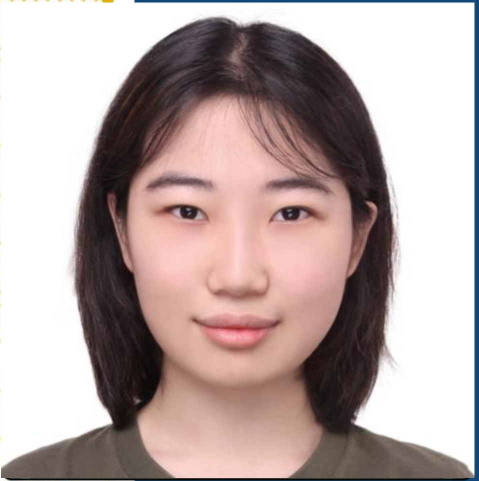
Tracil: AI-Powered Traceability Tool Across CDISC Standards

Kexin Guan, Scientist, Statistical Programming, Merck & Co., Inc.
Junze Zhang, Scientist, Statistical Programming, Merck & Co., Inc.
Anthony Chow, Executive Director, CDISC

Meet the Speakers



Kexin Guan

Title: Scientist, Statistical Programming

Organization: Merck & Co., Inc.

Kexin Guan is a Statistical Programmer at Merck & Co. Inc., where she has been part of the Oncology Early Development group since December 2022. She holds a Master's degree in Biostatistics and Bachelor's degree in Applied Mathematics and Statistics.

Junze Zhang

Title: Scientist, Statistical Programming

Organization: Merck & Co., Inc

Junze Zhang is a Scientist at Merck & Co. Inc., supporting early oncology statistical programming. He earned his Master's in Computer Engineering from NYU and Bachelor's in Computer Science from Oregon State University.

Disclaimer and Disclosures

- *The views and opinions expressed in this presentation are those of the author(s) and do not necessarily reflect the official policy or position of CDISC.*

- *The views and opinions expressed in this presentation are those of the author(s) and do not necessarily reflect the official policy or position of Merck & Co., Inc.*

Agenda

1. Background & Motivation
2. Demo
3. App Structure
4. Backend AI Workflow
5. Summary, Lessons Learned & Future Steps

Background & Motivation

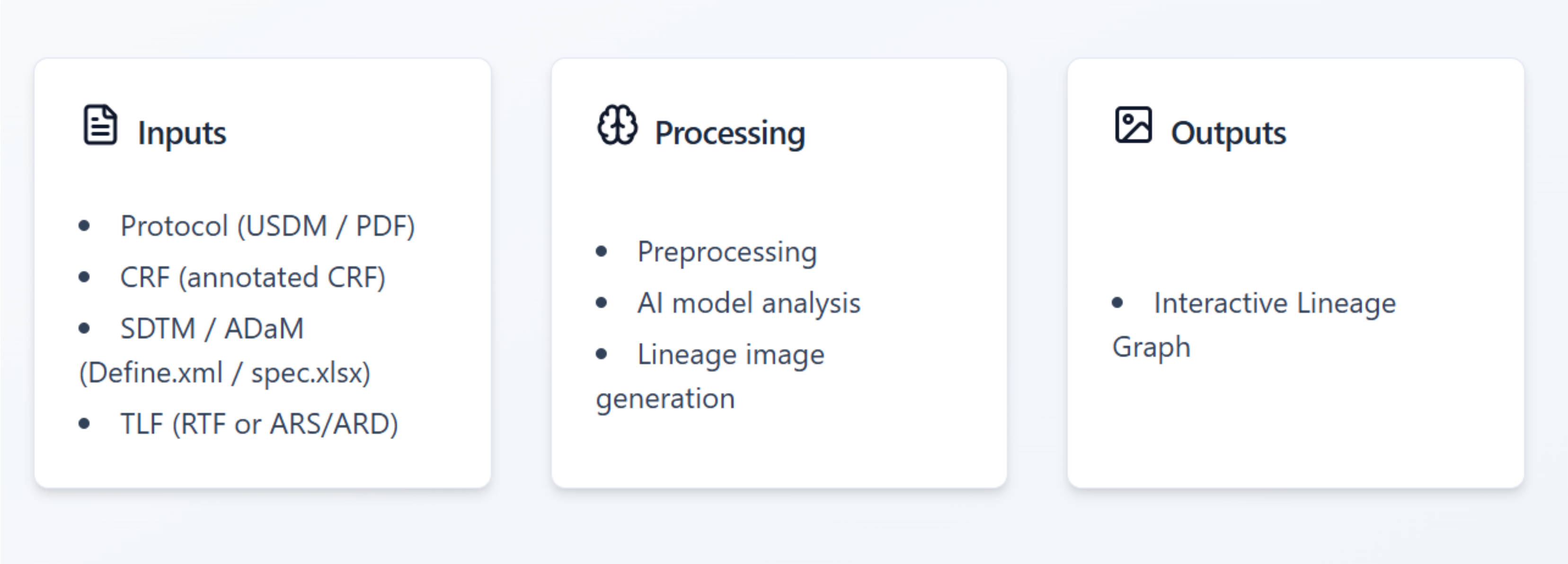
The Traceability Problem

- Clinical data flow is **complex,** spanning **multiple** systems and **silos**.
- Each layer (Protocol → CRF → SDTM → ADaM → TLFs) adds transformation logic.
- Manual tracing lineage = **heavy review time**.

