

RMAExpress Users Guide

1.0.5 Release

<http://rmaexpress.bmbolstad.com>

B. M. Bolstad
bmb@bmbolstad.com

May 22, 2010

Contents

1	Introduction	3
1.1	What is RMAExpress?	3
1.2	Installing the software	3
2	RMAExpress: the main application	8
2.1	Exploring RMAExpress	8
2.2	Setting the preferences	9
2.3	Loading in data	10
2.4	Computing Expression Values	12
2.5	Visualizing the Raw Data	14
2.6	Quality Assessment	17
2.6.1	Residuals Image	17
2.6.2	PLM based quality assessment	19
3	RMADDataConv: the data converter	26
3.1	The main dialog	26
3.2	Converting a set of CEL or a CDF file to RME format	27
3.3	Restricting the set of probesets used	28
3.4	Turning PGF and CLF files into CDFRME files for Exon Array Analysis	28
3.4.1	Using PS files	28
3.4.2	Using MPS files	29
3.5	Merging MG_U74A and MG_U74Av2 datasets	29
4	RMAExpressConsole: the console application	30
4.1	How long will it take?	31
4.2	Examples	32
A	Reference Material	34
B	Building RMAExpress from source code	35
B.1	Building native binaries for Linux	35
B.2	Building native binaries for Windows	35
B.3	How to install a cross-compiler on a Linux machine to produce RMAExpress Windows binaries.	35
B.4	How to build RMAExpress on a Windows machine using MinGW	36

C	Brief changelog/history	38
D	File format information	40
D.1	Binary format output file	40

Chapter 1

Introduction

This document is intended as a introductory guide to the RMAExpress 1.0.5 Release applications. It does not give great details on the underlying algorithms or their implementations. For such materials, the reader is referred to the publications at the conclusion of this document. Although RMAExpress is written to be cross platform, this documentation will concentrate on the Microsoft Windows binaries. The appearance of windows and menus shown in this document may be different on your operating system.

1.1 What is RMAExpress?

RMAExpress is a cross-platform program which provides methods for producing RMA expression values from Affymetrix CEL files. It's focus is on arrays used for expression analysis. In particular, 3' IVT expression arrays, Exon arrays and the WT Gene arrays can all be processed by RMAExpress.

There are three main applications making up this software package

- **RMAExpress**: see chapter 2
- **RMADDataConv**: see chapter 3
- **RMAExpressConsole**: see chapter 4

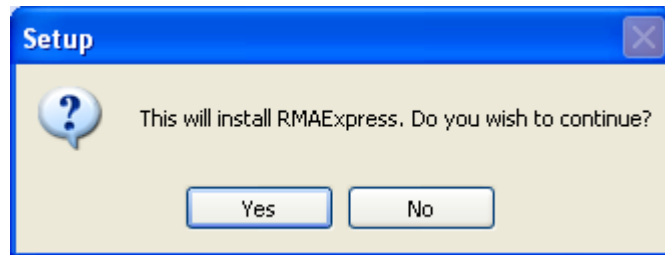
Most users will only need to use the first of these applications, although documentation is provided about all three in this users guide.

It should be noted that RMAExpress is open source software distributed under the GPL <http://www.gnu.org/copyleft/gpl.html>. Most users will not need to compile their own version of the software and will just use the pre-compiled binaries supplied. However, source code is available at the RMAExpress website and instructions for building it may be found in appendix B.

1.2 Installing the software

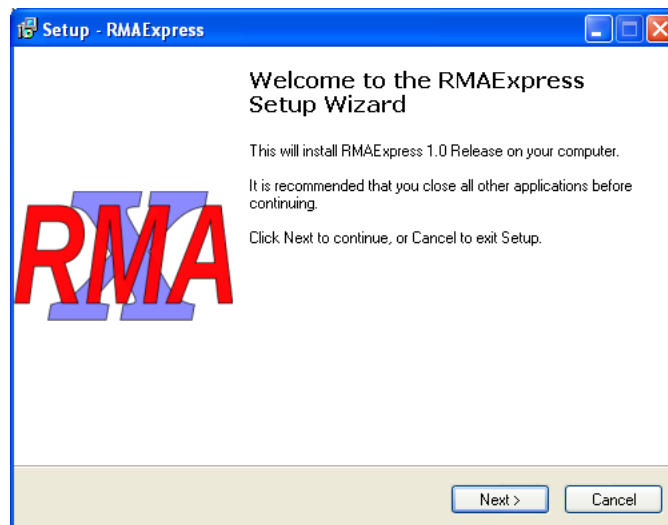
To install this software onto a Microsoft Windows operating system first download the installer from the website. Double clicking on the installer binary should open the installer. Installation proceeds as follows:

1. The first dialog presented to you should resemble the following:



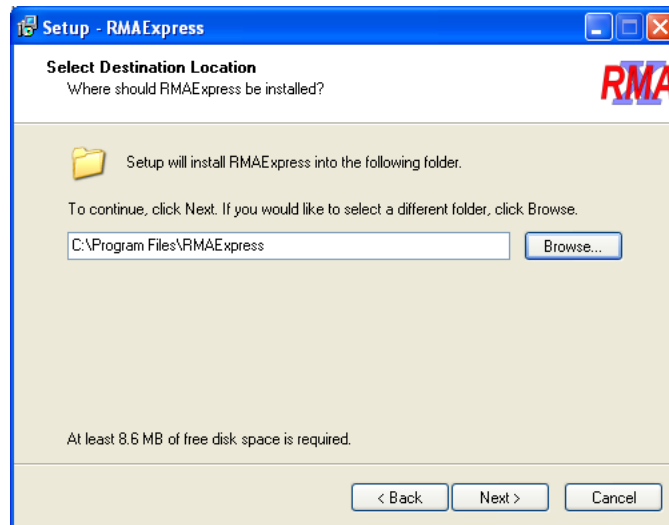
You should click *Yes* on this dialog. Answering *No* will close the installer and RMAExpress will not be installed.

2. If you clicked *Yes* then the following dialog is displayed:



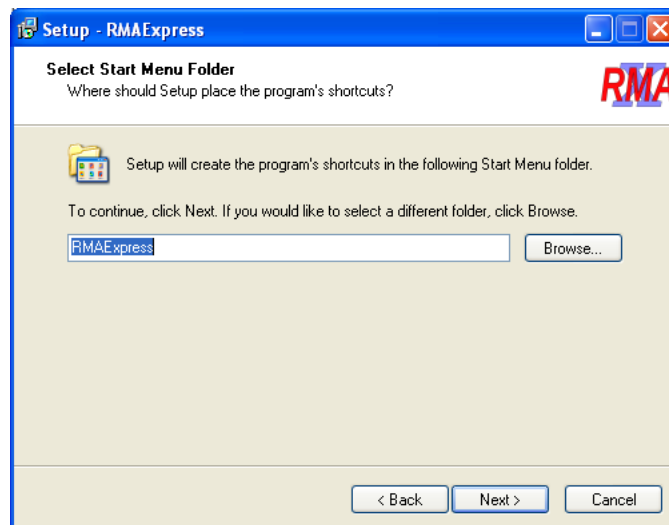
To continue installing click *Next*. Choosing *Cancel* will quit the installer with no further action taken.

3. The next stage of the installation is to choose where the application will be installed on your system. This is done using this dialog:

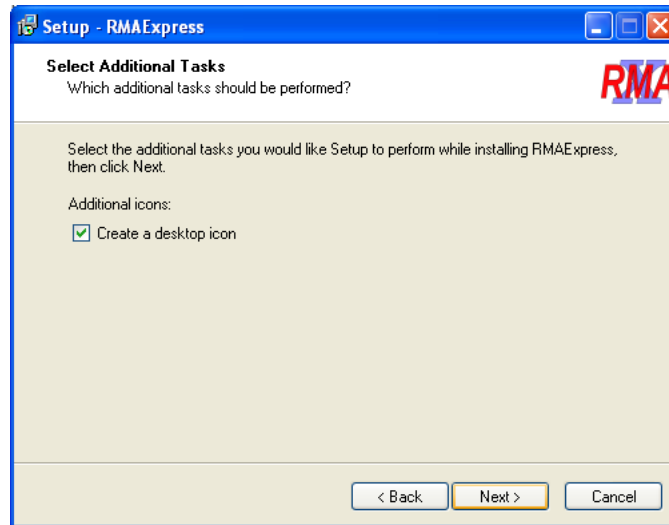


By default it will use C:/Program Files/RMAExpress but you may change that to a different location on your system. Click *Next* to continue. As before, *Cancel* will quit the installer with no further action taken.

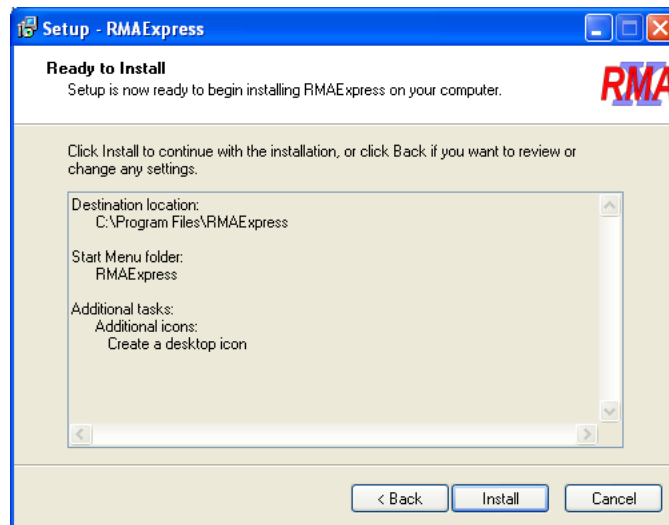
4. Now the installer wants to know where you want to put the applications in the Start Menu. By default a new group called RMAExpress will be created for the Start menu. It is probably best to just click *Next* here:



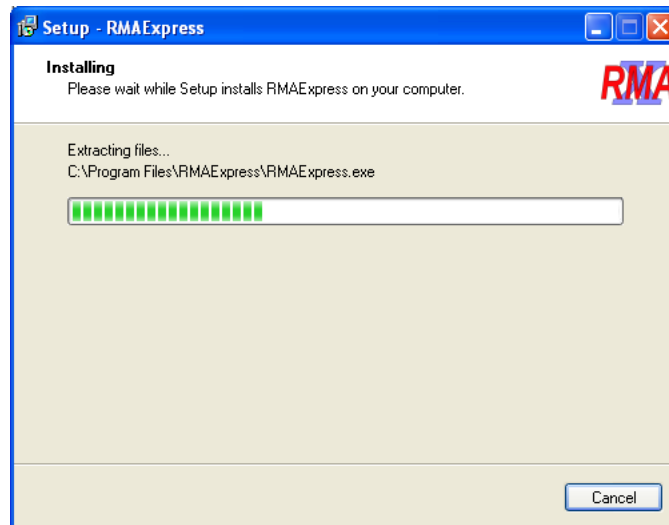
5. The next option you are given is whether or not you want to create desktop icons for RMAExpress and RMADataConv. By default the box is checked. If you do not want these icons to be created unclick the check box.



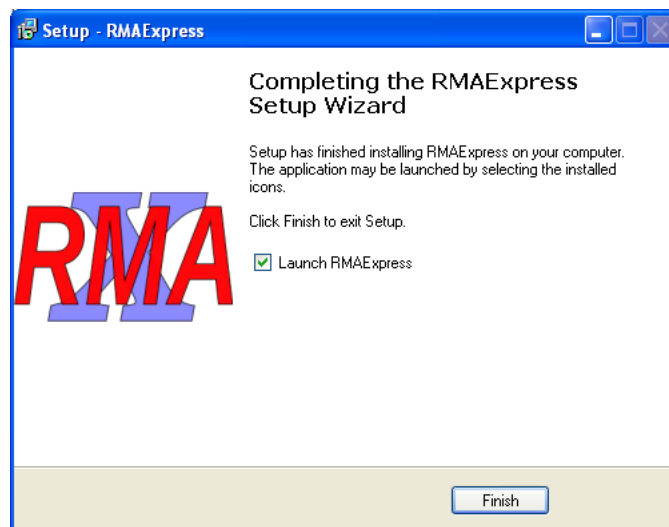
6. Finally you are given a brief outline of what the installer will do. Clicking *Install* will commence the installation program. Choosing *Cancel* will quit the installer with no further action taken. Note that this is the last possible time to quit the installer without installing the program.



