Package ‘StreamNet’

July 17, 2022

Title Automatic Integration of Stream Water Quality Data with Stream Network to Monitor Stream Health

Version 0.0.1

Description Functions in this package enable automatic integration of water quality data and stream flowlines to monitor the stream water quality. It allows the user to analyze stream networks to detect statistically significant changes in stream water quality caused by anthropogenic activities. This package is optimized using RCpp to facilitate the high-throughput monitoring of stream water quality.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.0

LinkingTo Rcpp

Imports data.table, doParallel, dplyr, forcats, foreach, geosphere, GGally, ggplot2, igraph, intergraph, iterators, mapdata, maps, MASS, network, parallel, purrr, Rcpp, readr, rgdal, sna, sp, statnet.common, stats, stringr, tibble, tidyr, utils

NeedsCompilation yes

URL `https://github.com/HANDS-Research-Group/StreamNet/tree/main/Rpackage_StreamNet`

BugReports `https://github.com/HANDS-Research-Group/StreamNet/issues`

Author Rohit Patil [aut],
Zhong Zheng [aut],
Amal Agarwal [aut],
Lingzhou Xue [aut],
Tao Wen [aut, cre]

Maintainer Tao Wen <`twen08@syr.edu`>

ExperimentalWindowsRuntime ucrt

Archs x64

R topics documented:

add_new_analyte *Add new water chemistry data at new sampling locations*

Description

This function can be used to add new water chemistry data at new sampling locations and run the statistical inference for all the provided stream location. This function can only be used after “whole_pipe” is executed. Execution of this function will overwrite the previously generated output. Therefore, it is advised to preserve a copy of the old outputs in case you need it.

Usage

```
add_new_analyte(
  path_total,
  df_analyte_to_append_filepath,
  upstream_thresh = 5,
  downstream_lower_thresh = 0,
  downstream_upper_thresh = 50,
  num_cores = 1,
  permission = "no"
)
```

Arguments

`path_total` This is the same root file path as that of the function “whole_pipe”. The previous output will be overwritten so backup is recommended.

`df_analyte_to_append_filepath` This is the full path to the csv file containing new water chemistry data for the given stream. The file formatting requirement is the same as that of “filename_chem” in “whole_pipe”.

`upstream_thresh` The same parameter as that in "whole_pipe".

`downstream_lower_thresh` The same parameter as that in "whole_pipe".

`downstream_upper_thresh` The same parameter as that in "whole_pipe".

`num_cores` The same parameter as that in "whole_pipe".

`permission` The same parameter as that in "whole_pipe".

Value

The same return files as those in "whole_pipe".

Note

This function assumes “whole_pipe” function has already been run and the outputs generated from it lie in “path_total” folder. Please refer to our website for examples.

`add_new_analyte_polluter`

Add new water chemistry data and polluting events at new locations

---

Description

This function can be used to add new water chemistry data and polluting events at new locations and run the statistical inference for all the polluting events. This function can only be used after “whole_pipe” is executed. Execution of this function will overwrite the previously generated output. Therefore, it is advised to preserve a copy of the old outputs in case you need it.

Usage

```
add_new_analyte_polluter(
  path_total,
  df_analyte_to_append_filepath,
  df_polluter_to_append_file_path,
  upstream_thresh = 5,
  downstream_lower_thresh = 0,
  downstream_upper_thresh = 50,
  num_cores = 1,
  permission = "no"
)
```

Arguments

`path_total` This is the same root file path as that of the function “whole_pipe”. The previous output will be overwritten so backup is recommended.

`df_analyte_to_append_filepath` This is the full path to the csv file containing new water chemistry data for the given stream. The file formatting requirement is the same as that of “filename_chem” in “whole_pipe”.

`df_polluter_to_append_file_path` This is the full path to the csv file containing new pollution site data for the given stream. The file formatting requirement is the same as that of “filename_poll” in “whole_pipe”.

`upstream_thresh` The same parameter as that in "whole_pipe".

`downstream_lower_thresh` The same parameter as that in "whole_pipe".

`downstream_upper_thresh` The same parameter as that in "whole_pipe".

`num_cores` The same parameter as that in "whole_pipe".

`permission` The same parameter as that in "whole_pipe".

Value

The same return files as those in "whole_pipe".

Note

This function assumes “whole_pipe” function has already been run and the outputs generated from it lie in “path_total” folder. Please refer to our website for examples.

