SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

July 2025, Vol. 68, Iss. 7, 170105:1–170105:18
https://doi.org/10.1007/s11432-024-4485-0

Special Topic: AI for Biology

From intention to implementation: automating biomedical research via LLMs

Yi LUO[1], Linghang SHI[1], Yihao LI[1], Aobo ZHUANG[2], Yeyun GONG[3*],
Ling LIU[4] & Chen LIN[1,5*]

[1]*School of Informatics, National Institute for Data Science in Health and Medicine, Xiamen University,
Xiamen 361101, China*
[2]*School of Medicine, Xiamen University, Xiamen 361102, China*
[3]*Microsoft Research Asia, Beijing 100080, China*
[4]*College of Computing, Georgia Institute of Technology, Atlanta 30332, USA*
[5]*Zhongguancun Academy, Beijing 100094, China*

Received 13 December 2024/Revised 11 April 2025/Accepted 19 May 2025/Published online 23 June 2025

Abstract Conventional biomedical research is increasingly labor-intensive due to the exponential growth of scientific literature and datasets. Artificial intelligence (AI), particularly large language models (LLMs), has the potential to revolutionize this process by automating various steps. Still, significant challenges remain, including the need for multidisciplinary expertise, logicality of experimental design, and performance measurements. This paper introduces BioResearcher, the first end-to-end automated system designed to streamline the entire biomedical research process involving dry lab experiments. BioResearcher employs a modular multi-agent architecture, integrating specialized agents for search, literature processing, experimental design, and programming. By decomposing complex tasks into logically related sub-tasks and utilizing a hierarchical learning approach, BioResearcher effectively addresses the challenges of multidisciplinary requirements and logical complexity. Furthermore, BioResearcher incorporates an LLM-based reviewer for in-process quality control and introduces novel evaluation metrics to assess the quality and automation of experimental protocols. BioResearcher successfully achieves an average execution success rate of 63.07% across eight previously unmet research objectives. The generated protocols, on average, outperform typical agent systems by 22.0% on five quality metrics. The system demonstrates significant potential to reduce researchers' workloads and accelerate biomedical discoveries, paving the way for future innovations in automated research systems.

Keywords biomedical research, AI for research, large language models, multi-agent systems, automation

Citation Luo Y, Shi L H, Li Y H, et al. From intention to implementation: automating biomedical research via LLMs. Sci China Inf Sci, 2025, 68(7): 170105, https://doi.org/10.1007/s11432-024-4485-0

1 Introduction

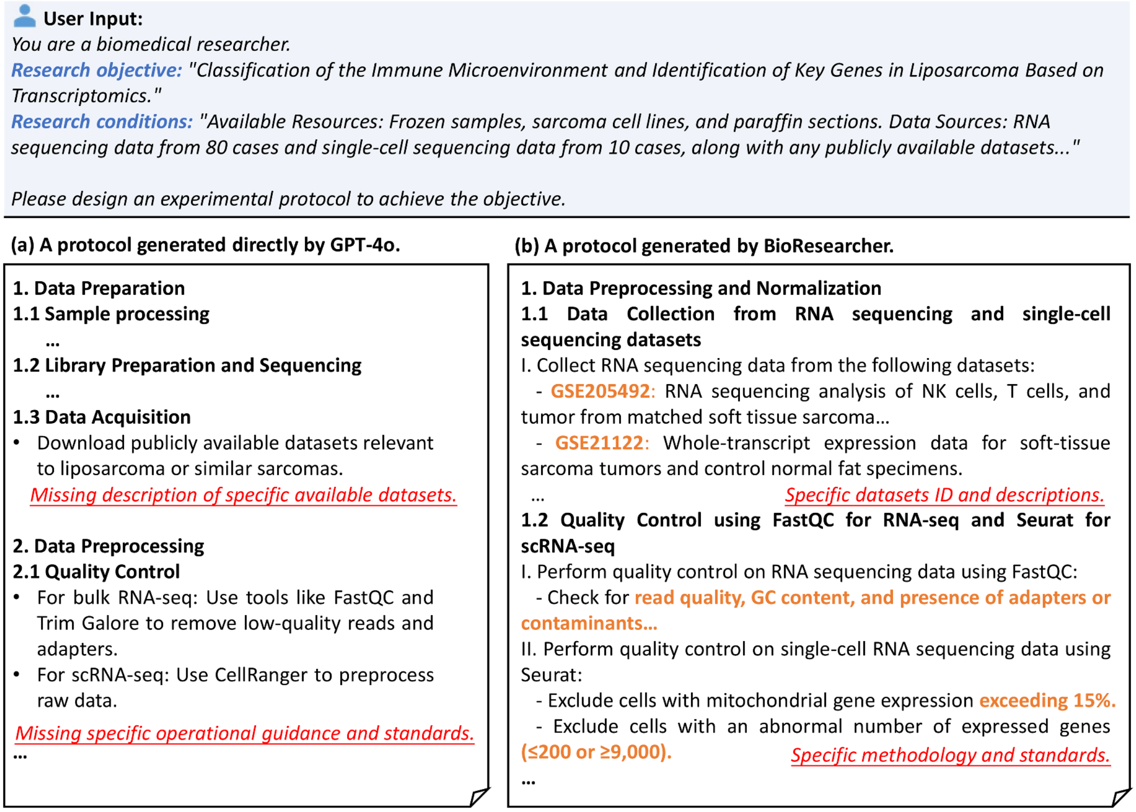
Biomedical research is a fundamental driving force behind human development. By uncovering the underlying mechanisms of diseases [1], biomedical research improves global health, extends life expectancy, and enhances life quality. It also fuels economic growth, scientific advancements, and societal well-being.

Traditional biomedical research relies heavily on labor-intensive processes like manual data collection, comprehensive literature reviews, complex experimental designs, and extensive data analysis. Although the conventional approach has facilitated notable breakthroughs in disease prevention [2–4], diagnosis [5–7], and treatment [8–10], it struggles to keep pace with the data explosion. For example, PubMed now hosts over 37 million citations[1)], overwhelming researchers and making it challenging to stay updated with the latest findings. Moreover, traditional research often demands repetitive tasks or interdisciplinary skills, such as coding and hindering research efficiency.

Artificial intelligence (AI) is emerging as a valuable tool in biomedical research, enhancing specific steps in the research pipeline. For instance, novel machine learning models are designed to analyze data and make decisions, including predicting drug-target from compound-protein interactions [11–13], detecting the presence of cancer from medical images [14, 15], and estimating patient outcomes from medical

[*] Corresponding author (email: yegong@microsoft.com, chenlin@xmu.edu.cn)

[1)] https://pubmed.ncbi.nlm.nih.gov/?Db=pubmed.



User Input:

You are a biomedical researcher.

Research objective: *"Classification of the Immune Microenvironment and Identification of Key Genes in Liposarcoma Based on Transcriptomics."*

Research conditions: *"Available Resources: Frozen samples, sarcoma cell lines, and paraffin sections. Data Sources: RNA sequencing data from 80 cases and single-cell sequencing data from 10 cases, along with any publicly available datasets..."*

Please design an experimental protocol to achieve the objective.

(a) A protocol generated directly by GPT-4o.

1. Data Preparation
1.1 Sample processing
…
1.2 Library Preparation and Sequencing
…
1.3 Data Acquisition

- Download publicly available datasets relevant to liposarcoma or similar sarcomas.

Missing description of specific available datasets.

2. Data Preprocessing
2.1 Quality Control

- For bulk RNA-seq: Use tools like FastQC and Trim Galore to remove low-quality reads and adapters.
- For scRNA-seq: Use CellRanger to preprocess raw data.

Missing specific operational guidance and standards.

…

(b) A protocol generated by BioResearcher.

1. Data Preprocessing and Normalization
1.1 Data Collection from RNA sequencing and single-cell sequencing datasets

I. Collect RNA sequencing data from the following datasets:
 - **GSE205492:** RNA sequencing analysis of NK cells, T cells, and tumor from matched soft tissue sarcoma…
 - **GSE21122:** Whole-transcript expression data for soft-tissue sarcoma tumors and control normal fat specimens.
 …

Specific datasets ID and descriptions.

1.2 Quality Control using FastQC for RNA-seq and Seurat for scRNA-seq

I. Perform quality control on RNA sequencing data using FastQC:
 - Check for **read quality, GC content, and presence of adapters or contaminants…**

II. Perform quality control on single-cell RNA sequencing data using Seurat:
 - Exclude cells with mitochondrial gene expression **exceeding 15%.**
 - Exclude cells with an abnormal number of expressed genes **(≤200 or ≥9,000).**

Specific methodology and standards.

…

Figure 1 Example of experimental protocol generation. For the same user input, (a) illustrates an experimental protocol generated by a single LLM (GPT-4o), while (b) demonstrates a protocol generated by BioResearcher.

history [16–18]. The advancement of large language models (LLMs) further supports AI's role in academic writing [19, 20] and literature summarizing [21]. However, biomedical research is still time-consuming. The potential for automating the entire biomedical research process remains largely unexplored [22], which allows researchers to concentrate on innovation and strategic decision-making.

Despite the potential of AI, several critical challenges must be overcome to build a fully automated biomedical research assistant.

Firstly, biomedical research demands a multidisciplinary skill set, including a fundamental understanding of biology and medicine, comprehension of literature for available datasets and effective approaches, proficiency in programming languages to write code, and knowledge of statistics to interpret results. General-purpose LLMs like OpenAI's GPT-4o lack the domain-specific expertise needed for these tasks. For instance, as illustrated in Figure 1(a), when GPT-4o is asked to design a biomedical protocol for a specific research objective, the response is not executable due to missing details such as datasets and operational guidance.

Secondly, biomedical research is logically complex. On one hand, it requires a coherent understanding of literature with intricate logical structures, yet existing LLMs perform poorly on critical literature analysis. For example, irrelevant information in lengthy research papers causes catastrophic forgetting of important facts. On the other hand, biomedical research involves breaking down complex problems into logically related subtasks. For example, studying pyroptosis in dedifferentiated liposarcoma (DDL) involves multiple interconnected tasks like analyzing the expression variation and genetic changes of pyroptosis-related genes (PRGs), performing immune infiltration analyses, identifying PRG-related clusters, characterizing the tumor microenvironment within these clusters, and developing a prognostic gene model based on these clusters [23]. Each task is logically interdependent, making planning and execution by LLMs challenging.

