Loading data with transmart-batch (Cell Line Use Case dataset)

Introduction

The CTMM TraIT project recently added the Cell Line Use Case (CLUC) to tranSMART. The CLUC is a collection of data on colorectal and prostate cell lines from an exceptionally broad set of platforms, as shown in the table below.

This diverse set is used to:

- Standardize data formats and data processing pipelines from the four domain
- Test the integration pipelines within the TraIT translational toolset

By incorporating the same platforms as used for ongoing research projects, this cell line set gives a representative test set comparable to real patient data, without the legal burden of handling personal data. The TraIT Cell Line Use Case transmart-ready files are available under the CC0 license for download here.

Please use the following citation when making use of this dataset: Bierkens, Mariska & Bijlard, Jochem *"The TraIT cell line use case."* Manuscript in preparation. More information can also be found on the Bio-IT World Poster *"Multi-omics data analysis in tranSMART using the Cell Line Use Case dataset"*.

	CACO2	CACO2_AURKA	COLO205	COLO320	DLD1	HCT116	HCT116_MLH1	HCT15	HT29	LIM1863	LS174T	LS513	PC346c	RKO	SW1398	SW48	SW480	SW480_AURKA	VCaP	Biological information
DNA and RNA arrays																				
ArrayCGH, Agilent, 180K			✓					✓		✓	✓					✓				CNA
ArrayCGH, Agilent, 244K	✓		✓	✓		✓	✓		✓			✓		✓	✓		✓			CNA
mRNA Arrays, Agilent, 44K	✓	✓	✓	✓		✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓		mRNA expression
Affymetrix Exon Array										✓									✓	mRNA expression
miRNA, Agilent										✓									✓	miRNA expression
DNA and RNA next generation sequencing																				
DNA-Seq, Complete Genomics													✓						✓	SNV
RNA-Seq, Illumina GAII	✓					✓			✓				✓				✓		✓	SNV & mRNA expression
mRNA-Seq, HiSeq 2000	✓																			SNV & mRNA expression
Mass spectrometry proteomics																				
LC-MS/MS	✓		✓		✓	✓			✓					✓	✓		✓			Protein quantification
Non-highthroughput molecular profiling																				
Microsatellite instability status	✓		✓	✓		✓		✓	✓	✓	✓	✓		✓	✓	✓	✓			Microsatellite (in)stability
DNA mutation	✓		✓	✓		✓		✓	✓	✓	✓	✓		✓	✓	✓	✓			DNA mutation status
MLPA-gain-P146	✓					✓			✓								✓			CNA
MLPA-loss-X006	✓								✓								✓			CNA
Promoter methylation	✓					✓			✓							✓				Promotor methlyation status
mRNA expression						✓														mRNA expression

Table of contents

General remarks

Note that folder structure is very important in the upload process, make sure to structure your data in the correct way (figure 1). For more detailed information about the data type you wish to load please refer to the section dedicated to that specific data type.

It is important to setup the batchdb.properties file to provide transmart-batch with the location and login information needed to load the data. A detailed explanation on the properties file can be found here.

For the tutorial the assumption is made that the data is loaded into a local database with default settings, meaning that the database is located on the same machine that has the data folders and the ETL pipeline scripts.

Important note: As transmart-batch currently does not have a pipeline for VCF data this data typ will have to be loaded with Kettle.

Setting up transmart-batch and general documentation

For the complete documentation on transmart-batch please look here.

To use transmart-batch with 16.1 or 16.2 you can use the V1.0 release. To use the latest version please clone the git repository and build tran smart-batch:

```
git clone https://github.com/thehyve/transmart-batch.git
cd transmart-batch
./gradlew capsule
```

After building you should see transmart-batch/build/lib/transmart-batch-1.1-SNAPSHOT-capsule.jar

Batchdb.properties file

The properties file contains information as the location of the database, the username and password that are used to upload the data to the database. The properties is build up of four lines indicating which database is being used, either PostgreSQL or Oracle, the location of the database and the user.

