Metabolic pathway prediction using non-negative matrix factorization with improved precision

Abdur Rahman M. A. Basher[1], Ryan J. McLaughlin[1], and Steven J. Hallam [1,2,3,4,5*]

[1] Graduate Program in Bioinformatics, University of British Columbia, Genome Sciences Centre, 100-570 West 7th Avenue, Vancouver, British Columbia V5Z 4S6, Canada
[2] Department of Microbiology & Immunology, University of British Columbia, 2552-2350 Health Sciences Mall, Vancouver, British Columbia V6T 1Z3, Canada.
[3] Genome Science and Technology Program, University of British Columbia, 2329 West Mall, Vancouver, BC V6T 1Z4, Canada
[4] Life Sciences Institute, University of British Columbia, Vancouver, British Columbia, Canada V6T 1Z3
[5] ECOSCOPE Training Program, University of British Columbia, Vancouver, British Columbia, Canada V6T 1Z3

Abstract

Machine learning provides a probabilistic framework for metabolic pathway inference from genomic sequence information at different levels of complexity and completion. However, several challenges including pathway features engineering, multiple mapping of enzymatic reactions and emergent or distributed metabolism within populations or communities of cells can limit prediction performance. In this paper, we present triUMPF, triple non-negative matrix factorization (NMF) with community detection for metabolic pathway inference, that combines three stages of NMF to capture myriad relationships between enzymes and pathways within a graph network. This is followed by community detection to extract higher order structure based on the clustering of vertices which share similar statistical properties. We evaluated triUMPF performance using experimental datasets manifesting diverse multi-label properties, including Tier 1 genomes from the BioCyc collection of organismal Pathway/Genome Databases and low complexity microbial communities. Resulting performance metrics equaled or exceeded other prediction methods on organismal genomes with improved precision on multi-organismal datasets.

Availability and implementation: The software package, and installation instructions are published on github.com/triUMPF

Contact: shallam@mail.ubc.ca

Keywords: non-negative matrix factorization, community detection, metabolic pathway prediction, BioCyc, multi-organismal genomes.

1 Introduction

Pathway reconstruction from genomic sequence information is an essential step in describing the metabolic potential of cells at the individual, population and community levels of biological organization [12, 18, 25]. Resulting pathway representations provide a foundation for defining regulatory processes, modeling metabolite flux and engineering cells and cellular consortia for defined process outcomes [11, 20]. The integral nature of the pathway prediction problem has prompted both gene-centric e.g. mapping annotated proteins onto known pathways using a reference database based on sequence homology, and heuristic or rule-based pathway-centric approaches including PathoLogic [16] and MinPath [38]. In parallel, the development of trusted sources of curated metabolic pathway information including the Kyoto Encyclopedia of Genes and Genomes (KEGG) [15] and MetaCyc [4] provides training data for the design of more flexible machine learning (ML) algorithms for pathway inference. While ML approaches have been adopted widely in metabolomics research [3,34] they have gained less traction when applied to predicting pathways directly from annotated gene lists.

Dale and colleagues conducted the first in-depth exploration of ML approaches for pathway prediction using Tier 1 (T1) organismal Pathway/Genome Databases (PGDB) [5] from the BioCyc collection randomly divided into training and test sets [7]. Features were developed based on rule-sets used by the PathoLogic algorithm in Pathway Tools to construct PGDBs [16]. Resulting performance metrics indicated that standard ML approaches rivaled PathoLogic performance with the added benefit of probability scores [7]. More recently Basher and colleagues developed multi-label based on logistic regression for pathway prediction (mlLGPR), a multi-label classification approach that uses logistic regression and feature vectors inspired by the work of Dale and colleagues to predict metabolic pathways from genomic sequence information at different levels of complexity and completion [25].

Although mlLGPR performed effectively on organismal genomes, pathway prediction outcomes for multi-organismal datasets were less optimal due in part to missing or noisy feature information. In an effort to solve this problem, Basher and Hallam evaluated the use of representational learning methods to learn a neural embedding-based low-dimensional space of metabolic features based on a three-layered network architecture consisting of compounds, enzymes, and pathways [24]. Learned feature vectors improved pathway prediction performance on organismal genomes and motivated the use of graphical models for multi-organismal features engineering.

Here we describe triple non-negative matrix factorization (NMF) with community detection for metabolic pathway inference (triUMPF) combining three stages of NMF to capture relationships between enzymes and pathways within a network [9] followed by community detection to extract higher order network structure [8]. Non-negative matrix factorization is a data reduction and exploration method in which the original and factorized matrices have the property of non-negative elements with reduced ranks or features [9]. In contrast to other dimension reduction methods, such as principal component analysis [2], NMF both reduces the number of features and preserves information needed to reconstruct the original data [37]. This has important implications for noise robust feature extraction from sparse matrices including datasets associated with gene expression analysis and pathway prediction [37].

For pathway prediction, triUMPF uses three graphs, one representing associations between pathways and enzymes indicated by enzyme commission (EC)) numbers [1], one representing interactions between enzymes and another representing interactions between pathways. The two interaction graphs adopt the *subnetworks* concept introduced in BiomeNet [32] and MetaNetSim [14], where a subnetwork is a linked series of connected nodes (e.g. reactions and pathways). In the literature, a subnetwork is commonly referred to as a *community* [30], which defines a set of densely connected nodes within a subnetwork. It is important to emphasize that unless otherwise indicated, the use of the term community in this work refers to a subnetwork community based on statistical properties of a network rather than a community of organisms. Community detection is performed on both interaction graphs (pathways and enzymes) to identify subnetworks among pathways.

We evaluated triUMPF's prediction performance in relation to other methods including MinPath, PathoLogic, and mlLGPR on a set of T1 PGDBs, low complexity microbial communities including symbiont genomes encoding distributed metabolic pathways for amino acid biosynthesis [26], genomes used in the Critical Assessment of Metagenome Interpretation (CAMI) initiative [31], and whole genome shotgun sequences from the Hawaii Ocean Time Series (HOTS) [33] following information hierarchy-based benchmarks initially developed for mlLGPR enabling more robust comparison between pathway prediction methods [25].

2 Methods

In this section, we provide a general description of triUMPF components, presented in Fig. 1. At the very beginning, MetaCyc is applied to: i)- extract three association matrices, indicated in step Fig. 1(a), one representing associations between pathways and enzymes (P2E) indicated by enzyme commission (EC)) numbers [27], one representing interactions between enzymes (E2E) and another representing interactions between pathways (P2P), and ii)- automatically generate features corresponding pathways and enzymes (or EC) from pathway2vec [24] in Fig. 1(b). Then, triUMPF is trained in two phases: i)- decomposition of the pathway EC association matrix in Fig. 1(c), and ii)- subnetwork or community reconstruction while, simultaneously, learning optimal multi-label pathway parameters in Figs 1(d-f). Below, we discuss these two phases while the analytical expressions of triUMPF are explained in Appx. Sections 5.1, 5.2, and 5.3.

2.1 Decomposing the Pathway EC Association Matrix

Inspired by the idea of non-negative matrix factorization (NMF), we decompose the P2E association matrix to recover low-dimensional latent factor matrices [9]. Unlike previous application of NMF to biological data [28], triUMPF incorporates constraints into the matrix decomposition process. Formally, let $\mathbf{M} \in \mathbb{Z}_{\geq 0}^{t \times r}$ be a non-negative matrix, where t is the number of pathways and r is the number of enzymatic reactions. Each row in $\mathbf{M}$ corresponds to a pathway and each column represent an EC, such that $\mathbf{M}_{i,j} = 1$ if an EC j is in pathway i and 0 otherwise. Given $\mathbf{M}$, the standard NMF decomposes this matrix into the two low-rank matrices, i.e. $\mathbf{M} \approx \mathbf{W}\mathbf{H}^\top$, where $\mathbf{W} \in \mathbb{R}^{t \times k}$ stores the latent factors for pathways while $\mathbf{H} \in \mathbb{R}^{r \times k}$ is latent factors associated with ECs and $k (\in \mathbb{Z}_{\geq 1}) \ll t, r$. However, triUMPF extends this standard NMF by leveraging features, obtained from *pathway2vec* [24], encoding two interactions: i)- within ECs or pathways and ii)- between pathways and ECs. For more details about this step, please see Appx. Section 5.2.1.

2.2 Community Reconstruction and Multi-label Learning

The community detection problem [23,30] is the task of discovering distinct groups of nodes that are densely connected. During this phase, triUMPF performs community detection to guide the learning process for pathways using binary P2P ($\mathbf{A} \in \mathbb{Z}_{\geq 0}^{t \times t}$) and E2E ($\mathbf{B} \in \mathbb{Z}_{\geq 0}^{r \times r}$) association matrices, where each entry in these matrices is a binary value indicating an interaction among corresponding entities. However, $\mathbf{A}$ and $\mathbf{B}$ capture pairwise first-order proximity among their related entities, consequently, they are inadequate to fully characterize distant relationships among pathways or ECs [30]. Therefore, triUMPF utilizes higher-order proximity using the following formula [23]:

$$\mathbf{A}^{\mathbf{prox}} = \sum_{i \in l_p} \omega_i \mathbf{A}^l, \qquad \mathbf{B}^{\mathbf{prox}} = \sum_{i \in l_e} \gamma_i \mathbf{B}^l \tag{1}$$

where $\mathbf{A}^{\mathbf{prox}}$ and $\mathbf{B}^{\mathbf{prox}}$ are polynomials of order $l_p \in \mathbf{Z}_{>0}$ and $l_e \in \mathbf{Z}_{>0}$, respectively, and $\omega \in \mathbf{R}_{>0}$ and $\gamma \in \mathbf{R}_{>0}$ are weights associated to each term. Using these higher order matrices, triUMPF applies two NMFs to recover communities (Appx. Section 5.2.2). Then, triUMPF uses $\mathbf{W}$ and $\mathbf{H}$ from the decomposition phases (Section 2.1) and the detected communities to optimize multi-label pathway parameters in an iterative process (Appx. Section 5.2.3) until the maximum number of allowed iterations is reached. At the end, the trained model can be used to perform pathway prediction from an organismal genome or multi-organismal dataset with high precision due to constraints embedded in the P2E, P2P, and E2E associations matrices.

