Collection of Biostatistics Research Archive
COBRA Preprint Series

Year 2008 *Paper* 43

Detection of Recurrent Copy Number Alterations in the Genome: a Probabilistic Approach

Oscar M. Rueda* Ramon Diaz-Uriarte†

*Spanish National Cancer Research Centre (CNIO)

†Spanish National Cancer Research Centre (CNIO), rdiaz02@gmail.com

Detection of Recurrent Copy Number Alterations in the Genome: a Probabilistic Approach

Oscar M. Rueda and Ramon Diaz-Uriarte

Abstract

Copy number variation (CNV) in genomic DNA is linked to a variety of human diseases (including cancer, HIV acquisition, autoimmune and neurodegenerative diseases), and array-based CGH (aCGH) is currently the main technology to locate CNVs. Several methods can analyze aCGH data at the single sample level, but disease-critical genes are more likely to be found in regions that are common or recurrent among samples. Unfortunately, defining recurrent CNV regions remains a challenge. Moreover, the heterogeneous nature of many diseases requires that we search for CNVs that affect only some subsets of the samples (without prior knowledge of which regions and subsets of samples are affected), but this is neglected by current methods.

We have developed two methods to define recurrent CNV regions. Our methods are unique and qualitatively different from existing approaches: they detect both regions over the complete set of arrays and alterations that are common only to some subsets of the samples and, thus, CNV alterations that might characterize previously unknown groups; they use probabilities of alteration as input (not discretized gain/loss calls, which discard uncertainty and variability) and return probabilities of being a shared common region, thus allowing researchers to modify thresholds as needed; the two parameters of the methods have an immediate, straightforward, biological interpretation. Using data from previous studies, we show that we can detect patterns that other methods miss and, by using probabilities, that researchers can modify, as needed, thresholds of immediate interpretability to answer specific research questions.

These methods are a qualitative advance in the location of recurrent CNV regions and will be instrumental in efforts to standardize definitions of recurrent CNVs and cluster samples with respect to patterns of CNV, and ultimately in the search for genomic regions harboring disease-critical genes.

Background

Copy number variations (CNVs) are often defined as DNA segments longer than 1 kb for which copy number differences are observed when comparing two or more genomes [1, 2]. CNVs have turned out to be much more abundant than previously thought [3–5] and have been linked to many different types of disease, including cancer, HIV acquisition and progression, autoimmune diseases, and Alzheimer and Parkinson's disease [4–7]. Identification of CNVs in individual samples nowadays uses mainly array-based Comparative Genomic Hybridization (aCGH), encompassing ROMA, oaCGH (including Agilent, NimbleGen, and many non-commercial, in-house oligonucleotide arrays), BAC, and cDNA arrays [8, 9], and SNP-based arrays [10, 11]. Location of CNVs in individual samples, however, is only the initial step in the search for "interesting genes". The regions more likely to harbor disease-critical genes are those that are recurrent among samples [9, 12–14]. In this context, we can define a CNV common region (CNVCR) as a set of contiguous genes (a region) that, as a group, shows a high enough probability (or evidence) of being altered (e.g., gained) in at least some samples or arrays (thus the usage of terms such as "common" or "recurrent"). Unfortunately, although many methods exist for analysing a single array of CGH (e.g., see references in [15, 16]), few papers deal with the problem of integrating several samples and finding common regions of alteration: merging data from several samples to define CNVCRs remains a challenge [2], both methodologically and conceptually.

One the most serious problems of existing methods is the inability to find common regions over subset of samples: existing methods try to find regions that are common to all the arrays in the sample and, thus, presuppose that a disease is homogeneous with respect to the pattern of CNVs. It is known, however, that many complex diseases, such as cancer [26] or autism "(...) consist largely of a constellation of rare, highly penetrant mutations" (p. S4 in [3]): we can observe a similar phenotype but we could arrive at this phenotype from several alternative DNA copy number alterations. Thus, it is absolutely crucial to differentiate between two different scenarios. In one scenario, we consider all the samples (subjects or arrays) in the study as a homogeneous set of individuals, so we want to focus on the major, salient, patterns in the data and thus we will try to locate regions of the genome that present a constant alteration over all (or most of) the samples. This is what existing methods for the study of CNVCR try to do. In a second scenario, we suspect that the subjects are really a heterogeneous group. What we really want here is to identify clusters or subgroups of samples that share regions of the genome that present a constant alteration. In other words, we want to detect recurrent alterations in subtypes of samples when we do not know in advance which are these recurrent alterations nor the subtypes of samples. This second scenario is

arguably much more common than the first one in many of the diseases where CNV studies are being conducted. In this second scenario, using an algorithm appropriate for the first scenario (one that, by construction, tries to find alterations common to most arrays) is clearly inappropriate: it does not answer the underlying biological question, risks missing relevant signals, and leads to conceptual confusion. A few methods [12, 13, 18, 25] can, under certain circumstances, find recurrent regions defined over a subset of the samples, but this is highly dependent on the amplitude of the alteration and the signal to noise ratios (e.g., pp. 1483–1484 in [25]); it is more an accidental by-product of very large alterations, and not an explicit objective of these methods *per se*. Thus, a method that directly and explicitly addresses the second scenario is sorely needed: it is the only way to obtain CNVCR under sample heterogeneity

In addition to being appropriate only for the first scenario (homogeneous samples), existing methods present other limitations. Most of the methods [12, 13, 17–22] try to find CNVCRs using, as starting point, the discrete output from an aCGH segmentation algorithm in the form of the classification of every probe into gained, normal or lost. By using this discretized output, these methods discard any possible measure of the uncertainty of these estimates; as a consequence, a gain for which there is strong evidence will have the same weight in subsequent calculations as another gain for which there is less certainty. Moreover, the majority of these methods ignore within- and among-array variability in aCGH ratios as they use a common threshold for all probes and arrays. Finally, some algorithms (e.g., [22]) weight the amplitude or magnitude (e.g., $\log_2$ ratio) of an alteration by the number of samples: we can obtain the same summary statistic from, say, 10 probes with a ratio of 1 as from 1 probe with a ratio of 10, but these are two clearly distinct biological scenarios. More generally, this scheme implicitly equates evidence of alteration with magnitude of alteration and makes it harder to detect small (but almost sure) amplitude alterations shared by many samples relative to large amplitude alterations shared by fewer samples. A few other methods perform the segmentation and search for CNVCR in the same step [23–25]. In [25], in addition to not using nor returning probabilities, elaborate and heuristic approaches are required to search over possible thresholds and adjustments for multiple testing. In [23] copy numbers of contiguous probes as treated as independent, which is clearly biologically unrealistic. Hidden Markov Models are used by [24], but the number of states is restricted to four; therefore, all the gains are grouped into a single state with a common mean, which is biologically unreasonable, and makes it impossible to differentiate between samples with moderate amplitude changes and large-amplitude changes.

Existing methods, therefore, have serious limitations and it is necessary to develop new approaches that fulfill the following three major requirements. First, we want to explicitly differentiate between the two

scenarios in the last paragraph. As a consequence, we want to be able to locate either regions common to most of the arrays or regions which might be common to only a subset of the arrays. Second, we want to preserve the uncertainty in the state of a probe (probability of alteration), and we want to return probabilities, as a probability is the single most direct answer to the question "is this region altered over this set of arrays?" (a p-value does not directly answer this question, but rather provides support against a specific null hypothesis). Third, we want that, biologically, the meaning of the regions found be immediate, and depend on few parameters of straightforward interpretation.

Results

Two different approaches for finding CNVCRs

Here we provide an intuitive understanding of our two different approaches. Further details are provided below.

Our first method, **pREC-A** (probabilistic recurrent copy number regions, common threshold over all arrays), finds only those regions that, over the complete set of arrays, show an average (over arrays) probability of being altered that is above a predefined threshold. When using **pREC-A** we only need to provide one threshold, p_a, the minimal probability of alteration of a region over a set of arrays. p_a is chosen by the researcher, but generally cannot be too stringent (e.g., will rarely be larger than 0.80) because even with a large number of arrays, only a few arrays without that alteration will prevent finding the region (as we are averaging over arrays).

Our second method, **pREC-S** (probabilistic recurrent copy number regions, subsets of arrays), identifies common regions over subsets of arrays; alternatively, we can think of this algorithm as identifying subsets of arrays that share regions of alteration. The regions of alteration found might not be common to most arrays, but within each array in the identified subset, the regions of alteration will have a probability of being altered above a threshold (p_w). When using **pREC-S**, therefore, the user needs to provide two thresholds, p_w, the minimal probability of alteration of a region in every array in the selected subset, and $freq.array$, the smallest number of arrays (i.e., the smallest size of the subset of arrays) that share a common region. Here we will often use more stringent thresholds for probability (e.g., $p_w = 0.90$), because those high probabilities might be attained over a highly homogeneous and small subset of arrays. We can use the output of **pREC-S** as the basis for clustering and to display patterns of groupings of arrays; an example is shown in section .

For both methods, we will use probabilities of alteration as returned, for example, by our RJaCGH

method [15]. RJaCGH is a Hidden Markov Model-based approach that returns probabilities of alteration of probes and segments; no hard thresholds are imposed, and thus the user decides what constituted sufficient evidence (in terms of probability of alteration) to call a probe gained (or lost). We have shown [15, 16] that our method performs as well as, or better than, competing methods in terms of calling gains and losses, and the relative advantage of our method increases as the variability in distance between probes increases. It is essential to understand that the probabilities that we use are not the marginal probabilities of alteration but the joint probabilities of alteration of a region of probes (see details in "Methods"). Our approach incorporates both within- and among-array variability (as it is based on the hidden process of alterations and uses the probability of every probe in every array): we use the information on the certainty of each call of gain/loss (i.e., the probability) in all computations of CNVCR. Therefore, our approach is qualitatively different from using the same threshold over all probes and arrays. See further details in "Methods". Moreover, using probabilities of alteration (instead of magnitude of change), in addition to differentiating between evidence of alteration and estimated fold change, prevents inter-array differences in range of log_2 ratios and tissue mixture to get confounded with evidence of alteration. Finally, note that we use at most two parameters and that their biological meaning is immediate: probability of alteration, and number of samples that share an alteration (the later only needed for **pREC-S**).

Algorithms

Before we can develop algorithms for the two approaches, **pREC-A** and **pREC-S**, we will need to develop methodology that will allow us to: 1) compute the joint probability of alteration of an arbitrary sequence of probes; 2) combine that probability over arrays. The first two parts of this section detail this machinery before showing the details of the algorithms. For the rest of this section, please bear in mind that we are always referring to probabilities of alteration, and never to p-values. We are working on a Bayesian framework and are estimating posterior probabilities; we are not conducting hypothesis tests.

Computation of the joint probability of an arbitrary sequence of probes in an array

To find altered regions, that is, sets of contiguous probes, we have to compute the joint probability of alteration for a sequence of probes. In other words, we need to compute, for each array $i = 1, ...r$, the probability that a subset of consecutive probes is, for example, gained (the problem for losses is equivalent). That is, if we denote as S_i the state of probe i and with 1 the state 'gain', we are interested in $P(S_j = 1, \ldots, S_{j+p} = 1)$ for a subset of p probes.

