

PyMolPredictor 项目设计文档

V1.0

版本历史

版本/ 状态	作者	参与者	日期	备注
1.0.0	李昇程、 兰方舟、 高睿齐		2020-1-11	创建

目 录

第一部分 引言.....	3
一、编写目的.....	3
二、读者对象.....	3
三、术语与缩写解释.....	3
1、QSAR.....	3
2、分子描述符.....	3
3、SMILES	3
4、RNN	3
5、VAE.....	3
第二部分 项目概述.....	4
一、项目描述.....	4
二、项目功能描述.....	4
1、数据处理.....	4
2、模型训练.....	4
3、结果分析.....	4
4、活性预测.....	4
5、分子设计.....	4
第三部分 设计约束.....	5
一、需求约束.....	5
二、隐含约束.....	5
第四部分 前端设计.....	6
一、前端整体结构.....	6
二、实体描述.....	6
三、操作说明.....	8
第五部分 模型设计.....	19
一、模型原理简述.....	19
二、实体描述.....	20
1、QSARDNN 实体描述	20
2、SmilesRnn 实体描述	20
3、CollateFn 实体描述	20
4、SmilesRNNPredictor 实体描述.....	20
5、SmilesVAERNN 实体描述.....	21
6、SmilesDesigner 实体描述	21
第六部分 运行环境和部署.....	22
一、运行环境.....	22
二、系统性能要求.....	22

第一部分 引言

一、编写目的

本文档编写目的是为读者提供 PyMolPredictor 软件的正确操作方法，浅显讲解用到的化学学科知识和深度学习技术，以期读者能够正确使用软件完成期望的功能，为正确运行和维护提供指引，同时允许用户根据自己的需求更改相关代码，完成更复杂的任务。

二、读者对象

计算化学学科从业者，并对深度学习技术有一定的了解与兴趣；对深度学习有一定知识，想从事化学的计算机行业人员等。

三、术语与缩写解释

1. **QSAR: 定量构效关系 (Quantitative Structure-Activity Relationship)** 是一种借助分子的理化性质参数或结构参数，以数学和统计学手段定量研究有机分子生理相关性质的方法。这种方法广泛应用于药物、农药、化学毒剂等生物活性分子的合理设计。
2. **分子描述符**: 描述分子在某一方面性能的量，可作为分子特征向量的某维输入 QSAR 模型中建模。常用分子描述符有 C 原子个数、N 原子个数、水溶性等。
3. **SMILES: Simplified molecular input line entry specification**, 简化分子线性输入规范，是一种用 ASCII 字符串明确描述分子结构的规范。SMILES 由 Arthur Weininger 和 David Weininger 于 20 世纪 80 年代晚期开发，现为计算化学常用分子表示形式之一。
4. **RNN: 循环神经网络 (Recurrent Neural Network, RNN)** 是一类以序列数据为输入，在序列的演进方向进行递归且所有节点按链式连接的神经网络，常用于文本处理等。由于其在面对较长序列时具有梯度消失等问题，研究人员陆续提出其升级版 LSTM 和 GRU 等。在本软件中，RNN 相关结构用于 SMILES 串的处理和学习，同时也用于 VAE 的编码和解码结构。
5. **VAE: Variational Autoencoder** 的简称。变分自编码器作为自编码器的升级版本，除了通用的编码器 (encoder) 和解码器 (decoder) 结构外，其还通过规约向量在隐层空间 (latent space) 上符合高斯分布，使得学习的向量更加规范化，往往具有比单个 encoder 更好的性能。

第二部分 项目概述

一、项目描述

本项目实现了一个具有图形界面的化学分子信息系统，该软件命名为 **PyMolPredictor**，目前版本号是 **V1.0**。

二、项目功能描述

本项目能够完成以下五项功能：

1. 数据处理：对于用户给定的包含 **QSAR** 关系的文件（默认格式为 **csv**），进行训练集-测试集的划分，对数据中缺失值进行处理，形成较为干净的数据，可直接用于后续训练测试；对数据进行特征分析，并可视化数据在低维空间的分布，给用户以直观感受；对指定属性查看直方图分布，了解相应属性的分布等。
2. 模型训练：对于用户想要建模的 **QSAR**，根据用户输入的模型参数进行训练并保存模型；支持用户在之前的模型的基础上继续训练，进行微调；可通过分子描述符向量进行建模（**DNN**），同时也可通过 **SMILES** 串进行建模（**RNN**）；为用户显示相关训练进度和细节等等。
3. 结果分析：对于用户的训练历史进行分析，根据用户选定的 **log** 可视化验证集上的损失函数变化曲线、验证集的预测结果分布和相关模型架构等。
4. 活性预测：根据用户指定的模型，在用户选定的数据上进行预测，以表格形式输出相关预测结果，并可视化相关活性最高的分子和预测结果分布等等。
5. 分子设计：根据用户指定的包含 **SMILES** 的数据，构建分子设计模型（训练时间较长，建议使用 **GPU** 机器）；使用指定的分子设计模型进行分子设计，并按活性高低可视化设计的结果。

第三部分 设计约束

一、需求约束

前后端分离：前端的 UI 负责装载数据和模型，调用后端进行训练、预测和设计，输出和保存模型。后端的 DNN、RNN、VAE 等深度学习模型负责具体的训练、预测和设计过程。

多线程：为防止界面卡死，在计算密集型的后端程序（如 RNN、DNN 的训练程序）时，需用 PyQt 的多线程进行处理，运算所用的线程和主线程间利用信号通信。

错误处理：用户出现错误操作时，不能仅仅让程序崩溃了事，而是应该让程序继续运行，并正确告诉用户相关错误原因以及可能的解决方法。为此，需要有鲁棒的错误处理系统和错误恢复模式。

