Software Discovery Index Meeting Report - Request for comments

Contents

INTRODUCTION 2

A. FRAMEWORK SUPPORTING THE SOFTWARE DISCOVERY INDEX 2

Unique identifiers 3

Connections to publishers 4

Complementarity with the Data Discovery Index 6

B. CHALLENGES AND REMAINING QUESTIONS 6

Defining relevant software 7

Integrating with other repositories 7

Evaluating progress and distinguishing this from other efforts 7

C. IMPLEMENTATION ROADMAP 8

D. CONCLUSIONS 9

E. APPENDIXES 10

Appendix 1: Minimal information about software (MIAS) 10

Appendix 2: Use cases 10

Appendix 3: Metrics and milestones 11

Appendix 4: Existing software indexes 11

Dear Reader,

This document is the result of a reported intended to summarize the recommendations generated from an NIH Software Discovery Meeting held in May 2014. We are now requesting comments from the larger community. We have contacted a broad set of constituents who represent software users, software developers, NIH staff, electronic repositories, and journal publishers.

The document is currently read only, but we strongly encourage you to leave comments by highlighting the relevant section and going to Insert-->Comment in the menu. The deadline for receiving comments is OCTOBER_SOMETHING.

INTRODUCTION

The National Institutes of Health (NIH), through the Big Data to Knowledge (BD2K) initiative, held a workshop in May of 2014 to explore challenges facing the biomedical research community in locating, citing, and reusing software. The workshop participants examined these issues and prepared this report with a general outline to address these concerns.

The constituents with the potential to benefit from an improved software discovery system include software users, developers, journal publishers, and funders. Software developers face challenges disseminating their software and measuring its adoption. Software users have difficulty identifying the most appropriate software for their work. Journal publishers lack a consistent way to handle software citations or to ensure reproducibility of published findings. Funding agencies and review panels have difficulties in making informed funding decisions about which software projects to support, and reviewers have a hard time understanding the relevancy and effectiveness of the proposed software in the context of data management plans and proposed analysis.

Though numerous changes are needed to address all these challenges, the workshop identified one fundamental prerequisite for success: an automated, broadly accessible system to enable comprehensive identification of biomedical software. This objectives of this "Software Discovery Index" would be: 1) to assign standard and unambiguous identifiers to reference all software, 2) to track specific metadata features that describe that software, and 3) to enable robust querying of all relevant information for users. The broad use of the Software Discovery Index will create an ecosystem that supports tools useful for software developers, software users, funding agencies, and journal publishers.

The workshop attendees agreed that technical resources exist to create both this ecosystem and the necessary tools. The success of such efforts, however, depends on their acceptance by the scientific community: software developers must obtain identifiers for their software; users must cite software in their publications; journals must leverage and expose these citations; and funding agencies must use this new wealth of information to shape funding decisions and long-term planning. It is only when each constituency sees benefits of engaging in this effort that significant progress can be made.

The ultimate goal of this effort is to ensure all publicly funded biomedical software are highly accessible to the research community with an emphasis on maintaining that software in common, open access repositories such as GitHub and SourceForge. Making software easier to find, easier to cite, and easier to reuse are all necessary steps. It is also critical, however, to support the continued development and availability of software tools. Without access to both the tools and the scientific literature describing their use, the research community will not be able to select and use the best tools. In all these areas, NIH is responsible for ensuring that its investment in biomedical research is being leveraged to the greatest effect.

A. FRAMEWORK SUPPORTING THE SOFTWARE DISCOVERY INDEX

The workshop identified many potential characteristics and features for an ecosystem in which users can locate, cite, and reuse software. As discussed at the workshop, the core of the system

consisted of the use of unique identifiers that are obtained by developers, and linked to software hosted at a software repository. In collaborations with publishers, the software identifiers could appear in journal publications. All software metadata could be hosted at a central site, and serve in the creation of a Software Discovery Index.

