Tranche Distributed Repository and ProteomeCommons.org

Bryan E. Smith, University of Michigan, besmit@umich.edu

James A. Hill, University of Michigan, augie@umich.edu

Mark A. Gjukich, University of Michigan, markgj@umich.edu

Philip C. Andrews, University of Michigan, andrewsp@umich.edu

Departments of Biological Chemistry, Bioinformatics, and Chemistry

Abstract

Tranche is a distributed repository designed to redundantly store and disseminate data sets for the proteomics community. It has several important features for researchers, including support for large data files, pre-publication access controls, licensing options, and ensuring both data provenance and integrity. Tranche tightly integrates with ProteomeCommons.org, an online community resource that offers a variety of useful tools for proteomics researchers, including project management and data annotation. In this chapter, we discuss the development of Tranche and ProteomeCommons.org, paying particular attention to why it is desirable that data be publicly available and unrestricted as well as the challenges facing data archiving and open access. We then provide a technical overview of Tranche and ProteomeCommons.org as well as step-by-step instructions for using these resources, including the GUI (graphical interface), command-line tools, and Application Programmer Interface (API). We end with a brief discussion of current and future development efforts and collaborations.

Key words

annotation, archiving, mass spectrometry, open access, proteomics, public access, project management, repository

Introduction

ProteomeCommons.org was developed and released in 2004 to provide data and software hosting to the proteomics community. Its founding goals were to support data and software re-use and dissemination as well as integration with other proteomics resources. ***(1)*** Data sets that were hosted on ProteomeCommons.org were available for download via HTTP. While this was a simple and immediately valuable service, a more scalable solution was needed to handle the raw data that is produced by high-throughput mass spectrometry instruments. Furthermore, given the unpredictable nature of hardware failures, redundancy was necessary for the long-term archiving of the hosted data sets. In response to these and other needs, the Tranche distributed repository project was developed in 2005 and publicly released at the 2006 American Society for Mass Spectrometry (ASMS) conference.

Proteomics data sets tend to be large, expensive to generate, and contain information beyond the immediate needs of the original investigators. There are many reasons to make data publicly available; public access to data sets protects the interests of peer review (e.g., critical evaluation and replication of results) as well as the future re-evaluation of data sets as new analytic tools and additional data sets become available. Another important reason to make data publicly available is to satisfy dissemination requirements for funding agencies.

Recent calls for broader data sharing have cited the genome project and other research efforts to share data. ***(2) (3) (4)*** The role of both pre- and post-publication data sharing in the genome project was particularly important in accelerating progress, allowing more rapid development of new tools, and providing broad dissemination of genome data which increased the impact of the genome project. Pre- and post-publication data sharing have different uses and some differences in data infrastructure, with the former usually requiring some degree of security in the form of encryption or limited data access.

The general infrastructure for data sharing has been limited, particularly in reducing the barriers to getting data into repositories and in the dissemination stage. ***(5)*** The resources available for building the infrastructure have been limited and decision making has been hampered by economics, the volume of data, and the rapidly changing technologies. While there is general consensus that sharing data sets is desirable, the technical and social challenges are significant and pessimism has been expressed over the feasibility ***(6)*** or the ultimate usefulness of data sharing in all cases. ***(7)*** As these authors point out, the cost benefit aspects of data sharing, the time lag involved in building infrastructure, obsolescence times, intellectual property issues, and many other concerns affect development of an effective data sharing infrastructure. Confounding this situation are the unique data features that must be accommodated in many fields of research. These concerns are being addressed by the proteomics community in several ways beyond Tranche and Proteomecommons.org. For example, centralized resources like Peptideatlas.org, ***(8)*** TheGPM.org, ***(9)*** PRIDE, ***(10)*** HPRD, ***(11)*** and Peptidome ***(12)*** all harvest data and place various levels of metadata in their databases for easy mining and access for investigators. Many of these resources download data sets from Tranche and some of these resources run the data through their own data pipelines to allow for improved comparisons across data sets.

Currently, some journals recommend public release of proteomics data sets. As of April 2007, *Molecular and Cellular Proteomics Journal* and *The Journal of Proteome Research* officially recommends depositing all mass spectra output data as supplemental material associated with protein identifications ***(13)*** following the recommendation of a group of leaders in proteomics in March 2005. In an editorial in 2007, *Nature Biotechnology* ***(14)*** recommended that all proteomics data associated with manuscript submissions be deposited in public repositories, sharing many of the concerns we will outline in this introduction. The submission requirements for the journal *Proteomics* (last updated in 2008) state that peak lists should be deposited in a public repository and submitted as supplemental material, though there is no statement that these peak lists are required or recommended. ***(15)*** Funding

agencies, on the other hand, are beginning to stipulate the submission of data sets to repositories. As of October 2003, any investigators receiving NIH funding of $500,000 or more within a year are required to provide a data sharing plan; if sharing the data is not feasible, the investigator must explain why. ***(16)***

