cghRA : a flexible workflow for CGH array analysis

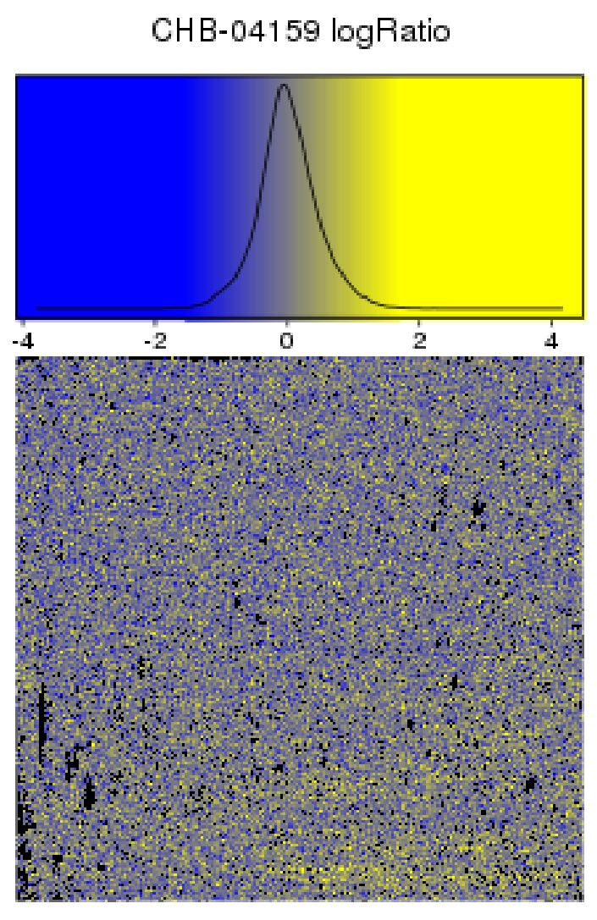
Sylvain Mareschal [1], Abdelilah Bouzelfen [1], Marion Alcantara [1], Philippe Ruminy [1], Martin Figeac [2], Christian Bastard [1], Hervé Tilly [1], Fabrice Jardin [1]

[1] INSERM 918, Centre Henri Becquerel, Rouen, France
[2] Plateforme de Génomique Fonctionelle, IRCL, France

Although Next Generation Sequencing technologies are becoming the new reference in pangenomic analysis, there is still a need for more affordable methods like CGH arrays to compare genomic alterations in large sample series. Despite the large collection of freely available un-interfaced algorithms and commercial software, biologists unfamiliar with command line interfaces and scripting lack a simple and efficient tool to handle such data. We describe here free interfaced software fulfilling this need, as well as several new algorithms able to enhance current handling of CGH array analysis, in areas like copy-number calling and polymorphism detection.

Interfaced R software ...