7. A progress bar will show how the installation process is proceeding.



8. After the program is installed a confirmation message is displayed to tell you that the installation completed successfully. If the box is checked then clicking on *Finish* will close the installer and launch RMAExpress. If you do not wish to launch RMAExpress immediately then uncheck the box before hitting close.



At this point installation is completed. If you have chosen to install icons they should now be available on your desktop to launch RMAExpress when needed.

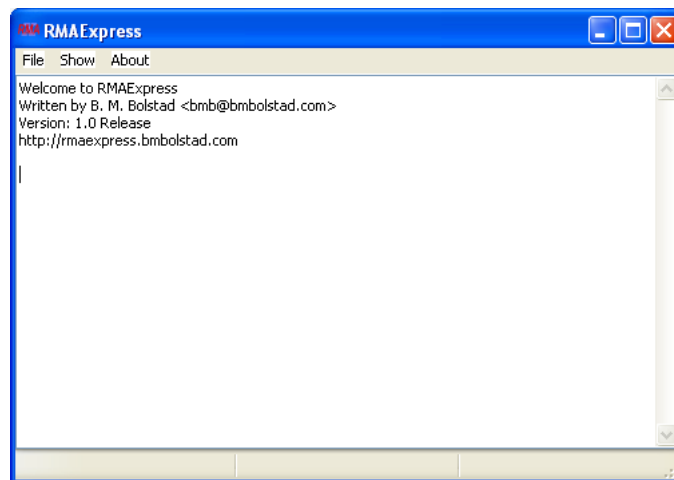
Chapter 2

RMAExpress: the main application

RMAExpress is the main application in the software package. It is the primary, and only, application in the collection that most users will need to use. The primary functionality is to read raw CEL files and produce RMA expression values.

2.1 Exploring RMAExpress

When you first open RMAExpress you will see a window that looks like the following:



This application consists of a main window where a log of events is displayed and several menus from which specific commands can be issued. The *File* menu is where most commands are issued. It includes commands for: reading in data, computing RMA expression values and writing the results to text files. The *Show* menu provides functionality for viewing data, examining residual images and generating quality assessment values. The *About* menu gives a dialog box showing the RMAExpress version number.

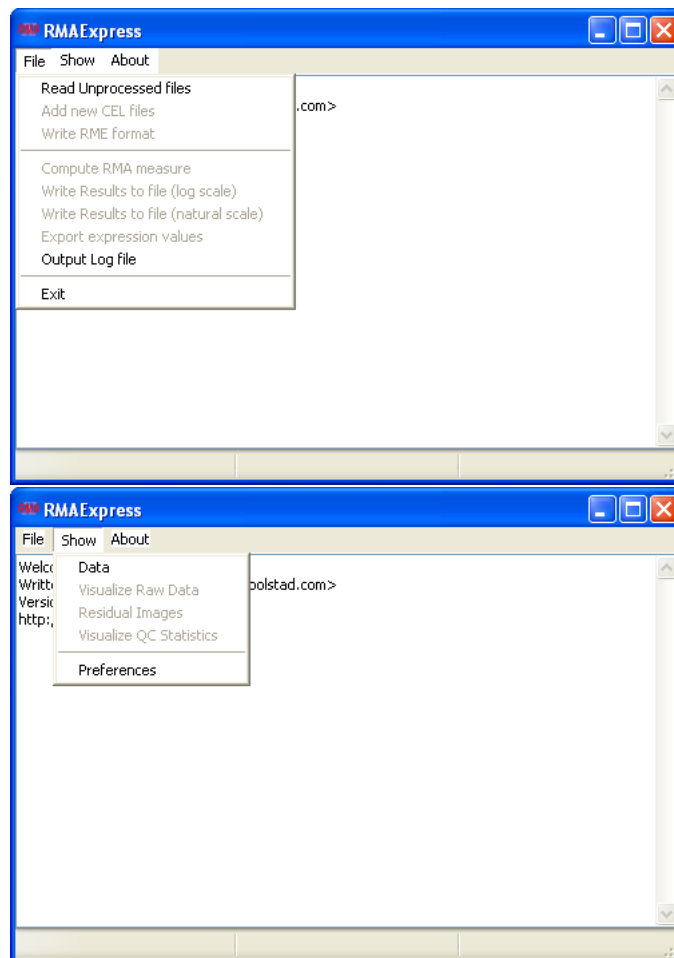
When the application is initially launched many menu options are not available. In the *File* menu the available options are:

- *Read Unprocessed files*: For reading in raw CDF and CEL file data.
- *Output Logfile*: For saving all the messages that appear in the window to a text file.
- *Exit*: For quitting the application.

In the *Show* menu the available options are

- *Data*: Which outputs to the main window text messages stating what data is currently loaded into the application.
- *Preferences*: For setting some preferences.

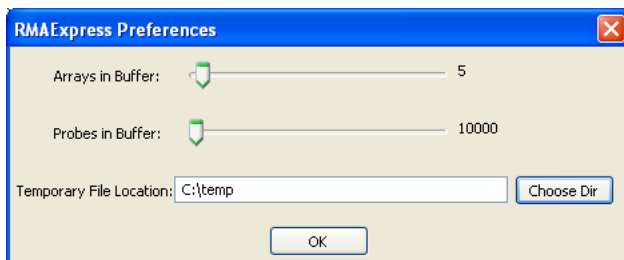
These two screen-shots show how the menus appear when the program is first launched:



2.2 Setting the preferences

Before you do anything else it is a good idea to set the preferences for the application. To allow the processing of large datasets RMAExpress 1.0.5 Release buffers data in and out from main memory to

disk. The user has some control over this buffering activity, specifically where temporary files are stored and how large the memory buffer is. Choosing *Preferences* from the *Show* menu will bring up the preferences dialog which will show the current settings:



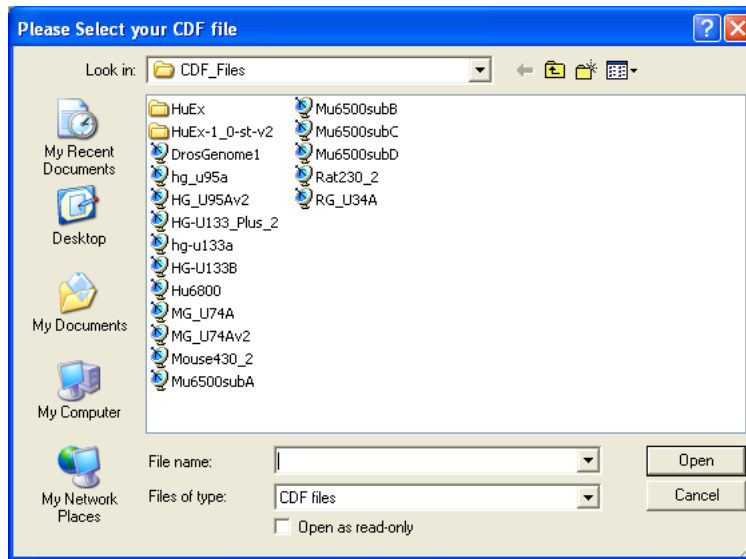
Two sliders control how much data is kept in active memory at any one time. The first slider can be used to control how many arrays will be kept completely in memory. That is every probe intensities for the specified number of arrays will be stored in memory. The second slider controls a slightly different memory buffer. This memory buffer stores a specified number of probes across all chips. For instance if you had this set to 10000 and a dataset with 200 arrays then probe intensities for 10000 probes for all 200 arrays would be kept in memory, irrespective of what arrays were in the first buffer.

Note that unless you have extremely large amounts of memory you should be conservative in your buffer settings and keep them reasonably sized, since increasing the buffer sizes too much may lead to decreased performance. For users of 32 bit windows operating systems best performance is usually achieved by setting the buffer values to their minimums. Increasing these values may actually cause slow downs in performance. Users of 64 bit operating systems can be more aggressive in their buffer settings.

The final choice the user should make in this dialog box is to specify the location where temporary files should be stored. The user may either type in the full path or click the *Choose Dir* button and navigate to the location where temporary files will be written. The chosen location should provide large amounts of disk space. It is very important that the user has read/write permissions to the location chosen. Note that any temporary files created will automatically be deleted when RMAExpress exits.

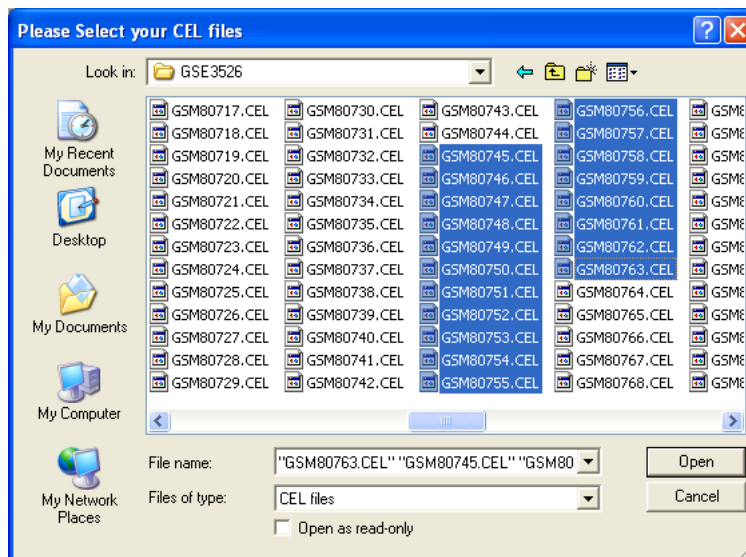
2.3 Loading in data

The user has two options in the *File* menu for loading data. Most users will wish to read data from raw CDF and CEL files. Choosing *Read Unprocessed files* opens this dialog:

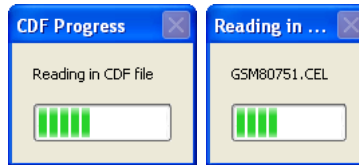


The user should navigate to the location that the appropriate CDF file is stored, select the appropriate CDF file and then click *Open*. Clicking *Cancel* will stop the process and no data will be loaded. Note that CDFRME (see the RMADataConv documentation) files may also be used.

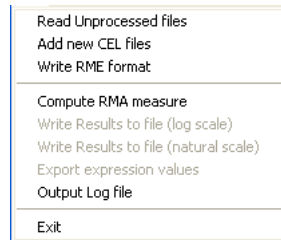
Next the user should select the CEL files that they wish to load, navigating to and selecting them using this dialog:



After the appropriate CEL files have been selected click *Open* to begin the process of reading the data into RMAExpress. A series of progress bars will appear on screen letting you know how the process is proceeding.



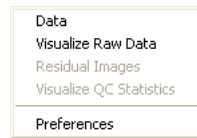
Once the data has been read in several new options are now available for use in the file menu:



Specifically,

- *Add new CEL files*: For reading additional CEL files that were missed in the initial read or stored in another directory.
- *Write RME format*: Output all the currently loaded CEL data in RME files (these are the same format as that produced by RMADataConv).
- *Compute RMA measure*: This option begins the process of computing RMA expression values from the read in data.

Also, an additional option may now be chosen from the show menu.



This item is

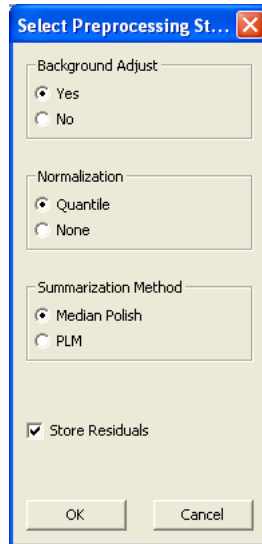
- *Visualize Raw Data*: For looking at boxplots and density plots of the raw unadjusted intensities.

More details about this option can be found in section 2.5.

