

FAQ on Exon Array Related Changes

Version 1: October 11th, 2006

The changes described here are specific to the Human, Mouse, and Rat Exon 1.0 ST arrays but relate to the release of the Expression Console™ Software 1.0 (EC) [1] and the Affymetrix Power Tools (APT) 1.6.0 [2].

Q. What is the relationship between ExACT, APT, and EC?

A. ExACT is the original software released to support exon array analysis. ExACT had both a graphical user interface for Windows and a command line interface. The ExACT source code has evolved into APT which provides an improved set of command line programs and the Expression Console software (EC) which provides an improved GUI for both Exon and 3' Expression Array analysis. The APT command line programs no longer only work with exon arrays, but now will analyze regular 3' Expression Arrays and the WGS based mapping arrays. APT is a DevNet tool which means there is no official support for the package. (See the Affymetrix Software Support Policy[3] for more information.). EC integrates the analysis of 3' Expression Arrays and Exon Arrays under one application. EC is a supported Affymetrix Tool application. (See the Affymetrix Software Support Policy[3] for more information about limits on support for Affymetrix Tools.)

Q. What has happened to the exon array library files.

A. A number of improvements were implemented for the ExACT library files to support EC and to provide extra functionality with regard to QC reports. For starters, individual files are now available via the NetAffx SDK. EC uses this to enable users to download all the files needed to analyze exon arrays at one time. Other changes include:

- The file names now include the chip type (ie HuEx-1_0-st-v2) along with a core library file version (ie r2).
- Probeset lists now have a “.ps” extension.
- Meta probeset lists now have a “.mps” extension.
- The PolyA and Bacterial spike probesets available for the Human, Mouse, and Rat Exon Arrays were synchronized. This involved adding the Affx-Bs-trp_x_st probeset to the Human Exon Array library files and removing extra and redundant probesets from the rat and mouse exon array library files.
- Removed various unused internal control features from the .pgf file (*e.g.*, the probeset containing the features used for the chip name text).
- Added a QC Control file(extension “.qcc”) to identify the probesets on the array that are controls, their name, what category to which they belong, and whether or not they should be reported in the CHP header by EC 1.0.. APT uses this file to create the report.txt file via the `-qc-probesets` option.
- Added a probeset list for all the expression style probesets on the array (for example this probeset list excludes the background control probesets listed in the .bgp files)
- Added Control probesets to the meta probeset files to ensure that QC report metrics were generated for both exon and gene level analysis in EC and APT.
- Added an annotation version (ie “dt1”) and the genome version (ie “hg16”) to the file names for files that are genome version dependent.
- Added probeset lists for core, extended, and full to go with the existing meta probeset lists for these levels of annotation. These lists are inclusive: extended includes core and full includes core and extended. Controls are also included in these probeset lists to ensure that full QC reports are generated when using these files.

- Updated the headers for all the files, including new GUIDs (global unique IDs) and updated create_date fields.
- Added the EC XML configuration file to the core library zip file. This was previously referred to as the ExACT library files and are now called Analysis library files.

