



MSc Project Report
2020-2021

**Building and testing R tools to extract
and explore data from the online
database [ClinicalTrials.gov](https://clinicaltrials.gov)**

Arturo de la Cruz Libardi

Submitted in part fulfilment of the requirements for the degree of MSc in
Health Data Science

September 2021

Table of Contents

List of Figures	3
Acknowledgments	4
Abstract	5
1. Introduction	6
1.1 Clinical Trials and Data Science	6
1.2 ClinicalTrials.gov (CTG)	6
1.3 Proposed project: collaboration with UCB	7
1.4 Tools for ClinicalTrials.gov	7
1.5 Motivation	8
1.6 Report Structure	8
2. Aim and Objectives	9
3. Data, Materials, and Methods	9
3.1 R Resources and Package Structure	9
3.2 Data and API	10
3.3 Github	10
3.4 Case-study	10
4. Results	13
4.1 rctapi	13
4.1.1 Development and Description	13
4.1.2 Usage	15
4.1.3 Case study: Searching and Downloading from ClinicalTrials.gov	16
4.2 rctexplorer	17
4.2.1 Development and Description	17
4.2.2 Usage	23
4.2.3 Case study: Systematic filtering of downloaded studies	24
4.2.4 Case study: Visualization and Network tools	26
4.2.5 Case study: Assessment of systematic filtering and trial identification	30
5. Discussion	33
5.1 Limitations and difficulties	34
5.2 Future directions	36
6. Conclusion	37
References	38
Appendices	40

List of Figures

Figure 1 : Title page of Mease et al., (2021).....	11
Figure 2: RStudio view of the object rctapi::field_lists_df.....	14
Figure 3: Example of a working url.....	14
Figure 4: Unparsed API response	16
Figure 5: Rctexplorer banner and tabs	19
Figure 6: Data Table tab from rctexplorer.....	20
Figure 7: Screen capture of stacked barplot and control toggles from rctexplorer Plots tab.	21
Figure 8: Screen capture of the Interventions tab rctexplorer.. ..	22
Figure 9: Filtering variables through categories, and logical and regular expressions	25
Figure 10: Treemap visualization of AgeRange and IsFDARegulatedDrug fields	26
Figure 11: Stacked barplot visualization of LeadSponsor and IsFDARegulatedDrug fields .	27
Figure 12: Scatter plot of LeadSponsorName and DesignMasking fields against number participants enrolled.....	28
Figure 13: Network graph built from 41 filtered studies.....	29
Figure 14: (a)Treatment network for outcome measure ACR20 from Mease et al. (2021). (b) Network recreated in rctexplorer.	32
Figure 15: Mouseover popup of apremilast to placebo edge showing NCTIDs, acronyms and results status.....	33

Acknowledgments

First of all, I would like to thank Fiona Grimson and Vanessa Taieb for developing a stimulating project offer, selecting me to work on it and providing guidance throughout. Thank you also for allowing me to present my work in front of a professional audience.

Thank you to my project supervisor Antonio Gasparrini, who helped me get started on the right footing. To my personal tutor Damien Tully, thank you for keeping a caring eye on your tutees during these exciting times. Thank you also to the entire Health Data Science team for an outstanding inaugural year.

Abstract

The public online database ClinicalTrials.gov (CTG) currently hosts 320.000 clinical trial records. The information in this database is regularly used to provide better care to patients suffering a wide-range of diseases. Accessing this database is easy but the ways of doing so are limited, especially through the R statistical software. For the first time, two R packages take advantage of the new official API to connect to the database. Through the `rctapi` package an R user can carry out a wide-ranging or targeted search of CTG and extract clinical trial records into a locally saved table. With the R-Shiny application housed inside `rctexplorer`, the user can interactively explore the downloaded records.

To test the tools, this report details the attempt to recreate the study selection process for a recent network meta-analysis for the drug Gusellkumab, Mease et al., (2021). A wide search of the database is performed and filtering steps are implemented within the application. The visualisation options are showcased as are the utilities conceived to assess the suitability of the studies. Among them are a flattened table of interventions and a network graph builder, both acting automatically on the previously filtered dataset.

With very few exceptions, the tools were able to narrow a large dataset into one that could be manually examined and that contained all the studies included in the network meta-analysis. The missing studies were either not in the database or were excluded during the filtering due to their misclassification. This process surfaced known data-quality issues within CTG and highlighted why it is essential to incentivise researchers to register through and accurate information.

To see the tools in action, view a demonstration through [this Vimeo link](#).

1. Introduction

Data Science is in a maturing phase, both as a field and as a set of tools with healthcare applications. Many public and private organizations are employing Data Science methods to achieve their goals, educational resources are being devoted to teaching Data Science. As a consequence, data related-technology is taking an ever-important role in the shaping our immediate surroundings, and the platforms and tools we use in our work and daily life.

Data Science feeds on data and data are generated continuously. This remains true in the Healthcare setting, where the gathering and processing of data is happening at a few different paces. Some data are treated at massive volumes. For example, the -omics disciplines, electronic health records, and even hospital staffing, now fall under the optimizing purview of Big Data Science. Some data on the other hand, does not readily lend itself to that treatment, yet. One of these types of data are clinical trial records. A clinical trial is the scientific study of a drug or other medical intervention, and they are the skeleton of the pharmaceutical endeavour.

1.1 Clinical Trials and Data Science

Individually, clinical trial records are primarily used to assess the safety and efficacy of drugs. Ensuring that a drug is both safe and effective is a requirement for its public release and sale.

Collectively, clinical trial data can be used for a variety of purposes. Combining the data is common practice to obtain a more advanced and contextualised understanding of the drugs' quality. Pairwise meta-analyses compare the same two treatments across several studies, coalescing evidence into stronger conclusions. Network meta-analyses (NMA) have similar purposes but can compare more interventions through the calculation of indirect effects.

Although these examples fall into the domain of statistical analysis, they are by no means disconnected from Data Science. Data Science has filled the role of maintaining and updating the programs and processes through which researchers store, gather, and analyse any data: including and most salient to researchers in biomedicine and healthcare, clinical trial records.

1.2 ClinicalTrials.gov (CTG)

Public online databases emerged closely after the release of the first ever web browser in 1991, only six years prior to the origins of clinicaltrials.gov (CTG). However, the concept of

a centralized clinical trial database was not new. In 1989 the AIDS epidemic in the United States brought to attention the need for such a resource (Congress 1988). In 1997, the Food and Drug Administration Modernization Act mandated the National Institute of Health to create a public information repository tracking research in drug development involving human trials (Congress 1997). Three years later, CTG was launched on the world-wide-web for public use (McCray and Ide 2000, Medicine 2000). Until 2002, most records were of publicly funded research. Further legislature facilitated the registration of industry-funded research and in 2007 the Food and Drug Administration Amendments Act made mandatory the inclusion of basic study results (Administration 2002, Congress 2007). This Act made CTG as it appears today: a publicly funded and accessible online database of registration and results information of public and industry-funded clinical trials. The records are self-reported by the researchers, usually at the conception of the trial. Once uploaded and reviewed each receive a unique National Clinical Trial Identifier (NCTId).

1.3 Proposed project: collaboration with UCB

This project's outline was initially designed and offered by individuals working within UCB, an international biomedicine and pharmaceutical company. The broad aim was to build a tool which enabled R users to interface with the online CTG database. More pointedly, the aim was to allow a user to search, extract and visually convey clinical trial registration and results data with the least R knowledge requirement. Ideally, the tools developed would have some application for network meta-analysis, and even more specifically, this application might consist of the transfer of numerical results into the R package netmeta, acting as a processing pipeline to easily gather data for NMA.

1.4 Tools for ClinicalTrials.gov

There have been efforts, both public and industry-backed, to build tools around CTG. Rclinicaltrials, an R package that enabled search and download was published on the Comprehensive R Archive Network (CRAN) in 2014. This tool was recently removed from CRAN as it could not be continuously updated. The most popular way of connecting to CTG is through the Clinical Trial Transformation Initiative's tool Aggregate Analysis of ClinicalTrials.gov (AACT), which allows access to a relational database version of the contents of CTG. Many programs, including R and pgAdmin can connect to the AACT and search and extract CTG data. This is the biggest, most powerful, open-source complement to CTG and is routinely used to study the contents of the database, recently for example, to monitor COVID-19 trials (Mayer and Huser 2020).

The industry projects have gone further in their applications but as they are private tools their exact mechanisms are hidden. SHERLOCK™, created in 2013 by employees of

Janssen, is allegedly capable of creating analysis ready databases from user prompts by automatically downloading CTG data (Cepeda, Lobanov et al. 2013). Three years later, SHERLOCK™ was combined with a newly developed tool to create an automatic NMA computer. According to a published poster, this tool, also property of Janssen, can generate NMA statistics and plots from a combination of user selected and machine processed study results (Karcher, Wiecek et al. 2016). It is possible that this tool is being used to estimate and assist in the construction of NMA and not being used to carry out the actual analysis to be published. This is mentioned because recent examples of SHERLOCK™ being used can easily be identified, but the same cannot be said of the unnamed NMA-building tool (Tan, Kern et al. 2021).

1.5 Motivation

While CTG is publicly accessible, the ways of interacting with it are very limited, especially if one expects to do so directly and through R software. This project attempts to fill this niche and expand the database's utility. The CTG website is well-designed and allows in-depth examination of the studies. However, and understandably, it lacks any functions on top of the simple single-study view. It is not possible to view aggregated characteristics of studies, or to view the complete results of a search of the database on a table. As it does not present aggregated data, it cannot build any visualizations. These absences motivated and guided the project.

1.6 Report Structure

This report concerns the ideation, development, and deployment of software. As such, parts more suited to appear in code documentation have been integrated within. Briefly laid-out next is the structure and contents of the report.

In the following section an overall aim and more practical objectives are set, these are derived from the initial project description and discussion with the UCB partners. After the aims are set, the section *Data, Materials, and Methods* discusses and justifies the software and add-ons utilized as well as any resources used throughout the entirety of the project. The *Results* section describes in depth the development and usage of each package. This is the most extensive section of the report and sets up the subsequent *Discussion* section which touches on limitations of the tools and of the database.

2. Aim and Objectives

The overall aim of the project was to develop an R-based tool that facilitated and expanded interaction with CTG. The resulting tool was intended to be free to use and open to ulterior development after completion of the project.

At the outset, the main aim was split into three problems to solve, each also identifiable as a stage and objective of the project. The first objective was to establish a connection with the online database, perform a search and transfer data onto the user's machine. The second was to provide a way to explore the data without extensive command of R. The third objective was to focus on achieving utility for meta-analyses.

3. Data, Materials, and Methods

3.1 R Resources and Package Structure

R was chosen as the overarching software due to its prevalence in the research and data science setting, and due to its open-source nature (Team 2013). The existence of a system of well-maintained R packages whose utilities can be deployed in combination, allows the construction of flexible and robust tools. R is commonly used for data manipulation, statistical analysis, data visualisation and application building. These tasks are usually accomplished by making use of the thousands of packages gathered and maintained in the CRAN.

An R package is a special type of file folder. This folder generally contains three sub-directories and a four single files.

Folders:

R – Function definitions and data building source code.

Man – Markdown documentation.

Data – Data objects (e.g. lists, dataframes).

Files:

Description – A machine-readable description of the package including a list of other packages that must be imported prior to its loading.

Namespace – Saves the names of objects in a protected context so that whatever these names are they do not interfere with other names in other packages and vice versa.

License – Provides licensing information of the package, indicating if and how it can be shared among other things.

ReadMe – A text file with instructions or any information the developer wants to share (see an example in *Appendix A*).

Throughout this project the package roxygen2 was used to create consistent documentation (Wickham 2020). The saved data objects can be made available to the user or restricted for internal use. The concept and practical use of a namespace in programming is notoriously complex and beyond the scope of this report. Within the packages built for this project, the namespace file is only used to hold the functions that are exported and available to the user. The chosen MIT License allows free and unrestricted reuse of the code conditional on the inclusion of the license and copyright notice.

The tidyverse package suite was used throughout the project for data manipulation (Wickham et al. 2019). The main packages used were glue, tidyr, dplyr, and plyr.

3.2 Data and API

The website clinicaltrials.gov is free to use, and public, meaning it is accessible to anyone with an internet connection. This also means that the contents of the database are mostly free of usage restrictions (clinicaltrials.gov 2014). In 2018, CTG started offering an application programming interface (API) as the official mechanism of querying the database. This is the most direct and flexible way to interact with the stored data. Using the API provides as much range in the queries as using the advanced search options on the website. To the author's knowledge these are the first R packages to employ the new and official API to handle download of data from CTG to R.

