“Phytochemical ontologies for analyzing the literature on essential oils”

SUBMITTED TO

THE NATIONAL INSTITUTE OF PLANT GENOME RESEARCH

AS A PARTIAL FULFILLMENT OF A DISSERTATION

SUBMITTED BY

RADHU KANTILAL LADANI

JUNE 2021

UNDER THE GUIDANCE OF,

PROJECT CO-ORDINATOR AND INTERN GUIDE

Dr. Gitanjali Yadav
Staff Scientist V
National Institute of Plant Genome Research (NIPGR), New Delhi, India
Lecturer Department of Plant Science,
University of Cambridge, UK
gy@nipgr.ac.in | gy@nipgr.ac.in

PROJECT HEAD AND INTERN SUPERVISOR

Dr. Peter Murray- Rust
Reader Emeritus in Molecular Informatics
Yusuf Hamied Department of Chemistry
University of Cambridge, UK
pm286@cam.ac.uk

THE NATIONAL INSTITUTE OF PLANT GENOME RESEARCH

CERTIFICATE

This is to certify that the project dissertation entitled, “Phytochemical ontologies for analyzing the literature on essential oils” to be submitted by Radhu Ladani for the partial fulfillment of M.Sc. (Bioinformatics) degree of Bharati Vidyapeeth Deemed University, Pune is absolutely based upon her own work which has been undertaken with my supervision since January’ 2021 to June’ 2021. Neither this review dissertation nor any part of it has been submitted for any degree or diploma elsewhere.

DECLARATION

I hereby declare that this written submission represents my ideas in my own words and where other’s ideas or words have been included; I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea, data, and facts in my submission.

Radhu Ladani

ACKNOWLEDGEMENT

I would like to acknowledge the support and guidance I have received during my dissertation. I am thankful to The **National Institute of Plant Genome Research (NIPGR)** for giving me the opportunity during the pandemic time.

I would like to express my sincere gratitude to **Dr. Gitanjali Yadav** (Scientist, NIPGR; Lecturer, University of Cambridge) for her support, supervision as well as encouragement during this internship. I am also obliged to **Dr. Peter Murray Rust** (Reader Emeritus, University of Cambridge) for his insightful ideas, patient guidance and life-impacting mentoring have helped me through this project. I am especially grateful for his advice, encouragement, and inspiration during my completion.

This endeavor would not have been possible without the constant and unconditional support of all my co-interns. I thank them from the bottom of my heart for their wonderful companionship. This internship has been a very good experience for me in a way that it has given me the chance to understand the real world outside the classroom.

TABLE OF CONTENTS

OBJECTIVES

The present research work entitled **“Phytochemical ontologies for analyzing the literature on essential oils”** was conducted remotely at the National Institute of Plant Genome Research (NIPGR), New Delhi, India with the following objectives:

- To establish a dictionary that can be used to search for annotated scientific literature.
- To formulate a corpus of medicinal activity and essential oils by utilizing the getpapers toolkit, which is a web scraper for open-source scientific literature.
- To get the possible insights from the open scientific literature regarding the association between the various essential oil plants and compounds with their medicinal activity.

CHAPTER 1

INTRODUCTION

To manage and resolve the critics in the open-source literature that is available and to derive useful information from the literature, it is critical to develop and maintain credible knowledge tools. This project will develop and implement modern and open source tools such as Wikidata (and Wikipedia), Python, Java, and data mining in combination with conceptual tools for discovering, combining, cleaning, and semantically categorizing scholarly documents that contain significant amounts of phytochemicals. In this project, we focused on the activity of oils derived from plants as reported in the open scientific literature, which contains thousands of articles describing oils extracted from specific plants, their chemical constituents, and their biological and medicinal properties. To accomplish the aforementioned goal, we develop an automated system that reads scientific literature and extracts to the meaningful context of phytochemicals.

The rationale of this project is to make information accessible to the community in an uncomplicated and coherent manner. In the OpenNotebook philosophy, we used GitHub as a storage portal, where all work is accomplished as OpenNotebookScience, where all activities are completely transparent and all resources are versioned. The project has grown significantly and branched out into different areas of research. These repositories are where the most recent dictionaries, mini-corpora, and software are located.

CEVOpen is a global network initiative led by a collaborative group of young scientists. This young scientific community is a group of individuals from diverse fields such as biosciences, statistics, mathematics, computational biology, computer science, etc. and they give their valuable contribution. CEVOpen's work is presented at various places, including Wikcite, COAR, and the Flash Forward Workshop, under the direction of Dr. Peter Murray-Rust.

This thesis describes the methodology performed particularly for essential oils titled ‘Phytochemicals ontologies for the analyzing the literature on essential oil’.

1.1 CEVOpen

CEVOpen project aims to develop coherent knowledge tools and resources for automatic conversion of the open scientific literature to a semantic atlas of plant chemistry and properties. CEVOpen (**ContentMine, EssoilDB,** and **Verriclear**) are the three organizations that have started this project which is an **Open**Notebook to facilitate the scientific approach which contains all primary records of research projects that are stored on any type of data portal that makes them publicly accessible as they are created (Murray-Rust).

- **ContentMine (https://github.com/petermr/contentMine)**

ContentMine was funded by the Shuttleworth Foundation (Fellowship to Peter Murray-Rust, University of Cambridge, UK). ContentMine uses machines to automatically extract and interpret content from the literature. ContentMine works on the philosophy of creating an open resource for everyone which is also created by everyone.

- **EssoilDB (http://www.nipgr.ac.in/Essoildb/)**

The **ESS**ential **OIL** **D**ata**B**ase (Dr. Gitanjali Yadav, NIPGR, New Delhi, India) is a knowledge resource for plant's volatile emissions, containing experimental records of essential oil composition data, from published reports.

- **Verriclear (https://verriclear.com/)**

Verriclear was founded by Emanuel Faria, creates 100% plant-based natural skincare formulations for skin conditions. Verriclear Natural Skin Essentials Ltd., an innovative developer of phytotherapy skincare products derived from bioactive plant extracts from around the world.

1.2 Open Literature

Identifying pertinent scientific literature is a critical routine effort for researchers. Rapid access to a great volume of scientific literature is enabled by digital libraries and web information retrieval algorithms. Transparency, reproducibility, collection, inclusiveness, accessibility, accuracy, and re-use are the principles of open science. One of the most significant benefits of open access publishing is that it increases the visibility and reuse of academic research findings.

1.3 Medicinal Activity

Medicinal activity: Treatment of several human diseases is based on medicinal plants as they are less harmful and include specific properties like antimicrobial, antioxidant, anti-inflammatory, anti-cancer. These properties acquire a significant advantage for a medicinal plant to treat diseases. Also, these plants are natural-specific compounds like for example, alkaloids, glycosides, saponines, polyphenols, and flavonoids which are more tolerable than synthetic drugs and therefore have fewer cumulation problems and it is helpful for a longer time(Akthar, Degaga, and Azam 2014).

1.4 Essential Oil

Essential oils: Essential oils which are obtained from plants have abundant significance for the treatment of many diseases. These essential oils are hydrophobic liquids that are concentrated and contain volatile compounds. Some known examples are ethereal, aetherolea, oil from clove, etc. In the present day scenario, scientific literature contains ample articles which have the main purpose of extracting oils from plants, their chemical constituents, and important medicinal properties. These oils are usually extracted with the process of distillation including other processes like solvent extraction, absolute oil extraction, resin tapping, wax embedding, and cold pressing. Thus, the composition of oils is dependent on many factors such as the genotype of plants, environmental conditions, etc(Jaradat 2021).

The general purpose of the thesis is to generate a rapid proof of concept that will illustrate the phytochemical activities that can be retrieved from the literature using current tools.

CHAPTER 2

LITERATURE REVIEW

Paracelsus (16th century) used the term "essential oil," referring to the active ingredient in each medication as "Quinta essentia." Terpenoids and aliphatic and aromatic chemicals such as aldehydes and phenols are among the 500 chemicals found in essential oils. It is conceivable for the major components to account for up to 85% of the oil's overall content. There are an estimated 3,000 well-known essential oils, with 300 of them being widely marketed. The essential oil composition is influenced by a number of elements, including environmental conditions, soil composition, and growing techniques.

The antibacterial properties of oregano and thyme essential oils are attributed to the presence of their phenolic components, carvacrol, and thymol. The antimicrobial activity of oregano and thyme essential oils was proven in experiments comparing the two oils. Bacillus cereus, oregano, and thyme essential oils were found to be a little less effective than cinnamon essential oil, which had the most potent effect. One of the most effective is oregano, which inhibits the growth of all bacteria at a concentration of 1%. The antibacterial action of Ocimum micranthum essential oil, which is high in eugenol, was found to be stronger than that of Ocimum basilicum against E. faecalis, P. aeruginosa, and E. coli(Sakkas and Papadopoulou 2017).

Artemisia jordanica (AJ) is a folklore medicinal herb that thrives in the harsh conditions of the Al-Naqab desert and is used by Palestinian Bedouins to treat diabetes and gastrointestinal ailments. The current study aims to identify the components of (AJ) essential oil (EO) and evaluate EO's antioxidant, anti-obesity, antidiabetic, antibacterial, anti-inflammatory, and cytotoxic effects for the first time. The antioxidant, anti-obesity, and anti-diabetic properties of (AJ) EO were evaluated using recognized biochemical techniques, while the antioxidant, anti-obesity, and anti-diabetic properties were evaluated using the gas chromatography-mass spectrometer (GC-MS) technique. The broth microdilution assay was used to determine the microbicidal efficacy of (AJ) EO. In addition, the cytotoxic activity was calculated using the (MTS) method. Finally, using a COX inhibitory screening test kit, the anti-inflammatory activity was determined.

The existence of 19 molecules in the (AJ) EO was discovered by an analytical examination. Oxygenated terpenoids, such as bornyl acetate (63.40%) and endo-borneol (17.75%), were shown to be significant components of the (AJ) EO. When compared to Trolox, the EO had a strong antioxidant impact, but a modest anti-lipase impact when compared to orlistat. In addition, as compared to the positive control acarbose, the tested EO had a significant -

amylase inhibiting action. In comparison to the positive control acarbose, the (AJ) EO had a high -glucosidase inhibitory capability. The EO exhibited a cytotoxic impact on all of the tumor cells that were tested. In fact, (AJ) EO exhibited antibacterial activity. The antioxidant, antibacterial, antifungal, anti-amylase, anti-glucosidase, and COX inhibitory properties of the (AJ) EO make it a promising option for the treatment of neurological illnesses caused by damaging free radicals, microbial resistance, diabetes, and inflammation. Further research on the significance of such therapeutic plants is required(Akthar, Degaga, and Azam 2014).

Essential oils are complex volatile molecules that are produced spontaneously by various areas of the plant during secondary metabolism. Due to their antimicrobial capabilities against bacterial, fungal, and viral infections, a wide range of medicinal plants has been researched and employed for the extraction of essential oils all over the world. Essential oils are more precise in their mode of action against a wide variety of pathogenic bacteria due to the presence of a significant number of alkaloids, phenols, terpene derivatives, and other antimicrobial chemicals. As a result, essential oils could be better supplements or alternatives in the fight against harmful bacteria. The goal of this review article is to focus on the antibacterial activity of essential oils released by medicinal plants, as well as the mechanisms involved in pathogenic microorganism suppression.

Antimicrobial activity:

Antimicrobial activity has been tested on a range of essential oils. Plant-derived essential oils have antibacterial properties that are used in a variety of applications, including food preservation, aromatherapy, and medicine. According to Cowan (1999), there are now around 3,000 essential oils known. Of these, 300 are economically significant and widely used in the pharmaceutical industry.

Antibacterial activity:

Conner (1993) discovered that cinnamon, clove, pimento, thyme, oregano, and rosemary plants showed potent antibacterial properties against a variety of microorganisms. Due to the presence of phenolic components such as carvacrol, eugenol, and thymol, essential oils derived from various medicinal plants have been shown to have antibacterial activity against all five tested food-borne pathogens (Kim et al.). Arora and Kaur (1999) looked at the antimicrobial activity of garlic, ginger, clove, black pepper, and green chili against human pathogenic bacteria such as Bacillus sphaericus, Enterobacter aerogenes, E. coli, P. aeruginosa, S. aureus, S. epidermidis, S. typhi, and Shiguella flexneri, and found that aqueous garlic extracts were the most sensitive of all. Sakagami et al., (2000) studied the effect of clove extracts on the synthesis of verotoxin by enterohemorrhagic Escherichia coli O157:H7, and the study revealed that clove extracts suppressed verotoxin formation. Elgayyar et al. (2001) investigated the efficacy of essential oils of cardamom, anise, basil, coriander, rosemary, parsley, dill, and angelica.

Sakandamis et al., (2002) investigated the effects of oregano essential oils on Salmonella typhimurium behavior in sterile and naturally contaminated beef fillets maintained in aerobic and modified environments. They concluded that adding oregano essential oils to the mix prevented the bacterial pathogens understudy from reducing their initial population.

Antifungal activity:

Molds have been successfully combated using essential oils and their components. Essential oil extracts from a variety of plants, including basil, citrus, fennel, lemongrass, oregano, rosemary, and thyme, have demonstrated significant antifungal action against a variety of fungal infections (Kivanc, 1991). The sensitivity of essential oils of spices against various fungal infections was studied by Arora and Kaur (1999), who concluded that garlic and clove were the most sensitive. Candida acutus, Candida albicans, Candida apicola, and Candida catenulata were all found to be resistant to the extracts. Trignopsis variabilis, C.inconspicua, C. tropicalis, Rhodotorula rubra, Sacharomyces cerevisae, and C.inconspicua. Delaquis and Mazza (1995) reported that isothiocyanates extracted from onion and garlic plants had antibacterial properties and that isothiocyanates may inactivate extracellular enzymes by oxidative disulfide bond cleavage. Because essential oils are present in most therapeutic plants, they have antibacterial properties. The antibacterial action of essential oils is influenced by nature, structural composition, and functional groups contained in them. Essential oils contain a wide range of volatile molecules, including terpenes and terpenoids, as well as aromatic and aliphatic chemicals generated from phenols. Essential oils immediately influence the pathogenic microorganism's cell membrane, increasing permeability and allowing key intracellular ingredients to seep out, ultimately disrupting cell respiration and the microbial enzyme system. Furthermore, according to their kind and concentration, they had cytotoxic effects on live cells. As a result, it has been proposed that essential oils be used(Jaradat 2021).