Using RJaCGH (or other methods) we can compute the probability for every probe to belong to any of the states of gain and to any of the states of loss. The problem of these probabilities is that they are marginal probabilities: they are the probability of the event of an alteration of a probe without considering the alteration of other probes, in particular of neighboring probes. But the states of the probes are not independent [15], and thus the probability of alteration of a region (within an array) can not be computed simply as the product of the probability of the individual probes.

With HMM it is customary to obtain the most likely path of hidden states using the Viterbi algorithm which returns the maximum a posteriori sequence (MAP). The Viterbi algorithm, however, does not return any distributional statements about the states of the path [27]. It is straightforward, however, to compute the marginal probabilities of the state of a probe or the joint probabilities of an arbitrary sequence of probes, because the sequence of hidden states conditioned on the parameters of the HMM is a Markov Chain [27]. For instance, we could compute the probability that the first three probes are jointly gained: $P(S_1 = 1, S_2 = 1, S_3 = 1)$ using straightforward conditional probabilities as $P(S_1 = 1)P(S_2 = 1|S_1 = 1)P(S_3 = 1|S_2 = 1)$, and these conditional probabilities can be computed by backward-smoothing. The problem is that the classification of probes or regions into states given by these two approaches (Viterbi and backward-smoothing) does not always coincide, leading to inconsistencies. For example, we might obtain a sequence of hidden states with maximum marginal probabilities that is not the same as we obtain with Viterbi; that sequence might even contain two consecutive altered probes that can not be jointly altered [28]. This is a common problem that can arise when using maximum likelihood approaches to HMM.

To avoid these problems, we can use, as RJaCGH does, Markov Chain Monte Carlo (MCMC) instead of ML. With MCMC, however, we can not average the conditional probabilities obtained through the MCMC iterations, because that would break the Markovian property [29], as we are averaging over different runs with (potentially) different parameters. For instance, suppose we want to compute the probability that the first three probes are jointly gained: $P(S_1 = 1, S_2 = 1, S_3 = 1)$. We cannot compute $P(S_1 = 1)P(S_2 = 1|S_1 = 1)P(S_3 = 1|S_2 = 1)$, with those conditional probabilities obtained by averaging over the multiple MCMC runs. What we can do, instead, is compute the probability of an alteration for any arbitrary sequence as the frequency of that sequence being altered in the MAPs from each of the MCMC draws. For the previous example, we would count in how many MAPs (from Viterbi) we found $S_1 = S_2 = S_3 = 1$. We must note that, in this case, we are not obtaining the real distribution of the hidden states per se, but the distribution of the hidden states as members of the maximum a posteriori hidden

sequence [30]. That is, we do not sample from the distribution of the hidden states, but from the distribution of the MAP. This is coherent with the classification method used with just one array, as every sequence is only accounted for if it has been part of the MAP sequence, and thus this is a stronger requirement as the regions obtained have always been part of the MAP.

Finally, the above scheme can be applied both to models that assign to hidden states probabilities of being altered of either 1 or 0, and to models that assign to hidden states probabilities of being altered between 0 and 1.

Combining regions over arrays

Once we have computed the probability that the above region is altered, for our first algorithm, **pREC-A**, we need to know how to average over the arrays to get a probability of alteration for that region over a set of arrays. Many HMM models (RJaCGH included) will model each array with a different HMM, to reflect the fact that they can have different characteristics, such as dispersion. Thus, for each array, we have a (potentially different) stochastic process for the log-ratios. Once the data are summarized as states (gain, loss, no-change), however, they are comparable across arrays as we are using the same approach to label probes as gained/lost/not-changed. In other words, a value of $S_j = 1$ has the same meaning regardless of the array. Thus, we can average directly all the probabilities for every array (the averages might be weighted if there are differences in the reliability or the precision of different arrays). Therefore, the probability that a given region of the genome is altered over a set of arrays is computed as:

$$P(S_i = 1, \ldots, S_{i+p} = 1) = \sum_{j=1}^{r} P(S_i = 1, \ldots, S_{i+p} = 1 | array_j) P(array_j) \quad (1)$$

where different $P(array_j)$ allow us to use different weights for different arrays (and, of course, the $P(array_j)$ are scaled, if needed, so that $\sum_j P(array_j) = 1$).

For notational convenience, when there is only one probe, we define

$$P(S_i = 1) = \sum_{j=1}^{r} P(S_i = 1 | array_j) P(array_j) \quad (2)$$

pREC-A: Finding regions with a probability of alteration of at least p_a

The following algorithm finds all the regions with an average (average over all arrays) probability of alteration of at least p_a. After the algorithm we provide a detailed explanation.

```
1  Start ← 1
2  while Start ≤ TotalNumberOfProbes do
3  |  P1 ← P(S_Start = 1);
4  |  if P1 ≥ p_a then
5  |  |  End ← Start + 1;
6  |  |  while End ≤ TotalNumberOfProbes do
7  |  |  |  P2 ← P(S_Start, ..., S_End = 1);
8  |  |  |  if P2 < p_a then
9  |  |  |  |  break out of the while loop;
10 |  |  |  else
11 |  |  |  |  P1 ← P2;
12 |  |  |  |  End ← End + 1;
13 |  |  UpdateRegionA(Start, End − 1, P1);
14 |  |  Start ← End;
15 |  else
16 |  |  Start ← Start + 1;
```

Algorithm 1: pREC-A algorithm

The search for common regions starts on the first probe of every chromosome. If the average probability of alteration over arrays fulfills the p_a criterion (line 4) we examine if we can add probes to this region, until no further probes can be added to the region, which is equivalent to $P2$ falling below p_a (line 8). If the probe we considered as *Start* does not fullfill p_a, the next probe is considered as starting probe (line 16). The function `UpdateRegionA` (called in line 13) adds a region to the set of regions already stored. `UpdateRegionA` records the first and last probes of the region (*Start* and $End - 1$) and the average probability of the region ($P1$, as computed in line 3 or $P2$ as computed in line 7). This function can only be called if at least the probe *Start* fulfills the p_a criterion (as the call is inside the "If" condition in line 4). We can call `UpdateRegionA` either if we are at the end of a chromosome (so there are no further probes to consider for extending a region: line 6 is not satisfied) or if the probe we just considered for addition to the region results in the average probability of the region ($P2$) to drop below p_a (line 8). The rest of the algorithm is mostly in charge of appropriately updating *Start*, *End*, $P1$, and $P2$, so that we can directly call `UpdateRegionA` (line 13) with the same arguments and without further conditional checks. Note that calling `UpdateRegionA` with $End - 1$ (and not *End*) is what we want to do to ensure that the correct last probe of a region is recorded, regardless of whether we reach line 13 from line 8 or from exiting the while

loop (line 6).

Computationally, when finding $P2$ (line 7), and for a given $Start$, we do not need to repeatedly compute $P2$ over all probes of a region: it is much faster to simply update the $P2$ probability as we add one probe at a time at the end of the region (i.e., as we increase End).

Line 14 ensures that, when we cannot add any probes to a region (because the probability falls below p_a), the probe that will be considered as $Start$ candidate for the next region is the one immediately following the End of the last accepted common region. As a consequence, this algorithm ensures that a probe that has a marginal probability higher that the threshold will always be part of a region (at least it will be a region itself), but does not uniquely define the regions (uniqueness is guaranteed for probes). For example, suppose we are interested in finding regions of gain of at least 0.90 probability. We can have the following situation with three probes:

$$\begin{aligned} P(S_1 = 1) &= 0.95 \\ P(S_1 = 1, S_2 = 1) &= 0.90 \\ P(S_1 = 1, S_2 = 1, S_3 = 1) &= 0.89 \\ P(S_3 = 1) &= 0.95 \\ P(S_2 = 1, S_3 = 1) &= 0.90 \end{aligned}$$

Our algorithm would return two regions, $\{S_1, S_2\}$ and $\{S_3\}$. But the regions $\{S_1\}$ and $\{S_2, S_3\}$ are also valid. Accounting for these effects computationally will slow down the algorithm and, biologically, it is of no relevance because all three probes are always included in the set of regions.

Of course, the joint probability of all regions returned by this algorithm is not necessarily larger than the threshold p_a: each region has a probability of at least p_a, but this does not guarantee that, jointly, all regions have a probability of at least p_a.

This algorithm is the one that is most similar to other existing approaches in objective. Notice, however, the simplicity of our algorithm, and the straightforward interpretation of its parameters.

pREC-S: Finding all the regions shared by at least $freq.array$ arrays where each region in each array has a probability of at least p_w

We are imposing two thresholds: 1) p_w, the minimum joint probability, within array, for each region; 2) $freq.arrays$, the minimum number of arrays that share the alteration. Notice that p_w in this algorithm is

different from p_a in the previous algorithm (where averaging over arrays is used).

```
1  for Start ← 1 to TotalNumberOfProbes do
2      SetArrays_A ← φ ;
3      for array ← 1 to TotalNumberOfArrays do
4          if P(S_Start = 1|array) ≥ p_w then
5              SetArrays_A ← SetArrays_A ∪ array;
6      if |SetArrays_A| ≥ freq.arrays then
7          End ← Start + 1;
8          while End ≤ TotalNumberOfProbes do
9              SetArrays_B ← φ;
10             foreach candidate_array in SetArrays_A do
11                 if P(S_Start, ..., S_End = 1|candidate_array) ≥ p_w then
12                     SetArrays_B ← SetArrays_B ∪ candidate_array;
13             if |SetArrays_B| < freq.arrays then
14                 break out of the while loop
15             else
16                 if |SetArrays_B| < |SetArrays_A| then
17                     UpdateRegionS(Start, End − 1, SetArrays_A);
18                     SetArrays_A ← SetArrays_B;
19                 End ← End + 1;
20         UpdateRegionS(Start, End − 1, SetArrays_A);
```

Algorithm 2: pREC-S algorithm

The logic of this algorithm is very similar to that of pREC-A, above. The function `UpdateRegionS` (called in lines 17 and 20) adds a region to the set of regions already stored. Adding a region means storing the first probe of the region ($Start$), the last probe of the region ($End - 1$), and the arrays that compose the region (those in $SetArrays\_A$). (Because of the way that End and $SetArrays\_A$ are updated, End and $SetArrays\_A$ are always the correct arguments to this function). The function `UpdateRegionS`, however, must check that the region to be added is not a subset of some previously added region. Suppose in the run that started with probe $S2$ we found the region $((S2, S3, S4), (A1, A2))$. Now, in the run that starts with probe $S3$ we find the region $((S3, S4), (A1, A2))$; obviously, the newly found region is simply a completely contained subset of the previously found region, and we should not add this newly found region as a new region.

The conditions in lines 4 and 11 refer to one of the conditions of the algorithm: an array can only be considered part of a common region if the probability of the given sequence of probes (starting at $Start$ and ending at End or, in the one-probe case, starting and ending at $Start$) is larger than p_w. Likewise, the conditions in lines 6 and 13 refer to the second condition: at least $freq.arrays$ arrays must fulfill that the

sequence has a probability larger than p_w.

