

# Package ‘msImpute’

March 30, 2021

**Type** Package

**Title** Imputation of label-free mass spectrometry peptides

**Version** 1.0.0

**Description** MsImpute is a package for imputation of peptide intensity in proteomics experiments. It additionally contains tools for MAR/MNAR diagnosis and assessment of distortions to the probability distribution of the data post imputation. Currently, msImpute completes missing values by low-rank approximation of the underlying data matrix.

**Depends** R (>= 4.0)

**SystemRequirements** python

**Imports** softImpute, methods, stats, graphics, pdist, reticulate, scan, data.table, FNN, matrixStats, rdetools, limma, mvtnorm

**License** GPL (>=2)

**Encoding** UTF-8

**LazyData** true

**BugReports** <https://github.com/DavisLaboratory/msImpute/issues>

**RoxygenNote** 7.1.1

**Suggests** BiocStyle, knitr, rmarkdown, ComplexHeatmap, imputeLCMD

**VignetteBuilder** knitr

**biocViews** MassSpectrometry, Proteomics, Software

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

**git\_branch** RELEASE\_3\_12

**git\_last\_commit** cf79c8f

**git\_last\_commit\_date** 2020-10-27

**Date/Publication** 2021-03-29

**Author** Soroor Hedyeh-zadeh [aut, cre]  
(<<https://orcid.org/0000-0001-7513-6779>>)

**Maintainer** Soroor Hedyeh-zadeh <[hedyehzadeh.s@wehi.edu.au](mailto:hedyehzadeh.s@wehi.edu.au)>

**R topics documented:**

computeStructuralMetrics . . . . .	2
CPD . . . . .	3
findVariableFeatures . . . . .	4
KNC . . . . .	5
KNN . . . . .	5
msImpute . . . . .	6
plotCV2 . . . . .	8
pxd007959 . . . . .	9
pxd010943 . . . . .	10
pxd014777 . . . . .	10
scaleData . . . . .	11
selectFeatures . . . . .	12

<b>Index</b>	<b>15</b>
--------------	-----------

---

computeStructuralMetrics

*Metrics for the assessment of post-imputation structural preservation*

---

**Description**

For an imputed dataset, it computes within phenotype/experimental condition similarity (i.e. preservation of local structures), between phenotype distances (preservation of global structures), and the Gromov-Wasserstein (GW) distance between original (source) and imputed data.

**Usage**

```
computeStructuralMetrics(x, group = NULL, y = NULL, k = 2)
```

**Arguments**

x	numeric matrix. An imputed data matrix of log-intensity.
group	factor. A vector of biological groups, experimental conditions or phenotypes (e.g. control, treatment).
y	numeric matrix. The source data (i.e. the original log-intensity matrix), preferably subsetted on highly variable peptides (see findVariableFeatures).
k	numeric. Number of Principal Components used to compute the GW distance. default to 2.

**Details**

For each group of experimental conditions (e.g. treatment and control), the group centroid is calculated as the average of observed peptide intensities. Withinness for each group is computed as sum of the squared distances between samples in that group and the group centroid. Betweenness is computed as sum of the squared distances between group centroids. When comparing imputation approaches, the optimal imputation strategy should minimize the within group distances, hence smaller withinness, and maximizes between group distances, hence larger betweenness. The GW metric considers preservation of both local and global structures simultaneously. A small GW distance suggests that imputation has introduced small distortions to global and local structures overall,

whereas a large distance implies significant distortions. When comparing two or more imputation methods, the optimal method is the method with smallest GW distance. The GW distance is computed on Principal Components (PCs) of the source and imputed data, instead of peptides. Principal components capture the geometry of the data, hence GW computed on PCs is a better measure of preservation of local and global structures. The PCs in the source data are recommended to be computed on peptides with high biological variance. Hence, users are recommended to subset the source data only on highly variable peptides (hvp) (see `findVariableFeatures`). Since the hvp peptides have high biological variance, they are likely to have enough information to discriminate samples from different experimental groups. Hence, PCs computed on those peptides should be representative of the original source data with missing values. If the samples cluster by experimental group in the first couple of PCs, then a choice of  $k=2$  is reasonable. If the desired separation/clustering of samples occurs in later PCs (i.e. the first few PCs are dominated by batches or unwanted variability), then it is recommended to use a larger number of PCs to compute the GW metric. If you are interested in how well the imputed data represent the original data in all possible dimensions, then set  $k$  to the number of samples in the data (i.e. the number of columns in the intensity matrix). GW distance estimation requires python. See example. All metrics are on log scale.