CHAPTER 3

MATERIALS AND METHODS

3.1 Tools and Databases

➢ getpapers

Getpapers is a simple yet powerful tool for searching scholarly article repositories using a single-line command. getpapers can retrieve article metadata, fulltexts (PDF or XML), and supplementary using any of the APIs, including EuropePMC, IEEE, ArXiv, and Crossref. In the getpapers default configuration, the EuropePMC API is used. getpapers is a convenient tool for rapidly acquiring large numbers of papers for reading or bibliometric analysis.

Primary URL: https://github.com/ContentMine/getpapers

Installation:https://github.com/petermr/tigr2ess/blob/master/installation/windows/INSTALLATION.md

1. Go to the nvm-windows and download the latest version of **`nvm-setup.zip`**
2. To install the node, run **`nvm install latest`** in the command-line
3. To install the getpapers, run **`npm install --global getapapers`**

Run the command **`getpapers`** in the command line to check the successful installation and it gives the command option used for getpapers as below:

```
C:\Users\DELL>getpapers
 Usage: getpapers [options]
 Options:

    -h, --help                output usage information
    -V, --version             output the version number
    -q, --query <query>       search query (required)
    -o, --outdir <path>       output directory (required - will be created if not found)
    --api <name>              API to search [eupmc, crossref, ieee, arxiv] (default: eupmc)
    -x, --xml                 download fulltext XMLs if available
    -p, --pdf                 download fulltext PDFs if available
    -s, --supp                download supplementary files if available
    -t, --minedterms          download text-mined terms if available
    -l, --loglevel            amount of information to log (silent,verbose,info*,data, warn, error, debug)
    -a, --all                 search all papers, not just open access
    -n, --noexecute           report how many results match the query,but don't actually download anything
    -f, --logfile             save log to specified file in output directory and printing to terminal
    -k, --limit <int>         limit the number of hits and downloads
    --filter <filter object>  filter by key value pair, passed straight to the crossref api only
    -r, --restart             restart file downloads after failure
```

Figure 1: getpapers visualization on command-line

General syntax: `getpapers -q <"project title"> -o <output directory> -x<xml> -p<pdf> -k <number of papers required>`

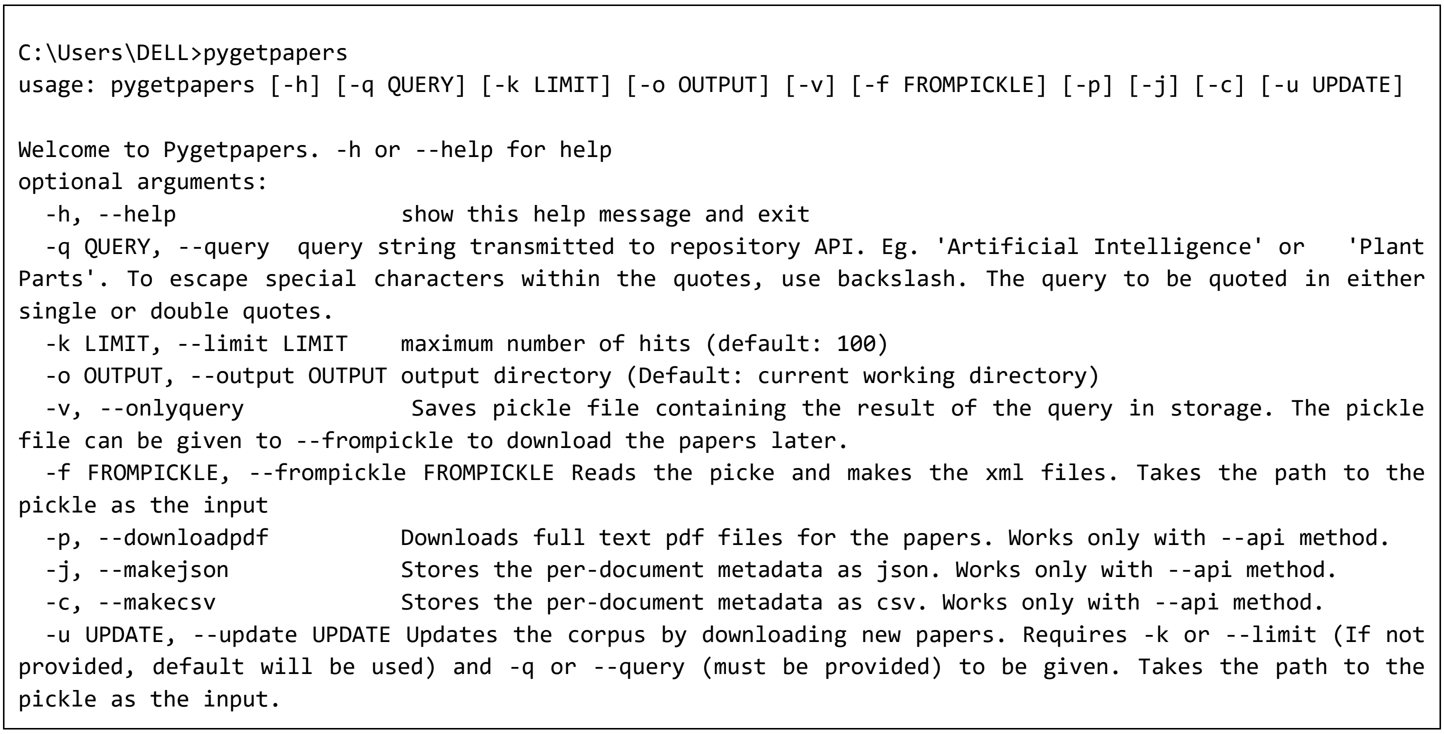
➢ pygetpapers

pygetpapers is a python version of getpapers that helps text miners with their work. This software has been developed to interface with access to open scientific text repositories, make requests to those repositories, gathers hits, and downloads the articles in a systematic and non-interactive manner.

Primary URL: https://github.com/petermr/pygetpapers
Installation: https://github.com/petermr/pygetpapers/blob/main/README.md#6-installation

1. Download python along with pip from: **`https://www.python.org/downloads/`**
2. Cloned the repository using **`git clone`** command to the local computer**: `git clone https://github.com/petermr/pygetpapers`**
3. Run the command: **`pip install git+git://github.com/petermr/pygetpapers`**

Run the command **`pygetpapers`** in the command line to check the successful installation and it gives the command option used for getpapers as below:



```
C:\Users\DELL>pygetpapers
usage: pygetpapers [-h] [-q QUERY] [-k LIMIT] [-o OUTPUT] [-v] [-f FROMPICKLE] [-p] [-j] [-c] [-u UPDATE]

Welcome to Pygetpapers. -h or --help for help
optional arguments:
  -h, --help                show this help message and exit
  -q QUERY, --query  query string transmitted to repository API. Eg. 'Artificial Intelligence' or   'Plant
Parts'. To escape special characters within the quotes, use backslash. The query to be quoted in either
single or double quotes.
  -k LIMIT, --limit LIMIT    maximum number of hits (default: 100)
  -o OUTPUT, --output OUTPUT output directory (Default: current working directory)
  -v, --onlyquery            Saves pickle file containing the result of the query in storage. The pickle
file can be given to --frompickle to download the papers later.
  -f FROMPICKLE, --frompickle FROMPICKLE Reads the picke and makes the xml files. Takes the path to the
pickle as the input
  -p, --downloadpdf          Downloads full text pdf files for the papers. Works only with --api method.
  -j, --makejson             Stores the per-document metadata as json. Works only with --api method.
  -c, --makecsv              Stores the per-document metadata as csv. Works only with --api method.
  -u UPDATE, --update UPDATE Updates the corpus by downloading new papers. Requires -k or --limit (If not
provided, default will be used) and -q or --query (must be provided) to be given. Takes the path to the
pickle as the input.
```

Figure 2: pygetpapers visualization on command-line

General syntax: `pygetpapers -q <"project title"> -o <output directory> -x<xml> -p<pdf> -k <number of papers required> -c <csv metadata file>`

➢ ami

Ami is a novel toolkit for querying and analyzing a small-to-medium collection of documents, usually on local storage. ami is a declarative system comprised of commands and data modules and is written in Java. ami turns documents into knowledge. It includes features tools for downloading scientific papers, processing documents into sections and XML, analyzing components (text, tables, diagrams), creating dictionaries, and searching.

Primary URL: https://github.com/petermr/ami3
Installation: https://github.com/petermr/openVirus/wiki/INSTALLING-ami3

1. Download the backend software such as java, jdk, maven and git and set the path for them
2. Open the command line and git clone the repository ami3: **`git clone https://github.com/petermr/ami3`**
3. In ami3 path, run the command: **`mvn install -Dmaven.test.skip=true`**

❖ ami section

Ami section is used to divide research papers into the following sections: front, body, back, floats, and groups. Sectioning downloaded files creates a tree structure for us, which aids in navigating the file's content. Sectioning is accomplished through the use of ami's section function. Which is executed via the command prompt.

General syntax: `ami -p <cproject> section`

❖ ami search

Ami search analyses and searches the keywords in the project repository, returning the term's frequency data table and the corpus's histogram.

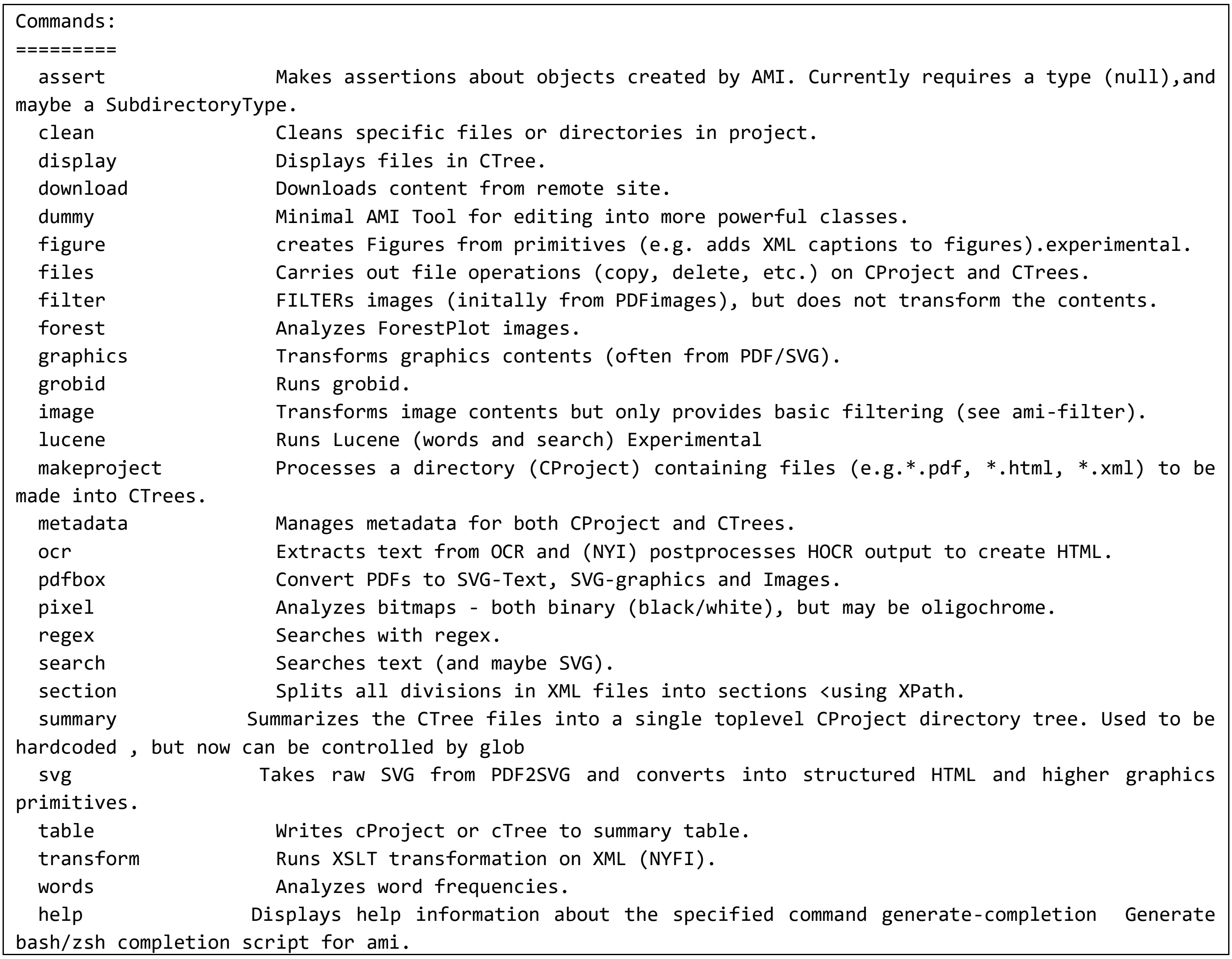
General syntax: `ami –p <cproject><directory> search –dictionary <path>`

❖ ami dict

Ami dict is a set of commands for converting the output of SPARQL endpoints to the dictionary format.

