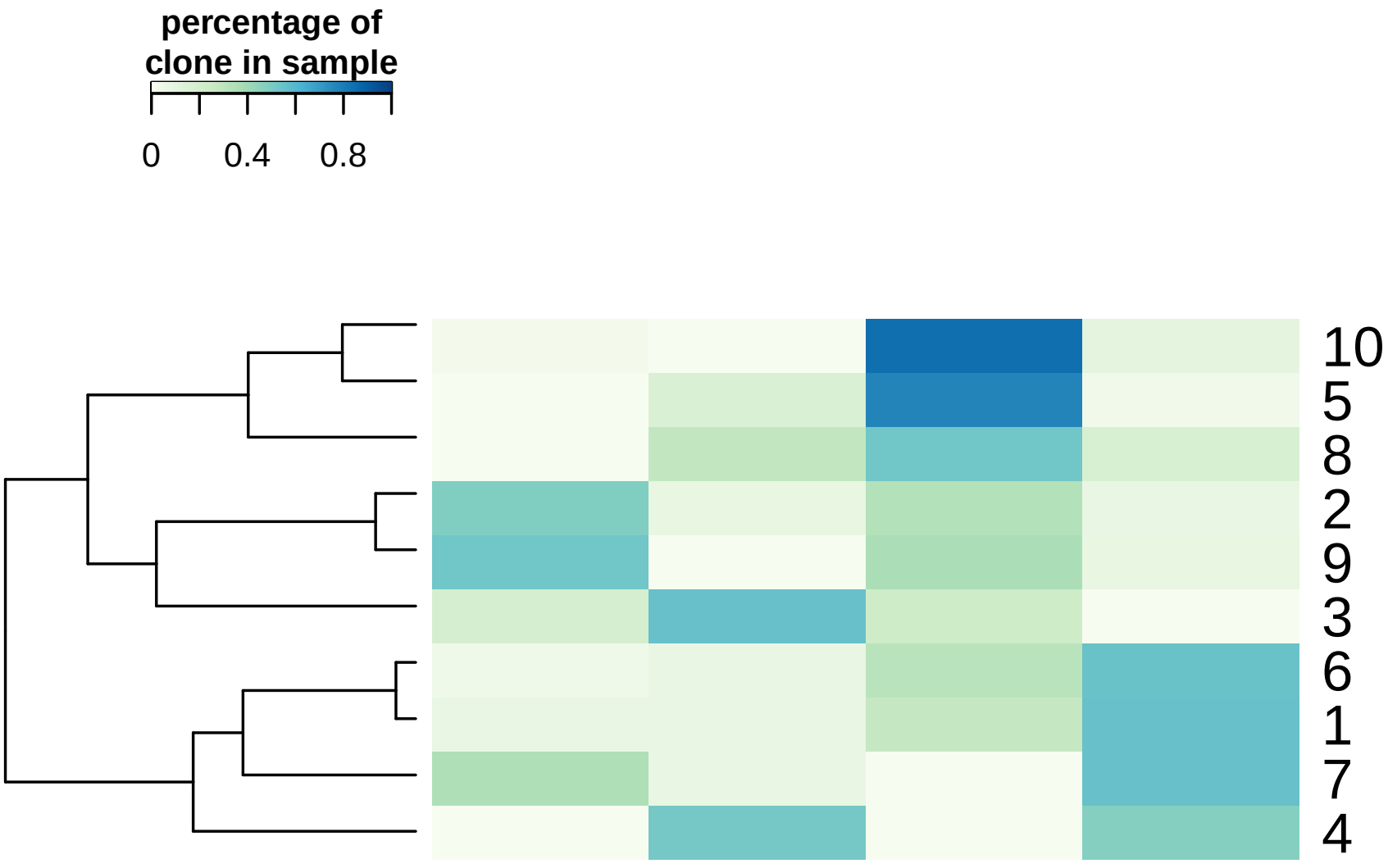
PVE plot

```
pvePlot(res, ylim=c(0.4,1))
```



