

# Package ‘mxnorm’

June 1, 2022

**Title** Apply Normalization Methods to Multiplexed Images

**Version** 1.0.2

**Description** Implements methods to normalize multiplexed imaging data, including statistical metrics and visualizations to quantify technical variation in this data type. Reference for methods listed here: Harris, C., Wrobel, J., & Vandekar, S. (2022). mxnorm: An R Package to Normalize Multiplexed Imaging Data. *Journal of Open Source Software*, 7(71), 4180, <[doi:10.21105/joss.04180](https://doi.org/10.21105/joss.04180)>.

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.1.1

**Depends** R (>= 3.5.0)

**Suggests** knitr, janitor (>= 2.1.0), testthat (>= 3.0.0), rmarkdown, markdown, covr

**Config/testthat/edition** 3

**Config/reticulate** list( packages = list( list(package = ``scikit-image" ) ) )

**Imports** magrittr, dplyr, rlang, fda, stats, data.table, reticulate, caret, e1071, ggplot2, tidyr, uwot, lme4, stringr, KernSmooth, kSamples, psych, fossil, glue, methods

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Coleman Harris [aut, cre] (<<https://orcid.org/0000-0002-6325-0694>>)

**Maintainer** Coleman Harris <[coleman.r.harris@vanderbilt.edu](mailto:coleman.r.harris@vanderbilt.edu)>

**Repository** CRAN

**Date/Publication** 2022-06-01 21:30:07 UTC

## R topics documented:

mx_dataset	2
mx_normalize	3
mx_sample	4
plot_mx_density	4
plot_mx_discordance	5
plot_mx_proportions	6
plot_mx_umap	6
print.summary.mx_dataset	7
run_otsu_discordance	8
run_reduce_umap	9
run_var_proportions	10
summary.mx_dataset	11

<b>Index</b>	<b>12</b>
--------------	-----------

---

mx_dataset	<i>Generates mx_dataset</i>
------------	-----------------------------

---

### Description

Takes in data from data.frame of cell-level multiplexed data to create a mx\_dataset S3 object.

### Usage

```
mx_dataset(data, slide_id, image_id, marker_cols, metadata_cols = NULL)
```

### Arguments

data	multiplexed data to normalize. Data assumed to be a data.frame with cell-level data.
slide_id	String slide identifier of input data. This must be a column in the data data.frame.
image_id	String image identifier of input data. This must be a column in the data data.frame.
marker_cols	vector of column name(s) in data corresponding to marker values.
metadata_cols	other identifiers of the input data (default=NULL). This must be a vector of column name(s) in the data data.frame.

### Value

data.frame object in the mx\_dataset format with attribute for input type

### Examples

```
mx_data = mx_dataset(mxnorm::mx_sample, "slide_id", "image_id",
  c("marker1_vals", "marker2_vals", "marker3_vals"),
  c("metadata1_vals"))
```

---

mx_normalize	<i>Normalizes multiplexed data</i>
--------------	------------------------------------

---

## Description

Normalizes some given image input according to the method specified

## Usage

```
mx_normalize(
  mx_data,
  transform = "None",
  method = "None",
  method_override = NULL,
  method_override_name = NULL,
  ...
)
```

## Arguments

mx_data	mx_dataset object to normalize
transform	transformation to perform on the input data. Options include: c("None", "log10", "mean_divide", "log10_mean_divide")
method	normalization method to perform on the input data. Options include: c("None", "ComBat", "Registration")
method_override	optional user-defined function to perform own normalization method (default=NULL). If using a user-defined function, it must include a mx_data parameter.
method_override_name	optional name for method_override (default=NULL).
...	optional additional arguments for normalization functions

## Value

Multiplexed data normalized according to the method specified, in the mx\_dataset format. Normalized data will be included a new table with normalized values and attributes describing the transformation.

