

Package ‘qat’

February 15, 2012

Type Package

Title Quality Assurance Toolkit

Version 0.53

Date 2012-02-14

Author Andre Duesterhus

Maintainer Andre Duesterhus <andue@uni-bonn.de>

Description Performing and documenting scientific quality assurance of meteorological data.

Depends R (>= 2.6.1), ncdf, gplots, XML, gdata, moments, boot

License GPL-2

Repository CRAN

Date/Publication 2012-02-15 08:30:01

R topics documented:

| | |
|---------------------------------------------|----|
| qat-package | 3 |
| qat_add_algorithm | 4 |
| qat_add_all_algorithms | 5 |
| qat_add_all_descriptions | 6 |
| qat_add_comment | 7 |
| qat_add_description | 8 |
| qat_add_resultfile | 9 |
| qat_analyse_block_distribution_1d | 10 |
| qat_analyse_boot_distribution_1d | 11 |
| qat_analyse_distribution_1d | 12 |
| qat_analyse_lim_rule_dynamic_1d | 14 |
| qat_analyse_lim_rule_sigma_1d | 15 |
| qat_analyse_lim_rule_static_1d | 17 |
| qat_analyse_noc_rule_1d | 18 |
| qat_analyse_roc_rule_dynamic_1d | 19 |

| | |
|-----------------------------------------------|----|
| qat_analyse_roc_rule_static_1d | 21 |
| qat_analyse_set_addup_1d | 22 |
| qat_analyse_set_mean_1d | 23 |
| qat_analyse_set_nans_1d | 24 |
| qat_analyse_set_nans_above_1d | 25 |
| qat_analyse_set_nans_below_1d | 26 |
| qat_analyse_slide_distribution_1d | 27 |
| qat_analyse_trimmed_distribution_1d | 28 |
| qat_call_block_distribution | 29 |
| qat_call_boot_distribution | 30 |
| qat_call_distribution | 32 |
| qat_call_lim_rule | 33 |
| qat_call_noc_rule | 34 |
| qat_call_plot_block_distribution | 36 |
| qat_call_plot_boot_distribution | 37 |
| qat_call_plot_distribution | 38 |
| qat_call_plot_lim_rule | 39 |
| qat_call_plot_noc_rule | 41 |
| qat_call_plot_roc_rule | 42 |
| qat_call_plot_slide_distribution | 43 |
| qat_call_plot_trimmed_distribution | 45 |
| qat_call_roc_rule | 46 |
| qat_call_save_block_distribution | 47 |
| qat_call_save_boot_distribution | 49 |
| qat_call_save_distribution | 50 |
| qat_call_save_lim_rule | 51 |
| qat_call_save_noc_rule | 53 |
| qat_call_save_roc_rule | 54 |
| qat_call_save_set_addup | 55 |
| qat_call_save_set_mean | 57 |
| qat_call_save_set_nans | 58 |
| qat_call_save_slide_distribution | 59 |
| qat_call_save_trimmed_distribution | 60 |
| qat_call_set_addup | 61 |
| qat_call_set_mean | 63 |
| qat_call_set_nans | 64 |
| qat_call_slide_distribution | 65 |
| qat_call_trimmed_distribution | 66 |
| qat_config_read_workflow | 67 |
| qat_config_write_workflow | 68 |
| qat_data_nameofvars_ncdf | 70 |
| qat_data_numofvars_ncdf | 71 |
| qat_data_read_ncdf | 72 |
| qat_data_varcontent_ncdf | 72 |
| qat_plot_block_distribution_1d | 73 |
| qat_plot_boot_distribution_1d | 74 |
| qat_plot_distribution_1d | 75 |
| qat_plot_lim_rule_dynamic_1d | 76 |

| | |
|--------------------------------------------|-----|
| qat_plot_lim_rule_sigma_1d | 77 |
| qat_plot_lim_rule_static_1d | 79 |
| qat_plot_noc_rule_1d | 80 |
| qat_plot_roc_rule_dynamic_1d | 81 |
| qat_plot_roc_rule_static_1d | 82 |
| qat_plot_slide_distribution_1d | 83 |
| qat_plot_trimmed_distribution_1d | 84 |
| qat_read_parameter | 85 |
| qat_run_workflow_check | 86 |
| qat_run_workflow_plot | 87 |
| qat_run_workflow_save | 89 |
| qat_save_block_distribution_1d | 90 |
| qat_save_boot_distribution_1d | 91 |
| qat_save_distribution_1d | 92 |
| qat_save_lim_rule_dynamic_1d | 93 |
| qat_save_lim_rule_sigma_1d | 94 |
| qat_save_lim_rule_static_1d | 95 |
| qat_save_noc_rule_1d | 96 |
| qat_save_result_ncdf | 97 |
| qat_save_roc_rule_dynamic_1d | 98 |
| qat_save_roc_rule_static_1d | 99 |
| qat_save_set_addup_1d | 100 |
| qat_save_set_mean_1d | 101 |
| qat_save_set_nans_1d | 102 |
| qat_save_set_nans_above_1d | 103 |
| qat_save_set_nans_below_1d | 104 |
| qat_save_slide_distribution_1d | 105 |
| qat_save_trimmed_distribution_1d | 106 |
| qat_style_plot | 107 |

Index**109**

qat-package

*Quality Assurance Toolkit***Description**

Performing and documenting scientific quality assurance of meteorological data.

Details

Package: qat
 Type: Package
 Version: 0.1
 Date: 2010-08-10
 License: GPL-2

Author(s)

Andre Duesterhus
 Maintainer: Andre Duesterhus <andue@uni-bonn.de>

Examples

```
library("qat")
# define testvector
testvector<-rnorm(500)
# read in workflow from systemfiles
filename_in <- system.file("extdata/workflowexample.xml", package="qat")
workflowlist <- qat_config_read_workflow(filename_in)
# define some additional vectors
maxlim <- seq(3,1,length.out=500)
minlim <- seq(-1,-3,length.out=500)
uproc <- seq(1,3,length.out=500)
downroc <- seq(3,1,length.out=500)
# run the workflow on the testvector
rlist <- qat_run_workflow_check(testvector,workflowlist,vec1=maxlim, vec2=minlim, vec3=uproc, vec4=downroc)
# produce some plots of the result in teh current directory
qat_run_workflow_plot(rlist, measurement_name="Test", basename="test")
# add some more informations for the workflow
workflowlist <- qat_add_all_descriptions(workflowlist)
workflowlist <- qat_add_all_algorithms(workflowlist)
workflowlist <- qat_add_comment(workflowlist, 1, "No problems")

filename_out <- "myworkflow_result.xml"
# write edited workflow in current directory
qat_config_write_workflow(workflowlist, output_filename=filename_out)
```

qat_add_algorithm *Algorithm of a check*

Description

For each check in the workflow it is possible to add a algorithm of the test. This will be saved into the XML result file under agolgorithm. This function adds a new or replace an existing algorithm.

Usage

```
qat_add_algorithm(workflowlist, listelem, algorithm_text)
```

Arguments

workflowlist A workflowlist like it will be created by qat_config_read_workflow
 listelem Number of check, where the algorithm should be added.
 algorithm_text Text of the algorithm.

Value

Give back the edited workflowlist.

Author(s)

Andre Duesterhus

See Also

[qat_config_read_workflow](#)

Examples

```
library("qat")
# read in workflow from systemfiles
filename_in <- system.file("extdata/workflowexample.xml", package="qat")
workflowlist <- qat_config_read_workflow(filename_in)
# add some more informations for the workflow
workflowlist <- qat_add_algorithm(workflowlist, 1, "Algorithm information")
filename_out <- "myworkflow_result.xml"
# write edited workflow in current directory
qat_config_write_workflow(workflowlist, output_filename=filename_out)
```

qat_add_all_algorithms

Adds all algorithms to a workflow

Description

For each check in the workflow it is possible to add a algorithm of the test. This will be saved into the XML result file under agolgorithm. This function adds for each test the known algorithm-information.

Usage

```
qat_add_all_algorithms(workflowlist)
```

Arguments

workflowlist A workflowlist like it will be created by qat_config_read_workflow

Details

This function use the informatio, which is stored in the system file qat_basetools.xml.

Value

Give back the edited workflowlist.

Author(s)

Andre Duesterhus

See Also

[qat_config_read_workflow](#)

Examples

```
library("qat")
# read in workflow from systemfiles
filename_in <- system.file("extdata/workflowexample.xml", package="qat")
workflowlist <- qat_config_read_workflow(filename_in)
# add some more informations for the workflow
workflowlist <- qat_add_all_descriptions(workflowlist)
workflowlist <- qat_add_all_algorithms(workflowlist)

filename_out <- "myworkflow_result.xml"
# write edited workflow in current directory
qat_config_write_workflow(workflowlist, output_filename=filename_out)
```

qat_add_all_descriptions

Adds all descriptions to a workflow

Description

For each check in the workflow it is possible to add a description of the test. This will be saved into the XML result file under the description. This function adds for each test the known description-information.

Usage

```
qat_add_all_descriptions(workflowlist)
```

Arguments

workflowlist A workflowlist like it will be created by `qat_config_read_workflow`

Details

This function use the informatio, which is stored in the system file `qat_basetools.xml`.

Value

Give back the edited workflowlist.

Author(s)

Andre Duesterhus

See Also[qat_config_read_workflow](#)**Examples**

```
library("qat")
# read in workflow from systemfiles
filename_in <- system.file("extdata/workflowexample.xml", package="qat")
workflowlist <- qat_config_read_workflow(filename_in)
# add some more informations for the workflow
workflowlist <- qat_add_all_descriptions(workflowlist)
workflowlist <- qat_add_all_algorithms(workflowlist)

filename_out <- "myworkflow_result.xml"
# write edited workflow in current directory
qat_config_write_workflow(workflowlist, output_filename=filename_out)
```

| | |
|-----------------|--------------------------|
| qat_add_comment | <i>Comment on result</i> |
|-----------------|--------------------------|

Description

For each check in the workflow it is possible to add a comment on the result of the test. This will be saved into the XML result file under the tag result/comment\on\result. This function adds a new or replace an existing comment.

Usage

```
qat_add_comment(workflowlist, listelem, comment_text)
```

Arguments

| | |
|--------------|-----------------------------------------------------------------------|
| workflowlist | A workflowlist like it will be created by qat_config_read_workflow |
| listelem | Number of check, which should be commented. |
| comment_text | Text of the comment |

Value

Give back the edited workflowlist.

Author(s)

Andre Duesterhus

See Also

[qat_config_read_workflow](#)

Examples

```
library("qat")
# read in workflow from systemfiles
filename_in <- system.file("extdata/workflowexample.xml", package="qat")
workflowlist <- qat_config_read_workflow(filename_in)
# add some more informations for the workflow
workflowlist <- qat_add_comment(workflowlist, 1, "No problems")
filename_out <- "myworkflow_result.xml"
# write edited workflow in current directory
qat_config_write_workflow(workflowlist, output_filename=filename_out)
```

| | |
|---------------------|-------------------------------|
| qat_add_description | <i>Description of a check</i> |
|---------------------|-------------------------------|

Description

For each check in the workflow it is possible to add a description of the test. This will be saved into the XML result file under the description. This function adds a new or replace an existing description.

Usage

```
qat_add_description(workflowlist, listelem, description_text)
```

Arguments

| | |
|------------------|---------------------------------------------------------------------------------|
| workflowlist | A workflowlist like it will be created by <code>qat_config_read_workflow</code> |
| listelem | Number of check, where the description should be added. |
| description_text | Text of the description. |

Value

Give back the edited workflowlist.

Author(s)

Andre Duesterhus

See Also

[qat_config_read_workflow](#)

Examples

```
library("qat")
# read in workflow from systemfiles
filename_in <- system.file("extdata/workflowexample.xml", package="qat")
workflowlist <- qat_config_read_workflow(filename_in)
# add some more informations for the workflow
workflowlist <- qat_add_description(workflowlist, 1, "How the test works...")
filename_out <- "myworkflow_result.xml"
# write edited workflow in current directory
qat_config_write_workflow(workflowlist, output_filename=filename_out)
```

qat_add_resultfile *Resultfile of a check*

Description

For each check in the workflow it is possible to add one or more result files of the test. This will be saved into the XML result file under result_file. This function adds a new resultfile.

Usage

```
qat_add_resultfile(workflowlist, listelem, resultfile_text)
```

Arguments

workflowlist A workflowlist like it will be created by qat_config_read_workflow.
listelem Number of check, where the resultfile should be added.
resultfile_text Text of the resultfile.

Value

Give back the edited workflowlist.

Author(s)

Andre Duesterhus

See Also

[qat_config_read_workflow](#)

Examples

```
library("qat")
# read in workflow from systemfiles
filename_in <- system.file("extdata/workflowexample.xml", package="qat")
workflowlist <- qat_config_read_workflow(filename_in)
# add some more informations for the workflow
workflowlist <- qat_add_resultfile(workflowlist, 1, "filename.png")
filename_out <- "myworkflow_result.xml"
# write edited workflow in current directory
qat_config_write_workflow(workflowlist, output_filename=filename_out)
```

```
qat_analyse_block_distribution_1d
```

Perform a block distribution check

Description

The measurement vector will be splitted into blocks, and on every block some statistical parameters will be calculated.

Usage

```
qat_analyse_block_distribution_1d(measurement_vector, blocksize)
```

Arguments

| | |
|--------------------|------------------------------------------------|
| measurement_vector | The measurement vector, which should be tested |
| blocksize | Length of the blocks |

Details

The measurement vector will be splitted into blocks, with the length of the given blocksize parameter. After this some statistical parameters will be calculated for every block. As a result a list will be given back, with these parameters, where every entry got a length of the length of the measurement vector divided by the blocksize, which is rounded down to the next integer.

Value

It returns a list with the following entries:

| | |
|--------------------|----------------------------------------------|
| first_moment | First moment of the measurement vector |
| second_moment | Second moment of the measurement vector |
| third_moment | Third moment of the measurement vector |
| fourth_moment | Fourth moment of the measurement vector |
| standard_deviation | Standard deviation of the measurement vector |

| | |
|--------------|-----------------------------------------------|
| skewness | Skewness of the measurement vector |
| kurtosis | Kurtosis of the measurement vector |
| median | Median of the measurement vector |
| p5_quantile | 5 percent quantile of the measurement vector |
| p95_quantile | 95 percent quantile of the measurement vector |
| p25_quantile | 25 percent quantile of the measurement vector |
| p75_quantile | 75 percent quantile of the measurement vector |
| blocksize | Length of the used blocks |

Author(s)

Andre Duesterhus

See Also

[qat_plot_block_distribution_1d](#)

Examples

```
vec <- rnorm(1000)
result <- qat_analyse_block_distribution_1d(vec, 50)
```

qat_analyse_boot_distribution_1d

Perform a bootstrapped distribution check

Description

The measurement vector will be bootstrapped and statistical parameters will be determined.

Usage

```
qat_analyse_boot_distribution_1d(measurement_vector, bootruns)
```

Arguments

measurement_vector

The measurement vector, which should be tested

bootruns

Number of bootstrap runs, which should be performed

Details

The measurement vector will be bootstrapped with the number of runs, which is given by the parameter bootruns. From each runs, some statistical parameters will be calculated and given back in the resultlist.

Value

It returns a list with the following entries:

| | |
|--------------------|-------------------------------------------------------------|
| first_moment | First moments of the bootstrapped measurement vector |
| second_moment | Second moments of the bootstrapped measurement vector |
| third_moment | Third moments of the bootstrapped measurement vector |
| fourth_moment | Fourth moments of the bootstrapped measurement vector |
| standard_deviation | Standard deviations of the bootstrapped measurement vector |
| skewness | Skewness of the bootstrapped measurement vector |
| kurtosis | Kurtosis of the bootstrapped measurement vector |
| median | Medians of the bootstrapped measurement vector |
| p5_quantile | 5 percent quantiles of the bootstrapped measurement vector |
| p95_quantile | 95 percent quantiles of the bootstrapped measurement vector |
| p25_quantile | 25 percent quantiles of the bootstrapped measurement vector |
| p75_quantile | 75 percent quantiles of the bootstrapped measurement vector |

Author(s)

Andre Duesterhus

See Also

[qat_plot_boot_distribution_1d](#)

Examples

```
vec <- rnorm(1000)
result <- qat_analyse_boot_distribution_1d(vec, 1000)
```

qat_analyse_distribution_1d

Perform a distribution check

Description

This check makes a histogram and gives back some statistical parameters of the given measurement vector.

Usage

```
qat_analyse_distribution_1d(measurement_vector, numofbars)
```

Arguments

| | |
|--------------------|------------------------------------------------|
| measurement_vector | The measurement vector, which should be tested |
| numofbars | Numbers of bars of the histogram plot |

Details

From a given measurement vector a histogram will be performed. The number of bars of this will be given by the parameter numofbars. Additionally some statistical parameters, like the first moments and some quantiles will be calculated.

Value

It returns a list with the following entries:

| | |
|--------------------|-----------------------------------------------|
| first_moment | First moment of the measurement vector |
| second_moment | Second moment of the measurement vector |
| third_moment | Third moment of the measurement vector |
| fourth_moment | Fourth moment of the measurement vector |
| standard_deviation | Standard deviation of the measurement vector |
| skewness | Skewness of the measurement vector |
| kurtosis | Kurtosis of the measurement vector |
| median | Median of the measurement vector |
| p5_quantile | 5 percent quantile of the measurement vector |
| p95_quantile | 95 percent quantile of the measurement vector |
| p25_quantile | 25 percent quantile of the measurement vector |
| p75_quantile | 75 percent quantile of the measurement vector |
| numofbars | Number of bars of the histogram |
| ... | Elements of the histogram |

Author(s)

Andre Duesterhus

See Also

[qat_plot_distribution_1d](#)

Examples

```
vec <- rnorm(1000)
result <- qat_analyse_distribution_1d(vec, 15)
```

qat_analyse_lim_rule_dynamic_1d

Perform a dynamic lim-rule-check

Description

This check tests data on whether it exceeds a dynamic threshold.

Usage

```
qat_analyse_lim_rule_dynamic_1d(measurement_vector, min_vector = NULL, max_vector = NULL, min_vector_name = NULL, max_vector_name = NULL, min_vector_identifier = NULL, max_vector_identifier = NULL)
```

Arguments

| | |
|-----------------------|--------------------------------------------------------------------------------------------------------------|
| measurement_vector | The measurement vector, which should be tested |
| min_vector | A vector which consists of the minimum threshold values, with the same dimension like the measurement vector |
| max_vector | A vector which consists of the maximum threshold values, with the same dimension like the measurement vector |
| min_vector_name | A name or title of the minimum vector, which will be given back in the result |
| max_vector_name | A name or title of the maximum vector, which will be given back in the result |
| max_vector_identifier | The identifier of the maximum vector |
| min_vector_identifier | The identifier of the minimum vector |

Details

This tests tests every element, on whether it exceeds the minimum or maximum threshold. The result will be given back as a list, which contains the result of the test as a flagvector and its parameters. For every element of the measurement vector the flagvector contains a -1, if its exceeding its dedicated minimum vector element, a 1, if its exceeding its dedicated maximum vector element, or a 0, when no exceeding has happend. NaN-values in the measurement vector will be given back as a 0 in the flagvector, NaN-values in the minimum or maximum-vector are considered as not existing. There is no checking, if the maximum-vector is greater than the minimum-vector.

