tranSMART 19.0 Release (May 2020)

Release 19.0 (May 2020) is a **major update** to the previous release 16.3.

Release Notes tranSMART 19.0

Version Number

This version is renumbered to reflect the year of release, and to indicate the major effort in rewriting and reorganizing code.

This is also the first official tranSMART release to be compatble with the parallel i2b2-transmart project. The intention is for i2b2-transmart to use this release, perhaps with a small number of additions to integrate with their latest changes in other codebases.

Installation instructions for tranSMART 19.0 are in preparation at Install the tranSMART 19.0 release

Test Version

A test version of the full release is available at http://postgres-test.transmartfoundation.org/transmart

Details of the beta test server data and of the features to be tested are in the beta test public instance wiki page

Single repository

TranSMART 19.0 code is reorganized in a single repository https://github.com/tranSMART-Foundation/transmart

The top level directories are merged copies of the tranSMART 16.3 repositories with minor changes.

The combined repository makes release branches simple to manage. A single branch in the main repository can be used to generate all the artifacts for a release dostribution.

The directories mirror the original organization of the source code for the tranSMART 17 project to simplify direct comparisons,

transmart-core-db and transmart-core-db-tests

The test code previously under transmart-core-db/transmart-core-db-tests has been relocated to its own top-level directory.

This makes building and testing simpler, and was also the organization chosen by the transmart 17 server-only project.

The two RNAseq datatypes are better separated in the core code.

transmart-etl

The release 16.3 tranSMART-ETL repository has been renamed to all lower case. Unused legacy directories have been purged from the repository, greatly reducing the size of the zip file generated for each release.

- FCL4tranSMART is superseded by transmart-ICE which has the latest released code from Sanofi for their ICE tool data loader (formerly FCL4tranSMART).
- The V1.2_Hackathon directory contents were unused
- The loader is now superseded by the copy built from the src directory
- The src-old directory is clearly redundant
- The duplicate Kettle scripts under Kettle-GPL, Oracle and Postgres are removed. Since the start of the release 16 series the release copy of these scripts has been in the database-specific directories under the Kettle directory. These are used by the make targets in transmart-data and should be used by any local ETL pipelines.

transmart-extensions

The transmart-extensions plugin in release 16.3 has been split into its three components:

- transmart-java
- biomart-domain
- search-domain

This also reflects the rearrangement in the tranSMART 17 project.

Galaxy-export

The old release 16.x 'blend4-plugin' is renamed galaxy-export-plugin.

Throughout the code the name 'blend4j' has been replaced to make the functionality clear.

SmartR

SmartR was developed by the eTRIKS project and released in tranSMART 16.2 with a set ogf interactive analysis workflows that supersede many of the functions of the "Advanced Workflows" tab.

We are testing new SmartR workflows developed by other partners in the eTRIKS project to provide the remainder of the "Advanced Workflows" functionality.

The Advanced Workflows remain active in this release. We anticipate that users will require them in order to reproduce previous analysis, and they can be used to compare results and encourage migration to SmartR.

Fractalis

Fractalis was developed for i2b2-transmart by the same author as SmartR (Sasha Herzinger at the University of Luxembourg) and superseded several of the SmartR workflows.

We are working on the full integration of Fractalis into tranSMART 19.0. This is a work in progress, involving new ETL interfaces between ttranSMART and Fractalis.

TransmartApp Web Application

Help pages

The revised transmart manual is added as help pages inder the defautl URL /transmartmanual. Links from the web interface bring up the appropriate section in a new tab.

Additional configuration parameters were added by the old transmartPro project to link to external help pages. These remain available ifor sites that have extended tranSMART (for example by adding their own local advanced workflows) so that they can be linked to local help pages.

Help links have been added to pages where they were missin gin previous releases, including the Comparison, Summary, and GridView tabs under Analyze.

Gene Signatures

The Gene Signature tab provides a way to maintain lists of genes, SNPs or probeIDs to define high-dimensional analyses (heatmaps etc.). These have been updated and tested. A new stored procedure is added to load platforms to match gene lists (the platform must be in the right table, with the species defined, for the gene signature to pick up the required genes.