## Examples

```
mx_data = mx_dataset(mxnorm::mx_sample, "slide_id", "image_id",
  c("marker1_vals", "marker2_vals", "marker3_vals"),
  c("metadata1_vals"))
mx_data = mx_normalize(mx_data, transform="log10", method="None")
```

---

mx_sample	<i>Sample multiplexed dataset for mxnorm.</i>
-----------	---

---

### Description

A simulated multiplexed dataset containing: 4 slides, with 3 images each, and 3 markers, and 1 metadata column.

### Usage

```
mx_sample
```

### Format

A data frame with 3000 rows and 6 variables:

**slide\_id** slide identifier

**image\_id** image identifier

**marker1\_vals** simulated marker 1 values

**marker2\_vals** simulated marker 2 values

**marker3\_vals** simulated marker 3 values

**metadata1\_vals** simulated metadata 1 values ...

---

plot_mx_density	<i>Visualize marker density before/after normalization by marker and slide</i>
-----------------	--

---

### Description

Visualize marker density before/after normalization by marker and slide

### Usage

```
plot_mx_density(mx_data)
```

### Arguments

mx_data	mx_dataset object that been used with run_otsu_discordance() to compute Otsu discordance scores (necessary for the density rug plot). Note that the table attribute must be set when running run_otsu_discordance().
---------	--

### Value

ggplot2 object with density plot

## Examples

```
mx_data = mx_dataset(mxnorm::mx_sample, "slide_id", "image_id",
  c("marker1_vals", "marker2_vals", "marker3_vals"),
  c("metadata1_vals"))
mx_data = mx_normalize(mx_data, transform="log10", method="None")
## using `threshold_override` here in case users haven't installed `scikit-image`
mx_data = run_otsu_discordance(mx_data, table="normalized",
  threshold_override = function(thold_data){quantile(thold_data, 0.5)})
plot_mx_density(mx_data)
```

---

plot\_mx\_discordance *Visualize Otsu discordance scores by marker and slide*

---

## Description

Visualize Otsu discordance scores by marker and slide

## Usage

```
plot_mx_discordance(mx_data)
```

## Arguments

`mx_data` `mx_dataset` object that been used with `run_otsu_discordance()` to compute Otsu discordance scores. Note that the `table` attribute must be set when running `run_otsu_discordance()`.

## Value

ggplot2 object with Otsu discordance scores plot

## Examples

```
mx_data = mx_dataset(mxnorm::mx_sample, "slide_id", "image_id",
  c("marker1_vals", "marker2_vals", "marker3_vals"),
  c("metadata1_vals"))
mx_data = mx_normalize(mx_data, transform="log10", method="None")
## using `threshold_override` here in case users haven't installed `scikit-image`
mx_data = run_otsu_discordance(mx_data, table="normalized",
  threshold_override = function(thold_data){quantile(thold_data, 0.5)})
plot_mx_discordance(mx_data)
```

---

plot\_mx\_proportions     *Visualize variance proportions by marker and table*

---

### Description

Visualize variance proportions by marker and table

### Usage

```
plot_mx_proportions(mx_data)
```

### Arguments

`mx_data`            `mx_dataset` object that been used with `run_var_proportions()` to run random effects modeling. Note that the `table` attribute must be set when running `run_var_proportions()`.

### Value

ggplot2 object with proportions plot

### Examples

```
mx_data = mx_dataset(mxnorm::mx_sample, "slide_id", "image_id",
  c("marker1_vals", "marker2_vals", "marker3_vals"),
  c("metadata1_vals"))
mx_data = mx_normalize(mx_data, transform="log10", method="None")
mx_data = run_var_proportions(mx_data, table="both")
plot_mx_proportions(mx_data)
```

---

plot\_mx\_umap            *Visualize UMAP dimension reduction algorithm*

---

### Description

Visualize UMAP dimension reduction algorithm

### Usage

```
plot_mx_umap(mx_data, metadata_col = NULL)
```

### Arguments

`mx_data`            `mx_dataset` object that been used with `run_reduce_umap()` to compute the UMAP dimensions for the dataset. Note that the `table` attribute must be set when running `run_reduce_umap()`.

