Supplementary Information: Computation of recurrent minimal genomic alterations from array-CGH data

C. Rouveirol, N. Stransky, Ph. Hupé, Ph. La Rosa, E. Viara, E. Barillot, F. Radvanyi

Appendix 1: Algebraic formalisation for (minimal) regions

Additional figures

The lattice of regions (i.e., closed sequences) of the context defined in table 1 of the paper is given in the figure 1.

Figure 2 illustrates the notions introduced in definition 10 of the paper.

Proofs of theorems

Proof of theorem 1.

Proof of \rightarrow . Let us prove that a minimal region r starts with an in-breakpoint. We note *in* and *out* the index of the first and last probe of the region (r = [in.out]). e denotes the extension of r.

Proof of (i). By contradiction. As $in \in r$, $\forall o_i \in e, M(o_i, in) = 1$. If in is not an in-breakpoint, then, according to definition 6, $\forall o_i \in e, M(o_i, in - 1) = 1$. As a consequence, in - 1 may well be added to the region, contradicting the statement that r is a closed sequence of probes. The demonstration that a minimal region ends with an out-breakpoint is of course similar.

Now let us prove that there is no other breakpoint in *in..out*[.

Proof of (ii). We will prove that if a single breakpoint b exists such that in < b < out, then the region r = [in..out] is not minimal. Let us assume, without loss of generality, that b is an in-breakpoint $(shift_in(b) \neq \emptyset)$. The region [b..out] has the extension $shift_in(b) \cup e$ and symmetrically, $ext([in..b] = shift_out(b) \cup e$. At least one of these regions [in..b] and [b..out] has an extension strictly larger than e, because $shift_in(b) \cup e = \emptyset$, and is a strict subsequence of r; according to definition 5, r is not a minimal region (see fig 3).

This concludes the proof of the \rightarrow part of theorem 1.

Proof of \leftarrow . Suppose that (1) r = [in..out], with *in* and *in-* and *out* an out-breakpoint, and *e* the extension of r (2) there is no breakpoint in]in..out[. First, let us prove that [in..out] is a region, i.e., it is closed in M. It cannot be extended on the left, because $ext(in - 1) \cap e = e \setminus shift _in(in)$ with $shift_in(in) \neq \emptyset$, by definition of an in-breakpoint (definition 6). Symmetrically, it cannot be extended on the right either $(ext(in+1) \cap e = e \setminus shift_out(out))$. Secondly, r is minimal: as there is no breakpoint in]in..out[, all probes between *in* and *out* have the same extension e, a strict subsequence of r that cannot be closed.

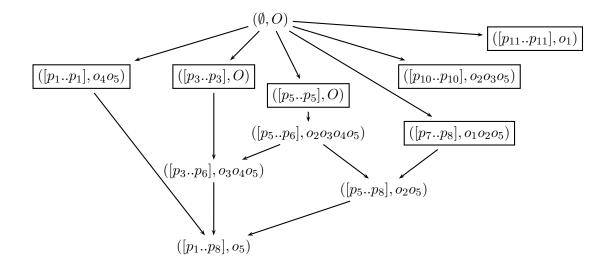


Figure 1: Lattice of closed sequences of probes according to table 1 of the paper. Each node of the lattice is a pair, the first element of which denotes a region, and the second element of which denotes the extension of the region. O denotes the set of all observations. Minimal regions are framed nodes.

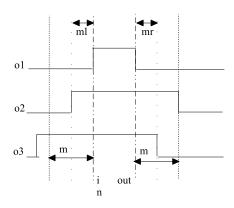


Figure 2: Left and right margins of a region [in..out], given a bound b = 2. [in..out] is well fuzzy bounded for bound 2 and for margin m.

Proof of Proposition 1. Each pair of closed sequences $cs_1 = [lb_1, ..., ub_1]$ and $cs_2 = [lb_2, ..., ub_2]$ has a single least upper bound. This lub is $cs_1 \cap cs_2$. $cs_1 \cap cs_2 = [max(lb_1, lb_2), ..., min(ub_1, ub_2)]$ if $max(lb_1, lb_2) \leq min(ub_1, ub_2)$ is a closed sequence. $cs_1 \cap cs_2$ is also the largest closed sequence subsequence of both cs_1 and cs_2 (follows from the anti-monotonicity of support with respect to \subseteq).

