PharmApp Suite

🧠 AI for Drug Discovery and Development 🧪

🧩 Topic: Ánh xạ gene – hợp chất với lớp phủ hoạt tính

Tuyệt vời! Tôi rất vui được hỗ trợ bạn trong dự án phân tích dữ liệu Chembl 35 kết hợp RDKit. Dựa trên thông tin bạn cung cấp, tôi sẽ giúp bạn phân tích, xây dựng mô hình, cung cấp code mẫu SQL và Python (song ngữ) để bạn có thể bắt đầu dự án một cách hiệu quả.

Tổng quan dự án

Dự án của bạn tập trung vào việc sử dụng dữ liệu từ Chembl 35, một cơ sở dữ liệu hóa học lớn, kết hợp với RDKit, một bộ công cụ hóa tin sinh học mạnh mẽ, để hỗ trợ quá trình nghiên cứu và phát triển thuốc. Bạn sẽ sử dụng PostgreSQL (psql) để truy vấn dữ liệu Chembl 35, sau đó sử dụng Jupyter Notebook và Python (với RDKit và scikit-learn) để phân tích và xây dựng mô hình.

1. Phân tích mô hình phân tích (Analysis Model Analysis)

Dựa trên mã chủ đề "Topic_CheMBL_35_87", tôi sẽ đề xuất một mô hình phân tích tập trung vào việc dự đoán hoạt tính sinh học của các hợp chất dựa trên cấu trúc hóa học của chúng. Cụ thể, chúng ta sẽ thực hiện các bước sau:

- **Trích xuất dữ liệu (Data Extraction):** Sử dụng SQL để truy vấn dữ liệu từ Chembl 35, tập trung vào các thông tin như cấu trúc SMILES, giá trị hoạt tính (ví dụ: IC50, Ki), mục tiêu (target) của hợp chất.
- **Tiền xử lý dữ liệu (Data Preprocessing):**
    - Làm sạch dữ liệu: Loại bỏ các giá trị bị thiếu, các hợp chất không có cấu trúc SMILES hợp lệ.
    - Chuyển đổi dữ liệu: Chuyển đổi giá trị hoạt tính về dạng pChEMBL (ví dụ: -log10(IC50)).
- **Tính toán đặc trưng (Feature Calculation):** Sử dụng RDKit để tính toán các đặc trưng hóa học (chemical descriptors) từ cấu trúc SMILES của các hợp chất. Các đặc trưng này có thể bao gồm:
    - Số lượng liên kết, nguyên tử.
    - Khối lượng phân tử.
    - LogP (hệ số phân vùng octanol-nước).
    - Các đặc trưng hình thái (topological features).
    - Fingerprints (ví dụ: Morgan fingerprints).
- **Xây dựng mô hình (Model Building):** Sử dụng các thuật toán học máy (machine learning) để xây dựng mô hình dự đoán hoạt tính sinh học. Một số thuật toán phù hợp bao gồm:
    - Hồi quy tuyến tính (Linear Regression).
    - Rừng ngẫu nhiên (Random Forest).
    - Máy vectơ hỗ trợ (Support Vector Machines).
    - Mạng nơ-ron (Neural Networks).
- **Đánh giá mô hình (Model Evaluation):** Sử dụng các chỉ số đánh giá phù hợp để đánh giá hiệu suất của mô hình, chẳng hạn như:
    - Mean Squared Error (MSE).
    - Root Mean Squared Error (RMSE).
    - R-squared (R2).
    - Area Under the ROC Curve (AUC).

- **Triển khai mô hình (Model Deployment):** Sau khi có mô hình tốt, bạn có thể sử dụng nó để dự đoán hoạt tính của các hợp chất mới, hỗ trợ quá trình sàng lọc ảo (virtual screening) và thiết kế thuốc.

2. Hướng dẫn song ngữ (Bilingual Instructions)

Dưới đây là hướng dẫn chi tiết (song ngữ) cho từng bước trong quy trình phân tích:

Bước 1: Trích xuất dữ liệu từ Chembl 35 (Data Extraction from Chembl 35)

- **English:** Use SQL to query the Chembl 35 database to extract relevant information such as SMILES structures, activity values (e.g., IC50, Ki), and target information.
- **Tiếng Việt:** Sử dụng SQL để truy vấn cơ sở dữ liệu Chembl 35 để trích xuất thông tin liên quan như cấu trúc SMILES, giá trị hoạt tính (ví dụ: IC50, Ki) và thông tin về mục tiêu (target).

