# PharmApp Suite

## 🧠 AI for Drug Discovery and Development 🧪

Copyright 2025 | Nghiên Cứu Thuốc | www.nghiencuuthuoc.com | Zalo: +84888999311 |

## 🧩 Topic: Gộp dữ liệu ChEMBL với CID từ PubChem

Tuyệt vời! Tôi rất vui được tham gia dự án phân tích dữ liệu ChEMBL 35 của bạn. Dựa trên thông tin bạn cung cấp, đây là phân tích chi tiết và hướng dẫn song ngữ, kèm theo code mẫu SQL và Python để bạn bắt đầu.

**1. Phân tích và Mô hình Phân tích (Analysis and Analysis Model)**

**1.1. Mục tiêu (Objective):**

Phân tích dữ liệu ChEMBL 35 để xác định các hợp chất tiềm năng có hoạt tính sinh học đối với một mục tiêu (target) cụ thể. Chúng ta sẽ sử dụng RDKit để tính toán các descriptor phân tử và xây dựng mô hình học máy để dự đoán hoạt tính.

The goal is to analyze ChEMBL 35 data to identify potential compounds with biological activity against a specific target. We will use RDKit to calculate molecular descriptors and build a machine learning model to predict activity.

**1.2. Mô hình (Model):**

Chúng ta sẽ sử dụng quy trình sau:

1. **Trích xuất Dữ liệu (Data Extraction):** Sử dụng SQL để truy vấn dữ liệu từ cơ sở dữ liệu ChEMBL 35, tập trung vào một mục tiêu cụ thể và các hoạt tính sinh học liên quan.
2. **Tiền xử lý Dữ liệu (Data Preprocessing):** Làm sạch dữ liệu, xử lý các giá trị thiếu (missing values) và chuyển đổi dữ liệu hoạt tính thành định dạng phù hợp (ví dụ: pIC50).
3. **Tính toán Descriptor (Descriptor Calculation):** Sử dụng RDKit để tính toán các descriptor phân tử từ cấu trúc SMILES của các hợp chất.
4. **Phân tích và Lựa chọn Đặc trưng (Feature Analysis and Selection):** Phân tích các descriptor để xác định những descriptor có liên quan nhất đến hoạt tính sinh học. Sử dụng các phương pháp lựa chọn đặc trưng (feature selection) để giảm số lượng descriptor và cải thiện hiệu suất mô hình.
5. **Xây dựng Mô hình Học Máy (Machine Learning Model Building):** Sử dụng các thuật toán học máy như Random Forest, Support Vector Machines (SVM) hoặc Neural Networks để xây dựng mô hình dự đoán hoạt tính.
6. **Đánh giá Mô hình (Model Evaluation):** Đánh giá hiệu suất của mô hình bằng các chỉ số phù hợp như RMSE, R-squared, AUC.
7. **Ứng dụng (Application):** Sử dụng mô hình để dự đoán hoạt tính của các hợp chất mới.

We will use the following process:

1. **Data Extraction:** Use SQL to query data from the ChEMBL 35 database, focusing on a specific target and related biological activities.
2. **Data Preprocessing:** Clean the data, handle missing values, and convert activity data into a suitable format (e.g., pIC50).
3. **Descriptor Calculation:** Use RDKit to calculate molecular descriptors from the SMILES structures of the compounds.
4. **Feature Analysis and Selection:** Analyze the descriptors to identify those most relevant to biological activity. Use feature selection methods to reduce the number of descriptors and improve model performance.
5. **Machine Learning Model Building:** Use machine learning algorithms such as Random Forest, Support Vector Machines (SVM), or Neural Networks to build a predictive model.
6. **Model Evaluation:** Evaluate the performance of the model using appropriate metrics such as RMSE, R-squared, AUC.
7. **Application:** Use the model to predict the activity of new compounds.

**2. Hướng dẫn Song ngữ (Bilingual Instructions)**

Dưới đây là các bước chi tiết kèm theo code mẫu.

Here are the detailed steps with sample code.

