PharmApp Suite

🧠 AI for Drug Discovery and Development 💉

🧩 Topic: Liên hệ hoạt tính với phân loại target

Tuyệt vời! Tôi rất vui được tham gia dự án phân tích dữ liệu ChEMBL 35 của bạn. Với vai trò là một chuyên gia R&D dược học, tôi sẽ giúp bạn xây dựng quy trình phân tích hiệu quả, tập trung vào các yêu cầu và hạn chế bạn đã nêu.

Tổng quan dự án

Chúng ta sẽ thực hiện các bước sau:

1. **Chuẩn bị dữ liệu:** Sử dụng SQL để truy vấn và lọc dữ liệu từ cơ sở dữ liệu ChEMBL 35, sau đó lưu kết quả vào file CSV.
2. **Phân tích dữ liệu:** Sử dụng RDKit và các thư viện Python khác để tiền xử lý, khám phá và trực quan hóa dữ liệu.
3. **Xây dựng mô hình:** Xây dựng và đánh giá mô hình học máy để dự đoán hoạt tính của các hợp chất.

1. Phân tích mô hình

Mô hình phân tích:

Chúng ta sẽ sử dụng một mô hình phân tích dữ liệu điển hình trong lĩnh vực phát triển thuốc, bao gồm các bước sau:

- **Data Acquisition (Thu thập dữ liệu):** Truy vấn dữ liệu từ cơ sở dữ liệu ChEMBL 35 sử dụng SQL.
- **Data Preprocessing (Tiền xử lý dữ liệu):**
    - Làm sạch dữ liệu: Loại bỏ các giá trị thiếu, xử lý các lỗi định dạng.
    - Chuẩn hóa dữ liệu: Đảm bảo tính nhất quán của dữ liệu.
    - Tính toán Descriptors: Sử dụng RDKit để tính toán các thuộc tính hóa lý của các hợp chất (ví dụ: Molecular Weight, LogP, Hydrogen Bond Donors/Acceptors).
- **Exploratory Data Analysis (EDA - Phân tích khám phá dữ liệu):**
    - Thống kê mô tả: Tính toán các thống kê cơ bản (mean, median, standard deviation) cho các thuộc tính.
    - Trực quan hóa dữ liệu: Sử dụng biểu đồ để khám phá các mối quan hệ giữa các thuộc tính và hoạt tính.
- **Feature Selection/Engineering (Chọn lọc/Xây dựng đặc trưng):**
    - Chọn các đặc trưng quan trọng nhất ảnh hưởng đến hoạt tính.
    - Kết hợp các đặc trưng hiện có để tạo ra các đặc trưng mới.
- **Model Building (Xây dựng mô hình):**
    - Chọn mô hình học máy phù hợp (ví dụ: Random Forest, Support Vector Machine, Neural Network).
    - Huấn luyện mô hình trên dữ liệu huấn luyện.
- **Model Evaluation (Đánh giá mô hình):**
    - Đánh giá hiệu suất của mô hình trên dữ liệu kiểm tra sử dụng các metrics phù hợp (ví dụ: RMSE, R-squared).
- **Model Interpretation (Giải thích mô hình):**
    - Hiểu các yếu tố nào đóng góp nhiều nhất vào dự đoán của mô hình.

- **Deployment (Triển khai):**
    - Sử dụng mô hình để dự đoán hoạt tính của các hợp chất mới.

2. Hướng dẫn song ngữ

2.1 SQL

- **Purpose:** To extract data from the ChEMBL database and save it as a CSV file.
- **Mục đích:** Trích xuất dữ liệu từ cơ sở dữ liệu ChEMBL và lưu nó dưới dạng tệp CSV.

```sql
-- SQL query to extract data from ChEMBL 35
-- Truy vấn SQL để trích xuất dữ liệu từ ChEMBL 35
SELECT
    md.molregno,
    cs.canonical_smiles,
    act.standard_type,
    act.standard_value,
    act.standard_units
FROM
    molecule_dictionary md
JOIN
    compound_structures cs ON md.molregno = cs.molregno
JOIN
    activities act ON md.molregno = act.molregno
WHERE
    act.standard_type = 'IC50'  -- Example: Filter by IC50 values
    AND act.standard_units = 'nM'  -- Example: Filter by nM units
    AND act.standard_value IS NOT NULL
    AND act.standard_value::text ~ '^[0-9\.]+$' --Fix:Check standard_value is numeric
LIMIT 100;
```

