# PharmApp Suite

## 🧠 AI for Drug Discovery and Development 🧪

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

## 🧩 Topic: Liên kết dữ liệu thuốc với các pha lâm sàng

Tuyệt vời! Tôi rất vui được hỗ trợ bạn trong dự án phân tích dữ liệu ChEMBL 35 sử dụng RDKit. Với vai trò là một chuyên gia nghiên cứu và phát triển dược học, tôi hiểu rõ tầm quan trọng của việc khai thác dữ liệu ChEMBL để tìm kiếm các hợp chất tiềm năng.

Dưới đây là phân tích chi tiết, hướng dẫn song ngữ, code SQL và Python mẫu, cùng với các ví dụ để giúp bạn bắt đầu dự án của mình.

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

* **Mục tiêu:**
  + Xây dựng mô hình dự đoán hoạt tính của các hợp chất dựa trên cấu trúc hóa học của chúng.
  + Xác định các đặc tính hóa học quan trọng ảnh hưởng đến hoạt tính.
  + Tìm kiếm các hợp chất tiềm năng có hoạt tính mong muốn.
* **Dữ liệu:**
  + Dữ liệu từ ChEMBL 35, bao gồm thông tin về cấu trúc hóa học (SMILES), hoạt tính sinh học (IC50, Ki, EC50, v.v.), và các thuộc tính khác.
  + Sử dụng RDKit để tính toán các descriptor phân tử từ cấu trúc SMILES.
* **Mô hình:**
  + **Hồi quy (Regression):** Dự đoán giá trị hoạt tính liên tục (ví dụ: pIC50).
    - **Ví dụ:** Hồi quy tuyến tính (Linear Regression), Hồi quy Ridge (Ridge Regression), Hồi quy Lasso (Lasso Regression), Máy vector hỗ trợ hồi quy (Support Vector Regression - SVR), Rừng ngẫu nhiên (Random Forest), Gradient Boosting.
  + **Phân loại (Classification):** Dự đoán một hợp chất có hoạt tính hay không (ví dụ: hoạt tính > ngưỡng).
    - **Ví dụ:** Hồi quy Logistic (Logistic Regression), Máy vector hỗ trợ phân loại (Support Vector Classification - SVC), Cây quyết định (Decision Tree), Rừng ngẫu nhiên (Random Forest), Gradient Boosting.
* **Quy trình:**
  1. **Thu thập dữ liệu (Data Acquisition):** Truy vấn cơ sở dữ liệu ChEMBL để lấy dữ liệu cần thiết.
  2. **Tiền xử lý dữ liệu (Data Preprocessing):**
     + Làm sạch dữ liệu (xử lý giá trị thiếu, loại bỏ dữ liệu trùng lặp).
     + Chuyển đổi dữ liệu (ví dụ: chuyển IC50 sang pIC50).
  3. **Tính toán descriptor phân tử (Molecular Descriptor Calculation):** Sử dụng RDKit để tính toán các descriptor phân tử từ cấu trúc SMILES.
  4. **Lựa chọn đặc trưng (Feature Selection):** Chọn các descriptor quan trọng nhất để đưa vào mô hình.
  5. **Xây dựng mô hình (Model Building):** Chọn và huấn luyện mô hình phù hợp.
  6. **Đánh giá mô hình (Model Evaluation):** Đánh giá hiệu suất của mô hình bằng các độ đo phù hợp (ví dụ: R-squared, RMSE, AUC).
  7. **Dự đoán và phân tích (Prediction and Analysis):** Sử dụng mô hình để dự đoán hoạt tính của các hợp chất mới và phân tích các đặc tính quan trọng.

