Package 'xvm'

April 30, 2025

Title Read, Parse and Visualize 'XVG'/'XPM' Files from Molecular Dynamics

Version 0.0.2

Description
Provides tools for reading, parsing and visualizing simulation data stored in 'xvg'/'xpm' file formats (commonly generated by 'GROMACS' molecular dynamics software). Streamlines post-processing
and analysis of molecular dynamics ('MD') simulation outputs, enabling efficient exploration of molecular stability and conformational changes. Supports import of trajectory metrics ('RMSD', energy, temperature) and creation of publication-ready visualizations through integration with 'ggplot2'.

URL https://github.com/RightSZ/xvm, https://rightsz.github.io/xvm/

BugReports https://github.com/RightSZ/xvm/issues

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Imports ggplot2, tidyr, ggnewscale, plotly

Suggests knitr, rmarkdown, ggpubr, stringr

VignetteBuilder knitr

License GPL (>= 3)

NeedsCompilation no

Author BeiHao Li [aut, cre]

Maintainer BeiHao Li <szright2000@gmail.com>

Contents

export_xvg 2
format_text 3
merge_xvg_data 3
parse_xpm 4
parse_xvg 4
plot_xpm 5
plot_xpm_3d 5
plot_xpm_facet 6

plot_xvg 7
read_xpm 8
read_xvg 9
summary_xvg 9

Index **11**

`export_xvg` *export xvg data*

Description

write the data component of an `xvg_data` object (or multiple objects) to a delimited text file, controlled via the `sep` parameter rather than file extension detection.

Usage

```
export_xvg(xvg_data, file, sep = "\t", row.names = FALSE, merge = FALSE, ...)
```

Arguments

`xvg_data`	An object of class `xvg_data`, or a list of `xvg_data` objects, as returned by `read_xvg()`.
`file`	Path to the output file (any extension is acceptable).
`sep`	Field separator (e.g., "\t" for TSV, "," for CSV). Default is "\t".
`row.names`	Logical, whether to write row names. Default is FALSE.
`merge`	Logical, whether to merge multiple xvg_data objects before exporting. Default is FALSE.
`...`	Additional arguments passed to `write.table()`.

Value

`Invisibly returns the path to the written file.`

Examples

```
## Not run:
xvg <- read_xvg(system.file("extdata/rmsd.xvg", package = "xvm"))
# Export as TSV
export_xvg(xvg, "rmsd.tsv", sep = "\t")
# Export as CSV
export_xvg(xvg, "rmsd.csv", sep = ",")

## End(Not run)
```

format_text *format special text*

Description

processes special formatting in xvg files, converting super/subscripts to ggplot2-compatible expressions. Specifically handles:

- Superscripts: Converts \S...\N to ^ notation
- Subscripts: Converts \s...\N to [] notation

Usage

```
format_text(text)
```

Arguments

text	character string containing xvg-formatted text

Value

formatted character string with ggplot2-compatible expressions

merge_xvg_data *merge multiple xvg data objects*

Description

combines multiple xvg data objects into a single structure, preserving metadata from the first object and adding a group identifier to track the source of each data point.

Usage

```
merge_xvg_data(xvg_data)
```

Arguments

xvg_data	a list of xvg data objects, each containing 'data' and 'metadata' components

Value

a merged xvg data object with:

- data - Combined data frame with an additional 'group' column identifying the source
- metadata - Metadata from the first object in the list

parse_xpm *parse xpm (X PixMap) file content into structured data*

Description

This function parses xpm file content, extracts metadata, color mappings, and matrix data, returning a structured list for further processing.

Usage

```
parse_xpm(xpm_content)
```

Arguments

xpm_content	a character string containing the xpm file content.

Value

a list with the following components:

- data - Data frame containing matrix values with coordinates
- title - Chart title extracted from xpm
- legend - Legend text extracted from xpm
- x_label - X-axis label extracted from xpm
- y_label - Y-axis label extracted from xpm
- color_map - Named list mapping color codes to hex values
- color_values - Named list mapping color codes to numeric values

parse_xvg *parse xvg File Content*

Description

parses content from a single GROMACS-generated xvg file

Usage

```
parse_xvg(lines, skip_comments = TRUE)
```

Arguments

lines	character vector of text lines from xvg file
skip_comments	logical indicating whether to skip comment lines (default: TRUE)

Value

list containing xvg data and metadata with following structure:

- data - Data frame containing numerical data
- metadata - List containing title, axis labels, legends and their formatted versions

plot_xpm *plot xpm data*

Description

plot xpm data using ggplot2

Usage

```
plot_xpm(xpm_data, interpolate = FALSE)
```

Arguments

xpm_data	a xpm object returned by read_xpm
interpolate	logical indicating whether to use raster interpolation (TRUE) or discrete tiles (FALSE). Default is FALSE.

Value

a ggplot2 object

Examples

```
library(xvm)
xpm_file_path <- system.file("extdata/gibbs.xpm", package = "xvm")
xpm_data <- read_xpm(xpm_file_path)
plot_xpm(xpm_data) # plot the xpm data using plot_xpm() function
```

plot_xpm_3d *generate 3d scatter plot from xpm Data*

Description

creates 3d visualization of xpm data with scatter plot.

