PharmApp Suite

🧠 AI for Drug Discovery and Development 🍼

🧩 Topic: Dự đoán LogP và TPSA của các hợp chất ChEMBL

Tuyệt vời! Tôi sẽ giúp bạn phân tích, hướng dẫn và cung cấp code mẫu cho dự án phân tích dữ liệu ChEMBL 35 kết hợp RDKit, tuân theo các quy tắc và yêu cầu bạn đã đặt ra.

Tổng quan dự án:

Dự án này tập trung vào việc phân tích dữ liệu từ cơ sở dữ liệu ChEMBL 35, sử dụng RDKit để tính toán các thuộc tính phân tử, và PostgreSQL để truy vấn dữ liệu. Mục tiêu là hỗ trợ quá trình nghiên cứu và phát triển thuốc.

1. Phân tích mô hình (Analysis Model):

Mô hình phân tích của bạn sẽ bao gồm các bước sau:

- **Trích xuất dữ liệu (Data Extraction):** Sử dụng SQL để truy vấn dữ liệu từ cơ sở dữ liệu ChEMBL 35. Dữ liệu này có thể bao gồm thông tin về các hợp chất, hoạt tính sinh học, mục tiêu, v.v.
- **Tiền xử lý dữ liệu (Data Preprocessing):** Làm sạch và chuẩn hóa dữ liệu. Loại bỏ các giá trị bị thiếu hoặc không hợp lệ.
- **Tính toán thuộc tính (Feature Calculation):** Sử dụng RDKit để tính toán các thuộc tính phân tử từ cấu trúc hóa học của các hợp chất. Các thuộc tính này có thể bao gồm trọng lượng phân tử, logP, số lượng vòng, v.v.
- **Phân tích thống kê và mô hình hóa (Statistical Analysis and Modeling):** Sử dụng các kỹ thuật thống kê và học máy để khám phá các mối quan hệ giữa các thuộc tính phân tử và hoạt tính sinh học. Ví dụ: bạn có thể xây dựng mô hình hồi quy để dự đoán hoạt tính sinh học của một hợp chất dựa trên các thuộc tính phân tử của nó.
- **Trực quan hóa dữ liệu (Data Visualization):** Sử dụng các biểu đồ và đồ thị để trực quan hóa dữ liệu và kết quả phân tích.

2. Hướng dẫn song ngữ (Bilingual Guide):

2.1 SQL (English/Vietnamese):

- **Purpose:** Retrieve data from the ChEMBL 35 database. / **Mục đích:** Truy xuất dữ liệu từ cơ sở dữ liệu ChEMBL 35.
- **Example:**

```sql
-- English
-- Select the molecule ChEMBL ID, standard value, and standard units for compounds with a specific activity type.
-- Lọc lấy ChEMBL ID của phân tử, giá trị chuẩn và đơn vị chuẩn cho các hợp chất có loại hoạt tính cụ thể.
SELECT
    md.chembl_id,
    act.standard_value,
    act.standard_units
FROM
    molecule_dictionary md
JOIN
    activities act ON md.molregno = act.molregno
WHERE
```

```
    act.standard_type = 'IC50'
LIMIT 100;
```

2.2 Python (English/Vietnamese):

- **Purpose:** Calculate molecular properties using RDKit and perform statistical analysis. / **Mục đích:** Tính toán các thuộc tính phân tử bằng RDKit và thực hiện phân tích thống kê.
- **Example:**

```python
# English
# Import necessary libraries
# Import các thư viện cần thiết
import pandas as pd
from rdkit import Chem
from rdkit.Chem import Descriptors
import numpy as np

# Function to calculate molecular weight
# Hàm tính toán trọng lượng phân tử
def calculate_mw(smiles):
    mol = Chem.MolFromSmiles(smiles)
    if mol:
        return Descriptors.MolWt(mol)
    else:
        return None

# Example usage: Assuming you have a DataFrame named 'df' with a column 'smiles'
# Ví dụ sử dụng: Giả sử bạn có một DataFrame tên là 'df' với một cột 'smiles'
# df['molecular_weight'] = df['smiles'].apply(calculate_mw)

# Example of handling missing values
# Ví dụ về xử lý các giá trị bị thiếu
# df = df.dropna(subset=['molecular_weight'])
```

3. Code mẫu (Code Examples):

Dưới đây là 5 ví dụ về code SQL và Python mẫu:

3.1 Ví dụ 1: Truy vấn dữ liệu cơ bản (Basic Data Query)

- **SQL:**

```sql
-- English
-- Select the ChEMBL ID and preferred name of the first 100 molecules.
-- Chọn ChEMBL ID và tên ưu tiên của 100 phân tử đầu tiên.
SELECT chembl_id, pref_name FROM molecule_dictionary LIMIT 100;
```

- **Python:**

```python
# English
# Import pandas
# Nhập pandas
import pandas as pd

# Function to read data from a CSV file
# Hàm đọc dữ liệu từ file CSV
def read_data(file_path):
    df = pd.read_csv(file_path)
    return df

# Example usage
# Ví dụ sử dụng
# file_path = os.path.join(base_path, 'data', 'molecule_dictionary.csv')
```

```
# df = read_data(file_path)
# print(df.head())
```

