CHAPTER 3

3

Automating Bibliometric Analyses Using Taverna Scientific Workflows

Arzu Tugce Guler[1], Cathelijn J. F. Waaijer[2], Yassene Mohammed[1], Magnus Palmblad[1]

[1] Center for Proteomics and Metabolomics, Leiden University Medical Center, Leiden, The Netherlands

[2] Faculty of Social and Behavioural Sciences, Centre for Science and Technology Studies, Leiden University, The Netherlands

This chapter has been published in:
Journal of Informetrics (2016) 10, 830-841 doi: 10.1016/j.joi.2016.05.00

Abstract

Quantitative analysis of the scientific literature is a frequent task in bibliometrics. Several large online resources collect and disseminate bibliographic information, paving the way for broad analyses and statistics. The Europe PubMed Central (PMC) and its Web Services is one of these resources, providing a rich platform to retrieve information and metadata on scientific publications. However, a complete bibliometric analysis that involves gathering information and deriving statistics on an author, topic, or country is laborious when consuming Web Services on the command-line or using low level automation. In contrast, scientific workflow managers can integrate different types of software tools to automate multi-step processes. The Taverna workflow engine is a popular open-source scientific workflow manager, giving easy access to available Web Services. In this tutorial, we demonstrate how to design scientific workflows for bibliometric analyses in Taverna by integrating Europe PubMed Central Web Services and statistical analysis tools. To our knowledge, this is also the first time scientific workflow managers have been used to perform bibliometric analyses using these Web Services.

Introduction

As science becomes more data intensive, access to data and the process of generating meaningful information from them become the main vehicle in the scientific process. In this process, the primary challenge is moving from generated or retrieved data to information. As in most fields, typical bibliometric analysis workflows require several discrete steps, each employing different software tools. Frameworks that allow users to efficiently but easily connect data access points to information generation play a key role here. However, it is not always straightforward to use a generic framework or design custom workflows every time a new analysis protocol is to be implemented. In the absence of a framework, users have to manually connect the inputs and outputs of individual steps through the entire analysis. This risks introducing errors and makes analyses difficult to reproduce, especially for other researchers.

Scientific workflow managers integrate several processing units to automate a data analysis procedure. They are field-independent, so analysis on data from any field, including bibliometrics, can be automated. Scientific workflows typically have inputs and outputs, where series of operations are performed on the inputs in order to produce the outputs. Thus various atomic processing units can be assembled to produce an analysis protocol that can run without manual intervention[1]. On the other hand, reusability and reproducibility are also important for *in silico* experiments, facilitating collaboration and combining efforts. These are promoted by online scientific workflow repositories such as myExperiment[2]. However, deciphering the hierarchical composition of a workflow, its control and connections could be difficult in a larger-scale workflow[3]. Taking a modular approach and defining the scope of each module in the workflow eases this process. Most of the freely available scientific workflow managers have a graphical user interface that helps to visualize the overall protocol, both when designing and when executing the workflow. Galaxy[4], KNIME[5] and Taverna[6] are popular examples of such scientific workflow managers that also allow modular design. Automating an analysis consisting of several steps, such as in bibliometrics, using scientific workflow managers makes the process less laborious and decreases the risk of human errors. Scientific workflow managers follow a different paradigm than interactive software tools, such as the domain-specific (or perhaps domain-limited) BibExcel[7], Publish or Perish[8] and Sci2[9] though Sci2 certainly provides some aspects of the modularity and tool integration of the workflow managers.

We have previously presented how scientific workflows can be used to solve simple bibliometrics problems, using Taverna Workbench[10]. Like any other scientific workflow manager, Taverna enables the user to integrate different types of components. What makes Taverna very useful for bibliometrics is that it already provides custom support for a number of tools and services that are easily adopted for performing such analyses, *e.g.* R tools and XPath, Beanshell and WSDL services. The programming language R is primarily developed for statistical computing and visualization. Specific R plug-ins or packages expands its capabilities to machine learning, text mining and natural language processing[11],[12]. The XPath service is a user-friendly tool for creating XPath queries to parse XML documents by simply selecting nodes from an XML tree with a few mouse clicks. This is highly convenient, as most bibliometric databases can export information in XML format. For tabular formats, the Spreadsheet import service provides a similarly minimalistic tool for parsing tables. For general tasks, Beanshell services allow inclusion of scripts using a Java-like language. Last but not least, integrated support for Web Services allows Taverna workflows to directly communicate with remote databases using WSDL queries[13]. As most Web Services use XML as the preferred message format, the Taverna XPath service is typically used to parse the results returned from Web Service calls.

An important functional aspect of Taverna is that iterations over individual processes or parts of the workflows are done implicitly by list handling. This feature provides great flexibility if a process or a sub-workflow has more than one input port. The user can specify whether the inputs are subjected to a “cross product” (all list elements in one input against all list elements in the other input) or a “dot product” (element-wise), or for processors with more than two inputs a combination of both; all while being able to define the order and precedence of the workflow operations on these input lists. A core set of built-in features and services provides basic list handling, such as flattening, merging a list to a string and removing duplicates.

Here we present a tutorial on how to use Taverna to build workflows that interact with the Europe PubMed Central Web Services. In principle, Taverna could interact with any Web Service that provide a SOAP or RESTful interface. The reason we are demonstrating the integration of Web Services in Taverna using PubMed rather than Scopus® or Web of Science™[14] is that, among these three, PubMed is currently the only that provides a free Web Service interface. PubMed is also the most used bibliographic resource in the life sciences. In this tutorial, we show how to retrieve information using Web Services, how to parse this information, and how to use the various built-in Taverna services and processors to calculate and visualize the results. In principle, the same approach could be taken using other resources, provided that the user has access to them. We also made an example Taverna interface for connecting to the Thomson Reuters Web of Science™ Web Services and made this available on myExperiment (https://edu.nl/gcxpg). We have built and tested the workflows in Taverna Workbench Bioinformatics 2.5.0, but in principle the workflows should run in any flavor of Taverna Workbench version 2.4.0 or later. For instructions on how to download and install Taverna, see https://edu.nl/6nhtk. For Rshells to be executable in Taverna, R, RServe and required R packages must be installed and deployed[15].

