PharmApp Suite

🧠 AI for Drug Discovery and Development 💉

🧩 Topic: Dự đoán tiềm năng hiệp đồng thuốc từ ChEMBL

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 R&D dược học, tôi sẽ cung cấp hướng dẫn chi tiết, code mẫu và phân tích để bạn có thể thực hiện dự án một cách hiệu quả.

Tổng quan dự án:

Dự án này tập trung vào việc khai thác dữ liệu từ cơ sở dữ liệu ChEMBL 35 để hỗ trợ quá trình nghiên cứu và phát triển thuốc. Bạn sẽ sử dụng SQL để truy vấn và trích xuất dữ liệu, RDKit để xử lý và phân tích cấu trúc hóa học, và Jupyter Notebook để xây dựng quy trình làm việc.

Cấu trúc thư mục (AIMLOps Template):

Để đảm bảo tính nhất quán và dễ quản lý, chúng ta sẽ tuân theo cấu trúc thư mục chuẩn AIMLOps. Dưới đây là ví dụ về cấu trúc thư mục bạn có thể sử dụng:

```
Topic_CheMBL_35_98/
├── data/               # Chứa dữ liệu CSV trích xuất từ SQL
├── notebooks/          # Chứa các Jupyter Notebook
│   ├── Topic_CheMBL_35_98_1_*.ipynb
│   └── Topic_CheMBL_35_98_2_*.ipynb
├── sql/                # Chứa các file SQL
└── README.md           # Mô tả dự án và hướng dẫn
```

Phân tích mô hình (Analysis Model):

Dựa trên mã chủ đề `Topic_CheMBL_35_98`, tôi sẽ giả định chủ đề này liên quan đến việc xây dựng mô hình QSAR (Quantitative Structure-Activity Relationship) để dự đoán hoạt tính của các hợp chất dựa trên cấu trúc hóa học của chúng.

Các bước thực hiện:

1. **Trích xuất dữ liệu từ ChEMBL 35 (Data Extraction from ChEMBL 35):**
    - Sử dụng SQL để truy vấn cơ sở dữ liệu ChEMBL 35 và trích xuất dữ liệu cần thiết cho mô hình QSAR. Dữ liệu này có thể bao gồm cấu trúc hóa học (SMILES), hoạt tính sinh học (ví dụ: IC50, Ki), và các thuộc tính khác.
    - Lưu dữ liệu trích xuất vào các file CSV trong thư mục `data`.
2. **Tiền xử lý dữ liệu (Data Preprocessing):**
    - Sử dụng RDKit để đọc và xử lý cấu trúc hóa học (SMILES).
    - Tính toán các descriptor phân tử (ví dụ: MW, LogP, HBA, HBD) từ cấu trúc hóa học.
    - Làm sạch và chuẩn hóa dữ liệu hoạt tính sinh học.
    - Chia dữ liệu thành tập huấn luyện và tập kiểm tra.
3. **Xây dựng mô hình QSAR (QSAR Model Building):**
    - Sử dụng các thuật toán học máy (ví dụ: Linear Regression, Random Forest, SVM) để xây dựng mô hình QSAR.
    - Huấn luyện mô hình trên tập huấn luyện.
    - Đánh giá hiệu suất mô hình trên tập kiểm tra.
4. **Đánh giá và tối ưu hóa mô hình (Model Evaluation and Optimization):**
    - Sử dụng các metrics phù hợp (ví dụ: R-squared, RMSE, MAE) để đánh giá hiệu suất mô hình.

    - Tối ưu hóa mô hình bằng cách điều chỉnh các siêu tham số hoặc sử dụng các kỹ thuật feature selection.
5. **Triển khai mô hình (Model Deployment):**
    - Lưu mô hình đã huấn luyện để sử dụng trong tương lai.
    - Xây dựng API hoặc giao diện người dùng để dự đoán hoạt tính của các hợp chất mới.

