

---

# **PEMT Documentation**

***Release 0.0.3-dev***

**Yojana Gadiya and Andrea Zaliani**

**Aug 31, 2023**



**CONTENTS:**

<b>1</b>	<b>PEMT: A patent enrichment tool for drug discovery.</b>	<b>3</b>
1.1	Welcome to PEMT's documentation! . . . . .	3
1.2	Command Line Interface . . . . .	5
1.3	Developmental Guide . . . . .	7
<b>2</b>	<b>Indices and tables</b>	<b>9</b>
	<b>Python Module Index</b>	<b>11</b>
	<b>Index</b>	<b>13</b>



**Release notes** : <https://github.com/Fraunhofer-ITMP/PEMT/releases>



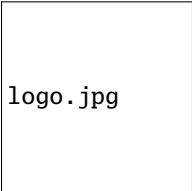
## PEMT: A PATENT ENRICHMENT TOOL FOR DRUG DISCOVERY.

PEMT takes a two-step approach to collect patent documents relevant for drug discovery.

1. The `chemical_extractor` module extraction of chemicals that directly regulate (i.e. activation or inhibition) genes of interest based on functional or biochemical assays found within ChEMBL.
2. The `patent_extractor` module interlinking these chemicals to patent documents by systematically querying SureChEMBL, a patent database.

### 1.1 Welcome to PEMT's documentation!

**Release notes** : <https://github.com/Fraunhofer-ITMP/PEMT/releases>



logo.jpg

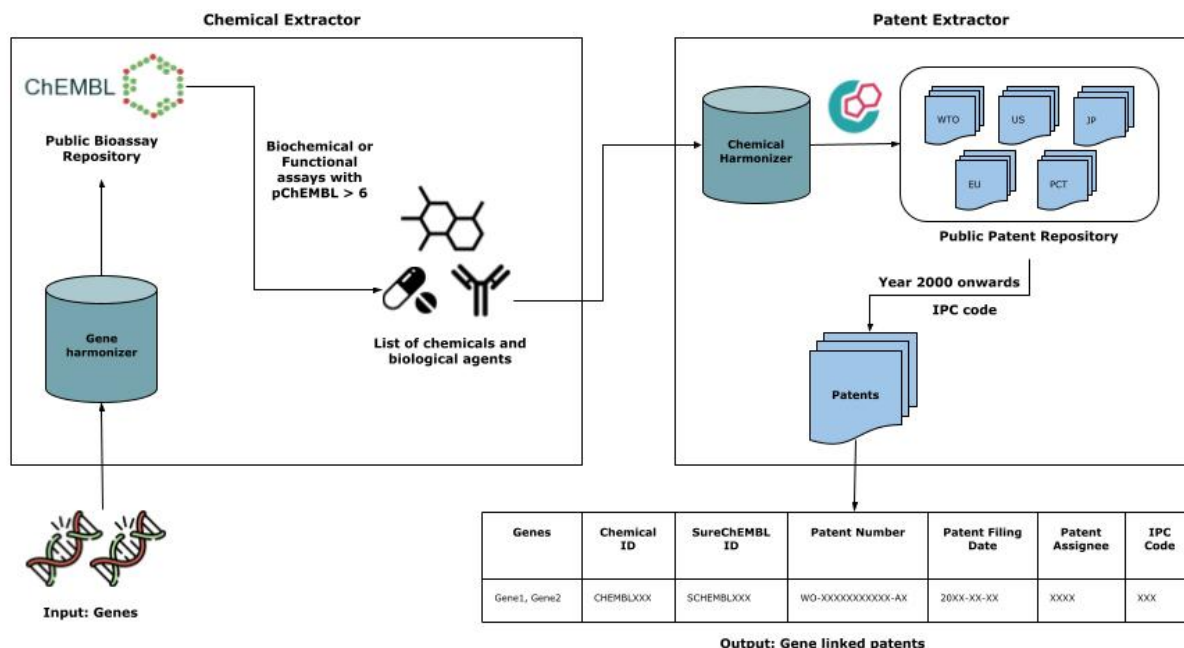
#### 1.1.1 PEMT: A patent enrichment tool for drug discovery.

PEMT takes a two-step approach to collect patent documents relevant for drug discovery.

1. The `chemical_extractor` module extraction of chemicals that directly regulate (i.e. activation or inhibition) genes of interest based on functional or biochemical assays found within ChEMBL.
2. The `patent_extractor` module interlinking these chemicals to patent documents by systematically querying SureChEMBL, a patent database.