There are many barriers to the general adoption of data sharing. Foremost, we should consider the motivations of individual researchers. For example, researchers might not share data due to the understandable perception that ad hoc experimental design and the lack of experimental standards might compromise the re-use of the data sets. ***(7)*** This is partly addressed by the proper annotation of data sets. While publications provide a depth of information about a data set that is difficult to capture by other means, they are not ideal as primary annotation sources because they are often incomplete, and the annotations are provided in natural language, which is a challenge for data mining applications. This situation is further improved when the experimental parameters are properly stored as metadata. Researchers also may not be sufficiently motivated to share their data when it is neither required nor sufficiently rewarded, which could change if funding and career advancement incorporated not only the generation but the public availability of properly annotated data sets. ***(17)*** Relatively minor shifts in faculty evaluation processes could go a long way in enhancing the public availability of scientific data. Funding agencies, publishers, and professional societies have an opportunity to influence this process through their policies.

Even where there is sufficient motivation to share data, there can be additional obstacles. The investigator may not be prepared to release the data set until all interpretations have been completely exhausted, feeling that unanalyzed or under-analyzed data remains; this is true despite the likelihood that most data sets will not be further analyzed in the original laboratory beyond the primary goals due to priorities and resource constraints. In the case that multiple publications are prepared using the same data set, the author might wish to withhold the data set from public release beyond the first publication.

This situation can arise when funding or promotion deadlines require early publication or when logistics interfere with timely follow-ups to initial publications. Some investigators may also have concerns about overlooked knowledge that can be derived from the data sets by other investigators with different perspectives or computational algorithms that might result in "lost" intellectual property. One response to this concern is that this is one of the anticipated and desired outcomes of data sharing that will result in faster progress in medical research and advancements in treatment. This issue is analogous to publication of a manuscript which may contain data sets subject to alternative interpretation and is one of the reasons for publication. It might also be addressable by more general access to computational tools. Uncertainty about interpretation of one's own data sets can also be an inhibitory factor; however, the peer review process exists to help authors confirm that their interpretations are reasonably accurate.

Even when funding sources and publishers do not require that data is publicly released, there is considerable value to the broader community fully understanding and embracing the value of open access to scientific data. This should be particularly clear from the field of genomics, considering how the availability of large genome databases has led to the development of multiple new fields of post-genome research, including proteomics. ***(18) (19)***

It is also important to recognize that just because a data set is *available* does not mean it is *usable,* which generally requires that the data set is unrestricted (or minimally restricted) and that it has the appropriate metadata so that it is *findable* and can be placed in an interpretable experimental context. This leads to two important topics: open data and annotations.

When data sets are unrestricted and freely available, they are said to be *open*. This unambiguously allows anyone to freely use the data. Restrictions primarily come in the form of copyright law, though other legal or ad hoc restrictions might apply. This is a complicated topic that will change over time as intellectual property continues to be defined. What is interpreted as a creative work, and hence

protected under copyright law, varies by jurisdiction. (For example, Database structure might be defined as a creative work, though the same may not be said of the underlying data; however, there is also the database right of the European Union, which is similar to copyright law but protects the underlying data.) Attribution is a related right, and many available licenses include attribution stipulations. In the research community, attribution is a de facto requirement, even when data are not legally protected. Since it is likely that data sets with the least licensing restrictions will see the broadest impact and citation, it is important to retain attribution without restricting the data. Because all data on Tranche is digitally signed upon upload, Tranche inherently supports provenance.

The uncertainties and difficulties associated with ascertaining the appropriate restrictions suggest the value of clearly-defined open data. One tool that protects open data is the Creative Commons CC0 waiver, which is a "no rights reserved" option that places data in the public domain as completely as is feasible. ***(20)*** CC0 also provides a machine-readable document, so that automated agents can recognize that the associated data can be used without restriction. The association of a machine-readable license or waiver is becoming increasingly important as more tools are developed to aggregate and analyze data.

The most practical justification for open access to data is for the re-evaluation of data sets. Without public data sets, there is no foundation for a broader bioinformatics community, which has the potential to contribute discoveries that require subtle statistical analysis of many data sets as well as develop improved algorithms. By comparison, consider how public genetic databases have provided essential evidence for post-transcriptional gene regulation as well as for the discovery of co-expressed genes. ***(21)*** Furthermore, the cost of producing high quality data sets can be quite high, so encouraging data re-use could prove to be an economical choice for limited proteomics research. This is particularly important when considering biological samples that are rare or difficult to obtain (such as with wildtype

gastrointestinal stromal tumors ***(22)***) or unique (for example, the tyrannosaurus rex collagen protein ***(23)*** and hadrosaur proteins ***(24)***). Mining these data sets to plan future experiments is another application of Tranche that allows investigators to estimate the variance, dynamic range, and other parameters important when optimizing experimental design. Additionally, aggregation of data sets from multiple studies can be used to improve statistical models.

The issue of whether data is findable is a key issue most directly addressed by providing the appropriate metadata in a searchable format, otherwise known as *annotating* (or *curating)* the data set. Much of the data currently generated in proteomics and related fields is not thoroughly annotated, which compromises the value of the data sets; however, as this issue is remedied, future experiments can be designed with more insight, meaning that researchers can be more productive. ***(17)*** As we will discuss later, *semantic searches* are generally more useful than keyword searches, and metadata that is highly structured offers more value, particularly when a controlled vocabulary is used.

