

# Package ‘detectXOR’

June 27, 2025

**Title** XOR Pattern Detection and Visualization

**Version** 0.1.0

**Description** Provides tools for detecting XOR-like patterns in variable pairs in two-class data sets. Includes visualizations for pattern exploration and reporting capabilities with both text and HTML output formats.

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**Imports** dplyr (>= 1.1.0), ggplot2 (>= 3.4.0), ggh4x (>= 0.2.3), tibble (>= 3.1.8), reshape2 (>= 1.4.4), glue (>= 1.6.0), magrittr (>= 2.0.0), stats, ggthemes, DescTools (>= 0.99.50), utils, methods, grDevices, knitr, kableExtra, htmltools, base64enc

**Suggests** testthat (>= 3.0.0), rmarkdown, doParallel, foreach, parallel (>= 4.2.0), future (>= 1.28.0), future.apply (>= 1.10.0), pbmcapply (>= 1.5.0)

**RoxygenNote** 7.2.3

**SystemRequirements** GNU make

**Depends** R (>= 3.5.0)

**URL** <https://github.com/JornLotsch/detectXOR>

**BugReports** <https://github.com/JornLotsch/detectXOR/issues>

**NeedsCompilation** no

**Author** Jorn Lotsch [aut, cre] (ORCID: <<https://orcid.org/0000-0002-5818-6958>>), Alfred Utsch [aut]

**Maintainer** Jorn Lotsch <j.lotsch@em.uni-frankfurt.de>

**Repository** CRAN

**Date/Publication** 2025-06-27 13:00:06 UTC

Contents

detectXOR-package . . . . .	2
detect_xor . . . . .	3
generate_spaghetti_plot_from_results . . . . .	6
generate_xor_reportConsole . . . . .	7
generate_xor_reportHTML . . . . .	8
generate_xy_plot_from_results . . . . .	9
XOR_data . . . . .	11
<b>Index</b>	<b>12</b>

---

detectXOR-package	<i>XOR Pattern Detection and Visualization</i>
-------------------	--

---

Description

Provides tools for detecting XOR-like patterns in variable pairs in two-class data sets. Includes visualizations for pattern exploration and reporting capabilities with both text and HTML output formats.

Details

Core Features:

- 1. Statistical detection using chi-square tests and Kendall’s tau
- 2. Spaghetti plots and xy plot for pattern visualization

Main Functions:

- [detect\\_xor](#): Core detection algorithm
- [generate\\_spaghetti\\_plot\\_from\\_results](#): Line plots
- [generate\\_xy\\_plot\\_from\\_results](#): Plot for pattern visualization

Author(s)

Jorn Lotsch <j.lotsch@em.uni-frankfurt.de>

References

Methodological foundations:

- Pattern detection in machine learning
- Statistical dependency measures (Kendall’s tau)

## See Also

Useful links:

- <https://github.com/JornLotsch/detectXOR>
- Report bugs at <https://github.com/JornLotsch/detectXOR/issues>

Related packages:

- [DescTools](#) for statistical tests
- [ggh4x](#) for advanced plotting
- [future](#) for parallel processing

## Examples

```
# Basic workflow with included dataset
data(XOR_data)

# Detect XOR patterns
results <- detect_xor(XOR_data, class_col = "class")

# Generate visualizations
generate_spaghetti_plot_from_results(
  results$results_df,
  XOR_data,
  class_col = "class"
)

generate_xy_plot_from_results(
  results$results_df,
  XOR_data,
  class_col = "class"
)
```

---

detect\_xor

*Detect XOR Patterns in Variable Pairs*

---

## Description

Identifies XOR-shaped relationships between variables using statistical tests and pattern detection.