Each pair of closed sequences cs_1 and cs_2 has a single greatest lower bound. This glb is $closure_P(cs_1 \cup cs_2)$. $cs_1 \cup cs_2$ is $[min(lb_1, lb_2), ..., max(ub_1, ub_2)]$ if its extension is not empty (its extension is upper bounded by $ext(cs_1) \cap ext(cs_2)$). This sequence is not necessarily closed and therefore a closure step has to be applied.

Proof of theorem 2. This proof is based on the fact that procedure Next_Cands, given a closed sequence r = [in..out] of R(P) of extension e, generates all smallest closed supersequences of r. There are two smallest supersequences of r = [in..out], namely [(in-1)..out] and [in..(out+1)]. None of these sequences is closed in the general case, therefore it is necessary to apply a closure operator to ensure that closed sequences are obtained. Let us name $succ_l(r) = closure_P([(in - 1)..out]) = [in'..out']$ and $succ_r(r) = closure_P([in..(out + 1)])$. Note that the lowest bound of $succ_l(r)$ is not necessarily the closest in-breakpoint left of in, denoted lin: if $shift_i(lin \cup e) = \emptyset$, [lin..out] is not a closed sequence (see fig 4).

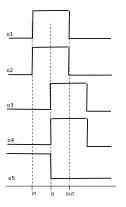


Figure 3: A minimal region does not contain any breakpoint. The figure shows that there would otherwise exist a smaller subregion of r = [in..out], which would be a candidate minimal region. According to definition 6, $shift_i(n) = \{o_1, o_2\}$, $shift_i(b) = \{o_3, o_4\}$, $shift_out(b) = \{o_5\}$, $shift_out(out) = \{o_1, o_2\}$. [in..out] has extension $\{o_1, o_2\}$, [in..b] has extension $\{o_1, o_2, o_5\}$, [b..out] has extension $\{o_1, o_2, o_3, o_4\}$.

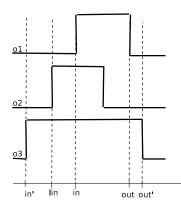


Figure 4: Left expansion of region [in..out] into the region [in'..out']

We need to prove the following lemma: crossing only one in-breakpoint (in) to expand r on the left and one out-breakpoint to expand the region on the right generates all smallest closed supersequences of r.

By contradiction for $succ_l(r)$: let us assume that there is a closed sequence r' such that $r \subset r' \subset succ_l(r)$. All these sequences are closed, so they have different extensions. By definition of extension $ext(succ_l(s)) \subset ext(s') \subset ext(s)$. If $ext(succ_l(s)) \subset ext(s')$, then there is necessarily an in-breakpoint between *in* and *in* - 1 that defines an intermediate extension for r', which is impossible.

 $succ_l(r)$ and $succ_r(r)$ are different in the general case $(shift_in(in) \neq shift_out(out))$. The first part of the algorithm returns both $succ_l(r)$ and $succ_r(r)$ if $ext(succ_l(r)) \not\subset ext(succ_r(s))$ and $ext(succ_r(s)) \not\subset ext(succ_l(r))$. Otherwise, it only returns the region with the largest extension (i.e. the smallest region).

Appendix 2: Complexity of CMAR algorithm

All candidate regions at a given level L, namely Cand(L), are evaluated against the constraints. The upper bound of |Cand(L)| is $N_P/2$ for L = 1, and $|Cand(L+1)| \leq 2 * |FailedOC(L)|$ for L > 1. If we assume that the context is a bit matrix of size $N_O * N_P$, computing the frequency of a region

r is in $O(N_O)$. The complexity of checking other constraints is negligible with respect to the cost of frequency computation.

