

PlsCO User Manual

PlsCO Prerequisites	1
Software	1
Database	1
PlsCO specification	2
PlsCO installation	2
PlsCO Registry Usage	3
Run server	3
Register metrics through the web interface	3
PlsCO Data and Monitoring Repository Usage	4
Run server	4
Export metrics metadata from Registry using the web interface	4
Data monitoring	5
Data visualization	5

A. PlsCO Prerequisites

1. Software

- i. Nodejs and npm:
 sudo apt-get update
 sudo apt-get install nodejs
 sudo apt-get install npm
- ii. MongoDB:
 <https://docs.mongodb.com/manual/installation/>

2. Database

- i. Create "metricsdb" database and "resources" collection:
 mkdir **your_path**/data
 mkdir **your_path**/data/db
 cd **your_path**/data/db

```
[16:11:57][n61806][admin]~$ mongo
MongoDB shell version: 3.0.4
connecting to: test
> use metricsdb
switched to db metricsdb
> db.createCollection("resources",{ })
{"ok" : 1}
> show dbs
local          0.078GB
metricsdb      0.078GB
> show collections
resources
system.indexes
```

We provided a copy of our metrics database which can be used to test the section [E.4](#)
Data visualization:

- Extract the metricsdb.tar.gz file.
- Start mongodb service (see in this section 2.ii)
- From your **your_path/data/db**, run command **mongorestore --db metricsdb path_where_extract_gz_file**

```
pisco-metrics-framework/data/db$ mongorestore --db metricsdb ~/projects/data/db/dump/metricsdb
2017-02-08T16:40:10.134+0000 building a list of collections to restore from /usr/users/ga002/artazah/projects/data/db/dump/metricsdb dir
2017-02-08T16:40:10.134+0000 reading metadata file from /usr/users/ga002/artazah/projects/data/db/dump/metricsdb/resources.metadata.json
2017-02-08T16:40:10.134+0000 restoring metricsdb.resources from file /usr/users/ga002/artazah/projects/data/db/dump/metricsdb/resources.bson
2017-02-08T16:40:10.393+0000 restoring indexes for collection metricsdb.resources from metadata
2017-02-08T16:40:10.394+0000 finished restoring metricsdb.resources
2017-02-08T16:40:10.394+0000 done
pisco-metrics-framework/data/db$ ll
total 163864
-rwxr-xr-x 4 artazah ga002 4096 Feb 8 16:40 ./
-rwxr-xr-x 3 artazah ga002 4096 Feb 7 15:55 ../
-rwxr-xr-x 2 artazah ga002 4096 Feb 8 16:39 journal/
-rw-r--r-- 1 artazah ga002 67108864 Feb 8 16:39 local.0
-rw-r--r-- 1 artazah ga002 16777216 Feb 8 16:39 local.ns
-rw-r--r-- 1 artazah ga002 67108864 Feb 8 16:40 metricsdb.0
-rw-r--r-- 1 artazah ga002 16777216 Feb 8 16:40 metricsdb.ns
-rwxr-xr-x 1 artazah ga002 6 Feb 8 16:39 mongod.lock*
-rw-r--r-- 1 artazah ga002 69 Feb 7 15:56 storage.bson
-rwxr-xr-x 2 artazah ga002 4096 Feb 8 16:40 _tmp/
```

- ii. Start mongodb service:
- Configure the file db_start.sh (this file is located on *pisco-metrics-framework*):

```
#!/bin/bash

####Start mongod Processes
database_path="your_path/data/db"

nohup mongod --dbpath $database_path > mongo_db_exc.log &
```

- Run command **bash db_start.sh**

- iii. Stop mongodb service:
- Configure the file db_stop.sh (this file is located on *pisco-metrics-framework*):

```
#!/bin/bash

##Start mongod Processes
database_path="your_path/data/db"

mongod --dbpath $database_path --shutdown
```

- Run command **bash db_stop.sh**

B. PlsCO specification

See specification document for more details:

<https://drive.google.com/file/d/0B3RWb8BCtZH1Uk1qbDJkN3ZoWEU/view?usp=sharing>

C. PlsCO installation

1. mkdir **work_home_directory**
2. cd **work_home_directory**
3. git clone <https://github.com/BioPisCO/pisco-metrics-framework.git>
4. cd pisco-metrics-framework
5. sudo npm install (this command installs all the dependencies in node_modules folder, see Fig 1.)

