# PharmApp Suite

## 🧠 AI for Drug Discovery and Development 🧪

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

## 🧩 Topic: Phân tích thuốc ChEMBL đã được FDA phê duyệt

Tuyệt vời! Tôi rất vui được hỗ trợ bạn trong dự án phân tích dữ liệu ChEMBL 35 này. Với vai trò là một chuyên gia R&D 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 và các ví dụ code để bạn tham khảo:

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

**1.1. Mục tiêu (Objective)**

* Xây dựng mô hình dự đoán hoạt tính sinh học của các hợp chất dựa trên cấu trúc hóa học của chúng.
* Phân tích mối quan hệ giữa các đặc tính cấu trúc (descriptors) và hoạt tính sinh học.
* Tìm kiếm các hợp chất tiềm năng có hoạt tính mong muốn.

**1.2. Dữ liệu (Data)**

* **Nguồn:** ChEMBL 35 (cơ sở dữ liệu lớn chứa thông tin về các phân tử sinh học và hoạt tính của chúng).
* **Bảng chính:**
  + activities: Chứa thông tin về hoạt tính sinh học của các hợp chất (ví dụ: IC50, Ki, EC50).
  + molecule\_dictionary: Chứa thông tin về cấu trúc hóa học của các hợp chất (ví dụ: SMILES).
  + compound\_structures: Chứa cấu trúc hóa học của các hợp chất ở nhiều định dạng.
* **Đặc trưng hóa học (Chemical descriptors):** Sử dụng RDKit để tính toán các đặc trưng hóa học từ cấu trúc SMILES. Các đặc trưng này có thể bao gồm:
  + Molecular Weight (Khối lượng phân tử)
  + LogP (Hệ số phân vùng octanol-water)
  + Hydrogen Bond Donors (Số lượng liên kết hydro cho)
  + Hydrogen Bond Acceptors (Số lượng liên kết hydro nhận)
  + Topological Polar Surface Area (Diện tích bề mặt cực topo)
  + Số vòng (Number of rings)
  + … và nhiều đặc trưng khác.

**1.3. Mô hình (Model)**

Chúng ta có thể sử dụng nhiều mô hình khác nhau để dự đoán hoạt tính sinh học, tùy thuộc vào mục tiêu cụ thể và tính chất của dữ liệu. Dưới đây là một số lựa chọn phổ biến:

* **Hồi quy tuyến tính (Linear Regression):** Đơn giản, dễ hiểu, phù hợp khi có mối quan hệ tuyến tính giữa các đặc trưng và hoạt tính.
* **Hồi quy Ridge/Lasso (Ridge/Lasso Regression):** Cải thiện hồi quy tuyến tính bằng cách thêm các điều kiện ràng buộc để tránh overfitting.
* **Máy vector hỗ trợ (Support Vector Machines - SVM):** Hiệu quả trong không gian đặc trưng cao, có thể xử lý dữ liệu phi tuyến tính.
* **Rừng ngẫu nhiên (Random Forest):** Mạnh mẽ, có khả năng chống overfitting, cung cấp thông tin về tầm quan trọng của các đặc trưng.
* **Mạng nơ-ron (Neural Networks):** Linh hoạt, có khả năng học các mối quan hệ phức tạp, nhưng đòi hỏi lượng dữ liệu lớn và điều chỉnh cẩn thận.

**1.4. Quy trình (Process)**

1. **Thu thập và chuẩn bị dữ liệu (Data Collection and Preparation):**
   * Truy vấn dữ liệu từ cơ sở dữ liệu ChEMBL bằng SQL.
   * Làm sạch và tiền xử lý dữ liệu (xử lý giá trị thiếu, loại bỏ các giá trị ngoại lệ).
   * Chuyển đổi dữ liệu SMILES thành các đối tượng phân tử RDKit.
   * Tính toán các đặc trưng hóa học bằng RDKit.
2. **Lựa chọn đặc trưng (Feature Selection):**
   * Chọn các đặc trưng quan trọng nhất bằng các phương pháp như:
     + Phân tích phương sai (Variance Thresholding)
     + Lựa chọn đặc trưng đơn biến (Univariate Feature Selection)
     + Lựa chọn đặc trưng dựa trên cây (Tree-based Feature Selection)
     + Loại bỏ đệ quy đặc trưng (Recursive Feature Elimination)
3. **Xây dựng và huấn luyện mô hình (Model Building and Training):**
   * Chia dữ liệu thành tập huấn luyện và tập kiểm tra.
   * Chọn một mô hình phù hợp và huấn luyện nó trên tập huấn luyện.
4. **Đánh giá mô hình (Model Evaluation):**
   * Đánh giá hiệu suất của mô hình trên tập kiểm tra bằng các độ đo phù hợp (ví dụ: R-squared, RMSE, MAE).
5. **Tối ưu hóa mô hình (Model Optimization):**
   * Điều chỉnh các siêu tham số của mô hình để cải thiện hiệu suất.
6. **Dự đoán và phân tích (Prediction and Analysis):**
   * Sử dụng mô hình đã huấn luyện để dự đoán hoạt tính của các hợp chất mới.
   * Phân tích tầm quan trọng của các đặc trưng để hiểu rõ hơn về mối quan hệ cấu trúc-hoạt tính.

