RImmPort: Enabling ready-for-analysis immunology research data

Ravi Shankar

2019-10-29

Contents

---

1 Introduction

Publicly available raw individual-level clinical trial data has created tremendous opportunity to evaluate new research hypotheses that were not originally formulated in the studies; by reanalyzing data from a study, by

performing cross analysis of multiple studies, or by combining study data with other public research datasets. But such analysis of disparate data presupposes a) uniform representation of clinical trial data using data standards, and b) easy access to such standard representations of clinical trial data in analytical environments. `ImmPort`, the Immunology Database and Analysis Portal (immport.niaid.nih.gov) system, warehouses clinical study data in all areas of immunology that is generated by scientific researchers supported by the National Institute of Allergy and Infectious Diseases (`NIAID`). Currently, more than 100 studies are publicly available in ImmPort. The `RImmPort` package simplifies access to ImmPort data for analysis, as the name implies, in the R statistical environment. It provides a standards-based interface to the ImmPort study data that is in a proprietary format.

2 Overview of ImmPort

`ImmPort`, the Immunology Database and Analysis Portal (immport.niaid.nih.gov) system, warehouses clinical trials data in all areas of immunology that is generated primarily by investigators funded by the US National Institute of Allergy and Infectious Diseases (NIAID). With more than 100 datasets now publicly available and hundreds of downloads per month, ImmPort is an important source for raw data and protocols from clinical trials, mechanistic studies, and novel methods for cellular and molecular measurements. The different types of study-related data that are found in ImmPort include:

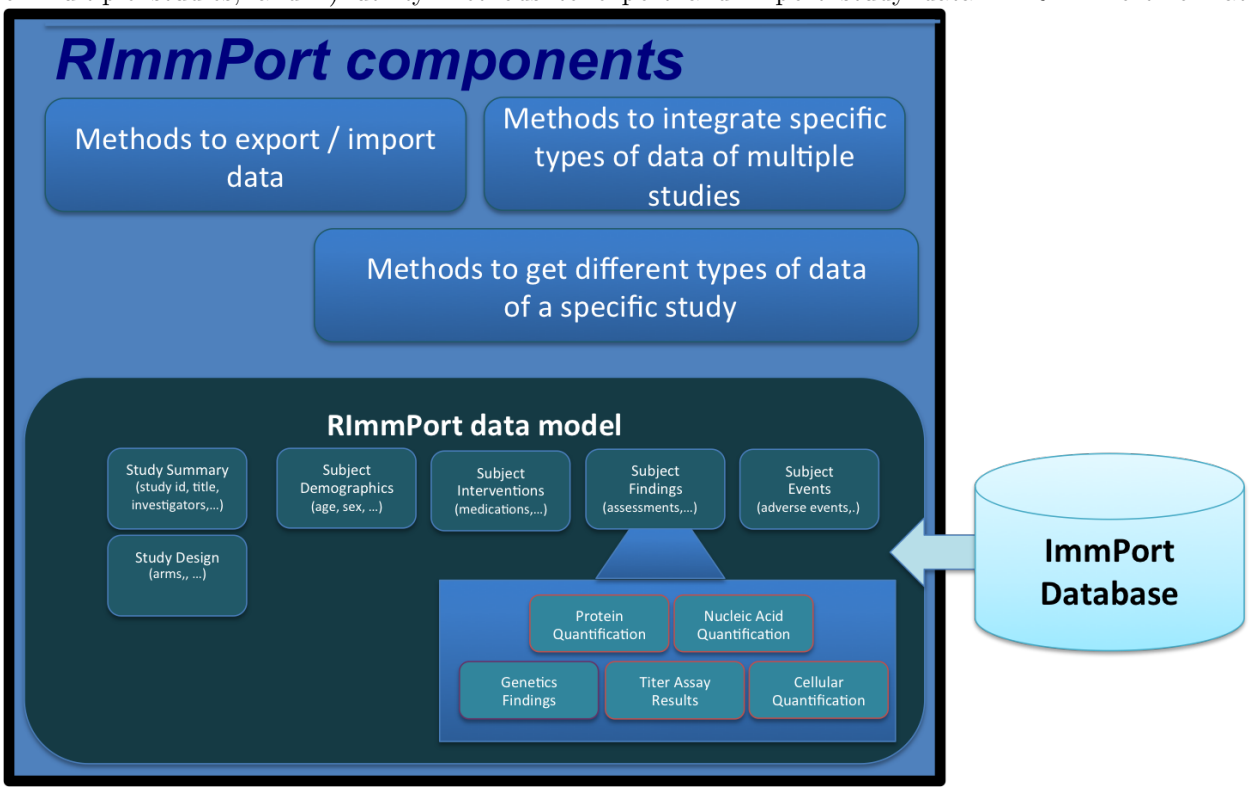
- Study Summary
- Arms
- Subject Demographics
- Assessments
- Concomitant Medications
- Study Interventions
- Laboratory Test Results
- Mechanistic Assay Results
- Adverse Events

The Open ImmPort provides interface to explore metadata of all studies that are publicly available, in order to locate the studies of interest. The ImmPort website houses all the study data. Users can download any of the data after registering at the ImmPort website (registration is free). The study data is available in 2 formats: MySQL format and Tab Separated Value (TSV) format. The RImmPort package works with both the formats. The schema and the entity-relationship diagrams document ImmPort's proprietary database model that is employed in the MySQL format of study data and is also reflected in the Tab format. Study data may be downloaded for each individual study, or the entire set of data for all studies may be downloaded as one file. In the case of MySQL format, users need to run a private instance of MySQL database, load the downloaded study data into that database instance, and then query the data for analysis. In the case of Tab format, users will read the tab-delimited files into their analytical environment and then perform data analysis.

