

# Package: proxirr (via r-universe)

October 31, 2024

**Type** Package

**Title** Alpha and Beta Proximity to Irreplaceability

**Date** 2024-01-25

**Version** 0.4

**Description** Functions to measure Alpha and Beta Proximity to Irreplaceability. The methods for Alpha and Beta irreplaceability were first described in: Baisero D., Schuster R. & Plumptre A.J. Redefining and Mapping Global Irreplaceability. Conservation Biology 2021;1-11. [doi:10.1111/cobi.13806](https://doi.org/10.1111/cobi.13806).

**License** MIT + file LICENSE

**Encoding** UTF-8

**Suggests** testthat (>= 3.0.0), utils

**RoxygenNote** 7.3.1

**Config/testthat/edition** 3

**NeedsCompilation** no

**Author** Daniele Baisero [aut, cre, cph]  
(<https://orcid.org/0000-0002-1266-7174>)

**Maintainer** Daniele Baisero <daniele.baisero@gmail.com>

**Date/Publication** 2024-01-26 15:50:02 UTC

**Repository** <https://danielebaisero.r-universe.dev>

**RemoteUrl** <https://github.com/cran/proxirr>

**RemoteRef** HEAD

**RemoteSha** f3c206278ed0564c4f18249a6e78d29e64961e3e

## Contents

alpha . . . . .	2
beta . . . . .	3
marxan_alphas . . . . .	5
marxan_betas . . . . .	6
marxan_run . . . . .	6

---

alpha	<i>Calculate Alpha Irreplaceability</i>
-------	---

---

### Description

This function calculates Alpha irreplaceability. Inputs can be single parameter values needed to calculate Alpha, vectors of parameter values to calculate a vector of Alpha values, or a `data.frame` with columns containing parameters needed to calculate a vector of Alpha values.

- **Single Alpha measurement:** If `local`, `global` and `target` are numeric, it calculates and returns Alpha irreplaceability.
- **Vectorized Alpha measurement:** If `local`, `global` and `target` are vectors of the same length, a vector of Alpha irreplaceability values will be calculated and returned.
- **Dataframe Alpha measurement:** If `df` is provided and `local`, `global` and `target` are strings representing field names in `df`, a vector of Alpha irreplaceability values will be calculated and returned. **Optionally:** If `alpha_col` is also provided, a `data.frame` identical to `df` will be returned with the calculated Alpha values in the `alpha_col` column.

### Usage

```
alpha(
  local,
  global,
  target,
  df = NULL,
  alpha_col = NULL,
  triage = FALSE,
  na.allow = NULL,
  overwrite = FALSE
)
```

### Arguments

<code>local</code>	number, vector or string - The feature's representation at the site, or the name of the column containing the feature's representation at the site.
<code>global</code>	number, vector or string - The feature's globally available representation, or the name of the column containing the feature's globally available representation.
<code>target</code>	number, vector or string - The feature's target, or the name of the column containing the feature's target.
<code>df</code>	<code>data.frame</code> - Optional; an input <code>data.frame</code> .
<code>alpha_col</code>	string - The name of the column where to write alpha values. If both <code>df</code> and <code>alpha_col</code> are provided, the output will be the input dataframe with the additional column.

trriage	logical - Should features with unachievable targets be ignored? Defaults to FALSE. If FALSE, these species will be always assigned an Alpha irreplaceability of 1 wherever they occur. If TRUE, these species will always be assigned an Alpha irreplaceability of 0.
na.allow	logical - Allow NA values in input? If TRUE, NA values in local, global or target values will result in NA being returned, otherwise an error will be raised. Defaults to FALSE for single Alpha calculations, and to TRUE for calculations over vectors or data.frames.
overwrite	logical - Should alpha_col be overwritten if it already exists?

### Value

A number, vector or data.frame

### Author(s)

Daniele Baisero, <daniele.baisero@gmail.com>

### Examples

```
alpha(1, 100, 45)
alpha(c(1,25,45), c(100,100,100), c(50,50,50))
dtfrm = data.frame(
  loc = c(1,25,45),
  glob = c(100,100,100),
  targ = c(50,50,50)
)
alpha('loc', 'glob', 'targ', df = dtfrm)
```

---

beta

*Calculate Beta Irreplaceability*

---

### Description

This function calculates Beta irreplaceability. Inputs can be either a vector of Alpha values, or a data.frame containing all necessary parameters needed to calculate Alpha values on a row-by-row basis.

- **Vector Beta measurement:** If data is a vector of Alpha irreplaceability values, a single Beta value will be calculated and returned.
- **Dataframe Beta measurement:** If data is a data.frame and local, global and target are strings representing field names in data, a vector of Alpha irreplaceability values will be calculated using these fields, and a Beta irreplaceability value will be calculated on these, and returned.

