

iModulon Chatbot Assistant

This Jupyter Notebook runs an AI chatbot assistant designed to interact with the [iModulon database](#). The chatbot utilizes OpenAI's GPT-4o model to answer queries, provide information, and assist with data analysis related to iModulons. The assistant also has access to gene information from the [ecocyc database](#).

Overview

The notebook is structured as follows:

1. **Imports and Setup:** This section includes necessary imports and configurations for running the chatbot.
2. **Environment Setup:** Here, you input your OpenAI API key to access the GPT-4o model.
3. **Chatbot Initialization:** This part sets up the chatbot, loading necessary tools and defining the chat prompt template.
4. **Chat Interface:** A simple interface for interacting with the chatbot, where users can input queries and receive responses.

Functionality

The chatbot supports a variety of functions related to iModulons, including but not limited to:

- Learning about iModulons
- Finding closest iModulon, genes, and conditions
- Getting detailed information about genes and conditions
- Plotting gene expression and iModulon activity
- Comparing gene expression and iModulon activities
- Executing Python code for custom analysis

Usage

To use the chatbot:

1. **Setup:** Ensure you have your OpenAI API key ready and input it when prompted.
2. **Run the Notebook:** Execute each cell in the notebook sequentially to initialize the chatbot.
3. **Interact:** Type your queries into the chat interface. Use commands like 'exit', 'quit', or 'q' to terminate the session. For detailed examples demonstrating the capabilities of the chatbot, refer to the [example conversations PDF](#).

```
In [1]: import os
import difflib
import traceback
import getpass
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from IPython.display import display, HTML, Markdown
from langchain_openai import ChatOpenAI
from langchain.agents import AgentExecutor, create_tool_calling_agent, load_tools
from langchain_core.prompts import ChatPromptTemplate
from langchain_core.messages import AIMessage, HumanMessage
from langchain_core.tools import tool
from imodulon_functions import *
```

```
In [2]: os.environ["OPENAI_API_KEY"] = getpass.getpass("OpenAI API Key: ")
```

OpenAI API Key:

```
In [3]: llm = ChatOpenAI(model="gpt-4o", temperature=0)
with open("imodulon_chat_prompt.txt", "r", encoding="utf8") as file:
    imodulon_chat_prompt = file.read()

tools = [
    learn_about_imodulons,
    find_closest_imodulon,
    find_closest_gene,
    find_closest_condition,
    get_genes_of_imodulons,
```

```

    get_condition_info,
    get_gene_info,
    get_imodulon_info,
    plot_gene_expression,
    plot_imodulon_activity,
    plot_all_imodulon_activities_for_condition,
    compare_gene_expression,
    compare_imodulon_activities,
    plot_dima,
    execute_python_code,
]
llm_with_tools = llm.bind_tools(tools)
prompt = ChatPromptTemplate.from_messages([
    ("system", imodulon_chat_prompt),
    ("placeholder", "{chat_history}"),
    ("human", "{input}"),
    ("placeholder", "{agent_scratchpad}")
])

agent = create_tool_calling_agent(llm_with_tools, tools, prompt)
agent_executor = AgentExecutor(agent=agent, tools=tools, verbose=False)
chat_history = []

display(HTML("""
<style>
    .output_wrapper, .output {
        height: auto !important;
        max-height: 1000px;
        overflow-y: auto;
    }
</style>
"""))

```

```

In [4]: display(Markdown("***Welcome to iModulon Chat** <br /> Type 'exit' 'quit' or 'q' to quit "))
while True:
    display(Markdown("***Input:***"))
    user_input = input()
    if user_input.lower() in ['exit', 'quit', 'q']:
        break

    # Prepare the input for the agent
    input_data = {
        "input": user_input,
        "chat_history": chat_history
    }

    # Run the agent
    response = agent_executor.invoke(input_data, handle_parsing_errors=True)
    string_response = f"***iM chat:** <br />{response['output']}"
    display(Markdown(string_response))
    #print(f"\n***iM chat: **{response['output']}\n")
    # Update the chat history
    chat_history.append(HumanMessage(content=user_input))
    chat_history.append(AIMessage(content=response['output']))

```

Welcome to iModulon Chat

Type 'exit' 'quit' or 'q' to quit

Input:

Are there any experimental conditions in the iModulon database that used antibiotics?

