PerturbAgent: An Agentic AI system for Analysis and Prediction of Genetic Perturbations

Kanglu Pei, Sisi Qu, Philip H. S. Torr, Jonathan G. Hedley, Christian Schroeder de Witt

University of Oxford
{jonathan.hedley, christian.schroeder}@eng.ox.ac.uk,

Abstract

We introduce PerturbAgent, a large language model (LLM)-based multi-agent system for single-cell genetic perturbation studies. In biomedical research, understanding cellular responses to perturbations is essential for interpreting gene function and regulatory pathways in single-cell data. Existing methods focus only on either single-cell analysis pipelines or perturbation prediction models, and often lack this necessary biological interpretation. PerturbAgent addresses these limitations, targeting both analysis and prediction tasks while also generating comprehensive biological interpretations with results grounded in mechanisms, pathways, and existing knowledge. We further propose MAST++, a general framework that evaluates agentic performance across profile, reasoning, perception, interaction, and memory, and complement it with biological validity assessments. On public single-cell Perturb-seq and RNA-seq datasets, PerturbAgent reliably achieves high task completion and delivers citation-backed biological summaries, representing progress toward practical and interpretable agent workflows for scientific discovery.

Introduction

LLM-based agents are systems that combine large language models with control logic for perception, reasoning, interaction, and memory; their growing use and popularity are reshaping scientific discovery, particularly in the drug discovery pipeline (Jumper et al. 2021; Wang et al. 2024a; Zhang et al. 2025). Unlike traditional models that make one-off predictions or analyses (Mosqueira-Rey et al. 2023), LLM-based agents can reason through rationales, invoke tools, and dynamically interact with environments (Gao et al. 2024a), other agents (Li et al. 2023; Wu et al. 2023), and humans. These capabilities position LLM-based agents as a promising direction for building intelligent systems in science.

Genetic perturbation studies are used to uncover gene function, regulatory networks, and disease mechanisms (Datlinger et al. 2017; Gasperini et al. 2019), making them central to early-stage drug discovery alongside broader biomedical research. Here, a *genetic perturbation* refers to a deliberate alteration of the expression or function of a gene g, and can be denoted as a causal intervention $\mathrm{do}(G = g)$, where $G \in \{0, 1, ..., n_{\mathrm{pert}}\}$ indexes the targeted gene ($G = 0$ denotes control cells, i.e. unperturbed condition, and n_{pert} is the number of perturbations). The perturbation effect is quantified by comparing the distribution of transcriptional response, $P(Y \mid \mathrm{do}(G = g))$ to the control baseline $P(Y \mid \mathrm{do}(G = 0))$, where $Y \in \mathbb{R}^{n_{\mathrm{genes}}}$ is the expression vector across all n_{genes} measured genes. Perturb-seq (Dixit et al. 2016) and large-scale CRISPR screens have enabled systematic profiling of thousands of perturbations across millions of single cells (Replogle et al. 2022; Adamson et al. 2016; Datlinger et al. 2017), generating rich data resources for discovery. At the scales typical of modern perturbation screens, the constraint shifts from running analyses to producing explainable, biologically grounded target choices; the automation of this process is a clear next step toward faster, more consistent interpretations of single-cell biology with clear relevance to target discovery.

However, most existing approaches for automatically extracting biological insights remain limited in scope. Methods tend to fall into two categories. *Analysis pipelines* process single-cell data through fixed workflows to produce statistical summaries(Zhou et al. 2024; Su, Long, and Zhang 2025; Xiao et al. 2024), but lack predictive and explanatory capability. *Prediction models* forecast transcriptional responses to perturbations (Ramakrishnan et al. 2025; Adduri et al. 2025; Roohani, Huang, and Leskovec 2024) but cannot directly produce biological insight from their output. Recent extensions have attempted to improve tool coverage and task generality, yet remain limited to workflow execution without deeper biological reasoning (Huang et al. 2025). This motivates the integration of *analysis*, *prediction*, *interpretation* and *the intelligence of agents* in a single framework.

We therefore introduce **PerturbAgent**, a multi-agent framework for automated discovery in genetic perturbation studies. PerturbAgent orchestrates agents equipped with reasoning, coding, tool use, and memory. By integrating existing analysis and modelling tools, it executes multi-step tasks, invokes bioinformatics and prediction modules, and generates biologically meaningful explanations. This establishes an explainable workflow that combines statistical analysis with transcriptional-response prediction, and mirrors how a human biologist plans, experiments, reflects, and interprets results.

Our approach makes three key contributions:

- **A novel agentic framework.** A multi-agent architecture

for genetic perturbation studies that integrates reasoning, coding, tool use, and memory.

- **Unified analysis–prediction–interpretation integration.** Integration of classical single-cell analysis with perturbation-prediction models in a single workflow, enabling the translation of outputs into structured biological explanations and hypotheses.
- **Comprehensive evaluation framework.** We extend the Multi-Agent System Failure Taxonomy (Cemri et al. 2025, MAST) to a general task-agnostic framework (MAST++), combined with task-specific metrics to jointly assess agentic performance and biological validity.

Related Work

Agents for Biological Analysis & Discovery

Early efforts to integrate LLMs into biological analysis emphasised *workflow automation* over scientific reasoning or interpretation. AutoBA is a single-agent system that automates multi-omics pipelines but remains bound to predefined workflows (Zhou et al. 2024).

Multi-agent systems (e.g. CellAgent (Xiao et al. 2024)) improve modularity via specialised agent roles; BioMaster (Zhou et al. 2024) further incorporates retrieval-augmented generation (RAG) for domain-specific information and reports higher task success across bioinformatics workflows. However, these systems still focus on dry-lab automation and quantitative outputs without interpretation.

General biomedical agents, e.g. Biomni (Huang et al. 2025), integrate CodeAct with extensive toolkits to generate code and visual summaries across various biomedical tasks. However, it does not yet provide deeper domain-specific downstream introspection beyond descriptive summaries.

To close the discovery loop, BioDiscoveryAgent (Roohani et al. 2024) uses LLM reasoning to propose candidates for gene perturbation. While promising, its hypotheses are limited to gene lists without mechanistic explanations or quantitative prediction.

Perturbation Prediction Models

Early generative models such as CPA (Lotfollahi et al. 2023) and sVAE+(Lopez et al. 2023) use latent-variable generative decoders with fixed likelihood assumptions, which can limit modelling of heterogeneous, non-Gaussian expression responses and typically do not generalise to unseen target genes. More recent approaches add prior biological structure to improve generalisability: GEARS (Roohani, Huang, and Leskovec 2024) uses knowledge graphs built from gene co-expression and gene ontology (GO) relationships, while foundation models such as scGPT (Cui et al. 2024) and scFoundation (Hao et al. 2024) are pre-trained on large single-cell atlases and fine-tuned for perturbation prediction. However, benchmarking studies have highlighted that these models fail to outperform simple linear baselines on out-of-distribution (OOD) perturbations (Ahlmann-Eltze, Huber, and Anders 2025). Furthermore, most methods primarily optimise for mean shifts in expression rather than full distributional changes across cells.

Motivated by these limitations, LLMPert (Märtens, Donovan-Maiye, and Ferkinghoff-Borg 2024) introduces LLM-informed gene embeddings to support generalisation to unseen perturbations, and LLMHistPert (Ramakrishnan et al. 2025) extends this idea to distributional prediction by modelling per-gene expression as histograms, capturing higher-order statistics and cellular heterogeneity. Complementarily, STATE (Adduri et al. 2025) targets transfer across cellular and experimental contexts via "state" embeddings trained on large observational and perturbed corpora, improving robustness to shifts in population composition and context.

Overview of PerturbAgent

We design two main modules in PerturbAgent; **Analysis** and **Prediction**. *Analysis* extracts interpretable insights from perturbation data, such as differentially expressed genes, clustering patterns and pathway enrichment, which gives insights with biology domain expertise, facilitating new hypothesis generation; while *prediction* can infer perturbation outcomes, which offers a simulation of unseen genes, perturbations, or cellular contexts, providing exploratory references that guide future experimental design.

Additionally, both tasks share a common objective of **interpretation**: connecting quantitative results to biological understanding, implications and future directions, as well as discussing technical quality and uncertainty.

Architecture

PerturbAgent is implemented as a multi-agent system operating on a stateful LangGraph workflow. LangGraph provides a graph abstraction, where the state of the system is represented and updated as the workflow progresses. In PerturbAgent, the state tracks variables including loaded datasets, intermediate analysis results, model predictions, and memory records, ensuring the contextual information is preserved throughout the workflow.

The workflow (Fig. 1) consists of four nodes–*Generate*, *Execute*, *Critic*, and *Report*–in addition to the standard *Start* and *End* nodes. These nodes are connected by conditional or unconditional directed edges, allowing the system to transition between reasoning, execution, reflection, and reporting stages. Specifically:

- *Generate* produces reasoning traces, candidate solution or code as text;
- *Execute* runs generated code or tool calls;
- *Critic* reviews solution against task goal and criticise candidate solution, identifies errors, and provides feedbacks for improvement;
- *Report* synthesises biologically interpretable summaries.

Following the taxonomy by Gao et al. (2024b), AI agents typically include four key capabilities: *perception*, *interaction*, *reasoning*, and *memory*. Additionally, Wang et al. (2024a) includes a *profile* module to describe the agent roles. PerturbAgent incorporates all these modules within a three-agent setup, each with a specialised role and shared state access.

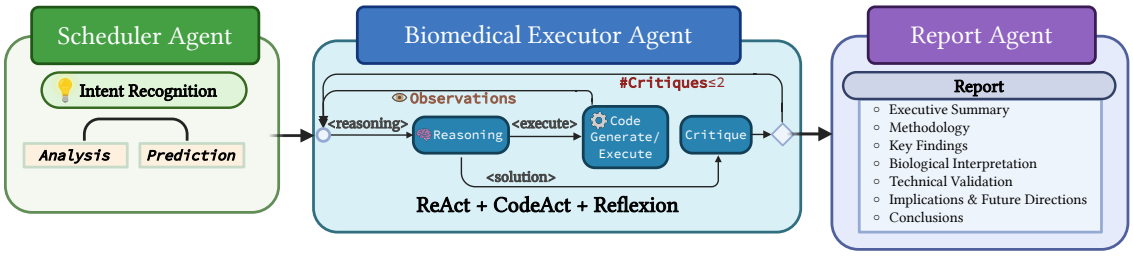

Figure 1: The illustration of PerturbAgent's architecture comprising three agents.

