

# Package ‘readat’

July 5, 2019

**Title** Functionality to Read and Manipulate SomaLogic ADAT files

**Version** 1.10.0

**Maintainer** Richard Cotton <richierocks@gmail.com>

**Description** This package contains functionality to import, transform and annotate data from ADAT files generated by the SomaLogic SOMAscan platform.

**URL** <https://bitbucket.org/graumannlabtools/readat>

**BugReports** <https://bitbucket.org/graumannlabtools/readat/issues>

**Depends** R (>= 3.4.0)

**Imports** assertive.base (>= 0.0-7), assertive.files (>= 0.0-2),  
assertive.numbers (>= 0.0-2), assertive.properties (>= 0.0-4),  
assertive.sets (>= 0.0-3), assertive.types (>= 0.0-3), Biobase  
(>= 2.34.0), data.table (>= 1.10.4), dplyr (>= 0.5.0), magrittr  
(>= 1.5), openxlsx (>= 4.0.17), pathological (>= 0.1-2),  
reshape2 (>= 1.4.2), stats, stringi (>= 1.1.5),  
SummarizedExperiment (>= 1.4.0), testthat (>= 1.0.2), tidyr (>= 0.6.2), utils

**Suggests** knitr, MSnbase, rmarkdown, withr

**License** GPL-3

**LazyData** true

**biocViews** GeneExpression, DataImport, Proteomics, OneChannel,  
ProprietaryPlatforms

**RoxygenNote** 6.0.1

**Collate** 'SummarizedExperiment.R' 'checkIds.R' 'dataset-docs.R'  
'defunct.R' 'extractSampleData.R' 'get-annotations.R'  
'getSequencesWithLargestBetweenGroupVariation.R' 'list-utils.R'  
'sfread.R' 'readAdat.R' 'methods.R'  
'read-sample-submission-input-files.R' 'soma2eset.R'  
'suppressSomeFeedback.R'

**VignetteBuilder** knitr

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

**git\_branch** RELEASE\_3\_9

**git\_last\_commit** 097758c

**git\_last\_commit\_date** 2019-05-02

**Date/Publication** 2019-07-04

**Author** Richard Cotton [cre, aut],  
Aditya Bhagwat [aut]

## R topics documented:

aptamers . . . . .	3
as.ExpressionSet . . . . .	4
as.SummarizedExperiment . . . . .	5
chromosomalPositions . . . . .	5
colnamesStartWithSeqId . . . . .	6
convertSeqIdToAptamer . . . . .	6
createSampleSubmission . . . . .	7
ensemblIds . . . . .	8
extractSampleData . . . . .	8
getChromosomalPositions . . . . .	9
getEnsemblIds . . . . .	9
getGoMolecularFunctions . . . . .	10
getIntensities . . . . .	11
getKeggDefinitions . . . . .	14
getPfam . . . . .	15
getSampleIdVar . . . . .	16
getSequencesWithLargestBetweenGroupVariation . . . . .	16
getUniProtKeywords . . . . .	17
goMolecularFunction . . . . .	18
isPass . . . . .	18
isSomaEset . . . . .	19
keggDefinitions . . . . .	19
melt.WideSomaLogicData . . . . .	20
orderPlatePosition . . . . .	20
pfam . . . . .	21
readAdat . . . . .	21
readat-defunct . . . . .	22
readComments . . . . .	23
readControls . . . . .	24
readFirstChar . . . . .	25
readSamples . . . . .	25
readSlides . . . . .	26
uniprotKeywords . . . . .	27
WideSomaLogicData . . . . .	28
writeSampleSubmissionForm . . . . .	28
[.WideSomaLogicData . . . . .	29

---

aptamers

*Sequence data for the Somalogic SOMAscan assay*

---

## Description

Sequence data for the SOMAmers in the 1129 and 1310 panels of the Somalogic SOMAscan assay.

## Format

A data frame with the following columns.

**AptamerId** Character, primary key. The identifier of the aptamer sequence identified using SELEX. It consists of an SomaLogic SeqId truncated at the underscore, to remove its sequence version number.

**SomaId** Character. The SomaLogic identifier for the protein target. For the 1129 and 1310 assays, there is a one-to-one correspondence between SeqId and SomaId, but in theory there is a many-to-one correspondence.

**Target** Character. The name of the protein target, from the protein standard supplier, sometimes with additional annotation by SomaLogic.

**TargetFullName** Character. The name of the protein target, from UniProt.

**UniProt** List of character vectors. UniProt IDs for the protein target.

**EntrezGeneID** List of character vectors. Entrez Gene IDs for the gene associated with protein target.

**EntrezGeneSymbol** List of character vectors. Entrez Gene symbols for the gene associated with protein target.

**Organism** Either "Human" or the name of a virus.

**Units** Should always be RFU, short for Relative Fluorescence Units.

**Type** The protein standard source for the original SELEX performed for the sequence. "Protein" refers to human protein; a few sequences used rat protein standard sources in the SELEX.

**PlasmaDilution** What dilution factor is used in plasma?

**SerumDilution** What dilution factor is used in serum?

**IsIn1129Panel** Is the aptamer in the 1129 panel of the SOMAscan assay?

**IsIn1310Panel** Is the aptamer in the 1310 panel of the SOMAscan assay?

## References

The SOMAmers in the SomaLogic SOMAscan 1310 assay can be found in this PDF: <http://www.somalogic.com/somalogic/media/Assets/PDFs/SSM-045-REV-1-SOMAscan-Assay-1-3k-Content.pdf> Those from the 1129 assay can be found here: <http://www.somalogic.com/somalogic/media/Assets/PDFs/SSM-011-Rev-11-SOMAscan-Assay-%28V1-1k%29-Content.pdf>

## Examples

```
head(aptamers)
```

---

as.ExpressionSet      *Convert objects into ExpressionSets or MSnSets.*

---

### Description

Converts objects into ExpressionSet or MSnSet instances.

### Usage

```
as.ExpressionSet(x, log2Transform = TRUE, ...)
```

```
## S3 method for class 'WideSomaLogicData'  
as.ExpressionSet(x, log2Transform = TRUE, ...)
```

```
soma2eset(somaObj, log2Transform = TRUE)
```

```
## S3 method for class 'WideSomaLogicData'  
as.MSnSet(x, log2Transform = FALSE, ...)
```

```
as.MSnSet(x, log2Transform = FALSE, ...)
```

### Arguments

x	An object to transform. Currently only WideSomaLogicData objects are supported.
log2Transform	whether to log2 transform intensities or not
...	Passed between methods.
somaObj	A WideSomaLogicData object to transform.

### Value

ExpressionSet or MSnSet object

### Author(s)

Aditya Bhagwat

### Examples

```
somaFile <- extractSampleData()  
wideSomaData <- readAdat(somaFile)  
as.ExpressionSet(wideSomaData)  
if(requireNamespace("MSnbase"))  
{  
  as.MSnSet(wideSomaData)  
}  
  
unlink(somaFile)
```

---

 as.SummarizedExperiment

*Convert objects into SummarizedExperiments*


---

### Description

Convert objects into SummarizedExperiments.

### Usage

```
as.SummarizedExperiment(x, ...)
```

```
## S3 method for class 'WideSomaLogicData'
as.SummarizedExperiment(x, ...)
```

### Arguments

**x** An object to transform. Currently only WideSomaLogicData objects are supported.

**...** Arguments passed between methods. Currently unused.

### Value

An object of class SummarizedExperiment.

### Examples

```
somaFile <- extractSampleData()
wideSomaData <- readAdat(somaFile)
as.SummarizedExperiment(wideSomaData)
unlink(somaFile)
```

---

 chromosomalPositions *Chromosomal Positions by SomaLogic AptamerId*


---

### Description

A lookup of chromosomal positions by SomaLogic Aptamer ID.

### Format

A [GRangesList](#) with the hg38 genome. Each Aptamer ID has an element that is a [GRanges](#) object with 3 metadata columns.

**AptamerId** Character, primary key. The identifier of the aptamer sequence identified using SELEX. It consists of an SomaLogic SeqId truncated at the underscore, to remove its sequence version number.

**UniProtId** Character. UniProt ID for the protein target.

**EntrezGeneId** Character. EntrezGene IDs for the gene that produces the target protein.

**Examples**

```
head(chromosomalPositions)
```

---

```
colnamesStartWithSeqId
```

*Does the input contain Sequence ID column names?*

---

**Description**

Checks if the input begins with "SeqID."

**Usage**

```
colnamesStartWithSeqId(x)
```

**Arguments**

x                    A character vector. Typically the column names of sequence data.

**Value**

A logical vector with the same length as x.

**Examples**

```
somaFile <- extractSampleData()
wideSomaData <- readAdat(somaFile)
colnamesStartWithSeqId(wideSomaData)
unlink(somaFile)
```

---

```
convertSeqIdToAptamer    Convert SeqIds to Aptamers
```

---

**Description**

Converts SomaLogic sequence IDs to aptamer IDs.

**Usage**

```
convertSeqIdToAptamer(seqId)
```

**Arguments**

seqId                A character vector or factor of sequence IDs.

**Value**

A character vector of factors.

**Examples**

```
convertSeqIdToAptamer("2717-12_5")
```

---

 createSampleSubmission

*Create a SomaLogic Sample Submission Form*


---

## Description

Creates a data table of contents for a SomaLogic sample submission form.

## Usage

```
createSampleSubmission(slides, controls, comments, samples,
  sampleMatrix = c("EDTA-Plasma", "Sodium Citrate Plasma", "Serum"),
  siteId = "WCQ", studyName = "", studyId = "", runName = "Set A")
```

## Arguments

slides	A data frame of slide data, as imported by <a href="#">readSlides</a> .
controls	A data frame of control data, as imported by <a href="#">readControls</a> .
comments	A data frame of comment data, as imported by <a href="#">readComments</a> .
samples	A data frame of sample data, as imported by <a href="#">readSamples</a> .
sampleMatrix	A string giving the type of samples (plasma or serum).
siteId	A string giving the SomaLogic ID of your site or institution.
studyName	A string giving your institution's name for the study.
studyId	A string giving the SomaLogic ID of the study
runName	A string naming the plate.

## Value

A data table with the 96 rows (one plate worth, with each row representing a sample). It contains the following columns:

**SampleNumber** One to ninety six.

**SlideId** From slides.csv.

**Subarray** Position in slide, from 1 to 8.

**PlatePosition** Position in plate, from A1 to H12

**PercentDilution** Always 40.

**BarCode** From controls.csv.

**SampleNotes** From controls.csv.

**AssayNotes** From controls.csv.

**SampleId** From samples.csv.

**SampleMatrix** sampleMatrix for samples; blank for controls.

**SiteId** siteId for samples; blank for controls.

**StudyName** studyName for samples; blank for controls.

**TimePoint** Currently blank.

**SampleGroup** Currently blank.

**SampleDescription** Currently blank.

**StudyId** studyId for samples; blank for controls.

**RunName** runName for samples; blank for controls.

**See Also**

[writeSampleSubmissionForm](#) for usage examples.

---

ensemblIds	<i>Ensembl IDs by SomaLogic AptamerId</i>
------------	---

---

**Description**

A lookup of Ensembl IDs by SomaLogic Aptamer ID.

**Format**

A list of character vectors. The names of the list are SomaLogic Seq IDs, and the character vectors contain Ensembl IDs for each Seq ID.

**References**

More information on Ensembl IDs can be found at: <http://www.ensembl.org/index.html>

**Examples**

```
head(ensemblIds)
```

---

extractSampleData	<i>Extract the sample datasets</i>
-------------------	------------------------------------

---

**Description**

Extracts the sample datasets from the ZIP files in the extdata directory.

**Usage**

```
extractSampleData(somascanMenuVersion = c("1.3k", "1.1k"),  
  intermediateDir = tempdir())
```

**Arguments**

`somascanMenuVersion`  
String describing which data file to extract. Either "1.3k" or "1.1k".

`intermediateDir`  
A path to a directory to extract the ADATY file into.

**Value**

A path to the extracted ADAT file.

**Examples**

```
extractSampleData() ##1.3k dataset by default  
extractSampleData("1.1k")
```



---

`getChromosomalPositions`*Get Chromosomal Positions by AptamerId*

---

**Description**

Gets the chromosomal positions associated with SomaLogic aptamer IDs.

**Usage**

```
getChromosomalPositions(aptamerIds = NULL, simplify = FALSE)
```

**Arguments**

<code>aptamerIds</code>	A character vector of SomaLogic aptamer IDs, or NULL to use all aptamer IDs.
<code>simplify</code>	Logical. Should the output be collapsed into a single data.frame?

**Value**

A list of data frames. The names of the list are the input SeqIds, and the data frame associated with that element contains:

**UniProtId** Character. UniProt ID for the protein target.

**Chromosome** Character. Either '1' to '22' or 'X'. (Currently no 'Y' values.)

**StartPosition** Integer. Distance in base pairs from the 5' end of the gene to the start of the protein.

**EndPosition** Integer. Distance in base pairs from the 5' end of the gene to the end of the protein.

**Examples**

```
# Each AptamerId may have one, many, or zero associated chromosomal positions
getChromosomalPositions(c("2278-61", "4703-87_2", "4916-2"))

# Get everything in the 1310 and 1129 panels.
## Not run:
getChromosomalPositions()

## End(Not run)
```

---

`getEnsemblIds`*Get Ensembl IDs by AptamerId*

---

**Description**

Gets the Ensembl IDs associated with SomaLogic aptamer IDs.

**Usage**

```
getEnsemblIds(aptamerIds = NULL, simplify = FALSE)
```

**Arguments**

aptamerIds      A character vector of SomaLogic aptamer IDs, or NULL to use all aptamer IDs.  
 simplify        Logical. Should the output be collapsed into a single data.frame?

**Value**

A list of character vectors. The names of the list are the input Sequence IDs, and the character vector associated with that element contains the Ensembl IDs.

**Examples**

```
# Each AptamerId may have one, many, or zero associated Ensembl IDs
getEnsemblIds(c("2278-61", "4703-87", "4916-2"))

# Get everything in the 1129 panel.
## Not run:
getEnsemblIds()

## End(Not run)
```

---

```
getGoMolecularFunctions
```

*GO definitions by AptamerId*

---

**Description**

Gets the GO definitions associated with SomaLogic Sequence IDs. There are three datasets, one for each of these domains: molecular function, biological process, and cellular compartment.

**Usage**

```
getGoMolecularFunctions(aptamerIds = NULL, simplify = FALSE)

getGoBiologicalProcesses(aptamerIds = NULL, simplify = FALSE)

getGoCellularComponents(aptamerIds = NULL, simplify = FALSE)
```

**Arguments**

aptamerIds      A character vector of SomaLogic Sequence IDs, or NULL to use all 1129 Sequence IDs.  
 simplify        Logical. Should the output be collapsed into a single data.frame?

**Value**

A list of data frames. The names of the list are the input aptamerIds, and the data frame associated with that element contains:

**UniProtId** Character. UniProt ID for the protein target.

**GoId** Character. GO ID for property of the target protein.

**GoName** Character. Name corresponding to the GO ID.

**GoDefinition** Character. Description corresponding to the GO ID.

**Examples**

```
# Each AptamerId may have one, many, or zero associated GO descriptions
getGoMolecularFunctions(c("2278-61", "3505-6", "4916-2"))
getGoBiologicalProcesses(c("2278-61", "3505-6", "4916-2"))
getGoCellularComponents(c("2278-61", "3505-6", "4916-2"))

# Get everything in the 1310 and 1129 panels.
## Not run:
getGoMolecularFunctions()
getGoBiologicalProcesses()
getGoCellularComponents()

## End(Not run)
```

---

```
getIntensities          Get WideSomaLogicData attributes
```

---

**Description**

Accessors and mutators (getters and setters) for objects of class `WideSomaLogicData` or `LongSomaLogicData`.

**Usage**

```
getIntensities(x, ...)

## S3 method for class 'WideSomaLogicData'
getIntensities(x, rowsContain = c("samples",
  "sequences"), reorder = FALSE, ...)

## S3 method for class 'LongSomaLogicData'
getIntensities(x, ...)

## S3 method for class 'WideSomaLogicData'
as.matrix(x, ...)

getSampleData(x, ...)

## S3 method for class 'WideSomaLogicData'
getSampleData(x, ...)

## S3 method for class 'LongSomaLogicData'
getSampleData(x, ...)

getSequenceData(x)

getMetadata(x)

getChecksum(x)

setSequenceData(x, value)
```

```

setMetadata(x, value)

setChecksum(x, value)

setSampleData(x, value)

setIntensities(x, value, prependSeqIdToColNames = NA)

```

### Arguments

<code>x</code>	An object of class <code>WideSomaLogicData</code> or <code>LongSomaLogicData</code> .
<code>...</code>	Variables passed to and from other methods.
<code>rowsContain</code>	Either samples or sequences.
<code>reorder</code>	If TRUE, rows are reordered by <code>ExtIdentifier</code> and columns are reordered by <code>SeqId</code> , alphabetically.
<code>value</code>	Value to set the attribute to.
<code>prependSeqIdToColNames</code>	Logical. Should "SeqId." be prepended to the column names of the intensities? If NA, auto-guess whether they should be.

### Value

`getIntensities` returns a numeric matrix of intensities for each protein. Row names are taken from the `ExtIdentifier` of the input. Column names are the protein sequence IDs. (If you set `rowsContain = "sequences"`, then rows and columns and their names are swapped.)

`getSampleData` returns a data table containing the sample data. That is, the input without the intensities or attributes. There is one compulsory column:

**ExternalId** Character. A unique identifier for the sample.

These columns are not compulsory for the file spec, but are always included by `SomaLogic`:

**SampleId** Character. The customer's original sample identifier. This is either the same as the subject identifier, or the calibrator and buffer separated by a hyphen. For example "1234" or "PPT-09".

**TimePoint** Character or numeric. The time point identified by the customer.

**SampleGroup** Character. Cohort information provided by the customer.

**SampleNotes** Character. Lab comments about the sample condition.

**AssayNotes** Character. Lab comments about assay adverse events.

Common optional columns:

**PlateId** Character. Identifier for the plate. For assay experiments, this also functions as an experiment identifier.

**SlideId** Character. Agilent slide barcode.

**Subarray** Integer. Agilent subarray number from 1 to 8.

**SampleType** Character. Either "Sample", "QC", "Buffer" or "Calibrator".

**Barcode** Character. 1D Barcode of aliquot.

**Barcode2d** Character. 2D Barcode of aliquot.

**SiteId** Character. The laboratory that ran the experiment, applicable if subcontracted.

**SampleUniqueID** Character. Where multiple experiments are combined, this column can be used to ensure a unique sample ID.

**Subject\_ID** Character. Identifier for the person/animal/thing being sampled.

**HybControlNormScale** Numeric. Hybridization control normalization scale factor. For example, 1.23.

**NormScale\_40** Numeric. Median normalization scale factor, for only the sequences diluted to 40% concentration. For example, 1.23.

**NormScale\_0\_005** Numeric. Median normalization scale factor, for only the sequences diluted to 0.005% concentration. For example, 1.23.

**NormScale\_1** Numeric. Median normalization scale factor, for only the sequences diluted to 1% concentration. For example, 1.23.

**PercentDilution** Numeric. How much was the sample diluted? Either 40, 1, or 0.005.

**SampleMatrix** Character. Either "EDTA-Plasma", "Sodium Citrate Plasma" or "Serum". Applicable if different matrices are used with different samples in the file, otherwise use the StudyMatrix metadata value.

**SampleDescription** Character. Free text describing the sample.

**TubeUniqueID** Character. A unique identifier for every sample tube, assigned by the customer.

**RowCheck** Character. Either "PASS" or "FAIL". Did the sample pass quality control checks?

getSequenceData returns a data table containing the sequence data. "Sequence" can be considered synonymous with "feature" or "SOMAmer reagent". The following columns are compulsory.

**SeqId** A unique identifier for the SOMAmer reagent.

**Target** The unique name for the targeted proteins.

The following columns are optional but should always be provided by SomaLogic.

**SomaId** Character. The SomaLogic identifier for the protein target. For the 1129 and 1310 assays, there is a one-to-one correspondence between SeqId and SomaId, but in theory there is a many-to-one correspondence.

**TargetFullName** Character. A description of the proteins targeted by the SOMAmer reagent, taken from UniProt.

**UniProt** Character. The UniProt identifiers for the targeted proteins. Older versions of the file format may also contain the value 'Family', referring to a whole family of proteins, though this behaviour is considered deprecated.

**EntrezGeneID** Character. The Entrez Gene identifiers for the genes corresponding to the targeted proteins.

**EntrezGeneSymbol** Character. The Entrez Gene symbols for the genes corresponding to the targeted proteins.

**Organism** Character. What animal does the target protein refer to?

**ColCheck** Character. Either PASS or FAIL. Did the sequence pass quality control checks?

**Units** Character. Unit of measurement for the SOMAmer reagent abundances. Should always be RFU.

**Type** Character. One of "Spuriomer", "Protein", "Hybridization Control", "Non-human Protein".

**Cal\_\*PlateId\*** Numeric. Calibration scale factor for each plate. For example, 1.23.

**CalReference** Numeric. Calibration reference intensity, in RFU. For example, 543.21.

**Dilution** Numeric. Concentration of the diluted sample compared to the neat sample, as a percentage. Either 40, 1, or 0.005.

getChecksum and getMetadata return the checksum (a string) and metadata (a list) attributes of the input.

**See Also**[readAdat](#)**Examples**

```

# Get the sample dataset
soma_file <- extractSampleData()
wide_soma_data <- readAdat(soma_file)

# Access its components
checksum <- getChecksum(wide_soma_data)
metadata <- getMetadata(wide_soma_data)
sequenceData <- getSequenceData(wide_soma_data)
sampleData <- getSampleData(wide_soma_data)

# Intensities of a WideSomaLogicData object are a matrix
intWideSamp <- getIntensities(wide_soma_data)

# Not run due to side effects of View
View(intWideSamp, "Wide intensities, samples per row")

intWideSeq <- getIntensities(           # The transpose
  wide_soma_data,
  rowsContain = "sequences"
)

View(intWideSeq, "Wide intensities, seqs per row")

# Sample data is always a data table
sampWide <- getSampleData(wide_soma_data)

View(sampWide, "Wide sample data")

# For LongSomaLogicData objects, the intensities are returned
# as a data.table
long_soma_data <- reshape2::melt(wide_soma_data)
intLong <- getIntensities(long_soma_data)

View(intLong, "Long intensities")

# Sample data has a different shape now
sampLong <- getSampleData(long_soma_data)

View(sampLong, "Long sample data")

```

---

getKeggDefinitions      *KEGG definitions, modules, and pathways by AptamerId*

---

**Description**

Gets the KEGG definitions, modules, and pathways associated with SomaLogic aptamer IDs.

**Usage**

```
getKeggDefinitions(aptamerIds = NULL, simplify = FALSE)
```

```
getKeggModules(aptamerIds = NULL, simplify = FALSE)
```

```
getKeggPathways(aptamerIds = NULL, simplify = FALSE)
```

**Arguments**

**aptamerIds** A character vector of SomaLogic aptamer IDs, or NULL to use all aptamer IDs.  
**simplify** Logical. Should the output be collapsed into a single data.frame?

**Value**

A list of data frames. The names of the list are the input aptamerIds, and the data frame associated with that element contains:

**UniProtId** Character. UniProt ID for the protein target.

**KeggId** Character. KEGG ID for the gene that produces the target protein.

**KeggDefinition** Character. Description corresponding to the KEGG ID.

**KeggCytogenicLocation** Character. KEGG determination of the gene's locus.

**Examples**

```
# Each AptamerId may have one, many, or zero associated KEGG descriptions
getKeggDefinitions(c("2278-61", "3505-6", "4916-2"))
getKeggModules(c("2278-61", "3505-6", "4916-2"))
getKeggPathways(c("2278-61", "3505-6", "4916-2"))

# Get everything in the 1310 and 1129 panels.
## Not run:
getKeggDefinitions()
getKeggModules()
getKeggPathways()

## End(Not run)
```

---

getPfam

*Get PFAM IDs by AptamerId*

---

**Description**

Gets the PFAM IDs and descriptions associated with SomaLogic aptamer IDs.

**Usage**

```
getPfam(aptamerIds = NULL, simplify = FALSE)
```

**Arguments**

**aptamerIds** A character vector of SomaLogic aptamer IDs, or NULL to use all aptamer IDs.  
**simplify** Logical. Should the output be collapsed into a single data.frame?

**Value**

A list of data frames. The names of the list are the input SeqIds, and the data frame associated with that element contains:

**EntrezGeneId** Character. EntrezGene IDs for the gene that produces the target protein.

**PfamId** Character. PFAM ID for a property of the target protein.

**PfamDescription** Character. Description of a PFAM protein property.

**Examples**

```
# Each AptamerId may have one, many, or zero associated PFAM descriptions
getPfam(c("2278-61", "4703-87", "4916-2"))

# Get everything in the 1310 and 1129 panels.
## Not run:
getPfam()

## End(Not run)
```

---

<code>getSampleIdVar</code>	<i>Get name of sample id variable</i>
-----------------------------	---------------------------------------

---

**Description**

Get name of sample id variable

**Usage**

```
getSampleIdVar(somaEset)
```

**Arguments**

somaEset          eset with soma data

**Value**

character

---

<code>getSequencesWithLargestBetweenGroupVariation</code>	<i>Get sequences with the largest between group variation</i>
---	---

---

**Description**

Get the sequences with the largest between group variation in mean log intensity.

**Usage**

```
getSequencesWithLargestBetweenGroupVariation(x, n = 10,
  group = ~SampleGroup, ...)
```



**Arguments**

x	An object of class LongSomaLogicData.
n	An integer of the number of sequences to return, passed to <code>top_n</code> .
group	A formula, string or quoted column name of the column that defines the groups to split by.
...	Passed to and from methods, but currently unused.

**Value**

A data table with n rows and the following columns.

**SeqId** SomaLogic sequence identifier.

**VariationBetweenGroups** The largest mean log intensity within a group divided by the smallest mean log intensity within a group.

---

getUniProtKeywords      *Get UniProt Keywords by AptamerId*

---

**Description**

Gets the UniProt Keywords associated with SomaLogic aptamer IDs.

**Usage**

```
getUniProtKeywords(aptamerIds = NULL, simplify = FALSE)
```

**Arguments**

aptamerIds	A character vector of SomaLogic aptamer IDs, or NULL to use all aptamer IDs.
simplify	Logical. Should the output be collapsed into a single data.frame?

**Value**

A list of data frames. The names of the list are the input SeqIds, and the data frame associated with that element contains:

**UniProtId** Character. The UniProt ID that the Keyword is associated with.

**Keyword** Character. A UniProt Keyword associated with the AptamerId and UniProt ID.

**Examples**

```
# Each AptamerId may have one, many, or zero associated Ensembl IDs
getUniProtKeywords(c("2278-61", "4703-87", "4916-2"))

# Get everything in the 1310 and 1129 panels.
## Not run:
getUniProtKeywords()

## End(Not run)
```

goMolecularFunction *GO definitions by SomaLogic AptamerId*

---

**Description**

A lookup of GO (Gene Ontology) definitions, for molecular function, biological process, and cellular component by SomaLogic Aptamer ID.

**Format**

A list of data frames, each with the following columns.

**UniProtId** Character. UniProt ID for the protein target.

**GoId** Character. GO ID for property of the target protein.

**GoName** Character. Name corresponding to the GO ID.

**GoDefinition** Character. Description corresponding to the GO ID.

**References**

More information on GO can be found at: <http://geneontology.org/>

**Examples**

```
head(goMolecularFunction)
head(goBiologicalProcess)
head(goCellularComponent)
```

---

isPass *Is the value a pass*

---

**Description**

Checks if a string is "PASS".

**Usage**

```
isPass(x)
```

**Arguments**

x A character vector or factor.

**Value**

A logical vector, the same length as the input, which is TRUE whenever the input is the string "PASS". Missing values return FALSE.

**Author(s)**

Richard Cotton

---

isSomaEset	<i>Is object a soma eset?</i>
------------	-------------------------------

---

**Description**

Is object a soma eset?

**Usage**

```
isSomaEset(esetObj)
```

**Arguments**

esetObj      eSet

**Value**

logical

---

keggDefinitions	<i>KEGG definitions, modules, and pathways by SomaLogic AptamerId</i>
-----------------	---

---

**Description**

A lookup of KEGG (Kyoto Encyclopedia of Genes and Genomes) definitions, modules, and pathways by SomaLogic Aptamer ID.

**Format**

A list of data frames, each with the following columns.

**UniProtId** Character. UniProt ID for the protein target.

**KeggId** Character. KEGG ID for the gene that produces the target protein.

**KeggDefinition** Character. Description corresponding to the KEGG ID.

**KeggCytogenicLocation** Character. KEGG determination of the gene's locus.

**References**

More information on KEGG can be found at: <http://www.kegg.jp/>

**Examples**

```
head(keggDefinitions)
head(keggModules)
head(keggPathways)
```

---

```
melt.WideSomaLogicData
```

*Melt a WideSomaLogicData object*

---

### Description

Convert a WideSomaLogicData object from wide format to long format.

### Usage

```
## S3 method for class 'WideSomaLogicData'
melt(data, ..., na.rm = FALSE,
      value.name = "Intensity")
```

### Arguments

data	An object of class WideSomaLogicData.
...	Currently unused.
na.rm	TODO
value.name	TODO

### Value

An object of class LongSomaLogicData that inherits from data.frame. This function melts the sample data contained in a WideSomaLogicData object so the sequence IDs are contained in a single column SeqID, with the corresponding intensities in a single column named Intensity. the SequenceData attribute of the input is then merged into this. the Metadata and Checksum attributes are preserved.

---

```
orderPlatePosition      Order Plate position
```

---

### Description

Order Plate position

### Usage

```
orderPlatePosition(pp)
```

### Arguments

pp	character vector with plate positions
----	---------------------------------------

### Value

ordered vector

---

pfam	<i>PFAM IDs by SomaLogic AptamerId</i>
------	--

---

### Description

A lookup of PFAM (Protein FAMilies) IDs and descriptions by SomaLogic Aptamer ID.

### Format

A list of data frames, each with the following columns.

**EntrezGeneId** Character. EntrezGene IDs for the gene that produces the target protein.

**PfamId** Character. PFAM ID for a property of the target protein.

**PfamDescription** Character. Description of a PFAM protein property.

### References

More information on PFAM IDs can be found at: <http://pfam.xfam.org/>

### Examples

```
head(pfam)
```

---

readAdat	<i>Read a SomaLogic data file</i>
----------	-----------------------------------

---

### Description

Reads a SomaLogic ADAT data file.

### Usage

```
readAdat(file, keepOnlyPasses = TRUE, keepOnlySamples = TRUE,
  dateFormat = "%Y-%m-%d", verbose = getOption("verbose"))
```

### Arguments

file	A string containing the path to the file to be read.
keepOnlyPasses	A logical value indicating whether or not to keep only the rows and columns where the data quality was considered to be passable.
keepOnlySamples	A logical value indicating whether or not to keep only the rows containing actual samples (as opposed to QC, buffer, and calibrator samples).
dateFormat	A string describing the format of the dates contained in the file's metadata. See <a href="#">strptime</a> for how to specify these.
verbose	Logical value indicating whether (lots of) diagnostic messages should be shown.

**Value**

An object of class `WideSomaLogicData`, which inherits from `data.table`. The return value consists of a data frame where each row represents a sample. Initial columns contain sample metadata and later columns contain intensities of proteins. The specific metadata columns are not fixed, but the most useful ones described below should always be present.

**SampleUniqueID** A unique identifier for the sample.

**Subject\_ID** A unique identifier for the person being sampled.

**RowCheck** This should have the value "PASS" if the data quality is acceptable.

Columns of proteins intensities have a name beginning `SeqId..` Return value also has three attributes.

**SequenceData** A data frame.

**Metadata** A list of experimental metadata values.

**Checksum** A SHA1 checksum to ensure file integrity.

**Author(s)**

Richard Cotton

**Examples**

```
somaFile <- extractSampleData()
wideSomaData <- readAdat(somaFile)
str(wideSomaData, list.len = 35)
unlink(somaFile)
```

---

readat-defunct

*Defunct objects in readat*

---

**Description**

Functions that are no longer used.

**Arguments**

... Arguments to functions you shouldn't use.

**Value**

Possibly nonsense; don't use these functions.

---

`readComments`*Read SomaLogic Sample Submission Comments File*

---

**Description**

Read SomaLogic Sample Submission Comments File

**Usage**

```
readComments(file = "comments.csv")
```

**Arguments**

`file` A string denoting the path to an input CSV file. See Input file specification section.

**Value**

A `data.table` with 96 rows and 2 columns.

**PlatePosition** A letter followed by a number, constructed from the Subarray ("A" for 1, "B" for 2, etc.) and the slide number from 1 to 12.

**SampleNotes** Either "red", "yellow", "turbid", "red, turbid", or "yellow, turbid", or no notes.

**AssayNotes** Free text describing assay notes, particularly problems.

**Input file specification**

A CSV file without a header line containing up to 96 rows and three columns as follows.

1. Plate positions. Not all positions need to be included, and they don't need to be in order.
2. Sample notes. Either "red", "yellow", "turbid", "red, turbid", or "yellow, turbid", or no notes.
3. Assay notes. Free text.

**See Also**

[readSlides](#), [readControls](#), and [readSamples](#) for reading other submission forms and [writeSampleSubmissionForm](#) for usage examples.

**Examples**

```
# See ?writeSampleSubmissionForm for a more complete example
withr::with_dir(
  system.file("extdata", package = "readat"),
  {
    (comments <- readComments())
  }
)
```

---

readControls	<i>Read SomaLogic Sample Submission Controls File</i>
--------------	---

---

**Description**

Read SomaLogic Sample Submission Controls File

**Usage**

```
readControls(file = "controls.csv")
```

**Arguments**

**file** A string denoting the path to an input CSV file. See Input file specification section.

**Value**

A data.table with 96 rows and 2 columns.

**PlatePosition** A letter followed by a number, constructed from the Subarray ("A" for 1, "B" for 2, etc.) and the slide number from 1 to 12.

**BarCode** Sample barcode for QC, Calibrator, and Buffer samples, in the form "I" followed by 6 digits.

**Input file specification**

A CSV file without a header line containing up to 96 rows and two columns as follows.

1. Plate positions from A1, A2, through to H12.
2. Barcodes in the form "I" followed by 6 digits.

**See Also**

[readSlides](#), [readComments](#), and [readSamples](#) for reading other submission forms and [writeSampleSubmissionForm](#) for usage examples.

**Examples**

```
# See ?writeSampleSubmissionForm for a more complete example
withr::with_dir(
  system.file("extdata", package = "readat"),
  {
    (controls <- readControls())
  }
)
```



---

readFirstChar	<i>Read the first character of each line</i>
---------------	--

---

**Description**

Reads the first character of each line of a text file.

**Usage**

```
readFirstChar(file)
```

**Arguments**

file            A string or readable connection.

**Value**

A character vector of single characters.

**References**

See <http://stackoverflow.com/q/27747426/134830>

---

readSamples	<i>Read SomaLogic Sample Submission Samples File</i>
-------------	--

---

**Description**

Read SomaLogic Sample Submission Samples File

**Usage**

```
readSamples(file = "samples.csv")
```

**Arguments**

file            A string denoting the path to an input CSV file. See Input file specification section.

**Value**

A data.table with 96 rows and 2 columns.

**PlatePosition** A letter followed by a number, constructed from the Subarray ("A" for 1, "B" for 2, etc.) and the slide number from 1 to 12.

**SampleId** 9 digits.

**Input file specification**

A CSV file without a header line containing up to 96 rows and at least two columns as follows.

1. Plate positions. Not all positions need to be included, and they don't need to be in order.
2. Samples IDs in the form of 9 digits, or "NO READ".

Additional columns are ignored.

**See Also**

[readSlides](#), [readComments](#), and [readControls](#) for reading other submission forms and [writeSampleSubmissionForm](#) for usage examples.

**Examples**

```
# See ?writeSampleSubmissionForm for a more complete example
withr::with_dir(
  system.file("extdata", package = "readat"),
  {
    (samples <- readSamples())
  }
)
```

---

readSlides

*Read SomaLogic Sample Submission Slides File*


---

**Description**

Read SomaLogic Sample Submission Slides File

**Usage**

```
readSlides(file = "slides.csv")
```

**Arguments**

**file** A string denoting the path to an input CSV file. See Input file specification section.

**Value**

A data.table with 96 rows and 5 columns.

**SampleNumber** The integers 1 to 96.

**SlideId** The slide IDs from the input file.

**Subarray** Integers from 1 to 8 denoting the sample position for the slide.

**PlatePosition** A letter followed by a number, constructed from the Subarray ("A" for 1, "B" for 2, etc.) and the slide number from 1 to 12.

**PercentDilution** Always 40.

### Input file specification

A CSV file without a header line containing up to twelve rows and one column as follows.

1. Slide IDs, each 12 digits long.

### Note

For partially filled plates (where there are less than 96 samples), there isn't enough information in the file to programmatically determine which slide IDs correspond to which sample IDs. In this case please manually create the slides dataset.

### See Also

[readControls](#), [readComments](#), and [readSamples](#) for reading other submission forms and [writeSampleSubmissionForm](#) for usage examples.

### Examples

```
# See ?writeSampleSubmissionForm for a more complete example
withr::with_dir(
  system.file("extdata", package = "readat"),
  {
    (slides <- readSlides())
  }
)
```

---

uniprotKeywords

*UniProt Keywords by SomaLogic AptamerId*

---

### Description

A lookup of UniProt keywords by SomaLogic Aptamer ID.

### Format

A list of data frames, each with the following columns.

**UniProtId** Character. UniProt ID for the protein target.

**Keyword** Character. A UniProt keyword.

### References

More information on UniProt keywords can be found at: <http://www.uniprot.org/help/keywords>

### Examples

```
head(uniprotKeywords)
```

---

WideSomaLogicData      *Create a WideSomaLogicData object*

---

### Description

Creates and object of class WideSomaLogicData.

### Usage

```
WideSomaLogicData(sampleAndIntensityData, sequenceData, metadata,
  checksum = paste0(rep.int("f", 40), collapse = ""))
```

### Arguments

sampleAndIntensityData	A data.table of sample and intensity data.
sequenceData	A data.table of sequence data.
metadata	A list of metadata.
checksum	A string containing a SHA1 checksum.

### Value

An object of class WideSomaLogicData.

### Examples

```
wsld <- WideSomaLogicData(
  data.frame(SampleStuff = letters, IntensityStuff = rnorm(26)),
  data.frame(SequenceStuff = LETTERS),
  list(MetadataStuff = Sys.Date())
)
str(wsld)
```

---

writeSampleSubmissionForm      *Write a SomaLogic sample submission form*

---

### Description

Writes a SomaLogic sample submission form to Excel XLSX file.

### Usage

```
writeSampleSubmissionForm(submission, outdir = ".")
```

### Arguments

submission	A data.frame created by <a href="#">createSampleSubmission</a> .
outdir	A string denoting the path to the directory where the output file should be written.

**Value**

The name of the XLSX file is invisibly returned, but the function is mostly called for the side effect of writing this file.

**See Also**

[readSlides](#) and [createSampleSubmission](#)

**Examples**

```
# Import the input files
withr::with_dir(
  system.file("extdata", package = "readat"),
  {
    slides <- readSlides()
    controls <- readControls()
    comments <- readComments()
    samples <- readSamples()
  }
)

# Create the sample submission form and write to Excel spreadsheet
submission <- createSampleSubmission(
  slides, controls, comments, samples,
  studyName = "TaHERi01", studyId = "WCQ-16-002"
)
writeSampleSubmissionForm(submission, tempdir())
```

---

[.WideSomaLogicData     *Indexing for WideSomaLogicData objects*

---

**Description**

Wrapper to [.data.table, ensuring that the SequenceData, Metadata and Checksum attributes are preserved.

**Usage**

```
## S3 method for class 'WideSomaLogicData'
x[...]
```

**Arguments**

x	A WideSomaLogicData object.
...	Passed to [.data.table.

**Value**

If the indexing returns a A WideSomaLogicData object.

**See Also**

[data.table](#)

**Examples**

```
soma_file <- extractSampleData()
wide_soma_data <- readAdat(soma_file)

# Indexing returns a data.table, so the WideSomaLogicClass is preserved
wide_soma_data[1:5, list(`SeqId.3896-5_2`)]
# Indexing simplifies to a numeric vector, so the class is lost
wide_soma_data[1:5, `SeqId.3896-5_2`]

# Ignore the intensity columns (as per getSampleData)
j <- !colnamesStartWithSeqId(wide_soma_data)
wide_soma_data[1:5, j, with = FALSE]
unlink(soma_file)
```

# Index

[.WideSomaLogicData, 29

aptamers, 3

as.ExpressionSet, 4

as.matrix.WideSomaLogicData  
(getIntensities), 11

as.MSnSet (as.ExpressionSet), 4

as.SummarizedExperiment, 5

chromosomalPositions, 5

colnamesStartWithSeqId, 6

convertSeqIdToAptamer, 6

createSampleSubmission, 7, 28, 29

data.table, 29

defunct (readat-defunct), 22

ensemblIds, 8

extractSampleData, 8

getChecksum (getIntensities), 11

getChromosomalPositions, 9

getEnsemblIds, 9

getGoBiologicalProcesses  
(getGoMolecularFunctions), 10

getGoCellularComponents  
(getGoMolecularFunctions), 10

getGoMolecularFunctions, 10

getIntensities, 11

getKeggDefinitions, 14

getKeggModules (getKeggDefinitions), 14

getKeggPathways (getKeggDefinitions), 14

getMetadata (getIntensities), 11

getPfam, 15

getSampleData (getIntensities), 11

getSampleIdVar, 16

getSequenceData (getIntensities), 11

getSequencesWithLargestBetweenGroupVariation,  
16

getUniProtKeywords, 17

goBiologicalProcess  
(goMolecularFunction), 18

goCellularComponent  
(goMolecularFunction), 18

goMolecularFunction, 18

GRanges, 5

GRangesList, 5

isPass, 18

isSomaEset, 19

keggDefinitions, 19

keggDefinitions1129 (keggDefinitions),  
19

keggModules (keggDefinitions), 19

keggModules1129 (keggDefinitions), 19

keggPathways (keggDefinitions), 19

keggPathways1129 (keggDefinitions), 19

melt.WideSomaLogicData, 20

orderPlatePosition, 20

pfam, 21

readAdat, 14, 21

readat-defunct, 22

readComments, 7, 23, 24, 26, 27

readControls, 7, 23, 24, 26, 27

readFirstChar, 25

readSamples, 7, 23, 24, 25, 27

readSlides, 7, 23, 24, 26, 26, 29

setChecksum (getIntensities), 11

setIntensities (getIntensities), 11

setMetadata (getIntensities), 11

setSampleData (getIntensities), 11

setSequenceData (getIntensities), 11

soma2eset (as.ExpressionSet), 4

strptime, 21

top\_n, 17

uniprotKeywords, 27

WideSomaLogicData, 28

WideSomaLogicDataAttributes  
(getIntensities), 11

writeSampleSubmissionForm, 8, 23, 24, 26,  
27, 28