# Package 'ArrayExpressHTS'

October 9, 2013

**Title** ArrayExpress High Throughput Sequencing Processing Pipeline

**Version** 1.10.0

**Author** Angela Goncalves, Andrew Tikhonov

**Maintainer** Angela Goncalves <filimon@ebi.ac.uk>, Andrew Tikhonov <andrew@ebi.ac.uk>

**Description** RNA-Seq processing pipeline for public ArrayExpress experiments or local datasets

**Depends** sampling, Rsamtools (>= 1.3.32), snow

**Imports** Biobase, Biostrings, DESeq, GenomicRanges, Hmisc, IRanges,R2HTML, RColorBrewer, Rsamtools, ShortRead, XML, biomaRt,edgeR, grDevices, graphics, methods, rJava, stats, svMisc,utils, sendmailR, bitops

**LinkingTo** Rsamtools

**License** Artistic License 2.0

**biocViews** RNAseq, Sequencing, HighThroughputSequencing

## R topics documented:

ArrayExpressHTS *ExpressionSet for RNA-Seq experiment submitted in ArrayExpress and ENA*

## Description

ArrayExpressHTS runs the RNA-Seq pipeline on a transcription profiling experiment available on the ArrayExpress database and produces an ExpressionSet R object. ArrayExpressHTS requires an Internet connection.

## Usage

```
ArrayExpressHTS( accession,
        options = list (
                stranded           = FALSE,
                insize             = NULL,
                insizedev          = NULL,
                reference          = "genome",
                aligner            = "tophat",
                aligner_options    = NULL,
                count_feature      = "transcript",
                count_options      = "",
                count_method       = "cufflinks",
                filter             = TRUE,
                filtering_options  = NULL ),
        usercloud = TRUE,
        rcloudoptions = list (
                nnodes     = "automatic",
                pool       = c("4G", "8G", "16G", "32G", "64G"),
                nretries   = 4 ),
        steplist = c("align", "count", "eset"),
        dir = getwd(),
        refdir = getDefaultReferenceDir(),
        want.reports = TRUE,
        stop.on.warnings = FALSE )
```

## Arguments

| | |
|---|---|
| accession | an ArrayExpress experiment accession identifier, e.g. "E-GEOD-16190" |
| options | defines pipeline options. See getDefaultProcessingOptions. |
| usercloud | defines if the R Cloud will be used to parallel experiment computation, if FALSE, experiment data files will be processed sequentially. |
| rcloudoptions | defines R Cloud options. See getDefaultRCloudOptions. |
| steplist | defines the steps the pipeline will perform. |

| | |
|---|---|
| dir | folder where experiment data will be stored and processed. Default is current directory. |
| refdir | the directory where reference data is located. |
| want.reports | defines if quality reports are produced. Reports usually make computation longer and eat up more memory. For faster computation use FALSE. |
| stop.on.warnings | self explanatory. Warnings are normally producesd when there are inconsistencies, which however would allow the result to be produced. |

## Value

The output is an object of class ExpressionSet containing expression values in assayData (corresponding to the raw sequencing data files), the information contained in the .sdrf file in phenoData, the information in the adf file in featureData and the idf file content in experimentData.

If executed on a local PC, make sure that tools are available to the pipeline. Check prepareAnnotation to see what needs to be done to make tools available.

## Author(s)

Andrew Tikhonov <andrew@ebi.ac.uk>, Angela Goncalves <angela.goncalves@ebi.ac.uk>

## See Also

ArrayExpressHTSFastQ, prepareReference, prepareAnnotation, prepareAnnotation getDefaultProcessingOption getPipelineOptions,

## Examples

```
if (isRCloud()) { # disabled on local configs
    # so as not to affect package building process

    # if executed on a local PC, make sure tools
    # are available to the pipeline.
    expfolder = tempdir();

    # run the pipeline
    #
    aehts = ArrayExpressHTS("E-GEOD-16190", dir = expfolder);

    # load the expression set object
    loadednames = load(paste(expfolder,
                        "/E-GEOD-16190/eset_notstd_rpkm.RData", sep=""));
    loadednames;

    get('library')(Biobase);

    # print out the expression values
    #
    head(assayData(eset)$exprs);
```

```
    # print out the experiment meta data
    experimentData(eset);
    pData(eset);
}
```

