Package ‘parseRPDR’

October 14, 2022

Type Package

Title Parse and Manipulate Research Patient Data Registry (‘RPDR’)

Text Queries

Version 0.2.3

Date 2022-06-07

Maintainer Marton Kolossvary <mkolossvary@mgh.harvard.edu>

Description Functions to load Research Patient Data Registry (‘RPDR’) text queries from Partners Healthcare institutions into R.


All functionalities are parallelized for fast and efficient analyses.

License AGPL (>= 3)

Depends R (>= 4.0)

Imports data.table (>= 1.13.2), stringr (>= 1.4.0), readr (>= 1.4.0), doParallel (>= 1.0.16), foreach (>= 1.5.1), bigmemory (>= 4.5.36), reticulate(>= 1.20)

LazyData TRUE

RoxygenNote 7.1.2

NeedsCompilation no

Suggests testthat (>= 3.0.0), knitr, rmarkdown

Encoding UTF-8

Config/testthat/edition 3

Author Marton Kolossvary [aut, cre]

Repository CRAN

Date/Publication 2022-06-07 12:50:04 UTC
R topics documented:

all_ids_mi2b2 .................................................. 3
convert_dia .................................................... 3
convert_enc .................................................... 5
convert_lab ..................................................... 6
convert_med .................................................... 7
convert_notes .................................................. 9
convert_phy ................................................... 11
convert_prc ................................................... 12
convert_rfv ................................................... 14
create_img_db .................................................. 15
data_con ....................................................... 17
data_con_raw ................................................... 18
data_dem ....................................................... 20
data_dem_raw ................................................... 21
datadia ......................................................... 22
data_dia ....................................................... 23
data_dia_raw .................................................... 23
data_enc ....................................................... 24
data_enc_raw .................................................... 26
data_lab ....................................................... 28
data_lab_raw .................................................... 30
data_med ....................................................... 31
data_med_raw .................................................... 32
data_mrn ....................................................... 34
data_mrn_raw .................................................... 35
data_rad ....................................................... 36
data_rad_raw .................................................... 37
data_rdt ....................................................... 38
data_rdt_raw .................................................... 39
find_exam ...................................................... 40
load_all ....................................................... 42
load_con ...................................................... 44
load_dem ...................................................... 46
load_dem_old ................................................... 49
load_dia ....................................................... 51
load_enc ...................................................... 53
load_lab ....................................................... 56
load_lno ....................................................... 58
load_med ...................................................... 60
load_mrn ...................................................... 62
load_notes ..................................................... 64
load_phy ...................................................... 66
load_prc ...................................................... 68
load_rdt ...................................................... 70
load_rfv ...................................................... 72
pretty_mrn .................................................... 74

Index ......................................................... 76
all_ids_mi2b2

**Description**

Gathers all possible IDs from different input sources to provide a vector of all possible MGH or BWH IDs to be used as a data request for mi2b2 workbench. mi2b2 workbench only works with MGH or BWH IDs, therefore curated IDs, such as PMRN cannot be used. However, as MGH and BWH IDs may change over time, to access all possible images for given patients, a full list of all IDs over time for each patient is needed. For this all possible IDs need to gathered and returned.

**Usage**

```r
all_ids_mi2b2(type = "MGH", d_mrn, d_con)
```

**Arguments**

- `type`: string, either "MGH" or "BWH" specifying which IDs to use.
- `d_mrn`: data.table, parsed mrn dataset using the `load_mrn` function.
- `d_con`: data.table, parsed con dataset using the `load_con` function.

**Value**

vector, with all MGH or BWH IDs that occur in the con and mrn datasources for all patients. This is required to request mi2b2 workbench allowing access to all possible images of the patients, even if the MGH or BWH changed over time.

**Examples**

```r
## Not run:
all_MGH_mrn <- all_ids_mi2b2(type = "MGH", d_mrn = data_mrn, d_con = data_con)
## End(Not run)
```

convert_dia

**Description**

Searches diagnosis columns for given diseases.

**Description**

Analyzes diagnosis data loaded using `load_dia`. Searches diagnosis columns for a specified set of diseases. By default, the data.table is returned with new columns corresponding to boolean values, whether given group of diagnoses are present in the given diagnosis. If `collapse` is given, then the information is aggregated based-on the `collapse` column and the earliest of latest time of the given diagnosis is provided.
convert_dia

Usage

convert_dia(
  d,
  code = "dia_code",
  code_type = "dia_code_type",
  codes_to_find = NULL,
  collapse = NULL,
  code_time = "time_dia",
  time_type = "earliest",
  nThread = 4
)

Arguments

d data.table, database containing diagnosis information data loaded using the load_dia function.

code string, column name of the diagnosis code column. Defaults to dia_code.

code_type string, column name of the code_type column. Defaults to dia_code_type.

codes_to_find list, a list of string arrays corresponding to sets of code types and codes separated by ;, i.e.: "ICD9:250.00". The function searches for the given disease code type and code pair and adds new boolean columns with the name of each list element. These columns are indicators whether any of the disease code type and code pair occurs in the set of codes.

collapse string, a column name on which to collapse the data.table. Used in case we wish to assess whether given disease codes are present within all the same instances of collapse. See vignette for details.

code_time string, column name of the time column. Defaults to time_dia. Used in case collapse is present to provide the earliest or latest instance of diagnosing the given disease.

time_type string, if multiple diagnoses are present within the same case of collapse, which timepoint to return. Supported are: "earliest" or "latest". Defaults to earliest.

nThread integer, number of threads to use by dopar for parallelization. If it is set to 1, then no parallel backends are created and the function is executed sequentially. On windows machines sockets are used, while on other operating systems fork parallelization is used.

Value

data.table, with indicator columns whether the any of the given diagnoses are reported. If collapse is present, then only unique ID and the summary columns are returned.

Examples

## Not run:
#Search for Hypertension and Stroke ICD codes
diseases <- list(HT = c("ICD10:I10"), Stroke = c("ICD9:434.91", "ICD9:I63.50"))
data_dia_parse <- convert_dia(d = data_dia, codes_to_find = diseases, nThread = 2)
# Searches for Hypertension and Stroke ICD codes and summarize per patient providing earliest time
diseases <- list(HT = c("ICD10:I10"), Stroke = c("ICD9:434.91", "ICD9:I63.50"))
data_dia_disease <- convert_dia(d = data_dia, codes_to_find = diseases, nThread = 2, collapse = "ID_MERGE", time_type = "earliest")
## End(Not run)

---

### `convert_enc`

**Searches columns for given diseases defined by ICD codes.**

**Description**

Analyzes encounter data loaded using `load_enc`. Converts columns with ICD codes and text to simple ICD codes. If requested, the data.table is returned with new columns corresponding to boolean values, whether given group of diagnoses are present in the given columns. If `collapse` is given, then the information is aggregated based on the `collapse` column and the earliest of latest time of the given diagnosis is provided.

**Usage**

```r
convert_enc(
  d,
  code = c("enc_diag_admit", "enc_diag_princ", paste0("enc_diag_{_", 1:10)),
  keep = FALSE,
  codes_to_find = NULL,
  collapse = NULL,
  code_time = "time_enc_admit",
  time_type = "earliest",
  nThread = 4
)
```

**Arguments**

- **d**
  - data.table, database containing encounter information data loaded using the `load_enc` function.
- **code**
  - string vector, an array of column names to convert to simple ICD codes. The new column names will be the old one with `ICD_` added to the beginning of it.
- **keep**
  - boolean, whether to keep original columns that were converted. Defaults to `FALSE`.
- **codes_to_find**
  - list, a list of arrays corresponding to sets of ICD codes. The function searches the columns in code and new boolean columns with the name of each list element will be created. These columns are indicators whether the given disease is present in the set of ICD codes or not.
- **collapse**
  - string, a column name on which to collapse the data.table. Used in case we wish to assess whether given diagnoses are present within all the same instances of `collapse`. See vignette for details.
code_time string, column name of the time column. Defaults to `time_enc_admit`. Used in case collapse is present to provide the earliest or latest instance of diagnosing the given disease.

time_type string, if multiple diagnoses are present within the same case of `collapse`, which timepoint to return. Supported are: "earliest" or "latest". Defaults to `earliest`.

nThread integer, number of threads to use by `dopar` for parallelization. If it is set to 1, then no parallel backends are created and the function is executed sequentially. On windows machines sockets are used, while on other operating systems fork parallelization is used.

Value
data.table, with formatted ICD code columns and possibly indicator columns if provided. If `collapse` is present, then only unique ID and the summary columns are returned.

Examples

```r
## Not run:
#Parse encounter ICD columns and keep original ones as well
data_enc_parse <- convert_enc(d = data_enc, keep = TRUE, nThread = 2)

#Parse encounter ICD columns and discard original ones, and create indicator variable for the following diseases
diseases <- list(HT = c("I10"), Stroke = c("434.91", "I63.50"))
data_enc_disease <- convert_enc(d = data_enc, keep = FALSE, codes_to_find = diseases, nThread = 2)

#Parse encounter ICD columns and discard original ones and create indicator variables for the following diseases and summarize per patient, whether there are any encounters where the given diseases were registered
diseases <- list(HT = c("I10"), Stroke = c("434.91", "I63.50"))
data_enc_disease <- convert_enc(d = data_enc, keep = FALSE, codes_to_find = diseases, nThread = 2, collapse = "ID_MERGE")
```

## End(Not run)

**convert_lab**

Converts lab results to normal/abnormal based-on reference values.

Description

Analyzes laboratory data loaded using `load_lab`. Converts laboratory results to values without ">" or "<" by creating a column where these characters are removed. Furthermore, adds two indicator columns where based-on the reference ranges or the Abnormal_Flag column in RPDR (lab_result_abn using `load_lab`), the value is considered normal or abnormal.
convert_med

Usage

convert_lab(
  d,
  code_results = "lab_result",
  code_reference = "lab_result_range",
  code_flag = "lab_result_abn"
)

Arguments

d data.table, database containing laboratory results data loaded using the load_lab function.
code_results string vector, column name containing the results. Defaults to: "lab_result".
code_reference string vector, column name containing the reference ranges. Defaults to: "lab_result_range".
code_flag string vector, column name containing the abnormal flags. Defaults to: "lab_result_abn".

Value

data.table, with three additional columns: "lab_result_pretty" containing numerical results. In case of ">" or "<" notation, the numeric value is returned, as we only have information that it is at least as much or not larger than a given value. The other column: "lab_result_abn_pretty" can take values: NORMAL/ABNORMAL, depending on whether the value is within the reference range. Please be aware that there can be very different representations of values, and in some cases this will result in misclassification of values. The third column: "lab_result_abn_flag_pretty" gives abnormal if the original Abnormal_Flag column contains any information. Borderline values are considered NORMAL.

Examples

## Not run:
#Convert loaded lab results
data_lab_pretty <- convert_lab(d = data_lab)
data_lab_pretty[, c("lab_result", "lab_result_pretty", "lab_result_range",
  "lab_result_abn_pretty", "lab_result_abn_flag_pretty")]

## End(Not run)

---

convert_med Adds boolean columns corresponding to a group of medications whether it is present in the given row.

Description

Analyzes medication data loaded using load_med. By default, the data.table is returned with new columns corresponding to boolean values, whether given group of diagnoses are present in the given diagnosis. If collapse is given, then the information is aggregated based-on the collapse column and the earliest of latest time of the given diagnosis is provided.
convert_med

Usage

convert_med(
  d,
  code = "med",
  codes_to_find = NULL,
  collapse = NULL,
  code_time = "time_med",
  time_type = "earliest",
  nThread = 4
)

Arguments

d data.table, database containing medication data loaded using the load_med function.
code string, column name of the medication column. Defaults to med.
codes_to_find list, a list of arrays corresponding to sets of medication names. New boolean columns with the name of each list element will be created. These columns are indicators whether the given medication is present in the set of medication names or not.
collapse string, a column name on which to collapse the data.table. Used in case we wish to assess whether given medications are present within all the same instances of collapse. See vignette for details.
code_time string, column name of the time column. Defaults to time_med. Used in case collapse is present to provide the earliest or latest instance of diagnosing the given disease.
time_type string, if multiple diagnoses are present within the same case of collapse, which timepoint to return. Supported are: "earliest" or "latest". Defaults to earliest.
nThread integer, number of threads to use by dopar for parallelization. If it is set to 1, then no parallel backends are created and the function is executed sequentially. On windows machines sockets are used, while on other operating systems fork parallelization is used.

Value
data.table, with indicator columns whether given group of codes_to_find is present or not. If collapse is present, then only unique ID and the summary columns are returned.

Examples

## Not run:
#Define medication group and add an indicator column whether
the given medication group was administered
meds <- list(statin = c("Simvastatin", "Atorvastatin"),
             NSAID = c("Acetaminophen", "Paracetamol"))
data_med_indic <- convert_med(d = data_med, codes_to_find = meds, nThread = 1)
#Summarize per patient if they ever had the given medication groups registered

data_med_indic_any <- convert_med(d = data_med, 
codes_to_find = meds, collapse = "ID_MERGE", nThread = 2)

## End(Not run)

**Description**

Analyzes notes loaded using `load_notes` or `load_lno`. Extracts information from the free text present in `abc_rep_txt`, where `abc` stands for the three letter abbreviation of the given type of note. An array of string is provided using the `anchors` argument. The function will return as many columns as there are anchor points. Each column will contain the text between the given anchor point and the next following anchor point. This way the free text report is split into corresponding smaller texts. By default, these are the common standard elements of given note types. Here are provided potential anchor points for the given types of notes:

- **Cardiology**: c("Report Number:", "Report Status:", "Type:", "Date:", "Ordering Provider:", "SYSTOLIC BLOOD PRESSURE", "DIASTOLIC BLOOD PRESSURE", "VENTRICULAR RATE EKG/MIN", "ATRIAL RATE", "PR INTERVAL", "QRS DURATION", "QT INTERVAL", "P AXIS", "R AXIS", "T WAVE AXIS", "LOC", "DX:", "REF:", "Electronically Signed", "report_end")
- **Discharge**: c("***This text report", "Patient Information", "Physician Discharge Summary", "Surgeries this Admission", "Items for Post-Hospitalization Follow-Up:", "Pending Results", "Hospital Course", "ED Course:", "Diagnosis", "Prescriptions prior to admission", "Family History:" "Physical Exam on Admission:" "Discharge Exam", "report_end")
- **Endoscopy**: c("NAME:", "DATE:" "Patient Information", "report_end")
- **History & Physical**: c("***This text report", "Patient Information", "H&P by", "Author:", "Service:", "Author Type:" "Filed:" "Note Time:" "Status:" "Editor:" "report_end")
- **Operative**: c("NAME:", "UNIT NO:" "DATE:" "SURGEON:" "ASST:" "PREOPERATIVE DIAGNOSIS:"" "POSTOPERATIVE DIAGNOSIS:" "NAME OF OPERATION:" "ANESTHESIA:" "INDICATIONS:" "OPERATIVE FINDINGS:" "DESCRIPTION OF PROCEDURE:" "Electronically Signed", "report_end")
- **Pathology**: c("Accession Number:" "Report Status:" "Type:" "Report:" "CASE:" "PATIENT:" "Date:" "Source Care Unit:" "Path Subspecialty Service:" "Results To:" "Signed Out by:" "CLINICAL DATA:" "FINAL DIAGNOSIS:" "GROSS DESCRIPTION:" "report_end")
- **Progress**: c("***This text report", "Patient Information", "History", "Overview", "Progress Notes", "Medications", "Relevant Orders", "Level of Service", "report_end")
- **Pulmonary**: c("The Pulmonary document", "Name:" "Unit #:" "Date:" "Location:" "Smoking Status:" "Pack Years:" "SPIROMETRY:" "LUNG VOLUMES:" "DIFFUSION:" "PLETHYSMOGRAPHY:" "Pulmonary Function Test Interpretation", "Spirometry", "report_end")

• Visit: c("***This text report", "Reason for Visit", "Reason for Visit", "Vital Signs", "Chief Complaint", "History", "Overview", "Medications", "Relevant Orders", "Level of Service", "report_end"

• LMR: c("Subject", "Patient Name:", "Reason for visit", "report_end"

However, these may be modified and extended to include sections of interest, i.e. if a given score is reported in a standard fashion, then adding this phrase (i.e. "CAD-RADS") would create a column where the text following this statement is returned. After this the resulting columns can be easily cleaned up if needed. Be aware to always include "report_end" in the anchors array, to provide the function of the last occurring statement in the report.

Usage

convert_notes(d, code = NULL, anchors = NULL, nThread = 4)

Arguments

d  data.table, database containing notes loaded using the load_notes function.

code  string vector, column name containing the results, which should be "abc_rep_txt", where abc stands for the three letter abbreviation of the given type of note.

anchors  string array, elements to search for in the text report.

nThread  integer, number of threads to use by dopar for parallelization. If it is set to 1, then no parallel backends are created and the function is executed sequentially. On windows machines sockets are used, while on other operating systems fork parallelization is used.

Value

data.table, with new columns corresponding to elements in anchors.

Examples

```r
# Not run:
#Create columns with specific parts of the radiological report defined by anchors

# End(Not run)
```
convert_phy

Description

Analyzes health history data loaded using `load_phy`. Searches health history columns for a specified set of codes. By default, the data.table is returned with new columns corresponding to boolean values, whether given group of diagnoses are present in the given diagnosis. If `collapse` is given, then the information is aggregated based-on the `collapse` column and the earliest of latest time of the given diagnosis is provided.

Usage

```r
convert_phy(
  d,
  code = "phy_code",
  code_type = "phy_code_type",
  codes_to_find = NULL,
  collapse = NULL,
  code_time = "time_phy",
  time_type = "earliest",
  nThread = 4
)
```

Arguments

- **d**: data.table, database containing health history information data loaded using the `load_phy` function.
- **code**: string, column name of the diagnosis code column. Defaults to `phy_code`.
- **code_type**: string, column name of the code_type column. Defaults to `phy_code_type`.
- **codes_to_find**: list, a list of string arrays corresponding to sets of code types and codes separated by `;`, i.e.: "LMR:3688". The function searches for the given health history code type and code pair and adds new boolean columns with the name of each list element. These columns are indicators whether any of the health history code type and code pair occurs in the set of codes.
- **collapse**: string, a column name on which to collapse the data.table. Used in case we wish to assess whether given health history codes are present within all the same instances of `collapse`. See vignette for details.
- **code_time**: string, column name of the time column. Defaults to `time_phy`. Used in case `collapse` is present to provide the earliest or latest instance of health history information.
- **time_type**: string, if multiple health histories are present within the same case of `collapse`, which timepoint to return. Supported are: "earliest" or "latest". Defaults to `earliest`. 
- **nThread**: integer, number of threads to use when processing data. Defaults to 4.
nThread

integer, number of threads to use by dopar for parallelization. If it is set to 1, then no parallel backends are created and the function is executed sequentially. On windows machines sockets are used, while on other operating systems fork parallelization is used.

