Package: rosettaPTF (via r-universe)

October 25, 2024

Title R Frontend for Rosetta Pedotransfer Functions

Version 0.1.5

Author Soil and Plant Science Division Staff

Maintainer Andrew G. Brown <andrew.g.brown@usda.gov>

Description Access Python rosetta-soil pedotransfer functions in an R environment. Rosetta is a neural network-based model for predicting unsaturated soil hydraulic parameters from basic soil characterization data. The model predicts parameters for the van Genuchten unsaturated soil hydraulic properties model, using sand, silt, and clay, bulk density and water content. The codebase is now maintained by Dr. Todd Skaggs and other U.S. Department of Agriculture employees. This R package is intended to provide for use cases that involve many thousands of calls to the pedotransfer function. Less demanding use cases are encouraged to use the web interface or API endpoint. There are additional wrappers of the API endpoints provided by the soilDB R package `ROSETTA()` method.

Config/reticulate list(packages = list(list(package = ``numpy''), list(package = ``rosetta-soil'')))

License GPL (>= 2)

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

Depends R (>= 3.5)

URL https://github.com/ncss-tech/rosettaPTF, https://ncss-tech.github.io/rosettaPTF/

BugReports https://github.com/ncss-tech/rosettaPTF/issues

Imports parallel, reticulate, terra

Suggests testthat

Repository https://brownag.r-universe.dev

RemoteUrl https://github.com/ncss-tech/rosettaPTF

RemoteRef HEAD

RemoteSha 94634a5945ca19c3f929586153ad7f8ab188f1e6

Contents

`ann_predict` *Extended* Rosetta *Predictions, Parameter Distributions and Summary Statistics after Zhang & Schaap (2017)*

Description

Extended *Rosetta* Predictions, Parameter Distributions and Summary Statistics after Zhang & Schaap (2017)

Usage

```
ann_predict(object, soildata, sum_data = TRUE)

## Default S3 method:
ann_predict(object, soildata, sum_data = TRUE)

## S3 method for class 'Rosetta'
ann_predict(object, soildata, sum_data = TRUE)
```

Arguments

`object`	*Rosetta* object containing class instance (e.g. from `Rosetta()`)
`soildata`	A list containing vectors; with number of parameters matching the model type of `object`
`sum_data`	Default: `TRUE`

Examples

```
# ann_predict(Rosetta(), list(c(30, 30, 40, 1.5), c(55, 25, 20, 1.1)))
```

`find_python` *Heuristics to Find Python*

Description

If you are using the rosettaPTF package for the first time you will need to have Python installed to obtain the necessary modules. You can set up reticulate to install into a virtual or Conda environment. Usually reticulate should cover most or all of the setup.

Usage

```
find_python(
  envname = NULL,
  pypath = NULL,
  arcpy_path = getOption("rosettaPTF.arcpy_path")
)
```

Arguments

envname	As in `reticulate::py_install()`: The name, or full path, of the environment in which Python packages are to be installed. When NULL (the default), the active environment as set by the RETICULATE_PYTHON_ENV variable will be used; if that is unset, then the r-reticulate environment will be used.
pypath	Optional: Path to `python` executable
arcpy_path	Optional: Path to ArcGIS Pro Python installation. For example: `"C:/Program Files/ArcGIS/Pro/bin/Python"`. Set as `NULL` to prevent use of ArcGIS Pro instance.

Details

If you have Python set up correctly you should be able to run `reticulate::py_config()` or `reticulate::py_discover_config()` and discover your shared library.

A common problem with reticulate is not pointing at the correct (or any) `python` binary or `libpython` shared library. Use `reticulate::use_python("/path/to/python", required = TRUE)` to set the path or, alternately, be sure `python` can be found on your PATH. reticulate has a preference for Python environments that have `numpy` installed.

Windows / Miniconda:

Use `reticulate::install_miniconda()` if you'd like to install a Miniconda Python environment. Conda is default on Windows.

For devices with limited ability to install new software that have ArcGIS Pro installed (some USDA computers), this method can look for a Python installation in `"C:/Program Files/ArcGIS/Pro/bin/Python/envs/` and Conda executable in `"C:/Program Files/ArcGIS/Pro/bin/Python/Scripts"`. The base file path to "Python" directory can be customized using the `arcpy_path` argument.