3 Overview of RImmPort

`RImmPort` package simplifies access to ImmPort data for analysis, as the name implies, in the R statistical environment. Essentially,it provides a standards-based interface to the proprietary MySQL-formatted or Tab-formatted ImmPort data. It comprises of four main components: 1) an RImmPort data model that encapsulates ImmPort study data. The model leverages of study data standards from the Clinical Data Interchange Standards Consortium (CDISC), and incorporates terms and semantics found in these standards, 2) foundational methods to query and load different types of data of a specific study in ImmPort, 3) methods to search and integrate specific types of study data

of multiple studies, and 4) utility methods to export and import study data in RImmPort format.



The RImmPort workflow is illustrated in the following figure. User creates a private MySQL database instance, downloads study data of interest from ImmPort website, and loads the data into the private database. In R, user connects to the private ImmPort database, and calls the methods in the RImmPort library, to load a specific study from the database into R.

Using RImmPort, an entire study can be loaded into R with a single command. For example, a researcher interested in analyzing a specific study ImmPort Identifier:SDY1, can use RImmPort to easily access different types of individual-level data -subject demographics, clinical assessments, adverse events, results of flow cytometry and ELISA experiments on 4211 biosamples collected at different time points over 12 weeks from 159 subjects.

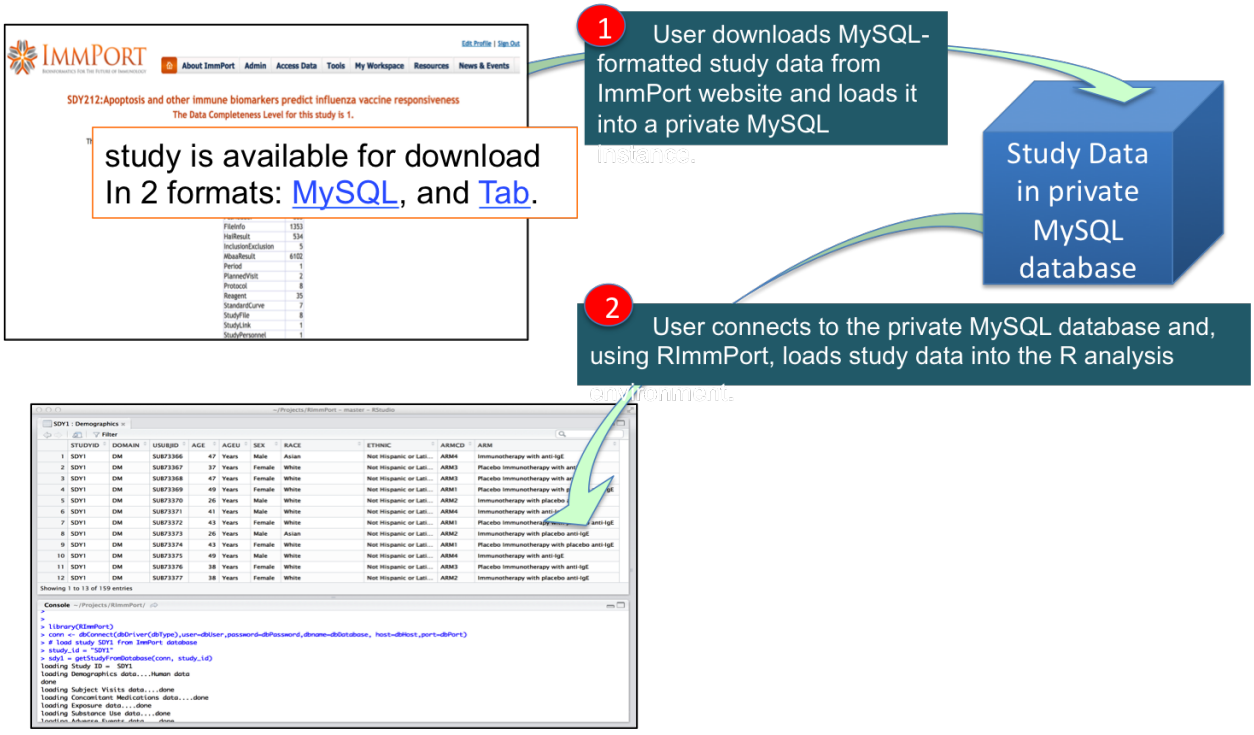


Figure 1: RImmPort Data Workflow

4 Overview of CDISC SDTM Data Standard

The Clinical Data Interchange Standards Consortium (CDISC) has been developing a suite of data standards, supporting different stages of the clinical research process: study design, study conduct, data analysis and reporting. The Study Data Tabulation Model (SDTM) defines a data tabulation standard that can be used to structure study data when submitting to regulatory agencies such as the United States Food and Drug Administration (FDA). In SDTM, the data is structured in a hierarchy, as Classes, Domains, Variables and Values. Here is a list of foundational classes and a partial set of domains under each class:

- Special Purpose Class
  - Demographics (DM) Domain
  - Subject Visits (SV) Domain
- Interventions Class
  - Concomitant Medications (CM) Domain
  - Exposure (EX) Domain
  - Substance Use (SU) Domain
- Events Class
  - Adverse Events (AE) Domain
  - Medical History (MH) Domain
- Findings Class
  - Laboratory Test Results (LB) Domain
  - Vital Signs (VS) Domain
  - Physical Examination (PE) Domain
- Trial Design Class
  - Trial Summary (TS) Domain
  - Trial Arms (TA) Domain

Each Domain defines a standard set of Variables that are appropriate for that Domain, and should be used to

structure data of that domain. As an example, the table shows a partial list of Variables for the Demographics (DE) Domain:

Variable Name	Variable Label
STUDYID	Study Identifier
DOMAIN	Domain Abbreviation
USUBJID	Unique Subject Identifier
AGE	Age
AGEU	Age Units
SEX	Sex
RACE	Race
ARMCD	Planned Arm Code

Continuing with this example, Demographics data of a specific study can be tabulated using the SDTM Demographics Domain:

row	STUDYID	DOMAIN	USUBJID	AGE	AGEU	SEX	RACE
1	SDY1	DM	SUB73366	47	Years	Male	White
2	SDY1	DM	SUB73367	37	Years	Female	Asian
3	SDY1	DM	SUB73368	47	Years	Female	Multiple

If the study data has variables that are not in the standard set, SDTM provides a Name-Value pair mechanism to create Supplemental Domain data. In the example, the standard variable RACE captures subject's race. If the study has collected data on all applicable races of a subject, then the Supplemental Demographics Domain data, can be tabulated as follows, with RACE1 and RACE2 as non-standard Variables:

row	STUDYID	RDOMAIN	USUBJID	QNAM	QLABEL	QVALUE
1	SDY1	DM	SUB73368	RACE1	Race 1	Black or African American
2	SDY1	DM	SUB73368	RACE2	Race 2	American Indian or Alaska Native

Thus, all of the study data can be structured as SDTM Domain data tables, and any supplemental Domain data tables. For new types of research data that does not naturally map to an existing SDTM domain, SDTM allows creating custom Domains. The custom Domain can then be proposed to the SDTM standards committee, to be developed into a standard.

5 RImmPort Data Model

The RImmPort data model implements the SDTM data standards in encapsulating the different types of the ImmPort study data. The model comprises of the following Classes and Domains (codes in parentheses):

- Special Purpose
  - Demographics (DM) - describes each subject in a clinical study
  - Subject Visits (SV) - collates the timing information on subject visits
- Interventions
  - Concomitant Medications (CM) - concomitant and prior medications/therapies used by the subject
  - Exposure (EX) - subject's exposure to protocol specified study treatment
  - Substance Use (SU) - substance use information
- Events
  - Adverse Events (AE) - adverse event information
  - Protocol Deviations (DV) - protocol violations and deviations during the course of the study
  - Medical History (MH) - subject's prior medical history at the start of the trial
  - Associated Persons Medical History (APMH) - medical history of persons associated with the subject
- Findings
  - Laboratory Test Results (LB) - laboratory test findings including, but not limited to hematology, clinical chemistry and urinalysis data.
  - Vital Signs (VS) - measurements including but not limited to blood pressure, temperature, respiration, body surface area, BMI, height and weight.
  - Questionnaires (QS) - information on questionnaires
  - Physical Examination (PE) - findings collected during a physical examination of the subject
  - Findings About (FA) - a dataset used to capture the findings about an event or intervention that cannot be represented within an event or intervention record or as a supplemental.
  - Skin Response (SR) - skin assessment results
  - Genetics Findings (PF) - genetics findings (Array and HLA Typing assays)
  - Protein Quantification (ZA) - non-standard domain for protein quantification data (ELISA and MBAA assays)
  - Cellular Quantification (ZB) - non-standard domain for cellular quantification data (Flow and ELISPOT assays)
  - Nucleic Acid Quantification (ZC) - non-standard domain for nucleic acid quantification data (PCR assays)
  - Titer Assay Results (ZD) - non-standard domain for titer assay results data (HAI and Neut Ab Titer assays)
- Trial Design
  - Trial Arms (TA) - all the planned arms in the trial
  - Trial Inclusion Exclusion Criteria (TI) - all the inclusion and exclusion criteria for the trial
  - Trial Summary (TS) - information about the planned and actual trial characteristics