---

ArrayExpressHTSFastQ *ExpressionSet for RNA-Seq raw data files*

---

## Description

ArrayExpressHTSFastQ runs the RNA-Seq pipeline on raw RNA-Seq data files and an .sdrf experiment descriptor and produces an ExpressionSet R object.

## Usage

```
ArrayExpressHTSFastQ( accession,
        organism = c("automatic", "Homo_sapiens", "Mus_musculus" ),
        quality = c("automatic", "FastqQuality", "SFastqQuality"),
        options = list(
                stranded          = FALSE,
                insize            = NULL,
                insizedev         = NULL,
                reference         = "genome",
                aligner           = "tophat",
                aligner_options   = NULL,
                count_feature     = "transcript",
                count_options     = "",
                count_method      = "cufflinks",
                filter            = TRUE,
                filtering_options = NULL),
        usercloud = TRUE,
        rcloudoptions = list(
                nnodes        = "automatic",
                pool          = c("4G", "8G", "16G", "32G", "64G"),
                nretries      = 4),
        steplist = c("align", "count", "eset"),
        dir = getwd(),
        refdir = getDefaultReferenceDir(),
        want.reports = TRUE,
        stop.on.warnings = FALSE )
```

## Arguments

| | |
|---|---|
| accession | name of the folder where experiment data is kept and computaton data is stored. The actual data files and descriptor .sdrf and .idf fles should be stored in the 'data' folder, which should be created in this folder. |

| | |
|---|---|
| organism | defines organism filter, "automatic" means all organism found in the descript files will be used. If a specific organism or an array e.g. c("Homo_sapiens", "Mus_musculus") is defined, those organisms will be used to filter out other unwanted organisms and associated data files from the result object. The value canot be "automatic" if no .sdrf is found and no automatic discovery can be performed. |
| quality | this parameter is used to explicitly select the quality scale used in the fastq data files. By default "automatic" is used. In case quality scale cannot be automatically detected, the user will be prompted to increase the detection depth or set the quality scale manually. "FastqQuality" corresponds to Phred+33, "SFastqQuality" corresponds to Phred+64. |
| options | defines pipeline options. See getDefaultProcessingOptions |
| usercloud | defines if the R Cloud will be used to parallel experiment computation. If set to FALSE, the experiment files will be processed sequentially. |
| rcloudoptions | defines R Cloud options. See getDefaultRCloudOptions |
| steplist | defines the steps the pipeline will perform |
| dir | folder where experiment data will be stored and processed. Default is current directory. |
| refdir | the directory where reference data is located. |
| want.reports | defines if quality reports are produced. Reports usually make computation longer and eat up more memory. For faster computation use FALSE. |
| stop.on.warnings | self explanatory. Warnings are normally producesd when there are inconsistencies, which however would allow the result to be produced. |

## Value

The output is an object of class ExpressionSet containing expression values in assayData (corresponding to the raw sequencing data files), the information contained in the .sdrf file in phenoData, the information in the adf file in featureData and the idf file content in experimentData.

## Author(s)

Andrew Tikhonov Angela Goncalves Maintainer: <andrew@ebi.ac.uk> Maintainer: <angela.goncalves@ebi.ac.uk>

## See Also

ArrayExpressHTS, prepareReference, prepareAnnotation, prepareAnnotation getDefaultProcessingOptions, getPipelineOptions,