Gene Signatures have rich metadata to docuyment their origins, but these depend heavily on the avalable concepts and metadata in the installed database.

Script are in preparation to add further metadata concepts.

If there is interest, we could automate the addition of missing platforms when they are first added to load new studies.

Gene siugnatures can be made public to make them visible and usabel by any user.

A public set of gene signatures can be a helpful addition to a tranSMART installation where a set of markers is of special local interest.

Gene Lists

Gene Lists are a simple version of the Gene Signatures.

Validation tests have been improved to mark any gene, SNP or probe that is not found in the database. It remains up to the user to check the markup before saving as the interface only saves these lists on successful validation.

The safest way is to load from a file of gene names or IDs as this is easier to edit and reload.

Administration

A se of extra pages are available to users logged in as administratotors via the Admin tab

Build Information

Lists the metadata for the tranSMART web application.

Status of Support Connections

Tests the Solr server is up and reports the number of items under each category.

Configuration Information

The page was added in 16.3 and is extended in 19.0. All configuration values are retrieved, categorized and reported.

Known parameters are documented with a description and a report of any assumed default value.

All other known values are added to the appropriate tables.

Any additional values are appended to an 'Unknown' table at the bottom off the page. These may be temprary variables from the Config.groovy script or possible errors in parameter names to be corrected.

New in 19.0 is a set of "Manifest" settings. These are the links to all the javascript, stylesheet, image and otehr files packaged in the transmartApp distribution. By default this table is closed so you only see the total count. Other tabls start open but can be closed to make the remaining tables more readable.

Customization

Some customization options have been added to the database in 19.0.

These include storing image files and icons so they can be updqated without restarting.

Further customization options are planned for future releases. Please contact the developers if you have suggestions for features you would like to see.

JavaScript cleanup

Multiple versions of jQuery have been replaced by a single version across all plugins.

Definitions of JavaScript elements used obsolete attributes. These have been updated to the attributes supported by, for example, ExtJS 2 (the last open source release of that library).

Browser console warnings have been fixed for FireFox.

Timing issues in the Analyze tab have been addressed. One that remains is the failure to display the current Query on the Advanced Workflow tab. Visiting another tab and returning to the Advanced Workflows ensures the tab is fully initialized and the query is then visible.

Location and timing of the definition for drag-and-drop in Analyze tab sub-pages has been updated.

R and tranSMART

The Rserve service is updated to provide better control for installations installing R in transmart-data.

The template can also be used for load R installations.

Whichever is used, the Admin page "Check support connections" page will test all the required R packages are available.

The R installation requires the latest Rserve from Rforge.Net as the current version in R has a fatal datatype error with some workflows. This has been an issue in R for at least two years.

The Rserve service template writes to a logfile to debugging output from R and error messages can be traced more easily when anlaysis has issues.

SolR Server

A template is provided to install a service to launch solr.

Apache solr is used to index and search the Browse tab metadata, the navigation tree and the SampleExplorer.

The solr server writes to a logfile to help tacing issues.

The solr server also provides help through its administration interface.

Database schemas

tranSMART and i2b2

The database schema has been updated to resolve, as far as possible, differences with the i2b2 1.7.12 schemas.

Where columns are date or time values in i2b2 and string values in transmart, they have been corrected to the appropriate date or time values. Initial testing found no conflicts in ETL procedures.

Column widths have been defined to be the same size across multiple tables (e.g. subject_id). Local installations are free to increase these sizes if they require longer strings but should beware of potential impact on database performance.

Required columns are updated in transmart so that any columns required by i2b2 in a common table are also required in transmart. One date needed to be defined in ETL procedures using an obvious default value. No other impacts have been noted in initial testing.

Triggers are required for some tables in i2b2. Although postgres can be configured automatically to increment unique identifiers for new rows in a table, the i2b2 code may include a call to increment a named sequence to generate a unique id value when a row is inserted. This necessitates defining a sequence with this name and using the same sequence in a trigger function to maintain database integrity.