**3. Code SQL và Python (SQL and Python Code)**

**3.1. SQL (Trích xuất Dữ liệu - Data Extraction):**

-- Lấy 100 dòng dữ liệu cho mục tiêu CHEMBL205 (ví dụ)  
-- Get 100 rows of data for target CHEMBL205 (example)  
SELECT  
 md.molregno,  
 cs.canonical\_smiles,  
 act.standard\_value,  
 act.standard\_units,  
 act.standard\_type  
FROM  
 molecule\_dictionary md  
JOIN  
 compound\_structures cs ON md.molregno = cs.molregno  
JOIN  
 activities act ON md.molregno = act.molregno  
JOIN  
 target\_dictionary td ON act.tid = td.tid  
WHERE  
 td.chembl\_id = 'CHEMBL205' -- Thay đổi thành mục tiêu bạn muốn  
 AND act.standard\_type = 'IC50'  
 AND act.standard\_units = 'nM'  
 AND act.standard\_value IS NOT NULL  
LIMIT 100;

**Lưu ý:** Sửa lỗi SQL bạn gặp phải. Hàm ~ trong PostgreSQL không hoạt động với kiểu dữ liệu numeric. Để lọc các giá trị số, bạn có thể sử dụng CASE WHEN hoặc chuyển đổi kiểu dữ liệu (nếu cần). Tuy nhiên, trong câu truy vấn trên, chúng ta đã loại bỏ điều kiện gây ra lỗi.

**Note:** Fix the SQL error you encountered. The ~ function in PostgreSQL does not work with numeric data types. To filter numeric values, you can use CASE WHEN or data type conversion (if needed). However, in the query above, we have removed the condition causing the error.

**3.2. Python (Tiền xử lý, Tính toán Descriptor, Xây dựng Mô hình - Preprocessing, Descriptor Calculation, Model Building):**

import os  
import pandas as pd  
from rdkit import Chem  
from rdkit.Chem import AllChem  
from rdkit.Chem import Descriptors  
import numpy as np  
from sklearn.model\_selection import train\_test\_split  
from sklearn.ensemble import RandomForestRegressor  
from sklearn.metrics import mean\_squared\_error, r2\_score  
  
# Đường dẫn cơ sở - Base path  
base\_path = "." # Thay đổi nếu cần - Change if needed  
data\_path = os.path.join(base\_path, "data")  
notebook\_path = os.path.join(base\_path, "notebooks")  
  
# 1. Đọc dữ liệu từ file CSV - Read data from CSV file  
data = pd.read\_csv(os.path.join(data\_path, "CHEMBL205\_IC50\_100.csv")) # Thay đổi tên file - Change filename  
  
# 2. Tiền xử lý dữ liệu - Data preprocessing  
data = data.dropna(subset=['canonical\_smiles', 'standard\_value']) # Loại bỏ giá trị thiếu - Remove missing values  
data['standard\_value'] = pd.to\_numeric(data['standard\_value'], errors='coerce') # Ép kiểu về số - Convert to numeric  
data = data.dropna(subset=['standard\_value'])  
data = data[data['standard\_value'] > 0] # Loại bỏ giá trị <= 0 - Remove values <= 0  
data['pIC50'] = -np.log10(data['standard\_value'] / 1e9) # Chuyển đổi sang pIC50 - Convert to pIC50  
  
# 3. Tính toán descriptor - Descriptor calculation  
def calculate\_descriptors(smiles):  
 mol = Chem.MolFromSmiles(smiles)  
 if mol is not None:  
 return [Descriptors.MolWt(mol), Descriptors.MolLogP(mol)] # Ví dụ: Khối lượng phân tử và LogP  
 else:  
 return None  
  
data['descriptors'] = data['canonical\_smiles'].apply(calculate\_descriptors)  
data = data.dropna(subset=['descriptors'])  
  
# Tách descriptor thành các cột riêng biệt - Separate descriptors into individual columns  
data['MolWt'] = data['descriptors'].apply(lambda x: x[0])  
data['MolLogP'] = data['descriptors'].apply(lambda x: x[1])  
  
# 4. Chuẩn bị dữ liệu cho mô hình - Prepare data for the model  
X = data[['MolWt', 'MolLogP']] # Chọn descriptor - Select descriptors  
y = data['pIC50'] # Chọn biến mục tiêu - Select target variable  
  
# 5. Chia dữ liệu thành tập huấn luyện và tập kiểm tra - Split data into training and testing sets  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)  
  
# 6. Xây dựng mô hình - Build model  
model = RandomForestRegressor(n\_estimators=100, random\_state=42)  
model.fit(X\_train, y\_train)  
  
# 7. Đánh giá mô hình - Evaluate model  
y\_pred = model.predict(X\_test)  
mse = mean\_squared\_error(y\_test, y\_pred)  
r2 = r2\_score(y\_test, y\_pred)  
  
print(f"Mean Squared Error: {mse}")  
print(f"R-squared: {r2}")

**Lưu ý:**

* Thay đổi tên file CSV trong pd.read\_csv cho phù hợp.
* Bạn có thể thêm nhiều descriptor hơn bằng cách sử dụng các hàm khác từ rdkit.Chem.Descriptors.
* Bạn có thể thử các thuật toán học máy khác.
* Phiên bản scikit-learn cũ có thể không hỗ trợ squared=False trong mean\_squared\_error. Hãy nâng cấp scikit-learn hoặc bỏ tham số này.
* Đảm bảo bạn đã cài đặt các thư viện cần thiết: pip install pandas rdkit scikit-learn.

