PharmApp Suite

🧠 AI for Drug Discovery and Development 💉

🧩 Topic: Xu hướng thuốc ung thư trong ChEMBL

Tuyệt vời! Tôi rất vui được hỗ trợ bạn trong dự án phân tích dữ liệu ChEMBL 35 kết hợp RDKit. Với vai trò là một chuyên gia nghiên cứu và phát triển dược học, tôi sẽ giúp bạn xây dựng quy trình phân tích hiệu quả, khắc phục các lỗi và cung cấp các ví dụ code hữu ích.

1. Phân tích Mô hình Phân tích (Analysis of Analytical Model)

Dựa trên thông tin bạn cung cấp, chúng ta sẽ xây dựng một mô hình phân tích dữ liệu ChEMBL 35 theo các bước sau:

- **Data Extraction (Trích xuất dữ liệu):** Sử dụng SQL để truy vấn và trích xuất dữ liệu từ cơ sở dữ liệu PostgreSQL (ChEMBL 35). Dữ liệu này sẽ bao gồm thông tin về các hợp chất (compounds), hoạt tính sinh học (bioactivities) và các thuộc tính liên quan.
- **Data Preprocessing (Tiền xử lý dữ liệu):**
    - **Data Cleaning:** Loại bỏ các giá trị bị thiếu (missing values) hoặc không hợp lệ (invalid values).
    - **Data Transformation:** Chuyển đổi dữ liệu sang định dạng phù hợp cho phân tích, ví dụ: chuẩn hóa (normalization) hoặc mã hóa (encoding).
- **Feature Engineering (Xây dựng đặc trưng):** Sử dụng RDKit để tính toán các đặc trưng phân tử (molecular features) từ cấu trúc hóa học của các hợp chất. Các đặc trưng này có thể bao gồm:
    - **Molecular Weight:** Khối lượng phân tử.
    - **LogP:** Hệ số phân bố octanol-water (độ tan).
    - **Hydrogen Bond Donors/Acceptors:** Số lượng liên kết hydro cho và nhận.
    - **Topological Polar Surface Area (TPSA):** Diện tích bề mặt cực.
    - **Rings Count:** Số lượng vòng trong phân tử.
    - **Fingerprints:** Biểu diễn cấu trúc phân tử dưới dạng vector bit (ví dụ: Morgan fingerprints).
- **Data Analysis and Modeling (Phân tích và mô hình hóa dữ liệu):**
    - **Exploratory Data Analysis (EDA):** Thăm dò dữ liệu để hiểu rõ hơn về phân phối, mối quan hệ giữa các biến và xác định các xu hướng tiềm năng.
    - **Machine Learning (ML):** Xây dựng các mô hình học máy để dự đoán hoạt tính sinh học của các hợp chất dựa trên các đặc trưng phân tử. Một số thuật toán phổ biến bao gồm:
        - **Regression Models:** Hồi quy tuyến tính, hồi quy Ridge, hồi quy Lasso.
        - **Classification Models:** Logistic Regression, Support Vector Machines (SVM), Random Forest, Gradient Boosting.
- **Model Evaluation and Validation (Đánh giá và xác thực mô hình):** Sử dụng các kỹ thuật đánh giá mô hình phù hợp (ví dụ: cross-validation) để đánh giá hiệu suất của mô hình và đảm bảo tính tổng quát hóa.

2. Hướng dẫn Song ngữ (Bilingual Instructions)

Dưới đây là hướng dẫn chi tiết bằng cả tiếng Anh và tiếng Việt:

English:

1. **Connect to the ChEMBL 35 database using PostgreSQL.** (Kết nối đến cơ sở dữ liệu ChEMBL 35 sử dụng PostgreSQL.)