At this point the user should either proceed with their analysis or read additional CEL files using the *Add new CEL files* option.

2.4 Computing Expression Values

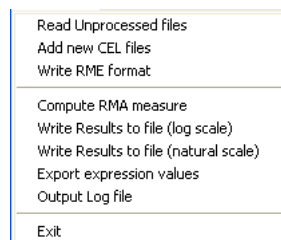
After data has been read into RMAExpress choosing *Compute RMA measure* from the *File* menu will the following dialog box to the user:



Using this dialog box the user can choose which preprocessing steps to carry out and whether or not to keep the residuals from the modeling procedure. Most users should keep the default background correction (*YES*) and normalization (*Quantile*) options selected. For the summarization step the user has two choices: *Median Polish*, which is the default, and *PLM*. These both fit the same summarization model in a robust manner, but do it in different ways. The median polish algorithm is the faster of the two options, and is what gives RMA expression values. Using PLM, which is an abbreviation for *probe-level model* will be slower, but it will allow you to examine the QC statistics described in section 2.6.2. You will not be able to examine these QC quantities if you chose to use the median polish. Checking the *Store Residuals* check box will make it possible to visualize chip pseudo-images for quality assessment purposes. These residual images plots can be generated no matter which summarization method was selected. Clicking *OK* will start the procedure of computing expression measures. Clicking *Cancel* will halt the process.

While RMAExpress is computing expression values a series of dialogs will appear to keep the user updated on progress. It may take some time for this procedure to finish.

When processing is finished, additional options are now available in the menus. All options should now be available in the file menu and additional options may have become available in the show menu.



In the *File* menu new options are:

- *Write Results to file (log scale)*: Output the computed RMA expression values to a text file.
- *Write Results to file (natural scale)*: Output natural scale RMA expression values to a text file.
- *Export expression values*: Export the computed RMA expression values to a binary format file.

Note that traditionally RMA expression values are used and expressed in the \log_2 scale. However, some external analysis programs may only accept natural scale values. It is for this reason that two different methods which output to text files have been provided. Details about the binary file format can be found in the appendix. The function `ReadRMAExpress` in the BioConductor *affyPLM* package will read this output file into R.

In the show menu

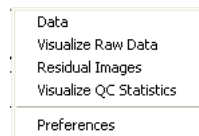


one new option may be available, provided the user chose to store the residuals when computing RMA expression values by clicking the check box in the pre-processing options dialog. Specifically,

- *Residual Images*: View chip pseudo-images of the residuals

The Residual Images option is explained in greater detail in section 2.6.1 of this users guide.

If the user chose the *PLM* summarization option then another option will become user selectable from the *Show* menu

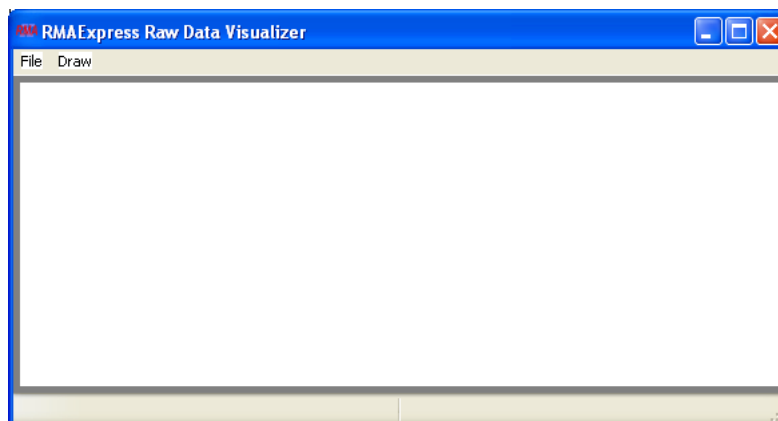


- *Visualize QC Statistics*: Allows you access to the PLM based NUSE and RLE statistics.

More details about this option can be found in section 2.6.2.

2.5 Visualizing the Raw Data

Choosing *Visualize Raw Data* from the show menu opens the RMAExpress Raw Data Visualizer window. This window looks like this:



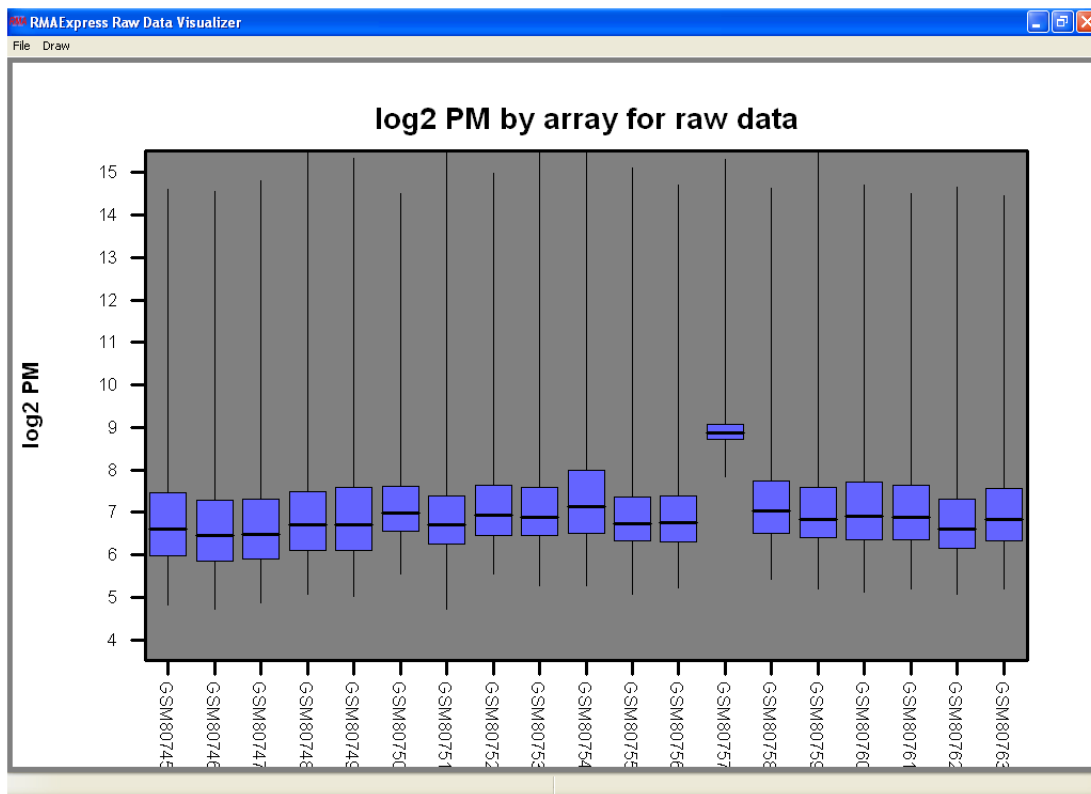
It has two menus. A *File* menu and a *Draw* menu:



The *File* menu options are:

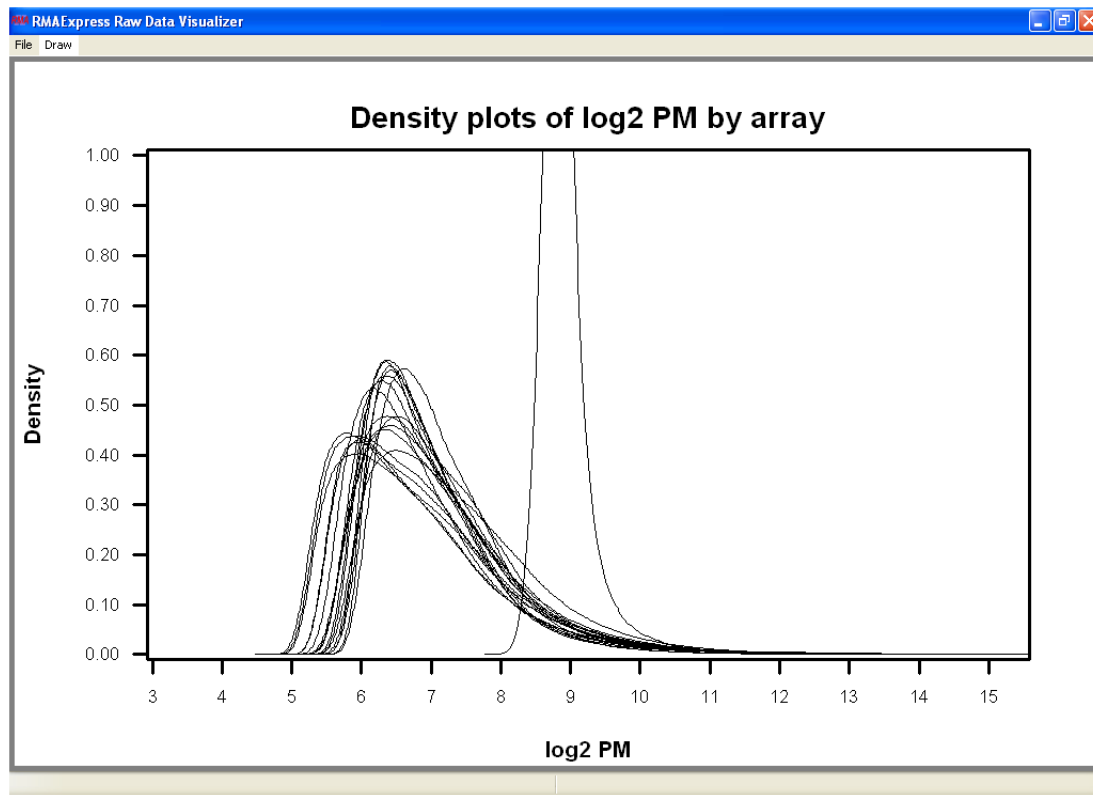
- Save: Save the current image to a file
- Print: Print the current image
- Exit: Close the Raw Data Visualizer and return to the main RMAExpress window

The first option in the *Draw* menu is *Boxplots*. Selecting this option will draw boxplots of the unadjusted PM intensities, one for each array. For better visualization the PM intensities are \log_2 transformed. The following screenshot shows a typical set of boxplots produced :



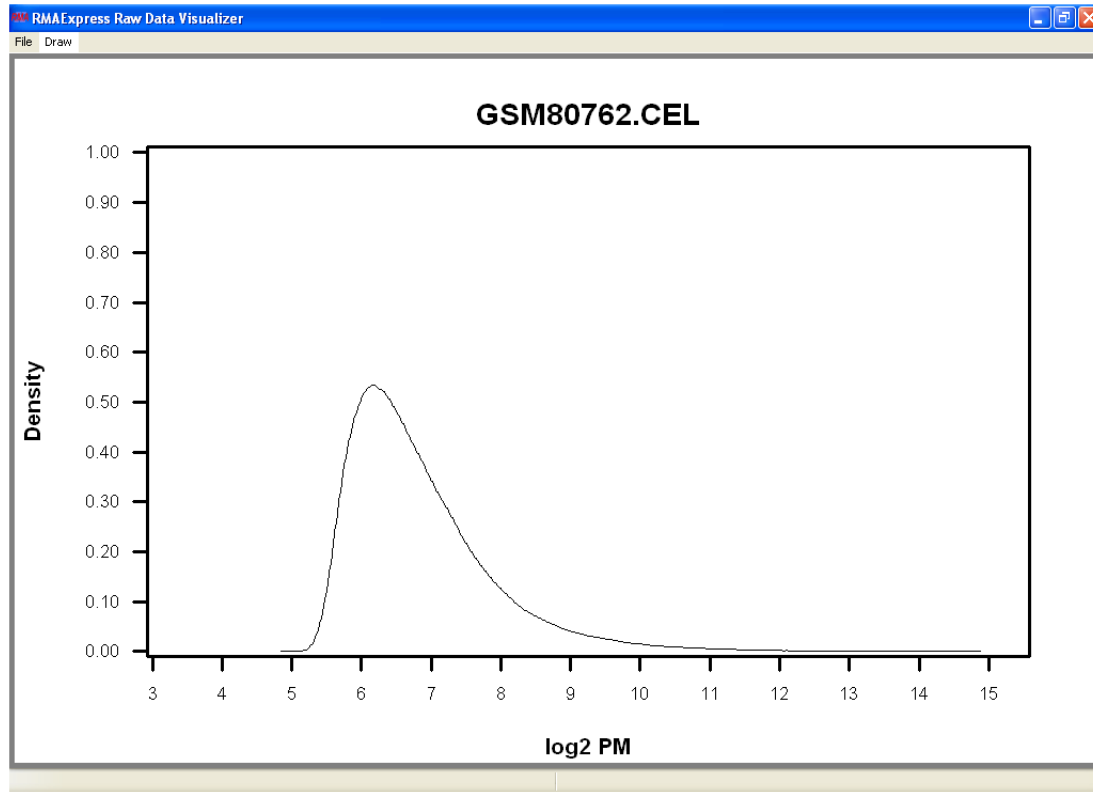
Notice that one array was significantly brighter than the other arrays in this dataset.