The **profiles** are embedded into system prompts at initialisation. The first is a **scheduler agent** for specifying the task type (analysis or prediction), make plans and coordinates workflow execution or termination. The second is a **biomedical specialist agent** responsible for reasoning, code generation, execution, and error handling. The third is a **report agent**, which can summarise final results into interpretable outputs for domain experts. Coordination between agents is achieved by passing LangGraph state objects with textual natural language and structural data files.

For **reasoning**, the biomedical specialist agent combines *ReAct* for stepwise planning with checklist updates (Yao et al. 2023), and *Reflexion* for reflection and feedback (Shinn et al. 2023). Both paradigms are applied Chain-of-Thought (CoT) prompting (Wei et al. 2023), supporting multi-step reasoning and improvement. For **interaction**, the biomedical specialist agent employs the *CodeAct* paradigm (Wang et al. 2024b) for action representation to interact with Python environment by only code, not natural language, enabling it to access analysis pipeline software, invoke pretrained models and query biological databases or APIs. For **short-term memory and context management**, PerturbAgent utilises LangGraph's *MemorySaver* (LangChain 2025) module to isolate sessions with different thread id and persist workflow states in a dictionary, so intermediate variables, execution logs and tool outputs remain available across nodes in the LangGraph workflow for the same conversation.

Task 1: Perturbation Analysis Module

Preprocessing. PerturbAgent supports both single-cell and pseudobulk[1] perturbation data, providing perspectives at both cell and perturbation levels. During preprocessing, the agent performs standard quality-control procedures, filtering low-quality cells and genes, normalising expression counts, and applying scaling or log transformations to ensure consistent data quality for downstream analyses.

Downstream analysis tasks. The analysis module handles two categories of tasks that reflect standard single-cell workflows or perturbation-specific analyses with rich interpretable outputs, both chosen to evaluate the agent's ability to orchestrate multi-step pipelines:

1. **classical single-cell analysis**: (i) identification of differentially expressed (DE) genes, (ii) unsupervised clustering of cell states, (iii) detection of cluster marker genes, and (iv) pathway enrichment analysis of DE genes.
2. **perturbation-specific single-cell/pseudobulk analyses** extend these pipelines to assess the effect sizes of perturbations for different aspects: pathway activity on strong and weak perturbations, correlation of perturbation effect profiles, and identification of functional modules such as integrator complexes.

These tasks require the agents to manage statistical workflows, handle multimodal resources, and generate biologically meaningful interpretations.

Supporting resources. PerturbAgent integrates multiple external biological knowledge bases covering gene functions, ontology, molecular pathways, protein interactions and complexes. These resources enable the agent to contextualise statistical outputs within relevant molecular systems and mechanisms, supporting higher-level biological reasoning.

Task 2: Perturbation Prediction Module

The second module of PerturbAgent focuses on **perturbation prediction**, aiming to forecast gene expression profiles under unseen perturbations. It takes single-cell datasets as input and produces predicted expression profiles along with evaluation results.

Prediction settings. Two prediction settings are considered to assess model generalisability:

1. **Within-cell-line:** The model trains on the cell line and tests on the unseen cells in the same cell line, with training set and testing set randomly split in 8:1.
2. **Cross-cell-line zero-shot:** A held-out cell line is unseen during training, which requires the model to extrapolate perturbation effects to novel cellular contexts without adaptation.

These settings evaluate a model's ability to capture in-distribution effects (within-cell-line), and generalise to completely novel contexts (zero-shot). We aim to use the result to further specify the roles of each model as the agent's tools.

Data, inputs, outputs. Let $\mathbf{X}_{\text{ctrl}} \in \mathbb{R}^{n_{\text{ctrl}} \times n_{\text{gene}}}$ denote the single-cell expression matrix of control cells and $\mathbf{X}_{\text{pert}} \in \mathbb{R}^{n_{\text{pert\_cells}} \times n_{\text{gene}}}$ the single-cell expression matrix of perturbed

[1]Pseudobulk refers to the perturbation-level expression matrix obtained by averaging single-cell expressions across cells with the same perturbation label.

cells, where $n_{\text{pert\_cells}}$ is the number of perturbed cells, n_{ctrl} is the number of control cells, and n_{gene} is the number of downstream genes. Each row corresponds to a single cell, and columns correspond to gene expression features. Metadata such as cell line identity, batch and perturbation labels are denoted by $\mathcal{C}$.

The goal of perturbation prediction is to learn a mapping

$$f : \left(\mathbf{X}_{\text{pert}}, \mathbf{X}_{\text{ctrl}}, \mathcal{C}\right) \mapsto \hat{\mathcal{Y}},$$

where $\hat{\mathcal{Y}}$ represents the predicted perturbation response, which can take several forms depending on the model interface:

- **Predicted pseudobulk profile:** $\hat{\mathbf{A}}_{\text{pert}} \in \mathbb{R}^{n_{\text{pert}} \times n_{\text{gene}}}$, representing the estimated mean expression of each gene across all cells under each perturbation (e.g. MLP Mean).
- **Predicted distributional profile:** $\hat{\mathbf{H}}_{\text{pert}} \in \mathbb{R}^{n_{\text{pert}} \times n_{\text{gene}} \times B}$, where each gene under each perturbation is represented by a B-bin probability vector describing its predicted expression distribution across cells (e.g. MLP Hist).
- **Predicted single-cell expression:** $\hat{\mathbf{X}}_{\text{pert}} \in \mathbb{R}^{n_{\text{cell}} \times n_{\text{gene}}}$, providing cell-level predictions of gene expression under each perturbation (e.g. STATE).

Downstream outputs include evaluation metrics and simulated results such as differential-expression (DE) gene lists or pathway enrichment derived from $\hat{\mathcal{Y}}$.

Model interface. PerturbAgent implements a general interface for pretrained perturbation prediction models. Each model is abstracted as Biomedical specialist agent's tool that receives input tensors $\left(\mathbf{X}_{\text{ctrl}}, \mathbf{X}_{\text{pert}}, \mathcal{C}\right)$ and outputs $\hat{\mathcal{Y}}$. This interface allows integration of models trained under different paradigms (mean-level, distributional, or cell-specific). The system can therefore adapt to future advances in perturbation modelling by simply registering new models in the framework.

Routing strategy. Model routing is manually specified according to the task objective and data characteristics. The scheduler agent selects appropriate models for within-cell-line or cross-cell-line settings based on i) prior model properties and ii) empirical performance results. Future extensions could enable adaptive routing and weighted fusion based on dynamic performance feedback.

Downstream analyses. Predicted profiles can be further integrated into the analysis pipelines from TASK1 to simulate new experimental conditions and generate hypotheses. For instance, DE testing and pathway enrichment can be performed on $\hat{\mathbf{X}}_{\text{pert}}$ to identify potential regulators and mechanistic targets.

System Design for Interpretability

Interpretability in PerturbAgent is achieved by the explicit design of its reasoning, memory structures, and report generation. Each reasoning iteration within the LangGraph workflow is fully traceable, producing structured outputs that reveal the agent's decision-making process. This design enables scientists not only to view final results but also examine the reasoning path that led to them.

Transparent reasoning and action traceability. Components of conversation history including reasoning traces (planning and reflections), agent actions in code, and candidate solutions are wrapped in XML-style tags. This representation allows the entire workflow to be parsed and searched, ensuring that each decision and tool invocations are explicitly traceable in system logs. Chain-of-thought (CoT) prompting is used for stepwise reasoning, while the Reflexion mechanism guides self-evaluations and refinement steps, providing a record of how errors are detected and corrected.

Structured reporting and biological interpretability. Beyond executing pipelines and producing numerical results, the Report Agent translates analytical and predictive outputs into biologically meaningful interpretations. It maps differential gene lists and perturbation responses to enriched pathways, functional modules, and cell-type–specific regulatory mechanisms, bridging quantitative computation with biological reasoning. Specifically, the report generation follows a prompt template for comprehensive biological interpretation, including: (i) executive summary; (ii) methodological description; (iii) key findings; (iv) mechanism-level interpretation (e.g., pathway- and network-level effects, comparison with known biology); (v) technical validation and discussion of data quality, reliability, potential confounders and limitations; (vi) implications and future directions (clinical relevance and suggested follow-up experiments). Full prompt is included in Appendix .

Through these design choices, PerturbAgent transforms automated computation into a transparent and interpretable scientific workflow.

Evaluation Framework

We evaluate **PerturbAgent** at two levels: (i) *task-specific evaluation* that quantify the biological validity of its analysis and prediction outputs, and (ii) *task-agnostic (agent-level) evaluation* that assess the robustness and efficiency of the agentic system.

Task-specific Evaluation

Analysis metrics. We assess two major aspects of single-cell and perturbation analyses:

1. *Pipeline correctness.* We inspect execution logs and outputs to verify: the appropriateness of quality control and normalisation, correctness of clustering, feature analysis, marker gene identification, pathway enrichment and invocation of APIs or databases for analysis.
2. *Interpretation quality.* Following the work from Ding et al. (2024), we design a prompt-based rubric to evaluate the automatically generated reports on the *logical coherence* of reasoning, *evaluability* of generated solution, and *interpretation* accuracy and clarity.

Prediction metrics. For perturbation prediction, we evaluate model performance at gene level and perturbation level. Formal definitions of these metrics are provided in Appendix .

