Package: agilp (via r-universe)

October 21, 2026

Type Package

Title Agilent expression array processing package

Version 3.45.0

Date 2012-06-10

Author Benny Chain <b.chain@ucl.ac.uk>

Maintainer Benny Chain <b.chain@ucl.ac.uk>

Depends R (>= 2.14.0)

Description More about what it does (maybe more than one line)

License GPL-3

Repository https://bioc.r-universe.dev

Date/Publication 2026-04-28 12:36:56 UTC

RemoteUrl https://github.com/bioc/agilp

RemoteRef HEAD

RemoteSha a161e370fa4bdf336e54838c31c5eecf72d20956

Contents

AALoess 2
AAProcess 3
Baseline 5
Equaliser 6
filenamex 8
IDswop 9
Loader 11

Index **13**

AALoess *Normalises a set of gene expression data files using LOESS*

Description

This script normalises a set of gene expression data files against a predefined reference data set. The normalisation uses LOESS regression. The major assumption is that the actual data and the reference set have a similar epxression frequency distribution. Typically, the reference set used the is average of a large number of datasets such as the output of Baseline script in this package. The script also calculates the Sum of Squared Errror (SSE) between each data point of the normalised data set and the reference, and plots the results as a histogram of the SSE of all the data sets processed. This function is useful to identify outliers, which are usually due to problems in sample prepration/processing, which are frequently not identified by using the internal scanner quality performance indicators.

Usage

```
AALoess(input=file.path(system.file(package="agilp"),"input",""),output=file.path(system.file(packa
```

Arguments

input	full path of directory where input data files are put; default is a folder named input within the agilp package directory; the input directory must contain ONLY the set of files to be processed. The input files must contain two columns only, the source identifier in the first column, and the expression data in the second column; see example files
output	full path of directory where output data files are put; default is a folder named output within the agilp package directory
LOG	if NORM="LOG", the data are log base 2 transformed before normalisation. The default is NORM="LOG"
baseline	full path of file containing the reference data set for normalisation; the default is agilp/input1/baseline.txt

Value

normalised datafiles
: One new file is written to the output folder for each input file. The format of the output files is the same as the input, but the data has been normalised. The normalised files are identified by the prefix nr_ or ng_

SSE_date.txt file
: A file in the output folder containing a list of all the filenames in the input folder, and the corresponding SSE value

.

Author(s)

Benny Chain; b.chain@ucl.ac.uk

References

In preparation; for further detail on normalisation see also Chain B, Bowen H, Hammond J, Posch W, Rasaiyaah J, Tsang J, Noursadeghi M. Error, reproducibility and sensitivity: a pipeline for data processing of Agilent oligonucleotide expression arrays. BMC Bioinformatics. 2010 Jun 24;11:344.

See Also

AAProcess filenamex Loader Baseline IDswop Equaliser

Examples

```
#Takes four files of raw data (output of AAProcess, in dataset/raw folder) , LOess normalises them and saves them in
inputdir<-file.path(system.file(package="agilp"),"extdata","raw/","", fsep = .Platform$file.sep)
outputdir<-file.path(system.file(package="agilp"),"output/", "", fsep = .Platform$file.sep)
baselinedir<-file.path(system.file(package="agilp"),"extdata","testbase.txt", fsep = .Platform$file.sep)
AALoess(input=inputdir, output=outputdir, baseline = baselinedir, LOG="TRUE")
## Not run:
#to remove these files again and empty the output directory use
unlink(paste(file.path(system.file(package="agilp"),"output",""),"*.*",sep=""), recursive=FALSE)

## End(Not run)
```

AAProcess — *Extracts raw expression data from Agilent expression array scanner files.*

Description

This function takes scanner "txt format" files produced by the Agilent microarray scanner, extracts just the raw median intensity values from the columns marked "gMedianSignal" or "rMedianSignal". Replicate probes are averaged. The expression data for each colour is written as separate txt tab delimited files with the same name as the original plus prefix rRaw_ or gRaw_. Probe names are in the first column, expression values in the second column.