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

| Bước (Step) | Tiếng Anh (English) | Tiếng Việt (Vietnamese) |
| --- | --- | --- |
| 1. Kết nối CSDL (Connect DB) | Connect to the ChEMBL database using the provided credentials. | Kết nối đến cơ sở dữ liệu ChEMBL bằng thông tin đăng nhập được cung cấp. |
| 2. Truy vấn dữ liệu (Query data) | Query the activities, molecule\_dictionary, and compound\_structures tables to retrieve relevant data. | Truy vấn các bảng activities, molecule\_dictionary và compound\_structures để lấy dữ liệu liên quan. |
| 3. Tiền xử lý (Preprocessing) | Clean and preprocess the data (handle missing values, outliers). | Làm sạch và tiền xử lý dữ liệu (xử lý giá trị thiếu, giá trị ngoại lệ). |
| 4. Đặc trưng (Feature Eng) | Calculate chemical descriptors using RDKit from SMILES strings. | Tính toán các đặc trưng hóa học bằng RDKit từ chuỗi SMILES. |
| 5. Xây dựng mô hình (Model Build) | Build and train a machine learning model to predict biological activity. | Xây dựng và huấn luyện một mô hình học máy để dự đoán hoạt tính sinh học. |
| 6. Đánh giá (Evaluation) | Evaluate the model’s performance using appropriate metrics. | Đánh giá hiệu suất của mô hình bằng các độ đo phù hợp. |
| 7. Phân tích (Analysis) | Analyze the results and identify potential lead compounds. | Phân tích kết quả và xác định các hợp chất tiềm năng. |

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

**3.1. SQL (ví dụ lấy 100 dòng)**

-- English  
-- Select 100 activities with standard\_type 'IC50' and valid standard\_value  
SELECT act.molregno, md.chembl\_id, act.standard\_value, act.standard\_units,md.molecule\_structures  
FROM activities act  
JOIN molecule\_dictionary md ON act.molregno = md.molregno  
WHERE act.standard\_type = 'IC50'  
 AND act.standard\_relation = '='  
 AND act.standard\_value IS NOT NULL  
 AND act.standard\_value > 0  
 AND act.standard\_units = 'nM'  
 AND md.molecule\_structures IS NOT NULL  
 AND md.molecule\_structures LIKE '%SMILES%'  
  
LIMIT 100;  
  
-- Vietnamese  
-- Chọn 100 hoạt động có standard\_type là 'IC50' và standard\_value hợp lệ  
SELECT act.molregno, md.chembl\_id, act.standard\_value, act.standard\_units,md.molecule\_structures  
FROM activities act  
JOIN molecule\_dictionary md ON act.molregno = md.molregno  
WHERE act.standard\_type = 'IC50'  
 AND act.standard\_relation = '='  
 AND act.standard\_value IS NOT NULL  
 AND act.standard\_value > 0  
 AND act.standard\_units = 'nM'  
 AND md.molecule\_structures IS NOT NULL  
 AND md.molecule\_structures LIKE '%SMILES%'  
LIMIT 100;

**Lưu ý:** Lỗi ERROR: operator does not exist: numeric ~ unknown thường xảy ra khi bạn cố gắng sử dụng toán tử ~ (regular expression match) trên một cột kiểu số. Để khắc phục, bạn có thể chuyển đổi cột số thành kiểu text trước khi so sánh: Không cần thiết vì đã lọc act.standard\_value IS NOT NULL AND act.standard\_value > 0

**3.2. Python (ví dụ tính toán đặc trưng và xây dựng mô hình)**

# English  
import os  
import pandas as pd  
from rdkit import Chem  
from rdkit.Chem import Descriptors  
from sklearn.model\_selection import train\_test\_split  
from sklearn.linear\_model import LinearRegression  
from sklearn.metrics import mean\_squared\_error  
import numpy as np  
# Define the base path  
base\_path = "../data"  
  