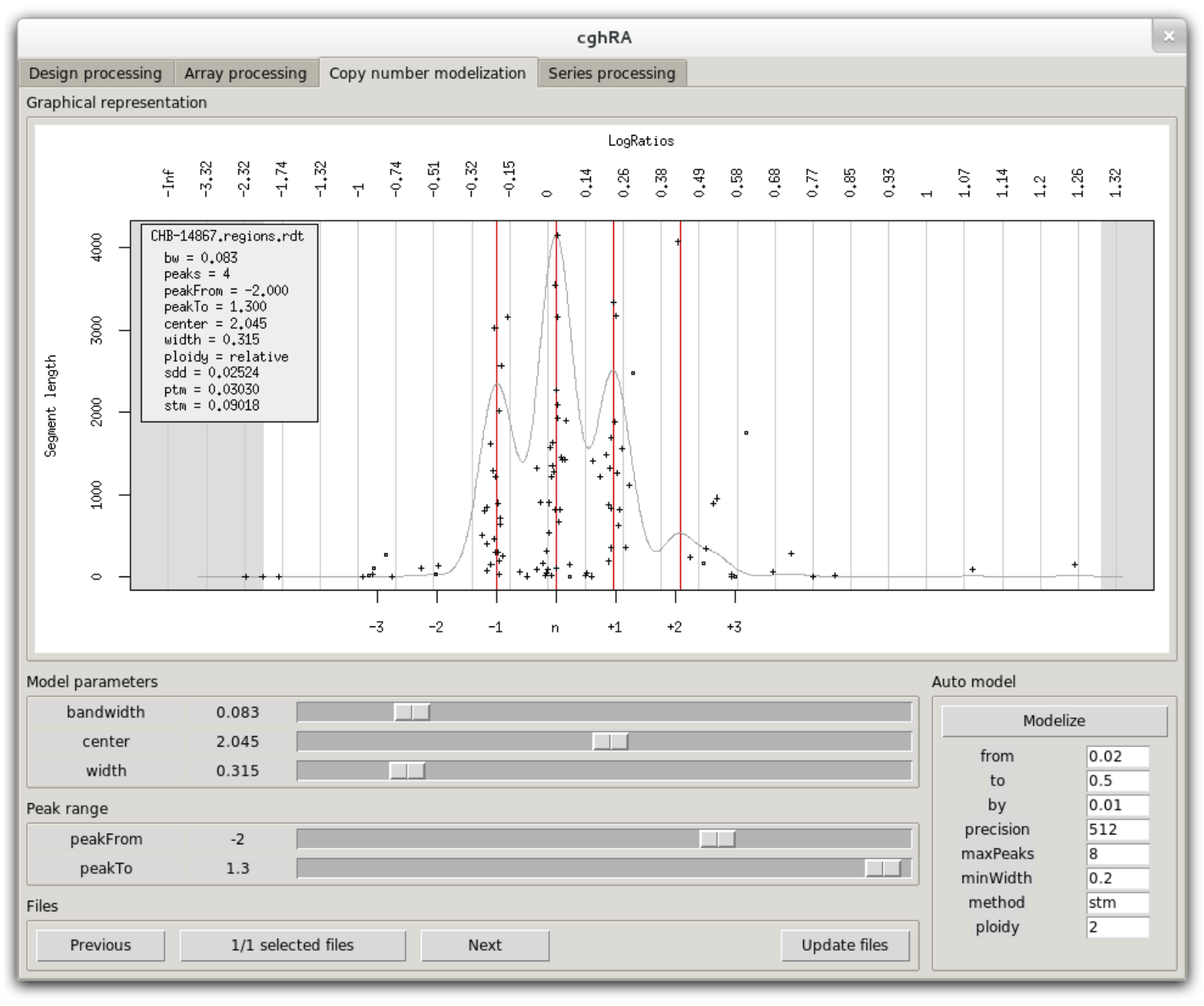
From raw files to biological results



cghRA is able to handle the whole analysis workflow, from array design processing to recurring event delineation. **Design processing** includes the parsing of design files provided by the array manufacturer, and optionally the remapping of probe sequences to any reference genome. **Array processing** proposes to filtrer probes based on scanning flags, visually check spatial biases, handle replicated probes and GC-related wave artifacts (*WACA, Lepretre et al NAR 2010*). Segmentation and copy-number calling based on a novel model are also provided. **Series processing** consists of segment annotation (genes, polymorphism score and custom tracks), recurring event definition using several algorithms and various graphical representations.

Graphical and Command-Line Interfaces

cghRA is implemented as a collection of "reference" classes, in a **R package**. As such, it is natively automatable and extendable, using a widespread scripting language offering robust statistical (via the R base) and CGH-related functions (via Bioconductor). A Tcl-tk **graphical interface** connects the various steps of the workflow, offering to users non familiar with R the opportunity to process their files in a "point and click" fashion. The graphical interface is itself a collection of independent R functions, offering the opportunity to **mix both interfaces** to make scripting easier.