## Usage

```
detect_xor(
  data,
  class_col = "class",
  check_tau = TRUE,
  compute_axes_parallel_significance = TRUE,
```

```

p_threshold = 0.05,
tau_threshold = 0.3,
abs_diff_threshold = 20,
split_method = "quantile",
max_cores = 1,
extreme_handling = "winsorize",
winsor_limits = c(0.05, 0.95),
scale_data = TRUE,
use_complete = TRUE
)

```

### Arguments

<code>data</code>	Data frame containing features and class column
<code>class_col</code>	Name of class column (default: "class")
<code>check_tau</code>	Logical - compute classwise tau coefficients (default: TRUE)
<code>compute_axes_parallel_significance</code>	Logical - compute Wilcoxon tests (default: TRUE)
<code>p_threshold</code>	Significance threshold (default: 0.05)
<code>tau_threshold</code>	Tau coefficient threshold (default: 0.3)
<code>abs_diff_threshold</code>	Absolute difference threshold for patterns (default: 20)
<code>split_method</code>	Method for splitting data ("quantile" or "range") (default: "quantile")
<code>max_cores</code>	Maximum cores for parallel processing (default: NULL = automatic)
<code>extreme_handling</code>	Method for handling extreme values; options include "winsorize" or "none" (default: "winsorize")
<code>winsor_limits</code>	Numeric vector of length 2 specifying lower and upper quantiles for winsorization (default: c(0.05, 0.95))
<code>scale_data</code>	Logical; whether to scale/standardize the data before analysis (default: TRUE)
<code>use_complete</code>	Logical; whether to use only complete cases (default: TRUE)

### Details

This function performs an analysis to detect XOR-like patterns in pairwise variable relationships within two-class data sets. The analysis pipeline includes:

1. Data preprocessing (winsorization, scaling, complete cases)
2. Tile pattern analysis using chi-squared tests
3. Classwise Kendall tau correlation analysis
4. Group-wise Wilcoxon significance tests

The function automatically handles parallel processing when multiple cores are available and returns both a summary data frame and detailed results for further analysis.

**Value**

List containing:

results_df	Data frame with detection results for all variable pairs
pair_list	Detailed analysis results for each variable pair

**See Also**

[generate\\_spaghetti\\_plot\\_from\\_results](#) for spaghetti plot visualization, [generate\\_xy\\_plot\\_from\\_results](#) for scatter plot visualization, [generate\\_xor\\_reportConsole](#) for console reporting, [generate\\_xor\\_reportHTML](#) for HTML report generation, [XOR\\_data](#) for example dataset

**Examples**

```
# Load example data
data(XOR_data)

# Run XOR detection
results <- detect_xor(data = XOR_data, class_col = "class")

# View summary of detected patterns
print(results$results_df["xor_shape_detected"])

# Generate visualizations
spaghetti_plot <- generate_spaghetti_plot_from_results(
  results = results,
  data = XOR_data,
  class_col = "class"
)

print(spaghetti_plot)

xy_plot <- generate_xy_plot_from_results(
  results = results,
  data = XOR_data,
  class_col = "class"
)

print(xy_plot)

# Generate console report (doesn't write files)
generate_xor_reportConsole(results, XOR_data, "class", show_plots = FALSE)

# View detailed results for detected pairs
detected_pairs <- results$results_df[results$results_df$xor_shape_detected == TRUE, ]
print(detected_pairs)
```

---

```
generate_spaghetti_plot_from_results
```

*Generate XOR Spaghetti Plots*

---

## Description

Creates connected line plots for variable pairs showing XOR patterns.

## Usage

```
generate_spaghetti_plot_from_results(  
  results,  
  data,  
  class_col,  
  scale_data = TRUE  
)
```

## Arguments

results	Either a data frame from <code>detect_xor()</code> \$results_df or the full list object returned by <code>detect_xor()</code>
data	Original dataset containing variables and classes
class_col	Character string specifying the name of the class column
scale_data	Logical indicating whether to scale variables before plotting (default: TRUE)

## Details

This function creates spaghetti plots (connected line plots) for variable pairs that have been flagged as showing XOR patterns by `detect_xor()`. The function automatically handles both original and rotated XOR patterns, applying the appropriate coordinate transformation when necessary.

The function accepts either the full results object returned by `detect_xor()` or just the `results_df` component extracted from it. Variable pairs are separated using "||" as the delimiter in plot labels.

If no XOR patterns are detected, an empty plot with an appropriate message is returned.

To save the plot, use `ggplot2::ggsave()` or other standard R plotting save methods.

## Value

Returns a ggplot object. No files are saved automatically.