日志记录：UI 的每个 Tab 中均应包含记录日志的窗口，记录重要的用户操作及操作结果，便于用户回顾排查。

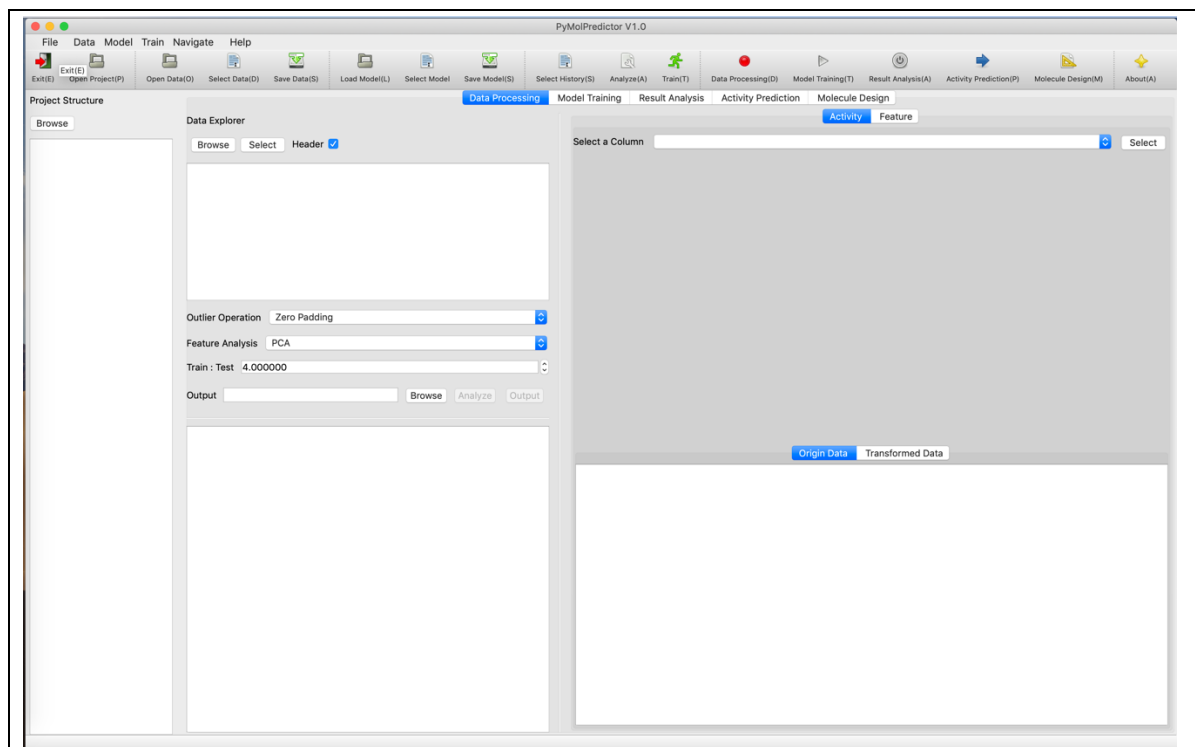
二、隐含约束

由于本软件是基于深度学习的分子预测设计系统，其面对的必然是结构化的数据，所以我们要求用户的输入必须是结构化的数据，如 csv, npy 或者 pt 等等。对此，我们要有正确处理的能力，并且当用户的输入不符合相应要求时，要能正确提醒，甚至在一定程度上自动修复。

默认用户一次只使用 GPU 或者 CPU 进行一个模型的训练或者预测，否则线程之间优先级打架，而 Python 本来是使用全局锁，多线程只是分时复用，会严重影响整体效率，故作此隐含约束。

第四部分 前端设计

一、前端整体结构



图表 1 整体界面

本项目使用 **PyQt5** 为前端。该软件的窗口分为三个部分：顶部的菜单栏，菜单栏下方的工具栏和菜单栏下方的主窗口。菜单栏包含各种操作。工具栏由菜单栏中的操作快捷方式组成。主窗口是装载数据和模型，进行预测和设计，输出和保存模型等的主要部分。在主窗口中，有 5 个选项卡：**Data Processing**, **Model Training**, **Result Analysis**, **Activity Prediction**, **Molecule Design**。

二、实体描述

前端界面的设计文件分别为主界面的 `mainwindow.ui`，`Tab0` 的 `mainwindow0.ui`，`Tab1` 的 `mainwindow1.ui`，`Tab2` 的 `mainwindow2.ui`，`Tab3` 的 `mainwindow3.ui`，`Tab4` 的 `mainwindow4.ui`。

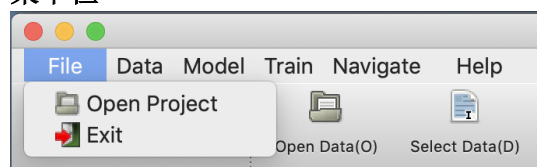
前端界面的主类为 `MainWindow`。`MainWindow` 负责加载主界面的设计文件，将 `Tab0`，`Tab1`，`Tab2`，`Tab3`，`Tab4` 嵌入 `QTabWidget` 中。此外，负责关联主界面中，各个 `Tab` 之间，以及全局性的 `signal` 和 `slot` 函数。

<p>projectBrowseSlot 函数用于浏览并选择工程文件夹；projectDoubleClickedSlot 用于工程文件夹中双击打开文件或进入文件夹的操作；closeEvent 用于重写默认的关闭窗口操作：弹出对话框询问用户是否关闭。</p>
<p>Tab0 为 Data Processing (Tab0)界面的主类，负责数据处理界面。</p> <p>outputBrowseSlot 函数负责浏览并选择输出 csv 数据的文件夹；outputSaveSlot 函数负责保存输出文件；dataBrowseSlot 函数负责浏览并选择数据文件夹；dataSelectSlot 函数用于选择数据文件夹中的数据文件；dataDoubleClickedSlot 函数用于数据文件夹中双击选择文件或进入文件夹的操作；columnSelectSlot 函数用于数据列展示的列选择；analyzeSlot 函数用于处理加载的 csv 数据，并进行相关数据分析。</p>
<p>Tab1 为 Model Training (Tab1)界面的主类，负责模型训练界面。</p> <p>startTrainingSlot 函数用于多线程深度学习的运算；modelBrowseSlot 函数用于浏览并加载深度学习模型文件；modelSelectSlot 函数用于选择模型列表中的模型；modelDoubleClickedSlot 函数用于双击选择模型列表中模型的操作；modelSaveSlot 函数用于保存训练好的深度学习模型；dataBrowseSlot 函数负责浏览并选择数据文件夹；dataSelectSlot 函数用于选择数据文件夹中的数据文件；dataDoubleClickedSlot 函数用于数据文件夹中双击选择文件或进入文件夹的操作。</p>
<p>Tab2 为 Result Analysis (Tab2)界面的主类，负责训练结果分析界面。</p> <p>trainingHistorySelectSlot 函数用于选择训练记录列表的记录；trainingHistoryDoubleClickedSlot 函数用于双击选择训练记录列表中记录的操作；analyzeSlot 函数用于分析选中的训练记录，并进行分析、绘图。</p>
<p>Tab3 为 Activity Prediction (Tab3)界面的主类，负责活性预测界面。</p> <p>modelBrowseSlot 函数用于浏览并加载深度学习模型文件；modelSelectSlot 函数用于选择模型列表中的模型；modelDoubleClickedSlot 函数用于双击选择模型列表中模型的操作；dataBrowseSlot 函数负责浏览并选择数据文件夹；dataSelectSlot 函数用于选择数据文件夹中的数据文件；dataDoubleClickedSlot 函数用于数据文件夹中双击选择文件或进入文件夹的操作；analyzeSlot 函数用于使用选中的测试集和模型进行预测、分析、绘图。</p>
<p>Tab4 为 Molecule Design (Tab4)界面的主类。</p> <p>modelBrowseSlot 函数用于浏览并加载深度学习模型文件；modelSelectSlot 函数用于选择模型列表中的模型；modelDoubleClickedSlot 函数用于双击选择模型列表中模型的操作；dataBrowseSlot 函数负责浏览并选择数据文件夹；dataSelectSlot 函数用于选择数据文件夹中的数据文件；</p>

