tranSMART 20.0 Roadmap (late 2020)

Project Summary

tranSMART 19.0 was a major project to update tranSMART to build the 16.3 release code with an updated version of GRAILS (2.5.4), to incorporate a code cleanup of the earlier 16.2 release by Paul Avillach's i2b2-tranSMART team at Harvard, to update the database schema to match the i2b2 schema for all the tables they have in common, to update the javascript stylesheets and images to use the asset pipeline plugin, and to review and improve the data loading procedures. Testing for this release is extensive because the major code revisions require everything to be validated.

Target developments

Grails 4.0 and Java 11

The upgrade to grails 2.5.4 takes tranSMART up to using Java8, but this is still no longer commercially supported by Oracle. To upgrade to Java 11 we need to migrate to the very latest (released in July 2019) Grails 4.0.

Grails 4.0 supports java 11 for the first time. It requires the use of the asset pipeline (already ported in tranSMART 19) and also requires the groovy and java source code to be reorganised. This is the same as the work required to port an earlier tranSMART branch (17.1) to grails 3. Code is moved to new locations, the BuildConfig files are rewritten, and a gradle build script is needed. The result is a far simpler and faster build environment for developers.

Database servers

Postgres 11, 12, 13 and beyond

TranSMART has supported Postgres releases from 9.2 up to 10 and 11 but has not exploited any of their new features.

Postgres now supports partitioned table (support started in Postgres 10 and is extended in Postgres 11). So far tranSMART has used a workaround for the large amounts of high-dimensional data for mRNA expression and RNAseq especially.

Some smal changes were needed to database creation and maintenance functions where the internal names of columns have changes. Old versions are also supported by the update functions.

It is comparatively straightforward to update the Postgres schema to use the officially supported partitioning. The stored procedures basically adopt the same methods as for the existing Oracle procedures. We could maintain the older Postgres schema under another name (PostgresLegacy) for any users with legacy installations.

SQLserver

I2b2 supports Oracle, Postgres and SQLserver. The database can be on a remote system using any of these. TranSMART initially supported only Oracle. When the open source tranSMART project started in 2012 Postgres was adopted as the database system, and Oracle code was neglected until release 1.1.

There are several benefits to adding SQLserver support. Existing i2b2 installations will be able to use tranSMART 20 together with their existing database to support and analyse clinical and high dimensional data from in progress and completed clinical trials. The SQLserver schema can be constructed using the existing transmart-data tools, and the existing comparison scripts to keep all the supported schemas in sync (Oracle, Postgres 11, Postgres 9+, SQLserver). Finally, this will provide an excellent test base for the remaining SQL code to ensure that we find any statements that are specific to only one platform.

Of course there could be some interest from the commercial database providers – Microsoft and Oracle – in supporting their products.

Oracle

There has been no change to the supported Oracle version since tranSMART was first released as open source.

Oracle 12.2 is now renamed 19c. We should test and update to support this version and the previously supported 12.0

Other database providers

Additional database providers have been suggested:

Clearly SQLServer should be the first, to be compatible with i2b2 and to work through the DBMS-specific areas of code.

MySQL is now owned by Oracle which is unfortunate given their approach with Java. There is also an open source derivative of MySQL 5.5 as MariaDB that aims to track Oracle's future changes and is a
better candidate for support - though presumably it would a a small overhead to support both.

Various "big data" databases are worth looking into.

MongoDb is used by code from Sanofi that is integrated in the Browse tab for indexing and retrieval from large files.

Standard R server installation

TranSMART installs R. This requires site administrators to download the latest R release and install from source. Over the life of a tranSMARt release teher can be a number of changes to the behavior of R, with older versions breaking so that it becomes necessary to change tohe makefile to load a more recent version of R or of one of the packages which we depend on. These can break tranSMART.

One workaround is to provide a standard R server, perhaps with docker or some more widely supported environment so that all the required versions are available. An alternative is to provide an R mirror with only the release-time versions to support installation.

A further complication is that R depends on system libraries. New versions of packages, or of R, may include dependencies on additional library for whihc the development versions must be instaklled. These are routinely added to the env/Makefile targets for each operating system for new users.

To keep the R installation updated, daily tests would be ideal to catch new dependencies, ideally for each supported operrating system as some will have libraries loaded by default.

