

# Package ‘TwoDiRef’

March 30, 2026

**Type** Package

**Title** Robust Estimation of Conditional 2D Reference Regions

**Description** Provides tools for constructing conditional two-dimensional reference regions in continuous data, particularly suited for clinical, biological, or epidemiological studies requiring robust multivariate assessment. The implemented methodology combines directional quantiles with median-based partial correlation models to produce reliable and interpretable reference regions even in the presence of outliers. Key features include robust conditional modeling for two responses conditioned on covariates, directional quantile regions, cross-validation of coverage, visualization tools, and flexible formula-based inputs.

**Version** 0.1.0

**License** GPL-3

**Encoding** UTF-8

**Depends** R (>= 4.0.0)

**Imports** qgam, mgcv, sp, ggplot2, scales, shiny

**Suggests** refreg

**RoxygenNote** 7.3.3

**NeedsCompilation** no

**Author** Oscar Lado-Baleato [aut, cre],  
Sara Rodriguez-Pastoriza [aut],  
Javier Roca-Pardinas [aut]

**Maintainer** Oscar Lado-Baleato <ladooscar@uniovi.es>

**Repository** CRAN

**Date/Publication** 2026-03-30 18:40:03 UTC

## Contents

cv_dirq_coverage . . . . .	2
dirq . . . . .	4
median_pcor . . . . .	5
plot.dirq . . . . .	7
plot.median_pcor . . . . .	7

predict.dirq . . . . .	9
print.dirq . . . . .	10
print.median_pcor . . . . .	10
shiny_region . . . . .	11

<b>Index</b>	<b>13</b>
--------------	-----------

---

cv_dirq_coverage	<i>Cross-validation evaluation of coverage for a conditional reference region (dirq)</i>
------------------	--

---

## Description

Performs a block cross-validation to evaluate the empirical coverage of a conditional quantile region estimated by `dirq`. For each fold, a median regression model is refitted and the corresponding region is recomputed, the coverage indicator (whether the held-out test points fall inside the region) is recorded. Optionally fits a logistic GAM to explore how coverage varies with the predictor, returns coverage by quartiles, and plots the GAM with confidence bands.

## Usage

```
cv_dirq_coverage(
  object,
  data,
  fold_size = 50,
  err = 0.01,
  formulas = NULL,
  verbose = FALSE,
  quiet = TRUE,
  show_warnings = FALSE,
  plot_gam = FALSE,
  ci = TRUE,
  return_quartiles = TRUE,
  ...
)
```

## Arguments

<code>object</code>	An object of class <code>dirq</code> . It must contain the components: <code>predictor</code> (character, name of the predictor variable), <code>response_vars</code> (character vector of length two, names of the two response variables), <code>qu</code> (numeric, the nominal coverage level, e.g. 0.95), and <code>mods</code> (a list of the two fitted models from which formulas can be extracted). If <code>mods</code> is not available, the formulas can be supplied directly via the <code>formulas</code> argument.
<code>data</code>	Data frame containing the original data used to fit the <code>dirq</code> object. Must include the predictor and both response variables.
<code>fold_size</code>	Integer, number of observations per cross-validation group. The data are split into groups of (approximately) this size. Default 50.

err	Numeric, tolerance parameter passed to dirq when recomputing the region. Default 0.01.
formulas	Optional list of exactly two formulas specifying the models for the two responses. If NULL (default), the formulas are extracted from object\$mods.
verbose	Logical; if TRUE, print progress messages (in English). Default FALSE.
quiet	Logical; if TRUE, suppress all output from median_pcor and dirq (e.g. the dots showing learning rate estimation). Default TRUE.
show_warnings	Logical; if FALSE (default), warnings from the final GAM fit are suppressed. Set to TRUE for debugging.
plot_gam	Logical; if TRUE, draw a plot of the GAM-estimated coverage (with 95% pointwise confidence intervals) on the percentage scale. Default FALSE.
ci	Logical; if TRUE (default) and plot_gam = TRUE, include the 95% confidence band in the plot.
return_quartiles	Logical; if TRUE, compute coverage by quartiles of the predictor variable and return the result. Default TRUE.
...	Additional arguments passed to median_pcor and dirq (e.g. control options).