Our Project Goals

- **Automate Lineage Inference**
  - Extract relationships from existing metadata without manual mapping.
- **Support CDISC Standards End-to-End**
  - Protocol (USDM) ↔ CRF ↔ SDTM (Define.xml) ↔ ADaM (Define.xml) ↔ TLF (ARS)
- **Provide Explainable AI Results**
  - Every link comes with a natural-language justification.
- **Deliver Human-Friendly Visualization All in One Place**
  - Interactive graphs for regulators, programmers, and statisticians.
- **Explore Data with Natural Language**

User Flow



Inputs

- Protocol (USDM / PDF)
- CRF (annotated CRF)
- SDTM / ADaM (Define.xml / spec.xlsx)
- TLF (RTF or ARS/ARD)

Processing

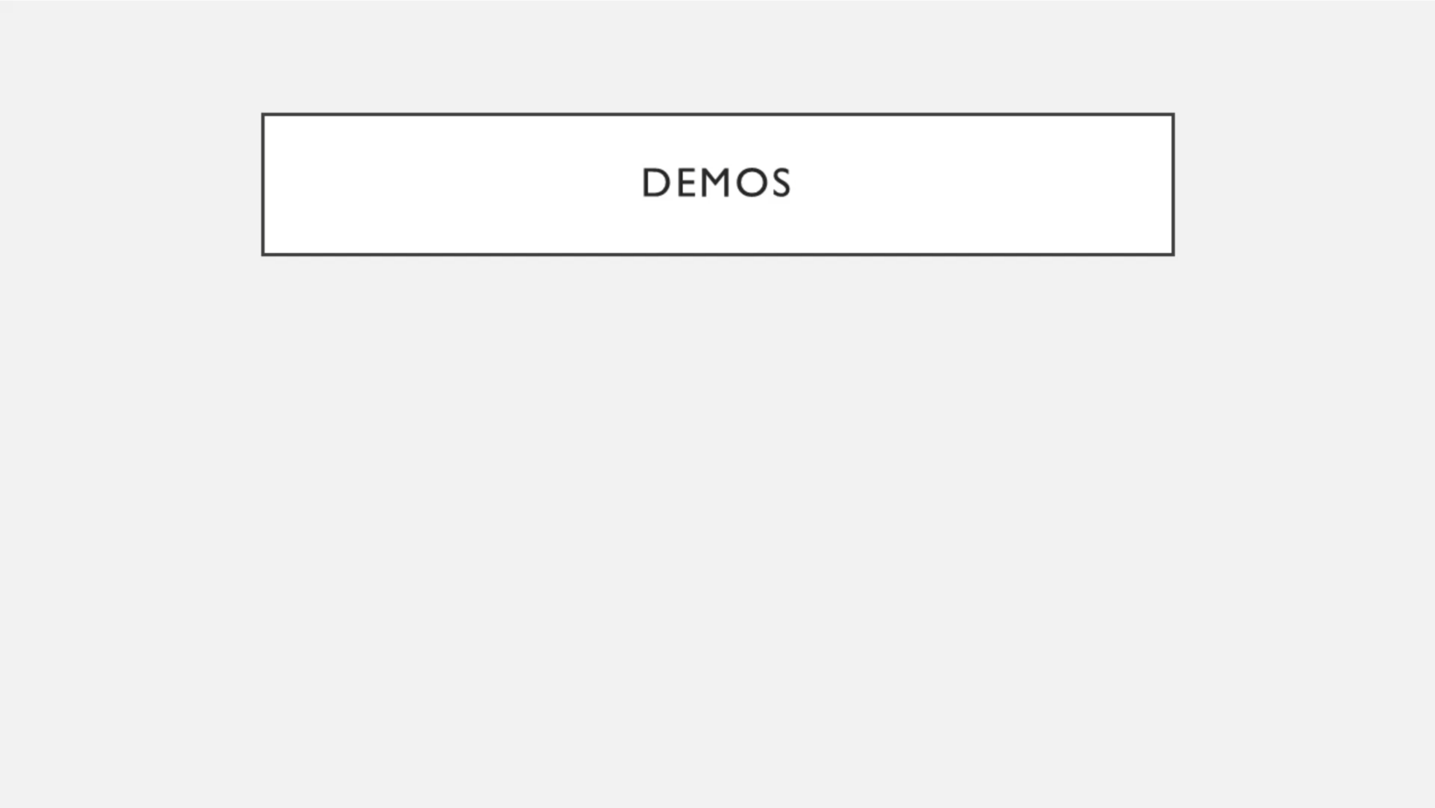
- Preprocessing
- AI model analysis
- Lineage image generation