5.1 RImmPort `Study` reference class

The `Study` reference class and its components (shown below) form the core of the RImmPort data model, and is designed to parallel SDTM's Class, Domain, Variable and Value model.

```
# Truncated code to improve clarity - for illustration purpose only

Study <- setRefClass("Study", fields = list(
  special_purpose="SpecialPurpose",
  interventions="Interventions",
  events="Events",
  findings="Findings",
```

```
  trial_design="TrialDesign"),
  ...
))

SpecialPurpose <- setRefClass("SpecialPurpose",
                              fields = list(
                                dm_l="list",
                                sv_l="list"),
                              ...

Interventions <- setRefClass("Interventions",
                                fields = list(
                                  cm_l="list",
                                  ex_l="list",
                                  su_l="list"),
                              ...

Events <- setRefClass("Events", fields = list(
                                  ae_l="list",
                                  dv_l="list",
                                  mh_l="list",
                                  apmh_l="list"),
                        ...

Findings <- setRefClass("Findings",
                          fields = list(
                                  lb_l="list",
                                  pe_l="list",
                                  vs_l="list",
                                  qs_l="list",
                                  fa_l="list",
                                  sr_l="list",
                                  pf_l="list",
                                  za_l="list",
                                  zb_l="list",
                                  zc_l="list",
                                  zd_l="list"),
                          ...

TrialDesign <- setRefClass("TrialDesign",
                          fields = list(
                            ta_l="list",
                            ti_l="list",
                            ts_l="list"),
                          ...
```

5.1.1 Notes on `Study` reference class and its components

- There is a reference class for each of the five SDTM Classes (SpecialPurpose, Interventions, Events, Findings, and TrialDesign).
- The Domains are listed as fields of respective reference classes. For example, the field `dm_l` in the `SpecialPurpose` reference class is associated with the Demographics domain. The convention in naming the Domain field is `<lower-case Domain code>_l`, for e.g., `dm_l` for Demographics domain.

- All the Domain fields (`dm_l`, `sv_l`, `lb_l`, etc.) are of type `list`. When the `'Study` reference class is instantiated with data of a specific study, each domain field will contain a list of 2 named dataframes, Domain data and any supplemental Domain data. For.e.g., `dm_l` is a list of `dm_f` (Demographics data), and `suppdm_f` (supplemental Demographics data).
- In a specific study, if there is specific Domain data, but no corresponding supplemental Domain data, then there will be an empty supplemental dataframe. For e.g., if there is no supplemental Demographics data, then `suppdm_df` will be an empty dataframe.
- In a specifc study, if there is no data for a specific domain, then the Domain field list will be empty. For e.g., if there no Substance Use data, then the `su_l` will be an empty list.
- RImmPort function `getStudy()` instantiates the `Study` reference class with data of a specific study in the ImmPort data source The hierarchical structure of the `Study` instance can then be traversed to access different types of study data (see Section `Get all data of a specific study`).

6 RImmPort Functions

RImmPort supports a set of functions to load different types of study data. The functions generally comprise of 3 steps: 1) query study data from the ImmPort data source, 2) map the data to CDISC SDTM-based RImmPort model, and 3) load the data into the R environment. The documentation on all these functions can be found in the RImmPort reference manual (`RImmPort-manual.pdf`). The documentation can also be accessed from the R console using the `help (?)` function, for e.g., `?getStudy`.

6.1 Foundational Functions

The foundational functions enable loading different types of study data of a specific ImmPort study. An intitial step, before invoking any of the "get data" functions, is setting the ImmPort data source that these functions operate upon. As mentioned earlier, ImmPort data comes in 2 formats, MySQL and Tab, and RImmPort can work with either one. User can set up a MySQL data source by creating a MySQL database and loading the database with ImmPort study data that is in MySQL format. For the Tab data source, RImmPort provides a utility function to create a private SQLite database and load the database with ImmPort study data that is in Tab format. The `setImmPortDataSource` function takes in as input a connection handle to either the MySQL database or the SQLite database, and sets that database as the ImmPort data source that other RImmPort functions use.

6.1.1 Load the RImmPort library

```
library(RImmPort)
library(DBI)
library(sqldf)
```

```
## Loading required package: gsubfn
## Loading required package: proto
## Loading required package: RSQLite
```

```
library(plyr)
```

6.1.2 Set the MySQL database as ImmPort data source

```
# provide appropriate MySQL database connection parameters
mysql_conn <- dbConnect(MySQL(), user="username", password="password",
                  dbname="database",host="host")

# set the data source as the ImmPort MySQL database.
```

```
setImmPortDataSource(mysql_conn)
```

6.2 Option 2: ImmPort SQLite database

6.2.1 Download zip files of ImmPort data, in Tab format. e.g.'SDY139' and 'SDY208'

```
# get the directory where ImmPort sample data is stored in the directory structure of RImmPort package
studies_dir <- system.file("extdata", "ImmPortStudies", package = "RImmPort")
```

```
# set tab_dir to the folder where the zip files are located
tab_dir <- file.path(studies_dir, "Tab")
list.files(tab_dir)
```