# Load data from CSV  
data = pd.read\_csv(os.path.join(base\_path, "chembl\_ic50\_data.csv"))  
  
# Function to calculate molecular descriptors  
def calculate\_descriptors(smiles):  
 mol = Chem.MolFromSmiles(smiles)  
 if mol is not None:  
 descriptors = {desc[0]: desc[1](mol) for desc in Descriptors.descList}  
 return descriptors  
 else:  
 return None  
  
# Apply descriptor calculation and handle errors  
descriptors\_list = []  
for index, row in data.iterrows():  
 try:  
 smiles = row['molecule\_structures'].split('"')[3]  
 descriptors = calculate\_descriptors(smiles)  
 if descriptors is not None:  
 descriptors['molregno'] = row['molregno'] # Keep molregno for merging  
 descriptors\_list.append(descriptors)  
 else:  
 print(f"Could not compute descriptors for molregno {row['molregno']}")  
 except Exception as e:  
 print(f"Error processing molregno {row['molregno']}: {e}")  
  
# Convert list of descriptors to DataFrame  
descriptors\_df = pd.DataFrame(descriptors\_list)  
  
# Merge descriptors with activity data  
merged\_data = pd.merge(data[['molregno', 'standard\_value']], descriptors\_df, on='molregno', how='inner')  
  
# Handle missing values by filling with the mean  
merged\_data.fillna(merged\_data.mean(), inplace=True)  
  
# Prepare data for modeling  
X = merged\_data.drop(['molregno', 'standard\_value'], axis=1)  
y = merged\_data['standard\_value']  
  
# 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)  
  
# Train a linear regression model  
model = LinearRegression()  
model.fit(X\_train, y\_train)  
  
# Make predictions and evaluate the model  
y\_pred = model.predict(X\_test)  
mse = mean\_squared\_error(y\_test, y\_pred)  
print(f"Mean Squared Error: {mse}")  
  
# --- Vietnamese ---  
# Import các thư viện cần thiết  
# Đường dẫn cơ sở đến thư mục dữ liệu  
# Đọc dữ liệu từ file CSV  
# Hàm tính toán các đặc trưng phân tử  
# Áp dụng tính toán đặc trưng và xử lý lỗi  
# Chuyển đổi danh sách các đặc trưng thành DataFrame  
# Gộp các đặc trưng với dữ liệu hoạt tính  
# Xử lý giá trị thiếu bằng cách điền bằng giá trị trung bình  
# Chuẩn bị dữ liệu cho mô hình hóa  
# Chia dữ liệu thành tập huấn luyện và tập kiểm tra  
# Huấn luyện mô hình hồi quy tuyến tính  
# Dự đoán và đánh giá mô hình  
# In ra lỗi bình phương trung bình

**Lưu ý:** Nếu bạn gặp lỗi liên quan đến phiên bản scikit-learn, hãy cập nhật lên phiên bản mới nhất hoặc loại bỏ tham số squared=False nếu không cần thiết.

**4. Ví dụ Code (Code Examples)**

Dưới đây là 5 ví dụ code SQL và Python mẫu để bạn tham khảo:

**Ví dụ 1: Lọc dữ liệu theo khoảng giá trị hoạt tính (Filtering data by activity range)**

* **SQL:**

-- English  
-- Select compounds with IC50 values between 100 and 1000 nM  
SELECT md.chembl\_id, act.standard\_value  
FROM activities act  
JOIN molecule\_dictionary md ON act.molregno = md.molregno  
WHERE act.standard\_type = 'IC50'  
 AND act.standard\_relation = '='  
 AND act.standard\_value >= 100  
 AND act.standard\_value <= 1000  
LIMIT 100;  
  
-- Vietnamese  
-- Chọn các hợp chất có giá trị IC50 nằm trong khoảng từ 100 đến 1000 nM  
SELECT md.chembl\_id, act.standard\_value  
FROM activities act  
JOIN molecule\_dictionary md ON act.molregno = md.molregno  
WHERE act.standard\_type = 'IC50'  
 AND act.standard\_relation = '='  
 AND act.standard\_value >= 100  
 AND act.standard\_value <= 1000  
LIMIT 100;

* **Python:**

# English  
# Filter compounds with IC50 values between 100 and 1000 nM  
filtered\_data = data[(data['standard\_value'] >= 100) & (data['standard\_value'] <= 1000)]  
print(filtered\_data.head())  
  
