

Programming in the Life Sciences #21: 2014 Screenshots #1

Egon Willighagen 

Published January 11, 2015

Citation

Willighagen, E. (2015, January 11). Programming in the Life Sciences #21: 2014 Screenshots #1. *Chem-bla-ics*. <https://doi.org/10.59350/vbvzk-2d025>

Keywords

Pra3006, Openphacts

Abstract

December saw the end of this year's PRA3006 course (aka #mcspils). Time to blog some screenshots of the student projects. Like last year, the aim is to use the Open PHACTS API to collect data with ops.js and which should then be visualized in a HTML page, preferably with d3.js. This year, all projects reached that goal.

Copyright

Copyright © Egon Willighagen 2015. Distributed under the terms of the [Creative Commons Attribution 4.0 International License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

December saw the end of this year's PRA3006 course (aka [#mcspils](#)). Time to blog some screenshots of the student projects. Like last year, the aim is to use the [Open PHACTS API](#) to collect data with [ops.js](#) and which should then be visualized in a HTML page, preferably with [d3.js](#). This year, all projects reached that goal.

ACE inhibitors

The first team (Mischa-Alexander and Hamza) focused on the [ACE inhibitors](#) (type:“drug class”) and the [WP554 from WikiPathways](#). They use a tree structure to list inhibitors along with their activity:

Compound Searcher for Angiotensin-converting enzyme (Homo sapiens)

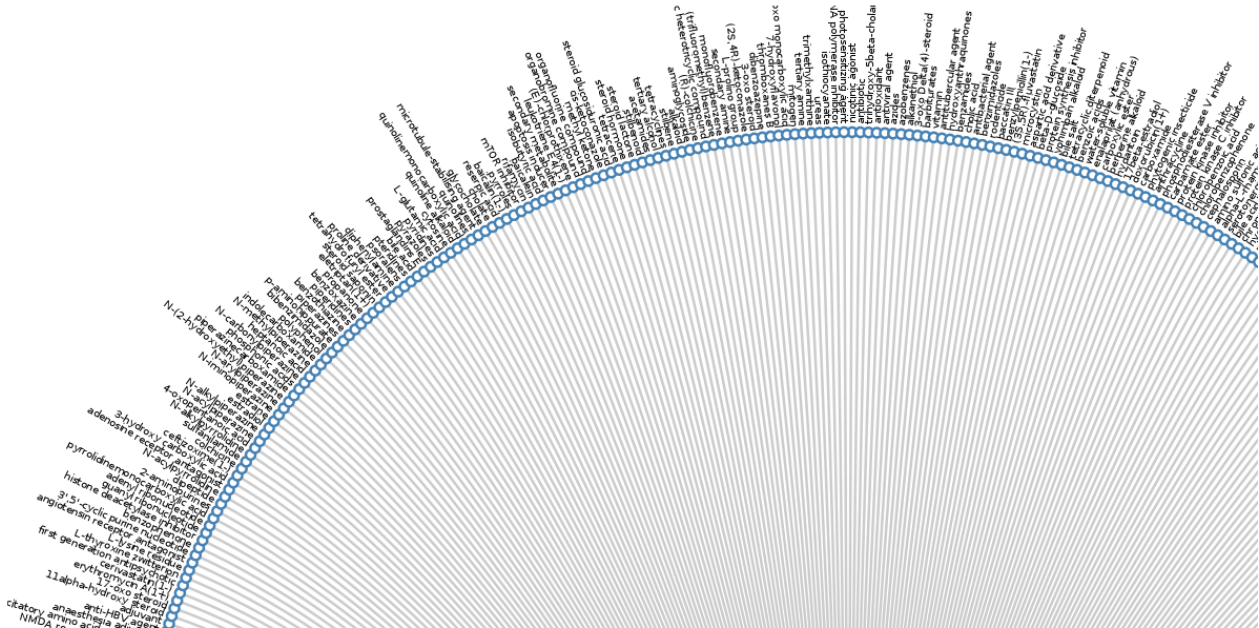


The source code for this project is available under a MIT license.

Diabetes

The second team (Catherine and Moritz) looked at compounds hitting diabetes mellitus targets. They take advantage from the new disease API methods and first ask for all targets for the disease, and then query for all compounds. Mind you, the compounds are not filtered by activity, so it mostly shows interactions that real targets.

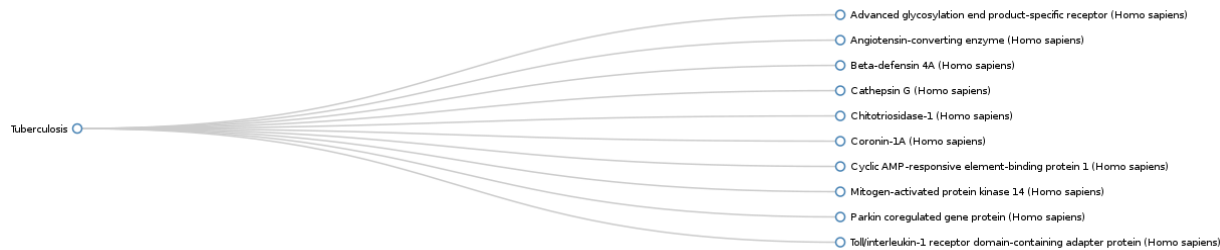
Active Coumpounds for all Targets of Diabetes Mellitus



This product too is available with the MIT license.

Tuberculosis

The third project (Nadia en Loic) also goes from disease to targets and they looked at tuberculosis.



Asynchronous calls

If you know the ops.js, d3.js, and JavaScript a bit, you know that these projects are not trivial. The remote web service calls are made in an asynchronous manner: each call comes with a callback function that gets called when the server returns an answer, at some future point in time. Therefore, if you want to visualization, for example, compounds with activities against targets for a particular disease, you need two web service calls, with the second made in the callback function of the first call. Now, try to globally collect the data from that with JavaScript and HTML, and make sure to call the visualization call when all information is collected! But even without that, the students need to convert the returned web service answer into a format that d3.js can handle. In short: quite a challenge for six practical days!