# PharmApp Suite

## 🧠 AI for Drug Discovery and Development 🧪

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

## 🧩 Topic: Các target virus/vi khuẩn phổ biến nhất

Tuyệt vời! Tôi rất vui được hỗ trợ bạn trong dự án phân tích dữ liệu Chembl 35 kết hợp RDKit. Dưới đây là phân tích chi tiết, hướng dẫn song ngữ, code SQL và Python mẫu, cùng với các ví dụ và giải quyết các lỗi bạn đã đề cập.

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

Mục tiêu của bạn là sử dụng dữ liệu Chembl 35 để hỗ trợ nghiên cứu và phát triển thuốc. Dưới đây là mô hình phân tích đề xuất:

* **Data Extraction (Trích xuất dữ liệu):** Sử dụng SQL để truy vấn và trích xuất dữ liệu từ cơ sở dữ liệu Chembl 35. Dữ liệu này bao gồm thông tin về các hợp chất, hoạt tính sinh học của chúng (ví dụ: IC50, Ki), và các thuộc tính khác.
* **Data Cleaning and Preprocessing (Làm sạch và tiền xử lý dữ liệu):** Loại bỏ các giá trị bị thiếu, xử lý các đơn vị không nhất quán, và chuyển đổi dữ liệu về định dạng phù hợp.
* **Feature Engineering (Xây dựng đặc trưng):** Sử dụng RDKit để tính toán các đặc trưng hóa học của các hợp chất. Các đặc trưng này có thể bao gồm:
  + **Molecular descriptors (Mô tả phân tử):** Khối lượng phân tử, logP, số lượng liên kết hydro, v.v.
  + **Fingerprints (Dấu vân tay):** Mã hóa cấu trúc phân tử thành một vector bit.
* **Data Analysis and Modeling (Phân tích dữ liệu và mô hình hóa):**
  + **Exploratory Data Analysis (EDA) (Phân tích khám phá dữ liệu):** Tìm hiểu dữ liệu bằng cách sử dụng các biểu đồ và thống kê mô tả.
  + **Machine Learning (Học máy):** Xây dựng các mô hình dự đoán hoạt tính sinh học của các hợp chất dựa trên các đặc trưng hóa học. Các mô hình phổ biến bao gồm:
    - Linear Regression (Hồi quy tuyến tính)
    - Random Forest (Rừng ngẫu nhiên)
    - Support Vector Machines (SVM) (Máy vector hỗ trợ)
* **Model Evaluation and Validation (Đánh giá và xác thực mô hình):** Đánh giá hiệu suất của mô hình bằng cách sử dụng các bộ dữ liệu kiểm tra và các chỉ số đánh giá phù hợp (ví dụ: R-squared, RMSE, AUC).

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

**English:**

This project aims to leverage Chembl 35 data for drug discovery and development. We will extract data using SQL, preprocess it, engineer features using RDKit, and build machine learning models to predict biological activity.

**Tiếng Việt:**

Dự án này nhằm mục đích tận dụng dữ liệu Chembl 35 cho việc khám phá và phát triển thuốc. Chúng ta sẽ trích xuất dữ liệu bằng SQL, tiền xử lý nó, xây dựng các đặc trưng bằng RDKit và xây dựng các mô hình học máy để dự đoán hoạt tính sinh học.

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

**SQL Code (English):**

-- SQL query to extract data from Chembl 35 (limited to 100 rows)  
-- Retrieve compound information, activities, and target information  
SELECT  
 cmp.chembl\_id,  
 cmp.pref\_name,  
 act.standard\_type,  
 act.standard\_value,  
 act.standard\_units,  
 act.pchembl\_value,  
 tgt.target\_type,  
 tgt.pref\_name AS target\_name  
FROM  
 compound\_structures cmp  
JOIN  
 activities act ON cmp.molregno = act.molregno  
JOIN  
 target\_dictionary tgt ON act.tid = tgt.tid  
WHERE act.standard\_type = 'IC50'  
 AND act.standard\_units = 'nM'  
 AND act.pchembl\_value IS NOT NULL  
LIMIT 100;  
  
-- fixing ERROR: operator does not exist: numeric ~ unknown  
-- make sure standard\_value is numeric  
SELECT  
 cmp.chembl\_id,  
 cmp.pref\_name,  
 act.standard\_type,  
 act.standard\_value,  
 act.standard\_units,  
 act.pchembl\_value,  
 tgt.target\_type,  
 tgt.pref\_name AS target\_name  
FROM  
 compound\_structures cmp  
JOIN  
 activities act ON cmp.molregno = act.molregno  
JOIN  
 target\_dictionary tgt ON act.tid = tgt.tid  
WHERE act.standard\_type = 'IC50'  
 AND act.standard\_units = 'nM'  
 AND act.pchembl\_value IS NOT NULL  
 AND act.standard\_value::text ~ '^[0-9\.]+$' -- Ensure standard\_value is numeric  
LIMIT 100;  
  