Outputs

- Interactive Lineage Graph

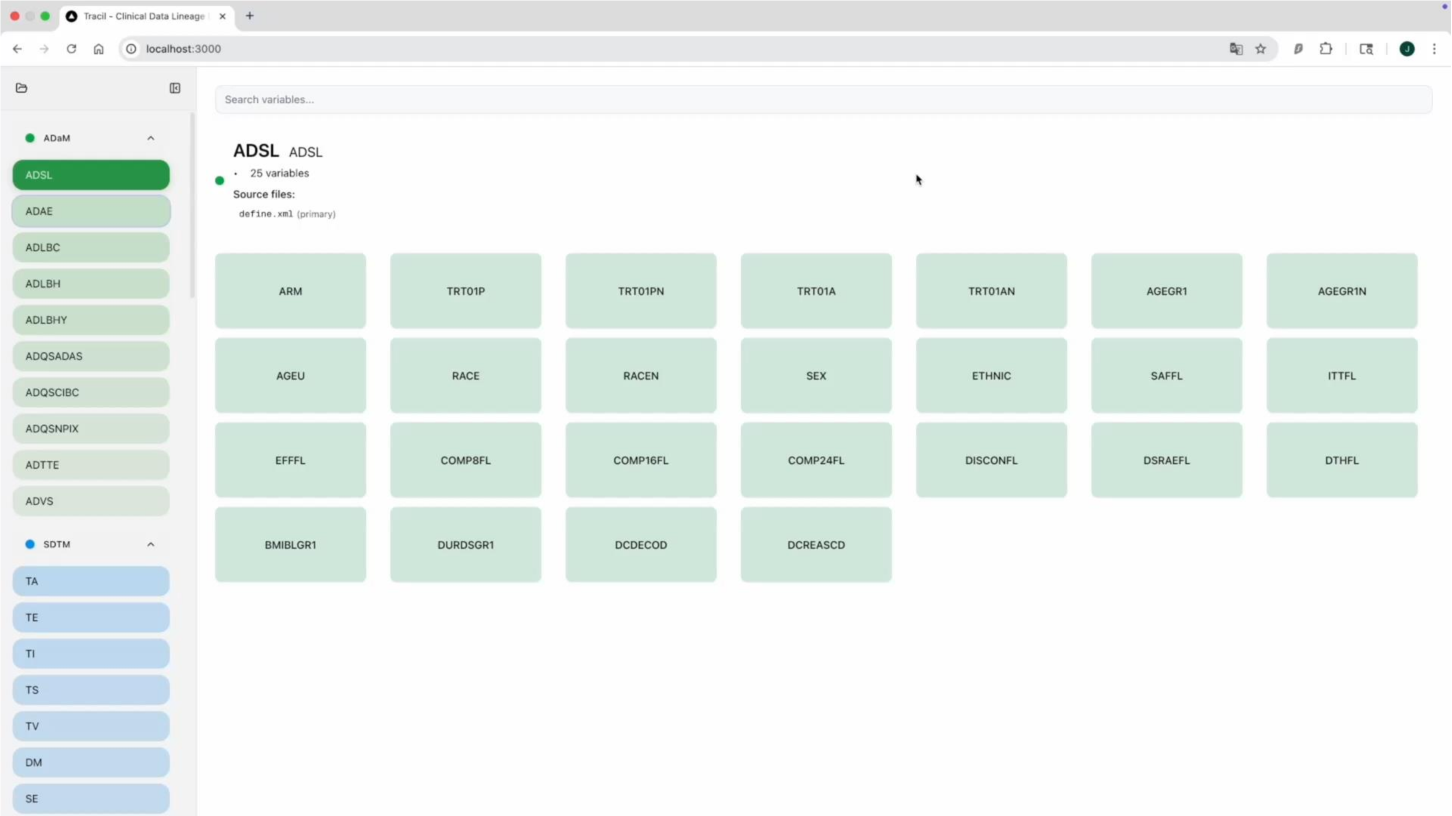
Demo

Upload & General UI



DEMOS

ADaM



ADSL ADSL

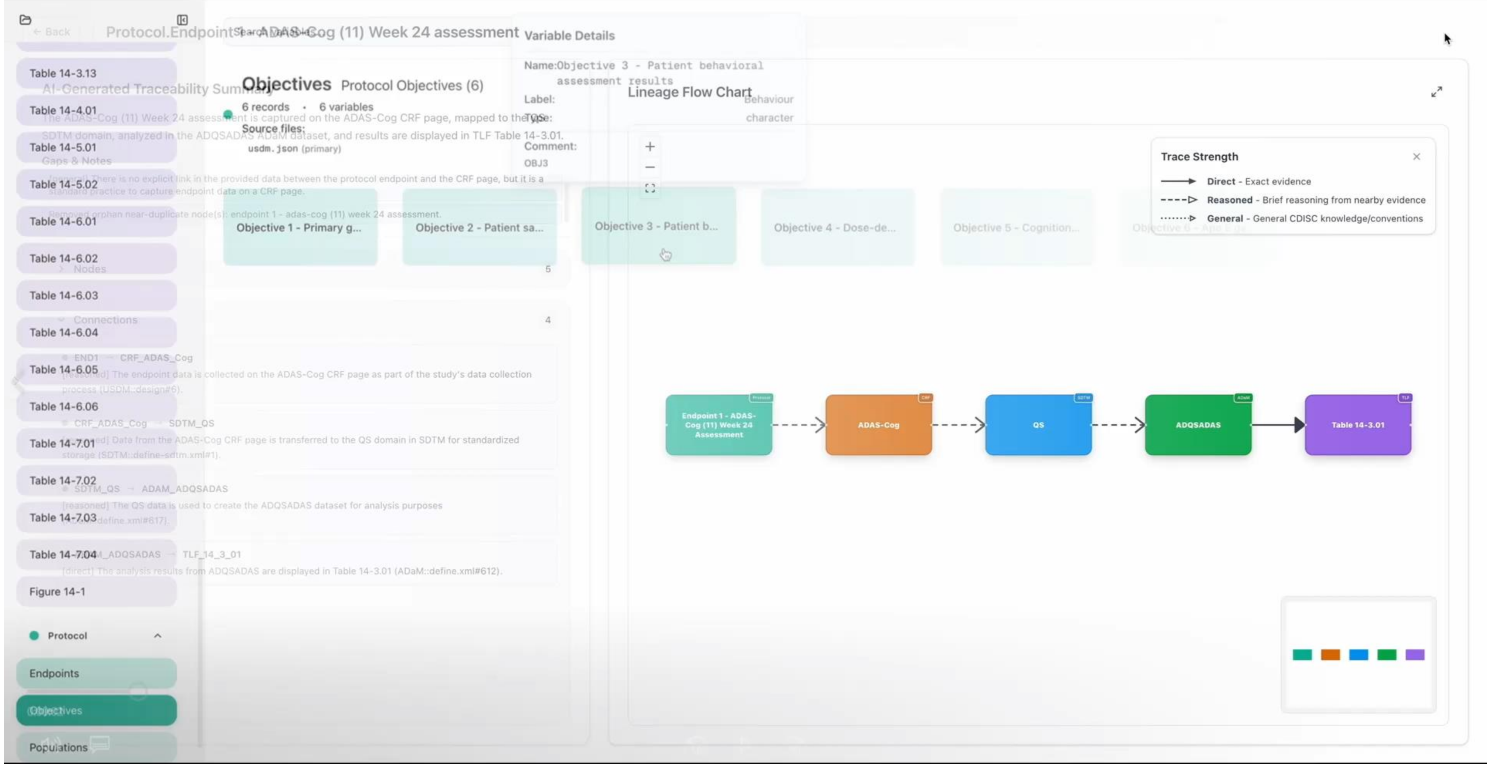
- 25 variables

Source files:

`define.xml` (primary)

ARM	TRT01P	TRT01PN	TRT01A	TRT01AN	AGEGR1	AGEGR1N
AGEU	RACE	RACEN	SEX	ETHNIC	SAFFL	ITTFL
EFFFL	COMP8FL	COMP16FL	COMP24FL	DISCONFL	DSRAEFL	DTHFL
BMIBLGR1	DURDSGR1	DCDECOD	DCREASCD			

Objectives



← Back

Protocol.Endpoint1 - ADAS-Cog (11) Week 24 assessment

Search variables...

AI-Generated Traceability Summary

The ADAS-Cog (11) Week 24 assessment is captured on the ADAS-Cog CRF page, mapped to the QS SDTM domain, analyzed in the ADQSADAS ADaM dataset, and results are displayed in TLF Table 14-3.01.

Gaps & Notes

[general] There is no explicit link in the provided data between the protocol endpoint and the CRF page, but it is a standard practice to capture endpoint data on a CRF page.

Removed orphan near-duplicate node(s): endpoint 1 - adas-cog (11) week 24 assessment.

> Nodes 5

⌄ Connections 4

END1 → CRF_ADAS_Cog

[reasoned] The endpoint data is collected on the ADAS-Cog CRF page as part of the study's data collection process (USDM::design#6).

CRF_ADAS_Cog → SDTM_QS

[reasoned] Data from the ADAS-Cog CRF page is transferred to the QS domain in SDTM for standardized storage (SDTM::define-sdtm.xml#1).

SDTM_QS → ADAM_ADQSADAS

[reasoned] The QS data is used to create the ADQSADAS dataset for analysis purposes (ADaM::define.xml#617).