3.3 Github

The code files were managed through the Github software suite. Github.com was used to hold the developing and completed project in publicly accessible repositories. This allowed for automated version control and other code management techniques such as branching and merging to be used throughout. All source code can be found at <https://github.com/AdlCruz/rctapi> and <https://github.com/AdlCruz/rctexplorer>.

3.4 Case study

The tools were put to the test in searching and reviewing studies, imitating the criteria followed by a recent network meta-analysis. The researchers carried out a systematic literature review to identify suitable studies to build an evidence network around the drug Guselkumab in the treatment of psoriatic arthritis (Mease, McInnes et al. 2021) (Figure 1). Table 1 gives the principal author and date, name and NCTId of the 26 included trials. It was adapted from the supplementary material as the original did not include NCTId codes

Figure 1: Title page of Mease et al., (2021)

Systematic review and meta analysis

Comparative effectiveness of guselkumab in psoriatic arthritis: results from systematic literature review and network meta-analysis

Philip J. Mease¹, Iain B. McInnes², Lai-Shan Tam³, Kiefer Eaton⁴, Steve Peterson⁵, Agata Schubert⁶, Soumya D. Chakravarty^{7,8}, Anna Parackal⁴, Chetan S. Karyekar⁵, Sandhya Nair⁹, Wolf-Henning Boehncke¹⁰ and Christopher Ritchlin¹¹

Abstract

Objective. The efficacy of the novel interleukin (IL)-23p19 inhibitor guselkumab for psoriatic arthritis (PsA) has recently been demonstrated in two phase 3 trials (DISCOVER-1 & -2) but has not been evaluated vs other targeted therapies for PsA. The objective was to compare guselkumab to targeted therapies for PsA for safety and joint and skin efficacy through network meta-analysis (NMA).

Methods. A systematic literature review was conducted in January 2020 to identify randomized controlled trials. Bayesian NMAs were performed to compare treatments on American College of Rheumatology (ACR) 20/50/70 response, mean change from baseline in van der Heijde-Sharp (vdH-S) score, Psoriasis Area Severity Index (PASI) 75/90/100 response, adverse events (AEs) and serious adverse events (SAEs).

Results. Twenty-six phase 3 studies evaluating 13 targeted therapies for PsA were included. For ACR 20 response, guselkumab 100 mg every 8 weeks (Q8W) was comparable to IL-17A inhibitors and subcutaneous tumor necrosis factor (TNF) inhibitors. Similar findings were observed for ACR 50 and 70. For vdH-S score, guselkumab Q8W was comparable to other agents except intravenous TNF therapies. Results for PASI 75 and PASI 90 response suggested guselkumab Q8W was better than most other agents. For PASI 100, guselkumab Q8W was comparable to other active agents. For AEs and SAEs, guselkumab Q8W ranked highly but comparative conclusions were uncertain. Similar results were observed for all outcomes for guselkumab 100 mg every four weeks.

Conclusions. In this NMA, guselkumab demonstrated favorable arthritis efficacy comparable to IL-17A and subcutaneous TNF inhibitors while offering better PASI response relative to many other treatments.

Key words: guselkumab, psoriatic arthritis, interleukin, TNF, biologics, NMA, SLR, ACR, PASI

Rheumatology key messages

- Guselkumab provides better PASI responses than many other agents available in PsA.
- Guselkumab offers joint efficacy comparable to IL-17A and subcutaneous TNF inhibitors available in PsA.

¹Swedish Medical Center/Providence St. Joseph Health & University of Washington, Seattle, WA, USA, ²University of Glasgow, Centre for Rheumatic Diseases, United Kingdom, ³The Chinese University of Hong Kong and The Prince of Wales Hospital, Department of Medicine & Therapeutics, Hong Kong, ⁴EVERSANA, Marketing and Market Access, Burlington, Ontario, Canada, ⁵Janssen Global Services LLC, Immunology, Global Commercial Strategy Organization, Horsham, PA, USA, ⁶Janssen-Cilag Ltd, Dermatology and Rheumatology, Warsaw, Poland, ⁷Janssen Scientific Affairs LLC, Immunology, Horsham, ⁸Drexel University

College of Medicine, Philadelphia, PA, USA, ⁹Janssen Pharmaceutical NV, Health Economics Design and Analytics, Beerse, Belgium, ¹⁰Geneva University Hospitals, Department of Medicine, Geneva, Switzerland and ¹¹University of Rochester, Department of Medicine, Rochester, NY, USA

Submitted 2 December 2020; accepted 30 January 2021

Correspondence to: Philip Mease, MD, Seattle Rheumatology Associates, 601 Broadway, Suite 600, Seattle, WA 98122, USA. E-mail: pmease@philipmease.com

Table 1: Studies included in the Network Meta-analysis. Adapted from Mease et al., (2021)

Author, Publication Date	Trial Name	NCTId
Nash 2018	ACTIVE	NCT01925768
Mease 2005	ADEPT	NCT00195689
McInnes 2015	FUTURE 2	NCT01752634
Nash 2018	FUTURE 3	NCT01989468
Kivitz 2019	FUTURE 4	NCT02294227
Mease 2018	FUTURE 5	NCT02404350
Kavanaugh 2009	GO-REVEAL	NCT00265096
Kavanaugh 2017	GO-VIBRANT	NCT02181673
Antoni 2005	IMPACT 2	NCT00051623
Genovese 2007	NA	NA
Gladman 2017	OPAL-BEYOND	NCT01882439
Mease 2017	OPAL-BROADEN	NCT01877668
Kavanaugh 2014	PALACE 1	NCT01172938
Cutolo 2016	PALACE 2	NCT01212757
Edwards 2016	PALACE 3	NCT01212770
Wells 2018	PALACE 4	NCT01307423
McInnes 2013	PSUMMIT 1	NCT01009086
Ritchlin 2014	PSUMMIT 2	NCT01077362
Mease 2013	RAPID-PSA	NCT01087788
Mease 2017	SPIRIT-P1	NCT01695239
Nash 2017	SPIRIT-P2	NCT02349295
Mease 2019	SPIRIT-H2H	NCT03151551
Mease 2017	ASTRAEA	NCT01860976
Mease 2004	NA	NA
Janssen 2019	DISCOVER 1	NCT03162796
Janssen 2019	DISCOVER 2	NCT03158285

4. Results

The result of employing the above materials and methods to solve the challenges set out in the Aims section was the creation of two R packages. `rctapi` facilitates the searching and downloading of data making use of CTG's API. `rctexplorer` performs some data manipulation and allows direct access and interaction with the data through a user interface.

4.1 `rctapi`

4.1.1 Development and Description

This package can be most succinctly defined as a wrapper around the ClinicalTrials.gov API. The API is thoroughly documented on the website which clearly delineates the way to perform queries and return data. To query the API, a Uniform Resource Locator (unique address of information on the internet, shortened to "url"), must be compiled following certain specifications. The url carries all the information describing a query in a machine-readable format so that the API can return the relevant data. Given this, there are four tasks the package should perform. First, to collect information from the user. Second, to adequately transform it into a url address. Third, to query and receive the API response. And fourth, to parse this response into an accessible format.

The first customizable part of the url address represents the key used to search the database and is called the search expression. The search expression can be a single word, but pseudo-code syntax set out in the API documentation allows the user to include logic and convey meta-information within the search. The words AND and NOT act as logical operators, and a number of other keywords can be used to create a targeted search. The user can indicate that they want a word to appear in a field by writing "AREA[Field]Term" or to prefer results containing a specified term with "TILT[Area]Term". For example, "AREA[StudyType]Interventional" would return studies of type "Interventional" and exclude those marked as anything else, and "TILT[BriefSummary]Mild" would give preference to studies that contain the word "mild" in their brief summary field.

The second customizable part is the field list. Since each record consists of 322 fields, selecting which parts of the study are important to the user is a necessity. The field list is simply a list of the fields to return. This list could be compiled and saved prior to every query, but this would be very time-consuming and error prone. Prepared lists have been saved as data objects within the package. These lists are gathered by theme (e.g., `registration_fields`, `results_fields`, `eligibility_fields`), but there is also a list of all fields. A saved dataframe named `field_lists_df` collects and characterises each list (Figure 2).

Figure 2: RStudio view of the object `rctapi::field_lists_df`

names	description	fields	length	help
<code>all_fields</code>	All queryable field names	<code>c("Acronym", "AgreementOtherDetails", "A...</code>	322	https://clinicaltrials.gov/api/gui/ref/cros...
<code>core_info_fields</code>	Smallest election of relevant fields providing...	<code>c("NCTId", "OverallStatus", "CompletionDa...</code>	24	https://clinicaltrials.gov/api/gui/ref/cros...
<code>extended_info_fields</code>	Large selection of fields from all registration ...	<code>c("NCTId", "OrgStudyId", "BriefTitle", "Acro...</code>	65	https://clinicaltrials.gov/api/gui/ref/cros...
<code>identification_and_status_fields</code>	Fields including study identification, study st...	<code>c("NCTId", "OrgStudyId", "BriefTitle", "Acro...</code>	21	https://clinicaltrials.gov/api/gui/ref/cros...
<code>study_design_arms_groups_and_interventions_fields</code>	Information about the study design, number...	<code>c("NCTId", "DesignPrimaryPurpose", "Phas...</code>	21	https://clinicaltrials.gov/api/gui/ref/cros...
<code>outcome_measures_info_fields</code>	Information about all outcome measures used	<code>c("NCTId", "PrimaryOutcomeMeasure", "Pri...</code>	10	https://clinicaltrials.gov/api/gui/ref/cros...
<code>eligibility_fields</code>	Information regarding eligibility requirements	<code>c("NCTId", "Gender", "GenderBased", "Gen...</code>	10	https://clinicaltrials.gov/api/gui/ref/cros...
<code>participant_flow_fields</code>	Information regarding the progress of partic...	<code>c("NCTId", "FlowRecruitmentDetails", "Flow...</code>	15	https://clinicaltrials.gov/api/gui/ref/cros...
<code>baseline_characteristics_fields</code>	Information regarding the baseline measure...	<code>c("NCTId", "BaselineGroupTitle", "Baseline...</code>	19	https://clinicaltrials.gov/api/gui/ref/cros...
<code>outcome_measures_results_fields</code>	Information regarding the outcomes measur...	<code>c("NCTId", "OutcomeMeasureType", "Outc...</code>	39	https://clinicaltrials.gov/api/gui/ref/cros...
<code>registration_fields</code>	Large selection of registration data elements...	<code>c("NCTId", "OrgStudyId", "BriefTitle", "Acro...</code>	62	https://clinicaltrials.gov/api/gui/ref/cros...
<code>results_fields</code>	Large selection of results data elements fields.	<code>c("NCTId", "NCTId", "FlowRecruitmentDeta...</code>	74	https://clinicaltrials.gov/api/gui/ref/cros...

There are hundreds of thousands of studies on CTG and for many wide searches such as “heart attack” or “psoriatic arthritis” there can be upwards of 500 matching results. The third customizable part of the url is a number. This number corresponds to the desired maximum number of studies to return from the ones that fulfil the search expression. The fourth changeable part is not acted on by the user, but it regards the format the data is returned in. For all but one purpose, this package receives data in .csv format. Json format is only required when returning an unparsed API response. The format is the last piece of information required before the full url (Figure 3) is compiled and some characters are substituted for their ASCII equivalent.

Figure 3: Example of a working url. The highlighted sections are customizable through the package. In blue, search expression, in green, fields to return, and in yellow, maximum number of studies.

```
https://clinicaltrials.gov/api/query/study_fields?expr=psoriatic+arthritis+TILT%5BBriefSummary%5DMild&fields=NCTId%2CBriefTitle%2CCondition&min_rnk=1&max_rnk=500&fmt=csv
```

Regarding the code itself, two imported functions exemplify the essence of the package. These are `glue` from the `glue` package and `GET` from the `httr` package. `glue` formats a string allowing for R code to be evaluated before it is concatenated with the adjacent objects. `GET`

is a method for retrieving information requested with a supplied url address. Rctapi works by letting the user enter the query values, formatting them and feeding the formatted url to the GET function. After the response is received, it is parsed into a dataframe through regular expression pattern matching (see the code snippet in Appendix B-1).