Finding closest conditions to ['antibiotic for selection: any']

iM chat:

In the presence of ciprofloxacin, the following iModulons were highly active:

1. **Cold Shock:** Activity level of 27.92
2. **FlhA:** Activity level of 26.14
3. **DNA Damage:** Activity level of 16.08
4. **FlhDC-2:** Activity level of 15.73
5. **UC-2:** Activity level of 15.65

These iModulons showed the highest activity levels under the condition with ciprofloxacin in M9 media.

Input:

What do those iModulons do? Why do you think they were active in this condition?

Retrieving info of ['Cold Shock'] imodulons

Retrieving info of ['FlhA'] imodulons

Retrieving info of ['DNA Damage'] imodulons

Retrieving info of ['FlhDC-2'] imodulons

Retrieving info of ['UC-2'] imodulons

iM chat:

Here is the information about the iModulons that were highly active in the presence of ciprofloxacin:

1. Cold Shock iModulon

- **Function:** Cold shock proteins
- **Highly Weighted Genes:** cspH, cspG, pinR, ynaE, ynfN, cspl, cspB, cspF, lpxP, stpA, cspA, ymcF, ynfQ, ynaM, ynfT

2. FliA iModulon

- **Regulator:** FliA
- **Function:** Sigma 28; flagellum biosynthesis and export, motility
- **Highly Weighted Genes:** ykfB, modA, modB, modC, flgN, flgM, flgK, flgL, bhsA, ycgR, trg, flxA, ves, ynjH, cheZ, cheY, cheB, cheR, tap, tar, cheW, cheA, motB, motA, fliZ, fliC, fliD, fliS, fliT, fliK, aer, yhjG, pdeH, uhpT, rdca, rdcB, tsr

3. DNA Damage iModulon

- **Regulator:** LexA
- **Function:** SOS response to DNA damage or inhibition of DNA replication
- **Highly Weighted Genes:** polB, dinB, uvrB, sula, dinI, umuD, umuC, ydjM, cho, yebF, yebG, recN, recX, recA, dinD, lexA, dinF, uvrA

4. FlhDC-2 iModulon

- **Regulator:** FlhDC
- **Function:** Flagellum biosynthesis and export
- **Highly Weighted Genes:** flgN, flgM, flgA, flgB, flgC, flgD, flgE, flgF, flgG, flgH, flgI, flgJ, flgK, flgL, ves, ynjH, flhE, flhA, flhB, tar, flhC, flhD, yecR, fliZ, fliA, fliD, fliS, fliT, fliE, fliF, fliG, fliH, fliI, fliJ, fliK, fliL, fliM, fliN, fliO, fliP, fliQ, fliR, tsr

5. UC-2 iModulon

- **Function:** Unknown
- **Highly Weighted Genes:** cra, sfsA, dxr, yaiZ, glsA, ylbH, glxK, acrZ, ybhC, fsaA, potH, hcr, cydD, ycaR, kdsB, mukF, aspC, pepN, zapC, hyaA, ymcE, csqC, yceM, potB, ompW, abgA, ldhA, ydcV, patD, sra, ydgH, yodC, yedI, yedA, cobT, cobS, yegR, mrp, yfaZ, nuoK, nuoJ, nuol, dedA, truA, alaC, glk, maeB, guaB, ascG, rpoS, nlpD, ygbE, cysC, queE, rlmD, gcvP, yggR, yggN, hybC, hldE, ygiF, yraR, yhbO, yrdB, aroE, kefG, pabA, trpS, asd, livJ, ftsE, ftsY, yhhH, bcsQ, yhjR, dppB, xylB, rbsD, ilvY, dapF, yigA, zraR, yjaA, pgi, ghxP, yjdI, nsrR, lptF, yjjU, yjjV, lplA, ytbB, hokB, yghX, yneO, ybgU

Possible Reasons for Activity in Ciprofloxacin Condition:

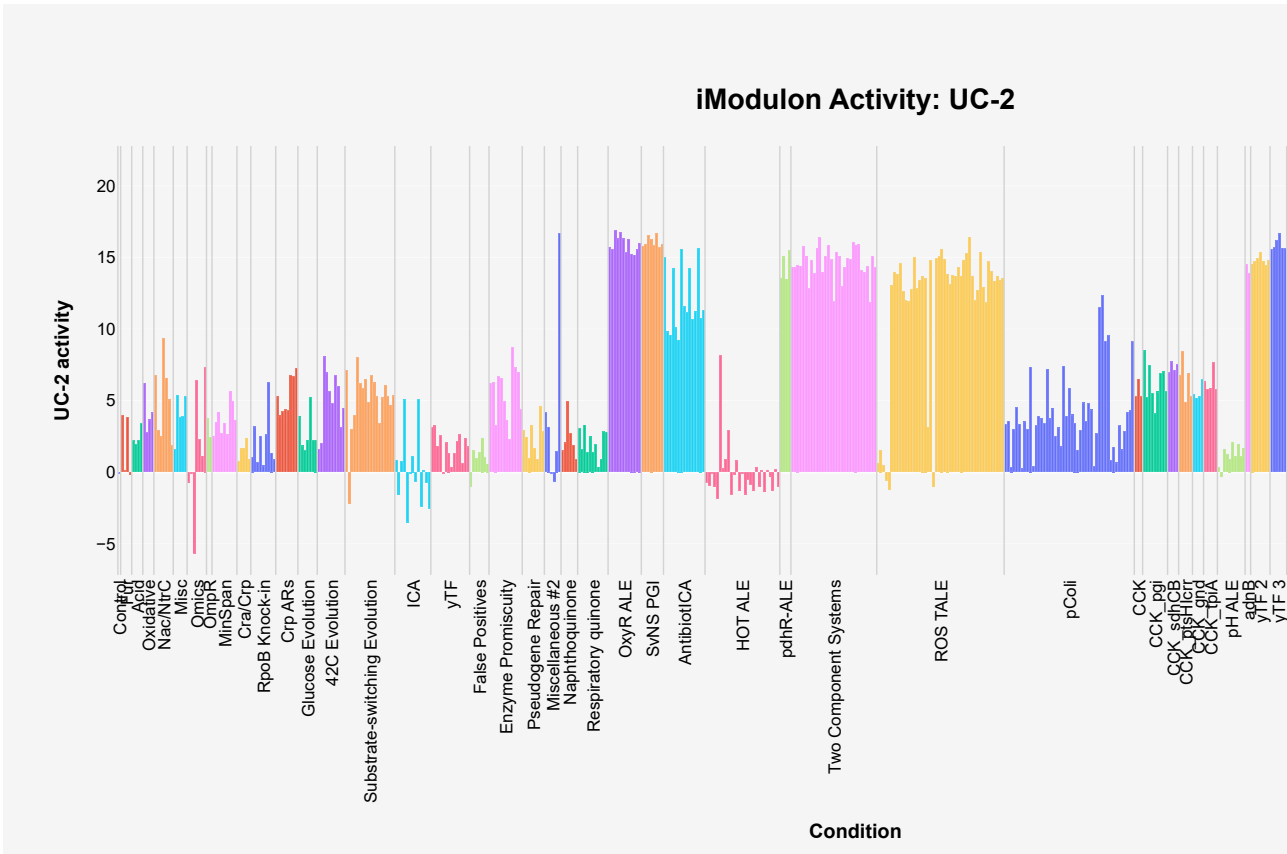
- 1. Cold Shock iModulon:** Ciprofloxacin can induce stress responses in bacteria, including cold shock proteins, which help in stabilizing RNA and proteins under stress conditions.
- 2. FliA and FlhDC-2 iModulons:** These iModulons are involved in flagellum biosynthesis and motility. Stress conditions, such as the presence of antibiotics, can trigger motility and biofilm formation as survival strategies.
- 3. DNA Damage iModulon:** Ciprofloxacin is known to cause DNA damage by inhibiting DNA gyrase and topoisomerase IV. The activation of the DNA damage response (SOS response) is a direct consequence of this damage.
- 4. UC-2 iModulon:** The function of this iModulon is unknown, but its activation might be related to a stress response or other regulatory mechanisms triggered by ciprofloxacin.

These iModulons' activities suggest that ciprofloxacin induces a multifaceted stress response in *E. coli*, involving DNA repair, motility, and possibly other unknown mechanisms.