Annotations, which provide the context for the data sets and hence aid their findability, involve two separate issues. The first is the definition of the annotations. The Human Proteome Organization (HUPO) Proteomics Standard Initiative (PSI) has developed the Minimum Information About a Proteomics Experiment (MIAPE) standard to specify the minimal metadata that should accompany proteomics data. ***(25)*** The second issue is compliance: annotations must be completed and accurate for maximal impact. Considering the potential size and complexity of these annotation standards as well as the long-term collaborative nature of proteomics projects, this is not a trivial task. For example, MIAPE offers separate modules for various stages and technologies related to a proteomics experiment, such as study design and sample generation, mass spectrometry, gel electrophoresis, and others. [http://www.psidev.info/miape/]. Each of these modules contain multiple categories of required information, with each category containing multiple related fields. Gathering all the required

information typically requires input from several individuals who collaborated in the study. Even in the case that software is used to extract as much information as possible from the data files, much of the annotation will need to be performed manually.

There are many technical and logistical challenges involved with disseminating and archiving large data sets. Foremost, there must be sufficient disk space to hold mass spectrometry data sets, which can be quite large. Current mass spectrometers can generate up to a 1 GB (or more) per hour of compressed data. ***(26)*** For example, the Thermo Fisher Orbitrap can produce up to 100 MB per hour while Brucker Daltonics' Micro TofQ can produce up to 500 MB per hour. ***(27)*** Additionally, server hardware failure, in the absence of redundancy, will generally result in data loss. Multiple disk failures over time can wipe out multiple replications; in the absence of a scheme to reintroduce redundancy, every data set will eventually be lost.

In addition to maintaining a valid copy of the data, the challenge of being able to access the data still remains. Changing storage media technologies (e.g., 9.5” floppy, 1.5MB floppy, zip disk, CD, DVD, flash drives, etc.) can make accessing data that is even only a few years old difficult. Additionally, there is the issue of the large and constantly evolving variety of mass spectrometer output file formats. Several standard formats (mzXML, analysisML and most recently, mzML) have been developed to deal with this problem, though their long-term success will likely be determined by their adoption by mass spectrometer manufacturers as well as the development of appropriate tools to convert from existing native formats. This is not only an issue for the long-term archiving of data, but will also present a continuing challenge for researchers.

Long-term preservation of data also requires an effective infrastructure. This begs the question: who is responsible for providing that infrastructure? The choices include federal agencies, the universities, individual researchers, and private industry. Each of these choices has its strengths and weaknesses and

no single solution is clearly applicable across all fields, applications, or even data types. One advantage of distributed storage systems like Tranche is that it provides the potential for all of these stakeholders to participate in supporting a common data infrastructure through investments in hardware and sharing the ongoing costs of maintenance and administration.

These issues, to varying degrees, have directed the development of both Tranche and ProteomeCommons.org. Dissemination and documentation requirements of data sets are still being defined by funding sources as well as journals, and issues like annotation standards are attracting the attention of standards organizations within the proteomics community, but these issues are far from settled. Despite the ongoing developments, the value of these resources has already been clearly demonstrated: during the six-month period starting in February through the end of July, over 3.9 TB of data were downloaded from the ProteomeCommons.org Tranche repository.

Below, we discuss how Tranche and ProteomeCommons.org were developed, including what we have learned during the process of hosting and sharing data. We also discuss how to get started using Tranche and ProteomeCommons.org. We will conclude with planned development efforts that will further address the challenges and issues outlined.

Methods

Tranche is a distributed repository designed using principles from peer-to-peer networking (redundancy and load balancing) combined with client-server architecture (authentication and reliability), which can best be described as a *distributed server model*. Data sets are uploaded to and downloaded from our network using any of our client tools. Figure 1 shows a screenshot of our graphical interface (GUI) with a list of data sets available for download.

[Fig 1 near here]

Our model is strongly decentralized--in the event that any number of servers are offline, the remaining servers can still provide service (though some data might be temporarily unavailable). The network is *federated*, as every server has its own list of trusted users and can be managed separately.

The key to data availability on a Tranche network is redundancy. Like a RAID array, we assume that servers will fail, so every chunk of data that is uploaded must be replicated a minimum number of times. Choosing the proper number of replications is complex and not easily modeled. Hardware failures and server downtime are generally unpredictable, with an innumerable set of factors, including lightning strikes and rodent damage as well as staff turnover and funding. The more servers that are online (to accommodate more data), the more redundancy that is required to accommodate the additional uncertainty. In other words, total number of servers online should be proportional to the number of replications. (With our current model, an increase in the number of servers requires a linear increase in the replications for data to remain available; the network model for the next version of Tranche, however, will only require a logarithmic increase in replications to accommodate additional servers.) In the case that it is rare for more than two servers to be offline, then three replications should be enough. The ProteomeCommons.org Tranche network currently requires three replications when data are uploaded.

