BMC Bioinformatics

BioMed Central

Software

Open Access

BGX: a Bioconductor package for the Bayesian integrated analysis of Affymetrix GeneChips

Ernest Turro*[1], Natalia Bochkina[2], Anne-Mette K Hein[3] and Sylvia Richardson[1]

Address: [1]Centre for Biostatistics, Imperial College London, UK, [2]Department of Mathematics, University of Edinburgh, UK and [3]Molecular Diagnostic Laboratory, Skejby Hospital, Aarhus University, Denmark

Email: Ernest Turro* - ernest.turro@ic.ac.uk; Natalia Bochkina - n.bochkina@ed.ac.uk; Anne-Mette K Hein - anne.mette.hein@ki.au.dk; Sylvia Richardson - sylvia.richardson@imperial.ac.uk

* Corresponding author

Published: 12 November 2007

BMC Bioinformatics 2007, **8**:439 doi:10.1186/1471-2105-8-439

Received: 11 September 2007
Accepted: 12 November 2007

This article is available from: http://www.biomedcentral.com/1471-2105/8/439

Abstract

Background: Affymetrix 3' GeneChip microarrays are widely used to profile the expression of thousands of genes simultaneously. They differ from many other microarray types in that GeneChips are hybridised using a single labelled extract and because they contain multiple 'match' and 'mismatch' sequences for each transcript. Most algorithms extract the signal from GeneChip experiments in a sequence of separate steps, including background correction and normalisation, which inhibits the simultaneous use of all available information. They principally provide a point estimate of gene expression and, in contrast to BGX, do not fully integrate the uncertainty arising from potentially heterogeneous responses of the probes.

Results: BGX is a new Bioconductor R package that implements an integrated Bayesian approach to the analysis of 3' GeneChip data. The software takes into account additive and multiplicative error, non-specific hybridisation and replicate summarisation in the spirit of the model outlined in [1]. It also provides a posterior distribution for the expression of each gene. Moreover, BGX can take into account probe affinity effects from probe sequence information where available. The package employs a novel adaptive Markov chain Monte Carlo (MCMC) algorithm that raises considerably the efficiency with which the posterior distributions are sampled from. Finally, BGX incorporates various ways to analyse the results, such as ranking genes by expression level as well as statistically based methods for estimating the amount of up and down regulated genes between two conditions.

Conclusion: BGX performs well relative to other widely used methods at estimating expression levels and fold changes. It has the advantage that it provides a statistically sound measure of uncertainty for its estimates. BGX includes various analysis functions to visualise and exploit the rich output that is produced by the Bayesian model.

Background

Oligonucleotide microarrays allow biomedical researchers to estimate the expression of thousands of genes simultaneously through their mRNA transcripts. A labelled, fragmented version of the RNA may be hybridised onto an array containing hundreds of thousands of complementary oligonucleotides and then scanned. Affymetrix 3' GeneChip arrays represent genes by sets of probe pairs, each of which consists of an oligonucleotide of length 25 which matches a corresponding RNA subsequence perfectly (PM) and an identical probe with an inverted oligonucleotide on position 13 (MM) that is intended to measure non-specific hybridisation.

The BGX model [1] is an integrated approach to the analysis of GeneChip microarrays in which correction for non-specific hybridisation and gene expression level estimation are performed simultaneously. Posterior distributions of parameters in the model may be obtained numerically. Based on these distributions, a powerful method for detecting differential expression has been developed [2].

The probes on Affymetrix GeneChips have been found to exhibit varying propensities to "shine" according to the base composition of their sequences [3] and methods for estimating expression levels from GeneChips that incorporate probe affinity effects have shown demonstrable advances over methods in which these effects are ignored (see, e.g. [4]). We present a new Bioconductor [5] package that implements the BGX model, includes an extension to incorporate probe affinity effects, employs novel algorithmic techniques to sample effectively from posterior distributions, and provides various analysis and plotting functions.

Implementation

Basic model

BGX [1] explicitly models probe intensities as arising partly from specific hybridisation, S (the signal), and partly from non-specific hybridisation, H, with only a fraction, ϕ, of the signal occuring at a PM probe also occurring at the corresponding MM probe. The Ss and Hs are gene (g), probe (j), condition (c) and replicate (r) specific, and the intensities are assumed to be affected by an additive array-specific noise:

$$\mathrm{PM}_{gjcr} \sim N(S_{gjcr} + H_{gjcr}, \tau_{cr}^2), \tag{1}$$

$$\mathrm{MM}_{gjcr} \sim N(\phi S_{gjcr} + H_{gjcr}, \tau_{cr}^2). \tag{2}$$

The log-transformed signal parameter, $\log(S_{gjcr} + 1)$, is assumed to follow a gene and condition specific distribution, while the log-transformed non-specific hybridisation term, $\log(H_{gjcr} + 1)$, is assumed to arise from an array-specific distribution:

$$\log(S_{gjcr} + 1) \sim TN(\mu_{gc}, \sigma_{gc}^2), \tag{3}$$

$$\log(H_{gjcr} + 1) \sim TN(\lambda_{cr}, \eta_{cr}^2), \tag{4}$$

where TN denotes the truncated normal distribution, truncated to the positive axis. The central parameter of equation (3), μ_{gc}, acts as the BGX expression measure, and equations (1) to (4) represent the basic BGX model.

The core of the model is implemented in the C++ programming language for efficiency and uses MCMC to sample from the full posterior distributions of each parameter. Parameters are estimated using Gibbs sampling where possible (ϕ and τ) and a Random Walk Metropolis-Hastings algorithm elsewhere (S, H, μ, σ, λ and η). Three C++ class templates are used to instantiate zero, one and two-dimensional MCMC update objects for each parameter according to the dimensionality of the corresponding suffixes. Each of the instantiated objects is updated in sequence using references to all other necessary parameters during a burn-in period, which is discarded, and a sampling period, which is used for the posterior distributions.