Line 16 represents the condition where the number of arrays that fulfill the condition when we add a probe decreases. In other words, at step t, with $End = Start + t$, we had a set of arrays that fulfilled p_w. As soon as we add a new probe (i.e., "stretch" the region by one probe, so we are at step $t + 1$ with $End = Start + t + 1$), at least one array no longer satisfies p_w. This means that at step t we had one common region over a set of arrays to which we cannot add another probe. Therefore, as soon as the number of arrays in $SetArrays\_B$ becomes smaller than $SetArrays\_A$, we know we found a common region in the previous step, and we have to update the set of regions.

Line 18 is needed to allow capturing subsequent decreases (if there were any) in the number of arrays that meet the condition as we keep enlarging the region by adding probes.

We only reach line 17 if we exit the while loop (line 8). This can happen in two ways: either because we no longer fulfill $freq.arrays$ (line 15) or if there are no further probes to consider because we are at the end of the chromosome. In the first case, we know we have to add the sequence in the previous iteration (so the argument $End - 1$ is correct, as it was End which lead to failing the condition in line 13). In the second case, we have to add the sequence up to the last probe (and again $End - 1$ is the correct argument as we increased End in line 19).

Analogous to what happened in **pREC-A**, computing $P(S_{Start}, \ldots, S_{End} = 1|candidate\_array)$ (line 11) requires only an update, not computing the probability of the complete set of probes each time.

In any specific implementation, it is not necessary to explicitly do assignments as in lines 2 and 9. In our current C implementation, we use two additional variables (one for the vector that represents $SetArrays\_A$ and one for the vector that represents $SetArrays\_B$) that tell us how many valid elements there are in each set, and we only access and use up to those valid elements. Likewise, the set union operation as in lines 5 and 12 can instead be implemented as an assignment to a specific position of a vector. Similar comments apply to line 18. For instance, we could rewritte lines 4 and 5 as:

```
1 valid_elements ← 0;
2 if P(S_Start = 1|array) ≥ p_w then
3 |   valid_elements ← valid_elements + 1;
4 |   SetArrays_A[valid_elements] = array;
```

$valid\_elements$ is also the cardinality of the set. (Note that in C and other languages that index arrays starting at 0 we would increase $valid\_elements$ after the assignment to $SetArrays$).

This algorithm has no equivalent in alternative methods.

Simple numerical example: **pREC-A**

Suppose we have fit a model to six probes and four arrays and, after using RJaCGH's model averaging, we have obtained the marginal probabilities of gain shown in Table . We want to use **pREC-A** with $p_a = 0.6$. First, we average the probability for probe 1 for the four arrays:

$$P(S1 = Gain) = \frac{0.17 + 0.16 + 0.08 + 0.16}{4} = 0.14$$

As it does not reach the threshold of 0.6, S1 can not belong to a region. We do the same for S2, obtaining 0.35. For S3 the averaged probability is 0.97, so the first region will include this probe. To see if we can extend this region to the next probe, we compute for every array the joint probability of probes 3 to 4 to be gained. This probability is not shown in the table above (which shows only marginal probabilities) but is obtained as explained above (see section "Computation of the joint probability of an arbitrary sequence of probes in an array"): the relative frequency of a sequence in the MAPs from all the MCMC samples.

$$P(S3 = Gain, S4 = Gain) = \frac{0.97 + 1 + 0.07 + 0.99}{4} = 0.76$$

As it is over the threshold, we join S4 to the region.

Now we check if S5 can be joined too. We compute the joint probability of gain for the probes 3 to 5 (again, the joint probability is computed from the relative frequency of this sequence in the MAPs from all the MCMC samples):

$$P(S3 = Gain, S4 = Gain, S5 = Gain) = \frac{0.97 + 0.15 + 0.06 + 0.99}{4} = 0.54$$

As it does not reach 0.6, S5 will not be part of the region, so we get:

Region 1: {(S3, S4)}.

Now we keep on searching from probe 5. S5 does not have a marginal probability higher than the threshold, so it will not form any region. But S6 will:

$$P(S6 = Gain) = \frac{0.17 + 1.00 + 0.92 + 1.00}{4} = 0.77$$

So it will form its own region. As there are no more probes, the regions found are {(S3, S4), (S6)}.

Note that boundaries of regions need not be common over arrays: the algorithm finds the common regions. For instance, the left boundary of the first region of gain of sample A2 is located in probes S2, whereas the

boundary for all the other three samples is located in S3. Thus, S2 is excluded from the first common region: a region that spanned {(S2, S3, S4)} would not reach, over all four arrays, the required $p_a = 0.6$.

Simple numerical example: **pREC-S**

We use the same data as above. We want to find all regions where at least two arrays have a joint probability of gain of at least 0.9 (note that we raise the probability threshold because we do not ask that, on average, all arrays reach it, but at least two of them do). In other words, we are using **pREC-S** with $freq.arrays = 2$ and $p_w = 0.90$. Line numbers below refer to the lines in the algorithm.

We start on S1, but there is no array that reaches the threshold of 0.9 for that probe (i.e., the condition in line 4 is not fulfilled for any array). We iterate (line 1) to the next probe, S2, but the threshold is reached only in Array 2, and we imposed that there should be at least 2 arrays. Thus, condition in line 6 is not met. We iterate to the next probe, S3. Here, when we iterate over all the arrays (line 3) we find all of the arrays reach the threshold, so in line 5 we end up with $SetArrays\_A = (A1, A2, A3, A4)$.

As the condition in line 6 is fulfilled we try to increase the region by one probe: we set End to S4 (line 7) and enter the "while" loop (line 8) as we are not yet at the end of the total number of probes.

After looping over all four arrays (line 10) we find that line 11 is only fulfilled for Arrays 1, 2 and 4:

$$P(S3 = Gain, S4 = Gain|A1) = 0.97$$

$$P(S3 = Gain, S4 = Gain|A2) = 1.00$$

$$P(S3 = Gain, S4 = Gain|A4) = 0.99$$

$$P(S3 = Gain, S4 = Gain|A3) < 0.90$$

Note that the last expression is obvious since $P(S4 = Gain|A3) = 0.07$.

Therefore (from the iteration over line 12) we have $SetArrays\_B = (A1, A2, A4)$. We still fulfill the condition about $freq.arrays$ in line 13, but the new set of arrays contains fewer than before (line 16) which means that in the step before a region was found. We call $UpdateRegionS$ so that the region $((S3), (A1, A2, A3, A4))$ is stored, and we set $SetArrays\_A = (A1, A2, A4)$ (line 18). We increase End to S5 (line 19), and consider it as the end of the new possible region. Iterating again (line 10) we find

$$P(S3 = Gain, S4 = Gain, S5 = Gain|A1) = 0.97$$

$$P(S3 = Gain, S4 = Gain, S5 = Gain|A4) = 0.99$$

$$P(S3 = Gain, S4 = Gain, S5 = Gain|A2) < 0.90$$

As above, this means that in the previous step we found a region (line 16 is true). Therefore, we call $UpdateRegionS$ to store the region from the previous step: $((S3, S4), (A1, A2, A4))$. We increase End to S6 and find

$$P(S3 = Gain, S4 = Gain, S5 = Gain, S6 = Gain|A1) < 0.90$$
$$P(S3 = Gain, S4 = Gain, S5 = Gain, S6 = Gain|A4) = 0.99 \quad (3)$$

Now, the condition in line 13 is true, because only one array satisfies being over p_w. We break out of the while loop (line 15) and we $UpdateRegionS$ in line 20, so we store the region from the previous step: $((S3, S4, S5), (A1, A4))$.

We continue iterating over $Start$ (line 1), so now $Start = S4$. Repeating the steps above we would find a first region $((S4), (A1, A3, A4))$, and a second region $((S4, S5), (A1, A4))$. However, when executing $UpdateRegionS$, we would find each of these regions is a subset of a previously found region $(((S4), (A1, A3, A4))$ of $((S3, S4), (A1, A3, A4))$; $((S4, S5), (A1, A4))$ of $((S3, S4, S5), (A1, A4)))$.

When we iterate over $Start$ to $Start = S5$, we find only the region $((S5), (A1, A4))$ which is again a subset of a previously found region.

Finally, we set $Start = S6$. We find (lines 3 and 4) that p_w is satisfied by arrays A2, A3, A4, so we end up with $SetArrays\_A = (A2, A3, A4)$. We fullfill the requirement about $freq.arrays$, but in line 8, however, we find we are at the end of the total number of probes, so we do not enter that loop (lines 9 to 19 are skipped). We therefore call $UpdateRegions$, and add the region $((S6), (A2, A3, A4))$. (Note that the call to $UpdateRegions$ in line 20 with $End - 1$ is correct, since we increased End one position over S6 in line 7).

Therefore, we end up with the regions:

Regions = {((S3), (A1, A2, A3, A4)), ((S3, S4), (A1, A2, A4)), ((S3, S4, S5), (A1, A4)), ((S6), (A2, A3, A4))}

We can see the regions obtained in Figure . In contrast to **pREC-A**, boundaries need not be common over arrays; with **pREC-S** differences in boundaries will lead to different subsets and different regions (for instance, that is why the common region (S3, S4) includes only samples A1, A2, A4, but not A3).

We can also use the output of this algorithm as the basis for clustering and to display patterns of groupings of arrays. We can measure similarity between two arrays as the number of common probes in CNVCRs between those two arrays or, alternatively, as the number of common regions (where the same probe might belong to more than one region) between two arrays. Once similarity is measured, we can

immediately apply any clustering method of our choice. An example is show in Figure . At this stage, clustering is mainly a device for representing patterns of similarity, since the grouping of arrays with respect to recurrent CNVs is the very output of the **pREC-S** algorithm.

Implementation and testing

The algorithms above are part of the freely available and open-source RJaCGH R package (available from the R repositories), which uses R and C (the later, dynamically loaded from within R). For storage and efficiency reasons, we do not save directly all of the Viterbi paths (i.e., each Viterbi from each iteration of the MCMC sampler) but only the jumps in paths and the counts of different paths. This requires less storage, allows for faster access to the information and computation of the joint sequence, and of course permits reconstructing all of the sequences. The Viterbi paths are obtained as part of the regular execution of the C code for RJaCGH, saved in R as gzipped files, and read back by the C functions for **pREC-A** and **pREC-S** only once.

Execution time in all the examples of the paper is negligible: all the examples of pREC-A execute in less than 5 seconds. Execution time for pREC-S goes up to 160 seconds for the examples from [31] but less than 4 seconds for the remaining examples. (All these timings from a workstation with and AMD 280 processor running Debian GNU/Linux).

Testing was carried out by comparing the output from the algorithms with manually computed examples. Code for the examples and comparisons is included in the repository for the package https://launchpad.net/rjacgh/main.

Examples with real data and comparison to other approaches

All the examples below were analysed with RJaCGH, thus providing the probabilities of alteration. We focus in the examples not on the RJaCGH results per se, but rather on the common regions detected. Our examples use arrays of BAC because these are four "classic" sets of data that have been analyzed before with other approaches. Our methods, however, can also be applied to other platforms, including custom and commercial oligonucleotide arrays and SNP arrays.

Colorectal cancer example (Nakao et al.)