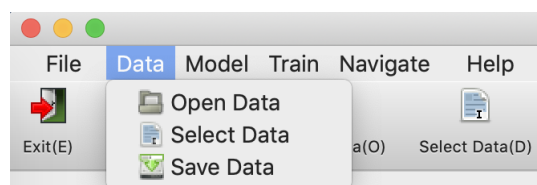
`dataDoubleClickedSlot` 函数用于数据文件夹中双击选择文件或进入文件夹的操作；`startTrainingSlot` 函数用于多线程启动 VAE 模型的训练；`designSlot` 函数用于使用加载的 VAE 模型进行分子设计。

三、操作说明

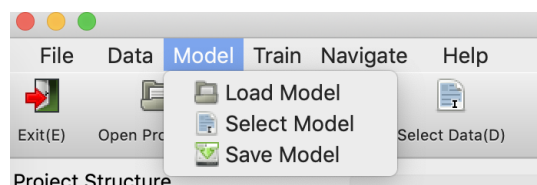
菜单栏



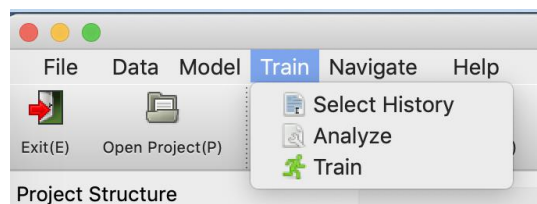
- **Open Project:** 浏览并选择当前项目所在文件夹。选定后，所有主窗口标签中的相应文件浏览窗口也会被设置项目所在文件夹。
- **Exit:** 退出软件。



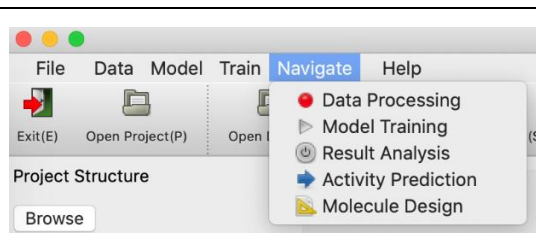
- **Open Data:** 浏览并选择为当前 tab 选择所处文件夹路径。
- **Select Data:** 选中当前 tab 中的数据项以供处理。



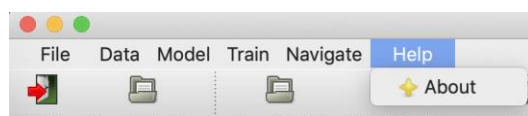
- **Load Model:** 浏览并为当前 tab 加载相应模型。
- **Select model:** 选择当前 tab 中被选定的模型并尝试加载。



- **Select History:** 在结果分析 tab 中，选定相关的训练历史问价用以分析结果。
- **Analyze:** 启动分析并在当前 tab 中展示相应结果。
- **Train:** 启动当前 tab 中的模型训练。

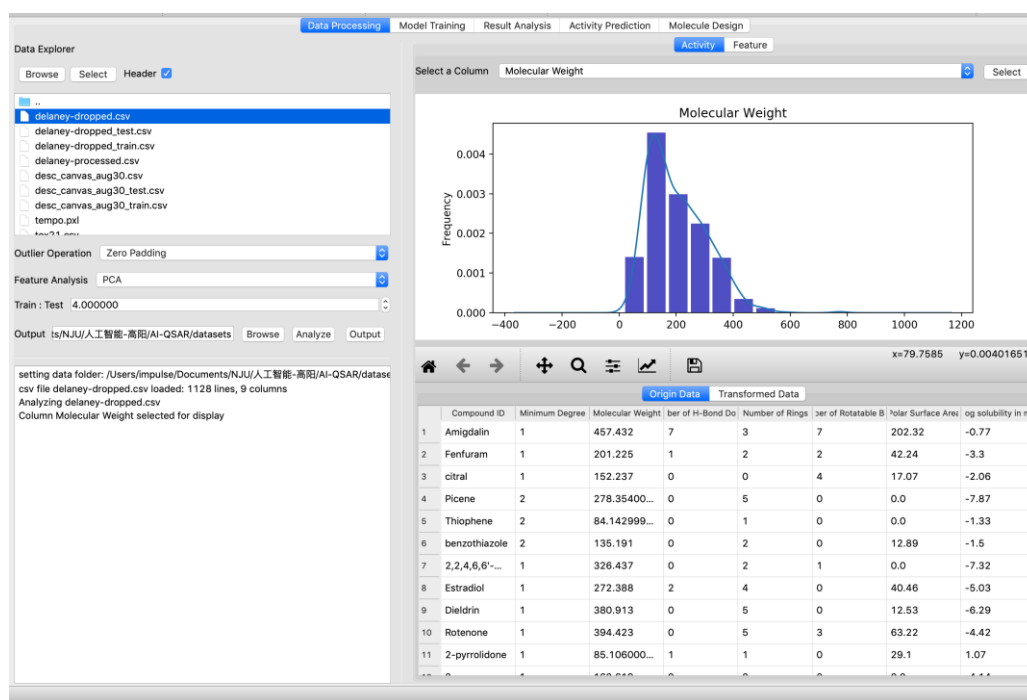


- Data Processing: 切换到数据处理 tab。
- Model Training: 切换到模型训练 tab。
- Result Analysis: 切换到结果分析 tab。
- Activity Prediction: 切换到活性预测 tab。
- Molecule Design: 切换到分子设计 tab。