A few technical challenges arose in the construction of this data-getting ensemble of functions. Early testing of the API detected a limit of 20 on the number of fields that could be requested simultaneously. Conceptually, this issue could be solved by taking the list of fields (e.g 45 fields), dividing it into as many lists of less or equal to 20 were needed (two lists of 20, list of 5), and executing one request per list. This issue was solved by constructing a matrix where each new list of 20 or less would form a column. Each column of the matrix would then be iterated over once, providing as many as 20 fields to the getting function. Newly returned fields are bound with previously returned fields as the function cycles through the matrix.

While implementing the requests in a loop solved the field request limit, it unearthed another potential issue. This is more commonly encountered in web-scraping and involves the limits placed on the timing and amount of API requests that can be accepted from the same IP address: “a numerical label assigned to each device connected to a computer network” (Cho 2020). These limits are unknown at the time of writing, so a preventative strategy was adopted. A delaying function, `q_delay`, was embedded into all request loops with the purpose of introducing a random time delay between requests.

4.1.2 Usage

Any R user can download the package with `devtools::install_github("Ad1Cruz/rctapi")`. After loading the package into the library with `library(rctapi)`, the user will have access to all its functionalities. The main function is named `get_study_fields`. This function has four parameters with only two default values. Three out of the four parameters were introduced in the previous section. These are the search term, the field list, and the maximum number of studies, which defaults to 500. These appear in the function definition as `search_expr`, `fields`, and `max_studies`. The fourth parameter, `response_content`, is a logical option defaulting to `FALSE` that switches the type of response saved when `get_study_fields` is called. If toggled `TRUE`, instead of returning a dataframe, the user will have access to the full API response. This includes useful information such as the last time the database was updated, the API version in use, the total number of studies in the database and the total number of studies that match the search.

4.1.3 Case-study: Searching and Downloading from ClinicalTrials.gov

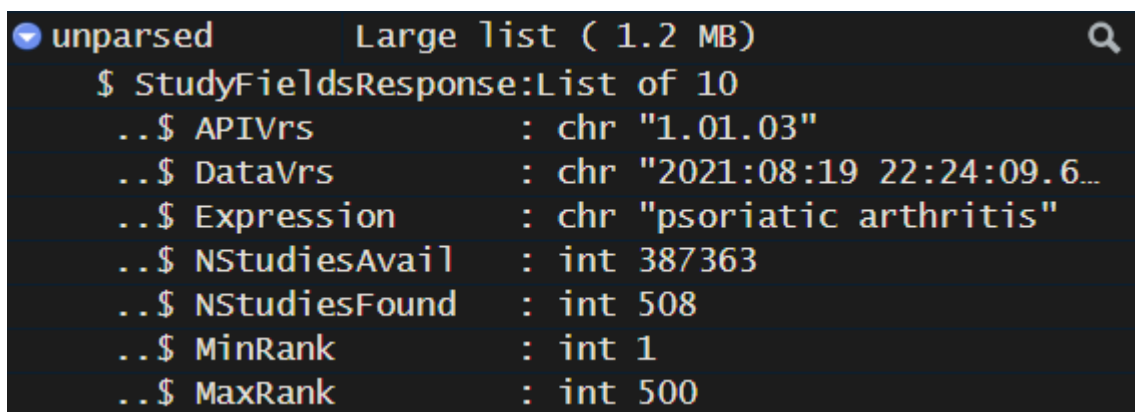
We can begin by establishing the search parameters that we will use to retrieve studies from CTG. In order to make sure that all the relevant studies are captured, we may use “psoriatic arthritis” as our search term. Since this is our first contact with the database it would be of interest to gather some metadata about the search so we will indicate that we want the unparsed response. Which fields we request is of little importance at this moment, since the unparsed response does not return a tidy dataframe. Our resulting function is the following:

```
API_resp <- get_study_fields(search_expr = "psoriatic arthritis",
                             fields = all_fields,
                             response_content = TRUE).
```

Within the output of this function (*Figure 4*), we find a key piece of information. Due to the default settings, we asked for 500 studies, however, 508 studies were found to match the search.

The first few values in the API response are the API version, the last database update, the search expression, the total number of studies in CTG and the number of studies that fit the search expression.

Figure 4: Unparsed API response



```
unparsed      Large list ( 1.2 MB)
$ StudyFieldsResponse: List of 10
..$ APIVrs      : chr "1.01.03"
..$ DataVrs     : chr "2021:08:19 22:24:09.6..."
..$ Expression  : chr "psoriatic arthritis"
..$ NStudiesAvail : int 387363
..$ NStudiesFound : int 508
..$ MinRank     : int 1
..$ MaxRank     : int 500
```

With the additional information we can carry out an improved request. Now, some thought must be put into the fields argument. If we wanted the smallest amount of information to describe the studies, we might use `core_info_fields`, or `extended_info_fields` if we wanted more than just the basic information. We could also use a theme for the fields using lists such as `identification_and_status_fields` or `eligibility_fields`. But, as we are looking to extract as much relevant information as possible within reason, we will be using `registration_fields`. This list includes 62 fields, and they represent the information that the researchers had to upload when registering the study on CTG. Finally, we have the `max_studies` argument. Since the previous search

revealed there to be 508 results matching our search that is the number we will use. Following all this we arrive at:

```
Mease_data <- get_study_fields(search_expr = "psoriatic arthritis",
                               fields = registration_fields,
                               max_studies = 508).
```

The output of the function above is a dataframe with 508 rows and 62 columns. We have too many results to sift through them easily. The solution that will be proposed here is to use the second tool instead, but first we will explore an alternative path. As mentioned previously, the API allows the use of a pseudo-code language to convey search specifications. We can take advantage of this to carry out a more targeted search from the outset. For example, knowing that we are looking for studies that use the American College of Rheumatologists (ACR)¹ or Psoriasis Area and Severity Index (PASI)² outcome measures, we can include these terms in our search as such:

```
psoriatic arthritis AND (AREA[PrimaryOutcomeMeasure](ACR OR PASI) OR
AREA[SecondaryOutcomeMeasure](ACR OR PASI) OR AREA[OtherOutcomeMeasure](ACR
OR PASI)).
```

This expression returns a more targeted list of 195 entries and ensures that the trials include the outcomes of interest.

Although limited, the functionality of this package undergirds a much more flexible second tool that can be used to explore information extracted from CTG without leaving R and without needing extensive R command.

4.2 rctexplorer

4.2.1 Development and Description

At the outset, the expected purposes of rctexplorer were to provide the user with a way to interactively visualise, filter, and summarise the study dataframe.

¹ ACR is a composite measure of improvement involving the number of swollen and tender joints and other disease criteria such as a dual global assessment of disease by patient and physician.

² PASI is the most commonly used assessment for grading psoriasis severity in clinical studies. A physician evaluates the lesioned areas and assigns a score depending on their aspect and body location.

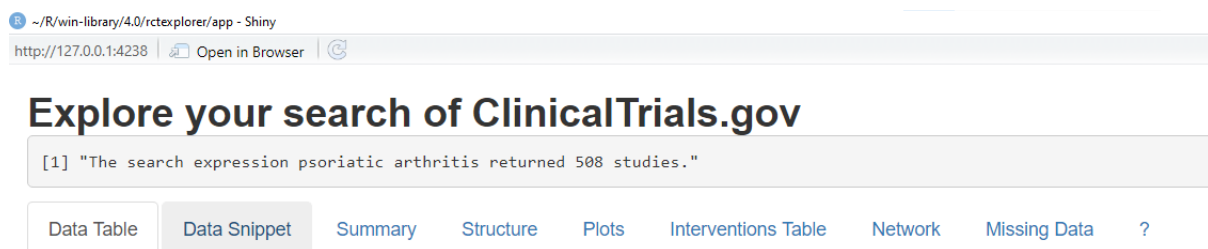
The central element of the `rctexplorer` package is an application, created with the Shiny suite of R packages. The rest of the packages used can be split in two groups, those for data handling and those for visualisation. The tidyverse suite (`dplyr`, `plyr`, `tidyr`) and the string manipulation package `stringr` were used for data handling (Gagolewski 2021). The primary packages used for visual and interface elements were `ggplot2` and `DT` (`datatable`) (Wickham 2016, Yihui 2021). Additionally, packages for specific plot types, `treemap`, `igraph` and `visNetwork` were used (Csardi 2006, Tennekes 2017, Almende 2019).

The application is supported by data processing functions which are executed before the application is launched. These functions act on the columns of the dataframe as they vary in expected ways. Although all the column (therefore field) names are known, they are not all useful, certainly not at the same time. A number of significant fields were selected to be included in the default field list, `for_explorer`, to be used with the application (see full list in **Error! Unknown switch argument.**). The goal of this list is to provide as much useful information to the application as possible without overwhelming the user. The fields selected combine to give an thorough and accurate representation of the trial type, intervention and outcome measures used, trial design, and enrolment and population characteristics. Elements that would help identify and classify the trials were prioritised over results elements.

The main supporting function is `set_app_input`. Given a search term, the only parameter without a default value, this function will request the API and subsequently clean and transform the returned dataframe to better serve the application. A data cleaning function will remove unnecessary rows and convert NA values to empty cells. Two transformations are conditional on the columns present in the dataframe. The variables `AgeRange` and `HasResults` are computed out of existing variables and added to the dataframe. Additionally, to convert as many variables as possible to type factor, each column name is checked against a curated list. All fields that include the word `Type` and most categorical text fields, meaning those whose input at study upload is restricted to categories, such as `OverallStatus` and `Phase`, are factorized. This will play a role later, affecting the way the data is displayed and filtered within the application.

Applications built with R-Shiny generally consist of two functions. One is the user interface function, which sets the interface the user will be interacting with, and the other is the server function, which programs the interface to react to events. `Rctexplorer` uses this configuration, holding the application in three files, `ui.R`, `server.R`, and `global.R` which are run together to launch the application with the function `launch_explorer`. This function's only parameter must be specified. Although the launch function's ideal input is the direct output of `set_app_input`, the application will attempt to launch with any dataframe object as argument. The application can be used within R or on a browser and appears as a window with nine tabs as shown in *Figure 5*. A subtitle shows the user the query performed and the number of studies retrieved.

Figure 5: Rctexplorer banner and tabs



The landing tab is named *Data Table* (*Figure 6*). On the left side a checkbox panel displays the names of the columns (fields) of the dataframe. On the right, the dataframe is displayed as a data table with the `datatable` function from the DT package. Small aesthetic changes were made to the table and a number of options were toggled on. The table is interactive and can react to many different user inputs. Each name in the checkbox panel can be toggled, hiding or showing the corresponding column. Two buttons atop the checkbox panel allow the user to toggle all the columns at once or to display the preset selected fields. The preset fields are `NCTId`, `Acronym`, `StudyType`, `OverallStatus` and `LeadSponsorName`. A text input box above every column allows for by-column filtering, and a global search box in the top right corner allows for global filtering. Both these boxes accept regular expressions. The table also includes buttons for copying or direct download of the filtered data in three different formats, `.csv`, `.pdf` and `.xlsx`.

Key to the utility of the application is the fact that any filtering carried out on the Data Table tab is propagated across all instances of data-processing or visualisation. This does not mean that only visible columns or rows are used, it means that only rows which have not been filtered out are used, and so that only part of the saved dataframe is being used. Columns may still be hidden as they are not being used to filter by, and rows may not fit on the data-table interface. To display all the rows on the same page, the user only needs to increase the number of entries shown with the appropriate drop-down menu.

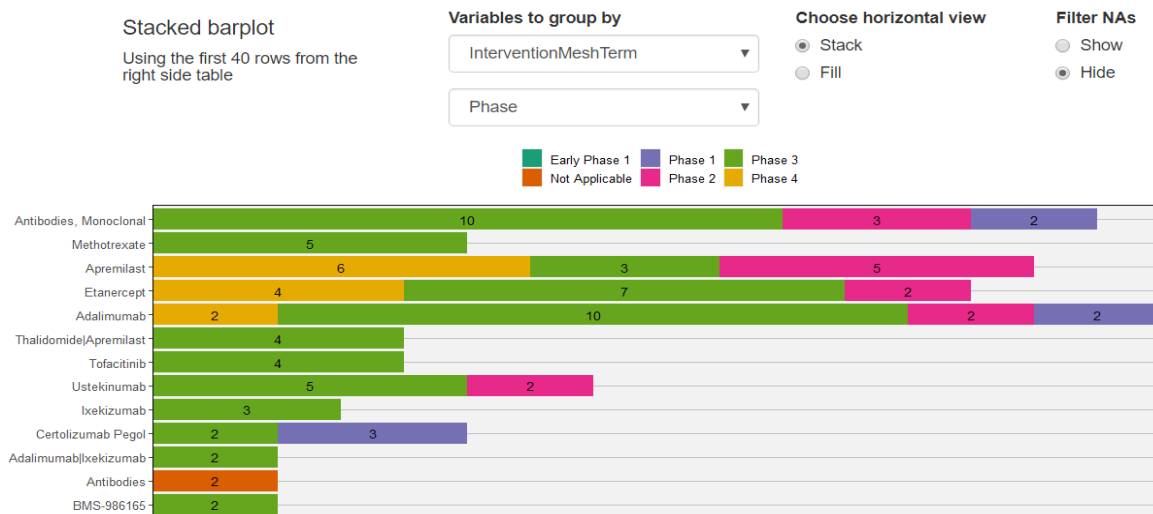
Figure 6: Data Table tab from *rctexplorer*.