Integers in postgres are defined as type 'int' unless extremely large values are expected. Values that can exceed 1 billion are defined as type 'bigint'.

New schemas

Four additional i2b2 schemas have been included:

- i2b2hive
- i2b2imdata
- i2b2pm
- i2b2workdata

Though not used (currently) by tranSMART the aim is to create a database that can be populated and used as an i2b2 platform.

New I2b2 Tablespaces

I2b2 does not define tablespaces by default. A set of tablespace names were agreed with the i2b2-transmart developers for implementation in the tranSMART schema.

All i2b2 tables are in tablespace I2B2 and all i2b2 indices are in tablespace I2B2_INDEX.

These reflect the tablespaces TRANSMART and INDX used for the transmart-specific tables and indexes.

Redundant tranSMART tablespaces

Earlier tranSMART releases defined 3 additional tablespaces: biomart, deapp and search_app. These were no longer used - though they were created with a new database.

They have been removed from tranSMART 19.0. For this release they are ignored by tranSMART code

New tables have been added to the schemas used by tranSMART 16.3.

Support for transmart_batch on Oracle is included by default. In tranSMART 16.3 this had to be installed before transmart_batch could be used with an Oracle database.

We are working on including support for tMDataLoader on Oracle and Postgres by default. This involves running the respective installation scripts and incorpporating the changes into the Postgres and Oracle schema definitions, including the tMDataLoader specific versions of ETL functions.

Default passwords for database roles

The usernames and passwords for the database roles are defined for Kettle when the database is created. By default the username and password are the same as the schema. While this is simple for developers, production instances should define unique passwords for each role.

By adding these as environment variables when launching the database creation target the Kettle properties files will be populated. We recommend retaining the role names as these are used in many places in the code. The kettle properties files and the transmart-data/vars file shjould be secured from read access by other users.

Default passwords for transmart login

A limited set of usernames is defined for this release, as for previous releases. These have a default password, usually 'transmart2016' except for username 'admin' with default password 'admin'. These are simple for developers.

Production instances should change the passwords through the transmartApp web interface using the 'Admin' tab to manage the available user accounts.

Notwithstanding this, there is also a configuration option (turned off by default) to enable a guest login that by default will allow a visitor to a transmart server to login as username 'guest'. This is useful for public servers to provide access without setting up a user-specific login.

If using this option, changing to another username (to become an 'admin' user) requires logout through the 'Utilities' menu, or explicitly using the login page URL.

Initial database content

A new database (for Postgres or Oracle) is defined as a target in transmart-data.

A common set of data is loaded for both databases. This includes a set of standard ontologies (or data dictionaries). In this release these are:

Data	Source	Date	Table
Human genes	NCBI		biomart.bio_marker

Mouse genes	NCBI		biomart.bio_marker
Diseases	Medline		biomart.bio_disease
Countries		Selected	biomart.bio_concept_code
Therapeutic Areas		Selected	biomart.bio_concept_code
Organisms	NCBI Taxonomy	Selected	biomart.concept_code biomart.taxonomy
Platforms	GEO etc.	Selected	biomart.assay_platform
Proteins	SwissProt/UniProt		biomart.
MicroRNAs	MiRBase		biomart.

The data from these sources is included in the *searchapp.search_keywords* table with synonyms in the *searchapp.search_keyword_terms* table.

Additional data dictionaries

Further data dictionaries can be loaded, and the data for the abvove dictionaries can be updated, using the loader utility under transmart-etl

Data	Source	Date	Table
Human Pathways	HMDB		biomart.
Pathways	KEGG	Last public release	biomart.

Grails upgrade

Grails version 2.5.4

This release is built using Grails 2.5.4.

Grals can be installed using:

sdk install grails 2.5.4

Earlier releases up to 16.3 were built using Grails 2.3.11 and Grails 2.3.7

Building individual components

Each directory has a build script.

