

DIMSpec Quick Guide - Installation

Jared M. Ragland*

2023-03-03

Getting set up quickly with DIMSpec

If your needs require only the database itself and not the attached toolkit, the SQLite file may be downloaded directly from the NIST Public Data Repository (this link is not yet active) and, along with the technical sections in the DIMSpec User Guide and the data dictionary describing its schema (a JSON file located in the project directory), used by any platform able to interact with SQLite databases.

To get the most from DIMSpec, the software toolkit ¹ provided with the project enables a wide variety of customized functionality including convenience, utility, and visualization functions written in the R language, an Application Programming Interface powered by `plumber`, web applications powered by `shiny`, and an integration of the python interface to the `rdkit` chemometrics package. A file is provided at `/config/env_glob.txt` which enables configuration customization including database title, database file name, and several other options affecting system integration. Due to its intended reuse and level of available customization, the DIMSpec project is not distributed as an R package.

The easiest way to get started with the full DIMSpec project is by forking the project repository from GitHub, though it may also be downloaded without a GitHub account as a zip file for ease of use (if so, simply extract it to a location of your choosing). Install the required software (see below) and run the compliance script (e.g. `source("/R/compliance.R")` from the project directory). This should in most cases establish the computational environment supporting the toolkit. When starting from scratch this may take a while as packages are downloaded and installed.

*NIST | MML | CSD | jared.ragland@nist.gov

¹NIST-developed software is provided by NIST as a public service. You may use, copy, and distribute copies of the software in any medium, provided that you keep intact this entire notice. You may improve, modify, and create derivative works of the software or any portion of the software, and you may copy and distribute such modifications or works. Modified works should carry a notice stating that you changed the software and should note the date and nature of any such change. Please explicitly acknowledge the National Institute of Standards and Technology as the source of the software.

NIST-developed software is expressly provided "AS IS." NIST MAKES NO WARRANTY OF ANY KIND, EXPRESS, IMPLIED, IN FACT, OR ARISING BY OPERATION OF LAW, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, AND DATA ACCURACY. NIST NEITHER REPRESENTS NOR WARRANTS THAT THE OPERATION OF THE SOFTWARE WILL BE UNINTERRUPTED OR ERROR-FREE, OR THAT ANY DEFECTS WILL BE CORRECTED. NIST DOES NOT WARRANT OR MAKE ANY REPRESENTATIONS REGARDING THE USE OF THE SOFTWARE OR THE RESULTS THEREOF, INCLUDING BUT NOT LIMITED TO THE CORRECTNESS, ACCURACY, RELIABILITY, OR USEFULNESS OF THE SOFTWARE.

You are solely responsible for determining the appropriateness of using and distributing the software and you assume all risks associated with its use, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and the unavailability or interruption of operation. This software is not intended to be used in any situation where a failure could cause risk of injury or damage to property. The software developed by NIST employees is not subject to copyright protection within the United States.

Requirements

The DIMSpec project ² is distributed as an R project built around R version 4.1 and later, and leverages a large number of packages to run as designed. A development environment such as RStudio ³ is highly recommended but is not required. For integration of chemometrics, which is used in several aspects of the project but is not a strict requirement, a python environment including `reticulate` and `rdkit` is required; this will in many cases be installed when the compliance script runs. This can be relaxed for those preferring to use an R chemometrics package, and if informatics is requested in the environment file but `rdkit` is not, ChemmineR and rcdk will be installed; note that these are not integrated into the project at this time. If you will be developing new databases using this infrastructure, SQLite and its command line interface (CLI) are required; this is what allows the R session to build a fresh database for population. As the project is hosted on GitHub, Git is recommended as the easiest method to download the project and provide version control. SQLite, its CLI, and Git must be available on your system PATH for proper functionality.

Step by Step

See the DIMSpec User Guide for full instructions and how to use DIMSpec. These instructions are a summary of those found there.