3 Results

We evaluated triUMPF performance across multiple datasets spanning the genomic information hierarchy [25]: i)- T1 golden consisting of EcoCyc, HumanCyc, AraCyc, YeastCyc, LeishCyc, and TrypanoCyc; ii)- three *E. coli* genomes composed of E. coli K-12 substr. MG1655 (TAX-511145), uropathogenic E. coli str. CFT073 (TAX-199310), and enterohemorrhagic E. coli O157:H7 str. EDL933 (TAX-155864); iii)- BioCyc (v20.5 T2 & 3) [5] composed of 9255 PGDBs (Pathway/Genome Databases) with 1463 pathways constructed using Pathway Tools v21 [16];

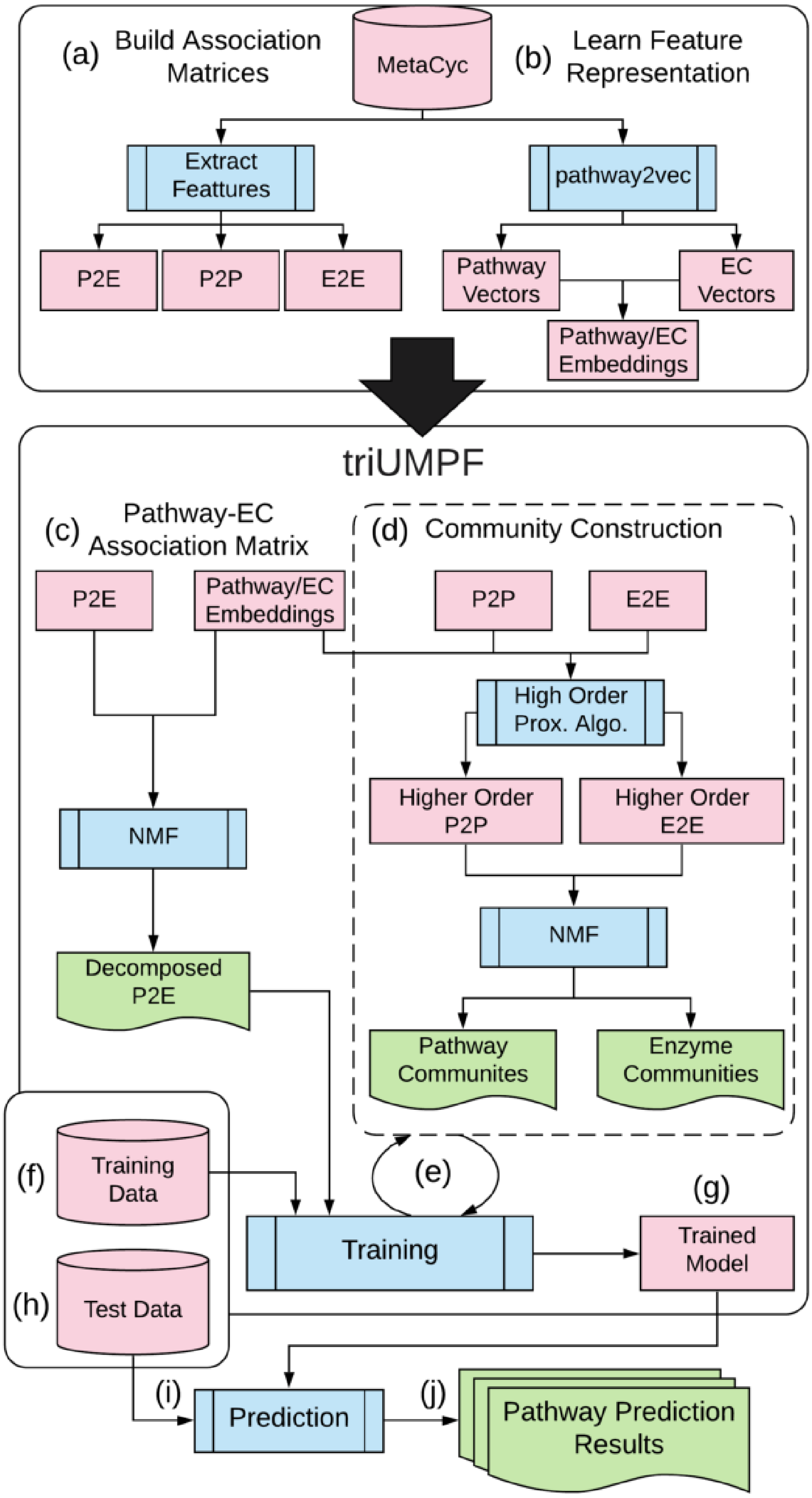


Figure 1: A workflow diagram showing the proposed triUMPF method. Initially, triUMPF takes the Pathway-EC association (P2E) information (a) to produce several low rank matrices (c) while, simultaneously, detecting pathway and EC communities (d) given two interaction matrices, corresponding Pathway-Pathway (P2P) and EC-EC (E2E) (a). For both steps (c) and (d), pathway and EC features obtained from pathway2vec package (b) are utilized. Afterwards, triUMPF iterates between updating community parameters (d) and optimizing multi-label parameters (e) with the use of training data (f). Once the training is achieved the learned model (g) can be used to predict a set of pathways (i-j) from an organismal genome or multi-organismal dataset (h).

Table 1: Average precision of each comparing algorithm on 6 golden T1 data.

Methods	Average Precision Score					
	EcoCyc	HumanCyc	AraCyc	YeastCyc	LeishCyc	TrypanoCyc
PathoLogic	0.7230	**0.6695**	0.7011	0.7194	**0.4803**	**0.5480**
MinPath	0.3490	0.3004	0.3806	0.2675	0.1758	0.2129
mlLGPR	0.6187	0.6686	0.7372	0.6480	0.4731	0.5455
triUMPF	**0.8662**	0.6080	**0.7377**	**0.7273**	0.4161	0.4561

iv)- *symbionts* genomes of *Moranella* (GenBank NC-015735) and *Tremblaya* (GenBank NC-015736) encoding distributed metabolic pathways for amino acid biosynthesis [26]; v)- Critical Assessment of Metagenome Interpretation (CAMI) initiative low complexity dataset consisting of 40 genomes [31]; and vi)- whole genome shotgun sequences from the Hawaii Ocean Time Series (HOTS) at 25m, 75m, 110m (sunlit) and 500m (dark) ocean depth intervals [33]. We applied BioCyc v20.5 to train triUMPF while the remaining datasets were used to report performance results. Since BioCyc v20.5 contains less than 1460 trainable pathways, we applied pathway2vec with RUST-norm (or "crt") configuration to improve prediction (see Section 5.4.3). For general statistics about these datasets are summarized in Appx. Table 4.

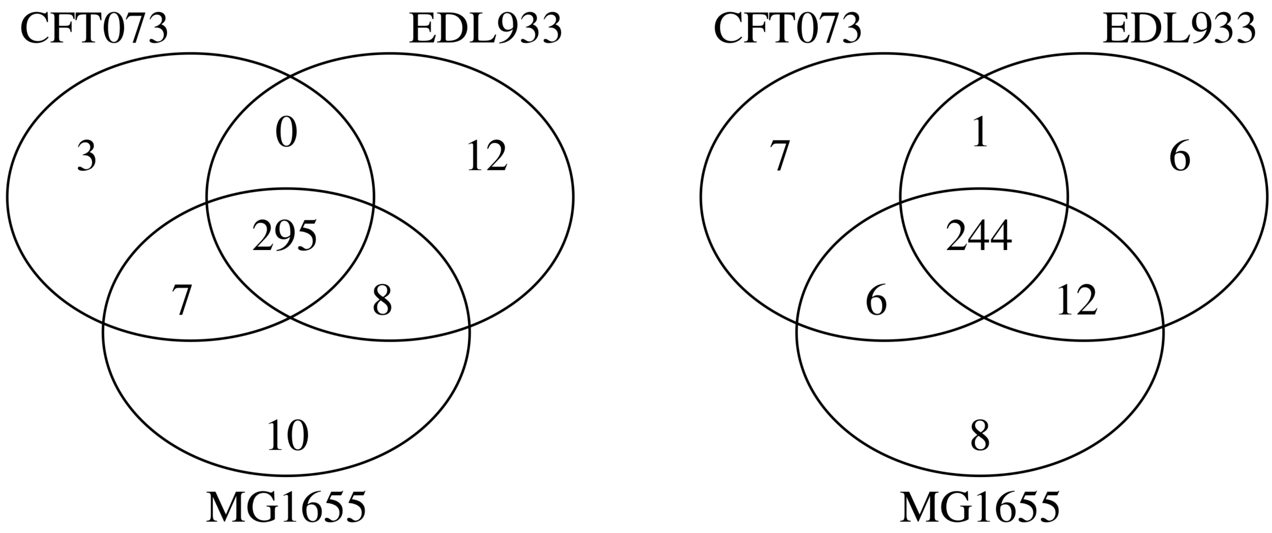
For comparative analysis, triUMPF's performance on T1 golden datasets was compared to three pathway prediction methods: i)- MinPath version 1.2 [38], which uses integer programming to recover a conserved set of pathways from a list of enzymatic reactions; ii)- PathoLogic version 21 [16], which is a symbolic approach that uses a set of manually curated rules to predict pathways; and iii)- mlLGPR which uses supervised multi-label classification and rich feature information to predict pathways from a list of enzymatic reactions [25]. In addition to testing on T1 golden datasets, triUMPF performance was compared to PathoLogic on three *E. coli* genomes and to PathoLogic and mlLGPR on mealybug symbionts, CAMI low complexity, and HOTS multi-organismal datasets. The following metrics were used to report on performance of pathway prediction algorithms including: *average precision*, *average recall*, *average F1 score (F1)*, and *Hamming loss* as described in [25]. For experimental settings and additional tests, see Appx. Sections 5.4 and 5.5.