Grails components use the *./grailsw* script with the targets package-plugin (or packagePlugin) and maven-install (or mavenInstall)

Maven builds use the *./gradlew* script with targets clean, build and publishToMavenLocal

In two comnponent where these scripts are not yet built, the build used maven directly.

For two external packages mydas and IpaApi, cd to the directory and build with:

mvn install

Component dependencies for transmartApp

The recommended build order to ensure dependencies are satisfied by previously built components is:

Directory	Name	Dependencies	Description
transmart-core-api	transmart-core-api	-	Core API
transmart-shared	transmart-shared	-	**New in 19.0.** Provides generic utilities to return security information about the current user to remove the need to pass the username around in service calls.
transmart-legacy-db	transmart-legacy-db	-	
transmart-fractalis	transmart-fractalis	-	**New in 19.0.** Adds the Fractalis interactive workflow tab, integration is a work in progress.

mydas	mydas	-	The original DAS code imported into tranSMART because there is no guarantee the original distribution will remain available. A small section of code is intended to be autogenerated, but this is easy to maintain by hand if any of the dependencies change.
dalliance	dalliance	-	The dalliance genome browser.
transmart-core-db	**transmart-core**	transmart-core-api transmart-shared	
transmart-core-db-tests	transmart-core-db-tests	transmart-core-api transmart-core	**Moved to a new directory for 19.0.** Tests for the transmart-core-db code, also used by SmartR and transmartApp.
transmart-mydas	transmart-mydas	transmart-core-api mydas	
transmart-java	transmart-java	-	**Moved to a new directory for 19.0.**
biomart-domain	biomart-domain	transmart-java	**Moved to a new directory for 19.0.** Domain definitions for the large number of tables in the biomart schema.
search-domain	search-domain	transmart-shared biomart-domain	**Moved to a new directory for 19.0.** Domain defintions for tables in the searchapp schema. This covers keyword searching and user, role and access management as both are defined ni the searchapp schema.
transmart-rest-api	transmart-rest-api	transmart-core-api transmart-shared transmart-core transmart-core-db-tests	
transmart-custom	transmart-custom	transmart-shared search-domain	**New in 19.0.** Provides services to customize aspects of tre user interface using new application tables. Documentation is needed for these new capabilities.
folder-management-plugin	**folder-management**	transmart-core search-domain	
Rmodules	**rdc-rmodules**	transmart-core-api transmart-metacore-plugin transmart-shared	Provides all the Advanced Workflows. There is also a dependency on data loaded into the searchapp schema to define the inputs,, outputs, parameters and scripts for each workflow and to defined their names and the order in wuich they are presented. This was intended in the pre-open source versions of tranSMART to allow users/administrators to edit these settings, but this makes little sense to control fixed scripts distributed in the transmart.war file. There is no known instance of a site developing their own Advanced Workflows thourh these mechanisms.
spring-security-auth0	spring-security-auth0	transmart-shared transmart-core search-domain transmart-custom	**New in 19.0.** A new Auth0 controller and services. Documentation is needed for these new capabilities.
IpaApi	IpaApi	-	This code provides one third-party SmartR workflow to interface to Ingenuity Pathway Analysis. SmartR includes hooks to load the IpaApi workflow
SmartR	**smart-r**	transmart-core-api transmart-core transmart-core-db-tests IpaApi	**Several potential new workflows from the eTRIKS project are candidates for inclusion.**
galaxy-export-plugin	galaxy-export-plugin	transmart-shared transmart-legacy-db rdc-rmodules	**Directory renamed from blend4j-plugin for 19.0.** A version of data export that transfers to an instance of Galaxy for further analysis. The user needs credentials to use the galaxy instance, defined in the server Config.groovy file.
transmart-metacore-plugin	transmart-metacore-plugin	transmart-shared	
transmart-xnat-viewer	**xnat-viewer**	transmart-core-api search-domain	
transmart-xnat-importer-plugin	**transmart-xnat-importer**	transmart-shared biomart-domain	