2. **Write SQL queries to extract relevant data, such as compound information and bioactivity data.** (Viết các truy vấn SQL để trích xuất dữ liệu liên quan, chẳng hạn như thông tin về hợp chất và dữ liệu hoạt tính sinh học.)
3. **Use RDKit in Python to calculate molecular features from the chemical structures.** (Sử dụng RDKit trong Python để tính toán các đặc trưng phân tử từ cấu trúc hóa học.)
4. **Preprocess the data, handling missing values and transforming data as needed.** (Tiền xử lý dữ liệu, xử lý các giá trị bị thiếu và chuyển đổi dữ liệu khi cần thiết.)
5. **Build machine learning models to predict bioactivity based on the molecular features.** (Xây dựng các mô hình học máy để dự đoán hoạt tính sinh học dựa trên các đặc trưng phân tử.)
6. **Evaluate the models using appropriate evaluation metrics and validation techniques.** (Đánh giá các mô hình bằng cách sử dụng các thước đo đánh giá phù hợp và các kỹ thuật xác thực.)

Tiếng Việt:

1. **Kết nối đến cơ sở dữ liệu ChEMBL 35 sử dụng PostgreSQL.**
2. **Viết các truy vấn SQL để trích xuất dữ liệu liên quan, chẳng hạn như thông tin về hợp chất và dữ liệu hoạt tính sinh học.**
3. **Sử dụng RDKit trong Python để tính toán các đặc trưng phân tử từ cấu trúc hóa học.**
4. **Tiền xử lý dữ liệu, xử lý các giá trị bị thiếu và chuyển đổi dữ liệu khi cần thiết.**
5. **Xây dựng các mô hình học máy để dự đoán hoạt tính sinh học dựa trên các đặc trưng phân tử.**
6. **Đánh giá các mô hình bằng cách sử dụng các thước đo đánh giá phù hợp và các kỹ thuật xác thực.**

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

SQL (Lấy 100 dòng dữ liệu):

```sql
-- English
SELECT
    cmp.chembl_id,
    cs.canonical_smiles,
    act.standard_type,
    act.standard_value,
    act.standard_units
FROM
    compound_structures cs
JOIN
    molecule_dictionary md ON cs.molregno = md.molregno
JOIN
    chembl_id_lookup cmp ON md.molregno = cmp.molregno
JOIN
    activities act ON md.molregno = act.molregno
WHERE act.standard_type = 'IC50'
  AND act.standard_units = 'nM'
  AND act.standard_value IS NOT NULL
  AND cs.canonical_smiles IS NOT NULL
LIMIT 100;

-- Vietnamese
-- Lấy thông tin về các hợp chất, cấu trúc, hoạt tính IC50 (nM) từ cơ sở dữ liệu
ChEMBL 35, giới hạn 100 dòng
SELECT
    cmp.chembl_id,  -- ID hợp chất
    cs.canonical_smiles,  -- Cấu trúc SMILES
    act.standard_type,  -- Loại hoạt tính
    act.standard_value,  -- Giá trị hoạt tính
```

```
    act.standard_units   -- Đơn vị hoạt tính
FROM
    compound_structures cs  -- Bảng cấu trúc hợp chất
JOIN
    molecule_dictionary md ON cs.molregno = md.molregno  -- Kết nối với bảng thông tin
phân tử
JOIN
    chembl_id_lookup cmp ON md.molregno = cmp.molregno  -- Kết nối với bảng ID hợp
chất
JOIN
    activities act ON md.molregno = act.molregno  -- Kết nối với bảng hoạt tính
WHERE act.standard_type = 'IC50'  -- Lọc theo loại hoạt tính IC50
  AND act.standard_units = 'nM'  -- Lọc theo đơn vị nM
  AND act.standard_value IS NOT NULL  -- Loại bỏ giá trị NULL
  AND cs.canonical_smiles IS NOT NULL  -- Loại bỏ SMILES NULL
LIMIT 100;  -- Giới hạn 100 dòng
```

