Package 'PressPurt'

October 20, 2020

Title Indeterminacy of Networks via Press Perturbations

Version 1.0.2

Description This is a computational package designed to identify the most sensitive interactions within a network which must be estimated most accurately in order to produce qualitatively robust predictions to a press perturbation. This is accomplished by enumerating the number of sign switches (and their magnitude) in the net effects matrix when an edge experiences uncertainty. The package produces data and visualizations when uncertainty is associated to one or more edges in the network and according to a variety of distributions. The software requires the network to be described by a system of differential equations but only requires as input a numerical Jacobian matrix evaluated at an equilibrium point. This package is based on Koslicki, D., & Novak, M. (2017) <doi:10.1007/s00285-017-1163-0>.

URL https://github.com/dkoslicki/PressPurt

BugReports https://github.com/dkoslicki/PressPurt/issues

Depends R (>= 3.1.0)

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.1.0

Imports data.table, ggplot2, grid, gridExtra, reticulate (>= 1.11), utils

Suggests knitr, rmarkdown, R.rsp, qpdf

VignetteBuilder R.rsp

NeedsCompilation no

Author David Koslicki [aut, cre], Dana Gibbon [aut, trl], Mark Novak [aut]

Maintainer David Koslicki <dmk333@psu.edu>

Repository CRAN

Date/Publication 2020-10-19 22:10:13 UTC

R topics documented:

`ComputeEntryWisePerturbationExpectation`
Compute Entry Wise Perturbation Expectation

Description

This function computes the expected number of sign switches from perturbing each entry individually. Run after PreprocessMatrix().

Usage

```
ComputeEntryWisePerturbationExpectation(
  input_folder = NULL,
  PreProsMatrix = NULL,
  prefix = NULL,
  distribution_type = "truncnorm",
  input_a = 0,
  input_b = -2,
  threads = 1
)
```

Arguments

`input_folder`	Input folder. The location of the files created by PreprocessMatrix if you specified an output_folder. If this option is specified, this is also where the num switch array will be saved. Must specify an input_folder OR PreProsMatrix. Default: NULL
`PreProsMatrix`	Object where the PreprocessMatrix output was saved. Must specify an input_folder OR PreProsMatrix. Default: NULL
`prefix`	Prefix of output files, if you so choose.

distribution_type
: Kind of distribution to use. Valid choices are: truncnorm, uniform, trunc_lognorm, beta. Default: "truncnorm"

input_a
: First parameter to the distribution you choose. For truncnorm, this is the mean. Default: 0

input_b
: First parameter to the distribution you choose. For truncnorm, this is the variance. Using a negative value indicates you want the standard deviation to be the length of the interval divided by the absolute value of the input parameter. Default: -2

threads
: Number of threads to use. Default: 1

Value

If an input folder is specified the objects will be saved to that folder. If the PreProsMatrix object is specified, an R list object with the following: original_matrix, matrix_size, column_names, row_names, non_zero, num_switch_functions, asymptotic_stability_start, asymptotic_stability_end, num_switch_funcs_r, distributions, expected_num_switch, distributions_object

Examples

```
## Not run:
# Set input file
infile <- system.file("extdata", "Modules", "IGP.csv",
    package = "PressPurt")
# Preprocess the matrix
PreProsMatrix <- PreprocessMatrix(input_file = infile,
    output_folder = NULL, max_bound = 10, threads = 2)
# Run ComputeEntryWisePerturbationExpectation
Entrywise <- ComputeEntryWisePerturbationExpectation(PreProsMatrix = PreProsMatrix,
    distribution_type = "truncnorm",
    input_a = 0, input_b = -2, threads = 1)

## End(Not run)
```

ComputeMultiEntryPerturbationExpectation

Compute Multi Entry Perturbation Expectation

Description

This function takes a jacobian matrix and computes the multi-entry perturbation expectation.