transmart-gwas-plugin	**transmart-gwas**	transmart-shared transmart-legacy-db transmart-core search-domain rdc-rmodules folder-management	
transmart-gwas-plink	transmart-gwas-plink	rdc-rmodules	
transmartApp	**transmart.war**	transmart-fractalis dalliance transmart-core-db-tests (test) transmart-mydas transmart-rest-api spring-security-auth0 smart-r galaxy-export-plugin transmart-metacore-plugin xnat-viewer transmart-xnat-importer transmart-gwas transmart-gwas-plink	Ths is the full transmart server, providingall the functions of the User Interface plus the access methods for the RESTful API to generate ans serve authentication tokens and to serve results when these topkens are presented.

Java version

Grails 2.5.4 uses Java 8.

Development is using the openjdk java 8. We will confirm later the suitability of oracle java 8 which is onlt available from third-party sourecs for the Ubuntu 18 test systems.

There is no longer a need for a legacy Java 7 install to work on tranSMART development.

The pivot utility in the Kettle ETLs has been recompiled with Java 8.

Asset Pipeline

Javascript, stylesheets, and other resources are packaged and provided through the asset-pipeline plugin.

This invoilves a major reorganization of the source files and changes to hardcoded file paths.

This upgrade is a major step towards preparing for an upgrade to Grails 3 or the newly released Grails 4 in a future tranSMART release

Code review

A major code review was conducted by Burt Beckwith at Harvard as part of the inclusion of tranSMART 16.1 and 16.2 code in the i2b2-transmart project.

The planned changes were described in the i2b2-transmart roadmap and summarized below

Summary of tranSMART 16-2 changes in i2b2/tranSMART 18.1-beta

optional support for Auth0 authentication

A new directory spring-security-auth0 provides Auth0 services.

Groovy code formatting and consistency

Coding standards have been applied to groovy code:

- leading white space
- code indentation
- positioning of braces:
    - if/else blocks: newline after {
    - } on new line
- splitting of long lines
- Map and List values on single line is possible
- replace -each for a Map or List by a loop with a datatype and named variable specified for the value
- Use boolean for all true/false variables

- Use int for integer variables
- Use single quotes for string values except to avoid escaping many single quotes for readability
- Use {} to insert values into strings

updated logging to use Slf4j wrapper and parameterized logging for performance and to help with Grails 3 Logback migration

All source files that used Log4j and calls to log.info:

- import groovy.util.logging.Slf4
- define Slf4j as 'logger'
- call logger.info (etc.) to write to log output

converted most Config lookups to use Spring's @Value annotation

Replace configuration parameter references with @Value definitions using with org.springframework.beans.factory.annotation.Value

use "private @Autowired" for dependency injection to reduce API pollution

Add @Autowired references with org.springframework.beans.factory.annotation.Autowired

replaced many uses of 'def' with actual types, particularly in method signatures

Throughout the code of release 16.3 types were undefined. Adding explicit types wherever possible provides validation of the type sactually passed and improves the usefulness of error messages when code breaks.

converted many cases of copy/paste to use methods

A single method can replace a set of identical code segments making testing and maintenance far more robust.

domain class cleanup, removed many unnecessary declarations

Domain classes have been reviewed and matched to the updates database schemas.

simplify access to current username, user id, roles, etc. via new SecurityService

The new SecurityService in transmart-shared provides calls to return information on users and roles for implementing access polcies.

removed passing AuthUser to methods that always work on the current authenticated user, moving the lookup to where it's used

A new directory transmart-shared provides utility functions. These inclide generic checks for the capabilities of the authenticated user, allowing the username to be removed from many method calls where it was being passed down.

upgraded to Grails 2.5.4, Servlet API 3, Java 8

Release 19.0 is built using grails 2.5.4 which depends only on java 8.

tests cleanup, converted to Spock

Tests updates in transart-core-db-tests

One test currently fails. It is testing something that is supposed to fail, but should be trapping the error condition and reposting a test success.

converted controller closures to methods

Closures are defined as methods with a set of parameters, replacing the closures and parameter fetching in earlier releases.

