artMS:Analytical R Tools for Mass Spectrometry

David Jimenez-Morales

2018-09-26

Contents

artMS provides a set of tools for the analysis of proteomics label-free datasets.

Overview

`artMS` is an R package that provides a set of tools for the analysis and integration of large-scale proteomics (mass-spectrometry-based) datasets
obtained using the popular proteomics software package MaxQuant. The functions available at `artMS` can be grouped in 4 major categories:

- Multiple quality control (QC) functions.
- Relative quantification using MSstats.
- Downstream analysis of quantifications
- A set of miscellaneous functions including:
  - Generates input file for SAINTq and SAINTexpress, Photon, and Phosfate
  - Enrichment analysis
  - Clustering Analysis
  - PCA analysis

  - Heatmaps of log2fc and abundance

`artMS` performs the different analyses taking as input the following files:

- `evidence.txt` file: The output of the quantitative proteomics software package `MaxQuant`.
- `keys.txt` (tab-delimited) txt file generated by the user describing the experimental designed (check below to learn how to create it).
- `contrast.txt` (tab-delimited) txt file generated by the user with the comparisons between conditions to be quantified (check below to learn how to create it).
- `config.yaml`: a configuration file which enables the customization of a number of parameters for the quantification (and other operations, including QC analyses, charts and annotations). A configuration file template can be generated by running `artms_writeConfigYamlFile()`

How to install

We assume that you have both R and RStudio already installed on your system.

`artMS` is currently under revision by BioConductor. Until officially accepted, the development version can be installed directly from Github.

Before starting, make sure that your system is running an `R version >= 3.5` or otherwise nothing will work (Bioconductor requirement). You can check the R version currently running on your system by executing this command in RStudio:

```
getRversion()
```

If the outcome is `>= 3.5.0`, congratulations! you can move forward

If it is not, then you need to install the latest version of R in your system. After updating to the latest R version, open RStudio and try again `getRversion()` to make sure it worked.

Once you have the latest `R (>= 3.5)` version running on your system, follow these steps:

- Install annotation packages from Bioconductor. The installation of these packages returns multiple warnings that do not affect the functionality of `artMS` but could generate problems during the installation of `artMS`.

```
source("https://bioconductor.org/biocLite.R")
biocLite(c('org.Ag.eg.db', 'org.At.tair.db', 'org.Bt.eg.db', 'org.Ce.eg.db', 'org.Cf.eg.db', 'org.Dm.eg
```

- Install `artMS`:

```
install.packages("devtools")
library(devtools)
install_github("biodavidjm/artMS", build_vignettes=TRUE)
```

- Check that it is up and running by checking, for example, the documentation of the qc function `artms_qualityControlEvidenceBasic`:

```
library(artMS)
?artms_qualityControlEvidenceBasic
```

Once installed, we suggest you to do a quick test by running the quality control functions using the "evidence" (`artms_data_ph_evidence`) and "keys" (`artms_data_ph_keys`) files included in `artMS` as test datasets.

```
# First go to a local working directory: several pdfs will be generated
# setwd("/path/to/your/working/directory/")

# And run:
```

```
artms_qualityControlEvidenceBasic(evidence_file = artms_data_ph_evidence,
                                  keys_file = artms_data_ph_keys,
                                  prot_exp =  "PH")
```

(To learn more about these testing datasets, check the documentation by running `?artms_data_ph_keys` or `?artms_data_ph_evidence` on the R console)

Once the QC is done, go to the folder `"/path/to/your/working/directory/"` and check out all the generated QC (pdf) files.

Input files

Three basic (tab-delimited) files are required to perform the full pack of operations:

`evidence.txt`

The output of the quantitative proteomics software package MaxQuant. It combines all the information about the identified peptides.

`keys.txt`

Tab delimited file generated by the user. It summarizes the experimental design of the evidence file. When using `artMS`, the `keys.txt` file will be merged with the `evidence.txt` by the "RawFile" column. Each RawFile corresponds to an unique individual experimental technical replicate / biological replicate / Condition / Run.

For any basic label-free proteomics experiment, the keys file must contain the following columns:

- **RawFile**: The name of the RAW-file for which the mass spectral data was derived from.
- **IsotopeLabelType**: `'L'` for label free experiments (`'H'` will be used for SILAC experiments, see below)
- **Condition**:
  - Use only numbers (0 - 9) and the letters (A - Z, both uppercase and lowercase) for the Conditions' names. The only special character allowed is underscore (_).
  - A condition name CANNOT start with a number.
- **BioReplicate**: biological replicate number. Use as prefix the corresponding `Condition` name, and add as sufix a `dash (-)` and the number of biological replicate. For example, if condition `H1N1_06H` has too biological replicates, name them as `H1N1_06H-1` and `H1N1_06H-2`
- **Run**: a number for all the MS runs. It will be specially useful when having technical replicates.