Hướng dẫn song ngữ (Bilingual Guide):

- **English:** This project aims to build a QSAR model to predict the activity of compounds based on their chemical structures. We will use SQL to extract data from ChEMBL 35, RDKit to process chemical structures, and machine learning algorithms to build the model.
- **Tiếng Việt:** Dự án này nhằm mục đích xây dựng mô hình QSAR để dự đoán hoạt tính của các hợp chất dựa trên cấu trúc hóa học của chúng. Chúng ta sẽ sử dụng SQL để trích xuất dữ liệu từ ChEMBL 35, RDKit để xử lý cấu trúc hóa học, và các thuật toán học máy để xây dựng mô hình.

Code mẫu (Code Examples):

1. SQL (Trích xuất dữ liệu - Data Extraction):

```sql
-- English: Extract 100 compounds with IC50 values for a specific target
-- Tiếng Việt: Trích xuất 100 hợp chất với giá trị IC50 cho một mục tiêu cụ thể
SELECT
    cmp.chembl_id,
    cmp.pref_name,
    md.molfile,
    act.standard_value,
    act.standard_units
FROM
    compound_structures cmp
JOIN
    activities act ON cmp.molregno = act.molregno
JOIN
    assays ass ON act.assay_id = ass.assay_id
JOIN
    target_dictionary td ON ass.tid = td.tid
JOIN
    molecule_dictionary md ON cmp.molregno = md.molregno
WHERE
    td.chembl_id = 'CHEMBL205'  -- Replace with your target CHEMBL_ID (Thay thế bằng
CHEMBL_ID mục tiêu của bạn)
    AND ass.assay_type = 'B'
    AND act.standard_type = 'IC50'
    AND act.standard_relation = '='
    AND act.standard_value IS NOT NULL
    AND act.standard_units = 'nM'
    AND act.standard_value ~ '^[0-9\.]+$' -- Giải quyết lỗi: Lọc các giá trị hợp lệ
LIMIT 100;
```

Lưu ý: Đoạn code `AND act.standard_value ~ '^[0-9\.]+$'` giải quyết lỗi `operator does not exist: numeric ~ unknown`. Nó lọc bỏ các giá trị `standard_value` không phải là số.

2. Python (Đọc dữ liệu và tính toán descriptor - Data Loading and Descriptor Calculation):

```python
# English: Load data from CSV and calculate molecular descriptors using RDKit
# Tiếng Việt: Đọc dữ liệu từ CSV và tính toán các descriptor phân tử bằng RDKit
import pandas as pd
from rdkit import Chem
from rdkit.Chem import Descriptors
import os
```

```
base_path = "."  # Thay đổi nếu cần thiết (Change if needed)
data_path = os.path.join(base_path, "data", "your_data.csv") #Sửa đường dẫn file csv

df = pd.read_csv(data_path)

def calculate_descriptors(smiles):
    mol = Chem.MolFromSmiles(smiles)
    if mol is None:
        return None
    descriptors = {}
    descriptors["MW"] = Descriptors.MolWt(mol)
    descriptors["LogP"] = Descriptors.MolLogP(mol)
    descriptors["HBA"] = Descriptors.NumHAcceptors(mol)
    descriptors["HBD"] = Descriptors.NumHDonors(mol)
    return descriptors

df['descriptors'] = df['molfile'].apply(calculate_descriptors)
df = df.dropna(subset=['descriptors'])  # Loại bỏ các hàng có lỗi (Remove rows with
errors)
df = pd.concat([df.drop(['descriptors'], axis=1), df['descriptors'].apply(pd.Series)],
axis=1)

print(df.head())
```

3. Python (Xây dựng mô hình - Model Building):

```
# English: Build a linear regression model
# Tiếng Việt: Xây dựng mô hình hồi quy tuyến tính
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

# Chuẩn bị dữ liệu (Prepare data)
X = df[["MW", "LogP", "HBA", "HBD"]]  # Chọn các descriptor (Select descriptors)
y = df["standard_value"]  # Chọn giá trị hoạt tính (Select activity values)

# Chia dữ liệu thành tập huấn luyện và tập kiểm tra (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)

# Xây dựng mô hình (Build model)
model = LinearRegression()

# Huấn luyện mô hình (Train model)
model.fit(X_train, y_train)

# Dự đoán trên tập kiểm tra (Predict on test set)
y_pred = model.predict(X_test)

# Đánh giá mô hình (Evaluate 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}")
```