Figure 1. npm install command output.

D. PlsCO Registry Usage

1. Run server

- cd **work_home_directory**/pisco-metrics-framework/componentregistry/server
- Run the Registry server: **node server** (see Fig. 2).

Figure 2. Run the Registry server.

2. Register metrics through the web interface

- To access the Registry web interface: <http://127.0.0.1:8082/register> (see Fig. 3)
- To register a new metric, you should insert the URL contained the metadata file. If you are using our metric examples:
 - Citation metric metadata:**
<https://raw.githubusercontent.com/BioPisCO/metrics-module-citation/master/schema.xml>.
 - Tweeter metric metadata:**
<https://raw.githubusercontent.com/BioPisCO/metrics-module-tweets/master/schema.xml>.
 - Pageviews metric metadata:**
<https://raw.githubusercontent.com/BioPisCO/metrics-module-pageviews/master/schema.xml>.

Name	Description	Authors	Frequency	Repository
Metric name	A brief description about this metric	TGAC, EBI, ELIXIR	execution time interval	https://github.com/metric-repository
metrics-module-citation	Citations in publications	Artaza Haydee, Manuel Corpas, Rafael Jimenez	monthly	https://github.com/BioPisCO/metrics-module-citation.git
metrics-module-tweets	Topic spread	Artaza Haydee, Manuel Corpas, Rafael Jimenez	monthly	https://github.com/BioPisCO/metrics-module-tweets.git
metrics-module-pageviews	Page views count	Artaza Haydee, Corpas Manuel, Jimenez Rafael	daily	https://github.com/BioPisCO/metrics-module-pageviews.git

Figure 3. Registry web interface. The first row is an example which describes each field in the table.

- iii. The new metric will be added at the end of the list of registered metrics and the metadata will be included in the file:
work_home_directory/pisco-metrics-framework/componentregistry/schema/registryschema.xml.

E. PIsCO Data and Monitoring Repository Usage

1. Run server

- i. cd **work_home_directory**/pisco-metrics-framework/datamonitoringrepository/server
- ii. Run the Data and Repository server: **node server** (see Fig. 4)

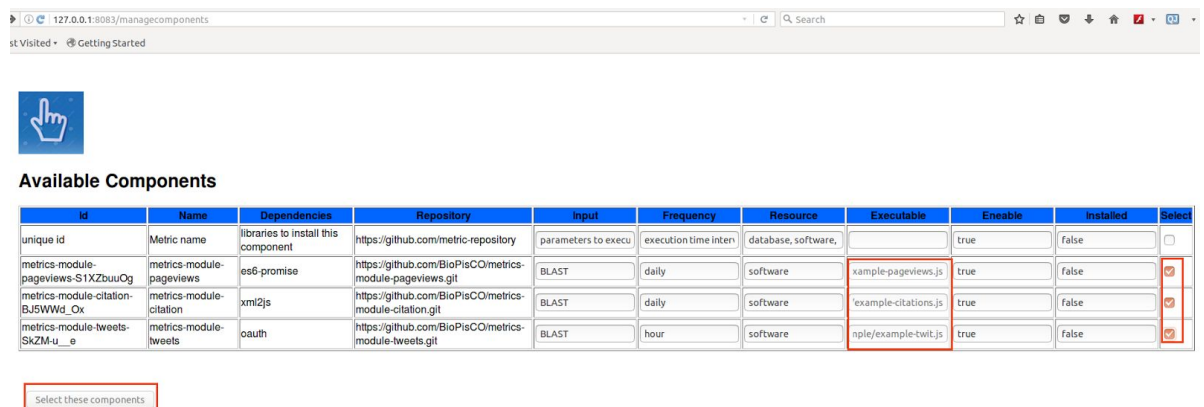
```
/pisco-metrics-framework/datamonitoringrepository/server$ ls
installcomponents.js  runcomponents.js  server.js
artazah@n80062:~/Downloads/pisco-metrics-framework/datamonitoringrepository/server$ node server
Server running at http://127.0.0.1:8083
```

Figure 4. Run Data and Monitoring server.