The second option in the *Draw* menu is *Density Plots*. Choosing this option produces smoothed density curves of the \log_2 PM intensities, with one curve drawn for each array. A typical set of density plots looks like this:

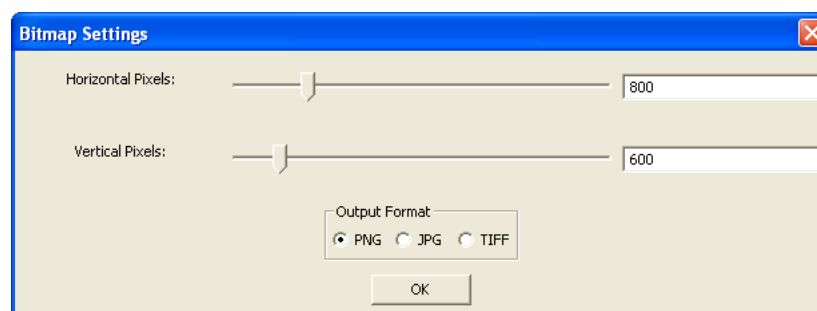


Notice that one array clearly stands out as being different. This may indicate that the data from this array is of poor quality, or it may be that normalization can correct this effect. When using density plots, potential low quality data is often indicated by density curves that are shifted away from the main set of curves or differently shaped.

Because it is difficult to discern which curve belongs to which array the *Draw* menu provides a third option *Individual Density Plots*. This option produces a density plot for a single array. The user may cycle through the arrays by using the up or down arrow keys. On some platforms the page up and page down buttons may also be used for this purpose.



If the *Save* option is selected from the *File* menu the user will be asked to choose the dimensions and file format of the output file. This is achieved using the following dialog box:

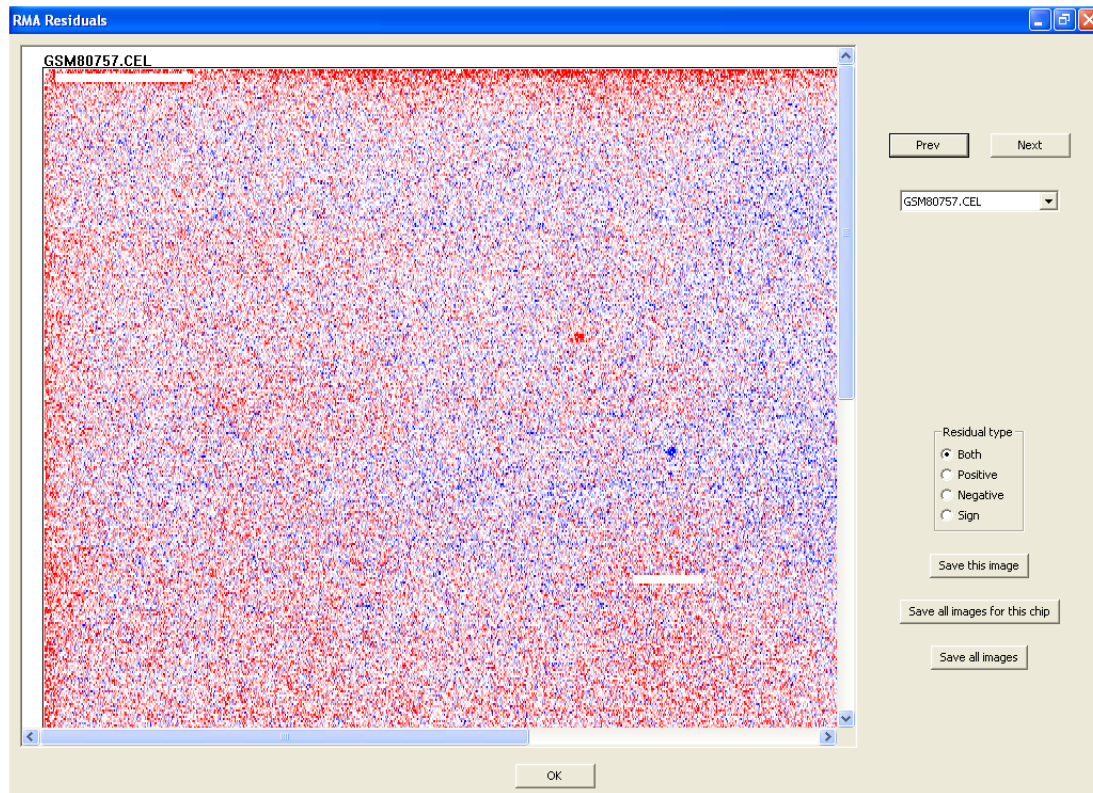


2.6 Quality Assessment

2.6.1 Residuals Image

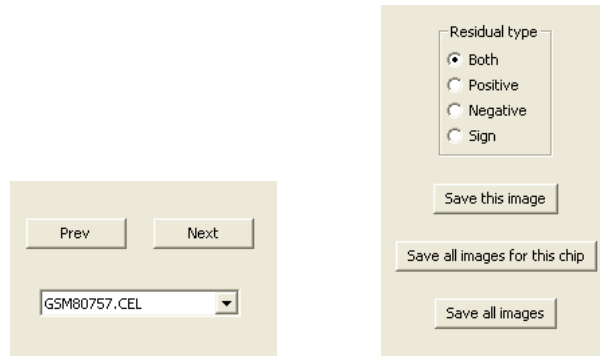
If the user chose to store residuals when computing expression values then it is possible to visualize the residuals from the RMA procedure on a chip by chip basis. Selecting *Residuals Images* from the *Show*

menu opens the following dialog box:



Chip pseudo-images are displayed in the main pane of this dialog box. Red is used to denote highly positive residuals and blue to denote low negative residuals. White is used for residuals near 0. The intensity of the red or blue designates how far from 0 the residual is. Poor quality data typically has large intense patches of a single color in distinct regions. In the image shown in this user guide doesn't have any specific artifacts, but because has such intense reds and blues, with little white, it is clear that it is of lesser quality. This user guide is insufficient in length to fully explain how to interpret these images. To get a better feel for typical images, both of good and poor quality, the user is referred to <http://PLMImageGallery.bmbolstad.com>.

The user can select which array to visualize by using the *Prev* and *Next* buttons or by selecting the array by name using the combo box. Radio buttons control which type of image the user is shown. The default setting is *Both* which means that both positive and negative residuals are shown. Selecting *Positive* or *Negative* will show only the red or blue parts of the image respectively. Choosing *Sign* means ignore the magnitude of the residual and just color by the sign.

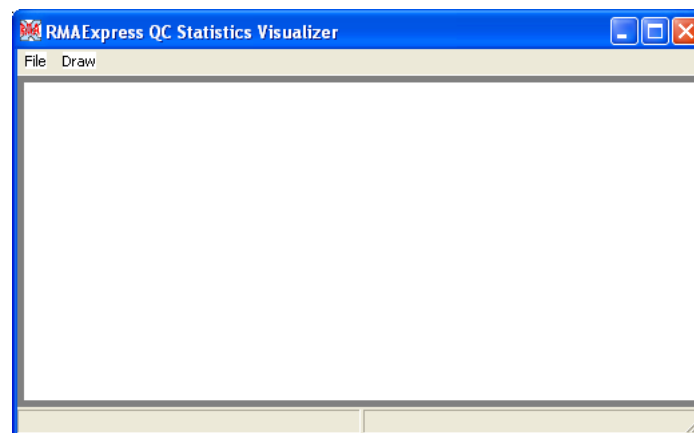


Three buttons can be used to save residuals images. The first *Save this image* saves the image that is currently drawn. Pushing *Save all images for this chip* saves the residuals, positive residuals, negative residuals and sign of residuals images for the current array. Finally using *Save all images* produces all of the images for all of the arrays.

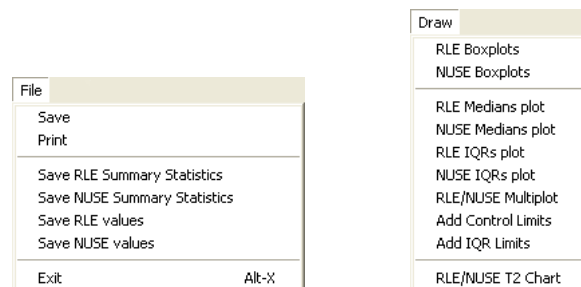
Clicking the *OK* button will close the Residual images window.

2.6.2 PLM based quality assessment

Choosing *Visualize QC Statistics* from the show menu opens the RMAExpress QC Statistics Visualizer window. This window looks like this:



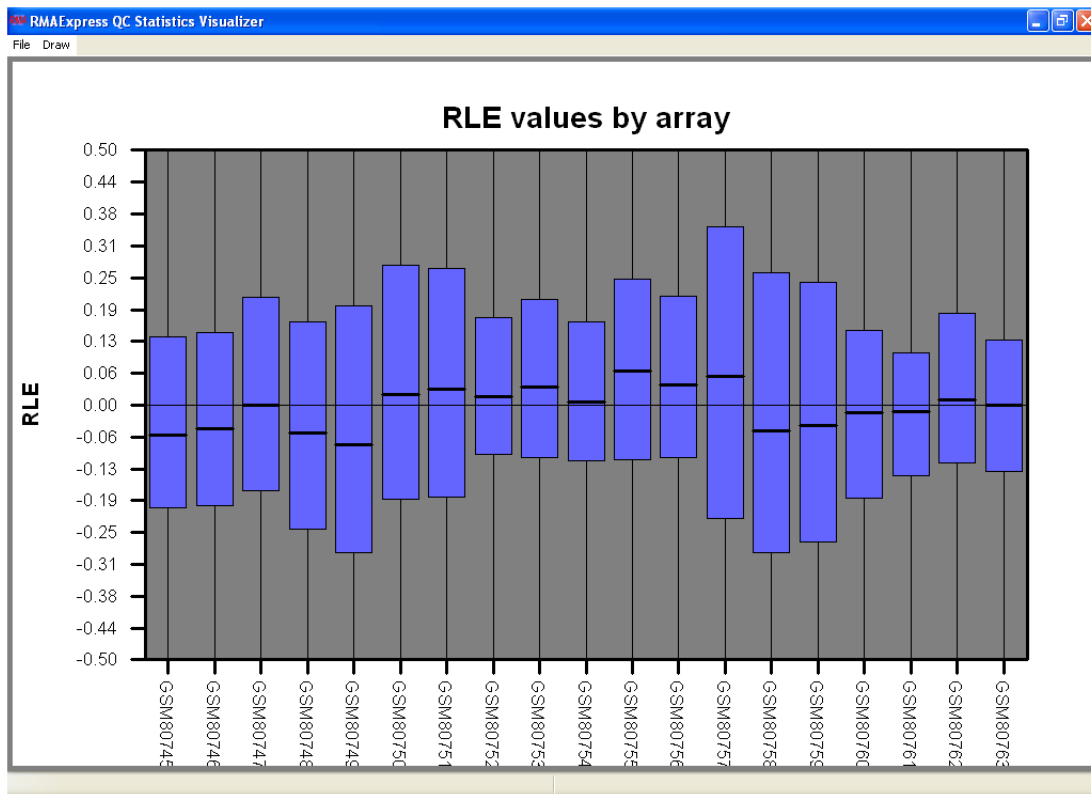
It has two menus. The *File* menu is for saving output to files and *Draw* menu for selecting possible plots. Both revolve around the two main PLM based quality statistics: Normalized Unscaled Standard Error (NUSE) and Relative Log Expression (RLE). These are the current options in the menus:

