MSstats: Next Generation Statistical Mass Spectrometry in R

Mateusz Staniak
Olga Vitek Lab

http://msstats.org/

MSstats

Statistical Tool For Quantitative Mass Spectrometry-Based Proteomics

Background for MSstats development

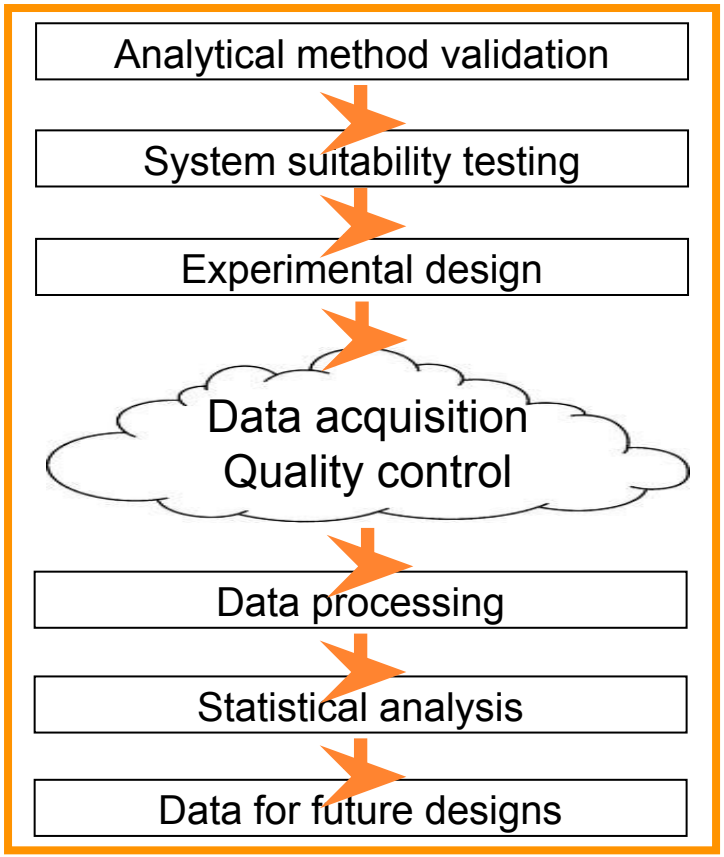
1. Computational efficiency of the analysis:
    a. more proteins,
    b. more samples,
    c. larger files.
2. Diversity of analyses: facilitate interoperability with various signal processing tools.
3. Reproducible research, documentation and sharing of the analyses.
4. New experimental workflows, for example PTM data.
    a. Enabled by modular interface.
5. Facilitate new developments.

Benefits of the new MSstats developments

1. With the new design, it is even easier to connect `MSstats` with signal processing tools to create larger MS data analysis pipelines.
2. The addition of automated testing infrastructure is an opportunity for development of optimized pipelines.
3. With the new abstraction for data preprocessing (`MSstatsConvert` package), `MSstats` is open for more opinionated approach to filtering and FDR, shared peptides, etc.

What is MSstats?

The MSstats ecosystem



Assay characterization
LOB/LOD

MSstatsQC
Software package
System suitability monitoring

Experimental design
Sample size for testing and classification

R software package	*Open source software*	*Commercial software*
MSnbase	**Skyline, MaxQuant**	**Proteome Discoverer**
RforProteomics	**OpenMS, OpenSWATH**	**Spectronaut, SpectroMine**
	DIA-Umpire	**Progenesis**