Usage

```
AAProcess(input=file.path(system.file(package="agilp"),"input",""),output=file.path(system.file(pac
```

Arguments

input	full path of directory where input data files are put; default is a folder named input within the agilp package directory
output	full path of directory where output data files are put; default is a folder named output within the agilp package directory
s	number of header lines to skip when reading in scanner files. Default is 9

Details

The input directory must have ONLY the data files to be analysed.

Value

`rawdata files` The function writes a set of tab delimited txt files containing the raw data for each channel to directory defined by output. The raw data files are given the same name as the input data files with the prefix Rawg and Rawr for green and red channel respectively. The output files can be readily opened in Excel. In R each file is a dataframe, which contains probe names in first column, and expression data in second column. Replicate probes are averaged.

Author(s)

Benny Chain; b.chain@ucl.ac.uk

References

In preparation

See Also

`Loader` `filenamex` `IDswop` `Baseline` `Equaliser` `AALoess`

Examples

```
dir()

#This examples extracts expression data from two sample array scanner files which are found in the folder "scanner"
#and places them in the folder output.
inputdir<-file.path(system.file(package="agilp"),"extdata","scanner","", fsep = .Platform$file.sep)
outputdir<-file.path(system.file(package="agilp"),"output/", "", fsep = .Platform$file.sep)
AAProcess(input = inputdir, output = outputdir, s = 9)

## Not run:
#to remove these files again and empty the output directory use
unlink(paste(file.path(system.file(package="agilp"),"output\","","*.*",sep=""), recursive=FALSE)

#The number of lines header to be skipped from scanner array files can be changed from the default value of 9 by setti
AAProcess(input = inputdir, output = outputdir, s = 12)

## End(Not run)
```

Baseline *Constructs a file with the mean of each probe from a set of raw expression array data files*

Description

Calculates the mean expression data for each probe or identifier for a set of expression array data sets, for example as produced by AAProcess. Identifiers (e.g. probe names) must lie in the first column of each file. Identifiers not common to all files in set are discarded. The set of files is either all files in input directory or a set of files defined using a template file. Template is typically an Excel spreadsheet saved as a tab delimited txt fiel, with file names as one column. The files to be used must be specified in a contiguous set of rows of the template file.

Usage

```
Baseline(NORM="LOG",allfiles="TRUE",r=2,A=2,B=3,input=file.path(system.file(package="agilp"),"input
```

Arguments

input	full path of directory where input data files are put; default is a folder named input within the agilp package directory
t	full name and path of template file; default is the file template.txt in folder agilp/input1
baseout	full name and path of output file with mean expression values; default is the file baseline.txt in folder agilp/output
NORM	if NORM="LOG", the data are log base 2 transformed before the mean is calculated. The default is NORM="LOG"
r	The column of template which contains the file names
A	The first row of template to be used; note a header row is NOT ignored.
B	The last row of template to be used
allfiles	If allfiles is TRUE, the script uses all files in the input directory, and template is ignored; if allfiles=FALSE, the scritp uses files defined by template

Details

This function is typically used as part of the data processing pipeline to calculate an average against which to normalise all data files using AALoessfunction

Value

baseout	A tab delimited text file, with each identifier and its mean value,is written to the filename and path defined by baseout

Warning

The function checks that files listed in template exist in folder, and outputs a list of all file names not found

Author(s)

