

# Package ‘opa’

May 3, 2023

**Type** Package

**Title** An Implementation of Ordinal Pattern Analysis

**Version** 0.8.1

**Description** Quantifies hypothesis to data fit for repeated measures and longitudinal data, as described by Thorngate (1987) <[doi:10.1016/S0166-4115\(08\)60083-7](https://doi.org/10.1016/S0166-4115(08)60083-7)> and Grice et al., (2015) <[doi:10.1177/2158244015604192](https://doi.org/10.1177/2158244015604192)>. Hypothesis and data are encoded as pairwise relative orderings which are then compared to determine the percentage of orderings in the data that are matched by the hypothesis.

**License** GPL (>= 3)

**URL** <https://github.com/timbeechey/opa>

**BugReports** <https://github.com/timbeechey/opa/issues>

**Encoding** UTF-8

**LinkingTo** Rcpp, RcppArmadillo

**Imports** Rcpp

**Suggests** tinytest

**SystemRequirements** C++17

**RoxygenNote** 7.2.3

**NeedsCompilation** yes

**Author** Timothy Beechey [aut, cre] (<<https://orcid.org/0000-0001-8858-946X>>)

**Maintainer** Timothy Beechey <[tim.beechey@proton.me](mailto:tim.beechey@proton.me)>

**Repository** CRAN

**Date/Publication** 2023-05-03 21:10:05 UTC

## R topics documented:

<code>compare_conditions</code> . . . . .	2
<code>compare_groups</code> . . . . .	3
<code>compare_hypotheses</code> . . . . .	3

cval_plot . . . . .	4
group_results . . . . .	5
individual_results . . . . .	6
opa . . . . .	6
pcc_plot . . . . .	9
plot.opafit . . . . .	9
plot_hypothesis . . . . .	10
print.opafit . . . . .	11
print.pairwiseopafit . . . . .	11
summary.opafit . . . . .	12

## Index 13

---

compare_conditions	<i>Calculates PCCs and c-values based on pairwise comparison of conditions.</i>
--------------------	---

---

### Description

Calculates PCCs and c-values based on pairwise comparison of conditions.

### Usage

```
compare_conditions(result, nreps = 1000L)
```

### Arguments

result	an object of class "opafit" produced by a call to opa().
nreps	an integer

### Value

compare\_conditions returns a list with the following elements

**pcc\_mat** A lower triangle matrix containing PCCs calculated from each pairing of data columns.  
**cval\_mat** A lower triangle matrix containing c-values calculated from each pairing of data columns.  
**pccs** A vector containing PCCs calculated from each pairing of data.  
**cvals** A vector containing c-values calculated from each pairing of data.  
**nreps** The number of permutations used to calculate the c-values.

### Examples

```
dat <- data.frame(t1 = c(9, 4, 8, 10),
                  t2 = c(8, 8, 12, 10),
                  t3 = c(8, 5, 10, 11),
                  t4 = c(10, 5, 11, 12))
opamod <- opa(dat, 1:4)
compare_conditions(opamod)
```

---

compare_groups	<i>Calculate the c-value of the difference in PCCs produced by two groups</i>
----------------	---

---

**Description**

Calculate the c-value of the difference in PCCs produced by two groups

**Usage**

```
compare_groups(m, group1, group2)
```

**Arguments**

**m** an object of class "opafit" produced by a call to opa().  
**group1** a character string which matches a group level passed to opa().  
**group2** a character string which matches a group level passed to opa().

**Value**

a list with the following elements

**pcc\_diff** A double. The absolute difference between group PCCs.

**cval** The chance that an absolute difference between PCCs generated by random ordering is greater than or equal to the difference in observed group PCCs.

