
SCP: Accelerating Discovery with a Global Web of Autonomous Scientific Agents

Yankai Jiang* **Wenjie Lou*** **Lilong Wang*** Zhenyu Tang Shiyang Feng
 Jiaxuan Lu Haoran Sun Yaning Pan Shuang Gu Haoyang Su Feng Liu
 Wangxu Wei Pan Tan Dongzhan Zhou Fenghua Ling Cheng Tan
 Bo Zhang Xiaosong Wang Lei Bai Bowen Zhou

¹Shanghai Artificial Intelligence Laboratory

Abstract

Autonomous AI scientists are beginning to reason, experiment, and collaborate with human researchers by coordinating data, computational tools, AI agents, and laboratory instruments. Yet, most agentic science systems remain difficult to deploy beyond a single lab: they are bespoke, tightly coupled to specific workflows, and lack a shared *protocol layer* that can unify heterogeneous dry-wet resources under a common, secure, and persistent scientific context. We introduce **SCP: the Science Context Protocol**, an open-source standard designed to accelerate discovery by enabling a *global web of autonomous scientific agents*. SCP is built on two foundational pillars: **(1) Unified Resource Integration:** At its core, SCP provides a universal specification for describing and invoking scientific resources—spanning software tools, models, datasets, and physical instruments. This protocol-level standardization enables AI agents and applications to *discover, call, and compose* capabilities seamlessly across disparate platforms and institutional boundaries. **(2) Orchestrated Experiment Lifecycle Management:** SCP complements the protocol with a secure service architecture—comprising a centralized **SCP Hub** and federated **SCP Servers**. This architecture manages the complete experiment lifecycle (registration, planning, execution, monitoring, and archival), enforces fine-grained authentication and authorization, and orchestrates traceable, end-to-end workflows that bridge computational and physical laboratories. Based on SCP, we have constructed a scientific discovery platform that offers researchers and agents a large-scale ecosystem of **1,600+** tool resources. Across diverse use cases, SCP transforms isolated agents and resources into interoperable building blocks. It facilitates secure, large-scale collaboration between heterogeneous AI systems and human researchers while significantly reducing integration overhead and enhancing reproducibility. By standardizing scientific context and tool orchestration at the protocol level, SCP establishes essential infrastructure for scalable, multi-institution, agent-driven science. The open-source SCP specification and reference implementation are available at <https://github.com/InternScience/scp>.

1 Introduction

AI scientists are emerging computational systems that can reason, experiment, and collaborate with human researchers throughout the scientific discovery process [16, 11, 36, 33, 10, 31]. However, building such agentic systems in practice remains difficult. Most current deployments are bespoke, tightly coupled to a single laboratory or platform, and hard-wired to specific tools and workflows [30, 20, 11]. They typically expose ad-hoc interfaces to data repositories, simulation codes, and laboratory instruments, making it challenging to reuse components, reproduce workflows across institutions, or

safely compose heterogeneous capabilities into end-to-end scientific pipelines in a common, secure environment. In particular, no widely adopted agentic system provides a unified protocol layer through which AI scientists, agents, and human-facing applications can interact with models, data, and instruments under a common, persistent scientific context [1].

Recent advances in machine learning, large language models (LLMs) and laboratory automation have given rise to autonomous scientific agent platforms that can carry out research tasks with minimal human intervention [28, 38, 37, 13, 3]. Early demonstrations have largely centered on materials science and chemistry. For example, A-Lab [32], an autonomous materials synthesis laboratory, has demonstrated the power of AI-driven experimental autonomy by integrating computational databases, ML planning, and robotic execution. Over 17 days of continuous operation, it successfully synthesized 41 novel inorganic compounds out of 58 targeted ones, showcasing its high efficiency in materials discovery. Similarly, LLM-based agents like ChemCrow [17] (augmented with 18 expert-designed tools) and Coscientist [5] have autonomously planned and executed complex chemical syntheses, from organocatalysts to optimizing cross-coupling reactions.

The paradigm is rapidly expanding to other critical scientific domains. In life sciences, platforms such as Origene [39] integrate structure prediction and sequence analysis tools to automate the design and engineering of functional proteins, accelerating the development of novel biocatalysts. For earth and environmental science, frameworks like EarthLink [14] connect heterogeneous data sources and knowledge models to enable autonomous reasoning for tasks ranging from climate analysis to ecological impact assessment. At a systemic level, multi-agent architectures like InternAgent [34] coordinate specialized sub-agents to manage end-to-end research tasks—from literature review to experimental validation—demonstrating the potential for orchestrating complex, cross-disciplinary workflows. Concurrently, broader frameworks are being developed to integrate the entire research lifecycle. Agent Laboratory [25] is a system of specialized LLM agents that autonomously generates comprehensive research outputs from a human-provided idea. Kosmos [19] employs a structured world model for parallel multi-agent reasoning over data and literature. Collectively, these efforts underscore a clear trajectory: when equipped with domain-specific tools and structured protocols, AI agents can automate critical research steps and execute end-to-end investigations at a scale and speed beyond the reach of individual human researchers.

The foregoing examples, drawn from diverse fields, illustrate **the emergence of an interconnected “web” of autonomous scientific agents**, each capable of carrying forward portions of the scientific method. However, existing orchestration frameworks [22, 9, 7, 21] and tool registries [11, 8] operate at the level of individual applications and do not provide a protocol-level abstraction for scientific context and lifecycle management. Moreover, true cross-system and cross-institution collaboration among scientific agents remains a challenge. Key obstacles include establishing a consistent shared context across platforms, managing the state of ongoing experiments, and enforcing data access and permission boundaries. Existing standards (*e.g.* the Model Context Protocol) do not fully resolve these issues, making it difficult for heterogeneous agent systems to interoperate seamlessly.

To address this infrastructure gap, we propose the *Science Context Protocol (SCP)*, a unified framework that enables secure, context-aware collaboration among researchers, tools, and autonomous scientific agents. By linking distributed AI agents into a cooperative global network, SCP aims to accelerate discovery through collective intelligence and resource sharing, laying the foundational infrastructure for a collaborative, accelerated, and globally connected paradigm of scientific discovery. SCP builds on model–tool interaction protocols and tool–centric ecosystems, and further extends them in three key directions that are critical for scientific use. First, SCP introduces a centralized **SCP Hub** that maintains global scientific context, enforces per-experiment authentication and authorization policies, and orchestrates calls to a federated set of **SCP Servers** that expose tools, data resources, and physical instruments. Beyond simple request routing, the Hub embeds an intelligent orchestration system that plans and supervises multi-stage workflows spanning computational tools, domain-specific agents, and both dry and wet laboratory operations. It continuously senses the current environment to detect available resources (tools, datasets, and agents), and, given a high-level experimental goal, automatically synthesizes candidate tasks. For each experiment, the Hub ranks the top- k executable plans and surfaces them to the user or AI scientist together with decision rationales such as dependency structure, expected latency, experimental risk, and cost estimates. The orchestration layer is coupled to an internal AI governance module that performs conflict detection and resource forecasting, issuing early warnings when workflows over-constrain instruments, data, or compute budgets. All of these capabilities are exposed through a programmatic API, allowing external AI

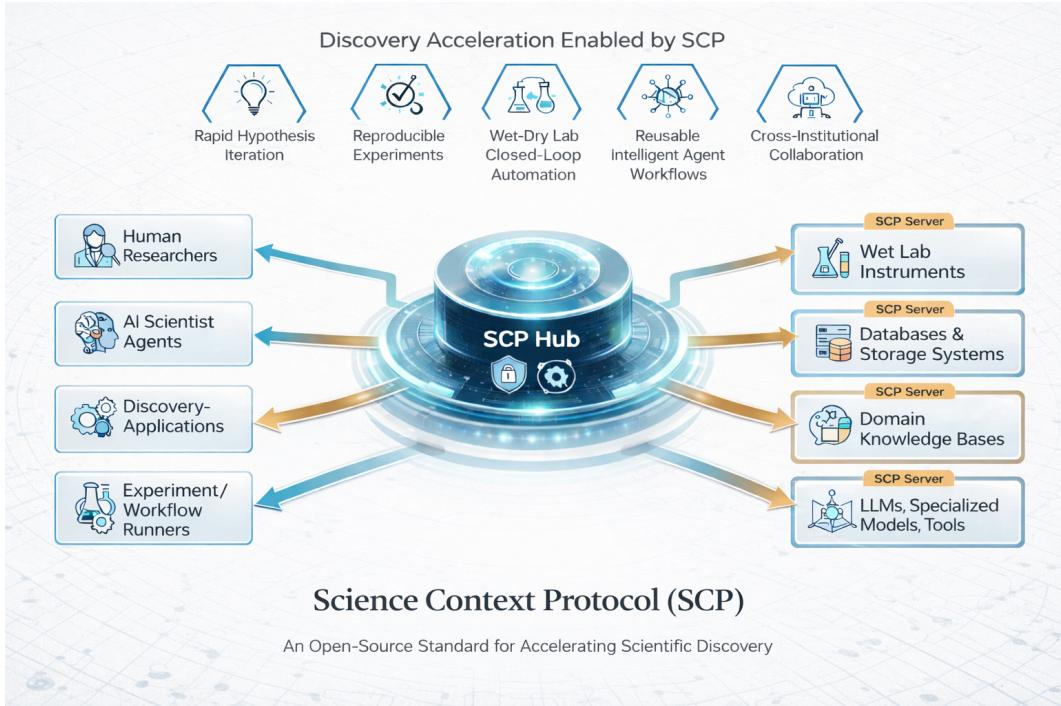


Figure 1: SCP overview. The Science Context Protocol (SCP) is an open-source standard specifically designed to accelerate scientific discovery. By establishing a standardized connectivity framework, it enables efficient interaction between discovery-oriented applications and external research assets—such as laboratory instruments, databases, knowledge repositories, large language models (LLMs), specialized computational models, tools, and APIs. SCP aims to foster a hybrid dry-wet, multi-institution collaborative research paradigm and serve as a novel support platform to enable the collaborative evolution of researchers, research tools, and research subjects in a new era of multi-agent-driven scientific investigation and discovery.

scientists and applications to request end-to-end plans rather than manually scripting low-level tool calls. The Hub thus acts as the “brain” of the system: it parses high-level intents from AI scientists or human users, decomposes them into multi-step experimental plans, and coordinates execution across dry and wet resources while maintaining a persistent audit trail. Second, SCP generalizes the notion of a “tool” to include not only software functions and models but also laboratory devices, workflow engines, and composite multi-agent procedures. Each SCP Server uses a common specification schema to describe the capabilities it offers, their parameters, side effects, and security requirements, allowing clients to reason about and safely compose heterogeneous resources. Third, SCP explicitly models the lifecycle of scientific tasks, from experiment registration and planning through execution, monitoring, and archival, providing first-class abstractions for experiment identifiers, context objects, logs, and provenance. Complementing the orchestration layer, SCP Hubs also drive an automated execution subsystem that instantiates chosen plans, performs secondary validation of preconditions, and manages task state over time (including running, pausing, resuming, and terminating experiments). This subsystem provides asynchronous anomaly notifications and live monitoring views that track tool status, data flows, and resource utilization, and it can trigger predefined fallback strategies when failures or abnormal patterns are detected.

SCP is implemented as an open-source reference platform that supports both local and remote deployment. SCP Hubs and Servers communicate over secure channels and can be integrated with existing identity and access management systems to respect institutional boundaries. This architecture enables AI scientists and scientific applications to orchestrate multi-institution workflows: for example, a single SCP experiment may combine literature retrieval, simulation on a remote high-performance computing cluster, and closed-loop control of robotic laboratory instruments, all expressed within one coherent context and governed by a common policy. Early deployments of SCP in scientific discovery platforms demonstrate that this protocol can break down data and

capability silos, turn curated scientific corpora into “data as a service” for AI models, and support safe collaboration between heterogeneous agents and human researchers.

2 SCP Architecture

2.1 Core Components

SCP adopts a hub-and-spoke architecture in which a centralized *SCP Hub* coordinates a federation of distributed *SCP Servers* (edge servers), user-facing *SCP Clients*, and a set of auxiliary messaging and storage components. Together, these elements form a uniform connectivity fabric that links AI scientists and scientific applications to heterogeneous resources across laboratories and institutions. Conceptually, this design extends model-tool protocols such as MCP [1] from generic tool invocation to the full lifecycle of scientific experiments, including experiment registration, planning, execution, and provenance tracking.

On top of the basic MCP-style client-server interaction model [1], SCP introduces four major extensions tailored to scientific workflows. (i) **Richer experiment metadata.** SCP defines a first-class experiment context that records a persistent experiment identifier, experiment type (dry, wet, or hybrid), high-level goals, data storage URIs, and configuration parameters. This structured context supports end-to-end traceability, versioning of experimental runs, and integration with institutional data-governance policies. (ii) **Centralized SCP Hub.** Unlike purely peer-to-peer protocols, SCP designates the Hub as a global registry for all experiment-facing services (data sources, computational models, laboratory instruments, and composite agents). The Hub manages service discovery, lifecycle management, experiment memory, and OAuth2.1-based authentication and authorization for experiments and users. (iii) **Intelligent workflow orchestration.** Beyond single tool calls, SCP adds an *experiment-flow API* layered over conventional agent APIs [11]. This API allows the Hub to synthesize and recommend candidate multi-step workflows—including resource allocation and follow-up actions—from a high-level experimental goal, enabling AI-driven generation and execution of complete experimental protocols. (iv) **Wet-lab device integration.** SCP extends the notion of a tool to cover real laboratory equipment by standardizing device drivers and capability descriptions. This enables dry (computational) and wet (physical) experiments to be composed into unified workflows under one protocol, making robotic platforms, analytical instruments, and in-silico models equally addressable through SCP.

We now describe the responsibilities of the core components in this architecture.

2.1.1 SCP Hub (central orchestrator and protocol authority)

The SCP Hub is the “brain” of the system. It maintains the global registry of tools, datasets, agents, and instruments; handles service discovery; performs task dispatch and tracking; and implements protocol-level security and governance. Given a high-level request from an AI scientist or user-facing application, the Hub interprets the intent under the current experiment context, decomposes it into a sequence of concrete experimental tasks, and coordinates their execution across multiple Servers.

Intelligent orchestration of domain knowledge, tools, agents, and experiments. As outlined in the Introduction, a key innovation of SCP is the intelligent orchestration layer embedded in the Hub. When a user submits a complex scientific task, the Hub first uses AI-driven intent-analysis models to translate natural-language instructions into a set of candidate task graphs involving tools, datasets, agents, and both dry and wet experimental operations. It then evaluates these candidates against the current environment: available tools and instruments, dataset readiness, user permissions, and resource budgets. For each experiment, the Hub ranks the top- k executable plans and surfaces them—together with dependency structure and coarse estimates of latency, cost, and experimental risk—to the human researcher or higher-level AI scientist for selection. Alternatively, users who prefer direct control can bypass the high-level intent analysis and manually compose their workflows using the underlying APIs or graphical interfaces. The intelligent orchestration mechanism is exposed via the experiment-flow API, which is layered on top of traditional agent APIs [11, 1] and enables AI scientists to request end-to-end workflows rather than manually chaining individual tool calls.

Fine-grained repeatable protocol-level workflow specification. Once a candidate plan has been selected, the Hub compiles it into a fine-grained, protocol-level workflow specification, represented

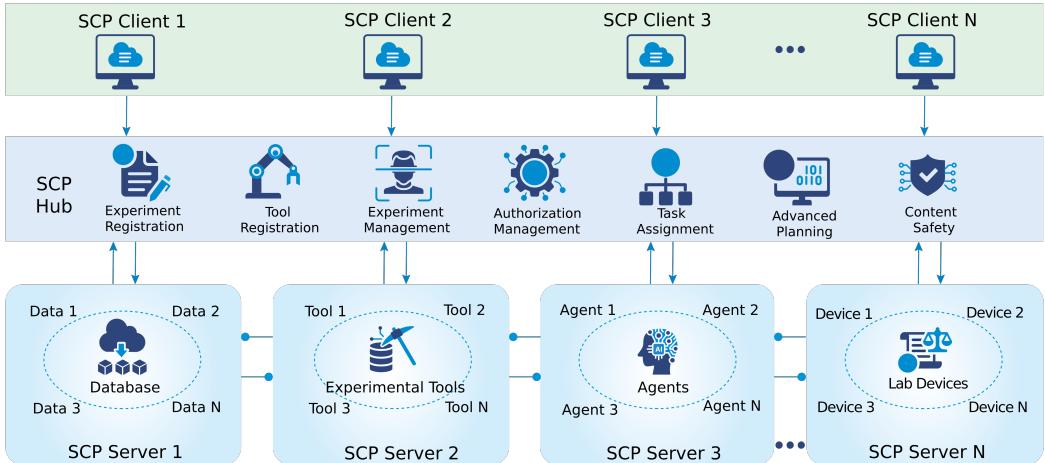


Figure 2: SCP architecture overview. The SCP Hub coordinates interactions between user-facing clients (top) and various SCP *edge servers* (bottom) that interface with laboratory instruments, databases, AI models, and other tools. Researchers interact with the system through an SCP client application, which communicates with the Hub. The Hub manages experiment context, planning, and task scheduling across the network of tools. Each SCP edge server registers available devices or services with the Hub, executes tasks on those resources, and streams results back to the Hub in real time. This design enables a seamless flow of information and commands between human researchers, AI-driven agents, and physical lab equipment under a unified protocol.

as a structured JSON task graph. Each node in this graph corresponds to a single operation—such as calling a data-cleaning pipeline, running a simulation, executing a trained model, or actuating a robotic liquid handler—and is annotated with the tool identifier, input parameters, expected outputs, and explicit dependencies on previous steps. This JSON representation standardizes the experimental protocol at the level of the SCP architecture: it serves as the contract between AI scientists, Servers, and instruments, and as the canonical record of “what was planned” for an experiment. The Hub versions and stores these workflow specifications as part of the experiment memory, enabling replay, comparison across runs, and downstream auditing of experimental decisions.

Automated execution, feedback, and interaction. Given a validated workflow specification, the Hub enters an automated execution phase. It dispatches each step in dependency order to the appropriate SCP Server, supplying the required parameters and experiment-context metadata. Servers execute the requested operations and stream back status updates and results. The Hub tracks task states (running, paused, completed, failed) over time and logs server response latency for each step, providing live monitoring views of tool status, data flows, resource utilization, and performance metrics.

Critically, the Hub implements a built-in validation and feedback loop. For data-centric steps, it can verify that returned values conform to expected schemas or basic sanity checks; for physical operations, it inspects device acknowledgements and sensor readings to confirm that actions were successfully applied. If a step fails validation or an anomaly is detected (for example, repeated tool errors, inconsistent outputs, or unexpected resource consumption), the Hub triggers an exception-handling policy: issuing warnings, pausing or aborting downstream steps, and optionally rolling back prior actions when safe to do so. These rollback and recovery policies are particularly important in multi-step, dry-wet workflows, where early errors can otherwise propagate and invalidate subsequent measurements. Through this dispatch–monitor–validate–rollback loop, the Hub acts as a robust execution coordinator that keeps large, multi-agent scientific workflows on track.

2.1.2 SCP Servers (edge nodes)

An SCP Server connects to and manages local scientific resources within a given environment (*e.g.*, a laboratory, data center, or cloud account). Typical resources include experimental instruments, domain-specific models, databases, data-processing pipelines, and higher-level agents. The Server is

responsible for registering these resources as SCP tools, exposing their capabilities and parameter schemas through a common specification format, and enforcing local access-control policies.

At runtime, each SCP Server receives task directives from the Hub and invokes the appropriate tools or devices to execute specific experimental steps. It streams intermediate states and final results back as structured messages, enabling the Hub to maintain a global view of progress. On startup, a Server automatically registers itself and its exported tools with one or more Hubs, making its local capabilities discoverable to AI scientists and clients across the SCP ecosystem. To facilitate integration, the SCP Hub provides a set of registration interfaces through which users can submit their own Servers—for example, via the Intern-Discovery platform [26]—to be incorporated into the unified management of the platform’s SCP Hub. Servers also perform continuous health monitoring, periodically reporting device status, model readiness, and resource utilization to the Hub. These reports inform global scheduling and failover decisions made at the Hub level.

2.1.3 SCP Clients

SCP Clients provide the interface layer for human researchers, AI scientists, and higher-level applications. A Client authenticates users against the Hub, retrieves the list of tools, agents, datasets, and instruments that are visible under the user’s permissions and project context, and offers interaction surfaces for experiment design and execution. In simple cases, a Client can present a catalog-style interface for discovery and direct invocation of individual tools. In more advanced scenarios, it exposes graphical or programmatic builders for composing high-level experimental goals, which are then handed off to the Hub’s orchestration layer.

Every invocation issued by a Client carries the user’s credentials and the relevant experiment identifier, allowing the Hub to enforce fine-grained access-control policies and attribute actions to specific users or AI agents. This design ensures that, even as SCP connects a growing ecosystem of Servers across institutional boundaries, experiment-level security and provenance remain centralized and auditable at the Hub.

2.2 High-Level Usage Patterns and Workflows

In this section, we outline how scientists, developers, and infrastructure architects could interact with SCP to carry out and support complex experiments. At a high level, SCP provides a unifying interface for researchers to design and execute experiments, for developers to integrate new tools and instruments, and for architects to deploy a scalable, secure infrastructure. Figure 2 illustrates the overall SCP platform architecture, showing how users, intelligent agents, and laboratory devices are connected via the centralized SCP Hub. Based on the above architecture, the user-facing process of conducting an experiment with SCP proceeds through several stages:

(1) Experiment Query and Design (Researcher Interaction): A scientist begins by formulating an experimental request or query using an SCP client interface. This could take the form of a structured query (selecting from available protocols) or a natural-language request describing the experiment’s goal. The SCP client helps the researcher by showing the inventory of available tools and data resources (filtered by the user’s permissions) and providing a user-friendly interface for composing an experiment workflow. For example, a biologist might ask the system to “find the optimal protocol to synthesize compound X and measure its activity,” or manually select a series of laboratory actions to perform. The request is sent to the SCP Hub along with relevant metadata (experiment name, type, priority, etc.), establishing a new experiment context within the platform.

(2) Protocol Generation and Planning (AI Agent & Hub Orchestration): Upon receiving the query, the SCP Hub interprets the researcher’s intent and generates a detailed experimental protocol. This process can involve intelligent planning agents (e.g. an LLM-based planner) that analyze the request and propose a sequence of steps to achieve the goal. The SCP Hub’s intent recognition module parses the input to identify high-level tasks and constraints. It then evaluates available methods and resources registered in the system, possibly consulting domain-specific models or knowledge bases for recommendations. The outcome is one or more candidate protocols expressed in a structured JSON-based format that enumerates the required steps, tools, and parameters for the experiment. Notably, the Hub may suggest multiple ways to proceed — for instance, different instruments or methods to accomplish a step — and present the top three execution plans to the user for review.

Each proposed plan includes metadata (estimated duration, resource usage, predicted outcomes) to help the researcher make an informed selection.

(3) Selection and Orchestration of Domain Knowledge and Tools (Researcher & Hub): The researcher reviews the suggested protocols and selects a plan (or manually edits it) to execute. Once a plan is confirmed, the SCP Hub orchestrates its execution. It decomposes the high-level protocol into a series of discrete actions and assigns these tasks to the appropriate SCP edge servers that manage the required tools or lab devices. For example, if the protocol calls for a temperature-controlled reaction followed by spectral analysis, the Hub will route the heating step to a connected thermostatic reactor device and the analysis step to a spectroscopy tool, via their respective SCP servers. All task assignments are done through the standard SCP interface, encapsulating each command and its parameters as defined by the protocol specification. The Hub coordinates scheduling, taking into account the availability and status of each device, and initiates each step in turn or in parallel as needed. Crucially, the researcher can monitor this orchestration in real time: the SCP client provides live updates or a dashboard view of which tasks are running, completed, or pending. The unified protocol ensures that whether a step is executed by a robotic lab instrument, an AI model, or a data processing pipeline, it is invoked in a consistent manner and tracked under the same experiment context.

(4) Execution on Laboratory Devices and Tools: Each SCP edge server receives tasks from the Hub and carries out the specified action on the local resource under its control. The SCP server software at the lab side translates the high-level protocol command into the device-specific operation (for instance, moving a robotic arm, running a simulation, calling an external API, or querying a database). Developers have extended the SCP servers with a library of device drivers and tool adapters, so that most actions simply call pre-defined routines exposed by the device's API. During execution, the edge server streams the action's progress and any intermediate data back to the Hub. This could include logging information, sensor readings, or partial results (e.g., an image just captured by a microscope). The Hub aggregates these updates and makes them available to the researcher in real time, ensuring transparency and allowing human oversight if needed.

(5) Feedback, Analysis, and Iteration: As the experiment progresses, the SCP Hub closes the loop by analyzing results and feeding them back into the context of the running experiment. If an intelligent agent is steering the experiment, it can use the incoming data to decide on-the-fly adjustments. For example, an AI planning agent might detect that a reaction's yield is below the expected threshold and suggest modifying the temperature or trying an alternative catalyst in a subsequent run. The researcher can either approve these adaptive changes or intervene with new instructions if necessary. Once all steps (including any conditional or looped steps) are completed, the SCP client presents the final results to the scientist. The outcomes might include processed data, graphs, lab notes, or even a draft report generated by an AI assistant. Because SCP assigns uniform identifiers to every data object and records every action in the protocol, the entire experiment is traceable and reproducible. Researchers (or other team members) can review the JSON protocol log to understand exactly what was done and to repeat the experiment in the future, satisfying a key requirement for scientific rigor.

The above workflow is made possible by developers who extend the SCP platform with new tools and capabilities. From a developer's perspective, using SCP involves writing adapters or drivers that wrap instruments and software tools behind the SCP protocol. For example, a developer can implement a new device action by subclassing the SCP server's device interface or using a decorator to register a function as a tool action. Once the code for a new tool is written, the developer deploys it on an SCP edge server. The edge server automatically registers its tools and services with the central Hub, advertising what actions it can perform. This modular design means developers can continually add support for new hardware or analysis routines without modifying the core of SCP. Scientists immediately gain access to those new capabilities via the unified interface. In practice, developers also use SCP's client APIs to script complex tasks or to build higher-level applications. They might write Python scripts that connect to the SCP Hub to programmatically submit experiment protocols or retrieve results, which is useful for integrating SCP into custom pipelines or GUIs. The SCP project's open-source repository provides documentation and examples for developers, including how to define custom actions and ensure they conform to the protocol's JSON schema.

Ultimately, the coordination provided by SCP – from low-level device control up to high-level experiment planning – allows researchers to focus on scientific questions while the software handles the complex choreography between AI agents and laboratory infrastructure. This synergy between

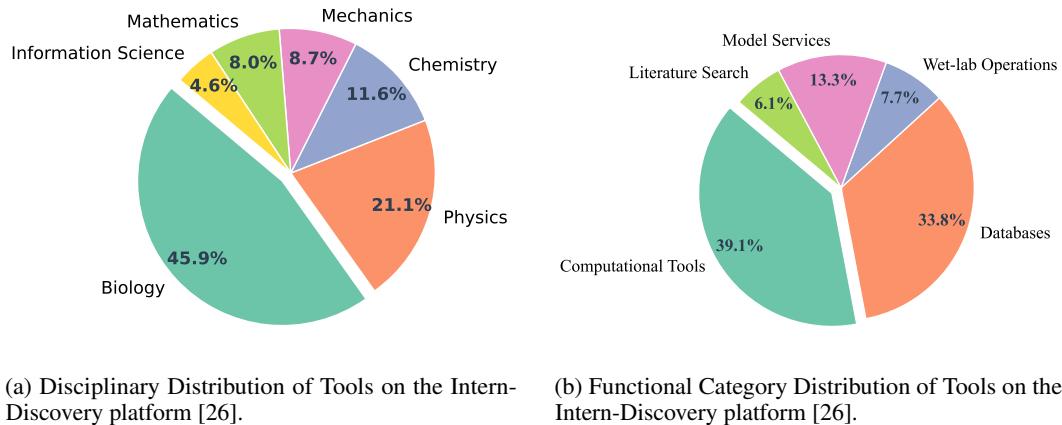
scientists, AI agents, and automated lab systems illustrates a new paradigm of conducting research, one that is markedly more efficient, collaborative, and adaptive than traditional methods [23, 27].

3 Tool Assets: The SCP-Based Intern-Discovery Platform

Building on SCP, we have constructed a large-scale, highly diverse tool assets on Intern-Discovery platform [26]. This platform integrates over 1,600 interoperable tools spanning databases, computational utilities, and specialized models across multiple scientific domains, forming one of the most comprehensive tool ecosystems reported for scientific agent systems. The complete scp server list and tool list is provided in Table 1 and Table 2 respectively in the Appendix.

From a functional perspective, the platform covers the full spectrum of scientific workflows, including biological and chemical database retrieval (e.g., UniProt, InterPro, PDB, NCBI), structure and sequence processing, molecular modeling, data preprocessing, numerical computation, and machine learning inference. From a disciplinary standpoint, these tools collectively support a broad range of fields such as molecular biology, protein engineering, bioinformatics, chemistry, materials science, and medical research.

Unlike prior agentic or tool-augmented frameworks that typically rely on a limited and task-specific toolset, the SCP protocol provides a unified, standardized, and extensible interface that enables large language model agents to seamlessly access and orchestrate this massive collection of heterogeneous tools through a common JSON-based schema and a centralized hub. This large-scale, protocol-enabled integration not only significantly expands the action space and problem-solving capacity of scientific agents but also facilitates cross-domain reasoning and complex multi-step workflows that are difficult to achieve with conventional systems. As a result, the SCP-based platform establishes a robust and general-purpose foundation for scalable, tool-driven scientific intelligence, demonstrating clear advantages in flexibility, coverage, and long-term extensibility.



(a) Disciplinary Distribution of Tools on the Intern-Discovery platform [26]. (b) Functional Category Distribution of Tools on the Intern-Discovery platform [26].

Figure 3: Distribution of disciplines and functions of tools available on the Intern-Discovery platform [26]. The tool collection is continuously updated and expanded.

In Figure 3a and Figure 3b, we present the disciplinary distribution and functional categorization of the tools available on the platform. The tools encompass a broad spectrum of disciplinary categories.

- **Biology and Related Technologies (45.9%):** This category covers the complete biomedical research pipeline, ranging from genomics and proteomics to drug discovery and disease research.
- **Physics (21.1%):** This includes major branches of physics, encompassing optics and electromagnetism, thermofluids, electromagnetics, and fundamentals across various sub-disciplines.
- **Chemistry (11.6%):** This comprises chemical molecular databases, chemical and reaction computations, and multiple chemistry branches such as computational chemistry and physical chemistry.

- Mechanics and Materials Science (8.7%).
- Mathematics (8.0%).
- Information Science and Computing Technology (4.6%).

This distribution closely reflects the trend in modern scientific research, particularly the deep integration of experimental disciplines (*e.g.*, biology, materials science) with computational methodologies.

4 Case Study

4.1 Case Study 1: Automated Experimental Protocol Design and Execution

A researcher begins by submitting a high-level experimental objective (for example, “design and run a PCR protocol to verify a gene knockout”) to the SCP hub. The SCP hub orchestrates this request by distributing tasks across its network of service nodes. A planning agent translates the objective into the standardized SCP JSON format, and the Thoth Server is invoked as a specialized SCP service node to generate the detailed laboratory protocol [29]. Thoth processes this input and returns a structured JSON protocol that describes each step of the experiment, including reagents, quantities, and timing. Next, the SCP hub proceeds with execution. It forwards the JSON protocol to the Thoth-OP Server, another specialized SCP service node dedicated to execution planning. The Thoth-OP Server decomposes the structured protocol into atomic operations and generates device-specific command sequences. Each command (for example, a pipetting instruction) is output in a standardized format. The SCP system then dispatches these commands to the appropriate laboratory instruments via their APIs. Throughout this process, the user sees the entire workflow seamlessly managed by SCP: the initial request is transformed into a protocol and executed automatically by the Thoth nodes.

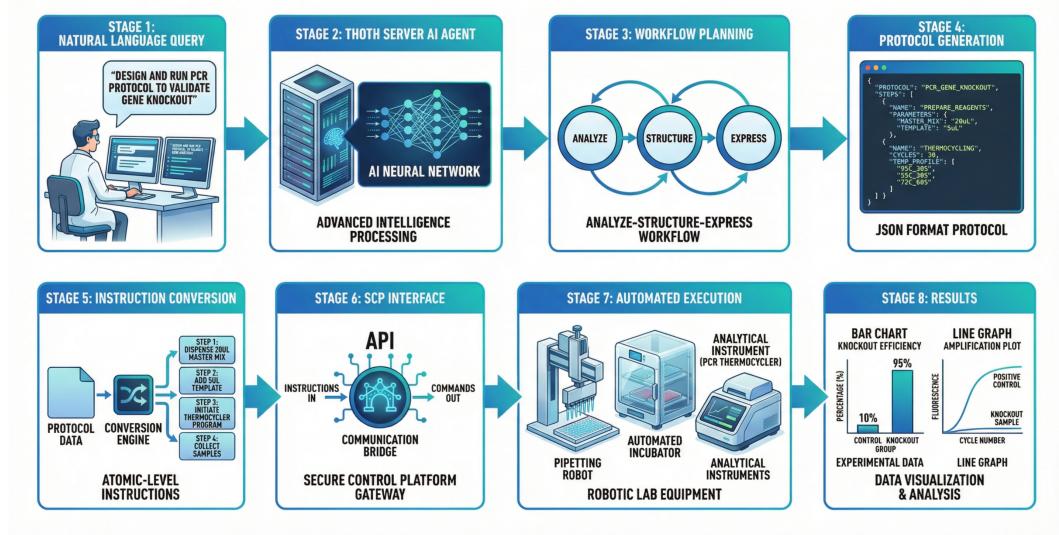


Figure 4: Case Study 1: Automated Experimental Protocol Design and Execution.

4.2 Case Study 2: Automated Reproduction of an Existing Protocol from PDF

In this scenario, SCP is used to automatically reproduce an experiment from an existing protocol document. A researcher does not need to rewrite or adapt the procedure manually: they simply upload a protocol PDF (for example, a method section from a paper or a lab SOP exported from an ELN) to the SCP system. The SCP hub classifies the task as protocol ingestion and routes the document to the Thoth Server, which functions as an SCP-compatible protocol understanding node.

Thoth parses the free-form PDF, extracting the experimental objective, materials, and each step’s actions, parameters, and dependencies. Even when the document uses narrative language or heterogeneous formatting, Thoth converts it into a standardized JSON protocol object that precisely

enumerates the workflow: which reagents are used, in what volumes, at which temperatures, for how long, and in what order. From the user’s perspective, “any reasonable wet-lab protocol PDF” is thus transformed into a machine-readable, executable representation without manual transcription.

Once the structured protocol has been generated and optionally reviewed, the SCP hub forwards it to the Thoth-OP execution node for operation planning. Thoth-OP decomposes each step into atomic device-level instructions and validates all parameters against the capabilities and safety limits of the available instruments. It then emits a sequence of standardized commands that the SCP infrastructure dispatches to the appropriate lab devices. As a result, the experiment described in the original PDF is automatically reproduced on the automation platform: pipetting, mixing, incubation, and measurement steps are executed end-to-end under SCP’s control. The researcher experiences the pipeline as: upload a protocol file, confirm the interpreted plan, and watch the system intelligently and safely replay the experiment on real hardware, demonstrating SCP’s ability to turn arbitrary protocol documents into fully automated, reproducible workflows.

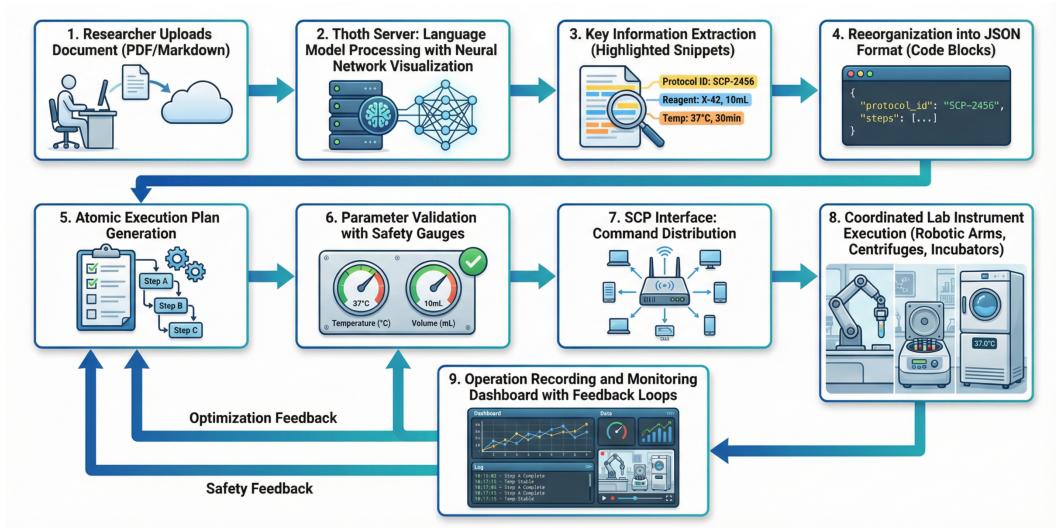


Figure 5: Case Study 2: Automated Reproduction of an Existing Protocol from PDF.

4.3 Case Study 3: AI-Driven Molecular Screening and Docking via SCP

This case study demonstrates how the SCP toolchain can support an end-to-end small-molecule screening and protein-ligand docking campaign in an automated manner. The workflow combines cheminformatics, ADMET prediction, structural biology, and molecular docking, all orchestrated through a unified SCP pipeline.

Step 1: Molecule property evaluation. The process starts from a library of 50 small molecules encoded in SMILES format. Using SCP tools such as `calculate_mol_drug_chemistry`, the system computes the QED (Quantitative Estimate of Drug-likeness) score for each molecule, providing a numerical measure of drug-likeness. In parallel, the `pred_molecule_admet` tool predicts the LD₅₀ toxicity metric for each compound. These two quantities jointly characterize how promising and safe each candidate is from a medicinal chemistry perspective.

Step 2: Initial filtering of candidate molecules. Based on pre-defined thresholds, the workflow filters molecules with QED ≥ 0.6 and LD₅₀ ≥ 3.0 . This filtering step eliminates compounds with poor drug-likeness or high predicted toxicity and yields a refined set of six candidate molecules. From a user’s perspective, this entire stage is expressed as a small number of SCP tool invocations over the input SMILES list, and the platform automatically returns a structured table of qualified compounds.

Step 3: Protein target preparation. Next, the workflow prepares the receptor protein for docking. Using PDB ID 6vkv as the target, the SCP toolchain performs a series of automated operations: (i) download the PDB structure; (ii) extract the main chain (default chain A); (iii) repair missing or

inconsistent regions using PDBFixer; and (iv) identify putative binding pockets with `fPocket`. The pocket with the highest score is selected as the docking site, and its coordinates are recorded for subsequent stages. All of these operations are invoked as SCP tools, so the user does not manually handle PDB manipulation or pocket detection; instead, they see a clean, standardized receptor object with an associated binding region.

Step 4: Format conversion for docking. To interface with an AutoDock Vina-style docking engine, both ligands and receptor must be converted into PDBQT format. The six filtered SMILES molecules are transformed into 3D structures and exported as ligand PDBQT files, while the repaired protein structure is likewise converted into a receptor PDBQT file. This conversion is triggered by SCP tools that hide the underlying cheminformatics steps, ensuring that all input files for docking are generated in a consistent, reproducible manner.

Step 5: Docking and final hit selection. The final stage invokes the `quick_molecule_docking` tool at the previously identified pocket center. Each of the six candidate ligands is docked into the binding site, and the workflow records the predicted binding affinities. Molecules with docking affinity ≤ -7.0 kcal/mol are retained as high-potential hits. In this example, two compounds survive this final filter:

- O=C(c1ccc(F)cc1F)N1CCN2C(=O)c3ccccc3C12c1ccc(Cl)cc1
- O=C(c1cccc(F)c1)N1CCN2C(=O)c3ccccc3C12c1ccc(Cl)cc1

From the user's perspective, the SCP pipeline transforms an initial list of 50 SMILES strings into a pair of prioritized hit molecules bound to a structurally prepared protein target. Each stage of the workflow—property prediction, filtering, protein preparation, format conversion, and docking—is encapsulated as composable SCP tools, illustrating how the platform supports an AI-driven early drug discovery paradigm with minimal manual intervention.

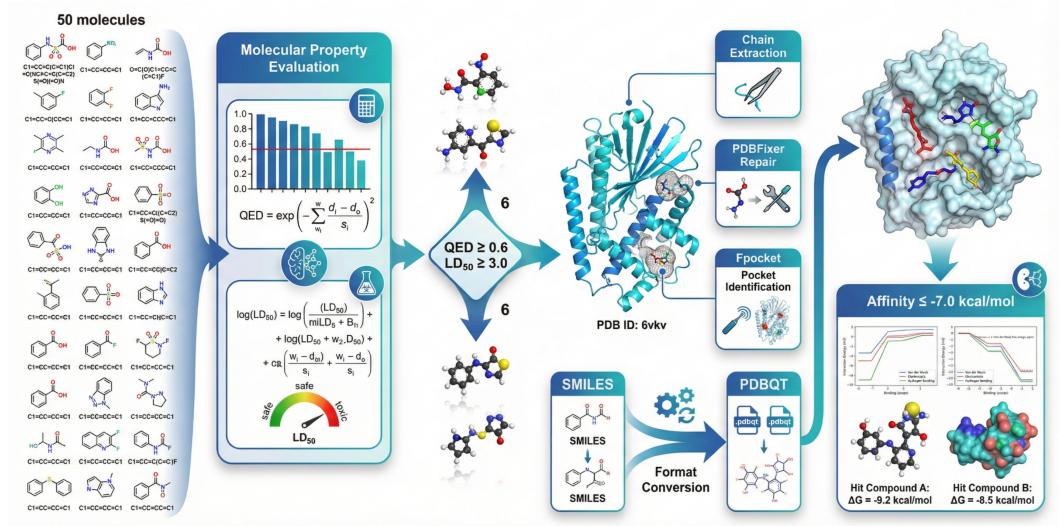


Figure 6: Case Study 3: AI-Driven Molecular Screening and Docking via SCP.

4.4 Case Study 4: AI-Assisted Fluorescent Protein Engineering with Dry–Wet Integration via SCP

This case study illustrates how SCP supports a tightly coupled dry–wet workflow for engineering fluorescent proteins.

On the “dry” side, the scientist begins by specifying an optimization goal in the SCP client, such as increasing brightness or photostability of a given fluorescent protein scaffold under specific experimental conditions. The request is sent to the SCP Hub, which orchestrates a set of computational

servers (e.g., sequence design, structural modeling, and property prediction tools) through standardized JSON task plans. These tools explore sequence space around the wild-type protein, perform in silico mutational scanning, and predict key properties such as folding stability, spectral shift, and expression level. The SCP Hub aggregates these predictions into a ranked list of candidate variants and encodes each design, together with its predicted properties and intended assay conditions, into a unified SCP experiment plan that can be directly reused by downstream wet-lab components.

On the “wet” side, the same SCP plan is automatically translated into executable experimental workflows. For each selected fluorescent protein variant, the SCP Hub dispatches structured JSON sub-plans to wet-lab servers, which generate detailed protocols for plasmid construction, transformation, cell culture, induction, and fluorescence readout. These protocols are further compiled into atomic operation sequences (such as pipetting, incubation, centrifugation, and plate reading) that can be executed on robotic platforms. During execution, the SCP Hub monitors instrument status and assay progress, streaming quantitative fluorescence measurements and quality-control metrics back into the shared SCP context. This creates a closed loop in which in silico design results, wet-lab measurements, and protocol variants are all represented in a single, standardized SCP timeline. As a result, the fluorescent protein engineering workflow depicted in the figure is no longer a loose combination of simulation and experiment, but a unified dry–wet pipeline: SCP provides the orchestration layer that connects AI-based design, automated experimentation, and iterative optimization into a single, reproducible scientific process.

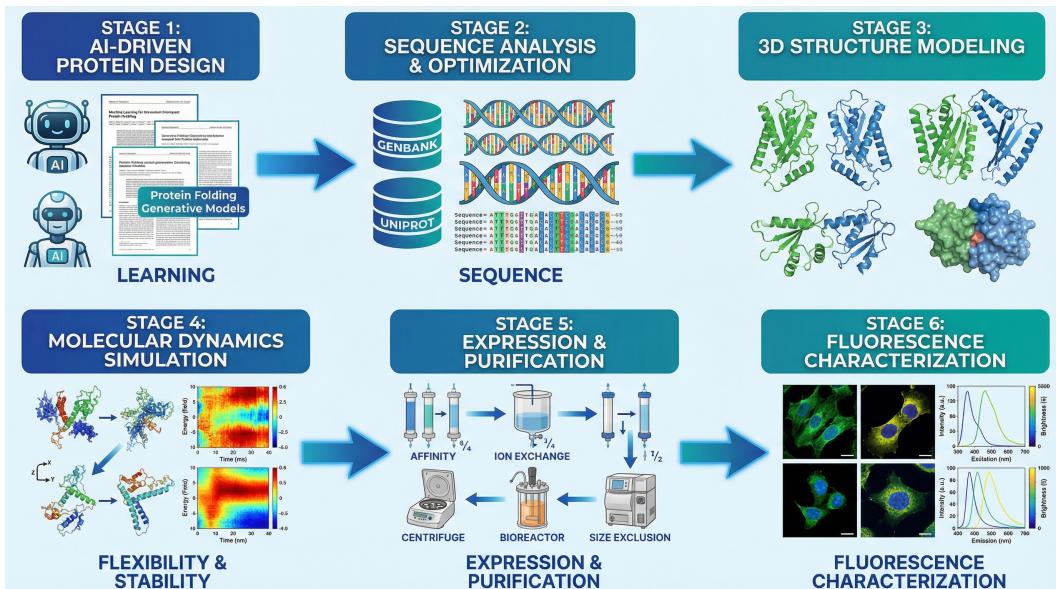


Figure 7: Case Study 4: AI-Assisted Fluorescent Protein Engineering with Dry–Wet Integration via SCP

5 Discussion: Comparative Analysis of SCP and MCP

SCP and MCP [2] represent two distinct approaches to orchestrating AI-driven research workflows. MCP provides a general-purpose standard for connecting AI models with tools and data sources. This generic design has been widely adopted for integrating LLMs with software APIs, databases, and other resources in a uniform way. However, when applying MCP to complex scientific domains, several limitations become evident. In contrast, SCP is a specialized framework built to address these shortcomings by introducing domain-specific structure and coordination mechanisms on top of the MCP paradigm [21]. In this section, we provide a rigorous comparison of MCP and SCP, focusing on MCP’s limitations in protocol standardization, high-throughput experimentation, multi-agent orchestration, and integration of wet-lab equipment, and how SCP’s design offers solutions in each of these areas.

Protocol Standardization and Contextual Structure. MCP was designed to standardize basic interactions (e.g., file I/O, function calls) between AI agents and tools across disparate systems. This works well for straightforward tasks (such as querying a database or calling a cloud API) [24, 18, 15, 35, 6, 4], but MCP by itself does not impose a structured format for complete scientific protocols or experimental plans. In scientific workflows, researchers require a high-level representation of the entire experiment (including objectives, parameters, and expected outcomes) that can be understood and shared among multiple agents and tools. MCP lacks this notion of a scientific context. It treats interactions as isolated messages rather than parts of a cohesive experiment plan. As a result, protocol standardization in a scientific sense (i.e., a consistent way to describe experiments end-to-end) is weak under MCP: different labs or projects might develop their own conventions on top of MCP, leading to fragmentation.

SCP directly tackles this issue by providing a standardized research workflow representation built into the protocol. Borrowing inspiration from MCP’s general interface, SCP defines a structured, JSON-based schema for planning and describing experiments. Each experiment in SCP carries rich metadata (e.g., unique experiment IDs, researcher identifiers, experiment type as dry/wet/hybrid, objectives and so on) and uses a formal JSON schema to outline the procedure steps and resources involved. This structured protocol planning ensures that every step of an experiment is explicit and machine-interpretable. For example, an experiment plan in SCP might specify the sequence of actions (data preprocessing, model training, hypothesis generation, wet-lab validation steps, etc.) in a nested JSON format, which the SCP system can parse and manage. The presence of a well-defined schema means that all agents and tools in the SCP ecosystem interpret the protocol in the same way, greatly enhancing standardization over what MCP alone can achieve. In practice, this means a laboratory adopting SCP can have all their instruments and AI agents follow the same protocol blueprint for any given experiment, reducing ambiguity and setup overhead. In summary, while MCP provides the low-level message format for tool access, SCP adds a higher-level grammar for scientific experimentation, bringing protocol uniformity to complex workflows.

High-Throughput Experimentation Support. Scientific research often demands running many experiments or iterative trials in order to explore large parameter spaces (for instance, screening dozens of candidate compounds or running thousands of simulations in a materials study). In such high-throughput scenarios, MCP’s lack of built-in experiment management becomes a bottleneck. Under MCP, each tool invocation is essentially stateless and context-agnostic, so coordinating a batch of 100 experiments requires external scheduling logic and careful tracking by the user or a separate program. There is no native concept of an “experiment queue” or a systematic way to chain results from one run into the next; the protocol does not remember past actions unless the agent itself implements memory. This limitation means researchers must implement additional layers on top of MCP to handle batching, concurrency, and result aggregation. The absence of a unified context across runs can lead to repeated setup overhead and potential errors when scaling up to high-throughput workflows. SCP offers enhancements geared towards high-throughput and iterative experimentation. Because SCP treats the entire workflow as a first-class object (with an experiment ID and a persistent context), multiple runs or trial variants can be managed under a common protocol umbrella. Concretely, SCP can automatically log the outcome of each high-throughput experiment instance (e.g., each combination of parameters in a grid search or each candidate molecule in a screening assay) with a uniform format, making it easy to compare and aggregate results. The structured metadata (including priority levels and unique identifiers for each run) allows the system to schedule experiments efficiently and in parallel where possible. Researchers can request an entire batch of experiments in one protocol submission (thanks to the JSON-based plan encoding all trials), and the SCP framework will coordinate their execution and collect all results. SCP improves support for high-throughput and large-scale experimentation by introducing context-aware batching and automated workflow execution, reducing the manual burden present in an MCP-only approach.

Multi-Agent Orchestration and Coordination. Modern scientific projects increasingly involve multi-agent systems—multiple AI agents (and humans) collaborating, each with specialized roles (e.g., a data-analysis agent, a hypothesis-generating agent, a robotic lab assistant, etc.). MCP, by design, is message-centric but not inherently multi-agent: it defines how an AI agent (client) communicates with external tools (servers), but it does not specify how multiple autonomous agents should coordinate with each other in a complex task. In fact, auxiliary protocols like Google’s A2A (Agent-to-Agent) were needed to handle direct inter-agent communication in the absence of

a higher-level framework [12]. Without an overarching orchestration mechanism, an MCP-based multi-agent system can devolve into a collection of point-to-point message exchanges with no global oversight. This makes it challenging to implement structured teamwork or division of labor among agents using MCP alone. In essence, MCP offers the messaging pipes, but the orchestration logic for multi-agent workflows in scientific settings is absent. This was recognized as a shortcoming in emerging AI Scientist ecosystems: existing protocols lacked a unified approach to coordinate diverse agents and tools within a single, reasoning-aware workflow. This leads to situations where human scientists must act as the “protocol” between AI agents, manually integrating their outputs.

The SCP framework incorporates multi-agent coordination as a core feature rather than an afterthought. It introduces a centralized SCP Hub that serves as an orchestrator for all participating agents and resources in an experiment. Through the Hub, agents do not just send blind messages; instead, each agent’s actions are contextualized within a shared experiment state managed by the hub. This extension allows the system to decompose high-level goals into sub-tasks, assign those sub-tasks to the appropriate specialized agents, and sequence their execution. The hub also tracks dependencies and completion status of tasks, enabling conditional branching or iteration in protocols (capabilities that raw MCP lacks). Overall, SCP transforms what would be a set of independent MCP conversations into a coordinated conversation among many agents within a single experimental narrative.

Integration of Wet-Lab Equipment and Complex Workflows. The most distinguishing requirement of scientific protocols (versus generic software tasks) is the integration of wet-lab experiments—interactions with physical laboratory instruments, robots, or assays. MCP was not specifically developed with lab hardware in mind; it can technically wrap instrument commands as tool APIs, but MCP does not define any standards for how lab equipment should be represented or controlled. This lack of built-in support makes integrating wet-lab steps cumbersome: every new piece of hardware might need a custom MCP adapter, and there is no universal format for lab actions (contrast this with how MCP standardized file access or HTTP requests). Moreover, scientific workflows are complex hybrids of dry (computational) and wet (experimental) steps. Under an MCP-only regime, an AI agent might plan an experiment and then output a text protocol for a human to carry out in the lab, because the agent cannot directly interface with lab gear in a standardized way. This gap significantly slows the cycle of experimentation.

SCP addresses wet-lab integration as a first-class objective. In addition to its experiment planning schema, SCP introduces explicit provisions for controlling laboratory devices and incorporating their outputs into the workflow. Specifically, SCP defines standardized device drivers and vendor-agnostic interfaces for common classes of lab equipment, much like instrument drivers in an operating system but adapted to the MCP-style JSON protocol. By having a uniform interface, an SCP-compliant lab instrument (e.g., a thermocycler, HPLC machine, or robotic arm) can be invoked in an experiment plan just as easily as a computational tool. The SCP Hub and associated edge servers handle the low-level communication with devices, so an AI agent can simply call a high-level action and the SCP middleware will translate that into the specific device commands required, irrespective of manufacturer differences. This dramatically lowers the barrier to including wet-lab steps in an automated workflow.

6 Conclusion

SCP (Scientific Context Protocol) provides an open, standardized *protocol layer* for connecting and orchestrating heterogeneous scientific resources—including **1,600+** software tools, models (including LLMs), datasets, workflow engines, and wet-lab instruments—under a unified, interoperable interface. By turning fragmented research components into composable *SCP Servers* that can be reliably discovered, invoked, and combined, SCP reduces integration overhead and enables reproducible, traceable end-to-end workflows spanning dry and wet laboratories. Moreover, a centralized *SCP Hub* maintains persistent scientific context, enforces per-experiment authentication and authorization, and manages the experiment lifecycle (planning, execution, monitoring, and archival), enabling secure cross-institution collaboration and scalable orchestration for autonomous AI scientists. Overall, SCP transforms isolated agents, tools, and instruments into an interoperable “web” of capabilities that can be safely composed as discovery services, laying foundational infrastructure for a more connected, extensible, and accelerated paradigm of agentic scientific discovery.

References

- [1] Anthropic. Introducing the model context protocol. Anthropic Blog, November 25, 2024, 2024. Accessed 2025.
- [2] Anthropic. Introducing the model context protocol. <https://www.anthropic.com/news/model-context-protocol>, 2024. Accessed: 2025-12-09.
- [3] Jinheon Baek, Sujay Kumar Jauhar, Silviu Cucerzan, and Sung Ju Hwang. Researchagent: Iterative research idea generation over scientific literature with large language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6709–6738, 2025.
- [4] Block (Square). Square mcp server: Team tools. <https://glama.ai/mcp/servers/atblock/square-mcp/tools/team>, 2025. Accessed: 2025-11-09.
- [5] Daniil A Boiko, Robert MacKnight, Ben Kline, and Gabe Gomes. Autonomous chemical research with large language models. *Nature*, 624(7992):570–578, 2023.
- [6] Cloudflare. Cloudflare. <https://www.cloudflare.com>, 2025. Accessed: 2025-11-09.
- [7] DeepSe Technologies / Bohrium. Bohrium: Ai for science – science navigator platform. <https://www.bohrium.com/>, 2025. Accessed: 2025-12-01.
- [8] Keyan Ding, Jing Yu, Junjie Huang, Yuchen Yang, Qiang Zhang, and Huajun Chen. Scitooolagent: a knowledge-graph-driven scientific agent for multitool integration. *Nature Computational Science*, pages 1–11, 2025.
- [9] Future House. Future house platform. <https://platform.futurehouse.org/>, 2025. Accessed: 2025-12-01.
- [10] Shanghua Gao, Ada Fang, Yepeng Huang, Valentina Giunchiglia, Ayush Noori, Jonathan Richard Schwarz, Yasha Ektefaie, Jovana Kondic, and Marinka Zitnik. Empowering biomedical discovery with ai agents. *Cell*, 187(22):6125–6151, 2024.
- [11] Shanghua Gao, Richard Zhu, Pengwei Sui, Zhenglun Kong, Sufian Aldogom, Yepeng Huang, Ayush Noori, Reza Shamji, Krishna Parvataneni, Theodoros Tsiligkaridis, and Marinka Zitnik. Democratizing ai scientists using tooluniverse. *arXiv preprint arXiv:2509.23426*, 2025.
- [12] Google DeepMind. A2a: Agent-to-agent communication protocol. Google Research Blog, 2025. Accessed 2025-12-09.
- [13] Juraj Gottweis, Wei-Hung Weng, Alexander Daryin, Tao Tu, Anil Palepu, Petar Sirkovic, Artiom Myaskovsky, Felix Weissenberger, Keran Rong, Ryutaro Tanno, et al. Towards an ai co-scientist. *arXiv preprint arXiv:2502.18864*, 2025.
- [14] Zijie Guo, Jiong Wang, Xiaoyu Yue, Wangxu Wei, Zhe Jiang, Wanghan Xu, Ben Fei, Wenlong Zhang, Xinyu Gu, Lijing Cheng, et al. Earthlink: A self-evolving ai agent for climate science. *arXiv e-prints*, pages arXiv–2507, 2025.
- [15] JetBrains. Jetbrains mcp server. <https://plugins.jetbrains.com/plugin/26071-mcp-server>, 2025.
- [16] Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. The ai scientist: Towards fully automated open-ended scientific discovery. *arXiv preprint arXiv:2408.06292*, 2024.
- [17] Andres M. Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D White, and Philippe Schwaller. Augmenting large language models with chemistry tools. *Nature Machine Intelligence*, 6(5):525–535, 2024.
- [18] Microsoft Copilot Studio. Introducing model context protocol (mcp) in copilot studio: Simplified integration with ai apps and agents. <https://www.microsoft.com/en-us/microsoft-copilot/blog/copilot-studio>, 2025.

- [19] Ludovico Mitchener, Angela Yiu, Benjamin Chang, Mathieu Bourdenx, Tyler Nadolski, Arvis Sulovari, Eric C Landsness, Daniel L Barabasi, Siddharth Narayanan, Nicky Evans, et al. Kosmos: An ai scientist for autonomous discovery. *arXiv preprint arXiv:2511.02824*, 2025.
- [20] Alexander Novikov, Ngan Vu, Marvin Eisenberger, Emilien Dupont, Po-Sen Huang, Adam Zsolt Wagner, Sergey Shirobokov, Borislav Kozlovskii, Francisco JR Ruiz, Abbas Mehrabian, et al. Alphaevolve: A coding agent for scientific and algorithmic discovery, 2025. URL: <https://arxiv.org/abs/2506.13131>, 2025.
- [21] Open Science Lab. Scientific intelligence context protocol (scp). <https://github.com/open-science-lab/scp>, 2025. Accessed: 2025-12-09.
- [22] OpenAI. Introducing deep research. <https://openai.com/index/introducing-deep-research/>, 2025. Accessed: 2025-11-30.
- [23] Yibo Qiu, Zan Huang, Zhiyu Wang, Handi Liu, Yiling Qiao, Yifeng Hu, Shuang Sun, Hangke Peng, Ronald X Xu, and Mingzhai Sun. Biomars: A multi-agent robotic system for autonomous biological experiments. *arXiv preprint arXiv:2507.01485*, 2025.
- [24] Replit. Replit. <https://replit.com>, 2025. Accessed: 2025-10-29.
- [25] Samuel Schmidgall, Yusheng Su, Ze Wang, Ximeng Sun, Jialian Wu, Xiaodong Yu, Jiang Liu, Zicheng Liu, and Emad Barsoum. Agent laboratory: Using llm agents as research assistants. *arXiv preprint arXiv:2501.04227*, 2025.
- [26] Shanghai Artificial Intelligence Laboratory. Shusheng scientific discovery platform. <https://discovery.intern-ai.org.cn/org/ailab/workspace/iframe?url=https://scphub.intern-ai.org.cn/>, 2025. Accessed: 2024-10-28.
- [27] Chenyang Shao, Dehao Huang, Yu Li, Keyu Zhao, Weiquan Lin, Yining Zhang, Qingbin Zeng, Zhiyu Chen, Tianxing Li, Yifei Huang, et al. Omniscientist: Toward a co-evolving ecosystem of human and ai scientists. *arXiv preprint arXiv:2511.16931*, 2025.
- [28] Jamshid Sourati and James A Evans. Accelerating science with human-aware artificial intelligence. *Nature human behaviour*, 7(10):1682–1696, 2023.
- [29] Haoran Sun, Yankai Jiang, Zhenyu Tang, Yaning Pan, Shuang Gu, Zekai Lin, Lilong Wang, Wenjie Lou, Lei Liu, Lei Bai, et al. Unleashing scientific reasoning for bio-experimental protocol generation via structured component-based reward mechanism. *arXiv preprint arXiv:2510.15600*, 2025.
- [30] K Swanson, W Wu, NL Bulaong, JE Pak, and J Zou. The virtual lab: Ai agents design new sars-cov-2 nanobodies with experimental validation. *biorxiv*, 2024. URL <https://www.biorxiv.org/content/early/2024/11/12/2024.11.11>, 2024.
- [31] Kyle Swanson, Wesley Wu, Nash L Bulaong, John E Pak, and James Zou. The virtual lab of ai agents designs new sars-cov-2 nanobodies. *Nature*, 646(8085):716–723, 2025.
- [32] Nathan J Szymanski, Bernardus Rendy, Yuxing Fei, Rishi E Kumar, Tanjin He, David Milsted, Matthew J McDermott, Max Gallant, Ekin Dogus Cubuk, Amil Merchant, et al. An autonomous laboratory for the accelerated synthesis of novel materials. *Nature*, 624(7990):86–91, 2023.
- [33] Jiabin Tang, Lianghao Xia, Zhonghang Li, and Chao Huang. Ai-researcher: Autonomous scientific innovation. *arXiv preprint arXiv:2505.18705*, 2025.
- [34] InternAgent Team, Bo Zhang, Shiyang Feng, Xiangchao Yan, Jiakang Yuan, Runmin Ma, Yusong Hu, Zhiyin Yu, Xiaohan He, Songtao Huang, et al. Internagent: When agent becomes the scientist—building closed-loop system from hypothesis to verification. *arXiv e-prints*, pages arXiv–2505, 2025.
- [35] TheiaAI/TheiaIDE. Theiaai/theiaide. https://theia-ide.org/docs/user_ai/, 2025.
- [36] H. Wang et al. Scientific discovery in the age of artificial intelligence. *Nature*, 620:47–60, 2023.

- [37] Yixuan Weng, Minjun Zhu, Qiujie Xie, Qiya Sun, Zhen Lin, Sifan Liu, and Yue Zhang. Deepscientist: Advancing frontier-pushing scientific findings progressively. *arXiv preprint arXiv:2509.26603*, 2025.
- [38] Zonglin Yang, Wanhai Liu, Ben Gao, Tong Xie, Yuqiang Li, Wanli Ouyang, Soujanya Poria, Erik Cambria, and Dongzhan Zhou. Moose-chem: Large language models for rediscovering unseen chemistry scientific hypotheses. *arXiv preprint arXiv:2410.07076*, 2024.
- [39] Zhongyue Zhang, Zijie Qiu, Yingcheng Wu, Shuya Li, Dingyan Wang, Zhuomin Zhou, Duo An, Yuhua Chen, Yu Li, Yongbo Wang, et al. Origene: A self-evolving virtual disease biologist automating therapeutic target discovery. *bioRxiv*, pages 2025–06, 2025.

A Complete SCP Server List

Table 1: Summary of current SCP servers

Server name	Domain	Description	Endpoint	Tool count
VenusFactory	Biology	VenusFactory is an AI-centric protein-engineering platform that unifies code, notebook, GUI and agent workflows around Venus and ESM protein models, covering mutation and function prediction, residue screening, data retrieval, model training, evaluation and deployment.	https://scp.intern-ai.org.cn/api/v1/mcp/1/VenusFactory	12
DrugSDATool	Biology	DrugSDATool is an integrated toolkit for drug screening, design and analysis that bundles Open Babel, RDKit and BioPython functions to retrieve data, interconvert formats, repair protein structures, normalize ligands, parse molecules, compute similarity and analyze binding pockets.	https://scp.intern-ai.org.cn/api/v1/mcp/2/DrugSDATool	33
DrugSDAModel	Biology	DrugSDAModel unites docking, pocket detection, affinity and ADMET prediction, protein structure modeling and disease-reversal scoring into one suite for end-to-end AI-driven drug discovery.	https://scp.intern-ai.org.cn/api/v1/mcp/3/DrugSDAModel	8
OrigeneChEMBL	Biology, Chemistry	Origene embeds the full ChEMBL engine for instant, on-demand querying of small-molecule bioactivity data.	https://scp.intern-ai.org.cn/api/v1/mcp/4/OrigeneChEMBL	58

OrigeneKEGG	Biology	Origene embeds the full KEGG engine for instant, on-demand pathway and functional annotation queries.	https://scp.intern-ai.org.cn/api/v1/mcp/5/OrigeneKEGG	6
OrigeneSTRING	Biology	Origene embeds the full STRING engine for instant, on-demand protein interaction and functional annotation queries.	https://scp.intern-ai.org.cn/api/v1/mcp/6/OrigeneSTRING	8
OrigeneSearch	Biology	Origene seamlessly integrates Tavily, Jina, ClinVar, GSEA, PubMed and other leading sources for one-click search and instant access.	https://scp.intern-ai.org.cn/api/v1/mcp/7/OrigeneSearch	7
OrigenePubChem	Chemistry	Origene embeds the full PubChem engine for instant, on-demand access to the world's largest open chemical database.	https://scp.intern-ai.org.cn/api/v1/mcp/8/OrigenePubChem	39
OrigeneNCBI	Biology	Origene embeds the full NCBI engine for instant, on-demand access to the world's largest open biological database.	https://scp.intern-ai.org.cn/api/v1/mcp/9/OrigeneNCBI	52
OrigeneUniProt	Biology	Origene embeds the full UniProt engine for instant, on-demand access to the authoritative protein sequence and function database.	https://scp.intern-ai.org.cn/api/v1/mcp/10/OrigeneUniProt	23
OrigeneTCGA	Biology	Origene embeds the full TCGA engine for instant, on-demand access to the authoritative cancer genomics atlas.	https://scp.intern-ai.org.cn/api/v1/mcp/11/OrigeneTCGA	3
OrigeneEnsembl	Biology	Origene embeds the full Ensembl engine for instant, on-demand access to the authoritative genome annotation and comparative database.	https://scp.intern-ai.org.cn/api/v1/mcp/12/OrigeneEnsembl	96
OrigeneUCSC	Biology	Origene embeds the full UCSC Genome Browser engine for instant, on-demand genome annotation and visualization.	https://scp.intern-ai.org.cn/api/v1/mcp/13/OrigeneUCSC	9
OrigeneFDADrug	Biology	Origene embeds the full FDA Drug engine for instant, on-demand access to authoritative drug regulatory and approval data.	https://scp.intern-ai.org.cn/api/v1/mcp/14/OrigeneFDADrug	155
OrigeneOpenTargets	Biology	Origene embeds the full Open Targets engine for instant, on-demand target discovery and validation.	https://scp.intern-ai.org.cn/api/v1/mcp/15/OrigeneOpenTargets	92

OrigeneMonarch	Biology	Origene embeds the full Monarch Initiative engine for instant, on-demand disease-phenotype-gene associations.	https://scp.intern-ai.org.cn/api/v1/mcp/16/OrigeneMonarch	3
BioInfoTools	Biology	BioInfoTools is a plug-and-play protein sequence analysis service that wraps InterProScan and BLAST into one unified API for domain detection, GO annotation and similarity search.	https://scp.intern-ai.org.cn/api/v1/mcp/17/BioInfoTools	3
ThothOP	Biology	ThothOP is the wet-lab execution engine that exposes stable atomic commands for pipetting, mixing, incubation, centrifugation and more, letting users or agents compose and run protocols safely and precisely.	https://scp.intern-ai.org.cn/api/v1/mcp/18/ThothOP	58
ThothPlan	Biology	ThothPlan is a reinforcement learning fine tuned LLM agent that turns research ideas into executable wet lab protocols, eliminating manual scripting and delivering accurate, logically sound and fully automated experiments.	https://scp.intern-ai.org.cn/api/v1/mcp/19/ThothPlan	4
Materials Mechanics and Fracture Analysis	Mechanics and Materials Science	Materials & Fracture Analysis-Tool is a one-stop library that unites stress-strain, fracture criteria, safety factors, elastic-plastic parameters, interface strength and residual stress calculations for rapid, accurate structural and failure analysis.	https://scp.intern-ai.org.cn/api/v1/mcp/20/Materials_Mechanics_and_Fracture_Analysis	107
Electrical Engineering and Circuit Calculations	Physics	Electrical & Circuit Calculation-Tool is a unified utility library that spans basic electrical quantities to advanced circuit simulation, covering DC/AC analysis, series-parallel equivalents, critical current density, magnetic field strength, electromagnetic quantification, error validation and duty-cycle detection for rapid design and experiment support.	https://scp.intern-ai.org.cn/api/v1/mcp/21/Electrical_Engineering_and_Circuit_Calculations	73

Thermal Fluid Dynamics	Physics	ThermoFluid-Tool is an integrated library for engineering thermodynamics and fluid mechanics, bundling energy conservation, heat transfer, phase change, pressure, velocity and property estimation to enable rapid numerical analysis, modeling and data processing of complex thermal-fluid systems.	https://scp.intern-ai.org.cn/api/v1/mcp/22/Thermal_Fluid_Dynamics	77
Optics and Electromagnetics	Physics	Optics & Electromagnetics-Tool is a general-purpose library that unifies light propagation, refraction, reflection, interference, electromagnetic wave parameters and field-matter interactions for lens design, optical simulation, field strength, energy density and experimental data processing.	https://scp.intern-ai.org.cn/api/v1/mcp/23/Optics_and_Electromagnetics	30
Chemistry and Reaction Calculations	Chemistry	Chemistry & Reaction Calculation-Tool is a unified toolkit that integrates reaction stoichiometry, concentration, rate and equilibrium constants, mass and energy balancing, yield estimation and solution preparation for rapid experimental and process design support.	https://scp.intern-ai.org.cn/api/v1/mcp/24/Chemistry_and_Reaction_Calculations	105
Geometry and mathematical calculations	Mathematics	Geometry & Math Calculation-Tool is a general-purpose library bundling numerical operations, algebraic derivation, function fitting, statistical analysis and spatial geometry for universal modeling, computing and engineering design.	https://scp.intern-ai.org.cn/api/v1/mcp/25/Geometry_and_mathematical_calculations	99
Data processing and statistical analysis	Information Science and Computing Technology	Data Processing & Statistical Analysis-Tool is a hands-on library that unifies cleaning, filtering, normalization, outlier detection, fitting, interpolation, error assessment, distribution and correlation analysis for research, engineering and data-driven decisions.	https://scp.intern-ai.org.cn/api/v1/mcp/26/Data_processing_and_statistical_analysis	41

Physical Quantities Conversion	Information Science and Computing Technology	Physical Quantities & Unit Conversion-Tool is an all-in-one library for converting common and specialized units, swapping SI and imperial systems, fetching physical constants, adjusting magnitudes and validating dimensions for seamless scientific and engineering calculations.	https://scp.intern-ai.org.cn/api/v1/mcp/27/Physical_Quantities_Conversion	81
InternAgent	Information Science and Computing Technology	InternAgent builds a suite of over 100 cross-domain scientific tools and deploys InternAgent-DeepResearch, an RL-tuned agent that decomposes complex tasks into dependent subtasks, explores them in parallel and self-optimizes to produce rigorous, fully referenced research reports.	https://scp.intern-ai.org.cn/api/v1/mcp/28/InternAgent	110
SciToolAgent-Bio	Biology	SciToolAgent-Bio is a one-stop toolkit covering proteomics and genomics: it computes protein properties, optimizes codons, predicts folds, aligns sequences, detects signal peptides, designs peptide libraries, maps cleavage sites, labels antibody CDRs, scores solubility and forecasts drug-target interactions.	https://scp.intern-ai.org.cn/api/v1/mcp/29/SciToolAgent-Bio	57
SciToolAgent-Mat	Mechanics and Materials Science	SciToolAgent-Mat is a materials-science toolkit that parses MOF lattices and topologies, mines Materials Project for bandgaps, elastic and dielectric data, evaluates battery voltage, capacity and cycling life, and computes thermodynamics, vibration, adsorption and stability in one systematic workflow.	https://scp.intern-ai.org.cn/api/v1/mcp/30/SciToolAgent-Mat	8
SciToolAgent-Chem	Chemistry	SciToolAgent-Chem is a unified cheminformatics suite that interconverts SMILES/InChI/CAS/-SELFIES, computes descriptors and fingerprints, predicts reactions and retrosynthesis, scores similarity, identifies functional groups, flags safety hazards, and clusters or classifies molecules with MLP/AdaBoost/RF for end-to-end drug design support.	https://scp.intern-ai.org.cn/api/v1/mcp/31/SciToolAgent-Chem	169

SCP-Workflow	Information Science and Computing Technology	SCP Workflow is the intelligent coordination hub that exposes a unified, extensible interface to discover, filter and compose any SCP tool on demand, evolving toward adaptive orchestration of wet-lab protocols.	https://scp.intern-ai.org.cn/api/v1/mcp/32/SCP-Workflow	2
--------------	--	--	---	---

B Example code

The following code demonstrates how to connect to SCP servers and invoke specialized tools to implement Case Study 3. Execution codes of more SCP examples are available in the code repository <https://github.com/InternScience/scp>.

```

import asyncio
import json
from mcp.client.streamable_http import streamablehttp_client
from mcp import ClientSession

## Server endpoints of DrugSDA-Model and DrugSDA-Tool
DrugSDA_Model_SERVER_URL = "https://scp.intern-ai.org.cn/api/v1/mcp/3/
    DrugSDA-Model"
DrugSDA_Tool_SERVER_URL = "https://scp.intern-ai.org.cn/api/v1/mcp/2/
    DrugSDA-Tool"

## Definition of the DrugSDA SCP client, including basic operations
# such as connect, disconnect, list_tools, and parse_result.
class DrugSDAClient:
    def __init__(self, server_url: str):
        self.server_url = server_url
        self.session = None

    async def connect(self):
        print(f"server url: {self.server_url}")
        try:
            self.transport = streamablehttp_client(
                url=self.server_url,
                headers={"SCP-HUB-API-KEY": "sk-xxx"})
        )
        self.read, self.write, self.get_session_id = await self.
transport.__aenter__()

        self.session_ctx = ClientSession(self.read, self.write)
        self.session = await self.session_ctx.__aenter__()

        await self.session.initialize()
        session_id = self.get_session_id()

        print(f"connect success")
        return True

    except Exception as e:
        print(f"connect failure: {e}")
        import traceback
        traceback.print_exc()
        return False

    async def disconnect(self):

```

```

    try:
        if self.session:
            await self.session_ctx.__aexit__(None, None, None)
        if hasattr(self, 'transport'):
            await self.transport.__aexit__(None, None, None)
            print("already disconnect")
    except Exception as e:
        print(f"disconnect error: {e}")

async def list_tools(self):
    try:
        tools_list = await self.session.list_tools()
        print(f"tool count: {len(tools_list.tools)}")

        for i, tool in enumerate(tools_list.tools, 1):
            print(f"{i:2d}. {tool.name}")
            if tool.description:
                desc_line = tool.description.split('\n')[0]
                print(f"    {desc_line}")

        print(f"Get tool list success")
        return tools_list.tools

    except Exception as e:
        print(f"Get tool list fail: {e}")
        return []

def parse_result(self, result):
    try:
        if hasattr(result, 'content') and result.content:
            content = result.content[0]
            if hasattr(content, 'text'):
                return json.loads(content.text)
        return str(result)
    except Exception as e:
        return {"error": f"parse error: {e}", "raw": str(result)}

## Execution code of calling tools to perform Case Study 3
async def main():
    tool_client = DrugSDAClient(DrugSDA_Tool_SERVER_URL)
    if not await tool_client.connect():
        print("connection failed")
        return

    model_client = DrugSDAClient(DrugSDA_Model_SERVER_URL)
    if not await model_client.connect():
        print("connection failed")
        return

    ## Input smiles, we don't show all the testing smiles here
    smiles_list = [
        'O=C(Nc1cccc2c1CCCC2)N1CCc2c([nH]c3cccc23)C1c1cccc(F)c1F',
        'Cc1cccccc1N1CCN(C2=Nc3cc(C1)ccc3Nc3ccc(F)cc32)CC1',
        'O=C(c1cccccc1F)N1CCN2C(=O)c3cccccc3C12c1cccc(C1)cc1',
        ...
        ...
        ...
        'O=C(NCc1cccc(-c2cccc(-c3cc4c[nH]ccc-4n3)c20)c1)Nc1ccc(F)cc1',
    ]

    ## step 1. calculate QED scores, call SCP tool
    calculate_mol_drug_chemistry
    result = await tool_client.session.call_tool(
        "calculate_mol_drug_chemistry",
        arguments={
            "smiles_list": smiles_list
        }
    ]

```

```

)
result_data = tool_client.parse_result(result)
QED_result = result_data["metrics"]
print ("Compute QED score finish ...")

## step 2. Calculate LD50 scores, call SCP tool
pred_molecule_admet
result = await model_client.session.call_tool(
    "pred_molecule_admet",
    arguments={
        "smiles_list": smiles_list
    }
)
result_data = model_client.parse_result(result)
LD50_result = result_data["admet_preds"]
print ("Predict admet finish ...")

## step 3. Filter molecules with QED larger than 0.6 and LD50_Zhu
larger than 3.0
select_smiles_list = []
for i in range(len(smiles_list)):
    smiles = smiles_list[i]
    QED = QED_result[i]["qed"]
    LD50 = LD50_result[i]["LD50_Zhu"]
    if QED >= 0.6 and LD50 >= 3.0:
        select_smiles_list.append(smiles)

print (len(select_smiles_list), select_smiles_list[0])

## step 4. Retrieve and download the target protein structure,
call SCP tool retrieve_protein_data_by_pdbcode.
pdb_code = "6vkv"
result = await tool_client.session.call_tool(
    "retrieve_protein_data_by_pdbcode",
    arguments={
        "pdb_code": pdb_code
    }
)

result_data = tool_client.parse_result(result)
pdb_path = result_data["pdb_path"]
print ("download protein structure: ", pdb_path)

## step 5. Extract main chain, call SCP tool save_main_chain_pdb
result = await tool_client.session.call_tool(
    "save_main_chain_pdb",
    arguments={
        "pdb_file_path": pdb_path,
        "main_chain_id": ""
    }
)

result_data = tool_client.parse_result(result)
pdb_path = result_data["out_file"]
print ("extract protein chain: ", pdb_path)

## step 6. Fix PDB, call SCP tool fix_pdb_dock
result = await tool_client.session.call_tool(
    "fix_pdb_dock",
    arguments={
        "pdb_file_path": pdb_path
    }
)

result_data = tool_client.parse_result(result)

```

```

pdb_path = result_data["fix_pdb_file_path"]
print ("fix protein pdb: ", pdb_path)

## step 7. Identify binding pockets, call SCP tool run_fpocket
result = await model_client.session.call_tool(
    "run_fpocket",
    arguments={
        "pdb_file_path": pdb_path
    }
)

result_data = model_client.parse_result(result)
best_pocket = result_data["pockets"][0]
print ('pocket info: ', best_pocket)

## step 8. Convert SMILES to PDBQT format, call SCP tool
convert_smiles_to_format
result = await tool_client.session.call_tool(
    "convert_smiles_to_format",
    arguments={
        "inputs": select_smiles_list,
        "target_format": "pdbqt"
    }
)
result_data = tool_client.parse_result(result)
ligand_paths = [x["output_file"] for x in result_data["convert_results"]]
print (ligand_paths)

## step 9. Convert receptor PDB to PDBQT format, call SCP tool
convert_pdb_to_pdbqt_dock
result = await tool_client.session.call_tool(
    "convert_pdb_to_pdbqt_dock",
    arguments={
        "pdb_file_path": pdb_path
    }
)
result_data = tool_client.parse_result(result)
receptor_path = result_data["output_file"]
print (receptor_path)

## step 10. Molecular docking, call SCP tool
quick_molecule_docking
result = await model_client.session.call_tool(
    "quick_molecule_docking",
    arguments={
        "receptor_path": receptor_path,
        "ligand_paths": ligand_paths,
        "center_x": best_pocket["center_x"],
        "center_y": best_pocket["center_y"],
        "center_z": best_pocket["center_z"],
        "size_x": best_pocket["size_x"],
        "size_y": best_pocket["size_y"],
        "size_z": best_pocket["size_z"]
    }
)
result_data = model_client.parse_result(result)
print ("docking finish ...")

## step 11. Select final candidate molecules with binding affinity
# smaller than -7.0 kcal/mol
final_smiles_list = []
index = 0
for item in result_data["docking_results"]:
    affinity = item['affinity']

```

```

    if affinity <= -7.0:
        final_smiles_list.append(select_smiles_list[index])
    index += 1

    print (final_smiles_list)

    await model_client.disconnect()
    await tool_client.disconnect()

if __name__ == '__main__':
    await main()

```

C Complete SCP Tool List

Table 2: The current list of all available SCP tools.

Tool Name	Description	Category
predict_protein_function	Predict protein function.	Computational Tools
predict_functional_residue	Predict functional residue.	Computational Tools
is_valid_protein_sequence	Check if the input protein sequence string is valid.	Computational Tools
is_valid_smiles	Check if the input SMILES string is valid	Computational Tools
convert_smiles_to_format	Convert a list of SMILES strings or a list of SMI file paths into other molecular formats.	Computational Tools
convert_pdb_to_pdbqt_dock	Convert a protein file from PDB format to PDBQT format for docking preparation.	Computational Tools
convert_complex_cif_to_pdb	Convert a protein-ligand complex file from CIF format to PDB format.	Computational Tools
visualize_protein	Visualize the protein structure and save as an image.	Computational Tools
visualize_molecule	Visualize the molecular structure and save as an image.	Computational Tools
visualize_complex	Visualize the protein-ligand complex structure and save as an image.	Computational Tools
fix_pdb_dock	Use PDBFixer to repair the protein structure PDB file in preparation for molecular docking.	Computational Tools
read_smi_file	Read the input smi file and extract the SMILES strings along with their corresponding compound names.	Computational Tools
read_fasta_file	Parse protein sequences from the input fasta file.	Computational Tools
calculate_mol_basic_info	Compute a set of basic molecular properties for each SMILES.	Computational Tools
calculate_mol_hydrophobicity	Compute hydrophobicity-related molecular descriptors for each SMILES.	Computational Tools

Tool Name	Description	Category
calculate_mol_hbond	Compute hydrogen bonding-related properties for each SMILES.	Computational Tools
calculate_mol_structure_complexity	Compute a set of molecular complexity descriptors for each SMILES.	Computational Tools
calculate_mol_topology	Compute a set of topological descriptors for each SMILES.	Computational Tools
calculate_mol_drug_chemistry	Compute key drug-likeness metrics for each SMILES.	Computational Tools
calculate_mol_charge	Compute Gasteiger partial charges and formal charge for each SMILES.	Computational Tools
calculate_mol_complexity	Compute custom molecular complexity-related descriptors for each SMILES.	Computational Tools
calculate_protein_sequence_properties	Compute a set of physico-chemical properties for the input protein sequence.	Computational Tools
calculate_pdb_basic_info	Read a protein pdb file and compute basic structural statistics.	Computational Tools
calculate_pdb_structural_geometry	Read a protein pdb file and compute key geometric properties based on C_{α} atom coordinates.	Computational Tools
calculate_pdb_quality_metrics	Read a protein pdb file and compute three key quality indicators.	Computational Tools
calculate_pdb_composition_info	Read a protein pdb file and analyze compositional details by counting occurrences of each atom name, residue name, and the number of atoms per chain.	Computational Tools
calculate_smiles_similarity	Compute the Tanimoto similarities between a target molecule and a list of candidate molecules using Morgan fingerprints.	Computational Tools
save_main_chain_pdb	Extract the specified chain or the longest amino-acid chain from the input protein structure file and save as a new PDB file.	Computational Tools
extract_pdb_chains	Extract the amino acid sequence of each chain from the PDB file.	Computational Tools
server_file_to_base64	Convert files smaller than 10MB to Base64 encoding.	Computational Tools
base64_to_server_file	Convert Base64 encoding back to a file.	Computational Tools
pred_pocket_prank	Use P2Rank to predict ligand binding pockets in the input protein.	Computational Tools

Tool Name	Description	Category
quick_molecule_docking	Perform molecular docking using QuickVina2-GPU.	Computational Tools
calculate_dleps_score	Enter a list of candidate small molecules.	Computational Tools
boltz_binding_affinity	Use Boltz to predict binding affinity between protein (receptor) and small molecules (ligands).	Computational Tools
run_fpocket	Use fpocket to predict protein pockets and set it as the default tool for pocket prediction.	Computational Tools
get_map_cds	Map CDS coordinates to genomic coordinates.	Computational Tools
get_variant_recoder	Translate between different variant nomenclature systems and representations.	Computational Tools
get_ld	Computes and returns LD values between the given variant and all other variants in a window.	Computational Tools
get_ld_pairwise	Computes and returns LD values between the given variants.	Computational Tools
get_ld_region	Computes and returns LD values between all pairs of variants in the defined region.	Computational Tools
get_map	Map coordinates between assemblies.	Computational Tools
get_transcript_haplotypes	Computes observed transcript haplotype sequences based on phased genotype data.	Computational Tools
post_vep_hgvs	Batch predict the functional effects of multiple variants using VEP with HGVS notation.	Computational Tools
get_track_data	Get data from a specific track for a genomic region.	Computational Tools
analyze_protein	Comprehensive protein analysis combining InterProScan (domain analysis) and BLAST (similarity search).	Computational Tools
exec_code	Complete code to perform a certain calculation.	Computational Tools
check	Perform a general validation/check operation.	Computational Tools
insert	Insert an item into a target.	Computational Tools
open	Open an item.	Computational Tools
software_analysis	Analyze real-time PCR run results using a specified analysis software.	Computational Tools
generate_executable_json	Generate executable JSON file from protocol text.	Computational Tools

Tool Name	Description	Category
extract_protocol_from_pdf	Extract experimental protocol from a provided PDF Params: pdf_url (str): PDF file URL Returns: protocol (str): protocol text Arg: arguments (dict) - Tool parameter dictionary	Computational Tools
calculate_geometric_term	Calculate the geometric term $\sqrt{\pi a}$.	Computational Tools
calculate_stress_intensity_factor	Calculate the stress intensity factor KI.	Computational Tools
determine_fracture	Determine if fracture occurs.	Computational Tools
calculate_mobility_ratio	Calculate the ratio of the new mobility relative to the original mobility.	Computational Tools
calculate_cell_volume	Calculate the unit cell volume of a crystal with lattice constant a.	Computational Tools
calculate_density	Calculate density as mass divided by volume.	Computational Tools
calculate_proportionality_constant	Calculate the proportionality constant K from known diameter, length, and weight.	Computational Tools
calculate_ring_moment_of_inertia	Calculate the moment of inertia of a ring.	Computational Tools
calculate_spoke_moment_of_inertia	Calculate the moment of inertia of a spoke (rod) about one end.	Computational Tools
calculate_total_spokes_inertia	Calculate total moment of inertia of all spokes.	Computational Tools
calculate_weight	Calculate the weight of concrete.	Computational Tools
convert_crack_length_to_meters	Convert crack length to meters based on the specified unit.	Computational Tools
calculate_critical_stress_griffith_irwin	Calculate critical stress using Griffith-Irwin equation.	Computational Tools
validate_strain_input	Validate that the axial strain is non-zero.	Computational Tools
calculate_material_density	Compute the material density based on specific gravity.	Computational Tools
calculate_volume_in_cubic_meters	Calculate the volume in cubic meters given mass and density.	Computational Tools
calculate_angular_frequency	Calculate the angular frequency of a harmonic oscillator.	Computational Tools
calculate_density_value	Calculate density in g/cm ³ .	Computational Tools
convert_GPa_to_MPa	Convert Young's modulus from GPa to MPa.	Computational Tools
determine_exponent_by_dimension	Calculate the exponent used in the Mott VRH model based on system dimension.	Computational Tools
calculate_magnetic_anisotropy_energy	Calculate the magnetic anisotropy energy.	Computational Tools
convert_gpa_to_pa	Convert elastic modulus from GPa to Pa.	Computational Tools

Tool Name	Description	Category
calculate_stiffness_contribution	Calculate the stiffness contribution ($E * A$) for a material.	Computational Tools
aggregate_stiffness	Sum individual stiffness contributions to get total stiffness.	Computational Tools
calculate_stress_in_Pa	Calculate tensile stress in Pascals.	Computational Tools
convert_Pa_to_MPa	Convert stress from Pascals to Megapascals.	Computational Tools
calculate_volume_from_mass	Calculate the volume of the oil droplet.	Computational Tools
calculate_packing_factor	Calculate atomic packing factor (APF).	Computational Tools
calculate_density_difference_percentage	Calculate percentage difference between two packing factors.	Computational Tools
validate_stress_unit	Validate that the stress unit is within the allowed units.	Computational Tools
calculate_elastic_modulus_rule_of_mixtures	Calculate elastic modulus of alloy using rule of mixtures.	Computational Tools
compute_moment_of_inertia	Calculate the moment of inertia of a hollow sphere about its diameter.	Computational Tools
calculate_hall_petch_strength	Calculate yield strength using Hall-Petch equation.	Computational Tools
calculate_strain	Calculate the strain (dimensionless) from elongation and original length.	Computational Tools
calculate_energy_density	Calculate the elastic energy density (J/m^3) from Young's modulus and strain.	Computational Tools
calculate_total_energy	Calculate the total elastic energy stored in the material.	Computational Tools
convert_stress_to_pa	Convert stress to Pascals (Pa).	Computational Tools
convert_pa_to_output_unit	Convert a stress value from Pascals to the specified output unit.	Computational Tools
compute_compression_ratio	Compute the compression ratio.	Computational Tools
calculate_mass_from_volume_density	Calculate mass from volume and density.	Computational Tools
calculate_weight_N	Calculate weight in Newtons.	Computational Tools
calculate_buoyancy_force_N	Calculate buoyant force in Newtons.	Computational Tools
calculate_thickness_nm	Calculate total thickness in nanometers.	Computational Tools
compute_transverse_strain_from_poisson	Calculate the transverse strain using Poisson's ratio and longitudinal strain.	Computational Tools
calculate_length_change	Calculate the length change due to thermal expansion.	Computational Tools
calculate_final_length	Calculate the final length after thermal contraction.	Computational Tools
calculate_relative_length_change	Calculate the relative change in length as a percentage.	Computational Tools

Tool Name	Description	Category
calculate_minimum_thickness_for_waterproofing	Calculate the minimum coating thickness needed for waterproofing ($\theta \geq 90^\circ$).	Computational Tools
calculate_atomic_radius_bcc	Calculate the atomic radius for BCC structure.	Computational Tools
calculate_density_bcc	Calculate the density of a BCC structured material.	Computational Tools
calculate_stress	Calculate stress according to Hooke's law.	Computational Tools
check_debonding	Determine whether interface debonding will occur.	Computational Tools
calculate_safety_factor	Calculate safety factor.	Computational Tools
convert_mpa_to_pa	Convert stress from MPa to Pascals.	Computational Tools
calculate_plastic_section_modulus	Calculate the plastic section modulus Z for a rectangular cross-section.	Computational Tools
calculate_plastic_moment	Calculate the plastic moment Mp in kN·m.	Computational Tools
compute_density_value	Compute the density from mass and volume.	Computational Tools
convert_GPa_to_Pa	Convert shear modulus from GPa to Pa.	Computational Tools
calculate_solid_cylinder_inertia	Calculate the moment of inertia of a hollow cylinder about its central axis.	Computational Tools
calculate_final_roughness	Calculate the final surface roughness after reduction.	Computational Tools
calculate_hardness_increase_percent	Calculate the percentage increase in hardness.	Computational Tools
calculate_impact_toughness_in_j_per_m2	Calculate impact toughness in J/m^2 .	Computational Tools
convert_j_per_m2_to_kj_per_m2	Convert impact toughness from J/m^2 to kJ/m^2 .	Computational Tools
calculate_solids_volume	Calculate volume occupied by suspended solids (m^3/m^2).	Computational Tools
calculate_stress_amplitude_value	Calculate stress amplitude.	Computational Tools
calculate_mean_stress	Calculate mean stress.	Computational Tools
calculate_stress_ratio	Calculate stress ratio $R = \sigma_{min}/\sigma_{max}$.	Computational Tools
compute_numerator	Compute the numerator for Jurin's law: $2 \times \sigma \times \cos(\theta)$.	Computational Tools
sort_stresses_descending	Sorts a list of three stresses in descending order.	Computational Tools
calculate_max_shear_stress	Calculates the maximum shear stress from principal stresses.	Computational Tools
calculate_plastic_zone_radius	Calculate the plastic zone radius.	Computational Tools
calculate_ssa	Calculate specific surface area (SSA) based on constant k and diameter.	Computational Tools
d_spacing_bcc	Calculate the interplanar spacing for a BCC crystal.	Computational Tools

Tool Name	Description	Category
is_allowed_reflection_bcc	Check if the (h,k,l) plane is allowed for diffraction in BCC structure.	Computational Tools
calculate_min_grain_size_from_corrosion_rate	Calculate minimum grain size according to maximum corrosion rate and porosity.	Computational Tools
calculate_elongation	Calculate the elongation of the specimen.	Computational Tools
check_allowable_stress	Determine if interface stress exceeds allowable stress.	Computational Tools
find_zero_strain_conductivity	Find the conductivity at zero strain.	Computational Tools
compute_characteristic_length	Calculate the characteristic length as the ratio of diffusion coefficient to front velocity.	Computational Tools
convert_modulus_to_SI	Convert Young's modulus to Pascals.	Computational Tools
calculate_total_mass	Calculate the total mass of a material.	Computational Tools
calculate_shear_stress_Pa	Calculate shear stress in pascals.	Computational Tools
assess_safety	Assess safety based on shear stress and allowable stress.	Computational Tools
calculate_rectangular_inertia	Calculate the moment of inertia for a rectangular cross-section.	Computational Tools
calculate_theoretical_packing_factor	Return the theoretical packing factor for FCC.	Computational Tools
calculate_stress_amplitude	Calculate the stress amplitude from maximum and minimum stresses.	Computational Tools
calculate_fatigue_life_Nf	Calculate the fatigue life (number of cycles) from stress ratio, fatigue strength coefficient, and exponent.	Computational Tools
determine_elastic_state	Determine if material is in elastic deformation stage.	Computational Tools
calculate_effective_elastic_modulus	Calculate effective elastic modulus of porous materials.	Computational Tools
set_material_parameters	Assign default parameters for a given material type.	Computational Tools
calculate_kd_over_sqrt_d	Calculate kd divided by sqrt(d).	Computational Tools
calculate_yield_strength	Calculate the yield strength based on σ_0 and $\frac{kd}{\sqrt{d}}$.	Computational Tools
calculate_wire_weight	Calculate the weight of the wire (gravitational force).	Computational Tools
calculate_moment_of_inertia.Solid_sphere	Calculate the moment of inertia for a solid sphere.	Computational Tools
determine_stronger_material	Determine which material has higher remaining strength.	Computational Tools
calculate_initial_volume	Calculate the initial volume of a material.	Computational Tools
calculate_maximum_force	Calculate the maximum compressive force.	Computational Tools

Tool Name	Description	Category
calculate_modulus_difference	Calculate the difference between fiber and matrix elastic moduli.	Computational Tools
calculate_fiber_contribution	Calculate the contribution of fibers to the composite elastic modulus.	Computational Tools
calculate_mass_from_density_volume	Calculate mass from density and volume.	Computational Tools
calculate_polar_moment_of_inertia	Calculate the polar moment of inertia $J = \frac{\pi}{2} \cdot r^4$.	Computational Tools
calculate_allowable_shear_stress	Calculate allowable shear stress, which is half of the yield strength.	Computational Tools
determine_plastic_deformation	Determine if plastic deformation will occur.	Computational Tools
calculate_material_volume	Calculate the volume of a material based on its mass and density.	Computational Tools
calculate_surface_roughness	Calculate surface roughness (Ra) based on thickness.	Computational Tools
calculate_interface_stress	Calculate interface stress.	Computational Tools
convert_resistance_kOhm_to_Ohm	Convert resistance from kilo-ohms to ohms.	Computational Tools
calculate_voltage	Calculate required voltage using Ohm's Law $V = I \times R$.	Computational Tools
calculate_noise_current	Calculate noise current.	Computational Tools
compute_capacitance_value	Calculate the capacitance in Farads based on physical parameters.	Computational Tools
calculate_initial_charge	Calculate the electric charge in a capacitor.	Computational Tools
calculate_new_capacitance	Calculate the capacitance after inserting dielectric.	Computational Tools
calculate_voltage_from_charge_and_capacitance	Calculate voltage from charge and capacitance.	Computational Tools
calculate_cell_potential	Calculate the cell potential (V) from cathode and anode potentials.	Computational Tools
calculate_conductivity	Compute the electrical conductivity using the Mott VRH model.	Computational Tools
calculate_electric_field	Calculate the electric field strength between two plates.	Computational Tools
calculate_total_charge	Calculate the total charge on the oil droplet.	Computational Tools
calculate_mass_from_electric_force	Calculate the mass of the oil droplet based on electric force balance.	Computational Tools
calculate_charge	Calculate the charge stored on a capacitor.	Computational Tools
calculate_energy_difference_eV	Calculate the energy difference between two energy levels (in eV).	Computational Tools
calculate_resistivity	Calculate resistivity from conductivity.	Computational Tools

Tool Name	Description	Category
calculate_vacuum_capacitance	Calculate the vacuum capacitance of a parallel plate capacitor.	Computational Tools
calculate_energy_stored	Calculate the energy stored in a capacitor.	Computational Tools
calculate_potential_at_position	Calculate the electric potential at a given position ratio between plates.	Computational Tools
calculate_electric_field_strength	Calculate the electric field strength produced by a point charge.	Computational Tools
calculate_force_on_charge	Calculate the force experienced by a charge in an electric field.	Computational Tools
calculate_parallel_resistance	Calculate the equivalent resistance of resistors in parallel.	Computational Tools
calculate_output_voltage	Calculate the maximum output voltage of a series-connected battery pack.	Computational Tools
calculate_duty_cycle	Calculate the duty cycle for a boost converter.	Computational Tools
calculate_induced_emf_magnitude	Calculate the magnitude of the induced EMF.	Computational Tools
calculate_voltage_across_resistor	Calculate the voltage across a resistor using Ohm's Law.	Computational Tools
calculate_output_current	Calculate the current needed for a specified output power given the battery voltage.	Computational Tools
calculate_total_current	Calculate total current consumption including quiescent and output currents.	Computational Tools
calculate_battery_life_hours	Calculate the battery life in hours.	Computational Tools
calculate_rc_time_constant	Calculate the RC time constant (τ) for an RC circuit.	Computational Tools
calculate_absolute_permittivity	Calculate the absolute permittivity of a dielectric material.	Computational Tools
calculate_capacitance	Calculate the capacitance of a parallel plate capacitor.	Computational Tools
calculate_stored_energy	Calculate the energy stored in a capacitor.	Computational Tools
calculate_current_from_heat	Calculate the current required to generate a specified heat energy in a resistor over a given time.	Computational Tools
calculate_vacuum_permeability	Calculate the vacuum magnetic permeability (μ_0).	Computational Tools
calculate_magnetic_flux_density	Calculate the magnetic flux density B.	Computational Tools
calculate_vacuum_permittivity	Return the vacuum permittivity (epsilon_0) in F/m.	Computational Tools
calculate_charge_coulombs	Convert energy to charge (Coulombs).	Computational Tools
convert_coulombs_to_mA	Convert Coulombs to milliamp-hours.	Computational Tools

Tool Name	Description	Category
calculate_time_constant	Calculate the RC time constant.	Computational Tools
calculate_collector_current	Calculate the collector current in a transistor circuit.	Computational Tools
calculate_emitter_current	Calculate the emitter current in a transistor circuit.	Computational Tools
calculate_voltage_drop	Calculate voltage drop across a resistor.	Computational Tools
calculate_power_supply_voltage	Calculate the power supply voltage Vcc.	Computational Tools
calculate_voltage_formula	Calculate Vcc using the analytical formula for verification.	Computational Tools
calculate_total_turns	Calculate the total number of turns in a solenoid.	Computational Tools
calculate_turns_per_meter	Calculate the number of turns per meter for a solenoid.	Computational Tools
calculate_conduction_electrons	Calculate the number of conduction electrons.	Computational Tools
calculate_current	Calculate the current in a circuit.	Computational Tools
calculate_terminal_voltage	Calculate the terminal voltage of the battery.	Computational Tools
calculate_resistivity_at_temperature	Calculate resistivity at a specific temperature.	Computational Tools
calculate_thickness_from_resistance	Calculate the coating thickness in meters.	Computational Tools
calculate_new_short_circuit_current	Calculate new short-circuit current.	Computational Tools
calculate_new_max_power	Calculate maximum power output (W).	Computational Tools
calculate_effective_voltage	Calculate the effective voltage after voltage drop.	Computational Tools
calculate_voltage_drop_percent	Calculate the percentage of voltage drop.	Computational Tools
calculate_average_current	Calculate the average current over a cycle.	Computational Tools
calculate_average_voltage_drop	Calculate the average voltage drop over a cycle.	Computational Tools
calculate_velocity	Calculate the velocity of a charged particle in a uniform magnetic field.	Computational Tools
calculate_conductivity_ratio	Calculate the conductivity ratio assuming proportionality to mobility.	Computational Tools
calculate_new_conductivity	Calculate the new conductivity if original conductivity is known.	Computational Tools
calculate_power_consumption	Calculate the power consumption of a resistor.	Computational Tools
assess_resistor_sufficiency	Assess whether the resistor's rated power is sufficient and provide recommendation.	Computational Tools
determine_particle_charge	Determine the electric charge of a particle based on its type.	Computational Tools

Tool Name	Description	Category
calculate_resistance_change_percentage	Calculate percentage change in resistance.	Computational Tools
calculate_lsb_voltage	Calculate voltage per LSB for an ADC.	Computational Tools
calculate_inherent_noise_voltage	Convert inherent noise in LSB to voltage value.	Computational Tools
find_zero_field_jc	Find the critical current density J_{c0} at zero magnetic field.	Computational Tools
find_zero_jc_field	Find the magnetic field strength H_0 at zero critical current density.	Computational Tools
determine_flux_quanta_integer	Round the exact flux quantum number to an integer using specified method.	Computational Tools
calculate_new_mobility	Calculate the new mobility after decrease.	Computational Tools
calculate_absolute_decrease	Calculate the absolute decrease between two values.	Computational Tools
calculate_minimum_field	Calculate the minimum magnetic field strength needed for magnetization.	Computational Tools
verify_duty_cycle_range	Verify that the duty cycle is within the valid range [0, 1].	Computational Tools
calculate_heat_released	Calculate heat released during combustion.	Computational Tools
calculate_net_buoyancy_coefficient	Calculate the net buoyancy coefficient from the density ratio.	Computational Tools
calculate_acceleration	Calculate the upward acceleration of the hot air balloon.	Computational Tools
calculate_ratio	Calculate the ratio of target phonon mean free path to reference value.	Computational Tools
calculate_new_speed	Calculate the new welding speed after applying a speed factor.	Computational Tools
calculate_new_power	Calculate the new laser power to maintain welding depth after increasing speed.	Computational Tools
calculate_theoretical_depth	Calculate the theoretical welding depth.	Computational Tools
convert_celsius_to_kelvin	Convert temperature from Celsius to Kelvin.	Computational Tools
calculate_thermal_energy	Calculate the thermal energy stored during a phase change.	Computational Tools
celsius_to_kelvin	Convert Celsius temperature to Kelvin.	Computational Tools
get_water_vapor_pressure	Get water vapor pressure at a specific temperature.	Computational Tools
calculate_thermal_resistance	Calculate the thermal resistance $R = d / k$.	Computational Tools
calculate_temperature_change	Calculate the temperature difference.	Computational Tools

Tool Name	Description	Category
explain_relationship_between_alpha_and_m	Explain the relationship between thermal expansion coefficient (α) and thermal strain rate sensitivity index (m).	Computational Tools
aggregate_results	Aggregate intermediate values and results into a dictionary.	Computational Tools
calculate_heat_absorbed	Calculate heat absorbed.	Computational Tools
calculate_heat_energy	Calculate the required heat energy.	Computational Tools
calculate_specific_heat_capacity	Calculate the specific heat capacity (c) using heat energy, mass, and temperature change.	Computational Tools
get_degeneracy	Return degeneracy array for each energy level.	Computational Tools
calculate_energy_ratio	Calculate the ratio of new energy to initial energy.	Computational Tools
calculate_thermal_conductivity_from_ratio	Calculate thermal conductivity of the target material from a known reference and ratio.	Computational Tools
generate_analytical_expression	Generate a string for analytical temperature distribution expression.	Computational Tools
calculate_temperature_difference	Calculate the temperature difference.	Computational Tools
calculate_cooling_time	Calculate the cooling time.	Computational Tools
compute_temperature_change	Calculate the temperature change.	Computational Tools
compute_temperature_gradient	Calculate the temperature gradient ($^{\circ}\text{C}/\text{m}$) along a material.	Computational Tools
calculate_heat_flux	Calculate the heat flux (W/m^2) using Fourier's law.	Computational Tools
calculate_total_heat_conduction_rate	Calculate the total heat conduction rate (W) through a material.	Computational Tools
calculate_kinetic_energy	Calculate kinetic energy from energy density and volume.	Computational Tools
convert_temperature_to_celsius	Convert normalized temperature to Celsius.	Computational Tools
calculate_temperature_points	Calculate temperature points T1 and T2 based on T_c and their ratios.	Computational Tools
calculate_energy_released	Calculate the energy released during phase change.	Computational Tools
calculate_potential_energy	Calculate the gravitational potential energy.	Computational Tools
calculate_absorbed_heat_energy	Calculate the heat energy absorbed.	Computational Tools
calculate_temperature_increase	Calculate the temperature increase.	Computational Tools
calculate_thermal_expansion_difference	Calculate the difference between the thermal expansion coefficients of two materials.	Computational Tools

Tool Name	Description	Category
convert_to_kelvin	Convert temperature to Kelvin.	Computational Tools
calculate_carnot_efficiency_value	Calculate Carnot efficiency.	Computational Tools
calculate_heat_flux_value	Calculate the heat flux using Fourier's law.	Computational Tools
calculate_heat_flow_rate	Calculate the total heat flow rate.	Computational Tools
calculate_heat_resistance_sum	Calculate the total thermal resistance by summing individual resistances.	Computational Tools
include_optional_resistance	Add optional resistance to total if specified.	Computational Tools
calculate_heat_required	Calculate the heat energy required for heating.	Computational Tools
calculate_heat_opposite	Calculate the opposite (negation) of a heat value.	Computational Tools
calculate_melted_mass_from_energy	Calculate the mass of ice melted from the given energy.	Computational Tools
adjust_energy_for_efficiency	Adjust energy to account for transfer efficiency.	Computational Tools
calculate_ideal_gas_work	Calculate the work done during an isothermal process for an ideal gas.	Computational Tools
calculate_internal_energy_change	Calculate the change in internal energy for an ideal gas during an isothermal process.	Computational Tools
calculate_heat_transfer	Calculate the heat transfer based on the first law of thermodynamics.	Computational Tools
convert_energy_MeV_to_J	Convert energy from MeV to Joules (J).	Computational Tools
calculate_total_heat	Calculate total heat based on enthalpy and moles.	Computational Tools
calculate_heat_per_gram	Calculate heat per gram of adsorbent.	Computational Tools
compute_temperature_difference	Calculate the temperature difference across a flat plate.	Computational Tools
compute_boltzmann_constant	Return the Boltzmann constant (J/K).	Computational Tools
calculate_heat_per_volume	Calculate heat generation rate per unit volume.	Computational Tools
calculate_ventilation_heat_coefficient	Calculate heat loss coefficient due to ventilation.	Computational Tools
calculate_final_temperature	Calculate the final temperature in Kelvin.	Computational Tools
calculate_time_per_mm	Calculate time (seconds) to weld per millimeter.	Computational Tools
calculate_heat_input	Calculate heat input in Joules per millimeter.	Computational Tools
compute_clearing_temperature	Calculate the clearing temperature of a liquid crystal polymer.	Computational Tools
calculate_dynamic_viscosity	Calculate dynamic viscosity $\mu = \nu \cdot \rho$.	Computational Tools

Tool Name	Description	Category
calculate_velocity_gradient_at_wall	Calculate the velocity gradient at $y = -h$ for the flow profile.	Computational Tools
calculate_drag_coefficient	Calculate the drag coefficient (C_d) based on roughness.	Computational Tools
calculate_drag_force	Calculate the drag force based on fluid properties and drag coefficient.	Computational Tools
calculate_total_influent_volume	Calculate total influent volume (m^3/m^2) over a given period.	Computational Tools
calculate_filter_pore_volume	Calculate the pore volume of the filter (m^3/m^2).	Computational Tools
calculate_fraction_occupied	Calculate the fraction of filter pore volume occupied by solids.	Computational Tools
calculate_traditional_infiltration	Calculate infiltration through traditional pavement.	Computational Tools
calculate_permeable_infiltration	Calculate infiltration through permeable pavement.	Computational Tools
calculate_additional_infiltration	Calculate the additional infiltration volume.	Computational Tools
calculate_water_mass	Calculate water mass in grams.	Computational Tools
compute_denominator	Compute the denominator for Jurin's law: $\rho * g * d$.	Computational Tools
calculate_capillary_height	Calculate the capillary rise height.	Computational Tools
calculate_liquid_mass	Calculate the mass of the liquid.	Computational Tools
compute_density_ratio	Calculate the ratio of two densities.	Computational Tools
calculate_permeability	Calculate gas permeability.	Computational Tools
calculate_minimum_release_height	Calculate the minimum release height for a rolling solid sphere on a circular track.	Computational Tools
calculate_incident_photon_rate	Calculate incident photon rate.	Computational Tools
calculate_power_increase_factor	Calculate the multiplicative effect of laser power increase on scattered photon rate.	Computational Tools
calculate_reflectance_fresnel	Calculate reflectance at normal incidence using Fresnel equation.	Computational Tools
compute_wavenumber	Calculate the wavenumber for a hydrogen spectral line.	Computational Tools
compute_wave_number_ratio	Calculate the wave number ratio k'/k in a dielectric material.	Computational Tools
calculate_wave_number_in_medium	Calculate the wave number in a medium.	Computational Tools
calculate_transmission_coefficient	Calculate the transmission coefficient at the interface.	Computational Tools
calculate_min_wavelength	Calculate the minimum wavelength based on film thickness.	Computational Tools

Tool Name	Description	Category
calculate_max_frequency	Calculate the maximum frequency for wave propagation.	Computational Tools
calculate_frequency_range	Calculate the frequency range of electromagnetic waves.	Computational Tools
calculate_absorption_ratio	Calculate the absorption ratio based on absorption coefficient and thickness.	Computational Tools
convert_wavelength_nm_to_meters	Convert wavelength from nanometers to meters.	Computational Tools
generate_wave_functions	Generate wave function expressions for each quantum number.	Computational Tools
calculate_photon_energy_J	Calculate photon energy in joules.	Computational Tools
determine_sufficiency	Determine if photon energy is sufficient to excite electron.	Computational Tools
calculate_numerical_aperture	Calculate the numerical aperture (NA) of an optical fiber.	Computational Tools
calculate_emission_rate_with_enhancement	Calculate the emission rate W given free space rate and enhancement.	Computational Tools
calculate_field_change_rate	Calculate the rate of change of magnetic field.	Computational Tools
electron_wavelength	Calculate the electron wavelength considering relativistic effects.	Computational Tools
bragg_angle	Calculate the Bragg angle in radians.	Computational Tools
calculate_intensity	Calculate laser intensity.	Computational Tools
calculate_radiation_pressure	Calculate radiation pressure.	Computational Tools
calculate_photon_energy_eV	Calculate photon energy in eV from wavelength.	Computational Tools
calculate_total_power	Calculate total power in the spot area.	Computational Tools
calculate_photon_flux	Calculate the photon flux per second.	Computational Tools
compute_minimum_thickness	Calculate the minimum film thickness for minimal normal incidence reflection.	Computational Tools
calculate_irradiance_ratio	Calculate the ratio of new irradiance to standard irradiance.	Computational Tools
calculate_new_mpp_current	Calculate new max power point current.	Computational Tools
compute_signal_difference	Compute the difference between light and dark signal means.	Computational Tools
convert_wavelength_to_meters	Convert wavelength to meters.	Computational Tools
calculate_volume_ml	Calculate volume in milliliters from mass and density.	Computational Tools
calculate_molar_mass	Calculate the molar mass of a compound.	Computational Tools

Tool Name	Description	Category
convert_mass_to_moles	Convert mass to moles.	Computational Tools
calculate_cell_mass	Calculate the mass of the unit cell based on density and volume.	Computational Tools
calculate_atoms_per_cell	Calculate number of atoms per unit cell.	Computational Tools
calculate_volume_in_liters	Calculate volume in liters.	Computational Tools
calculate_molarity	Calculate molarity of the solution.	Computational Tools
calculate_pH_from_pOH	Calculate pH from pOH.	Computational Tools
calculate_molar_mass_ratio	Calculate the square root of the ratio of two molar masses.	Computational Tools
calculate_solute_mass	Calculate the mass of solute in a saturated solution.	Computational Tools
calculate_solution_mass	Calculate total mass of the saturated solution.	Computational Tools
calculate_mass_percent	Calculate the mass percentage of the solute in the solution.	Computational Tools
calculate_reaction_rate_constant	Calculate the reaction rate constant k from the half-life.	Computational Tools
calculate_initial_moies	Calculate initial moles of gas using $PV = nRT$.	Computational Tools
calculate_molecules_from_moies	Calculate the number of molecules from moles.	Computational Tools
calculate_remaining_molecules_half_life	Calculate remaining molecules after reaction_time_hours based on half-life.	Computational Tools
calculate_hydrogen_energy_level	Calculate the energy of an electron in a hydrogen atom at principal quantum number n.	Computational Tools
calculate_ionization_energy	Calculate the ionization energy as the absolute value of the energy level.	Computational Tools
calculate_moies_hydrogen	Calculate moles of hydrogen gas using ideal gas law.	Computational Tools
calculate_moies	Calculate the amount (moles) of gas.	Computational Tools
calculate_partial_pressure	Calculate the partial pressure of a gas (atm).	Computational Tools
calculate_solution_volume	Calculate solution volume in mL from mass and density.	Computational Tools
determine_electron_transfer_number	Determine the number of electrons transferred in the reaction.	Computational Tools
calculate_gibbs_free_energy	Calculate Gibbs free energy change (J/mol).	Computational Tools
calculate_decay_constant	Calculate the decay constant λ given the known final mass, initial mass, and elapsed time.	Computational Tools
calculate_half_life	Calculate the half-life $T_{1/2}$ from decay constant λ .	Computational Tools

Tool Name	Description	Category
calculate_remaining_mass	Calculate the remaining mass after a certain time using decay constant.	Computational Tools
calculate_content_mass	Calculate the mass of the content substance.	Computational Tools
calculate_moies_from_mol_per cent	Calculate the molar amount of a component given total moles and mol%.	Computational Tools
get_molar_mass	Retrieve the molar mass of a specified acid.	Computational Tools
convert_density_to_g_per_L	Convert density from g/mL to g/L.	Computational Tools
calculate_acid_mass_in_solutio n	Calculate the mass of acid in 1 liter of solution.	Computational Tools
calculate_mass	Calculate mass of a component based on weight fraction.	Computational Tools
convert_weight_percentage_to _fraction	Convert weight percentage to weight fraction.	Computational Tools
compute_final_concentration	Calculate the final doping concentration after growth.	Computational Tools
get_molecular_mass_amu	Return molecular mass in amu for a given molecule type.	Computational Tools
calculate_moies_deposited	Calculate the moles of Ni deposited.	Computational Tools
calculate_mass_deposited	Calculate the mass of Ni deposited in grams.	Computational Tools
calculate_remaining_moies	Calculate remaining moles of Ni ions after deposition.	Computational Tools
calculate_final_moies	Calculate the target moles of Ni ions after treatment.	Computational Tools
calculate_moies_to_adsorb	Calculate the moles of Ni ions to be adsorbed.	Computational Tools
calculate_adsorbent_mass	Calculate the mass of adsorbent in kilograms.	Computational Tools
calculate_substance_mass	Calculate the mass of the solute in grams.	Computational Tools
calculate_pure_water_solubilit y	Calculate $Mg(OH)_2$ solubility in pure water.	Computational Tools
calculate_naoh_solution_solubi lity	Calculate $Mg(OH)_2$ solubility in NaOH solution.	Computational Tools
calculate_mass_in_grams	Calculate mass in grams from volume and density.	Computational Tools
calculate_required_hydroxide_ moles	Calculate the required moles of calcium hydroxide based on acetic acid moles.	Computational Tools
convert_concentration_mgL_to _gm3	Convert concentration from mg/L to g/m ³ .	Computational Tools
compute_decay_constant	Calculate decay constant λ from half-life.	Computational Tools
calculate_time_from_activity_r atio	Calculate time t from decay constant and activity ratio.	Computational Tools
calculate_total_moies	Calculate the total moles of a substance.	Computational Tools
calculate_total_atoms	Calculate total number of atoms from moles.	Computational Tools

Tool Name	Description	Category
calculate_nanoparticle_count	Calculate the number of nanoparticles.	Computational Tools
calculate_nanoparticle_moies	Calculate the total moles of nanoparticles.	Computational Tools
calculate_nanoparticle_concentration	Calculate the molar concentration of nanoparticles.	Computational Tools
calculate_scaling_factor	Calculate the scaling factor based on actual and reference PVC masses.	Computational Tools
calculate_additive_mass	Calculate the additive mass needed based on the scaling factor.	Computational Tools
calculate_co2_moies_from_propane	Calculate the moles of CO2 produced from propane.	Computational Tools
calculate_bond_order	Calculate the bond order given bonding and antibonding electrons.	Computational Tools
estimate_bond_length	Estimate the bond length for a given bond type.	Computational Tools
analyze_bond_length_vs_bond_order	Generate an analysis statement relating bond order to bond length.	Computational Tools
compute_time_from_alpha	Calculate the curing time needed to reach a target degree of curing.	Computational Tools
calculate_alpha_at_time	Calculate the degree of curing at time t.	Computational Tools
calculate_corrosion_rate	Calculate corrosion rate.	Computational Tools
calculate_cells_per_mole	Calculate the number of unit cells in one mole of the compound.	Computational Tools
calculate_total_volume_per_mole	Calculate the total volume occupied by one mole of the compound.	Computational Tools
calculate_molar_mass_from_volume_density	Calculate the molar mass of the compound.	Computational Tools
calculate_component_mass	Calculate the mass of a component.	Computational Tools
compute_oxides_masses	Calculate the mass of each oxide.	Computational Tools
calculate_component_masses	Calculate zinc and copper masses from total mass and zinc percentage.	Computational Tools
calculate_atoms	Calculate the number of atoms from moles.	Computational Tools
calculate_total_molecules	Calculate total gas molecules based on volume and molecules per cc.	Computational Tools
calculate_atomic_radius	Calculate the atomic radius in FCC structure.	Computational Tools
calculate_two_radii	Calculate twice the atomic radius.	Computational Tools
calculate_crystallinity_percentge	Calculate the crystallinity percentage based on density measurements.	Computational Tools

Tool Name	Description	Category
calculate_plating_time	Calculate the plating time needed to reach the target thickness.	Computational Tools
calculate_proof	Calculate alcohol proof from volume percentage.	Computational Tools
calculate_life_material_mass	Calculate required $LiFePO_4$ cathode material mass.	Computational Tools
calculate_number_of_atoms	Calculate the number of atoms from moles.	Computational Tools
determine_Z_number	Determine the number of atoms per unit cell based on crystal structure.	Computational Tools
calculate_mass_of_sodium_benzoate	Calculate the mass of sodium benzoate.	Computational Tools
compute_fukui_plus	Calculate the nucleophilic Fukui function f^+ .	Computational Tools
compute_fukui_minus	Calculate the electrophilic Fukui function f^- .	Computational Tools
compute_dual_descriptor	Calculate the Dual Descriptor f.	Computational Tools
explain_physical_meaning	Return detailed explanation of the physical meaning of adsorption enthalpy (ΔH_{ads}).	Computational Tools
calculate_total_coordination_number	Calculate total coordination number based on cyclic ligand and denticity and additional ligands.	Computational Tools
determine_typical_lanthanide_range	Return the typical coordination number range for lanthanide elements.	Computational Tools
assess_coordination_number_range	Assess whether the total coordination number is within the typical lanthanide range.	Computational Tools
calculate_distance	Calculate the sum of two ionic radii, representing the distance between ions.	Computational Tools
calculate_tolerance_factor	Calculate the tolerance factor t.	Computational Tools
assess_stability	Assess perovskite structure stability based on tolerance factor t.	Computational Tools
parse_electron_configuration	Parse electron configuration string, extract number of d electrons.	Computational Tools
calculate_expected_electrons	Calculate expected d electron count based on oxidation state, for verification.	Computational Tools
calculate_spin_S	Calculate total spin angular momentum S from number of d electrons.	Computational Tools
calculate_standard_cell_potential	Calculate the standard cell potential (E°) from cathode and anode potentials.	Computational Tools
calculate_reaction_quotient	Calculate the reaction quotient Q for the cell reaction.	Computational Tools

Tool Name	Description	Category
calculate_single_suppository_mass	Calculate the mass of a single suppository.	Computational Tools
calculate_gas_constant	Return gas constant R in J/(mol·K).	Computational Tools
calculate_faraday_constant	Return Faraday constant F in C/mol.	Computational Tools
calculate_coefficient_RT_over_nF	Calculate the coefficient (RT)/(nF) for the Nernst equation.	Computational Tools
generate_molecular_weight_range	Generate a sequence of molecular weight values within a specified range.	Computational Tools
calculate_total_solids_mass	Calculate total mass of suspended solids in grams.	Computational Tools
calculate_final_density	Calculate the final density of a material.	Computational Tools
calculate_total_peg_mass	Calculate total PEG mass needed for all suppositories.	Computational Tools
calculate_length_plus_width	Calculate the sum of length and width given the perimeter.	Computational Tools
calculate_length_times_width	Calculate the product of length and width given the volume and height.	Computational Tools
adjust_dimensions_order	Ensure that length >= width in the dimensions tuple.	Computational Tools
select_final_dimensions	Select the first dimension set as the final dimensions.	Computational Tools
round_to_decimal_places	Round a value to a specified number of decimal places.	Computational Tools
calculate_area	Calculate the area of a rectangle.	Computational Tools
calculate_theta_rad	Calculate the angle Θ in radians from shear strain.	Computational Tools
convert_rad_to_deg	Convert radians to degrees.	Computational Tools
calculate_phi_deg	Calculate Φ in degrees from Θ in degrees.	Computational Tools
calculate_cot_2phi	Calculate $\cot(2\Phi)$ as $\tan(\Theta)$.	Computational Tools
verify_cot_value	Verify if $\cot(2\Phi)$ is close to $\text{shear_strain} / 2$.	Computational Tools
calculate_volume	Calculate the volume of a rectangular prism.	Computational Tools
calculate_volume_cm3	Calculate volume in cubic centimeters.	Computational Tools
round_to_nearest_int	Round a float to the nearest integer.	Computational Tools
calculate_segment_area	Calculate the area of a curb segment.	Computational Tools
sum_areas	Sum a list of area values.	Computational Tools
round_volume_value	Round the volume in cubic feet to a specified number of decimal places.	Computational Tools
calculate_cross_section_area	Calculate the cross-sectional area of a rectangular prism.	Computational Tools
calculate_denominator	Return the denominator (plate separation d).	Computational Tools

Tool Name	Description	Category
compute_length_change	Calculate the change in length.	Computational Tools
round_to_significant_figures	Round a number to a specified number of significant figures.	Computational Tools
symbolic_gamma_mgf_derivation	Derive the gamma distribution's MGF symbolically using sympy.	Computational Tools
round_value	Round a value to a specified number of decimal places.	Computational Tools
calculate_diameter_from_radius	Calculate the diameter of a sphere given its radius.	Computational Tools
calculate_atoms_total_volume	Calculate total volume of all atoms in the unit cell.	Computational Tools
calculate_total_volume	Calculate total volume from component volumes.	Computational Tools
calculate_volume_fraction	Calculate volume fraction of a component.	Computational Tools
calculate_outer_volume	Calculate the volume of a sphere given its radius.	Computational Tools
generate_sphere_mesh	Generate mesh coordinates for a sphere surface.	Computational Tools
calculate_cross_sectional_area	Calculate cross-sectional area.	Computational Tools
calculate_increase_factor	Calculate the increase factor of K-points.	Computational Tools
calculate_percentage_decrease	Calculate the percentage decrease based on absolute decrease and initial value.	Computational Tools
convert_degrees_to_radians	Convert angle from degrees to radians.	Computational Tools
convert_percentage_to_ratio	Convert a percentage to a decimal ratio.	Computational Tools
compute_ratio	Compute the ratio of numerator to denominator.	Computational Tools
calculate_cylinder_volume	Calculate the volume of a hollow cylinder.	Computational Tools
calculate_c_analytically	Compute the constant c analytically so that the integral of f(x) = cx^2 over [0,1] equals 1.	Computational Tools
convert_percentage_to_decimal	Convert a percentage value to decimal.	Computational Tools
calculate_cosine	Calculate the cosine of an angle in radians.	Computational Tools
calculate_relative_change	Calculate the ratio of initial to final values.	Computational Tools
calculate_sine	Calculate the sine of an angle in radians.	Computational Tools
calculate_height_from_length_and_sine	Calculate the height of the ramp.	Computational Tools
round_to_nearest_tenth	Round a number to the nearest tenth.	Computational Tools
calculate_absolute_difference	Calculate the absolute difference between two numbers.	Computational Tools
calculate_bags_needed	Calculate the number of bags needed to cover an area.	Computational Tools

Tool Name	Description	Category
calculate_total_area	Calculate the total area of a rectangular region.	Computational Tools
calculate_grass_area	Calculate the area to be seeded with grass.	Computational Tools
calculate_rectangle_area	Calculate the area of a rectangle.	Computational Tools
calculate_unit_cell_volume	Calculate the volume of a unit cell.	Computational Tools
convert_to_percentage	Convert a ratio to a percentage.	Computational Tools
calculate_percentage_increase	Calculate the percentage increase from original_value to new_value.	Computational Tools
calculate_max_capacity	Calculate maximum number of items that can fit into the storage space.	Computational Tools
calculate_max_storage_days	Calculate maximum storage days based on volume and space utilization.	Computational Tools
calculate_face_diagonal	Calculate the face diagonal length of the cubic cell.	Computational Tools
calculate_atom_center_distance	Calculate the distance between neighboring atom centers.	Computational Tools
express_original_expression	Return the original algebraic expression string for the difference in tank volumes.	Computational Tools
factorize_volume_difference	Factorize the formula for the volume difference.	Computational Tools
validate_factorization	Validate the correctness of the factorization.	Computational Tools
parameter_curve_z	Calculate z-coordinate of the curve at parameter t.	Computational Tools
derivative_z	Calculate dz/dt at parameter t.	Computational Tools
compute_analytical_probability	Calculate the analytical probability that X < value.	Computational Tools
degrees_to_radians	Convert angle from degrees to radians.	Computational Tools
calculate_sqrt_grain_size	Calculate the square root of the grain size.	Computational Tools
calculate_area_of_square	Calculate the area of a square.	Computational Tools
calculate_minimum_velocity_at_top	Calculate the minimum velocity at the top of the track.	Computational Tools
format_volume_output	Format the volume to a string with two decimal places.	Computational Tools
calculate_orbital_L	Estimate total orbital angular momentum L, simplified.	Computational Tools
convert_set_to_sorted_list	Convert a set to a sorted list.	Computational Tools
calculate_surface_area_cube	Calculate the surface area of a cube-shaped warehouse.	Computational Tools
validate_non_zero_sum_square_s	Validate that the sum of squared Miller indices is not zero.	Computational Tools
compute_sum_of_squares	Compute the sum of squares of Miller indices.	Computational Tools

Tool Name	Description	Category
symbolic_expectation	Calculates the symbolic expectation $E[X]$ of the distribution using sympy.	Computational Tools
symbolic_second_moment	Calculates the symbolic second moment $E[X^2]$ using sympy.	Computational Tools
calculate_initial_area	Calculate the initial area of the window.	Computational Tools
convert_diameter_to_radius_m	Convert diameter (millimeter) to radius (meter).	Computational Tools
convert_diameter_to_radius	Convert wire diameter to radius.	Computational Tools
calculate_length_contraction	Calculate the length contraction due to thermal contraction.	Computational Tools
calculate_final_volume	Calculate the final volume after volume reduction.	Computational Tools
calculate_manufacturing_overhead	Calculate total manufacturing overhead costs.	Computational Tools
calculate_direct_manufacturing_costs	Calculate direct manufacturing costs.	Computational Tools
calculate_total_manufacturing_costs	Calculate total manufacturing costs.	Computational Tools
calculate_non_manufacturing_costs	Calculate non-manufacturing costs.	Computational Tools
calculate_total_costs	Calculate total costs.	Computational Tools
calculate_gross_profit	Calculate gross profit.	Computational Tools
calculate_net_profit	Calculate net profit.	Computational Tools
calculate_total_input_cost	Calculate total input cost from labor, material, and overhead costs.	Computational Tools
calculate_productivity_ratio	Calculate productivity ratio as output value divided by total input cost.	Computational Tools
calculate_explicit_costs	Calculate explicit costs.	Computational Tools
calculate_opportunity_costs	Calculate opportunity costs.	Computational Tools
calculate_accounting_profit	Calculate accounting profit.	Computational Tools
calculate_economic_profit	Calculate economic profit.	Computational Tools
calculate_total_units	Calculate total units to account for.	Computational Tools
calculate_completed_units	Calculate units completed and transferred out.	Computational Tools
calculate_indirect_materials	Calculate the indirect materials cost.	Computational Tools
calculate_indirect_labor	Calculate the indirect labor cost.	Computational Tools
calculate_total_direct_cost	Calculate total direct costs.	Computational Tools
calculate_element_contribution	Calculate individual element's contribution to total magnetic moment.	Computational Tools
calculate_total_materials_used	Calculate total materials used by summing direct and indirect materials.	Computational Tools
calculate_ending_inventory	Calculate ending inventory of raw materials.	Computational Tools

Tool Name	Description	Category
calculate_absolute_error	Calculate the absolute error between measured value and true value.	Computational Tools
calculate_percentage_error	Calculate the percentage error.	Computational Tools
analyze_error_reason	Analyze possible causes of measurement error.	Computational Tools
convert_to_numpy_array	Convert input data to a NumPy array.	Computational Tools
calculate_scientific_notation	Convert a number to scientific notation with specified significant digits.	Computational Tools
format_scientific_notation	Format mantissa and exponent into a scientific notation string.	Computational Tools
compute_absolute_error	Calculate the absolute error between measured and true values.	Computational Tools
compute_absolute_error_magnitude	Calculate the magnitude (absolute value) of an error.	Computational Tools
compute_relative_error_percent	Calculate the relative error as a percentage.	Computational Tools
format_result	Round a value to a specified number of decimal places.	Computational Tools
calculate_variance_from_Sxx	Calculate sample variance from Sxx and sample size.	Computational Tools
format_output	Format output pore volume and unit.	Computational Tools
calculate_percentage_change	Calculate percentage change from relative change ratio.	Computational Tools
calculate_percent_error	Calculate the percentage error based on absolute difference and reference value.	Computational Tools
calculate_mean_square	Calculate the mean square (MS) from sum of squares (SS) and degrees of freedom (DF).	Computational Tools
calculate_f_value	Calculate the F-value for an ANOVA test.	Computational Tools
compute_f_critical_value	Compute the critical F value for given significance level and degrees of freedom.	Computational Tools
determine_significance	Determine if the F value indicates significance.	Computational Tools
calculate_max_value	Calculate the maximum integer value for a given bit resolution.	Computational Tools
convert_humidity_to_percent	Convert normalized humidity to percentage.	Computational Tools
calculate_relative_uncertainty	Calculate the relative uncertainty.	Computational Tools
calculate_combined_relative_uncertainty	Calculate the combined relative uncertainty using root-sum-square.	Computational Tools
calculate_relative_error	Calculate the relative error as a ratio.	Computational Tools

Tool Name	Description	Category
check_data_length	Check if two arrays have the same length.	Computational Tools
print_parameters	Print the input parameters with formatting for clarity.	Computational Tools
calculate_additional_space_needed	Calculate extra space needed if storage is insufficient.	Computational Tools
convert_to_scientific_notation	Convert a ratio to scientific notation with mantissa rounded to one decimal.	Computational Tools
pdf	Calculate the probability density function at point(s) x.	Computational Tools
ensure_array	Convert input to a numpy array.	Computational Tools
generate_q_range	Generate a list of integer q values from start to end inclusive.	Computational Tools
calculate_theoretical_variance	Calculate the theoretical variance of a normal distribution.	Computational Tools
calculate_median	Calculate the median of a list of numerical data.	Computational Tools
calculate_mode	Calculate the mode(s) of a list of numerical data.	Computational Tools
convert_to_set	Convert an input iterable to a set.	Computational Tools
calculate_max_quantization_error	Calculate the maximum quantization error, equal to half the LSB voltage.	Computational Tools
compute_variance_difference	Compute the difference between light and dark signal variances.	Computational Tools
compute_standard_deviation	Calculate the standard deviation from the variance difference.	Computational Tools
validate_interval_vs_total	Validate that interval individuals do not exceed total individuals.	Computational Tools
compute_relative_density	Compute interval relative density.	Computational Tools
calculate_probability	Calculate probability that device life exceeds threshold.	Computational Tools
generate_collision_time_meshgrid	Create mesh grids for electron density and temperature ranges.	Computational Tools
convert_length_mm_to_m	Convert length from millimeters to meters.	Computational Tools
convert_ml_to_dl	Convert volume from milliliters to deciliters.	Computational Tools
convert_current_mA_to_A	Convert current from milliamperes to amperes.	Computational Tools
convert_pm_to_cm	Convert length from picometers to centimeters.	Computational Tools
convert_capacitance_unit	Convert capacitance from Farads to specified units.	Computational Tools
convert_kg_to_ton	Convert weight from kilograms to tons.	Computational Tools
convert_mass_kg_to_g	Convert mass from kilograms to grams.	Computational Tools

Tool Name	Description	Category
convert_pressure_mmHg_to_atm	Convert pressure from mm mercury to atmospheres.	Computational Tools
convert_cubic_meters_to_cubic_feet	Convert volume from cubic meters to cubic feet.	Computational Tools
convert_length_to_meters	Convert length from kilometers to meters.	Computational Tools
convert_cm_to_meters	Convert length from centimeters to meters.	Computational Tools
convert_mmHg_to_atm	Convert pressure from mm Hg to atm.	Computational Tools
convert_mass_to_grams	Convert mass to grams.	Computational Tools
convert_volume_to_mL	Convert volume to milliliters.	Computational Tools
convert_thickness_nm_to_m	Convert thickness from nanometers to meters.	Computational Tools
convert_thickness_to_meters	Convert film thickness to meters.	Computational Tools
convert_area_cm2_to_m2	Convert area from square centimeters to square meters.	Computational Tools
convert_mm2_to_m2	Convert cross-sectional area from mm ² to m ² .	Computational Tools
convert_mm_to_meters	Convert length from millimeters to meters.	Computational Tools
convert_kN_to_N	Convert force from kilonewtons to newtons.	Computational Tools
convert_kg_to_g	Convert mass from kilograms to grams.	Computational Tools
convert_grain_size_to_meters	Convert grain size from micrometers to meters.	Computational Tools
convert_nm_to_m	Convert length from nanometers to meters.	Computational Tools
convert_J_to_eV	Convert energy from Joules to electron volts.	Computational Tools
convert_area_to_square_meters	Convert area to square meters based on unit system.	Computational Tools
convert_mass_g_to_kg	Convert mass from grams to kilograms.	Computational Tools
convert_volume_L_to_m3	Convert volume from liters to cubic meters.	Computational Tools
convert_time_minutes_to_seconds	Convert time from minutes to seconds.	Computational Tools
convert_nm_to_um	Convert thickness from nanometers to micrometers.	Computational Tools
convert_force_pN_to_N	Convert force from picoNewtons to Newtons.	Computational Tools
convert_distance_to_meters	Convert distance to meters if input is in centimeters.	Computational Tools
convert_length_to_mm	Convert length change from meters to millimeters.	Computational Tools
convert_velocity_kmh_to_ms	Convert velocity from kilometers per hour to meters per second.	Computational Tools
convert_amu_to_kg	Convert atomic mass units to kilograms.	Computational Tools
convert_liters_to_milliliters	Convert volume from liters to milliliters.	Computational Tools

Tool Name	Description	Category
convert_grams_to_kilograms	Convert mass from grams to kilograms.	Computational Tools
convert_grams_to_pounds	Convert mass from grams to pounds.	Computational Tools
convert_work_function_to_joules	Convert work function to Joules.	Computational Tools
format_density_unit	Format the density unit string.	Computational Tools
convert_torr_to_atm	Convert pressure from Torr to atm.	Computational Tools
convert_energy_to_joules	Convert energy to Joules.	Computational Tools
convert_dimensions_to_meters	Convert dimension to meters.	Computational Tools
convert_hours_to_days	Convert time in hours to days.	Computational Tools
convert_radius_to_meters	Convert radius from centimeters to meters if necessary.	Computational Tools
convert_inches_to_centimeters	Convert length from inches to centimeters.	Computational Tools
convert_meters_to_centimeters	Convert length from meters to centimeters.	Computational Tools
convert_length_m_to_mm	Convert length from meters to millimeters.	Computational Tools
convert_ml_to_m3	Convert volume from milliliters to cubic meters.	Computational Tools
convert_mH_to_H	Convert inductance from mil-lihenries to henries.	Computational Tools
convert_h2_production_rate_to_molecules_per_s	Convert H ₂ production rate from mmol/cm ² /h to molecules/cm ² /s.	Computational Tools
convert_length_units	Convert length from meters to specified units.	Computational Tools
convert_length_to_SI	Convert length to meters.	Computational Tools
convert_area_to_SI	Convert area to square meters.	Computational Tools
convert_length_from_SI	Convert length from meters to target units.	Computational Tools
convert_work_function_to_J	Convert work function from electron volts to joules.	Computational Tools
convert_force_kN_to_N	Convert force from kilonewtons to newtons.	Computational Tools
convert_liters_to_cubic_centimeters	Convert volume from liters to cubic centimeters.	Computational Tools
convert_thickness_mm_to_um	Convert thickness from millimeters to micrometers.	Computational Tools
convert_modulus_to_mpa	Convert elastic modulus from GPa to MPa.	Computational Tools
convert_energy_units	Convert energy units based on a conversion factor.	Computational Tools
convert_speed_kmh_to_ms	Convert speed from km/h to m/s.	Computational Tools
convert_thickness_km_to_m	Convert thickness from kilometers to meters.	Computational Tools
convert_concentration_cm3_to_m3	Convert carrier concentration from cm ⁻³ to m ⁻³ .	Computational Tools
convert_mobility_cm2Vs_to_m2Vs	Convert mobility from cm ² /(V·s) to m ² /(V·s).	Computational Tools

Tool Name	Description	Category
convert_volume_to_cm3	Convert volume to cubic centimeters based on the input unit.	Computational Tools
convert_density_to_SI	Convert density from g/cm ³ to g/m ³ .	Computational Tools
convert_volume_liters_to_cm3	Convert volume from liters to cubic centimeters.	Computational Tools
convert_meters_to_micrometer_s	Convert length from meters to micrometers.	Computational Tools
convert_gallons_to_liters	Convert gallons to liters.	Computational Tools
convert_milliliters_to_cubic_ce_ntimeters	Convert milliliters to cubic centimeters.	Computational Tools
convert_activity_uCi_to_Bq	Convert microcuries (μ Ci) to becquerel (Bq).	Computational Tools
calculate_max_power_kw	Convert maximum power from watts to kilowatts.	Computational Tools
convert_ev_to_joules	Convert energy from electron volts to joules.	Computational Tools
convert_radius_m_to_cm	Convert radius from meters to centimeters.	Computational Tools
convert_distance_mm_to_m	Convert distance from mm to m.	Computational Tools
convert_pressure_kpa_to_pa	Convert pressure from kPa to Pa.	Computational Tools
convert_bulk_modulus_kpa_to_pa	Convert bulk modulus from kPa to Pa.	Computational Tools
convert_stress_pa_to_mpa	Convert stress from pascal (Pa) to megapascal (MPa).	Computational Tools
convert_speed_to_mm_per_s	Convert welding speed from m/min to mm/s.	Computational Tools
convert_cm_to_mm	Convert length from centimeters to millimeters, rounded to 2 decimal places.	Computational Tools
get_physical_constants	Return a dictionary of fundamental physical constants.	Computational Tools
ChemicalStructureAnalyzer	Complete structure analysis from compound name (SMILES + properties).	Computational Tools
MolecularWeightCalculator	Calculate molecular weight from compound name.	Computational Tools
FunctionalGroupAnalyzer	Identify all functional groups in a molecule.	Computational Tools
MolecularDescriptorCalculator	Calculate comprehensive molecular descriptors.	Computational Tools
LipinskiRuleChecker	Check Lipinski's Rule of Five for drug-likeness.	Computational Tools
MolecularSimilarityComparator	Compare two molecules for similarity using Tanimoto coefficient.	Computational Tools
FingerprintGenerator	Generate multiple types of molecular fingerprints.	Computational Tools
StereochemistryAnalyzer	Analyze stereochemical properties.	Computational Tools
RingSystemAnalyzer	Analyze ring systems in molecules.	Computational Tools
Conformation3DAnalyzer	Analyze 3D conformational properties.	Computational Tools

Tool Name	Description	Category
ElectronicPropertyCalculator	Calculate electronic properties of molecules.	Computational Tools
MolecularShapeDescriptor	Calculate molecular shape descriptors.	Computational Tools
ChiralityAnalyzer	Comprehensive chirality analysis.	Computational Tools
BondAnalyzer	Analyze bond properties in molecules.	Computational Tools
HeteroatomAnalyzer	Analyze heteroatom content.	Computational Tools
FormalChargeCalculator	Calculate formal charges.	Computational Tools
FragmentationAnalyzer	Analyze molecular fragments.	Computational Tools
TopologicalIndexCalculator	Calculate topological indices.	Computational Tools
KappaIndicesCalculator	Calculate molecular kappa shape indices.	Computational Tools
AromaticityAnalyzer	Analyze aromatic systems.	Computational Tools
StructureFormatConverter	Convert between molecular structure formats.	Computational Tools
NameToAllFormats	Convert compound name to all structure formats.	Computational Tools
InChIKeyResolver	Resolve InChIKey to other formats.	Computational Tools
SELFIESConverter	Bidirectional SELFIES conversion.	Computational Tools
StructureValidator	Validate and standardize molecular structures.	Computational Tools
TautomerGenerator	Generate tautomers of a molecule.	Computational Tools
InChIVValidator	Validate InChI and InChIKey strings.	Computational Tools
MoleculeStandardizer	Standardize molecular structures.	Computational Tools
SubstructureSearcher	Search for substructures in molecules.	Computational Tools
ProteinPropertyCalculator	Calculate comprehensive protein properties.	Computational Tools
ProteinStabilityAnalyzer	Analyze protein stability indicators.	Computational Tools
SequenceAlignmentAnalyzer	Perform sequence alignment analysis.	Computational Tools
ProteinMotifFinder	Find common protein motifs.	Computational Tools
ProteinSolubilityPredictor	Predict protein solubility.	Computational Tools
AntibodyAnalyzer	Analyze antibody sequence features.	Computational Tools
ProteinLocalizationPredictor	Predict subcellular localization.	Computational Tools
AminoAcidCompositionAnalyzer	Analyze amino acid composition.	Computational Tools
ProteinInteractionPredictor	Predict protein-protein interaction potential.	Computational Tools
DrugTargetInteractionPredictor	Predict drug-target interaction.	Computational Tools
DNASequenceAnalyzer	Comprehensive DNA sequence analysis.	Computational Tools
GeneticCodeTranslator	Translate DNA to protein.	Computational Tools

Tool Name	Description	Category
DNAComplementFinder	Find DNA complement and reverse complement.	Computational Tools
PalindromeFinder	Find palindromic sequences.	Computational Tools
CodonOptimizer	Optimize codon usage.	Computational Tools
DNARNACodonOptimizer	Optimize DNA/RNA codons.	Computational Tools
PCRPrimerDesigner	Design PCR primers.	Computational Tools
RestrictionSiteAnalyzer	Analyze restriction enzyme sites.	Computational Tools
CircularDNAAnalyzer	Analyze circular DNA.	Computational Tools
RandomDNAGenerator	Generate random DNA sequence.	Computational Tools
PeptidePropertyCalculator	Calculate peptide properties.	Computational Tools
AlanineScanningDesigner	Design alanine scanning library.	Computational Tools
TruncationLibraryDesigner	Design truncation library.	Computational Tools
OverlapPeptideDesigner	Design overlapping peptide library.	Computational Tools
PositionalScanningDesigner	Design positional scanning library.	Computational Tools
ProteaseDigestionAnalyzer	Analyze protease digestion patterns.	Computational Tools
DegenerateCodonCalculator	Calculate degenerate codons.	Computational Tools
OligonucleotideCalculator	Calculate oligonucleotide properties.	Computational Tools
PeptideToSMILESConverter	Convert peptide sequence to SMILES.	Computational Tools
ComputeAffinityCalculator	Compute binding affinity.	Computational Tools
ComprehensiveADMETPredictor	Predict ADMET properties for compounds.	Computational Tools
DrugLikenessAnalyzer	Analyze drug-likeness properties.	Computational Tools
BBBPenetrancePredictor	Predict blood-brain barrier penetrance.	Computational Tools
BioavailabilityPredictor	Predict oral bioavailability.	Computational Tools
CIDToPropertiesConverter	Convert CID to comprehensive properties.	Computational Tools
MolecularOptimizer	Optimize molecule for drug-likeness.	Computational Tools
LeadOptimizationAnalyzer	Analyze lead compound for optimization.	Computational Tools
ScaffoldAnalyzer	Analyze molecular scaffold.	Computational Tools
RandomMoleculeGenerator	Generate random molecules.	Computational Tools
TautomerEnumerator	Enumerate all tautomers.	Computational Tools
ProteinLigandInteractionPredictor	Predict protein-ligand interactions.	Computational Tools
TargetProteinProfiler	Complete target protein profile.	Computational Tools
DrugTargetValidator	Validate drug-target pair.	Computational Tools
SmallMoleculeAffinityCalculator	Calculate small molecule similarity.	Computational Tools
MOFStructureAnalyzer	Analyze MOF structure.	Computational Tools
MaterialDensityCalculator	Calculate material density.	Computational Tools
MaterialSymmetryAnalyzer	Analyze material symmetry.	Computational Tools
MaterialCompositionAnalyzer	Analyze element composition.	Computational Tools

Tool Name	Description	Category
MOFToCompoundConverter	Convert MOF to SMILES.	Computational Tools
MaterialLatticeAnalyzer	Analyze MOF lattice parameters.	Computational Tools
ComprehensiveMaterialAnalyzer	Complete material analysis.	Computational Tools
SMILESToCASConverter	Convert SMILES to CAS number.	Computational Tools
SmallMoleculeToProteinInteraction	Analyze small molecule-protein interaction.	Computational Tools
PeptideDrugDesigner	Design peptide-based drugs.	Computational Tools
BioactiveCompoundScreener	Screen bioactive compounds.	Computational Tools
ProteinSMILESConverter	Convert peptide to SMILES and analyze.	Computational Tools
DrugPeptideOptimizer	Optimize peptide for drug properties.	Computational Tools
CompoundToADMETProfile	Complete compound AD-MET profile.	Computational Tools
DrugCandidateScreener	Screen drug candidates.	Computational Tools
LeadCompoundProfiler	Profile lead compound.	Computational Tools
CompoundDrugLikenessScorer	Score compound drug-likeness.	Computational Tools
DrugSimilarityAnalyzer	Analyze drug similarity.	Computational Tools
ProteinTargetDrugDesigner	Design drugs for protein target.	Computational Tools
AntibodyDrugConjugateDesigner	Design antibody-drug conjugates.	Computational Tools
BiologicDrugAnalyzer	Analyze biologic drugs.	Computational Tools
TargetSequenceValidator	Validate target sequence for druggability.	Computational Tools
ProteinDrugInteractionProfiler	Profile protein-drug interactions.	Computational Tools
MOFCompoundAnalyzer	Analyze MOF as compound.	Computational Tools
MaterialCompoundConverter	Convert material to compound format.	Computational Tools
MOFPropertiesCalculator	Calculate MOF properties.	Computational Tools
CrystalStructureAnalyzer	Analyze crystal structure.	Computational Tools
MaterialDrugDeliveryAnalyzer	Analyze material for drug delivery.	Computational Tools
ComputeProtPara	Compute various physical and chemical parameters for a given protein sequence using Expasy ProtParam API.	Computational Tools
ComputeProtScale	Predict the hydrophilicity of a protein sequence.	Computational Tools
ComputeExtinctionCoefficient	This tool compute the molar extinction coefficient and protein concentration of the protein, and also provides information such as the protein isoelectric point.	Computational Tools
ComputePiMw	Compute the theoretical isoelectric point (pI) and molecular weight (mW) of a protein sequence.	Computational Tools

Tool Name	Description	Category
CipherOptimizer	Codon Optimization Tool: Used to optimize codons for expression of recombinant genes in mainstream hosts.	Computational Tools
CalculatorPeptideProperty	Peptide Property Calculator: Calculate molecular weight, extinction coefficient, net peptide charge, peptide isoelectric point, and average hydrophobicity (GRAVY) of peptide properties.	Computational Tools
CalculatorOligonucleotide	Oligonucleotide (primer) Calculator: The annealing temperature (Tm), molecular weight (MW), extinction coefficient (OD/ μ mol, μ g/OD) of the oligonucleotides were calculated.	Computational Tools
ProteinCodonOptimization	This tool optimize the expression of recombinant gene condons of the protein, Input: protein: protein sequence Returns: str: The Markdown content with new sequence.	Computational Tools
DNARNACodonOptimization	This tool optimize the expression of recombinant gene condons of the DNA/RNA, Input: protein: DNA or RNA sequence.	Computational Tools
ComputeHydrophilicity	This tool compute the hydrophilicity of the protein,	Computational Tools
ComputeAnnealingTemperature	This tool compute the annealing temperature of an oligonucleotide.	Computational Tools
ConvertingPeptide2SMILES	This tool translate the polypeptide sequence to SMILES.	Computational Tools
ProteinIsoelectricPointCalculator	This tool compute the isoelectric point of the protein or peptide.	Computational Tools
ComputeAffinity	This tool compute affinity based on the molar Gibbs free energy	Computational Tools
PeptideWeightCalculator	This tool compute the Average molecular weight of the polypeptide.	Computational Tools
PeptideFormulaCalculator	This tool compute the chemical formula of the polypeptide.	Computational Tools
DegenerateCodonCalculatorby AminoAcid	This tool calculates the optimal degenerate codons that encode one or more input amino acids.	Computational Tools
OverlapPeptideLibraryDesign	This tool design overlapping peptide library.	Computational Tools

Tool Name	Description	Category
AlanineScanningLibraryDesign	This tool design peptide library.	Computational Tools
TruncationLibraryDesign	This tool design truncation peptide library.	Computational Tools
PositionalScanningLibraryDesign	This tool design positional scanning peptide library.	Computational Tools
ProteaseDigestion	This tool can simulate the hydrolytic behavior of protein-degrading enzymes.	Computational Tools
CDRLabelingAntibody	This tool label the variable regions of antibodies with CDR and FR regions; Users need to choose a numbering system, and the numbering schemes include: imgt, chothia, kabat, martin; The definition scheme includes chothia, kabat, imgt, and contact.	Computational Tools
AntibodySequenceNumbering	This tool number the amino acid sequence of the antibody; Identify the input sequence and distinguish between immunoglobulin (IG) and T cell receptor (TR); The numbering system includes: IMGT, Chothia, Kabat, Martin (extended version Chothia), and AHo; TR sequences can only be numbered using IMGT or AHo.	Computational Tools
CircularDNAAlignment	This tool aligns circular DNA sequences.	Computational Tools
CompareSequenceByLogExpectation	Multiple Sequence Comparison by Log Expectation is a tool used to compare protein or nucleic acid sequences.	Computational Tools
DoubleSequenceGlobalAlignment	This tool compares two sequences in global alignment style.	Computational Tools
InherentDisorderedRegionsPredictor	This tool predict the inherent disordered regions of the protein based on sequence.	Computational Tools
DoubleSequenceLocalAlignment	This tool compare two sequences in local alignment style.	Computational Tools
ProteinMotifAnalysis	This tool analyse the motif of the protein based on sequence.	Computational Tools
SequenceSimilarityCalculator	Sequence similarity calculation takes a set of aligned sequences (FASTA or GCG format) as input to calculate their similarity	Computational Tools

Tool Name	Description	Category
ORFFind	The ORF search tool can help you find open reading frames in DNA sequences, and the returned results include the start and end positions of the ORF as well as the translation results of the open reading frames.	Computational Tools
TranslateDNAtoAminoAcidSequence	This tool translate DNA sequence to protein(amino acid) sequence.	Computational Tools
RepeatDNASequenceSearch	This tool search repeat DNA sequence in DNA sequence.	Computational Tools
RepeatProteinSequenceSearch	This tool search repeat protein sequence in DNA sequence.	Computational Tools
PalindromicSequencesFinder	This tool searches for palindrome sequences in the sequence and enters the length range of nucleic acid sequences and palindrome sequences in the text box below.	Computational Tools
CalculateAminoAcidbyDegenerateCodon	This tool calculate amino acid by degenerate codon.	Computational Tools
ProteinNuclearLocalizationSequencePrediction	This tool predict nuclear localization sequence of protein based on sequence.	Computational Tools
SmallMoleculeSimilarityCalculation	This tool calculate the similarity of small molecules based on SMILES.	Computational Tools
DNA Molecular Weight Calculator	This tool calculate the molecular weight of DNA based on sequence.	Computational Tools
CpGIslandPrediction	The CpG island prediction tool can predict potential CpG islands using the Gardiner Garden and Frommer (1987) method. The calculation method is to use a 200bp window, with each shift of 1 bp.	Computational Tools
PCRPrimerProperties	This tool calculate the properties of PCR primer based on sequence.	Computational Tools
AminoAcidStatistics	This tool count the number of amino acids in the protein.	Computational Tools
SummaryEnzymeCleavageSites	The enzyme digestion site summary tool counts the number and location of commonly used restriction endonucleating recognition sites in DNA sequences.	Computational Tools
RandomDNAGeneration	This tool generate random DNA sequence.	Computational Tools

Tool Name	Description	Category
CalculateMolecularWeight	Calculates the molecular weight of a structure in a PDB file using MDAnalysis.	Computational Tools
GetAminoAcidFrequency	Calculates the frequency of each amino acid in a protein sequence.	Computational Tools
GetReverseComplement	Generates the reverse complement of a DNA sequence.	Computational Tools
CalculateHydrophobicityAndPolarity	Calculates the hydrophobicity and polarity of a protein sequence.	Computational Tools
MOFToSMILES	Convert multiple MOF materials into SMILES representations and return the results in Markdown table format.	Computational Tools
GetStructureInfo	Reads a structure file and returns basic information about the structure.	Computational Tools
CalculateDensity	Calculates the density of a structure from a file.	Computational Tools
GetElementComposition	Returns the elemental composition of a structure from a file.	Computational Tools
CalculateSymmetry	Calculates the symmetry of a structure from a file.	Computational Tools
FuncGroups	Identify and list the functional groups in a molecule given its SMILES string.	Computational Tools
SMILESToWeight	Calculate the molecular weight of a molecule given its SMILES string.	Computational Tools
MolSimilarity	Calculate the Tanimoto similarity between two molecules given their SMILES strings.	Computational Tools
SMILESToInChI	Convert a SMILES string to an InChI string.	Computational Tools
InChIKeyToSMILES	Convert an InChIKey string to a SMILES string.	Computational Tools
InChIKeyToInChI	Convert an InChIKey string to InChI.	Computational Tools
InChIKeyToMOL	Convert an InChI string to a MOL string.	Computational Tools
IsValidInChIKey	Check if an InChIKey string is valid.	Computational Tools
InChIToSMILES	Convert an InChI string to a SMILES string.	Computational Tools
InChIToInChIKey	Convert an InChI string to an InChIKey string.	Computational Tools
InChIToCSID	Convert InChI to ChemSpider ID.	Computational Tools
SMILESToSELFIES	Translates a SMILES string into its corresponding SELFIES string.	Computational Tools
SELFIEStoSMILES	Translates a SELFIES string into its corresponding SMILES string.	Computational Tools

Tool Name	Description	Category
RandomMoelcule	Generates a random molecule.	Computational Tools
Length_SELFIES	Computes the length of a SELFIES string.	Computational Tools
Split_SELFIES	Splits a SELFIES string into its individual tokens.	Computational Tools
GetAtomPairFingerprintAsBitVect	Generate the atom pair fingerprint of a molecule as a SparseBitVect.	Computational Tools
AssignPattyTypes	Assign Patty types to the atoms of a molecule.	Computational Tools
TestMolecule	Perform a series of tests on a molecule, including sanitization, removal of hydrogens, and canonicalization check.	Computational Tools
ShowMol	Generate a molecule image from its SMILES representation and embed it directly in Markdown.	Computational Tools
TypeAtomsInMolecule	Assigns EState types to each atom in a molecule based on its SMILES representation.	Computational Tools
CalculateEstateIndices	Calculate EState indices for each atom in a molecule based on its SMILES representation.	Computational Tools
CalculateEstateVsa	Calculate EState VSA indices for a molecule based on its SMILES representation.	Computational Tools
GenerateEstateFingerprint	Generate the EState fingerprint for a molecule based on its SMILES representation.	Computational Tools
CalculateShapeSimilarity	Calculate shape similarity scores using USRCAT for a list of molecules defined by their SMILES.	Computational Tools
CalculatePmi	Calculate the normalized principal moments of inertia (NPR1 and NPR2) for a molecule.	Computational Tools
CalculateDistanceMatrix	Calculate the distance matrix for a list of molecules based on their fingerprints.	Computational Tools
ClusterMolecules	Clusters molecules based on their fingerprints and returns the clustering results in Markdown format.	Computational Tools
ProcessFingerprintMol	Process the molecular fingerprint generated by FingerprintMol function.	Computational Tools
FingerprintsFromSmiles	Generate fingerprints for a list of SMILES strings.	Computational Tools
GetRdkFingerprintFromSmiles	Generate an RDKit fingerprint from a SMILES string using default parameters.	Computational Tools

Tool Name	Description	Category
GenerateFraggleFragments	Generate all possible Fraggle fragmentations for a molecule represented by a SMILES string.	Computational Tools
CheckValidRingCut	Check if the molecule represented by a SMILES string is a valid ring cut.	Computational Tools
BuildAtomPairFpFromSmiles	Generate an Atom Pair Fingerprint from a SMILES string and display the results in a readable format.	Computational Tools
BuildTorsionsFpFromSmiles	Generate a Torsions Fingerprint from a SMILES string.	Computational Tools
BuildRdkitFpFromSmiles	Generate an RDKit fingerprint from a SMILES string.	Computational Tools
BuildPharm2DFpFromSmiles	Generate a Pharm2D fingerprint from a SMILES string.	Computational Tools
BuildMorganFpFromSmiles	Generate a Morgan fingerprint from a SMILES string.	Computational Tools
BuildAvalonFpFromSmiles	Generate an Avalon fingerprint from a SMILES string.	Computational Tools
ConvertSmilesToInchi	Converts a SMILES string to its corresponding InChI string.	Computational Tools
GenerateMolKeyFromSmiles	Generates a molecular key for a given molecule represented by a SMILES string.	Computational Tools
GetStereoCodeFromSmiles	Generates the stereo code for a given molecule represented by a SMILES string.	Computational Tools
DetermineBondOrders	The tool is used to determine the bond orders between atoms in a molecule based on their atomic coordinates.	Computational Tools
DetermineBonds	The tool is used to determine the bond orders between atoms in a molecule based on their atomic coordinates.	Computational Tools
GetPatternFingerprint	This tool is used to generate a pattern fingerprint for a molecule.	Computational Tools
IsSubstructure	This tool is used to check if a molecule(target) is a substructure of another molecule(template).	Computational Tools
GetTemplateMolecule	This tool is used to get the template molecule from a TautomerQuery object.	Computational Tools
GetTautomers	This tool obtains all possible tautomers of a TautomerQuery object.	Computational Tools
GetModifiedAtoms	This tool is used to get the modified atoms of a TautomerQuery object.	Computational Tools

Tool Name	Description	Category
GetModifiedBonds	This tool is used to get the modified bonds of a TautomerQuery object.	Computational Tools
GetSubstructMatches	This tool is to search for substructures in a given target molecule that match the tautomer query.	Computational Tools
CanSerialize	This tool is used to check if a TautomerQuery object can be serialized.	Computational Tools
AssignCIPLabels	This tool is used to assign CIP labels to the atoms in a molecule.	Computational Tools
Enumerate	The rdkit.	Computational Tools
Deprotect	The rdkit.	Computational Tools
CondenseAbbreviationSubstanceGroups	This tool finds and replaces abbreviation substance groups in a molecule, resulting in a compressed version of the molecule where the abbreviations are expanded.	Computational Tools
SLnToSmiles	This tool is used to convert a SLN string to a SMILES string.	Computational Tools
CreateShingling	This tool is used to create a shingling for a molecule.	Computational Tools
EncodeMolecule	This tool creates an MHFP vector from a molecule using MHFP encoder, capturing structural information of the molecule.	Computational Tools
EncodeSECFP	This tool creates an SECFP vector from a molecule using SECFP encoder, capturing structural information of the molecule.	Computational Tools
GetBCUT	This tool computes the 2D BCUT descriptors for a given molecule, representing mass, Gasteiger charge, Crippen logP, and Crippen MR values.	Computational Tools
GetAutocorrelation2D	This tool computes the 2D autocorrelation descriptors for a given molecule, capturing the spatial arrangement of atoms in the molecule.	Computational Tools
GetAutocorrelation3D	This tool computes the 3D autocorrelation descriptors for a given molecule, capturing the spatial arrangement of atoms in the molecule.	Computational Tools

Tool Name	Description	Category
GetAsphericity	This tool calculates the asphericity descriptor for a molecule, which measures how much the molecule deviates from a perfectly spherical shape.	Computational Tools
GetChi0n	This tool calculates the chi^0 (chi-zero) cluster index, which represents a topological descriptor related to molecular branching.	Computational Tools
GetChi0v	This function calculates the Chi^0v (Chi-zero-v) valence molecular graph index for a molecule, which is used to describe the topology of the molecule.	Computational Tools
GetChi1n	This tool calculates the chi^1 (chi-one) cluster index, which represents a topological descriptor related to molecular branching.	Computational Tools
GetChi1v	This function calculates the Chi^1v (Chi-one-v) valence molecular graph index for a molecule, which is used to describe the topology of the molecule.	Computational Tools
GetChi2n	This tool calculates the chi^2 (chi-two) cluster index, which represents a topological descriptor related to molecular branching.	Computational Tools
GetChi2v	This function calculates the Chi^2v (Chi-two-v) valence molecular graph index for a molecule, which is used to describe the topology of the molecule.	Computational Tools
GetChi3n	This tool calculates the chi^3 (chi-three) cluster index, which represents a topological descriptor related to molecular branching.	Computational Tools
GetChi3v	This function calculates the Chi^3v (Chi-three-v) valence molecular graph index for a molecule, which is used to describe the topology of the molecule.	Computational Tools
GetChi4n	This tool calculates the chi^4 (chi-four) cluster index, which represents a topological descriptor related to molecular branching.	Computational Tools

Tool Name	Description	Category
GetChi4v	This function calculates the Chi^4v (Chi-four-v) valence molecular graph index for a molecule, which is used to describe the topology of the molecule.	Computational Tools
GetCoulombMat	This tool calculates the Coulomb matrix for a molecule, which represents the electrostatic interactions between atoms in the molecule.	Computational Tools
GetCrippenDescriptors	This function calculates the Wildman-Crippen logP and MR (molecular refractivity) values for a given molecule in RDKit.	Computational Tools
GetEEMCharges	This function computes the EEM (Electronegativity Equalization Method) atomic partial charges for a given molecule using its atomic properties.	Computational Tools
GetEccentricity	This function calculates the eccentricity of a molecule, which is a measure of its shape.	Computational Tools
GetExactMolceularWeight	This function calculates the exact molecular weight of a molecule, which is the sum of the atomic weights of all atoms in the molecule.	Computational Tools
GetFractionCSP3	This function calculates the fraction of sp3-hybridized carbon atoms in a molecule, which is a measure of its shape.	Computational Tools
GetGETAWAY	This function calculates the GETAWAY descriptors for a molecule, which capture the shape and size of the molecule.	Computational Tools
GetHallKierAlpha	This function calculates the Hall-Kier alpha index for a molecule, which is a measure of its shape.	Computational Tools
GetInertialShapeFactor	This function calculates the Inertial Shape Factor of a molecule, which is a measure of its shape.	Computational Tools
GetKappa1	This function computes the Kappa1 (K_1) value of a molecule, which is a topological descriptor representing its shape complexity or branching degree.	Computational Tools

Tool Name	Description	Category
GetKappa2	This function computes the Kappa2 (K_2) value of a molecule, which is a topological descriptor representing its shape complexity or branching degree.	Computational Tools
GetKappa3	This function computes the Kappa3 (K_3) value of a molecule, which is a topological descriptor representing its shape complexity or branching degree.	Computational Tools
GetLabuteASA	This function calculates the Labute accessible surface area (ASA) value for a molecule, which is a measure of the solvent-accessible surface area of the molecule.	Computational Tools
GetMolFormula	This function calculates the molecular formula of a molecule, which is a string representing the number and type of atoms in the molecule.	Computational Tools
GetMORSE	This tool calculates the Molecule Representation of Structures based on Electron diffraction (MORSE) descriptors for a given molecule.	Computational Tools
GetNPR1	This function calculates the NPR1 (Normalized Principal Moments Ratio) descriptor for a molecule, which serves as a descriptor for the distribution of charges within the molecule.	Computational Tools
GetNPR2	This function calculates the NPR2 (Normalized Principal Moments Ratio) descriptor for a molecule, which serves as a descriptor for the distribution of charges within the molecule.	Computational Tools
GetAliphaticCarbocyclesNum	This function calculates the number of aliphatic carbocycles in a molecule.	Computational Tools
GetAliphaticHeterocyclesNum	This function calculates the number of aliphatic heterocycles in a molecule.	Computational Tools
GetAliphaticRingsNum	This tool calculates the number of aliphatic rings in a molecule.	Computational Tools
GetAmideBondsNum	This function calculates the number of amide bonds in a molecule.	Computational Tools

Tool Name	Description	Category
GetAromaticCarbocyclesNum	This function calculates the number of aromatic carbocycles in a molecule.	Computational Tools
GetAromaticHeterocyclesNum	This function calculates the number of aromatic heterocycles in a molecule.	Computational Tools
GetAromaticRingsNum	This tool calculates the number of aromatic rings in a molecule.	Computational Tools
GetAtomStereoCentersNum	This function calculates the number of atom stereo centers in a molecule.	Computational Tools
GetAtomsNum	This function calculates the number of atoms in a molecule.	Computational Tools
GetBridgeheadAtomsNum	This function calculates the number of bridgehead atoms in a molecule.	Computational Tools
GetHBANum	This function calculates the number of hydrogen bond acceptors (HBA) in a molecule.	Computational Tools
GetHBDNum	This function calculates the number of hydrogen bond donors (HBD) in a molecule.	Computational Tools
GetHeavyAtomsNum	This tool calculates the number of heavy atoms in a molecule.	Computational Tools
GetHeteroatomsNum	This tool calculates the number of heteroatoms in a molecule.	Computational Tools
GetHeterocyclesNum	This tool calculates the number of heterocycles in a molecule.	Computational Tools
GetLipinskiHBANum	This tool calculates the number of Lipinski hydrogen bond acceptors (HBA) in a molecule, which is a measure used in drug-likeness evaluation according to Lipinski's rule of five.	Computational Tools
GetLipinskiHBDNum	This tool calculates the number of Lipinski hydrogen bond donors (HBD) in a molecule, which is a measure used in drug-likeness evaluation according to Lipinski's rule of five.	Computational Tools
GetRingsNum	This tool calculates the number of rings in a molecule.	Computational Tools
GetRotatableBondsNum	This tool calculates the number of rotatable bonds in a molecule.	Computational Tools
GetSaturatedCarbocyclesNum	This function calculates the number of saturated carbocycles in a molecule.	Computational Tools

Tool Name	Description	Category
GetSaturatedHeterocyclesNum	This function calculates the number of saturated heterocycles in a molecule.	Computational Tools
GetSaturatedRingsNum	This tool calculates the number of saturated rings in a molecule.	Computational Tools
GetSpiroAtomsNum	This function calculates the number of spiro atoms in a molecule.	Computational Tools
GetUnspecifiedAtomStereoCentersNum	This tool calculates the number of unspecified atomic stereocenters in a molecule.	Computational Tools
GenerateRDKFingerprintsFromCSV	Generate RDKfingerprints for the SMILES strings in a CSV file and save to a new CSV file.	Computational Tools
GenerateMorganfingerprintsFromCSV	Generate morgan fingerprints for the SMILES strings in a CSV file and save to a new CSV file.	Computational Tools
GenerateElectricalDescriptorsFromCSV	Generate electrical RDKit descriptors for the SMILES strings in a CSV file and save to a new CSV file.	Computational Tools
MLPClassifier	General MLP classifier function that predicts based on processed feature files.	Computational Tools
AdaBoostClassifier	General AdaBoost classifier function that predicts based on processed feature files.	Computational Tools
RandomForestClassifier	General Random Forest classifier function that predicts based on processed feature files.	Computational Tools
AssignOxidationNumbers	Adds the oxidation number/state to the atoms of a molecule as property OxidationNumber on each atom.	Computational Tools
CalculatePBF	This tool calculates the PBF (plane of best fit) descriptor for a given molecule.	Computational Tools
CalculatePMI1	This tool calculates the first principal moment of inertia (PMI1) for a given molecule.	Computational Tools
CalculatePMI2	This tool is designed to compute the PMI2 (Partial Molecular Information 2) value of a molecule, which serves as a descriptor indicating the shape and structure of the molecule.	Computational Tools

Tool Name	Description	Category
CalculatePMI3	This tool is designed to compute the PMI3 (Partial Molecular Information 3) value of a molecule, which serves as a descriptor characterizing the shape and structure of the molecule.	Computational Tools
CalculatePhi	This tool calculates the Phi (Φ) angle of a molecule, which is a torsional angle describing the rotation about a single bond.	Computational Tools
CalculateRDF	This tool calculates the RDF (Radial Distribution Function) descriptor for a given molecule.	Computational Tools
CalculateRadiusOfGyration	This tool is designed to compute the radius of gyration for a given molecule, providing insights into its overall shape and compactness.	Computational Tools
CalculateSpherocityIndex	This function calculates the sphericity index for a given molecule.	Computational Tools
CalculateTPSA	This tool calculates the TPSA (Topological Polar Surface Area) descriptor for a given molecule, which is a measure of the accessible polar surface area in a molecule.	Computational Tools
CalculateWHIM	This tool calculates the WHIM (Weighted Holistic Invariant Molecular) descriptor for a given molecule.	Computational Tools
CustomPropertyVSA	This function computes a custom property for a given molecule using the Van der Waals Surface Area (VSA) method, based on user-defined parameters.	Computational Tools
GetAtomFeature	This function computes a set of atom features for a given molecule, including atomic number, valence, and hybridization.	Computational Tools
GetAtomPairFingerprint	This function computes the atom pair for a given molecule.	Computational Tools
GetConnectivityInvariants	This tool computes connectivity invariants, similar to ECFP (Extended Connectivity Fingerprints), for a given molecule.	Computational Tools

Tool Name	Description	Category
GetFeatureInvariants	This tool computes feature invariants, similar to FCFP (Feature Centroid Fingerprints), for a given molecule.	Computational Tools
GetAtomPairCode	This function computes atom pair code (hash) for each atom in a molecular.	Computational Tools
GetHybridization	This function computes the hybridization of each atom in a molecule.	Computational Tools
GetRingSystems	This function computes the ring systems of a molecule.	Computational Tools
GetMACCSKeysFingerprint	This function computes the Molecular ACCess System keys fingerprint for a given molecule.	Computational Tools
GetMorganFingerprint	This tool computes the Morgan fingerprint for a given molecule.	Computational Tools
GetTopologicalTorsionFingerprint	This tool computes the topological torsion fingerprint for a given molecule.	Computational Tools
GetUSR	The tool computes the USR (Ultrafast Shape Recognition) descriptor for a given conformer of a molecule and returns it as a list.	Computational Tools
GetUSRCAT	This function is designed to compute the USRCAT (Ultrafast Shape Recognition with Coordinate Asymmetric Torsions) descriptor for a specified conformer of a molecule.	Computational Tools
AddHydrogens	This function is used to add hydrogen atoms to the molecular graph of a molecule.	Computational Tools
AddWavyBondsForStereoAny	This tool adds wavy bonds around double bonds with STEREOANY stereochemistry.	Computational Tools
AssignAtomChiralTagsFromStructure	This tool sets chiral tags for atoms of the molecular based on the molParity property.	Computational Tools
AssignRadicals	This tool is used to assign radical counts to atoms within a molecule.	Computational Tools
AssignStereoChemistry	This tool is used for assigning Cahn–Ingold–Prelog (CIP) stereochemistry to atoms (R/S) and double bonds (Z/E) within a molecule.	Computational Tools
GetAdjacencyMatrix	This tool is used to obtain the adjacency matrix of a molecule.	Computational Tools

Tool Name	Description	Category
GetAllowNontetrahedralChirality	This tool is used to determine whether recognition of non-tetrahedral chirality from 3D structures is enabled or not.	Computational Tools
GetDistanceMatrix	The tool computes the topological distance matrix for a given molecule.	Computational Tools
GetFormalCharge	This tool is utilized to determine the total formal charge of a given molecule.	Computational Tools
GetFormalChargeOfAtoms	This tool is utilized to determine the formal charge of each atom in a given molecule.	Computational Tools
GetMolFrags	This tool identifies disconnected fragments within a molecule and returns them as atom identifiers or molecules.	Computational Tools
GetUseLegacyStereoPerception	This tool is used to determine whether the legacy stereo perception code is being used.	Computational Tools
HapticBondsToDative	This tool is used to convert a molecule that represents haptic bonds using a dummy atom with a dative bond to a metal atom into a molecule with explicit dative bonds from the atoms of the haptic group to the metal atom.	Computational Tools
HasQueryHs	This tool is used to check if a molecule contains query H (hydrogen) atoms.	Computational Tools
Kekulize	This tool is used to perform Kekulization on a molecule.	Computational Tools
MergeQueryHs	This tool is used to merge hydrogen atoms into their neighboring atoms as query atoms.	Computational Tools
MurckoDecompose	This tool is used to perform a Murcko decomposition on a molecule and return the scaffold.	Computational Tools
RemoveHydrogens	This tool is used to remove hydrogen atoms from a molecule's graph.	Computational Tools
RemoveStereochemistry	This tool is used to remove all stereochemistry information from a molecule.	Computational Tools
SetAromaticity	This tool is used to perform aromaticity perception on a molecule, which means determining the aromaticity of atoms and bonds in the molecule.	Computational Tools

Tool Name	Description	Category
SetBondStereoFromDirections	This tool is used to set the cis/trans stereochemistry on double bonds based on the directions of neighboring bonds.	Computational Tools
IsSubstructof	This tool is used to check if a molecule(target) is a substructure of another molecule(template).	Computational Tools
generate_presigned_url	Generate pre-signed URLs for uploading and downloading Alibaba Cloud OSS objects	Computational Tools
query_uniprot	Query UniProt for protein information.	Databases
query_interpro	Query InterPro for protein domain information.	Databases
download_pdb_structure	Download PDB structure file.	Databases
download_ncbi_sequence	Download NCBI sequence file.	Databases
download_alphafold_structure	Download AlphaFold structure file.	Databases
retrieve_protein_data_by_pdbc ode	Retrieve and download the protein sequence (.).	Databases
retrieve_smiles_by_compound name	Retrieve SMILES strings from PubChem using compound names.	Databases
search_uniprot_id	Search UniProt ID by gene name.	Databases
download_alphafold_structure	Download predicted protein structures from the AlphaFold database.	Databases
get_activity_by_id	Retrieve the details of a single bioactivity entry from the ChEMBL database by its unique activity ID.	Databases
get_activity_by_ids	Retrieves a list of bioactivity entries from the ChEMBL database.	Databases
search_activity	Performs a full-text search for bioactivity data in the ChEMBL database using a query string.	Databases
get_activity_supplementary_da ta_by_activity	Retrieves a default list of supplementary bioactivity data from the ChEMBL database.	Databases
get_activity_supplementary_da ta_by_activity_by_id	Retrieve single activitysupplementarydatabyactivity object details by ID.	Databases
get_activity_supplementary_da ta_by_activity_by_ids	Retrieve multiple activitysupplementarydatabyactivity objects by IDs.	Databases
get_assay_by_name	Retrieves the details for a single assay (experimental procedure) from the ChEMBL database using its name.	Databases

Tool Name	Description	Category
get_assay_by_names	Retrieves detailed information for multiple assays from the ChEMBL database using a list of their names.	Databases
search_assay	Performs a full-text search for assays (experimental procedures) in the ChEMBL database using a query string.	Databases
get_assay_class	Retrieves a default list of assay classifications from the ChEMBL database.	Databases
get_atc_class_by_level5	Retrieves the details for a single ATC (Anatomical Therapeutic Chemical) classification from the ChEMBL database using the level 5 ATC code.	Databases
get_atc_class_by_level5s	Retrieves the details for multiple ATC (Anatomical Therapeutic Chemical) classifications from the ChEMBL database using a list of their level 5 ATC codes.	Databases
get_binding_site_by_id	Retrieves the details for a single binding site from the ChEMBL database using its unique integer ID.	Databases
get_binding_site_by_ids	Retrieves detailed information for multiple binding sites from the ChEMBL database using a list of their unique integer IDs.	Databases
get_biotherapeutic_by_name	Retrieves the details for a single biotherapeutic from the ChEMBL database using its name.	Databases
get_biotherapeutic_by_names	Retrieves detailed information for multiple biotherapeutics from the ChEMBL database using a list of their names.	Databases
get_cell_line_by_id	Retrieves the details for a single cell line from the ChEMBL database using its unique integer ID.	Databases
get_cell_line_by_ids	Retrieves detailed information for multiple cell lines from the ChEMBL database using a list of their unique integer IDs.	Databases
get_compound_structural_alert	Retrieve compound structural alert object list.	Databases
get_compound_structural_alert_by_id	Retrieve compound structural alert object details by ID.	Databases

Tool Name	Description	Category
get_compound_structural_alert_by_ids	Retrieve multiple compound structural alert objects by IDs.	Databases
get_drug_by_name	Retrieve single drug object details by name.	Databases
get_drug_by_names	Retrieve multiple drug objects by names.	Databases
get_drug_indication_by_id	Retrieve drug indication object details by ID	Databases
get_drug_indication_by_ids	Retrieve multiple drug indication objects by IDs.	Databases
get_drug_warning_by_id	Retrieve single drug_warning object details by ID	Databases
get_drug_warning_by_ids	Retrieve multiple drug_warning objects by IDs.	Databases
get_go_slim_by_id	Retrieves the details for a single GO (Gene Ontology) slim classification from the ChEMBL database using its unique GO ID.	Databases
get_go_slim_by_ids	Retrieves the details for multiple GO (Gene Ontology) slim classifications from the ChEMBL database using a list of their unique GO IDs.	Databases
get_mechanism_by_id	Retrieves the details for a single drug mechanism of action from the ChEMBL database using its unique integer ID.	Databases
get_mechanism_by_ids	Retrieves detailed information for multiple drug mechanisms of action from the ChEMBL database using a list of their unique integer IDs.	Databases
get_metabolism_by_id	Retrieve single metabolism object details by ID.	Databases
get_metabolism_by_ids	Retrieves detailed information for multiple drug metabolism records from the ChEMBL database using a list of their unique integer IDs.	Databases
get_molecule_by_name	Retrieves the details for a single molecule from the ChEMBL database using its name.	Databases
get_molecule_by_names	Retrieves detailed information for multiple molecules from the ChEMBL database using a list of their names.	Databases
get_molecule_form	Retrieves a default list of molecule forms from the ChEMBL database.	Databases

Tool Name	Description	Category
get_molecule_form_by_name	Retrieves molecule form information for a given molecule from the ChEMBL database using its name.	Databases
get_molecule_form_by_names	Retrieves molecule form information for multiple molecules from the ChEMBL database using a list of their names.	Databases
get_organisation_by_id	Retrieves the details for a single organisation from the ChEMBL database using its unique integer ID.	Databases
get_organisation_by_ids	Retrieves detailed information for multiple organisations from the ChEMBL database using a list of their unique integer IDs.	Databases
get_protein_classification_by_id	Retrieves the details for a single protein classification from the ChEMBL database using its unique integer ID.	Databases
get_protein_classification_by_ids	Retrieves detailed information for multiple protein classifications from the ChEMBL database using a list of their unique integer IDs.	Databases
search_protein_classification	Search protein_classification object by query string.	Databases
get_similarity_by_smiles	Retrieve similarity details for compounds based on SMILES.	Databases
get_source_by_id	Retrieve single source object details by ID.	Databases
get_source_by_ids	Retrieve multiple source object details by IDs.	Databases
get_substructure_by_smiles	Retrieve substructure matches using SMILES.	Databases
get_target_by_name	Retrieve single target object details by name.	Databases
get_target_by_names	Retrieve multiple target objects by names	Databases
search_target	Search target using query string.	Databases
get_target_component	Retrieve target_component object list	Databases
get_target_relation_by_related_name	Retrieve single target_relation object details by related target name	Databases
get_target_relation_by_related_names	Retrieve multiple target_relation objects by related target names	Databases
get_tissue_by_id	Retrieve single tissue object details by ID.	Databases
get_tissue_by_ids	Retrieve multiple tissue object details by IDs.	Databases

Tool Name	Description	Category
get_compound_chembl_id_by_name	Get compound chembl id by name.	Databases
get_target_chembl_id_by_name	Get target chembl id by name.	Databases
get_assay_chembl_id_by_name	Get assay chembl id by name.	Databases
kegg_info	This operation displays the database release information with statistics for the databases.	Databases
kegg_find	KEGG find - Data search.	Databases
kegg_list	This operation can be used to obtain a list of all entries in each database.	Databases
kegg_get	This operation retrieves given database entries in a flat file format or in other formats with options.	Databases
kegg_conv	This operation can be used to convert entry identifiers (accession numbers) of outside databases to KEGG identifiers, and vice versa.	Databases
kegg_link	KEGG link - find related entries by using database cross-references.	Databases
mapping_identifiers	Maps common protein names, synonyms and UniProt identifiers into STRING identifiers	Databases
get_string_network_interaction	Retrieve STRING interaction network for one or multiple proteins in various text formats.	Databases
get_all_interaction_partners_of_the_protein_set	This method provides the interactions between your provided set of proteins and all the other STRING proteins.	Databases
get_similarity_scores_of_the_protein_set	STRING internally uses the Smith-Waterman bit scores as a proxy for protein homology.	Databases
get_best_similarity_hits_between_species	Retrieve the similarity from your input protein(s) to the best (most) similar protein from each STRING species.	Databases
get_functional_enrichment	STRING maps several databases onto its proteins, this includes: Gene Ontology, KEGG pathways, UniProt Keywords, PubMed publications, Pfam domains, InterPro domains, and SMART domains.	Databases

Tool Name	Description	Category
get_functional_annotation	STRING maps several databases onto its proteins, this includes: Gene Ontology, KEGG pathways, UniProt Keywords, PubMed publications, Pfam domains, InterPro domains, and SMART domains.	Databases
get_ppi_enrichment	Get protein-protein interaction enrichment for list of genes denoted by their STRING identifiers	Databases
clinvar_search	Providing Information of variants of human proteins, and whether these variants are benign, mutual or harmful.	Databases
gsea_search	Database containing gene, gene set and collection of gene sets information.	Databases
mousemine_search	A database containing the relationship between mouse gene sets and diseases.	Databases
ensemble_search	Ensemble is a database containing the relationship between human genes and chr locations.	Databases
search_pubchem_by_name	Search PubChem for compounds matching a chemical name.	Databases
search_pubchem_by_smiles	Search PubChem for compounds matching a SMILES string.	Databases
get_pubchem_compound_by_cid	Get detailed compound information by PubChem CID.	Databases
search_pubchem_advanced	Perform an advanced search on PubChem using a complex query.	Databases
get_substance_by_sid	Get substance information by PubChem SID.	Databases
get_compound_by_cid	Get compound information by PubChem CID.	Databases
get_compound_by_name	Get compound information by chemical name.	Databases
get_substance_by_name	Get substance information by name.	Databases
get_compound_property_by_name	Retrieve a specific chemical property for a compound by its name from PubChem.	Databases
get_compound_synonyms_by_name	Retrieve all known synonyms for a given compound by its chemical name from PubChem.	Databases
get_description_by_sid	Get detailed description information for a PubChem substance given its SID.	Databases

Tool Name	Description	Category
get_description_by_cid	Retrieve detailed description information for a PubChem compound given its CID.	Databases
get_general_info_by_compound_name	Get detailed description of a compound by name, including overall information, drug and medication information, pharmacology and biochemistry information.	Databases
get_description_by_aid	Retrieve detailed description information for a PubChem bioassay given its AID.	Databases
get_assay_summary_by_cid	Retrieve a summary of bioassay activities for a given PubChem compound CID.	Databases
get_assay_summary_by_sid	Retrieve a summary of bioassay activities for a given PubChem substance SID.	Databases
get_gene_summary_by_geneid	Get summary information for a gene by Gene ID.	Databases
get_protein_summary_by_accession	Retrieve summary information for a protein from PubChem by its UniProt or other accession number.	Databases
get_taxonomy_summary_by_taxonomyid	Retrieve summary information for a biological taxonomy entry given its NCBI Taxonomy ID.	Databases
get_conformers_by_cid	Retrieve available conformer identifiers for a given PubChem compound CID.	Databases
get_compounds_by_smiles	Retrieve compound objects from PubChem based on a given SMILES string.	Databases
get_compounds_by_formula	Retrieve compound objects from PubChem based on a molecular formula.	Databases
get_molecular_formula	Get the molecular formula of a compound.	Databases
get_molecular_weight	Get the molecular weight of a compound.	Databases
get_isomeric_smiles	Get the isomeric SMILES of a compound.	Databases
get_xlogp	Get the XLogP value of a compound.	Databases
get_iupac_name	Get the IUPAC name of a compound.	Databases
get_synonyms	Get the synonyms of a compound.	Databases
get_cids_by_smiles	Obtain the CID corresponding to the drug smiles	Databases
get_cids_by_formula	Get a list of CIDs by molecular formula.	Databases
get_sids_by_name	Get a list of SIDs by name.	Databases
get_substance_by_sid_pcp	Get a Substance object by SID using PubChemPy.	Databases

Tool Name	Description	Category
get_substances_by_name_pcp	Get a list of Substance objects by name using PubChemPy.	Databases
get_substances_source_id	Get the source ID (Unique identifier assigned to the compound or substance by the original database (e.	Databases
get_substances_synonyms	Get the synonyms (Different names or identifiers for the same chemical substance) of a substance by SID.	Databases
get_compound_dict	Get a dictionary of a compound's properties.	Databases
get_compounds_3d	Get a list of compound objects with 3D structures.	Databases
get_compounds_dict	Get a dictionary representation of a compound by CID.	Databases
get_substructure_cas	Get CAS Registry Numbers for compounds containing a specified substructure.	Databases
get_gene_metadata_by_gene_name	Get a gene summary by gene symbol.	Databases
get_gene_by_ids	Get gene information by gene IDs.	Databases
get_gene_by_accession	Get gene information by accession.	Databases
get_gene_by_accession_dataset_report	Get dataset reports by accession IDs	Databases
get_gene_by_accession_product_report	Get gene product reports by accession IDs	Databases
get_gene_download_summary_by_id	Get gene download summary by GeneID	Databases
get_gene_links_by_id	Get gene links by gene ID	Databases
get_gene_dataset_report_by_locus_tag	Get gene dataset reports by locus tag	Databases
get_gene_product_report_by_locus_tag	Get gene product reports by locus tags	Databases
get_gene_by_symbol_dataset_report	Get dataset reports by taxons	Databases
get_gene_by_symbol_product_report	Get product reports by taxon	Databases
get_gene_by_taxon_dataset_report	Get gene dataset reports by taxonomic identifier	Databases
get_gene_by_taxon_product_report	Get gene product reports by taxonomic identifier	Databases
get_gene_dataset_report_by_id	Get gene information by dataset report	Databases
get_genome_annotation_report	Get genome annotation reports by genome accession.	Databases
get_genome_annotation_summary	Get genome annotation report summary information.	Databases
get_genome_revision_history	Get revision history for assembly by accession	Databases
get_genome_sequence_reports	Get sequence reports by accessions	Databases
check_genome_accessions	Check the validity of genome accessions	Databases

Tool Name	Description	Category
get_genome_dataset_report_by_accession	Get dataset reports by accessions	Databases
get_genome_download	Get a genome dataset by accession	Databases
get_genome_download_summary	Preview genome dataset download	Databases
get_genome_links	Get assembly links by accessions	Databases
get_genome_dataset_report_by_assembly_name	Get dataset reports by assembly name	Databases
get_genome_dataset_report_by_bioproject	Get dataset reports by bioproject	Databases
get_genome_dataset_report_by_biosample	Get dataset reports by biosample id	Databases
get_sequence_assemblies	Get assembly accessions for a sequence accession	Databases
get_genome_dataset_report_by_taxon	Get dataset reports by taxons	Databases
get_genome_dataset_report_by_wgs	Get dataset reports by wgs accession	Databases
get_virus_annotation_report	Get virus annotation report by accessions.	Databases
check_virus_accessions	Check virus accessions validity	Databases
get_virus_dataset_report	Get virus dataset report by accessions	Databases
get_virus_by_taxon_annotation_report	Get virus annotation report by taxon	Databases
get_virus_by_taxon_genome	Get virus genome by taxon	Databases
get_virus_by_taxon_genome_table	Get virus genome table by taxon	Databases
get_taxonomy_related_ids	Get related taxonomy IDs	Databases
get_taxonomy_links	Get taxonomy links	Databases
get_taxonomy	Get taxonomy information.	Databases
get_taxonomy_dataset_report	Get taxonomy dataset report.	Databases
get_taxonomy_filtered_subtree	Get filtered taxonomy subtree	Databases
get_taxonomy_name_report	Get taxonomy name report	Databases
get_taxonomy_taxon_suggest	Get taxonomy suggestions	Databases
get_biosample_report	Get biosample report.	Databases
get_organelle_dataset_report	Get organelle dataset report.	Databases
get_organelle_by_taxon_dataset_report	Get organelle dataset report by taxon.	Databases
get_gene_product_report_by_id	Get gene product report by gene ID	Databases
get_gene_orthologs	Get gene orthologs by gene ID	Databases
get_gene_by_taxon	Get gene information by taxon	Databases
get_gene_counts_by_taxon	Get gene counts by taxon	Databases
get_chromosome_summary	Get chromosome summary by taxon and annotation name	Databases
get_genome_by_accession	Get genome information by accession	Databases

Tool Name	Description	Category
get_prokaryote_gene_dataset_by_refseq_protein_accession	Get a prokaryote gene dataset by RefSeq protein accession	Databases
get_general_info_by_protein_or_gene_name	Get general information of a protein or gene by name from UniProt database.	Databases
get_protein_sequence_by_name	Get the human protein sequence by protein name.	Databases
get_uniprotkb_entry_by_accession	Search UniProtKB by protein entry accession to return all data associated with that entry.	Databases
stream_uniprotkb_entries	Stream all UniProtKB entries associated with the search term in a single download.	Databases
search_uniprotkb_entries	Search UniProtKB entries using a query, returns paginated list.	Databases
get_uniref_cluster_by_id	Search UniRef entry by id to return all data associated with that entry.	Databases
get_uniref_cluster_members_by_id	Search UniRef entry by member id to return all data associated with that entry.	Databases
get_uniref_light_cluster_by_id	Search light UniRef entry by id to return all data associated with that entry.	Databases
stream_uniref_clusters	Stream all UniRef clusters associated with the search term in a single download.	Databases
search_uniref_clusters	Search UniRef clusters using a query, returns paginated list.	Databases
get_uniparc_entry_by_upi	Search UniParc entry by id (UPI) to return all data associated with that entry.	Databases
get_uniparc_light_entry_by_upi	Search UniParc entry by id (UPI) to return all data associated with that entry (light version).	Databases
get_uniparc_cross_references_by_upi	Get a page of database cross-reference entries by a UPI.	Databases
stream_uniparc_cross_references_by_upi	Stream database cross-reference entries for a specified UniParc UPI.	Databases
stream_uniparc_entries	Stream all UniParc entries associated with the specified search term in a single download.	Databases
search_uniparc_entries	Search UniParc entries using a query, returns paginated list.	Databases
get_gene_centric_by_accession	Retrieve a GeneCentric entry by UniProtKB accession.	Databases

Tool Name	Description	Category
get_gene_centric_by_proteome	Search GeneCentric entry by Proteome ID to return all data associated with that entry.	Databases
stream_gene_centric	Stream GeneCentric entries matching a query (max 10M entries).	Databases
search_gene_centric	Search GeneCentric entries with pagination.	Databases
get_proteome_by_id	Retrieve a proteome by UniProt Proteome ID.	Databases
stream_proteomes	Stream Proteome entries matching a query (max 10M entries).	Databases
search_proteomes	Search Proteome entries with pagination.	Databases
get_gene_specific_expression_in_cancer_type	Analyze the tissue-specific expression pattern of a given gene across different cancer types using the Firebrowse API (TCGA mRNASeq).	Databases
get_lookup_symbol	Look up Ensembl gene information by external gene symbol.	Databases
get_homology_symbol	Find evolutionary homologs (orthologs and paralogs) for a gene identified by symbol.	Databases
get_sequence_region	Retrieve genomic DNA sequence for a specific chromosomal region.	Databases
get_vep_hgvs	Predict the functional effects of variants using Variant Effect Predictor (VEP) with HGVS notation.	Databases
get_genetree_id	Retrieve a phylogenetic gene tree by its Ensembl stable identifier.	Databases
get_info_assembly	Retrieve genome assembly information for a species.	Databases
get_xrefs_symbol	Get cross references for a gene symbol.	Databases
get_archive_id	Get the latest version of an Ensembl stable identifier.	Databases
post_archive_id	Get the latest version for multiple Ensembl stable identifiers.	Databases
get_cafe_genetree_id	Retrieve a CAFE (Computational Analysis of gene Family Evolution) gene tree by ID.	Databases
get_cafe_genetree_member_symbol	Retrieve a CAFE gene tree for a gene identified by symbol.	Databases
get_cafe_genetree_member_id	Retrieve the gene tree containing a gene identified by its Ensembl ID.	Databases
get_genetree_member_symbol	Get gene tree by symbol.	Databases

Tool Name	Description	Category
get_alignment_region	Retrieve genomic alignments between species for a specific region.	Databases
get_homology_id	Find evolutionary homologs (orthologs and paralogs) for a gene identified by Ensembl ID.	Databases
get_xrefs_id	Get cross references by ID.	Databases
get_xrefs_name	Get cross references by name.	Databases
get_info_analysis	List the analyses and data processing pipelines used for a species genome.	Databases
get_assembly_region_info	Retrieve detailed information about a specific genomic region or chromosome.	Databases
get_info_biotypes	Retrieve the catalog of gene and transcript biotypes for a species.	Databases
get_info_external_dbs	Get external databases for a species.	Databases
get_map_cdna	Map cDNA coordinates to genomic coordinates.	Databases
get_map_translation	Map protein coordinates to genomic coordinates.	Databases
get_ontology_ancestors	Get ontology ancestors.	Databases
get_ontology_descendants	Get ontology descendants.	Databases
get_ontology_id	Get ontology by ID.	Databases
get_ontology_name	Get ontology by name.	Databases
get_overlap_id	Get features overlapping a region defined by an identifier.	Databases
get_overlap_region	Get features overlapping a genomic region.	Databases
get_overlap_translation	Get features overlapping a translation.	Databases
get_phenotype_region	Retrieve phenotype associations for variants in a genomic region.	Databases
get_phenotype_gene	Retrieve phenotype associations for a specific gene.	Databases
get_phenotype_accession	Retrieve genomic features associated with a specific phenotype ontology term.	Databases
get_sequence_id	Retrieve sequence associated with an Ensembl identifier.	Databases
get_vep_id	Predict the functional effects of variants using Variant Effect Predictor (VEP) with variant identifier.	Databases
get_vep_region	Predict the functional effects of variants using Variant Effect Predictor (VEP) with genomic coordinates.	Databases
get_variation	Retrieve detailed information about a genetic variant by its identifier.	Databases

Tool Name	Description	Category
get_info_genomes	Find information about a given genome.	Databases
get_info_genomes_accession	Find information about genomes containing a specified INSDC accession.	Databases
get_info_genomes_assembly	Find information about a genome with a specified assembly.	Databases
get_info_genomes_division	Find information about all genomes in a given division.	Databases
get_info_genomes_taxonomy	Find information about all genomes beneath a given node of the taxonomy.	Databases
get_info_variation	List all variation data sources used for a species in Ensembl.	Databases
get_info_variation_consequence_types	Lists all variant consequence types used by Ensembl.	Databases
get_info_variation_populations	List all variation populations for a species, or list all individuals in a specific population.	Databases
get_lookup_id	Look up details for any Ensembl stable identifier.	Databases
post_lookup_id	Look up details for multiple Ensembl stable identifiers in a single request.	Databases
post_lookup_symbol	Look up multiple gene symbols in a single request.	Databases
get_ontology_ancestors_chart	Reconstruct the entire ancestry of a term from is_a and part_of relationships.	Databases
get_taxonomy_classification	Return the taxonomic classification of a taxon node.	Databases
get_taxonomy_id	Search for a taxonomic term by its identifier or name	Databases
get_taxonomy_name	Search for a taxonomic id by a non-scientific name.	Databases
get_species_binding_matrix	Return the specified binding matrix	Databases
post_sequence_id	Request multiple types of sequence by a stable identifier list.	Databases
post_sequence_region	Get sequences by multiple regions.	Databases
post_vep_id	Batch predict the functional effects of multiple variants using VEP with variant identifiers.	Databases
post_vep_region	Get variant effect predictions by multiple regions.	Databases
post_variant_recoder	Translate a list of variant identifiers, HGVS notations or genomic SPDI notations to all possible variant IDs, HGVS and genomic SPDI	Databases

Tool Name	Description	Category
get_variation_pmcid	Fetch variants by publication using PubMed Central reference number (PMCID)	Databases
get_variation_pmids	Fetch variants by publication using PubMed reference number (PMID)	Databases
post_variation	Uses a list of variant identifiers (e.	Databases
get_ga4gh_beacon	Get Beacon information.	Databases
get_ga4gh_beacon_query	Query Beacon.	Databases
post_ga4gh_beacon_query	Query Beacon with POST.	Databases
get_ga4gh_features	Get GA4GH features by ID.	Databases
post_ga4gh_features_search	Get a list of sequence annotation features in GA4GH format	Databases
post_ga4gh_callsets_search	Search GA4GH callsets.	Databases
get_ga4gh_callsets	Get the GA4GH record for a specific CallSet given its identifier	Databases
post_ga4gh_datasets_search	Get a list of datasets in GA4GH format	Databases
get_ga4gh_datasets	Get the GA4GH record for a specific dataset given its identifier	Databases
post_ga4gh_featuresets_search	Search GA4GH feature sets.	Databases
get_ga4gh_featuresets	Return the GA4GH record for a specific featureSet given its identifier	Databases
get_ga4gh_variants	Get GA4GH variant by ID.	Databases
post_ga4gh_variantannotations_search	Return variant annotation information in GA4GH format for a region on a reference sequence	Databases
post_ga4gh_variants_search	Return variant call information in GA4GH format for a region on a reference sequence	Databases
post_ga4gh_variantsets_search	Search GA4GH variant sets.	Databases
get_ga4gh_variantsets	Return the GA4GH record for a specific VariantSet given its identifier	Databases
post_ga4gh_references_search	Return a list of reference sequences in GA4GH format	Databases
get_ga4gh_references	Return data for a specific reference in GA4GH format by id	Databases
post_ga4gh_referencesets_search	Search GA4GH reference sets.	Databases
get_ga4gh_referencesets	Search data for a specific reference set in GA4GH format by ID	Databases
post_ga4gh_variantannotations_sets_search	Return a list of annotation sets in GA4GH format	Databases
get_ga4gh_variantannotationsets	Return meta data for a specific annotation set in GA4GH format by ID	Databases

Tool Name	Description	Category
get_genetree_member_id	Retrieve the gene tree containing a gene identified by its Ensembl ID.	Databases
get_info_biotypes_groups	With :group argument provided, list the properties of biotypes within that group.	Databases
get_info_biotypes_name	List the properties of biotypes with a given name.	Databases
get_info_compara_species_sets	List all collections of species analysed with the specified compara method.	Databases
get_info_comparas	Get all available comparative genomics databases and their data release.	Databases
list_genomes	Get all supported genome assemblies from UCSC Genome Browser.	Databases
list_tracks	List all tracks for a specific genome assembly.	Databases
list_hub_tracks	List all tracks in a specific track hub for a genome.	Databases
list_chromosomes	List all chromosomes for a genome assembly.	Databases
list_public_hubs	Get list of all public UCSC track hubs.	Databases
get_chromosome_sequence	Get sequence for an entire chromosome.	Databases
get_sequence	Get DNA sequence for a genomic region.	Databases
get_cytoband	Get cytoband (chromosome banding) information for a specified genome and chromosome.	Databases
get_active_ingredient_info_by_drug_name	Fetch a list of active ingredients in a specific drug product.	Databases
get dosage_and_storage_information_by_drug_name	Retrieve dosage and storage information for a specific drug.	Databases
get_drug_names_by_abuse_info	Retrieve drug names based on information about types of abuse and adverse reactions pertinent to those types of abuse.	Databases
get_abuse_info_by_drug_name	Retrieve information about types of abuse based on the drug name.	Databases
get_drug_names_by_accessories	Retrieve drug names based on the accessories field information.	Databases
get_accessories_info_by_drug_name	Retrieve information about accessories based on the drug name.	Databases
get_drug_names_by_active_ingredient	Retrieve drug names based on the active ingredient information.	Databases

Tool Name	Description	Category
get_active_ingredient_application_number_manufacturer_name_NDC_number_administration_route_by_drug_name	Retrieve detailed information about a drug's active ingredient, FDA application number, manufacturer name, National Drug Code (NDC) number, and route of administration; all based on the drug name.	Databases
get_drug_names_by_application_number_manufacturer_name_NDC_number	Retrieve drug names based on the specified FDA application number, manufacturer name, or National Drug Code (NDC) number.	Databases
get_drug_name_by_adverse_reaction	Retrieve the drug name based on specific adverse reactions reported.	Databases
get_adverse_reactions_by_drug_name	Retrieve adverse reactions information based on the drug name.	Databases
get_drug_names_by_alarm	Retrieve drug names based on the presence of specific alarms, which are related to adverse reaction events.	Databases
get_alarms_by_drug_name	Retrieve alarms based on the specified drug name.	Databases
get_drug_names_by_animal_pharmacology_info	Retrieve drug names based on animal pharmacology and toxicology information.	Databases
get_animal_pharmacology_info_by_drug_name	Retrieve animal pharmacology and toxicology information based on drug names.	Databases
get_drug_name_by_info_on_conditions_for_doctor_consultation	Retrieve the drug names that require asking a doctor before use due to a patient's specific conditions and symptoms.	Databases
get_info_on_conditions_for_doctor_consultation_by_drug_name	Get information about when a doctor should be consulted before using a specific drug.	Databases
get_drug_names_by_info_on_consulting_doctor_pharmacist_for_drug_interactions	Retrieve drug names based on information about when a doctor or pharmacist should be consulted regarding drug interactions.	Databases
get_info_on_consulting_doctor_pharmacist_for_drug_interactions_by_drug_name	Get information about when a doctor or pharmacist should be consulted regarding drug interactions for a specific drug.	Databases
get_drug_names_by_assembly_installation_info	Retrieve drug names based on assembly or installation instructions.	Databases
get_assembly_installation_info_by_drug_name	Retrieve assembly or installation instructions based on drug names.	Databases

Tool Name	Description	Category
get_drug_names_by_boxed_warning	Retrieve drug names that have specific boxed warnings (The most serious risk alerts in drug labeling) and adverse effects.	Databases
get_boxed_warning_info_by_drug_name	Retrieve boxed warning (The most serious risk alerts in drug labeling) and adverse effects information for a specific drug.	Databases
get_drug_name_by_calibration_instructions	Retrieve the drug name based on the calibration instructions provided.	Databases
get_calibration_instructions_by_drug_name	Retrieve calibration instructions based on the specified drug name.	Databases
get_drug_names_by_carcinogenic_mutagenic_fertility_impairment_info	Retrieve drug names based on the presence of carcinogenic, mutagenic, or fertility impairment information.	Databases
get_carcinogenic_mutagenic_fertility_impairment_info_by_drug_name	Retrieve carcinogenic, mutagenic, or fertility impairment information based on the drug name.	Databases
get_drug_name_by_application_number_NUI_identifier_SPL_document_ID_SPL_set_ID	Retrieves the drug name based on various identifiers such as the FDA application number, NUI, SPL document ID, or SPL set ID.	Databases
get_drug_names_by_clinical_pharmacology	Retrieves drug names based on a search query within their clinical pharmacology information.	Databases
get_clinical_pharmacology_by_drug_name	Retrieves clinical pharmacology information for a specific drug from the FDA database by its name.	Databases
get_drug_names_by_clinical_studies	Retrieves drug names based on a search query within their clinical studies information.	Databases
get_clinical_studies_info_by_drug_name	Retrieves clinical studies information for a specific drug from the FDA database by its name.	Databases
get_drug_names_by_contraindications	Retrieves drug names based on a search query within their contraindications information.	Databases
get_contraindications_by_drug_name	Retrieve contraindications information based on the drug name.	Databases
get_drug_names_by_controlled_substance_DEA_schedule	Retrieves drug names based on a search query within their DEA controlled substance schedule information.	Databases

Tool Name	Description	Category
get_controlled_substance_DEA_schedule_info_by_drug_name	Retrieves information about the controlled substance DEA (Drug Enforcement Administration) schedule for a specific drug.	Databases
get_drug_name_by_dependence_info	Retrieve the drug name based on information about dependence characteristics.	Databases
get_dependence_info_by_drug_name	Retrieves information about dependence characteristics for a specific drug from the FDA database by its name.	Databases
get_drug_names_by_disposal_info	Retrieves drug names based on a search query within their disposal and waste handling information.	Databases
get_disposal_info_by_drug_name	Retrieves disposal and waste handling information for a specific drug from the FDA database by its name.	Databases
get_drug_name_by dosage_info	Retrieve the drug name based on dosage and administration information.	Databases
get_drug_names_by dosage_forms_and_strengths_info	Retrieves drug names based on a search query within their dosage forms and strengths information.	Databases
get dosage_forms_and_strengths_by_drug_name	Retrieves dosage forms and strengths information for a specific drug from the FDA database by its name.	Databases
get_drug_name_by_abuse_types_and_related_adverse_reactions_and_controlled_substance_status	Retrieves the drug name based on information about drug abuse and dependence, including whether the drug is a controlled substance, the types of possible abuse, and adverse reactions relevant to those abuse types.	Databases
get_abuse_types_and_related_adverse_reactions_and_controlled_substance_status_by_drug_name	Retrieves information about drug abuse and dependence for a specific drug from the FDA database by its name, including controlled substance status, abuse types, and related adverse reactions.	Databases
get_drug_names_by_lab_test_interference	Retrieves drug names based on a search query within their laboratory test interference information.	Databases
get_lab_test_interference_info_by_drug_name	Retrieves information about laboratory test interferences for a specific drug.	Databases
get_drug_names_by_drug_interactions	Retrieves a list of drug names that have known interactions with a specified term.	Databases

