Empowering Global Health with AI: Using NLP to Extract Medicinal Plants and Disease-fighting Compounds from PubMed

Rehan Khan*, Preenon Bagchi , Krutanjali Patil

Institute of Biosciences and Technology, MGM University, Chht.

Sambhajinagar, India

khanrehan9395@gmail.com

Abstract. PubMed is a free database maintained by the National Library of Medicine (NLM) at the National Institutes of Health (NIH) in the United States, and it contains more than 30 million citations and abstracts of biomedical literature and other scientific publications related to medicinal plants and phytocompounds from around the world. Natural Language Processing (NLP) and the Natural Language Toolkit (NLTK) is used to extract information on medicinal plants and disease-fighting compounds from PubMed, with the aim of empowering global health research. The methodology involved a Python-based NLP pipeline to extract information on medicinal plants and disease-fighting compounds from PubMed. The pipeline involved several stages, including text pre-processing, named entity recognition (NER), and relationship extraction. Text pre-processing involved cleaning and formatting the abstracts to remove irrelevant information and standardize the text. NER was performed using the libraries to identify chemical compounds, and disease targets. Relationship extraction involved using the NLTK to identify co-occurring terms and analyze their relationships based on their context and proximity. The use of NLP and NLTK can be powerful tools for extracting and analyzing information on medicinal plants and disease-fighting compounds from PubMed. The code developed in this study can be used to automate the extraction of key information from a large number of scientific articles, saving researchers time and effort. The results also showed that this approach can be used to identify relationships between different plants, compounds, and diseases, providing insights that may not be apparent through manual analysis.

Keywords: Natural Language Processing, PubMed, Python, Disease, Artificial Intelligence

1 Introduction

PubMed is a free online database maintained by the National Library of Medicine (NLM) at the U.S. National Institutes of Health (NIH). It contains over 30 million records of biomedical literature, including research articles, reviews, and other scientific publications from around the world. The rapid growth of biomedical literature makes searching for specific articles difficult. As a motivating example, PubMed Central (PMC) is a popular digital repository for biomedical and life science journals and contains more than 7.5 million articles [1]. PubMed is widely used by researchers, healthcare professionals, and the general public to access information on a wide range of topics related to health and medicine. The database is searchable and provides access to

R. Somashekhar et al. (eds.), *Proceedings of the International Conference on Advances in Nano-Neuro-Bio-Quantum (ICAN 2023)*,
Advances in Intelligent Systems Research 69,
https://doi.org/10.2991/978-94-6463-294-1_7

citations, abstracts, and full-text articles for many of the publications indexed. PubMed is considered a valuable resource for staying up-to-date on the latest research in the biomedical field. Author-assigned keywords are often used as a proxy for expert annotations and serve as the reference evaluation for many automatic key phrase extraction benchmark datasets including emails, computer science articles, and news articles [2], [3],[4],[5]. Despite the lack of consistency and standardization across articles, the author-assigned keywords are often correlated with the standardized descriptors assigned by professional indexers [6]. As such, considerable research in automatic key phrase extraction has been done in the general domain towards summarizing articles using author-assigned keywords to express the crucial aspects of the content [7], [8].

Discovering and Summarizing Relationships Between Chemicals, Genes, Proteins, and Diseases in PubChem [9]. This can help researchers find relevant information and insights for drug discovery, disease diagnosis, or treatment. Phytocompounds are chemical substances that are derived from plants and have biological activities. They can be used as natural sources of drugs or as lead compounds for drug development. Phytocompounds are chemical substances that are derived from plants and have biological activities. They can be used as natural sources of drugs or as lead compounds for drug development [10], [11], [12], [13], [14], [15]. Extracting information on medicinal plants and phytocompounds from PubMed is essential because it provides access to a vast repository of scientific literature on their therapeutic properties and potential uses in disease prevention and treatment. Medicinal plants and compounds have been used for centuries to treat various ailments and diseases.

Natural Language Processing (NLP) and the Natural Language Toolkit (NLTK) can be powerful tools in extracting medicinal plant and compound information across languages, which can have a significant impact on global health. Extracting information on medicinal plants and phytocompounds from PubMed using AI techniques can be very useful in the medical field. AI techniques such as natural language processing (NLP) and machine learning can help to automate the process of extracting relevant information from the vast amount of scientific literature available in PubMed. This can save researchers and healthcare professionals time and effort in searching through the database manually and can also help to identify new connections and insights that might not be immediately apparent to a human reader. For example, using NLP algorithms, it is possible to extract key information such as the chemical constituents, pharmacological properties, and clinical applications of medicinal plants and phytocompounds from scientific articles published in PubMed. This information can then be analyzed using machine learning techniques to identify patterns and relationships between different plants and compounds and their potential uses in disease prevention and treatment.