Probe affinity extension

It has been observed that the propensity of probes to hybridise to mRNA is affected by their base composition [3]. In particular, probes with a high number of cytosine bases have a high propensity to hybridise while probes with a high number of adenine bases exhibit the opposite tendency. Moreover, the nearer the bases are to the centre of the oligonucleotide, the greater the effect. We account for this in an extension to the core model that incorporate affnity effects in the modelling of non-specific hybridisation. We categorise probes in the following way: let α be a function which, for each gene and probe pair, (g, j), gives the affnity category of a given probe: $\alpha : (g, j) \to \{1, ..., K\}$ (defined below). We refine equation (4) by allowing for a category and array specific distribution of the non-specific hybridisation parameter:

$$\log(H_{gjcr} + 1) \sim TN(\lambda_{cr}^{\alpha(g,j)}, \eta_{cr}^2). \tag{5}$$

The extended model, which we denote GCBGX, is based on equations (1), (2), (3) and (5).

The probes on the arrays are, prior to analysis, grouped into a number of probe affinity categories. This is done by: (a) calculating the probe affinities using the *gcrma* Bioconductor package [4], (b) rounding them to the first decimal place, (c) assigning each value to a preliminary probe affinity category and (d) ensuring that the final categories contain a sufficient number of probes by collapsing small preliminary categories together. We enumerate the resulting probe affinity categories 1, ... , K by increasing affinity. Once categorised, the affinity-specific parameters are estimated from the data, simultaneously with all other parameters.

In some cases, Affymetrix do not directly provide the sequences for all probesets due to licensing restrictions and, consequently, there are Bioconductor probe packages that do not contain complete sequence information. For example, *hgu95aprobe* (version ≤ 1.16.2) lacks sequences for probes belonging to 172 probesets. We tackled this problem by treating $\alpha(g, j)$ as a random variable, taking values from 1 to K, with prior probability equal to the observed frequency of the categories, $p_k = \frac{N_k}{N}$, where N_k is the number of probes in category k and $N = \sum_{k=1}^{K} N_k$.

Adaptive MCMC

The full conditional distributions of S, H, μ, σ, λ and η are updated by drawing new values from a proposal distribution, typically a Random Walk (RW) Gaussian proposal centred on the current value with a chosen variance. A typical experiment consists of several hundred thousand probes, resulting in potentially millions of S and H components and tens of thousands of μ and σ components. Each component of a given parameter has a different support and consequently a different optimal RW proposal variance. Using a fixed variance for all components results in excessively low or high acceptance ratios for a large proportion of components, leading to highly autocorrelated chains.

In order to tackle this problem, we implemented the novel Adaptive Metropolis-Within-Gibbs algorithm recently proposed by Roberts and Rosenthal [6,7]. We used a unique proposal variance for each object, which adapts to its optimal value after successive batches of 50 iterations. The aim is to achieve an acceptance ratio of around 0.44, which has been shown to be optimal for one-dimensional proposals in certain settings [8,9], and is commonly accepted as being a sensible benchmark. An acceptance rate that is close to zero implies inefficient mixing, while an acceptance rate that is close to one implies the probability space is not efficiently explored. The algorithm proceeds as follows:

- For each component c of parameter p, assign a parameter-specific starting value to the corresponding proposal variance, σ_{cp}^2

- Choose a sequence $\delta(n) \to 0$. We chose $\delta(n) = \min(0.01, n^{-1/2})$

- Start the MCMC simulation

- After the n^{th} batch of 50 iterations, calculate the acceptance ratio over the last batch

- If the acceptance ratio is less than the optimal value of 0.44, increase $\log(\sigma_{cp}^2)$ by $\delta(n)$, else decrease it by $\delta(n)$

The algorithm preserves ergodicity as long as each kernel has the right stationary distribution; the total variation distance between successive kernels tends to zero in probability; and the convergence time of each kernel is bounded in probability [6].

R package

The C++ component of BGX is compiled as a shared object which is loaded and executed automatically from within the R package [10]. BGX integrates standard Bioconductor classes such as `AffyBatch` to store raw microarray data and `ExpressionSet` to store processed gene expression measures. Users interested in running BGX programmatically from a shell script, for instance, or in a more memory-efficient manner, also have the choice to run a standalone binary version of the program.

Results and Discussion

Usage

The *bgx* package and its dependencies, *affy* and *gcrma*, may be installed automatically from the Bioconductor repository from an R shell. The package contains documentation and executable examples in a "vignette" file available using `openVignette()`. Users who wish to compile *bgx* from source will require the Boost C++ libraries [11] and the `hgu95av2cdf` Bioconductor package. The core functionality of the package is contained in the bgx function, which takes an `AffyBatch` object instantiated from one or more GeneChip CEL files as its first argument and returns an `ExpressionSet` object containing expression values for each gene and condition:

```
aData <- Read
Affy("chip1.CEL","chip2.CEL")

eset <- bgx(aData)

assayData(eset)$exprs # Returns expression
values

assayData(eset)$se.exprs # Returns stand
ard errors for expression values
```

Optional arguments include `samplesets`, which specifies the experimental design; genes, which specifies a subset of genes to analyse; `burnin` and `iter`, which specify the number of iterations for the burn-in and post burn-in phases of the algorithm respectively; `probeAff`, which specifies whether or not to use the probe affinity extension to the original BGX model, and `adaptive`, which specifies whether or not to use Metropolis-Within-Gibbs step adaptation. Full documentation for the `bgx` function is available by running `help(bgx)`.

