

Open Data and Open Code for Big Science of Science Studies

Robert P. Light, David E. Polley, and Katy Börner

Abstract

Historically, science of science studies were/are performed by single investigators or small teams. As the size and complexity of data sets and analyses scales up, a “Big Science” approach (Price, 1963) is required that exploits the expertise and resources of interdisciplinary teams spanning academic, government, and industry boundaries. Big science of science studies utilize “big data”, i.e., large, complex, diverse, longitudinal, and/or distributed datasets that might be owned by different stakeholders. They apply a systems science approach to uncover hidden patterns, bursts of activity, correlations, and laws. They make available open data and open code in support of replication of results, iterative refinement of approaches and tools, and education. This paper introduces a database-tool infrastructure that was designed to support big science of science studies. The open access Scholarly Database (SDB) (<http://sdb.cns.iu.edu>) provides easy access to 26 million paper, patent, grant, and clinical trial records. The open source Science of Science (Sci2) tool (<http://sci2.cns.iu.edu>) supports temporal, geospatial, topical, and network studies. The scalability of the infrastructure is examined. Results show that temporal analyses scale linearly with the number of records and file size, while the geospatial algorithm showed quadratic growth. The number of edges rather than nodes determined performance for network based algorithms.

Introduction & Related Work

Many science of science studies use heterogeneous datasets and advanced data mining and visualization algorithms advance our understanding of the structure and dynamics of science.

The quality of results depends on the quality and coverage of the data used. Data cleaning and preprocessing can easily consume 80 percent or more of the overall project effort and budget. As the number of data records grows, different types of tools and expertise are required to handle the data. MS Excel can load a maximum of 1,048,576 rows of data by 16,384 columns per sheet. MS Access file sizes cap at 2 gigabytes, including indices, forms, and macros along with the data. Larger datasets need to be stored in a database designed with scalability in mind. As the diversity of datasets increases, the structures of different datasets need to be aligned. As data covers more and more years, dealing with format changes becomes necessary. Many studies require extensive preprocessing and augmentation of the data, such as identification of unique records or record values, geocoding of records in preparation for geospatial analysis, or the extraction of networks for network studies. For many researchers, the effort to compile ready-to-analyze-and-visualize data is extremely time consuming and challenging and sometimes simply insurmountable.

Many datasets relevant for science of science studies, e.g., papers, patents, grants, clinical trials, are freely available by different providers. However, they are stored in separate silos with diverse interfaces of varying usability that deliver data in many different formats. Research projects seeking to use one or many of these data sources face major data access, integration, and unification challenges. Indiana University's Scholarly Database (SDB), originally launched in 2005, makes over 26 million scholarly records freely available via a unified interface and in data formats that are easy to use and well documented. In the last four years, SDB has answered thousands of queries and delivered millions of records to users around the globe. The 2012 update to the SDB improves the quality of data offered and integrates new humanities and clinical trial datasets.

Equipped with high quality, high coverage data in standard data formats, tools that scale in terms of the number of records that can be read and processed are needed to truly make sense of big data (Robertson, Ebert, Eick et al., 2009). While most tools work well for micro and meso level studies (up to 100,000 records), few scale to macro level big-data studies with millions or even billions of records. Another type of scalability relates to the ease of usage and ease of interpretation of big data visualizations. How to best communicate temporal trends or burst of activity over a 100 year time span? How to depict the geospatial location of millions of records in a scalable fashion? Can the topical evolution of massive document datasets be communicated to a general audience? Most visualizations of million node networks resemble illegible spaghetti balls—do advanced network analysis algorithms scale and help to derive insights?

Frequently, different types of analysis have to be applied to truly understand a natural, social, or technological system. Examples are temporal studies that answer WHEN questions, geospatial studies that answer WHERE questions and draw heavily on research in cartography, topical studies that use linguistic analysis to answer WHAT questions, and network studies that employ algorithms and techniques developed in social sciences, physics, information science and other domains to answer WITH WHOM questions. However, most existing systems support only one general type of analysis and visualization and many require programming skills. For example, four of the top 20 data visualization tools listed by .net in September of 2012 support charts and graphs while six support geospatial maps exclusively (Suda, 2012). Only the D3 (Data-Driven Documents) and Raphaël JavaScript libraries, the Google Chart API, and R support a larger array of charts, graphs, and maps yet all three require programming or scripting skills that most users do not possess. Excel might be the only tool on the list that can be used by a large number of non-programmers. A listing of tools commonly used in science of science studies can

be found at

<http://sci2.wiki.cns.iu.edu/display/SCI2TUTORIAL/8.2+Network+Analysis+and+Other+Tools>

but most support a very limited range of workflows (Cobo, López-Herrera, Herrera-Viedma et al., 2011).

This paper presents a database-tool infrastructure that applies a divide-and-conquer approach to support big science of science studies. It combines an online database supporting bulk download of data in easy to process formats with a plug-and-play tool to read, clean, interlink, mine, and visualize data using easy to manipulate graphical user interfaces.