Figure.1 The steps to extract information on medicinal plants and phytocompounds for a particular disease.

1.1 To extract information of medicinal plants and phytocompounds for a particular disease, follow these steps:

[1] Identify the disease: Determine the disease for which you want to extract information on medicinal plants and phytocompounds.

[2] Collect data from PubMed: Collect relevant literature from PubMed using relevant keywords related to the disease of interest. This can be done using the PubMed API or by manually searching PubMed and downloading the relevant literature.

[3] Preprocess the data: Preprocess the collected data to remove irrelevant information and convert the text into a machine-readable format.

[4] Identify medicinal plant and phytocompound names: Use NLP and NLTK techniques such as NER to identify the names of medicinal plants and phytocompounds mentioned in the literature.

[5] Extract information on medicinal plants and phytocompounds: Extract information such as their traditional uses, chemical compositions, and pharmacological properties from the literature using techniques such as text mining and semantic analysis.

[6] Determine their efficacy for the disease of interest: Analyse the extracted information to determine the efficacy of the identified medicinal plants and phytocompounds for the disease of interest. This can be done by looking at clinical studies, animal studies, and in vitro studies.

[7] Evaluate the quality of the extracted information: Evaluate the quality of the extracted information by assessing the reliability of the sources and the methods used to extract the information.

1.2 The use of NLP and NLTK to extract medicinal plants and phytocompounds articles from PubMed can have several important applications in global health. Some of the uses of this technology include:

[1] Disease surveillance: Extracting disease-related articles from PubMed can be used for disease surveillance and monitoring, enabling public health officials to track disease

outbreaks and trends.

[2] Epidemiological research: By extracting relevant information from disease-related articles, researchers can gain insights into the epidemiology of the disease, including risk factors, symptoms, and treatments.

[3] Evidence-based medicine: NLP and NLTK can be used to extract and analyze medical literature, helping healthcare providers to make evidence-based decisions about treatment and care.

[4] Drug discovery: By extracting information on disease-related genes, proteins, and pathways, researchers can use NLP and NLTK to identify potential drug targets and accelerate drug discovery.

[5] Public health interventions: The information extracted from disease-related articles can be used to inform public health interventions and policies, such as vaccination programs or disease prevention campaigns.

Some of the commonly studied medicinal plants and their bioactive compounds include Curcuma longa (turmeric) and its active constituent curcumin, Camellia sinensis (green tea) and its catechins, Allium sativum (garlic) and its organosulfur compounds, and Ginkgo biloba (ginkgo) and its terpenoids. These and other plants and their bioactive compounds have been shown to possess various pharmacological properties that can help to prevent or treat a variety of diseases.

A. Objectives

The objective of this paper is to demonstrate how Natural Language Processing (NLP) and the Natural Language Toolkit (NLTK) can be used in extraction of medicinal plants and phytocompounds information from PubMed containing large number of scientific articles, saving researchers time and effort.

B. Abbreviations

NLM - National Library of Medicine

NIH - National Institutes of Health

PMC - PubMed Central

NLP - Natural Language Processing

NLTK - Natural Language Tool Kit

NER -Name Entity Recognition

API -Application Programming Interface

POS - Part-of-speech

UMLS - Unified Medical Language System

AI -Artificial Intelligence

2. Materials and Methodology

The NLP libraries used are Metapub, SpaCy, PUNKT, BeautifulSoup, Corpus, Pandas along with PubMed database and python codes.

2.1 The methodology for using NLP and NLTK to extract medicinal plants and disease-fighting compounds from PubMed can be summarized in the following steps:

[1] Data Collection: The first step is to collect relevant data from PubMed, a database of biomedical literature. This can be done using the PubMed API, which allows for programmatic access to PubMed records.

[2] Preprocessing: Once the data is collected, it needs to be preprocessed. This involves removing irrelevant information, such as metadata and non-textual data, and converting the remaining text into a machine-readable format.