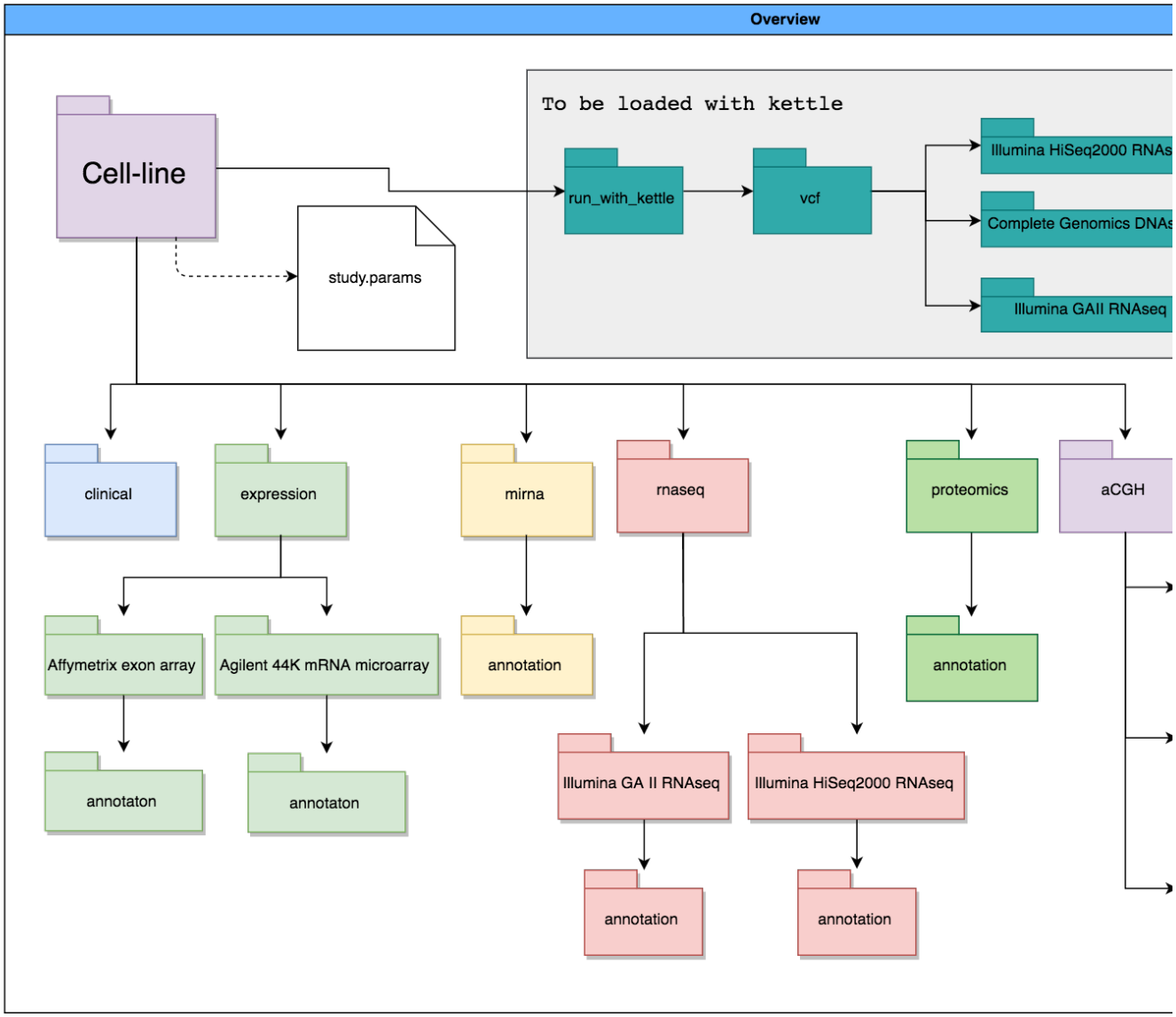
Example properties file (postgres)

```
PostgreSQL
    batch.jdbc.driver=org.postgresql.Driver
    batch.jdbc.url=jdbc:postgresql://localhost:5432/transmart
    batch.jdbc.user=tm_cz
    batch.jdbc.password=tm_cz
```

```
Oracle
    batch.jdbc.driver=oracle.jdbc.driver.OracleDriver
    batch.jdbc.url=jdbc:oracle:thin:@localhost:1521:ORCL
    batch.jdbc.user=tm_cz
    batch.jdbc.password=tm_cz
```

Data structure and loading the data

In order to load the data properly the scripts need to know were the data is located, in order to achieve this the data structure is more of less set. In the data (available here) the only thing you have to do is extract the files and you are ready to load. The following figure gives an overview of the data types and the way the folder structure is build up. More details about particular datatypes can be found in there respective sections.