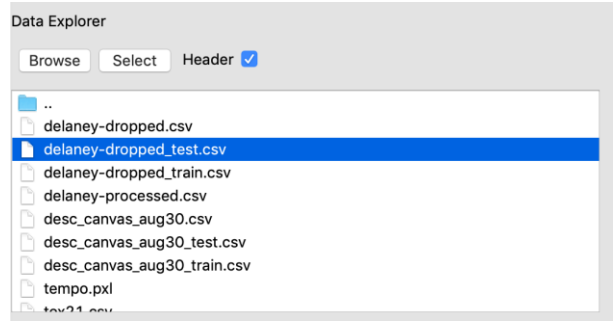


- About: 打开网页版手册。

主窗口: Data Processing

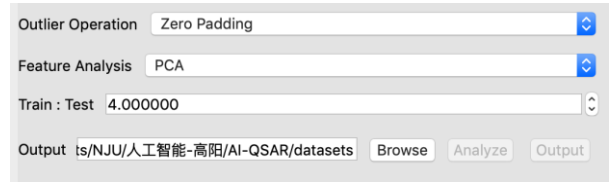


这个 tab 是用来将其他 tab 的输入数据进行预处理并输出为 csv 数据



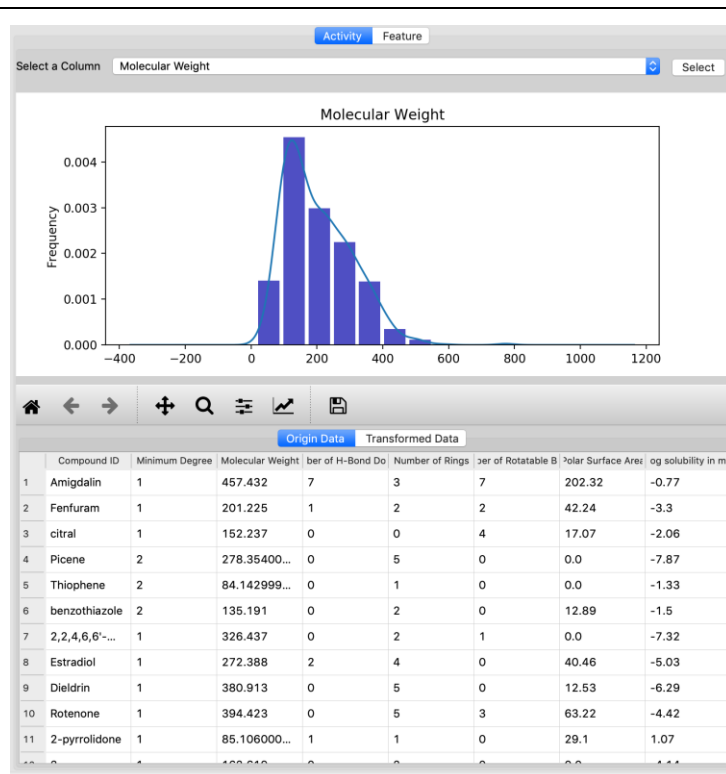
浏览并选择数据:

- **Browse:** 点击选择当前数据文件夹.
- **Select(or double click):** 点击来选择当前文件夹中的高亮数据文件.
- **Header:**如果选择了 **Header** 则把 **csv** 文件的第一行当作列名, 否则自动生成列名。



数据预处理:

- **Outlier Operation:** 选择缺省值处理方式。
- **Feature Analysis:** 选择一种降维方法将数据降维, 并在一个二维平面中绘制。
- **Train : Test:** 训练集与测试集的大小比率, 并根据给定比率将数据集划分为训练集和测试集。
- **Browse:** 点击来选择预处理并划分后的训练集与测试集的输出文件夹
- **Analyze:** 点击来分析原始数据集与处理后数据集并在右侧平面展示。
- **Output:** 点击将处理后的训练集和测试集输出到输出文件夹。



Analysis Result:

- **Activity:** 上面的屏幕用来显示选择列的分布。下面的屏幕展示原始数据与处理后数据的前 100 行。
 - **Select:** 选择一个属性并且绘制分布图。
 - **Origin Data:** 切换到原始数据
 - **Transformed Data:** 切换到处理后的数据
- **Feature:** 这个平面展示使用 PCA 或 SVD 降维后数据属性的二维图。

主窗口: Model Training

The screenshot displays the 'Model Training' tab in a software interface. It is divided into three main sections:

- Load Data:** Contains 'Browse' and 'Select' buttons, and a 'Header' checkbox which is checked. Below these is a list of files including 'delaney-dropped.csv', 'delaney-dropped_test.csv', 'delaney-dropped_train.csv', 'delaney-processed.csv', 'desc_canvas_aug30.csv', 'desc_canvas_aug30_test.csv', 'desc_canvas_aug30_train.csv', 'tempo.pkl', 'tox21.csv', 'tox21_test.csv', and 'tox21_train.csv'.
- Load Model:** Contains 'Browse' and 'Select' buttons. Below them is a file path: '/Users/impulse/Documents/NJU/人工智能-高阳/AI-QSAR/QSAR-DNN/delaney-...'.
- Training Parameters:** Includes a 'From Loaded Model' checkbox, a 'TargetType' dropdown set to 'regression', a 'Select Target' dropdown set to 'measured log solubility in', a 'Learning Rate' input field with '0.010000', an 'earlyStop' checkbox which is checked, an 'earlyStopEpochs' input field with '30', a 'Batch Size' input field with '50', and an 'Epochs' input field with '1000'. At the bottom are 'Train' and 'Save' buttons.

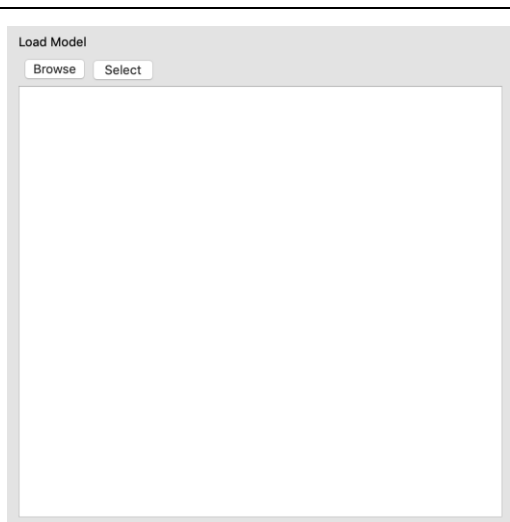
Below these sections is a 'Training Process' log showing the progress of the training, including validation R2 scores at various epochs (30, 40, 50, 60, 70).

该 tab 用来训练 QSAR 模型。

This is a close-up view of the 'Load Data' section. It shows the 'Browse', 'Select', and 'Header' (checked) buttons. Below the buttons is a list of files: 'delaney-dropped.csv', 'delaney-dropped_test.csv', 'delaney-dropped_train.csv', 'delaney-processed.csv', 'desc_canvas_aug30.csv', 'desc_canvas_aug30_test.csv', 'desc_canvas_aug30_train.csv', 'tempo.pkl', 'tox21.csv', 'tox21_test.csv', and 'tox21_train.csv'.

