PharmApp Suite

🧠 AI for Drug Discovery and Development 💉

🧩 Topic: Tích hợp dữ liệu tác dụng phụ (SIDER/FAERS)

Tuyệt vời! Tôi rất vui được tham gia vào dự án phân tích dữ liệu ChEMBL 35 của bạn. Với vai trò là một chuyên gia nghiên cứu và phát triển dược học, tôi sẽ giúp bạn xây dựng quy trình phân tích dữ liệu hiệu quả, tập trung vào việc giải quyết các vấn đề bạn gặp phải và cung cấp các ví dụ thực tế để bạn có thể áp dụng vào dự án của mình.

Dưới đây là phân tích chi tiết và hướng dẫn song ngữ cho chủ đề "Topic_CheMBL_35_86", cùng với các ví dụ code SQL và Python mẫu:

1. Phân tích mô hình (Analysis of the Model)

Chủ đề "Topic_CheMBL_35_86" có thể tập trung vào nhiều khía cạnh khác nhau của dữ liệu ChEMBL 35, ví dụ như:

- **Tìm kiếm các hợp chất có hoạt tính sinh học cao đối với một mục tiêu cụ thể:** Xác định các hợp chất có giá trị IC50, Ki, Kd thấp đối với một protein hoặc enzyme cụ thể, từ đó tìm ra các ứng viên tiềm năng cho việc phát triển thuốc.
- **Xây dựng mô hình QSAR (Quantitative Structure-Activity Relationship):** Liên kết cấu trúc hóa học của các hợp chất với hoạt tính sinh học của chúng để dự đoán hoạt tính của các hợp chất mới.
- **Phân tích các đặc tính dược động học (ADMET):** Đánh giá khả năng hấp thụ, phân phối, chuyển hóa và thải trừ của các hợp chất để đảm bảo tính an toàn và hiệu quả của thuốc.
- **Phân tích mối tương quan giữa cấu trúc và hoạt tính (SAR):** Xác định các nhóm chức hoặc cấu trúc con quan trọng cho hoạt tính sinh học của các hợp chất.

Mô hình phân tích chung có thể bao gồm các bước sau:

1. **Thu thập dữ liệu (Data Acquisition):** Truy vấn dữ liệu từ cơ sở dữ liệu ChEMBL 35 bằng SQL.
2. **Tiền xử lý dữ liệu (Data Preprocessing):**
    - Làm sạch dữ liệu: Xử lý các giá trị thiếu, loại bỏ các bản ghi trùng lặp.
    - Chuyển đổi dữ liệu: Chuyển đổi dữ liệu về định dạng phù hợp cho phân tích (ví dụ: chuyển đổi giá trị IC50 thành pIC50).
    - Tính toán các descriptor phân tử: Sử dụng RDKit để tính toán các đặc tính hóa học của các hợp chất.
3. **Phân tích dữ liệu (Data Analysis):**
    - Phân tích thống kê mô tả: Tính toán các thống kê cơ bản như trung bình, độ lệch chuẩn, min, max.
    - Phân tích tương quan: Xác định mối tương quan giữa các descriptor phân tử và hoạt tính sinh học.
    - Xây dựng mô hình học máy: Sử dụng các thuật toán như hồi quy tuyến tính, random forest, SVM để xây dựng mô hình QSAR.
4. **Đánh giá mô hình (Model Evaluation):** Đánh giá hiệu suất của mô hình bằng các chỉ số như R-squared, RMSE, MAE.
5. **Trực quan hóa dữ liệu (Data Visualization):** Sử dụng các biểu đồ để trực quan hóa kết quả phân tích.

2. Hướng dẫn song ngữ (Bilingual Guide)

2.1. Kết nối cơ sở dữ liệu ChEMBL 35 (Connecting to the ChEMBL 35 Database)

Tiếng Việt:

Để kết nối đến cơ sở dữ liệu ChEMBL 35, bạn có thể sử dụng thư viện `psycopg2` trong Python. Hãy đảm bảo rằng bạn đã cài đặt thư viện này trước khi thực hiện kết nối.