Usage

```
ComputeMultiEntryPerturbationExpectation(
  input_file,
  num_iterates = 1000,
  interval_length = 0.01,
```

```
  threads = 1
)
```

Arguments

input_file	Input comma separated file for the jacobian matrix.
num_iterates	Number of iterates in the Monte Carlo sampling to perform. Default: 10000
interval_length	Interval length over which to make the perturbations. Default: 0.01
threads	Number of threads to use. Default: 1

Value

returns a scalar

Examples

```
## Not run:
infile <- system.file("extdata", "Modules", "IGP.csv",
    package = "PressPurt")
ComputeMultiEntryPerturbationExpectation(input_file = infile)

## End(Not run)
```

---

create_conda_env *Make a new conda environment*

Description

This function creates a new conda environment and initializes the new conda environment. In doing so, this function sets your python version and one may specify a specific python version. This is useful if you have multiple versions of python installed. When making a new conda environment, if the python version isn't set, then your default one will be used.

Usage

```
create_conda_env(condaenv, version = NULL, verbose = TRUE)
```

Arguments

condaenv	Specify conda environment name
version	Set path to specific version of python.
verbose	TRUE or FALSE. When TRUE, shows python and conda configuration. Default: TRUE

Value

None

Examples

```
## Not run:
create_conda_env(
    condaenv = "r-reticulate",
    version = "~/anaconda3/bin/python",
    verbose = TRUE)

## End(Not run)
```

`create_virtual_env` *Make a new virtual environment*

Description

This function creates a new virtual environment and initializes the new virtual environment. In doing so, this function sets your python version and one may specify a specific python version. This is useful if you have multiple versions of python installed. When making a new virtual environment, if the python version isn't set, then your default one will be used.

Usage

```
create_virtual_env(virtualenv, version = NULL, verbose = TRUE)
```

Arguments

`virtualenv`	Specify conda environment name
`version`	Set path to specific version of python.
`verbose`	TRUE or FALSE. When TRUE, shows python and conda configuration. Default: TRUE

Value

None

Examples

```
## Not run:
create_virtual_env(version = "/usr/bin/python3",
    virtualenv = "r-reticulate",
    verbose = TRUE)

## End(Not run)
```

find_python *Find Python versions, Conda, & Virtual Environments*

Description

This function lists available python versions, conda environments, and virtual environments. One may show all three or just one.

Usage

```
find_python(python = TRUE, conda = TRUE, virtualenv = TRUE)
```

Arguments

python	If TRUE will list available python versions. Default: TRUE
conda	If TRUE will list available conda environments. Default: TRUE
virtualenv	If TRUE will list available virtual environments. Default: TRUE

Value

None

Examples

```
## Not run:
find_python()

## End(Not run)
```

GenerateEntryWiseFigures *Generate Entry Wise Figures*

Description

This function plots the number of mis-predictions versus perturbation value, overlaid with distribution over stable perturbation values. Run after ComputeEntryWisePerturbationExpectation()

Usage

```
GenerateEntryWiseFigures(
  input_folder = NULL,
  EntryWise = NULL,
  prefix = NULL,
  all_numswitch_plots = FALSE,
  list_of_numswitch_to_plot = NULL
)
```

Arguments

input_folder	Input folder. The location of the files created by PreprocessMatrix if you specified an output_folder. This is also where the num switch array was saved. Must specify an input_folder OR EntryWise object. Default: NULL
EntryWise	Object where the ComputeEntryWisePerturbationExpectation output was saved.
prefix	Prefix of output files, if you so choose.
all_numswitch_plots	set to TRUE if you ant to plot all num switch plots (potentially very large). Default: FALSE
list_of_numswitch_to_plot	List of entries you want visualized with num switch. Should be a list of vectors. Example: list(c(0, 0), c(0, 1))