The remaining paper is organized as follows: The next two sections present the database and tool functionalities. Subsequently, we test and discuss their scalability. We conclude the paper with a discussion of the presented work and an outlook to future work.

The Scholarly Database (SDB)

The Scholarly Database was created in 2005 to provide researchers and practitioners easy access to various datasets offered by different publishers and agencies (LaRowe, Ambre, Burgoon et al., 2009). The Scholarly Database is implemented using PostgreSQL 8.4, a free and open source relational database management system. Since the introduction of version 8.1, PostgreSQL developers have been focused on improving the scalable performance of the system and this software is now employed by many companies to provide large-scale data solutions, including Yahoo!, Sony Online and Skype. Today, the Scholarly Database provides easy access to paper, patent, grant, and clinical trials records authored by 13.8 million people in 208 countries (some, such as Yugoslavia, no longer in existence), interlinked by 58 million patent citation links, and over 2.5 million links connecting grant awards to publications and patents. As of November

2012, the SDB features over 26 million records from MEDLINE (19,039,860 records spanning from 1865-2010), USPTO patents (4,178,196, 1976-2010), NIH awards (2,490,837, 1972-2012), NSF awards (453,687, 1952-2010), NEH awards (47,197, 1970-2012) Clinical Trials (119,144, 1900-2012).

Unique features of SDB comprise:

- **Open Access:** The SDB is composed entirely of open data so there are no copyright or proprietary issues for the researcher to contend with in its use. Data is provided to researchers free of charge.
- **Ease of Use:** Simple user interfaces provide a one-stop data access experience making it possible for researchers to focus on answering their questions, rather than spending much time on parsing, searching, and formatting data.
- **Federated Search:** By aggregating the data into a single environment, SDB offers a federated search environment powered by a Solr core. Users can search one, some, or all of the available datasets over some or all years using the same set of terms and get a combined set of results that are ranked by relevance.
- **Bulk Download:** Most databases do not support downloads and those that do only permit access to a limited number of records. SDB supports bulk download of data records; data linkages—co-author, patent citations, grant-paper, grant-patent; burst analysis files. Users are granted a base number of downloads by default to prevent abuse of the system, but this number can be extended by request without charge.
- **Unified File Formats:** SDB source data comes in different file formats. NIH funding data is stored in flat files; clinical trials are offered in XML, while patents come in a variety of formats, depending on the year. Old patents come in a fixed width data format while

newer patents are provided in XML. Much time and effort was spent to normalize this data into easy-to-use file formats, e.g., comma-delimited tables for use in spreadsheet programs and common graph formats for network analysis and visualization.

- **Well-Documented:** SDB publishes data dictionaries for every dataset offered. Information on data provenance, table structure, data types, and individual field comments are available. In addition, the SDB offers a set of small sample files, giving researchers an easily usable test-bed for working out their algorithms before committing to analysis of a larger set.

The SDB Wiki (<http://sdb.wiki.cns.iu.edu>) provides more information including a user guide, information on each dataset, and release notes.

The Science of Science (Sci2) Tool

The Science of Science (Sci2) tool is a modular toolset specifically designed for the study of science. It supports the temporal, geospatial, topical, and network analysis and visualization of scholarly datasets at the micro(individual), meso (local), and macro (global) levels, see screenshot in Figure 1, general workflow in Figure 2 and specific workflows discussed in the scalability tests section.

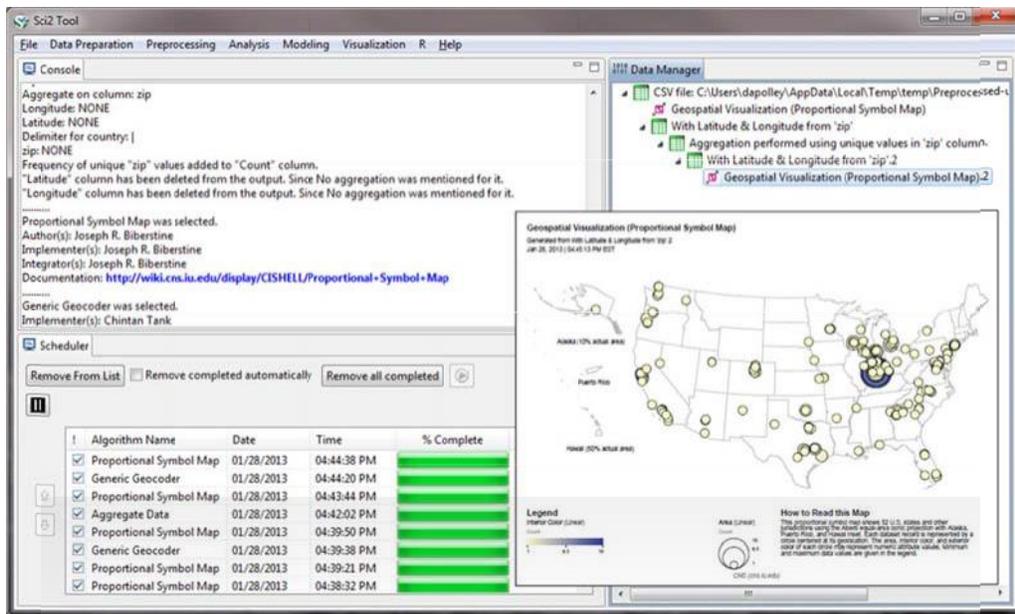


Figure 1: Sci2 tool user interface with proportional symbol map visualization.