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

| Bước | Tiếng Anh | Tiếng Việt |
| --- | --- | --- |
| 1. Thu thập dữ liệu | Data Acquisition: Query the ChEMBL database to retrieve the necessary data. | Thu thập dữ liệu: Truy vấn cơ sở dữ liệu ChEMBL để lấy dữ liệu cần thiết. |
| 2. Tiền xử lý dữ liệu | Data Preprocessing: Clean the data (handle missing values, remove duplicates). Transform the data (e.g., convert IC50 to pIC50). | Tiền xử lý dữ liệu: Làm sạch dữ liệu (xử lý giá trị thiếu, loại bỏ dữ liệu trùng lặp). Chuyển đổi dữ liệu (ví dụ: chuyển IC50 sang pIC50). |
| 3. Tính toán descriptor phân tử | Molecular Descriptor Calculation: Use RDKit to calculate molecular descriptors from SMILES structures. | Tính toán descriptor phân tử: Sử dụng RDKit để tính toán các descriptor phân tử từ cấu trúc SMILES. |
| 4. Lựa chọn đặc trưng | Feature Selection: Select the most important descriptors to include in the model. | Lựa chọn đặc trưng: Chọn các descriptor quan trọng nhất để đưa vào mô hình. |
| 5. Xây dựng mô hình | Model Building: Select and train an appropriate model. | Xây dựng mô hình: Chọn và huấn luyện mô hình phù hợp. |
| 6. Đánh giá mô hình | Model Evaluation: Evaluate the model’s performance using appropriate metrics (e.g., R-squared, RMSE, AUC). | Đánh giá mô hình: Đánh giá hiệu suất của mô hình bằng các độ đo phù hợp (ví dụ: R-squared, RMSE, AUC). |
| 7. Dự đoán và phân tích | Prediction and Analysis: Use the model to predict the activity of new compounds and analyze important characteristics. | Dự đoán và phân tích: Sử dụng mô hình để dự đoán hoạt tính của các hợp chất mới và phân tích các đặc tính quan trọng. |
| **Lỗi thường gặp (Common Errors)** | ERROR: operator does not exist: numeric ~ unknown: This error occurs because you’re trying to use the ~ operator (regular expression matching) on a numeric column. You need to cast the column to text first. | Lỗi: operator does not exist: numeric ~ unknown: Lỗi này xảy ra khi bạn cố gắng sử dụng toán tử ~ (so khớp biểu thức chính quy) trên một cột số. Bạn cần chuyển đổi cột thành văn bản trước. |
| Scikit-learn version issue | Older scikit-learn versions may not support squared=False in mean\_squared\_error. Update your scikit-learn version or remove the argument. | Phiên bản Scikit-learn cũ có thể không hỗ trợ squared=False trong mean\_squared\_error. Hãy cập nhật phiên bản Scikit-learn của bạn hoặc xóa đối số này. |

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

**SQL (Ví dụ: Lấy dữ liệu từ ChEMBL cho một target cụ thể):**

-- Get data for a specific target (e.g., target\_chembl\_id = 'CHEMBL205')  
SELECT  
 cmp.chembl\_id,  
 cmp.canonical\_smiles,  
 act.standard\_type,  
 act.standard\_value,  
 act.standard\_units  
FROM  
 compound\_structures cmp  
JOIN  
 activities act ON cmp.molregno = act.molregno  
JOIN  
 target\_dictionary td ON act.tid = td.tid  
WHERE  
 td.target\_chembl\_id = 'CHEMBL205'  
 AND act.standard\_type = 'IC50'  
 AND act.standard\_units = 'nM'  
 AND act.standard\_value IS NOT NULL  
 AND act.standard\_value::text ~ '^[0-9\.]+$' -- Corrected line: Casting to text for regex  
LIMIT 100;  
  
--Giải thích:  
-- Lấy thông tin về hợp chất, cấu trúc, loại hoạt tính, giá trị, và đơn vị  
-- Từ bảng cấu trúc hợp chất, hoạt động, và từ điển mục tiêu  
-- Lọc theo target\_chembl\_id cụ thể (ví dụ: CHEMBL205), loại hoạt tính là IC50, đơn vị là nM, giá trị không null, và giá trị chỉ chứa số và dấu chấm (để đảm bảo tính hợp lệ)  
-- Giới hạn kết quả 100 dòng  
  
-- Explanation:  
-- Get compound information, structure, activity type, value, and units  
-- From the compound structure, activities, and target dictionary tables  
-- Filter by specific target\_chembl\_id (e.g., CHEMBL205), activity type is IC50, units are nM, value is not null, and value contains only numbers and dots (to ensure validity)  
-- Limit the result to 100 rows

**Python (Ví dụ: Tính toán descriptor phân tử và xây dựng mô hình):**

import os  
import pandas as pd  
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  
from sklearn.pipeline import Pipeline  
  