Although the point measures returned in the `ExpressionSet` object are useful, the distinctive power of the BGX method is that it provides samples from the full posterior distributions of the expression parameter, μ_{gc}. These samples are, by default, saved in directories named `run.1`, `run.2`, *etc.* in R's current working directory, although this may be overridden with the `rundir` argument. They may be read into R in order to analyse the results of a simulation as follows:

```
bgxOutput <- readOutput.bgx("run.1")
```

The `bgxOutput` object is assigned to a list containing values from the full posterior distributions of μ_{gc} `(bgxOutput$mu)`, their expected values `(bgxOutput$muave)` and the gene names `(bgxOutput$geneNames)`. This object can be passed to a number of functions to analyse the results. In particular, this object provides a direct measure of the variance of gene expression, a quantity which is only sometimes available when fitting robust linear models (e.g. using the AffyPLM Bioconductor package). The difference in expression between two conditions, μ_{g2} - μ_{g1}, may be visualised with the `plotExpressionDensity` and `plotDEDensity` functions (Figure 1). `plotDEHistogram` fits a spline to the histogram of $P(\mu_{g2} - \mu_{g1} < 0)$ using Poisson regression, estimates the null distribution by spline fitting of the central part of the histogram, and uses the difference between the histogram fit and the null distribution to give a preliminary estimate of the number of differentially expressed genes (Figure 2) [2]. A more thorough approach to the problem of classifying genes by differential expression may be found in the BGmix Bioconductor package — an implementation of a fully Bayesian mixture model for differential expression – which can analyse BGX output directly (Lewin, Bochkina and Richardson: Fully Bayesian mixture model for differential gene expression: simulations and model checks, submitted). `rankByDE` returns a matrix that ranks genes by their standardised BGX differences between two conditions (see Equation (7)) and specifies each gene's name, index and differential expression measure. More information on each function is available via `help(analysis.bgx)`.

For the purposes of this paper, BGX was run with the "gold standard" of 16 k burn-in iterations and 64 k sampling iterations. However, the recommended 8 k burn-in iterations and 16 k sampling iterations are sufficient to provide good estimates of μ_g (Additional File 1). Under these settings, BGX takes approximately one hour per

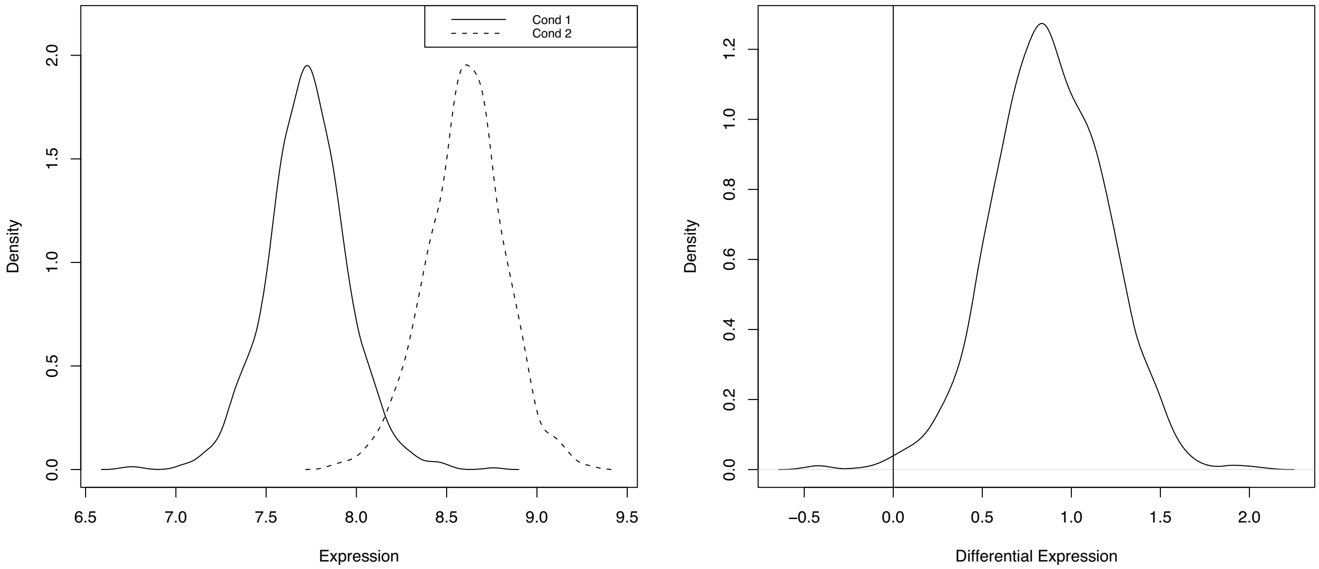

Figure 1

Expression and differential expression densities. `plotExpressionDensity` plots the density of the posterior distribution of a given gene under each condition (left). `plotDEDensity` plots the density of the difference in the posterior distributions of a given gene between two conditions (right).