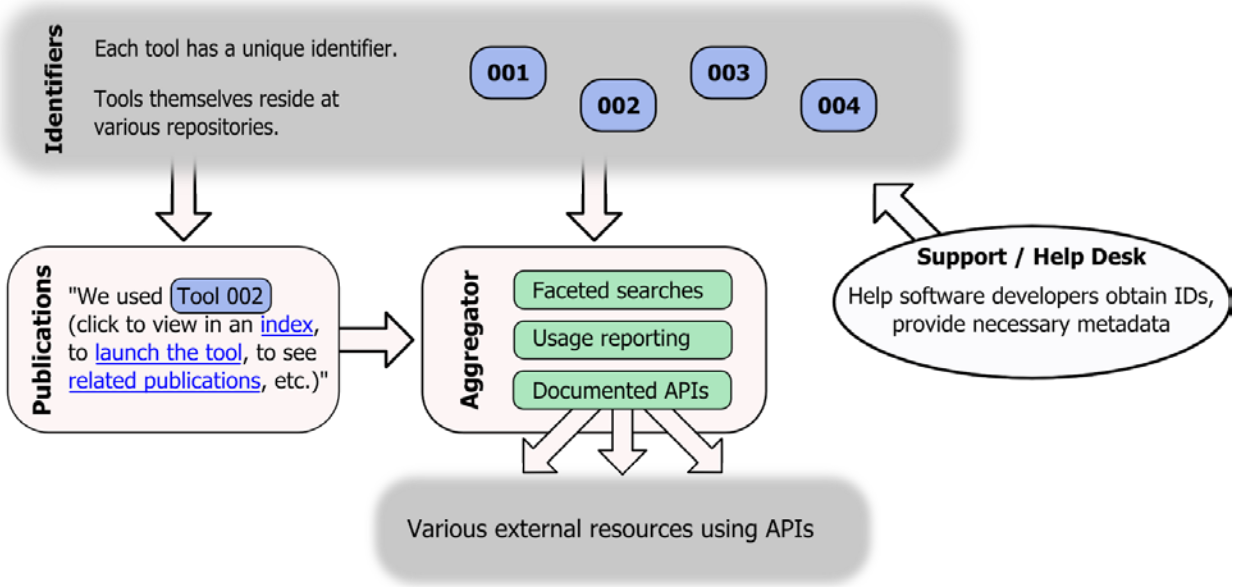

Figure 1: The proposed software ecosystem.

Unique identifiers

Unique identifiers for biomedical software are critical for all that follows. The specific system of identifiers used is of far less importance than the adoption of those identifiers among software developers, software users, and publishers. Even so, however, the choice of identifiers could make it easier or harder to meet the needs of each of these communities.

The temporally dynamic nature of software development makes unambiguous identification difficult. Individual software packages may have many versions, may be branched along different development paths, and may be bundled into collections with other packages. Identifiers must operate across all of these cases, both disambiguating and linking related tools.

The system of identifiers should also enable the association of metadata [Footnote: e.g., By 'metadata' we refer to the information model that is used to uniquely identify and minimally describe software entities. Community-driven standards would used to identify required versus optional metadata fields that would instantiate entities in the Index.] The metadata so associated should facilitate the identification of scientifically relevant software packages. Collecting this information as a static catalog or set of web pages runs the significant risk of perpetuating stale metadata. In facing that challenge, the open-source software community has developed multiple ways to capture metadata on projects with minimal duplication. The most common approach is to define a format in which the

project metadata can be stored as part of the project itself and then scraped by any interested parties. This means that the software developers only have to provide the metadata once, enabling the Software Discovery Index and other interested parties to scrape and use it. It also ensures that updates by the software developers are reflected in all repositories. In this effort, controlled vocabularies and ontologies may prove to be useful, but should not be the primary focus of the initial effort.

Connections to publishers

There is increasing recognition within the scientific community that recording how software is used is a critical part of the scientific record. The dissemination of scientific results, however performed, must unambiguously describe the software used to generate those results and the steps performed. With publications currently the *lingua franca* for disseminating biomedical research results, connections with journal publishers will be essential for this effort.

Comprehensively and efficiently tracking the use of software in research requires a new standard for software citations. At present, most software is cited indirectly by citing either a publication or a URL where the software is described. Citing publications leverages the existing publication citation infrastructure, but it is insufficient for tracking software. This system only enables citation of software described in publications. Even software described in publications, if actively-developed, is likely to cycle through many more released versions than publications. URLs pointing at descriptions of software similarly do not often meet requirements for tracking release versions and other metadata. Moreover, URLs have a tendency to break as documentation and source code moves. Finally, for reproducibility and archival purposes, we need a persistent mechanism to cite software that may no longer be under active development or even available.