1. Install software requirements for your operating system
 - R v4.1 or later (download)
 - RTools (optional but may be required for package installation on Windows) (download)
 - RStudio (optional but recommended) (download)
 - SQLite3 (download) and its command line interface (download) (both optional but recommended)
 - Git (optional but recommended) (download)
 - A lightweight database interaction interface (e.g. DBeaver Lite (download) or similar) (suggested)
 - Miniconda with python 3.8 or later (suggested only if there are problems, the compliance script will install RMiniconda for you if chemometrics support is requested) (download)
2. (Optional, advanced) Customize any settings for your implementation (see the Project Set Up section of the DIMSpec User Guide)
3. Run the script at `/R/compliance.R` to establish the compute environment, which by default will:
 - Install (if necessary) and load all necessary packages.
 - Integrate python and `rdkit` functionality.
 - Connect to a local SQLite database (one is included with the project download).
 - Establish logging and argument verification.
 - Launch a `plumber` API for database communication with applications.
 - Make three `shiny` web applications available.

These steps may take a while the first time. After that, start DIMSpec again from a clean R session by sourcing the compliance script to take care of loading necessary packages. The preferred method is to use it as an RStudio project.

²This release was tested on a fresh VMWare build of Ubuntu 20.04 LTS which carries several additional system requirements. Prior to running DIMSpec, install or make sure the following are available using:
`apt install -y build-essential libcurl4-openssl-dev libxml2-dev zlib1g-dev libssl-dev libsodium-dev ffmpeg libtiff-dev libpng-dev libblas-dev liblapack-dev libarpack2-dev gfortran libcairo2-dev libx11-dev libharfbuzz-dev libfribidi-dev libudunits2-dev libgeos-dev libgdal-dev libfftw3-3 libmagick++-dev`

After following the R installation instructions for Ubuntu, ensure additional requirements using:

`apt install -y --no-install-recommends r-cran-tidyverse r-cran-shiny`

³Any mention of commercial products is for information only; it does not imply recommendation or endorsement by NIST.

Developing With DIMSpec

More information on developing with DIMSpec is provided in the full DIMSpec User Guide or the Quick Guide - Developing with DIMSpec

A Note on Environments

While the provided convenience script for environment establishment (located at `/R/compliance.R`) should in most cases establish the proper environment for you, it is admittedly using a large number of packages. To increase flexibility across various computing and operating systems without modifying libraries that may already be in place, a resolved environment is not provided. The most common place where difficulties may be encountered therefore is package version conflicts and setting up the python environment providing `rdkit` integration. For troubleshooting purposes, the list of required packages can be located IN the `DEPENDS_ON` environment variable in the `/config/env_R.R` file. Generally, package issues in R will print a meaningful error message to the console and guide users toward a resolution. These occasionally arise due to conflicting dependencies; the order in which packages are installed in the compliance script should minimize this. On Windows systems, some R packages require that RTools be installed first. RStudio should provide a streamlined experience for installing packages compared with base R.

Speed Concerns

The full DIMSpec project can take a while to load at first. To speed this up, it is recommended that the compliance script be sourced prior to engaging in any development work with DIMSpec. Two developer resources are provided in the project that may slow down execution times. Both logging and argument validation are performed by default, which increases execution time but provides a wealth of information to assist with development. These may be turned off individually In the environment settings by setting `LOGGING_ON` or `VERIFY_ARGUMENTS` to `FALSE`, respectively; set `MINIMIZE` to `FALSE` to turn off both at once.

Development Assistance Functions

Navigating a project such as DIMSpec, especially one with several environment files determining behaviors, can be daunting. There are two functions specifically built with developers in mind to assist with opening project files from anywhere. Use the `open_env()` function to open an environment file for editing with the only argument being the environment you wish to edit; valid values are “R”, “global”, “logging”, “rdkit”, “shiny”, or “plumber”. This leverages `open_proj_file`, a more generic function to open any file in the project by partial name match. Both of these are tied to the project directory and leverage the `here` package.

Additionally, since this project is NOT a package, custom functions to assist with R function help documentation are also provided. Use `fn_guide` to open the DIMSpec function help index in your system browser, or use `fn_help(X)` where `X` is the name of a function (quoted or unquoted) to open that documentation directly (if using RStudio, this will open in the help pane).