

# Package ‘Risa’

February 24, 2024

**Version** 1.44.0

**Date** 2013-08-15

**Title** Converting experimental metadata from ISA-tab into Bioconductor data structures

**Author**

Alejandra Gonzalez-Beltran, Audrey Kauffmann, Steffen Neumann, Gabriella Rustici, ISA Team

**Maintainer** Alejandra Gonzalez-Beltran <alejandra.gonzalez.beltran@gmail.com>

**Depends** R (>= 2.0.9), Biobase (>= 2.4.0), methods, Rcpp (>= 0.9.13), biocViews, affy

**Imports** xcms

**Suggests** faahKO (>= 1.2.11)

**Description** The Investigation / Study / Assay (ISA) tab-delimited format is a general purpose framework with which to collect and communicate complex metadata (i.e. sample characteristics, technologies used, type of measurements made) from experiments employing a combination of technologies, spanning from traditional approaches to high-throughput techniques. Risa allows to access metadata/data in ISA-Tab format and build Bioconductor data structures. Currently, data generated from microarray, flow cytometry and metabolomics-based (i.e. mass spectrometry) assays are supported. The package is extendable and efforts are undergoing to support metadata associated to proteomics assays.

**License** LGPL

**biocViews** Annotation, DataImport, MassSpectrometry

**URL** <http://www.isa-tools.org/>

**BugReports** <https://github.com/ISA-tools/Risa/issues>

**git\_url** <https://git.bioconductor.org/packages/Risa>

**git\_branch** RELEASE\_3\_18

**git\_last\_commit** 447ce5a

**git\_last\_commit\_date** 2023-10-24

**Repository** Bioconductor 3.18

**Date/Publication** 2024-02-23

## R topics documented:

AssayTab-class	2
getAnnotatedDataFrameAssay	3
getAssayRawDataFileNames-methods	4
getExpressionSet	5
getMIAMEMetadata	6
getMicroarrayAssayFileNames	7
getMSAssayFileNames	7
getRawDataFileNames-methods	8
getStudyFilename	8
ISATab-class	9
isMicroarrayAssay	11
isMSAssay	12
processAssayXcmsSet	13
processAssayXcmsSet.l factor	14
readISATab	15
Risa	16
suggestBiocPackage	17
updateAssayMetadata	17
write.assay.file	18
write.investigation.file	19
write.ISATab	20
write.study.file	20
<b>Index</b>	<b>22</b>

---

AssayTab-class	<i>Class "AssayTab", a class for maintaing the information of AssayTab</i>
----------------	--

---

### Description

This class maintains information from an AssayTab dataset, including an investigation file, one or more study files, and one or more assay files for each study file.

### Objects from the Class

Objects can be created by calls of the form `new("AssayTab", ...)`.

### Slots

`path` Object of class "character", it represents the path of the ISA-Tab dataset to which this AssayTab belongs

`study.filename` Object of class "character" with the filename of the study to which this assay belongs to.

`study.identifier` Object of class "character" with the identifier of the study to which this assay belongs to.

assay.filename Object of class "character" with the filename of the assay.  
assay.file Object of class "data.frame" with the contents of the assay file.  
assay.technology.type Object of class "character" with the technology type of this assay.  
assay.measurement.type Object of class "character" with the measurement type of this assay.  
assay.names Object of class "data.frame" with the assay names for this assay file.  
data.fileNames Object of class "data.frame" with the data filenames.

## Methods

[ signature(x = "AssayTab"): ...

**getAssayRawDataFileNames** signature(.Object = "AssayTab", full.path="logical"): returns the raw data filenames for this assay

## Author(s)

Alejandra Gonzalez-Beltran, <isatools@googlegroups.com>

## See Also

ISATab-class, ISATab-methods,

## Examples

```
showClass("AssayTab")
showClass("MSAssayTab")
showClass("MicroarrayAssayTab")
showClass("SeqAssayTab")
showClass("NMRAssayTab")
```

---

getAnnotatedDataFrameAssay

*Retrieves an AnnotatedDataFrameAssay.*

---

## Description

Retrieves an AnnotatedDataFrameAssay.

## Usage

```
getAnnotatedDataFrameAssay(isa, assay.filename)
```

## Arguments

isa An object of the ISATab-class.  
assay.filename

**Value**

An annotated data frame.