[3] Tokenization: The next step is to tokenize the text, which involves breaking it down into individual words or phrases.

[4] Part-of-speech (POS) tagging: After tokenization, the text is POS tagged, which involves labeling each token with its corresponding part of speech (e.g., noun, verb, adjective).

[5] Named Entity Recognition (NER): The next step is to perform NER, which involves identifying and labeling named entities in the text, such as medicinal plant names and disease names.

[6] Entity Linking: Once the named entities are identified, the next step is to link them to a knowledge base, such as the Unified Medical Language System (UMLS), which provides a standard vocabulary for biomedical concepts.

[7] Relation Extraction: Finally, the relationships between the identified entities can be extracted using techniques such as dependency parsing and semantic role labeling. This can help identify relationships such as which plants are used to treat which diseases.

2.1 Extracting disease-related articles from PubMed using NLP and NLTK involves the following steps:

1. Selecting the disease(s) of interest: Choose one or more diseases of interest to extract relevant articles from PubMed.
2. Querying PubMed: Use the PubMed API or search interface to query PubMed with relevant search terms, such as the disease name(s), synonyms, and related keywords.
3. Collecting and preprocessing the data: Collect the relevant articles from PubMed and preprocess the data to remove any irrelevant information, such as advertisements or non-research articles.
4. Text cleaning and preprocessing: Use NLP and NLTK techniques to clean and preprocess the text data. This can include tasks such as tokenization, stemming, and stop-word removal.
5. Feature extraction: Extract relevant features from the text data, such as keywords or topics related to the disease(s) of interest.
6. Training a machine learning model: Train a machine learning model, such as a classification algorithm, on the preprocessed and feature-extracted data to identify disease-related articles.

In short it involves a series of steps, starting with identifying the disease of interest and collecting relevant literature from PubMed using relevant keywords. The collected data is then preprocessed to remove irrelevant information and convert the text into a machine-readable format. Using NLP and NLTK techniques, medicinal plant and phytocompound names are identified and information on their traditional uses, chemical compositions, and pharmacological properties are extracted from the literature using techniques such as text mining and semantic analysis. The efficacy of the identified medicinal plants and phytocompounds for the disease of interest is then analyzed, and the quality of the extracted information is evaluated by assessing the reliability of the sources and the methods used to extract the information.

3. Results

1. Obtain an API key from the PubMed API website (https://www.ncbi.nlm.nih.gov/home/develop/api/).

2. Use a programming language such as Python to make API calls and retrieve relevant articles. You can use the Biopython library to interact with the PubMed API.

3. Construct a search query that includes the disease of interest, as well as keywords related to medicinal plants and phytocompounds. For example, the query might include terms such as "diabetes," "phytocompounds"