Matrix W plot

```
Wplot(res, idxBest = 3, cexCol=0.6)
```



Clustering performance

```
best=3
WC1C2 <- res@fit[[best]]@W
WTCN <- resC@fit[[best]]@W
Wtrue <- as.matrix(cbind(W, 1-rowSums(as.matrix(W))))
clustTRUE <- cutree(hclust(dist(Wtrue), method="ward.D2"),4
clustC1C2 <- cutree(hclust(dist(WC1C2), method="ward.D2"),4
clustTCN <- cutree(hclust(dist(WTCN), method="ward.D2"),4)

mclust::adjustedRandIndex(clustTRUE,clustC1C2)
```

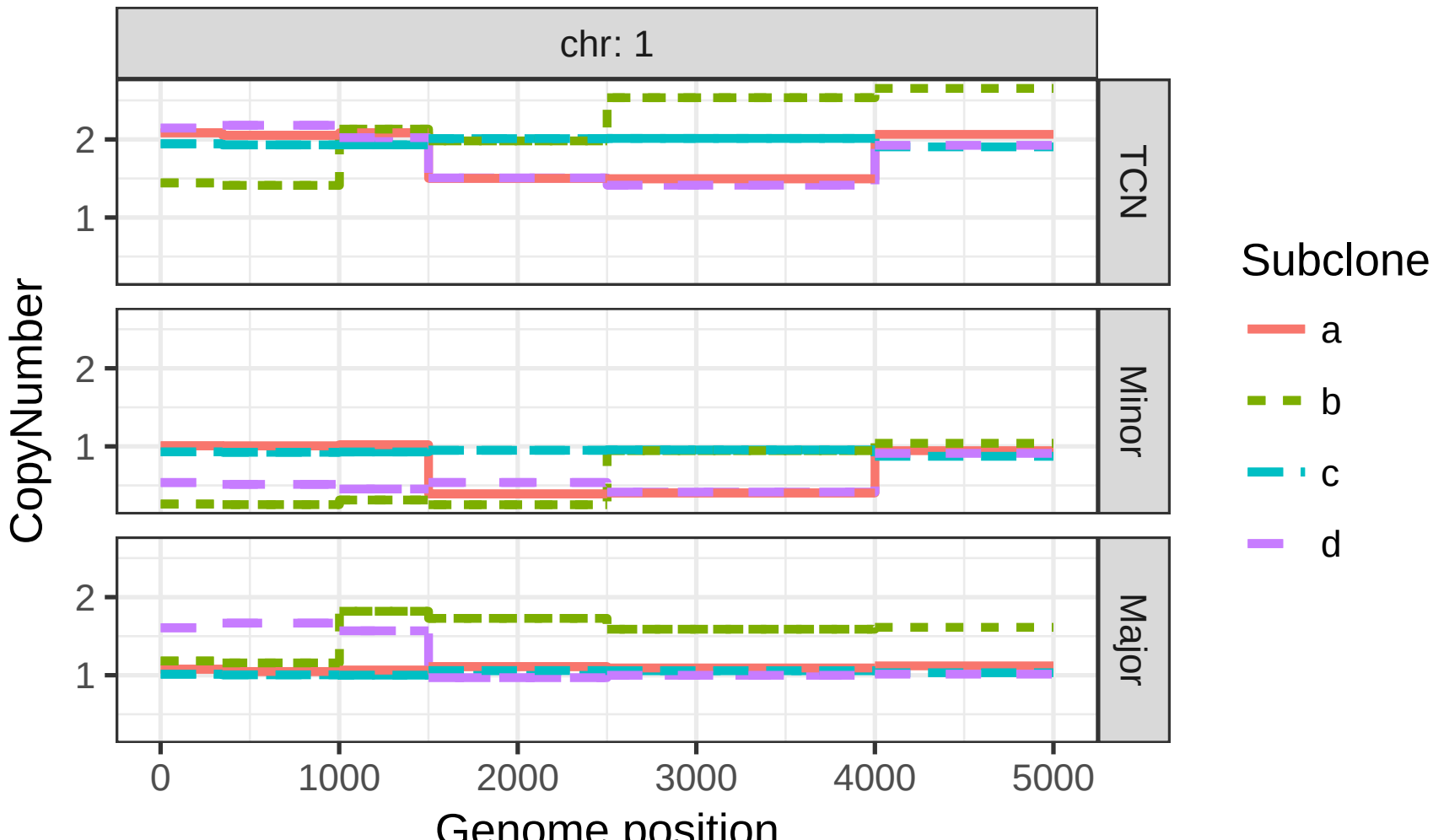
```
## [1] 1
```

```
mclust::adjustedRandIndex(clustTRUE,clustTCN)
```

```
## [1] 0.2354369
```