A better system for citing software becomes possible with the existence of a consistent system of unique identifiers for software. In order to enable identifying all the publications using a particular software tool, these identifiers will need to be accessible through an API. There are multiple ways that citations can be retrieved from publications, including direct submission from journals, extraction by MEDLINE, and full-text mining.

One major initiative that aims to address this issue is the use of Research Resource Identifiers (RRIDs). The RRID project is currently underway at FORCE11 (the Future of Research Communications and e-Scholarship) and is lead by a partnership between the University of California, San Diego, and Oregon Health & Science University. The RRID project was initiated to make it easier to track key research resources within the biomedical literature by ensuring that authors provide unique identifiers: RRID's, for each resource used to produce the results of a published study. The initial pilot project was launched in February, with agreements by over 30 journals to ask authors to provide RRIDs for 3 initial types of research resources for newly submitted manuscripts: antibodies, genetically modified animals, and software tools/databases. The project established a centralized portal (http://scicrunch.com/resources) that aggregates accession numbers from the registries that are authoritative sources for these types of resources: the Antibody Registry (http://antibodyregistry.org) for antibodies; model organism databases for genetically modified organisms, for example, MGI, ZFIN, Flybase, etc.; and content provided by the Neuroscience Information Framework resource registry

(http://neuinfo.org) for software tools/databases via a generic portal SciCrunch (http://scicrunch.org/resources), which itself aggregates from software resources like the NeuroImaging Tools and Resource Catalog (http://nitrc.org). The Resource Portal allowed authors to search for their resources and supplied the form of the identifier, which was composed of the prefix RRID followed by the accession number, e.g., "FreeSurfer (RRID:nif-0000-00304) software is publicly and freely available from the FreeSurferWiki resource (http://surfer.nmr.mgh.harvard.edu/fswiki/FreeSurferWiki)."

Another major initiative is to provision DOIs for archived code as part of a collaboration between Mozilla, Figshare, and GitHub. Add info here about Zenodo/Figshare/GitHub DOI minting efforts

Ultimately, successful use of unique identifiers for software requires not only a structure, but also social adoption. No tracking system will work unless authors properly cite the software used in their research, something that is inconsistently performed at present. Similar to how public access policy is implemented, one can imagine that if funders support the use of such identifiers in grant reports and systems such as MyNCBI, that both authors and publishers will be much more incentivized to comply. The RRID project has shown that both journals and authors are willing and able to adopt new citation styles and has also illustrated the types of components that are necessary for a successful system. Both the RRID and Zenodo projects have shown that citing a resource drives population of the registries, as significant numbers of resources were added to the appropriate registries in order to obtain the identifiers for citation. Software Discovery Index - Utilization

The combination of unique identifiers for software and the use of those identifiers in publications will enable the creation of a rich dataset of software relevant to the research community. This dataset will be captured through the Software Discovery Index, consolidating data on software packages and their use. The Index would not be a new repository for software code, but rather a resource collecting data from many repositories, publishers, and other sources, as was demonstrated for the RRID project. This Index is likely to be the most tangible element of the effort, but we anticipate it will be highly susceptible to feature creep. One of the essential tasks is defining what features are appropriate for the next phase of implementation. Not all of the features described here are likely to be included in the first phase of an official NIH Software Discovery Index.

One obvious function of the Discovery Index is to provide metadata describing software packages. The metadata selected for inclusion must be carefully considered for relevance to various users. A selection of metadata fields are listed in Appendix 1. The definition of such metadata will likely lead to other systems (outside the NIH) also using the same metadata, and the participants of this workshop believe that it is important to work in a broad community context that includes diverse organizations to define these metadata.

Aggregating data across multiple sources will help make software from multiple sources more comparable, enabling the calculation of utility scores for software. Though the specifics of these scores remain to be defined, it is clear that they will depend on more than just citations in the scientific literature. Other likely factors include documentation, codebase activity, and user community vitality.

Capturing metrics on these attributes will require a significant development effort, but consolidating this information and determining methods for and providing these scores will be of tremendous value to the scientific community.