The tool's OSGi/CIShell core architecture makes it possible for domain scientists to contribute new algorithms written in a variety of programming languages using a plug-and-play microscope approach (Börner, 2011).

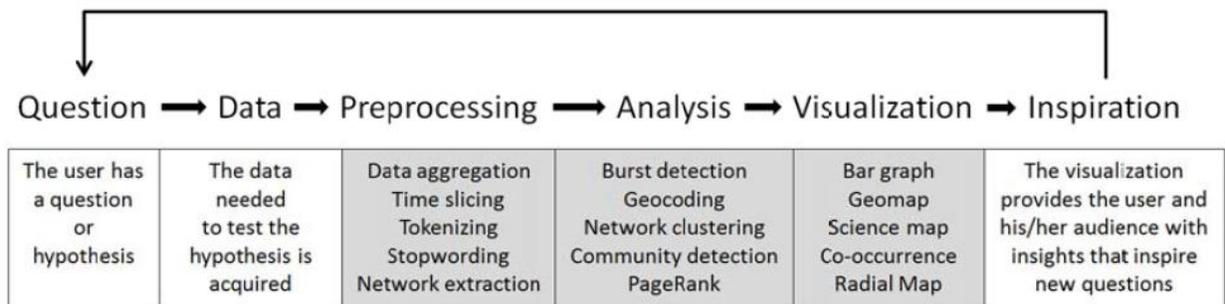


Figure 2: General Sci2-based visualization creation workflow (tool-specific tasks in gray).

As of November 2012, the Sci2 tool has 171 algorithms, 112 of which are visible to the user (see Figure 3) written in Java, C, C++, and Fortran. In addition, a number of tools (Gnuplot, Guess, and Cytoscape) were implemented as plugins and bridges to R and to Gephi were created,

allowing the seamless use of different tools. The Sci2 user interface and sample map is shown in Figure 1.

Category	Algorithms	Examples
Acquisition	5	Google Citation User ID Search Aglorithm
Data Preparation	13	Extract Co-Occurrence Network
Preprocessing	22	Slice Table by Time, Extract ZIP Code
Analysis	47	K-Nearest Neighbor, Burst Detection
Modeling	4	Watts-Strogatz Small World, TARL
R	4	Create an R Instance, Send a Table to R
Visualization	17	Choropleth Map, Bipartite Network Graph
Total	112	

Figure 3: Sci2 algorithm summary tables.

Unique features of Sci2 comprise:

- **Open Source:** Anybody can examine the source code and advance it.
- **Extensive use of well-defined reference systems:** To improve readability and to support interpretation, Sci2 uses a number of carefully designed reference systems, see Figure 4. Each comes with a title, legend, and a brief “How to read this visualization” section projections.
- **Interactivity:** While visualizations of small datasets can be explored interactively, visualizations of big data are rendered into Postscript files that can be converted to pdf files and examined using pan and zoom as well as filtered, e.g., by searching for specific text in the display.
- **Workflows:** All user actions are recorded in a log file to ensure proper documentation and easy replicability of workflows that might comprise 15-20 analysis and visualization algorithms with a range of parameter settings.

- Online documentation: All Sci2 plugins as well as major workflows are documented in the Sci2 Wiki (<http://sci2.wiki.cns.iu.edu>) together with release notes.