Value

plot or plots

Examples

```
## Not run:
# Set input file
infile <- system.file("extdata", "Modules", "IGP.csv",
    package = "PressPurt")
# Preprocess the matrix
PreProsMatrix <- PreprocessMatrix(input_file = infile,
    output_folder = NULL, max_bound = 10, threads = 2)

# Run ComputeEntryWisePerturbationExpectation
Entrywise <- ComputeEntryWisePerturbationExpectation(PreProsMatrix = PreProsMatrix,
    distribution_type = "truncnorm",
    input_a = 0, input_b = -2, threads = 1)

# Plot specific entries using entrywise object
list_of_numswitch_to_plot <- list(c(1, 1), c(1, 2))
GenerateEntryWiseFigures(EntryWise=Entrywise,
    all_numswitch_plots = FALSE,
    list_of_numswitch_to_plot=list_of_numswitch_to_plot)


# Plot specific entries from folder
GenerateEntryWiseFigures(input_folder = "test_r/test3",
    all_numswitch_plots = FALSE,
    list_of_numswitch_to_plot=list_of_numswitch_to_plot)

# Plot all numswitch plots
GenerateEntryWiseFigures(EntryWise=Entrywise,
    all_numswitch_plots = TRUE)

## End(Not run)
```

`get_distributions_single` *Get PDF distribution*

Description

This function retrieves the PDF (Probability Distribution Function) object from the scipy method <scipy.stats._distn_infrastructure.rv_frozen>.

Usage

```
get_distributions_single(
  matrix_entry,
  distribution_list,
  asymp_stab,
  points = 250
)
```

Arguments

`matrix_entry`	Position in the matrix. Example: c(1, 1)
`distribution_list`	list of scipy distributions
`asymp_stab`	asymptotic stability interval
`points`	the number of values in x range

Value

Probability Distribution Function from scipy

Examples

```
## Not run:
k <- 1
l <- 1
np <- reticulate::import("numpy")
distributions <- reticulate::py_load_object("distributions.pkl")
single_dist <- get_distributions_single(matrix_entry = c(k,l),
    distribution_list = distributions,
    asymp_stab = c(combined$asymptotic_stability_start[k,l],
    combined$asymptotic_stability_end[k,l]))

## End(Not run)
```

ns_to_step *Num Switch Function to step function*

Description

This function transforms a Num Switch Function to a plot ready step function with x and y values. Returns a data frame of x and y values to plot.

Usage

```
ns_to_step(asymp_stab_start, asymp_stab_end, num_switch_func)
```

Arguments

asymp_stab_start	start interval from asymptotic_stability
asymp_stab_end	end interval from asymptotic_stability
num_switch_func	a single num switch function

Value

plot ready x and y values from the Num Switch Function

Examples

```
## Not run:
# Set input file
infile <- system.file("extdata", "Modules", "IGP.csv",
    package = "PressPurt")
# Preprocess the matrix
PreProsMatrix <- PreprocessMatrix(input_file = infile,
    output_folder = NULL, max_bound = 10, threads = 2)

# Run ComputeEntryWisePerturbationExpectation
Entrywise <- ComputeEntryWisePerturbationExpectation(
    PreProsMatrix = PreProsMatrix,
    distribution_type = "truncnorm",
    input_a = 0, input_b = -2, threads = 1)

ns_step <- ns_to_step(
    asymp_stab_start = Entrywise$asymptotic_stability_start[1,1],
    asymp_stab_end = Entrywise$asymptotic_stability_end[1,1],
    num_switch_func = Entrywise$num_switch_funcs_r$`(1, 1)`)

## End(Not run)
```

PreprocessMatrix *Preprocess Matrix*

Description

This script pre-processes a matrix by figuring out what the intervals of asymptotic stability are, as well as finding which perturbation values lead to a sign switch.

