AltAnalyze Information and Instructions

Version 1.15

Table of Contents

Section 1 - Introduction

1.1 Program Description

AltAnalyze (http://altanalyze.org) is a freely available, cross-platform application that allows you to take raw (CEL file) or processed microarray data and assess alternative splicing or alternative promoter usage and then view how these changes may affect protein sequence, domain composition, and microRNA targeting. This software requires no advanced knowledge of bioinformatics programs or scripting. All you need are your microarray files or a list of regulated probe sets along with some simple descriptions of the conditions that you're analyzing. In addition to splicing sensitive microarrays (Affymetrix Exon 1.0, Gene 1.0 and AltMouse), AltAnalyze can analyze conventional gene expression arrays from Affymetrix (3' arrays), Agilent, Illumina, Codelink and others. Step-by-step tutorials are available from our website.

AltAnalyze is composed of a set of modules designed to (A) summarize, organize and filter transcript tiling data; (B) calculate scores for alternative splicing (AS), alternative promoter selection (APS) or alternative 3' end-processing; (C) annotate regulated alternative exon events; and (D) assess downstream predicted functional consequences at the level of protein domains and microRNA binding sites (miR-BS) and biological pathways. The resulting data will be a series of text files (results and over-representation analyses) that you can directly open in a computer spreadsheet program for analysis and filtering (Figure 1.1). In addition, export files are created for the Cytoscape (1) program DomainGraph, to graphically view domain and miR-BS probe set-exon alignments and AltAnalyze statistics.

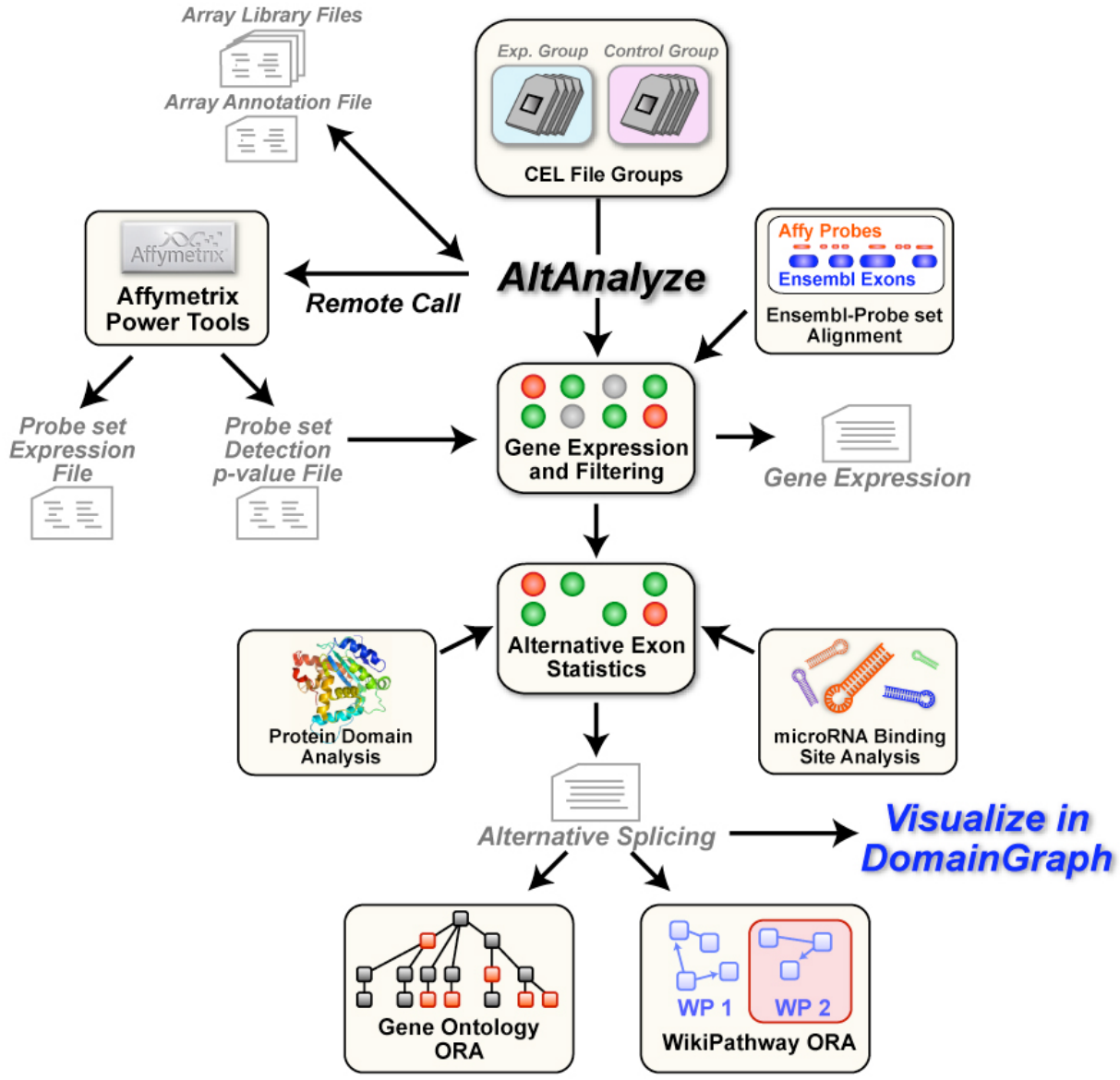


Figure 1.1. AltAnalyze Overview. Simplified graphic illustration of the analysis steps and output files produced by AltAnalyze. ORA, over-representation analysis.

Alternative exon analysis is currently compatible with the Affymetrix Exon 1.0 ST, Affymetrix Gene 1.0 ST array and the custom exon-junction Affymetrix AltMouse A array, however, it may be adapted to support other platforms on a per example basis

(contact author) or by other developers. Analysis of conventional Affymetrix microarrays is supported for array normalization, calculation of array group statistics, dataset annotation and pathway over-representation analysis. For non-Affymetrix arrays, all these steps are supported with exception to array normalization.

1.2 Updates

Version 1.15 Updates

New to version 1.15 AltAnalyze are new splicing algorithm options, new array support and improved DomainGraph connectivity. The main new features are:

1) Addition of **FIRMA** alternative exon analysis (Section 3.2)
2) Alternative exon analysis support or the Affymetrix **Gene 1.0** array (Section 2.3)
3) Universal import of **3rd party alternative exon results** for analysis (Section 3.2)
4) **Restricted analysis** of a pre-determined set of probe sets (Section 3.2)
5) Bundled integration of Cytoscape and **DomainGraph** (Sections 2.2 and 8)
6) Improved command-line options (Section 2.3)

Version 1.14 Updates

Several features have been added to AltAnalyze version 1.14. Please review these if you have used previous versions of AltAnalyze, since the method in which gene expression is evaluated for both gene expression and alternative exon analyses has been updated and will affect your results. The main new features are:

7) Updated Ensembl version 54 exon array annotations for constitutive gene probe sets (Section 3.4 and Section 6.2).
8) Support for core probe set only gene expression calculation (Section 3.4)
9) Improved filtering methods for gene expression data (Section 3.4).
10) Addition of advanced gene expression analysis statistics (Section 3.2).
11) Support for multi-group (>2) alternative exon analyses (Section 2.5).

12) GO-Elite support for adjusted p-value filtering optional settings (Section 3.2).

13) Improved automatic array type identification.

14) More streamlined graphical user interface options.

If you would like to compare results from version 1.14 to 1.13, please see the link on the AltAnalyze website to download and install the previous version (http://www.altanalyze.org/news_updates.htm). Please feel free to provide feedback to us at: genmapp@gladstone.ucsf.edu.

1.3 Implementation

AltAnalyze is provided as a stand-alone application that can be run on Windows or Mac OS X operating systems, without installation of any additional software. This software is composed of a set of distinct modules written in the programming language Python and distributed as stand-alone programs and source-code. Python is a cross-platform compatible language; therefore, AltAnalyze can be run on any operating system that has Python and Tkinter for Python installed. On many operating systems, including Linux and any Mac OS X operating systems the necessary python components are included by default, however, on some operating systems, such as Ubuntu, Tkinter may need to be installed when using the graphical user interface. AltAnalyze can be run from either an intuitive graphic user interface or from the command-line. To run AltAnalyze from source-code, rather than through the compiled executables, see section 2.2) for more information.

1.4 Requirements

The basic installation of AltAnalyze requires a minimum of 500MB of hard-drive space for all required databases and components. Species databases are downloaded

separately by the user from within the program, for various database versions. A minimum of 1GB of RAM and Intel Pentium III processor speed are further required. At least an additional 1GB of free hard-drive space is recommended for building the required output files. Additional RAM and hard-drive space is recommended for large exon array studies.

1.5 Pre-Processing, External Files and Applications

AltAnalyze can process raw Affymetrix image files (CEL files) using the RMA algorithm. This algorithm is provided through Affymetrix Power Tools (APT) binaries that are distributed with AltAnalyze in agreement with the GNU distribution license (see agreement in the AltDatabase/afffymetix/APT directory). Alternatively, users can pre-process their data outside of AltAnalyze to obtain expression values using any desired method. Example methods for obtaining such data include ExpressionConsole (Affymetrix) and R (Bioconductor), either of which can be used if the user desires another normalization algorithm rather than RMA (e.g., GC-RMA, PLIER). Likewise, users with non-Affymetrix data can use an appropriate normalization method. FIRMA alternative exon analysis is only supported when users are or have previously analyzed CEL files for the dataset of interest, since FIRMA scores are calculated from RMA probe-level residuals, rather than probe set expression values.

If Affymetrix CEL files are processed directly by AltAnalyze (using APT and RMA), two files will be produced; an expression file (containing probe set and expression values for each array in your study) and a detection above background (DABG) p-value file (containing corresponding DABG p-values for each probe set). As mentioned, if FIRMA is selected as the alternative exon analysis algorithm, APT will first perform a separate RMA run to produce probe residuals for gene-level metaprobesets (Ensembl associated AltAnalyze core, extended or full probe sets). The results produced by AltAnalyze will be identical to those produced by APT or

ExpressionConsole(http://www.affymetrix.com/products/software/specific/expression_console_software.affx). For some exon arrays, users can choose to exclude certain array probes based on genomic cross-hybridization (section 2.3).

For array summarization, all required components are either pre-installed or can be downloaded by AltAnalyze automatically (Affymetrix library and annotation files) for most array types. If the user is prompted for a species library file that cannot be downloaded, the user will be asked to download the appropriate file from the Affymetrix website. Offline analyses require the user to follow the instructions in Section 2.3.

1.6 Help with AltAnalyze

Additional documentation, tutorials, help, and user discussions are available at the AltAnalyze website http://www.altanalyze.org or at our google groups http://groups.google.com/group/alt_predictions. Tutorial 1 applies to conventional microarray analysis and Tutorial 2 applies to exon array analysis. Downloads, tutorials and help for DomainGraph can be found at http://domaingraph.bioinf.mpi-inf.mpg.de.

Section 2 – Running AltAnalyze for Splicing Arrays

2.1 Where to Save Input Expression Files?

When performing analyses in AltAnalyze, the user needs to store all of their Affymetrix CEL files in one directory. This directory can be placed anywhere on your computer and will be later selected in AltAnalyze. Example files can be downloaded from ftp://ftp.ncbi.nih.gov/pub/geo/DATA/supplementary/series/GSE13297/GSE13297_RAW.tar. Extract any downloaded TAR and/or GZIP compressed files prior to analysis.

If the user has already run normalization on their CEL files outside of AltAnalyze or have downloaded already analyzed expression data from another source, you can save the expression and DABG p-value file (optional) anywhere on your computer. These files should be tab delimited text files that only consist of probe sets, expression values and headers for each column. Example files can be downloaded http://AltAnalyze.org/normalized_hESC_differentiation.zip.

2.2 Running AltAnalyze from the Graphic User Interface

Windows and Mac Directions:

Once you have saved your CEL files or normalized expression values to your computer, open the AltAnalyze application folder and double-click on the executable file named “AltAnalyze.exe” (Windows) or “AltAnalyze” (Mac). This will open a set of user interface windows where you will be presented with a series of program options (see following sections).

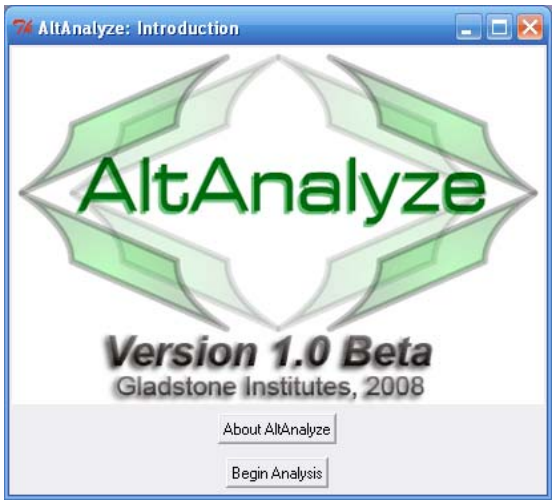
Unix/Linux and Source Code Directions:

On Linux, download the Linux executable and python source code archive version of AltAnalyze. To run the compiled version, double-click the executable file

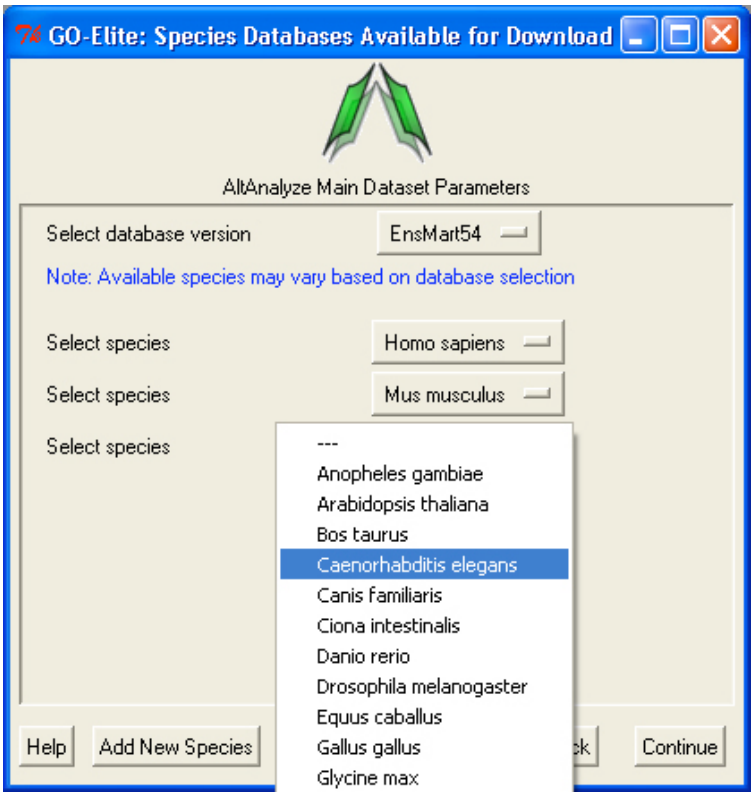
named “AltAnalyze” or open this file from command-line (./AltAnalyze). If this file is not compatible with your configuration, you can start AltAnalyze by opening a terminal window and going to the AltAnalyze main folder (e.g. “cd AltAnalyze_v1release” from the program parent directory). Once in this directory, typing “python AltAnalyze.py” in the terminal window will begin to run AltAnalyze (you should see the AltAnalyze main menu within a matter of seconds). Note: If there is a problem running the graphical user interface version of AltAnalyze, you can report this to our help desk or use the command-line options (Section 2.3), which do not require TK.

AltAnalyze Graphical Interface Options:

There are many options in AltAnalyze, which allow the user to customize their output, the types of analyses they run and the stringency of those analyses. The following sections show the sequential steps involved in running and navigating AltAnalyze. Interactive tutorials for different analyses are provided from the AltAnalyze website. ***Please note: if you will be using AltAnalyze on a machine that does NOT have internet access, follow instructions 1-5 below on an online machine and then copy the AltAnalyze program directory to an offline machine.***



1) Introduction Window – Upon opening AltAnalyze, the user is presented with the AltAnalyze splash screen and additional information. To directly open the AltAnalyze download page, follow the hyperlink under “About AltAnalyze”, otherwise select “Begin Analysis”.



2) Species Database Installation – The first time AltAnalyze is used, the user will be prompted to download one or more species database (requires internet connection). Independent of the array type you are analyzing, select a species and continue. The user can select from different versions of Ensembl. If your species is not present, select the button “Add New Species”.

A **B**

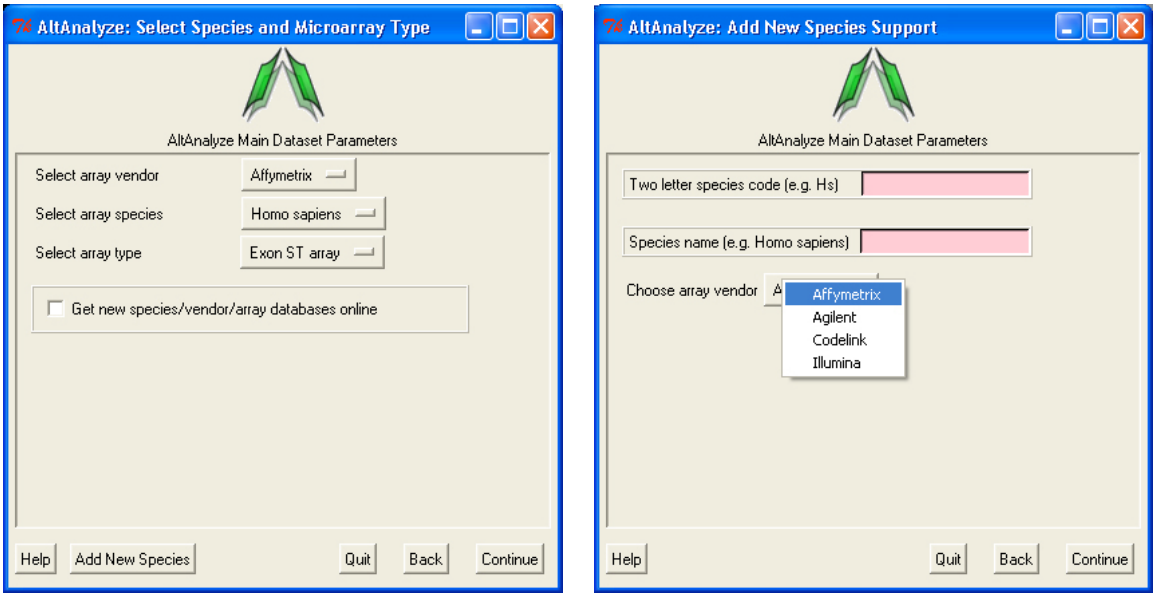


Figure 2.1. Select a Species and Array Type for Analysis. A) Options for selecting currently supported species, array vendors and array types and B) adding information for a non-supported species.

3) Select species and array – Next, the user must select a species, array vendor and array type for analysis. After selecting these options click “Continue”. If the species of interest is unavailable select the button “Add New Species Support”. Here you can add a two letter species code, full name and array vendor that will be used for all current and future analyses (Figure 2.1 B).

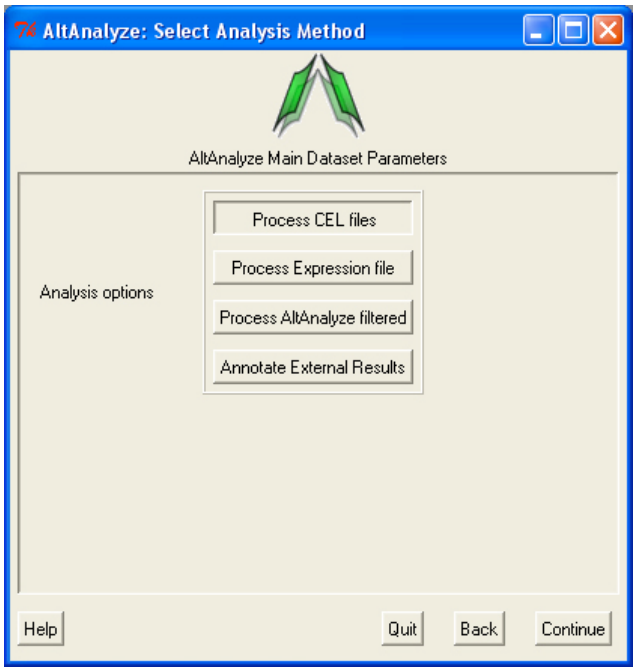
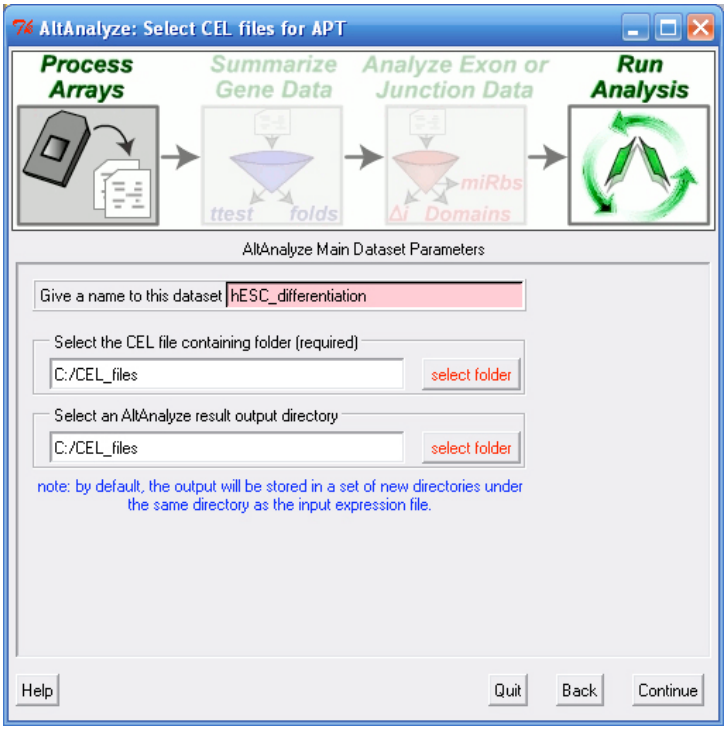


Figure 2.2. Select the Analysis Type. Options available for the select species and array type. AltAnalyze filtered is only available for alternative exon analyses (e.g., Affymetrix Exon ST 1.0 array or AltMouse A).