Some information may not be appropriate to factor into a utility score but should still be consolidated and presented whenever possible. The notorious difficulty in documenting software reliability will require the Index to flexibly capture a wide range of information regarding reliability. Completeness of documentation can be measured at several different levels including the ability to install the code, or understand the impact of changing run-time variables. The presence of unit or integration tests are also critically important, but impractical to rigorously measure. Inclusion of benchmarking results could also be utilized as part of a measure of reliability. By increasing the visibility of benchmarking results, the Index may encourage developer investment in high-quality benchmarking. By providing gold standard datasets against which benchmarks could be run, the Index could also simplify benchmarking. If the Index captures multiple measures of software utility and quality, it should be possible to offer certification levels for software, signaling compliance with various best practices. Such recognition could both help users wishing to find software and encourage software developers to follow those best practices.

Finally, to maximize the utility of this index, it is critical that its information be exposed via both a website and an API. The website should provide a convenient point of entry that allows various faceted searches and browsing software tools. This website is expected to evolve significantly as this effort matures, but it is important that the first iteration is usable and streamlined. Over the long term, however, the API is likely to be at least equally important as the website. It is likely that websites serving specific research communities will wish to provide their own filtered views of the data, and this should be encouraged through an API. Moreover, other resources such as Synapse, GitHub, Zenodo, figshare, SciCrunch/NIF, and others may wish to expose their software to this index, and the API should enable this as well. A thoroughly-documented and usable API for both providing data to this aggregator and retrieving data from it will be critical to its long-term success.

Complementarity with the Data Discovery Index

The Data Discovery Index (DDI) is a NIH Big Data to Knowledge (BD2K) project. The DDI will enable investigators discover, access, and cite biomedical big data. The DDI aims to cut across disciplines and provide an index that will broadly serve across all NIH investigators. We expect that the Software Discovery Index will be fully compatible with the DDI, with the goal of allowing DDI and Software Index objects appearing in electronic journal articles, and enabling comprehensive retrieval of both data and the software that is utilized to analyze or produce these datasets.

B. CHALLENGES AND REMAINING QUESTIONS

The framework proposed above consists of endorsing identifiers, collaborating with publishers, and developing a Software Discovery Index. Each of these tasks carries particular challenges. Some of these challenges must be solved now, while others should be considered and possible solutions proposed in the first iteration of this effort.

Two important early needs are to define the scope of this project and to provide a help desk. A key element of defining the scope of this initial project will be deciding what software should be covered. Though ultimately the system should not limit itself to NIH-supported software, a limited scope could be useful in the early stages. The help desk should help users navigate this system. Software developers, in particular, will need guidance on obtaining unique identifiers for their software and in crafting useful metadata files. Other users are also likely to benefit from assistance locating, citing software, and tracking software.

Defining relevant software

One challenge of this effort will be to define relevant software. Biomedical researchers use a tremendous amount of software that does not need to be captured in this effort. It is relatively clear that no citation is necessary for a text editor, even when the features of that text editor may have greatly helped a researcher. Likewise, it is relatively clear that the statistical analysis package used to analyze a dataset should be cited. A great deal of biomedical software, however, falls between these two extremes. Many researchers use a few lines of script to store parameters for command line programs. Sometimes, those simple scripts grow until they are used to pipe data between multiple tools. It will be important not to overwhelm users with tremendous quantities of software that is little more than a few lines of script. Multiple avenues exist for achieving this and it will be important to select with care the approach for the project. One aspect to consider is what degree of citation would be required in order to reproduce any given analysis. Whilst full reproducibility may be commonly out of scope, supporting readers' ability to attempt to utilize the results of any given study should be in scope.

Integrating with other repositories

Aggregating data from multiple sources, though it opens major opportunities to improve software development and design, also requires integration with multiple repositories. The goal of the Software Discovery Index is not to replace existing software repositories, but rather to pull as much information from them as possible and to present that information in a consistent and useful form. This is similar to the role that PubMed plays for journals - PubMed aggregates the results and provides them in a consistent form, but does not oversee curation or peer review. Similarly, the repositories will likely be the ones that ensure standard metadata and provide some degree of curation. This will require strong relationships with multiple existing repositories and a willingness to work with new repositories that contain relevant software.

