ImmPort Data Resources and Tutorials

Webinar August 2022

The purpose of this webinar is to introduce the types of data available for download, tools that can be used to download the data and example tutorials on preparing and analyzing the downloaded data.

The topics covered are:

- General Information
  - Funding
  - Help -ImmPort_Helpdesk@immport.org
  - Registration - https://immport-user-admin.niaid.nih.gov:8443/registrationuser/registration
  - Documentation - https://docs.immport.org
- Data Types
- Data Model
  - Interactive Overview - https://immport.org/shared/dataModel
  - Relational Model - https://docs.immport.org/developers/datamodel/study/
- DataBrowser UI - https://browser.immport.org/browser
- API Documentation Reference Material - https://docs.immport.org/apidocumentation/
- Online Tutorials JupyterHub - https://tutorials.immport.org
- Additional Tutorials - https://docs.immport.org/tutorials/additionaltutorials/
- Live Code - Time Permitting

General Information

Funding

ImmPort is funded by the NIH, NIAID and DAIT in support of the NIH mission to share data with the public. Data shared through ImmPort has been provided by NIH-funded programs, other research organizations and individual scientists ensuring these discoveries will be the foundation of future research.

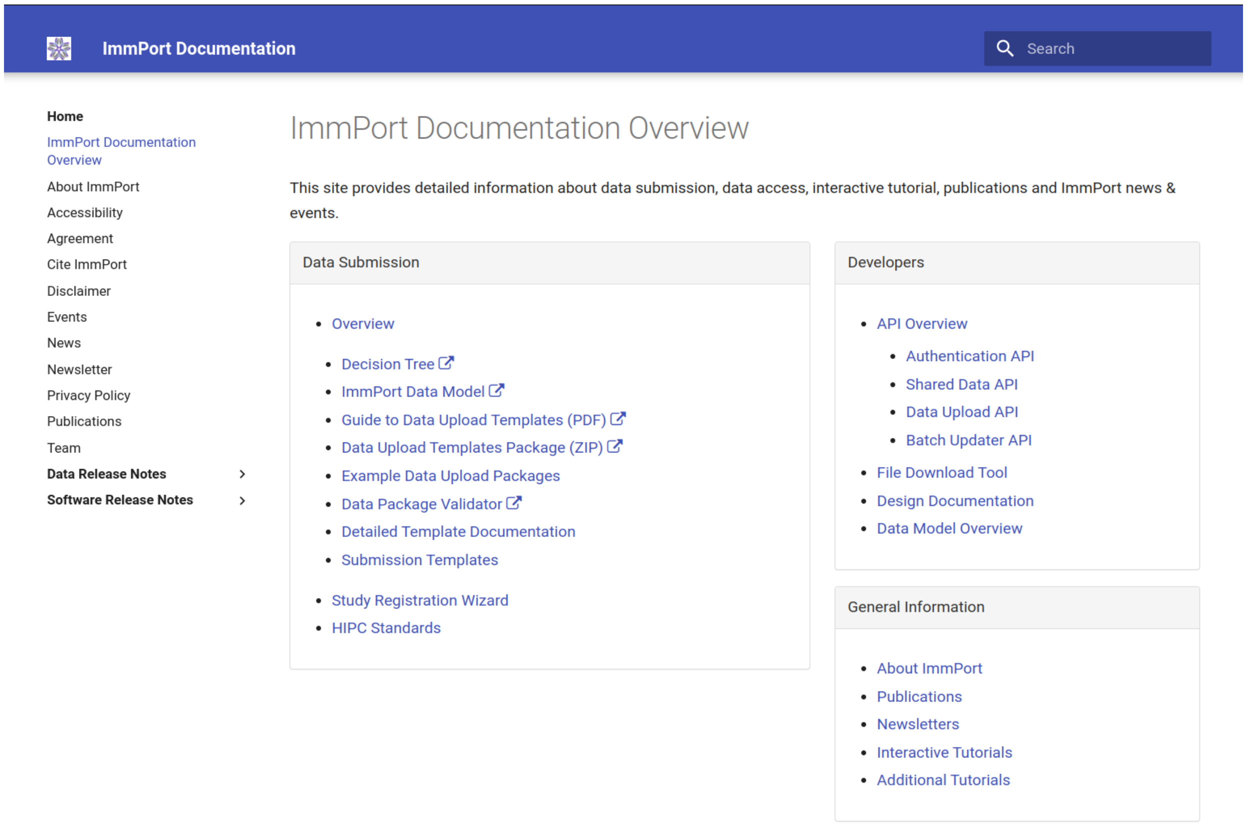
Help

If you have any questions about downloading data, the structure of the relational data model, example code, submitting data, please contact the ImmPort Help Desk. ImmPort_Helpdesk@immport.org

Registration

To download the data, you must be a registered user of ImmPort. Registration for use of ImmPort data is **Free**, here is the link to register.

Documentation - https://docs.immport.org/



ImmPort Documentation

Search

Home

ImmPort Documentation Overview

About ImmPort

Accessibility

Agreement

Cite ImmPort

Disclaimer

Events

News

Newsletter

Privacy Policy

Publications

Team

Data Release Notes

Software Release Notes

ImmPort Documentation Overview

This site provides detailed information about data submission, data access, interactive tutorial, publications and ImmPort news & events.

Data Submission

- Overview
  - Decision Tree
  - ImmPort Data Model
  - Guide to Data Upload Templates (PDF)
  - Data Upload Templates Package (ZIP)
  - Example Data Upload Packages
  - Data Package Validator
  - Detailed Template Documentation
  - Submission Templates
- Study Registration Wizard
- HIPC Standards

Developers

- API Overview
  - Authentication API
  - Shared Data API
  - Data Upload API
  - Batch Updater API
- File Download Tool
- Design Documentation
- Data Model Overview

General Information

- About ImmPort
- Publications
- Newsletters
- Interactive Tutorials
- Additional Tutorials

Data Types

Relational Table Data

The ImmPort application uses a relational database to store metadata for a study. We will take a closer look at the relational data model in the next section, because much of the data available for download via the UI or API's represent the contents of the relational tables. Examples of metadata include:

- Study - personnel, contract information, objectives, condition studied, links to paper and links to other repositories.
- ARM, Group - name, type
- Subject - age, gender, race, ethnicity, etc.
- Experiments - assay type, experiment samples

- Assay Results - for many of the standard immunology assay methods, ImmPort will load this results into structure tables.
    - Examples include: ELISA, ELISpot, HAI, etc.
- Assessments
- Lab Tests

Definition from ImmPort Model Paper: Toward an ontology-based framework for clinical research databases
Subject Assessment and Lab Test differ in that lab tests involves a specimen (biosample) as input and frequently utilize reagents for measurement purposes (substance or compound chemicals) whereas assessments involve a subject participant as input and do not utilize reagents. Physical Exam, Medical History, Family History and Questionnaire are examples of different types of assessments

Study Files

Many studies include supplementary files that can be used for future analysis. When possible and time permits the ImmPort curation team will parse the contents of these files into the proper relational tables.