4) Select analysis option– In this window, the user must select the type data being analyzed. There are four main types of data; A) CEL files, B) Expression files, C) AltAnalyze filtered files and D) results from 3rd party applications (External Results). Processing of CEL files will produce the two file types (expression and DABG). Processing of Expression files, allows the user to select tab-delimited text files where the data has already been processed (e.g. RMA), which will also produce AltAnalyze filtered files. AltAnalyze filtered files are written for any splicing array analysis (not for gene expression only arrays). These later files allow the user to directly perform splicing analyses, without performing the previous steps. The AltAnalyze filtered files are stored to the folder

"AltExpression" under the appropriate array and species directories in the user output folder. Since CEL file normalization and array filtering and summarization can take a considerable amount of time (depending on the number of arrays), if re-performing an alternative exon analysis with different parameters, it is recommended that the user select the "Expression files" or "AltAnalyze filtered files", depending on which options the user wants to change. Users can also import lists of regulated probe sets with statistics obtained from a 3rd party application (e.g., JETTA) other than AltAnalyze.

A



B

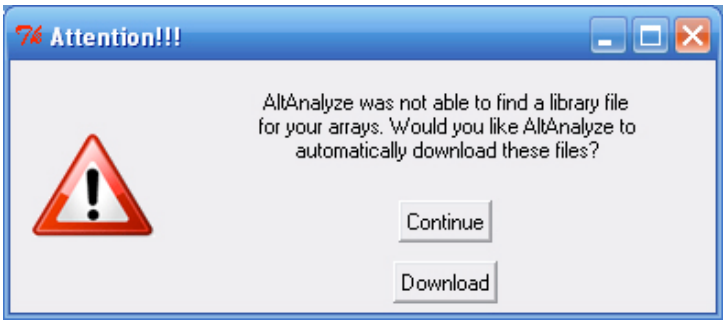


Figure 2.3. Select Folder and File Locations. (A) To properly run RMA using the APT software (included with AltAnalyze), the user must select a valid folder containing CEL files. (B) The first time you analyze a certain type of array you will be prompted to download a library file(s) for that array. For some arrays, you will be prompted for AltAnalyze to download these plus annotation files for you. Otherwise, you will be prompted by the program for such files.

5) Processing Affymetrix CEL Files - If you selected the “CEL files” data type from “Main Dataset Parameters”, you will be presented with a new window for selecting the location of your CEL files and desired output directory. Clicking the “select folder” icon will allow you to browse your hard-drive to select the folder with your CEL files. You can double-check the correct directory is selected by looking at the adjacent text display. This window will be followed by an indicator window that will automatically download the library and annotation files for that array (human, rat and mouse only). If the array type is unrecognized and you do not already have Affymetrix library files for your array (e.g. PGF or CDF), you will need to download these files from the Affymetrix website. To do so, select the link at the bottom left side of this window named “Download Library Files”. Select the array type being analyzed from the web page and select the appropriate library files to download and extract to your computer (requires an Affymetrix

username and password) (Figure 2.3 B).

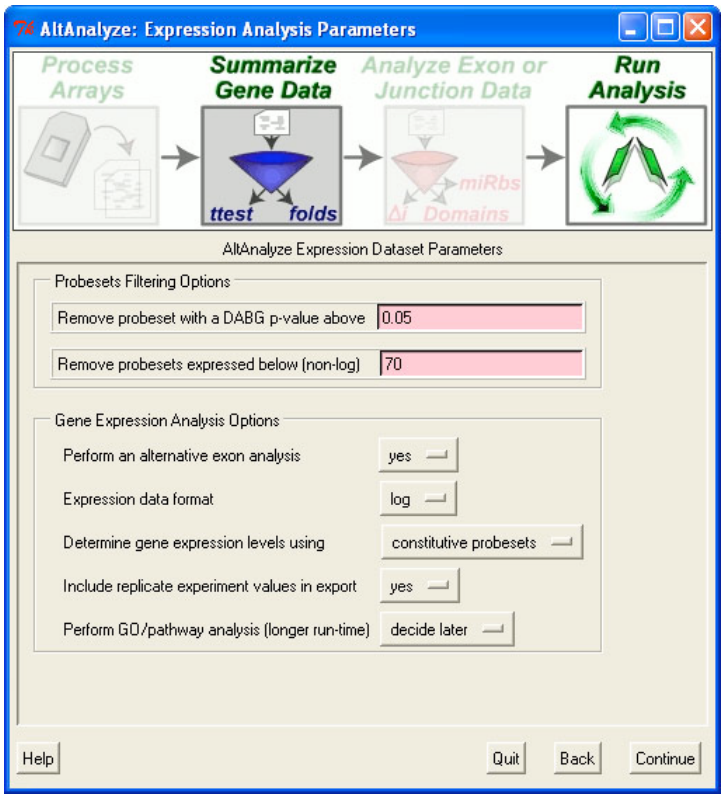
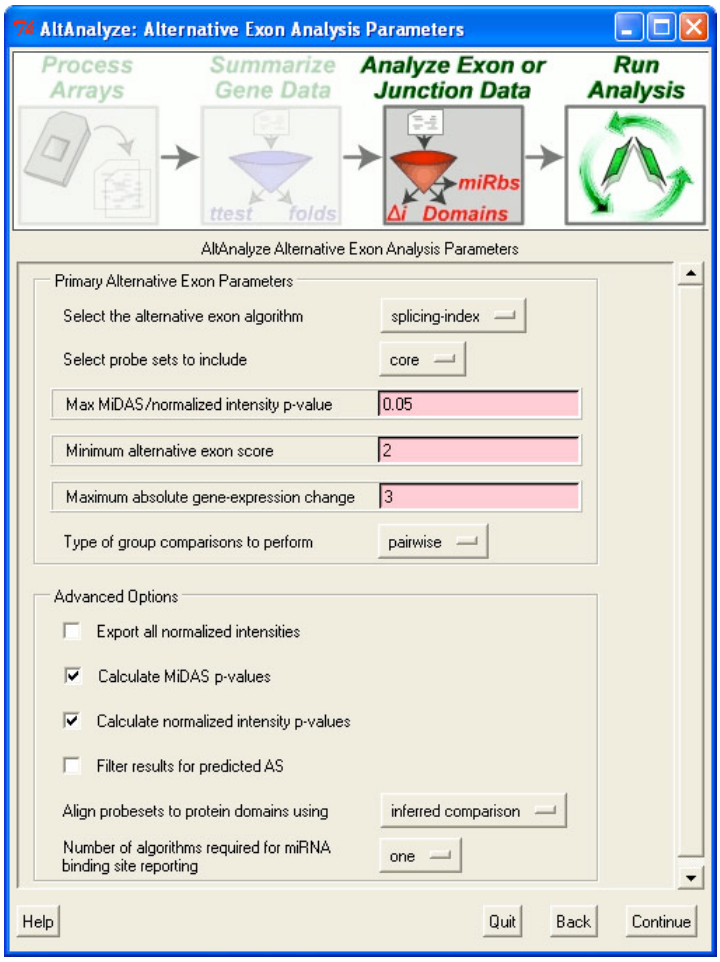


Figure 2.4. Select Summarization and Filtering Options. Users are presented with options for filtering probe sets for alternative exon analyses (DAGB and mean group expression) and options for how to derive gene expression values.

6) Summarizing Gene Data and Filtering For Expression – After CEL file data summarization, a number of options are available for summarizing gene level expression data and filtering out probe sets prior to alternative exon analysis. For splicing arrays, AltAnalyze calculates a “gene-expression” value based on the mean expression of all “core” (Affymetrix core probe sets and those aligning to known transcript exons) or “constitutive” (probe sets aligning to those exon regions most common among all transcripts) probe sets that have a mean DABG p-value less than and a mean expression value greater the user indicated

thresholds for each gene. These values are used to report predicted gene expression changes (independent of alternative splicing) for all user-defined comparisons (see following section). In addition, fold changes and ttest p-values are calculated for each of these group comparisons. These statistics along with several types of gene annotations exported to a file in the folder "ExpressionOutput" in the user-defined results directory. Along with this tab-delimited text file, a similar file with those values most appropriate for import into the pathway analysis program GenMAPP will also be produced (Figure 5.1). For splicing analyses, probe sets with user defined splicing cutoffs (expression and DABG p-values) will be retained for further analysis (see section 5.2 – ExpressionBuilder algorithm).

A


AltAnalyze: Alternative Exon Analysis Parameters
Process Arrays
Summarize Gene Data
Analyze Exon or Junction Data
Run Analysis
ttest
folds
miRbs
Δi
Domains
AltAnalyze Alternative Exon Analysis Parameters
Primary Alternative Exon Parameters
Select the alternative exon algorithm
splicing-index
Select probe sets to include
core
Max MiDAS/normalized intensity p-value
0.05
Minimum alternative exon score
2
Maximum absolute gene-expression change
3
Type of group comparisons to perform
pairwise
Advanced Options
Export all normalized intensities
Calculate MiDAS p-values
Calculate normalized intensity p-values
Filter results for predicted AS
Align probesets to protein domains using
inferred comparison
Number of algorithms required for miRNA binding site reporting
one
Help
Quit
Back
Continue

B

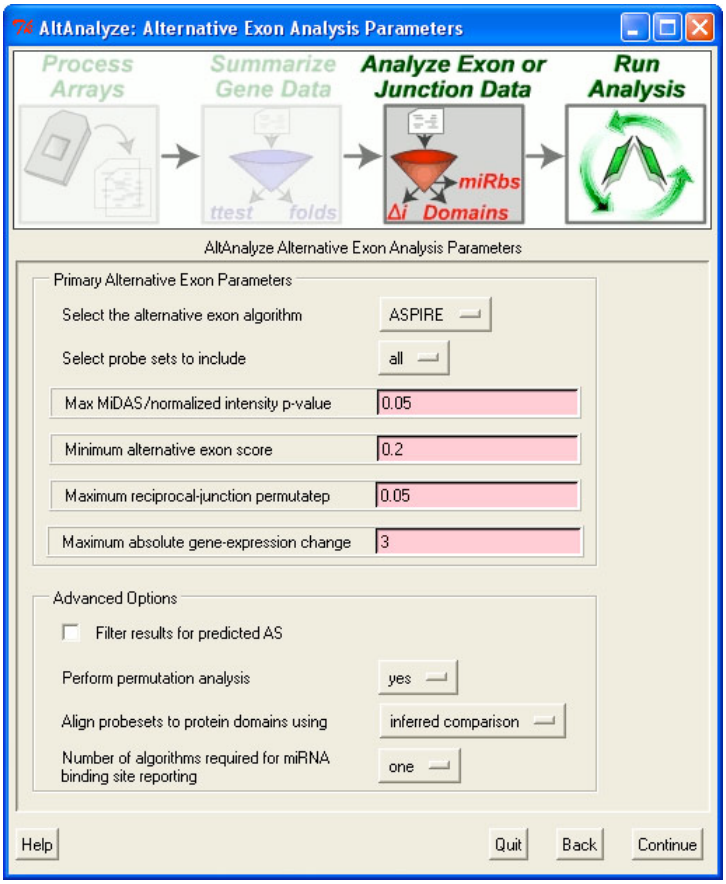


Figure 2.5. Select Summarization and Filtering Options. Splicing analysis options for (A) exon arrays and (B) junction arrays.

7) Select Alternative Exon Analysis Parameters – If using an exon-sensitive (e.g., Human Affymetrix 1.0 ST Exon array) or junction-sensitive microarray, the user will be presented with specific options for that microarray (Figure 2.5). These options include alternative exon analysis methods, statistical thresholds, and options for additional analyses (e.g., MiDAS), however, the default options are typically recommended. These include restricting the analysis to a conservative set of probe sets (e.g., “core”) and changing the threshold of splicing statistics. Note: that AltAnalyze’s “core” includes any probe set associated with a known

exon. When complete, the user can select “Continue” in AltAnalyze to incorporate these statistics into the analysis.

A **B**

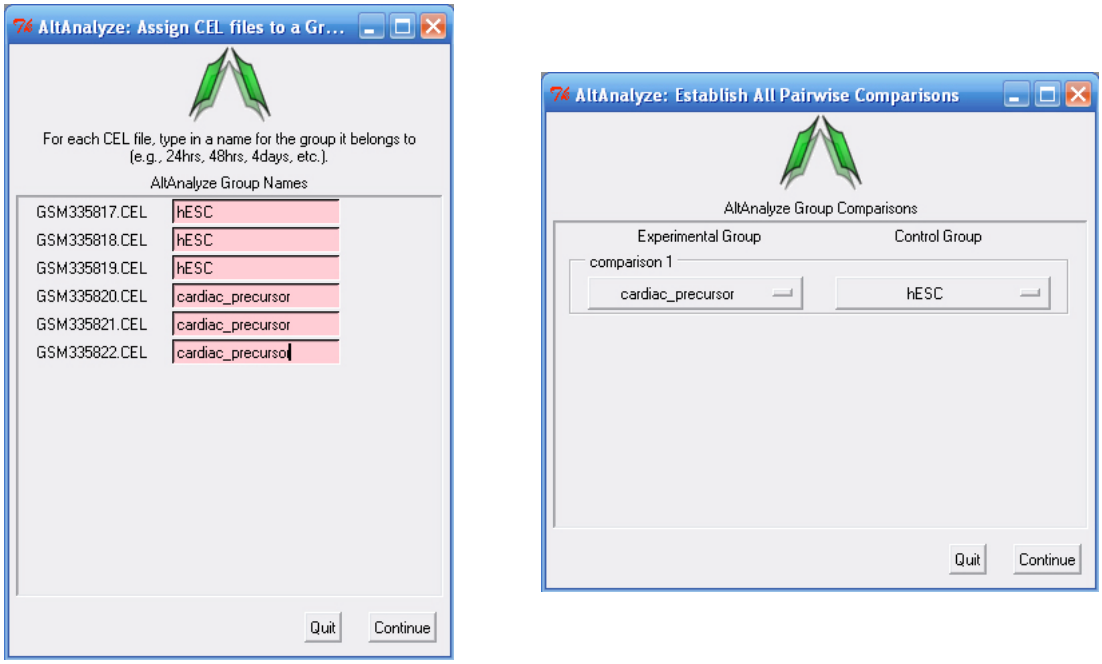


Figure 2.6. Establish Groups and Comparisons. (A) Enter a name for each group for all samples. (B) Enter all group comparisons for any possible pair-wise group comparisons (in this case there is only one).

8) Assigning Groups to Samples – When analyzing a dataset for the first time, the user will need to establish which samples correspond to which groups. Type in the name of the group adjacent to each sample name from in your dataset (Figure 2.6 A).
9) Establishing Comparisons between Groups – Once sample-to-group relationships are added, the user can list which comparisons they wish to be performed (Figure 2.6 B). For splicing and non-splicing arrays, folds and p-values will be calculated for each comparison for the gene expression summary file. For

splicing arrays, each comparison will be run in AltAnalyze to identify alternative exons. Thus, the more pair-wise comparisons the longer the analysis. If the user designates to compare “all groups” and not designate a pairwise comparison, this window will not be displayed.

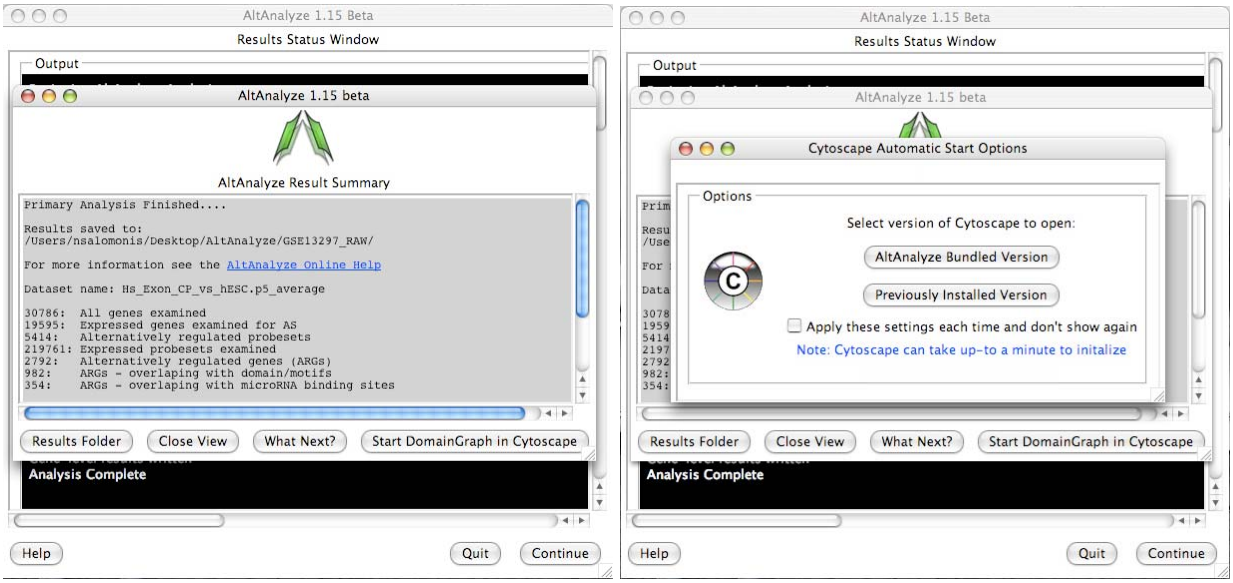


Figure 2.8. AltAnalyze Status. The AltAnalyze status window will appear once all user options are defined. Analysis run-time will depend on the number of samples, comparisons and array type. By Selecting the option “Start DomainGraph in Cytoscape”, users can immediately proceed to DomainGraph analysis.

10) AltAnalyze Status Window - While the AltAnalyze program is running, several intermediate results files will be created, including probe set, gene and dataset level summaries (Section 2.4). The results window (Figure 2.8) will indicate the progress of each analysis as it is running. When finished, AltAnalyze will prompt the user that the analysis is finished and a new “Continue” button will appear. A summary of results appears containing a basic summary of results from the

analysis. This window contains buttons that will open the folder containing the results and suggestions for downstream interpretation and analysis. Selecting the button “Start DomainGraph in Cytoscape” will allow the user to directly open a bundled version of Cytoscape and DomainGraph (Section 8). In addition to viewing the program report, this information is written to a log text file in the user-defined output directory.