3.1 T1 Golden Data

As shown in Table 1, triUMPF achieved competitive performance against the other methods in terms of average precision with optimal performance on EcoCyc (0.8662). However, with respect to average F1 scores, it under-performed on HumanCyc and AraCyc, yielding average F1 scores of 0.4703 and 0.4775, respectively (Appx. Table 5). Since the observed number of pathway labels in BioCyc v20.5 is 1463 pathways (a subset of 2526 MetaCyc pathways) (as explained in Section 3), triUMPF trained with this data (using features from pathway2vec [24]) can not infer pathways outside the trainable pathways. Consequently, this has translated into low average F1 scores of HumanCyc and AraCyc. A possible treatment would be incorporating additional PGDBs containing more pathways to train triUMPF. However, this would require substantially building many PGDBs from organismal genomes or using multiple versions of BioCyc data. A detailed analysis on this is left for future work.

3.2 Three E. coli data

Recall that community detection (Section 2.2) was used to guide the multi-label learning process. To demonstrate the influence of communities on pathway prediction, we compared pathways predicted for the T1 gold standard E. coli K-12 substr. MG1655 (TAX-511145), henceforth referred to as MG1655, using PathoLogic and triUMPF. Appx. Fig. 8a shows the results, where both methods inferred 202 true-positive pathways (green-colored) in common out of 307 expected true-positive pathways (using EcoCyc as a common frame of reference). In addition, PathoLogic uniquely predicted 39 (magenta-colored) true-positive pathways while triUMPF uniquely predicted 16 true-positives (purple-colored). This difference arises from the use of tax-



(a) PathoLogic (taxonomic pruning)

(b) triUMPF

Figure 2: A three way set difference analysis of pathways predicted for E. coli K-12 substr. MG1655 (TAX-511145), E. coli str. CFT073 (TAX-199310), and E. coli O157:H7 str. EDL933 (TAX-155864) using (a) PathoLogic (taxonomic pruning) and (b) triUMPF.

onomic pruning in PathoLogic which improves recovery of taxonomically constrained pathways and limits false-positive identification. With taxonomic pruning enabled, PathoLogic inferred 79 false-positive pathways, and over 170 when pruning was disabled. In contrast triUMPF which does not use taxonomic feature information inferred 84 false-positive pathways (Appx. Table 6). This improvement over PathoLogic with pruning disabled reinforces the idea that pathway communities improve precision of pathway prediction with limited impact on overall recall. Based on these results, it is conceivable to train triUMPF on subsets of organismal genomes resulting in more constrained pathway communities for pangenome analysis.

To further evaluate triUMPF performance on closely related organismal genomes, we performed pathway prediction on E. coli str. CFT073 (TAX-199310), and E. coli O157:H7 str. EDL933 (TAX-155864) and compared results to the MG1655 reference strain [36]. Both CFT073 and EDL933 are pathogens infecting the human urinary and gastrointestinal tracts, respectively. Previously, Welch and colleagues described extensive genomic mosaicism between these strains and MG1655, defining a core backbone of conserved metabolic genes interspersed with genomic islands encoding common pathogenic or niche defining traits [36]. Neither CFT073 nor EDL933 genomes are represented in the BioCyc collection of organismal pathway genome databases. A total of 335 and 319 unique pathways were predicted by PathoLogic and triUMPF, respectively. The resulting pathway lists were used to perform a set-difference analysis with MG1655 (Fig. 2). Both methods predicted more than 200 pathways encoded by all three strains including core pathways like the *TCA* cycle (Appx. Figs 8b and 8c). CFT073 and EDL933 were predicted to share a single common pathway (*TCA cycle IV (2-oxoglutarate decarboxylase)*) by triUMPF. However this pathway variant has not been previously identified in E. coli and is likely a false-positive prediction based on recognized taxonomic range. Both PathoLogic and triUMPF predicted the *aerobactin biosynthesis* pathway involved in siderophore production in CFT073 consistent with previous observations [36]. Similarly, four pathways (e.g. *L-isoleucine biosynthesis III* and *GDP-D-perosamine biosynthesis*) unique to EDL933 were inferred by both methods.

Given the lack of cross validation standards for CFT073 and EDL933 we were unable to determine which method inferred fewer false-positives across the complete set of predicted pathways. To constrain this problem on a subset of the data, we applied GapMind [29] to analyze amino acid biosynthesis pathways encoded in MG1655, CFT073 and EDL933 genomes. GapMind is a web-based application developed for annotating amino acid biosynthesis pathways in prokaryotic microorganisms (bacteria and archaea), where each reconstructed pathway is supported

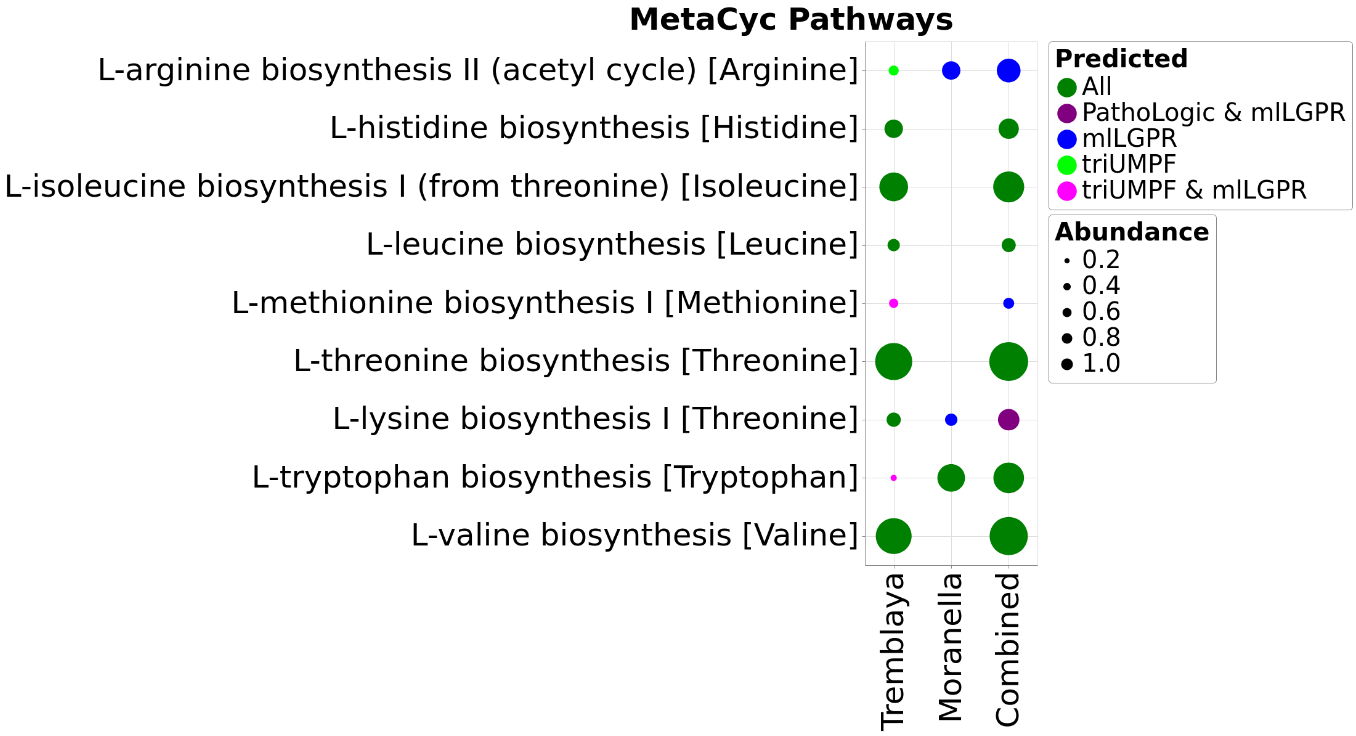


Figure 3: Comparative study of predicted pathways for symbiotic data between PathoLogic, mlLGPR, and triUMPF. The size of circles corresponds the associated coverage information.

by a confidence level. After excluding pathways that were not incorporated in the training set, a total of 102 pathways were identified across the three strains encompassing 18 amino acid biosynthesis pathways and 27 pathway variants with high confidence (Appx. Table 7). PathoLogic inferred 49 pathways identified across the three strains encompassing 15 amino acid biosynthesis pathways and 17 pathway variants while triUMPF inferred 54 pathways identified across the three strains encompassing 16 amino acid biosynthesis pathways and 19 pathway variants including *L-methionine biosynthesis* in MG1655, CFT073 and EDL933 that was not predicted by PathoLogic. Neither method was able to predict *L-tyrosine biosynthesis I* (Appx. Fig. 10).

3.3 Mealybug Symbionts data

To evaluate triUMPF performance on distributed metabolic pathways, we used the reduced genomes of *Moranella* and *Tremblaya* [26]. Collectively the two symbiont genomes encode intact biosynthesis pathways for 9 essential amino acids. PathoLogic, mlLGPR, and triUMPF were used to predict pathways on individual symbiont genomes and a composite genome consisting of both, and resulting amino acid biosynthesis pathway distributions were determined (Fig. 3). Both triUMPF and PathoLogic predicted 6 of the expected amino acid biosynthesis pathways on the composite genome while mlLGPR predicted 8 pathways. The pathway for phenylalanine biosynthesis (*L-phenylalanine biosynthesis I*) was excluded from analysis because the associated genes were reported to be missing during the ORF prediction process. False positives were predicted for individual symbiont genomes in *Moranella* and *Tremblaya* using both methods although pathway coverage was reduced in relation to the composite genome.