The procedure that generates next level candidates Cand(L+1) performs two closure operations for each region of FailedOC(L). Assuming that the context is represented as a bit matrix, computing the closure of a region r is in $O(N_P)$. Testing candidate regions against CMR and FailedAC is linear in the size of $CMR \cup FailedAC$.

More stringent pruning may take place, depending on the properties of constraints defining the minimal region mining problem. For instance, constraints in OC can be such that two regions satisfying OC necessarily have an empty intersection (this is the case for well boundedness and fuzzy well boundedness). In this case, any candidate region in Cand(L+1) that overlap with (rather than is a superset of) a region of CMR can be safely pruned.

Appendix 3: Preprocessing - Gene and probe positioning

Preprocessing

Noise may originate from the label assignment step of GLAD for a region (see [3] for more details), because it is difficult to detect the reference level (i.e., normal) in an array-CGH profile, especially for high-grade, high-stage tumours. Assuming that probes labeled as normal follow a normal distribution of mean μ_n and standard deviation σ_n (see appendix 4 and 5), regions with a smoothing value between $\mu_n \pm \sigma_n$ are assigned to normal, whatever the status assigned by GLAD.

In this set of experiments, as in [5], we have chosen to select gain outliers with high log_2 ratio values and symmetrically loss outliers with low log_2 ratio values. More precisely, given the distribution of gain and loss outlier values (see appendix 4 and 5), we have selected only those gain outliers such that their log_2 ratio value is greater than the third quartile of the observed gain outlier distribution. Symmetrically, we have selected only those loss outliers such that their log_2 ratio value is less than the first quartile of the observed loss outlier distribution. Tuning the thresholds for selecting outliers is very data dependent; therefore such thresholds are parameters of the preprocessing step and can easily changed when running CMAR.

Probe and gene positioning

For assigning positions to the probes and genes on the human genome, the public databases used in this study were the UCSC Human Genome Working Draft sequence and the annotation database from the May 2004 freeze (hg17), the NCBI *Homo sapiens* Genome View build 35.1, the Homo sapiens UniGene Build 177. Genomic positions of the CGH *BAC* clones were obtained by searching for their accession ID or their associated STS or *BAC* End IDs in the UCSC annotation database tables and in the NCBI genome view tables. The genomic positions of the IMAGE clones were obtained by assigning them to a Unigene cluster using the Matchminer interface [1] and then looking for the position of the corresponding Unigene cluster sequences in the NCBI *Homo sapiens* Genome View database. Probes (i.e. *BACs* or cDNAs) that could not be positioned based on this Working Draft were ignored, as were probes classified as unmeasured by *GLAD* in all profiles of a given dataset.

Genomic positions of known genes (19,218), that we will refer to as our *reference gene set*, were obtained using the *knownGene* table from the UCSC annotation database. As the *BAC* coverage of the genome in the datasets handled was incomplete, each gene may or may not overlap a *BAC*. Based on known *BAC* positions and gene positions, we generated lists of genes located within a region. For a region defined by probes [in..out], the genes located in this region are those located between the end position of probe in - 1 and the start position of probe out + 1.

Appendix 4: Nakao et al. dataset characteristics

Gain and loss outlier selection

Figure 5 gives the distribution of gain and loss log_2 ratios for outliers in the Nakao *et al.* dataset. By selecting the n^{th} and $(100-n)^{th}$ percentile of the gain and loss outlier distributions, we therefore select gain and loss outlier that exhibit significantly high and low log_2 ratios, respectively. Therefore, in the Nakao *et al.* dataset, we chose as thresholds the 90^{th} and 10^{th} percentiles for gain and loss outliers' log_2 ratio distribution (i.e., 0.65 and -0.62). Note that these thresholds are less restrictive than those set in [4] (0.9 and -0.75 for amplification and homozygous deletion).

