Personalizable Metabolic Models to Enable Immune Characterization and Drug Discovery

by

Josh Loecker

A DISSERTATION

Presented to the Faculty of

The Department of Biochemistry at the University of Nebraska

In Partial Fulfillment of Requirements

For the Degree of Doctor of Philosophy

Major: Biochemistry

(Bioinformatics)

Under the Supervision of Professors Tomáš Helikar and Lindsey Crawford

Lincoln, Nebraska

July, 2026

Personalizable Metabolic Models to Enable Immune Characterization and Drug Discovery

Josh Loecker, Ph.D.

University of Nebraska, 2026

Advisors: Tomáš Helikar and Lindsey Crawford

Cellular metabolism is reshaped by disease and physiological state, and genome-scale metabolic networks (GSMNs) offer a systematic way to interrogate this biology and identify therapeutic targets. Building a GSMN from expression data, however, requires a sequence of stages including read processing, normalization and gene-activity classification, multi-omic integration, model reconstruction, simulation, and biological interpretation. These are typically handled by disconnected tools spanning multiple programming languages, file formats, and databases. The resulting reliance on heterogeneous interfaces and manual steps threatens reproducibility. This work develops an ecosystem of computational tools that carry an analysis from raw sequencing reads to mechanistic biological interpretation. First, AutoRNAseq, a Snakemake-based bulk RNA-seq pipeline, alleviates manual, error-prone alignment pipelines by unifying data acquisition, reference genome preparation, quality control, alignment, and quantification. It was validated against a widely used reference pipeline (Pearson correlation > 98%). COMO consumes these outputs and integrates multi-omic processing, context-specific reconstruction, and drug-repurposing analysis in one workflow, only requiring parameter configuration. COMO was used to reconstruct B-cell metabolism in two disease contexts, and nominated ranked metabolic drug targets, several supported by clinical use or

independent literature. A Python reimplementation of the R-based zFPKM gene-activity normalization method reproduced the original classifications (with Pearson and Spearman correlations both 1.00). Utilizing COMO at scale, a study of immune aging reconstructed 5,705 donor- and cell-type-specific models from a single-cell atlas of two million cells across 166 donors (aged 25-85), finding that age-associated metabolic signal concentrates in a small set of pathways and cell types, and shifts in a cell-type-dependent direction rather than uniformly. Lastly, MechAInistic, a tool-grounded, Architect-Reviewer large-language-model system, translates natural-language questions into executable, GSMN-oriented workflows that produce literature-supported hypotheses, outperforming general-purpose frontier models in grounding and task completion; MechAInistic provides an easy-to-use web interface and generates a literature-backed report, eliminating the need to program analysis pipelines. Together, these projects provides a reproducible, end-to-end ecosystem for expression-driven metabolic modeling, with each project solving a specific problem while reducing, or eliminating, the need for researchers to write code.

Dedication

I would like to express my deepest gratitude to my wife, Skylar Loecker, for standing by my side during the creation of this work. I know, for an absolute fact, I would not be where I am today without her support.

Acknowledgements

I would like to thank my advisor, Tomáš Helikar, for his mentorship, guidance, and support throughout my doctoral work. His insight shaped both this research and my growth as a scientist, and I am grateful for the opportunity to have trained in his lab.

I would also like to thank the members of my committee, Lindsey Crawford, Xinghui Sun, Massimiliano Pierobon, and Toshihiro Obata, for their time, feedback, and guidance throughout this process.

Last, but most certainly not least, I am also grateful to the current and former members of the Helikar lab for their help, collaboration, and camaraderie, including Bhanwar Lal Puniya, Robert Moore II, Lauren Mayo, Ahmed Abdeen Hamed, Prakash Packrisamy, Dennis Startsev, Narayna Puraja, William Bryant, Shaifali, Rada Amin Ali, Sara Aghamiri, and any others who have helped me along the way.

Index

PERSONALIZABLE METABOLIC MODELS TO ENABLE IMMUNE CHARACTERIZATION AND DRUG DISCOVERY ..... **II**

DEDICATION ..... **IV**

ACKNOWLEDGEMENTS ..... **IV**

INDEX ..... **V**

LIST OF MULTIMEDIA OBJECTS ..... **IX**

CHAPTER I: INTRODUCTION **11**

REFERENCES 19

CHAPTER II: AUTOMATED BULK RNA-SEQ ALIGNMENT ACHIEVES NEAR-IDENTICAL GENE QUANTIFICATION TO NF-CORE/RNASEQ **24**

CONTRIBUTIONS 24

ABSTRACT 24

Summary *24*

Availability and Implementation *25*

INTRODUCTION 25

IMPLEMENTATION DETAILS 26

Pipeline Behavior *28*

COMPARISON TO OTHER TOOLS 31

RESULTS & USE CASE 32

CONCLUSION 33

FUNDING STATEMENT 34

REFERENCES 35

CHAPTER III: MULTI-OMICS METABOLIC MODELING OF NAIVE B CELLS IDENTIFIES REPURPOSABLE DRUG TARGETS FOR RHEUMATOID ARTHRITIS AND SYSTEMIC LUPUS ERYTHEMATOSUS **37**

CONTRIBUTIONS 37

ABSTRACT 38

INTRODUCTION 39

METHODS 41

Pipeline Design *41*

File Preparation and Omics Data Analysis *44*

RESULTS 48

Constructing Metabolic Models *48*

Metabolic Model of Naive B Cell *51*

Disease Gene Analysis *58*

Drug Target Analysis *58*

DISCUSSION 61

FUNDING 67

DATA AVAILABILITY 67

REFERENCES 67

CHAPTER IV: A PYTHON REIMPLEMENTATION OF R'S ZFPKM TRANSFORMATION **73**

CONTRIBUTION 73

PREFACE & ABSTRACT 74

BACKGROUND 74

zFPKM *74*

Conceptual Basis *76*

Importance *77*

MOTIVATION 78

METHODS AND IMPLEMENTATION 78

Algorithm *78*

Implementation Details *80*

RESULTS 85

LIMITATIONS 86

REFERENCES 86

CHAPTER V: GENOME-SCALE METABOLIC NETWORKS RECONSTRUCTED FROM DONOR-SPECIFIC SCRNA-SEQ ARE INDICATIVE OF SUB-CELLULAR, AGE-RELATED CHANGES 88

CONTRIBUTION 88

INTRODUCTION 88

METHODS 91

Single-cell RNA Sequencing *91*

Genome-scale Metabolic Modeling *92*

Statistical Testing *103*

RESULTS 105

Single-cell RNA Sequencing *105*

Genome-scale Metabolic Modeling *108*

Pathway-specific Age Correlations *113*

Pathway Investigation *117*

CONCLUSION 141

ACKNOWLEDGEMENTS 145

REFERENCES 145

CHAPTER VI: REVIEWER-SUPERVISED, MULTI-AGENT LLM ORCHESTRATION GENERATES MODEL-GROUNDED, DRUG-TARGET HYPOTHESES FROM PAIRED DISEASE/HEALTHY STATE METABOLIC MODEL 151

CONTRIBUTION 151

ABSTRACT 151

GRAPHICAL ABSTRACT 153

KEYWORDS 153

INTRODUCTION 153

RESULTS 155

Translation of Biological Questions into an Executable, Metabolic-modeling Analysis 155

Therapeutic Hypothesis Generation in Two Disease Models 157

Improved Model Grounding and Task Completion over General-purpose LLM Systems 158

Identification of OGDH as a Target Candidate in Rheumatoid Arthritis Naive B cells 162

Identification of IDH as a Target Candidate in Multiple Sclerosis Th17-cells 163

DISCUSSION 165

METHODS AND SYSTEM DESIGN 169

System Implementation and User Input 169

Agent roles and workflow control 171

LLM Configuration and Prompt Management 177

Tool registry and executable analysis layer 179

Constraint-based modeling backend 181

Metabolic Model Reconstruction and Use-case Setup 182

Comparator LLM Evaluation 183

Nine-axis Evaluation Rubric 184

Manual validation of MechAInistic outputs 186

External services and reproducibility 187

Acknowledgements 187
Author Contributions 188
Declaration of Interests 188
Declaration of Generative AI and AI-assisted Technology 188
References 188
CHAPTER VII: CONCLUSION 194

List of Multimedia Objects

Table 1: Available features in the COMO pipeline 46
Table 2: Model validation using specific behaviors. 52
Table 3: Drug targets identified for each context presented in COMO 59
Table 4: The number of validations across each cell type and the total count of passing and failing validations. 112
Table 5: Therapeutic recommendations issued by each system for the two use cases presented in MechAInistic 161
Table 6: Constraint-based modeling software and analysis configuration 181

Figure 1: Comparison between AutoRNAseq and nf-core/rnaseq. Panel 31
Figure 2: Flow chart of the COMO pipeline. GSMN refers to genome-scale metabolic networks. 42
Figure 3: Combined Z-transform scores and their effect on model building and performance. 49
Figure 4: Flux distribution through glycolysis, TCA cycle, and glutaminolysis pathways 56
Figure 5: Validation of B cell model. 57
Figure 6: Biological processes and pathways enriched in common drug targets between RA and SLE. . 61
Figure 7: FPKM versus zFPKM expression profile. 76

Figure 8: Heatmap displaying z-scored cell counts across various cell types (x-axis) for individual donors (y-axis). 108

Figure 9: Heatmap showing z-scored reaction counts across immune cell types for individual donors. 111

Figure 10: The number of average reactions across each cell type for all donors. 111

Figure 11: The number of cell types that have significant correlations for each pathway. 117

Figure 12: The number of pathways that have significant correlations for each cell type 117

Figure 13: Rank-Rank plot for Tetrahydrobiopterin Metabolism for cell types where it was indicated as significant ($p < 0.05$). 123

Figure 14: Rank-rank Kendall correlation coefficient plots for phenylalanine metabolism in cell types it was indicated as significant ($p < 0.05$). 126

Figure 15: Rank-rank Kendall correlation coefficient plots for vitamin A metabolism in cell types where it was indicated as significant ($p < 0.05$). 131

Figure 16: Rank-rank Kendall correlation coefficient plots for Vitamin B2 metabolism in cell types where it was indicated as significant. 134

Figure 17: Rank-rank Kendall correlation coefficient plots for hyaluronan metabolism in cell types where it was indicated as significant. 140

Figure 18: MechAInistic graphical abstract 153

Figure 19: The adjustable LLM settings and model upload section of MechAInistic 170

Figure 20: Overview of the MechAInistic system architecture. 177

Chapter I: Introduction

Cellular metabolism consists of reactions used to obtain nutrients from the environment, convert them into energy and biosynthetic precursors, and export the resulting byproducts[1]. Because these reactions are linked through shared metabolites, metabolism can also be described by the rate at which material moves through each reaction, an amount labeled "flux". A reaction's flux is related to the metabolic requirements of a cell; for example, a resting and active lymphocyte may contain the same set of reactions capable of uptaking a variety of metabolites, but differ in which of those reactions carry flux[2]. Following this logic, disease often reshapes metabolism by altering how a cell distributes flux across its available reactions[3], modifying the environmental nutrient availability[4], or preventing reactions from being expressed[5]. Each of these has a direct representation in a metabolic network, making them observable in computational predictions. Identifying pathways and specific enzymes responsible for these shifts is a key step toward identifying a therapeutic target and intervention capable of restoring a healthy metabolic state[3,6]. Identification of therapeutic targets, however, can be difficult due to the combinatorial number of potential gene and reaction targets, the interconnected nature of metabolic pathways, and the cost and time required to experimentally validate candidates[1,3,7]. Testing each candidate individually, whether by knockdown, knockout, or pharmacological inhibition, scales poorly with the number of reactions in most pathways, and is increasingly infeasible for a cell's full metabolic network.

Genome-scale metabolic networks (GSMNs) address this by computationally representing a cell's metabolism, allowing perturbations to be simulated and their impacts predicted before traditional experimental testing[8]. A GSMN encodes each reaction as a mass- and charge-balanced transformation of metabolites localized to a subcellular compartment (such as the cytosol), with explicit transport reactions that move metabolites between compartments and across the plasma membrane. Reactions are linked to the genome via gene-protein-reaction (GPR) rules, where Boolean logic utilizes AND relationships to connect enzymatic subunits required for a reaction to be catalyzed, and OR relationships to represent independent isozymes capable of catalyzing the same reaction. GPRs, reaction direction, and flux boundaries are joined to define a stoichiometrically consistent representation of an organism's metabolism, known as a reference model. By identifying active genes from -omics data, a specific organ- or cell-type "context" can be reconstructed by removing reactions whose GPR rules are unsatisfied in that context.

Reconstructing a context-specific GSMN requires knowledge about which reactions should be removed from the reference model. This is accomplished by normalizing gene expression and defining a threshold value for downstream algorithms to utilize in reaction removal. Threshold calculation can be accomplished by a variety of methods grouped into various categories, including fixed global cutoffs, rank/quantile-based, density-based, and background-based expression. While detailing each is beyond the scope of this work, additional literature can be found via Richelle et al.[2] and Opdam et al.[7] Reconstruction algorithms use the calculated thresholds to remove reactions from the

reference model. Common algorithms include iMAT[9], GIMME[10], INIT[5]/tINIT[6]/ftINIT[11], mCADRE[12], CORDA[13], and FASTCORE[14]/FASTCORMICS[3]. Threshold-free methods do exist, such as E-Flux[15], but they provide flux values rather than a reconstructed metabolic model.

Following reconstruction, in-silico simulations can be performed to predict the impact of an experimental procedure, including a gene knockout or enzyme inhibition, allowing targets and therapies to be ranked from highest to lowest priority. Importantly, a GSMN contains no rate constants, metabolite concentrations, or regulatory interactions, and predictions assume that intracellular metabolite pools are at a steady state. This imposes a set of limitations on the simulation results, which are indicative of stoichiometrically feasible flux distributions rather than metabolic dynamics[1].

Most reconstruction pipelines require a sequence of stages that, in many environments, are handled by disconnected tools[16]. Raw sequencing reads must be processed into a gene count matrix[17]; those counts must be normalized and classified into activity states, as previously described[2]; a context-specific model must be extracted from the reference model[7,18]; the model must be simulated and perturbed[1], and the results connected to supporting literature. These stages typically span multiple interfaces, including command-line tools for sequence alignment, R is commonly used for high-throughput data, and Python or MATLAB for reconstruction[19,20]. Together, a series of heterogeneous APIs, file formats, external databases, and error-prone manual steps can threaten the reproducibility of these workflows[21]. The work in this dissertation develops a set of

computational tools that span these stages and are built to work together. This enables a study to proceed from raw reads to a mechanistic biological interpretation within a single ecosystem. The chapters, as written, depend on one another in a sequence: a data-processing pipeline produces the input that the modeling and reconstruction pipelines consume directly, and the reimplementation of a normalization technique reduces the complexity of its usage. Reconstruction is then applied at a large scale to investigate immune aging, and the number of reconstructed metabolic models motivates a final, language-model-driven system for interpreting the metabolic models.

Workflow managers such as Snakemake[21] and Nextflow[22] mitigate this by combining file-based tracking with explicitly declared dependencies, but still demand considerable manual preparation. The widely used nf-core/rnaseq[23] pipeline, for example, requires users to supply properly formatted reference genome and annotation files, and relies on an independent pipeline for RNA sequencing retrieval, ultimately meaning complete analysis requires coordinating multiple pipelines and commands, which introduces a risk of error. Most such pipelines are, moreover, optimized for differential-expression reporting rather than for entry into systems-level modeling. Chapter II presents AutoRNAseq, a Snakemake-based bulk RNA-seq alignment pipeline that minimizes user intervention by integrating data acquisition from NCBI GEO, automatic preparation of Ensembl reference genomes, quality control, alignment, and quantification in a unified workflow. It writes outputs directly into the directory structure expected by COMO, the downstream modeling and reconstruction pipeline, which eliminates additional sources of error researchers may encounter. This work was validated against nf-core/rnaseq, and

AutoRNAseq produces gene counts in close agreement (Pearson correlations exceeded 98%).

Once -omics data have been processed, the next obstacle is using them to reconstruct a context-specific metabolic model and to use that model for downstream analysis and hypothesis generation. As the volume of disease-specific omics has grown, so too has the need to consolidate these data into a coherent workflow[24]. Chapter III presents COMO (Constraint-based Optimization of Metabolic Objectives), a pipeline that integrates multi-omic data processing with context-specific model reconstruction and drug-repurposing analysis. COMO consumes the processed expression matrices generated by AutoRNAseq and, as explained, classifies these measurements into gene activation states required by reconstruction algorithms. While COMO works best when used alongside AutoRNAseq, using AutoRNAseq is not a requirement. COMO was tested and validated by reconstructing naive B cell models in healthy, rheumatoid arthritis[25], and systemic lupus erythematosus[26] contexts. These models were validated against known behavior and used to nominate and rank metabolic drugs for both diseases, several of which correspond to drugs already in clinical use or supported by independent literature.

The reconstruction algorithms used to extract a context-specific model require each gene to be classified as active or inactive, and zFPKM is one established basis for that classification[27]. In the original zFPKM publication, the zFPKM algorithm was validated against open/closed chromatin states from the ENCODE[28] dataset and showed that low-expression genes were associated with repressed chromatin. zFPKM transforms each

gene's abundance estimates (specifically FPKM or TPM) as a standardized distance from the distribution of actively transcribed genes. This means a single, fixed threshold can consistently separate active from inactive genes across datasets. Prior to this work, the only existing zFPKM implementation was in an R package, and relying on it from within a Python pipeline, such as COMO, complicated the development process. Chapter IV describes a Python reimplementation that reproduces the R package's methodology, removing COMO's reliance on R. The reimplementation matches the R package's output exactly, with both Pearson and Spearman correlations equal to 1.00 and the same genes marked as active or inactive. This means the Python implementation can be used to obtain the exact outputs produced by the R package.

With an RNA-sequence-to-modeling pipeline constructed, the same components can be applied to a biological question at a scale that would otherwise be impractical due to manual data coordination and code execution. An aging immune system exhibits chronic, low-grade inflammation, termed "inflammaging"[29]. Immune cells follow distinct, cell-type-specific trajectories that are not uniformly observed across the immune system[30]. Previous studies have used single-cell RNA-seq data to capture these differences, but transcript abundance reports the expression state of metabolic genes rather than the downstream enzymatic flux[30–33]. A reduced expression of a gene does not necessarily correlate to reduced protein, enzymatic activity, or flux, because a variety of post-transcriptional and post-translational modifications influence gene activity. GSMNs partially address this by evaluating expression in the context of network stoichiometry, so that pathway-level feasibility rather than individual transcript levels determines the

predicted reaction flux. Prior work on age-related changes in immune cells has focused on cell-type proportions and transcriptomic expression, not on donor- and subtype-specific GSMNs at this scale to investigate metabolic capacity changes with age. Chapter V uses COMO to reconstruct 5,705 personalized metabolic models from a single-cell atlas of approximately 2 million cells spanning 166 donors aged 25 to 85[30], estimates feasible flux by sampling, and tests metabolic pathways for correlation with donor age. Each model represents the metabolic capacity of one immune subtype in one donor, which makes it possible to investigate if a given pathway's flux correlates with age in some subtypes but not others. The analysis finds that the age-associated signal is concentrated in a small number of pathways and a subset of cell types, and that the direction of the correlation between flux and age (increasing or decreasing) is cell-type-dependent rather than a coordinated, system-wide shift. The number of models this study produces is far more than can be individually interpreted by hand, which directly motivates the final chapter.

Interpreting an individual GSMN can be laborious, requiring tracing which reactions can carry flux and verifying their support in the literature[34]. Attempting to accomplish this at the scale of presented in Chapter V, with 5,705 GSMNs, this interpretation becomes the limiting factor rather than the reconstruction of the model. The underlying barrier, ultimately, is that constraint-based modeling requires both computational and biological expertise, as well as multi-step workflows that combine model setup, simulation, perturbation analysis, and interpretation. Large language models offer a route to such analyses, but their workflows can fail in scenarios where they are not anchored in

executable tools, mechanistic constraints, or verifiable intermediate results, and can produce fluent yet unsupported biological claims. Chapter VI presents MechAInistic, a reviewer, tool-based, system organized around an “Architect-Reviewer” concept. By translating natural-language questions about paired disease and healthy models into executable, GSMN-oriented workflows, MechAInistic can produce auditable, literature-supported reports. The system runs on models reconstructed by COMO from RNAseq-processed data, exercising the full ecosystem developed in the earlier chapters, and addresses the interpretive bottleneck identified in the immune-aging study. In two immune-cell use cases, MechAInistic generated hypotheses traced to specific metabolic network features and demonstrated stronger, model-based grounding and task completion than general-purpose, frontier language models.

Taken together these chapters describe an ecosystem of interconnected tools. AutoRNAseq alleviates data-acquisition and alignment, and writes its output into a structure expected by COMO. COMO classifies the gene counts, reconstructs context-specific models, and supports downstream perturbation analysis. The Python zFPKM implementation removes COMO’s dependency on R, which simplifies the pipeline for researchers to set up and use. When applied to immune aging, COMO demonstrated that it is capable of operating at a scale that manual reconstruction cannot support, and revealing that age-associated metabolic shifts are concentrated in specific pathways and cell types, rather than evenly distributed throughout metabolism. The number of models reconstructed exposes an interpretation as the final bottleneck in this pipeline, which MechAInistic addresses by grounding language-model reasoning in an executable,

constraint-based workflow. The result is a path from raw sequencing reads into an audible, mechanistically supported hypothesis within a cohesive pipeline, lowering the technical arrier to using genome-scale metabolic models for therapeutic target identification.

References

1. Orth, J. D., Thiele, I. & Palsson, B. Ø. What is flux balance analysis? Nat. Biotechnol. 28, 245–248 (2010).
2. Richelle, A., Joshi, C. & Lewis, N. E. Assessing key decisions for transcriptomic data integration in biochemical networks. PLoS Comput. Biol. 15, e1007185 (2019).
3. Pacheco, M. P. et al. Identifying and targeting cancer-specific metabolism with network-based drug target prediction. EBioMedicine 43, 98–106 (2019).
4. Sullivan, M. R. et al. Quantification of microenvironmental metabolites in murine cancers reveals determinants of tumor nutrient availability. eLife 8, e44235 (2019).
5. Agren, R. et al. Reconstruction of genome-scale active metabolic networks for 69 human cell types and 16 cancer types using INIT. PLoS Comput. Biol. 8, e1002518 (2012).
6. Agren, R. et al. Identification of anticancer drugs for hepatocellular carcinoma through personalized genome-scale metabolic modeling. Mol. Syst. Biol. 10, 721 (2014).

7. Opdam, S. et al. A Systematic Evaluation of Methods for Tailoring Genome-Scale Metabolic Models. Cell Syst. 4, 318-329.e6 (2017).
8. Bordbar, A. & Palsson, B. O. Using the reconstructed genome-scale human metabolic network to study physiology and pathology. J. Intern. Med. 271, 131–141 (2012).
9. Zur, H., Ruppin, E. & Shlomi, T. iMAT: an integrative metabolic analysis tool. Bioinformatics 26, 3140–3142 (2010).
10. Becker, S. A. & Palsson, B. O. Context-Specific Metabolic Networks Are Consistent with Experiments. PLoS Comput. Biol. 4, e1000082 (2008).
11. Gustafsson, J. et al. Generation and analysis of context-specific genome-scale metabolic models derived from single-cell RNA-Seq data. Proc. Natl. Acad. Sci. U. S. A. 120, e2217868120.
12. Wang, Y., Eddy, J. A. & Price, N. D. Reconstruction of genome-scale metabolic models for 126 human tissues using mCADRE. BMC Syst. Biol. 6, 153 (2012).
13. Schultz, A. & Qutub, A. A. Reconstruction of Tissue-Specific Metabolic Networks Using CORDA. PLoS Comput. Biol. 12, e1004808 (2016).
14. Vlassis, N., Pacheco, M. P. & Sauter, T. Fast reconstruction of compact context-specific metabolic network models. PLoS Comput. Biol. 10, e1003424 (2014).
15. Colijn, C. et al. Interpreting expression data with metabolic flux models: predicting Mycobacterium tuberculosis mycolic acid production. PLoS Comput. Biol. 5, e1000489 (2009).

16. Vieira, V., Ferreira, J. & Rocha, M. A pipeline for the reconstruction and evaluation of context-specific human metabolic models at a large-scale. PLoS Comput. Biol. 18, e1009294 (2022).
17. Liao, Y., Smyth, G. K. & Shi, W. The R package Rsubread is easier, faster, cheaper and better for alignment and quantification of RNA sequencing reads. Nucleic Acids Res. 47, e47 (2019).
18. Robaina Estévez, S. & Nikoloski, Z. Generalized framework for context-specific metabolic model extraction methods. Front. Plant Sci. 5, 491 (2014).
19. Ebrahim, A., Lerman, J. A., Palsson, B. O. & Hyduke, D. R. COBRApy: COnstraints-Based Reconstruction and Analysis for Python. BMC Syst. Biol. 7, 74 (2013).
20. Heirendt, L. et al. Creation and analysis of biochemical constraint-based models using the COBRA Toolbox v.3.0. Nat. Protoc. 14, 639–702 (2019).
21. Mölder, F. et al. Sustainable data analysis with Snakemake. F1000Research 10, 33 (2021).
22. Di Tommaso, P. et al. Nextflow enables reproducible computational workflows. Nat. Biotechnol. 35, 316–319 (2017).
23. Patel, H. et al. nf-core/rnaseq: nf-core/rnaseq v3.26.0 - Chromium Cuttlefish. https://doi.org/10.5281/zenodo.20072251 (2026) doi:10.5281/zenodo.20072251.
24. Ewald, J. D. et al. Web-based multi-omics integration using the Analyst software suite. Nat. Protoc. 19, 1467–1497 (2024).

25. de Vries, C. et al. Rheumatoid Arthritis Related B-Cell Changes Are Found Already in the Risk-RA Phase. Eur. J. Immunol. 55, e202451391 (2025).
26. Canny, S. P. & Jackson, S. W. B cells in systemic lupus erythematosus: from disease mechanisms to targeted therapies. Rheum. Dis. Clin. North Am. 47, 395–413 (2021).
27. Hart, T., Komori, H. K., LaMere, S., Podshivalova, K. & Salomon, D. R. Finding the active genes in deep RNA-seq gene expression studies. BMC Genomics 14, 778 (2013).
28. ENCODE Project Consortium. An integrated encyclopedia of DNA elements in the human genome. Nature 489, 57–74 (2012).
29. Franceschi, C., Garagnani, P., Parini, P., Giuliani, C. & Santoro, A. Inflammaging: a new immune-metabolic viewpoint for age-related diseases. Nat. Rev. Endocrinol. 14, 576–590 (2018).
30. Terekhova, M. et al. Single-cell atlas of healthy human blood unveils age-related loss of NKG2C+GZMB-CD8+ memory T cells and accumulation of type 2 memory T cells. Immunity 56, 2836-2854.e9 (2023).
31. Filippov, I., Schauser, L. & Peterson, P. An integrated single-cell atlas of blood immune cells in aging. Npj Aging 10, 59 (2024).
32. Sopena-Rios, M. et al. Single-cell analysis of the human immune system reveals sex-specific dynamics of immunosenescence. Nat. Aging 6, 932–949 (2026).

33. Wang, Y. et al. Integrating single-cell RNA and T cell/B cell receptor sequencing with mass cytometry reveals dynamic trajectories of human peripheral immune cells from birth to old age. Nat. Immunol. 26, 308–322 (2025).

34. Thiele, I. & Palsson, B. Ø. A protocol for generating a high-quality genome-scale metabolic reconstruction. Nat. Protoc. 5, 93–121 (2010).

Chapter II: Automated bulk RNA-seq alignment achieves near-identical gene quantification to nf-core/rnaseq

Josh Loecker, Brandt Bessell, Bhanwar Lal Puniya, Tomáš Helikar

Contributions

I conceived and built the AutoRNAseq pipeline, designed and carried out all statistical validation against the "nf-core/rnaseq" pipeline, and wrote the manuscript in its entirety. B. Bessell implemented several additional workflow steps. I subsequently rewrote the pipeline to improve robustness and eliminate Snakemake "checkpoint" rules so the full workflow outputs are resolved before execution, rather than during runtime. This rewrite touched most of the workflow, including portions of Bessell's contribution.

This chapter is available as a preprint in the bioRxiv database at https://doi.org/10.64898/2026.04.21.719844

Abstract

Summary

Improved accessibility of high-throughput RNA sequencing has increased the amount of data generated each year. This increase in data creates a need for reproducible pipelines that can process RNA-seq data consistently across experiments. AutoRNAseq addresses

this need by providing a Snakemake-based workflow for bulk RNA-seq analysis by automating data retrieval, quality control, and gene quantification. Unlike existing RNA-seq workflows that require users to coordinate multiple pipelines and pre-configure reference data, AutoRNAseq provides a single, end-to-end workflow that automates data acquisition, reference preparation, quality control, alignment, and quantification with minimal user intervention. AutoRNAseq is applicable to any domain requiring consistently processed RNA-seq datasets, including bioinformatics, computational biology, and drug-response studies.

Availability and Implementation

AutoRNAseq is implemented in Snakemake and available at https://gitlab.com/unebraska/lagbh-public/autornaseq. Documentation and example configuration files are provided in the GitLab README file and Chapter II's Supplementary Information. The code to reproduce the statistics presented here is in the GitLab repository under the "publication" folder.

Introduction

Genomics has experienced a significant expansion in sequencing capacity over the past decade.[1] Advances in next-generation sequencing technology have dramatically reduced per-sample costs, enabling researchers to generate increasingly large and complex RNA-seq datasets. This ease of sequencing has accelerated biological discovery but has also created a "reproducibility crisis", where subtle differences in software versions or

parameters can substantially affect results.[2] Workflow managers, such as Snakemake[3], address this by combining Python's flexibility with file-based dependency tracking, allowing explicit declaration of software dependencies through containers or Conda environments.

Despite these tools, many RNA-seq pipelines require substantial infrastructure setup and manual preparation. For example, the "nf-core/rnaseq" pipeline requires users to provide pre-formatted genome FASTA and GTF files, which can introduce errors if generated incorrectly.[4,5] Here, we present AutoRNAseq, a Snakemake-based bulk RNA-seq pipeline that significantly minimizes user intervention while maximizing reproducibility. AutoRNAseq directly fetches data from NCBI GEO accessions or reads from local repositories, prepares reference genomes from Ensembl, and produces gene count matrices for downstream analysis, such as differential expression analysis.

Implementation Details

AutoRNAseq is organized into the following five stages:

1) Workflow setup and reference genome preparation: AutoRNAseq parses the provided configuration file (discussed in the Pipeline Behavior section below) and writes metadata files describing sample layout and library preparation. Additionally, the reference FASTA and annotation (GTF) files for the specified release will be downloaded, and their related indices will be built using STAR. Following this, the corresponding transcriptome FASTA file is generated.

2) Data downloading: if the user chooses to use publicly available data, the "prefetch" command-line tool will be used to download SRR accession information from NCBI's Gene Expression Omnibus (GEO).
3) Data preprocessing and quality control: AutoRNAseq will, optionally, perform read trimming with Trim Galore!.[6] Quality control processing with FastQC is performed on the raw and, if available, trimmed, FASTQ files.[7] Finally, this section of the pipeline will scan for possible contamination using the FASTQ Screen command-line tool against the following genomes: adapter sequences, Arabidopsis, Drosophila, E. coli, human, lambda phages, mitochondria, mouse, PhiX phages, rat, vectors, worm, yeast, and rRNA.[8]
4) Alignment and quantification: Reads are aligned to the genome using STAR, producing a coordinate-sorted BAM file for downstream processing. Salmon then estimates transcript- and gene-level abundances.[9,10] Although lightweight transcriptome-mapping tools (e.g., Salmon and Kallisto) offer faster quantification, STAR was selected because it supports alignment-level quality control, visualization, and optional downstream analysis.
5) Metric calculation and report generation: AutoRNAseq computes RNA-seq metrics and insert sizes using Picard. A comprehensive quality control report is generated with MultiQC.[11,12] Additionally, fragment sizes, insert sizes, RNA-seq metrics, and gene count quantifications are copied into a directory structure directly readable by COMO, a Jupyter Notebook-based tool for automated constraint-based metabolic modeling, discussed later.[13]

Pipeline Behavior

AutoRNAseq provides two methods for data input, supporting both public datasets and local sequencing files. Both approaches require a metadata file containing sample information, including columns for GEO SRR accession numbers, sample names, end type (single- or paired-end), and sequencing method (total RNA or polyA). First, if SRR accession numbers are used, the pipeline will automatically download and process each dataset.[14] This approach helps combine multiple studies for additional downstream analysis. The second input method processes local FASTQ files. This approach suits individuals who have their own set of sequencing data to analyze. In this method, the first column of the metadata file should be left blank, and the directory path to the FASTQ files should be specified in the configuration file under the “LOCAL_FASTQ_FILES" key. The FASTQ filenames should be prefixed with the respective “sample” identifier, enabling automatic association between files and sample metadata.

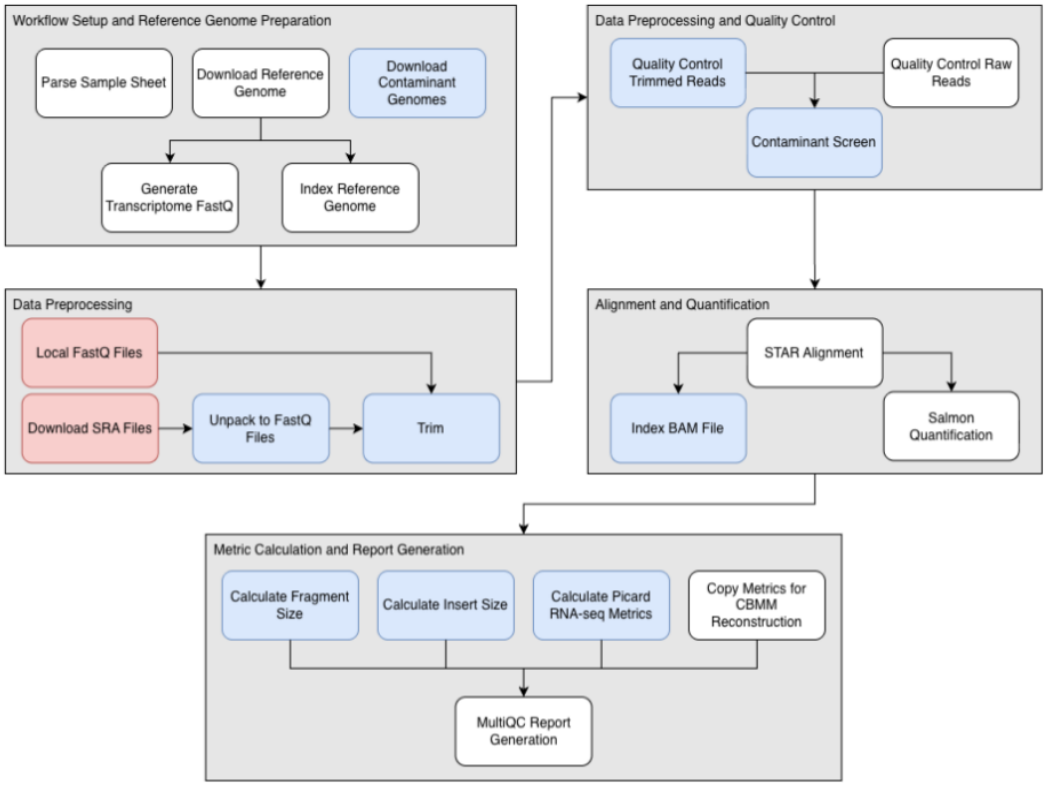
The pipeline needs a configuration file that controls its behavior. Key parameters include the path to the metadata CSV file and output directories. More granular control is also provided over which processing steps are executed: adapter trimming, contaminant screening, RNA-seq metrics calculation, insert size collection, and fragment size collection. The reference genome configuration requires specifying the taxonomy ID and Ensembl release version, which allows AutoRNAseq to automatically download the required reference genome and pre-computed alignment indices from Ensembl.[15] The outputs of this include the genome FASTA, annotation GTF, STAR’s genome index,

FASTA transcriptome (cDNA), and the Salmon index. Lastly, benchmarks are automatically calculated to track execution time, CPU usage, and memory usage for each rule.

Once users configure the above, executing the pipeline is straightforward. Two profiles are provided by default, for local or SLURM workflow execution. For local execution, use the command “snakemake --profile profiles/local”. Alternatively, if running the workflow on a computing cluster, use the command “snakemake --profile profiles/cluster”. This will automatically populate Snakemake with the required job names, account details, and resources to submit jobs to the cluster. Jobs are submitted to the cluster based on its available resources. Because of this, Snakemake does not submit a single job to the cluster requesting the sum of each rule’s runtime, memory, and CPU requirements, as this would unnecessarily inflate resource usage. Instead, Snakemake submits a single job per rule, using the resource limits we have carefully tweaked to increase the job submission rate and decrease the total runtime.

The figure below shows the information flow within AutoRNAseq. At a minimum, at least one red input must be provided (local FASTQ files or SRR accession values). After this, any blue step can be skipped by setting its corresponding configuration value to False.

A


Workflow Setup and Reference Genome Preparation
Parse Sample Sheet
Download Reference Genome
Download Contaminant Genomes
Generate Transcriptome FastQ
Index Reference Genome
Data Preprocessing and Quality Control
Quality Control Trimmed Reads
Quality Control Raw Reads
Contaminant Screen
Data Preprocessing
Local FastQ Files
Download SRA Files
Unpack to FastQ Files
Trim
Alignment and Quantification
STAR Alignment
Index BAM File
Salmon Quantification
Metric Calculation and Report Generation
Calculate Fragment Size
Calculate Insert Size
Calculate Picard RNA-seq Metrics
Copy Metrics for CBMM Reconstruction
MultiQC Report Generation

B


Metrics Across All Samples
Pearson Correlation
Spearman Correlation
Kolmogorov Smirnov Statistic
Sample Name

Figure 1: Comparison between AutoRNAseq and nf-core/rnaseq. Panel 1A: AutoRNAseq's information flow. At least one red input is required (local FASTQ files or NCBI GEO SRR accession numbers), and any blue processing step can be skipped by modifying the related setting. Panel 1B: Pearson, Spearman, and Kolmogorov-Smirnov statistics between AutoRNAseq and nf-core/rnaseq.

Comparison to Other Tools

RNA-seq pipelines differ in computational robustness, supporting downstream biological interpretation and integration with multi-omic modeling workflows. The most popular and comprehensive RNA-seq analysis pipeline is "nf-core/rnaseq", which provides extensive tool integration and rigorous testing.[4] However, "nf-core/rnaseq" relies on a separate workflow ("nf-core/fetchngs") for SRA downloads, requiring users to coordinate two distinct pipelines and separate terminal commands for complete data processing.[5] In contrast, AutoRNAseq integrates data acquisition and alignment in a single workflow, streamlining the process for individuals working primarily with GEO/SRA data.

Most RNA-seq pipelines are optimized for differential gene expression analysis and reporting. In contrast, AutoRNAseq was explicitly designed to serve as a bridge between transcriptomic preprocessing and downstream systems-level modeling workflows, such as genome-scale metabolic modeling, rather than as a standalone differential expression pipeline. Namely, AutoRNAseq outputs results in a format that is compatible with the "Constraint-based Optimization of Metabolic Objectives" (COMO) tool, a Jupyter notebook-based pipeline for multi-omic data integration in metabolic modeling and drug discovery.[13] COMO requires processed gene expression matrices as input, which AutoRNAseq generates automatically in its alignment and quantification steps. This

integration allows individuals to transition quickly and easily from bulk RNA-seq analysis to reconstructing context-specific, constraint-based metabolic models. The combination of these tools allows for multi-scale studies that connect transcriptomics, metabolic predictions, and drug target identification.

Results & Use Case

To demonstrate AutoRNAseq capabilities, we applied the pipeline to a use case involving multiple human RNA-seq studies from the Sequencing Quality Control Consortium.[16] This analysis evaluated the pipeline's ability to handle diverse experimental designs, but focused primarily on paired-end, total RNA preparation.

The analysis incorporated thirty-two SRR IDs from the GSE47792 identifier (Chapter II Supplementary Table S1). The samples in this table encode biological context, allowing multiple experiments to be aligned simultaneously. The data are encoded as follows: cell type or tissue, study identifier, and run number. If studies include technical replicates, a lowercase 'r' followed by the replicate number should be appended to the sample name. This structured naming facilitates downstream analysis while maintaining clear file tracking.

The pipeline was executed on a workstation with 24 cores and 256 GB RAM using the command “snakemake --profile profiles/cluster”. We validated pipeline accuracy by comparing gene counts generated by AutoRNAseq with those from the “nf-core/rnaseq” pipeline using the same Ensembl reference genome, release 115. Figure 1B shows the

results of these statistics and Supplementary Table S2 presents the values for graph creation. We compared log-transformed per-gene counts between AutoRNAseq and "nf-core/rnaseq". For each sample, all Pearson correlation coefficients exceed 0.9839 (P-values < 0.05), indicating a strong linear relationship between the two pipelines. Similarly, Spearman correlations are above 0.8770 (P-values < 0.05), indicating that genes are ranked similarly between the pipelines. Finally, the Kolmogorov-Smirnov statistics are all approximately 0.0025 (with only four P-values < 0.05), suggesting that we cannot reject the null hypothesis that gene count distributions from each pipeline are the same. Thus, taken together, AutoRNAseq produces results similar to those of the "nf-core/rnaseq" pipeline.

Conclusion

AutoRNAseq provides a reproducible, accessible solution for bulk RNA-seq alignment that addresses key challenges in modern genomics research. By integrating data acquisition, quality control, alignment, and quantification in a single Snakemake workflow, the pipeline reduces technical barriers while maintaining rigorous reproducibility standards. Dual input modes accommodate both public data meta-analyses and local sequencing projects, while Conda-based dependency management ensures consistent results across computing environments. Our validation results demonstrate that the pipeline outputs closely align with those of the existing "nf-core/rnaseq" tool. AutoRNAseq provides an easy-to-use, reproducible RNA-seq alignment pipeline that

requires minimal configuration. Documentation, example configuration files, and the complete source code are available at the GitLab repository.

While this work successfully addresses the need for automated, highly reproducible RNA-seq preprocessing and quantification, generating a reliable gene count matrix is often only the first step in a broader bioinformatics workflow. To translate these standardized transcriptomic profiles into actionable biological and therapeutic insights, the data must be integrated into systems-level modeling frameworks. As previously noted, AutoRNAseq is explicitly engineered to serve as this bridge by automatically outputting results into a directory structure and format directly compatible with the Constraint-based Optimization of Metabolic Objectives (COMO) tool, discussed in the next chapter. COMO is a comprehensive pipeline designed to ingest these processed gene count matrices (alongside other -omics datasets, such as proteomics) to reconstruct genome-scale metabolic networks. By transitioning the transcriptomic data generated by AutoRNAseq into COMO's modeling and drug perturbation simulations, a unified computational ecosystem can facilitate multi-scale studies that connect raw sequencing data to metabolic predictions and, ultimately, to novel hypotheses for drug target identification.

Funding Statement

This work is supported by funding to TH from the National Institute of Health (R35GM119770), University of Nebraska-Lincoln's "Grand Challenges" Catalyst

Award, and the National Strategic Research Institute, “Medical Countermeasure Drug Discovery and Development Increment 2 and 3” (FA4600-18-9001).

References

1. Chaudhari, J. K., Pant, S., Jha, R., Pathak, R. K. & Singh, D. B. Biological big-data sources, problems of storage, computational issues, and applications: a comprehensive review. Knowl. Inf. Syst. 66, 3159–3209 (2024).

2. Ziemann, M., Poulain, P. & Bora, A. The five pillars of computational reproducibility: bioinformatics and beyond. Brief. Bioinform. 24, bbad375 (2023).

3. Mölder, F. et al. Sustainable data analysis with Snakemake. F1000Research 10, 33 (2021).

4. Patel, H. et al. nf-core/rnaseq: nf-core/rnaseq v3.22.2 - Perfect Palladium Penguin. Zenodo https://doi.org/10.5281/zenodo.17909656 (2025).

5. Patel, H. et al. nf-core/fetchngs: nf-core/fetchngs v1.12.0 - Titanium Platypus. Zenodo https://doi.org/10.5281/zenodo.10728509 (2024).

6. Krueger, F. et al. FelixKrueger/TrimGalore. Zenodo https://doi.org/10.5281/zenodo.7598955 (2023).

7. Babraham Bioinformatics - FastQC A Quality Control tool for High Throughput Sequence Data. https://www.bioinformatics.babraham.ac.uk/projects/fastqc/.

8. Babraham Bioinformatics - FastQ Screen. https://www.bioinformatics.babraham.ac.uk/projects/fastq_screen/.

9. Dobin, A. et al. STAR: ultrafast universal RNA-seq aligner. Bioinforma. Oxf. Engl. 29, 15–21 (2013).

10. Patro, R., Duggal, G., Love, M. I., Irizarry, R. A. & Kingsford, C. Salmon provides fast and bias-aware quantification of transcript expression. Nat. Methods 14, 417–419 (2017).

11. broadinstitute/picard. Broad Institute (2026).

12. Ewels, P., Magnusson, M., Lundin, S. & Käller, M. MultiQC: summarize analysis results for multiple tools and samples in a single report. Bioinformatics 32, 3047–3048 (2016).

13. Bessell, B. et al. COMO: a pipeline for multi-omics data integration in metabolic modeling and drug discovery. Brief. Bioinform. 24, bbad387 (2023).

14. Edgar, R., Domrachev, M. & Lash, A. E. Gene Expression Omnibus: NCBI gene expression and hybridization array data repository. Nucleic Acids Res. 30, 207–210 (2002).

15. Dyer, S. C. et al. Ensembl 2025. Nucleic Acids Res. 53, D948–D957 (2025).

16. Su, Z. et al. A comprehensive assessment of RNA-seq accuracy, reproducibility and information content by the Sequencing Quality Control Consortium. Nat. Biotechnol. 32, 903–914 (2014).

Chapter III: Multi-omics metabolic modeling of naive B cells identifies repurposable drug targets for rheumatoid arthritis and systemic lupus erythematosus

Brandt Bessell*, Josh Loecker*, Zhongyuan Zhao, Sara Sadat Aghamiri, Sabyasachi Mohanty, Rada Amin, Tomáš Helikar, Bhanwar Lal Puniya

*These authors contributed equally

Contributions

Chapter III is adapted from Bessell, Loecker, et al., Briefings in Bioinformatics 2023, on which B. Bessell and I are co-first authors. B. Bessell developed the initial implementation of the COMO pipeline and performed the B cell model construction, validation, and drug target analyses presented here. My contribution was to the pipeline's software architecture and distribution. I restructured the codebase as an installable Python package, removed hardcoded, container-relative file paths so that the pipeline runs on arbitrary filesystems and on HPC systems outside a container, eliminated the pipeline's dependence on rpy2 by reimplementing its R-based analysis steps in Python, and rebuilt the Docker image definition, reducing build time from approximately two hours to ten minutes. These changes are what allow the pipeline to be executed reproducibly and at scale, as required by the analysis in Chapter IV. The concluding paragraph of the Discussion was written for this dissertation and does not appear in the published article.

This chapter is available as a peer-reviewed publication at
https://doi.org/10.1093/bib/bbad387

Abstract

Identifying potential drug targets using metabolic modeling requires integrating multiple modeling methods and heterogeneous biological datasets, which can be challenging without efficient tools. We developed Constraint-based Optimization of Metabolic Objectives (COMO), a user-friendly pipeline that integrates multi-omics data processing, context-specific metabolic model development, simulations, drug databases and disease data to aid drug discovery. COMO can be installed as a Docker Image or with Conda and includes intuitive instructions within a Jupyter Lab environment. It provides a comprehensive solution for the integration of bulk and single-cell RNA-seq, microarrays and proteomics outputs to develop context-specific metabolic models. Using public databases, open-source solutions for model construction and a streamlined approach for predicting repurposable drugs, COMO enables researchers to investigate low-cost alternatives and novel disease treatments. As a case study, we used the pipeline to construct metabolic models of B cells, which simulate and analyze them to predict metabolic drug targets for rheumatoid arthritis and systemic lupus erythematosus, respectively. COMO can be used to construct models for any cell or tissue type and identify drugs for any human disease where metabolic inhibition is relevant. The pipeline has the potential to improve the health of the global community cost-effectively by

providing high-confidence targets to pursue in preclinical and clinical studies. The source code of the COMO pipeline is available at https://github.com/HelikarLab/COMO. The Docker image can be pulled at https://github.com/HelikarLab/COMO/pkgs/container/como.

Keywords: immunometabolism, autoimmune diseases, drug discovery, computational pipeline

Introduction

Human diseases often alter cellular metabolism, making metabolic pathways potential targets for treatment.[1] Genome-scale metabolic networks (GSMNs) are powerful computational tools to investigate the effects of drugs on a cell's phenotype and generate testable hypotheses.[2,3] In recent years, the COBRA Toolbox and COBRApy have facilitated the construction and analysis of metabolic models.[4,5] However, constructing tissue- and cell-type-specific GSMNs and performing systems-level studies often require various tools based on different programming languages, APIs, data resources and data formats. For example, R/Bioconductor is well suited for analyzing high-throughput datasets, while Matlab or Python are often used for metabolic modeling and analysis. In addition, model integration with external databases such as DrugBank[6] to run simulations under different conditions often requires additional manual, tedious and error-prone steps. The complexity and diversity of tools and data resources handled with different programming languages with manual steps could pose significant challenges for researchers, potentially leading to errors, inefficiencies and delays in using metabolic

models for therapeutic interventions and other applications. Furthermore, as the amount of disease-specific omics data has increased in recent years, there has been a growing need for efficient bioinformatics tools to interpret and connect these data.[7]

To address these challenges, we developed COMO (Constraint-based Optimization of Metabolic Objectives), a pipeline that integrates data sources and analysis tools in a single, easy-to-use platform for researchers to harness the ever-growing volume of publicly available omics data. The pipeline allows users to analyze and integrate outputs of multi-omics data to build GSMNs for various tissue and cell types in different biological contexts. It combines transcriptomics and proteomics data from user-provided or publicly available databases using standardized meta-analysis techniques to synthesize constraint-based metabolic models and leverages existing knowledge from drug databases to identify potential therapeutic targets for user-specified diseases. COMO allows rapid, customizable, context-specific, constraint-based model generation, disease analysis and drug perturbation analysis. It comes packaged with popular standard tools such as Memote[8], Escher[9], and UMAP[10] for exploratory analysis and clustering. COMO is a general-purpose tool to develop GSMNs spanning organisms and biological scales. The drug discovery steps can be applicable to any disease area in which metabolic inhibition of specific cells and tissues is feasible.

In this study, we showed an application of the novel COMO pipeline to construct metabolic models of B cells and use them for drug target identification against autoimmune diseases: rheumatoid arthritis (RA) and systemic lupus erythematosus

(SLE). RA affects approximately 18 million individuals worldwide, while SLE impacts around 3.41 million people globally, and both of these diseases lack a definitive cure.[11,12] B-cell metabolism has been shown to play a role in the onset of autoimmune diseases. For example, B cells from SLE patients have altered metabolism because of increased activity of the mTOR pathway,[13] and B cells have already been explored as therapeutic targets against SLE.[14] Characterizing the naive B cell metabolism and understanding its role in autoimmune diseases could lead to new therapeutic strategies to restore normal immune function, reduce B cell activity, and prevent the production of harmful antibodies.

Methods

Pipeline Design

COMO consists of four major steps for identifying drug targets (Figure 2).

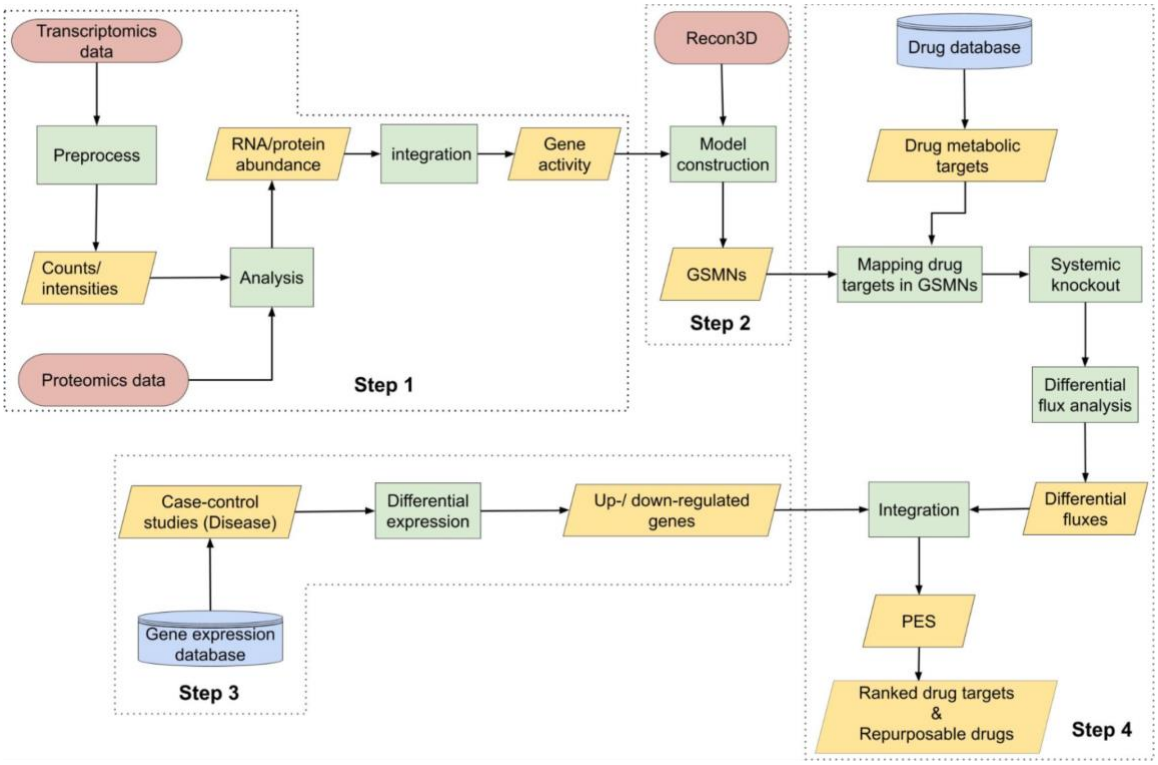


Figure 2: Flow chart of the COMO pipeline. GSMN refers to genome-scale metabolic networks.

The first step involves data analysis and consists of three substeps: (i) if bulk RNA-seq data are used for constructing the models, preprocessing the data and preparing the configuration files; (ii) analysis of any combinations of bulk-RNA-seq, single-cell RNA-seq, microarrays and proteomics data, and generation of a list of active genes for each data type; and (iii) checking for consensus among user-provided data types according to the desired level of rigor and merging them into a single set of active genes.

The second step uses the list of active genes generated in the first step and a reference model, for example, Recon3D[15], provided in the pipeline to build cell-type-specific

models. Users can manually refine these models and provide them in Step 4. Users can also provide their version of the reference model and input datasets, but these must have the same gene annotations (e.g., Entrez IDs or HGNC symbols) if both are user-provided.

The third step analyzes disease-specific data from a user-provided case–control transcriptomics study using bulk RNA-seq or microarrays. COMO uses a platform-based standard pipeline to preprocess, normalize, and identify differentially expressed genes by comparing control and patient groups. Ideally, the disease datasets come from the same cell types as the GSMN models, but if data for the specific cell types are unavailable, it advises mixed-cell populations data such as PBMCs for T cell lymphocytes.

The fourth step performs the drug perturbation simulation for drug discovery and consists of several substeps: (i) mapping of drug targets obtained from the repurposing tool of the ConnectivityMap[16] database to metabolic genes in the model; (ii) systematic knockouts of each mapped gene; (iii) comparison of flux profiles from perturbed and control models to identify differential fluxes; (iv) comparison of differentially regulated fluxes with differentially expressed genes identified in Step 3 to determine the number of genes up- and down-regulated in a disease whose fluxes are in the opposite direction after perturbation with the mapped drug; and (v) computation of the perturbation effect score (PES)[17] for each drug target. The pipeline output includes PES-based ranks of drug targets and mapped repurposable drugs for the studied disease.

File Preparation and Omics Data Analysis

RNA-seq Analysis

For RNA-seq analysis, COMO requires a matrix of gene counts from aligned FastQ reads for each context to build a context-specific constraint-based metabolic model. Users must place all the properly formatted files in designated user data directories to provide input to the COMO pipeline. The outputs of the pipeline will be saved in designated directories. COMO can integrate multiple types of RNA-seq data. The gene counts for each sample are normalized, filtered, binarized and combined. The output of this step is a CSV file containing all genes in all libraries with binary values for each gene, 0 for inactive and 1 for active.

Microarray Analysis

COMO can use microarray datasets from various platforms, including Affymetrix, Agilent and Illumina. The user provides a configuration file containing the GEO database's accession numbers grouped by cell types. The pipeline will automatically download datasets from the GEO database, normalize them and analyze them. Data across multiple samples are combined based on binarized expression values and user-defined ratios.

Proteomic Analysis

Protein abundance data can be used on its own to build context-specific GSMNs or integrated with transcriptomics data to enhance the model. Proteomics abundance data should be imported into COMO as a CSV or Excel file. Similar to the RNA-seq analysis, binarized activity states can be combined across samples using user-specified ratios.

Combining Activity of Different 'omics' Data Sources

COMO merges activity states for any combination of available 'omics' data sources using binarized data. Users define a minimum activity requirement, which indicates the minimum number of provided sources in which a gene should be active for the gene to be active in the model.

Context-specific Model Extraction

Context-specific models can be seeded from a general genome-scale model, such as Recon3D[15] for human tissue models or iMM1865[18] for mouse tissue models. COMO currently supports GLPK (GNU Linear Programming Kit)[19] and GUROBI[20] solvers to subset and solve a context-specific model from the reference global model using a reconstruction algorithm (i.e. GIMME, iMAT, and FASTCORE) implemented with Troppo[21].

Identifying Disease-related Genes

Disease-related genes can be determined using differential gene expression analysis (currently with transcriptomics data). COMO requires a configuration file specifying sample names and annotations for patient and control samples. This step outputs differentially expressed genes.

Identifying Drug Targets and Repurposable Drugs

COMO maps drug targets obtained from the ConnectivityMap[22] database to metabolic genes in the model and performs systematic gene knockouts. It identifies differential fluxes by comparing flux profiles of perturbed and control models and comparing them with differentially expressed disease genes. This comparison determines the number of up- and down-regulated genes in a disease reversed by each mapped drug. These values are input for the PES[17] that can be calculated for each drug target. The output of this step includes PES-based ranks of drug targets and mapped repurposable drugs for the disease of interest.

All the available features in the COMO pipeline are given in Table 1.

Table 1: Available features in the COMO pipeline

Features in COMO	
Data Integration	Model Seeding

Features in COMO	
Microarray Bulk-RNA seq Single-cell RNA-seq Protein Abundance	iMAT GIMME FastCORE
Peripheral Analysis (Jupyter Notebook) Differential Gene Expression Drug Target Analysis (inhibition) Sample Clustering Memote Integration Escher Mapping Integration	Peripheral Analysis (AutoRNAseq Pipeline) Fastq_screen FastQC Trim_galore Picard RNA-seq Metrics MultiQC STAR alignment

B Cell Data Processing and Analysis

We applied the COMO pipeline to drug discovery in B cells. We collected RNA-seq datasets published in GEO[23] and processed them using our accompanying alignment and quality control pipeline. We selected three RNA-seq datasets based on criteria defined in Supplementary Methods for COMO to analyze and merge to construct context-specific models (Supplementary Table 1). We also used published quantitative proteomics data from high-resolution mass spectrometry (Chapter III Supplementary Table 1).

We imported RNA-seq read counts and protein copy number data to the COMO container and processed them. The function 'rnaseq_preprocess.py' was used to generate count matrix files. The automatically generated configuration .xlsx files were modified to exclude RNA-seq datasets, which failed quality control checks (Supplementary Table 2). The parameters used in B cell data analysis are shown in Supplementary Table 3. The RNA-seq counts and protein copy number values were transformed to Z scores. Weights for RNA-seq-based Z-transformed scores and proteomics-based Z-transform scores were 6 and 10, respectively, to follow the 0.6 correlation coefficient found between transcription and protein expression.[24]

Results

Constructing Metabolic Models

We used the combined Z-transformed values based on transcriptomics and proteomics data of naive B cells (Figure 3A) to seed a model using the iMAT and GIMME algorithms. We used biomass maintenance for the objective function of naive B cells since they do not proliferate[25]. We defined exchange reactions based on literature about B cells' nutrient preferences (Supplementary Table 4). Using combined RNA-seq and proteomics datasets, we generated many models with combined Z-transformed value cutoffs ranging from −6 to 4. To assess the functionality of the default zFPKM cutoff of −3 to −2 proposed by Hart et al.[26], we generated calibration models using GIMME and iMAT methods. We started from the lowest functional cutoff of >−4 created from

binarized data for GIMME models. For iMAT calibration models, we used >−5 as a high threshold in combined Z-transform scores. A lower threshold of >−5 was used to prefer using genes that did not show near-zero expression. This will fulfill a minimum flux requirement through reactions associated with genes expressed above the higher threshold. The GIMME algorithm results in models with infeasible and dead-end reactions, which must be made fixed using an algorithm such as FASTCC[27]. The COBRApy version of FASTCC can be used in COMO. We used sanity checks to determine baseline characteristics for each calibration model, including metabolic tasks and reactions contributing to biomass, normalized by total feasible reactions.[4]

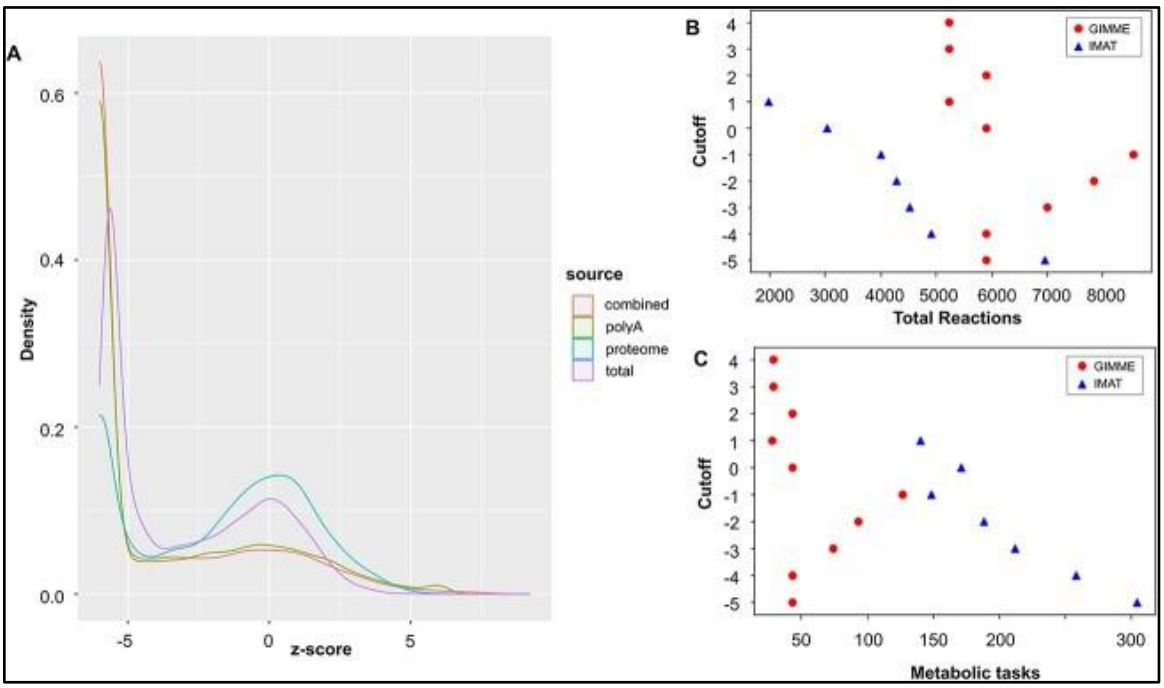


Figure 3: Combined Z-transform scores and their effect on model building and performance. A) Combined Z-transformed scores for different B cell datasets. B) Effect

of Z-score cutoffs on the total number of reactions in the model constructed using GIMME and iMAT algorithms. C) Number of metabolic tasks completed by GIMME- and iMAT-based models constructed with different Z-score cutoffs.

We used results from physiological tests and sanity checks to evaluate the models. To choose the best model among the models generated from different multi-omics data cutoffs, we investigated (i) the number of feasible reactions (Figure 3B) and the number of metabolic tasks passed (Figure 3C) and (ii) the number of reactions in the final model that contributed to maximizing the objective functions (Figure 3C and Supplementary Table 5). The ratio of metabolic tasks versus feasible reactions can help us identify the contextualized model with better retention of global metabolic functions while ensuring its reduction compared to the reference model (as the previously published CD4+ T cell models). A higher number of reactions contributing to the objective functions can indicate greater connectivity of reactions within the network. A Z-transformed value of −3 generated the best iMAT model with 4528 feasible reactions and 212 metabolic tasks passed (Supplementary Table 5). A −3 cutoff has also generated the best GIMME model with 6999 reactions and 74 metabolic tasks passed (Supplementary Table 5). Among iMAT and GIMME models, iMAT models showed better performance regarding the number of completed metabolic tasks and ATP production from different carbon sources (Chapter III Supplementary Data 1). For further analysis, we chose a model based on the iMAT algorithm, considering both the performance in metabolic tasks and the proportion of reactions contributing to the objective function relative to the set of feasible reactions.

Metabolic Model of Naive B Cell

The selected iMAT-based naive B cell model passed all the sanity checks and completed 212 metabolic tests out of global 460 human metabolic tasks (Supplementary Table 5). The model performance on metabolic tasks aligns with previously documented outcomes observed in other cell-specific metabolic models.[28] This model comprises 4528 reactions and 1463 genes (Chapter III Supplementary Data 2). 4064 reactions were internal enzyme–catalyzed and transport reactions distributed across 93 metabolic pathways in different compartments (extracellular, cytosol, mitochondria, endoplasmic reticulum, Golgi apparatus, lysosome, nucleus and peroxisome). Most reactions involved extracellular transport, followed by fatty acid oxidation, mitochondrial transport and peptide metabolism (Supplementary Figure 1).

The model's performance was assessed through global metabolic tasks, but it is imperative to conduct further validation involving B cell–specific metabolic functions. We performed a thorough literature search to collect information available on naive B cell metabolism and validated the naive B cell model by comparing it. We used flux balance analysis (FBA) to identify active pathways and minimization of metabolic adjustment (MoMA) to simulate knockouts. The results presented in Table 2 highlight the agreement between our model and the established biological scenarios in the literature. Among the tested seven complex scenarios, five were in full agreement, and two partially agreed, consistent with previously published literature.[17,29] Several major pathways agreed with the literature, including glycolysis, lactate production, the TCA cycle and

glucose transport (Figure 5). Additionally, in agreement with the literature, our model did not show any impact of glucose removal from the exchange media on growth (Figure 4A). The literature has shown that inhibition of HIF1 induces apoptosis and decreases mature B cells.[30] HIF-1 increases the expression of glycolytic enzymes.[31] We knocked out its target genes to simulate HIF-1's effect on B cells. Among all tested genes, GAPDH knockout reduced B cell growth to zero, while other enzymes affected growth but not significantly (Figure 4B). This approach allowed us to validate our models against known biological scenarios of naive B cell metabolic pathways.

Table 2: Model validation using specific behaviors.

#	Observation from Literature	*In silico* experiment	Simulation result	Agreement?
1	Glucose restriction has no impact on naive B cell function.[32]	Decreased glucose uptake rate and studied its effect on naive B cells growth	No effect on growth	Yes

2	Glucose distributes through G6P, F16BP, G3P and 3PG.[32]	Simulated the model and checked the flux through reactions producing these metabolites	The model produced all the metabolites	Yes
3	Naive B cells produce lactate.[32]	Checked flux through reactions producing lactate	Model produced lactate	Yes
4	In naive B cells, citrate, glutamate, glutamine, α-ketoglutarate, succinate,	Checked the flux inTCA pathway producing these metabolites	Model produced all the metabolites of TCA cycle	Yes

	fumarate and malate are active.[32]			
5	Untreated or anti-IgM treated B cells showed no significant increase in the ECAR in response to the glucose addition.[30,33]	Increased glucose uptake and showed its effect on the naive B cells lactate production, which is directly related to ECAR	Lactate production was unaffected by more availability of glucose	Yes
6	B cell survival requires proper maintenance of glycolytic and oxidative glucose pathways. Inhibition of HIF1	Inhibited glycolytic enzymes that are targets of HIF1 and checked their	Only GAPD and PGK significantly reduced the B cell growth	Partial

	induces apoptosis and decreases mature B cells.[30]	effect on B cell growth		
7	Naive B cells seem to rely on FA as their main fuel.[34]	Removed fatty acids uptake and investigated the effect on B cell growth	A slight decrease was observed in growth when fatty acid uptake was blocked, specifically under glucose-depleted conditions	Partial

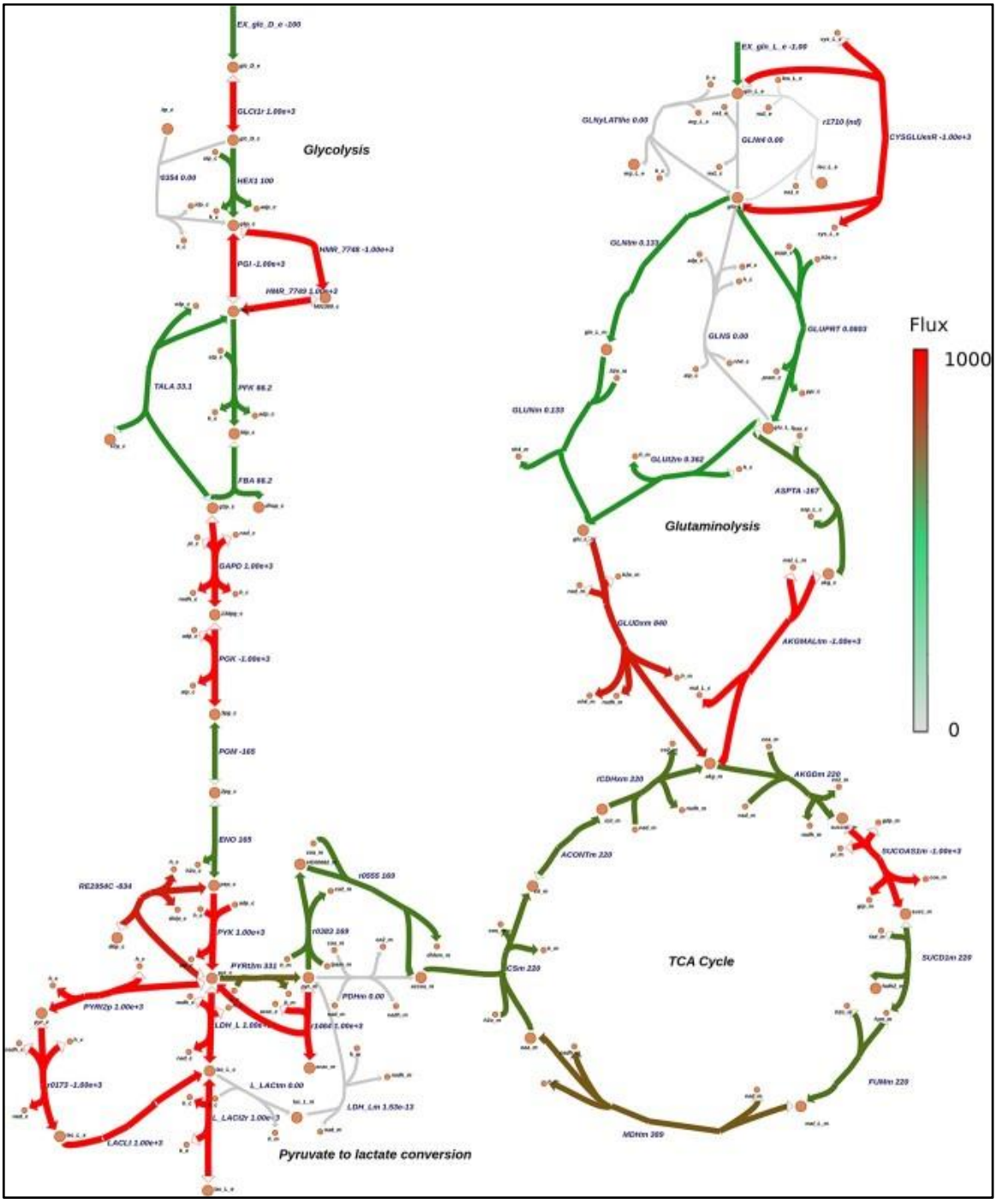


Figure 4: Flux distribution through glycolysis, TCA cycle, and glutaminolysis pathways.

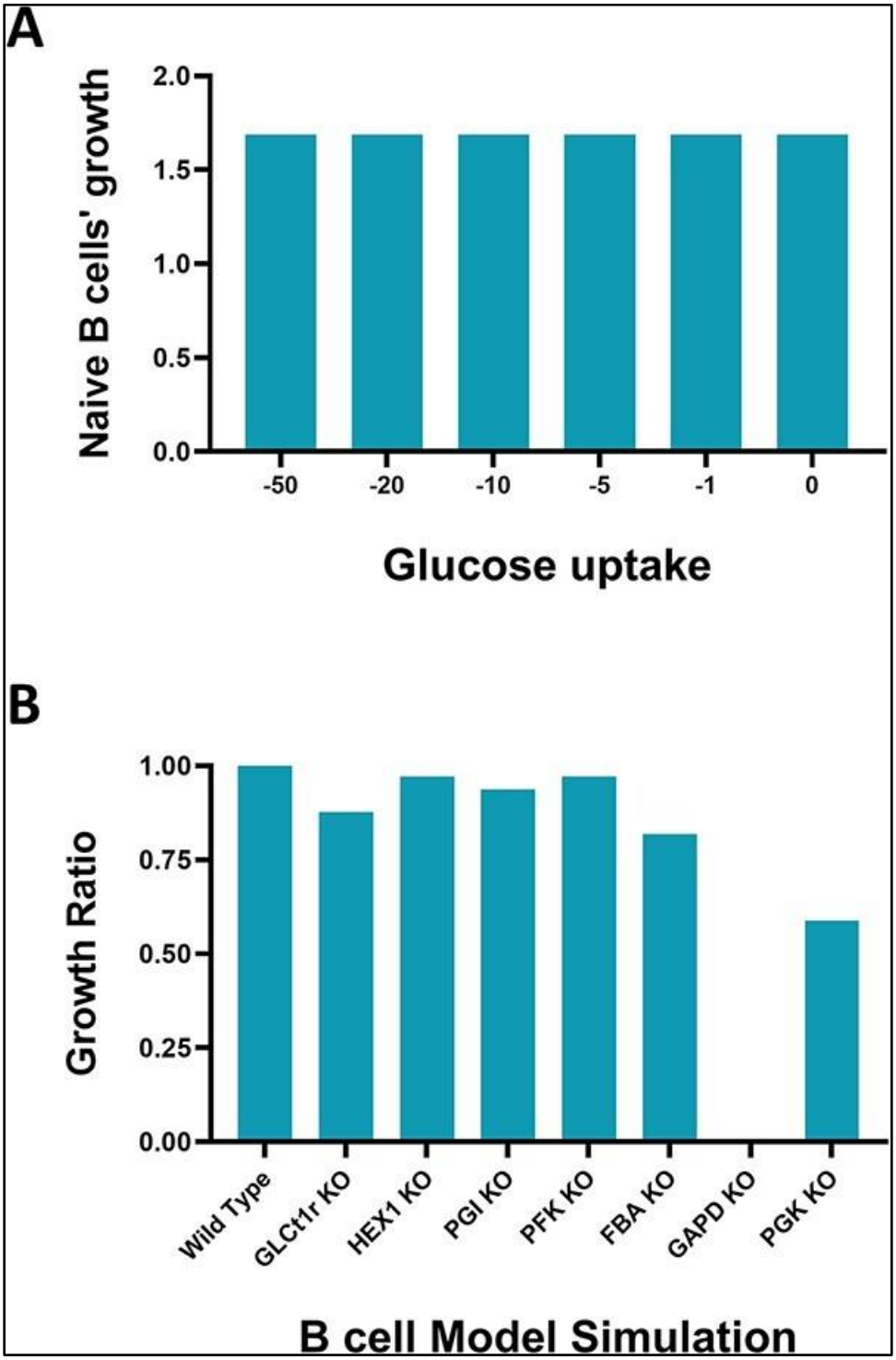


Figure 5: Validation of B cell model. (A) Limiting glucose in media has no impact on naive B cell growth. (B) Impact of glycolytic enzymes knockout (growth ratio compared to wild type). All the tested reactions showed some impact on B-cell growth.

Disease Gene Analysis

We collected a publicly available case–control RNA-seq dataset (GSE110999) of naive B cells for RA and SLE) (Supplementary Table 6). We used COMO's count matrix and configuration files and identified differentially expressed genes (adjusted P-values < 0.05) (Supplementary Table 3).

Drug Target Analysis

We mapped drug targets in the B cell model using connectivityMap's list of drugs and their targets. We simulated knockouts by inhibiting existing drugs' metabolic targets, compared the simulation results with genes up- and down-regulated in RA and SLE and computed the PES[17]. Finally, we used genes with a positive PES to predict potential targets and their associated drugs as potential candidates for repurposing that were validated through literature mining.

A positive PES indicates that reactions associated with differentially expressed genes—when inhibited by drugs—have fluxes in directions opposite to the differential expression. Thus, reactions associated with up-regulated genes have reduced flux, while reactions associated with down-regulated genes have increased flux. A higher PES indicates that the gene's perturbation has a stronger effect on differentially expressed genes. We identified 44 genes for RA and 52 for SLE with positive PES, which were associated with 123 and 120 drugs, respectively (Chapter III Supplementary Data 3).

Among the potential targets identified for RA, APRT (coagulation factor inhibitor) was ranked the highest, followed by GLB1, CTSA, ODC1, TBXAS1, KHK, ADK and ALDH5A1 (Supplementary Table 7). We identified five genes (TBXAS1—rank 5, ABCB1—rank 10, HPRT1—rank 13, SLC10A1—rank 33, SOAT1—rank 40) targeted by the drugs already used to treat RA or are under study for RA. Three targets (TBXAS1, ABCB1, HPRT1) had high-positive PES. Our literature search identified seven RA drugs targeting metabolism: sulfasalazine, azathioprine, mercaptopurine, cyclosporin-A, macrolides, methotrexate and leflunomide. We identified targets of five of these as drug targets based on positive PES (Table 3). The targets of methotrexate (DHFR) and leflunomide (DHODH) had negative PES and thus were not predicted as drug targets. The new potential drug targets included GLB1 (galactosidase beta 1), ODC1 (ornithine decarboxylase) and ALDH5A1 (aldehyde dehydrogenase 5 family member A1).

Table 3: Drug targets identified for each context presented in COMO

Disease	Drug Target	Support
Rheumatoid Arthritis	TBXAS1 (Thromboxane A Synthase 1)	Potential target of the anti-rheumatic drug sulfasalazine[35]
	ABCB1 (ATP binding cassette subfamily B member 1)	Target of macrolides used as RA treatment option[6,36]
	HPRT1 (hypoxanthine phosphoribosyltransferase 1)	Target of azathioprine and mercaptopurine used for RA treatment[37,38]

Disease	Drug Target	Support
	PPAT (phosphoribosyl pyrophosphate amidotransferase)	Target of disease-modifying anti-rheumatic drug cyclosporin-A[39]
	SLC10A1 (solute carrier family 10 member 1)	
Systemic Lupus Erythematosus	PPAT (phosphoribosyl pyrophosphate amidotransferase)	Target of the drug azathioprine used in SLE treatments.[40,41]
	DHFR (dihydrofolate reductase)	Target of the drug methotrexate, used to treat SLE.[6]
	SLC10A1 (solute carrier family 10 member 1)	Target of immunosuppressant drug cyclosporin-A[42]

For SLE, GGPS1 (geranylgeranyl diphosphate synthase 1), GSR (glutathione-disulfide reductase) and GLB1 (galactosidase beta 1) were ranked the top three drug targets (Supplementary Table 7). PPAT (phosphoribosyl pyrophosphate amidotransferase; rank 7), SLC10A1 (solute carrier family 10 member 1; Rank 19) and DHFR (dihydrofolate reductase; rank 23) are the targets of drugs used for SLE (Table 3). Our literature search for SLE drugs targeting metabolic genes identified four drugs (azathioprine, methotrexate, cyclosporin-A and leflunomide). Targets of three SLE drugs (azathioprine, methotrexate and cyclosporin-A) were identified as drug targets (Table 3). The target of immunosuppressant drug leflunomide was identified with negative PES, thus not predicted as a drug target. The other drug targets included GGPS, GSR and ALDH5A1.

Next, we investigated the biological processes that are common in both diseases to understand the similarities of target biological processes in B cells across two autoimmune diseases. Between RA and SLE, we identified 12 common drug targets in the top-ranked list (Supplementary Table 7). The gene enrichment analysis[43] showed that the sphingolipid translocation (ABCA1, ABCC1, ABCB1, GLB1, CTSA) and the de novo IMP biosynthetic process (ATIC, GART, PPAT, GSR, ABCC1, GCLM) were enriched highest in the commonly found genes (Figure 6).

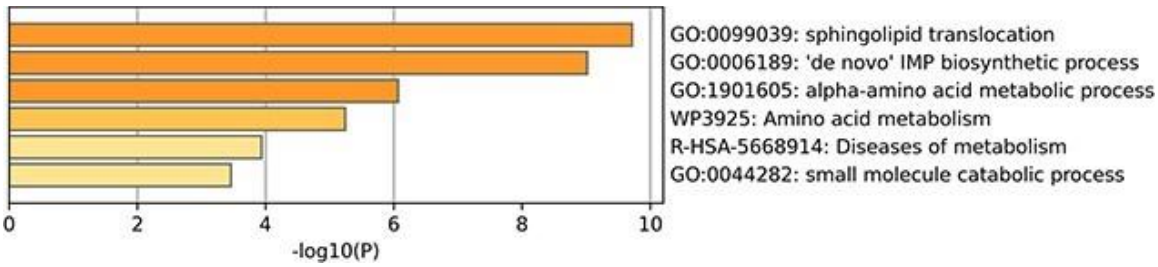


Figure 6: Biological processes and pathways enriched in common drug targets between RA and SLE.

Discussion

Dysregulation of metabolic pathways is often associated with various diseases, making them attractive targets for drug development.[44,45] For example, disrupted metabolic pathways in autoimmune diseases lead to producing harmful immune cells or activating inflammatory pathways.[45] Computational modeling of metabolism can help simulate the effects of gene or reaction perturbations in silico to identify potential drug targets. However, the systematic analysis of such models requires integrating various modeling methods and biological datasets.

COMO is a user-friendly pipeline that processes and integrates transcriptomics and proteomics data, builds context-specific metabolic models, runs simulations, investigates gene inhibition's effect and identifies repurposable drugs and drug targets. COMO uses open-source solutions to facilitate cost-effective disease treatment exploration by researchers. We previously used this approach in identifying drug targets in CD4+ T cell metabolism.[17] COMO is scalable and robust, accommodating single-cell RNA-seq, bulk RNA-seq and proteomics data, as well as microarrays. The integration of multi-omics data allows for a more comprehensive understanding of the metabolic state of a cell or tissue and can provide additional insights into the effects of perturbations on the cell's phenotype.

COMO also presents some limitations. COMO relies on the accuracy and completeness of the available omics data and reference models. If the data are noisy or incomplete, it can introduce biases leading to inaccurate predictions. The noisy data may inherit biases from the original datasets, such as over or under-representation of specific pathways, which could skew the analysis and increase the chance of false positives and false negatives in the target identification. COMO is also limited to the scope of the reference models and databases it interfaces with and will not be able to identify targets or repurposable drugs that are not included in these resources. Addressing these limitations requires rigorous validation and preprocessing of the input data to ensure it is of high quality and suitable for use in the pipeline.

GSMNs can identify potential therapeutic targets by simulating perturbations in metabolic pathways and analyzing the resulting changes in the cell's phenotype.[17,46] To identify drug targets, COMO integrates perturbation-induced flux changes with differentially expressed genes in diseases. Finding metabolic fluxes of up- and down-regulated disease genes closer to the non-disease baseline is based on the assumption that differential regulation of such genes impacts metabolic fluxes. In previous studies, it has been found that the relationship between gene expression and metabolic fluxes is rather complex.[47] However, many studies have showcased the utility of integrating—omics data with metabolic fluxes in various contexts.[17,29,48–50] By using differential gene expression with changes in metabolic fluxes, our group identified drug targets in CD4 T cells against autoimmune diseases, validated by in vitro experiments.[17] In a recent study, Kaste and Shachar-Hill used transcriptomic and proteomic-derived gene expression levels and integrated them with flux prediction, significantly improving the model's agreement with experimental data.[29] Another study combines genetic and environmental perturbation-based differential gene expression and metabolite abundance data with differential fluxes to analyze the altered physiological states.[49] Furthermore, a new approach was proposed incorporating data on differential gene expression to assess metabolic flux differences between two distinct conditions.[50]

We observed notable proportions of existing metabolic targets in the predicted drug targets compared to metabolic genes not predicted as potential drug targets for both RA and SLE. This observation validates the tool's significance in drug target prediction. In drug targets identified against RA, literature evidence supports TBXAS1 (thromboxane A

synthase 1), ABCB1 (ATP binding cassette subfamily B member 1) and HPRT1 (hypoxanthine phosphoribosyltransferase 1). TBXAS1 is a target of sulfasalazine, an older disease-modifying anti-rheumatic drug prescribed as a second- or third-line defense for RA. RA patients show significantly higher levels of thromboxane B2 (TBXAS1 product) in their serum compared to healthy controls.[51] ABCB1 is targeted by macrolides like erythromycin used in RA treatment. Initially, the control of periodontal bacteria was believed to be the reason, potentially contributing to RA progression.[52] However, subsequent studies suggest an anti-inflammatory mechanism, as discontinuing clarithromycin leads to a rapid rebound of RA symptoms.[36] HPRT1 is targeted by azathioprine and mercaptopurine, effective treatments for RA and SLE. Metabolites of each prodrug enter the cell and undergo breakdown in the purine salvage pathway, inhibiting purine synthesis, possibly through non-competitive inhibition of phosphoribosylpyrophosphate amidotransferase (PPAT).[53–55]

In SLE drug targets, PPAT and DHFR (dihydrofolate reductase) were supported by the literature. PPAT is involved in de novo purine biosynthesis and is a target of azathioprine, an immunosuppressive drug used in SLE treatments. However, this drug has not been studied in B cells. DHFR plays a role in the metabolism of folic acid and is an essential enzyme that converts the nutrient folate into its active form. DHFR is known to be a target of the drug methotrexate, used to treat SLE.[56] However, DHFR has not been investigated as a drug target for B cells in SLE.

Among RA targets, several are supported by literature experiments demonstrating the anti-inflammatory effect on various immune cells. GLB1 (galactosidase beta 1) encodes an enzyme that hydrolyzes terminal beta-linked galactose residue from ganglioside substrates and other glycoconjugates. D-Fagomine has been found to control inflammatory processes related to an overactivation of the humoral immune response.[57] ODC1 (ornithine decarboxylase) catalyzes ornithine decarboxylation. In type 3 innate lymphoid cells (ILC3s), deficiency of ODC1 reduced IL-22 production and protected mice from colitis.[58] ALDH5A1 breaks down GABA (gamma-aminobutyric acid), and inhibiting it increases available GABA, potentially suppressing the immune response as B cells produce GABA.[59]

In SLE targets, many have been previously studied and linked with SLE. GGPS1 (geranylgeranyl diphosphate synthase 1) is involved in isoprenoid synthesis, a pathway up-regulated in SLE patients.[60] However, GGPS1's role as a drug target in B cells remains unexplored. GSR (glutathione-disulfide reductase) catalyzes the reduction of glutathione disulfide into glutathione (GSH), an antioxidant crucial for protecting cells from oxidative stress. A decrease in GSH increases ROS production and affects T cell differentiation.[61] ALDH5A1 (aldehyde dehydrogenase 5 family member A1) and ABAT (4-aminobutyrate aminotransferase) break down GABA. ALDH5A1 inhibition makes more GABA available, potentially suppressing the immune response.[59] LTA4H (leukotriene A4 hydrolase) converts leukotriene A4 (LTA4) to leukotriene B4 (LTB4), a potent proinflammatory mediator.[62] FADS1 (fatty acid desaturase 1) regulates eicosanoid

production, proinflammatory mediators linked to SLE pathogenesis.[63] ACACB (acetyl-CoA carboxylase beta) study suggested the role of ACACB in the pathogenesis of SLE.[64]

In summary, using metabolic pathways as drug targets can provide new therapies for complex diseases. COMO can aid in identifying potential drug targets and repurposable drugs. The outputs from COMO, including potential drug targets and repurposable drugs, can be integrated into drug discovery pipelines, involving initial in silico studies for target confirmation, followed by in vitro and animal model experiments for efficacy and safety assessments, and advancing the candidates through clinical trials.

Finally, the B cell case study above used a single model that was refined and evaluated by hand. The work described in the next chapter requires one model across the combination of 166 healthy donors and 35 single-cell RNA-seq labeled cell types (totaling approximately 5,705 metabolic models). Repeating the reconstruction steps thousands of times by hand, while keeping the inputs and settings consistent across each model to compare results, is not easily done without errors or in a reasonable amount of time. This is the problem COMO is built to address. Thus, COMO turns a manual task into an automated pipeline that can be executed on a high-performance computing cluster. The single-cell immunometabolism study in the following chapter rests on this, because attempting to reconstruct each donor-cell-type-specific model under identical conditions without this pipeline (and to correlate age with metabolism) would not be possible.

Funding

This material is based upon work supported by the Defense Health Agency and U.S. Strategic Command under Contract No. FA4600-12-D-9000 and Contract No. FA4600-18-D-9001, and in part by the National Institutes of Health, grant #R35GM119770, and by the Layman seed grant by University of Nebraska Foundation. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Defense Health Agency, U.S. Strategic Command, or 55th Contracting Squadron. Similarly, the opinions or assertions contained herein are the private views of the authors and are not necessarily those of the Uniformed Services University of the Health Sciences, or the Department of Defense. The mention of specific therapeutic agents does not constitute endorsement by the U.S. Department of Defense, and trade names are used only for the purpose of clarification.

Data Availability

The pipeline code is available at https://github.com/HelikarLab/COMO, and the other data generated in this study are available as Supplementary Data available online at http://bib.oxfordjournals.org, or in the attached documents.

References

1. DeBerardinis, R. J. & Thompson, C. B. Cellular Metabolism and Disease: What Do Metabolic Outliers Teach Us? Cell 148, 1132–1144 (2012).

2. Jerby, L. & Ruppin, E. Predicting Drug Targets and Biomarkers of Cancer via Genome-Scale Metabolic Modeling. Clin. Cancer Res. 18, 5572–5584 (2012).

3. Uhlen, M. et al. A pathology atlas of the human cancer transcriptome. Science 357, eaan2507 (2017).

4. Heirendt, L. et al. Creation and analysis of biochemical constraint-based models using the COBRA Toolbox v.3.0. Nat. Protoc. 14, 639–702 (2019).

5. Ebrahim, A., Lerman, J. A., Palsson, B. O. & Hyduke, D. R. COBRApy: COnstraints-Based Reconstruction and Analysis for Python. BMC Syst. Biol. 7, 74 (2013).

6. Wishart, D. S. et al. DrugBank 5.0: a major update to the DrugBank database for 2018. Nucleic Acids Res. 46, D1074–D1082 (2018).

7. Dash, S., Shakyawar, S. K., Sharma, M. & Kaushik, S. Big data in healthcare: management, analysis and future prospects. J. Big Data 6, 54 (2019).

8. Lieven, C. et al. MEMOTE for standardized genome-scale metabolic model testing. Nat. Biotechnol. 38, 272–276 (2020).

9. Escher: A Web Application for Building, Sharing, and Embedding Data-Rich Visualizations of Biological Pathways | PLOS Computational Biology. https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1004321.

10. McInnes, L., Healy, J. & Melville, J. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. Preprint at https://doi.org/10.48550/arXiv.1802.03426 (2020).

11. Shi, G. et al. Estimation of the global prevalence, incidence, years lived with disability of rheumatoid arthritis in 2019 and forecasted incidence in 2040: results from the Global Burden of Disease Study 2019. Clin. Rheumatol. 42, 2297–2309 (2023).

12. Tian, J., Zhang, D., Yao, X., Huang, Y. & Lu, Q. Global epidemiology of systemic lupus erythematosus: a comprehensive systematic analysis and modelling study. Ann. Rheum. Dis. 82, 351–356 (2023).

13. Torigoe, M. et al. Metabolic Reprogramming Commits Differentiation of Human CD27+IgD+ B Cells to Plasmablasts or CD27−IgD− Cells. J. Immunol. 199, 425–434 (2017).

14. Atisha-Fregoso, Y., Toz, B. & Diamond, B. Meant to B: B cells as a therapeutic target in systemic lupus erythematosus. J. Clin. Invest. 131, (2021).

15. Brunk, E. et al. Recon3D enables a three-dimensional view of gene variation in human metabolism. Nat. Biotechnol. 36, 272–281 (2018).

16. Corsello, S. M. et al. The Drug Repurposing Hub: a next-generation drug library and information resource. Nat. Med. 23, 405–408 (2017).

17. Puniya, B. L. et al. Integrative computational approach identifies drug targets in CD4+ T-cell-mediated immune disorders. Npj Syst. Biol. Appl. 7, 4 (2021).

18. Khodaee, S., Asgari, Y., Totonchi, M. & Karimi-Jafari, M. H. iMM1865: A New Reconstruction of Mouse Genome-Scale Metabolic Model. Sci. Rep. 10, 6177 (2020).

19. Oki, E. Linear Programming and Algorithms for Communication Networks: A Practical Guide to Network Design, Control, and Management. (CRC Press, Boca Raton, FL, 2013).

20. Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual. (2026).

21. Ferreira, J., Vieira, V., Gomes, J., Correia, S. & Rocha, M. Troppo - A Python Framework for the Reconstruction of Context-Specific Metabolic Models. in Practical Applications of Computational Biology and Bioinformatics, 13th International Conference (eds Fdez-Riverola, F., Rocha, M., Mohamad, M. S., Zaki, N. & Castellanos-Garzón, J. A.) 146–153 (Springer International Publishing, Cham, 2020). doi:10.1007/978-3-030-23873-5_18.

22. Subramanian, A. et al. A Next Generation Connectivity Map: L1000 Platform and the First 1,000,000 Profiles. Cell 171, 1437-1452.e17 (2017).

23. NCBI GEO: archive for functional genomics data sets—update | Nucleic Acids Research | Oxford Academic. https://academic.oup.com/nar/article/41/D1/D991/1067995.

24. Buccitelli, C. & Selbach, M. mRNAs, proteins and the emerging principles of gene expression control. Nat. Rev. Genet. 21, 630–644 (2020).

25. Moscardó Garcia, M., Pacheco, M., Bintener, T., Presta, L. & Sauter, T. Importance of the biomass formulation for cancer metabolic modeling and drug prediction. iScience 24, 103110 (2021).

26. Hart, T., Komori, H. K., LaMere, S., Podshivalova, K. & Salomon, D. R. Finding the active genes in deep RNA-seq gene expression studies. BMC Genomics 14, 778 (2013).

27. Vlassis, N., Pacheco, M. P. & Sauter, T. Fast Reconstruction of Compact Context-Specific Metabolic Network Models. PLOS Comput. Biol. 10, e1003424 (2014).

28. Thiele, I. et al. A community-driven global reconstruction of human metabolism. Nat. Biotechnol. 31, 419–425 (2013).

29. Kaste, J. A. M. & Shachar-Hill, Y. Accurate flux predictions using tissue-specific gene expression in plant metabolic modeling. Bioinformatics 39, btad186 (2023).

30. Boothby, M. R., Brookens, S. K., Raybuck, A. L. & Cho, S. H. Supplying the trip to antibody production—nutrients, signaling, and the programming of cellular metabolism in the mature B lineage. Cell. Mol. Immunol. 19, 352–369 (2022).

31. Taylor, C. T. & Scholz, C. C. The effect of HIF on metabolism and immunity. Nat. Rev. Nephrol. 18, 573–587 (2022).

32. Waters, L. R., Ahsan, F. M., Wolf, D. M., Shirihai, O. & Teitell, M. A. Initial B Cell Activation Induces Metabolic Reprogramming and Mitochondrial Remodeling. iScience 5, 99–109 (2018).

33. Akkaya, M. et al. Second signals rescue B cells from activation-induced mitochondrial dysfunction and death. Nat. Immunol. 19, 871–884 (2018).

34. Wilson, C. S. & Moore, D. J. B Cell Metabolism: An Understudied Opportunity to Improve Immune Therapy in Autoimmune Type 1 Diabetes. Immunometabolism 2, e200016 (2020).

35. Stenson, W. F. & Lobos, E. Inhibition of platelet thromboxane synthetase by sulfasalazine. Biochem. Pharmacol. 32, 2205–2209 (1983).

36. Saviola, G., Benucci, M. & Cirino, G. Comments on:Effects of clarithromycin in patients with active rheumatoid arthritis. Curr. Med. Res. Opin. 23, 2763–2764 (2007).

37. Dean, L. & Kane, M. Mercaptopurine Therapy and TPMT and NUDT15 Genotype. in MINI Medical Genetics Summaries (eds Pratt, V. M. et al.) (National Center for Biotechnology Information (US), Bethesda (MD), 2012).

38. Suarez-Almazor, M. E., Spooner, C. & Belseck, E. Azathioprine for treating rheumatoid arthritis - Suarez-Almazor, ME - 2000 | Cochrane Library. https://www.cochranelibrary.com/cdsr/doi/10.1002/14651858.CD001461/full.

39. Wells, G. A. et al. Cyclosporine for treating rheumatoid arthritis - Wells, GA - 1998 | Cochrane Library. https://www.cochranelibrary.com/cdsr/doi/10.1002/14651858.CD001083/full.

40. Abu-Shakra, M. & Shoenfeld, Y. xAzathioprine therapy for patients with systemic lupus erythematosus. Lupus 10, 152–153 (2001).

41. Zhou, Y. et al. Therapeutic target database update 2022: facilitating drug discovery with enriched comparative data of targeted agents. Nucleic Acids Res. 50, D1398–D1407 (2022).

42. Germano, V. et al. Cyclosporine A in the long-term management of systemic lupus erythematosus. J. Biol. Regul. Homeost. Agents 25, 397–403 (2011).

43. Zhou, Y. et al. Metascape provides a biologist-oriented resource for the analysis of systems-level datasets. Nat. Commun. 10, 1523 (2019).

44. Du, D. et al. Metabolic dysregulation and emerging therapeutical targets for hepatocellular carcinoma. Acta Pharm. Sin. B 12, 558–580 (2022).

45. Piranavan, P., Bhamra, M. & Perl, A. Metabolic Targets for Treatment of Autoimmune Diseases. Immunometabolism 2, e200012 (2020).

46. Turanli, B. et al. Discovery of therapeutic agents for prostate cancer using genome-scale metabolic modeling and drug repositioning. EBioMedicine 42, 386–396 (2019).

47. Integrating proteomic or transcriptomic data into metabolic models using linear bound flux balance analysis | Bioinformatics | Oxford Academic. https://academic.oup.com/bioinformatics/article/34/22/3882/5033386.

48. Jenior, M. L., Jr, T. J. M., Dougherty, B. V. & Papin, J. A. Transcriptome-guided parsimonious flux analysis improves predictions with metabolic networks in complex environments. PLOS Comput. Biol. 16, e1007099 (2020).

49. Pandey, V., Hadadi, N. & Hatzimanikatis, V. Enhanced flux prediction by integrating relative expression and relative metabolite abundance into thermodynamically consistent metabolic models. PLOS Comput. Biol. 15, e1007036 (2019).

50. Ravi, S. & Gunawan, R. ΔFBA—Predicting metabolic flux alterations using genome-scale metabolic models and differential transcriptomic data. PLOS Comput. Biol. 17, e1009589 (2021).

51. Wang, M.-J. et al. Determination of Role of Thromboxane A2 in Rheumatoid Arthritis. Discov. Med. 19, 23–32 (2015).

52. Antibiotics for the treatment of rheumatoid arthritis | IJGM | Dove Medical Press. https://www.dovepress.com/antibiotics-for-the-treatment-of-rheumatoid-arthritis-peer-reviewed-fulltext-article-IJGM.

53. Fotoohi, A. K., Lindqvist, M., Peterson, C. & Albertioni, F. Involvement of the concentrative nucleoside transporter 3 and equilibrative nucleoside transporter 2 in the resistance of T-lymphoblastic cell lines to thiopurines. Biochem. Biophys. Res. Commun. 343, 208–215 (2006).

54. Karran, P. & Attard, N. Thiopurines in current medical practice: molecular mechanisms and contributions to therapy-related cancer. Nat. Rev. Cancer 8, 24–36 (2008).

55. Tay, B. S., Lilley, R. McC., Murray, A. W. & Atkinson, M. R. Inhibition of phosphoribosyl pyrophosphate amidotransferase from Ehrlich ascites-tumour cells by thiopurine nucleotides. Biochem. Pharmacol. 18, 936–938 (1969).

56. Bedoui, Y. et al. Methotrexate an Old Drug with New Tricks. Int. J. Mol. Sci. 20, 5023 (2019).

57. Segura, S. P., Mate, M. C. A., Pages, M. L. & Torras, M. de los A. C. D-fagomine for the control of inflammatory processes related to an overactivation of the humoral immune response. (2015).

58. Peng, V. et al. Ornithine decarboxylase supports ILC3 responses in infectious and autoimmune colitis through positive regulation of IL-22 transcription. Proc. Natl. Acad. Sci. 119, e2214900119 (2022).

59. Zhang, B. et al. B cell-derived GABA elicits IL-10+ macrophages to limit anti-tumour immunity. Nature 599, 471–476 (2021).

60. KURUP, R. K. & KURUP, P. A. Hypothalamic-Mediated Model for Systemic Lupus Erythematosis: Relation to Hemispheric Chemical Dominance. Int. J. Neurosci. 113, 1561–1577 (2003).

61. Glutathione de novo synthesis but not recycling process coordinates with glutamine catabolism to control redox homeostasis and directs murine T cell differentiation | eLife. https://elifesciences.org/articles/36158.

62. Fourie, A. M. Modulation of inflammatory disease by inhibitors of leukotriene A4 hydrolase. Curr. Opin. Investig. Drugs 10, 1173–1182 (2009).

63. Δ-5 Fatty Acid Desaturase FADS1 Impacts Metabolic Disease by Balancing Proinflammatory and Proresolving Lipid Mediators | Arteriosclerosis, Thrombosis, and Vascular Biology. https://www.ahajournals.org/doi/10.1161/ATVBAHA.117.309660.

64. Harley, I. et al. De Novo Mutation in ACACB in Childhood Onset SLE Highlights a Novel Role As Modulator of Nucleic Acid Sensor-Driven Type I Interferon Responses. in ARTHRITIS & RHEUMATOLOGY vol. 69 (WILEY 111 RIVER ST, HOBOKEN 07030-5774, NJ USA, 2017).

Chapter IV: A Python reimplementation of R's zFPKM transformation

Josh Loecker, Bhanwar Lal Puniya, Tomáš Helikar

Contribution

The decision to reimplement the zFPKM transformation in Python in order to remove COMO's dependency on R arose from discussions with my co-authors. All subsequent work in this chapter is my own. I determined the technical approach, including identifying the specific numerical components including R's density bandwidth selection and linear binning, and the "pracma" findpeaks logic, that had to be reproduced for downstream active/inactive gene classification to remain stable. I wrote all of the source code, including the kernel density estimation, the "bw.nrd0" bandwidth selector, the linear-binning density grid, the peak-detection and μ-selection routines, and the half-normal σ estimation. I designed and executed the validation strategy, including benchmarking against the R/Bioconductor reference implementation on the ENCODE cell-line FPKM matrix, the comparison against the originally published Hart et al. (2013) values, and the comparative assessment of SciPy's "gaussian_kde" and scikit-learn's "KernelDensity" as alternatives. I performed all analyses, generated all figures and tables, and wrote the manuscript in its entirety. B.L.P. and T.H. provided supervisory support; T.H. supervised the project as dissertation advisor.

The code for this chapter is available online at https://github.com/JoshLoecker/zFPKM

Preface & Abstract

zFPKM is a modified *z*-transformation of RNA-seq abundance estimates (FPKM or TPM) that converts each gene's expression as a standardized distance from the dataset's expression of actively transcribed genes. This allows for a binary active/inactive classification after filtering via a fixed threshold (e.g., zFPKM > -3). The original implementation is the R/Bioconductor package, found at "ronammar/zFPKM".[1] The purpose of a from-scratch Python reimplementation is to remove the R runtime and its dependencies from the COMO pipeline. This implementation translates R's "density" and "findpeaks" because neither SciPy nor scikit-learn reproduces R's bandwidth and grid behavior closely enough for downstream classification to remain stable. Against the ENCODE cell-line FPKM matrix used to calibrate the original method (18,915 genes across 17 samples), the reimplementation reproduces the R package output on every sample (Pearson and Spearman both 1.00 and a maximum absolute deviation less than $1 * 10^{-13}$ with identical active-gene identification) and agrees with the originally published Hart *et al.* (2013) values to within approximately 0.12 zFPKM units.

Background

zFPKM

RNA-seq abundance estimates (FPKM or TPM) span several orders of magnitude and contain (1) a larger set of genes that are not actively transcribed, and whose non-zero counts reflect technical and biological noise, and (2) a smaller set that are genuinely

expressed. Previous work demonstrated that, on the log2 scale, per-sample distributions of FPKM were approximately bimodal, where the lower mode corresponded to inactive/background genes, and the higher mode to actively transcribed genes.[2] The zFPKM transform fits a Gaussian to the upper/active peak, and translates every gene's expression as a *z*-score relative to that peak. Because the transform standardizes against the active-gene peak rather than the entire distribution's mean and variance, a single cutoff value can be used across datasets (zFPKM > -3) to indicate gene activity. This cutoff was calibrated against the ENCODE dataset using open- and closed-chromatin promoters, and it marked the point at which active promoters began to outnumber repressed promoters (Figure 7).

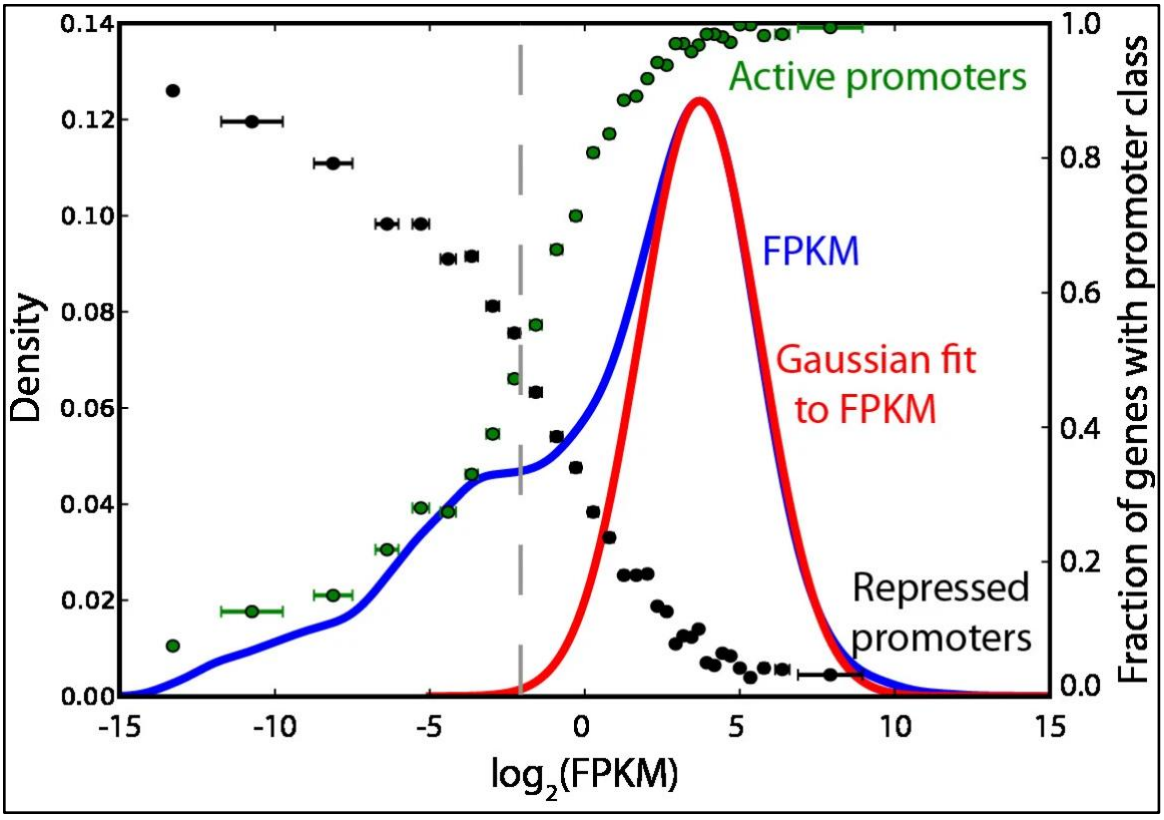


Figure 7: FPKM versus zFPKM expression profile. Black points represent repressed promoters, and green represents active promoters. At a log2(FPKM) of -3, the ratio of repressed to active drops to less than 1, indicating more active promoters are present than repressed promoters. Figure from Hart, 2013.

Conceptual Basis

For a single sample, let x = log2(FPKM). The method (i) estimates the density of x with a Gaussian kernel density estimation (KDE); (ii) identifies the location μ of the peak for active genes; (iii) estimates the spread of the active-gene peak using only the right half; and (iv) standardizes the result:

1. x = log2(FPKM)
2. d = KDE(x) evaluated on a fixed grid
3. μ = the peak position of the active-gene mode
4. U = mean of x values greater than μ (e.g., the mean of the right half of the active peak)
5. $\sigma = (U - \mu) * \sqrt{\pi/2}$ (half-normal mean-to-standard-deviation conversion)
6. $zFPKM = (x - \mu) / \sigma$

Step 5 is the key component of the transformation because for a normal distribution, the mean of the values exceeds the mode by exactly $\sigma \cdot \sqrt{(2/\pi)}$. Inverting this relationship recovers σ from the observed right-tail mean (U) without being influenced by the overlapping inactive-gene peak on the left.

Importance

In this work, the zFPKM transformation is an optional upstream step in the COMO genome-scale metabolic modeling pipeline. Genome-scale metabolic model extraction algorithms, such as iMAT and GIMME, consume a binarized form of gene activity, where active genes are marked with a "1" and inactive by "0", regardless of true expression.[3,4] This association allows a context-specific model to be reconstructed from a generic, genome-scale model, such as Recon3D.[5] zFPKM provides an activity threshold cutoff that can be used to binarize data into iMAT and GIMME algorithms.

Motivation

The reference zFPKM implementation is only available in R. Using this package in Python is possible, but difficult to manage from a development perspective.

An R installation with the correct Bioconductor packages is a strict dependency of a Python project. This couples the Python environment to a secondary language toolchain whose versions must be pinned and reproduced together, making code distribution (via containerization or virtual environments) more difficult. Additionally, it forces data transfers across the R/Python boundary, where Python matrices are converted to an R object and back. These issues result in runtime overhead, the risk of silent data type coercion, and multiprocessing complexities. A Python implementation eliminates the need for the R runtime entirely. It operates directly on the data, simplifies the development and code-execution environment by removing dependencies, streamlines packaging and distribution, and keeps the zFPKM transformation process testable and maintainable within a single programming language.

Methods and Implementation

Algorithm

The Python-based zFPKM accepts raw FPKM or TPM values and processes each sample independently. For a column of FPKM values, it computes a log2 transformation and performs the six steps listed under “Conceptual Basis”. The peak location μ is obtained as follows: the KDE curve d is searched for local maxima with a minimum peak height of

0.02, the x-positions of detected peaks are collected, and μ is set to the right-most of those positions (the largest log2(FPKM)). In Figure 1, this selects the "active-gene" peak even if it is not the tallest peak in the FPKM distribution. The standard deviation is calculated as $\sigma = (U - \mu) * \sqrt{\pi/2}$, where U is the mean of log2(FPKM) values above μ. The column is standardized using $(x - \mu)/\sigma$. Algorithm 1 demonstrates this process as pseudocode.

Algorithm 1: zFPKM transformation via kernel density estimate.

```
FUNCTION kernel_density_estimate(FPKM):    # vector of raw FPKM values for one sample
    x ← log2(FPKM)                         # log2 transformation
    d ← KDE(x, grid)                       # kernel density estimate on a fixed grid
    μ ← peak_position(active_gene_mode)    # peak of active-gene mode (selection rule,
§3.1)
    U ← mean( { x[i] : x[i] > μ } )        # mean of the right half of the active peak
    σ ← (U - μ) · sqrt(π / 2)              # half-normal mean→std-dev conversion
    zFPKM ← (x - μ) / σ                    # standardize
    RETURN zFPKM
END FUNCTION

FUNCTION zFPKM(matrix):                            # matrix of raw FPKM or TPM values
    results ← empty list                           # per-column transformed values
    metadata ← empty list                          # per-sample metadata

    FOR EACH column x_raw IN matrix:               # process each sample independently
        x ← log2(x_raw)                            # log2 transformation
        d ← kernel_density_estimate(x)             # KDE curve

        # --- find peak location μ ---
        peaks ← find_local_maxima(d, min_height = 0.02)
        peak_positions ← x_values_of(peaks)        # x-positions of detected peaks
        μ ← max(peak_positions)                    # right-most peak (largest log2(FPKM)
)

        # --- standard deviation ---
        U ← mean( x[i] FOR i WHERE x[i] > μ )      # mean of log2(FPKM) above μ
        σ ← (U - μ) · sqrt(π / 2)

        # --- standardize ---
        z ← (x - μ) / σ

        APPEND z TO results
        APPEND {
            density:       d,
            mu:            μ,
            sigma:         σ,
```

```
            density_at_mu: value_of(d, μ)          # density height at μ
        } TO metadata

    RETURN results AS matrix, metadata
END FUNCTION
```

Implementation Details

Kernel Density Estimate

A Kernel Density Estimate (KDE) is a non-parametric method for estimating the probability density of a variable from a given dataset.[6] KDEs construct a smooth density curve by placing what is known as a "kernel" function (typically a normal/Gaussian-shaped curve) centered at each data point and summing the contributions across all points. As kernels accumulate at a given point, the density increases. The width of each kernel is controlled by a parameter known as the "bandwidth" (described below), which determines the trade-off between the smoothness and fidelity of the resulting curve when compared to the raw data. A large bandwidth over-smooths and obscures the original data points, while a small bandwidth produces a noisy, jagged estimate. KDEs are particularly valuable when the underlying distribution is unknown or multimodal, as shown in the FPKM data in Figure 1. In the context of zFPKM normalization, the KDE is applied to the log2(FPKM) transformed expression values of a sample to identify the peak of the active gene population, from which the mean (μ) and standard deviation (σ) of the active distribution are derived. Because all subsequent z-score calculations and the active/inactive classification of genes are derived from this density estimate, the KDE accuracy is fundamental to reproducing the original results.

Bandwidth Selection

Bandwidth (denoted by the variable "*h"*) is a parameter of KDE calculations that controls the width of each kernel placed at every data point. It is the primary mechanism controlling the density estimate's smoothness. As *h* increases, each kernel broadens and overlaps with its neighbors, producing a smoother but increasingly biased density that risks eliminating features of the underlying distribution. As *h* decreases, the kernels become narrower, and the estimate better reflects individual observations, but progressively noisier and susceptible to local peaks. This trade-off makes bandwidth selection an important component in KDE calculations, and the default values provided by R's "density" function, SciPy[7], and scikit-learn[8] Python packages significantly contribute to the differences observed between implementations.

Several standard "selectors" exist to approximate an optimal bandwidth. All selectors assume the underlying data are drawn from a Gaussian distribution, and derive *h* accordingly. R's default selector ("bw.nrd0") implements a modification of Silverman's rule of thumb (Equation 1). The interquartile range is divided by 1.34 because, for a Gaussian distribution, the interquartile range is approximately 1.34 times the standard deviation. Thus, dividing the IQR by 1.34 makes it more robust to outliers. The minimum of whichever measurement is more conservative (sample standard deviation or IQR/1.34), which ultimately "protects" against bandwidth inflation when the data contains heavy tails or outliers that inflate the sample standard deviation. SciPy's "gaussian_kde" function, by contrast, defaults to Scott's rule, omitting both the leading

0.9 coefficient and the IQR-based fallback (Equation 2). Scikit-learn's "KernelDensity" function offers both "Silverman" and "Scott" selectors, but each relies solely on the sample standard deviation. For data that closely resembles a Gaussian curve, these missing pieces compound multiplicatively; the missing 0.9 coefficient widens the bandwidth, and for any data where the sample standard deviation is inflated (which is common in log-transformed RNA-seq expression values due to inactive genes), the divergence grows because R's implementation constraints it with the IQR. These differences in *h* propagate into downstream results because this value sets the width and height at which density features are calculated.

$$h = 0.9 * min(\hat{\sigma}, \frac{IQR}{1.34}) * n^{-1/5}$$

Equation 1: Silverman's Rule of Thumb, the default selector in R. $\hat{\sigma}$ is the sample's standard deviation, and IQR is the Interquartile Range. *n* is the number of data points in the sample

$$h = n^{-1/5} * \hat{\sigma}$$

Equation 2: Scott's Rule, the default selector in SciPy. $\hat{\sigma}$ is the sample's standard deviation. *n* is the number of data points in the sample

Density Grid

Evaluating the KDE continuously for every point (gene expression value) in the dataset would require computing the kernel function *n* times for every desired output location *m*, resulting in a very large "Big-O" complexity of O(n*m). Instead, R (and this Python reproduction) converts the output points into a uniform grid of *m* evenly spaced points and performs the density estimation on that grid. This approach is known as linear

binning and works in two stages. First, each observation x_i is distributed between its two nearest grid points proportionally to its distance from each, creating a weighted value at every point in the grid. Second, the Gaussian kernel is combined with these binned counts across the entire grid.

Peak and Mu Identification

Following this, the peaks of the calculated density are identified. This task is conceptually simple: a peak occurs wherever the density transitions from increasing to decreasing, i.e., wherever the slope changes sign from positive to negative. In the context of log2(FPKM) data, the density is bimodal, with a peak on the left side from the inactive gene population and a peak on the right from the actively transcribed gene population. Because μ in the zFPKM transform is defined as the peak of the active gene distribution, it is the right-most peak that must be identified accurately.

Peak detection, however, is sensitive to implementation details in subtle ways. A density estimated on a discrete grid, as described above, may contain many small fluctuations that technically match a local maximum without corresponding to any real peak of the underlying FPKM distribution, and different computation libraries implement different definitions of what is marked as a qualifying peak. This implementation translates the R package Practical Numerical Math[9] "findpeaks" function. At each consecutive pair of grid points, this function examines if density is rising or falling, and translates that sequence into a string of "+" (for rising) and "-" (for falling) characters. A peak is then simply any sequential series of increases immediately followed by a continuous series of

decreases. The spot that turns from "+" to "-" is the peak's location. Two additional filters are then applied to discard candidates that are unlikely to represent genuine gene populations. First, any peak whose density is too small to plausibly reflect a real cluster of genes is removed. Second, any peak that sits too close to a taller neighboring peak is ignored because it would otherwise prevent a single, broad peak from being reported as two separate peaks, simply because the density is not perfectly smooth. The rightmost peak that survives both filters is taken as μ.

Half-Gaussian Standard Deviation

At this point, μ has been calculated as the peak of the active gene population, but the population-level σ is unknown. Ideally, σ would be estimated by directly computing the standard deviation of all active genes, but the left tail of the active gene distribution is problematic because it overlaps with the inactive gene population. The expression levels of the inactive and low-expressed genes bias any estimate coming from observations below μ. The right side of the distribution, however, is largely free of this, because inactive genes do not contribute meaningfully above the active population's FPKM values, and thus the active gene's peak. This means the right half of the density (the genes with a log2(FPKM) above μ) can be used to generate the full Gaussian distribution's spread.

The conversion from the mean of the right half to calculate the standard deviation relies on the simple property of a Gaussian distribution, where if the active gene population is symmetric about μ, then the right half alone is a "half-normal Gaussian" distribution, and

the mean of a half-normal distribution with a standard deviation σ sits exactly $\sigma * \sqrt{2/\pi}$ above zero. After re-arranging, the standard deviation can be calculated as $\sigma = (U - \mu) * \sqrt{(\pi/2)}$, where U is the mean of all log2(FPKM) values above μ. Put simply, the observed average distance of genes above μ from the peak is a predictable fraction "σ" for any Gaussian distribution, and $\sqrt{\pi/2}$ is the constant that converts the observed average distance back into the standard deviation of the full symmetric distribution.

Results

The validation matrix is the ENCODE cell-line FPKM matrix, on which the method was originally calibrated. This dataset contains 18,915 genes across 17 samples (Supplementary Table 1).

All samples match the R package to approximately 16 decimal points, which is the smallest value current machines can represent, approximately 2.2×10^{-16}. The per-sample and pooled Pearson and Spearman correlations are 1.000. The active-gene classifications are identical. Of the 271,379 finite entries, 218,035 are active, and 53,344 are inactive by both this implementation and R. Assuming R is the gold standard, the Python reproduction shows zero false-positive and zero false-negative classifications with a Jaccard index of 1.000000. The SciPy implementation fails to reproduce the gold standard exactly. The Pearson and Spearman correlations are 0.99986 and 0.99978, and 1,029 of these entries are falsely marked as inactive (zero false positives). The scikit-learn implementation also fails to exactly reproduce the Python implementation described

here. The Pearson and Spearman correlations are 0.99966 and 0.99964, with 579 entries misclassified (62 false positives, 517 false negatives).

Limitations

The method inherits the same assumptions as the original ronammar/zFPKM package. It presumes a reasonably clear bimodal separation between inactive and active genes on the log2(FPKM) scale. If the two peaks overlap or the distribution becomes unimodal-like, the active-gene peak is poorly defined. Thus, peak detection becomes unreliable, and the resulting zFPKM score and active/inactive classification are similarly affected.

The method does not calibrate well for transcript-level quantifications. The original authors restrict its validated use to gene-level FPKM or TPM, and the same restriction applies here. Likewise, the transform was designed for bulk RNA-seq. Single-cell data and pseudobulk aggregates can violate the bimodality assumption due to dropout and differences in the mean and variance structure. These inputs were not validated in this work and are considered out of scope.

References

1. Ammar, R. & Thompson, J. zFPKM: A Suite of Functions to Facilitate zFPKM Transformations. (2024). doi:10.18129/B9.bioc.zFPKM.

2. Hart, T., Komori, H. K., LaMere, S., Podshivalova, K. & Salomon, D. R. Finding the active genes in deep RNA-seq gene expression studies. BMC Genomics 14, 778 (2013).

3. Zur, H., Ruppin, E. & Shlomi, T. iMAT: an integrative metabolic analysis tool. Bioinformatics 26, 3140–3142 (2010).

4. Becker, S. A. & Palsson, B. O. Context-specific metabolic networks are consistent with experiments. PLoS Comput. Biol. 4, e1000082 (2008).

5. Brunk, E. et al. Recon3D enables a three-dimensional view of gene variation in human metabolism. Nat. Biotechnol. 36, 272–281 (2018).

6. Silverman, B. W. Density Estimation for Statistics and Data Analysis. (Routledge, 2018).

7. Virtanen, P. et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nat. Methods 17, 261–272 (2020).

8. Pedregosa, F. et al. Scikit-learn: Machine Learning in Python. J. Mach. Learn. Res. 12, 2825–2830 (2011).

9. Borchers, H. W. Pracma: Practical Numerical Math Functions. (2025).

Chapter V: Genome-scale metabolic networks reconstructed from donor-specific scRNA-seq are indicative of sub-cellular, age-related changes

Josh Loecker, Narayana Puniya, William Bryant, Kashish Syed, Bhanwar Lal Puniya, Tomáš Helikar

Contribution

I performed all computational work presented in this chapter. I built the reconstruction pipeline that generated the 5,705 donor- and cell-type-specific metabolic models, wrote the flux sampling workflow, and implemented the distributed execution across UNL's Holland Computing Center. I designed the automated validation framework and wrote the majority of its implementation, with code contributions from Narayana Puniya, William Bryant, and Kashish Syed, undergraduate students whom I mentored. I performed all statistical analysis and selected the statistical tests used. I generated all figures and tables, interpreted the results, developed the mechanistic hypotheses presented, and wrote the chapter. Bhanwar Lal Puniya and Tomáš Helikar served as advisors, providing guidance on project direction.

Introduction

Beyond the well-documented shift in cell-type composition, such as the reduction of naive T cell quantities and the increase of memory, effector, and inflammatory subsets,

aging is accompanied by a chronic, low-grade inflammatory state often termed "inflammaging".[1,2] These changes are not uniform; individual cells often follow distinct trajectories, and the functional consequences of aging are traceable to specific subsets rather than the entirety of the immune system.[1,2]

Single-cell RNA sequencing (scRNA-seq) resolves these differences by quantifying gene expression in individual cells, allowing them to be grouped into specific types and states, and ultimately to be profiled separately.[3,4] Because the immune cell population shifts with age, a bulk measurement complicates the change in a cell's internal gene expression with the change in how many of that cell type are present. scRNA-seq separates these two ideas, making it possible to query how a particular cell type changes with age, independent of how its abundance changes in a given sample.

Gene expression alone, however, reports on transcriptional state rather than on metabolic function. Constraint-based, genome-scale metabolic network models (GSMNs) map this transcriptional state onto metabolic function. A GSMN is a stoichiometric reconstruction of the known metabolic reactions of an organism or cell, encoding which metabolites each reaction consumes and produces and which genes are transcribed (and later translated) to the responsible enzymes.[5,6] These relationships work under the assumption that the network is at a steady state and impose a mass-balance constraint on the system, which defines a "solution space" of feasible flux distributions. Methods such as flux balance analysis and flux sampling explore this space to estimate the amount of flux each reaction can carry.[7] By incorporating a cell's expression data as additional constraints, a

context-specific GSMN can be reconstructed that reflects the metabolic capabilities supported by that particular cell's (or tissue's) transcriptome.

GSMNs report flux at the scale of individual reactions, which can be grouped into pathways. Because of this resolution, metabolic hypotheses can be computationally tested for a given hypothesis by bridging a gap in the available data. Because metabolite concentrations and fluxes cannot be measured directly for thousands of donor- and cell-type- specific populations, nor were those metabolite concentrations reported in the upstream data utilized by this study, a GSMN built from expression data provides an estimate of the metabolic capacity each population could sustain.

Prior work has combined scRNA-seq with metabolic modeling, but largely in contexts and at resolutions distinct from the present question. Tools such as Compass[8], scFEA[9], and scFBA[10] estimate metabolic flux from single-cell data, but typically operate at the level of cell clusters or pre-defined groups, are often applied to cancer or to a single immune subtype, and may be restricted to a reduced portion of the metabolic network. Larger reconstruction efforts, such as the context-specific GSMNs spanning numerous human cell clusters[11], establish that genome-scale models can be derived from scRNA-seq at scale, but do not address immune aging. Conversely, the many single-cell studies of immune aging characterize aging primarily through changes in cell-type proportions and gene expression, rather than through the metabolic flux those genes can support.[12–15] To the current date, no prior study has reconstructed donor- and immune-subtype-

resolved genome-scale metabolic models at this scale and used them to ask which metabolic capacities change with age.

Using the Terekhova et al.[12] dataset of approximately two million cells from 166 donors aged 25 to 85, 5,705 donor-cell-type-specific GSMNs were reconstructed across the immune subtypes present in the data, the feasible flux through each model was estimated by sampling, and flux was tested for association with donor age at the pathway-specific level. The result is a prioritized set of metabolic pathways whose capacity differs between younger and older donors, framed as candidate hypotheses for mechanistic follow-up.

Methods

Single-cell RNA Sequencing

This study uses data from Terekhova et al., a single-cell dataset comprising approximately 2 million cells from 166 unique donors aged 25 to 85 years.[12] This data was provided as a single dataset and split into individual donor and cell type subsets. Donors are labeled in the format of [Age Group][Identifier], where the age group is a single letter from A to E. Female donors have an “F” prefixed to the age group. The identifiers are numerical values that do not correlate with the donor’s age (e.g., donor A01 may be older than A02). Up to three samples were collected from each donor over approximately 1 year.

Genome-scale Metabolic Modeling

Reconstruction of Cell-type, Donor-specific Metabolic Models

COMO was used to reconstruct genome-scale metabolic models for each donor's individual cell-type-specific population. Each reconstruction consisted of four groups of input data:

1. The single-cell expression dataset for the donor-specific cell type was loaded. Only cell types from a curated, pre-selected list were reconstructed; this specifically prevented the reconstruction of a GSMN from the "Unknown" sub-dataset.
2. Recon3D[6] was used as the reference model
3. Per-cell-type reaction constraints were provided. This includes (a) boundary conditions, which limit the input and output of specific metabolites, (b) force reactions, which will be forcibly incorporated into the reconstructed model regardless of its gene expression, and (c) exclusion reactions, which will be forcibly removed from the reconstructed model regardless of its gene expression.
4. The top and bottom 25th percentiles were calculated for a donor's cell-type gene count data. These values were used as thresholds for the iMAT algorithm to segment the data into low, medium, and high expression bins.
5. The reconstruction configuration, including the scRNA-seq dataset, cell type, linear programming solver, number of parallel workers, and CPUs per worker, was defined to allow for maximum parallel execution. Because linear

programming solvers do not show a significant speedup beyond approximately 2-4 CPUs per task, the number of solver CPUs was limited to 2 to maximize concurrent reconstructions.

The reconstruction process was made resumable, allowing existing intermediate and model outputs to be skipped, improving overall performance; it was also made to work on a local machine and on a high-performance SLURM compute cluster, specifically UNL's Holland Computing Center.

The reactions included in the reconstructed models are influenced by the following factors (accompanied with an explanation describing the impacts of that factor):

1. Gene activity threshold. iMAT was used as the reconstruction algorithm for all models. Supplementary Table 5 in COMO (Chapter III) describes the impact of modifying gene activity thresholds for iMAT. While threshold sensitivity was not evaluated here, the referenced supplementary tale demonstrates that as iMAT thresholds increase, the number of reactions included in the reconstructed model decreases linearly (R2 = 0.9059).
2. Boundary reactions. Modifying boundary reactions can substantially impact model flux values, most notably the flux distribution through the network and the value of the objective function. If a single exchange reaction is the only limiting factor, the objective value will scale linearly with the flux allowed through this exchange.
3. Forced reactions. Reactions added to a reconstructed model may introduce artifacts that are not readily apparent. If the reaction's reversibility is not confirmed, it can result in infeasible energy-generating cycles that inflate flux

values, and mass- or charge-imbalanced additions act as erroneous sources or sinks of metabolites. However, quality control steps can easily pinpoint these issues through mass and charge balance verification and testing for energy generation with all exchange reactions constrained to zero.

The boundary conditions for each cell type reconstruction are listed in Supplementary Table 2, and the force reactions in Supplementary Table 3. The same set of cell-type-specific boundary and force reactions was used across all donors. The boundary conditions are similar across all cell types, as a fundamental set of metabolites must be supplied to all cells for basal growth.

Validation of Cell-type, Donor-specific Metabolic Models

To assess if each donor-cell-type-specific GSMN behaves in a biologically plausible manner, an automated validation framework was constructed to test all models against a series of biologically motivated flux tests and reduce each result to a single qualitative outcome of “pass”, “fail”, or “partial pass” (with additional metadata). Each test encodes a specific testable claim about the expected behavior of that cell’s metabolism (e.g., that a given cell type utilizes the citric acid cycle for energy production, or that cell growth is inhibited when a specific reaction is knocked out). The framework is organized into layers with distinct responsibilities, ensuring that each component is decoupled from the cell-specific biology being verified. This separation allows the same validation logic to be reused across all cell types, with each type contributing only the parameters that make a generic test specific to a biological assertion. At the time of writing, not all sub-cell

types had validations available in this pipeline. Table 1 depicts cell types with validations that were present.

Architecture

The framework is built from the following six components. (1) A shared data format defines the set of possible outcomes (pass/fail/partial), the direction of optimization (maximize or minimize a target), the expected qualitative response of a reaction or pathway flux to perturbation (increase, decrease, or no change), and threshold values that separate passing and failing results. All downstream components are expressed in terms of these shared types. (2) Model transformations are reusable modifications to a GSM; for example, adding an exchange or sink reaction, changing the objective function, or closing a set of normally defined nutrient uptakes. Each transformation is applied using a reversible context, so the edits only last for the duration of a single test; the model is automatically reverted to its original afterward. This ensures that tests cannot contaminate one another through accumulated (and unintended) changes. (3) Validation strategies contain the actual testing logic. Each strategy follows a common step: (a) a “simulation” stage applies any configured transformations, simulates the model to compute flux values, and temporarily stores the results, and (b) an “evaluation” stage that transforms the results into a single pass, fail, or partially passing verdict with a template populated by the results of the validation. Each strategy contains an identifying key, the cell type, a description of the test, the objective function, the transformations to apply, the thresholds separating passing from failing results, and the final results. These validation strategies

operate either on a single GSMN or compare a baseline model against a second model. Because every validation exposes the same simulation and evaluation step, the Execution step can run any test interchangeably without knowing which biological question it represents. (4) Cell-type definitions are where biology is encoded. For each cell type, the framework defines a collection of preconfigured strategies, parameterized with reactions, metabolites, expectations, and thresholds, to represent that cell's known metabolism. (5) The Configuration and Execution steps tie the defined validation strategies to a specific GSMN. By default, a mapping exists between the cell type's file name and the collection of tests appropriate for that cell type. Given a model file, the framework identifies the matching test collection, bundles it with the model, and transfers it to the execution engine. (6) A series of supporting utilities provided more nuanced error messages with suggestions for mistyped labels, missing reactions, and helper routines for resolving how fluxes are compared and aggregated.

Execution

For a batch of models, the framework discovers the relevant model file and maps each model to its appropriate set of validations. Each model is then loaded, and its validations are executed. To handle large batches efficiently, independent models are processed concurrently across multiple worker processes. In the case of a single model's validation failing from a technical error, such as attempting to read the flux value of a non-existent reaction, that test is recorded with a unique identifier (e.g., "execution error") in the aggregated results and the remaining tests proceed unaffected.

Validation Tests

Validations are grouped below by the task they are designed to perform. In the following descriptions, a "fraction" means to multiply the original bounds of a reaction by a series of percentage values (90%, 75%, 50%, etc.) to determine the direction (increase/decrease/minimal change) the resulting flux is trending after perturbation.

Flux Capacity and Ratio

- Flux Capacity: Asserts that a target reaction or pathway is able to carry flux. The model is restricted to a fraction of its optimal objective value, and flux variability analysis is used to determine the possible range of flux values each target reaction can hold. A reaction's capability is taken as the larger of the "minimum" and "maximum" magnitudes, so that a reaction working in reverse (with a very negative "minimum" flux) still counts as capable.
- Flux Ratio: Asserts a minimum ratio between the aggregated flux of one reaction or pathway and a reference reaction or pathway. This is used to check the relative usage of two pathways within a single model.

Nutrient and Reaction Perturbation

- Nutrient Availability Response: Scales the uptake or secretion bounds of one or more reactions (such as an entire pathway) across a series of fractions and observes the resulting flux through a target reaction or pathway. The observed flux is aggregated to a single throughput value, and a ratio of the perturbed to the

original flux value is calculated. The result is compared against an expected direction.

- Reaction Inhibition: Scales a target reaction's bounds across a series of fractions, with a fraction of 0% being a true knockout, evaluated against a MOMA[16,17] reference set of flux values. The result is checked whether growth falls below a percentage of its original flux value.
- Inhibition with Re-routing: Inhibits a reaction or whole subsystem across fractions against a MOMA reference and observes how flux (if any) is re-routed through an alternative, secondary reaction/pathway. This probes how the GSMN compensates for a localized disruption.
- Exchange Compensation: Knocks down (or out) one exchange reaction (for metabolite uptake/section) and measures if the flux through a second exchange reaction changes in compensation.

Byproduct and Overflow Metabolism

- Byproduct Reachability: Adds a drain for a metabolite, makes that drain the objective, and checks whether the GSMN directs flux through reactions that produce this metabolite. This tests if a given metabolite production is feasible.
- Byproduct under Varying Substrate: Modified an exchange reaction by a series of fractions and, at each value, computes the "production envelope" (or phenotype phase plane[18], using CobraPy) between the byproduct section and the primary

growth objective. This identifies the substrate threshold at which the GSMN is forced to secrete byproducts while still maintaining growth.

- Production Dependency: Knocks out all reactions producing a given metabolite and measures the production of an alternative metabolite before and after the knockouts. This tests the metabolic dependency between two metabolite species.

Objective Flexibility

- Objective Shift: This replaces the GSMN's original objective function, simulates the network, and validates that the selected reaction(s)/pathway falls within an expected range. This captures the model's ability to re-route flux when the metabolic goal changes.

Multi-Model Comparison

- Comparative Flux Capability: Computes the range of possible flux for a given pathway in two models and counts the number of reactions that satisfy an expected direction between them (e.g., that the flux in the first model exceeds the flux in the second). This asserts relative pathway activity between the two cell types, but not its quantitative relationship.
- Comparative Nutrient Response: This applies the same boundary conditions to two models across a series of fractions, compares flux ratios for the given reaction(s)/pathway, and scores the percentage of comparisons that meet the provided thresholds.

Model Transformations

Transformations are reusable edits applied to a GSMN prior to a validation test. Each test is applied reversibly, meaning no model is permanently modified.

- Change Objective: Redefines the GSMN's goal (e.g., optimizing for ATP production as opposed to biomass)
- Set Reaction Bounds: Sets the lower and/or upper bounds of one or more reactions
- Add Exchange Reaction: Introduces a new exchange boundary condition for a metabolite
- Add Sink Reaction: Introduces a new reversible drain for a metabolite
- Add Demand Reaction: Introduces a new irreversible drain for a metabolite
- Add Internal Reaction: Introduces a new internal reaction
- Add Metabolite: Introduces a new metabolite
- Delete Genes: Removes genes and the reactions they code for
- Restrict to Metabolite Uptake: Closes the uptake of every macronutrient except one(s) being tested; vitamins, cofactors, and ions are also unmodified.
- Unify Exchanges: Sets all boundary conditions in a model to a common set; this is used to place two models on an equal "nutritional footing" before cross-model comparisons.

Flux Sampling

Flux sampling estimates the distribution of feasible reaction fluxes within that model's potential solution space, yielding a per-reaction flux distribution. Each reconstructed GSMN was sampled as follows. Models were loaded with CobraPy[7] and sampled via OptGpSampler[19]. For every model, 128 flux samples were drawn using a thinning factor of 2,000 steps to reduce correlation between samples. The original OptGpSampler algorithm saw convergence for the recon1 reference network at just 50 samples, indicating the choice of 2,000 is far above the minimum. Official convergence tests were not performed. The raw samples were saved as (1) a "Sample Store", where the full sample matrix was reshaped to a "long form", where one row existed per (reaction, flux sample) pair, and (2) a "Summary Store", where the samples were aggregated to one row per reaction, per donor, recording the mean flux for that reaction. The mean was selected because all flux vectors obtained via sampling are feasible, and as a result, there is a reduced need to exclude outliers by selecting a median value. By selecting the mean of the sampled donor-specific fluxes, the average feasible solution is represented. The Summary Store format is used in downstream regression analysis across donors. Both stores were written as Hive-partitioned Parquet datasets, with the Sample Store partitioned by cell type, age group, and superpathway, and the Summary Store partitioned by cell type alone. Hive partitioning allows column values to be encoded in the directory path itself, rather than stored as columns within the files. This provides a few benefits, including:

- The underlying PyArrow Python package can filter the dataset by directory name, thereby removing unnecessary data before any file content is read.
- Because the directory contains a column repeated across all files, a duplicate value is not saved in the file itself, significantly reducing the required file size.
- Because the flux samples for each model are independent, any output file(s) can be written without concern about overwriting existing data.

To make large runs resumable and avoid recomputation, each completed flux sample generates a completion file that records the cell type, donor ID, and the Python code used to generate that sample. Thus, only missing sampling results will be computed, or, if any component of the flux sampling pipeline is modified, the workflow will be executed in full.

Sampling was distributed across UNL's Holland Computing Center, which utilizes a SLURM job scheduler. One model was processed per job submission. A main "submission script" was created to manage independent sampling jobs. This manager queried all existing GSMN files and existing sampling records to identify models still requiring sampling. Only pending models were submitted as an independent job. SLURM limits the number of concurrent jobs a single user can submit to 1,000; because the entire submission consisted of 5,705 metabolic models, jobs were submitted in batches of 200.

Several factors can impact the outcome of flux sampling. Algorithm parameters, such as the thinning interval, determines the correlation between sampled points, while the number of processes splits the requested sample size into independent chains, which

alters individual chain length and the diversity of the resulting samples. If multiple sampling turns differ in the number of processes used, the resulting samples are not directly comparable. The geometry of the sampled solution space, however, often exerts a greater influence than these settings: broad default reaction bounds yield a high-dimensional, poorly conditioned polytope in which chains mix slowly, and any constraint imposed on the objective value defines a fundamentally different space rather than merely a subset of the original. Thermodynamically infeasible cycles introduce near-unbounded directions that further impede mixing, and retained blocked reactions inflate the dimensionality without contributing to the feasible space. Finally, apparent convergence at a given sample size is not evidence of convergence, and agreement of marginal distributions across independent seeds should be verified before drawing quantitative conclusions.

Statistical Testing

All analyses were performed per cell type. For a given cell type, the input was a matrix of donors (columns) and the mean of sampled reactions (rows).

Flux preprocessing and independent filtering

Reactions that could not carry an age signal were removed by independent filtering[20]. Two classes of reaction were discarded, (1) dead reactions, whose flux magnitude never exceeded the solver tolerance (10^{-6}) for any donor, and (2) near-constant reactions, for which a single flux value was shared by at least 95% of donors and whose spread of flux values was therefore too small to separate signal from sampling noise.

Following this, surviving reaction fluxes were z-score standardized within each reaction to a zero mean and unit variance across donors, so that each reaction was scaled relative to its own distribution. This allows downstream analyses to compare reaction fluxes across reactions with different absolute magnitudes and variances, ensuring that any patterns identified reflect the underlying biology rather than being dominated by reactions with inherently larger or more variable flux ranges. Reactions were mapped to pathways using Recon3D[6] as the reference genome-scale metabolic model. Pathway-level flux values were obtained by averaging the per-reaction z-scores within each pathway across donors and used in Kendall's tau statistic calculation.

Age Association by Kendall's Tau

To test if flux correlates with age, Kendall's rank correlation was calculated between the feature value and age across donors, at both the reaction and pathway levels.[21] Kendall's tau is a non-parametric measure of monotonic association, and it does not make assumptions about the underlying flux distribution. It is calculated by comparing all donor pairs, where concordant pairs are those in which the ranking of donor age matches the ranking of flux values for that reaction, while discordant pairs show the opposite ordering. The tau statistic is computed from the difference between concordant and discordant pairs, divided by the total number of pairs. A positive tau indicates flux increases with age, while a negative tau indicates flux decreases with age. Kendall's tau was chosen over Spearman's rho because it is a better measure for the small donor counts and frequent ties in flux values in this dataset. Two-sided p-values were calculated using

SciPy with default settings.[22] A limitation of Kendall's Tau is its monotonic association, meaning it is only able to discover a correlation for a consistent directional change, such as values that start low (e.g., donor age and flux) and end high, or vice versa. If a correlation is better described by a quadratic (or other) function, Kendall's tau is unable to identify this relationship. P-values were not adjusted for multiple comparisons, and the results presented here should be noted as nominal associations with age-related correlations.

Results

Single-cell RNA Sequencing

Figure 8 shows the number of labeled cells recovered for each donor and cell type from the Terekhova dataset. Counts are z-score standardized within each cell type. This scaling allows donors to be compared within a cell type but does not allow counts to be compared across cell types. A quantitative investigation of the sequencing and cell-type composition is available in Terekhova et al[12]. Supplementary Table 2 lists the total cell counts for each cell type and all donors.

Within the CD8+ T cell lineage, older donors tend to contribute fewer cells, with the most visibility in the naive CD8+ subset. This is consistent with the well-documented reduction in the naive T cell pool with age, primarily driven by reduced thymic output and a shift of these cells toward memory and effector states.[26,27] The "Unknown" population also shows a downward trend among older donors, suggesting a potentially

reduced set of intermediate cell types with age. Atypical memory B cells are more abundant in older donors. An age-associated expansion of this B cell subset (atypical, age-associated, or CD11c+ G cells) is a recurring finding in the literature.[28–30] The Vd2 gamma delta T cell subsets also reduce with age, while the Vd1 subset is tends to show a less consistent pattern.[31,32]

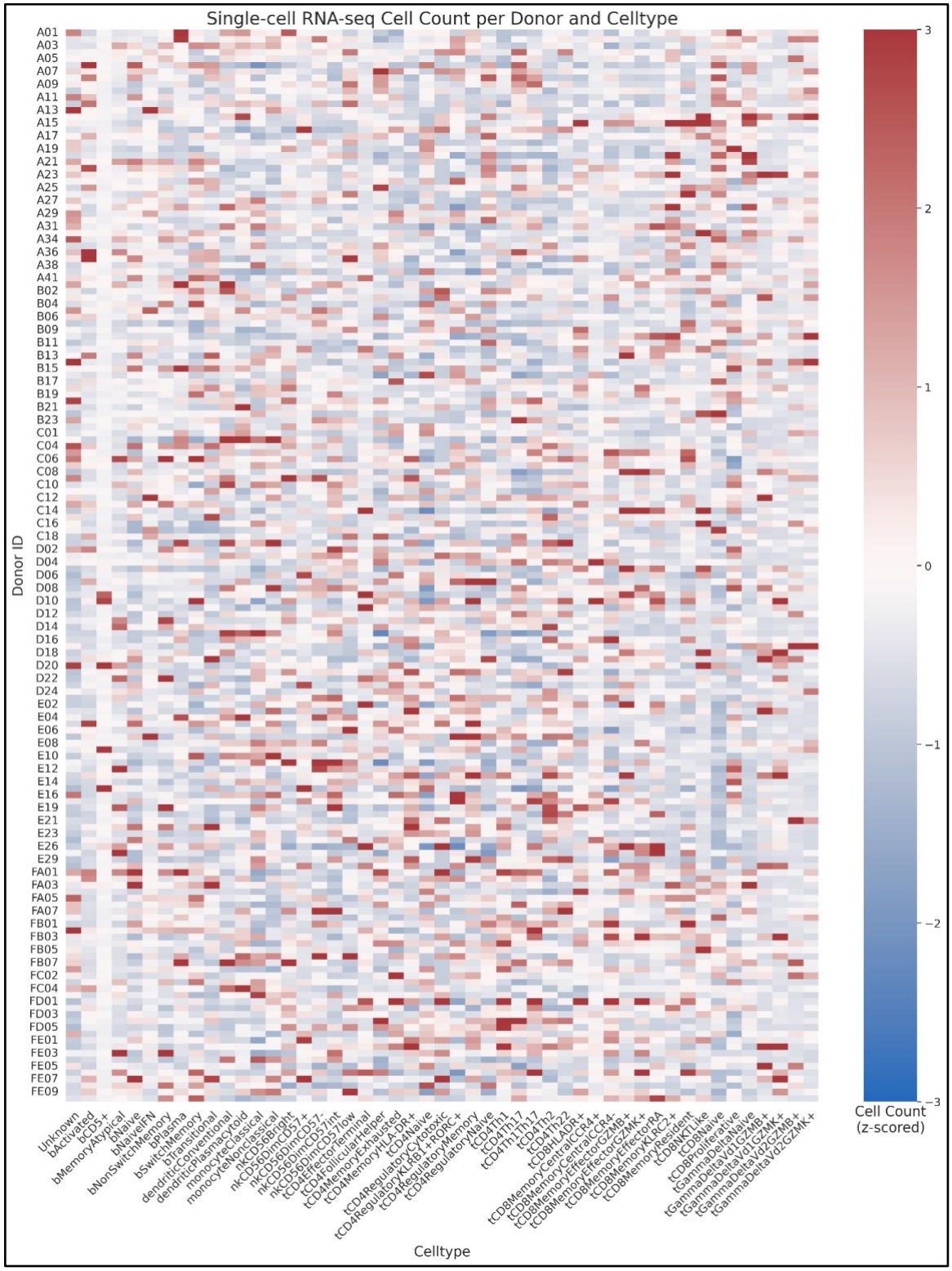

Single-cell RNA-seq Cell Count per Donor and Celltype
Donor ID
Celltype
Cell Count
(z-scored)
3
2
1
0
−1
−2
−3
Unknown
bActivated
bCD5+
bMemoryAtypical
bNaive
bNaiveIFN
bNonSwitchMemory
bPlasma
bSwitchMemory
bTransitional
dendriticConventional
dendriticPlasmacytoid
monocyteClassical
monocyteNonclassical
nkCD56BrightCD57-
nkCD56DimCD57+
nkCD56DimCD57-
nkCD56DimCD57Int
nkCD56DimCD57low
tCD4EffectorTerminal
tCD4FollicularHelper
tCD4MemoryExhausted
tCD4MemoryHLA-DR+
tCD4Naive
tCD4RegulatoryCytotoxic
tCD4RegulatoryKLRB1+RORC+
tCD4RegulatoryMemory
tCD4RegulatoryNaive
tCD4Th1
tCD4Th1Th17
tCD4Th17
tCD4Th2
tCD4Th22
tCD8HLADR+
tCD8MemoryCentralCCR4+
tCD8MemoryCentralCCR4-
tCD8MemoryEffectorGZMK+
tCD8MemoryEffectorGZMB+
tCD8MemoryEffectorRA
tCD8MemoryKLRC2+
tCD8MemoryResident
tCD8NKTLike
tCD8Naive
tCD8Proliferative
tGammaDeltaNaive
tGammaDeltaVd1GZMB+
tGammaDeltaVd1GZMK+
tGammaDeltaVd2GZMB+
tGammaDeltaVd2GZMK+

Figure 8: Heatmap displaying z-scored cell counts across various cell types (x-axis) for individual donors (y-axis). The color scale indicates relative abundances of each cell type per donor, ranging from lower relative counts (blue) to higher relative counts (red).

Genome-scale Metabolic Modeling

To account for differences in baseline reaction counts across donors, the raw reaction count were z-scored within each individual cell type column. This allows for a statistically meaningful comparison of reaction counts for a specific cell type across all donor samples. Cell-type comparisons (across rows) are not capable due to the z-score normalization process. Supplementary Table 4 shows a variety of model statistics across all cell types, including the number of models reconstructed for each cell type, average reaction, gene, and metabolite count, blocked reactions, and model feasibility. Feasibility was defined as an objective function capable of carrying flux greater than $1e^{-6}$, which is the default sensitivity for most linear programming solvers.[33]

Qualitative analysis reveals potential age-related patterns on reaction count alone. Within the effector terminal CD4+ T cells, reaction counts are consistently higher (indicated by more red values) among donors in older cohorts compared to the lower counts (lighter gray and blue hues) observed in younger cohorts. Furthermore, several CD4+ cell type columns (naive regulatory, memory, and effector terminal CD4+) show a scarcity of donor-specific data points near the population mean (e.g., z=0). Instead, the data is distributed as a divergent, classically bimodal pattern. Donors appear to fall into a “low-reaction” or “high-reaction” group. This observation suggests that these specific CD4+

populations may not follow a single, unified normal distribution across the entire population, but rather exist as distinct sub-populations.

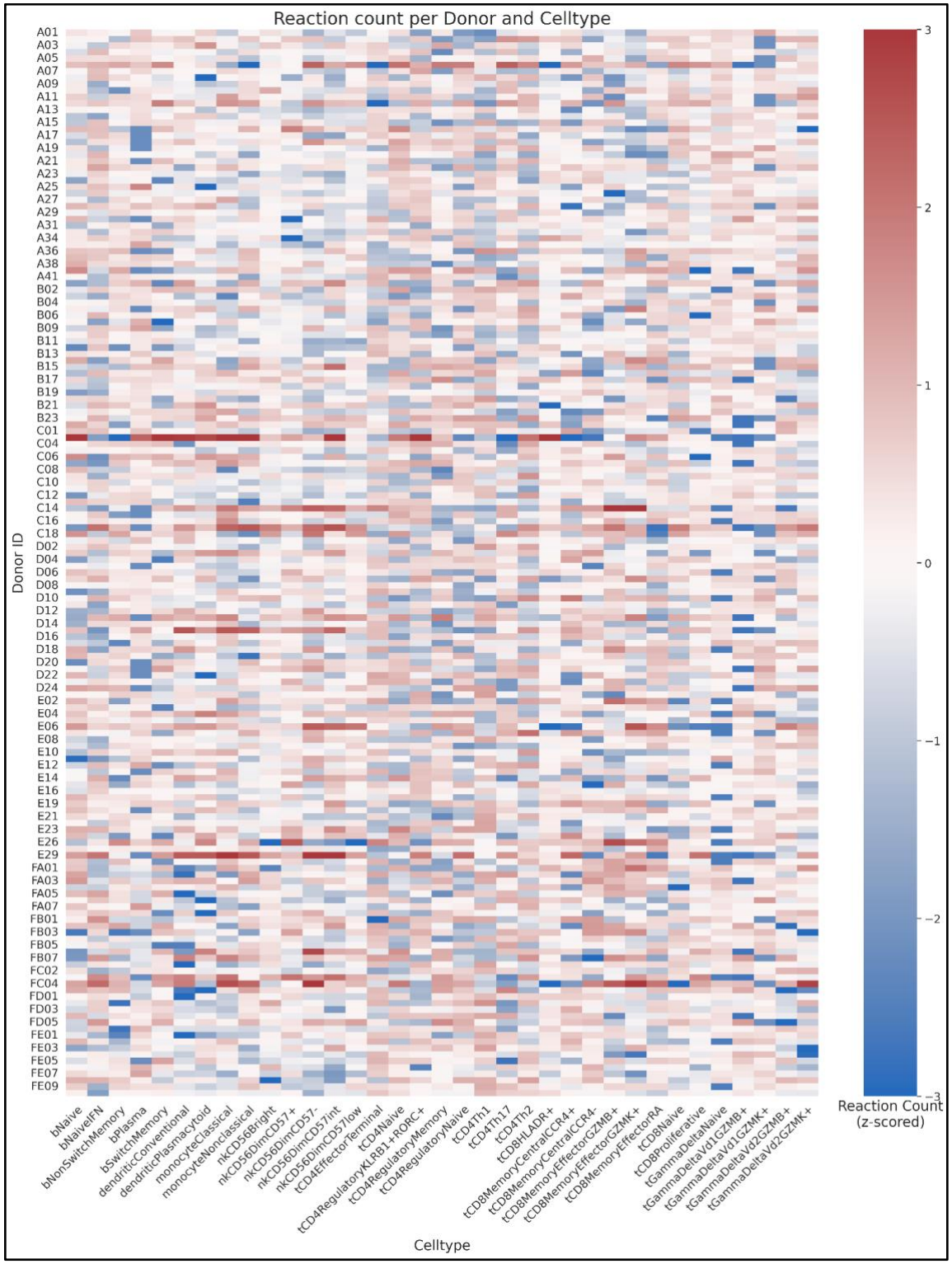

Reaction count per Donor and Celltype
Donor ID
Celltype
Reaction Count
(z-scored)
3
2
1
0
−1
−2
−3
bNaive
bNaiveIFN
bNonSwitchMemory
bPlasma
bSwitchMemory
dendriticConventional
dendriticPlasmacytoid
monocyteClassical
monocyteNonclassical
nkCD56Bright
nkCD56DimCD57+
nkCD56DimCD57-
nkCD56DimCD57low
tCD4EffectorTerminal
tCD4Naive
tCD4RegulatoryKLRB1+RORC+
tCD4RegulatoryMemory
tCD4RegulatoryNaive
tCD4Th1
tCD4Th17
tCD4Th2
tCD8HLADR+
tCD8MemoryCentralCCR4+
tCD8MemoryCentralCCR4-
tCD8MemoryEffectorGZMB+
tCD8MemoryEffectorGZMK+
tCD8MemoryEffectorRA
tCD8Naive
tCD8Proliferative
tGammaDeltaNaive
tGammaDeltaVd1GZMB+
tGammaDeltaVd1GZMK+
tGammaDeltaVd2GZMB+
tGammaDeltaVd2GZMK+

Figure 9: Heatmap showing z-scored reaction counts across immune cell types for individual donors. The color scale indicates relative abundance of each reaction count per donor.

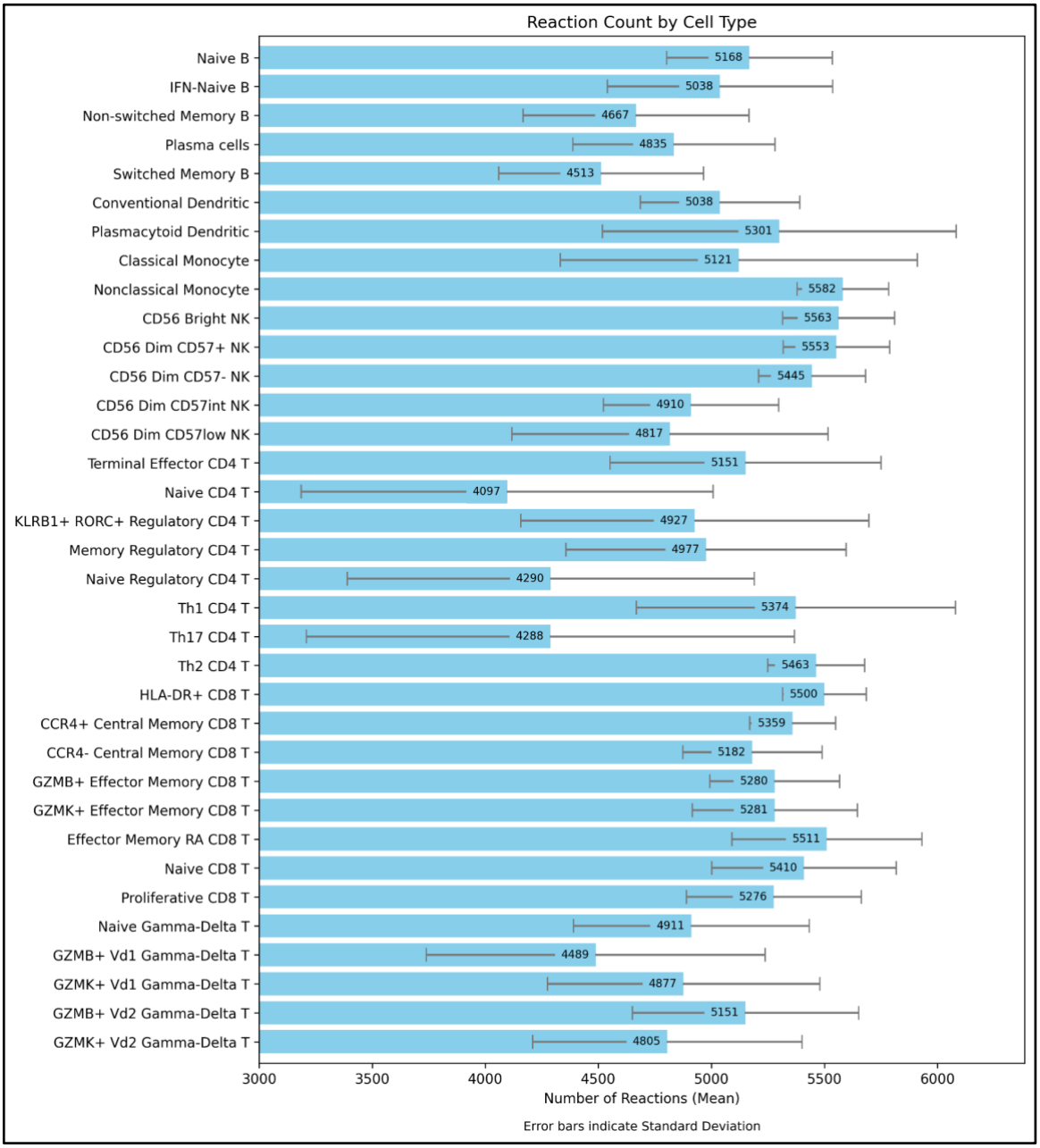


Figure 10: The number of average reactions across each cell type for all donors. The error bars represent the standard deviation for the reaction count.

Summary of Cell-type, Donor-specific Metabolic Models

Supplementary Figures 1 through 23 (attached separately) show the full validation report for metabolic models across all donors, summarized in Table 4 by cell type; a total of 77,205 individual validations were tested. For the validations across each cell type, a number of them are failing for every donor. These universal failures indicate incorrect parameterization of the validation itself rather than biological flaws or reconstruction errors in the models. Further work is needed to correct the parameterization of these validations to ensure they are testing the expected biology of the given cell type. For example, some validations check the flux through an ATP sink reaction. However, if this reaction is not explicitly included in the model, the validation will fail. This is not a result of incorrect biology, but an error in the validation itself. Thus, confirming the biology being validated is an additional, important step that must be completed.

Table 4: The number of validations across each cell type and the total count of passing and failing validations.

Cell Type	Validations per Model	Passing	Failing
Naive B	16	2,655	159
IFN-naive B	16	2,712	96
Non-switch Memory B	19	2,358	692
Switch Memory B	13	3,819	442
Plasma	31	2,152	2,588
Conventional Dendritic	22	2,262	1,202
Plasmacytoid Dendritic	22	2,273	1,311
Classical Monocyte	19	5,325	980
Nonclassical Monocyte	21	4,329	3,168

Cell Type	Validations per Model	Passing	Failing
CD56bright Natural Killer	42	3,931	2,085
CD56dim/CD57- Natural Killer	13	924	1,128
CD56dim/CD57+ Natural Killer	13	889	1,117
CD56dim/CD57int Natural Killer	13	801	1,194
CD56dim/CD57low Natural Killer	13	827	1,185
Naive CD4+ T	8	283	1,001
Th1 CD4+ T	8	1,015	282
Th17 CD4+ T	8	822	464
Th2 CD4+ T	8	978	296
HLADR+ CD8+ T	20	711	2,350
Memory Central CCR4+ CD8+ T	39	1,453	4,645
Memory Central CCR4- CD8+ T	48	1,171	6,435
Naive CD8+ T	14	2,364	1,863
Proliferative CD8+ T	20	758	2,355

Pathway-specific Age Correlations

Figure 11 summarizes, for each metabolic pathway, the number of cell types in which pathway-level flux showed a significant association with donor age as determined by Kendall's Tau statistic. Figure 12 performs the inverse, and for each cell type, summarizes the number of pathways significantly correlated with age. Pathways with a single significant cell type are excluded from the figure for brevity. Bars extending to the right indicate cell types with an increasing flux correlated with age, and bars extending left indicate decreasing flux correlated with age.

A few pathways showed significant age associations in many cell types, whereas most pathways were affected in only a small number. Tetrahydrobiopterin and vitamin A metabolism showed the most consistently increasing pattern, with flux correlating to age in five and six cell types, respectively. Among pathways with correlations predominantly increasing with age, the citric acid cycle, cholesterol metabolism, cytochrome metabolism, and androgen and estrogen synthesis and metabolism were each affected in four cell types. In the opposite direction, vitamin B2 metabolism stood out as the only pathway with exclusively decreasing flux, affecting five cell types and no increasing ones. Hyaluronan metabolism and phenylalanine metabolism were also predominantly decreasing, affecting four cell types each in that direction.

A number of pathways showed changes in both directions across different cell types, indicating that the direction of the age association is cell-type-dependent, rather than a single coordinated shift of a pathway across multiple cell types. Blood group synthesis (increasing in four and decreasing in three cell types), nucleotide interconversion (increasing in three and decreasing in three cell types), and intracellular demand (increasing in three and decreasing in three cell types) are examples where increasing and decreasing associations occur in roughly equal numbers of cell types.

Flux changes were concentrated in a subset of cell types. Naive gamma delta T cells, naive CD8+ T cells, terminal effector CD4+ T cells, Th1 CD4+ T cells, and Vd2 GZMK+ gamma delta T cells each had the largest number of significant pathways, while many other cell types only carried several significantly correlated pathways. Naive CD8+

T cells and terminal effector CD4+ T cells showed almost entirely increasing flux correlations across their significant pathways (20 of 22 and 18 of 20, respectively), whereas naive gamma delta T cells showed predominantly decreasing flux correlations (18 of 24). The same naive-versus-effector distinction therefore separates cell types by the direction of their age association, not only the number of pathways affected.

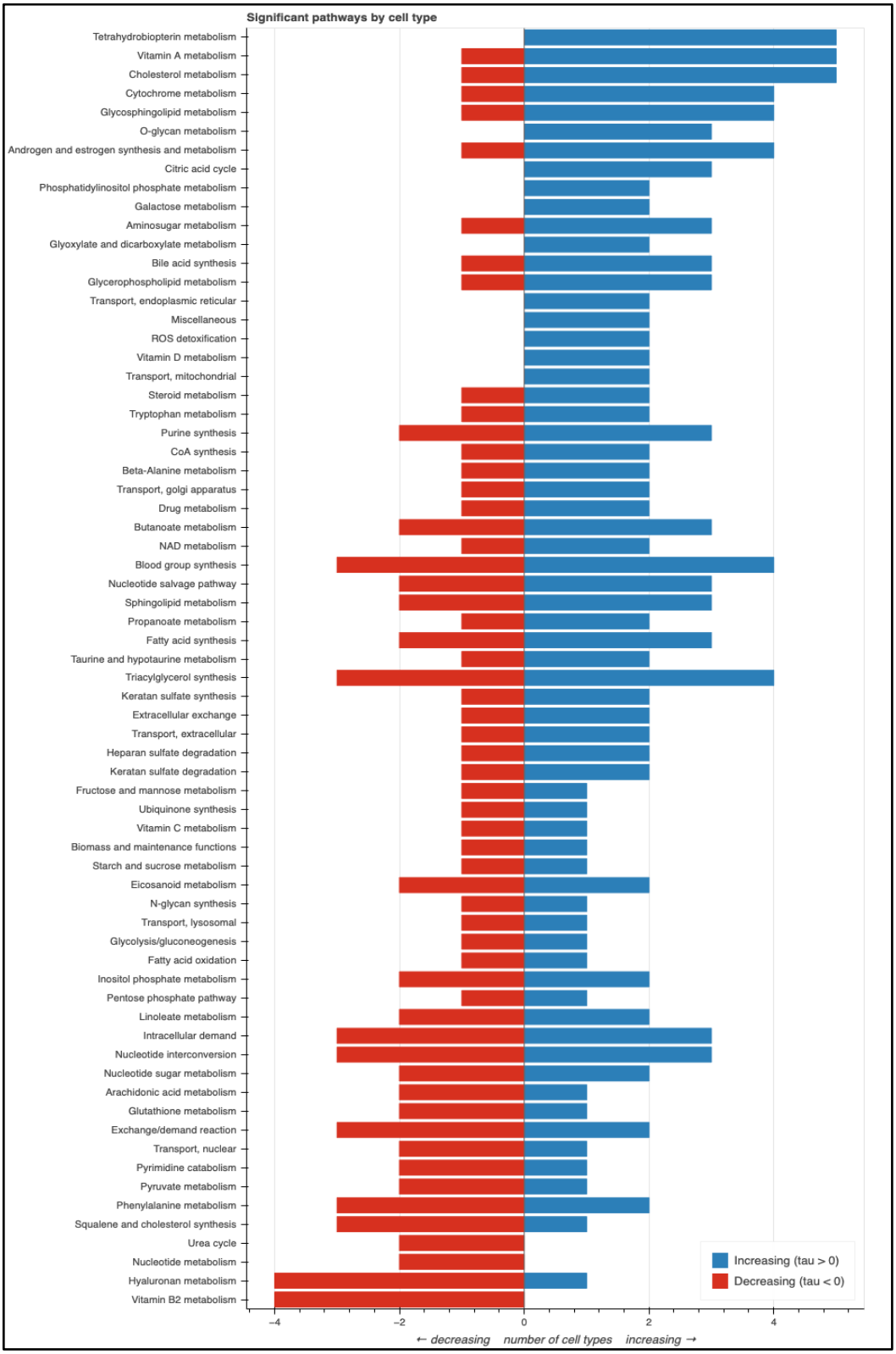

Significant pathways by cell type
Tetrahydrobiopterin metabolism
Vitamin A metabolism
Cholesterol metabolism
Cytochrome metabolism
Glycosphingolipid metabolism
O-glycan metabolism
Androgen and estrogen synthesis and metabolism
Citric acid cycle
Phosphatidylinositol phosphate metabolism
Galactose metabolism
Aminosugar metabolism
Glyoxylate and dicarboxylate metabolism
Bile acid synthesis
Glycerophospholipid metabolism
Transport, endoplasmic reticular
Miscellaneous
ROS detoxification
Vitamin D metabolism
Transport, mitochondrial
Steroid metabolism
Tryptophan metabolism
Purine synthesis
CoA synthesis
Beta-Alanine metabolism
Transport, golgi apparatus
Drug metabolism
Butanoate metabolism
NAD metabolism
Blood group synthesis
Nucleotide salvage pathway
Sphingolipid metabolism
Propanoate metabolism
Fatty acid synthesis
Taurine and hypotaurine metabolism
Triacylglycerol synthesis
Keratan sulfate synthesis
Extracellular exchange
Transport, extracellular
Heparan sulfate degradation
Keratan sulfate degradation
Fructose and mannose metabolism
Ubiquinone synthesis
Vitamin C metabolism
Biomass and maintenance functions
Starch and sucrose metabolism
Eicosanoid metabolism
N-glycan synthesis
Transport, lysosomal
Glycolysis/gluconeogenesis
Fatty acid oxidation
Inositol phosphate metabolism
Pentose phosphate pathway
Linoleate metabolism
Intracellular demand
Nucleotide interconversion
Nucleotide sugar metabolism
Arachidonic acid metabolism
Glutathione metabolism
Exchange/demand reaction
Transport, nuclear
Pyrimidine catabolism
Pyruvate metabolism
Phenylalanine metabolism
Squalene and cholesterol synthesis
Urea cycle
Nucleotide metabolism
Hyaluronan metabolism
Vitamin B2 metabolism
Increasing (tau > 0)
Decreasing (tau < 0)
−4
−2
0
2
4
← decreasing number of cell types increasing →

Figure 11: The number of cell types that have significant correlations for each pathway.

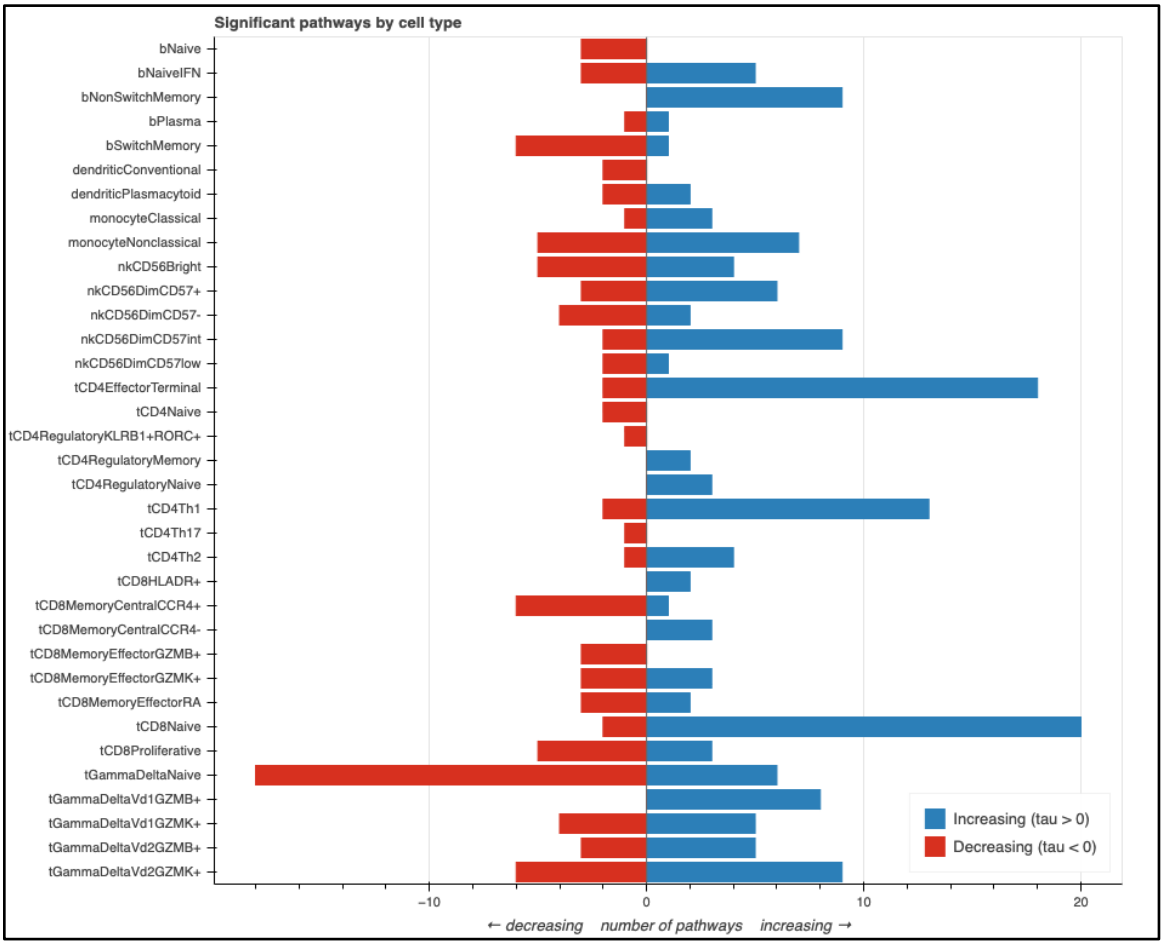


Figure 12: The number of pathways that have significant correlations for each cell type

Pathway Investigation

The following pathways were chosen because they have a positive correlation with age across the most cell types (tetrahydrobiopterin metabolism: 5 increasing; Vitamin A metabolism: 6 increasing, 1 decreasing) and a negative correlation with age across the most cell types (Vitamin B2 metabolism: 5 decreasing; Hyaluronan metabolism: 4 decreasing, 1 increasing).

Identifying correlations between age and cell type in donor-specific models is challenging because we cannot control an individual's diet, day-to-day or chronic stress, or lifestyle habits. As a result, the correlation coefficients are low to moderate (Kendall's Tau from ~0.1 to ~0.2), yet significant ($p < 0.05$). Despite this, a previous review describes these correlations as "poor" to "fair" for medicine-related statistical tests.[34]

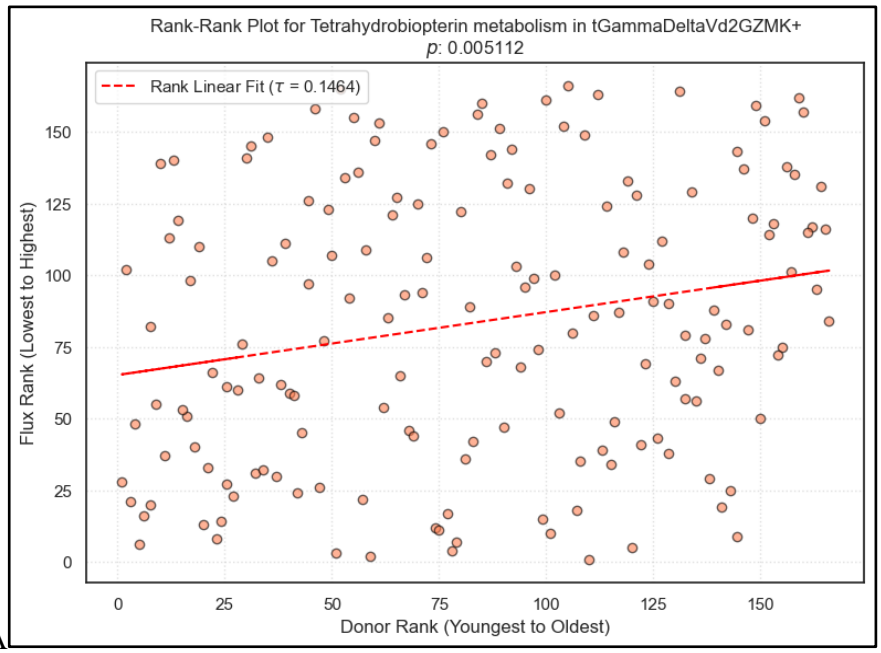
Tetrahydrobiopterin and Phenylalanine Metabolism

Tetrahydrobiopterin (BH4) is an essential cofactor for phenylalanine degradation, neurotransmitter synthesis, and nitric oxide production, synthesized by GTP cyclohydrolase 1 (GCH1).[35–37] Previous studies have shown that phenylalanine degradation declines with age, and supplementing older mice with BH4 reversed some age-related effects, specifically those related to cellular senescence, redox, and epigenetics, in the heart and liver.[38]

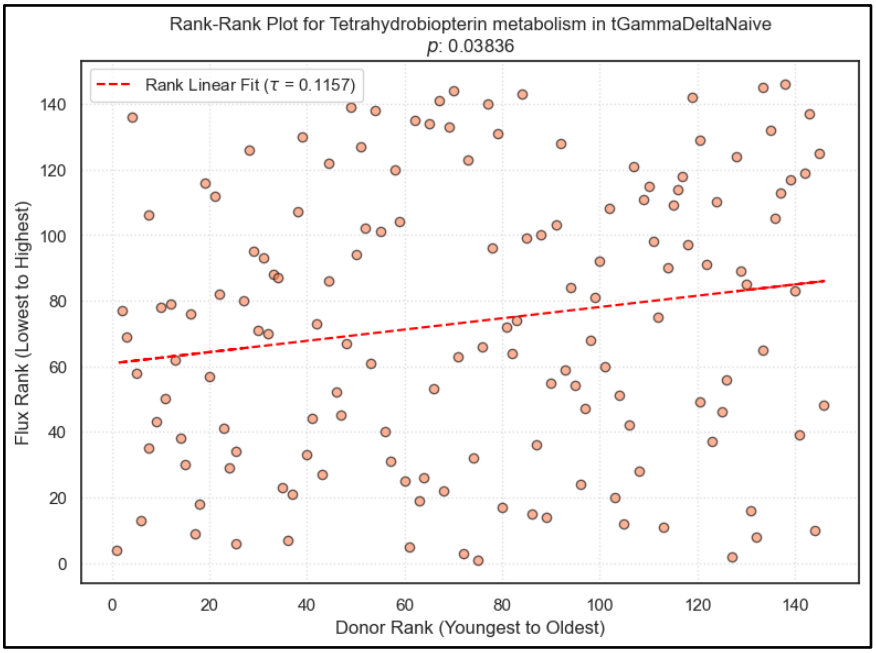
VD2 GZMK+ and naive gamma delta T cells show a positive correlation between flux and age in both tetrahydrobiopterin and phenylalanine metabolism (Figure 13 Panels A and B, Figure 14 Panels A and B). Naive CD8+ T cells and HLADR+ CD8+ T cells have a positive correlation in tetrahydrobiopterin alone (Figure 13 Panels C and D). Classical monocytes also show a positive correlation with age only in tetrahydrobiopterin metabolism (Figure 13 Panel E) In contrast, switch memory and naive B cells, along with CD56dim/CD57low natural killer cells, show a negative correlation between flux and age in phenylalanine metabolism alone (Figure 14 Panels C, D, and E).

These metabolic models do not include metabolite concentrations (only reaction flux values), and concentrations were not reported in the original Terekhova study. Future work, as a result, could include testing the following hypothesis: because aging reduces hepatic phenylalanine clearance and increases circulating concentrations, circulating lymphocytes will be exposed to higher phenylalanine concentrations. If age-related increases in circulating phenylalanine do not translate into altered intracellular phenylalanine metabolism, which remains uncertain given that its uptake depends on antiporter activity, an observed increase in GCH1 expression and de novo BH4 synthesis are induced by interferon-gamma and interleukin-1 beta alone, rather than plasma phenylalanine availability.[39,40] Although these cytokines are not directly measured in the RNA-seq data or represented in the metabolic models, their concentrations are known to increase with age as a part of inflammaging.[41] This provides a potential mechanistic explanation for the positive correlation between BH4 pathway flux observed here, as sustained inflammatory signaling in older donors increases GCH1-mediated synthesis, maintaining the BH4 cofactor supply for downstream inducible nitric oxide synthase (iNOS) during prolonged immune activation.[39,40] Whether phenylalanine substrate availability makes an additional contribution to this iNOS signal remains an open question. Answering it would require donor-matched plasma phenylalanine measurements and direct assessment of SLC7A5/LAT1 expression, activity, or localization through mRNA sequencing, substrate uptake experiments, and flow cytometry, respectively.

If the observed flux correlation were shown to be primarily GCH1-substrate-driven rather than a consequence of cytokine-induced GCH1 expression, the mechanistic interpretation would need to be revised. Cytokine-induced GCH1 upregulation is a response to the signaling environment (e.g., increased levels of IFN-gamma or IL-1). A substrate-driven increase, however, is independent of signaling; BH4 synthesis would rise as a result of increased phenylalanine availability, not because the cell has been signaled to become active. This distinction has direct consequences for downstream nitric oxide production, because BH4 is chemically unstable and readily oxidized into dihydrobiopterin (BH2), which cannot support iNOS catalysis, and instead causes the enzyme to produce superoxide rather than nitric oxide. In the signaling scenario, BH4 is produced in coordination with iNOS induction and is consumed in the process, limiting the opportunity for reactive oxygen generation. In the substrate-driven increase, BH4 may accumulate without a corresponding increase in iNOS activity, raising the likelihood of oxidation to BH2 (uncoupled from iNOS), thereby increasing intracellular oxidative stress rather than supporting immune function/activation. In the first, immune cells in older individuals are actively sustaining nitric oxide-dependent responses as a consequence of a chronic inflammatory environment; in the second, immune cells accumulate an oxidative burden as a consequence of age-related hepatic metabolic decline, with the age correlation in flux demonstrating a systemic amino acid dysregulation rather than a component related to the immune cell's specific biology.

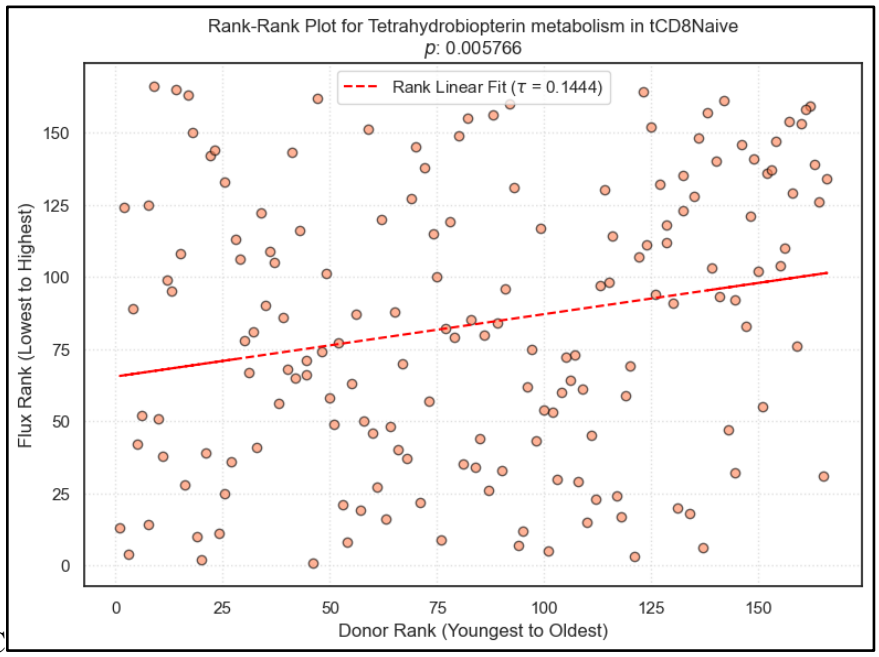

Rank-Rank Plot for Tetrahydrobiopterin metabolism in tGammaDeltaVd2GZMK+
p: 0.005112
Rank Linear Fit (τ = 0.1464)
Flux Rank (Lowest to Highest)
Donor Rank (Youngest to Oldest)
0
25
50
75
100
125
150

A

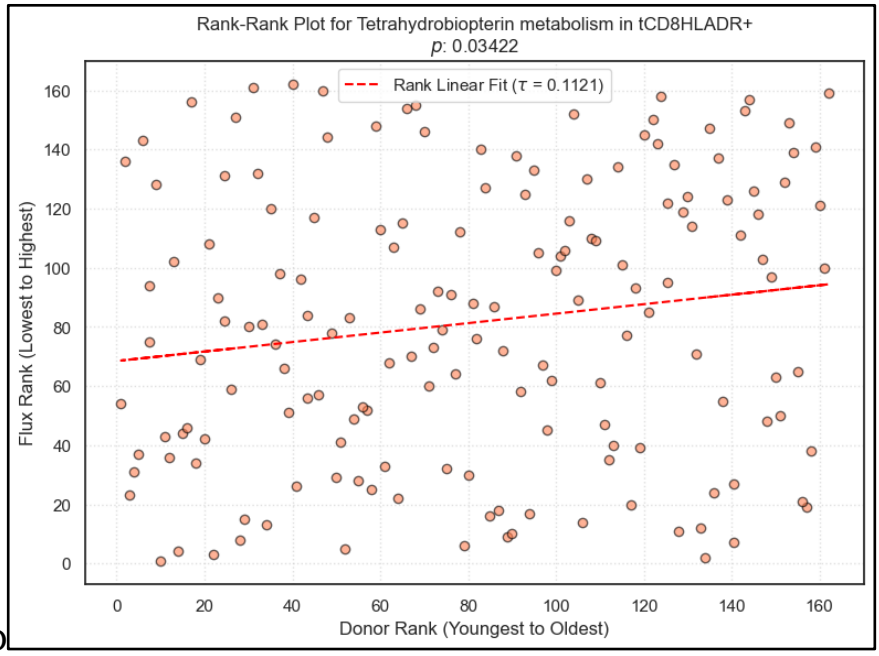

Rank-Rank Plot for Tetrahydrobiopterin metabolism in tGammaDeltaNaive
p: 0.03836
Rank Linear Fit (τ = 0.1157)
Flux Rank (Lowest to Highest)
Donor Rank (Youngest to Oldest)
0
20
40
60
80
100
120
140

B

C


Rank-Rank Plot for Tetrahydrobiopterin metabolism in tCD8Naive
p: 0.005766
Rank Linear Fit (τ = 0.1444)
Flux Rank (Lowest to Highest)
Donor Rank (Youngest to Oldest)
0
25
50
75
100
125
150

D


Rank-Rank Plot for Tetrahydrobiopterin metabolism in tCD8HLADR+
p: 0.03422
Rank Linear Fit (τ = 0.1121)
Flux Rank (Lowest to Highest)
Donor Rank (Youngest to Oldest)
0
20
40
60
80
100
120
140
160

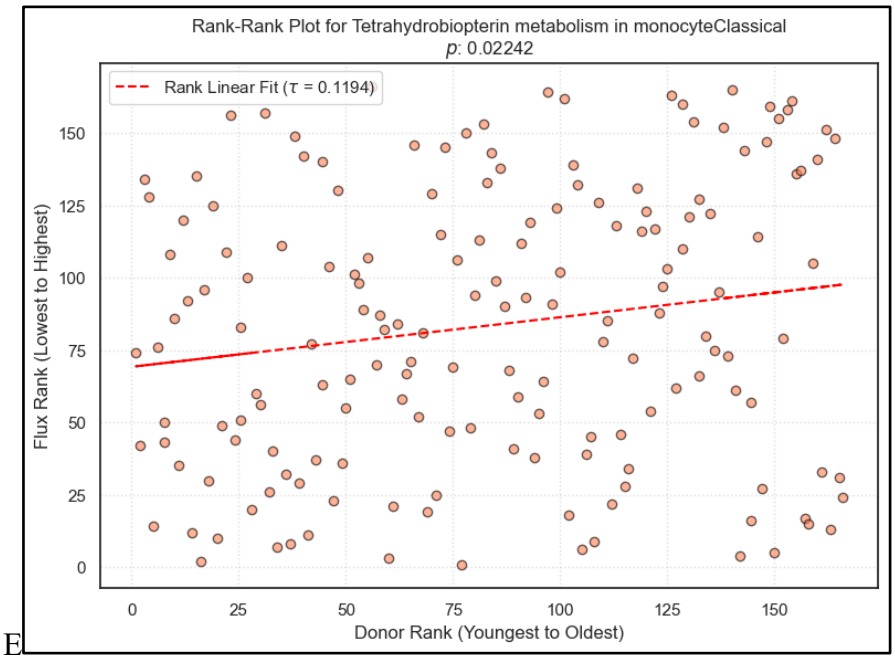
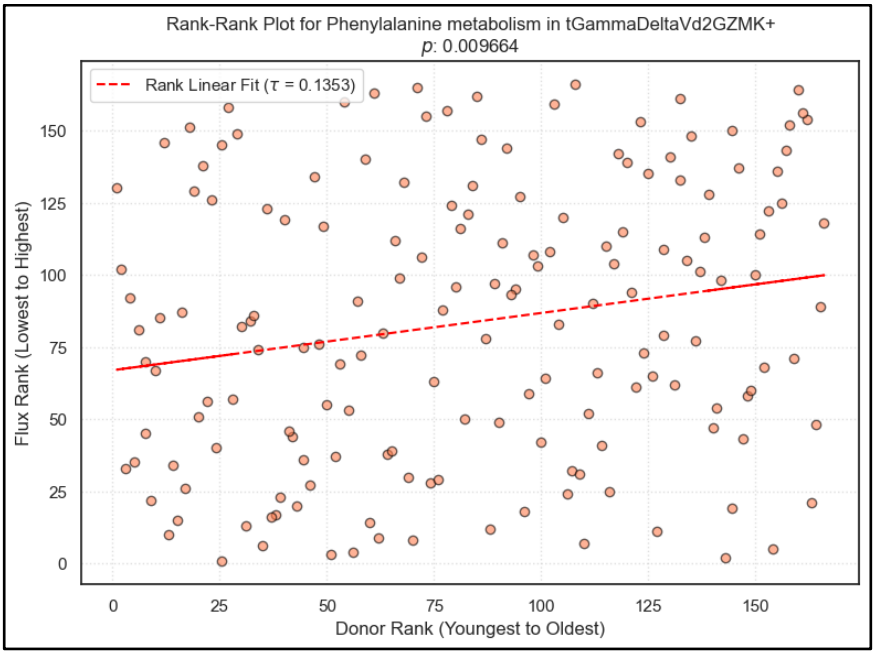
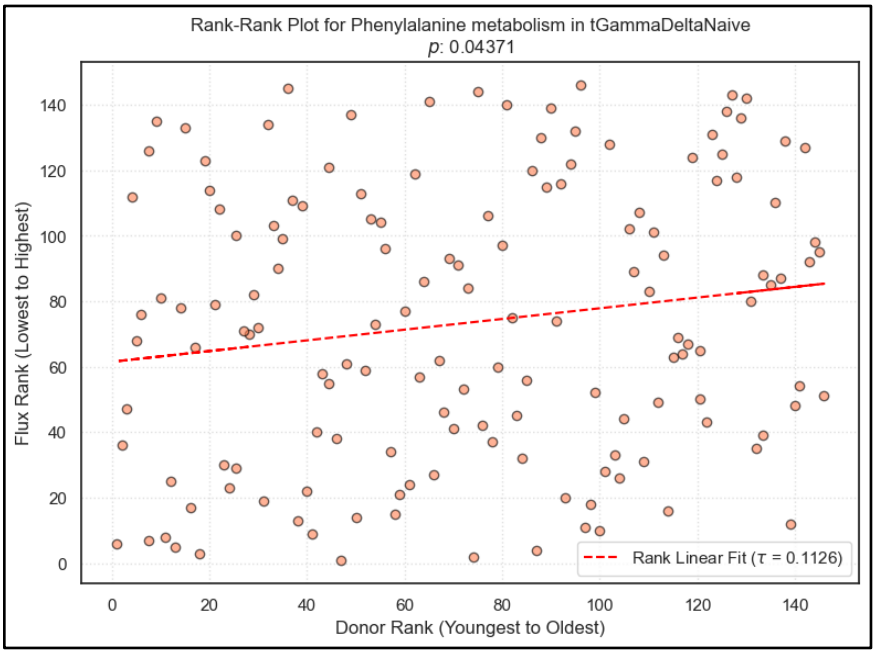
E 

Figure 13: Rank-Rank plot for Tetrahydrobiopterin Metabolism for cell types where it was indicated as significant ($p < 0.05$). Panel A) Vd2 GZMK+ gamma delta T cells (tau: 1.464e-01; p: 5.112e-03). Panel B) Naive gamma delta T cells (tau: 1.157e-01, p: 3.836e-02). Panel C) Naive CD8+ T cells (tau: 1.444e-01, p: 5.766e-03). Panel D) HLADR+ CD8+ T cells (tau: 1.121e-01, p: 3.422e-02). Panel E) Classical monocytes (tau: 1.194e-01, p: 2.242e-02).

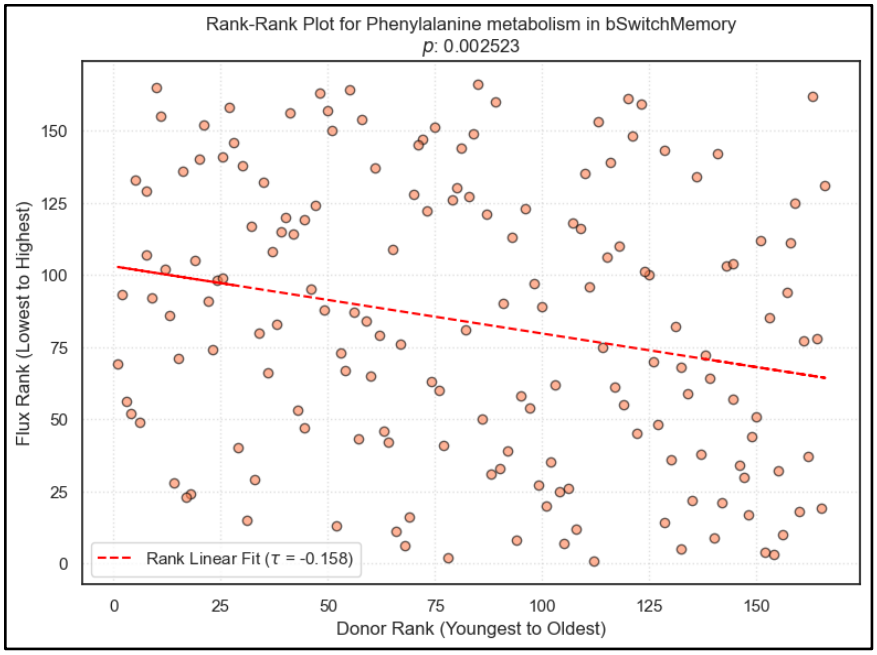
A


Rank-Rank Plot for Phenylalanine metabolism in tGammaDeltaVd2GZMK+
p: 0.009664
Rank Linear Fit (τ = 0.1353)
Flux Rank (Lowest to Highest)
Donor Rank (Youngest to Oldest)
0
25
50
75
100
125
150

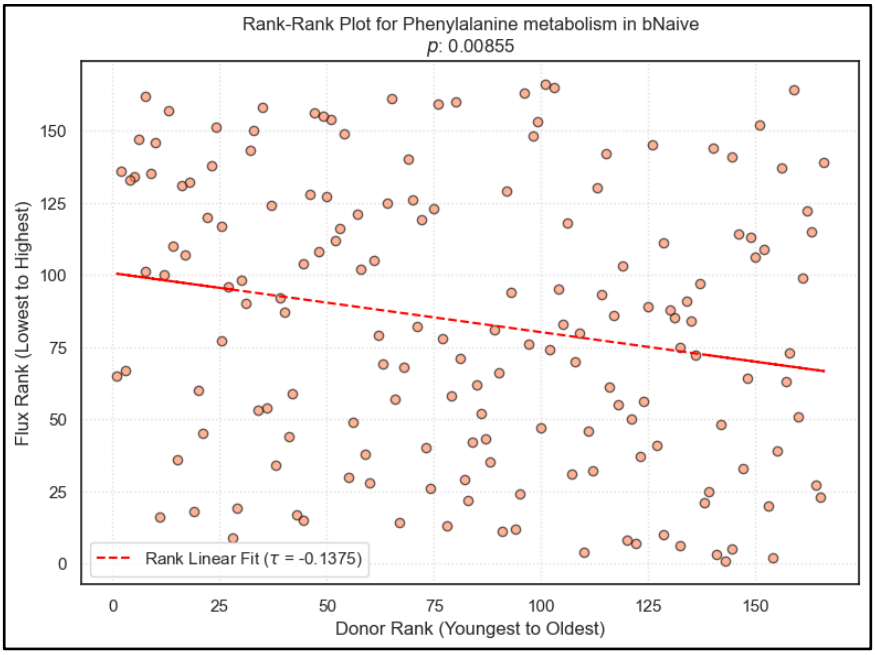
B


Rank-Rank Plot for Phenylalanine metabolism in tGammaDeltaNaive
p: 0.04371
Rank Linear Fit (τ = 0.1126)
Flux Rank (Lowest to Highest)
Donor Rank (Youngest to Oldest)
0
20
40
60
80
100
120
140

C


Rank-Rank Plot for Phenylalanine metabolism in bSwitchMemory
p: 0.002523
Flux Rank (Lowest to Highest)
Donor Rank (Youngest to Oldest)
Rank Linear Fit (τ = -0.158)
0
25
50
75
100
125
150

D



E

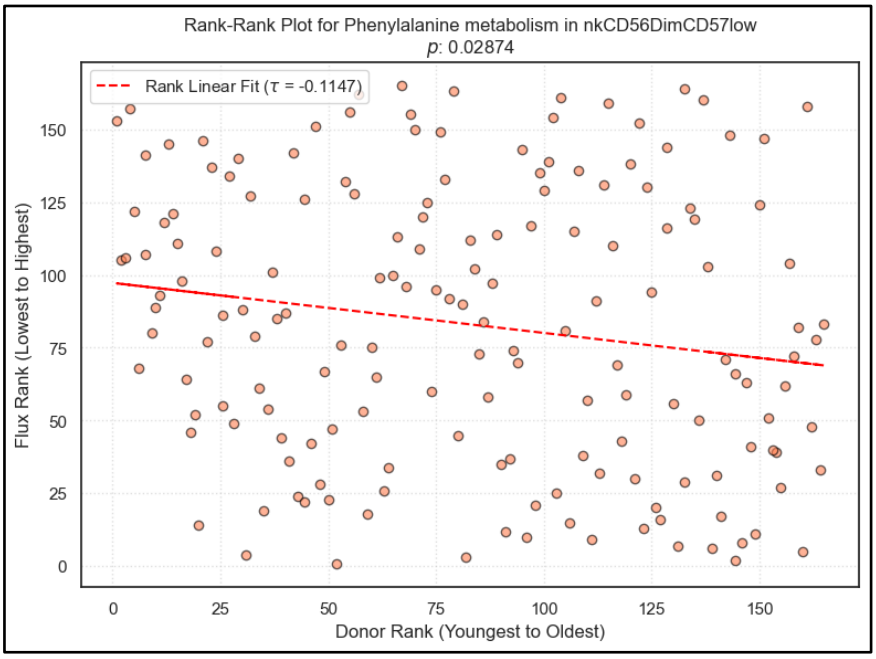


Figure 14: Rank-rank Kendall correlation coefficient plots for phenylalanine metabolism in cell types it was indicated as significant ($p < 0.05$). Panel A) VD2 GZMK+ gamma delta T cells (tau: 1.353e-01; p: 9.664e-03). Panel B) Naive gamma delta T cells (tau: 1.126e-01; p: 4.371e-02). Panel C) Switch memory B cells (tau: -1.580e-01; p: 2.523e-03). Panel D) Naive B cells (tau: -1.375e-01; p: 8.550e-03). Panel E) CD56dim/CD57low natural killer cells (tau: -1.147e-01; p: 2.874e-02).

Vitamin A Metabolism

Vitamin A is an essential fat-soluble metabolite that includes retinol and beta-carotene. It has fa variety of functions, but within the immune system, vitamin A acts as a signaling molecule that impacts the adaptive immune system.[42] The metabolic models presented here capture intracellular metabolism rather than signaling, so this signaling role is not measured directly here.

Classical and nonclassical monocytes show a positive correlation between flux and age in vitamin A metabolism (Figure 15 Panels A and B). CCR4- central memory and naive CD8+ T cells likewise show a positive correlation (Figure 15 Panels C and D). In contrast, proliferative CD8+ T cells show a negative correlation between flux and age in vitamin A metabolism (Figure 15 Panel E).

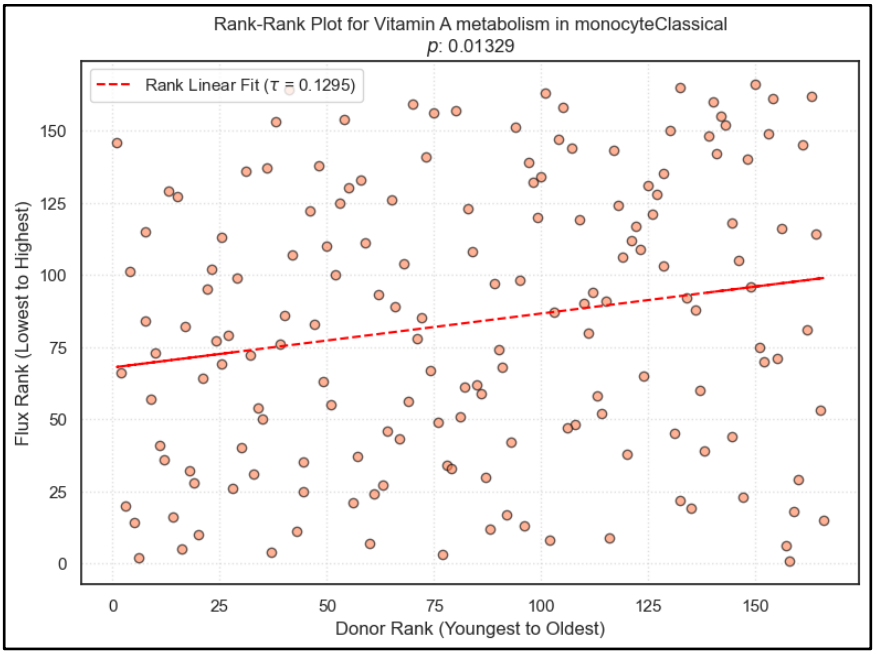
The positive correlation in monocytes is consistent with reports in inflammatory settings. In Crohn's disease, all three antigen-presenting cell populations examined ($CD103^+$ and $CD103^-$ dendritic cells and $CD14^+$ macrophages) had increased activity of the retinaldehyde dehydrogenase enzyme family, and in CD14+ macrophages, this was linked to increased ALDH1A1 expression.[43] In the same study, blocking retinaldehyde receptor signaling during GM-CSF-driven differentiation of monocytes into macrophages reduced the formation of tumor necrosis factor α-producing inflammatory macrophages and lowered CD14 and HLA-DR expression. To test whether this specific enzyme was

driving the monocyte result, Recon3D retinaldehyde dehydrogenase reaction ALDH1A1 was examined on its own. It had a tau of $-1.50 * 10^{-2}$ in nonclassical and $9.69 * 10^{-3}$ in classical monocytes, with p-values of $8.65 * 10^{-1}$ and $8.17 * 10^{-1}$. ALDH1A1 alone is therefore not responsible for the age correlation seen across the pathway in the results presented here. One possible reason is that the relevant RALDH family of enzyme activity is specific to macrophages rather than monocytes; macrophage-specific models could not be built because the Terekhova dataset contains no macrophage-labeled cells, so this possibility remains untested here.

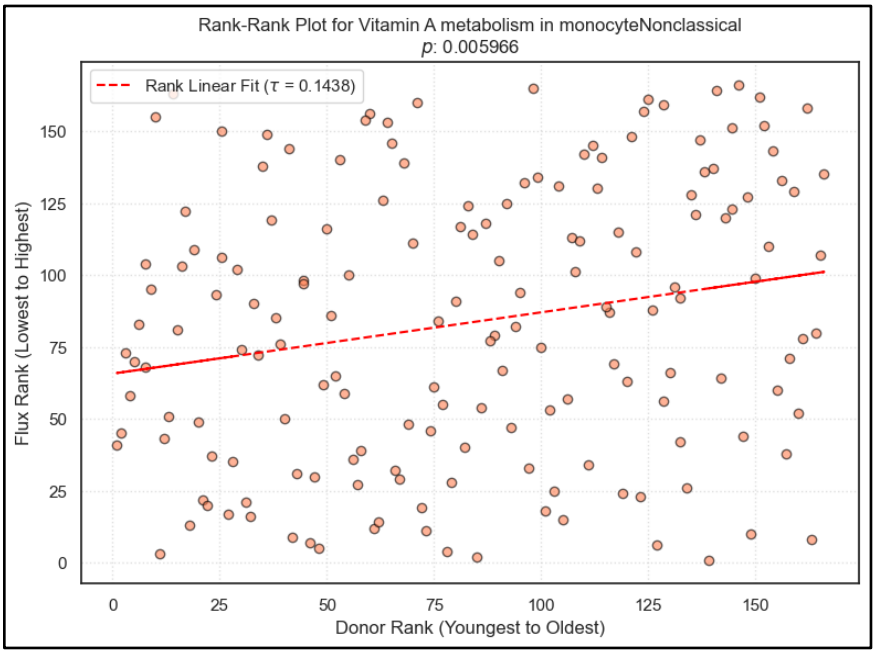
The monocyte and CD8+ results contradict existing literature, which state that vitamin A signaling receptors decrease with age[44], whereas vitamin A metabolism increases here. The difference could be reflective of (1) a reduced sensitivity to vitamin A, so that more is needed to trigger the same receptor response, or (2) the chronic, low-grade inflammation that rises with age.

Separately, these findings correlate with the observation from the cell-count data presented earlier (Figure 8), where the CD8+ T cell lineage, particularly the naive CD8+ subset, is less represented in older donors. The two measurements (vitamin A and cell counts) are distinct. The former represents modeled metabolic flux within the naive CD8+ subset, and the latter is the number of recovered cells. Still, the two independent results trend in a coherent direction with age (fewer cells in the lineage and increased modeled flux, except for proliferative CD8+ cells), making this subset a reasonable focus for a future study.

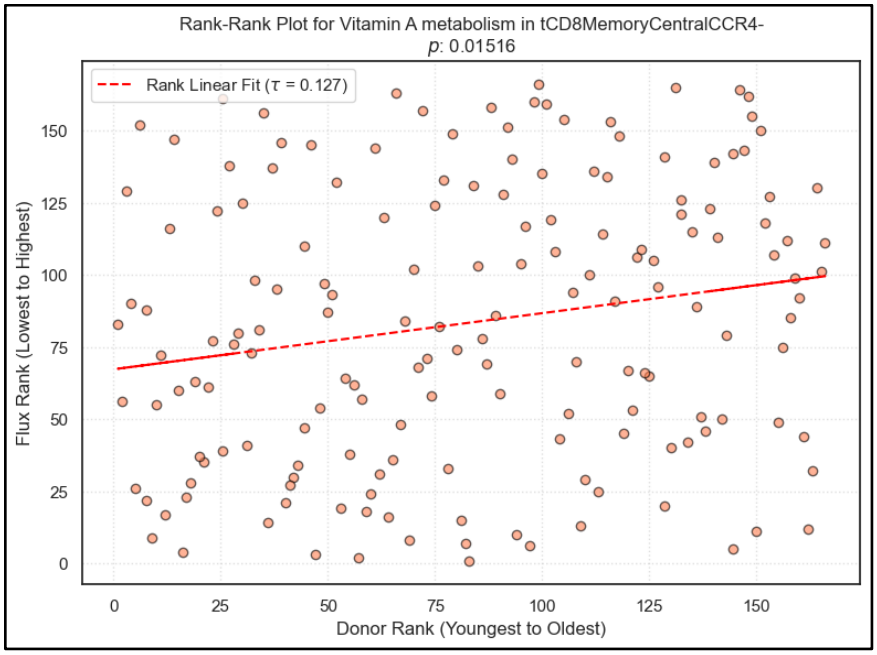
A


Rank-Rank Plot for Vitamin A metabolism in monocyteClassical
p: 0.01329
Rank Linear Fit (τ = 0.1295)
Flux Rank (Lowest to Highest)
Donor Rank (Youngest to Oldest)
0
25
50
75
100
125
150

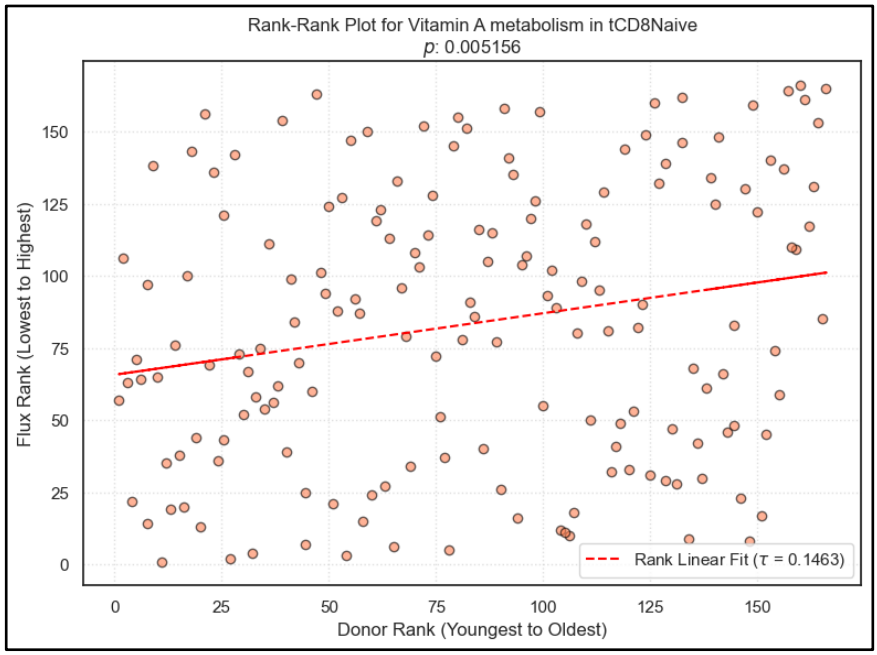
B


Rank-Rank Plot for Vitamin A metabolism in monocyteNonclassical
p: 0.005966
Rank Linear Fit (τ = 0.1438)
Flux Rank (Lowest to Highest)
Donor Rank (Youngest to Oldest)
0
25
50
75
100
125
150

C


Rank-Rank Plot for Vitamin A metabolism in tCD8MemoryCentralCCR4-
p: 0.01516
Rank Linear Fit (τ = 0.127)
Flux Rank (Lowest to Highest)
Donor Rank (Youngest to Oldest)
0
25
50
75
100
125
150

D


Rank-Rank Plot for Vitamin A metabolism in tCD8Naive
p: 0.005156
Rank Linear Fit (τ = 0.1463)
Flux Rank (Lowest to Highest)
Donor Rank (Youngest to Oldest)
0
25
50
75
100
125
150

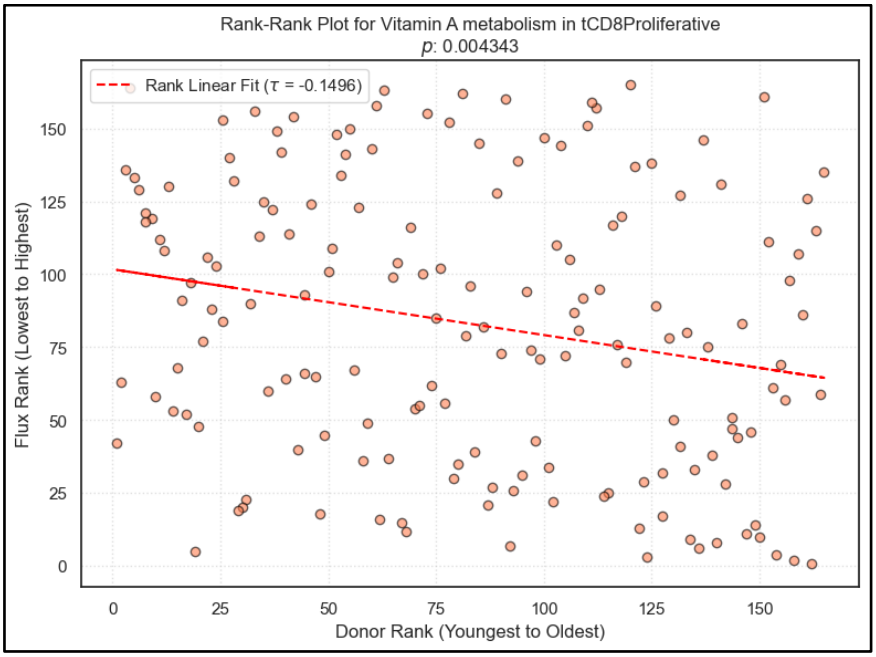
E 

Figure 15: Rank-rank Kendall correlation coefficient plots for vitamin A metabolism in cell types where it was indicated as significant ($p < 0.05$). Panel A) Classical monocytes (tau: 1.295e-01; p: 1.329e-02). Panel B) Nonclassical monocytes (tau: 1.438e-01; p: 5.966e-03). Panel C) CCR4- memory central CD8+ T cells (tau: 1.270e-01; p: 1.516e-02). Panel D) Naive CD8+ T cells (tau: 1.463e-01; p: 5.156e-03). Panel E) Proliferative CD8+ T cells (tau: -1.496e-01; p: 4.343e-03).

Vitamin B2 Metabolism

Vitamin B2 (riboflavin) is an essential, water-soluble vitamin that has known antioxidant properties and the precursor to the cofactors flavin adenine dinucleotide (FAD) and flavin mononucleotide (FMN).[45] These cofactors are required by flavin-dependent enzymes, including those that carry out oxidative energy metabolism and those that regenerate the cell's main reducing molecules. Two of the latter, glutathione reductase and thioredoxin

reductase, are FAD-dependent and restore the reduced forms of glutathione and thioredoxin that the cell uses to manage reactive oxygen species (ROS).

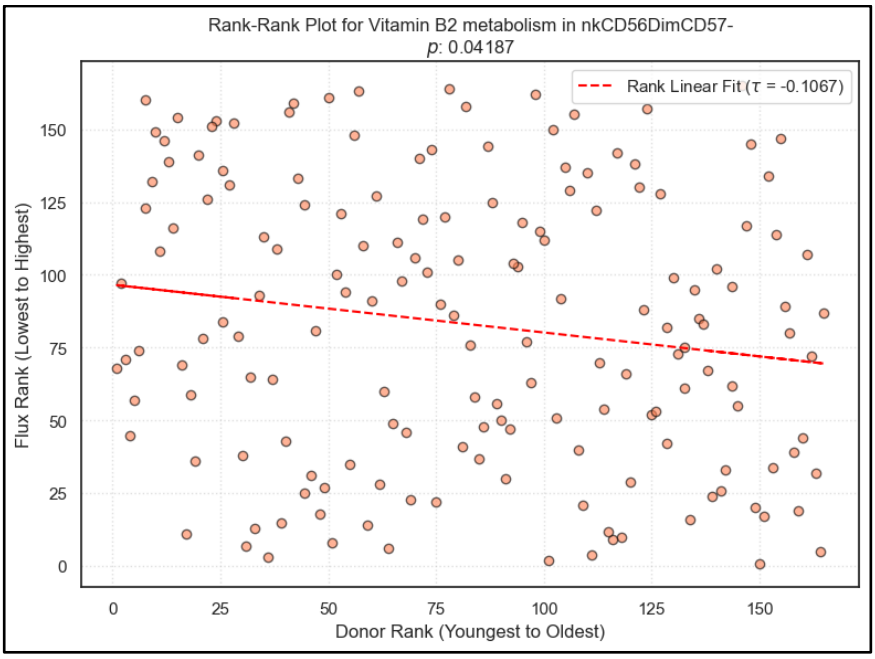
Across the cohort, CD56dim/CD57- and CD56dim/CD57low natural killer cells, and Vd2 GZMK+ gamma delta T cells each show a negative correlation between riboflavin-pathway flux and donor age (Figure 16 Panels A–D).

Direct evidence on riboflavin metabolism in these cell types is sparse. Most work on B vitamins in natural killer cells concerns other vitamins; for example, vitamin B3 increased surface CD62L, a receptor involved in recruiting NK cells to the bone marrow and lymph nodes from circulation[46] and vitamin B6 depletion reduced T cell cytotoxicity but did not affect NK cell activity in mice[47]. For gamma delta T cells, there is even less evidence; the riboflavin-immune literature is largely related to signaling and bacterial defense in unconventional T cells rather than metabolism[48–51].

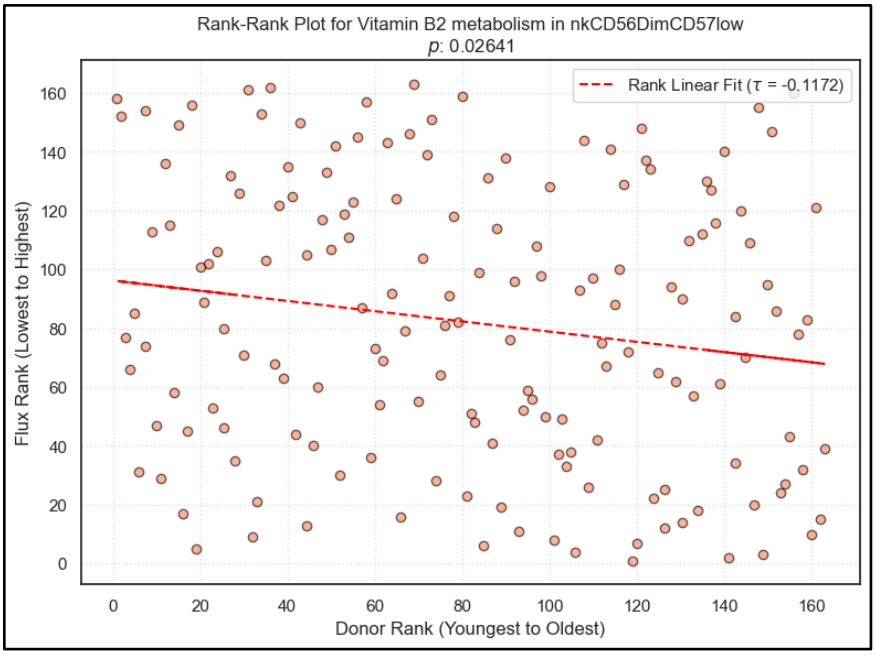
Because the affected reactions depend on flavin cofactor biochemistry rather than on behavior specific to any one cell type, a decline in riboflavin-pathway flux with age corresponds to a potential reduction in flavin-dependent synthesis capacity. As FAD-dependent enzymes, glutathione reductase and thioredoxin point to a shift in the cell's oxidative metabolism, its capacity to manage ROS, or both. This is consistent with the broader picture of immune aging, in which cells accumulate mitochondrial dysfunction and accumulated oxidative damage.[52] Because the GSMNs encode enzymatic capacity from transcript abundance rather than substrate availability, this hypothesis concerns the enzyme capacity itself; changes in flux values could reflect reduced FAD/FMN supply or

reduced downstream demand, but these alternatives cannot be evaluated from the data presented here alone.

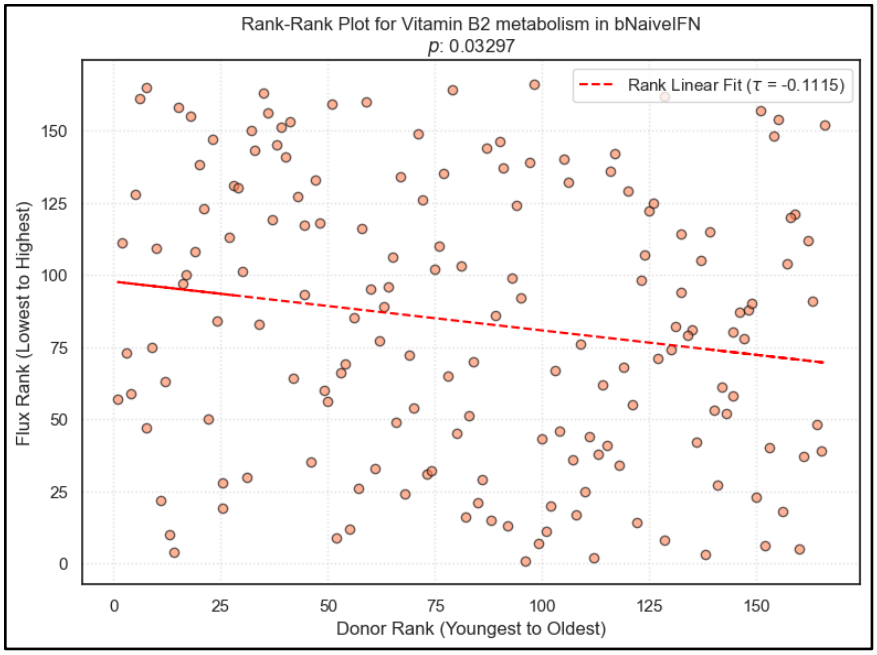
A



B



C



D

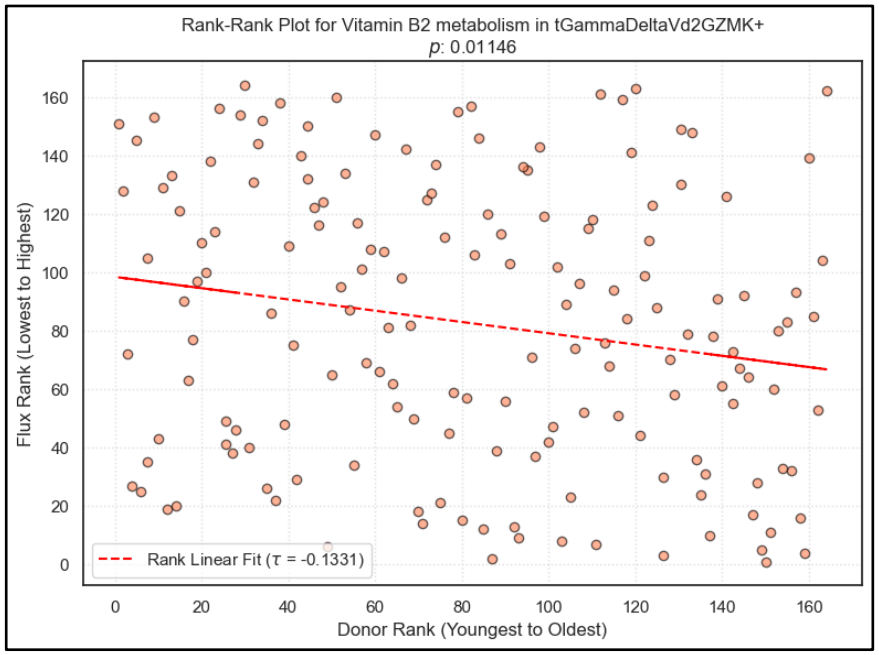


Figure 16: Rank-rank Kendall correlation coefficient plots for Vitamin B2 metabolism in cell types where it was indicated as significant. Panel A) CD56dim/CD57- natural killer cells (tau: -1.067e-01; p: 4.187e-02). Panel B) CD56dim/CD57low natural killer cells (tau: -1.172e-01; p: 2.641e-02). Panel C) IFN-naive B cells (tau: -1.115e-01; p: 3.297e-02). Panel D) Vd2 GZMK+ gamma delta T cells (tau: -1.331e-01; p: 1.146e-02).

Hyaluronan Metabolism

Hyaluronan is a long, unbranched sugar polymer built from a single repeating unit of D-glucuronic acid and N-acetyl-D-glucosamine.[53] Three hyaluronan enzymes (HAS1, HAS2, and HAS3) exist in the membrane and synthesize the chain from two sugar donors, and the yield of hyaluronan are directly impacted by the intracellular concentration (and ratio) of its upstream substrates (UDP-N-acetylglucosamine and UDP-glucuronic acid).[54] The three synthases are not interchangeable, differ in the amount of hyaluronan they make, and in the length of the chains they produce. HAS3 generally produces shorter chains than HAS1 and HAS2.[54] Longer, high-molecular-weight hyaluronan polymers tend to suppress inflammation and immune activity, but the shorter fragments (left behind by enzymatic degradation or oxidative damage) tend to promote inflammation.[54] Cells detect hyaluronan mostly through CD44, a transmembrane receptor that connects hyaluronan binding to adhesion, migration, and activation signaling.[54]

CD56dim/CD57int natural killer cells show a positive correlation between hyaluronan flux and age (Figure 17 Panel A), while naive B cells, naive CD4+ T cells, CCR4+ central memory CD8+ T cells, and Vd2 GZMK+ gamma delta T cells all show a negative correlation (Figure 17 Panels B-E).

The negative correlation in T cell subsets is most directly explained in the naive subset. Resting naive T cells maintain a hyaluronan surface coat that is built mostly by HAS3, is stripped away upon activation, and is then localized to an internal pool alongside CD44.[55] HAS3-driven hyaluronan synthesis is, as a result, a marker of a quiescent, uncommitted T

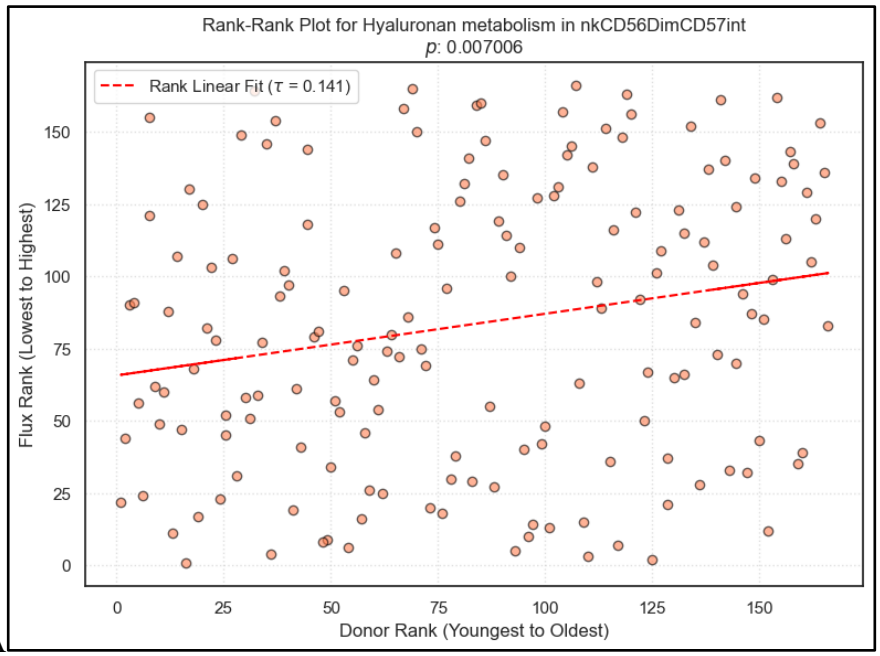
cell. As the immune system ages, the naive pool shrinks, and the overall T cell population shifts toward effector differentiation and an age-associated inflammatory state, as previously described.[52] This logic extends to the Vd2 GZMK+ gamma delta T cells. Older donors accumulate these cells in a more differentiated, inflammatory state, and because that state is associated with reduced HAS3 activity, their capacity for hyaluronan synthesis falls with age. Underlying this is a functional switch associated with T cell activation; resting cells depend on HAS3[55], but activated cells downregulate HAS3 and upregulate HAS1 and HAS2[56]. The decline therefore reflects the loss of HAS3-associated resting state, not reduced hyaluronan synthesis altogether.

In contrast to the T cell populations, the CD56dim/CD57int natural killer (NK) cell population shows an increasing correlation with age, and the proposed hypothesis is not a direct increase in hyaluronan synthesis. Most existing literature linking NK cells to hyaluronan concerns their response to hyaluronan synthesized by other cells and received via CD44, rather than to their own de novo synthesis. CD44 engagement has been shown to be directly tied to NK cell activation.[57] The increase in hyaluronan pathway activity in older donors' NK cells reflects a potential shift in NK cell metabolism rather than a function-specific increase in hyaluronan synthesis. Because age, as described throughout, is accompanied by a low-grade, persistent inflammation, NK cells in this environment are pushed to a chronic, primed metabolic state, potentially relying more on glucose and glutamine. Some of this increased uptake is routed into the hexosamine pathway, which expands the supply of upstream metabolites feeding into hyaluronan metabolism. A second, more speculative and currently unsupported possibility is that the NK cells are, in

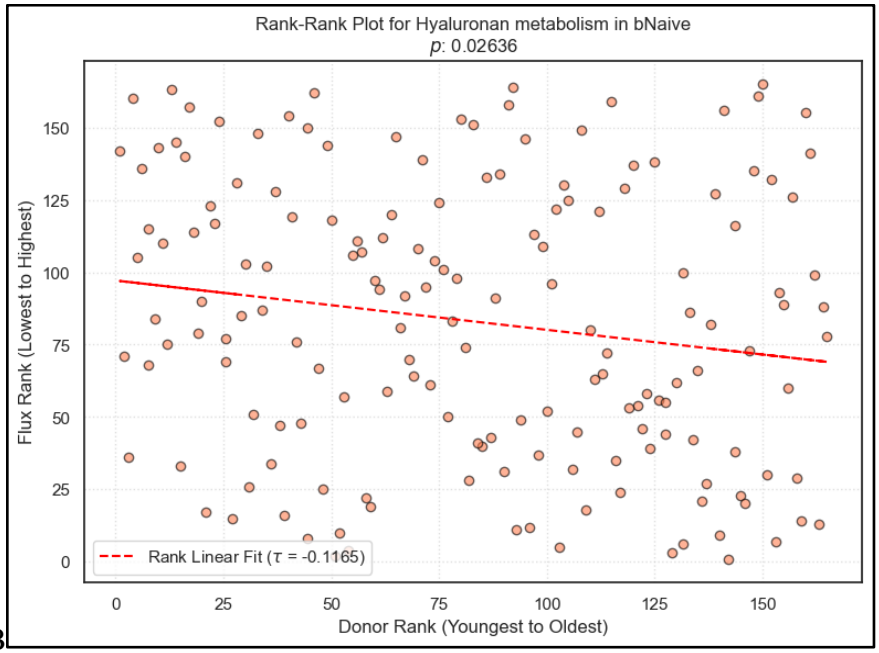
fact, making more hyaluronan for display at their surface, where the coat would serve a function. A thicker hyaluronan layer engaging CD44 could support the adhesion and movement that let NK cells travel through tissue[58–60], and because long-chain hyaluronan suppresses NK killing, such a coat could also restrain NK cell function[61]. This would fit a recognized feature of NK cell aging. Older NK cells are often present in normal or increased numbers, but are individually less effective at killing. A hyaluronan coat that limits cytotoxicity could contribute to this reduced effectiveness. Long-chain hyaluronan has been linked to reduced inflammation and better aging in animal studies, where raising hyaluronan synthase activity extended healthy lifespan, partly by calming the immune system[62], which suggests that interpreting these results as an inflammation-damping compensatory effect is plausible.

The naive B cell results correlate to a declining flux with age, as seen in T cells, but the most plausible explanation differs. No literature reports that B cells synthesize hyaluronan, though none reports that they do not, either. It is established that B cells respond to hyaluronan synthesized by other cells by binding to CD44 (as described for NK cells), which can push B cells toward activation. Naive B cells carry high levels of CD44[63], and are therefore well equipped to sense hyaluronan. A potential interpretation is that two things change in naive B cells with age. First, the cell's general capacity to manufacture activated nucleotide sugars, and second, its proximity to an activatable/primed state. The hyaluronan synthase enzymes sit at the end of a series of sequential reactions whose inputs are shared with bulk protein glycosylation across the entire cell. A cell that is ready to proliferate maintains larger pools, and higher turnover,

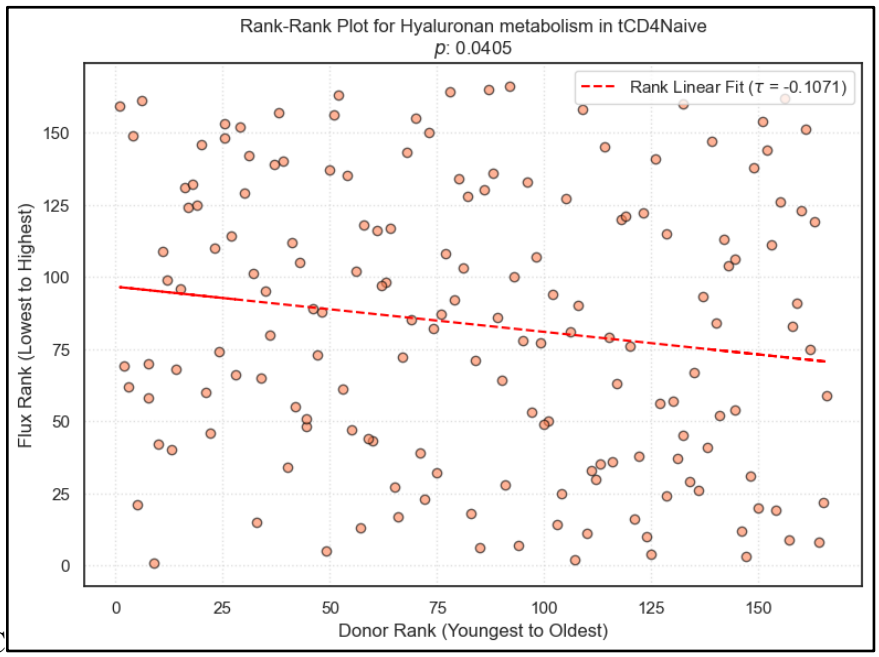
of these sugars[64,65], and because the models are built from each donor's specific expression data, genes expressed highly enough to be classified as active during the reconstruction process are retained as flux-carrying reactions, whereas reactions associated with low expression tend to be pruned or constrained toward inactivity. As individuals age, naive B cells become less abundant and enter an even more quiescent state than in younger individuals.[66] The declining hyaluronan flux is therefore best interpreted as a downstream reporter of reduced nucleotide-sugar availability and glycosylation capacity.

A 

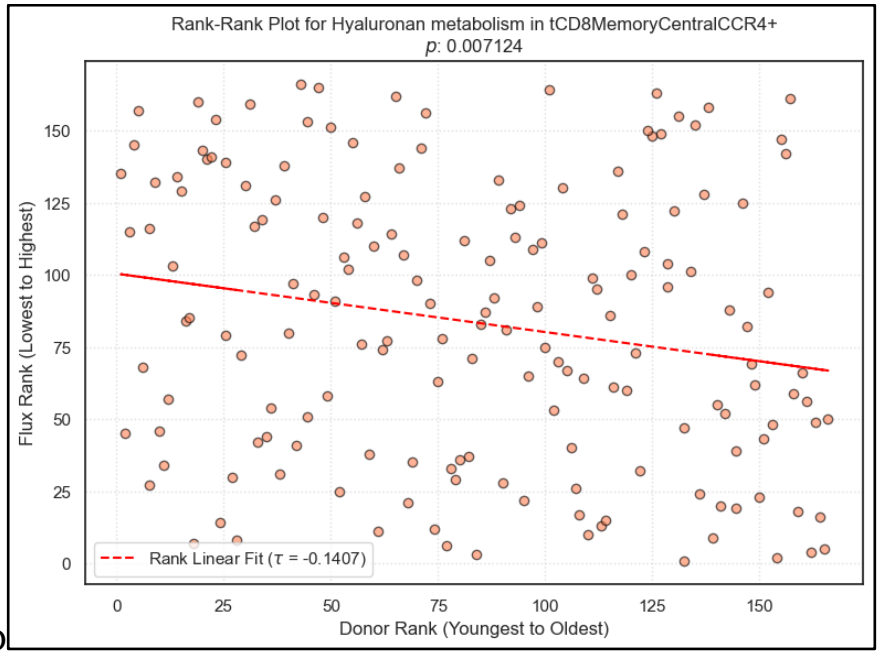
B


Rank-Rank Plot for Hyaluronan metabolism in bNaive
p: 0.02636
Flux Rank (Lowest to Highest)
Donor Rank (Youngest to Oldest)
Rank Linear Fit (τ = -0.1165)
0
25
50
75
100
125
150

C


Rank-Rank Plot for Hyaluronan metabolism in tCD4Naive
p: 0.0405
Flux Rank (Lowest to Highest)
Donor Rank (Youngest to Oldest)
Rank Linear Fit (τ = -0.1071)
0
25
50
75
100
125
150

D



E

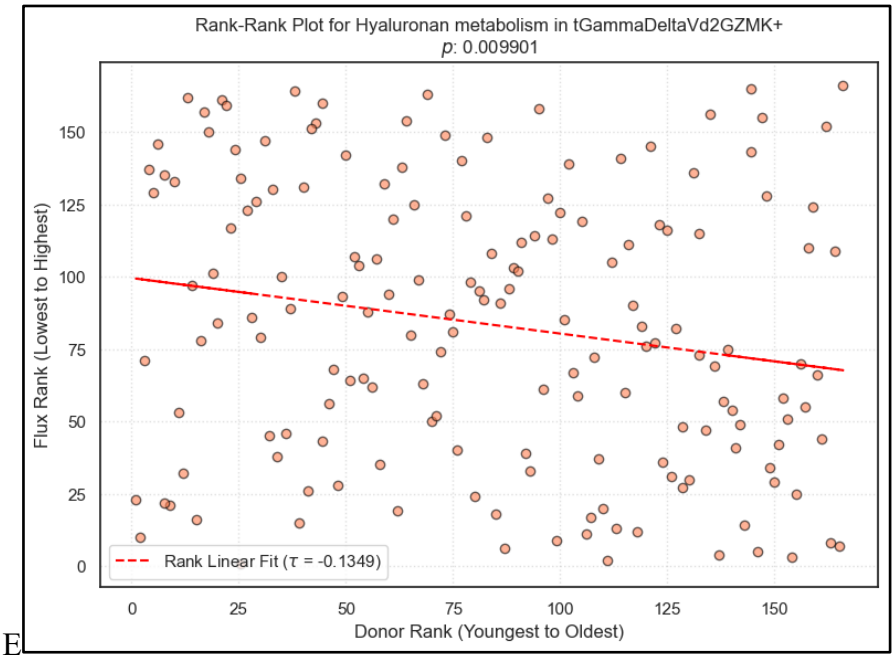


Figure 17: Rank-rank Kendall correlation coefficient plots for hyaluronan metabolism in cell types where it was indicated as significant. Panel A) CD56dim/CD57int Natural Killer cells (tau: 1.410e-01; p: 7.006e-03). Panel B) Naive B cells (tau: -1.165e-01; p: 2.636e-02). Panel C) Naive CD4+ T cells (tau: -1.071e-01; p: 4.050e-02). Panel D) CCR4+ Central Memory CD8+ T cells (tau: -1.407e-01; p: 7.124e-03). Panel E) Vd2 GZMK+ Gamma Delta T cells (tau: -1.349e-01; p: 9.901e-03)

Conclusion

5,705 donor- and cell-type-specific genome-scale metabolic models were reconstructed from donors aged 25 to 85, the range of feasible flux through each model was estimated by sampling, and then tested whether flux through each metabolic pathway changes with donor age. Two main findings are derived from this work. First, the age-associated signal is concentrated rather than spread evenly, with a small number of pathways being significant across many cell types. The signal is similarly concentrated in a subset of cell types (naive gamma delta, naive CD8+, terminal effector CD4+, Th1 CD4+, and Vd2 GZMK+ gamma delta T cells). Second, the direction of the age-related association depends on the cell type rather than a global, coordinated shift, so labeling a pathway "age-associated" overall does not predict its direction in any given cell type. Tetrahydrobiopterin and vitamin A metabolism increase with age across the largest number of cell types (five and six), the citric acid cycle, cholesterol, cytochrome, and androgen/estrogen metabolism increase across four each, and vitamin B2 is the one pathway that contains only significant, decreasing correlations, with hyaluronan and phenylalanine metabolism also predominantly decreasing. Because the models are built from single-cell RNA-sequencing expression data, they describe the flux a cell's metabolism could support, rather than the flux it actually carries.

Four pathways were examined in detail, and several point in a direction consistent with the chronic, low-grade inflammation that rises with age ("inflammaging"). The increase

in tetrahydrobiopterin pathway flux in gamma delta and CD8+ T cell subsets and in monocytes is consistent with cytokine-driven GCH1 activity, which would maintain cofactor supply for inducible nitric oxide synthase during sustained immune activation. The decline in riboflavin-pathway flux in natural killer and gamma delta T cell subsets points to reduced capacity for flavin-dependent reactions, including enzymes that support oxidative energy metabolism and those that regenerate the cell's main defense against reactive oxygen species. The loss of hyaluronan synthesis in naive T and B cell subsets tracks the shrinking of the resting, naive pool with age, while the increase in one natural killer subset (CD56dim/CD57int) is more plausibly a side effect of a broader shift toward glucose and glutamine use than a dedicated increase in hyaluronan production. These cases are consistent with older immune cells moving away from a quiescent, resting state, but the cell-type dependence of the direction described above argues against interpreting them as a coordinated shift in the entire immune system's metabolism. Instead, aging shifts immune-cell metabolism in a pathway- and cell-type-specific way, several of which align with an inflammatory response.

This work's contribution comprises two main components. The first is methodological, and shows that constraint-based metabolism models can be built at the resolution of a single donor and a single immune subtype, at a scale of thousands of models, and then used to ask which metabolic capacities differ with age. This is a different question from the one asked by most single-cell studies of immune aging, which describe how gene expression or cell-type proportions change; here, the expression data are used to build a metabolic network, so the analysis reports on the fluxes those genes can support rather

than on transcript levels directly. The second is biological. The approach produces specific, testable hypotheses tied to particular cell-type and pathway combinations, which would otherwise be difficult to view without modeling each subtype separately. Together, these narrow a large search space to a manageable set of candidates for experimental follow-up.

The models encode enzymatic capacity rather than the metabolites a cell has available, so a change in flux can reflect a change in enzyme capacity, a change in a downstream demand, or both. Additionally, circulating metabolite concentrations cannot influence the modeled flux because donor-matched measurements were unavailable to constrain the models. The experimental/technical design is cross-sectional, so every association reflects differences across donors of varying ages, rather than a change measured within a single individual over time. The effect sizes are small (Kendall's tau around 0.1 to 0.2), which are statistically distinguishable from no association but individually weak. This is expected, given that the donors' diet, stress, and general lifestyle could not be controlled.

Future work falls into three main groups. The first is validation. Each pathway listed here should be broken down by reaction to confirm that the age signal is in biologically meaningful reactions, rather than in a shared set of, e.g., exchange or uptake, reactions. Because Kendall's tau only detects associations that move consistently in one direction, distribution-based comparisons such as Kolmogorov-Smirnov tests and the Wasserstein distance between age groups could potentially recover pathways whose flux changes in a non-monotonic way, for example, rising in the middle and falling later, that the current

analysis would miss. The second group is biological. The interpretations offered here, specifically for tetrahydrobiopterin, riboflavin, and hyaluronan could be tested directly with donor-matched plasma metabolite measurements, transporters measurements supplying the relevant substrates (such as SLC7A5/LAT1 for phenylalanine), and enzyme expression and activity measurement in the implicated cell types. The third is prediction. The present analysis identifies pathways whose flux is associated/indicative of age, but does not ask whether that flux can estimate age. A natural extension is to treat the reaction fluxes as features in a machine learning model and ask if donor age can be predicted. In effect, a flux analogue to the existing transcriptomic and epigenetic "aging clocks".[67,68] This model would clarify if the age signal is concentrated in a small set of reactions or distributed throughout metabolism, and the features it weights most heavily could be cross-referenced against the pathways marked here as an independent validation on which associations carry predictive information.

The number of models reconstructed here points to a practical problem that this work does not solve. Examining 5,705 GSMNs by hand is not realistic. Each one contains thousands of reactions, and careful interrogation of even a single model, tracing which reactions carry flux and connecting the results to relevant literature, takes a considerable amount of time. The pathway-level validation described above, repeated across every pathway and cell type, is a task that does not scale to an individual working alone. This is the gap that MechAInistic, discussed in the next chapter, is meant to solve. MechAInistic is a large language model-based system that allows a researcher to ask a question about any GSMN in plain language, then plans, runs, and interprets a dynamically synthesized

multi-step analysis. The hypotheses produced in this chapter, one cell type and one pathway at a time, is the kind of analysis the system is meant to perform at scale. Pairing the metabolic modeling approach used here with a tool like MechAInistic allows investigating thousands of models a possibility, rather than nearly impossible.

Acknowledgements

This work was completed utilizing the Holland Computing Center of the University of Nebraska, which receives support from the UNL Office of Research and Innovation and the Nebraska Research Initiative.

References

1. Franceschi, C., Olivieri, F., Moskalev, A., Ivanchenko, M. & Santoro, A. Toward precision interventions and metrics of inflammaging. Nat. Aging 5, 1441–1454 (2025).

2. Franceschi, C., Garagnani, P., Parini, P., Giuliani, C. & Santoro, A. Inflammaging: a new immune–metabolic viewpoint for age-related diseases. Nat. Rev. Endocrinol. 14, 576–590 (2018).

3. Tang, F. et al. mRNA-Seq whole-transcriptome analysis of a single cell. Nat. Methods 6, 377–382 (2009).

4. Sheih, A. et al. Clonal kinetics and single-cell transcriptional profiling of CAR-T cells in patients undergoing CD19 CAR-T immunotherapy. Nat. Commun. 11, 219 (2020).

5. Gu, C., Kim, G. B., Kim, W. J., Kim, H. U. & Lee, S. Y. Current status and applications of genome-scale metabolic models. Genome Biol. 20, 121 (2019).

6. Brunk, E. et al. Recon3D enables a three-dimensional view of gene variation in human metabolism. Nat. Biotechnol. 36, 272–281 (2018).

7. Ebrahim, A., Lerman, J. A., Palsson, B. O. & Hyduke, D. R. COBRApy: COnstraints-Based Reconstruction and Analysis for Python. BMC Syst. Biol. 7, 74 (2013).

8. Wagner, A. et al. Metabolic modeling of single Th17 cells reveals regulators of autoimmunity. Cell 184, 4168-4185.e21 (2021).

9. Alghamdi, N. et al. A graph neural network model to estimate cell-wise metabolic flux using single-cell RNA-seq data. Genome Res. 31, 1867–1884 (2021).

10. Sompairac, N. et al. Metabolic and signalling network maps integration: application to cross-talk studies and omics data analysis in cancer. BMC Bioinformatics 20, 140 (2019).

11. Gustafsson, J. et al. Generation and analysis of context-specific genome-scale metabolic models derived from single-cell RNA-Seq data. Proc. Natl. Acad. Sci. U. S. A. 120, e2217868120 (2023).

12. Terekhova, M. et al. Single-cell atlas of healthy human blood unveils age-related loss of NKG2C+GZMB−CD8+ memory T cells and accumulation of type 2 memory T cells. Immunity 56, 2836-2854.e9 (2023).

13. Gong, Q. et al. Multi-omic profiling reveals age-related immune dynamics in healthy adults. Nature 648, 696–706 (2025).

14. Nehar-Belaid, D. et al. Single-cell map of the healthy human immune system across the lifespan reveals unique infant immune signatures. Nat. Commun. https://doi.org/10.1038/s41467-026-73729-2 (2026) doi:10.1038/s41467-026-73729-2.

15. Sayed, N. et al. An inflammatory aging clock (iAge) based on deep learning tracks multimorbidity, immunosenescence, frailty and cardiovascular aging. Nat. Aging 1, 598–615 (2021).

16. Segrè, D., Vitkup, D. & Church, G. M. Analysis of optimality in natural and perturbed metabolic networks. Proc. Natl. Acad. Sci. U. S. A. 99, 15112–15117 (2002).

17. Schellenberger, J. et al. Quantitative prediction of cellular metabolism with constraint-based models: the COBRA Toolbox v2.0. Nat. Protoc. 6, 1290–1307 (2011).

18. Edwards, J. S., Ramakrishna, R. & Palsson, B. O. Characterizing the metabolic phenotype: a phenotype phase plane analysis. Biotechnol. Bioeng. 77, 27–36 (2002).

19. Megchelenbrink, W., Huynen, M. & Marchiori, E. optGpSampler: an improved tool for uniformly sampling the solution-space of genome-scale metabolic networks. PloS One 9, e86587 (2014).

20. Bourgon, R., Gentleman, R. & Huber, W. Independent filtering increases detection power for high-throughput experiments. Proc. Natl. Acad. Sci. U. S. A. 107, 9546–9551 (2010).

21. Kendall, M. G. A NEW MEASURE OF RANK CORRELATION. Biometrika 30, 81–93 (1938).

22. Virtanen, P. et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nat. Methods 17, 261–272 (2020).

23. Pearson, K. LIII. On lines and planes of closest fit to systems of points in space. Lond. Edinb. Dublin Philos. Mag. J. Sci. 2, 559–572 (1901).

24. Mukaka, M. et al. Is using multiple imputation better than complete case analysis for estimating a prevalence (risk) difference in randomized controlled trials when binary outcome observations are missing? Trials 17, 341 (2016).

25. Pedregosa, F. et al. Scikit-learn: Machine Learning in Python. J. Mach. Learn. Res. 12, 2825–2830 (2011).

26. Appay, V. & Sauce, D. Naive T cells: the crux of cellular immune aging? Exp. Gerontol. 54, 90–93 (2014).

27. Pangrazzi, L. & Weinberger, B. T cells, aging and senescence. Exp. Gerontol. 134, 110887 (2020).

28. Phalke, S. et al. Molecular mechanisms controlling age-associated B cells in autoimmunity. Immunol. Rev. 307, 79–100 (2022).

29. Naradikian, M. S., Hao, Y. & Cancro, M. P. Age-associated B cells: key mediators of both protective and autoreactive humoral responses. Immunol. Rev. 269, 118–129 (2016).

30. Nipper, A. J., Smithey, M. J., Shah, R. C., Canaday, D. H. & Landay, A. L. Diminished antibody response to influenza vaccination is characterized by expansion of an age-associated B-cell population with low PAX5. Clin. Immunol. 193, 80–87 (2018).

31. Re, F. et al. Skewed representation of functionally distinct populations of Vgamma9Vdelta2 T lymphocytes in aging. Exp. Gerontol. 40, 59–66 (2005).

32. Kallemeijn, M. J. et al. Next-Generation Sequencing Analysis of the Human TCRγδ+ T-Cell Repertoire Reveals Shifts in Vγ- and Vδ-Usage in Memory Populations upon Aging. Front. Immunol. 9, 448 (2018).

33. Parameter Reference - Gurobi Optimizer Reference Manual. https://docs.gurobi.com/projects/optimizer/en/current/reference/parameters.html.

34. Akoglu, H. User's guide to correlation coefficients. Turk. J. Emerg. Med. 18, 91–93 (2018).

35. Khairallah, A., Ross, C. J. & Tastan Bishop, Ö. GTP Cyclohydrolase I as a Potential Drug Target: New Insights into Its Allosteric Modulation via Normal Mode Analysis. J. Chem. Inf. Model. 61, 4701–4719 (2021).

36. Eichwald, T. et al. Tetrahydrobiopterin: Beyond Its Traditional Role as a Cofactor. Antioxidants 12, 1037 (2023).

37. Fanet, H., Capuron, L., Castanon, N., Calon, F. & Vancassel, S. Tetrahydrobioterin (BH4) Pathway: From Metabolism to Neuropsychiatry. http://www.eurekaselect.com https://www.eurekaselect.com/article/108621.

38. Czibik, G. et al. Dysregulated Phenylalanine Catabolism Plays a Key Role in the Trajectory of Cardiac Aging. Circulation 144, 559–574 (2021).

39. Kasai, K. et al. Induction of tetrahydrobiopterin synthesis in rat cardiac myocytes: impact on cytokine-induced NO generation. Am. J. Physiol. 273, H665-672 (1997).

40. Geller, D. A., Di Silvio, M., Billiar, T. R. & Hatakeyama, K. GTP cyclohydrolase I is coinduced in hepatocytes stimulated to produce nitric oxide. Biochem. Biophys. Res. Commun. 276, 633–641 (2000).

41. Duong, L. et al. Macrophage function in the elderly and impact on injury repair and cancer. Immun. Ageing A 18, 4 (2021).

42. Mora, J. R., Iwata, M. & von Andrian, U. H. Vitamin effects on the immune system: vitamins A and D take centre stage. Nat. Rev. Immunol. 8, 685–698 (2008).

43. Sanders, T. J. et al. Increased production of retinoic acid by intestinal macrophages contributes to their inflammatory phenotype in patients with Crohn's disease. Gastroenterology 146, 1278-1288.e1–2 (2014).

44. Feart, C. et al. Aging affects the retinoic acid and the triiodothyronine nuclear receptor mRNA expression in human peripheral blood mononuclear cells. Eur. J. Endocrinol. 152, 449–458 (2005).

45. Cheung, I. M. Y., Mcghee, C. N. J. & Sherwin, T. Beneficial effect of the antioxidant riboflavin on gene expression of extracellular matrix elements, antioxidants and oxidases in keratoconic stromal cells. Clin. Exp. Optom. 97, 349–355 (2014).

46. Frei, G. M. et al. Nicotinamide, a Form of Vitamin B3, Promotes Expansion of Natural Killer Cells That Display Increased In Vivo Survival and Cytotoxic Activity,. Blood 118, 4035 (2011).

47. Ha, C., Miller, L. T. & Kerkvliet, N. I. The effect of vitamin B6 deficiency on cytotoxic immune responses of T cells, antibodies, and natural killer cells, and phagocytosis by macrophages. Cell. Immunol. 85, 318–329 (1984).

48. Gao, Y. & Williams, A. P. Role of Innate T Cells in Anti-Bacterial Immunity. Front. Immunol. 6, (2015).

49. Maerz, M. D., Cross, D. L. & Seshadri, C. Functional and biological implications of clonotypic diversity among human donor-unrestricted T cells. Immunol. Cell Biol. 102, 474–486 (2024).

50. Rice, M. T. et al. Recognition of the antigen-presenting molecule MR1 by a Vδ3+ γδ T cell receptor. Proc. Natl. Acad. Sci. 118, e2110288118 (2021).

51. Le Nours, J. et al. A class of γδ T cell receptors recognize the underside of the antigen-presenting molecule MR1. Science 366, 1522–1527 (2019).

52. Rousseau, L., Hajdu, K. L. & Ho, P.-C. Meta-epigenetic shifts in T cell aging and aging-related dysfunction. J. Biomed. Sci. 32, 51 (2025).

53. Lu, J. et al. Hyaluronidase: structure, mechanism of action, diseases and therapeutic targets. Mol. Biomed. 6, 50 (2025).

54. Palenčárová, K., Köszagová, R. & Nahálka, J. Hyaluronic Acid and Its Synthases—Current Knowledge. Int. J. Mol. Sci. 26, 7028 (2025).

55. Adhikari, L. P. et al. Naive CD4+ T cells synthesize a hyaluronan-rich glycocalyx that is diminished following activation. iScience 29, (2026).

56. Gebe, J. A. et al. Modulation of hyaluronan synthases and involvement of T cell-derived hyaluronan in autoimmune responses to transplanted islets. Matrix Biol. Plus 9, 100052 (2021).

57. Rahmati, A., Bigam, S. & Elahi, S. Galectin-9 promotes natural killer cells activity via interaction with CD44. Front. Immunol. 14, 1131379 (2023).

58. Reiprich, S. et al. Adhesive Properties of the Hyaluronan Pericellular Coat in Hyaluronan Synthases Overexpressing Mesenchymal Stem Cells. Int. J. Mol. Sci. 21, 3827 (2020).

59. Wang, D. et al. Hyaluronic acid-CD44 signaling from decidual stromal cells orchestrates dNK1 differentiation and immune tolerance in early pregnancy. Front. Immunol. 17, 1777567 (2026).

60. Sague, S. L., Tato, C., Puré, E. & Hunter, C. A. The regulation and activation of CD44 by natural killer (NK) cells and its role in the production of IFN-gamma. J. Interferon Cytokine Res. Off. J. Int. Soc. Interferon Cytokine Res. 24, 301–309 (2004).

61. Kong, C.-S. et al. Embryo biosensing by uterine natural killer cells determines endometrial fate decisions at implantation. FASEB J. 35, e21336 (2021).

62. Zhang, Z. et al. Increased hyaluronan by naked mole-rat Has2 improves healthspan in mice. Nature 621, 196–205 (2023).

63. Klein, U. et al. Transcriptional analysis of the B cell germinal center reaction. Proc. Natl. Acad. Sci. 100, 2639–2644 (2003).

64. Laurent, T. C. & Fraser, J. R. Hyaluronan. FASEB J. Off. Publ. Fed. Am. Soc. Exp. Biol. 6, 2397–2404 (1992).

65. Evanko, S. P. & Wight, T. N. Intracellular Localization of Hyaluronan in Proliferating Cells. J. Histochem. Cytochem. 47, 1331–1341 (1999).

66. Frasca, D. et al. Aging Down-Regulates the Transcription Factor E2A, Activation-Induced Cytidine Deaminase, and Ig Class Switch in Human B Cells. J. Immunol. 180, 5283–5290 (2008).

67. Zhu, H. et al. Human PBMC scRNA-seq-based aging clocks reveal ribosome to inflammation balance as a single-cell aging hallmark and super longevity. Sci. Adv. 9, eabq7599 (2023).

68. Gems, D., Virk, R. S. & de Magalhães, J. P. Epigenetic clocks and programmatic aging. Ageing Res. Rev. 101, 102546 (2024).

Chapter VI: Reviewer-supervised, multi-agent LLM orchestration generates model-grounded, drug-target hypotheses from paired disease/healthy state metabolic model

Josh Loecker, Narayna Puraja*, William Bryant*, Bhanwar Lal Puniya, Prakash Packrisamy, Ahmed Abdeen Hamed, and Tomáš Helikar

*These authors contributed equally

Contribution

I architected and implemented the MechAInistic platform, including its multi-agent architecture, workflow control logic, and constraint-based modeling backend. I mentored two undergraduate developers (Narayana Puniya and William Bryant), reviewed their contributions, and integrated their work into the codebase. I reconstructed the healthy and disease state Th17 constraint-based metabolic models. I designed and executed both therapeutic hypothesis-generation use cases with assistance from P.P. and A.A.H. I, along with A.A.H., were main contributors to the text.

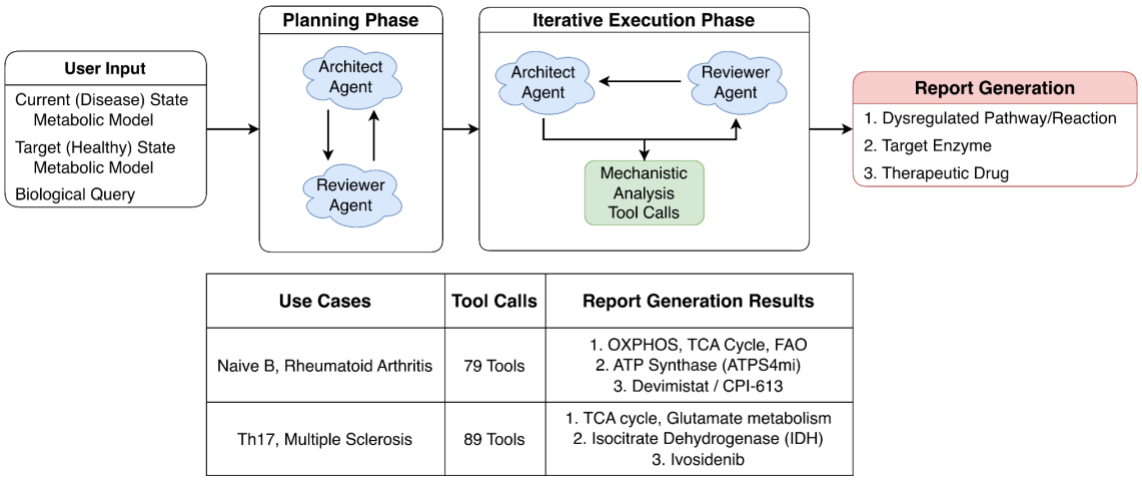
This chapter is available as a preprint at bioRxiv at https://doi.org/10.64898/2026.05.11.723319

Abstract

Constraint-based metabolic modeling is a powerful way to study the mechanistic basis of cellular states and disease, but effective use demands substantial computational expertise

and careful coordination of multi-step analyses. We developed MechAInistic to lower this barrier enabling researchers to ask complex biological questions in natural language. MechAInistic is a multi-agent system harnessing large language models organized around an Architect-Reviewer pattern that converts a natural-language question into an executable, model-grounded workflow and produces a structured report. It supports pathway comparison, perturbation analysis, drug-target exploration, and literature interpretation across healthy and disease paired states. We evaluated MechAInistic's therapeutic hypothesis generation using two immune-cell use-cases. For rheumatoid arthritis/healthy naive B models, it identified mitochondrial metabolic rewiring and nominated Devimistat/CPI-613 as an investigational OGDH-centered hypothesis. In CD4+ Th17 multiple sclerosis/healthy models, the workflow identified NADP-dependent isocitrate dehydrogenase as the optimal target and proposed Ivosidenib as an FDA-approved repurposing candidate.

Graphical Abstract



Use Cases	Tool Calls	Report Generation Results
Naive B, Rheumatoid Arthritis	79 Tools	1. OXPHOS, TCA Cycle, FAO 2. ATP Synthase (ATPS4mi) 3. Devimistat / CPI-613
Th17, Multiple Sclerosis	89 Tools	1. TCA cycle, Glutamate metabolism 2. Isocitrate Dehydrogenase (IDH) 3. Ivosidenib

Figure 18: MechAInistic graphical abstract

Keywords

Agentic AI; workflow orchestration; architect-reviewer pattern; metabolic modeling; constraint-based modeling; mechanistic AI; drug repurposing

Introduction

Genome-scale metabolic networks (GSMNs) have become foundational tools for investigating metabolic capabilities, predicting phenotypic outcomes, and characterizing disease-associated metabolic alterations across biological systems.[1,2] Despite their conceptual accessibility, effective use of GSMNs requires computational and biological expertise.[3,4] Frameworks such as COBRA Toolbox, COBRAxy, and COBRApy provide

powerful analytical capabilities, but they require proficiency in programming, numerical optimization, model configuration, and metabolic-network interpretation.[5–7] Even routine GSMN analyses often require multi-stage workflows that integrate model setup, simulation, perturbation analysis, and interpretation.[1,8,9] As a result, users must assemble analysis pipelines manually, manage tool configurations, and interpret intermediate outputs, creating barriers to reproducibility and limiting accessibility for researchers without specialized computational training.[10–12]

Large language models (LLMs) offer a potential route to more accessible model interaction by enabling natural-language-driven workflows. Recent systems use prompt engineering, in-context learning, retrieval-augmented generation, and agentic AI approaches to assist biological modeling and scientific workflow construction.[13–16] However, LLM-generated workflows can fail under realistic conditions because they are not consistently grounded in executable tools, model constraints, or verifiable intermediate results. For GSMN applications, a useful LLM-enabled system must translate biological questions into executable analyses, check intermediate outputs, adapt when tool calls fail or produce incomplete evidence, and preserve traceability from the original question to model-derived results.

To address this need, we developed MechAInistic, an agentic AI system that orchestrates GSMN analyses through a reviewer-supervised, tool-grounded workflow. MechAInistic separates workflow design from workflow verification, interfaces directly with GSMN tools, evaluates intermediate outputs, and transforms natural-language questions into

structured reports containing computational results, mechanistic interpretation, literature support, and limitations. By coupling LLM-based planning and review to executable metabolic-modeling tools, MechAInistic lowers the barrier to GSMN analysis while preserving the transparency required for model-grounded biological reasoning.

Results

We developed MechAInistic as a multi-agent, LLM-guided system for natural-language reasoning over paired GSMNs. The Results first present the reviewer-supervised architecture, then demonstrate model-grounded workflow generation, comparator performance against general-purpose LLMs, and two immune-cell therapeutic hypothesis-generation use cases.

Translation of Biological Questions into an Executable, Metabolic-modeling Analysis

We developed MechAInistic to reduce the technical barrier between biological questions and executable constraint-based metabolic modeling. The system accepts two paired metabolic models, representing the current and target states, along with a natural-language query. In the use cases presented here, the current state corresponds to a disease-state immune-cell model, and the target state corresponds to the matched healthy reference model. MechAInistic then translates the user's query into a structured computational workflow, executes model-grounded analyses, retrieves relevant literature,

and produces a final report that combines quantitative results, mechanistic interpretation, and stated limitations.

The platform is organized around three role-specialized agents. The Architect agent decomposes the user's question, develops an analytical plan, and proposes tool calls against the uploaded metabolic models. The Reviewer agent evaluates the plan and the accumulated evidence before the system proceeds to final synthesis. The Task agent supports PubMed query generation and summarizes tool outputs, enabling quantitative results to be reused during iterative reasoning without overwhelming the context window. This separation of roles was designed to prevent the system from moving directly from language-model inference to biological conclusion without intermediate checks.

MechAInistic grounds its reasoning in executable metabolic-modeling tools rather than language-model text alone. These tools support model simulation, flux-distance calculation, pathway and reaction comparison, graph-based network analysis, perturbation simulation, dose-response analysis, drug-target lookup, robustness analysis, and literature retrieval. The resulting workflow creates an auditable chain from the original biological query to executed tool calls, model-derived quantitative outputs, literature-grounded interpretation, and final synthesis. Detailed implementation information, including LLM configuration, tool parameters, prompt templates, workflow-control logic, and software dependencies, is provided in the Supplementary Information.

Therapeutic Hypothesis Generation in Two Disease Models

To demonstrate model-grounded workflow generation, we applied MechAInistic to two paired-model therapeutic hypothesis-generation tasks using the same prompt: “Identify a drug therapy that could restore metabolic flux in the disease-state model to the healthy state. Propose a single drug therapy that maximizes therapeutic efficacy while minimizing off-target metabolic disruption.” The first task used rheumatoid arthritis naive B-cell models, and the second used multiple sclerosis CD4+ Th17-cell models. Disease and cell-type labels were not included in the prompt and were available only through the uploaded model context. Illustrative examples of broader query types that MechAInistic can translate into GSMN workflows are provided in Table S1.

In the rheumatoid arthritis naive B-cell case, MechAInistic loaded the supplied JSON models, calculated baseline metabolic divergence, identified reaction- and pathway-level differences, simulated candidate perturbations, and generated a drug-therapy hypothesis. The disease-state model differed substantially from the healthy reference, with a Euclidean flux distance of 151.60 and cosine similarity of 0.093. The analysis identified increased mitochondrial flux, including elevated oxidative phosphorylation, citric acid cycle activity, and reactive oxygen species detoxification. At the reaction level, MechAInistic prioritized 2-oxoglutarate dehydrogenase/OGDH, represented in the model by AKGDm, because it carried disease-state forward flux of approximately 5 mmol $gDW^{-1}\ h^{-1}$ with no corresponding flux in the healthy reference model.

In the multiple sclerosis CD4+ Th17 case, MechAInistic again generated model-derived outputs from the paired disease and healthy metabolic models. Baseline characterization yielded a Euclidean flux distance of 25.247 between disease and healthy states. The system identified NADP-dependent isocitrate dehydrogenase, represented by ICDHy and mapped to the IDH1/IDH2 enzyme family, as a disease-associated reaction with forward flux of 3.441 in the disease model and no corresponding flux in the healthy model. Simulated inhibition reduced the Euclidean distance from 25.247 to 23.069 and produced no detectable reduction in the biomass objective across the tested inhibition range. The analysis also identified a broader glutamate-associated flux signature, including a 2.17-fold elevation in glutamate metabolism and reduced citric acid cycle flux relative to the healthy state.

These demonstrations show that MechAInistic can convert natural-language biological questions into executable, model-grounded workflows across distinct disease states, immune-cell types, and metabolic-model contexts. The outputs included model-derived flux distances, reaction identifiers, pathway-level changes, simulated perturbation effects, drug-target hypotheses, and explicit caveats about confidence and validation.

Improved Model Grounding and Task Completion over General-purpose LLM Systems

We compared MechAInistic with general-purpose LLM systems to determine whether a dedicated, tool-grounded agentic architecture provides practical advantages over standard language-model workflows for metabolic-model reasoning. Anthropic’s Claude,

OpenAI's ChatGPT, Microsoft's Copilot, and Google's Gemini were given the same paired metabolic model files and the same prompt used for MechAInistic: "Identify a drug therapy that could restore metabolic flux in the disease-state model to the healthy state. Propose a single drug therapy that maximizes therapeutic efficacy while minimizing off-target metabolic disruption."

We evaluated each system using a nine-axis rubric designed to distinguish plausible biological narrative from model-grounded computational reasoning. The rubric assessed: (1) model fidelity, defined as evidence that the supplied JSON metabolic models were loaded, parsed, and used; (2) methodological reproducibility, including named tools, solvers, scoring rules, or executable analysis steps; (3) literature evidence, including use of traceable peer-reviewed sources; (4) computational evidence, including flux values, perturbation outputs, or distance metrics; (5) direct relevance to the single-drug flux-restoration task; (6) whether the question was answered with a specific single therapy; (7) evidence of structured review or agentic control; (8) reproducibility and execution feasibility, including whether the workflow was rerunnable or limited by file-ingestion, context, rate-limit, or truncation failures; and (9) human-review traceability, including explicit indication of manual review, sign-off, or correction. Detailed scoring criteria are provided in the Methods and Supplementary Information.

Across both use cases, MechAInistic showed the most consistent workflow-level performance. It loaded and operated on the supplied JSON models, reported model-derived reaction fluxes and distance metrics, identified reaction-level control points,

simulated perturbations, retrieved supporting literature, and documented the analytical path leading to each recommendation. General-purpose LLM systems produced plausible biological recommendations and, in some cases, sophisticated model-aware reasoning, but their outputs varied substantially across file ingestion, quantitative model-derived evidence, methodological traceability, and task completion.

The comparator analyses also showed that the apparent sophistication of a final report was not sufficient evidence of model-grounded correctness. In one Claude run for the multiple sclerosis CD4+ Th17 use case, the final report presented an extensive quantitative analysis and recommended dimethyl fumarate. However, the audit of the underlying steps revealed execution-level issues not visible from the final report alone. Claude manually inhibited reactions associated with selected genes rather than applying COBRApy's gene-protein-reaction-aware knockout logic, leading to reaction inhibitions that would not occur when Boolean isozyme relationships were respected. It also mislabeled exchange-flux directionality by not accounting for uptake sign conventions, reported an analysis step that had been skipped, and made an unsupported mechanistic claim linking dimethyl fumarate to monocarboxylate transporter restoration. In addition, Claude used pFBA to evaluate post-perturbation states. Although pFBA is useful for obtaining parsimonious optimal flux distributions, it answers a different optimization question than MOMA/linear MOMA-style perturbation analysis, which explicitly evaluates perturbed solutions relative to a reference flux distribution.

Table 5: Therapeutic recommendations issued by each system for the two use cases presented in MechAInistic

System	RA Naive B-cell Recommendation	MS CD4+ Th17 Recommendation
MechAInistic (Kimi K2.6, Qwen3.5, Gemma-4)	Devimistat / CPI-613 (α-KGDH inhibitor[17])	Ivosidenib (IDH1 Inhibitor[18])
Claude (Opus 4.7 Max Thinking)	Etomoxir Extensive response, but audit identified execution-level errors	Dimethyl fumarate Extensive response, but audit identified execution-level errors
ChatGPT (GPT-5.5 Thinking)	Metformin (ETC Complex I Inhibitor[19])	Calcitriol (Vitamin D3 Inhibitor[20])
Copilot ("Think Deeper")	Metformin (ETC Complex 1 Inhibitor[19])	Metformin (ETC Complex 1 Inhibitor[19])
Gemini 3 Flash	No recommendation (Error in responding)	No recommendation (Error in responding)

Therapeutic recommendations differed substantially across systems: MechAInistic nominated Devimistat/CPI-613 and ivosidenib, whereas general-purpose LLM systems returned metformin, calcitriol, dimethyl fumarate, or no recommendation depending on the interface and use case. We did not treat these differences as evidence that one drug was correct, because the correct therapeutic answer cannot be established computationally. Instead, the divergence illustrates why drug identity alone is an insufficient endpoint. The primary distinction was whether the recommendation was supported by explicit model ingestion, flux-derived quantitative evidence, perturbation simulation, literature grounding, stated limitations, and an auditable path from user query to model-derived interpretation.

Identification of OGDH as a Target Candidate in Rheumatoid Arthritis Naive B cells

In the rheumatoid arthritis Naive B-cell use case, MechAInistic identified mitochondrial metabolism as a major contributor to the disease-state flux profile. B cells are relevant to rheumatoid arthritis pathogenesis and therapeutic targeting.[21] The disease model showed increased oxidative phosphorylation, citric acid cycle activity, and reactive oxygen species detoxification relative to the healthy reference, suggesting broader mitochondrial rewiring rather than a single isolated reaction change.

MechAInistic evaluated candidate mitochondrial reactions by comparing disease-state and healthy-state fluxes, simulating perturbations, and assessing whether each intervention moved the disease model toward the healthy flux distribution. Initial perturbation tests did not support all candidate mitochondrial targets equally: simulated succinate dehydrogenase inhibition increased the Euclidean distance from 151.60 to 152.44, while fumarate-associated inhibition produced only a small improvement, reducing the distance from 151.60 to 151.44. The system subsequently prioritized 2-oxoglutarate dehydrogenase/OGDH, represented in the model by AKGDm, because it carried disease-state flux but no corresponding healthy-state flux.

MechAInistic proposed that OGDH inhibition could reduce excess citric acid cycle flux and indirectly suppress downstream oxidative phosphorylation and reactive oxygen species detoxification. It nominated Devimistat/CPI-613 because published evidence supports inhibition of OGDH and pyruvate dehydrogenase by CPI-613.[17,22–26] The system

further reasoned that concurrent pyruvate dehydrogenase inhibition may be mechanistically compatible with the disease-to-healthy transition predicted by the model, because the healthy reference state showed reduced mitochondrial TCA/OXPHOS activity. Simulated inhibition preserved the biomass objective at 1.6889 despite a moderate negative correlation between OGDH perturbation and biomass maintenance, suggesting network-level compensation in the model.

Manual validation supported the model-derived quantitative outputs and the literature-supported connection between Devimistat/CPI-613 and OGDH/PDH inhibition. However, validation also showed that Devimistat/CPI-613 is investigational and not FDA-approved for any indication. The literature-retrieval and summarization process also produced at least one title-author mismatch despite retrieving biologically relevant PubMed-indexed sources. Thus, this use case demonstrates MechAInistic's ability to generate a model-grounded mitochondrial therapeutic hypothesis while also showing why human review remains necessary for clinical interpretation and citation fidelity.

Identification of IDH as a Target Candidate in Multiple Sclerosis Th17-cells

In the multiple sclerosis CD4+ Th17 use case, MechAInistic identified NADP-dependent isocitrate dehydrogenase, represented by ICDHy and mapped to the IDH1/IDH2 enzyme family, as the strongest single-target intervention. CD4+ Th17 cells are central to multiple sclerosis pathogenesis and autoimmune inflammation.[27,28] The disease-state model carried forward ICDHy flux of 3.441, whereas the healthy-state model carried no flux through this reaction. Simulated inhibition reduced the Euclidean distance between

disease and healthy flux states from 25.247 to 23.069, corresponding to an 8.6% improvement. ICDHy was not classified as a topological hub, and robustness and dose-response analyses showed no detectable change in the biomass objective across 0–100% inhibition, suggesting a potentially favorable therapeutic window within the model.

The biological interpretation generated by MechAInistic centered on a glutamate-overflow phenotype. In this interpretation, disease-associated ICDHy activity generates α-ketoglutarate that is diverted toward glutamate metabolism rather than continuing through oxidative TCA-cycle activity. This interpretation was consistent with the model-derived increase in glutamate metabolism and reduced citric acid cycle flux relative to the healthy reference. Although the prompt requested a single drug therapy, MechAInistic also explored whether a multi-target perturbation could better restore the healthy flux state. Combined inhibition of ICDHy, r2420, and MALtm improved Euclidean distance to 21.591 and yielded a cosine similarity of 0.984, outperforming ICDHy inhibition alone. Because no single available agent targeted all three nodes, the system retained this result as mechanistic context rather than the final recommendation.

MechAInistic initially stated incorrectly that ICDHy lacked a gene identifier in the model, but its literature-retrieval phase connected ICDHy to the IDH1/IDH2 enzyme family and identified clinically available IDH inhibitors, including ivosidenib and vorasidenib.[29–31] It selected ivosidenib as the primary candidate. Manual review supported the broader IDH-centered mechanism but suggested that vorasidenib may be preferable to ivosidenib in this context because it inhibits both IDH1 and IDH2, whereas

ivosidenib primarily targets IDH1. Literature linking IDH activity, α-ketoglutarate, 2-hydroxyglutarate, and Th17/Treg balance supported the broader target-class hypothesis.[32] However, human review suggested that vorasidenib may be mechanistically preferable to ivosidenib in this context because it inhibits both IDH1 and IDH2, whereas ivosidenib primarily targets IDH1.[33]

This use case illustrates both the strength and limitation of MechAInistic: the system identified a biologically meaningful target class, explored a broader perturbation space, and returned to the monotherapy constraint, but expert review remained necessary for annotation correction and compound-level prioritization. The resulting hypothesis was consistent with literature on Th17 metabolism and TCA-cycle breakpoints.[34,35]

Discussion

We developed MechAInistic as a supervised, tool-grounded agentic system for reasoning over paired constraint-based metabolic models. Across two immune-cell use cases, the system converted natural-language prompts into executable workflows, generated model-derived quantitative evidence, and produced mechanistic therapeutic hypotheses with explicit limitations. These results support a role for large language models not as unconstrained biological answer engines, but as orchestration layers for executable mechanistic modeling workflows.

A central finding is that workflow structure matters. General-purpose LLM systems can generate biologically plausible explanations and, in some cases, useful therapeutic

suggestions. However, plausibility alone is insufficient for model-driven biological reasoning.[36] The key question is whether the final answer can be traced to uploaded models, executable tool calls, quantitative outputs, and literature-supported interpretation. MechAInistic's Architect-Reviewer-Task architecture addresses this need by separating planning, execution, intermediate review, literature retrieval, and final synthesis.

The Architect-Reviewer pattern was motivated by a broader challenge in LLM-enabled biomedical reasoning: language models can generate fluent but unsupported biological claims unless constrained by external verification.[36] Prior work has shown that biomedical associations generated by LLMs can be improved when terminology is validated against specialized ontologies or databases, factual claims are checked against biomedical literature, and outputs are reviewed by independent models or verification workflows.[36–42] MechAInistic applies these principles to metabolic modeling by requiring the Architect's proposed analytical path and final interpretation to pass through executable GSMN tools, Reviewer-mediated critique, and literature-grounded synthesis.

The two use cases illustrate MechAInistic's role in therapeutic hypothesis generation. In rheumatoid arthritis Naive B cells, the system identified mitochondrial metabolic rewiring and nominated Devimistat/CPI-613 as an investigational OGDH-centered hypothesis. In multiple sclerosis CD4+ Th17 cells, it identified ICDHy/IDH as a candidate metabolic control point and proposed IDH inhibition as a therapeutic hypothesis. These outputs should not be interpreted as validated therapeutic recommendations; rather, they are model-grounded hypotheses for expert review and

experimental validation. We compared these with Claude's output, which were also based on the metabolic models. In RA, Devimistat may be a stronger prediction because it more selectively targets central mitochondrial TCA-cycle metabolism, while Etomoxir (the top candidate proposed by Claude) has known off-target effects and lower target specificity.[43] However, for MS, Ivosidenib represents an interesting mechanistic immunometabolic repurposing candidate, Claude identified dimethyl fumarate which is already FDA-approved for relapsing MS.[44]

Several limitations remain. First, MechAInistic depends on the quality and biological validity of uploaded metabolic models. Missing reactions, genes, metabolites, constraints, or annotations are not automatically repaired, so downstream analyses remain bounded by the input models. In this study, the RA and MS disease-state models were reconstructed from bulk RNA-seq datasets, and the models themselves require separate documentation and validation. Second, although tool calls provide a grounding layer, LLM behavior can vary across runs, model versions, providers, and inference settings. Reproducible use therefore requires reporting uploaded models, prompt text, objective functions, LLM configuration, solver, tool versions, and execution date.

Third, literature retrieval and summarization require verification. Manual validation identified at least one citation-level mismatch despite retrieval of biologically relevant literature, reinforcing the need to verify citations, drug-target claims, and mechanistic interpretations before downstream use. Such errors are consistent with broader limitations of LLM-mediated summarization and factuality across biomedical and scientific

contexts.[45–49] Fourth, the current interface supports self-contained analyses rather than open-ended multi-turn scientific conversations, and complex workflows require substantial token usage; token counts are provided in Supplementary Table S16. Finally, MechAInistic depends on external biological databases and literature services, including BiGG[50], DGIdb[51], MyGeneInfo[52], and PubMed[53], so incomplete or unavailable resources may degrade drug-target mapping or literature-grounded interpretation.

Together, these findings establish MechAInistic as a step toward natural-language access to executable mechanistic biology. The platform does not replace expert computational modeling or experimental validation. Instead, it provides an auditable interface between biological questions, GSMNs, computational tools, and literature-supported interpretation.

MechAInistic establishes a foundation for natural-language-driven reasoning over constraint-based metabolic models, and several avenues remain open for further development. Continued work will focus on broadening the range of supported analyses, evaluating the system across additional cell types and disease contexts, and incorporating advances in language model architectures as they become available. Because the language model field is continually advancing, we envision future updates that automatically route the user's query to the most cost-effective language model available at the time. We also anticipate ongoing refinement of the multi-agent framework informed by community feedback and user experience. Additional platform extensions are under active development and will be described in subsequent work.

Methods and System Design

This section describes the implementation of MechAInistic and the procedures used to generate and evaluate the results reported above. Detailed materials not required for interpreting the main Results, including full prompt templates, tool-parameter definitions, workflow-state variables, software dependencies, accession tables, and comparator outputs, are provided in the Supplementary Information.

System Implementation and User Input

MechAInistic is implemented as a web-accessible system for executing natural-language queries over paired constraint-based metabolic models. Users provide two COBRA-compatible metabolic models encoded in JSON format: a current-state model and a target-state model. In the use cases reported here, the current-state model corresponded to a disease-state immune-cell model, and the target-state model corresponded to a matched healthy reference model. Users also provide a natural-language biological question through the web interface.

Once MechAInistic receives the input files and the question, the user may submit the analysis request. The interface also exposes configurable settings for the LLMs and metabolic model objective functions, allowing users to adjust key analysis parameters before execution. By default, MechAInistic populates the LLM backend setting with the National Research Platform's (NRP) OpenAI-compatible endpoint, which is free for academic researchers to use upon signing up with NRP.[54] Alternatively, users can provide

a different URL endpoint (e.g., OpenAI, Anthropic) and API key, provided the endpoint is publicly accessible; additional information on setting up MechAInistic with local LLMs is available in the Supplementary Information section titled “LLM Configuration.” Figure 19 shows the adjustable configuration settings and the web interface after the modeling files have been uploaded, and the system is ready to process the request.

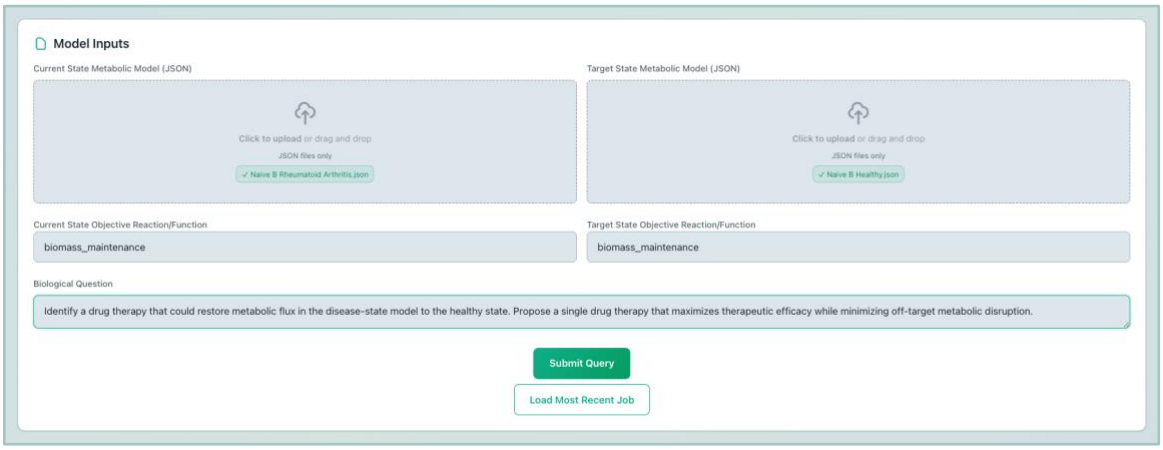


Figure 19: The adjustable LLM settings and model upload section of MechAInistic. The current-state (disease state) and target-state (healthy state) models have been uploaded, and the system is ready to process requests.

After submission, MechAInistic creates a session-specific analysis state that includes the uploaded models, user query, selected objective functions, LLM configurations, available tools, accumulated tool calls, summarized tool outputs, and the final report state. Reports are rendered in the browser and can be downloaded in Markdown format.

Architecturally, MechAInistic is implemented as a modular, containerized system, where the backend, frontend, a PostgreSQL database, and LLM host can each run as

independent services on separate servers or nodes. This architecture allows the platform to be deployed across one or more machines, provided that each component can communicate with its required dependency. In the current implementation, the main backend communicates with several external services during execution: the LLM host, external databases (PubMed, DGIdb, MyGeneInfo, etc.), and a PostgreSQL database. This separation supports flexible deployment on local or distributed hardware while preserving the same high-level workflow exposed through the web interface. Figure 3 illustrates the system's deployment architecture.

Agent roles and workflow control

MechAInistic uses three role-specialized agents: Architect, Reviewer, and Task. The Architect is responsible for generating an initial plan, synthesizing tool calls, and constructing a final report, while the Reviewer will provide feedback and suggestions to improve the analysis and interpretation during each step. Additionally, the Reviewer will score the analysis and serve as a stopping point to prevent the Architect from finalizing it without sufficient quantitative support from literature or the metabolic models. The Task agent supports two auxiliary functions: translating biological hypotheses into PubMed-oriented search queries and summarizing raw tool outputs into concise plain-text summaries that preserve quantitative values, identifiers, caveats, and errors for downstream reasoning. In this way, MechAInistic provides coordinated multi-agent execution while helping preserve the consistency and fidelity of the analytical process and its associated outputs.

After users submit their query, MechAInistic initializes a session-specific analysis state containing the uploaded data and available tools and creates placeholders for completed tool calls, summarized results, and Reviewer feedback (Figure 20A). The workflow follows a multi-stage, multi-agent design with "outer" and "inner" control systems. The outer loop has a non-user-configurable maximum of 5 retries to prevent infinite execution of the Agents; the inner loop has a maximum of 10 iterations, each executing 1 to 6 tool calls, to prevent infinite loops. The retry logic, tool calling state management, and agent outputs are defined in Supplementary Tables S4-S7. The workflow then proceeds through three main phases: Plan Development, Iterative Synthesis and Execution, and Report Generation.

First, during **Plan Development** (Figure 20B), the Architect model receives the user's biological query, a list of available analysis tools, and instructions to generate an experimental, in-silico workflow that, when executed, answers the user's question. Once the plan is constructed, the Reviewer evaluates the plan and returns feedback in the following format: (a) a score on a scale from 1 to 10, (b) feedback describing the reason for the provided score, and (c) a list of suggestions to improve the plan. The Reviewer assessment is based on five components, each worth two points: Completeness (does the plan address all aspects of the user's question?), Tool Appropriateness (are the selected tools suitable for the task?), Logical Sequence (are the steps performed in the correct order?), Feasibility (can the proposed plan be completed?), and Error Handling (are failure cases addressed?). The reviewer assigns each category a score of 0 (not answered), 1 (partially answered), or 2 (fully answered). The scores are summed, and if

the result is 7 or lower, the plan is considered a failure and must be regenerated. In this case, the Architect will utilize the Reviewer's provided suggestions when generating a new plan. If the plan passes, the workflow continues to the Iterative Synthesis and Execution phase.

During the **Iterative Synthesis and Execution** phase (Figure 20C), the Architect is provided with their approved plan and the Reviewer's feedback (regardless of whether the original plan passed or failed), and begins synthesizing tool calls as defined by the analysis. These tools are returned to MechAInistic's System Orchestrator, which parses the Architect's response and sequentially executes the tool calls. After execution, the Task model is used to summarize results; token usage decreases by approximately 50% when structured JSON results are summarized into plain text. After summarization, the Reviewer acts as an intermediary to determine if additional tool calls are required to answer the user's query. The Reviewer provides the following feedback: (a) a score on a scale from 1 to 10, (b) feedback describing what has been accomplished thus far, and (c) a list of requirements for improvement, including what is missing, potential next steps, or refinements for the analysis (Supplementary Table S8). Because the Iterative Synthesis phase can call multiple tools during its cycle, the scoring is more lenient to provide the Architect additional chances for correction. The tool calling results are assigned a score of 1 to 3 if there was a malfunction (e.g., software crash, wrong format, incomplete execution); 4 or 5 if the tool was executed correctly but the results are unusable (e.g., nonsensical output format or results do not correlate to the goal); 6 or 7 if the tool was executed properly and the results appear to be related to the goal but require additional

investigation; or an 8 to 10 if the tool was executed correctly, the results are biologically plausible, and ready to be used in the final report generation. During this phase, there is a maximum limit of 10 loops to prevent infinite tool calling. If the Architect signals no further steps are needed, or if the Reviewer determines that the analysis is complete, the Iterative Synthesis loop terminates. As described previously, during this phase, the initial plan will be expanded if it proves insufficient to answer the user's query.

Importantly, low-scoring Iterative Synthesis phases will not increment the outer loop's retry count, as both the Architect and Reviewer must independently indicate that the result is finalized. It is only at this point that the entire tool calling results are provided to the Reviewer, where a second, broader determination is made. At this point, if the Reviewer determines that the results are not complete, the retry counter is incremented.

In the **Result Evaluation** phase (Figure 20D), the Reviewer is provided with all summarized tool call results and decides whether sufficient information has been obtained to answer the user's query. If it is determined that yes, all components are answered, and the initial plan is complete, the workflow continues to the Report Generation phase. Otherwise, the outer loop retry count is incremented by one, and the Iterative Synthesis phase is re-entered.

Finally, in the **Report Generation** phase (Figure 20E), the Architect is given the entire workflow state, including the user's query, the original plan, all summarized tool call results, and synthesizes a final report of all findings, including their mechanistic

interpretation. The report is displayed in the browser and can be downloaded in Markdown format.

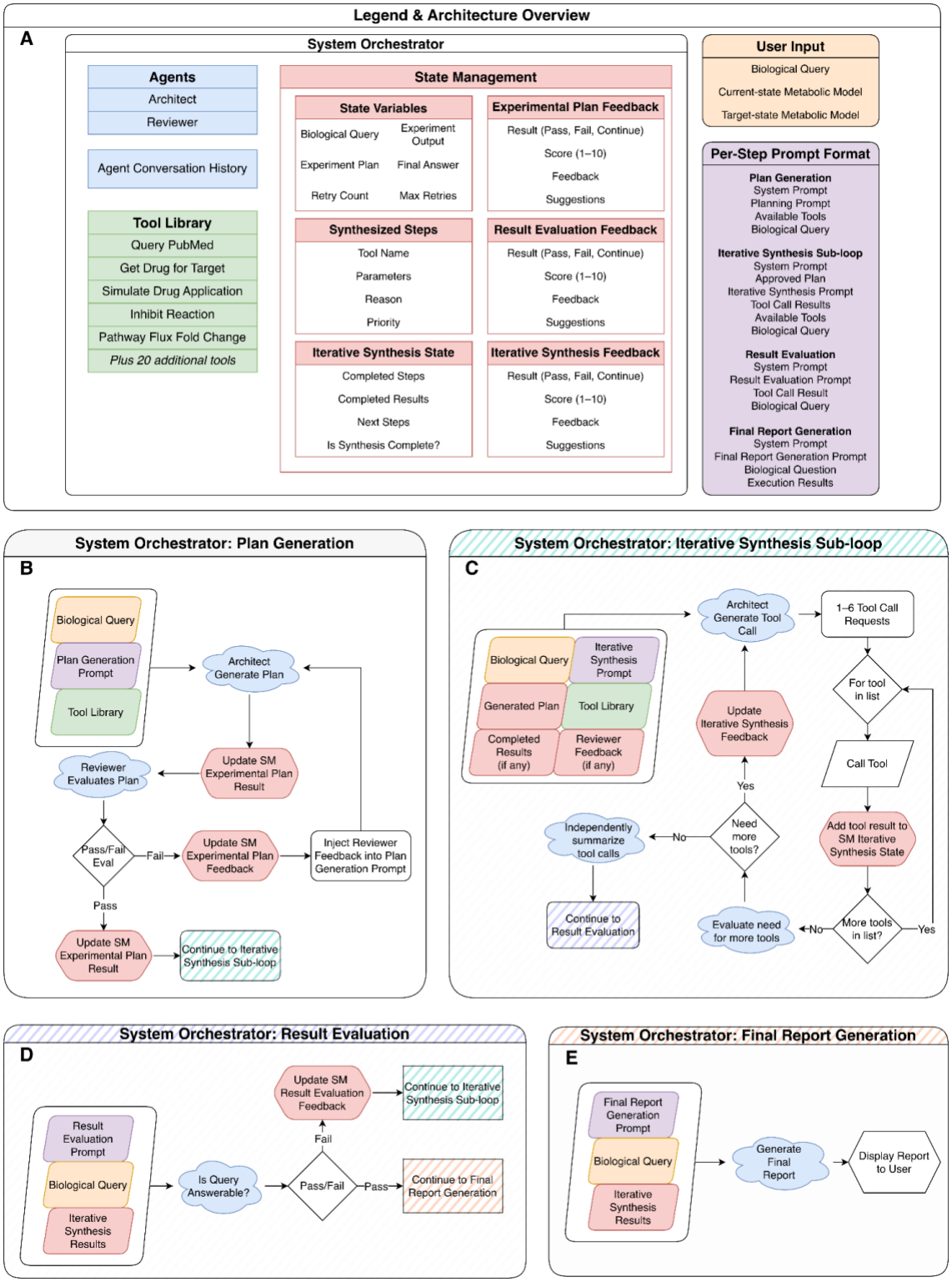

Legend & Architecture Overview
A
System Orchestrator
Agents
Architect
Reviewer
Agent Conversation History
Tool Library
Query PubMed
Get Drug for Target
Simulate Drug Application
Inhibit Reaction
Pathway Flux Fold Change
Plus 20 additional tools
State Management
State Variables
Biological Query
Experiment Output
Experiment Plan
Final Answer
Retry Count
Max Retries
Experimental Plan Feedback
Result (Pass, Fail, Continue)
Score (1–10)
Feedback
Suggestions
Synthesized Steps
Tool Name
Parameters
Reason
Priority
Result Evaluation Feedback
Result (Pass, Fail, Continue)
Score (1–10)
Feedback
Suggestions
Iterative Synthesis State
Completed Steps
Completed Results
Next Steps
Is Synthesis Complete?
Iterative Synthesis Feedback
Result (Pass, Fail, Continue)
Score (1–10)
Feedback
Suggestions
User Input
Biological Query
Current-state Metabolic Model
Target-state Metabolic Model
Per-Step Prompt Format
Plan Generation
System Prompt
Planning Prompt
Available Tools
Biological Query
Iterative Synthesis Sub-loop
System Prompt
Approved Plan
Iterative Synthesis Prompt
Tool Call Results
Available Tools
Biological Query
Result Evaluation
System Prompt
Result Evaluation Prompt
Tool Call Result
Biological Query
Final Report Generation
System Prompt
Final Report Generation Prompt
Biological Question
Execution Results
System Orchestrator: Plan Generation
B
Biological Query
Plan Generation Prompt
Tool Library
Architect Generate Plan
Update SM Experimental Plan Result
Reviewer Evaluates Plan
Pass/Fail Eval
Fail
Update SM Experimental Plan Feedback
Inject Reviewer Feedback into Plan Generation Prompt
Pass
Update SM Experimental Plan Result
Continue to Iterative Synthesis Sub-loop
System Orchestrator: Iterative Synthesis Sub-loop
C
Biological Query
Iterative Synthesis Prompt
Generated Plan
Tool Library
Completed Results (if any)
Reviewer Feedback (if any)
Architect Generate Tool Call
1–6 Tool Call Requests
For tool in list
Call Tool
Add tool result to SM Iterative Synthesis State
More tools in list?
Yes
No
Update Iterative Synthesis Feedback
Yes
Need more tools?
No
Independently summarize tool calls
Continue to Result Evaluation
Evaluate need for more tools
No
Yes
System Orchestrator: Result Evaluation
D
Result Evaluation Prompt
Biological Query
Iterative Synthesis Results
Is Query Answerable?
Pass/Fail
Fail
Update SM Result Evaluation Feedback
Continue to Iterative Synthesis Sub-loop
Pass
Continue to Final Report Generation
System Orchestrator: Final Report Generation
E
Final Report Generation Prompt
Biological Query
Iterative Synthesis Results
Generate Final Report
Display Report to User

Figure 20: Overview of the MechAInistic system architecture. Figure 2A) The legend and diagram overview. Colors indicated by this panel are replicated throughout the figure. The primary agents include the Architect and Reviewer, each with their own isolated conversation history; the Task agent is not depicted here as its role is limited to processing the input and output of specific tool calls. Figure 2B) The Plan Generation information flow. The Architect agent receives the user's Biological Query, the Plan Generation Prompt, and a list of tools plus their descriptions. The Reviewer evaluates the plan; if the plan succeeds, State Management is updated; otherwise, the Architect re-attempts plan generation. Figure 2C) The Iterative Synthesis Sub-loop. The architect receives the user's biological query, the Iterative Synthesis Prompt, the generated plan, all available tools, completed results (if available), and Reviewer agent feedback (if available). Figure 2D) The Result Evaluation process. The Reviewer agent receives the Result Evaluation Prompt, the user's biological query, and all results from the Iterative Synthesis Sub-loop. If the Reviewer agent indicates that the results are not sufficient to answer the user's query, the Iterative Synthesis Sub-loop is re-entered (Panel C); otherwise, a final report will be generated. Figure 2E) Final Report Generation. The Architect receives the Final Generation Report, the user's Biological Query, and all Iterative Synthesis Results. The Architect synthesizes a structured report in Markdown format, which is displayed to the user.

LLM Configuration and Prompt Management

The platform employs three large language models. Each model is configured at runtime using a common parameter set: model name, endpoint host, API key, temperature, top-p sampling value, and an optional random seed. This allows MechAInistic to make use of any LLM provider across any combination of hosts. When the user submits their biological question, these parameters are sent to the backend and passed to an OpenAI-compatible client interface, enabling the system to interact with externally hosted models via a standardized API. As described previously, we are using NRP-hosted language models as the default LLM provider. We chose separate models to serve as the Architect and Reviewer to incorporate model diversity into our evaluation pipeline, leverage their distinct training methodologies, and exploit their different token output probabilities. The

Task model was chosen for its fast inference speed. The temperature for each agent was set to 0.5 to balance output consistency with some degree of variability during tool calls, reviewing, and report generation.

The Architect, Reviewer, and Task agents are given deliberately asymmetric, role-specific instructions. The Architect is framed as a strategy-oriented analytical agent, emphasizing methodological transparency, evidence-based planning, risk anticipation, tool orchestration, and failure-mode mitigation. The Reviewer is framed as a rigorous, yet constructive, evaluator, emphasizing the identification of methodological weaknesses, balancing strengths and limitations, and providing actionable feedback. The Task agent's function is two-fold; first, it finds academic publications from PubMed and translates complex biological hypotheses into optimized search queries that balance precision with recall, ensuring relevance and adequate result coverage. Second, it summarizes tool call results from a JSON format into plain text. This prompt asymmetry operationalizes the division of labor between analytical generation, critical validation, and retrieval-augmented generation.

The execution layer exposes a curated set of metabolic analysis tools through a central registry in the agent pipeline. The tools are registered in the system as callable functions and are made available to the Architect agent during both the planning and iterative synthesis phases, so the Architect is explicitly informed of the available computational actions and their intended use, rather than relying solely on prior knowledge. For each available tool, MechAInistic dynamically extracts the tool name, a 1-sentence

description, and parameters required to call the tool. Examples of these parameters include a reaction, pathway, gene name, or a drug to apply to the metabolic model. The toolset supports several categories of analysis, including metabolic model simulation, reaction- and pathway-level flux analysis, graph-based interrogation of metabolic networks, intervention and perturbation analysis, metabolite lookup, gene knockout simulation, drug-target discovery, and literature retrieval.

To manage prompts for each phase, templates are organized by functional stage and agent role, then instantiated through placeholder substitution and summarized execution results at runtime. This design separates control logic from prompt text and supports role-specific prompting for planning, execution, evaluation, and final answer generation. Within each phase, a System Prompt primes the respective Agent for its task. These prompts are intended to serve as a computational methods template rather than a simple instruction; they require explicit treatment of assumptions, boundary conditions, workflow modularity, validation checkpoints, failure modes, sensitivity analysis, and evidence integration. Supplementary Table S5 summarizes the prompts for each phase and their expected outputs. The full prompt templates are provided in Supplementary Information Note 13 for transparency and reproducibility.

Tool registry and executable analysis layer

The execution layer exposes a curated registry of metabolic-analysis tools to the Architect agent. Each tool is registered with a name, short description, required parameters, optional parameters, and expected input types. During Plan Development and

Iterative Synthesis, MechAInistic provides the Architect with the available tool descriptions so that proposed workflows are constrained to executable operations rather than unconstrained language-model reasoning. When the Architect proposes a tool call, it must return the exact tool name and complete parameter assignments required for execution.

The backend parses the Architect's proposed tool calls, validates required parameters, injects the session-specific model/cache key, and executes the corresponding function. This session key identifies the uploaded current-state and target-state models and is handled by the backend rather than exposed to the LLM. Raw tool outputs are stored in the session state, then summarized by the Task agent before being returned to the Architect and Reviewer for subsequent reasoning. This design preserves the complete computational record while reducing the token burden of passing large structured outputs between agents.

The tool registry supports several categories of analysis: metabolic-model simulation, flux-distance calculation, reaction-level flux queries, pathway-level comparison, graph/network analysis, robustness analysis, gene and reaction perturbation, drug-effect simulation, dose-response analysis, drug-target lookup, metabolite lookup, and literature retrieval. These tools allow MechAInistic to move from a natural-language query to model-derived quantities, candidate reaction prioritization, perturbation testing, drug mapping, and literature-grounded interpretation.

For the use cases reported here, the executed workflows used tools for baseline flux comparison, pathway-level dysregulation, reaction prioritization, robustness evaluation, simulated inhibition, dose-response analysis, drug-target lookup, and PubMed retrieval. Full tool descriptions, parameter definitions, and representative tool categories are provided in Supplementary Tables S3 and S4.

Constraint-based modeling backend

The primary GSMN software stack is built around CobraPy, the HiGHS[55] solver, and additional supporting libraries such as NetworkX for graph-based analysis and Pandas for flux and result processing (Table 6, Supplementary Table S13).

Table 6: Constraint-based modeling software and analysis configuration

GSMN Component	System Interpretation
Core Modeling Library	COBRApy
LP Solver	HiGHS
Model representation	JSON format
Primary Simulation Type	Flux Balance Analysis
Additional analyses	Reaction/pathway flux comparison, robustness analysis, gene/reaction knockout, graph analysis, drug-effect simulation, dose-response, flux-transition optimization
Objective specification	User-supplied at runtime; can be changed later with a dedicated tool by the Architect Agent

GSMN Component	System Interpretation
Constraint handling	Inherited from the model; can be adjusted later through tool-specific modifications by the Architect Agent

The dominant analysis paradigm in the current system is flux balance analysis, which serves as the main simulation routine for each metabolic model. In addition to standard steady-state optimization, MechAInistic includes reaction-level and pathway-level flux comparisons, robustness (Phenotype Phase Plane) analysis using COBRApy's production envelope function[56], and perturbation analysis via single-gene, single-reaction, double-gene, and double-reaction knockouts. The repository also provides graph-based analyses of the metabolic models and intervention-oriented routines such as simulated drug effects, dose-response analysis, and optimization of reaction modifications to reduce flux-distance between disease and target states.

Metabolic Model Reconstruction and Use-case Setup

We evaluated MechAInistic using two paired immune-cell metabolic model use cases. (1) Naive B models in a Healthy (as described previously)[57] and Rheumatoid Arthritis (RA) disease state, and (2) CD4+ Th17 models in a Healthy (as described previously)[58] and Multiple Sclerosis (MS) disease state. To reconstruct the RA and MS disease models, bulk RNA-seq data were obtained from the NCBI Gene Expression Omnibus.[59] The Naive B RA datasets were very limited, and only a single study containing four samples of bulk RNA-seq RA data was identified.[60] For the MS reconstruction, 19 samples from a single study were identified. All samples were preprocessed using AutoRNAseq[61], which

aligned the data to the Homo Sapiens reference genome (Ensembl Release version 115[62]) using the STAR[63] aligner and quantified reads using Salmon[64]. Following preprocessing, exchange reactions for both the RA and MS conditions were constrained using appropriate disease-specific constraints, as well as lower and upper bounds. The model reconstruction was performed using COMO[57], a Jupyter Notebook-based tool designed to automate and ensure reproducibility in constraint-based metabolic model reconstruction, with Recon3D[12] as the reference model. The Supplementary Material section titled "Constraint-Based Model Reconstruction" contains details for all accession values, exchange reactions, and reaction bounds for model reconstruction.

The same natural-language prompt was used for both use cases: "Identify a drug therapy that could restore metabolic flux in the disease-state model to the healthy state. Propose a single drug therapy that maximizes therapeutic efficacy while minimizing off-target metabolic disruption." The disease and cell-type labels were not included in the prompt and were available to MechAInistic only through the uploaded model context.

Comparator LLM Evaluation

To compare MechAInistic with general-purpose LLM workflows, we submitted the same paired metabolic-model files and the same drug-therapy prompt to Anthropic's Claude, OpenAI's ChatGPT, Microsoft's Copilot, and Google's Gemini. Each system was asked to identify a single therapy that would restore disease-state metabolic flux toward the healthy reference while minimizing off-target metabolic disruption. Comparator outputs

were saved as standalone reports and evaluated using the nine-axis rubric described below.

Because general-purpose LLM systems differ in file ingestion capabilities, code execution environments, tool access, context length limits, and access to external literature, we evaluated the returned artifacts rather than assuming equivalent computational access. The assessment, therefore, focused on whether each output provided evidence that the uploaded models were used, whether quantitative model-derived values were reported, whether a specific therapy was recommended, whether literature support was traceable, and whether the reasoning path was reproducible and auditable.

The identity of the recommended drug was recorded as a descriptive outcome, but was not treated as the primary measure of correctness. Because the correct therapeutic answer cannot be established from computational analysis alone, the primary comparison focused on model grounding, quantitative evidence, methodological traceability, task completion, and explicit recognition of limitations. Per-system scoring justifications and full comparator outputs are provided in Supplementary Information Notes 10 and 14-24

Nine-axis Evaluation Rubric

MechAInistic and comparator outputs were assessed using a nine-axis rubric designed to distinguish plausible biological narrative from model-grounded computational reasoning. The rubric evaluated whether each system produced an answer that was not only

biologically plausible but also traceable to the supplied metabolic models, supported by quantitative evidence, and aligned with the user's requested task.

The nine evaluation axes were:

1. **Model fidelity:** evidence that the uploaded metabolic models were loaded, parsed, and used in the analysis.
2. **Methodological reproducibility:** inclusion of named tools, solvers, scoring rules, executable analysis steps, or other information needed to reproduce the workflow.
3. **Literature evidence:** use of traceable peer-reviewed sources, including PMIDs, DOIs, or clearly identifiable references where applicable.
4. **Computational evidence:** inclusion of flux values, perturbation outputs, distance metrics, dose-response results, or other quantitative model-derived outputs.
5. **Direct relevance:** adherence to the single-drug flux-restoration task rather than drifting into general disease biology or unrelated therapeutic framing.
6. **Question completion:** whether the system returned a specific single therapy in response to the prompt.
7. **Structured review or agentic control:** evidence of a review, checkpointing, or multi-agent control process before final synthesis.
8. **Execution feasibility and rerunnability:** completion without file-ingestion failure, truncation, unavailable context, or undocumented execution constraints.
9. **Human-review traceability:** explicit indication of manual review, sign-off, correction, or identifiable points where human review could be applied.

Each axis was scored qualitatively as full, partial, or failed based on pre-specified criteria. Drug identity was recorded descriptively but was not used as the main correctness criterion, because the computational analyses generate hypotheses rather than validated therapeutic answers. The primary evaluation focused on whether each recommendation was supported by model-grounded computation, quantitative evidence, methodological traceability, and explicit limitations.

Manual validation of MechAInistic outputs

Manual validation was performed after MechAInistic generated the final reports for each use case. The goal of this validation was not to prove therapeutic correctness, but to determine whether the model-derived results, literature-supported claims, and final interpretations were traceable and internally consistent.

Validation focused on five categories: (1) whether reported quantitative values matched the underlying tool-call outputs; (2) whether reaction identifiers, pathway names, drug names, and model states were transcribed correctly; (3) whether literature cited by MechAInistic could be traced to real publications; (4) whether the cited literature supported the specific mechanistic claim made in the report; and (5) whether the final therapeutic interpretation followed from the model-derived and literature-supported evidence.

Validation outcomes were classified as supported, partially supported, unsupported, or requiring expert interpretation. Supported findings were retained in the manuscript.

Partially supported findings were retained only with explicit caveats. Unsupported claims, citation mismatches, annotation-related errors, or cases requiring expert reinterpretation were reported as failure modes rather than removed from the analysis. This procedure was used to distinguish model-grounded therapeutic hypotheses from validated therapeutic recommendations.

External services and reproducibility

The analytical pipeline calls external scientific services during execution, including BiGG[50], DGIdb[51], MyGeneInfo[52], and PubMed[53]. These services support reaction and metabolite annotation, drug-gene interaction lookup, gene identifier resolution, and literature retrieval. MechAInistic also maintains local infrastructure for user/session data, drug-target caching, and literature-embedding search. External-service dependencies, rate limits, caching behavior, and software dependencies are described in the Supplementary Information.

Acknowledgements

This work was supported by the National Institutes of Health under Grant No. #R35GM119770, and the University of Nebraska-Lincoln Grand Challenges Catalyst Award to Tomáš Helikar.

We are thankful to Skylar Loecker and Robert Moore II for their valuable discussions during the methodology and investigation phases of this project.

Author Contributions

Conceptualization, T.H.; methodology, J.L., N.P., W.B., P.P., A.A.H., T.H.; Investigation, J.L., P.P., B.L.P., A.A.H., T.H.; writing—original draft, J.L., A.A.H., T.H.; writing—review & editing, J.L., P.P., B.L.P., A.A.H., T.H.; funding acquisition, T.H.; resources, T.H.; supervision, P.P., A.A.H., T.H.;

Declaration of Interests

T.H is a founder and shareholder of Discovery Collective, Inc., and ImmuNovus, Inc.

Declaration of Generative AI and AI-assisted Technology

During the preparation of this work, the authors used MechAInistic (an LLM-guided multi-agent system built on large language models including Qwen, Kimi, and Gemma) to execute all constraint-based metabolic analyses, generate analytical plans, interpret flux-based results, and produce structured reports for the use cases described in this manuscript. Claude (Anthropic) was used to reformat and restructure these outputs into manuscript-ready sections. All AI-generated content was reviewed, edited, and validated by the authors, who take full responsibility for the scientific content.

References

1. Benedicto, X., Flobak, Å., Ponce-de-Leon, M. & Valencia, A. Constraint based modeling of drug induced metabolic changes in a cancer cell line. *Npj Syst. Biol. Appl.* **11**, 111 (2025).

2. Bordbar, A., Monk, J. M., King, Z. A. & Palsson, B. O. Constraint-based models predict metabolic and associated cellular functions. *Nat. Rev. Genet.* **15**, 107–120 (2014).

3. Lubbock, A. L. R. & Lopez, C. F. Programmatic modeling for biological systems. *bioRxiv* (2024).

4. Puniya, B. L. & others. Perspectives on computational modeling of biological systems and the significance of the SysMod community. *Bioinforma. Adv.* **4**, vbae090 (2024).

5. Heirendt, L. & others. Creation and analysis of biochemical constraint-based models using the COBRA Toolbox v3.0. *Nat. Protoc.* **14**, 639–702 (2019).

6. Lapi, F. & others. COBRAxy: constraint-based metabolic modeling in Galaxy. *Bioinformatics* (2025).

7. Ebrahim, A., Lerman, J. A., Palsson, B. O. & Hyduke, D. R. COBRApy: COnstraints-Based Reconstruction and Analysis for Python. *BMC Syst. Biol.* **7**, 74 (2013).

8. Yasemi, M. & Jolicoeur, M. Modelling Cell Metabolism: A Review on Constraint-Based Steady-State and Kinetic Approaches. *Processes* **9**, 467 (2021).

9. Bennett, O. Benchmarking Constraint-Based Metabolic Models: How to Validate FBA Predictions Against Experimental Flux Data. *PlantChemica* (2026).

10. Ng, C. & others. Constraint-Based Reconstruction and Analyses of Metabolic Models: Open-Source Python Tools and Applications to Cancer. *Front. Oncol.* **12**, 829524 (2022).

11. Machado, D. & Herrgård, M. J. Systematic evaluation of methods for integration of transcriptomic data into constraint-based models. *PLoS Comput. Biol.* **10**, e1003580 (2014).

12. Brunk, E. & others. Recon3D enables a three-dimensional view of gene variation in human metabolism. *Cell Syst.* **6**, 1–14 (2018).

13. Wehling, L., Singh, G., Mulyadi, A. W., & others. Talk2Biomodels: AI agent-based open-source LLM initiative for kinetic biological models. *BMC Bioinformatics* **26**, 1–14 (2025).

14. Alam, T. & Roy, S. From Prompt to Pipeline: Large Language Models for Scientific Workflow Development in Bioinformatics. *bioRxiv* (2025).

15. Rao, R. & others. Generalist biological artificial intelligence in modeling the language of life. *Nat. Biotechnol.* (2026).

16. Bu, D. *et al.* Empowering AI data scientists using a multi-agent LLM framework with self-evolving capabilities for autonomous, tool-aware biomedical data analyses. *Nat. Biomed. Eng.* 1–16 (2026) doi:10.1038/s41551-026-01634-6.

17. Philip, P. A. *et al.* A Phase III Open-Label Trial to Evaluate Efficacy and Safety of CPI-613 Plus Modified FOLFIRINOX (MFFX) versus FOLFIRINOX (FFX) in Patients with Metastatic Adenocarcinoma of the Pancreas. *Future Oncol.* **15**, 3189–3196 (2019).

18. Merchant, S. L., Culos, K. & Wyatt, H. Ivosidenib: IDH1 Inhibitor for the Treatment of Acute Myeloid Leukemia. *J. Adv. Pract. Oncol.* **10**, 494–500 (2019).

19. Fontaine, E. Metformin-Induced Mitochondrial Complex I Inhibition: Facts, Uncertainties, and Consequences. *Front. Endocrinol.* **9**, 753 (2018).

20. Reinhart, G. A. Vitamin D analogs: novel therapeutic agents for cardiovascular disease? *Curr. Opin. Investig. Drugs* **5**, 947–951 (2004).

21. Nakken, B. *et al.* B-cells and their targeting in rheumatoid arthritis — Current concepts and future perspectives. *Autoimmun. Rev.* **11**, 28–34 (2011).

22. Nguyen, T. T. *et al.* OGDH and Bcl-xL loss causes synthetic lethality in glioblastoma. *JCI Insight* **9**, e172565 (2024).

23. Udumula, M. P. *et al.* Targeting mitochondrial metabolism with CPI-613 in chemoresistant ovarian tumors. *J. Ovarian Res.* **17**, 226 (2024).

24. Khan, H. Y. *et al.* Targeting Cellular Metabolism With CPI-613 Sensitizes Pancreatic Cancer Cells to Radiation Therapy. *Adv. Radiat. Oncol.* **8**, 101122 (2023).

25. Reddy, V. B. *et al.* In Vitro and In Vivo Metabolism of a Novel Antimitochondrial Cancer Metabolism Agent, CPI-613, in Rat and Human. *Drug Metab. Dispos. Biol. Fate Chem.* **50**, 361–373 (2022).

26. Xu, R. *et al.* Potassium ion efflux induces exaggerated mitochondrial damage and non-pyroptotic necrosis when energy metabolism is blocked. *Free Radic. Biol. Med.* **212**, 117–132 (2024).

27. Moser, T., Akgün, K., Proschmann, U., Sellner, J. & Ziemssen, T. The role of TH17 cells in multiple sclerosis: Therapeutic implications. *Autoimmun. Rev.* **19**, 102647 (2020).

28. Puniya, B. L. *et al.* A Mechanistic Computational Model Reveals That Plasticity of CD4+ T Cell Differentiation Is a Function of Cytokine Composition and Dosage. *Front. Physiol.* **9**, 878 (2018).

29. Fang, Y. *et al.* Isocitrate Dehydrogenase Mutations in Cancer: From Bench to Bedside Applications. https://doi.org/10.1002/mco2.70732 doi:10.1002/mco2.70732.

30. Ding, Y.-H. *et al.* Inhibition of SLC25A10 promotes cellular senescence and impedes hepatocellular carcinoma progression. *Transl. Cancer Res.* **14**, 4939–4954 (2025).

31. Anselme, M., He, H., Lai, C., Luo, W. & Zhong, S. Targeting mitochondrial transporters and metabolic reprogramming for disease treatment. *J. Transl. Med.* **23**, 1111 (2025).

32. Xu, T. *et al.* Metabolic control of TH17 and induced Treg cell balance by an epigenetic mechanism. *Nature* **548**, 228–233 (2017).

33. Mellinghoff, I. K. *et al.* Vorasidenib in IDH1- or IDH2-Mutant Low-Grade Glioma. *N. Engl. J. Med.* **389**, 589–601 (2023).

34. Wagner, A. *et al.* Metabolic modeling of single Th17 cells reveals regulators of autoimmunity. *Cell* **184**, 4168-4185.e21 (2021).

35. Kaushik, D. K. & Yong, V. W. Metabolic needs of brain-infiltrating leukocytes and microglia in multiple sclerosis. *J. Neurochem.* **158**, 14–24 (2021).

36. Li, B., Saini, A. K., Hernandez, J. G. & Moore, J. H. Agentic AI and the rise of in silico team science in biomedical research. *Nat. Biotechnol.* **44**, 711–725 (2026).

37. Zhao, W. *et al.* An agentic system for rare disease diagnosis with traceable reasoning. *Nature* **651**, 775–784 (2026).

38. Gao, Z. *et al.* Examining the Potential of ChatGPT on Biomedical Information Retrieval: Fact-Checking Drug-Disease Associations. *Ann. Biomed. Eng.* **52**, 1919–1927 (2024).

39. Wang, Z. *et al.* GeneAgent: self-verification language agent for gene-set analysis using domain databases. *Nat. Methods* **22**, 1677–1685 (2025).

40. Hamed, A. A. & Rocha, L. Protocol for Evaluating ChatGPT in Biomedical Association Generation and Verification Using a RAG-Enabled, Cross-Model Majority Voting Workflow. *STAR Protoc.* https://doi.org/10.1016/j.xpro.2026.104533 (2026).

41. Hamed, A. A., Crimi, A., Misiak, M. M. & Lee, B. S. From knowledge generation to knowledge verification: examining the biomedical generative capabilities of ChatGPT. *iScience* **28**, 112492 (2025).

42. Hamed, A. A., Zachara-Szymanska, M. & Wu, X. Safeguarding authenticity for mitigating the harms of generative AI: Issues, research agenda, and policies for detection, fact-checking, and ethical AI. *iScience* **27**, 108782 (2024).

43. Sathe, S., Li, Q., Jung, J. & Wu, J. Targeting the Mitochondria in High-Grade Gliomas. *Cancers* **17**, 3062 (2025).

44. ANTOCHI, F. Dimethyl Fumarat – a New Player in Oral Treatment Options for Relapsing Forms of Multiple Sclerosis. *Mædica* **9**, 408–409 (2014).

45. Woesle, C., Fischer-Brandies, L. & Buettner, R. A Systematic Literature Review of Hallucinations in Large Language Models. *IEEE Access* **13**, 148231–148253 (2025).

46. Ji, Z. *et al.* Survey of Hallucination in Natural Language Generation. *ACM Comput Surv* **55**, 248:1-248:38 (2023).

47. Maynez, J., Narayan, S., Bohnet, B. & McDonald, R. On Faithfulness and Factuality in Abstractive Summarization. in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (eds Jurafsky, D., Chai, J., Schluter, N. & Tetreault, J.) 1906–1919 (Association for Computational Linguistics, Online, 2020). doi:10.18653/v1/2020.acl-main.173.

48. Hsu, E. & Roberts, K. Leveraging large language models for knowledge-free weak supervision in clinical natural language processing. *Sci. Rep.* **15**, 8241 (2025).

49. Bzdok, D. *et al.* Data science opportunities of large language models for neuroscience and biomedicine. *Neuron* **112**, 698–717 (2024).

50. Norsigian, C. J. *et al.* BiGG Models 2020: multi-strain genome-scale models and expansion across the phylogenetic tree. *Nucleic Acids Res.* **48**, D402–D406 (2020).

51. Cannon, M. *et al.* DGIdb 5.0: rebuilding the drug–gene interaction database for precision medicine and drug discovery platforms. *Nucleic Acids Res.* **52**, D1227–D1235 (2024).

52. Xin, J. *et al.* High-performance web services for querying gene and variant annotation. *Genome Biol.* **17**, 91 (2016).

53. Sayers, E. W. *et al.* Database resources of the National Center for Biotechnology Information in 2025. *Nucleic Acids Res.* **53**, D20–D29 (2025).

54. Weitzel, D. *et al.* The National Research Platform: Stretched, Multi-Tenant, Scientific Kubernetes Cluster. in *Practice and Experience in Advanced Research Computing 2025: The Power of Collaboration* 1–5 (Association for Computing Machinery, New York, NY, USA, 2025). doi:10.1145/3708035.3736060.

55. Huangfu, Q. & Hall, J. A. J. Parallelizing the dual revised simplex method. *Math. Program. Comput.* **10**, 119–142 (2018).

56. Edwards, J. S., Ramakrishna, R. & Palsson, B. O. Characterizing the metabolic phenotype: A phenotype phase plane analysis. *Biotechnol. Bioeng.* **77**, 27–36 (2002).

57. Bessell, B. *et al.* COMO: a pipeline for multi-omics data integration in metabolic modeling and drug discovery. *Brief. Bioinform.* **24**, bbad387 (2023).

58. Puniya, B. L. *et al.* Integrative computational approach identifies drug targets in CD4+ T-cell-mediated immune disorders. *Npj Syst. Biol. Appl.* **7**, 1–18 (2021).

59. Clough, E. *et al.* NCBI GEO: archive for gene expression and epigenomics data sets: 23-year update. *Nucleic Acids Res.* **52**, D138–D144 (2023).

60. Wang, S. *et al.* IL-21 drives expansion and plasma cell differentiation of autoreactive CD11chiT-bet+ B cells in SLE. *Nat. Commun.* **9**, 1758 (2018).

61. Loecker, J., Bessell, B., Puniya, B. L. & Helikar, T. AutoRNAseq: Automated Bulk RNA-seq Analysis Pipeline. 2026.04.21.719844 Preprint at https://doi.org/10.64898/2026.04.21.719844 (2026).

62. Dyer, S. C. *et al.* Ensembl 2025. *Nucleic Acids Res.* **53**, D948–D957 (2025).

63. Dobin, A. *et al.* STAR: ultrafast universal RNA-seq aligner. *Bioinformatics* **29**, 15–21 (2013).

64. Patro, R., Duggal, G., Love, M. I., Irizarry, R. A. & Kingsford, C. Salmon provides fast and bias-aware quantification of transcript expression. *Nat. Methods* **14**, 417–419 (2017).

Chapter VII: Conclusion

With the ever decreasing cost of RNA-sequencing and, at single-cell resolution, becoming easier to generate and acquire, an expanded ability to analyze and interpret this data is needed. Historically, pipelines to convert raw RNA sequences into an investigatable format to query metabolism of a cell was spanned by tools that were difficult to combine. The chapters presented here aim to alleviate these difficulties, where each chapter solves a specific limitation that downstream chapters can utilize.

AutoRNAseq unifies data acquisition, reference preparation, quality control, alignment, and quantification in a single workflow. Its output is directly usable by COMO, which consumes them and merges transcriptomic and proteomic data to reconstruct metabolic models. With disease-specific information, COMO provides researchers the ability to identify drug targets specific to the dataset they are working with. This work was tested and validated by reconstructing healthy and disease-specific metabolic models of naive B cells, and it recovered targets of compounds used in clinical applications for rheumatoid arthritis and systemic lupus erythematosus. Following this, COMO was used to reconstruct 5,705 donor-cell-type-specific metabolic models for 166 donors aged 25 to 85 to investigate the relationship between age and immune metabolism. The main findings are that an age-associated signal is concentrated in T cell subtypes. The quantity of models reconstructed creates an interpretation issue, due to the sheer number of models available. This gap is addressed with MechAInistic, which is a large-language-model-based approach to analyze, interpret, and report on a question posed by a researcher.

MechAInistic was used to identify therapeutic targets in healthy and rheumatoid arthritis disease-specific models of naive B cells and healthy and multiple sclerosis disease-specific models of CD4+ Th17 cells. MechAInistic identified 2-oxoglutarate dehydrogenase as a potential target for the naive B use-case, and presented Devimistat as a potential therapeutic. For the Th17 use case, isocitrate dehydrogenase was identified as a target and Ivosidenib was proposed as a therapeutic treatment. Despite the advancements made by this work, metabolic models built from expression data provide hypotheses for enzymatic flux, rather than experimental truth. Associations presented should be taken into account to recognize these limitations, and that without experimental metabolite measurements, these remain mechanistic and literature supported interpretations.

Additional expansions exist for each chapter; the most straightforward is to interpret the 5,705 personalized metabolic models using MechAInistic, which brings its interpretive components into personalized metabolic modeling. Additionally, expanding the presented Kendall Tau statistics with Kolmogorov-Smirnov and Wasserstein metrics distribution-level tests would characterize differences in the shape of the flux samples between age groups. At the time of writing, nearly all large-language-model based systems struggle to utilize literature in a coherent manner. As a result, MechAInistic's literature reporting capabilities should be expanded by pairing HSNW-indexed embedding search, which broadens the retrieval capabilities of a given subject, with internal graph-based representations of academic literature within the index. This will allow a specific claim in one publication to not only be linked to other paragraphs within the same publication, but

also associated with the broader argument of another. Lastly, COMO can be expanded with additional multi-omic integration methods, and AutoRNAseq can be made more generic by incorporating single-cell alignment and machine-learning based algorithms for automated cell annotations.

Ultimately, the contribution of this work is best measured when asking a mechanistic, metabolic question. With the integration provided by each chapter, the limiting factor shifts from software-based issues toward the quality of the question and available data to answer it.