Browse and Select Data:

- **Browse:** 点击选择当前数据文件夹。
- **Select(or double click):** 点击来选择当前文件夹中的高亮数据文件。



Browse and Select Model:

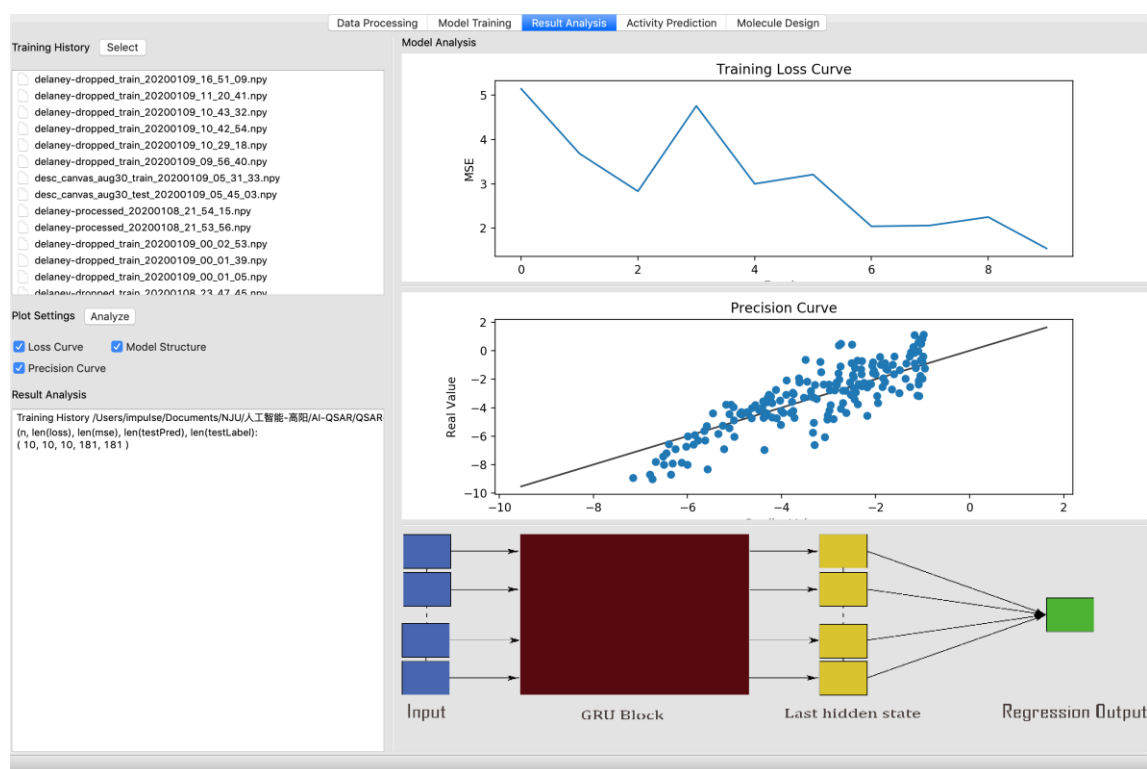
- **Browse:** 点击来浏览并且向模型列表添加一个模型文件。
- **Select(or double click):** 点击来选择模型列表里面高亮的模型文件。

Training Parameters:

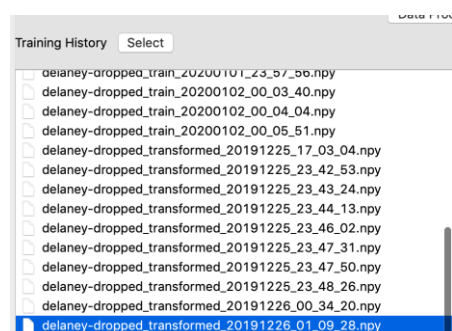
- **From Loaded Model:** 如果选择该项，会从模型文件中加载模型
- **ModelType:** 选训练模型(DNN 或 RNN).
- **TargetType:** 选择训练回归模型还是分类模型(当前仅支持回归模型)。
- **Select Target:** 选择用来预测的属性。
- **Learning Rate:** 梯度下降中的步长大小。
- **earlyStop:** 选择是否当模型损失函数在 **earlyStop** 轮数内不再下降时停止训练。
- **earlyStopEpochs:** 设定 **earlyStop** 的轮数限制。
- **Batch Size:** 训练过程中的批大小

- Epochs: 最大训练轮数。
- Train: 点击来开始训练模型，训练完成后可以保存模型
- Save: 点击来浏览并保存模型

主窗口: Result Analysis

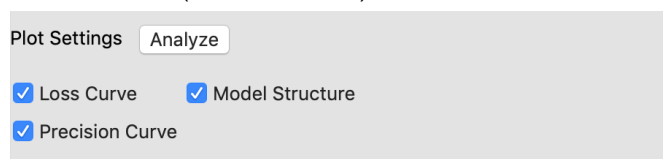


该 tab 用来分析训练的 QSAR 模型。



Training History: 在训练结束后训练历史文件被自动保存在 QSAR-DNN/___trainingcache___ .该文件夹中的所有训练历史文件均在这里展示。