Getting the data to the server

If you want to upload the data to a server you first need to get the data on the server. The easiest way to do this is by opening a terminal window and connect to the server:

```
ssh username@serverAddress
```

When the connection is made open a new terminal window (do not close the window where you connected to the server) and navigate to the study you want to copy. From the folder the study is located in run the following command:

```
scp -r study_name username@serverAddress:~(default, folder on server to
put the data, ~ is your home folder)
```

Loading the data

To load the data transmart-batch needs three files.

1. batchdb.properties file
2. study.params
3. data to be loaded params file, this can be the data type or the annotation platform params file

---

Clinical data

To load just the clinical data, run:

```
<path_to>/transmart-batch/build/lib/transmart-b
atch-1.1-SNAPSHOT-capsule.jar -c
<path_to>/batchdb.properties -p
<path_to>/clinical.params
```

If you are reloading data add the n flag, this forces transmart-batch to restart an already completed job again.

in the clinical data folder you need the following files:

- clinical.params

  Indicates where the column and word mapping file are located.

  ```
  # Mandetory
  COLUMN_MAP_FILE=Cell-line_columns.txt
  #Optional
  WORD_MAP_FILE=Cell-line_wordmap.txt
  ```

- Cell-line data file

There are three datafiles. The first contains the characteristics data, the second has the non-high throughput molecular profiling data (NHTMP) and the last was added to support EGA IDs. The names of the files can be arbitrarily chosen as long as they are specified in the column mapping file. The files should be tab-separated files.

column mapping file

The column mapping file contains 7 columns, filename, category code, column number, data label, data label source, control vocab cd and concept type. The filename is the file name of a tab separated datafile, the category code is used to indicate part of the tree shown in tranSMART (subject is a reserved term to indicate the patients). The column number indicates which column should be used from the datafile, the data label indicates the leaf note name (SUBJ_ID is a reserved term to indicate the subjets). The data label source and control vocab cd columns can be empty. The last column, concept type, is an optional column used in transmart-batch to indicate either NUMERICAL or CATEGORICAL data values. Each Category Code - Data label pair should have a unique name, and each unique name should have 1 column assigned from a data file.

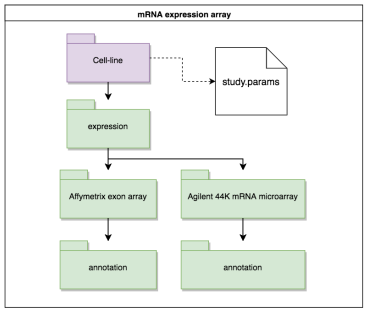
word mapping file

The word mapping file can be used to transform values in the data file according to for example a codebook. The file should contain 4 columns, a file name where the value is located that should be replaced, the column number of the concept in the data file, value to be replaced, new value.

```
FILENAME COLUMN_NUMBER FROM TO
Cell-line_data.txt 3 1 Male
Cell-line_data.txt 4 Yes 1
```

In case of the CLUC data there are 3 data files, the first contains the characteristics data, the second data file contains some non-high throughput molecular profiling (NHTMP) data describing gains or losses of selected genes and the last was added to support EGA IDs. The column mapping file maps the columns in the datafolder to the correct tree structure shown in tranSMART, it tells for example that column 5 in the data file is the age column and should be stored under a variable called Age.

Gene expression data



Before data can be loaded into tranSMART, the platform used to generate the data must be loaded. The annotation/platform files are located in annotation folders (see image to the right) and have their own params files to load the annotation data.