## Examples

```
if (isRCloud()) { # disabled on local configs
    # so as not to affect package building process

    # In ArrayExpressHTS/expdata there is testExperiment, which is
```

```
    # a very short version of E-GEOD-16190 experiment, placed there
    # for testing reasons.
    #
    # Experiment in ArrayExpress:
    # http://www.ebi.ac.uk/arrayexpress/experiments/E-GEOD-16190
    #
    # the following piece of code will take ~1.5 hours to compute
    # on local PC and ~10 minutes on R-Cloud
    #

    # if executed on a local PC, make sure tools are available
    # to the pipeline.
    #

    # create a temporary folder where experiment will be copied
    # computing experiment in the package folder may cause issues
    # with file permissions and therefore failures.
    #
    #
    srcfolder <- system.file("expdata",
                "testExperiment", package="ArrayExpressHTS");

    dstfolder <- tempdir();

    file.copy(srcfolder, dstfolder, recursive = TRUE);

    # run the pipeline
    #
    # set usercloud = FALSE if executing on local PC,
    # therefore parallel computation will be disabled
    #
    aehts = ArrayExpressHTSFastQ(accession = "testExperiment",
        organism = "Homo_sapiens", dir = dstfolder);

    # load the expression set object
    loadednames = load(paste(dstfolder,
                        "/testExperiment/eset_notstd_rpkm.RData", sep=""));
    loadednames;

    get('library')(Biobase);

    # print out the expression values
    #
    head(assayData(eset)$exprs);

    # print out the experiment meta data
    experimentData(eset);
    pData(eset);
}
```

## isRCloud *Check the code is running on R-Cloud*

### Description

isRCloud returns TRUE/FALSE indicating whether the configuration is an R-Cloud.

### Usage

```
isRCloud()
```

### Value

If TRUE, the configuration is the R-Cloud.

### Author(s)

Andrew Tikhonov <andrew@ebi.ac.uk>, Angela Goncalves <angela.goncalves@ebi.ac.uk>

### See Also

setPipelineOptions, ArrayExpressHTS

### Examples

```
if ( isRCloud() ) {

    # we're on the R-Cloud
    print("R-Cloud configuration");
} else {

    # we're somewhere else
    print("Other configuration");
}
```

## package-options *Functions to work with package level options*

### Description

Functions to work with ArrayExpressHTS package level options.

### Usage

```
getPipelineOptions()
getPipelineOption(name, default = NULL, options = defaultOptions)
setPipelineOptions(...)
```

## Arguments

| | |
|---|---|
| ... | a list ot options to set. |
| name | name of the option to return. In order to see all options, use getPipelineOptions; |
| default | default values in case the option is not defined |
| options | options environment, by default the ArrayExpresHTS environment; |

## Details

These are the basic functions to get and set pakcage options and perform necessary reinitialization following the change in required.

getPipelineOptions returns an environment where ArrayExpressHTS options are stored.

setPipelineOptions sets one or a number of ArrayExpressHTS options.

Please note that the following options are used to store locations of external tools used by the package: "ArrayExpressHTS.bowtie", "ArrayExpressHTS.tophat", "ArrayExpressHTS.bwa", "ArrayExpressHTS.cufflinks", "ArrayExpressHTS.mmseq", "ArrayExpressHTS.samtools", etc.

If any of the options needs to be changed, use setPipelineOption(...) to make necessary chnages. Following the change, PATH environment varibale automatically gets updated as necessary.

Please note that the package supports only certain versions of tools. Other versions may not be fully compatible, and if used, this can result in errors on certain steps of the pipeline. The list of supported versions can be obtained using getPieplineOption(...) with the corresponding tool option, e.g. "ArrayExpressHTS.bowtie". Not supported versions can be used at user's own risk.

getPipelineOption returns ArrayExpressHTS option specified by option name. If option is not defined the specified default value will be returned.

## Value

The output is the environment where pipeline options are stored.