General syntax: `amidict -vv –dictionary <name of dictionary> --directory <Path of the directory folder> --input <SPARQL endpoint output name> create --informat wikisparqlxml --sparqlmap wikidataURL = item, name = itemLabel, term = itemLabel --transformName wikidataID=EXTRACT(wikidataURL,./(.))`

Run the command **`ami -- help`** in the command line to check the successful installation and it gives the command option used for getpapers as below:



```
Commands:
=========
  assert               Makes assertions about objects created by AMI. Currently requires a type (null),and
maybe a SubdirectoryType.
  clean                Cleans specific files or directories in project.
  display              Displays files in CTree.
  download             Downloads content from remote site.
  dummy                Minimal AMI Tool for editing into more powerful classes.
  figure               creates Figures from primitives (e.g. adds XML captions to figures).experimental.
  files                Carries out file operations (copy, delete, etc.) on CProject and CTrees.
  filter               FILTERs images (initally from PDFimages), but does not transform the contents.
  forest               Analyzes ForestPlot images.
  graphics             Transforms graphics contents (often from PDF/SVG).
  grobid               Runs grobid.
  image                Transforms image contents but only provides basic filtering (see ami-filter).
  lucene               Runs Lucene (words and search) Experimental
  makeproject          Processes a directory (CProject) containing files (e.g.*.pdf, *.html, *.xml) to be
made into CTrees.
  metadata             Manages metadata for both CProject and CTrees.
  ocr                  Extracts text from OCR and (NYI) postprocesses HOCR output to create HTML.
  pdfbox               Convert PDFs to SVG-Text, SVG-graphics and Images.
  pixel                Analyzes bitmaps - both binary (black/white), but may be oligochrome.
  regex                Searches with regex.
  search               Searches text (and maybe SVG).
  section              Splits all divisions in XML files into sections <using XPath.
  summary            Summarizes the CTree files into a single toplevel CProject directory tree. Used to be
hardcoded , but now can be controlled by glob
  svg                 Takes raw SVG from PDF2SVG and converts into structured HTML and higher graphics
primitives.
  table                Writes cProject or cTree to summary table.
  transform            Runs XSLT transformation on XML (NYFI).
  words                Analyzes word frequencies.
  help               Displays help information about the specified command generate-completion  Generate
bash/zsh completion script for ami.
```

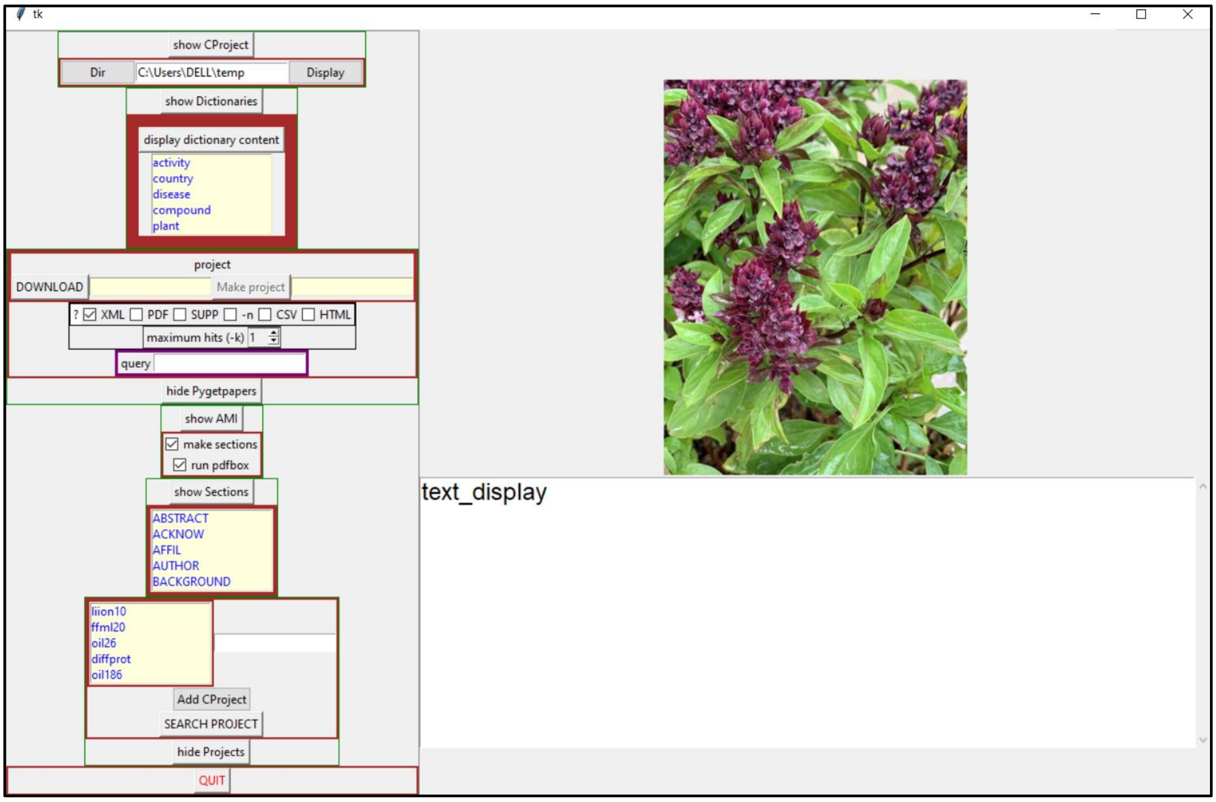
Figure 3: ami visualization on command-line

➢ pyami/ami_gui.py

pyami is a search engine similar to the ami interface for reading and analyzing documents. pyami displays the search frequency values graphically. ami_gui.py provides an interface from which we can conduct direct searches by clicking on the desired mini corpus, dictionaries, or sections, which return results in graphical format. Additionally, we can download papers from pygetpapers using this interface. The ami_gui.py interface includes the following checkboxes (Murray-Rust, https://github.com/petermr/openDiagram, 2021).

- Show dictionaries give main core dictionaries [activity, country, disease, compound, plant, plant_genus, organization, plant_compound, plant_part, invasive_plant]
- Show section extract only the sections of interest [abstract, acknowledge, affiliation, author, background, discussion, empty, ethics, fig_caption, front, introduction, jrnl_title, keyword, method, material, octree, pdfimage, pub_date, publisher, reference, results, results, sections, svg, table, title, word]
- Show Project gives number of hardcoded corpora / projects including [liion10, ffml20, oil26, oil186, cct, disease, diffprot, worc_synth, worc_explosion, activity, hydrodistil, invasive, plantpart]

We require the following to run ami gui: Cloned the repositories for CEVOpen, dictionary, and opemDiagram. To modify the source code in accordance with the HOME directory of the system. The file should be saved in the same directory as the previous one and executed via the command line. After running the command **python ami gui.py** on path **openDiagram\physchem\python** the command line, the following window screen is displayed.



Dir C:\Users\DELL\temp

? ☑ XML □ PDF □ SUPP □ -n □ CSV □ HTML

maximum hits (-k) 1

☑ make sections

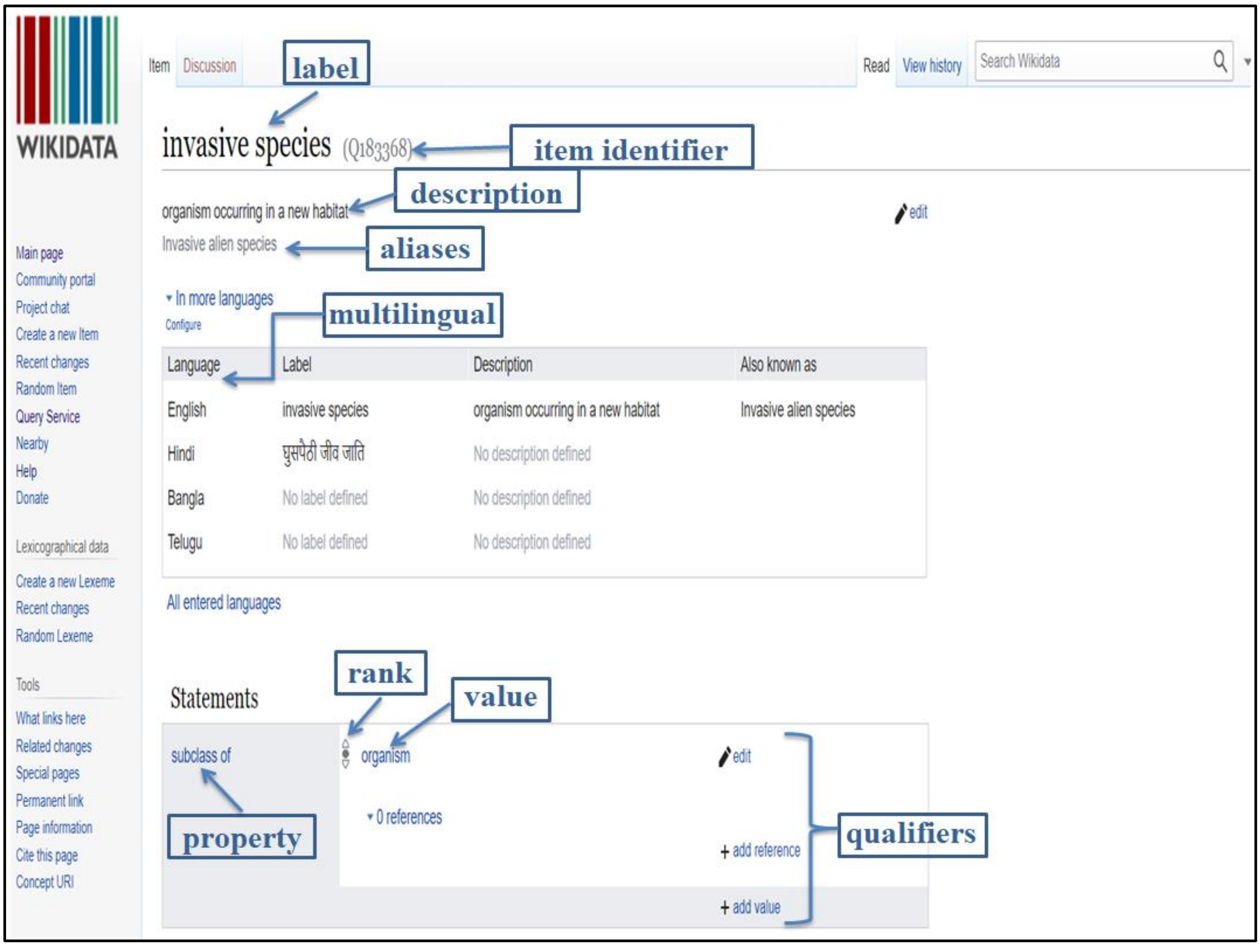
☑ run pdfbox

Figure 4: ami_gui interface visualization

➢ Wikidata

Wikidata is a collaborative knowledge bases secondary database for structured data that is primarily used by the Wikimedia family of projects. Wikidata possesses numerous necessary characteristics for scientific knowledge, including multilingualism, human and machine editability, and a linked functionality approach.

Wikidata is structured in triples and primarily consists of items, each of which has a label, a description, and an unlimited number of aliases. Items are uniquely identified by a Q followed by a number. All of this information is available in a variety of languages even if data originated in different languages. In wikidata, statements describe an item's detailed characteristics and are composed of a property and a value.



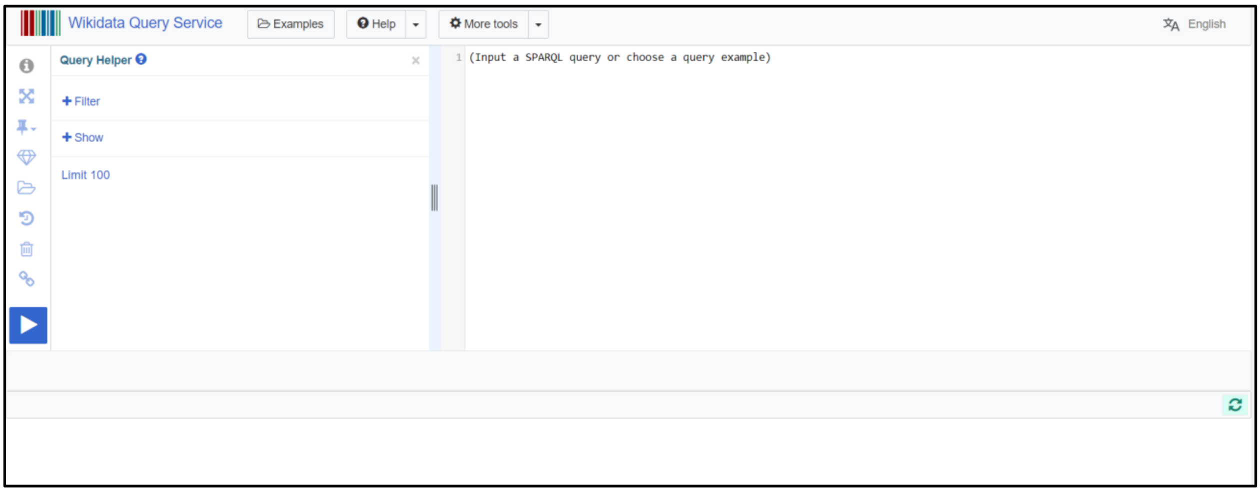
Language	Label	Description	Also known as
English	invasive species	organism occurring in a new habitat	Invasive alien species
Hindi	घुसपैठी जीव जाति	No description defined	
Bangla	No label defined	No description defined	
Telugu	No label defined	No description defined	

Figure 5: Wikidata search page

The popular tools for searching and examining wikidata items are the Wikidata Query Service, Geneawiki, Reasonator, and Tree of life. Additionally, we can independently retrieve all data via the wikidata API.

➢ Wikidata: SPARQL query service

The Wikidata Query Service is powered by SPARQL which is a semantic query language to formulate queries using knowledge databases. The pilot SPARQL endpoint included a graphical user interface for query construction. SPARQL enables the extraction of semantically rich data via queries composed of logical triple combinations. SPARQL operates on a knowledge graph database, such as Wikipedia, and enables the extraction of knowledge and information through the use of filters and constraints.



```
(Input a SPARQL query or choose a query example)
```

Figure 6: wikidata query service page

Wikidata Query Service enables the extraction of specific information from Wikidata's vast network of linked and structured data. SPARQL contains parameters such as SELECT returns values of variables or variable or expressions and results are table values, ASK to return true/false, DESCRIBE return a description of a resource, CONSTRUCT queries can build RDF triples/graphs.

3.2 Basic workflow of the tools

This is the process workflow for searching open repositories using linked data dictionaries, based on wikidata, retrieving metadata, and performing machine learning analysis to produce catalogs and knowledge graphs.

