

Description of gcrma 1.2.0: changes and usage

Zhijin(Jean) Wu, Rafael Irizarry

July 20, 2005

Contents

1	Introduction	1
2	What's new	2
3	options for gcrma	2
4	Getting started: the simplest example	3
5	More Examples with options	4

1 Introduction

The *gcrma* package is part of the Bioconductor¹ project. The following terms are used throughout this document:

probe oligonucleotides of 25 base pair length used to probe RNA targets.

perfect match probes intended to match perfectly the target sequence.

PM intensity value read from the perfect matches.

mismatch the probes having one base mismatch with the target sequence intended to account for non-specific binding.

MM intensity value read from the mis-matches.

probe pair a unit composed of a perfect match and its mismatch.

affyID an identification for a probe set (which can be a gene or a fraction of a gene) represented on the array.

¹<http://www.bioconductor.org/>

probe pair set *PMs* and *MMs* related to a common *affyID*.

CEL files contain measured intensities and locations for an array that has been hybridized.

CDF file contain the information relating probe pair sets to locations on the array.

probe file contain the information relating probe sequences to locations on the array.

2 What's new

We added a function `bg.adjust.gcrma` for the convenience of performing `gcrma` background adjustment only, without summarizing into gene level data.

We have improved the speed of the function and therefore changed the default setting in function `gcrma` from `fast=TRUE` to `fast=FALSE`.

In earlier versions, we compute the sequence-determined probe affinities using parameters estimated from data obtained in our non-specific binding (NSB) experiment. In version 1.2.0 we give various options for the users to choose their own sources of such data. Users can choose to

1. compute probe affinities based on their own non-specific binding experiment (typically all probes in such experiments will contain non-specific binding, but the users can choose their list of probes)
2. compute probe affinities using each experimental array, with the user-defined negative control (NC) probes (MMs will be used if NC not specified).

3 Options for gcrma

Here we explain the major options in `gcrma`. Examples will be given in more details in the next section.

1. `affinity.info`

This is in general an `AffyBatch` object but you can leave it as `NULL` and let `gcrma` compute it automatically. Instead of storing probe intensities, it contains probe affinities.

When `affinity.info` is set to `NULL` (default), it will be computed within the function `gcrma`. To compute the affinity of each probe, we first obtain base-position profiles (the contribution of each base type at each position along the probe) from non-specific binding data. The user can choose to use the developers reference data or use each experimental array with indexes of negative control (NC) probes. If the NC probe index is not provided, MM probes will be used as NC probes.

Some users express the concern that their array type may behave differently from the human hgu95 array, which was used in the developers' non-specific binding experiment. The user can choose to run an independent non-specific binding experiment for her/his own research. The affinity.info can be obtained separately by

```
my.affinity.info <- compute.affinities.local(myNsbData)
```

and this should be passed to function gcrma

```
est<-gcrma(myExprData,affinity.info=my.affinity.info)
```

2. *type*

The options for *type* are

- *fullmodel*: uses both probe sequence information and observed MM probe intensities
- *affinities*: uses probe sequence information and ignores MMs
- *MM*: uses MM probe intensities and ignores sequence information

3. *fast*

When *fast* is set to *TRUE*, an ad hoc procedure is used to speed up the non-specific binding correction. The default in previous version has been *fast=TRUE*, but is changed to *fast=FALSE* in version 1.2.0.

4 Getting started: the simplest example

You will need the following libraries: *affy*, *MASS*, *cdf* and probe packages for your chip type (such as *hgu95av2cdf* and *hgu95av2probe*).

The first thing you need to do is **load the package**.

```
R> library(gcrma) ##load the gcrma package
```

If all you want is to go from probe level data (*Cel* files) to expression measures here are some quick ways.

The quickest way of reading in data and getting expression measures is the following:

1. Create a directory, move all the relevant *CEL* files to that directory
2. Start R in that directory.
3. If using the Rgui for Microsoft Windows make sure your working directory contains the *Cel* files (use "File -> Change Dir" menu item).

4. Load the library.

```
R> library(gcrma) ##load the gcrma package
```

5. Read in the data and create an expression, using RMA for example.

```
R> Data <- ReadAffy() ##read data in working directory
R> eset <- gcrma(Data)
```

5 More Examples with options

For illustration we will use the Dilution data.

```
R> require(affydata)
R> data(Dilution)
```

1. Obtain expression values For example,

```
R> Dil.expr1<-gcrma(Dilution)
```

To use the faster ad hoc procedure one can call

```
R> Dil.expr2<-gcrma(Dilution,fast=TRUE)
```

Suppose the user has his/her own NSB experiment and wants to compute affinity.info with that experiment.

```
R> myNsbData <- ReadAffy('mynsb.cel')
R> my.affinity.info <- compute.affinities.local(myNsbData)
R> Dil.expr3 <- gcrma(myExprData,affinity.info=my.affinity.info)
```

Suppose the user would like to use NC probes in the experimental data to estimate probe affinities. Here we use MM probes as example for NC probes.

```
R> mmIndex <- unlist(indexProbes(Dilution,"mm"))
R> Dil.expr4 <- gcrma(Dilution,affinity.source="local",NCprobe=mmIndex)
```

Since the MM probes are default setting when NCprobe is not provided, the above gives identical result as

```
R> Dil.expr5 <- gcrma(Dilution,affinity.source="local")
```

2. Background adjustment only The function `bg.adjust.gcrma` allows one to perform background adjustment only.

```
R> Dil.bgadj <- bg.adjust.gcrma(Dilution)
R> Dil.expr6 <- rma(Dil.bgadj,background=FALSE)
```

`gcrma` also tries to adjust for specific binding using probe sequence. The user can turn off this feature by specifying *GSB.adjust=FALSE*.

```
R> Dil.bgadj <- bg.adjust.gcrma(Dilution,GSB.adjust=FALSE)
```