Getting started: connecting to Europe PMC Web Services

Europe PubMed Central, or PMC (http://europepmc.org) is one of the leading databases for peer-reviewed life science literature, providing access to 30.4 million abstracts and 3.3 million full-text articles and metadata (December 14, 2015). The goal of Europe PMC is to "build open, full-text scientific literature resources and support innovation by engaging users, enabling contributors and integrating related research data"[16]. This is achieved by providing access through a user-friendly Web interface, FTP, and SOAP and RESTful Web Service APIs. Here we will use the latter from within Taverna workflows. This is done as follows. First, the Europe PMC SOAP-based Web Services are imported into Taverna using "Import new services" in the Design pane using the WSDL https://www.ebi.ac.uk/europepmc/webservices/soap?wsdl. The available Web Services should now be listed as available in Taverna services menu. The 55-page Europe PMC SOAP Web Service Reference Guide[17] describes all details of the API to these newly imported services. Although strongly recommended, it is not absolutely necessary to read the entire manual before starting to integrate Europe PMC Web Services from within Taverna. A Web Service component is simply added to a workflow by dragging it from the service menu and dropping it into the workflow whiteboard. To expose the component's inputs and outputs, we add XML splitters. These are found in the component Edit menu. For example, the *searchPublications* service currently has six input ports: *email*, *offset*, *pageSize*, *queryString*, *resultType* and *synonym*. Of these, only the *queryString* is mandatory. This string corresponds to what one would normally enter in the search field on the Europe PMC website. The *email* address registers the user with Europe PMC, the *pageSize* the number of entries to be retrieved in one page, the *offset* refers to which page of size *pageSize* to retrieve, *synonym* whether to expand the query using the MeSH and UniProt synonyms. The *resultType* is used to limit the retrieval to the data we want. It has three settings: *idlist*, *lite* and *core*. If we only want the PubMed IDs (PMIDs) for subsequent queries, *idlist* would be sufficient. The *lite* results contain key metadata such as the author list and basic bibliographic information, and *core* all metadata, including abstracts and full journal details. The full article, if in Europe PMC, is retrieved using another service, *getFulltextXML*. The workflow in Figure 3.1 illustrates the use of the *searchPublications* service with its input and output XML splitters. The workflow performs a single search similar to using a Web browser and the Europe PMC Website. The *results* is an XML tree with the first 100 results of the Europe PMC search defined by *query* where the number 100 is defined by *records_to_retrieve* constant. This workflow is available on myExperiment (https://edu.nl/wef8y). In Figure 3.1, all input and output ports of every workflow components are shown. In subsequent workflows, the ports details are hidden for simplicity. However, these can easily be displayed in Taverna workbench.

The Europe PMC results are retrieved in XML, and the extraction of the precise information we want are done by further XML output splitters or XPath services in Taverna. An XPath is a query written in the XPath language for selecting elements and attributes in an XML document. XPath allows postfix conditional statements within square brackets. For example, to restrict the results to PMIDs of cited papers (having a *citedByCount* larger than 0), the XPath */resultList/result[citedByCount>0]/pubYear* could be used on the output of the workflow in Figure 3.1 to retrieve the publication year (*pubYear*) for cited papers only. The XPath service in Taverna provides a configuration pane to automatically generate simple XPath expressions, which the

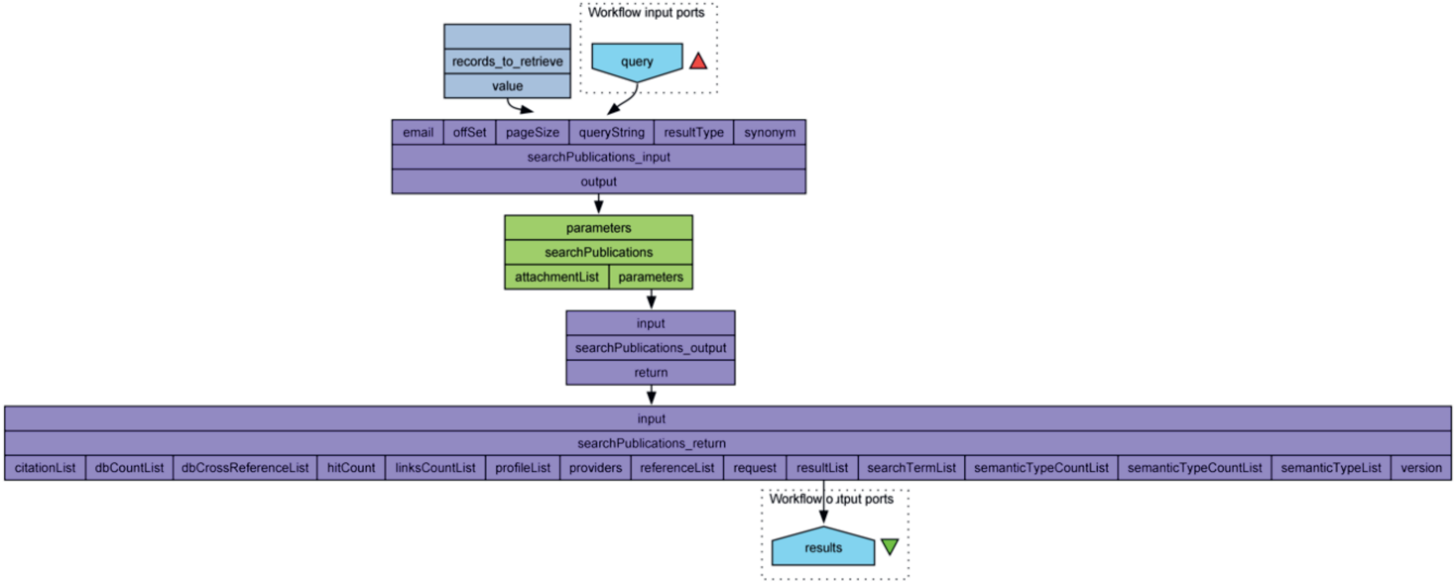


Figure 3.1. Basic workflow to access Europe PMC Web Services

user can then customize, for example by adding conditionals, or combining several expressions. The results of the Web Service and XML parsers can be passed to other workflow components either as text or as XML. A second workflow attaching an XPath statement to the workflow in Figure 3.1 is also available on myExperiment (https://edu.nl/pwyqv). The output of the workflow, after parsing the *searchPublications* output *resultList* with the XPath above, is a Taverna list of the publication years of the cited articles among the 100 first retrieved articles matching the search query *query*. As mentioned in the introduction, Taverna does iteration implicitly using lists. If a component for performing a certain task is given a list as input, the task will be performed on all elements in that list.

Publication records and citation networks

From these simple first steps, and using the same types of components, we will now construct more advanced workflows exploring the full power of the Europe PMC Web Services and Taverna. We do this using the notions of embedding and extensibility of scientific workflows. A simple workflow can be embedded in more complex workflows. Existing workflows, shared in the myExperiment community, can be accessed from the myExperiment pane in Taverna and modified or extended according to the user's needs.

The well-known Thomson Reuters Web of Science™ search provides a link to a "Citation Report" with two histograms, one over the number of published items in each year and one over the citations for these items in each year, based on the search results. In addition, the Citation Report provides simple statistics, such as average citations and the *h*-index for these search results (the *h*-index may be most relevant for a single author name search, but is calculated and reported for any set of publications). We can produce similar histograms based on the Europe PMC database using a Taverna workflow. For this, it is necessary to use two Web Services, *searchPublications* as before, and *getCitations* to get the publication year of papers citing the papers returned by the *searchPublications* query (for example on an author name). Figure 3.2 shows such a workflow, which is also available on myExperiment (https://edu.nl/uddhg). The workflow uses two Europe PMC Web Services:

searchPublications and *getCitations* to generate publication statistics in the form of a “citation report” for a particular author.