Tiếng Anh:

To connect to the ChEMBL 35 database, you can use the `psycopg2` library in Python. Make sure you have installed this library before making the connection.

```
import psycopg2

# Database credentials
db_params = {
    'host': '192.168.206.136',
    'user': 'rd',
    'password': 'rd',
    'database': 'chembl_35'
}

# Establish connection
try:
    conn = psycopg2.connect(**db_params)
    cursor = conn.cursor()
    print("Connected to ChEMBL 35 database successfully!")
except Exception as e:
    print(f"Error connecting to the database: {e}")
```

2.2. Khắc phục lỗi SQL (Fixing the SQL Error)

Tiếng Việt:

Lỗi "ERROR: operator does not exist: numeric ~ unknown" xảy ra do bạn đang cố gắng so sánh một cột kiểu số (`numeric`) với một chuỗi (`unknown`). Để khắc phục, bạn cần chuyển đổi cột `standard_value` sang kiểu số trước khi so sánh. Bạn có thể sử dụng hàm `CAST` để thực hiện việc này.

Tiếng Anh:

The error "ERROR: operator does not exist: numeric ~ unknown" occurs because you are trying to compare a numeric column (`numeric`) with a string (`unknown`). To fix this, you need to convert the `standard_value` column to a numeric type before comparison. You can use the `CAST` function to do this.

Ví dụ:

```
SELECT *
FROM activities act
WHERE act.standard_type = 'IC50'
  AND act.standard_relation = '='
  AND act.standard_value IS NOT NULL
  AND CAST(act.standard_value AS TEXT) ~ '^[0-9\.]+$' -- Check if it's a valid number
  AND CAST(act.standard_value AS NUMERIC) <= 100; -- Filter values <= 100
LIMIT 100;
```

2.3. Sửa lỗi phiên bản Scikit-learn cũ (Fixing Scikit-learn Version Error)

Tiếng Việt:

Nếu bạn đang sử dụng phiên bản Scikit-learn cũ, bạn có thể gặp lỗi khi sử dụng `mean_squared_error` với tham số `squared=False`. Để khắc phục, bạn có thể nâng cấp Scikit-learn lên phiên bản mới nhất hoặc tính căn bậc hai của kết quả `mean_squared_error` theo cách thủ công.

Tiếng Anh:

If you are using an older version of Scikit-learn, you may encounter an error when using `mean_squared_error` with the `squared=False` parameter. To fix this, you can upgrade Scikit-learn to the latest version or manually calculate the square root of the `mean_squared_error` result.

```python
from sklearn.metrics import mean_squared_error
import numpy as np

# Example predictions and actual values
y_true = [3, -0.5, 2, 7]
y_predicted = [2.5, 0.0, 2, 8]

# Calculate Mean Squared Error
mse = mean_squared_error(y_true, y_predicted)

# Calculate Root Mean Squared Error (RMSE) manually if squared=False is not supported
rmse = np.sqrt(mse)

print(f"MSE: {mse}")
print(f"RMSE: {rmse}")
```

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

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

Ví dụ 1: Truy vấn các hợp chất có IC50 < 100 nM đối với một mục tiêu cụ thể (Querying compounds with IC50 < 100 nM for a specific target)

SQL:

```sql
SELECT
    act.molregno,
    cmp.chembl_id,
    act.standard_value,
    act.standard_units
FROM activities act
JOIN molecule_dictionary cmp ON act.molregno = cmp.molregno
WHERE act.standard_type = 'IC50'
  AND act.standard_relation = '='
  AND act.standard_units = 'nM'
  AND act.target_chembl_id = 'CHEMBL205' -- Replace with your target CHEMBL_ID
  AND CAST(act.standard_value AS TEXT) ~ '^[0-9\.]+$'
  AND CAST(act.standard_value AS NUMERIC) <= 100
LIMIT 100;
```