#### 1.1.2 General info

PEMT is a patent extractor tool that enables users to retrieve patents relevant to drug discovery. The framework is depicted in the graphic below



### 1.1.3 Installation

You can install PEMT package from pypi.

```
# Use pip to install the latest release
$ python3 -m pip install pemt
```

You may instead want to use the development version from Github, by running

```
$ python3 -m pip install git+https://github.com/Fraunhofer-ITMP/PEMT.git
```

For contributors, the repository can be cloned from [GitHub](https://github.com/Fraunhofer-ITMP/PEMT) and installed in editable mode using:

```
$ git clone https://github.com/Fraunhofer-ITMP/PEMT.git
$ cd PEMT
$ python3 -m pip install -e .
```

### 1.1.4 Dependency

- Python 3.8+
- Installation of chromedriver



## Mandatory

- Pandas
- CheMBL Webresource
- PubChemPy
- Chromedriver

For API information to use this library, see the *Developmental Guide*.

### 1.1.5 Issues

If you have difficulties using PEMT, please open an issue at our [GitHub](#) repository.

### 1.1.6 Disclaimer

PEMT is a scientific tool that has been developed in an academic capacity, and thus comes with no warranty or guarantee of maintenance, support, or back-up of data.

## 1.2 Command Line Interface

PEMT commands.

### 1.2.1 pent

Run PEMT.

```
pent [OPTIONS] COMMAND [ARGS]...
```

#### run-chemical-extractor

Extract chemicals for genes of interest

```
pent run-chemical-extractor [OPTIONS]
```

### Options

**--name** <name>

**Required** Name of the analysis that is to be run

**--data** <data>

**Required** Path to tab-separated gene data file

**--input-type** <input\_type>

Type of data file i.e. 'tab' for tsv or 'comma' for csv files

**--uniprot**, **--no-uniprot**

Boolean value indicating whether the gene data file has uniprot ids or not.

## run-patent-extractor

Extract patent for filtered chemicals

```
pent run-patent-extractor [OPTIONS]
```

### Options

**--name** <name>

**Required** Name of the analysis that is to be run

**--os** <os>

The OS system on which is the script is running

#### Options

linux | mac | windows

**--chromedriver-path** <chromedriver\_path>

**Required** The path where the chromedriver can be found on the users computer

**--year** <year>

The year from which you want to retrieve patents from

**--chemical**, **--no-chemical**

Boolean value indicating whether the chemical data is provided by the user or not

**--chemical-data** <chemical\_data>

Path to tab-separated chemical data file with single column of chembl\_ids

## run-pemt

Run the PEMT tool with gene data

```
pent run-pemt [OPTIONS]
```

### Options

**--name** <name>

**Required** Name of the analysis that is to be run

**--data** <data>

**Required** Path to tab-separated gene data file

**--input-type** <input\_type>

Type of data file i.e. 'tab' for tsv or 'comma' for csv files

**--uniprot**, **--no-uniprot**

Boolean value indicating whether the gene data file has uniprot ids or not.

**--chromedriver-path** <chromedriver\_path>

**Required** The path where the chromedriver can be found on the users computer

**--os** <os>

The OS system on which the script is running

#### Options

linux | mac | windows

**--year** <year>

The year from which you want to retrieve patents from

## 1.3 Developmental Guide

### 1.3.1 Core Module APIs

#### Gene Harmonizer

`pent.utils.get_hgnc_id()` → `Dict[str, str]`

Mapping dictionary for HGNC symbol to HGNC identifiers

`pent.utils.hgnc_to_chembl(chemical_mapper: Dict[str, str], uniprot_mapper: Dict[str, str], hgnc_symbol: str) → str | None`

Mapping HGNC symbol to ChEMBL identifiers.

#### Parameters

- **chemical\_mapper** – A dictionary mapping the UNIPROT identifiers to ChEMBL
- **uniprot\_mapper** – A dictionary mapping the HGNC identifiers to UNIPROT
- **hgnc\_symbol** – A HGNC symbol

`pent.utils.uniprot_to_chembl(chemical_mapper: dict, uniprot_id: str) → str | None`

Mapping UniProt identifiers to ChEMBL identifiers.

#### Parameters

- **chemical\_mapper** – A dictionary mapping the UNIPROT identifiers to ChEMBL
- **uniprot\_id** – UNIPROT identifier of a protein

`pent.utils.get_chemical_names(chembl_id: str) → str`

Method to get chemical name from ChEMBL id.

#### Parameters

**chembl\_id** – ChEMBL identifier of a compound

`pent.utils.uniprot_to_chembl()`

Mapping UniProt identifiers to ChEMBL identifiers.

#### Parameters

- **chemical\_mapper** – A dictionary mapping the UNIPROT identifiers to ChEMBL
- **uniprot\_id** – UNIPROT identifier of a protein

`pent.utils.hgnc_to_chembl()`

Mapping HGNC symbol to ChEMBL identifiers.

