ITCR metrics of adoption/usage of technology

**Table of contents**

[**Motivation**](#_hvt61su7vrcf) **1**

[**Instructions**](#_upzyanueack5) **2**

[Required information](#_gyy20547ccjj) 2

[Possible questions to consider](#_n9ehvmscsvdp) 2

[**Responses so far**](#_bxdklfqdafyv) **3**

[UCSC Xena](#_pdud5wq1tt5w) 3

[3D Slicer](#_77im2jueb255) 4

[Bioconductor](#_bgayyh7m8014) 5

[NDEx - The Network Data Exchange](#_39hwojd1jumq) 6

[TIES](#_6rfr0ni9bcur) 10

[TRINITY](#_borzwm42l5tm) 11

[The Cancer Proteome Atlas (TCPA)](#_omc86v83j6y) 12

[LesionTracker](#_smi1efb1dwwt) 13

[Integrative Genomics Viewer (IGV)](#_cdmz00hexdk1) 14

[CRAVAT (Cancer-Related Analysis of VAriants Toolkit)](#_1smxutn1bmdk) 15

[GenePattern](#_d1b2la7n5pyp) 16

[**Related references**](#_2d1f7nauulfc) **18**

[**Summary**](#_kw2f0eekzsov) **18**

[Tools overview](#_1vhdcjt3df96) 18

[Impact metrics](#_jk52mlakjydi) 20

[Users](#_7c6ndy3u7zff) 20

[Developers](#_9e6qi5fynvy5) 21

[Academics](#_elq5w5e9hch4) 21

# Motivation

Tracking usage/adoption of software is an important and challenging topic. It is often required to justify continuing support and funding as well as to identify development priorities. However, there is no well-defined “recipe” for reliable and accurate usage tracking. Different groups have adopted different approaches and tricks to achieve this. ITCR Training and Outreach WG is collecting information about these site-specific strategies to provide a resource for the ITCR participants to learn from each other.

The subject of software adoption reporting is an important one not only for the developers, but also for the funding agencies. Considering the existing interest, and depending on the responses we receive, in the future we may consider publishing a report or a white paper summarizing ITCR usage tracking approaches and potentially recommendations for best practices. If and when we decide to proceed with this plan, the contact person submitting the response and listed in this document will be invited to participate in the preparation of the manuscript/report and be a co-author.

Submitting a response to this document is not mandatory.

Please do not share any information you consider private or limited access.

If you choose to participate, please follow the instructions below and add your response. You are encouraged to go over the existing responses before submitting yours for inspiration!

# Instructions

Append your response to this Google Document at the end of the Responses section.

The only required section is about the contact information and tool name. Below are some suggestions to consider, which may or may not applicable to your tool. However, you are welcome to shape your response as you see appropriate and not follow the suggestions, especially since this has historically lead to most interesting reports from tools.

## Required information

* Name of the tool
* Name and email of the person submitting the entry

## Possible questions to consider

* How large is the community you are supporting? How do you know you are supporting this community?
* How long do you intend users to use your tool? Is your tool more of a quick reference for many users or is it a tool that you envision a few users using very deeply?
* Is your platform extensible by outside developers by means of plugins etc, without the need to modify the main application? If yes:
	+ do you provide mechanism to track usage of individual plugins? How?
	+ How do you manage crediting plugin developers?
	+ How do you track usage of individual plugins?
* Do you implement “call home” feature (automatically contact central server on startup of the main application or plugin to query for updates etc, which also allows to track usage)?
* Measures of desktop usage: do you count downloads? Do you provide more fine-grained usage metrics, such as tracking of user invoking certain function of the tool?
* Do you track source code usage?
* Is your tool web-based or is your web presence only to redirect the user to a desktop/docker/etc - based tool? Do you track individual clicks or only track general visits?
* Do you use the cloud? Which cloud service? How do you track usage on the cloud?
* Do you use Docker? How do you measure docker usage?
* Do you measure citations? Mailing list usage? Biostars?
* What metrics have you used in the past but then discontinued?

# Responses so far

## UCSC Xena

Mary Goldman, mary@soe.ucsc.edu

* Distributed data hubs for all major platforms (Linux, Mac OSX, and Windows). Java based. We host several large public datasets like TCGA on our own public hubs. These hubs were originally hosted at UCSC but have been moved to AWS in the last few months.
* Users also have their own private hubs for viewing their own data.
* Web browser to view data from multiple hubs (private or public data) that is supported on recent versions of Chrome, Firefox and Safari. Similar to the public data hubs, we originally hosted it at UCSC but has been moved to AWS in the last few months. There are still parts of the browser that are at UCSC.
* In addition to providing visualization and analysis of the data in the hubs, we also allow people to download the data for their own analysis.
* Our front page is separate from our main web browser and is hosted on ghost.io.
* Source code for the hubs, the browser and the front page is hosted on github.