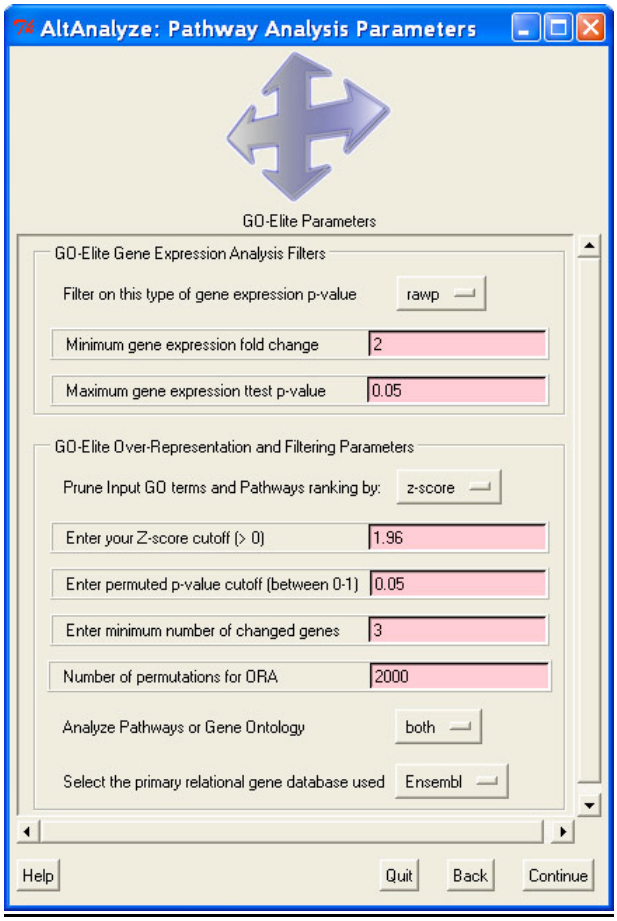


Figure 2.9. Perform Pathway Over-representation Analysis. Options to analyze both differentially expressed and alternative expressed genes from AltAnalyze summary statistics. Options include how stringent the gene expression statistics are and methods for redundancy filtering between Gene Ontology terms from the program GO-Elite.

11) If the user selected “decide later” for the analysis option “Perform GO-Elite Pathway Analysis”, the user will have the option to run GO-Elite after run completion. A similar summary results window as above will also appear with the GO-Elite pathway and Gene Ontology results.

2.3 Running AltAnalyze from Command-Line

In addition to using the default AltAnalyze graphical user interface (GUI), AltAnalyze can be run by command line options by calling the python source code in a terminal window or through other remote services. This option can be used to run AltAnalyze on a remote server, to batch script AltAnalyze services or avoid having to select specific options in the GUI. To do this, the user or program passes specific flags to AltAnalyze to direct it where files to analyze are, what options to use and where to save results. Note: the same alternative exon options exist for Gene and Exon 1.0 arrays.

Examples:

(Analyzing CEL files – Affymetrix 3’ array using default options and GO-Elite)

```
python AltAnalyze.py --species Mm --arraytype gene --celdir
"C:/CELFiles" --output "C:/CELFiles" --expname
"CancerCompendium"
```

(Analyzing CEL files – Exon 1.0 array using default options)

```
python AltAnalyze.py --species Hs --arraytype exon --celdir
"C:/CELFiles" --output "C:/CELFiles" --expname
"CancerCompendium"
```

(Analyzing CEL files – Exon 1.0 array using custom options)

```
python AltAnalyze.py --species Hs --arraytype exon --celdir
"C:/CELFiles" --output "C:/CELFiles" --expname
"CancerCompendium" --runGOElite no --dabgp 0.01 --rawexp 100 --
avgallss yes ––noxhyb yes --analyzeAllGroups "all groups" ––
GEcutoff 4 --probetype core --altp 0.001 --altmethod FIRMA ––
altscore 8 --exportnormexp yes ––runMiDAS no --ASfilter yes --
mirmethod "two or more" --calcNIp yes
```

(Analyzing Expression file – Gene 1.0 array using default options, without GO-Elite)

```
python AltAnalyze.py --species Mm --arraytype gene --expdir
"C:/CELFiles/ExpressionInput/exp.CancerCompendium.txt" --statdir
"C:/CELFiles/ExpressionInput/stats.CancerCompendium.txt" --
output "C:/CELFiles"
```

(Analyzing Filtered Expression file – Exon 1.0 array using default options)

```
python AltAnalyze.py --species Hs --arraytype exon --filterdir
"C:/CELFiles/Filtered/Hs_Exon_prostate_vs_lung.p5_average.txt" -
-output "C:/CELFiles"
```

(Annotate External Probe set results – Exon 1.0 array using default options)

```
python AltAnalyze.py --species Rn --arraytype exon --annotatedir
"C:/JETTA_Results/Hs_tumor_progression.txt" --output
"C:/JETTA_Results" --runGOElite yes
```

(Filter AltAnalyze results with predefined IDs – Exon 1.0 array using default options)

```
python AltAnalyze.py --species Mm --arraytype gene --celdir
"C:/CELFiles" --output "C:/CELFiles" --expname
"CancerCompendium" --returnAll yes
```

(Updating the database for a species)

```
python AltAnalyze.py --species Mm --update Official --version
EnsMart54
```

(Operating System Example Folder Locations)

PC: `"C:/CELFiles"`

Mac OSX: `"/root/user/admin/CELFiles`

Linux: `"/hd3/home/admin/CELFiles`

Primary Analysis Variables

No default value for these variables is given and must be supplied by the user if running an analysis. For example, if analyzing CEL files directly in AltAnalyze, you must include the flags ––species `––arraytype ––celdir ––expname` and `––output`, with corresponding values. Likewise, when analyzing an existing expression file you must include the flags `––species ––arraytype ––expdir` and `––output`. Most of the variable values are file or folder locations. These variable values will differ based on the directory path of your files and operating system (e.g, linux has a distinct path structure than windows – see above examples). The variable name used in the AltAnalyze source code for each flag is indicated below.

Universally Required Variables

`--arraytype`: long variable name “array_type”. No default value for this variable. Options are **exon**, **gene**, **AltMouse** and **“3’array”**. This variable indicates the general

array type correspond to the input CEL files or expression file. An example exon array is the Mouse Affymetrix Exon ST 1.0 array, an example gene array is the Mouse Affymetrix Gene ST 1.0 array and example 3'array is the Affymetrix Mouse 430 version 2.0 array. See Affymetrix website for array classifications.

`--species`: long variable name "cel_file_dir". No default value for this variable. Species codes are provided for this variable (e.g., Hs, Mm, Rn). Additional species can be added through the graphic user interface.

`--output`: long variable name "output_dir". No default value for this variable. Required for all analyses. This designates the directory which results will be saved to.

Analysis Specific Required Variables

`--expname`: long variable name "exp_name". No default value for this variable. Required when analyzing CEL files. This provides a name for your dataset. This name must match any existing groups and comps files that already exist. The groups and comps file indicate which arrays correspond to which biological groups and which to compare. These files must exist in the designated output directory in the folder "ExpressionInput" with the names "groups.*expname*.txt" and "comps.*expname*.txt" where expname is the variable defined in this flag. Alternatively, the user can name their CEL files such that AltAnalyze can directly determine which group they are (e.g., wildtype-1.CEL, cancer-1.CEL, cancer-2.CEL). See http://www.altanalyze.org/manual_groups_comps_creation.htm and http://www.altanalyze.org/automatic_groups_comps_creation.htm for more information

`--celdir`: long variable name "cel_file_dir". No default value for this variable. Required when analyzing CEL files. This provides the path of the CEL files to analyze. These must all be in a single folder.

`--expdir`: long variable name “input_exp_file”. No default value for this variable. Required when analyzing a processed expression file. This provides the path of the expression file to analyze.

`--statdir`: long variable name “input_stats_file”. No default value for this variable. Optional when analyzing a processed expression file. This provides the path of the DABG p-value file for the designated expression file to analyze (see `–expdir`).

`--filterdir`: long variable name “input_filtered_dir”. No default value for this variable. Required when aalyzing an AltAnalyze filtered expression file. This provides the path of the AltAnalyze filtered expression file to analyze.

`--cdfdir`: long variable name “input_cdf_file”. No default value for this variable. Required when directly processing *some* CEL file types. This variable corresponds to the location of the Affymetrix CDF or PGF annotation file for the analyzed array. If you are analyzing an exon, gene, AltMouse or human, mouse or rat 3’arrays, AltAnalyze has default internet locations for which to download these files automatically, otherwise, you must download the compressed CDF file from the Affymetrix website (support), decompress it (e.g., WinZip) and reference it’s location on your hard-drive using this flag. If you are unsure whether AltAnalyze can automatically download this file, you can try to exclude this variable and see if annotations are included in your gene expression results file.

`--csvdir`: long variable name “input_annotation_file”. No default value for this variable. Required when analyzing *some* expression files or CEL file types. This variable corresponds to the location of the Affymetrix CSV annotation file for the analyzed array. If you are analyzing an exon, gene, AltMouse or human, mouse or rat 3’arrays, AltAnalyze has default internet locations for which to download these files automatically, otherwise, you must download the compressed CSV file from the Affymetrix website (support), decompress it (e.g., WinZip) and reference it’s location on your hard-drive using this flag. If you are unsure whether AltAnalyze can automatically

download this file, you can try to exclude this variable and see if annotations are included in your gene expression results file.

`--annotatedir`: long variable name “external_annotation_dir”. No default value for this variable. Required when annotating a list regulated probe sets produced outside of AltAnalyze. This variable corresponds to the location of the directory containing one or more probe set files. These files can be in the standard JETTA export format, or otherwise need to have probe set IDs in the first column. Optionally, these files can have an associated fold change and p-value (2nd and 3rd columns) which will be reported in the results file.

`--groupdir`: long variable name “groups_file”. No default value for this variable. Location of an existing group file to be copied to the directory in which the expression file is located or will be saved to.

`--compdir`: long variable name “comps_file”. No default value for this variable. Location of an existing comps file to be copied to the directory in which the expression file is located or will be saved to.

Optional Analysis Variables

These variables are set as to default values when not selected. The default values are provided in the configurations text file in the Config directory of AltAnalyze (default-**.txt) and can be changed by editing in a spreadsheet program.

GO-Elite Analysis Variables

AltAnalyze can optionally subject differentially or alternatively expressed genes (AltAnalyze and user determined) to an over-representation analysis (ORA) along Gene Ontology (GO) and pathways (WikiPathways) using the program GO-Elite. GO-Elite is seamlessly integrated with AltAnalyze and thus can be run using default parameters

either the graphic user interface or command line. To run GO-Elite using default parameters in command line mode, include the first flag below with the option **yes**.

`--runGOElite`: long variable name “run_GOElite”, default value for this variable: **no**. Used to indicate whether to run GO-Elite analysis following AltAnalyze. Indicating **yes** would prompt GO-Elite to run.

`--mod`: long variable name “mod”, default value for this variable: **Ensembl**. Primary gene system for Gene Ontology (GO) and Pathway analysis to link Affymetrix probe sets and other output IDs to. Alternative values: **EntrezGene**.

`--elitepermut`: long variable name “goelite_permutations”, default value for this variable: **2000**. Number of permutation used by GO-Elite to calculate an over-representation p-value.

`--method`: long variable name “filter_method”, default value for this variable: **z-score**. Sorting method used by GO-Elite to compare and select the top score of related GO terms. Alternative values: **“gene number” combined**

`--zscore`: long variable name “z_threshold”, default value for this variable: **1.96**. Z-score threshold used following over-representation analysis (ORA) for reported top scoring GO terms and pathways.

`--elitepval`: long variable name “p_val_threshold”, default value for this variable: **0.05**. Permutation p-value threshold used ORA analysis for reported top scoring GO terms and pathways.

`--dataToAnalyze`: long variable name “resources_to_analyze”, default value for this variable: **both**. Indicates whether to perform ORA analysis on pathways, Gene Ontology terms or both. Alternative values: **Pathways** or **Gene Ontology**

`--num`: long variable name “change_threshold”, default value for this variable: **3**. The minimum number of genes regulated in the input gene list for a GO term or pathway after ORA, required for GO-Elite reporting.

`--GEelitepval`: long variable name “ge_pvalue_cutoffs”, default value for this variable: **0.05**. The minimum t-test p-value threshold for differentially expressed genes required for analysis by GO-Elite.

`--GEelitefold`: long variable name “ge_fold_cutoffs”, default value for this variable: **2**. The minimum fold change threshold for differentially expressed genes required for analysis by GO-Elite. Applied to any group comparisons designated by the user.

AltAnalyze Expression Filtering and Summarization

These variables are used to determine the format of the expression data being read into AltAnalyze, the output formats for the resulting gene expression data and filtering thresholds for expression values prior to alternative exon analysis. Since AltAnalyze can process both convention (3’array) as well as splicing arrays (exon, gene or AltMouse), different options are available based on the array type.

Universal Array Analysis Variables

`--logexp`: long variable name “expression_data_format”, default value for this variable: **log**. This is the format of the input expression data. If analyzing CEL files in AltAnalyze or in running RMA or GCRMA from another application, the output format of the expression data is log 2 intensity values. If analyzing MAS5 expression data, this is **non-log**.

`--inclraw`: long variable name “include_raw_data”, default value for this variable: **yes**. When the value of this variable is **no**, all columns that contain the expression intensities for individual arrays are excluded from the results file. The remaining columns are calculated statistics (groups and comparison) and annotations.

Exon or AltMouse Array Specific Variables

`--dabgp`: long variable name “dabg_p”, default value for this variable: **0.05**. This p-value corresponds to the detection above background (DABG) value reported in the

“stats.” file from AltAnalyze, generated along with RMA expression values. A mean p-value for each probe set for each of the compared biological groups with a value less than this threshold will be excluded, both biological groups don’t meet this threshold for a non-constitutive probe set or if one biological group does not meet this threshold for constitutive probe sets.

`--rawexp`: long variable name “expression_threshold”, default value for this variable: **70**. This value is the non-log RMA average intensity threshold for a biological group required for inclusion of a probe set. The same rules as the `--dabgp` apply to this threshold, accept that values below this threshold are excluded when the above rules are not met.

`--avgallss`: long variable name “avg_all_for_ss”, default value for this variable: **no**. Indicating **yes**, will force AltAnalyze to use all core (Affymetrix annotated core in addition to any exon aligning probe sets) rather than probe sets that align to predicted constitutive exons to determine transcriptional activity of gene. This option only applies to both the gene expression export file and to the alternative exon analyses.

`--runalt`: long variable name “perform_alt_analysis”, default value for this variable: **yes**. Designating **no** for this variable will instruct AltAnalyze to only run the gene expression analysis portion of the program, but not the alternative exon analysis portion.

AltAnalyze Expression Filtering and Summarization

Universal Array Analysis Variables

`--altmethod`: long variable name “analysis_method”, default value for this variable: **splicing-index** (exon and gene), **FIRMA** (exon and gene)and **ASPIRE** (AltMouse). These are the splicing algorithm default algorithms for either exon or AltMouse arrays. Currently, only one algorithm is available for exon arrays while **linearregres** is also available for AltMouse arrays.

`--altp`: long variable name “p_threshold”, default value for this variable: **0.05**. This variable is the p-value threshold for reporting alternative exons. This variable applies to both the MiDAS and splicing-index p-values.

`--probetype`: long variable name “filter_probe set_types”, default value for this variable: **core**. This is the class of probe sets to be examined by the alternative exon analysis. Other options include, **extended** and **full**.

`--altscore`: long variable name “alt_exon_fold_variable”, default value for this variable: **2** (exon) and **0.2** (AltMouse).This is the corresponding threshold for the default algorithms listed under `--altmethod`.

`--GEcutoff`: long variable name “gene_expression_cutoff”, default value for this variable: **3**. This value is the non-log gene expression threshold applied to the change in gene expression (fold) between the two compared biological groups. If a fold change for a gene is greater than this threshold it is not reported among the results, since gene expression regulation may interfere with detection of alternative splicing.

`--analyzeAllGroups`: long variable name “analyze_all_conditions”, default value for this variable: **pairwise**. This variable indicates whether to only perform pair-wise alternative exon analyses (between two groups) or to analyze all groups, without specifying specific comparisons. Other options are **“all groups”** and **both**.

`--altpermutep`: long variable name “permute_p_threshold”, default value for this variable: **0.05**. This is the permutation p-value threshold applied to AltMouse array analyses when generating permutation based alternative exon p-values. Alternative exon p-values can be applied to either ASPIRE or linregress analyses.

`--altpermute`: long variable name “perform_permutation_analysis”, default value for this variable: **yes**. This option directs AltAnalyze to perform the alternative exon p-value analysis for the AltMouse array (see `–altpermutep`).

`--exportnormexp`: long variable name “export_splice_index_values”, default value for this variable: **no**. This option directs AltAnalyze to export the normalized intensity

expression values (probe set intensity/constitutive expression) for all analyzed probe sets rather than perform the typical AltAnalyze analysis when its value is **yes**. This step can be useful for analysis of exon array data outside of AltAnalyze and comparison of alternative exon profiles for many biological groups (e.g., expression clustering).

`--runMiDAS`: long variable name “run_MiDAS”, default value for this variable: **yes**. This option directs AltAnalyze to calculate and filter alternative exon results based on the MiDAS p-value calculated using the program Affymetrix Power Tools.

`--calcNIp`: long variable name “calculate_normIntensity_p”, default value for this variable: **yes**. This option directs AltAnalyze to filter alternative exon results based on the t-test p-value obtained by comparing either the normalized intensities for the array groups examined (e.g., control and experimental) (splicing-index) or a t-test p-value obtained by comparing the FIRMA scores for the arrays in the two compared groups.

`--mirmethod`: long variable name “microRNA_prediction_method”, default value for this variable: **one**. This option directs AltAnalyze to return any microRNA binding site predictions (default) or those that are substantiated by multiple databases (**two or more**).

`--ASfilter`: long variable name “filter_for_AS”, default value for this variable: **no**. This option directs AltAnalyze to only analyze probe sets for alternative expression that have an alternative-splicing annotation (e.g., cassette-exon, alt-5’, alt-3’, intron-retention), when set equal to **yes**.

`--returnAll`: long variable name “return_all”, default value for this variable is **no**. When set to **yes**, returns all un-filtered alternative exon results by setting all associated filtering parameters to the lowest stringency values. This is equivalent to providing the following flags: `--dabgp` 1 `--rawexp` 1 `--altp` 1 `--probetype` full `--altscore` 1 `--GEcutoff` 10000. Since this option will output all alternative exon scores for all Ensembl annotated probe sets, the results file will be exceptionally large (>500,000 lines), unless the user has saved previously run alternative exon results (e.g., MADS) to

the directory “AltDatabase/filtering” in the AltAnalyze program directory, with a name that matches the analyzed comparison. For example, if the user has a list of 2,000 MADS regulated probe sets for cortex versus cerebellum, then the MADS results should be saved to “AltDatabase/filtering” with the name “Cortex_vs_Cerebellum.txt” and in AltAnalyze the CEL file groups should be named Cortex and Cerebellum and the comparison should be Cortex versus Cerebellum. When the filename for a file in the “filtering” directory is contained within the comparison filename (ignoring “.txt”), only these probe sets will be selected when exporting the results. This analysis will produce a results file with all AltAnalyze statistics (default or custom) for just the selected probe sets, independent of the value of each statistic.

AltAnalyze Database Updates

Universal Array Analysis Variables

`--update`: long variable name “update_method”, default value for this variable: **empty**. Setting this flag equal to **Official**, without specifying a version, will download the most up-to-date database for that species. Other options here are used internally by AltAnalyze.org for building each new database. See the method “commandLineRun” in AltAnalyze.py for more details.

`--version`: long variable name “ensembl_version”, default value for this variable: **current**. Setting this flag equal to a specific Ensembl version name (e.g. **EnsMart49**) supported by AltAnalyze will download that specific version for the selected species, while setting this to current will download the current version.

2.4 AltAnalyze Analysis Options

There are a number of analysis options provided through the AltAnalyze interface. This section provides an overview of these options for the different compatible array

analyses (gene expression arrays, exon arrays and junction arrays). For new users, we recommend first running the program with the pre-set defaults and then modifying the options as necessary.

Selecting the Microarray Type and Species

When beginning AltAnalyze, the user can select from a variety of species and array types. Only array manufacturers and array types supported for each downloaded species will be displayed. When multiple gene database versions are installed, a drop-down box at the top of this screen will appear that allows the user to select different gene database versions. These gene databases include all resources necessary for gene annotation, alternative exon analysis (where applicable) and Gene Ontology and pathway analysis. Microarray file normalization and summarization options are only available for Affymetrix arrays, while annotation and statistical analyses are supported for all Affymetrix, Agilent, Illumina and CodeLink arrays supported by Ensembl. At the bottom of this interface is a check-box that the user can select to download updated species gene databases, which will bring-up the database downloader window.

Selecting the Microarray Analysis Method

After the user has selected the species of interest, they must choose what type of data they will next be analyzing. Data can consist of; 1) Affymetrix CEL files, 2) an already processed expression text files, 3) properly formatted and filtered AltAnalyze expression input text file or 4) restricted list of probe sets to be directly analyzed. If beginning with Affymetrix CEL files all three of these file types are produced in series (see following section) and automatically processed without any user intervention. If all CEL files from your study already been previously in AltAnalyze or in another program, the user can load this file be selecting the option “expression file” and choosing this text file from your computer. This file needs to contain data from arrays corresponding to at least two

biological groups. Users may wish to re-analyze these files to change their expression filtering parameters to be more or less stringent. For the two or more biological groups (see how to define in Figure 2.6), AltAnalyze will segregate the raw data based on the user-defined pair-wise group comparisons and filter the containing probe sets based on whether they match the user-defined thresholds for inclusion and are associated with Ensembl genes (see the below section: Expression Analysis Parameters). These files will be saved to the folder “AltExpression” in the user-defined output directory. These files can be later selected by choosing the option “AltAnalyze filtered”, if the user wishes to re-run or use different AltAnalyze alternative exon analysis options (see below section: Alternative Exon Analysis Parameters).