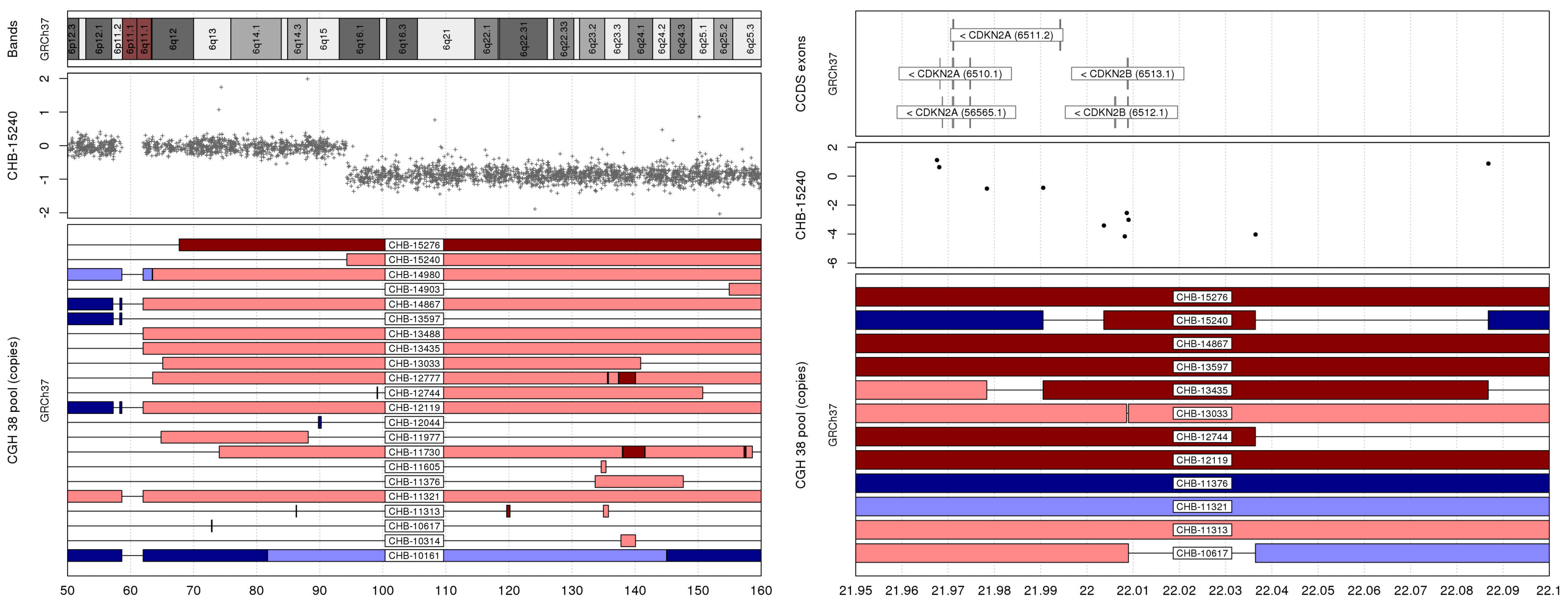


CHB-14867.regions.rdt
bw = 0.083
peaks = 4
peakFrom = -2.000
peakTo = 1.300
center = 2.045
width = 0.315
ploidy = relative
sdd = 0.02524
ptm = 0.03030
stm = 0.09018

bandwidth 0.083
center 2.045
width 0.315
peakFrom -2
peakTo 1.3

from 0.02
to 0.5
by 0.01
precision 512
maxPeaks 8
minWidth 0.2
method stm
ploidy 2

cghRA classes and files directly inherit from the **R Genome Browser** (*Rgb, Mareschal et al, Bioinformatics 2014*), allowing to interactively browse them along with provided and custom genome annotation.



Free and portable software

As fully implemented using **open-source** software, cghRA can be freely deployed on any operating system compatible with R, including **Windows**, **Mac OS** and a large collection of **Linux** distributions and architectures. R package sources and binaries, as well as Windows stand-alone versions will be made available at http://bioinformatics.ovsa.fr/cghRA.