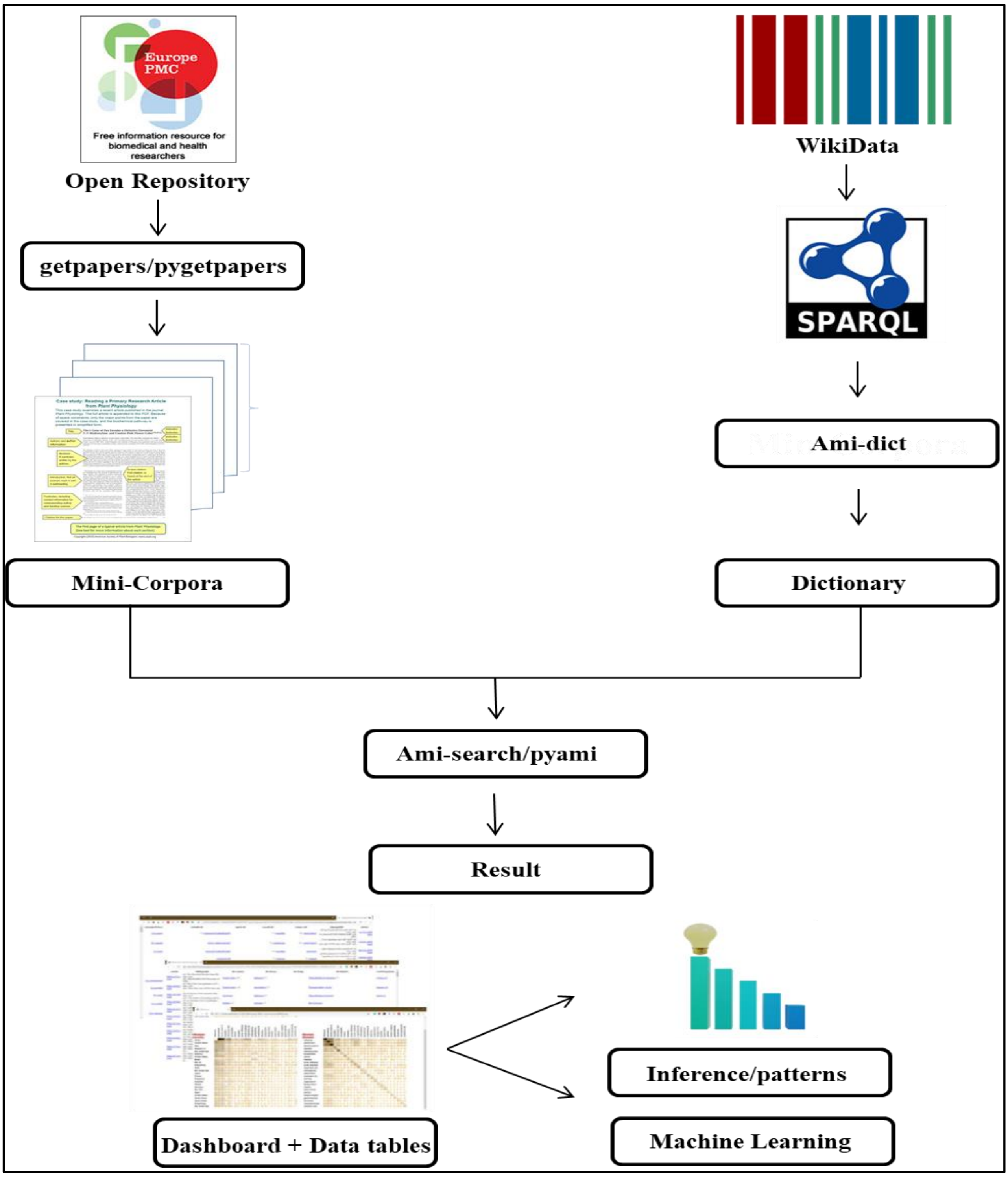


Figure 7: Workflow of the tools used

3.3 Methodology

3.3.1 Creation of Mini-Corpus “activity”

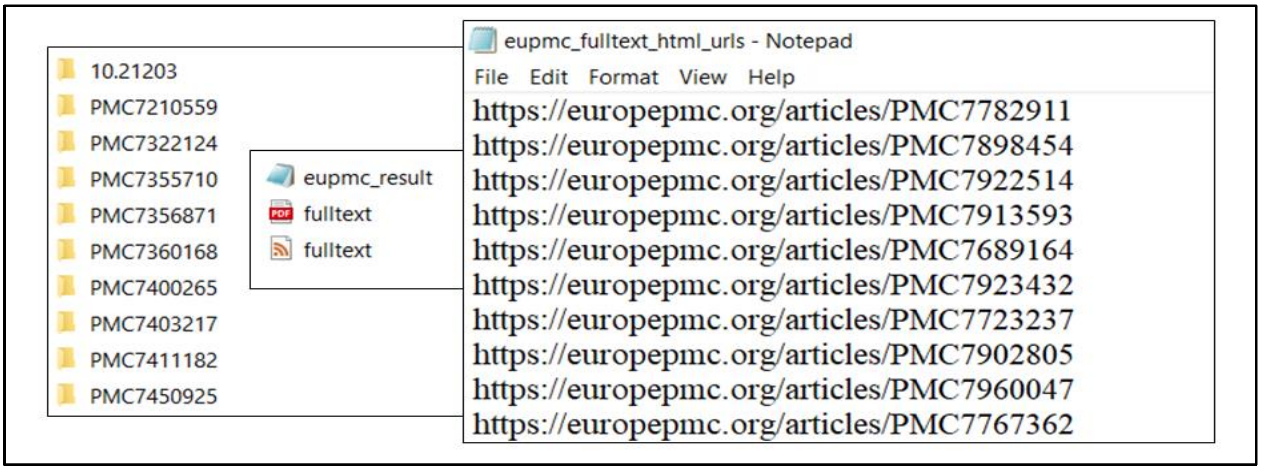
The tool named “getpapers”, was used to build a mini-corpus of open scientific literature on “medicinal activity and essential oil” from EuroPMC, a platform that offers free access to millions of articles in the field of biomedical science. The software is very quick to process (approximately 10 minutes), whereas downloading it individually could have taken 'n' number of hours. The command for the creation of the mini-corpus activity is listed below:

```
getpapers -q "(medicinal activity) AND (essential oil)" -o activity -x -p
-k 100 -f activity/log.txt
```

The command getpapers will initiate the process and -q refers to the query, which is to be searched. The query is entered in inverted commas as is done in "**(medicinal activity) AND (essential oil)**". The next element is -o which refers to the output directory and the parameter that follows it in the name of the directory, which is an **activity** in my case. Then, -x -p corresponds to xml and pdf files to be included in the search, and -k 100 limits our search to 100 files only. After successful completion of the command, I got the Corpus of 100 papers and is available at the following link:

https://github.com/petermr/CEVOpen/tree/master/minicorpora/activity

As shown in the image below, this query aids in the creation of a corpus of 100 research articles in full text and xml file format on a local machine.


eupmc_fulltext_html_urls - Notepad
File Edit Format View Help

10.21203
PMC7210559
PMC7322124
PMC7355710
PMC7356871
PMC7360168
PMC7400265
PMC7403217
PMC7411182
PMC7450925

eupmc_result
fulltext
fulltext

https://europepmc.org/articles/PMC7782911
https://europepmc.org/articles/PMC7898454
https://europepmc.org/articles/PMC7922514
https://europepmc.org/articles/PMC7913593
https://europepmc.org/articles/PMC7689164
https://europepmc.org/articles/PMC7923432
https://europepmc.org/articles/PMC7723237
https://europepmc.org/articles/PMC7902805
https://europepmc.org/articles/PMC7960047
https://europepmc.org/articles/PMC7767362

Figure 8: Directory of the local machine contains fulltext papers and EuroPMC URL

Additionally, pygetpapers can also be used to build a corpus and is well-suited to a modular approach in terms of both content and functionality.

3.3.2 Creation of Dictionary “activity”

Dictionaries are collections of terms accompanied by supporting information such as descriptions, provenance, and most importantly, links to other terminological resources, most notably Wikidata. The purpose of the project's dictionaries is to:

- Identify words and phrases ("entities") within the documents.
- To establish connections between their meaning and context ("ontologies").
- To assemble a subset of terms that express a high-level concept of plant chemistry and properties

Structure of dictionary

The format of dictionaries is straightforward and is best supported by XML or JSON. This section defines specific elements and their associated attributes.

- ❖ **Dictionary/Title:** This is the root element containing the title, and must be a single word and MUST be the filename's base.
- ❖ **Header/Description:** There are zero or more < desc >description elements in the header. These can include metadata about dates, maintenance, and provenance.
- ❖ **Entry/Body:** A dictionary's primary component is its entries. An entry is a well-defined object that is typically associated with a Wikidata item. This assigns it a unique identifier (Q-number), obviating the need for ongoing identifier maintenance.

The following procedure used to create an “activity” dictionary is as follows:

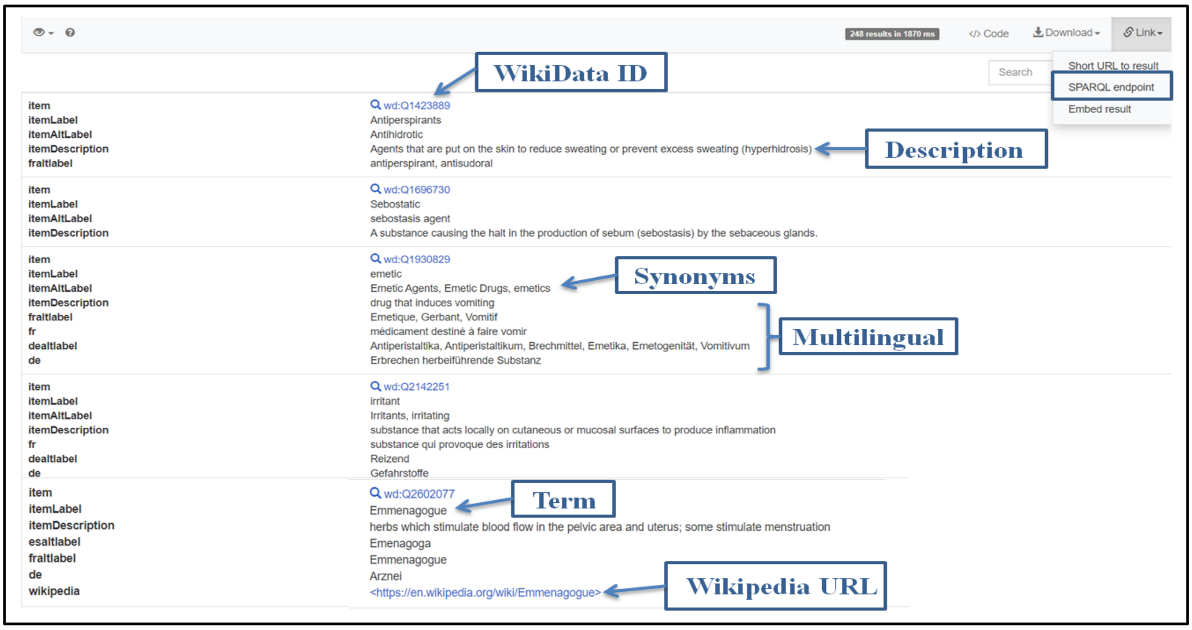
1. To collect wikidata IDs from the existing dictionary of activity, which is a dictionary of 438 essential oil or constituent compound biochemical and/or biological activities, 340 of which are resolved to wikidata IDs, and 336 with descriptions of 250 characters or less, created by Emanuel Faria.
2. In this project, I had to create a dictionary from the wikidata which is an open-source database that stores information in a semantically organized manner.
3. Go to https://www.wikidata.org/wiki/Wikidata:Main_Page and click on 'Query Service' in the left column. This will redirect to the Wikidata Query Service page where I can create a SPARQL query for the activity dictionary.