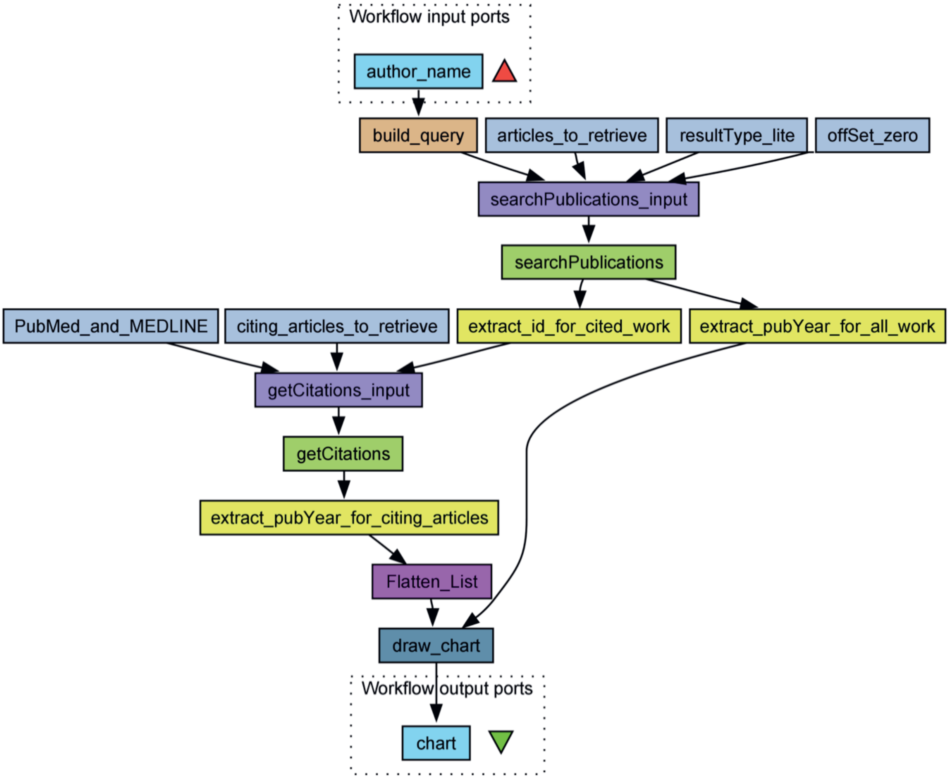


Figure 3.2. A workflow to generate simple statistics of the citation related to a specific author

The workflow in Figure 3.2 takes as input the full name of an author. This argument is passed to a Java BeanShell *build_query* that constructs the specific query "auth:\" " +author_name+"\" sort_date:y". Combined with a value 1,000 for the number of *articles_to_retrieve*, this will request the 1,000 most recent publications (sorted by date) for the author or authors matching *author_name*. The list of PMIDs returned by *searchPublications* is used as input to *getCitations*, which returns a list of lists of publication years for the papers citing the papers returned by *searchPublications*. In this workflow, XPaths are directly applied on the Web Services results. This skips the two XML output splitters and simplifies the visual appearance of the workflow. Whether to use output splitters and short XPaths, or longer XPaths directly on the

Web Service output, is mostly a matter of taste. The XPath extracting the *pubYear* for the citing articles produces a list of lists of publication years as output. In order to make a combined histogram over all citations to all papers from the author, we flatten this list of lists of publication years to a single list using the built-in *Flatten_List* local service. This single list of publication years is then passed to an Rshell component *draw_histogram* as data of the (semantic) type “integer vector”, as specified in the input port to this workflow component. The integer type in R exists to pass data to programs written in strongly typed languages that expects them, and so that integer data can be represented “exactly and compactly”[18]. In this workflow, the publication years could just as well be passed as “numeric” vectors. The Rshell is very simple and uses the hist() function[19] to generate the two histograms. For authors having a unique identifier, such as an ORCID, the *build_query* can be changed to "authorid:\"" + author_id + "\" sort_date:y".

An output of this workflow for the author “Jonas Bergquist” (Professor Jonas Bergquist, Department of Chemistry - Biomedical Centre, Uppsala University, Sweden) is shown in Figure 3.3. An extended version of this workflow is available on myExperiment (https://edu.nl/ptexf) that combines the publications and citations records in a two-dimensional heatmap showing the delay, increase and decrease of citations for papers over time. The workflow can easily be extended to accept a list of authors rather than a single author, generating either combined statistics or individual citation reports for each author in the list.

Suppose instead we are interested in who is cited by whom or citing the work of a particular researcher and how these authors in turn cite each other. To put it more simply: we would like to construct and visualize a co-citation network based on one researcher. Any network consists of multiple items (vertices or nodes) and their underlying relationships (edges). In the case of our co-citation network, the vertices are the single researcher and the authors cited by or citing this particular researcher. The edges are all the citations to and from the researchers in the co-citation network.

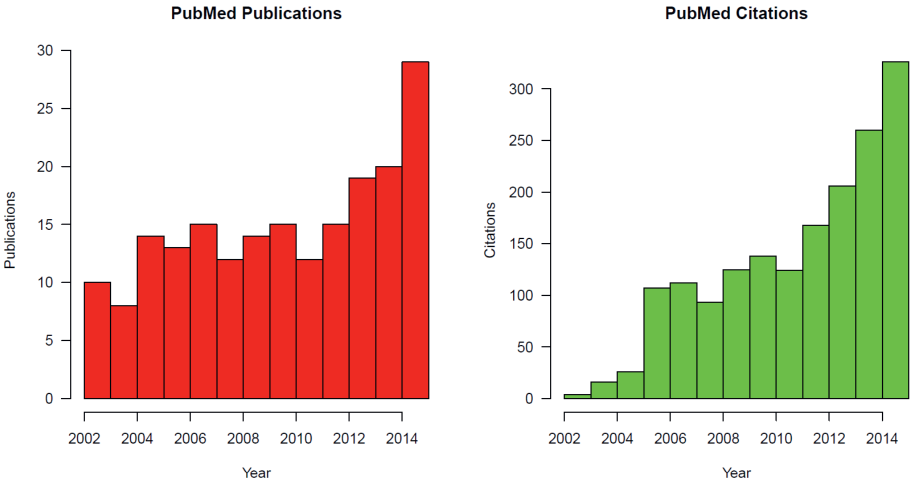


Figure 3.3. Citation report for an author ("Jonas Bergquist") generated by the workflow in Figure 3.2.

Constructing and visualizing this co-citation graph requires a slightly more elaborate workflow. Dividing up the task into smaller ones, we first look up all publications for the author using *searchPublications*. For each publication in the returned list, we then look up the references and citations in parallel using *getReferences* and *getCitations* respectively. These three requests returns all vertices in our co-citation graph, but does not retrieve citations, or edges, between papers by other authors. To retrieve these we combine all the vertices and call *getReferences* and *getCitations* again, once for each vertex. If the reference or citation is already represented by a vertex in the graph, we add a new edge to or from that vertex. To make the analysis more interesting, we can also have the workflow keep track of the author's own papers and self-citations. The best way to do this is by defining attributes to the edges and vertices in the co-citation graph. The workflow in Figure 3.4 does this by generating a description of the graph in Pajek[20] format using the BeanShell *combine_and_make_Pajek_file*. In most workflows, one would normally strive to use stream data between components or use simple tabular of XML file formats. When dealing with graphs, however, it is sensible to use a common format for defining graphs, such as GraphML[21], GML[22], LGL[23] or Pajek. The workflow in Figure 3.4 finds all papers citing and cited in articles published by an author, and all citations between them. The workflow generates a citation network graph that is captured by and displayed inside Taverna. The workflow also generates a Pajek file incorporating the information on self-citations using different edge attributes (color) and labels the vertices differently for the author (last name and publication year) than for the other vertices (PubMed ID).