Python:

```
# English
import os
import pandas as pd
from rdkit import Chem
from rdkit.Chem import AllChem
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

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

# Load data from CSV
data_path = os.path.join(base_path, "chembl_data.csv")  # Replace "chembl_data.csv"
with your actual filename
df = pd.read_csv(data_path)

# Drop any rows where canonical_smiles or standard_value is NaN
df = df.dropna(subset=['canonical_smiles', 'standard_value'])

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

# Function to calculate molecular features using RDKit
def calculate_features(smiles):
    mol = Chem.MolFromSmiles(smiles)
    if mol is not None:
        mw = Descriptors.MolWt(mol)
        logp = Descriptors.MolLogP(mol)
        hbd = Descriptors.NumHDonors(mol)
        hba = Descriptors.NumHAcceptors(mol)
        tpsa = Descriptors.TPSA(mol)
        rings = Descriptors.RingCount(mol)
        return pd.Series([mw, logp, hbd, hba, tpsa, rings])
    else:
        return pd.Series([None] * 6)

# Apply the function to create new columns
df[['MW', 'LogP', 'HBD', 'HBA', 'TPSA', 'Rings']] =
df['canonical_smiles'].apply(calculate_features)
```

```
# Drop rows with NaN values in features
df = df.dropna(subset=['MW', 'LogP', 'HBD', 'HBA', 'TPSA', 'Rings'])

# Prepare data for machine learning
X = df[['MW', 'LogP', 'HBD', 'HBA', 'TPSA', 'Rings']]
y = df['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
y_pred = model.predict(X_test)

# Calculate mean squared error
mse = mean_squared_error(y_test, y_pred) #squared=False is removed

print(f"Mean Squared Error: {mse}")

#Vietnamese
# Import thư viện cần thiết
import os
import pandas as pd
from rdkit import Chem
from rdkit.Chem import AllChem
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

# Xác định đường dẫn gốc
base_path = "../data"

# Load dữ liệu từ file CSV
data_path = os.path.join(base_path, "chembl_data.csv")  # Thay "chembl_data.csv" bằng
tên file thực tế của bạn
df = pd.read_csv(data_path)

# Loại bỏ các dòng có giá trị NaN trong cột 'canonical_smiles' hoặc 'standard_value'
df = df.dropna(subset=['canonical_smiles', 'standard_value'])

# Chuyển đổi cột 'standard_value' sang kiểu số
df['standard_value'] = pd.to_numeric(df['standard_value'], errors='coerce')

# Hàm tính toán các đặc trưng phân tử sử dụng RDKit
def calculate_features(smiles):
    mol = Chem.MolFromSmiles(smiles)
    if mol is not None:
        mw = Descriptors.MolWt(mol)
        logp = Descriptors.MolLogP(mol)
        hbd = Descriptors.NumHDonors(mol)
        hba = Descriptors.NumHAcceptors(mol)
        tpsa = Descriptors.TPSA(mol)
        rings = Descriptors.RingCount(mol)
```

```
        return pd.Series([mw, logp, hbd, hba, tpsa, rings])
    else:
        return pd.Series([None] * 6)

# Áp dụng hàm để tạo các cột đặc trưng mới
df[['MW', 'LogP', 'HBD', 'HBA', 'TPSA', 'Rings']] =
df['canonical_smiles'].apply(calculate_features)

# Loại bỏ các dòng có giá trị NaN trong các cột đặc trưng
df = df.dropna(subset=['MW', 'LogP', 'HBD', 'HBA', 'TPSA', 'Rings'])

# Chuẩn bị dữ liệu cho học máy
X = df[['MW', 'LogP', 'HBD', 'HBA', 'TPSA', 'Rings']]
y = df['standard_value']

# Chia dữ liệu thành tập huấn luyện và tập kiểm tra
X_train, X_test, y_train, y_test = train_test_split(X, 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 trên tập kiểm tra
y_pred = model.predict(X_test)

# Tính toán sai số bình phương trung bình
mse = mean_squared_error(y_test, y_pred)

print(f"Mean Squared Error: {mse}")
```

Giải thích lỗi và cách khắc phục:

- **Lỗi SQL:** `ERROR: operator does not exist: numeric ~ unknown LINE 12: AND act.standard_value ~ '^[0-9\.]+$'`
    - **Nguyên nhân:** Lỗi này xảy ra do bạn đang cố gắng sử dụng toán tử ~ (regex match) trên một cột kiểu số (`numeric`). Toán tử ~ chỉ hoạt động trên kiểu dữ liệu chuỗi.
    - **Khắc phục:** Loại bỏ điều kiện `AND act.standard_value ~ '^[0-9\.]+$'` hoặc chuyển đổi cột `standard_value` sang kiểu chuỗi trước khi sử dụng toán tử ~. Tuy nhiên, trong trường hợp này, bạn có thể bỏ qua điều kiện này vì bạn đã lọc `act.standard_value IS NOT NULL`, điều này đảm bảo rằng chỉ các giá trị không null được chọn.
- **Lỗi Python:** `phiên bản scikit-learn cũ không hỗ trợ tham số squared=False trong hàm mean_squared_error`
    - **Nguyên nhân:** Tham số `squared=False` trong hàm `mean_squared_error` chỉ được hỗ trợ trong các phiên bản scikit-learn mới hơn.
    - **Khắc phục:** Nâng cấp phiên bản scikit-learn của bạn lên phiên bản mới nhất hoặc loại bỏ tham số `squared=False` để tính toán Mean Squared Error (MSE) thay vì Root Mean Squared Error (RMSE). Tôi đã loại bỏ nó trong code.

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

Dưới đây là 5 ví dụ code SQL và Python mẫu, tập trung vào các khía cạnh khác nhau của phân tích dữ liệu ChEMBL 35:

Ví dụ 1: Trích xuất thông tin cơ bản về hợp chất (Extracting Basic Compound Information)

SQL:

```sql
-- English
SELECT
    cmp.chembl_id,
    cs.canonical_smiles,
    md.pref_name
FROM
    compound_structures cs
JOIN
    molecule_dictionary md ON cs.molregno = md.molregno
JOIN
    chembl_id_lookup cmp ON md.molregno = cmp.molregno
LIMIT 10;

-- Vietnamese
-- Truy vấn để lấy ID ChEMBL, cấu trúc SMILES và tên ưu tiên của 10 hợp chất đầu tiên
SELECT
    cmp.chembl_id,  -- ID ChEMBL
    cs.canonical_smiles,  -- Cấu trúc SMILES
    md.pref_name   -- Tên ưu tiên
FROM
    compound_structures cs  -- Bảng cấu trúc hợp chất
JOIN
    molecule_dictionary md ON cs.molregno = md.molregno  -- Kết nối với bảng thông tin
phân tử
JOIN
    chembl_id_lookup cmp ON md.molregno = cmp.molregno  -- Kết nối với bảng ID hợp
chất
LIMIT 10;  -- Giới hạn 10 dòng
```

Python:

```python
# English
import pandas as pd
import os

base_path = "../data"
data_path = os.path.join(base_path, "basic_compound_info.csv") # Replace
"basic_compound_info.csv" with your filename
df = pd.read_csv(data_path)
print(df.head())

#Vietnamese
import pandas as pd
import os

base_path = "../data"
data_path = os.path.join(base_path, "basic_compound_info.csv") # Thay
"basic_compound_info.csv" bằng tên file của bạn
df = pd.read_csv(data_path)
print(df.head())
```

Ví dụ 2: Lọc hợp chất theo loại hoạt tính (Filtering Compounds by Activity Type)

SQL:

```sql
-- English
SELECT
    cmp.chembl_id,
    cs.canonical_smiles,
    act.standard_type,
    act.standard_value
```

```
FROM
    compound_structures cs
JOIN
    molecule_dictionary md ON cs.molregno = md.molregno
JOIN
    chembl_id_lookup cmp ON md.molregno = cmp.molregno
JOIN
    activities act ON md.molregno = act.molregno
WHERE act.standard_type = 'Ki'
LIMIT 10;

-- Vietnamese
-- Truy vấn để lấy thông tin về các hợp chất có hoạt tính Ki (hằng số ức chế)
SELECT
    cmp.chembl_id,  -- ID ChEMBL
    cs.canonical_smiles,  -- Cấu trúc SMILES
    act.standard_type,  -- Loại hoạt tính
    act.standard_value   -- Giá trị hoạt tính
FROM
    compound_structures cs  -- Bảng cấu trúc hợp chất
JOIN
    molecule_dictionary md ON cs.molregno = md.molregno  -- Kết nối với bảng thông tin
phân tử
JOIN
    chembl_id_lookup cmp ON md.molregno = cmp.molregno  -- Kết nối với bảng ID hợp
chất
JOIN
    activities act ON md.molregno = act.molregno  -- Kết nối với bảng hoạt tính
WHERE act.standard_type = 'Ki'  -- Lọc theo loại hoạt tính Ki
LIMIT 10;  -- Giới hạn 10 dòng
```

