POMA

An User-friendly Workflow for Pre-processing and Statistical Analysis of Mass Spectrometry Data

European *Bioconductor* Meeting 2020

Pol Castellano-Escuder
University of Barcelona
Dec 18, 2020

Outline

1) Scope of the package

2) POMA workflow (with examples)

    2.1) Data formatting

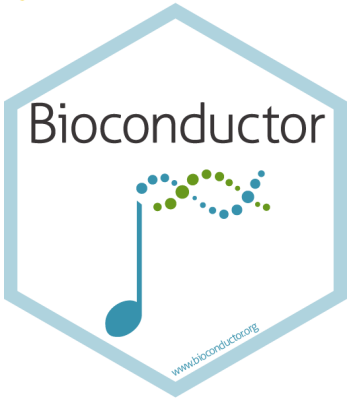
    2.2) Pre-processing

    2.3) Exploratory Data Analysis (EDA)

    2.4) Statistical Analysis

3) Conclusions

4) Next steps...



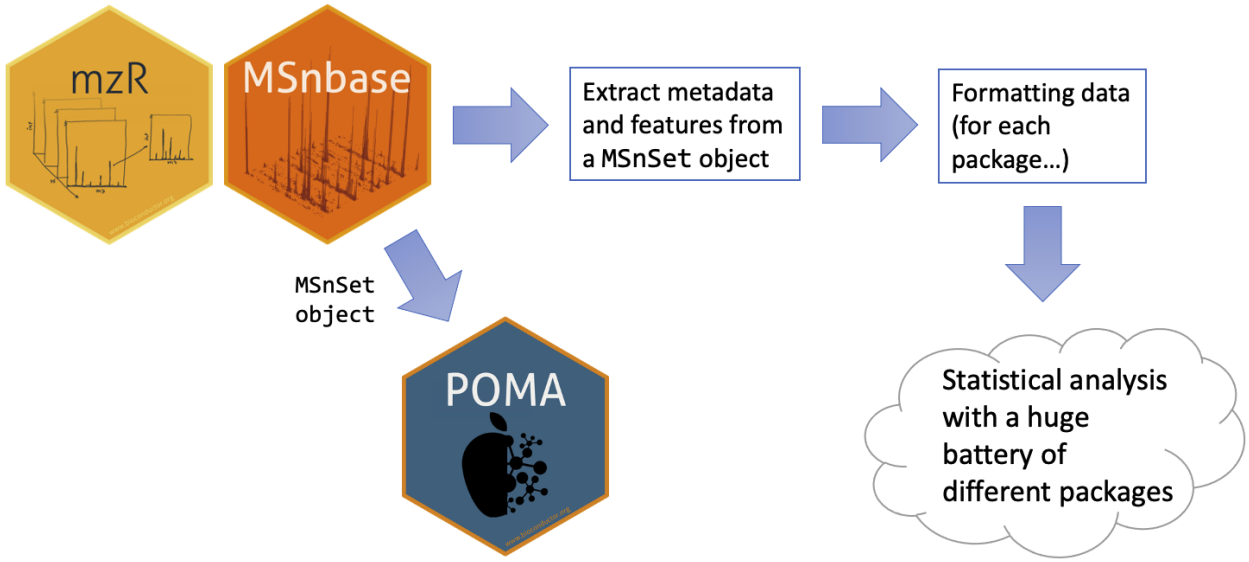
Scope of POMA

`POMA` package focuses on **statistical analysis** of metabolomics and proteomics

1) Directly from sheet (xlsx, csv, etc.) ❌❌

2) Extracting quantitative data from `MSnbase::MSnSet` objects ❌

3) Using `MSnbase::MSnSet` objects directly (++ reproducibility and interoperability) ✅



POMA workflow: 1) Data formatting

The input format required in all `POMA` functions in a `MSnbase::MSnSet` object

Target or "*phenotype data*"

ID	Group	age_at_consent	gender	smoking_condition	alcohol_consumption
sample_109	CRC	45	1	0	2
sample_11	CRC	56	1	1	1
sample_115	CRC	53	0	0	2
sample_12	CRC	55	1	0	2
sample_122	CRC	55	0	1	1
sample_13	CRC	29	1	1	1
sample_132	CRC	51	0	1	0
sample_133	CRC	76	1	0	2
sample_139	CRC	66	0	1	1
sample_14	CRC	46	0	1	1