	NCTid	Acronym	StudyType	OverallStatus	LeadSponsorName
1	NCT02414633		Observational	Completed	AbbVie
2	NCT03374527		Observational	Completed	Istituto Ortopedico Galeazzi
3	NCT02470481	PARIS	Observational	Completed	Janssen Research & Development, LLC
4	NCT01516957		Interventional	Terminated	Bausch Health Americas, Inc.
5	NCT04180904	DIPSA	Interventional	Recruiting	University of Pennsylvania
6	NCT01111240		Observational	Completed	AbbVie (prior sponsor, Abbott)
7	NCT04325724	EchoPRO	Interventional	Not yet recruiting	Assistance Publique - Hôpitaux de Paris
8	NCT00245960		Interventional	Completed	Wyeth is now a wholly owned subsidiary of Pfizer
9	NCT01147874	PREPARE	Interventional	Completed	Pfizer

There are three summary tabs, named Data Snippet, Summary and Structure. Each displays the output of calling a summary function on the dataframe. The Data Snippet tab shows the first 15 entries. The Summary tab shows the output of calling `summary(data)`, and equally with the Structure tab and `str(data)`. Through these tabs the user can get a glimpse of the data, a summary of categories and their frequency, and an idea of the data types and structure of the dataframe.

The plots tab has three sections, each with different plots, from a univariate treemap and a stacked barplot to a scatter plot charting two categorical and one numerical variable. All the non-numerical variables to be plotted are user selected. The main colour palette employed is Dark2 from the RColorBrewer package (Neuwirth 2014). When required it is programmatically expanded to include as many distinct colours as needed (see code snippet in Appendix D-1).

Figure 7: Screen capture of stacked barplot and control toggles from *rctexplorer* Plots tab.



A treemap is a rectangular plot separated into further rectangular areas, proportionally sized to the number of instances of each category in a variable. This type of univariate plot is a less controversial and misleading univariate visualisation than the pie chart (Midway 2020). The stacked barplot displays two variables and includes two toggles (Figure 7). The user can choose whether to ‘fill’ the bars, and whether to include missing values in the plot. In some cases, the number of instances of a category will not be clearly visible in the plot unless filled to occupy 100% of the available space. This happens often as many fields have a high number of missing values, making the other categories proportionally minute. The toggles can be used to work around these issues. The third plotting area contains a scattered group plot and a helper barplot. The points in the scatterplot represent individual studies and are arranged in the x-axis by the number of participants enrolled. The user can filter by number of participants through a slider rule. The bar plot besides the scatterplot shows the frequency distribution of one of the non-numerical variables.

The next two tabs, Interventions and Network, are geared to aid in the planning of network meta-analyses. The purpose of the Interventions tab is to provide a clutter-free table of the interventions used in each clinical trial (Figure 8). To do this, first an appropriate field was identified, this had to be a field that carried as much information as possible about the study’s interventions. After that, a function was built to parse the values in the column’s cells to obtain a wider table with one column per intervention. This function initially separates the contents of a single cell given a separator character. Then the resulting dataframe is made long, gathering the interventions in one column and their identifiers in two: study NCTId, identifying all treatments in one trial, and label, identifying each intervention within a trial.

Lastly, minor cleaning is carried out to display placebo arms more clearly, and the dataframe is returned to a wide format. The field used in this process is `ArmGroupInterventionName` as it usually carries the type of intervention (Drug, Procedure, Other etc.) followed by a name, dosage, and schedule. If this field is empty in the original dataframe, it will be filled with “Empty” here to block the row’s removal. The intervention columns are named `label1`, `label2`, `label3` ... until each intervention has been allocated a single cell in its row. The table interface is the

Figure 8: Screen capture of the *Interventions* tab *rctexplorer*. The table is being globally filtered for entries containing the highlighted word “placebo”.

	NCTid	Acronym	LeadSponsorName	Phase	label1	label2	label3	label4	label5
4	NCT01516957		Bausch Health Americas, Inc.	Phase 2	Drug: AMG 827 140	Drug: Placebo	Drug: AMG 827 280	Drug: AMG 827 210	
21	NCT04209205	INVIGORATE 2	Novartis Pharmaceuticals	Phase 3	Drug: Secukinumab	Drug: Placebo			
26	NCT00456092		Amgen	Phase 2	Drug: Apremilast	Drug: Apremilast	Drug: Placebo		
29	NCT01892436	FUTURE 1 ext	Novartis Pharmaceuticals	Phase 3	Drug: Secukinumab	Drug: Secukinumab	Drug: Secukinumab	Drug: Placebo	Drug: Secukinumab
31	NCT02141763		UCB Celltech	Phase 1	Drug: UCB4940 160 mg	Drug: UCB4940 240 mg	Other: Placebo	Drug: UCB4940 80 mg	Drug: UCB4940 160 mg
34	NCT00809614		Novartis Pharmaceuticals	Phase 2	Biological: AIN457	Biological: Placebo			
37	NCT01925768		Amgen	Phase 3	Drug: Apremilast 30 mg	Drug: Placebo			
39	NCT02188654		University of Alexandria	Not Applicable	Drug: Metformin	Drug: Placebo			
40	NCT02065713	GO-DACT	Instituto de Medicina Molecular João Lobo Antunes	Phase 3	Drug: Golimumab	Drug: Methotrexate	Drug: Methotrexate	Drug: Placebo	

same as in the landing tab meaning that the columns can be hidden and their contents searched, filtered, and downloaded.

In the Network tab the Interventions table is turned into a network graph. Network graphs are commonly used in the development of network meta-analyses as they provide information at a glance, such as the number of treatments being compared, the number of studies comparing them, and more importantly, they represent how the data to be analysed is structured. After investigating the available R tools for building network plots, `visNetwork` was chosen for its embedded interactive features and relative underlying simplicity.

Generally, a minimum of two data objects are needed to construct a network graph. These objects store the two essential ingredients of any network. One is the nodes, and the other is the links connecting them, here referred to as edges. The edges are gathered in a dataframe with at least two columns, each row representing a pair of interventions, many

rows can refer to the same study if this study employed many different interventions. The nodes are all the unique interventions in the dataset. Before building the edges-object, the original data was processed into a long format dataframe of intervention arms and the intervention names cleaned. This cleaning involved regular expression matching and replacing to eliminate information not directly indicative of the intervention's name (Appendix B-2). While the dosage and schedule information are visible and helpful in the Interventions table, it would massively clutter the network graph by creating many different treatment nodes that in reality refer to the same drug or procedure.

A relatively complex function was written to create the edges. This function cycles through each study via the unique study identifier (NCTId) and creates a table of all the potential intervention pairs. This table is then made conditional to discard single-arm trials, pruned to retain only non-repeat pairs, and bound to the study processed immediately beforehand (Appendix D-2). The output of the edges-making function is fed into a nodes-making function which returns the dataframe of unique interventions. Both the edges and nodes dataframes carry additional information about themselves. Accompanying the edges are study identifiers, acronyms and results-status. And qualifying the nodes is the number of patients that underwent the treatment. These attributes affect the appearance of the network graph, for example by making nodes with more patients appear larger.

The Missing Data tab displays a simple bar plot and table charting the number of empty cells for each retrieved field in the filtered dataset. Missing data is a common issue in massive databases such as CTG. It is useful to visualise this to ensure that the set of data one is examining is not mostly missing. It is also of interest to see if any fields stand out as missing many more values than others.

The final tab is meant as a Help page. In it, the user can find short descriptions of each tab's contents and advice on how to use the application. Specifically, the filtering functionalities of the data tables are briefly explained as well as the plots' interactivity. Finally, three useful links are included which will take the user to the CTG API home page, a regular expression builder page, and the study-field definitions dictionary (Appendix E).

4.2.2 Usage

The package can be downloaded with `devtools::install_github("Ad1Cruz/rctexplorer")`. After loading the package, the first step is to set the input to the application. This is done through the function `set_app_input`, which contains the same main arguments as `rctapi::get_study_fields()`: search expression, fields list and number of studies. Now the fields argument defaults to the list `for_explorer`. The return of

`set_app_input()`, or a dataframe, can be used as the argument for the launcher function, `launch_explorer()`.

To continue with the use-case started above we can reuse the non-specific search expression and save the output of the function to an object. We run:

```
PsA_Mease_21 <- set_app_input(search_expr = "psoriatic arthritis" ,  
  fields = for_explorer,  
  max_studies = 508)
```

This returns a list with the dataframe of studies and the search key. It is true that we are opting to begin very generally and narrow the search with the application, when the narrowing could be done in the previous step, by means of a more targeted search expression. However, large searches of generic terms are part of the intended use of the application and in this case, it serves for demonstrative purposes.

Launching the application with `launch_explorer(data = PsA_Mease_21)` will open a window in R, from here one could also open the application on a browser tab. We can delve now into the retrieved data beginning with the interactive data table in the landing tab

4.2.3 Case study: Systematic filtering of downloaded studies.

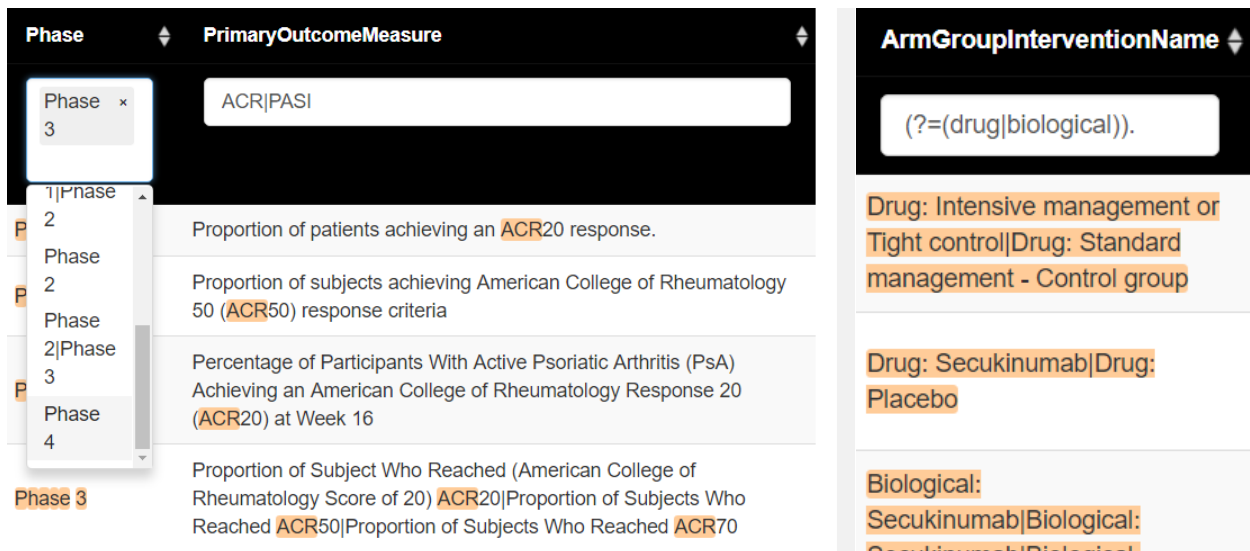
The goal now is to filter down the over 500 studies to approach the list of 26 studies included in Mease et al. (2021) (Table 1). A preliminary look at this list reveals four studies carried out before 2008. Since the database was much smaller at the time and its use was less compulsory, it is likely that these older studies will not be in the database at all.