Lastly, it is crucial to measure the performance of the research assistant. From the quality control perspective, assessing the results of intermediate steps ensures that the research assistant provides reliable final outputs. From the evaluation perspective, detecting the errors of end-to-end responses helps to identify the weaknesses and strengths of different systems and sheds insight for future improvements. Due to the complexity of biomedical research tasks, manually conducting fine-grained evaluation is infeasible; for example, researchers are not patient with labeling errors in an experimental protocol line by line. Current automatic evaluations, such as ROUGE [24] and BLEU [25], compute the overlap between the LLM's generation and the ground-truth answer, which is unavailable. Furthermore, they focus on textual quality while ignoring aspects such as technical completeness and correctness, which are more important in assessing the quality of experiment protocols.

This paper introduces BioResearcher[2], an intelligent research assistant designed to automate the entire biomedical research process. BioResearcher can take any research objective and conditions the user provides, survey relevant literature, design an appropriate experiment protocol[3], and write programs to implement the protocol and derive meaningful conclusion. BioResearcher develops novel techniques to address the three challenges above.

BioResearcher employs a modular multi-agent architecture to integrate multidisciplinary skills. It comprises four modules: search, literature processing, experimental design, and programming, each containing multiple specialized agents. These agents specialize in distinct tasks, including literature and dataset search, filtering, reports generation from literature, reports analysis, experimental protocol design, dry lab experiment extraction, code writing and execution, and review. The specialization adapts LLMs to different task requirements; e.g., a search agent is more professional in retrieving relevant literature than a general-purpose LLM. The collaboration among multiple agents and modules increases the overall performance. As illustrated in Figure 1(b), the experimental design module generates a comprehensive experimental protocol detailing specific datasets, methodologies, and standards. This achievement is facilitated by the effective collection, processing, and analysis of relevant literature and datasets by the search and literature processing modules. Compared with agent systems employing planning agents powered by LLMs to determine agent participation and intervention strategies throughout task execution [26, 27], our system adopts a professionally designed and rigorously structured workflow framework that constrains agent generation processes through systematic procedural constraints, thereby ensuring enhanced stability and reproducibility. Furthermore, our framework diverges from existing human-curated multi-agent systems [28–30]. In computer science and other fields, current studies often prioritize novelty. However, due to the nature of our biomedical applications, we introduce the literature processing module to ensure the reliability and feasibility of our approach in the biomedical domain.

BioResearcher adopts a hierarchical learning approach to decompose the complex logical structure. The literature processing module standardizes relevant papers into experimental reports to minimize unimportant information and provides analyses. The experimental design module then uses the retrieval-augmented generation (RAG) [26] technique, aided by the analyses, to learn knowledge at different levels of granularity, including relevant headings, outlines, and experimental details, thereby facilitating the design of new protocols in a stepwise manner.

BioResearcher introduces an LLM-based reviewer to provide feedback and refine itself for in-process quality control. This approach allows for the ongoing assessment of the generated content, ensuring it meets quality standards and aligns with research objectives. Moreover, we propose new evaluation metrics to assess the quality of the end-to-end performance, including five dimensions for protocol quality, completeness, level of detail, correctness, logical soundness, and structural soundness, and two metrics for experimental automation: execution success rate and error level.

Our contributions are summarized as follows.

- We present the first end-to-end automated system designed specifically for biomedical research. At its core is a multi-agent framework powered by LLMs, which decomposes complex research tasks into specialized subtasks. By enabling collaborative execution among domain-specific agents, the system enhances overall performance, significantly reduces manual effort, and improves efficiency.
- We propose new evaluation metrics to assess the quality of the end-to-end performance, including five dimensions for protocol quality and two metrics for experimental automation.
- Our system successfully achieves an average execution success rate of 63.07% across eight previously unmet research objectives composed by senior researchers, with protocols outperforming typical agent systems by 22.0% regarding the proposed quality metrics.

[2] Our code and prompts are available at https://github.com/XMUDM/BioResearcher.

[3] Our effort primarily focuses on dry lab experiments, which rely on computational methods to conduct bioinformatics analysis. Future research endeavors may extend to wet lab experiments.

- This study explores BioResearcher's potential to automate biomedical research, paving the way for future innovations and accelerating discoveries.

2 Related work

AI has improved biomedical applications for processing different textual data [31–33]. Conventionally, small-scale language models [34] were used, but scaling laws indicate that increasing model parameters brings enhanced performance, leading to superior reasoning capability and higher answer accuracy. Therefore, we have seen large amounts of applications based on LLMs [35–39].

Biomedical LLMs. LLMs are generally pre-trained on vast open-domain corpora but lack domain-specific knowledge. To enhance domain-specific performance, several techniques are used. (1) Fine-tuning optimizes the total or a part of parameters to improve the model performance on a small, specific dataset [35,37,38,40]. (2) Reinforcement learning with human feedback (RLHF) or AI feedback (RLAIF) updates the model parameters to align LLM's responses with responses from humans or a teacher-model via a reinforcement learning framework [36,39,41]. (3) Prompt engineering [42] involves giving instructions or examples to LLM to enforce rules or enhance reasoning [43–45]. These methods mainly focus on question-answering (QA) tasks, like answering medical consultations [36], taking medical examinations [35], and summarizing clinical reports [33]. However, LLMs still struggle with complex tasks that require more profound understanding and reasoning [22]. Compared with QA tasks, responses for scientific research tasks are much longer and more professional and involve intricate, long logical chains. Fine-tuning and RLHF are infeasible due to the lack of instruction data, and simple prompt engineering cannot empower LLM with deep logical reasoning for professional topics.

LLM-based agents for research. LLM-based agents offer significant advantages over the direct use of LLMs, such as actively acquiring information, interacting with environments, and stronger reasoning and planning [22]. LLMs can function as a single agent by being assigned with specific roles through in-domain fine-tuning [29] or role-specific prompts [33,46–51]. In contrast, a multi-agent system (MAS) comprises multiple LLM-based agents, enabling task completion through various cooperative methods. One approach involves assigning distinct roles to agents, who then reach a consensus through negotiation, like discussing clinical diagnosis via several medical experts [52–54]. However, the consensus can be unreliable due to potential instability and hallucinations from LLMs. Alternatively, MAS can distribute complex tasks into sub-tasks among agents, such as dividing a scientific discovery process into creating ideas, experimentation, and writing [28]. MAS is promising for tasks with complex logical chains and highly detailed requirements. After decomposing tasks and assigning them to agents, we can synthesize the results effectively. Furthermore, employing a professional and rigorous workflow framework helps constrain the agents' generative processes, ensuring highly reliable outcomes.

AI for research. AI for research (AI4R) [55] has gained significant attention, particularly in automating scientific workflows. Existing studies are categorized into four automation levels [22]: Level 0 automation performs specific predefined tasks [56]; Level 1 automation designs simple experimental protocols with in-silico or lab tools [46,57]; Level 2 automation develops rigorous experimental protocols and employs statistical methods for hypothesis evaluation [28,58]. Level 3 agents, which remain undeveloped, are envisioned to discover new methods and employ diverse techniques to measure biological phenomena.

Our work belongs to Level 2 automation and differs from current AI4R systems. (1) The system is not limited to solving a specific biomedical task. For example, CRISPR-GPT [59] customizes the workflow only for gene editing experiments. (2) We automate the entire process without manual maintenance of a template database like Genesis [60]. (3) Most Level 2 AI4R systems are designed for computer science (CS) [28–30,61,62], where public datasets like OpenReview are available, facilitating training and evaluation. Such extensive review data are lacking in the biomedical domain. (4) Existing studies emphasize novelty while we focus on reliability and feasibility, necessitating fine-grained literature analysis. For example, AIScientist [28] improves existing solutions based on a given task and initial experimental code. Conversely, BioResearcher designs a series of executable experiments for a new research subject.

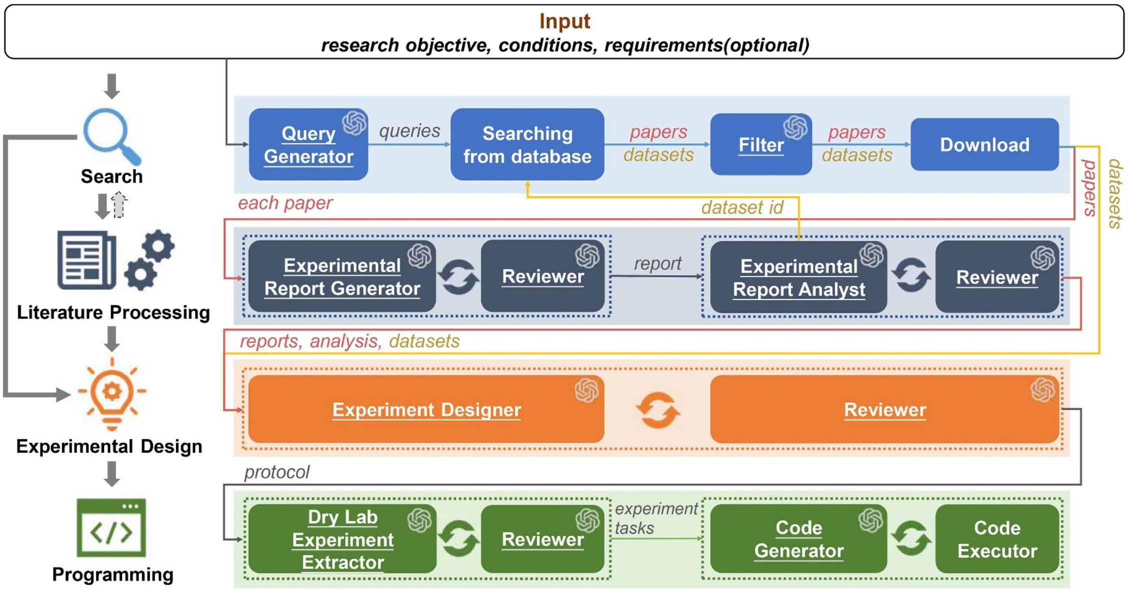


Figure 2 Flowchart of BioResearcher. On the left, the system is divided into four main modules: search, literature processing, experimental design, and programming. The right side illustrates the system's workflow: it initiates with the search module that retrieves pertinent literature and datasets based on the user input. The literature processing module then converts the obtained literature into standardized experimental reports, analyzing them against the research objective, conditions, and requirements. It also works with the search module to evaluate the applicability of the datasets mentioned in the reports. These processed reports, analyses, and suitable datasets are then sent to the experimental design module, which designs an experimental protocol aligned with the research objective. Finally, the programming module extracts the dry lab experimental tasks from the protocol and generates executable code to perform these tasks. The components underlined are agents based on LLMs.