4. The following SPARQL query was used for the creation of the dictionary activity using the wikidata ID as values. This query contains the item, label, itemLabel, itemAltLabel, and different languages such as English, Hindi, Tamil, Spanish, French, German, Chinese, and Urdu.

```
## Selecting the preferred label
SELECT * WHERE {
  VALUES ?item {
wd:Q1069606  wd:Q11905748  wd:Q1225289  wd:Q12529398  wd:Q131207  wd:Q131656  wd:Q131746  wd:Q133948   wd:Q1340459
wd:Q1349821  wd:Q1384342  wd:Q1423889  wd:Q1468324  wd:Q14862699  wd:Q1509074  wd:Q1517948  wd:Q1536078  wd:Q1642182
wd:Q1660194  wd:Q166774  wd:Q167377  wd:Q16909071  wd:Q1696730  wd:Q17100700  wd:Q173235  wd:Q1734091  wd:Q178266
wd:Q181322  wd:Q18349602  wd:Q18356742  wd:Q18388377  wd:Q18663259  wd:Q186752  wd:Q187661  wd:Q187689  wd:Q190012
wd:Q190334 wd:Q1926141 wd:Q1930829 wd:Q193237 wd:Q1941660 wd:Q194270 wd:Q1976211 wd:Q1981368 wd:Q200656 wd:Q206348
wd:Q2088972 wd:Q209717 wd:Q21045470 wd:Q211036 wd:Q21139980 wd:Q211420 wd:Q2142251 wd:Q215118 wd:Q217972 wd:Q223417
wd:Q2255024  wd:Q246181  wd:Q249619  wd:Q2592323  wd:Q2602077  wd:Q26697606  wd:Q274083  wd:Q2742649  wd:Q274493
wd:Q2853144  wd:Q2853342  wd:Q2853347  wd:Q288280  wd:Q3009547  wd:Q309035  wd:Q324089  wd:Q3410946  wd:Q3411675
wd:Q3427345  wd:Q3446580  wd:Q3482889  wd:Q352551  wd:Q3560867  wd:Q357896  wd:Q3705665  wd:Q377458  wd:Q3774852
wd:Q3774857 wd:Q378705 wd:Q3817359 wd:Q3922210 wd:Q4008956 wd:Q407752 wd:Q416014 wd:Q41602594 wd:Q421700 wd:Q421978
wd:Q430719  wd:Q445580  wd:Q4669896  wd:Q4713968  wd:Q4742080  wd:Q4803792  wd:Q486061  wd:Q4990531  wd:Q50377176
wd:Q50415114 wd:Q50429885 wd:Q50430113 wd:Q50430144 wd:Q50430264 wd:Q50430265 wd:Q5119340 wd:Q513122 wd:Q5166679
wd:Q521616  wd:Q522817  wd:Q56062995  wd:Q567709  wd:Q572294  wd:Q575062  wd:Q575136  wd:Q575222  wd:Q575890  wd:Q576618
wd:Q578726  wd:Q581102  wd:Q581996  wd:Q582559  wd:Q582687  wd:Q584209  wd:Q5958197  wd:Q608085  wd:Q62903  wd:Q62962
wd:Q66295 wd:Q676558 wd:Q68541106 wd:Q6881918 wd:Q7187720 wd:Q721432 wd:Q7250944 wd:Q84958628 wd:Q7251487 wd:Q73984
wd:Q7431537  wd:Q745130  wd:Q76797715  wd:Q7833952  wd:Q7902662  wd:Q827658  wd:Q846227  wd:Q847705  wd:Q84944531
wd:Q84951095   wd:Q84953056   wd:Q84953547   wd:Q84953576   wd:Q84953633   wd:Q84953651   wd:Q84954230   wd:Q84954685
wd:Q84955111   wd:Q84955132   wd:Q84955175   wd:Q84956389   wd:Q84956474   wd:Q84956492   wd:Q84956495   wd:Q84956500
wd:Q84956514   wd:Q84956686   wd:Q84956852   wd:Q84956887   wd:Q84957317   wd:Q84957398   wd:Q84957440   wd:Q84957471
wd:Q84957488   wd:Q84957489   wd:Q84957495   wd:Q84957504   wd:Q84957506   wd:Q84957510   wd:Q84957514   wd:Q84957515
wd:Q84958628   wd:Q84958741   wd:Q84958793   wd:Q84959117   wd:Q84959304   wd:Q84959377   wd:Q84959751   wd:Q84959790
wd:Q84960246   wd:Q84960335   wd:Q84960524   wd:Q84961334   wd:Q84961500   wd:Q84961820   wd:Q84961856   wd:Q84961940
wd:Q84962003   wd:Q84962361   wd:Q84962587   wd:Q84962840   wd:Q84962992   wd:Q84963984   wd:Q84997245   wd:Q84997315
wd:Q84997332   wd:Q84997335   wd:Q84997870   wd:Q84998040   wd:Q84998042   wd:Q84998043   wd:Q84998051   wd:Q84998059
wd:Q84998172   wd:Q84998248   wd:Q84998654   wd:Q84999146   wd:Q84999154   wd:Q85001503   wd:Q85001558   wd:Q85001732
wd:Q85001844   wd:Q85001852   wd:Q85001855   wd:Q85001861   wd:Q85002068   wd:Q85002288   wd:Q85002611   wd:Q85002666
wd:Q85002964 wd:Q85003091 wd:Q85003128 wd:Q85003208 wd:Q85003209 wd:Q85003234 wd:Q85003391 wd:Q886593 wd:Q901434
wd:Q901537 wd:Q901656 wd:Q905101 wd:Q905648 wd:Q910391 wd:Q911854 wd:Q911922 wd:Q927234 wd:Q93978 wd:Q955332
  }
  SERVICE wikibase:label {
    bd:serviceParam wikibase:language "en".
    ?item rdfs:label ?itemLabel;
      skos:altLabel ?itemAltLabel;
      schema:description ?itemDescription.}
   SERVICE wikibase:label {
    bd:serviceParam wikibase:language "hi".
    ?item skos:altLabel ?hindialtlabel;
      rdfs:label ?hindiLabel;
      schema:description ?hindi.}
  SERVICE wikibase:label {
    bd:serviceParam wikibase:language "ta".
    ?item skos:altLabel ?tamilaltlabel;
      rdfs:label ?tamilLabel;
      schema:description ?tamil.}
  SERVICE wikibase:label {
    bd:serviceParam wikibase:language "es".
  ?item skos:altLabel ?esaltlabel;
    rdfs:label ?esLabel;
    schema:description ?es.}
SERVICE wikibase:label {
  bd:serviceParam wikibase:language "fr".
  ?item skos:altLabel ?fraltlabel;
    rdfs:label ?frLabel;
    schema:description ?fr.}
SERVICE wikibase:label {
  bd:serviceParam wikibase:language "de".
  ?item skos:altLabel ?dealtlabel;
    rdfs:label ?deLabel;
    schema:description ?de.}
SERVICE wikibase:label {
  bd:serviceParam wikibase:language "zh".
  ?item skos:altLabel ?zhaltlabel;
    rdfs:label ?zhLabel;
    schema:description ?zh.}
SERVICE wikibase:label {
  bd:serviceParam wikibase:language "ur".
  ?item skos:altLabel ?uraltlabel;
    rdfs:label ?urLabel;
    schema:description ?ur.}
OPTIONAL { ?wikipedia schema:about ?item; schema:isPartOf <https://en.wikipedia.org/> }
}
```

Figure 9: SPARQL Query

5. The output of the above query generates the following results including a description, a WikiData ID, synonyms, terms, and the Wikipedia URL, as well as descriptions and synonyms in multiple languages. To download the entire result of the SPARQL document on a local machine follow the click on the 'Link' option and then 'SPARQL endpoint'.



```
item                wd:Q1423889
itemLabel           Antiperspirants
itemAltLabel        Antihidrotic
itemDescription     Agents that are put on the skin to reduce sweating or prevent excess sweating (hyperhidrosis)
fraltlabel          antiperspirant, antisudoral

item                wd:Q1696730
itemLabel           Sebostatic
itemAltLabel        sebostasis agent
itemDescription     A substance causing the halt in the production of sebum (sebostasis) by the sebaceous glands.

item                wd:Q1930829
itemLabel           emetic
itemAltLabel        Emetic Agents, Emetic Drugs, emetics
itemDescription     drug that induces vomiting
fraltlabel          Emetique, Gerbant, Vomitif
fr                  médicament destiné à faire vomir
dealtlabel          Antiperistaltika, Antiperistaltikum, Brechmittel, Emetika, Emetogenität, Vomitivum
de                  Erbrechen herbeiführende Substanz

item                wd:Q2142251
itemLabel           irritant
itemAltLabel        Irritants, irritating
itemDescription     substance that acts locally on cutaneous or mucosal surfaces to produce inflammation
fr                  substance qui provoque des irritations
dealtlabel          Reizend
de                  Gefahrstoffe

item                wd:Q2602077
itemLabel           Emmenagogue
itemDescription     herbs which stimulate blood flow in the pelvic area and uterus; some stimulate menstruation
esaltlabel          Emenagoga
fraltlabel          Emmenagogue
de                  Arznei
wikipedia           <https://en.wikipedia.org/wiki/Emmenagogue>
```

Figure 10: SPARQL Output

Further, after end SPARQL point downloaded from the links and saved the file in.xml extension at local machine. Attributes in SPARQL looks like as follows:

```
    <head>
            <variable name='item'/>
            <variable name='itemLabel'/>
            <variable name='itemAltLabel'/>
            <variable name='itemDescription'/>
            <variable name='hindialtlabel'/>
            <variable name='hindi'/>
            <variable name='tamilaltlabel'/>
            <variable name='tamil'/>
            <variable name='esaltlabel'/>
            <variable name='es'/>
            <variable name='fraltlabel'/>
            <variable name='fr'/>
            <variable name='dealtlabel'/>
            <variable name='de'/>
            <variable name='zhaltlabel'/>
            <variable name='zh'/>
            <variable name='uraltlabel'/>
            <variable name='ur'/>
            <variable name='wikipedia'/>
    </head>
```

Figure 11: SPARQL attribute

To retrieve the overall result file below mentioned link provides the SPARQL output into the xml format on GitHub.

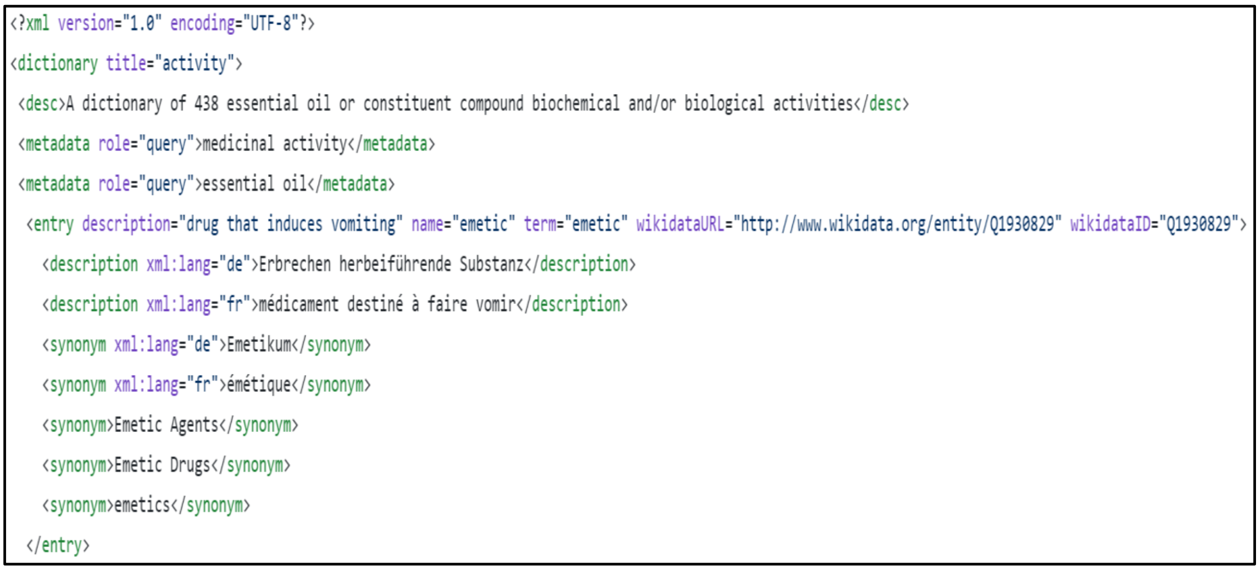
https://github.com/petermr/CEVOpen/blob/master/dictionary/eoActivity/eo_activity/sparql.xml

6. The following amidict command was given in the command prompt to convert the SPARQL endpoint output into the standard xml format of the dictionary.

```
amidict -vv --dictionary Activity --directory Activity --input sparql create --informat
wikisparqlxml –sparqlmap wikidataURL=item, wikipediaPage=wikipedia, name=itemLabel,
term=itemLabel, Description=itemDescription, Hindi=hindiLabel, Hindi_description=hindi,
Hindi_altLabel=hindialtLabel,Tamil=tamilLabel,Tamil_description=tamil,
Tamil_altLabel=tamilaltLabel,Spanish=esLabel,Spanish_description=es,
Spanish_altLabel=esaltLabel,French=frLabel,French_description=fr,
French_altLabel=fraltLabel,Germam=deLabel,German_description=de,
German_altLabel=dealtLabel,Chinese=zhLabel,Chinese_altLabel=zhaltLabel,
Chinese_description=zh, Urdu=urLabel, Urdu_altLabel=uraltLabel, Urdu_description=ur --
transformName wikidataID=EXTRACT(wikidataURL,./(.)) --synonyms=itemAltLabel
```

The output directory will contain a dictionary in xml format. The following is the output from the CEVOpen GitHub repository's activity dictionary:

https://github.com/petermr/CEVOpen/blob/master/dictionary/eoActivity/eo_activity/activity.xml



```
<?xml version="1.0" encoding="UTF-8"?>
<dictionary title="activity">
 <desc>A dictionary of 438 essential oil or constituent compound biochemical and/or biological activities</desc>
 <metadata role="query">medicinal activity</metadata>
 <metadata role="query">essential oil</metadata>
  <entry description="drug that induces vomiting" name="emetic" term="emetic" wikidataURL="http://www.wikidata.org/entity/Q1930829" wikidataID="Q1930829">
    <description xml:lang="de">Erbrechen herbeiführende Substanz</description>
    <description xml:lang="fr">médicament destiné à faire vomir</description>
    <synonym xml:lang="de">Emetikum</synonym>
    <synonym xml:lang="fr">émétique</synonym>
    <synonym>Emetic Agents</synonym>
    <synonym>Emetic Drugs</synonym>
    <synonym>emetics</synonym>
  </entry>
```

Figure 12: Activity dictionaries entity

The activity dictionary provides the following attributes for biological activities, as well as metadata about the different entities as follows:

The **description** parameter defines a human-readable string that describes the entry. It is frequently generated directly from Wikidata and can be used for grouping or disambiguation purposes.

The **name** is the preferred name for the term. It is case-sensitive and frequently appears in the text; the name and term may or may not be synonymous.

The **term** is the entry's one-of-a-kind lexical string (word). Terms are always written in lowercase and begin with a letter. In documents, the term may or may not be the linguistic entity.

The **wikidata ID & URL** is the Wikidata item's identifier. It resolves to the following address: https://wikidata.org/wiki/wikidata>. A Wikidata item has a unique identifier, and the relationships and graphs are language-independent.

The **Wikipedia** page is referred to as Wikipedia. It is frequently used as the term (for single words). It may lack spaces and contain escaped punctuation. It resolves to the following address: https://en.wikipedia.org/wiki/wikipedia>.

Utilize the git commands to commit all data to GitHub at the following location: https://github.com/petermr/CEVOpen/tree/master/dictionary/eoActivity/eo_activity

2.3.3 Test the search tools with DICTIONARY against MINICORPUS

1. The ami section is used to categories scientific articles into the following sections: front, body, back, floats, and groups. Sectioning downloaded files creates a tree structure for us, which aids in navigating the file's content. Sectioning is accomplished by executing the following command at the command prompt:

```
ami -p "activity" section
```

2 Ami search performs a search and analysis of the terms in the project repository, returning the term's frequency data table and the corpus's histogram. The command below states that our purpose to compare our corpus activity with the CEVOpen dictionaries such as activity, country, essential oil plant, and plant compound that gives insights into the phytochemistry and its relevance to medical plants and chemicals.

```
ami -p "activity" search --dictionary activity.xml eoPlant.xml
plant_compound.xml
```

Additionally, we are looking for alternative dictionaries on essential oils and plant compounds to get the possible insights from the open scientific literature regarding the association between the various essential oil plants and compounds with their medicinal activity. These Alternative dictionaries are available at the following link: https://github.com/petermr/CEVOpen/tree/master/dictionary

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Results

We obtain the following results from the aforementioned materials and methods, which enable us to address the scientific question regarding medicinal activity, plant compound essential oil plants, and the countries associated with them.

➢ Results of ami section

Generally, the dataset is sectioned for greater precision. When we open the folder 'sections' in the cProject directory successful completion of the ami section command, the directory's papers are divided into the following sections.