Much of the code remains unchanged within the method aside for parameter handling and other coding standard changes (see above)

controller params simplifications

Parameters are explicitly defined in each method

This give cleaner code where it is obvious what parameters are used and what parameters are available to control the result.

some removal of logic in GSPs, moving to controller/services (much more needed)

Groovy Servlet Pages (.gsp files) cleaned up to avoid interpreted code critical to functionality

Standard indenting of HTML within GSP pages.

updated sql queries to include schema+table instead of adding many db synonyms

Especially in Oracle code, tranSMART 16.3 defined synonyms to allow reference to tables without the schema.

Explicit schema references are cleaner.

They also make it possible to derive the permissions needed for functions/procedures to operate across schemas.

removed many unused methods, some unused classes, org.json source classes, duplicate classes (e.g. many in both transmartApp and transmart-java)

Many ususued methods were retaine dbacause it is difficult to be certain they wil never be invoked.

The level of testing undergone by transmart 19.0 makes this an ideal time to remove these methods and check they were indeed redundant.

However, some removed methods in the i2b2-transmart code were unused on Oracle but were required when running on postres and have been reinstated. Examples include handling large objects as strings.

deleted deprecated AccessLog class and changed to use AccessLogEntry and AccessLogService

Simplifying the access logging code

added simple Spring Security role hierarchy

Thsio requiers furthe rtesting to make sure the required tranSMART functionality is supported.

deleted many lib directory jars and replaced with BuildConfig dependencies

These jar files are removed from the code repository. They are downloaded though code dependencies in BuildConfig.grooxy or pom.xml.

These jar files remain in the repository. They may be removed later.

where possible annotated classes and methods with @CompileStatic for performance

Many classes amd methods now have @CompileStatic. Testing found very few instances where the annotation had to be removed.

StringBuffer -> StringBuilder

StringBuild is used to carete a string and to append to it using '<<' syntax

split transmart-extensions into three projects, split out transmart-core-db-tests

See the organization of the new single transmart repository

moved filters classes from grails-app/conf to grails-app/filters

Filters are now in grails-app/filters/org/transmart...

some SQL injection fixes

All SQL statements in the code need careful testing to check they work for both Oracle and Postgres

some conversion of simple Java classes to CompileStatic Groovy

Testing is needed to ensure that code functions as in earlier releases.

Oracle support

Oracle is fully supported using the same release (12.1 or 12.2) as previous versions of tranSMART.

Testing relies on an Oracle Docker instance.

Postgres support

This release has been tested on Postgres up to 9.6, and on Postgres 10, Postgres 11 and Postgres 12. No version-specific issues have been identified.

To date no attempt has been made to take advantage of the new partitioning features in more recent Postgres versions. We continue to monitor these and will consider supporting them in a future release. It is likely that legacy support for the current Postgres schema will be continued in tranSMART.

SQLserver support (potential)

TranSMART does not support SQL server.

Upgrading the schemas to include SQLserver versions of the tables and stored procedures is relatively straightforward.

Upgrading the source code to include support for a third database would require significant work, but would also test and clean many sections of code with obvious benefits to the quality and robustness of tranSMART.

Ubuntu 18 support

Changes have been made to support installation on Ubuntu 18.04, Ubuntu 16.04 and Ubuntu 14.04.

Automated install scripts have versions for each version with only limited divergence. For example, Ubuntu 18.04 uses tomcat 8.

Targets for Ubuntu installation targets in transmart-data are updated as appropriate (for example, a different PHP version is available in Ubuntu 16).

Scripts for Ubuntu 18.04 are updated with system libraries installed to cover dependencies for installation of R packages.

Ubuntu 20 support

TranSMART 19 is being tested on Ubuntu 20 (released in Spring 2020).

Fedora support

TranSMART 19 is being tested on Fedora 32 (released end-April 2020). Code has been built and tested on Fedora 31.