Benny Chain; b.chain@ucl.ac.ucl

References

In preparation

See Also

AAProcess filenamex IDswop Loader Equaliser AALoess

Examples

```
#Takes four files of raw data in folder agilp/extdata/raw, calculates the mean expression value for each probe, and

inputdir<-file.path(system.file(package="agilp"),"extdata","raw/","", fsep = .Platform$file.sep)
outputbase<-file.path(system.file(package="agilp"),"output", "testbase.txt", fsep = .Platform$file.sep)
template<-file.path(system.file(package="agilp"),"extdata","sample_template.txt", fsep = .Platform$file.sep)
Baseline(NORM="LOG",allfiles="TRUE",r=2,A=2,B=3,input=inputdir, baseout=outputbase, t = template)

#Alternatively the following example uses only those data files defined in column 2, rows 2-5 of the template file. I
Baseline(NORM="FALSE",allfiles="FALSE",r=2,A=2,B=5,input=inputdir, baseout=outputbase, t = template)

## Not run:
#to remove the output files again and empty the output directory use
unlink(paste(file.path(system.file(package="agilp"),"output",""),"*.*",sep=""), recursive=FALSE)

## End(Not run)
```

Equaliser — *Trims a set of gene expression data files to include only the set of identifiers common to all files*

Description

This script processes a set of gene expression data files which have different number of entries (i.e. different numbers of identifiers). Further processing of data files usually requires equal number of identifiers in each file (e.g. AALoess or Baseline in this package). The Equaliser script cycles through a series of such datafiles, determines the set of identifiers common to all files in the directory, and outputs a set of files which retain only those identifiers common to the whole set.

Usage

```
Equaliser(input=file.path(system.file(package="agilp"),"input",""),output=file.path(system.file(pac
```

Arguments

`input`	full path of directory where input data files are put; default is a folder named input within the agilp package directory; the input directory must contain ONLY the set of files to be processed. The input files must contain two columns only, the source identifier in the first column, and the expression data in the second column; see example files
`output`	full path of directory where output data files are put; default is a folder named output within the agilp package directory

Details

This utility is particularly useful to process the output of IDswop in the agilp package. IDswop excludes some identifiers on the basis of a cutoff which is data dependent, and may therefore generate files with different numbers of identifiers between samples.

Value

`equalised datafiles`

One new file is produced for each input file. The file name has an additonal s_ as a suffix.

Author(s)

Benny Chain; b.chain@ucl.ac.uk

References

In preparation

See Also

`AAProcess` `filenamex` `Loader` `Baseline` `IDswop` `AALoess`

Examples

```
##Takes four files of raw data (output of AAProcess, in folder agilp/dataset/raw) selects all entries common to all

inputdir<-file.path(system.file(package="agilp"),"extdata","raw/","", fsep = .Platform$file.sep)

outputdir<-file.path(system.file(package="agilp"),"output/", "", fsep = .Platform$file.sep)

Equaliser(input = inputdir, output = outputdir)

## Not run:
#to remove these files again and empty the output directory use
unlink(paste(file.path(system.file(package="agilp"),"output",""),"*.*",sep=""), recursive=FALSE)


## End(Not run)
```

filenamex *A file name listing utility*

Description

Generates a text file with the name of all files in a given directory and saves it as a tab delimited txt file. Helpful for filling in template files (see Loader) without making mistakes

Usage

```
filenamex(input=file.path(system.file(package="agilp"),"input",""),output=file.path(system.file(pac
```

Arguments

input	full path of directory where input data files are put; default is a folder named input within the agilp package directory
output	full path of directory where output data files are put; default is a folder named output within the agilp package directory

Details

None

Value

names.txt	A tab delimited text file with the names of all the files in the input directory

Author(s)

Benny Chain; b.chain@ucl.ac.ucl

References

In preparation

See Also

AAProcess Loader IDswop Baseline Equaliser AALoess

Examples

```
#This example makes a list of files in the folder agilp/extdata/raw and saves it in a file called names.txt (tab deli
inputdir<-file.path(system.file(package="agilp"),"extdata","raw/","", fsep = .Platform$file.sep)
outputdir<-file.path(system.file(package="agilp"),"output/", "", fsep = .Platform$file.sep)
filenamex(input=inputdir,output=outputdir)

## Not run:
#to remove these files again and empty the output directory use
unlink(paste(file.path(system.file(package="agilp"),"output",""),"*.*",sep=""), recursive=FALSE)
```

```
## End(Not run)
```