Value

It returns a list with the following entries:

| | |
|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| flagvector | A vector of length of measurement vector. For every element of the measurement vector the flagvector contains a -1, if its exceeding its dedicated minimum vector element, a 1, if its exceeding its dedicated maximum vector element, or a 0, when no exceeding has happend. |
|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

min_vector Give back the given min_vector
max_vector Give back the given max_vector
min_vector_name Give back the given min_vector_name
max_vector_name Give back the given max_vector_name

Warning

There is no checking, if the maximum-vector is greater than the minimum-vector.

Author(s)

Andre Duesterhus

References

Meek, D.W., Hatfield, J.L. (1994) Data quality checking for single station meteorological databases, *_Agricultural and Forest Meteorology*_, *69* (1-2), 85-109.

See Also

[qat_plot_lim_rule_dynamic_1d](#), [qat_call_lim_rule](#), [qat_analyse_lim_rule_static_1d](#), [qat_analyse_lim_rule_...](#)

Examples

```
vec <- rnorm(1000)
min_vector<-seq(-1,-2,length.out=1000)
max_vector<-seq(1,2,length.out=1000)
result <- qat_analyse_lim_rule_dynamic_1d(vec, min_vector, max_vector, min_vector_name="minimum vector", max_vect...
```

qat_analyse_lim_rule_sigma_1d

Perform a sigma lim-rule-check

Description

This check tests data on whether it exceeds a threshold formed by multiple standard deviations away from the mean.

Usage

```
qat_analyse_lim_rule_sigma_1d(measurement_vector, sigma_factor)
```

Arguments

| | |
|--------------------|-----------------------------------------------------------------------------------------------|
| measurement_vector | The measurement vector, which should be tested |
| sigma_factor | Multiplier of standard derivation, which determin the maximum allowed deviation from the mean |

Details

First the mean and the standard derivation of the measurement vector will be calculated. After this the limits will be determined by

$$lim_{\pm} = \mu \pm f\sigma,$$

where f is the given sigma factor.

Value

It returns a list with the following entries:

| | |
|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| flagvector | A vector of length of measurement vector. For every element of the measurement vector the flagvector contains a -1, if its exceeding its dedicated minimum vector element, a 1, if its exceeding its dedicated maximum vector element, or a 0, when no exceeding has happend. |
| sigma_factor | Give back the given sigma_factor |
| meanofvector | Give back the calculated mean of the measurement vector |
| sdfvector | Give back the calculated standard deviation of the measurement vector |

Author(s)

Andre Duesterhus

References

Meek, D.W., Hatfield, J.L. (1994) Data quality checking for single station meteorological databases, *_Agricultural and Forest Meteorology_, *69* (1-2), 85-109.*

See Also

[qat_plot_lim_rule_sigma_1d](#), [qat_call_lim_rule](#), [qat_analyse_lim_rule_static_1d](#), [qat_analyse_lim_rule_dyn](#)

Examples

```
vec <- rnorm(1000)
result <- qat_analyse_lim_rule_sigma_1d(vec, 2)
```

qat_analyse_lim_rule_static_1d
Perform a static lim-rule-check

Description

This check tests data on whether it exceeds a static threshold.

Usage

```
qat_analyse_lim_rule_static_1d(measurement_vector, min_value, max_value)
```

Arguments

| | |
|--------------------|------------------------------------------------|
| measurement_vector | The measurement vector, which should be tested |
| min_value | The minimum threshold |
| max_value | The maximum threshold |

Details

This check tests every element, on whether it exceeds the minimum or maximum threshold. The result will be given back as a list, which contains the result of the test as a flagvector and its parameters. For every element of the measurement vector the flagvector contains a -1, if its exceeding the minimum value, a 1, if its exceeding the maximum value, or a 0, when no exceeding has happend.

Value

It returns a list with the following entries:

| | |
|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| flagvector | A vector of length of measurement vector. For every element of the measurement vector the flagvector contains a -1, if its exceeding the minimum value, a 1, if its exceeding the maximum value, or a 0, when no exceeding has happend. |
| min_value | Give back the given min_value |
| max_value | Give back the given max_value |

Warning

There is no checking, if the maximum-value is greater than the minimum-value.

Author(s)

Andre Duesterhus

References

Meek, D.W., Hatfield, J.L. (1994) Data quality checking for single station meteorological databases, *_Agricultural and Forest Meteorology_, *69* (1-2), 85-109.*

See Also

[qat_plot_lim_rule_static_1d](#), [qat_call_lim_rule](#), [qat_analyse_lim_rule_dynamic_1d](#), [qat_analyse_lim_rule_...](#)

Examples

```
vec <- rnorm(1000)
result <- qat_analyse_lim_rule_static_1d(vec, -2,2)
```

qat_analyse_noc_rule_1d

Perform a noc-rule-check

Description

This check tests data on whether it changes after a given amount of values.

Usage

```
qat_analyse_noc_rule_1d(measurement_vector, max_return_elements)
```

Arguments

measurement_vector

The measurement vector, which should be tested

max_return_elements

Number of coherent elements, which are allowed to have no change between the single values, without indicate an error

Details

This check tests the given measurement vector from the beginning to the end, how much values in a row got the same value. If the number of values, which is defined by max_return_elements prior to the actual element got the same value as the actual element, the resulting flagvector will be set to 1 on the actual position. Else it will be set to 0.

Value

It returns a list with the following entries:

flagvector flagvektor with the dimension of measurement vector, where a 0 indicates no error and a 1 that there is a repetition error

max_return_elements

Give back the given max_return_elements

Author(s)

Andre Duesterhus

References

Meek, D.W., Hatfield, J.L. (1994) Data quality checking for single station meteorological databases, *_Agricultural and Forest Meteorology_, *69* (1-2), 85-109.*

See Also

[qat_plot_noc_rule_1d](#), [qat_call_noc_rule](#)

Examples

```
vec <- c(1,2,3,4,4,4,5,5,4,3,NaN,3,2,1)
result <- qat_analyse_noc_rule_1d(vec, 1)
```

```
qat_analyse_roc_rule_dynamic_1d
      Perform a dynamic roc-rule-check
```

Description

This check tests data on whether the change between two consecutive data points exceeds a dynamic threshold.

Usage

```
qat_analyse_roc_rule_dynamic_1d(measurement_vector, max_upward_vector = NULL, max_downward_vector = N
```

Arguments

```
measurement_vector
      The measurement vector, which should be tested
max_upward_vector
      A vector which consists of the threshold values for upward changes, with the
      same dimension like the measurement vector
max_downward_vector
      A vector which consists of the threshold values for downward changes, with the
      same dimension like the measurement vector and have to be positive definite
upward_vector_name
      A name or title of the upward vector, which will be given back in the result
downward_vector_name
      A name or title of the downward vector, which will be given back in the result
upward_vector_identifier
      The identifier of the upward vector
downward_vector_identifier
      The identifier of the downward vector
```

Details

This check tests two consecutive elements, on whether the change of values between those two exceeds the upward or downward threshold. The result will be given back as a list, which contains the result of the test as a flagvector and its parameters. For every change between two elements of the measurement vector the flagvector contains a -1, if its exceeding its dedicated downward vector element, a 1, if its exceeding its dedicated upward vector element, or a 0, when no exceeding has happened. NaN-values in the measurement vector will be given back as a 0 in the flagvector, NaN-values in the upward or downward-vector are considered as not existing.

Value

It returns a list with the following entries:

| | |
|----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| flagvector | A vector of length of measurement vector. For every change between two elements of the measurement vector the flagvector contains a -1, if its exceeding its dedicated downward vector element, a 1, if its exceeding its dedicated upward vector element, or a 0, when no exceeding has happened. |
| max_upward_vector | Give back the given max_upward_vector |
| max_downward_vector | Give back the given max_downward_vector |
| upward_vector_name | Give back the given upward_vector_name |
| downward_vector_name | Give back the given downward_vector_name |

Author(s)

Andre Duesterhus

References

Meek, D.W., Hatfield, J.L. (1994) Data quality checking for single station meteorological databases, *_Agricultural and Forest Meteorology_, *69* (1-2), 85-109.*

See Also

[qat_plot_roc_rule_dynamic_1d](#), [qat_call_roc_rule](#), [qat_analyse_roc_rule_static_1d](#)

Examples

```
vec <- rnorm(100)
min_vector<-seq(1,2,length.out=1000)
max_vector<-seq(1,2,length.out=1000)
result <- qat_analyse_roc_rule_dynamic_1d(vec, min_vector, max_vector, upward_vector_name="upward vector", downward_vector_name="downward vector")
```

`qat_analyse_roc_rule_static_1d`*Perform a static roc-rule-check*

Description

This check tests data on whether the change between two consecutive data points exceeds a static threshold.

Usage

```
qat_analyse_roc_rule_static_1d(measurement_vector, max_upward_value, max_downward_value)
```

Arguments`measurement_vector`

The measurement vector, which should be tested

`max_upward_value`

The upward threshold

`max_downward_value`

The downward threshold, which should be positive definite

Details

This check tests two consecutive elements, on whether the change of values between those two exceeds the upward or downward threshold. The result will be given back as a list, which contains the result of the test as a flagvector and its parameters. For every change between two elements of the measurement vector the flagvector contains a -1, if its exceeding the downward value, a 1, if its exceeding the upward value, or a 0, when no exceeding has happened.

Value

It returns a list with the following entries:

`flagvector`

A vector of length of measurement vector. For every change between two elements of the measurement vector the flagvector contains a -1, if its exceeding the downward value, a 1, if its exceeding the upward value, or a 0, when no exceeding has happened.

`max_upward_value`Give back the given `max_upward_value``max_downward_value`Give back the given `max_downward_value`**Author(s)**

Andre Duesterhus

References

Meek, D.W., Hatfield, J.L. (1994) Data quality checking for single station meteorological databases, *_Agricultural and Forest Meteorology*, *69* (1-2), 85-109.

See Also

[qat_plot_roc_rule_static_1d](#), [qat_call_roc_rule](#), [qat_analyse_roc_rule_dynamic_1d](#)

Examples

```
vec <- rnorm(100)
result <- qat_analyse_roc_rule_static_1d(vec, 2,2)
```

qat_analyse_set_addup_1d

Addup values of a vector

Description

This function adds up successive values of a given vector

Usage

```
qat_analyse_set_addup_1d(measurement_vector, blocksize)
```

Arguments

measurement_vector

The measurement vector, which should be tested

blocksize

Number of elements, which should be added up

Details

Starting with the first element the measurement vector will be splitted up into blocks of the size of the parameter block size. In a second step the elements of these blocks will be summed up. If the last block haven't the size of block size, this block will be ignored.

Value

Give back a list, which includes the vector with the results of the blocks.

Author(s)

Andre Duesterhus

See Also

[qat_analyse_set_mean_1d](#), [qat_analyse_set_nans_1d](#)

Examples

```
vec <- c(1,2,3,4,5,4,3,2,1)
result <- qat_analyse_set_addup_1d(vec, 3)
```

qat_analyse_set_mean_1d

Mean of values of a vector

Description

This function makes a mean of successive values of a given vector.

Usage

```
qat_analyse_set_mean_1d(measurement_vector, blocksize)
```

Arguments

| | |
|--------------------|------------------------------------------------|
| measurement_vector | The measurement vector, which should be tested |
| blocksize | Number of elements, which should be added up |

Details

Starting with the first element the measurement vector will be splitted up into blocks of the size of the parameter block size. In a second step a mean will be formed with the elements of these blocks. If the last block haven't the size of block size, this block will be ignored.

Value

Give back a list, which includes the vector with the results of the blocks.

Author(s)

Andre Duesterhus

Examples

```
vec <- c(1,2,3,4,5,4,3,2,1)
result <- qat_analyse_set_mean_1d(vec, 3)
```

qat_analyse_set_nans_1d

Set given values of a vector to NaN

Description

This function set a specified value of a vector to NaN.

Usage

```
qat_analyse_set_nans_1d(measurement_vector, nan_value)
```

Arguments

measurement_vector

The measurement vector, which should be worked on

nan_value

Value, which should be replaced by NaN

Details

In the given measurement vector, the value, which is specified by nan_value, will be replaced by NaN.

Value

Retrun a list, which includes the measurement vector with the replaced values.

Author(s)

Andre Duesterhus

See Also

[qat_analyse_set_addup_1d](#), [qat_analyse_set_mean_1d](#)

Examples

```
vec <- c(1,2,3,4,5,4,3,2,1)
result <- qat_analyse_set_nans_1d(vec, 4)
```

qat_analyse_set_nans_above_1d
Set values above threshold to NaN

Description

This function set a values of a vector above a given value to NaN.

Usage

```
qat_analyse_set_nans_above_1d(measurement_vector, nan_above)
```

Arguments

measurement_vector The measurement vector, which should be worked on
nan_above Value, above the values should be replaced by NaN

Details

In the given measurement vector, the values, which are above nan_above, will be replaced by NaN.

Value

Return a list, which includes the measurement vector with the replaced values.

Author(s)

Andre Duesterhus

See Also

[qat_analyse_set_addup_1d](#), [qat_analyse_set_mean_1d](#), [qat_analyse_set_nans_1d](#), [qat_analyse_set_nans_below_1d](#)

Examples

```
vec <- c(1,2,3,4,5,4,3,2,1)
result <- qat_analyse_set_nans_above_1d(vec, 4)
```

qat_analyse_set_nans_below_1d

Set values below threshold to NaN

Description

This function set a values of a vector below a given value to NaN.

Usage

```
qat_analyse_set_nans_below_1d(measurement_vector, nan_below)
```

Arguments

measurement_vector The measurement vector, which should be worked on
nan_below Value, below the values should be replaced by NaN

Details

In the given measurement vector, the values, which are below nan_below, will be replaced by NaN.

Value

Return a list, which includes the measurement vector with the replaced values.

Author(s)

Andre Duesterhus

See Also

[qat_analyse_set_addup_1d](#), [qat_analyse_set_mean_1d](#), [qat_analyse_set_nans_1d](#), [qat_analyse_set_nans_above_1d](#)

Examples

```
vec <- c(1,2,3,4,5,4,3,2,1)
result <- qat_analyse_set_nans_below_1d(vec, 4)
```

qat_analyse_slide_distribution_1d
Perform a slide distribution check

Description

The measurement vector will be scanned stepwise by a sliding window, and on every step some statistical parameters will be calculated.

Usage

```
qat_analyse_slide_distribution_1d(measurement_vector, blocksize)
```

Arguments

| | |
|--------------------|------------------------------------------------|
| measurement_vector | The measurement vector, which should be tested |
| blocksize | Length of the sliding window |

Details

The emasurement vector will be scanned stepwise by a sliding window, which got a length of the given parameter blocksize. At every step some statistical parameters will be calculated for the actual window. As a result a list will be given back, with these parameters, where every entry got a length of the length of the measurement vector minus the blocksize plus one.

Value

It returns a list with the following entries:

| | |
|--------------------|-----------------------------------------------|
| first_moment | First moment of the measurement vector |
| second_moment | Second moment of the measurement vector |
| third_moment | Third moment of the measurement vector |
| fourth_moment | Fourth moment of the measurement vector |
| standard_deviation | Standard deviation of the measurement vector |
| skewness | Skewness of the measurement vector |
| kurtosis | Kurtosis of the measurement vector |
| median | Median of the measurement vector |
| p5_quantile | 5 percent quantile of the measurement vector |
| p95_quantile | 95 percent quantile of the measurement vector |
| p25_quantile | 25 percent quantile of the measurement vector |
| p75_quantile | 75 percent quantile of the measurement vector |
| blocksize | Length of the used blocks |

Author(s)

Andre Duesterhus

See Also

[qat_plot_slide_distribution_1d](#)

Examples

```
vec <- rnorm(100)
result <- qat_analyse_slide_distribution_1d(vec, 10)
```

qat_analyse_trimmed_distribution_1d

Perform a trimmed distribution check

Description

The measurement vector will be trimmed at each side stepwise and at every step some statistical parameters will be calculated.

Usage

```
qat_analyse_trimmed_distribution_1d(measurement_vector)
```

Arguments

measurement_vector

The measurement vector, which should be tested

Details

The measurement vector will be trimmed at each side stepwise, with a step of 1 percent. At each step some statistical parameters will be calculated. As a result a list will be given back, with these parameters, where every entry got a length of 50.

Value

It returns a list with the following entries:

| | |
|--------------------|----------------------------------------------|
| first_moment | First moment of the measurement vector |
| second_moment | Second moment of the measurement vector |
| third_moment | Third moment of the measurement vector |
| fourth_moment | Fourth moment of the measurement vector |
| standard_deviation | Standard deviation of the measurement vector |
| skewness | Skewness of the measurement vector |

| | |
|--------------|-----------------------------------------------|
| kurtosis | Kurtosis of the measurement vector |
| median | Median of the measurement vector |
| p5_quantile | 5 percent quantile of the measurement vector |
| p95_quantile | 95 percent quantile of the measurement vector |
| p25_quantile | 25 percent quantile of the measurement vector |
| p75_quantile | 75 percent quantile of the measurement vector |

Author(s)

Andre Duesterhus

See Also

[qat_plot_trimmed_distribution_1d](#)

Examples

```
vec <- rnorm(1000)
result <- qat_analyse_trimmed_distribution_1d(vec)
```

qat_call_block_distribution

Perform a block distribution check

Description

This function calls the described tests, which are defined by the parameters in the workflowlist_part. The possible called functions are qat_analyse_block_distribution_1d.

Usage

```
qat_call_block_distribution(measurement_vector, workflowlist_part, element = -999, time = NULL, height
```

Arguments

| | |
|--------------------|-------------------------------------------------------------------------------|
| measurement_vector | The measurement vector, which should be tested |
| workflowlist_part | A list with the parameters of the check |
| element | Element-identifier for the result, which will be given back in the resultlist |
| time | A vector of time elements with the length of the measurement vector |
| height | A vector of height elements with the length of the measurement vector |
| lat | A vector of latitude elements with the length of the measurement vector |
| lon | A vector of longitude elements with the length of the measurement vector |
| vec1 | An additional vector, which is named as vec1 |

| | |
|-------------------|----------------------------------------------|
| vec2 | An additional vector, which is named as vec2 |
| vec3 | An additional vector, which is named as vec3 |
| vec4 | An additional vector, which is named as vec4 |
| resultlist | A list with results of tests |
| resultlistcounter | Number of elements of the resultlist |

Details

This function calls the described tests, which are defined by the parameters in the workflowlist_part. The possible called functions are qat_analyse_block_distribution_1d. As a result the resultlist will get additional entries, which are defined by the tests, which may called by this function.

Value

The given resultlist will be returned, with included results of the functions which may called in this function.

Author(s)

Andre Duesterhus

See Also

[qat_analyse_slide_distribution_1d](#)

Examples

```
vec <- rnorm(1000)
workflowlist_part <- list(blocksize=50)
result <- qat_call_block_distribution(vec, workflowlist_part)
```

qat_call_boot_distribution

Perform a bootstrapped distribution check

Description

This function calls the described tests, which are defined by the parameters in the workflowlist_part. The possible called functions are qat_analyse_boot_distribution_1d.