Example Study File types include: Case Report Form, Demographics, Interventions, Data Dictionary, etc.

Protocols

Every study available in ImmPort has an associated protocol document, that should provide additional information useful for evaluating a study.

Result Files

These files are used to provide raw or summarized data from experimental assays. Examples file types are: Flow Cytometry FCS, Illumina BeadArray, ELISA, ELISpot, MBAA, PCR, etc.

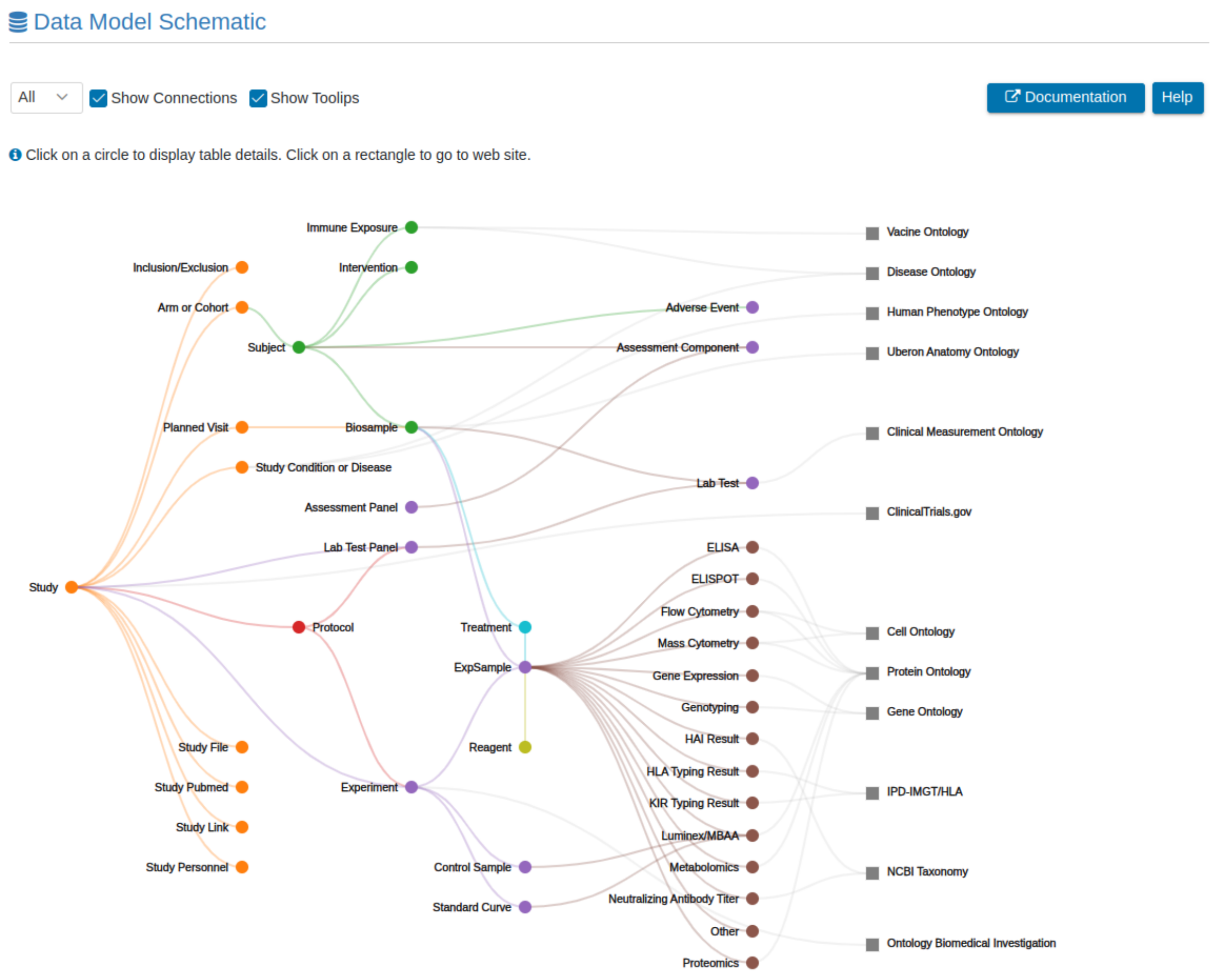
Links to other Repositories

In many cases, Immport has links to primary data stored in data repositories, Like Geo, SRA, dbSNP etc.