## Author(s)

Andrew Tikhonov <andrew@ebi.ac.uk>, Angela Goncalves <angela.goncalves@ebi.ac.uk>

## See Also

getDefaultProcessingOptions, ArrayExpressHTS, ArrayExpressHTSFastQ

## Examples

```
# get options
pipelineOptions = getPipelineOptions()

# list all names
ls(pipelineOptions)

# set pipeline options
```

```
setPipelineOptions("trace" = "disabled", "memorymonitor" = "disabled");

# get options
getPipelineOption("trace");
getPipelineOption("memorymonitor");
```

---

`prepareAnnotation` *Prepare annotation data for the RNA-Seq Pipeline*

---

## Description

`prepareAnnotation` downloads the required annotation file for the selected organism from Ensembl and processes it so that it can be used by the pipeline. `prepareAnnotation` requires an Internet connection.

## Usage

```
prepareAnnotation(organism, version = "current",
    location = getDefaultReferenceDir(), refresh = FALSE, run = TRUE)
```

## Arguments

| | |
|---|---|
| `organism` | supported organism names can be viewed in the Ensemble database. Check 'ftp://ftp.ensembl.org/pub'. |
| `version` | `"current"` or other appropriate version. Check 'ftp://ftp.ensembl.org/pub'. |
| `location` | indicates where the annotation data should be stored. |
| `refresh` | if TRUE, existing annotation data will be rebuilt. |
| `run` | if FALSE, the commands to obtain and process the annotation will not be executed. |

## Value

The output is the version of the organism annotation that has been downloaded and processed. The annotation files are kept in the folder defined in `location` parameter.

## Author(s)

Andrew Tikhonov <andrew@ebi.ac.uk>, Angela Goncalves <angela.goncalves@ebi.ac.uk>

## See Also

`ArrayExpressHTS`, `ArrayExpressHTSFastQ`, `prepareReference`

## Examples

```
if (isRCloud()) { # disabled on local configs
    # so as not to affect package building process

    par(ask = FALSE)

    # the following piece of code will take ~1.5 hours to complete
    #

    # if executed on a local PC, make sure tools are available
    # to the pipeline.
    #

    # create directory
    #
    # Please note, tempdir() is used for automamtic test
    # execution. Select directory more appropriate and
    # suitable for keeping reference data.
    #
    referencefolder = paste(tempdir(), "/reference", sep = "")

    dir.create(referencefolder)

    # download and prepare annotation
    prepareAnnotation("Homo_sapiens", "current", location = referencefolder)
    prepareAnnotation("Mus_musculus", "NCBIM37.61", location = referencefolder)

}
```

---

# prepareReference *Prepare reference data for the RNA-Seq Pipeline*

---

## Description

prepareReference downloads reference genome or transcriptome for the selected organism from the Ensembl database and processes it so that it can be used by the pipeline. prepareReference requires an Internet connection.

## Usage

```
prepareReference(organism, version = "current",
    type = c("genome", "transcriptome"),
    location = getDefaultReferenceDir(),
    aligner = c("bwa", "bowtie", "tophat"),
    refresh = FALSE, run = TRUE)
```

## Arguments

| | |
|---|---|
| organism | supported organism names can be viewed in the Ensemble database. Check 'ftp://ftp.ensembl.org/pub'. |
| version | "current" or other appropriate version. Check 'ftp://ftp.ensembl.org/pub'. |
| type | two values are supported: "genome", "transcriptome" |
| location | indicates where the reference data should be stored. |
| aligner | 3 types of aligners are supported: "bwa", "bowtie" and "tophat". |
| refresh | if TRUE, existing reference data will be rebuilt. |
| run | if FALSE, the downloading and processing commands will not be executed. |

## Value

The output is the version of the organism reference that has been downloaded and processed. The reference files are kept in the folder defined in location parameter.