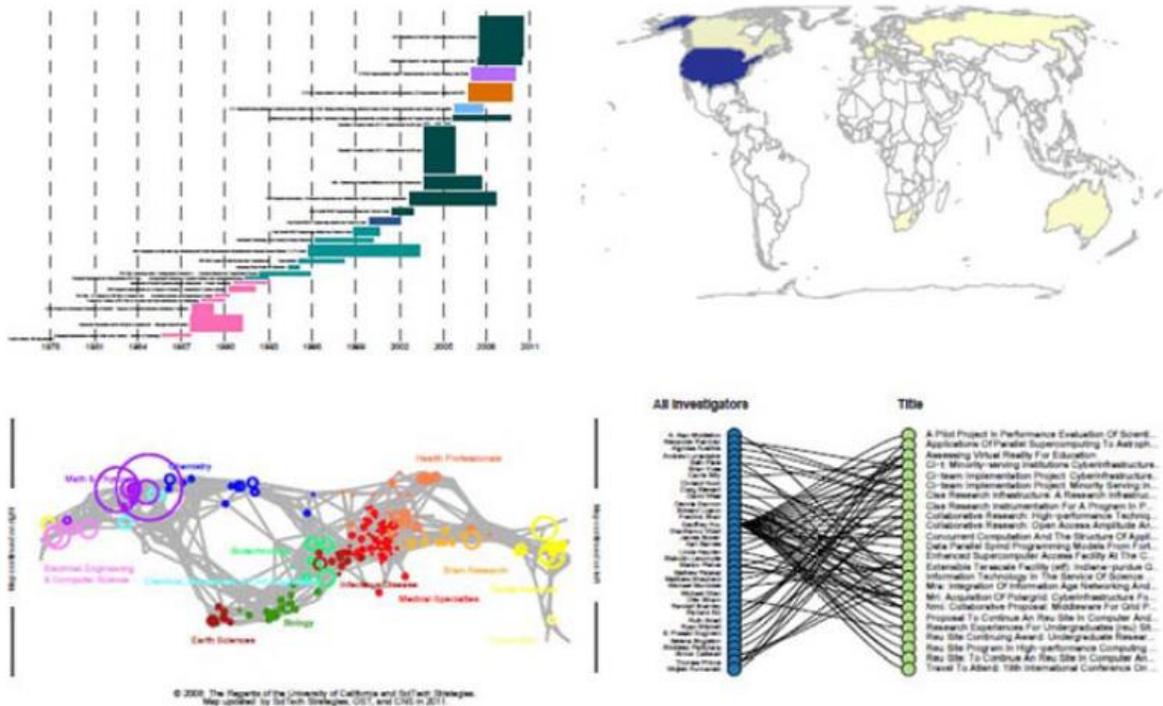


Figure 4: Exemplary reference systems supported by Sci2 including Temporal Bar Graph (top, left), Choropleth map (top, right), UCSD science map (bottom, left), bimodal network visualization (bottom, right) Full versions available at <http://wiki.cns.iu.edu/display/SCI2TUTORIAL/1+Introduction>.

Scalability Tests

To demonstrate the scalability of the database and tool, tests were performed using synthetic datasets with pre-defined properties generated in Python and datasets retrieved from the Scholarly Database. All four types of analysis supported by Sci2 were tested: temporal analysis, geospatial analysis, topical analysis, and network analysis. Initially, we identified workflows indicative of these four main types of analysis. From there, we broke down each workflow into the specific steps (algorithms) involved in the workflow, starting with

loading the data and ending in visualization. For each algorithm, e.g., data reader, analysis, visualization, we measured (in seconds) the length of time it took for an algorithm to finish processing. We considered the start of the algorithm to be the point at which the user inputs his or her parameters (where applicable) and then executes the algorithm. We considered all algorithms to be finished when the associated data files appeared in the Data Manager and were displayed as complete in the Scheduler. For each test, we calculated the average for 10 trials. Between trials, we closed down Sci2 in order to minimize any adverse effects of residual memory. Tests were performed on a common system: an Intel(R) Core(TM) Duo CPU E8400 3.00GHz processor and 4.0GB of memory running a 64bit version of Windows 7 and a 32bit version of Java 7. Memory allotted to Sci2 was extended to 1500 MB.

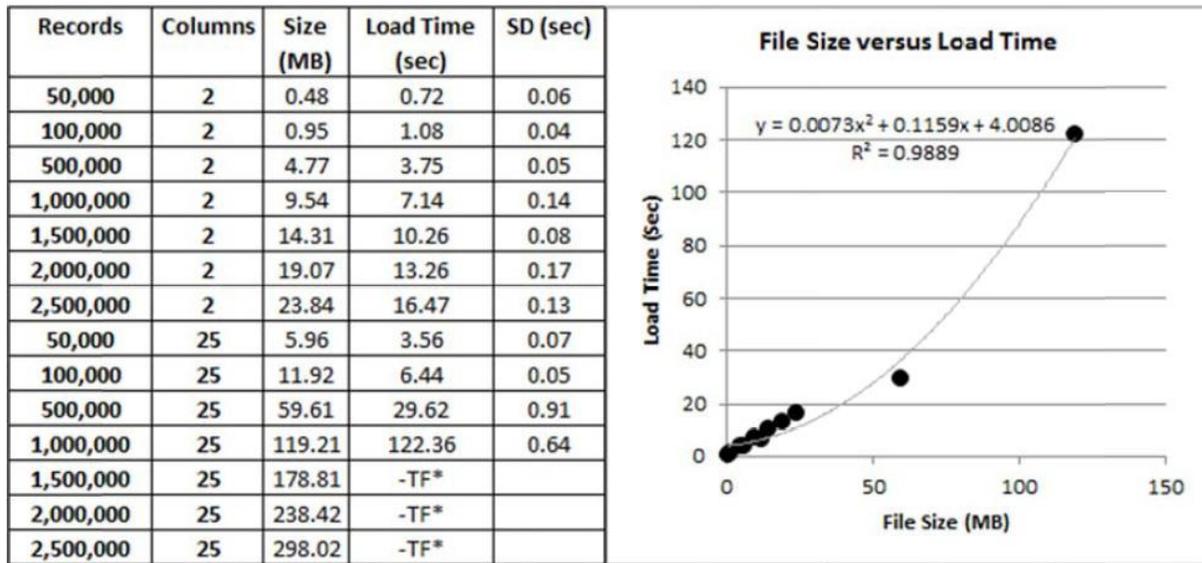


Figure 5: Comparison of load times, measured in seconds, across standardized datasets, tabulated (left) and plotted with quadratic regression line (right).