Features or "*expression data*"

x1_methyladenosine	x1_methylhistamine	x2_aminoadipate	x2_deoxyuridine	x3_nitro_tyrosi
390953.972	34627.336	141257.364	13115.813	92919
567984.436	52845.928	528024.686	14327.696	8453
558842.755	31507.343	188272.302	12756.865	82712
476949	29397	107076.092	10539	84197
398317	38877	158299.071	11689	89298
411224	25493	136600.498	12425	90133
412250.639	44478.185	235936.477	15692.631	89317
414501.111	27448.735	419826.634	15743.581	83879
416166	34757	351043.702	8418	91736
384549	26494	124681.876	9693	70050

```
# create MSnSet object from target and features data frames
msnset_object <- PomaMSnSetClass(tatget = "target",
                                 features = "features")
```

POMA workflow: 2) Pre-processing

All `POMA` pre-processing methods return a pre-processed `MSnSet` object

2.1) Missing value imputation

```
msnset_object %>%
  PomaImpute(ZerosAsNA = FALSE, # treat zeros as zeros instead of NAs
             RemoveNA = TRUE, cutoff = 20, # remove columns with >20% of NA
             method = "knn") # k-nearest neighbors (default)
```

2.2) Normalization

```
msnset_object %>%
  PomaNorm(method = "log_pareto") # log Pareto scaling (default)
```

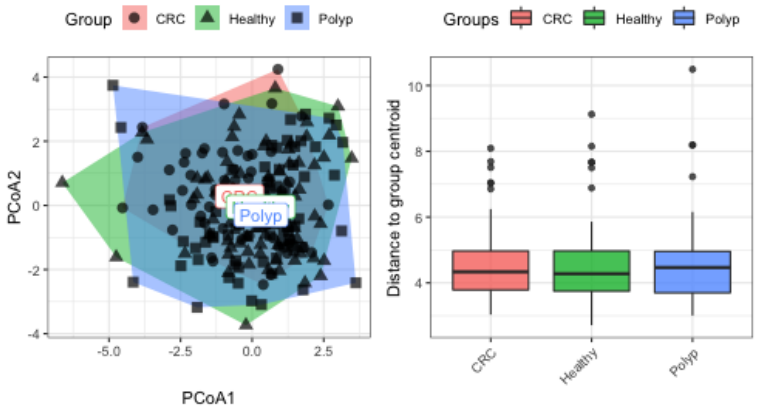
POMA workflow: 2) Pre-processing

2.3) Outlier detection and cleaning

Detection:

```
msnset_object %>%
    PomaOutliers(do = "analyze")
```

Output:



Clean:

```
msnset_object %>%
    PomaOutliers(do = "clean")
```

Output:

A clean `MSnSet` object (without sample outliers)

POMA workflow: 2) Pre-processing

Pre-processing output:

```
MSnSet (storageMode: lockedEnvironment)
assayData: 113 features, 208 samples
  element names: exprs
protocolData: none
phenoData
  sampleNames: sample_109 sample_11 ... sample_99 (208 total)
  varLabels: Group age_at_consent ... alcohol_consumption (5 total)
  varMetadata: labelDescription
featureData: none
experimentData: use 'experimentData(object)'
Annotation:
- - - Processing information - - -
Imputed (knn): Mon Dec 14 22:07:24 2020
Normalised (log_pareto): Mon Dec 14 22:07:25 2020
Outliers removed (euclidean and median): Mon Dec 14 22:07:25 2020
 MSnbase version: 2.16.0
```