### Value

list of three metrics: withinness (sum of squared distances within a phenotype group), betweenness (sum of squared distances between the phenotypes), and gromov-wasserstein distance (if `xna` is not NULL). if `group` is NULL only the GW distance is returned. All metrics are on log scale.

### References

Hediyeh-zadeh, S., Webb, A. I., & Davis, M. J. (2020). MSImpute: Imputation of label-free mass spectrometry peptides by low-rank approximation. *bioRxiv*.

### Examples

```
data(pxd010943)
y <- log2(data.matrix(pxd010943))
y <- y[complete.cases(y),]
group <- as.factor(gsub("_[1234]", "", colnames(y)))
computeStructuralMetrics(y, group, y=NULL)
```

---

CPD

*CPD*

---

### Description

Spearman correlation between pairwise distances in the original data and imputed data. CPD quantifies preservation of the global structure after imputation. Requires complete datasets - for developers/use in benchmark studies only.

### Usage

```
CPD(xorigin, ximputed)
```

**Arguments**

xorigin            numeric matrix. The original log-intensity data. Can not contain missing values.  
ximputed           numeric matrix. The imputed log-intensity data. Can not contain missing values.

**Value**

numeric

**Examples**

```
data(pxd007959)
y <- pxd007959$y
y <- y[complete.cases(y),]
# for demonstration we use same y for xorigin and ximputed
CPD(y, y)
```

---

findVariableFeatures    *Find highly variable peptides*

---

**Description**

For each peptide, the total variance is decomposed into biological and technical variance using package `scrn`

**Usage**

```
findVariableFeatures(y)
```

**Arguments**

y                    numeric matrix giving log-intensity. Can contain NA values. Peptides with insufficient observations will be ignored.

**Details**

A loess trend is fitted to total sample variances and mean intensities. For each peptide, the biological variance is then computed by subtracting the estimated technical variance from the loess fit from the total sample variance.

**Value**

A data frame where rows are peptides and columns contain estimates of biological and technical variances. Peptides are ordered by biological variance.

**See Also**

`computeStructuralMetrics`

**Examples**

```
data(pxd007959)
findVariableFeatures(data.matrix(log2(pxd007959$y)))
```

---

KNC	<i>k-nearest class means (KNC)</i>
-----	------------------------------------

---

### Description

The fraction of k-nearest class means in the original data that are preserved as k-nearest class means in imputed data. KNC quantifies preservation of the mesoscopic structure after imputation. Requires complete datasets - for developers/use in benchmark studies only.

### Usage

```
KNC(xorigin, ximputed, class, k = 3)
```

### Arguments

xorigin	numeric matrix. The original log-intensity data. Can contain missing values.
ximputed	numeric matrix. The imputed log-intensity data.
class	factor. A vector of length number of columns (samples) in the data specifying the class/label (i.e. experimental group) of each sample.
k	number of nearest class means. default to k=3.

### Value

numeric The proportion of preserved k-nearest class means in imputed data.

### Examples

```
data(pxd007959)
y <- pxd007959$y
y <- y[complete.cases(y),]
# for demonstration we use same y for xorigin and ximputed
KNC(y, y, class = as.factor(pxd007959$samples$group))
```

---

KNN	<i>k-nearest neighbour (KNN)</i>
-----	----------------------------------

---

### Description

The fraction of k-nearest neighbours in the original data that are preserved as k-nearest neighbours in imputed data. KNN quantifies preservation of the local, or microscopic structure. Requires complete datasets - for developers/use in benchmark studies only.

### Usage

```
KNN(xorigin, ximputed, k = 3)
```

**Arguments**

xorigin            numeric matrix. The original log-intensity data. Can not contain missing values.  
ximputed           numeric matrix. The imputed log-intensity data. Can not contain missing values.  
k                    number of nearest neighbours. default to k=3.

**Value**

numeric The proportion of preserved k-nearest neighbours in imputed data.

