Harnessing Data-Intelligence-Intensive Multi-Agent System for Life Science Research

Yang Liu[1,2,8], Rongbo Shen[1,2,8], Lu Zhou[1,2,8], Qingyu Xiao[1,2], Jiao Yuan[1,2*], Yixue Li[1,2,3,4,5,6,7*]

1, Guangzhou National Laboratory, Guangzhou 510005, China.

2, GMU-GIBH Joint School of Life Sciences, The Guangdong-Hong Kong-Macau Joint Laboratory for Cell Fate Regulation and Diseases, Guangzhou Medical University, Guangzhou 511436, China.

3, Shanghai Institute of Nutrition and Health, Chinese Academy of Sciences, Shanghai 200030, China.

4, Key Laboratory of Systems Health Science of Zhejiang Province, School of Life Science, Hangzhou Institute for Advanced Study, University of Chinese Academy of Sciences, Hangzhou 310024, China.

5, School of Life Sciences and Biotechnology, Shanghai Jiao Tong University, Shanghai 200240, China.

6, Collaborative Innovation Center for Genetics and Development, Fudan University, Shanghai 200433, China.

7, Shanghai Institute for Biomedical and Pharmaceutical Technologies, Shanghai 200032, China.

8, These authors contributed to the work equally and should be regarded as co-first authors.

*Corresponding authors: Jiao Yuan (yuan_jiao@gzlab.ac.cn), Yixue Li (li_yixue@gzlab.ac.cn).

Abstract

Advancements in high-throughput sequencing technologies and artificial intelligence offer unprecedented opportunities for groundbreaking discoveries while posing significant analytical challenges. This study introduces a data-intelligence-intensive scientific research paradigm that synergizes human expertise with AI to facilitate hypothesis-free exploratory research in life science. We propose a multi-agent system (DII-MAS) based on large language models (LLMs), enabling efficient human-agent interaction, agent group management, interdisciplinary knowledge empowerment, and continuous learning. This novel framework is demonstrated through the construction of a human lung cell atlas, showcasing its capability to overcome the limitations of standalone AI applications, improve research efficiency, and adapt to complex life science tasks. This study substantiates three key hypotheses: the collective intelligence workflow can significantly propel life science tasks, proactive interactions within DII-MAS mitigate comprehension biases and incomplete information issues, and continuous learning empowers DII-MAS to make optimal decisions and tool selections. The contributions of this study comprise the delineation of a data-intelligence-intensive research paradigm, the development of DII-MAS, and the introduction of novel evaluation metrics for agent performance. This study underscores the potential of integrating AI with expert knowledge to accelerate discoveries and navigate uncharted territories in life sciences.

I. Introduction

A. Background and Significance

Bioinformatics, as an interdisciplinary field that marries biology, computer science, mathematics, and statistics, has experienced unprecedented growth over the past few decades[1,2], driven primarily by advancements in high-throughput sequencing technologies[3,4] such as genomics[5,6,7,8], transcriptomics[9,10], proteomics[11,12,13] and metabolomics[14,15]. With the plummeting costs and the accelerated proliferation of these technologies, the amount of biological data being generated has exploded exponentially, presenting a wealth of opportunities for groundbreaking discoveries[16,17]. This torrential influx of data has fundamentally revolutionized our understanding of life sciences, ranging from fundamental molecular biology[18] to practical clinical applications and the burgeoning domain of personalized medicine[19,20].

In recent years, the scientific research landscape in life sciences has been undergoing a profound shift[21,22], propelled by the confluence of two major factors: the exponential growth of complex and interconnected biological data and the rapid advancement of artificial intelligence (AI) technologies. Despite the significant strides achieved in merging bioinformatics with AI technologies, current life science research still faces several critical challenges[23,24]. Firstly, integrating the strengths of AI with big biological data to propel advancements in life sciences research constitutes an ongoing and arduous pursuit[25,26]. Many researchers in the field are limited by a single area of expertise, but lack interdisciplinary knowledge and the ability to conduct cross-disciplinary scientific research. The effective application of AI in bioinformatics often requires a deep understanding of both the biological domain knowledge and advanced computational techniques, therefore there is a need for enhanced interdisciplinary collaboration to bridge the domain gap and unlock the AI-driven

analytical capabilities for life science research[27]. Secondly, the rapid accumulation of high-dimensional and multimodal biological data, characterized by large-scale, complexity, diversity, noise, and contextual dependencies, necessitates the development and application of advanced AI technologies. AI must not only enable scalable, adaptable, and interpretable processing of big biological data, but also promote synergistic integration with expert knowledge, thereby enhancing overall intelligent capacity.

B. Research Statement

In light of the revolutionary progress in AI technology and the exponential growth of biological data, this study delineates a data-intelligence-intensive scientific research paradigm, depicted in Fig. 1a. This paradigm integrates human expertise with AI capabilities in a cohesive collaboration to form collective intelligence[28], that leverage big biological data to propel hypothesis-free exploratory research driven by a data-intelligence-intensive methodology. This novel scientific research paradigm aligns with the future trend of life science research, characterized by data-intensive and computation-intensive[29] in big biological data processing, and knowledge-intensive and intelligence-intensive[30,31] in the convergence of interdisciplinary. Within this paradigm, human expertise continually enhances and refines AI, while AI, in turn, leverages multidisciplinary knowledge fusion and big biological data mining to generate novel insights, thereby improving our understanding of life sciences. By leveraging their complementary strengths, life sciences research gains the ability to traverse uncharted territories of knowledge, generating testable hypotheses, delineating novel research directions, and refining experimental designs to accelerate discoveries, thereby revealing previously unexplored associations and emergent characteristics within complex biological systems.

The data-intelligence-intensive scientific research paradigm presents a promising perspective for addressing contemporary challenges in the life sciences. It embeds knowledge from diverse scientific disciplines into AI agents, who, as integral members of a collective intelligence, actively participate in life science research workflows, facilitating interdisciplinary collaboration and overcoming long-standing knowledge barriers. Moreover, this novel research paradigm, leveraging the complementary integration of expert knowledge with AI[32,33], accelerates life science workflows by feature extraction from big biological data, proposing innovative scientific hypotheses, optimizing research designs, orchestrating computational resources for high-performance computing, and instituting fair and objective evaluation metrics.

In the data-intelligence-intensive scientific research paradigm, both human experts and AI agents are core participants in scientific investigations. Thus, we propose a multi-agent system[34,35,36] (DII-MAS) based on large language models[37,38,39,40] (LLMs). DII-MAS utilizes the potent natural language processing capabilities of LLMs and their broad perspective on scientific issues to enable interdisciplinary knowledge embedding, interaction with human experts, inter-agent communication, and agent self-reflective learning, thereby forming a high-quality collective intelligence through tight collaboration between multiple agents and human experts, conducting data-intelligence-intensive investigations in life sciences. The architecture of DII-MAS comprises multiple agents with diverse roles, a coordination layer, and the environment, depicted in Fig. 1b. Agents, the core components of DII-MAS, employ LLMs as their controlling nucleus and exhibit a high degree of autonomous capability[41,42,43]. They are capable of perceiving environmental information, deciphering complex natural language instructions, conducting cognitive reasoning, and executing actions through tool invocations. The coordination layer bridges the agents and the environment, while manages and orchestrates the behaviors of agents. Within the coordination layer, we devise an agent group management

strategy, an effective human-agent interaction mechanism, a shared interdisciplinary knowledge database, and continuous learning strategies for agents, thereby augmenting the capabilities of DII-MAS. The environment, as an external context for DII-MAS, influences the behaviors of agents, while the actions of these agents in turn modify the state of the environment. Initiated by insights from LLMs, DII-MAS first decomposes life science tasks into modular, hierarchical sub-tasks, then configures agent groups and specifies the roles of agents within each group according to subtask characteristics, alongside formulating rules for task allocation and scheduling. Thereafter, researchers collaborate with the planning agent groups to generate more explicit execution plans for the respective sub-tasks. Finally, according to the generated execution plans, the action execution agent groups proceed gather necessary resources from the environment and sequentially accomplish each execution plan.

Guided by the data-intelligence-intensive scientific research paradigm, this study employs DII-MAS to investigate the construction of a human lung cell atlas[44], aiming to substantiate the following scientific hypothesis: Firstly, the workflow orchestrated by a collective intelligence planning process, involving collaboration between researchers and DII-MAS, can propel life science tasks to achieve remarkable results, significantly enhancing research efficiency and overcoming limitations[45] inherent to standalone LLM applications. Secondly, proactive interactions within DII-MAS can mitigate comprehension biases in LLMs pertaining to prompts, context, and both long-term and short-term memory, as well as address incomplete information issues during action execution, thereby enabling adaptability to complex and dynamic tasks in life sciences. Thirdly, strengthened by multidisciplinary knowledge empowerment and continuous learning of its agents, DII-MAS can make better decisions and select optimal tools for executing life science tasks.

C. Main Contributions

The main contributions of this study are enumerated as follows:

a) This study delineates a data-intelligence-intensive scientific research paradigm in life sciences, enabling hypothesis-free exploratory research.

b) This study anticipates the future trend of life science research, characterized by data-intensive and computation-intensive in big biological data processing, and knowledge-intensive and intelligence-intensive in the convergence of interdisciplinary.

c) This study proposes a multi-agent system (DII-MAS) based on LLMs, which forms a high-quality collective intelligence through close collaboration between multiple agents and human experts to conduct data-intelligence-intensive life science research.

d) To augment the capabilities of DII-MAS, this study devises an agent group management strategy, an effective human-agent interaction mechanism, a shared interdisciplinary knowledge database, and continuous learning strategies for agents.

e) Through the application of DII-MAS in constructing a human lung cell atlas, this study substantiates the feasibility of the data-intelligence-intensive scientific research paradigm and demonstrates the capabilities of DII-MAS.

f) This study proposes a novel suite of evaluation metrics for assessing agent performance across task planning, human-agent interaction, and the quality of code generation.

II. Literature Review

A. The development trend of bioinformatics

The evolution of bioinformatics, tracing back to the mid-20th century, has lately entered into a phase of accelerated development, catalyzed by the successful completion of the Human Genome Project[46,47,48,49] and the groundbreaking advancements in high-throughput sequencing technologies[3,4,50,51]. The explosive growth of high-throughput sequencing data, exemplified by genomics[5,6,7,8], transcriptomics[9,10,52], proteomics[11,12,13], and metabolomics[14,15], has provided unprecedented data resources for life sciences research. Driven by the data-intensive research paradigm, researchers have developed more efficient methods for processing and analyzing large-scale biological datasets to elucidate the structures and functions of biomolecules, cells, tissues, organs, and entire organisms, encompassing gene regulatory networks[53,54], epigenetic modifications[55,56], cellular heterogeneity[57,58], evolutionary relationships[59,60], and disease mechanisms[61,62,63]. In precision medicine[19,20,64], the integration of genomic, cellular phenotype, and clinical data enables multi-modal data analyses[65,66], advancing applications in disease prevention, diagnosis, personalized treatment planning, and drug response prediction. Recently, with the breakthroughs in AI technology, machine learning and deep learning methods have become paramount tools in large-scale biological data analysis, excelling in handling complex nonlinear relationships and demonstrating substantial potential in microscopy image analysis[67,68], protein structure prediction[69,70,71,72], deciphering gene regulatory networks[53,54,73,74], biomarker detection for diseases[75], diagnostic and prognostic assessments[76], and drug discovery and design[77,78,79].

To address the mounting volume of biological data and comprehensively unravel complex life systems, the progression of bioinformatics evolves advancements in technology, computational methodology innovation, and interdisciplinary collaboration, propelling the development of life sciences

through deeper analysis and understanding of biological data. For instance, with the maturation and cost reduction of single-cell sequencing technologies, single-cell multi-omics[80,81] studies integrate multi-omics data in spatiotemporal dimensions, elucidating the fundamental laws governing life systems across different molecular levels (i.e., genetic, transcriptional, proteomic, and metabolic), thereby enabling a finer dissection of cellular heterogeneity, intercellular interactions, and dynamic cellular processes, which fosters novel discoveries in precision medicine, tumor biology, and developmental biology. Third-generation sequencing technologies[82], with their high sensitivity, long reads, and real-time sequencing capabilities, generate full-length transcript sequencing data, facilitating rapid epigenetic modification identification, and contributing to the study of alternative splicing[83] and non-coding RNA structures[84,85,86], thereby enhancing our understanding of gene transcription regulation, non-coding RNA biology, and the mechanisms underlying disease development. AI technologies are becoming increasingly embedded in life science research, capable of handling complex pattern recognition, big data analytics, and predictive modeling, enhancing accuracy and efficiency from omics data analysis to protein structure prediction, to disease diagnostics and drug development. To enhance the efficiency and reproducibility of bioinformatics research, automated workflows provide standardized data processing, elastic resource allocation with distributed computing, and workflow management and monitoring, thereby reinforcing the rigor and objectivity of scientific investigations.

B. Large Language Models

Large language models (LLMs) are deep learning architectures based on the Transformer[87], featuring stacked multi-layered structures and positional encoding, which leverage self-attention mechanisms to process input text

sequences in parallel, thereby efficiently capturing long-range dependencies and learning linguistic structures and patterns. LLMs typically undergo pre-training on vast amounts of unlabeled text data, acquiring general linguistic patterns and structures. Following the pre-training phase, LLMs can be fine-tuned to specific tasks such as question answering, text classification, or text generation, tailored to the precise requirements of the given application. LLMs, with their massive parameters and deep network architectures, excel at understanding complex contextual relationships and generating coherent, high-quality text. With technological progress, some LLMs incorporate multimodal learning capabilities, enabling comprehension of data such as images and audio in addition to text, and thereby achieving cross-modal information processing and synthesis.

The proposal of Transformer[87] architecture in 2017 marked the dawn of the era of LLMs. In 2018, the unveiling of BERT[88] by Google and GPT-1[89] by OpenAI introduced pre-training techniques, which substantially improved the performance capacities of LLMs. The successive releases of the GPT series[90,91] have propelled the rapid advancement of LLMs, with model parameters swiftly surging into trillions. Particularly, ChatGPT[90] and GPT-4[91] have demonstrated unparalleled capabilities in language generation and comprehension, achieving remarkable performance across diverse applications such as question answering, text composition, code generation, and conversational system[92,93,94]. Subsequently, LLMs integrated multimodal processing capabilities, enabling joint modeling of diverse data types such as images and text, and thereby offering enhanced visual generation abilities (e.g., Sora[95]).

While LLMs have demonstrated remarkable capabilities, they also come with inherent limitations[45]. The output of LLMs is highly contingent upon training data, thereby inheriting biases, factual inaccuracies, and illusion issues from biased,

erroneous, or incomplete training datasets. While capable of handling and generating sophisticated language structures, LLMs, as statistical prediction engines, lack the comprehension, common sense, autonomous planning, and self-learning capabilities inherent to humans, rendering them potentially inadequate in tasks requiring profound understanding or logical reasoning. Owing to the complexity of LLMs, the decision-making processes of LLMs tend to be opaque, rendering explanations for specific predictions or generations elusive.

C. AI Agent

The origins of AI agent development can be traced back to the emergence of expert systems, which leveraged rule-based knowledge databases to mimic human experts in solving domain-specific problems. Subsequently, reinforcement learning[96,97] emerged as a vital machine learning paradigm integrated into AI agents, offering a pathway for AI agents to learn through iterative interactions with their environments. In recent years, the revolutionary advancements in deep learning technologies, highlighted by the introduction of LLMs such as the GPT series, have empowered LLM-based agents with formidable natural language processing capabilities. Agents[41,42,43,98], capable of autonomously perceiving their environment, cognitive reasoning, decision-making, and executing actions through tool invocation, have emerged as a highly promising direction in the pursuit of general artificial intelligence. Specifically, an AI agent comprises four modules: the perception module for gathering environmental information, the cognition and decision-making module to analyze inputs and devise action strategies, the memory module to archive knowledge and past behaviors, and the action module to implement decisions by manipulating tools to impact the environment. AI Agents execute tasks through a perceptual-decisional-action loop, subsequently engaging in

reflective evaluation of the results, which informs learning to acquire knowledge, stored in the memory module, thereby optimizing future decision-making and actions. Agents demonstrate human-like intelligence behavior[99] through autonomous environmental perception, decision-making process, and an aptitude for learning and adapting via dynamic environmental engagements.

Prominent existing single-agent frameworks such as AutoGPT[100], BabyAGI, and HuggingGPT[101] facilitate task planning and tool invocation, thereby accomplishing user-defined objectives effectively. For instance, the Coscientist[102] system, powered by GPT-4, autonomously conducts chemical research, involving automated experiment planning and execution, along with predicting and refining reaction pathways through sophisticated modeling capabilities. Researchers interact with the Coscientist system to accomplish six pivotal tasks: planning synthesis routes of known compounds using public data, efficiently retrieving and navigating hardware documentation, executing advanced directives in cloud labs, precisely controlling liquid handling equipment, multitasking across multiple hardware modules for complex operations, integrating diverse datasets and analyzing past experimental data for optimization problem-solving. For another example, the ChemCrow[103] system, driven by LLMs，autonomously plans and executes tasks including the synthesis of organic compounds, drug discovery, and material design. By incorporating tools such as web searches, literature exploration, and interactive coding interfaces, the ChemCrow system augments its capability to acquire and comprehend external information, thereby showcasing the potential for enhanced scientific task management. While LLMs have propelled the evolution of AI agent technologies, single-agent systems still grapple with inherent limitations. Firstly, the capabilities of AI agents are confined by the prowess of their underlying LLMs，which can lead to biased, hallucinatory, erroneous, or outdated action decisions during complex tasks, often due to the quality of training data. When dealing with extensive texts, iterative

conversations, or complex contexts, AI agents may misconstrue user intentions and offer partially relevant responses, hampered by the finite memory retention of LLMs and limitations in comprehending prompt cues and contextual associations. Secondly, while individual agents can adopt multiple roles, they often rapidly converge onto a specific viewpoint influenced by prompts or earlier conversation turns, thereby missing out multiperspective and holistic abilities for problem analysis. Lastly, single-agent systems typically decompose tasks into linear workflows, resulting in inefficient execution and inadequate responsiveness to dynamically changing environments.