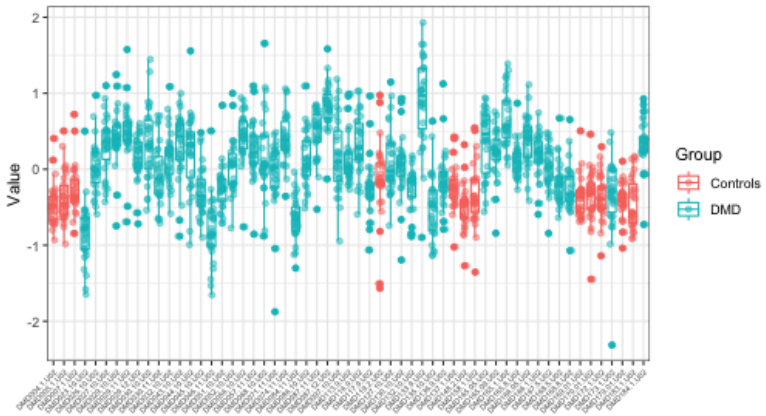
POMA workflow: 3) EDA

Include several flexible visualization options such as boxplots, density plots, heatmaps, etc.

Boxplot examples:

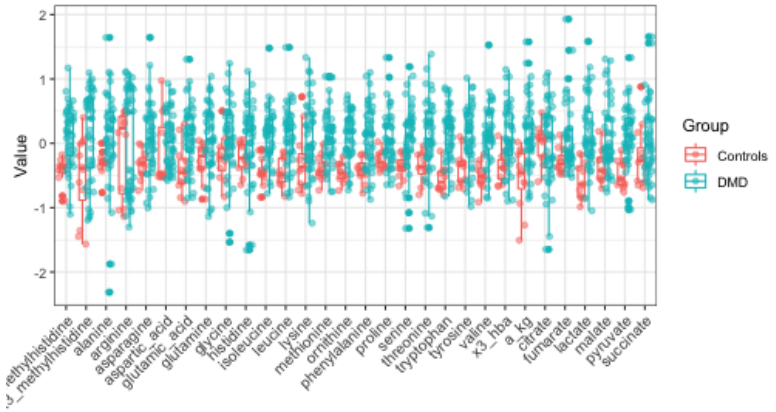
Display samples:

```
clean_object %>%
  PomaBoxplots(group = "samples",
               label_size = 5)
```



Display features:

```
clean_object %>%
  PomaBoxplots(group = "features")
```

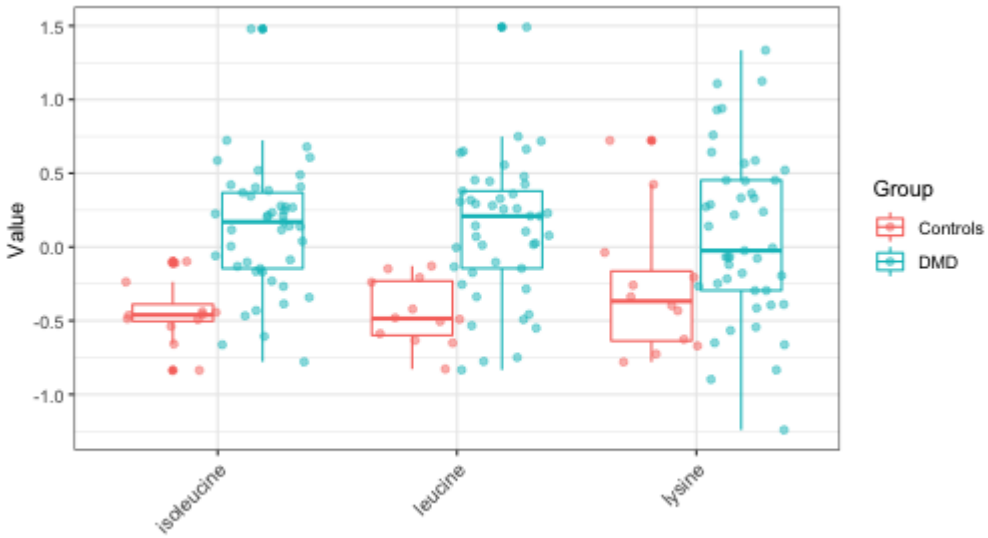


POMA workflow: 3) EDA

Boxplot examples:

Display features of interest:

```
clean_object %>%
  PomaBoxplots(group = "features",
               feature_name = c("isoleucine", "leucine", "lysine"))
```



POMA workflow: 4) Statistical analysis

Statistical methods covered by `POMA`:

- **Univariate analysis:** T-test, ANOVA, ANCOVA, Wilcoxon test, and Kruskal Wallis
- **Multivariate analysis:** PCA, PLS-DA, and sPLS-DA
- **Cluster analysis:** *k*-means (clusters projected in a MDS plot)
- **Limma:** Both designs with and without covariates (from `limma` *Bioconductor* package)
- **Correlation analysis:** Pairwise correlations, correlogram, network correlations and Gaussian Graphical Models
- **Regularization methods:** LASSO, Ridge, and Elasticnet (**allow train/test split**)
- **Random forest:** Classical Random forest algorithm (**allow train/test split**)
- **Rank products:** From `RankProd` *Bioconductor* package
- **Odds ratios:** Based on a logistic regression model (two-group analysis)

POMA workflow: 4) Statistical analysis

All `POMA` functions are focused on **simplifying and compacting** the analyses, grouping different methodologies of the same class within a single function instead of maintaining many "single purpose" functions

Univariate analysis: 1 function, 4 methods

```
clean_object %>%

# parametric methods
PomaUnivariate(method = "ttest")

PomaUnivariate(method = "anova")

# non-parametric methods
PomaUnivariate(method = "mann")

PomaUnivariate(method = "kruskal")
```

Multivariate analysis: 1 function, 3 methods

```
clean_object %>%

PomaMultivariate(method = "pca")

PomaMultivariate(method = "plsda")

PomaMultivariate(method = "splsda")
```

POMA workflow: 4) Statistical analysis

`POMA` predictive features allow users to split data into random *train* and *test* sets in order to perform an **external cross-validation** (CV)

LASSO for **feature selection**: All data used to create the model and no *test* set created

```
clean_object %>%
  PomaLasso(alpha = 1, # LASSO
            ntest = NULL)
```

Ridge regression for **predictive modeling**: External CV using the 20% of the data as *test* set

```
clean_object %>%
  PomaLasso(alpha = 0, # Ridge
            ntest = 20)
```

Random forest for **predictive modeling**: External CV using the 20% of the data as *test* set

```
clean_object %>%
  PomaRandForest(ntree = 500,
                 ntest = 20)
```

	CRC	Healthy	Polyp	class.error
CRC	10	7	0	0.4118
Healthy	2	5	4	0.5455
Polyp	0	12	4	0.7500