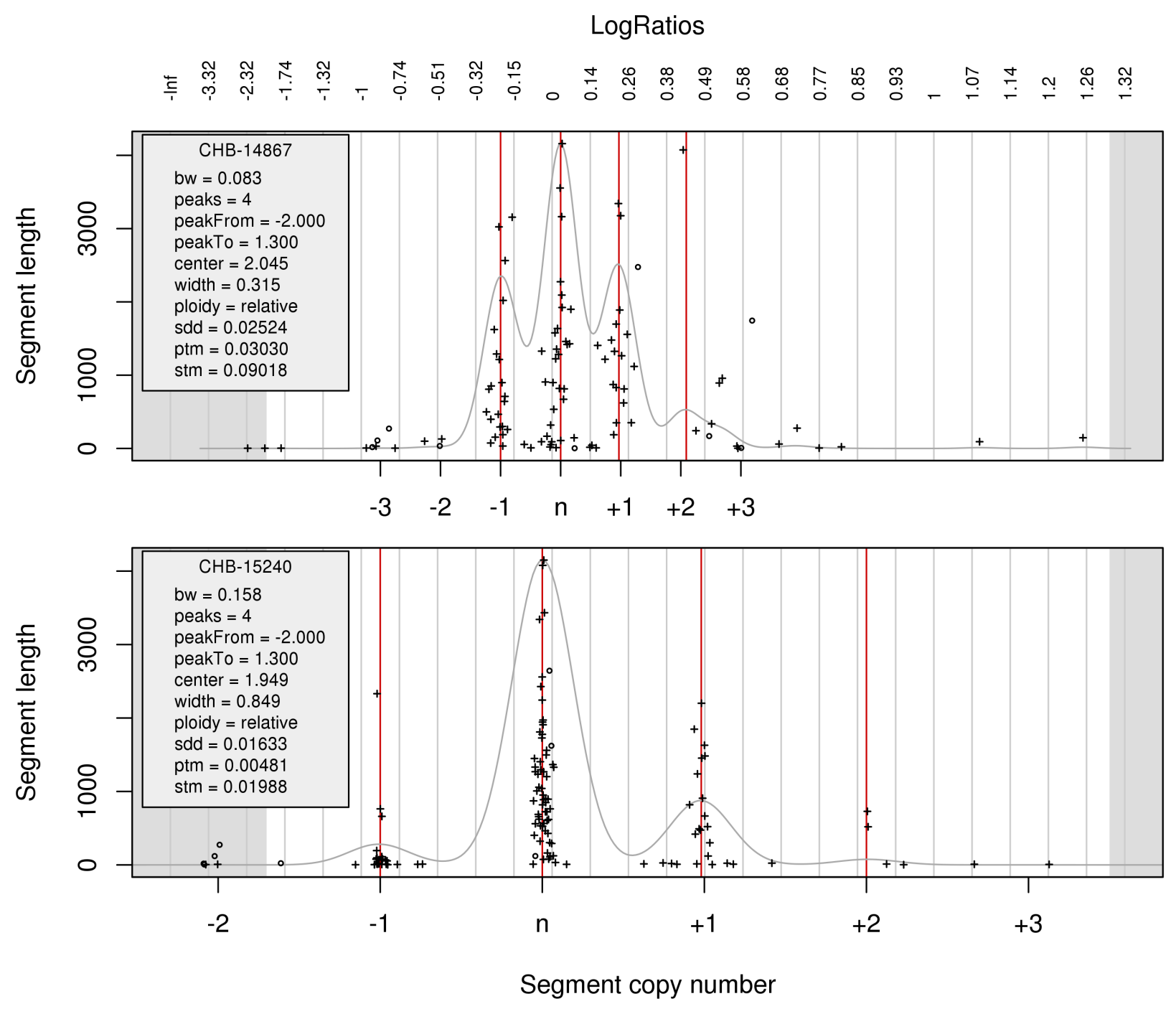
23th Annual International Conference on Intelligent Systems for Molecular Biology (ISMB)
14th European Conference on Computational Biology (ECCB)
2015 July 10th-14th, Dublin, Ireland