CEL File Summarization

CEL files are one of the file types produced after scanning an Affymetrix microarray. The CEL file is produced automatically from the DAT file (an image file, similar to a JPEG), by the Affymetrix software by overlaying a grid over the microarray florescent image and assigning a numeric value to each cell or probe. From this file, expression values for each probe set can be calculated and normalized for all arrays in the study using various algorithms.

When choosing to analyze CEL files in AltAnalyze, the user will be prompted to identify the folder containing the CEL files and the folder in which to save these other results to. The user will also need to assign a name to the dataset. These CEL files will be summarized using the RMA algorithm using the program Affymetrix Power Tools (APT). The APT C++ module “apt-probeset-summarize” is directly called by AltAnalyze when running AltAnalyze on a Mac, PC or Linux operating system. Unlike some other applications, APT is packaged with AltAnalyze and thus does not require separate installation. However, because it is a separate application there may be unknown compatibility issues that exist, depending on your specific system configuration and

account privileges. For human and mouse exon arrays, AltAnalyze also allows for the masking of probes with cross-hybridization potential, prior to running RMA. This is performed through an experimental APT function (--kill-list), masking probes that are indicated in files produced for the MADS application (http://biogibbs.stanford.edu/~yxing/MADS/Annotation.html) that cross-hybridize to an off-target transcript within 3bp mismatches and a person correlation coefficient > 0.55, as per the MADS recommendations. The probes with cross-hybridiation potential are indicated in the AltAnalyze directory "AltDatabase/Hs/exon/Hs_probes_to_remove.txt".

APT requires the presence of a library file(s) specific for that array. AltAnalyze will automatically determine the array type and can install these files if the user wishes (currently most human, rat and mouse arrays supported). If AltAnalyze does not recognize the specific array type or the user chooses to download these files themselves, they will need to select the appropriate files when prompted in AltAnalyze. For exon arrays, a PGF, CLF and antigenomic BGP file are required. These files will be automatically downloaded and installed if the user selects "Download" when prompted. For non-exon arrays, the appropriate CDF file will be downloaded. In addition to these library files, a NetAffx CSV annotation file will be downloaded that allows for addition of gene annotations (non-exon arrays) and Gene Ontology pathway annotations (all arrays). Once installed, AltAnalyze will recognize these files and automatically use them for all future analyses. Once the user selects the appropriate directories and files, the user will be prompted to select the remaining options in AltAnalyze, before APT is run. Once run, a tab-delimited text expression file will be produced for all probe sets on the array and a detection-above background (DABG) p-value file (exon array only).

Loading a Processed Expression File

If CEL files are processed outside of AltAnalyze, the user must save the resulting expression text file in tab-delimited format. It is alright if the first rows in the file have run information as long as they are proceeded by a pound sign (#).

Expression Analysis Parameters

The options presented in this interface (Figure 2.4) allow the user to determine what fields are present in the gene expression output file, what scale the data is in (e.g. logarithmic), which probe sets to use when calculating gene expression and how to filter probe sets for subsequent analyses.

1) Perform an alternative exon analysis - Selecting the option “just expression” will halt the analysis after the gene expression result file has been written, such that no splicing analysis is performed. This option is only available for splicing-sensitive arrays.
2) Expression data format - Indicates the format in which the CEL file summarized data has been written. When the CEL files have been processed by AltAnalyze, ExpressionConsole, APT, RMAExpress or through R, the file format will be logarithmic base 2 (log). If the user designates “not-log”, then expression values will be log base 2 (log_2) transformed prior to analysis.
3) Determine gene expression levels using - For splicing sensitive microarrays, the user has the choice to alter the way in which gene expression values are calculated and how to filter their probe set expression files prior to alternative exon analysis. When “yes” is selected for this option, all core probe sets (Affymetrix core annotated and any exon aligning probest) linked to a unique gene will be used to calculate a measure of gene expression by taking the mean expression of all probe set values. When the “no” is selected, only those probe sets that have been annotated as constitutive or common to the most isoforms will be used for gene expression calculation. In either case, only probe sets with

at least one array possessing a DABG p-value less than the user threshold will be retained (if a DABG p-value file is present). In order to exclude this threshold, set the minimum DABG p-value equal to 1.

4) Include replicate experimental values in the export - Instructs AltAnalyze whether to include the expression values associated with each CEL file in the output file. If not selected only the mean expression value of all CEL files for each biological group will be written.
5) Remove probesets with a DABG p-value above – When a DABG file has been produced (default when summarizing CEL files with AltAnalyze for exon-arrays), this option is applied. The default DABG p-value cutoff is $p<0.05$. This will filter out any non-constitutive probe set that has a mean DABG $p>0.05$ for both compared biological groups. For probe sets used in determining gene expression levels, both biological groups must have a DABG p < user-value. In order to exclude this option, you can remove the DABG file (contains the prefix “stats.”), or set this value equal to 1.
6) Remove probesets expressed below (non-log) – This statistic is treated the same as the DABG p-value cutoff except in that probe sets with a mean expression value less than this cutoff will be excluded. The same rules apply to this value as to the DABG value, where both variables must be true for probe set inclusion (e.g., $p<0.05$ and mean expression > 70). To exclude this option, set the default value to 100,000 (greater than the maximum intensity value of most expression summaries).
7) Perform GO/pathway analysis (longer run-time) – Choosing “decide later” will allow the user to view the GO-Elite pathway and Gene Ontology over-representation analysis options after the main gene expression and/or alternative exon analysis is run. This will prompt a separate status window and results summary window displaying over-representation statistics for pathway analysis. If

the option “run immediately” is selected, GO-Elite will run right away without a separate window. Please note, GO-Elite analysis can take up to an hour per criterion when using the default parameters (e.g., 2000 permutations and analysis of both Gene Ontology and Pathways). For more details on this analysis see: http://www.genmapp.org/go_elite/help_main.htm

Alternative Exon Analysis Parameters

The options presented in this menu (Figure 2.5) instruct AltAnalyze what statistical methods to use when determining alternative exon expression, which probe sets to select for analysis, what domain-level and miR-BS analyses to perform and what additional values to export for analyses in other tools. Details on each analysis algorithm are covered in detail in section 3.2.

1) Select the alternative exon algorithm – For exon arrays, the splicing index method is the only method currently provided, however, for junction arrays, several methods are available. These methods are used to calculate an alternative exon score, relative to gene expression levels. The default value for splicing-index analysis is 2, indicating that an adjusted expression difference greater than two fold (up- or down-regulated) is required for the probe set to be reported. Based on the algorithm, different values and scales will apply. For junction-arrays, the ASPIRE algorithm default cutoff is 0.2, whereas the linear-regression algorithm is 2. For linear-regression (linearregres), a minimum value of 2 will select any linear-regression fold greater than 2 (result folds are reported in log 2 scale, however), up- or down-regulated, whereas ASPIRE’s scores ranging from -1 to 1. See algorithm descriptions for more details (section 3.2).
2) Minimum alternative exon score – This value will vary based on the alternative exon analysis method chosen (see above options).

3) Max MiDAS/normalized intensity p-value – This is the p-value cutoff applied to MiDAS and splicing-index or FIRMA ttest p-values for exon array analyses. Currently, the user cannot set different p-value thresholds for these two statistics. More on MiDAS can be found below and in section 3.2.
4) Select probe sets to include – This option is used to increase or decrease the stringency of the analysis. In particular, this option allows the user to restrict what type of probe sets are to be used to calculate an alternative exon score. In the case of junction arrays, this option includes the ability to merge the expression values of probe sets that measure the same differential inclusion of an exon (combined-junctions). For exon arrays, there are three options, “core”, “extended” and “full”. Although these are the same probe set class names used by Affymetrix to group probe sets, AltAnalyze uses a modification of these annotations. Specifically, probe sets with the core annotation include all Affymetrix core probe sets that specifically overlap with a single Ensembl gene (2) (based on genomic position) along with any probe set that overlaps with an Ensembl or UCSC exon (3). Likewise, extended and full probe sets are those remaining probe sets that also align to a single Ensembl gene, with the Affymetrix extended or full annotation.
5) Maximum absolute gene-expression change – This value indicates maximum gene expression fold change (non-log, up- or down-regulated) that is allowed for a gene to be reported as alternatively regulated. The default is 3-fold, up or down-regulated. This filter is used with assumption that alternative splicing is a less critical factor when a gene is highly differentially expressed.
6) Perform permutation analysis - (***junction arrays only***) This analysis reports a p-value that represents the likelihood of the observed alternative exon score occurring by change, after randomizing the expression values of all samples.

7) Maximum reciprocal-junction permutatep – (***junction arrays only***) This p-value cutoff applies to the permutation based alternative exon score p-values when performing ASPIRE or linearregress (see section 3.2).
8) Export all normalized intensities – This option can be used to compare alternative exon scores prior to filtering for biological multiple comparisons, outside of AltAnalyze. For example, if comparing multiple tissues, the user may wish to export all splicing-index or FIRMA scores for all tissue comparisons. The results will be stored to the AltResults folder in the user-defined output-directory.
9) Calculate MiDAS p-values – This statistic is analogous to the ttest p-value calculated during the splicing-index analysis (see section 3.2 for more details). If not selected, then only the splicing-index or FIRMA (depending on the user selection) fold and p-value will be used to filter alternative exon results.
10) Calculate normalized intensity p-values – Indicates whether to calculate the splicing-index or FIRMA ttest p-values and filter using the above threshold.
11) Filter results for predicted AS – This option instructs AltAnalyze to only include regulated probe sets in the output that have been assigned a valid splicing annotation (e.g., alternative-cassette exon) provided by AltAnalyze. These annotations exclude probe sets with no annotations or those with only an alternative N-terminal exon or alternative promoter annotation.
12) Align probesets to protein domains using – This option is used to restrict the annotation source for domain/feature over-representation analysis. If “direct-alignment” is chosen, only those probe sets that overlap with the genomic coordinates of a protein domains/features will be included in the over-representation analysis, otherwise, the inferred method is used (see section 3.2 for more details).

13) Number of algorithms required for miRNA binding site reporting – This option is used to filter out miR-BS predictions that only occur in one of the four miR-BS databases examined. For more miR database information see section 6.5.

14) Type of group comparison to perform –This option indicates whether to only perform pair-wise alternative exon analyses (between two groups) or to analyze all groups, without specifying specific comparisons.

2.5 Overview of Analysis Results

AltAnalyze will output two main types of files:

1) Gene expression (GE) summary
2) Alternative exon summary

Gene Expression Summary Data

The GE summary files are two files that contain all computed gene expression values from your dataset, saved to the folder "ExpressionOutput" in the user-defined output directory. The first is a file is a complete dataset summary file with the prefix "DATASET" followed by the user-defined dataset name containing all array expression values (gene-level for exon arrays), calculated group statistics (mean expression, folds, raw and adjusted t-test and f-test p-values) and gene annotations (e.g., gene symbol, description, Gene Ontology, pathway and some custom groups). For exon arrays, the gene expression values are derived from either probe sets that align to exons the are found in the largest number of transcripts (constitutive) and thus are informative for transcription or all core probe sets for that gene (see "Select expression analysis parameters", in Section 2.3). Constitutive probe sets are determined by finding discrete exon regions that are common to the most mRNA transcripts (Ensembl and UCSC) for all transcripts used in the AltAnalyze database build (section 6). When one or more

gene expression reporting probe sets have DABG p-values with at least one biological group with a mean value below the user defined threshold, these probe sets will be used to calculate gene expression, otherwise, all gene expression reporting probe sets will be used.

The second file, called the GenMAPP input file, contains a subset of columns from the dataset summary file for import into GenMAPP (4) or PathVisio (5) (http://www.pathvisio.org/PathVisio2). This file has the prefix “GenMAPP” and excludes all gene annotations and individual array expression values.

Alternative Exon Summary Data

These results are produced from all probe sets that may suggest alternative splicing, alterative promoter regulation, or any other variation relative to the gene expression for that gene (derived from comparisons file). When the user chooses to either analyze all groups rather than just pair-wise comparisons or both, the same output files will be produced but report MiDAS p-values comparing all conditions and the maximum possible splicing index fold between all conditions (Section 3.2). Each set of results corresponds to a single pair-wise comparison (e.g., cancer vs. normal) and will be named with the group names you assigned. Four sets of results files are produced in the end:

1) Probe set-level - Probe set-level statistics, exon annotations, AS/APS annotations, and functional predictions (protein, domain and miRNA binding site).
2) Gene-level – Gene-level summary of data in probe set-level file.
3) Domain-level – Over-representation analysis of gene-level domain changes due alternative exon regulation.
4) miRNA binding sites - Over-representation analysis of gene-level, predicted miRNA binding sites present in alternatively regulation exons.

5) Summary statistics file – Global statistics, reporting the number of genes alternatively regulated, number differentially expressed and summary protein association information (e.g, mean regulated protein length).

Each file is a tab delimited text file that can be opened, sorted and filtered in a spreadsheet program. These files are saved to the user-defined output directory under "AltResults/AlternativeOutput", all with the same prefix (pair-wise group comparisons). AltAnalyze will analyze all pair-wise comparisons in succession and combine the probe set-level and gene-level results into two additional separate files (named based on the splicing algorithm chosen).

Probe set- and Gene-Level Alternative Exon Result Files

The probe set-level file contains alternative exon data for either one probe set (exon-array) or reciprocal probe sets (junction array). This includes:

- Gene and probe set annotations (e.g., description, symbol, probe set ID, probe set exon ID, transcript clusters, links to Ensembl/UCSC exons, ordered exon-region IDs).
- Mean probe-set expression values for the regulated probe set(s).
- Gene expression changes and baseline expression.
- Statistical results (e.g., splicing-index score and p-value, MiDAS p-values, raw probe set p-value).
- Alternative exon annotations (e.g., splicing-events, alternative promoters, alternative annotation confidence score).
- Protein- and microRNA-level associations (e.g., associated IDs, sequence, pattern of regulation, regulated domains/microRNA binding sites).

The gene-level file contains a summary of the data at the gene level, with each row representing a unique gene. This file also includes:

- Gene Ontology and pathway information for each gene extracted from any Affymetrix CSV annotation files for that species present in the directory “AltDatabase/Affymetrix/*species*”.

Protein Feature and MicroRNA Binding Site Over Representation Files

Over-representation analyses, (files 3 and 4) have the same structure:

- Column A is the name of the miR-BS or protein domain (e.g., term).
- Column B is the number of unique genes associated with alternatively regulated probe sets for that term (aka Changed).
- Column C is the number of genes analyzed for over-representation that correspond to that term (aka Measured).
- Column D is the percentage Changed (Changed/Measured).
- Column E is the over-representation z-score (see Section 3 - Algorithms) for all unique genes aligning to the feature that are alternative regulated by the analysis that correspond to that term. A value of 1.96 is approximate to a p-value of 0.05 assuming a normal distribution.
- Column F is the permutation based p-value to assess the likelihood of the observed z-score occurring by chance over 2000 random permutations of measured probe sets.
- Column G is the Benjamini-Hochberg adjusted p-value of F, to take into account multiple hypothesis correction.
- Column H contains all gene symbols for all unique genes changed.

Section 3 – Algorithms

Multiple algorithms are available in AltAnalyze to identify individual probe sets (for exon arrays (EA)) or reciprocal probe sets (exon-exon junction array (JA)) that are differentially regulated relative to gene expression changes. These include the splicing index method (EA), MiDAS (EA), ASPIRE (JA) and Linear Regression (JA). In addition to these statistical methods, several novel methods are used to predict which alternative proteins correspond to a regulated exon, which protein domain/features differ between these and which RNA regulatory sequences differ between the associated transcripts (e.g., miR-BS).

3.1 Default Methods

The default options are stored in external text files in the folder "Config" as "defaults-expr.txt", "defaults-alt_exon.txt", and "defaults-funct.txt".

defaults-expr.txt	Default expression analysis options (Figure 2.4)
defaults-alt_exon.txt	Default alternative exon analysis options (Figure 2.5)
defaults-funct.txt	Default functional analysis options (Figure 2.5)

These options correspond to those found in the configuration file "options.txt". The user is welcome to modify the defaults and theoretically even the options in the "options.txt" file, however, care is required to ensure that these options are supported the by the program. Since AltAnalyze is an open-source program, it is feasible for the user to add new species and array support or to do so with AltAnalyze support. The default algorithms for the currently supported arrays are as follows:

Exon	splicing-index (score > 2 and t-test p<0.05), no MiDAS
AltMouse	ASPIRE (score > 0.2 and permute p<0.05)
3' array	NA

3.2 Algorithm Descriptions

Gene Expression Analysis

For the simple gene expression output files saved to the ExpressionOutput directory, several basic expression statistics are calculated. These statistics are performed for any user specified pair-wise comparisons (e.g., cancer versus normal) and for between all groups in the user dataset (e.g., time-points of differentiation). These statistics are comprised of the following: (1) rawp, (2) adjp, (3) log-fold, (4) fold change, (5) ANOVA rawp, (6) ANOVA adjp and () max log-fold. The rawp is a one-way analysis of variance (ANOVA) p-value calculated for each pair-wise comparison (two groups only). The adjp is the Benjamini-Hochberg (BH) 1995 adjusted value of the rawp. The log-fold is the log2 fold calculated by geometric subtraction of the experimental from the control groups for each pair-wise comparison. The fold change is the non-log2 transformed fold value. The ANOVA rawp is the same as the comparison rawp, but for all groups analyzed (note, this is reduced to the rawp when only two groups are analyzed). The ANOVA adj is the BH adjusted of the ANOVA rawp. The max log-fold is the log2 fold value between the lowest group mean and the highest group expression mean for all conditions in the dataset. These statistics are intended for further data filtering and prioritization in order to assess putative transcription differences between genes.

Splicing Index Method

This algorithm is described in detail in the following publications:
(6, 7). In brief, the expression value of each probe sets for a condition is converted to log space (if necessary). For each probe sets examined, its expression (log_2) is subtracted from the mean gene expression of value to create a gene expression

corrected log ratio (subtract instead of divide when these values are in log space). This value is the normalized intensity. This normalized intensity is calculated for each microarray sample, using only data from that sample. To derive the splicing-index value, the group normalized intensity of the control is subtracted from the experimental. This value is the change in exon-inclusion (delta I, δI, or splicing index fold change).

$$NI(probeset_i) = \left(\frac{probeset\ intensity}{expression\ level\ of\ gene} \right)$$

$$SI(probeset_i) = \log_2 \left(\frac{NI(probeset_i)_{sample1})}{NI(probeset_i)_{sample2})} \right)$$

A t-test p-value is calculated (two tailed, assuming unequal variance) by comparing these normalized intensity for all samples between the two experimental groups. A negative δI score of -1, thus indicates a two fold change in the expression of probe set, relative to the mean gene expression, with expression being higher in the experimental versus the control. When more than two-groups are being compared in AltAnalyze ("all groups" or "both options for "Type of group comparisons to perform"), the splicing-index value is calculated between the two biological groups with the lowest and highest normalized intensities. The two groups being compared is indicated in the alternative exon results file.

FIRMA Analysis

FIRMA or Finding Isoforms using Robust Multichip Analysis (8) is an alternative to the splicing-index approach, to calculate alternative splicing statistics. Rather than using the probe set expression values to determine differences in the relative expression of an exon for two or more conditions, FIRMA uses the residual values produced by the RMA algorithm for each probe, corresponding to a gene. The median

of the residuals for each probe set, for each array sample are compared to the median absolute deviation for all residuals and samples for the gene.

Although the core FIRMA methods are the same as the original implementation in R, AltAnalyze FIRMA differs in several important ways:

1) The probe composition of each gene is defined by the standard AltAnalyze core, extended or full probe set definitions, rather than the Affymetrix transcript cluster definitions. Thus, each probe must correspond to a single Ensembl gene to be analyzed.
2) While FIRMA scores for each sample and probe set are calculated, only summary statistics are reported in the standard AltAnalyze output files. This statistic is the average FIRMA score for all samples in the experimental group minus the average of the FIRMA scores in the designated baseline group. If no comparisons are specified, the two groups with the largest difference in scores are reported.
3) FIRMA scores are organized into groups for calculation of summary statistics (FIRMA fold change and t-test p-values). However, scores for each probe set and sample can be optionally exported.

Users can define whether to use the AltAnalyze core, extended or full annotations in the program interface or using the --probetype flag in the command-line interface. An unique numerical ID corresponding to each Ensembl gene and all associated gene probe sets for FIRMA analysis are stored in a metaprobeset file in the array annotation directory (e.g., AltDatabase/EnsMart54/Hs/exon/ Hs_exon_core.mps). This file is used to define gene level probe sets by the program APT.

