medExtractR Vignette

2022-06-06

Introduction

The `medExtractR` package uses a natural language processing (NLP) system called *medExtractR*.[1] This system is a medication extraction system that uses regular expressions and rule-based approaches to identify key dosing information including drug name, strength, dose amount, frequency or intake time, dose change, and last dose time. Function arguments can be specified to allow the user to tailor the `medExtractR` system to the particular drug or dataset of interest, improving the quality of extracted information.

The `medExtractR` system forms the basis of the *Extract-Med* module in Choi *et al.*'s[2] pipeline approach for performing pharmacokinetic/pharmacodynamic (PK/PD) analyses using electronic health records (EHRs). This approach and corresponding R package, `EHR`,[3] convert raw output from `medExtractR` into a format that is usable for PK/PD analyses. Since `medExtractR` is integral to the *Extract-Med* module in `EHR`, parts of this vignette are taken and adapted from the `EHR` package vignette.

Basic `medExtractR`

The function `medExtractR` is primarily responsible for identifying and creating search windows for all mentions of the drug of interest within a note. This function then calls the `extract_entities` subfunction, which identifies and extracts entities within the search window. The entities that can be identified with the basic version of `medExtractR` include: drug name (entity name in output: "DrugName"), strength ("Strength"), dose amount ("DoseAmt"), dose given intake ("DoseStrength"), frequency ("Frequency"), intake time ("IntakeTime"), keywords indicating an increase or decrease in dose ("DoseChange"), route of administration ("Route"), duration of dosing regimen ("Duration"), and time of last dose ("LastDose"). In order to run `medExtractR`, certain function arguments must be specified, including:

- `note`: A character string containing the note on which you want to run `medExtractR`.

- `drug_names`: Names of the drugs for which we want to extract medication dosing information. This can include any way in which the drug name might be represented in the clinical note, such as generic name (e.g., `"lamotrigine"`), brand name (e.g., `"Lamictal"`), or an abbreviation (e.g., `"LTG"`).

- `unit`: The unit of the drug(s) listed in `drug_names`, for example `"mg"`.

- `window_length`: Length of the search window around each found drug name in which to search for dosing information. There is no default for this argument, requiring the user to carefully consider its value through tuning (see tuning section below).

- `max_dist`: The maximum edit distance allowed when identifying `drug_names`. Maximum edit distance determines the difference between two strings, and is defined as the number of insertions, deletions, or substitutions required to change one string into the other. This allows us to capture misspellings in the drug names we are searching for, and its value should be carefully considered through tuning (see tuning section below).

  - The default value is '0', or exact spelling matches to `drug_names`. A value of 0 is always used for drug names with less than 5 characters regardless of the value set by `max_dist`.
  - A value of 1 would capture mistakes such as a single missing or extra letter, e.g., "tacrlimus" or "tacroolimus" instead of "tacrolimus"
  - A value of 2 would capture these mistakes or a single transposition, e.g., "tcarolimus" instead of "tacrolimus"
  - Higher values (3 or above) would capture increasingly more severe mistakes, though setting the value too high can cause similar words to be mistaken as the drug name, likely increasing the false positive rate.

Generally, the function call to `medExtractR` is

```
note <- paste(scan(filename, '', sep = '\n', quiet = TRUE), collapse = '\n')
medExtractR(note, drug_names, unit, window_length, max_dist, ...)
```

where `...` refers to additional arguments to `medExtractR`. Examples of additional arguments include:

- `drug_list`, a list of other drug names (besides the drug names of interest). This list is used to shorten the search window in which `medExtractR` looks for dosing entities by truncating at the nearest mentions of a competing drug name. By default, this calls `rxnorm_druglist`, a partially cleaned and processed list of brand name and ingredient drug names in the RxNorm database.[4] This list could also incorporate other competing information besides drug names, such as drug abbreviations, symptoms, procedures, or names of laboratory measurements.
- `strength_sep`, where users can specify special characters to separate doses administered at different times of day. For example, consider the drug name *"lamotrigine"* and the phrase *"Patient is on lamotrigine 200-300"*, indicating that the patient takes 200 mg of the drug in the morning and 300 mg in the evening. Setting `strength_sep = c('-')` would allow `medExtractR` to identify the expression *200-300* as "DoseStrength" (i.e., dose given intake) since they are separated by the special character "-". The default value is `NULL`.
- `lastdose`, a logical input specifying whether or not the last dose time entity should be extracted. Default value is `FALSE`.
- `<entity>_dict` and `<entity>_fun`, where `<entity>` is a dictionary-based entity (e.g., frequency, intake time, route, duration). These optional arguments allow for user-customized dictionaries and extraction functions. Default dictionaries are provided within `medExtractR`, as is a default extraction function (`extract_generic`).