Matrix Z plot

```
df <- createZdf(res, chromosomes=1, idxBest = best)
Zplot(df)
```



Conclusion

Conclusion

c3co package aims to

- Solve a specific dictionary learning problem in cancerology context
- Determine composition for each sample in one tumor (dimension reduction)
- Recover alterations in subclones

Results :

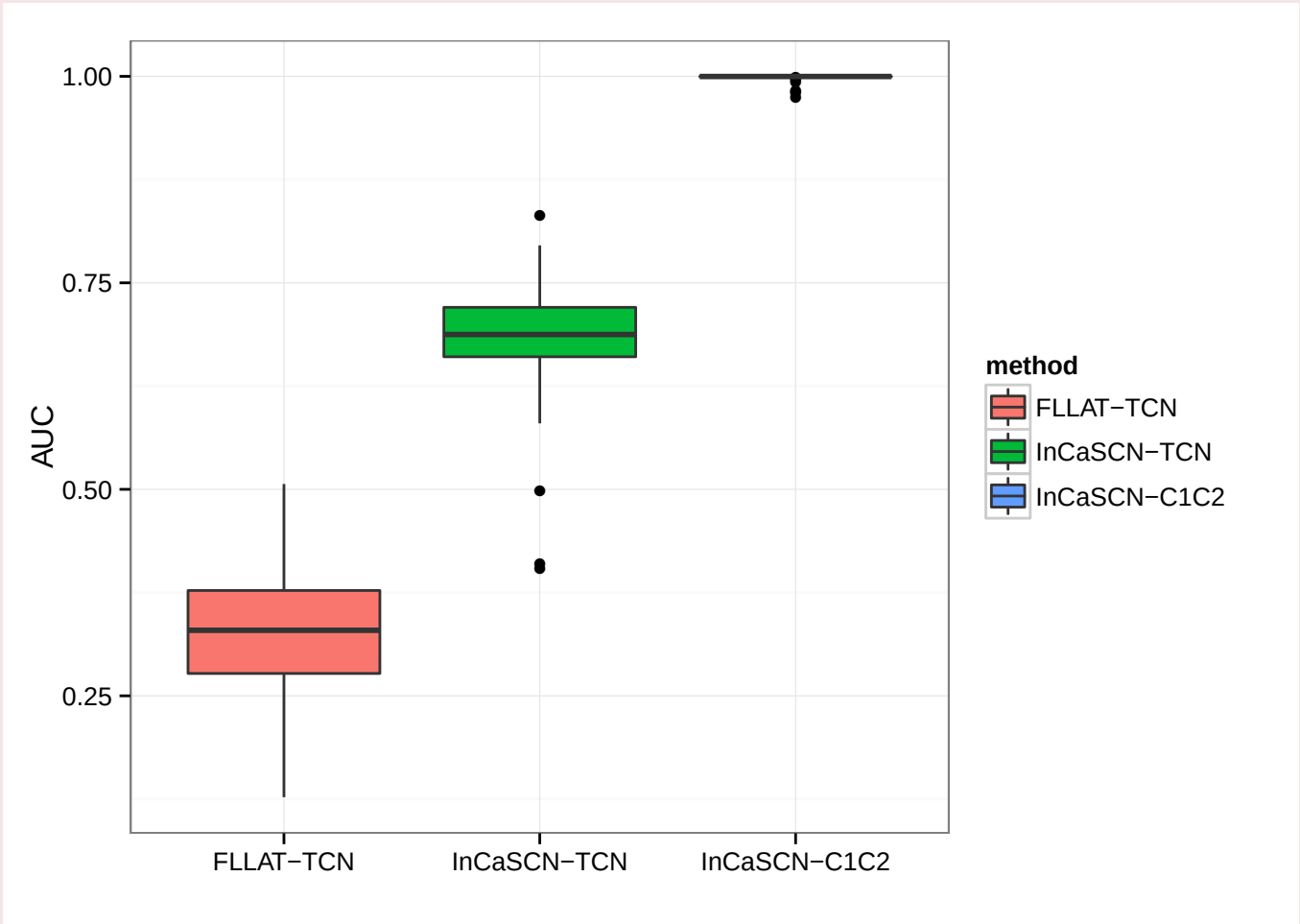
- c3co provides pretty good results on simulated data
- c3co has been applied on real data set