**Author(s)**

Alejandra Gonzalez-Beltran

**See Also**

ISATab-class, AnnotatedDataFrame

**Examples**

```
faahkoISAZip <- system.file("extdata","faahKO-metadata.zip", package="Risa")
faahkoISA <- readISATab(zipfile = faahkoISAZip, path = file.path(system.file("extdata", package="Risa"),"faahkoI
faahkoADF <- getAnnotatedDataFrameAssay(faahkoISA, faahkoISA@assay.files[[1]])
str(faahkoADF)
```

```
armstrongISAZip <- system.file("extdata","ARMSTRONG-S-3-metadata.zip", package="Risa")
armstrongISA <- readISATab(zipfile = armstrongISAZip, path = file.path(system.file("extdata", package="Risa"),"a
armstrongADF <- getAnnotatedDataFrameAssay(armstrongISA, armstrongISA@assay.files[[1]])
str(armstrongADF)
```

---

getAssayRawDataFileNames-methods

*~~ Methods for Function getAssayRawDataFileNames in Package*

**Risa** ~~

---

**Description**

~~ Methods for function getAssayRawDataFileNames in package **Risa** ~~ They retrieve the raw data files for a particular assay type.

**Methods**

signature(.Object = "AssayTab", full.path = "logical") Retrieves the raw data filenames for a generic assay.

signature(.Object = "MicroarrayAssayTab", full.path = "logical") Retrieves the raw data filenames for an assay whose technology type is DNA microarray.

signature(.Object = "MSAssayTab", full.path = "logical") Retrieves the raw data filenames for an assay whose technology type is mass spectrometry. These data filenames correspond to those specified in the column 'Raw Spectral Data File'.

signature(.Object = "NMRAssayTab", full.path = "logical") Retrieves the raw data filenames for an assay whose technology type is NMR spectroscopy.

signature(.Object = "SeqAssayTab", full.path = "logical") Retrieves the raw data filenames for an assay whose technology type is nucleotide sequencing.

---

getExpressionSet	<i>Retrieves an ExpressionSet for a DNA microarray assay.</i>
------------------	---

---

### Description

Retrieves an ExpressionSet for a DNA microarray assay.

### Usage

```
getExpressionSet(isa, assay.filename)
```

### Arguments

`isa` An object from the ISATab-class.  
`assay.filename` A character vector with the assay filename of the microarray assay.

### Value

It returns an ExpressionSet.

### Author(s)

Alejandra Gonzalez-Beltran

### References

affy

### See Also

[ISATab-class](#)

### Examples

```
data.dir <- system.file("extdata", package="Risa")
isazip <- system.file("extdata", "ARMSTRONG-S-3-metadata.zip", package="Risa")
path <- file.path(data.dir, "ARMSTRONG_ISA")
isa <- readISATab(zipfile = isazip, path = path, verbose = TRUE)
path <- file.path(data.dir, "ARMSTRONG_ISA/")

cel.files <- c("GSM510377_SA2006121413.CEL", "GSM510378_SA2006121414.CEL", "GSM510379_SA2006121415.CEL", "GSM510380_SA2006121416.CEL", "GSM510381_SA2006121417.CEL", "GSM510382_SA2006121418.CEL", "GSM510383_SA2006121419.CEL", "GSM510384_SA2006121420.CEL", "GSM510385_SA2006121421.CEL", "GSM510386_SA2006121422.CEL", "GSM510387_SA2006121423.CEL", "GSM510388_SA2006121424.CEL", "GSM510389_SA2006121440.CEL", "GSM510390_SA2006121441.CEL", "GSM510391_SA2006121442.CEL")

base.url <- "http://perdera.sph.harvard.edu/biidata/microarray/study_ARMSTRONG-S-3_3Zd4KeJNJ7/raw_data/"

#####download all the cel files - this is commented out, uncomment the following two lines if you want to build a list of files
#lapply(cel.files, function(x) download.file(paste(base.url, x, sep=""), destfile=paste(path, x, sep="")))

```

```
#getExpressionSet(isa, isa@assay.filenamees[[1]])
```

---

getMIAMEMetadata	<i>Retrieves the MIAME information about a microarray experiment</i>
------------------	--

---

### Description

If the `assay.filename` uses the microarray technology, it retrieves an object of the class `MIAME` with the minimum information about the experiment. Otherwise, it outputs a message indicating that the method is not valid for assays not using the microarray technology.