## See Also

[detect\\_xor](#) for XOR pattern detection, [generate\\_xy\\_plot\\_from\\_results](#) for scatter plots

## Examples

```
# Using full results object (recommended)
data(XOR_data)
results <- detect_xor(data = XOR_data, class_col = "class")
spaghetti_plot <- generate_spaghetti_plot_from_results(
  results = results,
  data = XOR_data,
  class_col = "class"
)

# Display the plot
print(spaghetti_plot)

# Save the plot if needed
# ggplot2::ggsave("my_spaghetti_plot.png", spaghetti_plot)

# Using extracted results_df (also works)
xy_plot <- generate_spaghetti_plot_from_results(
  results = results$results_df,
  data = XOR_data,
  class_col = "class"
)
```

---

generate\_xor\_reportConsole

*Generate XOR Detection Report (Console-friendly)*

---

## Description

Creates a report with formatted table and plots for XOR pattern detection results.

## Usage

```
generate_xor_reportConsole(
  results,
  data,
  class_col,
  scale_data = TRUE,
  show_plots = TRUE,
  quantile_lines = c(1/3, 2/3),
  line_method = "quantile"
)
```

## Arguments

results	Either a data frame from <code>detect_xor\$results_df</code> or the full list returned by <code>detect_xor</code> .
---------	---

data	Original dataset containing variables and classes.
class_col	Character specifying the class column name.
scale_data	Logical indicating whether to scale variables in plots. Default: TRUE.
show_plots	Logical indicating whether to display plots. Default: TRUE.
quantile_lines	Numeric vector of quantiles for reference lines in XY plots. Default: c(1/3, 2/3).
line_method	Method for boundary calculation ("quantile" or "range"). Default: "quantile".

**Value**

Invisibly returns a list containing the formatted table and plots (if generated).

**See Also**

[detect\\_xor](#) for XOR pattern detection, [generate\\_xor\\_reportHTML](#) for HTML report generation

---

generate\_xor\_reportHTML

*Generate XOR Detection HTML Report*

---

**Description**

Creates an HTML report with formatted table and plots for XOR pattern detection results.

**Usage**

```
generate_xor_reportHTML(
  results,
  data,
  class_col,
  output_file = "xor_detection_report.html",
  open_browser = TRUE,
  scale_data = TRUE,
  quantile_lines = c(1/3, 2/3),
  line_method = "quantile"
)
```

**Arguments**

results	Either a data frame from <code>detect_xor\$results_df</code> or the full list returned by <code>detect_xor</code> .
data	Original dataset containing variables and classes.
class_col	Character specifying the class column name.
output_file	Character specifying the output HTML file name. Default: "xor_detection_report.html".
open_browser	Logical indicating whether to open the report in browser automatically. Default: TRUE.



scale_data	Logical indicating whether to scale variables in plots. Default: TRUE.
quantile_lines	Numeric vector of quantiles for reference lines in XY plots. Default: c(1/3, 2/3).
line_method	Method for boundary calculation ("quantile" or "range"). Default: "quantile".

**Value**

Invisibly returns the file path of the generated HTML report.

**See Also**

[detect\\_xor](#) for XOR pattern detection, [generate\\_xor\\_reportConsole](#) for text-based report generation

---

generate\_xy\_plot\_from\_results

*Generate XOR Scatter Plots*

---

**Description**

Creates scatterplots with decision boundaries for variable pairs showing XOR patterns.

**Usage**

```
generate_xy_plot_from_results(
  results,
  data,
  class_col,
  scale_data = TRUE,
  quantile_lines = c(1/3, 2/3),
  line_method = "quantile"
)
```

**Arguments**

results	Either a data frame from <code>detect_xor()</code> \$results_df or the full list object returned by <code>detect_xor()</code>
data	Original dataset containing variables and classes
class_col	Character string specifying the name of the class column
scale_data	Logical indicating whether to scale variables before plotting (default: TRUE)
quantile_lines	Numeric vector of length 2 specifying quantiles for reference lines (default: c(1/3, 2/3))
line_method	Character string specifying the boundary calculation method, either "quantile" or "range" (default: "quantile")

## Details

This function creates scatter plots for variable pairs that have been flagged as showing XOR patterns by `detect_xor()`. The plots include dashed reference lines that help visualize the decision boundaries used in XOR pattern detection.

The function automatically handles both original and rotated XOR patterns, applying the appropriate coordinate transformation when necessary. Variable pairs are separated using "||" as the delimiter in plot labels.

The `line_method` parameter controls how reference lines are calculated:

- "quantile": Lines are placed at the specified quantiles of the data distribution
- "range": Lines divide the data range into three equal parts

If no XOR patterns are detected, an empty plot with an appropriate message is returned.

To save the plot, use `ggplot2::ggsave()` or other standard R plotting save methods.

## Value

Returns a ggplot object. No files are saved automatically.

## See Also

[detect\\_xor](#) for XOR pattern detection, [generate\\_spaghetti\\_plot\\_from\\_results](#) for spaghetti plots

## Examples

```
# Using full results object (recommended)
data(XOR_data)
results <- detect_xor(data = XOR_data, class_col = "class")
xy_plot <- generate_xy_plot_from_results(
  results = results,
  data = XOR_data,
  class_col = "class"
)

# Display the plot
print(xy_plot)

# Using different boundary method
xy_plot_range <- generate_xy_plot_from_results(
  results = results,
  data = XOR_data,
  class_col = "class",
  line_method = "range"
)

# Save the plot if needed
# ggplot2::ggsave("my_xy_plot.png", xy_plot)

# Using extracted results_df (also works)
```

```
xy_plot_df <- generate_xy_plot_from_results(  
  results = results$results_df,  
  data = XOR_data,  
  class_col = "class"  
)
```

---

XOR\_data

*Synthetic XOR Pattern Dataset*

---

### Description

Simulated classification dataset containing 400 observations with 5 features demonstrating XOR patterns, linear class differences, and random noise.

### Usage

```
data("XOR_data")
```

### Format

A data frame with 400 rows and 6 variables:

**class** Binary class labels (1 or 2)

**Variable\_A** Normally distributed with subtle class difference ( $\Delta \mu = 0.25$ )

**Variable\_B** High-variance normal distribution ( $\sigma = 3$ ) with moderate class separation ( $\Delta \mu = -0.7$ )

**Variable\_C** XOR pattern component 1 ( $\mu = 3$  vs  $10$  between classes)

**Variable\_D** XOR pattern component 2 ( $\mu = 3$  vs  $10$  between classes)

**Variable\_E** Uniform noise (1-10)

### Source

Synthetic data generated with `rnorm()` and `runif()`

### Examples

```
data(XOR_data)  
str(XOR_data)  
summary(XOR_data)
```

# Index

- \* **XOR**
  - detect\_xor, [3](#)
  - detectXOR-package, [2](#)
  - generate\_spaghetti\_plot\_from\_results,  
[6](#)
  - generate\_xy\_plot\_from\_results, [9](#)
- \* **datasets**
  - XOR\_data, [11](#)
- \* **detection**
  - detect\_xor, [3](#)
- \* **machine learning**
  - detectXOR-package, [2](#)
- \* **package**
  - detectXOR-package, [2](#)
- \* **pattern detection**
  - detectXOR-package, [2](#)
- \* **pattern**
  - detect\_xor, [3](#)
- \* **scatter**
  - generate\_xy\_plot\_from\_results, [9](#)
- \* **spaghetti**
  - generate\_spaghetti\_plot\_from\_results,  
[6](#)
- \* **visualization**
  - generate\_spaghetti\_plot\_from\_results,  
[6](#)
  - generate\_xy\_plot\_from\_results, [9](#)

DescTools, [3](#)

detect\_xor, [2](#), [3](#), [6](#), [8–10](#)

detectXOR-package, [2](#)

future, [3](#)

generate\_spaghetti\_plot\_from\_results,  
[2](#), [5](#), [6](#), [10](#)

generate\_xor\_reportConsole, [5](#), [7](#), [9](#)

generate\_xor\_reportHTML, [5](#), [8](#), [8](#)

generate\_xy\_plot\_from\_results, [2](#), [5](#), [6](#), [9](#)

ggh4x, [3](#)