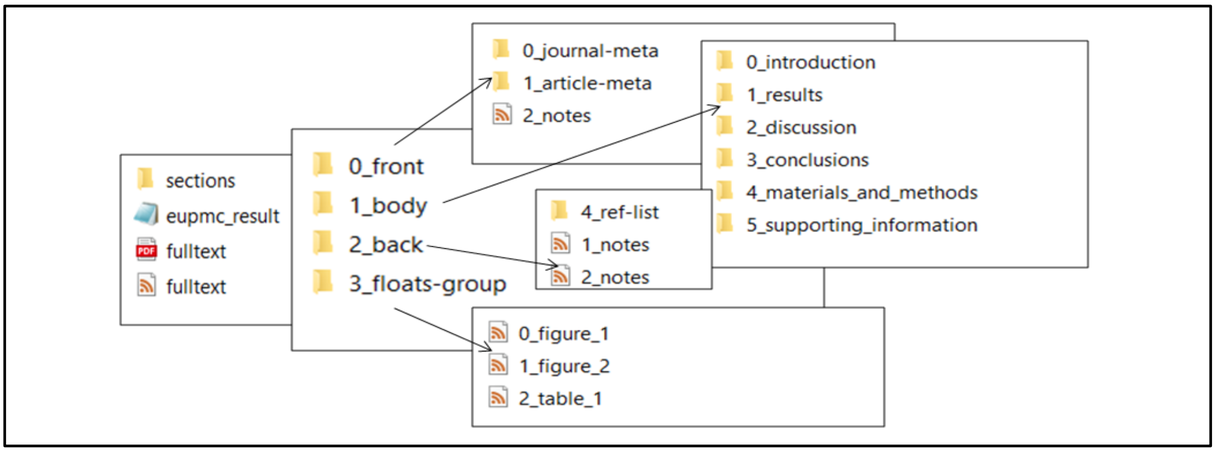


Figure 13: Local machine visualization of ami section result

➢ Results of ami search

ami search returns the following results in the form of a table, a histogram, and results for each folder.

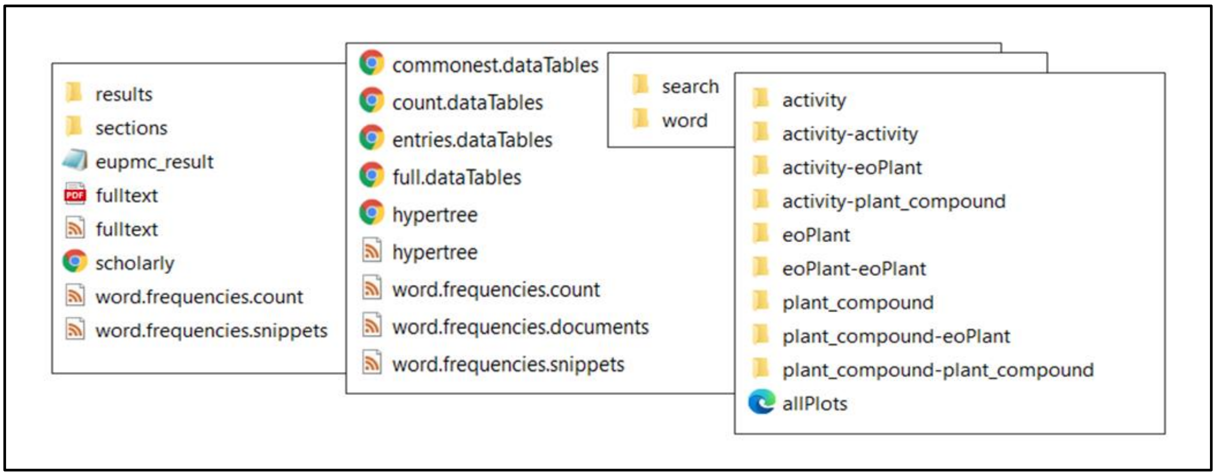
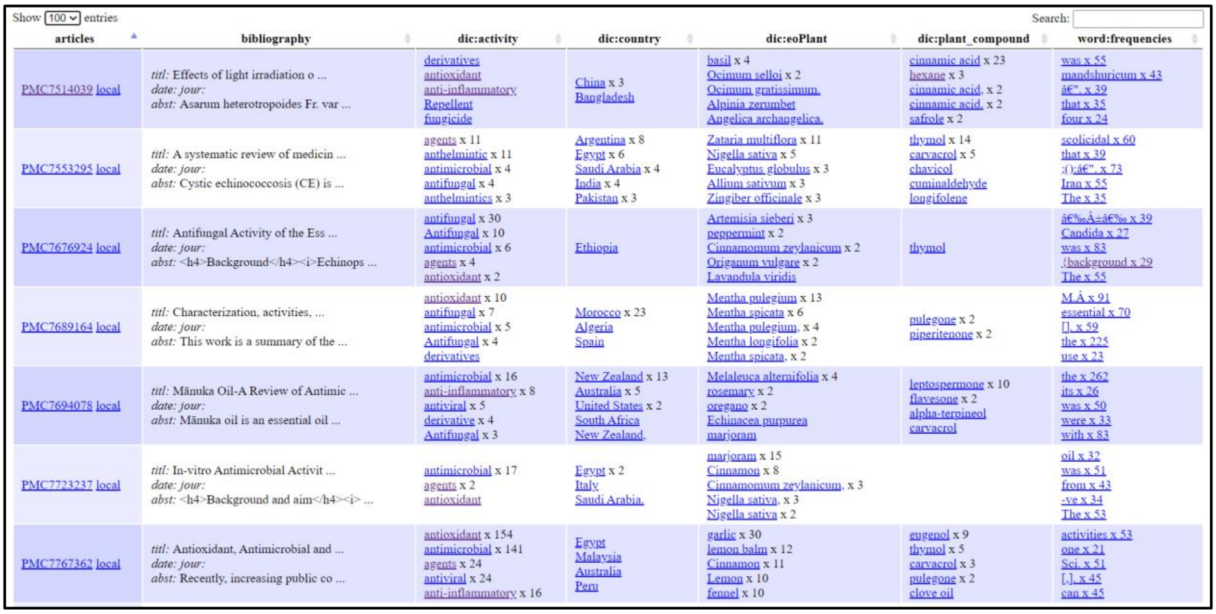


Figure 14: Local machine visualization of ami search result

The most fundamental output is the complete data table. It is a rectangular table with columns representing the searches and rows representing the papers. In a web browser, open full.dataTables.html. It appears as follows:



articles	bibliography	dic:activity	dic:country	dic:eoPlant	dic:plant_compound	word:frequencies
PMC7514039 local	*titl:* Effects of light irradiation o ... *date: jour:* *abst:* Asarum heterotropoides Fr. var ...	derivatives antioxidant anti-inflammatory Repellent fungicide	China x 3 Bangladesh	basil x 4 Ocimum selloi x 2 Ocimum gratissimum. Alpinia zerumbet Angelica archangelica.	cinnamic acid x 23 hexane x 3 cinnamic acid, x 2 cinnamic acid. x 2 safrole x 2	was x 55 mandshuricum x 43 â€". x 39 that x 35 four x 24
PMC7553295 local	*titl:* A systematic review of medicin ... *date: jour:* *abst:* Cystic echinococcosis (CE) is ...	agents x 11 anthelmintic x 11 antimicrobial x 4 antifungal x 4 anthelmintics x 3	Argentina x 8 Egypt x 6 Saudi Arabia x 4 India x 4 Pakistan x 3	Zataria multiflora x 11 Nigella sativa x 5 Eucalyptus globulus x 3 Allium sativum x 3 Zingiber officinale x 3	thymol x 14 carvacrol x 5 chavicol cuminaldehyde longifolene	scolicidal x 60 that x 39 ;():â€". x 73 Iran x 55 The x 35
PMC7676924 local	*titl:* Antifungal Activity of the Ess ... *date: jour:* *abst:* <h4>Background</h4><i>Echinops ...	antifungal x 30 Antifungal x 10 antimicrobial x 6 agents x 4 antioxidant x 2	Ethiopia	Artemisia sieberi x 3 peppermint x 2 Cinnamomum zeylanicum x 2 Origanum vulgare x 2 Lavandula viridis	thymol	â€‰Â±â€‰ x 39 Candida x 27 was x 83 .{background x 29 The x 55
PMC7689164 local	*titl:* Characterization, activities, ... *date: jour:* *abst:* This work is a summary of the ...	antioxidant x 10 antifungal x 7 antimicrobial x 5 Antifungal x 4 derivatives	Morocco x 23 Algeria Spain	Mentha pulegium x 13 Mentha spicata x 6 Mentha pulegium. x 4 Mentha longifolia x 2 Mentha spicata. x 2	pulegone x 2 piperitenone x 2	M.Â x 91 essential x 70 []. x 59 the x 225 use x 23
PMC7694078 local	*titl:* Mānuka Oil-A Review of Antimic ... *date: jour:* *abst:* Mānuka oil is an essential oil ...	antimicrobial x 16 anti-inflammatory x 8 antiviral x 5 derivative x 4 Antifungal x 3	New Zealand x 13 Australia x 5 United States x 2 South Africa New Zealand.	Melaleuca alternifolia x 4 rosemary x 2 oregano x 2 Echinacea purpurea marjoram	leptospermone x 10 flavesone x 2 alpha-terpineol carvacrol	the x 262 its x 26 was x 50 were x 33 with x 83
PMC7723237 local	*titl:* In-vitro Antimicrobial Activit ... *date: jour:* *abst:* <h4>Background and aim</h4><i> ...	antimicrobial x 17 agents x 2 antioxidant	Egypt x 2 Italy Saudi Arabia.	marjoram x 15 Cinnamon x 8 Cinnamomum zeylanicum. x 3 Nigella sativa. x 3 Nigella sativa x 2		oil x 32 was x 51 from x 43 -ve x 34 The x 53
PMC7767362 local	*titl:* Antioxidant, Antimicrobial and ... *date: jour:* *abst:* Recently, increasing public co ...	antioxidant x 154 antimicrobial x 141 agents x 24 antiviral x 24 anti-inflammatory x 16	Egypt Malaysia Australia Peru	garlic x 30 lemon balm x 12 Cinnamon x 11 Lemon x 10 fennel x 10	eugenol x 9 thymol x 5 carvacrol x 3 pulegone x 2 clove oil	activities x 53 one x 21 Sci. x 51 [.]. x 45 can x 45

Figure 15: Full Data table of ami result

- The following link contains the complete data table for the ami search result: https://drive.google.com/file/d/1mNTwHEOjYG17DlJp3pyKyVk9z8bGr_fx/view?usp=sharing
- The following link contains the co-occurrence data for the ami search result: https://drive.google.com/file/d/148P0zZQFD3iTva3SgUyWI9SnJQVXO9EH/view?usp=sharing

The co-occurrence of ami search results provides numerical values associated with each entity in our objective and the graphical relationships between them, which aids us in addressing scientific questions from the open-access literature as followed.

- To determine the most commonly encountered activity associated with essential oil compounds, we create a graphical representation of the data from the ami search co-occurrence result.

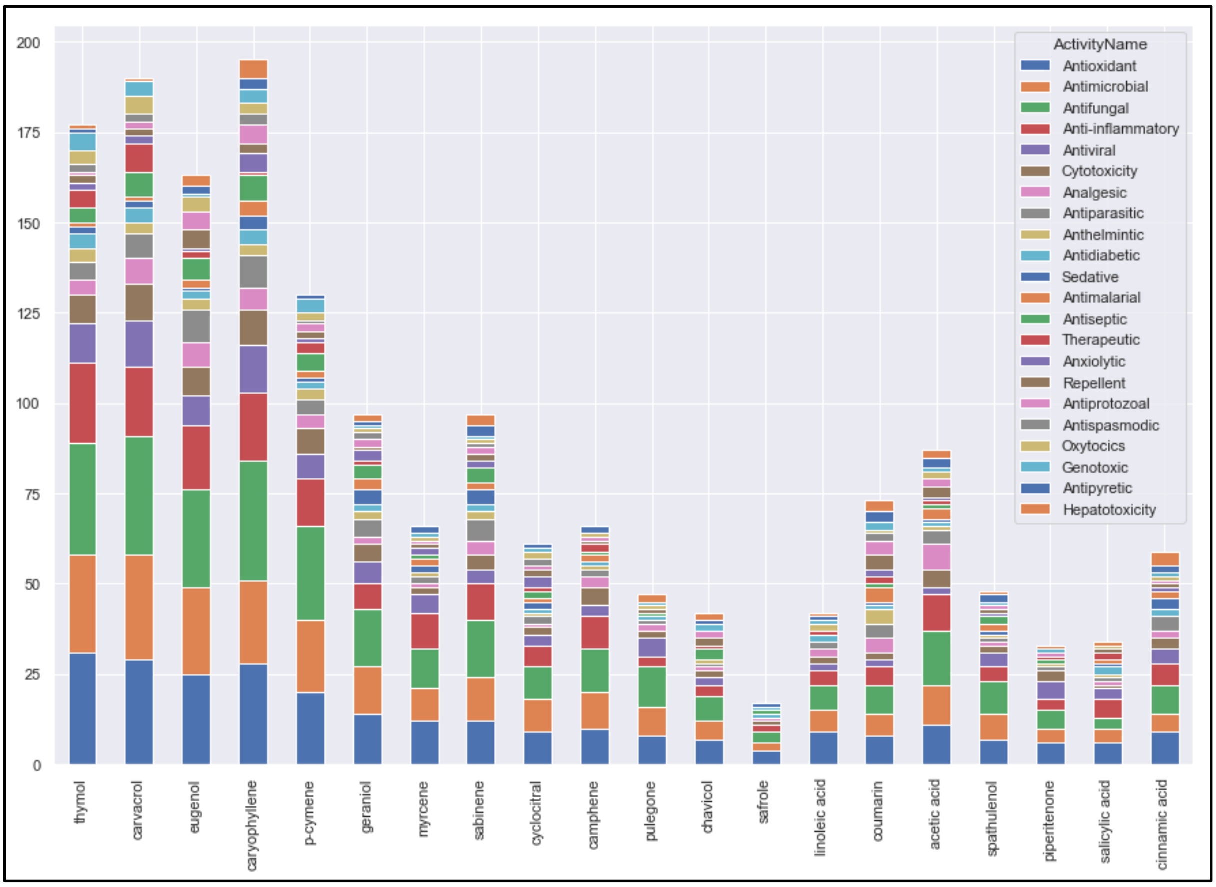


Figure 16: Comparative visualization of medicinal activity with essential oil compounds

According to the graph above, the medicinal activities ‘antioxidant’ and ‘antimicrobial’ are associated with the essential oil compounds ‘thymo’, ‘carvacrol’, and ‘caryophyllene’ in open scientific literature.

- To identifies which essential oil plants are associated with specific activities, we create a graphical representation of the data from the ami search co-occurrence result.

