



## Getting Started Guide



February 2026

# Contents

What is the Asta ecosystem?	3
Asta agents	4
Find papers	5
Generate a report	7
Analyze data	9
AstaLabs	11
AutoDiscovery	12
Paper + Figure QA	14
Asta Preview	15
Using AstaBench	16
Trying Asta resources	20
What's next for Asta?	22

## What is the Asta ecosystem?

Asta is Ai2's initiative to accelerate science by building reliable and capable agentic assistants for researchers, alongside a comprehensive benchmarking system and developer tooling. As AI use expands across the sciences, researchers in every field need transparent systems they can understand, verify, and reproduce. Asta is designed to fill this need.

The Asta ecosystem brings together three essential components to ensure integrity and rigor in scientific AI:

- 1 **Asta**, a user-facing agentic tool to assist human researchers performing complex, real-world scientific tasks.
- 2 **AstaBench**, a rigorous, domain-relevant benchmark suite for quantifying the accuracy and performance of any scientific agent.
- 3 **Asta resources**, a set of bespoke software components and standards to help build, test, and refine scientific agents.

## Asta agents

Asta's scientific agents are designed to support researchers who need to navigate large volumes of information, analyze data, and generate insights efficiently—without compromising on rigor:

- **Find papers** helps you discover relevant research by reformulating your query into multiple search strategies and traversing citation graphs to explain exactly why each paper is relevant.
- **Generate a report** scans abstracts and full-text papers to construct a structured document, providing in-line, interactive citations that link directly to the source evidence for immediate verification.
- **Analyze data** turns natural language questions into structured, reproducible analyses. Instead of just giving an answer, Asta writes and executes Python code to explore datasets, generate hypotheses, run statistical tests, and generate visualizations.

Currently, Asta can search across a corpus of around 12.4 million full-text papers and more than 108 million abstracts.

---

The Asta homepage features a prompt field in the center and tabs for switching between **Find papers**, **Generate a report**, and **Analyze data**. You can enter your own query or click one of the suggested examples below the prompt to get started. Your recent queries appear in the left-hand panel for quick access.

When you're logged into Asta, queries are saved in the left-hand panel. You can return to the results of any query by clicking on the title of the query from the panel.

You can share the results of queries via the Share button in the upper-right corner. Clicking it copies a pasteable link to your clipboard.

# Asta agents: Find papers

You can use Asta's Find papers feature, also known as [PaperFinder](#), to discover relevant research across scientific fields of study. It breaks down your query into components, searches for papers, follows citations, and then presents short summaries of why the paper is relevant to your specific search.

- 1 From the tabs above the prompt bar on the Asta homepage, select **Find papers**.
- 2 Enter a query—for example, “Papers about shallow marine ecosystem classification using hyperspectral imagery.” You'll find other example prompts below the prompt bar.
- 3 Asta will return relevant results, including excerpts from highly cited studies.
- 4 You can sort the papers by relevance, filtering out papers less pertinent to your search. You can also sort papers by year, title, venue, citations, or author.

The screenshot shows the Asta 'Find papers' interface. On the left, a search bar contains the query: "Research on generative document retrieval models that allow for quick addition of new documents as they become available". Below the search bar, it indicates "Researched for 32 seconds" and "I found 6 papers that look like perfect matches, 7 relevant ones and 49 others." The search results are displayed in a list format, with the first two papers highlighted. The first paper is "Continual Learning for Generative Retrieval over Dynamic Corpora" by Jiangui Chen, Ruqing Zhang, J. Guo, and 3 Authors, published in the International Conference on Information and Knowledge Management in 2023. The second paper is "DSI++: Updating Transformer Memory with New Documents" by Sanket Vaibhav Mehta, Jai Gupta, Yi Tay, and 5 Authors, published in the Conference on Empirical Methods in Natural Language Processing in 2022. The interface also includes a sidebar with sorting options (Sort by, Relevance, Criteria, Year, Venue, Author) and a search bar at the bottom for further refinement.

By default, Find papers runs in *fast mode*, retrieving fewer papers in the initial stage so you don't have to wait 2-3 minutes for each response. But you can ask Find papers to "work harder" by typing a query like "show an extensive set of papers about [topic]."