6.2.2 Build a local SQLite ImmPort database instance.

```
# set db_dir to the folder where the database file 'ImmPort.sqlite' should be stored
db_dir <- file.path(studies_dir, "Db")
```

```
# build a new ImmPort SQLite database with the data in the downloaded zip files
buildNewSqliteDb(tab_dir, db_dir)
```

```
list.files(db_dir)
```

6.2.3 Set the SQLite database as ImmPort data source

```
# get the directory of a sample SQLite database that has been bundled into the RImmPort package
db_dir <- system.file("extdata", "ImmPortStudies", "Db", package = "RImmPort")

# connect to the database
sqlite_conn <- dbConnect(SQLite(), dbname=file.path(db_dir, "ImmPort.sqlite"))

# set the ImmPort SQLite database as the ImmPort data source
setImmPortDataSource(sqlite_conn)
```

```
## [1] 1
```

6.2.4 NOTE: In rest of this document, all RImmPort functions will use the SQLite ImmPort database as the ImmPort data source.

6.2.5 Get all data of a specific study

The `getStudy` queries the ImmPort database for the entire dataset of a specifc study, and instantiates the `Study` reference class with that data.

```
# view documentation on Study
?Study

# get all study ids
getListOfStudies()
```

```
## [1] "SDY139" "SDY208"
```

```
# load the entire data of study: `SDY139`
study_id <- 'SDY139'
sdy139 <- getStudy(study_id)
```

```
## loading Study ID =  SDY139
## loading Demographics data....done
## loading Concomitant Medications data....done
## loading Exposure data....done
## loading Substance Use data....done
```

```
## loading Adverse Events data....done
## loading Protocol Deviations data....done
## loading Medical History data....done
## loading Associated Persons Medical History data....done
## loading Laboratory Test Results data....done
## loading Physical Examination data....done
## loading Vital Signs data....done
## loading Questionnaires data....done
## loading Findings About data....done
## loading Skin Response data....done
## loading Genetics Findings data....loading HLA Typing Results data....done
## loading Array Results data....done
## done
## loading Protein Quantification data....loading ELISA Results data....done
## loading MBAA Results data....done
## done
## loading Cellular Quantification data....loading FCS Results data....done
## loading ELISPOT Results data....done
## done
## loading Nucleic Acid Quantification data....loading PCR Results data....done
## done
## loading Titer Assay Results data....loading HAI Assay Results data....done
## loading Neut. Ab Titer Results data....done
## done
## loading TrialArms data....done
## loading Trial Visits data....done
## loading  TrialInclusionExclusionCriteria data....done
## loading TrialSummary data.... SDY139  done
## done loading Study ID =  SDY139
```

```
# access Demographics data of SDY139
dm_df <- sdy139$special_purpose$dm_l$dm_df
head(dm_df)
```

```
##   STUDYID DOMAIN   USUBJID AGE   AGEU     SEX RACE ETHNIC      SPECIES
## 1  SDY139     DM SUB118053   2 Months Unknown <NA>   <NA> Mus musculus
## 2  SDY139     DM SUB118054   2 Months Unknown <NA>   <NA> Mus musculus
## 3  SDY139     DM SUB118055   2 Months Unknown <NA>   <NA> Mus musculus
## 4  SDY139     DM SUB118056   2 Months Unknown <NA>   <NA> Mus musculus
## 5  SDY139     DM SUB118057   2 Months Unknown <NA>   <NA> Mus musculus
## 6  SDY139     DM SUB118058   2 Months Unknown <NA>   <NA> Mus musculus
##   STRAIN SBSTRAIN  ARMCD    ARM
## 1 BALB/c     <NA> ARM678 BALB/c
## 2 BALB/c     <NA> ARM678 BALB/c
## 3 BALB/c     <NA> ARM678 BALB/c
## 4 BALB/c     <NA> ARM678 BALB/c
## 5 BALB/c     <NA> ARM678 BALB/c
## 6 BALB/c     <NA> ARM678 BALB/c
```

```
# access Concomitant Medications data of SDY139
cm_df <- sdy139$interventions$cm_l$cm_df
head(cm_df)
```

```
## NULL
```

```
# get Trial Title from Trial Summary
ts_df <- sdy139$trial_design$ts_l$ts_df
title <- ts_df$TSVAL[ts_df$TSPARMCD== "TITLE"]
title
```

```
## [1] "The peptide specificity of the endogenous T follicular helper cell repertoire generated after p
```