# Đường dẫn cơ sở (Base path)  
base\_path = "../data"  
  
# Tên file CSV (CSV file name)  
csv\_file = "chembl\_205\_ic50\_100.csv" # Replace with your actual file name  
file\_path = os.path.join(base\_path, csv\_file)  
  
# Đọc dữ liệu từ file CSV (Read data from CSV file)  
try:  
 df = pd.read\_csv(file\_path)  
except FileNotFoundError:  
 print(f"Error: File not found at {file\_path}")  
 exit()  
  
# In ra thông tin cơ bản của DataFrame (Print basic DataFrame info)  
print("DataFrame Info:")  
print(df.info())  
  
# Hàm tính toán descriptor phân tử (Function to calculate molecular descriptors)  
def calculate\_descriptors(smiles):  
 mol = Chem.MolFromSmiles(smiles)  
 if mol is None:  
 return None  
 descriptors = {}  
 for name, func in Descriptors.descList:  
 try:  
 descriptors[name] = func(mol)  
 except:  
 descriptors[name] = np.nan # Handle potential errors  
 return descriptors  
  
# Áp dụng hàm tính toán descriptor (Apply descriptor calculation function)  
df['descriptors'] = df['canonical\_smiles'].apply(calculate\_descriptors)  
  
# Loại bỏ các dòng có lỗi (Remove rows with errors)  
df = df.dropna(subset=['descriptors'])  
  
# Chuyển đổi descriptor thành DataFrame (Convert descriptors to DataFrame)  
descriptors\_df = pd.DataFrame(df['descriptors'].tolist())  
  
# Gộp DataFrame descriptor với DataFrame chính (Merge descriptor DataFrame with main DataFrame)  
df = pd.concat([df, descriptors\_df], axis=1)  
  
# Chuyển đổi IC50 thành pIC50 (Convert IC50 to pIC50)  
df['pIC50'] = -np.log10(df['standard\_value'] \* 1e-9) # Convert nM to M  
  
# Chọn các đặc trưng và biến mục tiêu (Select features and target variable)  
features = descriptors\_df.columns  
target = 'pIC50'  
  
# Loại bỏ các giá trị vô cực và NaN (Remove infinite and NaN values)  
df = df[np.isfinite(df[target])]  
df = df.dropna(axis=1, how='any')  
  
# Chia dữ liệu thành tập huấn luyện và tập kiểm tra (Split data into training and test sets)  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(df[features].fillna(0), df[target], test\_size=0.2, random\_state=42)  
  
# Chuẩn hóa dữ liệu và xây dựng mô hình hồi quy tuyến tính (Standardize data and build linear regression model)  
pipeline = Pipeline([  
 ('scaler', StandardScaler()),  
 ('model', LinearRegression())  
])  
  
# Huấn luyện mô hình (Train the model)  
pipeline.fit(X\_train, y\_train)  
  
# Dự đoán trên tập kiểm tra (Predict on the test set)  
y\_pred = pipeline.predict(X\_test)  
  
# Đánh giá mô hình (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}')  
  
# In ra 10 dòng đầu của DataFrame (Print the first 10 rows of the DataFrame)  
print("First 10 rows of DataFrame:")  
print(df.head(10))

**Giải thích:**

1. **Import thư viện:** Import các thư viện cần thiết (pandas, RDKit, scikit-learn).
2. **Đọc dữ liệu:** Đọc dữ liệu từ file CSV.
3. **Tính toán descriptor:** Tính toán descriptor phân tử bằng RDKit.
4. **Tiền xử lý dữ liệu:** Xử lý giá trị thiếu, chuyển đổi IC50 sang pIC50.
5. **Chia dữ liệu:** Chia dữ liệu thành tập huấn luyện và tập kiểm tra.
6. **Xây dựng mô hình:** Xây dựng mô hình hồi quy tuyến tính.
7. **Đánh giá mô hình:** Đánh giá hiệu suất của mô hình.