Microarray data

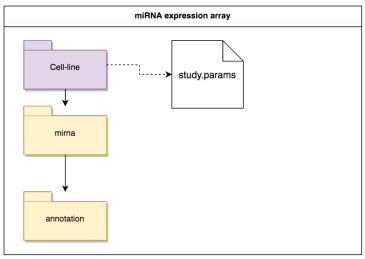
mRNA array

Annotation data:

```
<path_to>/transmart-batch/build/lib/transmart-b
atch-1.1-SNAPSHOT-capsule.jar -c
<path_to>/batchdb.properties -p
<path_to>/mrna_annotation.params
```

Measured data

```
<path_to>/transmart-batch/build/lib/transmart-b
atch-1.1-SNAPSHOT-capsule.jar -c
<path_to>/batchdb.properties -p
<path_to>/expression.params
```



miRNA

> **Requires transmart-data and kettle**
> This requires you to setup transmart-data, or use an already setup version. How to setup transmart-data for this can be found here.

Next to platform data miRNA also needs a dictionary before data can be used in advanced analysis. The dictionary maps the small miRNA parts to genes which allows tranSMART to use the miRNA in the advanced analysis work flow. With a clean tranSMART installation only the gene dictionary is loaded, so it could be the miRNA dictionary is not loaded for your instance of tranSMART. To load the dictionary run the following command from the `transmart-data/` folder:

```
make -C data/postgres/ load_mirna_dictionary
```

if the dictionary is already present running this command will just return that the dictionary is already loaded.

Agilent miRNA microarray

The annotation/platform information is loaded together with the data. The image on the right shows the file structure for this to work.

miRNA microarray annotation and data

```
<path_to>/transmart-batch/build/lib/transmart-b
atch-1.1-SNAPSHOT-capsule.jar -c
<path_to>/batchdb.properties -p
<path_to>/mirna_annotation.params

<path_to>/transmart-batch/build/lib/transmart-b
atch-1.1-SNAPSHOT-capsule.jar -c
<path_to>/batchdb.properties -p
<path_to>/mirna.params
```

in the miRNA folder there are 4 files that are needed for the upload to be successful:

- mirna.params
  Example file:

```
# Mandatory
DATA_FILE_PREFIX=mirna_data
MAP_FILENAME=mirna_subject_sample_mapping.txt
SAMPLE_MAP_FILENAME=mirna_sample_mapping.txt
MIRNA_TYPE=MIRNA_QPCR
INC_LOAD=N
DATA_TYPE=R
# Optional
```

mirna_data.txt

Name of the data file is specified in the params file. The file should be a tab separated file with in the first column "id_ref" which refers to the annotation and the second to nth column representing samples. All the values should be in quotes:

```
"ref_id"
"1"
"2"
.........

sample 1
"-0.84"
"0"
...sample n
"-0.225"
"0"
```

mirna_sample_mapping.txt

Empty file, this file needs to be present for the upload to work.

mirna_subject_sample_mapping.txt

Name of the subject sample map is specified in the params file. The tab separated file has 10 columns, from left to right "trial_name", "site_id", "subject_id", "sample_cd", "platform", "tissue_type", "attr1", "attr2", "cat_cd", "src_cd". The "platform" column should contain the platform id under which the miRNA annotation was uploaded. The "cat_cd" column contains the path to the data as shown in the folder structure in tranSMART. All fields should be enclosed in quotes.

RNAseq data

For the RNAseq data the folder structure is different from the other data types. The platform annotation files for the HiSeq2000 datasets are directly in HiSeq2000 folder and not nested with each dataset. This is done to reduce redundancy, all six different datasets use the same platform meaning it only has to be loaded once. The annotations should be loaded before the rest of the data.
Loading annotation data

```
<path_to>/transmart-batch/build/lib/transmart-batch-1.1-SNAPSHOT-capsule.jar -c <path_to>/batchdb.properties -p <path_to>/rnaseq_annotation.params
```

Illumina GA II RNAseq

The RNAseq data contains (sequence) read counts for transcripts, i.e. it is a measurement for the relative abundance of transcripts.
Loading data

```
<path_to>/transmart-batch/build/lib/transmart-batch-1.1-SNAPSHOT-capsule.jar -c <path_to>/batchdb.properties -p <path_to>/rnaseq.params
```

The folder should contain the following three files:

- Cell-line-data.txt

  No strict file name, is specified in the params file.