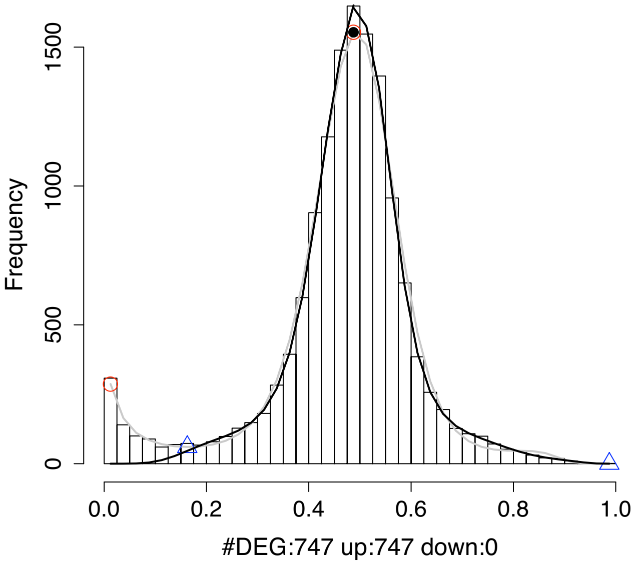

Figure 2
Estimating the number of differentially expressed genes. `plotDEHistogram` plots a histogram of $P(\mu_{g2} - \mu_{g1} < 0)$, which is used to estimate the number of up and down-regulated genes between two conditions using a routine incorporated in the package.

array on a standard 64-bit 3 GHz computer. Analyses of up to 100 arrays ought to "fit" in a computer equipped with 4 GB of memory. However, BGX may be run separately for each condition, and the output subsequently combined in R by passing multiple output directories to the `readOutput.bgx` function. Since ϕ, the only parameter that is shared between conditions, is very stable for a given type of array, the impact on the output of running BGX separately on each condition is negligible.

Estimation of non-specific hybridisation

The GCBGX model groups MM probes into categories that have similar probe affinities based on their oligonucleotide content. Instead of using a single array-specific parameter, λ_{cr} (Equation (4)), we associate an appropriate $\lambda_{cr}^{\alpha(g,j)}$ component with each probe (Equation (5)), which should correlate positively with probe affinity categories. Figure 3 shows a colour-coded density plot of the $\lambda_{cr}^{\alpha(g,j)}$ parameter obtained in an analysis of the Golden Spike data [12]. As probe affinity categories increase, the distributions shift from left to right. The black density line corresponds to the λ_{cr} distribution in the original BGX model and highlights the discriminatory power of the probe affinity extension.

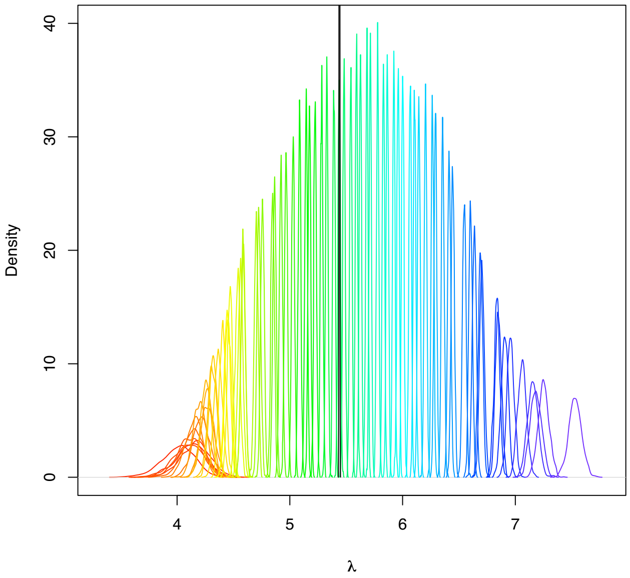

Figure 3
Distributions of λ for each probe affinity category. Colour-coded density plot of the $\lambda_{cr}^{\alpha(g,j)}$ parameter of an analysis of the Golden Spike data set. As probe affinity categories increase, the distributions shift from left to right. The black density line is the λ_{cr} distribution from the original BGX model and illustrates the discriminatory power of the probe affinity extension.

Probes with unknown sequences have their affinity categories estimated from the data. In order to check the effectiveness of this approach, we performed cross-validation on one out of every 100 genes from the Golden Spike data set, and compared the median estimated category to its true value. Figure 4 shows a positive correlation between estimated and true categories, particularly for high-affinity probes.

Performance of adaptive MCMC

BGX is a computationally intensive program and it is therefore desirable to use an MCMC algorithm that mixes efficiently. One quantity of interest in this respect is the integrated autocorrelation time (IACT), which inflates the variance of the sample mean, $\bar{X}$ [13]. To be precise, the variance of the sample mean may be expressed as follows [14]:

$$var(\bar{X}) = E\left[\frac{S^2(X)}{n}\right] \cdot \frac{n-1}{(n/a(X))-1} \qquad (6)$$

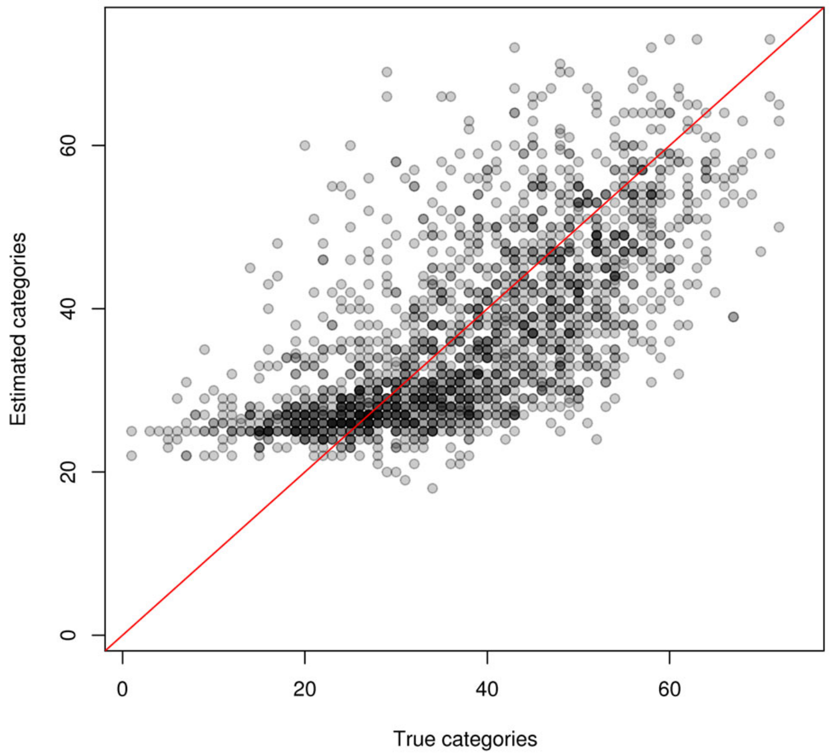

Figure 4
Estimating the affinity of probes with unknown sequence. Sequence information was artificially removed from one out ever 100 genes from the Golden Spike data set and the corresponding probes' affinity categories were estimated from the data. There is a positive correlation between estimated and true categories, particularly for high-affinity probes.