6.2.6 Get specific domain data of a specific study

The `getDomainDataOfStudies` function takes as inputs, the name of the Domain of interest, and one or more study identifiers. The function returns a list of 2 named dataframes, Domain data and any supplemental Domain data; for e.g.,`dm_f` (Demographics data), and `suppdm_f` (supplemental Demographics data). The Domain name should be exact to what is found in the list of Domain names.

```
# get the list of names of all supported Domains
getListOfDomains()
```

```
##                           Domain Name Domain Code
## 1                       Adverse Events          AE
## 2             Concomitant Medications          CM
## 3                         Demographics          DM
## 4                             Exposure          EX
## 5                      Medical History          MH
## 6  Associated Persons Medical History        APMH
## 7             Laboratory Test Results          LB
## 8                Physical Examination          PE
## 9                 Protocol Deviations          DV
## 10                          Trial Arms          TA
## 11                        Trial Visits          TV
## 12 Trial Inclusion Exclusion Criteria          TI
## 13                       Trial Summary          TS
## 14                       Substance Use          SU
## 15                         Vital Signs          VS
## 16                       Questionnaires          QS
## 17                       Findings About          FA
## 18                        Skin Response          SR
## 19                    Genetics Findings          PF
## 20             Protein Quantification          ZA
## 21            Cellular Quantification          ZB
## 22        Nucleic Acid Quantification          ZC
## 23                Titer Assay Results          ZD
```

```
?"Demographics Domain"

# get domain code of Demographics domain
domain_name <- "Demographics"
getDomainCode(domain_name)
```

```
## [1] "DM"
```

```
dm_l <- getDomainDataOfStudies(domain_name, "SDY139")
```

```
## loading Demographics data....done
```

```
if (length(dm_l) > 0)
  names(dm_l)
```

```
## [1] "dm_df"     "suppdm_df"
```

```
head(dm_l$dm_df)
```

```
##   STUDYID DOMAIN   USUBJID AGE   AGEU     SEX RACE ETHNIC      SPECIES
## 1  SDY139     DM SUB118053   2 Months Unknown <NA>   <NA> Mus musculus
## 2  SDY139     DM SUB118054   2 Months Unknown <NA>   <NA> Mus musculus
## 3  SDY139     DM SUB118055   2 Months Unknown <NA>   <NA> Mus musculus
## 4  SDY139     DM SUB118056   2 Months Unknown <NA>   <NA> Mus musculus
## 5  SDY139     DM SUB118057   2 Months Unknown <NA>   <NA> Mus musculus
## 6  SDY139     DM SUB118058   2 Months Unknown <NA>   <NA> Mus musculus
##   STRAIN SBSTRAIN  ARMCD    ARM
## 1 BALB/c     <NA> ARM678 BALB/c
## 2 BALB/c     <NA> ARM678 BALB/c
## 3 BALB/c     <NA> ARM678 BALB/c
## 4 BALB/c     <NA> ARM678 BALB/c
## 5 BALB/c     <NA> ARM678 BALB/c
## 6 BALB/c     <NA> ARM678 BALB/c
```

6.3 Search and Integrate Functions

RImmPort supports methods to search and integrate specific types of study data of multiple studies.

6.3.1 Get studies with specifc domain data

The `getStudiesWithSpecificDomainData` takes as input the name of Domain of interest, and outputs a list of studies that have the specific Domain data.

```
# get list of studies with Cellular Quantification data
domain_name <- "Cellular Quantification"
study_ids_l <- getStudiesWithSpecificDomainData(domain_name)
study_ids_l
```

```
## [1] "SDY139" "SDY208"
```

6.3.2 Get specifc domain data of a set of studies

The `getDomainDataOfStudies` function takes as inputs, the name of the Domain of interest, and a list of study identifiers, and outputs a Domain dataframe and any supplemental Domain dataframe that contain integrated Domain data.

```
# get Cellular Quantification data of studies `SDY139` and `SDY208`

# get domain code of Cellular Quantification domain
domain_name <- "Cellular Quantification"
getDomainCode(domain_name)
```

```
## [1] "ZB"
```

```
study_ids <- c("SDY139", "SDY208")
domain_name <- "Cellular Quantification"
zb_l <- getDomainDataOfStudies(domain_name, study_ids)
```

```
## loading Cellular Quantification data....loading FCS Results data....done
## loading ELISPOT Results data....done
```

```
## done
## loading Cellular Quantification data....loading FCS Results data....done
## loading ELISPOT Results data....done
## done
```

```
if (length(zb_l) > 0)
  names(zb_l)
```

```
## [1] "zb_df"     "suppzb_df"
```

```
head(zb_l$zb_df)
```

```
##   STUDYID DOMAIN   USUBJID ZBSEQ       ZBTEST                   ZBCAT
## 1  SDY139     ZB SUB118078     1 Figure-7_FCM Cellular Quantification
## 2  SDY139     ZB SUB118078     2 Figure-7_FCM Cellular Quantification
## 3  SDY139     ZB SUB118078     3 Figure-7_FCM Cellular Quantification
## 4  SDY139     ZB SUB118078     4 Figure-7_FCM Cellular Quantification
## 5  SDY139     ZB SUB118078     5 Figure-7_FCM Cellular Quantification
## 6  SDY139     ZB SUB118078     6 Figure-7_FCM Cellular Quantification
##         ZBMETHOD ZBPOPDEF ZBPOPNAM ZBORRES ZBORRESU ZBBASPOP ZBSPEC
## 1 Flow Cytometry                                                Cell
## 2 Flow Cytometry                                                Cell
## 3 Flow Cytometry                                                Cell
## 4 Flow Cytometry                                                Cell
## 5 Flow Cytometry                                                Cell
## 6 Flow Cytometry                                                Cell
##   VISITNUM                                                   VISIT ZBELTM
## 1        1 Day 0 Protein/peptide inoculation, SL_Sant_Plos1_2012_d0    POD
## 2        1 Day 0 Protein/peptide inoculation, SL_Sant_Plos1_2012_d0    POD
## 3        1 Day 0 Protein/peptide inoculation, SL_Sant_Plos1_2012_d0    POD
## 4        1 Day 0 Protein/peptide inoculation, SL_Sant_Plos1_2012_d0    POD
## 5        1 Day 0 Protein/peptide inoculation, SL_Sant_Plos1_2012_d0    POD
## 6        1 Day 0 Protein/peptide inoculation, SL_Sant_Plos1_2012_d0    POD
##                                 ZBTPTREF  ZBREFID
## 1 Time of initial vaccine administration ES662746
## 2 Time of initial vaccine administration ES662746
## 3 Time of initial vaccine administration ES662766
## 4 Time of initial vaccine administration ES662766
## 5 Time of initial vaccine administration ES662786
## 6 Time of initial vaccine administration ES662786
##                          ZBXFN
## 1   Tfh_Tfh CLN D0-1.297191.fcs
## 2   Tfh_Tfh CLN D0-1.297192.txt
## 3 Tfh 1_Tfh EAR D0-1.297261.fcs
## 4 Tfh 1_Tfh EAR D0-1.297262.txt
## 5   Tfh_Tfh ILN D0-1.297339.fcs
## 6   Tfh_Tfh ILN D0-1.297340.txt
```

6.3.3 Get the list of assay types

The function `getListOfAssayTypes` returns a list of assay types that ImmPort has data for.

```
getListOfAssayTypes()
```

```
## [1] "ELISA"         "ELISPOT"       "Array"         "PCR"
## [5] "HLA Typing"    "MBAA"          "HAI"           "Neut Ab Titer"
```

```
## [9] "Flow"
```

6.3.4 Get specific assay data of one or more studies

The results of assays that were performed in the study are organized into 5 specialized Findings domains: Genetics Findings (PF), Protein Quantification (ZA), Cellular Quantification (ZB), Nucleic Acid Quantification (ZC) and Titer Assay Results (ZD). There can be a many-to-many relationship between the 5 domains and the assay types. The RImmPort function `getAssayDataOfStudies()` takes as input an assay type and and a list of study identifiers, and outputs a list of dataframes of the 5 Findings domains that have the assay data.

```
# get 'ELISPOT' data of study `SDY139`
assay_type <- "ELISPOT"
study_id = "SDY139"
elispot_l <- getAssayDataOfStudies(study_id, assay_type)
```

```
## loading Protein Quantification data....done
## loading Cellular Quantification data....loading ELISPOT Results data....done
## done
## loading Nucleic Acid Quantification data....done
## loading Titer Assay Results data....done
## loading Genetics Findings data....done
```

```
if (length(elispot_l) > 0)
  names(elispot_l)
```

```
## [1] "zb_df"     "suppzb_df"
```

```
head(elispot_l$zb_df)
```

```
##   STUDYID DOMAIN   USUBJID ZBSEQ           ZBTEST                   ZBCAT
## 1  SDY139     ZB SUB118053  8675 Figure-4_ELISPOT Cellular Quantification
## 2  SDY139     ZB SUB118053  8658 Figure-4_ELISPOT Cellular Quantification
## 3  SDY139     ZB SUB118053  8673 Figure-4_ELISPOT Cellular Quantification
## 4  SDY139     ZB SUB118053  8660 Figure-4_ELISPOT Cellular Quantification
## 5  SDY139     ZB SUB118053  8662 Figure-4_ELISPOT Cellular Quantification
## 6  SDY139     ZB SUB118053  8663 Figure-4_ELISPOT Cellular Quantification
##   ZBMETHOD ZBPOPDEF ZBPOPNAM  ZBORRES ZBORRESU ZBBASPOP ZBSPEC VISITNUM
## 1  ELISPOT     IL-2     IL-2 622.8571 1000000             Cell        3
## 2  ELISPOT    IL-21    IL-21   8337.5 1000000             Cell        3
## 3  ELISPOT     IL-2     IL-2 1048.571 1000000             Cell        3
## 4  ELISPOT    IL-21    IL-21   3925.0 1000000             Cell        3
## 5  ELISPOT    IL-21    IL-21    600.0 1000000             Cell        3
## 6  ELISPOT    IL-21    IL-21 798.5714 1000000             Cell        3
##                                           VISIT ZBELTM
## 1 Day 8 Sample collection, SL_Sant_Plos1_2012_d8    P8D
## 2 Day 8 Sample collection, SL_Sant_Plos1_2012_d8    P8D
## 3 Day 8 Sample collection, SL_Sant_Plos1_2012_d8    P8D
## 4 Day 8 Sample collection, SL_Sant_Plos1_2012_d8    P8D
## 5 Day 8 Sample collection, SL_Sant_Plos1_2012_d8    P8D
## 6 Day 8 Sample collection, SL_Sant_Plos1_2012_d8    P8D
##                                 ZBTPTREF  ZBREFID ZBXFN
## 1 Time of initial vaccine administration ES661770
## 2 Time of initial vaccine administration ES661753
## 3 Time of initial vaccine administration ES661768
## 4 Time of initial vaccine administration ES661755
```

```
## 5 Time of initial vaccine administration ES661757
## 6 Time of initial vaccine administration ES661758
```

6.4 Utility functions

RImmPort has a set of utility methods to export and import study data in RImmPort format, and to build the SQLite database from Tab formatted ImmPort downloads.