IDswop *Mapping expression data across bioinformatic identifiers*

Description

This script maps expression data associated with a source identifier (typically a platform specific identifier such as ProbeID, as output by AAProcess in the agilp package for example), to another identifier of the user's choice. The script uses an annotation file which contains the mappings between identifiers. Each column denotes a different identifier, which is shown in the column heading. Typical examples include ProbeID, EntrezGeneID, EnsemblTranscriptID etc.) Annotation files are provided by microarray manufacturers,or, preferably are created using, for example the biomaRt package in Bioconductor.

Usage

```
IDswop(input=file.path(system.file(package="agilp"),"input",""),output=file.path(system.file(packag
```

Arguments

input	full path of directory where input data files are put; default is a folder named input within the agilp package directory; the input directory must contain ONLY the set of files to be processed. The input files must contain two columns only, the source identifier in the first column, and the expression data in the second column; see example files in datasets folder
output	full path of directory where output data files are put; default is a folder named input within the agilp package directory
fun	the function to be applied in cases where multiple probes map to the same identifier; currently allowed values are "mean" or "max".
annotation	full path of annotation file; default is a file called annotation.txt within a folder named input1 within the agilp package directory
source_ID	The identifier from which the data is mapped. This must appear exactly as it appears in the heading of the appropriate column of the annotation file.
target_ID	The identifier to which the data is mapped. This must appear exactly as it appears in the heading of the appropriate column of the annotation file
ERR	The cut-off for coefficient of variation allowed when multiple probes map to a single identifier. ERR must lie between 0 and 1. If ERR=0, any difference between such probes will result in the identifier being omitted from the final list. If ERR=1, all identifiers are retained. The default is 0.2, allowing a maximum coefficient of variation between multiple probes mapping to a single identifier of 20%.

Details

This function is designed for merging expression data sets from different platforms using common bioinformatic identifiers such as gene names, transcript names etc. The major problem lies in how to deal with the "many to many" mapping which characterise the relationships between most standard identifiers. IDswop deals with the problem as follows. If multiple source identifiers (typically platform specific probe IDs) map to a single target (for example Ensembl_Transcript_ID) , the different probes are either averaged (fun="mean") or the maximum is selected (fun="max"). However, in cases where the discrepancy between such sets of probes is too large, the identifier is discarded, since it is considered as being unreliable. The degree of error allowed can be set by the user, as described under a descrition of the ERR variable given above. In cases where a single source identifier maps to multple targets (e.g. one probe to several transcript IDs) each transcript is retained as a separate entry in the final list, with an identical expression value.

Value

`remapped datafiles`

One new file is produced for each input file. Each file contains two columns; the first contains the new set of identifiers; the second contains the corresponding expression value. The file name contains the new identifier as a suffix.

Author(s)

Benny Chain; b.chain@ucl.ac.uk

References

In preparation

See Also

`AAProcess` `filenamex` `Loader` `Baseline` `Equaliser` `AALoess`

Examples

```
#This examples maps expression data from Agilent ProbeIDs (found in column one of input files) to corresponding Ense


inputdir<-file.path(system.file(package="agilp"),"extdata","raw/","", fsep = .Platform$file.sep)
outputdir<-file.path(system.file(package="agilp"),"output/", "", fsep = .Platform$file.sep)
annotation<-file.path(system.file(package="agilp"),"extdata","annotations_sample.txt", fsep = .Platform$file.se
IDswop(input=inputdir,output=outputdir,annotation=annotation,source_ID="ProbeID",target_ID="EnsemblID", ERR=0.2

#Alternatively the following keeps all output IDs, even when multiple probes map to a single

IDswop(input=inputdir,output=outputdir,annotation=annotation,source_ID="ProbeID",target_ID="EnsemblID", ERR=1)

## Not run:
#to remove the output files again and empty the output directory use
unlink(paste(file.path(system.file(package="agilp"),"output",""),"*.*",sep=""), recursive=FALSE)

## End(Not run)
```

Loader *A file choser utility file.*

Description

Uses a template file (typically an Excel spreadsheet with file names as one column) to select a set of files and either combine them into one file or copy them individually to a new directory. The files to be used must be specified in a contiguous set of rows of the template file.