## Details

The function implements the following steps for each fold:

1. Split the data into training (all but the current group) and test set.
2. Refit a median regression model (quantile 0.5) using median\_pcor with the two formulas.
3. Recompute the quantile region using dirq with the original nominal level object\$qu and tolerance err.
4. For each observation in the test set, predict the polygon for its predictor value and test whether the point (response1, response2) lies inside it using sp::point.in.polygon.

After all folds, a GAM is fitted to explore how the probability of coverage changes with the predictor. If plot\_gam = TRUE, the estimated coverage curve (with 95% pointwise confidence intervals) is plotted on the percentage scale. The y-axis limits are set dynamically: if the nominal level  $\tau < 0.30$  the minimum is 0%, otherwise it is  $\max(0, 100\tau - 15)\%$ ; the maximum is always 100%.

The function requires the packages **sp** and **mgcv** to be installed. It also assumes that the functions median\_pcor and dirq are available in the calling environment (typically from the package that produced the dirq object, e.g. **refreg**).

## Value

An invisible list with the following components:

- overall\_coverage** Numeric, overall empirical coverage (in percent) rounded to two decimals.
- quartile\_coverage** A data frame with columns quartile (Q1–Q4), coverage (percentage) and n (number of observations in each quartile), or NULL if return\_quartiles = FALSE.
- gam\_fit** A fitted gam object from mgcv: :gam with binomial family, modelling  $.coverage \sim s(\text{predictor})$ . NULL if the predictor is not found in the data.
- data** The original data frame augmented with a column .coverage (0/1 indicator that the point fell inside its cross-validated region).

## Examples

```
# Load example data (from the refreg package)
library(refreg)
data("aegis", package = "refreg")
no_dm <- subset(aegis, dm == "no")

# Fit median regression and compute 95% region
f <- list(fpg ~ s(age), hba1c ~ s(age))
modelo <- median_pcor(f = f, data = no_dm, qu = 0.50)
region <- dirq(modelo, qu = 0.95, err = 0.01)

# Perform cross-validation with groups of 100 observations
cv_results <- cv_dirq_coverage(region, no_dm, fold_size = 100,
                              verbose = TRUE, quiet = TRUE,
                              plot_gam = TRUE, ci = TRUE,
                              return_quartiles = TRUE)

# Overall coverage (percentage)
cv_results$overall_coverage

# Coverage by age quartiles
cv_results$quartile_coverage

# Data with .coverage column
head(cv_results$data)
```

---

dirq	<i>Fit Directional Quantile Boundary on Residuals Dependent on Co- variates</i>
------	---

---

## Description

After removing the conditional dependence between two responses via a [median\\_pcor](#) model, this function fits a directional quantile boundary to the bivariate residuals in polar coordinates. The boundary is estimated as a quantile of the log-radius  $r = \log(\sqrt{x^2 + y^2})$  as a smooth function of the angle  $\alpha = \text{atan2}(y, x)$  and the original predictor (e.g., age). The result can be used to construct a prediction region for the original variables.

## Usage

```
dirq(x, qu = 0.95, err = 0.05, ...)
```

## Arguments

x	An object of class "median_pcor" returned by <a href="#">median_pcor</a> .
qu	Numeric; the quantile to estimate for the boundary (e.g., 0.95 for a 95% region). Passed to <a href="#">qgam</a> .

err                Numeric; the target error rate for the quantile fit in `qgam`. See [qgam](#) for details.  
 ...                Additional arguments passed to `qgam`.