Usage

```
qat_call_boot_distribution(measurement_vector, workflowlist_part, element = -999, time = NULL, height
```

Arguments

| | |
|--------------------|-------------------------------------------------------------------------------|
| measurement_vector | The measurement vector, which should be tested |
| workflowlist_part | A list with the parameters of the check |
| element | Element-identifier for the result, which will be given back in the resultlist |
| time | A vector of time elements with the length of the measurement vector |
| height | A vector of height elements with the length of the measurement vector |
| lat | A vector of latitude elements with the length of the measurement vector |
| lon | A vector of longitude elements with the length of the measurement vector |
| vec1 | An additional vector, which is named as vec1 |
| vec2 | An additional vector, which is named as vec2 |
| vec3 | An additional vector, which is named as vec3 |
| vec4 | An additional vector, which is named as vec4 |
| resultlist | A list with results of tests |
| resultlistcounter | Number of elements of the resultlist |

Details

This function calls the described tests, which are defined by the parameters in the workflowlist_part. The possible called functions are qat_analyse_boot_distribution_1d. As a result the resultlist will get additional entries, which are defined by the tests, which may called by this function.

Value

The given resultlist will be returned, with included results of the functions which may called in this function.

Author(s)

Andre Duesterhus

See Also

[qat_analyse_boot_distribution_1d](#)

Examples

```
vec <- rnorm(1000)
workflowlist_part <- list(bootruns=1000)
result <- qat_call_boot_distribution(vec, workflowlist_part)
```

qat_call_distribution *Perform a distribution check*

Description

This function calls the described tests, which are defined by the parameters in the workflowlist_part. The possible called functions are qat_analyse_distribution_1d.

Usage

```
qat_call_distribution(measurement_vector, workflowlist_part, element = -999, time = NULL, height = NULL)
```

Arguments

| | |
|--------------------|-------------------------------------------------------------------------------|
| measurement_vector | The measurement vector, which should be tested |
| workflowlist_part | A list with the parameters of the check |
| element | Element-identifier for the result, which will be given back in the resultlist |
| time | A vector of time elements with the length of the measurement vector |
| height | A vector of height elements with the length of the measurement vector |
| lat | A vector of latitude elements with the length of the measurement vector |
| lon | A vector of longitude elements with the length of the measurement vector |
| vec1 | An additional vector, which is named as vec1 |
| vec2 | An additional vector, which is named as vec2 |
| vec3 | An additional vector, which is named as vec3 |
| vec4 | An additional vector, which is named as vec4 |
| resultlist | A list with results of tests |
| resultlistcounter | Number of elements of the resultlist |

Details

This function calls the described tests, which are defined by the parameters in the workflowlist_part. The possible called functions are qat_analyse_distribution_1d. As a result the resultlist will get additional entries, which are defined by the tests, which may be called by this function.

Value

The given resultlist will be returned, with included results of the functions which may be called in this function.

Author(s)

Andre Duesterhus

See Also

[qat_analyse_distribution_1d](#)

Examples

```
vec <- rnorm(1000)
workflowlist_part <- list(numofbars=15)
result <- qat_call_distribution(vec, workflowlist_part)
```

qat_call_lim_rule *Perform a LIM Rule Check*

Description

This function calls the described tests, which are defined by the parameters in the workflowlist_part. The possible called functions are qat_analyse_lim_rule_dynamic_1d, qat_analyse_lim_rule_static_1d and qat_analyse_lim_rule_sigma_1d.

Usage

```
qat_call_lim_rule(measurement_vector, workflowlist_part, element = -999, time = NULL, height = NULL, lat = NULL, lon = NULL, vec1 = NULL, vec2 = NULL, vec3 = NULL, vec4 = NULL, resultlist = NULL, resultlistcounter = 0)
```

Arguments

| | |
|--------------------|-------------------------------------------------------------------------------|
| measurement_vector | The measurement vector, which should be tested |
| workflowlist_part | A list with the parameters of the check |
| element | Element-identifier for the result, which will be given back in the resultlist |
| time | A vector of time elements with the length of the measurement vector |
| height | A vector of height elements with the length of the measurement vector |
| lat | A vector of latitude elements with the length of the measurement vector |
| lon | A vector of longitude elements with the length of the measurement vector |
| vec1 | An additional vector, which is named as vec1 |
| vec2 | An additional vector, which is named as vec2 |
| vec3 | An additional vector, which is named as vec3 |
| vec4 | An additional vector, which is named as vec4 |
| resultlist | A list with results of tests |
| resultlistcounter | Number of elements of the resultlist |

Details

This function calls the described tests, which are defined by the parameters in the workflowlist_part. The possible called functions are qat_analyse_lim_rule_dynamic_1d, qat_analyse_lim_rule_static_1d and qat_analyse_lim_rule_sigma_1d. As a result the resultlist will get additional entries, which are defined by the tests, which may called by this function.

Value

The given resultlist will be returned, with included results of the functions which may called in this function.

Author(s)

Andre Duesterhus

See Also

[qat_analyse_lim_rule_static_1d](#), [qat_analyse_lim_rule_dynamic_1d](#), [qat_analyse_lim_rule_sigma_1d](#), [qat_plot_lim_rule_dynamic_1d](#), [qat_plot_lim_rule_static_1d](#), [qat_plot_lim_rule_sigma_1d](#)

Examples

```
vec <- rnorm(1000)
min_vector<-seq(-1,-2,length.out=1000)
max_vector<-seq(1,2,length.out=1000)
workflowlist_part <- list(minimum_value=-2, maximum_value=2,minimum_vector="vec1",maximum_vector="vec2",minimum
result <- qat_call_lim_rule(vec, workflowlist_part, vec1=min_vector, vec2=max_vector)
```

qat_call_noc_rule *Perform a NOC Rule Check*

Description

This function calls the described tests, which are defined by the parameters in the workflowlist_part. The possible called functions are qat_analyse_noc_rule_1d.

Usage

```
qat_call_noc_rule(measurement_vector, workflowlist_part, element = -999, time = NULL, height = NULL, la
```

Arguments

measurement_vector

The measurement vector, which should be tested

workflowlist_part

A list with the parameters of the check

element

Element-identifier for the result, which will be given back in the resultlist

| | |
|-------------------|--------------------------------------------------------------------------|
| time | A vector of time elements with the length of the measurement vector |
| height | A vector of height elements with the length of the measurement vector |
| lat | A vector of latitude elements with the length of the measurement vector |
| lon | A vector of longitude elements with the length of the measurement vector |
| vec1 | An additional vector, which is named as vec1 |
| vec2 | An additional vector, which is named as vec2 |
| vec3 | An additional vector, which is named as vec3 |
| vec4 | An additional vector, which is named as vec4 |
| resultlist | A list with results of tests |
| resultlistcounter | Number of elements of the resultlist |

Details

This function calls the described tests, which are defined by the parameters in the workflowlist_part. The possible called functions are qat_analyse_noc_rule_1d. As a result the resultlist will get additional entries, which are defined by the tests, which may be called by this function.

Value

The given resultlist will be returned, with included results of the functions which may be called in this function.

Author(s)

Andre Duesterhus

See Also

[qat_plot_noc_rule_1d](#)

Examples

```
vec <- c(1,2,3,4,4,4,5,5,4,3,NaN,3,2,1)
workflowlist_part <- list(max_return_elements=1)
result <- qat_call_noc_rule(vec, workflowlist_part)
```

qat_call_plot_block_distribution

Plot a result of a block distribution check

Description

A result of qat\analyse\block\distribution\1d will be plotted.

Usage

```
qat_call_plot_block_distribution(resultlist_part, measurement_vector = NULL, time=NULL, height= NULL,
```

Arguments

| | |
|--------------------|--------------------------------------------------------------------------------------------------------|
| resultlist_part | A list with the result, which is directly or indirectly produced by qat\analyse\block\distribution\1d. |
| measurement_vector | The measurement vector, which was tested |
| time | A vector of time elements with the length of the measurement vector |
| height | A vector of height elements with the length of the measurement vector |
| lat | A vector of latitude elements with the length of the measurement vector |
| lon | A vector of longitude elements with the length of the measurement vector |
| measurement_name | Name of the data, which will be used as an indicator on the plot |
| directoryname | Definition of the directory, where the plot should be stored |
| basename | Basic name of the resulting file |
| plotstyle | A list with a qat color scheme |

Details

A plot will be produced, which base on the resulting list of qat\analyse\block\distribution\1d. The measurement_name will be used as a title of the plot and the plotstyle list define the colors of the plot. When no plotstyle is defined the standard-colorscheme will be used. The resulting plot will be stored in the folder, which is defined by directory. As a filename the basename with additional information will be used (number of test and a label, which indicate which test was performed).

Value

No return value.

Author(s)

Andre Duesterhus

See Also

[qat_plot_block_distribution_1d](#)

Examples

```
vec <- rnorm(1000)
workflowlist_part <- list(blocksize=50)
resultlist <- qat_call_block_distribution(vec, workflowlist_part, element=1)
# this example produce the files exampleplot_1_blockdist_1.png, exampleplot_1_blockdist_2.png and exampleplot_1_b
qat_call_plot_block_distribution(resultlist[[2]], measurement_vector=vec, measurement_name="Result of Check", ba
```

```
qat_call_plot_boot_distribution
```

Plot a result of a bootstrapped distribution check

Description

A result of `qat\analyse\boot\distribution\1d` will be plotted.

Usage

```
qat_call_plot_boot_distribution(resultlist_part, measurement_vector = NULL, time = NULL, height = NULL
```

Arguments

| | |
|---------------------------------|------------------------------------------------------------------------------------------------------------------------|
| <code>resultlist_part</code> | A list with the result, which is directly or indirectly produced by <code>qat\analyse\trimmed\distribution\1d</code> . |
| <code>measurement_vector</code> | The measurement vector, which was tested |
| <code>time</code> | A vector of time elements with the length of the measurement vector |
| <code>height</code> | A vector of height elements with the length of the measurement vector |
| <code>lat</code> | A vector of latitude elements with the length of the measurement vector |
| <code>lon</code> | A vector of longitude elements with the length of the measurement vector |
| <code>measurement_name</code> | Name of the data, which will be used as an indicator on the plot |
| <code>directoryname</code> | Definition of the directory, where the plot should be stored |
| <code>basename</code> | Basic name of the resulting file |
| <code>plotstyle</code> | A list with a qat color scheme |

Details

A plot will be produced, which base on the resulting list of `qat\analyse\boot\distribution\1d`. The `measurement_name` will be used as a title of the plot and the `plotstyle` list define the colors of the plot. When no `plotstyle` is defined the standard-colorscheme will be used. The resulting plot will be stored in the folder, which is defined by `directory`. As a filename the `basename` with additional information will be used (number of test and a label, which indicate which test was performed).

Value

No return value.

Author(s)

Andre Duesterhus

See Also

[qat_plot_boot_distribution_1d](#)

Examples

```
vec <- rnorm(1000)
workflowlist_part <- list(bootruns=1000)
resultlist <- qat_call_boot_distribution(vec, workflowlist_part, element=1)
# this example produce the file exampleplot_1_bootdist.png in the current directory
qat_call_plot_boot_distribution(resultlist[[2]], measurement_vector=vec, measurement_name="Result of Check", bas
```

qat_call_plot_distribution

Plot a result of a distribution check

Description

A result of `qat\analyse\distribution\1d` will be plotted.

Usage

```
qat_call_plot_distribution(resultlist_part, measurement_vector = NULL, time = NULL, height = NULL, lat
```

Arguments

| | |
|---------------------------------|----------------------------------------------------------------------------------------------------------------|
| <code>resultlist_part</code> | A list with the result, which is directly or indirectly produced by <code>qat\analyse\distribution\1d</code> . |
| <code>measurement_vector</code> | The measurement vector, which was tested |
| <code>time</code> | A vector of time elements with the length of the measurement vector |
| <code>height</code> | A vector of height elements with the length of the measurement vector |
| <code>lat</code> | A vector of latitude elements with the length of the measurement vector |
| <code>lon</code> | A vector of longitude elements with the length of the measurement vector |
| <code>measurement_name</code> | Name of the data, which will be used as an indicator on the plot |
| <code>directoryname</code> | Definition of the directory, where the plot should be stored |
| <code>basename</code> | Basic name of the resulting file |
| <code>plotstyle</code> | A list with a qat color scheme |

Details

A plot will be produced, which base on the resulting list of `qat\analyse\distribution\1d`. The `measurement_name` will be used as a title of the plot and the `plotstyle` list define the colors of the plot. When no `plotstyle` is defined the standard-colorscheme will be used. The resulting plot will be stored in the folder, which is defined by `directory`. As a filename the `basename` with additional information will be used (number of test and a label, which indicate which test was performed).

Value

No return value.

Author(s)

Andre Duesterhus

See Also

[qat_plot_distribution_1d](#)

Examples

```
vec <- rnorm(1000)
workflowlist_part <- list(numofbars=15)
resultlist <- qat_call_distribution(vec, workflowlist_part, element=1)
# this example produce a file exampleplot_1_dist.png in the current directory
qat_call_plot_distribution(resultlist[[2]], measurement_vector=vec, measurement_name="Result of Check", basename="exampleplot_1_dist.png")
```

qat_call_plot_lim_rule

Plot a result of a LIM rule check

Description

A result of `qat\analyse_lim_rule_static_1d`, `qat\analyse_lim_rule_sigma_1d` or `qat\analyse_lim_rule_dynamics_1d` will be plotted.

Usage

```
qat_call_plot_lim_rule(resultlist_part, measurement_vector = NULL, time = NULL, height = NULL, lat = NULL)
```

Arguments

`resultlist_part`

A list with the result, which is directly or indirectly produced by `qat\analyse_lim_rule_static_1d`, `qat\analyse_lim_rule_sigma_1d` or `qat\analyse_lim_rule_dynamics_1d`.

`measurement_vector`

The measurement vector, which was tested

| | |
|------------------|--------------------------------------------------------------------------|
| time | A vector of time elements with the length of the measurement vector |
| height | A vector of height elements with the length of the measurement vector |
| lat | A vector of latitude elements with the length of the measurement vector |
| lon | A vector of longitude elements with the length of the measurement vector |
| measurement_name | Name of the data, which will be used as an indicator in the plot |
| directoryname | Definition of the directory, where the plot should be stored |
| basename | Basic name of the resulting file |
| plotstyle | A list with a qat color scheme |

Details

A plot will be produced, which base on the resulting list of `qat\analyse\lim\rule\static\1d`, `qat\analyse\lim\rule\sigma\1d` or `qat\analyse\lim\rule\dynamics\1d`. The `measurement_name` will be used as a title of the plot and the `plotstyle` list define the colors of the plot. When no `plotstyle` is defined the `standard-colorscheme` will be used. The resulting plot will be stored in the folder, which is defined by `directory`. As a filename the `basename` with additional information will be used (number of test and a label, which indicate which test was performed).

Value

No return value.

Author(s)

Andre Duesterhus

See Also

[qat_plot_lim_rule_dynamic_1d](#), [qat_plot_lim_rule_static_1d](#), [qat_plot_lim_rule_sigma_1d](#)

Examples

```
vec <- rnorm(1000)
min_vector<-seq(-1,-2,length.out=1000)
max_vector<-seq(1,2,length.out=1000)
workflowlist_part <- list(minimum_value=-2, maximum_value=2,minimum_vector="vec1",maximum_vector="vec2",minimum
resultlist <- qat_call_lim_rule(vec, workflowlist_part, element=1, vec1=min_vector, vec2=max_vector)
# this example produce the files exampleplot_1_lim_sigma.png, exampleplot_1_lim_static.png and exampleplot_1_lim_
for (ii in 2:4) {
qat_call_plot_lim_rule(resultlist[[ii]], measurement_vector=vec, measurement_name="Result of Check", basename="e
}
```

qat_call_plot_noc_rule

Plot a result of a NOC rule check

Description

A result of qat\analyse\noc\rule\1d will be plotted.

Usage

```
qat_call_plot_noc_rule(resultlist_part, measurement_vector = NULL, time = NULL, height = NULL, lat = NU
```

Arguments

| | |
|--------------------|----------------------------------------------------------------------------------------------|
| resultlist_part | A list with the result, which is directly or indirectly produced by qat\analyse\noc\rule\1d. |
| measurement_vector | The measurement vector, which was tested |
| time | A vector of time elements with the length of the measurement vector |
| height | A vector of height elements with the length of the measurement vector |
| lat | A vector of latitude elements with the length of the measurement vector |
| lon | A vector of longitude elements with the length of the measurement vector |
| measurement_name | Name of the data, which will be used as an indicator on the plot |
| directoryname | Definition of the directory, where the plot should be stored |
| basename | Basic name of the resulting file |
| plotstyle | A list with a qat color scheme |

Details

A plot will be produced, which base on the resulting list of qat\analyse\noc\rule\1d. The measurement_name will be used as a title of the plot and the plotstyle list define the colors of the plot. When no plotstyle is defined the standard-colorscheme will be used. The resulting plot will be stored in the folder, which is defined by directory. As a filename the basename with additional information will be used (number of test and a label, which indicate which test was performed).

Value

No return value.

Author(s)

Andre Duesterhus

See Also

[qat_plot_noc_rule_1d](#)

Examples

```
vec <- c(1,2,3,4,4,4,5,5,4,3,NaN,3,2,1)
workflowlist_part <- list(max_return_elements=1)
resultlist <- qat_call_noc_rule(vec, workflowlist_part,element=1)
# this example produce a file exampleplot_1_noc.png in the current directory
qat_call_plot_noc_rule(resultlist[[2]], measurement_vector=vec, measurement_name="Result of Check", basename="ex
```

qat_call_plot_roc_rule

Plot a result of a ROC rule check

Description

A result of `qat_analyse_roc_rule_static_1d` or `qat_analyse_roc_rule_dynamics_1d` will be plotted.

Usage

```
qat_call_plot_roc_rule(resultlist_part, measurement_vector = NULL, time = NULL, height = NULL, lat = NU
```

Arguments

| | |
|---------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>resultlist_part</code> | A list with the result, which is directly or indirectly produced by <code>qat_analyse_roc_rule_static_1d</code> or <code>qat_analyse_roc_rule_dynamics_1d</code> . |
| <code>measurement_vector</code> | The measurement vector, which was tested |
| <code>time</code> | A vector of time elements with the length of the measurement vector |
| <code>height</code> | A vector of height elements with the length of the measurement vector |
| <code>lat</code> | A vector of latitude elements with the length of the measurement vector |
| <code>lon</code> | A vector of longitude elements with the length of the measurement vector |
| <code>measurement_name</code> | Name of the data, which will be used as an indicator in the plot |
| <code>directoryname</code> | Definition of the directory, where the plot should be stored |
| <code>basename</code> | Basic name of the resulting file |
| <code>plotstyle</code> | A list with a qat color scheme |

Details

A plot will be produced, which base on the resulting list of `qat\analyse\roc\rule\static\1d` or `qat\analyse\roc\rule\dynamics\1d`. The `measurement_name` will be used as a title of the plot and the `plotstyle` list define the colors of the plot. When no `plotstyle` is defined the `standard-colorscheme` will be used. The resulting plot will be stored in the folder, which is defined by `directory`. As a filename the `basename` with additional information will be used (number of test and a label, which indicate which test was performed).

Value

No return value.