As mentioned above, some arguments to `medExtractR` should be specified through a tuning process. In a later section, we briefly describe the process by which a user could tune the `medExtractR` system using a validated gold standard dataset.

Running `medExtractR`

Below, we demonstrate how to run `medExtractR` using sample notes for two drugs: tacrolimus (simpler prescription patterns, used to prevent rejection after organ transplant) and lamotrigine (more complex prescription patterns, used to treat epilepsy). The arguments specified for each drug here were determined based on training sets of 60 notes for each drug.[1] We specify `lastdose=TRUE` for tacrolimus to extract information about time of last dose, and `strength_sep="-"` for lamotrigine which can have varying doses depending on the time of day.

```
library(medExtractR)

# tacrolimus note file names
```

```r
tac_fn <- list(
  system.file("examples", "tacpid1_2008-06-26_note1_1.txt", package = "medExtractR"),
  system.file("examples", "tacpid1_2008-06-26_note2_1.txt", package = "medExtractR"),
  system.file("examples", "tacpid1_2008-12-16_note3_1.txt", package = "medExtractR")
)

# execute medExtractR
tac_mxr <- do.call(rbind, lapply(tac_fn, function(filename){
  tac_note <- paste(scan(filename, '', sep = '\n', quiet = TRUE), collapse = '\n')
  fn <- sub(".+/", "", filename)
  cbind("filename" = fn,
        medExtractR(note = tac_note,
             drug_names = c("tacrolimus", "prograf", "tac", "tacro", "fk", "fk506"),
             unit = "mg",
             window_length = 60,
             max_dist = 2,
             lastdose=TRUE))
}))

# lamotrigine note file name
lam_fn <- c(
  system.file("examples", "lampid1_2016-02-05_note4_1.txt", package = "medExtractR"),
  system.file("examples", "lampid1_2016-02-05_note5_1.txt", package = "medExtractR"),
  system.file("examples", "lampid2_2008-07-20_note6_1.txt", package = "medExtractR"),
  system.file("examples", "lampid2_2012-04-15_note7_1.txt", package = "medExtractR")
)

# execute medExtractR
lam_mxr <- do.call(rbind, lapply(lam_fn, function(filename){
  lam_note <- paste(scan(filename, '', sep = '\n', quiet = TRUE), collapse = '\n')
  fn <- sub(".+/", "", filename)
  cbind("filename" = fn,
        medExtractR(note = lam_note,
              drug_names = c("lamotrigine", "lamotrigine XR",
                             "lamictal", "lamictal XR",
                             "LTG", "LTG XR"),
              unit = "mg",
              window_length = 130,
              max_dist = 1,
              strength_sep="-"))
}))
```

The format of raw output from the `medExtractR` function is a `data.frame` with 3 columns:

- `entity`: The label of the entity for the extracted expression.
- `expr`: Expression extracted from the clinical note.
- `pos`: Position of the extracted expression in the note, in the format `startPosition:stopPosition`. Note that we slightly modify the stop position by adding one to avoid output for single-character entities appearing to have zero length (for example, `entity expr pos` output of `DoseAmt 2 33:33`)

In the output presented below, we manually attached the corresponding file name to each note's output before combining results across notes.