ADAM_ADQSADAS → TLF_14_3_01

[direct] The analysis results from ADQSADAS are displayed in Table 14-3.01 (ADaM::define.xml#612).

Objectives Protocol Objectives (6)

6 records · 6 variables

Source files:
usdm.json (primary)

Objective 1 - Primary g...
Objective 2 - Patient sa...
Objective 3 - Patient b...
Objective 4 - Dose-de...
Objective 5 - Cognition...
Objective 6 - Apo E ...

Variable Details

Name: Objective 3 - Patient behavioral assessment results

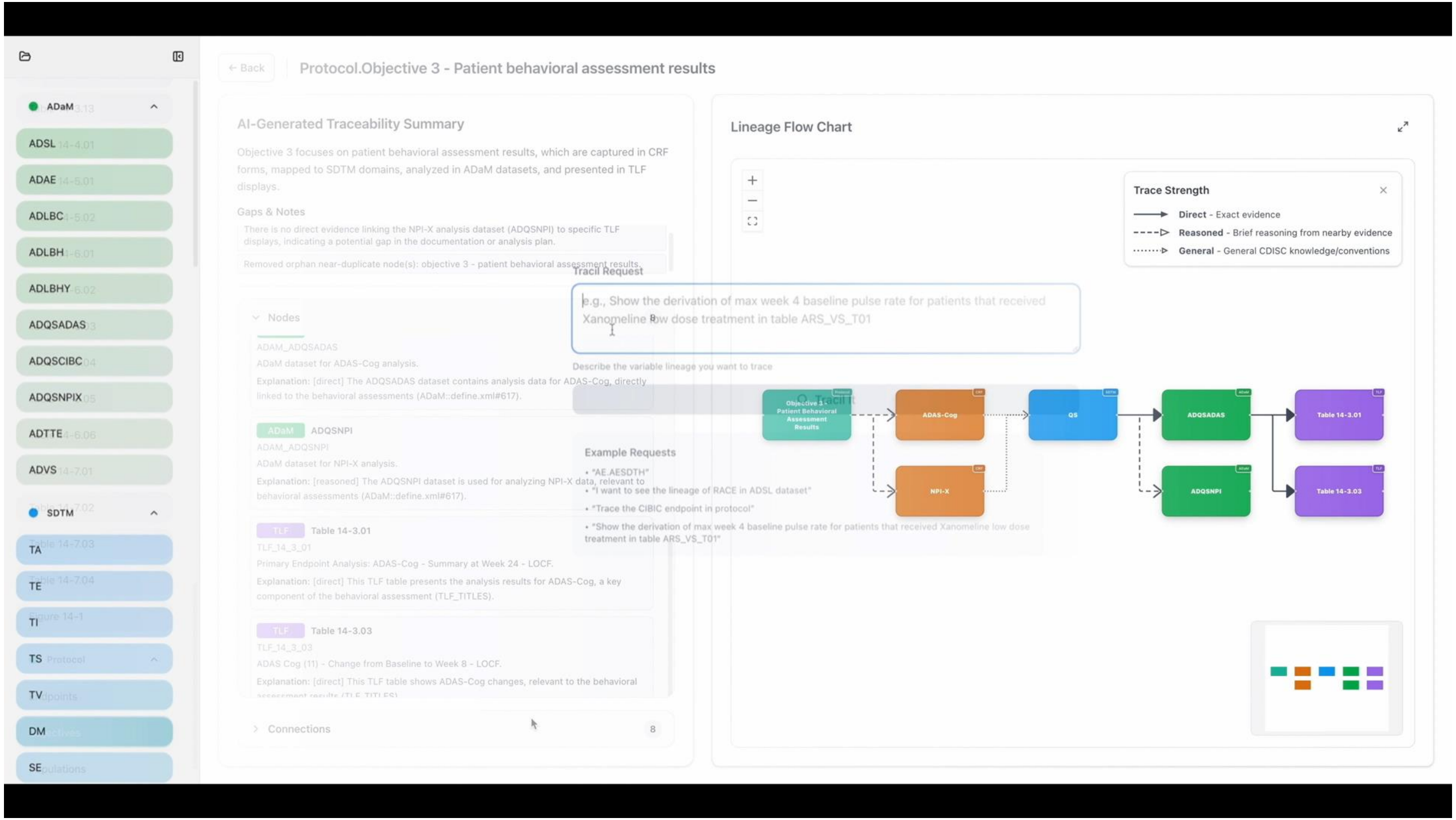
Label: Behaviour character

Type:

Comment: OBJ3

Lineage Flow Chart

Search With Natural Language



Protocol.Objective 3 - Patient behavioral assessment results

AI-Generated Traceability Summary

Objective 3 focuses on patient behavioral assessment results, which are captured in CRF forms, mapped to SDTM domains, analyzed in ADaM datasets, and presented in TLF displays.