**Examples**

```
dat <- data.frame(group = c("a", "b", "a", "b"),
                  t1 = c(9, 4, 8, 10),
                  t2 = c(8, 8, 12, 10),
                  t3 = c(8, 5, 10, 11))
dat$group <- factor(dat$group, levels = c("a", "b"))
opamod <- opa(dat[,2:4], 1:3, group = dat$group)
compare_groups(opamod, "a", "b")
```

---

compare_hypotheses	<i>Calculate the c-value of the difference in PCCs produced by two hypotheses</i>
--------------------	---

---

**Description**

Calculate the c-value of the difference in PCCs produced by two hypotheses

**Usage**

```
compare_hypotheses(m1, m2)
```

**Arguments**

**m1** an object of class "opafit" produced by a call to opa().  
**m2** an object of class "opafit" produced by a call to opa().

**Value**

a list with the following elements

**pcc\_diff** A double. The absolute difference between group PCCs of the two opafit objects.

**cval** The chance that an absolute difference between PCCs generated by random ordering is greater than or equal to the difference in observed group PCCs.

**Examples**

```
dat <- data.frame(t1 = c(9, 4, 8, 10),
                 t2 = c(8, 8, 12, 10),
                 t3 = c(8, 5, 10, 11),
                 t4 = c(10, 5, 11, 12))
opamod1 <- opa(dat, c(1, 2, 3, 4))
opamod2 <- opa(dat, c(1, 4, 2, 3))
compare_hypotheses(opamod1, opamod2)
```

---

cval\_plot

*Plot individual chance values*


---

**Description**

Plot individual chance values

**Usage**

```
cval_plot(m, threshold = NULL, title = TRUE, legend = TRUE)
```

**Arguments**

**m** an object of class "opafit"  
**threshold** a boolean indicating whether to plot a threshold abline  
**title** a boolean indicating whether to include a plot title  
**legend** a boolean indicating whether to include a legend when n groups > 1

**Value**

No return value, called for side effects.

**Examples**

```
dat <- data.frame(t1 = c(9, 4, 8, 10),
                 t2 = c(8, 8, 12, 10),
                 t3 = c(8, 5, 10, 11))
opamod <- opa(dat, 1:3)
cval_plot(opamod)
cval_plot(opamod, threshold = 0.1)
```

---

`group_results`*Group-level PCC and chance values.*

---

**Description**

Group-level PCC and chance values.

**Usage**

```
group_results(m, digits)
```

**Arguments**

`m` an object of class "opafit" produced by `opa()`.  
`digits` a positive integer.

**Details**

If the model was fitted with no grouping variable, a single PCC and c-value are returned. If a grouping variable was specified in the call to `opa` then PCCs and c-values are returned for each factor level of the grouping variable.

**Value**

a matrix with 1 row per group.

**Examples**

```
dat <- data.frame(t1 = c(9, 4, 8, 10),
                 t2 = c(8, 8, 12, 10),
                 t3 = c(8, 5, 10, 11))
opamod <- opa(dat, 1:3)
group_results(opamod)
```

---

`individual_results`      *Individual-level PCC and chance values.*

---

### Description

Individual-level PCC and chance values.

### Usage

```
individual_results(m, digits)
```

### Arguments

`m`                      an object of class "opafit" produced by `opa()`  
`digits`                  an integer

### Details

If the model was fitted with no grouping variable, a matrix of PCCs and c-values are returned corresponding to the order of rows in the data. If the `opa` model was fitted with a grouping variable specified, a table of PCCs and c-values is returned ordered by factor level of the grouping variable.

### Value

a matrix containing a column of PCC values and a column of c-values with 1 row per row of data.

### Examples

```
dat <- data.frame(t1 = c(9, 4, 8, 10),  
                 t2 = c(8, 8, 12, 10),  
                 t3 = c(8, 5, 10, 11))  
opamod <- opa(dat, 1:3)  
individual_results(opamod)
```

---

`opa`                      *Fit an ordinal pattern analysis model*

---

### Description

`opa` is used to fit ordinal pattern analysis models by computing the percentage of pair orderings in each row of data which are matched by corresponding pair orderings in an hypothesis, in addition the chance of a permutation of the data producing a percentage match as great.