  
-- Save the result as a CSV file (e.g., data.csv)

**SQL Code (Tiếng Việt):**

-- Truy vấn SQL để trích xuất dữ liệu từ Chembl 35 (giới hạn 100 dòng)  
-- Lấy thông tin hợp chất, hoạt tính và thông tin mục tiêu  
SELECT  
 cmp.chembl\_id,  
 cmp.pref\_name,  
 act.standard\_type,  
 act.standard\_value,  
 act.standard\_units,  
 act.pchembl\_value,  
 tgt.target\_type,  
 tgt.pref\_name AS target\_name  
FROM  
 compound\_structures cmp  
JOIN  
 activities act ON cmp.molregno = act.molregno  
JOIN  
 target\_dictionary tgt ON act.tid = tgt.tid  
WHERE act.standard\_type = 'IC50'  
 AND act.standard\_units = 'nM'  
 AND act.pchembl\_value IS NOT NULL  
LIMIT 100;  
  
-- sửa lỗi ERROR: operator does not exist: numeric ~ unknown  
-- đảm bảo standard\_value là số  
SELECT  
 cmp.chembl\_id,  
 cmp.pref\_name,  
 act.standard\_type,  
 act.standard\_value,  
 act.standard\_units,  
 act.pchembl\_value,  
 tgt.target\_type,  
 tgt.pref\_name AS target\_name  
FROM  
 compound\_structures cmp  
JOIN  
 activities act ON cmp.molregno = act.molregno  
JOIN  
 target\_dictionary tgt ON act.tid = tgt.tid  
WHERE act.standard\_type = 'IC50'  
 AND act.standard\_units = 'nM'  
 AND act.pchembl\_value IS NOT NULL  
 AND act.standard\_value::text ~ '^[0-9\.]+$' -- Đảm bảo standard\_value là số  
LIMIT 100;  
  
-- Lưu kết quả vào một file CSV (ví dụ: data.csv)

**Python Code (English):**

import pandas as pd  
import os  
from rdkit import Chem  
from rdkit.Chem import Descriptors  
import numpy as np  
from sklearn.model\_selection import train\_test\_split  
from sklearn.linear\_model import LinearRegression  
from sklearn.metrics import mean\_squared\_error, r2\_score  
from sklearn.preprocessing import StandardScaler  
  
# Define base path  
base\_path = "../data" # Adjust if needed  
  
# Load data from CSV  
data\_file = os.path.join(base\_path, "data.csv")  
df = pd.read\_csv(data\_file)  
  
# Data Cleaning and Preprocessing  
df.dropna(subset=['standard\_value', 'pchembl\_value'], inplace=True)  
df['standard\_value'] = pd.to\_numeric(df['standard\_value'], errors='coerce')  
df.dropna(subset=['standard\_value'], inplace=True)  
  
# Function to calculate molecular descriptors using RDKit  
def calculate\_descriptors(smiles):  
 try:  
 mol = Chem.MolFromSmiles(smiles)  
 if mol is None:  
 return None  
 descriptors = {desc[0]: desc[1](mol) for desc in Descriptors.descList}  
 return descriptors  
 except:  
 return None  
  
# Apply descriptor calculation to each compound  
# Assuming you have a column named 'smiles' in your dataframe  
# Example: create a smiles column (replace chembl\_id with a proper smiles source)  
smiles\_dict = {'CHEMBL206': 'c1ccccc1', 'CHEMBL465': 'C1CCCCC1'}  
df['smiles'] = df['chembl\_id'].map(smiles\_dict)  
df.dropna(subset=['smiles'], inplace=True)  
  
df['descriptors'] = df['smiles'].apply(calculate\_descriptors)  
df.dropna(subset=['descriptors'], inplace=True)  
  
# Convert descriptors to DataFrame  
descriptors\_df = pd.DataFrame(df['descriptors'].tolist())  
df = pd.concat([df, descriptors\_df], axis=1)  
df.dropna(inplace=True)  
  
# Prepare data for machine learning  
X = df[[col for col in df.columns if col not in ['chembl\_id', 'pref\_name', 'standard\_type', 'standard\_value', 'standard\_units', 'target\_type', 'target\_name', 'smiles', 'descriptors', 'pchembl\_value']]] # Feature selection  
y = df['pchembl\_value'] # Target variable  
  