Q. In the older files I have a file named X. What is the file name in these new files?

A. Below is an example table for the Human Exon 1.0 ST array.

| Old Zip Package | Old Files | New Zip Package | New Files |
|---|---|---------------------------------------|---|
| HuEx-1_0-st-v2.zip | HuEx-1_0-st-v2/HuEx-1_0-st-v2.clf HuEx-1_0-st-v2/HuEx-1_0-st-v2.pgf HuEx-1_0-st-v2/antigenomic.bgp HuEx-1_0-st-v2/genomic.bgp HuEx-1_0-st-v2/probeset-list.control.affx.txt HuEx-1_0-st-v2/probeset-list.main.txt HuEx-1_0-st-v2/probeset-list.normgene.exon.txt HuEx-1_0-st-v2/probeset-list.normgene.intron.txt HuEx-1_0-st-v2/probeset-list.rescue.FLmRNA.unmapped.txt | HuEx-1_0-st-v2.r2.zip | HuEx-1_0-st-v2.r2.clf HuEx-1_0-st-v2.r2.pgf HuEx-1_0-st-v2.r2.antigenomic.bgp HuEx-1_0-st-v2.r2.genomic.bgp HuEx-1_0-st-v2.r2.controls.ps HuEx-1_0-st-v2.r2.unmapped-transcripts.ps HuEx-1_0-st-v2.r2.qcc HuEx-1_0-st-v2.r2.all.ps HuEx-1_0-st-v2.exon_analysis_configuration |
| HuEx-1_0-st-v2.design-annot-hg16.zip | HuEx-1_0-st-v2/design-annot-hg16/HuEx-1_0-st-v2.annot.hg16.csv HuEx-1_0-st-v2/design-annot-hg16/meta-probeset.core.txt HuEx-1_0-st-v2/design-annot-hg16/meta-probeset.extended.txt HuEx-1_0-st-v2/design-annot-hg16/meta-probeset.full.txt | HuEx-1_0-st-v2.r2.dt1.hg16.zip | HuEx-1_0-st-v2.r2.dt1.hg16.csv HuEx-1_0-st-v2.r2.dt1.hg16.core.mps HuEx-1_0-st-v2.r2.dt1.hg16.extended.mps HuEx-1_0-st-v2.r2.dt1.hg16.full.mps HuEx-1_0-st-v2.r2.dt1.hg16.core.ps HuEx-1_0-st-v2.r2.dt1.hg16.extended.ps HuEx-1_0-st-v2.r2.dt1.hg16.full.ps |
| HuEx-1_0-st-v2.design-annot-hg17.zip | HuEx-1_0-st-v2/design-annot-hg17/HuEx-1_0-st-v2.annot.hg17.csv HuEx-1_0-st-v2/design-annot-hg17/meta-probeset.core.txt HuEx-1_0-st-v2/design-annot-hg17/meta-probeset.extended.txt HuEx-1_0-st-v2/design-annot-hg17/meta-probeset.full.txt | HuEx-1_0-st-v2.r2.dt1.hg17.zip | HuEx-1_0-st-v2.r2.dt1.hg17.csv HuEx-1_0-st-v2.r2.dt1.hg17.core.mps HuEx-1_0-st-v2.r2.dt1.hg17.extended.mps HuEx-1_0-st-v2.r2.dt1.hg17.full.mps HuEx-1_0-st-v2.r2.dt1.hg17.core.ps HuEx-1_0-st-v2.r2.dt1.hg17.extended.ps HuEx-1_0-st-v2.r2.dt1.hg17.full.ps |
| HuEx-1_0-st-v2.design-annot-hg18.zip | HuEx-1_0-st-v2/design-annot-hg18/HuEx-1_0-st-v2.annot.hg18.csv HuEx-1_0-st-v2/design-annot-hg18/meta-probeset.core.txt HuEx-1_0-st-v2/design-annot-hg18/meta-probeset.extended.txt HuEx-1_0-st-v2/design-annot-hg18/meta-probeset.full.txt | HuEx-1_0-st-v2.r2.dt1.hg18.zip | HuEx-1_0-st-v2.r2.dt1.hg18.csv HuEx-1_0-st-v2.r2.dt1.hg18.core.mps HuEx-1_0-st-v2.r2.dt1.hg18.extended.mps HuEx-1_0-st-v2.r2.dt1.hg18.full.mps HuEx-1_0-st-v2.r2.dt1.hg18.core.ps HuEx-1_0-st-v2.r2.dt1.hg18.extended.ps HuEx-1_0-st-v2.r2.dt1.hg18.full.ps |

Q. What zip files do I need to download to get started?

A. If your using Expression Console™ and are connected the internet, then you should use the library file download feature which is a part of EC. If your not connected to the internet or are using APT, then at a minimum you will need to download the core Analysis zip file (ie HuEx-1_0-st-v2.r2.zip) and the most recent genome version release (ie HuEx-1_0-st-v2.r2.dt1.hg18.zip). These files can be unzipped and placed in the Expression Console Library File Folder or in a convenient location for use with APT. Note that the EC configuration file provided in the core Analysis zip file (ie HuEx-1_0-st-

v2.exon_analysis_configuration) also references files in the most recent genome version release. You will need to create a custom configuration file if you want to use older genome version releases in EC.

Q. Can I use these new library files with ExACT and APT?

A. Yes, the new library files will work with ExACT and APT. New refers to the new PGF/CLF and related files provided for all the exon arrays on Oct 9th, 2006 as part of the EC launch.

Q. Can I use the old library files with ExACT and APT?

A. Yes, you can continue to use the old library files with ExACT and APT. Old refers to the original PGF/CLF and related files provided when the exon arrays were first launched.

Q. Can I use the old library files with EC?

A. Yes, but an advanced exon configuration will need to be created to use them. From the menu bar select **Analysis-Advanced Exon Configurations-new**. Use the advanced configuration dialog and select old pgf, clf, and bgp files. To select the probelist and meta-probe list files you will need to change the file extension to “ps” and “mps” respectively. The application needs a qcc file to execute the analysis. The qcc file provided by with the current release of the library files will enable the analysis to run, but information on the controls will only be available if the controls listed in the qcc files are in the selected meta-probeset and probeset lists (*e.g.*, the original meta-probeset lists did not contain the controls, so gene level analyses using these files with the new qcc file will not have any information on the control probes in the reports).