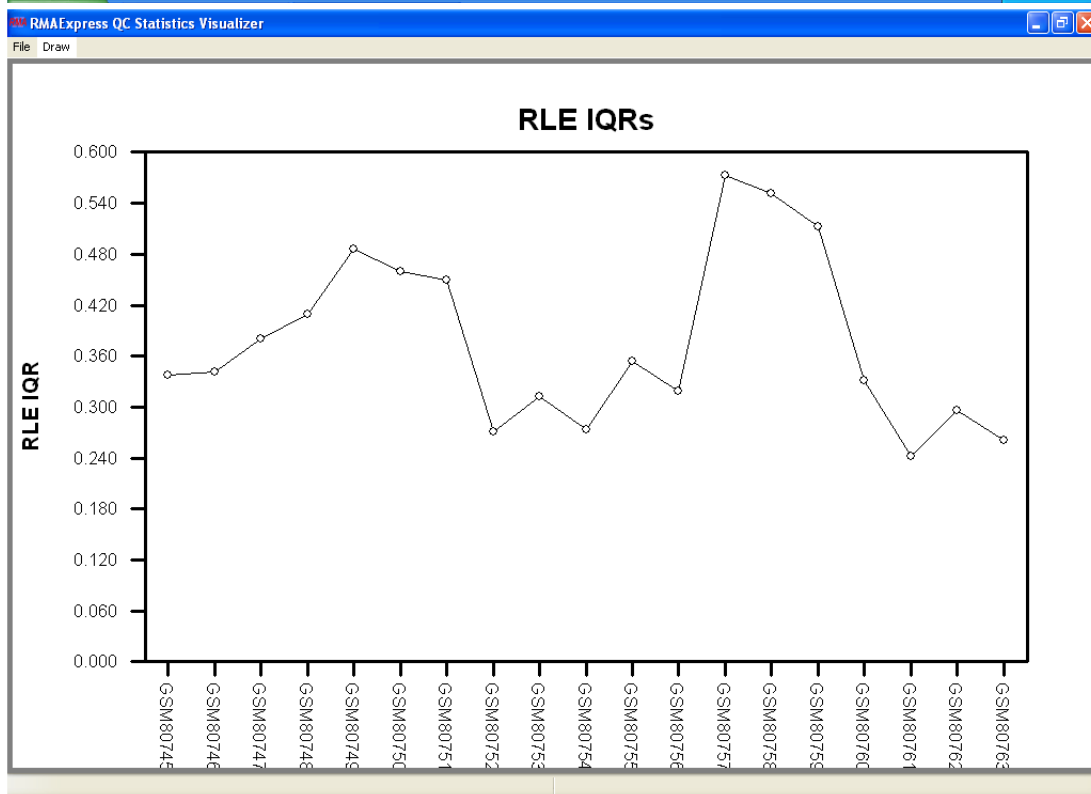
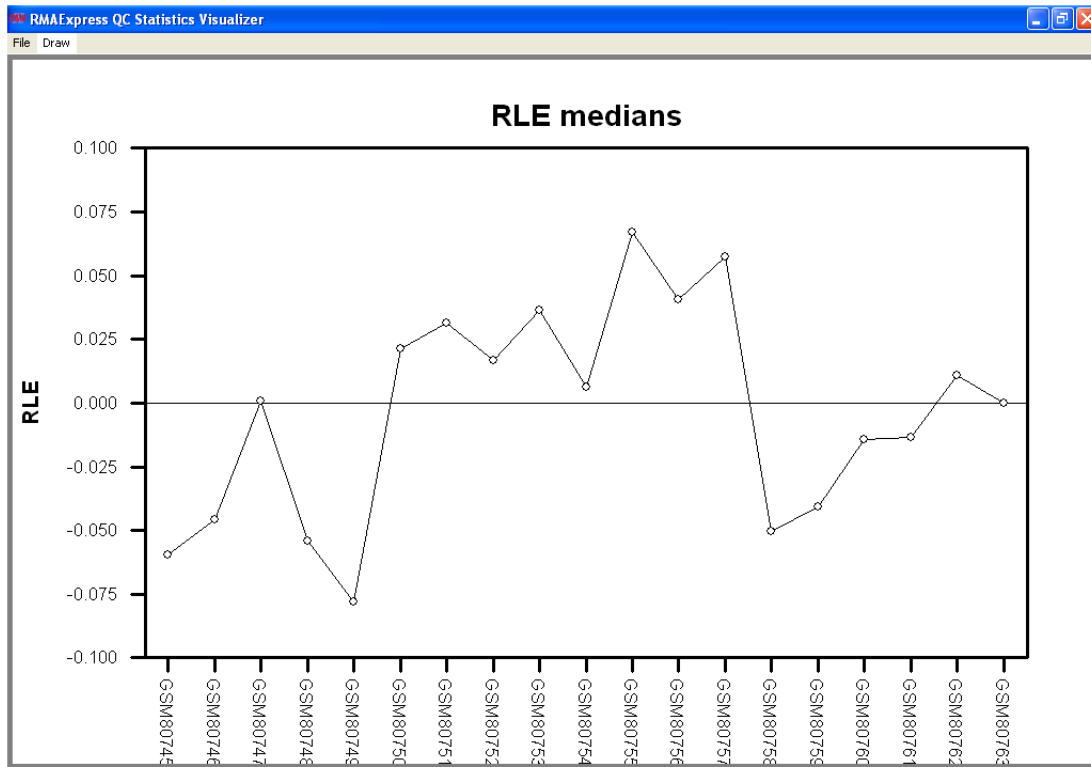


The *File* menu options are:

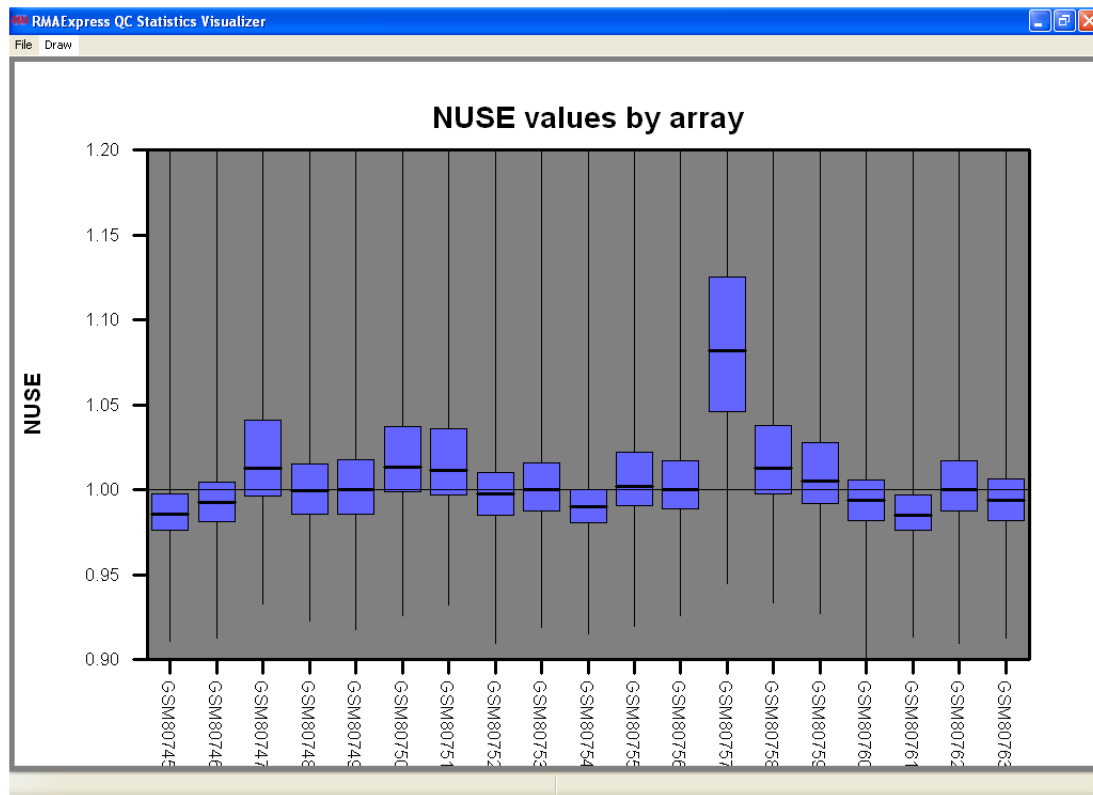
- Save: Save the current image to a file
- Print: Print the current image
- Save RLE Summary Statistics: Save a summary table to tab delimited text file of the RLE values
- Save NUSE Summary Statistics: Save a summary table to tab delimited text file of the NUSE values
- Save RLE Values: Save RLE values to a tab delimited text file
- Save NUSE Values: Save NUSE values to a tab delimited text file
- Exit: Close the QC Statistics Visualizer and return to the main RMAExpress window

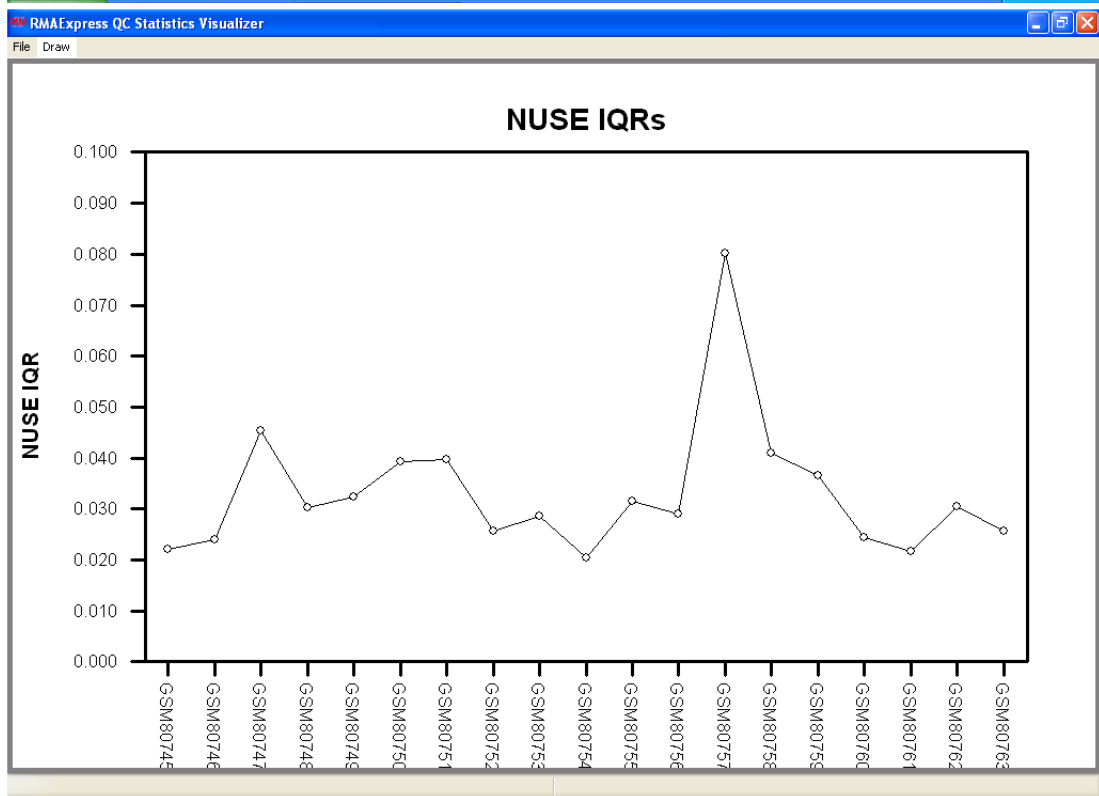
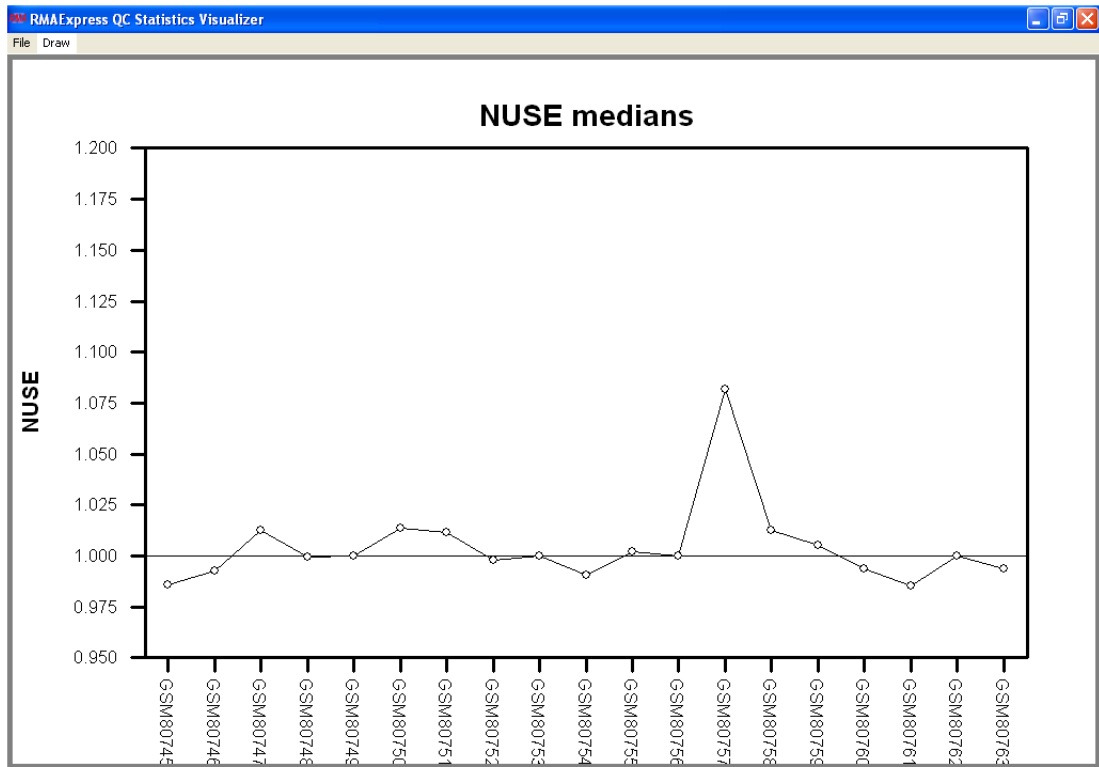
There are three different options in the *Draw* menu for visualizing the RLE statistic. The first *RLE Boxplots* draws boxplots of RLE values, one for each array. Low quality arrays will have greater spreads, or will not be centered near 0. A closer look at the medians can be found by looking at *RLE Medians plot*. A third plot shows the RLE IQRs. The following three screenshots show these plots.