File Loading

Synthetic data was used to measure how file loading times vary in terms of number of records and length of individual record in bytes. Two series of datasets were generated, one with only two rows, a small integer, and a short string and one with 25 rows, a small integer and 24 short strings, each with increasing numbers of rows. Average loading times over ten trials are given in Figure 5. The three largest datasets did not load but returned a Java heap space error (-TF*). At first glance, there seems to exist a direct relationship between file size and loading time ($R^2 = 0.9384$), a closer look at the plot of size versus time reveals that a quadratic regression line has a noticeably better fit ($R^2=0.9889$). This is likely a result of the tool having to devote resources to file management that would otherwise be available for completing functions more efficiently.

- Next, SDB data prepared for usage in science of science workflows was read comprising:
- NIH data at 3.4GB, NSF data at 489MB, NIH data at 139MB, and NEH data at 12.1MB data prepared for temporal analysis.
- Data from NIH, NSF, MEDLINE, UPSTO, and Clinical Trials at 11.5 MB and MEDLINE data at 1GB to be used in geospatial analysis.
- MEDLINE data at 514KB for topical analysis.
- NSF data at 11.9MB and UPSTO data at 1.04GB network analysis.

Average load times measured across ten trials are shown in Table 1. The three largest datasets, would not load but returned a Java heap space error (-TF*).

Table 1: Comparison of load times, measured in seconds, across nine different datasets.

Dataset	Size	Number of Records	Mean	Standard Deviation	Minimum	Maximum
NIH (year, title, abstract)	3.4GB	2,490,837	-TF*			

USPTO (patent, citations)	1.04GB	57,902,504	-TF			
MEDLINE (geospatial)	1.0GB	9,646,117	-TF			
NSF (year, title, abstract)	489MB	453,740	64.54	0.991	63.2	65.9
NIH (title, year)	139MB	2,490,837	83.86	1.32	82.3	85.6
NEH (year, title, abstract)	12.1MB	47,197	2.05	0.070	1.9	2.1
NSF (co-author network)	11.9MB	341,110	4.52	0.063	4.4	4.6
Combined geo-spatial	11.5MB	11,549	1.91	0.056	1.8	2.0
MEDLINE journals	0.5MB	20,775	0.44	0.096	0.3	0.6

Temporal Studies (“When”)

To test the scalability of temporal analysis within Sci2 we selected the Burst Detection algorithm as described by Kleinberg (2003). To test this in a standardized fashion, we generated a randomized set of years from 1980 to 2000, assigning each year a distribution of short strings to test the accuracy of the algorithm. We then calculated the average time, minimum time, and the maximum time it took the Burst Detection algorithm to complete across ten trials. In all cases, the algorithm was able to detect a pre-programmed burst of a word over a short time frame.

A look at the table and graph in Figure 6 shows linear growth with number of records that holds equally true with file size. It is possible that with larger files, this may begin to show the same quadratic tendency as the file loading, but 2.5 million records was the largest file loaded. The data does illustrate that, barring resource exhaustion issues, Sci2 runs this algorithm in a linear timescale.

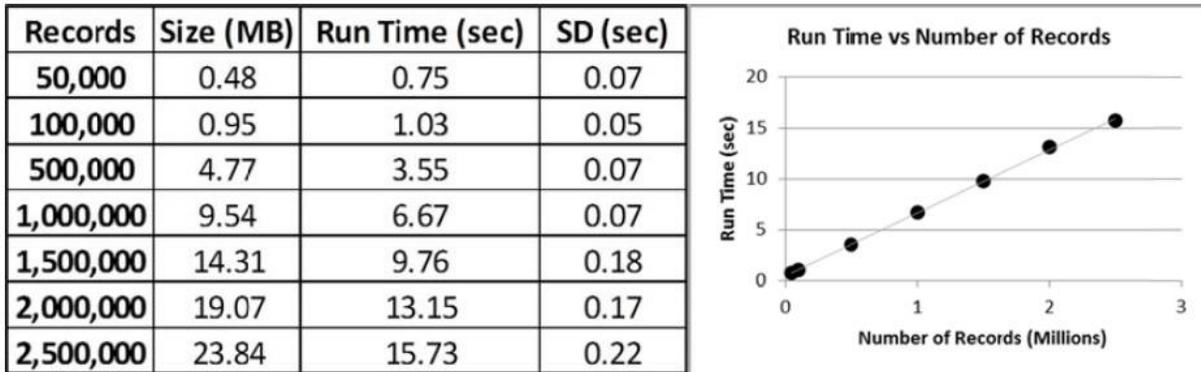


Figure 6: Comparison of Burst Detection run times, measured in seconds, across standardized datasets, tabulated (left) and plotted (right).