Statistics for private hubs: Our admins set up [awstats](http://www.awstats.org/) for us. This allows us to monitor how many people download a private hub.

We also support automatic updates of hubs. Each private hub (Windows and MAC, not Linux) tries to connect to UCSC when it is first started up to download a file to determine if an update is available. We monitor the downloading of this 'update' file (also with awstats) as a way to measure usage of private hubs. Note that this method underestimates usage since it assumes the hub is not behind a firewall and users do have the option of disabling automatic updates.

Statistics for public hubs: Originally on awstats, but now are on AWS. We are still figuring out the usage analytics there as well as which tools might be useful to help interpret them.

Statistics for web browser: Google analytics, though now that the browser is on AWS, we're moving to the usage stats there. May end up with a mixture of google analytics and AWS stats.

Statistics for the front page: Mixpanel

Data download statistics and costs: Again, awstats, now AWS. We also are starting to provide data download via S3 buckets on AWS. We saw large amount of download when we leave it to be free-download, which means we pay for the cost of download (2TB over 10 days out of AWS, it is not affordable on our part). We changed it to be requester-pay, which means whoever download the data will pay for the download out AWS. Please note that the data is still open-access, anyone can download the data.

Other notes: We do monitor youtube hits of our tutorial videos, as well as twitter followers and newsletter sign ups. We do not keep track of citations. There is no way to keep track of github downloads, as far as we know.

## 3D Slicer

Contact: Andrey Fedorov andrey.fedorov@gmail.com

* Desktop tool, binaries available for download for all major platforms (Linux, Mac OSX, and Windows)
* 3D Slicer is a platform, with a lot of (most?) functionality available as extensions that are not included in the main repository and are maintained separately
* Source code is hosted on github, documentation is on in-house wiki, with fragmentation for individual modules based on the maintainers’ preference
* **Official and easily accessible metrics we use:**
	+ Download statistics: <http://download.slicer.org/download-stats/> (only one download per IP for a given release is counted towards the final number)
	+ 
	+ Citation counts for the platform papers (<http://wiki.slicer.org/slicerWiki/index.php/Documentation/4.5/FAQ#How_to_cite_Slicer.3F>)
	+ Individual modules may have their own suggestions how they would like to be cited.
	+ information on how often a given extension was downloaded can be extracted from the extension hosting server using the ExtensionStats module.
	+ We maintain manually-curated list of publications that utilize 3D Slicer: <http://wiki.slicer.org/slicerWiki/index.php/Main_Page/SlicerCommunity>
* **Unofficial/exploratory metrics:**
	+ Github stars, followers, code checkout counts, source repository visitor count (caveat: limited to the last 15 days!)
	+ there is an emerging science behind github stars and followers:
		- <https://www.cockroachlabs.com/blog/what-can-we-learn-from-our-github-stars/>
		- <https://github.com/spencerkimball/stargazers>
		- <https://arxiv.org/pdf/1507.00604.pdf>
	+ Google scholar
		- Over 2,200 hits for slicer.org (<https://scholar.google.com/scholar?hl=en&q=slicer.org>)
		- Over 4,000 for “3D Slicer” (https://scholar.google.com/scholar?q=%223d+Slicer%22)



* + Mailing list subscriptions:
		- >1380 unique email address subscribers to the slicer-devel and/or slicer-users mailing lists used for development discussion and user help requests. (<http://massmail.spl.harvard.edu/mailman/listinfo/slicer-users> / ../slicer-devel)
* **Metrics/approaches being discussed:**
	+ Add automatic check for software update by contacting central server each time application is started: see discussion here: <https://www.slicer.org/wiki/Documentation/Labs/ApplicationUsageAnalytics>
	+ Google analytics for website and documentation access counts (partially implemented but not generally accessible).

## Bioconductor

Contact: Martin Morgan Martin.Morgan@RoswellPark.org

Available for local (personal computer or institutional cluster) installation on Windows, Mac, Linux. Available as Amazon [AMI](http://bioconductor.org/help/bioconductor-cloud-ami/), [docker](http://bioconductor.org/help/docker/) image.