Gaps & Notes

There is no direct evidence linking the NPI-X analysis dataset (ADQSNPI) to specific TLF displays, indicating a potential gap in the documentation or analysis plan.

Removed orphan near-duplicate node(s): objective 3 - patient behavioral assessment results.

Nodes

ADAM_ADQSADAS

ADaM dataset for ADAS-Cog analysis.

Explanation: [direct] The ADQSADAS dataset contains analysis data for ADAS-Cog, directly linked to the behavioral assessments (ADaM::define.xml#617).

ADaM ADQSNPI

ADAM_ADQSNPI

ADaM dataset for NPI-X analysis.

Explanation: [reasoned] The ADQSNPI dataset is used for analyzing NPI-X data, relevant to behavioral assessments (ADaM::define.xml#617).

TLF Table 14-3.01

TLF_14_3_01

Primary Endpoint Analysis: ADAS-Cog - Summary at Week 24 - LOCF.

Explanation: [direct] This TLF table presents the analysis results for ADAS-Cog, a key component of the behavioral assessment (TLF_TITLES).

TLF Table 14-3.03

TLF_14_3_03

ADAS Cog (11) - Change from Baseline to Week 8 - LOCF.

Explanation: [direct] This TLF table shows ADAS-Cog changes, relevant to the behavioral assessment results (TLF_TITLES).

Connections 8

Tracil Request

e.g., Show the derivation of max week 4 baseline pulse rate for patients that received Xanomeline low dose treatment in table ARS_VS_T01

Describe the variable lineage you want to trace

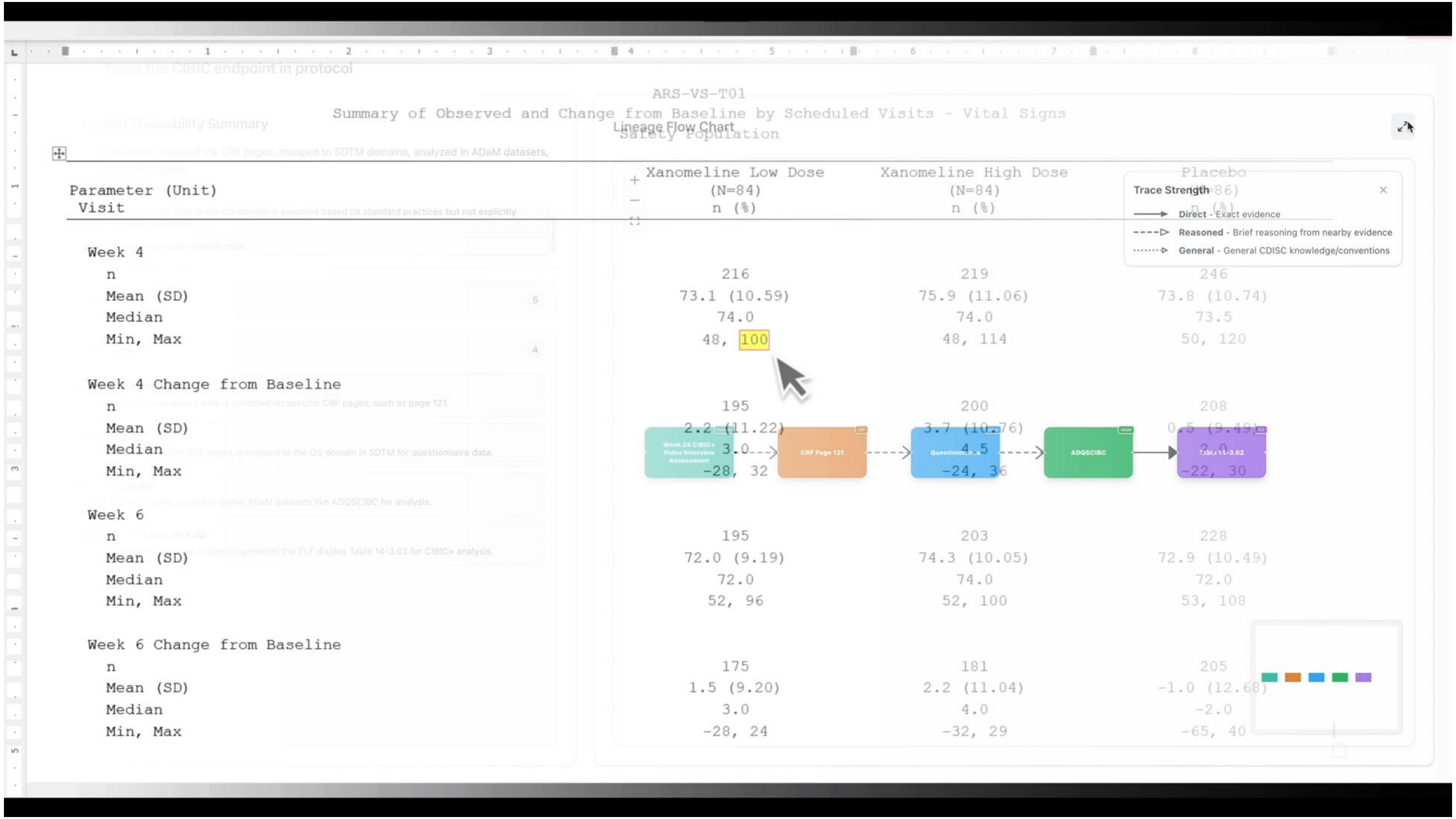
Example Requests

- "AE.AESDTH"
- "I want to see the lineage of RACE in ADSL dataset"
- "Trace the CIBIC endpoint in protocol"
- "Show the derivation of max week 4 baseline pulse rate for patients that received Xanomeline low dose treatment in table ARS_VS_T01"

Trace Strength

- Direct - Exact evidence
- Reasoned - Brief reasoning from nearby evidence
- General - General CDISC knowledge/conventions

TLF Details with ARS



Trace the CIBIC endpoint in protocol

Limited Traceability Summary

ARS-VS-T01

Summary of Observed and Change from Baseline by Scheduled Visits – Vital Signs

Safety Population

Lineage Flow Chart

Parameter (Unit) Visit	Xanomeline Low Dose (N=84) n (%)	Xanomeline High Dose (N=84) n (%)	Placebo (N=86) n (%)
Week 4			
n	216	219	246
Mean (SD)	73.1 (10.59)	75.9 (11.06)	73.8 (10.74)
Median	74.0	74.0	73.5
Min, Max	48, 100	48, 114	50, 120
Week 4 Change from Baseline			
n	195	200	208
Mean (SD)	2.2 (11.22)	3.7 (10.76)	0.5 (9.49)
Median	3.0	4.5	2.0
Min, Max	-28, 32	-24, 36	-22, 30
Week 6			
n	195	203	228
Mean (SD)	72.0 (9.19)	74.3 (10.05)	72.9 (10.49)
Median	72.0	74.0	72.0
Min, Max	52, 96	52, 100	53, 108
Week 6 Change from Baseline			
n	175	181	205
Mean (SD)	1.5 (9.20)	2.2 (11.04)	-1.0 (12.68)
Median	3.0	4.0	-2.0
Min, Max	-28, 24	-32, 29	-65, 40

Trace Strength

- ⟶ **Direct** - Exact evidence
- - - -▷ **Reasoned** - Brief reasoning from nearby evidence
- ·······▷ **General** - General CDISC knowledge/conventions

...is assessed via CRF pages, mapped to SDTM domains, analyzed in ADaM datasets,

...of CRF data to the QS domain is assumed based on standard practices but not explicitly

...CIBIC+ endpoint data is collected on specific CRF pages, such as page 121.

...on CRF pages is mapped to the QS domain in SDTM for questionnaire data.

...SDTM QS data is used to derive ADaM datasets like ADQSCIBC for analysis.

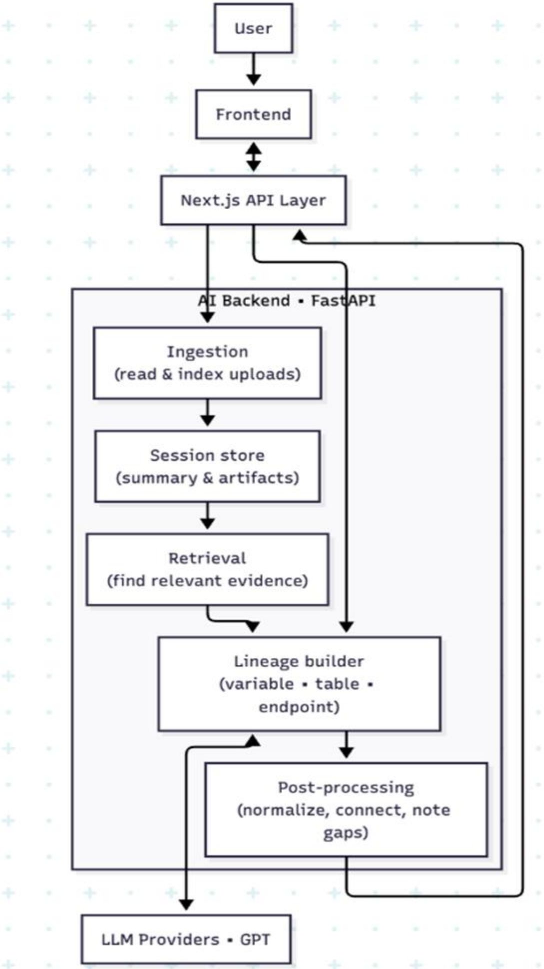
Table_14-3.02

...is used to generate the TLF display Table 14-3.02 for CIBIC+ analysis.

App Structure

App Structure

- Frontend: Javascript with React/Next.js
- Backend: FastAPI
- AI core: Python AI Engine
  - GPT-4o (mini)
  - Text-embedding-3-model from OpenAI



Backend AI Workflow

How Tracil’s Backend Works

- **Pre-processing**: Python code to convert structured (XML, JSON) and semi-structured (PDF/CRF) inputs into a unified, machine-readable JSON file

- **LLM Reasoning:** passes the inputs to an LLM reasoning layer to infer variable derivations, dependencies, and gaps, returns a JSON graph

- **Post-processing:** standardize the JSON lineage graph so that it can be visualized interactively in the frontend UI

Pre-Processing: Parsing & Normalization Layer

Input formats supported: Specification (.xlsx), define.xml, aCRF, ARS JSON, Protocol PDF, USDM, and TLFs (RTF)

FastAPI endpoint `/process-files`:

- Extracts variable metadata, derivation notes, and dataset context
- Normalizes the information extracted into unified schema of names, domain tags, and relationships information

Why important:

- Provides the frontend with structured data for visualization and user interaction.
- gives LLM a clean, consistent input so it can reason semantically

LLM Lineage Builders

Three Routes by Target

- ○ Detects what you’re tracing: protocol endpoint, ADaM/SDTM variable, table/cell; sends it to the right builder

Gather the Evidence

- ○ Collects all supporting pre-processed metadata from the session (aCRF index, protocol text, unified JSON returned by /process-files API, etc.)
- ○ These documents form the “evidence base” for reasoning

Find What Matters (Chunk + Retrieve)

- ○ Splits large documents into small readable sections
- ○ Converts each into numerical “embeddings” so the AI can compare meanings
- ○ Selects only the Top K (≈12) most relevant pieces to focus the analysis

Ask the AI Model

- ○ Sends the selected context to GPT-4o for reasoning (mini model as fallback).
- ○ The model returns a structured JSON lineage, showing each variable, link, and explanation.

Parse + Check

- ○ Cleans the graph and validates to ensure it’s proper JSON

Prompt Design

- **Clear Role Definition:**
  - *System* = “Senior CDISC standards expert”; explicit backtrace & forward instructions; closed node types and canonical IDs (e.g., `ADSL.AGE`, `DM.BRTHDTC`)
- **Structured Thinking:**
  - Each task follows a fixed schema (variable, endpoint, or table) so the AI always knows what format to produce
- **Evidence packing:**
  - The AI reviews only the top relevant document sections, tagged as evidence, before reasoning; pins the target node (e.g., `Target variable: ADSL.TRT01AN`)
- **Explainable Results:**
  - Every link in the lineage includes a short explanation starting with [direct], [reasoned], or [general], and cites where the information came from (define.xml, ARS, CRF, protocol)
- **Direction rules:**
  - hard requirement to emit edges upstream → downstream only; forbid illegal shortcuts (e.g., ADaM → CRF)
- **Resilience:**
  - chat called with `temperature=0.0` + `response_format={"type":"json_object"}`; Built-in checks handle small formatting issues automatically, so results remain valid and reproducible

Post-Processing: Normalize, Validate, Connect

Normalize the Structure

•Add tags and standardize names so the graph speaks one language. (e.g., `ADVS.AVAL → ADaM variable, VS.VSORRES → SDTM variable`)

•Fix missing labels and edge directions for clear flow (`source → result`)

Validate the Connections

•Check each link has real evidence; flag gaps if missing

•Remove duplicates/orphans, and add short explanations

Connect Missing Pieces

•Re-query files when links are incomplete (e.g., ADaM variable with no SDTM parent)

•Ensure full trace: Protocol → CRF → SDTM → ADaM → TLF

Conclusion & Future Steps

Key Takeaways

- Tracil automates lineage across Protocol → CRF → SDTM → ADaM → TLF using AI reasoning
- Converts protocol endpoints, aCRF, define.xml, specifications, ARS/ARD into a **unified JSON schema**
- Provides **explainable AI outputs** with clear variable relationships
- Makes traceability **easy, fast, interactive, and follows CDISC standards.**

Limitations

- **Model Accuracy & Stability:** Same input can yield slightly different results due to the nature of LLM
- **Limited Data:** Few open, CDISC-compliant datasets restrict realistic fine-tuning
- **Validation Gap:** Need standardized validation methods to ensure reliable outputs

Future Steps & Vision

- **Model Expansion:** Test across different LLMs (GPT-5, Gemini, Claude, etc.)
- **Confidence Scoring + User Feedback:** Quantify AI certainty and learn from human corrections

Thank You!

Contact Information

- Junze Zhang
  - Junze.zhang@merck.com
- Kexin Guan
  - Kexin.guan@merck.com
- Anthony Chow
  - Achow@cdisc.org

- Tracil GitHub Repository:
  - https://github.com/1mgroot/Tracil