  The first column in the file is the GENE_ID/PROBE_ID as specified in the platform annotation. The second till the last column start with sample name with an added .readcount, .normalizedreadcount or .zscore. See the full documentation here.

- Cell-line_subjectmapping.txt

  No strict file name, is specified in the params file

  Tab separated file containing 10 columns, STUDY_ID, SITE_ID, SUBJECT_ID, SAMPLE_ID, PLATFORM, SAMPLETYPE, TISSUETYPE, TIMEPOINT, CATEGORY_CD and SOURCE_CD. For more information please follow this link or check the example file.

- rnaseq.params

  Name is predetermined, contains the names of the data file and mapping file and optional settings for the source_cd.

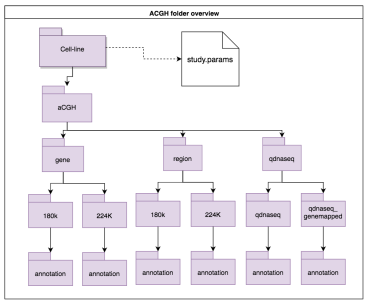
```
# Mandatory
RNASEQ_DATA_FILE=Cell-line-data.txt
SUBJECT_SAMPLE_MAPPING=Cell-line-subject_sample_mapping.txt
# Optional
SOURCE_CD=RNASEQGAIIMRNA
```

Illumina HiSeq2000 RNAseq

Similar to GA II RNAseq but more sub folders, each subfolder contains 1 sample. Each sample sub folder has the same three files as displayed above. To load all the datasets you will have to run the data loading command six times, once for each subfolder.

Copy Number Variation ('aCGH') data

For the array CGH the data is available from two different microarrays and one low coverage sequencing, 180k and 224k Agilent microarrays and qDNAseq respectively. The data has been processed on gene and region level generating a total of six data sets to load. The figure on the right shows the folder structure of the data. The platform annotation needs to be loaded before the actual data. Note that for this datatype the parameter files are called 'cnv.params' instead of 'acgh.params'.



Loading the annotation:

```
<path_to>/transmart-batch/build/lib/transmart-b
atch-1.1-SNAPSHOT-capsule.jar -c
<path_to>/batchdb.properties -p
<path_to>/cnv_annotation.params
```

Loading the data:

```
<path_to>/transmart-batch/build/lib/transmart-b
atch-1.1-SNAPSHOT-capsule.jar -c
<path_to>/batchdb.properties -p
<path_to>/cnv.params
```

All of the folders containing data should have the following three files:

Cell-line_samples.txt

No strict file name, is specified in the params file.

The file structure is build up as the first column containing either gene id or region id and each set of 7 columns following this first describe a sample. From left to right these columns are: sample.chip, sample.segm, sample.flag, sample.loss, sample.norm, sample.gain, sample.amp. The columns are separated by tabs. For more information please follow this link.

Cell-line_subjectmapping.txt

No strict file name, is specified in the params file

Tab separated file containing 10 columns, STUDY_ID, SITE_ID, SUBJECT_ID, SAMPLE_ID, PLATFORM, SAMPLETYPE, TISSUETYPE, TIMEPOINT, CATEGORY_CD and SOURCE_CD. For more information please follow this link or check the example file.

cnv.params

Name is predetermined, contains the names of the data file and mapping file and optional settings for the source_cd.

```
# Mandatory
DATA_FILE_PREFIX=Cell-line_samples.txt
MAP_FILENAME=Cell-line_subjectmapping.txt
# Optional
SOURCE_CD=STD2
```

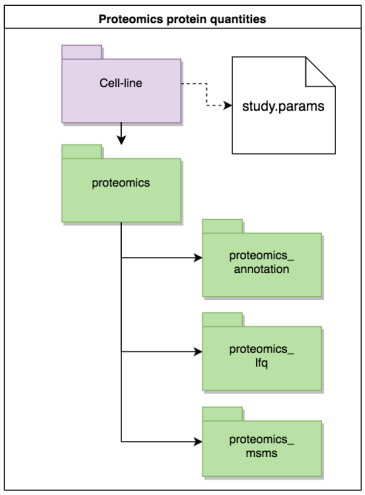
Proteomics

> **Requires transmart-data and kettle**
> This requires you to setup transmart-data, or use an already setup version. How to setup transmart-data for this can be found here.