Example of Multiple Data Types: DataBrowser- SDY1

Data Model

Interactive Overview - https://immport.org/shared/dataModel



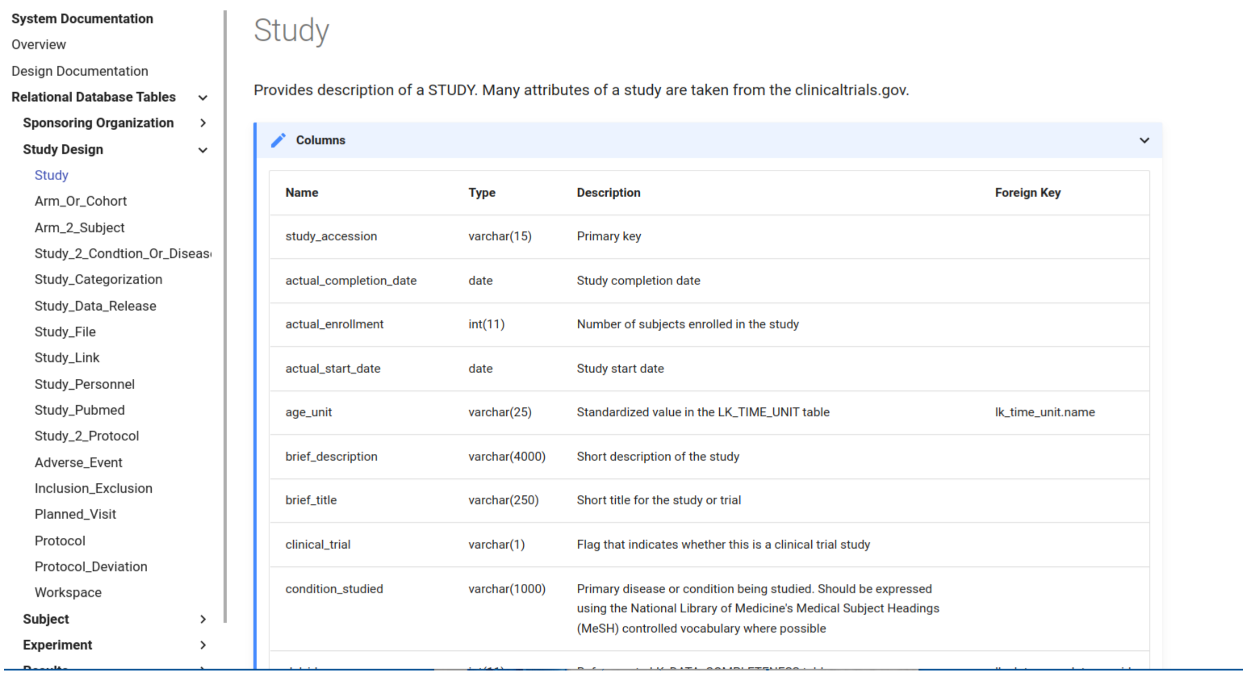
Overview of Study metadata tables:

1. STUDY - is the central object in the ImmPort data model. Ancillary tables like study_personnel, study_pubmed, study_links, protocols, etc. contain additional information about a study.
2. ARM_OR_COHORT (Group) a group of individuals who share a characteristic at some specific time and who are then followed forward in time, with data being collected at one or more suitable intervals.
3. ARM_2_SUBJECT - The ImmPort data model does not use a one-to-one relationship to a Study, because a Subject may participate in multiple studies. This is particularly immportant in the example where a subject participates in multiple studies across several years, so the age of a subject can vary from study to study.
4. SUBJECT - contains basic demographic information, gender, race, species, etc.
5. PLANNED_VISIT - This table represents the order and timing of patient encouters. Examples: Day 1, Day 14 and Day 28
6. BIOSAMPLE - Material extracted from a subject during a PLANNED VISIT. Examples: Subject had Blood Drawn on Day 1, a Nasal Swab on Day 14, etc.

7. ASSESSMENT - Subject level measurements. Types of questions that might be asked in a Case Report Form. Examples: Smoking Status, Marital Status, etc.
8. LABTEST - Biosample level measurents. Types of measurements returned by a Blood Panel, Metabolic Panel on a specific Planned Visit
9. EXPERIMENT - Types of assays performed on a Biosample. For example ELISA was run on blood samples for Day 1 and Day 28.
10. EXPSAMPLE - This table was added to the model in case where was some tranformation applied to the Biosample before it was assayed by an Experiment.

The square boxes on the far right of the schematic represent links to ontologies and controlled vocabularies used to standarize and harmonize data.

Relational Model - https://docs.immport.org/developers/datamodel/study/

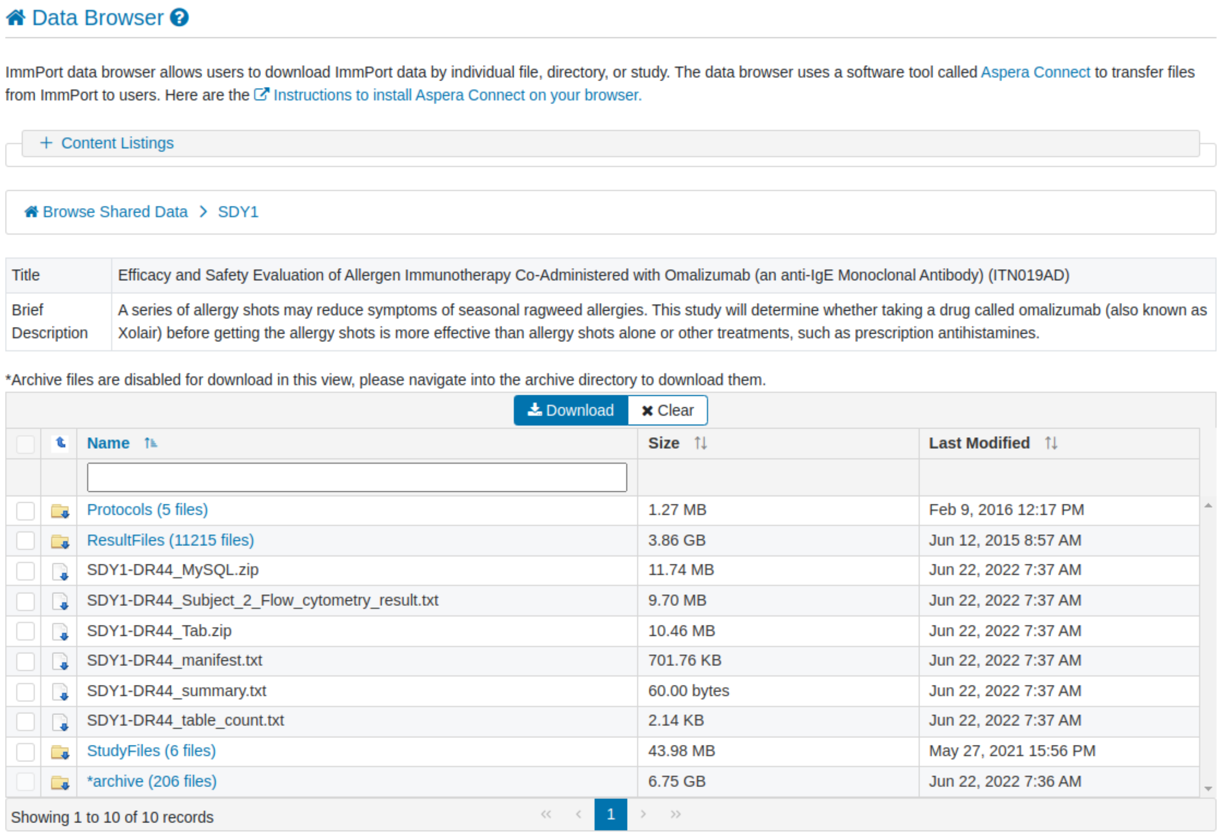


Study

Provides description of a STUDY. Many attributes of a study are taken from the clinicaltrials.gov.