Example of keys file: chech the data object: `artms_data_ph_keys`

RawFile	IsotopeLabelType	Condition	BioReplicate	Run
qx006145	L	Cal33	Cal33-1	1
qx006148	L	Cal33	Cal33-4	4
qx006151	L	HSC6	HSC6-2	6
qx006152	L	HSC6	HSC6-3	7

Tip: it is recommended to use Microsoft Excel (OpenOffice Cal / or similar) to generate the keys file. *Do not forget* to choose the *format = Tab Delimited Text (.txt)* when saving the file (use *save as* option)

`contrast.txt`

The comparisons between conditions that the user wants to quantify. For example, to quantify changes in protein abundance between wild type `WT_A549` relative to two additional experimental conditions with drugs `WT_DRUG_A` and `WT_DRUG_B`, but also changes in protein abundance between `DRUG_A` and `DRUG_B`, the contrast file would look like this:

```
WT_DRUG_A-WT_A549
WT_DRUG_B-WT_A549
WT_DRUG_A-WT_DRUG_B
```

Requirements:

- The two conditions to be compared must be separated by a dash symbol (`-`), and only one dash symbol is allowed, i.e., only one comparison per line.

As a result of the quantification, the condition on the left will take the positive log2FC sign -if the protein is more abundant in condition `WT_DRUG_A`, and the condition on the right the negative log2FC -if a protein is more abundant in condition `WT_A549`.

The configuration file (`.yaml`)

The configuration file (in `yaml` format) contains the configuration details for the quantification performed by `artMS` using `MSstats`.

To generate a sample configuration file, go to the project folder (`setwd(/path/to/your/working/folder/)`) and execute:

```
artms_writeConfigYamlFile(config_file_name = "config.yaml" )
```

Open the `config.yaml` file with your favorite editor (RStudio works very well as well). *Although it might look complex, the default options work very well.*

The configuration (`yaml`) file contains the following sections:

Section: `files`

```
files :
  evidence : /path/to/the/evidence.txt
  keys : /path/to/the/keys.txt
  contrasts : /path/to/the/contrast.txt
  output : /path/to/the/results_folder/ph-results.txt
```

The file `path/name` of the required files. It is recommended to create a new folder in your folder project (for example, `results_folder`). The results file name (e.g. `-results.txt`) will be used as prefix for the several files (`txt` and `pdf`) that will be generated.

Section: `qc`

```
qc:
  basic: 1 # 1 = yes; 0 = no
  extended: 1 # 1 = yes; 0 = no
```

Select to perform both 'basic' and 'extended' quality control. Read below to find out more about the details of each type of analysis.

Section: `data`

```
data:
  enabled : 1 # 1 = yes; 0 = no
  fractions:
    enabled : 0 # 1 for protein fractionation
  silac:
    enabled : 0 # 1 for SILAC experiments
  filters:
    enabled : 1
    contaminants : 1
    protein_groups : remove # remove, keep
    modifications : AB # PH, UB, AB, APMS
  sample_plots : 1 # correlation plots
```

Let's break it down `data`:

- `enabled`:
  - `1`: to pre-process the data provided in the *files* section.
  - `0`: won't process the data (and a pre-generated MSstats file will be expected)
- `fractions`: Multiple fractionation or separation methods are often combined in proteomics to improve signal-to-noise and proteome coverage and to reduce interference between peptides in quantitative proteomics.
  - `enabled : 1` for fractionation dataset. See **Special case: Protein Fractionation** below for details
  - `enabled : 0` no fractions
- `silac`:
  - `enabled : 1`: check if the files belong to a SILAC experiment. See **Special case: SILAC** below for details
  - `enabled : 0`: it does not
- `filters`:
  - `enabled : 1` Enables filtering
  - `contaminants : 1` Removes contaminants (`CON__` and `REV__` labeled by MaxQuant)
  - `protein_groups : remove` choose whether `remove` or `keep` protein groups
  - `modifications : AB` any of the proteomics experiments, `PH`, `UB`, or `AC` for posttranslational modifications, `AB` or `APMS` otherwise.
- `sample_plots`
  - `1` Generate correlation plots
  - `0` otherwise