Usage

```
PreprocessMatrix(
  input_file,
  output_folder = NULL,
  prefix = NULL,
  max_bound = 10,
  zero_perturb = FALSE,
  threads = 1,
  verbose = FALSE
)
```

Arguments

input_file	Input comma separated file for the jacobian matrix.
output_folder	Optional output folder to save python objects to disk. A number of files will be created in the form 'output_folder/<prefix>_*.npy'. Default is NULL.
prefix	Prefix of output files, if you so choose.
max_bound	some of the matrices are unbounded stable towards one end, this is the limit the user imposes. Default: 10
zero_perturb	Flag to indicate you want to perturb the zero entries. Default: FALSE
threads	Number of threads to use. Default: 1
verbose	Default: FALSE

Value

A list of with the following objects: matrix_size, column_names, row_names, non_zero, num_switch_functions, asymptotic_stability_start, asymptotic_stability_end, num_switch_funcs_r

Examples

```
## Not run:
infile <- system.file("extdata", "Modules", "IGP.csv",
    package = "PressPurt")
PreProsMatrix <- PreprocessMatrix(input_file = infile,
    output_folder = NULL, max_bound = 10, threads = 2)

## End(Not run)
```

process_data *Convert data to R format if saved to disk*

Description

This function will convert objects saved to disk to R friendly objects, or the same output as ComputeEntryWisePerturbationExpectation. If you used the "save to disk" option or ran via python directly, run this function to read the data into R. Files read in: asymptotic_stability.npy, column_names.txt, distributions.pkl, expected_num_switch.csv, num_non_zero.npy, num_switch_funcs.pkl, row_names.txt and size.npy. Note how most of these objects are python based objects- numpy or pickle objects.

Usage

```
process_data(matrix, type = "csv", folder, prefix = NULL)
```

Arguments

matrix	path to the original matrix.
type	csv or tab. Is the original matrix comma separated or tab separated? Default: csv
folder	path to the folder where output data was saved.
prefix	optional prefix to file names

Value

object formatted in the same way the output of ComputeEntryWisePerturbationExpectation

Examples

```
## Not run:
infile <- system.file("extdata", "Modules", "IGP.csv",
    package = "PressPurt")
data <- process_data(matrix = infile,
    type = "csv", folder = "output")

## End(Not run)
```

py_depend *Install Python Dependencies*

Description

This function installs needed python libraries into the specified conda environment OR virtual environment. Should be the same as the one specified in set_python. Required python libraries: matplotlib, numpy, pandas, pathos, scipy and sympy On CentOS 7 pandas & scipy may need to be installed with pip install from the command line. Will get the error: /lib/libstdc++.so.6: version 'CXXABI_1.3.9' not found See vignette for more information.

Usage

```
py_depend(condaenv = NULL, virtualenv = NULL)
```

Arguments

condaenv	Name of conda environment to install python libraries to. Default: NULL
virtualenv	Name of virtual environment to install python libraries to. Default: NULL

Value

None

Examples

```
## Not run:
# Cond env
py_depend(condaenv = "r-reticulate",
    virtualenv = NULL)
# virtualenv:
py_depend(virtualenv = "r-reticulate",
    condaenv = NULL)

## End(Not run)
```

set_python_conda *Set Python Conda environment*

Description

This function sets your conda environment. Run this command before PreprocessMatrix. Install python dependencies in the same conda environment that you set here. To make a new conda environment use the create_conda_env function.

Usage

```
set_python_conda(condaenv, verbose = TRUE)
```

Arguments

condaenv	Specify conda environment name
verbose	TRUE or FALSE. When TRUE, shows python and conda configuration. Default: TRUE

Value

None

Examples

```
## Not run:
set_python_conda(
    condaenv = "r-reticulate",
    verbose = TRUE)

## End(Not run)
```

---

set_python_virtual *Set your Python Virtual environment*

Description

This function sets your virtual environment. Run this command before PreprocessMatrix. Install python dependencies in the same virtual environment that you set here. To make a new virtual environment use the create_virtual_env function.

Usage

```
set_python_virtual(virtualenv, verbose = TRUE)
```

Arguments

virtualenv	Specify virtual environment name
verbose	TRUE or FALSE. When TRUE, shows python and virtual environment configuration. Default: TRUE

Value

None

Examples

```
## Not run:
set_python_virtual(
    virtualenv = "r-reticulate",
    verbose = TRUE)

## End(Not run)
```

Index