Name	Type	Description	Foreign Key
study_accession	varchar(15)	Primary key	
actual_completion_date	date	Study completion date	
actual_enrollment	int(11)	Number of subjects enrolled in the study	
actual_start_date	date	Study start date	
age_unit	varchar(25)	Standardized value in the LK_TIME_UNIT table	lk_time_unit.name
brief_description	varchar(4000)	Short description of the study	
brief_title	varchar(250)	Short title for the study or trial	
clinical_trial	varchar(1)	Flag that indicates whether this is a clinical trial study	
condition_studied	varchar(1000)	Primary disease or condition being studied. Should be expressed using the National Library of Medicine's Medical Subject Headings (MeSH) controlled vocabulary where possible	

Data Browser - UI for Downloading Data - https://browser.immport.org/browser



Data Browser

ImmPort data browser allows users to download ImmPort data by individual file, directory, or study. The data browser uses a software tool called Aspera Connect to transfer files from ImmPort to users. Here are the Instructions to install Aspera Connect on your browser.

+ Content Listings

Browse Shared Data > SDY1

Title	Efficacy and Safety Evaluation of Allergen Immunotherapy Co-Administered with Omalizumab (an anti-IgE Monoclonal Antibody) (ITN019AD)
Brief Description	A series of allergy shots may reduce symptoms of seasonal ragweed allergies. This study will determine whether taking a drug called omalizumab (also known as Xolair) before getting the allergy shots is more effective than allergy shots alone or other treatments, such as prescription antihistamines.

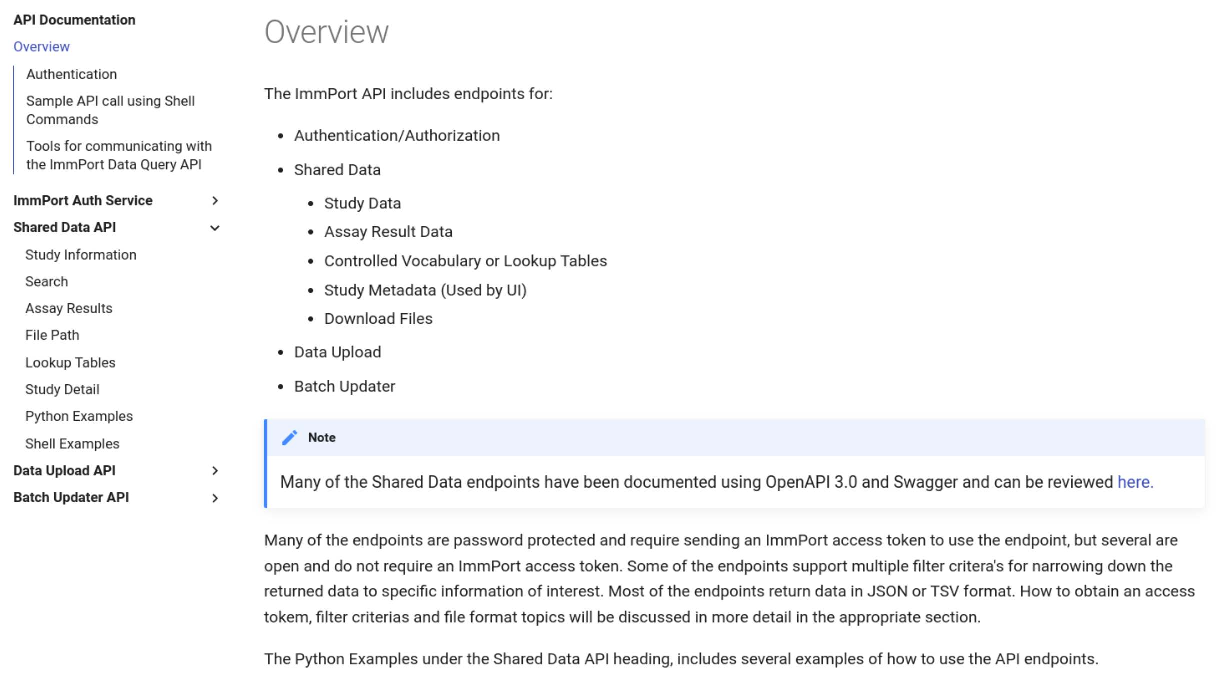
*Archive files are disabled for download in this view, please navigate into the archive directory to download them.

Download Clear

Name	Size	Last Modified
Protocols (5 files)	1.27 MB	Feb 9, 2016 12:17 PM
ResultFiles (11215 files)	3.86 GB	Jun 12, 2015 8:57 AM
SDY1-DR44_MySQL.zip	11.74 MB	Jun 22, 2022 7:37 AM
SDY1-DR44_Subject_2_Flow_cytometry_result.txt	9.70 MB	Jun 22, 2022 7:37 AM
SDY1-DR44_Tab.zip	10.46 MB	Jun 22, 2022 7:37 AM
SDY1-DR44_manifest.txt	701.76 KB	Jun 22, 2022 7:37 AM
SDY1-DR44_summary.txt	60.00 bytes	Jun 22, 2022 7:37 AM
SDY1-DR44_table_count.txt	2.14 KB	Jun 22, 2022 7:37 AM
StudyFiles (6 files)	43.98 MB	May 27, 2021 15:56 PM
*archive (206 files)	6.75 GB	Jun 22, 2022 7:36 AM