The Pajek content created by *combine_and_make_Pajek_file* and written to file by *Write_Text_File* is read by the Rshell *draw_graph* using the igraph R package[24]. The igraph package contains functions for reading and writing graphs in several formats, including those mentioned here. The outputs of the workflow are a simplified graph in Sugiyama layout[25] created by igraph using simplify() and layout.sugiyama(), and the corresponding Pajek file created by write_graph() after simplification. A static but visual representation of the graph is captured by Taverna as well as written to a PDF file. To interactively explore and analyze the graph, the Pajek file can be opened in Pajek or a tool such as the VOSviewer[26], which are both tools for the analysis and visualization of (bibliometric) networks. The Pajek file created by the Taverna workflow (run November 30, 2015) was opened in VOSviewer 1.6.3, showing the largest set of connected items (3,782) out of the 3,851 vertices in this citation graph (Figure 3.5; can also be opened as an interactive Java application by clicking on the https://edu.nl/avgwc). The clustering was performed with clustering resolution 0.05 and minimum cluster size 50. The author's own papers are annotated with first author, last name and year, other papers with PMID. The publication record in the example above, including citations, can be visualized as a several highly interconnected and overlapping clusters, the largest of which (red) is on proteomics. In the center of this large cluster is a highly cited review by Aebersold and Mann on mass spectrometry-based proteomics (PMID 12634793)[27]. The dark blue cluster covers work in psychophysiology and neuroscience, excluding proteomics but including new methods for analysis of cerebrospinal fluid[28],[29]. In addition to this core of work in proteomics and neuroscience, we see a few protuberances representing collaborations with researchers in different disciplines, such example veterinary science applications[30] (light brown) and surface chemistry techniques[31],[32] (magenta). The full VOSviewer map is also included as supplemental information.

Another way to view the research topics of a particular author is to count words and noun phrases in the titles and abstracts, visualizing the results as a graph or tag cloud. A workflow using the *searchPublications* Web Service and the R packages tm[11] for text mining and wordcloud for visualization is also available on myExperiment (https://edu.nl/hgwkc).

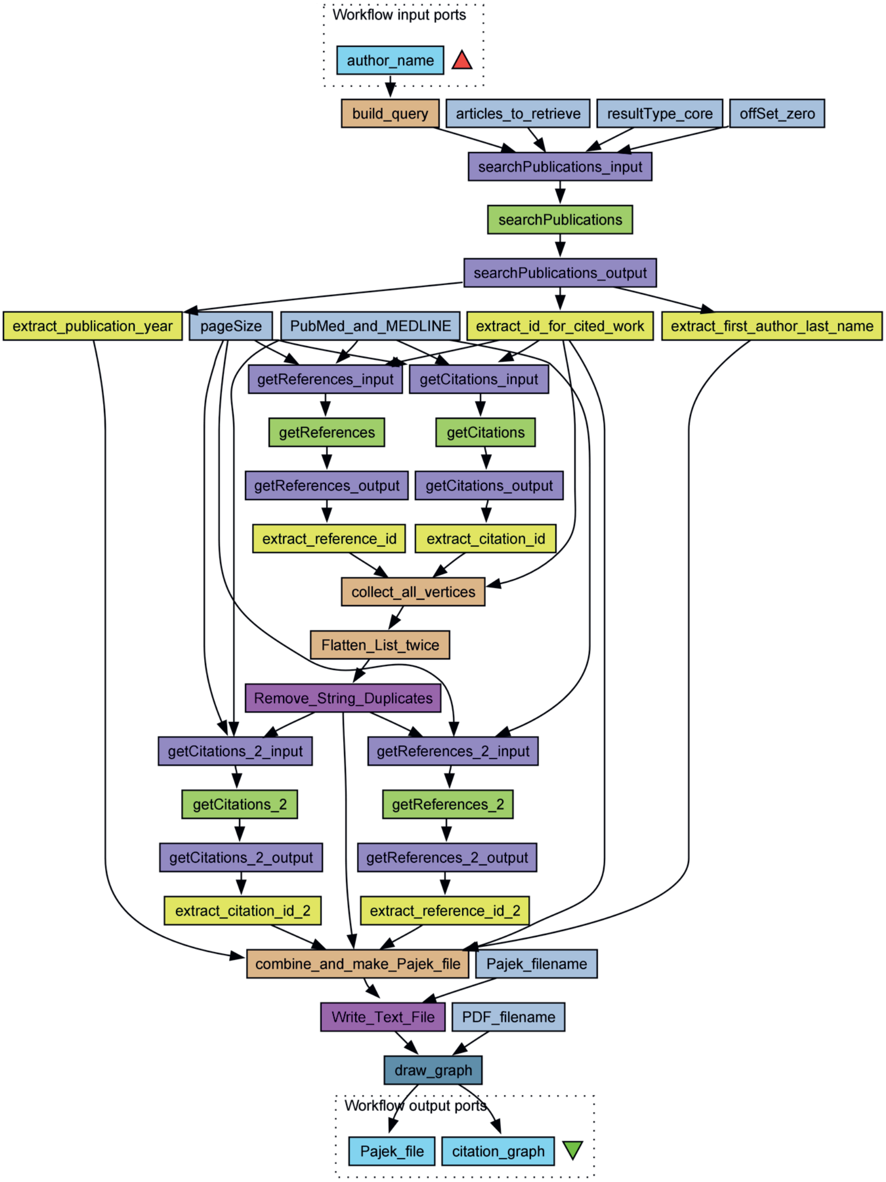
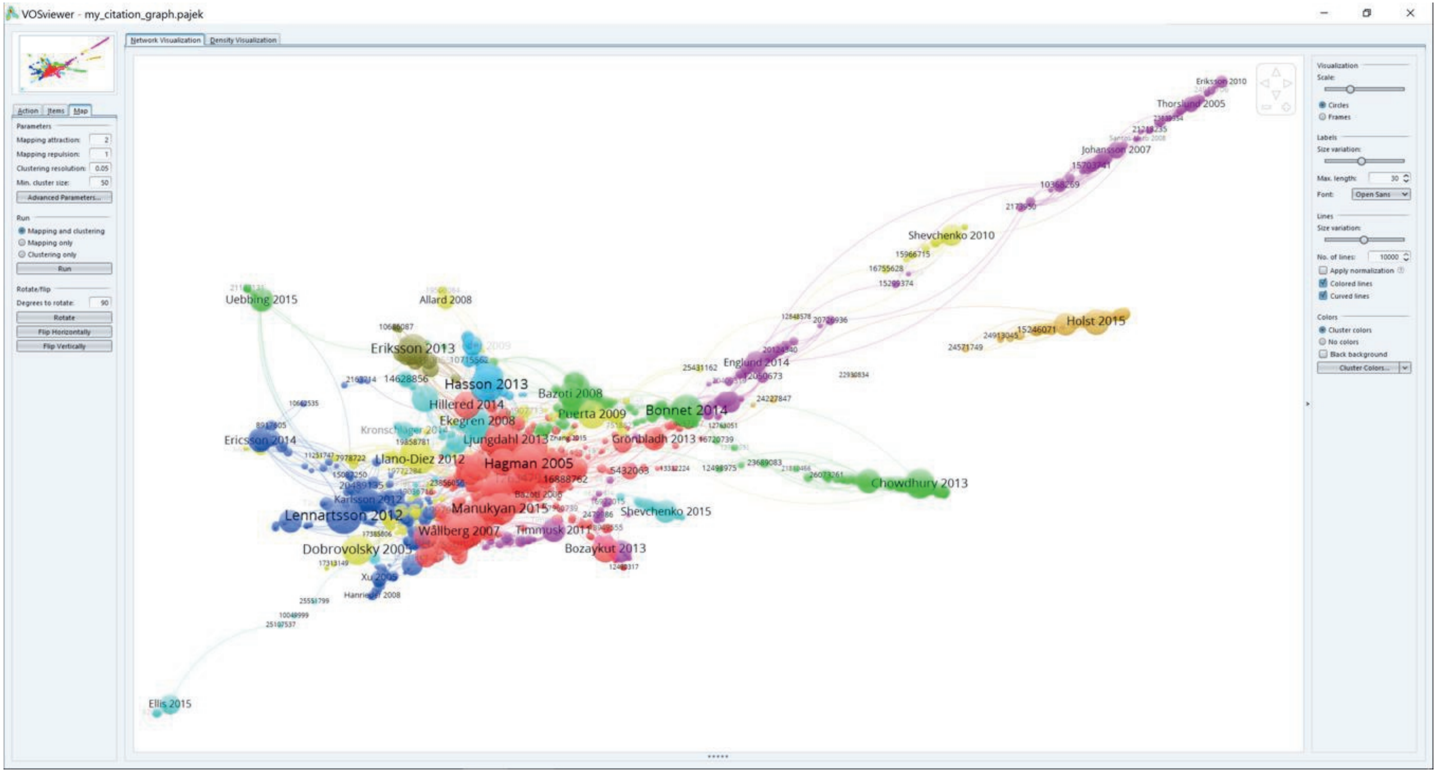