Thanks

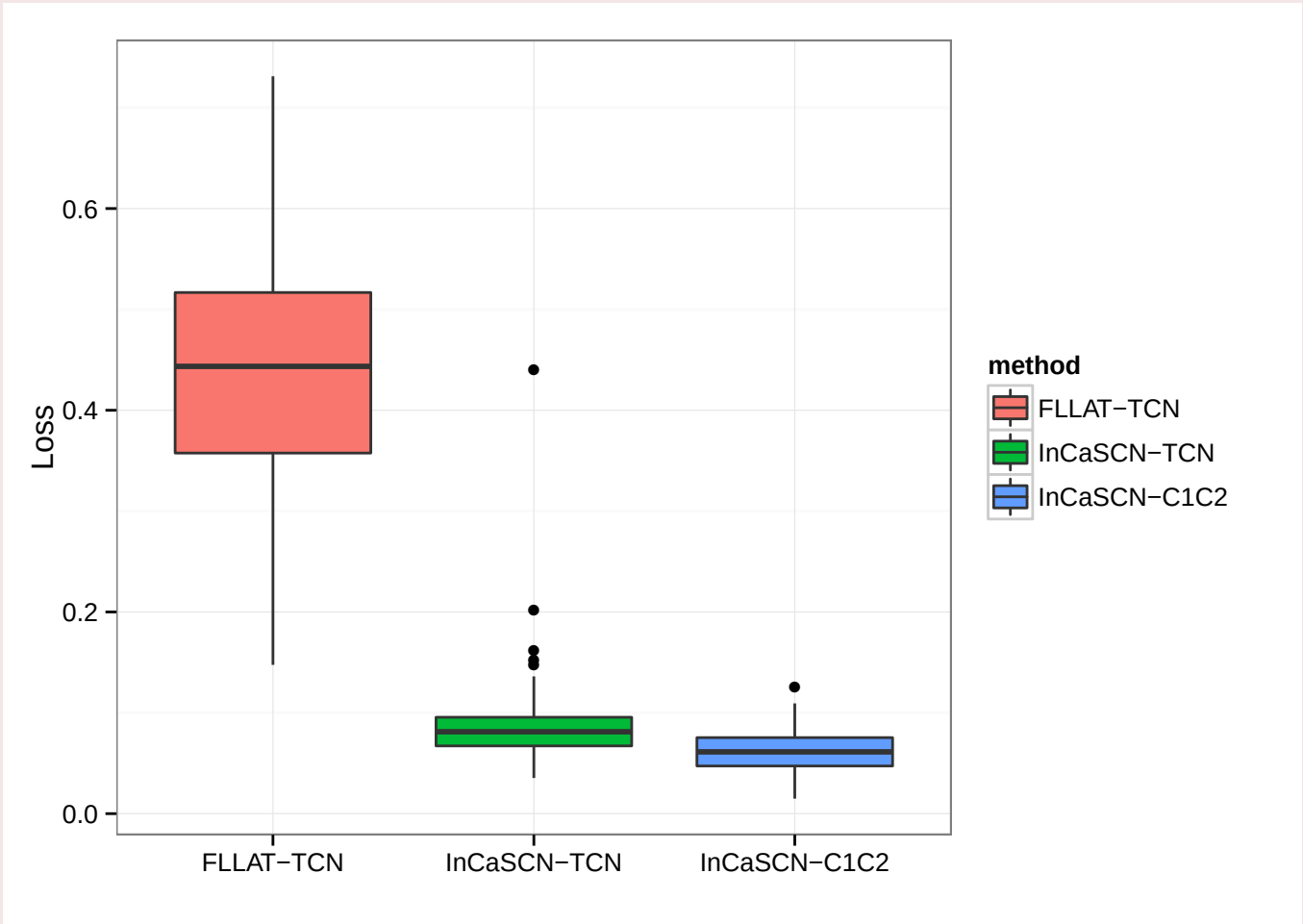
Thanks to : Henrik Bengtsson, Julien Chiquet and Pierre Neuvial

Some results on simulated data

Detection of alterations



Composition of tumors



Some results on simulated data

Clustering of samples