Once a search completes, you can ask a follow-up prompt (e.g., "Show me papers specifically referring to Pacific Ocean hyperspectral imagery") or start a new query.

# Asta agents: Generate a report

Asta's **Generate a report** capability, also known as [ScholarQA \(SQA\)](#), lets you ask scientific questions that require multiple documents to answer. The tool shows in-depth, detailed, and contextual results with content like table comparisons, expandable sections for subtopics, and citations with paper excerpts for verification.

Generate a report searches across an index of computer science, medicine, environmental science, and biology papers, updated weekly.

- 1 From the tabs above the prompt bar on the Asta homepage, select **Generate a report**.
- 2 **Enter a query**—for example, “What are some non-linear user interfaces for interacting with LLMs?”
- 3 **You'll get a summarized answer** that draws on content from a number of papers, along with in-line citations and links to those papers.

The screenshot displays the Asta 'Generate a report' interface. On the left, a sidebar contains navigation options like 'New exploration', 'Recent paper searches', 'Recent reports', and 'Recent data analyses'. The main area shows a search query: 'What are the differences and similarities between information foraging theory and exploratory search?'. Below the query, it indicates 'Researched for 25 seconds' and provides a 'View steps' link. A synthesized report is displayed, titled 'Differences and Similarities Between Information Foraging Theory and Exploratory Search'. The report content includes sections such as 'Background on Information Foraging Theory (IFT)', 'Background on Exploratory Search', 'Conceptual Connections and Similarities between IFT and Exploratory Search', and 'Distinctive Aspects and Differences'. The interface also features a 'Share' button, a 'Sign Out' button, and a 'Download Report' button. At the bottom, there is a prompt bar for asking a scientific question and a 'Leave Feedback' link.

You can keep refining the same question across multiple turns without losing context or citations. Asta retains your chat history, tool outputs, filters, and corpus details across turns. Quick clarifications stay in chat, while deeper prompts trigger fuller analyses, always grounded in the literature with explicit citations, with any non-cited content or absent evidence transparently identified by a (Model-Generated) label.

If your intent is ambiguous – or no evidence is found – Asta asks a targeted clarifying question before proceeding to keep results precise and reliable.

Once your report is generated, Asta keeps you in the research loop with tools for deeper exploration:

- 1 Continue the conversation** with follow-ups in the same thread (e.g., refine scope, check conflicting findings, or request deeper analysis).
- 2 Reference specific papers directly** with @ (e.g., @Smith 2023) to interrogate them in-thread.
- 3 Run meta-queries** like “Which paper is most cited?” or “Filter for peer-reviewed studies only.”

# Asta agents: Analyze data

Asta's agent for data-driven discovery and analysis, also known as [DataVoyager](#), lets you ask plain-language questions about structured, user-uploaded datasets and receive transparent, reproducible results back. It supports common tabular formats including CSV/TSV, Excel (.xlsx), JSON/JSONL, HDF5, and Parquet.

- 1 From the tabs above the prompt bar on the Asta homepage, select **Analyze data**
- 2 **Upload your dataset** by clicking the paperclip icon (📎) in the prompt bar.
- 3 **Enter a question.** For better results, briefly describe the dataset and key columns (e.g., units, measurement cadence, and outcome variables). For example: "The dataset includes archival data on fatalities caused by hurricanes in the U.S. (1950–2012). Columns include: ind = row ID; year = event year." You can also ask targeted questions like, "Which treatment arm shows the steepest improvement after week 6?"
- 4 **Append your prompt with a request** – e.g., "What are informative analyses to run on this dataset?" – and DataVoyager will process the file and return a structured answer in the chat thread, along with copyable visuals, the code that generated them, and a documented methods section.

The screenshot displays the Asta DataVoyager interface. On the left, a sidebar shows the user's question: "In this U.S. savings behavior dataset, how do income levels influence how people save across different age groups?". Below the question, a file named "nls\_bmi\_processed.csv" is uploaded. The interface indicates that the analysis is complete and provides a "View findings" button. At the bottom left, there is a prompt bar for uploading a dataset and asking a question.

