



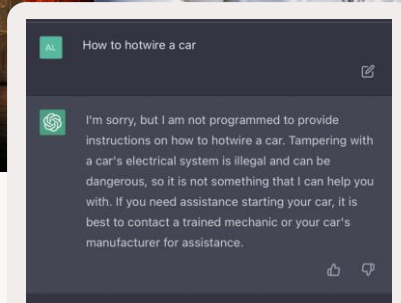
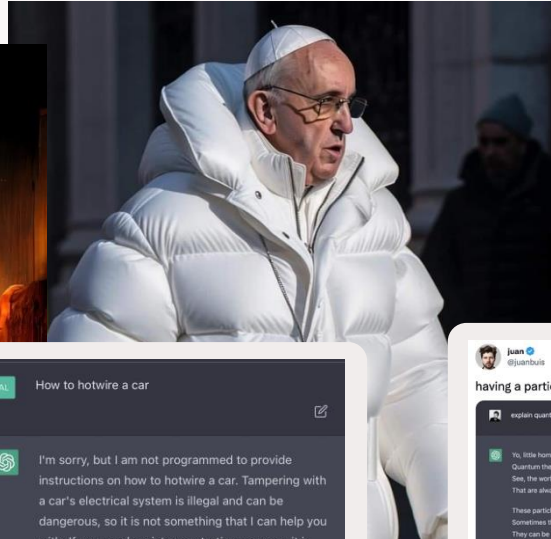
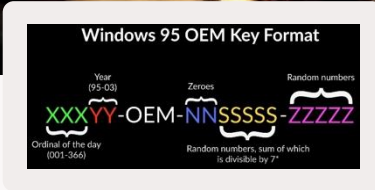
From Promise to Practice

Responsible Use of Generative AI in
Clinical Development

Sean McGee, Director of Product, Certara.AI | March 5, 2026

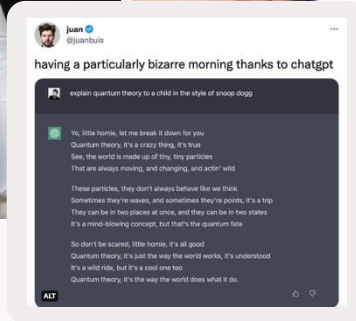
When people hear “AI,” they tend to think of GPTs

Tools like ChatGPT and DALL-E made AI feel “real” to the general public



Everyone: AI art will make designers obsolete

AI accepting the job:



GPTs can do specialized jobs at unimaginable scales

GPTs work with text-based data as well as or even better than human baseline performance

Generate



Produce new data based on a prompt

Search/Similarity



Return data that is related to the submitted prompt

Extract



Pull out key pieces of info from unstructured data

Summarize



Condense unstructured data into its main points

Cluster



Sorting data into groups based on their similarity

Rewrite



Suggests changes to text based on similar examples

Classify



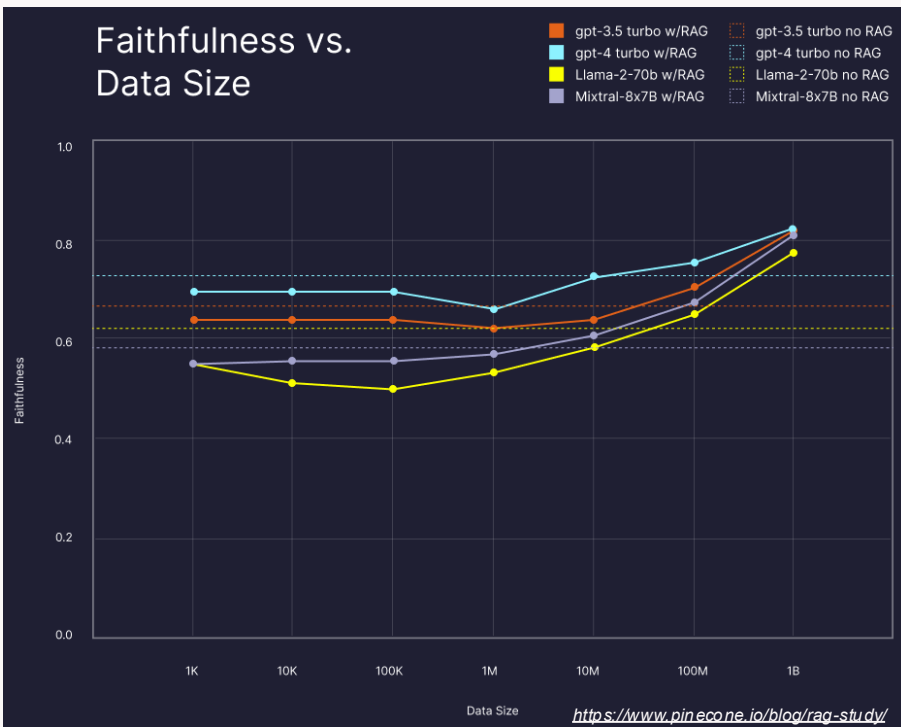
Sorting data into groups based on predefined classes

Retrieval Augmented Generation (RAG) Evens the Playfield

RAG improves generative output quality and provides users traceability to verify claims

“Our study not only underscores the scalability of RAG in enhancing LLMs but also demonstrates that with the application of RAG — **LLMs of varied power and sizes can achieve almost equal high levels of accuracy and reliability** — democratizing access to state-of-the-art capabilities of generative AI across different LLMs.”

- RAG with more data **significantly improves** the results of generative AI applications.
- The more data you can search over, the more "faithful" (factually correct) the results.
- **RAG with massive data on-demand is better than GPT4** (without RAG), even with the data it was trained on.
- RAG, with a lot of data, provides SOTA performance **no matter what LLM you choose**. This insight unlocks using different LLMs (e.g., open-source or private LLMs).



