Version 2

Apr 08, 2021

How to expose existing R code as a web service V.2

Sonia García-Ruiz[1]

[1]University College London, University of London

1 *Works for me* This protocol is published without a DOI.

Ryten Laboratory

Sonia García-Ruiz

ABSTRACT

This protocol shows how to expose any existing R method as a web service. The code shown here has been used as part of the development of the CoExp Web Application https://rytenlab.com/coexp/. For further information on CoExp Web, please see our paper: https://www.frontiersin.org/articles/10.3389/fgene.2021.630187/full

The R methods used in this protocol correspond to the family of CoExpNets R packages (https://github.com/juanbot/CoExpNets).

EXTERNAL LINK

https://rytenlab.com/swagger/index.html

THIS PROTOCOL ACCOMPANIES THE FOLLOWING PUBLICATION

García-Ruiz S, Gil-Martínez AL, Cisterna A, Jurado-Ruiz. F, Reynolds RH, Cookson MR, Hardy J, Ryten M and Botía JA (2021) CoExp: A Web Tool for the Exploitation of Co-expression Networks. Front. Genet. 12:630187. doi: 10.3389/fgene.2021.630187

EXTERNAL LINK

https://rytenlab.com/swagger/index.html

PROTOCOL CITATION

Sonia García-Ruiz 2021. How to expose existing R code as a web service. **protocols.io**
https://protocols.io/view/how-to-expose-existing-r-code-as-a-web-service-bt3wnqpe
Version created by Sonia García-Ruiz

MANUSCRIPT CITATION please remember to cite the following publication along with this protocol

García-Ruiz S, Gil-Martínez AL, Cisterna A, Jurado-Ruiz. F, Reynolds RH, Cookson MR, Hardy J, Ryten M and Botía JA (2021) CoExp: A Web Tool for the Exploitation of Co-expression Networks. Front. Genet. 12:630187. doi: 10.3389/fgene.2021.630187

WHAT'S NEW

Two dockerized versions of CoExp, which can be found on https://hub.docker.com/r/soniaruiz/coexp

KEYWORDS

CoExp, R Plumber, API, REST API, HTTP, co-expression networks

LICENSE

CREATED

Apr 08, 2021

LAST MODIFIED

Apr 08, 2021

PROTOCOL INTEGER ID

48982

MATERIALS TEXT

Two dockerized versions of CoExp - backend and frontend - can be found on https://hub.docker.com/r/soniaruiz/coexp

BEFORE STARTING

All steps and code snippets shown in this protocol have been tested in a machine with the following characteristics:

- Operating system: CentOS Linux 7 (Core)
- R version 3.5.1 (2018-07-02) -- "Feather Spray".
- CoExpNets R packages installed.

1 First, we install the R package Plumber. To install the latest stable version from CRAN, we can type the following R command:

```
install.packages("plumber")
```

2 The next step will consist of creating the API specification to expose our R methods of interest. In this case, we are going to expose the method 'CoExpNets::getNetworkCategories()' from the CoExpNets family of R packages (https://github.com/juanbot/CoExpNets). This step might sound daunting, but it is very straightforward: we only need to generate an R file to include on it all methods we would like to expose through the web API.

In this example, the contents of the new API R file (let's call it 'api.R') should look as follows:

```
#' @get /getNetworkCategories #this is the name we will type as part of the URL on our
browser when calling the API method
function(){
print("getNetworkCategories() GET method has requested!") # a friendly message
CoExpNets::getNetworkCategories() # calling the R method
}
```

3 Once changes have been saved, the next step would consist of running the web API. To do so, we first need to choose an empty port, open an R console and type the following commands:

```
## Code to load CoExpNets libraries
library(CoExpNets)
library(CoExpROSMAP)
CoExpROSMAP::initDb(mandatory=T)

## Code to expose the API
library(plumber)
pr("/pathtoyourfile/api.R") %>% pr_run(port=8080)
```

4 Finally, we test the API:

```
curl "localhost:8080/getNetworkCategories"
```

Also available via URL:

```
http://localhost:8080/getNetworkCategories
```

In both cases, the response should be similar to:

```
["CoExpROSMAP"]
```