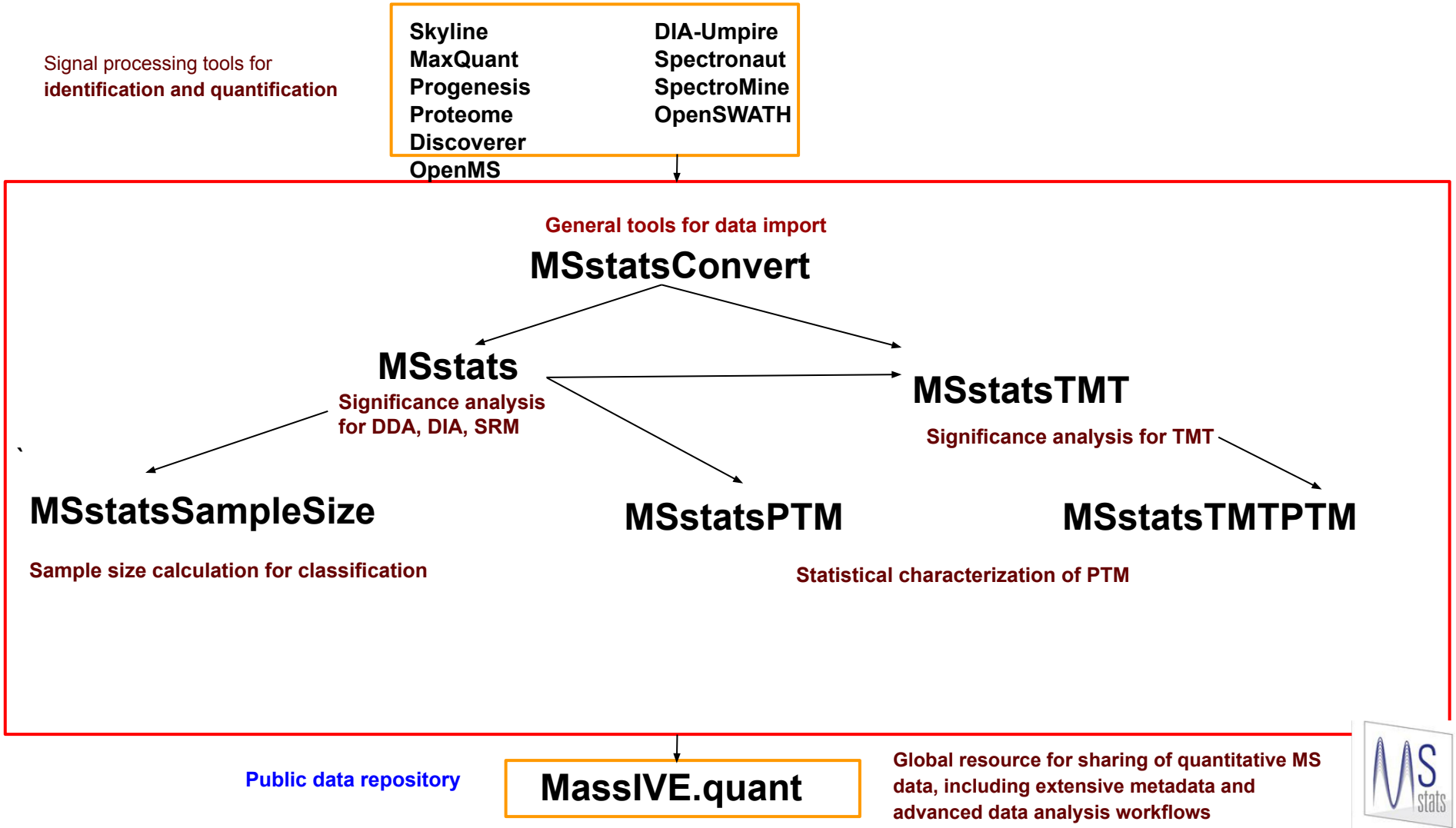
MSstats
Software package
Significance analysis for DDA, SRM, DIA

MSstatsTMT
Software package
Significance analysis for TMT

MSstatsBioData
Experiment package
Published studies with DDA or SRM

42 datasets, > 178 reanalyses

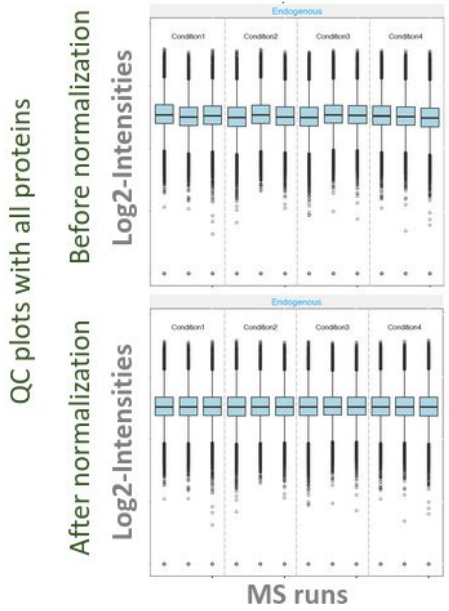
MassIVE
Mass Spectrometry Interactive Virtual Environment


