

# Reproducible Routing Rituals - Tutorial

By Cédric H. David ([cedric.david@jpl.nasa.gov](mailto:cedric.david@jpl.nasa.gov)),

29 Mar 2016

## Introduction

### Foreword

This tutorial explains how to install and use the Reproducible Routing Rituals (RRR <https://github.com/c-h-david/rrr>).

### Knowledge prerequisites

It is assumed here that the reader has some familiarity with Unix-like operating systems and with the Python programming language.

### Hardware prerequisites

Completion of this tutorial requires access to a personal computer with an Internet connection.

### Software prerequisites

The computer needs an Internet browser and a Unix-like terminal (this type of software is either readily available or can be installed on Windows, Mac and Linux computers).

## Installation of RRR

The easiest way to run RRR is to do so in a Unix-like environment. This is native for Linux and Mac computers, and can be emulated on Windows using Cygwin (<https://www.cygwin.com/>) although the installation and usage of Cygwin is beyond the scope of this tutorial. Such environments include a “terminal” in which command line instructions allow interactions with the computer.

Note that, while it is attempted to keep this tutorial in line with the latest developments in RRR, up-to-date installation information can always be found in the following two files:

- <https://github.com/c-h-david/rrr/blob/master/.travis.yml>
- <https://github.com/c-h-david/rrr/blob/master/requirements.txt>

## Software requirements for RRR

The following software need be installed on your computer:

- The Python programming language in its version 2.6.9 (or later) can be obtained from <https://www.python.org/>. Make sure that Pip Installs Packages (`pip`) is included in your distribution of Python, otherwise it can be obtained from <https://pypi.python.org/pypi/pip>.
- The netCDF Common Operators (NCO) in its version 4.3.6 (or later) can be obtained from <http://nco.sourceforge.net/>.
- The NCAR Command Language (NCL) in its version 6.3.0 (or later) can be obtained from <https://www.ncl.ucar.edu/>
- The Geospatial Data Abstraction Library (GDAL) in its version 1.7.3 (or later) can be obtained from <http://www.gdal.org/>.
- A distributed version control system (`git`) in its version 1.7.12 (or later) can be obtained from <https://git-scm.com/>.

The installation of the above software is beyond the scope of this tutorial. However, example installation instructions can be found in one of the files on the GitHub repository (<https://github.com/c-h-david/rrr/blob/master/.travis.yml>). This file provides up-to-date information on the software required prior to install RRR; and is automatically used by a hosted Continuous Integration service called Travis CI in order to build and test RRR (<https://travis-ci.org/c-h-david/rrr>) on a blank machine every time the source code is updated on GitHub. The details of Travis CI usage are also beyond the scope of this tutorial.

## Download RRR

RRR is kept in a repository on GitHub. The “bleeding edge” version of RRR is always at: <https://github.com/c-h-david/rrr>. The latest official release is always at: <https://github.com/c-h-david/rrr/releases/latest>. Each one of these two links allows to download an archive file (e.g. `.zip`) directly from an internet browser. However, the preferred method for downloading RRR is with `git`:

```
[/home/name/tutorial/]$ git clone https://github.com/c-h-david/rrr.git
[/home/name/tutorial/]$ cd rrr/
[/home/name/tutorial/rrr/]$
```

## Install RRR

At time of writing, the following Python packages are used by RRR:

- dbf (0.96.005) <https://pypi.python.org/pypi/dbf/>
- pyshp (1.2.3) <https://pypi.python.org/pypi/pyshp/>
- netCDF4 (1.0.2) <https://pypi.python.org/pypi/netCDF4/>
- ulmo (0.7.8) <https://pypi.python.org/pypi/ulmo/>
- GDAL (1.7.3) <https://pypi.python.org/pypi/GDAL/>

All of these packages are available through the Python Package Index (<https://pypi.python.org/pypi>).

Up-to-date information on the Python packages required by RRR is in one of the files on the GitHub repository (<https://github.com/c-h-david/rrr/blob/master/requirements.txt>). This file was downloaded with the RRR repository and can be used to install all required Python packages:

```
[/home/name/tutorial/rrr/]$ pip install -r requirements.txt
```

At this point, all Python packages required by RRR should be readily available.

To check what version of a Python package is installed, one can use (for example):

```
[/home/name/tutorial/rrr/]$ pip show netCDF4
```

To install a specific version of a package, one can use (for example):

```
[/home/name/tutorial/rrr/]$ pip install netCDF4==1.0.2
```

## Code architecture in RRR

### Directory and file structure

The RRR file and directory structure is organized as follows:

<code>rrr/</code>	the main code directory
<code>input/</code>	the RRR input data directory
<code>Domain/</code>	the input data directory for a given domain
<code>*.*</code>	all input files for the given domain
<code>output/</code>	the RRR output data directory
<code>Domain/</code>	the output data directory for a given domain
<code>*.*</code>	all output files for the given domain
<code>tst/</code>	the RRR test directory
<code>*.*</code>	all scripts used for testing
<code>src/</code>	the RRR source code directory
<code>*.sh</code>	Shell scripts (bash shell)
<code>*.py</code>	Python scripts
<code>.gitignore</code>	specifies the few files to be ignored in version control
<code>.travis.yml</code>	provides installation and testing procedures.
<code>LICENSE</code>	the Berkeley Software Distribution 3-Clause License.
<code>README.md</code>	describes RRR
<code>requirements.txt</code>	describes the Python Packages required
<code>version.sh</code>	exposes the version of RRR used

### Naming conventions

The file names in the source code directory follow a specific convention in which groups of characters are used to specify what each specific script accomplishes. Each group of characters is separated by an underscore. The first three letters are always `rrr`. The following three letters specify what is the subject of the processing task: land surface model outputs (`lsm`), contributing catchments (`cat`), river reaches (`riv`), coupling land surface and river reaches (`cpl`), stream gauge observations (`obs`), and data analysis (`anl`). The remaining characters are used to further describe the tasks performed.

### Additional information is inside the code

Each script in RRR contains comments that describe the tasks that are performed. These comments are designed to help users understand what each script accomplishes. Perhaps the most important comments of each script are those available at the top: a paragraph called `Purpose` explains the main tasks performed and includes some information on the format of the file that is created, and a section called `Declaration of variables` which lists the inputs used in each script.

## Using RRR in the San Antonio and Guadalupe Basins with NHDPlus v1 and NLDAS2 data

For this tutorial, we will use the smaller of the two domains used in David et al. [2011]: the combination of the San Antonio and Guadalupe River Basins. This domain is one of those shown in Figure 1.

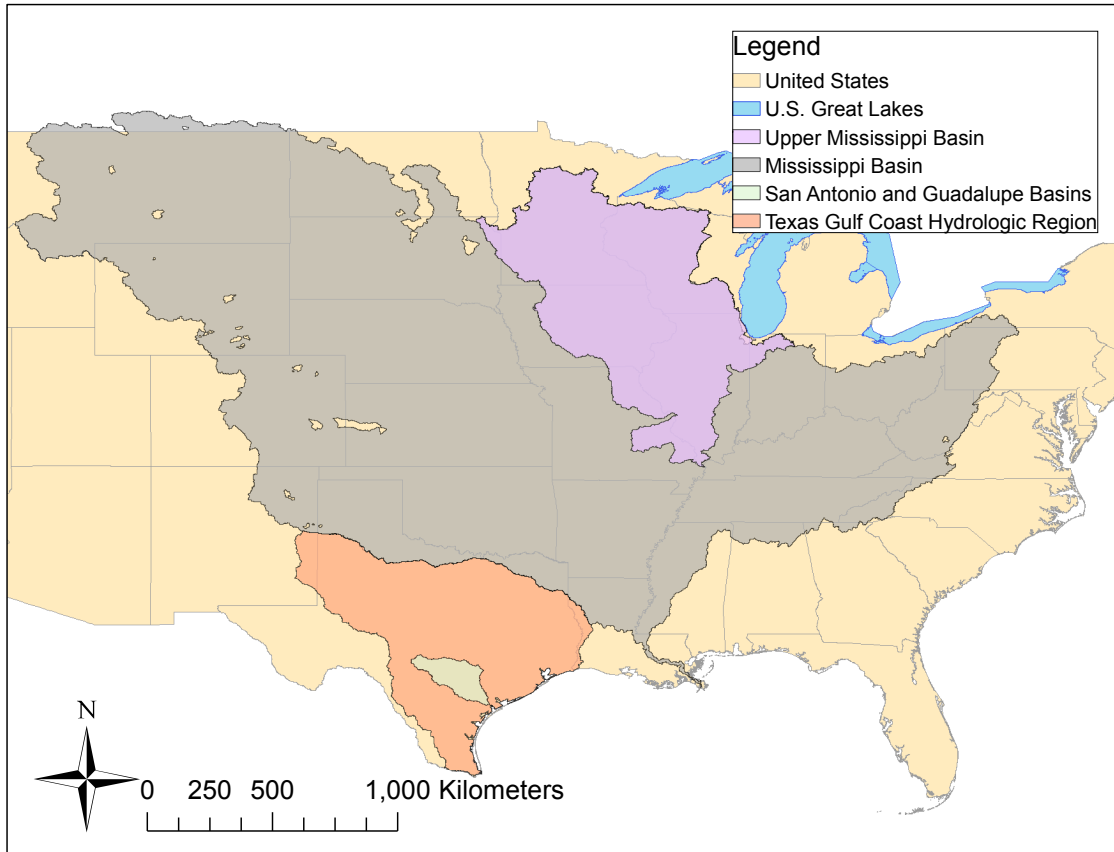


Figure 1 Selected applications of RAPID in the United States, from David et al. [2016]

The hydrographic data corresponding to this domain are actually to be extracted from the larger Texas Gulf Coast Hydrologic Region (Figure 1) and are obtained from the enhanced National Hydrography Dataset [NHDPlus, USEPA and USGS, 2010]. The polygons describing the catchments are at:

[ftp://ftp.horizon-systems.com/NHDplus/NHDPlusV1/TexasGulf/NHDPlus12V01\\_02\\_Catshape.zip](ftp://ftp.horizon-systems.com/NHDplus/NHDPlusV1/TexasGulf/NHDPlus12V01_02_Catshape.zip). The file needed here is called `catchment.shp` (.dbf and .shx extensions are also necessary). The polylines describing the river reaches are at: [ftp://ftp.horizon-systems.com/NHDplus/NHDPlusV1/TexasGulf/NHDPlus12V01\\_02\\_NHD.zip](ftp://ftp.horizon-systems.com/NHDplus/NHDPlusV1/TexasGulf/NHDPlus12V01_02_NHD.zip). The file needed here is called `NHDFlowline.shp` (.dbf and .shx extensions are also necessary). The Guadalupe and San Antonio Basins are identified by their 6-digit hydrologic unit codes from the classic USGS classification (121002 and 121003, respectively). Use a Geographic Information System to generate a subset of the river network that includes only those river reaches for which the REACHCODE attribute starts by 121002 or

121003. The new shapefile should be named `NHFlowline_San_Guad.shp` (.dbf and .shx extensions are also necessary).

The meteorological data used David et al. [2011] are not from off-the-shelf dataset, and this tutorial instead takes advantage of the North American Land Data Assimilation System, phase 2 [NLDAS2, Xia et al., 2012] in a way similar to David et al. [2015]. These data can be obtained found at: [ftp://hydro1.sci.gsfc.nasa.gov/data/s4pa/NLDAS/NLDAS\\_VIC0125\\_H.002/](ftp://hydro1.sci.gsfc.nasa.gov/data/s4pa/NLDAS/NLDAS_VIC0125_H.002/). We will download here all files and directories corresponding to years 2010 to 2013 included.

In the examples given below

- Grey font is for the processing tool name.
- Green font is for the inputs to the tool.
- Red font is for the outputs of the tool.

## Processing land surface model data

### Upcoming

## Processing information on contributing catchments

Example:

```
[/home/name/tutorial/rrr/src/]$ rrr_cat_tot_gen_one_nhdplus.py
  ../input/San_Guad_DDD/catchment.dbf
  ../output/San_Guad_DDD/rapid_catchment_Reg12.csv
```

## Processing information on the network of river reaches

Example:

```
[/home/name/tutorial/rrr/src/]$ ./rrr_riv_tot_gen_all_nhdplus.py
  ../input/San_Guad_DDD/NHDFlowline_San_Guad.dbf
  ../input/San_Guad_DDD/NHDFlowlineVAA.dbf
  4
  ../output/San_Guad_DDD/rapid_connect_San_Guad.csv
  ../output/San_Guad_DDD/kfac_San_Guad_1km_hour.csv
  ../output/San_Guad_DDD/xfac_San_Guad_0.1.csv
  ../output/San_Guad_DDD/sort_San_Guad_hydroseq.csv
  ../output/San_Guad_DDD/coords_San_Guad.csv
```

Example:

```
[/home/name/tutorial/rrr/src/]$ ./rrr_riv_tot_scl_prm.py
  ../output/San_Guad_DDD/kfac_San_Guad_1km_hour.csv
  ../output/San_Guad_DDD/xfac_San_Guad_0.1.csv
  0.35
  3.0
  ../output/San_Guad_DDD/k_San_Guad_ag.csv
  ../output/San_Guad_DDD/x_San_Guad_ag.csv
```

Example:

```
[/home/name/tutorial/rrr/src/]$ ./rrr_riv_bas_gen_one_nhdplus.py
  ../input/San_Guad_DDD/NHDFlowline_San_Guad.dbf
  ../output/San_Guad_DDD/rapid_connect_San_Guad.csv
  ../output/San_Guad_DDD/sort_San_Guad_hydroseq.csv
```

```
sv ..../output/San_Guad_DDD/riv_bas_id_San_Guad_hydroseq.c
```

## Coupling land surface models and river networks

Example:

```
[/home/name/tutorial/rrr/src/]$ rrr_cpl_riv_lsm_lnk.py
    ../output/San_Guad_DDD/rapid_connect_San_Guad.csv
    ../output/San_Guad_DDD/rapid_catchment_Reg12_arc.csv
    ../output/San_Guad_DDD/NLDAS_VIC0125_3H_20100101_2013
1231_cst.nc
    ../output/San_Guad_DDD/rapid_coupling_San_Guad_NLDAS2
.csv
```

Example:

```
[/home/name/tutorial/rrr/src/]$ ./rrr_cpl_riv_lsm_vol.py
    ../output/San_Guad_DDD/rapid_connect_San_Guad.csv
    ../output/San_Guad_DDD/coords_San_Guad.csv
    ../output/San_Guad_DDD/NLDAS_VIC0125_3H_20100101_2013
1231_cst.nc
    ../output/San_Guad_DDD/rapid_coupling_San_Guad_NLDAS2
_arc.csv
    ../output/San_Guad_DDD/m3_riv_San_Guad_20100101_20131
231_VIC0125_cst.nc
```

## References

- David, C. H., D. R. Maidment, G.-Y. Niu, Z.-L. Yang, F. Habets, and V. Eijkhout (2011), River Network Routing on the NHDPlus Dataset, *J. Hydrometeorol.*, 12(5), 913–934.
- David, C. H., J. S. Famiglietti, Z.-L. Yang, and V. Eijkhout (2015), Enhanced fixed-size parallel speedup with the Muskingum method using a trans-boundary approach and a large subbasins approximation, *Water Resour. Res.*, 51, 1–25, doi:10.1002/2014WR016650.
- David, C. H., J. S. Famiglietti, Z.-L. Yang, F. Habets, and D. R. Maidment (2016), A Decade of RAPID - Reflections on the Development of an Open Source Geoscience Code, *Earth Space Sci.*, *accepted on 2016-03-22*.
- USEPA, and USGS (2010), *NHDPlus User Guide*, available online at [ftp://ftp.horizon-systems.com/NHDPlus/NHDPlusV1/documentation/NHDPLUSV1\\_UserGuide.pdf](ftp://ftp.horizon-systems.com/NHDPlus/NHDPlusV1/documentation/NHDPLUSV1_UserGuide.pdf).
- Xia, Y. et al. (2012), Continental-scale water and energy flux analysis and validation for the North American Land Data Assimilation System project phase 2 (NLDAS-2): 1. Intercomparison and application of model products, *J Geophys Res*, 117(D3), D03109.