It is helpful to decide in advance which fields will be used for filtering. Apart from giving structure to our strategy, there is a practical reason for this. A technical limitation of the application is that every time a column is toggled any filtering previously set is wiped. Guided by the study selection criteria in page 42 of the supplementary material in Mease et al., (2021) (Appendix F) and a reading of the published abstract we will be toggling the following fields: Phase, DesignMasking, DesignAllocation, ArmGroupInterventionName, PrimaryOutcomeMeasure and PrimaryOutcomeTimeFrame, We will also be filtering by the preset fields StudyType and OverallStatus. More fields can be toggled on for additional context.

The abstract mentions that all selected studies were Phase III, and that the outcomes of these studies were ACR and PASI. Filtering for Phase 3 studies is straightforward. The column for Phase is categorical, so clicking on the box on top opens a drop-down of the

categories (Figure 9). Next, we filter for the outcome measures used, this can be done through the column filter. The filtering expression is “ACR|PASI” with the bar signifying OR.

Figure 9: Filtering variables through categories, and logical and regular expressions



We could attempt to recreate the search by specific drug that Mease et al., (2021) carry out, but this approach would be too time consuming as the search terms employed in the study spans 17 pages (supplementary material p.25-41). Instead, we can use a regular expression termed positive look-ahead (Figure 9) in the ArmGroupInterventionName column to filter for trials that used a drug or biological intervention. A strength of using strict regular expressions instead of the simple logical expression used for the PrimaryOutcomeMeasure column is that empty fields are not excluded. This is important because ArmGroupInterventionName being empty does not suggest the study should be discarded. Adding filters for Phase, Outcome and Intervention type removes 429 studies leaving 79 in the data table.

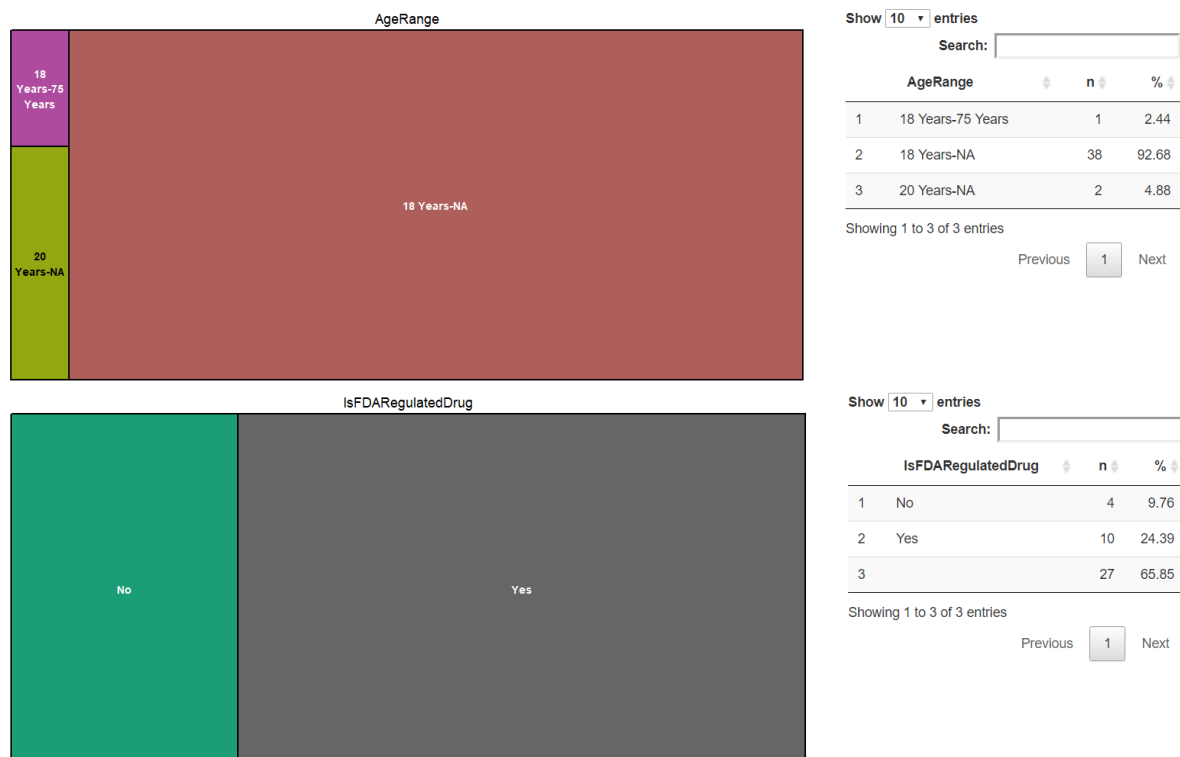
This is certainly an approach to the Mease et al., (2021) dataset but we can filter further as there are a few more exclusion criteria we can add. Non-randomized studies can be filtered-out through the DesignAllocation field and open-label studies can be excluded through the DesignAllocation field. These filters bring the number of entries to 64. Although not present in the supplementary material it is understood that the trials evaluated have concluded successfully so we can filter by field OverallStatus: Complete. This results in a final dataset of 41 trials.

Some eligibility criteria are much more difficult to implement in the application. In this case, the exclusion of studies with a time frame of less than 12 weeks is likely possible through regular expression matching but the free-text, non-standardized nature of the PrimaryOutcomeTimeFrame field, with values as varied as “Week 16”, “Month 3”, “Day 169” and “Week 12|Throughout the Study”, makes it a great challenge.

4.2.4 Case study: Visualization and Network tools

The Plots tab can be visited at any time during the filtering without losing any filters added. Besides the first two types of plot is a dynamic data table displaying the variable data being plotted.

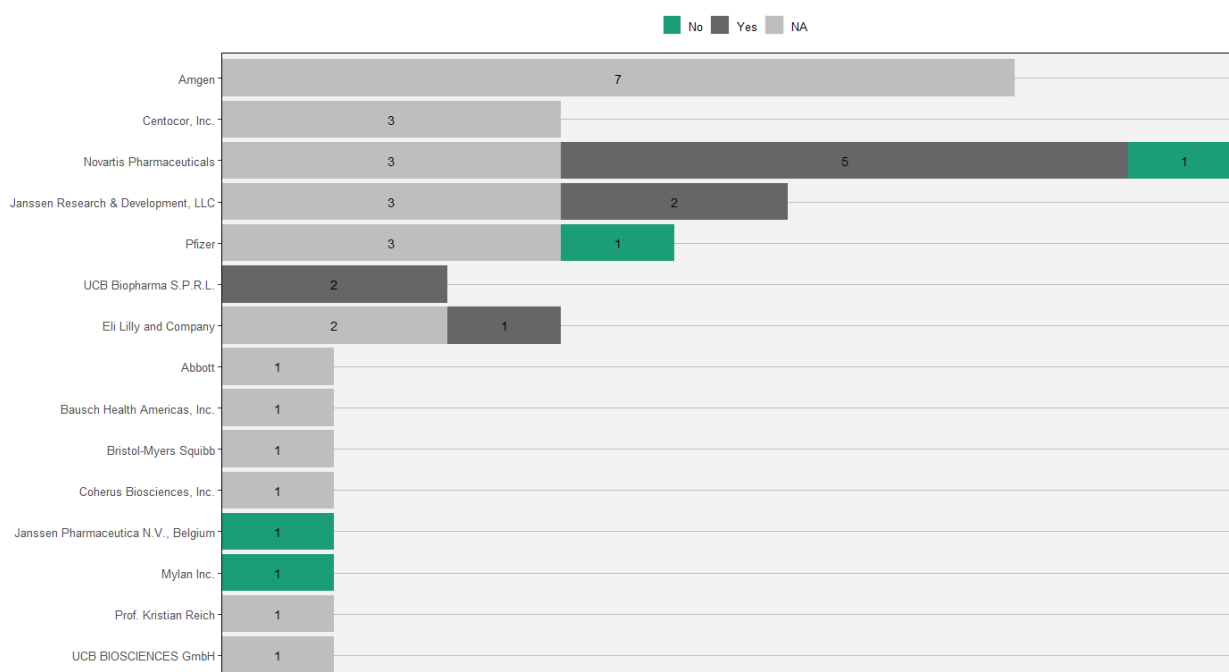
Figure 10: Treemap visualization of AgeRange and IsFDARegulatedDrug fields with accompanying table.



The treemap plot is pre-set to show the distribution of OverallStatus. Since we have already implemented a few filters including one for OverallStatus, a univariate view will not bring much new information. We can still use it to ask questions of other fields. For example, what are the age ranges of participants in the filtered studies? and, how many of the trials involve FDA-regulated drugs?

Figure 10 answers these questions, both through the treemap and the adjacent table. The minimum age is 18 years and most studies do not specify a maximum. According to the FDA regulation treemap most studies use an FDA regulated drug. However, the table conveys the reality of the data as most studies (27/41) are not disclosing this information.

Figure 11: Stacked barplot visualization of LeadSponsor and IsFDARegulatedDrug fields

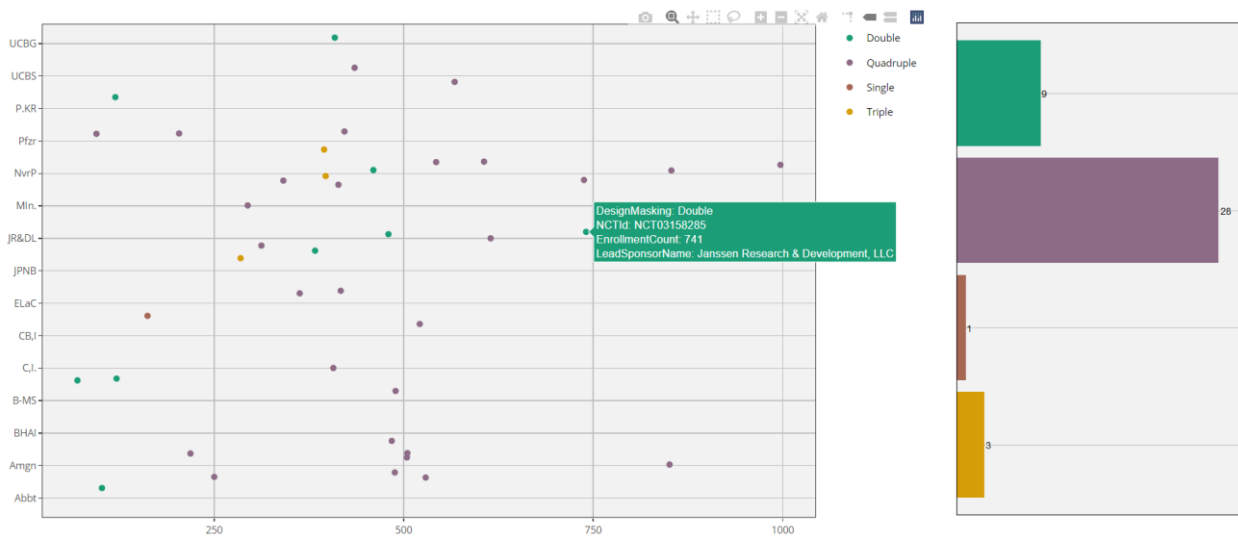


Any pair of variables can also be visualised as a stacked barplot or scattered across their participant numbers (field EnrollmentCount). We delve slightly deeper into the FDA regulated drug question by plotting this field against the trial sponsor in Figure 11.

The filtering carried out has selected for a particular type of trial. However, we can take this sample to represent the larger landscape of drug development. Novartis Pharmaceuticals, without counting its subsidiaries, seems quite invested in the psoriatic arthritis space with 9 trials that fit our criteria. Janssen equals this, if we include the trials carried out by their now subsidiary company Centocor. Interestingly, none of the selected Amgen trials report if the drug under investigation has received FDA approval. In general, it seems that reporting the FDA status of a drug is not enforced at trial registration and is therefore not often done.

The scatter plot is most useful to appraise the number of participants enrolled in the trials selected. In *Figure 12* we combine this with the lead sponsor and degree of masking used. From this we learn that the three largest trials count with 750 to 1000 participants and that all use quadruple masking. Hovering on the fourth largest trial shows expanded information and the helper barplot to besides the legend provides the aggregated distribution of the masking type variable.

Figure 12: Scatter plot of LeadSponsorName and DesignMasking fields against number participants enrolled.



After visualizing the trials we can move on to the interventions table. This table is most helpful when the name of the intervention is accompanied by the dosage and schedule. If the `ArmGroupInterventionName` field does not carry any information (displays “Empty”) then this table allows us to examine related fields such as `InterventionName` which might provide the missing information. Studies with an empty `ArmGroupInterventionName` cannot be processed to create the network graph.