Contact : sylvain.mareschal@chb.unicancer.fr

... including exclusive algorithms

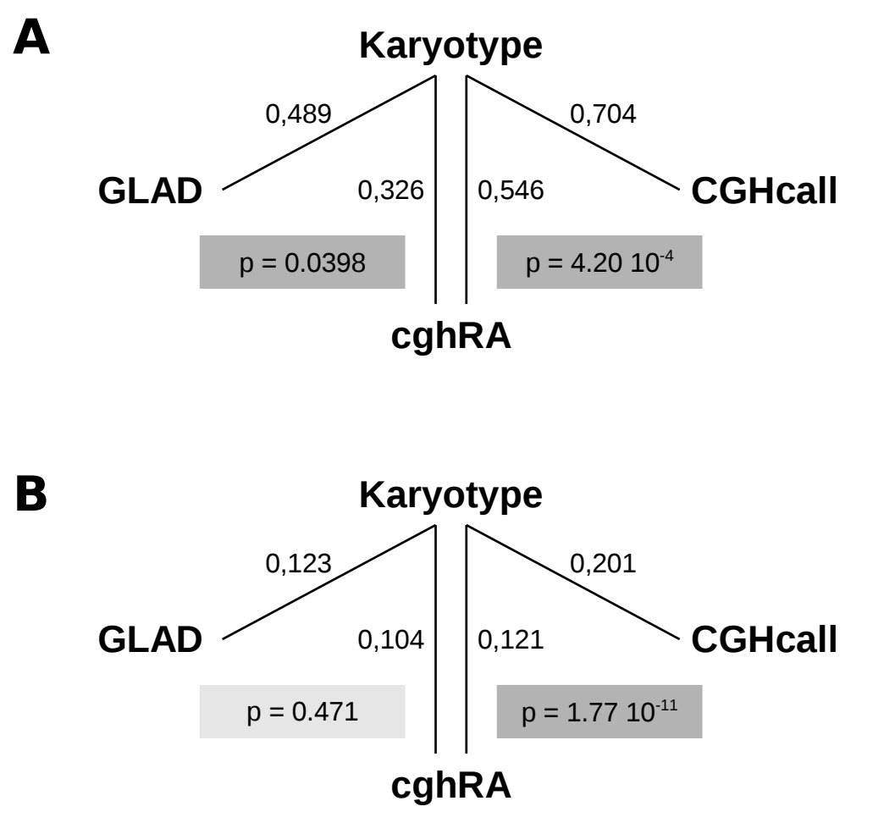
cghRA.copies – Sample-specific copy calling

Despite the high level of **heterogeneity in tumoral content** that can be observed between samples (from 20% to 90% in the lymphoma series used here), many researchers still use fixed arbitrary thresholds to define copy gains and losses, missing events in highly contaminated samples and over-interpreting the significance of subclonal events in pure samples.