Bước 2: Tiền xử lý dữ liệu (Data Preprocessing)

- **English:** Clean the data by removing missing values and invalid SMILES structures. Convert activity values to pChEMBL format (e.g., -log10(IC50)).
- **Tiếng Việt:** Làm sạch dữ liệu bằng cách loại bỏ các giá trị bị thiếu và các cấu trúc SMILES không hợp lệ. Chuyển đổi giá trị hoạt tính sang định dạng pChEMBL (ví dụ: -log10(IC50)).

Bước 3: Tính toán đặc trưng (Feature Calculation)

- **English:** Use RDKit to calculate chemical descriptors from the SMILES structures of the compounds. These descriptors can include the number of bonds, atoms, molecular weight, LogP, topological features, and fingerprints (e.g., Morgan fingerprints).
- **Tiếng Việt:** Sử dụng RDKit để tính toán các đặc trưng hóa học từ cấu trúc SMILES của các hợp chất. Các đặc trưng này có thể bao gồm số lượng liên kết, nguyên tử, khối lượng phân tử, LogP, các đặc trưng hình thái và fingerprints (ví dụ: Morgan fingerprints).

Bước 4: Xây dựng và đánh giá mô hình (Model Building and Evaluation)

- **English:** Use machine learning algorithms (e.g., Linear Regression, Random Forest, SVM, Neural Networks) to build a predictive model. Evaluate the model's performance using appropriate metrics such as MSE, RMSE, R2, and AUC.
- **Tiếng Việt:** Sử dụng các thuật toán học máy (ví dụ: Hồi quy tuyến tính, Rừng ngẫu nhiên, SVM, Mạng nơ-ron) để xây dựng mô hình dự đoán. Đánh giá hiệu suất của mô hình bằng cách sử dụng các chỉ số phù hợp như MSE, RMSE, R2 và AUC.

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

Dưới đây là các ví dụ code SQL và Python để bạn có thể bắt đầu dự án của mình:

Ví dụ 1: Trích xuất dữ liệu từ Chembl 35 (SQL)

```sql
-- English: Extract SMILES and standard_value from the activities table for a specific target.
-- Tiếng Việt: Trích xuất SMILES và standard_value từ bảng activities cho một mục tiêu cụ thể.
SELECT
    md.chembl_id,
    cs.canonical_smiles,
    act.standard_value,
    act.standard_units
FROM
    activities act
JOIN
    molecule_dictionary md ON act.molregno = md.molregno
JOIN
    compound_structures cs ON md.molregno = cs.molregno
```

```
WHERE
    act.standard_type = 'IC50'
    AND act.standard_units = 'nM'
    AND act.standard_relation = '='
    AND act.target_chembl_id = 'CHEMBL205' -- Replace with your target of interest
    AND act.standard_value IS NOT NULL
    AND act.standard_value > 0
    AND act.standard_value ~ '^[0-9\.]+$' -- Corrected regex
LIMIT 100;
```

Lưu ý sửa lỗi: Lỗi `ERROR: operator does not exist: numeric ~ unknown` xảy ra do bạn đang cố gắng sử dụng toán tử ~ (regular expression match) trên một cột kiểu số (numeric). Để khắc phục, bạn cần đảm bảo rằng cột `act.standard_value` được ép kiểu về kiểu text trước khi sử dụng toán tử ~. Ngoài ra, biểu thức chính quy (regular expression) `'^[0-9\.]+$'` cần được sửa để đảm bảo nó khớp với các giá trị số hợp lệ.