Signal processing tools for
identification and quantification
Skyline
MaxQuant
Progenesis
Proteome
Discoverer
OpenMS
DIA-Umpire
Spectronaut
SpectroMine
OpenSWATH
General tools for data import
MSstatsConvert
MSstats
Significance analysis
for DDA, DIA, SRM
MSstatsTMT
Significance analysis for TMT
MSstatsSampleSize
Sample size calculation for classification
MSstatsPTM
MSstatsTMTPTM
Statistical characterization of PTM
Public data repository
MassIVE.quant
Global resource for sharing of quantitative MS
data, including extensive metadata and
advanced data analysis workflows

R packages in the ecosystem

- **MSstats**: for label-free and SRM experimenets (DDA and DIA),
- **MSstatsTMT**: for TMT data,
- **MSstatsPTM** and **MSstatsTMTPTM**: for experiments focused on post-translational modifications,
- **MSstatsConvert**: for converting data from any signal processing tool into a consistent format,
- **MSstatsQC**: quality control tools,
- **MSstatsSampleSize**: sample size simulations,
- **MSstatsBioData**: MS datasets.

MSstats workflow

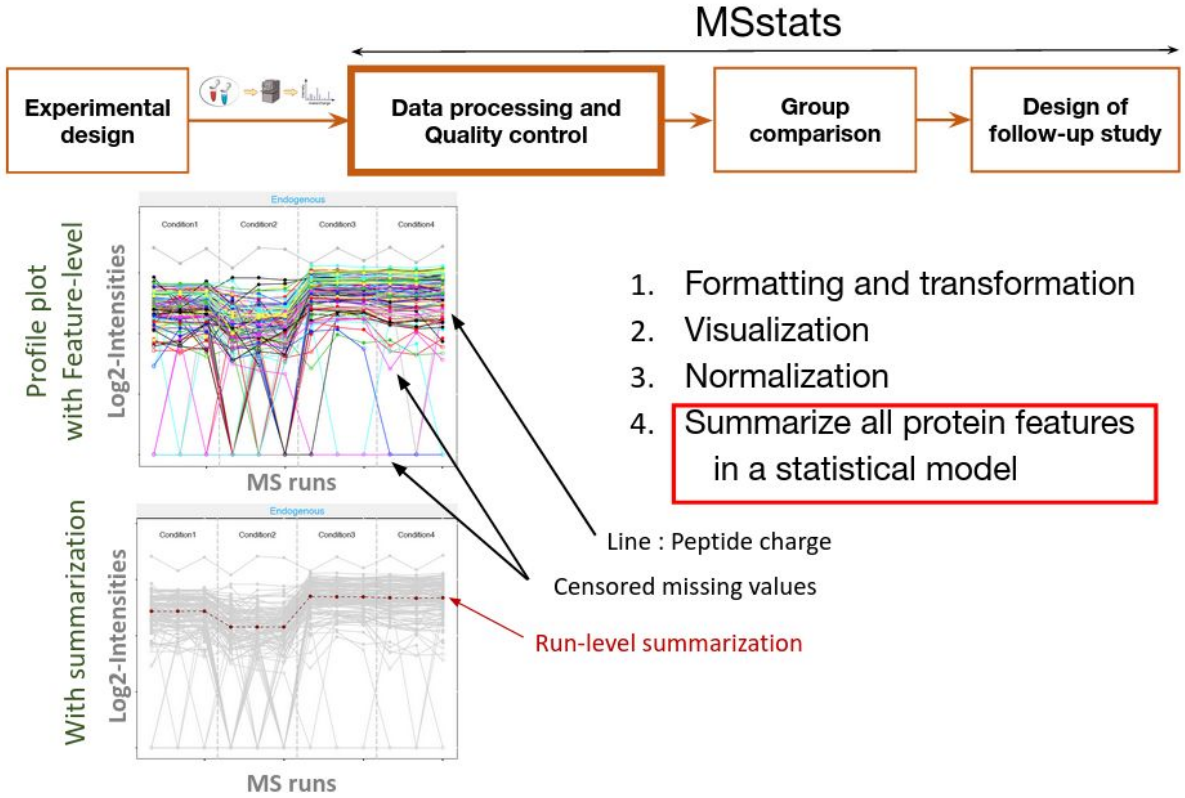


1. Formatting and transformation
2. Visualization
3. Normalization
4. Summarize all protein features in a statistical model