Python:

```python
import psycopg2
import pandas as pd

# Database credentials (same as before)
db_params = {
    'host': '192.168.206.136',
    'user': 'rd',
    'password': 'rd',
    'database': 'chembl_35'
}

# SQL query
sql_query = """
```

```
SELECT
    act.molregno,
    cmp.chembl_id,
    act.standard_value,
    act.standard_units
FROM activities act
JOIN molecule_dictionary cmp ON act.molregno = cmp.molregno
WHERE act.standard_type = 'IC50'
  AND act.standard_relation = '='
  AND act.standard_units = 'nM'
  AND act.target_chembl_id = 'CHEMBL205' -- Replace with your target CHEMBL_ID
  AND CAST(act.standard_value AS TEXT) ~ '^[0-9\.]+$'
  AND CAST(act.standard_value AS NUMERIC) <= 100
LIMIT 100;
"""

# Execute query and load into pandas DataFrame
try:
    conn = psycopg2.connect(**db_params)
    df = pd.read_sql_query(sql_query, conn)
    print(df.head())
except Exception as e:
    print(f"Error executing query: {e}")
finally:
    if conn:
        conn.close()
```

Ví dụ 2: Tính toán các descriptor phân tử sử dụng RDKit (Calculating molecular descriptors using RDKit)

Python:

```
import psycopg2
import pandas as pd
from rdkit import Chem
from rdkit.Chem import Descriptors

# Database credentials (same as before)
db_params = {
    'host': '192.168.206.136',
    'user': 'rd',
    'password': 'rd',
    'database': 'chembl_35'
}

# SQL query to retrieve SMILES strings
sql_query = """
SELECT
    cmp.chembl_id,
    cmp.smiles
FROM molecule_dictionary cmp
LIMIT 100;
"""

# Execute query and load into pandas DataFrame
try:
    conn = psycopg2.connect(**db_params)
    df = pd.read_sql_query(sql_query, conn)
except Exception as e:
    print(f"Error executing query: {e}")
finally:
```

```
    if conn:
        conn.close()

# Function to calculate molecular weight
def calculate_mw(smiles):
    mol = Chem.MolFromSmiles(smiles)
    if mol:
        return Descriptors.MolWt(mol)
    else:
        return None

# Apply the function to the SMILES column
df['mol_weight'] = df['smiles'].apply(calculate_mw)

print(df.head())
```

Ví dụ 3: Phân tích thống kê mô tả (Descriptive statistical analysis)

Python:

```
import psycopg2
import pandas as pd

# Database credentials (same as before)
db_params = {
    'host': '192.168.206.136',
    'user': 'rd',
    'password': 'rd',
    'database': 'chembl_35'
}

# SQL query to retrieve IC50 values
sql_query = """
SELECT
    act.standard_value
FROM activities act
WHERE act.standard_type = 'IC50'
  AND act.standard_relation = '='
  AND act.standard_units = 'nM'
  AND CAST(act.standard_value AS TEXT) ~ '^[0-9\.]+$'
LIMIT 100;
"""

# Execute query and load into pandas DataFrame
try:
    conn = psycopg2.connect(**db_params)
    df = pd.read_sql_query(sql_query, conn)
except Exception as e:
    print(f"Error executing query: {e}")
finally:
    if conn:
        conn.close()

# Convert standard_value to numeric (handling potential errors)
df['standard_value'] = pd.to_numeric(df['standard_value'], errors='coerce')
df = df.dropna(subset=['standard_value'])  # Remove rows with NaN

# Descriptive statistics
print(df['standard_value'].describe())
```

Ví dụ 4: Xây dựng mô hình hồi quy tuyến tính (Building a linear regression model)