Nakao et al. [32] analyze 125 colorectal tumors. They apply a segmentation method based on a threshold and then find common regions of alteration studying the frequency of alterations. Rouveirol et al. [18]

apply both of their algorithms for minimal common regions to the same data.

We have applied our first method, pREC-A, and found, with a probability of at least 0.35, basically the same regions of alteration. As we can see in Table and the frequency plot of alterations (Figure), most of the reported differences come from regions with a probability (or frequency, in the case of [32]) in the limit of 35% (the same threshold that [32] uses). The only remarkable case is the gain in 11q which has a much lower probability in our analysis, probably because that alteration is based on a single BAC and the segmentation analysis used in [32] is based on a threshold and therefore is more likely to be affected by outliers. We can obtain more detail by focusing on regions at least 0.5 probability, as shown in Table . The results are also similar to [18], but they only provide a small excerpt in their paper, so direct comparisons are difficult to make.

Breast cancer example (Pollack et al.)

Pollack et al. [31] analyze data from 44 breast tumors and 10 cancer cell lines. They search for common regions of alteration and then compare the frequency of aberrations in each arm of every chromosome as a function of other variables such as tumor grade, estrogen receptor (ER) and TP53 mutations. Rouveirol et al. [18] also analyze these data, though they restrict their study to 37 tumors and only give brief details about the regions obtained. We have applied our second method, pREC-S, to the 44 tumors to examine if there is any similarity in the alterations shared by the groups of arrays defined by those variables. We have computed common regions of at least 0.50 probability of alteration (Gains or Losses) shared by at least two arrays (i.e., $freq.array = 2$, $p_w = 0.50$).

To compare our approach with the results of [31], and to gain more insight on the patterns of CNVCR and their relationship to the other three variables (tumor grade, ER, TP53), we have defined a simple statistic to measure within-group CNVCR homogeneity. Let Y_{ij} be the number of probes that array i and array j have altered in common, k a group of arrays (typically, with some common characteristic), n_k the number of different pairs of arrays in a given group k and n_k the number of different pairs formed by arrays in group k and arrays in a different group. Let us define

$$\bar{Y}_k = \sum_{i,j \in k} \frac{Y_{ij}}{n_k}$$

$$\bar{Y}_{-k} = \sum_{i \notin k, j \in k} \frac{Y_{ij}}{n_{-k}}$$

That is, $\bar{Y}_k$ is the average number of common altered probes between two arrays of group k, and $\bar{Y}_{-k}$ is the

average number of common altered probes between one array of group k and other in a different group. We define the proportion of common alterations shared by the group k as $\bar{Y}_k/\bar{Y}_{-k}$. This index measures the homogeneity of the genomic alterations within a subset of arrays compared to the alterations shared with arrays of other group. If this index is greater than 1, the arrays of this group share more alterations between themselves than arrays of different groups do. If this index is 0, no alterations are shared between any two arrays in the group. A value of ∞ means that no alteration is shared between arrays of this group and others. We can compute this index for the groups defined by the three variables tumor grade, ER, and TP53 mutations.

In Table we see that the gains in chromosomes 4 and 5 and the losses in chromosome 8 are very homogeneous in the estrogen receptor negative samples. Table shows that the gains in chromosomes 2 and 10 and the losses in chromosomes 17 and 21 are more homogeneous in tumors harboring TP53 mutations. Finally, Table also shows differences in the pattern of homogeneity of alterations with respect to the grade of the tumor. We can also show in a figure the similarity within and between groups by plotting the number of common alterations. Figure shows the gains in chromosome 8.

These results are not easy to compare with [31], because they define the regions and compare subgroups at chromosome arm resolution, while our method works at BAC resolution. Furthermore, they consider every chromosome arm as altered or not without taking into account the number of altered probes in it. Instead of only comparing subgroups according to the number of alterations, we can try to examine the data further by analyzing how homogeneous each group is over the whole genome (not chromosome by chromosome, as in previous tables). This is shown in Table . When we divide arrays according to tumor grade, Grade I and Grade III show high homogeneity within groups, meaning that the alterations are consistent in arrays within those grades. Arrays of grade II, however, show much more heterogeneity, sharing many aberrations with arrays of Grade I and/or Grade III. This is an indication that arrays of Grade II can be classified in one of the other two groups according to the pattern of alterations. In Figure we see an example: four arrays of Grade II are very similar to the arrays of Grade III.

Colorectal cancer example (Douglas et al.)

Our algorithms to detect common regions may also be used to compare directly the probability of alteration between groups of samples. Douglas et al. [33] present data from 37 primary cancers. Seven show microsatellity instability (MSI) and 30 show chromosomal instabillity (CIN). (For a a definition of genetic alterations, see [34]). They call alterations using a threshold-based method and compare their frequency

between the two types using a chi-square statistic. van de Wiel and van Wieringen [21] analize the same data using a dimension reduction technique (CGHRegions) over the alterations detected by DNACopy [35]. They then use a Wilcoxon test with FDR correction for the difference between the two levels.

We can first use a moderate threshold, such as 0.35, to find common regions of alteration over arrays. Thus, we are using pREC-A with $p_a = 0.35$. As expected for this kind of genetic alteration [33] there are many more gains and losses in CIN cancers than in MSI ones and, in particular, there are very few common losses in MSI samples. When comparing the type of alterations between both groups, there are some regions of loss in CIN that correspond to gains in MSI. The main difference lies in chromosome 8, wholly gained in the MSI group with a joint probability of 0.3993, while in the CIN there is a region lost of 50 clones (also reported in [33]). These patterns are also seen in chromosome 4 and chromosome 21, but just in a few probes.

To circumvent this contradiction, we used a higher threshold of $p_a = 0.50$ to find the common regions of gain/loss and then compared the probability of alteration in those regions for the two groups of samples. We obtained a total of 21 regions of gain and 11 of loss. Due to the unequal size of the groups (30 vs. 7), we could have bias in detecting alterations that occur only in the smallest group. An easy solution would be to apply different weights to the groups when computing the probabilities. The lack of common regions in the MSI group makes this unnecessary. In Figure we can see the common regions with at least 0.50 probability of alteration and the joint probability for those regions for the two groups. The striking differences between MSI and CIN in chromosome 8 have disappeared, because the gains in the MSI group do not reach the threshold of 0.50.

Next, for every region (of the 21 regions of gain and 11 regions of loss found above) we computed the joint probability of alteration for each of the 30 arrays of class CIN and the seven arrays of class MSI and, by region, we calculated the absolute value of the difference in mean probability between the MSI and CIN groups. To assess the significance of this statistic, we used a permutation test to obtain a two-sided p-value. Finally, we applied the FDR method [36] for multiple testing correction (to account for the multiple testing arising from comparing multiple regions). The regions found significantly different (at 0.05 level) between groups are listed in table : [33] report differences between both groups in gain of chromosome 20, loss of 18q and the short arm of chromosome 17 and loss of 8p. Our regions do not include the complete chromosome 20 because the p arm is gained with probability less than 0.5. We found also a difference in all of chromosome 18, but [33] report some losses in certain clones in the MSI group that we have not found. The rest of the regions we found are reported in [33] as common regions of alteration but

with no difference. The later could be related to the higher precision that our method gives, but [33] do not provide details about frequency of those regions.

Comparing these results to [21], we find some interesting differences. First, almost all of the regions of [21] are discovered with our method, but their length or the location of the breakpoints sometimes differ. This difference is probably related to the method of [21], CGHregions: CGHregions is a dimension reduction method, and thus the complexity of the sample profiles is simplified. Second, two regions in their paper, a small loss region of only two clones in the 8th chromosome and a big region of 29 clones in chromosome 18th, are not detected by our method because the probability of loss of those regions is just below 0.50 of probability. Of course, our method allows to adjust the threshold at whichever value is considered reasonable, and to check how conclusions change with changes in the threshold (as we have shown in this example). Finally, there are other regions detected by our method that show signficant differences between the two groups and are not reported in [21], such as losses in chromosome 17 (detected in [33]) and gains in chromosome 7.

Discussion

We have developed two very different approaches for finding recurrent, or common, copy number variation regions (CNVCR). The lack of gold standards and the current non-existence of an unambiguous definition of what a recurrent CNV is [2], and the unique and qualitatively different nature of our approaches from previous ones, make it difficult to compare the performance of our methods to previous approaches, but at the same time highlight the relevance of our methods for current and future studies of CNV, their relation to phenotypic variation, and their usage for subject clustering.

The two methods we have developed share that they use as input probabilities of alteration and return probabilities. Regardless of whether the input probabilities are obtained from our RJaCGH method [15] or some other approach, it can be argued that probabilities are much better suited to the task at hand than p-values or discrete classifications into “gained”, “lost”, “not changed”. By using probabilities as input, we incorporate uncertainty in the estimates of copy number estates. By returning probabilites and using probabilities throughout all the analysis, the user can decide the appropriate thresholds (or, even, modify them depending on context) and define distances between arrays that incorporate the strength of evidence in favor of alteration. Precisely because of the conceptual simplicity of using probabilities, we can approach within a unified framework both questions related to “unsupervised problems” (e.g., identify subsets of regions that are common to subsets of arrays) and to “supervised problems” (e.g., measure how different

two groups of arrays with respect to recurrent regions of alteration). This unified approach is unique to our methods, and not shared by any others.

Our first method, **pREC-A**, searches for general, broad patterns of common gains (or losses) over all the samples in the study. This is the approach which is most similar to previous ones. This method is well suited to comparing pre-defined groups of samples. By its very nature (e.g., that an overall pattern is identified by a mean probability larger than a threshold) this method can only detect regions for which there is at least moderate evidence (medium probability of alteration) over almost all samples, or very strong evidence (high probability of alteration) over an important fraction of the samples. Thus, it is easy to miss regions that are present with very high probability in a small subset of the samples. As well, mixing in the same sample very heterogenous groups will tend to smooth out the evidence of alteration, so that few common regions will be found. Alternatively, if there are very different sample sizes (different number of arrays) in the different heterogenous groups, the detected common regions will often be a subset of the common regions among the most abundant group. These features can be controlled to answer the specific study questions. First, as equation 1 (see "Methods") shows, it is easy to weight different arrays differently, so as to increase the influence of some arrays in the final analysis. Moreover, if we know in advance that there are different subgroups of samples, we can use **pREC-A** independently in the different subgroups; for instance, when we have already subdivided the subjects in the study into homogeneous groups with respect to disease (e.g., [37]), and want to locate CNVCRs common to most samples within a subgroup and possibly different from other subgroups. Finally, as our last example with the data of [33] shows, a user that understands these features of **pREC-A** can employ this algorithm to highlight the differences between subgroups and how these change as we modify the minimum required threshold for the probability of alteration. In particular, note the easy formulation of a permutation-based test for identifying the differences in the probabilities of alteration of regions between subgroups. This type of approach might be even more useful when two or more suspected subgroups are compared against a larger, reference group. The main advantages of this algorithm are that it is most similar to previous approaches, has a simple interpretation in terms of global patterns across most of the samples, and requires the specification of only one parameter. Thus, **pREC-A** will often be the method of choice if we are trying to relate major, global, recurrent patterns of CNV to variations in phenotype or to differentiate between subroups of samples.