LLM-based single-agent systems focus on internal reasoning and decision-making processes as well as interactions with the external environment. Inspired by human collaboration in tackling complex tasks, LLM-based multi-agent systems emphasize diverse agent groups, inter-agent interactions, and collective decision-making protocols. Compared to single-agent systems, multi-agent systems orchestrate a consortium of diverse agents with distinct capabilities and roles, simulating intricate real-world scenarios via inter-agent cooperation, harnessing the collective intelligence born of their individual strengths to tackle more dynamic and intricate tasks such as software development, scientific experiment, and embodied AI.

Recently, the open-source community has witnessed the rise of several prominent multi-agent frameworks[104], such as MetaGPT[105], CAMEL[106], and AutoGen[35]. MetaGPT embeds human workflow procedures into the actions of LLM-based agents, employing structured outputs and a publish-subscribe mechanism, effectively resolving information redundancy in multi-agent conversations and mitigating the prevalence of illusions in complex tasks. CAMEL leverages prompt engineering techniques to delineate the roles of agents, facilitating autonomous collaboration between them, and directing agent conversations to progress in alignment with mission objectives. AutoGen

emerges as a highly customizable multi-agent framework, enabling researchers to customize the LLMs of agents, communication protocols and workflow management, thereby tackling intricate tasks. However, multi-agent systems also face some challenges. Fundamentally, the design and implementation of multi-agent systems entail heightened complexity, requiring meticulous attention to issues of inter-agent communication, collaboration, and conflict resolution mechanisms. Additionally, multi-agent systems encounter sophisticated interaction patterns in the dynamic contexts, which can lead to information overlaps and non-productive discourse, ultimately hindering the effectiveness of task execution. Furthermore, multi-agent systems are characterized by significant computational resource consumption and a heightened cost associated with customization.

III. Results

A. Advancements and architecture of Proposed multi-agent system

Architecture

Within the data-intelligence-intensive scientific research paradigm, where both human experts and AI agents are core participants in scientific investigations, we propose a multi-agent system (DII-MAS) leveraging LLMs to collaborate with human experts in the field of life sciences research. The architecture of DII-MAS, depicted in Figure 1b, comprises agents, a coordination layer, and the environment. Agents, the core components of the system, are empowered by LLMs as their controlling nucleus, exhibiting a high degree of autonomous capability. They are capable of perceiving environmental information, deciphering complex natural language instructions (such as the prompts from

experts in the environment), conducting cognitive reasoning, and executing actions through tool invocations. Agents can act various roles with specific responsibilities, such as information acquisition, task planning, code generation, task execution, and etc. The coordination layer serves as a bridge for agents to perceive environmental information and feedback to influence the environmental state, while harmonizing the behaviors of agents such as task planning and allocation, resource scheduling, conflict resolution, collaborative strategies. The coordination layer also administers shared interdisciplinary knowledge databases, human-agent interfaces, inter-agent communication protocols, and continuous learning strategies. The environment, as an external context for multi-agent systems, influences the behavior of agents, while the actions of these agents in turn modify the state of the environment. Specifically, the environment comprises elements from both the real world, such as users, instruments, devices and tools, and the virtual world, including datasets, documents, knowledge databases, and software.

Advancements

The proposed DII-MAS employs an innovative design approach that emphasizes interdisciplinary collaboration, shared knowledge resources, and continuous interactive feedback, devising novel strategies in efficient human-agent interaction, agent group management, interdisciplinary knowledge empowerment, and continuous learning. Firstly, to address the complexity and dynamism of life science problems, we devise user-friendly, highly real-time human-agent interaction mechanism for DII-MAS, aimed to alleviate the misunderstandings of agents related to prompts, context, and both long-term and short-term memory, alongside issues of incomplete information during action execution. The human-agent interfaces support natural language interactions and immediate feedback, enabling human experts to promptly

direct agent activities and assimilate analytical insights from agents, thereby establishing a tightly coupled human-agent collaborative workflow. Secondly, to enhance collaboration efficiency, we devise agent group management protocols for DII-MAS that group agents strategically to reduce redundant information exchange and inconsequential interactions. Thirdly, acknowledging the interdisciplinary nature of bioinformatics, we establish a shared knowledge database within DII-MAS, encoding domain-specific expertise from diverse fields into a format intelligible to the agents. Empowered by interdisciplinary knowledge, agents can dynamically retrieve relevant information from the shared knowledge database to formulate rational plans for task-specific requirements, thereby mitigating biases and illusions prevalent in LLMs during reasoning and decision-making. Lastly, to swiftly accommodate emerging and challenging life science research tasks, we devise continuous learning strategies for DII-MAS. Specifically, the agents in the system engage in both self-reflective learning[107] from their own behaviors and competitive knowledge acquisition with fellow agents, progressively refining their decision-making and execution capabilities over time to enhance collective intelligence. The aforementioned innovative design enhances the multi-agent system in terms of language intelligence, social intelligence, knowledge empowerment, and learning intelligence.

B. Application for the construction of lung cell atlas

Task description and workflow

In recent years, a series of single-cell sequencing studies have dissected molecular profiles of cells in both normal and diseased tissues, and the rapidly advancing single-cell sequencing technology has made it feasible to establish human cellular atlas (HCA)[108], which was listed among the seven notable

technologies in 2024 by the scientific journal Nature[109]. A comprehensive cellular atlas of human lung tissue systematically unveils molecular distinctions among various pulmonary cell populations, poised to revolutionize our understanding of cells in healthy and diseased lungs, thereby offering crucial insights for elucidating respiratory disease pathogenesis, vaccine development, immunotherapy, and tissue regeneration. Several independent studies have embarked on constructing a cellular atlas for lung tissue. However, limitations in experimental techniques, sequencing methodologies, sample selection, and cell quantity constrain comprehensive and unbiased capture of the diversity and heterogeneity inherent to lung tissues. Consequently, integrating datasets from these independent studies in an unbiased manner to construct a comprehensive cellular atlas of lung tissue would address these limitations and further facilitate the identification of variability patterns across different demographics, such as age, gender, BMI, smoking status, and etc.

This study employs DII-MAS to construct a human lung cell atlas, thereby validating the capability of DII-MAS in handling large-scale single-cell sequencing data analysis task. Referencing the HLCA study[44], the workflow for this task is collaboratively generated by researchers and DII-MAS, which is segmented into four sub-tasks: dataset integration and batch effect correction, hierarchical clustering and differential expression analysis of the integrated dataset, cell type annotation, and variance analysis for technical and biological covariates, as shown in Figure 2a. DII-MAS adopts a consistent procedure for each sub-task, comprising coarse-level planning, fine-level planning, and action execution, as depicted in Figure 2b. Specifically, for each sub-task objective, DII-MAS formulates an initial plan, which then iteratively refined via extensive human-agent interactions, guided by researchers feedback informed by insights and experience. During the action execution of plan, researchers can intervene via human-agent interaction to correct any erroneous actions and behaviors. Comprehensive workflow and implementation details for

constructing the human lung cell atlas are in the Experimental setup section of Methodology.

System evaluation

The evaluation of DII-MAS focuses on the following three aspects: task planning, human-agent interaction, code quality.

The task planning evaluation assesses the ability of DII-MAS in decomposing complex tasks and formulating execution plans, with LLMs serving as the control group. We evaluate both the system and the control group based on metrics including clarity, technical feasibility, structurality, comprehensiveness, and rationality of method selection. To ensure objectivity in the evaluation, we employ GPT-4 to score the execution plans generated by each group. For the hierarchical clustering sub-task within the overall workflow, the evaluation results of DII-MAS and the control group are shown in Figure 2c. Compared with the control group using GPT-4 and GPT-3.5, DII-MAS achieves the best performance in each metric. It is further substantiated that execution plans generated through collective intelligence of DII-MAS and researchers exhibit higher implementability in scientific research contexts, compared to standalone LLMs.

The human-agent interaction evaluation aims to assess the user-friendliness during human-agent collaboration and effectiveness in achieving objectives, with AutoGen serving as the control group. In the control group, we configure agents with identical roles and communication protocols, and separately test the performance of AutoGen under two distinct modes, i.e., 'Never' and 'Always'. The evaluation of DII-MAS and the control group, based on metrics including the frequency of user interaction, the frequency of internal agent interaction,

and the task completion rate, yields quantifiable results through direct counting and statistical analysis. For the variance analysis sub-task, the evaluation results between DII-MAS and the control group are illustrated in Figure 2d. Notably, the control group instance, AutoGen in 'Never' mode, fails to accomplish the sub-task. In comparison to AutoGen in 'Always' mode, DII-MAS completes the sub-task with fewer rounds of human-agent interactions, highlighting a higher level of automation. This also substantiates the efficacy of our designed human-agent interaction mechanism, enabling DII-MAS to effectively tackle complex and dynamic tasks in life sciences. Besides, the dynamic interaction interface enables DII-MAS to proactively initiate conversation when encountering indecision, thereby preventing task interruptions or failures due to misunderstanding or inadequate information.

Code quality evaluation is conducted to assess the system's understanding of execution plans, with the control group comprising code generated by LLMs, and code from reference study that serve as the de facto standard. We assess the code generated by both DII-MAS and the control group according to execution plans, based on metrics including accuracy, efficiency, completeness, readability, logicality and portability. The evaluation methodology employs GPT-4 to score the code in each group. For the hierarchical clustering sub-task, evaluation results compare DII-MAS with the control group, as shown in Figure 2e. Notably, the code from reference study, that is the de facto standard, achieves the highest performance across all evaluation metrics. DII-MAS outperforms GPT-4 across all metrics except readability, evidencing a more comprehensive understanding of execution plans.

Details on the evaluation metrics and the GPT-4 assessment approach are provided in the Evaluation Metrics section of Methodology.

C. Features of lung cell atlas

Data integration, Hierarchical clustering and DEA

Conforming to the human cell atlas workflow, DII-MAS begins the sub-task of integration and batch effect correction across multiple independent datasets. Utilizing the scANVI[110] algorithm, DII-MAS accomplishes dimensionality reduction to embed cells from these datasets into a unified feature space. The scANVI algorithm mitigates batch effects arising from variations in sampling techniques and sequencing technologies across datasets, enabling cells of the same type to cluster in the feature space, thereby facilitating downstream analyses and visualizations.

Subsequently, DII-MAS conducts the sub-task of hierarchical clustering on the embeddings of the integrated dataset to identify distinct cell populations. By clustering cells in a hierarchical manner from coarse to fine resolutions, this approach enables researchers to identify cell populations at various levels and decipher the intricate relationships between them. Specifically, DII-MAS performs a four-level hierarchical clustering to categorize all cells within the dataset into distinct clusters at each level. Figures 3a-c illustrate the UMAP visualizations of the hierarchical clustering results from level 2 to level 4. To compare with the original annotations, Supplementary Figures 1a-c exhibit cell types from levels 2 to level 4 based on the original cell type nomenclature, while Supplementary Figure 2a illustrates the inter-layer relationships of these original labels. These visualizations reveal that the planning and execution by DII-MAS successfully replicate the data integration and hierarchical clustering as reported in the original study. Furthermore, Supplementary Figures 1a-c reveals that the batch effects between originally independent datasets are effectively mitigated following integration. In the original annotations of the datasets, a considerable number of cells show unfilled categories (prefixed with

numbers and inherited from upper levels during preprocessing) from level 2 to level 4，evidencing incomplete cell annotations under the original cell type nomenclature. Therefore, to refine the cell atlas and harmonize the cell type nomenclature, it is imperative to re-annotate the cell types of all clusters at each level. Accordingly, DII-MAS conducts differential expression analysis (DEA) on cell clusters at each level to identify marker genes for each cluster, which aid in cell type annotation. DII-MAS automatically invokes computational tools for marker gene calculation, and subsequently saves the results guided by researcher prompts. The prompts between researchers and DII-MAS, and the detailed steps of hierarchical clustering, differential expression analysis are illustrated in Figure 3d and Supplementary Figure 1d. It can be found that DII-MAS autonomously generates code, installs required software packages, executes the code, and proactively initiates conversation with researchers to confirm subsequent actions. More details of the data integration, hierarchical clustering, and differential expression analysis in the workflow and implementation refer to the Experimental Setup section in Methodology.

Cell type annotation

Based on the cell clusters derived from hierarchical clustering and the marker genes identified through differential expression analysis, DII-MAS performs the sub-task of cell type annotation for each cell cluster. DII-MAS implements two separate annotation approaches: invoking Decoupler[111] for annotations and directly leveraging LLMs for annotations[112]. These dual annotation approaches enable comparison, allowing observations of discrepancies between the annotations generated by tool invocations versus those produced directly by LLMs. In the case of decoupler annotation, DII-MAS ingests a human cell marker gene database, and subsequently Decoupler utilizes this to identify cell types based on the marker genes identified within each cell cluster.

Alternatively, the LLM annotation approach directly utilizes the marker genes of cell clusters to assign cell types. Notably, the LLM annotation approach requires specification of the parent cell type when annotating subclusters.

Following the detailed execution plan, DII-MAS sequentially completes annotations for all cell clusters from level 2 to level 4, where the results of Decoupler annotation and LLM annotation depicted in Figure 4a and Supplementary Figure 3a, respectively. In comparison to the original labels (Supplementary Figure 2a), both annotation approaches successfully complete annotations for all clusters at each level, concurrently exhibiting a clear delineation of relationships among cell populations. The Decoupler annotation results present a more stable hierarchical structure, embodying favorable lineage characteristics. Subsequently, we calculated the distribution of cells within each cell type at level 3 of Decoupler annotations across the original independent datasets, alongside entropy values for each cell type, as illustrated in Figure 4b. From these computations, it is evident that the majority of cell types comprise cells from multiple datasets, whereas a few rare cell types originate predominantly from individual datasets. Furthermore, comparing Decoupler annotations against the original labels, we calculate entropy for each cell type in relation to the distribution of original annotations, while calculate adjusted rand index (ARI=0.66) between them, as depicted in Figure 4c. It is observed that some cell clusters encompass multiple original labels, partly due to the aggregation of similar cells from multiple datasets after integration, and partly due to the emergence of ambiguous cell clusters following reclustering. LLM annotations exhibit a similar characteristic with respect to the distribution of original labels, with ARI of 0.54.

Lastly, we compared the distributions of Decoupler annotations on level 3 cell clusters with those from the LLM, with ARI of 0.78. As cell clusters sharing identical names are consolidated into one category, it is noticeable that some

cell types are inherently susceptible to being misidentified or confused with one another. This also underscores the continuity in cellular states within cell atlases, rendering single-cluster definitions inadequate for fully capturing the complexity of cell types. Thus, employing a hierarchical annotation strategy for cell atlas construction facilitates lineage tracing and the discovery of novel subpopulations[129]. Notably, the annotations generated by LLMs are subject to variation due to updates in model versions and differences in prompt formulations, reflecting the inherent stochastic nature of generative processes.

Variance explained by covariates

Following the integration of datasets, hierarchical clustering and DEA, and cell type annotation, we have successfully constructed a large-scale human lung cell atlas. While the data integration algorithm scANVI effectively mitigates batch effects, residual batch effects persist. DII-MAS employs the embeddings of scANVI to conduct variance analyses across different cell types and covariates, with results depicted in Figure 5a. These analyses reveal that technical covariates, such as tissue dissociation protocol, tissue sampling method, and single cell platform, still demonstrate noticeable variation contributions across many cell types. Specifically, tissue sampling methods such as 'brush', 'biopsy', 'donor_lung', 'scraping', and 'surgical_resection' show varied contributions to differential expression across cell types, as illustrated in Figure 5b. Notably, cell types that incorporate a greater variety of tissue sampling methods display more significant variation, with a total correlation coefficient of 0.68. Biological covariates, in contrast, exhibit lower contributions to variation across different cell types. This also underscores that adopting uniform technical protocols can significantly alleviate batch effects. Automated variance analysis conducted by DII-MAS yields conclusions consistent with those derived from the manual analysis presented in the original study.

IV. Discussion

A. Summary

In the future trend of data-intensive and computation-intensive in big biological data processing and knowledge-intensive and intelligence-intensive in the convergence of interdisciplinary, this study delineates a data-intelligence-intensive scientific research paradigm in life sciences, enabling hypothesis-free exploratory research. In this paradigm, this study proposes a multi-agent system (DII-MAS) based on LLMs, which forms a high-quality collective intelligence through close collaboration between multiple agents and human experts to conduct data-intelligence-intensive life science research. This study devises an agent group management strategy, an effective human-agent interaction mechanism, a shared interdisciplinary knowledge database, and continuous learning strategies to augment the capabilities of DII-MAS. The feasibility of the data-intelligence-intensive research paradigm and the capabilities of DII-MAS are substantiated in the construction of human lung cell atlas. Throughout a suite of evaluation metrics, the multi-agent system consistently demonstrated outstanding performance. The construction of human lung cell atlas not only replicates the processes detailed in the original study but also employs a hierarchical annotation strategy to illustrate the continuous nature of cellular states.

B. Limitations

Although DII-MAS exhibits enormous promise in tackling data-intelligence-intensive research tasks, there remain limitations and areas requiring further development. Multi-agent systems usually require a large amount of computing

resource overhead, especially when dealing with complex tasks and large-scale data, how to use these resources efficiently remains a critical challenge. Promising solutions include dynamic resource allocation, elastic computing, containerization, and microservice architectures that dynamically increase or decrease compute resources based on demand to improve resource utilization efficiency. Furthermore, complex tasks usually need to be decomposed into sub-tasks parallel or data parallel, and how to maintain load balance and control energy consumption and cost is also a key problem. Equipping multi-agent systems with task prioritization scheduling algorithms and load balancing mechanisms emerges as a promising strategy to address these complexities. Additionally, incorporating energy-efficient management strategies, designing more efficient algorithms, leveraging heterogeneous computing resources, and employing model compression techniques can significantly reduce the cost of computation and energy consumption. Efficiently collecting, transmitting, processing, storing, and managing data to accurately monitor task progress and resource utilization poses another critical challenge. Current promising solutions entail lightweight data acquisition, data compression, streaming data processing, distributed data storage, and dashboard visualizations for real-time insights.

Designing effective prompts[113] for user interaction with multi-agent systems, ensuring clear communication and efficient task delegation, presents a critical challenge. Vague, inconsistent, overly complex prompts, or those lacking coherent context, can lead to misdirected agent decisions and erroneous outputs, highlighting the importance of precise and contextually appropriate prompting strategies. Current prompt engineering approaches include: utilizing concise, clear, and neutral prompts; adopting stepwise guidance, incremental processing, and conversational prompts for complex tasks; reusing context, implementing positive feedback loops, and incorporating knowledge retrieval mechanisms to mitigate comprehension deviations. Furthermore, enhancing

the diversity and adaptability of interactions to varying user needs and contexts poses a critical challenge. This includes supporting multifaceted interaction modes such as voice, vision, and text, alongside adaptive learning algorithms and feedback mechanisms to meet personalized interaction requirements. Potential solutions include multimodal fusion, cross-modal learning, online learning, user profiling.