Quantitative Systems Pharmacology (QSP)

Increasing Regulatory Acceptance of QSP Modeling

ARTICLE

Quantitative systems pharmacology: Landscape analysis of regulatory submissions to the US Food and Drug Administration

Jane P. F. Bai | Justin C. Earp | Jeffry Florian | Rajanikanth Madabushi |
David G. Strauss | Yaning Wang | Hao Zhu

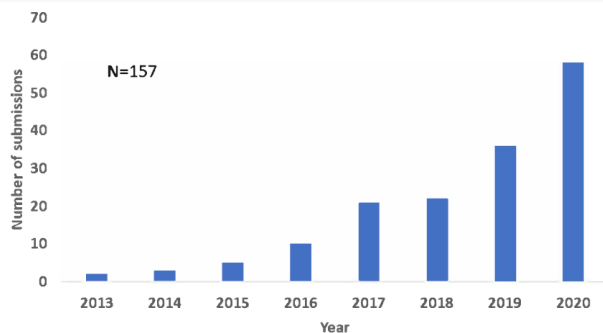
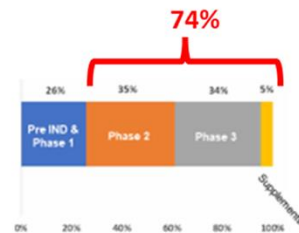
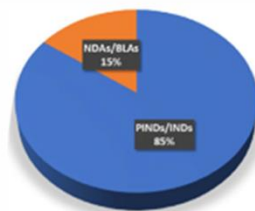


FIGURE 1 The yearly trend of quantitative systems pharmacology submissions since 2013

QSP Submissions Equally Span Different Phases of Drug Development



- 66% were efficacy related submissions,
- 97% supporting dosing:
 - 27% phase 1, 39% phase 2, 18% phase 3 & PSP, 12% NDA/BLA
- 34% were safety related: liver toxicity, bone density, cardiotoxicity, plasma ion conc

Oncology: largest TA
32%

[fda.gov](https://www.fda.gov)

CPT Pharmacometrics Syst Pharmacol. 2021;10:1479–1484.

5

Automatic Code Generation

Generative AI to accelerate model analysis scripting

- Helps simplify repeatable tasks
- Can help modelers more quickly parse complex model scripts
- Simplifies the conversion of models from one code format to another

The screenshot displays a Jupyter Notebook environment with a dark theme. A prompt box at the top left contains the text: "Write a cell to read invitro.csv into a table and summarize the included assays by the count, fixed concentration, and concentration range". Below this, a code cell contains Python code for reading a CSV file and summarizing assay data. To the right, the notebook shows the generated content, including a section titled "2. Objectives" with two bullet points, a section titled "3. Analysis Data Sets" with a paragraph of text, and a table titled "Summary of In Vitro Assays:". The table has six columns: Assay Type, Number of Measurements, Fixed Antibody (pM), TNFα Range (pM), Fixed TNFα (pM), and Antibody Range (pM). The table contains two rows of data. A red box highlights the table and the section header "4. Model description and assumptions".

Write a cell to read invitro.csv into a table and summarize the included assays by the count, fixed concentration, and concentration range

Automatic code generation to make tables and figures

```
df = pd.read_csv('data/invitro.csv')

# Create summary for each assay type
summary = []
for assay in df['assay'].unique():
    assay_data = df[df['assay'] == assay]

    # Determine which concentration is varying vs fixed
    if 'varying TNF' in assay:
        fixed_conc = assay_data['Ab_pM'].iloc[0]
        varying_range = (assay_data['L_pM'].min(), assay_data['L_pM'].max())
        fixed_type = 'Antibody'
        varying_type = 'TNFα'
    else:
        fixed_conc = assay_data['L_pM'].iloc[0]
        varying_range = (assay_data['Ab_pM'].min(), assay_data['Ab_pM'].max())
        fixed_type = 'TNFα'
        varying_type = 'Antibody'

    summary.append({
        'Assay Type': assay,
        'Number of Measurements': len(assay_data),
        f'Fixed {fixed_type} (pM)': fixed_conc,
        f'{varying_type} Range (pM)': f'{varying_range[0]} - {varying_range[1]}'
    })

# Create summary dataframe
summary_df = pd.DataFrame(summary)
print("Summary of In Vitro Assays:")
display(summary_df)
```

2. Objectives

- Identify the binding affinity of antibody for TNFα
- Predict the dose level that reduces free TNFα levels by 90%

3. Analysis Data Sets

The data used to inform the model originate from *in vitro* assays measuring TNFR signaling in response to various concentrations of antibody and TNFα. In one assay, the antibody concentration is fixed while TNFα concentration is varied, and vice versa for the second assay. TNFα signal is measured via a luciferase reporter. Table 1 reports the number of data points and the tested concentrations.

Assay Type	Number of Measurements	Fixed Antibody (pM)	TNFα Range (pM)	Fixed TNFα (pM)	Antibody Range (pM)
0 varying TNFα	12	30.0	0.1 - 100000.0	NaN	NaN
1 varying antibody	12	NaN	NaN	30.0	0.1 - 100000.0

4. Model description and assumptions

A model diagram may be found in Figure 1. The 1-compartment monospecific anti-ligand model is a single-compartment model describing drug administration, target-binding, and elimination. Drug administration can be described as an intravenous (IV) bolus, or subcutaneous (SC) administration with a 1st order absorption rate. Target ligand and its cognate receptor are synthesized in the compartment with a 0th order rate. Ligand binds reversibly to the receptor, specified by a monovalent equilibrium dissociation constant (Kd). Drug binds reversibly to the target ligand, specified by a separately parameterized Kd, and blocks ligand-receptor interactions. All species are eliminated through 1st order processes.

ut.caption('Figure 1. Model diagram.', 'Figure was created using BioRender.') ***

Orientation to data sets

AI can look at a CSV file and provide the modeler an overview of what's in their including a guess at the underlying experiment design

The screenshot displays a Jupyter Notebook environment with a dark theme. On the left, a sidebar shows a file explorer with 'invitro.csv' selected. The main area is divided into two panes. The left pane, titled 'ABM Assistant', shows an AI-generated analysis of the CSV file. The right pane shows the Python code used for the analysis and its output, which includes a table summarizing the data.