Showing 1 to 10 of 10 records

API Documentation - https://docs.immport.org/apidocumentation/



Overview

The ImmPort API includes endpoints for:

- Authentication/Authorization
- Shared Data
  - Study Data
  - Assay Result Data
  - Controlled Vocabulary or Lookup Tables
  - Study Metadata (Used by UI)
  - Download Files
- Data Upload
- Batch Updater

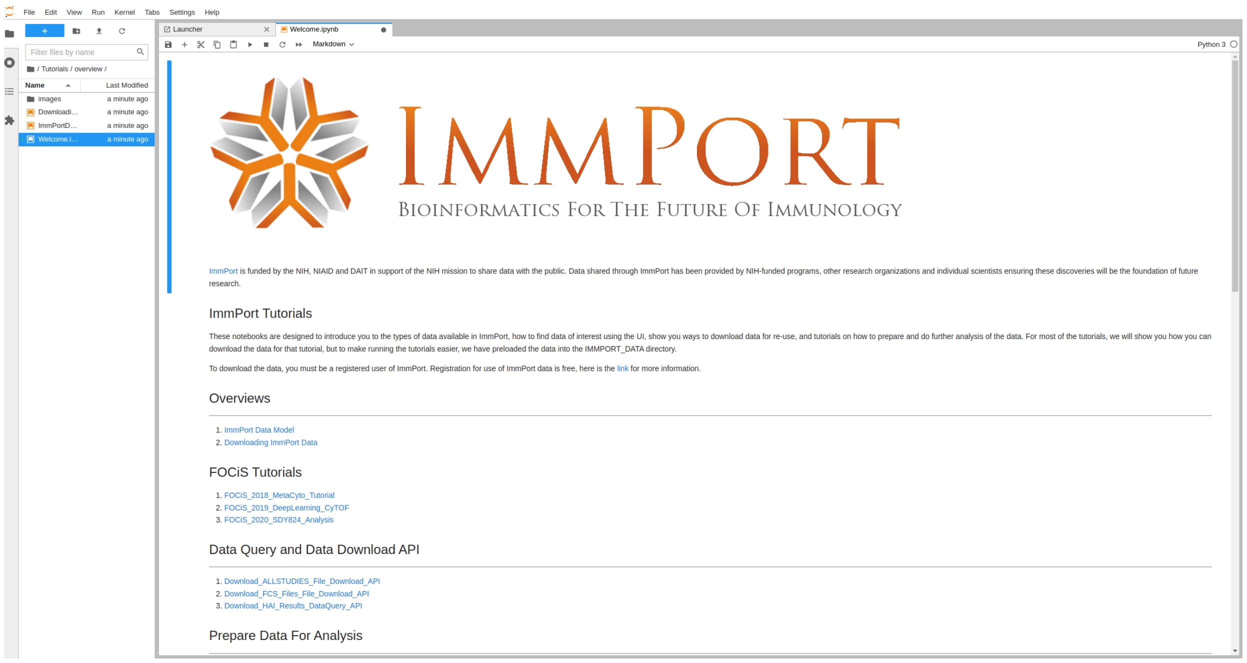
Note

Many of the Shared Data endpoints have been documented using OpenAPI 3.0 and Swagger and can be reviewed here.

Many of the endpoints are password protected and require sending an ImmPort access token to use the endpoint, but several are open and do not require an ImmPort access token. Some of the endpoints support multiple filter critera's for narrowing down the returned data to specific information of interest. Most of the endpoints return data in JSON or TSV format. How to obtain an access tokem, filter criterias and file format topics will be discussed in more detail in the appropriate section.

The Python Examples under the Shared Data API heading, includes several examples of how to use the API endpoints.

Online Tutorials JupyterHub - https://tutorials.immport.org



ImmPort is funded by the NIH, NIAID and DAIT in support of the NIH mission to share data with the public. Data shared through ImmPort has been provided by NIH-funded programs, other research organizations and individual scientists ensuring these discoveries will be the foundation of future research.

ImmPort Tutorials

These notebooks are designed to introduce you to the types of data available in ImmPort, how to find data of interest using the UI, show you ways to download data for re-use, and tutorials on how to prepare and do further analysis of the data. For most of the tutorials, we will show you how you can download the data for that tutorial, but to make running the tutorials easier, we have preloaded the data into the IMMPORT_DATA directory.

To download the data, you must be a registered user of ImmPort. Registration for use of ImmPort data is free, here is the link for more information.

Overviews

1. ImmPort Data Model
2. Downloading ImmPort Data

FOCiS Tutorials

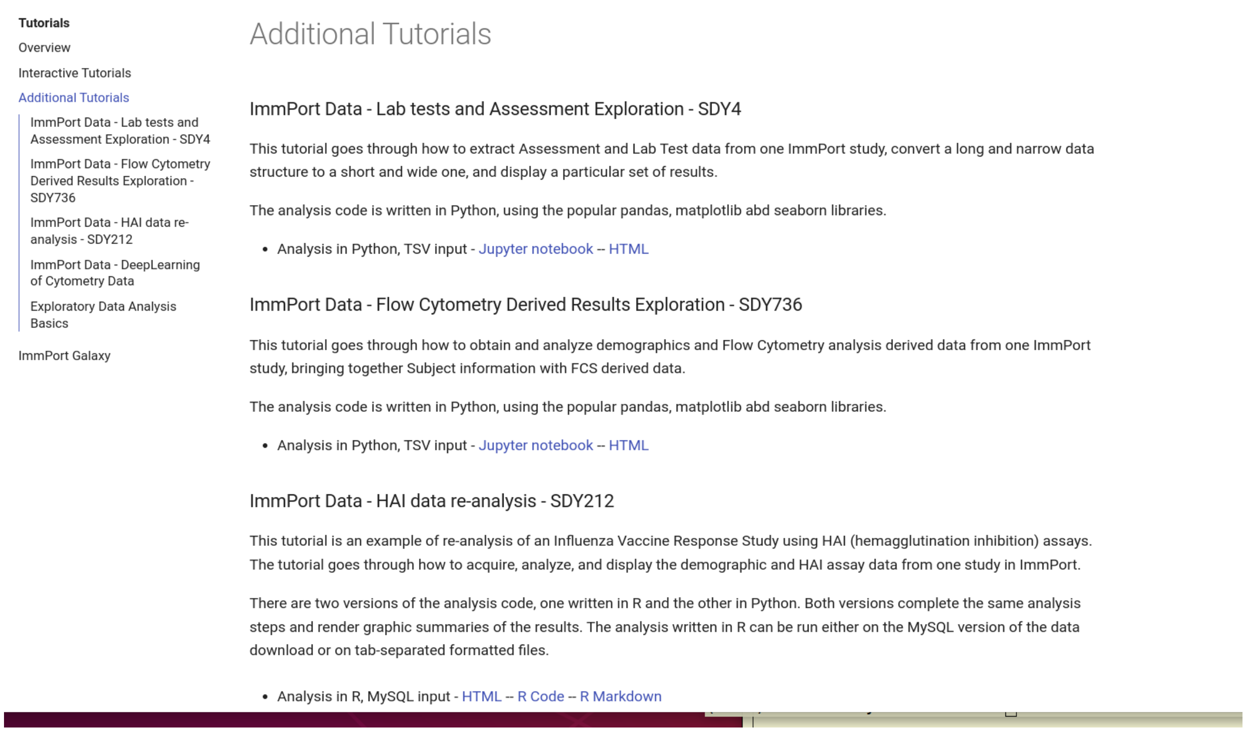
1. FOCiS_2018_MetaCyto_Tutorial
2. FOCiS_2019_DeepLearning_CyTOF
3. FOCiS_2020_SDY824_Analysis