**Examples**

```
data(pxd007959)
y <- pxd007959$y
y <- y[complete.cases(y),]
# for demonstration we use same y for xorigin and ximputed
KNN(y, y)
```

---

msImpute

*Imputation of peptide log-intensity in mass spectrometry label-free proteomics by low-rank approximation*

---

**Description**

Returns a completed matrix of peptide log-intensity where missing values (NAs) are imputed by low-rank approximation of the input matrix. Non-NA entries remain unmodified. msImpute requires at least 4 non-missing measurements per peptide across all samples. It is assumed that peptide intensities (DDA), or MS1/MS2 normalised peak areas (DIA), are log2-transformed and normalised (e.g. by quantile normalisation).

**Usage**

```
msImpute(  
  y,  
  method = c("v2-mnar", "v2", "v1"),  
  group = NULL,  
  a = 0.2,  
  rank.max = NULL,  
  lambda = NULL,  
  thresh = 1e-05,  
  maxit = 100,  
  trace.it = FALSE,  
  warm.start = NULL,  
  final.svd = TRUE,  
  biScale_maxit = 20  
)
```

**Arguments**

<code>y</code>	Numeric matrix giving log-intensity where missing values are denoted by NA. Rows are peptides, columns are samples.
<code>method</code>	character. Allowed values are "v2" for msImputeV2 imputation (enhanced version) for MAR. <code>method="v2-mnar"</code> (modified low-rank approx for MNAR), and "v1" initial release of msImpute
<code>group</code>	character or factor vector of length <code>ncol(y)</code>
<code>a</code>	numeric. the weight parameter. default to 0.2.
<code>rank.max</code>	Numeric. This restricts the rank of the solution. is set to $\min(\dim(y)-1)$ by default in "v1".
<code>lambda</code>	Numeric. Nuclear-norm regularization parameter. Controls the low-rank property of the solution to the matrix completion problem. By default, it is determined at the scaling step. If set to zero the algorithm reverts to "hardImputation", where the convergence will be slower. Applicable to "v1" only.
<code>thresh</code>	Numeric. Convergence threshold. Set to 1e-05, by default. Applicable to "v1" only.
<code>maxit</code>	Numeric. Maximum number of iterations of the algorithm before the algorithm is converged. 100 by default. Applicable to "v1" only.
<code>trace.it</code>	Logical. Prints traces of progress of the algorithm. Applicable to "v1" only.
<code>warm.start</code>	List. A SVD object can be used to initialize the algorithm instead of random initialization. Applicable to "v1" only.
<code>final.svd</code>	Logical. Shall final SVD object be saved? The solutions to the matrix completion problems are computed from U, D and V components of final SVD. Applicable to "v1" only.
<code>biScale_maxit</code>	number of iteration for the scaling algorithm to converge. See <code>scaleData</code> . You may need to change this parameter only if you're running <code>method=v1</code> . Applicable to "v1" only.

**Details**

msImpute operates on the `softImpute-als` algorithm in `softImpute` package. The algorithm estimates a low-rank matrix ( a smaller matrix than the input matrix) that approximates the data with a reasonable accuracy. `SoftImpute-als` determines the optimal rank of the matrix through the `lambda` parameter, which it learns from the data. This algorithm is implemented in `method="v1"`. In v2 we have used a information theoretic approach to estimate the optimal rank, instead of relying on `softImpute-als` defaults. Similarly, we have implemented a new approach to estimate `lambda` from the data. Low-rank approximation is a linear reconstruction of the data, and is only appropriate for imputation of MAR data. In order to make the algorithm applicable to MNAR data, we have implemented `method="v2-mnar"` which imputes the missing observations as weighted sum of values imputed by msImpute v2 (`method="v2"`) and random draws from a Gaussian distribution. Missing values that tend to be missing completely in one or more experimental groups will be weighted more (shrunk) towards imputation by sampling from a Gaussian parameterised by smallest observed values in the sample (similar to `minProb`, or `Perseus`). However, if the missing value distribution is even across the samples for a peptide, the imputed values for that peptide are shrunk towards low-rank imputed values. The judgment of distribution of missing values is based on the EBM metric implemented in `selectFeatures`, which is also a information theory measure.

**Value**

Missing values are imputed by low-rank approximation of the input matrix. If input is a numeric matrix, a numeric matrix of identical dimensions is returned.

**Author(s)**

Soroor Hediye-zadeh

**References**

Hastie, T., Mazumder, R., Lee, J. D., & Zadeh, R. (2015). Matrix completion and low-rank SVD via fast alternating least squares. *The Journal of Machine Learning Research*, 16(1), 3367-3402.