Author(s)

Andre Duesterhus

See Also

[qat_plot_roc_rule_dynamic_1d](#), [qat_plot_roc_rule_static_1d](#)

Examples

```
vec <- rnorm(100)
downward_vector<-seq(1,2,length.out=100)
upward_vector<-seq(1,2,length.out=100)
workflowlist_part <- list(downward_value=2, upward_value=2,downward_vector="vec1",upward_vector="vec2",downward
resultlist <- qat_call_roc_rule(vec, workflowlist_part, element=1, vec1=downward_vector, vec2=upward_vector)
# this example produce the files exampleplot_1_roc_static.png and exampleplot_1_roc_dynamic.png in the current dir
for (ii in 2:3) {
qat_call_plot_roc_rule(resultlist[[ii]], measurement_vector=vec, measurement_name="Result of Check", basename="e
}
```

`qat_call_plot_slide_distribution`

Plot a result of a slide distribution check

Description

A result of `qat\analyse\slide\distribution\1d` will be plotted.

Usage

```
qat_call_plot_slide_distribution(resultlist_part, measurement_vector = NULL, time = NULL, height = NUL
```

Arguments

| | |
|--------------------|----------------------------------------------------------------------------------------------------------------------|
| resultlist_part | A list with the result, which is directly or indirectly produced by <code>qat\analyse\slide\distribution\1d</code> . |
| measurement_vector | The measurement vector, which was tested |
| time | A vector of time elements with the length of the measurement vector |
| height | A vector of height elements with the length of the measurement vector |
| lat | A vector of latitude elements with the length of the measurement vector |
| lon | A vector of longitude elements with the length of the measurement vector |
| measurement_name | Name of the data, which will be used as an indicator on the plot |
| directoryname | Definition of the directory, where the plot should be stored |
| basename | Basic name of the resulting file |
| plotstyle | A list with a qat color scheme |

Details

A plot will be produced, which base on the resulting list of `qat\analyse\slide\distribution\1d`. The `measurement_name` will be used as a title of the plot and the `plotstyle` list define the colors of the plot. When no `plotstyle` is defined the standard-colorscheme will be used. The resulting plot will be stored in the folder, which is defined by `directory`. As a filename the `basename` with additional information will be used (number of test and a label, which indicate which test was performed).

Value

No return value.

Author(s)

Andre Duesterhus

See Also

[qat_plot_slide_distribution_1d](#)

Examples

```
vec <- rnorm(100)
workflowlist_part <- list(blocksize=10)
resultlist <- qat_call_slide_distribution(vec, workflowlist_part, element=1)
# this example produce the files exampleplot_1_slidedist_1.png, exampleplot_1_slidedist_2.png and exampleplot_1_s
qat_call_plot_slide_distribution(resultlist[[2]], measurement_vector=vec, measurement_name="Result of Check", ba
```

qat_call_plot_trimmed_distribution

Plot a result of a trimmed distribution check

Description

A result of `qat\analyse_trimmed_distribution_1d` will be plotted.

Usage

```
qat_call_plot_trimmed_distribution(resultlist_part, measurement_vector = NULL, time = NULL, height = N
```

Arguments

| | |
|---------------------------------|---------------------------------------------------------------------------------------------------------------------------|
| <code>resultlist_part</code> | A list with the result, which is directly or indirectly produced by <code>qat\analyse_trimmed_distribution_1d</code> . |
| <code>measurement_vector</code> | The measurement vector, which was tested |
| <code>time</code> | A vector of time elements with the length of the measurement vector |
| <code>height</code> | A vector of height elements with the length of the measurement vector |
| <code>lat</code> | A vector of latitude elements with the length of the measurement vector |
| <code>lon</code> | A vector of longitude elements with the length of the measurement vector |
| <code>measurement_name</code> | Name of the data, which will be used as an indicator on the plot |
| <code>directoryname</code> | Definition of the directory, where the plot should be stored |
| <code>basename</code> | Basic name of the resulting file |
| <code>plotstyle</code> | A list with a qat color scheme |

Details

A plot will be produced, which base on the resulting list of `qat\analyse_trimmed_distribution_1d`. The `measurement_name` will be used as a title of the plot and the `plotstyle` list define the colors of the plot. When no `plotstyle` is defined the standard-colorscheme will be used. The resulting plot will be stored in the folder, which is defined by `directory`. As a filename the `basename` with additional information will be used (number of test and a label, which indicate which test was performed).

Value

No return value.

Author(s)

Andre Duesterhus

See Also

[qat_plot_trimmed_distribution_1d](#)

Examples

```
vec <- rnorm(1000)
workflowlist_part <- list()
resultlist <- qat_call_trimmed_distribution(vec, workflowlist_part, element=1)
# this example produce a file exampleplot_1_trimmeddist.png in the current directory
qat_call_plot_trimmed_distribution(resultlist[[2]], measurement_vector=vec, measurement_name="Result of Check",
```

qat_call_roc_rule *Perform a ROC Rule Check*

Description

This function calls the described tests, which are defined by the parameters in the workflowlist_part. The possible called functions are qat_analyse_roc_rule_dynamic_1d and qat_analyse_roc_rule_static_1d.

Usage

```
qat_call_roc_rule(measurement_vector, workflowlist_part, element = -999, time = NULL, height = NULL, la
```

Arguments

| | |
|--------------------|-------------------------------------------------------------------------------|
| measurement_vector | The measurement vector, which should be tested |
| workflowlist_part | A list with the parameters of the check |
| element | Element-identifier for the result, which will be given back in the resultlist |
| time | A vector of time elements with the length of the measurement vector |
| height | A vector of height elements with the length of the measurement vector |
| lat | A vector of latitude elements with the length of the measurement vector |
| lon | A vector of longitude elements with the length of the measurement vector |
| vec1 | An additional vector, which is named as vec1 |
| vec2 | An additional vector, which is named as vec2 |
| vec3 | An additional vector, which is named as vec3 |
| vec4 | An additional vector, which is named as vec4 |
| resultlist | A list with results of tests |
| resultlistcounter | Number of elements of the resultlist |

Details

This function calls the described tests, which are defined by the parameters in the workflowlist_part. The possible called functions are qat_analyse_roc_rule_dynamic_1d and qat_analyse_roc_rule_static_1d. As a result the resultlist will get additional entries, which are defined by the tests, which may called by this function.

Value

The given resultlist will be returned, with included results of the functions which may called in this function.

Author(s)

Andre Duesterhus

See Also

[qat_analyse_roc_rule_static_1d](#), [qat_analyse_roc_rule_dynamic_1d](#), [qat_plot_roc_rule_dynamic_1d](#), [qat_plot_roc_rule_static_1d](#)

Examples

```
vec <- rnorm(100)
downward_vector<-seq(1,2,length.out=1000)
upward_vector<-seq(1,2,length.out=1000)
workflowlist_part <- list(downward_value=2, upward_value=2,downward_vector="vec1",upward_vector="vec2",downward
result <- qat_call_roc_rule(vec, workflowlist_part,vec1=downward_vector,vec2=upward_vector)
```

qat_call_save_block_distribution

Produce a savelist-entry for a Block Distribution Test

Description

This function calls qat_save_block_distribution_1d. As a result a part of a savelist is constructed, which can be used to construct a netCDF file.

Usage

```
qat_call_save_block_distribution(resultlist_part, element = -999, time = NULL, height = NULL, lat = NUL
```

Arguments

| | |
|-----------------|-----------------------------------------------------------------------------|
| resultlist_part | A list with the results of the check |
| element | Element-identifier for the result, which will be given back in the savelist |
| time | A vector of time elements with the length of the measurement vector |

| | |
|------------------------------|--------------------------------------------------------------------------|
| <code>height</code> | A vector of height elements with the length of the measurement vector |
| <code>lat</code> | A vector of latitude elements with the length of the measurement vector |
| <code>lon</code> | A vector of longitude elements with the length of the measurement vector |
| <code>vec1</code> | An additional vector, which is named as <code>vec1</code> |
| <code>vec2</code> | An additional vector, which is named as <code>vec2</code> |
| <code>vec3</code> | An additional vector, which is named as <code>vec3</code> |
| <code>vec4</code> | An additional vector, which is named as <code>vec4</code> |
| <code>baseunit</code> | The unit of the original measurement vector |
| <code>savelist</code> | A list with save elements |
| <code>savelistcounter</code> | Numbers of elements of the <code>savelist</code> |

Details

This function calls the described saving-function, which transform the `resultlist` elements to a `savelist` element. The possible called function is `qat_save_block_distribution_1d`. As a result the given `savelist` will get an additional entry.

Value

The given `savelist` will be returned, with included results of the functions which may be called in this function.

Author(s)

Andre Duesterhus

See Also

[qat_save_block_distribution_1d](#), [qat_run_workflow_save](#)

Examples

```
vec <- rnorm(1000)
workflowlist_part <- list(blocksize=50)
resultlist <- qat_call_block_distribution(vec, workflowlist_part, element=1)
savelist <- qat_call_save_block_distribution(resultlist[[2]])
```

qat_call_save_boot_distribution

Produce a savelist-entry for a Boot Distribution Test

Description

This function calls `qat_save_boot_distribution_1d`. As a result a part of a savelist is constructed, which can be used to construct a netCDF file.

Usage

```
qat_call_save_boot_distribution(resultlist_part, element = -999, time = NULL, height = NULL, lat = NULL,
```

Arguments

| | |
|------------------------------|-----------------------------------------------------------------------------|
| <code>resultlist_part</code> | A list with the results of the check |
| <code>element</code> | Element-identifier for the result, which will be given back in the savelist |
| <code>time</code> | A vector of time elements with the length of the measurement vector |
| <code>height</code> | A vector of height elements with the length of the measurement vector |
| <code>lat</code> | A vector of latitude elements with the length of the measurement vector |
| <code>lon</code> | A vector of longitude elements with the length of the measurement vector |
| <code>vec1</code> | An additional vector, which is named as <code>vec1</code> |
| <code>vec2</code> | An additional vector, which is named as <code>vec2</code> |
| <code>vec3</code> | An additional vector, which is named as <code>vec3</code> |
| <code>vec4</code> | An additional vector, which is named as <code>vec4</code> |
| <code>baseunit</code> | The unit of the original measurement vector |
| <code>savelist</code> | A list with save elements |
| <code>savelistcounter</code> | Numbers of elements of the savelist |

Details

This function calls the described saving-function, which transform the resultlist elements to a savelist element. The possible called function is `qat_save_boot_distribution_1d`. As a result the given savelist will get an additional entry.

Value

The given savelist will be returned, with included results of the functions which may be called in this function.

Author(s)

Andre Duesterhus

See Also

[qat_save_boot_distribution_1d](#), [qat_run_workflow_save](#)

Examples

```
vec <- rnorm(1000)
workflowlist_part <- list(bootruns=1000)
resultlist <- qat_call_boot_distribution(vec, workflowlist_part, element=1)
savelist <- qat_call_save_boot_distribution(resultlist[[2]])
```

qat_call_save_distribution

Produce a savelist-entry for a Distribution Test

Description

This function calls `qat_save_distribution_1d`. As a result a part of a savelist is constructed, which can be used to construct a netCDF file.

Usage

```
qat_call_save_distribution(resultlist_part, element = -999, time = NULL, height = NULL, lat = NULL, lon
```

Arguments

| | |
|-----------------|-----------------------------------------------------------------------------|
| resultlist_part | A list with the results of the check |
| element | Element-identifier for the result, which will be given back in the savelist |
| time | A vector of time elements with the length of the measurement vector |
| height | A vector of height elements with the length of the measurement vector |
| lat | A vector of latitude elements with the length of the measurement vector |
| lon | A vector of longitude elements with the length of the measurement vector |
| vec1 | An additional vector, which is named as vec1 |
| vec2 | An additional vector, which is named as vec2 |
| vec3 | An additional vector, which is named as vec3 |
| vec4 | An additional vector, which is named as vec4 |
| baseunit | The unit of the original measurement vector |
| savelist | A list with save elements |
| savelistcounter | Numbers of elements of the savelist |

Details

This function calls the described saving-function, which transform the resultlist elements to a sav-inglist element. The possible called function is qat_save_distribution_1d. As a result the given savelist will get an additional entry.

Value

The given savelist will be returned, with included results of the functions which may be called in this function.

Author(s)

Andre Duesterhus

See Also

[qat_save_distribution_1d](#), [qat_run_workflow_save](#)

Examples

```
vec <- rnorm(1000)
workflowlist_part <- list(numofbars=15)
resultlist <- qat_call_distribution(vec, workflowlist_part, element=1)
qat_call_plot_distribution(resultlist[[2]], measurement_vector=vec, measurement_name="Result of Check", basename)
savelist <- qat_call_save_distribution(resultlist[[2]])
```

qat_call_save_lim_rule

Produce a savelist-entry for a LIM-RULE Test

Description

This function calls qat_save_lim_rule_static_1d, qat_save_lim_rule_sigma_1d or qat_save_lim_rule_dynamic_1d. As a result a part of a savelist is constructed, which can be used to construct a netCDF file.

Usage

```
qat_call_save_lim_rule(resultlist_part, element = -999, time = NULL, height = NULL, lat = NULL, lon = NU
```

Arguments

| | |
|-----------------|-----------------------------------------------------------------------------|
| resultlist_part | A list with the results of the check |
| element | Element-identifier for the result, which will be given back in the savelist |
| time | A vector of time elements with the length of the measurement vector |
| height | A vector of height elements with the length of the measurement vector |
| lat | A vector of latitude elements with the length of the measurement vector |

| | |
|-----------------|--------------------------------------------------------------------------|
| lon | A vector of longitude elements with the length of the measurement vector |
| vec1 | An additional vector, which is named as vec1 |
| vec2 | An additional vector, which is named as vec2 |
| vec3 | An additional vector, which is named as vec3 |
| vec4 | An additional vector, which is named as vec4 |
| baseunit | The unit of the original measurement vector |
| savelist | A list with save elements |
| savelistcounter | Numbers of elements of the savelist |

Details

This function calls the described saving-function, which transform the resultlist elements to a savinglist element. The possible called functions are `qat_save_lim_rule_static_1d`, `qat_save_lim_rule_sigma_1d` or `qat_save_lim_rule_dynamic_1d`. As a result the given savelist will get an additional entry.

Value

The given savelist will be returned, with included results of the functions which may be called in this function.

Author(s)

Andre Duesterhus

See Also

[qat_save_lim_rule_static_1d](#), [qat_save_lim_rule_sigma_1d](#), [qat_save_lim_rule_dynamic_1d](#), [qat_run_workflow_save](#)

Examples

```
vec <- rnorm(1000)
min_vector<-seq(-1,-2,length.out=1000)
max_vector<-seq(1,2,length.out=1000)
workflowlist_part <- list(minimum_value=-2, maximum_value=2,minimum_vector="vec1",maximum_vector="vec2",minimum
resultlist <- qat_call_lim_rule(vec, workflowlist_part, element=1, vec1=min_vector, vec2=max_vector)
savelist <- list()
savelistcounter <- 1
for (ii in 2:4) {
savelist <- qat_call_save_lim_rule(resultlist[[ii]], savelist=savelist, savelistcounter=savelistcounter)
if (length(which(names(savelist)=="element"))==0) {
savelistcounter<-length(savelist)
} else {
savelistcounter<-1
}
}
```

qat_call_save_noc_rule

Produce a savelist-entry for a NOC RULE Test

Description

This function calls `qat_save_noc_rule_1d`. As a result a part of a savelist is constructed, which can be used to construct a netCDF file.

Usage

```
qat_call_save_noc_rule(resultlist_part, element = -999, time = NULL, height = NULL, lat = NULL, lon = NU
```

Arguments

| | |
|------------------------------|-----------------------------------------------------------------------------|
| <code>resultlist_part</code> | A list with the results of the check |
| <code>element</code> | Element-identifier for the result, which will be given back in the savelist |
| <code>time</code> | A vector of time elements with the length of the measurement vector |
| <code>height</code> | A vector of height elements with the length of the measurement vector |
| <code>lat</code> | A vector of latitude elements with the length of the measurement vector |
| <code>lon</code> | A vector of longitude elements with the length of the measurement vector |
| <code>vec1</code> | An additional vector, which is named as <code>vec1</code> |
| <code>vec2</code> | An additional vector, which is named as <code>vec2</code> |
| <code>vec3</code> | An additional vector, which is named as <code>vec3</code> |
| <code>vec4</code> | An additional vector, which is named as <code>vec4</code> |
| <code>baseunit</code> | The unit of the original measurement vector |
| <code>savelist</code> | A list with save elements |
| <code>savelistcounter</code> | Numbers of elements of the savelist |

Details

This function calls the described saving-function, which transform the resultlist elements to a sav-inglist element. The possible called function is `qat_save_noc_rule_1d`. As a result the given savelist will get an additional entry.

Value

The given savelist will be returned, with included results of the functions which may be called in this function.

Author(s)

Andre Duesterhus

See Also

[qat_save_noc_rule_1d](#), [qat_run_workflow_save](#)

Examples

```
vec <- c(1,2,3,4,4,4,5,5,4,3,NaN,3,2,1)
workflowlist_part <- list(max_return_elements=1)
resultlist <- qat_call_noc_rule(vec, workflowlist_part,element=1)
savelist <- qat_call_save_noc_rule(resultlist[[2]])
```

qat_call_save_roc_rule

Produce a savelist-entry for a ROC-Rule Test

Description

This function calls `qat_save_roc_rule_static_1d` or `qat_save_roc_rule_dynamic_1d`. As a result a part of a savelist is constructed, which can be used to construct a netCDF file.

Usage

```
qat_call_save_roc_rule(resultlist_part, element = -999, time = NULL, height = NULL, lat = NULL, lon = NU
```

Arguments

| | |
|-----------------|-----------------------------------------------------------------------------|
| resultlist_part | A list with the results of the check |
| element | Element-identifier for the result, which will be given back in the savelist |
| time | A vector of time elements with the length of the measurement vector |
| height | A vector of height elements with the length of the measurement vector |
| lat | A vector of latitude elements with the length of the measurement vector |
| lon | A vector of longitude elements with the length of the measurement vector |
| vec1 | An additional vector, which is named as vec1 |
| vec2 | An additional vector, which is named as vec2 |
| vec3 | An additional vector, which is named as vec3 |
| vec4 | An additional vector, which is named as vec4 |
| baseunit | The unit of the original measurement vector |
| savelist | A list with save elements |
| savelistcounter | Numbers of elements of the savelist |

Details

This function calls the described saving-function, which transform the resultlist elements to a sav-inglist element. The possible called functions are `qat_save_roc_rule_static_1d` and `qat_save_roc_rule_static_1d`. As a result the given savelist will get an additional entry.

Value

The given savelist will be returned, with included results of the functions which may be called in this function.

Author(s)

Andre Duesterhus

See Also

[qat_save_roc_rule_static_1d](#), [qat_save_roc_rule_dynamic_1d](#), [qat_run_workflow_save](#)

Examples

```
vec <- rnorm(100)
downward_vector<-seq(1,2,length.out=1000)
upward_vector<-seq(1,2,length.out=1000)
workflowlist_part <- list(downward_value=2, upward_value=2,downward_vector="vec1",upward_vector="vec2",downward
resultlist <- qat_call_roc_rule(vec, workflowlist_part, element=1, vec1=downward_vector, vec2=upward_vector)
savelist <- list()
savelistcounter <- 1
for (ii in 2:3) {
savelist <- qat_call_save_roc_rule(resultlist[[ii]], savelist=savelist, savelistcounter=savelistcounter)
if (length(which(names(savelist)=="element"))==0) {
savelistcounter<-length(savelist)
} else {
savelistcounter<-1
}
}
```

`qat_call_save_set_addup`

Produce a savelist-entry for a set Addup

Description

This function calls `qat_save_set_addup_1d`. As a result a part of a savelist is constructed, which can be used to construct a netCDF file.