3.4 CAMI and HOTS data

To evaluate triUMPF's performance on more complex multi-organismal genomes, we used the CAMI low complexity [31] and HOTS datasets [33] comparing resulting pathway predictions to both PathoLogic and mlLGPR. For CAMI low complexity, triUMPF achieved an average F1 score of 0.5864 in comparison to 0.4866 for mlLGPR which is trained with more than 2500 labeled pathways (Table 2). Similar results were obtained for HOTS (see Appx. Section 5.5.4).

Table 2: Predictive performance of mlLGPR and triUMPF on CAMI low complexity data. For each performance metric, '↓' indicates the smaller score is better while '↑' indicates the higher score is better.

Metric	mlLGPR	triUMPF
Hamming Loss (↓)	0.0975	**0.0436**
Average Precision Score (↑)	0.3570	**0.7027**
Average Recall Score (↑)	**0.7827**	0.5101
Average F1 Score (↑)	0.4866	**0.5864**

Among a subset of 180 selected water column pathways, PathoLogic and triUMPF predicted a total of 54 and 58 pathways, respectively, while mlLGPR inferred 62. From a real world perspective none of the methods predicted pathways for *photosynthesis light reaction* nor *pyruvate fermentation to (S)-acetoin* although both are expected to be prevalent in the water column. Perhaps, the absence of specific ECs associated with these pathway limits rule-based or ML prediction. Indeed, closer inspection revealed that the enzyme *catabolic acetolactate synthase* was missing from the *pyruvate fermentation to (S)-acetoin* pathway, which is an essential rule encoded in PathoLogic and represented as a feature in mlLGPR. Conversely, although this pathway was indexed to a community, triUMPF did not predict its presence, constituting a false-negative.

4 Discussion and Conclusion

In this paper we introduced a novel ML approach for metabolic pathway inference that combines three stages of NMF to capture relationships between enzymes and pathways within a network followed by community detection to extract higher order network structure. First, a Pathway-EC association ($\mathbf{M}$) matrix, obtained from MetaCyc, is decomposed using the NMF technique to learn a constrained form of the pathway and EC factors, capturing the microscopic structure of $\mathbf{M}$. Then, we obtain the community structure (or mesoscopic structure) jointly from both the input datasets and two interaction matrices, Pathway-Pathway interaction and EC-EC interaction. Finally, the consensus relationships between the community structure and data, and between the learned factors from $\mathbf{M}$ and the pathway labels coefficients are exploited to efficiently optimize metabolic pathway parameters.

We evaluated triUMPF performance using a corpora of experimental datasets manifesting diverse multi-label properties comparing pathway prediction outcomes to other prediction methods including PathoLogic [16] and mlLGPR [25]. During benchmarking we realized that the BioCyc collection suffers from a class imbalance problem [13] where some pathways infrequently occur across PGDBs. This results in a significant sensitivity loss on T1 golden data, where triUMPF tended to predict more frequently observed pathways while missing more infrequent pathways. One potential approach to solve this class-imbalance problem is subsampling the most informative PGDBs for training, hence, reducing false-positives [19]. Despite the observed class imbalance problem, triUMPF improved pathway prediction precision without the need for taxonomic rules or EC features to constrain metabolic potential. From an ML perspective this is a promising outcome considering that triUMPF was trained on a reduced number of pathways relative to mlLGPR. Future development efforts will explore subsampling approaches to improve sensitivity and the use of constrained taxonomic groups for pangenome and multi-organismal genome pathway inference.

5 Appendices

5.1 Appendix A1: Definitions and Problem Formulation

Here, the default vector is considered to be a column vector and is represented by a boldface lowercase letter (e.g., $\mathbf{x}$) while matrices are represented by boldface uppercase letters (e.g., $\mathbf{X}$).

The $\mathbf{X}_i$ matrix indicates the i-th row of $\mathbf{X}$ and $\mathbf{X}_{i,j}$ denotes the (i, j)-th entry of $\mathbf{X}$ while, for a vector, $\mathbf{x}_i$ denotes an i-th cell of $\mathbf{x}$. The transpose of $\mathbf{X}$ is denoted as $\mathbf{X}^\top$ and the trace of it is symbolized as $\mathbf{tr}(\mathbf{X})$. The Frobenius norm of $\mathbf{X}$ is defined as $||\mathbf{X}||_F = \sqrt{\sum_{i\in n}\sum_{j\in m}\mathbf{X}_{i,j}^2}$. Occasional superscript, $\mathbf{x}^{(i)}$, suggests an index to a sample, a power, or a current epoch during a learning period. We use calligraphic letters to represent sets (e.g., $\mathcal{E}$) while we use the notation $|.|$ to denote the cardinality of a given set. With these notations in mind, we introduce several concepts integral to the problem formulation.

Metabolic pathway inference from genomic sequence information at different levels of complexity and completion requires a trusted source of labeled pathway information in which the set of ordered reactions within and between cells is linked to substrates and products (compounds or metabolites). This information can be represented in graphs corresponding to reactome and pathway-level interactions. In this study, we use MetaCyc, a multi-organism member of the BioCyc collection of Pathway/Genome Databases (PGDB) as the trusted source for reactome and pathway information [5]. MetaCyc contains only experimentally validated metabolic pathways across all domains of life. To simplify computational complexity, we consider the reaction and pathway graphs to be undirected.

Definition 5.1. Reaction Graph Topology. Let the reaction graph be represented by an undirected graph $\mathcal{G}^{(\mathbf{rxn})} = \{\mathcal{C}, \mathcal{Z}^{(c)}\}$, where $\mathcal{C}$ is a set of c metabolites and $\mathcal{Z}^{(c)}$ represents r' links between compounds. Each link indicates a reaction, derived from a set of biochemical reactions $\mathcal{R}$ of size r'. Then, the reaction graph topology is defined by a matrix $\Omega^{(c)} \in \mathbb{Z}_{\geq 0}^{r'\times c}$, where each entry $\Omega_{i,j}^{(c)}$ is a binary value of 1 or 0, indicating either the compound j is a substrate/product in a reaction i or not involved in that reaction, respectively. ■

Definition 5.2. Pathway Graph Topology. Let $\mathcal{G}^{(\mathbf{path})} = \{\mathcal{R}, \mathcal{Z}'^{(r)}\}$ be an undirected graph, where $\mathcal{R}$ is presented in Def. 5.1, and $\mathcal{Z}'^{(r)}$ represents a set of t' links between reactions. Then, the pathway graph topology is defined by a matrix $\Omega^{(r)} \in \mathbb{Z}_{\geq 0}^{t\times r'}$, where each entry $\Omega_{i,j}^{(r)}$ is either 0 or a positive integer, corresponding the absence or the frequency of the reaction j in pathway i, respectively. And, t is the number of pathways in a set $\mathcal{T}$. ■

Note that reactions in $\mathcal{G}^{(\mathbf{path})}$ may be annotated as a *spontaneous reaction* or a reaction catalyzed by one or more enzymes, *enzymatic reaction* and classified by an *enzyme commission* number (EC) [27]. In addition, a number of enzymes referred to as *promiscuous enzymes* can participate in more than one pathway. Given this information we associate EC numbers to pathways and formulate three graphs, one representing associations between pathways and enzymes indicated by enzyme commission (EC)) numbers, one representing interactions between enzymes and another representing interactions between pathways.

Definition 5.3. Pathway-EC Association (P2E). Let $\mathcal{G}^{\prime,(\mathbf{path})} = \{\mathcal{E}, \mathcal{Z}^{(r)}\}$ be a subgraph of $\mathcal{G}^{(\mathbf{path})}$, such that $\mathcal{E} \subset \mathcal{R}$ with $r \ll r'$ enzymatic reactions. Then, the Pathway-EC association is defined as a matrix $\mathbf{M} \in \mathbb{Z}_{\geq 0}^{t\times r}$, where each row corresponds to a pathway, and each column represent an EC, such that $\mathbf{M}_{i,j} = 1$ if an EC j is in pathway i and 0 otherwise. ■

Typically, the association matrix $\mathbf{M}$ is extremely sparse. Using reaction and pathway graph topology, we build interaction adjacency matrices as follows.

Definition 5.4. EC-EC Interaction (E2E). Given $\mathcal{G}'^{(\mathbf{rxn})} \subset \mathcal{G}^{(\mathbf{rxn})}$, we define an EC-EC interaction matrix $\mathbf{B} \in \mathbb{Z}_{\geq 0}^{r\times r}$ such that an entry $\mathbf{B}_{i,j}$ is a binary value encoding an interaction between two ECs i and j iff they both share a compound, i.e., $\Omega_{i,k}^{(c)} \wedge \Omega_{j,k}^{(c)} = 1$ where $k \in \mathcal{C}$. ■

Definition 5.5. Pathway-Pathway Interaction (P2P). Given $\mathcal{G}^{(\mathbf{path})}$, we define a Pathway-Pathway interaction matrix $\mathbf{A} \in \mathbb{Z}_{\geq 0}^{t\times t}$ such that an entry $\mathbf{A}_{i,j}$ is a binary value indicating an interaction between pathways i and j if there exists a reaction $k \in \mathcal{R}$ where associated compounds are either substrate or product in both i and j pathways. ■

After determining relationships within each graph, we define a *multi-label* metabolic pathway dataset.