Hediye-zadeh, S., Webb, A. I., & Davis, M. J. (2020). MSImpute: Imputation of label-free mass spectrometry peptides by low-rank approximation. *bioRxiv*.

**See Also**

selectFeatures

**Examples**

```
data(pxd010943)
y <- log2(data.matrix(pxd010943))
group <- gsub("_[1234]", "", colnames(y))
yimp <- msImpute(y, method="v2-mnar", group=group)
```

plotCV2

*Plot mean-CV<sup>2</sup> trend***Description**

For each peptide, the squares of coefficient of variations are computed and plotted against average log-intensity. Additionally, a loess trend is fitted to the plotted values. Outlier observations (possibly originated from incorrect match between runs), are detected and highlighted. Users can use this plot as a diagnostic plot to determine if filtering by average intensity is required.

**Usage**

```
plotCV2(y, trend = TRUE, outlier = FALSE, sigma = 1, eps = 0.001, main = NULL)
```

**Arguments**

y	numeric matrix of log-intensity
trend	logical. Should a loess trend be fitted to CV <sup>2</sup> and mean values. Default to TRUE.
outlier	logical. Should outlier points be highlighted? Defaults to FALSE.
sigma	numeric. Kernel width in RBF kernel. Default to 1.
eps	numeric. This is threshold used to call a data point as an outlier. Default to 0.001
main	character string. Title of the plot. Default to NULL

**Details**

Outliers are determined by computing the RBF kernels, which reflect the chance that an observed point belong to the dataset (i.e. is close enough in distance to other data points). Users can determine the cut-off for intensity-based filtering with respect to the mean log-intensity of the outlier points.



**Value**

A plot is created on the current graphics device.

**Examples**

```
data(pxd010943)
y <- pxd010943
y <- log2(y)
ppCV2 <- plotCV2(y)
```

---

pxd007959

*Processed peptide intensity matrix and experimental design table from PXD007959 study*

---

**Description**

Extracellular vesicles isolated from the descending colon of pediatric patients with inflammatory bowel disease and control patients. Characterizes the proteomic profile of extracellular vesicles isolated from the descending colon of pediatric patients with inflammatory bowel disease and control participants. This object contains data from peptide.txt table output by MaxQuant. Rows are Modified Peptide IDs. Charge state variations are treated as distinct peptide species. Reverse complements and contaminant peptides are discarded. Peptides with more than 4 observed intensity values are retained. Additionally, qualified peptides are required to map uniquely to proteins. Two of the samples with missing group annotation were excluded. The peptide.txt and experimentalDesignTemplate files can be downloaded as RDS object from <https://github.com/soroorh/proteomicscasestudies>. Code for data processing is provided in package vignette.

**Usage**

```
pxd007959
```

**Format**

A list of two: samples (data frame of sample descriptions), and y (numeric matrix of peptide intensity values)

**Source**

<http://proteomecentral.proteomexchange.org/cgi/GetDataset?ID=PX007959>

**References**

Zhang X, Deeke SA, Ning Z, Starr AE, Butcher J, Li J, Mayne J, Cheng K, Liao B, Li L, Singleton R, Mack D, Stintzi A, Figey D, Metaproteomics reveals associations between microbiome and intestinal extracellular vesicle proteins in pediatric inflammatory bowel disease. *Nat Commun*, 9(1):2873(2018)

---

pxd010943

*SWATH-MS Analysis of Gfi1-mutant bone marrow neutrophils*

---

### Description

Contains Peak Area for peptides in PXD010943. This study investigates the proteomic alterations in bone marrow neutrophils isolated from 5-8 week old Gfi1+/-, Gfi1K403R/-, Gfi1R412X/-, and Gfi1R412X/R412X mice using the SWATH-MS technique. This dataset consists of 13 SWATH-DIA runs on a TripleTOF 5600 plus (SCIEX). Rows are peptides. Charge state variations are treated as distinct peptide species. Peptides with more than 4 observed intensity values are retained. The peptide.txt and experimentalDesignTemplate files can be downloaded as RDS object from <https://github.com/soroorh/proteomicscasestudies>. Code for data processing is provided in package vignette.