Data Query and Data Download API

1. Download_ALLSTUDIES_File_Download_API
2. Download_FCS_Files_File_Download_API
3. Download_HAI_Results_DataQuery_API

Prepare Data For Analysis

Additional Tutorials - https://docs.immport.org/tutorials/additionaltutorial


Tutorials
Overview
Interactive Tutorials
Additional Tutorials
ImmPort Data - Lab tests and Assessment Exploration - SDY4
ImmPort Data - Flow Cytometry Derived Results Exploration - SDY736
ImmPort Data - HAI data re-analysis - SDY212
ImmPort Data - DeepLearning of Cytometry Data
Exploratory Data Analysis Basics
ImmPort Galaxy

Additional Tutorials

ImmPort Data - Lab tests and Assessment Exploration - SDY4

This tutorial goes through how to extract Assessment and Lab Test data from one ImmPort study, convert a long and narrow data structure to a short and wide one, and display a particular set of results.

The analysis code is written in Python, using the popular pandas, matplotlib abd seaborn libraries.

- Analysis in Python, TSV input - Jupyter notebook -- HTML

ImmPort Data - Flow Cytometry Derived Results Exploration - SDY736

This tutorial goes through how to obtain and analyze demographics and Flow Cytometry analysis derived data from one ImmPort study, bringing together Subject information with FCS derived data.

The analysis code is written in Python, using the popular pandas, matplotlib abd seaborn libraries.

- Analysis in Python, TSV input - Jupyter notebook -- HTML

ImmPort Data - HAI data re-analysis - SDY212

This tutorial is an example of re-analysis of an Influenza Vaccine Response Study using HAI (hemagglutination inhibition) assays. The tutorial goes through how to acquire, analyze, and display the demographic and HAI assay data from one study in ImmPort.

There are two versions of the analysis code, one written in R and the other in Python. Both versions complete the same analysis steps and render graphic summaries of the results. The analysis written in R can be run either on the MySQL version of the data download or on tab-separated formatted files.

- Analysis in R, MySQL input - HTML -- R Code -- R Markdown

Live Code

In this section we will demonstrate real time code execution, which will use some library functions already written in Python packages. The Cell below will import these packages into working enviroment. These packages were written as examples of how you might want to develop your own packages to factor out common methods.

One nice feature is you can always review the source code by entering ??api or ??ui in a cell.

In [1]:

```
import sys
import os
import json
import requests
import pandas as pd

from pathlib import Path
from dotenv import load_dotenv

sys.path.append('../common')
import immport_api as api
import immport_util as iu
```

In [2]:

```
env_path = Path('../common')/'.env'
load_dotenv("../common/.env")
OUTPUT_DIR = os.environ.get("OUTPUT_DIR")
API_ENDPOINT_BASE_URL = os.environ.get("API_ENDPOINT_BASE_URL")
USER_NAME = os.environ.get("USER_NAME")
USER_PASSWORD = os.environ.get("USER_PASSWORD")
```

In [3]:

```
##??api
```

Request a Token

Many of the API endpoints require an Access Token to download data. The first example uses a Python program to obtain a token, then the next example shows how to use a Shell script to obtain a token.

Python

This is an example of using the api.request_immport_token method in the immport_api package.

In [4]:

```
access_token = api.request_immport_token(USER_NAME, USER_PASSWORD)
print(access_token)
```

```
eyJraWQiOiIwMDYwMGI0Zi02NjI3LTQxMjYtOTZkOC1jNGU4N2E5NDE3MGQiLCJhbGciOiJSUzI1NiJ
9.eyJzdWIiOiJjYW1wam8iLCJhdWQiOiJpbW1wb3J0LWF1dGgtdG9rZW4tY2xpZW50IiwibmJmIjoxNj
YxODc2OTExLCJzY29wZSI6WyJkb3dubG9hZCIsIm9wZW5pZCIsImJyb3dzZSJdLCJpc3MiOiJodHRwcz
pcL1wvd3d3LmltbXBvcnQub3JnXC9hdXRoIiwiZXhwIjoxNjYxOTYzMzExLCJpYXQiOjE2NjE4NzY5MT
EsImF1dGhvcml0aWVzIjpbIlJPTEVfVVNFUiIsIlJPTEVfQ1VSQVRJT05fVVNFUiIsIlJPTEVfQURNSU
4iLCJST0xFX0lNTVBPUlRfREFUQV9NQU5BR0VNRU5UX0FETUlOIiwiUk9MRV9VU0VSX0FETUlOSVNUUk
```

```
FUSU9OX0FETUlOIiwiUk9MRV9TSEFSSU5HX1VTRVIiLCJST0xFX0RBVEFfQlJPV1NFUl9BRE1JTiJdf
Q.ielki8bux1efA-zLAmNHT5QHdhePB7rkdBs0a4Fa6BNOhUBu9J6iCRGEmsFK1hZI0c3c1eMBh9xq10
9VW6_HEYcEh49FjB6_B6efckekSI3x1CT0l0xqOEvdTARbQj72GNVAZhZHcPV4cnpTRGb6A6aij6fSC1
aM940EwgD24IH67lbQWJR8YqYv27LkFQa0nLRndO4GWkRR8I0aYlbms7WrfR8bM6PHgOx_m-OUOIOe1c
D41YGSTXEbLn-ZjbjKOd1mjdy3NvmzIH16LC-TVE2uP9NYk8H1vczU_QMBQbxhj8CZv_bSrohe7LuEAL
vMhwhcSgu2UWF5YCJR1O15mQ
```