Similar to splicing-index analysis of exon-tiling data, expression and DAGB filtering of probe set expression, FIRMA score p-value filtering and gene expression reporting/filtering based on either constitutive or all exon aligning probe sets. To export FIRMA scores for each probe set and sample, select the option “Export all normalized intensities” from the “Alternative Exon Analysis Parameters” window, or by using the flag `--exportnormexp` in the command-line interface.

MiDAS

The MiDAS statistic is described in detail in the white paper: www.affymetrix.com/support/technical/whitepapers/exon_alt_transcript_analysis_whitepaper.pdf. This analysis method is available from the computer program APT, mentioned previously. APT uses a series of text files to examine the expression values of each probe set compared to the expression of user supplied gene expression reporting probe sets. Since AltAnalyze uses only probe sets found to align to a single Ensembl gene, AltAnalyze creates it’s own unique numerical gene identifiers (different than the Affymetrix transcript clusters). When written, a conversion file is also written that allows AltAnalyze to translate from this arbitrary numerical ID back to an Ensembl gene ID. These relationships are stored in the following files along with the probe set expression values:

meta-Hs_Exon_cancer_vs_normal.txt	Relates probe set to gene
gene-Hs_Exon_cancer_vs_normal.txt	Gene expression values (non-log)
exon-Hs_Exon_cancer_vs_normal.txt	Probe set expression values (non-log)
commands-Hs_Exon_cancer_vs_normal.txt	Contains user commands for APT
Celfiles-Hs_Exon_cancer_vs_normal.txt	Relates sample to group
probe set-conversion-Hs_Exon_cancer_vs_normal.txt	Relates arbitrary gene IDs back to Ensembl

When the user selects the option “Calculate MiDAS p-values”, AltAnalyze first exports expression data for selected probe sets (e.g., AltAnalyze “core” annotated – Figure 2.5) to these files for all pair-wise comparisons. Once exported, AltAnalyze will communicate with the APT binary files packaged with AltAnalyze to run the analysis remotely. MiDAS will create a folder with the pair-wise comparison dataset name and a file with MiDAS p-values that will be automatically read by AltAnalyze and used for statistical filtering (stored in the AltAnalyze program directory under “AltResults/MiDAS”). These statistics will be clearly labeled in the results file for each probe set and used for filtering based on the user-defined p-value thresholds (Figure 2.5). When more than two-groups are being compared in AltAnalyze (“all groups” or “both options for “Type of group comparisons to perform”), the MiDAS p-value is reported for all groups designated by the user in combined array files or expression dataset file.

Note: Different versions of the APT MiDAS binary have been distributed. AltAnalyze is distributed with two versions that report slightly different p-values. The older version (1.4.0) tends to report larger p-values than the most recent distributed (1.10.1). Previous versions of AltAnalyze used version 1.4.0, while AltAnalyze version 1.1 uses MiDAS 1.10.1 that produces p-values that are typically equivalent to those as the AltAnalyze calculated splicing-index p-values (larger).

ASPIRE

For exon-exon junction microarray data (e.g., AltMouseA), the algorithm analysis of splicing by isoform reciprocity or ASPIRE was adapted from the original report (9) for inclusion into AltAnalyze. This algorithm uses the expression of probe sets aligning to two competitive exon-exon junctions, or one exon-exon junction and an exon aligning probe set along with gene expression values calculated as described with the splicing-

index method. These probe set relationships were derived using the Affymetrix exon or exon-exon junction names (e.g., E1-E3 and E2-E3 or E1-E3 and E2), obtained by the Affymetrix AltMerge transcript assembly program (10). For exon-exon junctions and exons aligning to the same gene, reciprocal probe set pairs were extracted using the `ExonAnnotate_module.py` program in AltAnalyze using the identifyPutativeSpliceEvents function. Such splicing events are classified as mutually-exclusive (mx-mx) or exon-inclusion/exon-exclusion (ei-ex). Mutually-exclusive splicing events represent an exchange of one exon for another (e.g., when E2-E4 and E1-E3 are compared, the E2 and E3 are considered regulated exons). For ei-ex, the proximal exon in the compared junctions is considered the regulated exon (e.g., when E1-E2 and E1-E3 are compared, E2 is the regulated exon).

Similar to the splicing-index method, for each reciprocal probe set, a ratio is calculated for expression of the probe set (non-log) divided by the mean of all gene expression reporting probe sets (non-log), for the baseline and experimental groups. The ASPIRE δI was then calculated for the inclusion (ratio1) and exclusion (ratio2) probe sets, as such:

R_{in} = baseline_ratio1/experimental_ratio1

R_{ex} = baseline_ratio2/experimental_ratio2

I_1=baseline_ratio1/(baseline_ratio1+baseline_ratio2)

I_2=experimental_ratio1/(experimental_ratio1+experimental_ratio2)

in_1= ((R_{ex}-1.0)*R_{in})/(Rex-Rin)

in_2= (R_{ex}-1.0)/(R_{ex}-R_{in})

δI = ((in_2-in_1)+(I_2-I_1))/2.0

If (R_{in}>1 and R_{ex}<1) or (R_{in}<1 and R_{ex}>1) and the absolute δI score is greater than the user supplied threshold (default is 0.2), then the δI is retained for the next step

in the analysis. If designated by the user, this next step will be a permutation analysis of the raw input data to determine the likelihood of each ASPIRE score occurring by chance alone. This permutation p-value is calculated by first storing all possible combinations of the two group comparisons. For example, if there are 4 samples (A-D) corresponding to the control group and 5 (E-H) samples in the experimental group, then all possible combinations of 4 and 5 samples would be stored (e.g, [B, C, G, H] and [A, D, E, F]). For each permutation set, ASPIRE scores were re-calculated and stored for all of these combinations. The permutation p-value is the number of times that the absolute value of a permutation ASPIRE score is greater than or equal to the absolute value of the original ASPIRE score (value = x) divided by the number of possible permutations that produced a valid ASPIRE score ((R_{in}>1 and R_{ex}<1) or (R_{in}<1 and R_{ex}>1)). If this p-value is less than user defined threshold, or x<2 (since some datasets have a small number of samples and thus little power for this analysis), the reciprocal probe sets are reported.

Linear Regression

When working with the same type of reciprocal probe set data as ASPIRE, a linear regression based approach can be used with similar results. This method is based on previously described approach (11). This algorithm uses the same input ASPIRE (junction comparisons, constitutive adjusted expression ratios). To derive the slope for each of the two biological conditions (control and experimental), the constitutive corrected expression of all samples for both reciprocal junctions is plotted against each other to calculate a slope for each of the two biological groups using the least squared method. In each case, the slope is forced through the origin of the graph (model = y ~ x – 1 as opposed to y ~ x). The final linear regression score is the $\log_2$ ratio of the slope of the experimental group divided by the baseline group. This ratio is analogous to a fold change, where 1 is equivalent to a 2-fold change. When establishing

cut-offs, select 2 to designate a minimum 2-fold change. The same permutation analysis used for ASPIRE is also available for this algorithm.

Note: For previous published analyses ((11) and Salomonis et al. in preparation), linear regression was implemented using the algorithm `rlm`, which is apart of the R `mass` package from bioconductor. The Python R interpreter `rpy`, was used to run these analyses (which requires installation of R). To use this option, select "linearregres-rlm" under "select the alternative exon algorithm" (Figure 2.5). If a related warning appears while running, you may need to load `AltAnalyze.py` and associated source code directly (requires installation of R version specific `rpy` – see Linux and Source Code AltAnalyze instructions).

External Alternative Exon Analysis File Import

In addition to providing several alternative exon analysis algorithms options in AltAnalyze (MiDAS, FIRMA, splicing-index), users can import results from other programs for downstream functional annotation analyses. These analyses consist of alternative exon annotation (alternative splicing and alternative promoter selection), protein isoform, protein domain and microRNA binding site disruption analysis and pathway over-representation. Two options for external probe set analysis are now available:

1) Annotation of 3rd party alternative exon data in AltAnalyze
2) Restricted analysis of probe sets using a pre-defined list

Both options are conceptually similar; provide AltAnalyze with a list of probe sets and optionally statistics, and AltAnalyze will return alternative exon statistics (option 2) and/or functional annotations for the input probe sets (option 1 and 2).

Option 1 – Import and annotation of 3rd party probe set results

For this option, no raw data is required (e.g., CEL files or expression values), just a list of probe sets of interest. This option is simply available by selecting the main analysis option “Annotate External Results” and selecting a tab-delimited probe set list. The probe sets supplied in this list can be produced by any alternative exon analysis program the user prefers, as long as the output is exon-level Affymetrix probe sets. Users can provide direct output files from JETTA or provide results in a more generic format, consisting of regulated probe sets and result statistics (optional).

The generic file format for alternative exon results import is:

1) (Column 1) Probe set ID ***(required)***
2) (Column 2) Alternative-exon fold ***(optional)***
3) (Column 3) Alternative-exon p-value ***(optional)***
4) (Column 4-100) Ignored data ***(optional)***

In addition to this generic format, results from the program JETTA can also be directly imported. Since this method produces two p-values for the MADS algorithm, the smallest of the two p-values is used. The results from this analysis are typical AltAnalyze result files, including input for DomainGraph and include any alternative exon statistics supplied with the probe set list.

Option 2 – Restricted probe set analysis (Advanced Feature)

This option can be performed with any of the main AltAnalyze analysis options (e.g., Process CEL Files, Expression Files or AltAnalyze Filtered). It is triggered by supplying a list of probe sets for each comparison of interest in the directory “filtering” in the AltAnalyze program directory (AltAnalyze_v1release/AltDatabase/filtering). The restricted filtering files are tab-delimited text files containing the corresponding exon probe set ID (Column 1) and optionally exon statistics and annotations (Columns 1-100). While the statistics and annotations provided in the restricted filtering file are not used for any computation, they are appended as extra columns in the alternative exon

results file. Multiple files, corresponding to specific biological comparisons (e.g., Tumor_vs_wild-type) can exist in this folder and only those files who's name matches the comparison name will be matched and used for restricted analysis. For example, if biological sample groups A, B and C are being compared, to filter C_vs_A, the restricted probe set file must be named C_vs_A.txt. Note: If the user enters any filters (e.g., maximum DABG p value threshold, minimum alternative exon score), these will further restrict which "significant" probe sets are reported. To report all probe sets found in the AltAnalyze database, make sure to set all thresholds to the minimum or maximum value (e.g., `--dabgp` 1 `--rawexp` 1 `--altp` 1 `--probetype` full `--altscore` 1 `--GEcutoff` 10000 – Section 2.3). Alternatively, users working with the command-line interface can use the option **`--returnAll`** to automatically enter this minimum values (see section 2.3).

Domain/miR-BS Over-Representation Analysis

A z-score is calculated to assess over-representation of specific protein features (e.g., domains) and miR-BS's found to overlap with probe sets that are alternatively regulated according to the AltAnalyze user analysis. This z-score is calculated by subtracting the expected number of genes in with a protein feature or miR-BS meeting the criterion (alternatively regulated with the user supplied thresholds) from the observed number of genes and dividing by the standard deviation of the observed number of genes. This z-score is a normal approximation to the hypergeometric distribution. This equation is expressed as:

$$z = \frac{(observed - expected)}{std.deviation(observed)} \qquad z = \frac{\left(r - n\frac{R}{N}\right)}{\sqrt{n\left(\frac{R}{N}\right)\left(1 - \left(\frac{R}{N}\right)\left(1 - \frac{n-1}{N-1}\right)\right.}}$$

n = All genes associated with a given element

r = Alternatively regulated genes associated with a given element

N = All genes examined

R = All alternatively regulated genes

Once z-scores have been calculated for all protein features and miR-BS linked to alternatively regulated probe sets, a permutation analysis is performed to determine the likelihood of observing these z-scores by chance. This is done by randomly selecting the same number of regulated probe sets from all probe sets examined and recalculating z-scores for all terms 2000 times. The likelihood of a z-score occurring by chance is calculated as the number of times a permutation z-score is greater than or equal to the original z-score divided by 2000. A Benjamini-Hochberg correction is used to transform this p-value to adjusted for multiple hypothesis testing.

Gene Ontology and Pathway Over-Representation Analysis

To perform advanced pathway and Gene Ontology (GO) over-representation AltAnalyze includes core python modules from the program GO-Elite (version 1.2) in AltAnalyze (http://www.genmapp.org/go_eite). GO-Elite performs Gene Ontology (GO) and WikiPathway over-representation using the same algorithms listed for domain/miR-BS over-representation analysis (ORA) (e.g., z-score, permutation p-value, BH p-value calculation). After ORA, all GO terms and pathways are filtered using user-defined statistical cut-offs (z-score, permutation p-value and number of genes changed), with GO terms further pruned to identify a minimally redundant set of reported GO terms, based on hierarchical relationships and ORA scores (see GO-Elite documentation). The gene database used for GO-Elite is specific to the version of Ensembl in the downloaded AltAnalyze gene database (Plus versions only). All result files normally produced by GO-Elite will be produced through AltAnalyze.

AltAnalyze generates two types of gene lists for automated analysis in GO-Elite; differentially expressed genes and alternatively regulated genes. Criterion for

differentially expressed genes are defined by the user in the GO-Elite parameters window (e.g., fold difference > 2 and t-test p < 0.05). The user can choose to use the rawp (one-way ANOVA, two groups) or adjp (BH adjusted value of the rawp) in the software. All genes associated with alternative exons are used for GO-Elite analysis. Genes with alternative exons that also have alternative splicing annotations can be further selected using the “Filter results for predicted AS” option. Results are exported to the “GO-Elite” directory in the user-defined output directory, while input gene lists can be found in the folders “GO-Elite/input” and “GO-Elite/denominator”. The gene lists for differentially expressed genes have the prefix “GE.” while the alternative exon files have the prefix “AS.”. The appropriate denominator gene files for each is selected by GO-Elite. Species and array specific databases for GO-Elite analysis are downloaded automatically from AltAnalyze when the species database is installed. Array types supported for each species include any Affymetrix supported at the time the Ensembl database was released along with any arrays supported by Ensembl.

If a particular array is not supported by AltAnalyze but has a valid Affymetrix CSV annotation file, AltAnalyze will build the necessary GO-Elite relational databases to perform both GO and WikiPathway ORA. Probe set, gene and GO relationships will be extracted automatically from the Affymetrix annotation CSV file and gene to WikiPathway relationships from the http://www.wikipathways.org (if relationships are present).

3.3 Probe set Filtering

Prior to alternative exon analysis, AltAnalyze can be used to remove probe sets that are not deemed as sufficiently expressed. For the two conditions that AltAnalyze compares (e.g., cancer versus normal), a probe set will be removed if neither condition has a

mean detection above background (DABG) p value less than the user threshold (e.g., 0.05). Likewise, if neither condition has a mean probe set intensity greater than the user threshold (e.g., 70), then the probe set will be excluded from the analysis. When comparing two conditions (pairwise comparison) for probe sets used to determine gene transcription (e.g., constitutive aligning), both conditions will be required to meet these expression thresholds in order to ensure that the genes are expressed in both conditions and thus reliable for detecting alternative exons as opposed to changes in transcription. When comparing all biological groups in the user dataset, however, these additional filters are not used.

3.4 Constitutive and gene expression calculation

Identifying probe sets for gene expression calculations

A reliable estimate of transcriptional activity for each gene is required to both report basic gene expression statistics and perform alternative exon analysis. Exon arrays probe most well annotated exon regions, many introns, untranslated regions and theoretically transcribed regions. Gene expression values can be calculated in one of two ways from AltAnalyze; (A) using constitutive probe sets, (B) using core probe sets. The constitutive probe sets are recommended for use for all analyses.

Constitutive probe sets in AltAnalyze are classified as probe sets that align to a pre-determined set of exon regions that are most common to all mRNA transcripts used when the AltAnalyze database is created. RNA transcript exon structures and genomic positions are extracted from analogous versions of Ensembl and the UCSC genome browser database. While AltAnalyze database versions EnsMart49 and EnsMart52 used mRNA annotations from the Affymetrix probe set annotation file, EnsMart54 exclusively derives constitutive probe set annotation from UCSC and Ensembl exon structures (Figure 3.1).

In addition to constitutive probe sets, users have the option to use all probe sets that have an Affymetrix core probe set annotation or align to an AltAnalyze defined exon region. Affymetrix defines probe sets according to three criterion, "core", "extended" and "full". The Affymetrix core annotations are recommended for calculation of constitutive levels by Affymetrix, however, these are often a mix of known alternative exons and AltAnalyze annotated constitutive exons. In addition to these Affymetrix core exons, the AltAnalyze core set consists of any probe set that aligns to an exon region, but excludes probe sets aligning to any non-exon features, Including introns and untranslated regions (Figure 3.1).

Calculating gene expression values

Gene expression values are calculated twice during an exon array analysis; (1) To report gene expression fold changes and summary statistics, (2) To calculate gene expression normalized intensities for alternative exons. While both analyses use the same set of starting probe sets (constitutive or core), they differ in how filtering of these probe sets occurs. These differences are important when trying to eliminate false positive alternative exon predictions.

For gene expression summary reporting, expression values from all constitutive or core probe sets (user defined) are filtered and then averaged. For this analysis, the constitutive or core probe sets are first identified from the AltAnalyze database (Ensembl_probeset file) and corresponding expression and detection probabilities extracted from the APT RMA result files from the input array files. For each biological group (typically containing replicates), the mean expression and mean detection above background probabilities are calculated for each probe set. If the mean expression and the mean detection above background probabilities for at least one group does not meet the user defined thresholds (by default $p<0.05$ and expression>70), then this probe set will not be used to calculate the gene expression

value for the corresponding gene. The average expression for all remaining probe sets, for each array will calculated to obtain a mean gene expression value for each probe set for each replicate. Thus, any differences between probe sets used to calculate gene expression will be averaged. For example, when all core probe sets are used, the expression of any alternative exons will be averaged with the non-alternative exons. If no constitutive or core probe sets for a gene meet these criterion (see below), then all will be used to calculate gene expression. Only one set of gene expression values is reported for each Ensembl gene.

This same strategy is used for calculating a gene expression value for the alternative exon analysis with the following differences: (1) a probe set used to calculate gene expression must demonstrate evidence expression (expression and detection probability filtering) in both conditions (pairwise comparisons only) or it is removed and (2) if no probe sets remain after expression and detection probability filtering, then no probe sets for the gene will NOT be analyzed for alternative splicing. If all groups are being compared for the alternative exon analysis, rather than direct comparison of two groups, then only one biological group must meet the expression and detection probability thresholds for the gene expression (same as for gene expression summary reporting, above). These additional requirements ensure that the gene is expressed at a sufficient levels to assess alternative splicing in both compared biological groups. This is important in eliminating false positive changes in probe set expression that can occur when the gene is expressed in only one condition.

3.5 Alternative Splicing Prediction

To predict whether or not a single probe set (EA) or reciprocal probe set-pair (JA) associates with an alternative splicing or alternative promoter sequence, AltAnalyze

uses two strategies; 1) Identify alternative exons/introns based on *de novo* isoform comparison and 2) Incorporating splicing predictions from UCSC's "known_alt.txt" file.

De Novo Splicing Prediction and Exon Annotation

In order to identify exons with alternative splicing or promoters, AltAnalyze compares all available gene transcripts from UCSC and Ensembl to look for shared and different exons. To achieve this, all mRNA transcripts from UCSC's species-specific "mRNA.txt" file that have genomic coordinates aligning to a single Ensembl gene and all Ensembl transcripts from each Ensembl build are extracted. Only UCSC transcripts that have a distinct exon composition from Ensembl transcripts are used in this analysis, excluding those that have a distinct genomic start or stop position for the first and last exon respectively (typically differing 5' and 3' UTR agreement), but identical exon-structure.

When assessing alternative splicing, cases of intron retention are identified first. These regions consist of a single exon that spans an two adjacent exons at least one another transcript for that same gene. These retained introns are stored for later analysis, but eliminated as annotated exons. Remaining exons are clustered based on whether their genomic positions overlap (e.g., alternate 5' or 3' start sites). Each exon cluster is considered an exon block with one or more regions, where each block and region is assigned a numerical ID based on genomic rank (e.g., E1.1, E1.2, E2.1, E3.1). For each exon in a transcript, the exon is annotated as corresponding to an exon block and region number (Figure 3.1). All possible pair-wise transcript comparisons for each gene are then performed to identify exon pairs that show evidence of alternative exon-cassettes, alternative 3' or 5' splice sites or alternative-N or -C terminal exons (Figure 3.1). All transcript exon pairs are considered except for those adjacent to a retained intron. This analysis is performed by comparing the exon block ID and region IDs of an exon and it's neighboring exons to the exon blocks and

regions in the compared transcript. Ultimately, a custom heuristic assigns the appropriate annotation based on these transcript comparisons.