tacrolimus `medExtractR` output:

```
##                          filename       entity        expr       pos
## 1  tacpid1_2008-06-26_note1_1.txt     DrugName     Prograf 1219:1226
## 2  tacpid1_2008-06-26_note1_1.txt     Strength        1 mg 1227:1231
## 3  tacpid1_2008-06-26_note1_1.txt      DoseAmt           3 1236:1237
## 4  tacpid1_2008-06-26_note1_1.txt        Route    by mouth 1247:1255
## 5  tacpid1_2008-06-26_note1_1.txt    Frequency twice a day 1256:1267
## 6  tacpid1_2008-06-26_note1_1.txt     LastDose        10PM 1278:1282
## 7  tacpid1_2008-06-26_note1_1.txt     DrugName     porgraf 3873:3880
## 8  tacpid1_2008-06-26_note1_1.txt DoseStrength         3mg 3881:3884
## 9  tacpid1_2008-06-26_note1_1.txt    Frequency         bid 3885:3888
## 10 tacpid1_2008-06-26_note2_1.txt     DrugName     Prograf   618:625
## 11 tacpid1_2008-06-26_note2_1.txt        Route        Oral   626:630
## 12 tacpid1_2008-06-26_note2_1.txt     Strength        1 mg   639:643
## 13 tacpid1_2008-06-26_note2_1.txt      DoseAmt           3   644:645
## 14 tacpid1_2008-06-26_note2_1.txt        Route    by mouth   655:663
## 15 tacpid1_2008-06-26_note2_1.txt    Frequency twice a day   664:675
## 16 tacpid1_2008-06-26_note2_1.txt     LastDose       14 hr   678:683
## 17 tacpid1_2008-12-16_note3_1.txt     DrugName  Tacrolimus   722:732
## 18 tacpid1_2008-12-16_note3_1.txt        Route        Oral   733:737
## 19 tacpid1_2008-12-16_note3_1.txt     DrugName     Prograf   761:768
## 20 tacpid1_2008-12-16_note3_1.txt     Strength        1 mg   770:774
## 21 tacpid1_2008-12-16_note3_1.txt      DoseAmt           3   775:776
## 22 tacpid1_2008-12-16_note3_1.txt        Route    by mouth   786:794
## 23 tacpid1_2008-12-16_note3_1.txt    Frequency twice a day   795:806
## 24 tacpid1_2008-12-16_note3_1.txt   DoseChange    decrease 2170:2178
## 25 tacpid1_2008-12-16_note3_1.txt     DrugName     Prograf 2179:2186
## 26 tacpid1_2008-12-16_note3_1.txt DoseStrength         2mg 2190:2193
## 27 tacpid1_2008-12-16_note3_1.txt    Frequency         bid 2194:2197
## 28 tacpid1_2008-12-16_note3_1.txt     DrugName     Prograf 2205:2212
## 29 tacpid1_2008-12-16_note3_1.txt     LastDose    10:30 pm 2231:2239
```

lamotrigine `medExtractR` output:

```
##                          filename       entity           expr       pos
## 1  lampid1_2016-02-05_note4_1.txt     DrugName       Lamictal   810:818
## 2  lampid1_2016-02-05_note4_1.txt DoseStrength         300 mg   819:825
## 3  lampid1_2016-02-05_note4_1.txt    Frequency            BID   826:829
## 4  lampid1_2016-02-05_note4_1.txt     DrugName    Lamotrigine   847:858
## 5  lampid1_2016-02-05_note4_1.txt     Strength          200mg   859:864
## 6  lampid1_2016-02-05_note4_1.txt      DoseAmt            1.5   865:868
## 7  lampid1_2016-02-05_note4_1.txt    Frequency    twice daily   873:884
## 8  lampid1_2016-02-05_note4_1.txt     DrugName Lamotrigine XR   954:968
## 9  lampid1_2016-02-05_note4_1.txt     Strength         100 mg   969:975
## 10 lampid1_2016-02-05_note4_1.txt      DoseAmt              3 1000:1001
## 11 lampid1_2016-02-05_note4_1.txt        Route       by mouth 1010:1018
## 12 lampid1_2016-02-05_note4_1.txt   IntakeTime  every morning 1019:1032
## 13 lampid1_2016-02-05_note4_1.txt      DoseAmt              2 1037:1038
## 14 lampid1_2016-02-05_note4_1.txt        Route       by mouth 1047:1055
## 15 lampid1_2016-02-05_note4_1.txt   IntakeTime  every evening 1056:1069
## 16 lampid1_2016-02-05_note4_1.txt     DrugName       Lamictal 1915:1923
## 17 lampid1_2016-02-05_note4_1.txt     Duration       2 months 1952:1960
## 18 lampid1_2016-02-05_note5_1.txt     DrugName            ltg   442:445
## 19 lampid1_2016-02-05_note5_1.txt     Strength         200 mg   446:452
```

```
## 20 lampid1_2016-02-05_note5_1.txt      DoseAmt            1.5   454:457
## 21 lampid1_2016-02-05_note5_1.txt    Frequency          daily   459:464
## 22 lampid1_2016-02-05_note5_1.txt     DrugName         ltg xr   465:471
## 23 lampid1_2016-02-05_note5_1.txt     Strength         100 mg   472:478
## 24 lampid1_2016-02-05_note5_1.txt      DoseAmt              3   479:480
## 25 lampid1_2016-02-05_note5_1.txt   IntakeTime          in am   481:486
## 26 lampid1_2016-02-05_note5_1.txt      DoseAmt              2   488:489
## 27 lampid1_2016-02-05_note5_1.txt   IntakeTime          in pm   490:495
## 28 lampid1_2016-02-05_note5_1.txt     DrugName Lamotrigine XR 1125:1139
## 29 lampid1_2016-02-05_note5_1.txt DoseStrength        300-200 1140:1147
## 30 lampid2_2008-07-20_note6_1.txt     DrugName    lamotrigine 1267:1278
## 31 lampid2_2008-07-20_note6_1.txt     DrugName       lamictal 1280:1288
## 32 lampid2_2008-07-20_note6_1.txt DoseStrength         150 mg 1289:1295
## 33 lampid2_2008-07-20_note6_1.txt        Route             po 1296:1298
## 34 lampid2_2008-07-20_note6_1.txt    Frequency           q12h 1299:1303
## 35 lampid2_2008-07-20_note6_1.txt   DoseChange       Increase 2264:2272
## 36 lampid2_2008-07-20_note6_1.txt     DrugName       Lamictal 2273:2281
## 37 lampid2_2008-07-20_note6_1.txt DoseStrength          200mg 2285:2290
## 38 lampid2_2008-07-20_note6_1.txt        Route             po 2291:2293
## 39 lampid2_2008-07-20_note6_1.txt    Frequency            BID 2294:2297
## 40 lampid2_2012-04-15_note7_1.txt     DrugName    lamotrigine   103:114
## 41 lampid2_2012-04-15_note7_1.txt     Strength         150 mg   115:121
## 42 lampid2_2012-04-15_note7_1.txt     DrugName       Lamictal   141:149
## 43 lampid2_2012-04-15_note7_1.txt      DoseAmt              1   151:152
## 44 lampid2_2012-04-15_note7_1.txt        Route       by mouth   160:168
## 45 lampid2_2012-04-15_note7_1.txt    Frequency    twice a day   169:180
```

