**Pfizer tranSMART version 1.1**

1. **GENERAL INTRODUCTION**

The tranSMART application reflects the efforts of various informatics groups to integrate data from internal and external data sources within a single data warehouse, and to provide scientific end users the tools to search for, view, and analyze the data in the warehouse.

tranSMART provides the following tools:

1. **Search** – Search across internal and external data sources for research data and literature related to search terms that you provide. **(Our Main focus)**

**Dataset Explorer** – View study data for subjects that you select, based on criteria that you specify. Also, compare data generated for subjects in two different study groups, based on criteria and points of comparison that you specify. **(Our Main focus)**

1. **Gene Signature/Lists** – View definitions of existing gene signatures and add new gene signature definitions.
2. **Utilities** – Contains the following submenus:
3. **Help** – Display links to the tranSMART documentation set.
4. **Contact Us –** Email questions, problem reports, enhancement requests, or any other feedback about the tranSMART application.
5. **About** – Displays the version of tranSMART.
6. **Pfizer STUDY DOMAINS**

Two domains:

1. Clinical Data Study – Details regarding the various tests performed in clinical trials
2. GWAS – Genome Wide Association Study
3. **GENERAL WORKFLOW**

Clinical Trial / GWAS Data is obtained from the Client

Preparation of required files (mapping files / converter)

ETL Execution for transformation of data before loading

Loading of data in staging & prduction environment via oracle SQL developer

Testing and verification of data and output of statistical analysis (e.g. heat map) in tranSMART UI for the end users

1. **TECHNOLOGY PRE-REQUISITES**
2. **Web browser (Internet Explorer)**

* Go to [www.virtual.pfizer.com](http://www.virtual.pfizer.com) 🡪client application server 🡪 login
* Enter Username: karan02
* Enter password: Jan\_2015
* Security connected blue colored sign will appear in Taskbar, once connected.

1. **Virtual Desktop Machine:** VMWare Horizon View (VDI)

Steps involved:

* Connection Server: vdi.pfizer.com
* Click on Connect
* Enter Username: karan02
* Enter password: Jan\_2015
* Domain APAC 🡪 click on connect
* Legal Notice 🡪 Click Ok.

1. **Pfizer Outlook and Communicator in remote desktop.**

**(Username and password not known)**

1. **Oracle SQL Developer**

* Four working environments are present:

1. Development: For rough draft work usage by the developer. Not available with all the clients.
2. New Staging: Hit and Trial purpose for developer, tester and validator.
3. Production: Used by Developer and Tester. Final Data when incorporated here is visible to the end user.
4. Stage Database: **Not clear**

* Select the Staging environment 🡪 tranSMART New Stg DB 🡪 right click 🡪options 🡪host name**🡪?? (doubtful)**
* Username and password both are tm\_cz
* **Got experience only in New Staging environment and not worked on rest three environments.**

1. **Putty for Linux environment in remote desktop**

* Click on centrify 🡪 putty
* User: tmetl/Pf384916
* **Password not known**
* If date appears, that means we are logged in.

1. **WinSCP in remote desktop**

* For transfer of files from windows to linux environment.
* **Username and password not known.**

1. **tranSMART website in remote desktop**

* **Staging:** transmartstg.pfizer.com/tranSMART/RWG/index
* **Production:** transmart.pfizer.com/tranSMART/RWG/index **(not worked on this)**

1. **Extraction Transformation & Loading (ETL) UI – Pentaho in remote desktop**

* My Documents 🡪 Data Integration 🡪 spoon.bat

1. **CLINICAL DATA LOADING**
2. **REQUIRED FILES**
3. **Raw Data File**

* It is provided by the client – .txt, .csv, .xls
* Has to be converted into a tab delimited text file (if not).
* Rows represent the subjects / patients / cases and columns represent the variables / features.
* First column of the data file should always contain unique subject identifiers (subject or study ID), otherwise there will be discrepancy later while mapping and hence loading by the ETL. We need to shift the study ID column if not present on first column.
* Rest columns pertain to the clinical trial tests or related data, that appear as node names in tranSMART UI application for the end user.
* **Path:** /hpc/grid/omics\_data/tranSMART/ClinicalStudies/copdgene/data (can be changed as per our convenience.)

