

```

from Bio import Entrez
import time

Entrez.email = "a.petzold@ucl.ac.uk"

def search_pubmed(query_terms, additional_combinations, specific_terms,
start_date, end_date):
    date_range = f"{start_date}:{end_date}[PDAT]"
    query = " OR ".join(query_terms)
    additional_query = " OR ".join([f"({comb[0]}[AB] AND {comb[1]}[AB])" for
comb in additional_combinations])
    specific_terms_query = " OR ".join([f"\{term}\[AB]" for term in
specific_terms])
    full_query = f"({query}) AND ({additional_query}) AND
({specific_terms_query}) AND {date_range}"
    handle = Entrez.esearch(db="pubmed", term=full_query, retmax=1000)
    record = Entrez.read(handle)
    handle.close()
    return record["IdList"]

def fetch_details(id_list):
    ids = ",".join(id_list)
    handle = Entrez.efetch(db="pubmed", id=ids, rettype="medline",
retmode="text")
    records = handle.read()
    handle.close()

    parsed_records = []
    for record in records.split("\n\n"):
        pmid = title = edat = None
        in_title = False
        title_lines = []
        for line in record.split("\n"):
            if line.startswith("PMID- "):
                pmid = line.replace("PMID- ", "").strip()
            elif line.startswith("TI - "):
                in_title = True
                title_lines.append(line.replace("TI - ", "").strip())
            elif line.startswith(" "):
                if in_title:
                    title_lines.append(line.strip())
            else:
                in_title = False
                if line.startswith("EDAT- "):
                    edat = line.replace("EDAT- ", "").strip()
        if pmid and title_lines and edat:
            title = " ".join(title_lines)
            parsed_records.append({"PMID": pmid, "Title": title, "EDAT": edat})

    return parsed_records

def main():
    query_terms = ["neurofilament", "neurofilaments", "tau protein", "T-tau",
"P-tau", "glial fibrillary acidic protein", "amyloid beta", "ubiquitin C-
terminal hydrolase 1", "neurogranin", "YKL-40"]
    additional_combinations = [("method", "development")]
    specific_terms = ["linearity", "parallelism", "doubling dilution"]
    start_date = "2010/12/09"
    end_date = "2024/07/12"
    id_list = search_pubmed(query_terms, additional_combinations,
specific_terms, start_date, end_date)
    time.sleep(1) # Respect NCBI rate limits
    results = fetch_details(id_list)

```

```
with open("parallelism-literature.txt", "w") as file:
    for result in results:
        file.write(f"PMID: {result['PMID']}\n")
        file.write(f>Title: {result['Title']}\n")
        file.write(f"EDAT: {result['EDAT']}\n")
        file.write("\n")

if __name__ == "__main__":
    main()
```