Package: qcTAF (via r-universe)

June 9, 2026

Version 2026.05.23

Title TAF Checks

Imports TAF

Description Tools to check TAF analyses.

License GPL-3

URL https://github.com/ices-tools-prod/qcTAF

Encoding UTF-8

Config/roxygen2/version 8.0.0

Roxygen list(old_usage=TRUE)

Repository https://ices-tools-prod.r-universe.dev

Date/Publication 2026-06-09 02:34:47 UTC

RemoteUrl https://github.com/ices-tools-prod/qcTAF

RemoteRef HEAD

RemoteSha ae7861eee42d570292e70f8b2bae042e90ff759a

Contents

---

qcTAF-package *TAF Checks*

Description

Tools to check TAF analyses.

Details

Quality control:

qc	run all quality checks
qc.boot.exists	boot directory exists
qc.data.bib.exists	DATA.bib exists
qc.data.bib.valid	DATA.bib is valid
qc.data.bib.processed	DATA.bib processed
qc.data.declared	data files are declared
qc.software.bib.exists	SOFTWARE.bib exists
qc.software.bib.valid	SOFTWARE.bib is valid
qc.software.bib.processed	SOFTWARE.bib processed
qc.software.declared	software files are declared
qc.initial.data	initial data files
qc.only.relative.paths	scripts use only relative paths
qc.any.scripts.exist	any scripts exist
qc.all.scripts.exist	all scripts exist

Author(s)

Arni Magnusson and Colin Millar.

See Also

Useful links:

- https://github.com/ices-tools-prod/qcTAF

qc *Quality Check*

Description

Run all quality checks for a given analysis.

Usage

```
qc(analysis = ".")
```

Arguments

`analysis`	directory containing a TAF analysis.

Value

Logical vector indicating the result from each test.

See Also

The checks are run in the following order:

`dir.exists` checks if directory exists.

`qc.boot.exists` checks if boot directory exists.

`qc.data.bib.exists` checks if `DATA.bib` exists.

`qc.data.bib.valid` checks if `DATA.bib` is valid.

`qc.data.bib.processed` checks if `DATA.bib` entries have been processed.

`qc.data.declared` checks if data files are declared.

`qc.software.bib.exists` checks if `SOFTWARE.bib` exists.

`qc.software.bib.valid` checks if `SOFTWARE.bib` is valid.

`qc.software.bib.processed` checks if `SOFTWARE.bib` entries have been processed.

`qc.data.declared` checks if software files are declared.

`qc.initial.data` checks if initial data files have the same contents as boot data files.

`qc.any.scripts.exist` checks if any scripts exist.

`qc.all.scripts.exist` checks if all scripts exist.

`qc.only.relative.paths` checks if scripts use only relative paths.

`qcTAF-package` gives an overview of the package.

Examples

```
## Not run:
qc("rjm-347d")

## End(Not run)
```

qc.all.scripts.exist *All Scripts Exist*

Description

Check if all TAF scripts (`data.R`, `model.R`, `output.R`, and `report.R`) exist.

Usage

```
qc.all.scripts.exist(analysis = ".")
```

Arguments

`analysis`	directory containing a TAF analysis.

Value

`TRUE` if test succeeds, otherwise `FALSE`.

See Also

`qc` runs all `qc.*` tests.

`qcTAF-package` gives an overview of the package.

Examples

```
## Not run:
qc.all.scripts.exist("rjm-347d")

## End(Not run)
```

qc.any.scripts.exist *Any Scripts Exist*

Description

Check if any TAF scripts (`data.R`, `model.R`, `output.R`, and `report.R`) exist.

Usage

```
qc.any.scripts.exist(analysis = ".")
```

Arguments

`analysis`	directory containing a TAF analysis.

Value

TRUE if test succeeds, otherwise FALSE.

See Also

qc runs all qc.* tests.

qcTAF-package gives an overview of the package.

Examples

```
## Not run:
qc.any.scripts.exist("rjm-347d")

## End(Not run)
```

---

qc.boot.exists *Boot Directory Exists*

Description

Check if a boot directory exists.

Usage

```
qc.boot.exists(analysis = ".")
```

Arguments

analysis	directory containing a TAF analysis.

Value

TRUE if test succeeds, otherwise FALSE.

See Also

qc runs all qc.* tests.

qcTAF-package gives an overview of the package.

Examples

```
## Not run:
qc.boot.exists("rjm-347d")

## End(Not run)
```

qc.data.bib.exists *DATA.bib Exists*

Description

Check if a DATA.bib file exists.

Usage

```
qc.data.bib.exists(analysis = ".")
```

Arguments

analysis	directory containing a TAF analysis.

Value

TRUE if test succeeds, otherwise FALSE.

See Also

qc runs all qc.* tests.

qcTAF-package gives an overview of the package.

Examples

```
## Not run:
qc.data.bib.exists("rjm-347d")

## End(Not run)
```

qc.data.bib.processed *DATA.bib Processed*

Description

Check if a DATA.bib entries appear to have been processed, i.e., found as files and directories inside the boot folder.

Usage

```
qc.data.bib.processed(analysis = ".")
```

Arguments

analysis	directory containing a TAF analysis.

Value

TRUE if test succeeds, otherwise FALSE.

Note

qc.data.bib.processed checks if all bib entries have made it into the boot data folder:

```
DATA.bib | boot/data
---------+----------
    A    |
    B   ==>
    C    |
```

qc.data.declared checks if all boot data files were declared as bib entries:

```
DATA.bib | boot/data
---------+----------
         |    A
        <==   B
         |    C
```

When used together, those two functions ensure that DATA.bib and boot/data contain precisely the same entries.