where X is an autocorrelated sample of size n with empirical mean $\bar{X}$, $S^2(X) = \frac{1}{n-1}\sum_i (X_i - \bar{X})^2$, and $a(X)$ is the integrated autocorrelation time (IACT), defined as

$$a(X) = \left[1 + 2\sum_{j=1}^{n-1}\left(1 - \frac{j}{n}\right)P_j \right]$$

where

$$P_j = \frac{cov(X_i, X_{i+j})}{\sqrt{var(X_i)var(X_{i+j})}}.$$

Evidently, if the sample is not autocorrelated, then $P_j = 0$ for all j, $a(X)$ collapses to 1 and $var(\bar{X})$ becomes equal to the familiar expression for an IID sample, E $[S^2(X)/n]$. From (6), the IACT of a chain relates positively with the variability of its mean, and thus highly autocorrelated chains lead to poor estimates of our gene expression measure.

One way of improving our estimates is to increase the number of iterations while maintaining a fixed subsample size. This translates to subsamples being further apart on the original chain and therefore less correlated. It is faster and more attractive, however, to use an adaptive algorithm that explores the probability space more efficiently. Using the Golden Spike data set [12] for our investigation, we found that the adaptive method led to a range of optimal proposal magnitudes for the Metropolis-Hastings parameters. Figure 5 illustrates this with a histogram of the optimal log variance for S proposals on one array and the original fixed step size overlaid in black. Figure 6 (left & centre) shows a dramatic reduction in the IACT of the S parameters and a milder improvement on the μ_g parameters of expressed genes. A similar improvement was observed for the IACT of the H parameters, this time for all genes (Figure 6 right).

Differential expression can be quantified by the standardised BGX differences between two conditions:

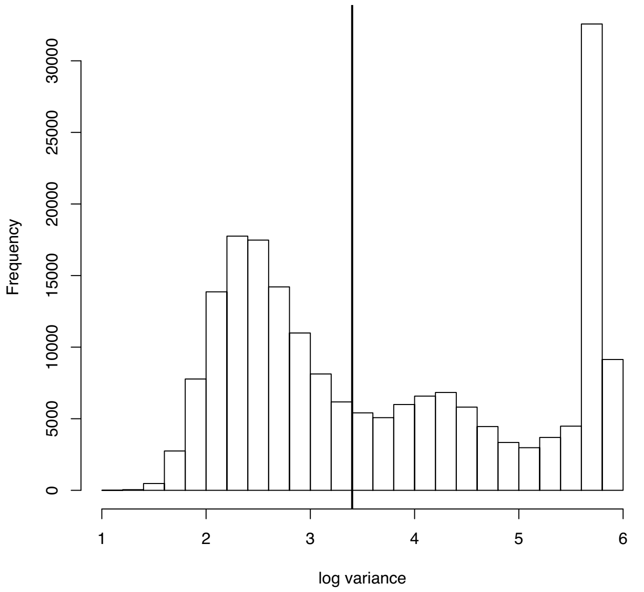

Figure 5

Adapting the size of MCMC proposal steps. When the adaptive MCMC algorithm is used, the variance of the Metropolis-Hastings proposal step is adapted independently for each *S* component. The plot shows a histogram of the optimal log variance for *S* proposals and the fixed step size used in the non-adaptive version overlaid in black, highlighting that a wide range of proposal variances are needed.

$$z_g = \frac{\bar{d}_g}{\sqrt{var(\bar{d}_g)}} \tag{7}$$

where, from (6), $var(\bar{d}_g)$ is estimated by

$$\widehat{var(\bar{d}_g)} = \frac{S^2(d_g)}{n} \cdot \frac{n-1}{(n/\widehat{a(d_g)})-1},$$

$dg = \mu_{g2} - \mu_{g1}$, $g = 1 \ldots, G$, that is, samples from the posterior distribution of the difference in the BGX expression measure for each gene, and $\widehat{a(d_g)}$, the estimate of the Monte Carlo standard error, is calculated using Sokal's adaptive truncated periodogram estimator [?]. Our z-score differs from the measure used in [2] by a factor of $\sqrt{((n/\widehat{a(d_g)})-1)/(n-1)}$, which takes into account the autocorrelation structure of the sequence of values generated by the algorithm. Since the adaptive MCMC algorithm has the effect of decreasing $a(d_g)$ for expressed genes while keeping it approximately constant for non-expressed genes, it leads to an increase in the ranking of expressed genes and consequently in BGX's capacity to detect differential expression.

Performance on spike-in datasets

We illustrate the performance of *bgx* by presenting detailed results from analyses of arrays from the Affymetrix Latin Square data [15] and the Golden Spike data set [12].