### Usage

```
getMIAMEMetadata(isa, assay.filename)
```

### Arguments

<code>isa</code>	An object of the <code>ISAtab</code> -class.
<code>assay.filename</code>	A string indicating the name of an assay file. It must correspond to a microarray-based assay for the function to return a valid value.

### Value

It retrieves an object from the `MIAME` class or an error message.

### Author(s)

Alejandra Gonzalez-Beltran

### See Also

`isMicroarrayAssay`

### Examples

```
armstrongISAzip <- system.file("extdata", "ARMSTRONG-S-3-metadata.zip", package="Risa")
armstrongISA <- readISAtab(zipfile = armstrongISAzip, path = file.path(system.file("extdata", package="Risa"), "a
assay.filename <- armstrongISA@assay.filenamees[[1]]
miame <- getMIAMEMetadata(armstrongISA, assay.filename)
str(miame)
```

---

`getMicroarrayAssayFileNames`*Listing filenames of microarray-based assy from an ISAtab dataset.*

---

**Description**

Retrieves the list of assy filenames in an ISAtab dataset, which correspond to microarray-based assays.

**Usage**

```
getMicroarrayAssayFileNames(isa)
```

**Arguments**

`isa` An object of the ISAtab-class.

**Value**

a list of assay filenames

**Author(s)**

Alejandra Gonzalez-Beltran

**See Also**

`isMicroarrayAssay`

**Examples**

```
armstrongISAzipped <- system.file("extdata", "ARMSTRONG-S-3-metadata.zip", package="Risa")
armstrongISA <- readISAtab(zipfile = armstrongISAzipped, path = file.path(system.file("extdata", package="Risa"), "armstrongISA"))
getMicroarrayAssayFileNames(armstrongISA)
```

---

`getMSAssayFileNames` *Retrieves a vector with the assay filenames corresponding to mass spectrometry.*

---

**Description**

Given an object from the ISAtab-class, this method retrieves a vector with the assay filenames corresponding to mass spectrometry (identified by having a column called 'Raw Spectral Data File').

**Usage**

```
getMSAssayFileNames(isa)
```

**Arguments**

isa                    an object from the ISATab class

**Value**

A character vector with the assay filenames corresponding to mass spectrometry.

**Author(s)**

Alejandra Gonzalez-Beltran

**Examples**

```
data.dir <- system.file("extdata", package="Risa")
isazip <- system.file("extdata","faahKO-metadata.zip", package="Risa")
faahkoISA <- readISATab(zipfile = isazip, path = file.path(data.dir,"faahKOISA"), verbose =TRUE)
ms.assay.filenames <- getMSAssayFilenames(faahkoISA)
```

---

getRawDataFilenames-methods

~~ *Methods for Function getRawDataFilenames in Package **Risa*** ~~

---

**Description**

~~ Methods for function getRawDataFilenames in package **Risa** ~~\ To retrieve the raw data filenames in the ISATab dataset.

**Methods:**

signature(.Object = "ISATab", full.path = "logical") Retrieves the raw data filenames in the ISATab dataset given an object from the ISATab-class and a boolean value indicating if the filenames are retrieved with their full path or not.

---

getStudyFilename

*Retrieves the study filename to which an assay.filename corresponds.*

---

**Description**

Retrieves the study filename to which an assay.filename corresponds.

**Usage**

```
getStudyFilename(isa, assay.filename)
```



**Arguments**

`isa` an object from the ISATab-class.  
`assay.filename` a string with the filename of an assay

**Value**

a string with the study filename.

**Author(s)**

Alejandra Gonzalez-Beltran

**See Also**

ISATab-class

**Examples**

```
faahkoISA <- readISATab(find.package("faahKO"))
getStudyFilename(faahkoISA, faahkoISA@assay.files[[1]])
```

---

ISATab-class

*Class "ISATab", a class for maintaing ISATab information*

---

**Description**

This class maintains information from an ISATab dataset, including an investigation file, one or more study files, and one or more assay files for each study file.

**Objects from the Class**

Objects can be created by calls of the form `new("ISATab", ...)`.