In contrast to **pREC-A**, the second method, **pREC-S**, can detect small subgroups of samples with respect to common alterations, without being adversely affected by averages over arrays or differences in number of samples in different subgroups. Moreover, different subgroups can be detected with respect to

different alterations; for instance, out of a set of arrays A1, A2, ..., A10, arrays A1, A2, A3, might be grouped with respect to a common alteration in region R1, and arrays A1, A3, and A4 with respect to a common alteration in region R2. In this sense, **pREC-S** resembles biclustering [38], with the advantage that in **pREC-S** the objectives and criteria are well defined as "common region over a set of arrays" is unambiguously defined once the two parameters of the algorithm are chosen. In those disorders that involve a potentially large set of alterations that lead to the same disease (e.g., [3, 26]), identifying subgroups of patients with respect to common alterations is of key relevance to pinpoint the possible genetic basis of disease. The only other method that tries to find such subsets of samples (or regions over subsets) is the one by [19], but this is a clustering method, and thus the finding of common regions ("markers" in their terminology) is only a step for the final objective which is clustering samples, not a search conducted exhaustively for its own sake. Therefore, **pREC-S** is a qualitatively different algorithm from available ones, and it addresses a common and distinct need that arises in any study of CNV with heterogeneous samples. As seen in the results, the usage of this second algorithm allows us to elegantly approach some of the questions in the second example (breast cancer example, [31]). First, the derivation of a specially tailored statistic, $\bar{Y}_k/\bar{Y}_b$, to answer the relevant questions in this study is straightforward. More importantly, the second algorithm finds homogeneous subgroups, with respect to alterations, and these differences are associated with differences in three other markers (estrogen receptor status, TP53 mutation, tumor grade; see Tables , , ,). In other words, **pREC-S** finds CNV that differentiate between groups. It must be emphasized that **pREC-S** has been applied to the complete set of data after specifying that the within-array probability of alteration be larger than 0.5 (i.e., $p_w = 0.50$) and that these regions be shared among, at least, two arrays (i.e., $freq.array = 2$), but the algorithm is blind to the "labels" of the arrays regarding the other markers (estrogen receptor, TP53, grade). Therefore, **pREC-S** allows to find CNV that differentiate between known groups (as in this case), but its systematic usage also opens the door to finding patterns of CNV that might differentiate between previously unknown groups. Moreover, there is no need for the association CNVCRs-marker to be similar among different markers, specially since, as explained above, different subgroups of arrays can be detected with respect to different CNV recurrence patterns. These are features unique and characteristic of **pREC-S**, compared to all the alternative available methods.

We suggest that **pREC-S** is the method of choice when there is unknown heterogeneity among arrays in CNV, and when we want to relate possibly non-identical subsets of samples, defined in terms of recurrent patterns of CNV, to phenotypic variation. Moreover, routine use of **pREC-S** even with apparently

homogeneous groups of samples might help discover possible subtypes of diseases that might generate novel hypothesis or uncover previously unknown heterogeneities.

pREC-S is also a key method for clustering. Integrative studies that combine CNV data with other data (e.g., mRNA, SNP) often use clustering of subjects based upon the CNV data (e.g., [39,40]). The problem of most of these approaches is that, when clustering based upon the CNV data (either the gain/loss calls or the smoothed data), the measure of distance or similarity used ignores that some of the data show strong serial dependence (probes next to each other) whereas some of the data (e.g., probes in different chromosomes) are independent. Thus, in most cases the distance computed is likely to introduce serious distortions in the true distances among subjects (see also [19,41]). This problem is in addition to the aforementioned issues of not integrating variability and uncertainty in the gain/loss calls or smoothed means. In contrast, by using a biologically motivated and probabilistically based approach to CNV common regions, such as **pREC-S**, it will be possible to construct distance metrics and, therefore, clustering approaches, that make full usage of CNV data when searching for groups of subjects. Fully developing a method for clustering based upon CNV data is outside the scope of this paper, but we have presented a simple example to motivate further work.

Moreover, an additional distinct feature of our methods is that both **pREC-S** and **pREC-A** have at most two parameters of straightforward biological interpretation (probability of alteration, number of samples that share the alteration). An added advantage of the type of input and output used by our methods is that probabilities allow researchers to modify thresholds as needed, and to easily (and intelligibly) examine the sensitivity of results to changes in thresholds.

Finally, as both methods are based on a Hidden Markov Model (HMM) with no restrictions on the number of states [15], it is also immediate to restrict finding CNVCR to alterations above a certain threshold of amplitude or magnitude of change. The HMM (probabilistically) assigns probes to hidden states, but it is up to subsequent analysis to later assign those states to specific or interesting "copy number states". Thus, we keep the two different concepts of "amplitude (or magnitude) of change" and "evidence of alteration" separate. Therefore, we allow filtering and customized analysis that can focus only on alterations of a certain type.

Conclusion

We have developed methods for finding regions of copy number variation (CNV) common to several arrays. Our methods have an immediate and intuitive biological interpretation, and incorporate both within- and

among-array variability. Reanalysis of several data sets in the literature show that our methods can indeed recover patterns previously found but can also uncover additional patterns. Moreover, probabilities allow researchers to modify thresholds as needed, and to easily examine the sensitivity of results to changes in thresholds. In addition, the examples show how it is straightforward to derive tailored statistics and summary measures to answer specific research questions. The development of these two distinct algorithms highlights a key idea that has often been neglected: recurrent or common CNVs can refer to very distinct patterns in a group of samples, specially concerning heterogeneity among arrays and probability of alteration. We expect that these two algorithms will help advance efforts to standardize definitions of recurrent or common CNV regions, and ultimately the search for genomic regions harboring disease-critical genes.

Authors contributions

Oscar M. Rueda developed the statistical model, did most of the programming and conducted all of the analysis. Ramon Diaz-Uriarte conceived the original HMM model and participated in model development and programming. Both authors wrote the paper.

Acknowledgments

C. Lázaro-Perea for comments on the ms.

References

1. Lee C, Iafrate AJ, Brothman AR: **Copy number variations and clinical cytogenetic diagnosis of constitutional disorders**. *Nature Genetics* 2007, **39**:S48–S54, [http://www.nature.com/ng/journal/v39/n7s/full/ng2092.html].
2. Scherer SW, Lee C, Birney E, Altshuler DM, Eichler EE, Carter NP, Hurles ME, Feuk L: **Challenges and standards in integrating surveys of structural variation.** *Nat Genet* 2007, **39**(7 Suppl), [http://dx.doi.org/10.1038/ng2093].
3. Sebat J: **Major changes in our DNA lead to major changes in our thinking**. *Nature Genetics* 2007, **39**:S3–S5, [http://www.nature.com/ng/journal/v39/n7s/full/ng2095.html].
4. Beckmann JS, Estivill X, Antonarakis SE: **Copy number variants and genetic traits: closer to the resolution of phenotypic to genotypic variability**. *Nat Rev Genet* 2007, **8**(8):639–646, [http://view.ncbi.nlm.nih.gov/pubmed/17637735].
5. Redon R, Ishikawa S, Fitch KR, Feuk L, Perry GH, Andrews DT, Fiegler H, Shapero MH, Carson AR, Chen W, Cho EK, Dallaire S, Freeman JL, Gonzalez JR, Gratacos M, Huang J, Kalaitzopoulos D, Komura D, Macdonald JR, Marshall CR, Mei R, Montgomery L, Nishimura K, Okamura K, Shen F, Somerville MrJ, Tchinda J, Valsesia A, Woodwark C, Yang F, Zhang J, Zerjal T, Zhang Ja, Armengol L, Conrad DF, Estivill X, Tyler-Smith C, Carter NP, Aburatani C Hiroyuki an d Lee, Jones KW, Scherer SW, Hurles ME: **Global variation in copy number in the human genome**. *Nature* 2006, **444**(7118):444–454.
6. Lupski JR: **Genomic rearrangements and sporadic disease**. *Nature Genetics* 2007, **39**:S43–S47, [http://www.nature.com/ng/journal/v39/n7s/full/ng2084.html].

7. McCarroll SA, Altshuler DM: **Copy-number variation and association studies of human disease.** *Nat Genet* 2007, **39**(7 Suppl):S37–S42, [http://dx.doi.org/10.1038/ng2080].

8. Ylstra B, van den Ijssel P, Carvalho B, Brakenhoff RH, Meijer GA: **BAC to the future! or oligonucleotides: a perspective for micro array comparative genomic hybridization (array CGH).** *Nucleic Acids Res* 2006, **34**:445–450.

9. Pinkel D, Albertson D: **Array comparative genomic hybridization and its application in cancer**. *Nature Genetics* 2005, **37**(Supplement):S11–S17.

10. Huang J, Wei W, Chen J, Zhang J, Liu G, Di X, Mei R, Ishikawa S, Aburatani H, Jones KW, Shapero MH: **CARAT: a novel method for allelic detection of DNA copy number changes using high density oligonucleotide arrays.** *BMC Bioinformatics* 2006, **7**, [http://dx.doi.org/10.1186/1471-2105-7-83].

11. Carter NP: **Methods and strategies for analyzing copy number variation using DNA microarrays.** *Nat Genet* 2007, **39**(7 Suppl):S16–S21, [http://dx.doi.org/10.1038/ng2028].

12. Diskin S, Eck T, Greshock J, Mosse Y, Naylor T, Stoeckert CJ, Weber B, Maris J, Grant G: **STAC: A method for testing the significance of DNA copy number aberrations across multiple array-CGH experiments.** *Genome Res.* 2006, **16**(9):1149–1158.

13. Tonon G, Wong KK, Maulik G, Brennan C, Feng B, Zhang Y, Khatry DB, Protopopov A, Ỹou MJ, Aguirre AJ, Martin ES, Yang Z, Ji H, Chin L, Depinho RA: **High-resolution genomic profiles of human lung cancer.** *Proc Natl Acad Sci U S A* 2005, **102**:9625–9630.

14. Misra A, Pellarin M, Nigro J, Smirnov I, Moore D, Lamborn KR, Pinkel D, Albertson DG, Feuerstein BG: **Array comparative genomic hybridization identifies genetic subgroups in grade 4 human astrocytoma.** *Clin Cancer Res* 2005, **11**:2907–2918.

15. Rueda OM, Diaz-Uriarte R: **Flexible and accurate detection of genomic copy-number changes from aCGH.** *PLoS Comput Biol.* 2007, **3**(6):1115–1122.

16. Rueda OM, Diaz-Uriarte R: **A response to Yu et al. 'A forward-backward fragment assembling algorithm for the identification of genomic amplification and deletion breakpoints using high-density single nucleotide polymorphism (SNP) array', BMC Bioinformatics 2007, 8: 145**. *BMC Bioinformatics* 2007, **8**:394+, [http://view.ncbi.nlm.nih.gov/pubmed/17939873].

17. Aguirre AJ, Brennan C, Bailey G, Sinha R, Feng B, Leo C, Zhang Y, Zhang J, Gans JD, Bardeesy N, Cauwels C, Cordon-Cardo C, Redston MS, Depinho RA, Chin L: **High-resolution characterization of the pancreatic adenocarcinoma genome**. *Proc Natl Acad Sci U S A* 2004, **101**:9067–9072.