ETL with transmart-data load targets

Kettle 8 support

Releases up to 16.3 were tested only with Kettle 4.2.

tranSMART now supports Kettle versions up to pdi-ce-8.2.0.0. We will test Kettle 9.0 as part of the final release process.

Only minor updates to Kettle scripts were needed to satisfy an additional validation. These had prevented upgrading Kettle in earlier tranSMART releases.

New ETL targets

New targets are in preparation.

Study

Changes introduced into tranSMART 16 supported loading clinical and all high dimensional data in one step through a series of scripts and new parameter file for the TraIT Cell-Line Use Case poroject.

These scripts can be added to a structured set of directories to support the loading of all data for a study in a single step.

Should there be a failure at any point, going to the appropriate directory and running the load script there will resume just that part of the load after the issue is resolved.

Browse Tab Program Metadata

A set of utility scripts are mad einto a load target to create a new Program under the Browse tab.

The disease and therapeutic area fields are validated on loading.

Browse Tab Study Metadata

A set of utility scripts to load study metadata for the Browse tab can now be involked as load_browse targets.

The input files can be created for studies in the existing curated data library.

The input data includes the text from GEO (reduced to 2000 characters), disease and therapeutic area, number of patients, citation details, study type and objectives, etc.

The program must be loaded before the study

Browse Tab Assay Metadata

A potential load target can add Assay metadata into the Browse tab using scripts in preparation.

The platform information should be validated against the database ontologies before loading.

Sample Explorer Tab

A set of utility scripts are in prearation to load sample data into the Sample Explorer tab.

The input files can be made available for studies in the existing curated data library.

In GEO samples have limited informations but at least includes sample ID and organism.

Coding standards

Cleanup of SQL source code

- Standard indentation of lines and within SQL statements
- Consistent use of upper and lower case
- Commas in lists of columns/values positioned to ensure the correct row is indicated in SQL error messages
- Changed 'select ... into rtncd' to 'perform' where the rtncd value was ignored.
- Datatypes cleaned up e.g. int v. bigint
- Inputs and output matched to closest equivalents in Kettle
- Added 'explain' blocks for postgres to improve performance of slowest steps

ETL Data Loading

General ETL performance

Loading raw high-dimensional data in earlier versions could take a very long time. A SQL statement testing whether log intensity could be calculated for each raw intensity was creating very large loads on the server.

Preprocessing the raw intensities to identify usable values allowed this step to be simplified. Log intensity values are now calculated on a simple pass through the data using very low resources.

RNAseq ETL performance

A missing condition in a SQL statement caused RNAseq gene expression to load extremely slowly, and to consume vary large memory and tmp space resources. No other datatypes were affected.

High-Dimensional Data Columns

Previous releases loaded high-dimensional data (Microarray mRNA expression, RNAseq counts, etc.) with columns labelled as TISSUE_TYPE, SAMPLE_TYPE amd TIMEPOINT.

The stored procedures all reversed the meaning of the first two columns internally. This is corrected in release 19.0.

As recent released removed the ability to select on these columns when launching analyses the issue shas not been noticed.

A future release may restore the picking of sample and tussue types, and of timepoints when launching heatmaps and other workflows.

The libraries of curated studies at library.transmartfoundation.org will be reviewed to ensure these terms are in the correct column.

Consistent usage will be applied to tissue types, timepoints, and the sample treatments across these 200+ studies.

Clinical Data Ontologies

The libraries of curated studies at library.transmartfoundation.org

have a variety of representations for common terms in the clinical data tree.

These studies will be reviewed to conform to a common set of terms to make it easier to work with multiple studies in tranSMART. Terms in common use (e.g. 'Medical History') should appear in the same place for each study.

Kettle ETL debugging

Previous releases have been hard to debug when loading data using Kettle. A number of issues are addressed in tranSMART 19:

Kettle logging level can be set with an environment variable KETTLE_LOG_LEVEL with possible values

- BASIC (default)
- DETAILED
- ROWLEVEL
- ERROR
- NOTHING

The value is passed to Kettle as the -level parameter

ETL Stored Procedure debugging

When ETL procedures run for a very long time (see notes above for high-dimensional data, but also an issue for some very large clinical data loads) it is difficult in earlier tranSMART releases to identify the step causing problems.