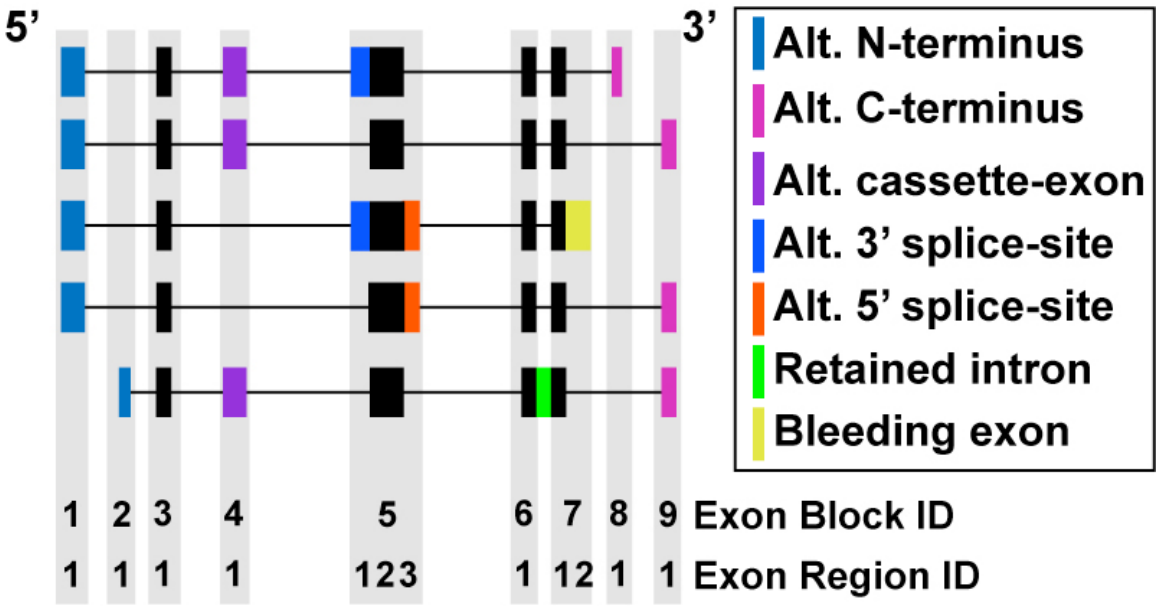


Figure 3.1. Comparison of mRNA Exon Composition. To determine alternative splicing and alternative promoter regulation, all analyzed transcripts (Ensembl and UCSC) were compared based on exon genomic positions and subsequently annotated by a custom heuristic. Exon block and regions definitions are shown. The different types of alternative exon events are illustrated by different colored exons among five theoretical transcripts for the same gene. The black filled boxes represent exon regions that are most common to all mRNA transcripts. These regions are annotated as constitutive by AltAnalyze. Probe sets that overlap with the constitutive regions can be used to calculate gene expression. All non-intron regions that align to an exon block can also optionally be used to calculate gene expression.

Incorporating UCSC Splicing Predictions

In addition to all *de novo* splicing annotations, additional splicing annotations are imported from the UCSC genome database and linked to existing exon blocks and regions based on genomic coordinate overlap. This comparison is performed by the `alignToKnownAlt.py` module of AltAnalyze (called from `EnsemlbImport.py`). *De novo* and UCSC splicing annotations are stored along with probe set Ensembl gene alignment data in the file <species>_Ensembl_probesets.txt. These annotations are used by AltAnalyze and DomainGraph. In the AltAnalyze result file, UCSC KnownAlt the major splicing annotation types are altFivePrime, altThreePrime, cassetteExon, altPromoter, bleedingExon and retainedIntron. In comparison, the major AltAnalyze determined splicing annotation types alt-5', alt-3', cassette-exon, alt-N-term, intron-retention, exon-region-exclusion and alt-C-term. These splicing annotations are determined by comparing relative exon-junction positions for all analyzed transcripts for each gene (see proceeding section). The annotation exon-region-exclusion is the opposite of intron-retention (region most commonly described as exon rather than intron), alt-N-term is similar to altPromoter (distinct 5' distal-transcript exon with shared 3' exons) and alt-C-term is the opposite of alt-N-term. To better understand these annotations, look at specific probe set examples through the NetAffx website (www.affymetrix.com/analysis/index.affx) using the UCSC browser option.

Filtering for Alternative Splicing

As mentioned in previous sections, AltAnalyze includes the option restrict alternative exon results to only those probe sets or reciprocal probe set-pairs predicted to indicate alternative splicing. AltAnalyze considers alternative splicing as any alternative exon annotation other than an alternative N-terminal exon or alternative promoter annotation derived from *de novo* or UCSC genome database annotations.

3.6 Protein/RNA Inference Analysis

Identifying Alternative Proteins Protein Domains

Probe set sequences present on exon or junction arrays are used to identify which proteins align to or are missing from transcripts for that gene. To do this, all Ensembl and UCSC mRNA transcripts are extracted for a gene that corresponds to a given probe set. For each transcript, all exon genomic coordinates are stored. For exon arrays, transcripts with exons that contain the probe set genomic coordinates are considered transcript aligning, while all others are considered non-aligning. For junction arrays, if two reciprocal junction probe sets align to distinct isoforms, these relationships are stored rather than the aligning and non-aligning isoforms, however, if both isoforms do not align to distinct transcripts, then the one with a aligning and non-aligning set of transcripts is stored for further exon comparisons.

If a set of aligning and non-aligning isoforms (competitive) is identified for a probe set or reciprocal junction pair, all possible aligning and non-aligning pair-wise combinations are identified to find those pair-wise comparisons with the smallest difference in exon composition. This is accomplished by determining the number of different and common exons each transcript pair contains (based on genomic start and stop of the exon). When comparing the different transcript pairs, the most optimal pair is selected by first considering the combined number of distinct exons in both transcripts and second the number of common transcripts. Thus, if one transcript pair has 4 exons in common and 2 exons not in common, while a second pair has 5 exons in common and 3 exon not in common, the prior will be selected as the optimal since it contains less overall differences in exon composition (even though it has less common exons than the other pair). A theoretical example is illustrated in Figure 3.2.

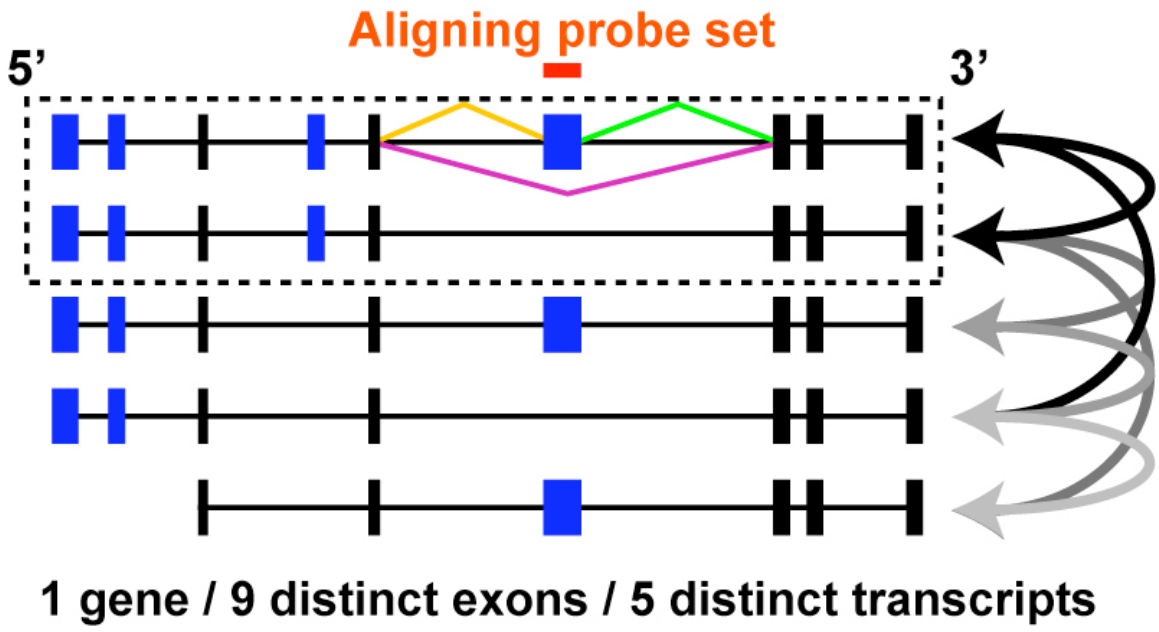


Figure 3.2. Comparison of probe set aligning and non-aligning mRNAs. A theoretical gene is shown with 9 distinct exons and 5 distinct mRNA transcripts with different exon combinations. Four exons are alternatively expressed in different transcripts (blue exons) while five are common or constitutive to all transcripts (black exons). All possible pair-wise transcript combinations are shown (arrows) between mRNAs that contain and probe set aligning exon and those that do not. Ultimately, a single pair is selected that has the most common exons and the least uncommon exons (dashed box). The same strategy is used for junction arrays as exon, however, for junction arrays, aligning transcripts are identified by direct probe set (orange, green, purple) sequence alignment to the mRNAs prior to comparing exon composition as described above.

Once a single optimal isoform pair has been identified, protein sequence is obtained for each by identifying protein IDs that correspond to the mRNA (Ensembl or NCBI) and if not available, a predicted protein sequence is derived based on *in silico*

translation. Although such a protein sequence may not be valid, given that translation of the protein may not occur, these sequences provide AltAnalyze the basis for identifying conservative changes predictions for a change in protein size, sequence and domain composition. Domain/protein features are obtained directly from UniProt's sequence annotation features or from Ensembl's InterPro sequence annotations (alignment e-value <1) (see section 5 data extraction protocols). Any InterPro sequences with a description field or any UniProt sequence annotation feature that is not of the type "CHAIN", "CONFLICT", "VARIANT", "VARSPLIC" and "VAR_SEQ" are examined by AltAnalyze. To compare domain or motif sequence composition differences, the protein sequence that corresponds to the amino-acid start and stop positions of a domain for each transcript is searched for in each of the compared protein isoforms. If the length of a motif sequence is less than 6 amino-acids, flanking sequence is included. If a domain is present in one but not another isoform, that domain is stored as differentially present. To identify differences in protein sequence (e.g., alternative-N-terminus, C-terminus, truncation coding sequence and protein length), the two protein sequences are directly compared for shared sequence in the first and last five residues and comparison of the entire sequences. If the N-terminal sequence is common to both isoforms but there is a reduction in more than 50% of the sequence length, the comparison is annotated as truncated. All of these annotations are stored for each probe set for import into AltAnalyze, with each new database build.

Direct Domain/Motif Genomic Alignment

The above strategy allows AltAnalyze to identify predicted protein domains and motifs that are found in one isoform but not the other (aligning to a probe set and not aligning). In addition to these "inferred" domain predictions, that include protein domains/motifs that do not necessarily overlap with the regulated probe set, AltAnalyze includes a distinct set of annotations that only corresponds to domain/motif protein

sequence that directly overlaps with a probe set. Alignment of probe sets to InterPro IDs is achieved by comparing probe set genomic coordinates to InterPro genomic coordinates. To obtain InterPro genomic start and end positions are determined by first identifying the relative amino-acid positions of an InterPro region in an Ensembl protein, finding which exon and at what position the InterPro region begins and ends and finally stroing the genomic position of these relative exon coordinates.

These probe set-InterPro overlaps can be of two types; 1) probe sets whose sequence is present in the domain coding RNA sequence and 2) probe sets whose sequence is not present in the domain coding RNA sequence. The second type of overlap typically occurs in the gene introns. While these associations are typically meaningful, false “indirect” associations are possible. To reduce the occurrences of these false positives, any probe set that aligns to the UTR of an Ensembl gene or that occurs in the first or last exon of an mRNA transcript are excluded. These heuristics were chosen after looking at specific examples that the authors considered to be potential false positives. For junction arrays, the alternative exon genomic position rather probe set was used.

Identifying microRNA Binding Sites associated with Alternative Exons

MicroRNA binding site (miR-BS) sequence is obtained and compared from four different microRNA databases (see section 6.5), and compared to identify miR-BSs in common to or distinct to different databases. Probe set sequences or exon sequences (aligning to two reciprocal junctions) are obtained from Affymetrix (see section 6.7). Each miR-BS sequence is searched for within probe set or exon sequence to identify a match. These relationships are stored with each new database build and are used by AltAnalyze and DomainGraph.

Exhaustive Protein Domain/Motif Analysis

Very similar to the competitive protein domain/motif analysis is the exhaustive protein comparison analysis. This feature is currently not available by default in AltAnalyze and requires replacement database files from AltAnalyze support. The purpose of these files is to obtain the most conservative possible domain-level prediction results from the competitive analysis. This done by storing all pair-wise aligning and non-aligning isoform comparisons (Figure 3.2) and then obtaining protein sequence for each transcript, as described in earlier sections and storing all domain/motifs differentially found between all possible competitive isoforms. From these stored results, all possible competitive isoforms are themselves compared to find isoforms that ideally show only differences in central regions of the protein (no N-terminal or C-terminal differences), next for those that contain as few possible domain-level predictions and finally for those with the smallest overall differences in protein length. For detailed algorithm information see the `IdentifyAltIsoforms` module and the function “compareProteinFeaturesForPairwiseComps”.

Section 4 – Using R with AltAnalyze

While AltAnalyze is a largely a stand-alone program, some statistical analyses can be included that depend on external applications. These require prior installation of these tools using operating system installers and properly interfacing them with AltAnalyze.

4.1 Configuring R

Although installation of R is not required for any of the AltAnalyze analyses, for users who wish to use more advanced statistics, it will be necessary. Currently, the only statistic that requires installation of R is the linear regression method `rlm`. `rlm` is a regression statistical method apart of the R package `mss`. This method is preferred by some over the alternative linear regression method provided by default in AltAnalyze. Both methods produce very similar statistics, with only a few probe sets differing between threshold parameters out of hundreds of results. However, since the `rlm` method was used in the published linear regression analyses, other users wishing to replicate these results, may wish to use this algorithm.

To run the option “linearegress-rlm” from the “Alternative Exon Analysis Parameters” window, you will need to install a compatible version R (only version 2.1 has been extensively tested). Along with R, the user will need to install the R statistical package `mss`. R is interpreted by Python using the Python program Rpy, which is packaged with the compiled versions of AltAnalyze, but not the source code (http://rpy.sourceforge.net/download.html). With some compiled installations of AltAnalyze, you will receive a warning when running “linearegress-rlm”. This error occurs due to an Rpy compiling error that is specific to the OS executable version of

AltAnalyze, but should work fine when Rpy is installed for the associated version of R by the user (each version of Rpy corresponds to several versions of R).

Whether dealing with a compiled or source version of AltAnalyze, if Python reports that it cannot find the current version of R, the user may need to update the computers Environment Variables setting Path (Windows only). This is accessed through by opening Control Panels>System Properties>Advanced>Environment Variables and selecting the Variable “Path” and entering the path location of R (for example, “;C:\Program Files\R\rw2010;” – **No spaces before or after ;**). Contact AltAnalyze support if problems persist.

Section 5 - Software Infrastructure

5.1 Overview

AltAnalyze consists of more than 30 modules and over 10,000 lines of code. The core modules for AltAnalyze consist of the programs ExpressionBuilder and AltAnalyze, which can be used in tandem or separately through the AltAnalyze GUI. The user will never likely need to deal directly with these modules names when running AltAnalyze, but these distinct core modules are used for different analysis functions (Figure 5.1).

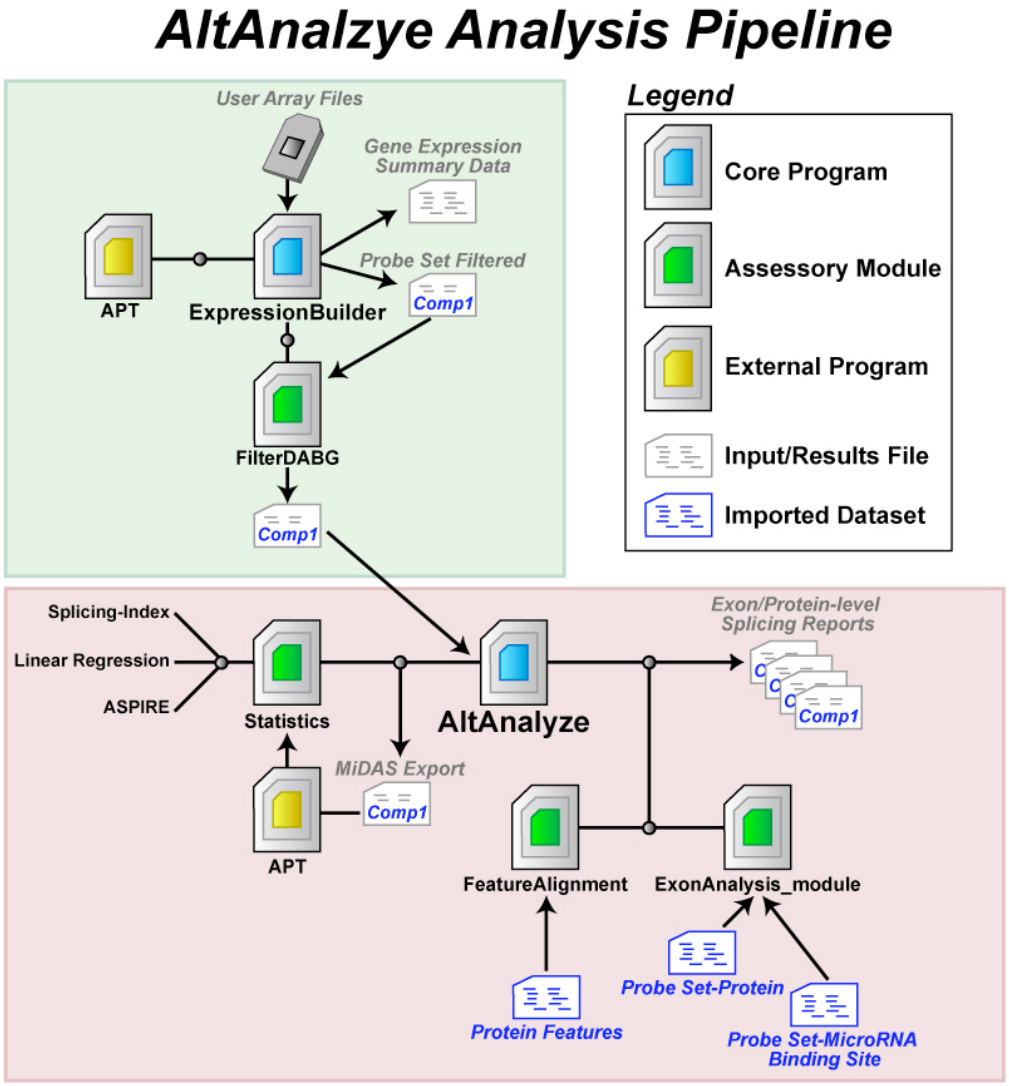


Figure 5.1. AltAnalyze Analysis Pipeline. The AltAnalyze workflow is depicted. The transparent green box highlights functions performed by the ExpressionBuilder module of AltAnalyze whereas the transparent red box highlights the AltAnalyze module. (A) User microarray data (probe set expression values and DABG p-values) or CEL files are imported into AltAnalyze via the ExpressionBuilder module, which separates data for different biological array groups into user designated pair-wise comparisons (e.g., cancer vs. normal). For each pair-wise comparison, probe set expression values and DABG p-values are exported to separate files, and then analyzed by the module FilterDABG to exclude probe sets with poor detection parameters. The resulting files are inputs for alternative exon analysis. In parallel, a gene expression summary file is produced with Ensembl gene level expression values (based on constitutive probe set expression) for each gene and array along with summary statistics (average expression, fold, and t-test p-value for all pair-wise comparisons) and annotations. (B) Using the ExpressionBuilder pair-wise comparison files, AltAnalyze re-calculates constitutive expression values, evaluates changes in probe set expression relative to constitutive (statistics module), and links probe sets with “significant” changes to aligning alternative protein sequence and predicted changes in protein and miR-BS architecture (ExonAnalyze and FeatureAlignment modules). The result is a series of probe set and gene summary files along with over-representation statistics for the regulation of protein and miR-BS features. Optionally, probe set and constitutive expression values can be exported to the bundled application Affymetrix Power Tools to calculate additional alternative exon statistics to be included in the AltAnalyze analysis.

The ExpressionBuilder component builds constitutive gene expression summary files as well as filters the probe set expression data prior to alternative-exon analysis. The AltAnalyze module performs all of the alternative-exon analysis and MiDAS p-value calculations.