The Network tab turns the filtered studies into an interactive network graph. One node corresponding to one treatment and one edge corresponding to one or more trials. The size of each node is proportional to the number of participants that received the treatment. The width of each edge is indicative of the number of different trials which include the nodes they link. This number is labelled on top of each edge, as is the name of each treatment labelled besides the nodes.

placebo arm and two appear outside the network. Of note, one of these isolated studies is there due to the unconventional naming of the drug adalimumab.

4.2.5 Case study: Assessment of systematic filtering and trial identification

There are many differences between the process carried out by Mease et al., (2021) in their systematic literature review (SLR) and what was done in the systematic filtering section in this report. To begin with, this report cannot say to have carried out any sort of literature review, even an informal one, as the raw data were not peer-reviewed studies but clinical trials. In an SLR there is a more automatic phase where a large volume of information is sifted and a manual phase where papers are examined closely. In this case-study, the review process consisted of adding filtering parameters and applying them to the whole dataset as one. Given these considerations, how close did the filtering process get to the list of studies in the published NMA?

To begin answering this question, a list of NCTIDs was compiled by searching the internet and CTG for the trials in Mease et al., (2021). This initial search revealed that two of the oldest trials were missing from the database and so did not have an NCTId (Mease, Kivitz et al. 2004, Genovese, Mease et al. 2007). This was expected, prior to 2008 considerably less studies and even less study results were being published on CTG (Zarin, Tse et al. 2011). Knowing that only a maximum of 24 trials could possibly be found in the retrieved dataset, the vector of NCTIDs was entered into the filter box for the NCTId column. The table transformed to show the 22 entries with matching NCTIDs.

Further investigation was carried out to understand why the trials SPIRIT H2H - NCT03151551 and ADEPT - NCT00195689 had escaped the filtering process (Abbott 2005-2007, Company 2017-2019). In both cases the issue was one of incongruent data. The ADEPT trial is marked as Non-Randomized in its DesignAllocation field and as None(OpenLabel) in its DesignMasking field. These labels are at odds with the eligibility criteria used by Mease et al., (2021) which state that non-randomized and open label studies would be excluded, but they agree with the published paper which confirms the study to be an open-label extension (Mease, Ory et al. 2009). The data incongruence resulting in the exclusion of SPIRIT-H2H trial is similar. Mease et al., (2021) state in the selection criteria that Phase I would mean exclusion while Phases II, II-III, and III would mean inclusion. Then, in the abstract and text body they write that all studies included were of Phase III. This is why during the case-study filtering process all but Phase III trials were excluded. The original published paper identifies SPIRIT-

H2H as a “phase IIIb/IV” trial (Mease, Smolen et al. 2020) and this is further shifted in the CTG record, which only shows Phase IV.

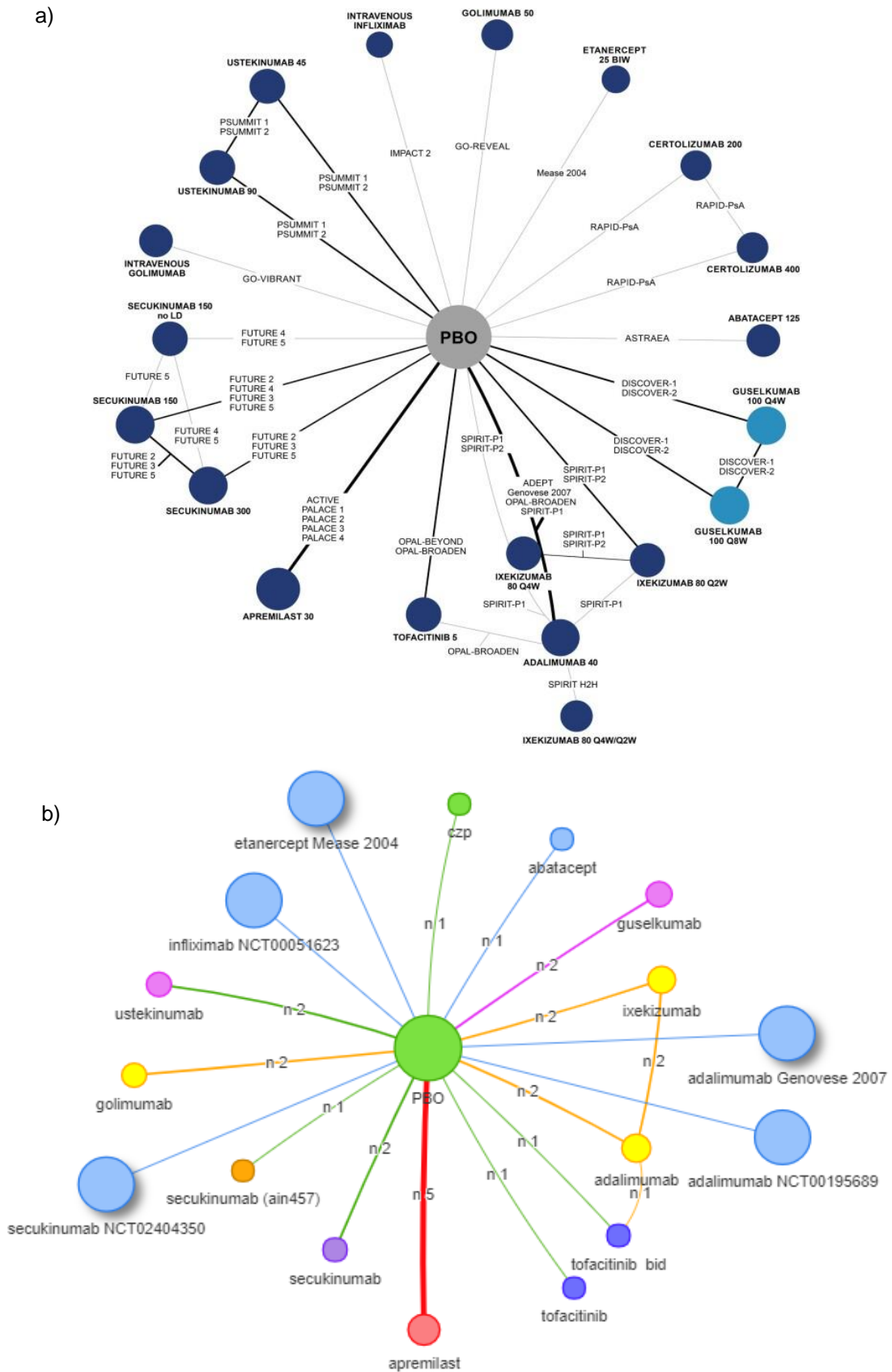
Although these disagreements could raise questions about the appropriateness of including the studies in the meta-analysis, it is likely that the author, being the same for these studies and the NMA, had additional information that allowed him to make the right decision. In any case, the incongruences explain why the trials were removed during the filtering process on `rctexplorer`. The issue with the Phase field was dependent on the chosen filters, we could have chosen to begin by casting a wider net and including all trials but Phase I and II for example. The differing DesignAllocation and DesignMasking fields on the other hand could not have been identified by better filtering steps, not without using alternative resources to `rctexplorer` and therefore searching outside ClinicalTrials.gov.

To recapitulate, we narrowed the 508 studies returned from the database to 41 records through filtering steps. Twenty-two of the 26 studies included in Mease et al., (2021) were found among these 41 records. Two trials could not be in the filtered dataset because they were not in the database, and two trials were not in the dataset due to a combination of data incongruence and suboptimal filtering steps.

Using `rctexplorer` we closely approximated the full list of trial-originating studies that were included in Mease et al., (2021). If not for misclassification issues, all the trials present in the database would have been in the filtered group. However, in order to further reduce this list, a granular approach must be taken, perhaps beginning by examining the free-texts in the eligibility criteria and primary outcome timeframe fields. All in all, this tool was successful at quickly and easily identifying studies with shared characteristics. The main advantage and difference to the SLR carried out by Mease et al., (2021) is the accessibility and rapidity of the process. There are drawbacks to this tool. The accessibility derives from being connected to the public and free CTG, which can provide confusing information, causing in turn the dismissal of otherwise valid studies.

There were a few more issues with the study-data, all related to missing or faulty fields. Referring to studies' acronyms is a practical way of uniquely identifying them. Although helpful, it would not have been possible to refer to studies by their acronym in this case-study. Nine out of 24 studies present in Mease et al., (2021) and CTG do not include their acronym information. Two trials, NCT00195689 and NCT00051623, appear empty in the ArmGroupInterventionName field, which is used to construct the interventions table and network graph (van der Heijde, Kavanaugh et al. 2007, Mease, Ory et al. 2009). If a study is missing this field, it will still appear on the Interventions tab, but in order to examine the interventions used, the user would have to toggle additional columns.

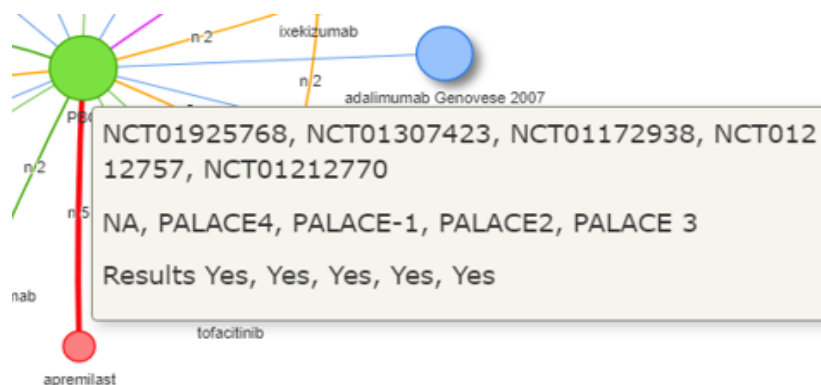
Figure 14: (a) Treatment network for outcome measure ACR20 from Mease et al. (2021). (b) Network recreated in *rctexplorer*.



In the trial record NCT02404350, (Mease, van der Heijde et al. 2018), the interventions field display the exact same intervention a number of times and do not indicate that a placebo intervention took place. Examining the study on the CTG website, we see that different dosages of the treatment were used and that the trial did include a placebo arm. Since the `rctexplorer` application will detect only one intervention, this trial would not be included in the network graph.

Data issues notwithstanding, it was possible to recreate the evidence network featured in Mease et al., (2021) with `rctexplorer` (Figure 14). The NCTIDs were used to select the studies and the network-building functions carried out almost all the rest of the work. Two nodes were added manually due to their `ArmGroupInterventionName` field being empty. One node was added due to its `origin` field being incorrectly formatted. These newly added nodes and edges are visibly different as they do not carry additional information within them and they do not offer additional information at mouse hover (Figure 15).

Figure 15: Mouseover popup of apremilast to placebo edge showing NCTIDs, acronyms and results status.



5. Discussion

This project was motivated by the lack of a robust open-source resource connecting R directly to the public online database ClinicalTrials.gov. The tool created was intended to allow the input of accurate commands for the download of clinical trial data, as can be done within the CTG website, but also to greatly expand what one could do with the downloaded data. Specifically, to provide interactive filtering and visualisation functions. The ability to probe CTG in this flexible yet reliable way is the value that the project adds to the UCB partners and any member of the research community sufficiently versed in R to operate it.

The R packages created, `rctapi` and `rctexplorer`, connect to CTG through the website's official API. The official nature of the API means the connection is robust and will change little and only towards improved performance. `rctapi` serves to return study data directly from the website in an R-compatible format. Through this package's main function, the user can specify complex searches of the database and the specific fields within each study to retrieve. At its limit, it can retrieve as many as 322 information fields from 1000 records. `rctexplorer` builds on the previous package and provides the user with an R-Shiny application through which to explore the downloaded data.

Embedded in the application are powerful and flexible filtering functionalities. The user can parse the data at different levels of complexity from simple category selection to regular-expression filtering. These filters propagate across the application, meaning that the three types of interactive visualisation of user selected variables, the table of flattened interventions and the network graph reflect the filtering the user has implemented.

The interventions table provides a straightforward way to examine the interventions, dosage and schedule used in each arm of the trial in question. The Network graph automatically builds an approximation of the treatment network formed by the filtered studies. Due to the package and methods used to build the graph, it is packed with interactive features, allowing in the extreme to build an entire network graph from scratch by dragging and dropping new nodes and edges. Given all the above functionalities we can judge the tools successful in meeting the objectives set at the beginning of the project.

5.1 Limitations and challenges

The main potential functionality that could not be implemented was the processing of clinical trials result data. The previously mentioned proprietary tools (SHERLOCK™), are allegedly capable of extracting actual results from the database in such a way that they can be used directly to compute a network meta-analysis. The development of this functionality was discussed and attempted.