Definition 5.6. Multi-label Pathway Dataset [25]. A general form of pathway dataset is characterized by $\mathcal{S} = \{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}) : 1 < i \leqslant n\}$ consisting of n examples, where $\mathbf{x}^{(i)}$ is a vector indicating the abundance information corresponding to each enzymatic reaction. An enzymatic reaction, in turn, is denoted by e, which is an element of a set of enzymatic reactions $\mathcal{E} = \{e_1, e_2, ..., e_r\}$, having r possible reactions. The abundance of an enzymatic reaction i, for example $e_l^{(i)}$, is defined as $a_l^{(i)} (\in \mathbb{R}_{\geq 0})$. The class labels $\mathbf{y}^{(i)} = [y_1^{(i)}, ..., y_t^{(i)}] \in \{-1, +1\}^t$ is a pathway label vector of size t that represents the total number of pathways, which are derived from a set of labeled metabolic pathway $\mathcal{Y}$. The matrix form of $\mathbf{x}^{(i)}$ and $\mathbf{y}^{(i)}$ are symbolized as $\mathbf{X}$ and $\mathbf{Y}$, respectively. ■

The input space is assumed to be encoded as r-dimensional feature vector and is symbolized as $\mathcal{X} = \mathbb{R}^r$. Furthermore, each example in $\mathcal{S}$ is considered to be drawn independent, identically distributed (i.i.d) from an unknown distribution $\mathcal{D}$ over $\mathcal{X} \times 2^{|\mathcal{Y}|}$. Now we state the problem considered in this paper.

Metabolic Pathway Prediction. Given: i)- Pathway-EC matrix $\mathbf{M}$, ii)- a Pathway-Pathway interaction matrix $\mathbf{A}$, iii)- an EC-EC interaction matrix $\mathbf{B}$, and iv)- a dataset $\mathcal{S}$, the goal is to efficiently reconstruct pathway labels for a hitherto unseen instance $\mathbf{x}^*$.

5.2 Appendix A2: Detailed Description of triUMPF Method

In this section, we provide a description of triUMPF components, presented in Fig. 1 of main manuscript, including: i)- decomposing the pathway EC association matrix , ii)- subnetwork or community reconstruction, and iii)- the multi-label learning process.

5.2.1 Decomposing the Pathway EC Association Matrix

Given the non-negative $\mathbf{M}$, we formulate the following minimization objective function:

$$\begin{aligned} \mathcal{J}^{\textbf{fact}}(\mathbf{W}, \mathbf{H}, \mathbf{U}, \mathbf{V}) = \min_{\mathbf{W},\mathbf{H},\mathbf{U},\mathbf{V}} & ||\mathbf{M} - \mathbf{W}\mathbf{H}^\top||_F^2 + \lambda_1 ||\mathbf{W} - \mathbf{P}\mathbf{U}||_F^2 \\ & + \lambda_2 ||\mathbf{H} - \mathbf{E}\mathbf{V}||_F^2 + \lambda_3 ||\mathbf{U} - \mathbf{V}||_F^2 \\ & + \lambda_4 (||\mathbf{W}||_F^2 + ||\mathbf{H}||_F^2 + ||\mathbf{U}||_F^2 + ||\mathbf{V}||_F^2) \\ \text{s.t. } \{\mathbf{W}, \mathbf{H}, \mathbf{U}, \mathbf{V}\} \geq 0 & \end{aligned} \tag{2}$$

where $\mathbf{W} \in \mathbb{R}^{t \times k}$ stores the latent factors for pathways while $\mathbf{H} \in \mathbb{R}^{r \times k}$, known as the basis matrix, can be thought of as latent factors associated with ECs and $k \ll t, r$ and λ_* are regularization hyperparameters. The leftmost term is the well-known squared loss function that penalizes the deviation of the estimated entries in both $\mathbf{W}$ and $\mathbf{H}$ from the true association matrix $\mathbf{M}$. The second term corresponds to the relative differences of latent matrix $\mathbf{W}$ from the pathway features $\mathbf{P} \in \mathbb{R}^{t \times m}$, learned using pathway2vec framework, where the matrix $\mathbf{U} \in \mathbb{R}^{m \times k}$ absorbs different scales of matrices $\mathbf{W}$ and $\mathbf{P}$. Similarly, the third term indicates the squared loss of $\mathbf{H}$ from $\mathbf{E} \in \mathbb{R}^{r \times m}$, which denotes the feature matrix of ECs, and their differences are captured by $\mathbf{V} \in \mathbb{R}^{m \times k}$. In the fourth term, we minimize the differences between factors $\mathbf{U}$ and $\mathbf{V}$, capturing shared prominent features for the low dimensional coefficients.

5.2.2 Subnetwork or Community Reconstruction

Recall from the main manuscript, the higher order proximity of the two matrices $\mathbf{A}$ and $\mathbf{B}$ is defined according to the formula [23]:

$$\mathbf{A}^{\textbf{prox}} = \sum_{i \in l_p} \omega_i \mathbf{A}^l, \qquad \mathbf{B}^{\textbf{prox}} = \sum_{i \in l_e} \gamma_i \mathbf{B}^l \tag{3}$$

where $\mathbf{A}^{\textbf{prox}}$ and $\mathbf{B}^{\textbf{prox}}$ are polynomials of order $l_p \in \mathbf{Z}_{>0}$ and $l_e \in \mathbf{Z}_{>0}$, respectively, and $\omega \in \mathbf{R}_{>0}$ and $\gamma \in \mathbf{R}_{>0}$ are weights associated to each term. Using these higher order matrices, we invoke NMF to recover communities.

Formally, let $\mathbf{T} \in \mathbb{R}^{m \times p}$ be a non-negative community representation matrix of size p communities for pathways, where the j-th column in $\mathbf{T}_{:,j}$ denotes the representation of community j.

The pathway community indicator matrix is denoted by $\mathbf{C} \in \mathbb{R}^{t \times p}$ conditioned on $\mathbf{tr}(\mathbf{C}^\top \mathbf{C}) = t$, where each entry $\mathbf{C}_{i,l}$ and $\mathbf{C}_{j,l}$ encodes the probability that pathways i and j generates an edge belonging to a community l. The probability of i and j belonging to the same community can be assessed as: $\widehat{\mathbf{A}^{\mathbf{prox}}_{i,j}} = (\mathbf{P}_i \mathbf{C}_{:,l} \mathbf{T}^\top_{l,i})^\top (\mathbf{P}_j \mathbf{C}_{:,l} \mathbf{T}^\top_{l,j})$. Similar discussion follows for the non-negative representation matrix $\mathbf{R} \in \mathbb{R}^{m \times v}$ and the EC community indicator matrix $\mathbf{K} \in \mathbb{R}^{r \times v}$ of v communities, conditioned on $\mathbf{tr}(\mathbf{K}^\top \mathbf{K}) = r$. Unfortunately, due to the constraints emphasized on $\mathbf{C}$ and $\mathbf{K}$, it is not straightforward to analytically derive an expression, instead, we resort to a more tractable solution provided in [35], and relax the condition to be an orthogonal constraint, resulting in the following objective function:

$$\begin{aligned}\mathcal{J}^{\mathbf{comm}}(\mathbf{C}, \mathbf{K}) = &\min_{\mathbf{C},\mathbf{K}} ||\mathbf{A}^{\mathbf{prox}} - \mathbf{P}\mathbf{T}\mathbf{C}^\top||_F^2 \\ &+ ||\mathbf{B}^{\mathbf{prox}} - \mathbf{E}\mathbf{R}\mathbf{K}^\top||_F^2 \\ &+ \alpha||\mathbf{C}^\top\mathbf{C} - \mathbf{I}||_F^2 + \beta||\mathbf{K}^\top\mathbf{K} - \mathbf{I}||_F^2 \\ &+ \lambda_5(||\mathbf{C}||_F^2 + ||\mathbf{K}||_F^2) \\ \text{s.t. } &\{\mathbf{C}, \mathbf{K}\} \geq 0\end{aligned} \tag{4}$$

where $\mathbf{I}$ denotes an identify matrix, λ_5 is a regularization hyperparameter while α and β are both positive hyperparameters. The value of these hyperparameters is usually set to a large number, e.g. 10^9 in this work, for adjusting the contribution of corresponding terms. The obtained communities in Eq 4 are directly linked to the underlying graph topologies, i.e., $\mathbf{A}^{\mathbf{prox}}$ and $\mathbf{B}^{\mathbf{prox}}$.

5.2.3 Multi-label Learning Process

We now bring together the NMF and community detection steps with multi-label classification for pathway prediction. The learning problem must balance between information in $\mathbf{M}$ while being lenient towards the dataset $\mathcal{S}$, which should provide enough evidence to generate representations of communities among pathways and ECs, as suggested by $\mathbf{A}^{\mathbf{prox}}$ and $\mathbf{B}^{\mathbf{prox}}$. We present a weight term $\Theta \in \mathbb{R}^{t \times r}$ that enforces $\mathbf{X}$ to be close enough to both $\mathbf{Y}$ and $\mathbf{M}$. We also introduce two auxiliary terms $\mathbf{L} \in \mathbb{R}^{n \times m}$, which capture correlations between $\mathbf{X}$ and $\mathbf{Y}$ and $\mathbf{Z} \in \mathbb{R}^{r \times r}$, enforcing the pathway coefficients associated with $\mathbf{M}$ resulting in the following objective function:

$$\begin{aligned}\mathcal{J}^{\mathbf{path}}(\mathbf{T}, \mathbf{R}, \Theta, \mathbf{L}, \mathbf{Z}) = &\min_{\mathbf{T},\mathbf{R},\Theta,\mathbf{L},\mathbf{Z}} \sum_{i \in n} \sum_{k \in t} \log\left(1 + e^{-\mathbf{y}_k^{(i)} \Theta_k^\intercal \mathbf{x}^{(i)}}\right) \\ &+ ||\mathbf{X} - \mathbf{L}\mathbf{R}\mathbf{K}^\top||_F^2 + ||\mathbf{Y} - \mathbf{L}\mathbf{T}\mathbf{C}^\top||_F^2 \\ &+ \rho||\Theta - \mathbf{Z}\mathbf{H}\mathbf{W}^\top||_F^2 \\ &+ \lambda_5(||\mathbf{T}||_F^2 + ||\mathbf{R}||_F^2) \\ &+ \lambda_6(||\Theta||_{2,1} + ||\mathbf{L}||_F^2 + ||\mathbf{Z}||_F^2) \\ \text{s.t. } &\{\mathbf{T}, \mathbf{R}\} \geq 0\end{aligned} \tag{5}$$

where λ_5, λ_6, and ρ are regularization hyperparameters, and $||.||_{2,1}$ represents the sum of the Euclidean norms of columns of a matrix introduced to emphasize sparseness. Notice that we do not restrict the terms $\mathbf{L}$ and $\mathbf{Z}$ to be non-negative. Both the second and the third terms in Eq. 5, are needed to discover pathway and EC communities, i.e., $\mathbf{C}$ and $\mathbf{K}$, respectively.