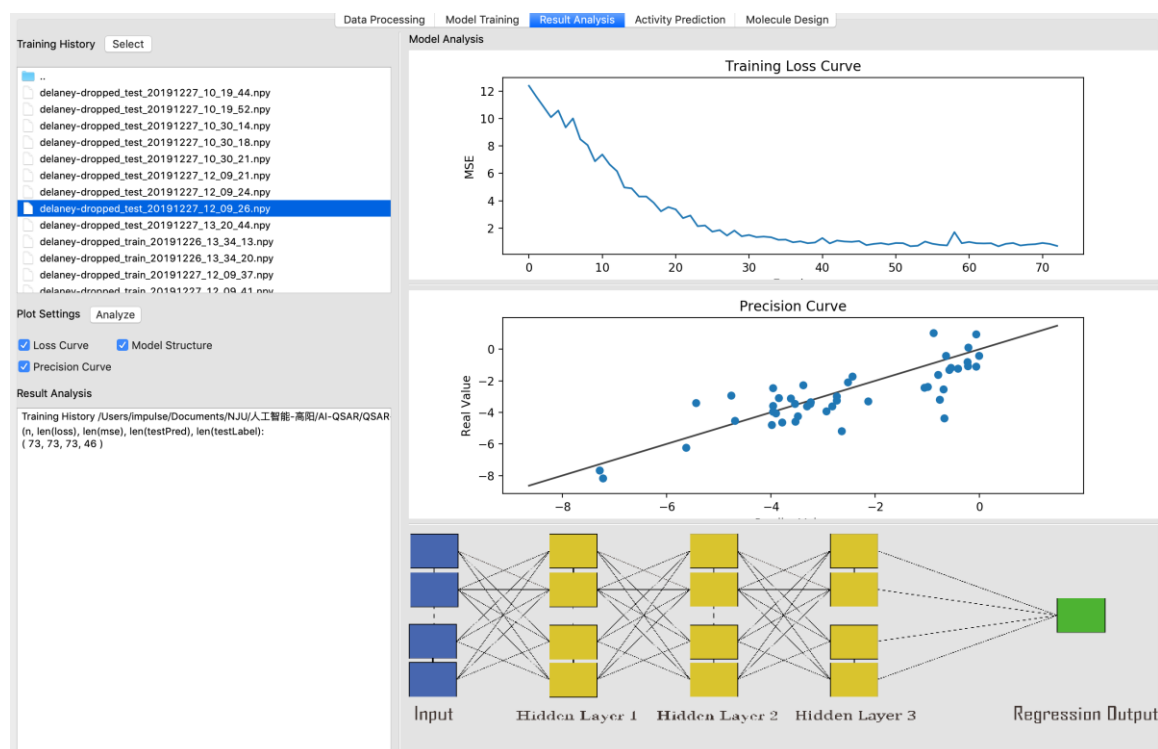
- Select(or double click): 点击来选择高亮的训练历史文件。



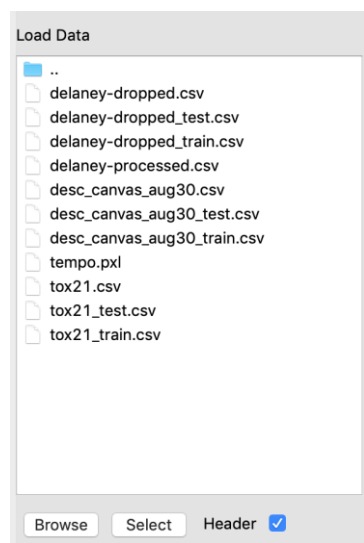
Plot Settings:

- Loss Curve: 选择是否绘制 loss 曲线
- Model Structure: 选择是否绘制模型结构
- Precision Curve: 选择是否绘制精度曲线
- Analyze: 点击来绘制选择的历史文件的分析图

主窗口: Activity Prediction

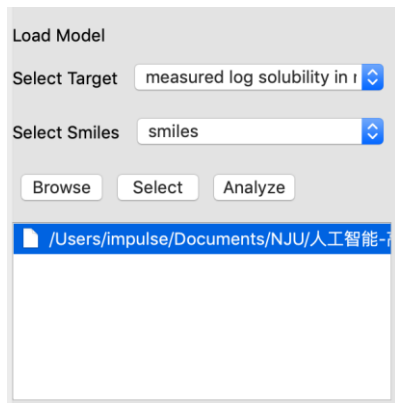


该 tab 用来在训练集上进行 QSAR 预测，绘制预测的分子模型并且绘制拟合曲线。



Browse and Select Data:

- **Browse:** 点击来选择当前文件夹
- **Select(or double click):** 点击来选择当前文件夹中的高亮文件
- **Header:** 如果选择了 **Header** 则把 **csv** 文件的第一行当作列名，否则自动生成列名。



Browse and Select Model:

- **Select Target:** 选择测试集的目标属性
- **Select Smiles:** 选择测试集中的 **SMILES** 属性列，该属性用于绘制分子结构。
- **Browse:** 点击来浏览并向模型列表中添加模型文件。
- **Select(or double click):** 点击来选择模型列表中的高亮模型文件。
- **Analyze:** 开始计算预测信息，使用目标值绘制分子，并且绘制拟合曲线。

主窗口: Molecule Design

Load Data

Browse Select Header ☒

- delaney-dropped.csv
- delaney-dropped_test.csv
- delaney-dropped_train.csv
- delaney-processed.csv
- desc_canvas_aug30.csv
- desc_canvas_aug30_test.csv
- desc_canvas_aug30_train.csv
- desc_canvas_version.pkl
- tempo.pkl
- tox21.csv
- tox21_test.csv
- tox21_train.csv

Load Model

Select Target Minimum Degree

Select Smiles smiles

Browse Select Save

/Users/impulse/Desktop/VAEModel.pkl

Train Design

No. 5 molecule designed!
No. 6 molecule designed!
No. 7 molecule designed!
No. 8 molecule designed!
No. 9 molecule designed!
No. 10 molecule designed!
No. 11 molecule designed!
No. 12 molecule designed!
-----End Designing-----

Design Info

	1	2	3	4	5	6
1						
2	-1.204725623130...	-2.141696691513...	-2.207330703735...	-2.46753191947937	-2.625004529953...	-2.940363168716...
3						
4	-3.254647016525...	-3.518069982528...	-3.534428358078...	-3.623475790023...	-3.667318105697...	-4.411048889160...

该 tab 用来训练并分析 VAE 模型来进行分子设计。

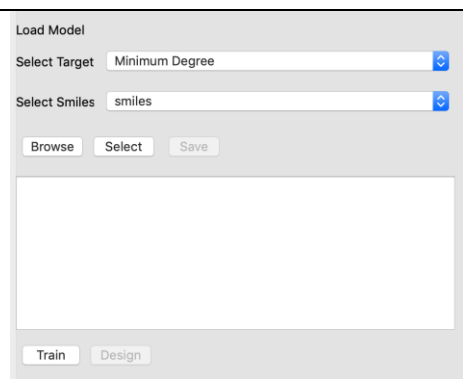
Load Data

Browse Select Header ☒

- delaney-dropped.csv
- delaney-dropped_test.csv
- delaney-dropped_train.csv
- delaney-processed.csv
- desc_canvas_aug30.csv
- desc_canvas_aug30_test.csv
- desc_canvas_aug30_train.csv
- tempo.pkl
- tox21.csv
- tox21_test.csv
- tox21_train.csv

Browse and Select Data:

- Browse: 点击来选择当前文件夹。
- Select(or double click): 点击来选择当前文件夹中的高亮文件
- Header: 如果选择了 Header 则把 csv 文件的第一行当作列名，否则自动生成列名。



Browse and Select Model:

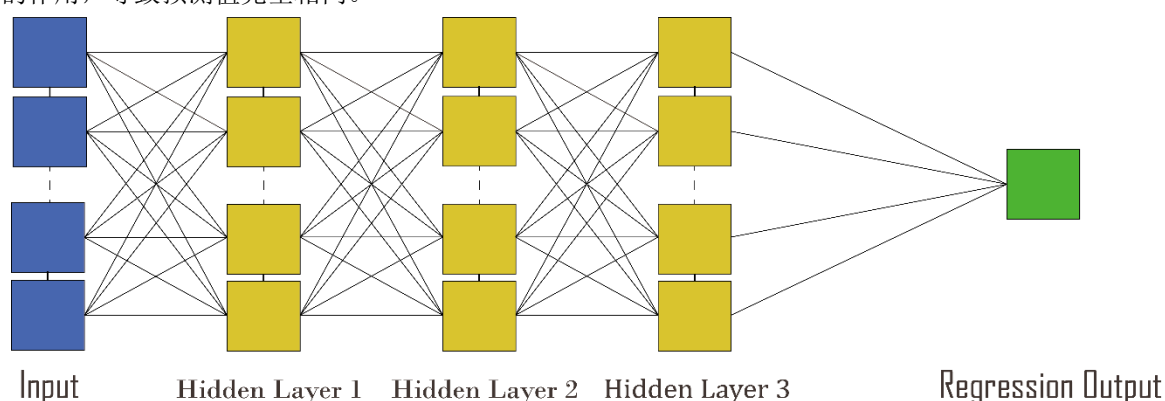
- **Select Target:** 选择训练集的目标属性
- **Select Smiles:** 选择训练集的 **SMILES** 属性，该属性用于绘制分子结构。
- **Train:** 点击来开始在选择的训练集上训练 VAE 模型。训练结束后你可以点击 **SAVE** 来保存当前模型。
- **Browse:** 点击来浏览并向模型列表中添加模型
- **Select(or double click):** 点击来选择当前模型列表中的模型文件
- **Save:** 点击来浏览并保持模型

Design: 开始载入给定 VAE 模型并根据目标属性设计并绘制分子。c

第五部分 模型设计

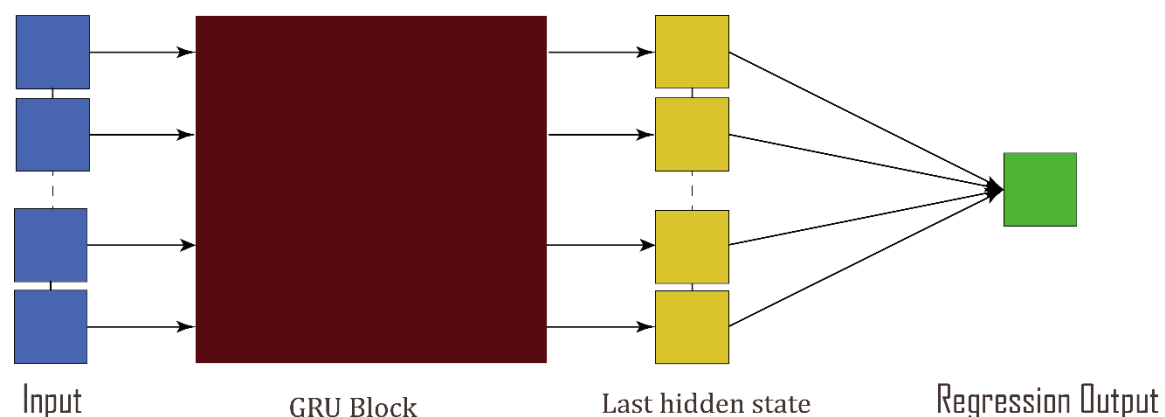
一、模型原理简述

基于分子描述符的 QSAR 建模使用多层感知机作为深度学习模型，可以进行回归预测与分类预测。回归模型使用 L2 loss 函数(均方误差)，分类模型使用 CrossEntropy Loss 函数(交叉熵)。模型为四层感知机，并且在中间层加入 batchnorm 批标准化，使得深度神经网络训练过程中每一层神经网络的输入保持相同分布的，加快模型收敛速度。模型激活函数为 RELU 函数，防止梯度下降现象。模型输入为分子描述符向量，在加载数据前需要先将数据集上的数据进行标准化，使得每一种属性在整个数据集上的分布变为均值为 0，方差为 1 的标准化数据，否则过多离散值的引入会导致部分属性失去在模型中的作用，导致预测值完全相同。



图表 2 QSAR-DNN 结构

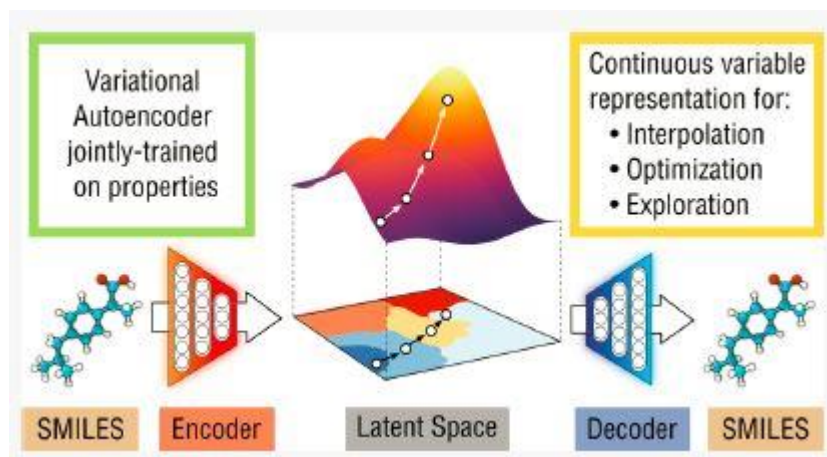
基于 SMILES 的 QSAR 建模使用 RNN 的升级版 GRU 作为深度学习模型，将分子的 SMILES 串作为输入，通过一个词典 embedding 编码后输入 GRU，将 GRU 的隐含状态的最后一层作为分子的特征向量，对该向量通过双层 ReLU 感知机得到最终预测结果。



图表 3 SMILES-RNN 结构

分子设计通过使用 RNN 作为 encoder 和 decoder 的 VAE 实现，其同样将分子的 SMILES 串作为输入，同时将 encoder 得到的向量，配套搭建一个多层感知机进行回归，另外将该向量通过 decoder 重建，返回过程中所有相关变量，整个网络各组件同步训练。训练结束后，通过直接通过在隐含空间上进行无