C. Potential future improvements

With the rapid advancement of AI technologies, AI agents have demonstrated substantial potential across various domains and ushered in new development opportunities. In cutting-edge technologies such as embodied intelligence[114,115,116], world models[117], and digital twins[118], the integration of AI agent technology with these domains continues to drive innovation and advancements in various fields.

Embodied intelligence serves as a physical extension of AI agents into the real world, which can manipulate the senses and act of robots to interact with physical environment, and combine physical experience with cognitive processes. AI agents focus on autonomous learning, decision-making, and task execution based on data and algorithms, while embodied intelligence emphasizes physical interaction capabilities. When AI agents integrate embodied intelligence, they gain enhanced capacity to manipulate the physical environment, which requires the agent to have strong cognitive processing and physical interaction capabilities. For example, medical robots use the cognitive abilities of AI to diagnose patients and then perform specific physical therapies, embodying the synergy of intelligence and physical interaction. Embodied intelligence also has broad prospects in life science research, which can bring

revolutionary changes to precision medicine, automated analysis of omics data, cell and molecular experiments, and ecological environment research.

The world model is the cognition and abstraction of AI agent's environment, which can enhance the agent's ability to perceive and understand changes in environmental states, and improve decision-making and action capabilities. Integrating AI agents with world models enables them to better understand and adapt to complex, dynamic environments. For example, in robotic navigation and autonomous driving, world models representing roads, traffic, and pedestrians enable agents to make decisions and navigate, avoiding obstacles and identifying optimal routes. In life sciences, world models apply to simulating disease spread, drug discovery, studies in biological evolution and genetics, integrating diverse data from organisms, compounds, and physical environments, thereby empowering agents with heightened environmental perception, informed decision-making, and precise actions.

Digital twins serve as virtual replicas of physical entities or environments, facilitating bidirectional communication between the physical realm and its digital counterpart. The convergence of AI agents with digital twins amplifies their capabilities and adaptability, encompassing real-time data analytics, simulation-based optimization, fusion of multi-source data, and environmental simulations. In smart cities, digital twins supplying real-time traffic data empower AI agents to optimize signal timing through refined decision-making, effectively alleviating congestion. In life sciences, digital twins integrate multi-source biological data and furnish virtual environments that enable AI agents to perform numerous simulations, thereby diminishing risks associated with physical experiments and expediting the research process.

In the future, AI agents will transcend their roles as mere human assistants to become collaborative partners, augmenting human capabilities across diverse

tasks and delivering more intelligent decision-making support. AI agents will facilitate broader and more convenient applications in domains such as autonomous driving, smart factories, precision medicine, drug discovery, and scientific research, seamlessly integrating into daily life and work routines.

V. Methodology

A. Technical specifications of multi-agent system

Overall framework.

The overall architecture of the proposed DII-MAS comprises agents, a coordination layer, and the environment, as illustrated in Figure 1b. Agents, the core components of the system, utilize LLMs as their central control nucleus. Based on task allocation, agents with different roles undertake distinct responsibilities. The coordination layer is responsible for mediating interactions between agents and the environment, as well as orchestrating the behaviors of agents. The environment, as an external context for DII-MAS, influences the behavior of agents, while the actions of these agents in turn modify the state of the environment.

DII-MAS employs AutoGen as its underlying framework that enables developers to customize task-specific agents from the predefined agent modules, and incorporates natural language conversation protocols to facilitate interactions between users and agents. In the coordination layer of DII-MAS, we design a community structure management strategy to administer agents in various roles and coordinate inter-group communications, optimized for addressing large-scale life science research tasks involving the engagement of numerous agents. In this study, the task of constructing a lung cell atlas

employs three agent groups: coarse-level planning, fine-level planning, and action execution. Additionally, we devise three effective strategies within the coordination layer: A user-friendly and efficient human-agent interaction mechanism supports tightly coupled collaboration between human experts and agents; A shared interdisciplinary knowledge database enhances the cognitive capabilities of agents through knowledge empowerment, and supports task-driven extraction of relevant information; Continuous learning strategies, such as self-reflective learning and competitive learning, enhance the decision-making and execution capabilities of agents, thereby rapidly accommodating emerging and challenging life science research tasks.

Agent groups.

We illustrate the community structure management strategy within DII-MAS using the case of agent groups engaged in the task of with constructing a lung cell atlas. Each group comprises multiple agents fulfilling distinct roles, and those agents sharing the same role across different groups exhibiting slightly different variations in responsibilities.

Coarse-level planning group. As the name suggests, the coarse-level planning group aims to formulate a coarse-grained task execution plan based on user inputs that typically comprise task specifications and data descriptions. This group has three agents: 'Planner', 'User Proxy' and 'Reviewer'. 'Planner' initially receives user inputs and generate an overall plan. Subsequently, 'Reviewer' meticulously reviews the overall plan and propose s suggestions for revisions. Afterwards, 'Planer' revises the overall plan based on suggestions, and relays the revised overall plan to 'User Proxy'. 'User Proxy' provides a static human-agent interaction interface, enabling users to verify whether the overall plan requires any further adjustments.

Fine-level planning group. The fine-level planning group has similar agent roles and internal communication protocols to the coarse-level planning group, consisting of 'Refined Planner', 'User Proxy', and 'Reviewer'. The difference is that the fine-level planning group receives a single step from the overall plan devised by the coarse-level planning group as input, to generate and refine a more detail plan for that step.

Action execution group. Action execution refers to code generation, execution and debugging. This group consists of three agents including 'Coder', 'User Proxy' and 'Checker'. Specifically, 'Coder' receives the detail plan for the current step from the fine-level planning group and generates the corresponding code. 'User Proxy', equipped with a Python kernel and a code execution environment, is responsible for receiving and executing the code generated by 'Coder'. Moreover, within the action execution group, 'User Proxy' offers two types of human-agent interaction interfaces (i.e., static and dynamic). The static interface is exposed following the initial receipt of code by 'Coder', so that users can inspect the code for any potential issues prior to execution. The timing of the dynamic interface exposure is determined by 'Checker'. When the code is executed successfully, 'User Proxy' will terminate the group conversation. Following code execution, 'Checker' receives outcomes returned by 'User Proxy', undertakes code debugging, identifies issues within the current code, suggests remedial modifications, and returns these to 'Coder' as feedback. 'Checker' will instruct 'User Proxy' to expose the dynamic human-agent interface in three specific situations. Firstly, 'Checker' receives a repeated execution error, indicating 'Coder' inability to correct the error. Secondly, 'Checker' receives more than five error messages pertaining to code anomalies. Thirdly, 'Checker' identifies instances necessitating manual specification of names within errors, such as actual gene names present in the dataset. Additionally, the code generation capability of 'Coder' is enhanced through continuous learning by feeding its output from the previous step into the current one, ensuring the continuity and consistency of generated code.

Communications between agent groups. With in each agent group, we have introduced how agents interact. In this section, we will comprehensively elaborate the communication mechanisms between agent groups. The proposed multi-agent system comprises three agent groups, i.e., coarse-level planning group, fine-level planning group and action execution group, which communicate approximately linearly. As illustrated in Figure 2b, the coarse-level planning group decomposes user-assigned task into a multi-step overall plan following predefined rules, with subsequent refinements. Then, the fine-level planning group iteratively conducts further detailed planning on each step of the overall plan in a loop manner, subsequently refining the detailed execution plans for each step. The action execution group is responsible for executing the detailed execution plan of each step, involving code generation, running, and debugging. Upon completion of a step, the process iterates to the next one until the entire overall plan is accomplished.

Effective human-agent collaboration.

Although multi-agent systems can autonomously carry out routine bioinformatics analysis tasks, human-agent collaboration remains the most effective approach in tackling complex and dynamic tasks. There are three primary reasons: Firstly, bioinformatics analysis tasks often entail personalized data processing stages, where agents relying on their inherent knowledge may struggle to accommodate personalized requirements. Secondly, when agents cannot autonomously resolve encountering errors, they risk being locked in infinite loops, hindering progression to subsequent steps. Lastly, insufficient user input can lead to misinterpretations by agents due to information gaps, resulting in erroneous decisions or failed executions.

In the data-intelligence-intensive scientific research paradigm, both human experts and AI agents serve as core participants in the scientific investigation. To support the tightly coupled human-agent collaborative workflow, we design a user-friendly and efficient human-agent interaction mechanism for the proposed DII-MAS. Departing from conventional static human-agent interaction models, where the timing of human interaction with DII-MAS is fixed, we devise a dynamic human-agent interaction mechanism that enables agents to determine the timing of exposing the human-agent interface based on the fulfillment of predetermined conditions, thereby ensuring high real-time interaction capabilities. This dynamic human-agent interaction mechanism is extensively employed within the action execution group, that can enhance the capabilities of agents to tackle complex and dynamic tasks. The proposed DII-MAS integrates both dynamic (appears in action execution group) and static (appears in all agent groups) human-agent interaction mechanisms, optimizing autonomous task execution by DII-MAS while minimizing unnecessary human intervention.

Shared interdisciplinary knowledge database.

We establish a shared interdisciplinary knowledge database within DII-MAS, thereby augmenting the cognitive capacities of agents with diverse roles through knowledge empowerment. By collecting and encoding knowledge from multidisciplinary research articles into a form comprehensible to agents, the shared interdisciplinary knowledge database enables agents to retrieve relevant information. For instance, during coarse-level planning, 'Planner' initiates its process by retrieving relevant information from the knowledge database, which is then strategically integrated as supplementary cues into its planning workflow. Knowledge-driven insights augment the input of 'Planner', thereby the planning quality of 'Planner' is enhanced by enriching the decision-

making process with informed perspectives. In the proposed DII-MAS, knowledge empowerment is crucial for achieving high autonomy and efficiency in handling complex biological data and tasks, which can alleviate biases and illusions inherent in reasoning and decision-making processes.

Continuous learning.

In order to rapidly accommodate emerging and challenging tasks in life sciences research, we devise continuous learning strategies for the proposed DII-MAS, incorporating both self-reflective learning and competitive learning. In the system, agents engage in self-reflective learning by reviewing their own behaviors, progressively refining autonomous decision-making and execution capabilities over time, thereby enhancing task efficiency in subsequent missions through process streamlining, selecting optimal tools, and minimizing manual intervention. For example, the self-reflective learning mechanism is embodied in two kinds of agents, i.e., 'Reviewer' and 'Checker', that are responsible for examining the plans formulated by 'Planner' (or 'Refined Planner') and the code produced by 'Coder'. Through iterative interactions and self-reflective learning, they have refined their capabilities to autonomously suggest modifications, thereby improving the quality of feedback. Furthermore, agents gain new knowledge through competition with other agents, serving as a strategy to augment the collective intelligence of agent group. Reflected explicitly in 'Coder', we employ evaluation matrices to evaluate the diverse results generated by multiple Coder agents from different ideas on code generation, imposing penalties or rewards to facilitate competitive learning and enhance their capabilities.

B. Experimental Setup

Datasets

Data for constructing the core HLCA (human lung cell atlas) is derived from 11 independent studies[119,120,121,122,123,124,125,126,127,128], integrating 14 datasets comprising 168 health lung tissue samples from a total of 107 individual donors, with diversity in age, sex, ethnicity, BMI and smoking status. The HLCA project has preprocessed these datasets, involving harmonization of cell type nomenclature, alignment of gene nomenclature, data quality control, and data normalization. Specifically, the HLCA project established a five-level hierarchical cell type nomenclature to harmonize cell types across datasets, aligned gene nomenclature to the GRCh38[130] human genome reference with Ensembl[131] standardization, applied quality control to exclude cells with fewer than 200 genes and genes expressed in less than 10 cells, and performed SCRAN[132] normalization on raw counts for uniformizing UMI counts per cell.

Lung cell atlas workflow

The development of lung cell atlas workflow is accomplished through a collaborative effort between researchers and DII-MAS. Researchers specify the task objectives, prompting DII-MAS to generate an initial workflow design, which then iteratively refined via extensive human-machine interactions, guided by researcher feedback informed by insights and experience. Additionally, during the task execution phase of DII-MAS, researchers can actively engage in human-machine interactions to identify and rectify incorrect actions of the system. Finally, this collaborative effort results in a detailed workflow for the construction of lung cell atlas.

Data integration. To integrate disparate single-cell datasets of lung tissue, batch correction algorithms are employed to dimensionally reduce the preprocessed datasets, yielding an integrated embedding that aligns cells of the same type across independent datasets within a feature expression space. This unified embedding facilitates the construction of a comprehensive and unbiased cell atlas, enabling insightful analyses into the biological diversity and heterogeneity within lung tissue.

Currently, there exist numerous approaches for integrating single-cell datasets, such as BBKNN[133], fastMNN[134], Harmony[135], Scanorama[136], scGEN[137], scANVI[110], scVI[138], and Seurat[139]. The scIB[140] benchmarking framework, by assessing the performance of various integration methods across multiple public datasets, evidences the capability of scANVI in achieving the optimal comprehensive performance. Hence, DII-MAS employs scANVI for single-cell dataset integration within its workflow, which requires the input of the top 2,000 highly variable genes (HVGs) derived from raw count data. Specifically, multiple datasets are initially merged to perform SCRAN normalization, yielding the standardized gene expression profile for each cell, constituting the raw count data. Next, the scanpy.pp.highly_variable_genes function provided by the scanpy[141] toolkit is employed to identify the top 2000 HVGs, configured with parameters: batch_key="dataset", min_mean=0.6, flavor="cell_ranger", n_top_genes=2000, leaving default settings for all other parameters. Furthermore, scANVI requires cell type annotations, which are curated during preprocessing to preserve reliable original labels in accordance with the methodology described in the HLCA study, while the unreliable original labels are designated as an unknown category. Guided by user prompts, scANVI utilizes the names of the independent study datasets as batch-covariates, sets the number of hidden features to 30, while other parameters are automatically configured by the multi-agent system to their defaults: n_layers=2, encode_covariates=True, deeply_inject_covariates=False,

gene_likelihood="nb", with GPU acceleration enabled via use_cuda=True. Employing GPUs significantly accelerates the training process of the scANVI method. The extracted hidden embeddings of the integrated datasets from the trained scANVI model, which have undergone batch-effect correction by scANVI, are utilized for subsequent clustering analyses and visualization tasks.

Hierarchical clustering and DEA. Following integration and batch-effect correction by scANVI, the cell atlas of lung tissue comprises roughly 580,000 cells, requiring cluster analysis for cell type classification. To gain deeper insights into cellular diversity and heterogeneity, a hierarchical clustering approach is employed to delineate cell populations by iterating from coarse to finer resolutions. Firstly, the K-nearest neighbor graph of the hidden embeddings is constructed using the scanpy.pp.neighbors function, with the parameter for the number of neighbors set to n_neighbors=30. At the first level of clustering (i.e., the coarsest level), a leiden clustering function scanpy.tl.leiden is employed to generate the first level cell population clusters, with the clustering resolution parameter of resolution=0.01. Subsequently, new K-nearest neighbor graphs for each cluster from the first level cell populations are constructed using the hidden embeddings and the scanpy.pp.neighbors function with n_neighbors=30 again. Reapplying leiden clustering to each cluster from the first level, with a clustering resolution parameter of resolution=0.1, produces the second level cell populations. Next, sequentially proceeding the third and fourth level analysis like second level ultimately generates a hierarchical, four-level structure of cell populations. In order to track relationships among cell populations effectively, hierarchical clusters are labeled according to their parental lineage and ordinal position among sisters, such that Cluster 1.2 denotes the third sub-cluster (counting from 0) of Cluster 1. Furthermore, for visualizing cell clusters at each level, the K-nearest neighbor graph of the hidden embeddings is fed to the scanpy.tl.umap function for dimensionality reduction followed by scanpy.pl.umap function to plot 2D

UMAPs, where the color parameter in scanpy.pl.umap is set to the level key of hierarchical clustering and the palette parameter defines the colors for cell clusters.

Following hierarchical clustering, in order to streamline cell type annotation, differential expression analysis is performed independently on cell clusters from each hierarchical level to uncover marker genes characterizing each cell cluster. Considering the dependence of marker gene identification on the cell populations under analysis, in anticipation of supporting diverse cell type annotation methodologies, two differential expression analysis strategies with different scopes are conducted on cell clusters at every hierarchical level. The first strategy compares gene expression differences across all clusters within the same hierarchical level to identify differentially expressed genes for each cluster, aiming to distinguish one cluster from all other clusters in the same hierarchical level. The second strategy compares gene expression differences across the clusters that have the same parent in the upper level, aiming to reveal differentially expressed genes distinguishing one cluster from its sister clusters. The scanpy.tl.rank_genes_groups function in scanpy toolkit is utilized for differential expression analysis, with the groupby parameter set to the level key of hierarchical clustering, n_genes specifying the number of top differentially expressed genes to retrieve, and the default t-test employed for hypothesis testing. To refine the selection for more significant differentially expressed genes, the scanpy.tl.filter_rank_genes_groups function is applied for additional filtering with default parameters: min_in_group_fraction=0.25, max_out_group_fraction=0.5, min_fold_change=1. After filtering, the differentially expressed genes are ranked based on significance and inter-group variation, enabling the selection of a subset of top-ranked genes as marker genes for each cell cluster.

Cell type annotation. Leveraging the hierarchical, four-level cell clusters and their specific marker genes from previous steps, cell types are assigned to each cluster. Considering the coarse granularity of first level clusters renders their annotation less informative, thus the cell type annotation is performed from the second level cell clusters. To ensure optimal annotation results, two annotation strategies are conducted to make comparison, enabling DII-MAS to automatically select the more effective strategy in the future version. Firstly, annotations are performed on each cell cluster within each hierarchical level using the Decoupler[111] tool in conjunction with the human cell-type marker gene database from CellMarker[142]. Secondly, LLMs are employed for the annotation of individual cell clusters[112]. Initially, all cell clusters in the second level are annotated using marker genes computed across all cell clusters in this level. Subsequently, based on the annotated second level cell clusters, each third level cell cluster is annotated under the context of its assigned parental cell type, leveraging marker genes calculated from comparisons with sister clusters within the same layer. This same strategy is then replicated for the annotation of fourth level cell clusters. Refinements are required for the annotation results, because each of the two annotation strategies has unique features. For example, if all subclusters are indistinguishable within a parent cluster, then all subclusters inherit the cell type of the parent cluster. Clusters that remain uncharacterized require additional investigation to ascertain if they denote rare or previously unreported cell types.