We then conducted a burst analysis of the title fields for NIH, NSF, and NEH grant data. The NSF and NEH datasets contain three columns: title, abstract, and year. The NIH data contains only two columns: title and year. The NIH grant data set is the largest at 139MB and 2,490,837 records, followed by the NSF grant data at 489MB and 453,740 records, and finally the NEH grant data at 12.1MB with 47,197 records. In order to obtain accurate results with the Burst Detection algorithm we had to normalize the title text with the Lowercase, Tokenize, Stem, and Stopword Text algorithm prior to running the Burst Detection algorithm, a step not necessary with the synthetic data since it was optimized for burst analysis. Due to the number of records in the NIH dataset, the Lowercase, Tokenize, Stem, and Stopword Text algorithm failed to terminate and as a result the Burst Detection algorithm was not tested with this dataset (-NT*).

Table 2: Temporal Analysis Algorithm Run Time in seconds.

Burst Detection						
Dataset	Size	Rows	Mean	SD	Min	Max
NSF	489 MB	453,740	13.64	0.648	12.9	14.8
NIH	139 MB	2,490,837	-NT*			
NEH	12.1 MB	47,197	1.57	0.094	1.4	1.7

Geospatial Studies (“Where”)

In order to test Sci2 performance for geomapping, randomized datasets with lists of U.S. cities and associated longitude and latitude, were generated. There was only one distinct step (algorithm) involved in this geospatial workflow: visualizing the geolocated data with the Proportional Symbol Map (Biberstine, 2012), see U.S. geomap in Figure 2. We projected this on a map of the United States, as this data set only included locations within the U.S. Average run times are shown in Figure 7. Like with file loading, the Proportional Symbol Map data is better fit by a quadratic model (R^2 of 0.997 as opposed to 0.9834 for a linear fit).

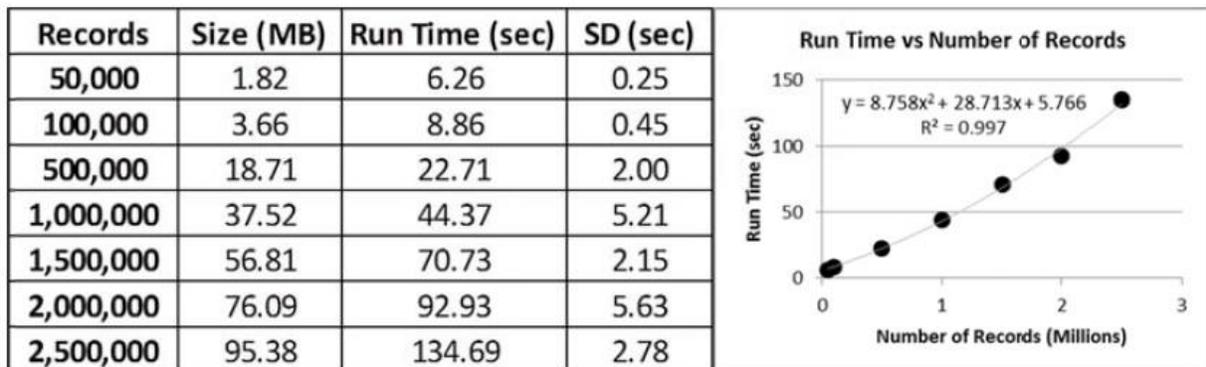


Figure 7: Comparison of Proportional Symbol Map run times, measured in seconds, across standardized datasets.

Next, 11,848 SDB records related to gene therapy funding (NIH, NSF), publications (MEDLINE), patents (USPTO), and clinical trials were loaded and the Proportional Symbol Map was used to display the geocoded data. Exactly 299 records had no or incomplete geolocation data and were removed resulting in 11,549 rows at 11.5MB. The run time, at 4.37 sec is lower than predicted by the model (6.11 sec), implying that the quadratic model may not perfectly describe the run time, particularly with smaller sets.

Table 3: Geospatial Analysis Algorithm Run Time in seconds.

Algorithm 1: Proportional Symbol Map						
Dataset	Size	Rows	Mean	SD	Min	Max
Pre-located	11.5 MB	11,549	4.37	0.125	4.2	4.6

Topical Studies (“What”)

The Sci2 tool supports the generation of science map overlays. Specifically, it uses the UCSD map of science and classification system (Börner, Klavans, Patek et al., 2012), a visual representation of 554 sub-disciplines within 13 disciplines of science and their relationships to one another, see lower left map in Figure 2. This basemap is then used to show the result of mapping a data set's journals to the underlying subdiscipline(s) those journals represent (Biberstine, 2011). Mapped subdisciplines are shown with node sizes relative to the number of articles matching journals and color is based on the discipline as defined in the basemap. To create a standardized dataset, random lists of valid journal names were generated. The number of records and run time results are tabulated in plotted in Figure 8. Linear and quadratic models fit about equally well, but both show that the intercept is about 1.5 seconds, more than half of the run time for all but the largest sets. This stands to reason as the lookup tables must be loaded and accessed regardless of the size of the dataset being used.