Although tranSMART ETL procedures log each step to the audit tables, this logging is part of the ETL transaction. If the transaction should fail or if the ETL job is canceled the logging data will also be lost.

The audit log utilities in tranSMART 19 can also write to the database log file. This is an immediate write and output can be followed while the ETL job is running. An added benefit is that when run from the command line the log output is also printed to the console. This provides an immediate report of the audit messages so an inspection of the code can indicate which step is currently running.

To set up this additional logging, create a row in the new *tm_cz.etl_settings* table:

psql -c "insert into tm_cz.etl_settings (paramname,paramvalue) values ('debug','yes')"

A second parameter is tested to skip the deletion of temporary tables so that their content can be inspected after an ETL job has run. The tables will be cleared at the start of a new ETL job for the same datatype.

psql -c "insert into tm_cz.etl_settings (paramname,paramvalue) values ('cleantables','no')"

cz_job_audit message position

Many messages reported “loading” with a row count for the previous step in earlier tranSMART versions.

All such messages should report the end of the step with its row count and description.

Function/stored procedure error checks

Several stored procedures reported errors, and returned an error status, but this was ignored by the calling procedure.

Release 19.0 checks the return status for all calls that have a return value.

Leading zeroes in Kettle job output

The job_id and log_base values werre reported with a large number of leading zeroes.

The datatypes used as outputs by stored procedures and the Kettle scripts have been changed and specific formats introduced to report only the integer value.

Cleanup of Kettle scripts

Kettle jobs have been pretty-printed to make the XML easier to read.

The return values from failed stored procedures have been standardized as zero for success and any other value for a failure. The tests in Kettle have been modified where the meaning of zero and one have been changed.

ETL stored procedure failures

Stored procedures called during ETL by Kettle and other ETL systems have been reviewed and updated.

As noted above, return values of non-zero now all indicate an error condition.

Many instances were found of procedures (functions in postgres) called by other procedures with no tests for their return values. In all cases these are now tested and will cause immediate action (usually a return with an error) by the calling procedure.

An example was an error in the calculation of log intensities and zscore values for some high-dimensional datatypes failing silently.

Validation of platform

All platform annotation files for all datatypes should include only one platform.

RNAseq platforms

RNAseq expression data now uses a named platform. Loading the platform annotation populates gene names and gene IDs as for Expression platforms.

Where the probe ID is an Ensembl gene ID a single platform for each species can cover all supposed GEO platforms. For RNAseq data GEO records the sequencing technology and the organism as the platform.

We plan to add inremental updating of these platforms on a per-study basis to catch any additional probe IDs a study may use where there is no Ensembl Gene ID defined, and also to catch the addition of new IDs by Ensemnbl after the original platform load.

Test for missing gene ID and gene name information are more efficient in this release. In earlier versions loading RNAseq platform data as an anopnymized incremental update could take considerable time.

RNAseq multiple procedures

tranSMART has two sets of RNAseq ETL procedures. One is for gene-based expression counts, the other is for expression mapped to chromosome position. They were implemented around the same time for the tranSMART 16.1 release.

The names are defined interbally in several places in the source code. They have been made clearer in tranSMART 19. The internal name RNASEQ_COG (developed by Cognizant for Sanofi) is used by RNAseq expression counts.

Renaming/moving a study

The postgresql script tm_cz.i2b2_move_study missed many of the changes needed to rename or move a study. The updated script requires two inputs: the original path of the top node for the study and the new path. Any new nodes are automatically created. The function takes an additional jobId parameter which is NULL when run from the command line.

Example:

```
psql -U tm_cz -W -c "select* from tm_cz.i2b2_move_study('\Public Studies\Asthma_Barczak GSE34466\', '\Public Studies\Asthma\Barczak GSE34466', NULL)"
```