Usage

```
qat_call_save_set_addup(resultlist_part, element = -999, time = NULL, height = NULL, lat = NULL, lon = N
```

Arguments

| | |
|-----------------|-----------------------------------------------------------------------------|
| resultlist_part | A list with the results of the check |
| element | Element-identifier for the result, which will be given back in the savelist |
| time | A vector of time elements with the length of the measurement vector |
| height | A vector of height elements with the length of the measurement vector |
| lat | A vector of latitude elements with the length of the measurement vector |
| lon | A vector of longitude elements with the length of the measurement vector |
| vec1 | An additional vector, which is named as vec1 |
| vec2 | An additional vector, which is named as vec2 |
| vec3 | An additional vector, which is named as vec3 |
| vec4 | An additional vector, which is named as vec4 |
| baseunit | The unit of the original measurement vector |
| savelist | A list with save elements |
| savelistcounter | Numbers of elements of the savelist |

Details

This function calls the described saving-function, which transform the resultlist elements to a savinglist element. The possible called function is `qat_save_set_addup_1d`. As a result the given savelist will get an additional entry.

Value

The given savelist will be returned, with included results of the functions which may be called in this function.

Author(s)

Andre Duesterhus

See Also

[qat_save_set_addup_1d](#), [qat_run_workflow_save](#)

Examples

```
## still to come
```

qat_call_save_set_mean

Produce a savelist-entry for a Set Mean

Description

This function calls `qat_save_set_mean_1d`. As a result a part of a savelist is constructed, which can be used to construct a netCDF file.

Usage

```
qat_call_save_set_mean(resultlist_part, element = -999, time = NULL, height = NULL, lat = NULL, lon = NU
```

Arguments

| | |
|------------------------------|-----------------------------------------------------------------------------|
| <code>resultlist_part</code> | A list with the results of the check |
| <code>element</code> | Element-identifier for the result, which will be given back in the savelist |
| <code>time</code> | A vector of time elements with the length of the measurement vector |
| <code>height</code> | A vector of height elements with the length of the measurement vector |
| <code>lat</code> | A vector of latitude elements with the length of the measurement vector |
| <code>lon</code> | A vector of longitude elements with the length of the measurement vector |
| <code>vec1</code> | An additional vector, which is named as <code>vec1</code> |
| <code>vec2</code> | An additional vector, which is named as <code>vec2</code> |
| <code>vec3</code> | An additional vector, which is named as <code>vec3</code> |
| <code>vec4</code> | An additional vector, which is named as <code>vec4</code> |
| <code>baseunit</code> | The unit of the original measurement vector |
| <code>savelist</code> | A list with save elements |
| <code>savelistcounter</code> | Numbers of elements of the savelist |

Details

This function calls the described saving-function, which transform the resultlist elements to a savelist element. The possible called function is `qat_save_set_mean_1d`. As a result the given savelist will get an additional entry.

Value

The given savelist will be returned, with included results of the functions which may be called in this function.

Author(s)

Andre Duesterhus

See Also

[qat_save_set_mean_1d](#), [qat_run_workflow_save](#)

Examples

```
## still to come
```

```
qat_call_save_set_nans
```

Produce a savelist-entry for a set NAN

Description

This function calls `qat_save_set_nans_1d`, `qat_save_set_nans_above_1d` or `qat_save_set_nans_below_1d`. As a result a part of a savelist is constructed, which can be used to construct a netCDF file.

Usage

```
qat_call_save_set_nans(resultlist_part, element = -999, time = NULL, height = NULL, lat = NULL, lon = NU
```

Arguments

| | |
|------------------------------|-----------------------------------------------------------------------------|
| <code>resultlist_part</code> | A list with the results of the check |
| <code>element</code> | Element-identifier for the result, which will be given back in the savelist |
| <code>time</code> | A vector of time elements with the length of the measurement vector |
| <code>height</code> | A vector of height elements with the length of the measurement vector |
| <code>lat</code> | A vector of latitude elements with the length of the measurement vector |
| <code>lon</code> | A vector of longitude elements with the length of the measurement vector |
| <code>vec1</code> | An additional vector, which is named as <code>vec1</code> |
| <code>vec2</code> | An additional vector, which is named as <code>vec2</code> |
| <code>vec3</code> | An additional vector, which is named as <code>vec3</code> |
| <code>vec4</code> | An additional vector, which is named as <code>vec4</code> |
| <code>baseunit</code> | The unit of the original measurement vector |
| <code>savelist</code> | A list with save elements |
| <code>savelistcounter</code> | Numbers of elements of the savelist |

Details

This function calls the described saving-function, which transform the resultlist elements to a savinglist element. The possible called functions are `qat_save_set_nans_1d`, `qat_save_set_nans_above_1d` or `qat_save_set_nans_below_1d`. As a result the given savelist will get an additional entry.

Value

The given savelist will be returned, with included results of the functions which may be called in this function.

Author(s)

Andre Duesterhus

See Also

[qat_save_set_nans_1d](#), [qat_save_set_nans_above_1d](#), [qat_save_set_nans_below_1d](#), [qat_run_workflow_save](#)

Examples

```
## still to come
```

```
qat_call_save_slide_distribution
```

Produce a savelist-entry for a Slide Distribution Test

Description

This function calls `qat_save_slide_distribution_1d`. As a result a part of a savelist is constructed, which can be used to construct a netCDF file.

Usage

```
qat_call_save_slide_distribution(resultlist_part, element = -999, time = NULL, height = NULL, lat = NUL
```

Arguments

| | |
|------------------------------|-----------------------------------------------------------------------------|
| <code>resultlist_part</code> | A list with the results of the check |
| <code>element</code> | Element-identifier for the result, which will be given back in the savelist |
| <code>time</code> | A vector of time elements with the length of the measurement vector |
| <code>height</code> | A vector of height elements with the length of the measurement vector |
| <code>lat</code> | A vector of latitude elements with the length of the measurement vector |
| <code>lon</code> | A vector of longitude elements with the length of the measurement vector |
| <code>vec1</code> | An additional vector, which is named as <code>vec1</code> |
| <code>vec2</code> | An additional vector, which is named as <code>vec2</code> |
| <code>vec3</code> | An additional vector, which is named as <code>vec3</code> |
| <code>vec4</code> | An additional vector, which is named as <code>vec4</code> |
| <code>baseunit</code> | The unit of the original measurement vector |
| <code>savelist</code> | A list with save elements |
| <code>savelistcounter</code> | Numbers of elements of the savelist |

Details

This function calls the described saving-function, which transform the resultlist elements to a sav-
inglist element. The possible called function is qat_save_slide_distribution_1d. As a result the
given savelist will get an additional entry.

Value

The given savelist will be returned, with included results of the functions which may be called in
this function.

Author(s)

Andre Duesterhus

See Also

[qat_save_slide_distribution_1d](#), [qat_run_workflow_save](#)

Examples

```
vec <- rnorm(100)
workflowlist_part <- list(blocksize=10)
resultlist <- qat_call_slide_distribution(vec, workflowlist_part, element=1)
savelist <- qat_call_save_slide_distribution(resultlist[[2]])
```

qat_call_save_trimmed_distribution

Produce a savelist-entry for a Trimmed Distribution Test

Description

This function calls qat_save_trimmed_distribution_1d. As a result a part of a savelist is con-
structed, which can be used to construct a netCDF file.

Usage

```
qat_call_save_trimmed_distribution(resultlist_part, element = -999, time = NULL, height = NULL, lat = N
```

Arguments

| | |
|-----------------|-----------------------------------------------------------------------------|
| resultlist_part | A list with the results of the check |
| element | Element-identifier for the result, which will be given back in the savelist |
| time | A vector of time elements with the length of the measurement vector |
| height | A vector of height elements with the length of the measurement vector |
| lat | A vector of latitude elements with the length of the measurement vector |

| | |
|-----------------|--------------------------------------------------------------------------|
| lon | A vector of longitude elements with the length of the measurement vector |
| vec1 | An additional vector, which is named as vec1 |
| vec2 | An additional vector, which is named as vec2 |
| vec3 | An additional vector, which is named as vec3 |
| vec4 | An additional vector, which is named as vec4 |
| baseunit | The unit of the original measurement vector |
| savelist | A list with save elements |
| savelistcounter | Numbers of elements of the savelist |

Details

This function calls the described saving-function, which transform the resultlist elements to a savinglist element. The possible called function is `qat_save_trimmed_distribution_1d`. As a result the given savelist will get an additional entry.

Value

The given savelist will be returned, with included results of the functions which may be called in this function.

Author(s)

Andre Duesterhus

See Also

[qat_save_trimmed_distribution_1d](#), [qat_run_workflow_save](#)

Examples

```
vec <- rnorm(1000)
workflowlist_part <- list(bootruns=1000)
resultlist <- qat_call_boot_distribution(vec, workflowlist_part, element=1)
savelist <- qat_call_save_boot_distribution(resultlist[[2]])
```

`qat_call_set_addup` *Addup values of a vector*

Description

This function adds up successive values of a given vector

Usage

```
qat_call_set_addup(measurement_vector, workflowlist_part, element = -999, time = NULL, height = NULL, l
```

Arguments

| | |
|--------------------|-------------------------------------------------------------------------------|
| measurement_vector | The measurement vector, which should be tested |
| workflowlist_part | A list with the parameters of the check |
| element | Element-identifier for the result, which will be given back in the resultlist |
| time | A vector of time elements with the length of the measurement vector |
| height | A vector of height elements with the length of the measurement vector |
| lat | A vector of latitude elements with the length of the measurement vector |
| lon | A vector of longitude elements with the length of the measurement vector |
| vec1 | An additional vector, which is named as vec1 |
| vec2 | An additional vector, which is named as vec2 |
| vec3 | An additional vector, which is named as vec3 |
| vec4 | An additional vector, which is named as vec4 |
| resultlist | A list with results of tests |
| resultlistcounter | Number of elements of the resultlist |

Details

This function calls the described method, which are defined by the parameters in the workflowlist_part. The possible called function is qat_analyse_set_addup_1d. As a result the function will give back a list, which include the corrected measurement vector.

Value

Give back a list, which include the vector with the results of the block.

Author(s)

Andre Duesterhus

See Also

[qat_analyse_set_addup_1d](#)

Examples

```
vec <- c(1,2,3,4,5,4,3,2,1)
workflowlist_part <- list(blocksize=3)
result <- qat_call_set_addup(vec, workflowlist_part)
```

| | |
|-------------------|-----------------------------------|
| qat_call_set_mean | <i>Mean of values of a vector</i> |
|-------------------|-----------------------------------|

Description

This function make a mean of successive values of a given vector.

Usage

```
qat_call_set_mean(measurement_vector, workflowlist_part, element = -999, time = NULL, height = NULL, la
```

Arguments

| | |
|--------------------|-------------------------------------------------------------------------------|
| measurement_vector | The measurement vector, which should be tested |
| workflowlist_part | A list with the parameters of the check |
| element | Element-identifier for the result, which will be given back in the resultlist |
| time | A vector of time elements with the length of the measurement vector |
| height | A vector of height elements with the length of the measurement vector |
| lat | A vector of latitude elements with the length of the measurement vector |
| lon | A vector of longitude elements with the length of the measurement vector |
| vec1 | An additional vector, which is named as vec1 |
| vec2 | An additional vector, which is named as vec2 |
| vec3 | An additional vector, which is named as vec3 |
| vec4 | An additional vector, which is named as vec4 |
| resultlist | A list with results of tests |
| resultlistcounter | Number of elements of the resultlist |

Details

This function calls the described method, which are defined by the parameters in the workflowlist_part. The possible called function is qat_analyse_set_mean_1d. As a result the function will give back a list, which include the corrected measurement vector.

Value

Give back a list, which include the vector with the results of the block.

Author(s)

Andre Duesterhus

See Also

[qat_analyse_set_mean_1d](#)

Examples

```
vec <- c(1,2,3,4,5,4,3,2,1)
workflowlist_part <- list(blocksize=3)
result <- qat_call_set_mean(vec, workflowlist_part)
```

qat_call_set_nans *Set given values of a vector to NaN*

Description

This function set a specified value of a vector to NaN.

Usage

```
qat_call_set_nans(measurement_vector, workflowlist_part, element = -999, time = NULL, height = NULL, lat = NULL, lon = NULL, vec1 = NULL, vec2 = NULL, vec3 = NULL, vec4 = NULL, resultlist = NULL, resultlistcounter = 0)
```

Arguments

| | |
|--------------------|-------------------------------------------------------------------------------|
| measurement_vector | The measurement vector, which should be tested |
| workflowlist_part | A list with the parameters of the check |
| element | Element-identifier for the result, which will be given back in the resultlist |
| time | A vector of time elements with the length of the measurement vector |
| height | A vector of height elements with the length of the measurement vector |
| lat | A vector of latitude elements with the length of the measurement vector |
| lon | A vector of longitude elements with the length of the measurement vector |
| vec1 | An additional vector, which is named as vec1 |
| vec2 | An additional vector, which is named as vec2 |
| vec3 | An additional vector, which is named as vec3 |
| vec4 | An additional vector, which is named as vec4 |
| resultlist | A list with results of tests |
| resultlistcounter | Number of elements of the resultlist |

Details

This function calls the described method, which are defined by the parameters in the workflowlist\part. The possible called functions are qat_analyse_set_nans_1d, qat_analyse_set_nans_above_1d and qat_analyse_set_nans_below_1d. As a result the function will give back a list, which include the corrected measurement vector.

Value

Give back a list, which include the measurement vector with the replaced values.

Author(s)

Andre Duesterhus

See Also

[qat_analyse_set_nans_1d](#)

Examples

```
vec <- c(1,2,3,4,5,4,3,2,1)
workflowlist_part <- list(nan_value=4)
result <- qat_call_set_nans(vec, workflowlist_part)
```

qat_call_slide_distribution

Perform a slide distribution check

Description

This function calls the described tests, which are defined by the parameters in the workflowlist_part. The possible called functions are qat_analyse_slide_distribution_1d.

Usage

```
qat_call_slide_distribution(measurement_vector, workflowlist_part, element = -999, time = NULL, height)
```

Arguments

| | |
|--------------------|-------------------------------------------------------------------------------|
| measurement_vector | The measurement vector, which should be tested |
| workflowlist_part | A list with the parameters of the check |
| element | Element-identifier for the result, which will be given back in the resultlist |
| time | A vector of time elements with the length of the measurement vector |
| height | A vector of height elements with the length of the measurement vector |
| lat | A vector of latitude elements with the length of the measurement vector |
| lon | A vector of longitude elements with the length of the measurement vector |
| vec1 | An additional vector, which is named as vec1 |
| vec2 | An additional vector, which is named as vec2 |
| vec3 | An additional vector, which is named as vec3 |
| vec4 | An additional vector, which is named as vec4 |
| resultlist | A list with results of tests |
| resultlistcounter | Number of elements of the resultlist |

Details

This function calls the described tests, which are defined by the parameters in the workflowlist_part. The possible called functions are qat_analyse_slide_distribution_1d. As a result the resultlist will get additional entries, which are defined by the tests, which may called by this function.

Value

The given resultlist will be returned, with included results of the functions which may called in this function.

Author(s)

Andre Duesterhus

See Also

[qat_analyse_slide_distribution_1d](#)

Examples

```
vec <- rnorm(100)
workflowlist_part <- list(blocksize=10)
result <- qat_call_slide_distribution(vec, workflowlist_part)
```

qat_call_trimmed_distribution

Perform a trimmed distribution check

Description

This function calls the described tests, which are defined by the parameters in the workflowlist_part. The possible called functions are qat_analyse_trimmed_distribution_1d.

Usage

```
qat_call_trimmed_distribution(measurement_vector, workflowlist_part, element = -999, time = NULL, height = NULL)
```

Arguments

measurement_vector

The measurement vector, which should be tested

workflowlist_part

A list with the parameters of the check

element

Element-identifier for the result, which will be given back in the resultlist

time

A vector of time elements with the length of the measurement vector

height

A vector of height elements with the length of the measurement vector

| | |
|-------------------|--------------------------------------------------------------------------|
| lat | A vector of latitude elements with the length of the measurement vector |
| lon | A vector of longitude elements with the length of the measurement vector |
| vec1 | An additional vector, which is named as vec1 |
| vec2 | An additional vector, which is named as vec2 |
| vec3 | An additional vector, which is named as vec3 |
| vec4 | An additional vector, which is named as vec4 |
| resultlist | A list with results of tests |
| resultlistcounter | Number of elements of the resultlist |

Details

This function calls the described tests, which are defined by the parameters in the workflowlist_part. The possible called functions are qat_analyse_trimmed_distribution_1d. As a result the resultlist will get additional entries, which are defined by the tests, which may called by this function.

Value

The given resultlist will be returned, with included results of the functions which may called in this function.

Author(s)

Andre Duesterhus

See Also

[qat_analyse_trimmed_distribution_1d](#)

Examples

```
vec <- rnorm(1000)
workflowlist_part <- list()
result <- qat_call_trimmed_distribution(vec, workflowlist_part)
```

qat_config_read_workflow

Read an XML workflow

Description

This functions read a XML-workflow-file.

Usage

```
qat_config_read_workflow(filename)
```

Arguments

filename Path and filename of the xml-file, in which the workflow is defined

Details

This functions read a file, which got a XML-workflow in it. This will be transformed to a workflowlist, which may be processed by `qat_run_workflow_check`.

Value

A workflowlist, which consists of the tests and its parameters, which should be performed.

Author(s)

Andre Duesterhus

See Also

[qat_run_workflow_check](#)

Examples

```
library("qat")
# read in workflow from systemfiles
filename_in <- system.file("extdata/workflowexample.xml", package="qat")
workflowlist <- qat_config_read_workflow(filename_in)
```

```
qat_config_write_workflow
      Write a result
```

Description

A workflowlist will be stored into a XML-file.

Usage

```
qat_config_write_workflow(workflowlist, name = "", description = "", author = "", date = "", sample_time = "")
```

Arguments

workflowlist A workflowlist, which may be loaded by `qat_config_read_workflow` and used for tests.

name Name of the tests, which were performed with this workflowlist

description Description of the workflowlist

author Author who used the workflowlist for a test.

date Date of the test.

| | |
|-------------------|--------------------------------------------------------------------------------------|
| sample_time_start | Start time of the sample, which was tested |
| sample_time_stop | End time of the sample, which was tested |
| sample_place | Location of the sample, which was tested |
| config_filename | A filename of the configuration file, which was read in at qat\config\read\workflow. |
| output_filename | Filename, where the result should be stored. |

Details

The workflow will be stored at the location of `output_filename`. As additional information in the header of this file, the other arguments will be used.

Value

The information, which was stored, will be given back.

Author(s)

Andre Duesterhus

See Also

[qat_config_read_workflow](#)

Examples

```
library("qat")
# read in workflow from systemfiles
filename_in <- system.file("extdata/workflowexample.xml", package="qat")
workflowlist <- qat_config_read_workflow(filename_in)
# add some more informations for the workflow
workflowlist <- qat_add_all_descriptions(workflowlist)
workflowlist <- qat_add_all_algorithms(workflowlist)

filename_out <- "myworkflow_result.xml"
# write edited workflow in current directory
qat_config_write_workflow(workflowlist, output_filename=filename_out)
```

qat_data_nameofvars_ncdf

Variable names of ncdf object

Description

Give back the names of the variables in a netCDF-object.