Usage

```
Loader(input=file.path(system.file(package="agilp"),"input",""),output=file.path(system.file(packag
```

Arguments

input	full path of directory where input data files are put; default is a folder named input within the agilp package directory;
output	full path of directory where output data files are put; default is a folder named output within the agilp package directory;
t	full name and path of template file
f	if f="TRUE", each file is copied individually to output folder; if "f" IS NOT true, all files are merged into one big dataframe , with each original file data as column; this file is called "all_data.txt" and written to output folder; it is also returned as an R dataframe object called dataout
r	The column of template which contains the file names
A	The first row of template to be used; note a header row is NOT ignored.
B	The last row of template to be used

Details

This function is a tool for working with selected files output by AAProcess. Typical usage would be to hold all the array file names with relevant experimental detail and annotation in a file called template. The filenames would then be sorted according to some annoation (e.g. tissue type, date, experimenter etc and then this list is used to pull out the relevant raw data files corresponding to those epxeriments from a big collection of such files. A toy example of a template file is provided in data folder

Value

rawdata files	If f is "TRUE" the function simply copies the files defined by template from input to output folder.
dataout	If f is not = "TRUE" , the individual data files are merged by rows. The row-names are the probe names; each column contains a set of expression data with the column name giving the file name from which the data were derived
all_data	If f is not = "TRUE" the function also writes a tab delimited txt file corresponding to dataout, called all_data.txt to the output directory.

Warning

The function checks that files listed in template exist in folder, and outputs a list of all file names not found

Author(s)

Benny Chain; b.chain@ucl.ac.ucl

References

In preparation

See Also

AAProcess filenamex IDswop Baseline Equaliser AALoess

Examples

```
inputdir<-file.path(system.file(package="agilp"),"extdata","raw/","", fsep = .Platform$file.sep)
outputdir<-file.path(system.file(package="agilp"),"output/", "", fsep = .Platform$file.sep)
template<-file.path(system.file(package="agilp"),"extdata","sample_template.txt", fsep = .Platform$file.sep)

#This will copy the files with names given in column 2 , rows 2:3 of the sample_#template file from input to output.
Loader(input=inputdir,output=outputdir,t=template,f="TRUE",r=2,A=2,B=5)

#Alternatively this will output a single file all_data.txt with the same data in a merged file. The file is also outp
Loader(input=inputdir,output=outputdir,t=template,f="FALSE",r=2,A=2,B=5)

dim(dataout)

## Not run:
#to remove the output files again and empty the output directory use
unlink(paste(file.path(system.file(package="agilp"),"output",""),"*.*",sep=""), recursive=FALSE)

## End(Not run)
```

Index

∗ **data**
AALoess, 2
AAProcess, 3
Baseline, 5
Equaliser, 6
IDswop, 9
Loader, 11
∗ **file**
filenamex, 8
∗ **utilities**
AALoess, 2
Equaliser, 6
IDswop, 9
Loader, 11

AALoess, 2, *4*, *6–8*, *10*, *12*
AAProcess, 3, *3*, *6–8*, *10*, *12*

Baseline, *3, 4*, 5, *7, 8*, *10*, *12*

Equaliser, *3, 4*, 6, *6*, *8*, *10*, *12*

filenamex, *3, 4*, *6, 7*, 8, *10*, *12*

IDswop, *3, 4*, *6–8*, 9, *12*

Loader, *3, 4*, *6–8*, *10*, 11