Load balancing is another strength of using multiple servers. This allows multiple servers to share the burden of user requests. As a heuristic, during an upload or a download, a client will select a server on a moment-by-moment basis based on how much outstanding work the server has combined with its latency. This is a simple implementation of the 'nearest neighbor', where the cost of using a server is approximated based on recent performance.

Tranche does not store intact files on servers. Instead, for reasons of performance and scalability, a file is separated into data chunks, which have a maximum size of 1MB. (For example, a 5.3MB file would

require five 1MB-data chunks and one 0.3MB data chunk, for a total of six data chunks.) All the data chunks in a file are described by a metadata chunk. Given a metadata chunk, all data chunks can be downloaded and reassembled into the original file.

[Fig 2 near here]

A data set is any directory and all of its contents; in Tranche, there are no structural requirements imposed on a submission. Figure 2 demonstrates how a data set is stored on Tranche. Note that the individual files (each with one metadata chunk and associated data chunks) are described by a *ProjectFile*, which points to the metadata chunks for each file as well as describes their location relative to the data set's root directory. Just like any other file, the ProjectFile has a metadata chunk and data chunks; you can download and view the ProjectFile just like any other, though it is intended to be used behind-the-scenes by the download tool or for other client applications. (In Figure 2, the ProjectFile and its constituent metadata chunk and data chunks have a thick outline.)

Tranche servers store chunks in a b-tree structure, meaning that insertions, searches and deletes all run in logarithmic time, $O(\log n)$. This b-tree structure is made up of hierarchically-arranged nodes, each containing a maximum of 1,000 chunks, with 256 nodes branching from each parent node. The tree is rebalanced as data is added, which is important since most servers will hold millions of chunks. For example, assume the average data chunk is 400KB (which is not too far from what we have observed), and that its associated metadata is 4KB. A 4TB server could then hold, on average, over 21 million chunks. If we were to search for a particular chunk, it would take around an average of four operations to identify the node containing the chunk. Though a node holds a maximum of 1,000 chunks, a node always branches to 256 new nodes, so the effective seek time for a node when n chunks are in the tree is $\log_{256} n$. At that point, a linear search of the header of the node will quickly identify the chunk.

Each node in the b-tree structure is stored as a separate file. One potential source of data loss is the

corruption of one of these “data block” files, such as might happen if the server is killed in the middle of a write operation. Upon starting up, each data block file is checked for corruption and repaired using data from other servers when possible.

There are other sources of potential data loss. On a large network, disk failures must be continually accommodated. Furthermore, chunks can be corrupted during transmission. The redundancy on the network is only maintained in the long-run if lost replications are repaired or reintroduced. To this end, each server spends time downloading desired chunks, deleting unnecessary chunks, and searching for corrupted chunks and replacing them.

The first and last activities help mitigate the above sources of data loss, though with considerable latency. (We will discuss a better solution when we discuss the future of Tranche.) It is worth mentioning an additional source of data loss: a malicious attack. An attack would probably only impact a single replication of any given chunk on a compromised server; but in the worst-case scenario in which an attacker gains authenticated access to the Tranche network, the attacker might delete some or all copies of a chunk. If this occurs, we have a separate Tranche network with different authentication that actively copies over any missing data that it has available.

As mentioned above, servers will attempt to download 'desired' chunks as well as delete 'unnecessary' ones. This brings us to two very important concepts: the *hash* and the *hash span*. Every chunk (and indeed every file and every data set) has a hash associated with it. These hashes are unique identifiers, and they can be recalculated at any time (i.e., they are deterministic). Each associated hash is generated by combining the four sources: the MD5, SHA-1 and SHA-256 hashes of the chunk along with the number of bytes. (Though it is possible that two separate chunks might have the same hash, resulting in a collision and hence data loss, it is highly unlikely. Since a hash is 76 bytes, the odds of randomly generating the same hash is 1 out of $1.06 * 10^{183}$.)

Since each chunk has an identifier, it is possible to allow a server to accommodate a portion of all the network by assigning it a range of desired hashes, which is the server's hash span. By analogy, this is similar to providing multiple lines to pick up reserved tickets to a concert based on the first letter of your last name. If there are going to be many people, you might want two lines: A-M and N-Z. If there are many reserved tickets, then there might need to be several lines with shorter ranges. Hash spans not only allow a network to find a chunk more quickly, but they allow each server to accommodate a portion of the network based on available disk space or any other factor. Additionally, an administrator can remove a hash span from a particularly troublesome server so that it will not likely receive many more chunks, but its current data will continue to be available. (A server can also be flagged as 'read-only' to entirely prevent any data from being stored on it.)

When a server is downloading 'desired' chunks, it is downloading chunks with hashes within its hash span ranges. If it is deleting an 'unnecessary' chunk, it is removing a chunk that does not belong to its hash span--but only if there are already enough copies on the network.

Figure 3 illustrates the storage of data sets and files as data and metadata chunks as well as the upload and download process.