Evaluating progress and distinguishing this from other efforts

This system is not the first attempt to create an Software Index for NIH-supported software, and we should learn from prior efforts. For example, NIH dedicated significant support to the BioSiteMaps effort. Numerous researchers have created lists of significant software in their own fields, for example curated projects like the Neuroimaging Tools and Resource Clearinghouse (NITRC). Finally, the RRID project and underlying Neuroscience Information Framework Resource Registry have been broadly populated and are used by a broad community. It will be critical to consider what distinguishes this effort from previous efforts as well as any overlap in order to define metrics for success. Some of the key features of this effort that distinguish it from prior efforts include the automated indexing of

software, the integration with multiple registries, and the provision of APIs enabling the creation of community-specific user interfaces.

C. IMPLEMENTATION ROADMAP

What follows is a preliminary listing of milestones that would be involved with implementing a Software Discovery Index:

- Define a checklist for the Minimal information about software (MIAS). A preliminary for the MIAS list is described in Appendix 1. The MIAS must be defined by a broad international community utilizing methods that been used to achieve minimal lists such as the MIAME and MIxS.
- Develop and utilize methods to assign unique identifiers to software systems, these methods should take advantage of existing approaches where possible e.g., Research Resource Identifiers (RRIDs).
- Establish and maintain an API. Capability of the API should include but not be limited to search and browsing capability, direct entry of new software tools by existing repositories, as well as facile interaction between the Software Index and electronic journal articles.
- Establish and maintain a website with search and browsing capability for software tools. Use cases are described in Appendix 2 which should be developed further, and design of the web interface should accomplish all use cases in a facile manner.
- Develop partnerships with journal editors in order to migrate the use of Software Index unique identifiers into electronic publications. At a minimum this would require: developing relationship with an extensive number of journals that use Software Index identifiers, providing electronic file formats and APIs for data exchange, and providing documentation for authors to make use of the identifiers.
- Establish and maintain an Advisory Working Group composed of international members of the user community, software developers, software repositories, other relevant electronic repositories and representatives of electronic journals.
- The leadership of the Software Index will engage with the Data Discovery Index management teams to ensure effective management of these two activities.
- Establish and utilize performance metrics to monitor the on-going success of the Software Discovery Index. A preliminary list of recommended performance metrics is described in Appendix 3. Detailed results of the performance metrics would be reported to any relevant funding agency as well as the Advisory Working Group. Summaries of the performance metrics (e.g., number of indexed software packages) would be reported on the Software Index website.
- Engage in extensive promotion the Software Discovery Index by advertising in journals, presenting at scientific conferences, utilizing social media and publishing peer-reviewed journal articles.

D. CONCLUSIONS

The Software Discovery Index Meeting held May 12-13, 2014 served as a forum that resulted in several important conclusions. There was universal agreement that given the unprecedented abundance of electronically encoded information such as `omic data, imaging and EHR, the software required to manage and understand this data is has also become increasingly critical to biomedical research.

Meeting attendees also agreed that software is no longer incidental to the data, the systems used to analyze or produce raw data must be indexed in such a way to promote reproducibility for analysis of data sets. We also agreed that due diligence should be exerted towards reviewing existing projects, including the RRID project, to provide important insights about the operation of a Software Discovery Index. In exploring this problem, we also concluded that by assigning universal locators to software we could also significantly ease the process of locating, citing, and reusing software. This workshop proposed the use of unique Identifiers for software packages, the formation of collaborations with Publishers to track software used in publications, and the creation of a Software Discovery Index to provide information on software packages. If successful, an implementation of these three efforts would benefit software developers, software users, journal publishers, and funding agencies.