**Explanation:**

1. **Import libraries:** Import necessary libraries (pandas, RDKit, scikit-learn).
2. **Read data:** Read data from CSV file.
3. **Calculate descriptors:** Calculate molecular descriptors using RDKit.
4. **Data preprocessing:** Handle missing values, convert IC50 to pIC50.
5. **Split data:** Split data into training and test sets.
6. **Build model:** Build a linear regression model.
7. **Evaluate model:** Evaluate the model’s performance.

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

**Ví dụ 1: Lọc các hợp chất có hoạt tính IC50 dưới 100 nM (SQL)**

SELECT  
 cmp.chembl\_id,  
 cmp.canonical\_smiles,  
 act.standard\_value  
FROM  
 compound\_structures cmp  
JOIN  
 activities act ON cmp.molregno = act.molregno  
JOIN  
 target\_dictionary td ON act.tid = td.tid  
WHERE  
 td.target\_chembl\_id = 'CHEMBL205'  
 AND act.standard\_type = 'IC50'  
 AND act.standard\_units = 'nM'  
 AND act.standard\_value <= 100  
LIMIT 100;

**Ví dụ 1: Tính toán logP và MW (Python)**

from rdkit import Chem  
from rdkit.Chem import Descriptors  
  
def calculate\_logp\_mw(smiles):  
 mol = Chem.MolFromSmiles(smiles)  
 if mol is None:  
 return None, None  
 logp = Descriptors.MolLogP(mol)  
 mw = Descriptors.MolWt(mol)  
 return logp, mw  
  
smiles = 'CC(=O)Oc1ccccc1C(=O)O'  
logp, mw = calculate\_logp\_mw(smiles)  
print(f'LogP: {logp}, MW: {mw}')

**Ví dụ 2: Lấy các hợp chất tác động lên một protein cụ thể (SQL)**

SELECT  
 cmp.chembl\_id,  
 cmp.canonical\_smiles  
FROM  
 compound\_structures cmp  
JOIN  
 activities act ON cmp.molregno = act.molregno  
JOIN  
 target\_dictionary td ON act.tid = td.tid  
WHERE  
 td.target\_chembl\_id = 'CHEMBL301' -- Replace with desired target  
LIMIT 100;

**Ví dụ 2: Tính toán TPSA (Python)**

from rdkit import Chem  
from rdkit.Chem import rdMolDescriptors  
  
def calculate\_tpsa(smiles):  
 mol = Chem.MolFromSmiles(smiles)  
 if mol is None:  
 return None  
 tpsa = rdMolDescriptors.CalcTPSA(mol)  
 return tpsa  
  
smiles = 'c1ccccc1O'  
tpsa = calculate\_tpsa(smiles)  
print(f'TPSA: {tpsa}')

**Ví dụ 3: Tìm các hợp chất có Ki < 10 nM (SQL)**

SELECT  
 cmp.chembl\_id,  
 cmp.canonical\_smiles,  
 act.standard\_value  
FROM  
 compound\_structures cmp  
JOIN  
 activities act ON cmp.molregno = act.molregno  
JOIN  
 target\_dictionary td ON act.tid = td.tid  
WHERE  
 td.target\_chembl\_id = 'CHEMBL205'  
 AND act.standard\_type = 'Ki'  
 AND act.standard\_units = 'nM'  
 AND act.standard\_value < 10  
LIMIT 100;

**Ví dụ 3: Tính toán số lượng vòng (Python)**

from rdkit import Chem  
from rdkit.Chem import Descriptors  
  
def calculate\_ring\_count(smiles):  
 mol = Chem.MolFromSmiles(smiles)  
 if mol is None:  
 return None  
 ring\_count = Descriptors.RingCount(mol)  
 return ring\_count  
  
smiles = 'c1ccccc1Cc2ccccc2'  
ring\_count = calculate\_ring\_count(smiles)  
print(f'Ring Count: {ring\_count}')

**Ví dụ 4: Lấy các hợp chất có hoạt tính trên một đường dẫn sinh học cụ thể (SQL)**

SELECT  
 cmp.chembl\_id,  
 cmp.canonical\_smiles  
FROM  
 compound\_structures cmp  
JOIN  
 activities act ON cmp.molregno = act.molregno  
JOIN  
 target\_dictionary td ON act.tid = td.tid  
JOIN  
 target\_components tc ON td.tid = tc.tid  
JOIN  
 component\_sequences cs ON tc.component\_id = cs.component\_id  
WHERE  
 cs.accession = 'P00533' -- Example: EGFR  
LIMIT 100;