Q. I’ve seen talks from Affymetrix and/or the gene summarization whitepaper which recommend use of IterPLIER for gene level estimates. EC defaults to RMA. Which should I use.

A. IterPLIER provides more robust results when including speculative content (*i.e.*, the full level meta probeset list) for the gene level estimate. It does this by aggressively removing outliers and basing the final signal estimate on only 11 probes.. When using more conservative content (*i.e.*, core or extended) this benefit is less pronounced with the downside that the final estimate may be based on a suboptimal set of 11 probes. RMA has a couple benefits. First, it is faster than PLIER because in comparison it is a computationally simpler method. The second benefit is that RMA is already recognized by the broader expression analysis community as the “Best Practice” approach to analyzing 3’ Expression Arrays. So based on this RMA was chosen as the default for gene level analysis within EC. We anticipate that gene level signal estimation methods will continue to improve for Exon Arrays, but for now RMA made for a reasonable and conservative choice. If you’ve been using IterPLIER or PLIER and are satisfied with the performance or if you want to try these methods out, you can use them within EC by defining an advanced configuration.

Q. How do can I run iterPLIER for my exon arrays?

A, From the menu bar select **Analysis-Advanced Exon Configurations-new**. In the **Analysis Type section** select the desired annotation level and confidence. Select **iterPLIER** as the Summarization Method, **PM-GCBG** for the Background Correction, and **Sketch-Quantile** as the Normalization Method. Use the default library files for the analysis.

Q. Will Affymetrix continue to create CDF files moving forward or is the shift to PGF files here to stay?

A. For 3’ Expression Arrays and genotyping arrays there are no short term plans to move away from CDF files. For Exon Arrays, there is likewise no short term plans to move away from PGF (and CLF) files. There are some initial efforts to unify the various content related files (ie CDF, PGF/CLF, bmap)

however this effort is still in a prototype/research phase. We will provide more information about library file changes as those plans mature.

Q. What is the current format of the PGF/CLF files and are there any plans on changing this format in the future?

A. The APT 1.6.0 source code release contains draft document specifications for both of these files. There are some discussions (but no plans yet) on extending the current file specification. Some specific items under consideration are:

- Include additional headers:
 - rows and cols in cel file
 - # probesets in the PGF file
 - add an MD5 sum of the non-header content of the file
- Add the atom count to the probeset entries
- Add the probe count to the atom entries
- Drop the CLF file (assume sequential=1 and order=col_major for all arrays; see the draft CLF spec for more information about sequential and order)

Q. Is there a PGF and CLF parser?

A. In the APT 1.6.0 source code under sdk/file/TsvFile is a low level parser for both these files as well as the TSV files which both are based on. See sdk/chipstream/ChipLayout.cpp for an example of how these parsers are used.

Q. Is there a CDF to PGF/CLF parser? And the reverse?

A. No. The information allowed in these two file formats does not allow for lossless conversion between the two. There are no plans to provide such a converter.

Q. Are PGF files specific to the new Command Console world, or are there GCOS and command console versions of the CDF files as well?

A. PGF and CDF files are not specific to GCOS or Command Console. Rather they are array specific and related to the analysis applications down stream of GCOS and Command Console. Currently CDF files can exist in either a text or binary (XDA) format. PGF files only exist in a text format. There are no Command Console Binary Format CDF files. In contrast, there are Command Console Binary Format CHP files.

Q. Will PGF files work with GCOS?

A. Essentially no. PGF/CLF files are not provided in the GCOS library file installer and the GCOS database knows nothing about these files. PGF/CLF files are used by Expression Console, APT, and some 3rd party software. For arrays (*i.e.*, exon arrays) that have a PGF file, you still use GCOS to scan the chips and create a CEL file. Those CEL files are then exported from GCOS and used by EC, APT, or another 3rd party application to generate signal estimates and other probeset summaries.