Latin Square data

Affymetrix published two data sets for assessing the performance of expression algorithms on their microarrays. The HGU95A data set consists of 16 genes spiked in at known concentrations ranging from 0 to 1024 pM and arrayed in a Latin Square format. We considered 16 instead of the original 14 genes described by Affymetrix because we included two extra spike-ins, 546_at and 33818_at, as reported in [16]. We used two replicates and 14 unique concentration configurations labelled A to M and Q. 2716 of the probes in this data set had no sequence information and therefore their probe affinity categories were estimated from the data as part of the model. The HGU133A data set consists of 64 genes spiked in at known concentrations ranging from 0 to 512 pM. We considered 64 instead of the original 42 genes described by Affymetrix because we included 22 extra spike-ins, as reported in [17]. We used all 3 replicates for each of the 14 concentration groups.

The data from these experiments were analysed using BGX, GCBGX, RMA [18], GCRMA [4] and MAS5 [?], and the average expression for each concentration level was recorded. Figure 7 (left) shows a steeper gradient at levels lower than 4 pM in the HGU95A data set using GCBGX instead of BGX, pointing to an increased ability to detect concentration changes. For both data sets, BGX and GCBGX are more sensitive to changes within the low range than RMA, GCRMA or MAS5 (Figure 7 left & right).

Golden Spike data set

The Golden Spike data set consists of six DrosGenome1 GeneChips, with three technical replicates from two conditions: C and S. There are 14010 probe sets in each array representing 14010 genes. 2535 of these are expressed equally under both conditions while 1331 genes are upregulated in S relative to C. The data is highly valuable for comparing chip analysis methods because it is fully controlled and contains very realistic noise. Due to the asymmetry of the spike-ins, a normalisation of the posterior distributions similar to that advocated in [12] was carried out by fitting a loess curve to the MA plot [19] of the posterior mean values of μ_{gc} for the non-differentially expressed genes, predicting a curve from the fit for all genes, and subtracting the curve from the posterior distri-

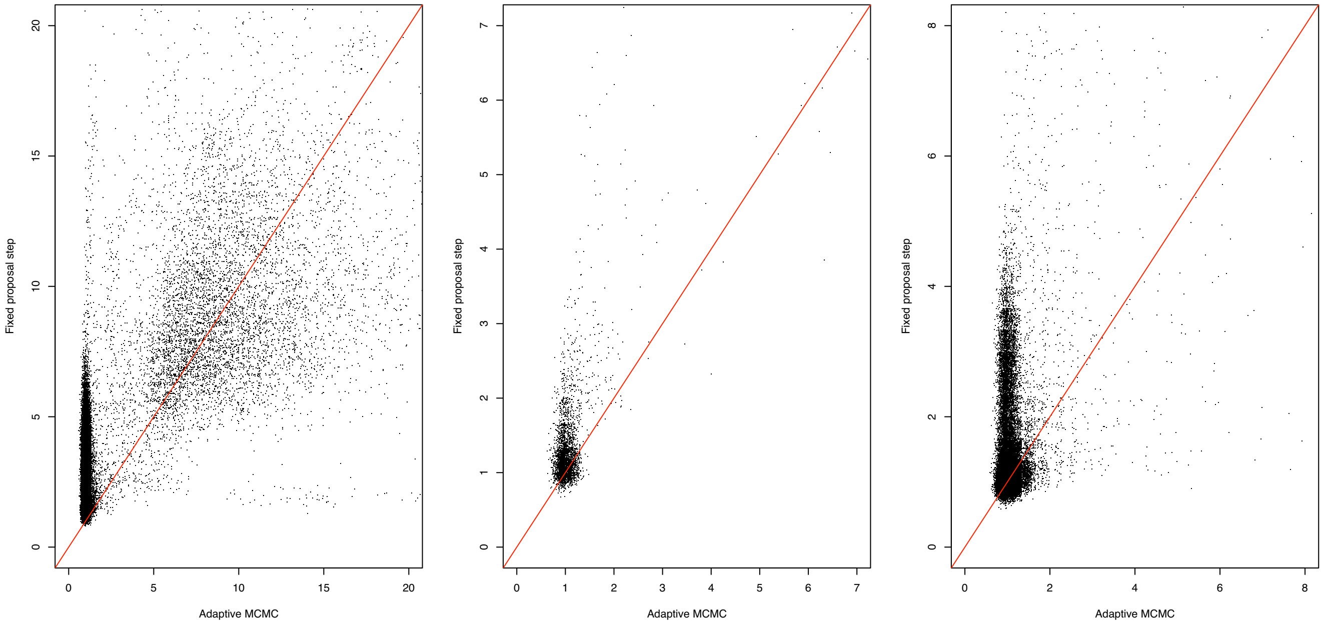

Figure 6
Decreasing the IACT of expressed genes. The plots show a dramatic reduction in the IACT of the *S* parameter and a milder improvement on the μ parameter of expressed genes (left & centre). A similar improvement was observed for the IACT of the *H* parameter, for all genes (right).