2. Export metrics metadata from Registry using the web interface

Web interface in <http://127.0.0.1:8083/managecomponents>. This interface manages every metric properties that will be included in Selected Components XML file (export the metadata from Registry). Figure 5 shows the list of metrics available:

- i. Properties in the text boxes can be modified.
- ii. Select the checkbox to include the metrics that you want to export.
- iii. Introduce the executable path and name file. If you are using our metric examples, you must introduce:
 - o *In Citation metric*: example/example-citations.js
 - o *In Tweeter metric*: example/example-twit.js
 - o *In Pageviews metric*: example/example-pageviews.js
- iv. Finally, click in "selected these components" button
- v. The metrics metadata will be exported to:
work_home_directory/pisco-metrics-framework/datamonitoringrepository/schema/selectedcomponents.xml (see Fig. 6).



The screenshot shows a web browser window with the address bar displaying '127.0.0.1:8083/managecomponents'. Below the browser window, there is a blue hand icon pointing to the 'Available Components' section. This section contains a table with the following columns: id, Name, Dependencies, Repository, Input, Frequency, Resource, Executable, Enable, Installed, and Select. The table lists three components: 'metrics-module-pageviews-S1XZbuuOg', 'metrics-module-citation-BJ5WWd_Ox', and 'metrics-module-tweets-SkZM-u_e'. Each component has a 'Select' checkbox in the last column, which is checked for all three. Below the table, there is a button labeled 'Select these components'.

id	Name	Dependencies	Repository	Input	Frequency	Resource	Executable	Enable	Installed	Select
unique id	Metric name	libraries to install this component	https://github.com/metric-repository	parameters to execu	execution time inter	database, software,		true	false	☐
metrics-module-pageviews-S1XZbuuOg	metrics-module-pageviews	es6-promise	https://github.com/BioPisCO/metrics-module-pageviews.git	BLAST	daily	software	example-pageviews.js	true	false	☒
metrics-module-citation-BJ5WWd_Ox	metrics-module-citation	xml2js	https://github.com/BioPisCO/metrics-module-citation.git	BLAST	daily	software	example-citations.js	true	false	☒
metrics-module-tweets-SkZM-u_e	metrics-module-tweets	oauth	https://github.com/BioPisCO/metrics-module-tweets.git	BLAST	hour	software	example-twit.js	true	false	☒

Select these components

Figure 5. Manage Components web interface. Data in this table comes from Component Registry (see section D)

Figure 6. Metadata exportation message.

3. Data monitoring

- i. File `selectedcomponents.xml` contains the list of metrics selected to be monitored. You can find an example in:
work_home_directory/pisco-metrics-framework/datamonitoringrepository/schema/selectedcomponents.xml.example. Otherwise, you can use the web interface to export this data (see section [E.2](#)).
- ii. Run the command **sudo node installcomponents** located in
work_home_directory/pisco-metrics-framework/datamonitoringrepository/server. This script brings the metrics code from their Source Code Management (described in the Repository URL in `selectedcomponents.xml`) and installs each metric in **components** directory. Also the metrics dependencies are installed in **node_modules** directory (see Fig. 7)

```
/pisco-metrics-framework/datamonitoringrepository/server$ ll
total 24
-rwxr-xr-x 2 4096 Feb 7 15:05 ./
-rwxr-xr-x 10 4096 Feb 7 14:54 ../
-rw-r--r-- 1 189 Feb 6 10:15 installcomponents.js
-rw-r--r-- 1 474 Feb 6 10:15 runcomponents.js
-rw-r--r-- 1 6055 Feb 7 12:16 server.js

/pisco-metrics-framework/datamonitoringrepository/server$ sudo node installcomponents
https://nodeload.github.com/BioPisCO/metrics-module-tweets/zip/master
https://nodeload.github.com/BioPisCO/metrics-module-pageviews/zip/master
PisCO@0.1.0 /pisco-metrics-framework
└─ oauth@0.9.15

https://github.com/BioPisCO/metrics-module-tweets.git download into ../components/metrics-module-tweets-4JHxt54kW completed
Installing dependencies...
Library oauth installed
Component metrics-module-tweets-4JHxt54kW has been installed

PisCO@0.1.0 /pisco-metrics-framework
└─ es6-promise@4.0.5
└─ mongodb@2.2.22
└─ es6-promise@3.2.1

https://github.com/BioPisCO/metrics-module-pageviews.git download into ../components/metrics-module-pageviews-SJ6c11c- completed
Installing dependencies...
Library es6-promise installed
Component metrics-module-pageviews-SJ6c11c- has been installed

PisCO@0.1.0 /pisco-metrics-framework
└─ xml2js@0.4.17

https://github.com/BioPisCO/metrics-module-citation.git download into ../components/metrics-module-citation-4yyFa7VQx completed
Installing dependencies...
Library xml2js installed
Component metrics-module-citation-4yyFa7VQx has been installed

/pisco-metrics-framework/datamonitoringrepository/server$ ll ../components/
total 24
-rwxr-xr-x 5 4096 Feb 7 15:05 ./
-rwxr-xr-x 10 4096 Feb 7 14:54 ../
-rwxr-xr-x 5 root root 4096 Feb 7 15:05 metrics-module-citation-4yyFa7VQx/
-rwxr-xr-x 5 root root 4096 Feb 7 15:05 metrics-module-pageviews-SJ6c11c-/
-rwxr-xr-x 5 root root 4096 Feb 7 15:05 metrics-module-tweets-4JHxt54kW/
-rw-r--r-- 1 183 Oct 31 14:44 readme.txt
```

