



# TranSMART 17.1

## data loading workshop

Using Python and Jupyter Notebook

Alessia Peviani, Ewelina Grudzień, Gijs Kant

# Goal

- Experience useful Python tools for Data Science
    - Jupyter Notebook (interactive working environment)
    - Pandas (data science library)
  - Learn about TranSMART-specific tools
    - Data loading: TMTK & the Arborist, transmart-copy
    - API queries: TranSMART API Client
- > explore data interactively, while saving your processing steps and output
- > especially developed to visualize and process tabular data (“data frames”)
- 
- > a more powerful and user-friendly alternative to R?

# Program

- Setup (10-15 mins)
- Part 1: Loading data to tranSMART (45 mins)
  - Reading in and exporting the data using TMTK (15)
  - Correcting the data tree structure with the Arborist plugin (5)
  - Data loading to TranSMART using transmart-copy (5)
  - Data cleaning with Pandas (20)

Break (10-15 mins)

- Part 2: Querying the data in TranSMART (35 mins)
  - API calls to TranSMART using the Python API Client
- Wrap-up (5 mins)

Here to help:

Alessia  
Ewelina  
Gijs

# Setup

How to access the workshop environment

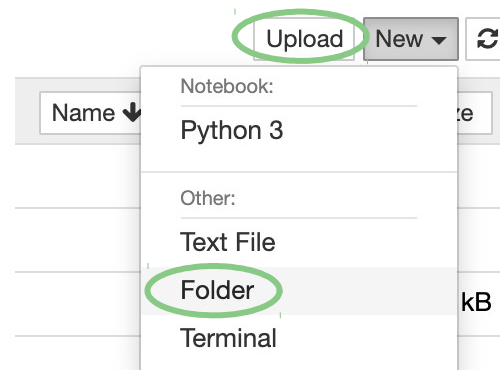
# Setup

- Get data from:  
<https://github.com/thehyve/workshop-tuebingen2019>
- Log in to Jupyter Hub server (*login credentials on paper*)

<https://notebook<N>.tuebingen1.thehyve.net>

+ Create **input** folder

- ↑ Upload **data\_part1** files to **input** folder
- ↑ Upload the notebook **part1\_data\_loading.ipynb** to workspace
- ↑ Upload the notebook **part2\_API\_calls.ipynb** to workspace



# Setup

..Are we all here?

Files

Running

Clusters

Select items to perform actions on them.

Upload

New ▾

↺

<input type="checkbox"/> 0 ▾	📁 /	Name ▾	Last Modified	File size
<input type="checkbox"/>	📁 input		32 minutes ago	
<input type="checkbox"/>	📄 part1_data_loading.ipynb		6 minutes ago	19.1 kB
<input type="checkbox"/>	📄 part2_API_calls.ipynb		seconds ago	125 kB

# Part 1: Data Loading to TranSMART

Tools: TMTK & the Arborist, transmart-copy

# 1.1 Data import with TMTK

TMTK allows data processing using a template file:

- Sheet “Tree structure” must be filled in
- Level 1 must always contain study name (you can change it later though)
- Sheets “Modifier”, “Trial visit”, “Ontology mapping”, “Value substitution” need to be present, but can be empty
- Optionally, you can add sheets with clinical data (e.g. “patient data”, “tumor data”, “lab data”, “survival data”; otherwise data can be stored in separate files (excel / csv / tsv)
- Data files/sheets must contain a SUBJ\_ID column



# 1.1 Data import with TMTK

## EXERCISE 1

- ✓ Have a look at `template_file.xlsx` in the input folder
- ✓ Try to import the template file using TMTK; you will get 2 errors
- ✓ Find and correct errors in the tree structure file using previous TMTK messages for guidance

# 1.1 Data import with TMTK

## **SOLUTION**











### Errors

- "Tree structure" references csv file, but not found -> correct file extension
- "Tree structure" references column, but not found -> correct column name

# 1.1 Data import with TMTK

## EXERCISE 2

- ✓ The data file lab\_data.xlsx contains an extra field with relevant experiment results; add this missing variable to the Tree structure, so that it is nested at the same level of "CEA (blood/serum)"

- ▲  Clinical characteristics
  -  Tags
  - ▶  1. Subjects
  - ▶  2. Primary tumor
  - ▶  3. Metastasis
  - ▶  4. Treatments
  - ▲  5. Lab results
    - ▶  LDH (serum)
    - ▶ **123**  CEA (blood/serum)
  - ▶  6. Endpoints