Next to platform data Proteomics also needs a dictionary before data can be used in advanced analysis. The dictionary maps the proteins to genes which allows tranSMART to use the proteins in

the advanced analysis work flow. With a clean tranSMART installation only the gene dictionary is loaded, so it could be the protein dictionary is not loaded for your instance of tranSMART. To load the dictionary run the following command from the `transmart-data/` folder:

```
make -C data/postgres/
load_proteomics_dictionary
```



LFQ and MS/MS

Protein quantities

In the proteomics folder you will find a annotation folder and two data folders. To load the proteomics data first load the annotation file, which is the same for both data sets. Afterwards you can load the LFQ and MSMS datasets which are different representations of the same data (Label Free Quantification vs Mass spec)
Loading the annotation:

```
<path_to>/transmart-batch/build/lib/transmart-b
atch-1.1-SNAPSHOT-capsule.jar -c
<path_to>/batchdb.properties -p
<path_to>/proteomics_annotation.params
```

Loading the data:

```
<path_to>/transmart-batch/build/lib/transmart-b
atch-1.1-SNAPSHOT-capsule.jar -c
<path_to>/batchdb.properties -p
<path_to>/proteomics.params
```

In the proteomics folder there are three files that are needed for the upload to be successful:

- proteomics.params

  Name is predetermined, contains the names of the data file and mapping file and optional settings for the source_cd.

  ```
  # Mandatory
  MAP_FILENAME=proteomics_subject_sample_mappin
  g.txt
  DATA_FILE_PREFIX=proteomics_data.txt
  # Optional
  DATA_TYPE=R
  ```

- proteomics_data.txt

  Tab separated file containing the data.

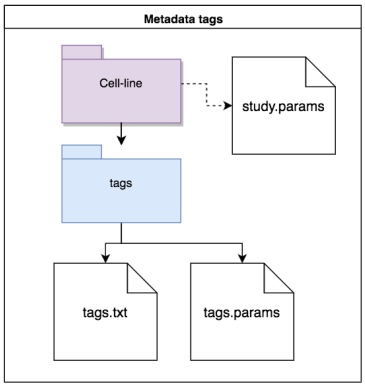
  The first column in the file should correspond to the platform probe IDs, rest of the columns need to be sample IDs as used in the subject-sample mapping.

- proteomics_subject_sample_mapping

  No strict file name, is specified in the params file

  Tab separated file containing 10 columns, STUDY_ID, SITE_ID, SUBJECT_ID, SAMPLE_ID, PLATFORM, SAMPLETYPE, TISSUETYPE, TIMEPOINT, CATEGORY_CD and SOURCE_CD. For more information please follow this link or check the example file.

Metadata tags

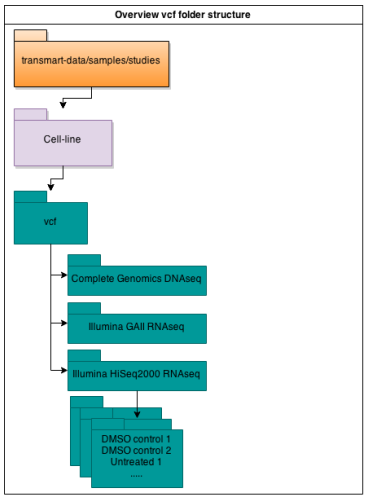


Now that all data, apart from VCF, has been loaded it is possible to load the metadata for the Cell-line use case. The loading is similar to loading clinical data and can be done by running the following command:

```
<path_to>/transmart-batch/build/lib/transmart-batch-1.1-SNAPSHOT-capsule.jar -c <path_to>/batchdb.properties -p <path_to>/tags.params
```

Note: This load job will give feedback that not all the tags have been loaded, this is due to the VCF data not being loaded yet and will be fixed in the VCF section when you read the VCF tags section.