3 BioResearcher

3.1 Framework overview

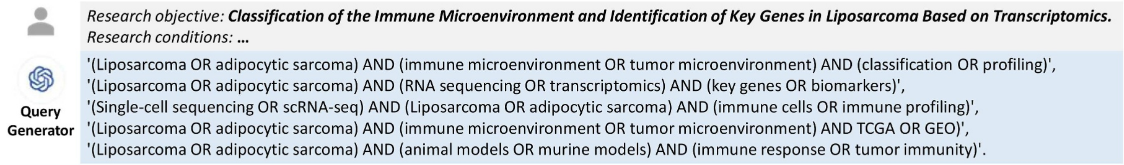
BioResearcher is designed to automate the research process for biomedical studies. Users provide a research objective[4)] and the conditions under which the experiments will be conducted. Users can also specify the research requirements, such as desired experimental steps or outcomes. All user input is in natural language.

To emulate the workflow of human researchers, BioResearcher consists of four primary modules: search, literature processing, experimental design, and programming. As shown in Figure 2, the research process begins with the search module, which comprehends the user input, generates appropriate queries, and searches for relevant research papers and datasets from online repositories. The retrieved literature is then filtered, downloaded, and forwarded to the literature processing module. Here, each research paper is standardized into an experimental report, and each report is analyzed in light of the user-specified objective, conditions, and requirements. This module also interacts with the search module to identify the usability of the datasets mentioned in the reports. The processed reports, analyses, and datasets are then forwarded to the experimental design module, which constructs an experimental protocol. Finally, the programming module extracts a sequence of dry lab experiment tasks from the protocol and generates accurate and executable codes for these tasks.

3.2 Search

Scientific research initiates with a thorough review of related literature. Literature searching is especially important in BioResearcher because, unlike previous studies that use literature surveys to identify research gaps and propose novel ideas, BioResearcher ensures the process is built upon existing knowledge and preserves the reliability of the output. The search module comprehensively explores pertinent literature and datasets throughout various research phases. Its internal procedure is as follows.

[4)] Since our goal is to automate the entire research process, BioResearcher currently only supports studies involving only dry lab experiments. Studies that require wet experiments are out of the scope because wet experiments generally require human operation, e.g., hands-on tasks in a laboratory setting, and cannot be fully automated.



Research objective: ***Classification of the Immune Microenvironment and Identification of Key Genes in Liposarcoma Based on Transcriptomics.***
Research conditions: **...**

'(Liposarcoma OR adipocytic sarcoma) AND (immune microenvironment OR tumor microenvironment) AND (classification OR profiling)',
'(Liposarcoma OR adipocytic sarcoma) AND (RNA sequencing OR transcriptomics) AND (key genes OR biomarkers)',
'(Single-cell sequencing OR scRNA-seq) AND (Liposarcoma OR adipocytic sarcoma) AND (immune cells OR immune profiling)',
'(Liposarcoma OR adipocytic sarcoma) AND (immune microenvironment OR tumor microenvironment) AND TCGA OR GEO)',
'(Liposarcoma OR adipocytic sarcoma) AND (animal models OR murine models) AND (immune response OR tumor immunity)'.

Figure 3 Example of query generation.

(1) Query generation. Searching directly with the user input can yield imprecise results for two reasons. Firstly, the user describes the research objective and conditions in natural language, while most databases use Boolean query logic, making it difficult to retrieve results that completely match the lengthy user input. Secondly, the user intention is provided with key high-level concepts, while the literature may use a more detailed description or synonyms, making it challenging to obtain all relevant materials. Thus, query rewriting is crucial [63]. However, manually crafting effective queries costs a lot of expertise, labor, and time [64]. To address this, the search module in BioResearcher employs an LLM-based query generator agent to create Boolean queries based on the user input. As displayed in Figure 3, the query generator extracts keywords to improve the retrieval accuracy. It also performs synonym expansion (e.g., expanding "Single-cell sequencing" with "scRNA-seq") to improve the retrieval recall. These structured queries allow the module to interpret the research objective effectively and focus on the most relevant materials.

(2) Retrieval. The module interfaces with databases through their APIs, enabling the retrieval of relevant literature and datasets from established repositories such as PubMed Central (PMC), PubMed, and GEO. Additional databases can be integrated as needed to expand the system's capabilities.

(3) Filtration. To retain only the most relevant and useful materials for further stages, our system employs an LLM-based filter agent to filter the returned research articles and datasets. For the research articles, we define a set of criteria detailed in Appendix D (Table D1). The filter examines the titles and abstracts of each article to determine their potential contribution to the research objectives. Each article receives a helpfulness score ranging from one to five. Articles scoring above four are downloaded and forwarded to the literature processing module. For the datasets, the filter assesses its usefulness based on the metadata (i.e., the attached online descriptions) of datasets and assigns a binary usability score. The useful datasets' descriptions are forwarded to the experimental design module.

The search module thus ensures that subsequent stages of BioResearcher are provided with high-quality, targeted resources, establishing a robust foundation for further processes.

3.3 Literature processing

Literature comprehension can be challenging for researchers and LLMs because the research papers are massive, lengthy, and unstructured, with complex logic. To streamline literature comprehension, enhance comprehension, and provide valuable references for experimental design, we introduce the literature processing module. This module first standardizes research papers into highly structured experimental reports and analyzes them systematically. It then interacts with the search module to identify the usability of the datasets mentioned in the reports. Thus, this module operates through two primary phases: report generation and report analysis. Figure 4 illustrates an example of the module in operation.

Report generation. Each biomedical research paper contains experiment-related contents scattered across various sections. We aim to extract and reorganize these contents into a condensed, highly structured experimental report. Doing so brings three advantages. Firstly, the experimental report is shorter than the original paper, enhancing the efficiency of LLMs. Secondly, uniform formatting across reports provides logical structure coherence and format consistency, ensuring the LLMs grasp a big-picture idea of the most commonly acknowledged methods. Finally, the report is modularized with sections focusing on different aspects of the experiments, making it inherently suitable as a unit of analysis and a segment within RAG techniques.

Consequently, we first introduce the hierarchical report generation process in this module, which consists of the following steps, carried out by an LLM-based report generator agent. (1) First-level heading generation. The first-level headings for the experimental report are generated based on the paper's content, establishing the foundation for the report's overall structure. (2) Outline development. A comprehensive outline of the experimental report is developed, guided by the first-level headings and the

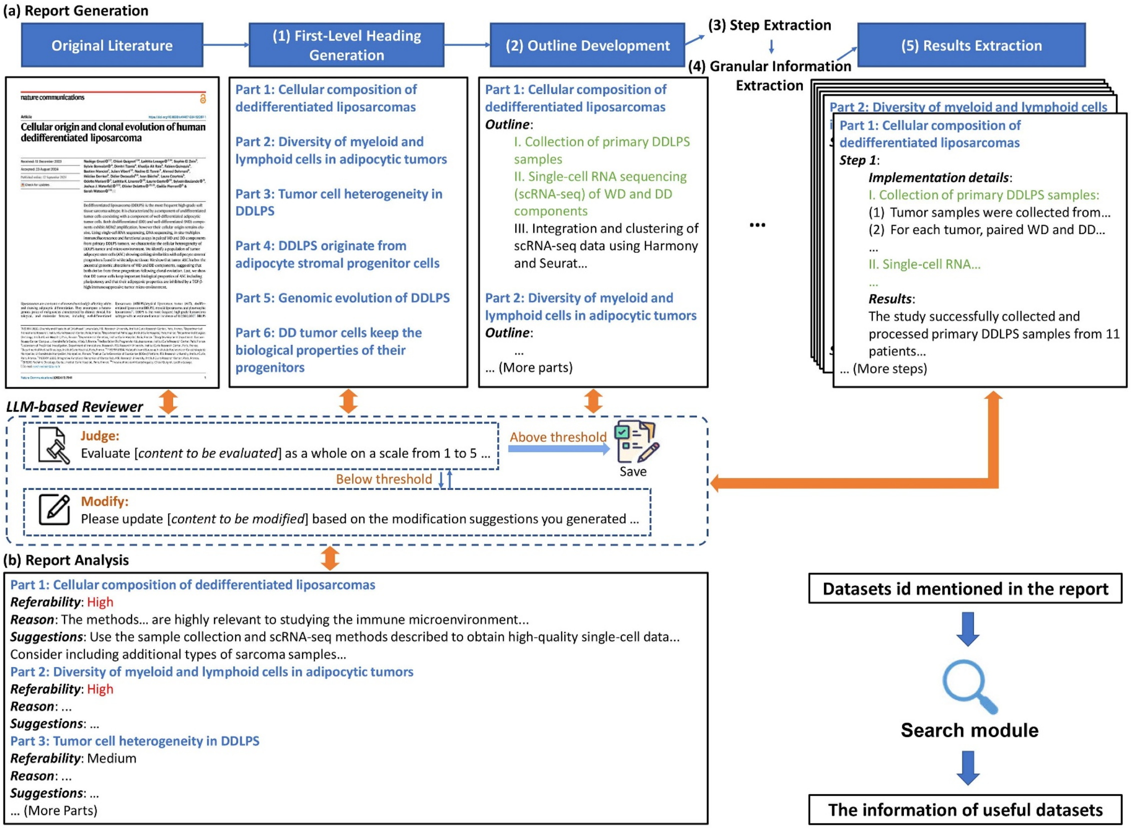

(a) Report Generation
Original Literature
(1) First-Level Heading Generation
(2) Outline Development
(3) Step Extraction
(4) Granular Information Extraction
(5) Results Extraction
nature communications
Cellular origin and clonal evolution of human dedifferentiated liposarcoma
Part 1: Cellular composition of dedifferentiated liposarcomas
Part 2: Diversity of myeloid and lymphoid cells in adipocytic tumors
Part 3: Tumor cell heterogeneity in DDLPS
Part 4: DDLPS originate from adipocyte stromal progenitor cells
Part 5: Genomic evolution of DDLPS
Part 6: DD tumor cells keep the biological properties of their progenitors
Part 1: Cellular composition of dedifferentiated liposarcomas
Outline:
I. Collection of primary DDLPS samples
II. Single-cell RNA sequencing (scRNA-seq) of WD and DD components
III. Integration and clustering of scRNA-seq data using Harmony and Seurat…
Part 2: Diversity of myeloid and lymphoid cells in adipocytic tumors
Outline:
…
… (More parts)
…
Part 2: Diversity of myeloid and lymphoid cells
Part 1: Cellular composition of dedifferentiated liposarcomas
Step 1:
Implementation details:
I. Collection of primary DDLPS samples:
(1) Tumor samples were collected from…
(2) For each tumor, paired WD and DD…
…
II. Single-cell RNA…
…
Results:
The study successfully collected and processed primary DDLPS samples from 11 patients…
… (More steps)
LLM-based Reviewer
Judge:
Evaluate [content to be evaluated] as a whole on a scale from 1 to 5 …
Above threshold
Save
Below threshold
Modify:
Please update [content to be modified] based on the modification suggestions you generated …
(b) Report Analysis
Part 1: Cellular composition of dedifferentiated liposarcomas
Referability: High
Reason: The methods… are highly relevant to studying the immune microenvironment…
Suggestions: Use the sample collection and scRNA-seq methods described to obtain high-quality single-cell data… Consider including additional types of sarcoma samples…
Part 2: Diversity of myeloid and lymphoid cells in adipocytic tumors
Referability: High
Reason: …
Suggestions: …
Part 3: Tumor cell heterogeneity in DDLPS
Referability: Medium
Reason: …
Suggestions: …
… (More Parts)
Datasets id mentioned in the report
Search module
The information of useful datasets

Figure 4 Example of literature processing. (a) Workflow for report generation; (b) example of report analysis.

paper's content, offering a structured framework for detailing experimental procedures. (3) Step extraction. Experimental steps are extracted from the paper and organized according to the generated outline. (4) Granular information extraction. Detailed information about each experimental step is extracted from the paper. This step is critical for refining the experimental report and ensuring that all essential details are included for a thorough understanding of the experimental procedures. (5) Results extraction. Results related to each experimental step are extracted, facilitating the interpretation of outcomes and their relevance to the research objective.

To mitigate potential performance degradation caused by excessively long input contexts [65], the report is divided into sections corresponding to the first-level headings, allowing parallel processing and enhancing efficiency.

Report analysis. Drawing inspiration from the chain of thought (COT) framework [66], which emphasizes step-by-step reasoning, we recognize that analyzing reports in relation to research objectives is a critical step in designing new experiments. Consequently, we introduce a report analysis process following report generation. Specifically, an LLM-based analyst agent evaluates each section of the experimental report for its referability, considering the research objective, conditions, and requirements. The analyst also provides suggestions for references and modifications, akin to proposing innovations on existing methods. Additionally, this module interfaces with the search module to identify usable datasets within the report, using regular expressions to extract dataset identifiers (IDs) for retrieval from the database while bypassing the query generation step. As mentioned above, the usefulness of each dataset is determined based on its description, and useful datasets are then integrated with previously collected ones.

To ensure the quality and accuracy of the generated outputs, an LLM-based reviewer agent is introduced to interact with the experimental report generator and analyst agents at each step. Outputs are refined based on the reviewer's feedback until final approval.

To provide fine-grained retrieval that eliminates irrelevant information, upon a comprehensive review of the relevant literature, we extract and integrate the sections deemed highly referential during the analysis phase, along with their corresponding analysis content from all reports. This consolidated information

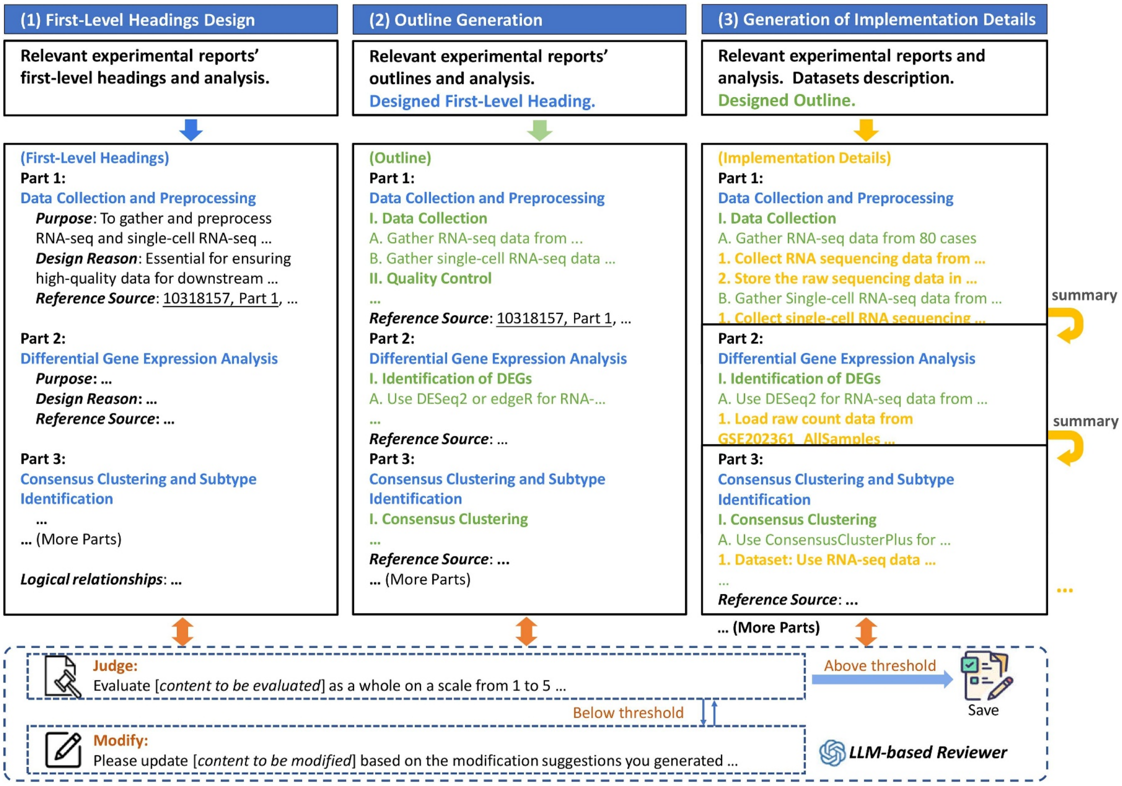

(1) First-Level Headings Design
Relevant experimental reports' first-level headings and analysis.
(First-Level Headings)
Part 1:
Data Collection and Preprocessing
Purpose: To gather and preprocess RNA-seq and single-cell RNA-seq ...
Design Reason: Essential for ensuring high-quality data for downstream ...
Reference Source: 10318157, Part 1, ...
Part 2:
Differential Gene Expression Analysis
Purpose: ...
Design Reason: ...
Reference Source: ...
Part 3:
Consensus Clustering and Subtype Identification
...
... (More Parts)
Logical relationships: ...
(2) Outline Generation
Relevant experimental reports' outlines and analysis.
Designed First-Level Heading.
(Outline)
Part 1:
Data Collection and Preprocessing
I. Data Collection
A. Gather RNA-seq data from ...
B. Gather single-cell RNA-seq data ...
II. Quality Control
...
Reference Source: 10318157, Part 1, ...
Part 2:
Differential Gene Expression Analysis
I. Identification of DEGs
A. Use DESeq2 or edgeR for RNA-...
...
Reference Source: ...
Part 3:
Consensus Clustering and Subtype Identification
I. Consensus Clustering
...
Reference Source: ...
... (More Parts)
(3) Generation of Implementation Details
Relevant experimental reports and analysis. Datasets description.
Designed Outline.
(Implementation Details)
Part 1:
Data Collection and Preprocessing
I. Data Collection
A. Gather RNA-seq data from 80 cases
1. Collect RNA sequencing data from ...
2. Store the raw sequencing data in ...
B. Gather Single-cell RNA-seq data from ...
1. Collect single-cell RNA sequencing ...
summary
Part 2:
Differential Gene Expression Analysis
I. Identification of DEGs
A. Use DESeq2 for RNA-seq data from ...
1. Load raw count data from GSE202361_AllSamples ...
summary
Part 3:
Consensus Clustering and Subtype Identification
I. Consensus Clustering
A. Use ConsensusClusterPlus for ...
1. Dataset: Use RNA-seq data ...
...
Reference Source: ...
...
... (More Parts)
Judge:
Evaluate [content to be evaluated] as a whole on a scale from 1 to 5 ...
Above threshold
Save
Below threshold
Modify:
Please update [content to be modified] based on the modification suggestions you generated ...
LLM-based Reviewer

Figure 5 Demonstration of experimental design.

serves as a reference for the experimental design module.

3.4 Experimental design

Scientific findings in biomedical research frequently face reproducibility issues, wasting resources and time while undermining the credibility of scientific outcomes [67]. A well-designed experimental protocol is crucial for obtaining reliable results and optimizing resource use [68]. Therefore, we develop the experimental design module, which uses an LLM-based experiment designer agent to create scientific and reproducible protocols. We propose a hierarchical learning approach to ensure the rationality of the logical structure of the generated experimental protocols. This method employs the RAG technique aided by the analysis (i.e., using relevant reports' sections with high referability as reference materials), enabling the model to subsequently learn first-level headings independently, then outlines, and then experimental details from the reorganized reports. The design process is structured into three essential steps, as demonstrated in Figure 5.

(1) First-level heading design. The designer begins by reviewing the first-level headings and corresponding analyses of relevant experimental reports, paying particular attention to the reference and modification suggestions. Integrating this information with the research objective, conditions, and requirements, the designer crafts first-level headings for the new protocol. Additionally, for each section, the designer provides a rationale detailing the section's purpose, design rationale, and reference source. This step is crucial as it lays the foundation for the experimental framework, ensuring alignment with the research objective.

(2) Outline generation. The designer then constructs a brief protocol outline, referencing the outlines of pertinent reports and analyses. The designer also includes the sources of reference following the outlines. The generated outline serves as a framework for organizing the protocol.

(3) Generation of implementation details. Finally, the designer generates complete and specific implementation details for each part of the experimental protocol. Relevant sections of experimental reports and corresponding analyses are extracted based on the reference sources provided in the previous step. This information, along with the useful datasets, the protocol outline, and the summaries from earlier sections, is incorporated to produce a detailed protocol. The emphasis on detail and specificity ensures the reproducibility of the experiments, enabling researchers to follow the protocol precisely in subsequent studies.

Furthermore, an LLM-based reviewer agent, like in the literature processing module, is involved at each stage, providing continuous feedback to refine the design. This iterative process ensures the quality and accuracy of the final experimental design, thus contributing to the overall robustness of the research.

3.5 Programming

Programming in biomedical research presents a unique challenge to researchers, necessitating a combination of programming proficiency and domain-specific knowledge. To address this, we propose the programming module, which is crucial in enhancing the reproducibility of experimental designs and automating systems.

To reduce the complexity of coding and debugging, the programming module employs an LLM-based dry lab experiment extractor agent to derive a series of dry experiment tasks from the designed experimental protocol. Each task includes a task ID, a description of the task, and the types and descriptions of the input and output.

Subsequently, this module utilizes an LLM-based code generator agent to create R language code for each task. More programming languages will be supported in future work. To ensure code executability, code execution within a Docker container is employed, which provides a controlled, isolated environment. Execution results, whether error reports or successful outputs, are fed back to the code generator. Based on these results, the code generator determines the next course of action: either further modification of the original code or termination of the generation process. Through this iterative cycle, the system refines the code until it achieves correctness and operational validity.

By systematically bridging the gap between experimental design and execution, this module significantly contributes to the efficiency and efficacy of the research workflow, making it a critical module of the system's overall functionality.

4 Evaluation metrics

Quantitative evaluation metrics are demanded to fully reflect a research assistant's ability to advance the automation of biomedical research. However, no existing study has proposed such metrics. In this context, we propose a comprehensive method for assessing the quality of the resulting experimental protocols and programs.

4.1 Protocol evaluation

We evaluate experimental protocols from five dimensions: completeness, level of detail, correctness, logical soundness, and structural soundness. The definitions and formulas for these five metrics are as follows.

- **Completeness.** Completeness assesses how thoroughly each section of the protocol is described, considering the necessary steps that should be added to achieve the design purpose of this part. The formula for completeness is given by

$$\text{Completeness} = \frac{\sum_{i=1}^{m} n_{\text{es}}^{i}}{\sum_{i=1}^{m} n_{\text{ns}}^{i}} = \frac{n_{\text{ts}}}{n_{\text{ts}} + n_{\text{as}}}, \tag{1}$$

where m represents the number of sections in a protocol. Here, n_{es}^{i} refers to the number of existing steps in the i-th section, while n_{ns}^{i} denotes the total necessary steps for that section. Additionally, n_{ts} indicates the total number of steps in a protocol, and n_{as} represents the number of steps that need to be added.

- **Level of detail.** The level of detail measures the degree to which a protocol provides sufficient information for each step, ranging from 0 (no detail) to 1 (fully detailed).

- **Correctness.** Correctness assesses the proportion of protocol steps that are free from factual errors. Our analysis reveals that protocols with shorter steps tend to exhibit higher correctness scores, as the probability of factual errors increases with longer and more detailed steps. Drawing inspiration from the BLEU [25] metric in machine translation, we also introduce a brevity penalty (BP) for shorter steps. The BP is constrained to a minimum of 0.5, and the formula is as follows:

$$\text{BP} = \begin{cases} 1, & \text{if } l_{\text{steps}} > L, \\ \max(\mathrm{e}^{(1-\frac{L}{l_{\text{steps}}})},\ 0.5), & \text{if } l_{\text{steps}} \leqslant L, \end{cases} \tag{2}$$

Table 1 Grading criteria and error types for R code execution.

Level	Description	Example error types
1	Minor errors	Missing or incorrect file paths, missing necessary libraries or packages, network timeouts
2	Moderate errors	Syntax errors, incorrect function or variable names, data type mismatches
3	Major errors	Parameter mismatches, index out of bounds or invalid index, out of memory
4	Severe errors	Incorrect algorithms or logic, disorganized code structure, key components missing or incorrect

where l_{steps} represents the length of steps, quantified by the number of sentences containing more than six words within each step (this criterion is employed to exclude steps' titles from the count). The parameter L denotes the average number of sentences within a step, calculated to be 4.42 from 315 protocols generated by different methods.

Then, the formula for Correctness is defined as

$$\text{Correctness} = \text{BP} \cdot \frac{n_{\text{cs}}}{n_{\text{ts}}}, \tag{3}$$

where n_{cs} denotes the number of the correct steps.

• **Logical soundness.** Logical soundness evaluates the proportion of steps that are placed in a reasonable order within a protocol. Reasonable steps are those that are logically ordered and appropriately positioned. The formula for Logical Soundness is given by

$$\text{Logical Soundness} = \frac{n_{\text{rs}}}{n_{\text{ts}}}, \tag{4}$$

where n_{rs} denotes the number of reasonable steps.

• **Structural soundness.** Structural soundness evaluates the logical coherence and organizational integrity of a protocol's overall framework, with scores from 0 (completely unsound) to 1 (perfectly sound).

To speed up evaluation, we employ an LLM (GPT-4o) as the judge to evaluate the experimental protocols and obtain the above five metrics independently. For completeness, the LLM generates the additional steps required to achieve the design purpose of each section. For correctness and logical soundness, it evaluates the accuracy and coherence of each step. We then calculate these three metrics using their respective formulas. For the remaining two metrics, level of detail and structural soundness, the LLM directly generates the corresponding scores. Ultimately, the overall score for the experimental protocol is determined by summing the scores of all five dimensions.

4.2 Program evaluation

As a critical component of research automation, the extent to which the Programming module can augment research efficiency deserves attention. As detailed in Subsection 3.5, this module generates tailored code for each dry lab experiment task. To assess its effectiveness, we propose two scoring systems. The first computes the execution success rate, reflecting the percentage of successfully completed tasks per protocol. The second metric assigns error levels to the tasks that remain incomplete, reflecting the severity of errors encountered during code execution. The detailed grading criteria are outlined in Table 1.

5 Experiment

In this section, we design comprehensive experiments to answer the following research questions.

RQ1: How does BioResearcher perform in automating the entire biomedical research process?

RQ2: How does each component of BioResearcher perform in their specific sub-tasks, including the search module (RQ2.1), report generation module (RQ2.2), and experimental design module (RQ2.3)?

5.1 Experiment setup

In our experiments, we utilize the GPT-4o model as the foundational LLM for all the agents in BioResearcher. The temperature settings for various agents within the system are as follows: the query generator is configured with a temperature of 0.7 to introduce a moderate level of variability in the generated queries. Conversely, the reviewer and the LLM used for evaluation are set to a lower temperature of

Table 2 Quality of experimental protocols generated by various systems. The best results are in bold. 'Detail' and 'Structure' refer to 'level of detail' and 'structural soundness', respectively.

Method	Completeness	Detail	Correctness	Logical soundness	Structure	Overall	l_{steps}	$n_{\text{total\_steps}}$
RAG	0.405	0.687	0.483	**0.973**	**0.908**	3.456	1.286	9.167
ReAct	0.364	0.577	0.484	0.963	0.897	3.285	1.237	5.792
Plan and execute	0.380	0.587	0.483	0.965	0.900	3.314	1.194	6.375
BioResearcher	$\mathbf{0.659}_{\uparrow \mathbf{0.254}}$	$\mathbf{0.893}_{\uparrow \mathbf{0.206}}$	$\mathbf{0.895}_{\uparrow \mathbf{0.411}}$	$0.953_{\downarrow 0.020}$	$0.891_{\downarrow 0.017}$	$\mathbf{4.292}_{\uparrow \mathbf{0.836}}$	7.327	33.958

0.1 to ensure more deterministic and consistent evaluations. All other agents operate at a temperature of 0.5, balancing between randomness and determinism to maintain overall coherence and reliability in the agents' outputs. Additionally, the query generator generates five queries for each user input, with a maximum retrieval quantity of 10 per query for each database. The maximum number of interaction rounds for the LLM reviewer in a single session is 6.

Baselines. We evaluate BioResearcher by comparing it with three well-known agent systems. (1) ReAct [69], which integrates reasoning and action within LLMs to effectively manage complex reasoning and decision-making tasks. (2) Plan-and-Execute [27], which utilizes an iterative framework to accomplish tasks through sequential planning and execution. (3) RAG, which employs a naive RAG module to search for relevant content and generates the answer in a single step. Appendix B provides a detailed description of the implementation of these baseline systems.

5.2 Performance of end-to-end automation

We collect eight ongoing research objectives from a biomedical laboratory to ensure that no published work has addressed these research objectives, as detailed in Appendix C (Table C1). Each objective is processed for three runs to minimize randomness. We evaluate three baseline systems and our system for designing and executing experiments for these objectives. We equip the React and Plan-and-Execute systems with four tools: (1) a search tool utilizing the NCBI API[5] to retrieve descriptions of relevant papers, (2) a download tool for acquiring these papers and storing them in a chunked and vectorized format, (3) a search tool using the NCBI API to obtain descriptions of relevant datasets and storing them in a chunked and vectorized format, and (4) a search tool that extracts pertinent content from the resulting vector database.

5.2.1 *Performance by automatic evaluation*

Table 2 presents the average scores for protocols generated by different systems, including completeness, level of detail, correctness, logical soundness, and structural soundness. To minimize single-model evaluation bias, we employed four LLMs (GPT-4o, O3-mini-2025-01-31, Gemini-2.0-Flash, and DeepSeek-V3) for assessment, reporting their average scores across five distinct metrics. The detailed scoring of each model can be found in Appendix F (Table F1). We also calculate the overall performance, the average number of sentences per step (l_{steps}), and the average number of steps per protocol ($n_{\text{total\_steps}}$). (1) The results highlight BioResearcher's exceptional performance in completeness, level of detail, and correctness, surpassing the best baseline by 0.254 (62.7%), 0.206 (30.0%), and 0.411 (84.9%), respectively. (2) Our system is comparable to the best baseline in logical soundness and structural soundness, exceeding 0.89, consistently maintaining a high performance in these aspects. (3) Furthermore, the protocols generated by BioResearcher have significantly more sentences per step and steps per protocol, 5.9× and 4.8× greater than the average performance of different baselines, respectively, indicating the generation of more detailed and comprehensive protocols. (4) The comparative analysis of the three baselines indicates that RAG outperforms the other two iterative agent systems. This superiority can be attributed to the fact that, in the long-context environment characterized by iteration and lengthy retrieved information, the summarization and integration capabilities of the latter two systems degrade, resulting in the loss of substantial amounts of valuable information in the final generated protocol. In contrast, we design tailored workflows to effectively integrate results from multiple modules and steps, thereby preserving critical information throughout the integration process. A case comparison of protocols from the four systems is illustrated in Appendix F (Figure F1).

[5] https://www.ncbi.nlm.nih.gov/home/develop/api/.

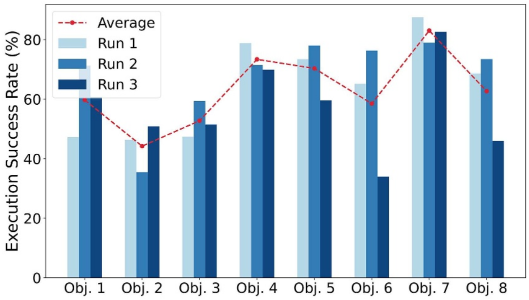


Figure 6 Execution success rate.

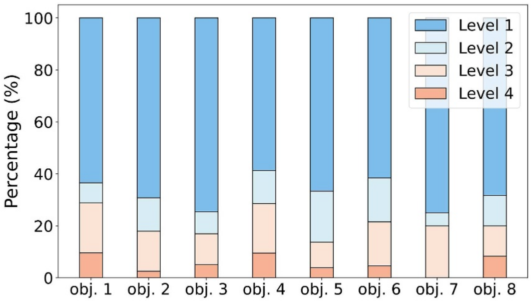


Figure 7 Score frequency distribution.

Table 3 Performance of the LLM judge, as evaluated by human experts. Human consistency is measured using Fleiss' kappa. 'Detail' and 'Structure' refer to 'level of detail' and 'structural soundness', respectively.

	Completeness	Detail	Structure	Logical soundness	Correctness	Avg.
LLM's accuracy (%)	91.1	88.9	75.9	75.0	71.3	80.4
Human consistency	0.66	0.92	0.90	0.88	0.95	0.86

Notably, the first three metrics are crucial for the feasibility of protocols. Consequently, in subsequent code generation and execution, our experiments reveal that the baselines almost invariably fail to produce executable code, with success rates near zero. Our approach, however, achieves an average execution success rate of 63.07% across eight topics, with a maximum of 87.50%, as detailed in Figure 6.

Furthermore, we conduct an error analysis of tasks that failed during code execution. We assess the corresponding code using the criteria in Table 1 and the error messages. Figure 7 illustrates the distribution of error levels. Across eight objectives, the majority of errors are minor errors (Level 1 errors), with an average portion of 67.19%, while severe errors requiring significant manual correction account for only 5.46%. This indicates that even when some tasks are not executed successfully, they can be corrected with minimal human intervention.

These findings suggest that the programming module has significant potential to enhance research efficiency. While some tasks still experience errors, the majority are successfully completed without the need for human intervention. This greatly reduces the time researchers spend on coding and debugging. By further optimizing the module to reduce error rates, research automation can be improved, thereby substantially increasing overall research productivity.

5.2.2 *Quality of automatic evaluation*

To validate the reliability of evaluations conducted by the LLM judge, we engage three domain experts to independently assess the LLM judge's evaluation outcomes. Given the time-consuming and labor-intensive nature of manual evaluation, we employed only GPT-4o, the best LLM currently, as the foundation LLM in BioResearcher to conduct multiple sampling generations on one research objective, resulting in the generation of 18 protocols used in this experiment. This focused approach also reduces evaluation complexity while enhancing assessment accuracy through deeper contextual understanding by experts.

We engage three domain experts to systematically review the LLM judge's evaluation rationales and results for each protocol across five dimensions. This process involves verifying the factual accuracy and logical credibility of the LLM's outputs. Expert judgments are aggregated through majority voting to determine the validity of each step-level evaluation. The final analysis quantifies the LLM's assessment accuracy across five dimensions.

Results are shown in Table 3, demonstrating that the LLM achieves an overall accuracy of 80.4% in protocol evaluations, with exceptional performance in completeness (91.1%). These metrics confirm the model's capacity to deliver comprehensive and efficient automation assessments. We also calculate the inter-expert consistency using Fleiss' kappa to ensure evaluation reliability. Substantial agreement is observed across all dimensions: detail ($\kappa = 0.92$), structure ($\kappa = 0.90$) and correctness ($\kappa = 0.95$) reaches near-perfect consensus, while other dimensions maintain $\kappa > 0.65$. The p-values for consistency across the three dimensions are uniformly below 0.05. This agreement underscores the methodological rigor and

Table 4 Quality of experimental protocols evaluated by human experts. Consistency for the correctness metric is measured using Fleiss' kappa, while consistency for the detail and structure metrics is assessed using Kendall's W. 'Detail' and 'Structure' refer to 'level of detail' and 'structural soundness', respectively.

	Correctness	Detail	Structure
Human experts	0.85	0.87	0.87
Human consistency	0.85	0.68	0.61

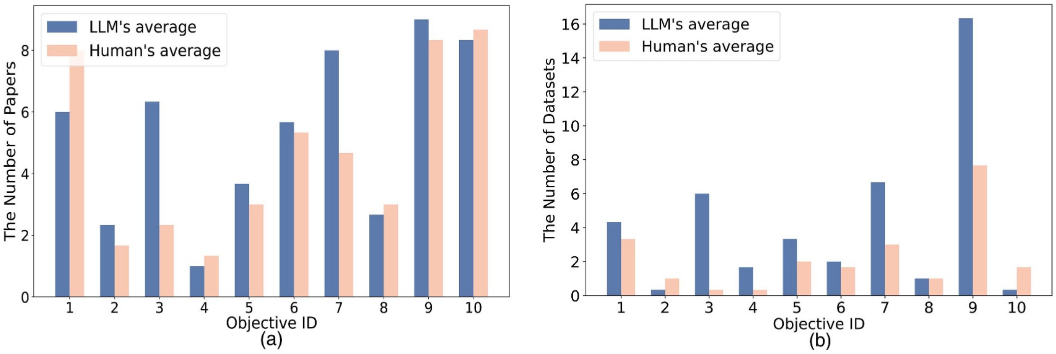


Figure 8 Comparison of the effects of LLM-generated queries and human-generated queries. (a) Comparison of the number of useful papers retrieved by LLM- and human-generated queries; (b) comparison of the number of useful datasets retrieved by LLM- and human-generated queries.

reproducibility of the evaluation framework.

5.2.3 *Performance by manual evaluation*

To further validate the quality of the protocols generated by BioResearcher, we engage three domain experts to independently assess the 18 original protocols generated on one research objective, same as in Subsection 5.2.2, using consistent criteria in Subsection 4.1. Concentrating on one research objective mitigates assessment complexity and enhances accuracy through deeper contextual understanding by the experts.

We employ experts to evaluate these protocols across three dimensions: correctness, detail, and structure, following the criteria in Subsection 4.1. Specifically, the experts systematically evaluate each step of the protocols, determining whether it is correct or not to derive the n_{cs} in (3) for calculating the correctness. For the remaining two metrics, the experts assign a direct score to each protocol. We exclude completeness and logical soundness due to their susceptibility to subjective interpretation, as different researchers may design divergent yet valid experimental approaches. We then analyze inter-rater reliability using appropriate statistical measures. For correctness, which quantifies the proportion of correct experimental steps in a protocol, we employ Fleiss' kappa. For detail and structure (ordinal scoring), we employ Kendall's W. As shown in Table 4, the results in three dimensions are all over 85%, and all consistency metrics exceed 0.6, with correctness over 0.8. The p-values for consistency across the three dimensions are uniformly below 0.05. These results confirm strong consensus among evaluators across all dimensions.

5.3 Performance of search module

5.3.1 *Effect of generated queries*

We conduct a comparative experiment to assess the effectiveness of LLM-generated queries. An LLM and three human participants independently generate five queries for each user input, which specifies a research objective, conditions, and requirements. The evaluation metric is the number of relevant papers or datasets retained after retrieval and filtering. The LLM generates queries three times, with the results averaged, and a similar average is calculated across the three human participants. This experiment, covering ten research objectives listed in Appendix C (Table C2), presents its results in Figure 8.

LLM-generated queries generally outperform human-generated ones across most objectives. (1) In Figure 8(a), which compares the number of useful papers retrieved, the LLM-generated queries show significantly stronger performance in objectives 3 and 7, but the increases are modest on objectives 2, 5, 6, and 9. Human-generated queries perform slightly better in three objectives, suggesting that human

Table 5 Evaluation results of the LLM-based filter agent. Human consistency in paper reviews is measured by Kendall's W, and in dataset reviews, Fleiss' kappa is employed. All p-values are less than 0.05, indicating statistical significance. LLM accuracy is calculated using the ground truth from the majority of human ratings.

	Obj.1	Obj.2	Obj.3	Obj.4	Obj.5	Obj.6	Obj.7	Obj.8	Obj.9	Obj.10	Total
					Paper reviews						
Human's consistency	0.67	0.80	0.84	0.91	0.86	0.80	0.75	0.75	0.80	0.68	0.82
LLM's accuracy (%)	80	73	80	93	93	93	80	80	87	80	84
					Dataset reviews						
Human's consistency	0.66	0.7	0.66	1.00	0.70	0.70	1.00	1.00	1.00	0.73	0.84
LLM's accuracy (%)	80	80	80	80	60	80	100	80	100	100	84

intuition can offer an edge in certain cases. (2) Figure 8(b) presents a similar pattern in dataset retrieval. The LLM outperforms human participants in retrieving useful datasets for eight objectives, particularly objective 3, where the LLM retrieved an average of six datasets compared to the human average of one-third. This indicates that only one human participant successfully generated queries that retrieved one useful dataset. Conversely, for objectives such as objectives 2 and 10, human-generated queries show a slight advantage.

However, a key advantage of LLMs lies in their efficiency and scalability. Unlike human participants, who may take longer to generate queries and may experience fatigue, LLMs can quickly generate multiple queries with minimal effort. Furthermore, LLMs can mitigate performance gaps in challenging objectives by repeating query generation to increase the chances of retrieving relevant papers and datasets. This iterative capability allows LLMs to adapt and improve results, making them highly effective for large-scale or repetitive search tasks.

5.3.2 *Effect of LLM-based filter agent*

To assess the precision of the ratings assigned by the LLM-based filter agent, we engage human reviewers to undertake a parallel assessment using the scoring criteria outlined in Appendix D (Table D1). Both the filter agent and human reviewers evaluate the same set of papers and datasets, with their ratings based solely on each paper's title and abstract or the dataset's description. We use Kendall's W to assess agreement for the ordinal ratings of 150 papers, as it is suitable for measuring concordance in ordinal data. For the binary ratings of 50 datasets, we apply Fleiss' kappa, which evaluates inter-rater agreement for categorical data among multiple raters. These papers and datasets are sampled from the search results of 10 topics listed in Appendix C (Table C2). Higher values in both metrics indicate stronger agreement, enhancing the reliability of the human ratings. A majority voting mechanism establishes the ground truth for human ratings, which serves as the benchmark for calculating the model's accuracy. For papers, those with scores of 4 or higher are classified as useful, while others are deemed not useful, and the accuracy is calculated based on this binary classification. The results of this study are represented in Table 5.