Variance explained by covariates. To elucidate the correlations between covariates in distinct sequencing methodologies, experimental protocols, biological characteristics, and inter-individual variations in cellular atlases, a quantitative assessment of the variance in gene expression profiles attributable to each covariate within every cell type is performed. To ensure robust statistical analysis, the cell types represented by fewer than two samples or those with a total cell count below 50 are excluded, while any sample within a

cell type containing less than ten cells is also excluded. Initially, the covariate statuses and scANVI latent embeddings per sample for every cell type are statistically summarized to generate matrices. Statistical summaries for scANVI latent embeddings, UMI counts, and mitochondrial ratios involve averaging across all cells within a sample, whereas other covariates are sampled from any cell within the sample, as they remain invariant among cells from the same sample. Within matrices derived for each cell type, linear regressions between each covariate and each scANVI latent embedding are conducted to predict the embedding feature explained by each covariate, subsequently calculating variances. Finally, the variances across all scANVI latent embeddings are summed to quantify the differential impact of covariates across individual samples.

Implementation Details

Computing resource. All analyses presented in the study run on Bio-OS[143] with elastic resource, at least 128Gb RAM memory, 10 cores of 2.5GHz CPU, and a GPU with 32Gb memory. And the following python (v3.11) packages support for DII-MAS and lung cell atlas workflow are required: python==3.11.7, chainlit==1.0.100, autogen==1.0.16, pyautogen==0.2.19, openai==1.14.0, jupyter==1.0.0, httpx==0.23.0, jupyterlab==4.0.12, notebook==7.0.8, torch=1.7.1+cu110, scanpy==1.10.0, scarches[144]==0.3.5, decoupler==1.6.0, numpy==1.26.4, pandas=2.2.2, scikit-learn==1.4.2

Details of the system. The preliminary input from the user includes the task and dataset descriptions, separated by a line break. Upon receipt of this preliminary input, the entire multi-agent system is initiated into operation. Specifically, the user's task requirements are subjected to the coarse-level planning group to obtain a high-level plan that is used to extract sub-steps with

a sub-step extractor. Then, each sub-step is sent to the fine-level planning group for a detailed plan. Moreover, at the end stage of the coarse-level planning and the fine-level planning groups, the multi-agent system will automatically provide an interactive interface for users to determine whether the plan completed by the agents still needs to be modified. The action execution group receives information from the fine-level planning group and is responsible for generating, debugging, and executing code. During the code debugging process, if a code error arises that the group cannot resolve independently, the group will provide an interactive interface for users to participate in solving the error. Once the code for the current sub-step runs successfully, the action execution is terminated. The multi-agent system cyclically executes fine-level planning and action execution groups according to the sub-steps division until all are completed. The multi-agent system has completed the user's task, exits the operational mode, and returns to a waiting state.

Details in the lung cell atlas task.

scANVI integration. First, the user gives the task description to the multi-agent system, that is, "Perform single-cell data integration based on scANVI on the provided h5ad data." and the data description, that is, "/path/to/your/data/: all single-cell data need to be integrated and analysis for constructing the human lung cell atlas. The adata column of "scanvi_label" contains the label information for scANVI integration.". The main steps of coarse-level planning include: (i) Load and preprocess the data; (ii) Integrate the data using scANVI; (iii) Generate Integrated Embeddings; (iv) Save the integrated data. Next, these sub-steps will be cyclically sent to the fine-level planning group to obtain more planning details, such as how to preprocess data, what function to use for data integration, etc., providing rich information for the action execution group. Moreover, the user needs to provide necessary parameter settings for the multi-agent system through human-agent interaction. For instance, in this task, the

user can provide parameter settings for the data integration function through the fine-level planning group as "Set n_epochs_unsupervised=500, n_epochs_semisupervised=200 in train() function". The user can also provide GPU setting information through the action execution group as "Use scANVI on GPU. So you need to add os.environ['CUDA_VISIBLE_DEVICES'] = '0'".

Hierarchical clustering. Similarly, the user gives the task description: "Perform multi-level clustering of the cells in the HLCA. Note that counting the initial clustering, the total number of clustering levels is 4. For the final HLCA, you also need calculate marker genes for every clustering." and the data description, i.e., "/path/to/your/data/: the integrated single-cell data with scANVI that needed to be multi-level clustered. The integrated representation is in "X_scANVI" of the adata.". The main steps of coarse-level planning include: (i) Load the dataset; (ii) Initial clustering (level 1); (iii) Subsequent clustering (levels 2-4); (iv) Calculate marker genes for each cluster; (v) Save the final annotated dataset. Next, these sub-steps will be sent to the fine-level planning group cyclically to obtain more planning details, for example, the specific logic implemented in steps (iii) and (iv). Users need to provide critical parameter information in fine-level planning and action execution; for example, in the initial clustering step (step (ii)), "Set n_neighbors = 30 in sc.pp.neighbors and resolution_parameter = 0.01 in sc.tl.leiden". In addition, when errors occur in the logic of multi-agent or data storage methods, the user needs to intervene promptly through human-agent interaction. For instance, in calculating marker genes (step (iii)), when the depth of the cluster is greater than 1, all clusters included in the current cluster parent class need to be used as a group to calculate marker genes. The multi-agent system cannot capture this logic independently, so the user needs to provide a prompt as "If the cluster depth is more than 1, you need also provide group parameters (elements are a list of fine-grained clusters under the parent of current cluster (e.g. [0_1,0_2,0_3] under 0)) in sc.tl.rank_genes_groups function.".

Cell type annotation. Similarly, users provide a task description stating, "We want to achieve cell types annotation by Decoupler," and a description of the dataset: "/path/to/your/data/: stores 3 level clustering labels (cluster_level_2, cluster_level_3, cluster_level_4) which is used for annotation. /path/to/your/data/: stores cell marker manual annotation database." The primary steps of the coarse-level planning include: (i) Load the necessary libraries for analysis; (ii) Load and preprocess the data; (iii) Prepare the marker gene list for Decoupler; (iv) Run Decoupler for cell type annotation; (v) Save the annotated data. Subsequently, these sub-steps are cyclically sent to the fine-level planning group to obtain more detailed planning specifics, such as the concrete logic implemented in steps two and four. During Fine-level Planning and Action Execution, users must provide critical parameter information or guiding suggestions, for instance, in the data loading and preprocessing step (step ii), to eliminate unnecessary processes, "Only retain 'Load the .h5ad File' and 'Load the Cell Marker Database'." Additionally, when agents fail to recognize data structures or when individualized data processing is required, users must intervene timely through a manual interaction interface. For example, in preparing the marker gene list step (step iii), guidance like "Set adata.var_names to 'gene_symbols' in adata.var," or "Retain data where adata.var.highly_variable is true for cell annotation and copy it," is necessary. When the multi-agent system cannot correctly use the Decoupler tool (step iv), users must also provide prompts, such as "The function for performing ORA is dc.run_ora, and its parameters contain mat=adata_filtered, net=filtered_df, source='cell_name', target='Symbol', min_n=3, use_raw=False. There's no n_jobs in dc.run_ora."

Similarly, users provide a task description, "We want to achieve cell type annotation by Azure GPT-4 for HLCA. The GPT-4 is requested by curl, and the messages are a single round chat", along with a dataset description,

"/path/to/your/data/: the structure of the data is ..... We will use this data to achieve that: .....". (Refer to appendix or omitted for brevity) The primary steps of coarse-level planning include: (i) Parse the Annotation Preprocess Data; (ii) Level 2 Cell Type Annotation; (iii) Level 3 Cell Type Annotation; (iv) Level 4 Cell Type Annotation; (v) Final Output Preparation. These sub-steps are repetitively sent to the fine-level planning group to garner more detailed planning specifics, particularly how steps two and three are to be executed, requiring users to provide critical parameter information during Fine-level Planning and Action Execution. For instance, in the initial clustering step (step ii), prompts such as "Batch size is 10 for every curl request, and the response is one cell type for each row. Set prompt for GPT-4 in level 2: 'Identify cell types of human lung cells using the following markers. Identify one cell type for each row. Only provide the cell type name," along with GPT-4 API call information, are essential. Additionally, as the steps progress, users must intervene through the manual interaction interface when agents miss previous information. For example, in the Level 3 Cell Type Annotation step (step iii), users need to re-input prompts reminding the agent of missed information, "Batch size is 10 for every curl request, and the response is one cell type for each row."

Similarly, the user gives the task description: "Determine how much of the variance in the integrated HLCA embedding is explained by each of the metadata covariates including ... . You need to use the results of the third level clustering for analysis. Note that your analysis should be conducted from sample-level observations." and a description of the data (see the appendix or omitted). The main steps of coarse-level planning include: (i) Load the data, (ii) Extract and aggregate relevant data, (iii) Perform variance analysis, (iv) Save the model results. Next, these sub-steps are sent to the fine-level planning group to obtain more details, such as the detailed logic of integrating data and which method to choose for variance analysis. Since the task contains a lot of personalized data processing and conditional judgments, which is a weak point

of the multi-agent system, the user needs to provide relevant prompts. For instance, when integrating data from the sample level (step (ii)), the multi-agent system will save all dimensions of "X_scANVI" into one parameter, which makes subsequent variance analysis impossible. Therefore, the user must provide the prompt as "Store each component of the X_scanvi_embed as a separate row in the aggregated_row. You can first fill other data into aggregated_row. Then use a for loop to add component keys into aggregated_row with changing key names". In variance analysis (step (iii)), the multi-agent will not be able to remember to deal with NaN and character values independently. The user needs to give a prompt like "Y may be NAN or string value, you should deal with these conditions.". Besides, users can provide necessary parameter settings to the multi-agent system. Otherwise, the multi-agent system will automatically set parameter values or generate code according to default values.

C. Evaluation Metrics

Metrics for multi-agent system evaluation

Evaluation on Planning. This section aims to verify whether the proposed multi-agent system can achieve superior planning performance compared to a single-agent system. To this end, we have conducted a comparison of the performance of the proposed multi-agent system with two advanced single-agent models, GPT-4 and GPT-3.5. We have selected the following five metrics to measure the performance of different systems:

a) Clarity. It evaluates the explicitness and logical coherence of the generated plan in its expression. A clear plan should allow the executor to understand and implement it without ambiguity.

b) Technical Feasibility. It considers the practical operability of the solutions proposed in the plan. Technical feasibility includes assessing the necessary resources, current technological limitations, and potential risks.

c) Structurality. The structurality metric focuses on the organization and layout of the plan. Good structurality implies that the plan has a clear beginning, middle, and end phases, with logical coherence between each stage.

d) Comprehensiveness. The comprehensiveness metric measures whether the plan has considered all relevant constraints and objectives. A comprehensive plan should cover all aspects of the issue without omitting key execution steps or objectives.

e) Rationality of Method Selection. It assesses the match between the chosen algorithm and the needs of the plan. A rational choice of algorithm should be based on the problem's characteristics and constraints to ensure the plan's efficiency and effectiveness.

We employed a scoring method using GPT-4, selecting multi-level clustering and DEA tasks to evaluate the metrics above. Specifically, we first provide task and data descriptions for the single-agent model GPT-4 and GPT-3.5, consistent with the information used in our multi-agent system. Subsequently, we ask these models to generate corresponding planning schemes. The prompt used is designed as follows:

"I provide a bioinformatic task and the data description as the following, do a plan for the task, do not contain any code and assume all needed packages are installed and should not write install software as a separate step:\nPerform multi-level clustering of the cells in the HLCA. Note that counting the initial clustering, the total number of clustering levels is 4. For the final HLCA, you also need calculate marker genes for every clustering.\nxxx.h5ad: the integrated single-cell data with scANVI that needed to be multi-level clustered. The integrated representations is in 'X_scANVI' of the adata."

Finally, we utilized GPT-4 to score three planning schemes derived from the multi-agent system, as well as from GPT-4 and GPT-3.5. The scoring prompt used is as follows:

“I provide you with the user's task need and data description:\nPerform multi-level clustering of the cells in the HLCA. Note that counting the initial clustering, the total number of clustering levels is 4. For the final HLCA, you also need calculate marker genes for every clustering.\nxxx.h5ad: the integrated single-cell data with scANVI that needed to be multi-level clustered. The integrated representations is in 'X_scANVI' of the adata.\n\nNow there are three plans. The main purpose of planning is for subsequent code generation. I hope that following the planning will make it easier to generate executable code and obtain results. Based on the mentioned prupose, rate these three plans from different dimensions including clarity, feasibility, structure, comprehensiveness, and methods. Feasibility mainly refers to technical feasibility, which means the assessment of the feasibility of implementing the plan using code. Methods represents the assessment of the suitability and rationality of the selected algorithm in planning. \nI give a detailed introduction to some of these dimensions for your reference as following. \nThe score range can be 0-10.\nThe first plan:\n"+first_plan+'\nThe second plan:\n'+second_plan+'\nThe third plan:\n'+third_plan”

Evaluation on Human-Agent Interaction. This section aims to verify the efficiency of the human-agent interaction modality proposed within the multi-agent system. We designed two interaction modes: static and dynamic. The static mode draws from the 'Always' mode in AutoGEN, which provides an interaction interface at preset time points. The dynamic mode, on the other hand, integrates rule-based recognition with the autonomous identification of the agent to enhance the stability of the interaction. To demonstrate the superiority of our approach, we compared the 'Always' and 'Never' modes of AutoGEN under the condition of maintaining the same architecture of our agent

system. We selected three key performance indicators for evaluation: the frequency of user interaction, the frequency of internal agent interaction, and the task completion rate. Since the variance explained by the covariates task has a relatively short execution time, we conduct experiments on this task to more accurately measure interaction frequencies. When statistically analyzing interaction frequencies, each conversation (initiated by the user or the agent) is counted as one interaction round. As for the task completion rate, we calculate it based on the proportion of completed sub-steps to the total number of sub-steps.

Evaluation on Code Generation. This section aims to validate the effectiveness of the proposed multi-agent system in generating reliable and executable task code. To this end, we have evaluated the performance of our multi-agent system and conducted an in-depth comparative analysis with an advanced single-agent model—GPT-4, as well as code written by humans (serving as a benchmark reference). We have adopted the following six indicators to measure and compare the performance of different systems:

a) Accuracy. Assesses whether the generated code can accurately implement the established functions and requirements.
b) Efficiency. It focuses on the performance of the code in terms of resource consumption, including but not limited to execution time, memory usage, and CPU utilization rate.
c) Completeness. Measures whether the generated code includes all necessary components to achieve the intended functionality.
d) Readability. Evaluates the clarity and understandability of the code, involving the appropriateness of naming conventions, structural organization, and comments.
e) Logicality. It checks the coherence and rationality of the internal logical flow of the code, including the correct application of conditional statements, loop structures, and exception handling.

f) Portability. Evaluates the code's ability to be transferred between different environments or datasets.

To conduct the assessment, we utilized GPT-4's scoring mechanism and selected multi-level cluster and DEA tasks to quantify the indicators above. The specific implementation steps are as follows: First, we use the plan obtained from the planning evaluation part as a prompt to guide GPT-4 in generating the corresponding code. Notably, we do not require GPT-4 to generate code in steps during this process, thereby indirectly highlighting the advantages of our adopted hierarchical planning strategy from coarse to fine levels. Finally, we score the code generated by DII-MAS, the code generated by GPT-4, and the code written by humans through GPT-4. The prompt used in the scoring process is as follows:

“You are a bioinformaticse expert with extensive knowledge. I provide you with the user's bioinformatic analysis task need and data description:\nPerform multi-level clustering of the cells in the HLCA. Note that counting the initial clustering, the total number of clustering levels is 4. For the final HLCA, you also need calculate marker genes for every clustering.\nxxx.h5ad: the integrated single-cell data with scANVI that needed to be multi-level clustered. The integrated representations is in 'X_scANVI' of the adata.\n\nNow I will provide you with three codes, please rate these codes from different dimensions including accuracy, efficiency, completeness, logicality, readability, and portability.\nI give a detailed introduction to some of these dimensions for your reference as following. Accuracy is the assessment of whether a code correctly solves the problem and produces the expected output. Comprehensiveness refers to whether most of the functions have been implemented compared with the standard reference. Portability refers to the ability of code implementation to adapt to different data sets or different environments. \nThe score range can be 0-10.\nThe first code:\n''+'python\n'+first_code + '\n' +'\nThe second

code:\n'+'python\n'+second_code + '\n' + '\nThe third code:\n'+'python\n'+third_code+ '\n'"

Metrics for lung cell atlas construction

Entropy. We employ Shannon entropy to quantify the amount of uncertainty or information content inherent in a discrete distribution. Formally, it measures the average amount of information conveyed by each event in a discrete probability distribution. The Shannon entropy is defined as

$$H(X) = -\sum_{i=1}^{n} p(x_i) log_b(p(x_i))$$

where n is the number of possible outcomes, $p(xi)$ represents the probability of the ith outcome occurring, and the base of the logarithm b determines the unit of measurement (commonly, b = 2 for bits). A higher entropy value suggests higher uncertainty or disorder; conversely, a lower entropy implies more predictability or order within the system. In this study, Shannon entropy helps in identifying cell type label disagreement for a specific level of annotation, and calculating the diversity of donors or tissue sampling methods in a cluster.

ARI. The Adjusted Rand Index (ARI) is a widely used evaluation metric in the field of clustering analysis and machine learning. It measures the similarity between two sets of clusters or partitions, typically the ground truth grouping and the clustering output generated by an algorithm. Unlike simpler measures like the Rand Index (RI), which counts pairs of elements correctly classified into the same or different clusters, the ARI adjusts for chance agreement, making it a more reliable indicator especially when comparing partitions of differing sizes. Formally, ARI adjusts the raw Rand Index by comparing the observed agreement of cluster assignments with what would be expected by chance, given the cluster sizes. The index ranges from -1 to 1, with a value of 1

indicating perfect agreement between the two partitions, 0 representing random assignment agreement (what would be expected by chance), and negative values suggesting less agreement than expected randomly, which could imply conflicting clustering structures. In practical applications, such as evaluating the performance of single-cell RNA sequencing (scRNA-seq) data clustering, the ARI provides a standardized measure that accounts for the complexity and stochasticity inherent in real-world datasets. Thus, it is particularly useful for researchers seeking to validate the robustness and biological relevance of their clustering results. By employing the ARI, scientists can objectively assess how closely the inferred clusters mirror the underlying true structure, thereby enhancing confidence in downstream biological interpretations.

