PharmApp Suite

🧠 AI for Drug Discovery and Development 💉

🧩 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
-- 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):

```sql
-- 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
```

```sql
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):

```python
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)
```

```python
# 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:

  ```python
  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:

```sql
-- Select compound IDs and their preferred names
SELECT chembl_id, pref_name FROM compound_structures LIMIT 5;
```

Python:

```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:

```sql
-- Select activities with standard_type 'Ki'
SELECT chembl_id, standard_value FROM activities WHERE standard_type = 'Ki' LIMIT 5;
```

Python:

```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:

```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:

```sql
-- Calculate the average pchembl_value
SELECT AVG(pchembl_value) FROM activities WHERE pchembl_value IS NOT NULL;
```

Python:

```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:

```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.