Linux and OS X:

On Linux and OS X you can create a virtual environment using `reticulate::virtualenv_create()`. The default environment name will be `"r-reticulate"`.

Value

character path to `python` or `NULL` if no suitable environment can be found. The result is stored as the package option `rosettaPTF.python_path`.

Examples

```
find_python()
```

`get_rosetta_module` *Get Rosetta module object reference*

Description

Get Rosetta module object reference

Usage

```
get_rosetta_module()
```

Value

An R object wrapping a Python module. Module attributes can be accessed via the `$` operator, or via `py_get_attr()`.

`install_rosetta` *Install Rosetta Python package*

Description

Wrapper around `reticulate::py_install()` to install the Rosetta Pedotransfer Function Python package

Usage

```
install_rosetta(
  envname = NULL,
  method = "auto",
  conda = "auto",
  pip = TRUE,
  user = FALSE,
  upgrade = TRUE,
  system = FALSE,
  arcpy_path = getOption("rosettaPTF.arcpy_path")
)
```

Arguments

envname	The name, or full path, of the environment in which Python packages are to be installed. When NULL (default), the active environment (RETICULATE_PYTHON_ENV variable) will be used; if that is unset, then the "r-reticulate" environment will be used.
method	"auto", "virtualenv", or "conda"; Default: "auto"
conda	Default: "auto"
pip	*logical*. Use pip for package installation? Default: TRUE. This is only relevant when Conda environments are used, as otherwise packages will be installed from the Conda repositories.
user	*logical*. Default: FALSE. Pass --user flag. This should only be done if other installation methods fail and it is impossible to use a virtual environment.
upgrade	*logical*. Install latest versions of Python packages by passing --upgrade flag? Default: TRUE.
system	*logical*. Default: FALSE. If TRUE, try installing to system (user) site library with system() and set reticulate to use system Python.
arcpy_path	Argument passed to find_python(). Path to ArcGIS Pro Python installation e.g. ". Set as NULL (default) to prevent use of ArcGIS Pro instance.

Details

From reticulate::py_install(): On Linux and OS X the "virtualenv" method will be used by default ("conda" will be used if virtualenv isn't available). On Windows, the "conda" method is always used.

MUKEY_PROP *Sample SSURGO Mapunit Properties by* mukey

Description

Sample SSURGO Mapunit Properties by mukey

MUKEY_WCS *Sample SSURGO Mapunit Web Coverage Service Data:* matrix *of* mukey

Description

Sample SSURGO Mapunit Web Coverage Service Data: *matrix* of mukey

predict.Rosetta *Predict Rosetta Parameter Values and Standard Deviations from a* Rosetta *instance*

Description

Predict Rosetta Parameter Values and Standard Deviations from a *Rosetta* instance

Usage

```
## S3 method for class 'Rosetta'
predict(object, soildata, ...)
```

Arguments

object	*Rosetta* object containing class instance (e.g. from Rosetta())
soildata	A list containing vectors; with number of parameters matching the model type of object
...	not used

Examples

```
# predict(Rosetta(), list(c(30, 30, 40, 1.5), c(55, 25, 20, 1.1)))
```

Rosetta *Make a Rosetta object instance for running* `predict()` *methods*

Description

Make a Rosetta object instance for running `predict()` methods

Usage

```
Rosetta(rosetta_version = 3, model_code = 3)
```

Arguments

`rosetta_version`	Default: `3`
`model_code`	One of `2`, `3`, `4`, `5`, or `-1`. Corresponding to options described in *Details*.