ABM Assistant

This appears to be *in vitro* experimental data measuring receptor occupancy under different conditions for what seems to be TNF α and antibody (likely adalimumab) interactions. Let me break down the key aspects of this dataset:

- Data Structure:**
 - The file contains comma-separated values with columns for species, assay type, concentrations, time, and measurements
 - Key columns include L_pM (likely ligand/TNF α concentration), Ab_pM (antibody concentration), and receptor_occupancy (output measurement)
- Experimental Design:**
 - Two distinct assay types are present:
 - "varying TNF α ": where antibody concentration is fixed at 30 pM
 - "varying antibody": where TNF α concentration is fixed at 30 pM
- Concentration Ranges:**
 - Both TNF α and antibody concentrations span from 0.1 pM to 100,000 pM (0.1-100 nM)
 - Uses logarithmic spacing between concentration points
- Measurements:**
 - Receptor occupancy is measured as a percentage (values range from approximately -6 to 103%)
 - Time point appears constant at 172800 seconds (48 hours)
 - Includes a "constant_error" column suggesting standardized error measurements

This appears to be a typical dose-response dataset used to characterize the binding dynamics between TNF α , its receptor, and an inhibitory antibody in an *in vitro* setting.

Ask ABM Assistant

Automatic interpretation of the contents of a CSV File

- Type of Experiment
- Data Structure
- Experiment Design
- Ranges of Conditions

The data used to inform the model originate from *in vitro* assays measuring TNFR signaling in response to various concentrations of antibody and TNF α . In one assay, the antibody concentration is fixed while TNF α concentration is varied, and vice versa for the second assay. TNF α signal is measured via a luciferase reporter. Table 1 reports the number of data points and the tested concentrations.

```
[2]: ut.caption(
    "Table 1. Summary of data used for model calibration.",
    r"Data used included assays varying TNF\alpha concentrations " \
    "while fixing antibody concentrations and vice versa.")

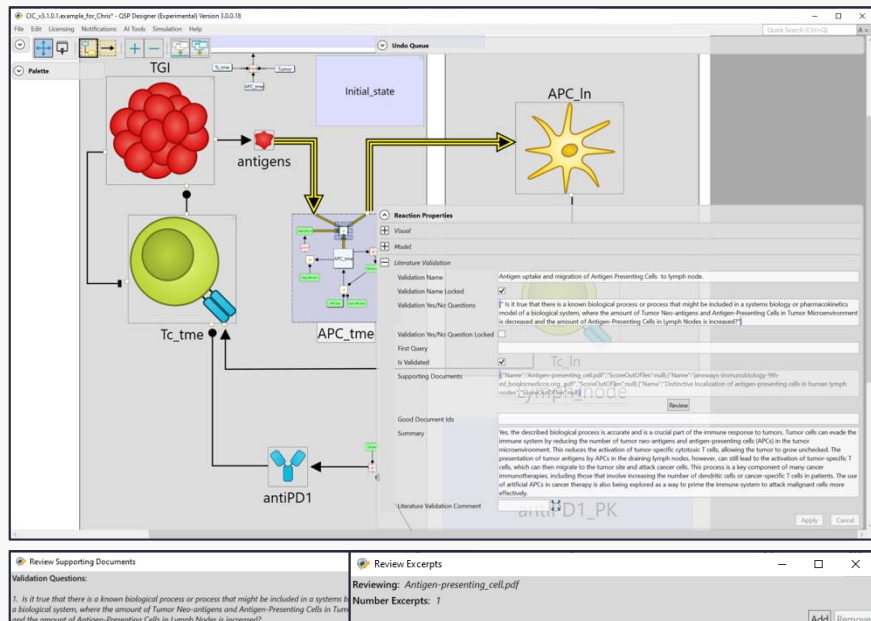
ut.summarize_data(
    data = pd.read_csv('data/invitro.csv'),
    group_variables = 'assay',
    columns = lambda grp,df: {
        'assay': grp,
        'n': df.shape[0],
        'fixed concentration (pM)': df.L_pM.iat[0] if "varying antibody" in grp
        'varying concentration range (pM)': f"{min(df.conc_pM)} - {max(df.conc_pM)}"
    }
).qsp.display(sigfigs=3)
```

Table 1. Summary of data used for model calibration. Data used included assays varying TNF α concentrations while fixing antibody concentrations and vice versa.

	assay	n	fixed concentration (pM)	varying concentration range (pM)
	varying TNF α	12	30	0.1 - 100000.0
	varying antibody	12	30	0.1 - 100000.0

Copyright 2026 Certara, Inc. All rights reserved.

Increase credibility with biology review by an AI expert



The credibility of a QSP model depends on how accurately the model represents the known biology

AI tools make it practical to do this at scale

Automatically generated GPT prompts based on model:

"Is it true that there is a known biological process or process that might be included in a systems biology or pharmacokinetics model of a biological system, where the amount of Tumor Neo-antigens and Antigen-Presenting Cells in Tumor Microenvironment is decreased and the amount of Antigen-Presenting Cells in Lymph Nodes is increased?"

AI generated summary of biological evidence:

Yes, the described biological process is accurate and is a crucial part of the immune response to tumors. Tumor cells can evade the immune system by reducing the number of tumor neo-antigens and antigen-presenting cells (APCs) in the tumor microenvironment. This reduces the activation of tumor-specific cytotoxic T cells, allowing the tumor to grow unchecked. The presentation of tumor antigens by APCs in the draining lymph nodes, however, can still lead to the activation of tumor-specific T cells, which can then migrate to the tumor site and attack cancer cells. This process is a key component of many cancer immunotherapies, including those that involve increasing the number of dendritic cells or cancer-specific T cells in patients. The use of artificial APCs in cancer therapy is also being explored as a way to prime the immune system to attack malignant cells more effectively.