`metadata_col`      column denoted in the `run_reduce_umap()` to change the `scale_color` attribute of the ggplot (default=NULL)

## Value

ggplot2 object with density plot

## Examples

```
mx_data = mx_dataset(mxnorm::mx_sample, "slide_id", "image_id",
  c("marker1_vals", "marker2_vals", "marker3_vals"),
  c("metadata1_vals"))
mx_data = mx_normalize(mx_data, transform="log10", method="None")
mx_data = run_reduce_umap(mx_data, table="normalized",
  c("marker1_vals", "marker2_vals", "marker3_vals"))
plot_mx_umap(mx_data)
```

---

print.summary.mx\_dataset

*Extension of print S3 method to print summary.mx\_dataset objects*

---

## Description

Extension of print S3 method to print summary.mx\_dataset objects

## Usage

```
## S3 method for class 'summary.mx_dataset'
print(x, ...)
```

## Arguments

x                   summary.mx\_dataset object to summarize  
...                   option for additional params given S3 logic

## Examples

```
mx_data = mx_dataset(mxnorm::mx_sample, "slide_id", "image_id",
  c("marker1_vals", "marker2_vals", "marker3_vals"),
  c("metadata1_vals"))
print(summary(mx_data))
```

---

run\_otsu\_discordance    *Calculate Otsu discordance scores using specified threshold for an mx\_dataset object.*

---

### Description

The Otsu discordance analysis quantifies slide-to-slide agreement by summarizing the distance between slide-level Otsu thresholds and the global Otsu threshold for a given marker in a single metric.

### Usage

```
run_otsu_discordance(
    mx_data,
    table,
    threshold_override = NULL,
    plot_out = FALSE,
    ...
)
```

### Arguments

mx_data	mx_dataset object used to compute Otsu discordance scores
table	dataset in mx_data used to compute metrics. Options include: c("raw", "normalized", "both"), e.g. a y-axis parameter.
threshold_override	optional user-defined function or alternate thresholding algorithm adaptable from Python skimage module <code>filters</code> (Note: not all algorithms in <code>filters</code> adapted). Options include supplying a function or any of the following: c("isodata", "li", "mean", "otsu", "triangle", "yen"). More detail available here: <a href="https://scikit-image.org/docs/dev/api/skimage.filters.html">https://scikit-image.org/docs/dev/api/skimage.filters.html</a> . If using a user-defined function, it must include a <code>thold_data</code> parameter.
plot_out	boolean to generate Otsu discordance plots (default=FALSE)
...	optional additional arguments for Otsu discordance functions

### Value

mx\_dataset object with analysis results of Otsu discordance in `otsu_data` table

### References

Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1), 62-66.



**Examples**

```

mx_data = mx_dataset(mxnorm::mx_sample, "slide_id", "image_id",
  c("marker1_vals", "marker2_vals", "marker3_vals"),
  c("metadata1_vals"))
mx_data = mx_normalize(mx_data, transform="log10", method="None")
## using `threshold_override` here in case users haven't installed `scikit-image`
mx_data = run_otsu_discordance(mx_data, table="normalized",
  threshold_override = function(thold_data){quantile(thold_data, 0.5)})

```

---

run_reduce_umap	<i>Run UMAP dimension reduction algorithm on an mx_dataset object.</i>
-----------------	--

---

**Description**

Run UMAP dimension reduction algorithm on an mx\_dataset object.