约束优化找到性能较好的隐层向量，解码后即得到设计的分子。由于解码过程为概率化过程，我们的方法可以产生多组满足相应性质的分子。训练过程中使用 KL 散度退火技术同时保证较高的复建率和预测的准确率。



图表 4 分子设计示意图

二、实体描述

1. QSARDNN 实体描述

本项目中我们将基于分子描述符的 QSAR 模型封装为 QSARDNN 类，该类提供了 train, test, load, save 以及 setPropertyNum 五种接口。

- setPropertyNum 用于设定模型输入的维度(分子描述符的维度)以实例化多层感知机模型。
- train 函数用于模型的训练, 用户可以自己指定训练的学习率, 批大小, 训练的轮数以及是否需要 earlystop; 训练过程中程序会自动划分 validation 集作为判断是否过拟合, 若指定 earlystop 则当在 validation 集合上预测效果下降时提前退出。
- test 函数使用当前训练好的模型进行预测, 返回预测的标签集。
- save 和 load 函数分别负责保存和加载当前模型。

2. SmilesRnn 实体描述

该类为基于 SMILES 的 QSAR 模型的实现, 该类仅重写了 forward 函数。

- forward 用于产生预测结果, 对输入的编码为 torch.long 类型的 SMILES 串进行编码后送入内置的 GRU, 将隐藏状态的最后一层通过两个全连接层输出, 产生预测结果。

3. CollateFn 实体描述

该类为基于 RNN 相关模型的辅助类, 帮助处理变长 batch。经测验, 未处理变长 batch、仅将其全部补 0 到最长串长度的数据性能远差于处理后的版本, 故该类有助于提高整体性能。

4. SmilesRNNPredictor 实体描述

该类为基于 SMILES 的 QSAR 模型的封装类, 包含 initFromData, train, saveModel, loadFromModel, predict 等接口与函数。

- initFromData(smiles, properties): 接受 smiles 数组和 properties 数组作为输入, 在进行分词后产生符合 SmilesRnn 规格的数据形式, 同时自动划分出训练集和验证集用以 earlyStop。
- train(trainData=None, trainProp=None, nRounds=1000, lr=0.01, earlyStop=True, earlyStopEpoch=10, batchSize=12, signal=None): 根据 trainData 和 trainProp, 决

	<p>定在输入数据或者之前初始化的数据上进行训练；根据其他指定参数调节训练细节。其中 signal 信号专门用于传递 debug 信息和与前端通信。</p> <ul style="list-style-type: none"> - saveModel(modelPath): 保存网络参数和 SMILES 编码解码字典到指定路径。 - loadFromModel(modelPath): 从指定路径加载模型，包括网络参数和 SMILES 编码解码字典。 - predict(data, batchSize=50): 对指定数据 data 进行预测，预测中的批输入大小默认为 50。建议该批输入大小和模型训练时批输入大小相同以获取最佳性能。
5.SmilesRNNVAE 实体描述	
	<p>该类为基于 SMILES 的分子设计模型的实现，使用 VAE 的相关原理，包含 encode, decode, reparameterize, forward 和 middleRepresentation 等接口与函数。</p> <ul style="list-style-type: none"> - encode(x): 将输入通过 embedding 编码后输入编码器 GRU 中，得到的隐藏状态的最后一维作为输出，并将该输出通过全连接层获得用于重参数化的均值和方差向量。 - decode(z, maxLength): 将输入 z 视为上下文变量，输入解码器 GRU 中解码出长度为 maxLength 的串，返回相应的重建结果。 - reparameterize(mu, logVar): 返回以 mu 为均值，logVar 为方差的高斯分布上采样获得的某个向量。此举被称为 Reparametrization trick，能同时保证变分编码和网络链式法则有效。 - middleRepresentation(x): 获取输入 x 的中间隐含空间表示。 - forward(x): 对输入 x，返回解码重建的 x 的概率分布、均值 mu、方差 logVar 和预测结果 predY，用于计算 vae 重建误差。
6.SmilesDesigner 实体描述	
	<p>该类为基于 SMILES 的分子设计模型的封装类，包含 initFromSmilesAndProps, trainVAE, loadVAE, initFromModel, encodeDataset, molecularRandomDesign 等接口与函数。</p> <ul style="list-style-type: none"> - initFromSmilesAndProps(smiles, properties): 接受 smiles 数组和 properties 数组作为输入，在进行分词后产生符合 SmilesRnn 规格的数据形式，同时自动划分出训练集和验证集用以 earlyStop。 - trainVAE(nRounds=1000, lr=0.01, earlyStop=True, earlyStopEpoch=10, batchSize=12, signal=None): 对内含 VAE 网络以 lr 的学习率、batchSize 的批大小训练 nRounds 轮，若 earlyStop 为真，则在验证集连续 earlyStopEpoch 性能没有优化时即停止训练。其中 signal 信号专门用于传递 debug 信息和与前端通信。 - loadVAE(path): 从指定路径加载 VAE 模型。 - initFromModel(modelPath): 从指定路径加载模型，包括网络参数和 SMILES 编码解码字典。 - encodeDataset(batchSize=30): 以默认为 30 的批大小将训练集编码为其隐层空间上的向量，用于后续处理。 - molecularRandomDesign(aimNumber=100, batchSize=10, signal=None): 通过在隐层空间上随机采样进行分子设计，直到设计出 aimNumber 个符合 SMILES 串语法的分子。signal 信号用于传递 debug 信息和与前端通信。 - testLatentModel(batchSize=12): 检查构造的 VAE 构建的隐层空间的性质，分别在命令行打印随机森林回归器和在该隐层空间配套的回归器在测试集上的表现。

第六部分 运行环境和部署

一、 运行环境

配置:

本软件需要 python3.6+环境。对于本软件需要的 python 包, 请参阅 requirements.txt 并运行 `pip install -r requirements.txt` 进行安装。

此外, 您必须安装有效版本的 rdkit。对于 conda virtualenv 中的 python3.6, 可以使用 `conda install conda-forge::rdkit` 进行 rdkit 的安装。如果要创建 python3.6 的 conda 虚拟环境, 请运行 `conda create --name py36 python=3.6`。您可以使用 `source activate py36` 和 `source deactivate py36` 来激活和停用虚拟环境。

运行:

进入文件夹 QSAR-GUI 并运行 `python main.py`

二、 系统性能要求

DNN 模型训练有 CPU 即可, RNN 训练如果对性能有需求请使用较好的 CPU, 或者 GPU; 分子设计模型 VAE 请使用较好的 CPU 或 GPU (显存至少 10G)。