In-house extensions

In the pharmaceutical industry the practice has been to develop their own extensions to tranSMART. At Sanofi the Browse tab study metadata and a series of new data types were developed, but they continued to maintain their own transmart code including support for additional large files in MongoDB, an ETL package (ICE) and alternative authentication mechanisms. At Pfizer extensions to handle GWAS results were added. In both cases the native code only supported Oracle and needed to be ported to Postgres. Some ETL issues remain to be resolved. The Hyve developed new analysis methods and support for aCGH data for the TraIT projects in the Netherlands but Oracle support for these extensions has gaps (e.g. data loading).

Database utilities

TranSMART 19.1 adds some new utility functions to support local updates to tables.

Examples include adding a new species to the biomart.bio_convcept_code table for use in the Browse tab. The concept must be added to four tables (also biomart.bio_data_uid and thee two search_keyword tables).

These are initially for Postgres but will be added to all supported databases.

Additional routines can be added to allow local updates of other metadata values.

The transmart-etl loader should also be updated. It has been unchanged for the last few releases. A useful update would be to amend gene synonyms by removing obsolete names and adding new ones where a gene already exists. The old names should be removed to cater for their reuse for otehr genes.

Analysis with High Dimensional Data

TranSMART 1.2 onwards pops up a High Dimensional Data panel which prompts for gene lists, gene signatires, proteins, etc. but only displays the other parameters (platform, tissue, sample type) in the selected high-dimensional data node(s). Timepoint shoould be included but it is not displayed.

Earlier tranSMART versions up to release 1.1 displayed a similar popup but with va;ues for each of these, and for each of the two subsets, that could be specified by the user. This allowed two identical subsets to be compared using different subsets of the high-dimensional data. Even with expression data split by tissue, sample type and timepoint then dropped together into the data box both subsets currently have to work with the same set of high-dimensional data.

If there is interest in restoring the earlier capabilities this will require only modest changes. Should we also add warnings about comparing different samples in the varuious analysis methods?

Time series data

There is limited support for time series in tranSMART. Customisations by Sanofi introduced in 16.2 recognize metadata for series labelled by hour, day, week, month, year and can produce a time scale on, for example, a LineGraph in the Advanced Workflows.

This feature relied mostly on capabilities added through the tMDataLoader ETL tool to define time units in additional columns.

These capabilities could be made more generally available for other ETL tools.

TranSMART Pro

The TranSMART Pro project (2016-17) was funded by 4 pharamaceutical Industry partners to address a set of Use Cases involving longitudinal data, improvements to clinical data and workflows for genomics. The code was developed and released as "tranSMART 17 server-only" with the capability to query longitudinal data through a new Application Programming Interface (REST API v2).

There were some significant changes to the tranSMART schema to incorporate visits.

New data types

At the Paris 2017 i2b2/tranSMART European users meeting a group gave a talk describing their integration of flow cytometry data into tranSMART. This could be imported into tranSMART 20.

Other users have imported flow cytometry as clinical data. We need to review the alternatives, describe what can be dome using clinical data, and implement a new data type with additional benefits.

Users in Oxford, England have proteomics data for protein clusters, requiring adjustment to the way identifiers are used so they can use the cluster or protein ids in queries and visualise them in results

I2b2 integration

Supporting the i2b2 schema is only part of the way to full integration. I2b2 loads all clinical data into a single ontology, while tranSMART loads data by study. More work is needed to establish how to allow both platforms to work with a common set of patient data (for clinical trial cohort selection in i2b2) and study data (for clinical trial analysis in tranSMART).

The schema for transmart 19 has been matched to i2b2. A few minor changes can be added post-release (reducing the size of some columns to the i2b2 value unless there is known transmart data that needs longer text, extending ids to 8-byte integers where very large data volumes may be loaded - and especially if they may be reloaded.

Certain tables are key to the i2b2 model. These should be made clean so either they are identical or i2b2 has agreed to ignore any additional transmart columns.

Other tables may be used by both platforms. These must each be documented and regularly maintained at release time.

We need to review how queries are processed and stored to check whether there could be a clash is both platforms are active on a common database.We need to check ownership and access rights within the data is a user logs in to transmart or to i2b2. They should be consistent at least with the access a guest user would have, and should not grant any additional rights that are otherwise limited in one platform.