Lưu ý:

- Sửa lỗi `ERROR: operator does not exist: numeric ~ unknown`: Sử dụng `act.standard_value::text ~ '^[0-9\.]+$'` để kiểm tra xem `standard_value` có phải là số hay không.
- The `LIMIT 100` clause restricts the output to the first 100 rows. Remove this line if you need more data.
- Mệnh đề `LIMIT 100` giới hạn đầu ra chỉ còn 100 hàng đầu tiên. Xóa dòng này nếu bạn cần nhiều dữ liệu hơn.

2.2 Python

- **Purpose:** To load the CSV file, preprocess data using RDKit, and perform basic analysis.
- **Mục đích:** Tải tệp CSV, tiền xử lý dữ liệu bằng RDKit và thực hiện phân tích cơ bản.

```python
import pandas as pd
from rdkit import Chem
from rdkit.Chem import Descriptors
import os

# Define the base path
base_path = "../data"

# Construct the file path
file_path = os.path.join(base_path, "chembl_data.csv")

# Load the CSV file
try:
    df = pd.read_csv(file_path)
except FileNotFoundError:
```

```
    print(f"Error: The file '{file_path}' was not found.")
    exit()

# Print the first few rows of the dataframe
print(df.head())

# Function to calculate molecular weight using RDKit
def calculate_molecular_weight(smiles):
    try:
        mol = Chem.MolFromSmiles(smiles)
        if mol:
            return Descriptors.MolWt(mol)
        else:
            return None
    except:
        return None

# Apply the function to the 'canonical_smiles' column
df['molecular_weight'] = df['canonical_smiles'].apply(calculate_molecular_weight)

# Display some basic statistics
print(df.describe())

# Handle missing values
df = df.dropna(subset=['standard_value', 'molecular_weight'])

# Convert standard_value to numeric
df['standard_value'] = pd.to_numeric(df['standard_value'], errors='coerce')

# Basic filtering of activity values (example: IC50 < 1000 nM)
df_filtered = df[df['standard_value'] < 1000]

print(df_filtered.head())
```

3. Ví dụ code SQL và Python

3.1 SQL Examples

```
-- 1. Get the number of compounds for each standard_type
-- 1. Lấy số lượng hợp chất cho mỗi standard_type
SELECT standard_type, COUNT(DISTINCT molregno) AS num_compounds
FROM activities
GROUP BY standard_type;

-- 2. Find compounds with a specific substructure (using SMARTS)
-- 2. Tìm các hợp chất có một cấu trúc con cụ thể (sử dụng SMARTS)
SELECT md.molregno, cs.canonical_smiles
FROM molecule_dictionary md
JOIN compound_structures cs ON md.molregno = cs.molregno
WHERE cs.canonical_smiles LIKE '%C(=O)N%'; -- Example: Compounds containing amide bond

-- 3. Get the average molecular weight for compounds with IC50 < 1000 nM
-- 3. Lấy trọng lượng phân tử trung bình cho các hợp chất có IC50 < 1000 nM
SELECT AVG(md.mol_weight)
FROM molecule_dictionary md
JOIN activities act ON md.molregno = act.molregno
WHERE act.standard_type = 'IC50' AND act.standard_value < 1000;

-- 4. Retrieve data for a specific target (e.g., by target_chembl_id)
-- 4. Truy xuất dữ liệu cho một mục tiêu cụ thể (ví dụ: theo target_chembl_id)
```

```sql
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.target_chembl_id = 'CHEMBL205'; -- Example: Dopamine D4 receptor

-- 5. Find compounds with specific properties (e.g., LogP > 3 and Molecular Weight < 500)
-- 5. Tìm các hợp chất có các thuộc tính cụ thể (ví dụ: LogP > 3 và Trọng lượng phân tử < 500)
SELECT md.molregno, cs.canonical_smiles
FROM molecule_dictionary md
JOIN compound_structures cs ON md.molregno = cs.molregno
WHERE md.ALOGP > 3 AND md.mol_weight < 500;
```