Established metrics

* Summarized in [Annual Report](http://bioconductor.org/about/annual-reports/) sections 1 and 2.
* Semi-annual [release metrics](http://bioconductor.org/about/release-announcements/) (e.g., [current release](http://bioconductor.org/news/bioc_3_4_release/)) -- packages per release, new packages.
* [Per-package](http://bioconductor.org/packages/stats/) and [project-wide](http://bioconductor.org/packages/stats/bioc/index.html) download statistics -- per-month unique IP addresses
* [Support site](https://support.bioconductor.org) users, visitors, top-level posts, answers / comments.
* [Developer mailing list](https://stat.ethz.ch/mailman/listinfo/bioc-devel) subscriptions and traffic.
* [Training and outreach](http://bioconductor.org/help/course-materials/) activities and resources.
* PubMedCentral [full-text citations](https://www.ncbi.nlm.nih.gov/pmc/?term=bioconductor&sort=ePubDate) (approximate; term ‘Bioconductor’); [daily and featured citations](http://bioconductor.org/help/publications/)
* Web & support site google analytics -- unique IP addresses (current and change from previous period); platform; geographic location; day-of-week (mostly for fun).

Additional (not reported consistently) metrics

* AMI usage.
* AnnotationHub resource access.

Abandoned metrics

* User mailing list. Abandoned when mailing list replaced by support site.
* Key package PubMed citations. Too difficult to know how to associate publications with packages (the publication introducing the package, versus where the package reached a wide audience, versus use by the package author in a landmark study, versus…). Too difficult to know how to count citations when there are multiple papers supporting the package (e.g., paper A cites package via reference B and C; is that one citation or two?). Too contentious for individual package authors, both competition for ‘most cited’ and because of perceived importance of citation in individual careers.

Insights / challenges

* Most developers are on Linux / Mac; almost 1/2 our users are on Windows despite the challenges this platform poses to informatic analysis.
* Uniqueness of IP addresses depends on window (weekly / monthly / annually); does not represent users (e.g., single user installs from institutional and home IP).
* Download can be to personal IP addresses for single use, or by sys admin for institutional use.
* Individual package download stats can be skewed by unNousual events, e.g., a malformed script running on a 1000-node cluster in Iowa that starts by downloading a particular package.
* Interpreting changes over time can be obscured by changes in infrastructure, e.g., visitors to our web site is growing less quickly, probably because of the emerging utility of the support site.
* Some metrics that seem like ‘bigger is better’ (e.g., packages in a release) may not correspond to best interests of the project (we would rather one high-quality package of general use than two packages of questionable quality or utility).
* It is not in our interest to advertise metrics that might be interpreted in a negative way. E.g., classic microarray packages continue to be widely used almost 15 years after introduction (indicating an important aspect of reproducibility) but this in isolation makes the project seem ‘out-of-date’ and is not countered by the fact that say 80% of users are NGS-focused.

## NDEx - The Network Data Exchange

**Contact: Rudolf T. Pillich, Rudi@ndexbio.org**

**Tool’s features**

* The [NDEx Public server](http://www.ndexbio.org) is maintained by the NDEx team and deployed to an AWS instance and accessible via web browser.
* Private NDEx server instances can also be deployed using the installation package available for download from our FTP site.
* NDEx server instances can communicate with [Cytoscape](http://www.cytoscape.org) using the [CyNDEx App](http://apps.cytoscape.org/apps/cyndex).
* Available client libraries: Java, Python.
* Client libraries in development: Javascript, R.

**Metrics**

**NDEx Public Server (**[**www.ndexbio.org**](http://www.ndexbio.org)**)**

On the NDEx Public Server, we currently track:

* # of registered users
* # of uploaded networks
* # of groups



Additional metrics we plan on tracking/implement include but are not limited to:

* # of visitors
* # of searches
* ranking of most used search terms
* most accessed networks with ranking
* # of neighborhood queries
* Most queried networks with ranking
* Ranking of most used query strings
* # of downloaded networks
* # of exported networks

We also measure a number of parameters to evaluate and compare performance between different builds.

The graph below is an example to show how the “Export time” from Cytoscape to NDEx is greatly improved in the new NDEx 2.0 vs NDEx v1.3. It also shows how faster the NDEx Web application is capable of rendering the graphic view of the tested network (medium size, 11k interactions).



**NDEx Informational Website (**[**www.home.ndexbio.org**](http://www.home.ndexbio.org)**)**

On the NDEx informational website, we currently track:

* # of visitors
* # of views
* Geographical details
* Referrers
* Most clicked links
* Most visited pages

This is accomplished through the Site Stats provided by Jetpack on Wordpress (see image below). The Informational website is in the process of being migrated out of wordpress to become a static, plain HTML website hosted on AWS and a new tracking systems will be adopted.