Small Genomic Variants ('VCF')



For this transmart-data with the old pipeline is used as there is no pipeline to load VCF data in transmart-batch. To use this pipeline you need to add the VCF data to the transmart-data folder as shown in the image to the right. To setup transmart-data please look here.

In the Cell-line use case study folder there is a folder called run-with-kettle, in this folder the VCF data is located. This folder also has its own study.params file as the format of this parameter file changed when migrating from kettle to transmart-batch loading. A nice way to add the data to your transmart-data folder is to add a symbolic link:

Adding a symbolic link

```
ln -s <path_to>/run-with-kettle <path_to>/transmart-data/samples/studies/Cell-line
```

After executing this command you should now see a folder called 'Cell-line' in transmart-data/samples/studies. The Cell-line folder should contain the study.params and the vcf folder.

There a total of 8 datasets available obtained from 3 different platforms. To load all of them at once go into the Cell-line/vcf directory and run

```
bash load.sh
```

Note: make sure you have sourced the vars file from transmart-data or this will not work.

As these datasets are quite large compared to the other data types loading may take some time.
Note: uploading a vcf dataset twice will result in undefined behaviour, because the "old" dataset is not removed.

Complete Genomics DNAseq & Illumina GAII RNAseq both have one vcf file to load while Illumina HiSeq2000 RNAseq contains the remanning 6. Each folder with a vcf file should have the following three files

- Cell-line.vcf

  The actual VCF file. The VCF files for the Cell-line use case are annotated with HGNC gene symbols and Ensembl Gene Identifiers. For more information about the VCF file format please follow this link.

- Subject_sample_mapping

The subject sample mapping file maps the actual sample names to the sample IDs given in the VCF file. For example VCaP is given the ID GS000008107-ASM in the VCF file.

vcf.params

Specifies the VCF file to upload, the subject sample map to use, genome build used to process the samples and builds the concept path shown in the tranSMART tree. Click here to see an example file with more detailed explanation.

IMPORTANT!

As the VCF pipeline has a separate way of handling study IDs you will probably see greyed out VCF nodes in the database. To make these available you need to connect to your database with pgadmin, psql or Oracle developer and execute the following SQL query:

Make VCF available

```
insert into i2b2metadata.i2b2_secure
        (c_hlevel,c_fullname,
        c_name,c_synonym_cd,
        C_VISUALATTRIBUTES,C_BASECODE,
        C_FACTTABLECOLUMN,C_TABLENAME,
        C_COLUMNNAME,C_COLUMNDATATYPE,
        C_OPERATOR,C_DIMCODE,
        C_COMMENT,C_TOOLTIP,
        UPDATE_DATE,DOWNLOAD_DATE,IMPORT_DATE,
        SOURCESYSTEM_CD,I2B2_ID,
        M_APPLIED_PATH,SECURE_OBJ_TOKEN)
select
        c_hlevel,c_fullname,
        c_name,c_synonym_cd,
        C_VISUALATTRIBUTES,C_BASECODE,
        C_FACTTABLECOLUMN,C_TABLENAME,
        C_COLUMNNAME,C_COLUMNDATATYPE,
        C_OPERATOR,C_DIMCODE,
        C_COMMENT,C_TOOLTIP,
        UPDATE_DATE,DOWNLOAD_DATE,IMPORT_DATE,
        SOURCESYSTEM_CD,null,
        M_APPLIED_PATH,'EXP:PUBLIC'
from
        i2b2metadata.i2b2
where
    sourcesystem_cd like 'TraIT-Cell-line_%';
```

Adding additional meta data tags to the VCF nodes:

Because of the study ID quirk mentioned above the transmart-batch pipeline is unable to load metadata on the actual high dimensional nodes. In the Cell-line_use_case_V1.1/tags folder there is a file called vcf_tags.txt, in this file there is a SQL query that can be used to insert the final metadata into the database.