In theory, if the numerical results of the analysis are present in the dataset, it should be possible to gather, appropriately label and save them for later use. In practice, it would be considerably more complex. Once the dataset had been downloaded, the fields that held more than one value – such as the field carrying the results of the analyses – were collapsed into one cell. As this happened across every multi-value field it became more difficult to recuperate the context of each value. To do this, the program would have to read the trial record and figure out, among other things, which interventions were being used and how the trial arms were organised, which analyses were carried out, which comparison were made, and which outcome measure each comparison result referred to. The complexity of this task is increased by the general data quality issues present in the database.

Overall, CTG is an exceptional resource, made so primarily by the volume of information it hosts. Its data, not unlike other large online databases, varies in quality, and there are warnings for those wanting to use it. The issues encountered during this project relate mainly to insufficient standardization and field incompleteness. Many fields in each record are standardized, but two important fields, those denoting condition and intervention, are not. There must be an option to input custom conditions and treatments, but for the vast majority of trials this is not necessary. The standardization to terms found in the Medical Subject Headings dictionary would enhance the quality of the database (Miron, Gonçalves et al. 2020), and would have greatly benefited this project. Similarly, the fields indicating eligibility requirements and outcome measures are currently free text with flexible recommendations. Further standardization of these fields could help more easily identify if a trial can be included or not in a meta-analysis, as well as facilitate the search for similar populations across different trials. Although incomplete fields are a common occurrence, there is little CTG can do short of requiring field completion prior to upload. Ultimately, it is the responsibility of researchers to upload accurate information in a timely manner and CTG's role to incentivise good behaviour.

Two minor limitations were encountered, and persisted, during the construction of the user interface within the shiny application. One is a glitch that makes the stacked barplot occasionally overlap with the selection toggles of the scatter plot underneath. This issue appears seemingly at random. Attempts at modifying the html code that builds the interface were unsuccessful, so this issue persists. The network graph also exhibits an interface issue. The visNetwork viewer window would benefit from being much larger. Unfortunately, re-sizing elements that are not directly accounted for by the main R-Shiny packages is not straightforward and so the attempts at making the network graph window larger were unsuccessful.

5.2 Future directions

The open-source nature of the project means that it will be open to modifications for the foreseeable future. The next step might be to rigorously test the tools with the ultimate objective of upholding the high standards that CRAN requires to host a package. If uploaded to this archive, the packages would also be more accessible.

A potential improvement on the current `rctexplorer` program could consist of redesigning the functions that build the Interventions tab. The goal would be to ameliorate the issue of missing and faulty intervention fields which reduces the utility of both the Interventions table and the Network graph. For example, a function could be directed to look inside several fields all carrying intervention related information, and to stop only when the most informative term is identified.

Another avenue to explore is the creation of a results-processing function. This promises to be very challenging but would massively expand the functionality of ClinicalTrials.gov, so much so that it might benefit the database itself to facilitate this sort of operation by modifying the way analysis results are stored.

6. Conclusion

Two R packages were developed to provide a direct and uncomplicated way of extracting and exploring data from the database ClinicalTrials.gov. This report has described their conception and development. The objectives of creating open-source, multi-functional tools were accomplished as the packages are freely available to be downloaded from the author's Github repositories and are packed with practical interactive features.

One of the intended applications of the tools is the selection of studies for meta-analyses. This was put to the test by simulating the systematic literature review carried out by Mease et al., (2021) with the intention of concluding the process with a list of records that was as similar as the original as possible. The package `rctexplorer` facilitated this by providing powerful filtering functions and helpful visualisations. Additionally, the application inside includes utilities to explore each study's interventions through a data table, and to visualise how the studies relate to each other through a network graph. Most studies in the network meta-analyses were present in the final filtered dataset, attesting to the vastness of CTG and the accuracy and practicality of the tools created. The studies that did not appear in the final dataset were either missing entirely from the database, highlighting the importance of searching multiple sources when planning a meta-analysis; or missing due to incongruent data that led to their exclusion during the filtering steps, hinting at the deep underlying difficulty of incentivising accurate reporting.

References

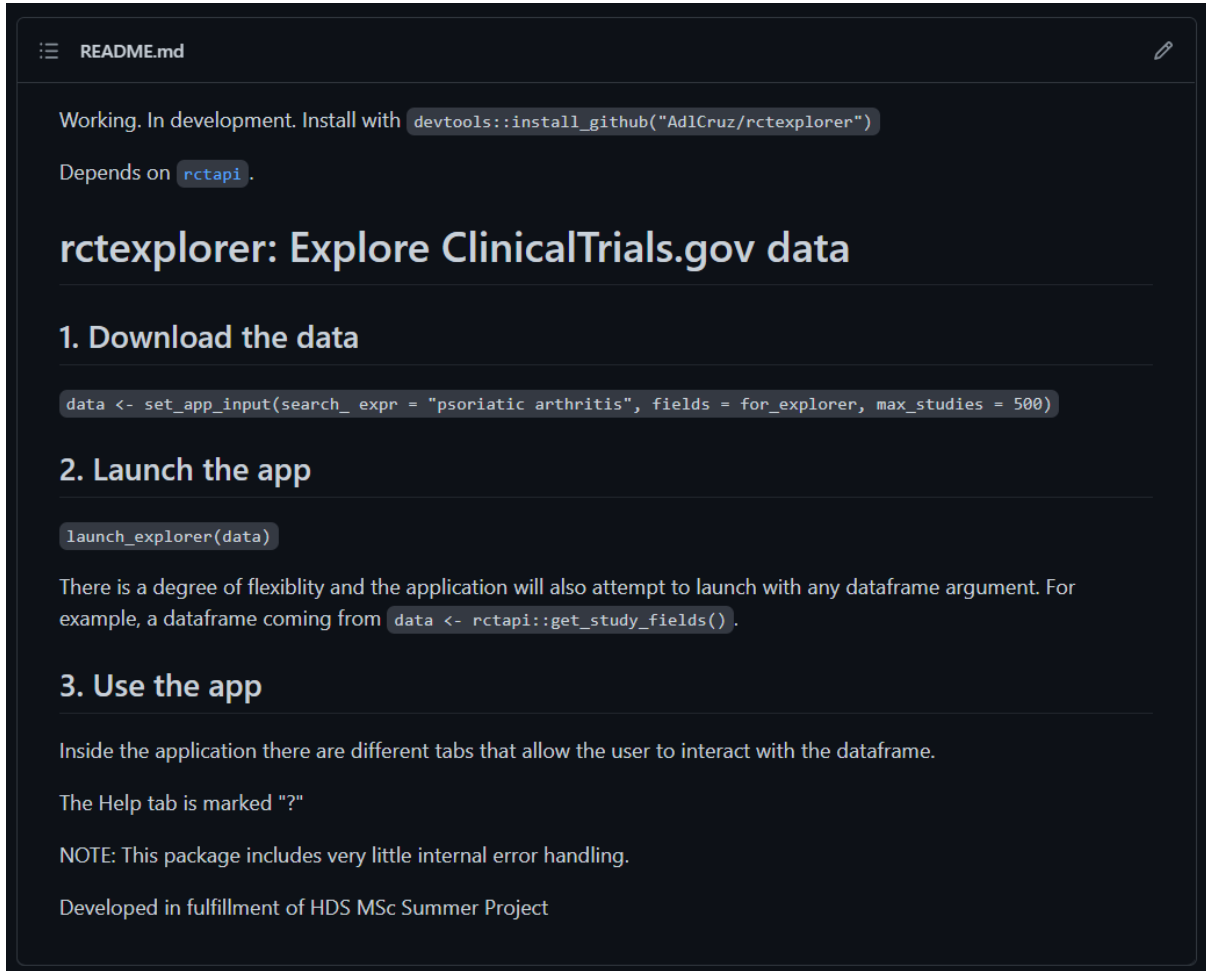
- Abbott. (2005-2007). "Safety and Efficacy of Adalimumab in Patients With Moderate to Severely Active Psoriatic Arthritis." from <https://clinicaltrials.gov/ct2/show/NCT00195689>.
- Administration, C. f. D. E. a. R. C. f. B. E. a. R. F. a. D. (2002). Guidance for Industry Information Program on Clinical Trials for Serious or Life Threatening Diseases and Conditions.
- Almende, B. V. T., Benoit; Robert, Titouan (2019). "visNetwork: Network Visualization using 'vis.js' Library."
- Cepeda, M. S., V. Lobanov and J. A. Berlin (2013). "From ClinicalTrials.gov trial registry to an analysis-ready database of clinical trial results." *Clinical Trials* **10**(2): 347-348.
- Cho, W. (2020). "IP Address." from https://support.ucsd.edu/its?id=kb_article_view&sys_kb_id=c1a708bc1b831898e519c9176e4bcb5b.
- clinicaltrials.gov. (2014). "Terms and Conditions." from <https://clinicaltrials.gov/ct2/about-site/terms-conditions>.
- Company, E. L. a. (2017-2019). "A Study of Ixekizumab (LY2439821) Versus Adalimumab in Participants With Psoriatic Arthritis (SPIRIT-H2H)." from <https://clinicaltrials.gov/ct2/show/NCT03151551>.
- Congress, t. (1988). S.2889 - Health Omnibus Extension of 1988. Section Title: AIDS Amendments of 1988.
- Congress, t. (1997). "Food and Drug Modernization Act of 1997."
- Congress, t. (2007). Food and Drugs Administration Amendments Act of 2007.
- Csardi, G. N., Tamas (2006). "The igraph software package for complex network research." *InterJournal Complex Systems*.
- Gagolewski, M. (2021). "stringi: Character String Processing Facilities."
- Genovese, M. C., P. J. Mease, G. T. Thomson, A. J. Kivitz, R. J. Perdok, M. A. Weinberg, J. Medich and E. H. Sasso (2007). "Safety and efficacy of adalimumab in treatment of patients with psoriatic arthritis who had failed disease modifying antirheumatic drug therapy." *J Rheumatol* **34**(5): 1040-1050.
- Karcher, H., W. Wiecek, M. Nikodem, E. Voss, A. Sena and S. Cepeda (2016). "PRM111 - A NEW TOOL TO AUTOMATE NETWORK META-ANALYSES OF STUDIES EXTRACTED FROM CLINICALTRIALS.GOV." *Value in Health* **19**(3): A91.
- Mayer, C. S. and V. Huser (2020). "Computerized monitoring of COVID-19 trials, studies and registries in ClinicalTrials.gov registry." *PeerJ* **8**: e10261-e10261.
- McCray, A. T. and N. C. Ide (2000). "Design and implementation of a national clinical trials registry." *J Am Med Inform Assoc* **7**(3): 313-323.
- Mease, P., D. van der Heijde, R. Landewé, S. Mpofo, P. Rahman, H. Tahir, A. Singhal, E. Boettcher, S. Navarra, K. Meiser, A. Readie, L. Pricop and K. Abrams (2018). "Secukinumab improves active psoriatic arthritis symptoms and inhibits radiographic progression: primary results from the randomised, double-blind, phase III FUTURE 5 study." *Annals of the Rheumatic Diseases* **77**(6): 890.
- Mease, P. J., A. J. Kivitz, F. X. Burch, E. L. Siegel, S. B. Cohen, P. Ory, D. Salonen, J. Rubenstein, J. T. Sharp and W. Tsuji (2004). "Etanercept treatment of psoriatic arthritis: safety, efficacy, and effect on disease progression." *Arthritis Rheum* **50**(7): 2264-2272.
- Mease, P. J., I. B. McInnes, L.-S. Tam, K. Eaton, S. Peterson, A. Schubert, S. D. Chakravarty, A. Parackal, C. S. Karyekar, S. Nair, W.-H. Boehncke and C. Ritchlin (2021). "Comparative effectiveness of guselkumab in psoriatic arthritis: results from systematic literature review and network meta-analysis." *Rheumatology (Oxford, England)* **60**(5): 2109-2121.