4. Python (Giải quyết lỗi scikit-learn - Addressing scikit-learn error):

```
#Nếu bạn gặp lỗi về squared=False, hãy cập nhật scikit-learn hoặc bỏ tham số này
#If you encounter an error about squared=False, update scikit-learn or remove this
parameter
#Ví dụ:
#Example:
#mse = mean_squared_error(y_test, y_pred, squared=False) #Nếu phiên bản scikit-learn
mới
#mse = mean_squared_error(y_test, y_pred) #Nếu phiên bản scikit-learn cũ
```

5. SQL (Ví dụ truy vấn phức tạp hơn - More complex query example):

```
--English: Extract compounds active against a specific protein target with additional
physicochemical properties
--Tiếng Việt: Trích xuất các hợp chất có hoạt tính chống lại một mục tiêu protein cụ
thể với các thuộc tính lý hóa bổ sung
SELECT
    cmp.chembl_id,
    cmp.pref_name,
    md.molfile,
    act.standard_value,
    act.standard_units,
    props.mw,
    props.alogp
FROM
    compound_structures cmp
JOIN
    activities act ON cmp.molregno = act.molregno
JOIN
    assays ass ON act.assay_id = ass.assay_id
JOIN
    target_dictionary td ON ass.tid = td.tid
JOIN
    molecule_dictionary md ON cmp.molregno = md.molregno
LEFT JOIN
    compound_properties props ON cmp.molregno = props.molregno
WHERE
    td.chembl_id = 'CHEMBL205'
    AND ass.assay_type = 'B'
    AND act.standard_type = 'IC50'
    AND act.standard_relation = '='
    AND act.standard_value IS NOT NULL
    AND act.standard_units = 'nM'
    AND props.mw IS NOT NULL  -- Ensure physicochemical properties are available
LIMIT 100;
```

Năm ví dụ code SQL và Python mẫu (Five SQL and Python Code Examples):

SQL:

1. **Trích xuất số lượng hợp chất cho mỗi mục tiêu (Extract the number of compounds for each target):**

```
SELECT td.chembl_id, td.pref_name, COUNT(DISTINCT cmp.molregno) AS num_compounds
FROM target_dictionary td
JOIN assays ass ON td.tid = ass.tid
JOIN activities act ON ass.assay_id = act.assay_id
JOIN compound_structures cmp ON act.molregno = cmp.molregno
GROUP BY td.chembl_id, td.pref_name
ORDER BY num_compounds DESC
LIMIT 10;
```

2. **Tìm các hợp chất có hoạt tính cao nhất (Find the most potent compounds):**

```sql
SELECT cmp.chembl_id, cmp.pref_name, act.standard_value, act.standard_units
FROM compound_structures cmp
JOIN activities act ON cmp.molregno = act.molregno
WHERE act.standard_type = 'IC50'
AND act.standard_relation = '='
ORDER BY act.standard_value ASC
LIMIT 10;
```

3. **Thống kê các loại assay phổ biến nhất (Statistics on the most common assay types):**

```sql
SELECT ass.assay_type, COUNT(*) AS num_assays
FROM assays ass
GROUP BY ass.assay_type
ORDER BY num_assays DESC;
```

4. **Trích xuất các hợp chất và các thuộc tính vật lý của chúng (Extract compounds and their physical properties):**

```sql
SELECT
    cmp.chembl_id,
    cmp.pref_name,
    md.molfile,
    props.mw,
    props.alogp,
    props.hba,
    props.hbd
FROM
    compound_structures cmp
JOIN
    molecule_dictionary md ON cmp.molregno = md.molregno
LEFT JOIN
    compound_properties props ON cmp.molregno = props.molregno
LIMIT 100;
```