4. Use the ESearch method to retrieve a list of PubMed IDs that match the search query.

5. Use the EFetch method to retrieve the full text of the articles associated with the PubMed IDs.

6. Parse the article text to extract relevant information on medicinal plants and phytocompounds.

3.1 Program Code

```
!pip install metapub

import pandas as pd

#initialise the keyword to be searched and number of articles to
be retrieved

keyword="medicinal plants and phytocompounds"

num_of_articles=10


from metapub import PubMedFetcher

fetch = PubMedFetcher()


# get the  PMID for first 3 articles with keyword medicinal plants
and phytocompounds

pmids = fetch.pmids_for_query(keyword, retmax=num_of_articles)


# get  articles

articles = {}

for pmid in pmids:

    articles[pmid] = fetch.article_by_pmid(pmid)

# get title for each article:

titles = {}

for pmid in pmids:

    titles[pmid] = fetch.article_by_pmid(pmid).title

Title      =      pd.DataFrame(list(titles.items()),columns      =
['pmid','Title'])

Title


# get abstract for each article:
```

```
abstracts = {}
for pmid in pmids:
    abstracts[pmid] = fetch.article_by_pmid(pmid).abstract
Abstract = pd.DataFrame(list(abstracts.items()),columns =
['pmid','Abstract'])
Abstract


import pandas as pd
#initialise the keyword to be searched and number of articles to
be retrieved


keyword="medicinal plants and phytocompounds"
num_of_articles=10


from metapub import PubMedFetcher
fetch = PubMedFetcher()


# get the  PMID for first 3 articles with keyword medicinal plants
and phytocompounds
pmids = fetch.pmids_for_query(keyword, retmax=num_of_articles)


# get  articles
articles = {}
for pmid in pmids:
    articles[pmid] = fetch.article_by_pmid(pmid)


# get title for each article:
titles = {}
for pmid in pmids:
    titles[pmid] = fetch.article_by_pmid(pmid).title
Title = pd.DataFrame(list(titles.items()),columns =
['pmid','Title'])
Title


# get abstract for each article:
abstracts = {}
for pmid in pmids:
    abstracts[pmid] = fetch.article_by_pmid(pmid).abstract
```

```
Abstract    =    pd.DataFrame(list(abstracts.items()),columns    =
['pmid','Abstract'])

Abstract


# get author for each article:

authors = {}

for pmid in pmids:

    authors[pmid] = fetch.article_by_pmid(pmid).authors

Author     =     pd.DataFrame(list(authors.items()),columns     =
['pmid','Author'])

Author


# get year for each article:

years = {}

for pmid in pmids:

    years[pmid] = fetch.article_by_pmid(pmid).year

Year      =      pd.DataFrame(list(years.items()),columns      =
['pmid','Year'])

Year


# get volume for each article:

volumes = {}

for pmid in pmids:

    volumes[pmid] = fetch.article_by_pmid(pmid).volume

Volume     =     pd.DataFrame(list(volumes.items()),columns     =
['pmid','Volume'])

Volume


# get issue for each article:

issues = {}

for pmid in pmids:

    issues[pmid] = fetch.article_by_pmid(pmid).issue

Issue      =      pd.DataFrame(list(issues.items()),columns      =
['pmid','Issue'])

Issue


# get journal for each article:

journals = {}
```

```
for pmid in pmids:
    journals[pmid] = fetch.article_by_pmid(pmid).journal
Journal = pd.DataFrame(list(journals.items()),columns = ['pmid','Journal'])
Journal


# get citation for each article:
citations = {}
for pmid in pmids:
    citations[pmid] = fetch.article_by_pmid(pmid).citation
Citation = pd.DataFrame(list(citations.items()),columns = ['pmid','Citation'])
Citation


links={}
for pmid in pmids:
    links[pmid] = "https://pubmed.ncbi.nlm.nih.gov/"+pmid+"/"
Link = pd.DataFrame(list(links.items()),columns = ['pmid','Link'])
Link


data_frames = [Title,Abstract,Author,Year,Volume,Issue,Journal,Citation,Link]
from functools import reduce
df_merged = reduce(lambda left,right: pd.merge(left,right,on=['pmid'],
                                              how='outer'), data_frames)
df_merged
```

3.2 Program Code

```
!pip install nltk
!pip install Translator
!pip install translate


import nltk
```

```
from nltk.corpus import treebank_chunk
from nltk.chunk import ne_chunk

import nltk
from nltk.tokenize import word_tokenize
from nltk.tag import pos_tag

!pip install Beautifulsoup4

nltk.download('averaged_perceptron_tagger')
nltk.download('maxent_ne_chunker')

from sklearn.feature_extraction.text import CountVectorizer
from nltk import word_tokenize
from nltk.corpus import stopwords
from string import punctuation
from sklearn.decomposition import PCA
from functools import partial
from wordcloud import WordCloud
from bs4 import BeautifulSoup

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

from urllib.request import urlopen
from bs4 import BeautifulSoup

import nltk
nltk.download('punkt')

import nltk
from nltk.tag.stanford import StanfordNERTagger
```

```
import requests
from bs4 import BeautifulSoup
import nltk
from nltk.tokenize import word_tokenize
from nltk.tag import pos_tag
from nltk.chunk import ne_chunk

# Define search terms for PubMed
search_terms = "medicinal plants and phytocompounds for diabetes"

# Use PubMed API to retrieve articles
url                                                                    =
f"https://eutils.ncbi.nlm.nih.gov/entrez/eutils/esearch.fcgi?db=
pubmed&retmax=20&term={search_terms}&sort=relevance"
response = requests.get(url)
soup = BeautifulSoup(response.text, "xml")
articles = soup.find_all("Id")

# Extract abstracts or full texts of articles
for article in articles:
    article_id = article.text
    abstract_url                                                       =
f"https://eutils.ncbi.nlm.nih.gov/entrez/eutils/efetch.fcgi?db=p
ubmed&id={article_id}&retmode=xml&rettype=abstract"
    response = requests.get(abstract_url)
    soup = BeautifulSoup(response.text, "xml")
    abstract = soup.find("AbstractText").text
    if abstract_tag:
        abstract = abstract_tag.text
    # Identify mentions of medicinal plants and phytocompounds
using NER
    words = word_tokenize(abstract)
    tagged = pos_tag(words)
    ne_chunked = ne_chunk(tagged)
    for chunk in ne_chunked:
        if hasattr(chunk, "label") and chunk.label() == "PERSON":
            plant_name = " ".join(c[0] for c in chunk.leaves())
            #   Extract   information   about   the   plant   using
information extraction tools
```

```
        else:
            print(f"Abstract found for article {article_id}")
```