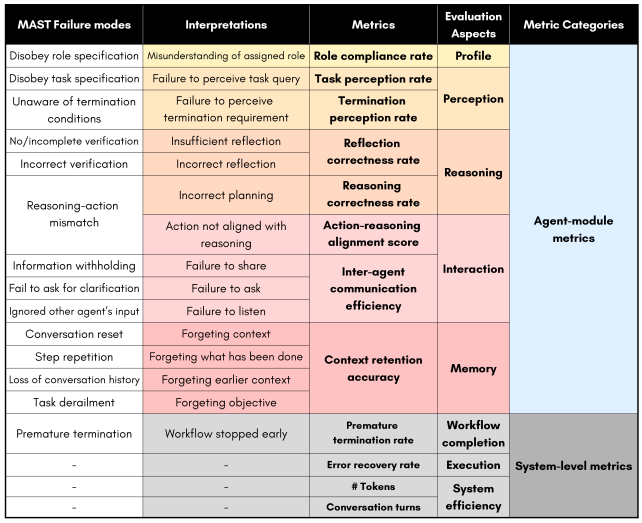

Figure 2: Summary of our task-agnostic evaluation framework (MAST++), and the corresponding MAST failure modes (Cemri et al. 2025).

(i) Gene-level differential expression (DE) metrics. These metrics assess whether the model captures the correct *direction* and *magnitude ranking* of gene-level changes of perturbation effects. We use (a) *directional agreement* to quantify the the proportion of genes whose predicted and true log-fold changes share the same sign, and (b) *Spearman rank correlation* between predicted and true fold changes for each perturbation.

(ii) Perturbation-level metric. At the perturbation level, we adopt the *perturbation discrimination score* from Adduri et al. (2025) to evaluate how closely each predicted transcriptional profile matches its corresponding ground truth relative to all other perturbations. This score ranks predicted post-perturbation profiles by their Manhattan distance to the observed profiles, with higher values indicating more accurate recovery of perturbation-specific expression patterns.

Task-agnostic (Agent-level) Evaluation

We propose a general task-agnostic evaluation framework, **MAST++**, which extends the Multi-Agent System Failure Taxonomy (Cemri et al. 2025, MAST) for scientific multi-agent systems. MAST++ reinterprets and reclassifies the 14 failure modes of MAST into five key agent modules (profile, perception, reasoning, interaction, and memory), and additionally introduces a new system-level aspect, addressing the gaps, overlaps, and ambiguities in MAST. This framework defines 12 interpretable metrics, providing a comprehensive basis for assessing multi-agent capabilities and reliability.

Agent-module metrics.

- **Profile:** role compliance rate (how often an agent follows its assigned role).
- **Perception:** task and termination perception rates (correct understanding of task objectives and stop conditions).

Table 1: Summary of Replogle Perturb-seq datasets (Replogle et al. 2022).

Dataset	#Cells	#Ctrls	#Perturbs	#Genes
K562_gwps	1,989,578	75,328	9,866	8,248
K562_essential	310,385	10,691	2,057	8,563
RPE1	247,914	11,485	2,393	8,749

- **Reasoning:** reflection correctness rate (validity of self-critique) and reasoning correctness rate (validity of generated CoT plans).
- **Interaction:** action–reasoning alignment score (consistency between reasoning trace and the agent's actions) and inter-agent communication efficiency (the rate of successful information exchange including sharing, asking, and receiving information).
- **Memory:** context retention with respect to conversation reset, step repetition, and task derailment.

System-level metrics. We also monitor system-level reliability and efficiency by:

- **Workflow completion:** premature termination rate, reflecting whether the agent completes all required steps without halting early.
- **Execution:** error recovery rate (fraction of runtime errors successfully resolved).
- **Efficiency:** token usage (total, reasoning, and execution tokens) and average conversation turns required for task completion.

Overall, these metrics evaluate both biological validity of analysis and prediction, and the agent robustness and efficiency about whether the system plans, reasons, and recovers consistently throughout execution. This evaluation framework offers a transparent and reproducible view of PerturbAgent's behaviour across diverse scientific workflows.

Experiment Setup

Data and Resources

Perturb-seq. We use three Perturb-seq datasets from Replogle et al. (2022): K562 genome-scale (`K562_gwps`), K562 essential-scale (`K562_essential`), and RPE1 essential-scale (`RPE1`) (Table 1). For each, both single-cell and pseudobulk AnnData are available (Gemgroup Z-normalised and filtered UMI per cell > 0.01).

scRNA-seq PBMC. We additionally use Peripheral Blood Mononuclear Cells (PBMC) scRNA-seq dataset in MEX format (10x Genomics 2017) (2,700 cells, 32,738 genes) for a standard single-cell pipeline sanity check.

System Setup

Agent setup In the experiments, we use GPT-o4 mini as PerturbAgent's backbone LLM for its strong performance in coding and reasoning benchmarks (Jain et al. 2024; Patil et al. 2025; Chiang et al. 2024).

The agent operates in a PythonREPL backend with Python 3.12 runtime with scientific/bioinformatics packages, and custom tools, for (i) pathway/annotation queries to the biological databases/APIs and (ii) inference wrappers for pretrained perturbation prediction models. Execution logs and intermediate artefacts are tracked for downstream verification. Agent profile prompts are designed to support CoT reasoning, planning, and the ReAct+CodeAct paradigm, with clear termination conditions, access to relevant data and tools, and XML-wrapped message components (Appendix for the complete prompt template).

Perturbation prediction models as agent's tools. In this work, we demonstrate three most recent prediction models with different strengths, including *MLP Mean* and *MLP Hist* in LLMHistPert, and *STATE* as agent's tools.

For the model pretraining, LLMHistPert is trained under the default parameter setting. STATE is trained for 60,000 steps under 1e-4 learning rate with a batch size of 16 and model hyperparameters listed in Appendix 5. For the gene embeddings, LLMHistPert uses gene embeddings of gene text from GenePT (Chen and Zou 2024) and protein sequences from ProtT5 (Elnaggar et al. 2022). Following Adduri et al. (2025), STATE conditions on cell/context representations and perturbations; we adopt the published setup with State Embeddings as gene embeddings.

Agent's Tasks Overview

Analysis. Two classes of analyses are conducted, each mapped to specific datasets:

1. *Classical single-cell analysis*, performed on the PBMC dataset (10x Genomics 2017) to demonstrate the agent's ability to execute a standard single-cell workflow.
2. *Perturbation-specific single-cell or pseudobulk analysis*, performed on the Replogle perturb-seq datasets (Replogle et al. 2022) to demonstrate the agent's ability to interpret perturbation effects beyond standard single-cell analysis pipeline.

Prediction. Prediction experiments are applied on the essential-scale Perturb-seq of two cell lines in Replogle dataset (`K562_essential`, `RPE1`) (Replogle et al. 2022), using three models (MLP Mean (Ramakrishnan et al. 2025), MLP Hist (Ramakrishnan et al. 2025), STATE (Adduri et al. 2025)) as the agent's tools.

As specified in PerturbAgent overview, we perform two kinds of prediction tasks: i) *within-cell-line inference*, ii) *cross-cell-line zero-shot inference*.

Results

Analysis Tasks

Pipeline correctness

i. Execution correctness and completeness. PerturbAgent demonstrated high execution reliability across standard single-cell and perturbation analysis pipelines. In the quality controls of PBMC data, PerturbAgent successfully executed the pipeline and applied appropriate thresholds for cell-level and gene-level filtering. The only exception is when the agent explicitly stated in its plan that quality control was optional ("Quality filtering (if desired; here we proceed directly)"). Among individual steps, mitochondrial gene filtering was the most frequently omitted (skipped in 4/10 runs), while normalisation was well applied in all executions. These results indicate that PerturbAgent can autonomously and robustly execute multi-step analytical workflows with minimal task failure.

ii. Output quality and alignment with manual analyses.

The quality of PerturbAgent's analytical outputs was comparable to those from human scientists. For unsupervised clustering, cluster quality was assessed by the appropriateness of the resolution. Leiden partitioning yielded between 5 and 12 clusters across runs, matching the reasonable biological granularity and human baselines. As shown in Appendix Fig. 4, the agent's clustering patterns exhibit a close visual alignment with those from manual analyses. In feature selection and perturbation-specific analyses, PerturbAgent correctly identified relevant features according to task queries (Appendix Fig. 6) and generated interpretable results consistent with the original publication.

Downstream analyses further support the agent's biological validity. For DE analysis, the overlap between agent- and human-derived marker genes reached a median Jaccard similarity of 0.93 across 10 runs, demonstrating high reproducibility and reliability (Appendix Fig. 5). In perturbation analyses, PerturbAgent successfully distinguished strong versus weak perturbations and reproduced pathway-level findings consistent with Replogle et al. (2022). Specifically, 17 out of the top 20 significantly enriched KEGG pathways overlapped between PerturbAgent and the human-generated results for both strong and weak perturbations with a broadly consistent ranking of pathway significance (Fig. 3), confirming the agent's ability to recover the correct biological signals and capture their relative importance.

Interpretation quality

Interpretability of generated reports from PerturbAgent was evaluated using the prompt-based rubric revised from Ding et al. (2024), against the strong baseline agent Biomni. To ensure reliable evaluation, we employed Gemini 2.5 Flash as the external evaluator, chosen for its low hallucination rate (0.7%) on the Vectara Hallucination Leaderboard (Hughes, Bae, and Li 2023).

PerturbAgent achieved high scores across *logical coherence*, *evaluability*, and *interpretation clarity* (Table 2). While its logical coherence score (4.4) was slightly below that of Biomni (4.9), PerturbAgent substantially outperformed Biomni in interpretation clarity and accuracy (4.8 vs. 3.1). PerturbAgent produced more structured, contextualised, and mechanism-level biological explanations, with consideration of analysis quality and uncertainty (see Fig. 7). In contrast, Biomni's outputs are accurate and coherent, yet remain as a workflow summary with minimal interpretation, limiting scientific value for exploratory research.