Traceability to the primary literature:

are carried by lymph flow to the draining lymph nodes. Each lymph node is a collection point where APCs can interact with T cells.[1] During the migration, DCs undergo a process of maturation: they lose most of their ability to further engulf pathogens and they mature by changing surface expression of MHC and co-stimulatory molecules, as well as increased production of cytokines. The internalized antigen is digested into smaller peptides containing epitopes, which are then presented to T cells by the MHC.[2][12]

Predicting Clinical Outcomes from QSP Models (Chung et al.)

Quantitative systems pharmacology: Landscape analysis of regulatory submissions to the US Food and Drug Administration

Jane P. F. Bai | Justin C. Earp | Jeffry Florian | Rajanikanth Madabushi |
David G. Strauss | Yaning Wang | Hao Zhu

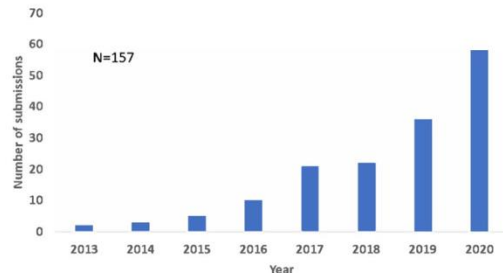
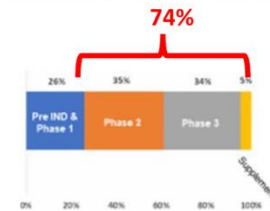
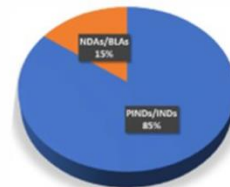


FIGURE 1 The yearly trend of quantitative systems pharmacology submissions since 2013

QSP Submissions Equally Span Different Phases of Drug Development



- 66% were efficacy related submissions,
- 97% supporting dosing:
 - 27% phase 1, 39% phase 2, 18% phase 3 & PSP, 12% NDA/BIA
- 34% were safety related: liver toxicity, bone density, cardiotoxicity, plasma ion conc

www.fda.gov

CPT Pharmacometrics Syst Pharmacol. 2021;10:1479-1484.

5

Inflammatory Bowel Disease
QSP MODEL



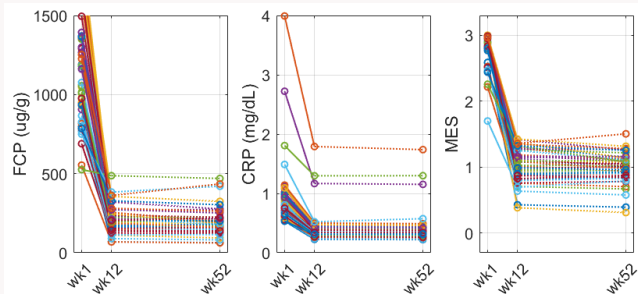
Biomarkers

**Clinical
Endpoints**

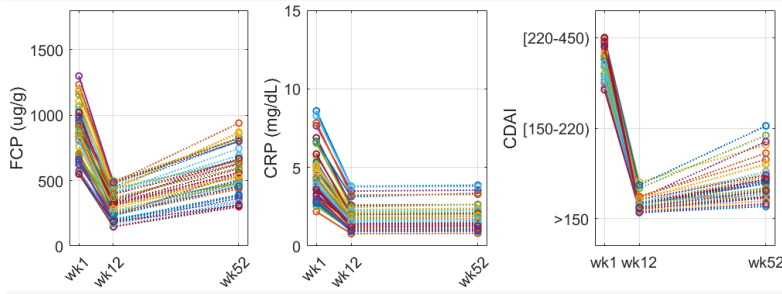
**Presented at ACoP14, Nov. 2023*

The IBD QSP Model Matches Clinical Trials

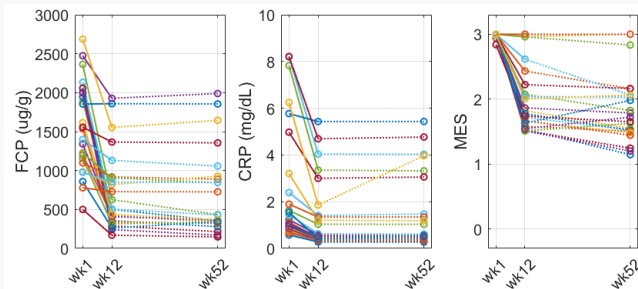
UC Seed VPs - Responders (MES $\leq$ 1 at Wk 12)



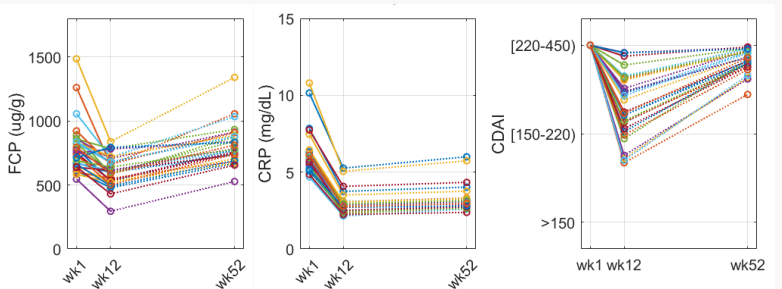
CD Seed VPs - Responders (CDAI $\leq$ 150 at Wk 12)



UC Seed VPs - Inadeq Resp (MES $\geq$ 2 at Wk 12)



CD Seed VPs - Inadeq Resp (CDAI $>$ 150 at Wk 12)



IBD QSP Model
matches clinical trial
studies:

RIPK1 Inhibitor, GSK
60mg TID

(Weisel et al. 2021)

Anti-IL-23p19,
Mirikizumab 600mg SC
Q4W

(Sandborn et al. 2019)

(Sands et al. 2022)

Anti-TNF α , Adalimumab
160 mg/80 mg SC Q2W

(Loftus Jr et al. 2020)

(Shinzaki et al. 2021)