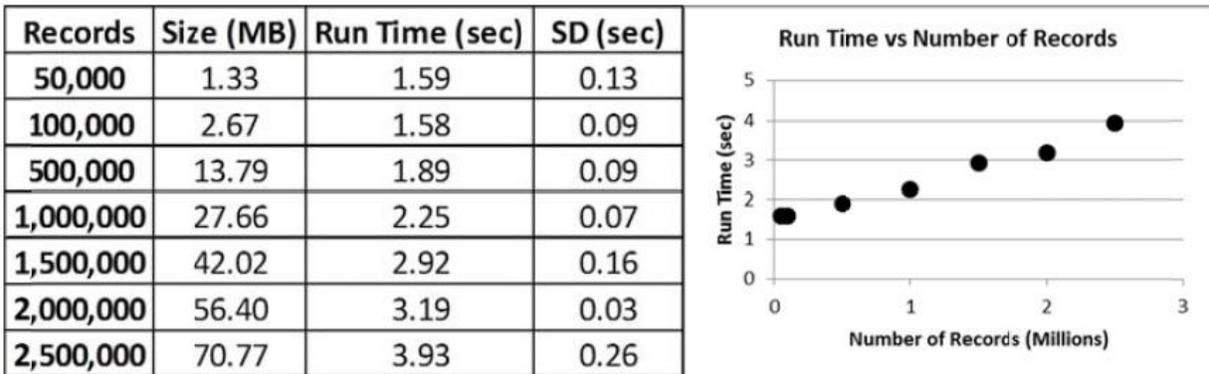


Figure 8: Comparison of UCSD Map of Science Generation run times, measured in seconds, across standardized datasets.

Next, MEDLINE data was obtained from SDB including all 20,773 journals indexed in MEDLINE and the number of articles published in those journals. Average Map of Science via Journals run times are given in Table 4.

Records	% Conn	Edges	Size (MB)	Run (sec)	SD (sec)	Records	% Conn	Edges	Size (MB)	Run (sec)	SD (sec)
500	2	5,000	0.017	1.13	0.05	250	50	31,250	0.124	1.86	0.05
500	5	12,500	0.045	1.44	0.07	500	50	125,000	0.546	5.89	0.1
500	10	25,000	0.093	1.92	0.04	1,000	50	500,000	2.28	20.74	0.12
500	25	62,500	0.247	3.46	0.08	1,500	50	1,125,000	5.21	45.28	0.44
500	50	125,000	0.546	5.89	0.1	2,000	50	2,000,000	9.33	79.41	0.62

Figure 9: Average Directed Network Extraction run times, measured in seconds versus the number of edges in the dataset, across standardized datasets, tabulated with varying connectivity (left) and number of nodes (right) (top) and plotted (below).

Next we retrieved from the SDB all 6,206 USPTO patents that cite patents with numbers 591 and 592 in the patent number field. We ran the Extract Directed Network algorithm, creating a network pointing from the patent numbers to the numbers those patents reference in the dataset and results are given in Table 5. While the scalability of Sci2 third-party visualization tools such as GUESS, Cytoscape, and Gephi do not pertain to Sci2 in a direct way, we were interested to understand their scalability. Neither Cytoscape nor GUESS were capable of rendering the network in a Fruchterman-Reingold layout, while Gephi loaded the network in 2.1 seconds and rendered it in about 40 seconds (the actual process in Gephi is non-terminating, but this was the time to a reasonably defined network). Gephi is able to achieve higher performance due to its ability to leverage GPUs in computing intensive tasks.

Table 5: Network Analysis Algorithm Run Time in seconds.

Algorithm 1: Extract Co-Occurrence Network							
Dataset	Size in MB	Nodes	Edges	Mean	SD	Min	Max

U.S. Patent References	0.147	12,672	7,940	7.88	0.103	7.7	8.1
------------------------	-------	--------	-------	------	-------	-----	-----

Discussion and Future Work

This paper introduced and examined the scalability of a database-tool infrastructure for big science of science studies. SDB relational database functionality was exploited to store, retrieve, and preprocess datasets. Subsequently, the data were processed using the Sci2 Tool. The scalability of this approach was tested for exemplary analysis workflows using synthetic and SDB data. Techniques used were similar to those employed in testing the performance of web-native information visualizations (Johnson & Jankun-Kelly, 2008). Most run-times scale linearly or exponentially with file size. The number of records impacts run-time more than file size. Files larger than 1.5 million records (synthetic data) and 500MB (SDB) cannot be loaded and hence not be analyzed. Run times for rather large datasets are commonly less than 10 seconds. Only large datasets combined with complex analysis require more than one minute to execute.