5.2 ExpressionBuilder Module

The ExpressionBuilder program is principally designed to perform the following tasks:

1) Import user expression data from tab-delimited files.
2) (Splicing Arrays Only) Exclude probe sets where no samples have a DABG p<user-threshold (only applicable when DABG p-values exist).
3) Organize your data according to biological groups and comparisons (specified by the user from custom text files).
4) (Exon Arrays Only) Calculate gene transcription levels for all Ensembl genes.
5) Export calculated gene expression values along with folds, t-test and f-test p-values and gene annotations for all genes and all user indicated comparisons.
6) (Splicing Arrays Only) Export all gene linked probe set expression and DABG data for all pairwise comparisons for further filtering (next step).
7) (Splicing Arrays Only) Filter the resulting probe set data using mean expression values and probabilities specific for each pairwise comparison and user-defined thresholds (Figure 2.4).

The above tasks are performed in order by the ExpressionBuilder module. Detection probabilities are assessed at two steps (2 and 7). In step 2, import of DABG p-values are for the purpose of calculating a transcription intensity value only for those constitutive probe sets (present in all or most transcripts) that show detection above background, since some probe sets will have weaker expression/hybridization profiles as others. If no probe sets have a DABG p-value less than the default or user supplied

threshold (for at least one sample in your dataset), all selected probe sets will be used to calculate expression (constitutive aligning only if default is selected).

In step 7, the probe set DABG p-values are examined to include or exclude probe sets for alternative splicing analysis. This step is important in minimizing false positive splicing calls. False positive splicing calls can occur when a probe set is expressed below detectable limits and results in a transcription-corrected expression value that artificially appears to be alternatively regulated. For the expression and DABG p-value files output from APT, probe set expression values are initially filtered to remove any probe sets where the expression and dabg p-values are below the user defined threshold for all biological groups examined and export all pairwise comparison group files (expression and dabg) for further filtering. For probe sets used to determine gene expression levels (constitutive or all core), the module FilterDABG is used to remove probe sets that are expressed below user defined thresholds (expression and dabg) in the two comparison groups for all pairwise comparisons. If a probe set is not used to determine gene expression levels, then for at least one of the two biological groups, the same criterion must be true (mean DABG p-value and expression). These filters ensure that: 1) the gene is “expressed” in both conditions and 2) the probe set is “expressed” in at least one condition. The probe sets and expression values passing these user defined filters are exported to a new file that is ready to use for splicing analyses, stored to the user output directory under “AltExpression/ExonArray/*species*”. This file can also be directly selected in future AltAnalyze runs as input for analysis (“AltAnalyze filtered file” – Figure 2.2).

Runtime of ExpressionBuilder is dependent on the number of conditions and array type being analyzed (10 minutes plus for Affymetrix Exon 1.0 ST arrays). If multiple comparisons are present in a single expression file, input files for AltAnalyze will all be generated at once and thus runtimes will take longer.

***Note 1**: While this pipeline is mainly for use with exon or junction arrays, it is also compatible with a standard 3' Affymetrix microarray dataset to calculate folds, t-test p-values and assign annotations to this data. This is useful when you have many comparisons in your dataset and you don't wish to manually calculate these values.*

***Note 2**: You do not need to run ExpressionBuilder if you have an alternative way of building AltAnalyze input files. To do so, your file headers for each array must have the name "group:sample_name", where your group names are different for each group and the denominator group is listed first and the numerator is listed second. Below the header line should only be probe set IDs and log2 expression values.*

5.3 AltAnalyze Module

The AltAnalyze module is the primary software used for all alternative exon analyses. This software imports the filtered expression data and performs all downstream statistical and functional analyses. This program will analyze any number of input comparison files that are in the "AltExpression" results directory for that array type. The main analysis steps in this program are:

1) Import exon or junction annotations, to determine which probe sets to analyze, which are predicted constitutive probe sets and which correspond to known AS or APS events.
2) Import the user expression data for the pair-wise comparison.
3) Store probe set level data for all probe sets corresponding to either a constitutive exon or for splicing event, while storing the group membership for each value.
4) Calculate a constitutive expression value for each gene and each sample (used for the splicing score later on). OPTIONAL: If the user selected a cut-off for constitutive fold changes allowed to look at alternative exon regulation, then

remove genes from the analysis that have a gene-expression difference between the two groups > cut-off (up or down).

5) ***(Exon array only)*** OPTIONAL: exports input for the Affymetrix Power Tools (APT) program to calculate a MiDAS p-value for each probe set. If using this option.
6) Calculate a splicing score and t-test p-value from the probe set and constitutive expression values. This calculation requires that splicing ratios are calculate for each sample (exon/constitutive expression) and then compared between groups. For exon arrays, the splicing index method (SI) is calculated for each probe set. For junction arrays, ASPIRE, Linear Regression are used with the pre-determined reciprocal junctions or alternatively are calculated for individual probe sets using the SI method.
7) ***(Junction array only)*** OPTIONAL: performs a permutation analysis of the sample ASPIRE input values or Linear Regression values to calculate a likelihood p-value for all possible sample combinations.
8) Retain only probe sets meeting the scoring thresholds for these statistics (splicing score, splicing t-test p, permutation p, MiDAS p – see Section 3.2).
9) Import probe set-protein, probe set-domain and probe set-miRNA associations pre-built local from flat files (see Section 6).
10) For the remaining probe sets, import all protein domain and miR-BS to all pre-built probe set associations (see Section 3.3 for details). Import all probe set-domain and –miR-BS associations for all genes to calculate an over-representation z-score for all domains and miR-BS's along with a non-adjusted and adjusted p-value. Export the resulting statistics and annotations to tab-delimited files in the "AltResutls/AlternativeOutput" folder in the user-defined output directory.

11) ***(Junction array only)*** Import splicing and exon annotations for regulated exons corresponding to each set of reciprocal probe sets (e.g. for E1-E3 compared to E1-E2, E2 is the regulated exon). These annotation files are the same as those for exon arrays, except that the probe set is replaced by the exon predicted to be regulated by the reciprocal probe set-pair (see Section 6.2).
12) For protein domain and miR-BS annotations, reformat the direction/inclusion status of the annotation. For example, if a kinase domain is only found in a protein that aligns to a robe set, but was down-regulated, then the annotation is listed as (-) kinase domain, but if up-regulated is listed as (+) kinase domain.
13) Export the results from this analysis to the "AltResutls/AlternativeOutput" folder in the user-defined output directory.
14) Summarize the probe set or reciprocal junction data at the level of genes and export these results (along with Gene Ontology/Pathway annotations).
15) Export overall statistics from this run (e.g. number of genes regulated, splicing events).
16) ***(Junction array only)*** Combine and export the exported probe set and gene files for each comparison analyzed, to compare and contrast differences.

Result File Types

When finished AltAnalyze will have generated five files.

1) name-scoringmethod-exon-inclusion-GENE-results.txt
2) name-scoringmethod-exon-inclusion-results.txt
3) name-scoringmethod-ft-domain-zscores.txt
4) name-scoringmethod-miRNA-zscores.txt
5) name-scoringmethod-DomainGraph.txt

Here, “name” indicates the comparison file name from ExpressionBuilder, composed of the species + array_type + comparison_name (e.g. Hs_Exon_H9-CS-d40_vs_H9-ESC-d0), “scoringmethod” is the type of algorithm used (e.g. SI) and the suffix indicates the type of file.

The annotation files used by AltAnalyze are pre-built using other modules with this application (see Section 6). Although the user should not need to re-build these files on their own, advanced users may wish to modify these tables manually or with programs provided (see Section 6.7).

For protein-level functional annotations (e.g., domain changes), this software assumes that if an exon is up-regulated in a certain condition, that the protein domain is also up-regulated and indicates it as such. For example, for exon array data, if a probe set is up-regulated (relative to gene constitutive expression) in an experimental group and this domain is found in the protein aligning to this probe set, in the results file this will be annotated as (+) domain. If the probe set were down-regulated (and aligns as indicated), this would be annotated as (-) domain.

Section 6 – Building AltAnalyze Annotation Files

6.1 Splicing Annotations and Protein Associations

A number of annotation files are built prior to running AltAnalyze that are necessary for:

1) Organizing exons and introns from discrete transcripts into consistently ordered sequence blocks (`UCSCImport.py` and `EnsemblImport.py`).
2) Identifying which exons and introns align to alternative annotations (`alignToKnownAlt.py` and `EnsemblImport.py`).
3) Identifying probe sets with likely constitutive annotations (`ExonArrayAffyRules.py`).
4) Identifying which probe sets align to which exons and introns (`ExonArrayEnsemblRules.py`).
5) Extracting out protein sequences with functional annotations (`ExtractUniProtFunctAnnot.py` and `EnsemblSQL.py`).
6) Identifying probe sets that overlap with microRNA binding sites (`MatchMiRTargetPredictions.py` and `ExonSeqModule.py`).
7) Matching probe set genomic coordinates to cDNA exon coordinates and identify the optimal matching and non-matching mRNA/protein for each probe set (`IdentifyAltIsoforms.py and ExonAnalyze_module.py`)

These annotation files are necessary for all exon and junction array analyses. Junction array analyses further require:

8) Matching reciprocal junction probe sets to annotated exons or introns (`JunctionArray.py`, `EnsemblImport.py` and `JunctionArrayEnsemblRules.py`), creating a file analogous to (4) above.

9) Matching reciprocal junction probe set sequence to microRNA binding sites (`JunctionSeqSearch.py`), creating a file analogous to (6) above.
10) Matching probe set sequence to cDNA sequences (`mRNASeqAlign.py`), prior to identification of optimal matching and non-matching mRNA/proteins (`IdentifyAltIsoforms.py`).

With the creation/update of these files, the user is ready perform alternative exon analyses for the selected species and array type. Since many of these analyses utilize genomic coordinate alignment as opposed to direct sequence comparison, it is import to ensure that all files were derived from the same genomic assembly.

Note: Although all necessary files are available with the AltAnalyze program at installation and some files can be updated automatically from the AltAnalyze server, users can use these programs to adjust the content of these files, use the output for alternative analyses, or create custom databases for currently unsupported species.

6.2 Building Ensembl-Probe Set Associations

Exon and Gene Arrays

Affymetrix Exon 1.0 ST arrays are provided with probe set sequence, transcript cluster and probe set genomic location from Affymetrix. Each of these annotations is used by AltAnalyze to provide gene, transcript and exon associations. Although transcript clusters represent putative genes, the AltAnalyze pipeline derives new gene associations to Ensembl genes, so that each probe set aligns to a single gene from a single gene database. This annotation schema further allows AltAnalyze to determine

which probe sets align to defined exons regions (with external exon annotations), introns, and untranslated regions (UTR).

To begin this process, Ensembl exons (each with a unique ID) and their genomic location and transcript associations are downloaded for the most recent genomic assembly using the AltAnalyze `EnsemblSQL.py` module, which parses various files on the Ensembl FTP SQL database server to assemble the required fields. This file is saved to the directory "AltDatabase/ensembl/*species*/" with the filename "*species*_Ensembl_transcript-annotations.txt". Since Ensembl transcript associations are typically conservative, transcript associations are further augmented with exon-transcript structure data from the UCSC genome database, from the file "all_mrna.txt" (Downloads/*species*/Annotation database/all_mrna.txt.gz). This file encodes genomic coordinates for exons in each transcript similar to Ensembl. Transcript genomic coordinates and genomic strand data from UCSC is matched to Ensembl gene coordinates to identify genes that specifically overlap with Ensembl genes with the Python program `UCSCImport.py`. Unique transcripts, with distinct exon structures from Ensembl, are exported to the folder "AltDatabase/ucsc/*species* "*species*_UCSC_transcript_structure_filtered_mrna.txt", with the same structure as the Ensembl_transcript-annotations file.

Once both transcript-structure files have been saved to the appropriate directory, `ExonArrayAffyRules.py` calls the program `EnsemblImport.py` to perform the following steps:

1) Imports these two files, stores exon-transcript associations identify exon regions to exclude from further annotations. These are exons that signify intron-retention (overlapping with two adjacent spliced exons) and thus are excluded as valid exon IDs. These regions are also flagged as intron-retention regions for later probe set annotations.

2) Assembles exons from all transcripts for a gene into discrete exon clusters. If an exon cluster contains multiple exons with distinct boundaries, the exon cluster is divided into regions that represent putative alternative splice sites (Text S1 Figure 1). These splice sites are explicitly annotated downstream. Each exon cluster is ordered and number from the first to the last exon cluster (e.g., E1, E2, E3, E4, E5), composed of one or more regions. These exon cluster/region coordinates and annotations are stored in memory for downstream probe set alignment in the module `ExonArrayAffyRules.py`.
3) Identifies alternative splicing events (cassette-exon inclusion, alternative 3' or 5' splice sites, alternative N-terminal and C-terminal exons, and combinations there in) for all Ensembl and UCSC transcripts by comparing exon cluster and region numbers for all pairs of exons in each transcript (see proceeding sections for more information). Alternative exons/exon-regions and corresponding exon-junctions are stored in memory for later probe set annotation and exported to summary files for creation of databases for the Cytoscape exon structure viewer, SubgeneViewer (currently in development).
4) (New to AltAnalyze 1.14) Constitutive exon regions are defined by counting the number of unique Ensembl and UCSC mRNA transcripts (based on structure) associated with each unique exon region. If multiple exon regions have the same number of transcript associations, then these are grouped. The grouped exon regions that contained the most transcripts for the gene are defined as constitutive exon regions. If only one exon region is considered constitutive then the grouped exon regions with the second highest ranking (based on number of associated transcripts) are included as constitutive (when there at least 3 ranks) (Figure 3.1).

Upon completion, `ExonArrayAffyRules.py`:

1) Imports Affymetrix Exon 1.0 ST probe sets genomic locations and transcript cluster annotations from the Affymetrix probe set.csv annotation file (e.g., HuEx-1_0-st-v2.na23.hg18.probeset.csv). Note: prior to AltAnalzye 1.14, mRNA alignment count numbers were also downloaded from this file to deduce constitutive probe sets. Although transcript clusters will be disregarded at the end of the analysis, these are used initially to group probe sets.
2) Transcript cluster genomic locations are matched to Ensembl genes genomic locations (gene start and stop) to identify single transcript clusters that align to only one Ensembl gene for the respective genomic strand. For transcript clusters aligning to more than one Ensembl gene, coordinates for each individual probe set are matched to aligning Ensembl genes, to identify unique matches. If multiple transcript clusters align to a single Ensembl gene, only probe sets with an Affymetrix annotated annotation corresponding to that Ensembl gene, from the probeset.csv file, are stored as proper relationships. This ensures that if other genes, not annotated by Ensembl exist in the same genomic interval, that they will not be inaccurately combined with a nearby Ensembl gene. If multiple associations or other inconsistencies are found, probe set coordinates are matched directly to the exon cluster locations derived in `EnsemblImport.py`.
3) (The following method is deprecated) - For AltAnalyze version 1.13 and below, once unique probe set-to-Ensembl gene associations have been defined, constitutive probe sets are identified using the Affymetrix mRNA counts provided in the program `ExonArrayEnsemblRules.py`. The mRNA counts are distributed based on the types of mRNAs they align to (full-length, Ensembl, and EST), where the probe sets with the largest number of high quality mRNA associations are chosen as constitutive. Probe sets for a given gene are ranked based on the: A) number of Ensembl transcripts, B) full-length and C) ESTs associated, in descending order, where multiple associations are required for

each annotation type. If all probe sets have the same number of Ensembl and full-length transcript associations, then the number of EST aligning are compared. If no difference in these mRNA assignments exists, no constitutive probe sets are annotated.

4) Each probe sets is then aligned to exon clusters, regions, retained introns, constitutive exon regions and splicing annotations for that gene. In addition to splicing annotations from `EnsemblImport.py`, splicing annotations from the UCSC genome annotation file "knownAlt.txt" (found in the same server directory at UCSC as "all_mRNA.txt") using the program `alignToKnownAlt.py`, are aligned to these probe sets. If a probe set does not align to an `Ensemblmport.py` defined exon or intron and is upstream of the first exon or downstream of the last exon, the probe set is assigned a UTR annotation (e.g., U1.1). All aligning probe sets are annotated based on the exon cluster number and the relative position of that probe set in the exon cluster, based on relative 5' genomic start (e.g., E2.1). This can mean that probe set E2.1 actually aligns to the second exon cluster in that gene in any of the exon regions (not necessarily the first exon region), if it is the most 5' aligning.
5) These probe set annotations are exported to the directory "AltDatabase/*species*/exon" with the filename "*species* _Ensembl_probe sets.txt". Exon block and region annotations for each probe set are designated in the AltAnalyze result file.

Junction Arrays

For the exon-junction array AltMouseA, the same process is applied to the highlighted exon(s) from all pre-determined reciprocal probe sets, exported by the program `ExonAnnotate_module.py`. A highlighted exon is an exon that is considered to be regulated as the result of two alternative junctions. For example, if

examining the exon-junctions E1-E2 and E1-E3, E2 would be the highlighted exon. Alternatively, for the mutually-exclusive splicing event E2-E4 and E1-E3, E2 and E3 would be considered to be the highlighted exons. To obtain the genomic locations of these exons, sequences for each are obtained from a static build of the mouse AltMerge program (March 2002) (`ExonAnalyze_module.py`) and searched for in FASTA formatted sequence obtained from BioMart for all Ensembl genes with an additional 2 kb upstream and downstream sequence (`JunctionArray.py` and `EnsemblImport.py`). This allows for the export of an exon-coordinate file analogous to the exon probeset.csv file. The main difference in this file is that AltMouseA gene to Ensembl ID associations are obtained by comparing gene symbol names and external GenBank accession numbers in common, as opposed to coordinate comparisons. The resulting highlighted exon file is named "Mm_Ensembl_AltMouse_probe sets.txt" and is saved to "AltDatabase/Mm/AltMouse", with the same structure as its exon array analogue.

6.3 Extracting UniProt Protein Domain Annotations Overview

The UniProt protein database is a highly curated protein database that provides annotations for whole proteins as well as protein segments (protein features or domains). These protein feature annotations correspond to specific amino acid (AA) sequences that are annotated using a common vocabulary, including a class (feature key) and detailed description field. An example is the TCF7L1 protein (http://www.uniprot.org/uniprot/Q9HCS4), which has five annotated feature regions, ranging in size from 7 to 210 AA. One of these regions has the feature key annotation "DNA binding" and the description "HMG box". To utilize these annotations in AltAnalyze, these functional tags are extracted along with full protein sequence, and external annotations for each protein (e.g., Ensembl gene) from the

“uniprot_sprot_*taxonomy*.dat” file using the `ExtractUniProtFunctAnnot.py` program. FTP file locations for the UniProt database file can be found in the file “Config/Default-file.csv” for each supported species. To improve Ensembl-UniProt annotations, these relationships are also downloaded from BioMart and stored in the folder “AltDatabase/uniprot/*species*” as “*species*_Ensembl-UniProt.txt”, which are gathered at `ExtractUniProtFunctAnnot.py` runtime to include in the UniProt sequence annotation file. These files are saved to “AltDatabase/uniprot/*species*” as “uniprot_feature_file.txt” and “uniprot_sequence.txt”.

6.4 Extracting Ensembl Protein Domain Annotations Overview

In addition to protein domains/features extracted from UniProt, protein features associated with specific Ensembl transcripts are extracted from the Ensembl database. One advantage of these annotations over UniProt, is that alternative exon changes that alter the sequence of a feature but not its inclusion will be reported as a gain and loss of the same feature, as opposed to just one with UniProt. This is because protein feature annotations in UniProt only typically exist for one isoform of a gene and thus, alteration of this feature in any way will result in this feature being called regulated. Although an Ensembl annotated feature with a reported gain and loss can be considered not changed at all, functional differences can exist do to a minor feature sequence change that would not be predicted if the gain and loss of the feature were not reported.

Three separate annotation files are built to provide feature sequences and descriptions, “Ensembl_Protein”, “Protein”, and “ProteinFeatures” files (“AltDatabase/ensembl/*species*”). The “ProteinFeatures” contains relative AA positions for protein features for all Ensembl protein IDs, genomic start and end locations along with InterPro annotations and IDs. Only those InterPro domains/features with an alignment e-score < 1 are stored for alignment to regulated exons. The “Protein”

file contains AA sequences for each Ensembl protein. The “Ensembl_Protein " provides Ensembl gene, transcript and protein ID associations. Data for these files are downloaded and extracted using the previously mentioned `EnsemblSQL.py` module. The feature annotation source in these files is InterPro, which provides a description similar to UniProt. As an example, see: http://ensembl.genomics.org.cn/Homo_sapiens/protview?db=core;peptide=ENSP00000282111, which has similar feature descriptions to UniProt for the same gene, TCF7L1.