In [5]:

```
### Shell Command
```

In [6]:

```
%%bash
export token=`curl -X POST https://www.immport.org/auth/token -d username="$USEF
echo $token
```

```
eyJraWQiOiIwMDYwMGI0Zi02NjI3LTQxMjYtOTZkOC1jNGU4N2E5NDE3MGQiLCJhbGciOiJSUzI1NiJ
9.eyJzdWIiOiJjYW1wam8iLCJhdWQiOiJpbW1wb3J0LWF1dGgtdG9rZW4tY2xpZW50IiwibmJmIjoxNj
YxODc2OTEyLCJzY29wZSI6WyJkb3dubG9hZCIsIm9wZW5pZCIsImJyb3dzZSJdLCJpc3MiOiJodHRwcz
pcL1wvd3d3LmltbXBvcnQub3JnXC9hdXRoIiwiZXhwIjoxNjYxOTYzMzEyLCJpYXQiOjE2NjE4NzY5MT
IsImF1dGhvcml0aWVzIjpbIlJPTEVfVVNFUiIsIlJPTEVfQ1VSQVRJT05fVVNFUiIsIlJPTEVfQURNSU
4iLCJST0xFX0lNTVBPUlRfREFUQV9NQU5BR0VNRU5UX0FETUlOIiwiUk9MRV9VU0VSX0FETUlOSVNUUk
FUSU9OX0FETUlOIiwiUk9MRV9TSEFSSU5HX1VTRVIiLCJST0xFX0RBVEFfQlJPV1NFUl9BRE1JTiJdf
Q.PxhtCMoFbA2xC_iOJ6kwSUhuUvn5f957QEhZy46wWK5OQh4zZhtD29Yl0s5bYQN4_Cu0MrGeOz-Cfv
48csLOfPQ6NiiI67_ePwZIB6h4wr4hzLiJdxQxIhCCYDssGBNPgsqTVE54jOb_jnNyvLw6iOLLiLOJZM
5sb_qz-wzHwYBmlGE_tJpetCVcq_P6VHs9mUw7dVFuLCujWZNo5-QsvgheLO1U6Ao9_kST3FuYlkjQmy
PbJSAEQWXvM7o2q0L8qbVuYbPU5Z9hy7uxqLMb3a2aEhMvvutFPvet0zWCF5vGVzf-EfeTQnCASQyRic
8b2pNgE4SfYO8q7evP07iGtA
```

Download Study Package

A study package contains all the metadata contained in the relational database for one study. The ALLSTUDIES package is also available which contains all the metadata for all studies.

Python

This example will use the api.download_file method in the immport_api package. If you review the source code, you will see this method is a wrapper around the api.request_immport_token and the api.request_aspera_token and the api.retrieve_file methods.

NOTE: Aspera is software from IBM that is used by several NIH repositories and companies to improve file transfer speed. The client software is free and is included in the ImmPort Download Tool zip package available. More information about Aspera and the Download tool is available here

In [7]:

```
##??api.download_file
```

In [8]:

```
# Using API download the SDY1_Tab file from ImmPort
output_directory = OUTPUT_DIR + "/SDY1"

# Remove the directory in case it exists
# This is commented out, in case you do not want to execute this step.
# Be careful using /bin/rm -rf
# !/bin/rm -rf output_directory

%mkdir $output_directory
```

```
#
# The file_path represent to directory structure visible from the DataBrowser
#
file_path = '/SDY1/SDY1-DR44_Tab.zip'

# The process for downloading a file takes 3 steps, which are below:
# immport_token = api.request_immport_token(user_name, user_password)
# aspera_token = api.request_aspera_token(file_name, immport_token)
# api.retrieve_file(file_name, "./downloads-api", aspera_token)
# To make it easier, there is a download_file method, the encompases the 3 steps

%time status = api.download_file(USER_NAME, USER_PASSWORD, file_path, output_di
```

```
Completed: 10708K bytes transferred in 7 seconds
 (12086K bits/sec), in 1 file.
CPU times: user 103 ms, sys: 5.98 ms, total: 109 ms
Wall time: 9.13 s
```

In [9]:

```
# Change to the downloads directory, then quietly unzip the file, and finally mo
# text files up to the SDY1 directory. If you run into problems with this step r
# from the unzip command.

%time !cd $output_directory; unzip --qq SDY1-DR44_Tab.zip; mv SDY1-DR44_Tab/Tab/
##%time !ls $output_directory
```

```
CPU times: user 9.12 ms, sys: 5.45 ms, total: 14.6 ms
Wall time: 661 ms
```

Download Subject Demographics Using API

In this section we will demostrate using the api.api_data_query method which is a wrapper around the api.request_token and the request.get methods.