Usage

```
plot_xpm_3d(xpm_data, reversescale = FALSE, point_size = 2)
```

Arguments

xpm_data	a xpm object (from read_xpm()) or list containing parsed objects.
reversescale	whether to reverse the color scale; default is FALSE
point_size	the size of the points in the scatter plot; default is 2

Value

a plotly object

Examples

```
library(xvm)
xpm_file_path <- system.file("extdata/gibbs.xpm", package = "xvm")
xpm_data <- read_xpm(xpm_file_path)
plot_xpm_3d(xpm_data) # plot 3D scatter plot from xpm file
```

`plot_xpm_facet` *generate faceted plots from xpm Data*

Description

creates dual-panel visualizations of xpm data with scatter or area plots.

Usage

```
plot_xpm_facet(xpm_data, plot_type = "scatter")
```

Arguments

`xpm_data`	a xpm object (from `read_xpm()`) or list containing parsed objects.
`plot_type`	visualization type: "scatter" (default) or "area".

Value

a ggplot2 object with:

- Dual facets showing x/y axis relationships
- Automatic data transformation for visualization
- NULL if invalid plot_type specified

Examples

```
library(xvm)
xpm_file_path <- system.file("extdata/gibbs.xpm", package = "xvm")
xpm_data <- read_xpm(xpm_file_path)
plot_xpm_facet(xpm_data) # plot pseudo-3D from xpm file
```

plot_xvg *plot xvg data*

Description

plot xvg data using ggplot2

Usage

```
plot_xvg(
  xvg_data,
  merge = FALSE,
  title = NULL,
  subtitle = NULL,
  use_color_scale = NULL,
  ...
)
```

Arguments

xvg_data	xvg data object returned by read_xvg
merge	logical; if TRUE and multiple datasets provided, merge them (default: FALSE)
title	chart title (default uses xvg file's title)
subtitle	chart subtitle (default uses xvg file's subtitle)
use_color_scale	custom color scale function (e.g., ggsci::scale_color_bmj) to override default colors
...	additional parameters passed to ggplot2::geom_line

Value

a ggplot2 object

Examples

```
library(xvm)
rmsd_file_path <- system.file("extdata/rmsd.xvg", package = "xvm")
rmsd_data <- read_xvg(rmsd_file_path)
plot_xvg(rmsd_data) # plot the xvg data using plot_xvg() function
```

`read_xpm` *read xpm files*

Description

This function reads xpm (X PixMap) files, validates their existence, and returns parsed data structures in a list format.

Usage

```
read_xpm(xpm_files)
```

Arguments

`xpm_files`	a character vector containing paths to one or more xpm files.

Details

The function performs the following operations:

1. Validates input type (must be character vector)
2. Checks for file existence and filters missing files with warnings
3. Reads valid files and parses them using `parse_xpm()`
4. Returns aggregated results in a named list

Value

list with the following components:

- data - Data frame containing matrix values with coordinates
- title - Chart title extracted from xpm
- legend - Legend text extracted from xpm
- x_label - X-axis label extracted from xpm
- y_label - Y-axis label extracted from xpm
- color_map - Named list mapping color codes to hex values
- color_values - Named list mapping color codes to numeric values

Examples

```
library(xvm)
# Retrieve the path to the example file included in the package
xpm_file_path <- system.file("extdata/gibbs.xpm", package = "xvm")
xpm_data <- read_xpm(xpm_file_path) # read the xpm file using read_xpm() function
names(xpm_data)
```

read_xvg *read xvg files*

Description

read one or more GROMACS-generated xvg files

Usage

```
read_xvg(xvg_files, skip_comments = TRUE)
```

Arguments

xvg_files	character vector of xvg file paths
skip_comments	logical indicating whether to skip comment lines (default: TRUE)

Value

Named list containing xvg data, using filenames (without extension) as keys

Examples

```
library(xvm)
# Retrieve the path to the example file included in the package:
rmsd_file_path <- system.file("extdata/rmsd.xvg", package = "xvm")
rmsd_data <- read_xvg(rmsd_file_path) # read the xvg file using read_xvg() function
names(rmsd_data)
```

summary_xvg *summarize xvg Data*

Description

compute basic summary statistics (mean, sd, min, median, max) for each variable in one or more xvg_data objects.

Usage

```
summary_xvg(xvg_data, merge_results = FALSE)
```

Arguments

xvg_data	a list of class 'xvg_data' or a list containing multiple 'xvg_data' objects, as returned by `read_xvg()`.
merge_results	logical, whether to combine results from multiple objects (default: FALSE). When TRUE, results will include a 'group' column identifying the source.

Value

a data.frame with columns:

group (Optional) Source identifier when processing multiple objects with merge_results=TRUE.

variable Name of the variable (column) in the xvg data.

mean Arithmetic mean of that variable.

sd Standard deviation.

min Minimum value.

median Median value.

max Maximum value.

Examples

```
path <- system.file("extdata/rmsd.xvg", package = "xvm")
xvg <- read_xvg(path)
summary_xvg(xvg)
```

Index

∗ **internal**
format_text, 3
merge_xvg_data, 3
parse_xpm, 4
parse_xvg, 4

export_xvg, 2

format_text, 3

merge_xvg_data, 3

parse_xpm, 4
parse_xpm(), *8*
parse_xvg, 4
plot_xpm, 5
plot_xpm_3d, 5
plot_xpm_facet, 6
plot_xvg, 7

read_xpm, 8
read_xpm(), *5, 6*
read_xvg, 9

summary_xvg, 9