The Eqs 2, 4, and 5 are jointly non-convex due to non-negative constraints on the original and the approximation factorized matrices, implying the solutions to triUMPF are only unique up to scalings and rotations [37]. Hence, we adopt an alternating optimization algorithm to solve each objective function simultaneously, which is provided in Section 5.3.

5.3 Appendix A3: Optimization

In this section, we derive the optimization for triUMPF's objective function:

$$\mathcal{J} = \mathcal{J}^{\mathbf{fact}}(\mathbf{W}, \mathbf{H}, \mathbf{U}, \mathbf{V}) + \mathcal{J}^{\mathbf{comm}}(\mathbf{C}, \mathbf{K}) + \mathcal{J}^{\mathbf{path}}(\mathbf{T}, \mathbf{R}, \Theta, \mathbf{Z}, \mathbf{L}) \tag{6}$$

where,

$$
\begin{aligned}
\mathcal{J}^{\mathbf{fact}}(\mathbf{W},\mathbf{H},\mathbf{U},\mathbf{V}) = &\min_{\mathbf{W},\mathbf{H},\mathbf{U},\mathbf{V}} ||\mathbf{M}-\mathbf{W}\mathbf{H}^\top||_F^2 + \lambda_1||\mathbf{W}-\mathbf{P}\mathbf{U}||_F^2 \\
&+ \lambda_2||\mathbf{H}-\mathbf{E}\mathbf{V}||_F^2 + \lambda_3||\mathbf{U}-\mathbf{V}||_F^2 \\
&+ \lambda_4(||\mathbf{W}||_F^2 + ||\mathbf{H}||_F^2 + ||\mathbf{U}||_F^2 + ||\mathbf{V}||_F^2) \\
\text{s.t. } &\{\mathbf{W},\mathbf{H},\mathbf{U},\mathbf{V}\} \geq 0 \\
\mathcal{J}^{\mathbf{comm}}(\mathbf{C},\mathbf{K}) = &\min_{\mathbf{C},\mathbf{K}} ||\mathbf{A}^{\mathbf{prox}}-\mathbf{P}\mathbf{T}\mathbf{C}^\top||_F^2 + ||\mathbf{B}^{\mathbf{prox}}-\mathbf{E}\mathbf{R}\mathbf{K}^\top||_F^2 \\
&+ \alpha||\mathbf{C}^\top\mathbf{C}-\mathbf{I}||_F^2 + \beta||\mathbf{K}^\top\mathbf{K}-\mathbf{I}||_F^2 \\
&+ \lambda_5(||\mathbf{C}||_F^2 + ||\mathbf{K}||_F^2) \qquad (7) \\
\text{s.t. } &\{\mathbf{C},\mathbf{K}\} \geq 0 \\
\mathcal{J}^{\mathbf{path}}(\mathbf{T},\mathbf{R},\Theta,\mathbf{L},\mathbf{Z}) = &\min_{\mathbf{T},\mathbf{R},\Theta,\mathbf{L},\mathbf{Z}} \sum_{i\in n}\sum_{k\in t} \log\left(1+e^{-\mathbf{y}_k^{(i)}\Theta_k^\intercal\mathbf{x}^{(i)}}\right) \\
&+ ||\mathbf{X}-\mathbf{L}\mathbf{R}\mathbf{K}^\top||_F^2 + ||\mathbf{Y}-\mathbf{L}\mathbf{T}\mathbf{C}^\top||_F^2 \\
&+ \rho||\Theta-\mathbf{Z}\mathbf{H}\mathbf{W}^\top||_F^2 + \lambda_5(||\mathbf{T}||_F^2 + ||\mathbf{R}||_F^2) \\
&+ \lambda_6(||\Theta||_{2,1} + ||\mathbf{L}||_F^2 + ||\mathbf{Z}||_F^2) \\
\text{s.t. } &\{\mathbf{T},\mathbf{R}\} \geq 0
\end{aligned}
$$

The objective function in Eq. 7 is non-convex due to multiple non-negative constraints. Numerous algorithms have been proposed to optimize the objective function, including alternating non-negative least squares [17] and hierarchical alternating least squares [6]. Here, we employ the original algorithm for NMF which was introduced in [22] and consists of simple multiplicative update rules (with auxiliary variables) that are based on the gradient descent technique [10]. Beginning with random positive initialization, element-wise updates of Eq 6 w.r.t **W**, **H**, **U**, **V**, **C**, **K**, **T**, **R**, Θ, **Z**, and **L** at each iteration are applied until convergence. The gradient descent aims to search for a local minima of the cost function by moving in the direction of its steepest descent. By introducing Lagrangian multipliers (auxiliary variables), which are ψ, ϕ, φ, ϱ, ζ, ϖ, κ, and ξ to enforce the constraints for **W**, **H**, **U**, **V**, **C**, **T**, **R**, **K**, respectively, Eq. 7 can be reformulated as:

$$
\begin{aligned}
\mathcal{J}^{\mathbf{fact}}(\mathbf{W},\mathbf{H},\mathbf{U},\mathbf{V}) = &\min_{\mathbf{W},\mathbf{H},\mathbf{U},\mathbf{V}} \mathbf{tr}\Big((\mathbf{M}-\mathbf{W}\mathbf{H}^\top)^\top(\mathbf{M}-\mathbf{W}\mathbf{H}^\top)\Big) \\
&+ \lambda_1\mathbf{tr}\Big((\mathbf{W}-\mathbf{P}\mathbf{U})^\top(\mathbf{W}-\mathbf{P}\mathbf{U})\Big) \\
&+ \lambda_2\mathbf{tr}\Big((\mathbf{H}-\mathbf{E}\mathbf{V})^\top(\mathbf{H}-\mathbf{E}\mathbf{V})\Big) \\
&+ \lambda_3\mathbf{tr}\Big((\mathbf{U}-\mathbf{V})^\top(\mathbf{U}-\mathbf{V})\Big) \\
&+ \lambda_4\Big(\mathbf{tr}(\mathbf{W}^\top\mathbf{W}) + \mathbf{tr}(\mathbf{H}^\top\mathbf{H}) + \mathbf{tr}(\mathbf{U}^\top\mathbf{U}) + \mathbf{tr}(\mathbf{V}^\top\mathbf{V})\Big) \\
&+ \mathbf{tr}(\psi\mathbf{W}) + \mathbf{tr}(\phi\mathbf{H}) + \mathbf{tr}(\varphi\mathbf{U}) + \mathbf{tr}(\varrho\mathbf{V})
\end{aligned} \qquad (8)
$$

$$
\begin{aligned}
\mathcal{J}^{\mathbf{comm}}(\mathbf{C},\mathbf{K}) = &\min_{\mathbf{C},\mathbf{K}} \mathbf{tr}\Big((\mathbf{A}^{\mathbf{prox}}-\mathbf{P}\mathbf{T}\mathbf{C}^\top)^\top(\mathbf{A}^{\mathbf{prox}}-\mathbf{P}\mathbf{T}\mathbf{C}^\top)\Big) \\
&+ \mathbf{tr}\Big((\mathbf{B}^{\mathbf{prox}}-\mathbf{E}\mathbf{R}\mathbf{K}^\top)^\top(\mathbf{B}^{\mathbf{prox}}-\mathbf{E}\mathbf{R}\mathbf{K}^\top)\Big) \\
&+ \alpha\mathbf{tr}\Big((\mathbf{C}^\top\mathbf{C}-\mathbf{I})^\top(\mathbf{C}^\top\mathbf{C}-\mathbf{I})\Big) \\
&+ \beta\mathbf{tr}\Big((\mathbf{K}^\top\mathbf{K}-\mathbf{I})^\top(\mathbf{K}^\top\mathbf{K}-\mathbf{I})\Big) \\
&+ \lambda_5\Big(\mathbf{tr}(\mathbf{C}^\top\mathbf{C}) + \mathbf{tr}(\mathbf{K}^\top\mathbf{K})\Big) + \mathbf{tr}(\varpi\mathbf{C}) + \mathbf{tr}(\xi\mathbf{K})
\end{aligned} \qquad (9)
$$

$$
\begin{aligned}
\mathcal{J}^{\mathbf{path}}(\mathbf{T},\mathbf{R},\Theta,\mathbf{L},\mathbf{Z}) = \min_{\mathbf{T},\mathbf{R},\Theta,\mathbf{L},\mathbf{Z}} & \sum_{i\in n}\sum_{k\in t} \log\left(1+e^{-\mathbf{y}_k^{(i)}\Theta_k^{\intercal}\mathbf{x}^{(i)}}\right) \\
& + \mathbf{tr}\Big((\mathbf{X}-\mathbf{LRK}^\top)^\top(\mathbf{X}-\mathbf{LRK}^\top)\Big) \\
& + \mathbf{tr}\Big((\mathbf{Y}-\mathbf{LTC}^\top)^\top(\mathbf{Y}-\mathbf{LTC}^\top)\Big) \\
& + \rho\mathbf{tr}\Big((\Theta-\mathbf{ZHW}^\top)^\top(\Theta-\mathbf{ZHW}^\top)\Big) \\
& + \lambda_5\Big(\mathbf{tr}(\mathbf{T}^\top\mathbf{T})+\mathbf{tr}(\mathbf{R}^\top\mathbf{R})\Big) \\
& + \lambda_6\Big(||\Theta||_{2,1}+\mathbf{tr}(\mathbf{L}^\top\mathbf{L})+\mathbf{tr}(\mathbf{Z}^\top\mathbf{Z})\Big) \\
& + \mathbf{tr}(\zeta\mathbf{T})+\mathbf{tr}(\kappa\mathbf{R})
\end{aligned}
\tag{10}
$$