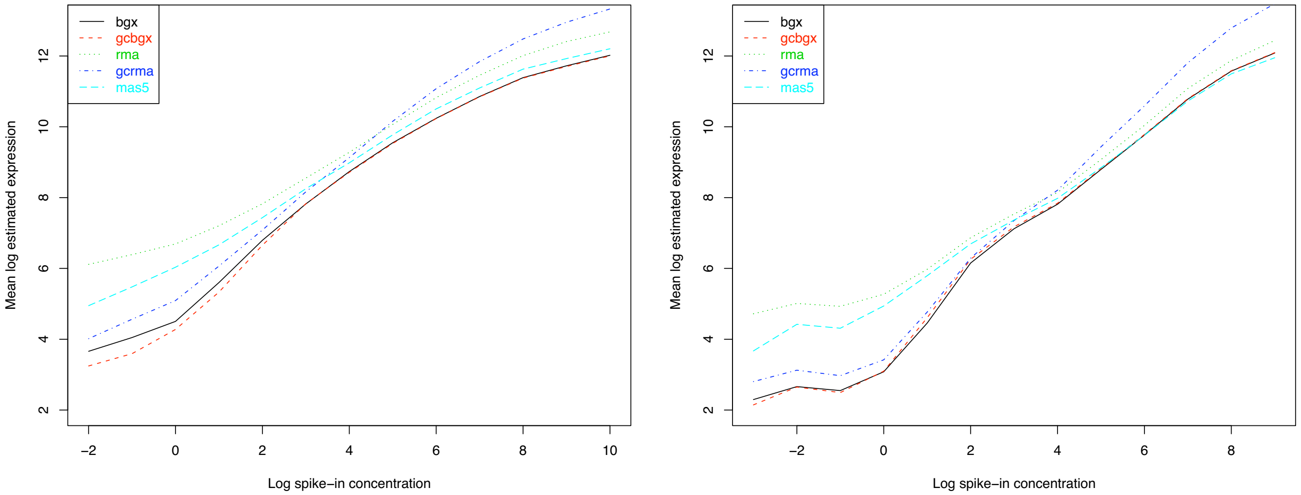

Figure 7
Performance on the Latin Square data. Log concentration vs. mean log estimated expression of spike-in genes in the HGU95A (left) and HGU133A (right) Latin Square data sets. There is an increased ability to detect concentration changes at levels lower than 4 pM in the HGU95A data set using GCBGX instead of BGX. Both BGX and GCBGX are more sensitive to changes within the low range than RMA or GCRMA and more sensitive to changes between 0.5 pM and 16 pM than MAS5. The MAS5 concentration line was shifted down by 2.5 units to facilitate the comparison of its gradient with those of the other lines.

butions of the differences in expression. The RMA, GCRMA and MAS5 expression measures were similarly adjusted using loess normalisation at the probeset level instead of the default quantile normalisation at the probe level.

Receiver operating characteristic (ROC) curves depict the observed false discovery rate vs. the true positive rate as the cut-off of a ranked gene list is varied. Figure 8 (left) shows average ROC curves for the nine single-array comparisons between condition C and condition S while Figure 8 (right) shows ROC curves for three-replicate comparisons. GCBGX has a small advantage over BGX and both models perform well. In the single-array comparisons, BGX and GCBGX outperform RMA, GCRMA and MAS5 for false discovery rates below 30%. The number of differentially expressed genes (DEGs) were estimated by running `plotDEHistogram` on the output of the nine comparisons involving one array from condition C versus one array from condition S. The number of genes detected as up-regulated with GCBGX ranged from 681 to 883 (mean 783) and for BGX from 560 to 867 (mean 742). Both methods produced an average true positive rate across the nine comparisons of over 97% for up-regulated genes. In the three-replicate comparisons, BGX and GCBGX outperform RMA, GCRMA and MAS5 for false discovery rates below 20%. An analysis of a three-replicate comparison yielded 1002 and 958 DEGs with 96.6% and 95.7% true positive rates using GCBGX and BGX respectively.

Conclusion

BGX is a new Bioconductor R package for analysing 3' Affymetrix GeneChips. BGX implements a fully integrated Bayesian hierarchical model with the option to take into account sequence-dependent probe affinities. BGX uses a novel adaptive MCMC algorithm that improves the efficiency with which the posterior distributions of parameters are sampled from. BGX compares favourably to RMA and GCRMA at detecting differential expression, particularly at low concentration levels.

Availability and requirements

Project name: BGX

Project homepage: http://bgx.org.uk

Operating systems: Platform independent

Programming language: C++, R

Other requirements: R, Bioconductor

License: GNU GPL

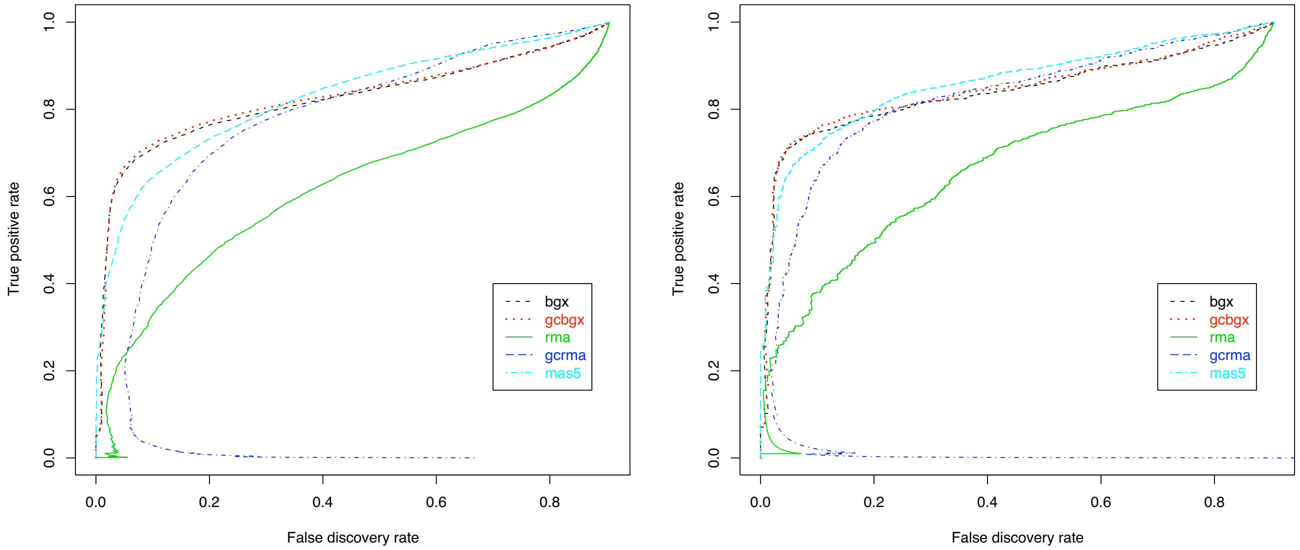

Figure 8
Performance on the Golden Spike data. Nine single-array comparisons (left) and three-replicate comparisons (right) between condition C and S were performed using BGX, GCBGX, RMA, GCRMA and MAS5. The ROC curves show that GCBGX has a small advantage over BGX and that both models perform well. For false discovery rates below 20%, BGX and GCBGX outperform RMA, GCRMA and MAS5.