Value

data.table, with indicator columns whether the any of the given health histories are reported. If collapse is present, then only unique ID and the summary columns are returned.

Examples

## Not run:
#Search for Height and Weight codes
anthropometrics <- list(Weight = c("LMR:3688", "EPIC:WGT"), Height = c("LMR:3771", "EPIC:HGT"))
data_phy_parse <- convert_phy(d = data_phy, codes_to_find = anthropometrics, nThread = 2)

#Search for for Height and Weight codes and summarize per patient providing earliest time
anthropometrics <- list(Weight = c("LMR:3688", "EPIC:WGT"), Height = c("LMR:3771", "EPIC:HGT"))
data_phy_parse <- convert_phy(d = data_phy, codes_to_find = anthropometrics, nThread = 2, collapse = "ID_MERGE", time_type = "earliest")

## End(Not run)

---

convert_prc

Searches procedures columns for given procedures

Description

Analyzes procedure data loaded using load_prc. Searches procedures columns for a specified set of procedures. By default, the data.table is returned with new columns corresponding to boolean values, whether given group of procedures are present in the given procedure. If collapse is given, then the information is aggregated based-on the collapse column and the earliest of latest time of the given procedure is provided.

Usage

convert_prc(
  d,
  code = "prc_code",
  code_type = "prc_code_type",
  codes_to_find = NULL,
  collapse = NULL,
  code_time = "time_prc",
  time_type = "earliest",
  nThread = 4
)
Arguments

d   data.table, database containing procedure information data loaded using the `load_prc`
       function.

code   string, column name of the procedure code column. Defaults to `prc_code`.

code_type   string, column name of the code_type column. Defaults to `prc_code_type`.

codes_to_find   list, a list of string arrays corresponding to sets of code types and codes separated
       by `;`, i.e.: "CPT:00104". The function searches for the given procedure code type
       and code pair and adds new boolean columns with the name of each list element.
       These columns are indicators whether any of the procedure code type and code
       pair occurs in the set of codes.

collapse   string, a column name on which to collapse the data.table. Used in case we wish
       to assess whether given disease codes are present within all the same instances
       of `collapse`. See vignette for details.

code_time   string, column name of the time column. Defaults to `time_prc`. Used in case
       collapse is present to provide the earliest or latest instance of diagnosing the
       given disease.

time_type   string, if multiple diagnoses are present within the same case of `collapse`, which
       timepoint to return. Supported are: "earliest" or "latest". Defaults to `earliest`.

nThread   integer, number of threads to use by `dopar` for parallelization. If it is set to 1,
       then no parallel backends are created and the function is executed sequentially.
       On windows machines sockets are used, while on other operating systems fork
       parallelization is used.

Value

data.table, with indicator columns whether the any of the given procedures are reported. If `collapse`
       is present, then only unique ID and the summary columns are returned.

Examples

```r
## Not run:
#Search for Anesthesia CPT codes
procedures <- list(Anesthesia = c("CPT:00410", "CPT:00104"))
data_prc_parse <- convert_prc(d = data_prc, codes_to_find = procedures, nThread = 2)

#Search for Anesthesia CPT codes
procedures <- list(Anesthesia = c("CPT:00410", "CPT:00104"))
data_prc_procedures <- convert_prc(d = data_prc, codes_to_find = procedures, nThread = 2, collapse = "ID_MERGE", time_type = "earliest")

## End(Not run)
```
**convert_rfv**

*Searches columns for given reason for visit defined by ERFV codes.*

**Description**

Analyzes reason for visit data loaded using `load_rfv`. If requested, the data.table is returned with new columns corresponding to boolean values, whether given group of ERFV are present in the given columns. If `collapse` is given, then the information is aggregated based-on the `collapse` column and the earliest of latest time of the given diagnosis is provided.

**Usage**

```r
call_rfv(
  d,
  code = "rfv_concept_id",
  codes_to_find = NULL,
  collapse = NULL,
  code_time = "time_rfv_start",
  time_type = "earliest",
  nThread = 4
)
```

**Arguments**

- `d` : data.table, database containing reason for visit information data loaded using the `load_rfv` function.
- `code` : string vector, an array of column names to search.
- `codes_to_find` : list, a list of arrays corresponding to sets of ERFV codes. The function searches the columns in code and the name of each list element will be created. These columns are indicators whether the given disease is present in the set of ERFV codes or not.
- `collapse` : string, a column name on which to collapse the data.table. Used in case we wish to assess whether given ERFV are present within all the same instances of `collapse`. See vignette for details.
- `code_time` : string, column name of the time column. Defaults to `time_rfv_start`. Used in case collapse is present to provide the earliest or latest instance of diagnosing the given disease.
- `time_type` : string, if multiple diagnoses are present within the same case of `collapse`, which timepoint to return. Supported are: "earliest" or "latest". Defaults to `earliest`.
- `nThread` : integer, number of threads to use by `dopar` for parallelization. If it is set to 1, then no parallel backends are created and the function is executed sequentially. On windows machines sockets are used, while on other operating systems fork parallelization is used.
Value

data.table, with indicator columns if provided. If `collapse` is present, then only unique ID and the summary columns are returned.

Examples

```
## Not run:
#Parse reason for visit columns
#and create indicator variables for the following reasons and summarize per patient,
#whether there are any encounters where the given reasons were registered
reasons <- list(Pain = c("ERFV:160357", "ERFV:140012"), Visit = c("ERFV:501"))
data_rfv_disease <- convert_rfv(d = data_rfv, keep = FALSE, 
codes_to_find = reasons, nThread = 2, collapse = "ID_MERGE")
```

create_img_db

Create a database of DICOM headers.

Description

The function creates a database of DICOM headers present in a folder structure. Each series should be in its own folder, but they can be in a nested folder structure. Files where there are also folder present next to them at the same level will not be parsed. That is the folder structure needs to comply with the DICOM standard. Be aware that the function requires python and pydicom to be installed!

The function cycles through all folders present in the provided path and recursively goes through them, every subfolder, and extracts the DICOM header information from the files using the dcmread function of the pydicom package. The extension of the files can be provided by the `ext` argument, as DICOM files may have different extensions then that of .dcm. Also, using the `all` boolean argument, you can specify whether the function provides output for each file, or only for the first file, which is beneficial if you are analyzing multi-slice series, as all instances have almost all the same header information. Furthermore, using the `keywords` argument you can manually specify which DICOM keywords you wish to extract. These need to be a valid keyword specified in the DICOM standard.

Usage

```
create_img_db(
  path,
  ext = c(".dcm", ",.dicom", ",.ima"),
  all = TRUE,
  keywords = c("StudyDate", "StudyTime", "SeriesDate", "SeriesTime", "AcquisitionDate", 
    "AcquisitionTime", "ConversionType", "Manufacturer", "InstitutionName", 
    "InstitutionalDepartmentName", "ReferringPhysicianName", "Modality", 
    "ManufacturerModelName", "StudyDescription", "SeriesDescription", "StudyComments", 
    "ProtocolName", "RequestedProcedureID", "ViewPosition", "StudyInstanceUID", 
    "SeriesInstanceUID", "SOPInstanceUID", "AccessionNumber", "PatientName", "PatientID", 
    "IssuerOfPatientID", "PatientBirthDate", "PatientSex", "PatientAge", 
```
nThread = 4,
na = TRUE,
identical = TRUE
)

Arguments

path string vector, full folder path to folder that contains the images.

ext string array, possible file extensions to parse. It is advised to add . before the extensions as the given character patterns may be present elsewhere in the file names. Furthermore, if DICOM files without an extension should also be parsed, then add "" to the extensions as then the script will try to read all files without an extension. Also, the file names and the extensions are converted to lower case before matching to avoid mismatches due to capitals.

all boolean, whether all files in a series should be parsed, or only the first one.

keywords string array, of valid DICOM keywords.

nThread integer, number of threads to use for parsing data.

na boolean, whether to remove columns with only NA values. Defaults to TRUE.

identical boolean, whether to remove columns with identical values. Defaults to TRUE.

Value
data.table, with DICOM header information return unchanged. However, the function also provides additional new columns which help further data manipulations, these are:

time_study POSIXct, StudyDate and StudyTime concatenated together to POSIXct.

time_series POSIXct, SeriesDate and SeriesTime concatenated together to POSIXct.

time_acquisition POSIXct, AcquisitionDate and AcquisitionTime concatenated together to POSIXct.

name_img string, PatientName with special characters removed.

time_date_of_birth_img POSIXct, PatientBirthDate as POSIXct.

img_pixel_spacing numeric, PixelSpacing value of the first element in the array returned as numerical value.

Examples

## Not run:
#Create a database with DICOM header information
all_dicom_headers <- create_img_db(path = "/Users/Test/Data/DICOM/")
all_dicom_headers <- create_img_db(path = "/Users/Test/Data/DICOM/", ext = c(".dcm", ".DICOM"))
#Create a database with DICOM header information for only IDs and accession numbers
all_dicom_headers <- create_img_db(path = "/Users/Test/Data/DICOM/",
keywords = c("PatientID", "AccessionNumber"))

## End(Not run)

data_con

**Example of processed con.txt output from RPDR using the load_con() function.**

**Description**

Result of a con.txt output from RPDR loaded into a data table in R using load_con().

**NOTE**: Due to potential issues with PHI and PPI, the example datasets can be downloaded from the Partners Gitlab repository under *parserpdr-sample-data*.

**Usage**

data_con

**Format**

data.table

**Value**

data table, with contact information data.

- **ID_MERGE** numeric, defined IDs by merge_id, used for merging later.
- **ID_con_EMPI** string, Unique Partners-wide identifier assigned to the patient used to consolidate patient information from con datasource, corresponds to EMPI in RPDR. Data is formatted using pretty_mrn().
- **ID_con_PMRN** string, Epic medical record number. This value is unique across Epic instances within the Partners network from con datasource, corresponds to EPIC_PMRN in RPDR. Data is formatted using pretty_mrn().
- **ID_con_loc** string, if mrn_type == TRUE, then the data in MRN_Type and MRN are parsed into IDs corresponding to locations. Data is formatted using pretty_mrn().
- **ID_con_loc_list** string, if prevalence of IDs in Patient_ID_List > perc, then they are included in the output. Data is formatted using pretty_mrn().
- **name_last** string, Patient’s last name, corresponds to Last_Name in RPDR.
- **name_first** string, Patient’s first name, corresponds to First_Name in RPDR.
- **name_middle** string, Patient’s middle name or initial, corresponds to Middle_Name in RPDR.
- **name_previous** string, Any alternate names on record for this patient, corresponds to Previous_Name in RPDR.
- **SSN** string, Social Security Number, corresponds to SSN in RPDR.
VIP character, Special patient statuses as defined by the EMPI group, corresponds to VIP in RPDR.

address1 string, Patient’s current address, corresponds to address1 in RPDR.

address2 string, Additional address information, corresponds to address2 in RPDR.

city string, City of residence, corresponds to City in RPDR.

state string, State of residence, corresponds to State in RPDR.

country_con string, Country of residence from con datasource, corresponds to Country in RPDR. Punctuation marks are removed.

zip_con numeric, Mailing zip code of primary residence from con datasource, corresponds to Zip in RPDR. Formatted to 5 character zip codes.

direct_contact_consent boolean, Indicates whether the patient has given permission to contact them directly through the RODY program, corresponds to Direct>Contact>Consent in RPDR.

phone_home number, Patient’s home phone number, corresponds to Home_Phone in RPDR. Formatted to 10 digit phone numbers.

phone_day number, Phone number where the patient can be reached during the day, corresponds to Day_Phone in RPDR. Formatted to 10 digit phone numbers.

insurance1 string, Patient’s primary health insurance carrier and subscriber ID information, corresponds to Insurance_1 in RPDR. Punctuation marks are removed.

insurance2 string, Patient’s secondary health insurance carrier and subscriber ID information, if any, corresponds to Insurance_2 in RPDR. Punctuation marks are removed.

insurance3 string, Patient’s tertiary health insurance carrier and subscriber ID information, if any, corresponds to Insurance_3 in RPDR. Punctuation marks are removed.

primary_care_physician string, Comma-delimited list of all primary care providers on record for this patient per institution, along with contact information (if available), corresponds to Primary_Care_Physician in RPDR. Punctuation marks are removed.

primary_care_physician_resident string, Comma-delimited list of any Resident primary care providers on record for this patient per institution, along with contact information (if available), corresponds to Resident_Primary_Care_Physician in RPDR. Punctuation marks are removed.

---

**data_con_raw**

Example of con.txt output from RPDR.

Description

A con.txt output from RPDR loaded into a data table in R using `data.table::fread()`.

**NOTE**: Due to potential issues with PHI and PPI, the example datasets can be downloaded from the Partners Gitlab repository under *parserpdr-sample-data*.

Usage

data_con_raw
Format

data.table

Value

data table, imported from con.txt

EMPI numeric, Unique Partners-wide identifier assigned to the patient used to consolidate patient information.

EPIC_PMRN numeric, Epic medical record number. This value is unique across Epic instances within the Partners network.

MRN_Type string, Indicates the institution associated with a specific MRN. This can appear as a comma-delimited list if MRNs from multiple Partners Health System institutions are available.

MRN string, Unique Medical Record Number for the site identified in the 'MRN_Type' field. This can appear as a comma-delimited list if multiple MRNs from Partners hospitals are available.

Last_Name string, Patient’s last name.

First_Name string, Patient’s first name.

Middle_Name string, Patient’s middle name or initial.

Direct_Contact_Consent boolean, Indicates whether the patient has given permission to contact them directly through the RODY program.

Address1 string, Patient’s current address.

Address2 string, Additional address information.

City string, City of residence.

State string, State of residence.

Zip numeric, Mailing zip code of primary residence from con datasource.

Country string, Country of residence from con datasource.

Home_Phone number, Patient’s home phone number.

Day_Phone number, Phone number where the patient can be reached during the day.

SSN string, Social Security Number.

VIP character, Special patient statuses as defined by the EMPI group.

Previous_Name string, Any alternate names on record for this patient.

Patient_ID_List string, Comma-delimited list of all hospital-specific identifiers on record for this patient.

Insurance_1 string, Patient’s primary health insurance carrier and subscriber ID information.

Insurance_2 string, Patient’s secondary health insurance carrier and subscriber ID information.

Insurance_3 string, Patient’s tertiary health insurance carrier and subscriber ID information.

Primary_Care_Physician string, Comma-delimited list of all primary care providers on record for this patient per institution, along with contact information (if available).

Resident_Primary_Care_Physician string, Comma-delimited list of any Resident primary care providers on record for this patient per institution, along with contact information (if available).
Example of processed dem.txt output from RPDR using the load_dem() function.

Description

Result of a dem.txt output from RPDR loaded into a data table in R using load_dem().

**NOTE**: Due to potential issues with PHI and PPI, the example datasets can be downloaded from the Partners Gitlab repository under *parserpdr-sample-data*.

Usage

data_dem

Format

data.table

Value

data table, with demographic information data.

- **ID_MERGE** numeric, defined IDs by merge_id, used for merging later.
- **ID_dem_EMPI** string, Unique Partners-wide identifier assigned to the patient used to consolidate patient information. from dem datasource, corresponds to EMPI in RPDR. Data is formatted using pretty_mrn().
- **ID_dem_PMRN** string, Epic medical record number. This value is unique across Epic instances within the Partners network. from dem datasource, corresponds to EPIC_PMRN in RPDR. Data is formatted using pretty_mrn().
- **ID_dem_loc** string, if mrn_type == TRUE, then the data in MRN_Type and MRN are parsed into IDs corresponding to locations. Data is formatted using pretty_mrn().
- **gender** string, Patient’s legal sex, corresponds to Gender in RPDR. Punctuation marks and white spaces are removed.
- **time_date_of_birth** POSIXct, Patient’s date of birth, corresponds to Date_of_Birth in RPDR. Converted to POSIXct format.
- **age** string, Patient’s current age (or age at death), corresponds to Age in RPDR.
- **language** string, Patient’s preferred spoken language, corresponds to Language in RPDR. Punctuation marks and white spaces are removed.
- **race** string, Patient’s primary race, corresponds to Race in RPDR. Punctuation marks and white spaces are removed.
- **marital** string, Patient’s current marital status, corresponds to Marital_Status in RPDR. Punctuation marks and white spaces are removed.
- **religion** string, Patient-identified religious preference, corresponds to Religion in RPDR. Punctuation marks and white spaces are removed.
**veteran** string, Patient’s current military veteran status, corresponds to `Is_a_veteran` in RPDR. Punctuation marks and white spaces are removed.

**country_dem** string, Patient’s current country of residence from dem datasource, corresponds to `Country` in RPDR. Punctuation marks and white spaces are removed.

**zip_dem** string, Mailing zip code of patient’s primary residence from dem datasource, corresponds to `Zip_code` in RPDR. Formatted to 5 character zip codes.

**vital_status** string, Identifies if the patient is living or deceased. This data is updated monthly from the Partners registration system and the Social Security Death Master Index, corresponds to `Vital_Status` in RPDR. Punctuation marks are removed.

**time_date_of_death** POSIXct, Recorded date of death from source in `Vital_Status`. Date of death information obtained solely from the Social Security Death Index will not be reported until 3 years after death due to privacy concerns. If the value is independently documented by a Partners entity within the 3 year window then the date will be displayed. corresponds to `Date_of_Death` in RPDR. Converted to POSIXct format.

---

**data_dem_raw**

Example of dem.txt output from RPDR.

---

**Description**

A dem.txt output from RPDR loaded into a data table in R using `data.table::fread()`.

**NOTE**: Due to potential issues with PHI and PPI, the example datasets can be downloaded from the Partners Gitlab repository under *parserpdr-sample-data*.

**Usage**

`data_dem_raw`

**Format**

`data.table`

**Value**

`data table, imported from dem.txt`

**EMPI** numeric, Unique Partners-wide identifier assigned to the patient used to consolidate patient information.

**EPIC_PMRN** numeric, Epic medical record number. This value is unique across Epic instances within the Partners network.

**MRN_Type** string, Indicates the institution associated with a specific MRN. This can appear as a comma-delimited list if MRNs from multiple Partners Health System institutions are available.

**MRN** string, Unique Medical Record Number for the site identified in the ‘MRN_Type’ field. This can appear as a comma-delimited list if multiple MRNs from Partners hospitals are available.