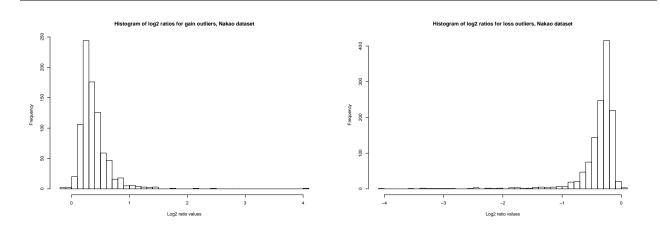


Figure 5: Distribution of log_2 ratio values for gain and loss outliers, Nakao *et al.* dataset

Normal probes distribution

Figure 6 shows the distribution of log_2 ratios for *BACs* flagged as normal in the Nakao *et al.* dataset. This distribution is a normal law, of s.d. 0.06.

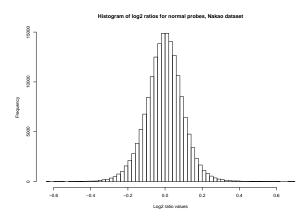


Figure 6: Distribution of the log_2 ratio values for normal probes, Nakao *et al.* dataset

Tuning the gain and loss margins

Figures 7 and 8 give the distribution of distances (in Mbp) between two consecutive breakpoints occurring on the same chromosome in the Nakao *et al.* dataset for gain and loss regions. This

distribution has been observed on the dataset after selection of gain and loss outliers, as described in the above section.

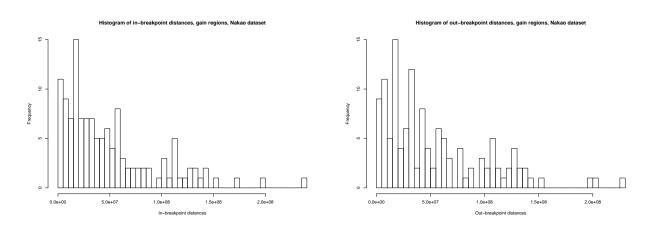


Figure 7: Distribution of the distances between two consecutive in-breakpoints (left curve) and between two consecutive out-breakpoints (right curve) occurring on the same chromosome, gain regions, Nakao *et al.* dataset

We distinguished, when computing these distributions, in- and out-breakpoints, and gain and loss regions. However, we did not observe a significant difference between left and right margins, computed as the 25th percentile of in- and out- consecutive breakpoint distance distribution, for a given status (gain or loss). Therefore, for regions of a given status, we have computed a single margin, denoted m_g and m_l . Each margin stands for both left and right margins for regions of a given status, and is computed as the maximum of both first quartiles of in- and out-breakpoint distance distributions for that status. This gives $m_g = 17.1$ Mbp (see fig. 7) and $m_l = 15.7$ Mbp (see fig. 8).

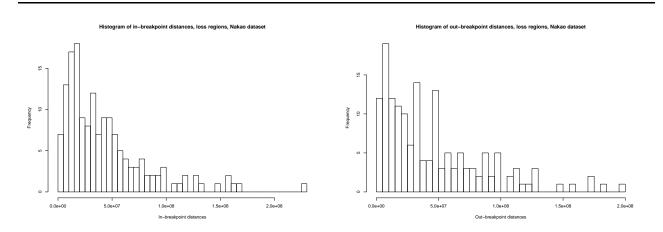


Figure 8: Distribution of the distances between two consecutive in-breakpoints (left curve) and between two consecutive out-breakpoints (right curve) occurring on the same chromosome, loss regions, Nakao *et al.* dataset

Appendix 5: Pollack et al. dataset characteristics

Gain and loss outlier selection

Figure 9 gives the distribution of gain and loss log_2 ratios for outliers in the Pollack *et al.* dataset. As for the Nakao dataset, we are interested in gain and loss outliers that exhibit significantly high and low log_2 ratios, respectively. However, selecting the 90th and 10th percentiles for gain and loss outliers for tumoural profiles was far too restrictive, as we obtained after selection too few and small regions. Therefore, we chose to set the pruning threshold to the 10th percentile of the log_2 ratio distribution for loss outliers in *normal* profiles, namely -1.13. This value corresponds to approximately the 20th percentile of the loss outlier distribution in tumoural samples. Symmetrically, we selected the 90th percentile of the log_2 ratio distribution of gain outliers in normal profiles (0.94), which corresponds roughly to the 80th percentile of the log_2 ratio distribution of gain outliers in tumoural samples.