Any restrictions to use by non-academics: No

Authors' contributions

ET implemented and tested the code and developed the adaptive MCMC. AKH and SR extended the BGX model to take into account probe affinity effects. NB and ET modelled and implemented probe affinity estimation of probes with missing sequences. ET, AKH, NB and SR provided comments and discussion and wrote the paper. All authors read and approved the final manuscript.

Additional material

Additional file 1

Varying the number of sampling iterations*. Plot of mean posterior values of μ_g obtained using a sampling length of 64 k iterations versus those obtained using sampling lengths of 16 k, 32 k and 64 k using an alternative seed for the pseudo random number generator. The values for the 32 k and 64 k comparisons are shifted upwards for clarity. Runs of more than 16 k confer a small increase in stability of estimation for non-expressed genes. At 32 k, estimates are as stable as between two 64 k runs using different seeds.*

Click here for file

[http://www.biomedcentral.com/content/supplementary/1471-2105-8-439-S1.pdf]

Additional file 2

R script to reproduce plots*. R script to reproduce the plots in this paper.*

Click here for file

[http://www.biomedcentral.com/content/supplementary/1471-2105-8-439-S2.r]

Acknowledgements

We thank Jeffrey S. Rosenthal, Gareth O. Roberts for advice on adaptive Metropolis-Hastings algorithms, and Alex Lewin and Marta Blangiardo for valuable discussion. This work was supported by BBSRC 'Exploiting Genomics' grant 28EGM16093 and the John and Birthe Meyer Foundation.

References

1. Hein AMK, Richardson S, Causton HC, Ambler GK, Green PJ: **BGX: a fully Bayesian integrated approach to the analysis of Affymetrix GeneChip data.** *Biostatistics* 2005, **6(3):**349-373.
2. Hein AMK, Richardson S: **A powerful method for detecting differentially expressed genes from GeneChip arrays that does not require replicates.** *BMC Bioinformatics* 2006, **7:**353.
3. Naef F, Magnasco MO: **Solving the riddle of the bright mismatches: labeling and effective binding in oligonucleotide arrays.** *Phys Rev E Stat Nonlin Soft Matter Phys* 2003, **68(1 Pt 1):**011906.
4. Wu Z, Irizarry RA, Gentleman R, Martinez-Murillo F, Spencer F: **A Model-Based Background Adjustment for Oligonucleotide Expression Arrays.** *Journal of the American Statistical Association* 2004, **99:**909-917(9).
5. **Bioconductor** [http://bioconductor.org]
6. Roberts GO, Rosenthal JS: **Coupling and Ergodicity of Adaptive MCMC.** *Journal of Applied Probability* 2005 [http://www.probability.ca/jeff/ftpdir/adapt.pdf]. [To appear]
7. Roberts GO, Rosenthal JS: **Examples of Adaptive MCMC.** 2006 [http://probability.ca/jeff/ftpdir/adaptex.pdf].
8. Roberts GO, Gelman A, Gilks WR: **Weak Convergence and Optimal Scaling of Random Walk Metropolis Algorithms.** *The Annals of Applied Probability* 1997, **7:**110-120.
9. Roberts GO, Rosenthal JS: **Optimal Scaling for Various Metropolis-Hastings Algorithms.** *Statistical Science* 2001, **16(4):**351-367.
10. **The R Project for Statistical Computing** [http://www.r-project.org]
11. **Boost C++ Libraries** [http://boost.org]
12. Choe SE, Boutros M, Michelson AM, Church GM, Halfon MS: **Preferred analysis methods for Affymetrix GeneChips revealed by a wholly defined control dataset.** *Genome Biol* 2005, **6(2):**R16.
13. Anderson T: *The Statistical Analysis of Time Series* New York: John Wiley and Sons; 1971.
14. Law AM: **Confidence intervals in discrete event simulation: a comparison of replication and batch means.** *Naval Res Logistics Quart* 1977, **23:**667-678.
15. **Affymetrix – Latin Square Data** [http://www.affymetrix.com/support/technical/sample_data/datasets.affx]
16. Cope LM, Irizarry RA, Jaffee HA, Wu Z, Speed TP: **A benchmark for Affymetrix GeneChip expression measures.** *Bioinformatics* 2004, **20(3):**323-331.
17. McGee M, Chen Z: **New Spiked-In Probe Sets for the Affymetrix HGU-133A Latin Square Experiment.** *COBRA Preprint Series* 2006.
18. Irizarry RA, Hobbs B, Collin F, Beazer-Barclay YD, Antonellis KJ, Scherf U, Speed TP: **Exploration, normalization, and summaries of high density oligonucleotide array probe level data.** *Biostatistics* 2003, **4(2):**249-264.
19. Yang YH, Dudoit S, Luu P, Lin DM, Peng V, Ngai J, Speed TP: **Normalization for cDNA microarray data: a robust composite method addressing single and multiple slide systematic variation.** *Nucl Acids Res* 2002, **30(4):**e15.