VI. Data availability and Code availability

Data and code are available in https://doi.org/10.5281/zenodo.11210015 and https://github.com/lyyang01/DII-MAS respectively.

Acknowledgements

This work was supported by self-supporting Program of Guangzhou National Laboratory (Grant No. SRPG22007), youth science foundation of Guangzhou National Laboratory (Grant No. QNPG23-12).

Author contributions

Conceptualization: Yixue Li.

Supervision: Jiao Yuan, Yixue Li.

Algorithm development and implementation: Yang Liu, Lu Zhou, Rongbo Shen.

Datasets collection, processing and application: Rongbo Shen, Yang Liu, Lu Zhou.

Methods comparisons: Yang Liu, Lu Zhou, Rongbo Shen.

Biological interpretation: Rongbo Shen.

Manuscript writing and figure generation: Rongbo Shen, Yang Liu, Lu Zhou.

Manuscript reviewing: Yixue Li.

All authors approved the manuscript.

Competing interests

The authors declare no competing interests.

Reference

1. Kanehisa, Minoru, and Peer Bork. “Bioinformatics in the post-sequence era.” Nature genetics vol. 33 Suppl (2003): 305-10. doi:10.1038/ng1109
2. Gauthier, Jeff et al. “A brief history of bioinformatics.” Briefings in bioinformatics vol. 20,6 (2019): 1981-1996. doi:10.1093/bib/bby063
3. Soon, Wendy Weijia et al. “High-throughput sequencing for biology and medicine.” Molecular systems biology vol. 9 (2013): 640. doi:10.1038/msb.2012.61
4. Lan, Freeman et al. “Single-cell genome sequencing at ultra-high-throughput with microfluidic droplet barcoding.” Nature biotechnology vol. 35,7 (2017): 640-646. doi:10.1038/nbt.3880

5. Shendure, Jay, and Hanlee Ji. “Next-generation DNA sequencing.” Nature biotechnology vol. 26,10 (2008): 1135-45. doi:10.1038/nbt1486
6. Kalisky, Tomer, and Stephen R Quake. “Single-cell genomics.” Nature methods vol. 8,4 (2011): 311-4. doi:10.1038/nmeth0411-311
7. Gawad, Charles et al. “Single-cell genome sequencing: current state of the science.” Nature reviews. Genetics vol. 17,3 (2016): 175-88. doi:10.1038/nrg.2015.16
8. Tanay, Amos, and Aviv Regev. “Scaling single-cell genomics from phenomenology to mechanism.” Nature vol. 541,7637 (2017): 331-338. doi:10.1038/nature21350
9. Sandberg, Rickard. “Entering the era of single-cell transcriptomics in biology and medicine.” Nature methods vol. 11,1 (2014): 22-4. doi:10.1038/nmeth.2764
10. Stegle, Oliver et al. “Computational and analytical challenges in single-cell transcriptomics.” Nature reviews. Genetics vol. 16,3 (2015): 133-45. doi:10.1038/nrg3833
11. Doerr, Allison. “Single-cell proteomics.” Nature methods vol. 16,1 (2019): 20. doi:10.1038/s41592-018-0273-y
12. Marx, Vivien. “A dream of single-cell proteomics.” Nature methods vol. 16,9 (2019): 809-812. doi:10.1038/s41592-019-0540-6
13. Schoof, Erwin M et al. “Quantitative single-cell proteomics as a tool to characterize cellular hierarchies.” Nature communications vol. 12,1 3341. 7 Jun. 2021, doi:10.1038/s41467-021-23667-y
14. Zenobi, R. “Single-cell metabolomics: analytical and biological perspectives.” Science (New York, N.Y.) vol. 342,6163 (2013): 1243259. doi:10.1126/science.1243259
15. Seydel, Caroline. “Single-cell metabolomics hits its stride.” Nature methods vol. 18,12 (2021): 1452-1456. doi:10.1038/s41592-021-01333-x
16. Cook, Charles E et al. “The European Bioinformatics Institute in 2016: Data growth and integration.” Nucleic acids research vol. 44,D1 (2016): D20-6. doi:10.1093/nar/gkv1352

17. Manzoni, Claudia et al. “Genome, transcriptome and proteome: the rise of omics data and their integration in biomedical sciences.” Briefings in bioinformatics vol. 19,2 (2018): 286-302. doi:10.1093/bib/bbw114
18. Akhtar, Asifa et al. “A decade of molecular cell biology: achievements and challenges.” Nature reviews. Molecular cell biology vol. 12,10 669-74. 23 Sep. 2011, doi:10.1038/nrm3187
19. Hamburg, Margaret A, and Francis S Collins. “The path to personalized medicine.” The New England journal of medicine vol. 363,4 (2010): 301-4. doi:10.1056/NEJMp1006304
20. Chin, Lynda et al. “Cancer genomics: from discovery science to personalized medicine.” Nature medicine vol. 17,3 (2011): 297-303. doi:10.1038/nm.2323
21. Topol, Eric J. “High-performance medicine: the convergence of human and artificial intelligence.” Nature medicine vol. 25,1 (2019): 44-56. doi:10.1038/s41591-018-0300-7
22. Elemento, Olivier et al. “Artificial intelligence in cancer research, diagnosis and therapy.” Nature reviews. Cancer vol. 21,12 (2021): 747-752. doi:10.1038/s41568-021-00399-1
23. Stegle, Oliver et al. “Computational and analytical challenges in single-cell transcriptomics.” Nature reviews. Genetics vol. 16,3 (2015): 133-45. doi:10.1038/nrg3833
24. Argelaguet, Ricard et al. “Computational principles and challenges in single-cell data integration.” Nature biotechnology vol. 39,10 (2021): 1202-1215. doi:10.1038/s41587-021-00895-7
25. Min, Seonwoo et al. “Deep learning in bioinformatics.” Briefings in bioinformatics vol. 18,5 (2017): 851-869. doi:10.1093/bib/bbw068
26. Wainberg, Michael et al. “Deep learning in biomedicine.” Nature biotechnology vol. 36,9 (2018): 829-838. doi:10.1038/nbt.4233
27. Mahmud, Mufti et al. “Applications of Deep Learning and Reinforcement Learning to Biological Data.” IEEE transactions on neural networks and learning systems vol. 29,6 (2018): 2063-2079. doi:10.1109/TNNLS.2018.2790388

28. Woolley, Anita Williams et al. “Evidence for a collective intelligence factor in the performance of human groups.” Science (New York, N.Y.) vol. 330,6004 (2010): 686-8. doi:10.1126/science.1193147
29. Karpatne, Anuj, et al. "Theory-guided data science: A new paradigm for scientific discovery from data." IEEE Transactions on knowledge and data engineering vol. 29,10 (2017): 2318-2331. doi:10.1109/TKDE.2017.2720168
30. Holzinger, Andreas et al. “AI for life: Trends in artificial intelligence for biotechnology.” New biotechnology vol. 74 (2023): 16-24. doi:10.1016/j.nbt.2023.02.001
31. Messeri, Lisa, and M J Crockett. “Artificial intelligence and illusions of understanding in scientific research.” Nature vol. 627,8002 (2024): 49-58. doi:10.1038/s41586-024-07146-0
32. Sourati, Jamshid, and James A Evans. “Accelerating science with human-aware artificial intelligence.” Nature human behaviour vol. 7,10 (2023): 1682-1696. doi:10.1038/s41562-023-01648-z
33. Inkpen, Kori, et al. "Advancing Human-AI complementarity: The impact of user expertise and algorithmic tuning on joint decision making." ACM Transactions on Computer-Human Interaction vol. 30,5 (2023): 1-29. doi:10.1145/3534561
34. Amirkhani, Abdollah, and Amir Hossein Barshooi. "Consensus in multi-agent systems: a review." Artificial Intelligence Review vol. 55,5 (2022): 3897-3935. doi:10.1007/s10462-021-10097-x
35. Wu, Qingyun, et al. "Autogen: Enabling next-gen llm applications via multi-agent conversation framework." arXiv preprint arXiv:2308.08155 (2023). doi:10.48550/arXiv.2308.08155. Preprint.
36. Talebirad, Yashar, and Amirhossein Nadiri. "Multi-agent collaboration: Harnessing the power of intelligent llm agents." arXiv preprint arXiv:2306.03314 (2023). doi:10.48550/arXiv.2306.03314. Preprint.
37. Thirunavukarasu, Arun James et al. “Large language models in medicine.” Nature medicine vol. 29,8 (2023): 1930-1940. doi:10.1038/s41591-023-02448-8

38. Singhal, Karan et al. “Large language models encode clinical knowledge.” Nature vol. 620,7972 (2023): 172-180. doi:10.1038/s41586-023-06291-2
39. Zhao, Wayne Xin, et al. "A survey of large language models." arXiv preprint arXiv:2303.18223 (2023). doi:10.48550/arXiv.2303.18223. Preprint.
40. Yao, Shunyu, et al. "Tree of thoughts: Deliberate problem solving with large language models." Advances in Neural Information Processing Systems 36 (2024).
41. Xi, Zhiheng, et al. "The rise and potential of large language model based agents: A survey." arXiv preprint arXiv:2309.07864 (2023). doi:10.48550/arXiv.2309.07864. Preprint.
42. Wang, Lei, et al. "A survey on large language model based autonomous agents." Frontiers of Computer Science vol. 18,6 (2024): 1-26. doi: 10.1007/s11704-024-40231-1
43. Cheng, Yuheng, et al. "Exploring Large Language Model based Intelligent Agents: Definitions, Methods, and Prospects." arXiv preprint arXiv:2401.03428 (2024). doi:10.48550/arXiv.2401.03428. Preprint.
44. Sikkema, Lisa et al. “An integrated cell atlas of the lung in health and disease.” Nature medicine vol. 29,6 (2023): 1563-1577. doi:10.1038/s41591-023-02327-2
45. Hadi, Muhammad Usman, et al. "A survey on large language models: Applications, challenges, limitations, and practical usage." Authorea Preprints (2023). doi:10.36227/techrxiv.23589741.v1. Preprint.
46. Lander, E S et al. “Initial sequencing and analysis of the human genome.” Nature vol. 409,6822 (2001): 860-921. doi:10.1038/35057062
47. International Human Genome Sequencing Consortium. “Finishing the euchromatic sequence of the human genome.” Nature vol. 431,7011 (2004): 931-45. doi:10.1038/nature03001
48. Gibbs, Richard A. “The Human Genome Project changed everything.” Nature reviews. Genetics vol. 21,10 (2020): 575-576. doi:10.1038/s41576-020-0275-3

49. Nurk, Sergey et al. “The complete sequence of a human genome.” Science (New York, N.Y.) vol. 376,6588 (2022): 44-53. doi:10.1126/science.abj6987
50. Mardis, Elaine R. “Next-generation DNA sequencing methods.” Annual review of genomics and human genetics vol. 9 (2008): 387-402. doi:10.1146/annurev.genom.9.081307.164359
51. Goodwin, Sara et al. “Coming of age: ten years of next-generation sequencing technologies.” Nature reviews. Genetics vol. 17,6 (2016): 333-51. doi:10.1038/nrg.2016.49
52. Wang, Zhong et al. “RNA-Seq: a revolutionary tool for transcriptomics.” Nature reviews. Genetics vol. 10,1 (2009): 57-63. doi:10.1038/nrg2484
53. Davidson, Eric H, and Douglas H Erwin. “Gene regulatory networks and the evolution of animal body plans.” Science (New York, N.Y.) vol. 311,5762 (2006): 796-800. doi:10.1126/science.1113832
54. Badia-I-Mompel, Pau et al. “Gene regulatory network inference in the era of single-cell multi-omics.” Nature reviews. Genetics vol. 24,11 (2023): 739-754. doi:10.1038/s41576-023-00618-5
55. Feng, Suhua et al. “Epigenetic reprogramming in plant and animal development.” Science (New York, N.Y.) vol. 330,6004 (2010): 622-7. doi:10.1126/science.1190614
56. Sharma, Shikhar, Theresa K. Kelly, and Peter A. Jones. "Epigenetics in cancer." Carcinogenesis 31.1 (2010): 27-36. doi: 10.1093/carcin/bgp220
57. Deng, Qiaolin et al. “Single-cell RNA-seq reveals dynamic, random monoallelic gene expression in mammalian cells.” Science (New York, N.Y.) vol. 343,6167 (2014): 193-6. doi:10.1126/science.1245316
58. Carter, Benjamin, and Keji Zhao. “The epigenetic basis of cellular heterogeneity.” Nature reviews. Genetics vol. 22,4 (2021): 235-250. doi:10.1038/s41576-020-00300-0
59. Ciccarelli, Francesca D et al. “Toward automatic reconstruction of a highly resolved tree of life.” Science (New York, N.Y.) vol. 311,5765 (2006): 1283-7. doi:10.1126/science.1123061

60. Koonin, Eugene V. “The origin and early evolution of eukaryotes in the light of phylogenomics.” Genome biology vol. 11,5 (2010): 209. doi:10.1186/gb-2010-11-5-209
61. Hanahan, D, and R A Weinberg. “The hallmarks of cancer.” Cell vol. 100,1 (2000): 57-70. doi:10.1016/s0092-8674(00)81683-9
62. Hanahan, Douglas, and Robert A Weinberg. “Hallmarks of cancer: the next generation.” Cell vol. 144,5 (2011): 646-74. doi:10.1016/j.cell.2011.02.013
63. Dobson, Christopher M. “Protein folding and misfolding.” Nature vol. 426,6968 (2003): 884-90. doi:10.1038/nature02261
64. Collins, Francis S, and Harold Varmus. “A new initiative on precision medicine.” The New England journal of medicine vol. 372,9 (2015): 793-5. doi:10.1056/NEJMp1500523
65. Robinson, Dan R et al. “Integrative clinical genomics of metastatic cancer.” Nature vol. 548,7667 (2017): 297-303. doi:10.1038/nature23306
66. Cancer Genome Atlas Network. “Comprehensive molecular portraits of human breast tumours.” Nature vol. 490,7418 (2012): 61-70. doi:10.1038/nature11412
67. Carpenter, Anne E et al. “CellProfiler: image analysis software for identifying and quantifying cell phenotypes.” Genome biology vol. 7,10 (2006): R100. doi:10.1186/gb-2006-7-10-r100
68. Bannon, Dylan et al. “DeepCell Kiosk: scaling deep learning-enabled cellular image analysis with Kubernetes.” Nature methods vol. 18,1 (2021): 43-45. doi:10.1038/s41592-020-01023-0
69. Senior, Andrew W et al. “Improved protein structure prediction using potentials from deep learning.” Nature vol. 577,7792 (2020): 706-710. doi:10.1038/s41586-019-1923-7
70. Jumper, John et al. “Highly accurate protein structure prediction with AlphaFold.” Nature vol. 596,7873 (2021): 583-589. doi:10.1038/s41586-021-03819-2

71. Baek, Minkyung et al. “Accurate prediction of protein structures and interactions using a three-track neural network.” Science (New York, N.Y.) vol. 373,6557 (2021): 871-876. doi:10.1126/science.abj8754
72. Abramson, Josh et al. “Accurate structure prediction of biomolecular interactions with AlphaFold 3.” Nature, 10.1038/s41586-024-07487-w. 8 May. 2024, doi:10.1038/s41586-024-07487-w
73. Aibar, Sara et al. “SCENIC: single-cell regulatory network inference and clustering.” Nature methods vol. 14,11 (2017): 1083-1086. doi:10.1038/nmeth.4463
74. Bravo González-Blas, Carmen et al. “SCENIC+: single-cell multiomic inference of enhancers and gene regulatory networks.” Nature methods vol. 20,9 (2023): 1355-1367. doi:10.1038/s41592-023-01938-4
75. Nishino, Mizuki et al. “Monitoring immune-checkpoint blockade: response evaluation and biomarker development.” Nature reviews. Clinical oncology vol. 14,11 (2017): 655-668. doi:10.1038/nrclinonc.2017.88
76. Golub, T R et al. “Molecular classification of cancer: class discovery and class prediction by gene expression monitoring.” Science (New York, N.Y.) vol. 286,5439 (1999): 531-7. doi:10.1126/science.286.5439.531
77. Keiser, Michael J et al. “Predicting new molecular targets for known drugs.” Nature vol. 462,7270 (2009): 175-81. doi:10.1038/nature08506
78. Zhavoronkov, Alex et al. “Deep learning enables rapid identification of potent DDR1 kinase inhibitors.” Nature biotechnology vol. 37,9 (2019): 1038-1040. doi:10.1038/s41587-019-0224-x
79. Zhang, Linlin et al. “Crystal structure of SARS-CoV-2 main protease provides a basis for design of improved α-ketoamide inhibitors.” Science (New York, N.Y.) vol. 368,6489 (2020): 409-412. doi:10.1126/science.abb3405
80. Vandereyken, Katy et al. “Methods and applications for single-cell and spatial multi-omics.” Nature reviews. Genetics vol. 24,8 (2023): 494-515. doi:10.1038/s41576-023-00580-2

81. Cui, Haotian et al. “scGPT: toward building a foundation model for single-cell multi-omics using generative AI.” Nature methods, 10.1038/s41592-024-02201-0. 26 Feb. 2024, doi:10.1038/s41592-024-02201-0
82. Eid, John et al. “Real-time DNA sequencing from single polymerase molecules.” Science (New York, N.Y.) vol. 323,5910 (2009): 133-8. doi:10.1126/science.1162986
83. Keren, Hadas et al. “Alternative splicing and evolution: diversification, exon definition and function.” Nature reviews. Genetics vol. 11,5 (2010): 345-55. doi:10.1038/nrg2776
84. Gebert, Luca F R, and Ian J MacRae. “Regulation of microRNA function in animals.” Nature reviews. Molecular cell biology vol. 20,1 (2019): 21-37. doi:10.1038/s41580-018-0045-7
85. Rupaimoole, Rajesha, and Frank J Slack. “MicroRNA therapeutics: towards a new era for the management of cancer and other diseases.” Nature reviews. Drug discovery vol. 16,3 (2017): 203-222. doi:10.1038/nrd.2016.246
86. Kozomara, Ana et al. “miRBase: from microRNA sequences to function.” Nucleic acids research vol. 47,D1 (2019): D155-D162. doi:10.1093/nar/gky1141
87. Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems 30 (2017).
88. Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018). Doi:10.48550/arXiv.1810.04805. Preprint.
89. Radford, Alec, et al. "Improving language understanding by generative pre-training." (2018).
90. Brown, Tom, et al. "Language models are few-shot learners." Advances in neural information processing systems 33 (2020): 1877-1901.
91. Achiam, Josh, et al. "Gpt-4 technical report." arXiv preprint arXiv:2303.08774 (2023). doi:10.48550/arXiv.2303.08774. Preprint.