Anti-TNF α and Anti-IL-
23, VEGA trial

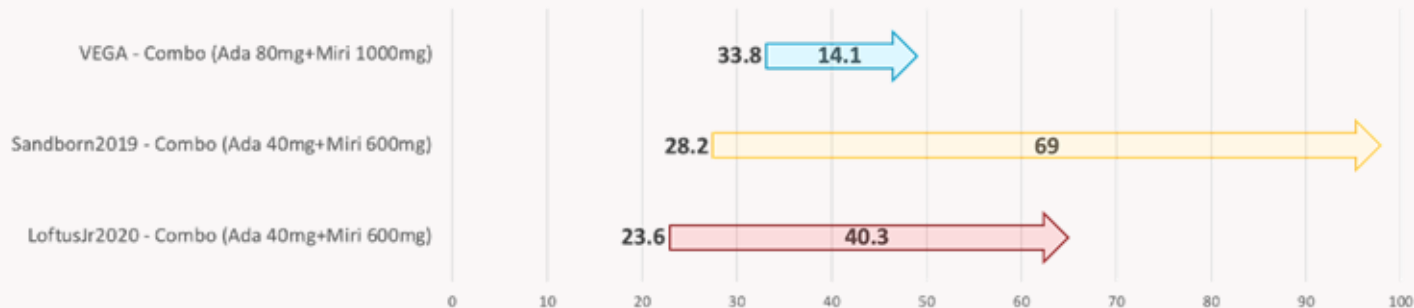
(Feagan et al. 2023)

The IBD QSP model VPs and Vpop capture observed variability at pre- and post-treatment using various treatment modalities

Predicting Clinical Outcomes from QSP Models (Chung et al.)

UC Clinical Trials	Adalimumab 80mg Q2W	Mirikizumab 600mg Q4W	Mirikizumab 1000mg Q4W	Ada 40mg + Miri 600mg	Ada 80mg + Miri 1000mg
VEGA (Feagan et al. 2023) anti-TNF α /anti-IL-23				Original Design → Increase Dose	
Sandborn et al. 2019 Mirikizumab		Original Design → Add Ada			
Loftus Jr et al. 2020 Adalimumab	Original Design → Add Miri				
	Anti-TNF α monoRx	Anti-IL-23 monoRx		Anti-TNF α / Anti-IL-23 comboRx	

Percentage Clinical Response (Mayo Endoscopic Score ≤ 1)



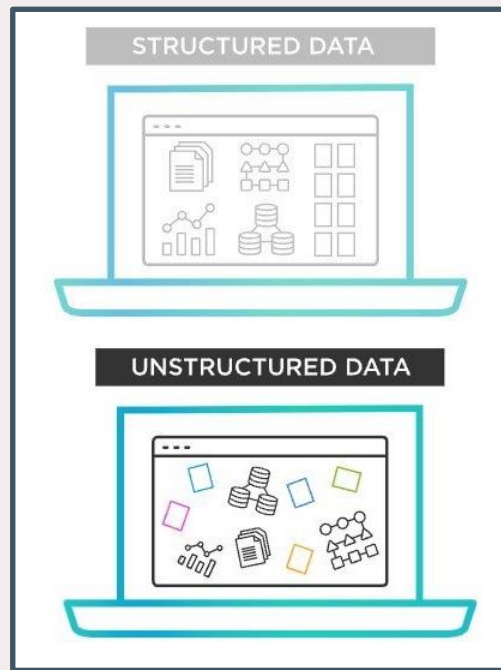
- The IBD QSP model predicts that the percentage clinical response of patients in previous clinical trials could have been increased with alternate dosing regimens

Unstructured Data Extraction

The Challenge of Unstructured Data

Making the “unusable” usable

- Most analyses rely on “structured” data
- However, **~80% of enterprise data is unstructured**, i.e., “trapped” in text, images, PDFs, etc.
- This content is still important, but it’s **hard to access and analyze**
- Our goal, then, is to **extract key data** to present alongside structured content



Granular Data Extraction via GPTs

Add structure to unstructured data sets

- GPTs can extract highly targeted data points utilizing natural language questions/prompts
- Coupled with RAG, we can automatically generate datasets with content linked back to source documents
- This allows researchers to “read a bunch of papers” without needing to read them



What is the C_{max} ?

What are the reported efficacy outcomes?

Describe the route of administration.

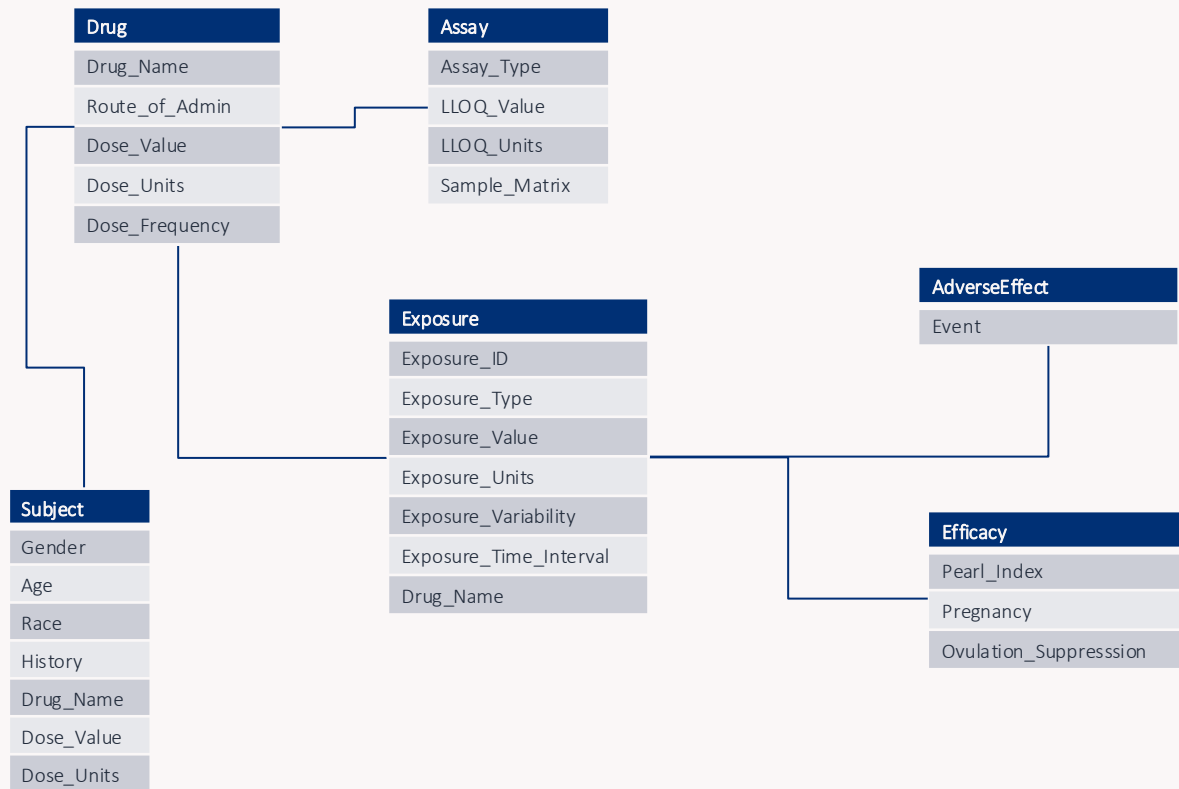
List the drugs mentioned in this study.