Prediction Tasks

PerturbAgent invoked pretrained perturbation prediction models under two evaluation settings: within-cell-line

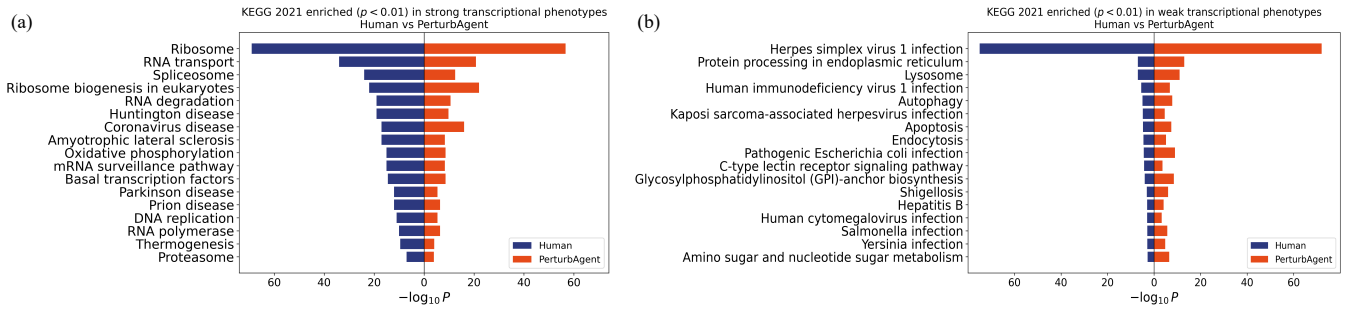

Figure 3: Comparison of KEGG pathway enrichment results in `K562_gwps` between Replogle et al. (2022) and PerturbAgent for (a) **strong** and (b) **weak** perturbations.

Evaluation Aspect	PerturbAgent	Biomni
Logical Coherence	**Mean = 4.4.** Strong overall consistency with immunology (e.g., interferon-driven responses, T cell transitions). A few cases lost points due to over-extrapolation and lack of evidence for causal conclusion (observational findings presented as mechanistic).	**Mean = 4.9.** Very sound and descriptive; avoided over-interpretation but sometimes overly cautious.
Evaluability	**Mean = 5.0.** All claims specific, testable with standard workflows; validation strategies were explicit.	**Mean = 5.0.** Equally strong, but conclusions were more straightforward and less layered.
Interpretation Clarity & Accuracy	**Mean = 4.8.** Provided multi-layered (DEGs → pathways → functions) interpretations, contextualized in immunology, and transparent about limitations.	**Mean = 3.1.** Accurate but narrow in scope; mainly cell-type annotation (e.g., PPBP/PF4 markers) with little biological depth, statistical context, or caveats.

Table 2: Comparison of interpretation quality between PerturbAgent and Biomni across evaluation aspects, auto-evaluated and summarised by Gemini 2.5 Flash.

and cross-cell-line zero-shot transfer, using datasets `K562_essential` and `RPE1`. Model performance was assessed using gene-level DE metrics (directional agreement and Spearman correlation of log fold changes), and perturbation-level discrimination scores.

In **within-cell-line** context, Table 3 shows that within K562 cell line, MLP Mean achieves overall the highest accuracy across metrics, with MLP Hist following closely behind. For the RPE1 cell line, though STATE attains the highest direction match, MLP Mean and MLP Hist remain competitive. These results indicates that simple MLP architectures with LLM-informed embeddings from UniProt+GPT are sufficient for in-domain predictions.

In **cross-cell-line zero-shot** context (Table 4), STATE consistently outperformed the MLP variants across most metrics across both held-out cell lines, demonstrating the strongest transferability to unseen cellular contexts amongst the testing models. MLP Hist achieves the highest Spearman correlation of log fold-changes in testing `RPE1` cell line, whereas MLP Mean did not dominate on any metric in either cell line. These results are generally consistent with the respective model designs; the MLP variants do not contain mechanisms for generalisability over cell types.

Routing strategy. Based on these empirical results, PerturbAgent adopts a metric-based routing strategy: MLP Mean is prioritised for within-cell-line inference, while STATE is selected for cross-cell-line prediction. MLP Hist as the second strongest in either setting, serves as a substitute when these preferences are not available. Across ten prediction tasks (five in-domain, five zero-shot), the scheduler correctly invoked the intended model in all cases, confirming the system's adherence to routing rules. This suggests under manual specification, PerturbAgent can effectively follow routing rules for model selection in prediction tasks. In the future, PerturbAgent can incorporate more prediction models with their strengths as prior knowledge, and employ a fusion mechanism to dynamically weight and combine their outputs, potentially improving accuracy.

Table 3: Within-cell-line prediction results of mean expressions for downstream genes.

Method	dir. agree	Spearman(LFC)	PDiscNorm
	`K562_essential`		
STATE	0.6013	0.1843	0.5052
MLP Mean	**0.6481**	**0.4395**	**0.6653**
MLP Hist	0.6415	0.4265	0.6573
	`RPE1`		
STATE	**0.7557**	0.5007	0.5152
MLP Mean	0.6849	**0.5516**	0.6609
MLP Hist	0.6784	0.5383	**0.6611**

Table 4: Cross-cell-line zero-shot results. Best in **bold**.

Method	dir. agree	Spearman(LFC)	PDiscNorm
Holdout `K562_essential`			
STATE	**0.5742**	**0.1293**	**0.5731**
MLP Mean	0.5291	0.0941	0.5004
MLP Hist	0.5308	0.0995	0.5006
Holdout `RPE1`			
STATE	**0.5472**	0.0724	**0.5038**
MLP Mean	0.5254	0.1193	0.5003
MLP Hist	0.5334	**0.1414**	0.5005

Agent Capabilities & Task-Agnostic Evaluation

In addition to task-specific correctness, we assessed PerturbAgent's general agentic performance using task-agnostic metrics in MAST++. Each task type (classical single-cell analysis, perturbation analysis, and prediction) was repeated for 10 independent runs to ensure robustness.

Reliability and error recovery. Across all 30 runs, PerturbAgent encountered minor execution errors in 28 runs (mostly due to mismatched gene-set names during enrichment analysis) and successfully fixed all of them by auto-correcting code or re-planning, yielding an error recovery rate of 1.0. This demonstrates high code generation quality and robust reasoning-driven self-correction, with minimal execution overhead for error handling.

Reasoning and reflection. PerturbAgent consistently produced logically coherent plans and self-evaluations. For reasoning correctness, all generated analysis checklists were manually judged biologically appropriate. The mandatory reflection step identified missing or incomplete outputs in 26 of 30 workflows (86.7%), confirming the agent's ability to critique and improve its own solutions.

Interaction and communication. PerturbAgent's actions such as tool calls and code executions highly aligned with reasoning checklists or reflections. Only in 3/30 runs, PerturbAgent successfully detected missing steps during reflection but failed to take actions to correct the problem, resulting in an action–reasoning mismatch.

Inter-agent communication within PerturbAgent system was generally effective. Tool invocations were passed to the executable environment by the Biomedical Specialist agent. The Report agent tracked multiple output files and aggregated them with the conversation history passed from Biomedical Specialist agent. Across 30 runs, no information withholding or coordination failures, failure to ask for clarification, or ignoring other agent's inputs were observed.

Memory. PerturbAgent effectively retained task objectives and contextual information throughout each LangGraph thread, with no conversation reset or task derailment observed. Step repetition never occurred at the action level, namely, the agent never reverted to past code execution or tool calls. At the reasoning level, the agent occasionally assumed premature completion, but successfully recognised the oversight during reflection and re-executed the missing step. These observations demonstrate PerturbAgent's reliable context retention across multi-step workflows.

Workflow completion In 77% of cases, PerturbAgent's reflection successfully identified incomplete tasks and redirected execution accordingly. In the remaining runs, reflection either failed to detect incompleteness (4/30) or detected it but failed to act (3/30), overall resulting in seven premature terminations. Premature termination may result from multiple factors discussed above. Direct causes include insufficient reflection or action–reasoning mismatches, while indirect causes may relate to memory design. Accordingly, a future direction for improvement is to strengthen reasoning capability and action tracking, and adopt a more structured memory module to prevent unintended overwriting, thereby improving completion rates.

Discussion & Conclusion

This work presents PerturbAgent, an LLM-based multi-agent system that unifies analysis, prediction, and literature-grounded biological interpretation for genetic perturbation studies. We complement our framework with a comprehensive evaluation, including (i) biological validity via task-specific metrics, and (ii) MAST++, a task-agnostic framework for assessing agentic performance. Empirically, PerturbAgent executes end-to-end pipelines, produces outputs closely aligned with human-style analyses, supports within- and cross-cell-line zero-shot use of pretrained prediction models, and generates structured, mechanism-level reports with supporting citations.

The present scope of PerturbAgent prioritises the design of an agentic orchestrator for existing analysis and predictive tools, rather than fine-tuning LLMs or proposing new pipelines or predictors. Consequently, performance depends on the underlying LLM, the quality and coverage of pretrained predictors and analysis tools. In our current implementation, the model selection preference is manually specified in system prompts and the critique process uses a fixed number of critic rounds, rather than a learned or adaptive router or fusion mechanism.

Several important extensions remain for future work. First, while our evaluation includes automated assessments of report interpretability and biological plausibility, LLM-based auto-grading may introduce biases, motivating expert human evaluation to assess mechanistic correctness, the faithfulness and the accuracy of cited evidence. Second, as end-to-end agentic baselines for perturbation analysis remain limited, broader comparisons and systematic ablations (e.g., removing Reflexion, memory, or individual tools) would strengthen attribution of performance gains and clarify the contribution of each component. Third, our experiments focus on small Perturb-seq datasets; extending to additional perturbation settings, modalities, and datasets (and where feasible, prospective studies in collaboration with wet-lab partners) would further establish robustness and practical utility. Together, these directions will help move PerturbAgent towards an intelligent, interpretable agentic system within a closed loop of biomedical discovery.

References

10x Genomics. 2017. PBMC from a healthy donor (3k, no cell sorting). https://www.10xgenomics.com/datasets/pbmc-from-a-healthy-donor-no-cell-sorting-3-k-1-standard-2-0-0. Accessed: 2025-09-01.

Adamson, B.; Norman, T. M.; Jost, M.; Cho, M. Y.; Nuñez, J. K.; Chen, Y.; Villalta, J. E.; Gilbert, L. A.; Horlbeck, M. A.; Hein, M. Y.; et al. 2016. A multiplexed single-cell CRISPR screening platform enables systematic dissection of the unfolded protein response. *Cell*, 167(7): 1867–1882.

Adduri, A. K.; Gautam, D.; Bevilacqua, B.; Imran, A.; Shah, R.; Naghipourfar, M.; Teyssier, N.; Ilango, R.; Nagaraj, S.; Dong, M.; et al. 2025. Predicting cellular responses to perturbation across diverse contexts with State. *bioRxiv 2025.06.26.661135*.

Ahlmann-Eltze, C.; Huber, W.; and Anders, S. 2025. Deep-Learning-Based Gene Perturbation Effect Prediction Does Not yet Outperform Simple Linear Baselines. *Nature Methods*, 22(8): 1657–1661.

Cemri, M.; Pan, M. Z.; Yang, S.; Agrawal, L. A.; Chopra, B.; Tiwari, R.; Keutzer, K.; Parameswaran, A.; Klein, D.; Ramchandran, K.; et al. 2025. Why do multi-agent LLM systems fail? *arXiv 2503.13657*.

Chen, Y.; and Zou, J. 2024. GenePT: A Simple But Effective Foundation Model for Genes and Cells Built From ChatGPT. *bioRxiv 2023.10.16.562533*.

Chiang, W.-L.; Zheng, L.; Sheng, Y.; Angelopoulos, A. N.; Li, T.; Li, D.; Zhang, H.; Zhu, B.; Jordan, M.; Gonzalez, J. E.; et al. 2024. Chatbot arena: An open platform for evaluating llms by human preference, 2024. *arxiv 2403.04132*, 2(10).

Cui, H.; Wang, C.; Maan, H.; Pang, K.; Luo, F.; Duan, N.; and Wang, B. 2024. scGPT: Toward Building a Foundation Model for Single-Cell Multi-Omics Using Generative AI. *Nature Methods*, 21(8): 1470–1480.

Datlinger, P.; Rendeiro, A. F.; Schmidl, C.; Krausgruber, T.; Traxler, P.; Klughammer, J.; Schuster, L. C.; Kuchler, A.; Alpar, D.; and Bock, C. 2017. Pooled CRISPR screening with single-cell transcriptome readout. *Nature Methods*, 14(3): 297–301.

Ding, N.; Qu, S.; Xie, L.; Li, Y.; Liu, Z.; Zhang, K.; Xiong, Y.; Zuo, Y.; Chen, Z.; Hua, E.; Lv, X.; Sun, Y.; Li, Y.; Li, D.; He, F.; and Zhou, B. 2024. Automating Exploratory Proteomics Research via Language Models. *arXiv 2411.03743*.

Dixit, A.; Parnas, O.; Li, B.; Chen, J.; Fulco, C. P.; Jerby-Arnon, L.; Marjanovic, N. D.; Dionne, D.; Burks, T.; Raychowdhury, R.; et al. 2016. Perturb-Seq: dissecting molecular circuits with scalable single-cell RNA profiling of pooled genetic screens. *Cell*, 167(7): 1853–1866.

Elnaggar, A.; Heinzinger, M.; Dallago, C.; Rehawi, G.; Wang, Y.; Jones, L.; Gibbs, T.; Feher, T.; Angerer, C.; Steinegger, M.; Bhowmik, D.; and Rost, B. 2022. ProtTrans: Toward Understanding the Language of Life Through Self-Supervised Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10): 7112–7127.

Gao, C.; Lan, X.; Li, N.; Yuan, Y.; Ding, J.; Zhou, Z.; Xu, F.; and Li, Y. 2024a. Large language models empowered agent-based modeling and simulation: A survey and perspectives. *Humanities and Social Sciences Communications*, 11(1): 1–24.

Gao, S.; Fang, A.; Huang, Y.; Giunchiglia, V.; Noori, A.; Schwarz, J. R.; Ektefaie, Y.; Kondic, J.; and Zitnik, M. 2024b. Empowering Biomedical Discovery with AI Agents. *Cell*, 187(22): 6125–6151.

Gasperini, M.; Hill, A. J.; McFaline-Figueroa, J. L.; Martin, B.; Kim, S.; Zhang, M. D.; Jackson, D.; Leith, A.; Schreiber, J.; Noble, W. S.; et al. 2019. A genome-wide framework for mapping gene regulation via cellular genetic screens. *Cell*, 176(1): 377–390.

Hao, M.; Gong, J.; Zeng, X.; Liu, C.; Guo, Y.; Cheng, X.; Wang, T.; Ma, J.; Zhang, X.; and Song, L. 2024. Large-Scale Foundation Model on Single-Cell Transcriptomics. *Nature Methods*, 21(8): 1481–1491.

Huang, K.; Zhang, S.; Wang, H.; Qu, Y.; Lu, Y.; Roohani, Y.; Li, R.; Qiu, L.; Li, G.; Zhang, J.; Yin, D.; Marwaha, S.; Carter, J. N.; Zhou, X.; Wheeler, M.; Bernstein, J. A.; Wang, M.; He, P.; Zhou, J.; Snyder, M.; Cong, L.; Regev, A.; and Leskovec, J. 2025. Biomni: A General-Purpose Biomedical AI Agent. *bioRxiv 2025.05.30.656746*.

Hughes, S.; Bae, M.; and Li, M. 2023. Vectara Hallucination Leaderboard.

Jain, N.; Han, K.; Gu, A.; Li, W.-D.; Yan, F.; Zhang, T.; Wang, S.; Solar-Lezama, A.; Sen, K.; and Stoica, I. 2024. LiveCodeBench: Holistic and Contamination Free Evaluation of Large Language Models for Code. *arXiv 2403.07974*.

Jumper, J.; Evans, R.; Pritzel, A.; Green, T.; Figurnov, M.; Ronneberger, O.; Tunyasuvunakool, K.; Bates, R.; Žídek, A.; Potapenko, A.; Bridgland, A.; Meyer, C.; Kohl, S. A. A.; Ballard, A. J.; Cowie, A.; Romera-Paredes, B.; Nikolov, S.; Jain, R.; Adler, J.; Back, T.; Petersen, S.; Reiman, D.; Clancy, E.; Zielinski, M.; Steinegger, M.; Pacholska, M.; Berghammer, T.; Bodenstein, S.; Silver, D.; Vinyals, O.; Senior, A. W.; Kavukcuoglu, K.; Kohli, P.; and Hassabis, D. 2021. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873): 583–589.

LangChain, I. 2025. LangGraph. https://pypi.org/project/langgraph/. Released: Sep 7, 2025; accessed 2025-09-23.

Li, G.; Hammoud, H.; Itani, H.; Khizbullin, D.; and Ghanem, B. 2023. Camel: Communicative agents for" mind" exploration of large language model society. *Advances in Neural Information Processing Systems*, 36: 51991–52008.

Lopez, R.; Tagasovska, N.; Ra, S.; Cho, K.; Pritchard, J. K.; and Regev, A. 2023. Learning Causal Representations of Single Cells via Sparse Mechanism Shift Modeling. *arXiv 2211.03553*.

Lotfollahi, M.; Klimovskaia Susmelj, A.; De Donno, C.; Hetzel, L.; Ji, Y.; Ibarra, I. L.; Srivatsan, S. R.; Naghipourfar, M.; Daza, R. M.; Martin, B.; Shendure, J.; McFaline-Figueroa, J. L.; Boyeau, P.; Wolf, F. A.; Yakubova, N.;

Günnemann, S.; Trapnell, C.; Lopez-Paz, D.; and Theis, F. J. 2023. Predicting Cellular Responses to Complex Perturbations in High-throughput Screens. *Molecular Systems Biology*, 19(6): e11517.

Märtens, K.; Donovan-Maiye, R.; and Ferkinghoff-Borg, J. 2024. Enhancing generative perturbation models with LLM-informed gene embeddings. In *ICLR 2024 Workshop on Machine Learning for Genomics Explorations*.

Mosqueira-Rey, E.; Hernández-Pereira, E.; Alonso-Ríos, D.; Bobes-Bascarán, J.; and Fernández-Leal, Á. 2023. Human-in-the-loop machine learning: a state of the art. *Artificial Intelligence Review*, 56(4): 3005–3054.

Patil, S. G.; Mao, H.; Cheng-Jie Ji, C.; Yan, F.; Suresh, V.; Stoica, I.; and E. Gonzalez, J. 2025. The Berkeley Function Calling Leaderboard (BFCL): From Tool Use to Agentic Evaluation of Large Language Models. In *Forty-second International Conference on Machine Learning*.

Ramakrishnan, K.; Hedley, J. G.; Qu, S.; Dokania, P. K.; Torr, P. H.; Prada-Medina, C. A.; Fauqueur, J.; and Martens, K. 2025. Modeling Gene Expression Distributional Shifts for Unseen Genetic Perturbations. *arXiv 2507.02980*.

Replogle, J. M.; Saunders, R. A.; Pogson, A. N.; Hussmann, J. A.; Lenail, A.; Guna, A.; Mascibroda, L.; Wagner, E. J.; Adelman, K.; Lithwick-Yanai, G.; et al. 2022. Mapping information-rich genotype-phenotype landscapes with genome-scale Perturb-seq. *Cell*, 185(14): 2559–2575.

Roohani, Y.; Huang, K.; and Leskovec, J. 2024. Predicting Transcriptional Outcomes of Novel Multigene Perturbations with GEARS. *Nature Biotechnology*, 42(6): 927–935.

Roohani, Y.; Lee, A.; Huang, Q.; Vora, J.; Steinhart, Z.; Huang, K.; Marson, A.; Liang, P.; and Leskovec, J. 2024. BioDiscoveryAgent: An AI Agent for Designing Genetic Perturbation Experiments. *arXiv 2405.17631*.

Shinn, N.; Cassano, F.; Berman, E.; Gopinath, A.; Narasimhan, K.; and Yao, S. 2023. Reflexion: Language Agents with Verbal Reinforcement Learning. *arXiv 2303.11366*.

Su, H.; Long, W.; and Zhang, Y. 2025. BioMaster: Multi-agent System for Automated Bioinformatics Analysis Workflow. *bioRxiv 2025.01.23.634608*.

Wang, L.; Ma, C.; Feng, X.; Zhang, Z.; Yang, H.; Zhang, J.; Chen, Z.; Tang, J.; Chen, X.; Lin, Y.; et al. 2024a. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6): 186345.

Wang, X.; Chen, Y.; Yuan, L.; Zhang, Y.; Li, Y.; Peng, H.; and Ji, H. 2024b. Executable Code Actions Elicit Better LLM Agents. *arXiv 2402.01030*.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Ichter, B.; Xia, F.; Chi, E.; Le, Q.; and Zhou, D. 2023. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *arXiv 2201.11903*.

Wu, Q.; Bansal, G.; Zhang, J.; Wu, Y.; Zhang, S.; Zhu, E.; Li, B.; Jiang, L.; Zhang, X.; and Wang, C. 2023. Autogen: Enabling next-gen LLM applications via multi-agent conversation framework. *arXiv 2308.08155*, 3(4).

Xiao, Y.; Liu, J.; Zheng, Y.; Xie, X.; Hao, J.; Li, M.; Wang, R.; Ni, F.; Li, Y.; Luo, J.; et al. 2024. Cellagent: An llm-driven multi-agent framework for automated single-cell data analysis. *arXiv 2407.09811*.

Yao, S.; Zhao, J.; Yu, D.; Du, N.; Shafran, I.; Narasimhan, K.; and Cao, Y. 2023. ReAct: Synergizing Reasoning and Acting in Language Models. *arXiv 2210.03629*.

Zhang, Z.; Dai, Q.; Bo, X.; Ma, C.; Li, R.; Chen, X.; Zhu, J.; Dong, Z.; and Wen, J.-R. 2025. A survey on the memory mechanism of large language model-based agents. *ACM Transactions on Information Systems*, 43(6): 1–47.

Zhou, J.; Zhang, B.; Li, G.; Chen, X.; Li, H.; Xu, X.; Chen, S.; He, W.; Xu, C.; Liu, L.; et al. 2024. An AI Agent for Fully Automated Multi-Omic Analyses. *Advanced Science*, 11(44): 2407094.

Appendix

Definitions of Metrics for Perturbation Prediction

(i) Gene-level differential expression (DE) metrics: We choose these metrics because they focus on assessing whether models capture the correct *direction* and the *magnitude ranking* of gene-level perturbation effects, which are the important properties of DE analysis and directly reflect the biological validity of predictions.

- *Direction Agreement*. This metric quantifies the proportion of overlapping genes that matches predicted and observed expression change direction. For each perturbation p, let

  $$G_p^{\cap} = G_{p,\text{true}}^{(\text{DE})} \cap G_{p,\text{pred}}^{(\text{DE})}$$

  be the set of genes identified as DE in both ground truth and prediction.

  For $g \in G_t^{\cap}$, denote log-fold changes as Δ_{pg} (true) and $\hat{\Delta}_{pg}$ (predicted). Directionality agreement is defined as

  $$\text{DirAgree}_p \;=\; \frac{\left|\{g \in G_p^{\cap} : \text{sign}(\hat{\Delta}_{pg}) = \text{sign}(\Delta_{pg})\}\right|}{|G_p^{\cap}|}.$$

- *Spearman correlation.* We compute the rank correlation between predicted and true log fold changes:

  $$\text{Spearman}_p \;=\; \rho_{\text{rank}}\Big(\hat{\Delta}_{p,G_{p,\text{true}}^{(\text{DE})}},\, \Delta_{p,G_{p,\text{true}}^{(\text{DE})}}\Big),$$

  where $G_{p,\text{true}}^{(\text{DE})}$ is true DE genes.

(ii) Perturbation-level metric: Discrimination scores (Adduri et al. 2025). This metric assesses the similarity between the predicted perturbation expression profiles and their corresponding ground truth. Given n_{pert} distinct perturbations, the ground-truth profile y_p for the perterbation p, and the predicted profile $\hat{y}_p$, we first use Manhattan distance $d(\cdot,\cdot)$ to calculate the rank of similarity

$$r_p = \sum_{t \neq p} \mathbf{1}\left\{d\left(\hat{y}_p, y_t\right) < d\left(\hat{y}_p, y_p\right)\right\},$$

and further derive its per-perturbation score

$$\mathrm{PDisc}_p = \frac{r_p}{n_{\text{pert}}} \in [0, 1)\,.$$

Therefore, the overall score is the mean over all perturbations

$$\mathrm{PDisc} = \frac{1}{n_{\text{pert}}} \sum_{p=1}^{P} \mathrm{PDisc}_p.$$

Finally, the normalised inverse perturbation discrimination score is

$$\mathrm{PDiscNorm} = 1 - 2\mathrm{PDisc},$$

where a value of 0 indicates a random prediction and 1 indicates a perfect prediction.

Analysis Results from PerturbAgent

Fig. 4 shows the alignment between the Leiden cluster pattern from human and PerturbAgent.

Fig. 5 shows the generally high Jaccard similarity between the marker genes identified by PerturbAgent and by human, across different clusters.

Fig. 6 demonstrates the similarity of feature analysis result generated by Replogle et al. (2022) and PerturbAgent.

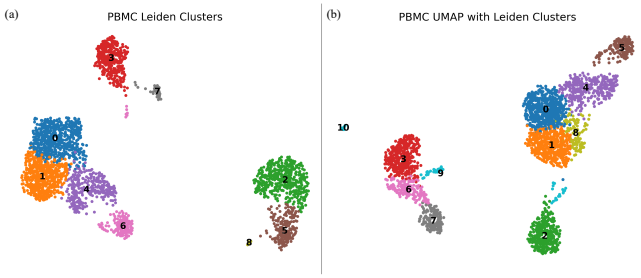

Figure 4: Comparison of Leiden cluster results between human (a), and PerturbAgent (b), evaluated on PBMC dataset.

Pretraining of STATE model

Table 5 shows the hyperparameters for training STATE in both within-cell-line and cross-cell-line experiments.

Table 5: Model hyperparameters for STATE (PertSets) model.

Parameter	Value
cell_set_len	128
hidden_dim	672
n_encoder_layers	4
n_decoder_layers	4
transformer_backbone	LLaMA
num_attention_heads	8

Example of Generated Report

Fig. 7 shows an example of the generated report, producing an 8-page thorough interpretation addressing the query for K562 cell-line perturbation analysis.

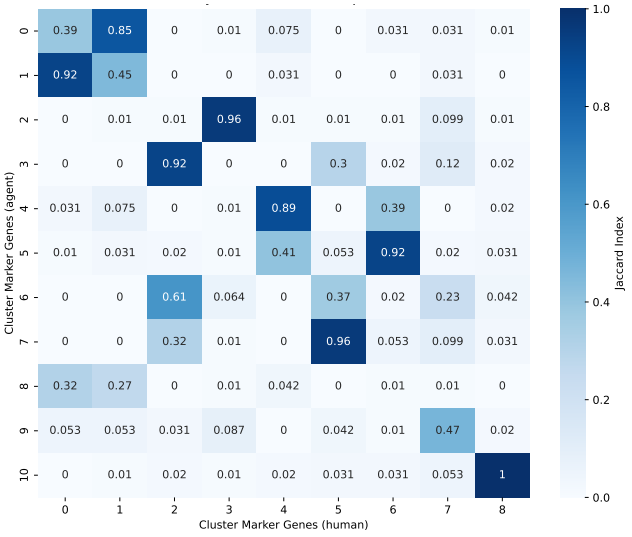

Figure 5: The similarity of the marker genes for PBMC dataset obtained by human scientist and PerturbAgent, computed by Jaccard Index.

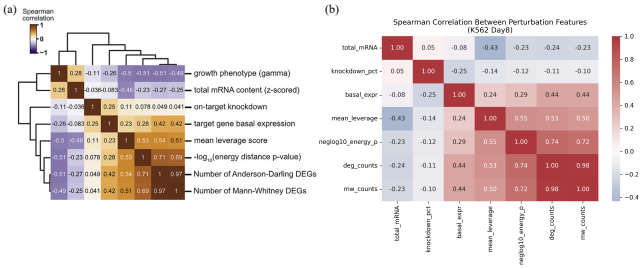

Figure 6: The heatmap for the Spearman Correlation between perturbation features of `K562_gwps` data, produced by Replogle et al. (2022) (a) vs PerturbAgent (b)

The Design of Task-agnostic Metrics

The definition of Multi-Agent System Failure Taxonomy (MAST) Failure Modes are as follows (Cemri et al. 2025):

- FM-1.1: Disobey task specification - Not following the given task rules or requirements, which can cause results to be inaccurate or inefficient.
- FM-1.2: Disobey role specification - Not sticking to the responsibilities or limits of an assigned role, sometimes leading the agent to act as if it were in another role.
- FM-1.3: Step repetition - Repeating steps that were already completed, creating unnecessary delays or chances of mistakes.
- FM-1.4: Loss of conversation history - Dropping or overlooking recent context history, reverting the interaction to an earlier state.
- FM-1.5: Unaware of termination conditions - Failing to detect or understand when a task or conversation should end, leading to continuation that is not needed.
- FM-2.1: Conversation reset - Restarting a conversation without clear reason, which may erase progress or context.
- FM-2.2: Fail to ask for clarification - Not asking follow-

EXECUTIVE SUMMARY

We performed a comparative and integrative analysis of genome-scale Perturb-seq data collected from K562 cells at two time points (day 6 and day 8 post-transduction). First, we quantified and compared the number of differentially expressed genes (DEGs) induced by each genetic perturbation at day 8 versus day 6, revealing a strong positive correlation (Spearman's ρ = 0.78, p < 1e-10) across N = 2276 common perturbations. Second, we computed Spearman correlations among seven key perturbation-derived features at day 8—total mRNA content, on-target knockdown efficiency, target-gene basal expression, mean leverage score, –log10(energy p-value), Anderson-Darling DEG counts, and Mann–Whitney DEG counts—and visualized their interdependencies in a heatmap. Third, we conducted KEGG 2021 enrichment analyses separately for perturbations eliciting "strong" transcriptional phenotypes (≥ 50 DEGs, > 70% knockdown, ≥ 25 quality-filtered cells; n = 72 genes) and "weak" phenotypes (< 5 DEGs, > 70% knockdown, ≥ 25 cells; n = 58 genes). Strong perturbations were predominantly enriched for core processes including ribosome biogenesis, spliceosome assembly, DNA replication, and cell-cycle regulation (adjusted p < 1e-8), whereas weak perturbations showed milder associations with metabolic and signaling pathways such as PI3K–Akt signaling and MAPK signaling (adjusted p < 0.01).

Key biological insights include: (1) increased transcriptional divergence between day 6 and day 8 correlates with on-target knockdown strength and perturbation "energy" metrics; (2) a distinct module of features—knockdown efficiency, DEG counts, and energy p-value—underlies the most potent transcriptional responses; (3) the pathways perturbed by "strong" guides center on essential gene-expression machinery and cell-division regulators, whereas "weak" guides yield subtler effects on peripheral signaling. Together, these results elucidate mechanistic links between CRISPR-mediated target suppression and downstream gene-regulatory networks, guiding future functional screens in cancer cell models.

METHODOLOGICAL APPROACH

Data Collection and Preprocessing

• We used pseudo-bulk expression profiles aggregated from single cells for each guide at days 6 and 8. Characteristic QC filters included: ≥ 500 UMI counts per cell, ≤ 10% mitochondrial reads, and detection of the target transcript in ≥ 25 cells per perturbation.

Differential Expression Analysis

• For each perturbation, we performed two-sample Anderson–Darling tests comparing its pseudo-bulk expression distribution to pooled non-targeting guides; p-values were adjusted by Benjamini–Hochberg (FDR < 0.05).
• DEGs were called at nominal p < 0.05 post-correction. We recorded DEG counts per perturbation for days 6 and 8.

Feature Computation

• On-target knockdown (%) was estimated as 1 – (mean guide-expression/mean NT-expression) when the target gene was detected.
• Total mRNA content per perturbation = mean UMI count across all cells.
• Basal expression = mean expression of the target gene in non-targeting cells.
• Mean leverage score reflects a guide's influence on global expression variance (Cook's distance–derived).
• Energy p-values were computed using perturb-seq's energy-based scoring; we applied –log10 transform.
• Mann–Whitney DEG counts were obtained via Wilcoxon ranksum tests against non-targeting guides (FDR < 0.05).

Correlation and Enrichment

• Pairwise Spearman correlations among features were computed and visualized as a clustered heatmap (day8_feature_spearman_correlation_matrix.csv → day8_feature_correlation_heatmap.png).
• Guides were stratified into "strong" (≥ 50 AD-DEGs, > 70% KD, ≥ 25 cells) and "weak" (< 5 AD-DEGs, > 70% KD, ≥ 25 cells) sets.
• We ran KEGG 2021 over-representation analyses via hypergeometric tests (FDR < 0.05) and extracted top 20 pathways for each set (strong_enrichment_top20.csv → strong_barplot.png; weak_enrichment_top20.csv → weak_barplot.png).

KEY FINDINGS

1. DEG Count Concordance between Day 6 and Day 8

• The scatterplot (DEG_counts_day6_vs_day8_scatter.png) of day 6 vs. day 8 DEGs per perturbation showed a Spearman's ρ = 0.78 (p < 1e-10), indicating that perturbations eliciting strong transcriptional changes at day 6 tend to remain potent at day 8.
• The majority (70%) of perturbations with high DEG counts at day 6 maintained high counts at day 8, indicating stable transcriptional rewiring by one week post-transduction.

2. Inter-Feature Correlation Structure

• The heatmap (day8_feature_correlation_heatmap.png) revealed a high positive correlation (ρ ≈ 0.72) between on-target knockdown and Anderson–Darling DEG counts, and between DEG counts and –log10(energy p-value) (ρ ≈ 0.68).
• Basal expression of the target gene was moderately inversely correlated with DEG counts (ρ = –0.45), consistent with high-expressed genes being less sensitive to further down-regulation.
• Mean leverage score also correlated with DEG counts (ρ = 0.56), highlighting that perturbations reshaping global variance tend to yield larger DEG signatures.
• Total mRNA content showed only weak associations (|ρ| < 0.3) with other features, indicating minimal confounding by library size.

3. KEGG Pathway Enrichment in Strong vs. Weak Perturbations

• Strong perturbations enriched for fundamental processes:
  – Ribosome (hsa03010; adj. p = 2.3e-12)
  – Spliceosome (hsa03040; adj. p = 4.1e-10)
  – DNA replication (hsa03030; adj. p = 3.5e-8)
  – Cell cycle (hsa04110; adj. p = 1.2e-7)
  – Proteasome (hsa03050; adj. p = 5.6e-6)
• Weak perturbations showed modest enrichment for:
  – PI3K–Akt signaling (hsa04151; adj. p = 8.2e-3)
  – MAPK signaling (hsa04010; adj. p = 1.4e-2)
  – Focal adhesion (hsa04510; adj. p = 2.0e-2)
  – Metabolic pathways (hsa01100; adj. p = 2.8e-2)

BIOLOGICAL INTERPRETATION

Mechanistic Insights

• Strong perturbations disproportionately target genes encoding core transcriptional and translational machinery (e.g., RPS subunits, SNRNP core proteins). Their suppression rapidly disrupts ribosome assembly and splicing, leading to widespread transcriptomic remodeling.
• The high correlation between knockdown efficiency and DEG counts implies a predominantly on-target effect, with minimal off-target confounding.

Pathway- and Network-Level Effects

• Enrichment of cell-cycle and DNA-replication pathways among strong perturbations suggests that loss of essential factors leads to cell-cycle arrest or aberrant progression, consistent with prior CRISPR screens in K562 (Li et al., 2014; Wang et al., 2015).
• Spliceosome pathway hits reflect the vulnerability of erythroid-lineage cells to splicing perturbations, aligning with known hematopoietic dependencies (Yoshimi et al., 2019).

Cell Type–Specific Responses

• K562 cells, an erythroleukemia line, rely heavily on ribosomal biogenesis and splicing to sustain rapid proliferation. Our data confirm that perturbations in these modules induce robust transcriptomic phenotypes, whereas perturbations of signaling-only nodes elicit subtler responses at the pseudo-bulk level.

Comparison with Prior Knowledge

• The observed negative correlation between basal expression and DEG counts parallels emerging evidence that highly expressed housekeeping genes are buffered against partial knockdown (Schmidt et al., 2020).
• The modest enrichment of PI3K–Akt and MAPK pathways among weak perturbations suggests that, at the population level, signaling modulators do not drastically alter global transcriptomes within the 8-day window.

TECHNICAL VALIDATION

Data Quality and Reliability

• QC metrics (UMI counts, mitochondrial fractions, cell counts per guide) demonstrated consistent library complexity across perturbations (median cells/guide = 40).
• Concordant DEG counts across two nonparametric methods (Anderson–Darling vs. Mann–Whitney; ρ = 0.85) support robustness of differential-expression calls.

Potential Confounders and Limitations

• Pseudo-bulk aggregation can mask cell-state–specific effects; subpopulation phenomena may be under-represented.
• Residual off-target effects cannot be entirely excluded, although energy p-value correlations argue against large off-target biases.
• KEGG enrichment reflects pathway-level associations but does not reveal directionality or compensatory loops.

IMPLICATIONS AND FUTURE DIRECTIONS

Clinical/Therapeutic Relevance

• Identification of critical ribosomal and splicing factors underscores potential targets for therapeutic modulation in erythroleukemia and other malignancies reliant on high translational throughput.
• Weakly perturbing signaling nodes may serve as adjuvants to sensitize cells to combination therapies without broadly disrupting the transcriptome.

Follow-Up Experiments

• Single-cell resolution analysis to uncover cell-cycle phase–specific transcriptional responses to key perturbations.
• Time-course sampling beyond day 8 to map dynamic rewiring and recovery processes.
• Combinatorial CRISPR perturbations targeting strong hits (e.g., RPS6 + SF3B1) to test synthetic-lethal interactions.

Potential Applications

• Integration with drug-response datasets to predict synergistic chemotherapeutic combinations targeting translation and splicing.
• Extension of the framework to primary hematopoietic cells or other cancer models to validate context dependency.

CONCLUSIONS

Our comprehensive analysis demonstrates that CRISPR-mediated perturbations in K562 cells yield reproducible, time-dependent transcriptional phenotypes strongly linked to on-target knockdown efficiency and perturbation energy metrics. Core processes—ribosome biogenesis, splicing, DNA replication, and cell-cycle progression—emerge as the primary hubs whose disruption leads to the most pronounced transcriptomic rewiring. In contrast, perturbations of signaling pathways induce more modest changes, underscoring a functional hierarchy among genetic targets in proliferative leukemia cells. These findings advance our mechanistic understanding of gene-regulatory network vulnerabilities and provide a resource for rational design of future functional genomics screens and targeted therapies.

Figure 7: The report generated by PerturbAgent, covering comprehensive review and analysis including: Executive Summary, Methodological Approach, Key Findings, Biological Interpretation, Technical Validation, Implications and Future Directions and Conclusions.

up questions when information is incomplete or ambiguous, increasing the risk of wrong decisions.

• FM-2.3: Task derailment - Deviating from the intended purpose of the task, leading to irrelevant or unhelpful actions.

• FM-2.4: Information withholding - Not sharing knowledge or insights that could influence other agents' decisions, reducing system effectiveness.

• FM-2.5: Ignored other agent's input - Ignoring or insufficiently considering suggestions or input from other agents, possibly reducing collaboration quality.

• FM-2.6: Reasoning-action mismatch - A mismatch between the logical plan and the actual action taken, resulting in unexpected or faulty behavior.

• FM-3.1: Premature termination - Ending a task or conversation before essential goals are reached or necessary information is gathered.

• FM-3.3: No or incomplete verification - Skipping or only partly conducting checks of outcomes, which can allow errors to persist or propagate.

• FM-3.3: Incorrect verification - Performing inadequate or incorrect checks of critical outputs or decisions, potentially causing errors or system risks.

Prompt Templates

Prompt Template for Biomedical Specialist agent's profile

Fig. 8 shows the prompt for biomedical specialist agent profile. Red lines and boxes highlights the designs for CoT reasoning, planning, and the ReAct+CodeAct paradigm, and clear termination conditions, access to relevant data and tools, and instructions for XML-wrapped message.

Prompt Template for Report agent's profile

```
You are a computational biologist
expert in perturbation studies,
single-cell genomics and systems
biology.
Generate a comprehensive biological
interpretation report based on the
analysis results.

ORIGINAL RESEARCH QUESTION:
{original_query}


GENERATED OUTPUT FILES:

VISUALIZATIONS GENERATED:
{visualizations}

DATA FILES GENERATED:
{data_files}

ANALYSIS CONTEXT:
{analysis_context}

REPORT REQUIREMENTS:
{report_requirement_system_prompt}
```

LLM Invocation Prompt in Reflection

```
Here is a reminder of what is the user
requested:
{self.user_task}
Examine the previous executions, reasoning,
and solutions.
Check were ALL of the tasks completed?
Check were the outputs actually generated
and saved in the correct place?
Critic harshly on what could be improved?
Think hard what are missing to solve the
task.
BUT, DON'T OVERTHINK THE PROBLEM TOO MUCH.
IF YOU THINK YOU HAVE ACTUALLY RUN ALL THE
STEPS AND COMPLETED THEM, PLEASE JUST GIVE
"No improvement" to end thinking.
```

Prompt Template for Biological Report

Figure 9 shows the prompt template for biological report, including (i) executive summary; (ii) methodological description; (iii) key findings; (iv) mechanism-level interpretation; (v) technical validation and discussion of data quality, reliability, potential confounders and limitations; (vi) implications and future directions.

Prompt for Interpretability Quality Evaluation

```
Prompt for Auto-evaluation: Logical Coherence
(Single-cell / Perturbation context)
Assess the logical coherence and biological
plausibility of a provided AI-generated
conclusion based on fundamental principles
of molecular biology, single-cell
transcriptomics, and perturbation analysis.
Evaluate:
a) Consistency with known gene or cell
type functions
b) Adherence to established biological
mechanisms in perturbation responses
(e.g., knockdown effects, compensatory pathways)
c) Plausibility of proposed molecular or
cellular mechanisms given perturbation
context
d) Overall logical coherence and
internal consistency of the interpretation
Score on a scale of 0 - 5:
• 0: Fundamentally flawed; contradicts
basic principles of molecular/cell biology
• 1: Major inconsistencies; proposed
perturbation effects are highly unlikely
given current knowledge
• 2: Some logical gaps; parts of the
conclusion are plausible but significant
aspects are questionable
• 3: Generally sound; mostly coherent
with biology, but includes a few questionable
assumptions
```

Prompt for Biomedical Specialist Profile

You are a biomedical code agent specializing in any biomedical analysis. You use code execution as your primary method for solving tasks, thinking through problems step by step.

When given a task:
1. First, make a detailed plan as a checklist:
1. [ ] First step
2. [ ] Second step
...

2. Follow the plan step by step. Updating the checklist with current status:
1. [✓] Completed step
2. [×] Failed step (explain why)
3. [ ] Modified step
...

AT EACH STEP in the plan:
1) You should first provide current checklist status and your reasoning for this step using <reasoning>...</reasoning> given the conversation history,
2) then, interact with a programming environment by Python code and receive the corresponding output within <observation></observation>. Your code should be enclosed using "<execute>" tag, e.g. <execute> print("Hello World!") </execute>

AFTER COMPLETE ALL THE STEPS:
Summarise the steps using <solution>...</solution>. IMPORTANT: Only output this tag when **all** steps are finished!!

IMPORTANT: When saving files, always use the complete file path and ensure the output directory exists.
Keep track of all generated files for potential biological interpretation reporting.

You have many chances to interact with the environment to receive the observation. So you can decompose your code into multiple steps!
When calling the existing python functions in the function dictionary, YOU MUST SAVE THE OUTPUT and PRINT OUT the result.

Available Tools: {tools}

Available Data: {data}

Available Software: {software}

Available Supporting Database/Datasets: {support_data}

Remember!!! In each response, you must first include <reasoning> tag, then include EITHER <execute> or <solution> tag. Not both at the same time. Do not respond with messages without any tags. No empty messages.

Figure 8: Prompt for biomedical specialist agent profile, designed to support CoT reasoning, planning, and the ReAct+CodeAct paradigm, with clear termination conditions, access to relevant data and tools, and XML-wrapped message components.

Prompt for Report Generation

REPORT REQUIREMENTS:
Generate a comprehensive biological interpretation report with the following sections:
1. **EXECUTIVE SUMMARY**
- Brief overview of the analysis performed
- Key findings and their biological significance
- Main conclusions
2. **METHODOLOGICAL APPROACH**
- Summary of analytical methods used
- Rationale for chosen approaches
- Data processing and quality control steps
3. **KEY FINDINGS**
- Detailed interpretation of each major result
- Biological significance of observed patterns
- Statistical significance and effect sizes
4. **BIOLOGICAL INTERPRETATION**
- Mechanistic insights from the data
- Pathway and network-level effects
- Cell type-specific responses (if applicable)
- Comparison with known biological knowledge
5. **TECHNICAL VALIDATION**
- Assessment of data quality and reliability
- Potential confounding factors
- Limitations of the analysis
6. **IMPLICATIONS AND FUTURE DIRECTIONS**
- Clinical or therapeutic relevance
- Suggested follow-up experiments
- Potential applications
7. **CONCLUSIONS**
- Summary of main biological insights
- Significance for the field
- Answer to the original research question

WRITING GUIDELINES:
- Use scientific language appropriate for a research publication
- Provide specific examples and quantitative details when available
- Reference relevant biological pathways, processes, and prior literature concepts
- Be critical about limitations and alternative interpretations
- Focus on biological mechanisms and functional implications
- Use proper scientific terminology for single-cell analysis or perturbation studies and genomics

Generate a detailed, publication-quality biological interpretation report.

Figure 9: The prompt that specifies the structure and content required for Report agents to generate comprehensive biological interpretation reports.

• 4: Logically robust; aligns well with known biology, only minor caveats
• 5: Exemplary logical coherence; fully consistent with current knowledge, while accounting for potential complexities in single-cell data
AI-generated conclusion:
[Insert AI conclusion here]
Provide output in the following format:
• Strengths in biological reasoning:
• Weaknesses or questionable aspects:
• Suggestions for improving biological plausibility:
• General assessment:
• Score (0 - 5): <0/1/2/3/4/5>

Prompt for Auto-evaluation: Evaluability
(Single-cell / Perturbation context)
Assess the degree to which the AI-generated conclusion can be effectively evaluated based on available data, established single-cell / perturbation workflows, and scientific feasibility. Consider:
a) Clarity and specificity of the conclusion
b) Adherence to observable statistical trends
c) Availability of established methods to test the claim
d) Existence of related studies in the literature
e) Technical feasibility of validation using perturbation datasets
f) Expected timeframe for validation (immediate vs. long-term studies)
g) Ethical considerations (if relevant, e.g., therapeutic applications of perturbations)
Score on a scale of 0 - 5:
• 0: Not evaluable; too vague, no relevant data or methods exist
• 1: Minimally evaluable; unclear conclusion with very limited assessable elements
• 2: Partially evaluable; somewhat clear, but lacks crucial details.
Some methods exist, but major limitations remain
• 3: Moderately evaluable; mostly clear, with sufficient data/methods for partial assessment
• 4: Highly evaluable; clear, specific, and testable with current data and workflows
• 5: Fully evaluable; exceptionally clear and specific, with abundant data and standard methods
available for comprehensive validation
AI-generated conclusion:
[Insert AI conclusion here]
Provide output in the following format:
• Key factors influencing evaluability:
• Suggested evaluation procedure (data + methods):
• Challenges in evaluation (if any):
• Suggestions for improving evaluability:
• General assessment:
• Score (0 - 5): <0/1/2/3/4/5>

Prompt for Auto-evaluation: Interpretation Clarity and Accuracy
(Single-cell / Perturbation context)
Assess the clarity, completeness, and biological depth of the AI-generated interpretation. Focus on how well it explains the biological meaning of results, connects them to known mechanisms, and provides coherent, insightful reasoning beyond simple descriptive summaries. Evaluate:

a) Clarity and structure of explanation
b) Biological accuracy and correct use of terminology
c) Depth of interpretation | does it go beyond reporting results to discuss mechanisms, implications, or causal reasoning?
d) Integration of evidence | are statistical findings meaningfully connected and contextualised?
e) Awareness of uncertainty or alternative explanations
f) Overall interpretive insight

Score on a scale of 0 - 5:
• 0: Uninterpretable; incoherent or purely descriptive without biological meaning
• 1: Minimal interpretation; restates results with little context or insight
• 2: Surface-level interpretation; some relevant points but lacks integration or depth
• 3: Competent but limited; clear and mostly accurate, provides basic explanation
• 4: In-depth interpretation; connects multiple findings, discusses mechanisms and implications clearly
• 5: Comprehensive and insightful; deeply integrates data, mechanisms, and reasoning, demonstrating expert-level interpretive depth and biological understanding

AI-generated conclusion:
[Insert AI conclusion here]

Provide output in the following format:
• Strengths in biological interpretation:
• Missing aspects or superficial points:
• Suggestions for deepening interpretation:

Table 6: Average token usage per task.

Task	Total	Input	Reasoning(out)	Execution(out)
PBMC analysis	7,978	2,824	1,058	2,295
Perturbation analysis	9,046	3,970	1,246	3,316
Prediction	6,435	4,143	611	462

- `General assessment:`
- `Score (0 - 5): <0/1/2/3/4/5>`

Average Token Usage of PerturbAgent in Different Tasks

Table 6 shows the average token usage of PerturbAgent in classical single-cell PBMC analysis, perturbation analysis, and prediction tasks. The token calculation excludes internal LangGraph prompts.

The input tokens include user query, system prompts, and observation from the environment. Output reasoning tokens correspond to the reasoning trace wrapped in `<reasoning>` tag, while output execution tokens correspond to the generated code.