3.3 Program Code

```
import requests
from bs4 import BeautifulSoup


# Define search terms for PubMed
search_terms = "medicinal plants and phytocompounds for diabetes"


# Use PubMed API to retrieve articles
url                                                              =
f"https://eutils.ncbi.nlm.nih.gov/entrez/eutils/esearch.fcgi?db=
pubmed&retmax=20&term={search_terms}&sort=relevance"
response = requests.get(url)
soup = BeautifulSoup(response.text, "xml")
articles = soup.find_all("Id")


# Extract abstracts or full texts of articles
for article in articles:
    article_id = article.text
    abstract_url                                                 =
f"https://eutils.ncbi.nlm.nih.gov/entrez/eutils/efetch.fcgi?db=p
ubmed&id={article_id}&retmode=xml&rettype=abstract"
    response = requests.get(abstract_url)
    soup = BeautifulSoup(response.text, "xml")
    abstract_tag = soup.find("AbstractText")
    if abstract_tag:
        abstract = abstract_tag.text
        # Extract information about the medicinal plants and
phytocompounds mentioned in the abstract
    else:
        print(f"Abstract found for article {article_id}")
```

3.3 Steps performed

[1] Obtain an API key from the PubMed API website (https://www.ncbi.nlm.nih.gov/home/develop/api/).

[2] Use a programming language such as Python to make API calls and retrieve relevant articles. You can use the Biopython library to interact with the PubMed API.

[3] Construct a search query that includes the disease of interest, as well as keywords related to medicinal plants and phytocompounds. For example, the query might include terms such as "diabetes," "phytocompounds".

[4] Use the ESearch method to retrieve a list of PubMed IDs that match the search query.

[5] Use the EFetch method to retrieve the full text of the articles associated with the PubMed IDs.

[6] Parse the article text to extract relevant information on medicinal plants and phytocompounds.

Links for the same are here: -

https://github.com/RehanKhan-007/NLP/blob/0fecbcdcb49e7d827879f8b045b1e9500b787c4d/Extraction%20of%20articles%20from%20PubMed.ipynb

https://github.com/RehanKhan-007/NLP/blob/0fecbcdcb49e7d827879f8b045b1e9500b787c4d/PubMed_Articles_Extraction.ipynb

The extracted information from PubMed on medicinal plants and phytocompounds has shown promising results for a wide range of diseases, including cancer, cardiovascular diseases, diabetes, and neurological disorders. These natural compounds are believed to have a wide range of biological activities, such as antioxidant, anti-inflammatory, antimicrobial, and anticancer properties, making them potentially valuable resources for the development of new therapeutic agents.

Moreover, this study highlights the potential of AI and NLP techniques to empower global health research and accelerate the development of new treatments and therapies.

4. Discussion

Diseases are one of the major concerns in the medical field, and effective research is crucial in understanding and treating them. PubMed is an important resource for researchers, health practitioners, and the general public interested in natural medicine and alternative therapies. By searching PubMed, one can find studies on the chemical composition of medicinal plants, their biological activities, and their potential applications in various diseases. Additionally, PubMed provides access to clinical trials and systematic reviews that evaluate the safety and efficacy of phytocompounds and plant extracts in human subjects.

Moreover, extracting information on medicinal plants and phytocompounds from PubMed allows for the identification of knowledge gaps and research needs in the field. For example, if a specific plant extract has shown promising results in preclinical studies but has not been tested in clinical trials, PubMed can be used to identify the gaps in the existing literature and highlight the need for further research.

5. Conclusion

In conclusion, is extraction of medicinal plants and phytocompounds information from PubMed essential because it provides access to a wealth of scientific information on the potential uses of natural products in disease prevention and treatment, and it can help identify knowledge gaps and research needs in the field.