Python:

```
# English
import pandas as pd
import os

base_path = "../data"
data_path = os.path.join(base_path, "compounds_with_ki.csv") # Replace
"compounds_with_ki.csv" with your filename
df = pd.read_csv(data_path)
print(df.head())

#Vietnamese
import pandas as pd
import os

base_path = "../data"
data_path = os.path.join(base_path, "compounds_with_ki.csv") # Thay
"compounds_with_ki.csv" bằng tên file của bạn
df = pd.read_csv(data_path)
print(df.head())
```

Ví dụ 3: Tính toán LogP bằng RDKit (Calculating LogP with RDKit)

Python:

```
# English
from rdkit import Chem
from rdkit.Chem import Descriptors
import pandas as pd
```

```
import os

base_path = "../data"
data_path = os.path.join(base_path, "compounds_with_smiles.csv") # Replace
"compounds_with_smiles.csv" with your filename
df = pd.read_csv(data_path)

def calculate_logp(smiles):
    mol = Chem.MolFromSmiles(smiles)
    if mol is not None:
        return Descriptors.MolLogP(mol)
    else:
        return None

df['LogP'] = df['canonical_smiles'].apply(calculate_logp)
print(df.head())

#Vietnamese
from rdkit import Chem
from rdkit.Chem import Descriptors
import pandas as pd
import os

base_path = "../data"
data_path = os.path.join(base_path, "compounds_with_smiles.csv") # Thay
"compounds_with_smiles.csv" bằng tên file của bạn
df = pd.read_csv(data_path)

def calculate_logp(smiles):
    mol = Chem.MolFromSmiles(smiles)
    if mol is not None:
        return Descriptors.MolLogP(mol)
    else:
        return None

df['LogP'] = df['canonical_smiles'].apply(calculate_logp)
print(df.head())
```

Ví dụ 4: Phân tích thống kê mô tả (Descriptive Statistics Analysis)

Python:

```
# English
import pandas as pd
import os

base_path = "../data"
data_path = os.path.join(base_path, "chembl_data_with_logp.csv") # Replace
"chembl_data_with_logp.csv" with your filename
df = pd.read_csv(data_path)

print(df.describe())

#Vietnamese
import pandas as pd
import os

base_path = "../data"
data_path = os.path.join(base_path, "chembl_data_with_logp.csv") # Thay
"chembl_data_with_logp.csv" bằng tên file của bạn
```

```
df = pd.read_csv(data_path)

print(df.describe())
```

Ví dụ 5: Trực quan hóa dữ liệu (Data Visualization)

Python:

```
# English
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import os

base_path = "../data"
data_path = os.path.join(base_path, "chembl_data_with_logp.csv") # Replace
"chembl_data_with_logp.csv" with your filename
df = pd.read_csv(data_path)

sns.scatterplot(x='LogP', y='standard_value', data=df)
plt.show()

#Vietnamese
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import os

base_path = "../data"
data_path = os.path.join(base_path, "chembl_data_with_logp.csv") # Thay
"chembl_data_with_logp.csv" bằng tên file của bạn
df = pd.read_csv(data_path)

sns.scatterplot(x='LogP', y='standard_value', data=df)
plt.show()
```

Lưu ý quan trọng:

- Hãy nhớ thay thế `"chembl_data.csv"`, `"basic_compound_info.csv"`, `"compounds_with_ki.csv"`, `"chembl_data_with_logp.csv"` bằng tên file CSV thực tế của bạn.
- Đảm bảo rằng bạn đã cài đặt tất cả các thư viện cần thiết (RDKit, pandas, scikit-learn, matplotlib, seaborn) trước khi chạy code Python. Bạn có thể cài đặt chúng bằng pip: `pip install rdkit pandas scikit-learn matplotlib seaborn`.
- Điều chỉnh các tham số và thuật toán trong code cho phù hợp với mục tiêu nghiên cứu cụ thể của bạn.

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