## Author(s)

Andrew Tikhonov <andrew@ebi.ac.uk>, Angela Goncalves <angela.goncalves@ebi.ac.uk>

## See Also

ArrayExpressHTS, ArrayExpressHTSFastQ, prepareAnnotation

## Examples

```
if (isRCloud()) {

    par(ask = FALSE)

    # the following piece of code will take ~3 hours to complete
    #

    # if executed on a local PC, make sure tools are available
    # to the pipeline.
    #

    # create directory
    #
    # Please note, tempdir() is used for automamtic test
    # execution. Select directory more appropriate and
    # suitable for keeping reference data.
    #
    referencefolder = paste(tempdir(), "/reference", sep = "")

    dir.create(referencefolder)

    # download and prepare reference
    prepareReference("Homo_sapiens", version = "GRCh37.61",
```

```
        type = "genome", aligner = "bowtie", location = referencefolder )
    prepareReference("Homo_sapiens", version = "GRCh37.61",
        type = "transcriptome", aligner = "bowtie", location = referencefolder )
    prepareReference("Mus_musculus", version = "current",
        type = "genome", aligner = "bowtie", location = referencefolder )
    prepareReference("Mus_musculus", version = "current",
        type = "transcriptome", aligner = "bowtie", location = referencefolder )
}
```

---

`processing-options` *Functions to work with ArrayExpressHTS processing options*

---

## Description

Functions to get default ArrayExpressHTS processing options.

## Usage

```
getDefaultFilteringOptions()
getDefaultProcessingOptions()
getDefaultRCloudOptions()
```

## Arguments

No arguments.

## Value

These functions return default options used for ArrayExpressHTS processing.

Function `getDefaultFilteringOptions` returns a `list` of filtering options. These options control filtering of BAM files after alignemnt.

| | |
|---|---|
| `mismatches` | `2` by default, number of mismatches |
| `chr_ignore` | `c("MT")` by default, reference names to ignore |
| `minqual` | `10` by default |
| `minmapq` | `1` by default |
| `maxN` | `2` by default |
| `maxpol` | `0.75` by default, ither a proportion (<0) or an integer |
| `duplicates` | `"remove"` by default, supported value: `"keep"`, `"remove"` |
| `multihits` | `"remove"` by default, supported value: `"keep"`, `"remove"` |
| `gapped` | `"remove"` by default, supported value: `"keep"`, `"remove"` |

—

Function `getDefaultProcessingOptions` returns a `list` of options used throught the processing. These options control alignment and counting.

These options include:

| | |
|---|---|
| stranded | FALSE by default, set to TRUE if a strand specific protocol was used |
| insize | NULL by default, the insert size, an integer, which will be automatically determined if set to NULL |
| insizedev | NULL by default, insert size deviation, an integer, which will be automatically determined if set to NULL |
| reference | "genome" by default, controlls whether alignment is performed using "genome" or "transcriptome" reference. |
| aligner | "tophat" by default, supported values are: "tophat", "bowtie", "bwa" or "custom" |
| aligner_options | NULL by default, string of options to be passed to the aligner according to manual. |
| count_feature | "transcript" by default, count over "gene" or "transcript" |
| count_method | "cufflinks" by default, supported: "cufflinks", "mmseq" or "count" |
| count_options | options for count software |
| standardise | FALSE by default, supported: TRUE, FALSE |
| normalisation | "rpkm" by default, supported: "none" or "rpkm" |
| filter | "TRUE" by defult, supported: "TRUE", "FALSE" |
| filtering_options | these are the default filtering options, see above. |

—

Function getDefaultRCloudOptions returns a set of R Cloud options related to R Cloud resources and execution flow.

These include:

| | |
|---|---|
| nnodes | "automatic" by default, supported: "automatic" or a numeric value, e.g. 10 |
| pool | "32G" by default, supported: "4G", "8G", "16G", "32G", "64G" |
| nretries | 4 by default, a numberic value. |

## Author(s)

Andrew Tikhonov <andrew@ebi.ac.uk>, Angela Goncalves <angela.goncalves@ebi.ac.uk>

## See Also

getPipelineOptions, ArrayExpressHTS, ArrayExpressHTSFastQ

## Examples

```
# get filtering options
getDefaultFilteringOptions()

# get alignment and counting options
getDefaultProcessingOptions()
```

```
# get R Cloud options
getDefaultRCloudOptions()
```

# Index