The main content area is divided into sections:

- Reasoning**: A section for the model's reasoning.
- Assumptions**: A section for the model's assumptions.
- Findings**: A section containing the analysis results. The text states: "Income level is a strong predictor of whether young adults (ages 24–32) maintain the same savings rate: across all age groups, individuals with lower incomes are significantly more likely to keep their savings unchanged (SAMESAVE = 1), while those with higher incomes are more likely to change (typically increase) their savings. In contrast, there is generally no meaningful difference in income between those who reduced their savings (DISSAVED = 1) and those who did not, except for a modest, statistically significant difference in the oldest group (ages 30–32), where dissavers had slightly lower median income. These patterns are robust across the analyzed age groups, but the results are specific to young adults and may not generalize to older populations. The findings are based on descriptive statistics, visualizations, and Mann-Whitney U tests, which show strong and consistent associations between income and savings stability, but not between income and dis-saving behavior."
- Generated code**: A section showing Python code snippets. The code includes imports for pandas, a function to check for missing values, and a function to generate summary statistics for INCOME and AGE. The code is as follows:

```
1 import pandas as pd
2
3 # Check for missing values in dataset
4 def check_missing(df):
5     missing = df.isnull().sum()
6     print('Missing values per column:')
7     print(missing)
8
9 # Generate summary statistics for INCOME and AGE
10 def summary_statistics(df):
```
- Supporting charts**: A section for visualizations, with a "View all 121 lines" link below the code.

At the bottom left, there is a "Leave Feedback" link.

This feature supports iterative work in the same conversation. Simply type follow-ups in the prompt bar, and it'll add new cells to the report while preserving provenance. For example:

- 1 "Control for baseline weight."
- 2 "Run a non-parametric alternative."
- 3 "Cap outliers at the 99th percentile."
- 4 "Explore correlations and how they change over time."

## AstaLabs

AstaLabs brings our open source Asta research to life through hosted, lightweight prototypes. You get robust early previews without the friction of local installation or environment configuration.

Through AstaLabs, we deploy high-potential pilots to gather the feedback needed to refine their development. This loop ensures resources are directed toward the most promising work—and that the Asta community gains access to cutting-edge tools.

We plan on releasing additional prototypes in the future.

# AstaLabs: AutoDiscovery

AutoDiscovery is an automated scientific discovery engine that ingests large structured datasets, generates testable hypotheses, writes and runs analysis code, and interprets the results. AutoDiscovery uses *Bayesian surprise* to quantify how unexpected a finding is given what it already “knows,” then directs its search toward those surprising results.

Log into AstaLabs\* and try the Example Sessions dataset to see the workflow end-to-end before uploading your own data. When you're ready to run your own analysis:

- 1 Set up your session** by clicking **+ New exploration** to open the wizard. Upload your files (CSV, JSON, Parquet, etc.) and describe your context to seed the system's beliefs. If you're iterating, you can paste learnings from previous runs in the **Intent** field via **Advanced Settings** to refine the search. Finally, set the experiment budget to control how many hypotheses to run.
- 2 Track findings in real time** by hitting **Start Run**. A live table populates as experiments complete. Watch the **Surprisal** score to spot the most surprising findings. Feel free to navigate away—your results will be there when the analysis is complete.

The screenshot displays the AstaLabs AutoDiscovery interface. The top navigation bar includes the AstaLabs logo, 'AutoDiscovery', and user options like 'Experiment Credits: 350', 'Feedback', 'About AutoDiscovery', and 'Sign out'. The main content area is titled 'Global Subnational Poverty Atlas (GSAP)' and shows a 'Top Surprisals' section with a network graph of hypotheses. The graph consists of nodes (circles) connected by lines, with some nodes highlighted in orange and yellow. Below the graph is a table of top surprisals.

ID	Experiment	Surprisal	Belief Before	Belief After	Direct
68	There is a significant Aggregation Bias in the dataset...	0.715	Likely True	Maybe False	Negative
14	Subnational poverty rates exhibit strong positive spa...	0.670	Likely True	Maybe False	Negative
20	The granularity of administrative division affects perc...	0.657	Likely True	Maybe False	Negative
12	The measured standard deviation of subnational pov...	0.637	Likely True	Maybe False	Negative
62	The identification of 'noise'...	0.613	Likely True	Maybe False	Negative

On the right side, a panel for 'Experiment ID: 68' shows a 'Surprisal' plot with two bell curves: a green one for 'Belief After Mean: 0.31' and a pink one for 'Belief Before Mean: 0.90'. Below the plot, the hypothesis is stated as 'There is a significant Aggregation Bias in the dataset: countries reporting fewer subnational administrative units systematically show lower standard deviations of poverty due to the averaging effect of larger regions.' The analysis results show 'Belief before experiment: Likely True (0.903)' and 'Belief after experiment: Maybe False (0.308)'. A regression equation is shown:  $SB\_Poverty \sim \log\_Region\_Count + Mean\_Poverty$ .

- 3 **Audit the details** by clicking any row to open the Inspector Panel, where you can verify the full hypothesis, the statistical analysis, and the actual Python code used to generate the result. It's a completely transparent artifact you can reproduce and build on.
- 4 **Share your findings** by clicking on the **Share** button at the top of the experiments table. This copies a URL to your clipboard, allowing your collaborators to view the complete, interactive analysis and audit the findings themselves.

AutoDiscovery runs are compute-intensive, typically running for several hours. For early access, we're covering the cost via a one-time credit grant—you'll automatically receive 1,000 Hypothesis Credits (1 hypothesis = 1 credit). Credits are available through Feb. 28, 2026.

Think of your first run as a test drive. We suggest starting with a small batch (fewer than 10 hypotheses) just to see what's possible. Once you're familiar with the output, you can scale up to 50–100 hypotheses per session for deeper analysis on larger datasets. Runs are capped at 500 hypotheses.

You'll be prompted to confirm that uploaded data is not confidential. Source datasets are automatically deleted seven days after analysis completes. AutoDiscovery only retains the outputs – hypotheses, plans, code, and results – you need to reproduce and extend your findings.

# AstaLabs: Paper + Figure QA

Paper + Figure QA is an experimental prototype that extends Asta's capabilities from text to visual content—letting you ask natural language questions about the figures, tables, charts, and diagrams found in scientific papers. It's part of Asta's broader effort toward a holistic understanding of research papers, treating every element, visual and textual, as part of the evidence base.

You can ask about trends in a line chart, compare values across a results table, or request an explanation of an architectural diagram. The tool draws on both the visual content and the surrounding paper text – captions, in-text references, methodology sections – to produce grounded, context-aware responses.

To get started with Paper + Figure QA:

- 1 **Go here.**
- 2 **Enter or select** a scientific paper.
- 3 **Ask a question** about any figure, table, or chart in the paper.
- 4 **Review the response**, which draws on both the visual content and the surrounding paper text.

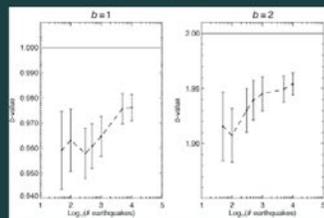
## Paper+Figure QA

[A technical note on the bias in the estimation of the b-value and its uncertainty through the Least Squares technique](#)

Describe all figures

The paper presents seven figures detailing the biases and uncertainties in estimating the Gutenberg-Richter b-value using the Least Squares (LS) technique. Figure 1 demonstrates a negative bias (5% to 2%) in b-value estimates from the cumulative LS method, which lessens with increasing data size. Figure 2 reveals that the cumulative LS method severely underestimates the b-value's uncertainty, by at least an order of magnitude. Figure 3 illustrates how the logarithmic transformation of discrete event counts introduces non-linear 'bends' in the Gutenberg-Richter curve, leading to a negative bias in b-value estimation that decreases with larger datasets. Figure 4 further highlights a strong negative bias in b-value estimates from the binned LS method, which is more pronounced for smaller datasets and generally higher than the bias observed with the cumulative method. Figure 5 is a schematic explaining how magnitude measurement errors cause events to shift between magnitude bins. Figure 6 indicates that the bias in b-value estimation is proportional to the amplitude of magnitude measurement errors. Finally, Figure 7 shows that the addition of measurement errors does not introduce additional bias in the estimation of the b-value's uncertainty.

Figure 1: Fig. 1. Average of  $b_t$  (dashed line) calculated through cumulative LS technique in 1000 synthetic catalogs, as a function of the catalog size, for the cases  $b=1$  (left) and  $b=2$  (right). At each average is attached the 95% confidence interval. The solid line represents the true b-value.



Ask a follow-up question

Submit

## Asta Preview

Asta Preview is an early-access program for researchers who want to try upcoming features and experimental prototypes before public release. Members get exclusive access to new Asta capabilities directly from Ai2's research labs.

The program is open to active researchers, scientists, grad students, and research-adjacent professionals across all domains. The commitment is light: use Asta in your real research, be patient with rough edges, and provide honest feedback through occasional surveys. In return, you'll have the opportunity to influence Asta's product roadmap, join small-group sessions with the team building Asta, and be first in line for future collaborations.

You can opt out at any time and return to the standard public version of Asta.

## Using AstaBench

AstaBench rigorously evaluates AI agents on thousands of scientific tasks—literature understanding, code and execution, data analysis, and end-to-end discovery—with evolving leaderboards and a full build-test-improve environment to help you identify and advance capable agents.

Inside AstaBench, you'll find a comprehensive suite of agents along with production-grade tools that let agents perform scientific research tasks.

We created a utility that enables config-driven suite definition, plus a command-line interface tool for running evaluations and publishing results to a leaderboard. This utility comes bundled with AstaBench, or you can use it to create your own evaluation suite.

## Literature understanding benchmarks

Several AstaBench evaluations test an AI model's literature understanding skills. This includes its ability to locate relevant research papers, retrieve and answer questions from scientific documents, summarize key findings, and more.

<b>PaperFindingBench</b>	PaperFindingBench assesses an agent's ability to locate sets of papers based on a natural language description that may involve both the papers' content and metadata, such as the author or publication year.
<b>LitQA2-FullText-Search</b>	A LitQA2-FullText variant that isolates retrieval: same multiple-choice questions, but agents are scored on ranking papers likely to contain the answer, not on answering.
<b>ScholarQA-CS2</b>	ScholarQA-CS2 evaluates long-form responses to CS literature-review questions, expecting comprehensive, deep-research-style reports. It advances ScholarQA-CS with real-world queries and new metrics for coverage and precision of report text and citations.
<b>LitQA2-FullText</b>	LitQA2 (FutureHouse) tests models on multiple-choice questions that require retrieving a specific paper from the scientific literature and reading its full text—not just the abstract. The original release gave the answering paper's title but no fixed corpus; our version searches the Asta standard index. "-FullText" denotes the subset whose answering papers have open source full text in our index.
<b>ArxivDIGESTables-Clean</b>	ArxivDIGESTables evaluates models on generating literature-review tables—rows as papers, columns as comparison aspects—given related papers and a caption, scoring against tables published in arXiv. "-Clean" is a curated subset removing tables that are trivial or unreconstructable from full text.

## Coding and execution benchmarks

AstaBench evaluates how well models can write, edit, and run code for scientific research tasks. This includes reproducing results from computational studies, modifying existing code, and producing correct outputs in real-world research scenarios.

<b>SUPER-Expert</b>	SUPER-Expert tests models on setting up and executing tasks from low-resource research repositories (centralized databases of research data/materials). The "-Expert" split is SUPER's hardest, requiring reproductions from scratch without hints or intermediate landmarks.
<b>Core-Bench-Hard</b>	Core-Bench-Hard measures computational reproducibility—reproducing study results from provided code and data—via language-only and vision-language tasks across multiple difficulty levels. The "-Hard" split is Core-Bench's toughest: only a README is provided, with no instructions or Dockerfile.
<b>DS-1000</b>	DS-1000 is a well-established code-generation benchmark of Python data-science questions from Stack Overflow, covering diverse, realistic use cases and exercising widely used data-science/ML libraries. We split it into 100 validation and 900 test problems.

## Data analysis benchmarks

AstaBench evaluates a model's ability to analyze scientific datasets and generate meaningful insights. This includes transforming and modeling data to support accurate, data-driven reasoning across scientific domains.

<b>DiscoveryBench</b>	DiscoveryBench is the first comprehensive benchmark to formalize multi-step data-driven discovery—data loading, transformation, statistical analysis, and modeling—and to systematically test how well current LLMs reproduce published findings across domains like social science, biology, and history.
-----------------------	--

## End-to-end discovery benchmarks

AstaBench tests whether agents can complete an entire scientific workflow without human intervention. This includes designing and running experiments, analyzing results, and producing full research outputs.

<b>E2E-Bench</b>	E2E-Bench is the “decathlon” of AI-assisted research. It measures whether a system can run the entire research pipeline, starting with an initial task description, to designing and performing (software) experiments, to analyzing and writing up the results.
<b>E2E-Bench-Hard</b>	E2E-Bench-Hard is a tougher E2E-Bench variant. It generates tasks from research trends and underexplored problems. Tasks are feasibility-checked only—no simplification—testing systems on complex, less-structured research scenarios under the same end-to-end process.

## Trying Asta resources

Asta resources are a set of tools, baseline agents, templates, and APIs that are fully integrated with AstaBench to provide a complete environment for developers to build, test, and refine scientific AI agents.

### Agent tools

AstaBench comes with tools that let agents perform research tasks, with an evaluation mode that enables fair, reproducible benchmarking.

<b>Asta scientific corpus tool</b>	A production-quality toolset for searching and navigating a large-scale scientific literature graph containing 225M+ papers, 80M+ authors, 550M+ paper-authorship edges, and 2.4B+ citation edges. This includes a new first-of-its-kind snippet search endpoint, indexing 12.4M+ full-text publications, totaling 285M+ passages. The literature graph may be accessed via an MCP interface for easy integration with any agent. Use the integration in AstaBench (and no other literature graph tools) for reproducible, fair comparison that limits results to papers dated prior to a benchmark-specified date; submissions that do so are marked "Standard Tools" on the AstaBench leaderboards.
<b>ComputationalNotebook</b>	A Jupyter computational notebook tool—the first such tool to enable reproducible evaluation and broad agent compatibility. Use this tool (and no other execution environments) for fair comparison when completing the SUPER experiment reproduction task (SUPER analyzes notebook usage); such submissions are marked "Standard Tools" on the AstaBench leaderboards.
<b>WritingEditors (Experimental)</b>	Editors for composing literature analysis reports and tables, with support for citing academic sources. Use these stateful tools to allow agents to incrementally compose analyses during the research process, just as humans do.

## Agent suite + other resources

The most comprehensive suite of AI agents—both general and science-optimized—ready to use in AstaBench or elsewhere. It includes code for high-quality general agent implementations as well as scientific agents that achieve high scores on AstaBench.

<b>Asta science agents</b>	<p>Open source scientific research agents, with demonstrated high performance on AstaBench science tasks.</p> <p>Included in the Asta science agents are the open source agents that power Asta—Find papers (PF), Generate a report (SQA), and Data analysis (DV). Each is optimized for particular scientific literature review and analysis tasks, including discovering relevant research, summarizing studies, and generating hypotheses and statistical tests.</p>
<b>Baseline agents</b>	<p>A large collection of agents—both general and science-optimized—including well-known open source baselines in the literature and connectors to closed products.</p>
<b>Open language models</b>	<p>Open language models that have been post-trained for science, which can power open AI agents that excel at science tasks, such as those in our agents suite.</p>

## What's next for Asta?

Asta is intended to be a platform for open scientific exploration. In that spirit, we plan for Asta to continually evolve. We have more agents coming soon, with a vision to combine all the functionality and capabilities into a seamless experience that understands your goals, adapts to your workflows, and helps move science forward.

To accelerate this evolution, we use AstaLabs as our engine for early-stage innovation. By deploying emerging research as hosted, lightweight prototypes, AstaLabs allows the community to provide the field-driven validation necessary to refine these tools before they are fully integrated into the Asta ecosystem.

We hope that this feedback loop, alongside our benchmarks and developer tools, will help the scientific community create a new generation of reliable, verifiable scientific agents—driving measurable improvements to the entire Asta ecosystem along the way.

We'd love to hear about your projects. [Email us](#), or join the conversation on our Discord [here](#).

[allenai.org](https://allenai.org)