6.4.1 Serialize study data in RImmPort format

The function `serialzeStudyData()` serializes the entire RImmPort-formatted study data as .rds files

```
# get the directory where ImmPort sample data is stored in the directory structure of RImmPort package
studies_dir <- system.file("extdata", "ImmPortStudies", package = "RImmPort")

# serialize all of the data of studies `SDY139` and `SDY208'
study_ids <- c('SDY139', 'SDY208')

# the folder where the .rds files will be stored
rds_dir <- file.path(studies_dir, "Rds")

serialzeStudyData(study_ids, rds_dir)
list.files(rds_dir)
```

6.4.2 Load serialized study data into the R environment

The function `loadSerializedStudyData()` loads the serialzed data (.rds) files of a specific domain of a study from the directory where the files are located

```
# get the directory where ImmPort sample data is stored in the directory structure of RImmPort package
studies_dir <- system.file("extdata", "ImmPortStudies", package = "RImmPort")

# the folder where the .rds files will be stored
rds_dir <- file.path(studies_dir, "Rds")

# list the studies that have been serialized
list.files(rds_dir)
```

```
## [1] "SDY139" "SDY208"
```

```
# load the serialized data of study `SDY208`
study_id <- 'SDY208'
dm_l <- loadSerializedStudyData(rds_dir, study_id, "Demographics")
```

```
##
## domain_file_path =  /tmp/RtmpXxsneZ/Rinst61d43dbcc62d/RImmPort/extdata/ImmPortStudies/Rds/SDY208/dm.
## suppdomain_file_path =  /tmp/RtmpXxsneZ/Rinst61d43dbcc62d/RImmPort/extdata/ImmPortStudies/Rds/SDY208/
```

```
head(dm_l[[1]])
```

```
##   STUDYID DOMAIN   USUBJID AGE  AGEU    SEX RACE ETHNIC      SPECIES
## 1  SDY208     DM SUB120516   6 Weeks Female <NA>   <NA> Mus musculus
## 2  SDY208     DM SUB120517   6 Weeks Female <NA>   <NA> Mus musculus
## 3  SDY208     DM SUB120518   6 Weeks Female <NA>   <NA> Mus musculus
```

```
## 4  SDY208     DM SUB120519   6 Weeks Female <NA>   <NA> Mus musculus
## 5  SDY208     DM SUB120520   6 Weeks Female <NA>   <NA> Mus musculus
## 6  SDY208     DM SUB120521   6 Weeks Female <NA>   <NA> Mus musculus
##   STRAIN SBSTRAIN  ARMCD
## 1   <NA>     <NA> ARM881
## 2   <NA>     <NA> ARM882
## 3   <NA>     <NA> ARM883
## 4   <NA>     <NA> ARM884
## 5   <NA>     <NA> ARM885
## 6   <NA>     <NA> ARM886
##
## 1                                                  Microneedle vaccination- 5 ug inactivated A/Califo
## 2                                                Subcutaneous vaccination-  5 ug inactivated A/Califo
## 3                                                                      Uncoated microneedle vacci
## 4  Microneedle vaccination- 5 ug inactivated A/California/04/09 virus, Challenged: 10x LD50 A/Califo
## 5 Subcutaneous vaccination- 5 ug inactivated A/California/04/09 virus, Challenged: 10x LD50 A/Califo
## 6                          Uncoated microneedle vaccination-  Placebo, Challenged: 10x LD50 A/Califo
```

6.4.3 Build a private SQLite ImmPort database instance.

The funciton `buildNewSqliteDb` reads the Tab-formatted study data zip files that the user has downloaded from the ImmPort website, and creates a SQLite database.

```
# get the directory where ImmPort sample data is stored in the directory structure of RImmPort package
studies_dir <- system.file("extdata", "ImmPortStudies", package = "RImmPort")

# set tab_dir to the folder where the zip files are located
tab_dir <- file.path(studies_dir, "Tab")
list.files(tab_dir)
```

```
## [1] "SDY139-DR25_Tab.zip" "SDY208-DR25_Tab.zip"
```

```
# set db_dir to the folder where the database file 'ImmPort.sqlite' should be stored
db_dir <- file.path(studies_dir, "Db")
```

```
# build a new ImmPort SQLite database with the data in the downloaded zip files
buildNewSqliteDb(tab_dir, db_dir)
```

```
list.files(db_dir)
```

```
## [1] "ImmPort.sqlite"
```

7 Conclusion

By basing RImmPort on open formalism and by making it available in open source platforms, we ensure that clinical study data in ImmPort is ready for analysis, thus enabling innovative bioinformatics research in immunology.