Were there any adverse events?

What is the dose frequency?



Data relationships need to be preserved when extracting data points



An example schema of some PK/PD data; without this context individual numbers are meaningless!

GPTs Can Programmatically Model Data Relationships

Utilizing Chained GPT Prompts Preserves Relationships Inherent in the Data

- The further context that can be given to a GPT, whether through specificity in the prompt or through RAG, significantly enhances its ability to draw inferences
- By chaining prompts, i.e. using the answer from one prompt to provide the basis for another, we can begin to establish the relationships between data points that are disjointed/separated within the text
- With some minor tweaking and prompt engineering, we can begin to programmatically extract granular data points while preserving the relationships between them from raw text (ex. `drugName >> drugDose`)

Python Example:

```
def run_extraction(source_id):
    ex = Extractor(test_settings)
    ex.run_top_query(query=f"You extracted these sections from a clinical trial document: {context[id]}\nWhat study arms were reported in the PK section?",
                    formatting_instructions="Always format your answer with this typescript definition: {'answer': list}",
                    sources=[],
                    var_name="arms",
                    output_type="list")
    ex.run_top_query(query=f"You extracted these sections from a clinical trial document: {context[id]}\nWhat metrics were recorded in the PK section?",
                    formatting_instructions="Always format your answer with this typescript definition: {'answer': list}",
                    sources=[],
                    var_name="endpoints",
                    output_type="list")
    ex.run_top_query(query=f"You extracted these sections from a clinical trial document: {context[id]}\nWhat drugs are being studied? If there is a drug, what is the drug name?",
                    formatting_instructions="Always format your answer with this typescript definition: {'answer': list}",
                    sources=[],
                    var_name="drugs",
                    output_type="list")
    ex.run_chain_query(query=f"You extracted these sections from a clinical trial document: {context[id]}\n\nFor the reported {} endpoint what were the drug names?",
                    formatting_instructions="Always format your answer with this typescript definition: {'answer': list}",
                    sources=[],
                    top_var_name=["endpoints"],
                    var_name="metrics",
                    output_type="list")
    ex.run_chain_query(query=f"You extracted these sections from a clinical trial document: {context[id]}\n\nFor the {} arm of the drug {} what was the drug dose?",
                    formatting_instructions="Always format your answer with this typescript definition: {'answer': str}",
                    sources=[],
                    top_var_name=["arms", "drugs", "endpoints"],
                    var_name="metrics",
                    output_type="string")
```

Levonorgestrel Benchmarking: Granular Results

Data Extraction Pipeline Performance

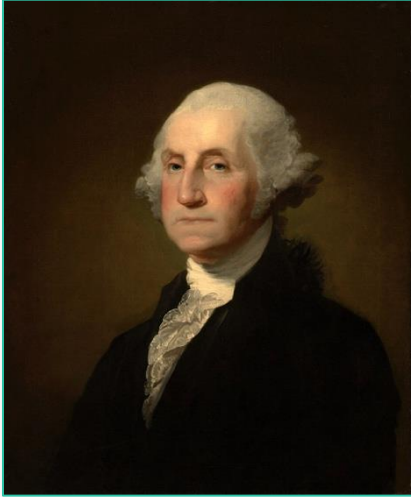
Data Point	Precision	Recall	Content Extracted
Drug Information	~89	~80	Drug name, active ingredients, dose, frequency, route,
Efficacy Information	~55	~50	Metric used, value, interval
Assay Information	~84.5	~77.5	Assay(s) used, Sample matrix
Exposure Information	~90	~57	Endpoints, values and units, interval, S.D.
Safety Information	~80	~65	Adverse Event types, number of patients, when event occurred.

Evaluation was conducted using 13 publications – a mix of trials, literature reviews, and meta-analysis. A team created manual annotations for all papers and these results were compared with model output.

Agentic AI in Report Writing

What do people mean by “Agentic AI?”

And how is that different from “generative” AI?



Generative: creates something new

“Write me a poem about George Washington.”



Agentic: uses reason to solve a complex task

“Do I need to bring an umbrella on a trip to Barcelona?”