#### Parameters

- **chemical\_mapper** – A dictionary mapping the UNIPROT identifiers to ChEMBL

- **uniprot\_mapper** – A dictionary mapping the HGNC identifiers to UNIPROT
- **hgnc\_symbol** – A HGNC symbol

## Chemical Extractor

`pent.chemical_extractor.experimental_data_extraction.extract_chemicals()`

Enrich genes with chemical data from ChEMBL bioassays.

### Parameters

- **analysis\_name** – The name of the analysis you want to run. This name would be used to save the resultant file
- **gene\_list** – The list of gene you want to extract chemicals for.
- **gene\_file\_path** – The path of the gene file
- **file\_separator** – The separator used within the file. This can be ‘comma’, ‘tab’, or ‘semi-colon’. By default,

the file separator is set to csv. :param is\_uniprot: A boolean value indicating whether the given gene list or file containing uniprot ids or HGNC symbols. By default, the value is set to False indicating that a “symbol” column is present with the respective HGNC symbols. If set to True, the file with “uniprot” column is expected.

## Chemical Harmonizer

`pent.patent_extractor.patent_chemical_harmonizer.harmonize_chemicals()`

Method that allows mapping from ChEMBL to SureChEMBL identifiers.

### Parameters

- **analysis\_name** – The name of the analysis you want to run. This name would be used to save the resultant file.
- **from\_genes** – Boolean indicating where the process needs to get chemicals based on genes or not.

## Patent Extractor

`pent.patent_extractor.patent_enrichment.extract_patent()`

Extract and store all valid patent document metadata.

### Parameters

- **analysis\_name** – Name of the analysis.
- **os\_system** – The OS on which the code is running. It can be either of these: linux, mac, window.
- **chrome\_driver\_path** – The path of the chrome driver is located.
- **patent\_year** – The cut-off year for searching the patent documents

## INDICES AND TABLES

- `genindex`
- `modindex`
- `search`



## PYTHON MODULE INDEX

### p

pent, [7](#)

pent.utils, [7](#)





## Symbols

--chemical  
     pemt-run-patent-extractor command line  
         option, 6

--chemical-data  
     pemt-run-patent-extractor command line  
         option, 6

--chromedriver-path  
     pemt-run-patent-extractor command line  
         option, 6

    pemt-run-pemt command line option, 6

--data  
     pemt-run-chemical-extractor command  
         line option, 5

    pemt-run-pemt command line option, 6

--input-type  
     pemt-run-chemical-extractor command  
         line option, 5

    pemt-run-pemt command line option, 6

--name  
     pemt-run-chemical-extractor command  
         line option, 5

    pemt-run-patent-extractor command line  
         option, 6

    pemt-run-pemt command line option, 6

--no-chemical  
     pemt-run-patent-extractor command line  
         option, 6

--no-uniprot  
     pemt-run-chemical-extractor command  
         line option, 5

    pemt-run-pemt command line option, 6

--os  
     pemt-run-patent-extractor command line  
         option, 6

    pemt-run-pemt command line option, 6

--uniprot  
     pemt-run-chemical-extractor command  
         line option, 5

    pemt-run-pemt command line option, 6

--year  
     pemt-run-patent-extractor command line

    option, 6

    pemt-run-pemt command line option, 7

## E

extract\_chemicals() (in module  
     pemt.chemical\_extractor.experimental\_data\_extraction),  
     8

extract\_patent() (in module  
     pemt.patent\_extractor.patent\_enrichment),  
     8

## G

get\_chemical\_names() (in module pemt.utils), 7

get\_hgnc\_id() (in module pemt.utils), 7

## H

harmonize\_chemicals() (in module  
     pemt.patent\_extractor.patent\_chemical\_harmonizer),  
     8

hgnc\_to\_chembl() (in module pemt.utils), 7

## M

module  
     pemt, 7

    pemt.utils, 7

## P

pemt  
     module, 7

pemt.utils  
     module, 7

pemt-run-chemical-extractor command line  
     option  
         --data, 5

        --input-type, 5

        --name, 5

        --no-uniprot, 5

        --uniprot, 5

pemt-run-patent-extractor command line  
     option  
         --chemical, 6

        --chemical-data, 6

- `--chromedriver-path`, 6
- `--name`, 6
- `--no-chemical`, 6
- `--os`, 6
- `--year`, 6

`pent-run-pemt` command line option

- `--chromedriver-path`, 6
- `--data`, 6
- `--input-type`, 6
- `--name`, 6
- `--no-uniprot`, 6
- `--os`, 6
- `--uniprot`, 6
- `--year`, 7

## U

`uniprot_to_chembl()` (*in module `pent.utils`*), 7