**Slots**

`path`: Object of class "character", it represents the path of the ISA-Tab dataset  
`investigation.filename`: Object of class "character", it represents the name of the Investigation file  
`investigation.file`: Object of class "data.frame", it is a data frame with the contents of the Investigation file  
`investigation.identifier`: Object of class "character", it is a string with the identifier for the investigation  
`study.identifiers`: Object of class "character", it is the list of study identifiers  
`study.titles`: Object of class "character", it is the list of study titles  
`study.descriptions`: Object of class "character", it is the list of study descriptions

`study.contacts`: Object of class "character", it is the list of study contacts

`study.contacts.affiliations`: Object of class "character", it is the list of affiliations of the study contacts

`study.fileNames`: Object of class "character", the names of the study files

`study.files`: Object of class "list", a list of data frames with the contents of the study files

`assay.fileNames`: Object of class "character", the names of the assay files

`assay.fileNames.per.study`: Object of class "list", the names of the assay files according to the study they belong to

`assay.files`: Object of class "list", a list of data frames with the contents of the assay files

`assay.files.per.study`: Object of class "list", a list of data frames with the contents of the assay files divided per study they belong to

`assay.names`: Object of class "list", which is a list of data frames, containing, per each assay file, the column of the assay file corresponding to the "Assay Name"

`assay.technology.types`: Object of class "character", a list with the technology types corresponding to each assay

`assay.measurement.types`: Object of class "character", a list with the names of the data files

`data.fileNames`: Object of class "list", a list with the names of the samples

`samples`: Object of class "character"

`samples.per.study`: Object of class "list",

`samples.per.assay.filename`: Object of class "list", the samples classified according to the assay filename they belong to

`assay.fileNames.per.sample`: Object of class "list", where each element is a list of assay file names per each sample name

`sample.to.rawdatafile`: Object of class "list", a list of data frames with samples associated with raw data files

`sample.to.assayname`: Object of class "list", maintains the association between samples and assay names

`rawdatafile.to.sample`: Object of class "list", maintains the association raw data file and samples

`assayname.to.sample`: Object of class "list", maintains the associations between assay names and samples

`factors`: Object of class "list", which contains R 'factor' objects for each of the studies (with factor names and factor levels).

`treatments`: Object of class "list" of data frames with the combination of factor values, per study file, or a factor object, if there is a single study

`groups`: Object of class "list", with one element per study file, and each element is a list of the samples for the corresponding treatment

`assay.tabs`: Object of class "list", where each element is an object of the class AssayTab or its specialisations (MSAssayTab or MicroarrayAssayTab), representing the information of each of the assay files in the investigation.

## Methods

```
[ signature(x = "ISATab"): ...
```

```
getRawDataFileNames signature(.Object="ISATab", full.path=TRUE): get the raw data file-  
names for all the assays in the ISATab dataset
```

## Note

No notes yet.

## Author(s)

Alejandra Gonzalez-Beltran, <isatools@googlegroups.com>

## References

Risa is a component of the ISA software infrastructure <http://isa-tools.org/>. The source code and latest version can be found in the GitHub repository <https://github.com/ISA-tools/Risa>. Please, submit all 'bugs' and feature requests through <https://github.com/ISA-tools/Risa/issues>.

## See Also

ISATab-methods

## Examples

```
showClass("ISATab")
```

---

isMicroarrayAssay	<i>Determines if an assay is microarray-based or not.</i>
-------------------	---

---

## Description

Retrieves a boolean value indicating whether the assay filename given as parameter corresponds to a microarray-based assay or not.

## Usage

```
isMicroarrayAssay(isa, assay.filename)
```

## Arguments

isa	An object of the ISATab-class.
assay.filename	A string indicating the name of an assay file. It must correspond to a microarray-based assay for the function to return a valid value.

**Value**

Retrieves a boolean value.

**Author(s)**

Alejandra Gonzalez-Beltran

**See Also**

getMIAMEMetadata

**Examples**

```
data.dir <- system.file("extdata", package="Risa")
isazip <- system.file("extdata", "ARMSTRONG-S-3-metadata.zip", package="Risa")
isa <- readISAtab(zipfile = isazip, path = file.path(data.dir, "ARMSTRONG_ISA"), verbose =TRUE)

assay.filename <- isa@assay.files[[1]]
isMicroarrayAssay(isa, assay.filename)
```

---

isMSAssay

*Indicates if an assay filename corresponds to a mass spectrometry assay.*

---

**Description**

Indicates if an assay filename corresponds to a mass spectrometry assay.