### Value

An object of S3 class "dirq" with the following components:

qgam\_fit            The fitted `qgam` model for `r`.  
 qu                 The quantile used.  
 predictor         Name of the predictor variable (the one shared by the two median models).  
 response\_vars    Names of the two original response variables.  
 original\_residuals  
                    Data frame containing the standardized residuals (`x` and `y`) and the predictor values.  
 mods              List with the three sub-models from `median_pcor`: `y1`, `y2`, and `cor`.  
 scales            List with the MAD scaling factors for the two responses (`x` and `y`).

### Examples

```
if (requireNamespace("refreg", quietly = TRUE)) {
  # Load data and fit a median_pcor model
  data_no_dm <- subset(refreg::aegis, dm == "no")
  f <- list(fpg ~ s(age), hba1c ~ s(age))
  model <- median_pcor(f, data = data_no_dm, qu = 0.5)

  # Fit a 95% directional quantile boundary
  region <- dirq(model, qu = 0.95, err = 0.01)
  print(region)

  # Plotting the region on residuals scale
  plot(region)
  plot(region, newdata = data.frame(age = c(20, 40, 60)))
  # Predicting the region for a given covariate value
  predict(region, newdata = data.frame(age = 18))
} else {
  message("Package 'refreg' is needed to run this example.")
}
```

---

median\_pcor

*Conditional median and partial correlation model (via QGAM)*

---

### Description

Fits two quantile GAM (QGAM) models to estimate conditional medians, standardizes the residuals using the Median Absolute Deviation (MAD), and removes smooth conditional dependence between the two responses by means of varying-coefficient GAM terms.

**Usage**

```
median_pcor(f, data, qu = 0.5)
```

**Arguments**

f	A list of exactly two formulas of the form ‘list(y1 ~ s(z1) + s(z2), y2 ~ s(z1) + s(z2))’. Both formulas must share the same right-hand side structure.
data	A data frame containing all variables used in the formulas.
qu	Quantile to estimate (default is 0.5 for the conditional median).

**Value**

An object of class “median\_pcor” with components:

x.c	Standardized residuals of the first response.
y.c	Centered residuals of the second response.
y.c_res	Residuals of the second response after removing conditional dependence.
mod.y1	QGAM model for the first response.
mod.y2	QGAM model for the second response.
mod.pcor	QGAM varying-coefficient model for partial correlation.
scale.x	MAD of the first-response residuals (used for scaling).
scale.y	MAD of the second-response residuals.
data	Augmented data frame with the computed variables (‘x.c’, ‘y.c’, ‘y.c_res’).
qu	Quantile levels used (default = 0.50).
call	Original call.

**Examples**

```
if (requireNamespace("refreg", quietly = TRUE)) {
  # Subset of the aegis data where diabetes is absent
  data_no_dm <- subset(refreg::aegis, dm == "no")

  # Formulas with the same right-hand side
  f <- list(fpg ~ s(age), hba1c ~ s(age))

  # Fit the model
  model <- median_pcor(f, data = data_no_dm, qu = 0.5)
  print(model)

  # Plot methods would be called as:
  # plot(model, type = "residuals")
  # plot(model, type = "effects")
} else {
  message("Package 'refreg' is needed to run this example.")
}
```

---

plot.dirq                      *Plot method for dirq objects*

---

### Description

Produces a ggplot2 visualization of the directional quantile boundaries fitted by [dirq](#). The plot shows the standardized residuals (after removing conditional dependence) as points, and overlays the estimated boundaries for specified values of the predictor variable. A reference dashed line through the origin is added for orientation.

### Usage

```
## S3 method for class 'dirq'  
plot(x, newdata = NULL, ...)
```

### Arguments

x	An object of class "dirq" returned by <a href="#">dirq</a> .
newdata	Optional vector or one-column data frame containing new predictor values for which to draw the boundaries. If NULL (the default), boundaries are shown at the quartiles of the original predictor.
...	Additional arguments (currently ignored).

### Value

The function is called for producing a plot.

### See Also

[dirq](#) for model fitting.

---

plot.median\_pcor              *Plot method for median\_pcor objects*

---

### Description

Produces two types of plots for objects of class "median\_pcor":

"residuals" scatterplot of the standardized residuals after removing conditional dependence, to check for independence.

"effects" three-panel display showing the estimated median trends for each response and the varying partial correlation coefficient ( $\rho$ ) along the shared covariate.

## Usage

```
## S3 method for class 'median_pcor'  
plot(x, type = c("residuals", "effects"), cex = 1.1, ...)
```

## Arguments

x	An object of class "median_pcor".
type	Character string; the type of plot. Can be abbreviated. Options: "residuals" or "effects".
cex	Numeric multiplier for text size. Passed to <code>par(cex = ...)</code> and used in point size for the residuals plot.
...	Additional graphical parameters.

## Details

For `type = "residuals"`, the function plots `x$y.c_res` against `x$x.c` (the standardized residuals of the second response after removing conditional dependence vs. the standardized residuals of the first response). A reference line at zero is added, and a grid helps assess independence.

For `type = "effects"`, a layout with three panels is created:

1. Median trend of the first response (with pointwise  $1.96 \times \text{SE}$  bands).
2. Median trend of the second response (with pointwise  $1.96 \times \text{SE}$  bands).
3. Estimated varying partial correlation coefficient  $\rho(y_1, y_2 | \mathbf{z})$  as a function of the covariate.

The confidence bands are based on the standard errors returned by `predict(..., se.fit = TRUE)` for the two median models. The correlation panel does not display bands because uncertainty is not trivial to show. Future package version will estimate such estimation error through a bootstrap resampling scheme.

A professional colour palette is used, and graphical parameters are set to produce clean, publication-ready figures.

## Value

The function is called for its side effects (plotting).

## See Also

[median\\_pcor](#) for model fitting.

---

predict.dirq	<i>Predict method for dirq objects</i>
--------------	--

---

## Description

Obtains the fitted directional quantile boundary for new predictor values. The function reconstructs the boundary in the original scale of the response variables by combining the quantile model for the log-radius (fitted in polar coordinates) with the conditional median models and scaling factors stored in the `dirq` object.

## Usage

```
## S3 method for class 'dirq'
predict(object, newdata, ...)
```

## Arguments

<code>object</code>	An object of class "dirq" returned by <code>dirq</code> .
<code>newdata</code>	A vector or one-column data frame containing the new predictor value(s) for which the boundary is desired. If multiple values are supplied, only the first one is used (see Details).
<code>...</code>	Additional arguments (currently ignored, but kept for compatibility with the generic <code>predict</code> ).

## Details

The prediction proceeds as follows:

1. For a fixed predictor value  $z_0$ , a sequence of angles  $\alpha \in [-\pi, \pi]$  is generated.
2. The fitted quantile model for  $r = \log(\sqrt{x^2 + y^2})$  is used to obtain  $\hat{r}(\alpha, z_0)$ ; the boundary radius is then  $R(\alpha, z_0) = \exp(\hat{r})$ .
3. Standardised residuals are recovered as  $x_{\text{std}} = R \cos(\alpha)$  and  $y_{\text{std, res}} = R \sin(\alpha)$ .
4. The conditional correlation effect is added back:  $y_{\text{std}} = y_{\text{std, res}} + \hat{\beta}(z_0)x_{\text{std}}$ , where  $\hat{\beta}(z_0)$  comes from the varying-coefficient model.
5. Finally, the residuals are back-transformed to the original scale using the median predictions and the MAD scaling factors:  $x_{\text{orig}} = x_{\text{std}} \cdot \text{scale}_x + \hat{\mu}_1(z_0)$ ,  $y_{\text{orig}} = y_{\text{std}} \cdot \text{scale}_y + \hat{\mu}_2(z_0)$ .

If `newdata` contains more than one value, only the first is employed, because the function returns a single boundary curve (a closed loop) for a given predictor. To obtain boundaries for multiple predictor values, call `predict` repeatedly or use `plot.dirq`.

## Value

A data frame with two columns named after the original response variables (as stored in `object$response_vars`). Each row corresponds to a point on the predicted boundary curve for the supplied predictor value.

**See Also**

[dirq](#), [plot.dirq](#)

---

print.dirq	<i>Print method for dirq objects</i>
------------	--------------------------------------

---

**Description**

Displays basic information about the fitted directional quantile boundary.

**Usage**

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

**Arguments**

x	An object of class "dirq".
...	Additional arguments (ignored).

**Value**

A text output describing main details of directional quantile model fit.

---

print.median_pcor	<i>Print method for median_pcor objects</i>
-------------------	---

---

**Description**

Print method for median\_pcor objects

**Usage**

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

**Arguments**

x	An object of class median_pcor.
...	Additional arguments (ignored).

**Value**

A text output describing main details of median\_pcor estimation.

---

`shiny_region`*Interactive Shiny App for Visualizing DIRQ Reference Regions*

---

### Description

Launches a Shiny application that displays how the bivariate reference region estimated by `dirq` evolves along a predictor variable. The app shows a sliding window of data points around a chosen predictor value, overlays the corresponding region boundary, and computes the empirical coverage of the region.

### Usage

```
shiny_region(modelo_region, data, window_width = 1, ...)
```

### Arguments

<code>modelo_region</code>	An object of class "dirq" returned by <code>dirq</code> . It contains the fitted directional quantile boundary and the necessary meta-information (predictor name, response names, scaling factors, and sub-models).
<code>data</code>	A data frame containing the original variables used to fit the <code>median_pcor</code> model (and subsequently the <code>dirq</code> object). Must include the predictor and the two response variables with the same names as stored in <code>modelo_region</code> .
<code>window_width</code>	Numeric; the half-width of the moving window used to select data points around the current predictor value. Defaults to 1.
<code>...</code>	Additional arguments passed to the <code>sliderInput</code> for the predictor (e.g., <code>step</code> , <code>animate</code> ) or to the plot (currently ignored).

### Details

The app creates a slider for the predictor variable (with range taken from `data`). For each selected predictor value  $z_0$ , it:

1. Computes the predicted boundary curve using `predict.dirq`.
2. Subsets the data to those observations whose predictor lies within  $[z_0 - w, z_0 + w]$  where  $w$  is `window_width`.
3. Plots the subset as points, fills the region polygon, and draws its outline.
4. Calculates the proportion of points inside the polygon (using `point.in.polygon`) and displays it as "Empirical Coverage".

The slider can be animated (via the built-in play button) to visualise the evolution smoothly.

### Value

A Shiny app object. The function is called for its side effect of launching an interactive web application.

**See Also**

[dirq](#), [predict.dirq](#), [plot.dirq](#)

**Examples**

```
if (requireNamespace("refreg", quietly = TRUE) && interactive()) {  
  # Example using the aegis data  
  data_no_dm <- subset(refreg::aegis, dm == "no")  
  f <- list(fpg ~ s(age), hba1c ~ s(age))  
  model <- median_pcor(f, data = data_no_dm, qu = 0.5)  
  region <- dirq(model, qu = 0.95, err = 0.01)  
  
  # Launch the interactive app  
  shiny_region(region, data_no_dm)  
}
```

# Index

`cv_dirq_coverage`, 2

`dirq`, 4, 7, 9–12

`median_pcor`, 4, 5, 8

`plot.dirq`, 7, 9, 10, 12

`plot.median_pcor`, 7

`point.in.polygon`, 11

`predict`, 9

`predict.dirq`, 9, 11, 12

`print.dirq`, 10

`print.median_pcor`, 10

`qgam`, 4, 5

`shiny_region`, 11

`sliderInput`, 11