# 1.1 Data import with TMTK

## SOLUTION

Column name	Level 3	Level 3 metadata	Level 3 metadata value	Level 4
Gender	1. Subjects			Gender
Age				Age
History of colon polyps				History of colon polyps
Hypermutated	2. Primary tumor			Hypermutated
Location tumor				Location tumor
T stage				Staging
M stage				
Location primary tumor	3. Metastasis			Location
Number of affected organs				Number of affected organs
Previous adjuvant therapy	4. Treatments			Previous adjuvant therapy
Radiotherapy				Radiotherapy
Treatment arm				Treatment arm
LDH (serum)	5. Lab results			LDH (serum)
CEA (blood/serum)				CEA (blood/serum)
Cause of death	6. Endpoints			Cause of death
Overall survival event				Overall survival
Overall survival				

# 1.2 Tree structure editing with the Arborist

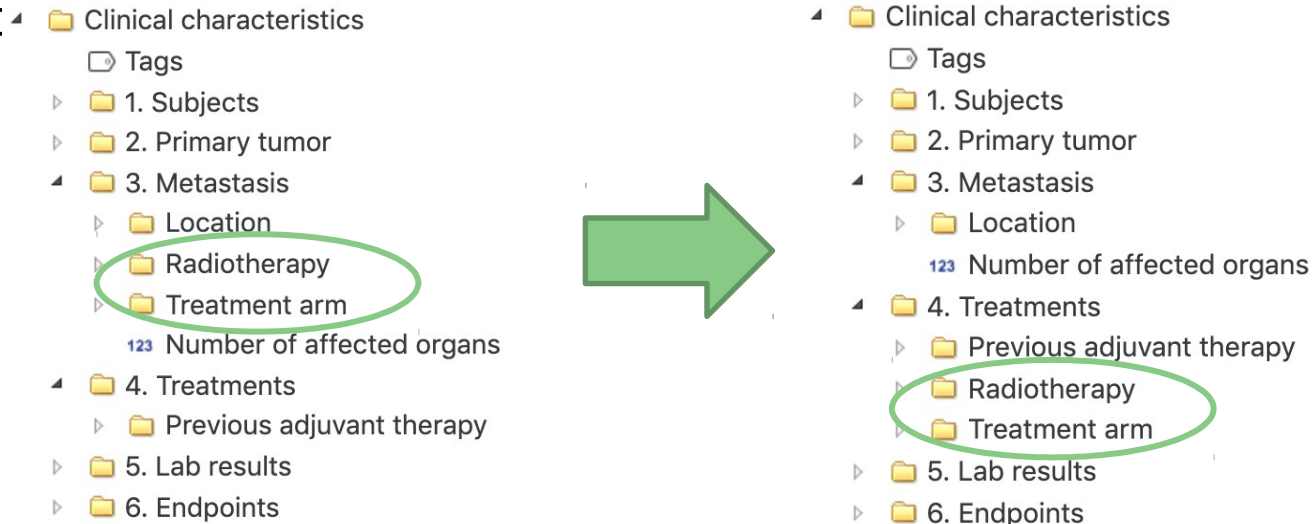
WHAT THE ARBORIST DOES..

# 1.2 Tree structure editing with the Arborist

## EXERCISE 3

Oops, we nested some concepts in the wrong place in the Tree structure!

Start the Arborist and drag&drop the tree nodes to where they belong



# 1.3 Export the data

## HOW TO EXPORT IN TRANSMART COPY FORMAT

- ✓ Export study to “output” folder using TMTK export commands
- ✓ The data is now structured into the folders i2b2demodata & i2b2metadata, containing the corresponding TranSMART database schemas (TSV tables)

# 1.4 Load to TranSMART using transmart-copy

From Jupyter Notebook (see code there),  
Or from the terminal:



```
export PGUSER=biomart_user
export PGPASSWORD=biomart_user
export PGHOST=transmart-database
export PGPORT=5432
```

```
java -jar /transmart-copy.jar -d output
```