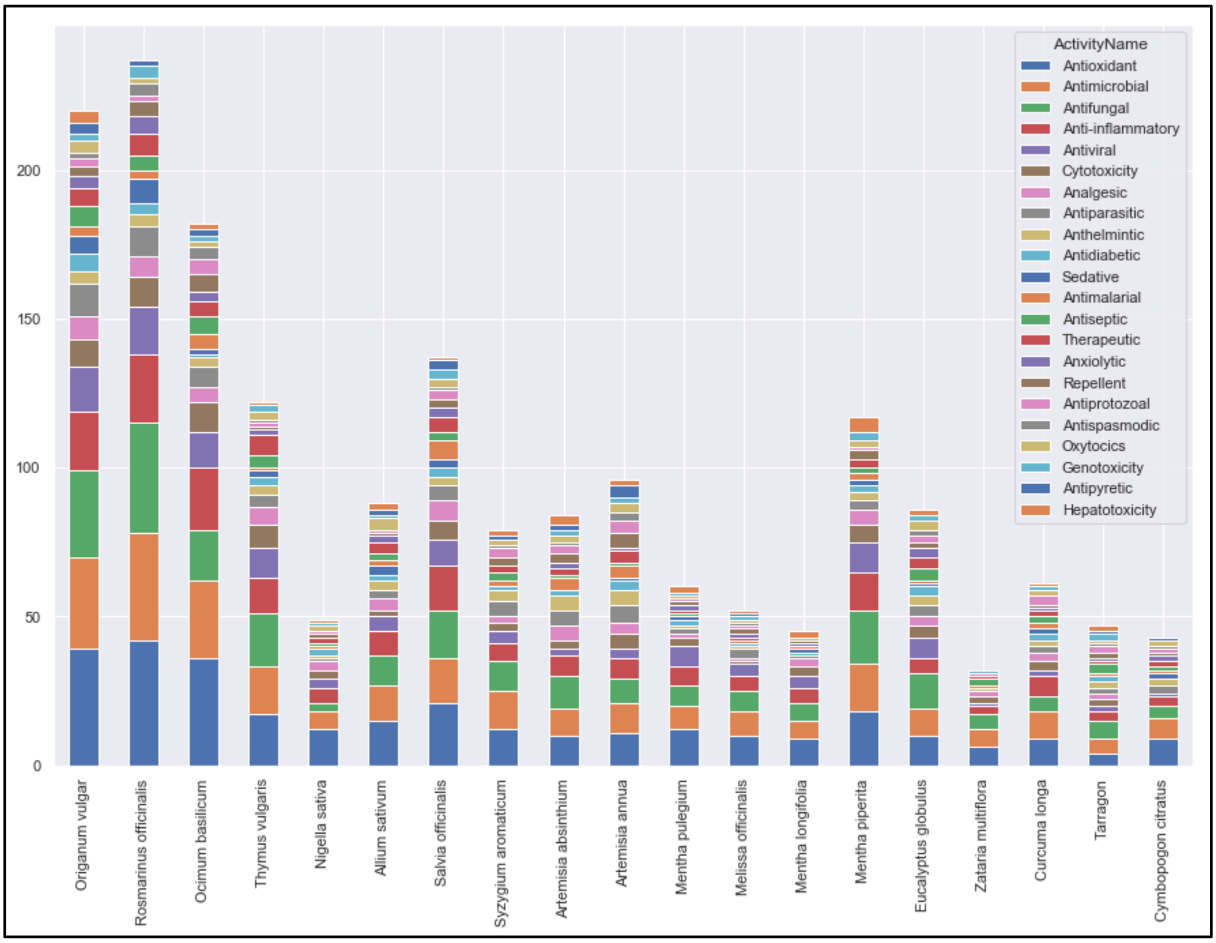


Figure 17: Comparative visualization of medicinal activity with essential oil plants

According to the graph above, the essential oil plants ‘Origanum vulgar’, ‘Rosmarinus officinalis’, ‘Ocimum basilicum’ mostly consist the medicinal activities such as ‘antioxidant’ and ‘antimicrobial’.

- To identifies which essential oil plants are associated with specific essential oil compounds, we create a graphical representation of the data from the ami search co-occurrence result.

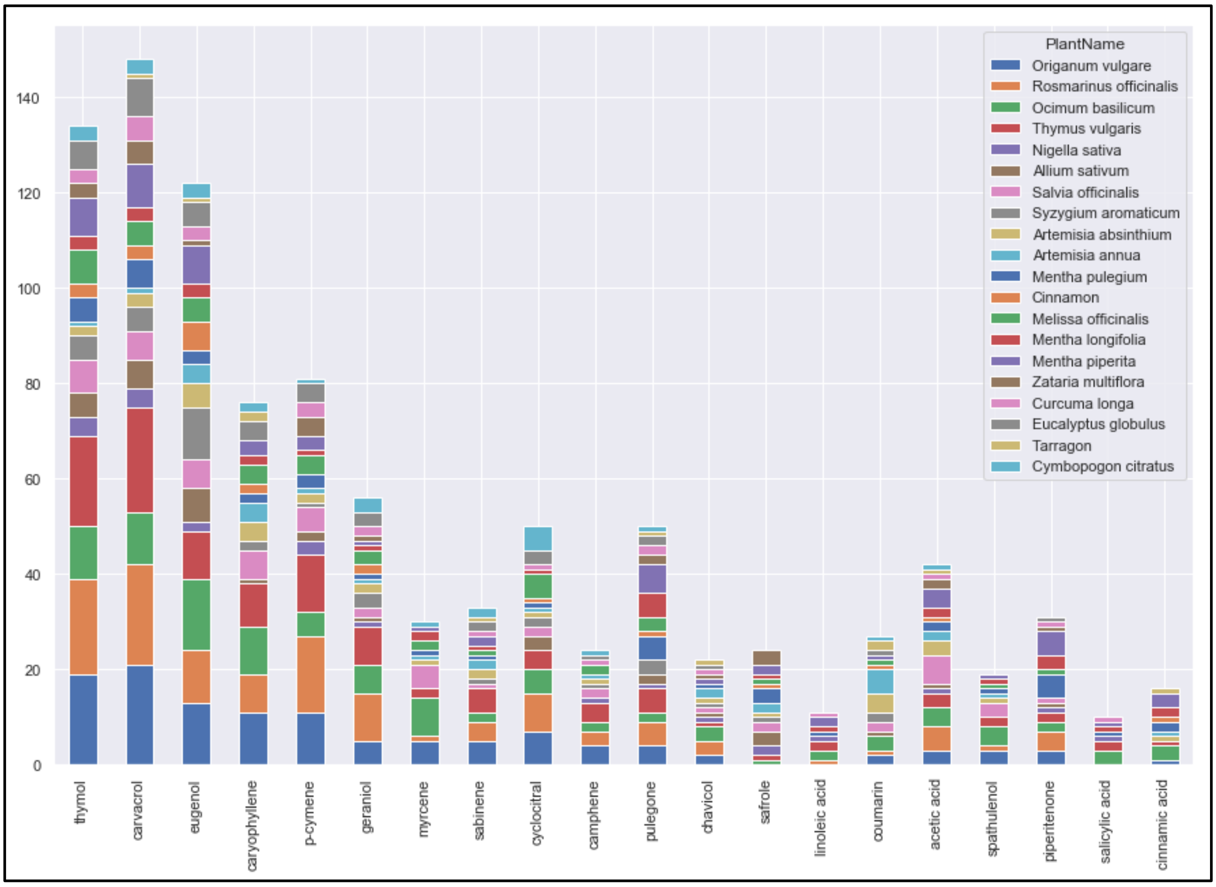


Figure 18: Comparative visualization of essential oil plants with essential oil compounds

According to the graph above, the essential oil plants ‘Origanum vulgar’, ‘Rosmarinus officinalis’, ‘Ocimum basilicum’ mostly associated with the compound of the essential oil ‘thymo’, ‘carvacrol’.

- To find out exactly which plants are the most common and in which countries they are most likely to occur

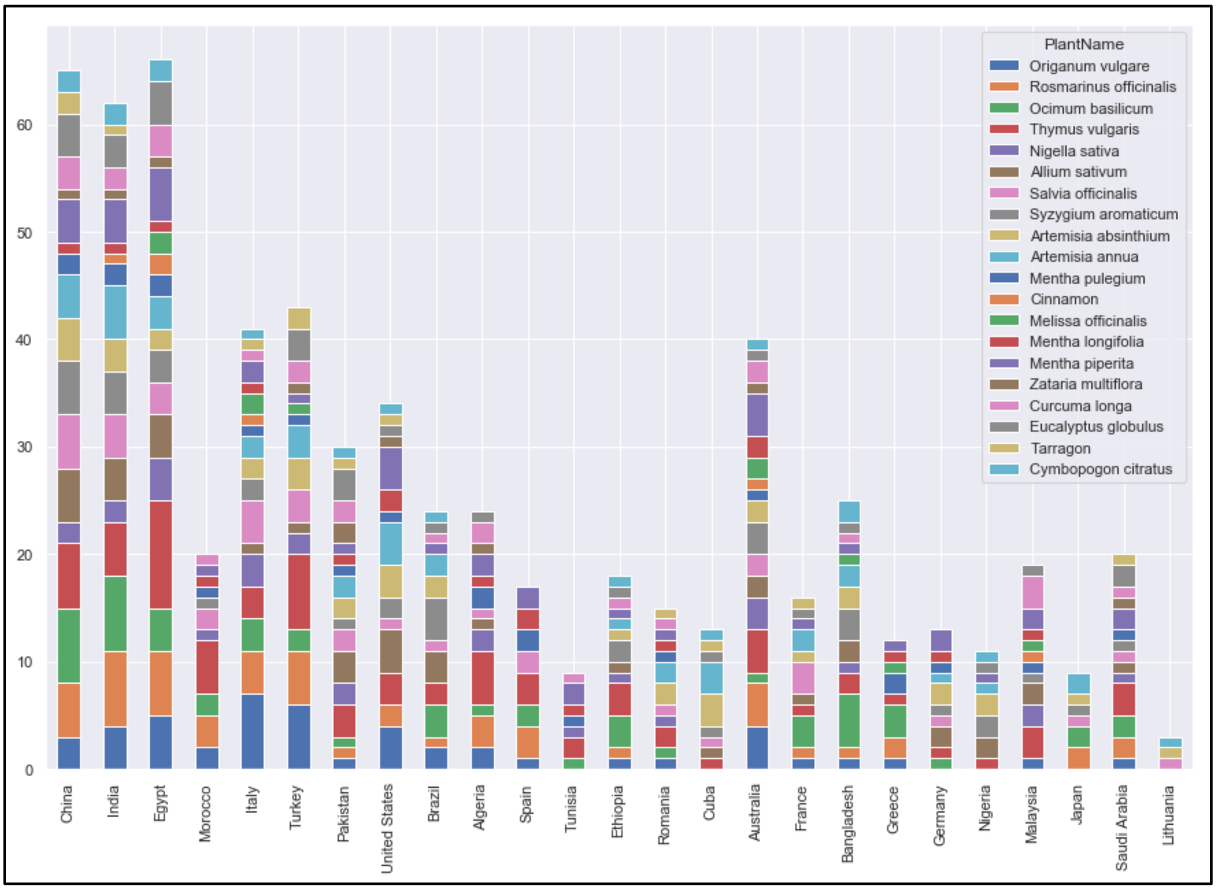


Figure 19: Comparative visualization of essential oil plants with the countries

According to the graph above, the essential oil plants ‘Origanum vulgar’, ‘Rosmarinus officinalis’, ‘Ocimum basilicum’ mostly occur in the countries such as ‘China’, `India` and ‘Turkey’ which mentioned in the open scientific literature.

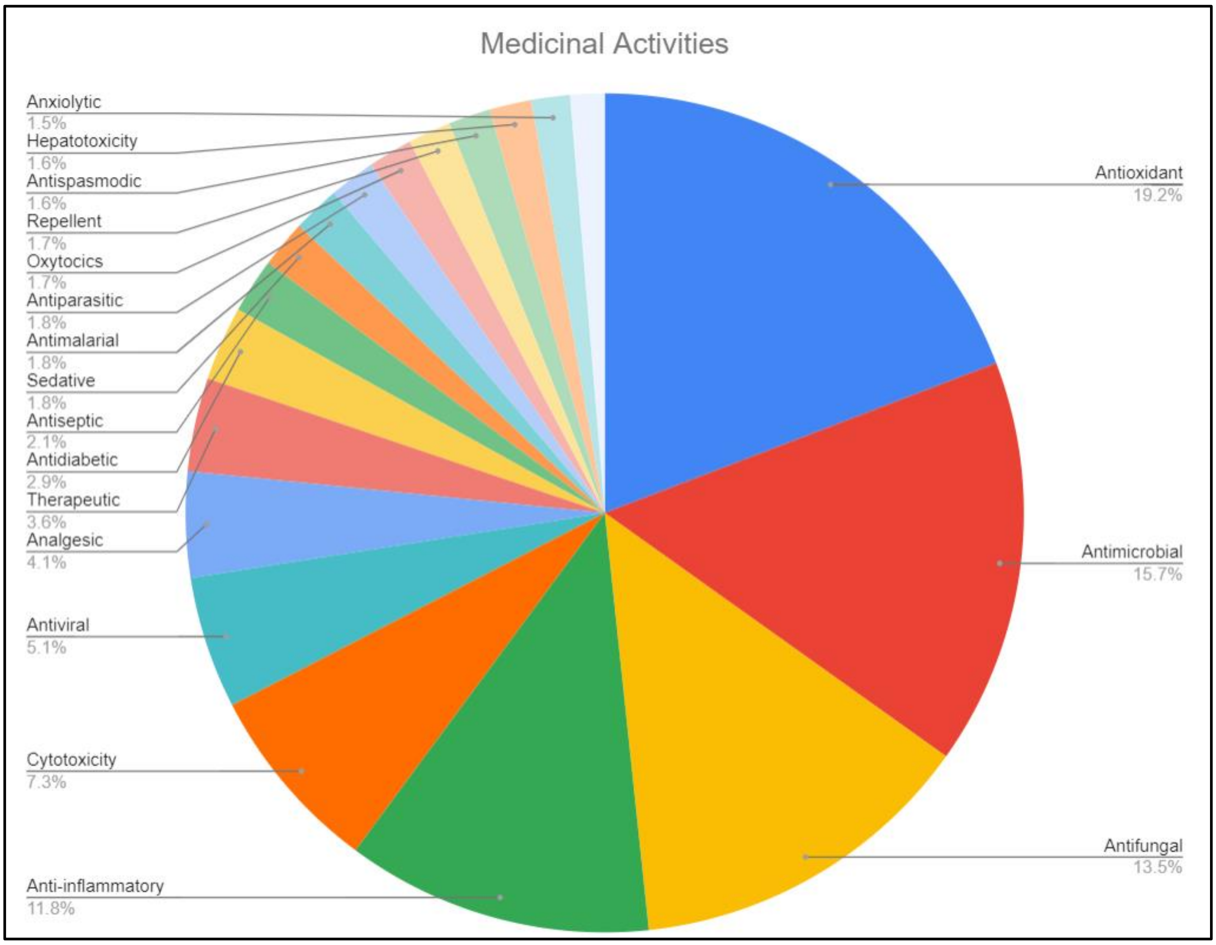


Figure 20: Medicinal Activities observed in the mini corpus of the open scientific literature

This graph gives an insight into the total medicinal activities present in the mini corpus of open access scientific literature. From the above graphs, we state the antioxidant and antimicrobial activity is 19.2 % and 15.7 % respectively while antifungal and anti-inflammatory activity is 13.5 % and 11.8% respectively in the overall mini corpus. Other activities are also observed but it is present in very less amount.