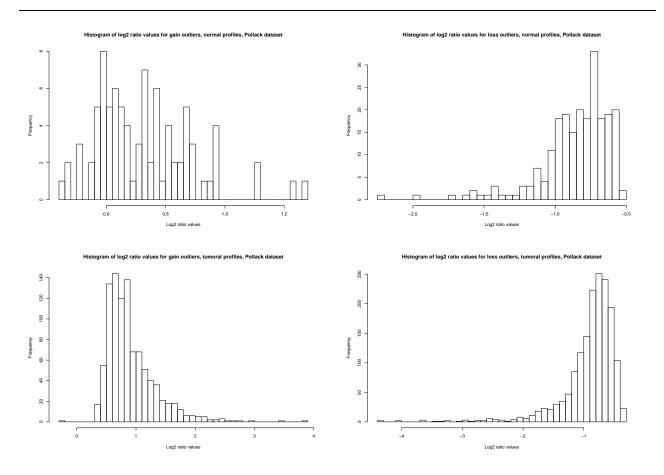


Figure 9: Distribution of the outlier log_2 ratio values for gain and loss outliers in normal (top curves) and tumoural profiles, Pollack *et al.* dataset.

Normal probes distribution

Figure 10 shows the distribution of log_2 ratios for probes flagged as normal in the Pollack *et al.* dataset. This distribution is a normal law, of s.d. 0.14.

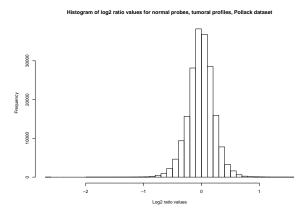


Figure 10: Distribution of the log_2 ratio values for normal probes, Pollack *et al.* dataset.

Tuning the gain and loss margin

Figures 11 and 12 give the distribution of distances (in Mbp) between two consecutive in- and outbreakpoints occurring on the same chromosome in the Pollack *et al.* dataset, after selection of outliers as described above.

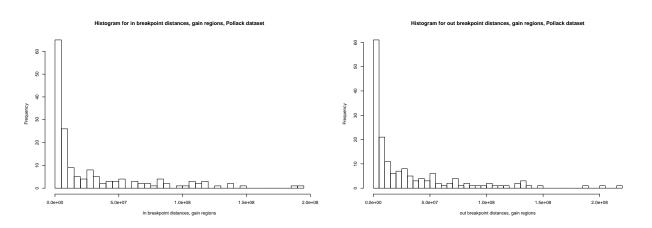


Figure 11: Distribution of the distances between two consecutive in-breakpoints (left curve) and between two consecutive out-breakpoints (right curve) occurring on the same chromosome, gain regions, Pollack *et al.* dataset.

Computing the margin for gain and loss regions as described in appendix 4 gives the following values for gain and loss margins within the Pollack *et al.* dataset: $m_g = 2.3$ Mbp (see fig. 11) and $m_l = 9.3$ Mbp (see fig. 12).

Appendix 6: De Leeuw et al. dataset characteristics

We have provided GLAD with the raw data as distributed by authors of [2], without probe realignement nor selection.

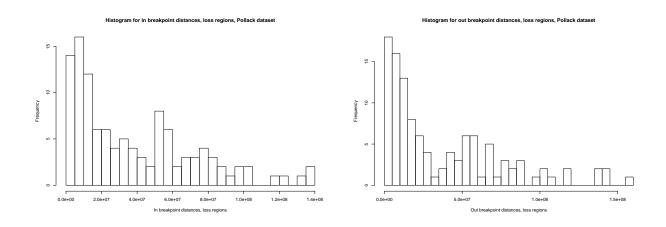


Figure 12: Distribution of the distances between two in-breakpoints (left curve) and between two out-breakpoints (right curve) occurring on the same chromosome, loss regions, Pollack *et al.* dataset.

Normal probes distribution

Figure 13 shows the distribution of log_2 ratios for probes flagged as normal in the De Leeuw *et al.* dataset. This distribution is a normal law, of s.d. 0.07.