5. **Tìm các mục tiêu liên quan đến một bệnh cụ thể (Find targets associated with a specific disease):**

```sql
--Cần bảng 'target_disease' hoặc tương tự để thực hiện truy vấn này
--Requires a 'target_disease' or similar table to perform this query
--Ví dụ (Example):
--SELECT td.chembl_id, td.pref_name
--FROM target_dictionary td
--JOIN target_disease tdisease ON td.tid = tdisease.tid
--WHERE tdisease.disease_name = 'Cancer';
```

Python:

1. **Đọc dữ liệu từ file CSV và hiển thị thông tin cơ bản (Read data from CSV and display basic information):**

```python
import pandas as pd
import os

base_path = "."
data_path = os.path.join(base_path, "data", "your_data.csv")

df = pd.read_csv(data_path)
print(df.head())
print(df.info())
```

2. **Tính toán và vẽ biểu đồ phân phối của LogP (Calculate and plot the distribution of LogP):**

```python
import pandas as pd
import os
import matplotlib.pyplot as plt
```

```
base_path = "."
data_path = os.path.join(base_path, "data", "your_data.csv")

df = pd.read_csv(data_path)
plt.hist(df['LogP'].dropna(), bins=30)  # Loại bỏ giá trị NaN (Remove NaN values)
plt.xlabel('LogP')
plt.ylabel('Frequency')
plt.title('Distribution of LogP')
plt.show()
```

3. **Tính toán descriptor và vẽ biểu đồ scatter (Calculate descriptors and plot scatter plot):**

```
import pandas as pd
import os
from rdkit import Chem
from rdkit.Chem import Descriptors
import matplotlib.pyplot as plt

base_path = "."
data_path = os.path.join(base_path, "data", "your_data.csv")

df = pd.read_csv(data_path)

def calculate_mw(smiles):
    mol = Chem.MolFromSmiles(smiles)
    if mol:
        return Descriptors.MolWt(mol)
    return None

df['MW'] = df['molfile'].apply(calculate_mw)
df = df.dropna(subset=['MW'])

plt.scatter(df['MW'], df['standard_value'])
plt.xlabel('Molecular Weight (MW)')
plt.ylabel('IC50 (nM)')
plt.title('Scatter Plot of MW vs IC50')
plt.show()
```

4. **Sử dụng Pandas để lọc dữ liệu (Using Pandas to filter data):**

```
import pandas as pd
import os

base_path = "."
data_path = os.path.join(base_path, "data", "your_data.csv")

df = pd.read_csv(data_path)

# Lọc các hợp chất có IC50 < 100 nM (Filter compounds with IC50 < 100 nM)
active_compounds = df[df['standard_value'] < 100]
print(active_compounds.head())
```

5. **Chuyển đổi SMILES sang cấu trúc 2D và lưu (Convert SMILES to 2D structure and save):**

```
from rdkit import Chem
from rdkit.Chem.Draw import MolToFile
import os
import pandas as pd

base_path = "."
data_path = os.path.join(base_path, "data", "your_data.csv")
```

```
df = pd.read_csv(data_path)

def draw_molecule(smiles, filename):
    mol = Chem.MolFromSmiles(smiles)
    if mol:
        MolToFile(mol, filename + ".png")

# Chọn một SMILES từ DataFrame (Select a SMILES from DataFrame)
smiles = df['molfile'][0]
output_filename = "molecule_image"
draw_molecule(smiles, output_filename)
print(f"Image saved to {output_filename}.png")
```

Lưu ý quan trọng:

- Hãy nhớ thay thế `"your_data.csv"` bằng tên file CSV thực tế của bạn.
- Điều chỉnh các tham số và thuật toán trong code mẫu để phù hợp với yêu cầu cụ thể của dự án của bạn.
- Đảm bảo bạn đã cài đặt tất cả các thư viện cần thiết (ví dụ: RDKit, scikit-learn, pandas, matplotlib).
- Kiểm tra và làm sạch dữ liệu của bạn cẩn thận để đảm bảo chất lượng của mô hình.
- Nếu phiên bản Scikit-learn báo lỗi, hãy nâng cấp lên bản mới nhất hoặc hạ cấp và bỏ `squared=False`

Chúc bạn thành công với dự án của mình! Nếu bạn có bất kỳ câu hỏi nào khác, đừng ngần ngại hỏi.