18. Rouveirol C, Stransky N, Hupé P, La Rosa P, Viara E, Barillot E, Radvanyi F: **Computation of recurrent minimal genomic alterations from array-CGH data.** *Bioinformatics* 2006, **22**:2066–2073.

19. Liu J, Ranka S, Kahveci T: **Markers improve clustering of CGH data**. *Bioinformatics* 2007, **23**(4):450–457.

20. Ben-Dor A, Lipson D, Tsalenko A, Reimers M, Baumbusch L, Barrett M, Weinstein J, Borresen-Dale A, Yakhini Z: **Framework for Identifying Common Aberrations in DNA Copy Number Data**. *Proceedings of RECOMB '07* 2007, **4453**:122–136.

21. van de Wiel MA, van Wieringen W: **CGHregions: Dimension reduction for array CGh data with minimal information loss**. *Cancer Informatics* 2007, **2**:55–63.

22. Weir B, Woo M, Getz G, Perner S, Ding L, Beroukhim R, Lin W, Province M, Kraja A, Johnson L, Shah K, Sato M, Thomas R, Barletta J, Borecki I, Broderick S, Chang A, Chiang D, Chirieac L, Cho J, Fujii Y, Gazdar A, Giordano T, Greulich H, Hanna M, Johnson B, Kris M, Lash A, Lin L, Lindeman N, Mardis E, Mcpherson J, Minna J, Morgan M, Nadel M, Orringer M, Osborne J, Ozenberger B, Ramos A, Robinson J, Roth J, Rusch V, Sasaki H, Shepherd F, Sougnez C, Spitz M, Tsao MS, Twomey D, Verhaak R, Weinstock G, Wheeler D, Winckler W, Yoshizawa A, Yu S, Zakowski M, Zhang Q, Beer D, Wistuba I, Watson M, Garraway L, Ladanyi M, Travis W, Pao W, Rubin M, Gabriel S, Gibbs R, Varmus H, Wilson R, Lander E, Meyerson M: **Characterizing the cancer genome in lung adenocarcinoma**. *Nature* 2007, **450**:893–898, [http://www.nature.com/nature/journal/vaop/ncurrent/full/nature06358.html].

23. Lipson D, Aumann Y, Ben-Dor A, Linial N, Yakhinim Z: **Efficient calculation of interval scores for DNA copy number data analysis**. *J Comput Biol.* 2006, **13**(2):215–228.

24. Shah S, Lam W, Ng R, Murphy K: **Modeling recurrrent CNA copy number alterations in array CGH data**. *Bioinformatics* 2007, **23**(13):i450–i458.

25. Guttman M, Mies C, Dudycz-Sulicz K, Diskin SJ, Baldwin DA, Stoeckert CJ, Grant GR: **Assessing the Significance of Conserved Genomic Aberrations Using High Resolution Genomic Microarrays**. *PLoS Genetics* 2007, **3**(8):e143+, [http://view.ncbi.nlm.nih.gov/pubmed/17722985].

26. Wood LDD, Parsons DWW, Jones S, Lin J, Sjöblom T, Leary RJJ, Shen D, Boca SMM, Barber T, Ptak J, Silliman N, Szabo S, Dezso Z, Ustyanksky V, Nikolskaya T, Nikolsky Y, Karchin R, Wilson PAA, Kaminker JSS, Zhang Z, Croshaw R, Willis J, Dawson D, Shipitsin M, Willson JKVK, Sukumar S, Polyak K, Park BHH, Pethiyagoda CLL, Pant PVKV, Ballinger DGG, Sparks ABB, Hartigan J, Smith DRR, Suh E, Papadopoulos N, Buckhaults P, Markowitz SDD, Parmigiani G, Kinzler KWW, Velculescu VEE, Vogelstein B: **The Genomic Landscapes of Human Breast and Colorectal Cancers.** *Science* 2007, **318**:1108–1113, [http://www.sciencemag.org/cgi/content/abstract/318/5853/1108].

27. Cappé O, Moulines E, Ryden T: *Inference in Hidden Markov Models (Springer Series in Statistics)*. Springer 2005.

28. Rabiner LR: **A tutorial on hidden Markov models and selected applications in speech recognition**. *Proceedings of the IEEE* 1990, **77**:257–286.

29. Scott S: **Bayesian methods for hidden Markov models: Recursive computing in the 21st century**. *JASA* 2002, **97**:337–351.

30. Bilmes J: **What HMMs can do**. *IEICE Trans Inf & Syst* 2006, **E89-D**(3):869–891.

31. Pollack J, Sorlie T, Perou C, Rees C, Jeffrey S, Lonning P, Tibshirani R, Botstein D, Borresen-Dale A, Brown P: **Microarray analysis reveals a major direct role of DNA copy number alteration in the transcriptional program of human breast tumors.** *Proc Natl Acad Sci U S A.* 2002, **99**(20):12963, 12968.

32. Nakao K, Mehta K, Fridlyand J, Moore D, Jain A, Lafuente A, Wiencke J, Terdiman J, Waldman F: **High-resolution analysis of DNA copy number alterations in colorectal cancer by array-based comparative genomic hybridization.** *Carcinogenesis* 2004, **25**(8):1345–1357.

33. Douglas E, Fiegler H, Rowan A, Halford S, Bicknell D, Bodmer W, Tomlinson I, Carter N: **Array comparative genomic hybridization analysis of colorectal cancer cell lines and primary carcinomas.** *Cancer Res.* 2004, **64**(14):4817–4825.

34. Lengauer C, Kinzler K, Vogelstein B: **Genetic instabilities in human cancers.** *Nature* 1998, **396**:643–649.

35. Olshen AB, Venkatraman ES, Lucito R, Wigler M: **Circular binary segmentation for the analysis of array-based DNA copy number data.** *Biostatistics* 2004, **5**:557–572.

36. Benjamini Y, Hochberg Y: **Controlling the false discovery rate: a practical and powerful approach to multiple testing**. *J. Roy. Statist. Soc. Ser. B* 1995, **57**:289–300.

37. Kim JH, Dhanasekaran SM, Mehra R, Tomlins SA, Gu W, Yu J, Kumar-Sinha C, Cao X, Dash A, Wang L, Ghosh D, Shedden K, Montie JE, Rubin MA, Pienta KJ, Shah RB, Chinnaiyan AM: **Integrative analysis of genomic aberrations associated with prostate cancer progression.** *Cancer Res* 2007, **67**(17):8229–8239, [http://cancerres.aacrjournals.org/cgi/content/abstract/67/17/8229].

38. Prelic A, Bleuler S, Zimmermann P, Wille A, Bühlmann P, Gruissem W, Hennig L, Thiele L, Zitzler E: **A systematic comparison and evaluation of biclustering methods for gene expression data.** *Bioinformatics* 2006, **22**(9):1122–1129, [http://view.ncbi.nlm.nih.gov/pubmed/16500941].

39. Chin SF, Teschendorff AE, Marioni JC, Wang Y, Barbosa-Morais NL, Thorne NP, Costa JL, Pinder SE, van de Wiel MA, Green AR, Ellis IO, Porter PL, Tavare S, Brenton JD, Ylstra B, Caldas C: **High-resolution array-CGH and expression profiling identifies a novel genomic subtype of ER negative breast cancer**. *Genome Biology* 2007, **8**:R215+, [http://view.ncbi.nlm.nih.gov/pubmed/17925008].

40. Garraway LA, Widlund HR, Rubin MA, Getz G, Berger AJ, Ramaswamy S, Beroukhim R, Milner DA, Granter SR, Du J, Lee C, Wagner SN, Li C, Golub TR, Rimm DL, Meyerson ML, Fisher DE, Sellers WR: **Integrative genomic analyses identify MITF as a lineage survival oncogene amplified in malignant melanoma**. *Nature* 2005, **436**(7047):117–122, [http://dx.doi.org/10.1038/nature03664].

41. Liu J, Mohammed J, Carter J, Ranka S, Kahveci T, Baudis M: **Distance-based clustering of CGH data.** *Bioinformatics* 2006, **22**(16):1971–1978, [http://view.ncbi.nlm.nih.gov/pubmed/16705014].

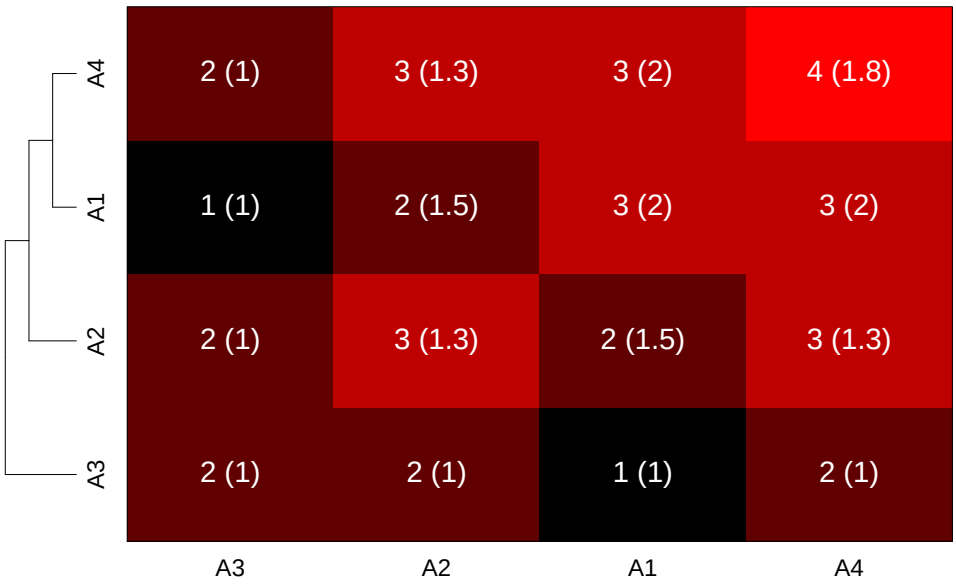

Figure 1 - pREC-S, simple numerical example

Subsets of at least 2 arrays that share common regions of gain of at least 0.90 probability: $freq.arrays = 2, p_w = 0.90$. Boxes of the same color represent the same region. In circles, the marginal probabilities of gain. In boxes, the joint probabilities.