3.2 Python Examples

```python
import pandas as pd
from rdkit import Chem
from rdkit.Chem import Descriptors
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score
import numpy as np
import os

# Define the base path
base_path = "../data"

# Construct the file path
file_path = os.path.join(base_path, "chembl_data.csv")


# Load the CSV file
try:
    df = pd.read_csv(file_path)
except FileNotFoundError:
    print(f"Error: The file '{file_path}' was not found.")
    exit()

# Function to calculate molecular weight using RDKit
def calculate_molecular_weight(smiles):
    try:
        mol = Chem.MolFromSmiles(smiles)
        if mol:
            return Descriptors.MolWt(mol)
        else:
            return None
    except:
        return None

# Apply the function to the 'canonical_smiles' column
df['molecular_weight'] = df['canonical_smiles'].apply(calculate_molecular_weight)

# Handle missing values
df = df.dropna(subset=['standard_value', 'molecular_weight', 'canonical_smiles'])

# Convert standard_value to numeric
df['standard_value'] = pd.to_numeric(df['standard_value'], errors='coerce')
```

```
# Drop rows where 'standard_value' is NaN after conversion
df = df.dropna(subset=['standard_value'])


# 1. Calculate LogP using RDKit
# 1. Tính LogP bằng RDKit
def calculate_logp(smiles):
    try:
        mol = Chem.MolFromSmiles(smiles)
        if mol:
            return Descriptors.MolLogP(mol)
        else:
            return None
    except:
        return None

df['logp'] = df['canonical_smiles'].apply(calculate_logp)
df = df.dropna(subset=['logp'])


# 2. Calculate TPSA using RDKit
# 2. Tính TPSA bằng RDKit
def calculate_tpsa(smiles):
    try:
        mol = Chem.MolFromSmiles(smiles)
        if mol:
            return Descriptors.TPSA(mol)
        else:
            return None
    except:
        return None

df['tpsa'] = df['canonical_smiles'].apply(calculate_tpsa)
df = df.dropna(subset=['tpsa'])

# 3. Convert SMILES to Morgan Fingerprints (ECFP4)
# 3. Chuyển đổi SMILES thành Morgan Fingerprints (ECFP4)
from rdkit.Chem import AllChem

def calculate_morgan_fingerprint(smiles, radius=2, nBits=2048):
    try:
        mol = Chem.MolFromSmiles(smiles)
        if mol:
            fp = AllChem.GetMorganFingerprintAsBitVect(mol, radius, nBits=nBits)
            return np.array(list(fp.ToBitString()), dtype=int)
        else:
            return None
    except:
        return None

df['morgan_fp'] = df['canonical_smiles'].apply(calculate_morgan_fingerprint)
df = df.dropna(subset=['morgan_fp'])

# 4. Example: Train a Random Forest Regressor model to predict standard_value
# 4. Ví dụ: Huấn luyện mô hình Random Forest Regressor để dự đoán standard_value
X = np.stack(df['morgan_fp'].values)
y = df['standard_value'].values
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

model = RandomForestRegressor(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

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}')

# 5. Feature Importance
# 5. Độ quan trọng của đặc trưng
feature_importances = model.feature_importances_
print(f'Feature Importances: {feature_importances}')
```

Lưu ý:

- Code Python trên chỉ là ví dụ minh họa. Bạn cần điều chỉnh nó cho phù hợp với mục tiêu nghiên cứu của mình.
- Ensure that you have installed the necessary libraries (e.g., pandas, rdkit, scikit-learn) using pip.
- Đảm bảo bạn đã cài đặt các thư viện cần thiết (ví dụ: pandas, rdkit, scikit-learn) bằng pip.
- Sửa lỗi `ValueError: Solver produced results that are not finite` bằng cách loại bỏ các giá trị `NaN` hoặc `infinite` trong dữ liệu của bạn.

Tôi hy vọng những hướng dẫn này sẽ giúp bạn bắt đầu dự án phân tích dữ liệu ChEMBL 35. Nếu bạn có bất kỳ câu hỏi nào, đừng ngần ngại hỏi. Chúc bạn thành công!