The results indicate strong consistency among human reviewers and notable accuracy of the LLM-based filter. For paper reviews, human consistency, measured by Kendall's W, shows an overall concordance of 0.82, reflecting substantial agreement. Based on the established human ground truth, the filter achieves an average accuracy of 84% in classifying papers as useful or not. The filter's accuracy peaks at 93% for several objectives, underscoring its effectiveness in aligning with human assessments. In dataset reviews, human consistency, assessed using Fleiss' kappa, exhibits strong agreement with a value of 0.84. The filter maintains an average accuracy of 84%, with perfect accuracy in specific objectives, highlighting its reliability in dataset classification. The statistical significance of these results, confirmed by p-values less than 0.05, reinforces the robustness of the filter in achieving high concordance with human evaluations across both domains.

5.4 Performance of report generation

To test the impact of our hierarchical report generation method, we compare its reports against those produced by ReAct, Plan-and-Execute, and a naive single-step LLM approach. Specifically, we standardized 20 papers into an experimental report format using each method. None of the methods are equipped with any additional tools. We prompt an LLM to evaluate the reports across four dimensions: logical

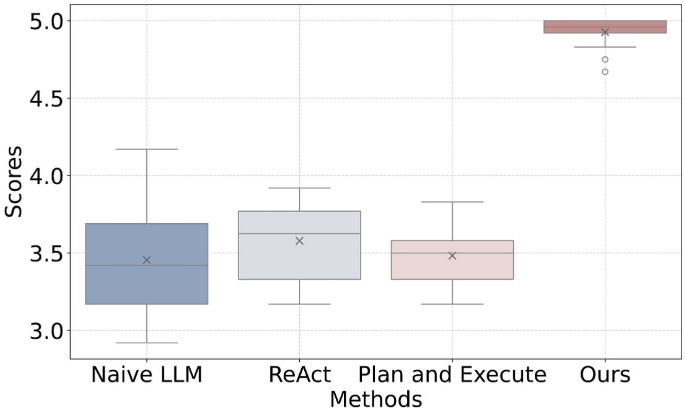


Figure 9 Boxplot of scores for experimental reports generated by different methods.

Table 6 Scores of experimental protocols generated by various methods. The best results are in bold. 'Detail' and 'Structure' refer to 'level of detail' and 'structural soundness', respectively.

Method	Completeness	Detail	Correctness	Logical soundness	Structure	Overall	l_{steps}	$n_{\text{total\_steps}}$
RAG	0.451	0.843	0.506	0.970	**0.941**	3.711	2.005	10.178
ReAct	0.445	0.778	0.516	0.972	0.926	3.636	1.949	9.244
Plan and execute	0.388	0.727	0.544	0.965	0.920	3.544	1.987	8.133
Our method	**0.582**	**0.901**	**0.946**	**0.979**	0.913	**4.321**	6.477	15.289

soundness, level of detail, consistency with the original paper, and readability. The model assigns a score ranging from 1 to 5 for each dimension, with the scoring criteria detailed in Appendix D (Table D2). The final score is calculated as the average across these four dimensions. Figure 9 presents a box plot of the scores for reports generated by the four different methods.

As illustrated in Figure 9, our method consistently outperforms the others, demonstrating significantly higher median values and narrower interquartile ranges, indicative of both superior performance and stability. Conversely, the three baseline models fail to generate high-quality reports. Among them, the Naive LLM shows the greatest score variability, ranging from 2.92 to 4.17, underscoring its instability. In contrast, the results for ReAct and Plan-and-Execute are concentrated around 3.5 points and do not exceed 4 points, reflecting their limited potential. We attribute the poor performance of the baselines to their failure to generate high-quality, long-length outputs when handling the extensive context of an entire paper. Our method, employing a hierarchical approach, enables the model to iteratively generate shorter outputs, which are then integrated into coherent, high-quality reports. These detailed, accurate, and well-structured reports provide excellent references for subsequent experimental design.

5.5 Performance of experimental design

To validate the efficacy of the experimental design module, we construct a comparison experiment against the three baselines introduced in Subsection 5.1. We employ the search module and literature processing module of BioResearcher for 15 research objectives listed in Appendix C (Table C3), thereby obtaining reports and analyses that serve as a knowledge base. The three baselines can invoke search tools to retrieve relevant content from this knowledge base. All experiments are repeated three times, and the evaluation results are averaged and presented in Table 6.

The results in Table 6 show that our methodology surpasses the three baseline methods in all metrics except structural soundness. Specifically, our approach exhibits superior performance in completeness, level of detail, correctness, and logical soundness, exceeding the best baseline scores by 0.131 (29.0%), 0.058 (6.9%), 0.402 (73.9%), and 0.007 (0.7%), respectively. This indicates a more comprehensive, detailed, accurate, and logically sound generation of experimental protocols. Although our method does not lead in the structural soundness metric, it still achieves a commendable score of 0.913, and its overall performance is the highest among all methods. These findings confirm that our hierarchical learning approach, which incrementally designs experimental protocols, effectively addresses the challenges posed by lengthy inputs and outputs, resulting in higher-quality experimental protocols.

Table 7 Scores of experimental protocols generated by BioResearcher without experimental report analyst or reviewer. 'Detail' and 'Structure' refer to 'level of detail' and 'structural soundness', respectively.

Method	Completeness	Detail	Correctness	Logical soundness	Structure	Overall	l_{steps}	$n_{\text{total\_steps}}$
Our method	0.582	0.901	0.946	0.979	0.913	4.321	6.477	15.289
w/o reviewers	0.550	0.884	0.901	0.970	0.918	4.223	4.583	15.333
w/o analyst	0.559	0.901	0.947	0.982	0.919	4.308	6.586	14.978

Referring to Table 2, we observe an improvement in the average performance of all three baseline methods, particularly in completeness (0.373 to 0.428), level of detail (0.666 to 0.783), and correctness (0.481 to 0.522). Their average l_{steps} and $n_{\text{total\_steps}}$ also increase, with the averages rising from 1.239 to 1.980 and 7.111 to 9.185, respectively. The complexity of the research objectives discussed in this section is consistent with those in Table 2, rendering this comparison meaningful. It also underscores the significant role of our report generation in reducing irrelevant information and providing a valuable reference for experimental design.

Additionally, we assess the role of the LLM Reviewers in quality control for the automation of biomedical research. Removing the reviewers from both the literature processing and experimental design modules leads to a decline in performance across most metrics, as shown in Table 7. Specifically, the inclusion of the LLM Reviewers results in an overall score increase of 0.098, with the most notable improvement of 0.045 in Correctness. This indicates that the LLM reviewers effectively identify and correct errors in literature processing and experimental design, thereby enhancing the accuracy of the output. Moreover, our system with reviewers generates more detailed and comprehensive protocols, as evidenced by higher scores in completeness, level of detail, and the average number of sentences per step.

To evaluate the impact of the report analyst agent within the literature processing module, we remove this agent from BioResearcher. Instead, we use the same retrieval model employed in the baselines to extract relevant content from reports as references for protocol generation. As shown in Table 7, the completeness score significantly declines from 0.582 to 0.559. This drop is due to the analyst providing specific references and modification suggestions that enhance protocol completeness. However, in the other four dimensions, removing the report analyst does not negatively affect outcomes and even results in slight improvements. These minor improvements, averaging an increase of only 0.0025, can be considered normal fluctuations. Furthermore, from the perspectives of system interpretability and user-friendliness, the report analyst helps users understand how the system designs new experimental protocols based on relevant materials.

6 Conclusion

In this study, we introduced BioResearcher, an intelligent research assistant that automates the biomedical research process. Utilizing a modular LLM-based multi-agent architecture, BioResearcher addresses the multidisciplinary demands, logical complexities, and performance evaluation challenges of biomedical research. It automates tasks such as literature review, experimental protocol design, and code implementation, significantly improving research efficiency and reducing manual workload. We developed novel evaluation metrics focusing on protocol quality and experimental automation, providing a robust framework for assessing performance. Our results show that BioResearcher designs executable experimental protocols with a high success rate, outperforming existing typical agent systems.

The practical significance of BioResearcher lies in its ability to automate the research pipeline, allowing researchers to focus on strategic decision-making and innovation. This advancement accelerates biomedical discoveries and future developments in automated research systems. By potentially extending its capabilities to wet lab experiments, BioResearcher promises broader applications. This study lays the groundwork for enhancing automated research technologies, contributing to global health and scientific progress.

Acknowledgements Chen LIN was supported by National Natural Science Foundation of China (Grant No. 62432011) and Zhongguancun Academy (Grant No. 20240310).

Supporting information Appendixes A–F. The supporting information is available online at info.scichina.com and link.springer.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

References

1 National Research Council (US) Committee for Monitoring the Nation's Changing Needs for Biomedical, Behavioral, and Clinical Personnel. Advancing the Nation's Health Needs: NIH Research Training Programs. Washington: National Academies Press, 2005

2 Huang Y, Liu Y, Pandey N K, et al. Iron oxide nanozymes stabilize stannous fluoride for targeted biofilm killing and synergistic oral disease prevention. Nat Commun, 2023, 14: 6087

3 Wang P, Song M, Eliassen A H, et al. Optimal dietary patterns for prevention of chronic disease. Nat Med, 2023, 29: 719–728

4 Agarwal A, Mehta P M, Jacobson T, et al. Fixed-dose combination therapy for the prevention of atherosclerotic cardiovascular disease. Nat Med, 2024, 30: 1199–1209

5 Therriault J, Janelidze S, Benedet A L, et al. Diagnosis of Alzheimer's disease using plasma biomarkers adjusted to clinical probability. Nat Aging, 2024, 4: 1529–1537

6 Zheng J, Sun Q, Zhang M, et al. Noninvasive, microbiome-based diagnosis of inflammatory bowel disease. Nat Med, 2024, 30: 3555–3567

7 Sepahi N, Samsami S, Mansoori Y, et al. Development of a novel colorimetric assay for the rapid diagnosis of coronavirus disease 2019 from nasopharyngeal samples. Sci Rep, 2024, 14: 12125

8 Chang V K, Imperial M Z, Phillips P P J, et al. Risk-stratified treatment for drug-susceptible pulmonary tuberculosis. Nat Commun, 2024, 15: 9400

9 Apperloo E M, Gorriz J L, Soler M J, et al. Semaglutide in patients with overweight or obesity and chronic kidney disease without diabetes: a randomized double-blind placebo-controlled clinical trial. Nat Med, 2025, 31: 278–285