E. APPENDIXES

Appendix 1: Minimal information about software (MIAS)

A common set of metadata fields are critical for useful indexing. If this effort only provides refined free-text searching capabilities, it will not be a major improvement over currently-available resources. It is necessary, therefore, to define a key set of minimal fields can that provide maximum value. At the workshop, the following fields were described as candidates for inclusion in this list:

- Persistent identifier
- Software title
- Software version
- Software license
- Links to code repository
- Human-readable synopsis
- Author names and affiliations
- Terms to describe software objectives or functions, and/or the following two bullets (controlled by an appropriate ontology)
- Formats for data inputs and outputs
- Platform, environment, and dependencies
- Associated grants and publications

Appendix 2: Use cases

- Developer: A developer registers their software, she is able to track and quantify all use of their software in scientific publications, through comprehensive and accurate citation of the index-associated identifier. With the ability to find similar types of software packages (e.g., other assembly programs), she would also identify benchmarking data sets, and other related software development efforts.
- User: An NIH funded researcher is seeking software for analysis. They are able to identify the most appropriate software relevant for their study on their data on their computer systems and objectives, and be provided with all information necessary to locate, obtain, and deploy the software.
- NIH: A program officer can identify both the creation and the use of all software funded by a grant they have awarded, analogous to how they can track all papers and citations to those papers funded by a grant they have awarded. They can also identify similar or overlapping products. Review panels can assess software choices in funding proposals and data management plans.
- Publisher: A publisher can associate software with their publications during & for peer review and upon publication for citation. They can also pull & display metrics related to all the research objects surrounding the article, including software based on the software identifier.

Appendix 3: Metrics and milestones

It is critical to define metrics for this effort. These metrics should be evaluated both in absolute terms and in relative terms, monitoring the growth with time. These metrics are particularly significant because this is not the first effort to make biomedical software more accessible to researchers. This effort will face many of the same challenges faced by previous efforts and it is critical to closely monitor whether it is accomplishing its purpose. Specific metrics proposed for the initial effort included:

- Number of developers contributing software
- Number of software records created
- Software identifiers appearing in and extracted from publications
- Links from publications to software records
- Links between indexed software and other resources, people, and data
- Annotation of existing collections of software packages (e.g., Bioconductor)
- The number of interoperating resources, including repositories, aggregation resources, and user forums
- The use of the APIs to re-package the data for specific use cases
- The proportion of NIH-supported software tracked by the software discovery index

Tracking these metrics will provide insight into the progress of this effort. Progress against these milestones should, wherever possible, be evaluated against milestones. Specific milestones could include the fraction of NIH-supported software included in the first year, the time for machine-actionable links to software in PubMed, and the time for API establishment.

Appendix 4: Existing software indexes

There are numerous existing software indexes serving specific communities, many of unrelated to biomedical researchers. Some of the challenges that this effort will face have also been addressed by these indexes.

There are existing package management systems, notably RPM and dpkg for Linux. These tools facilitate the installation, upgrading, and uninstallation of software packages. Both systems have ways to unambiguously track software packages and ways to aggregate data on those packages, significant requirements articulated at this workshop. It is also interesting to note that these are both low-level tools and that users typically interact with them via higher level interfaces. This sort of modular model fits well with what was described at the workshop, with its focus on providing an extendable framework that others can leverage. This model differs from that famously employed in the Apple App Store and Google Play, where the software is directly hosted and managed on the index itself. The index, as described here, would be a lower-level construct that supports various package management functions but does not itself perform those functions.

SciCrunch/NIF/RRID: The Neuroscience Information Framework, (http://neuinfo.org) is a project of the NIH Blueprint Consortium that has been surveying, cataloging and federating data and resources (tools, materials, services) of relevance to neuroscience since 2008. It has maintained and populated the NIF Registry, a high level metadata catalog of research resources, currently comprising over 11,000 resources, and tracked them over the past 6 years. Through its unique data ingestion and query platform, it has created a search engine for data that searches across over 200 independent databases comprising over 800 million records. Although NIF was developed for neuroscience, it has expanded well beyond primary neuroscience resources to biomedical resources as a whole. Thus, the software sitting behind NIF was rechristened SciCrunch. SciCrunch allows different communities, e.g., NIF, to use the same infrastructure and data sources to create their own communities. The SciCrunch Registry provides the unique identifiers for software tools and databases for the Resource Identification Initiative (RRID project; http://scicrunch.com/resources). SciCrunch aggregates software tools from multiple repositories, e.g., NITRC. It utilizes authoritative identifiers where possible, and assigns an identifier when the source repository does not.