In [10]:

```
##??api.api_data_query
```

In [11]:

```
pd.set_option('display.max_columns', None)
API_ENDPOINT_BASE_URL = "https://www.immport.org"
DATA_QUERY_URL = API_ENDPOINT_BASE_URL + "/data/query"
ASPERA_TOKEN_URL = API_ENDPOINT_BASE_URL + "/data/download/token"
IMMPORT_TOKEN_URL = "https://www.immport.org/auth/token"

#endpoint = DATA_QUERY_URL + "/api/study/pubmed/SDY1?format=json"
endpoint = DATA_QUERY_URL + "/api/study/demographic/SDY1?format=json"
results = api.api_data_query(USER_NAME, USER_PASSWORD, endpoint, IMMPORT_TOKEN_U
#print(results)
df = pd.read_json(json.dumps(results))
df.head(2)
```

Out[11]:

studyAccession	briefTitle	armAccession	armName	armDescription	armTypeReported	armT

	studyAccession	briefTitle	armAccession	armName	armDescription	armTypeReported	armT
0	SDY1	Efficacy and Safety Evaluation of Allergen Imm...	ARM1	Placebo Immunotherapy with placebo anti-IgE	Placebo omalizumab pre-treatment, placebo RIT,...	Placebo Comparator	Co
1	SDY1	Efficacy and Safety Evaluation of Allergen Imm...	ARM1	Placebo Immunotherapy with placebo anti-IgE	Placebo omalizumab pre-treatment, placebo RIT,...	Placebo Comparator	Co

Construct Subject Demographics from Text File

Using the SDY1 study package we download previously we will construct a similar data set to what we just downloaded using the **demographic** endpoint in the previous section.

When we load in the data, we will be restricting the columns from each table, to only the columns needed for the final DataFrame. When the data is read in from the text file, the usecols option in the pd.read_csv method is used to restrict the input to only the columns of interest. After the pd.read_csv method completes, we use how the columns were specified in the usecols array, to order the columns in the DataFrame.

In the case of the ARM_2_SUBJECT table, we will include the NAME column, but in many cases the name provided might not be informative, so you may need to read the protocol for that study, to determine the purpose for the arm.

In [12]:

```
SDY1_DIR = OUTPUT_DIR + "/SDY1"
study_cols = ['STUDY_ACCESSION', 'BRIEF_TITLE']
study = pd.read_csv(SDY1_DIR + "/study.txt", sep="\t", usecols=study_cols)[study
iu.describe_df(study, "Study")

arm_cols = ['STUDY_ACCESSION', 'ARM_ACCESSION', 'NAME']
arm = pd.read_csv(SDY1_DIR + "/arm_or_cohort.txt", sep="\t", usecols=arm_cols)[a
arm = arm.rename(columns={'NAME': "ARM_NAME"})
iu.describe_df(arm, "Arm_or_Cohort")

arm_2_subject_cols = ['ARM_ACCESSION', 'SUBJECT_ACCESSION', 'AGE_UNIT', 'MIN_SUB
arm_2_subject = pd.read_csv(SDY1_DIR + "/arm_2_subject.txt", usecols=arm_2_subje
iu.describe_df(arm_2_subject, "Arm_2_Subject")

subject_cols = ['SUBJECT_ACCESSION', 'GENDER', 'RACE', 'ETHNICITY', 'SPECIES']
subject = pd.read_csv(SDY1_DIR + "/subject.txt", usecols=subject_cols, sep="\t")
iu.describe_df(subject, "Subject")
```

```
Study contains 1 rows and 2 columns
Arm_or_Cohort contains 4 rows and 3 columns
Arm_2_Subject contains 159 rows and 5 columns
Subject contains 159 rows and 5 columns
```

Review Individual Study Frames

In [13]:
```
pd.options.display.max_colwidth = 200
display(study.head(3))
display(arm.head(3))
display(arm_2_subject.head(3))
display(subject.head(3))
```

	STUDY_ACCESSION	BRIEF_TITLE
0	SDY1	Efficacy and Safety Evaluation of Allergen Immunotherapy Co-Administered with Omalizumab (an anti-IgE Monoclonal Antibody) (ITN019AD)

	STUDY_ACCESSION	ARM_ACCESSION	ARM_NAME
0	SDY1	ARM1	Placebo Immunotherapy with placebo anti-IgE
1	SDY1	ARM2	Immunotherapy with placebo anti-IgE
2	SDY1	ARM3	Placebo Immunotherapy with anti-IgE

	ARM_ACCESSION	SUBJECT_ACCESSION	AGE_UNIT	MIN_SUBJECT_AGE	MAX_SUBJECT_AGE
0	ARM1	SUB73369	Years	49	49
1	ARM1	SUB73372	Years	43	43
2	ARM1	SUB73374	Years	43	43

	SUBJECT_ACCESSION	GENDER	RACE	ETHNICITY	SPECIES
0	SUB73366	Male	Asian	Not Hispanic or Latino	Homo sapiens
1	SUB73367	Female	White	Not Hispanic or Latino	Homo sapiens
2	SUB73368	Female	White	Not Hispanic or Latino	Homo sapiens

Merge Individual Study Frames

In this step we will use the pd.merge method to merge the information from the 4 tables into one DataFrame. In relational database terms, we will be using the key columns in each table to join to the other tables. We will also be using a practice called method chaining to simplify the structure of the code. For more examples of using Pandas method chaining and the pipe function, you can google "Pandas Method Chaining Pipe".

In the code below we start by merging study to arm_or_cohort, the result from this merge is then used to merge arm_2_subject, and finally subject.

In [14]:
```
subject_demographics = (
    study.merge(arm, left_on="STUDY_ACCESSION", right_on="STUDY_ACCESSION")
         .merge(arm_2_subject, left_on="ARM_ACCESSION", right_on="ARM_ACCESSION'
         .merge(subject, left_on="SUBJECT_ACCESSION", right_on="SUBJECT_ACCESSI(
)
iu.describe_df(subject_demographics, "Subject Demographics")
subject_demographics.head(2)
```

```
Subject Demographics contains 159 rows and 12 columns
```

Out[14]:

	STUDY_ACCESSION	BRIEF_TITLE	ARM_ACCESSION	ARM_NAME	SUBJECT_ACCESSION	AGI
0	SDY1	Efficacy and Safety Evaluation of Allergen Immunotherapy Co-Administered with Omalizumab (an anti-IgE Monoclonal Antibody) (ITN019AD)	ARM1	Placebo Immunotherapy with placebo anti-IgE	SUB73369	
1	SDY1	Efficacy and Safety Evaluation of Allergen Immunotherapy Co-Administered with Omalizumab (an anti-IgE Monoclonal Antibody) (ITN019AD)	ARM1	Placebo Immunotherapy with placebo anti-IgE	SUB73372	

In [ ]: