ABSTRACT
Motivation: Plots of array Comparative Genomic Hybridization (CGH) data often show special patterns: stretches of constant level (copy number) with sharp jumps between them. There can also be much noise. Classic smoothing algorithms do not work well, because they introduce too much rounding. To remedy this, we introduce a fast and effective smoothing algorithm based on penalized quantile regression. It can compute arbitrary quantile curves, but we concentrate on the median to show the trend and the lower and upper quartile curves showing the spread of the data. Two-fold cross-validation is used for optimizing the weight of the penalties.

Results: Simulated data and a published dataset are used to show the capabilities of the method to detect the segments of changed copy numbers in array CGH data.

Availability: Software for R and Matlab is available.

Contact: p.eilers@lumc.nl

1 INTRODUCTION
Genomic arrays give a detailed picture of deletions and amplifications along chromosomes. If changes in copy numbers occur, we expect these to be visible in segments that cover multiple probes, because fragments of chromosomes are generally affected. The spatial information can be used to reduce noise and increase the reliability of detecting changes.

Using spatial information means that some form of smoothing is being applied. But classical methods are not very helpful here: they blur the jumps that occur at the sudden changes of copy numbers and they round, rather than flatten the segments between the jumps.

One alternative approach is to model the data explicitly as a series of segments, with unknown boundaries and unknown heights. These have to be estimated from the data. One can set up a performance measure that accounts for the quality of the fit, but also penalizes the number of jumps. The optimization of such a criterion is far from trivial. Jong et al. (2004) use stochastic genetic algorithms. They also provide a computer program that implements it. There are several parameters that the user has to tune.

One can also work systematically along a data series and introduce (or remove) breakpoints iteratively, to optimize a quality criterion. This approach is used by Olshen et al. (2004).

Others have modelled the problem as a hidden Markov chain (Fridlyand et al., 2004). One chooses a set of possible hidden states and a transition matrix between these states and the observations and applies an appropriate algorithm for hidden Markov models.

Hsu et al. (2004) propose to fit wavelets to the data and shrink the coefficients. In the flat parts most of the higher frequency coefficients will become zero, but near the jumps they will be retained. The positions of the jumps will be indicated in that way. Once these are set, one has to only average the data in the segments between them to get the final result.

Hupé et al. (2004) use adaptive weights smoothing to detect breaks and report the competitive results.

We emphasize visualization instead of breakpoint detection and present a smoothing method inspired by penalized least squares (Eilers, 2003). We use the $L_1$ norm, the sum of absolute values, both in the measure of fit and in the roughness penalty. This leads to a large but sparse linear program, which can be solved efficiently with an interior point algorithm. When combined with $0/1$ weights, the penalty makes smooth interpolation of left-out observations trivial, allowing efficient and elegant cross-validation. We also introduce asymmetric norms to estimate quantile curves. These provide powerful visual cues to the variability of the data.

For completeness, we note that Lutz Dümbgen (University of Bern) directed us towards the taut string algorithm (Davies and Kovac, 2001). This computes a piecewise linear (no jumps), least squares fit to a data series. If we present cumulative data ($x_i, \sum_{k=1}^i y_k$) to the taut string algorithm and then compute the (trivial) derivative of the result, we get a piecewise constant (with jumps) fit to the original data. Our limited experience indicates that the taut string algorithm is fast and effective and as such is a strong competitor to our algorithm. However, it misses the regression structure and quantile estimation.

2 THEORY AND COMPUTATION
A smoothing algorithm, with a long history, is proposed by Whittaker (1923). If $y$ is a series of $m$ data points and $z$ is the smooth series that should approximate $y$, we minimize:

$$Q_z = \sum_{i=1}^m (y_i - z_i)^2 + \lambda \sum_{i=2}^m (z_i - z_{i-1})^2.$$  (1)

The first term measures the fit of $z$ to $y$, the second term is a so-called penalty: it discourages changes in $z$. The influence of the penalty is tuned by the parameter $\lambda$: the larger $\lambda$ is chosen, the smoother $z$ will be, at the cost of a worse fit to the data. Eilers (2003) presents a detailed treatment with several applications. This smoothing algorithm is not a good choice for array Comparative Genomic Hybridization (CGH) data, as shown in Figure 1. It converts jumps into gradual changes and tends to round plateaus.
Quantile smoothing of array CGH data

A solution for this is to change the objective function in (1) into

\[ Q_1 = \sum_{i=1}^{m} |y_i - z_i| + \lambda \sum_{i=2}^{m} |z_i - z_{i-1}|. \]  (2)

The sums of squares have been replaced by sums of absolute values: we have moved from the \( L_2 \) norm to the \( L_1 \) norm. As shown in Figure 2, we now get a desirable result: sudden jumps and flat plateaus. However, we pay a price: minimization of \( Q_1 \) typically involves more work.

We can look at the right-hand side of (2) as a single summation of \( 2m - 1 \) absolute values of terms which are functions of \( y, \lambda \). Then, we can borrow ideas from quantile regression and linear programming to fit the data.

Quantile regression (Koenker and Basset, 1984) considers the following problem: we have a vector \( y \), a regression basis \( B \) and \( n \) regression coefficients \( \alpha \). With \( \tau \) a parameter between 0 and 1, we minimize

\[ S(\tau) = \sum_{i=1}^{m} \rho_{\tau}(y_i - \sum_{j} b_{ij} \alpha_j). \]  (3)

Here, \( \rho_{\tau}(u) \) is the so-called check function; it is \( \tau u \) when \( u > 0 \) and \( (\tau - 1)u \) when \( u \leq 0 \). The effect in (3) is to return weighted absolute values of residuals, such that positive residuals get weight \( \tau \) and negative residuals get weight \( 1 - \tau \). When \( \tau = 0.5 \), the weights are independent of the sign, so solving (3) is equivalent to solving (2). We could call this median regression.

Portnoy and Koenker (1997) give a detailed account of quantile regression. They also present efficient algorithms, based on the interior point method for linear programming. These have been implemented in the package quantreg for R, available from http://www.r-project.org. We now show how to re-write (2) as a median regression problem. Let

\[ y^* = \left(\begin{array}{c} y \\ 0 \end{array}\right) \quad \text{and} \quad B = \left(\begin{array}{c} I \\ \lambda D \end{array}\right). \]  (4)

where \( I \) is the \( m \times m \) identity matrix, \( 0 \) is a vector of \( m - 1 \) zeros and \( D \) is a matrix such that \( Dz = \Delta z \). Thus \( D \) is the \( m - 1 \times m \) matrix that transforms \( z \) into differences of neighboring elements. In R it is trivial to obtain \( D \): \( \text{D} = \text{diff(diag(m))} \). If we call the function \text{rq.fit} with \( y^* \) and \( B \), we get back the desired results. The following code fragment shows the details

```r
# Median smoothing; data vectors y and x are given
m = length(y)
lambda = 1
E = diag(m)
D = diff(E)
```

Fig. 1. Smoothing of simulated data with an \( L_2 \) norm (sums of squares) for different levels of normally distributed noise (\( \sigma \): standard deviation).
Fig. 2. Smoothing of simulated data with an $L_1$ norm (sums of absolute values). Two values of $\lambda$ (upper panel and lower panel).

```r
B = rbind(E, lambda \(^(*)\) D)
ystar = c(y, rep(0, m - 1))
myrq = rq.fit(B, ystar, method = "fn")
plot(x, y,)
lines(x, myrq$coeff, col = 'red')
```

For small datasets the default method of `rq.fit` (Barrodale–Roberts) can be used. We experienced that Frisch–Newton (an interior point method), chosen by using `method = "fn"`, is much faster and more reliable for larger datasets. Computation time roughly increases with the third power of $m$. For $m = 500$, it takes 1 s on a 3 GHz Pentium 4 PC running R 1.9.1 under Windows.

A similar implementation in Matlab, based on original Ox code by Koenker and Morillo is available from the first author and is used to handle sparse matrices. This is an advantage, as both $I$ and $D$ in (4) are very sparse. In 1 s, a series of 2000 observations can be smoothed.

It is interesting to consider values of $\tau$ other than 0.5, which gives the median curve. With $\tau = 0.25$ ($\tau = 0.75$) we get the lower (upper) quartile curve. They help to give a good impression of the local spread in the data. Note that $\tau = 0.5$ should be fixed in the penalty. Both the R library `quantreg` and our Matlab implementation accept a vector of $\tau$s, giving complete freedom for separately choosing the asymmetry for fit and slopes of $z$.

The beauty of the penalty is that it can automatically and smoothly interpolate missing data via weighting. Define a vector of weights, $w$, with a 1 (0) corresponding to an available (missing) observation. The quantile regression software accepts weights, such as in the following goal function:

\[
S(\tau) = \sum_{i} w_i \rho_{\tau} \left( y_i - \sum_{j} b_{ij} \alpha_j \right),
\]

where $w_i = 0$, values for $z_i$ that minimize the penalty are automatically filled in.

Automatic interpolation makes cross-validation easy: leave out part of the observations by manipulating the weights, fit and interpolate and measure the fit to the left-out data using the check function. We have experimented with a simple scheme: leave out the even observations ($y_i$ with $i$ even), fit to the odd observations (the ones with $i$ odd) and interpolate the even ones. Then we exchange the roles of odd and even observations and repeat the procedure. The two results are added, giving coefficient of variation (CV). We vary $\lambda$ on a grid and search for the value that minimizes CV. Figure 3 shows the cross-validation trace and the fit for the ‘optimal’ $\lambda$.

The $L_1$ norm brings with it some remarkable properties. If one minimizes a sum of absolute values, the derivative is the sum of signs. This means that, in median regression, only the signs of the residuals are essential. Hence, any monotone scaling of $y$ will yield essentially the same regression fit. In particular, fitting raw
array data or log-scaled data yields the same fits (scaled in the same way on $y$).

We have not yet considered the positions $x$ of the observed $y$. In general, the distances between adjacent $x$ will be variable. One might be tempted to correct for this and use divided differences, \((z_i - z_{i-1})/(x_i - x_{i-1})\), in the penalty but again we find that only the signs matter. Because the $x$ are ordered such that $x_i > x_{i-1}$, the divisions do not change the signs. The conclusion is that any monotone change of scale of $x$ will give the same results. Therefore, in practice clone positions are relevant only for visualization.

The Matlab implementation is superior to the R one due to the use of sparse matrices. These enable us to reasonably quickly fit models to a chromosome with a large (>1000, say) number of clones. To improve performance in large problems in R, an approximation can be used as follows: replace the identity matrix in $B^*$ in (4) by an indicator matrix, $C$. We minimize $|y - Cz| + \lambda |Dz|$. The domain of $x$ is divided into \(n\) intervals, say 100. Each row of $C$ contains \(n-1\) zeros and one 1. The 1 in row $i$ indicates the interval in which $x_i$ lies. The use of the indicator matrix reduces the resolution of the position of change points to 1%, but this will be accurate enough for most applications.

The regression structure of the \(L_1\) smoother can be exploited in a variety of ways, for example, the estimation of common quantile functions for a set of (replicated) array CGHs. Multiple identity matrices $I$ can be put on top of each other to construct $R^*$ and multiple $y$ vectors to form $y^*$ in (4). Missing observations can be accommodated with zero weights.

3 BIOLOGICAL ASPECTS OF CHROMOSOME CHANGES

The array CGH is used to assess relative chromosome copy number, typically between the sample under study and a control sample. Many cytogenetically distinct mechanisms can yield chromosome copy number change. We can divide these mechanisms into two groups, depending on the association they imply on relative copy number of the genomic region.

The first, commonly known as chromosome gain or loss depending on the direction in which it acts, consists of changes implying strong association between relative copy number of neighboring clones over a wide region. We refer to this sort of effect as wide change, where ‘wide’ refers to the extent of the correlation.

The second, known as amplification or deletion, refers to the mechanisms acting on small genomic regions, in such a way that little association is found between chromosome copy number in this small region and in the neighboring segments. For this reason, we refer to it as local change.

Mechanisms of both types can play a relevant role in pathogenesis. See references in Nakao et al. (2004), of which we highlight Knuutila et al. (1998), on the role of chromosome gain or loss and Albertson et al. (2000) on amplification and deletion.

4 APPLICATIONS

Nakao et al. (2004) collected 125 samples of colon carcinomas. The DNA from these samples was extracted and frozen; the non-amplified
material was hybridized to the arrays. Array CGHs, using a bacterial artificial chromosome (BAC) clone library with clones 1.5 Mb apart, were provided by the UCSF Cancer Centre Array Core. These arrays use a two-colour system with a common reference sample. The data were made available in the form of log2-ratios of sample versus reference per array.

The downloaded file included 2120 BAC clones, of which 151 have all values missing. Thus, 1969 clones were used here for analysis. We focus on chromosome 1, which involves 133 clones. Missing values were given zero weight.

We apply our proposed smoother to the data, using $\lambda = 4$ as smoothing parameter. We also compute bounds for the data based

---

**Fig. 4.** Chromosome 1 from sample X59. Smoothed trends computed with $\lambda = 4$. Horizontal axis indicates chromosome position (Mb) of each clone. Upper panel: observed data. Lower panel: smoothed trends were added; solid line represents the median trend, dashed lines represent both 15 and 85% quantiles. No change level is represented by dotted lines.

**Fig. 5.** Chromosome 1 from sample X524. Smoothed trends computed with $\lambda = 4$. Horizontal axis indicates chromosome position (Mb) of each clone. Upper panel: observed data. Lower panel: smoothed trends were added; solid line represents the median trend, dashed lines represent both 15 and 85% quantiles. No change level is represented by dotted lines.
Chromosome 1 for sample X204. Smoothed trends computed with $\lambda = 4$. Horizontal axis indicates chromosome position (Mb) of each clone. Upper panel: observed data. Lower panel: smoothed trends were added; solid line represents the median trend, dashed lines represent both 15 and 85% quantiles. No change level is represented by dotted lines.

We first choose two samples with clear trends to illustrate the way in which our smoother works. Sample X59 (Fig. 4) has both loss of 1p and gain of the first segment of 1q, as evident from the smoothed medians lying beyond defined detection bounds, with the telomere seemingly preserved as normal. Sample X524 (Fig. 5) displays chromosome loss over a couple of regions on 1p, as well as...
a high-level amplification within a much smaller region, the latter a seemingly local effect. It is noted that wide changes are captured by the smoothed median, whereas local changes lie well outside the bounds.

Sometimes, however, patterns are not evident from the observed data and our smoother can help to visualize the median trend, as well as quantifying the effect detected. Consider for example the top graphs in Figures 6 and 7, which correspond to samples X186 and X204. For X186, the overall log$_2$-ratio on 1p seems to be $\approx 0$, but it is not clear whether the change is biologically meaningful, i.e. if it lies outside the defined detection bounds. Similarly, for sample X204, the log$_2$-ratio on 1q seems to be $>0$, especially in a segment between 220 and 250 Mb, but it is not clear whether this is biologically meaningful.

Once the smoothed median and bounds are added to the graphs, a much clearer picture emerges. The smoothed median trend of X186 is $\approx -0.1$, so this can be considered to be within the noise bounds ($-0.14, 0.14$) for entire chromosome arm loss. However, the smoothed percentile of 85% is $<0$ for almost the entire length of 1p suggesting that, although not clearly detectable, the trend is discernible from the noise. We are now more confident to discard any chromosome gain over 1q in sample X204, since the computed smoothed median trend between 220 and 250 Mb is below the detection level of 0.225.

5 DISCUSSION

We have presented a simple, efficient and accessible algorithm for smoothing array CGH data. Simulations and applications to real data illustrate its performance. A unique feature is that it shows trends as well as variability around them. Software is freely available in both Matlab and R.

Most statistical models proposed to analyze array CGH data involve modelling the association between changes in neighboring clones. While this is helpful to find wide changes, it tends to ignore local changes. The use of our proposed smoother fitted via quantile regression yields the ability to visualize both these changes: via changes in the smoothed median trend wide changes can be detected, whilst sets of consecutive copy number measurements outside smoothed quantile bounds suggest local changes.

We position our work primarily as a tool for visualization and exploration. In some cases the genomic trend is clear enough from the data, whereas in others the trend or lack of it, is unclear at its initial observation. A flexible, non-parametric tool as the smoother presented here makes for an ideal tool for this purpose.

We have purposely stayed away from significance testing. While this is possible within the context of quantile regression (Koenker and Machado, 1999; Redden et al., 2004), many more issues play a role specifically in the application to array CGH data. First, statistical significance in general refers to the consistent effects, i.e. effects that occur in a large enough proportion to rule out chance. However, in one of the most currently important applications of array CGH, most effects of carcinogenesis occur in a much too small proportion of the cases. While analysing 125 arrays from colon carcinomas, Nakao et al. (2004) found out that 95% of the chromosome changes occur in $<35\%$ of the cases. Moreover, any changes found with the array must be verified by confirmatory experiments. The decision about which patterns to confirm is, almost invariably, made based upon the biological context. Biologists are more likely to verify changes in a region where a known gene is (perhaps involved in other diseases) than a region with a small $P$-value. For these reasons, we believe that statistical testing in this context is unlikely to be helpful.

The amount of dispersion, as well as the overall signal-to-noise ratio, of the data is determined by the protocol used in array production. In particular, the use of DNA amplification plays an important role (see Nygaard et al., 2003). Therefore, the choice of smoothing parameter is unlikely to be unique. We suggest that biologists play with the smoothing parameter, $\tau$, with some guidance by cross-validation and see whether strong patterns present themselves. One idea is to use the data from pilot experiments to estimate the amount of smoothing needed for the protocol and platform used. It might also be fruitful to explore a range of values of the asymmetry parameter, $\tau$, to get good impression of the pattern of variability in the data.

The regression structure is a clear advantage of our model, compared both with simpler smoothing methods such as moving medians and more complex ones such as the taut string algorithm (Davies and Kovac, 2001). Our approach can be embedded in larger models, in which certain components, like reference arrays, are shared. It can be specifically used in a context where the distribution of copy number can be expressed in terms of a regression model which depends on experimental variables and the two channels are decoupled, so intensities for the two channels are handled separately. In principle any ANOVA-like structure can be combined with penalties for smoothness of components. This extension is beyond the scope of this work and will be explored elsewhere.

Application of the model is not limited to array CGH data. At present, we are investigating median smoothing for expression data, in order to study correlations with genomic outcomes, as done via other methods by Aguirre et al. (2004). Expression arrays can have thousands of probes per chromosome. The use of an indicator matrix is needed to get good performance.

ACKNOWLEDGEMENTS

We are grateful to J. Cardoso and B. Ylstra for their fruitful discussions in the preparation of this work.

REFERENCES