## TIES

* Desktop tool available for personal and institutional groups (Linux and Windows OS). Java.
* Source code and documentation is hosted on Apiary and SourceForge. Some source code and documentation is hosted on [TIES website](http://ties.dbmi.pitt.edu/documentation/user-manual/getting-started/).
* Allow each TCRN group member to conduct their own analytics for their website and research

**Official Metrics:**

* [SourceForge](https://sourceforge.net/p/caties/discussion/626701/) tracks number of downloads and for bug maintenance with software. Also used for forums to help with initial set-up of TIES software

##

* [Apiary](https://apiary.io/) helps to view if something goes wrong with developers’ API calls and sends debugging reports
* [Noble Coder](https://github.com/dbmi-pitt/nobletools/graphs/contributors) tracks the number of NLP coder downloads and users

##

* Google Analytics to view day to day traffic on website and length of time spent on website
* Insightly to maintain user mailing lists of people using TIES software and to keep track of people who have downloaded a demo
* Internal TIES software audit to view number of users and sessions

**Unofficial Metrics:**

* MailChimp Analytics to view newsletter results
* Hootsuite analytics to read cross social media platform analytics (Twitter, LinkedIn, etc.)
* Youtube and vimeo for video watches

##

## TRINITY

* Primarily a linux command-line driven application, but available to users via Galaxy (<https://galaxy.ncgas-trinity.indiana.edu/user/login>). The code is written using a combination of C++, Java, Perl, and R, and Python. Recently we’ve included Docker to simplify access to and execution of the software: <https://github.com/trinityrnaseq/trinityrnaseq/wiki/Trinity-in-Docker>
* To monitor usage, we have the Trinity software do a live version check at runtime. When it checks for the latest version, it indicates what current version is running, and from the IP address of the caller, we can determine the geographical region. We also collect statistics regarding the use of our Galaxy portal at Indiana University, including numbers of new users and counts of software executions. An example report for version use by our version-checking routines is shown below:

Yes



* We used to track numbers of software downloads when we had our software hosted on Sourceforge, but since we’ve moved to GitHub, we haven’t found a straightforward way to do this, and so we’ve since been relying on the above version tracking for similar data.

## The Cancer Proteome Atlas (TCPA)

Han Liang <HLiang1@mdanderson.org>

· TCPA is a web-based data portal for accessing, visualizing and analyzing cancer functional proteomics data.

· The programing languages used in TCPA include JavaScript and R.

· We use Google Analytics (GA) as our official usage-tracking tool.

**Official metrics:**

TCPA is publicly available at<http://tcpaportal.org>. Our first release was in April 2013 and the Google Analytics (GA) was implemented in TCPA in Oct 2013. Since then, ~8,500 users have visited our website according to the records in GA (**Fig. 1**). Based on the records, the average session duration is ~4 mins, which indicates our application is not a quick reference for most of the users. The highest number of page views comes from the home page. The second highest one is the Analysis page which indicates instead of downloading the proteomics data from our download page, users visit analytic modules more often. Within the analytic modules, the page of survival analysis has the longest average time users spent indicating our users are more interested in the survival analysis compared to other analytic modules.

**Other metrics:**

 We have a dedicated e-mail address <tcpa\_info@mdanderson.org> for technical support of the web app as well as the data related questions and other inquiries. Since our first release, we have received many emails from our users all over the world.

 We also track the usage of TCPA by periodically checking the citation of our paper that was published in 2013 (Li et. al., *Nature Methods*, 2013). Since its publication, TCPA received 63 citations from peer-reviewed journal articles.

 We have recently moved to our new domain <http://tcpaportal.org> from the old landing page <http://bioinformatics.mdanderson.org/main/TCPA:Overview>. The main reason is the old one is very long which is bad for user experience. The number of page views recorded at the old one is >33,000.



Figure 1. An overview of TCPA tracking metrics shown in Google Analytics.

## LesionTracker

Gordon Harris <gjharris@partners.org>

· LesionTracker is an extensible, open-source, zero-footprint web-viewer for oncology clinical trials image assessment.

· The programing languages used include JavaScript and Mongo dB, as well as the Meteor framework and Cornerstone open-source DICOM libraries and tools.

· We are planning to integrate LesionTracker into our Precision Imaging Metrics web-based informatics platform, which manages the on-line ordering, assessment worklists, communication, reporting, billing, audit trails, protocol compliance, training and certification for oncology trials imaging. Precision Imaging Metrics is currently in use at six NCI-designated Cancer Centers around the U.S.