Note: because transmart started out as i2b2 the i2b2 tables tend to be included in transmart. We need to review the code to track any references to these tables and figure out what functionality is involved, and where records may be created, modified or deleted by transmart. Transmart 19 includes all the i2b2 tables, but as only i2b2metadata and 'i2b2demodata' were used by tranSMART 16 we know already we can ignore the other 4 i2b2 schemas. We also know i2b2 will ignore the transmart schemas so only i2b2metadata and i2b2demodata need attention.

In i2b2 other schemas can be created equivalent to i2b2demodata. We need a mechanism for these to be usable by transmart, following something like the route i2b2 uses to find them.

Another key distinction is i2b2's support for SQLserver. transmart started out as Oracle only. The open source projects added postgres but nobody has addressed any SQLserver issues. Creating the database is relatively simple - it is very close to the Postgres code. Issues will arise in the remaining SQL code and in translating and testing ETL stored procedures/functions, but as these are repetitive the task is lighter than it appears at first.

Common i2b2/tranSMART model for clinical data

While both platforms load and query clinical data, there are some key differences in their current approaches.

TranSMART loads data by study. All concept paths begin with a top node down to the highest node of the study. All other concepts have paths which are extensions of this highest level.

I2b2 loads all clinical data under a single common tree. Taking Age as an example, there is a single Age concept in i2b2 under Subject. In tranSMART there is a seperate Age concept path for each study.

A further complication is that tranSMART uses additional sample-based data linked to the high dimensional data (gene expressoin etc.). These appear as concepts but do not fit any current i2b2 ontology. There are some plans to incorporate sample ata into i2b2, but these are at an early stage,.

We should aim to incorporate common concepts such as Age, Sex, Ethnicity, Race... in tranSMART by allowing them to be loaded as common concepts with observation facts linked directly to the study they originate from. Thsi was partly achieved in the TranSMART-Pro project. We can hope to make some progress using changes to ETL procedures. We can expect changes will be needed in the ontology tree display and in generating and executing queries.

We should also aim to address studies with patients in common. There are examples where a cohort of patients is used across two or more studies. Currently we can try to load under 'Across Trials' with special code in a number of places to resolve the alternative ways to handle the data. The key problem is the current restriction to query on one study only which needs to be relaxed to allow queries on multiple studies, or at least to retrieve matching patient data from other studies.

The i2b2 team are reviewing the ways in which they handle visits and encounters. We can aim for a common approach so that tranSMART can try to store data in a way that will make sense to users when viewed from i2b2. Time data for tranSMART should be stored in ways that are in common with i2b2 where possible, This is complicated for tranSMARt by studies which record time intervals but not the actual dates . In these cases, one option would be to use the study publication date as a baseline for time interval calculations.

Similarly, data loaded in i2b2 should be usable from within tranSMART. An obvious apporach is to assume all i2b2 data is part of a common study which we could call 'i2b2'. We should also pay attention to the i2b2 'projects' and treat each project as a 'study' in tranSMART.

Analysis

Fractalis

TranSMART 19 added code for the new interactive analysis plugin "Fractalis." This project is a successor to SmartR in tranSMART 16.2, which in turn supersedes the Advanced Workflows in tranSMART 16.1 and earlier.

Implementation of Fractalis depends on the development of an Application Programming Interface between Fractalis and tranSMART to export data for analysis. Available prototypes cover only the modified schema developed in the tranSMART Pro project. A fresh development is needed for tranSMART.

Coverage of the available analyses is incomplete. There are Advanced Workflows that are not in SmartR (though part-tested code is available to provide them) and similarly workflows that are not
provided by Fractalis.

SmartR

The eTRIKS project developed implementations for the remaining Advanced Workflows as SmartR workflows. These are available for download. They could be implemented and compared to the Advanced Workflow versions to validate the results.

Advanced Workflows

The Advanced Workflow capabilities should be cleaned up. Parameters were originally intended to be user-editable though the scripts are hard-coded in the server. The configurations shoukld be removed from database tables (under searchapp) and stored in the application.

External Analysis

Third-party approaches to data export and analysis in various systems should be brought into tranSMART with full suport where they can be of general use. This would allow comprehensive tested for each release of either tranSMART or the third-party system.