---

add_new_polluter *Add polluting events at new locations*

Description

This function can be used to add polluting events at new locations and run the statistical inference for this new locations. This function can only be used after “whole_pipe” is executed. Execution of this function will overwrite the previously generated output. Therefore, it is advised to preserve a copy of the old outputs in case you need it.

Usage

```
add_new_polluter(
  path_total,
  df_polluter_to_append_file_path,
  upstream_thresh = 5,
  downstream_lower_thresh = 0,
  downstream_upper_thresh = 50,
  num_cores = 1,
  permission = "no"
)
```

Arguments

`path_total` This is the same root file path as that of the function “whole_pipe”. The previous output will be overwritten so backup is recommended.

`df_polluter_to_append_file_path` This is the full path to the csv file containing new pollution site data for the given stream. The file formatting requirement is the same as that of “filename_poll” in “whole_pipe”.

`upstream_thresh` The same parameter as that in "whole_pipe".

`downstream_lower_thresh` The same parameter as that in "whole_pipe".

`downstream_upper_thresh` The same parameter as that in "whole_pipe".

`num_cores` The same parameter as that in "whole_pipe".

`permission` The same parameter as that in "whole_pipe".

Value

The same return files as those in "whole_pipe".

Note

This function assumes “whole_pipe” function has already been run and the outputs generated from it lie in “path_total” folder. Please refer to our website for examples.

add_temporal_analyte *Add new water chemistry data at existing locations*

Description

This function can be used to add a new water chemistry data at an existing location and run the statistical inference for all the potential polluting sites. This function can only be used after “whole_pipe” is executed. Execution of this function will overwrite the previously generated output. Therefore, it is advised to preserve a copy of the old outputs in case you need it.

Usage

```
add_temporal_analyte(
  path_total,
  df_analyte_to_append_filepath,
  upstream_thresh = 5,
  downstream_lower_thresh = 0,
  downstream_upper_thresh = 50,
  num_cores = 1,
  permission = "no"
)
```

Arguments

`path_total`	This is the same root file path as that of the function “whole_pipe”. The previous output will be overwritten so backup is recommended.
`df_analyte_to_append_filepath`	This is the full path to the csv file containing new water chemistry data for the given stream. The file formatting requirement is the same as that of “filename_chem” in “whole_pipe”.
`upstream_thresh`	The same parameter as that in "whole_pipe".
`downstream_lower_thresh`	The same parameter as that in "whole_pipe".
`downstream_upper_thresh`	The same parameter as that in "whole_pipe".
`num_cores`	The same parameter as that in "whole_pipe".
`permission`	The same parameter as that in "whole_pipe".

Value

The same return files as those in "whole_pipe".

Note

This function assumes “whole_pipe” function has already been run and the outputs generated from it lie in “path_total” folder. Please refer to our website for examples.

add_temporal_analyte_polluter

Add new water chemistry data and polluting events at existing locations

Description

This function can be used to add new water chemistry data and polluting events at existing locations and run the statistical inference for all the polluting events. This function can only be used after “whole_pipe” is executed. Execution of this function will overwrite the previously generated output. Therefore, it is advised to preserve a copy of the old outputs in case you need it.

Usage

```
add_temporal_analyte_polluter(
  path_total,
  df_analyte_to_append_filepath,
  df_polluter_to_append_file_path,
  upstream_thresh = 5,
  downstream_lower_thresh = 0,
  downstream_upper_thresh = 50,
  num_cores = 1,
  permission = "no"
)
```

Arguments

`path_total` This is the same root file path as that of the function “whole_pipe”. The previous output will be overwritten so backup is recommended.

`df_analyte_to_append_filepath` This is the full path to the csv file containing new water chemistry data for the given stream. The file formatting requirement is the same as that of “filename_chem” in “whole_pipe”.

`df_polluter_to_append_file_path` This is the full path to the csv file containing new pollution site data for the given stream. The file formatting requirement is the same as that of “filename_poll” in “whole_pipe”.

`upstream_thresh` The same parameter as that in "whole_pipe".

`downstream_lower_thresh` The same parameter as that in "whole_pipe".

`downstream_upper_thresh` The same parameter as that in "whole_pipe".

`num_cores` The same parameter as that in "whole_pipe".

`permission` The same parameter as that in "whole_pipe".

Value

The same return files as those in "whole_pipe".

Note

This function assumes “whole_pipe” function has already been run and the outputs generated from it lie in “path_total” folder. Please refer to our website for examples.

add_temporal_polluter *Add polluting events at existing locations*

Description

This function can be used to add polluting events at existing locations and run the statistical inference for this event. This function can only be used after “whole_pipe” is executed. Execution of this function will overwrite the previously generated output. Therefore, it is advised to preserve a copy of the old outputs in case you need it.