where $\mathbf{tr}(.)$ denotes the trace of a matrix. Using the addition property of the transpose, $(\mathbf{X}+\mathbf{Y})^\top=\mathbf{X}^\top+\mathbf{Y}^\top$, and its multiplication property, $(\mathbf{XY})^\top=\mathbf{Y}^\top\mathbf{X}^\top$, we can expand the trace of the first term as

$$
\mathbf{tr}\Big((\mathbf{M}-\mathbf{WH}^\top)^\top(\mathbf{M}-\mathbf{WH}^\top)\Big) = \mathbf{tr}\Big(\mathbf{M}^\top\mathbf{M}-\mathbf{M}^\top\mathbf{WH}^\top-\mathbf{W}^\top\mathbf{HM}+\mathbf{HW}^\top\mathbf{WH}^\top\Big) \tag{11}
$$

By expanding the remaining terms in Eq. 8 and using the trace of a sum of matrix property, $\mathbf{tr}(\mathbf{X}+\mathbf{Y})=\mathbf{tr}(\mathbf{X})+\mathbf{tr}(\mathbf{Y})$, we obtain the following formula:

$$
\begin{aligned}
\mathcal{J}^{\mathbf{fact}}(\mathbf{W},\mathbf{H},\mathbf{U},\mathbf{V}) = \min_{\mathbf{W},\mathbf{H},\mathbf{U},\mathbf{V}} & \mathbf{tr}(\mathbf{M}^\top\mathbf{M})-\mathbf{tr}(\mathbf{M}^\top\mathbf{WH}^\top)-\mathbf{tr}(\mathbf{W}^\top\mathbf{HM})+\mathbf{tr}(\mathbf{HW}^\top\mathbf{WH}^\top) \\
& + \lambda_1\Big(\mathbf{tr}(\mathbf{W}^\top\mathbf{W})-\mathbf{tr}(\mathbf{W}^\top\mathbf{PU})-\mathbf{tr}(\mathbf{U}^\top\mathbf{P}^\top\mathbf{W})+\mathbf{tr}(\mathbf{U}^\top\mathbf{P}^\top\mathbf{PU})\Big) \\
& + \lambda_2\Big(\mathbf{tr}(\mathbf{H}^\top\mathbf{H})-\mathbf{tr}(\mathbf{H}^\top\mathbf{EV})-\mathbf{tr}(\mathbf{V}^\top\mathbf{E}^\top\mathbf{H})+\mathbf{tr}(\mathbf{V}^\top\mathbf{E}^\top\mathbf{EV})\Big) \\
& + \lambda_3\Big(\mathbf{tr}(\mathbf{U}^\top\mathbf{U})-2\mathbf{tr}(\mathbf{U}^\top\mathbf{V})+\mathbf{tr}(\mathbf{V}^\top\mathbf{V})\Big) \\
& + \lambda_4\Big(\mathbf{tr}(\mathbf{W}^\top\mathbf{W})+\mathbf{tr}(\mathbf{H}^\top\mathbf{H})+\mathbf{tr}(\mathbf{U}^\top\mathbf{U})+\mathbf{tr}(\mathbf{V}^\top\mathbf{V})\Big) \\
& + \mathbf{tr}(\psi\mathbf{W})+\mathbf{tr}(\phi\mathbf{H})+\mathbf{tr}(\varphi\mathbf{U})+\mathbf{tr}(\varrho\mathbf{V})
\end{aligned}
\tag{12}
$$

Similar to the process of getting Eq. 12, we expand the Eq. 9 as:

$$
\begin{aligned}
\mathcal{J}^{\mathbf{comm}}(\mathbf{C},\mathbf{K}) = \min_{\mathbf{C},\mathbf{K}} & \mathbf{tr}(\mathbf{A}^{\mathbf{prox}\top}\mathbf{A}^{\mathbf{prox}})-\mathbf{tr}(\mathbf{A}^{\mathbf{prox}\top}\mathbf{PTC}^\top) \\
& - \mathbf{tr}(\mathbf{CT}^\top\mathbf{P}^\top\mathbf{A}^{\mathbf{prox}})+\mathbf{tr}(\mathbf{CT}^\top\mathbf{P}^\top\mathbf{PTC}^\top) \\
& + \mathbf{tr}(\mathbf{B}^{\mathbf{prox}\top}\mathbf{B}^{\mathbf{prox}})-\mathbf{tr}(\mathbf{B}^{\mathbf{prox}\top}\mathbf{ERK}^\top) \\
& - \mathbf{tr}(\mathbf{KR}^\top\mathbf{E}^\top\mathbf{B}^{\mathbf{prox}})+\mathbf{tr}(\mathbf{KR}^\top\mathbf{E}^\top\mathbf{ERK}^\top) \\
& + \alpha\Big(\mathbf{tr}(\mathbf{C}^\top\mathbf{CC}^\top\mathbf{C})-2\mathbf{tr}(\mathbf{C}^\top\mathbf{C})+t\Big) \\
& + \beta\Big(\mathbf{tr}(\mathbf{K}^\top\mathbf{KK}^\top\mathbf{K})-2\mathbf{tr}(\mathbf{K}^\top\mathbf{K})+r\Big) \\
& + \lambda_5\Big(\mathbf{tr}(\mathbf{C}^\top\mathbf{C})+\mathbf{tr}(\mathbf{K}^\top\mathbf{K})\Big)+\mathbf{tr}(\varpi\mathbf{C})+\mathbf{tr}(\xi\mathbf{K})
\end{aligned}
\tag{13}
$$

Expand the Eq. 10, we obtain the following:

$$\begin{aligned}\mathcal{J}^{\mathbf{path}}(\mathbf{T},\mathbf{R},\Theta,\mathbf{L},\mathbf{Z}) = \min_{\mathbf{T},\mathbf{R},\Theta,\mathbf{L},\mathbf{Z}} & \sum_{i\in n}\sum_{k\in t} \log\left(1+e^{-\mathbf{y}_k^{(i)}\Theta_k^{\intercal}\mathbf{x}^{(i)}}\right) \\ & + \mathbf{tr}(\mathbf{X}^\top\mathbf{X}) - \mathbf{tr}(\mathbf{X}^\top\mathbf{LRK}^\top) - \mathbf{tr}(\mathbf{KR}^\top\mathbf{L}^\top\mathbf{X}) \\ & + \mathbf{tr}(\mathbf{KR}^\top\mathbf{L}^\top\mathbf{LRK}^\top) + \mathbf{tr}(\mathbf{Y}^\top\mathbf{Y}) \\ & - \mathbf{tr}(\mathbf{Y}^\top\mathbf{LTC}^\top) - \mathbf{tr}(\mathbf{CT}^\top\mathbf{L}^\top\mathbf{Y}) + \mathbf{tr}(\mathbf{CT}^\top\mathbf{L}^\top\mathbf{LTC}^\top) \\ & + \rho\Big(\mathbf{tr}(\Theta^\top\Theta) - \mathbf{tr}(\Theta^\top\mathbf{ZHW}^\top) - \mathbf{tr}(\mathbf{WH}^\top\mathbf{Z}^\top\Theta) \\ & + \mathbf{tr}(\mathbf{WH}^\top\mathbf{Z}^\top\mathbf{ZHW}^\top)\Big) \\ & + \lambda_5\Big(\mathbf{tr}(\mathbf{T}^\top\mathbf{T}) + \mathbf{tr}(\mathbf{R}^\top\mathbf{R})\Big) + \lambda_6\Big(||\Theta||_{2,1} + \mathbf{tr}(\mathbf{L}^\top\mathbf{L}) \\ & + \mathbf{tr}(\mathbf{Z}^\top\mathbf{Z})\Big) + \mathbf{tr}(\zeta\mathbf{T}) + \mathbf{tr}(\kappa\mathbf{R})\end{aligned} \tag{14}$$

As explained earlier, the objective functions in Eqs 12, 13, and 14, are not convex with respect to all parameters combined. Instead in NMF, $\mathbf{W}$, $\mathbf{H}$, $\mathbf{U}$, $\mathbf{V}$, $\mathbf{C}$, $\mathbf{K}$, $\mathbf{T}$, $\mathbf{R}$, Θ, $\mathbf{L}$, and $\mathbf{Z}$ are individually optimized in an iterative process, where we update one matrix at a time while keeping the remaining matrices fixed. This ensures convergence to a local minima for each subproblem. This methods is called *block-coordinate descent*. Hence, the update of parameters occur in the following four alternate optimization steps for $\mathcal{J}^{\mathbf{fact}}$: i)- the basis matrix $\mathbf{W}$, representing pathway factors, ii)- the latent coefficient matrix $\mathbf{H}$, representing EC factors, iii)- the linear transformation $\mathbf{U}$, and iv)- the other linear transformation $\mathbf{V}$. For $\mathcal{J}^{\mathbf{comm}}$, we alternate between the community indicator matrix $\mathbf{C}$ for pathways and the other community indicator matrix $\mathbf{K}$ for ECs. Finally, we optimize, alternatively, the two community representation matrices $\mathbf{T}$ and $\mathbf{R}$ for pathways and ECs, respectively, the two auxiliary matrices $\mathbf{L}$ and $\mathbf{Z}$, and the input weight matrix Θ. The three objective functions, $\mathcal{J}^{\mathbf{fact}}$, $\mathcal{J}^{\mathbf{comm}}$, and $\mathcal{J}^{\mathbf{path}}$ are run simultaneously in a divide and conquer strategy. Detailed rules for updating all the variables are outlined below.