**Usage**

```
beta(  
  data,  
  local = NULL,  
  global = NULL,  
  target = NULL,  
  triage = FALSE,  
  na.rm = TRUE  
)
```

**Arguments**

<code>data</code>	vector or <code>data.frame</code> - The input over which to calculate Beta.
<code>local</code>	string - The name of the column containing the feature's representation at the site. Needed if <code>data</code> is a <code>data.frame</code>
<code>global</code>	string - The name of the column containing the feature's total available representation. Needed if <code>data</code> is a <code>data.frame</code>
<code>target</code>	string - The name of the column containing the feature's target. Needed if <code>data</code> is a <code>data.frame</code>
<code>triage</code>	logical - Should features with unachievable targets be ignored? Defaults to <code>FALSE</code> . If <code>FALSE</code> , these species will be always assigned an Alpha irreplaceability of 1 wherever they occur. If <code>TRUE</code> , these species will always be assigned an Alpha irreplaceability of 0.
<code>na.rm</code>	logical - Should lines with missing values (NA) be ignored? If <code>data</code> is a vector, NA values will be removed when calculating Beta. If <code>data</code> is a <code>data.frame</code> , Alpha values will be calculated using <code>alpha</code> with <code>na.allow</code> set to <code>TRUE</code> , and then Beta calculated ignoring NA values.

**Value**

A number

**Author(s)**

Daniele Baisero, <daniele.baisero@gmail.com>

**Examples**

```
beta(c(0.01, 0.5, 0.5))  
dtfrm = data.frame(  
  loc = c(1,25,45),  
  glob = c(100,100,100),  
  targ = c(50,50,50)  
)  
beta(dtfrm, local = 'loc', global = 'glob', target = 'targ')
```

---

`marxan_alphas`*Marxan: Alpha irreplaceabilities*

---

## Description

Given a valid Marxan input.dat file, it returns a copy of the PUVSPRNAME file (puvspr.dat), with additional columns containing the feature's global value, target value, and alpha irreplaceability.

## Usage

```
marxan_alphas(input, triage = FALSE)
```

## Arguments

<code>input</code>	string - The address of the input.dat file.
<code>triage</code>	logical - Should features with unachievable targets be given an irreplaceability of 0? See <a href="#">alpha</a> .

## Details

The global value is obtained as the sum of the feature's "amount" column in PUVSPRNAME.

The target value is obtained from the SPECNAME file (spec.dat), either as the proportion of the global value (if the "prop" column is present in SPECNAME), or as the "target" value in SPECNAME (otherwise). This behaviour mirrors Marxan's default behaviour.

## Value

A data.frame

## Author(s)

Daniele Baisero, <daniele.baisero@gmail.com>

## Examples

```
## Not run:  
marxan_alphas('/data/marxan/analysis01/input.dat')  
marxan_alphas('C:\data\marxan\analysis01\input.dat')  
  
## End(Not run)
```

---

marxan_betas	<i>Marxan: Beta irreplacabilities</i>
--------------	---------------------------------------

---

**Description**

Given a valid Marxan input.dat file, returns a copy of the PUNAME file (pu.dat), with an additional column containing the planning unit's beta irreplacability.

**Usage**

```
marxan_betas(input, triage = FALSE)
```

**Arguments**

input	string - The address of the input.dat file.
triage	logical - Should features with unachievable targets be given an irreplacability of 0? See <a href="#">beta</a> .

**Value**

A data.frame

**Author(s)**

Daniele Baisero, <daniele.baisero@gmail.com>

**Examples**

```
## Not run:
marxan_betas('/data/marxan/analysis01/input.dat')
marxan_betas('C:\data\marxan\analysis01\input.dat')

## End(Not run)
```

---

marxan_run	<i>Marxan: Save Alpha and Beta Irreplacabilities</i>
------------	--

---

**Description**

Saves the outputs of [marxan\\_alphas](#) and [marxan\\_betas](#) in the marxan output folder.

**Usage**

```
marxan_run(input, alphas = TRUE, betas = TRUE, triage = FALSE)
```

### Arguments

input	string - The address of the input.dat file.
alphas	logical - Should the Alpha Irreplaceability output be saved?
betas	logical - Should the Beta Irreplaceability output be saved?
trriage	logical - Should features with unachievable targets be given an irreplaceability of 0? See <a href="#">alpha</a> and <a href="#">beta</a> .

### Details

Two files ('\_proxirr\_alphas.csv', '\_proxirr\_betas.csv') will be created prefixed with the scenario name indicated in 'input.dat' ('SCENNAME').

### Value

TRUE

### Author(s)

Daniele Baisero, <daniele.baisero@gmail.com>

### Examples

```
## Not run:  
marxan_run('/data/marxan/analysis01/input.dat')  
marxan_run('C:\data\marxan\analysis01\input.dat')  
  
## End(Not run)
```

# Index

alpha, [2](#), [4](#), [5](#), [7](#)

beta, [3](#), [6](#), [7](#)

marxan\_alphas, [5](#), [6](#)

marxan\_betas, [6](#), [6](#)

marxan\_run, [6](#)