A forthcoming paper will compare the runtime of Sci2 with other tools that have similar functionality, e.g., TEXTrend or VOSViewer for topical analysis and visualization; CiteSpace, Leydesdorff's Software, DynaNets, SISOB, Cytoscape, and Gephi for network analysis and visualization, see below and (Cobo, López-Herrera, Herrera-Viedma et al., 2011) for links and references.

Recent work has added web services to the Sci2 Tool and selected workflows can now be run online. Other efforts aim to expand the adoption of OSGi/CIShell in support of algorithm and tool plugin implementation and sharing across scientific boundaries. Tools that are OSGi/CIShell compatible comprise TEXTrend (<http://textrend.org>) led by George Kampis at Eötvös Loránd University, Budapest, Hungary supports natural language processing (NLP),

classification/mining, and graph algorithms for the analysis of business and governmental text corpuses with an inherently temporal component and DynaNets (<http://www.dynanets.org>) coordinated by Peter Sloot at the University of Amsterdam for the study of evolving networks, or SISOB (<http://sisob.lcc.uma.es>) an observatory for science in society based in social models.

Much of the development time for the SDB for the last year has been focused on adding data to the system and refactoring code to make it easier to manage and update. Going forward, we plan to implement an API to further ease access and usage of the SDB and we are exploring an RDF conversion to add SDB to the Web of Linked Open Data (Heath & Bizer, 2011). In addition, we are considering a visual interface to SDB that uses Sci2 Web services to empower users to interactively explore, analyze, and visualize search results.

Documentation and teaching of tool functionality and workflows are important for research and practice. SDB and Sci2 are used in the Information Visualization MOOC (<http://ivmooc.cns.iu.edu>) which debuted in Spring 2013 to over 1,700 users, making existing and new workflows available via video tutorials to a much broader audience.

Acknowledgements

The Scholarly Database is funded by the National Science Foundation under Grant No. IIS-0238261. SDB and Sci2 are supported by the National Science Foundation under Grants No. IIS-0513650, SBE-0738111, a James S. McDonnell Foundation grant, and the Cyberinfrastructure for Network Science center at the School of Library and Information Science at Indiana University. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

- Alencar, A. (2010). CISHell: Extract Directed Network Retrieved January 24, 2013, from <http://wiki.cns.iu.edu/display/CISHELL/Extract+Directed+Network>
- Belter, C. (2012). Visualizing Networks of Scientific Research. Information Today, Inc., 36(3). Retrieved from <http://www.infotoday.com/online/may12/Belter-Visualizing-Networks-of-Scientific-Research.shtml>
- Biberstine, J. R. (2011). CISHell: Proportional Symbol Map Retrieved January 24, 2013, from <http://wiki.cns.iu.edu/display/CISHELL/Map+of+Science+via+Journals>
- Biberstine, J. R. (2012). CISHell: Proportional Symbol Map Retrieved January 24, 2013, from <http://wiki.cns.iu.edu/display/CISHELL/Proportional+Symbol+Map>
- Börner, K. (2011). Plug-and-Play Macroscopes. *Communications of the ACM* 54(3), 60-69.
- Börner, K., Klavans, R., Patek, M., Zoss, A., Biberstine, J. R., Light, R., Boyack, K. W. (2012). Design and Update of a Classification System: The UCSD Map of Science PLoS ONE, 7(7), e39464. doi: doi:10.1371/journal.pone.0039464
- Cobo, M. J., López-Herrera, A. G., Herrera-Viedma, E., & Herrera, F. (2011). Science mapping software tools: Review, analysis, and cooperative study among tools. *Journal of the American Society for Information Science and Technology*, 62(7), 1382-1402.
- Heath, T., & Bizer, C. (2011). *Linked Data: Evolving the Web Into a Global Data Space*. San Rafael, CA: Morgan & Claypool Publishers.
- Johnson, D. W., & Jankun-Kelly, T. J. (2008). A scalability study of web-native information visualization. Paper presented at the Graphics Interface Conference 2008, Windsor, Ontario, Canada.
- Kleinberg, J. (2003). Bursty and hierarchical structure in streams. *Data Mining and Knowledge Discovery*, 7(4), 373-397.
- Kosecki, S., Shoemaker, R., & Baer, C. K. (2001). Scope, characteristics, and use of the U.S. Department of Agriculture intramural research. *Scientometrics*, 88(3), 707-728.
- LaRowe, G., Ambre, S., Burgoon, J., Ke, W., & Börner, K. (2009). The scholarly database and its utility for Scientometrics research. *Scientometrics*, 79(2), 219-234.
- Price, D. J. d. S. (1963). *Little Science, Big Science*. New York: Columbia University Press.
- Robertson, G., Ebert, D., Eick, S., Keim, D., & Joy, K. (2009). Scale and complexity in visual analytics. *Information Visualization*, 8(4), 247-253. doi: 10.1057/ivs.2009.23
- Suda, B. (2012). The top 20 data visualization tools. .net. Retrieved from <http://www.netmagazine.com/features/top-20-data-visualisation-tools>