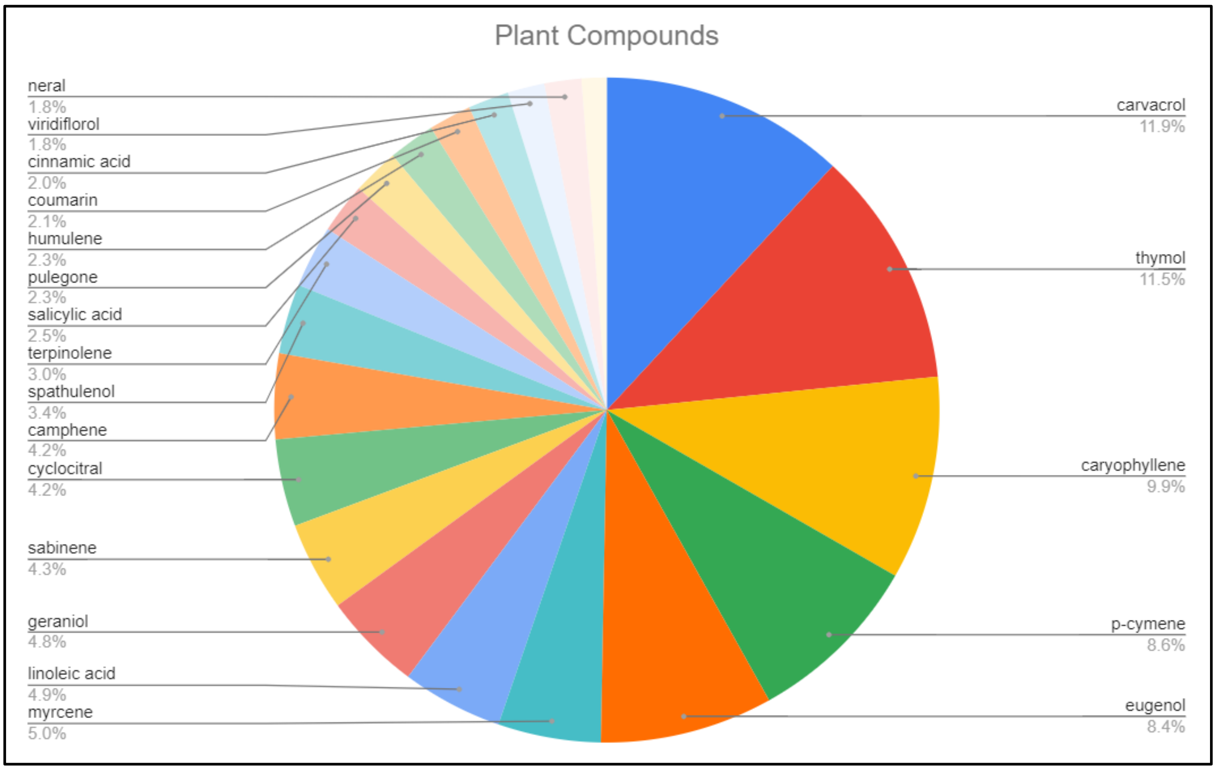


Figure 21: Plant compounds observed in the mini corpus of the open scientific literature

This graph gives an insight into the total essential oil compound present in the mini corpus of open access scientific literature. From the above graphs, we state the carvacrol and thymol are 11.9% and 11.5% respectively while caryophyllene and p-cymene are 9.9% and 8.6 % respectively in the overall mini corpus. Other essential oil compounds are also observed but it is present in very less amount.

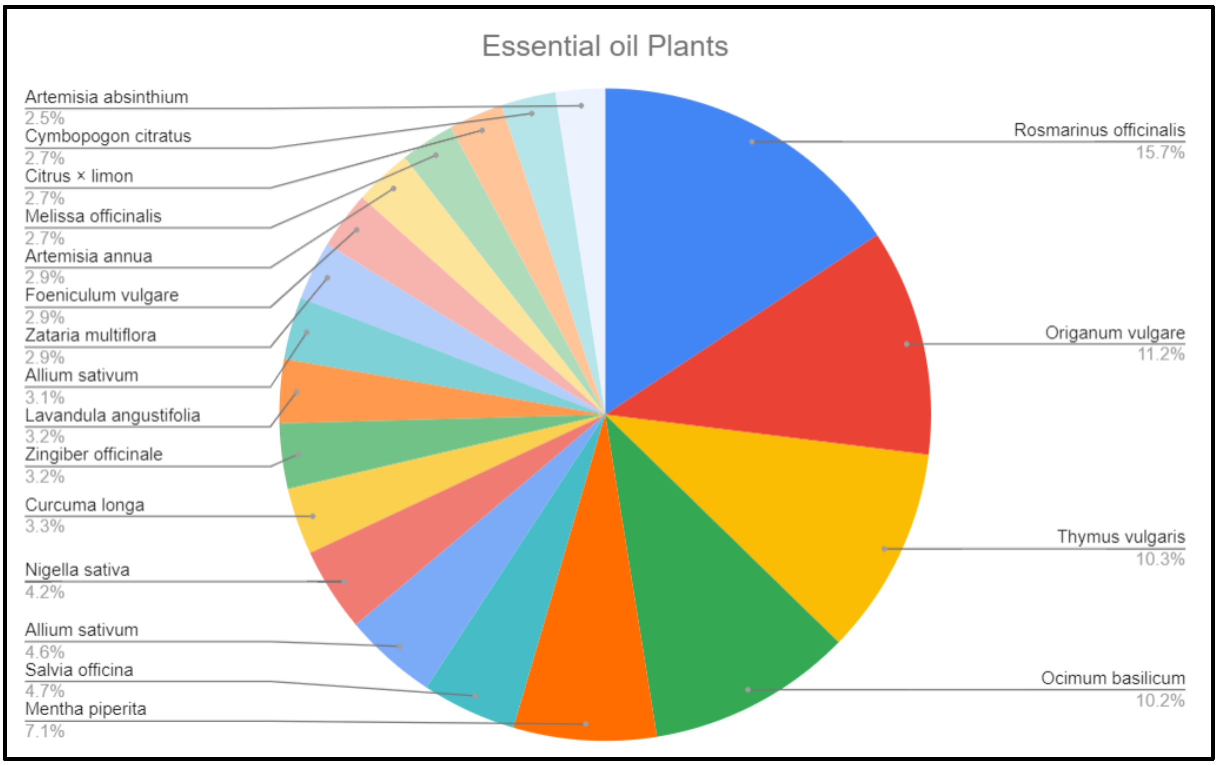


Figure 22: Essential oil plants observed in the mini corpus of the open scientific literature

This graph gives an insight into the total essential oil plants present in the mini corpus of open access scientific literature. From the above graphs, we state the Rosmarinus officinalis and Origanum vulgar is 15.7% and 11.2% respectively while Thymus vulgaris and is Ocimum basilicum 10.3% and 10.2 respectively in the overall mini corpus. Other essential oil plants are also observed but it is present in very less amount.

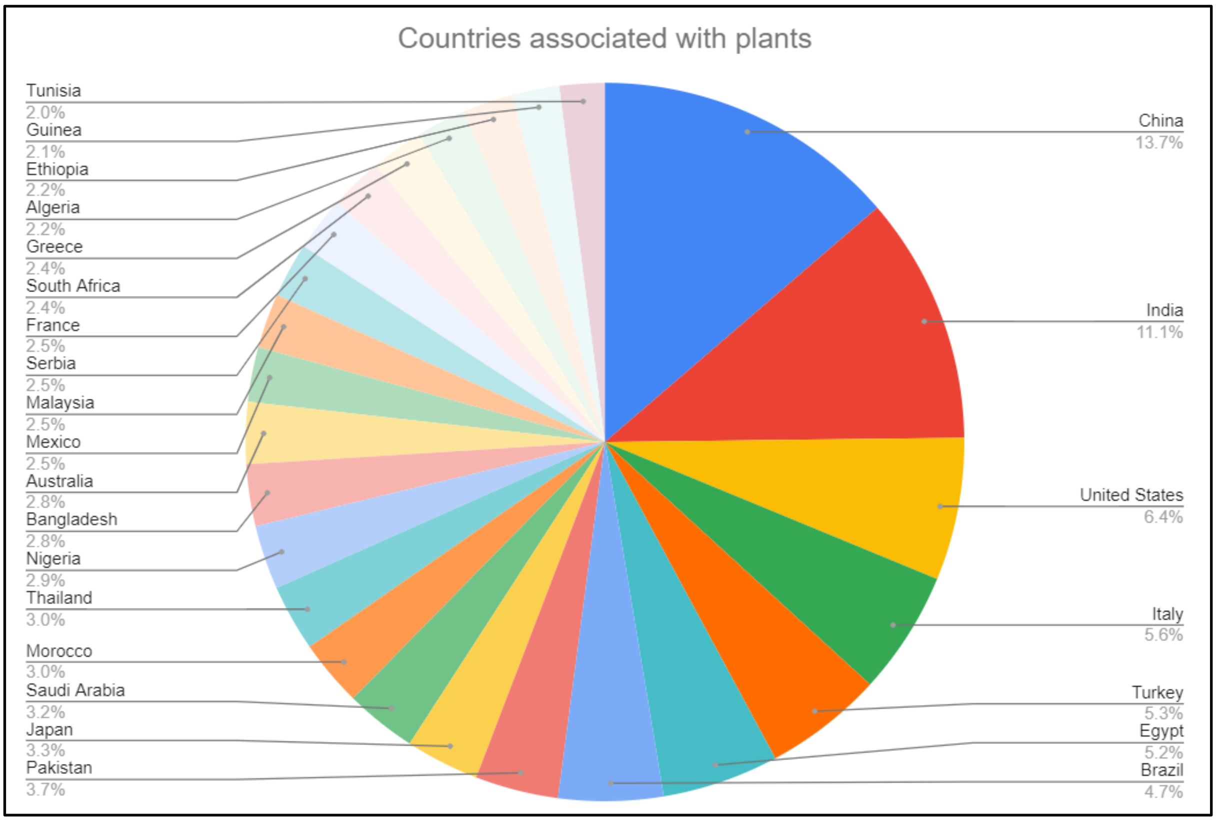


Figure 22: Countries observed in the mini corpus of the open scientific literature

This graph gives an insight into the total countries where essential oil plants occur and this data are present in the mini corpus of open access scientific literature. From the above graphs, we state the China and India are 13.7% and 11.1% respectively while the United States and Italy are 6.4 % and 5.6% respectively in the overall mini corpus. Other countries where essential oil plants occur are also observed but it is present in very less amount.

4.2 Discussion

The results interpreted describes antioxidant, antimicrobial activities of various essential oil compounds like thymol which is responsible to improve digestion and attenuates respiratory problems, carvacrol is found to have the most potent anti-microbial and anti-inflammatory properties. Based on our searches and results the interpretation of different compounds usually results to have many hits several times. Thus it is of utmost importance to know the application range of these compounds. Organum vulgar which is most suitable as a flavoring compound to many dishes is found to have abundant medicinal activity and anti-oxidant activity. Similarly, Rosmarinus officinalis and Occimum bacilicum are also relevant to oregano. It is also suggested that essential oil compounds like carvacrol, thymol which are mentioned in the first graph are associated with the compounds mentioned in the second draft. Thus, it can be discussed that all these essential oil compounds are stronger and have the most significant properties to solve many therapeutics issues to treat human diseases. The major countries where some of these oil compounds occur are China, India, Turkey. However, this information is available in the open scientific literature for future studies.

CHAPTER 5

CONCLUSION

Ontologies for phytochemicals were developed based on the literature, with a particular emphasis on essential oil compounds from plants. This was possible because a dictionary search was created to analyse scientific literature, and to solve this problem, a corpus of medicinal activity of essential oil compounds was created, and all methods were performed and appropriate results were determined with the assistance of getpapers. The project's major findings are based on open source literature on essential oil compounds and their medicinal properties.

CHAPTER 6

REFERENCES

1. Murray-Rust, D. P. (2021.). *https://github.com/petermr/Openvirus*
2. Murray-Rust, D. P. (2021.). *https://github.com/petermr/CEVOpen*
3. Murray-Rust, D. P. (2021). *https://github.com/petermr/openDiagram.*
4. Akthar, Mohd Sayeed, Birhanu Degaga, and Tanweer Azam. 2014. "Antimicrobial Activity of Essential Oils Extracted from Medicinal Plants against the Pathogenic Microorganisms : A Review." *Issues in Biological Sciences and Pharmaceutical Research* 2 (January): 1–7.
5. Jaradat, Nidal. 2021. "Phytochemical Profile and In Vitro Antioxidant, Antimicrobial, Vital Physiological Enzymes Inhibitory and Cytotoxic Effects of Artemisia Jordanica Leaves Essential Oil from Palestine." *Molecules* 26 (9): 2831. https://doi.org/10.3390/molecules26092831.
6. Sakkas, Hercules, and Chrissanthy Papadopoulou. 2017. "Antimicrobial Activity of Basil, Oregano, and Thyme Essential Oils." *Journal of Microbiology and Biotechnology* 27 (3): 429–38. https://doi.org/10.4014/jmb.1608.08024.