Figure 3.4. A scientific workflow for generating an author citation network



VOSviewer - my_citation_graph.pajek

Network Visualization | Density Visualization

Action | Items | Map

Parameters

- Mapping attraction: 2
- Mapping repulsion: 1
- Clustering resolution: 0.05
- Min. cluster size: 50

Advanced Parameters...

Run

- ● Mapping and clustering
- ○ Mapping only
- ○ Clustering only

Run

Rotate/flip

- Degrees to rotate: 90

Rotate

Flip Horizontally

Flip Vertically

Visualization

- Scale:
- ● Circles
- ○ Frames

Labels

- Size variation:
- Max. length: 30
- Font: Open Sans

Lines

- Size variation:
- No. of lines: 10000
- □ Apply normalization
- ☑ Colored lines
- ☑ Curved lines

Colors

- ● Cluster colors
- ○ No colors
- □ Black background

Cluster Colors...

Figure 3.5. Visualization of the citation network for “Jonas Bergquist” using the VOSviewer

Biomolecular interactions

Resources such as UniProt[33], IntAct[34] and the RCSB PDB[35] provide a wealth of curated information on proteins, their functions, interactions and structures. Importantly, they always cite the original source of the information, which most commonly is a peer-reviewed scientific publication. Europe PMC is also cross-referenced to these and several other databases. The *getDatabaseLinks* Web Service is used to access the UniProtKB, IntAct or PDB records associated with an article. This may seem like a trivial service, but is in fact a programmatic access that allows us to explore the scientific literature, not only for bibliometrics, but also to investigate what the publications are about, *e.g.*, the chemical compounds, genes, proteins, diseases or biological species. For example, consider a researcher who is interested in protein *P* and would like to find all proteins mentioned in connection with this protein in the scientific literature in a specific context. This context could be a molecular interaction, being part of the same protein complex, one protein activating the other, etcetera. Most researchers would use databases such as IntAct or UniProtKB to search for this information under the entry for protein *P*. But suppose the researcher wants to look a bit more broadly at what has been reported in the scientific literature but not yet annotated in UniProtKB, IntAct, or any other database as a specific type of protein-protein interaction. This can be accomplished using *searchPublications* and *getDatabaseLinks* in the same workflow (Figure 3.6). The workflow looks up the proteins in UniProtKB most frequently co-occurring in the literature with a query protein and in a specified context, *e.g.*, type of protein-protein interaction or disease. The workflow then builds a network with the proteins as nodes and the weights of the edges corresponding to the number of co-occurrences in the literature.

For simplicity, the input and output port splitters are embedded with the Web Services as Taverna components in the workflow in Figure 3.6. The workflow builds a query string from user provided input to search for a particular UniProt identifier in the context of a certain phrase appearing in the title or abstract. The list of retrieved article identifiers (PMIDs) is then passed to *getDatabaseLinks*, which, like *getCitations*, returns a list of lists of all UniProt identifiers co-occurring in those publications. These may be very few, or number in the thousands for large proteomics studies. In general, we would expect a co-occurrence in a publication with few linked UniProt IDs to be more relevant than a co-occurrence in a list of several thousand proteins. The results can be weighted using the returned *dbCountList*, or by limiting the number of UniProt IDs retrieved for each PMID to a small number to reduce the influence of proteomic studies. For example, a *searchPublications* query for UniProt ID P29083, or the Transcription factor IIE alpha subunit, with the phrase "complex" in the title or abstract returns a list of PMIDs for 9 publications (November 30, 2015). Passing this list of PMIDs to *getDatabaseLinks* and specifying a pageSize of 10 to retrieve at most 10 identifiers per PMID produces a list of lists with a total of 62 UniProt IDs, of which 53 are unique. The workflow in Figure 3.6 then counts the frequencies of these UniProt IDs and sort them in descending order using sort(table(UniProt_IDs), decreasing = TRUE) in the Rshell *count_frequencies*. The protein most frequently occurring in these lists is the query protein itself (5 occurrences). The runner-up is unsurprisingly UniProt ID P29084 or the beta subunit of the Transcription factor IIE with 3 appearances. Three other UniProt identifiers occur twice and the remainder once. Raising the *pageSize* limit to the maximum allowed 1,000 returns 2,948 identifiers, 2,550 of which are unique. Two sublists from two large-scale proteomics reports[36] reached the maximum of 1,000 UniProt IDs, reporting 2,932 and 5,159 identifiers respectively. The query protein is again in the top (13 occurrences), but the beta subunit is now only in 95th place, still with only 3 co-occurrences.