**Need to look into the data file again for all the column names and their significance**

1. **Dictionary File**

* It is provided by the client.
* It denotes the name of the nodes to be displayed in tranSMART UI for the end users.
* It is a reference file required while preparing column and word mapping files and is not required in the ETL execution process.
* **Path:** Not required

1. **Mapping files**

* There are two types of mapping files, the column mapping file handling the column of the raw data files, and the word mapping file handling the term of the raw data files.
* **Path:** Same as raw data file both the mapping files

1. **Column Mapping File**

* Tab delimited file
* All the lines must have at least six columns (provided by the client).
* First Column: Filename - Name of the raw data file (e.g. abc.txt)
* Second Column: Category\_Code – Navigation string or the path of the tree that appears on tranSMART UI. ‘+’ denotes the node demarcation and ‘\_’denotes space in category code. The path defined here is the one that starts after the path defined in top node (in .sh file).
* Third Column: Column\_Number – Denotes the corresponding column number in raw data file.
* Fourth Column: Data\_Label - **?? (Wherever the data needs to be grouped this column is denoted as “ \ ”).**
* Fifth Column:Data\_Label\_Source **–** For grouping purpose: denotes the column number of data file, according to which the grouping has to be done. (Informed by the client). In case of double grouping, the column numbers are separated by “ **;** ” (e.g.: 3;4).
* Sixth Column: Controlled\_Vocabulary\_Code - **??**
* Seventh Column: Data\_Value\_Code - **??**
* Eighth Column: Preserve\_Null\_Value - **??**
* Important Points:
* The second line of the column mapping file will be different from the remaining lines, as follows:
* **Filename.txt <tab> <tab> <tab> 1 <tab> SUBJ\_ID**
* This line defines that the first column in the raw data file is the subject Id which is the primary key.
* The third line of the column mapping file will be different from the remaining lines, as follows: **(doubtful)**
  + - * **Filename.txt <tab> Path <tab> 3 <tab> DATA\_LABEL**
      * Third column name is stored in a variable namely DATA\_LABEL and is used in the Category\_Code of subsequent lines.
* Rest of the lines will be in similar format.

**Got the following information from a tranSMART help file and I could not understand the following points regarding column mapping file.**

* All the raw data files listed must have been selected in the step “Select raw files”.
* A line has to be set as the subject identifiers.
* All column numbers have to be set, and have to be numbers.
* All data labels have to be set.
* All category codes corresponding to properties have to be set.
* If a property has a data label source (data label is set to “\”), this data label source has to be provided.

1. **Word Mapping file**

* All the lines must have four columns
* To be checked and prepared in accordance with the dictionary file which is obtained from the client.
* Column 1: Raw data filename (e.g. abc.txt)
* Column 2: Column Number of the data file that requires categorical variable name display instead of numerical value.
* Column 3: Old entry (e.g. 1)
* Column 4: New Entry (e.g. Visit 1)

**Got the following information from a tranSMART help file and I could not understand the following points regarding word mapping file.**

* All the raw data files listed must have been selected in the step “Select raw files”.

1. **Executable .sh file**

* Shell script file that stores the bash commands to read the data and mapping file, run the ETL job and load the tables into Oracle database.
* First line: **Defines the path of another shell script named kitchen.sh** which helps in executing the ETL script file (.kjb)
* Second line: **Defines the navigation path (location) of the ETL script file** that needs to be executed.
* Third line: **Defines the name of log file** that is created after completion of .sh file execution.
* Fourth line: **Column mapping file parameter set:** Name of the column mapping file needs to be added.
* Fifth line: **Word mapping file parameter set:** Name of the word mapping file needs to be added.
* Sixth line: **Data Location parameter set:** Navigation path of the three files viz data file, word mapping file and column mapping files should be the same and added here.

E.g.: /hpc/grid/omics\_data05/tranSMART/Clinical\_Studies/copdgene/Data

* Seventh Line: Load type = I (fixed, not to be changed) **Doubtful**
* Eighth Line: Security Required =N (fixed, not to be changed) **Doubtful**
* Ninth Line: SORT\_DIR = /hpc/grid/omics\_data05/tranSMART/tmp (fixed, not to be changed)
* Tenth Line: Top\_Node = ‘\Respiratory Tract diseases\COPD\_COPDGENE\’. It is required as the common top nodes below which the subsequent nodes will be added, according to the categorical code mentioned in the column mapping file.
* Eleventh Line: Record\_exclusion\_file = x (fixed, not to be changed) **Doubtful**
* Twelfth Line: log\_date: To record the end time of successful ETL execution and database loading.
* **Path:** /hpc/grid/omics\_data05/tranSMART/Karan01/GWAS (working directory).
* The path where .sh file is kept becomes the working directory; the log file is also created here.