Figure 7. Install components output. Logs show the dependencies installed. Also, it shows the **component** directory with the metrics installed.

- iii. After install all the metrics, run the command **node runcomponents.js**. This script starts the monitoring of every metric and it will store the results in the **metricsdb** database (see Fig. 8). You must have started the mongo service before run this command (see [A.2](#)). You can manage the time configuration (we use Cron-parse format) modifying the config file:
work_home_directory/pisco-metrics-framework/datamonitoringrepository/config/registryschemas.xml.

```
/pisco-metrics-framework/datamonitoringrepository/server$ node runcomponents
../components/metrics-module-pageviews-S1XZbuu0g/example/example-pageviews.js daily: 0 34 10 * * *
../components/metrics-module-citation-BJ5Wwd_Ox/example/example-citations.js daily: 0 34 10 * * *
../components/metrics-module-tweets-SkZM-u_e/example/example-twit.js hour: 0 34 10 * * *
Wed Feb 08 2017 10:34:00 GMT+0000 (GMT)---> Run command: node ../components/metrics-module-pageviews-S1XZbuu0g/example/example-pageviews.js "BLAST"
Wed Feb 08 2017 10:34:00 GMT+0000 (GMT)---> Run command: node ../components/metrics-module-citation-BJ5Wwd_Ox/example/example-citations.js "BLAST"
Wed Feb 08 2017 10:34:00 GMT+0000 (GMT)---> Run command: node ../components/metrics-module-tweets-SkZM-u_e/example/example-twit.js "BLAST"
stdout: 745
wiki article(s): BLAST found.

Connection established to mongodb://localhost:27017/metricsdb
Inserted data into the "resources" collection
Component metrics-module-pageviews-S1XZbuu0g was executed
stdout: citation Test: Running.
130347 citations for BLAST
data save into > citation.txt

Connection established to mongodb://localhost:27017/metricsdb
Inserted data into the "resources" collection
Component metrics-module-citation-BJ5Wwd_Ox was executed
```

Figure 8. Metrics monitoring. Logs show the frequency of execution, the resources monitoring and other messages obtained from each metric.

4. Data visualization

This section explains the details of the web interface which makes available data from metrics monitoring. You must remember that the main purpose of PlsCO framework is to collect, register and share data from metrics. You can use this data to elaborate your own statistics and graphics, and interpret this information at your convenience. We have implemented a first GUI version to show some graphics with data extracted from the metrics database.

- i. You must set up the file:
work_home_directory/pisco-metrics-framework/datamonitoringrepository/schema/availablecomponents.xml. If you are testing our metrics you can change the metric ID to adjust this parameter with your data (see Fig. 9).