Details

Explanation of Model Codes:

- `2`: sand, silt, clay ("SSC")
- `3`: sand, silt, clay + bulk density ("BD")
- `4`: sand, silt, clay + bulk density + field capacity water content (1/3 bar or 33 kPa tension)
- `5`: sand, silt, clay + bulk density + field capacity water content + wilting point water content (15 bar or 1500 kPa tension)
- `-1`: no result returned, inadequate or erroneous data

Value

an instance of the `Rosetta` class defined by the Python module; suitable for running `predict` or `ann_predict` methods.

rosetta_module_available *Check if Rosetta module is available for import from local Python environment*

Description

Check if Rosetta module is available for import from local Python environment

Usage

```
rosetta_module_available()
```

Value

logical

`run_rosetta.default` *Run* `rosetta()` *method from Python module*

Description

Run `rosetta()` method from Python module

Usage

```
## Default S3 method:
run_rosetta(soildata, vars = NULL, rosetta_version = 3, ...)

## S3 method for class 'data.frame'
run_rosetta(soildata, vars = NULL, rosetta_version = 3, ...)

## S3 method for class 'matrix'
run_rosetta(soildata, vars = NULL, rosetta_version = 3, ...)

## S3 method for class 'RasterStack'
run_rosetta(
  soildata,
  vars = NULL,
  rosetta_version = 3,
  cores = 1,
  core_thresh = 20000L,
  file = paste0(tempfile(), ".tif"),
  nrows = nrow(soildata)/(terra::ncell(soildata)/core_thresh),
  overwrite = TRUE
)

## S3 method for class 'RasterBrick'
run_rosetta(
  soildata,
  vars = NULL,
  rosetta_version = 3,
  cores = 1,
  core_thresh = 20000L,
  file = paste0(tempfile(), ".tif"),
  nrows = nrow(soildata)/(terra::ncell(soildata)/core_thresh),
  overwrite = TRUE
)

## S3 method for class 'SpatRaster'
```

```
run_rosetta(
  soildata,
  vars = NULL,
  rosetta_version = 3,
  cores = 1,
  core_thresh = 20000L,
  file = paste0(tempfile(), ".tif"),
  nrows = nrow(soildata)/(terra::ncell(soildata)/core_thresh),
  overwrite = TRUE
)
```

Arguments

soildata	A list of numeric vectors each containing 3 to 6 values: "sand", "silt", "clay", "bulkdensity", "th33", "th1500", a *data.frame* or *matrix* with 3 to 6 columns OR a Raster*/SpatRaster object with 3 to 6 layers.
vars	*character*. Optional: names and order of custom column names if soildata is a *data.frame*, *RasterStack*, *RasterBrick* or *SpatRaster*. Default NULL assumes input column order follows sand, silt, clay, bulkdensity, th33, th1500 and does not check names.
rosetta_version	Default: 3
...	additional arguments not used
cores	number of cores; used only for processing *SpatRaster* or *Raster** input
core_thresh	Magic number for determining processing chunk size. Default 20000L. Used to calculate default nrows
file	path to write incremental raster processing output for large inputs that do not fit in memory; passed to terra::writeStart() and used only for processing *SpatRaster* or *Raster** input; defaults to a temporary file created by tempfile() if needed
nrows	number of rows to use per block chunk; passed to terra::readValues() and terra::writeValues(); used only for processing *SpatRaster* or *Raster** inputs. Defaults to the total number of rows divided by the number of cells divided by core_thresh.
overwrite	logical; overwrite file? passed to terra::writeStart(); defaults to TRUE if needed

Value

A *data.frame* containing mean and stdev for following five columns (parameters for van Genuchten-Mualem equation)

- "theta_r", residual water content
- "theta_s", saturated water content
- "log10(alpha)", 'alpha' shape parameter, log10(1/cm)
- "log10(npar)", 'n' shape parameter
- "log10(Ksat)", saturated hydraulic conductivity, log10(cm/day)

SoilDataFromArray *Convert list of numeric vectors to* SoilData *Python object*

Description

SoilDataFromArray: convert a list of numeric vectors containing soil properties to a rosetta.rosetta.SoilData class
py_to_r(<rosetta.rosetta.SoilData>): Wrapper S3 method for SoilData objects to prevent automatic conversion of SoilData (subclass of "python.builtin.list") to an R "list"

Usage

```
SoilDataFromArray(x)

## S3 method for class 'rosetta.rosetta.SoilData'
py_to_r(x)
```

Arguments

x	a list of numeric vectors

Value

an object reference to a Rosetta *SoilData* Python object constructed from x

Index