# Vietnamese  
# Lọc các hợp chất có giá trị IC50 nằm trong khoảng từ 100 đến 1000 nM  
filtered\_data = data[(data['standard\_value'] >= 100) & (data['standard\_value'] <= 1000)]  
print(filtered\_data.head())

**Ví dụ 2: Tính toán LogP (Calculating LogP)**

* **Python:**

# English  
# Function to calculate LogP using RDKit  
from rdkit.Chem import AllChem  
from rdkit import DataStructs  
  
def calculate\_logp(smiles):  
 mol = Chem.MolFromSmiles(smiles)  
 if mol is not None:  
 logp = Descriptors.MolLogP(mol)  
 return logp  
 else:  
 return None  
  
# Apply LogP calculation  
data['logp'] = data['molecule\_structures'].apply(lambda x: calculate\_logp(x.split('"')[3]) if isinstance(x, str) else None)  
print(data[['molecule\_structures', 'logp']].head())  
  
# Vietnamese  
# Hàm tính toán LogP sử dụng RDKit  
def calculate\_logp(smiles):  
 mol = Chem.MolFromSmiles(smiles)  
 if mol is not None:  
 logp = Descriptors.MolLogP(mol)  
 return logp  
 else:  
 return None  
  
# Áp dụng tính toán LogP  
data['logp'] = data['molecule\_structures'].apply(lambda x: calculate\_logp(x.split('"')[3]) if isinstance(x, str) else None)  
print(data[['molecule\_structures', 'logp']].head())

**Ví dụ 3: Sử dụng Random Forest để dự đoán hoạt tính (Using Random Forest for activity prediction)**

* **Python:**

# English  
# Use Random Forest to predict activity  
from sklearn.ensemble import RandomForestRegressor  
  
# Prepare data (assuming X and y are already defined)  
X = merged\_data.drop(['molregno', 'standard\_value'], axis=1)  
y = merged\_data['standard\_value']  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)  
  
# Train a Random Forest model  
rf\_model = RandomForestRegressor(n\_estimators=100, random\_state=42)  
rf\_model.fit(X\_train, y\_train)  
  
# Make predictions and evaluate the model  
y\_pred = rf\_model.predict(X\_test)  
mse = mean\_squared\_error(y\_test, y\_pred)  
print(f"Mean Squared Error (Random Forest): {mse}")  
  
# Vietnamese  
# Sử dụng Random Forest để dự đoán hoạt tính  
# Chuẩn bị dữ liệu (giả sử X và y đã được định nghĩa)  
# Huấn luyện mô hình Random Forest  
# Dự đoán và đánh giá mô hình

**Ví dụ 4: Phân tích tầm quan trọng của đặc trưng (Feature importance analysis)**

* **Python:**

# English  
# Analyze feature importance  
import matplotlib.pyplot as plt  
  
importances = rf\_model.feature\_importances\_  
feature\_names = X.columns  
feature\_importances = pd.DataFrame({'feature': feature\_names, 'importance': importances})  
feature\_importances = feature\_importances.sort\_values('importance', ascending=False)  
  
plt.figure(figsize=(10, 6))  
plt.bar(feature\_importances['feature'], feature\_importances['importance'])  
plt.xticks(rotation=90)  
plt.xlabel('Features')  
plt.ylabel('Importance')  
plt.title('Feature Importance')  
plt.tight\_layout()  
plt.show()  
  
# Vietnamese  
# Phân tích tầm quan trọng của đặc trưng  
# Lấy tầm quan trọng của các đặc trưng từ mô hình  
# Vẽ biểu đồ tầm quan trọng của các đặc trưng

**Ví dụ 5: Lưu kết quả dự đoán (Saving prediction results)**

* **Python:**

# English  
# Save prediction results  
predictions = pd.DataFrame({'actual': y\_test, 'predicted': y\_pred})  
predictions.to\_csv(os.path.join(base\_path, "predictions.csv"), index=False)  
print("Predictions saved to predictions.csv")  
  
# Vietnamese  
# Lưu kết quả dự đoán  
# Tạo DataFrame chứa giá trị thực tế và giá trị dự đoán  
# Lưu DataFrame vào file CSV

Hy vọng những hướng dẫn và ví dụ này sẽ giúp bạn tiến xa hơn trong dự án của mình. Nếu bạn có bất kỳ câu hỏi nào khác, đừng ngần ngại hỏi!