# Data scaling  
scaler = StandardScaler()  
X\_scaled = scaler.fit\_transform(X)  
  
# Split data into training and testing sets  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(X\_scaled, y, test\_size=0.2, random\_state=42)  
  
# Train a linear regression model  
model = LinearRegression()  
model.fit(X\_train, y\_train)  
  
# Make predictions  
y\_pred = model.predict(X\_test)  
  
# Evaluate the model  
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}")

**Python Code (Tiếng Việt):**

import pandas as pd  
import os  
from rdkit import Chem  
from rdkit.Chem import Descriptors  
import numpy as np  
from sklearn.model\_selection import train\_test\_split  
from sklearn.linear\_model import LinearRegression  
from sklearn.metrics import mean\_squared\_error, r2\_score  
from sklearn.preprocessing import StandardScaler  
  
# Định nghĩa đường dẫn cơ sở  
base\_path = "../data" # Điều chỉnh nếu cần  
  
# Tải dữ liệu từ file CSV  
data\_file = os.path.join(base\_path, "data.csv")  
df = pd.read\_csv(data\_file)  
  
# Làm sạch và tiền xử lý dữ liệu  
df.dropna(subset=['standard\_value', 'pchembl\_value'], inplace=True)  
df['standard\_value'] = pd.to\_numeric(df['standard\_value'], errors='coerce')  
df.dropna(subset=['standard\_value'], inplace=True)  
  
# Hàm tính toán các mô tả phân tử bằng RDKit  
def calculate\_descriptors(smiles):  
 try:  
 mol = Chem.MolFromSmiles(smiles)  
 if mol is None:  
 return None  
 descriptors = {desc[0]: desc[1](mol) for desc in Descriptors.descList}  
 return descriptors  
 except:  
 return None  
  
# Áp dụng tính toán mô tả cho mỗi hợp chất  
# Giả sử bạn có một cột tên là 'smiles' trong dataframe của bạn  
# Ví dụ: tạo một cột smiles (thay thế chembl\_id bằng một nguồn smiles thích hợp)  
smiles\_dict = {'CHEMBL206': 'c1ccccc1', 'CHEMBL465': 'C1CCCCC1'}  
df['smiles'] = df['chembl\_id'].map(smiles\_dict)  
df.dropna(subset=['smiles'], inplace=True)  
  
df['descriptors'] = df['smiles'].apply(calculate\_descriptors)  
df.dropna(subset=['descriptors'], inplace=True)  
  
  
# Chuyển đổi các mô tả thành DataFrame  
descriptors\_df = pd.DataFrame(df['descriptors'].tolist())  
df = pd.concat([df, descriptors\_df], axis=1)  
df.dropna(inplace=True)  
  
# Chuẩn bị dữ liệu cho học máy  
X = df[[col for col in df.columns if col not in ['chembl\_id', 'pref\_name', 'standard\_type', 'standard\_value', 'standard\_units', 'target\_type', 'target\_name', 'smiles', 'descriptors', 'pchembl\_value']]] # Lựa chọn đặc trưng  
y = df['pchembl\_value'] # Biến mục tiêu  
  
# Chia tỷ lệ dữ liệu  
scaler = StandardScaler()  
X\_scaled = scaler.fit\_transform(X)  
  
# Chia dữ liệu thành tập huấn luyện và tập kiểm tra  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(X\_scaled, y, test\_size=0.2, random\_state=42)  
  
# Huấn luyện mô hình hồi quy tuyến tính  
model = LinearRegression()  
model.fit(X\_train, y\_train)  
  
# Dự đoán  
y\_pred = model.predict(X\_test)  
  
# Đánh giá mô hình  
mse = mean\_squared\_error(y\_test, y\_pred)  
r2 = r2\_score(y\_test, y\_pred)  
  
print(f"Sai số bình phương trung bình: {mse}")  
print(f"R-squared: {r2}")