Section: `msstats`

```
msstats :
  enabled : 1
  msstats_input :
  profilePlots : none # before, after, before-after (recommended), none
  normalization_method : equalizeMedians # globalStandards (include a reference protein(s) ), equalizeM
  normalization_reference :  # if globalStandards is chosen, then an UNIPROT ID must be provided.
  summaryMethod : TMP # "TMP"(default) means Tukey's median polish, which is robust estimation method.
  censoredInt : NA  # Missing values are censored or at random. 'NA' (default) assumes that all 'NA's i
  cutoffCensored : minFeature  # Cutoff value for censoring. only with censoredInt='NA' or '0'. Default
  MBimpute : 1 # only for summaryMethod="TMP" and censoredInt='NA' or '0'. TRUE (default) imputes 'NA'
  feature_subset: all # all|highQuality  : highQuality seems to be buggy right now
```

Let's break it down:

- `enabled :` Choose `1` to run MSstats, `0` otherwise.
- `msstats_input :` blank if MSstats is going to be run (`enabled : 1`). But if otherwise (`enabled : 0`) `then provide the path to the previously generated`evidence-msss.txt‘
- `profilePlots :` Several profile plots available.
  - `before` plots only before normalization
  - `after` plots only after normalization
  - `before-after`: recommended, although computational expensive (time consuming)
  - `none` no normalization plots (convenient if time limitations)
- `normalization_method :` available options:
  - `equalizeMedians`
  - `quantile`
  - `0`: no normalization (not recommended)
  - `globalStandards` if selected, specified the reference protein in `normalization_reference` (next)
- `normalization_reference :` an UniProt id if `globalStandards` is chosen as the `normalization_method` (above)
- `summaryMethod :` TMP # "TMP"(default) means Tukey's median polish, which is robust estimation method. "linear" uses linear mixed model. "logOfSum" conducts log2 (sum of intensities) per run.
- `censoredInt :`
  - `NA` Missing values are censored or at random. 'NA' (default) assumes that all 'NA's in 'Intensity' column are censored.
  - `0` uses zero intensities as censored intensity. In this case, NA intensities are missing at random. The output from Skyline should use `0`. Null assumes that all `NA` intensities are randomly missing.
- `cutoffCensored :`
  - `minFeature` Cutoff value for censoring. Only with `censoredInt='NA'` or `0`. Default is 'minFeature', which uses minimum value for each feature.
  - `minFeatureNRun` uses the smallest between minimum value of corresponding feature and minimum value of corresponding run.
  - `minRun` uses minimum value for each run.
- `MBimpute :`
  - `TRUE` only for `summaryMethod="TMP"` and `censoredInt='NA'` or `0`. TRUE (default) imputes 'NA' or '0' (depending on censoredInt option) by Accelerated failure model.
  - `FALSE` uses the values assigned by cutoffCensored.
- `feature_subset :`
  - `all` : default
  - `highQuality` : this option seems to be buggy right now

Check MSstats documentation to find out more about every option.

Section: `output_extras`

```
enabled : 1 # if 0, won't process anything on this section
annotate :
  enabled: 1 # if 1, will generate a `-results-annotated.txt` file that including Gene and Protein.Na
  specie : HUMAN  # Supported species: HUMAN, MOUSE, ANOPHELES, ARABIDOPSIS, BOVINE, WORM, CANINE, FL
plots:
  volcano: 1
  heatmap: 1
  LFC : -1.5 1.5 # Range of minimal log2fc
  FDR : 0.05
  heatmap_cluster_cols : 0
  heatmap_display : log2FC # log2FC or pvalue
```

Extra actions to perform based on the MSstats results, including *annotations* and *plots* (heatmaps and volcano plots)

Special case: Protein fractionation

To handle protein fractionation experiments, two options need to be activated

1. The keys' file must contain an additional column named "`FractionKey`" with the information about fractions. For example:

Raw.file	IsotopeLabelType	Condition	BioReplicate	Run	FractionKey
S9524_Fx1	L	AB	AB-1	1	1
S9524_Fx2	L	AB	AB-1	1	2
S9524_Fx3	L	AB	AB-1	1	3
S9524_Fx4	L	AB	AB-1	1	4
S9524_Fx5	L	AB	AB-1	1	5
S9524_Fx6	L	AB	AB-1	1	6
S9524_Fx7	L	AB	AB-1	1	7
S9524_Fx8	L	AB	AB-1	1	8
S9524_Fx9	L	AB	AB-1	1	9
S9524_Fx10	L	AB	AB-1	1	10
S9525_Fx1	L	AB	AB-2	2	1
S9525_Fx2	L	AB	AB-2	2	2
S9525_Fx3	L	AB	AB-2	2	3
S9525_Fx4	L	AB	AB-2	2	4
S9525_Fx5	L	AB	AB-2	2	5
S9525_Fx6	L	AB	AB-2	2	6
S9525_Fx7	L	AB	AB-2	2	7
S9525_Fx8	L	AB	AB-2	2	8
S9525_Fx9	L	AB	AB-2	2	9
S9525_Fx10	L	AB	AB-2	2	10
S9526_Fx1	L	AB	AB-3	3	1
S9526_Fx2	L	AB	AB-3	3	2
S9526_Fx3	L	AB	AB-3	3	3
S9526_Fx4	L	AB	AB-3	3	4
S9526_Fx5	L	AB	AB-3	3	5
S9526_Fx6	L	AB	AB-3	3	6
S9526_Fx7	L	AB	AB-3	3	7
S9526_Fx8	L	AB	AB-3	3	8
S9526_Fx9	L	AB	AB-3	3	9
S9526_Fx10	L	AB	AB-3	3	10

2. Enable *fractions* in the configuration file as follow:

```
fractions:
  enabled : 1 # 1 for protein fractions, 0 otherwise
```

Special case: SILAC

One of the most widely used techniques that enable relative protein quantification is *stable isotope labeling by amino acids in cell culture* (SILAC). The keys file can capture the typical SILAC experiment. The following example shows a SILAC experiment with two conditions, two biological replicates, and two technical

replicates:

RawFile	IsotopeLabelType	Condition	BioReplicate	Run
QE20140321-01	H	iso	iso-1	1
QE20140321-02	H	iso	iso-1	2
QE20140321-04	L	iso	iso-2	3
QE20140321-05	L	iso	iso-2	4
QE20140321-01	L	iso_M	iso_M-1	1
QE20140321-02	L	iso_M	iso_M-1	2
QE20140321-04	H	iso_M	iso_M-2	3
QE20140321-05	H	iso_M	iso_M-2	4

It is also required to activate the *silac* option in the yaml file as follows:

```
silac:
  enabled : 1 # 1 for SILAC experiments
```

Quality Control Analysis

Three functions are available to perform QC analyses. For illustrative purposes, an example dataset consisting of a reduced version of two head and neck cancer cell lines (conditions `"Cal33"` and `"HSC6"`), 2 biological replicates each. The number of peptides was reduced to 1/5 due to bioconductor limitations on data size.

- Evidence file: `artms_data_ph_evidence`
- Keys file: `artms_data_ph_keys`

The full data set (2 conditions, 4 biological replicates) can be found at the following urls:

```
url_evidence <- 'http://kroganlab.ucsf.edu/artms/ph/evidence.txt'
url_keys <- 'http://kroganlab.ucsf.edu/artms/ph/evidence.txt'
```

Basic QC (`evidence.txt`-based)

The basic quality control analysis takes as input both the `evidence.txt` and `keys.txt`.

```
artms_qualityControlEvidenceBasic(evidence_file = artms_data_ph_evidence,
                                  keys_file = artms_data_ph_keys,
                                  output_name = "qcPlots_evidence",
                                  prot_exp = "PH")
```

Running `artms_qualityControlEvidenceBasic()` generates the following `pdf` files:

- **-basicReproducibility.pdf**: correlation dot plot for all the combinations of biological replicates of conditions, based on MS Intensity values using features (peptide+charge)
- **-correlationMatrixBR.pdf**: It contains 3 pages. *Correlation matrix* for all the biological replicates using MS Intensity values, *Clustering matrix* of the MS Intensities and correlation distribution *histogram*.
- **-correlationMatrixBR.pdf**: Same as the previous one, but based on MS Intensity values of Conditions
- **-IntensityDistributions.pdf**: 2 pages. *Box-dot plot* and *Jitter plot* of biological replicates based on MS (raw) intensity values.
- **-intensityStats.pdf**: several pages, including bar plots of *Total Sum of Intensities in BioReplicates*, *Total Sum of Intensities in Conditions*, *Total Peptide Counts in BioReplicates*, *Total Peptide Counts in conditions* separated by categories (`CON`: contaminants, `PROT` peptides, `REV` reversed sequences used by MaxQuant to estimate the FDR); *Box plots* of MS Intensity values per biological replicates and

conditions; *bar plots* of total intensity (excluding contaminants) by bioreplicates and conditions; Barplots of *total feature counts* by bioreplicates and conditions.
- **-ptmStats.pdf**: If any PTM is selected (`PH`, `UB`, `AC`) an extra pdf file will be generated with stats related to the selected modification, including: *bar plot of peptide counts and intensities*, broken by `PTM/other` categories; bar plots of *total sum-up of MS intensity values* by other/PTM categories.