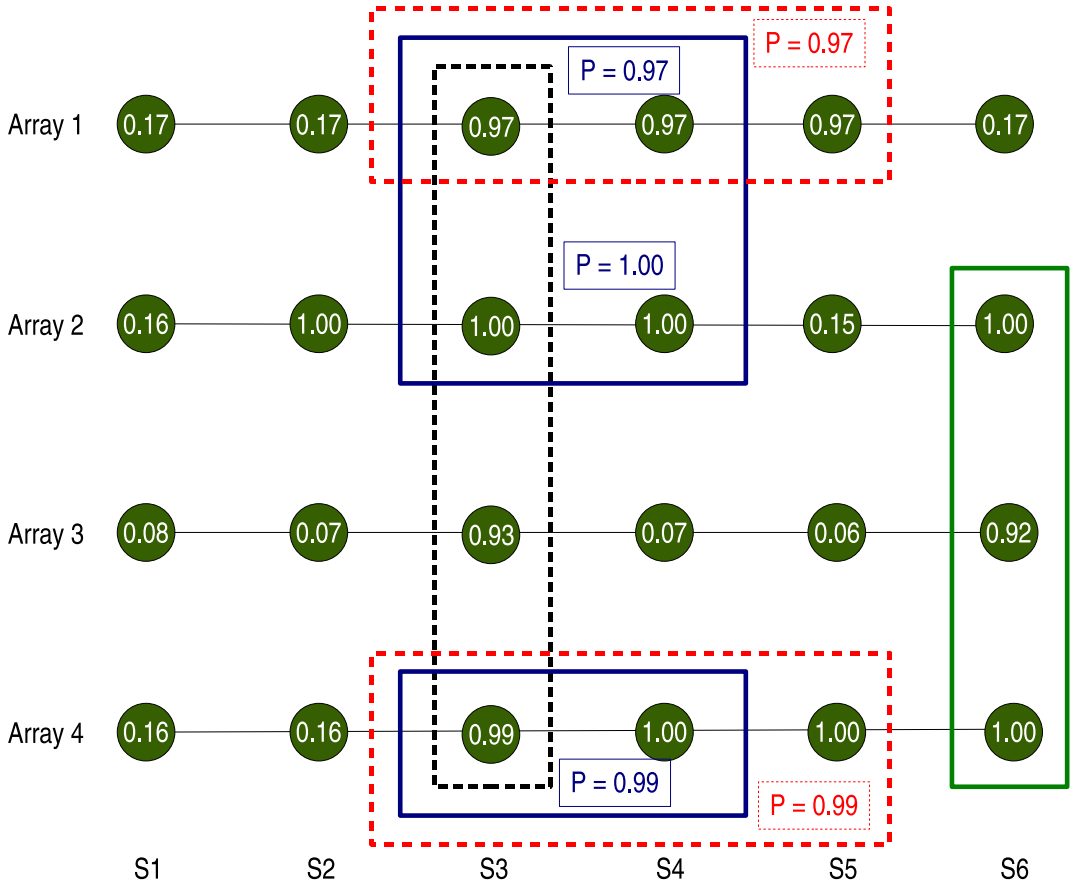

Figure 2 - Clustering based upon pREC-S

Number of common regions shared by pairs of arrays. In parenthesis, the average length in probes of the regions. On the left, a dendrogram using hierarchical clustering (complete linkage) with number of common regions shared by pairs of arrays as similarity measure.

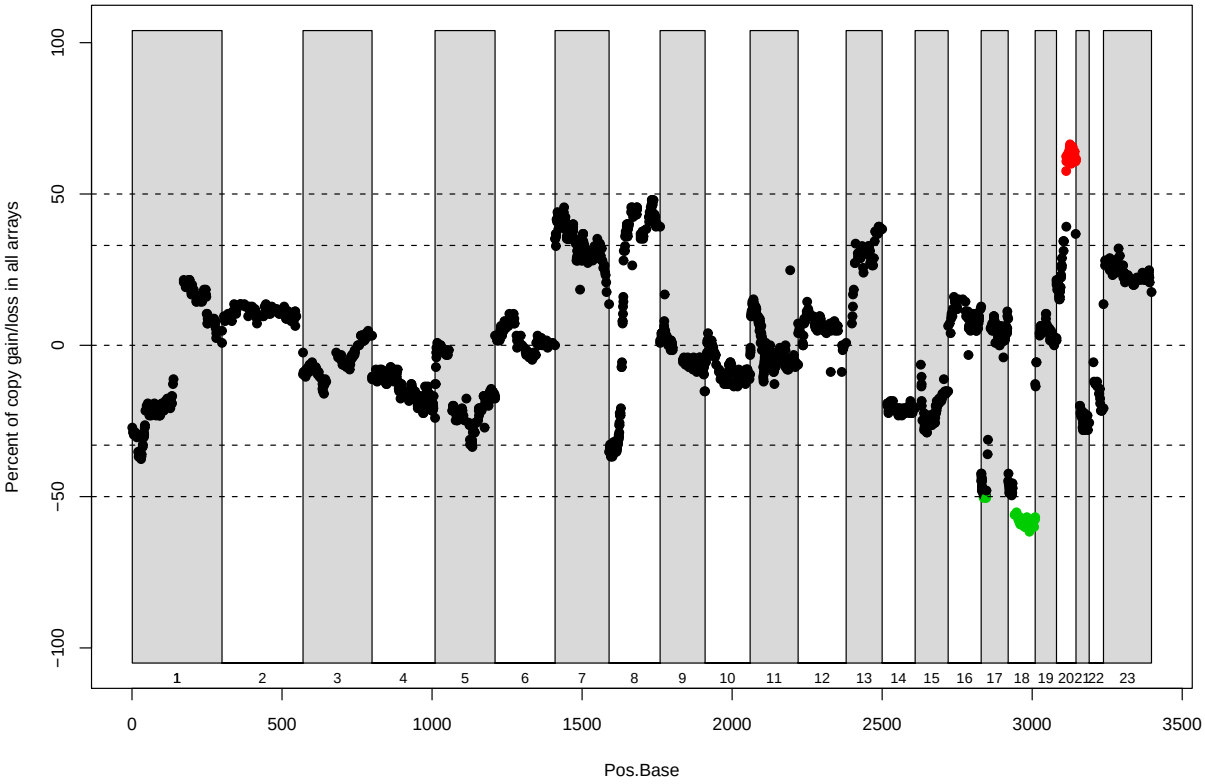

Figure 3 - Frequency plot of the alterations in 125 colorectal tumor samples in Nakao et al.

The red dots show gains found in more than 50% of the samples, and the green dots losses in more than 50%. The dotted lines show the 33% and the 50% frequency.

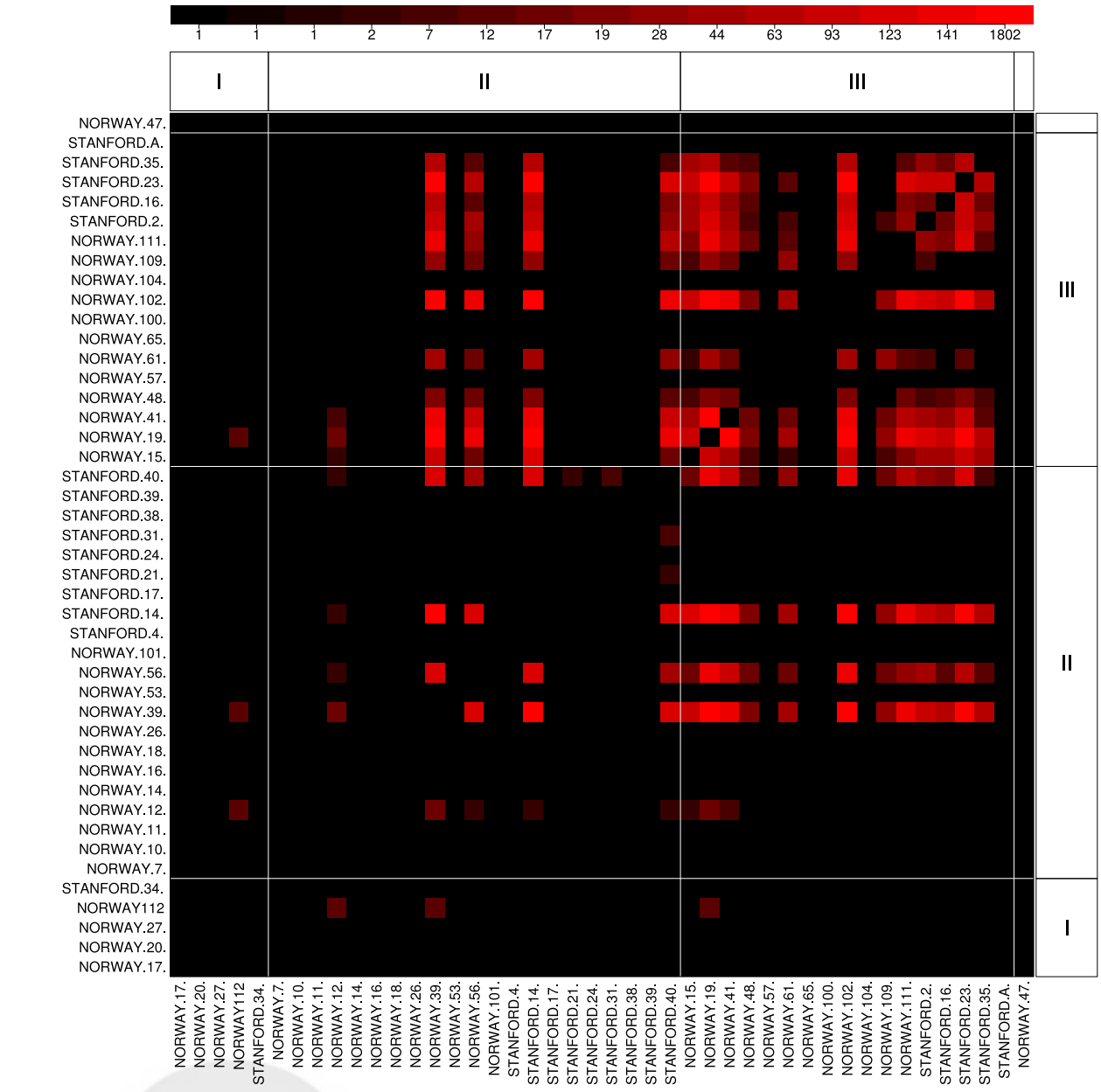

Figure 4 - Chromosome 8 from the Pollack et al. example

Number of regions of gain with at least 0.50 probability shared by at least two arrays (i.e., pREC-S, $freq.arrays = 2, p_w = 0.50$). The arrays are ordered according to tumor grade. Arrays with grade III share many more alterations between them than the other arrays. Four arrays with grade II share the same gains in copy number with tumors of higher grade, so they are probably related. There is one array unidentified.

