Tox21 Enricher Setup Guide

University of North Dakota

Parker Combs

2021

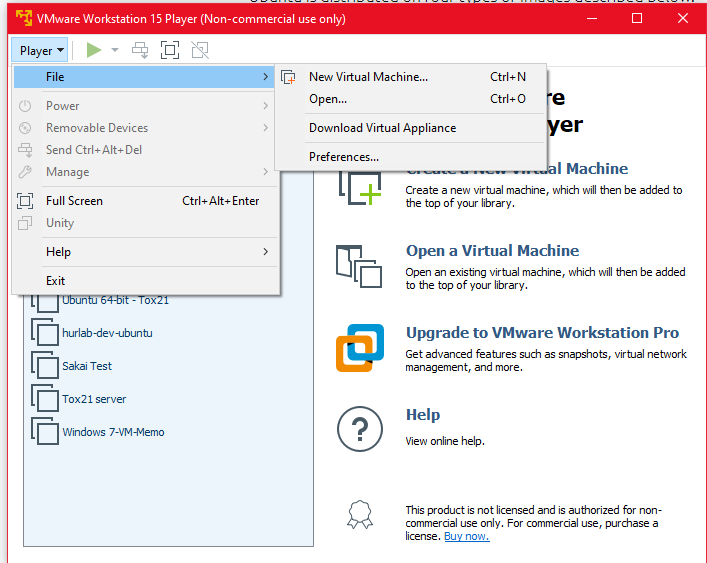
**Table of Contents**

1. Setting Up the Work Environment
   1. OS and Virtual Machine Settings
   2. Software and Frameworks
   3. Packages and Libraries
   4. Networking
2. Getting and Installing the Files
   1. Tox21 Enricher Program Files
   2. Tox21 Enricher Database
3. Running the Application in a Test Environment
   1. Tox21 Enricher (Grails)
   2. Tox21 Enricher (R Shiny)
4. Deployment
5. Updating the Database
6. Directory Structure

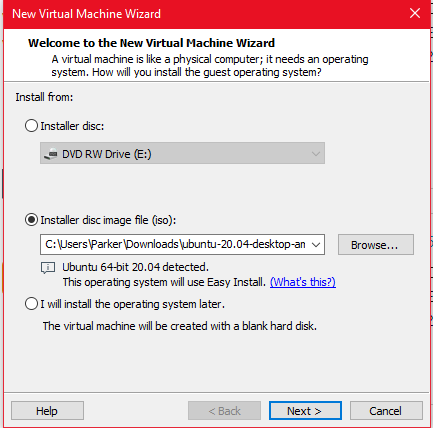
# Setting Up the Work Environment

## 1.a. OS and Virtual Machine Settings

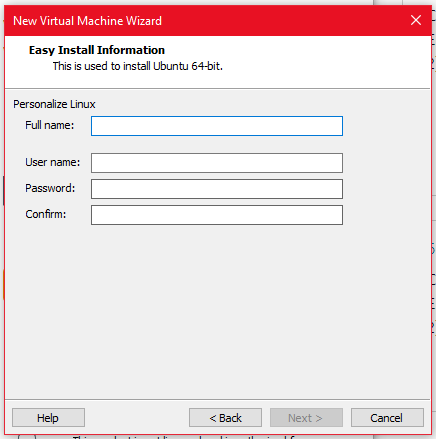
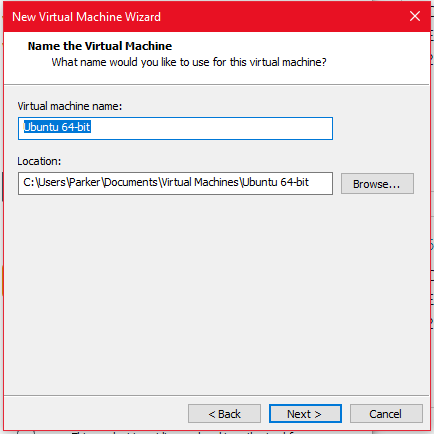
1. Tox21 Enricher was developed on a VMware Workstation 15 Player virtual machine running Ubuntu 20.04 LTS. First, install VMware Workstation 15 Player (or the most recent version) from vmware.com/products/workstation-player.html.
2. Next, download an Ubuntu 20.04 LTS (Focal Fossa) desktop image from <https://releases.ubuntu.com/20.04/>.
3. Open VMware Player and create a new VM by clicking **Player** 🡪 **File** 🡪 **New Virtual Machine…**



1. Select **Installer disc image file (iso)** on the next menu and choose the Ubuntu image we just downloaded.



1. Fill in the personal information in the next screen.

1. When specifying disk capacity, specify at least 50 GB of storage.
2. Create the default user “hurlab.” We will store the application files under the **/home/hurlab/** directory.