[Fig 3 near here]

The upload tool processes each file in a data set separately. The tool must first identify the data chunks, and generate a metadata chunk so that the data chunks can be reassembled back into the original file upon download. Each chunk is then uploaded to three servers in the network. Preferably, the chunks will be put on three servers with hash spans that contain the chunk, but if this is not possible, then other servers will be heuristically selected for time efficiency. To complete the upload process, the tool generates a ProjectFile, which describes the contents of the data set.

Earlier we mentioned that each server has a list of trusted users. Tranche uses public-key cryptography

(using X.509 public key infrastructure), so that every chunk that is uploaded to a server is authenticated. If a user is not recognized, meaning their certificate is not signed by one of the appropriate Tranche certificates or has insufficient permissions, the request to store the chunk will be denied. The business of certificates is handled by the client tool; when the user logs in, a certificate is downloaded from our web server. Note that while authentication is required for uploads, it is not required for downloads.

The download tool essentially works in reverse. It starts with the hash for a data set, which is used to retrieve the ProjectFile's metadata chunk. From this, the ProjectFile can be downloaded and reassembled, which will provide a list of all the metadata chunks and the relative path for each file. Each metadata chunk, in turn, provides enough information to download and reassemble the individual files. When downloading any given chunk, the tool will simultaneously query the entire network with requests to determine which servers have the chunk. The first positive response will be used to retrieve the chunk, and all other requests will be canceled. While quite verbose, this has been experimentally determined to be the fastest method given our current average network load. Even considering that the client tools are highly parallelized, meaning at any given moment there are many requests for each client, these requests are quite small--and as described previously, searching for chunks on a server is quite fast. However, we will discuss a better solution when we discuss the future developments planned for Tranche.

Tranche offers several features that are useful for researchers:

- **Pre-publication encryption**: data sets can be optionally AES encrypted, and require a passphrase to download and decrypt. Upon publication, a user may 'release' a data set, which means it will become publicly available for download without a passphrase.
- **Data pedigree**: data sets are signed so that the individual who uploaded will always be known.

- **Data integrity**: since a hash can always be recalculated, data integrity can be verified at any time.
- **Immutability and versioning**: since Tranche uses hashes to determine data integrity, a data set may not be changed (*immutable*). However, new versions of the data set can be uploaded and linked with the previous data set version.

Three main interfaces are available for Tranche: the graphical interface, command-line tools, and the Java API. Investigators interested in downloading data sets can use these immediately; however, if you wish to upload data, you must register [https://proteomecommons.org/signup.jsp]. (Applications might take several days to process.)

The graphical interface (GUI) is of most interest to casual users. The GUI is launched using Java Web Start, which means that anyone with Java 5 (or greater) can use the tool simply by clicking on a link [https://proteomecommons.org/tranche/]. No installation is required.

[Fig 4 near here]

Once the user interface is loaded, it will begin loading information about available servers and data sets from the network. The full process may require several minutes for completion; you do not need to wait for the process to complete, but the process will consume a significant amount of your processor's time.

Figure 4 highlights four areas of the user interface. More detailed guides are available online. [https://trancheproject.org/users/] However, this figure covers the majority of tasks users perform:

1. **Log In**: Use your ProteomeCommons.org username and passphrase. Once logged in, the tool will handle all authentication for you, including your public/private key management.
2. **Upload Project**: A wizard will launch to walk you through the process of uploading a data set. You will be asked if you would like to encrypt the project, which legal license or waiver you'd

like to use (with the option to provide your own custom license), as well as several more advanced (and less frequently used) options.

3. **Download By Hash**: Downloads may be accomplished in several ways, but if you have a hash for a data set, then you can quickly download the project. This will launch a wizard to walk you through the download process, similar to the upload tool.
4. **Projects**: If you do not have a hash or do not know which data sets you want, you can browse the list of data sets. By selecting the project you want from the list, you can download the data set, view its contents, as well as access more advanced options.

(Note that ProteomeCommons.org also offers stand-alone versions of both the upload and download tools, as well as more advanced search and browse tools. The upload tool is offered on the member page when a user logs in to ProteomeCommons.org, and the download tool is launched when a user attempts to download a data set from the web site.)

The command-line tools are useful in several circumstances, e.g., when working remotely over SSH or in a "headless" environment (no windowing environment), or when there is a good deal of work to perform and you wish to automate some of the tasks. The upload [https://www.proteomecommons.org/tranche/files/CommandLineAddFileTool.zip] and download [https://www.proteomecommons.org/tranche/files/CommandLineGetFileTool.zip] tools are easy to use, but are heavily parameterized, so that users will likely want to start by simply noting the required parameters. The first thing you'll want to do is view the usage. For the upload tool:

```
java -jar -Xmx512m Tranche-Uploader.jar -help
```

Similarly, for the download tool:

```
java -jar -Xmx512m Tranche-Downloader.jar -help
```

Note that the command sets 512MB available for heap space. You may allocate more memory if it is

available, though this should be sufficient. Both the upload and download tools provide some advanced parameters which could potentially increase the speed (such as increasing the number of threads available for certain tasks, which increases the amount of work that is done in parallel). If these parameters are changed, it may be necessary to increase available memory! In general, advanced parameters should only be used when instructed by a Tranche developer to troubleshoot an issue.