The *Draw* menu also provides three similar options for the NUSE statistic. The first *NUSE Boxplots* draws boxplots of the NUSE values. Low quality data will have greater spread or be not centered around 1. The following three screenshots show these plots.

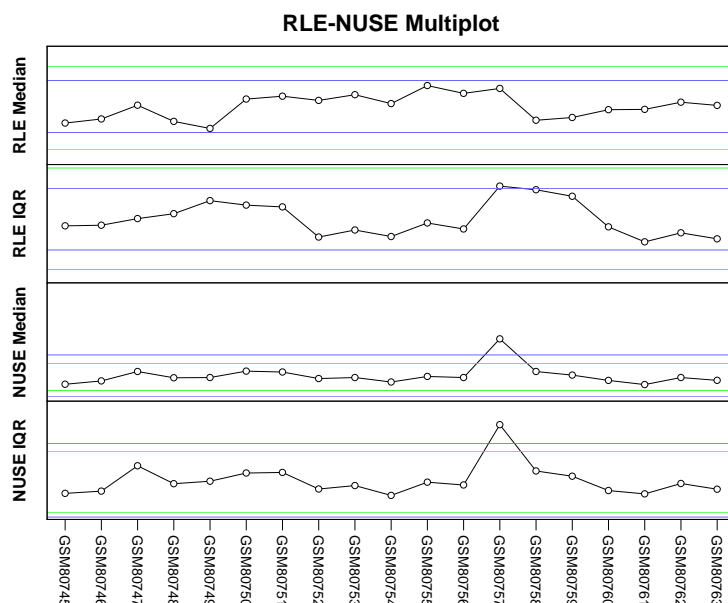




One array is clearly of lower quality based on the NUSE plots. This corresponds to the same array discussed in the residuals images section, and also shown in the visualizing raw data section of this user guide.

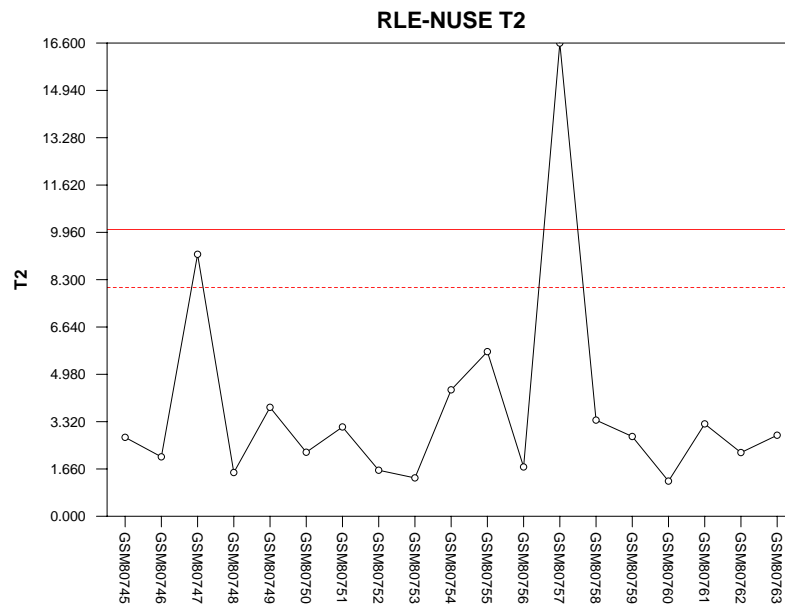
RMAExpress also provides QC cutoffs that may be applied to the RLE and NUSE single summary plots. These may be triggered by selecting the *Add Control Limits* and *Add IQR Limits* options in the *Draw* menu. The *Add Control Limits* option produces upper and lower control limits that are derived using the methodology for XmR control charts. Blue lines are used to indicate these control limits. The *Add IQR Limits* option produces control limits derived based on normal boxplot outlier identification rules. Specifically, the limits are at $1.5 \times \text{IQR}$ above the upper quartile and $1.5 \times \text{IQR}$ below the lower quartile. The IQR Limits are drawn using green lines. Note that the user should use the control limits when attempting to identify lower quality arrays, but there are not hard boundaries and should only be considered indicative of arrays for further investigation. Note that the control limit options require the dataset include at least 6 arrays.

In most datasets there is some degree of correlation exists between the RLE and NUSE summary values. The *RLE/NUSE Multiplot* option in the *Draw* menus combines plots for the four QC summary values together into a single plot. An example plot is shown here (with both types of control limits shown).



One array falls well outside the control limits for both NUSE metrics and is borderline on the RLE IQR plot.

Another QC method is to reduce the multiple RLE and NUSE summaries down to a single number. The *RLE-NUSE T2* is one such multivariate statistic. This option is only available when 6 or more arrays are analysed together. The following shows the plot created when the *RLE-NUSE T2* option is selected.



Two control limits are shown on this plot. A red dotted line indicates a 95% cutoff and the solid red line is a 99% cutoff. Expression values for arrays exceeding the 99% cutoff should be removed from down stream analysis. Arrays which only exceed the 95% cutoff warrant further investigation.

Note that the control limits drawn by RMAExpress in the QC statistic visualizer work best when 20 or more arrays are analyzed together.

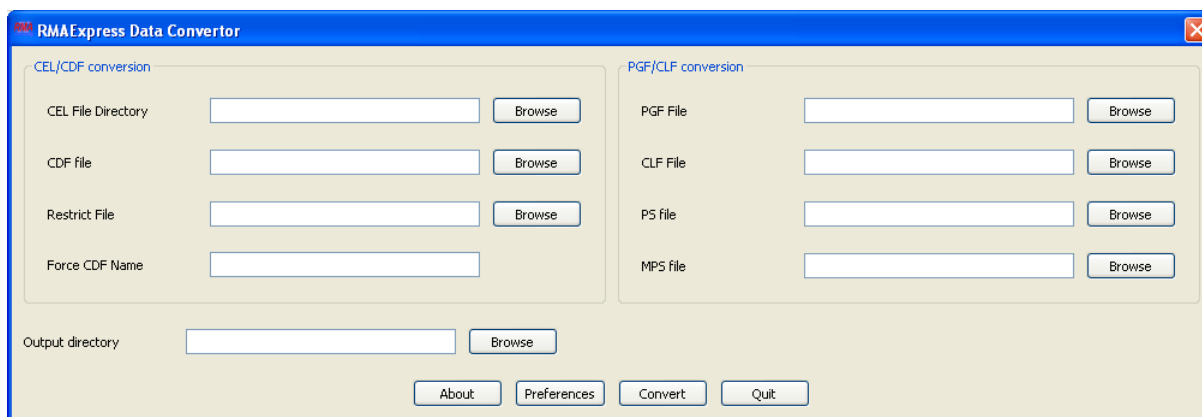
Chapter 3

RMADDataConv: the data converter

RMADDataConv is an application for converting CDF and CEL files to an intermediate format that may be read into RMAExpress. Most users will not need to use it since the RMAExpress application supports CDF and CEL files natively. However, the data converter is still useful in some circumstances. This section explains the RMADDataConv application and some of its uses.

3.1 The main dialog

The data converter is a dialog based application. When you launch RMADDataConv you are presented with this main dialog box:



There are two main groupings of fields, corresponding to the two main file conversion processes. These are for

- *CEL/CDF conversion*: For converting CDF and CEL files into CDFRME and RME format files.
- *PGF/CLF conversion*: Converting PGF/CLF files into CDFRME format files
- *Output directory*: The location to store RME files when processed. This field is mandatory for all work flows

For CEL/CDF file conversion there are four main fields into which the user can enter text or file paths. These include

- *CEL File Directory*: A directory containing CEL files to be converted into RME format files.
- *CDF File*: A CDF file to be converted into RME format
- *Restrict File*: A file containing probeset names, 1 per line
- *Force CDF Name*: A string specifying how the CDF information should be identified

For PGF/CLF file conversion there are four main fields into which the user can enter text or file paths. Currently two of these are disabled. These include

- *PGF File*: Location of PGF format file
- *CLF File*: Location of CLF format file
- *PS file*: Location of PS file
- *MPS file*: Location of MPS file

Note that typically only some of these fields need to be filled for a conversion job. There are also four main buttons

- *About*: Show version number of RMADataConv
- *Preferences*: Set preferences about buffering.
- *Convert*: Start the process of converting to RME format files
- *Quit*: For quitting RMADataConv

3.2 Converting a set of CEL or a CDF file to RME format

The simplest procedure for which you might use RMADataConv is to convert CEL and CDF files to RME format. Note that normally you would not want to do this, since RMAExpress can read CEL and CDF files directly. However, if it happens that you expect to have to re-read in your CDF and CEL files repeatedly into RMAExpress you may find that RME files can be read into RMAExpress faster than CEL and CDF files. The speed gains are most impressive when your CDF and CEL files are in text format.

To carry out the conversion you should specify a directory containing all your CEL files in the *CEL File Directory* field and the full path, including file name, to the CDF file in the *CDF File* field. The *Output directory* field should specify the location to store the processed RME files. The other two fields may be left blank. Clicking *Convert* will start the conversion process. When it finishes you should find a number of .RME files in the location specified.

RMADataConv can also convert CEL files or a CDF file alone to RME format. Just leave the other field blank.

3.3 Restricting the set of probesets used

Sometimes you may have a reason to remove some probesets from your dataset, since you do not wish to have them included when RMA does the quantile normalization and you do not require expression values for these probesets.

To do this you need a text file containing only the names of the probesets which should be kept in the dataset. There should be one name per line in this file. Specify the full path to this file in the *Restrict File* field. You should specify a directory containing all your CEL files in the *Cel File Directory* field and the full path, including file name, to the CDF file in the *CDF File* field. It is also recommended that you specify a new name for the RME format CDF file, so as to not confuse it with the original file. A recommended nomenclature would be *CDFNAMErestrict*. So suppose that you were dealing with HG_U133A chips then you would put the string *HGU_133Arestrict* in the *Force CDF Name* field. Finally, the *Output directory* field should specify the location to store the processed RME files. Clicking *Convert* will start the conversion process. When it finishes you should find the .RME files in the specified output location.

3.4 Turning PGF and CLF files into CDFRME files for Exon Array Analysis

Affymetrix provides unsupported CDF files for the Exon arrays (at the time at which this manual was written). While RMAExpress may work with these CDF files, they are also to be considered unsupported (ie use at your own risk). Instead, the PGF and CLF files should be used and converted into an appropriate CDFRME file. The RMADataConv application is designed to handle this process. In particular, you should specify the PGF and CLF files in the appropriate fields. The *Output directory* field should specify the location to store the output CDFRME file. Clicking *Convert* will start the conversion process. When it finishes you should find the CDFRME files in the location specified. A CDFRME file may be used interchangeably with a CDF file in the main RMAExpress application.

The naming convention for the CDFRME file that is produced is as follows:

```
lib_set_name.lib_set_version_pgfcdf.CDFRME
```

so for example

```
HuEx-1_0-st.r2.pgfcdf.CDFRME
```

3.4.1 Using PS files

In some cases rather than summarizing all exons, only RMA values for certain subsets of exons will be desired. Affymetrix classifies Exons into three sets: core, extended and full. Each set encompasses a greater number of exons. PS files may be found on the Affymetrix website. If this is specified in the *PS file* field along with appropriate CLF and PGF files, then the conversion process will produce a CDFRME file containing only the specified exons.

The naming convention for the CDFRME file that is produced is as follows:

```
lib_set_name.lib_set_version_ps_psfilename.CDFRME
```

so for example

```
HuEx-1_0-st.r2_ps_HuEx-1_0-st-v2.r2.dt1.hg18.full.CDFRME
```

3.4.2 Using MPS files

It is also possible to do gene-level analysis with Affymetrix Exon arrays. In particular, groups of related probes can be grouped together to get a gene-level expression summary. MPS files are provided for this purpose. If this is specified in the *MPS file* filed along with appropriate CLF and PGF files, then the conversion process will produce a CDFRME file that produces gene-level (rather than exon-level) expression summaries.

The naming convention for the CDFRME file that is produced is as follows:

```
lib_set_name.lib_set_version_mps_mpsfilename.CDFRME
```

so for example

```
HuEx-1_0-st.r2_mps_HuEx-1_0-st-v2.r2.dt1.hg18.extended.CDFRME
```

3.5 Merging MG_U74A and MG_U74Av2 datasets

Sometimes there are two very closely related versions of the same chip. In particular if the majority of the probesets are in common between the two chips, in terms of both location and sequence, then RMADataConv can be used to create hybrid CDF and CEL files in the RME format. At the time of writing the only two types of chips for which this is possible are MG_U74A and MG_U74Av2 or HG_U95A and HG_U95Av2. It is not recommended that you try to merge any other chip types. In these instructions we will assume that the user wants to merge together MG_U74A and MG_U74Av2 data.

The conversion happens in two steps. First all the version 1 chips are converted, then all the version 2 chips are converted.

To start, set the path to all your version 1 cel files in the *Cel File Directory* field (only MG_U74A CEL files should be in this directory) and the full path, including file name, to the MG_U74A CDF file in the *CDF File* field. In the *Restrict File* field you want a text file which contains the names of the probesets that are to be conserved. For MG_U74A/Av2 datasets you can get this from <http://bmbolstad.com/misc/mixtureCDF/MGU74Aoverlap.txt>. To be sure that the hybrid data is not confused with the original data you should put a string in the *Force CDF Name* field, in this case the recommended name would be MGU74Amix. Last, set the *Output directory* field to specify where the processed RME files will be stored. Click *Convert* to start the conversion process. When it finishes you should find the .RME files for your MG_U74A arrays in the specified output location.

Now you will need to repeat the process with the MG_U74Av2 CEL files. So now set the path to all your version 2 cel files in the *Cel File Directory* field and the full path, including file name, to the MG_U74Av2 CDF file in the *CDF File* field. The *Restrict File*, *Force CDF Name* and *Output directory* fields should be kept the same as before. Click *Convert* to start the conversion process. When it finishes you should now also find the .RME files for your MG_U74Av2 arrays in the specified output location. You should now be able to load your hybrid dataset into RMAExpress using the *Read Processed files* option in the *File* menu.

Chapter 4

RMAExpressConsole: the console application

RMAExpressConsole is a console (command-line) application. It has no GUI and is designed simply to process a specified set of CEL files and return RMA expression values. This makes it ideal for use in situations where RMA processing is done as batch jobs or perhaps to provide a web service which does RMA processing.

RMAExpressConsole expects two command line arguments. Each argument is the name of a file. The first file contains a list of files to process. The second file contains processing settings.

In particular, the first file contains the name and path of the CDF file on the first line followed by paths/filenames of each CEL file to be processed on subsequent lines.

The second file can be one of two different formats. The first line in this file should be the version number for the format of this file. Currently this version number can be 1, 2, 3 or 4.

For version 1: the second line should contain the name of the file to store the RMA expression values (including full path if not current directory). Subsequent lines could be one of: `no_background` or `no_normalization`, to turn off some of the pre-processing stages. However, it is not recommended you turn off these off.

For version 2: the second line should contain the name of the file to store the RMA expression values (including full path if not current directory). The third line should give a path location for storing temporary files, if needed. The fourth line states what sort of images should be produced. This can be any of *residuals*, *pos.resids*, *neg.resids*, *sign.resids*, *all.resids* and *none*. These images will be stored in the same directory as the RMA expression values. Subsequent lines could be one of: `no_background` or `no_normalization`, to turn off some of the pre-processing stages. However, it is not recommended you turn off these off.

For version 3: (introduced at 0.5 alpha 3) the second line should contain the name of the file to store the RMA expression values (including full path if not current directory). The third line should be one of *text* or *binary* which will control whether the outputted expression values are written as text or in the binary format. The fourth line should give a path location for storing temporary files, if needed. The fifth line states what sort of images should be produced. This can be any of *residuals*, *pos.resids*, *neg.resids*, *sign.resids*, *all.resids* and *none*. These images will be stored in the same directory as the RMA expression values. Subsequent lines could be one of: `no_background` or `no_normalization`, to turn off some of the pre-processing stages. However, it is not recommended you turn off these off. As of version 1.0 beta 1 you may also use the `plm.summarize` term here. This will cause the PLM

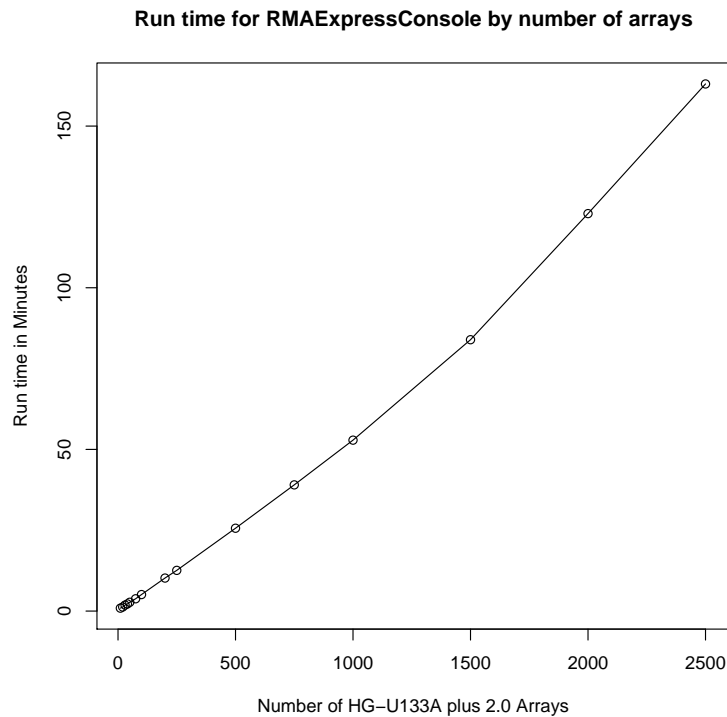


Figure 4.1: Running times using RMAExpressConsole 1.0 Release

summarization method to be used instead of the default median polish summarization. Additionally using this option will cause the console application to compute RLE and NUSE summary values and return these in separate text file outputs. Note that the `plm.summarize` option will be slower than the default median polish.

For [version 4](#): (introduced at 1.0 beta 7) the second line should contain the name of the file to store the RMA expression values (including full path if not current directory). The third line should be one of *text* or *binary* which will control whether the outputted expression values are written as text or in the binary format. The fourth line should give a path location for storing temporary files, if needed. The fifth line states what sort of images should be produced. This can be any of *residuals*, *pos.resids*, *neg.resids*, *sign.resids*, *all.resids* and *none*. These images will be stored in the same directory as the RMA expression values. The sixth line should be the number of rows (probes) to keep in the memory buffer and should be a positive integer value. The seventh line should be the number of columns (arrays) to keep in the column buffer and should be a positive integer value. Subsequent lines could be one of: `no.background` or `no.normalization`, to turn off some of the pre-processing stages. Another option is `plm.summarize` to use PLM summarization rather than median polish (the default).

4.1 How long will it take?

The amount of time it will take to process a set of CEL files and process them to RMA expression values depends on a number of factors including:

- Number of probes and probesets on the chip
- Number of CEL files
- Configuration of computer including processor speed, amount of RAM, operating system.
- Configuration of RMAExpress (or RMAExpressConsole)

A set of timing tests were run using RMAExpressConsole 0.4b1 on a machine with the following basic configuration:

- Version: RMAExpressConsole 0.4b1 with version 2 output file but no output of plots
- OS: Fedora Core 9 x86_64 Linux running kernel 2.6.25.6-55.fc9.x86_64
- Processor: AMD Athlon X2 5600+
- RAM: 8 GB DDR2 PC6400
- Chip type: HGU-133A Plus 2.0
- Number of CEL files: Varied from 10 to 2500
- Buffer size: 1 arrays, 25000 probes
- Timing: done using the shell command `time` and the output "Real".

with the results shown in Figure 4.1. It can be seen from this plot that, at least for this range of the number of CEL files, the processing time is roughly linear. The largest dataset run was 2500 CEL files and had a total running of approximately 163 minutes. This highlights the relative efficiency of RMAExpress and its ability to handle extremely large datasets.

4.2 Examples

For these examples the paths are given in Unix machine format, but if using on a Windows machine you would use different path names.

For the first file (call this `inputs.dat`):

```
/mnt/hd/GeneLogic/dilution.new/HG_U95Av2.CDF
/mnt/hd/GeneLogic/dilution.new/94394hgu95v2a11.cel
/mnt/hd/GeneLogic/dilution.new/94395hgu95v2a11.cel
```

For the second file (call this `outputs.dat`) a valid version 1 file would be:

```
1
/tmp/myRMAExpressValues.txt
```

For the second file (call this `outputs.dat`) a valid version 2 file would be:

```
2
/tmp/testRMAExpressValues.txt
/tmp
all.resids
```

For the second file (call this outputs.dat) a valid version 3 file would be:

```
3
/tmp/testRMAExpressValues.txt
binary
/tmp
all.resids
plm_summarize
```

For the second file (call this outputs.dat) a valid version 4 file would be:

```
4
/tmp/testRMAExpressValues.txt
text
/tmp
residuals
25000
1
```

Then the application would be executed like this:

```
RMAExpressConsole inputs.dat outputs.dat
```

Appendix A

Reference Material

The main references for the RMA algorithm are the following three manuscripts:

- Irizarry, RA, Bolstad, BM, Collin, F, Cope, LM, Hobbs, B and Speed, TP (2003), Summaries of Affymetrix GeneChip probe level data *Nucleic Acids Research* 31(4):e15
- Bolstad, BM, Irizarry RA, Astrand, M, and Speed, TP (2003), A Comparison of Normalization Methods for High Density Oligonucleotide Array Data Based on Bias and Variance. *Bioinformatics* 19(2):185-193
- Irizarry, RA, Hobbs, B, Collin, F, Beazer-Barclay, YD, Antonellis, KJ, Scherf, U, Speed, TP (2003) Exploration, Normalization, and Summaries of High Density Oligonucleotide Array Probe Level Data. *Biostatistics* .Vol. 4, Number 2: 249-264

The PLM methodology is described in:

- Bolstad, BM (2004) Low Level Analysis of High-density Oligonucleotide Array Data: Background, Normalization and Summarization. Dissertation. University of California, Berkeley. http://bmbolstad.com/Dissertation/Bolstad_2004_Dissertation.pdf

Quality assessment using the PLM methodology is described in:

- Bolstad BM, Collin F, Brettschneider J, Simpson K, Cope L, Irizarry RA, and Speed TP. (2005) Quality Assessment of Affymetrix GeneChip Data in *Bioinformatics and Computational Biology Solutions Using R and Bioconductor*. Gentleman R, Carey V, Huber W, Irizarry R, and Dudoit S. (Eds.), Springer, 2005.
- Brettschneider, J, Collin, F, Bolstad, BM, and Speed, TP (2008) Quality assessment for short oligonucleotide microarray data. To appear in *Technometrics*.

Appendix B

Building RMAExpress from source code

This section describes how to build RMAExpress from source code. Most users will not need to do this and instead should just use the pre-built binaries supplied on the website. **Warning:** do not try to build the application from source code unless you think you have a good reason to. Since RMAExpress uses wxWidgets (formerly known as wxWindows) you will need to install that before you can compile the source code. wxWidgets can be downloaded from <http://www.wxwidgets.org/>. Source code for RMAExpress can be downloaded from the website. Note that RMAExpress is licensed under the GPL version 2. Note that as of RMAExpress 1.0 Release it is built against wxWidgets 2.8.7.

B.1 Building native binaries for Linux

How you install wxWidgets depends very much on your distribution of Linux. For some distributions there is pre-built packages, while for others you will need to install it from source code. This user guide assumes that you have already have wxWidgets installed. To build native Linux binaries using the source code you should do the following

```
cp Makefile.unx Makefile
make
make console
```

B.2 Building native binaries for Windows

Beginning with 1.0beta6 the Windows builds have been made using Visual C++ Express Edition 2008 and wxWidgets 2.8.7. Visual C++ project and solution files are provided with the source distribution, but users who wish to build their own binaries via this methodology are unsupported.

B.3 How to install a cross-compiler on a Linux machine to produce RMAExpress Windows binaries.

These are the instructions for using a Linux machine to compile the source code and produce binaries that will be executable. This is the method that is to build the binaries that are currently supplied on the website. At the time of writing this manual, this workflow is know not to produce binaries that

function correctly on Windows Vista operating systems. This is the method that was used to build the windows binaries up until 1.0 beta 3.

1. The first step is to install a cross-compiler on your machine. To make it easy use the shell script provided on the website. You might have to change some of the version numbers for some of the files in the script.

Use `build-cross.sh` to download, configure, build and install the cross-compiler and requisite tools on your machine

Next you should download and put these scripts somewhere in your path (eg `/usr/local/bin` or `/bin`)

```
cross-configure.sh
cross-make.sh
```

2. The next step is to download and install wxWindows into the cross-compiled environment.

First download `wxAll-2.8.4.tar.gz`. Next extract and build the library.

```
tar xzvf wxAll-2.8.4.tar.gz
cd wxWidgets-2.8.4
cross-configure.sh --prefix=/usr/local/cross-tools --enable-unicode
cross-make.sh
cross-make.sh install
```

3. Finally you'll want to actually build the RMAExpress applications. So change to the location where you placed the source code for RMAExpress then

```
cp Makefile.dos Makefile
make all
make console
```

B.4 How to build RMAExpress on a Windows machine using MinGW

These are instructions for installing requisite software on to a Windows machine so that you have a sufficient environment within which to compile the source code for RMAExpress. At the time of writing this manual, this workflow is known not to produce binaries that function correctly on Windows Vista operating systems. Note this procedure has not been fully tested in several years and the make files may not be completely synchronized.

1. From <http://www.mingw.org/download.shtml> download and install the following

```
MinGW-3.1.0-1.exe
MSYS-1.0.10.exe
msysDTK-1.0.1.exe
```

2. Next we want to download the source code for wxWindows/wxWidgets and build it. In particular download

```
wxAll-2.6.2.tar.gz
```

3. Uncompress the source code and put it somewhere. Then launch MSYS, navigate to where you uncompressed the source code for wxWidgets (eg `cd /c/wxWidgets-2.6.2`)

```
configure
make
make install
```

4. Unpack the source code for RMAExpress and change (within MSYS) to the location where you stored the source code.

```
cp Makefile.msw Makefile
make
make console
```

5. Finally, you might want to test your compilation by running it. The first thing to do is put the `/usr/local/lib` directory in your path. Or copy the dlls to somewhere in your path.

```
export PATH=$PATH:/usr/local/lib
```

```
Finally lets test it.
RMAExpress
RMAExpressConsole
```

Appendix C

Brief changelog/history

Version	Date	Description
0.1 beta 1	Apr 25, 2003	First Public version
0.1 beta 2	Apr 30, 2003	Fixes/Optimizations to the CDF input routines
0.1 beta 3	May 20, 2003	A few warning messages added. A small memory leak eliminated
0.1 beta 4	Jun 04, 2003	A check that memory was properly allocated in normalization routine
0.1 Release	Jun 11, 2003	No changes from 0.1 beta 4, only a bump in version number
0.2 alpha 1	Jul 22, 2003	A processed data format is introduced. This will speed up reloading data sets.
0.2 alpha 2	Aug 14, 2003	You can add additional CEL files after you have already loaded some in
0.2 alpha 3	Sep 12, 2003	A batch file convertor
0.2 alpha 4	Sep 18, 2003	Fixes some problems with cdf filepaths (in convertor) on Windows
0.2 alpha 5	Oct 9, 2003	Faster CEL file parser
0.2 alpha 6	Oct 19, 2003	Preliminary support for the new binary cel file format.
0.2 beta 1	Oct 31, 2003	Show menu. Low memory Overhead normalization step.
0.2 beta 2	Nov 16, 2003	Critical fix for binary cel file support (previous versions will give incorrect results)
0.2 Release	Jan 11, 2004	No changes from 0.2 beta 2. Only bump in version number
0.3 alpha 1	Jan 27, 2004	It is now possible to store and visualize RMA residuals.
0.3 alpha 2	Feb 29, 2004	The RMA residual images may now be saved.
0.3 alpha 3	Jun 27, 2004	RMAExpressConsole application introduced.
0.3 alpha 4	Jul 7, 2004	Support for chips with PM only probesets
0.3 alpha 5	Oct 13, 2004	Minor bug fixes, deals better with sense transcript arrays, output in either log2 scale (traditional) or natural scale
0.3 alpha 6	Oct 19, 2004	Minor bug fixes
0.3 beta 1	Nov 9, 2004	Fixes to deal with soybean chips
0.3 Release	Dec 14, 2004	No changes from 0.3 beta 1, Only a bump in version number
0.4 alpha 1	Feb 19, 2005	Preliminary support for binary (xda) format cdf files.
0.4 alpha 2	Mar 25, 2005	Fix a minor bug in background correction routine that on rare occasions causes slight difference in expression measures than those computed using R/BioConductor (usually difference is in 3rd decimal place). Some changes/additional progress bars.
0.4 alpha 3	Apr 1, 2005	Experimental support for dealing with extremely large datasets (200 or more arrays)

Version	Date	Description
0.4 alpha 4	Jun 5, 2005	Initial User Guide, max arrays in buffer now 150
0.4 alpha 5	Jul 11, 2005	Added "signs" image option. Corrected assignment of Red and Blue colors in residuals images (which were the reverse of what they should be). Code now built against wxWidgets 2.6.x.
0.4 alpha 6	Aug 23, 2005	Fix console application so that filename for output is fully pathable. Bug fix for "Write process files" with PM only chips.
0.4 alpha 7	Aug 30, 2005	Fixes for console application.
0.4 beta 1	Oct 29, 2005	Add basic residual images ability to console application. sign of residuals images now set unused regions to white
0.4 Release	Nov 10, 2005	Preserve some user controllable options when application quits.
0.4.1 Release	Jan 30, 2006	Fixes for residual images dialog box with large chips.
0.5 alpha 1	Mar 30, 2006	Preliminary experimental support for exon arrays
0.5 alpha 2	Apr 4, 2006	Improved support for exon arrays. An export function.
0.5 alpha 3	May 1, 2006	Option for binary output from console app.
0.5 alpha 4	May 5, 2006	Bug fixes for console application.
0.5 alpha 5	Aug 3, 2006	Fixes problem when large number of binary files on Windows platforms.
0.5 alpha 6	Aug 31, 2006	Fix lock up situation when CDF filenames don't match.
0.5 alpha 7	Sep 17, 2006	Fix source code so it compiles successfully on Unicode builds of wxWidgets. Rebuild windows binary.
0.5 Release	Feb 26, 2007	Fix output in console application.
1.0 beta 1	Mar 24, 2007	First release incorporating PLM, NUSE, RLE
1.0 beta 2	Jun 17, 2007	Fix plot placement when printed to high resolution output device
1.0 beta 3	Aug 23, 2007	Additional QC assessment plots, Code now built against wxWidgets 2.8.x
1.0 beta 4	Oct 28, 2007	Add support for reading AGCC format CEL files. Significant restructuring of CEL file parsing code. Significant changes to source code to improve portability.
1.0 beta 5	Jan 20, 2008	PGF/CLF parsing in RMADDataConv RME and CDFRME files may be read and combined with regular CEL/CDF files
1.0 beta 6	Feb 2, 2008	Fix crash on reading non RME format cel files affecting XP, Windows 2000
1.0 beta 7	Feb 16, 2008	Allow minimum of 1 array in buffer (previous value was 5) Version 4 of outputsettings for console application Console Application prints out more details Add PS file support to RMADDataConv
1.0 beta 8	Feb 29, 2008	Fix indexing crash in extremely large datasets
1.0 beta 9	Mar 10, 2008	Improved CEL file corruption checking
1.0 beta 10	Mar 20, 2008	MPS file support added to RMADDataConv
1.0 Release	Jun 29, 2008	Small fix for bg correction crash
1.0.1 Release	May 16, 2009	Small fix for parsing binary format CDF files
1.0.2 Release	May 19, 2009	Fix in PGF/CLF to CDFRME conversion
1.0.3 Release	May 21, 2009	Fix in PGF/CLF with MPS to CDFRME conversion
1.0.4 Release	Jul 20, 2009	Fix in PGF/CLF with MPS to CDFRME conversion for Windows build
1.0.5 Release	May 22, 2010	Small fixes for MoGene 1.1 and other non rectangular WT gene arrays

Appendix D

File format information

D.1 Binary format output file

This file is in little-endian format.

Field	Details	Number
File Descriptor	int - length of string char* - string which should be "RMAExpressionValues"	1
File Format Version Number	int - Currently this is 1	1
RMAExpress Version Number	int - length of string char* - string giving version of RMAExpress used.	
CDF NAME	int - length of string char* - string giving CDF file used to process data	1
Number of arrays	int - number of arrays	1
Number of probesets	int - number of probsets	1
Array names	int - length of string char* - string giving cel file names	Number of arrays
Probeset names	int - length of string char* - string giving probeset names	Number of probesets
Expression values	double - stored in column order	Number of arrays by number of probesets