Setting up transmart-data

To upload the Cell-line use case data you need the transmart-data folder, which can be found here:

```
git clone https://github.com/transmart/transmart-data
```

transmart-data contains a collection of scripts and the basic folder structure you need to upload the data. After downloading the transmart-data folder from github you need to update the ETL-pipeline by running the following command from the transmart-data folder:

```
make -C env update_etl_git
make -C env data-integration
```

After the transmart-ETL update is done there should now be two folders in /transmart-data/env called tranSMART-ETL and data-integration.

The next step is to configure the `vars` file in transmart-data, there is a sample file called `vars.sample`, make a copy of this and name it `vars`.
The vars file contains information for both oracle and postgres databases, as we are using postgres so only the following parameters must be set correctly:

Example vars file

```
PGHOST=localhost
PGPORT=5432
PGDATABASE=transmart
PGUSER=tm_cz
PGPASSWORD=tm_cz
PGSQL_BIN="/usr/bin/"
KETTLE_JOBS_PSQL=/opt/transmart-data/env/tranSMART-ETL/Postgres/GPL-1
.0/Kettle/Kettle-ETL/
KETTLE_JOBS=$KETTLE_JOBS_PSQL
R_JOBS_PSQL=<path_to>/transmart-data/env/tranSMART-ETL/Postgres/GPL-1
.0/R
KITCHEN=<path_to>/transmart-data/env/data-integration/kitchen.sh
KETTLE_HOME=<path_to>/transmart-data/samples/postgres/kettle-home
PATH=<path_to>/transmart-data/samples/postgres:/opt/R/bin:$PATH
export PGHOST PGPORT PGDATABASE PGUSER PGPASSWORD PGSQL_BIN \
        KETTLE_JOBS_PSQL KETTLE_JOBS R_JOBS_PSQL KITCHEN KETTLE_HOME
PATH
```

More information on the variables

PGHOST

By default set to localhost

PGPORT

This is the port on which the database can be reached. Default this is set for a localhost to port 5432. When using a server you need to forward this to the port the SSH connection is established.

PGDATABASE

Database name to which the data will be uploaded, the default name of the database is transmart

PGUSER

Your username to access the database. Default set to tm_cz
NOTE: this is not the same as a login used to access the data via the web client.

PGPASSWORD

Password to access the database. Leave empty if there is no password. Default is tm_cz

PGSQL_BIN

Path to the directory where postgres is installed. When installed locally with a package manager the location probably will be `/usr/`

`local/bin/`. be sure to end the pathname with a /

- KETTLE_JOBS_PSQL

  path to where the kettle scripts are located. If you updated the ETL-pipeline used above the location will be `<path_to>/transmart-data/env/tranSMART-ETL/Postgres/GPL-1.0/Kettle/Kettle-ETL/`. be sure to replace `<path_to>` with actual location of the folder.

- KETTLE_JOBS

  Same as KETTLE_JOBS_PSQL. KETTLE_JOBS=$KETTLE_JOBS_PSQL

- KETTLE_HOME

  This is in the transmart-data folder under `samples/postgres/kettle-home`. KETTLE_HOME=`<path_to>/transmart-data/samples/postgres/kettle-home`

- KITCHEN

  This is in the transmart-data folder under env/data-integration/kitchen.sh. KITCHEN=`<path_to>/transmart-data/env/data-integration/kitchen.sh`

- R_JOBS_PSQL

  Points to R script used when uploading clinical data. Can be found in the transmart-data folder under: `<path_to>/transmart-data/env/tranSMART-ETL/Postgres/GPL-1.0/R`

- PATH (adding loading script location)

  For the `load.sh` scripts to find the loading scripts the location needs to be added to the path. The loading scripts are in `<path_to>/transmart-data/samples/postgres`.

  PATH=`<path_to>/transmart-data/samples/postgres:$PATH`

- export

  Lastly the parameters set need to be exported to the environment.

  export PGHOST PGPORT PGDATABASE PGUSER PGPASSWORD PGSQL_BIN \

  KETTLE_JOBS_PSQL KETTLE_JOBS R_JOBS_PSQL KITCHEN KETTLE_HOME PATH

When you are done setting up the vars file, in the transmart-data folder run:

```
source vars
```