### Usage

pxd010943

### Format

A matrix

### Source

<http://proteomecentral.proteomexchange.org/cgi/GetDataset?ID=PX010943>

### References

Muench DE, Olsson A, Ferchen K, Pham G, Serafin RA, Chutipongtanate S, Dwivedi P, Song B, Hay S, Chetal K, Trump-Durbin LR, Mookerjee-Basu J, Zhang K, Yu JC, Lutzko C, Myers KC, Nazor KL, Greis KD, Kappes DJ, Way SS, Salomonis N, Grimes HL, Mouse models of neutropenia reveal progenitor-stage-specific defects. *Nature*, 582(7810):109-114(2020)

---

pxd014777

*Processed peptide intensity matrix from PXD014777 study*

---

### Description

A Trapped Ion Mobility Spectrometry (TIMS) dataset of blood plasma from a number of patients acquired in two batches. This is a technical dataset published by MaxQuant to benchmark their software for ion mobility enhanced shotgun proteomics. Rows are Modified Peptide IDs. Charge state variations are treated as distinct peptide species. For peptides with multiple identification types, the intensity is considered to be the median of reported intensity values. Reverse complements and contaminant peptides are discarded. Peptides with more than 4 observed intensity values are retained. This object contains data from peptide.txt table output by MaxQuant. The evidence.txt file can be downloaded as RDS object from <https://github.com/soroorh/proteomicscasestudies>. Code for data processing is provided in package vignette.

**Usage**

```
pxd014777
```

**Format**

A matrix

**Source**

<http://proteomecentral.proteomexchange.org/cgi/GetDataset?ID=PXD014777>

**References**

Prianchnikov N, Koch H, Koch S, Lubeck M, Heilig R, Brehmer S, Fischer R, Cox J, MaxQuant Software for Ion Mobility Enhanced Shotgun Proteomics. Mol Cell Proteomics, 19(6):1058-1069(2020)

---

scaleData	<i>Standardize a matrix to have optionally row means zero and variances one, and/or column means zero and variances one.</i>
-----------	--

---

**Description**

Standardize a matrix to have optionally row means zero and variances one, and/or column means zero and variances one.

**Usage**

```
scaleData(
  object,
  maxit = 20,
  thresh = 1e-09,
  row.center = TRUE,
  row.scale = TRUE,
  col.center = TRUE,
  col.scale = TRUE,
  trace = FALSE
)
```

**Arguments**

object	numeric matrix giving log-intensity where missing values are denoted by NA. Rows are peptides, columns are samples.
maxit	numeric. maximum iteration for the algorithm to converge (default to 20). When both row and column centering/scaling is requested, iteration may be necessary.
thresh	numeric. Convergence threshold (default to 1e-09).
row.center	logical. if row.center==TRUE (the default), row centering will be performed resulting in a matrix with row means zero. If row.center is a vector, it will be used to center the rows. If row.center=FALSE nothing is done.
row.scale	if row.scale==TRUE, the rows are scaled (after possibly centering, to have variance one. Alternatively, if a positive vector is supplied, it is used for row centering.

col.center	Similar to row.center
col.scale	Similar to row.scale
trace	logical. With trace=TRUE, convergence progress is reported, when iteration is needed.

### Details

Standardizes rows and/or columns of a matrix with missing values, according to the biScale algorithm in Hastie et al. 2015. Data is assumed to be normalised and log-transformed. Please note that data scaling might not be appropriate for MS1 data. A good strategy is to compare mean-variance plot (plotCV2) before and after imputation. If the plots look differently, you may need to skip data scaling. The MS1 data are more variable (tend to have higher CV<sup>2</sup>), and may contain outliers which will skew the scaling.

### Value

A list of two components: E and E.scaled. E contains the input matrix, E.scaled contains the scaled data

### References

Hastie, T., Mazumder, R., Lee, J. D., & Zadeh, R. (2015). Matrix completion and low-rank SVD via fast alternating least squares. *The Journal of Machine Learning Research*, 16(1), 3367-3402.

Hediyeh-zadeh, S., Webb, A. I., & Davis, M. J. (2020). MSImpute: Imputation of label-free mass spectrometry peptides by low-rank approximation. *bioRxiv*.