## 1.b. Software and Frameworks

Tox21 Enricher requires the following languages be installed to run:

|  |  |
| --- | --- |
| **Name** | **Version** |
| Groovy | 2.5.6 |
| Java | OpenJDK ver. 11.0.\* |
| Perl | 5.30.0 |
| Python | 3.6.10 |
| R | 3.6.3 |

Additionally, the application also requires the following frameworks and tools:

|  |  |
| --- | --- |
| **Name** | **Version** |
| Anaconda | Anaconda 3 ver. 1.7.2 |
| Conda | 4.9.2 |
| Grails | 4.0.2 |
| PostgreSQL | 12.2 |
| RDKit | 2021\_03\_1 (Q1 2021) Release |

1. To manage Java, Groovy, and Grails installations, we recommend using the SDKMAN! tool. It can be found here: <https://sdkman.io/>. You may need to install the libcurl4/curl packages to download SDKMAN!
   1. After installing SDKMAN!, use the command **sdk ls java** to find the names of all the available Java candidates. Use the command **sdk install java 11.0.11-open** to install Java 11.
   2. You will need to install Java to the **/usr/lib/jvm/default-java/** directory as well.
   3. Use **sdk install grails 4.0.2** to install Grails.
   4. Use **sdk install groovy 2.5.6** to install Groovy.
2. Perl 5.30.0 should be pre-installed with Ubuntu 20.04 LTS. If it is not, for whatever reason, download it here: <https://www.perl.org/get.html>.
3. To install R, follow the instructions here: <https://cloud.r-project.org/>. Installation of R packages will be covered in section 1.c. We recommend using the R Studio IDE, which can be downloaded here: <https://www.rstudio.com/products/rstudio/download/#download>. You may need to install the **gdebi** package to install R Studio.
4. Finally, install Anaconda, Conda, Python, RDKit, and PostgreSQL.
   1. First, download the Anaconda installer here: <https://www.anaconda.com/products/individual>. Then, follow the instructions here to install: <https://conda.io/projects/conda/en/latest/user-guide/install/linux.html>.
   2. Next, to install the RDKit PostgreSQL cartridge and Python 3.6, first create a new environment by running the command **conda create -n my-rdkit-env python=3.6**. When this finishes, use the command **conda activate my-rdkit-env** to activate the environment.
   3. With the environment activated, use the command **conda install -c rdkit rdkit-postgresql** to install the PostgreSQL cartridge.

## 1.c. Packages and Libraries

1. The R API component the Tox21 Enricher application requires loading a number of additional packages on top of the base R packages. They are:

* config
* ggplot2
* httr
* parallel
* plumber
* plyr
* pool
* rjson
* RPostgreSQL
* stringr
* tidyverse
* uuid
* xlsx

1. Additionally, the Shiny version of Tox21 Enricher requires:

* catmaply
* *CePa*\*\*
* DT
* plotly
* rjson
* shiny
* shinyBS
* shinycssloaders
* shinydashboard
* shinyjs

\*\* To install the CePa package, you must first install the BiocManager package, then use the commands **BiocManager::install(“Rgraphviz”) and BiocManager::install(“graph”)**.

These can be installed using the **install.packages(“package\_name”)** function in R. Additionally, some of these packages have dependencies not listed here. Some packages may require a suitable C or C++ compiler to install, such as **gcc** or **g++**. Additionally, the **make** utility is required to build some of these packages.

1. If you have trouble installing the above packages, you may need to install the following Linux packages using **sudo apt install <package\_name>**:

* g++
* gcc
* libcurl4-openssl-dev
* libpq-dev
* libsodium-dev
* libssl-dev
* libxml2-dev
* libz-dev
* make
* r-base-dev

## 1.d. Networking

Tox21 Enricher uses a few ports. These are:

* 5432 – PostgreSQL server
* 8082 – R Plumber API

It is recommended that you host the Tox21 Enricher Grails application on port 8080 or 8081.