Usage

```
qat_data_nameofvars_ncdf(obj)
```

Arguments

obj A netcdf object, which will be produced by qat_data_read_ncdf.

Details

The names of the variables, which are stored in the ncdf-object will be given back as a list.

Value

List of names.

Author(s)

Andre Duesterhus

See Also

[qat_data_read_ncdf](#), [qat_data_numofvars_ncdf](#), [qat_data_varcontent_ncdf](#)

Examples

```
#still to come
```

qat_data_numofvars_ncdf

Number of Variables of netcdf object

Description

Give back the number of the variables in a netCDF-object.

Usage

qat_data_numofvars_ncdf(obj)

Arguments

obj A netcdf object, which will be produced by qat_data_read_ncdf.

Details

The number of variables, which are stored in the netcdf-object will be given back.

Value

Number of variables.

Author(s)

Andre Duesterhus

See Also

[qat_data_read_ncdf](#), [qat_data_nameofvars_ncdf](#) [qat_data_varcontent_ncdf](#)

Examples

#still to come

qat_data_read_ncdf *Read in netCDF-file*

Description

A netCDF file will be read in and a ncdf-object will be given back.

Usage

```
qat_data_read_ncdf(filename)
```

Arguments

filename Path and filename of the netCDF-file, which should be read in.

Value

A ncdf-Object, with the content of the file.

Author(s)

Andre Duesterhus

See Also

[qat_data_nameofvars_ncdf](#), [qat_data_numofvars_ncdf](#), [qat_data_varcontent_ncdf](#)

Examples

```
#still to come
```

qat_data_varcontent_ncdf
 Content of a variable

Description

Give back the content of a specified variable of an ncdf-object.

Usage

```
qat_data_varcontent_ncdf(obj, numofvar)
```

Arguments

obj A netcdf object, which will be produced by qat_data_read_ncdf.
numofvar Number of variable, which content should be delivered.

Details

The content of the variable, which is specified by its number in numofvars will be given back.

Value

The content of the variable.

Author(s)

Andre Duesterhus

See Also

[qat_data_read_ncdf](#), [qat_data_nameofvars_ncdf](#) [qat_data_numofvars_ncdf](#)

Examples

```
#still to come
```

```
qat_plot_block_distribution_1d
    Plot a block distribution check result
```

Description

A plot of the result of a block distribution check will be produced.

Usage

```
qat_plot_block_distribution_1d(resultlist, filename, blocksize = -1, measurement_name = "", directoryname)
```

Arguments

| | |
|------------------|--------------------------------------------------------|
| resultlist | List of results from qat\analyse\block\distribution\1d |
| filename | Name of the file without extension. |
| blocksize | Length of the blocks |
| measurement_name | Name of the measurement. |
| directoryname | Directory, where the resulted file should be stored. |
| plotstyle | A list with a qat color scheme. |

Details

A plot will be produced, which base on the resulting flagvector of qat\analyse\block\distribution\1d. Additional information on the parameters, which were used while performing the test, will be included into the plot. When no plotstyle is defined the standard-colorscheme will be used. The resulting plot will be stored in the folder, which is defined by directory under the given filename, with the extension png.

Value

No return value.

Author(s)

Andre Duesterhus

See Also

[qat_analyse_block_distribution_1d](#)

Examples

```
vec <- rnorm(1000)
result <- qat_analyse_block_distribution_1d(vec, 50)
# this example produce a file exampleplot_blockdist.png in the current directory
qat_plot_block_distribution_1d(result$stat, "exampleplot_blockdist", blocksize=result$blocksize, measurement_name="")
```

qat_plot_boot_distribution_1d

Plot a bootstrapped distribution check result

Description

A plot of the result of a booted distribution check will be produced.

Usage

```
qat_plot_boot_distribution_1d(resultlist_stat, filename, bootruns = -1, measurement_name = "", directoryname = "")
```

Arguments

| | |
|------------------|-------------------------------------------------------|
| resultlist_stat | List of results from qat\analyse\boot\distribution\1d |
| filename | Name of the file without extension. |
| bootruns | Number of bootstrap runs used in the test. |
| measurement_name | Name of the measurement. |
| directoryname | Directory, where the resulted file should be stored. |
| plotstyle | A list with a qat color scheme. |

Details

A plot will be produced, which base on the resulting vectors of qat\analyse\boot\distribution\1d. When no plotstyle is defined the standard-colorscheme will be used. The resulting plot will be stored in the folder, which is defined by directory under the given filename, with the extension png.

Value

No return value.

Author(s)

Andre Duesterhus

See Also

[qat_analyse_boot_distribution_1d](#)

Examples

```
vec <- rnorm(1000)
result <- qat_analyse_boot_distribution_1d(vec, 1000)
# this example produce a file exampleplot_bootdist.png in the current directory
qat_plot_boot_distribution_1d(result$stat, "exampleplot_bootdist", bootruns=result$bootruns, measurement_name="
```

qat_plot_distribution_1d

Plot a distribution check result

Description

A plot of the result of a distribution check will be produced.

Usage

```
qat_plot_distribution_1d(resultlist_hist, filename, resultlist_stat, numofbars = -1, measurement_name
```

Arguments

| | |
|------------------|------------------------------------------------------|
| resultlist_hist | Result of a hist function. |
| filename | Name of the file without extension. |
| resultlist_stat | List of statistical parameters. |
| numofbars | Numbers of bars of the histogram plot. |
| measurement_name | Name of the measurement. |
| directoryname | Directory, where the resulted file should be stored. |
| plotstyle | A list with a qat color scheme. |

Details

A plot will be produced, which base on the resulting flagvector of `qat\analyse\distribution\1d`. Additional information on the parameters, which were used while performing the test, will be included into the plot. When no plotstyle is defined the standard-colorscheme will be used. The resulting plot will be stored in the folder, which is defined by directory under the given filename, with the extension png.

Value

No return value.

Author(s)

Andre Duesterhus

See Also

[qat_analyse_distribution_1d](#)

Examples

```
vec <- rnorm(1000)
result <- qat_analyse_distribution_1d(vec, 15)
# this example produce a file exampleplot_dist.png in the current directory
qat_plot_distribution_1d(result$hist, "exampleplot_dist", result$stat, numofbars=result$numofbars, measurement_
```

```
qat_plot_lim_rule_dynamic_1d
Plot a dynamic LIM rule result
```

Description

A plot of the result of a dynamic LIM rule check will be produced.

Usage

```
qat_plot_lim_rule_dynamic_1d(flagvector, filename, measurement_vector = NULL, min_vector = NULL, max_v
```

Arguments

| | |
|---------------------------------|-----------------------------------------------------------------------------|
| <code>flagvector</code> | The resulting flagvector of <code>qat\analyse\lim_rule_dynamic_1d</code> |
| <code>filename</code> | Name of the file without extension. |
| <code>measurement_vector</code> | The measurement vector, which should be plotted |
| <code>min_vector</code> | The vector with the minimum values. |
| <code>max_vector</code> | The vector with the maximum values. |

| | |
|------------------|------------------------------------------------------|
| min_vector_name | Name of the vector of the minimum values. |
| max_vector_name | Name of the vector of the minimal values. |
| measurement_name | Name of the measurement. |
| directoryname | Directory, where the resulted file should be stored. |
| plotstyle | A list with a qat color scheme. |

Details

A plot will be produced, which base on the resulting flagvector of `qat_analyse_lim_rule_dynamic_1d`. With additional information on the parameters, which were used while performing the test, this function will produce a more detailed plot. When no `plotstyle` is defined the standard-colorscheme will be used. The resulting plot will be stored in the folder, which is defined by `directory` under the given filename, with the extension `png`.

Value

No return value.

Author(s)

Andre Duesterhus

See Also

[qat_analyse_lim_rule_dynamic_1d](#), [qat_plot_lim_rule_static_1d](#), [qat_plot_lim_rule_sigma_1d](#)

Examples

```
vec <- rnorm(1000)
min_vector<-seq(-1,-2,length.out=1000)
max_vector<-seq(1,2,length.out=1000)
result <- qat_analyse_lim_rule_dynamic_1d(vec, min_vector, max_vector, min_vector_name="minimum vector", max_vect
# this example produce a file exampleplot_lim_dyn.png in the current directory
qat_plot_lim_rule_dynamic_1d(result$flagvector, "exampleplot_lim_dyn", measurement_vector=vec, min_vector=resul
```

```
qat_plot_lim_rule_sigma_1d
```

Plot a sigma LIM rule result

Description

A plot of the result of a dynamic lim rule check will be produced.

Usage

```
qat_plot_lim_rule_sigma_1d(flagvector, filename, measurement_vector = NULL, sigma_factor = NULL, meano
```

Arguments

| | |
|--------------------|------------------------------------------------------------------------|
| flagvector | The resulting flagvector of <code>qat_analyse_lim_rule_sigma_1d</code> |
| filename | Name of the file without extension. |
| measurement_vector | The measurement vector, which should be plotted |
| sigma_factor | The sigma factor, which was used, when the test were performed. |
| meanofvector | The mean of the measurement vector |
| sdofvector | The standard deviation of the measurement vector |
| measurement_name | Name of the measurement. |
| directoryname | Directory, where the resulted file should be stored. |
| plotstyle | A list with a qat color scheme. |

Details

A plot will be produced, which base on the resulting flagvector of `qat_analyse_lim_rule_sigma_1d`. With additional information on the parameters, which were used while performing the test, this function will produce a more detailed plot. When no plotstyle is defined the standard-colorscheme will be used. The resulting plot will be stored in the folder, which is defined by directory under the given filename, with the extension png.

Value

No return value

Author(s)

Andre Duesterhus

See Also

[qat_analyse_lim_rule_static_1d](#), [qat_plot_lim_rule_dynamic_1d](#), [qat_plot_lim_rule_sigma_1d](#)

Examples

```
vec <- rnorm(1000)
result <- qat_analyse_lim_rule_sigma_1d(vec, 2)
# this example produce a file exampleplot_lim_sig.png in the current directory
qat_plot_lim_rule_sigma_1d(result$flagvector, "exampleplot_lim_sig", measurement_vector=vec, sigma_factor=result
```

`qat_plot_lim_rule_static_1d`*Plot a static lim rule result*

Description

A plot of the result of a dynamic LIM rule check will be produced.

Usage

```
qat_plot_lim_rule_static_1d(flagvector, filename, measurement_vector = NULL, min_value = NULL, max_val
```

Arguments

| | |
|---------------------------------|-------------------------------------------------------------------------|
| <code>flagvector</code> | The resulting flagvector of <code>qat\analyse\lim\rule\static\1d</code> |
| <code>filename</code> | Name of the file without extension. |
| <code>measurement_vector</code> | The measurement vector, which should be plotted |
| <code>min_value</code> | The used minimum value of the test. |
| <code>max_value</code> | The used maximum value of the test. |
| <code>measurement_name</code> | Name of the measurement. |
| <code>directoryname</code> | Directory, where the resulted file should be stored. |
| <code>plotstyle</code> | A list with a qat color scheme. |

Details

A plot will be produced, which base on the resulting flagvector of `qat\analyse\lim\rule\static\1d`. With additional information on the parameters, which were used while performing the test, this function will produce a more detailed plot. When no `plotstyle` is defined the standard-colorscheme will be used. The resulting plot will be stored in the folder, which is defined by `directory` under the given `filename`, with the extension `png`.

Value

No return value.

Author(s)

Andre Duesterhus

See Also

[qat_analyse_lim_rule_static_1d](#), [qat_plot_lim_rule_dynamic_1d](#), [qat_plot_lim_rule_sigma_1d](#)

Examples

```
vec <- rnorm(1000)
result <- qat_analyse_lim_rule_static_1d(vec, -2,2)
# this example produce a file exampleplot_lim_sta.png in the current directory
qat_plot_lim_rule_static_1d(result$flagvector, "exampleplot_lim_sta", measurement_vector=vec, min_value=result$
```

qat_plot_noc_rule_1d *Plot a NOC rule result*

Description

A plot of the result of a NOC rule check will be produced.

Usage

```
qat_plot_noc_rule_1d(flagvector, filename, measurement_vector = NULL, max_return_elements = 0, measure
```

Arguments

flagvector The resulting flagvector of qat_analyse_noc_rule_1d

filename Name of the file without extension.

measurement_vector The measurement vector, which should be plotted

max_return_elements The number of maximum reruning elements, which was used in the test.

measurement_name Name of the measurement.

directoryname Directory, where the resulted file should be stored.

plotstyle A list with a qat color scheme.

Details

A plot will be produced, which base on the resulting flagvector of qat_analyse_noc_rule_1d. With additional information on the parameters, which were used while performing the test, this function will produce a more detailed plot. When no plotstyle is defined the standard-colorscheme will be used. The resulting plot will be stored in the folder, which is defined by directory under the given filename, with the extension png.

Value

No return value.

Author(s)

Andre Duesterhus

See Also[qat_analyse_noc_rule_1d](#)**Examples**

```
vec <- c(1,2,3,4,4,4,5,5,4,3,NaN,3,2,1)
result <- qat_analyse_noc_rule_1d(vec, 1)
# this example produce a file exampleplot_noc.png in the current directory
qat_plot_noc_rule_1d(result$flagvector, "exampleplot_noc", measurement_vector=vec, max_return_elements=result$M
```

```
qat_plot_roc_rule_dynamic_1d
```

Plot a dynamic ROC rule result

Description

A plot of the result of a dynamic ROC rule check will be produced.

Usage

```
qat_plot_roc_rule_dynamic_1d(flagvector, filename, measurement_vector = NULL, max_upward_vector = NULL
```

Arguments

| | |
|----------------------|----------------------------------------------------------------|
| flagvector | The resulting flagvector of qat\analyse\roc_rule_dynamic_1d |
| filename | Name of the file without extension. |
| measurement_vector | The measurement vector, which should be plotted |
| max_upward_vector | The vector with the upward values. |
| max_downward_vector | The vector with the downward values. |
| upward_vector_name | Name of the vector of the upward values. |
| downward_vector_name | Name of the vector of the downward values. |
| measurement_name | Name of the measurement. |
| directoryname | Directory, where the resulted file should be stored. |
| plotstyle | A list with a qat color scheme. |

Details

A plot will be produced, which base on the resulting flagvector of qat\analyse\roc_rule_dynamic_1d. With additional information on the parameters, which were used while performing the test, this function will produce a more detailed plot. When no plotstyle is defined the standard-colorscheme will be used. The resulting plot will be stored in the folder, which is defined by directory under the given filename, with the extension png.

Value

No return value.

Author(s)

Andre Duesterhus

See Also

[qat_analyse_roc_rule_dynamic_1d](#), [qat_plot_roc_rule_static_1d](#)

Examples

```
vec <- rnorm(100)
min_vector<-seq(1,2,length.out=100)
max_vector<-seq(1,2,length.out=100)
result <- qat_analyse_roc_rule_dynamic_1d(vec, min_vector, max_vector, upward_vector_name="upward vector", downward_vector_name="downward vector")
# this example produce a file exampleplot_roc_dyn.png in the current directory
qat_plot_roc_rule_dynamic_1d(result$flagvector, "exampleplot_roc_dyn", measurement_vector=vec, max_upward_value=2, max_downward_value=2)
```

qat_plot_roc_rule_static_1d

Plot a static ROC rule result

Description

A plot of the result of a static ROC rule check will be produced.

Usage

```
qat_plot_roc_rule_static_1d(flagvector, filename, measurement_vector = NULL, max_upward_value = 0, max_downward_value = 0, measurement_name = "", directoryname = "", plotstyle = "qat")
```

Arguments

| | |
|--------------------|-------------------------------------------------------------------------|
| flagvector | The resulting flagvector of <code>qat_analyse_roc_rule_static_1d</code> |
| filename | Name of the file without extension. |
| measurement_vector | The measurement vector, which should be plotted |
| max_upward_value | The used maximum upward value. |
| max_downward_value | The used maximum downward value. |
| measurement_name | Name of the measurement. |
| directoryname | Directory, where the resulted file should be stored. |
| plotstyle | A list with a qat color scheme. |

Details

A plot will be produced, which base on the resulting flagvector of `qat_analyse_roc_rule_static_1d`. With additional information on the parameters, which were used while performing the test, this function will produce a more detailed plot. When no plotstyle is defined the standard-colorscheme will be used. The resulting plot will be stored in the folder, which is defined by `directory` under the given filename, with the extension `png`.

Value

No return value.

Author(s)

Andre Duesterhus

See Also

[qat_analyse_roc_rule_static_1d](#), [qat_plot_roc_rule_dynamic_1d](#)

Examples

```
vec <- rnorm(100)
result <- qat_analyse_roc_rule_static_1d(vec, 2,2)
# this example produce a file exampleplot_roc_sta.png in the current directory
qat_plot_roc_rule_static_1d(result$flagvector, "exampleplot_roc_sta", measurement_vector=vec, max_upward_value=
```

`qat_plot_slide_distribution_1d`

Plot a slide distribution check result

Description

A plot of the result of a slide distribution check will be produced.

Usage

```
qat_plot_slide_distribution_1d(resultlist, filename, blocksize = -1, measurement_name = "", directoryname = "", plotstyle = "standard")
```

Arguments

| | |
|-------------------------------|---------------------------------------------------------------------|
| <code>resultlist</code> | List of results from <code>qat_analyse_slide_distribution_1d</code> |
| <code>filename</code> | Name of the file without extension. |
| <code>blocksize</code> | Length of the blocks |
| <code>measurement_name</code> | Name of the measurement. |
| <code>directoryname</code> | Directory, where the resulted file should be stored. |
| <code>plotstyle</code> | A list with a qat color scheme. |

Details

A plot will be produced, which base on the resulting flagvector of `qat\analyse\slide\distribution\1d`. Additional information on the parameters, which were used while performing the test, will be included into the plot. When no `plotstyle` is defined the standard-colorscheme will be used. The resulting plot will be stored in the folder, which is defined by `directoryname` under the given filename, with the extension `png`.

Value

No return value.

Author(s)

Andre Duesterhus

See Also

[qat_analyse_slide_distribution_1d](#)

Examples

```
vec <- rnorm(100)
result <- qat_analyse_slide_distribution_1d(vec, 10)
# this example produce a file exampleplot_slidedist.png in the current directory
qat_plot_slide_distribution_1d(result$stat, "exampleplot_slidedist", blocksize=result$blocksize, measurement_name="")
```

```
qat_plot_trimmed_distribution_1d
```

Plot a trimmed distribution check result

Description

A plot of the result of a trimmed distribution check will be produced.

Usage

```
qat_plot_trimmed_distribution_1d(resultlist, filename, measurement_name = "", directoryname = "", plotstyle = "standard")
```

Arguments

| | |
|-------------------------------|-----------------------------------------------------------------------|
| <code>resultlist</code> | List of results from <code>qat\analyse\trimmed\distribution\1d</code> |
| <code>filename</code> | Name of the file without extension. |
| <code>measurement_name</code> | Name of the measurement. |
| <code>directoryname</code> | Directory, where the resulted file should be stored. |
| <code>plotstyle</code> | A list with a qat color scheme. |

Details

A plot will be produced, which base on the resulting flagvector of `qat_analyse_trimmed_distribution_1d`. When no `plotstyle` is defined the `standard-colorscheme` will be used. The resulting plot will be stored in the folder, which is defined by `directory` under the given `filename`, with the extension `png`.