Check `?artms_evidenceQCbasic()` to find out more options about this function.

Extended QC (`evidence.txt`-based)

It takes as input the `evidence.txt` and `keys.txt` files as follows:

```
artms_qualityControlEvidenceExtended(evidence_file = artms_data_ph_evidence,
                                     keys_file = artms_data_ph_keys)
```

It generates the following QC files:

- **QC-ID-Overlap.pdf**
- **QC-IntCorrelation.pdf**
- **QC-SamplePrep.pdf**
- **QC_Plots_CHARGESTATE.pdf**
- **QC_Plots_IONS.pdf**
- **QC_Plots_MASSERROR.pdf**
- **QC_Plots_MZ.pdf**
- **QC_Plots_PepDetect.pdf**
- **QC_Plots_PEPINT.pdf**
- **QC_Plots_PepIonOversampling.pdf**
- **QC_Plots_PEPTIDES.pdf**
- **QC_Plots_ProtDetect.pdf**
- **QC_Plots_PROTEINS.pdf**
- **QC_Plots_ProtInt.pdf**
- **QC_Plots_PSM.pdf**
- **QC_Plots_TYPE.pdf**

Extended QC (`summary.txt` based)

requires two files:

- `keys.txt`
- MaxQuant `summary.txt` file. As described by MaxQuant's `table.pdf`, the summary file contains summary information for all the raw files processed with a single MaxQuant run, including statistics on the peak detection. `artms_qualityControlSummaryExtended()` gathers a quick overview on the quality of every RawFile based on this `summary.txt`.

Run it as follows:

```
artms_qualityControlSummaryExtended(summary_file = "summary.txt",
                                    keys_file = artms_data_ph_keys)
```

It generates the following `pdf` plots:

- **QC_Plots_summary_ISOTOPE.pdf**
- **QC_Plots_summary_MS1SCANS.pdf**
- **QC_Plots_summary_MS2.pdf**
- **QC_Plots_summary_MSMS.pdf**

Relative quantification

The relative quantification is the core of this package. All the information required to run a relative quantification analysis using `MSstats` is provided through a configuration file (`.yaml` format). Check the above to find out more about the different sections of the configuration file.

A template of the configuration file can be generated by running `artms_writeConfigYamlFile()`.

Different types of proteomics experiments can be analyzed such as protein abundance (ab), affinity purification mass spectrometry (apms), and different type of posttranslational modifications, including phosphorylation (ph), ubiquitination (ub), and acetylation (ac)

Quantification of Changes in Protein Abundance

It quantifies changes in protein abundance between two different conditions. These are the specific sections that the user has to filled up:

```
files:
  evidence : /path/to/the/evidence.txt
  keys : /path/to/the/keys.txt
  contrasts : /path/to/the/contrast.txt
  output : /path/to/the/output/results_ptmGlobal/results.txt
  .
  .
  .
data:
  .
  .
  .
  filters:
    modifications : AB
```

Make sure that the filter `modifications` is labeled as `AB`.

Finally, run the following `artMS` function:

```
artms_quantification(yaml_config_file = '/path/to/config/file/artms_ab_config.yaml')
```

Quantification of Changes in Global Phosphorylation / Ubiquitination

The **global phosphorylation / ubiquitination** quantification analysis calculates changes in phosphorylation/ubiquitination at the *protein level*. This means that all the **modified** peptides are used to quantify changes in protein phosphorylation/ubiquitination between different conditions. The **site-specific** (explained next) quantifies changes at the *peptide level*, i.e., each modified peptide independently between the different conditions.

Only two sections need to be filled up on the **default** configuration (`yaml`) file:

```
files:
  evidence : /path/to/the/evidence.txt
  keys : /path/to/the/keys.txt
  contrasts : /path/to/the/contrast.txt
  output : /path/to/the/output/results_ptmGlobal/results.txt
  .
  .
  .
```

```
data:
  .
  .
  .
  filters:
    modifications : PH # Use UB for ubiquination
```

The remaining options can be left unmodified.