**Usage**

```
isMSAssay(isa, assay.filename)
```

**Arguments**

`isa` an object from the ISAtab-class.  
`assay.filename` the filename of an assay file

**Value**

It retrieves a boolean value indicating whether the assay is a mass spectrometry assay or not.

**Author(s)**

Alejandra Gonzalez-Beltran

**Examples**

```
faahkoISA = readISAtab(find.package("faahKO"))
assay.filename <- faahkoISA["assay.files"][[1]]
isMSAssay(faahkoISA, assay.filename)
```

---

processAssayXcmsSet     *Build an xcmsSet object given a mass spectrometry assay and considering all factors defined*

---

## Description

processAssayXcmsSet retrieves an xcmsSet object given an ISA-tab object and one of its assay filenames.

## Usage

```
processAssayXcmsSet(isa, assay.filename, ...)
```

## Arguments

isa	an ISA object, as retrieved by the function readISAtab
assay.filename	a boolean indicating to show messages for the different steps, if TRUE, or not to show them, if FALSE
...	extra arguments that can be passed down to the xcmsSet function from the xcms package

## Value

the xcmsSet object built from the assay metadata, considering all the factors defined, and data files

## Author(s)

Steffen Neumann, Alejandra Gonzalez-Beltran / Maintainer: Alejandra Gonzalez-Beltran <isatools@googlegroups.com>

## See Also

readISAtab

## Examples

```
faahkoISA = readISAtab(find.package("faahKO"))
assay.filename <- faahkoISA["assay.filenames"][[1]]
xset = processAssayXcmsSet(faahkoISA, assay.filename)
```

processAssayXcmsSet.1factor

*Build an xcmsSet object given a mass spectrometry assay and considering the first factor defined*

---

## Description

processAssayXcmsSet.1factor retrieves an xcmsSet object given an ISA-tab object and one of its assay filenames.

## Usage

```
processAssayXcmsSet.1factor(isa, assay.filename, ...)
```

## Arguments

isa	an ISA object, as retrieved by the function readISAtab
assay.filename	a boolean indicating to show messages for the different steps, if TRUE, or not to show them, if FALSE
...	extra arguments that can be passed down to the xcmsSet function from the xcms package

## Value

the xcmsSet object built from the assay metadata, considering all the factors defined, and data files

## Author(s)

Alejandra Gonzalez-Beltran / Maintainer: Alejandra Gonzalez-Beltran <isatools@googlegroups.com>

## See Also

readISAtab

## Examples

```
faahkoISA = readISAtab(find.package("faahKO"))
assay.filename <- faahkoISA["assay.filenames"][[1]]
xset = processAssayXcmsSet.1factor(faahkoISA, assay.filename)
```

---

readISAtab	<i>Reads an ISAtab dataset into an R object.</i>
------------	--

---

### Description

Reads an ISAtab dataset given as a zip file or as a set of files in a specific folder, and builds an object from the ISAtab class.

### Usage

```
readISAtab(path = getwd(), zipfile = NULL, verbose = FALSE)
```

### Arguments

path	the name of the directory in which the ISAtab files are located (if the parameter zipfile is not provided or if it is equal to NULL), or the name of the directory where the zip file will be extracted (if the parameter zipfile is not NULL.) The default is the working directory.
zipfile	a zip archive containing ISAtab files. The default value is NULL.
verbose	a boolean indicating to show messages for the different steps, if TRUE, or not to show them, if FALSE

### Value

The result is an object from the ISAtab class.

### Author(s)

Alejandra Gonzalez-Beltran, Audrey Kauffmann / Maintainer: Alejandra Gonzalez-Beltran, ISA Team <isatools@googlegroups.com>