For the tacrolimus output, we chose to also extract the last dose time entity by specifying `lastdose=TRUE`. The last dose time entity is extracted as raw character expressions from the clinical note, and must first be converted to a standardized datetime format. The `EHR`[3] package provides for parsing and standardizing raw `medExtractR` last dose times when laboratory measurements are available with its `processLastDose` function.

Tuning the `medExtractR` system

In a previous section, we mentioned that parameters within the `medExtractR` should be tuned in order to ensure higher quality of extracted drug information. This section provides recommendations for how to implement this tuning procedure.

In order to tune `medExtractR`, we recommend selecting a small set of tuning notes, from which the parameter values can be selected. Below, we describe this process with a set of three notes (note that these notes were chosen for the purpose of demonstration, and we recommend using tuning sets of at least 10 notes).

Once a set of tuning notes has been curated, they must be manually annotated by reviewers to identify the information that should be extracted. This process produces a gold standard set of annotations, which identify the correct drug information of interest. This includes entities like the drug name, strength, and frequency. For example, in the phrase

Patient is taking **lamotrigine** *300 mg* in the morning and *200 mg* in the evening

bolded, italicized, and underlined phrases represent annotated drug names, dose strength (i.e., dose given intake), and intake times, respectively. These annotations are stored as a dataset.

First, we read in the annotation files for three example tuning notes, which can be generated using an annotation tool, such as the Brat Rapid Annotation Tool (BRAT) software.[5] By default, the output file from

BRAT is tab delimited with 3 columns: an annotation identifier, a column with labeling information in the format "label startPosition stopPosition", and the annotation itself, as shown in the example below:

```
##   id           entity  annotation
## 1 T1   DrugName 19 30 lamotrigine
## 2 T2       Dose 31 37      300 mg
## 3 T3 IntakeTime 45 52     morning
## 4 T4       Dose 57 63      200 mg
## 5 T5 IntakeTime 71 78     evening
```

In order to compare with the `medExtractR` output, the format of the annotation dataset should be four columns with:

1. The file name of the corresponding clinical note
2. The entity label of the annotated expression
3. The annotated expression
4. The start and stop position of the annotated expression in the format "start:stop"