**Usage**

```
opa(
  dat,
  hypothesis,
  group = NULL,
  pairing_type = "pairwise",
  diff_threshold = 0,
  nreps = 1000L
)
```

**Arguments**

<code>dat</code>	a data frame
<code>hypothesis</code>	a numeric vector
<code>group</code>	an optional factor vector
<code>pairing_type</code>	a string
<code>diff_threshold</code>	a positive integer or floating point number
<code>nreps</code>	an integer, ignored if <code>cval_method = "exact"</code>

**Details**

Data is expected in **wide** format with 1 row per individual and 1 column per measurement condition. Data must contain only columns consisting of numerical values of the *dependent* variable.

The length of the hypothesis must be equal to the number of columns in the dependent variable data.frame `dat`.

Any *independent* variable must be passed separately as a vector with the `group` keyword. The grouping vector must be a *factor*.

`pairing_type` must be either "pairwise" or "adjacent". The "pairwise" option considered the relative ordering of every pair of observations in the data and every pair of elements of the hypothesis. The "adjacent" option considers the ordering of adjacent pairs only. If unspecified, the default is "pairwise".

`diff_threshold` may be a positive integer or double. If unspecified a default zero threshold is used. The `diff_threshold` is never applied to the hypothesis.

`nreps` specifies the number of random reorderings to use in the calculation of chance-values.

**Value**

`opa` returns an object of class "opafit".

An object of class "opafit" is a list containing the following components:

**group\_pcc** the percentage of pairwise orderings from all pooled data rows which were correctly classified by the hypothesis.

**individual\_pccs** a vector containing the percentage of pairwise orderings that were correctly classified by the hypothesis for each data row.

- correct\_pairs** an integer representing the number of pairwise orderings pooled across all data rows that were correctly classified by the hypothesis.
- total\_pairs** an integer, the number of pair orderings contained in the data.
- group\_cval** the group-level chance value.
- individual\_cvals** a vector containing chance values for each data row
- rand\_pccs** A vector of PCCS calculated from each random ordering with length equal to nreps, a list of vectors if a group vector was passed to opa().
- call** The matched call
- hypothesis** The hypothesis vector passed to opa()
- pairing\_type** A string indicating the method of pairing passed to opa().
- diff\_threshold** The numeric difference threshold used to calculate PCCs. If no value was passed in the diff\_threshold, the default of 0 is used.
- data** The data.frame passed to opa().
- groups** The vector of groups passed to opa. If no group vector was passed to opa() the default of NULL is used.
- nreps** an integer, the number of random re-orderings of the data used to compute chance values.

## References

- Grice, J. W., Craig, D. P. A., & Abramson, C. I. (2015). A Simple and Transparent Alternative to Repeated Measures ANOVA. *SAGE Open*, 5(3), 215824401560419. <<https://doi.org/10.1177/2158244015604192>>
- Thorngate, W. (1987). Ordinal Pattern Analysis: A Method for Assessing Theory-Data Fit. *Advances in Psychology*, 40, 345–364. <[https://doi.org/10.1016/S0166-4115\(08\)60083-7](https://doi.org/10.1016/S0166-4115(08)60083-7)>

## Examples

```
dat <- data.frame(group = c("a", "b", "a", "b"),
                 t1 = c(9, 4, 8, 10),
                 t2 = c(8, 8, 12, 10),
                 t3 = c(8, 5, 10, 11))
dat$group <- factor(dat$group, levels = c("a", "b"))
opamod <- opa(dat[,2:4], 1:3)
opa(dat[,2:4], 1:3)
opa(dat[,2:4], 1:3, nreps = 500)
opa(dat[,2:4], 1:3, pairing_type = "adjacent")
opa(dat[,2:4], 1:3, diff_threshold = 1)
opa(dat[,2:4], 1:3, group = dat$group)
```

---

pcc\_plot *Plot individual PCCs.*

---

**Description**

Plot individual PCCs.

**Usage**

```
pcc_plot(m, threshold = NULL, title = TRUE, legend = TRUE)
```

**Arguments**

m	an object of class "opafit"
threshold	a boolean indicating whether to plot a threshold abline
title	a boolean indicating whether to include a plot title
legend	a boolean indicating whether to include a legend when n groups > 1

**Value**

No return value, called for side effects.