**Gender** string, Patient’s legal sex.
**Date_of_Birth** string, Patient’s date of birth.

**Age** string, Patient’s current age (or age at death).

**Language** string, Patient’s preferred spoken language.

**Race** string, Patient’s primary race.

**Marital_status** string, Patient’s current marital status.

**Religion** string, Patient-identified religious preference.

**Is_a_veteran** string, Patient’s current military veteran status.

**Zip_code** string, Mailing zip code of patient’s primary residence from dem datasource.

**Country** string, Patient’s current country of residence from dem datasource.

**Vital_status** string, Identifies if the patient is living or deceased. This data is updated monthly from the Partners registration system and the Social Security Death Master Index.

**Date_Of_Death** string, Recorded date of death from source in ‘Vital_Status’. Date of death information obtained solely from the Social Security Death Index will not be reported until 3 years after death due to privacy concerns. If the value is independently documented by a Partners entity within the 3 year window then the date will be displayed.

---

**data_dia**  
Example of processed dia.txt output from RPDR using the load_dia() function.

---

**Description**

Result of a dia.txt output from RPDR loaded into a data table in R using load_dia().

**NOTE**: Due to potential issues with PHI and PPI, the example datasets can be downloaded from the Partners Gitlab repository under *parserpdr-sample-data*.

**Usage**

data_dia

**Format**

data.table

**Value**

data table, with diagnoses information.

**ID_MERGE** numeric, defined IDs by merge_id, used for merging later.

**ID_dia_EMPI** string, Unique Partners-wide identifier assigned to the patient used to consolidate patient information from dia datasource, corresponds to EMPI in RPDR. Data is formatted using pretty_mrn().
**ID_dia_PMRN** string, Epic medical record number. This value is unique across Epic instances within the Partners network from *dia* datasource, corresponds to EPIC_PMRN in RPDR. Data is formatted using pretty_mrn().

**ID_dia_loc** string, if mrn_type == TRUE, then the data in *MRN_Type* and *MRN* are parsed into IDs corresponding to locations (loc). Data is formatted using pretty_mrn().

**time_dia** POSIXct, Date when the diagnosis was noted, corresponds to Date in RPDR. Converted to POSIXct format.

**dia_name** string, Name of the diagnosis, diagnosis-related group, or phenotype. For more information on available Phenotypes visit https://phenotypes.partners.org/phenotype_list.html, corresponds to Diagnosis_Name in RPDR.

**dia_code** string, Diagnosis, diagnosis-related group, or phenotype code, corresponds to Code in RPDR.

**dia_code_type** string, Standardized classification system or custom grouping associated with the diagnosis code, corresponds to Code_type in RPDR.

**dia_flag** string, Qualifier for the diagnosis, if any, corresponds to Diagnosis_flag in RPDR.

**dia_enc_num** string, Unique identifier of the record/visit. This values includes the source system, hospital, and a unique identifier within the source system, corresponds to Encounter_number in RPDR.

**dia_provider** string, Provider of record for the encounter where the diagnosis was entered, corresponds to Provider in RPDR. Punctuation marks are removed.

**dia_clinic** string, Specific department/location where the patient encounter took place, corresponds to Clinic in RPDR. Punctuation marks are removed.

**dia_hosp** string, Facility where the encounter occurred, corresponds to Hospital in RPDR.

**dia_inpatient** string, Identifies whether the diagnosis was noted during an inpatient or outpatient encounter, Punctuation marks are removed.

---

**data_dia_raw**

Example of *dia.txt* output from RPDR.

---

**Description**

A *dia.txt* output from RPDR loaded into a data table in R using `data.table::fread()`.

**NOTE**: Due to potential issues with PHI and PPI, the example datasets can be downloaded from the PPartners Gitlab repository under *parserpdr-sample-data*.

**Usage**

    data_dia_raw

**Format**

    data.table
Value

data table, imported from dia.txt

**EMPI** numeric, Unique Partners-wide identifier assigned to the patient used to consolidate patient information.

**EPIC_PMRN** numeric, Epic medical record number. This value is unique across Epic instances within the Partners network.

**MRN_Type** string, Indicates the institution associated with a specific MRN. This can appear as a comma-delimited list if MRNs from multiple Partners Health System institutions are available.

**MRN** string, Unique Medical Record Number for the site identified in the 'MRN_Type' field. This can appear as a comma-delimited list if multiple MRNs from Partners hospitals are available.

**Date** string, Date when the diagnosis was noted.

**Diagnosis_Name** string, Name of the diagnosis, diagnosis-related group, or phenotype. For more information on available Phenotypes visit https://phenotypes.partners.org/phenotype_list.html.

**Code_type** string, Standardized classification system or custom grouping associated with the diagnosis code.

**Code** string, IDiagnosis, diagnosis-related group, or phenotype code.

**Diagnosis_flag** string, Qualifier for the diagnosis, if any.

**Provider** string, Provider of record for the encounter where the diagnosis was entered.

**Clinic** string, Specific department/location where the patient encounter took place.

**Hospital** numeric, Facility where the encounter occurred.

**Inpatient_Outpatient** string, Identifies whether the diagnosis was noted during an inpatient or outpatient encounter.

**Encounter_number** string, Unique identifier of the record/visit. This values includes the source system, hospital, and a unique identifier within the source system.

| data_enc | Example of processed enc.txt output from RPDR using the load_enc() function. |

Description

Result of a enc.txt output from RPDR loaded into a data table in R using load_enc().

**NOTE**: Due to potential issues with PHI and PPI, the example datasets can be downloaded from the Partners Gitlab repository under *parserpdr-sample-data*.

Usage

data_enc

Format

data.table
**Value**

data table, with encounter information.

- **ID_MERGE** numeric, defined IDs by *merge_id*, used for merging later.
- **ID_enc_EMPI** string. Unique Partners-wide identifier assigned to the patient used to consolidate patient information from *enc* datasource, corresponds to EMPI in RPDR. Data is formatted using pretty_mrn().
- **ID_enc_PMRN** string, Epic medical record number. This value is unique across Epic instances within the Partners network from *enc* datasource, corresponds to EPIC_PMRN in RPDR. Data is formatted using pretty_mrn().
- **ID_enc_loc** string, if *mrn_type* == TRUE, then the data in *MRN_Type* and *MRN* are parsed into IDs corresponding to locations. Data is formatted using pretty_mrn().
- **enc_numb** string, Unique identifier of the record/visit. This values includes the source system, hospital, and a unique identifier within the source system, corresponds to Encounter_number in RPDR.
- **time_enc_admit** POSIXct, Date when the patient was admitted or entered the facility, corresponds to Admit_Date in RPDR. Converted to POSIXct format.
- **time_enc_disch** POSIXct, Date when the patient was discharged or left the facility, corresponds to Discharge_Date in RPDR. Converted to POSIXct format.
- **enc_status** string, Billing account-related notes about the encounter. This will not be populated for all encounters, corresponds to Encounter_Status in RPDR. Punctuation marks are removed.
- **enc_hosp** string, Facility where the encounter occurred, corresponds to Hospital in RPDR. Punctuation marks and white spaces are removed.
- **enc_inpatient** string, Classifies the type of encounter as either Inpatient or Outpatient. ED visits are currently classified under the 'Outpatient' label, corresponds to Inpatient_or_Outpatient in RPDR. Punctuation marks are removed.
- **enc_service** string, Hospital service line assigned to the encounter, corresponds to Service_Line in RPDR. Punctuation marks are removed.
- **enc_attending** string, The attending provider associated with the encounter. For Epic professional billing, this is the billing provider, corresponds to Attending_MD in RPDR. Punctuation marks are removed.
- **enc_length** numeric, Length of stay for the encounter, corresponds to LOS_days in RPDR.
- **enc_clinic** string, Specific department/location where the encounter occurred, corresponds to Clinic_Name in RPDR. Punctuation marks are removed.
- **enc_admit_src** string, Location where the patient was admitted when entering the hospital/clinic, corresponds to Admit_Source in RPDR. Punctuation marks are removed.
- **enc_pat_type** string, Provides information regarding the specific patient classifications and status of the patient visit. This field is only populated for McLean Hospital encounters, corresponds to Patient_Type in RPDR. Punctuation marks are removed.
- **enc_ref_disp** string, Location where the patient has been directed for treatment or follow-up by a staff member. This field is only populated for McLean Hospital encounters, corresponds to Referrer_Discipline in RPDR. Punctuation marks are removed.
- **enc_disch_disp** string, Patient’s anticipated location or status following the encounter, corresponds to Discharge_Disposition in RPDR. Punctuation marks are removed.
enc_pay string, Payors responsible for the hospital account. Multiple payors (primary, secondary, etc.) may be listed, corresponds to Payor in RPDR. Punctuation marks are removed.

enc_diag_admit string, Initial working diagnosis documented by the admitting or attending physician, corresponds to Admitting_Diagnosis in RPDR. Punctuation marks and white spaces are removed.

enc_diag_princ string, Condition established, after study, to be chiefly responsible for occasioning the admission of the patient to the hospital for care, corresponds to Principle_Diagnosis in RPDR. Punctuation marks and white spaces are removed.

enc_diag_1 string, Additional diagnoses associated with this encounter or visit, corresponds to Diagnosis_1 in RPDR. Punctuation marks and white spaces are removed.

enc_diag_2 string, Additional diagnoses associated with this encounter or visit, corresponds to Diagnosis_2 in RPDR. Punctuation marks and white spaces are removed.

enc_diag_3 string, Additional diagnoses associated with this encounter or visit, corresponds to Diagnosis_3 in RPDR. Punctuation marks and white spaces are removed.

enc_diag_4 string, Additional diagnoses associated with this encounter or visit, corresponds to Diagnosis_4 in RPDR. Punctuation marks and white spaces are removed.

enc_diag_5 string, Additional diagnoses associated with this encounter or visit, corresponds to Diagnosis_5 in RPDR. Punctuation marks and white spaces are removed.

enc_diag_6 string, Additional diagnoses associated with this encounter or visit, corresponds to Diagnosis_6 in RPDR. Punctuation marks and white spaces are removed.

enc_diag_7 string, Additional diagnoses associated with this encounter or visit, corresponds to Diagnosis_7 in RPDR. Punctuation marks and white spaces are removed.

enc_diag_8 string, Additional diagnoses associated with this encounter or visit, corresponds to Diagnosis_8 in RPDR. Punctuation marks and white spaces are removed.

enc_diag_9 string, Additional diagnoses associated with this encounter or visit, corresponds to Diagnosis_9 in RPDR. Punctuation marks and white spaces are removed.

enc_diag_10 string, Additional diagnoses associated with this encounter or visit, corresponds to Diagnosis_10 in RPDR. Punctuation marks and white spaces are removed.

enc_diag_group string, Diagnosis-Related Group for the encounter, in the following format: SYSTEM:CODE - Description, corresponds to DRG in RPDR. Punctuation marks and white spaces are removed.

data_enc_raw

---

Example of enc.txt output from RPDR.

---

Description

A enc.txt output from RPDR loaded into a data table in R using `data.table::fread()`.

**NOTE**: Due to potential issues with PHI and PPI, the example datasets can be downloaded from the Partners Gitlab repository under *parserpdr-sample-data*.

Usage

data_enc_raw
### Format

data.table

### Value

data table, imported from enc.txt

**EMPI** numeric, Unique Partners-wide identifier assigned to the patient used to consolidate patient information.

**EPIC_PMRN** numeric, Epic medical record number. This value is unique across Epic instances within the Partners network.

**MRN_Type** string, Indicates the institution associated with a specific MRN. This can appear as a comma-delimited list if MRNs from multiple Partners Health System institutions are available.

**MRN** string, Unique Medical Record Number for the site identified in the 'MRN_Type' field. This can appear as a comma-delimited list if multiple MRNs from Partners hospitals are available.

**Encounter_number** string, Unique identifier of the record/visit. This values includes the source system, hospital, and a unique identifier within the source system.

**Encounter_Status** string, Billing account-related notes about the encounter. This will not be populated for all encounters.

**Hospital** string, Facility where the encounter occurred.

**Inpatient_Outpatient** string, Classifies the type of encounter as either Inpatient or Outpatient. ED visits are currently classified under the 'Outpatient' label.

**Service_Line** string, Hospital service line assigned to the encounter.

**Attending_MD** string, The attending provider associated with the encounter. For Epic professional billing, this is the billing provider.

**Admit_Date** string, Date when the patient was admitted or entered the facility.

**Discharge_Date** string, Date when the patient was discharged or left the facility.

**LOS_Days** numeric, Length of stay for the encounter.

**Clinic_Name** string, Specific department/location where the encounter occurred.

**Admit_Source** string, Location where the patient was admitted when entering the hospital/clinic.

**Discharge_Disposition** string, Patient’s anticipated location or status following the encounter.

**Payor** string, Payors responsible for the hospital account. Multiple payors (primary, secondary, etc.) may be listed.

**Admitting_Diagnosis** string, Initial working diagnosis documented by the admitting or attending physician.

**Principal_Diagnosis** string, Condition established, after study, to be chiefly responsible for occasioning the admission of the patient to the hospital for care.

**Diagnosis_1** string, Additional diagnoses associated with this encounter or visit, corresponds to Diagnosis_1 in RPDR.

**Diagnosis_2** string, Additional diagnoses associated with this encounter or visit, corresponds to Diagnosis_2 in RPDR.
**Diagnosis_3** string, Additional diagnoses associated with this encounter or visit, corresponds to Diagnosis_3 in RPDR.

**Diagnosis_4** string, Additional diagnoses associated with this encounter or visit, corresponds to Diagnosis_4 in RPDR.

**Diagnosis_5** string, Additional diagnoses associated with this encounter or visit, corresponds to Diagnosis_5 in RPDR.

**Diagnosis_6** string, Additional diagnoses associated with this encounter or visit, corresponds to Diagnosis_6 in RPDR.

**Diagnosis_7** string, Additional diagnoses associated with this encounter or visit, corresponds to Diagnosis_7 in RPDR.

**Diagnosis_8** string, Additional diagnoses associated with this encounter or visit, corresponds to Diagnosis_8 in RPDR.

**Diagnosis_9** string, Additional diagnoses associated with this encounter or visit, corresponds to Diagnosis_9 in RPDR.

**Diagnosis_10** string, Additional diagnoses associated with this encounter or visit, corresponds to Diagnosis_10 in RPDR.

**DRG** string, Diagnosis-Related Group for the encounter, in the following format: SYSTEM:CODE
- Description.

**Patient_Type** string, Provides information regarding the specific patient classifications and status of the patient visit. This field is only populated for McLean Hospital encounters.

**Referrer_Discipline** string, Location where the patient has been directed for treatment or follow-up by a staff member. This field is only populated for McLean Hospital encounters.

---

**data_lab**

Example of processed `lab.txt` output from RPDR using the `load_lab()` function.

---

**Description**

Result of a `lab.txt` output from RPDR loaded into a data table in R using `load_lab()`.

**NOTE**: Due to potential issues with PHI and PPI, the example datasets can be downloaded from the Partners Gitlab repository under **parserpdr-sample-data**.

**Usage**

```
data_lab
```

**Format**

```
data.table
```
**Value**

data table, with laboratory exam information.

**ID_MERGE** numeric, defined IDs by `merge_id`, used for merging later.

**ID_lab_EMPI** string. Unique Partners-wide identifier assigned to the patient used to consolidate patient information from `lab` datasource, corresponds to EMPI in RPDR. Data is formatted using `pretty_mrn()`.

**ID_lab_PMNR** string, Epic medical record number. This value is unique across Epic instances within the Partners network from `lab` datasource, corresponds to EPIC_PMNR in RPDR. Data is formatted using `pretty_mrn()`.

**ID_lab_loc** string, if mrn_type == TRUE, then the data in `MRN_Type` and `MRN` are parsed into IDs corresponding to locations. Data is formatted using `pretty_mrn()`.

**time_lab_result** POSIXct, Date when the specimen was collected, corresponds to Seq_Date_Time in RPDR. Converted to POSIXct format.

**lab_group** string, Higher-level grouping concept used to consolidate similar tests across hospitals, corresponds to Group_ID in RPDR.

**lab_loinc** string, Standardized LOINC code for the laboratory test, corresponds to Loinc_Code in RPDR.

**lab_testID** string, Internal identifier for the test used by the source system, corresponds to Test_ID in RPDR.

**lab_descript** string, Name of the lab test, corresponds to Test_Description in RPDR.

**lab_result** string, Result value for the test, corresponds to Result in RPDR. Punctuation marks and white spaces are removed.

**lab_result_txt** string, Additional information included with the result. This can include instructions for interpretation or comments from the laboratory, corresponds to Result_Text in RPDR. Punctuation marks and white spaces are removed.

**lab_result_abn** string, Flag for identifying if values are outside of normal ranges or represent a significant deviation from previous values, corresponds to Abnormal_Flag in RPDR. Punctuation marks and white spaces are removed.

**lab_result_unit** string, Units associated with the result value, corresponds to Reference_Unit in RPDR. Punctuation marks and white spaces are removed.

**lab_result_range** string, Normal or therapeutic range for this value, corresponds to Reference_Range in RPDR. Punctuation marks and white spaces are removed.

**lab_result_toxic** string, Reference range of values defined as being toxic to the patient, corresponds to Toxic_Range in RPDR. Punctuation marks and white spaces are removed.

**lab_spec** string, Type of specimen collected to perform the test, corresponds to Specimen_Type in RPDR. Punctuation marks and white spaces are removed.

**lab_spec_txt** string, Free-text information about the specimen, its collection or its integrity, corresponds to Specimen_Text in RPDR. Punctuation marks and white spaces are removed.

**lab_correction** string, Free-text information about any changes made to the results, corresponds to Correction_Flag in RPDR. Punctuation marks and white spaces are removed.

**lab_status** string, Flag which indicates whether the procedure is pending or complete, corresponds to Test_Status in RPDR. Punctuation marks and white spaces are removed.
### data_lab_raw

- **lab_ord_pys** string, Name of the ordering physician, corresponds to Ordering_Doc in RPDR. Punctuation marks are removed.
- **lab_accession** string, Internal tracking number assigned to the specimen for identification in the lab, corresponds to Accession in RPDR.
- **lab_source** string, Database source, either CDR (Clinical Data Repository) or RPDR (internal RPDR database), corresponds to Source in RPDR. Punctuation marks and white spaces are removed.

---

**Example of lab.txt output from RPDR.**

---

### Description

A lab.txt output from RPDR loaded into a data table in R using `data.table::fread()`.

**NOTE**: Due to potential issues with PHI and PPI, the example datasets can be downloaded from the Partners Gitlab repository under *parserpdr-sample-data*.

### Usage

```r
data(data.lab_raw)
```

### Format

`data.table`

### Value

- **EMPI** numeric, Unique Partners-wide identifier assigned to the patient used to consolidate patient information.
- **EPIC_PMRN** numeric, Epic medical record number. This value is unique across Epic instances within the Partners network.
- **MRN_Type** string, Indicates the institution associated with a specific MRN. This can appear as a comma-delimited list if MRNs from multiple Partners Health System institutions are available.
- **MRN** string, Unique Medical Record Number for the site identified in the 'MRN_Type' field. This can appear as a comma-delimited list if multiple MRNs from Partners hospitals are available.
- **Seq_Date_Time** string, Date when the specimen was collected.
- **Group_Id** string, Higher-level grouping concept used to consolidate similar tests across hospitals.
- **Loinc_Code** string, Standardized LOINC code for the laboratory test.
- **Test_Id** string, Internal identifier for the test used by the source system.
- **Test_Description** string, Name of the lab test.
- **Result** string, Result value for the test.
Result_Text string, Additional information included with the result. This can include instructions for interpretation or comments from the laboratory.

Abnormal_Flag string, Flag for identifying if values are outside of normal ranges or represent a significant deviation from previous values.

Reference_Units numeric, Units associated with the result value.

Reference_Range string, Normal or therapeutic range for this value.

Toxic_Range string, Reference range of values defined as being toxic to the patient.

Specimen_Type string, Type of specimen collected to perform the test.

Specimen_Text string, Free-text information about the specimen, its collection or its integrity.

Correction_Flag string, Free-text information about any changes made to the results.

Test_Status string, Flag which indicates whether the procedure is pending or complete.

Ordering_Doc_Name string, Name of the ordering physician.

Accession string, Internal tracking number assigned to the specimen for identification in the lab.

Source string, Database source, either CDR (Clinical Data Repository) or RPDR (internal RPDR database).

---

**data_med**

Example of processed med.txt output from RPDR using the load_med() function.

---

Description

Result of a med.txt output from RPDR loaded into a data table in R using load_med().

**NOTE**: Due to potential issues with PHI and PPI, the example datasets can be downloaded from the Partners Gitlab repository under *parserpdr-sample-data*.

Usage

data_med

Format

data.table

Value

data table, with encounter information.

ID_MERGE numeric, defined IDs by merge_id, used for merging later.

ID_med_EMPI string, Unique Partners-wide identifier assigned to the patient used to consolidate patient information from enc datasource, corresponds to EMPI in RPDR. Data is formatted using pretty_mrn().
**ID_med_PMRN** string, Epic medical record number. This value is unique across Epic instances within the Partners network from `enc` datasource, corresponds to `EPIC_PMRN` in RPDR. Data is formatted using `pretty_mrn()`.

**ID_med_loc** string, if `mrn_type` == TRUE, then the data in `MRN_Type` and `MRN` are parsed into IDs corresponding to locations (`loc`). Data is formatted using `pretty_mrn()`.

**med_enc_numb** string, Unique identifier of the record/visit, displayed in the following format: Source System - Institution Number, corresponds to `Encounter_number` in RPDR.

**time_med** POSIXct, Completion status of the requested test/transfusion. Converted to POSIXct format, corresponds to `Medication_Date` in RPDR.

**time_med_detail** string, To clarify when patients may have stopped taking a medication, this column provides the statuses of 'Listed' or 'Removed'. This is provided on pre-Epic (LMR) medication dates (1997-2017). The 'Listed' value denotes that a medication was on the patient’s medication list on the date indicated. The ‘Removed’ value denotes that a medication was removed from a patient’s medication list on the date indicated. Corresponds to `Medication_Date_Detail` in RPDR.

**med** string, Name of the medication. This may be appended with the source system in the case of OnCall and LMR medications, corresponds to Medication in RPDR.

**med_code** string, Medication code associated with the "Code_type" value, corresponds to Code in RPDR.

**med_code_type** string, Standardized classification system or custom source value used to identify the medication, corresponds to `Code_Type` in RPDR.

**med_quant** string, Number of units of the medication ordered, corresponds to Quantity in RPDR.

**med_prov** string, Ordering provider for the medication, corresponds to Provider in RPDR. Punctuation marks are removed.

**med_clinic** string, Specific department/location where the medication was ordered or administered, corresponds to Clinic in RPDR. Punctuation marks are removed.

**med_hosp** string, Facility where the medication was ordered or administered, corresponds to Hospital in RPDR.

**med_inpatient** string, Identifies whether the medication was ordered with an Inpatient or Outpatient indication, corresponds to `Inpatient_Outpatient` in RPDR. Punctuation marks are removed.

**med_add_info** string, Additional administration information about the medication, corresponds to Additional_Info in RPDR.

---

**data_med_raw**

**Example of med.txt output from RPDR.**

---

**Description**

A med.txt output from RPDR loaded into a data table in R using `data.table::fread()`.

**NOTE**: Due to potential issues with PHI and PPI, the example datasets can be downloaded from the Partners Gitlab repository under *parserpdr-sample-data*.
Usage
data_med_raw

Format
data.table

Value
data table, imported from med.txt

EMPI numeric, Unique Partners-wide identifier assigned to the patient used to consolidate patient information.

EPIC_PMRN numeric, Epic medical record number. This value is unique across Epic instances within the Partners network.

MRN_Type string, Indicates the institution associated with a specific MRN. This can appear as a comma-delimited list if MRNs from multiple Partners Health System institutions are available.

MRN string, Unique Medical Record Number for the site identified in the 'MRN_Type' field. This can appear as a comma-delimited list if multiple MRNs from Partners hospitals are available.

Medication_Date string, Completion status of the requested test/transfusion.

Medication_Date_Detail string, To clarify when patients may have stopped taking a medication, this column provides the statuses of 'Listed' or 'Removed'. This is provided on pre-Epic (LMR) medication dates (1997-2017). The 'Listed' value denotes that a medication was on the patient’s medication list on the date indicated. The 'Removed' value denotes that a medication was removed from a patient’s medication list on the date indicated.

Medication string, Name of the medication. This may be appended with the source system in the case of OnCall and LMR medications.

Code_Type boolean, Standardized classification system or custom source value used to identify the medication.

Code string, Medication code associated with the "Code_type" value.

Quantity string, Number of units of the medication ordered.

Provider string, Ordering provider for the medication.

Clinic string, Specific department/location where the medication was ordered or administered.

Hospital numeric, Facility where the medication was ordered or administered.

Inpatient_Outpatient string, Identifies whether the medication was ordered with an Inpatient or Outpatient indication.

Encounter_number number, Unique identifier of the record/visit, displayed in the following format: Source System - Institution Number.

Additional_Info number, Additional administration information about the medication.
| data_mrn | Example of processed mrn.txt output from RPDR using the load_mrn() function. |

**Description**

Result of a mrn.txt output from RPDR loaded into a data table in R using `load_mrn()`.

**NOTE**: Due to potential issues with PHI and PPI, the example datasets can be downloaded from the Partners Gitlab repository under *parserpdr-sample-data*.

**Usage**

data_mrn

**Format**

data.table

**Value**

data table, with MRN data.

- **ID_MERGE** numeric, defined IDs by `merge_id`, used for merging later.
- **ID_mrn_INCOMING** string, Patient identifier, usually the EMPI, corresponds to IncomingId in RPDR. Data is formatted using pretty_mrn().
- **ID_mrn_INCOMING_SITE** string, Source of identifier, e.g. EMP for Enterprise Master Patient Index, MGH for Mass General Hospital, corresponds to IncomingSite in RPDR.
- **ID_mrn_PMRN** string, Epic medical record number. This value is unique across Epic instances within the Partners network, corresponds to EPIC_PMRN in RPDR. Data is formatted using pretty_mrn().
- **ID_mrn_EMPI** string, Unique Partners-wide identifier assigned to the patient used to consolidate patient information, corresponds to Enterprise_Master_Patient_Index in RPDR. Data is formatted using pretty_mrn().
- **ID_mrn_MGH** string, Unique Medical Record Number for Mass General Hospital, corresponds to MGH_MRN in RPDR. Data is formatted using pretty_mrn().
- **ID_mrn_BWH** string, Unique Medical Record Number for Brigham and Women’s Hospital, corresponds to BWH_MRN in RPDR. Data is formatted using pretty_mrn().
- **ID_mrn_FH** string, Unique Medical Record Number for Faulkner Hospital, corresponds to FH_MRN in RPDR. Data is formatted using pretty_mrn().
- **ID_mrn_SRH** string, Unique Medical Record Number for Spaulding Rehabilitation Hospital, corresponds to SRH_MRN in RPDR. Data is formatted using pretty_mrn().
- **ID_mrn_NWH** string, Unique Medical Record Number for Newton-Wellesley Hospital, corresponds to NWH_MRN in RPDR. Data is formatted using pretty_mrn().
ID_mrn_NSNC string, Unique Medical Record Number for North Shore Medical Center, corresponds to NSMC_MRN in RPDR. Data is formatted using pretty_mrn().

ID_mrn_MCL string, Unique Medical Record Number for McLean Hospital, corresponds to MCL_MRN in RPDR. Data is formatted using pretty_mrn().

ID_mrn_MEE string, Unique Medical Record Number for Mass Eye and Ear, corresponds to MEE_MRN in RPDR. Data is formatted using pretty_mrn().

ID_mrn_DFC string, Unique Medical Record Number for Dana Farber Cancer center, corresponds to DFC_MRN in RPDR. Data is formatted using pretty_mrn().

ID_mrn_WDH string, Unique Medical Record Number for Wentworth-Douglass Hospital, corresponds to WDH_MRN in RPDR. Data is formatted using pretty_mrn().

ID_mrn_STATUS string, Status of the record, corresponds to Status in RPDR.

Description
A mrn.txt output from RPDR loaded into a data table in R using data.table::fread().

**NOTE**: Due to potential issues with PHI and PPI, the example datasets can be downloaded from the Partners Gitlab repository under *parserpdr-sample-data*.

Usage

data_mrn_raw

Format

data.table

Value
data table, imported from mrn.txt

IncomingId numeric, Patient identifier, usually the EMPI.

IncomingSite string, Source of identifier, e.g. EMP for Enterprise Master Patient Index, MGH for Mass General Hospital.

Status string, Status of the record.

Enterprise_Master_Patient_Index numeric, Unique Partners-wide identifier assigned to the patient used to consolidate patient information.

EPIC_PMRN numeric, Epic medical record number. This value is unique across Epic instances within the Partners network.

MGH_MRN numeric, Unique Medical Record Number for Mass General Hospital.

BWH_MRN numeric, Unique Medical Record Number for Brigham and Women’s Hospital.

FH_MRN numeric, Unique Medical Record Number for Faulkner Hospital.
SRH_MRN numeric, Unique Medical Record Number for Spaulding Rehabilitation Hospital.
NWH_MRN numeric, Unique Medical Record Number for Newton-Wellesley Hospital.
NSMC_MRN numeric, Unique Medical Record Number for North Shore Medical Center.
MCL_MRN numeric, Unique Medical Record Number for McLean Hospital.
MEE_MRN numeric, Unique Medical Record Number for Mass Eye and Ear.
DFC_MRN numeric, Unique Medical Record Number for Dana Farber Cancer center.
WDH_MRN numeric, Unique Medical Record Number for Wentworth-Douglass Hospital.

<table>
<thead>
<tr>
<th>data_rad</th>
<th>Example of processed rad.txt output from RPDR using the load_rad() function.</th>
</tr>
</thead>
</table>

**Description**

Result of a rad.txt output from RPDR loaded into a data table in R using `load_rad()`.

**NOTE**: Due to potential issues with PHI and PPI, the example datasets can be downloaded from the Partners Gitlab repository under *parserpdr-sample-data*.

**Usage**

`data_rad`

**Format**

`data.table`

**Value**

Data table, with radiological notes information.
- **ID_MERGE** numeric, defined IDs by `merge_id`, used for merging later.
- **ID_rad_EMPI** string, Unique Partners-wide identifier assigned to the patient used to consolidate patient information from `rad` datasource, corresponds to EMPI in RPDR. Data is formatted using `pretty_mrn()`.
- **ID_rad_PMRN** string, Epic medical record number. This value is unique across Epic instances within the Partners network from `rad` datasource, corresponds to EPIC_PMRN in RPDR. Data is formatted using `pretty_mrn()`.
- **ID_rad_loc** string, if `mrn_type == TRUE`, then the data in `MRN_Type` and `MRN` are parsed into IDs corresponding to locations (`loc`). Data is formatted using `pretty_mrn()`.
- **time_rad_exam** POSIXct, Date when the report was filed, corresponds to Report_Date_Time in RPDR. Converted to POSIXct format.
- **rad_rep_num** string, Source-specific identifier used to reference the report, corresponds to Report_Number in RPDR.
**data_rad_raw**

- **rad_rep_desc** string, Type of procedure detailed in the report, corresponds to Report_Description in RPDR.
- **rad_rep_status** string, Completion status of the note/report, corresponds to Report_Status in RPDR.
- **rad_rep_type** string, This will always default to RAD, corresponds to Report_Type in RPDR.
- **rad_rep_txt** string, Full narrative text contained in the note/report, corresponds to Report_Text in RPDR.

---

**Description**

A rad.txt output from RPDR loaded into a data table in R using `data.table::fread()`.

**NOTE**: Due to potential issues with PHI and PPI, the example datasets can be downloaded from the Partners Gitlab repository under "parserpdr-sample-data".

**Usage**

data_rad_raw

**Format**

data.table

**Value**

data table, imported from rad.txt

- **EMPI** numeric, Unique Partners-wide identifier assigned to the patient used to consolidate patient information.
- **EPIC_PMRN** numeric, Epic medical record number. This value is unique across Epic instances within the Partners network.
- **MRN_Type** string, Indicates the institution associated with a specific MRN. This can appear as a comma-delimited list if MRNs from multiple Partners Health System institutions are available.
- **MRN** string, Unique Medical Record Number for the site identified in the 'MRN_Type' field. This can appear as a comma-delimited list if multiple MRNs from Partners hospitals are available.
- **Report_Number** string, Source-specific identifier used to reference the report.
- **Report_Date_Time** string, Date when the report was filed.
- **Report_Description** string, Type of procedure detailed in the report.
- **Report_Status** string, Completion status of the note/report.
- **Report_Type** string, This will always default to RAD.
- **Report_Text** string, Full narrative text contained in the note/report.
Example of processed rdt.txt output from RPDR using the load_rdt() function.

Description

Result of a rdt.txt output from RPDR loaded into a data table in R using load_rdt().

**NOTE**: Due to potential issues with PHI and PPI, the example datasets can be downloaded from the Partners Gitlab repository under *parserpdr-sample-data*.

Usage

data_rdt

Format

data.table

Value

data table, with radiological exam information.

- **ID_MERGE** numeric, defined IDs by merge_id, used for merging later.
- **ID_rdt_EMPI** string, Unique Partners-wide identifier assigned to the patient used to consolidate patient information from rdt datasource, corresponds to EMPI in RPDR. Data is formatted using pretty_mrn().
- **ID_rdt_PMRN** string, Epic medical record number. This value is unique across Epic instances within the Partners network from rdt datasource, corresponds to EPIC_PMRN in RPDR. Data is formatted using pretty_mrn().
- **ID_con_loc** string, if mrn_type == TRUE, then the data in MRN_Type and MRN are parsed into IDs corresponding to locations. Data is formatted using pretty_mrn().
- **time_rad_exam** POSIXct, Date of the radiology exam, corresponds to Date in RPDR. Converted to POSIXct format.
- **rad_mode** string, Modality of the exam, corresponds to Mode in RPDR. Punctuation marks and white spaces are removed.
- **rad_group** string, Higher-level grouping concept used to consolidate similar procedures across hospitals, corresponds to Group in RPDR. Punctuation marks and white spaces are removed.
- **rad_test_code** string, Internal identifier for the procedure used by the source system, corresponds to Test_Code in RPDR.
- **rad_test_desc** string, Full name of the exam/study performed, corresponds to Test_Description in RPDR.
- **rad_accession** string, Identifier assigned to the report or procedure for Radiology tracking purposes, corresponds to Accession_Number in RPDR.
**rad_provider** string, Ordering or authorizing provider for the study, corresponds to Provider in RPDR. Punctuation marks are removed.

**rad_clinic** string, Specific department/location where the procedure was ordered or performed, corresponds to Clinic in RPDR. Punctuation marks are removed.

**rad_hosp** string, Facility where the order was entered, corresponds to Hospital in RPDR. Punctuation marks and white spaces are removed.

**rad_inpatient** string, Classifies the type of encounter where the procedure was performed, corresponds to Inpatient_Outpatient in RPDR. Punctuation marks are removed.

---

**data_rdt_raw**

Example of rdt.txt output from RPDR.

---

### Description

A rdt.txt output from RPDR loaded into a data table in R using `data.table::fread()`.

**NOTE**: Due to potential issues with PHI and PPI, the example datasets can be downloaded from the Partners Gitlab repository under *parserpdr-sample-data*.

### Usage

```r
data_rdt_raw
```

### Format

`data.table`

### Value

data table, imported from rdt.txt

- **EMPI** numeric, Unique Partners-wide identifier assigned to the patient used to consolidate patient information.

- **EPIC_PMRN** numeric, Epic medical record number. This value is unique across Epic instances within the Partners network.

- **MRN_Type** string, Indicates the institution associated with a specific MRN. This can appear as a comma-delimited list if MRNs from multiple Partners Health System institutions are available.

- **MRN** string, Unique Medical Record Number for the site identified in the ‘MRN_Type’ field. This can appear as a comma-delimited list if multiple MRNs from Partners hospitals are available.

- **Date** string, Date of the radiology exam.

- **Mode** string, Modality of the exam.

- **Group** string, Higher-level grouping concept used to consolidate similar procedures across hospitals.

- **Test_Code** string, Internal identifier for the procedure used by the source system.

- **Test_Description** string, Full name of the exam/study performed.
**Accession_Number**  string. Identifier assigned to the report or procedure for Radiology tracking purposes.

**Provider**  string. Ordering or authorizing provider for the study.

**Clinic**  string. Specific department/location where the procedure was ordered or performed.

**Hospital**  numeric. Facility where the order was entered.

**Inpatient_Outpatient**  string. Classifies the type of encounter where the procedure was performed.

---

**find_exam**  
*Find exam data within a given timeframe using parallel CPU computing and possibly shared RAM management.*

---

**Description**

Finds all, earliest or closest examination to a given timepoints using parallel computing

**Usage**

```r
find_exam(
  d_from,
  d_to,
  d_from_ID = "ID_MERGE",
  d_to_ID = "ID_MERGE",
  d_from_time = "time_rad_exam",
  d_to_time = "time_enc_admit",
  time_diff_name = "timediff_exam_to_db",
  before = TRUE,
  after = TRUE,
  time = 1,
  time_unit = "days",
  multiple = "closest",
  add_column = NULL,
  keep_data = FALSE,
  nThread = 4,
  shared_RAM = FALSE
)
```

**Arguments**

- **d_from**  data table, the database which is searched to find examinations within the timeframe.
- **d_to**  data table, the database to which we wish to find examinations within the timeframe.
- **d_from_ID**  string, column name of the patient ID column in d_from. Defaults to ID_MERGE.
- **d_to_ID**  string, column name of the patient ID column in d_to. Defaults to ID_MERGE.
- **d_from_time**  string, column name of the time variable column in d_from. Defaults to time_rad_exam.
**find_exam**

- **d_to_time** string, column name of the time variable column in d_to. Defaults to time_enc_admit.
- **time_diff_name** string, column name of the new column created which holds the time difference between the exam and the time provided by d_to. Defaults to timediff_exam_to_db.
- **before** boolean, should times before the given time be considered. Defaults to TRUE.
- **after** boolean, should times after the given time be considered. Defaults to TRUE.
- **time** integer, the timeframe considered between the exam and the d_to timepoints. Defaults to 1.
- **time_unit** string, the unit of time used. Time variables are in d_to and d_from are truncated to the supplied time unit. For example: "2005-09-18 08:15:01 PDT" would be truncated to "2005-09-18 PDT" if time_unit is set to days. Then the time differences is calculated using difftime passing the argument to units. The following time units are supported: "secs", "mins", "hours", "days", "months" and "years" are supported. Defaults to days.
- **multiple** string, which exams to give back. closest gives back the exam closest to the time provided by d_to. all gives back all occurrences within the timeframe. earliest the earliest exam within the timeframe. In case of ties for closest or earliest, all are returned. Defaults to closest.
- **add_column** string, a column name in d_to to add to the output. Defaults to NULL.
- **keep_data** boolean, whether to include empty rows with only the d_from_ID column filled out for cases that have data in the d_from, but not within the time range. Defaults to FALSE.
- **nThread** integer, number of threads to use by dopar for parallelization. If it is set to 1, then no parallel backends are created and the function is executed sequentially. On windows machines sockets are used, while on other operating systems fork parallelization is used.
- **shared_RAM** boolean, whether to use shared memory during parallelization using the bigmemory package. This allows to process d_from and/or d_to datasets with >1M rows. Be aware that shared RAM usually results in slower run times, therefore by default it is set to FALSE, but it allows to run more threads when the datasets are large providing overall faster run times. Be aware that the optimal number of clusters might be different setting it to TRUE or FALSE, and this has to be determined empirically per machine.

**Value**

data table, with d_from filtered to ones only within the timeframe. The columns of d_from are returned with the corresponding time column in data_to where the rows are instances which comply with the time constraints specified by the function. An additional column specified in time_diff_name is also returned, which shows the time difference between the time column in d_from and d_to for that given case. Also the time column from d_to specified by d_to_time is returned under the name of time_to_db. An additional column specified in add_column may be added from data_to to the data table.
Examples

```r
# Not run:
# Filter encounters for first emergency visits at one of MGH's ED departments
data_enc_ED <- data_enc[enc_clinic == "MGH EMERGENCY 10020010608"]
data_enc_ED <- data_enc_ED[!duplicated(data_enc_ED$ID_MERGE)]

# Find all radiological examinations within 3 day of the ED registration
day_to <- find_exam(d_from = data_rdt, d_to = data_enc_ED,
d_from_ID = "ID_MERGE", d_to_ID = "ID_MERGE",
d_from_time = "time_rdt_exam", d_to_time = "time_enc_admit", time_diff_name = "time_diff_ED_rdt",
before = TRUE, after = TRUE, time = 3, time_unit = "days", multiple = "all",
nThread = 2, shared_RAM = FALSE)

# Find earliest radiological examinations within 3 days of the ED registration
day_to <- find_exam(d_from = data_rdt, d_to = data_enc_ED,
d_from_ID = "ID_MERGE", d_to_ID = "ID_MERGE",
d_from_time = "time_rdt_exam", d_to_time = "time_enc_admit", time_diff_name = "time_diff_ED_rdt",
before = TRUE, after = TRUE, time = 3, time_unit = "days", multiple = "earliest",
nThread = 2, shared_RAM = FALSE)

# Find closest radiological examinations on or after 1 day of the ED registration
# and add primary diagnosis column from encounters
day_to <- find_exam(d_from = data_rdt, d_to = data_enc_ED,
d_from_ID = "ID_MERGE", d_to_ID = "ID_MERGE",
d_from_time = "time_rdt_exam", d_to_time = "time_enc_admit", time_diff_name = "time_diff_ED_rdt",
before = FALSE, after = TRUE, time = 1, time_unit = "days", multiple = "earliest",
add_column = "enc_diag_princ", nThread = 2, shared_RAM = FALSE)

# Find closest radiological examinations on or after 1 day of the ED registration
# but also provide empty rows for patients with exam data but not within the timeframe
day_to <- find_exam(d_from = data_rdt, d_to = data_enc_ED,
d_from_ID = "ID_MERGE", d_to_ID = "ID_MERGE",
d_from_time = "time_rdt_exam", d_to_time = "time_enc_admit", time_diff_name = "time_diff_ED_rdt",
before = FALSE, after = TRUE, time = 1, time_unit = "days", multiple = "earliest",
add_column = "enc_diag_princ", keep_data = TRUE nThread = 2, shared_RAM = FALSE)

## End(Not run)
```

load_all  

*Loads all RPDR text outputs into R.*

Description

Loads all RPDR text outputs into R and returns a list of data tables processed. Currently supported outputs are: *Mrn.txt, Con.txt, Dem.txt, Enc.txt, Rdt.txt, Lab.txt, Med.txt, Dia.txt, Rfv.txt, Prc.txt, Car.txt, Dis.txt, End.txt, Hnp.txt, Opn.txt, Pat.txt, Prv.txt, Pul.txt, Rad.txt and Vis.txt.* If multiple text files of the same type are available (if the query is larger than 25000 patients), then add a "_" and a number to merge the same data sources into a single output in the order of the provided number.
load_all

Usage

load_all(
  folder,
  old_dem = FALSE,
  merge_id = "EMPI",
  sep = ":",
  id_length = "standard",
  perc = 0.6,
  na = TRUE,
  identical = TRUE,
  nThread = 4,
  many_sources = TRUE,
  load_report = TRUE
)

Arguments

folder string, full folder path to RPDR text files.
which_data string vector, an array of abbreviation corresponding to the datasources wished to load. Currently supported values and the default is: c("mrn", "con", "dem", "enc", "rdt", "lab", "med", "dia", "rfv", "prc", "lno", "car", "dis", "end", "hnp", "opn", "pat", "prg", "pul", "rad", "vis")
old_dem boolean, should old load_dem function be used for loading demographic data. Defaults to TRUE, should be set to FALSE for Dem.txt datasets prior to 2022.
merge_id string, column name to use to create ID_MERGE column used to merge different datasets. Defaults to EMPI, as it is the preferred MRN in the RPDR system. In case of mrn dataset, leave at EMPI, as it is automatically converted to: "Enterprise_Master_Patient_Index".
sep string, divider between hospital ID and MRN. Defaults to :
id_length string, indicating whether to modify MRN length based-on required values id_length = standard, or to keep lengths as is id_length = asis. If id_length = standard then in case of MGH, BWH, MCL, EMPI and PMRN the length of the MRNs are corrected accordingly by adding zeros, or removing numeral from the beginning. In other cases the lengths are unchanged. Defaults to standard.
perc numeric, a number between 0-1 indicating which parsed ID columns to keep. Data present in perc x 100% of patients are kept.
na boolean, whether to remove columns with only NA values. Defaults to TRUE.
identical boolean, whether to remove columns with identical values. Defaults to TRUE.
nThread integer, number of threads to use for parallelization.
many_sources boolean, if TRUE, then parallelization is done on the level of the datasources. If FALSE, then parallelization is done within the datasources. If there are many datasources, then it is advised to set this TRUE, as then each different datasource will be processed in parallel. However, if there are only a few datasources selected to load, but many files per datasource (result of large queries), then it
may be faster to parallelize within each datasource and therefore should be set to FALSE. If there are only a few sources each with one file then set to TRUE.

load_report boolean, should the report text be returned for notes.

Value
list of parsed data tables containing the information.

Examples

```r
## Not run:
#Load all Con, Dem and Mrn datasets processing all files within given datasource in parallel
load_all(folder = folder_rpdr, which_data = c("con", "dem", "mrn"),
nThread = 2, many_sources = FALSE)

#Load all supported file types paralleling on the level of datasources
load_all(folder = folder_rpdr, nThread = 2, many_sources = TRUE)

## End(Not run)
```

```r
load_con

Description

Loads patient contact, insurance, and PCP information into the R environment.

Usage

```r
load_con(
  file,
  merge_id = "EMPI",
  sep = ":",
  id_length = "standard",
  perc = 0.6,
  na = TRUE,
  identical = TRUE,
  nThread = 4,
  mrn_type = TRUE
)
```

Arguments

- **file** string, full file path to Con.txt.
- **merge_id** string, column name to use to create ID_MERGE column used to merge different datasets. Defaults to EPIC_PMRN, as it is the preferred MRN in the RPDR system.
load_con

sep string, divider between hospital ID and MRN. Defaults to :.

id_length string, indicating whether to modify MRN length based-on required values. If id_length = standard, then in case of MGH, BWH, MCL, EMPI and PMRN, the length of the MRNs are corrected accordingly by adding zeros, or removing numeral from the beginning. In other cases the lengths are unchanged. Defaults to standard.

perc numeric, a number between 0-1 indicating which parsed ID columns to keep. Data present in perc x 100% of patients are kept.

na boolean, whether to remove columns with only NA values. Defaults to TRUE.

identical boolean, whether to remove columns with identical values. Defaults to TRUE.

nThread integer, number of threads to use to load data.

mrn_type boolean, should data in MRN_Type and MRN be parsed. Defaults to TRUE only for Con.txt, as it is not advised to parse these for all data sources as it takes considerable time.

Value
data table, with contact information data.

ID_MERGE numeric, defined IDs by merge_id, used for merging later.

ID_con_EMPI string, Unique Partners-wide identifier assigned to the patient used to consolidate patient information from con datasource, corresponds to EMPI in RPDR. Data is formatted using pretty_mrn().

ID_con_PMRN string, Epic medical record number. This value is unique across Epic instances within the Partners network from con datasource, corresponds to EPIC_PMRN in RPDR. Data is formatted using pretty_mrn().

ID_con_loc string, if mrn_type == TRUE, then the data in MRN_Type and MRN are parsed into IDs corresponding to locations (loc). Data is formatted using pretty_mrn().

ID_con_loc_list string, if prevalence of IDs in Patient_ID_List > perc, then they are included in the output. Data is formatted using pretty_mrn().

name_last string, Patient’s last name, corresponds to Last_Name in RPDR.

name_first string, Patient’s first name, corresponds to First_Name in RPDR.

name_middle string, Patient’s middle name or initial, corresponds to Middle_Name in RPDR.

name_previous string, Any alternate names on record for this patient, corresponds to Previous_Name in RPDR.

SSN string, Social Security Number, corresponds to SSN in RPDR.

VIP character, Special patient statuses as defined by the EMPI group, corresponds to VIP in RPDR.

address1 string, Patient’s current address, corresponds to address1 in RPDR.

address2 string, Additional address information, corresponds to address2 in RPDR.

city string, City of residence, corresponds to City in RPDR.

state string, State of residence, corresponds to State in RPDR.

country_con string, Country of residence from con datasource, corresponds to Country in RPDR. Punctuation marks are removed.
load_dem numeric, Mailing zip code of primary residence from con datasource, corresponds to Zip in RPDR. Formatted to 5 character zip codes using pretty_numbers().

direct_contact_consent boolean, Indicates whether the patient has given permission to contact them directly through the RODY program, corresponds to Direct_Contact_Consent in RPDR.

phone_home number, Patient’s home phone number, corresponds to Home_Phone in RPDR. Formatted to 10 digit phone numbers using pretty_numbers().

phone_day number, Phone number where the patient can be reached during the day, corresponds to Day_Phone in RPDR. Formatted to 10 digit phone numbers using pretty_numbers().

insurance1 string, Patient’s primary health insurance carrier and subscriber ID information, corresponds to Insurance_1 in RPDR. Punctuation marks are removed.

insurance2 string, Patient’s secondary health insurance carrier and subscriber ID information, if any, corresponds to Insurance_2 in RPDR. Punctuation marks are removed.

insurance3 string, Patient’s tertiary health insurance carrier and subscriber ID information, if any, corresponds to Insurance_3 in RPDR. Punctuation marks are removed.

primary_care_physician string, Comma-delimited list of all primary care providers on record for this patient per institution, along with contact information (if available), corresponds to Primary_Care_Physician in RPDR. Punctuation marks are removed.

primary_care_physician_resident string, Comma-delimited list of any Resident primary care providers on record for this patient per institution, along with contact information (if available), corresponds to Resident_Primary_Care_Physician in RPDR. Punctuation marks are removed.

Examples

```r
## Not run:
#Using defaults
d_con <- load_con(file = "test_Con.txt")

#Use sequential processing
d_con <- load_con(file = "test_Con.txt", nThread = 1)

#Use parallel processing and parse data in MRN_Type and MRN columns (default in load_con) and keep all IDs
d_con <- load_con(file = "test_Con.txt", nThread = 20, mrn_type = TRUE, perc = 1)

## End(Not run)
```

load_dem Loads demographic information into R for new demographic tables following changes in the beginning of 2022.

Description

Loads patient demographic and vital status information into the R environment. Since version 0.2.2 of the software this function supports the new demographics table data definitions.
load_dem

Usage

load_dem(
  file,
  merge_id = "EMPI",
  sep = ":",
  id_length = "standard",
  perc = 0.6,
  na = TRUE,
  identical = TRUE,
  nThread = 4,
  mrn_type = FALSE
)

Arguments

file string, full file path to Dem.txt.
merge_id string, column name to use to create ID_MERGE column used to merge different datasets. Defaults to EPIC_PMRN, as it is the preferred MRN in the RPDR system.
sep string, divider between hospital ID and MRN. Defaults to ".
id_length string, indicating whether to modify MRN length based-on required values id_length = standard, or to keep lengths as is id_length = asis. If id_length = standard then in case of MGH, BWH, MCL, EMPI and PMRN the length of the MRNs are corrected accordingly by adding zeros, or removing numeral from the beginning. In other cases the lengths are unchanged. Defaults to standard.
perc numeric, a number between 0-1 indicating which parsed ID columns to keep. Data present in perc x 100% of patients are kept.
na boolean, whether to remove columns with only NA values. Defaults to TRUE.
identical boolean, whether to remove columns with identical values. Defaults to TRUE.
nThread integer, number of threads to use to load data.
mrn_type boolean, should data in MRN_Type and MRN be parsed. Defaults to FALSE, as it is not advised to parse these for all data sources as it takes considerable time.

Value

data table, with demographic information data.

ID_MERGE numeric, defined IDs by merge_id, used for merging later.
ID_dem_EMPI string, Unique Partners-wide identifier assigned to the patient used to consolidate patient information. from dem datasource, corresponds to EMPI in RPDR. Data is formatted using pretty_mrn().
ID_dem_PMRN string, Epic medical record number. This value is unique across Epic instances within the Partners network. from dem datasource, corresponds to EPIC_PMRN in RPDR. Data is formatted using pretty_mrn().
ID_dem_loc string, if mrn_type == TRUE, then the data in MRN_Type and MRN are parsed into IDs corresponding to locations (loc). Data is formatted using pretty_mrn().
**gender_legal_sex** string, Patient’s legal sex, corresponds to Gender_Legal_Sex in RPDR. Punctuation marks and white spaces are removed.

**sex_at_birth** string, Patient’s sex at time of birth, corresponds to Sex_at_Birth in RPDR. Punctuation marks and white spaces are removed.

**gender_identity** string, Patient’s personal conception of their gender, corresponds to Gender_Identity in RPDR. Punctuation marks and white spaces are removed.

**time_date_of_birth** POSIXct, Patient’s date of birth, corresponds to Date_of_Birth. Converted to POSIXct format.

**age** string, Patient’s current age (or age at death), corresponds to Age in RPDR.

**language** string, Patient’s preferred spoken language, corresponds to Language in RPDR. Punctuation marks and white spaces are removed.

**language_group** string, Patient’s preferred language: English or Non-English, corresponds to Language_Group in RPDR. Punctuation marks and white spaces are removed.

**race_1** string, Patient's primary race, corresponds to Race1 in RPDR. Punctuation marks and white spaces are removed.

**race_2** string, Patient’s primary race if more than one race, corresponds to Race2 in RPDR. Punctuation marks and white spaces are removed.

**race_group** string, Patient’s Race Group as determined by Race1 and Race2, corresponds to Race_Group in RPDR. Punctuation marks and white spaces are removed.

**ethnic_group** string, Patient’s Ethnicity: Hispanic or Non Hispanic, corresponds to Ethnic_Group in RPDR. Punctuation marks and white spaces are removed.

**marital** string, Patient’s current marital status, corresponds to Marital_Status in RPDR. Punctuation marks and white spaces are removed.

**religion** string, Patient-identified religious preference, corresponds to Religion in RPDR. Punctuation marks and white spaces are removed.

**veteran** string, Patient’s current military veteran status, corresponds to Is_a_veteran in RPDR. Punctuation marks and white spaces are removed.

**country_dem** string, Patient’s current country of residence from dem datasource, corresponds to Country in RPDR. Punctuation marks and white spaces are removed.

**zip_dem** string, Mailing zip code of patient’s primary residence from dem datasource, corresponds to Zip_code in RPDR. Formatted to 5 character zip codes.

**vital_status** string, Identifies if the patient is living or deceased. This data is updated monthly from the Partners registration system and the Social Security Death Master Index, corresponds to Vital_Status in RPDR. Punctuation marks are removed.

**time_date_of_death** POSIXct, Recorded date of death from source in 'Vital_Status'. Date of death information obtained solely from the Social Security Death Index will not be reported until 3 years after death due to privacy concerns. If the value is independently documented by a Partners entity within the 3 year window then the date will be displayed. corresponds to Date_of_Death in RPDR. Converted to POSIXct format.

**Examples**

```r
## Not run:
#Using defaults
```
load_dem_old

 loads demographic information into R for demographics tables before 2022.

Description

Loads patient demographic and vital status information into the R environment. Since version 0.2.2 of the software, this function supports the old demographics table data definitions and is identical to the load_dem function of previous versions of the software.

Usage

load_dem_old(
  file,
  merge_id = "EMPI",
  sep = ":",
  id_length = "standard",
  perc = 0.6,
  na = TRUE,
  identical = TRUE,
  nThread = 4,
  mrn_type = FALSE
)

Arguments

file string, full file path to Dem.txt.
merge_id string, column name to use to create ID MERGE column used to merge different datasets. Defaults to EPIC PMRN, as it is the preferred MRN in the RPDR system.
sep string, divider between hospital ID and MRN. Defaults to ":".
id_length string, indicating whether to modify MRN length based-on required values id_length = standard, or to keep lengths as is id_length = asis. If id_length = standard then in case of MGH, BWH, MCL, EMPI and PMRN the length of the MRNs are corrected accordingly by adding zeros, or removing numeral from the beginning. In other cases the lengths are unchanged. Defaults to standard.
perc numeric, a number between 0-1 indicating which parsed ID columns to keep. Data present in \( \text{perc} \times 100\% \) of patients are kept.

na boolean, whether to remove columns with only NA values. Defaults to \text{TRUE}.

identical boolean, whether to remove columns with identical values. Defaults to \text{TRUE}.

nThread integer, number of threads to use to load data.

mrn_type boolean, should data in \text{MRN\_Type} and \text{MRN} be parsed. Defaults to \text{FALSE}, as it is not advised to parse these for all data sources as it takes considerable time.

Value
data table, with demographic information data.

ID\_MERGE numeric, defined IDs by \text{merge\_id}, used for merging later.

ID\_dem\_EMPI string, Unique Partners-wide identifier assigned to the patient used to consolidate patient information. from \text{dem} datasource, corresponds to EMPI in RPDR. Data is formatted using pretty\_mrn().

ID\_dem\_PMRN string, Epic medical record number. This value is unique across Epic instances within the Partners network. from \text{dem} datasource, corresponds to EPIC\_PMRN in RPDR. Data is formatted using pretty\_mrn().

ID\_dem\_loc string, if mrn\_type == \text{TRUE}, then the data in \text{MRN\_Type} and \text{MRN} are parsed into IDs corresponding to locations \text{(loc)}. Data is formatted using pretty\_mrn().

gender string, Patient’s legal sex, corresponds to Gender in RPDR. Punctuation marks and white spaces are removed.

time\_date\_of\_birth POSIXct, Patient’s date of birth, corresponds to Date\_of\_Birth in RPDR. Converted to POSIXct format.

age string, Patient’s current age (or age at death), corresponds to Age in RPDR.

language string, Patient’s preferred spoken language, corresponds to Language in RPDR. Punctuation marks and white spaces are removed.

race string, Patient’s primary race, corresponds to Race in RPDR. Punctuation marks and white spaces are removed.

marital string, Patient’s current marital status, corresponds to Marital\_Status in RPDR. Punctuation marks and white spaces are removed.

religion string, Patient-identified religious preference, corresponds to Religion in RPDR. Punctuation marks and white spaces are removed.

veteran string, Patient’s current military veteran status, corresponds to Is\_a\_veteran in RPDR. Punctuation marks and white spaces are removed.

country\_dem string, Patient’s current country of residence from \text{dem} datasource, corresponds to Country in RPDR. Punctuation marks and white spaces are removed.

zip\_dem string, Mailing zip code of patient’s primary residence from \text{dem} datasource, corresponds to Zip\_code in RPDR. Formatted to 5 character zip codes.

vital\_status string, Identifies if the patient is living or deceased. This data is updated monthly from the Partners registration system and the Social Security Death Master Index, corresponds to Vital\_Status in RPDR. Punctuation marks are removed.
time\_date\_of\_death  POSIXct, Recorded date of death from source in ‘Vital\_Status’. Date of death information obtained solely from the Social Security Death Index will not be reported until 3 years after death due to privacy concerns. If the value is independently documented by a Partners entity within the 3 year window then the date will be displayed. corresponds to Date\_of\_Death in RPDR. Converted to POSIXct format.

Examples

```r
## Not run:
#Using defaults
d_dem <- load_dem_old(file = "test_Dem.txt")

#Use sequential processing
d_dem <- load_dem_old(file = "test_Dem.txt", nThread = 1)

#Use parallel processing and parse data in MRN\_Type and MRN columns and keep all IDs
d_dem <- load_dem_old(file = "test_Dem.txt", nThread = 20, mrn\_type = TRUE, perc = 1)

## End(Not run)
```

---

load\_dia

Loads diagnoses into R.

Description

Loads diagnoses information into the R environment.

Usage

```r
load\_dia(
  file,
  merge\_id = "EMPI",
  sep = ":",
  id\_length = "standard",
  perc = 0.6,
  na = TRUE,
  identical = TRUE,
  nThread = 4,
  mrn\_type = FALSE
)
```

Arguments

- `file` string, full file path to Dia.txt.
- `merge\_id` string, column name to use to create `ID\_MERGE` column used to merge different datasets. Defaults to `EPIC\_PMRN`, as it is the preferred MRN in the RPDR system.
- `sep` string, divider between hospital ID and MRN. Defaults to `:`.
id_length string, indicating whether to modify MRN length based-on required values id_length = standard, or to keep lengths as is id_length = asis. If id_length = standard then in case of MGH, BWH, MCL, EMPI and PMRN the length of the MRNs are corrected accordingly by adding zeros, or removing numeral from the beginning. In other cases the lengths are unchanged. Defaults to standard.

perc numeric, a number between 0-1 indicating which parsed ID columns to keep. Data present in perc x 100% of patients are kept.

na boolean, whether to remove columns with only NA values. Defaults to TRUE.

identical boolean, whether to remove columns with identical values. Defaults to TRUE.

nThread integer, number of threads to use to load data.

mrn_type boolean, should data in MRN_Type and MRN be parsed. Defaults to FALSE, as it is not advised to parse these for all data sources as it takes considerable time.

Value data table, with diagnoses information.

ID_MERGE numeric, defined IDs by merge_id, used for merging later.

ID_dia_EMPI string, Unique Partners-wide identifier assigned to the patient used to consolidate patient information from dia datasource, corresponds to EMPI in RPDR. Data is formatted using pretty_mrn().

ID_dia_PMRN string, Epic medical record number. This value is unique across Epic instances within the Partners network from dia datasource, corresponds to EPIC_PMRN in RPDR. Data is formatted using pretty_mrn().

ID_dia_loc string, if mrn_type == TRUE, then the data in MRN_Type and MRN are parsed into IDs corresponding to locations (loc). Data is formatted using pretty_mrn().

time_dia POSIXct, Date when the diagnosis was noted, corresponds to Date in RPDR. Converted to POSIXct format.

dia_name string, Name of the diagnosis, diagnosis-related group, or phenotype. For more information on available Phenotypes visit https://phenotypes.partners.org/phenotype_list.html, corresponds to Diagnosis_Name in RPDR.

dia_code string, Diagnosis, diagnosis-related group, or phenotype code, corresponds to Code in RPDR.

dia_code_type string, Standardized classification system or custom grouping associated with the diagnosis code, corresponds to Code_type in RPDR.

dia_flag string, Qualifier for the diagnosis, if any, corresponds to Diagnosis_flag in RPDR.

dia_enc_num string, Unique identifier of the record/visit. This values includes the source system, hospital, and a unique identifier within the source system, corresponds to Encounter_number in RPDR.

dia_provided string, Provider of record for the encounter where the diagnosis was entered, corresponds to Provider in RPDR. Punctuation marks are removed.

dia_clinic string, Specific department/location where the patient encounter took place, corresponds to Clinic in RPDR.

dia_hosp string, Facility where the encounter occurred, corresponds to Hospital in RPDR.

dia_inpatient string, Identifies whether the diagnosis was noted during an inpatient or outpatient encounter, corresponds to Inpatient_Outpatient in RPDR. Punctuation marks removed.
**Examples**

```r
## Not run:
# Using defaults
d_dia <- load_dia(file = "test_Dia.txt")

# Use sequential processing
d_dia <- load_dia(file = "test_Dia.txt", nThread = 1)

# Use parallel processing and parse data in MRN_Type and MRN columns and keep all IDs
d_dia <- load_dia(file = "test_Dia.txt", nThread = 20, mrn_type = TRUE, perc = 1)

## End(Not run)
```

**load_enc**

Loads encounter information into R.

**Description**

Loads encounter-level detail information into the R environment.

**Usage**

```r
load_enc(
  file,
  merge_id = "EMPI",
  sep = ":",
  id_length = "standard",
  perc = 0.6,
  na = TRUE,
  identical = TRUE,
  nThread = 4,
  mrn_type = FALSE
)
```

**Arguments**

- **file**: string, full file path to Enc.txt or Exc.txt
- **merge_id**: string, column name to use to create `ID_MERGE` column used to merge different datasets. Defaults to `EPIC_PMRN`, as it is the preferred MRN in the RPDR system.
- **sep**: string, divider between hospital ID and MRN. Defaults to `:`.
- **id_length**: string, indicating whether to modify MRN length based-on required values `id_length = standard`, or to keep lengths as is `id_length = asis`. If `id_length = standard` then in case of `MGH`, `BWH`, `MCL`, `EMPI` and `PMRN` the length of the MRNs are corrected accordingly by adding zeros, or removing numeral from the beginning. In other cases the lengths are unchanged. Defaults to `standard`.  
  
  ```r
<table>
<thead>
<tr>
<th>Argument</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>file</td>
<td></td>
<td>string, full file path to Enc.txt or Exc.txt</td>
</tr>
<tr>
<td>merge_id</td>
<td>&quot;EMPI&quot;</td>
<td>string, column name to use to create <code>ID_MERGE</code> column used to merge different datasets. Defaults to <code>EPIC_PMRN</code>, as it is the preferred MRN in the RPDR system.</td>
</tr>
<tr>
<td>sep</td>
<td>&quot;:&quot;</td>
<td>string, divider between hospital ID and MRN. Defaults to <code>:</code>.</td>
</tr>
<tr>
<td>id_length</td>
<td>&quot;standard&quot;</td>
<td>string, indicating whether to modify MRN length based-on required values <code>id_length = standard</code>, or to keep lengths as is <code>id_length = asis</code>. If <code>id_length = standard</code> then in case of <code>MGH</code>, <code>BWH</code>, <code>MCL</code>, <code>EMPI</code> and <code>PMRN</code> the length of the MRNs are corrected accordingly by adding zeros, or removing numeral from the beginning. In other cases the lengths are unchanged. Defaults to <code>standard</code>.</td>
</tr>
</tbody>
</table>
perc numeric, a number between 0-1 indicating which parsed ID columns to keep. Data present in perc x 100% of patients are kept.

na boolean, whether to remove columns with only NA values. Defaults to TRUE.

identical boolean, whether to remove columns with identical values. Defaults to TRUE.

nThread integer, number of threads to use to load data.

mrn_type boolean, should data in MRN_Type and MRN be parsed. Defaults to FALSE, as it is not advised to parse these for all data sources as it takes considerable time.

Value
data table, with encounter information.

ID_MERGE numeric, defined IDs by merge_id, used for merging later.

ID_enc_EMPI string, Unique Partners-wide identifier assigned to the patient used to consolidate patient information from enc datasource, corresponds to EMPI in RPDR. Data is formatted using pretty_mrn().

ID_enc_PMRN string, Epic medical record number. This value is unique across Epic instances within the Partners network from enc datasource, corresponds to EPIC_PMRN in RPDR. Data is formatted using pretty_mrn().

ID_enc_loc string, if mrn_type == TRUE, then the data in MRN_Type and MRN are parsed into IDs corresponding to locations (loc). Data is formatted using pretty_mrn().

e enc numb string, Unique identifier of the record/visit. This values includes the source system, hospital, and a unique identifier within the source system, corresponds to Encounter_number in RPDR.

time_enc_admit POSIXct, Date when the patient was admitted or entered the facility, corresponds to Admit_Date in RPDR. Converted to POSIXct format.

time_enc_disch POSIXct, Date when the patient was discharged or left the facility, corresponds to Discharge_Date in RPDR. Converted to POSIXct format.

enc_status string, Billing account-related notes about the encounter. This will not be populated for all encounters, corresponds to Encounter_Status in RPDR. Punctuation marks are removed.

enc_hosp string, Facility where the encounter occurred, corresponds to Hospital in RPDR.

enc_inpatient string, Classifies the type of encounter as either Inpatient or Outpatient. ED visits are currently classified under the 'Outpatient' label, corresponds to Inpatient_or_Outpatient in RPDR. Punctuation marks are removed.

enc_service string, Hospital service line assigned to the encounter, corresponds to Service_Line in RPDR. Punctuation marks are removed.

enc_attending string, The attending provider associated with the encounter. For Epic professional billing, this is the billing provider, corresponds to Attending_MD in RPDR. Punctuation marks are removed.

enc_length numeric, Length of stay for the encounter, corresponds to LOS_days in RPDR.

enc_clinic string, Specific department/location where the encounter occurred, corresponds to Clinic_Name in RPDR.

enc_admit_src string, Location where the patient was admitted when entering the hospital/clinic, corresponds to Admit_Source in RPDR. Punctuation marks are removed.
**enc_pat_type** string, Provides information regarding the specific patient classifications and status of the patient visit. This field is only populated for McLean Hospital encounters, corresponds to Patient_Type in RPDR. Punctuation marks are removed.

**enc_ref_disp** string, Location where the patient has been directed for treatment or follow-up by a staff member. This field is only populated for McLean Hospital encounters, corresponds to Referrer_Discipline in RPDR. Punctuation marks are removed.

**enc_disch_disp** string, Patient’s anticipated location or status following the encounter, corresponds to Discharge_Disposition in RPDR. Punctuation marks are removed.

**enc_pay** string, Payors responsible for the hospital account. Multiple payors (primary, secondary, etc.) may be listed, corresponds to Payor in RPDR. Punctuation marks are removed.

**enc_diag_admit** string, Initial working diagnosis documented by the admitting or attending physician, corresponds to Admitting_Diagnosis in RPDR. Punctuation marks and white spaces are removed.

**enc_diag_princ** string, Condition established, after study, to be chiefly responsible for occasioning the admission of the patient to the hospital for care, corresponds to Principle_Diagnosis in RPDR. Punctuation marks and white spaces are removed.

**enc_diag_1** string, Additional diagnoses associated with this encounter or visit, corresponds to Diagnosis_1 in RPDR. Punctuation marks and white spaces are removed.

**enc_diag_2** string, Additional diagnoses associated with this encounter or visit, corresponds to Diagnosis_2 in RPDR. Punctuation marks and white spaces are removed.

**enc_diag_3** string, Additional diagnoses associated with this encounter or visit, corresponds to Diagnosis_3 in RPDR. Punctuation marks and white spaces are removed.

**enc_diag_4** string, Additional diagnoses associated with this encounter or visit, corresponds to Diagnosis_4 in RPDR. Punctuation marks and white spaces are removed.

**enc_diag_5** string, Additional diagnoses associated with this encounter or visit, corresponds to Diagnosis_5 in RPDR. Punctuation marks and white spaces are removed.

**enc_diag_6** string, Additional diagnoses associated with this encounter or visit, corresponds to Diagnosis_6 in RPDR. Punctuation marks and white spaces are removed.

**enc_diag_7** string, Additional diagnoses associated with this encounter or visit, corresponds to Diagnosis_7 in RPDR. Punctuation marks and white spaces are removed.

**enc_diag_8** string, Additional diagnoses associated with this encounter or visit, corresponds to Diagnosis_8 in RPDR. Punctuation marks and white spaces are removed.

**enc_diag_9** string, Additional diagnoses associated with this encounter or visit, corresponds to Diagnosis_9 in RPDR. Punctuation marks and white spaces are removed.

**enc_diag_10** string, Additional diagnoses associated with this encounter or visit, corresponds to Diagnosis_10 in RPDR. Punctuation marks and white spaces are removed.

**enc_diag_group** string, Diagnosis-Related Group for the encounter, in the following format: SYSTEM:CODE - Description, corresponds to DRG in RPDR. Punctuation marks and white spaces are removed.

---

**Examples**

```r
## Not run:
#Using defaults
```
d_enc <- load_enc(file = "test_Enc.txt")

#Use sequential processing
d_enc <- load_enc(file = "test_Enc.txt", nThread = 1)

#Use parallel processing and parse data in MRN_Type and MRN columns and keep all IDs
d_enc <- load_enc(file = "test_Enc.txt", nThread = 20, mrn_type = TRUE, perc = 1)

## End(Not run)

### load_lab

**Loads laboratory results into R.**

**Description**

Loads laboratory results into the R environment.

**Usage**

```r
load_lab(
  file,
  merge_id = "EMPI",
  sep = ":",
  id_length = "standard",
  perc = 0.6,
  na = TRUE,
  identical = TRUE,
  nThread = 4,
  mrn_type = FALSE
)
```

**Arguments**

- **file** string, full file path to Enc.txt or Exc.txt
- **merge_id** string, column name to use to create ID_MERGE column used to merge different datasets. Defaults to `EPIC_PMRN`, as it is the preferred MRN in the RPDR system.
- **sep** string, divider between hospital ID and MRN. Defaults to `:`.
- **id_length** string, indicating whether to modify MRN length based-on required values `id_length = standard`, or to keep lengths as is `id_length = asis`. If `id_length = standard` then in case of `MGH, BWH, MCL, EMPI and PMRN` the length of the MRNs are corrected accordingly by adding zeros, or removing numeral from the beginning. In other cases the lengths are unchanged. Defaults to `standard`.
- **perc** numeric, a number between 0-1 indicating which parsed ID columns to keep. Data present in `perc x 100%` of patients are kept.
- **na** boolean, whether to remove columns with only NA values. Defaults to `TRUE`.

identical boolean, whether to remove columns with identical values. Defaults to TRUE.
nThread integer, number of threads to use to load data.
mrn_type boolean, should data in MRN_Type and MRN be parsed. Defaults to FALSE, as it is not advised to parse these for all data sources as it takes considerable time.

Value
data table, with laboratory exam information.

ID_MERGE numeric, defined IDs by merge_id, used for merging later.

ID_lab_EMPI string, Unique Partners-wide identifier assigned to the patient used to consolidate patient information from lab datasource, corresponds to EMPI in RPDR. Data is formatted using pretty_mrn().

ID_lab_PMRN string, Epic medical record number. This value is unique across Epic instances within the Partners network from lab datasource, corresponds to EPIC_PMRN in RPDR. Data is formatted using pretty_mrn().

ID_lab_loc string, if mrn_type == TRUE, then the data in MRN_Type and MRN are parsed into IDs corresponding to locations (loc). Data is formatted using pretty_mrn().

time_lab_result POSIXct, Date when the specimen was collected, corresponds to Seq_Date_Time in RPDR. Converted to POSIXct format.

lab_group string, Higher-level grouping concept used to consolidate similar tests across hospitals, corresponds to Group_ID in RPDR.

lab_loinc string, Standardized LOINC code for the laboratory test, corresponds to Loinc_Code in RPDR.

lab_testID string, Internal identifier for the test used by the source system, corresponds to Test_ID in RPDR.

lab_descript string, Name of the lab test, corresponds to Test_Description in RPDR.

lab_result string, Result value for the test, corresponds to Result in RPDR.

lab_result_txt string, Additional information included with the result. This can include instructions for interpretation or comments from the laboratory, corresponds to Result_Text in RPDR.

lab_result_abn string, Flag for identifying if values are outside of normal ranges or represent a significant deviation from previous values, corresponds to Abnormal_Flag in RPDR.

lab_result_unit string, Units associated with the result value, corresponds to Reference_Unit in RPDR.

lab_result_range string, Normal or therapeutic range for this value, corresponds to Reference_Range in RPDR.

lab_result_toxic string, Reference range of values defined as being toxic to the patient, corresponds to Toxic_Range in RPDR.

lab_spec string, Type of specimen collected to perform the test, corresponds to Specimen_Type in RPDR.

lab_spec_txt string, Free-text information about the specimen, its collection or its integrity, corresponds to Specimen_Text in RPDR.

lab_correction string, Free-text information about any changes made to the results, corresponds to Correction_Flag in RPDR.
**lab_status** string, Flag which indicates whether the procedure is pending or complete, corresponds to Test_Status in RPDR.

**lab_ord_pys** string, Name of the ordering physician, corresponds to Ordering_Doc in RPDR. Punctuation marks are removed.

**lab_accession** string, Internal tracking number assigned to the specimen for identification in the lab, corresponds to Accession in RPDR.

**lab_source** string, Database source, either CDR (Clinical Data Repository) or RPDR (internal RPDR database), corresponds to Source in RPDR.

### Examples

```r
## Not run:
#Using defaults
d_lab <- load_lab(file = "test_Lab.txt")

#Use sequential processing
d_lab <- load_lab(file = "test_Lab.txt", nThread = 1)

#Use parallel processing and parse data in MRN_Type and MRN columns and keep all IDs
d_lab <- load_lab(file = "test_Lab.txt", nThread = 20, mrn_type = TRUE, perc = 1)

## End(Not run)
```

---

**load_lno**

**Loads LMR note documents into R.**

### Description

Loads notes from the LMR legacy EHR system.

### Usage

```r
load_lno(
  file,
  merge_id = "EMPI",
  sep = ",",
  id_length = "$standard",
  perc = 0.6,
  na = TRUE,
  identical = TRUE,
  nThread = 4,
  mrn_type = FALSE
)
```
Arguments

file string, full file path to Lno.txt.
merge_id string, column name to use to create ID_MERGE column used to merge different datasets. Defaults to EPIC_PMRN, as it is the preferred MRN in the RPDR system.
sep string, divider between hospital ID and MRN. Defaults to :.
id_length string, indicating whether to modify MRN length based-on required values id_length = standard, or to keep lengths as is id_length = asis. If id_length = standard then in case of MGH, BWH, MCL, EMPI and PMRN the length of the MRNs are corrected accordingly by adding zeros, or removing numeral from the beginning. In other cases the lengths are unchanged. Defaults to standard.
perc numeric, a number between 0-1 indicating which parsed ID columns to keep. Data present in perc x 100% of patients are kept.
na boolean, whether to remove columns with only NA values. Defaults to TRUE.
identical boolean, whether to remove columns with identical values. Defaults to TRUE.
nThread integer, number of threads to use to load data.
mrn_type boolean, should data in MRN_Type and MRN be parsed. Defaults to FALSE, as it is not advised to parse these for all data sources as it takes considerable time.

Value
data table, with LMR notes information.

ID_MERGE numeric, defined IDs by merge_id, used for merging later.
ID_Lno_EMPI string. Unique Partners-wide identifier assigned to the patient used to consolidate patient information from Lno datasource, corresponds to EMPI in RPDR. Data is formatted using pretty_mrn().
ID_Lno_PMRN string, Epic medical record number. This value is unique across Epic instances within the Partners network from Lno datasource, corresponds to EPIC_PMRN in RPDR. Data is formatted using pretty_mrn().
ID_Lno_loc string, if mrn_type == TRUE, then the data in MRN_Type and MRN are parsed into IDs corresponding to locations (loc). Data is formatted using pretty_mrn().
time_Lno POSIXct, Date when the report was filed, corresponds to LMRNote_Date in RPDR. Converted to POSIXct format.
lno_rec_id string, Internal identifier for this report within the LMR system, corresponds to Record_Id in RPDR.
lno_status string, Completion status of the note, corresponds to Status in RPDR.
lno_author string, Name of user who created the note, corresponds to Author in RPDR.
lno_author_mrn string, Author’s user identifier within the LMR system, corresponds to Author_MRN in RPDR.
lno_COD string, Hospital-specific user code of the note author. The first character is a hospital-specific prefix, corresponds to COD in RPDR. Punctuation marks are removed.
lno_hosp string, Facility where the encounter occurred, corresponds to Institution in RPDR.
**load_med**

Loads medication order detail into R.

**Description**

Loads medication order detail information into the R environment.

**Usage**

```r
load_med(
  file,
  merge_id = "EMPI",
  sep = ":",
  id_length = "standard",
  perc = 0.6,
  na = TRUE,
  identical = TRUE,
  nThread = 4,
  mrn_type = FALSE
)
```

**Arguments**

- **file** string, full file path to Enc.txt or Exc.txt
- **merge_id** string, column name to use to create `ID MERGE` column used to merge different datasets. Defaults to `EPIC PMRN`, as it is the preferred MRN in the RPDR system.
- **sep** string, divider between hospital ID and MRN. Defaults to `:`.

**Examples**

```
## Not run:
#Using defaults
d_lno <- load_lno(file = "test_Lno.txt")

#Use sequential processing
d_lno <- load_lno(file = "test_Lno.txt", nThread = 1)

#Use parallel processing and parse data in MRN_Type and MRN columns and keep all IDs
d_lno <- load_lno(file = "test_Lno.txt", nThread = 20, mrn_type = TRUE, perc = 1)

## End(Not run)
```

**Ino_subject** string, Type of note. This value is derived from the "Subject" line of the narrative text, corresponds to Subject in RPDR.

**Ino_rep_txt** string, Full narrative text of the note, corresponds to Comments in RPDR.
id_length string, indicating whether to modify MRN length based-on required values id_length = standard, or to keep lengths as is id_length = asis. If id_length = standard then in case of MGH, BWH, MCL, EMPI and PMRN the length of the MRNs are corrected accordingly by adding zeros, or removing numeral from the beginning. In other cases the lengths are unchanged. Defaults to standard.

perc numeric, a number between 0-1 indicating which parsed ID columns to keep. Data present in perc x 100% of patients are kept.

na boolean, whether to remove columns with only NA values. Defaults to TRUE.

identical boolean, whether to remove columns with identical values. Defaults to TRUE.

nThread integer, number of threads to use to load data.

mrn_type boolean, should data in MRN_Type and MRN be parsed. Defaults to FALSE, as it is not advised to parse these for all data sources as it takes considerable time.

Value
data table, with medication order information.

ID_MERGE numeric, defined IDs by merge_id, used for merging later.

ID_med_EMPI string, Unique Partners-wide identifier assigned to the patient used to consolidate patient information from enc datasource, corresponds to EMPI in RPDR. Data is formatted using pretty_mrn().

ID_med_PMRN string, Epic medical record number. This value is unique across Epic instances within the Partners network from enc datasource, corresponds to EPIC_PMRN in RPDR. Data is formatted using pretty_mrn().

ID_med_loc string, if mrn_type == TRUE, then the data in MRN_Type and MRN are parsed into IDs corresponding to locations (loc). Data is formatted using pretty_mrn().

med_enc_numb string, Unique identifier of the record/visit, displayed in the following format: Source System - Institution Number, corresponds to Encounter_number in RPDR.

time_med POSIXct, Completion status of the requested test/transfusion. Converted to POSIXct format, corresponds to Medication_Date in RPDR.

time_med_detail string, To clarify when patients may have stopped taking a medication, this column provides the statuses of 'Listed' or 'Removed'. This is provided on pre-Epic (LMR) medication dates (1997-2017). The 'Listed' value denotes that a medication was on the patient’s medication list on the date indicated. The 'Removed' value denotes that a medication was removed from a patient’s medication list on the date indicated. Corresponds to Medication_Date_Detail in RPDR.

med string, Name of the medication. This may be appended with the source system in the case of OnCall and LMR medications, corresponds to Medication in RPDR.

med_code string, Medication code associated with the "Code_type" value, corresponds to Code in RPDR.

med_code_type string, Standardized classification system or custom source value used to identify the medication, corresponds to Code_Type in RPDR.

med_quant string, Number of units of the medication ordered, corresponds to Quantity in RPDR.

med_prov string, Ordering provider for the medication, corresponds to Provider in RPDR. Punctuation marks are removed.
**med_clinic** string. Specific department/location where the medication was ordered or administered, corresponds to Clinic in RPDR.

**med_hosp** string. Facility where the medication was ordered or administered, corresponds to Hospital in RPDR.

**med_inpatient** string. Identifies whether the medication was ordered with an Inpatient or Outpatient indication, corresponds to Inpatient_Outpatient in RPDR. Punctuation marks are removed.

**med_add_info** string. Additional administration information about the medication, corresponds to Additional_Info in RPDR.

### Examples

```
## Not run:
#Using defaults
d_med <- load_med(file = "test_Med.txt")

#Use sequential processing
d_med <- load_med(file = "test_Med.txt", nThread = 1)

#Use parallel processing and parse data in MRN_Type and MRN columns and keep all IDs
d_med <- load_med(file = "test_Med.txt", nThread = 20, mrn_type = TRUE, perc = 1)

## End(Not run)
```

---

**load_mrn**

*Loads MRN data into R.*

**Description**

Loads patient identifiers for Partners institutions, including hospital-specific MRNs into the R environment.

**Usage**

```
load_mrn(
  file,
  merge_id = "EMPI",
  sep = ":",
  id_length = "standard",
  perc = 0.6,
  na = TRUE,
  identical = TRUE,
  nThread = 4,
  mrn_type = FALSE
)
```
Arguments

- **file**: string, full file path to Mrn.txt.
- **merge_id**: string, column name to use to create \texttt{ID\_MERGE} column used to merge different datasets. Defaults to \texttt{EPIC\_PMRN}, as it is the preferred MRN in the RPDR system.
- **sep**: string, divider between hospital ID and MRN. Defaults to `:`.
- **id_length**: string, indicating whether to modify MRN length based-on required values \texttt{id\_length = standard}, or to keep lengths as is \texttt{id\_length = asis}. If \texttt{id\_length = standard} then in case of \texttt{MGH, BWH, MCL, EMPI and PMRN} the length of the MRNs are corrected accordingly by adding zeros, or removing numeral from the beginning. In other cases the lengths are unchanged. Defaults to \texttt{standard}.
- **perc**: numeric, a number between 0-1 indicating which parsed ID columns to keep. Data present in \texttt{perc x 100\%} of patients are kept. Not used for loading mrn data.
- **na**: boolean, whether to remove columns with only NA values. Defaults to \texttt{TRUE}.
- **identical**: boolean, whether to remove columns with identical values. Defaults to \texttt{TRUE}.
- **nThread**: integer, number of threads to use to load data.
- **mrn_type**: boolean, should data in \texttt{MRN\_Type} and \texttt{MRN} be parsed. Defaults to \texttt{FALSE}, as it is not advised to parse these for all data sources as it takes considerable time.

Value

Data table, with MRN data.

- **ID\_MERGE**: numeric, defined IDs by \texttt{merge_id}, used for merging later.
- **ID\_mrn\_INCOMING**: string, Patient identifier, usually the EMPI, corresponds to IncomingId in RPDR. Data is formatted using \texttt{pretty\_mrn()}.
- **ID\_mrn\_INCOMING\_SITE**: string, Source of identifier, e.g. EMP for Enterprise Master Patient Index, MGH for Mass General Hospital, corresponds to IncomingSite in RPDR.
- **ID\_mrn\_PMRN**: string, Epic medical record number. This value is unique across Epic instances within the Partners network, corresponds to EPIC\_PMRN in RPDR. Data is formatted using \texttt{pretty\_mrn()}.
- **ID\_mrn\_EMPI**: string, Unique Partners-wide identifier assigned to the patient used to consolidate patient information, corresponds to Enterprise\_Master\_Patient\_Index in RPDR. Data is formatted using \texttt{pretty\_mrn()}.
- **ID\_mrn\_MGH**: string, Unique Medical Record Number for Mass General Hospital, corresponds to MGH\_MRN in RPDR. Data is formatted using \texttt{pretty\_mrn()}.
- **ID\_mrn\_BWH**: string, Unique Medical Record Number for Brigham and Women’s Hospital, corresponds to BWH\_MRN in RPDR. Data is formatted using \texttt{pretty\_mrn()}. 
- **ID\_mrn\_FH**: string, Unique Medical Record Number for Faulkner Hospital, corresponds to FH\_MRN in RPDR. Data is formatted using \texttt{pretty\_mrn()}. 
- **ID\_mrn\_SRH**: string, Unique Medical Record Number for Spaulding Rehabilitation Hospital, corresponds to SRH\_MRN in RPDR. Data is formatted using \texttt{pretty\_mrn()}. 
- **ID\_mrn\_NWH**: string, Unique Medical Record Number for Newton-Wellesley Hospital, corresponds to NWH\_MRN in RPDR. Data is formatted using \texttt{pretty\_mrn()}. 

**ID_mrn_NSNC** string, Unique Medical Record Number for North Shore Medical Center, corresponds to NSMC_MRN in RPDR. Data is formatted using pretty_mrn().

**ID_mrn_MCL** string, Unique Medical Record Number for McLean Hospital, corresponds to MCL_MRN in RPDR. Data is formatted using pretty_mrn().

**ID_mrn_MEE** string, Unique Medical Record Number for Mass Eye and Ear, corresponds to MEE_MRN in RPDR. Data is formatted using pretty_mrn().

**ID_mrn_DFC** string, Unique Medical Record Number for Dana Farber Cancer center, corresponds to DFC_MRN in RPDR. Data is formatted using pretty_mrn().

**ID_mrn_WDH** string, Unique Medical Record Number for Wentworth-Douglass Hospital, corresponds to WDH_MRN in RPDR. Data is formatted using pretty_mrn().

**ID_mrn_STATUS** string, Status of the record, corresponds to Status in RPDR.

**Examples**

```r
## Not run:
# Using defaults
d_mrn <- load_mrn(file = "test_Mrn.txt")

# Use sequential processing
d_mrn <- load_mrn(file = "test_Mrn.txt", nThread = 1)

# Use parallel processing and parse data in MRN_Type and MRN columns and keep all IDs
d_mrn <- load_mrn(file = "test_Mrn.txt", nThread = 20, mrn_type = TRUE, perc = 1)

## End(Not run)
```

---

**load_notes**

Loads note documents into R.

**Description**

Loads documents information into the R environment, which are:

- Cardiology: "car"
- Discharge: "dis"
- Endoscopy: "end"
- History & Physical: "hnp"
- Operative: "opn"
- Pathology: "pat"
- Progress: "prg"
- Pulmonary: "pul"
- Radiology: "rad"
- Visit: "vis"
load_notes

Usage

```r
load_notes(
    file,
    type,
    merge_id = "EMPI",
    sep = ":",
    id_length = "standard",
    perc = 0.6,
    na = TRUE,
    identical = TRUE,
    nThread = 4,
    mrn_type = FALSE,
    load_report = TRUE
)
```

Arguments

**file** string, full file path to given type of note i.e. Hnp.txt.

**type** string, the type of note to be loaded. May be one of: "car", "dis", "end", "hnp", "opn", "pat", "prg", "pul", "rad" or "vis".

**merge_id** string, column name to use to create ID_MERGE column used to merge different datasets. Defaults to EPIC_PMRN, as it is the preferred MRN in the RPDR system.

**sep** string, divider between hospital ID and MRN. Defaults to ":".

**id_length** string, indicating whether to modify MRN length based-on required values id_length = standard, or to keep lengths as is id_length = asis. If id_length = standard then in case of MGH, BWH, MCL, EMPI and PMRN the length of the MRNs are corrected accordingly by adding zeros, or removing numeral from the beginning. In other cases the lengths are unchanged. Defaults to standard.

**perc** numeric, a number between 0-1 indicating which parsed ID columns to keep. Data present in perc x 100% of patients are kept.

**na** boolean, whether to remove columns with only NA values. Defaults to TRUE.

**identical** boolean, whether to remove columns with identical values. Defaults to TRUE.

**nThread** integer, number of threads to use to load data.

**mrn_type** boolean, should data in MRN_Type and MRN be parsed. Defaults to FALSE, as it is not advised to parse these for all data sources as it takes considerable time.

**load_report** boolean, should the report text be returned in the data table. Defaults to TRUE. However, be aware that some notes may take up more memory than available on the machine.

Value

data table, with notes information. abc stands for the three letter abbreviation of the given type of note.

**ID_MERGE** numeric, defined IDs by merge_id, used for merging later.
load_phy

ID_abc_EMPI string. Unique Partners-wide identifier assigned to the patient used to consolidate patient information from abc datasource, corresponds to EMPI in RPDR. Data is formatted using pretty_mrn().

ID_abc_PMRN string. Epic medical record number. This value is unique across Epic instances within the Partners network from abc datasource, corresponds to EPIC_PMRN in RPDR. Data is formatted using pretty_mrn().

ID_abc_loc string. If mrn_type == TRUE, then the data in MRN_Type and MRN are parsed into IDs corresponding to locations (loc). Data is formatted using pretty_mrn().

abc_rep_num string. Source-specific identifier used to reference the report, corresponds to Report_Number in RPDR.

time_abc POSIXct, Date when the report was filed, corresponds to Report_Date_Time in RPDR. Converted to POSIXct format.

abc_rep_desc string. Type of report or procedure documented in the report, corresponds to Report_Description in RPDR.

abc_rep_status string. Completion status of the note/report, corresponds to Report_Status in RPDR.

abc_rep_type string. See specification in RPDR data dictionary, corresponds to Report_Type in RPDR.

abc_rep_txt string. Full narrative text contained in the note/report, corresponds to Report_Text in RPDR. Only provided if load_report is TRUE.

Examples

## Not run:
# Using defaults
d_hnp <- load_notes(file = "test_Hnp.txt", type = "hnp")

# Use sequential processing
d_hnp <- load_notes(file = "test_Hnp.txt", type = "hnp", nThread = 1)

# Use parallel processing and parse data in MRN_Type and MRN columns and keep all IDs
d_hnp <- load_notes(file = "test_Hnp.txt", type = "hnp", nThread = 20, mrn_type = TRUE, perc = 1)

## End(Not run)

load_phy

Loads helath history information into R.

Description

Loads vital signs, social history, immunizations, and various other health history details into the R environment.
**Usage**

```r
load_phy(
  file,
  merge_id = "EMPI",
  sep = ":",
  id_length = "standard",
  perc = 0.6,
  na = TRUE,
  identical = TRUE,
  nThread = 4,
  mrn_type = FALSE
)
```

**Arguments**

- **file** string, full file path to Phy.txt.
- **merge_id** string, column name to use to create `ID_MERGE` column used to merge different datasets. Defaults to `EPIC_PMRN`, as it is the preferred MRN in the RPDR system.
- **sep** string, divider between hospital ID and MRN. Defaults to `:`.
- **id_length** string, indicating whether to modify MRN length based-on required values. `id_length = standard` or to keep lengths as is `id_length = asis`. If `id_length = standard` then in case of `MGH, BWH, MCL, EMPI and PMRN` the length of the MRNs are corrected accordingly by adding zeros, or removing numeral from the beginning. In other cases the lengths are unchanged. Defaults to `standard`.
- **perc** numeric, a number between 0-1 indicating which parsed ID columns to keep. Data present in `perc x 100%` of patients are kept.
- **na** boolean, whether to remove columns with only NA values. Defaults to `TRUE`.
- **identical** boolean, whether to remove columns with identical values. Defaults to `TRUE`.
- **nThread** integer, number of threads to use to load data.
- **mrn_type** boolean, should data in `MRN_Type` and `MRN` be parsed. Defaults to `FALSE`, as it is not advised to parse these for all data sources as it takes considerable time.

**Value**

data table, with health history information.

- **ID_MERGE** numeric, defined IDs by `merge_id`, used for merging later.
- **ID_phy_EMPI** string, Unique Partners-wide identifier assigned to the patient used to consolidate patient information from `phy` datasource, corresponds to EMPI in RPDR. Data is formatted using `pretty_mrn()`.
- **ID_phy_PMRN** string, Epic medical record number. This value is unique across Epic instances within the Partners network from `phy` datasource, corresponds to `EPIC_PMRN` in RPDR. Data is formatted using `pretty_mrn()`.
- **ID_phy_loc** string, if `mrn_type == TRUE`, then the data in `MRN_Type` and `MRN` are parsed into IDs corresponding to locations (loc). Data is formatted using `pretty_mrn()`.
**time_phy** POSIXct, Date when the diagnosis was noted, corresponds to Date in RPDR. Converted to POSIXct format.

**phy_name** string, Type of clinical value/observation recorded, corresponds to Concept_Name in RPDR.

**phy_code** string, Source-specific identifier for the specific type of clinical observation, corresponds to Code in RPDR.

**phy_code_type** string, Source system for the value, corresponds to Code_type in RPDR.

**phy_result** string, Value associated with the clinical observation. Note: BMI results are calculated internally in the RPDR, corresponds to Results in RPDR. Punctuation marks and white spaces are removed.

**phy_unit** string, Units associated with the clinical observation, corresponds to Units in RPDR. Punctuation marks and white spaces are removed.

**phy_provider** string, Provider of record for the encounter where the observation was recorded, corresponds to Providers in RPDR. Punctuation marks are removed.

**phy_clinic** string, Specific department/location where the patient observation was recorded, corresponds to Clinic in RPDR.

**phy_hosp** string, Facility where the observation was recorded, corresponds to Hospital in RPDR.

**phy_inpatient** string, Classifies the type of encounter where the observation was entered, corresponds to Inpatient_Outpatient in RPDR. Punctuation marks are removed.

**phy_enc_num** string, Unique identifier of the record/visit. This values includes the source system and a unique identifier within the source system, corresponds to Encounter_number in RPDR.

**Examples**

```r
## Not run:
#Using defaults
d_phy <- load_phy(file = "test_Phy.txt")

#Use sequential processing
d_phy <- load_phy(file = "test_Phy.txt", nThread = 1)

#Use parallel processing and parse data in MRN_Type and MRN columns and keep all IDs
d_phy <- load_phy(file = "test_Phy.txt", nThread = 20, mrn_type = TRUE, perc = 1)

## End(Not run)
```

### load_prc

*Loads procedures into R.*

**Description**

Loads Clinical procedure information into the R environment.
Usage

```r
load_prc(
    file,
    merge_id = "EMPI",
    sep = ":",
    id_length = "standard",
    perc = 0.6,
    na = TRUE,
    identical = TRUE,
    nThread = 4,
    mrn_type = FALSE
)
```

Arguments

- **file**: string, full file path to Prc.txt.
- **merge_id**: string, column name to use to create ID_MERGE column used to merge different datasets. Defaults to *EPIC_PMRN*, as it is the preferred MRN in the RPDR system.
- **sep**: string, divider between hospital ID and MRN. Defaults to ":".
- **id_length**: string, indicating whether to modify MRN length based-on required values. `id_length = standard` or to keep lengths as is `id_length = asis`. If `id_length = standard` then in case of MGH, BWH, MCL, EMPI and PMRN the length of the MRNs are corrected accordingly by adding zeros, or removing numeral from the beginning. In other cases the lengths are unchanged. Defaults to `standard`.
- **perc**: numeric, a number between 0-1 indicating which parsed ID columns to keep. Data present in `perc x 100%` of patients are kept.
- **na**: boolean, whether to remove columns with only NA values. Defaults to *TRUE*.
- **identical**: boolean, whether to remove columns with identical values. Defaults to *TRUE*.
- **nThread**: integer, number of threads to use to load data.
- **mrn_type**: boolean, should data in `MRN_Type` and `MRN` be parsed. Defaults to *FALSE*, as it is not advised to parse these for all data sources as it takes considerable time.

Value

data table, with procedural information.

- **ID_MERGE**: numeric, defined IDs by `merge_id`, used for merging later.
- **ID_prc_EMPI**: string, Unique Partners-wide identifier assigned to the patient used to consolidate patient information from `prc` datasource, corresponds to EMPI in RPDR. Data is formatted using `pretty_mrn()`.
- **ID_prc_PMRN**: string, Epic medical record number. This value is unique across Epic instances within the Partners network from `prc` datasource, corresponds to EPIC_PMRN in RPDR. Data is formatted using `pretty_mrn()`.
- **ID_prc_loc**: string, if `mrn_type == TRUE`, then the data in `MRN_Type` and `MRN` are parsed into IDs corresponding to locations (`loc`). Data is formatted using `pretty_mrn()`.
time_prc  POSIXct, Date when the procedure was performed, corresponds to Date in RPDR. Converted to POSIXct format.

prc_name  string, Name of the procedure or operation performed, corresponds to Procedure_Name in RPDR.

prc_code  string, Procedure code associated with the "Code_type" value, corresponds to Code in RPDR.

prc_code_type  string, Standardized classification system or custom source value associated with the procedure code, corresponds to Code_type in RPDR.

prc_flag  string, Qualifier for the diagnosis, corresponds to Procedure_Flag in RPDR.

prc_quantity  string, Number of the procedures that were ordered for this record, corresponds to Quantity in RPDR.

prc_provider  string, Provider identifies the health care clinician performing the procedure, corresponds to Provider in RPDR. Punctuation marks are removed.

prc_clinic  string, Specific department/location where the procedure was ordered or performed, corresponds to Clinic in RPDR.

prc_hosp  string, Facility where the procedure was ordered or performed, corresponds to Hospital in RPDR.

prc_inpatient  string, classifies the type of encounter where the procedure was performed or ordered, Punctuation marks are removed.

prc_enc_num  string, Unique identifier of the record/visit, displayed in the following format: Source System - Institution Number, corresponds to Encounter_number in RPDR.

Examples

```r
## Not run:
#Using defaults
d_prc <- load_prc(file = "test_Prc.txt")

#Use sequential processing
d_prc <- load_prc(file = "test_Prc.txt", nThread = 1)

#Use parallel processing and parse data in MRN_Type and MRN columns and keep all IDs
d_prc <- load_prc(file = "test_Prc.txt", nThread = 20, mrn_type = TRUE, perc = 1)

## End(Not run)
```

load_rdt  Loads radiology procedures data into R.

Description

Loads radiology procedures information into the R environment.
Usage

load_rdt(
    file,
    merge_id = "EMPI",
    sep = ":",
    id_length = "standard",
    perc = 0.6,
    na = TRUE,
    identical = TRUE,
    nThread = 4,
    mrn_type = FALSE
)

Arguments

file string, full file path to Rdt.txt.
merge_id string, column name to use to create ID_MERGE column used to merge different datasets. Defaults to EPIC_PMRN, as it is the preferred MRN in the RPDR system.
sep string, divider between hospital ID and MRN. Defaults to :.
id_length string, indicating whether to modify MRN length based-on required values id_length = standard, or to keep lengths as is id_length = asis. If id_length = standard then in case of MGH, BWH, MCL, EMPI and PMRN the length of the MRNs are corrected accordingly by adding zeros, or removing numeral from the beginning. In other cases the lengths are unchanged. Defaults to standard.
perc numeric, a number between 0-1 indicating which parsed ID columns to keep. Data present in perc x 100% of patients are kept.
na boolean, whether to remove columns with only NA values. Defaults to TRUE.
identical boolean, whether to remove columns with identical values. Defaults to TRUE.
nThread integer, number of threads to use to load data.
mrn_type boolean, should data in MRN_Type and MRN be parsed. Defaults to FALSE, as it is not advised to parse these for all data sources as it takes considerable time.

Value
data table, with radiological exam information.

ID_MERGE numeric, defined IDs by merge_id, used for merging later.
ID_rdt_EMPI string. Unique Partners-wide identifier assigned to the patient used to consolidate patient information from rdt datasource, corresponds to EMPI in RPDR. Data is formatted using pretty_mrn().
ID_rdt_PMRN string, Epic medical record number. This value is unique across Epic instances within the Partners network from rdt datasource, corresponds to EPIC_PMRN in RPDR. Data is formatted using pretty_mrn().
ID_rdt_loc string, if mrn_type == TRUE, then the data in MRN_Type and MRN are parsed into IDs corresponding to locations (loc). Data is formatted using pretty_mrn().
time_rdt_exam  POSIXct, Date of the radiology exam, corresponds to Date in RPDR. Converted to POSIXct format.

rdt_mode  string, Modality of the exam, corresponds to Mode in RPDR.

rdt_group  string, Higher-level grouping concept used to consolidate similar procedures across hospitals, corresponds to Group in RPDR.

rdt_test_code  string, Internal identifier for the procedure used by the source system, corresponds to Test_Code in RPDR.

rdt_test_desc  string, Full name of the exam/study performed, corresponds to Test_Description in RPDR.

rdt_accession  string, Identifier assigned to the report or procedure for Radiology tracking purposes, corresponds to Accession_Number in RPDR.

rdt_provider  string, Ordering or authorizing provider for the study, corresponds to Provider in RPDR. Punctuation marks are removed.

rdt_clinic  string, Specific department/location where the procedure was ordered or performed, corresponds to Clinic in RPDR.

rdt_hosp  string, Facility where the order was entered, corresponds to Hospital in RPDR.

rdt_inpatient  string, Classifies the type of encounter where the procedure was performed, corresponds to Inpatient_Outpatient in RPDR. Punctuation marks are removed.

Examples

## Not run:
#Using defaults
d_rdt <- load_rdt(file = "test_Rdt.txt")

#Use sequential processing
d_rdt <- load_rdt(file = "test_Rdt.txt", nThread = 1)

#Use parallel processing and parse data in MRN_Type and MRN columns and keep all IDs
d_rdt <- load_rdt(file = "test_Rdt.txt", nThread = 20, mrn_type = TRUE, perc = 1)

## End(Not run)

---

load_rfv  Loads reason for visit data into R.

Description

Loads reason for visit information into the R environment.
load_rfv

Usage

load_rfv(
  file,
  merge_id = "EMPI",
  sep = ":",
  id_length = "standard",
  perc = 0.6,
  na = TRUE,
  identical = TRUE,
  nThread = 4,
  mrn_type = FALSE
)

Arguments

file string, full file path to Rfv.txt.
merge_id string, column name to use to create ID_MERGE column used to merge different datasets. Defaults to EPIC_PMRN, as it is the preferred MRN in the RPDR system.
sep string, divider between hospital ID and MRN. Defaults to :
id_length string, indicating whether to modify MRN length based-on required values id_length = standard, or to keep lengths as is id_length = asis. If id_length = standard then in case of MGH, BWH, MCL, EMPI and PMRN the length of the MRNs are corrected accordingly by adding zeros, or removing numeral from the beginning. In other cases the lengths are unchanged. Defaults to standard.
perc numeric, a number between 0-1 indicating which parsed ID columns to keep. Data present in perc x 100% of patients are kept.
na boolean, whether to remove columns with only NA values. Defaults to TRUE.
identical boolean, whether to remove columns with identical values. Defaults to TRUE.
nThread integer, number of threads to use to load data.
mrn_type boolean, should data in MRN_Type and MRN be parsed. Defaults to FALSE, as it is not advised to parse these for all data sources as it takes considerable time.

Value
data table, with reason for visit information.

ID_MERGE numeric, defined IDs by merge_id, used for merging later.
ID_rfv_EMPI string, Unique Partners-wide identifier assigned to the patient used to consolidate patient information from dia datasource, corresponds to EMPI in RPDR. Data is formatted using pretty_mrn().
ID_rfv_PMRN string, Epic medical record number. This value is unique across Epic instances within the Partners network from rfv datasource, corresponds to EPIC_PMRN in RPDR. Data is formatted using pretty_mrn().
ID_rfv_loc string, if mrn_type == TRUE, then the data in MRN_Type and MRN are parsed into IDs corresponding to locations (loc). Data is formatted using pretty_mrn().
time_rfv_start POSIXct, Start date of the encounter, corresponds to Start_Date in RPDR. Converted to POSIXct format.

time_rfv_end POSIXct, End date of the encounter, corresponds to End_Date in RPDR. Converted to POSIXct format.

rfv_provider string, Primary provider for the encounter, corresponds to Provider in RPDR. Punctuation marks are removed.

rfv_hosp string, Facility where the encounter occurred, corresponds to Hospital in RPDR.

rfv_clinic string, Specific department/location where the patient encounter took place, corresponds to Clinic in RPDR.

rfv_chief_complaint string, Description of the chief complaint/reason for visit, corresponds to Chief_Complaint in RPDR.

rfv_concept_id string, Epic identifier for the chief complaint/reason for visit, corresponds to Concept_id in RPDR.

rfv_comment string, Free-text comments regarding the chief complaint/reason for visit, corresponds to Comments in RPDR.

rfv_enc_numb string, Unique identifier of the record/visit. This value includes the source system, hospital, and a unique identifier within the source system, corresponds to Encounter_number in RPDR.

Examples

```r
## Not run:
# Using defaults
d_rfv <- load_rfv(file = "test_Rfv.txt")

# Use sequential processing
d_rfv <- load_rfv(file = "test_Rfv.txt", nThread = 1)

# Use parallel processing and parse data in MRN_Type and MRN columns and keep all IDs
d_rfv <- load_rfv(file = "test_Rfv.txt", nThread = 20, mrn_type = TRUE, perc = 1)

## End(Not run)
```

pretty_mrn

Converts MRN integer to string compatible with RPDR.

Description

Adds or removes zeros from integers to comply with MRN code standards for given institution and adds institution prefix.

Usage

```
pretty_mrn(v, prefix = "MGH", sep = ":", id_length = "standard", nThread = 1)
```
pretty_mrn

Arguments

- `v` vector, integer or sting vector with MRNs.
- `prefix` string or vector, hospital ID from where the MRNs are from. Defaults to `MGH`. If a vector is provided then it must be the same length as `v`. This allows to potentially use different prefixes for different IDs using the same vector of values.
- `sep` string, divider between hospital ID and MRN. Defaults to `:`.
- `id_length` string, indicating whether to modify MRN length based-on required values `id_length = standard`, or to keep lengths as is `id_length = asis`. If `id_length = standard` then in case of `MGH, BWH, MCL, EMPI and PMRN` the length of the MRNs are corrected accordingly by adding zeros, or removing numeral from the beginning. In other cases the lengths are unchanged. Defaults to `standard`.
- `nThread` integer, number of threads to use by `dopar` for parallelization. If it is set to 1, then no parallel backends are created and the function is executed sequentially. On windows machines sockets are used, while on other operating systems fork parallelization is used.

Value

vector, with characters formatted to specified lengths. If length of the ID does not match the required length, then leading zeros are added to the ID. If the ID is longer then the required length, then numerals from the beginning of the ID are cut off until it is the required length.

Examples

```r
## Not run:
mrn <- sample(1e4:1e7, size = 10) #Simulate MRNs

#MGH format
test_mrn(v = mrns, prefix = "MGH")

#BWH format
test_mrn(v = mrns, prefix = "BWH")

#Multiple sources using space as a separator
test_mrn(v = mrns[1:3], prefix = c("MGH", "BWH", "EMPI"), sep = " ")

#Keeping the length of the IDs despite not adhering to the requirements
test_mrn(v = mrns, prefix = "EMPI", id_length = "asis")

## End(Not run)
```
Index

all_ids_mi2b2, 3
convert_dia, 3
convert_enc, 5
convert_lab, 6
convert_med, 7
convert_notes, 9
convert_phy, 11
convert_prc, 12
convert_rfv, 14
create_img_db, 15
load_con, 17
load_con_raw, 18
data_con, 17
load_dem, 20
data_dem, 20
data_dem_raw, 21
load_dem, 20
data_dem_raw, 21
load_dia, 22
data_dia, 22
data_dia_raw, 23
load_enc, 24
data_enc, 24
data_enc_raw, 26
load_lab, 28
data_lab, 28
data_lab_raw, 30
load_med, 31
data_med, 31
data_med_raw, 32
load_mrn, 34
data_mrn, 34
data_mrn_raw, 35
load_notes, 64
load_notes, 64
load_med, 60
load_med, 60
load_rdt, 70
load_rdt, 70
load_prc, 68
load_prc, 68
load_rfv, 72
load_rfv, 72
load_mrn, 62
load_mrn, 62
load_rdt, 70
load_rdt, 70
load_rfv, 72
load_rfv, 72
load_lno, 58
load_lno, 58
load_med, 60
load_med, 60
load_phy, 66
load_phy, 66
load_rdt, 70
load_rdt, 70
load_rfv, 72
load_rfv, 72
pretty_mrn, 74
find_exam, 40
load_all, 42
load_con, 44
load_dem, 46
load_dem, 46
load_notes, 64
load_notes, 64
load_enc, 53
load_enc, 53
load_lab, 56
load_lab, 56