Q. How can I run MiDAS on EC output?

A. You will need to install EC as well as MiDAS. MiDAS is available as a command line application in the APT package (apt-midas). The steps are as follows:

- Part I: Generate Gene and Exon Summaries:
 - Start EC
 - Load CEL files into a new study

- Run a gene level analysis on the data set using EC by clicking on “**Run Analysis**” and selecting a gene level analysis option
- Click on “**Check Group**” and select the gene level CHP results
- From the main menu select **Export->Export Probeset Results (pivot table) to txt...** and provide a file name for the gene level summaries
- Click on “**Check Group**” and select the Probe Cell Intensity Data
- Run an exon level analysis on the data set using EC by clicking on “**Run Analysis**” and selecting an exon level analysis option
- Click on “**Check Group**” and select the exon level CHP results
- From the main menu select **Export->Export Probeset Results (pivot table) to txt...** and provide a file name for the exon level summaries
- Note, if you attempt to export the Gene and Exon level data at the same time one file is created with the two groups “stacked” on top of each other. This will not work with the current MiDAS application.
- Part II: Prepare Files for Use by MiDAS
 - Gene Summary Export File
 - add “probeset_id” header to first column
 - for each of the other column headers, remove the extension added to the original CEL file name (*e.g.*, “huex_wta_breast_A.rma-gene-core.chp” becomes “huex_wta_breast_A”)
 - Exon Summary Export File
 - add “probeset_id” header to first column
 - for each of the other column headers, remove the extension added to the original CEL file name (*e.g.*, “huex_wta_breast_A.plier-exon-all-dabg.chp” becomes “huex_wta_breast_A”)
 - These names should match (be identical) to the column headers in the gene summary file
 - Cel Group File: This file defines how to group the CEL files together for the MiDAS test. The format of the file is a tab separated text file with a column with the CEL file names (the values here need to match the column headers in the summary files) and a column with the group name. A mock example:


```
cel_files [tab] group_id
cel1 [tab] normal
cel2 [tab] normal
cel3 [tab] normal
cel4 [tab] treated
cel5 [tab] treated
cel6 [tab] treated
```

 - The file must have a header line at the top indicating which column has the cel file names (“cel_files”) and which defines the groups (“group_id”).
 - Note that if your using an older version of MiDAS (ie in APT 1.4.0 or ExACT 1.2.1) then the column label is “cel_file” rather than “cel_files”. The label was changed in newer versions of MiDAS to improve consistency with other applications that use files listing CEL files.
 - Extra columns of information are ignored.
 - There must be a row for each sample to be included in the analysis.

- Any sample with the same value in the group_id column will be grouped together for the MiDAS test.
 - One way to generate the cel group file is within EC to export the study as text.
 - Select Export->Export Study to txt ... and provide a file name
 - Open the new file in a text editor or Excel
 - Remove all but the header and CEL file rows/lines
 - Change the header for the “File” column to “cel_files”
 - Remove the “.CEL” extension for the entries in the “File” column
 - the resulting values should match the headers in the gene and exon summary files created above.
 - In this example above, you could also create a new column, “cel_files” rather than changing the “File” column.
- Part III: Run MiDAS
 - From the Windows Start menu select Start->Affymetrix Power Tools->apt-x.x.x->APT Command Prompt
 - Change the directory (using “cd” command) to the location of the files created above or copy those files to the default location of the command prompt (typically the desktop)
 - Run MiDAS. An example command line (you will need to enter on one line which may wrap):


```
C:\Documents and Settings\AWilli\Desktop>apt-midas
-g gene.txt -e exon.txt -o midas
-m c:\ec-library-files\HuEx-1_0-st-v2.r2.dt1.hg18.core.mps
--cel-files groups.txt
```

 - use “-g” to specify the gene level summary file
 - use “-e” to specify the exon level summary file
 - use “-o” to specify the output folder
 - use “-m” to specify the meta probeset file (which tells the software which exons go with which genes). You can find Affymetrix provided meta probeset files (have .mps extension) in the Expression Console library file folder.
 - use “--cel-files” to specify the cel grouping file created above
 - filenames and paths which have spaces will need to be surrounded by double quotes
 - See the apt-midas manual provided with APT for more options
 - In the MiDAS output, probeset_list_id refers to an exon level probeset ID and the probeset_id column refers to the gene level transcript cluster ID

Q. Running MiDAS on EC output seems awfully complicated. Are there plans to simplify this?

A. As best practices emerge from both within Affymetrix and from the broader community of Affymetrix users, we will consider rolling those into a more turn-key application for altsplise analysis. Also note that some of the 3rd party analysis software offerings provide such a turn-key solution.

References:

[1] http://www.affymetrix.com/products/software/specific/expression_console_software.affx: Expression Console Product Page

[2] <http://www.affymetrix.com/support/developer/powertools/index.affx>: APT DevNet Page

[3] http://www.affymetrix.com/support/technical/software_support_policy.affx: Affymetrix Software Support Policy

ERROR: undefined
OFFENDING COMMAND:

STACK: