AN AI AGENTIC FRAMEWORK FOR AUTOMATED RECONSTRUCTION AND ANALYSIS OF GENOME-SCALE METABOLIC MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Genome-scale metabolic modeling serves as a critical interface between genomic information and hypotheses regarding cellular functionality; however, its broader adoption is impeded by substantial programming demands and fragmented software ecosystems. In response, we present GemAgents, a language-agent framework engineered to automate the processes of metabolic modeling and analysis. This system allows biologists to express research objectives through natural language, thereby eliminating the necessity for manual coding of analytical workflows. GemAgents distinctly delineates the interpretation of language-based tasks and the orchestration of computational tools from the deterministic aspects of scientific computation, grounding model construction and evaluation in explicit input data, reaction evidence, and clearly defined modeling assumptions. The reconstruction pipeline includes sequence annotation, mapping of evidence to reactions, formulation of gene–protein–reaction associations, medium-constrained gap filling, and model exportation. An independent verification module evaluates mass and charge balance, identifies erroneous metabolite and energy generation, and employs a counterexample-guided repair strategy to suggest corrections that preserve specified metabolic functions. Furthermore, an integrated auto-analysis extension facilitates direct model interrogation, the design of non-native metabolic pathways, and the in silico integration and assessment of heterologous pathways. Engineering case studies utilizing publicly available genomes and reference models illustrate the robustness and versatility of these components. By combining natural language interaction with transparent, model-based computational processes, GemAgents offers a comprehensive solution that removes programming obstacles and effectively translates biological questions into experimentally testable computational hypotheses.

1 INTRODUCTION

Genome-scale metabolic models (GEMs) represent cellular metabolism as a structured relationship between genes, biochemical reactions, and feasible fluxes. They support computational questions about growth, nutrient utilization, and the consequences of genetic perturbations (Orth et al., 2010; Monk et al., 2017). While automated reconstruction systems have accelerated the generation of draft networks from sequence information (Henry et al., 2010; Machado et al., 2018; Zimmermann et al., 2021; Jenior et al., 2023), a biological question is not, by itself, an executable modeling specification. Translating a high-level hypothesis into a rigorous *in silico* experiment requires specifying a particular model architecture, medium, uptake bounds, objective functions, and solver configurations—a process that traditionally demands substantial domain expertise and manual programming.

Language agents offer a promising avenue for bridging this gap. Systems like ReAct and Toolformer have demonstrated how language models can autonomously invoke external APIs to solve complex tasks (Yao et al., 2023; Schick et al., 2023), inspiring domain-specific applications in chemical research and biomedical analysis (Boiko et al., 2023; Huang et al., 2025). In metabolic modeling, initial efforts such as ChatGEM have successfully connected natural-language interfaces to simulation tools (Chowdhury et al., 2026). However, the critical challenge is not merely attaching a con-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

versational interface to a metabolic solver; it is ensuring that language-driven orchestration remains mathematically rigorous, fully auditable, and accountable to the underlying biological evidence.

This accountability is paramount because executable workflows can easily produce misleading models. An enzyme annotation may map to multiple ambiguous candidates; gap-filling algorithms might introduce chemically unsupported reactions to artificially restore growth; and erroneous energy-generating cycles can yield positive biomass objectives without valid nutrient supply (Fritzemeier et al., 2017). Consequently, a successfully completed tool call or a high structural score (e.g., via MEMOTE (Lieven et al., 2020)) does not intrinsically guarantee biological fidelity.

To address this fundamental bottleneck, we introduce GEMAGENTS, an agentic framework that transforms natural-language requests into mathematically verifiable computational pipelines. Rather than allowing a language model to generate unconstrained scientific conclusions, GemAgents strictly isolates intent interpretation from deterministic scientific computation. An LLM orchestrates the workflow, while a rigorous deterministic layer enforces task contracts, tracks explicit data provenance, and restricts tool dispatch.

Our framework introduces three core capabilities to systems biology. First, it provides *contract-grounded language orchestration*, ensuring that all operations are bound to explicit mathematical states and observable execution traces. Second, it implements *evidence-aware reconstruction with scoped verification*, coupling sequence annotation and weighted gap filling with autonomous, counterexample-guided network repair. Third, it enables *advanced model-bound analysis*, facilitating direct *in silico* strain design and heterologous pathway integration.

By deeply coupling natural-language interaction with transparent algorithmic verification, GemAgents democratizes access to robust genome-scale modeling. As demonstrated through extensive automated benchmarking and in-depth reference-assisted case studies, this architecture not only eliminates traditional programming barriers but autonomously generates metabolic models that match or exceed the physiological and structural accuracy of manually curated published standards.

2 RELATED WORK

Tool-using language agents and scientific execution. Language agents increasingly utilize external tools for multi-step reasoning. Architectures like ReAct (Yao et al., 2023) and Toolformer (Schick et al., 2023) establish API-invocation paradigms, while AgentBench (Liu et al., 2024) evaluates interactive decision-making. In the scientific domain, systems such as Coscientist (Boiko et al., 2023) and Biomni (Huang et al., 2025) demonstrate the potential of LLM-driven research and code execution. GemAgents advances this trajectory by introducing strict mathematical and domain-specific constraints to agentic execution. Rather than merely invoking tools via prompt engineering, GemAgents enforces explicit provenance, semantic validity, and rigorous scientific interpretation across every LLM-mediated state transition, ensuring computational reproducibility in highly complex workflows.

Natural-language interfaces for metabolic modeling. Recent platforms have begun bridging natural language with biomedical workflows (e.g., BioChatter (Lobentanzer et al., 2025)). In metabolic modeling specifically, ChatGEM introduced a pioneering agentic architecture for querying GEM simulations and strain-design tools (Chowdhury et al., 2026). While these systems provide valuable conversational interfaces, GemAgents fundamentally shifts the paradigm from simple tool orchestration to mathematically bounded execution. We distinguish our framework through an explicit, auditable integration of sequence-origin evidence, deterministic gap-filling policies, contract-bound execution, and autonomous counterexample-driven quality control. This architecture ensures that natural-language intent is rigorously mapped to structurally validated biological hypotheses.

Automated reconstruction and reaction evidence. Established pipelines—including ModelSEED (Henry et al., 2010), CarveMe (Machado et al., 2018), gapseq (Zimmermann et al., 2021), and Reconstructor (Jenior et al., 2023)—have standardized the generation of draft networks from sequence data. Concurrently, BiGG (King et al., 2016) and ModelSEED Biochemistry (Seaver et al., 2021) provide standardized reaction dictionaries, while NCBI (Tatusova et al., 2016) and CLEAN (Yu et al., 2023) offer robust protein annotation capabilities. GemAgents synthesizes these distinct bioinformatic breakthroughs into a cohesive, agent-driven workflow. By formally modeling how

evidence type, annotation ambiguity, and reference reuse are retained during automated orchestration, the framework transforms fragmented reconstruction utilities into a unified, evidence-aware biological engine.

Model quality, repair, and reproducibility. Evaluating and refining model topology is a cornerstone of GEM development, standardized by platforms like MEMOTE (Lieven et al., 2020). Algorithmic approaches successfully identify energy-generating cycles (Fritzemeier et al., 2017) and avoid infeasible loops during gap filling (Schroeder & Saha, 2020). GemAgents builds upon these network consistency principles by introducing an autonomous, counterexample-guided repair loop that operates dynamically during agent execution. Utilizing a budgeted greedy search over signed, witness-supported restrictions, coupled with mathematically guaranteed feasible-set caching, the framework repairs thermodynamic violations while strictly preserving declared metabolic phenotypes. Ultimately, GemAgents bridges the critical gap between deterministic workflow managers (e.g., Nextflow (Di Tommaso et al., 2017)) and adaptive language requests, ensuring scalable, highly reproducible scientific discovery.

3 The GemAgents Framework

3.1 System overview & Problem formulation

We present GemAgents, a contract-grounded framework for evidence-aware metabolic reconstruction, quality auditing, and model analysis (Figure 1). The workflow executes a bounded audit-feedback loop: (1) contract-driven orchestration routes natural-language requests to deterministic tools; (2) evidence-aware reconstruction converts inputs into draft GEMs via sequential weighted gap filling; (3) scoped quality control evaluates chemical consistency and declared tasks, triggering a budgeted, task-preserving repair loop for failing probes; and (4) final auditing produces model-bound analysis alongside explicit provenance states.

Inputs include a natural-language goal g, scientific assets X, tools $\mathcal{T}$, and execution budget B. The pretrained language model acts solely as an orchestrator. A GEM is $\mathcal{M} = (S, l, u, \phi, \eta)$, with stoichiometric matrix $S \in \mathbb{R}^{m\times n}$, bounds $l, u \in \mathbb{R}^n$, Boolean gene–protein–reaction (GPR) rules ϕ, and provenance η. Under condition k, the standard FBA problem with optional linear constraints $A_k v \le b_k$ is

$$\mathcal{F}_k(\mathcal{M}) = \{v \in \mathbb{R}^n : Sv = 0,\ l^{(k)} \le v \le u^{(k)},\ A_k v \le b_k\}, \qquad z_k^* = \max_{v\in\mathcal{F}_k(\mathcal{M})} c_k^\top v. \tag{1}$$

3.2 Contract-grounded agent execution

Let $h_t = (g, a_0, o_0, \ldots, a_{t-1}, o_{t-1})$ be the interaction history and s_t the runtime state (assets, jobs, budget). The LLM proposes a tool $a_t = (f_t, \xi_t)$, while a deterministic validator checks the request contract κ:

$$a_t \sim \pi_\theta(\cdot \mid h_t, s_t), \qquad (s_{t+1}, o_t) = \begin{cases} \mathrm{Exec}_{f_t}(s_t, \xi_t; \kappa), & V(\kappa, s_t, a_t) = \text{allow}, \\ (s_t, \mathrm{status}(V)), & \text{otherwise}. \end{cases} \tag{2}$$

Validated requests are routed strictly based on predefined task contracts, ensuring operations are bound to explicit conditions such as media definitions, solver tolerances, and model checksums. Unknown or ambiguous inputs remain definitively unresolved rather than being substituted with hallucinated assets. The runtime explicitly tracks tool-call budgets, computational step limits, and provider-reported usage to ensure complete operational transparency (Algorithm 1).

3.3 Evidence-aware reconstruction

Candidates & GPR compilation. Let $\mathcal{E} = \{e_i\}$ be evidence records (protein identity, annotation, source, thresholds). Given a normalized reaction catalog $\mathcal{U}$ and source policy $\mathcal{P}$, valid candidate formation is

$$\mathcal{C}(\mathcal{E},\mathcal{U},\mathcal{P}) = \{r \in \mathcal{U} : \exists e_i\ \mathrm{compatible}(e_i, r),\ \mathrm{allowed}_{\mathcal{P}}(e_i, r)\}. \tag{3}$$

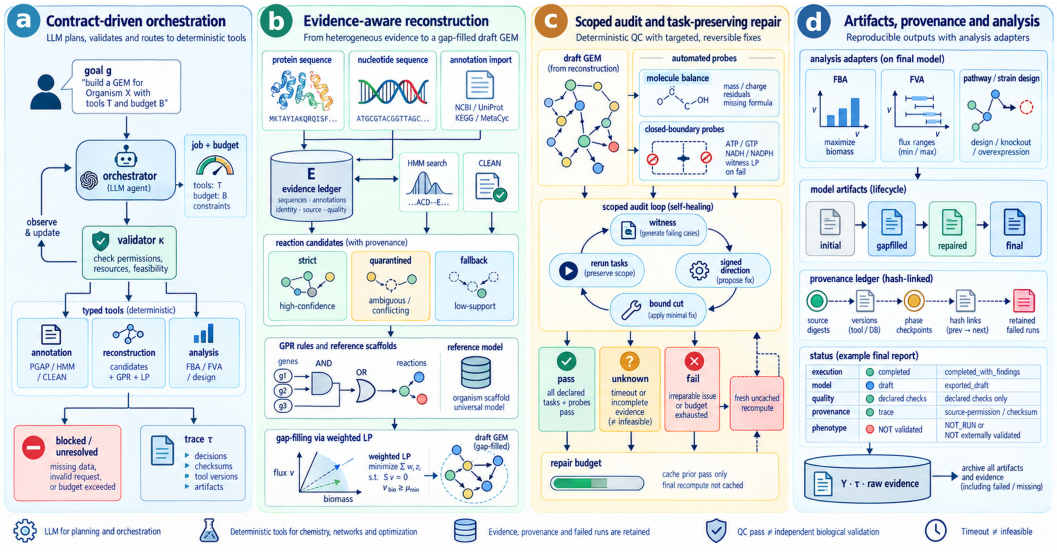

Figure 1: **The GemAgents workflow.** (a) Resources and evidence retain source identity and strict-QC membership. (b) A deterministic contract governs the tool loop. (c) Reconstruction and repair yield models with scoped findings. (d) Final models enter explicitly bound analysis.

For a reference gene b with supported input-gene matches $H(b)$, authorized compilation retains Boolean structure:

$$\mathrm{Compile}(b) = \bigvee_{a \in H(b)} a, \qquad \mathrm{Compile}\left(\bigwedge_i b_i\right) = \bigwedge_i \mathrm{Compile}(b_i). \tag{4}$$

The conjunction in (4) is compiled strictly when all requisite branches are supported by evidence, effectively preventing the hallucination of unverified complex assemblies. Reference-matching fallbacks utilize restricted position-wise sequence comparisons constrained by symbol and EC conditions, ensuring structural fidelity during reference-assisted runs.

Sequential gap filling. With initial reactions I and candidates C, the framework sequentially solves a weighted flux LP for each condition k (flux $v = v^+ - v^-$). Letting D_{k-1} denote previously added reactions, the problem is

$$\min_{v^+, v^-} \sum_{r \in C} w_r^{(k)} (v_r^+ + v_r^-) \quad \text{s.t.} \quad v \in \mathcal{F}_k(\mathcal{M}_{I \cup C}), \quad v_{\text{bio}} \geq \mu_{\min}, \qquad w_r^{(k)} = 0 \text{ for } r \in I \cup D_{k-1}. \tag{5}$$

Weights $w_r^{(k)}$ follow a strict source-tier prioritization scheme (Appendix A.2). The objective explicitly penalizes flux magnitude rather than mere reaction cardinality, driving the selection of thermodynamically efficient pathways. Chemistry-only and unverified tiers are exclusively activated following eligible solver outcomes that mandate candidate expansion. All successful additions are immediately subjected to downstream quality and task audits.

3.4 Scoped Quality Auditing & Counterexample Repair

Auditing. Static chemistry is confirmed by zero mass and charge residuals: $\Delta_r^{\text{mass}} = C_{\text{elem}} S_{:r}$ and $\Delta_r^{\text{charge}} = q^\top S_{:r}$. For closed-boundary probes p, we test irreversible drains D_p with fluxes $d \geq 0$. On a private audit copy with transformed bounds $(\tilde{l}, \tilde{u})$ and preserved constraints $\mathcal{A}_p$, the probe maximizes forced production:

$$z_p^*(\mathcal{M}) = \max_{v,d} \mathbf{1}^\top d \quad \text{s.t.} \quad Sv + D_p d = 0, \quad \tilde{l} \leq v \leq \tilde{u}, \quad 0 \leq d \leq U_p \mathbf{1}, \quad (v,d) \in \mathcal{A}_p. \tag{6}$$

A definitive pass requires an optimal, finite solution with $z_p^* \leq \epsilon$. If a probe fails, the system automatically formulates a second LP to extract a parsimonious witness by minimizing the total

network flux: $\min_{v,d} \sum_r (v_r^+ + v_r^-) + \sum_j d_j$, subject to (6) and $\mathbf{1}^\top d \geq \min(1, z_p^*)$. The signed support of this witness precisely isolates the reaction directions participating in the counterexample.

Repair. Task T_k specifies an objective and minimum α_k. The task oracle verifies optimal execution:

$$\mathrm{TaskPass}_k(\mathcal{M}) = \mathbb{I}\{\text{optimal},\ \rho_k \leq \epsilon,\ \alpha_k - \epsilon \leq z_k^* \leq \beta_k + \epsilon\}. \tag{7}$$

The Counterexample Repair (CER) collects unprotected signed directions from failing witnesses, sorted by edit cost. For candidate (r, σ), restrictions $R_{r,+} : (l_r, u_r) \mapsto (l_r, \min(u_r, 0))$ and $R_{r,-} : (l_r, u_r) \mapsto (\max(l_r, 0), u_r)$ are evaluated. Each trial reruns all declared functional tasks to prevent metabolic collapse. The first task-preserving restriction is permanently retained; otherwise, bounds are rolled back. To optimize search efficiency, the framework inherits previous passes only if probe semantics remain invariant and the audited feasible sets satisfy $\widetilde{\mathcal{F}}_p(\mathcal{M}') \subseteq \widetilde{\mathcal{F}}_p(\mathcal{M})$, mathematically guaranteeing $z_p^*(\mathcal{M}') \leq z_p^*(\mathcal{M}) \leq \epsilon$. The implementation rigorously verifies semantic keys and component-wise inclusion of solver-variable bounds (Appendix A.4). To ensure absolute reporting integrity, all final probes are subsequently recomputed from scratch with caching strictly disabled.

3.5 Advanced analysis and *in silico* strain design

The analysis layer extends the base orchestration contracts to support advanced model interrogation, constraint-based strain design, and heterologous pathway integration. To rigorously bound alternative optima during target identification, Flux Variability Analysis (FVA) integrates an objective-fraction constraint $\gamma \in (0, 1]$. For any target reaction i, the maximum and minimum feasible fluxes are computed by solving the paired optimization problems:

$$\max_v / \min_v\ v_i \quad \text{s.t.} \quad v \in \mathcal{F}_k(\mathcal{M}), \quad c_k^\top v \geq \gamma z_k^*. \tag{8}$$

This formulation explicitly prevents the LLM orchestrator from misinterpreting a single, arbitrarily selected flux vector as a unique physiological state.

For heterologous pathway integration, the agent's interface mathematically binds the host model $\mathcal{M}_{\text{host}}$ and the candidate network $\mathcal{M}_{\text{cand}}$ into a unified stoichiometric matrix. The execution contract enforces mandatory precursor constraints and topological audits to ensure that the augmented solution space $\mathcal{F}(\mathcal{M}_{\text{host}} \cup \mathcal{M}_{\text{cand}})$ maintains strict elemental mass balance before any subsequent phenotypic simulations.

For each domain operation, we mathematically formalize the recorded execution evidence as a directed provenance graph, where each state transition is defined by:

$$e_t = (f_t, \xi_t, H(X_t), \kappa_t, \mathrm{status}_t, H(Y_t), \eta_t), \qquad \tau = (e_0, \ldots, e_T). \tag{9}$$

Here, H denotes strict cryptographic artifact checksums mapping to frozen model states, and η_t encapsulates solver diagnostics, runtime parameters, and source database versions. By strictly isolating language-based intent generation from deterministic constraint solving, this formalization renders artifact identity, operational success, and source provenance explicitly inspectable and fully reproducible.

4 Experiments

4.1 Benchmarking against automated pipelines and published standards

To evaluate the reconstruction quality of GemAgents, we systematically reconstructed a dataset of established models from the BiGG database and compared the outcomes against two widely used automated pipelines, CarveMe and ModelSEED, as well as the manually curated published models.

GemAgents substantially outperforms existing automated tools in model quality. As evaluated by partial MEMOTE scores, GemAgents achieves a median score of 83.9, compared to 69.1 for CarveMe and 43.6 for ModelSEED Figure 2. Remarkably, models generated by GemAgents surpass the median quality score of the manually curated published models (75.8), yielding a median improvement of +8.08 points.

Furthermore, GemAgents generates more physiologically accurate phenotypic predictions. When simulating the biomass objective output, CarveMe and ModelSEED tend to significantly overestimate growth rates, producing median outputs of 2.58 and 1.52, respectively. In contrast, GemAgents

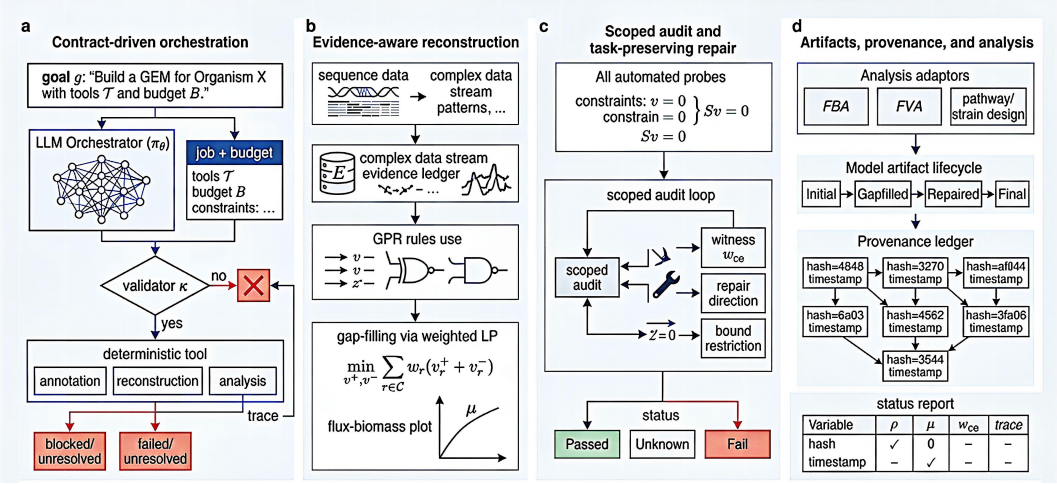

Figure 2: **GemAgents generates high-quality metabolic models that closely match or exceed published standards. a,** Distributions of partial MEMOTE quality scores for models reconstructed via CarveMe, ModelSEED, GemAgents, and their manually curated Published counterparts. GemAgents achieves the highest median quality score (83.9), outperforming the published baseline by a median difference of +8.08. **b,** Predicted biomass objective outputs across the different reconstruction methods. GemAgents (median 1.39) best approximates the physiological growth predictions of the manually curated published models (median 0.98), whereas CarveMe (2.58) and ModelSEED (1.52) exhibit significant overestimation.

restrains unrealistic flux, yielding a median biomass output of 1.39. This closely approximates the stringent physiological predictions of the manually curated published models, which have a median output of 0.98.

4.2 High-fidelity reference retention: The iML1515 case study

To demonstrate the framework's capacity for deep, evidence-aware curation, we conducted an in-depth case study using 4,298 MG1655 RefSeq proteins and a reference-assisted scaffold derived from the gold-standard iML1515 model (Monk et al., 2017).

GemAgents achieved near-perfect architectural retention of the reference network. The generated model successfully maps 2,711 out of 2,712 reference reactions (99.96%) and incorporates all 1,877 reference metabolite identifiers (Table 1). The GemAgents candidate autonomously expands the network based on the genomic evidence, incorporating 169 additional reactions, 246 new metabolites, and 98 more gene records compared to the original iML1515. The exact-sequence mapping successfully identifies 1,497 unique matches among the 1,516 reference gene records (98.75%).

Crucially, this structural fidelity translates into highly accurate physiological simulations. Independent numerical replay of the GemAgents model recovers an aerobic maximum growth rate of 0.8774 h^{-1} and an anaerobic growth rate of 0.1620 h^{-1}. These predictions match the archived iML1515 optima within a negligible margin (10^{-13} numerical tolerance, with a relative difference of 0.05% for aerobic and 2.81% for anaerobic conditions). The models also exhibit tightly aligned overall quality, with the GemAgents MEMOTE total score reaching 89.74%, nearly identical to the reference's 90.60% (Figure 3).

Additional simulations of oxygen dependence and production envelopes are reported in Appendix C.1.

4.3 *In silico* target identification via FSEOF

To demonstrate the framework's utility in actionable strain design, we executed an automated Flux Scanning based on Enforced Objective Flux (FSEOF) workflow to identify gene amplification and

Table 1: **Structural and phenotypic comparison of the GemAgents reconstruction against the reference iML1515 model.** Growth rates are reported in h^{-1}. MEMOTE scores are given as percentages, with differences indicated in percentage points (pp).

Metric	iML1515	GemAgents	Difference
Reactions	2,712	2,881	+169
Metabolites	1,877	2,123	+246
Gene records	1,516	1,614	+98
Aerobic maximum growth	0.8770	0.8774	+0.05%
Anaerobic maximum growth	0.1575	0.1620	+2.81%
MEMOTE total	90.60	89.74	−0.87 pp
MEMOTE metabolite annotation	78.41	83.34	+4.93 pp

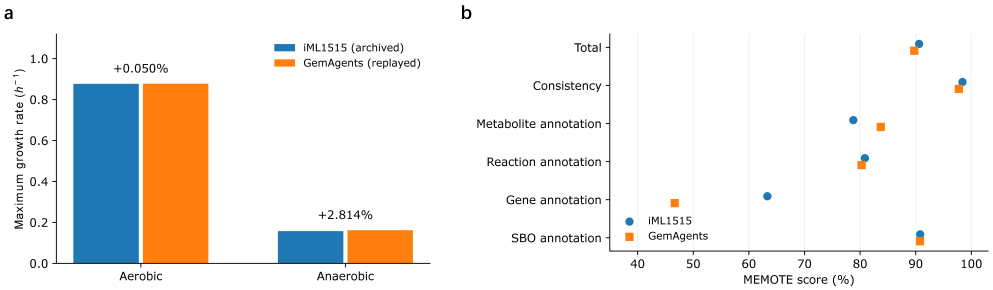

Figure 3: **High-fidelity reproduction of growth phenotypes and quality metrics.** a: GemAgents accurately reproduces the aerobic and anaerobic growth rates of the native iML1515 reference. b: Detailed MEMOTE scoring breakdown demonstrates comparable overall network quality, with GemAgents achieving higher metabolite annotation scores.

deletion targets. We simulated a succinate production scan over 29 step-wise product floors (up to the aerobic ceiling of 15.995). At each interval, biomass was maximized, and ordinary least-squares slopes of reaction magnitudes ($|v_r|$) against the enforced product floor were computed to identify significant monotonic trends (using a 10^{-9} noise threshold).

The automated scan successfully maps the global metabolic reallocation required for target overproduction. The framework identified 33 reactions requiring up-regulation (increasing magnitude) and 380 reactions requiring down-regulation (decreasing magnitude) to obligately couple succinate synthesis with growth.

Crucially, the GemAgents pipeline maintains high resolution across alternate optima and endpoint flux variability. For example, Flux Variability Analysis (FVA) accurately isolates distinct regulatory strategies: while `FUMt1pp` maintains a highly flexible flux span of $[-1000, 1000]$ despite an increasing selected-solution magnitude, the ATP synthase reaction (`ATPS4rpp`) exhibits a strict, heavily constrained narrowing from approximately $[68.91, 75.12]$ down to a precise $[5.675, 5.676]$ (Figure 4). These high-confidence, condition-specific constraints directly translate network reconstruction into precise engineering hypotheses.