**Examples**

```
dat <- data.frame(t1 = c(9, 4, 8, 10),
                 t2 = c(8, 8, 12, 10),
                 t3 = c(8, 5, 10, 11))
opamod <- opa(dat, 1:3)
pcc_plot(opamod)
pcc_plot(opamod, threshold = 85)
```

---

plot.opafit *Plots individual-level PCCs and chance-values.*

---

**Description**

Plots individual-level PCCs and chance-values.

**Usage**

```
## S3 method for class 'opafit'
plot(x, pcc_threshold = NULL, cval_threshold = NULL, ...)
```

**Arguments**

x an object of class "opafit" produced by opa()  
pcc\_threshold a number used as the x-intercept to plot a PCC threshold abline  
cval\_threshold a number used as the x-intercept to plot a c-value threshold abline  
... ignored

**Value**

No return value, called for side effects.

**Examples**

```
dat <- data.frame(t1 = c(9, 4, 8, 10),
                 t2 = c(8, 8, 12, 10),
                 t3 = c(8, 5, 10, 11))
opamod <- opa(dat, 1:3)
plot(opamod)
```

---

plot\_hypothesis      *Plot a hypothesis.*

---

**Description**

Plot a hypothesis.

**Usage**

```
plot_hypothesis(h, title = TRUE)
```

**Arguments**

h a numeric vector  
title a boolean indicating whether to include a plot title

**Value**

No return value, called for side effects.

**Examples**

```
h <- c(1,2,3,3,3)
plot_hypothesis(h)
```

---

print.opafit	<i>Displays the call used to fit an ordinal pattern analysis model.</i>
--------------	---

---

**Description**

Displays the call used to fit an ordinal pattern analysis model.

**Usage**

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

**Arguments**

x	an object of class "opafit".
...	ignored

**Value**

No return value, called for side effects.

**Examples**

```
dat <- data.frame(t1 = c(9, 4, 8, 10),  
                 t2 = c(8, 8, 12, 10),  
                 t3 = c(8, 5, 10, 11))  
opamod <- opa(dat, 1:3)  
print(opamod)
```

---

print.pairwiseopafit	<i>Displays the results of a pairwise ordinal pattern analysis.</i>
----------------------	---

---

**Description**

Displays the results of a pairwise ordinal pattern analysis.

**Usage**

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

**Arguments**

x	an object of class "pairwiseopafit".
...	ignored

**Value**

No return value, called for side effects.

**Examples**

```
dat <- data.frame(t1 = c(9, 4, 8, 10),
                 t2 = c(8, 8, 12, 10),
                 t3 = c(8, 5, 10, 11))
opamod <- opa(dat, 1:3)
pw <- compare_conditions(opamod)
print(pw)
print(pw, digits = 2)
```

---

summary.opafit	<i>Prints a summary of results from a fitted ordinal pattern analysis model.</i>
----------------	--

---

**Description**

Prints a summary of results from a fitted ordinal pattern analysis model.

**Usage**

```
## S3 method for class 'opafit'
summary(object, ..., digits = 2L)
```

**Arguments**

object	an object of class "opafit".
...	ignored
digits	an integer used for rounding values in the output.

**Value**

No return value, called for side effects.

**Examples**

```
dat <- data.frame(t1 = c(9, 4, 8, 10),
                 t2 = c(8, 8, 12, 10),
                 t3 = c(8, 5, 10, 11))
opamod <- opa(dat, 1:3)
summary(opamod)
summary(opamod, digits = 3)
```

# Index

`compare_conditions`, 2  
`compare_groups`, 3  
`compare_hypotheses`, 3  
`cval_plot`, 4  
  
`group_results`, 5  
  
`individual_results`, 6  
  
`opa`, 6  
  
`pcc_plot`, 9  
`plot.opafit`, 9  
`plot_hypothesis`, 10  
`print.opafit`, 11  
`print.pairwiseopafit`, 11  
  
`summary.opafit`, 12