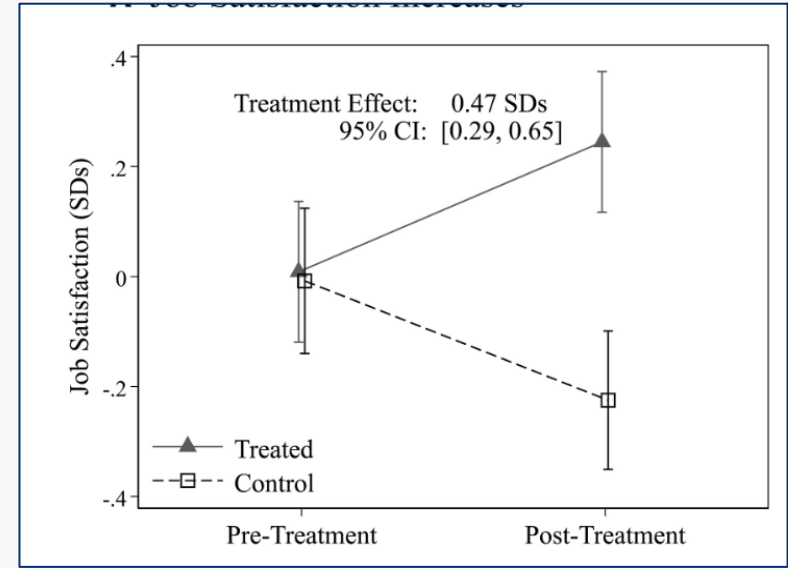
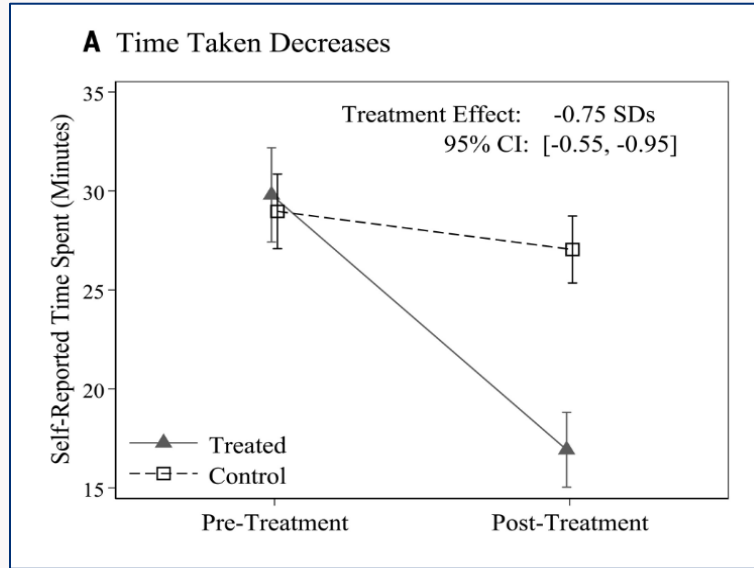
Limitations in the sciences

AI is like a colleague; you need to be able to check their work!

- Agents are “black boxes”
- Scientific use cases may have multiple potential solutions
- Reasoning relies on previous pattern recognition, not scientific knowledge
- GPTs are non-deterministic by design
- Agents require access to other systems and data



Improving Writing Productivity & Job Satisfaction



Randomized experiment shows leveraging GPTs for professional writing tasks present a 40% productivity increase while improving output quality by 18%.

Writers exposed to GPTs reported higher job satisfaction.

Shakked Noy, Whitney Zhang, Experimental evidence on the productivity effects of generative artificial intelligence. *Science* **381**, 187-192 (2023). DOI: [10.1126/science.adh2586](https://doi.org/10.1126/science.adh2586)

Regulatory Documents Are Templated by Design

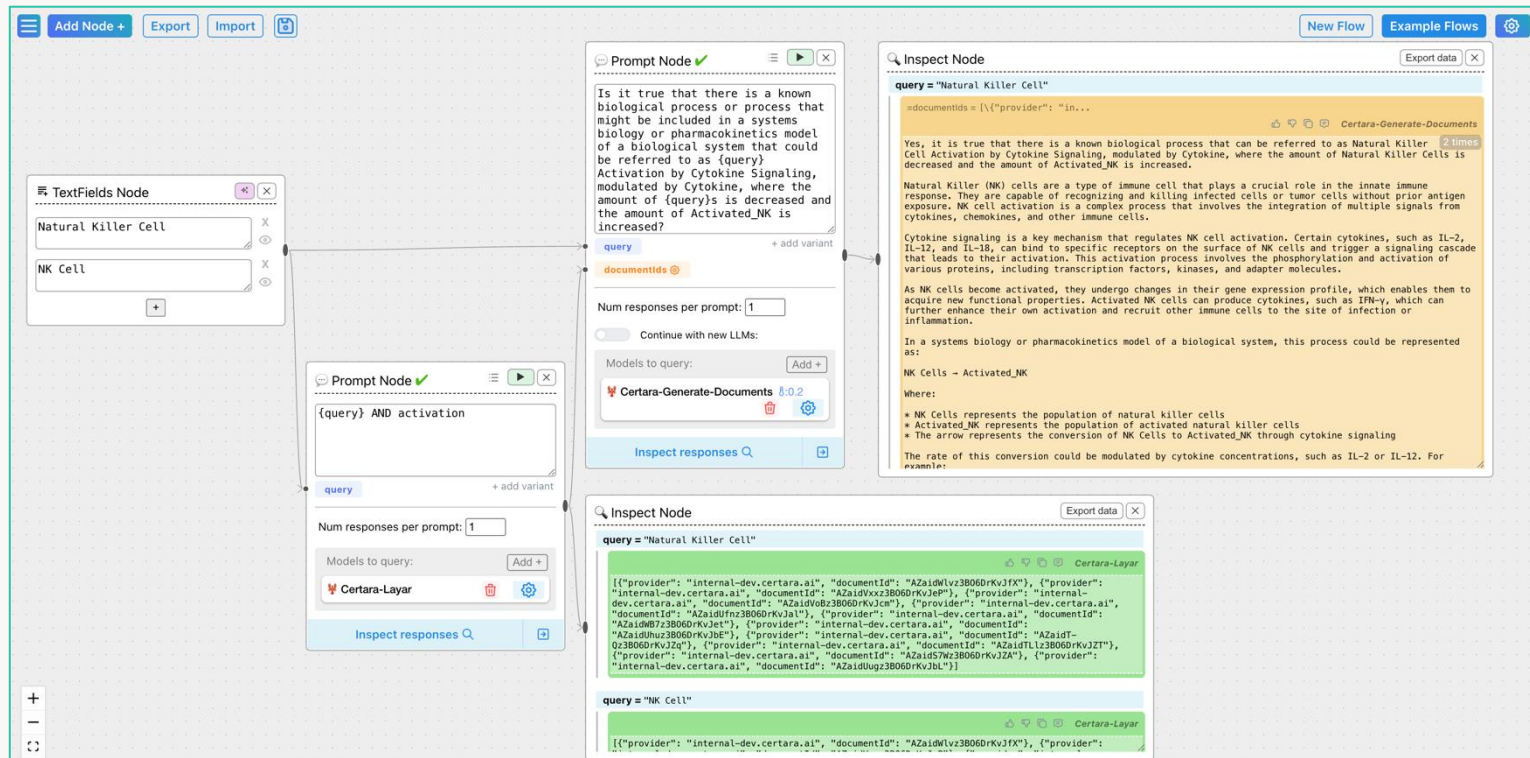
The screenshot displays the Certara CoAuthor software interface, which is used for creating and editing regulatory documents. The interface is divided into three main sections: a top ribbon, a central document editor, and a right-hand sidebar.