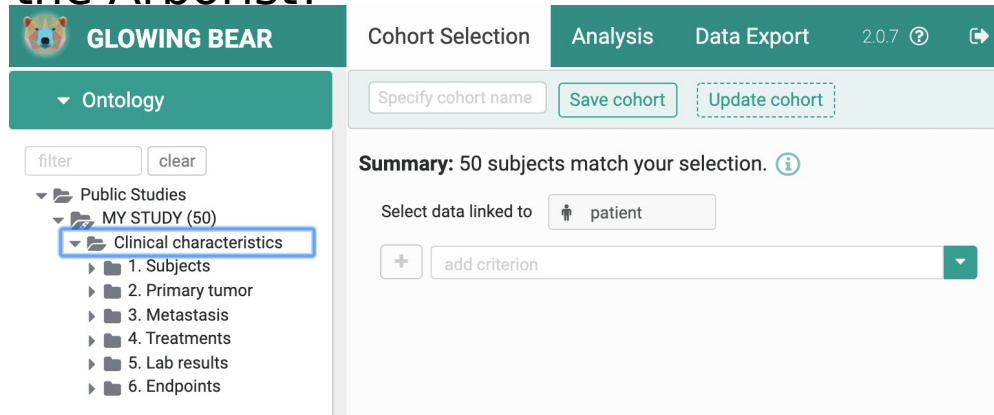


# 1.5 Check the uploaded data in Glowing Bear

- ✓ Refresh the cached results (see code in the notebook)
- ✓ Check the data in Glowing Bear! (see *login instructions on paper*)

<https://gb<N>.tuebingen<M>.thehyve.net>

- ✓ Do you recognize the tree structure you saw in the template file and the Arborist?



# 1.5 Data cleaning using Pandas

In the folder there are 2 other data files which were not included in the tree structure, and therefore were not uploaded.

We want to combine these files into a single one and perform some data cleaning because of issues with the information contained in the files.

Additionally, you can add the merged file to the original template file, and upload the data to TranSMART following the same steps we just performed.

# 1.5 Data cleaning with Pandas

## EXERCISE 4A

- ✓ Read in files (one from csv / one from excel)
- ✓ Join dataframes based on “Experiment code” column
  - ✓ Ensure codes (XXXX.XX) are read in as strings, not numbers (leading zero will be lost otherwise)
  - ✓ Change code formatting so that it matches between dataframes (add “C” letter in one)
- ✓ Add new (derived) values
  - ✓ Column X is only partially filled. Calculate the value as the mean of column Y1-3
- ✓ Replace values
  - ✓ Replace NaN values in column Y with 0s
  - ✓ Replace value X in column Y with value Z, but only if value X2 in column Y2 is Z2
- ✓ Remove values
  - ✓ Some experiments have value X in column Y, which means they should be excluded. Remove those rows entirely
- ✓ Write the merged dataframe to disk as an Excel file

# Break

We start again at 15:50!

# Part 2: Querying TranSMART via API calls

Tools: Python API Client

## 2.1 Why using an API Client?

You could spend 2 minutes to show a simple query in GB, and then say you could perform that same query from Jupyter Notebook, much more convenient if you want to work directly with the data (perform analyses etc) = programmatic access

## 2.2 Connecting to TranSMART

NOTE: Not the same instance on which data where loaded  
But feel free to try that after you get this one right!

## 2.3 Explore data in TranSMART

General calls to show tree structure, available studies etc..



## 2.4 Create queries with specific constraints

1. Show **simple constraint example**,  
then ask to create another one to answer a different question
2. Show **complex constraint example**,  
then again ask a different question

-> examples with both patients / observations?

-> examples with path or concept code? Would be nice to ask to look up data using previous commands to find path/concept for something asked in exercise..

# Wrap-up

Questions and links to additional material

# Wrap-up

## Links

- Download this presentation at [some-github-repo-link!](#)
- Jupyter Notebook and Pandas:
  - <https://www.anaconda.com/distribution/> (download both as part of Anaconda distribution)
  - <https://www.datacamp.com/community/tutorials/tutorial-jupyter-notebook> (Jupyter Notebook tutorial)
  - <https://pandas.pydata.org/pandas-docs/stable/10min.html> (Pandas tutorial)
- Tmtk & the Arborist plugin:
  - <https://github.com/thehyve/tmtk/>
- TranSMART API client:
  - <https://github.com/thehyve/transmart-api-client-py>
- transmart-copy:
  - <https://github.com/thehyve/transmart-core/tree/dev/transmart-copy>

For more information on tranSMART 17.X and our tools, contact us at  
<https://thehyve.nl/about-us/contact/>

# Wrap-up

## Questions

- ..About Jupyter Notebook / Pandas?
- ..About data loading tools / API client?
- General questions?
- Suggestions for additional tools / features?



We empower scientists by building on open source software