The exact formatting performed below is specific to the format of the annotation files, and may vary if an annotation software other than BRAT is used.

```
# Read in the annotations - might be specific to annotation method/software
ann_filenames <- list(system.file("mxr_tune", "tune_note1.ann", package = "medExtractR"),
                      system.file("mxr_tune", "tune_note2.ann", package = "medExtractR"),
                      system.file("mxr_tune", "tune_note3.ann", package = "medExtractR"))

tune_ann <- do.call(rbind, lapply(ann_filenames, function(fn){
  annotations <- read.delim(fn,
                            header = FALSE, sep = "\t", stringsAsFactors = FALSE,
                            col.names = c("id", "entity", "annotation"))

  # Label with file name
  annotations$filename <- sub(".ann", ".txt", sub(".+/", "", fn), fixed=TRUE)

  # Separate entity information into entity label and start:stop position
  # Format is "entity start stop"
  ent_info <- strsplit(as.character(annotations$entity), split="\\s")
  annotations$entity <- unlist(lapply(ent_info, '[[', 1))
  annotations$pos <- paste(lapply(ent_info, '[[', 2),
                           lapply(ent_info, '[[', 3), sep=":")

  annotations <- annotations[,c("filename", "entity", "annotation", "pos")]

  return(annotations)
}))
head(tune_ann)
```

```
##         filename    entity  annotation       pos
## 1 tune_note1.txt  DrugName     Prograf 1219:1226
## 2 tune_note1.txt  Strength        1 mg 1227:1231
## 3 tune_note1.txt   DoseAmt           3 1236:1237
## 4 tune_note1.txt     Route    by mouth 1247:1255
## 5 tune_note1.txt Frequency twice a day 1256:1267
## 6 tune_note1.txt  DrugName     porgraf 3873:3880
```

To select appropriate tuning parameters, we identify a range of possible values for each of the `window_length` and `max_dist` parameters. Here, we allow `window_length` to vary from 30 to 120 characters in increments of 30, and `max_dist` to take a value of 0, 1, or 2. We then obtain the `medExtractR` results for each combination.

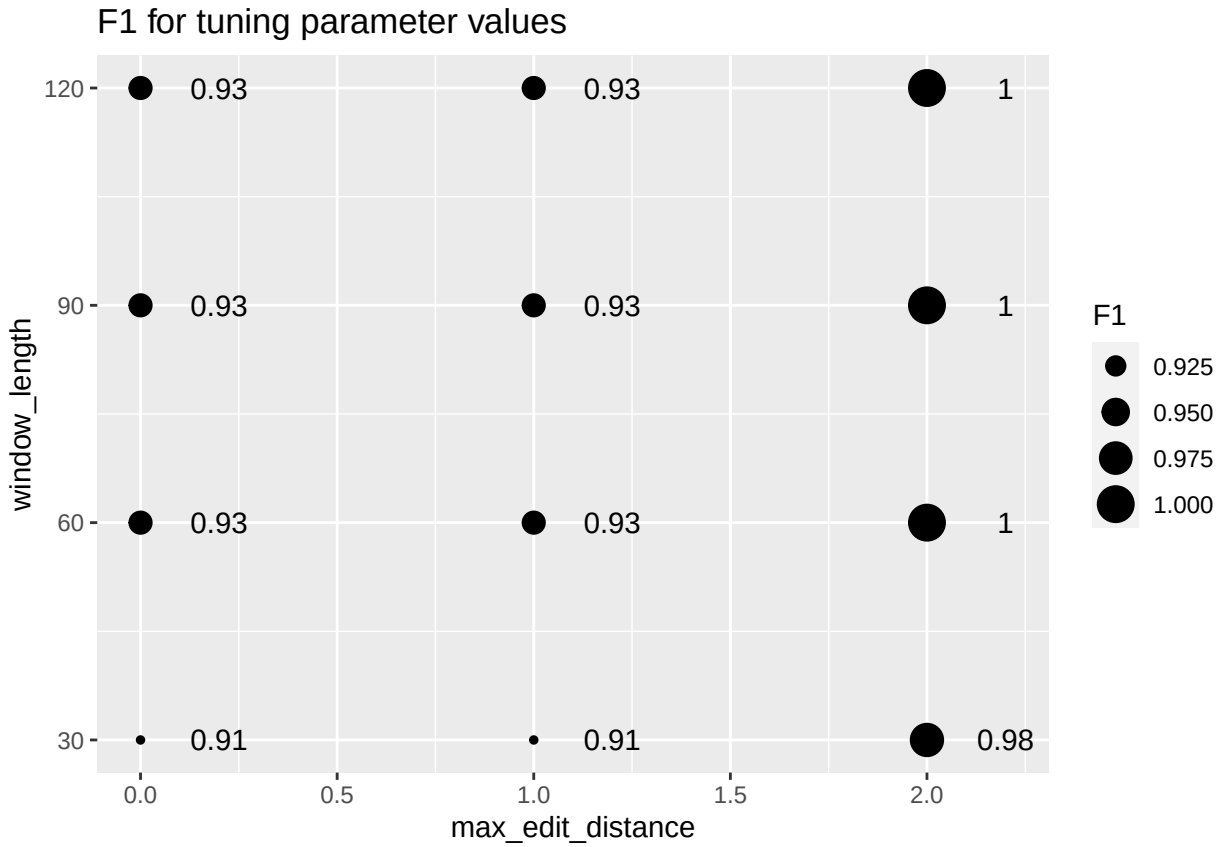
```
wind_len <- seq(30, 120, 30)
max_edit <- seq(0, 2, 1)
tune_pick <- expand.grid("window_length" = wind_len,
                         "max_edit_distance" = max_edit)
# Run the Extract-Med module on the tuning notes
note_filenames <- list(system.file("mxr_tune", "tune_note1.txt", package = "medExtractR"),
                       system.file("mxr_tune", "tune_note2.txt", package = "medExtractR"),
                       system.file("mxr_tune", "tune_note3.txt", package = "medExtractR"))

# List to store output for each parameter combination
mxr_tune <- vector(mode="list", length=nrow(tune_pick))
for(i in 1:nrow(tune_pick)){

  mxr_tune[[i]] <- do.call(rbind, lapply(note_filenames, function(filename){
    tune_note <- paste(scan(filename, '', sep = '\n', quiet = TRUE), collapse = '\n')
    fn <- sub(".+/", "", filename)
    cbind("filename" = fn,
          medExtractR(note = tune_note,
                      drug_names = c("tacrolimus", "prograf", "tac", "tacro", "fk", "fk506"),
                      unit = "mg",
                      window_length = tune_pick$window_length[i],
                      max_dist = tune_pick$max_edit_distance[i]))
  }))

}
```