The network produced by the workflow in Figure 3.6 can be further analyzed in Cytoscape, a common tool for network visualization and analysis in bioinformatics[37]. The workflow output can be opened either directly as GML in Cytoscape. Figure 3.7 shows Cytoscape 3.3.0 with the output from the workflow in Figure 3.6 on Apolipoprotein A-I (UniProt ID P02647) and "complex" as before with the "Edge-weighted Spring Embedded" Cytoscape layout. The cluster of proteins associated with Apolipoprotein A-I was analyzed for enrichment of Gene Ontology biological processes by BiNGO 3.0.3[38]. In Figure 3.7, proteins frequently co-occurring with Apolipoprotein A-I and being involved in "macromolecular complex remodeling" (as well as "protein-lipid complex remodeling" and "plasma lipoprotein particle remodeling") are highlighted in yellow. Again, these results are not surprising given that Apolipoprotein A-I is the dominant protein component of high density lipoprotein

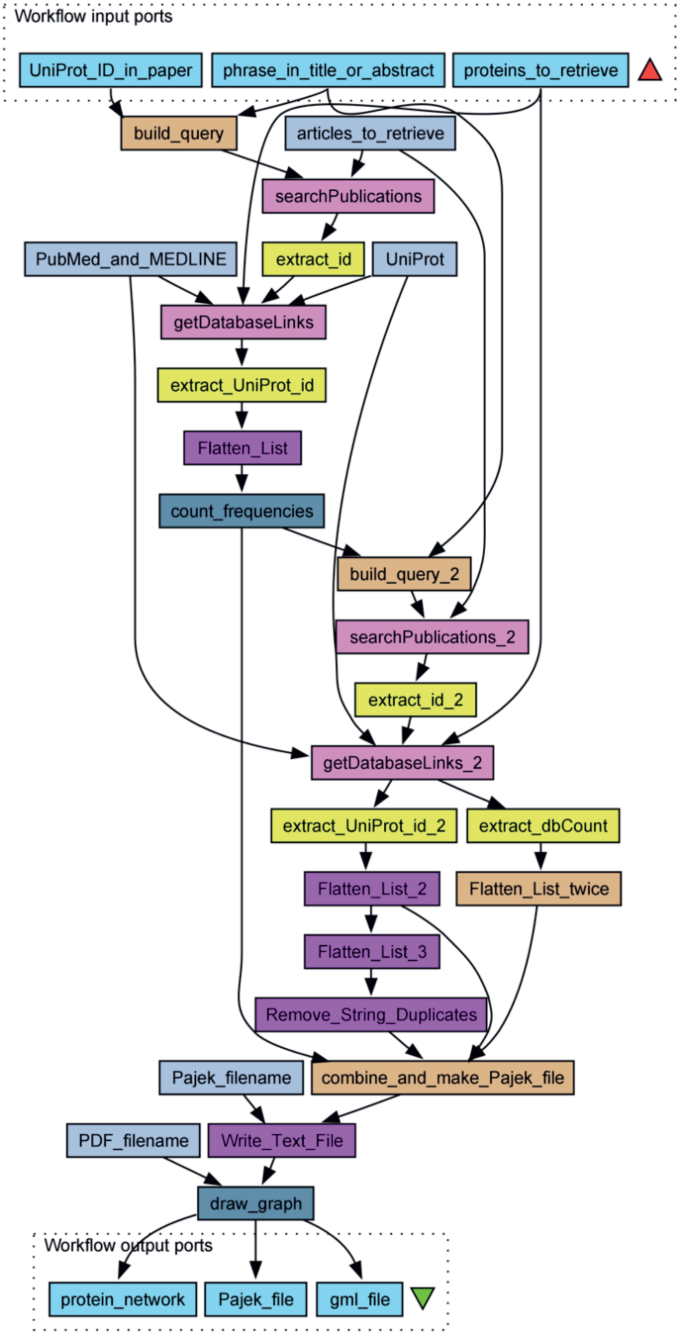
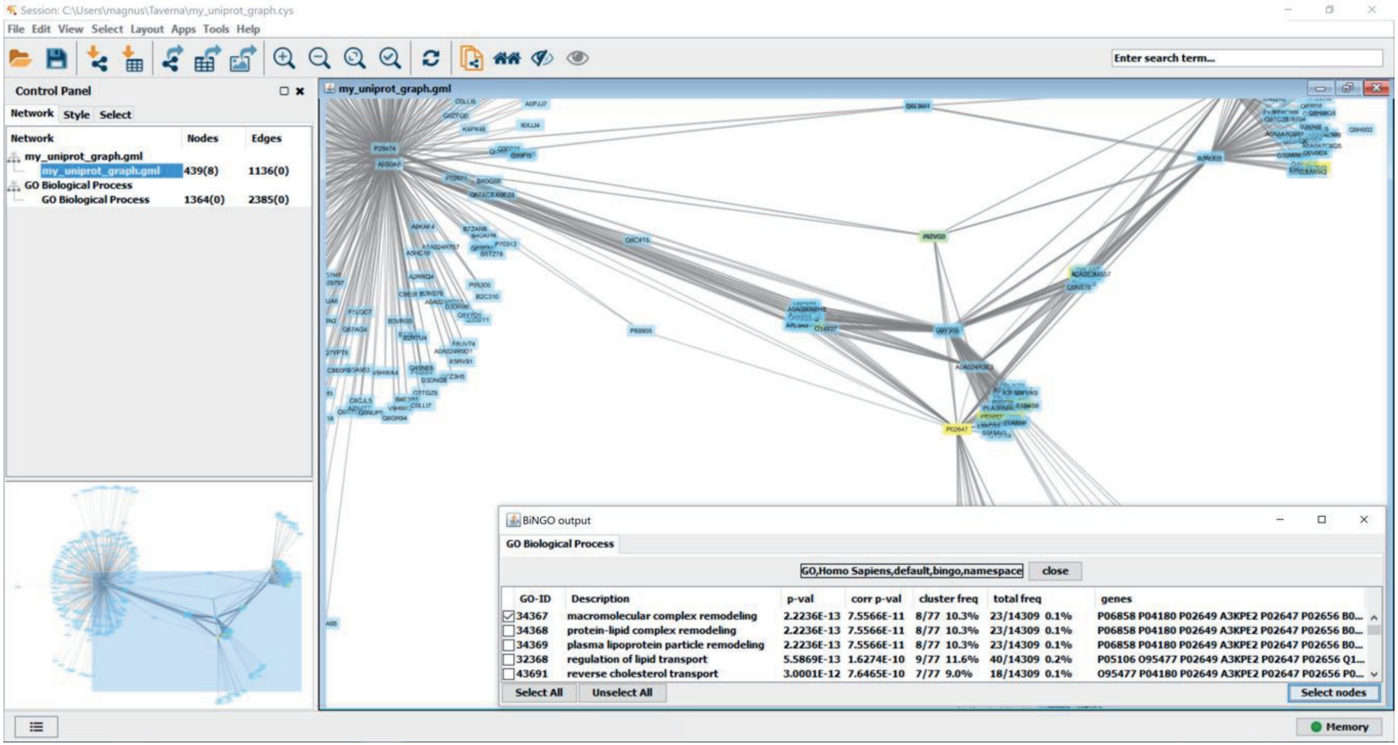


Figure 3.6. Workflow to generate a protein-protein network based on co-occurrence of UniProtKB accessions

(the "good cholesterol") in plasma.