### Examples

```
## Example for mass spectrometry dataset
temp = tempdir()
datafiles = c(file.path(system.file("cdf/K0", package="faahK0"), grep("CDF",dir(system.file("cdf/K0", package="faahK0"), recursive=TRUE))),
file.path(system.file("txt",dir(system.file("cdf/K0", package="faahK0"), recursive=TRUE))),
file.path(system.file("txt",dir(system.file("cdf/K0", package="faahK0"), recursive=TRUE)))
isafiles = file.path(system.file("cdf/K0", package="faahK0"), grep("CDF",dir(system.file("cdf/K0", package="faahK0"), recursive=TRUE)))
file.copy(isafiles, temp, recursive=TRUE)
isaObject1 = readISAtab(path = temp)

## Example of readISAtab for a mass spectrometry experiment
isazip <- system.file("extdata", "faahK0-metadata.zip", package="Risa")
isaObject2 <- readISAtab(zipfile = isazip, path = file.path(system.file("extdata", package="Risa"), "ISAexample"))
```

## Description

The Risa package builds R objects from ISA-tab format files and it is a component of the ISA software infrastructure (see <http://isa-tools.org/>). ISA stands for Investigation, Study, Assay and the infrastructure revolves around the ISA-Tab format, which supports descriptions of biological experiments, with focus but not limited to high-throughput experiments. The format is supported by a series of open-source tools (all available in [There is a growing international user community of the ISA infrastructure, featured in the ISA commons website \(<http://isacommons.org>\)](http://isa-tools.org/)).

The source code and latest version of the Risa package can be found in the GitHub repository <https://github.com/ISA-tools/Risa>. Please, submit all 'bugs' and feature requests through <https://github.com/ISA-tools/Risa/issues>.

The function `readISAtab` reads ISA-tab files from a local zip archive or from a local folder.

The package also contains methods to save the ISA-tab files and to update the metadata after analysis. For mass spectrometry assays, the package includes several methods to build necessary object for processing using the `xcms` package (available in Bioconductor).

There are also functions to deal with DNA microarray-based assays, such as building an `ExpressionSet` object, relying on the `affy` package (available in Bioconductor)

In addition, there is a function called `suggestBiocPackage` that, given an object from the `ISAtab`-class - i.e. an `ISAtab` dataset - it will retrieve a list of potentially useful packages from Bioconductor, according to the assays measurement and technology types. This functionality is based on the annotations provided by the `BiocViews`.

The Risa package bridges the ISA metadata with analysis pipelines in R, and this is the reason of its name: R + ISA = Risa. Anecdotically, we note that Risa is the Pleasure Planet in Star Trek (see <http://en.memory-alpha.org/wiki/Risa>) and that 'Risa' means laughter in Spanish.

## Author(s)

Alejandra Gonzalez-Beltran, Steffen Neumann, Audrey Kauffmann, Gabriella Rustici, Philippe Rocca-Serra, Eamonn Maguire, Susanna-Asunta Sansone. Maintainer: Alejandra Gonzalez-Beltran <[isatools@googlegroups.com](mailto:isatools@googlegroups.com)>

## See Also

`readISAtab`, `processAssayXcmsSet`, `updateAssayMetadata`, `write.ISAtab`

## Examples

```
## Example of readISAtab for a mass spectrometry experiment from a zip file
isazip = system.file("extdata", "faahK0-metadata.zip", package="Risa")
isaset = readISAtab(zipfile = isazip, path = file.path(system.file("extdata", package="Risa"), "ISAexample"), v

require("faahK0")
## Example of readISAtab for mass spectrometry data
```



```
temp = tempdir()
datafiles = c(file.path(system.file("cdf/K0", package="faahK0"), grep("CDF",dir(system.file("cdf/K0", package="faahK0"), recursive=TRUE))),
file.path(system.file("cdf/K0", package="faahK0"), grep("CDF",dir(system.file("cdf/K0", package="faahK0"), recursive=TRUE)))
file.copy(datafiles, temp, recursive=TRUE)
isafiles = file.path(system.file(package="faahK0"), grep("txt",dir(system.file(package="faahK0"), recursive=TRUE)), value=TRUE)
file.copy(isafiles, temp, recursive=TRUE)
isatab.object = readISAtab(path = temp)
```

---

suggestBiocPackage      *Suggest Bioconductor packages*

---

### Description

Suggest a list of packages available in Bioconductor, which may be useful for processing the assay according to its technology and measurement types.

### Usage

```
suggestBiocPackage(isa, bioc.version)
```

### Arguments

isa	An ISAtab-class object.
bioc.version	The version number of Bioconductor to be considered as the source of the packages for suggestions.

### Author(s)

Alejandra Gonzalez-Beltran

---

updateAssayMetadata      *Update metadata into the assay file*

---

### Description

updateAssayMetadata Updates metadata into a particular assay file. See an example of use for a metabolite dataset at <https://github.com/sneumann/mtbls2>.

### Usage

```
updateAssayMetadata(isa, assay.filename, col.name, values)
```

**Arguments**

isa	An isatab object, as retrieved by the readISAtab function.
assay.filename	the filename of the assay file to be augmented/modified
col.name	the name of the column of the assay file to be modified
values	the values to be added to the column of the assay file: it could be a single value, and in this case the value is repeated across the column, or it could be a list of values (whose length must match the number of rows of the assay file)

**Value**

The updated ISA-Tab object.

**Author(s)**

Alejandra Gonzalez-Beltran / Maintainer: Alejandra Gonzalez-Beltran, ISA Team <isatools@googlegroups.com>

**See Also**

readISAtab, <https://github.com/sneumann/mtbls2>

**Examples**

```
### This example shows how to add values to the column "Derived Spectral Data File", assuming that the results are s
faahkoISA = readISAtab(find.package("faahKO"))
assay.filename <- faahkoISA["assay.fileNames"][[1]]
updateAssayMetadata(faahkoISA, assay.filename, "Derived Spectral Data File", "faahkoDSDF.txt" )
```

---

write.assay.file	<i>Write an Assay File from the ISA-Tab dataset</i>
------------------	---

---

**Description**

write.assay.file writes a specific assay file from the ISA-tab dataset into disk.

**Usage**

```
write.assay.file(isa, assay.filename, path = getwd())
```

**Arguments**

isa	the ISA-Tab object (as retrieved by the <a href="#">readISAtab</a> function )
assay.filename	the name of the assay file to be written to disk
path	the output path in which the study file is going to be written, by default is the working directory

**Author(s)**

Alejandra Gonzalez-Beltran / Maintainer: Alejandra Gonzalez-Beltran, ISA Team <isatools@googlegroups.com>

**See Also**

write.ISAtab, write.investigation.file, write.study.file, readISAtab

**Examples**

```
### Example of writing the study file of faahKO ISA-Tab dataset into the temp directory
faahkoISA <- readISAtab(find.package("faahKO"))
temp = tempdir()
write.assay.file(faahkoISA, faahkoISA["assay.filenames"][[1]], temp)
```

---

write.investigation.file

*Writes the Investigation File of an ISA-Tab dataset*

---

**Description**

write.investigation.file reads a ISA-tab file.

**Usage**

```
write.investigation.file(isa, path = getwd())
```

**Arguments**

isa	the ISA-Tab object (as retrieved by the <a href="#">readISAtab</a> function)
path	the output path in which the study file is going to be written, by default is the working directory

**Author(s)**

Alejandra Gonzalez-Beltran / Maintainer: Alejandra Gonzalez-Beltran, ISA Team <isatools@googlegroups.com>

**See Also**

write.ISAtab, write.study.file, write.assay.file, readISAtab

**Examples**

```
### Example of writing the study file of faahKO ISA-Tab dataset into the temp director
faahkoISA <- readISAtab(find.package("faahKO"))
temp = tempdir()
write.investigation.file(faahkoISA, temp)
```

---

write.ISAtab	<i>Write an ISA-Tab dataset</i>
--------------	---------------------------------

---

**Description**

write.ISAtab writes a ISA-Tab dataset.

**Usage**

```
write.ISAtab(isa, path = getwd())
```

**Arguments**

isa	the ISA-Tab object as built with the function readISAtab.
path	the directory where the files are going to be written, by default is the working directory

**Author(s)**

Alejandra Gonzalez-Beltran / Maintainer: Alejandra Gonzalez-Beltran, ISA Team <isatools@googlegroups.com>

**See Also**

write.investigation.file, write.study.file, write.assay.file, readISAtab

**Examples**

```
### Example of writing the study file of faahKO ISA-Tab dataset into the temp director
data.dir <- system.file("extdata", package="Risa")
isazip <- system.file("extdata", "faahKO-metadata.zip", package="Risa")
faahkoISA <- readISAtab(zipfile = isazip, path = file.path(data.dir, "faahKOISA"), verbose =TRUE)
temp = tempdir()
write.ISAtab(faahkoISA, temp)
```

---

write.study.file	<i>Write a Study File of an ISA-Tab dataset</i>
------------------	---

---

**Description**

write.study.file writes a specific Study file from the ISA-tab dataset.

**Usage**

```
write.study.file(isa, study.filename, path = getwd())
```

**Arguments**

`isa` the ISA-Tab object (as retrieved by the [readISAtab](#) function)  
`study.filename` the name of the study file to be written to disk  
`path` the output path in which the study file is going to be written, by default is the working directory

**Author(s)**

Alejandra Gonzalez-Beltran / Maintainer: Alejandra Gonzalez-Beltran, ISA Team <isatools@googlegroups.com>

**See Also**

`write.study.file`, `write.ISAtab`, `readISAtab`

**Examples**

```
### Example of writing the study file of faahKO ISA-Tab dataset into the temp director
faahkoISA <- readISAtab(find.package("faahKO"))
temp = tempdir()
write.study.file(faahkoISA, faahkoISA["study.fileNames"][[1]], temp)
```

# Index

- \* **ISAtab datasets**
  - readISAtab, [15](#)
- \* **affy**
  - getExpressionSet, [5](#)
- \* **classes**
  - AssayTab-class, [2](#)
  - ISAtab-class, [9](#)
- \* **methods**
  - getAssayRawDataFileNames-methods, [4](#)
  - getRawDataFileNames-methods, [8](#)
- \* **microarray**
  - getExpressionSet, [5](#)
- \* **raw data filenames**
  - getAssayRawDataFileNames-methods, [4](#)
- [, AssayTab, ANY, ANY, ANY-method (AssayTab-class), [2](#)
- [, AssayTab-method (AssayTab-class), [2](#)
- [, ISAtab, ANY, ANY, ANY-method (ISAtab-class), [9](#)
- [, ISAtab-method (ISAtab-class), [9](#)
- [<-, AssayTab, ANY, ANY, ANY-method (AssayTab-class), [2](#)
- [<-, AssayTab-method (AssayTab-class), [2](#)
- [<-, ISAtab, ANY, ANY, ANY-method (ISAtab-class), [9](#)
- [<-, ISAtab-method (ISAtab-class), [9](#)
- AssayTab-class, [2](#)
- getAnnotatedDataFrameAssay, [3](#)
- getAssayRawDataFileNames (getAssayRawDataFileNames-methods), [4](#)
- getAssayRawDataFileNames, AssayTab, logical-method (getAssayRawDataFileNames-methods), [4](#)
- getAssayRawDataFileNames, MicroarrayAssayTab, logical-method (getAssayRawDataFileNames-methods), [4](#)
- getAssayRawDataFileNames, MSAssayTab, logical-method (getAssayRawDataFileNames-methods), [4](#)
- getAssayRawDataFileNames, NMRAssayTab, logical-method (getAssayRawDataFileNames-methods), [4](#)
- getAssayRawDataFileNames, SeqAssayTab, logical-method (getAssayRawDataFileNames-methods), [4](#)
- getAssayRawDataFileNames-methods, [4](#)
- getExpressionSet, [5](#)
- getMIAMEMetadata, [6](#)
- getMicroarrayAssayFileNames, [7](#)
- getMSAssayFileNames, [7](#)
- getRawDataFileNames (getRawDataFileNames-methods), [8](#)
- getRawDataFileNames, ISAtab, logical-method (getRawDataFileNames-methods), [8](#)
- getRawDataFileNames-methods, [8](#)
- getStudyFilename, [8](#)
- ISAtab-class, [9](#)
- isMicroarrayAssay, [11](#)
- isMSAssay, [12](#)
- MicroarrayAssayTab-class (AssayTab-class), [2](#)
- MSAssayTab-class (AssayTab-class), [2](#)
- NMRAssayTab (AssayTab-class), [2](#)
- processAssayXcmsSet, [13](#)
- processAssayXcmsSet.1factor, [14](#)
- readISAtab, [15](#), [18](#), [19](#), [21](#)
- Risa-package (Risa), [16](#)

SeqAssayTab (AssayTab-class), [2](#)  
suggestBiocPackage, [17](#)

updateAssayMetadata, [17](#)

write.assay.file, [18](#)

write.investigation.file, [19](#)

write.ISAtab, [20](#)

write.study.file, [20](#)