· We do not currently have a system for tracking usage of LesionTracker, although we are considering several options to implement for usage tracking.

· Precision Imaging Metrics is currently used to manage imaging assessments for over 1,500 active clinical trials, and over 20,000 imaging assessments per year across the six active participating Cancer Centers. We anticipate integrating LesionTracker into Precision Imaging Metrics in 2018, replacing our current PC-based open-source imaging assessment module.

· LesionTracker source code is available on our github site, which can be accessed on the developers’ page at [www.ohif.org](http://www.ohif.org)

· The LesionTracker application can be accessed at lesiontracker.ohif.org

· A summary video of LesionTracker is available at the NCIPHub ITCR website: <https://nciphub.org/groups/itcr/collections/itcr-training-assets>

In the past 18 months of our ITCR funding, we have spent about $450K on development, plus another $150K or so on project management, Q/A, testing, etc. During that time, I have also added $130K of development funding from our lab’s Precision Imaging Metrics software development funds, and Chris Hafey’s company has invested about $350K in the open-source platform, so the NCI development dollars have been essentially matched 1:1 by our academic and industry support funds

##

## Integrative Genomics Viewer (IGV)

**Contact:** Helga Thorvaldsdottir, helga@broadinstitute.org

* Desktop IGV
	+ Java based application
	+ Can be launched via JNLP from the web site, and is available for download for all major platforms (Mac OSX Windows, Linux)
	+ Source code hosted on GitHub
	+ [User documentation](http://www.broadinstitute.org/igv) on a Drupal site we host
* igv.js

o JavaScript component for embedding in web pages / web apps

o Source code hosted on GitHub

o  [Developer documentation](https://github.com/igvteam/igv.js/wiki) on GitHub wiki

* [www.igv.org](http://www.igv.org) : simple site that serves as the general IGV landing page and has a simple easy-to-remember URL. It directs users to the IGV Desktop site and the igv.js doc

* [Public help forum](https://groups.google.com/group/igv-help) via Google Groups. Developers also post issues on GitHub

**Metrics**

* **IGV launches.** Since May 2011, the IGV desktop application checks for IGV software updates when the program is launched. We use an Oracle database to store the IP address of the user, the IGV version, and a timestamp. We check the logs on a weekly basis and over the past year we have logged an average ~32,000 launches from ~8,500 different IP addresses per week. The IP addresses can give us an idea of geographic location of our users. With respect to the number of IP addresses, we note that several individuals from the same institution can have the same IP address, and the same individual may use IGV from different IP addresses. Recently we similarly started to log uses of igv.js.
* **User registrations.** From our initial release in August 2008 and until August 2016, we required users to register to access the downloads page on our website. During this period, we logged ~165,000 user registrations. We stored the user-supplied name, organization, and email address in an Oracle database, along with an automatically generated timestamp and the user’s IP address. A number of data/analysis portals (e.g. MSKCC cBio) include a link to launch IGV. We did not require these users to register on our site. **Since August 2016, we no longer require any registration.** Similar applications do not require registration, and we felt we had reached a stage where it was clear how widely used IGV is. We do still track the actual usage based on the IGV launches, and we get an idea of the number of new users each week based on the number of new IP addresses (as opposed to the total number of IP addresses logged for the week). Some of our other projects (e.g. GenePattern and GenePattern Notebooks) still require registration, as users need to have an account to use the service.

* **Citations.** We use a combination of methods to estimate the number of citations for IGV. We use the Web of Science to find the number of publications that cite our papers describing IGV. Unfortunately, many papers do not cite their use of software tools or only do so by referring to them by name or by referring to the tool’s website. These are more difficult to collect and count. We use a combination of Google Scholar alerts and manual inspection. This is labor intensive and we have not updated the list since the end of 2015. Our citation library currently includes ~3600 publications, of these ~2400 from the Web of Science, with an additional ~1200 from our curated list. We initially tried using only the numbers returned by Google Scholar to estimate our total citation count, but found the numbers to be artificially high and it was not easy to find and eliminate the duplicates and invalid entries.
* **Help questions and bug reports.** We sometimes report on number of members in our igv-help Google Group, the number of postings there, and the number issues reported on GitHub. We remain unsure how useful these statistics are.

* **Website usage.** We have started experimenting with Google Analytics (GA) to monitor website usage. There are also open-source tools available for scraping the web logs for usage statistics.