3.2 Ví dụ 2: Tính toán logP (Calculating LogP)

- **Python:**

```
# English
# Import RDKit libraries
# Nhập các thư viện RDKit
from rdkit import Chem
from rdkit.Chem import Descriptors

# Function to calculate LogP
# Hàm tính toán LogP
def calculate_logp(smiles):
    mol = Chem.MolFromSmiles(smiles)
    if mol:
        return Descriptors.MolLogP(mol)
    else:
        return None

# Example usage
# Ví dụ sử dụng
# df['logp'] = df['smiles'].apply(calculate_logp)
# print(df.head())
```

3.3 Ví dụ 3: Lọc dữ liệu hoạt tính (Filtering Activity Data)

- **SQL:**

```
-- English
-- Select activity data for a specific target protein (e.g., with target_chembl_id
'CHEMBL205').
-- Chọn dữ liệu hoạt tính cho một protein mục tiêu cụ thể (ví dụ: với target_chembl_id
'CHEMBL205').
SELECT act.standard_value, act.standard_units
FROM activities act
JOIN target_dictionary td ON act.tid = td.tid
WHERE td.target_chembl_id = 'CHEMBL205'
  AND act.standard_type = 'IC50'
LIMIT 100;
```

- **Python:**

```
# English
# Example of filtering data in pandas
# Ví dụ về lọc dữ liệu trong pandas
# df_filtered = df[(df['standard_type'] == 'IC50') & (df['standard_units'] == 'nM')]
# print(df_filtered.head())
```

3.4 Ví dụ 4: Kết hợp dữ liệu từ nhiều bảng (Joining Data from Multiple Tables)

- **SQL:**

```
-- English
-- Select molecule information and activity data by joining molecule_dictionary and
activities tables.
-- Chọn thông tin phân tử và dữ liệu hoạt tính bằng cách kết hợp bảng
molecule_dictionary và activities.
SELECT md.chembl_id, md.pref_name, act.standard_value, act.standard_units
FROM molecule_dictionary md
JOIN activities act ON md.molregno = act.molregno
LIMIT 100;
```

3.5 Ví dụ 5: Phân tích thống kê đơn giản (Simple Statistical Analysis)

- **Python:**

```python
# English
# Import numpy for statistical calculations
# Nhập numpy để tính toán thống kê
import numpy as np

# Example of calculating the mean of a column
# Ví dụ về tính trung bình của một cột
# mean_value = np.mean(df['standard_value'])
# print(f"Mean standard value: {mean_value}")

# Example of calculating the standard deviation
# Ví dụ về tính độ lệch chuẩn
# std_value = np.std(df['standard_value'])
# print(f"Standard deviation of standard value: {std_value}")
```

4. Giải quyết lỗi (Error Resolution):

- **Lỗi a: `ERROR: operator does not exist: numeric ~ unknown LINE 12: AND act.standard_value ~ '^[0-9\.]+$'`**
    - **Nguyên nhân:** Lỗi này xảy ra vì bạn đang cố gắng sử dụng toán tử ~ (regular expression match) trên một cột kiểu `numeric`. Toán tử này chỉ hoạt động trên kiểu dữ liệu `text`.
    - **Giải pháp:** Chuyển đổi cột `standard_value` sang kiểu `text` trước khi sử dụng toán tử ~.

```sql
-- English
-- Convert the standard_value column to text before using the regular expression match.
-- Chuyển đổi cột standard_value sang kiểu text trước khi sử dụng regular expression match.
SELECT
    md.chembl_id,
    act.standard_value,
    act.standard_units
FROM
    molecule_dictionary md
JOIN
    activities act ON md.molregno = act.molregno
WHERE
    act.standard_type = 'IC50'
    AND CAST(act.standard_value AS TEXT) ~ '^[0-9\.]+$'
LIMIT 100;
```

- **Lỗi b: Phiên bản scikit-learn cũ không hỗ trợ tham số `squared=False` trong hàm `mean_squared_error`**
    - **Nguyên nhân:** Tham số `squared=False` được thêm vào hàm `mean_squared_error` trong các phiên bản scikit-learn mới hơn.
    - **Giải pháp:** Cập nhật phiên bản scikit-learn của bạn lên phiên bản mới nhất, hoặc loại bỏ tham số `squared=False` và tự tính căn bậc hai của kết quả.

```python
# English
# Option 1: Update scikit-learn
# pip install -U scikit-learn

# Option 2: Remove squared=False and calculate the square root manually
# from sklearn.metrics import mean_squared_error
# mse = mean_squared_error(y_true, y_predicted)
```

```
# rmse = mse**0.5
# print(f"RMSE: {rmse}")
```

5. Lưu ý quan trọng:

- **Đường dẫn:** Sử dụng `os.path.join(base_path, 'data', 'your_file.csv')` để đảm bảo tính di động của code.
- **Giới hạn dữ liệu:** Chỉ lấy 100 dòng dữ liệu cho mục đích thử nghiệm và phát triển. Khi triển khai, bạn có thể tăng số lượng dòng hoặc sử dụng toàn bộ dữ liệu.
- **Kiểm tra dữ liệu:** Luôn kiểm tra dữ liệu sau khi trích xuất và tiền xử lý để đảm bảo tính chính xác và đầy đủ.
- **Tài liệu:** Đọc kỹ tài liệu của RDKit và scikit-learn để hiểu rõ các hàm và tham số.

Tôi hy vọng những hướng dẫn và ví dụ này sẽ giúp bạn trong quá trình phân tích dữ liệu ChEMBL 35. Nếu bạn có bất kỳ câu hỏi nào khác, đừng ngần ngại hỏi!