Same medians across MS runs

Author: Meena Choi

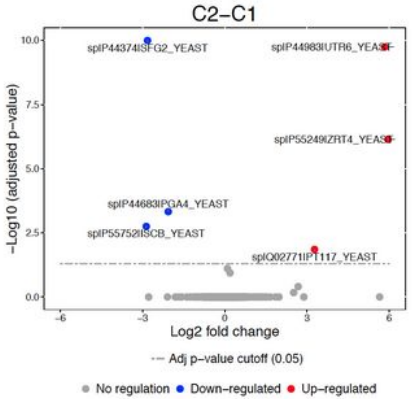
MSstats workflow



Author: Meena Choi

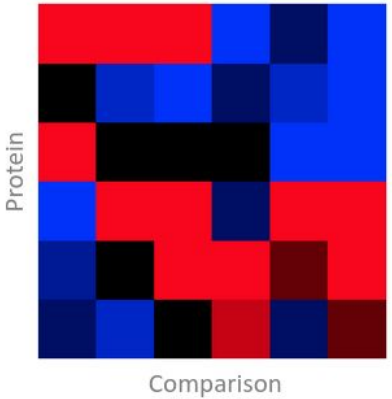
MSstats workflow

Volcano plot



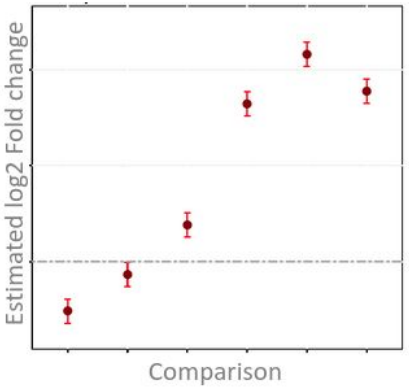
- Per comparison
- All proteins
- Adjusted p-value and log fold change

Heatmap



- With all comparisons
- All proteins
- Adjusted p-value and cut-off log fold change

Comparison plot



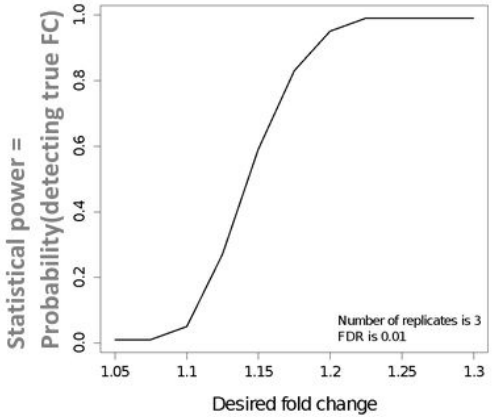
- With all comparisons
- Per protein
- Log fold change and CI

Author: Meena Choi

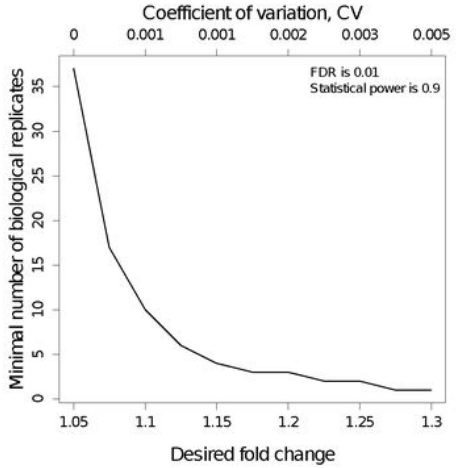
MSstats workflow

Statistical power estimation and sample size calculation

Power of an analysis at any desired fold change with a fixed number of biological replicates and FDR



Minimal number of biological replicates required to observe a desired fold change at a fixed statistical power and FDR



Author: Meena Choi

What’s new in MSstats?