Lastly, the API allows integration of Tranche into software or scripts, so long as Java is used or the appropriate language bindings have been established. Though the API is quite extensive, performing an upload or a download is simple. Figure 5 contains code for the simplest use cases.

[Fig 5 near here]

To perform uploads using the API, there must be an account registered with ProteomeCommons.org. All data will be signed by the associated user, so some care should be taken to select the appropriate user name, particularly if it represents a group or organization.

Tranche is used by ProteomeCommons.org to host data sets; in fact, it might be useful to think of ProteomeCommons.org as a layer of functionality that is built on top of Tranche. Many operations, such as deleting a data set and publishing a passphrase (which will allow an encrypted data set to be automatically decrypted for all users), can only be performed from ProteomeCommons.org, since Tranche users will not have sufficient permissions to do this from other client tools. ProteomeCommons.org also provides additional functionality that goes far beyond what Tranche alone can offer, including project management and annotations, as we will discuss shortly.

ProteomeCommons.org is an online community, offering public access to user-contributed news, publications, and software. Data sets are automatically added following an upload to the ProteomeCommons.org Tranche network. Figure 6 features a screenshot of the ProteomeCommons.org home page.

[Fig 6 near here]

Registered users can form groups and projects. Project management through ProteomeCommons.org allows members to share news, publications, tools, messages and data sets with privacy restrictions. Any project or group can be public or private, and private groups can be hidden from anyone who is not a member. Individual members of groups have finely-defined permissions, so that groups can establish rules for the management of all its resources. Subgroups and projects can be added to groups, offering additional control.

A particularly useful feature of groups is the management of annotation duties. Annotations are information that are associated with a data set describing how that data set was produced, processed and interpreted. The user selects an *annotation standard*, which is a set of categories containing requested fields. Generally, an annotation standard is defined by a standards body, like the previously mentioned MIAPE standard. Every available standard is versioned in the event of new releases. After selecting a standard, a user can edit the annotation set from the Proteomecommons.org annotation editor, as shown in diagram 7. Progress summaries are shown for individual categories as well as for the entire annotation set.

[Fig 7 near here]

Administrators of groups and projects can assign duties to members based on individual annotation categories allowing domain experts on the projects to be assigned responsibility for each category. This feature is optional and intended to promote annotation accuracy and completeness. Although domain experts may be assigned responsibility for each annotation category, any member with sufficient privileges can edit an annotation category field regardless of whether it is assigned to anyone or who it is assigned to. When a group member is annotating a particular annotation category, that category is locked to prevent concurrent modifications.

A data set is more findable in ProteomeCommons.org if it is accurately and completely annotated. Complete annotation information ranging from the nature of the biological sample to the mass spectrometer instrumentation appear on the data page generated by ProteomeCommons.org. This means the data set can be found more easily from an external search engine such as Yahoo! or Google, which index data set pages. Furthermore, anyone can search for data sets matching criteria, and only data sets with the appropriate annotations will be listed. Other non-browser interfaces can offer sophisticated semantic searches, which permits more useful bioinformatics applications. For example, the annotations manager was recently granted caBIG silver-level compliance [https://cabig.nci.nih.gov/], and users of caBIG will soon be able to search the annotations for data sets matching their criteria. There is also value in recording as much information about a data set as possible to provide the experimental context necessary for researchers to reinterpret the data sets and to complement the limited annotation often provided in publications describing the data sets. An archived data set missing basic information such as sample preparation or analysis conditions will have limited use.

To fully appreciate the value of annotations, it might be useful to contrast a semantic search versus a traditional keyword search. First, it is much easier to explore data sets when they are semantically defined. Though language is ambiguous, the use of ontologies not only offer controlled vocabularies, but also allow the definition of relations between data. For example, if an investigator wished to find all Tandem MS data sets involving yeast, a keyword search would only produce those that matched the descriptive text. The individual who performed the search could not be confident that everything was matched, and might perform additional searches to determine the relevant data sets. With ontologies, not only are the results unambiguous, but they can also be further divided by species of yeast or mass spectrometer manufacturer. Second, semantic searches offer the possibility of new functionality that is not possible with keyword searches. For example, assume that a data set has exactly 51,276 files. Human memory is rarely that specific, though it might remember that the data set had around 50,000

files. Semantically, it would make sense to search for data sets between 50,000 and 60,000 files. Combined with other information, such as the approximate date that the data set was produced, data sets become much more findable. At the present, ProteomeCommons.org search provides limited semantic search capabilities, particularly regarding data set size and dates of uploads; further functionality will depend on community standards and availability of completed annotations, but will continue to include keyword searches. caBIG searches of the ProteomeCommons.org annotations, however, will be entirely semantic.

The Tranche and ProteomeCommons.org development team is working with the broader proteomics community to adopt ontologies and controlled vocabularies. This is a long and difficult process, but is important for the development of reliable annotations, particularly considering the findability of data.