1. **Steps Involved**

On bash command line (via putty):

**Step 1:** All the data files are transferred to a specific location (customizable) within linux environment using WinSCP or ask the client to transfer them into that location directly.

**Step 2:** Now in the linux environment, using putty, set the environmental variable i.e. $home (export command) i.e. the navigation path is set for the folder containing all the ETL files.

**export KETTLE\_HOME = /hpc/grid/omics\_data03/tmetl1/TMSTAGE**

**Step 3:** Change directory (cd command) to the working directory location i.e. where the .sh file has been kept.

**cd /hpc/grid/omics\_data05/tranSMART/karan01/GWAS**

**Step 4:** Verify if present in the correct folder (pwd) and list (ls –lrt command) the folder contents and check if .sh file is present or not.

**pwd: Print Working Directory**

**ls -lrt**

l-> Long Listing Format (permission,file count,owner,filesize,modif time..)

r->Reverse order while sorting

t->sort by modification Time

**Step 5:** Execution of the .sh file in the background using nohup command (no hang up), that runs the given command with hangup signals ignored, so that the command can continue running in the background after one log out.

**nohup ./load\_copdgene.sh**

**Ctrl+C**

Nohup.out file is generated containing the log of the ETL execution, errors (if any) and loading into database, in the same working directory that contains the .sh file.

**(Doubtful about location of nohup.out)**

Execution steps that are documented in nohup.out file will also start to appear on the command line using following command line.

**tail –f nohup.out**

**Step 6:** Once the ETL execution is complete, then a job ID is received that signifies the commencement of loading of data in the Oracle Database. To view the status of loading, Oracle SQL Developer is used.

Once the job Id is reflected on the command line, it is copied and used to check the status of loading steps or errors while data transfer into the database using the SQL Developer, which understands SQL commands as below:

**Select \* from tm\_cz.cz\_job\_audit where job\_id = “12345” order by 1 desc;**

**Select \* from tm\_cz.cz\_job\_error where job\_id = “12345”**

where, tm\_cz corresponds to the user and cz\_job\_audit & cz\_job\_error refer to the tables containing the log of data transfer and error details respectively.

Once loading into database is accomplished, command line echoes “Load Finished”.

**Step 7: Verification using target tables:**

Whether the data is loaded into the database or not, certain **target tables** in the Oracle database are required to be checked for new entries using Oracle SQL Developer namely:

1. **patient\_dimension:** One can verify if the new data has been loaded or not using unique study name, system\_source (vendor).
2. **concept\_dimension:** One can verify the concept\_path (nodes navigation)
3. **observation\_fact:** This table is linked to the tables namely patient\_dimension and concept\_dimension via separate primary keys (patient\_num and concept\_id respectively) and provide other relevant information regarding the loaded clinical data.
4. **wt\_trial\_nodes:** Stores the information regarding the nodes.
5. **i2b2:** Describes the total no. of nodes for a navigation path. **(Doubtful regarding the role)**

**Important SQL commands:** These can again be executed in similar fashion as above in Oracle SQL Developer:

1. select \* from concept\_dimension where concept\_path like ‘%abc%’;
2. select \* from observation\_fact where concept\_id = ‘%abc%’;
3. select \* from i2b2 where c\_fullname like ‘%abc%’
4. **GWAS DATA LOADING**

**Steps Involved:**

**STEP 1:** **Raw Data files** (abc.txt) received form the clients are transferred into specific location in the linux environment using WINSCP or ask the client to transfer them into that location directly. rsid is one of the important columns in the raw data file. Chromosome number and position are automatically fetched during ETL execution and are not required in the data file.

**Path:** /hpc/grid/omics\_data05/tranSMART/karan01/GWAS

**STEP 2:** **Modified Data File Preparation:** The raw data needs to be modified into a format that is understood by the ETL using a converter (e.g. Perl converter). This step can be skipped if the client directly provides the file in the required format.

* In the perl converter, the pvalue array position needs to be changed and position of output array elements needs to be changed according to the required header that is understood by ETL.
* Place the perl converter file in the same path/location.
* **Path:** /hpc/grid/omics\_data05/tranSMART/karan01/GWAS
* Execute the file using the command: perl\_converter.pl raw\_data\_file.txt raw\_data
* Output will be a modified version of the raw data understood by ETL: raw\_data.conv.txt or raw\_data.pass.txt (depending upon the extension mentioned in the perl converter).