Modular and general interface enables working with any signal processing tool

MSstats v 3.22, MSstatsTMT v 1.8:

- `SkylinetoMSstatsFormatFormat(...)`
- `OpenSWATHtoMSstatsFormat(...)`
- `OpenMStoMSstatsTMTFormat(...)`

(and other converters)

Newest MSstats and MSstatsTMT:

- `MSstatsImport(...)`
- `MSstatsClean(...)`
- `MSstatsPreprocess(...)`

-> common framework for all existing converters + possible new converters

New interface enables all preprocessing steps required before statistical analysis

`MSstatsImport()`,

`MSstatsClean()`:

-> convenient wrappers for multi-file inputs to reduce any set of files into a single table (almost) in the MSstats format.

Core function:

```
MSstatsPreprocess(
        input, annotation, feature_columns,
        remove_shared_peptides,
        remove_single_feature_proteins,
        feature_cleaning, score_filtering,
        exact_filtering, pattern_filtering,
        columns_to_fill, aggregate_isotopic
        …
)
```

-> allows fast and flexible implementation of new or custom converters.

Modular design enables using implementations from MSstats and other sources

Old interface

```
MSstats::dataProcess(
    raw = sl, normalization = "equalizeMedians",
    summaryMethod = "TMP", censoredInt = "0")
```

New interface

```
sl = SkylinetoMSstatsFormat(raw_input)
input = MSstatsPrepareForDataProcess(sl, 2,
                                     FALSE, FALSE)
input = MSstatsNormalize(input,
                         "EQUALIZEMEDIANS")
input = MSstatsMergeFractions(input)
input = MSstatsHandleMissing(input, "TMP",
                             TRUE, "0", 0.999)
summary = MSstatsSummarize(input,
                           censoredInt = "0")
```

Modular design enables flexible logging for reproducible research

Flexible logging system based on the log4r package:

```
library(MSstatsConvert)
# default - creates a new file
MSstatsLogsSettings(use_log_file = TRUE, append = FALSE)

# default - creates a new file
MSstatsLogsSettings(use_log_file = TRUE, append = TRUE,
                    log_file_path = "log_file.log")

# switches logging off
MSstatsLogsSettings(use_log_file = FALSE, append = FALSE)

# switches off logs and messages
MSstatsLogsSettings(use_log_file = FALSE, verbose = FALSE)
```

