PMC explorer app demo overview

Goals:

* Make some package functionality available for people who don’t write R code
* Look at trends in data sharing information across the years 2017 to 2019, and not just for 2018 (the data used for training the model zoo)
* Work on a new way to extract data sharing information from individual documents (this turned into a text-annotation function, inspired by ODDPub)

Introduction and walk-through each tab:

* First you’ll see a pop-up screen explaining a bit about the project and data sharing in general. You can exit this by clicking out of the box.
* There’s also the option of doing an intro tour, to understand a bit better what each tab does
* We can preview this, but will also walk through each tab in more depth
* Tabs: explore by id, data sharing trends, annotate document
  + Explore by id: input a PMCID and get associated information: title, abstract, MeSH terms, and data availability information
  + Data sharing trends: input a date range (month and year) and get visualizations of different trends
  + Annotate text: input a PMCID and get text highlighted with different data sharing words and phrases

Explore by ID:

* This is just leveraging a few tools in the pmcexplore package to access the E-Utils API
* Retrieves basic information: title, abstract, MeSH terms (if available), and data availability (if available)
* Through the functions, you can access all papers, not just ones that are open access
  + If a paper’s XML is available, the DAS function will just return “The full XML of this document is not available on PubMed Central.”
* Added another function this morning that prints out the metadata associated with the document (author, journal, date published, etc)
* This tab serves to just give a broader overview of a document, with special attention to data sharing information

Data sharing trends:

* Here we get a bit more into the focus of this project, by looking at the ways data availability is tagged in papers
* In pmcexplore, I wrote a function called `get\_DAS` which gets all the data availability information of a document. This function picks up on data availability statements, sections with titles related to data availability (like Data Accessibility or Data Archiving Statement), supplementary materials, and inline figures and tables (this was more recently added)
* In pmcexplore, there are a few datasets included in the package itself. I used the get\_DAS function to get all data sharing information from January 2017 to December 2019. This data was predownloaded into the package for more easy accessibility (especially for the app, which should have pretty instant reactions).
* We can see a few different visualizations made from these data
* The most interesting and significant find that I noticed from playing around with this functionality of the app was seeing the rise in actual DASs from January 2017 to December 2019
  + Pull this up!
  + We can see that DAS tagging goes from 0 to around 1000 papers per month with data availability statements

Annotate text:

* The last function is actually a new feature in the pmcexplore package
* Near the beginning of the summer, Lisa sent information on this very cool package called ODDPub, which determines if a document has open data or open code based off the document text
* ODDPub uses a series of rules (like the appearance of a certain word in proximity of a different certain word) to determine whether a phrase indicates open data or open code
* I took inspiration from this, used the same keywords to pick out data information, and created a newer (and simpler) set of rules to detect data sharing information.
* The algorithm detects 5 different categories of phrases related to data sharing:
  + Data availability statement
  + No data available
  + External sources (like repositories, github, etc)
  + Supplementary materials and files
  + Any tandem of data-related words near availability-related words
* You can choose to highlight any of these categories of data sharing in the annotation function by clicking on the check boxes
* I think this is the most exciting new component I’ve worked on, because it also has a lot of potential to be expanded such as:
  + The rules can be revised to increase specificity and/or sensitivity
  + Instead of using this set of rules, what if annotating was done by a machine learning model?
  + There’s a possibility of using the current annotations to curate a training dataset, where we could learn which phrases are associated with certain data sharing practices

Over the past 5 weeks, I’ve also done the following:

* Revised the DAS function to pick up inline figures and tables
* Created a suite of functions to handle MeSH terms, including one that embeds the MeSH terms of a document into data that can be used to train models
* Extended documentation and added some datasets to the pmcexplore package
* I am also almost done with some transition documents, so that folks in the future can easily use the package or write more code for the package