6.5 Extracting microRNA Binding Annotations Overview

To examine the potential gain or loss of microRNA bindings sites as the direct result of exon-inclusion or exclusion, AltAnalyze requires putative microRNA sequences from multiple prediction algorithms. These binding site annotations are extracted from the following flat files:

- TargetScan conserved predicted targets (http://www.targetscan.org/cgi-bin/targetscan/data_download.cgi?db=vert_42). Gene symbol and putative microRNA associations are extracted (no sequence). The primary gene ID, gene-symbol, is linked to Ensembl based on BioMart downloaded gene-symbol to Ensembl gene annotations (from several Ensembl builds – outdated symbols are often used).
- Miranda human centric with multi-species alignment information was obtained from target predictions organized by Ensembl gene ID (http://cbio.mskcc.org/research/sander/data/miRNA2003/mammalian/index.html). A larger set of associations was also pulled from species-specific files (http://www.microrna.org/microrna/getDownloads.do), where gene symbol was related to Ensembl gene. Both files provided target microRNA sequence.

- Sanger center (miRBase) sequence was provided as a custom (requested) dump of their version 5 target predictions (http://microrna.sanger.ac.uk/targets/v5/), containing Ensembl gene IDs, microRNA names, and putative target sequences, specific for either mouse or human. Currently, rat has not been obtained.
- PicTar conserved predicted targets (from Dog to Human) were provided as supplementary data (Supplementary Table 3) at http://www.nature.com/ng/journal/v37/n5/suppinfo/ng1536_S1.html, with conservation in human, chimp, mouse, rat, and dog for a set of 168 microRNAs. For mouse, human gene symbols were searched for in the BioMart derived "Mm_ Ensembl_annotation.txt" table after converting these IDs to a mouse compatible format (e.g., TCF7L1 to Tcf7l1). The same was used for aligning to PicTar. The same strategy is used for rat.

Ensembl gene to microRNA name and sequence are stored for all prediction algorithm flat files and directly compared to find genes with one or more lines of microRNA binding site evidence using the program `MatchMiRTargetPredictions.py`. The flat file produced from this program ("combined_gene-target-sequences.txt") was used by the program `ExonSeqSearch.py` to search for these putative microRNA binding site sequences among all probe sets from the "*species*_Ensembl_probe set.txt" file built by `ExonArrayEnsemblRules.py` and probe set sequence from the Affymetrix 1.0 ST probe set fasta sequence file (Affymetrix) or the reciprocal junction highlighted exon sequence file (see section 6.2). For gene arrays, NetAffx currently only provides probe sequences, thus, probe set sequences from the exon array are used for probe set found to overlap in genomic space. Two resulting files, one with any binding site predictions and another required to have evidence from at least two algorithms, are saved to

“AltDatabase/*species*/*array_type*/” as “*species*_probe set_microRNAs_any.txt” and “*species*_probe set_microRNAs_multiple.txt”, respectively.

6.6 Inferring Protein-Probe Set Associations Overview

To obtain associations between specific probe sets and proteins, the programs `IdentifyAltIsoforms.py` (exon and junction arrays) and `ExonSeqModule.py` (junction arrays only) were written. The program `IdentifyAltIsoforms.py` grabs all gene mRNA transcripts and associated exon genomic coordinates from Ensembl and UCSC and compares these to probe set coordinates (or critical exon for junction arrays) to find pairs of transcripts (one containing the probe set or critical exon and the other not) that have the least number of differing exons (see Section 3.4). These transcript pairs are thus most likely to be similar with exception to the region containing the probe set or critical exon. Once these best matches are identified, corresponding protein sequences for each mRNA are downloaded from Ensembl using `EnsemblSQL.py` or are downloaded using NCBI webservices via BioPython’s `Entrez` function.

If protein sequences are unavailable for an mRNA accession, the mRNA sequence for that identifier is downloaded (using the above mentioned services) and translated using the custom `IdentifyAltIsoforms.py` function “BuildInSilicoTranslations”, which uses functions from the BioPython module to translate an mRNA based on all possible start and stop sites to identify the longest putative translation that also shares either the first or last 5 AA of its sequence with the N-terminus or C-terminus (respectfully) of a UniProt protein.

While this same protocol is used for junction arrays, prior to this analysis reciprocal probe sets are mapped to all possible aligning and non-aligning transcripts through direct probe set sequence comparison via `ExonSeqModule.py`. This module takes the consensus probe set sequence for all reciprocal probe set pairs (defined in

the file
“AltMouse_junction_comparisons.txt”) and searches for a match among mRNA transcript FASTA formatted sequences from Ensembl and UCSC mRNAs that correspond to that gene. Only a 100% sequence match is allowed, (matches may not occur do to polymorphisms between sequence sources and genomic assemblies). These associations are then used by `IdentifyAltIsoforms.py` as described above.

At this point, only two mRNAs are matched to each probe set or junctions, matching and non-matching mRNAs. Next, differences in protein feature composition between these two proteins, alternative N or C-terminal sequences, coding sequence and protein length are assessed using `ExonAnalyze_module.py` and exported to text files (associations and protein sequences) for import when AltAnalyze is run. These two files have the suffix “exoncomp.txt”. Two analogous files with the suffix “seqcomp.txt”, are derived by identifying the best matching and non-matching mRNAs based on comparison of protein sequence as compared exon composition (e.g. least differences in protein domain composition), but are currently used only for internal analyses (contact the authors for the seqcomp files to replace exoncomp).

6.7 Required Files for Manual Update

Many external databases are required for the above build strategies. Advanced users may wish to update databases using the `update.py` module in the folder “Source_code” (currently requires moving all source files to the main directory and running Python through a terminal program). From there on, you will be presented with several options. To find or change the download location of any automated downloads, see the file “Config/Default-file.csv” and “Config/EnsemblSQL.txt”. In the current

“Default-file.csv” file, the web location of all Ensembl files to download is pointed towards build 49 of Ensembl. To update the data for a new build of Ensembl, these URL will need to be replaced with the address of the target build, identified by examining the URLs in the parent directories. These changes are all that should be required when downloading a new version of Ensembl, since AltAnalyze will download and parse these files using the `update.py`.

When updating the Affymetrix, Ensembl, UniProt and UCSC genome database associations, the user must ensure that all of these databases are built off the same genomic build (e.g., NCBI36 aka HG18). Currently, two files must be downloaded manually from the Affymetrix website (probe set sequence and annotation files for exon arrays). Additional functionality and user-interface based control for obtaining AltAnalyze updates will be included in subsequent builds of AltAnalyze. For the AltMouseA array, full gene sequences in fasta format with 2kb upstream and downstream must be downloaded from BioMart in order to update genomic coordinate alignments for any new NCBI genomic build.

Section 7 – Evaluation of AltAnalyze Predictions

To assess AltAnalyze exon array analysis performance relative to other published approaches, we analyzed published experimental confirmation results for a dataset of splicing factor knockdown (mouse polypyrimidine tract binding protein (PTB) short-hairpin RNA (shRNA)). From two independent analyses (12, 13), alternative splicing (AS) for 109 probe sets was assessed in the mouse PTB shRNA dataset by RT-PCR (Supplemental Tables 1,2 and 4 from the referenced study) (13). Among these, 25 were false positives, one was a true negative, one undetermined and 81 were true positives.

Summary of Published MADS Results

In the analysis by Xing et al., alternative exons discovered by multiple splicing array platforms and RT-PCR were examined using a new algorithm named microarray analysis of differential splicing or MADS. MADS implements a modification of the splicing index method on gene expression values obtained using multiple scripts from GeneBase and filtering of probes predicted to hybridize to multiple genomic targets. Using this algorithm, the authors were able to verify AS detected by RT-PCR of mouse Affymetrix Exon 1.0 array data as well as predict and validate 27 novel splicing events by RT-PCR. Using the microarray CEL files posted by Xing and colleagues, we ran AltAnalyze using default options (same as used in the corresponding primary report), which includes quantile normalization via RMA-sketch using AltAnalyze's interface to Affymetrix Power Tools.

AltAnalyze Results

Of the 109 probe sets linked to splicing events characterized by RT-PCR, 78 were analyzed by AltAnalyze. Those probe sets not analyzed by AltAnalyze were either

not apart of the AltAnalyze ”core” probe sets or were excluded due to high detection p-value or low expression thresholds. A break-down of the number of RT-PCR true positive, false positive, undetermined and false negative (out of the 81 documented true positives) is shown for various AltAnalyze filters (Table 7.1).

Of the 78 analyzed probe sets, 26 were called by AltAnalyze to be alternatively regulated (using default parameters), out of 194 probe sets called by AltAnalyze as alternatively regulated. All 26 probe sets were annotated as true positives according to the published RT-PCR data. Although 17 probe sets were RT-PCR false positives among the 78 probe sets with RT-PCR data, only one false positive was considered to be alternative regulated by AltAnalyze with any of the AltAnalyze filters alone (splicing-index $p<0.05$ alone without additional default options). The MADS algorithm was able to validate AS for 27 novel splice events corresponding to 41 probe sets. Of these 41 probe sets, 33 were examined by AltAnalyze and 23 were considered alternatively regulated.

Conclusions

This analysis suggests that AltAnalyze analysis using default parameters produces conservative results with high specificity (100% true positives in this analysis) with reasonable sensitivity (~42% that of MADS). For smaller datasets, such as the PTB knock-down comparison, the decreased sensitivity results will have a significant impact on the number of true splicing events detected, however, for larger datasets with thousands of regulated probe sets, AltAnalyze is likely to reduce the number of false positives and reduction in overall noise. It is important to note however, for the MADS analysis only a p-value threshold was used and for AltAnalyze, both a MiDAS and splicing-index p-value in addition to splicing-index fold change thresholds were used. For these analyses, we have not filtered probe sets based on association with annotated splicing events (see Materials and Methods for description), which should

further decrease false positives. If probe sets with annotated splicing events are filtered out, 20 versus 26 true positives will remain.

Although a false positive rate of up-to 50% has been reported with the conventional splicing-index implementation (e.g., using the Affymetrix ExACT software) (13), AltAnalyze's analysis differs in several ways. First, in this analysis RMA-sketch was used as the method for quantile normalization. After obtaining expression values for probe sets (no low level filtering currently implemented), probe sets for each of the two biological groups are filtered based on two main parameters: DABG p-value and mean expression filters. Any probe set with a DABG p-value > 0.05 in both biological groups or mean expression < 70 are excluded.

Additional AltAnalyze specific parameters relate to how probe set to gene associations are obtained, which probe sets are selected for analysis and how probe sets are selected for calculation of gene expression. Unlike ExACT, probe set to gene association are via genomic coordinate alignment to Ensmebl/UCSC mRNA transcripts for unique Ensembl genes rather than to Affymetrix transcript clusters. This process ensures that a probe set only aligns to one Ensembl gene. Probe sets can align to an exon, intron or UTR of a gene. Any probe set aligning to an analyzed mRNA is used for analysis in the AltAnalyze "core" set along with any Affymetrix annotated "core" probe set. To determine gene expression from the exon-level a two-step method is employed. First, probe sets that are most over-represented among mRNAs or mRNAs and ESTs (associations from Affymetrix probe set annotation file) are selected as constitutive. Next if constitutive probe set is "expressed" in both biological groups (using the DABG and mean expression filters listed above), the probe set is retained for constitutive expression calculation. If more than one "expressed" constitutive probe set is present, the mean expression of all constitutive probe sets for each array is calculated. As a final step, probe sets with an associated gene expression difference between the two array groups greater than 3 are not reported. These analysis steps result in a unique

splicing-index, splicing-index *t* test p-value and MiDAS p-value calculation from other analysis methods.

Since this validation set provides a limited test case for analysis of type I (false positive) and type II (false negative) errors, the AltAnalyze algorithms will continue to be assessed as additional validation data is made available. Future implementations will likely include "low-level" analyses that reduce the occurrence of type I errors (e.g., elimination of expression data for specific probes that introduce additional noise). However, this data supports the concept that AltAnalyze produces conservative results with a level of confidence.

				AltAnalyze RMA Analysis				
	All	**MADS p<0.05**	**MADS p<0.01**	**All**	**SF>2**	**SP <0.05**	**SP|MP <0.05**	**SF>2, GEF<3, SP|MP <0.05**
TP	81	62	55	60	30	36	32	26
FP	25	6	2	17	0	1	0	0
UD	1	1	1	1	0	1	1	0
FN	0	19	26	21	51	45	49	55

Table 7.1 - Analysis of PTB shRNA verified splicing events with AltAnalyze. The number of probe sets matched to different RT-PCR absence or presence calls for AS from an analysis of mouse PTB shRNA knockdown of a neuroblastoma cell line compared to empty vector shRNA knockdown using the Affymetrix Mouse 1.0 Exon array [12]. Results are shown for all probe sets (All), MADS p<0.05 or p<0.01 (after removing cross-hybridizing probes), AltAnalyze "core" probe sets (All) analyzed by RT-PCR, splicing-index fold change >2, splicing-index t-test p (SP) <0.05, MiDAS p (MP) <0.05 or combination of all three options (SF>2 & SP & MP <0.05 – default option) along with gene expression filtering (GEF) of < 3 fold, for true positive (TP), false positive (FP), un-determined (UD), and

false negative (FN) RT-PCR results. False negative probe set counts are relative to all original true positives experimentally identified, independent of whether they were considered by AltAnalyze.

Section 8 - Analysis of AltAnalyze Results DomainGraph

Once alternative probe sets have been identified from an AltAnalyze exon array analysis, you can easily load this data in the Cytoscape plugin DomainGraph (http://domaingraph.bioinf.mpi-inf.mpg.de/) to:

1) View prioritized AltAnalyze highlighted alternative exons and annotations.
2) Assess which probe sets overlap with a set of loaded genes and which specific protein domains at a high-level (protein/domain/miRNA binding stie network/pathway) and low-level (domain/exon/probe set view).
3) View alternative probe set data in the context of WikiPathways and Reactome gene networks.

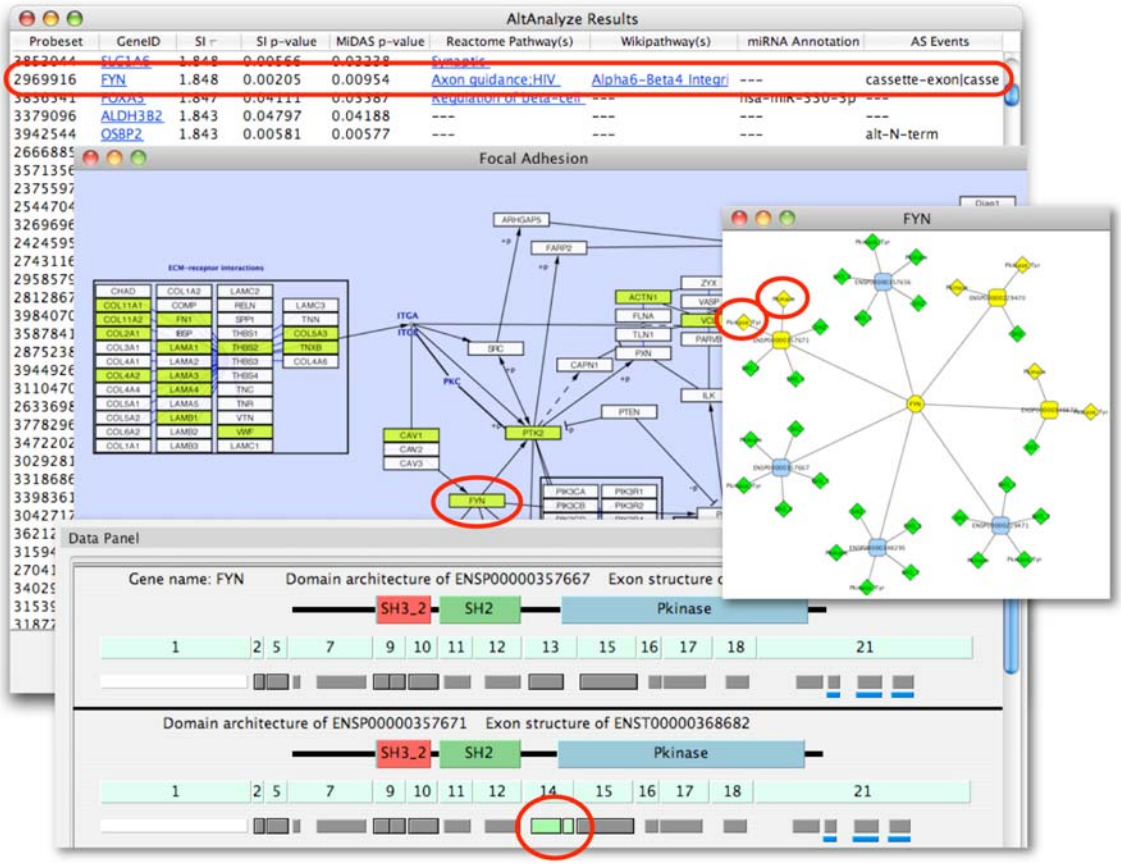


Figure 8.1. Visualization of AltAnalyze regulated probe sets along exons and protein domains. In the top panel, a loaded Cytoscape DomainGraph network is shown for the gene FYN, with relevant protein domain interactions shown between two alternative isoforms of each gene. Rounded boxes represent gene nodes and diamonds, protein domains and other functional elements. Greenish yellow nodes represent those containing AltAnalyze regulated probe sets, whereas green do not overlap with an AltAnalyze regulated probe set. The gene FYN has been selected in the main network that creates a domain architecture and exon structure view for the select FYN isoform in the Cytoscape "Data Panel". Domains (top), exons (middle) and probe sets (bottom) are shown that correspond to the FYN isoforms ENSP00000229470 and ENSP00000229471, with AltAnalyze down-regulated probe sets in green. Probe sets with a solid black border are associated with an alternative splicing (alternative cassette exon) or alternative promoter annotation. In this example, the probe set with an alternatively splicing annotation overlaps with exon 8 and the Protein Kinase domain of the protein. Probe sets with a blue bar beneath them overlap with predicted microRNA binding sites. Details about each domain, exon, probe set and microRNA binding site, including AltAnalyze statistics and functional annotations are accessible by mousing-over the respective feature and by left-clicking the object to link-out to resources on the web.

Installing DomainGraph with AltAnalyze

- Simply download and extract AltAnalyze 1.15 (or higher) and Cytoscape, DomainGraph and the GPML puglin will be ready to use. This plugin can be updated through the Cytoscape plugin manager.

- Start Cytoscape directly from AltAnalyze after the alternative exon results are produced.
- Download the species gene database of interest (first time only – see below)

Running DomainGraph

Detailed instructions on running DomainGraph can be found here:

http://www.altanalyze.org/domaingraph.htm

and at:

http://domaingraph.bioinf.mpi-inf.mpg.de/

or in the AltAnalyze application folder:

AltAnalyze_v1release/Documentation/domain_graph.pdf

References

1. Cline MS*, et al.* (2007) Integration of biological networks and gene expression data using Cytoscape. *Nat Protoc* 2(10):2366-2382.
2. Hubbard TJ*, et al.* (2007) Ensembl 2007. *Nucleic Acids Res* 35(Database issue):D610-617.
3. Karolchik D*, et al.* (2008) The UCSC Genome Browser Database: 2008 update. *Nucleic Acids Res* 36(Database issue):D773-779.
4. Salomonis N*, et al.* (2007) GenMAPP 2: new features and resources for pathway analysis. *BMC Bioinformatics* 8:217.
5. van Iersel MP*, et al.* (2008) Presenting and exploring biological pathways with PathVisio. *BMC Bioinformatics* 9:399.
6. Srinivasan K*, et al.* (2005) Detection and measurement of alternative splicing using splicing-sensitive microarrays. *Methods* 37(4):345-359.
7. Gardina PJ*, et al.* (2006) Alternative splicing and differential gene expression in colon cancer detected by a whole genome exon array. *BMC Genomics* 7:325.
8. Purdom E*, et al.* (2008) FIRMA: a method for detection of alternative splicing from exon array data. *Bioinformatics* 24(15):1707-1714.
9. Ule J*, et al.* (2005) Nova regulates brain-specific splicing to shape the synapse. *Nat Genet* 37(8):844-852.
10. Wheeler R (2002) A method of consolidating and combining EST and mRNA alignments to a genome to enumerate supported splice variants *Algorithms in Bioinformatics: Second International Workshop, Springer Berlin / Heidelberg* Volume 2452/2002:201–209.
11. Sugnet CW*, et al.* (2006) Unusual intron conservation near tissue-regulated exons found by splicing microarrays. *PLoS Comput Biol* 2(1):e4.
12. Boutz PL*, et al.* (2007) A post-transcriptional regulatory switch in polypyrimidine tract-binding proteins reprograms alternative splicing in developing neurons. *Genes Dev* 21(13):1636-1652.
13. Xing Y*, et al.* (2008) MADS: a new and improved method for analysis of differential alternative splicing by exon-tiling microarrays. *Rna* 14(8):1470-1479.