Once the configuration `yaml` file is ready, run the following command:

```
artms_quantification(yaml_config_file = '/path/to/config/file/artms_phglobal_config.yaml')
```

Site-specific Quantification of Changes in Phosphorylation / Ubiquitination

The `site-specific` analysis quantifies changes at the modified peptide level. This means that changes in every modified (ph/ub) peptide of a given protein will be quantified individually. The caveat is that the proportion of missing values should increase relative to the **global** analysis. Both **site** and **global** ptm analysis are highly correlated due to the fact that only one or two peptides drive the overall changes in PTMs for every protein.

To run a site specific analysis follow these steps:

1. A pre-processing step is required to be run on the evidence file to enable the site-specific ph analysis.

For phosphorylation

```
artms_proteinToSiteConversion(evidence_file = "/path/to/the/evidence.txt", ref_proteome_file = "/path/t
```

For ubiquitination

```
artms_proteinToSiteConversion(evidence_file = "/path/to/the/evidence.txt", ref_proteome_file = "/path/t
```

2. Generate a new configuration file (`phsites_config.yaml` or `ubsites_config.yaml`) as explained above, but using the "new" `ph-sites-evidence.txt` (or `ub-sites-evidence.txt`) file instead of the original `evidence.txt` file.

Once the new `yaml` file has been created, execute:

```
artms_quantification(yaml_config_file = '/path/to/config/file/phsites_config.yaml')
```

Analysis of Quantifications

Comprehensive analysis of the quantification obtained running `artms_quantification()`. It includes:

- Annotations
- Summary files in different formats (xls, txt) and shapes (long, wide)
- Numerous summary plots
- Enrichment analysis using Gprofiler
- PCA of protein abundance
- PCA of quantification
- Clustering analysis

It takes as input two files generated from the previous quantification step (`artms_quantification()`)

- `-results.txt`
- `-results_modelQC.txt`

To run this analysis

1. Set as the working directory the folder with the results obtained from `artms_quantification()`.

```
setwd('~/path/to/the/results_quantification/')
```

And then run the following function (for an "AB" experiment)

A few comments:

- `isPTM`. For both protein abundance (`ab`), Affinity Purification-Mass Spectrometry (`apms`), and global analysis of posttranslational modifications (PH and UB) analyses use the option `"noptm"`. For a site specific PTM analysis use `"yesptmsites"`.
- `specie`. The host-species supported for now are `"human"` and `"mouse"`
- `isBackground`. If `enrich = TRUE`, the user can provide a background gene file of gene names. Indicate the path to the file in this argument.
- `mnbr`: Minimal Number of Biological Replicates for imputation. Missing values will be imputed and this argument is set to specified the minimal number of biological replicates that are required in at least one of the conditions, but for all the comparisons. For example, `mnbr = 2` would mean that only proteins found in *at least* two biological replicates will be imputed. In addition, for any protein, in at least one of the conditions the protein should be identified at least in two biological replicates or it will be removed.
- `l2fc_thres` is the log2fc cutoff for enrichment analysis, absolute value, i.e., if it is set to 1, it will consider significant log2fc$>+1$ and log2fc $<$ -1.
- `ipval`: specify whether `pvalue` or `adjpvalue` should use for the analysis. The default option is `adjpvalue` (multiple testing correction). But if the number of biological replicates for a given experiment is too low (for example n = 2), then `pvalue` is recommended.

Miscellaneous: a little bit of everything

`artMS` also provides a number of very handy functions.

```
artms_avg_intensity_RT(evidence_file = '/path/to/the/evidence.txt)
```

Taking as input the evidence file location, it will summarize and report back the average intensity, average retention time, and the average caliberated retention time for each protein. If a list of proteins is provided, then only those proteins will be summarized and returned. Check `?artms_avg_intensity_RT()` to find out more options.

it generates reproducibility plots, as specificied by the user. It requires an additional file with the following information:

condition1	**rep1__1**	**rep1__2**	**condition2**	**rep2__1**	**rep2__2**
Cal33	Cal33-1	Cal33-2	HSC6	HSC6-1	HSC6-2
Cal33	Cal33-3	Cal33-4	HSC6	HSC6-3	HSC6-4

Help

- Errors or warnings? Please, submit them as a new issue at the official github repository
- Any other inquiries: artms.help@gmail.com