Top Ribbon: The ribbon includes tabs for Home, Insert, Draw, Design, Layout, References, Mailings, Review, View, and Acrobat. The Home tab is active, showing various formatting options like font face (Times New Roman), size (12), bold, italic, underline, and color. There are also options for bullet points, numbered lists, and indentation. The Design tab shows style galleries for AaBbCcDdEe, 1 AaBbCc, 1.1 AaBbCc, 1.1.1 AaBbCc, 1.1.1.1 AaBbCc, and AaBbCcDdEe. Other icons include Styles Pane, Dictate, Sensitivity, Add-ins, Editor, CertaraGPT, CoAuthor, Create PDF and share link, and Request Signatures.

Central Document Editor: The document is titled "10 SUBJECTS". It contains two main sections: "10.1 Disposition" and "10.2 Demographics". Each section has a placeholder "[Insert table here]". The "Disposition" section is highlighted with a blue box. The text in the "Disposition" section reads: "The disposition table provides a summary of the number of subjects in each category according to their treatment group for the clinical study. In the safety population, which includes all subjects who received the study medication, there were 60 subjects in both treatment groups. Similarly, the PK population and PK evaluable population also consisted of 60 and 15 subjects respectively in each treatment group. A total of 57 subjects in the safety population completed the study, with no discontinuations observed in the second treatment group. The reasons for discontinuation in the first treatment group included adverse events (3.33%) and withdrawal by subject (1.67%). However, no other reasons for discontinuation, such as failure to meet continuation criteria, lost to follow-up, physician decision, protocol violation, or other reasons were reported in either treatment group. In summary, the disposition table shows that both treatment groups had similar numbers of subjects in each category throughout the study. The majority of subjects completed the study with a low rate of discontinuation due to adverse events or withdrawal by subject observed only in the first treatment group."

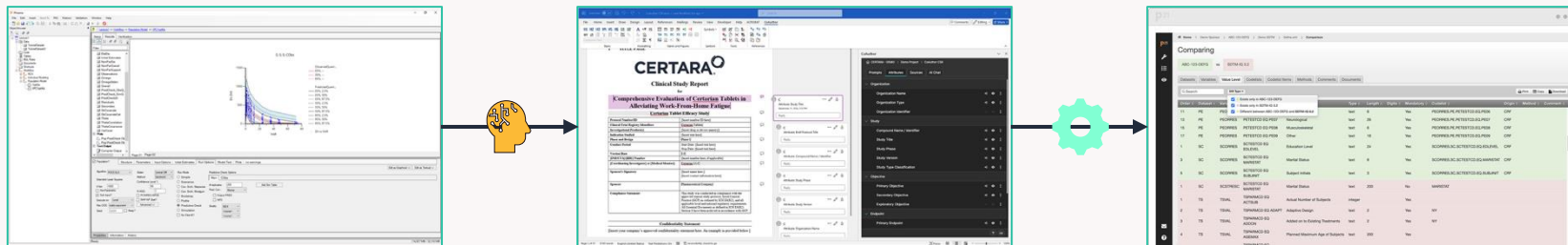
Right-hand Sidebar: The sidebar is titled "[TEST] CoAuthor" and shows the document's structure. It includes a "Nick's Demo" section with tabs for Prompts, Attributes, and Sources. The "Prompts" tab is active, showing a list of prompts. The first prompt is "Disposition Table Summary Copy". Below the prompts, there is a "TASK" section with a dropdown menu set to "Free Text". The "PROMPT" section contains a text box with the following text: "As a medical writer creating a CSR, summarize the attached disposition table with respect to the number of subjects in each category according to their treatment group." Below the prompt, there is an "Attachments (1)" section with a list of attachments. The first attachment is "t_10_1_Disposition DRUGAPIL.docx" with a document icon and a close button. At the bottom of the sidebar, there is a "Save" button and a "UN" button.

Chaining Activities Models Agentic Behavior



Chaining Activities Models Agentic Behavior

Creating a toxicology report



Scientist analyzes findings from study

AI generates report from findings and other content

Human adds reviewed report to submission

Summary

- GPT technologies are a novel and valuable addition to the life sciences toolkit
- While we may go through a hype cycle regarding GPTs, now is the time for life sciences, healthcare and governmental organizations to identify and derive value from these technologies
- RAG architectures enable specialized, secure, highly effective deployment of GPT technologies for life sciences specific tasks at a level that is similar to large GPT models and for a fraction of the cost
- There are numerous highly specialized applications of GPT technologies in the life sciences space. Identifying areas of value followed by focused application in these specialized areas is the critical next step for organizations who want to harness these technologies
- Expert knowledge of critical use cases is just as important now as it was before the advent of these technologies. Working with organizations that have the expertise in GPT technologies AND specialized use cases is critical



Questions?

Thank you!