Automated testing

Testing for transmart 16.3 used Geb and Selenium with a firefox browser driver. We have suggestions for a UI testing framework that may be helpful - it was used with Glowing Bear.
We should also update the current tests to Geb 3.4 (released April 2020) and check them against the updated tranSMART19 codebase and schema.
We need to first set out a detailed list of what needs to be tested, and prerequisites (mainly studies and metadata required).

We need to implement tests in all areas, covering the major features so they can be routinely examined.

We need a test database that can be used for tests that do not need to update the database significantly (so that several test sites can be running against the same database)

Curated datasets

Testing datasets with more than one species. For example, cells from tissues with viral infection, blood cells with parasite infection.

Possible improvements include adding a species prompt before selecting genes so only genes for the species of interest are displayed. This can be tested with human, mouse and rat genes for selected studies.

Extensions to ETL

(Some of these are also being considered for early adopion in tranSMART 19)

The TraIT Cell-Line Use Case project added the capability to load entire studies with platform annotations and high-dimensional data in a single step. This could be extended to the current set of 200+ curated studies providing a simpler solution that does not require checking for multiple platforms and multuple high-dimensional datatypes as used in a few of these (mainly GEO) studies.

ETL targets could be provided to load Browse tab metadata. The appropriate text and associated terms could be generated for each curated GEO study and automatically loaded into the Browse tab for text and faceted searches. Careful curation is needed for misspellings in the GEO text and to trim the total length of descriptions to 2000 characters. Some GEO studies exceeed this length.

The Browse features were developed by Sanofi and include the ability to create folders and store large data and text files. These typically included the GEO series matrix file at hte time of data loading (these files are occasionally updated by the authors). There is also a little-used MongoDB capability in the Browse tab to search large data files. This could be tested and updated for general use.

We could also ctreate test scripts to validate that data is loaded correctly. The tMDataLoader already supports this for clinical data, but a more general; approach would be very useful. If studies were documented with the age and gender profiles and other useful statistics these could be checked against the results of data loading, along with expression levels for genes and other high-dimensional datatypes.

The sample explorer tab could also be populated. For many studies there is only a limited set of values available. The sample explorer is more useful for local clinical samples.

Internationalization

There is a capability to internationalize messages and text by defining a language-specific file with the English versions.

Matching files can then be generated for other languages, with the most important features prioritized.

These message files are used by grails. We also need files to be used by javascriptWe may also need translations of HTML files as a cleaner implementation than changing/generating all the text.

We can make a start by trying to define the English text and adding an alternate language with the
text reversed so we can understand what is happening.

Local customization within the UI

We have several requests to change the terminology in the UI which we avoid because it would differ from the standard i2b2 view. However, it should be reasonable to allow sites to customize these, subject to a warning that the documentation will differ from their instance.

Examples include:

- Display “Navigate Terms” as “Navigate Concepts”
- Display “Sex” as “Gender”

We can add configuration variables for common terms - including others on the same pages for completeness - and check the alternate version is used by defining e.g. “AltSex” and making sure this replaces all occurrences.

Reducing autocomplete lists

Lists of genes can be complicated where too many similar results can be displayed. For example, the genes, pathways, etc lists for the high dimensional data selection. Especially genes as human and mouse are distinguished only by upper/lower case and other species may be added.

We should be able to display the values with some prefix to make it clear what is being selected, then use the underlying value within the application.

Alternatively we could add some way to limit the results to applicable terms e.g. by species.

Authentication

We support severel authentication methods in tranSMART. Some are also supported by (or planned for future releases) in i2b2. Howvere, the two platforms use different approaches to maintaining authorization data in the database. TranSMART stores user and role data in the searchapp schema. I2b2 stores in the PM cell.. Now that tranSMART also has the i2b2_pm schema for completreness we could update our code to store ruser and role information in common with i2b2 and to follow i2b2 rules for access to tables and data records. This could be made configurable so that transmart can continue with the existing system for users who do not wish to change.

Standard test authentication providers are needed for each method for both the current release and future releases. These could be shared with i2b2.

As a first step we need to document all the existing authentication methods - how to configure them, how to manage users and roles, and how to test them on a new installation. Testing needs to not only include authentication but also a range of error coditions to ensure logins and appropriate diagnostics are presented to users and administrators.