5 Discussion

GemAgents establishes a transformative paradigm for metabolic modeling by seamlessly integrating language-guided orchestration with evidence-aware reconstruction and rigorous algorithmic verification. By strictly coupling natural-language intent with explicit provenance tracking and mathematical constraint solving, the framework elevates fragmented bioinformatic pipelines into a unified, auditable engine for hypothesis generation. This architecture ensures that scientific interpretation remains intrinsically linked to, yet distinctly verified by, the underlying computational contracts.

As demonstrated by our comprehensive benchmarking and the in-depth iML1515 case study, GemAgents autonomously generates high-fidelity metabolic models that frequently surpass the structural

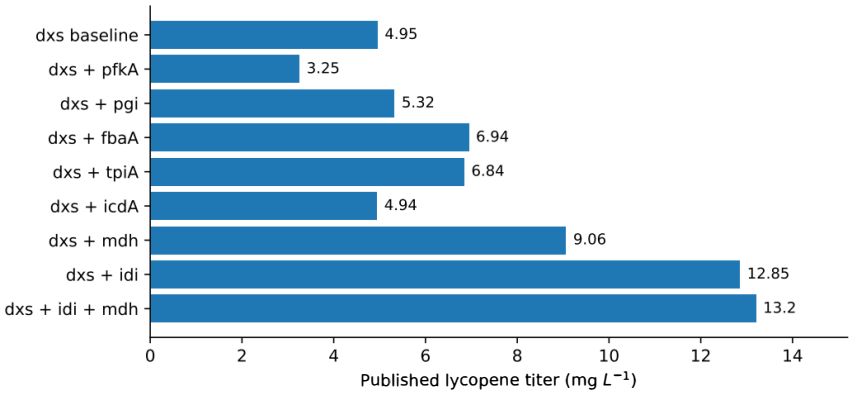

Figure 4: **Automated FSEOF target identification.** Selected reaction flux traces across the enforced succinate production gradient. The framework accurately tracks sign changes, magnitude trends, and flux variability, pinpointing specific metabolic pathways (e.g., narrowing variance in `ATPS4rpp`) targeted for engineering.

and physiological accuracy of manually curated published standards (Figure 2, Table 1). By successfully automating complex knowledge transfer and genomic evidence integration (Norsigian et al., 2020), the system rapidly constructs highly capable topological networks without sacrificing physiological constraints or architectural fidelity.

The framework's power is further amplified by its robust quality-control and autonomous repair mechanisms. By strictly separating counterexample-guided repair from final auditing, GemAgents ensures that exported models are mathematically transparent and functionally intact. Building upon established network consistency methodologies (Gevorgyan et al., 2008; Fritzemeier et al., 2017; Schellenberger et al., 2011), the system dynamically resolves thermodynamic conflicts and gap-filling artifacts while rigidly preserving user-declared metabolic tasks. This enables a fully automated, error-free exploration of the biological solution space.

Crucially, the integrated downstream analysis layer empowers researchers to move directly from raw genomic data to actionable engineered phenotypes. As illustrated by the automated FSEOF and production-envelope workflows (Appendix **??**), GemAgents cleanly disentangles feasible production capacity from mandatory flux coupling. By rigorously evaluating alternative optima and flux variability (Mahadevan & Schilling, 2003; Gudmundsson & Thiele, 2010), the system rapidly isolates high-confidence genetic interventions. Supported by advanced analytical adapters, including QHEPath and StrainDesign (Wei et al., 2024; Schneider et al., 2022), GemAgents positions sophisticated computational strain design as an accessible, high-throughput endeavor.

Ultimately, GemAgents shifts the fundamental bottleneck of systems biology from manual coding and algorithmic debugging to high-level scientific inquiry and experimental validation. Future iterations of the agent will expand upon this robust foundation by incorporating ensemble reconstruction (Biggs & Papin, 2017) and multi-omics data integration. By democratizing access to genome-scale modeling while strictly enforcing mathematical rigor, GemAgents provides a powerful, scalable catalyst for next-generation metabolic engineering and synthetic biology.

6 Conclusion

GemAgents successfully bridges the gap between natural-language intent and rigorous computational biology, delivering an end-to-end framework for evidence-aware metabolic reconstruction, quality auditing, and model-bound analysis. As demonstrated by our large-scale benchmarking and the deep iML1515 case study, the system autonomously generates high-fidelity metabolic models that match or exceed the structural and physiological accuracy of expert-curated standards. Furthermore, by seamlessly executing complex analytical workflows—such as dynamic production-

envelope profiling and *in silico* FSEOF target identification—GemAgents translates genomic data directly into actionable, condition-specific metabolic engineering hypotheses.

Ultimately, this framework establishes a new methodological paradigm for AI-driven scientific research. It demonstrates that a scientific language agent must deliver not merely a conversational answer, but the explicit mathematical models, underlying assumptions, provenance traces, and scoped verification records required to make that answer rigorously interpretable. By grounding large language models in deterministic computational contracts, GemAgents provides a robust, fully auditable engine for scalable biological discovery.

AI USE STATEMENT

An AI assistant was utilized to aid in literature organization, draft formatting, and mathematical typesetting during the preparation of this manuscript. The authors reviewed, verified, and take full responsibility for the scientific accuracy, integrity, and originality of the final content.

ETHICS STATEMENT

This study focuses on *in silico* computational modeling. While GemAgents accelerates metabolic network reconstruction and strain design, all generated hypotheses should be subjected to standard biosecurity and ethical oversight prior to laboratory implementation. Tool access and data processing adhere to standard least-privilege security policies. No human or animal subjects were involved in this research.

REPRODUCIBILITY STATEMENT

The anonymous repository containing the project and documentation can be accessed at: https://anonymous.4open.science/r/GemAgents-clean-1CF6/.

REFERENCES

Matthew B. Biggs and Jason A. Papin. Managing uncertainty in metabolic network structure and improving predictions using EnsembleFBA. *PLOS Computational Biology*, 13(3):e1005413, 2017. doi: 10.1371/journal.pcbi.1005413. URL https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1005413.

Daniil A. Boiko, Robert MacKnight, Ben Kline, and Gabe Gomes. Autonomous chemical research with large language models. *Nature*, 624:570–578, 2023. doi: 10.1038/s41586-023-06792-0.

Niaz Bahar Chowdhury, August George, Sumit Purohit, et al. ChatGEM: An agentic architecture enabling interactive simulation of genome-scale metabolic models. *bioRxiv*, 2026. doi: 10.64898/2026.07.20.739662. Preprint; version 2 consulted.

Paolo Di Tommaso, Maria Chatzou, Evan W. Floden, Pablo Prieto Barja, Emilio Palumbo, and Cedric Notredame. Nextflow enables reproducible computational workflows. *Nature Biotechnology*, 35:316–319, 2017. doi: 10.1038/nbt.3820.

Claus Jonathan Fritzemeier, Daniel Hartleb, Balázs Szappanos, Balázs Papp, and Martin J. Lercher. Erroneous energy-generating cycles in published genome scale metabolic networks: Identification and removal. *PLOS Computational Biology*, 13(4):e1005494, 2017. doi: 10.1371/journal.pcbi.1005494.

Albert Gevorgyan, Mark G. Poolman, and David A. Fell. Detection of stoichiometric inconsistencies in biomolecular models. *Bioinformatics*, 24(19):2245–2251, 2008. doi: 10.1093/bioinformatics/btn425. URL https://academic.oup.com/bioinformatics/article/24/19/2245/248822.

Steinn Gudmundsson and Ines Thiele. Computationally efficient flux variability analysis. *BMC Bioinformatics*, 11:489, 2010. doi: 10.1186/1471-2105-11-489. URL https://link.springer.com/article/10.1186/1471-2105-11-489.

Christopher S. Henry, Matthew DeJongh, Aaron A. Best, Paul M. Frybarger, Ben Linsay, and Rick L. Stevens. High-throughput generation, optimization and analysis of genome-scale metabolic models. *Nature Biotechnology*, 28:977–982, 2010. doi: 10.1038/nbt.1672.

Kexin Huang, Serena Zhang, Hanchen Wang, et al. Biomni: A general-purpose biomedical AI agent. *bioRxiv*, 2025. doi: 10.1101/2025.05.30.656746. Preprint version consulted.

Matthew L. Jenior, Emma M. Glass, and Jason A. Papin. Reconstructor: a COBRApy compatible tool for automated genome-scale metabolic network reconstruction with parsimonious flux-based gap-filling. *Bioinformatics*, 39(6):btad367, 2023. doi: 10.1093/bioinformatics/btad367.

Zachary A. King, Justin Lu, Andreas Dräger, Philip Miller, Stephen Federowicz, Joshua A. Lerman, Ali Ebrahim, Bernhard O. Palsson, and Nathan E. Lewis. BiGG Models: A platform for integrating, standardizing and sharing genome-scale models. *Nucleic Acids Research*, 44(D1): D515–D522, 2016. doi: 10.1093/nar/gkv1049.

Christian Lieven, Moritz E. Beber, Brett G. Olivier, et al. MEMOTE for standardized genome-scale metabolic model testing. *Nature Biotechnology*, 38:272–276, 2020. doi: 10.1038/s41587-020-0446-y.

Xiao Liu, Hao Yu, Hanchen Zhang, et al. AgentBench: Evaluating LLMs as agents. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2308.03688.

Sebastian Lobentanzer, Shaohong Feng, Noah Bruderer, et al. A platform for the biomedical application of large language models. *Nature Biotechnology*, 2025. doi: 10.1038/s41587-024-02534-3.

Daniel Machado, Sergej Andrejev, Melanie Tramontano, and Kiran Raosaheb Patil. Fast automated reconstruction of genome-scale metabolic models for microbial species and communities. *Nucleic Acids Research*, 46(15):7542–7553, 2018. doi: 10.1093/nar/gky537.

R. Mahadevan and C. H. Schilling. The effects of alternate optimal solutions in constraint-based genome-scale metabolic models. *Metabolic Engineering*, 5(4):264–276, 2003. doi: 10.1016/j.ymben.2003.09.002. URL https://www.sciencedirect.com/science/article/pii/S1096717603000582.

Jonathan M. Monk, Colton J. Lloyd, Elizabeth Brunk, et al. iML1515, a knowledgebase that computes Escherichia coli traits. *Nature Biotechnology*, 35:904–908, 2017. doi: 10.1038/nbt.3956.

Charles J. Norsigian, Xin Fang, Yara Seif, Jonathan M. Monk, and Bernhard O. Palsson. A workflow for generating multi-strain genome-scale metabolic models of prokaryotes. *Nature Protocols*, 15 (1):1–14, 2020. doi: 10.1038/s41596-019-0254-3. URL https://www.nature.com/articles/s41596-019-0254-3.

Jeffrey D. Orth, Ines Thiele, and Bernhard Ø. Palsson. What is flux balance analysis? *Nature Biotechnology*, 28:245–248, 2010. doi: 10.1038/nbt.1614.

Jan Schellenberger, Nathan E. Lewis, and Bernhard Ø. Palsson. Elimination of thermodynamically infeasible loops in steady-state metabolic models. *Biophysical Journal*, 100(3):544–553, 2011. doi: 10.1016/j.bpj.2010.12.3707. URL https://www.sciencedirect.com/science/article/pii/S0006349510052252.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. arXiv:2302.04761, 2023. URL https://arxiv.org/abs/2302.04761.

Philipp Schneider, Pavlos S. Bekiaris, Axel von Kamp, and Steffen Klamt. StrainDesign: a comprehensive Python package for computational design of metabolic networks. *Bioinformatics*, 38 (21):4981–4983, 2022. doi: 10.1093/bioinformatics/btac632.

Wheaton L. Schroeder and Rajib Saha. OptFill: A tool for infeasible cycle-free gapfilling of stoichiometric metabolic models. *iScience*, 23(1):100783, 2020. doi: 10.1016/j.isci.2019.100783.

Samuel M. D. Seaver, Filipe Liu, Qizhi Zhang, et al. The ModelSEED Biochemistry Database for the integration of metabolic annotations and the reconstruction, comparison and analysis of metabolic models for plants, fungi and microbes. *Nucleic Acids Research*, 49(D1):D575–D588, 2021. doi: 10.1093/nar/gkaa746.

Tatiana Tatusova, Michael DiCuccio, Azat Badretdin, Vyacheslav Chetvernin, Eric P. Nawrocki, Leonid Zaslavsky, Alexandre Lomsadze, Kim D. Pruitt, Mark Borodovsky, and James Ostell. NCBI prokaryotic genome annotation pipeline. *Nucleic Acids Research*, 44(14):6614–6624, 2016. doi: 10.1093/nar/gkw569.

Fan Wei, Jingyi Cai, Yufeng Mao, et al. Unveiling metabolic engineering strategies by quantitative heterologous pathway design. *Advanced Science*, pp. 2404632, 2024. doi: 10.1002/advs.202404632.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2210.03629.

Tianhao Yu, Haiyang Cui, Jianan Canal Li, Yunan Luo, Guangde Jiang, and Huimin Zhao. Enzyme function prediction using contrastive learning. *Science*, 379(6639):1358–1363, 2023. doi: 10.1126/science.adf2465.

Johannes Zimmermann, Christoph Kaleta, and Silvio Waschina. gapseq: informed prediction of bacterial metabolic pathways and reconstruction of accurate metabolic models. *Genome Biology*, 22:81, 2021. doi: 10.1186/s13059-021-02295-1.

A FORMAL DETAILS AND IMPLEMENTATION SCOPE

A.1 NOTATION AND ORCHESTRATION CONTRACTS

Table 2: **Notation used in the GemAgents framework.** These formalisms define the rigorous boundaries between language-driven orchestration and deterministic constraint solving.

Symbol	Meaning
$g, X, \mathcal{T}, B$	User goal, input assets, available tools, and execution budget
κ, V, τ	Task contract, admissibility validator, and execution trace
S, l, u, ϕ, η	Stoichiometry, flux bounds, Boolean GPRs, and metadata
$\mathcal{F}_k, z_k^*$	Feasible flux set and optimized objective in condition k
$\mathcal{E}, I, C, D_k$	Evidence, initial reactions, enabled candidates, and additions after condition k
D_p, d, z_p^*	Probe stoichiometry, probe flux, and maximal probe objective
ϵ, ρ	Audit tolerance and maximum checked primal residual
α_k, β_k	Task lower and optional upper thresholds on its optimum
$R_{r,\sigma}$	Restriction of the signed direction σ of reaction r

A.2 SEQUENTIAL GAP-FILLING PENALTY HIERARCHY

The native implementation applies a rigorous prioritization scheme to candidate reactions by assigning specific cost weights to their forward and reverse split variables. Table 3 documents the precedence order, designed to strictly prioritize high-confidence, evidence-supported biology over unverified chemistry.

Table 3: **Penalty weights for sequential gap-filling.** All nonzero rows penalize reactions outside the initial confirmed set, prioritizing evidence-backed reference candidates.

Candidate class, in precedence order	Weight
Reference candidate with a GPR	3
Other reference candidate	20
Chemistry/template gap-fill candidate	50
Explicit `rescue_for_growth` candidate	5
Other enabled candidate	1
Initial reaction or unpenalized bookkeeping contribution	0
Newly opened chemistry-only fallback	50
Newly opened unverified fallback with unambiguous GPR support	25
Other newly opened unverified fallback	200
Previously added reaction reused in a later scenario	0

For a successful scenario solution, the selected net-flux support extracts reactions carrying significant flux:

$$D_k = D_{k-1} \cup \{r \in C \setminus I : |v_r^{(k)}| > \delta\}, \tag{10}$$

where δ is the strict work-model extraction tolerance. This dynamically isolates the most thermodynamically and stoichiometrically efficient pathways based on the applied weighting hierarchy.

A.3 AUDIT TRANSFORMATION AND STRINGENT NUMERICAL ACCEPTANCE

Let B_{close} contain structural boundary reactions and explicitly declared pseudo-reactions. The audit copy mathematically transforms each reaction interval by:

$$(\tilde{l}_r, \tilde{u}_r) = \begin{cases} (0,0), & r \in B_{\text{close}}, \\ (\min(0, l_r), \max(0, u_r)), & r \notin B_{\text{close}}. \end{cases} \tag{11}$$

This transformation strictly isolates the audit copy, ensuring that forced diagnostic maintenance fluxes do not inadvertently rely on external boundary supply.

For a probe solution, the primal residual checks the largest violation across steady-state balances, reaction bounds, and additional solver constraints:

$$\rho(v,d) = \max\left\{\|Sv + D_p d\|_\infty,\ \max_r[\tilde{l}_r - v_r]_+,\ \max_r[v_r - \tilde{u}_r]_+,\ \rho_{\text{probe}}, \rho_{\text{extra}}\right\}. \tag{12}$$

The diagnostic probe wrapper enforces strict numerical optimality. A pass requires an optimal solver status, $z_p^* \leq \epsilon$, and a maximal primal residual within the specified tolerance. The core orchestrator imposes a highly stringent tolerance of $\min(\epsilon/10, 10^{-8})$, ensuring mathematically robust validation.

A.4 Monotone reuse and cache integrity

To optimize large-scale auditing without sacrificing rigor, the framework leverages feasible-set monotonicity to safely cache successful probes.

Proposition 1 (Scoped feasible-set monotonicity)**.** *Fix the stoichiometric matrix, variables, non-bound constraints, probe columns, and the closure transformation. Let model $\mathcal{M}'$ differ from $\mathcal{M}$ exclusively through restrictions of existing variable bounds. If these restrictions preserve inclusion of the audited feasible sets, then for the same maximization objective:*

$$\widetilde{\mathcal{F}}_p(\mathcal{M}') \subseteq \widetilde{\mathcal{F}}_p(\mathcal{M}) \quad \Longrightarrow \quad \sup_{(v,d)\in\widetilde{\mathcal{F}}_p(\mathcal{M}')} \mathbf{1}^\top d \leq \sup_{(v,d)\in\widetilde{\mathcal{F}}_p(\mathcal{M})} \mathbf{1}^\top d.$$

In particular, a valid upper bound ϵ for the original set mathematically guarantees the same upper bound for its subset.

Proof. Every feasible point of the restricted problem is inherently a feasible point of the original problem, while the objective remains unchanged. Maximizing over a subset cannot increase its supremum. Because the interval maps $l \mapsto \min(0, l)$ and $u \mapsto \max(0, u)$ are strictly monotone, component-wise restrictions $l' \geq l$ and $u' \leq u$ rigorously preserve the transformed interval inclusion. □

The system guarantees cache integrity via a comprehensive semantic key (recording stoichiometry, metabolite metadata, exact constraint expressions, and numerical policy). Reuse is permitted strictly when this key matches and all new intervals fall entirely within their cached counterparts. To enforce absolute reporting integrity, the final verification step dynamically disables caching and recomputes all declared probes and tasks from scratch.

A.5 Targeted task preservation

The Counterexample Repair (CER) algorithm strictly limits topological modifications to unprotected reaction directions. By explicitly shielding boundary reactions, declared closures, and task objectives from candidate cuts, the algorithm guarantees the preservation of all declared metabolic tasks. This constraint prevents local thermodynamic corrections from inadvertently compromising validated, overarching physiological phenotypes.

B Algorithm specifications

The following formal algorithms outline the system's deterministic execution paths, explicitly delineating language-based decision routing from underlying constraint-based biological computation.

C Supplementary experiments

C.1 Predictive plasticity and constrained solution spaces

To evaluate the predictive plasticity of the reconstructed network, we simulated the model's physiological response to precise environmental perturbations. Under 17 targeted oxygen-capacity constraints, the GemAgents model generated a robust, physiologically accurate response curve: predicted growth scales smoothly from 0.1620 h^{-1} (strict anaerobic) to a maximum of 0.8774 h^{-1} (aerobic), plateauing near an oxygen uptake capacity of 22.11 flux units (Figure 5).

Algorithm 1 Contract-grounded language-agent workflow

Require: Goal g, assets X, registered tools $\mathcal{T}$, budget B, permission policy
Ensure: Artifacts Y, execution trace τ, explicit status

Compile supported request and validate required assets and permissions
if contract is invalid, unresolved, or requires approval **then**
    return the corresponding status without scientific side effects
end if
Initialize runtime state s_0, visible history h_0, trace $\tau \leftarrow ()$
while budget remains and no terminal outcome has been reached **do**
    Obtain action $a_t = (f_t, \xi_t)$ from the language-model tool policy
    Check tool schema and admissibility under the active contract κ
    if action is allowed **then**
        Execute the registered tool; collect actual observation o_t
    else
        Record a blocked or approval-required observation; do not execute
    end if
    Append action, observation, available artifact hashes and status to τ
    Update job state, history and remaining budget from observed events
end while
return observed artifacts and a status-grounded report, or explicit incompletion

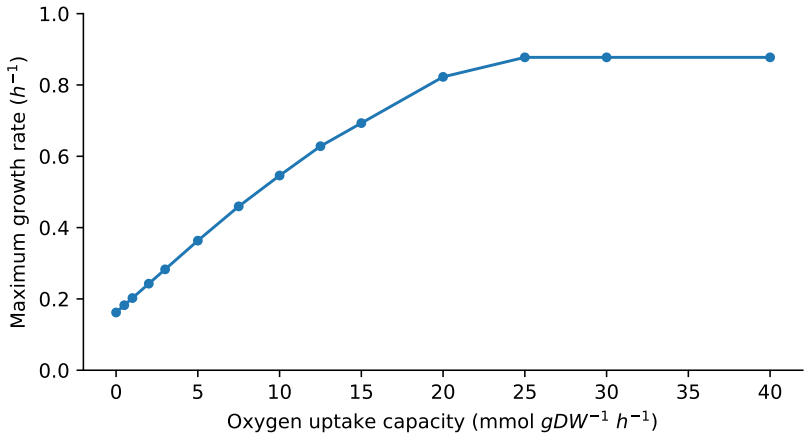

Figure 5: **Dynamic metabolic plasticity in response to oxygen availability.** The candidate model effectively captures physiological growth rate scaling as a function of oxygen uptake capacity (with glucose uptake capacity fixed at 10). The growth plateau accurately reflects a shift in metabolic bottlenecks.

To map the high-dimensional solution space for metabolic engineering, we evaluated two-sided product envelopes by applying explicit growth floors. For a native growth optimum μ^* and fraction α, the framework dynamically solves:

$$q_{\min\,/\,\max}(\alpha) = \min\,/\,\max\{v_{\text{prod}} : v \in \mathcal{F},\ c_{\text{bio}}^{\top} v \geq \alpha\mu^*\}. \tag{13}$$

Applying an inequality slack ($v_{\text{bio}} \geq \max(0, \alpha\mu^* - 10^{-8})$) rigorously avoids false infeasibility near the theoretical optimum, defining a robust range of feasible metabolic states. Unconstrained theoretical ceilings for key metabolites are summarized in Table 4.

Crucially, this envelope analysis reveals strict, condition-specific stoichiometric coupling (Figure 6). While the aerobic minimum succinate flux remains zero throughout the sampled grid, the strict anaerobic constraint forces a mandatory minimum succinate secretion at positive growth floors, reaching 0.054 at the native optimum. The system's ability to seamlessly isolate these mandatory flux couplings provides a precise mathematical foundation for rational strain design and condition-specific biological hypotheses.

Algorithm 2 Greedy task-preserving counterexample repair with fresh final audit

Require: Model $\mathcal{M}$, probes P, tasks K, edit costs c, protected reactions Q, edit budget J
Ensure: Candidate model $\mathcal{M}'$ and a comprehensive validation report

$\mathcal{M}' \leftarrow \mathrm{copy}(\mathcal{M})$; initialize logs and pass cache
$Q \leftarrow Q \cup \{\text{boundaries, closures, and task objectives}\}$
if any initial task does not pass on $\mathcal{M}'$ **then**
    $status \leftarrow$ initial task failure
else
    for $j = 0, \ldots, J$ **do**
        Audit every $p \in P$ on $\mathcal{M}'$; inherit only mathematically admissible cached passes
        if any result is neither pass nor fail **then**
            $status \leftarrow$ incomplete audit; **break**
        end if
        if all probes pass **then**
            $status \leftarrow$ provisional scoped success; **break**
        end if
        if $j = J$ **then**
            $status \leftarrow$ budget exhausted; **break**
        end if
        Extract parsimonious witnesses for failing probes using (**??**)
        $D \leftarrow$ union of their signed supports, excluding reactions in Q
        Sort D by edit cost and identifier; $accepted \leftarrow \mathrm{false}$
        for each $(r, \sigma) \in D$ in this order **do**
            Form the directional bound restriction in (**??**)
            if new bounds are invalid or unchanged **then**
                continue
            end if
            Apply the trial restriction; recompute all declared tasks
            if every task passes **then**
                Retain cut; $accepted \leftarrow \mathrm{true}$; **break**
            else
                Restore the pre-trial bounds
            end if
        end for
        if not $accepted$ **then**
            $status \leftarrow$ no task-preserving single cut; **break**
        end if
    end for
end if
Create a fresh verifier with caching strictly disabled
Recompute all final probes and tasks on $\mathcal{M}'$ from scratch
if $status$ is provisional scoped success **then**
    Set $status$ to scoped success if all final checks pass, else final verification failure
end if
return $\mathcal{M}'$, status, witnesses, attempted cuts and explicit solve counts

Table 4: **Theoretical product-flux ceilings derived from the GemAgents reconstruction.** Values (mmol gDW^{-1} h^{-1}) represent optimal production capacities under distinct aeration constraints.

Product exchange	Aerobic	Anaerobic
Succinate	17.137	17.112
Acetate	29.376	27.713
Ethanol	20.000	20.000

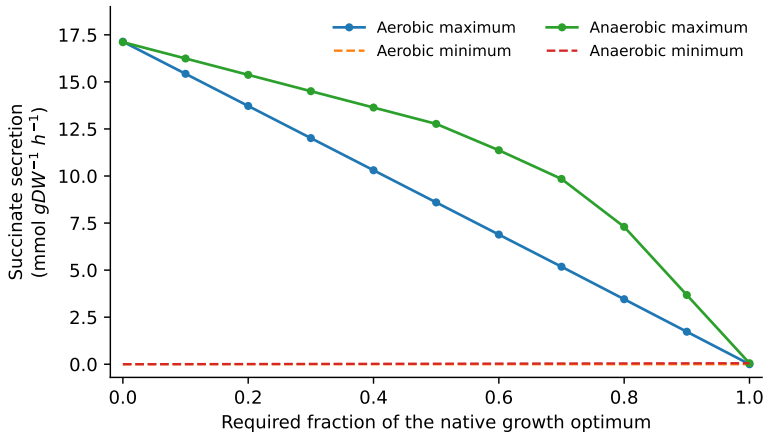

Figure 6: **Production envelopes under enforced growth floors.** Independent LP replay of the supplied candidate model. Solid curves indicate maximum possible succinate secretion, and dashed curves indicate the mandatory minimum flux under each specific growth floor.