- Mease, P. J., P. Ory, J. T. Sharp, C. T. Ritchlin, F. Van den Bosch, F. Wellborne, C. Birbara, G. T. Thomson, R. J. Perdok, J. Medich, R. L. Wong and D. D. Gladman (2009). "Adalimumab for long-term treatment of psoriatic arthritis: 2-year data from the Adalimumab Effectiveness in Psoriatic Arthritis Trial (ADEPT)." Ann Rheum Dis **68**(5): 702-709.
- Mease, P. J., J. S. Smolen, F. Behrens, P. Nash, S. Liu Leage, L. Li, H. Tahir, M. Gooderham, E. Krishnan, H. Liu-Seifert, P. Emery, S. G. Pillai and P. S. Helliwell (2020). "A head-to-head comparison of the efficacy and safety of ixekizumab and adalimumab in biological-naïve patients with active psoriatic arthritis: 24-week results of a randomised, open-label, blinded-assessor trial." Annals of the Rheumatic Diseases **79**(1): 123.
- Medicine, N. L. o. (2000). National Institutes of Health Launches "ClinicalTrials.gov".
- Midway, S. R. (2020). "Principles of Effective Data Visualization." Patterns **1**(9): 100141.
- Miron, L., R. S. Gonçalves and M. A. Musen (2020). "Obstacles to the reuse of study metadata in ClinicalTrials.gov." Scientific Data **7**(1): 443.
- Neuwirth, E. (2014). "RcolorBrewer: ColorBrewer Palettes."
- Tan, X.-L., D. M. Kern and M. S. Cepeda (2021). "Identifying Anticipated Events of Future Clinical Trials by Leveraging Data from the Placebo Arms of Completed Trials." Therapeutic innovation & regulatory science **55**(2): 454-461.
- Team, R. C. (2013). "R: A Language and Environment for Statistical Computing." R Foundation for Statistical Computing.
- Tennekes, M. (2017). "treemap: Treemap visualization."
- van der Heijde, D., A. Kavanaugh, D. D. Gladman, C. Antoni, G. G. Krueger, C. Guzzo, B. Zhou, L. T. Dooley, K. de Vlam, P. Geusens, C. Birbara, D. Halter and A. Beutler (2007). "Infliximab inhibits progression of radiographic damage in patients with active psoriatic arthritis through one year of treatment: Results from the induction and maintenance psoriatic arthritis clinical trial 2." Arthritis Rheum **56**(8): 2698-2707.
- Wickham et al. (2019). "Welcome to the tidyverse." Journal of Open Source Software **4**(43): 1686.
- Wickham, H. (2016). ggplot2: Elegant Graphics for Data Analysis. New York, Springer-Verlag.
- Wickham, H. D., Peter; Csardi, Gabor; Eugster, Manuel; RStudio (2020). "roxygen2: In-Line Documentation for R."
- Yihui, X. J., Cheng; Xianying, Tan (2021). "DT: A Wrapper of the JavaScript Library 'DataTables'."
- Zarin, D. A., T. Tse, R. J. Williams, R. M. Califf and N. C. Ide (2011). "The ClinicalTrials.gov Results Database — Update and Key Issues." New England Journal of Medicine **364**(9): 852-860.

Appendices

Appendix A

ReadMe example file.



The image shows a dark-themed screenshot of a README.md file. At the top, it says 'Working. In development. Install with' followed by a code block: `devtools::install_github("AdlCruz/rctexplorer")`. Below that, it says 'Depends on' followed by `rctapi`. The main title is 'rctexplorer: Explore ClinicalTrials.gov data'. There are three numbered sections: 1. Download the data, with a code block `data <- set_app_input(search_expr = "psoriatic arthritis", fields = for_explorer, max_studies = 500)`; 2. Launch the app, with a code block `launch_explorer(data)` and a paragraph explaining flexibility and an example code block `data <- rctapi::get_study_fields()`; 3. Use the app, with a paragraph about tabs and a note about error handling. The footer says 'Developed in fulfillment of HDS MSc Summer Project'.

Appendix B

Instances of regular expression pattern matching:

2- Parsing API response, "con", into a dataframe of "records"

```
split_hdrs <- stringi::stri_split(con, regex = "\\n\\s*\\n")
split_lines <- stringi::stri_split_regex(split_hdrs[[1]], "\\n")
records <- read.table(text = split_lines[[2]], sep = ",", header = TRUE)
```

2- Parsing intervention names to clean them for Network graph

```
long_arms$value <- tolower(long_arms$value)
# remove chars before colon
```



```

long_arms$value <- gsub(".", "", long_arms$value)
# remove mg dosage
long_arms$value <- gsub("[0-9]+.mg", "", long_arms$value)
# remove schedule
long_arms$value <- gsub("q2w|q4w|q8w", "", long_arms$value)
# remove empty parenthesis
long_arms$value <- gsub("\\(\\)", "", long_arms$value)
# subst for or to placebo to just placebo
long_arms$value
ifelse(grepl("placebo.(for.|to.)", long_arms$value), "PBO", long_arms$value) <-
  long_arms$value
ifelse(grepl("placebo", long_arms$value), "PBO", long_arms$value) <-
  long_arms$value <- trimws(long_arms$value) # trim whitespace

```

Appendix C

For_explorer list as is saved within the package rctapi.

```
for_explorer <- c("NCTId", "OfficialTitle", "BriefTitle", "Acronym",  
"StudyType", "overallStatus", "StartDate", "CompletionDate",  
"LeadSponsorName", "IsFDAREgulatedDrug", "IsFDAREgulatedDevice",  
"IsUnapprovedDevice", "OversightHasDMC", "Condition", "Keyword",  
"WhyStopped", "DesignPrimaryPurpose", "Phase", "DesignInterventionModel",  
"DesignMasking", "DesignAllocation", "EnrollmentCount", "ArmGroupLabel",  
"ArmGroupType", "ArmGroupInterventionName", "InterventionType",  
"InterventionName", "InterventionMeshId", "InterventionMeshTerm",  
"PrimaryOutcomeMeasure", "PrimaryOutcomeTimeFrame",  
"SecondaryOutcomeMeasure", "SecondaryOutcomeTimeFrame", "Gender",  
"MinimumAge", "MaximumAge", "HealthyVolunteers", "EligibilityCriteria",  
"ResultsFirstPostDate", "OutcomeAnalysisParamValue",  
"OutcomeAnalysisParamType")
```

Appendix D

Miscellaneous code snippets

- 1- Programmatically expanding treemap plot colour palette (highlighted) according to number of levels present in the input variable `treemap_var`.

```
treemap(tree_dat,
        index=input$treemap_var,
        vsize="n",
        type="index",
        title = input$treemap_var,
        title.legend = NA,
        algorithm = "pivotSize",
        sortID = "size",
        palette = colorRampPalette(brewer.pal(n = 8, name =
"Dark2"))(length(levels(df[,input$treemap_var]))),
        draw = TRUE)
```

- 2- Defining the function that turns clinical trial arm data into a network-ready dataframe of edges.

```
to_edges <- function(long_arms) {
# getting unique ids, creating empty dataframe
  r1 <- unique(long_arms$NCTid)
  tib_df <- data.frame()
# iterating ids and extracting unique intervention names
  for (i in 1:length(r1)) {
    crnt_stdy <- long_arms[long_arms$NCTid == r1[i],]
    unique_trts <- dplyr::distinct(crnt_stdy, value, .keep_all =
TRUE)
# excluding single treatment trials
    if (length(unique_trts$NCTid) < 2 ) {
      next
    }
# constructing matrix of treatment pairs
    else {
      combi <- t(combn(unique_trts$value,2))
      sorted_combi <- t(apply(combi,1,sort))
```

```
    othr_vars
unique_trts[1:length(sorted_combi[1]),1:(ncol(unique_trts)-2)] <-
    othr_vars$n_trts <- length(unique_trts$NCTId)
    cmplt_combi <- cbind(sorted_combi,othr_vars)
# binding to previously processed trials
    tib_df <- rbind(tib_df,cmplt_combi)
  }
}
names(tib_df)[1] <- "from"
names(tib_df)[2] <- "to"
return(tib_df)
}
```

Appendix E

Screenshot of Help tab from rctexplorer

Tabs Overview:

Data Table : The dataframe entered into the application through the launcher function. Boxes at the top of each column show the categories held when possible and allow filtering.

Data summaries : 'Data Snippet' displays the first 15 observations in the dataframe. 'Summary' and 'Structure' display the output from calling `summary()` and `str()` functions on the dataframe

Plots : Show frequency of chosen grouped variables and studies by number of participants enrolled.

Interventions Table : Table showing one study per row with a column for each of its intervention arms (split from var. 'ArmGroupInterventionName')

Network : Network plot of interventions for filtered studies

Missing Data : Shows number of missing values for each variable present in the filtered dataframe

Filtering

Filtering implemented on the main data table will be carried on to all other instances of the data used in the application. At startup the Data Table tab will display only a few columns. More columns can be toggled on via the checkbox. The boxes atop of each column and the global search bar both understand regular expressions. Warning: all filtering resets when columns are toggled on/off.

Useful regular expression templates

`^(?!word|otherword).*` This is a negative lookahead and will exclude all words between ! and). The bar works as an alternate symbol.

`(?=[abcdefgh]).*` This is a positive lookahead and will include any results with any of the characters inside the brackets.

Interactivity

Most tabs have a degree of interactivity. The data table tab includes download buttons. The variables plotted are changeable as are some features of the plot (e.g hide/show NAs values). The scatter plot is wrapped in a plotly function, allowing for zooming, panning and filtering by clicking on the plot and legend. The network graph is implemented with `visNetwork` and also includes interactive features such as dragging, creating, and removing nodes.

Useful links

<https://clinicaltrials.gov/api/gui>

<https://regex101.com/>

<https://clinicaltrials.gov/api/gui/ref/crosswalks>

Appendix F

8.2 Study Eligibility Criteria

Supplementary Table S1: Study selection criteria

Item	Inclusion Criteria	Exclusion Criteria
Population	<ul style="list-style-type: none"> Active psoriatic arthritis ≥18 years of age 	<ul style="list-style-type: none"> <18 years of age
Interventions/Comparators	<ul style="list-style-type: none"> Anti-TNFα agents and their biosimilars: adalimumab, etanercept, infliximab, certolizumab, golimumab Anti-IL-12/23 agent: ustekinumab Anti-IL-23 agents: guselkumab, tildrakizumab, risankizumab Anti-IL-17A agents: brodalumab, ixekizumab, secukinumab, bimekizumab Anti PDE-4 agent: apremilast JAK inhibitor agent: tofacitinib, upadacitinib CTLA-4 agent: abatacept cDMARDs: methotrexate, azathioprine, ciclosporin/ciclosporin A, leflunomide, sulfasalazine, oral/parenteral gold, 6-mercaptopurine, chloroquine, hydroxychloroquine, D-penicillamine, colchicine, etretinate, photochemotherapy/8-methoxypsoralen, somatostatin, bromocriptine, cimetidine, fumaric acid, 2-chlorodeoxyadenosine, parenteral nitrogen mustard, peptide T, radiation synovectomy with yttrium 90, total lymph node irradiation Placebo 	Non-pharmacologic treatments
Outcomes	<ul style="list-style-type: none"> No restriction on outcomes 	<ul style="list-style-type: none"> NA
Study design	<ul style="list-style-type: none"> Published phase II, II/III, and III RCTs Conference abstracts and posters (title and abstract screening phase) 	<ul style="list-style-type: none"> Non-randomized, single-arm, or observational studies Open-label extension phases of RCTs Pre-clinical studies, case reports, expert opinion articles, letters, narrative (non-systematic) reviews Conference abstracts and posters (full-text screening phase) Phase I and Phase I/II RCTs Pilot studies Phase IV studies
Study Duration	<ul style="list-style-type: none"> ≥12 weeks 	<ul style="list-style-type: none"> <12 weeks
Study Language	<ul style="list-style-type: none"> English 	<ul style="list-style-type: none"> Non-English
Date Restrictions	<p>Title and Abstract Screening Phase</p> <ul style="list-style-type: none"> Conference abstracts and posters from the last two years (2018-2020)^a No date restrictions for full-text publications 	<p>Title and Abstract Screening Phase</p> <ul style="list-style-type: none"> Conference abstracts and posters before 2018^a <p>Full-text Screening Phase</p> <ul style="list-style-type: none"> All conference abstracts and posters

^aAbstracts published from 2016 to 2018 were eligible for inclusion in the title and abstract screening phase in the original SLR. Only abstracts published from 2018 to 2020 were eligible for inclusion in the title and abstract screening phase in the updated SLR. cDMARD: conventional disease-modifying anti-rheumatic drugs; CTLA-4: cytotoxic T-lymphocyte-associated antigen 4; IL: interleukin; JAK: janus kinase; PDE: phosphodiesterase; RCT: randomized controlled trial; TNF: tumor necrosis factor alpha.

Study Eligibility Criteria from page 42 of supplementary material accompanying Mease et al., (2021).