**Ví dụ 4: Scale dữ liệu sử dụng StandardScaler (Python)**

from sklearn.preprocessing import StandardScaler  
import numpy as np  
  
# Ví dụ dữ liệu  
data = np.array([[1.0], [2.0], [3.0], [4.0], [5.0]])  
  
# Khởi tạo StandardScaler  
scaler = StandardScaler()  
  
# Fit và transform dữ liệu  
scaled\_data = scaler.fit\_transform(data)  
  
print("Original data:\n", data)  
print("Scaled data:\n", scaled\_data)

**Ví dụ 5: Lấy dữ liệu pIC50 từ ChEMBL (SQL)**

SELECT  
 cmp.chembl\_id,  
 cmp.canonical\_smiles,  
 -LOG10(act.standard\_value \* POW(10, -9)) AS pIC50 -- Calculate pIC50  
FROM  
 compound\_structures cmp  
JOIN  
 activities act ON cmp.molregno = act.molregno  
JOIN  
 target\_dictionary td ON act.tid = td.tid  
WHERE  
 td.target\_chembl\_id = 'CHEMBL205'  
 AND act.standard\_type = 'IC50'  
 AND act.standard\_units = 'nM'  
 AND act.standard\_value IS NOT NULL  
 AND act.standard\_value::text ~ '^[0-9\.]+$' -- Ensure the value is numeric  
LIMIT 100;

**Ví dụ 5: Sử dụng Random Forest Regressor (Python)**

from sklearn.ensemble import RandomForestRegressor  
from sklearn.model\_selection import train\_test\_split  
from sklearn.metrics import mean\_squared\_error, r2\_score  
import pandas as pd  
import numpy as np  
  
# Tạo dữ liệu ví dụ  
data = {'feature1': np.random.rand(100),  
 'feature2': np.random.rand(100),  
 'target': 2 \* np.random.rand(100)}  
df = pd.DataFrame(data)  
  
# Chia dữ liệu thành features (X) và target (y)  
X = df[['feature1', 'feature2']]  
y = df['target']  
  
# Chia dữ liệu thành training set và test set  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)  
  
# Khởi tạo Random Forest Regressor  
rf\_regressor = RandomForestRegressor(n\_estimators=100, random\_state=42)  
  
# Huấn luyện mô hình  
rf\_regressor.fit(X\_train, y\_train)  
  
# Dự đoán trên test set  
y\_pred = rf\_regressor.predict(X\_test)  
  
# Đánh giá mô hình  
mse = mean\_squared\_error(y\_test, y\_pred)  
r2 = r2\_score(y\_test, y\_pred)  
  
print("Mean Squared Error:", mse)  
print("R-squared:", r2)

**Lưu ý quan trọng:**

* **Thay đổi đường dẫn:** Thay đổi base\_path và csv\_file cho phù hợp với cấu trúc thư mục và tên file của bạn.
* **Xử lý lỗi:** Các đoạn code trên chỉ là ví dụ, bạn cần thêm các xử lý lỗi để đảm bảo chương trình chạy ổn định.
* **Lựa chọn mô hình:** Lựa chọn mô hình phù hợp với dữ liệu và mục tiêu của bạn.
* **Tối ưu hóa mô hình:** Tối ưu hóa các tham số của mô hình để đạt được hiệu suất tốt nhất.
* **Phiên bản scikit-learn:** Hãy đảm bảo bạn đang sử dụng phiên bản scikit-learn tương thích. Nếu bạn gặp lỗi với tham số squared=False trong mean\_squared\_error, hãy cập nhật scikit-learn hoặc bỏ tham số đó.
* **Kiểm tra kiểu dữ liệu:** Đảm bảo kiểu dữ liệu của các cột phù hợp với các phép toán và hàm bạn sử dụng.

Hy vọng những hướng dẫn và ví dụ này sẽ giúp bạn bắt đầu dự án phân tích dữ liệu ChEMBL 35 của mình một cách hiệu quả. Nếu bạn có bất kỳ câu hỏi nào, đừng ngần ngại hỏi!