Input:

Which other conditions is UC-2 active in?

Plotting UC-2 activity



iM chat:

The UC-2 iModulon is highly active in the following conditions:

1. **minE_ph5 (Minicoli):** Activity level of 21.29
2. **minE_osm (Minicoli):** Activity level of 20.29
3. **minE_cyt (Minicoli):** Activity level of 19.64
4. **minE_glyc (Minicoli):** Activity level of 19.25
5. **minE_Fe_starve (Minicoli):** Activity level of 19.09

These conditions are part of the "Minicoli" study, indicating that the UC-2 iModulon is particularly active under various stress conditions in this study.

The UC-2 iModulon is least active in the following conditions:

1. **wt_thr (Omics):** Activity level of -5.69
2. **leu_glc (ICA):** Activity level of -3.5
3. **ade_glc (ICA):** Activity level of -2.53
4. **tyr_glc (ICA):** Activity level of -2.41
5. **wt_glc (Substrate-switching Evolution):** Activity level of -2.18

In the control conditions, the activity levels of UC-2 iModulon are as follows:

1. **wt_glc (Control):** Activity level of -0.1
2. **wt_glc (MinSpan):** Activity level of 3.44
3. **wt_glc (Substrate-switching Evolution):** Activity level of -2.18
4. **wt_glc (ICA):** Activity level of 0.83
5. **wt_glc (yTF):** Activity level of 3.13
6. **wt_glc (OxyR ALE):** Activity level of 15.67
7. **wt_glc (yTF 2):** Activity level of 14.54

The UC-2 iModulon shows varying levels of activity across different conditions, with the highest activity observed in the "Minicoli" study under various stress conditions.

Input:

Speculate on what the UC-2 iModulon's function is based on the available information

iM chat:

Based on the available information, we can speculate on the potential function of the UC-2 iModulon by examining the conditions in which it is highly active and the genes it includes.

Conditions of High Activity:

The UC-2 iModulon is highly active in the following conditions:

1. **minE_ph5 (Minicoli)**: Low pH stress
2. **minE_osm (Minicoli)**: Osmotic stress
3. **minE_cyt (Minicoli)**: Cytoplasmic stress
4. **minE_glyc (Minicoli)**: Glycerol as a carbon source
5. **minE_Fe_starve (Minicoli)**: Iron starvation

Conditions of Low Activity:

The UC-2 iModulon is least active in conditions involving standard growth media or specific amino acid limitations, such as:

1. **wt_thr (Omics)**: Wild-type with threonine
2. **leu_glc (ICA)**: Leucine limitation with glucose
3. **ade_glc (ICA)**: Adenine limitation with glucose
4. **tyr_glc (ICA)**: Tyrosine limitation with glucose
5. **wt_glc (Substrate-switching Evolution)**: Wild-type with glucose

Genes in UC-2 iModulon:

The UC-2 iModulon includes a diverse set of genes, some of which are involved in stress responses, metabolism, and regulatory functions. Here are a few notable genes:

- **cra**: Catabolite repressor/activator
- **glk**: Glucokinase
- **rpoS**: RNA polymerase sigma factor (stress response)
- **cydD**: Cytochrome D oxidase subunit
- **hyaA**: Hydrogenase 1 small subunit
- **ftsE**: Cell division protein
- **pgi**: Phosphoglucose isomerase

Speculation on Function:

Given the high activity of the UC-2 iModulon under various stress conditions (low pH, osmotic stress, cytoplasmic stress, iron starvation) and its low activity under standard growth conditions, it is likely that the UC-2 iModulon is involved in a general stress response mechanism. The presence of genes like **rpoS** (a key regulator of the general stress response), **cydD** (involved in oxidative stress response), and **hyaA** (involved in hydrogen metabolism) supports this hypothesis.

Additionally, the inclusion of metabolic genes like **glk** and **pgi** suggests that the UC-2 iModulon may also play a role in adjusting metabolic pathways in response to stress conditions.

Conclusion:

The UC-2 iModulon likely functions as a general stress response iModulon, helping the cell to adapt to various environmental stresses by regulating genes involved in stress response, metabolism, and possibly other protective mechanisms. This broad activation under different stress conditions indicates its role in maintaining cellular homeostasis and survival under adverse conditions.

Input:

q