Participation in the RRID project was voluntary, i.e., not a condition of publication, and was requested by the journals through an email request to the author. The project deliberately did not require journals to modify their journal submission system in order to allow broad participation in the project. To date, ~50 papers have appeared from 11 different journals that use RRID's. Over 200 RRID's have been reported. The FORCE11 working group is collecting data regarding the use of RRIDs in the literature and is making it freely available here. The error rate to date is ~7%. Papers using RRID's can be retrieved from Google Scholar by searching for a particular ID (**Figure 2**). A resolving service has also been developed such so that 3rd party tools can utilize the RRID's to link to a resolvable record as well as to map identifiers where needed. Automated routines based on NLP as being developed to recognize RRID's and to suggest appropriate RRID's based on the resources described. Currently, RRID's are only assigned at the level of the software tool or data resource, that is, it does not specify versioning information. This was a calculated decision, as the primary objective of the RRID pilot was to determine if unique identification of software and other resources would be achievable by publishers and authors. The RRID project in the future will aim to include more detailed and machine actionable information as per outcomes of these and other related community discussions.

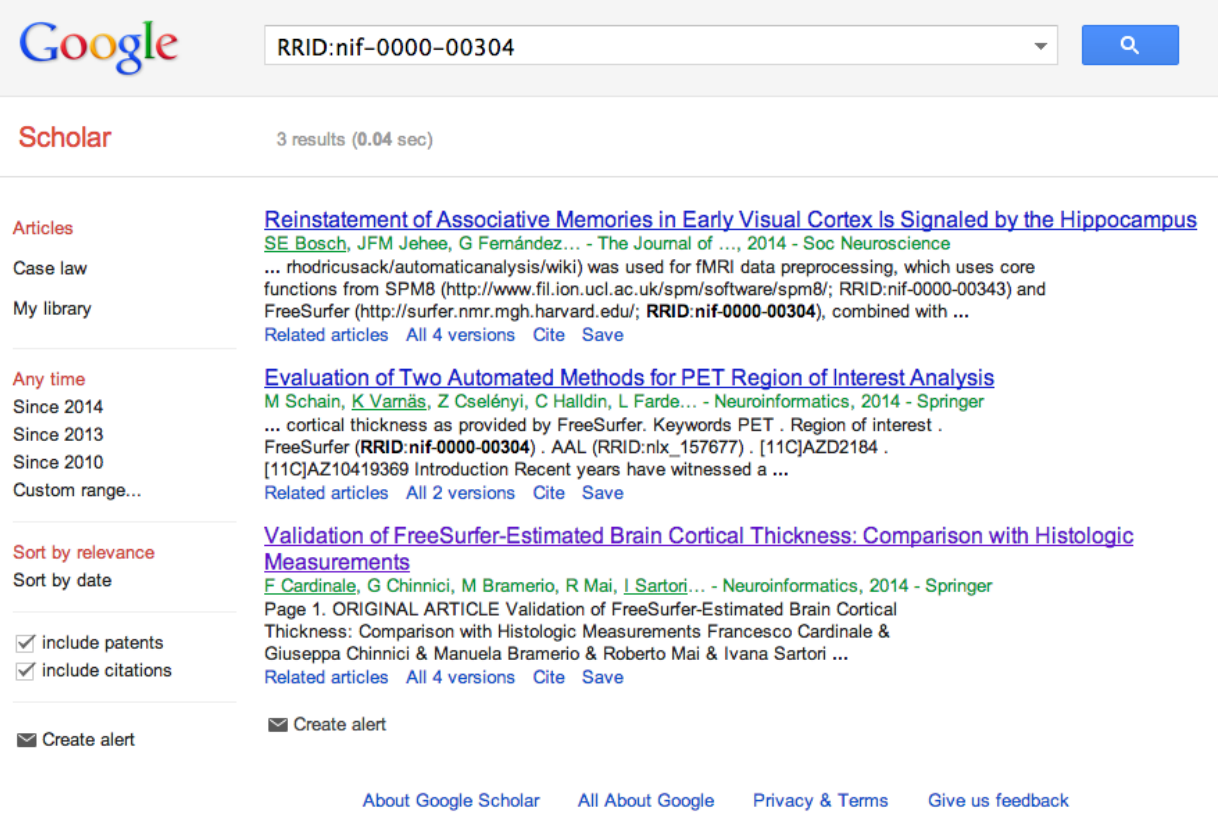

Figure 2: Google Scholar results showing papers citing the RRID for FreeSurfer

Neuroimaging Informatics Tools and Resources Clearinghouse (NITRC): Since 2006, the Neuroimaging Informatics Tools and Resources Clearinghouse (NITRC) has provided a comprehensive support infrastructure for resources, including software, in the neuroimaigng domain (including MRI, PET, EEG, MEG, SPECT, CT and optical neuroimaging tools and resources). NITRC fosters a user-friendly clearinghouse environment for the neuroimaging informatics community. NITRC's goal is to support researchers dedicated to enhancing, adopting, distributing, and contributing to the evolution of previously funded neuroimaging analysis tools and resources for broader community use. Located at www.nitrc.org, NITRC promotes software tools, workflows, resources, vocabularies, test data, and now, pre-processed, community- generated data sets (1000 Functional Connectomes, ADHD-200) through