Embedding Scilab within Galaxy, a fast way to build a web application including workflows without line commands

Jean-Claude BOULET

INRA, Montpellier, France

CheMoocs, a collaborative project, was launched in 2016. This MOOC massive open online courses was intended to bring knowledge to the students. But knowledge alone is of no use. So a web application named ChemFlow was also designed, for students to practice with chemometrics.

Students were expected to learn the mathematical insights of chemometrics. It was hardly possible to ask them to learn a programming language. Thus, we needed an interface without command lines. And as students could come from all over the world, it had to be available on the net. Galaxy fitted to this requirements (https://galaxyproject.org). Most of the functions were already available into FACT, an Atoms module of Scilab. So it was pretty easy to run them from Galaxy. ChemFlow was created.

Galaxy and Scilab are installed on a server (https://vm-chemflow-francegrille.eu). The Galaxy window is opened with any navigator. Let us call "tool" the Galaxy interface for a Scilab function. A xml wrapper has been created for each tool, including a Scilab script. When the tool is called, the wrapper calls “scilab-cli”, which runs the script whose main features are: importing Galaxy variables within Scilab, processing the function, sending back the output to Galaxy, closing Scilab.

Benefits are:

1- recycling code: an example was given with Scilab, but almost all programming languages can be run within Galaxy (Octave, R, Python, C for example)
2- possibility of workflows = a sequence of successive tools
3- online / shared histories / free
4- merging several languages: a tool can process data with Scilab, then edit figures with R

“we can choose the best tools among several languages, not a language a priori”

A drawback is the use of the Matlab format to exchange dataset within Galaxy, not readable by an editor. Our next challenge will be a common dataset format with an editor within Galaxy. The HDF5 format is our best challenger, and by luck Scilab is based on HDF5, contrary to Matlab or R! This is a strength that Scilab should be proud of.