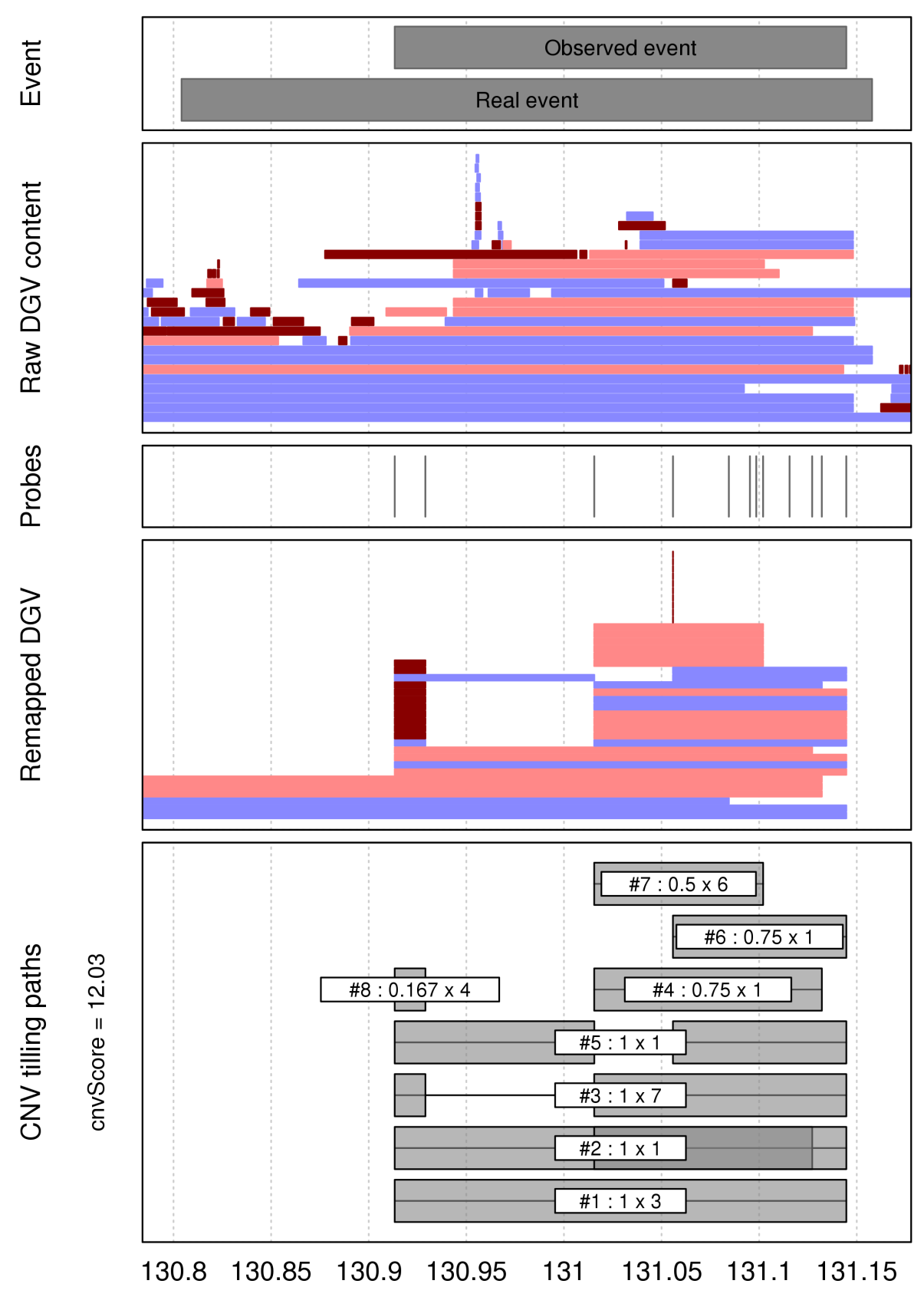
cghRA.copies proposes a copy-number model for each sample, based on segmented **log-ratio density distributions**. In an exponentiated space, evenly distributed peaks can be observed, as long as most copy number states are represented by at least a few segments. Several CBS segmentations are tried, and the model with minimal residuals is selected for copy calling.

cghRA.copies was compared to **CGHcall** (*van de Wiel et al, Cancer Inform. 2007*) and **GLAD** (*Hupé et al, BOE 2004*), two equivalent solutions proposed by Bioconductor, in a series of **77 Diffuse Large B-Cell Lymphomas** for which conventional karyotyping was also performed. It proved to be significantly more accurate than GLAD on the 17 polyploid samples (A), and than CGHcall in both diploid and polyploid (B) subsets.