# Getting and Installing the Files

## 2.a. Tox21 Enricher Program Files

1. The Tox21 Enricher files for the Grails application, R Shiny application, and R Plumber API can be downloaded from the public Tox21 Enricher GitHub repository: <https://github.com/hurlab/tox21enricher>.
2. Clone the repository to your home directory. The path to the repository files should be **/home/<user\_name>/tox21enricher/**.
3. Open the root **tox21enricher/** directory and make sure it at least contains the **grails-app/**, **src/**, **Input/**, and **Output/** directories. If the **Input/** and **Output/** directories are not in the **tox21enricher/** directory by default, just create two empty folders with those names.
4. Next, navigate to the **tox21enricher/src/main/r/** directory. If this directory contains **Input/** and **Output/** directories, delete them and create symbolic links from **tox21enricher/Input/** to **tox21enricher/src/main/r/Input/** and **tox21enricher /Output/** to **tox21enricher/src/main/r/Output/**.
5. Next, navigate to the **tox21enricher/src/main/perl/HClusterLibrary/** directory. Make sure the clusterLinux64 application has the correct permissions to be run as an executable by running **chmod 755**.
6. Finally, ensure all the R packages as described in section 1.c are properly installed.

## 2.b. Tox21 Enricher Database Files

1. Contact Parker Combs or Dr. Junguk Hur to receive a backup copy of the Tox21 Enricher database.
2. Use the **initdb** utility to initialize the Tox21 Enricher database. You can follow the directions here: <https://www.rdkit.org/docs/Install.html>.
3. Make sure you change the PostgreSQL login credentials in the Tox21 Enricher code to match your installation of PostgreSQL. Files that contain the login credentials are:
   1. **tox21enricher/** **grails-app/conf/application.yml**
   2. **tox21enricher/** **grails-app/conf/application.groovy**
   3. **tox21enricher/src/main/r/config.yml**
4. Load **psql** as the “postgres” user and run the command **CREATE DATABASE tox21enricher;**.
5. To restore the database copy to the **tox21enricher** database, use the **psql** utility as described here <https://www.postgresql.org/docs/12/backup-dump.html>. Run the command **anaconda3/envs/my-rdkit-env/bin/psql tox21enricher < <name\_of\_databse\_backup>**.

# Running the Application in a Test Environment

## 3.a. Tox21 Enricher (Grails)

1. First, make sure that the PostgreSQL daemon is running by using the command: **<path\_to\_anaconda\_installation>/envs/my-rdkit-env/bin/pg\_ctl -D <path\_to\_database> start**.
2. Next, open a terminal and navigate to the **tox21enricher/src/main/r/** directory. Run the startPlumberAPI.sh script to start the R Plumber API server. On startup, the API must query the database and fetch some data, so it may take a few seconds for it to completely start up. Ensure that the startPlumberAPI.sh script has the correct permissions to be run as an executable. Alternatively, you can run the R command **R -e 'library(plumber); r <- plumb("./performEnrichment.R"); r$run(host="127.0.0.1",port=8082)'**.
3. Next, open a terminal and navigate to the **tox21enricher/** directory. Use the command **grails run-app** to run the Grails app in a test environment.
4. Open a web browser and navigate to **localhost:8080/tox21enricher** (or whatever port you have the application running on) to access the Grails app’s interface. Navigate to localhost:8082 to view the API’s interface.

## 3.b. Tox21 Enricher (R Shiny)

1. First, make sure that the PostgreSQL daemon is running by using the command: **<path\_to\_anaconda\_installation>/envs/my-rdkit-env/bin/pg\_ctl -D <path\_to\_database> start**.
2. Next, open a terminal and navigate to the **tox21enricher/src/main/r/** directory. Run the startPlumberAPI.sh script to start the R Plumber API server. On startup, the API must query the database and fetch some data, so it may take a few seconds for it to completely start up. Ensure that the startPlumberAPI.sh script has the correct permissions to be run as an executable. Alternatively, you can run the R command **R -e 'library(plumber); r <- plumb("./performEnrichment.R"); r$run(host="127.0.0.1",port=8082)'**.
3. In the same R directory, you can run the Tox21 Enricher Shiny application from the command line by using the command: **R -e "shiny::runApp('./tox21enricher')"**. You can also open the R files in the **tox21enricher/src/main/r/tox21enricher** directory in R Studio and run the application from within R Studio.