10 Douvaras P, Buenaventura D F, Sun B, et al. Ready-to-use iPSC-derived microglia progenitors for the treatment of CNS disease in mouse models of neuropathic mucopolysaccharidoses. Nat Commun, 2024, 15: 8132

11 Gupta R, Srivastava D, Sahu M, et al. Artificial intelligence to deep learning: machine intelligence approach for drug discovery. Mol Divers, 2021, 25: 1315–1360

12 Zhu J, Wang J, Wang X, et al. Prediction of drug efficacy from transcriptional profiles with deep learning. Nat Biotechnol, 2021, 39: 1444–1452

13 Mak K K, Wong Y H, Pichika M R. Artificial intelligence in drug discovery and development. In: Drug Discovery and Evaluation: Safety and Pharmacokinetic Assays. Cham: Springer, 2024. 1461–1498

14 Bizzo B C, Almeida R R, Alkasab T K. Artificial intelligence enabling radiology reporting. Radiologic Clin N Am, 2021, 59: 1045–1052

15 European Society of Radiology (ESR). What the radiologist should know about artificial intelligence—an ESR white paper. Insights Imaging, 2019, 10: 44

16 Li C, Zhang Y, Weng Y, et al. Natural language processing applications for computer-aided diagnosis in oncology. Diagnostics, 2023, 13: 286

17 Nordin N, Zainol Z, Noor M H M, et al. An explainable predictive model for suicide attempt risk using an ensemble learning and Shapley additive explanations (SHAP) approach. Asian J Psychiatry, 2023, 79: 103316

18 Malibari A A. An efficient IoT-artificial intelligence-based disease prediction using lightweight CNN in healthcare system. Meas Sens, 2023, 26: 100695

19 Bom H S H. Exploring the opportunities and challenges of ChatGPT in academic writing: a roundtable discussion. Nucl Med Mol Imag, 2023, 57: 165–167

20 Lingard L. Writing with ChatGPT: an illustration of its capacity, limitations & implications for academic writers. Perspect Med Education, 2023, 12: 261–270

21 Sami A M, Rasheed Z, Kemell K K, et al. System for systematic literature review using multiple AI agents: concept and an empirical evaluation. 2024. ArXiv:2403.08399

22 Gao S, Fang A, Huang Y, et al. Empowering biomedical discovery with AI agents. Cell, 2024, 187: 6125–6151

23 Chen W, Cheng J, Cai Y, et al. The pyroptosis-related signature predicts prognosis and influences the tumor immune microenvironment in dedifferentiated liposarcoma. Open Med, 2024, 19: 20230886

24 Lin C Y. ROUGE: a package for automatic evaluation of summaries. In: Text Summarization Branches Out. Barcelona: Association for Computational Linguistics, 2004. 74–81

25 Papineni K, Roukos S, Ward T, et al. BLEU: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, 2002. 311–318

26 Gao Y F, Xiong Y, Gao X Y, et al. Retrieval-augmented generation for large language models: a survey. 2023. ArXiv:2312.10997

27 Wang L, Xu W Y, Lan Y H, et al. Plan-and-solve prompting: improving zero-shot chain-of-thought reasoning by large language models. In: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics, Toronto, 2023. 1: 2609–2634

28 Lu C, Lu C, Lange R T, et al. The AI scientist: towards fully automated open-ended scientific discovery. 2024. ArXiv:2408.06292

29 Weng Y, Zhu M, Bao G, et al. CycleResearcher: improving automated research via automated review. In: Proceedings of the 13th International Conference on Learning Representations, Singapore, 2025

30 Baek J, Jauhar S K, Cucerzan S, et al. ResearchAgent: iterative research idea generation over scientific literature with large language models. In: Proceedings of the Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies, Albuquerque, 2025. 1: 6709–6738

31 Jin Q, Leaman R, Lu Z. PubMed and beyond: biomedical literature search in the age of artificial intelligence. eBioMedicine, 2024, 100: 104988

32 Wang X, Li J, Wu X, et al. Huatuo-26M, a large-scale Chinese medical QA dataset. In: Findings of the Association for Computational Linguistics: Annual Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics, Albuquerque, 2025. 3828–3848

33 Sudarshan M, Shih S, Yee E, et al. Agentic LLM workflows for generating patient-friendly medical reports. 2024. ArXiv:2408.01112

34 Devlin J, Chang M W, Lee K, et al. BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, 2019. 1: 4171–4186

35 Singhal K, Azizi S, Tu T, et al. Large language models encode clinical knowledge. Nature, 2023, 620: 172–180

36 Zhang H, Chen J, Jiang F, et al. HuatuoGPT, towards taming language model to be a doctor. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Singapore, 2023. 10859–10885

37 Chen Y R, Wang Z Y, Xing X F, et al. BianQue: balancing the questioning and suggestion ability of health LLMs with multi-turn health conversations polished by ChatGPT. 2023. ArXiv:2310.15896

38 Bolton E, Venigalla A, Yasunaga M, et al. BioMedLM: a 2.7B parameter language model trained on biomedical text. 2024. ArXiv:2403.18421

39 Li Q, Yang X Y, Wang H W, et al. From beginner to expert: modeling medical knowledge into general LLMs. 2023. ArXiv:2312.01040

40 Hu E J, Shen Y, Wallis P, et al. LoRA: low-rank adaptation of large language models. In: Proceedings of the 10th International Conference on Learning Representations, 2022

41 Ouyang L, Wu J, Jiang X, et al. Training language models to follow instructions with human feedback. In: Proceedings of Advances in Neural Information Processing Systems, 2022. 35: 27730–27744

42 White J, Fu Q, Hays S, et al. A prompt pattern catalog to enhance prompt engineering with ChatGPT. In: Proceedings of the 30th Conference on Pattern Languages of Programs, Monticello, 2023. 1–31

43 Maharjan J, Garikipati A, Singh N P, et al. OpenMedLM: prompt engineering can out-perform fine-tuning in medical question-answering with open-source large language models. Sci Rep, 2024, 14: 14156

44 Nachane S S, Gramopadhye O, Chanda P, et al. Few shot chain-of-thought driven reasoning to prompt LLMs for open-ended medical question answering. In: Findings of the Association for Computational Linguistics: Conference on Empirical Methods in Natural Language Processing, Miami, 2024. 542–573

45 Wang J, Shi E, Yu S, et al. Prompt engineering for healthcare: methodologies and applications. 2023. ArXiv:2304.14670

46 Boiko D A, MacKnight R, Kline B, et al. Autonomous chemical research with large language models. Nature, 2023, 624: 570–578

47 Nori H, Lee Y T, Zhang S, et al. Can generalist foundation models outcompete special-purpose tuning? Case study in medicine. 2023. ArXiv:2311.16452

48 Park J S, O'Brien J, Cai C J, et al. Generative agents: interactive simulacra of human behavior. In: Proceedings of the 36th annual ACM Symposium on User Interface Software and Technology, San Francisco, 2023. 1–22

49 Wang G, Xie Y, Jiang Y, et al. Voyager: an open-ended embodied agent with large language models. 2024. ArXiv:2305.16291

50 Fernando C, Banarse D, Michalewski H, et al. Promptbreeder: self-referential self-improvement via prompt evolution. In: Proceedings of the 41st International Conference on Machine Learning, Vienna, 2024. 235: 13481–13544

51 Yang C, Wang X, Lu Y, et al. Large language models as optimizers. In: Proceedings of the 12th International Conference on Learning Representations, Vienna, 2024

52 Tang X R, Zou A N, Zhang Z S, et al. MedAgents: large language models as collaborators for zero-shot medical reasoning. In: Findings of the Association for Computational Linguistics, Bangkok, 2024. 599–621

53 Fan Z H, Wei L, Tang J L, et al. AI hospital: benchmarking large language models in a multi-agent medical interaction simulator. In: Proceedings of the 31st International Conference on Computational Linguistics, Abu Dhabi, 2025. 10183–10213

54 Li J K, Wang S Y, Zhang M, et al. Agent hospital: a simulacrum of hospital with evolvable medical agents. 2024. ArXiv:2405.02957

55 Li G. AI4R: the fifth scientific research paradigm. Bull Chin Acad Sci, 2024, 39: 1–9

56 Lim Y, Tamayo-Orrego L, Schmid E, et al. In silico protein interaction screening uncovers DONSON's role in replication initiation. Science, 2023, 381: eadi3448

57 Zhou J, Zhang B, Chen X, et al. Automated bioinformatics analysis via AutoBA. 2023. ArXiv:2309.03242

58 Bran A M, Cox S, Schilter O, et al. Augmenting large language models with chemistry tools. Nat Mach Intell, 2024, 6: 525–535

59 Huang K, Qu Y, Cousins H, et al. CRISPR-GPT: an LLM agent for automated design of gene-editing experiments. 2024. ArXiv:2404.18021

60 Tiukova I A, Brunnsaker D, Bjurström E Y, et al. Genesis: towards the automation of systems biology research. 2024. ArXiv:2408.10689

61 Su H, Chen R, Tang S, et al. Two heads are better than one: a multi-agent system has the potential to improve scientific idea generation. 2024. ArXiv:2410.09403

62 Yang X, Chen H, Feng W, et al. Collaborative evolving strategy for automatic data-centric development. 2024. ArXiv:2407.18690

63 Clark J. Systematic reviewing. In: Methods of Clinical Epidemiology. Berlin: Springer, 2013. 187–211

64 Scells H, Zuccon G, Koopman B, et al. Automatic boolean query formulation for systematic review literature search. In: Proceedings of the Web Conference, Taipei, 2020. 1071–1081

65 Bai Y S, Lv X, Zhang J J, et al. LongBench: a bilingual, multitask benchmark for long context understanding. In: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics, Bangkok, 2024. 1: 3119–3137

66 Wei J, Wang X, Schuurmans D, et al. Chain-of-thought prompting elicits reasoning in large language models. In: Proceedings of Advances in Neural Information Processing Systems, 2022. 35: 24824–24837

67 Begley C G, Ioannidis J P A. Reproducibility in science. Circ Res, 2015, 116: 116–126

68 Larijani B, Tayanloo-Beik A, Payab M, et al. Design of experimental studies in biomedical sciences. In: Biomedical Product Development: Bench to Bedside. Cham: Springer, 2020. 37–47

69 Yao S, Zhao J, Yu D, et al. ReAct: synergizing reasoning and acting in language models. In: Proceedings of the 11th International Conference on Learning Representations, Kigali, 2023