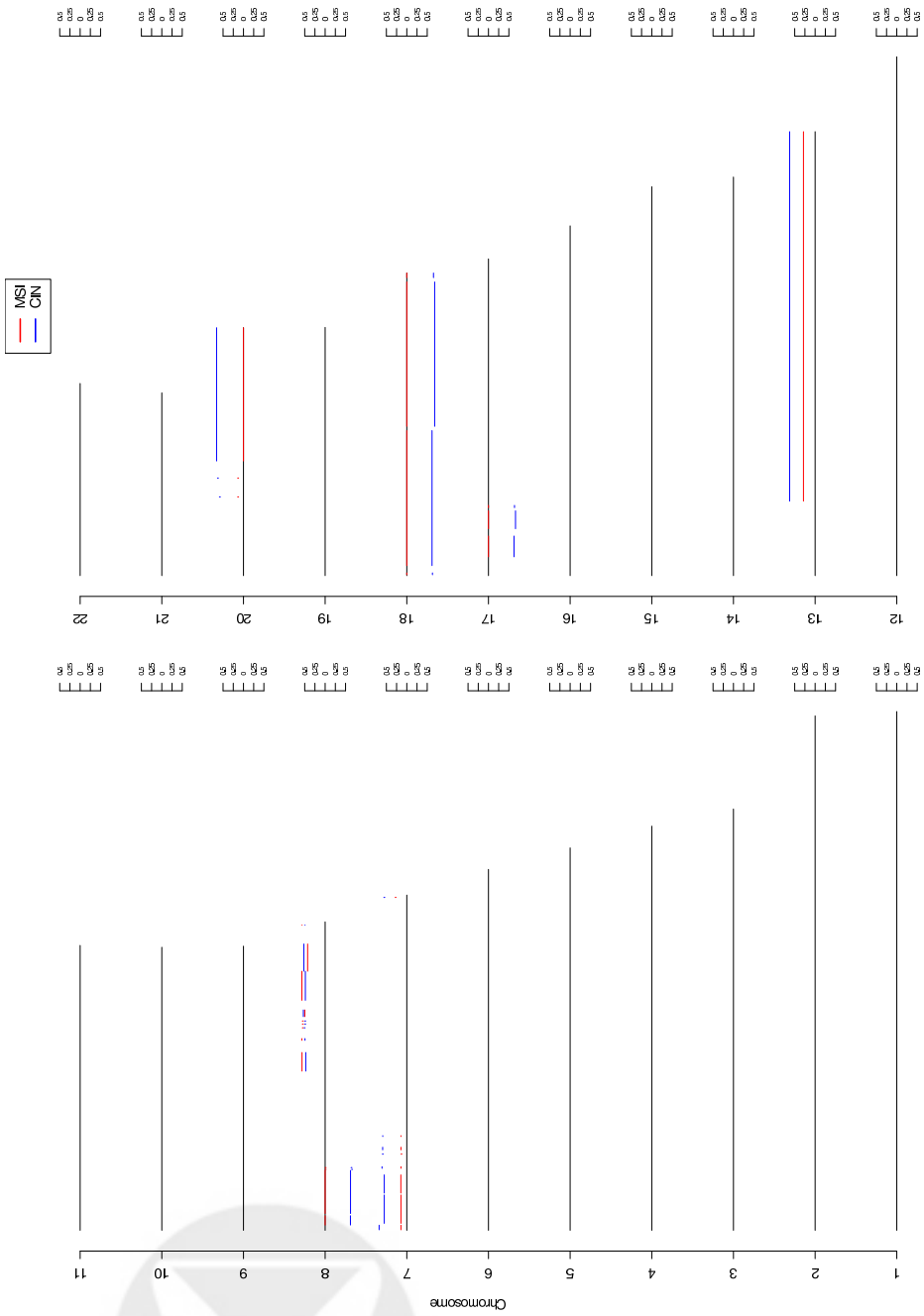

Figure 5 - Douglas et al. example.

Joint probability in MSI and CIN alterations for the Common Regions of at least 0.5 probability (i.e., pREC-A, $p_a = 0.5$). Along the abscissa, for each chromosome, the position; the coordinate indicates the probability, with values below 0 indicating loss, and above 0 gain.

Tables

Table 1 - Simulated data example. Marginal probabilities of being gained.

	S1	S2	S3	S4	S5	S6
A1	0.17	0.17	0.97	0.97	0.97	0.17
A2	0.16	1.00	1.00	1.00	0.15	1.00
A3	0.08	0.07	0.93	0.07	0.06	0.92
A4	0.16	0.16	0.99	1.00	1.00	1.00

Table 2 - Common regions in [32]

Results using pREC-A, $p_a = 0.35$. This analysis shows results at cytoband resolution.

Gain		Loss	
Nakao et al.	pREC-A	Nakao et al.	pREC-A
7p	7p	-	1p
7q	7q	5q	5q
8q	8q	8p	8p
11q	-	17p	17p
-	13q	18	18
20q	20q	21q	-

Table 3 - Common regions in [32]

pREC-A, $p_a = 0.50$. Analysis at BAC resolution.

Chrom.	Start	End	#Probes	Prob.	Alteration
20	32.33	32.33	2	0.5976	Gain
20	32.718	46.643	19	0.5003	Gain
20	47.321	60.461	29	0.5948	Gain
20	63.878	63.878	1	0.5026	Gain
20	64.021	65	2	0.6219	Gain
17	7.518	7.518	1	0.5002	Loss
17	8.17	8.17	1	0.5054	Loss
17	9.118	9.118	1	0.5038	Loss
17	10.004	16.581	5	0.5002	Loss
18	3.731	5.014	3	0.5019	Loss
18	6.251	6.251	1	0.5026	Loss
18	9.188	10.75	2	0.5038	Loss
18	10.862	11.925	3	0.5119	Loss
18	13.559	13.559	1	0.5087	Loss
18	14.77	58.594	15	0.5009	Loss
18	62.332	90	18	0.5438	Loss

Table 4 - Alterations in [31] by Estrogen Receptor

Values shown: $\bar{Y}_k/\bar{Y}_{-k}$

	Gain		Loss	
Chrom.	ER='+'	ER='-'	ER='+'	ER='-'
1	0.61	1.16	1.54	0
2	0.12	0	1.93	0.17
3	46.67	0	2.77	0.51
4	0	3.43	0.66	0.65
5	0.21	2.72	0.46	1.37
6	0.12	2.1	0.39	1.4
7	2.27	0.09	0.25	0.85
8	0.5	1.54	0.22	1.52
9	0.55	1.7	1.68	0.11
10	3.62	0	2.06	0
11	2.82	0.19	0.26	0.07
12	11.03	0	0.83	0
13	0.94	0.21	0.38	1.26
14	0.99	0	0.85	0.59
15	0.71	0.3	∞	0
16	2.18	0.34	0.72	1.25
17	0.55	1.55	2.14	0.9
18	0.27	1.73	2.61	0
19	1.04	0	1.22	0
20	0.8	0.94	0.58	1.49
21	0.66	0.68	0.92	0.56
22	0.48	0.67	2.09	0
X	1.56	0	0.53	0.38

Table 5 - Alterations in [31] by TP53 mutations.

Values shown: $\bar{Y}_k/\bar{Y}_{-k}$

	Gain		Loss	
Chrom.	p53='Wt'	p53='Mutant'	p53='Wt'	p53='Mutant'
1	0.83	0.86	3.96	0.5
2	0	36.71	0.82	0.66
3	0.45	0.79	0.2	1.51
4	0	2.82	0.73	0.73
5	0.17	1.44	1.73	0.51
6	0.12	1.29	0.88	0.74
7	0.55	1.24	1.17	0.5
8	0.3	2.31	0.35	0.96
9	1.46	0.79	0.67	1.13
10	0.2	33.18	1.65	0.25
11	0.63	0.84	0.78	0.14
12	1.51	0	0.14	0.51
13	5.23	0.48	0.63	0.88
14	0.89	0.6	0.04	3.76
15	0.06	1.01	0.26	0.24
16	0.52	1.51	2.32	0.34
17	0.43	1.72	0.63	8.23
18	0.19	2.21	2.24	0.28
19	0.58	0.49	1.14	0.26
20	0.64	1.24	0.52	1.51
21	0.66	0.91	0.35	1.9
22	0.25	1.78	1.6	0.07
X	0	0.71	0.37	1.12

Table 6 - Alterations in [31] by tumor grade

Values shown: $\bar{Y}_k/\bar{Y}_{-k}$

	Gain			Loss		
Chrom.	Grade I	Grade II	Grade III	Grade I	Grade II	Grade III
1	1	0.39	1.04	0	1.07	0.28
2	0	0.03	0	4	0.2	0.53
3	0	2.14	0.02	0.86	0.6	2
4	0	0	0	0	0.65	1.12
5	0	0.01	3.45	0.89	0.57	0.75
6	0	1.59	0.42	1.33	1.06	0.72
7	2.58	0.22	0.75	0	1.52	0.41
8	0	0.47	2.47	0	0.74	0.64
9	5.36	0.32	0.49	0.81	1.06	0.81
10	0	0	1.67	1.87	1.08	0.05
11	0.16	0.47	1.48	0	0.86	0.12
12	0	0.83	0.09	6.33	0.44	0.81
13	1.91	0.14	0.82	0.15	0.34	1.85
14	0	0.6	1.54	0.79	0.25	1.38
15	0	1.05	0.5	0	0.55	0
16	0	0.77	1.32	0	1.4	1.15
17	0	0.29	3.56	0.64	0.11	1.89
18	1.12	0.22	1.47	0.57	1.1	0.47
19	0	1.02	0.36	2.71	2.86	0
20	0	0.39	2.49	0	0.55	1.11
21	1.14	0.68	0.66	1.57	0.71	0.86
22	0	0.39	0.7	0	1.1	0.18
23	0	0	∞	1.83	1.57	0.1

Table 7 - Alterations in [31], genomewide

.

The homogeneity index, $\bar{Y}_k/\bar{Y}_b$, is computed over the whole genome, not chromosome by chromosome, as done in previous tables.

		pREC-S (Homogeneity index)
ER	Positive	0.75
	Negative	1.12
p53	Wild Type	0.67
	Mutant	1.23
Grade	I	1.21
	II	0.56
	III	1.40

Table 8 - Region differences in Douglas et al.

Regions that differ with respect to copy number alterations between CIN and MSI groups in the data set from [33]. See text for details of test.

Chrom.	Start	End	#Probes	Prob. alteration	Alteration	p-value	FDR-adjusted p-value
7	254610	2436414	5	0.5493	Gain	0.0001	0.0002
7	3293630	16702590	19	0.5001	Gain	0.0104	0.0166
7	17587872	26210052	9	0.5020	Gain	0.0111	0.0169
7	29279112	30070392	2	0.5195	Gain	0.0042	0.0090
7	35993377	35993377	1	0.5130	Gain	0.0048	0.0090
7	38011390	39281976	2	0.5065	Gain	0.0061	0.0103
7	44307040	44727994	2	0.5045	Gain	0.0053	0.0094
13	19104448	113866204	103	0.5327	Gain	0.0278	0.0404
20	20191940	20191940	1	0.5055	Gain	0.0047	0.0090
20	25023262	25023262	1	0.5441	Gain	0.0006	0.0014
20	29402772	63589868	51	0.5535	Gain	< 0.0001	< 0.0001
8	2520596	6933218	10	0.5023	Loss	< 0.0001	< 0.0001
8	7938098	28300098	25	0.5040	Loss	< 0.0001	< 0.0001
8	28775788	28775788	1	0.5252	Loss	< 0.0001	< 0.0001
8	29649361	29649361	1	0.5135	Loss	< 0.0001	< 0.0001
17	4824380	10156678	12	0.5072	Loss	< 0.0001	< 0.0001
17	12025982	16624989	6	0.5371	Loss	< 0.0001	< 0.0001
17	17432136	18029867	2	0.5170	Loss	< 0.0001	< 0.0001
18	225168	707954	3	0.5091	Loss	< 0.0001	< 0.0001
18	2572772	37207434	41	0.5011	Loss	< 0.0001	< 0.0001
18	38298595	75324734	47	0.5531	Loss	< 0.0001	< 0.0001
18	76423282	77615559	6	0.5297	Loss	< 0.0001	< 0.0001