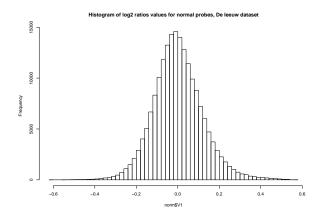


Figure 13: Distribution of the log_2 ratio values for normal probes, De Leeuw *et al.* dataset

Outlier selection

Dealing with noise (and therefore selecting outliers) is a less crucial issue here, because of the tiling technology used for the array: every isolated outlier can be safely smoothed. We have adopted the same parameters as in the publication of [2]: CMAR parameters were set to look for minimal regions that occurred in at least three out of the eigth samples, with a bound of 2.

Tuning the gain and loss margins

Figures 14 and 15 give the distribution of distances (in Mbp) between breakpoints occurring on the same chromosome in the De Leeuw *el al.* dataset for gain and loss regions. This distribution has been observed on the dataset after selection of gain and loss outliers, as described in the above section.

As for the previous datasets, we have set the margin parameter to the 25^{th} percentile of consecutive breakpoint distances distributions for gain and loss regions, namely about 1.5 Mbp for both gain and

loss margins.

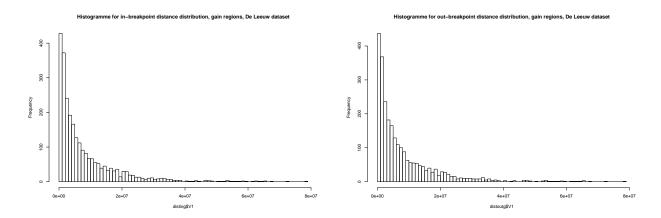


Figure 14: Distribution of the distances between two in-breakpoints (left curve) and between two outbreakpoints (right curve) occurring on the same chromosome, gain regions, De Leeuw *et al.* dataset

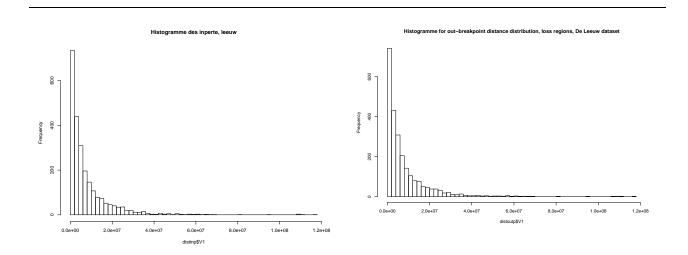


Figure 15: Distribution of the distances between two in-breakpoints (left curve) and between two outbreakpoints (right curve) occurring on the same chromosome, loss regions, De Leeuw *et al.* dataset

References

- KJ Bussey, D Kane, M Sunshine, S Narasimhan, S Nishizuka, WC Reinhold, B Zeeberg, A Weinstein, and JN Weinstein. Matchminer: a tool for batch navigation among gene and gene product identifiers. *Genome Biology*, 4, 2003.
- [2] RJ de Leeuw, JJ Davies, A Rosenwald, G Bebb, RD Gascoyne, MJ Dyer, LM Staudt, JA Martinez-Climent, and WL Lam. Comprehensive whole genome array cgh profiling of mantle cell lymphoma model genomes. *Hum Mol Genet*, 13:1827–37, 2004.
- [3] P Hupé, N Stransky, JP Thiery, F Radvanyi, and E Barillot. Analysis of array CGH data: from signal ratio to gain and loss of DNA regions. *Bioinformatics*, 20:3413–22, 2004.

- [4] K Nakao, KR Mehta, J Friedland, DH Moore, AN Jain, A Lafuente, JW Wiencke, JP Terdiman, and FW Waldman. High resolution analysis of DNA copy number alterations in colorectal cancer by array-based comparative genomic hybridization. *Carcinogenesis*, 25(8):1345–1357, 2004.
- [5] P Wang, Y Kim, J Pollack, and B Narasimhanand R Tibshirani. A method for calling gains and losses in array cgh data. *Biostatistics*, 6:45–58, 2005.