**Usage**

```

run_reduce_umap(
  mx_data,
  table,
  marker_list,
  downsample_pct = 1,
  metadata_cols = NULL
)

```

**Arguments**

mx_data	mx_dataset object used to compute UMAP dimensions
table	dataset in mx_data used to compute metrics. Options include: c("raw", "normalized", "both"), e.g. a y-axis parameter.
marker_list	list of markers in the mx_dataset object to use for UMAP algorithm
downsample_pct	double, optional percentage (0, 1] of sample rows to include when running UMAP algorithm. (default=1)
metadata_cols	other identifiers of the input data (default=NULL). This must be a vector of column name(s) in the mx_dataset object

**Value**

mx\_dataset object with analysis results of UMAP dimension results in umap\_data table

**Examples**

```
mx_data = mx_dataset(mxnorm::mx_sample, "slide_id", "image_id",
  c("marker1_vals", "marker2_vals", "marker3_vals"),
  c("metadata1_vals"))
mx_data = mx_normalize(mx_data, transform="log10", method="None")
mx_data = run_reduce_umap(mx_data, table="normalized",
  c("marker1_vals", "marker2_vals", "marker3_vals"))
```

---

run\_var\_proportions     *Run random effects modeling on mx\_dataset object to determine proportions of variance at the slide level*

---

**Description**

Run random effects modeling on `mx_dataset` object to determine proportions of variance at the slide level

**Usage**

```
run_var_proportions(
  mx_data,
  table,
  metadata_cols = NULL,
  formula_override = NULL,
  save_models = FALSE,
  ...
)
```

**Arguments**

<code>mx_data</code>	<code>mx_dataset</code> object used to compute UMAP dimensions
<code>table</code>	dataset in <code>mx_data</code> used to compute metrics. Options include: <code>c("raw", "normalized", "both")</code> , e.g. a y-axis parameter.
<code>metadata_cols</code>	other identifiers of the input data to use in the modeling process (default=NULL). This must be a vector of column name(s) in the <code>mx_dataset</code> object
<code>formula_override</code>	String with user-defined formula to use for variance proportions modeling analysis (default=NULL). This will be the RHS of a formula with <code>marker~</code> as the LHS.
<code>save_models</code>	Boolean flag to save <code>lme4::lmer()</code> models in a list to the <code>mx_dataset</code> object
<code>...</code>	optional additional arguments for <code>lme4::lmer()</code> modeling

**Value**

`mx_dataset` object with modeling results in `var_data` table

**Examples**

```
mx_data = mx_dataset(mxnorm::mx_sample, "slide_id", "image_id",
  c("marker1_vals", "marker2_vals", "marker3_vals"),
  c("metadata1_vals"))
mx_data = mx_normalize(mx_data, transform="log10", method="None")
mx_data = run_var_proportions(mx_data, table="both")
```

---

summary.mx\_dataset      *Extension of summary S3 method to summarize mx\_dataset objects*

---

**Description**

Extension of summary S3 method to summarize mx\_dataset objects

**Usage**

```
## S3 method for class 'mx_dataset'
summary(object, ...)
```

**Arguments**

object	mx_dataset object to summarize
...	option for additional params given S3 logic

**Value**

summary.mx\_dataset object

**Examples**

```
mx_data = mx_dataset(mxnorm::mx_sample, "slide_id", "image_id",
  c("marker1_vals", "marker2_vals", "marker3_vals"),
  c("metadata1_vals"))
summary(mx_data)
```

# Index

## \* datasets

- [mx\\_sample, 4](#)
  
- [mx\\_dataset, 2](#)
- [mx\\_normalize, 3](#)
- [mx\\_sample, 4](#)
  
- [plot\\_mx\\_density, 4](#)
- [plot\\_mx\\_discordance, 5](#)
- [plot\\_mx\\_proportions, 6](#)
- [plot\\_mx\\_umap, 6](#)
- [print.summary.mx\\_dataset, 7](#)
  
- [run\\_otsu\\_discordance, 8](#)
- [run\\_reduce\\_umap, 9](#)
- [run\\_var\\_proportions, 10](#)
  
- [summary.mx\\_dataset, 11](#)