See Also

qc runs all qc.* tests.

qcTAF-package gives an overview of the package.

Examples

```
## Not run:
qc.data.bib.processed("rjm-347d")

## End(Not run)
```

---

qc.data.bib.valid *DATA.bib Valid*

Description

Check if a DATA.bib file is valid.

Usage

```
qc.data.bib.valid(analysis = ".")
```

Arguments

analysis	directory containing a TAF analysis.

Value

TRUE if test succeeds, otherwise FALSE.

See Also

qc runs all qc.* tests.

qcTAF-package gives an overview of the package.

Examples

```
## Not run:
qc.data.bib.valid("rjm-347d")

## End(Not run)
```

---

qc.data.declared *Data Declared*

Description

Check if all files in the boot data folder are declared in the DATA.bib file.

Usage

```
qc.data.declared(analysis = ".")
```

Arguments

analysis	directory containing a TAF analysis.

Value

TRUE if test succeeds, otherwise FALSE.

Note

qc.data.bib.processed checks if all bib entries have made it into the boot data folder:

```
DATA.bib | boot/data
---------+----------
    A    |
    B   ==>
    C    |
```

qc.data.declared checks if all boot data files were declared as bib entries:

```
DATA.bib | boot/data
---------+----------
         |    A
        <==   B
         |    C
```

When used together, those two functions ensure that DATA.bib and boot/data contain precisely the same entries.

See Also

qc runs all qc.* tests.

qcTAF-package gives an overview of the package.

Examples

```
## Not run:
qc.data.declared("rjm-347d")

## End(Not run)
```

---

qc.initial.data *Initial Data*

Description

Check if initial data files have the same contents as boot data files.

Usage

```
qc.initial.data(analysis = ".")
```

Arguments

analysis	directory containing a TAF analysis.

Value

TRUE if test succeeds, otherwise FALSE.

See Also

qc runs all qc.* tests.

qcTAF-package gives an overview of the package.

Examples

```
## Not run:
qc.initial.data("rjm-347d")

## End(Not run)
```

`qc.only.relative.paths` *Only Relative Paths*

Description

Check if scripts (`*.R`, `*.Rmd`, `Rnw`, `*.qmd`) use only relative paths.

Usage

```
qc.only.relative.paths(analysis = ".")
```

Arguments

`analysis`	directory containing a TAF analysis.

Value

`TRUE` if test succeeds, otherwise `FALSE`.

See Also

`qc` runs all `qc.*` tests.

`qcTAF-package` gives an overview of the package.

Examples

```
## Not run:
qc.only.relative.paths("rjm-347d")

## End(Not run)
```

qc.software.bib.exists *SOFTWARE.bib Exists*

Description

Check if a SOFTWARE.bib file exists.

Usage

```
qc.software.bib.exists(analysis = ".")
```

Arguments

analysis	directory containing a TAF analysis.

Value

TRUE if test succeeds, otherwise FALSE.

See Also

qc runs all qc.* tests.

qcTAF-package gives an overview of the package.

Examples

```
## Not run:
qc.software.bib.exists("rjm-347d")

## End(Not run)
```

qc.software.bib.processed *SOFTWARE.bib Processed*

Description

Check if a SOFTWARE.bib entries appear to have been processed, i.e., found as files and directories inside the boot folder.

Usage

```
qc.software.bib.processed(analysis = ".")
```

Arguments

analysis	directory containing a TAF analysis.

Value

TRUE if test succeeds, otherwise FALSE.

Note

qc.software.bib.processed checks if all bib entries have made it into the boot software folder:

```
SOFTWARE.bib | boot/software
-------------+--------------
      A      |
      B     ==>
      C      |
```

qc.software.declared checks if all boot software files were declared as bib entries:

```
SOFTWARE.bib | boot/software
-------------+--------------
             |      A
            <==     B
             |      C
```

When used together, those two functions ensure that SOFTWARE.bib and boot/software contain precisely the same entries.

See Also

qc runs all qc.* tests.

qcTAF-package gives an overview of the package.

Examples

```
## Not run:
qc.software.bib.processed("rjm-347d")

## End(Not run)
```

qc.software.bib.valid *SOFTWARE.bib Valid*

Description

Check if a SOFTWARE.bib file is valid.

Usage

```
qc.software.bib.valid(analysis = ".")
```

Arguments

analysis	directory containing a TAF analysis.

Value

TRUE if test succeeds, otherwise FALSE.

See Also

qc runs all qc.* tests.

qcTAF-package gives an overview of the package.

Examples

```
## Not run:
qc.software.bib.valid("rjm-347d")

## End(Not run)
```

qc.software.declared *Software Declared*

Description

Check if all files in the boot software folder are declared in the SOFTWARE.bib file.

Usage

```
qc.software.declared(analysis = ".")
```

Arguments

analysis	directory containing a TAF analysis.

Value

TRUE if test succeeds, otherwise FALSE.

Note

qc.software.bib.processed checks if all bib entries have made it into the boot software folder:

```
SOFTWARE.bib | boot/software
-------------+--------------
      A      |
      B     ==>
      C      |
```

qc.software.declared checks if all boot software files were declared as bib entries:

```
SOFTWARE.bib | boot/software
-------------+--------------
             |      A
            <==     B
             |      C
```

When used together, those two functions ensure that SOFTWARE.bib and boot/software contain precisely the same entries.

See Also

qc runs all qc.* tests.

qcTAF-package gives an overview of the package.

Examples

```
## Not run:
qc.software.declared("rjm-347d")

## End(Not run)
```

Index