# Deployment

1. To generate a .war archive for web deployment (such as on an Apache Tomcat server) for the Grails application, first navigate to the **tox21enricher/** directory.
2. Open a terminal in this directory and use the command **grails war** to generate a .war archive.
3. The resulting .war can be found in **tox21enricher/build/libs/tox21enricher-0.1.war**.

# Updating the Database

To add additional annotations to the Tox21 Enricher database, follow these steps:

1. Create a working directory. Make sure that it contains (at the minimum) the following items:
   1. 01\_ReOrganize\_Annotation\_Data\_v2.1-20161003.pl script
   2. 02\_Split-perform\_Fisher-Exact-Test.pl script
   3. 03\_Merge\_Pvalues-and-calculate-qvalues.pl script
   4. 04\_Create\_complete\_annotation\_matrix\_v1.0\_11182020.pl script
   5. matrix\_names.py script
   6. prepare\_matrix.py script
   7. A folder called “Annotation” that contains the Tox21\_CASRN\_Names\_Full.anno file
2. Prepare a file for each set of annotations you would like to add. Each line must be formatted starting with a CASRN, followed by a tab, followed by a list of associated terms, delimited by a semicolon and a space. For example: **100-01-6 GO:0070887|cellular response to chemical stimulus; GO:0051716|cellular response to stimulus;**
3. Give each of these files a unique name and append the prefix “DrugMatrix\_” to the beginning of the name. Save each file as a text (.txt) file.
4. Place all these formatted annotation files inside of the Annotation folder.
5. The following three scripts process and format the data that populates the database:
   1. Run the 01 Perl script.
   2. When the 01 script completes, run the 02 Perl script. This may take a very long time (multiple days).
   3. When the 02 script completes, run the 03 Perl script.
6. The following three scripts process and format the annotation matrix. These can be run while the previous three scripts are running.
   1. Run the 04 Perl script.
   2. When the 04 script completes, run the matrix\_names.py script.
   3. When the 04 script completes, run the prepare\_matrix.py script.
7. After the above scripts complete, several new files should be created in the directory in which the scripts are located. Some of these files are used to populate the Tox21 Enricher database as described below:

|  |  |
| --- | --- |
| **File** | **Database table** |
| table\_annotation\_class\_v2.1.txt | annotation\_class |
| table\_annotation\_detail\_v2.1.txt | annotation\_detail |
| term2casrn\_mapping\_v2.1.txt | term2casrn\_mapping |
| table\_annoterm\_pairwise\_v2.1\_FINAL.txt | annoterm\_pairwise |
| names.txt | annotation\_matrix\_terms |
| out.txt | annotation\_matrix |

After loading the contents of the files into the database, all the files that were created can be deleted.

# Directory Structure

* tox21enricher (root folder for project)
  + grails-app
  + Input
    - Input files for each enrichment process are generated by the application and are stored here.
  + Output
    - Result files for each enrichment process are generated by the application and are stored here.
  + src
    - integration-test (not used, will likely remove in a future update)
    - main
      * groovy
        + tox21\_test (not really sure what this is used for, I don’t think it’s used by the application)
      * perl
        + Annotation (This must contain the Tox21\_CASRN\_Names\_Full.anno file)
        + HClusterLibrary (This contains the GenePattern modules that perform the clustering and heatmap generation)
      * Python (this contains various Python scripts that are called from the Grails app. They are mostly used to process chemical inputs)
      * r
        + Annotation (This must contain the Tox21\_CASRN\_Names\_Full.anno file)
        + Input 🡪 tox21enricher/Input (symlink)
        + Output 🡪 tox21enricher/Output (symlink)
        + shinyJSME-master (Contains shiny files)
        + tox21enricher (Contains the R scripts that make up the R Shiny version of the application)

www (Contains some files needed for the R Shiny version of the application)

* + - * templates
      * webapp
    - test (not used, will likely remove in a future update)