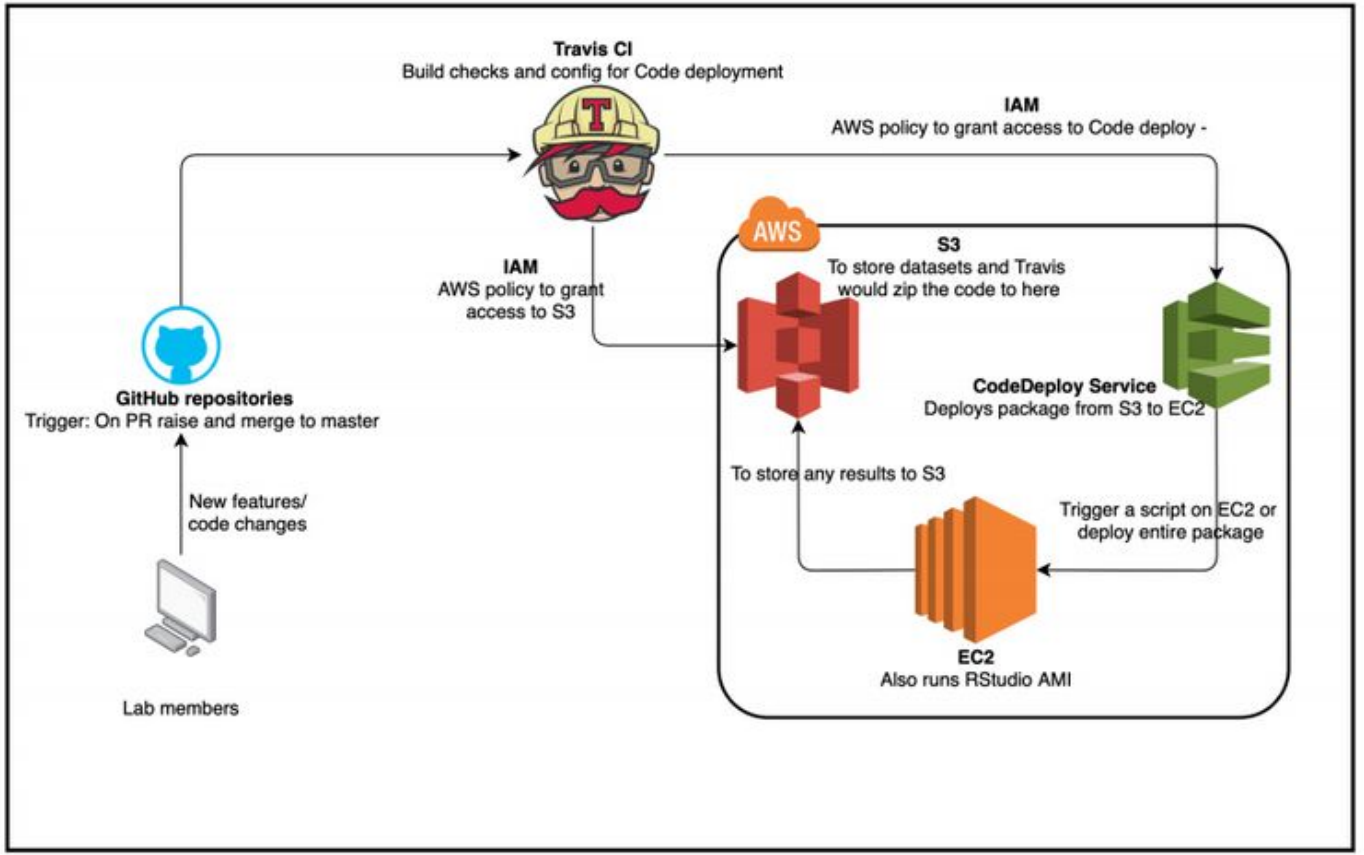
Logging example

```
INFO  [2020-10-28 12:57:47] ** Raw data from Skyline imported successfully.
INFO  [2020-10-28 12:57:48] ** Raw data from Skyline cleaned successfully.
INFO  [2020-10-28 12:57:48] ** Using annotation extracted from quantification data.
INFO  [2020-10-28 12:57:48] ** Run labels were standardized to remove symbols such as '.' or '%'.
INFO  [2020-10-28 12:57:48] ** The following options are used:
  - Features will be defined by the columns: PeptideSequence, PrecursorCharge, FragmentIon, ProductCharge
  - Shared peptides will be removed.
  - Proteins with single feature will not be removed.
  - Features with less than 3 measurements across runs will be removed.
INFO  [2020-10-28 12:57:48] ** Rows with values of StandardType equal to iRT are removed
INFO  [2020-10-28 12:57:48] ** Intensities with values of Truncated equal to TRUE are replaced with NA
INFO  [2020-10-28 12:57:48] ** Intensities with values smaller than 0.01 in DetectionQValue are replaced with 0
INFO  [2020-10-28 12:57:48] ** Sequences containing DECOY, Decoys are removed.
INFO  [2020-10-28 12:57:48] ** Features with all missing measurements across runs are removed.
INFO  [2020-10-28 12:57:48] ** Shared peptides are removed.
INFO  [2020-10-28 12:57:48] ** Multiple measurements in a feature and a run are summarized by summaryforMultipleRows:
sum
INFO  [2020-10-28 12:57:48] ** Features with one or two measurements across runs are removed.
INFO  [2020-10-28 12:57:48] ** Run annotation merged with quantification data.
INFO  [2020-10-28 12:57:48] ** Features with one or two measurements across runs are removed.
INFO  [2020-10-28 12:57:48] ** Fractionation handled.
INFO  [2020-10-28 12:57:48] ** Updated quantification data to make balanced design. Missing values are marked by NA
INFO  [2020-10-28 12:57:48]

INFO  [2020-10-28 12:57:48] ** Finished preprocessing. The dataset is ready to be processed by the dataProcess
function.
INFO  [2020-10-28 12:57:48]
```