**STEP 3: Metadata File Preparation**

STEP 3a: Study Metadata File Preparation:

* **Step i**: Check if study ID already exists in the “**bio\_experiment”** table using following SQL command in the oracle SQL developer.

**Select \* from BIOMART.bio\_experiment where title like ’%study\_name%**’

* If the study ID already exists, then goto to step 3b, else follow the below mentioned steps:
* **Step ii:** Prepare study metadata file, by editing the Study Name, Study Description and institution name.
* **Step iii:** Edit the study metadata executable file (.sh) by adding the name of the study metadata file in it.
* **Step iv:** Execute the .sh file using the command: **./study\_metadatafile.sh**. Permission mode needs to be changes before execution using command:

**chmod 755 study\_metadata\_file.sh**

* **Step v:** Verify if the Study ID has been loaded using following command in Oracle SQL developer: **Select \* from BIOMART.bio\_experiment order by desc 1**

STEP 3b: Analysis Metadata File Preparation:

* **Step i**: Using the dictionary file provided by the client, the analysis metadata file is created manually (e.g. analysis\_metadata\_file.txt). One of the columns in metadata file would be input location, where the full path of the converted/modified data file is entered.
* **Step ii:** Edit the analysis metadata executable file (.sh) by adding the name of the analysis metadata file in it.
* **Step iii:** Execute the .sh file using the command:

**nohup ./analysis\_metadata\_file.sh analysis\_metadata\_file.txt &**

* The ETL runs and in order to fetch the etl ID, type the following command in the Oracle SQL Developer:

**Select \* from TM\_LZ.lz\_src\_analysis\_metadata order by etl\_date desc;**

Get the ETL Id from the target table displayed. This ETL Id can then be used to track any error while ETL executions using the following command:

**Select \* from TM\_LZ.lz\_src\_analysis\_metadata where etl\_id=’’;**

* **Step iv:** Data loading in Oracle Database: Same day at 10 pm (IST), a stored procedure is run by the client to load the data directly into the database or else can be executed manually by us, using following steps too:

1. Procedure name that needs to be executed is “**I2B2\_MOVE\_ANALYSIS\_TO\_PROD\_NEW**”.
2. Select the procedure 🡪 A dialogue box appears 🡪 Run PL/SQL 🡪Edit the ETL ID at I\_ETL\_ID=’’. 🡪 Click ok 🡪 a red box will appear on the top and running status will appear at the bottom of the page 🡪 Execution / Job Completed and red box will disappear. Note the bio\_assay\_analysis ID in the output table.

* **Step v:** Verification of data by Target tables using following commands:

1. **Select \* from BIOMART\_STAGE.bio\_assay\_analysis\_gwas where etl\_id =’’;** check the no. of rows.
2. **Select \* from BIOMART.bio\_assay\_analysis\_gwas where bio\_assay\_analysis\_id=’’;** and check the no. of rows.
3. In command prompt (putty), use the command wc –l data\_file.conv.txt to check the no. of lines in the modified version of the raw data file. The row count should be same in both the tables signifying the data after etl execution and after loading in the oracle database respectively.
4. For top 50 entries, use the following command:

**Select \* from BIOMART.bio\_asy\_analysis\_gwas\_top50 where bio\_assay\_analysis\_id=’’;**

**The data can be best verified directly in the Pfizer staging application directly**

TO DO LIST

1. What is cohort?
2. Expression/ biochemical data files, high dimensional and incremental data loading not known yet, as Pfizer doesn’t deal with these data types – Need to have KT and hands on.
3. ETL UI and architecture in depth understanding required, especially to troubleshoot errors, which is the one of the main tasks that requires expertise.
4. Oracle in depth understanding required – going through oracle KT on MyLearning
5. Need KT and hands on for Java Groovie UI.
6. Need KT and hands on for statistical output and heat map testing on tranSMART application.
7. Hands on using a fresh clinical study data and target tables usage is still pending. GWAS KT along with fresh GWAS data loading is complete.
8. Got experience only in New Staging environment and not worked on rest three environments (e.g. production, staging and development) yet.
9. Dhritiman and Debasish – Feel free to include any other points that are indispensable for us to know.