## CRAVAT (Cancer-Related Analysis of VAriants Toolkit)

**Contact:** Michael Ryan, mryan@insilico.us.com

* **Web Server** - CRAVAT was initially implemented and is still heavily used as a web-based service available on our servers (http://cravat.us). Users submit large sets of genomic variants for scoring / annotation and can then use our browser-based interactive results viewer to sort / filter / explore their results.
* **Docker Image** - In the past year, we have made the entire application available as a Docker image that can be used to instantiate a local CRAVAT server. This is popular with groups that have protected data or that want dedicated compute resources and control of results retention policies.
* **Programmatic Interfaces** - Many CRAVAT annotations / services are available via web services calls that integrate easily into other applications. External systems can also integrate with our interactive 3D structure visualizations via http links. Finally, we provide Galaxy tools that are able to perform many of the CRAVAT analysis function in the open source Galaxy bioinformatics platform.

**Metrics**

* **Job submission logging**. The public CRAVAT server was fairly easy to instrument via custom logging that tracks submission of user jobs. This logging is useful not only to evaluate how many jobs are run and the number of mutations processed but also to provide feedback on which types of analysis are more popular and which input formats are more common.
* **User registration**. User registration is optional for our Web Server but you need to register to get access to our interactive / graphical results browser. We have emails and high-level usage statistics for our user base.



* **Google Analytics**. Both the main CRAVAT service and the MuPIT 3D structure viewer use Google Analytics to collect usage statistics. Google analytics provides good background information on site usage, user demographics, user behavior, and in the case of integration with other tools, tracks ‘Referrals’ under ACQUISITION’ to give a picture of which external partnerships are driving usage.
* **Docker**. We don’t know much about docker usage other than the number of pulls for our image in Docker Hub. Many users that install their own local instance have data privacy policies so it did not seem to be a good idea to leave Google analytics and/or job logging enabled. As more users run Docker versions of CRAVAT, we start to loose some of the feedback we get from the public Web Server.
* **Web Service Stats**. One of the key aspects of a web service is that it be very fast. This makes it difficult to log usage if logging will slow the service down. We opted for a custom logging module that tracks hits with in memory counters and then writes out totals every ~1000 hits. We could lose a bit of usage information if the system comes down hard but this is likely to be infrequent and the current performance is excellent.

## GenePattern

**Contact**: Michael Reich, mmreich@cloud.ucsd.edu

**GenePattern public servers**

Freely-available servers, currently with over 40,000 registered users hosting hundreds of genomic analysis and visualization methods:

* Broad Institute: <http://genepattern.broadinstitute.org>
* Indiana University: <http://gp.indiana.edu/gp/>

**Metrics**: a cron job runs weekly on each public GenePattern server, querying the GenePattern analysis and registration databases to determine:

* Number of analysis jobs run
* Number of new users registered
* Number of returning users who ran at least one analysis
* Links to output reports for analysis jobs that resulted in errors
* Future: we are interested in adding information about the run time of each analysis as well as the time that analyses spend in the queue before being executed. This will allow us to better utilize load balancing resources and communicate expected wait times to users.

**GenePattern Notebook Repository**

An online repository and workspace where users can develop their own GenePattern notebooks:

* <http://notebook.genepattern.org>

**Metrics**: a cron job runs weekly on the notebook repository, reporting on:

* Number of new users registered
* Number of logins
* Number of new notebooks created
* Repository disk space used

**GenePattern Notebook Python extension**

A Python package that implements GenePattern Notebook functionality within the Jupyter Notebook environment, turning a Jupyter notebook into a GenePattern notebook. It is installable via the standard pip and conda package managers or as a Docker image.

* <https://pypi.python.org/pypi/genepattern-notebook>

**Metrics**: weekly report on:

* Number of jobs launched from a GenePattern notebook on a public GenePattern server. This is to determine how many analyses are run on the “classic” GenePattern interface vs. the GenePattern Notebook environment. We modified the GenePattern RESTful API to include a “point of origin” parameter that identifies whether a job was submitted through the GenePattern Web UI, GenePattern Notebook UI, a programming language interface, or another source.
* Number of Dockerhub image downloads
* Number of PyPI downloads

**Web sites**

Web sites hosting information for the GenePattern and GenePattern Notebook projects: project description, tutorial materials, knowledge base, links to further resources, etc.

* <http://www.genepattern.org>
* <http://www.genepattern-notebook.org>

**Metrics**: weekly report on:

* Total traffic (non-crawler)
* Number of unique visitors
* Top 20 pages requested
* Top 20 requesting domains
* Number of downloads of GenePattern installable local server
* Top 20 analysis module documentation pages visited

**GParc: GenePattern Archive**

Repository of community-contributed GenePattern modules, available for any GenePattern user to install

* <http://www.gparc.org>

**Metrics**: weekly report on:

* Number of new users registered
* Number of returning users
* Number of analysis modules downloaded
* List of analysis modules downloaded

Andrey TODO

* Summarize for CellProfiler? <http://cellprofiler.org/about/>
* What other high-profile tools should we consider? ProteoWizard?