92. Touvron, Hugo, et al. "Llama: Open and efficient foundation language models." arXiv preprint arXiv:2302.13971 (2023). doi:10.48550/arXiv.2302.13971. Preprint.
93. Touvron, Hugo, et al. "Llama 2: Open foundation and fine-tuned chat models." arXiv preprint arXiv:2307.09288 (2023). doi:10.48550/arXiv.2307.09288. Preprint.
94. Raffel, Colin, et al. "Exploring the limits of transfer learning with a unified text-to-text transformer." Journal of machine learning research vol. 21,140 (2020): 1-67.
95. Liu, Yixin, et al. "Sora: A Review on Background, Technology, Limitations, and Opportunities of Large Vision Models." arXiv preprint arXiv:2402.17177 (2024). doi:10.48550/arXiv.2402.17177. Preprint.
96. Mnih, Volodymyr, et al. "Playing atari with deep reinforcement learning." arXiv preprint arXiv:1312.5602 (2013). doi:10.48550/arXiv.1312.5602. Preprint.
97. Hernandez-Leal, Pablo, Bilal Kartal, and Matthew E. Taylor. "A survey and critique of multiagent deep reinforcement learning." Autonomous Agents and Multi-Agent Systems vol. 33,6 (2019): 750-797. doi: 10.1007/s10458-019-09421-1
98. Gao, Shanghua, et al. "Empowering Biomedical Discovery with AI Agents." arXiv preprint arXiv:2404.02831 (2024). doi:10.48550/arXiv.2404.02831. Preprint.
99. Park, Joon Sung, et al. "Generative agents: Interactive simulacra of human behavior." Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology (2023).
100. Yang, Hui, Sifu Yue, and Yunzhong He. "Auto-gpt for online decision making: Benchmarks and additional opinions." arXiv preprint arXiv:2306.02224 (2023). doi:10.48550/arXiv.2306.02224. Preprint.
101. Shen, Yongliang, et al. "Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face." Advances in Neural Information Processing Systems 36 (2024).
102. Boiko, Daniil A et al. “Autonomous chemical research with large language models.” Nature vol. 624,7992 (2023): 570-578. doi:10.1038/s41586-023-06792-0
103. M. Bran, Andres, et al. "Augmenting large language models with chemistry tools." Nature Machine Intelligence (2024): 1-11. doi:10.1038/s42256-024-00832-8

104. Talebirad, Yashar, and Amirhossein Nadiri. "Multi-agent collaboration: Harnessing the power of intelligent llm agents." arXiv preprint arXiv:2306.03314 (2023). doi:10.48550/arXiv.2306.03314. Preprint.
105. Hong, Sirui, et al. "Metagpt: Meta programming for multi-agent collaborative framework." arXiv preprint arXiv:2308.00352 (2023). doi:10.48550/arXiv.2308.00352. Preprint.
106. Li, Guohao, et al. "Camel: Communicative agents for" mind" exploration of large language model society." Advances in Neural Information Processing Systems 36 (2024).
107. Shinn, Noah, et al. "Reflexion: Language agents with verbal reinforcement learning." Advances in Neural Information Processing Systems 36 (2024).
108. Regev, Aviv et al. “The Human Cell Atlas.” eLife vol. 6 e27041. 5 Dec. 2017, doi:10.7554/eLife.27041
109. Eisenstein, Michael. “Seven technologies to watch in 2024.” Nature vol. 625,7996 (2024): 844-848. doi:10.1038/d41586-024-00173-x
110. Xu, Chenling et al. “Probabilistic harmonization and annotation of single-cell transcriptomics data with deep generative models.” Molecular systems biology vol. 17,1 (2021): e9620. doi:10.15252/msb.20209620
111. Badia-I-Mompel, Pau et al. “decoupleR: ensemble of computational methods to infer biological activities from omics data.” Bioinformatics advances vol. 2,1 vbac016. 8 Mar. 2022, doi:10.1093/bioadv/vbac016
112. Hou, Wenpin, and Zhicheng Ji. “Assessing GPT-4 for cell type annotation in single-cell RNA-seq analysis.” Nature methods, 10.1038/s41592-024-02235-4. 25 Mar. 2024, doi:10.1038/s41592-024-02235-4
113. White, Jules, et al. "A prompt pattern catalog to enhance prompt engineering with chatgpt." arXiv preprint arXiv:2302.11382 (2023). doi:10.48550/arXiv.2302.11382. Preprint.
114. Chrisley, Ron. "Embodied artificial intelligence." Artificial intelligence vol. 149,1 (2003): 131-150.

115. Jin, Dongdong, and Li Zhang. "Embodied intelligence weaves a better future." Nature Machine Intelligence vol. 2,11 (2020): 663-664. doi:10.1038/s42256-020-00250-6
116. Gupta, Agrim et al. “Embodied intelligence via learning and evolution.” Nature communications vol. 12,1 5721. 6 Oct. 2021, doi:10.1038/s41467-021-25874-z
117. Ha, David, and Jürgen Schmidhuber. "World models." arXiv preprint arXiv:1803.10122 (2018). doi:10.48550/arXiv.1803.10122. Preprint.
118. Juarez, Maria G., Vicente J. Botti, and Adriana S. Giret. "Digital twins: Review and challenges." Journal of Computing and Information Science in Engineering vol. 21,3 (2021): 030802. doi:10.1115/1.4050244
119. Habermann, Arun C et al. “Single-cell RNA sequencing reveals profibrotic roles of distinct epithelial and mesenchymal lineages in pulmonary fibrosis.” Science advances vol. 6,28 eaba1972. 8 Jul. 2020, doi:10.1126/sciadv.aba1972
120. Natri, Heini M et al. “Cell type-specific and disease-associated eQTL in the human lung.” bioRxiv : the preprint server for biology 2023.03.17.533161. (2023). doi:10.1101/2023.03.17.533161. Preprint.
121. Deprez, Marie et al. “A Single-Cell Atlas of the Human Healthy Airways.” American journal of respiratory and critical care medicine vol. 202,12 (2020): 1636-1645. doi:10.1164/rccm.201911-2199OC
122. Travaglini, Kyle J et al. “A molecular cell atlas of the human lung from single-cell RNA sequencing.” Nature vol. 587,7835 (2020): 619-625. doi:10.1038/s41586-020-2922-4
123. Morse, Christina et al. “Proliferating SPP1/MERTK-expressing macrophages in idiopathic pulmonary fibrosis.” The European respiratory journal vol. 54,2 1802441. 22 Aug. 2019, doi:10.1183/13993003.02441-2018
124. Madissoon, E et al. “scRNA-seq assessment of the human lung, spleen, and esophagus tissue stability after cold preservation.” Genome biology vol. 21,1 1. 31 Dec. 2019, doi:10.1186/s13059-019-1906-x

125. Bharat, Ankit et al. “Lung transplantation for patients with severe COVID-19.” Science translational medicine vol. 12,574 (2020): eabe4282. doi:10.1126/scitranslmed.abe4282
126. Reyfman, Paul A et al. “Single-Cell Transcriptomic Analysis of Human Lung Provides Insights into the Pathobiology of Pulmonary Fibrosis.” American journal of respiratory and critical care medicine vol. 199,12 (2019): 1517-1536. doi:10.1164/rccm.201712-2410OC
127. Vieira Braga, Felipe A et al. “A cellular census of human lungs identifies novel cell states in health and in asthma.” Nature medicine vol. 25,7 (2019): 1153-1163. doi:10.1038/s41591-019-0468-5
128. Goldfarbmuren, Katherine C et al. “Dissecting the cellular specificity of smoking effects and reconstructing lineages in the human airway epithelium.” Nature communications vol. 11,1 2485. 19 May. 2020, doi:10.1038/s41467-020-16239-z
129. Zeng, Hongkui. "What is a cell type and how to define it?." Cell 185.15 (2022): 2739-2755. doi:10.1016/j.cell.2022.06.031
130. Church, Deanna M et al. “Extending reference assembly models.” Genome biology vol. 16,1 13. 24 Jan. 2015, doi:10.1186/s13059-015-0587-3
131. Harrison, Peter W et al. “Ensembl 2024.” Nucleic acids research vol. 52,D1 (2024): D891-D899. doi:10.1093/nar/gkad1049
132. Lun, Aaron T L et al. “Pooling across cells to normalize single-cell RNA sequencing data with many zero counts.” Genome biology vol. 17 75. 27 Apr. 2016, doi:10.1186/s13059-016-0947-7
133. Polański, Krzysztof et al. “BBKNN: fast batch alignment of single cell transcriptomes.” Bioinformatics (Oxford, England) vol. 36,3 (2020): 964-965. doi:10.1093/bioinformatics/btz625
134. Haghverdi, Laleh et al. “Batch effects in single-cell RNA-sequencing data are corrected by matching mutual nearest neighbors.” Nature biotechnology vol. 36,5 (2018): 421-427. doi:10.1038/nbt.4091

135. Korsunsky, Ilya et al. “Fast, sensitive and accurate integration of single-cell data with Harmony.” Nature methods vol. 16,12 (2019): 1289-1296. doi:10.1038/s41592-019-0619-0
136. Hie, Brian et al. “Efficient integration of heterogeneous single-cell transcriptomes using Scanorama.” Nature biotechnology vol. 37,6 (2019): 685-691. doi:10.1038/s41587-019-0113-3
137. Lotfollahi, Mohammad et al. “scGen predicts single-cell perturbation responses.” Nature methods vol. 16,8 (2019): 715-721. doi:10.1038/s41592-019-0494-8
138. Lopez, Romain et al. “Deep generative modeling for single-cell transcriptomics.” Nature methods vol. 15,12 (2018): 1053-1058. doi:10.1038/s41592-018-0229-2
139. Stuart, Tim et al. “Comprehensive Integration of Single-Cell Data.” Cell vol. 177,7 (2019): 1888-1902.e21. doi:10.1016/j.cell.2019.05.031
140. Luecken, Malte D et al. “Benchmarking atlas-level data integration in single-cell genomics.” Nature methods vol. 19,1 (2022): 41-50. doi:10.1038/s41592-021-01336-8
141. Wolf, F Alexander et al. “SCANPY: large-scale single-cell gene expression data analysis.” Genome biology vol. 19,1 15. 6 Feb. 2018, doi:10.1186/s13059-017-1382-0
142. Hu, Congxue et al. “CellMarker 2.0: an updated database of manually curated cell markers in human/mouse and web tools based on scRNA-seq data.” Nucleic acids research vol. 51,D1 (2023): D870-D876. doi:10.1093/nar/gkac947
143. Bio-OS Team. “Bio-OS: Open System for Bioinformatics Analysis.” Https://Github.Com/Bio-OS, 19 May 2024, https://github.com/Bio-OS.
144. Lotfollahi, Mohammad et al. “Mapping single-cell data to reference atlases by transfer learning.” Nature biotechnology vol. 40,1 (2022): 121-130. doi:10.1038/s41587-021-01001-7

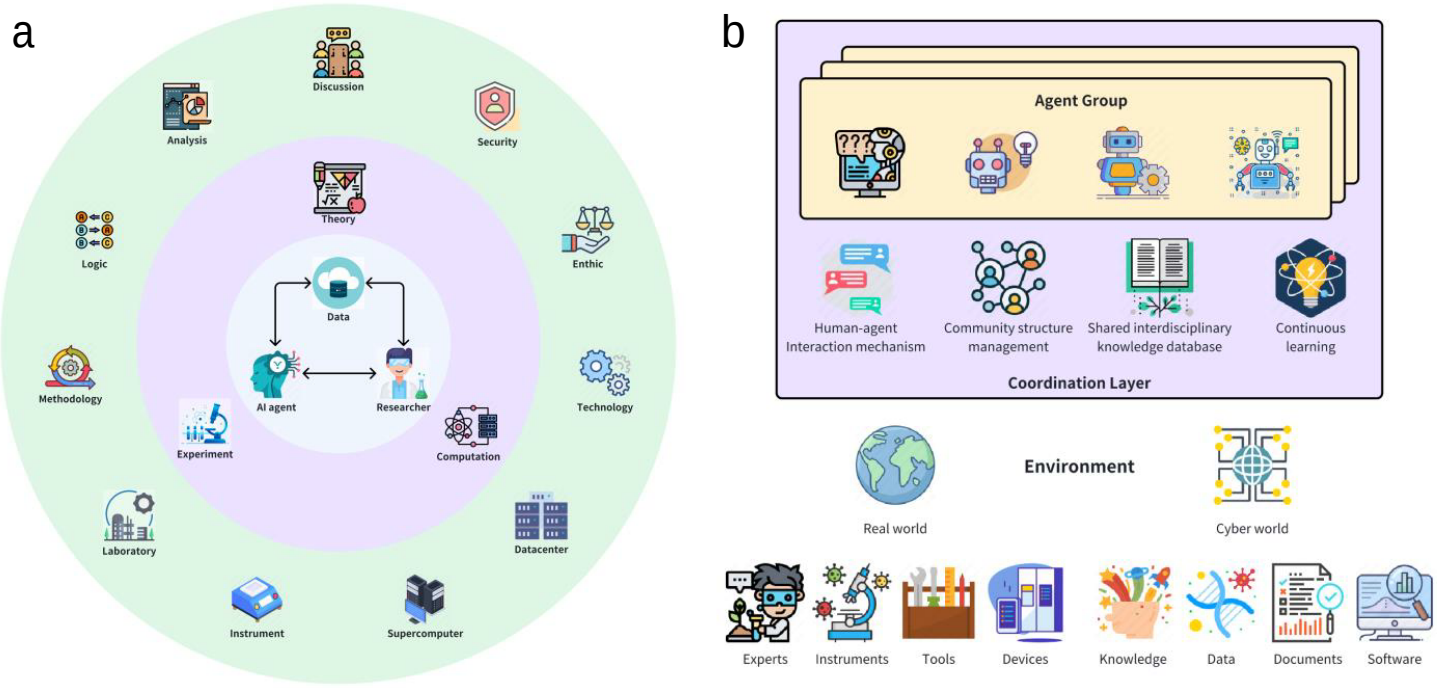


Figure 1. a. Data-intelligence-intensive scientific research paradigm. Synergizing human expertise with AI capabilities forms collective intelligence, that leverage big biological data to propel hypothesis-free exploratory research driven by a data-intelligence-intensive methodology. b. The multi-agent system (DII-MAS) based on LLMs. The architecture of DII-MAS comprises multiple agents with diverse roles, a coordination layer, and the environment.

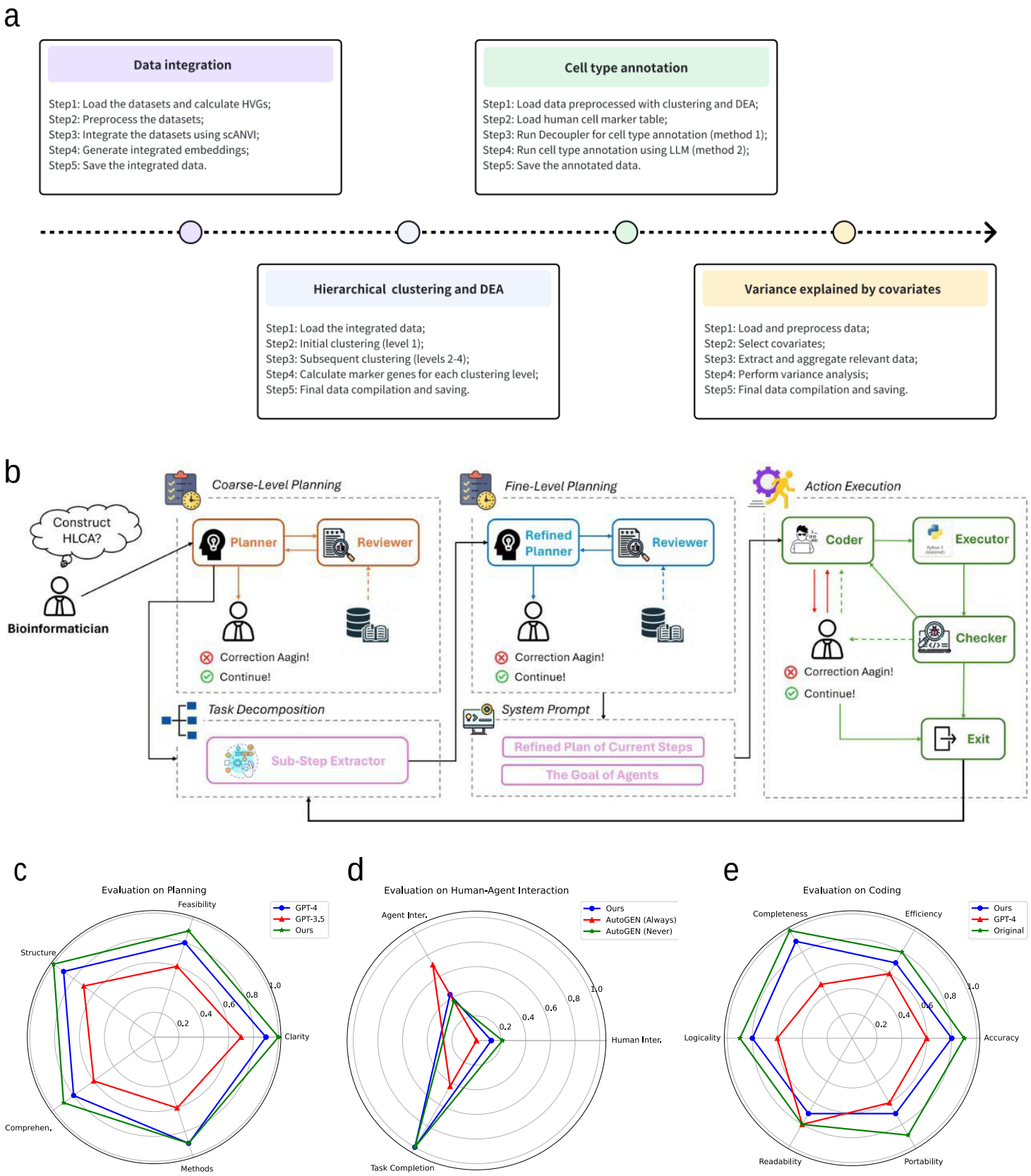


Figure 2. a. the workflow for constructing human lung cell atlas. Referencing the HLCA study, the workflow for this task is collaboratively generated by researchers and DII-MAS, comprising four sub-tasks. b. The collaborative process between researchers and DII-MAS. DII-MAS adopts a consistent procedure for each sub-task, comprising coarse-level planning, fine-level planning, and action execution. c. task planning evaluation. d. human-agent interaction evaluation. e. code quality evaluation.

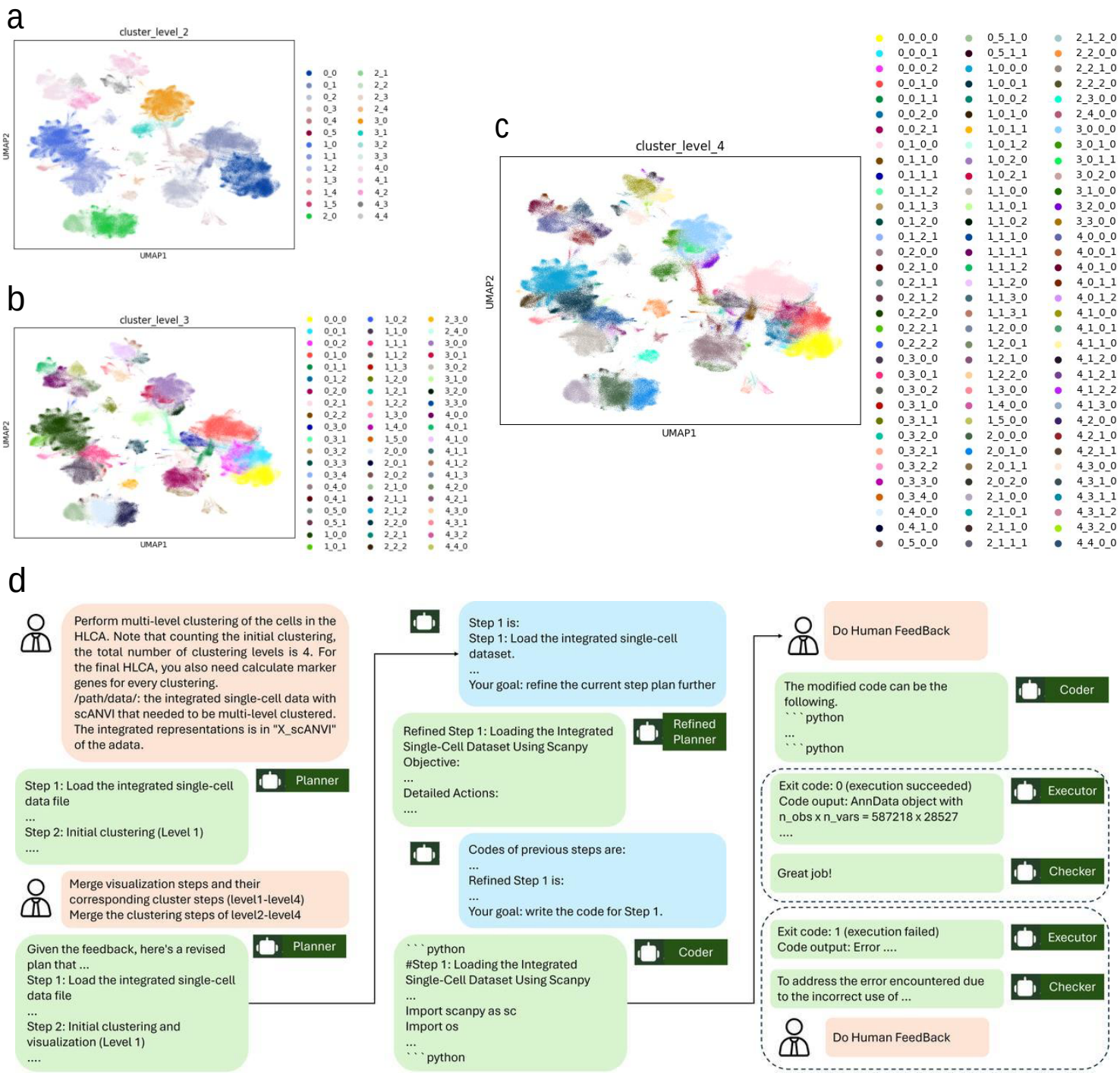


Figure 3. a-c. Visualizations of hierarchical clustering results. Figures a-c illustrate the hierarchical cell cluster from level 2 to level 4 after integration, where hierarchical clustering is performed using the embeddings of scANVI algorithm. d. The human-agent interaction process of hierarchical clustering sub-task.

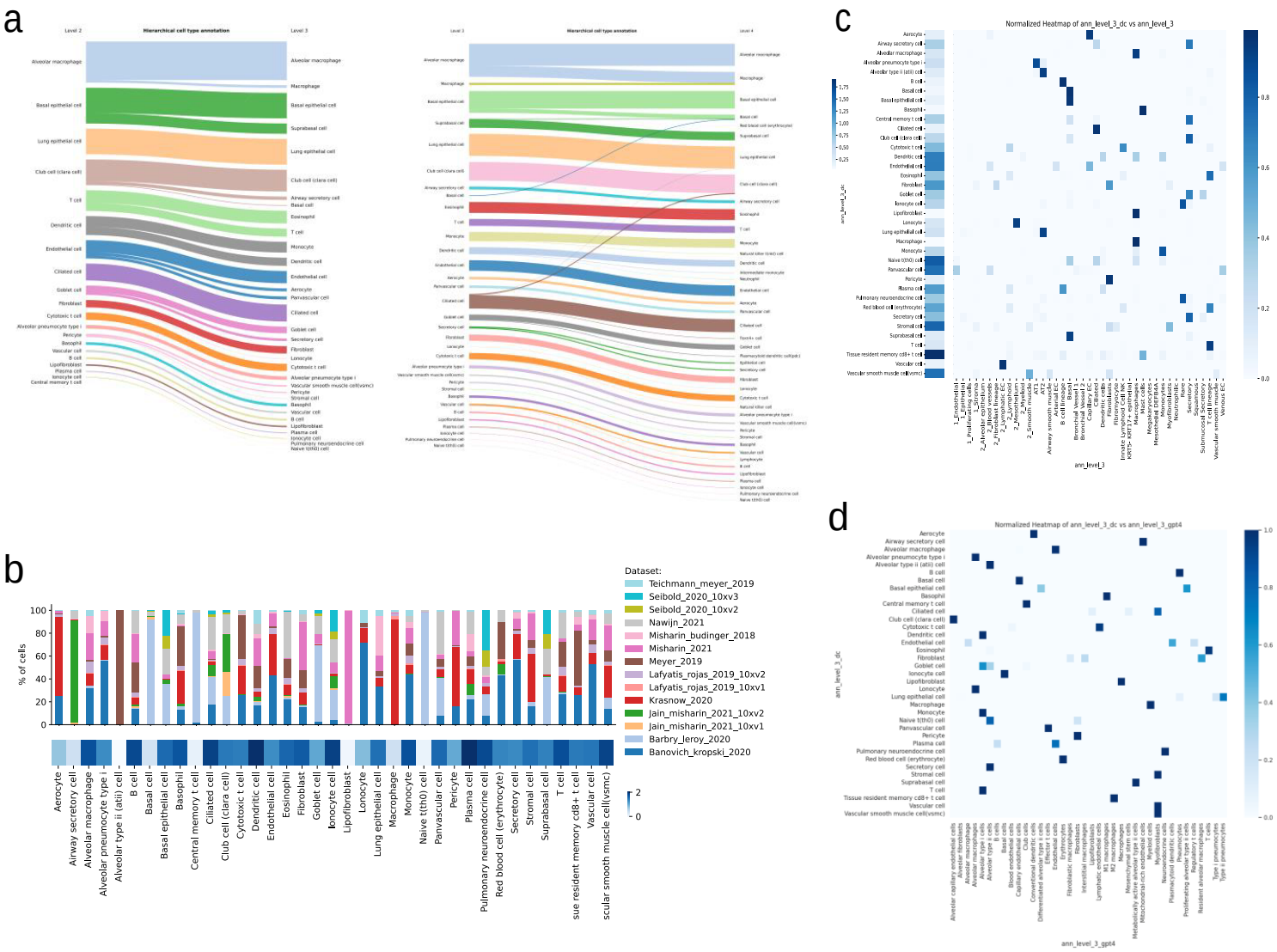


Figure 4. a. cell type annotations by Decoupler. DII-MAS employs the Decoupler tool to annotate the cell clusters from level 2 to level 4. b. the diversity of original independent datasets in each cell type at level 3. c. A Heatmap demonstrates the Decoupler annotations against the original labels. d. A Heatmap demonstrates the Decoupler annotations against the LLM annotations.

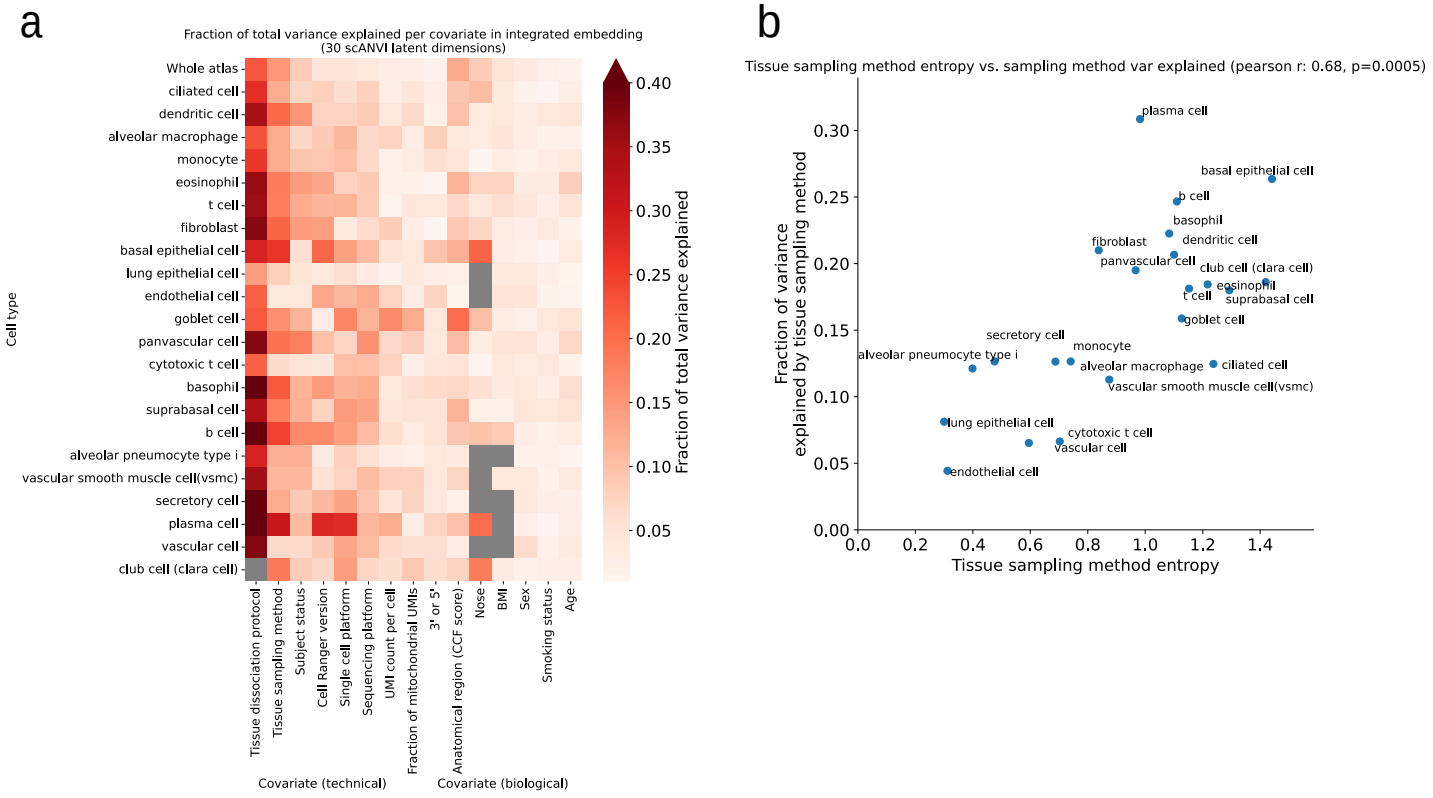
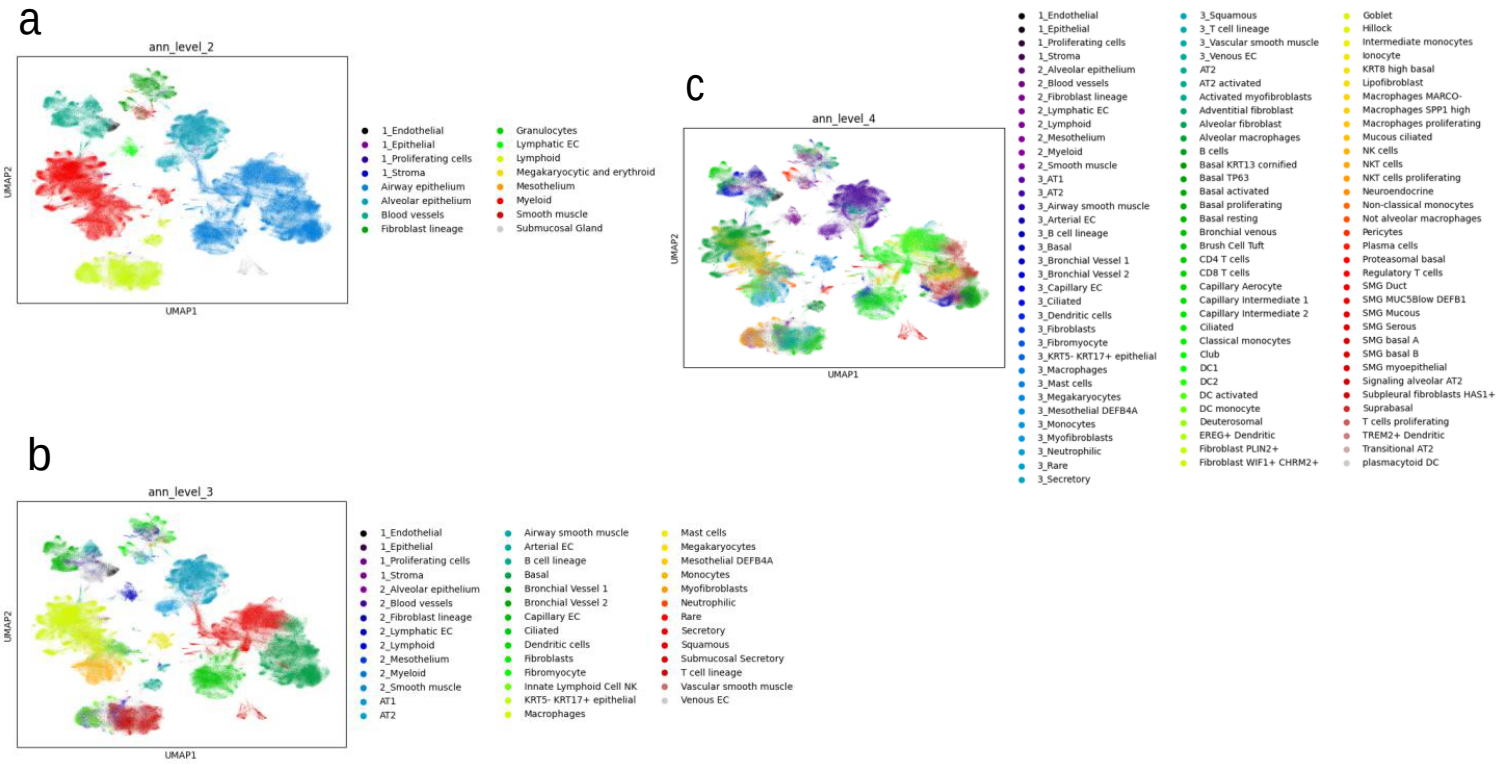


Figure 5. a. Variance analysis. DII-MAS employs the embeddings of scANVI to conduct variance analyses across different cell types and covariates. b. variation contributions of tissue sampling method for each cell type.



d

Step 1: Load the integrated single-cell data file

Utilize scanpy to load the .h5ad file, ensuring the dataset is correctly loaded by verifying the structure and the presence of the X_scANVI in the adata.

Step 2: Initial clustering and visualization (Level 1)

(1) Perform the first level of clustering on the X_scANVI representations using scanpy.tl.leiden or scanpy.tl.louvain, adjusting the resolution parameter to achieve an optimal balance between the number of clusters and their biological relevance. Save the result in adata.obs['cluster_level_1'].

(2) Immediately follow with a UMAP or t-SNE visualization to inspect the clustering result visually, coloring the plot by cluster_level_1 to distinguish different clusters. Save the visualization as level_1_clustering.png.

Sub-Steps in Multi-Level Cluster by Agent

Step 3: Sub-clustering and visualization for Levels 2, 3, and 4

(1) For each subsequent level of clustering (Levels 2, 3, and 4), iterate through each cluster identified in the previous level and perform sub-clustering. Adjust the resolution parameter as necessary for each level to ensure meaningful biological subdivisions. Save the sub-clustering results in adata.obs for each level (cluster_level_2, cluster_level_3, cluster_level_4).

(2) After each level of sub-clustering, generate a UMAP or t-SNE visualization for the new sub-clusters, coloring the plots by the current level's clustering result (cluster_level_2, cluster_level_3, cluster_level_4). Save the visualizations as level_X_clustering.png, where X corresponds to the clustering level (2, 3, or 4).

Step 4: Calculate Marker Genes for Each Cluster at Every Level

(1) Utilize scanpy.tl.rank_genes_groups to calculate marker genes for each cluster at every level of clustering, comparing each cluster to all other clusters at the same level. This step is crucial for identifying the biological significance of each cluster.

(2) Save the marker gene lists for each cluster and level in separate files, named marker_genes_level_X_cluster_Y.csv, where X is the level and Y is the cluster number.

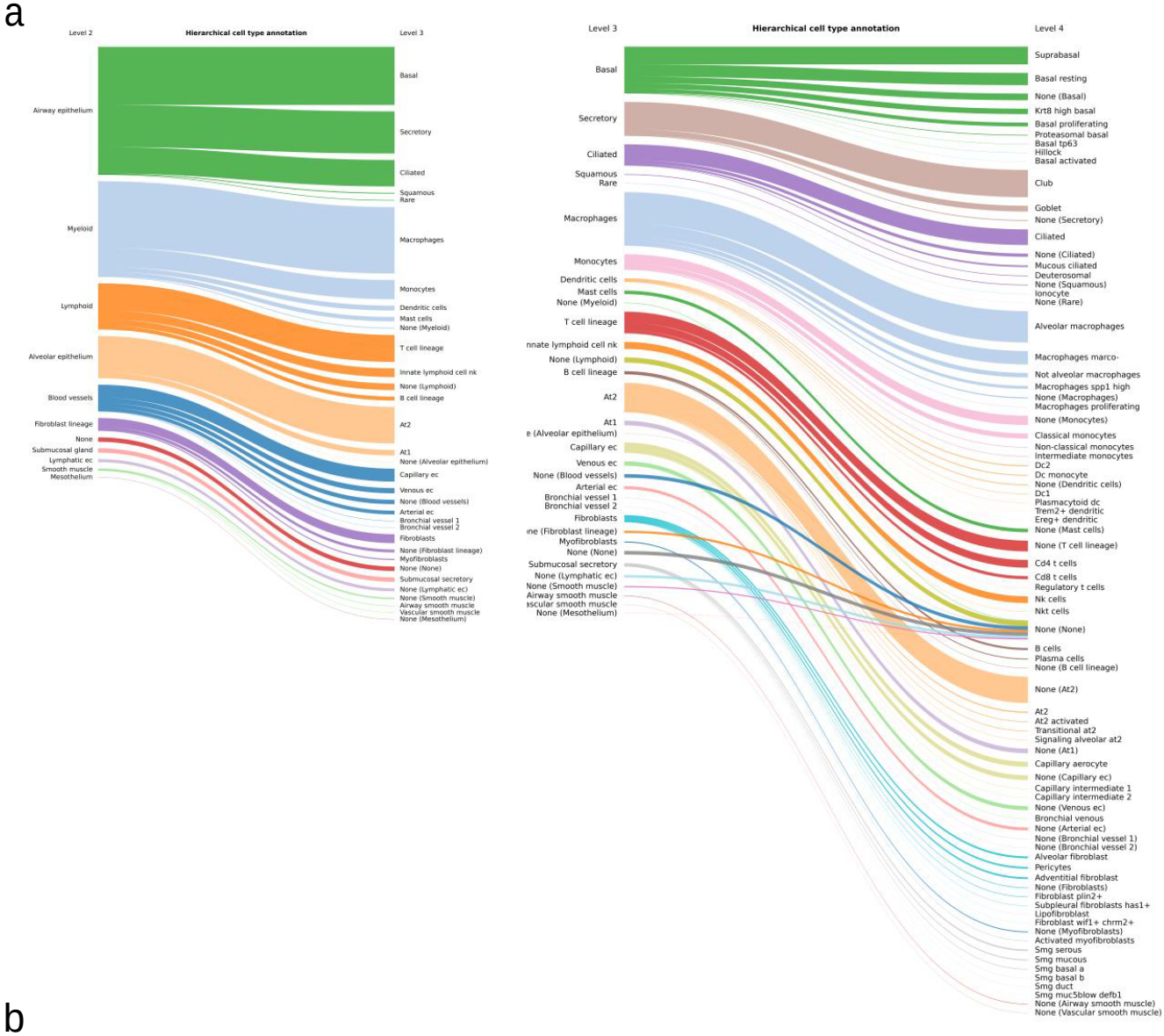
Step 5: Final Data Compilation and Saving

(1) Compile all clustering results, visualization files, and marker gene lists into a comprehensive report or a structured data file, ensuring clear distinction between different levels of clustering and associated marker genes.

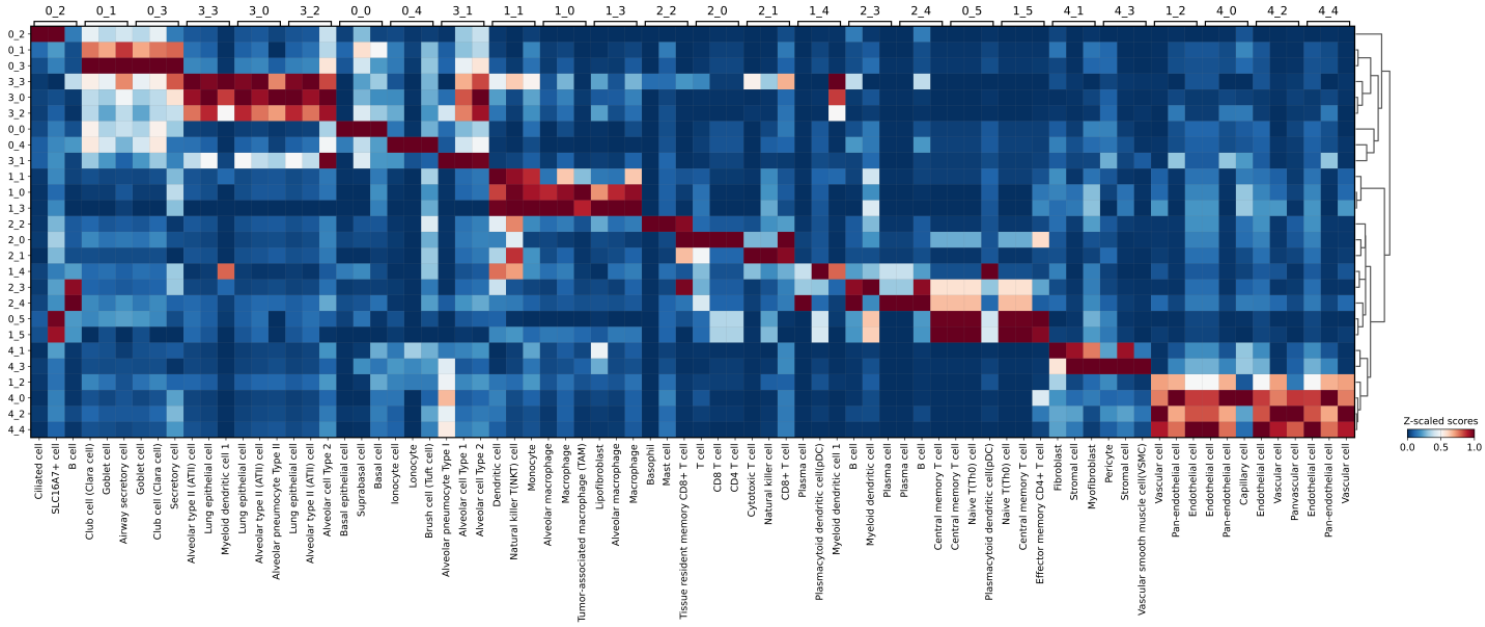
(2) Save the final compiled data as final_HLCA_clustering_and_markers.h5ad or in a similar structured format, providing a complete overview of the multi-level clustering process and its outcomes.

Supplementary Figure 1. a-c. Visualizations of original cell type labels. A considerable number of cells are filled categories prefixed with numbers and inherited from upper levels during preprocessing. d. Execution plan for the hierarchical clustering sub-task.

a



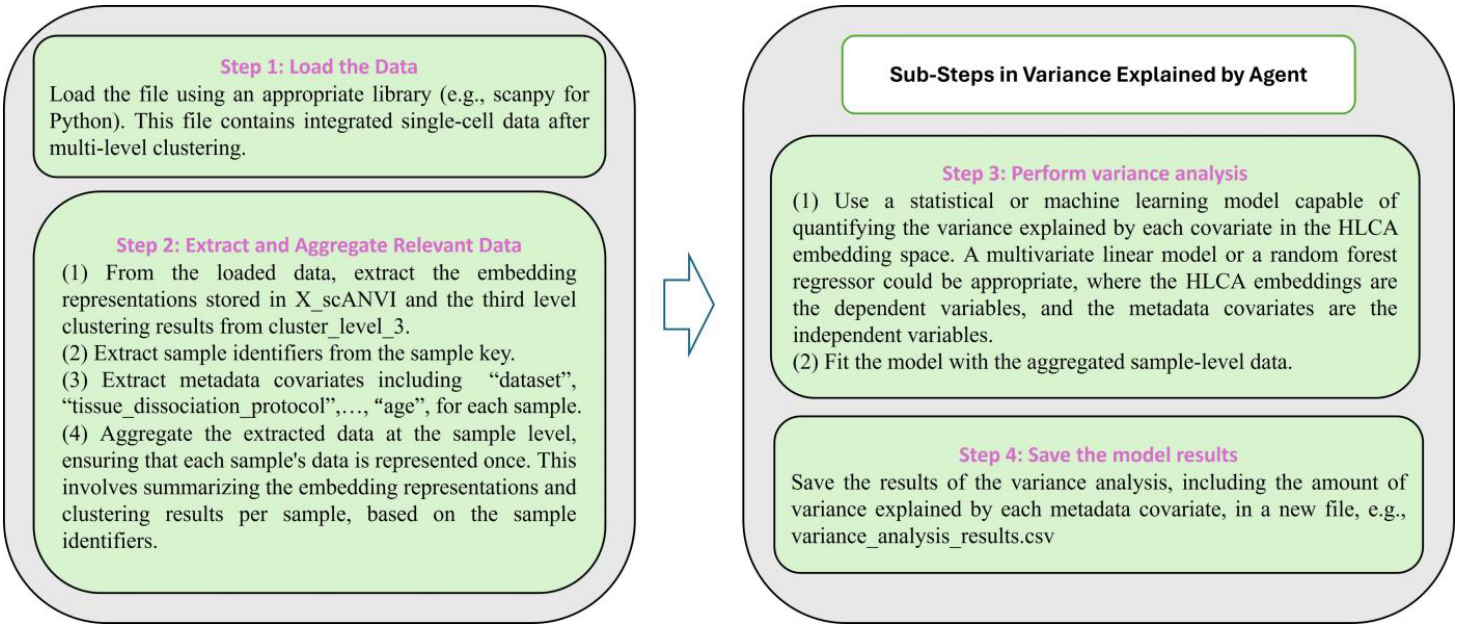
b



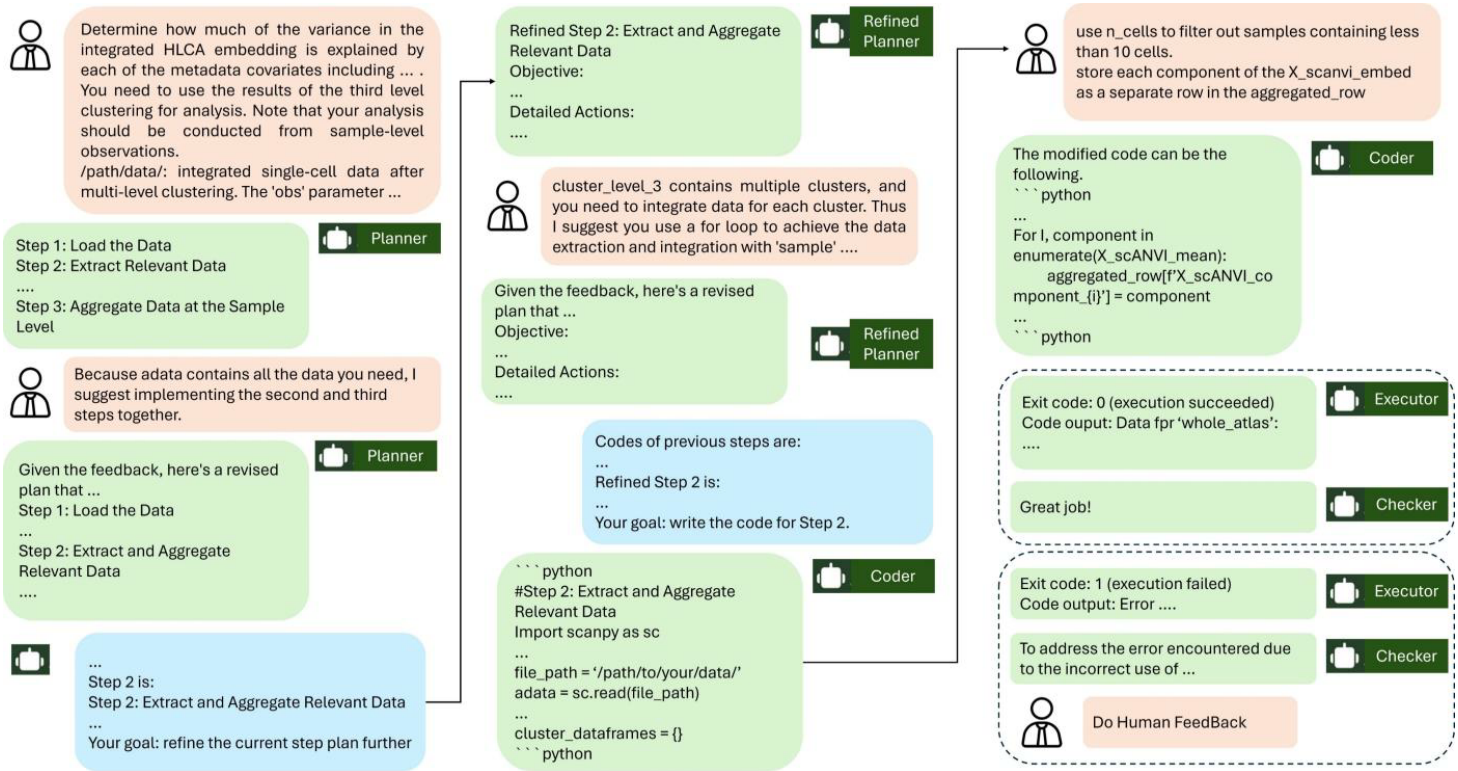
Supplementary Figure 2. a. Sankey diagram of original cell type labels. b. The Decoupler tool provides the top 3 cell type annotations for each cell subcluster at level 2.

Supplementary Figure 3. a. Sankey diagram of LLM annotations. b. A Heatmap demonstrates the LLM annotations against the original cell type labels.

a

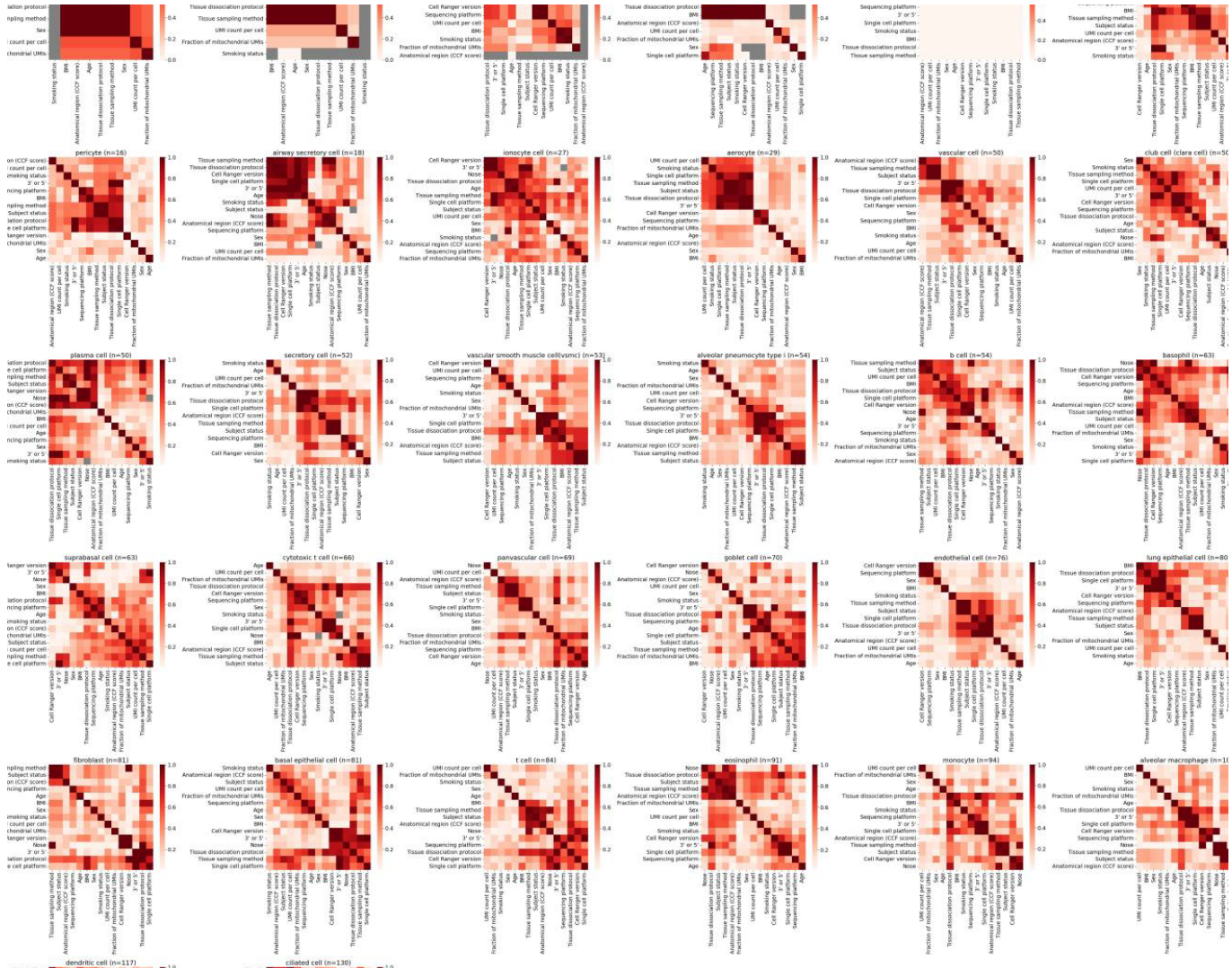


b



Supplementary Figure 4. a. Execution plan for the variance analysis sub-task. b. The human-agent interaction process of variance analysis sub-task.

a



Supplementary Figure 5. a. The correlation between covariates within each cell type.