**Note:**

* Change the CSV filename in pd.read\_csv accordingly.
* You can add more descriptors using other functions from rdkit.Chem.Descriptors.
* You can try other machine learning algorithms.
* Older versions of scikit-learn may not support squared=False in mean\_squared\_error. Upgrade scikit-learn or remove this parameter.
* Make sure you have installed the necessary libraries: pip install pandas rdkit scikit-learn.

**4. Ví dụ Code SQL và Python (SQL and Python Code Examples)**

Dưới đây là 5 ví dụ khác nhau để bạn tham khảo:

Here are 5 different examples for your reference:

**Ví dụ 1: SQL - Lọc theo khoảng giá trị hoạt tính**

-- Lấy các hợp chất có IC50 từ 100 nM đến 1000 nM  
-- Get compounds with IC50 from 100 nM to 1000 nM  
SELECT  
 md.molregno,  
 cs.canonical\_smiles,  
 act.standard\_value  
FROM  
 molecule\_dictionary md  
JOIN  
 compound\_structures cs ON md.molregno = cs.molregno  
JOIN  
 activities act ON md.molregno = act.molregno  
JOIN  
 target\_dictionary td ON act.tid = td.tid  
WHERE  
 td.chembl\_id = 'CHEMBL205'  
 AND act.standard\_type = 'IC50'  
 AND act.standard\_units = 'nM'  
 AND act.standard\_value BETWEEN 100 AND 1000  
LIMIT 100;

**Ví dụ 2: Python - Tính toán nhiều descriptor hơn**

def calculate\_descriptors(smiles):  
 mol = Chem.MolFromSmiles(smiles)  
 if mol is not None:  
 return [Descriptors.MolWt(mol), Descriptors.MolLogP(mol), Descriptors.NumHAcceptors(mol), Descriptors.NumHDonors(mol)]  
 else:  
 return None  
  
data['descriptors'] = data['canonical\_smiles'].apply(calculate\_descriptors)

**Ví dụ 3: SQL - Lấy dữ liệu từ nhiều bảng hơn (ví dụ: protein target)**

-- Lấy dữ liệu kết hợp với thông tin protein target  
-- Get data combined with protein target information  
SELECT  
 md.molregno,  
 cs.canonical\_smiles,  
 act.standard\_value,  
 td.pref\_name AS target\_name  
FROM  
 molecule\_dictionary md  
JOIN  
 compound\_structures cs ON md.molregno = cs.molregno  
JOIN  
 activities act ON md.molregno = act.molregno  
JOIN  
 target\_dictionary td ON act.tid = td.tid  
WHERE  
 td.target\_type = 'PROTEIN'  
 AND act.standard\_type = 'IC50'  
 AND act.standard\_units = 'nM'  
LIMIT 100;

**Ví dụ 4: Python - Sử dụng SVM để xây dựng mô hình**

from sklearn.svm import SVR  
  
# Xây dựng mô hình SVM - Build SVM model  
model = SVR(kernel='rbf') # Hoặc 'linear', 'poly', 'sigmoid'  
model.fit(X\_train, y\_train)

**Ví dụ 5: SQL - Tìm kiếm các hợp chất tương tự dựa trên SMILES**

-- Tìm kiếm các hợp chất có cấu trúc tương tự với một hợp chất cụ thể (ví dụ: CHEMBL121)  
-- Search for compounds with similar structures to a specific compound (e.g., CHEMBL121)  
SELECT  
 md.molregno,  
 cs.canonical\_smiles  
FROM  
 molecule\_dictionary md  
JOIN  
 compound\_structures cs ON md.molregno = cs.molregno  
WHERE  
 cs.canonical\_smiles LIKE '%[Cl]c1ccccc1%' -- Ví dụ: Tìm các hợp chất có vòng benzen chứa clo  
LIMIT 100;

Hy vọng những hướng dẫn và ví dụ này sẽ giúp bạn bắt đầu dự án của mình một cách suôn sẻ. Chúc bạn thành công! Nếu bạn có bất kỳ câu hỏi nào khác, đừng ngần ngại hỏi.