Complex workflows require automated testing infrastructure

- refactoring code and adding new features requires testing,
- it's hard or impossible for unit tests to cover all cases,
- we tested all our updates on >30 datasets,
- testing with large datasets is extremely time-consuming:
  - some operations may take several hours on a single dataset,
  - running time does not only depend on our code (for example, due to use of model-fitting functions),
  - bugs require manual checking, fixes require re-running the checks.
- to save time and make testing effortless, we developed an automated, cloud-based infrastructure



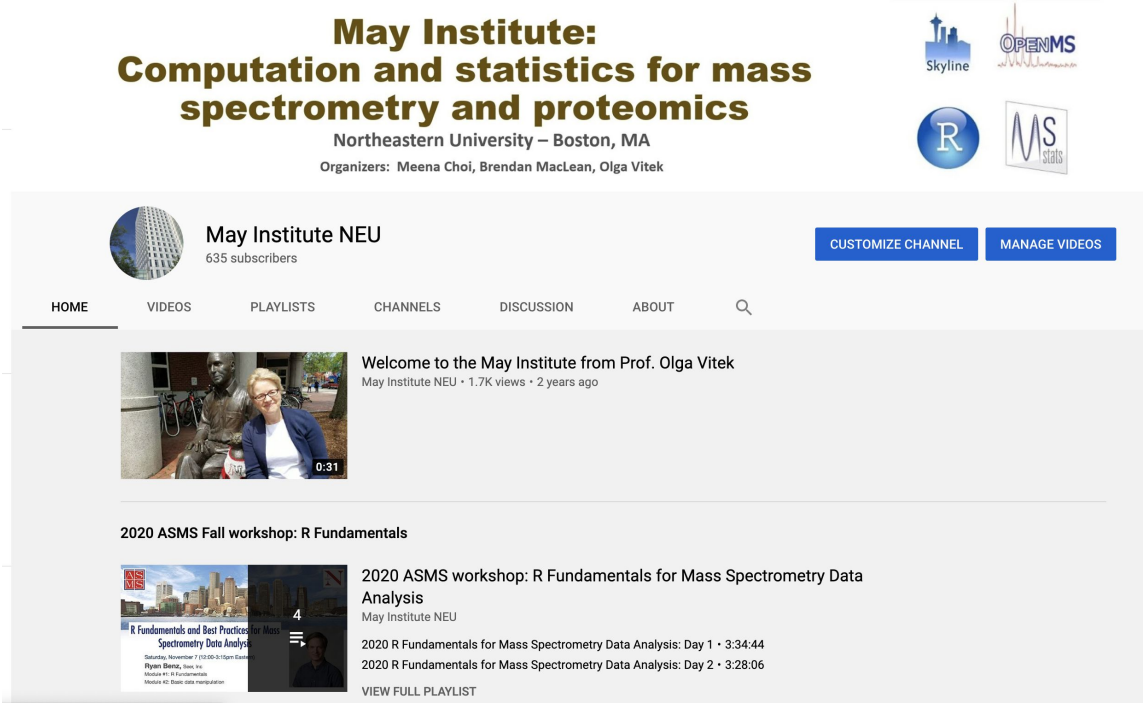
Author: Ajeya Kempegowda

Fantastic MSstats Packages and Where to Find Them

- Vitek Lab @ Github: https://github.com/Vitek-Lab/
- Official website and documentation: http://msstats.org/
- Main package: https://github.com/MeenaChoi/MSstats/
- Google group: https://groups.google.com/g/msstats
- Bioconductor: MSstatsConvert, MSstats, MSstatsTMT, MSstatsPTM
- MassIVE.quant: https://massive.ucsd.edu/ProteoSAFe/static/massive-quant.jsp

+ Cardinal MSI – A mass spectrometry imaging toolbox for statistical analysis

MSstats training



MSstats Dev Team
(recent contributions)

- Meena Choi (lead)
- Ting Huang (MSstatsTMT)
- Mateusz Staniak (MSstatsConvert, MSstats + MSstatstMT)
- Tsung-Heng Tsai (MSstats, MSstatsPTM)
- Devon Kohler (MSstatsPTM)
- Ajeya Kempegowda (testing infrastructure)
- Dhaval Mohandas (Shiny GUI)

Thank you for your attention!