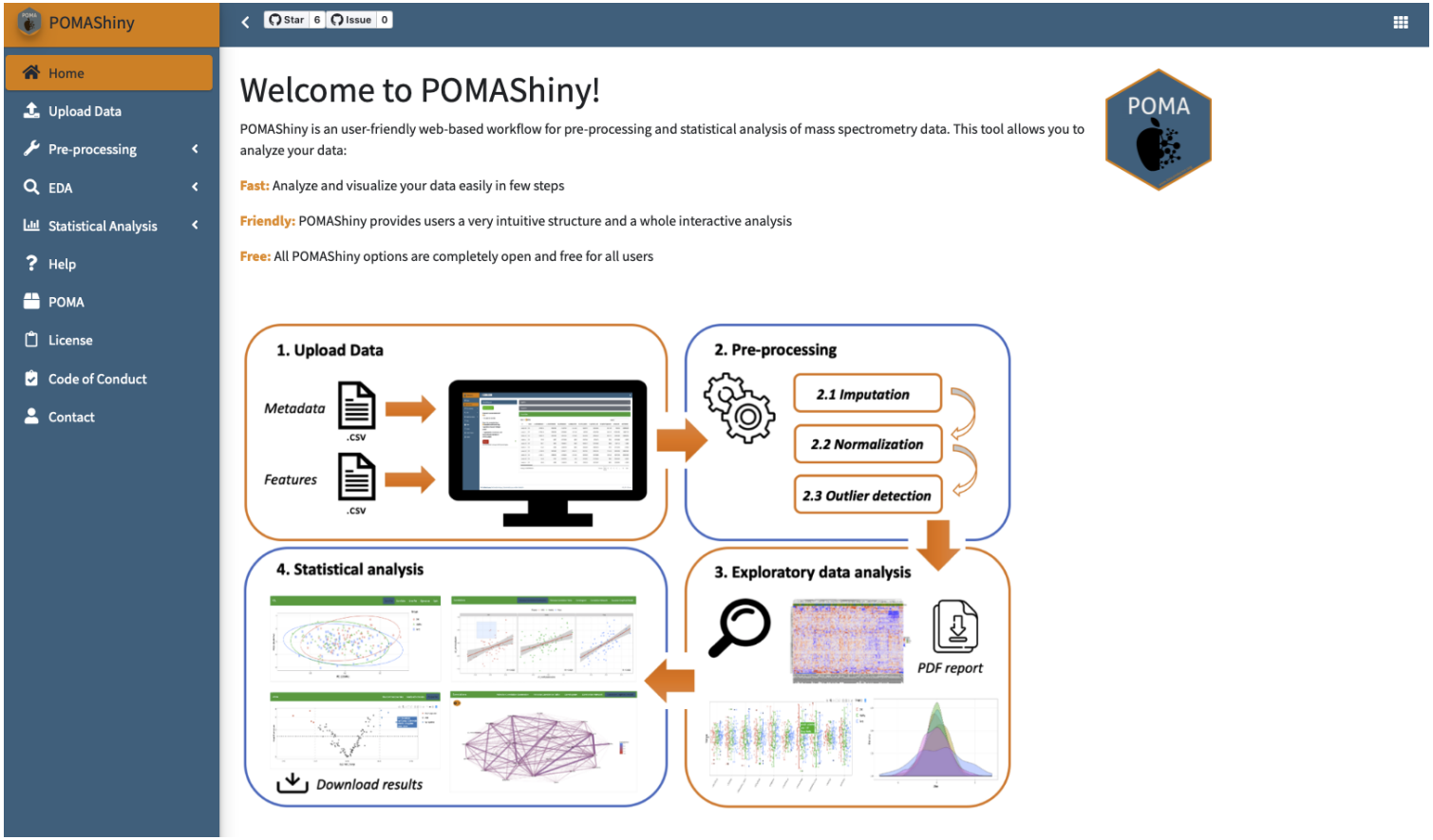
POMA workflow: 4) Statistical analysis

"Top-down" example: From raw data to *limma* model adjusted for covariates

```
raw_MSnSet_data %>%
  PomaImpute(method = "knn") %>%
  PomaNorm(method = "log_pareto") %>%
  PomaOutliers(do = "clean") %>%
  PomaLimma(contrast = "CRC-Polyp", covariates = TRUE)
```

	logFC	AveExpr	t	P.Value	adj.P.Val	B
lysine	-0.2669858	0.0132740	-5.561896	1.0e-07	0.0000093	7.611024
linolenic_acid	-0.3703412	-0.0038798	-5.367994	2.0e-07	0.0000121	6.701319
glyceraldehyde	0.3143264	0.0020721	5.127753	7.0e-07	0.0000254	5.607997
dimethylglycine	-0.3455135	0.0181250	-5.045950	1.0e-06	0.0000279	5.244478
methionine	-0.2224866	0.0160721	-4.693964	4.9e-06	0.0001103	3.732799

POMA Shiny version



https://github.com/pcastellanoescuder/POMAShiny

https://webapps.nutrimetabolomics.com/POMAShiny

Conclusions

- POMA provides a robust, reproducible, and user-friendly workflow for the statistical analysis of mass spectrometry data
- POMA allows users to include different covariates in the analysis
- POMA also provides its own interactive Shiny version called POMAShiny
- POMA is an open source tool and everybody is welcome to contribute!

Next steps...

- Implementation of more functions and methods focusing mainly on multivariate approaches
- Explore the feasibility of a new *Bioconductor* class to store the statistical analysis results, or extend an existing *Bioconductor* class for this purpose (community feedback and collaborations are indispensable)
- Gradually migrate the `MSnbase::MSnSet` structures used by `POMA` to the `QFeatures` structures for mass spectrometry assays included in `QFeatures` package

Thank you all and welcome to contribute!

Statistics and Bioinformatics Research Group and Biomarkers and Nutritional & Food Metabolomics Research Group from University of Barcelona

Slides available at https://github.com/pcastellanoescuder/POMA_slides_EuroBioc2020

---

polcaes@gmail.com
pcastellanoescuder.github.io
@polcastellano_
@pcastellanoescuder
University of Barcelona