Tool Name	Description	Category
get_drug_interactions_by_drug_name	Retrieve drug interactions based on the specified drug name.	Databases
get_drug_names_by_effective_time	Retrieve drug names based on the effective time of the labeling document.	Databases
get_effective_time_by_drug_name	Retrieve effective time of the labeling document based on the drug name.	Databases
get_drug_name_by_environmental_warning	Retrieve the drug name based on the specified environmental warnings.	Databases
get_environmental_warning_by_drug_name	Fetch environmental warnings for a specific drug based on its name.	Databases
get_drug_names_by_food_safety_warnings	Retrieve drug names based on specific food safety warnings.	Databases
get_drug_names_by_general_precautions	Retrieve drug names based on specific general precautions information.	Databases
get_general_precautions_by_drug_name	Retrieve general precautions information based on the drug name.	Databases
get_drug_names_by_geriatric_use	Retrieve drug names that have specific information about geriatric use.	Databases
get_geriatric_use_info_by_drug_name	Retrieve information about geriatric use based on the drug name.	Databases
get_dear_health_care_provider_letter_info_by_drug_name	Fetch information about dear health care provider letters for a specific drug.	Databases
get_drug_names_by_dear_health_care_provider_letter_info	Fetch drug names based on information about dear health care provider letters.	Databases
get_drug_names_by_health_claim	Retrieve drug names based on specific health claims.	Databases
get_health_claims_by_drug_name	Retrieve health claims associated with a specific drug name.	Databases
get_drug_name_by_document_id	Retrieve the drug name based on the document ID.	Databases
get_document_id_by_drug_name	Retrieve the document ID based on the drug name.	Databases
get_drug_name_by_inactive_ingredient	Retrieve the drug name based on the inactive ingredient information.	Databases
get_inactive_ingredient_info_by_drug_name	Fetch a list of inactive ingredients in a specific drug product based on the drug name.	Databases
get_drug_names_by_indication	Retrieve a list of drug names based on a specific indication or usage.	Databases
get_indications_by_drug_name	Retrieve indications and usage information based on a specific drug name.	Databases

Tool Name	Description	Category
get_drug_names_by_information_for_owners_or_caregivers	Retrieve drug names based on information for owners or caregivers.	Databases
get_information_for_owners_or_caregivers_by_drug_name	Retrieves information for owners or caregivers for a specific drug.	Databases
get_info_for_patients_by_drug_name	Retrieves patient information for a specific drug from the FDA database by its name.	Databases
get_drug_names_by_instructions_for_use	Retrieves drug names based on a search query within their instructions for use.	Databases
get_instructions_for_use_by_drug_name	Retrieves instructions for use information for a specific drug from the FDA database by its name.	Databases
retrieve_drug_name_by_device_use	Retrieves drug names based on a search query within the intended use of their associated medical device.	Databases
retrieve_device_use_by_drug_name	Retrieves the intended use of the device associated with a specific drug.	Databases
get_drug_names_by_child_safety_info	Retrieves drug names based on a search query within their child safety information.	Databases
get_child_safety_info_by_drug_name	Retrieves child safety information for a specific drug based on its name.	Databases
get_drug_name_by_labor_and_delivery_info	Retrieve the drug name based on information about the drug's use during labor or delivery.	Databases
get_labor_and_delivery_info_by_drug_name	Retrieves information about the drug's use during labor or delivery based on the drug name.	Databases
get_drug_names_by_lab_tests	Retrieves drug names based on a search query within their laboratory tests information.	Databases
get_lab_tests_by_drug_name	Retrieves laboratory tests information for a specific drug from the FDA database by its name.	Databases
get_mechanism_of_action_by_drug_name	Retrieves the mechanism of action information for a specific drug.	Databases
get_drug_names_by_mechanism_of_action	Retrieves drug names based on a search query within their mechanism of action information.	Databases
get_drug_name_by_microbiology	Retrieve the drug name based on microbiology field information.	Databases

Tool Name	Description	Category
get_microbiology_info_by_drug_name	Retrieves microbiology information for a specific drug from the FDA database by its name.	Databases
get_drug_names_by_nonclinical_toxicology_info	Retrieves drug names based on a search query within their nonclinical toxicology information.	Databases
get_nonclinical_toxicology_info_by_drug_name	Retrieves nonclinical toxicology information for a specific drug from the FDA database by its name.	Databases
get_drug_names_by_nonteratogenic_effects	Retrieve drug names based on the presence of nonteratogenic effects information.	Databases
get_nonteratogenic_effects_by_drug_name	Retrieves information about nonteratogenic effects for a specific drug from the FDA database by its name.	Databases
get_drug_names_by_info_for_nursing_mothers	Retrieves drug names based on a search query within their information for nursing mothers.	Databases
get_info_for_nursing_mothers_by_drug_name	Retrieves information about nursing mothers for a specific drug.	Databases
get_drug_name_by_other_safety_info	Retrieves drug names based on a search query within their 'other safety information' section.	Databases
get_other_safety_info_by_drug_name	Retrieves other safety information for a specific drug from the FDA database by its name.	Databases
get_drug_names_by_overdosage_info	Retrieves drug names based on a search query within their overdosage information.	Databases
get_overdosage_info_by_drug_name	Retrieves information about signs, symptoms, and laboratory findings of acute overdosage based on the drug name.	Databases
get_drug_name_by_principal_display_panel	Retrieve the drug name based on the content of the principal display panel of the product package.	Databases
get_principal_display_panel_by_drug_name	Retrieve the content of the principal display panel of the product package based on the drug name.	Databases
retrieve_drug_names_by_patient_medication_info	Retrieve drug names based on patient medication information, which is about safe use of the drug.	Databases

Tool Name	Description	Category
retrieve_patient_medication_in fo_by_drug_name	Retrieve patient medication information (which is about safe use of the drug) based on drug names.	Databases
get_drug_names_by_pediatric_ use	Retrieve drug names based on pediatric use information.	Databases
get_pediatric_use_info_by_dru g_name	Retrieve pediatric use information based on drug names.	Databases
get_drug_name_by_pharmacod ynamics	Retrieve the drug name based on pharmacodynamics information.	Databases
get_pharmacodynamics_by_dr ug_name	Retrieve pharmacodynamics information based on the drug name.	Databases
get_pharmacogenomics_info_b y_drug_name	Retrieve pharmacogenomics information based on the drug name.	Databases
get_drug_names_by_pharmaco kinetics	Retrieve drug names based on specific pharmacokinetics information, such as absorption, distribution, elimination, metabolism, drug interactions, and specific patient populations.	Databases
get_pharmacokinetics_by_drug _name	Retrieve pharmacokinetics information (e.	Databases
get_drug_name_by_precaution s	Retrieve the drug name based on the precautions field information.	Databases
get_precautions_by_drug_nam e	Retrieve precautions information based on the drug name.	Databases
get_drug_names_by_pregnancy _effects_info	Retrieves drug names based on a search query within their pregnancy effects information.	Databases
get_pregnancy_effects_info_by _drug_name	Retrieves information about the effects on pregnancy for a specific drug.	Databases
get_drug_name_by_pregnancy _or_breastfeeding_info	Retrieve the drug names based on pregnancy or breastfeeding information.	Databases
get_pregnancy_or_breastfeedin g_info_by_drug_name	Retrieves pregnancy or breastfeeding information for a specific drug.	Databases
get_contact_for_questions_info _by_drug_name	Retrieves information on who to contact with questions about the drug based on the provided drug name.	Databases
get_recent_changes_by_drug_n ame	Retrieves recent major changes in labeling for a specific drug.	Databases
get_drug_name_by_reference	Retrieves the drug name based on the reference information provided in the drug labeling.	Databases

Tool Name	Description	Category
get_reference_info_by_drug_name	Retrieves reference information for a specific drug from the FDA database by its name.	Databases
get_drug_names_by_residue_warning	Retrieves drug names based on a search query within their residue warning information.	Databases
get_residue_warning_by_drug_name	Retrieves the residue warning for a specific drug from the FDA database by its name.	Databases
get_drug_names_by_risk	Retrieves drug names based on a search query within their risk information, especially regarding pregnancy or breastfeeding.	Databases
get_risk_info_by_drug_name	Retrieves risk information (especially regarding pregnancy or breastfeeding) for a specific drug from the FDA database by its name.	Databases
get_drug_names_by_route	Retrieves drug names based on a search query within their route of administration.	Databases
get_route_info_by_drug_name	Retrieve the route of administration information based on the drug name.	Databases
get_drug_names_by_safe_handling_warning	Retrieves drug names based on a search query within their safe handling warning information.	Databases
get_safe_handling_warnings_by_drug_name	Retrieves safe handling warnings for a specific drug from the FDA database by its name.	Databases
get_drug_name_by_set_id	Retrieve the drug name based on the Set ID of the labeling.	Databases
get_drug_names_by_spl_indexing_data_elements	Retrieve drug names based on Structured Product Labeling (SPL) indexing data elements.	Databases
get_spl_indexing_data_element_by_drug_name	Retrieve Structured Product Labeling (SPL) indexing data elements based on drug names.	Databases
get_drug_names_by_medication_guide	Retrieve drug names based on the presence of specific information in the medication guide.	Databases
get_medication_guide_info_by_drug_name	Retrieve medication guide information based on the drug name.	Databases

Tool Name	Description	Category
get_drug_name_from_patient_package_insert	Retrieve the drug name based on the information provided in the patient package insert.	Databases
get_patient_package_insert_from_drug_name	Retrieve the patient package insert information based on the drug name.	Databases
get_drug_names_by_ingredient	Retrieve drug names based on a specific ingredient present in the drug product.	Databases
get_ingredients_by_drug_name	Retrieve a list of drug ingredients based on the drug name.	Databases
get_spl_unclassified_section_by_drug_name	Retrieve the SPL unclassified section information (Content not yet clearly categorized) based on the drug name.	Databases
get_drug_name_by_stop_use_info	Retrieve the drug name based on the stop use information provided.	Databases
get_stop_use_info_by_drug_name	Retrieve stop use information based on the drug name provided.	Databases
get_drug_name_by_storage_and_handling_info	Retrieve the drug name based on storage and handling information.	Databases
get_storage_and_handling_info_by_drug_name	Retrieve storage and handling information based on the drug name.	Databases
get_drug_names_by_safety_summary	Retrieve drug names based on the summary of safety and effectiveness information.	Databases
get_safety_summary_by_drug_name	Retrieve a summary of safety and effectiveness information based on the drug name.	Databases
get_drug_names_by_teratogenic_effects	Retrieve drug names based on specific teratogenic effects categories.	Databases
get_teratogenic_effects_by_drug_name	Retrieve teratogenic effects information based on the drug name.	Databases
get_drug_names_by_population_use	Retrieve drug names based on their use in specific populations, such as pregnant women, nursing mothers, pediatric patients, and geriatric patients.	Databases
get_population_use_info_by_drug_name	Retrieve information about the use of a drug in specific populations based on the drug name.	Databases
get_user_safety_warning_by_drug_names	Retrieve specific user safety warnings based on drug names.	Databases
get_drug_names_by_user_safety_warning	Retrieve drug names that have specific user safety warnings.	Databases

Tool Name	Description	Category
get_drug_name_by_warnings	Retrieve the drug names based on specific warning information.	Databases
get_warnings_by_drug_name	Retrieve warning information based on the drug name.	Databases
get_warnings_and_cautions_by_drug_name	Retrieve warnings and cautions information for a specific drug based on its name.	Databases
get_drug_names_by_warnings_and_cautions	Retrieve drug names based on specific warnings and cautions information.	Databases
get_when_using_info	Retrieve information about side effects and substances or activities to avoid while using a specific drug.	Databases
get_brand_name_generic_name	Retrieve the brand name and generic name from generic name or brand name of a drug.	Databases
get_do_not_use_info_by_drug_name	Retrieve information about all contraindications for use based on the drug name.	Databases
get_purpose_info_by_drug_name	Retrieve about the drug product's indications for use based on the drug name.	Databases
get_drug_generic_name	Get the drug's generic name based on the drug's generic or brand name.	Databases
get_associated_targets_by_disease_efoId	Find targets associated with a specific disease or phenotype based on efoId.	Databases
get_associated_diseases_phenotypes_by_target_ensemblID	Find diseases or phenotypes associated with a specific target using ensemblID.	Databases
target_disease_evidence	Explore evidence that supports a specific target-disease association.	Databases
get_drug_warnings_by_chemblId	Retrieve warnings for a specific drug using ChEMBL ID.	Databases
get_drug_mechanisms_of_action_by_chemblId	Retrieve the mechanisms of action associated with a specific drug using chemblId.	Databases
get_associated_drugs_by_disease_efoId	Retrieve known drugs associated with a specific disease by disease efoId.	Databases
get_similar_entities_by_disease_efoId	Retrieve similar entities for a given disease efoId using a model trained with PubMed.	Databases
get_similar_entities_by_target_ensemblID	Retrieve similar entities for a given target ensemblID using a model trained with PubMed.	Databases
get_associated_phenotypes_by_disease_efoId	Find HPO phenotypes associated with the specified disease efoId.	Databases

Tool Name	Description	Category
get_drug_withdrawn_blackbox_status_by_chemblId	Find withdrawn and black-box warning statuses for a specific drug by chemblId.	Databases
search_category_counts_by_query_string	Get the count of entries in each entity category (disease, target, drug) based on a query string.	Databases
get_disease_id_description_by_name	Retrieve the efoId and additional details of a disease based on its name.	Databases
get_drug_id_description_by_name	Fetch the drug chemblId and description based on the drug name.	Databases
get_drug_indications_by_chemblId	Fetch indications (treatable phenotypes/diseases) for a given drug chemblId.	Databases
get_target_gene_ontology_by_ensemblID	Retrieve Gene Ontology annotations for a specific target by Ensembl ID.	Databases
get_target_homologues_by_ensemblID	Fetch homologues for a specific target by Ensembl ID.	Databases
get_target_safety_profile_by_ensemblID	Retrieve known target safety liabilities for a specific target Ensembl ID.	Databases
get_biological_mouse_models_by_ensemblID	Retrieve biological mouse models, including allelic compositions and genetic backgrounds, for a specific target.	Databases
get_target_genomic_location_by_ensemblID	Retrieve genomic location data for a specific target, including chromosome, start, end, and strand.	Databases
get_target_subcellular_location_s_by_ensemblID	Retrieve information about subcellular locations for a specific target Ensembl ID.	Databases
get_target_synonyms_by_ensemblID	Retrieve synonyms for a specified target, including alternative names and symbols, using the given Ensembl ID.	Databases
get_target_tractability_by_ensemblID	Retrieve tractability assessments, including modality and values, for a specific target Ensembl ID.	Databases
get_target_classes_by_ensemblID	Retrieve the target classes associated with a specific target Ensembl ID.	Databases
get_target_enabling_packages_by_ensemblID	Retrieve the Target Enabling Packages (TEP) associated with a specific target Ensembl ID.	Databases
get_target_interactions_by_ensemblID	Retrieve interaction data for a specific target Ensembl ID, including interaction partners and evidence.	Databases

Tool Name	Description	Category
get_disease_ancestors_parents_by_efoId	Retrieve the disease ancestors and parents in the ontology using the disease EFO ID.	Databases
get_disease_descendants_children_by_efoId	Retrieve the disease descendants and children in the ontology using the disease EFO ID.	Databases
get_disease_locations_by_efoId	Retrieve the disease's direct location and indirect location disease terms and IDs using the disease EFO ID.	Databases
get_disease_synonyms_by_efoId	Retrieve disease synonyms by its EFO ID.	Databases
get_disease_description_by_efoId	Retrieve disease description, name, database cross-references, obsolete terms, and whether it's a therapeutic area, all using the specified EFO ID.	Databases
get_disease_therapeutic_areas_by_efoId	Retrieve the therapeutic areas associated with a specific disease EFO ID.	Databases
get_drug_adverse_events_by_chemblId	Retrieve significant adverse events reported for a specific drug ChEMBL ID.	Databases
get_known_drugs_by_drug_chemblId	Get a list of known drugs and associated information using the specified ChEMBL ID.	Databases
get_parent_child_molecules_by_drug_chembl_ID	Get parent and child molecules of specified drug ChEMBL ID.	Databases
get_approved_indications_by_drug_chemblId	Retrieve detailed information about multiple drugs using a list of ChEMBL IDs.	Databases
get_drug_description_by_chemblId	Get drug name, year of first approval, type, cross references, and max clinical trial phase based on specified chemblId.	Databases
get_drug_synonyms_by_chemblId	Retrieve the synonyms associated with a specific drug ChEMBL ID.	Databases
get_drug_trade_names_by_chemblId	Retrieve the trade names associated with a specific drug ChEMBL ID.	Databases
get_drug_approval_status_by_chemblId	Retrieve the approval status of a specific drug ChEMBL ID.	Databases
get_chemical_probes_by_target_ensemblID	Retrieve chemical probes associated with a specific target using its Ensembl ID.	Databases
drug_pharmacogenomics_data	Retrieve pharmacogenomics data for a specific drug, including evidence levels and genotype annotations.	Databases

Tool Name	Description	Category
get_associated_drugs_by_target_ensemblID	Get known drugs associated with a specific target Ensembl ID, including clinical trial phase and mechanism of action of the drugs.	Databases
get_associated_diseases_by_drug_chemblId	Retrieve the list of diseases associated with a specific drug ChEMBL ID based on clinical trial data or post-marketed drugs.	Databases
get_associated_targets_by_drug_chemblId	Retrieve the list of targets linked to a specific drug ChEMBL ID based on its mechanism of action.	Databases
multi_entity_search_by_query_string	Perform a multi-entity search based on a query string, filtering by entity names and pagination settings.	Databases
get_gene_ontology_terms_by_gOID	Retrieve Gene Ontology terms based on a list of GO IDs.	Databases
get_target_constraint_info_by_ensemblID	Retrieve genetic constraint information for a specific target Ensembl ID, including expected and observed values, and scores.	Databases
get_publications_by_target_ensemblID	Retrieve publications related to a target Ensembl ID, including PubMed IDs and publication dates.	Databases
get_publications_by_drug_chemblId	Retrieve publications related to a drug ChEMBL ID, including PubMed IDs and publication dates.	Databases
get_target_id_description_by_name	Get the Ensembl ID and description based on the target name.	Databases
get_general_info_by_disease_name	Get disease EFO ID and description by disease name from OpenTargets.	Databases
get_target_ensembl_id	Get target Ensembl ID by target name.	Databases
get_disease_efo_id	Get disease EFO ID by disease name.	Databases
get_drug_chembl_id_by_name	Find drug ChEMBL ID by drug name.	Databases
get_associated_targets_by_disease_name	Find targets associated with a specific disease or phenotype based on its name.	Databases
get_associated_diseases_phenotypes_by_target_name	Find diseases or phenotypes associated with a specific target.	Databases
get_target_disease_evidence_by_name	Explore evidence that supports a specific target-disease association.	Databases
get_drug_warnings_by_name	Retrieve warnings for a specific drug.	Databases

Tool Name	Description	Category
get_drug_mechanisms_of_action_by_name	Retrieve the mechanisms of action associated with a specific drug.	Databases
get_associated_drugs_by_disease_name	Retrieve known drugs associated with a specific disease by disease name.	Databases
get_similar_entities_by_disease_name	Retrieve similar entities for a given disease using a model trained with PubMed.	Databases
get_similar_entities_by_target_name	Retrieve similar entities for a given target using a model trained with PubMed.	Databases
get_associated_phenotypes_by_disease_name	Find HPO phenotypes associated with the specified disease.	Databases
get_drug_indications_by_name	Fetch indications (treatable phenotypes/diseases) for a given drug.	Databases
get_target_gene_ontology_by_name	Retrieve Gene Ontology annotations for a specific target.	Databases
get_target_homologues_by_name	Fetch homologues for a specific target.	Databases
get_target_safety_profile_by_name	Retrieve known target safety liabilities for a specific target.	Databases
get_biological_mouse_models_by_target_name	Retrieve biological mouse models, including allelic compositions and genetic backgrounds, for a specific target.	Databases
get_target_genomic_location_by_name	Retrieve genomic location data for a specific target, including chromosome, start, end, and strand.	Databases
get_target_subcellular_locations_by_name	Retrieve information about subcellular locations for a specific target.	Databases
get_target_synonyms_by_name	Retrieve synonyms for specified target, including alternative names and symbols.	Databases
get_target_tractability_by_name	Retrieve tractability assessments, including modality and values.	Databases
get_target_classes_by_name	Retrieve the target classes associated with a specific target.	Databases
get_target_enabling_packages_by_name	Retrieve the Target Enabling Packages (TEP) associated with a specific target.	Databases
get_target_interactions_by_name	Retrieve interaction data for a specific target, including interaction partners and evidence.	Databases
get_disease_ancestors_parents_by_name	Retrieve the ancestors and parents of a specific disease.	Databases

Tool Name	Description	Category
get_disease_descendants_children_by_name	Retrieve the descendants and children of a specific disease.	Databases
get_disease_locations_by_name	Retrieve the locations of a specific disease.	Databases
get_disease_synonyms_by_name	Retrieve synonyms for a specific disease.	Databases
get_disease_description_by_name	Retrieve the description of a specific disease.	Databases
get_disease_therapeutic_areas_by_name	Retrieve the therapeutic areas associated with a specific disease.	Databases
get_chemical_probes_by_target_name	Retrieve chemical probes associated with a specific target.	Databases
get_associated_drugs_by_target_name	Get known drugs associated with a specific target, including clinical trial phase and mechanism of action of the drugs.	Databases
get_associated_diseases_by_drug_name	Retrieve the list of diseases associated with a specific drug based on clinical trial data or post-marketed drugs.	Databases
get_associated_targets_by_drug_name	Retrieve the list of targets linked to a specific drug based on its mechanism of action.	Databases
get_target_constraint_info_by_name	Retrieve genetic constraint information for a specific target, including expected and observed values, and scores.	Databases
get_joint_associated_diseases_by_HPO_ID_list	Retrieve diseases associated with a list of phenotypes or symptoms by a list of HPO IDs.	Databases
get_phenotype_by_HPO_ID	Retrieve a phenotype or symptom by its HPO ID.	Databases
get_HPO_ID_by_phenotype	Retrieve one or more HPO ID of a phenotype or symptom.	Databases
interproscan_analyze	Analyze protein sequence using InterProScan to identify functional domains, families, and GO terms.	Databases
blast_search	Search for similar protein sequences in UniProt Swiss-Prot database using BLAST.	Databases
get_material_coefficient	Retrieve the coefficient for the specified material.	Databases
get_element_magnetic_moment	Get the magnetic moment value for an element.	Databases
CASNumberLookup	Lookup CAS number for a compound.	Databases
PubChemCompoundLookup	Complete PubChem compound information.	Databases
CompoundSimilaritySearch	Search for similar compounds.	Databases

Tool Name	Description	Category
CompoundPropertiesRetriever	Retrieve compound properties from PubChem.	Databases
CompoundNameResolver	Resolve compound name to all identifiers.	Databases
UniProtProteinAnalyzer	Analyze protein from UniProt.	Databases
MaterialStructureInfoRetriever	Get structure information.	Databases
CompoundToMaterialPrice	Get material price from compound.	Databases
CASToPrice	Fetches average price for multiple chemical substances identified by their CAS numbers.	Databases
SMILESToCAS	Query a SMILES and return their CAS numbers in string.	Databases
MofLattice	Obtain lattice structure information from the provided MOF cif file name.	Databases
NameToSMILES	Query a molecule name and return its SMILES string in Markdown format.	Databases
NameToCas	Query a molecule name and return its CAS number in Markdown format.	Databases
search_literature	Search literature.	Literature Search
tavily_search	Run the search engine with a given query, retrieving and filtering results.	Literature Search
jina_search	Run the search engine with a given query, retrieving and filtering results.	Literature Search
pubmed_search	Search PubMed for academic articles and retrieve abstracts.	Literature Search
get_publications_by_disease_ef old	Retrieve publications related to a disease EFO ID, including PubMed IDs and publication dates.	Literature Search
get_publications_by_disease_n ame	Retrieve publications related to a disease name, including PubMed IDs and publication dates.	Literature Search
get_publications_by_target_na me	Retrieve publications related to a target, including PubMed IDs and publication dates.	Literature Search
get_publications_by_drug_n ame	Retrieve publications related to a drug, including PubMed IDs and publication dates.	Literature Search
predict_zero_shot_sequence	Predict zero-shot sequence.	Model Services
predict_zero_shot_structure	Predict zero-shot structure.	Model Services
predict_protein_properties	Predict protein properties.	Model Services
pred_protein_structure_esmfol d	Use the ESMFold model for protein 3D structure prediction.	Model Services
pred_molecule_admet	Predict the ADMET properties of a molecule.	Model Services

Tool Name	Description	Category
pred_mutant_sequence	Given a protein sequence and its structure, employ the ProSST model to predict mutation effects and obtain the top-k mutated sequences based on their scores.	Model Services
get_similar_entities_by_drug_c hemblId	Retrieve similar entities for a given drug chemblId using a model trained with PubMed.	Model Services
get_similar_entities_by_drug_n ame	Retrieve similar entities for a given drug using a model trained with PubMed.	Model Services
CompoundToADMET	From compound name to ADMET prediction.	Model Services
ProteinSolubilityPredictor	Predicts the solubility of a given protein sequence using a fine-tuned BioT5 model for the protein solubility prediction task.	Model Services
GenerateMoleculeDescription	Given a molecule SELFIES, generates its English description using a pre-trained T5 model.	Model Services
TexToMoleculeSELFIES	Given a molecule description in English, generates its SELFIES and SMILES representation using a pre-trained T5 model.	Model Services
PredictDrugTargetInteraction	Predicts whether a given molecule (SELFIES format) and a protein sequence can interact with each other and returns the result in Markdown format with a brief explanation.	Model Services
PredictHumanProteinInteraction	Predicts whether two given protein sequences can interact with each other using a fine-tuned BioT5 model for the protein-protein interaction task with human dataset.	Model Services
PredictYeastProteinInteraction	Predicts whether two given yeast protein sequences can interact with each other using a fine-tuned BioT5 model for the protein-protein interaction task with yeast dataset.	Model Services
PredictProteinSolubility	Predicts the solubility of a given protein sequence using a fine-tuned BioT5 model for the protein solubility prediction task.	Model Services

Tool Name	Description	Category
PredictProteinBinaryLocalization	Predicts the binary localization of a given protein sequence using a fine-tuned BioT5 model for the protein binary localization prediction task.	Model Services
store	Manually adjust storage condition for a container.	Wet-lab Operations
spin	Apply acceleration (spin) to a container.	Wet-lab Operations
incubate	Incubate a container at a location for a duration.	Wet-lab Operations
seal	Seal a container.	Wet-lab Operations
cover	Place a lid on a container.	Wet-lab Operations
wash	Perform washing of a target.	Wet-lab Operations
dry	Dry an item by a method.	Wet-lab Operations
close	Close an item (e.	Wet-lab Operations
filter	Filter a sample through a filter type.	Wet-lab Operations
record	Record data of a specified type.	Wet-lab Operations
stand	Let an item stand for some duration.	Wet-lab Operations
thaw	Thaw materials at a specific temperature.	Wet-lab Operations
aspirate	Aspirate liquid from a source.	Wet-lab Operations
thermocycle	Thermocycle a container with groups of steps.	Wet-lab Operations
wait	Wait for a specified duration.	Wet-lab Operations
heat	Heat a target to a temperature for a duration.	Wet-lab Operations
vortex	Vortex an item.	Wet-lab Operations
separate	Separate phases/components from a target.	Wet-lab Operations
lyse	Lyse a sample using a method and optional buffer.	Wet-lab Operations
add	Add a component to a target container.	Wet-lab Operations
inoculate	Inoculate medium with inoculum.	Wet-lab Operations
inactivate	Inactivate a target using specified method and parameters.	Wet-lab Operations
anneal	Anneal material at a temperature for a duration with optional cycles.	Wet-lab Operations
transfer	Transfer material from source to destination.	Wet-lab Operations
resuspend	Resuspend a material/pellet in buffer.	Wet-lab Operations
mix	Mix components in a target container.	Wet-lab Operations
extract	Extract target material from a sample.	Wet-lab Operations
grind	Grind material with specified method and parameters.	Wet-lab Operations
prepare	Prepare reagents or mixtures.	Wet-lab Operations

Tool Name	Description	Category
freeze	Freeze an item at given temperature.	Wet-lab Operations
pipetting	Perform pipetting from source to destination.	Wet-lab Operations
extend	Perform extension (e.	Wet-lab Operations)
invert	Invert a container repeatedly to mix.	Wet-lab Operations
place	Place an item at a location.	Wet-lab Operations
fill	Fill a container with a substance.	Wet-lab Operations
tap	Tap or flick a container to mix or settle.	Wet-lab Operations
centrifuge	Centrifuge a container at a speed for a duration.	Wet-lab Operations
pellet	Pellet material by specified method (e.	Wet-lab Operations)
aliquot	Aliquot a source into destination containers.	Wet-lab Operations
eliminate	Eliminate a target (e.	Wet-lab Operations)
measure	Measure a parameter on a sample.	Wet-lab Operations
balance	Balance items for equipment (e.	Wet-lab Operations)
shake	Shake a target container.	Wet-lab Operations
decant	Decant liquid from source to destination.	Wet-lab Operations
purify	Purify a sample using a method/kit.	Wet-lab Operations
sequence	Perform sequencing run with specified parameters.	Wet-lab Operations
digest	Digest a sample with an enzyme under specified conditions.	Wet-lab Operations
elute	Elute material from a sample using a buffer.	Wet-lab Operations
stain	Stain a target with an agent.	Wet-lab Operations
discard	Discard items or materials safely.	Wet-lab Operations
dilute	Dilute a sample by a factor.	Wet-lab Operations
dissolve	Dissolve material in a solvent under specified conditions.	Wet-lab Operations
run_gel_and_image	Abstract instruction to run an agarose gel and acquire an image using an appropriate DNA stain and imaging system.	Wet-lab Operations
protocol_generation	Generation experiment protocol given user prompt.	Wet-lab Operations

Tool Name	Description	Category
execute_json	Execute JSON with PCR operation server Params: json (str): executable JSON server_url (str): PCR operation server URL; default to Thoth-OP server URL Returns: exec_info (str): execution information Arg: arguments (dict) - Tool parameter dictionary	Wet-lab Operations