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<availablecomponents>
  <component>
    <id>metrics-module-citation-4ywN_j5H</id>
    <name>Citation</name>
    <description>Provides evidence of the resource impact in
publications saved in Pubmed Central (PMC)
repository</description>
    <detail>
      <api>NCBI API (E-utilities)</api>
      <query>Keyword separated by comma</query>
      <output>Total publication that includes the
keyword</output>
    </detail>
    <frequency>daily</frequency>
    <resource>
      <name>BLAST</name>
      <type>software</type>
    </resource>
  </component>
</availablecomponents>
```

Figure 9. Availablecomponents.xml file. This file contains the metadata that will be shown in the GUI. The id parameter must be the same id that the database metric id.

- ii. Access the web interface in : <http://127.0.0.1:8083/availablecomponents>. The interface shows the list of available metrics, a brief description, and the number of records in the database (see Fig. 10).

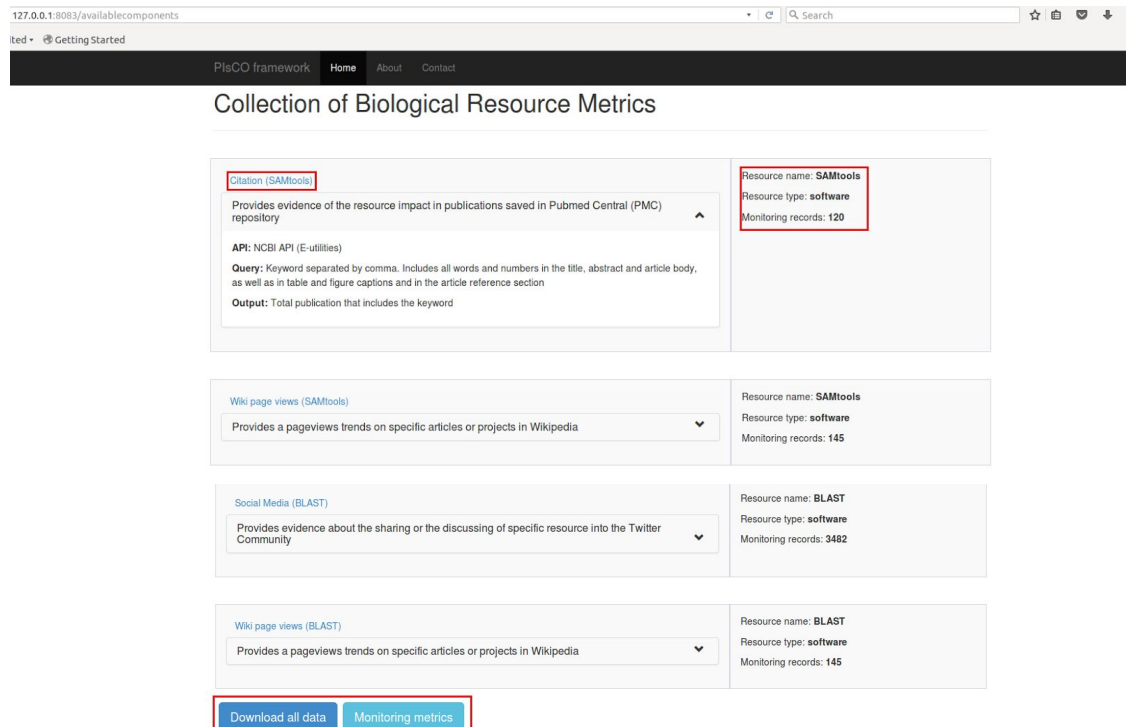


Figure 10. Web interface with the list of available metrics.

- iii. Click in the metric link (blue text) to view the graphic of each metric values filtered by date. Also, you can download this data en CSV and JSON format (see Fig. 11).

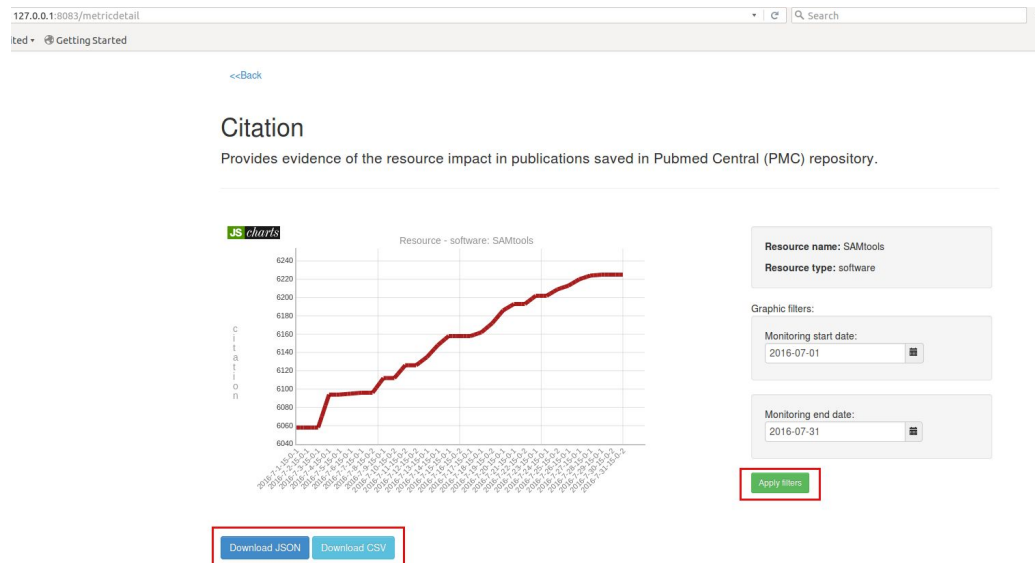


Figure 11. Metric Citation. The graphic shows the values obtained for this metric in entire July.

- iv. Go back and click in "Download all data" button to access the page information in text format.
- v. Click in "Monitoring metrics" button to view the graphic of all metrics filtered by date and type (group by metric o resource, see Fig. 12).

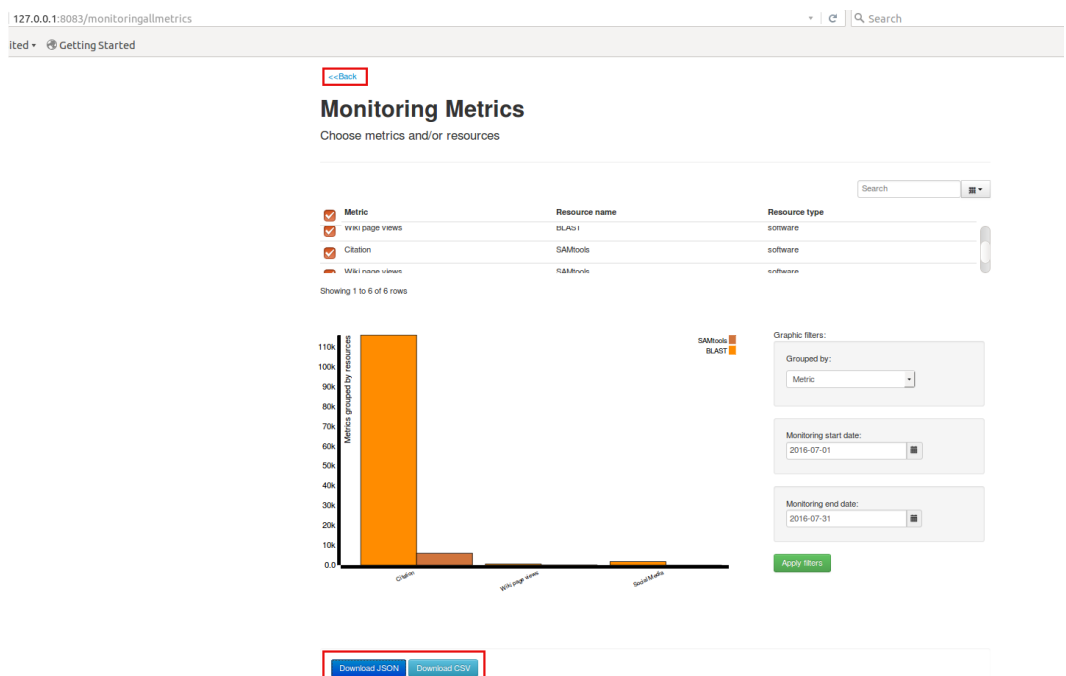


Figure 12. Monitoring metrics. The graphic show a bar chart grouped by metric and with data from entire July