its Image Repository (NITRC-IR). NITRC gives researchers greater and more efficient access to the tools and resources they need, including: better categorizing and organizing existing tools and resources via a controlled vocabulary; facilitating interactions between researchers and developers through forums, direct email contact, ratings and reviews; and promoting better use through enhanced documentation.

nanoHUB.org: Starting in 2002, the NSF-sponsored Network for Computational Nanotechnology established a web site at nanoHUB.org to support the National Nanotechnology Initiative. Any user

within the community can contribute a simulation/modeling or analysis tool to this platform. Tools are not only cataloged, but hosted, so that any user can run the tool through the web via the click of a button--without having to download or install any software. In 2013, more than 13,100 users launched some 500,000 simulation jobs using more than 340 different simulators contributed by the community and installed on nanoHUB. These tools have been used by 22,649 students across 1,165 courses at 185 institutions. nanoHUB also hosts more than 4,000 other resources—including seminars, tutorials, animations, and even complete courses—that help to document the tools and educate new users. In the last 12 months alone, nanoHUB served more than 300,000 unique users with this content, and that number has been doubling every 18 months. In June 2011, the National Science and Technology Council's Materials Genome Initiative for Global Competitiveness highlighted nanoHUB as an exemplar of "open innovation" that is critical for global competitiveness. The HUBzero software that powers nanoHUB.org is available as open source, and more than 60 other projects have used the same software to create similar "hubs" for their own scientific community.

Bioinformatics Links Directory: Initiated in 2001 at the University of British Columbia as a tool to manage "links" for a bioinformatics core facility (akin to "Pedro's list") Francis Ouellette's group has maintained this bioinformatics links directory to its current maturity which is now a resource with more than 1,400 links providing provenance for a given resources, databases and tools. This is now maintained at bioinformatics.ca, but will be migrated to a ".org" URL in the near future. **http://bioinformatics.ca/links_directory/**

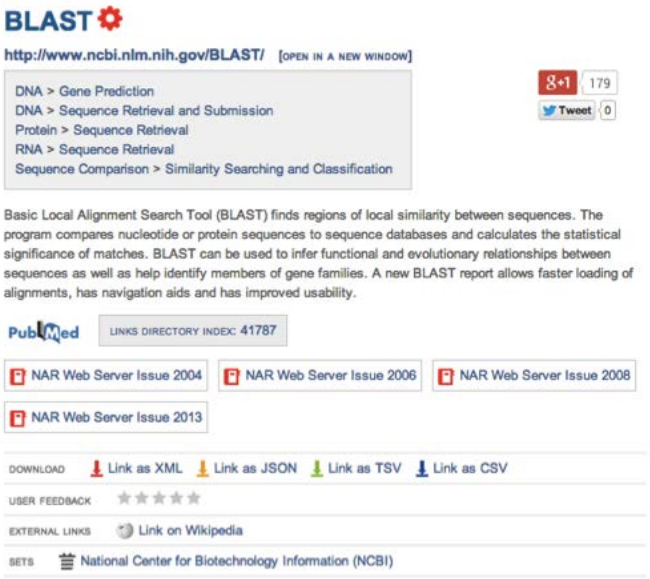

Figure 3: Links directory result for NCBI's BLAST