Finally, we determine which parameter combination yielded the highest performance, quantified by some metric. For our purpose, we used the F1-measure (F1), the harmonic mean of precision $\left(\frac{\text{true positives}}{\text{true positives + false positives}}\right)$ and recall $\left(\frac{\text{true positives}}{\text{true positives + false negatives}}\right)$. Tuning parameters were selected based on which combination maximized F1 performance within the tuning set. The code below determines true positives as well as false positives and negatives, used to compute precision, recall, and F1.



The plot shows that the highest F1 achieved was 1, and occurred for three different combinations of parameter values: a maximum edit distance of 2 and a window length of 60, 90, or 120 characters. The relatively small number of unique F1 values is likely the result of only using 3 tuning notes. In this case, we would typically err on the side of allowing a larger search window and decide to use a maximum edit distance of 2 and a window length of 120 characters. In a real-world tuning scenario and with a larger tuning set, we would also want to test longer window lengths since the best case scenario occurred at the longest window length we used. Additional information for the tuning process of `medExtractR` can be found in Weeks *et al.*[1]

References

1. Weeks HL, Beck C, McNeer E, Williams ML, Bejan CA, Denny JC, Choi L. medExtractR: A targeted, customizable approach to medication extraction from electronic health records. Journal of the American Medical Informatics Association. 2020 Mar;27(3):407-18. doi: 10.1093/jamia/ocz207.

2. Choi L, Beck C, McNeer E, Weeks HL, Williams ML, James NT, Niu X, Abou-Khalil BW, Birdwell KA, Roden DM, Stein CM. Development of a System for Post-marketing Population Pharmacokinetic and Pharmacodynamic Studies using Real-World Data from Electronic Health Records. Clinical Pharmacology & Therapeutics. 2020 Apr;107(4):934-43. doi: 10.1002/cpt.1787.

3. Choi L, Beck C, Weeks HL, and McNeer E (2020). EHR: Electronic Health Record (EHR) Data Processing and Analysis Tool. R package version 0.3-1. https://CRAN.R-project.org/package=EHR

4. Nelson SJ, Zeng K, Kilbourne J, Powell T, Moore R. Normalized names for clinical drugs: RxNorm at 6 years. Journal of the American Medical Informatics Association. 2011 Jul-Aug;18(4)441-8. doi: 10.1136/amiajnl-2011-000116. Epub 2011 Apr 21. PubMed PMID: 21515544; PubMed Central PMCID: PMC3128404.

5. Stenetorp P, Pyysalo S, Topić G, Ohta T, Ananiadou S, Tsujii JI. BRAT: a web-based tool for NLP-assisted text annotation. InProceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics 2012 Apr 23 (pp. 102-107). Association for Computational Linguistics.