Value

No return value.

Author(s)

Andre Duesterhus

See Also

[qat_analyse_trimmed_distribution_1d](#)

Examples

```
vec <- rnorm(1000)
result <- qat_analyse_trimmed_distribution_1d(vec)
# this example produce a file exampleplot_trimmeddist.png in the current directory
qat_plot_trimmed_distribution_1d(result$stat, "exampleplot_trimmeddist", measurement_name="Result of Check")
```

`qat_read_parameter` *Informations on a method*

Description

This functions delivers informations of methods, which are stored under the given `filename`.

Usage

```
qat_read_parameter(filename, methodname)
```

Arguments

| | |
|-------------------------|-----------------------------------------------------------|
| <code>filename</code> | Filename of the file with the descriptions of the methods |
| <code>methodname</code> | Name of the method, where informations are required. |

Details

This functions delivers informations of methods, which are stored under the given `filename`. For this the `methodname` will be used as a search parameter. The informations will be given back as a list.

Value

A list with the following elements:

| | |
|-----------------------|---------------------------------------------------------------------------|
| name | Name of the method, may be corrected to standard name. |
| analysis_function | Name of the analysis function, which should be called for this method |
| plot_function | Name of the plot function, which should be called for this method |
| manipulation_function | Name of the manipulation function, which should be called for this method |
| description | Description of the method |
| algorithm | Algorithm of the method |

Author(s)

Andre Duesterhus

Examples

```
#still to come
```

```
qat_run_workflow_check
      Perform a workflow of checks
```

Description

This function performs a workflow of checks by a given workflowlist on a given vector.

Usage

```
qat_run_workflow_check(measurement_vector, workflowlist, time = NULL, height = NULL, lat = NULL, lon = NULL)
```

Arguments

| | |
|--------------------|------------------------------------------------|
| measurement_vector | The measurement vector, which should be tested |
| workflowlist | The workflowlist, which should be performed. |
| time | A time vector of the measurement_vector |
| height | A height vector of the measurement_vector |
| lat | A latitude vector of the measurement_vector |
| lon | A longitude vector of the measurement_vector |
| vec1 | A potential additional vector |
| vec2 | A potential additional vector |
| vec3 | A potential additional vector |
| vec4 | A potential additional vector |

Details

This function performs a workflow of checks by a given workflowlist on a given measurement vector. Additional vectors can be used in the tests.

Value

A resultlist, with the results of the performed tests will be given back.

Author(s)

Andre Duesterhus

See Also

[qat_config_read_workflow](#), [qat_run_workflow_plot](#)

Examples

```
library("qat")
# define testvector
testvector<-rnorm(500)
# read in workflow from systemfiles
filename_in <- system.file("extdata/workflowexample.xml", package="qat")
workflowlist <- qat_config_read_workflow(filename_in)
# define some additional vectors
maxlim <- seq(3,1,length.out=500)
minlim <- seq(-1,-3,length.out=500)
uproc <- seq(1,3,length.out=500)
downroc <- seq(3,1,length.out=500)
# run the workflow on the testvector
rlist <- qat_run_workflow_check(testvector,workflowlist,vec1=maxlim, vec2=minlim, vec3=uproc, vec4=downroc)
# produce some plots of the result in teh current directory
qat_run_workflow_plot(rlist, measurement_name="Test", basename="test")
# add some more informations for the workflow
workflowlist <- qat_add_all_descriptions(workflowlist)
workflowlist <- qat_add_all_algorithms(workflowlist)
workflowlist <- qat_add_comment(workflowlist, 1, "No problems")

filename_out <- "myworkflow_result.xml"
# write edited workflow in current directory
qat_config_write_workflow(workflowlist, output_filename=filename_out)
```

qat_run_workflow_plot *Produce plots of a workflow*

Description

This function produces plots of the results, which were produced by a workflow.

Usage

```
qat_run_workflow_plot(resultlist, measurement_name = "", directoryname = "", basename = "", plotstyle =
```

Arguments

| | |
|------------------|--------------------------------------------------------------|
| resultlist | The results, which are produced by qat_run_workflow_check |
| measurement_name | The measurement vector, which is used at the tests |
| directoryname | Directory, where the resulting plots should be stored |
| basename | Basic name of the filename |
| plotstyle | A list with a qat color scheme. |

Details

The resultlist contains the parameters and results of the tests. From this the plots will be constructed and stored in the given directory. As filename the basename is used, with further extensions to indicate the tests. When no plotstyle is defined the standard-colorscheme will be used.

Value

No return value.

Author(s)

Andre Duesterhus

See Also

[qat_run_workflow_check](#)

Examples

```
library("qat")
# define testvector
testvector<-rnorm(500)
# read in workflow from systemfiles
filename_in <- system.file("extdata/workflowexample.xml", package="qat")
workflowlist <- qat_config_read_workflow(filename_in)
# define some additional vectors
maxlim <- seq(3,1,length.out=500)
minlim <- seq(-1,-3,length.out=500)
uproc <- seq(1,3,length.out=500)
downroc <- seq(3,1,length.out=500)
# run the workflow on the testvector
rlist <- qat_run_workflow_check(testvector,workflowlist,vec1=maxlim, vec2=minlim, vec3=uproc, vec4=downroc)
# produce some plots of the result in teh current directory
qat_run_workflow_plot(rlist, measurement_name="Test", basename="test")
# add some more informations for the workflow
workflowlist <- qat_add_all_descriptions(workflowlist)
workflowlist <- qat_add_all_algorithms(workflowlist)
```

```

workflowlist <- qat_add_comment(workflowlist, 1, "No problems")

filename_out <- "myworkflow_result.xml"
# write edited workflow in current directory
qat_config_write_workflow(workflowlist, output_filename=filename_out)

```

qat_run_workflow_save *Performing a workflow of constructing saving elements by a given resultlist*

Description

This function performs a workflow of constructing a savelist by a given resultlist.

Usage

```
qat_run_workflow_save(resultlist, baseunit = "", time = NULL, height = NULL, lat = NULL, lon = NULL, vec1 = NULL, vec2 = NULL, vec3 = NULL, vec4 = NULL)
```

Arguments

| | |
|------------|-----------------------------------------------|
| resultlist | Resultlist with results of checks |
| baseunit | Unit of the original measurement vector |
| time | A time vector of the measurement_vector |
| height | A height vector of the measurement_vector |
| lat | A latitude vector of the measurement_vector |
| lon | A longitude vector of the measurement_vector |
| vec1 | A potential additional vector |
| vec2 | A potential additional vector |
| vec3 | A potential additional vector |
| vec4 | A potential additional vector |

Details

This function performs a workflow of constructing a savelist by a given resultlist. This can be used to build netCDF-files by the function `qat_save_result_ncdf`.

Value

A savelist, with the results of the performed tests will be given back.

Author(s)

Andre Duesterhus

See Also

[qat_config_read_workflow](#), [qat_run_workflow_check](#), [qat_run_workflow_plot](#)

Examples

```
library("qat")
# define testvector
testvector<-rnorm(500)
# read in workflow from systemfiles
filename_in <- system.file("extdata/workflowexample.xml", package="qat")
workflowlist <- qat_config_read_workflow(filename_in)
# define some additional vectors
maxlim <- seq(3,1,length.out=500)
minlim <- seq(-1,-3,length.out=500)
uproc <- seq(1,3,length.out=500)
downroc <- seq(3,1,length.out=500)
# run the workflow on the testvector
rlist <- qat_run_workflow_check(testvector,workflowlist,vec1=maxlim, vec2=minlim, vec3=uproc, vec4=downroc)
# produce the savelist
savelist <- qat_run_workflow_save(rlist)
filename_out <- "myresults.nc"
# write netCDF-file of the results in current directory
qat_save_result_ncdf(testvector, savelist=savelist, filename_out, workflowlist=workflowlist ,vec1=maxlim, vec2=m
```

qat_save_block_distribution_1d

Produce a savelist from a resultlist for a Block Distribution Test

Description

This function takes the results, produced by `qat\analyse\block\distribution\1d` and construct a savelist, which may be used to produce a netCDF output.

Usage

```
qat_save_block_distribution_1d(resultlist_part, baseunit = "")
```

Arguments

| | |
|-----------------|---------------------------------------------|
| resultlist_part | A list with the results of the check |
| baseunit | The unit of the original measurement vector |

Details

This function takes the resultlist and transfer the content to a newly organized list. This also consists of more information, which help to generate an output like a netCDF-file.

Value

Returning a savelist with the content of the resultlist.

Author(s)

Andre Duesterhus

See Also

[qat_call_save_block_distribution](#), [qat_run_workflow_save](#)

Examples

```
vec <- rnorm(1000)
result <- list(result=qat_analyse_block_distribution_1d(vec, 50))
savelist <- qat_save_block_distribution_1d(result)
```

qat_save_boot_distribution_1d

Produce a savelist from a resultlist for a Boot Distribution Test

Description

This function takes the results, produced by `qat_analyse_boot_distribution_1d` and construct a savelist, which may be used to produce a netCDF output.

Usage

```
qat_save_boot_distribution_1d(resultlist_part, baseunit = "")
```

Arguments

| | |
|-----------------|---------------------------------------------|
| resultlist_part | A list with the results of the check |
| baseunit | The unit of the original measurement vector |

Details

This function takes the resultlist and transfer the content to a newly organized list. This also consists of more information, which help to generate an output like a netCDF-file.

Value

Returning a savelist with the content of the resultlist.

Author(s)

Andre Duesterhus

See Also

[qat_call_save_boot_distribution](#), [qat_run_workflow_save](#)

Examples

```
vec <- rnorm(1000)
result <- list(result=qat_analyse_boot_distribution_1d(vec, 1000))
savelist <- qat_save_boot_distribution_1d(result)
```

qat_save_distribution_1d

Produce a savelist from a resultlist for a Distribution Test

Description

This function takes the results, produced by `qat_analyse_distribution_1d` and construct a savelist, which may be used to produce a netCDF output.

Usage

```
qat_save_distribution_1d(resultlist_part, baseunit = "")
```

Arguments

| | |
|-----------------|-------------------------------------------------------------------------------------|
| resultlist_part | |
| baseunit | A list with the results of the check The unit of the original measurement vector |

Details

This function takes the resultlist and transfer the content to a newly organized list. This also consists of more information, which help to generate an output like a netCDF-file.

Value

Returning a savelist with the content of the resultlist.

Author(s)

Andre Duesterhus

See Also

[qat_call_save_distribution](#), [qat_run_workflow_save](#)

Examples

```
vec <- rnorm(1000)
result <- list(result=qat_analyse_distribution_1d(vec, 15))
savelist <- qat_save_distribution_1d(result)
```

`qat_save_lim_rule_dynamic_1d`*Produce a savelist from a resultlist for a LIM Rule Dynamic Test*

Description

This function takes the results, produced by `qat\analyse\lim_rule_dynamic_1d` and construct a savelist, which may be used to produce a netCDF output.

Usage

```
qat_save_lim_rule_dynamic_1d(resultlist_part, baseunit = "")
```

Arguments

| | |
|------------------------------|---------------------------------------------|
| <code>resultlist_part</code> | A list with the results of the check |
| <code>baseunit</code> | The unit of the original measurement vector |

Details

This function takes the resultlist and transfer the content to a newly organized list. This also consists of more information, which help to generate an output like a netCDF-file.

Value

Returning a savelist with the content of the resultlist.

Author(s)

Andre Duesterhus

See Also

[qat_call_save_lim_rule](#), [qat_run_workflow_save](#)

Examples

```
vec <- rnorm(1000)
min_vector <- seq(-1, -2, length.out=1000)
max_vector <- seq(1, 2, length.out=1000)
result <- list(result=qat_analyse_lim_rule_dynamic_1d(vec, min_vector, max_vector, min_vector_name="minimum vect
savelist <- qat_save_lim_rule_dynamic_1d(result)
```

`qat_save_lim_rule_sigma_1d`*Produce a savelist from a resultlist for a LIM Rule Sigma Test*

Description

This function takes the results, produced by `qat\analyse\lim_rule_sigma_1d` and construct a savelist, which may be used to produce a netCDF output.

Usage

```
qat_save_lim_rule_sigma_1d(resultlist_part, baseunit = "")
```

Arguments

| | |
|------------------------------|---------------------------------------------|
| <code>resultlist_part</code> | A list with the results of the check |
| <code>baseunit</code> | The unit of the original measurement vector |

Details

This function takes the resultlist and transfer the content to a newly organized list. This also consists of more information, which help to generate an output like a netCDF-file.

Value

Returning a savelist with the content of the resultlist.

Author(s)

Andre Duesterhus

See Also

[qat_call_save_lim_rule](#), [qat_run_workflow_save](#)

Examples

```
vec <- rnorm(1000)
result <- list(result=qat_analyse_lim_rule_sigma_1d(vec, 2))
savelist <- qat_save_lim_rule_sigma_1d(result)
```

`qat_save_lim_rule_static_1d`*Produce a savelist from a resultlist for a LIM Rule Static Test*

Description

This function takes the results, produced by `qat\analyse\lim_rule\static_1d` and construct a savelist, which may be used to produce a netCDF output.

Usage

```
qat_save_lim_rule_static_1d(resultlist_part, baseunit = "")
```

Arguments

| | |
|------------------------------|---------------------------------------------|
| <code>resultlist_part</code> | A list with the results of the check |
| <code>baseunit</code> | The unit of the original measurement vector |

Details

This function takes the resultlist and transfer the content to a newly organized list. This also consists of more information, which help to generate an output like a netCDF-file.

Value

Returning a savelist with the content of the resultlist.

Author(s)

Andre Duesterhus

See Also

[qat_call_save_lim_rule](#), [qat_run_workflow_save](#)

Examples

```
vec <- rnorm(1000)
result <- list(result=qat_analyse_lim_rule_static_1d(vec, -2,2))
savelist <- qat_save_lim_rule_static_1d(result)
```

qat_save_noc_rule_1d *Produce a savelist from a resultlist for a NOC Rule Test*

Description

This function takes the results, produced by `qat_analyse_noc_rule_1d` and construct a savelist, which may be used to produce a netCDF output.

Usage

```
qat_save_noc_rule_1d(resultlist_part, baseunit = "")
```

Arguments

| | |
|-----------------|---------------------------------------------|
| resultlist_part | A list with the results of the check |
| baseunit | The unit of the original measurement vector |

Details

This function takes the resultlist and transfer the content to a newly organized list. This also consists of more information, which help to generate an output like a netCDF-file.

Value

Returning a savelist with the content of the resultlist.

Author(s)

Andre Duesterhus

See Also

[qat_call_save_noc_rule](#), [qat_run_workflow_save](#)

Examples

```
vec <- c(1,2,3,4,4,4,5,5,4,3,NaN,3,2,1)
result <- list(result=qat_analyse_noc_rule_1d(vec, 1))
savelist <- qat_save_noc_rule_1d(result)
```

qat_save_result_ncdf *Writing a savelist to a netCDF-file*

Description

A savelist, which is constructed by the function `qat_run_workflow_save` will be written to a given filename in netCDF format. Additional needed informations are the workflowlist, which constructed the savelist.

Usage

```
qat_save_result_ncdf(measurement_vector, savelist, filename, workflowlist = NULL, time = NULL, height = NULL)
```

Arguments

| | |
|--------------------------|--------------------------------------------------------------------------|
| measurement_vector | The measurement vector, which was tested |
| savelist | The resulted savelist |
| filename | The name of the file, which should be written |
| workflowlist | The used workflowlist for the tests |
| time | A vector of time elements with the length of the measurement vector |
| height | A vector of height elements with the length of the measurement vector |
| lat | A vector of latitude elements with the length of the measurement vector |
| lon | A vector of longitude elements with the length of the measurement vector |
| vec1 | An additional vector, which is named as vec1 |
| vec2 | An additional vector, which is named as vec2 |
| vec3 | An additional vector, which is named as vec3 |
| vec4 | An additional vector, which is named as vec4 |
| store_mes_vec | A boolean variable if the measurement vector should also be stored |
| baseunit | Unit of the measurement vector |
| addunits | Vector of units for the other vectors |
| directoryname | Directory, where the resulting file should be stored |
| nan_value | Fill value for NaN in vectors |
| variable_name | Name of the original variable |
| transformationonvariable | Information on transformation of the original variable |
| authorname | Name of the author who performed the tests |
| original_filename | Filename, where the original data was stored |
| data_level | Data level of the original variable |
| workflow_filename | Filename of the workflow |

Details

The savelist, which is a result of the function `qat_run_workflow_save`, which transformed the resultlist of `qat_run_workflow_check` to a here usable format, delivers all necessary information to construct a netCDF-file. The workflowlist is needed, because further informations, like algorithms, descriptions and comments on results are simpler to edit in this list. This can be also saved by `qat_config_write_workflow` to a XML-format. The netCDF-format used here is the QAD-convention. This allows to store the modifications of a tests and also the results into one file.

Value

No return value.

Author(s)

Andre Duesterhus

See Also

[qat_run_workflow_save](#)

Examples

```
library("qat")
# define testvector
testvector<-rnorm(500)
# read in workflow from systemfiles
filename_in <- system.file("extdata/workflowexample.xml", package="qat")
workflowlist <- qat_config_read_workflow(filename_in)
# define some additional vectors
maxlim <- seq(3,1,length.out=500)
minlim <- seq(-1,-3,length.out=500)
uproc <- seq(1,3,length.out=500)
downroc <- seq(3,1,length.out=500)
# run the workflow on the testvector
rlist <- qat_run_workflow_check(testvector,workflowlist,vec1=maxlim, vec2=minlim, vec3=uproc, vec4=downroc)
# produce the savelist
savelist <- qat_run_workflow_save(rlist)
filename_out <- "myresults.nc"
# write netCDF-file of the results in current directory
qat_save_result_ncdf(testvector, savelist=savelist, filename_out, workflowlist=workflowlist,vec1=maxlim, vec2=mi
```

`qat_save_roc_rule_dynamic_1d`

Produce a savelist from a resultlist for a ROC Rule Dynamic Test

Description

This function takes the results, produced by `qat_analyse_roc_rule_dynamic_1d` and construct a savelist, which may be used to produce a netCDF output.

Usage

```
qat_save_roc_rule_dynamic_1d(resultlist_part, baseunit = "")
```

Arguments

| | |
|-----------------|---------------------------------------------|
| resultlist_part | A list with the results of the check |
| baseunit | The unit of the original measurement vector |

Details

This function takes the resultlist and transfer the content to a newly organized list. This also consists of more information, which help to generate an output like a netCDF-file.

Value

Returning a savelist with the content of the resultlist.

Author(s)

Andre Duesterhus

See Also

[qat_call_save_roc_rule](#), [qat_run_workflow_save](#)

Examples

```
vec <- rnorm(100)
min_vector <- seq(1, 2, length.out=1000)
max_vector <- seq(1, 2, length.out=1000)
result <- list(result=qat_analyse_roc_rule_dynamic_1d(vec, min_vector, max_vector, upward_vector_name="upward ve
savelist <- qat_save_roc_rule_dynamic_1d(result)
```

qat_save_roc_rule_static_1d

Produce a savelist from a resultlist for a ROC Rule Static Test

Description

This function takes the results, produced by `qat_analyse_roc_rule_static_1d` and construct a savelist, which may be used to produce a netCDF output.