# Related references

* Katz DS., Niemeyer KE., Smith AM. 2016. *Strategies for biomedical software management, sharing, and citation*. PeerJ Preprints. DOI: [10.7287/peerj.preprints.2640v1](http://dx.doi.org/10.7287/peerj.preprints.2640v1).
* Smith AM., Katz DS., Niemeyer KE. 2016. Software citation principles. *PeerJ Computer Science* 2:e86. DOI: [10.7717/peerj-cs.86](https://peerj.com/articles/cs-86/) and <https://www.force11.org/software-citation-principles>.
* Singh Chawla D. 2016. The unsung heroes of scientific software. *Nature* 529:115–116. DOI: [10.1038/529115a](http://www.nature.com/news/the-unsung-heroes-of-scientific-software-1.19100) (also see <http://depsy.org/>).
* <https://github.com/duecredit/duecredit>
* Thain D., Tannenbaum T., Livny M. 2006. How to measure a large open-source distributed system. *Concurrency and computation: practice & experience* 18:1989–2019. DOI: 10.1002/cpe.1041.
* Highly relevant blog article: <https://danielskatzblog.wordpress.com/2015/10/29/software-metrics-what-to-count-and-how/>

# Summary

## Tools overview

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Tool name** | **Established** | **Type** | **Extensible by non-core developers?** | **Home page** | **Recommended acknowledgment** |
| 1 | UCSC Xena | 2013 ([first message in user archives](https://groups.google.com/forum/#!topic/ucsc-cancer-genomics-browser/duubaErgF2Y)) | Central web server, reusable private web server | Yes | <http://xena.ucsc.edu/>  | N/A?! |
| 2 | 3D Slicer | 2000 ([first year of user archives](http://massmail.spl.harvard.edu/public-archives/slicer-users/)) | Desktop application, extensible via plugins | Yes | <http://slicer.org>  | [Paper citation](https://www.slicer.org/wiki/Documentation/4.x/Acknowledgments) |
| 3 | Bioconductor | 2002 ([first annual report](http://bioconductor.org/about/annual-reports/)) | Desktop application, extensible via packages, [collection of Docker images](https://hub.docker.com/u/bioconductor/), AMI | Yes | <https://www.bioconductor.org/>  | [Paper citation for the whole project or individual package](https://www.bioconductor.org/help/faq/) |
| 4 | NDEx | 2015 ([citation paper published](http://www.home.ndexbio.org/about-ndex/)) | Central web server, reusable private web server | No? | <http://www.home.ndexbio.org/>  | [Paper citation](http://www.home.ndexbio.org/about-ndex/) |
| 5 | TIES | 2006 ([first message on user forum](https://sourceforge.net/p/caties/discussion/626701/thread/d75e1ce6/)) | Desktop tool | No? | <http://ties.dbmi.pitt.edu/>  | [Funding grant citation](http://ties.dbmi.pitt.edu/about/publications/) |
| 6 | TRINITY | 2011 ([year of the recommended reference](https://github.com/trinityrnaseq/trinityrnaseq/wiki)) | Desktop tool, also available via Galaxy, [Docker](https://hub.docker.com/u/trinityrnaseq/) image | No? | <https://github.com/trinityrnaseq/trinityrnaseq/wiki>  | [Paper citation](https://github.com/trinityrnaseq/trinityrnaseq/wiki) |
| 7 | The Cancer Proteome Atlas (TCPA) | 2013 (date of first release) | Central web server | No | [http://tcpaportal.org](http://tcpaportal.org/)  | [NA](http://tcpaportal.org/about.html)? |
| 8 | LesionTracker | 2015 ([first commit to the source code repository](https://github.com/OHIF/Viewers/graphs/contributors)) | Reusable web component, available as a source code, plugin to Orthanc | N/A | <https://github.com/OHIF/Viewers>  | NA? |
| 9 | Integrative Genomics Viewer (IGV) | 2011 ([year of the citation](http://www.igv.org/)) | Desktop tool and a reusable web component | ? | <http://www.igv.org/>  | [Paper citation](http://www.igv.org/) |
| 10 | CRAVAT | 2013 ([year of citation](https://www.cravat.us/CRAVAT/help.jsp)) | Central web server, reusable private server, [Docker image](https://hub.docker.com/r/karchinlab/cravatmupit/) | N/A? | <https://www.cravat.us/CRAVAT/>  | [Paper citation](https://www.cravat.us/CRAVAT/) |
| 11 | GenePattern | 2006 ([year of citation](http://software.broadinstitute.org/cancer/software/genepattern/FAQ#cite)) | Central web servers, Python notebook ([also in Docker images](https://hub.docker.com/u/genepattern/)), extensible via GenePattern modules | Yes | <https://genepattern.broadinstitute.org/> | [Paper citation](http://software.broadinstitute.org/cancer/software/genepattern/FAQ#cite) |

