TBsim App installation guide

Ron Keizer (InsightRX)

Introduction

TBsimApp is a web application to simulate the in-vivo quantitative pharmacology of tuberculosis drug combinations. It is based on a command-line based tool developed at UCSF (TBsim), and implemented as a Shiny app.

Requirements

Software requirements (for cloud installation)

- GridEngine (SGE or GE) installed. Basically any gridengine that uses standard submit (`qsub`), and info (`qstat`) commands should work. Refer to GE installation guide for more info.
- Shiny server (Open Source version) installed. See RStudio's Shiny Server Administrator's guide for more details: Shiny Server v1.5.6 Configuration Reference.
- Several custom R packages, mainly TBsim (BitBucket), TBsimApp (GitHub), RGE (https://github.com/ronkeizer/Rge)

Server requirements

Instances optimized for computation as well as with significant memory available are recommended. Please note that the results from the simulations for each user will be saved in `/data/tbsim`. Make sure that when running this on a cloud machine there is enough space available to support the number of intended users. E.g. when running on AWS it is suggested to attach EBS for this instead of using the EC2-instance's own storage.

Required R libraries

Various supporting R libraries need to be installed, see below the commands to install them.

```
## Base packages:
install.packages("devtools", "Rcpp", "stringr", "lubridate")
```

```
## Shiny extensions
install.packages(c("shinythemes", "shinydashboard",
                   "shinyjs", "shinyTable", "shinyBS",
                   "rhandsontable", "googleAuthR"))

## non-CRAN libraries
library(devtools)
install_github("MarkEdmondson1234/googleID")
install_github("ronkeizer/Rge")

## TBsim libraries
install_bitbucket("ucsf_ip/TBsim")
install_github("InsightRX/TBsimApp")

## Need to check if really necessary:
install_github("ronkeizer/PKPDplot")
install_github("ronkeizer/vpc")
```

Running TBsimApp locally

It should be fairly straightforward to set up the app locally, since both the backend tool (`TBsim` , currently on BitBucket) and the Shiny app (`TBsimApp` , currently on Github at `InsightRX/TBsimApp`) are distributed as R libraries. The Shiny app is included in the `TBsimApp` library in the `inst` folder, so will be copied to the library repository along with the R functions in the `TBsimApp` library.

After installing the libraries listed above, go to the `/inst/app` folder inside the `TBsimApp` repository and run on the command line:

```
PORT=1410 LOCAL=1 Rscript run.R
```

Running TBsimApp in the cloud

The major difference between running in the cloud and running locally are that:

- the *Server* version of Shiny is used

- Grid Engine is used for queuing of population simulations (allow multiple users and long runs)
- Google authentication is used to be able to restrict and track usage, and store simulated data on the server.

Installing TBsimApp

For details on running apps in Shiny Server, please refer to the Shiny Server manual from RStudio. After installing Shiny Server, and making sure that the demo app included with Shiny Server works, just copy the contents of the `inst/app` folder into the Shiny Server app repository at `/srv/shiny-server/`, either overwriting the demo app, or into a separate folder. After reloading the Shiny Server, the app should show when browsing to the URL (e.g. `http://localhost:1410`).

Updating TBsimApp

In brief, to update the app, the following steps are generally required:

- pull the latest R libraries `TBsim` and `TBsimApp`, and re-install. Make sure that they are installed from the `shiny` user account on the Linux box, otherwise they might not be picked up by Shiny Server.
- copy the `inst/app` folder from the `TBsimApp` repository to the Shiny Server app repository on the server (`/srv/shiny-server/`)
- Restart the Shiny Server:

```
sudo stop shiny-server
sudo start shiny-server
```

(According to RStudio, this is better than using `sudo restart shiny-server`, since the former destroys any cache the server might hold.)

To perform the last two steps, we used the following bash script:

```
sudo rm -rf /srv/shiny-server/app
sudo cp -R /home/shiny/git/TBsimApp/inst/app /srv/shiny-server/app
sudo touch /srv/shiny-server/app/.httr-oauth
sudo stop shiny-server
sudo start shiny-server
```

Nginx configuration

We chose to run an NGINX server in front of the Shiny app to allow domain redirection (in this case to point from tbsim.insight-rx.com to the actual URL of the Shiny app). NGINX also makes it easier to set up quotas for file upload, and potentially load distribution (if needed in the future).

The configuration file we used is shown below.

```
server {
    server_name tbsim.insight-rx.com;
    listen 80;
    client_max_body_size 2M;

    location / {
      proxy_http_version 1.1;
      proxy_set_header Upgrade $http_upgrade;
      proxy_set_header Connection "upgrade";
      proxy_pass http://127.0.0.1:3838/app;
      proxy_redirect http://127.0.0.1:3838/app $scheme://$host/;
    }
    location /app {
      proxy_http_version 1.1;
      proxy_set_header Upgrade $http_upgrade;
      proxy_set_header Connection "upgrade";
      proxy_pass http://127.0.0.1:3838/app;
      proxy_redirect http://127.0.0.1:3838/app $scheme://$host/;
    }
 }
```

Google authentication

Google OAuth is used in the app to allow user login, which basically allows restriction of the usage of the app (if desired) as well as a way to keep track of usage and to identify users and store data per-user. The R packages `googleID` and `googleAuthR` are used for this purpose.