The relevance of a co-occurrence (of genes or proteins) decreases with the number of co-occurring genes or proteins in a particular publication, as long lists are the results of broad proteomics studies rather than specific experiments probing the interactions of a particular protein or dissecting a specific protein-protein complex. But to take all proteins into account, a simple trick to retrieve an arbitrary number of results from any Europe PMC Web Service in a Taverna workflow is to supply the Web Service call with a sufficiently long list of *offSet* values, counting from zero. This list can be created inside the *build_query* BeanShell to hide the details from the workflow view. For example, adding a simple piece of code defining a new output *offSets* as int [] offSets = new int [] { 0, 1, 2, 3, 4 }; to *build_query* and connecting the output port *offSets* to the *offSet* input port of the Web Service will retrieve at most 5 pages of *pageSize* results from a Europe PMC Web Service (5,000 results with the maximum *pageSize* of 1,000). This approach is generally fine for literature on genes and proteins, but the Europe PMC Web Services, or Web Services generally, are not intended for piecemeal retrieval of millions of records (or the entire Europe PMC). This can better be done using the FTP access. UniProt is just one of many molecular databases linked with Europe PMC. The API to access these database links is the same for all molecular databases, all using the *getDatabaseLinks* Web Service.



Session: C:\Users\magnus\Taverna\my_uniprot_graph.cys

Network	Nodes	Edges
my_uniprot_graph.gml		
my_uniprot_graph.gml	439(8)	1136(0)
GO Biological Process		
GO Biological Process	1364(0)	2385(0)

GO,Homo Sapiens,default,bingo,namespace | close

GO-ID	Description	p-val	corr p-val	cluster freq	total freq	genes
☑ 34367	macromolecular complex remodeling	2.2236E-13	7.5566E-11	8/77 10.3%	23/14309 0.1%	P06858 P04180 P02649 A3KPE2 P02647 P02656 B0...
□ 34368	protein-lipid complex remodeling	2.2236E-13	7.5566E-11	8/77 10.3%	23/14309 0.1%	P06858 P04180 P02649 A3KPE2 P02647 P02656 B0...
□ 34369	plasma lipoprotein particle remodeling	2.2236E-13	7.5566E-11	8/77 10.3%	23/14309 0.1%	P06858 P04180 P02649 A3KPE2 P02647 P02656 B0...
□ 32368	regulation of lipid transport	5.5869E-13	1.6274E-10	9/77 11.6%	40/14309 0.2%	P05106 O95477 P02649 A3KPE2 P02647 P02656 Q1...
□ 43691	reverse cholesterol transport	3.0001E-12	7.6465E-10	7/77 9.0%	18/14309 0.1%	O95477 P04180 P02649 A3KPE2 P02647 P02656 P0...

Figure 3.7. Workflow results visualized in Cytoscape with BiNGO Gene Ontology Annotations

Discussion

Most scientific workflow managers interact with the user through intuitive and visual interfaces. Like all software, Taverna also has some peculiarities. For example, to execute R scripts, a connection to RServe must first be established. The advantage is that this can be on a remote server just as easily as on the local machine, something that may be useful for computationally demanding tasks. Most workflow managers also support multiple scripting languages and types of workflow components. While this brings a lot of flexibility and power, it also makes it more difficult for those not familiar with all of these languages or services to understand the details of heterogeneous workflows. In this tutorial, we have used only Beanshell, Rshell, XPath and WSDL components. For simplicity, and because they were not needed, we did not include any local tools or shells, JSONPaths or REST services in these workflows. However, we have also uploaded a REST equivalent of the Figure 3.2 workflow to myExperiment (https://edu.nl/hw6wy).

Taverna is flexible, and can be used to organize the running of locally installed software, arrange a series of R scripts, shuffle data between external Web Services, or any combination thereof. Unlike KNIME, Taverna is free both as in 'speech' (open source) and as in 'beer' (gratis). Taverna's emphasis on Web Services makes it a perfect partner to bibliometric resources such as Europe PMC. The Taverna codebase is in Java, whereas Galaxy's is in Python. This is also reflected in the default scripting language in the workflow managers (Java in Taverna, Python in Galaxy). The programming paradigm is shared between all workflow managers however, and there have even been efforts to enact Taverna workflows through Galaxy[39].

Documentation is important, in particular for sharing or collaborative development of workflows. All elements (processors, data links, inputs and outputs) in Taverna workflows can be annotated individually. These annotations follow the components when imported from one workflow to another and are found under the "Details" tab in the Service panel in Taverna Workbench. Components and connections only have a generic "Description" field whereas inputs and outputs also have an "Example" field that can be used as a default value when executing the workflow. The workflow itself has "Author" and "Title" fields, in addition to a description. Workflows can be shared on myExperiment, as we have done. When uploading a Taverna workflow, myExperiment attempts to extract workflow metadata such as title and description directly from these annotations. This works for Taverna, Galaxy and several other workflow managers. myExperiment also provide basic version control and allow users to comment on and discuss workflows. All this information can then be used to find workflows using a keyword search on the myExperiment website. Currently (May 2016), there are 3,752 workflows shared on myExperiment, so it is not practical to browse all workflows to find the one closest to what one needs (or the best starting point for one's workflow).

The examples in this paper do not use any nested structures which are otherwise common in large workflows. The legibility of complex workflows such as in Figure 3.4 may be improved by boxing the Web Service calls, hiding the details of the input/output splitters and XPaths and allowing the user to first grasp the overall logic of the workflow.

Conclusions

Bibliometric analyses often involve several steps that are carried out in different software tools. This requires much manual orchestration from one software tool to the other, which makes the process labor intensive and error prone. Scientific workflow managers, which are increasingly being used in other data intensive fields but have not yet seen widespread usage in bibliometrics, are useful tools to connect these different data retrieval and computational steps in an automated way. One such workflow manager is Taverna. In this study, we argue the direct support of Web Services, XML parsers and R in Taverna workflows make Taverna particularly useful for bibliometrics. With R comes direct access to a great number of powerful software packages such as igraph, wordcloud and rworldmap[40] for visualization, tm and openNLP for text mining and natural language processing. One limitation of using a scientific workflow manager such as Taverna, is that they are not meant for interactive exploration of large datasets. For this, it is more sensible to use domain-specific tools such as Pajek or VOSviewer for scientometrics, or Cytoscape for bioinformatics, as we demonstrated here.

In addition, software such as Taverna supplies repeatability and reusability to bibliometrics analyses. For example, all workflows discussed in this paper can be found in the myExperiment group for Bibliometrics and Scientometrics (https://edu.nl/4r8x3) for anyone to open and run from within Taverna, using the exactly the same or any other input parameters to define the query. Other Taverna workflows in the Bibliometrics and Scientometrics group on myExperiment use rworldmap to map differences in the geographic distribution of author affiliations between two PubMed search results. Such workflows can for example look at geographical patterns of research on particular diseases, or geographical bias in different journals.

The bibliometric analyses illustrated in this study are exemplative of the kind of analyses we do in our research and here focused on the use of the Europe PMC Web Services. In a previous paper[10] we have used Taverna for other types of bibliometric analyses, such as geographic and temporal analyses of publication patterns, word usage and co-citation analysis. Here we have shown how to access Europe PMC through a Web Service API and how to perform bibliometric analyses using the Taverna scientific workflow manager, but, more importantly, how to combine the two.

Acknowledgments

The authors acknowledge Dr. Nees Jan van Eck at CWTS for advice on the VOSviewer and Professor Jonas Bergquist, Uppsala University for allowing us to use his publication record to demonstrate the workflows. The authors also wish to thank Michele Ide-Smith and Vid Vartak at the European Bioinformatics Institute for answering our questions regarding Europe PMC data.