## Impact metrics

Common metrics:

* Website usage. Measured through Google analytics/Mixpanel/other. Even if tool is based on the desktop, every tool has a web presence and all tools measure this web traffic. However the measurements that each tool looks at varies greatly, depending greatly on the tool and its individual architecture
* Downloads and/or 'phone home' counts. For desktop applications only. This has been a difficult thing to measure for docker containers. Also, some concern exist about the danger of perceived risks to data privacy, especially for the tools that can be applied to clinical data with PHI present.
* Outside contributions to the tool or around the tool. Plugins, packages, direct code contributions by folks not paid by you, other auxiliary contributions (e.g. someone outside of Xena wrote an R package to download our data directly from a hub)
* Active user engagement outside the direct use of the application. These are interactions that require real effort and/or commitment on the part of the user. This can include mailing list/support forum questions, user registration, conference attendance, workshop attendance, etc.
* Passive user engagement. These are interactions that 'cost' the user very little. Users engaging in this manner will range from power users to users who think the tool is interesting and are hoping to really dive in soon. This is measured in a variety of ways depending on the platform: twitter followers, youtube views, email newsletters opened, github stars, etc
* Publication citations. This seems important but no standardized way to do this and some folks don't measure it at all.

Important points to make in the overall discussion about this:

* Distinguish metrics that are available publicly and easy to access (download summary), publicly but not easy to access/aggregate by an outsider (YouTube view count, user list subscribers), and private - those that are available only to the core team (Google Analytics, registrations, logins, job count)
* Distinguish between monolithic applications and those that are extensible with plugins etc, and highlight the challenges of monitoring usage and acknowledging contributions of non-core developers
* Distinguish between centralized web servers (full and easy control over usage) and other types of technology (source code components, desktop tools, docker images)
* Distinguish different ways of using tools, which may be dependent on the tool's functionality. E.g. New users vs returning users. Number of visits vs length of visit.
* Discuss publication citations in papers. Difficulty of doing this and what strategies people use. Additionally, what if people use your tool in the discovery phase of research and don't cite you?
* Discuss the difficulty in measuring all types of metrics in the cloud.

### Users

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tool** | **Registration required to use** | **Tool usage count** | **Support forum user count** | **Download count** |
| UCSC Xena | No | Google analytics, AWS stats;“Phone home” for private hubs (neither are publicly exposed) | Can be checked (google group) | For private hubs (not exposed?) |
| 3D Slicer | No | No | Can be done but not tracked routinely | Yes ([officially tracked for main application](http://download.slicer.org/download-stats), tracked but not easily available for plugins) |
| Bioconductor | No | No | Yes | Yes, both for the [overall app](http://bioconductor.org/packages/stats/bioc/index.html), and [packages](http://bioconductor.org/packages/stats/) |
| NDEx | Yes, for central server | Surrogate measure: number of network uploads | No | No? |
| TIES | No | Yes (not publicly exposed?) |  | Yes |
| TRINITY | No | “Phone home” (not publicly exposed?) | No | No (due to limitation of GitHub infrastructure) |
| The Cancer Proteome Atlas (TCPA) | Yes | Google analytics (not publicly exposed); inquiries to dedicated email (not publicly exposed) | No | N/A |
| LesionTracker | No | No | No? | N/A |
| Integrative Genomics Viewer (IGV) | No more, as of Aug 2016 | “Phone home” (not publicly exposed) | Occasionally (not publicly exposed?) |  |
| CRAVAT | Optional | Yes, Google Analytics and job submissions to central server (publicly exposed?) | No? | Can be tracked on DockerHub |
| GenePattern | Yes | Interactions with the server are logged (not publicly exposed?) | No | Yes (not publicly exposed?) |

Additional metrics that could be tracked for some of the tools:

* DockerHub pulls

### Developers

Contributors, non-contributors using the code base, developers support list usage

Additional metrics that could be tracked for some of the tools:

* GitHub contributors to the source code
* GitHub contributors the discussions (issues etc)

### Academics

Do all tools have a publication?