Overall, the extraction of medicinal plants and phytocompounds information from PubMed suggests that natural products derived from plants have great potential for the development of new therapies. However, further research is necessary to fully understand their therapeutic effects and to determine their optimal use in the treatment and prevention of various diseases.

References

1. Canese Kathi and Weis Sarah. 2013. Pubmed: the bibliographic database. The NCBI Handbook, 2, 1.
2. Gallina Ygor, Boudin Florian, and Daille Béatrice. 2020. Large-scale evaluation of keyphrase extraction models. In Proceedings of the ACM/IEEE Joint Conference on Digital Libraries in 2020, 271–278.
3. Hulth Anette. 2003. Improved automatic keyword extraction given more linguistic knowledge. In Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing, 216-223.
4. Loza Vanessa, Lahiri Shibamouli, Mihalcea Rada, and Lai Po-Hsiang. 2014. Building a dataset for summarization and keyword extraction from emails. In Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14), 2441–2446.
5. Meng Rui, Zhao Sanqiang, Han Shuguang, He Daqing, Brusilovsky Peter, and Chi Yu. 2017. Deep keyphrase generation. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: LongPapers), 582–592.
6. Isidoro Gil-Leiva and Adolfo Alonso-Arroyo. 2007. Keywords given by authors of scientific articles in database descriptors. Journal of the American Society for Information Science and Technology 58(8) DOI:10.1002/asi.20595.
7. Andrade MA and Valencia A. 1998. Automatic extraction of keywords from scientific text: application to the knowledge domain of protein families. Bioinformatics, 14, 7, 600–607.
8. Waffenschmidt Siw, Knelangen Marco, Sieben Wiebke, Bühn Stefanie, and Pieper Dawid. 2019. Single screening versus conventional double screening for study selection in systematic reviews: a methodological systematic review. BMC Medical Research Methodology, 19, 132.
9. Zaslavsky Leonid, Cheng Tiejun, Gindulyte Asta, He Siqian, Kim Sunghwan, Li Qingliang, Thiessen Paul, Yu Bo, Bolton Evan E. 2021. Discovering and Summarizing Relationships Between Chemicals, Genes, Proteins, and Diseases in PubChem. Frontiers in Research Metrics and Analytics, Vol. 6
10. Wen CC, Chen HM, Yang NS. Developing Phytocompounds from Medicinal Plants as Immunomodulators. Adv Bot Res. 2012;62:197-272. doi: 10.1016/B978-0-12-394591-4.00004-0. Epub 2012 Jun 19. PMID: 32300254; PMCID: PMC7150268.
11. Chen TH, Tsai MJ, Chang CS, Xu L, Fu YS, Weng CF. The exploration of phytocompounds theoretically combats SARS-CoV-2 pandemic against virus entry, viral replication and immune evasion. J Infect Public Health. 2023 Jan;16(1):42-54. doi: 10.1016/j.jiph.2022.11.022. Epub 2022 Nov 19. PMID: 36470006; PMCID: PMC9675089.
12. Luthra R, Roy A. Role of Medicinal Plants against Neurodegenerative Diseases. Curr Pharm Biotechnol. 2022;23(1):123-139. doi: 10.2174/1389201022666210211123539. PMID: 33573549.
13. Cherian A, Vadivel V, Thiruganasambandham S, Madhavankutty S. Phytocompounds and their molecular targets in immunomodulation: a review. J Basic Clin Physiol Pharmacol. 2021 Nov 15. doi: 10.1515/jbcpp-2021-0172. Epub ahead of print. PMID: 34786892.
14. Chowdhury S, Kumar S. Bioactive Phytocompounds: Anti-amyloidogenic Effects Against Hen Egg-White Lysozyme Aggregation. Protein J. 2021 Feb;40(1):78-86. doi: 10.1007/s10930-020-09946-5. Epub 2021 Jan 3. PMID: 33392981.
15. Kar B, Dehury B, Singh MK, Pati S, Bhattacharya D. Identification of phytocompounds as newer antiviral drugs against COVID-19 through molecular docking and simulation based study. J Mol Graph Model. 2022 Jul;114:108192. doi: 10.1016/j.jmgm.2022.108192. Epub 2022 Apr 14. PMID: 35468453; PMCID: PMC9007924.