### See Also

selectFeatures, msImpute

### Examples

```
data(pxd010943)
y <- pxd010943
y <- log2(y)
keep <- (rowSums(!is.na(y)) >= 4)
y <- as.matrix.data.frame(y[keep,])
y <- scaleData(y, maxit=30)
```

---

selectFeatures

*Select features for MAR/MNAR pattern examination*

---

### Description

Two methods are provided to identify features (peptides or proteins) that can be informative of missing patterns. Method hvp fits a linear model to peptide dropout rate (proportion of samples were peptide is missing) against peptide abundance (average log<sub>2</sub>-intensity). Method emb is a information theoretic approach to identify missing patterns. It quantifies the heterogeneity (entropy) of missing patterns per biological (experimental group). This is the default method.

**Usage**

```
selectFeatures(
  x,
  method = c("ebm", "hvp"),
  group,
  n_features = 500,
  suppress_plot = TRUE
)
```

**Arguments**

x	Numeric matrix giving log-intensity where missing values are denoted by NA. Rows are peptides, columns are samples.
method	character. What method should be used to find features? options include method=' hvp ' and method=' ebm '
group	character or factor vector specifying biological (experimental) group e.g. control, treatment, WT, KO
n_features	Numeric, number of features with high dropout rate. 500 by default. Applicable if method="hvp".
suppress_plot	Logical show plot of dropouts vs abundances. Default to TRUE. Applicable if method="hvp".

**Details**

In general, the presence of group-wise (structured) blocks of missing values, where peptides are missing in one experimental group can indicate MNAR, whereas if such patterns are absent (or missingness is uniform across the samples), peptides are likely MAR. In the presence of MNAR, left-censored MNAR imputation methods should be chosen. Two methods are provided to explore missing patterns: method=hvp identifies top n\_features peptides with high average expression that also have high dropout rate, defined as the proportion of samples where peptide is missing. Peptides with high (potentially) biological dropouts are marked in the hvp column in the output dataframe. This method does not use any information about experimental conditions (i.e. group). Another approach to explore and quantify missing patterns is by looking at how homogeneous or heterogeneous missing patterns are in each experimental group. This is done by computing entropy of distribution of observed values. This is the default and recommended method for selectFeatures. Entropy is reported in EBM column of the output. A NaN EBM indicates peptide is missing at least in one experimental group. Features set to TRUE in msImpute\_feature column are the features selected by the selected method. Users are encouraged to use the EBM metric to find informative features, hence why the group argument is required.

**Value**

A data frame with a logical column denoting the selected features

**Author(s)**

Soroor Hediyezh-zadeh

**References**

Hediyezh-zadeh, S., Webb, A. I., & Davis, M. J. (2020). MSImpute: Imputation of label-free mass spectrometry peptides by low-rank approximation. bioRxiv.

**See Also**

msImpute

**Examples**

```
data(pxd007959)
group <- pxd007959$samples$group
y <- data.matrix(pxd007959$y)
y <- log2(y)
hdp <- selectFeatures(y, method="ebm", group = group)
# construct matrix M to capture missing entries
M <- ifelse(is.na(y),1,0)
M <- M[hdp$msImpute_feature,]
# plot a heatmap of missingness patterns for the selected peptides
require(ComplexHeatmap)
hm <- Heatmap(M,
  column_title = "dropout pattern, columns ordered by dropout similarity",
  name = "Intensity",
  col = c("#8FBC8F", "#FFEFDB"),
  show_row_names = FALSE,
  show_column_names = TRUE,
  cluster_rows = TRUE,
  cluster_columns = TRUE,
  show_column_dend = TRUE,
  show_row_dend = FALSE,
  row_names_gp = gpar(fontsize = 7),
  column_names_gp = gpar(fontsize = 8),
  heatmap_legend_param = list(direction = "horizontal",
  heatmap_legend_side = "bottom",
  labels = c("observed", "missing"),
  legend_width = unit(6, "cm")),
  )
hm <- draw(hm, heatmap_legend_side = "left")
```

# Index

## \* datasets

pxd007959, [9](#)  
pxd010943, [10](#)  
pxd014777, [10](#)

computeStructuralMetrics, [2](#)  
CPD, [3](#)

findVariableFeatures, [4](#)

KNC, [5](#)  
KNN, [5](#)

msImpute, [6](#)

plotCV2, [8](#)  
pxd007959, [9](#)  
pxd010943, [10](#)  
pxd014777, [10](#)

scaleData, [11](#)  
selectFeatures, [12](#)  
softImpute, [7](#)