Usage

```
add_temporal_polluter(
  path_total,
  df_polluter_to_append_file_path,
  upstream_thresh = 5,
  downstream_lower_thresh = 0,
  downstream_upper_thresh = 50,
  num_cores = 1,
  permission = "no"
)
```

Arguments

`path_total`	This is the same root file path as that of the function “whole_pipe”. The previous output will be overwritten so backup is recommended.
`df_polluter_to_append_file_path`	This is the full path to the csv file containing new pollution site data for the given stream. The file formatting requirement is the same as that of “filename_poll” in “whole_pipe”.
`upstream_thresh`	The same parameter as that in "whole_pipe".
`downstream_lower_thresh`	The same parameter as that in "whole_pipe".
`downstream_upper_thresh`	The same parameter as that in "whole_pipe".
`num_cores`	The same parameter as that in "whole_pipe".
`permission`	The same parameter as that in "whole_pipe".

Value

The same return files as those in "whole_pipe".

Note

This function assumes “whole_pipe” function has already been run and the outputs generated from it lie in “path_total” folder. Please refer to our website for examples.

`whole_pipe` *The Function for the whole pipeline.*

Description

This function generates statistical inference of the stream chemistry change between upstream and downstream of all stream locations provided by the user and gives a verdict if the stream is polluted at each location.

Usage

```
whole_pipe(
  path_total,
  filename_chem = "Water_Chemistry_R_Package.csv",
  filename_poll = "Pollution_Site_R_Package.csv",
  upstream_thresh = 5,
  downstream_lower_thresh = 0,
  downstream_upper_thresh = 50,
  num_cores = 1,
  permission = "no"
)
```

Arguments

`path_total` This is the root file path where the input data is stored and the output would be generated along with intermediate files. Different levels of folders should be separated by "/" and the path should be ended with a "/". This path should contain a folder named "data", which contains files for water chemistry and the polluters and a folder named "Flowline_R_Package" that contains the water flowline files.

`filename_chem` This is the filename of the csv file containing water chemistry data for the given stream. This file should locate at the “data” folder in the directory of “path_total”. The user provided input should be a string of the filename containing “.csv” instead of the full path. The file should contain four columns which are in the order of “Chemical”,” Longitude”,” Latitude” and “Date”. Columns names should be provided but does not need to exactly match the provided names.

`filename_poll` This is the filename of the csv file containing the polluting event location information. This file should locate at the “data” folder in the directory of “path_total”. The user provided input should be a string of the filename containing “.csv” instead of the full path. The file should contain four columns which are in the order of “ID”,” Longitude”,” Latitude” and “Date”. Columns names should be provided but does not need to exactly match the provided names.

`upstream_thresh`
Upstream threshold distance in kilometers between provided stream location and upstream samples, within which upstream water samples to be used in the statistical analysis will be defined.

`downstream_lower_thresh`
Lower limit for downstream threshold distance in kilometers between provided stream location and downstream water samples, above which downstream water samples to be used in the statistical analysis will be considered.

`downstream_upper_thresh`
: Upper limit for downstream threshold distance in kilometers between provided stream location and downstream water samples, below which downstream water samples to be used in the statistical analysis will be considered.

`num_cores`
: Maximum number of cores to be used to run certain functions in parallel.

`permission`
: This is a flag about the permission for the function to get the “write” permission for the directory “path_total”. In detail, this function will create sub-directories, write intermediate and final output files in this directory and sub-directory. Any file with the same filename will be overwritten in this process. So, before executing this function, please consider making backups for the directory “path_total” if necessary. Users should set this input to the string “yes” to confirm that they know what will happen and grant the write permission.

Value

The outputs are generated as RData files in the “inference” folder in “path_total”. Their names are begun with ‘df_polluter_test_mean_’ and ‘df_polluter_test_median_’ and end with the parameter “downstream_upper_thresh”. The files contain the statistical test results of the stream chemistry upstream and downstream of all the provided stream locations and a verdict whether the stream is polluted or not at each provided stream location.

Note

Please refer to our website for examples.

Examples

```
#path_total = "test_envir_whole_pipe/"
#filename_chem = "Water_Chemistry_R_Package.csv"
#filename_poll = "Pollution_Site_R_Package.csv"
#num_cores = 26
#permission = "yes"
#try(whole_pipe(path_total, filename_chem, filename_poll, num_cores, permission))
```

Index