Python:

```python
import psycopg2
import pandas as pd
from rdkit import Chem
from rdkit.Chem import Descriptors
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
import numpy as np

# Database credentials (same as before)
db_params = {
    'host': '192.168.206.136',
    'user': 'rd',
    'password': 'rd',
    'database': 'chembl_35'
}

# SQL query to retrieve SMILES and IC50 values
sql_query = """
SELECT
    cmp.smiles,
    act.standard_value
FROM activities act
JOIN molecule_dictionary cmp ON act.molregno = cmp.molregno
WHERE act.standard_type = 'IC50'
  AND act.standard_relation = '='
  AND act.standard_units = 'nM'
  AND CAST(act.standard_value AS TEXT) ~ '^[0-9\.]+$'
LIMIT 100;
"""

# Execute query and load into pandas DataFrame
try:
    conn = psycopg2.connect(**db_params)
    df = pd.read_sql_query(sql_query, conn)
except Exception as e:
    print(f"Error executing query: {e}")
finally:
    if conn:
        conn.close()

# Function to calculate molecular weight
def calculate_mw(smiles):
    mol = Chem.MolFromSmiles(smiles)
    if mol:
        return Descriptors.MolWt(mol)
    else:
        return None

# Apply the function to the SMILES column
df['mol_weight'] = df['smiles'].apply(calculate_mw)

# Convert standard_value to numeric and drop rows with NaN
df['standard_value'] = pd.to_numeric(df['standard_value'], errors='coerce')
df = df.dropna(subset=['standard_value', 'mol_weight'])
```

```
# Prepare data for linear regression
X = df[['mol_weight']]
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)

# Create and train the linear regression model
model = LinearRegression()
model.fit(X_train, y_train)

# Make predictions on the test set
y_pred = model.predict(X_test)

# Evaluate the model
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)

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

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

Python:

```
import psycopg2
import pandas as pd
import matplotlib.pyplot as plt

# Database credentials (same as before)
db_params = {
    'host': '192.168.206.136',
    'user': 'rd',
    'password': 'rd',
    'database': 'chembl_35'
}

# SQL query to retrieve IC50 values
sql_query = """
SELECT
    act.standard_value
FROM activities act
WHERE act.standard_type = 'IC50'
  AND act.standard_relation = '='
  AND act.standard_units = 'nM'
  AND CAST(act.standard_value AS TEXT) ~ '^[0-9\.]+$'
LIMIT 100;
"""

# Execute query and load into pandas DataFrame
try:
    conn = psycopg2.connect(**db_params)
    df = pd.read_sql_query(sql_query, conn)
except Exception as e:
    print(f"Error executing query: {e}")
finally:
    if conn:
        conn.close()
```

```
# Convert standard_value to numeric and drop rows with NaN
df['standard_value'] = pd.to_numeric(df['standard_value'], errors='coerce')
df = df.dropna(subset=['standard_value'])

# Create a histogram of IC50 values
plt.hist(df['standard_value'], bins=20)
plt.xlabel('IC50 (nM)')
plt.ylabel('Frequency')
plt.title('Distribution of IC50 Values')
plt.show()
```

Lưu ý quan trọng:

- Hãy thay thế `'CHEMBL205'` trong các truy vấn SQL bằng CHEMBL_ID của mục tiêu bạn quan tâm.
- Các ví dụ trên chỉ là điểm khởi đầu. Bạn có thể tùy chỉnh chúng để phù hợp với nhu cầu cụ thể của dự án của bạn.
- Đảm bảo rằng bạn đã cài đặt tất cả các thư viện cần thiết (psycopg2, RDKit, scikit-learn, pandas, matplotlib) trước khi chạy code.

Hy vọng những hướng dẫn và ví dụ này sẽ giúp bạn tiến xa hơn trong dự án phân tích dữ liệu ChEMBL 35 của mình. Nếu bạn có bất kỳ câu hỏi nào khác, đừng ngần ngại hỏi!