**4. Giải quyết Lỗi (Error Resolution)**

* **Lỗi SQL: ERROR: operator does not exist: numeric ~ unknown**
* Lỗi này xảy ra do bạn đang cố gắng sử dụng toán tử ~ (so khớp biểu thức chính quy) trên một cột kiểu số. Để khắc phục, hãy đảm bảo rằng cột standard\_value là kiểu số hoặc chuyển đổi nó thành kiểu văn bản trước khi so khớp. Code SQL đã được sửa đổi để bao gồm act.standard\_value::text ~ '^[0-9\.]+$' để đảm bảo rằng chỉ các giá trị số mới được chọn.
* **Lỗi Python: squared=False trong mean\_squared\_error**
* Nếu bạn đang sử dụng phiên bản scikit-learn cũ, tham số squared=False có thể không được hỗ trợ trong hàm mean\_squared\_error. Bạn có thể khắc phục bằng cách tính căn bậc hai của MSE theo cách thủ công:
* mse = mean\_squared\_error(y\_test, y\_pred)  
  rmse = np.sqrt(mse) # Calculate RMSE manually  
  print(f"Root Mean Squared Error: {rmse}")

**5. Ví dụ Code (Code Examples)**

Dưới đây là 5 ví dụ code SQL và Python mẫu:

**Ví dụ 1: Trích xuất dữ liệu cơ bản (Basic Data Extraction)**

**SQL:**

-- Select compound IDs and their preferred names  
SELECT chembl\_id, pref\_name FROM compound\_structures LIMIT 5;

**Python:**

import pandas as pd  
import os  
  
base\_path = "../data"  
data\_file = os.path.join(base\_path, "data.csv")  
df = pd.read\_csv(data\_file)  
  
print(df[['chembl\_id', 'pref\_name']].head())

**Ví dụ 2: Lọc dữ liệu theo loại hoạt tính (Filter by Activity Type)**

**SQL:**

-- Select activities with standard\_type 'Ki'  
SELECT chembl\_id, standard\_value FROM activities WHERE standard\_type = 'Ki' LIMIT 5;

**Python:**

import pandas as pd  
import os  
  
base\_path = "../data"  
data\_file = os.path.join(base\_path, "data.csv")  
df = pd.read\_csv(data\_file)  
  
df\_ki = df[df['standard\_type'] == 'Ki']  
print(df\_ki[['chembl\_id', 'standard\_value']].head())

**Ví dụ 3: Tính toán mô tả phân tử đơn giản (Calculate Simple Molecular Descriptor)**

**SQL:**

Không thể thực hiện tính toán mô tả phân tử trực tiếp trong SQL.

**Python:**

from rdkit import Chem  
from rdkit.Chem import Descriptors  
  
smiles = 'CC(=O)Oc1ccccc1C(=O)O' # Example SMILES string  
mol = Chem.MolFromSmiles(smiles)  
molecular\_weight = Descriptors.MolWt(mol)  
  
print(f"Molecular Weight: {molecular\_weight}")

**Ví dụ 4: Phân tích thống kê đơn giản (Simple Statistical Analysis)**

**SQL:**

-- Calculate the average pchembl\_value  
SELECT AVG(pchembl\_value) FROM activities WHERE pchembl\_value IS NOT NULL;

**Python:**

import pandas as pd  
import os  
  
base\_path = "../data"  
data\_file = os.path.join(base\_path, "data.csv")  
df = pd.read\_csv(data\_file)  
  
average\_pchembl = df['pchembl\_value'].mean()  
print(f"Average pchembl\_value: {average\_pchembl}")

**Ví dụ 5: Xây dựng mô hình học máy đơn giản (Simple Machine Learning Model)**

**SQL:**

Không thể xây dựng mô hình học máy trực tiếp trong SQL.

**Python:**

import pandas as pd  
import os  
from sklearn.linear\_model import LinearRegression  
from sklearn.model\_selection import train\_test\_split  
from sklearn.metrics import mean\_squared\_error  
  
# Create a dummy dataset  
data = {'feature1': [1, 2, 3, 4, 5], 'target': [2, 4, 5, 4, 5]}  
df = pd.DataFrame(data)  
  
# Prepare data  
X = df[['feature1']]  
y = df['target']  
  
# Split data  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)  
  
# Train model  
model = LinearRegression()  
model.fit(X\_train, y\_train)  
  
# Predict  
y\_pred = model.predict(X\_test)  
  
# Evaluate  
mse = mean\_squared\_error(y\_test, y\_pred)  
print(f"Mean Squared Error: {mse}")

**Lưu ý:**

* Đảm bảo bạn đã cài đặt tất cả các thư viện cần thiết (ví dụ: pandas, rdkit, scikit-learn).
* Điều chỉnh đường dẫn file và tên cột cho phù hợp với dữ liệu của bạn.
* Các ví dụ này chỉ là điểm khởi đầu. Bạn có thể mở rộng chúng để thực hiện các phân tích phức tạp hơn.

Chúc bạn thành công với dự án của mình! Nếu bạn có bất kỳ câu hỏi nào khác, đừng ngần ngại hỏi.