There are many more advanced features available with Tranche and ProteomeCommons.org, and these are documented on the Tranche Project website [www.trancheproject.org] and on ProteomeCommons.org.

Notes

The first version of ProteomeCommons.org was developed and released in 2004 as a web resource and service. Tranche was developed the following year and released in 2006 to address the specific needs of the proteomics community for storage and dissemination of data sets. Since 2006, development of both Tranche and ProteomeCommons.org has been based on the feedback from individual users, journals, and funding agencies as well as the anticipated needs of the community.

With over two years of operational experience and hundreds of users, several major changes have been made to Tranche to increase system reliability and robustness. We modified the functionality of Tranche to accommodate unforeseen events including a broad range of hardware failures. Much of our

error detection, such as detecting and repairing corrupted data files and chunks helped mitigate these problems. Problems that arose during production took longer to solve than it took to initially develop and release Tranche and involved extensive testing and development. System maintenance is necessarily a major effort for dissemination and archiving of data sets in a production environment.

We began redesigning ProteomeCommons.org in 2008, and the new version was released in February of 2009. Many features remain to added in response to user input. As indicated in the methods section, the annotation standards, ontologies and controlled vocabularies are still being defined, though two versions of the MIAPE standard are already available.

We are currently working on the second version of the Tranche Distributed Repository, which will address many challenges related to scalability, performance and security. One of the most significant developments will be the network model, especially the two specialized roles of Tranche servers: routers and data servers. As shown in diagram 8, a router will interface with any data servers to which it is connected, meaning the user will need to be connected to fewer servers. (A client may also make an unmediated connection to any data server, as is also shown in the diagram.) Furthermore, when uploading data, chunks are replicated by the servers, shifting responsibility from the client, further minimizing the number of required connections, and improving the performance by simplifying the protocol. This will require less user bandwidth, which will improve the overall performance.

In this new model, servers will have write permissions to other servers. This trust offers a particularly beneficial feature: if a server does not have a chunk that a client is requesting, that server can download the chunk from another trusted server, and then store the chunk and return it to the user. Not only will this result in a higher hit rate for servers having a requested chunk, thus improving the overall performance of the network, but it will also help keep the network more fully replicated. This will be particularly valuable for the long term availability of data sets.

Servers will store a change log of all activities that impacted their stored data. In the event that a server is temporarily offline, other servers will continue to record changes to the network. When the offline server starts up again and before it makes its data available to the network, it will request all logged activities from other trusted servers on the network. Note that chunks will always be accepted from trusted servers; however, if other servers log deletions, the credentials of the users who requested the deletes will be provided so that the querying server can ascertain whether it should also delete based on its own managed list of trusted certificates. Not only does this help with the overall replication of data on the network, but this also prevents deleted data from being salvaged by offline servers. Following these and other planned features, the Tranche network will be able to scale into the foreseeable future to accommodate all reasonable growth of users and data sets.

Several future developments are planned for ProteomeCommons.org. Since a good deal of this chapter was devoted to the ability of users to find data sets, we plan to add a simple HTTP "RESTful" interface for accessing resources on ProteomeCommons.org, which would permit users to develop more there own data mining applications using our resources. Also, we are aware of the effort that annotations require, and wish to lower the barrier to annotation as much as possible. We plan to add functionality to automatically read in as much information from data sets as possible and add the metadata parsed from the data files to the associated data sets. Additionally, we plan to provide export functionality to formats such as mzML and mzIdentML, so that stored annotations can be exported in a useful way from ProteomeCommons.org. Using these new tools that read and write tandem mass spectrometry data, we will provide more semantically useful information and statistics within ProteomeCommons.org and Tranche, as well as provide some limited file conversions. To increase the usefulness of data sets for the community, we will continue to work with publishers to automatically link publications with corresponding data sets in Tranche.

The ProteomeCommons.org Tranche network with PRIDE, PeptideAtlas, and Peptidome are founders of the ProteomExchange. ProteomExchange will allow free exchange of metadata between data resources, provide a universal accession number, and link to the raw data sets deposited in Tranche. Thus investigators will have to submit data for a study only once, and it will be available in all participating repositories and databases. ***(28)*** This collaboration is particularly beneficial for users since individual repositories may accommodate different types of data, based on their intended purpose. The ProteomExchange allows other resources to use Tranche to support their work, and in exchange, Tranche gains highly structured interfaces to its data.

Tranche and ProteomeCommon.org provide support for and maintain collaborations with a number of research entities, including Clinical Proteomic Technology Assessment for Cancer (CPTAC), the National Cancer Institute (NCI) Mouse Proteomics Technologies Initiative (MPTI), as well as with the PRIDE and PeptideAtlas repositories. We also have collaborated with Science Commons during the development and adoption of CC0, and have a current collaboration with the Personal Genome Project (PGP). The Tranche Project is responsive to the needs of the community and new collaborations are welcome.