Ví dụ 2: Tính toán đặc trưng với RDKit (Python)

```
# English: Calculate Morgan fingerprints using RDKit.
# Tiếng Việt: Tính toán Morgan fingerprints sử dụng RDKit.
import os
import pandas as pd
from rdkit import Chem
from rdkit.Chem import AllChem
import numpy as np

base_path = "../data"  # adjust path if needed
csv_file = "chembl_data.csv"  # replace with your csv file name
file_path = os.path.join(base_path, csv_file)

df = pd.read_csv(file_path)

def calculate_morgan_fingerprint(smiles, radius=2, nBits=2048):
    """Calculates the Morgan fingerprint for a given SMILES string."""
    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

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

# Remove rows with None values in 'morgan_fp'
df = df.dropna(subset=['morgan_fp'])

# Convert 'morgan_fp' column to a list of lists
df['morgan_fp'] = df['morgan_fp'].apply(lambda x: x.tolist())

# Split the 'morgan_fp' column into separate columns
fp_df = pd.DataFrame(df['morgan_fp'].tolist(), index=df.index)
df = pd.concat([df, fp_df], axis=1)

print(df.head())
```

Ví dụ 3: Xây dựng mô hình hồi quy tuyến tính (Python)

```python
# English: Build a linear regression model using scikit-learn.
# Tiếng Việt: Xây dựng mô hình hồi quy tuyến tính sử dụng scikit-learn.
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.preprocessing import MinMaxScaler
import numpy as np

# Assuming 'df' is your DataFrame with features and target variable

# Convert standard_value to pChEMBL
df['pchembl_value'] = -np.log10(df['standard_value'] * 1e-9)  # Convert nM to M

# Prepare data for model training
X = df.iloc[:, df.columns.get_loc('morgan_fp') + 1:].values  # Use Morgan fingerprints
as features
y = df['pchembl_value'].values  # pChEMBL values as target

# Scale features
scaler = MinMaxScaler()
X_scaled = scaler.fit_transform(X)

# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X_scaled, 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)
r2 = r2_score(y_test, y_pred)

print(f"Mean Squared Error: {mse}")
print(f"R-squared: {r2}")
```

Lưu ý: Nếu bạn gặp lỗi liên quan đến tham số `squared=False` trong hàm `mean_squared_error`, hãy cập nhật phiên bản scikit-learn của bạn lên phiên bản mới nhất.

Ví dụ 4: Lưu trữ mô hình (Python)

```python
# English: Save the trained model using pickle.
# Tiếng Việt: Lưu trữ mô hình đã huấn luyện sử dụng pickle.
import pickle

# Save the model to a file
filename = 'linear_regression_model.pkl'
pickle.dump(model, open(filename, 'wb'))

# Load the model from the file
loaded_model = pickle.load(open(filename, 'rb'))

# Use the loaded model to make predictions
y_pred = loaded_model.predict(X_test)
```

Ví dụ 5: Sử dụng Pipeline để tiền xử lý và xây dựng mô hình (Python)

```python
# English: Use a Pipeline to streamline preprocessing and model building.
# Tiếng Việt: Sử dụng Pipeline để đơn giản hóa quá trình tiền xử lý và xây dựng mô hình.
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestRegressor

# Create a pipeline
pipeline = Pipeline([
    ('scaler', StandardScaler()),  # Standardize features
    ('model', RandomForestRegressor(random_state=42))  # Random Forest model
])

# 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 the pipeline
pipeline.fit(X_train, y_train)

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

# Evaluate the 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}")
```

Lưu ý quan trọng:

- **Đường dẫn:** Đảm bảo rằng bạn đã điều chỉnh đường dẫn `base_path` và tên file CSV cho phù hợp với cấu trúc thư mục của bạn.
- **Cài đặt thư viện:** Đảm bảo rằng bạn đã cài đặt tất cả các thư viện cần thiết (ví dụ: RDKit, scikit-learn, pandas) trước khi chạy code Python.
- **Giới hạn dữ liệu:** Bạn nên bắt đầu với một tập dữ liệu nhỏ (ví dụ: 100 dòng) để kiểm tra code và mô hình của bạn trước khi mở rộng sang toàn bộ tập dữ liệu.
- **Lưu trữ dữ liệu:** Dữ liệu fingerprint không nên lưu trữ trong dataframe mà nên lưu trữ ở ổ cứng định dạng feather để đảm bảo tốc độ truy xuất.
- **Tài nguyên:** Nên sử dụng tài nguyên card đồ họa để tăng tốc độ tính toán.

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