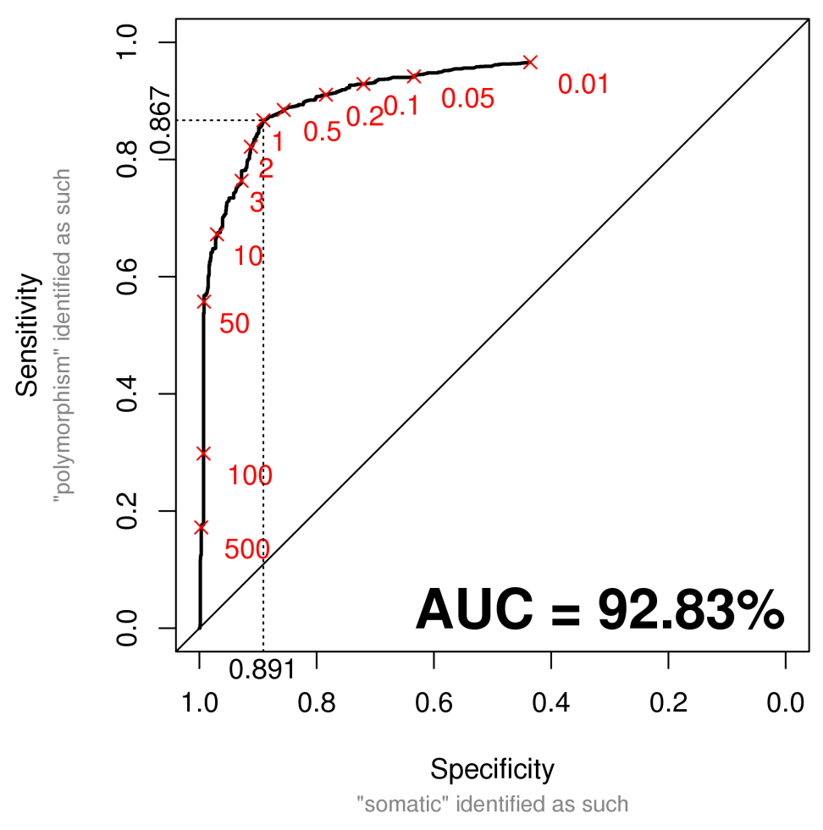
cnvScore – Filtering out polymorphisms

As matched normal sample is not always available for competitive hybridization against tumoral DNA, many CGH series use DNA pools as reference. This leads to the detection of **copy-number polymorphisms**, that need to be separated from somatic events.

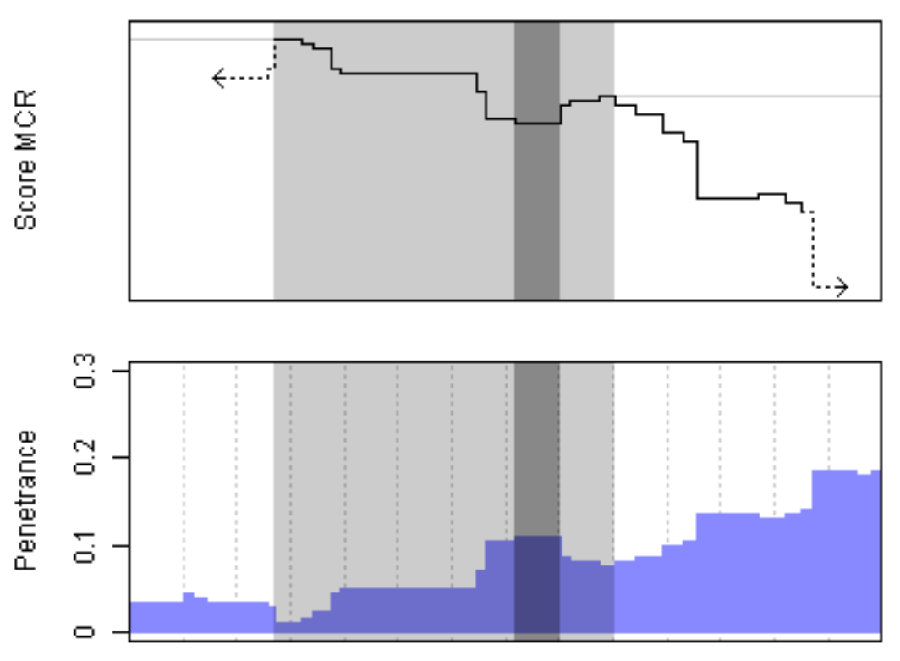


To this purpose, cnvScores that reflect the similarity between each segment and polymorphic events reported in the **Database of Genomic Variants** are computed. To minimize biases due to the very diverse techniques used to fill this database, several precautions were taken: only high-resolution CGH / SNP / NGS datasets were considered, and supporting events were **remapped to the user CGH design**. Distance between a segment to score and single or combined polymorphic events is then computed using the **Jaccard index**, relying on probe intersection rather than genomic spans.

87% sensitivity and **89% specificity** could be achieved in a validation series composed of **true somatic events** (38 DLBCL hybridized against matched normal DNA samples, Agilent 105k) and **true polymorphic events** not yet included in the DGV (NCBI dbVar estd212, 1000 Affymetrix CytoScan HD).



STEPS – Recurring event prioritization



While many attempts were made to devise statistical models able to assess significantly recurring events, available solutions such as GISTIC suffer from a lack of sensitivity that limits the conclusions that can be drawn from modest datasets (30 to 100 arrays). The STEPS algorithm we propose rather aims at **prioritizing Minimal Common Regions** (MCR), identified as localized peaks in the penetrance. While still in development, the resulting MCR score is less biased by independent chromosome arm level events than a simple prioritization based on maximal penetrance.