Acknowledgments

Special thanks to Jayson Falkner, who led the initial development for both Tranche and ProteomeCommons.org. The authors would also like to thank Peter Ulintz, Jared Falkner, Brian Maso, and Panagiotis Papoulias for their contributions. The ProteomeCommons.org and Tranche Repository community resources are primarily sponsored by NCRR grant #P41-RR018627 and the NCI CPTC subcontract #27XS115. We also thank all the users of Tranche who have provided invaluable feedback and suggestions for the Tranche Project and Proteomecommons.org.

References

1. Falkner, J.A., Ulintz, P.J., and Andrews, P.C. (2006) A code and data archival and dissemination tool for the proteomics community. *American Biotechnology Laboratory* **38**: 28-30.
2. Toronto International Data Release Workshop Authors. (2009) Prepublication data sharing. *Nature* **461**: 168-170.
3. Schofield, P.N., Bubela, T., Weaver, T., Portilla, L., et al. (2009) Post-publication sharing of data and tools. *Nature* **461**: 171-173.
4. Editorial. (2009) Data's shameful neglect. *Nature* **461**: 145.
5. Salo, D. (2008) Innkeeper at the Roach Motel. *Library Trends* **57**: 98-123.
6. Heidorn, P.B. (2008) Shedding light on the dark data in the long tail of science. *Library Trends* **57**: 280-299.
7. Wiley, S. (2009) Why don't we share data? *The Scientist* **23**: 33.
8. Deutsch, E.W., Lam, H., Aebersold, R. (2008) PeptideAtlas: a resource for target selection for emerging targeted proteomics workflows. *EMBO Reports* **9**: 429-434.
9. Craig, R., Cortens, J.P., Beavis, R.C. (2004) An open source system for analyzing, validating and storing protein identification data. *Proteome Res* **3**: 1234-42.
10. Martens, L., Hermjakob, H., Jones, P., Taylor, C., et al. (2005) The PRoteomics IDEntification database. *Proteomics* **5**: 3537-3545.
11. Prasad, T.S., Goel, R., Kandasamy, K., Keerthikumar, S., et al. (2009) Human Protein Reference Database -- 2009 update. *Nucleic Acids Res* **37**: D767-72.
12. Slotta, D.J., Barrett, T., Edgar, R. (2009) NCBI Peptidome: a new public repository for mass spectrometry peptide identifications. *Nature Biotechnology* **27**: 600-601.
13. (2007) Publication guidelines for the analysis and documentation of peptide and protein

identifications. *Molecular & Cellular Proteomics.* (http://www.mcponline.org/misc/ParisReport_Final.dtl)

14. Editorial. (2007) Democratizing proteomics data. *Nature Biotechnology* **25**: 262.
15. (2008) Instructions to authors. *Proteomics*. (http://www3.interscience.wiley.com/cgi-bin/jabout/76510741/2120_instruc.pdf)
16. (2003) Final NIH statement on sharing research data. (http://grants.nih.gov/grants/guide/notice-files/NOT-OD-03-032.html)
17. Howe, D., Costanzo, M., Fey, P., Gojobori, T., et al. (2008) The future of biocuration. *Nature* **455**: 47-50.
18. Martin, D.B., and Nelson, P.S. (2001) From genomics to proteomics: techniques and applications in cancer research. *Trends in Cell Biology* **11**: 61-65.
19. Tyshenko, M.G. (2005) Current trends in publicly available genetic databases. *Health Informatics Journal* **11**: 295-308.
20. (2009) About CC0--"No Rights Reserved". (http://creativecommons.org/about/cc0)
21. Prince, J.T., Carlson, M.W., Wang, R., Lu, P., and Marcotte, E.M. (2004) The need for a public proteomics repository. *Nature Biotechnology* **22**: 471-472.
22. Why tumor samples are so important for research. (http://www.pediatricgist.cancer.gov/Source/Research/ResearchArticles/TumorSampleImpArticle.aspx)
23. Schweitzer, M.H., Suo, Z., Avci, R., Asara, et al. (2007) Analyses of Soft Tissue from Tyrannosaurus rex Suggest the Presence of Protein. *Science* **316**: 277-280.
24. Schweitzer, M.H., Zheng, W., Organ, C.L., Avci, R., et al. (2009) Biomolecular Characterization and Protein Sequences of the Campanian Hadrosaur B. canadensis. *Science* **324**: 626-631.
25. Taylor, C.F., Paton, N.W., Lilley, K.S., Binz, P., et al. (2007) The minimum information about a

proteomics experiment (MIAPE). *Nature Biotechnology* **25**: 887-893.

26. Pedrioli, P.G.A., Eng, J.K., Hubley, R., Vogelzang, M., et al. (2004) A common open representation of mass spectrometry data and its application to proteomics research. *Nature Biotechnology* **22**: 1459-1466.
27. Hamacher, M., Stephan, C., Meyer, H.E., and Eisenacher, M. (2009) Data handling and processing in proteomics. *Expert Reviews in Proteomics* **6**: 217-219.
28. Martens, L., Deutsch, E., Hemjakob, H., and Omenn, G. (2009) Proteomics data submission strategy for ProteomeExchange. (http://proteomexchange.org/doc/ProteomExchange_data_submission_strategy_final.pdf)