References

1. de Bruin, J. S., Deelder, A. M. & Palmblad, M. Scientific Workflow Management in Proteomics. *Mol. Cell. Proteomics* **11**, M111.010595 (2012).
2. Goble, C. A. *et al.* myExperiment: A repository and social network for the sharing of bioinformatics workflows. *Nucleic Acids Res.* **38**, W677–W682 (2010).
3. Lu, S. & Zhang, J. Collaborative scientific workflows. in *2009 IEEE Int. Conf. Web Serv.* 527–534 (2009).
4. Goecks, J., Nekrutenko, A. & Taylor, J. Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. *Genome Biol.* **11**, R86 (2010).
5. Berthold, M. R. *et al.* KNIME-the Konstanz information miner. in *Data Analysis, Machine Learning and Applications. Studies in Classification, Data Analysis, and Knowledge Organization* (eds. Preisach C., Burkhardt H., Schmidt-Thieme L., Decker R.) (Springer Berlin, Heidelberg 2008).
6. Oinn, T. *et al.* Taverna: A tool for the composition and enactment of bioinformatics workflows. *Bioinformatics* **20**, 3045–3054 (2004).
7. Persson, O. BibExcel. https://homepage.univie.ac.at/juan.gorraiz/bibexcel/ (2016).
8. Harzing, A. W. Publish or Perish. https://harzing.com/resources/publish-or-perish (2007).
9. Sci2 Team. Science of Science (Sci2) Tool. *Indiana University and SciTech Strategies* https://sci2.cns.iu.edu (2009).
10. Guler, A. T., Waaijer, C. J. F. & Palmblad, M. Scientific workflows for bibliometrics. *Scientometrics* **107**, 385–398 (2016).
11. Feinerer, I., Hornik, K. & Meyer, D. Text Mining Infrastructure in R. *J. Stat. Softw.* **25**, 1–54 (2008).
12. Hornik, K. openNLP: Apache OpenNLP Tools Interface. *R package version 0.2-4* https://cran.r-project.org/package=openNLP (2015).
13. Wolstencroft, K. *et al.* The Taverna workflow suite: designing and executing workflows of Web Services on the desktop, web or in the cloud. *Nucleic Acids Res.* **41**, W557–W561 (2013).
14. Falagas, M. E., Pitsouni, E. I., Malietzis, G. A. & Pappas, G. Comparison of PubMed, Scopus, Web of Science, and Google Scholar : strengths and weaknesses. *FASEB J.* **22**, 338–342 (2008).
15. Wassink, I. *et al.* Using R in Taverna: RShell v1.2. *BMC Res. Notes* **2**, 138 (2009).

16. Gou, Y. *et al.* Europe PMC: A full-text literature database for the life sciences and platform for innovation. *Nucleic Acids Res.* **43**, D1042–D1048 (2015).

17. Europe PMC. EBI Europe PMC SOAP Web Service 4.4 Reference Guide. http://europepmc.org/docs/EBI_Europe_PMC_Web_Service_Reference.pdf (2015).

18. Becker, R A, Chambers, J. M. *The New S Language: a Programming Environment for Data Analysis and Graphics*. (Wadsworth & Brooks/Cole, 1988).

19. Venables, W. N. & Ripley, B. D. Modern Applied Statistics with S. *Technometrics.* **45,** 111 (2003).

20. Batagelj, V. & Mrvar, A. Pajek – program for large network analysis. *Connections* **21**, 47–57 (1998).

21. Eiglsperger, M., Brandes, U., Lerner, J. & Pich, C. Graph Markup Language (GraphML). in *Handbook of Graph Drawing and Visualization* 517–541 (Chapman & Hall/CRC, 2013).

22. Himsolt, M. *GML: A portable Graph File Format*. *Technical report* . University of Passau, 94030 Passau, Germany (1997).

23. Adai, A. T., Date, S. V., Wieland, S. & Marcotte, E. M. LGL: Creating a map of protein function with an algorithm for visualizing very large biological networks. *J. Mol. Biol.* **340**, 179–190 (2004).

24. Csárdi, G. & Nepusz, T. The igraph software package for complex network research. *InterJournal Complex Syst.* **1695**, 1–9 (2006).

25. Sugiyama, K., Tagawa, S. & Toda, M. Methods for Visual Understanding of Hierarchical System Structures. *IEEE Trans. Syst. Man Cybern.* **SMC-11**, 109–125 (1981).

26. van Eck, N. J. & Waltman, L. Software survey: VOSviewer, a computer program for bibliometric mapping. *Scientometrics* **84**, 523–538 (2010).

27. Aebersold, R. & Mann, M. Mass spectrometry-based proteomics. *Nature* **422**, 198–207 (2003).

28. Wetterhall, M., Zuberovic, A., Hanrieder, J. & Bergquist, J. Assessment of the partitioning capacity of high abundant proteins in human cerebrospinal fluid using affinity and immunoaffinity subtraction spin columns. *J. Chromatogr. B* **878**, 1519–1530 (2010).

29. Dahlin, A. P. *et al.* Multiplexed quantification of proteins adsorbed to surface-modified and non-modified microdialysis membranes. *Anal. Bioanal. Chem.* **402**, 2057–2067 (2012).

30. Holst, B. S., Kushnir, M. M. & Bergquist, J. Liquid chromatography-tandem mass spectrometry (LC-MS/MS) for analysis of endogenous steroids in the luteal phase and early pregnancy in dogs: A pilot study. *Vet. Clin. Pathol.* **44**, 552–558 (2015).

31. Thorslund, S., Lindberg, P., Andrén, P. E., Nikolajeff, F. & Bergquist, J. Electrokinetic-driven microfluidic system in poly(dimethylsiloxane) for mass spectrometry detection integrating sample injection, capillary electrophoresis, and electrospray emitter on-chip. *Electrophoresis* **26**, 4674–4683 (2005).

32. Eriksson, A. *et al.* Optimized protocol for on-target phosphopeptide enrichment prior to matrix-assisted laser desorption-ionization mass spectrometry using mesoporous titanium dioxide. *Anal. Chem.* **82**, 4577–4583 (2010).

33. Magrane, M. & UniProt Consortium. UniProt Knowledgebase: A hub of integrated protein data. *Database* **2011**, 1–13 (2011).

34. Orchard, S. *et al.* The MIntAct project - IntAct as a common curation platform for 11 molecular interaction databases. *Nucleic Acids Res.* **42**, D358–D363 (2014).

35. Berman, H. M. *et al.* The Protein Data Bank. *Nucleic Acids Res.* **28**, 235–242 (2000).

36. Choudhary, C. *et al.* Lysine acetylation targets protein complexes and co-regulates major cellular functions. *Science* **325**, 834–840 (2009).

37. Shannon, P. *et al.* Cytoscape : A Software Environment for Integrated Models of Biomolecular Interaction Networks. *Genome Res.* **13**, 2498–2504 (2003).

38. Maere, S., Heymans, K. & Kuiper, M. Systems biology BiNGO : a Cytoscape plugin to assess overrepresentation of Gene Ontology categories in Biological Networks. **21**, 3448–3449 (2005).

39. Karasavvas, K. *et al.* Opening new gateways to workflows for life scientists. *Stud. Health Technol. Inform.* **175**, 131–141 (2012).

40. South, A. rworldmap : A New R package for Mapping Global Data. *R J.* **3**, 35–43 (2011)