

Hi all,

My name is Georg Summer and I am a PhD Student at the Maastricht University and TNO (tno.nl). My research focus is the application of network biology concepts and methods in cardiology and nutritional health. Additionally I want to further these applications of network biology with some solid technologies like Cytoscape and Neo4j.

I did my BSc and MSc at the Upper Austrian University of Applied Sciences in Bioinformatics. I did research internships at the Visual Genomics Centre of the University of Calgary and at emergentec biodevelopment in Vienna.

My Master thesis was written about network inference from time-series data at the Perkins Lab of the Ottawa Health Research Institute [accompanying publication: [PMID:21143801](#)].

After that I worked as a bioinformatician for the Heymans group at the Maastricht University which eventually transitioned into my PhD. During that time (and still) I did transcriptomics analysis and visualization, data integration tasks and clinical data collection and curation. Graphs, networks and network biology are topics that followed me throughout my studies and work experience.

My daily work requires mostly R as a programming and analysis language. C++ and Java are the two languages I use for my software development needs. I am experienced in Matlab and php / html / java script but do not use these regularly.

I mostly use open source software and except some bug reports have not contributed to any larger open source development. Changing that is also a major reason why I wanted to get on board with the GSoC this year.

### **Project Idea Introduction:**

Neo4j provides a potential computational backend for more performance intensive tasks required of Cytoscape. The idea is to allow the user to connect and upload a local Cytoscape graph to a Neo4j instance (preferably on a high performance machine) and then invoke operations on this instance via Cytoscape. The necessary calculations are executed on the Neo4j instance and then streamed back to the client. This approach turns Cytoscape to a thin client and treats a Neo4j server as a mainframe.

# Cytoscape

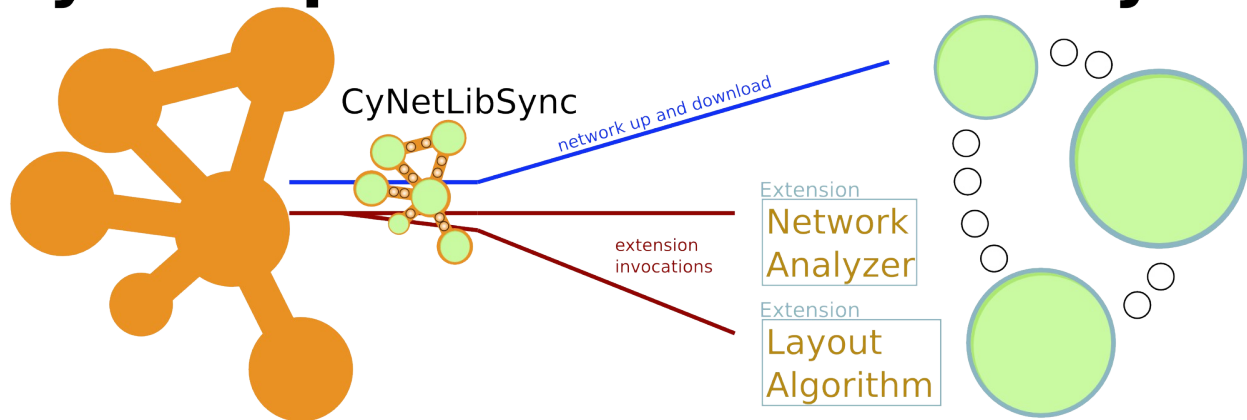


Figure 1: Conceptual view of the Cytoscape to Neo4j connection

I developed a prototype providing the up- and download functionality as well the invocation of Neo4j extensions. The source code and a brief demo are available at:

- <https://github.com/gsummer/CyNetLibSync>
- <http://www.youtube.com/watch?v=XRRK4xtEM8s>

Out of the box Neo4j servers do not provide layouts algorithms or graph statistics for large graphs but Neo4J has an extension facility that allows for the implementation of such algorithms.

Alternatively algorithms could also be submitted directly to Neo4j as it is an open source project. This proposal aims to recreate the Network Analysis Toolbox of Cytoscape (NetworkAnalyzer App) as a Neo4j extension as well as 1-2 layout algorithms. This small collection of extensions will serve as a showcase of how computationally expensive algorithms (especially on large networks) can be deployed and used.

Additionally the implementation also serves as a study to investigate how feasible it is to port existing algorithms bound to Cytoscape to Neo4j extensions.

As a real world showcase a network integrating the following data sources is planned:

- STRING
- WikiPathways
- miRNA - gene (importing CyTargetLinker RegINS)
- drug - gene (importing CyTargetLinker RegINS)

These data sources can either be imported in Neo4j directly via various scripts (Network Builder <https://github.com/thomaskelder/network-builder>) or can first be combined in Cytoscape and then synchronized to a Neo4j instance (already supported in the prototype)

**Project Deliverables:**

### *1. Neo4j extension implementation Network Analysis Toolbox*

The Neo4j version of the toolbox will do the same calculations as the Cytoscape app and return to the client the results in the same way as the Cytoscape app stores the network statistics. If possible this will then be automatically fed into the app and displayed as normal, otherwise user interaction or an adaptation of the existing app will be necessary.

### *2. Layout Algorithms as Neo4j extensions*

Implement at least 1 layout algorithm as a Neo4j extension. The choice of algorithm is to be decided. After the execution of the extension it streams back the updated x/y coordinates for the nodes and then applied to the current network view.

### *3. Updated CyNetLibSync to support and integrate the new extensions*

The existing prototype provides a facility to invoke Neo4j extensions. To properly support the planned ones and integrate them into Cytoscape the app has to be updated with the necessary event handling.

## Project Plan:

Date	Task
2014-04-21	<i>Project Start</i>
	Check if Cytoscape App to Neo4j Extension wrapper is feasible
	Identify NetworkAnalyzer algorithms needed
	Identify layout algorithm(s) to be implemented
	Final decision which datasets to use in the showcase network
2014-05-19	<i>Start Coding</i>
	Implementation of NetworkAnalyzer algorithms as Neo4j extensions
	Implementation of layout algorithms as Neo4j extensions
2014-06-27	<i>Mid-term evaluation deadline</i>
	Improvement of data transfer from Cytoscape to Neo4j
	Showcase network assembly
	UI improvement for Neo4j extension invocation
	Source documentation and tutorial write up
	Draft of application note for the App and Extensions
2014-08-22	<i>Firm 'pencils down'</i>

## Hurdles / Complications:

### *Algorithm Performance*

Neo4j has a set of algorithms implemented but most of them are not considered production quality (<http://components.neo4j.org/neo4j-graph-algo/snapshot/>).

While these could be used for the implementation of the Neo4j extensions, the algorithms might not be sufficiently efficient to serve the goal to analyze large networks. Implementing and testing other algorithms might be required and more time consuming than the straight forward use of predefined ones. This could increase the required time for the algorithm implementation significantly.