Usage

```
qat_save_roc_rule_static_1d(resultlist_part, baseunit = "")
```

Arguments

`resultlist_part` A list with the results of the check
`baseunit` The unit of the original measurement vector

Details

This function takes the `resultlist` and transfer the content to a newly organized list. This also consists of more information, which help to generate an output like a netCDF-file.

Value

Returning a `savelist` with the content of the `resultlist`.

Author(s)

Andre Duesterhus

See Also

[qat_call_save_roc_rule](#), [qat_run_workflow_save](#)

Examples

```
vec <- rnorm(100)
result <- list(result=qat_analyse_roc_rule_static_1d(vec, 2,2))
savelist <- qat_save_roc_rule_static_1d(result)
```

`qat_save_set_addup_1d` *Produce a savelist from a resultlist for a Set Addup*

Description

This function takes the results, produced by `qat_analyse_set_addup_1d` and construct a `savelist`, which may be used to produce a netCDF output.

Usage

```
qat_save_set_addup_1d(resultlist_part, baseunit = "")
```

Arguments

`resultlist_part` A list with the results of the check
`baseunit` The unit of the original measurement vector

Details

This function takes the resultlist and transfer the content to a newly organized list. This consists mainly of a text, which is use in the parameter description for a netCDF-file.

Value

Returning a savelist with the content of the resultlist.

Author(s)

Andre Duesterhus

See Also

[qat_call_save_set_addup](#)

Examples

```
## still to come
```

qat_save_set_mean_1d *Produce a savelist from a resultlist for a Set Mean*

Description

This function takes the results, produced by qat\analyse_set_mean_1d and construct a savelist, which may be used to produce a netCDF output.

Usage

```
qat_save_set_mean_1d(resultlist_part, baseunit = "")
```

Arguments

| | |
|-----------------|---------------------------------------------|
| resultlist_part | A list with the results of the check |
| baseunit | The unit of the original measurement vector |

Details

This function takes the resultlist and transfer the content to a newly organized list. This consists mainly of a text, which is use in the parameter description for a netCDF-file.

Value

Returning a savelist with the content of the resultlist.

Author(s)

Andre Duesterhus

See Also

[qat_call_save_set_mean](#)

Examples

```
## still to come
```

```
qat_save_set_nans_1d Produce a savelist from a resultlist for a Set NAN
```

Description

This function takes the results, produced by `qat\analyse\set\nan\1d` and construct a savelist, which may be used to produce a netCDF output.

Usage

```
qat_save_set_nans_1d(resultlist_part, baseunit = "")
```

Arguments

| | |
|-----------------|---------------------------------------------|
| resultlist_part | A list with the results of the check |
| baseunit | The unit of the original measurement vector |

Details

This function takes the resultlist and transfer the content to a newly organized list. This consists mainly of a text, which is use in the parameter description for a netCDF-file.

Value

Returning a savelist with the content of the resultlist.

Author(s)

Andre Duesterhus

See Also

[qat_call_save_set_nans](#)

Examples

```
## still to come
```

`qat_save_set_nans_above_1d`*Produce a savelist from a resultlist for a Set NAN above*

Description

This function takes the results, produced by `qat_analyse_set_nan_above_1d` and construct a savelist, which may be used to produce a netCDF output.

Usage

```
qat_save_set_nans_above_1d(resultlist_part, baseunit = "")
```

Arguments

`resultlist_part`

A list with the results of the check

`baseunit`

The unit of the original measurement vector

Details

This function takes the resultlist and transfer the content to a newly organized list. This consists mainly of a text, which is use in the parameter description for a netCDF-file.

Value

Returning a savelist with the content of the resultlist.

Author(s)

Andre Duesterhus

See Also

[qat_call_save_set_nans](#)

Examples

```
## still to come
```

qat_save_set_nans_below_1d

Produce a savelist from a resultlist for a Set NAN below

Description

This function takes the results, produced by `qat_analyse_set_nan_below_1d` and construct a savelist, which may be used to produce a netCDF output.

Usage

```
qat_save_set_nans_below_1d(resultlist_part, baseunit = "")
```

Arguments

| | |
|------------------------------|---------------------------------------------|
| <code>resultlist_part</code> | A list with the results of the check |
| <code>baseunit</code> | The unit of the original measurement vector |

Details

This function takes the resultlist and transfer the content to a newly organized list. This consists mainly of a text, which is use in the parameter description for a netCDF-file.

Value

Returning a savelist with the content of the resultlist.

Author(s)

Andre Duesterhus

See Also

[qat_call_save_set_nans](#)

Examples

```
## still to come
```

`qat_save_slide_distribution_1d`*Produce a savelist from a resultlist for a Slide Distribution Test*

Description

This function takes the results, produced by `qat\analyse\slide\distribution\1d` and construct a savelist, which may be used to produce a netCDF output.

Usage

```
qat_save_slide_distribution_1d(resultlist_part, baseunit = "")
```

Arguments

| | |
|------------------------------|---------------------------------------------|
| <code>resultlist_part</code> | A list with the results of the check |
| <code>baseunit</code> | The unit of the original measurement vector |

Details

This function takes the resultlist and transfer the content to a newly organized list. This also consists of more information, which help to generate an output like a netCDF-file.

Value

Returning a savelist with the content of the resultlist.

Author(s)

Andre Duesterhus

See Also

[qat_call_save_slide_distribution](#), [qat_run_workflow_save](#)

Examples

```
vec <- rnorm(100)
result <- list(result=qat_analyse_slide_distribution_1d(vec, 10))
savelist <- qat_save_slide_distribution_1d(result)
```

qat_save_trimmed_distribution_1d

Produce a savelist from a resultlist for a Trimmed Distribution Test

Description

This function takes the results, produced by `qat\analyse_trimmed_distribution_1d` and construct a savelist, which may be used to produce a netCDF output.

Usage

```
qat_save_trimmed_distribution_1d(resultlist_part, baseunit = "")
```

Arguments

| | |
|------------------------------|---------------------------------------------|
| <code>resultlist_part</code> | A list with the results of the check |
| <code>baseunit</code> | The unit of the original measurement vector |

Details

This function takes the resultlist and transfer the content to a newly organized list. This also consists of more information, which help to generate an output like a netCDF-file.

Value

Returning a savelist with the content of the resultlist.

Author(s)

Andre Duesterhus

See Also

[qat_call_save_trimmed_distribution](#), [qat_run_workflow_save](#)

Examples

```
vec <- rnorm(1000)
result <- list(result=qat_analyse_trimmed_distribution_1d(vec))
savelist <- qat_save_trimmed_distribution_1d(result)
```

| | |
|----------------|--------------------------------|
| qat_style_plot | <i>Produce a plotstylelist</i> |
|----------------|--------------------------------|

Description

Loads a plotstyle, when a filename is given. When not a standard plotstyle will be given back

Usage

```
qat_style_plot(filename = "")
```

Arguments

| | |
|----------|-----------------------------|
| filename | Filename of a plotstyle-XML |
|----------|-----------------------------|

Details

A plotstyle is a possibility to include a colorscheme in every plot, which is produced by the qat-package. With given filename a certain plotstyle will be loaded. Without a standard sheme will be used.

Value

A list with the inforamtion of the colorscheme.

Author(s)

Andre Duesterhus

See Also

[qat_run_workflow_plot](#)

Examples

```
library("qat")
# define testvector
testvector<-rnorm(500)
# read in workflow from systemfiles
filename_in <- system.file("extdata/workflowexample.xml", package="qat")
workflowlist <- qat_config_read_workflow(filename_in)
# define some additional vectors
maxlim <- seq(3,1,length.out=500)
minlim <- seq(-1,-3,length.out=500)
uproc <- seq(1,3,length.out=500)
downroc <- seq(3,1,length.out=500)
# load plotstyle
filename_ps <- system.file("extdata/plotstyle1.xml", package="qat")
ps<-qat_style_plot(filename_ps)
```

```
# run the workflow on the testvector
rlist <- qat_run_workflow_check(testvector, workflowlist, vec1=maxlim, vec2=minlim, vec3=uproc, vec4=downroc)
# produce some plots of the result in the current directory with new plotstyle
qat_run_workflow_plot(rlist, measurement_name="Test", basename="test", plotstyle=ps)
# add some more informations for the workflow
workflowlist <- qat_add_all_descriptions(workflowlist)
workflowlist <- qat_add_all_algorithms(workflowlist)
workflowlist <- qat_add_comment(workflowlist, 1, "No problems")

filename_out <- "myworkflow_result.xml"
# write edited workflow in current directory
qat_config_write_workflow(workflowlist, output_filename=filename_out)
```

Index

*Topic **IO**

- qat_config_read_workflow, [67](#)
- qat_config_write_workflow, [68](#)
- qat_data_read_ncdf, [72](#)
- qat_read_parameter, [85](#)
- qat_style_plot, [107](#)

*Topic **iplot**

- qat_style_plot, [107](#)

*Topic **manip**

- qat_add_algorithm, [4](#)
- qat_add_all_algorithms, [5](#)
- qat_add_all_descriptions, [6](#)
- qat_add_comment, [7](#)
- qat_add_description, [8](#)
- qat_add_resultfile, [9](#)
- qat_analyse_set_addup_1d, [22](#)
- qat_analyse_set_mean_1d, [23](#)
- qat_analyse_set_nans_1d, [24](#)
- qat_analyse_set_nans_above_1d, [25](#)
- qat_analyse_set_nans_below_1d, [26](#)

*Topic **package**

- qat-package, [3](#)

*Topic **ts**

- qat_analyse_block_distribution_1d, [10](#)
- qat_analyse_boot_distribution_1d, [11](#)
- qat_analyse_distribution_1d, [12](#)
- qat_analyse_lim_rule_dynamic_1d, [14](#)
- qat_analyse_lim_rule_sigma_1d, [15](#)
- qat_analyse_lim_rule_static_1d, [17](#)
- qat_analyse_noc_rule_1d, [18](#)
- qat_analyse_roc_rule_dynamic_1d, [19](#)
- qat_analyse_roc_rule_static_1d, [21](#)
- qat_analyse_set_addup_1d, [22](#)
- qat_analyse_set_mean_1d, [23](#)
- qat_analyse_set_nans_1d, [24](#)

- qat_analyse_set_nans_above_1d, [25](#)
- qat_analyse_set_nans_below_1d, [26](#)
- qat_analyse_slide_distribution_1d, [27](#)
- qat_analyse_trimmed_distribution_1d, [28](#)
- qat_call_set_addup, [61](#)
- qat_call_set_mean, [63](#)
- qat_plot_block_distribution_1d, [73](#)
- qat_plot_boot_distribution_1d, [74](#)
- qat_plot_distribution_1d, [75](#)
- qat_plot_lim_rule_dynamic_1d, [76](#)
- qat_plot_lim_rule_sigma_1d, [77](#)
- qat_plot_lim_rule_static_1d, [79](#)
- qat_plot_noc_rule_1d, [80](#)
- qat_plot_roc_rule_dynamic_1d, [81](#)
- qat_plot_roc_rule_static_1d, [82](#)
- qat_plot_slide_distribution_1d, [83](#)
- qat_plot_trimmed_distribution_1d, [84](#)

*Topic **utilities**

- qat_call_block_distribution, [29](#)
- qat_call_boot_distribution, [30](#)
- qat_call_distribution, [32](#)
- qat_call_lim_rule, [33](#)
- qat_call_noc_rule, [34](#)
- qat_call_plot_block_distribution, [36](#)
- qat_call_plot_boot_distribution, [37](#)
- qat_call_plot_distribution, [38](#)
- qat_call_plot_lim_rule, [39](#)
- qat_call_plot_noc_rule, [41](#)
- qat_call_plot_roc_rule, [42](#)
- qat_call_plot_slide_distribution, [43](#)
- qat_call_plot_trimmed_distribution, [45](#)
- qat_call_roc_rule, [46](#)

- qat_call_save_block_distribution, 47
- qat_call_save_boot_distribution, 49
- qat_call_save_distribution, 50
- qat_call_save_lim_rule, 51
- qat_call_save_noc_rule, 53
- qat_call_save_roc_rule, 54
- qat_call_save_set_addup, 55
- qat_call_save_set_mean, 57
- qat_call_save_set_nans, 58
- qat_call_save_slide_distribution, 59
- qat_call_save_trimmed_distribution, 60
- qat_call_set_nans, 64
- qat_call_slide_distribution, 65
- qat_call_trimmed_distribution, 66
- qat_data_nameofvars_ncdf, 70
- qat_data_numofvars_ncdf, 71
- qat_data_varcontent_ncdf, 72
- qat_read_parameter, 85
- qat_run_workflow_check, 86
- qat_run_workflow_plot, 87
- qat_run_workflow_save, 89
- qat_save_block_distribution_1d, 90
- qat_save_boot_distribution_1d, 91
- qat_save_distribution_1d, 92
- qat_save_lim_rule_dynamic_1d, 93
- qat_save_lim_rule_sigma_1d, 94
- qat_save_lim_rule_static_1d, 95
- qat_save_noc_rule_1d, 96
- qat_save_result_ncdf, 97
- qat_save_roc_rule_dynamic_1d, 98
- qat_save_roc_rule_static_1d, 99
- qat_save_set_addup_1d, 100
- qat_save_set_mean_1d, 101
- qat_save_set_nans_1d, 102
- qat_save_set_nans_above_1d, 103
- qat_save_set_nans_below_1d, 104
- qat_save_slide_distribution_1d, 105
- qat_save_trimmed_distribution_1d, 106
- qat (qat-package), 3
- qat-package, 3
- qat_add_algorithm, 4
- qat_add_all_algorithms, 5
- qat_add_all_descriptions, 6
- qat_add_comment, 7
- qat_add_description, 8
- qat_add_resultfile, 9
- qat_analyse_block_distribution_1d, 10, 74
- qat_analyse_boot_distribution_1d, 11, 31, 75
- qat_analyse_distribution_1d, 12, 33, 76
- qat_analyse_lim_rule_dynamic (qat_analyse_lim_rule_dynamic_1d), 14
- qat_analyse_lim_rule_dynamic_1d, 14, 16, 18, 34, 77
- qat_analyse_lim_rule_sigma_1d, 15, 15, 18, 34
- qat_analyse_lim_rule_static_1d, 15, 16, 17, 34, 78, 79
- qat_analyse_noc_rule_1d, 18, 81
- qat_analyse_roc_rule_dynamic_1d, 19, 22, 47, 82
- qat_analyse_roc_rule_static_1d, 20, 21, 47, 83
- qat_analyse_set_addup_1d, 22, 24–26, 62
- qat_analyse_set_mean_1d, 22, 23, 24–26, 64
- qat_analyse_set_nans_1d, 22, 24, 25, 26, 65
- qat_analyse_set_nans_above_1d, 25, 26
- qat_analyse_set_nans_below_1d, 25, 26
- qat_analyse_slide_distribution_1d, 27, 30, 66, 84
- qat_analyse_trimmed_distribution_1d, 28, 67, 85
- qat_call_block_distribution, 29
- qat_call_boot_distribution, 30
- qat_call_distribution, 32
- qat_call_lim_rule, 15, 16, 18, 33
- qat_call_noc_rule, 19, 34
- qat_call_plot_block_distribution, 36
- qat_call_plot_boot_distribution, 37
- qat_call_plot_distribution, 38
- qat_call_plot_lim_rule, 39
- qat_call_plot_noc_rule, 41
- qat_call_plot_roc_rule, 42
- qat_call_plot_slide_distribution, 43
- qat_call_plot_trimmed_distribution, 45
- qat_call_roc_rule, 20, 22, 46

- qat_call_save_block_distribution, [47](#),
[91](#)
- qat_call_save_boot_distribution, [49](#), [92](#)
- qat_call_save_distribution, [50](#), [92](#)
- qat_call_save_lim_rule, [51](#), [93–95](#)
- qat_call_save_noc_rule, [53](#), [96](#)
- qat_call_save_roc_rule, [54](#), [99](#), [100](#)
- qat_call_save_set_addup, [55](#), [101](#)
- qat_call_save_set_mean, [57](#), [102](#)
- qat_call_save_set_nans, [58](#), [102–104](#)
- qat_call_save_slide_distribution, [59](#),
[105](#)
- qat_call_save_trimmed_distribution, [60](#),
[106](#)
- qat_call_set_addup, [61](#)
- qat_call_set_mean, [63](#)
- qat_call_set_nans, [64](#)
- qat_call_slide_distribution, [65](#)
- qat_call_trimmed_distribution, [66](#)
- qat_config_read_workflow, [5–9](#), [67](#), [69](#), [87](#),
[90](#)
- qat_config_write_workflow, [68](#)
- qat_data_nameofvars_ncdf, [70](#), [71–73](#)
- qat_data_numofvars_ncdf, [70](#), [71](#), [72](#), [73](#)
- qat_data_read_ncdf, [70](#), [71](#), [72](#), [73](#)
- qat_data_varcontent_ncdf, [70](#), [71](#), [72](#), [72](#)
- qat_plot_block_distribution_1d, [11](#), [37](#),
[73](#)
- qat_plot_boot_distribution_1d, [12](#), [38](#),
[74](#)
- qat_plot_distribution_1d, [13](#), [39](#), [75](#)
- qat_plot_lim_rule_dynamic_1d, [15](#), [34](#), [40](#),
[76](#), [78](#), [79](#)
- qat_plot_lim_rule_sigma_1d, [16](#), [34](#), [40](#),
[77](#), [77–79](#)
- qat_plot_lim_rule_static_1d, [18](#), [34](#), [40](#),
[77](#), [79](#)
- qat_plot_noc_rule_1d, [19](#), [35](#), [42](#), [80](#)
- qat_plot_roc_rule_dynamic_1d, [20](#), [43](#), [47](#),
[81](#), [83](#)
- qat_plot_roc_rule_static_1d, [22](#), [43](#), [47](#),
[82](#), [82](#)
- qat_plot_slide_distribution_1d, [28](#), [44](#),
[83](#)
- qat_plot_trimmed_distribution_1d, [29](#),
[46](#), [84](#)
- qat_read_parameter, [85](#)
- qat_run_workflow_check, [68](#), [86](#), [88](#), [90](#)
- qat_run_workflow_plot, [87](#), [87](#), [90](#), [107](#)
- qat_run_workflow_save, [48](#), [50–52](#), [54–56](#),
[58–61](#), [89](#), [91–96](#), [98–100](#), [105](#), [106](#)
- qat_save_block_distribution_1d, [48](#), [90](#)
- qat_save_boot_distribution_1d, [50](#), [91](#)
- qat_save_distribution_1d, [51](#), [92](#)
- qat_save_lim_rule_dynamic_1d, [52](#), [93](#)
- qat_save_lim_rule_sigma_1d, [52](#), [94](#)
- qat_save_lim_rule_static_1d, [52](#), [95](#)
- qat_save_noc_rule_1d, [54](#), [96](#)
- qat_save_result_ncdf, [97](#)
- qat_save_roc_rule_dynamic_1d, [55](#), [98](#)
- qat_save_roc_rule_static_1d, [55](#), [99](#)
- qat_save_set_addup_1d, [56](#), [100](#)
- qat_save_set_mean_1d, [58](#), [101](#)
- qat_save_set_nans_1d, [59](#), [102](#)
- qat_save_set_nans_above_1d, [59](#), [103](#)
- qat_save_set_nans_below_1d, [59](#), [104](#)
- qat_save_slide_distribution_1d, [60](#), [105](#)
- qat_save_trimmed_distribution_1d, [61](#),
[106](#)
- qat_style_plot, [107](#)