1. **Update the basis matrix W.** To update the feature matrix $\mathbf{W}$, we fix $\mathbf{H}$, $\mathbf{U}$ and $\mathbf{V}$. Then, the objective function in Eq. 12 w.r.t $\mathbf{W}$ is reduced to the following formula (after dropping the min operation):

$$\begin{aligned}\mathcal{J}^{\mathbf{fact}}(\mathbf{W}) = & -\mathbf{tr}(\mathbf{M}^\top\mathbf{WH}^\top) - \mathbf{tr}(\mathbf{W}^\top\mathbf{HM}) + \mathbf{tr}(\mathbf{HW}^\top\mathbf{WH}^\top) \\ & + \lambda_1\Big(\mathbf{tr}(\mathbf{W}^\top\mathbf{W}) - \mathbf{tr}(\mathbf{W}^\top\mathbf{PU}) - \mathbf{tr}(\mathbf{U}^\top\mathbf{P}^\top\mathbf{W})\Big) \\ & + \lambda_4\mathbf{tr}(\mathbf{W}^\top\mathbf{W}) + \mathbf{tr}(\psi\mathbf{W})\end{aligned} \tag{15}$$

where ψ is the Lagrange multiplier for the constraint $\mathbf{W} \geq 0$. For computing the gradient of this equation, we use the following properties with respect to $\mathbf{X}$:

$$\begin{aligned}\nabla_{\mathbf{X}}\mathbf{tr}(\mathbf{X}^\top\mathbf{X}) =& 2\mathbf{X} \\ \nabla_{\mathbf{X}}\mathbf{tr}(\mathbf{XY}) =& \mathbf{Y}^\top \\ \nabla_{\mathbf{X}}\mathbf{tr}(\mathbf{X}^\top\mathbf{Y}) =& \mathbf{Y} \\ \nabla_{\mathbf{X}}\mathbf{tr}(\mathbf{X}^\top\mathbf{YX}) =& (\mathbf{Y}+\mathbf{Y}^\top)\mathbf{X} \\ \nabla_{\mathbf{X}}\mathbf{tr}(\mathbf{XYX}^\top) =& \mathbf{X}(\mathbf{Y}^\top+\mathbf{Y}) \\ \nabla_{\mathbf{X}}\mathbf{tr}(\mathbf{YXZ}) =& \mathbf{Y}^\top\mathbf{Z}^\top \\ \nabla_{\mathbf{X}}\mathbf{tr}(\mathbf{YX}^\top\mathbf{Z}) =& \mathbf{ZY}\end{aligned} \tag{16}$$

By computing the gradient of the cost function in Eq. 15 w.r.t $\mathbf{W}$ to 0, we have:

$$\psi = 2\mathbf{MH} - 2\mathbf{W}(\mathbf{H}^\top\mathbf{H} + Q) + 2\lambda_1\mathbf{PU} \tag{17}$$

where $Q = (\lambda_1 + \lambda_4)$. Following the Karush-Kuhn-Tucker (KKT) condition for the non-negativity of $\mathbf{W}$, we have the following equation:

$$2\Big(\mathbf{MH} - \mathbf{W}(\mathbf{H}^\top\mathbf{H} + Q) + \lambda_1\mathbf{PU}\Big)_{k,j}\mathbf{W}_{j,k} = \psi_{j,k}\mathbf{W}_{j,k} = 0 \tag{18}$$

Given an initial value of $\mathbf{W}$, the successive updating rule of $\mathbf{W}$ is:

$$\mathbf{W} \leftarrow \mathbf{W} \circ \frac{\mathbf{MH} + \lambda_1 \mathbf{PU}}{\mathbf{W}(\mathbf{H}^\top \mathbf{H} + Q)} \tag{19}$$

The iterative update rules in Eq. 19 is transformed into multiplicative update rules, which cannot generate negative elements since all values are positive and only multiplications and divisions are involved at each iteration [21].

2. **Update the latent coefficient matrix $\mathbf{H}$.** The feature matrix $\mathbf{H}$ is updates as described above in which $\mathbf{W}$, $\mathbf{U}$ and $\mathbf{V}$ are fixed to obtain the objective function for Eq. 12 w.r.t $\mathbf{H}$ as:

$$\begin{aligned}\mathcal{J}^{\mathbf{fact}}(\mathbf{H}) =& -\mathbf{tr}(\mathbf{M}^\top \mathbf{W}\mathbf{H}^\top) - \mathbf{tr}(\mathbf{W}^\top \mathbf{H}\mathbf{M}) + \mathbf{tr}(\mathbf{H}\mathbf{W}^\top \mathbf{W}\mathbf{H}^\top) \\ &+ \lambda_1 \Big(\mathbf{tr}(\mathbf{H}^\top \mathbf{H}) - \mathbf{tr}(\mathbf{H}^\top \mathbf{E}\mathbf{V}) - \mathbf{tr}(\mathbf{V}^\top \mathbf{E}^\top \mathbf{H})\Big) \\ &+ \lambda_4 \mathbf{tr}(\mathbf{H}^\top \mathbf{H}) + \mathbf{tr}(\phi \mathbf{H})\end{aligned} \tag{20}$$

Taking the derivative of the cost function in Eq. 20 w.r.t $\mathbf{H}$ to 0 and using the gradient properties in Eq. 16, we obtain the following:

$$\phi = 2\mathbf{M}^\top \mathbf{W} - 2\mathbf{H}(\mathbf{W}^\top \mathbf{W} + Q) + 2\lambda_1 \mathbf{E}\mathbf{V} \tag{21}$$

where $Q = (\lambda_1 + \lambda_4)$. With the KKT complementary condition for the nonnegativity of $\mathbf{H}$, we have:

$$2\Big(\mathbf{M}^\top \mathbf{W} - \mathbf{H}(\mathbf{W}^\top \mathbf{W} + Q) + \lambda_1 \mathbf{E}\mathbf{V}\Big)_{j,k} \mathbf{H}_{j,k} = \phi_{j,k} \mathbf{H}_{j,k} = 0 \tag{22}$$

The multiplicative updates after some algebraic manipulation w.r.t parameter $\mathbf{H}$:

$$\mathbf{H} \leftarrow \mathbf{H} \circ \frac{\mathbf{M}^\top \mathbf{W} + \lambda_1 \mathbf{E}\mathbf{V}}{\mathbf{H}(\mathbf{W}^\top \mathbf{W} + Q)} \tag{23}$$

3. **Update the linear transformation $\mathbf{U}$.** Suppose that $\mathbf{W}$, $\mathbf{H}$ and $\mathbf{V}$ are fixed, then Eq. 12 w.r.t $\mathbf{U}$ is reduced to:

$$\begin{aligned}\mathcal{J}^{\mathbf{fact}}(\mathbf{U}) =& \lambda_1 \Big(-\mathbf{tr}(\mathbf{W}^\top \mathbf{P}\mathbf{U}) - \mathbf{tr}(\mathbf{U}^\top \mathbf{P}^\top \mathbf{W}) + \mathbf{tr}(\mathbf{U}^\top \mathbf{P}^\top \mathbf{P}\mathbf{U})\Big) \\ &+ \lambda_3 \Big(\mathbf{tr}(\mathbf{U}^\top \mathbf{U}) - 2\mathbf{tr}(\mathbf{U}^\top \mathbf{V})\Big) + \lambda_4 \mathbf{tr}(\mathbf{U}^\top \mathbf{U}) + \mathbf{tr}(\varphi \mathbf{U})\end{aligned} \tag{24}$$

Then we take the derivative of above formula with respect to the transformation matrix $\mathbf{U}$ to 0:

$$\varphi = 2\lambda_1 \mathbf{P}^\top \mathbf{W} - 2(\lambda_1 \mathbf{P}^\top \mathbf{P} + D)\mathbf{U} + 2\lambda_3 \mathbf{V} \tag{25}$$

where $D = (\lambda_3 + \lambda_4)$. Formulating the above equation based on Karush–Kuhn–Tucker conditions for the nonnegativity of $\mathbf{U}$ results in:

$$2\Big(\lambda_1 \mathbf{P}^\top \mathbf{W} - (\lambda_1 \mathbf{P}^\top \mathbf{P} + D)\mathbf{U} + \lambda_3 \mathbf{V}\Big)_{j,k} \mathbf{U}_{j,k} = \varphi_{j,k} \mathbf{U}_{j,k} = 0 \tag{26}$$

Then, the parameter $\mathbf{U}$ is updated according to:

$$\mathbf{U} \leftarrow \mathbf{U} \circ \frac{\lambda_1 \mathbf{P}^\top \mathbf{W} + \lambda_3 \mathbf{V}}{(\lambda_1 \mathbf{P}^\top \mathbf{P} + D)\mathbf{U}} \tag{27}$$

4. **Update the linear transformation $\mathbf{V}$.** To update the linear transformation matrix $\mathbf{V}$, that $\mathbf{W}$, $\mathbf{H}$ and $\mathbf{U}$ are fixed, then the transformation matrix $\mathbf{V}$ is updated such that the error is minimized:

$$\begin{aligned}\mathcal{J}^{\mathbf{fact}}(\mathbf{V}) =& \lambda_2 \Big(-\mathbf{tr}(\mathbf{H}^\top \mathbf{E}\mathbf{V}) - \mathbf{tr}(\mathbf{V}^\top \mathbf{E}^\top \mathbf{H}) + \mathbf{tr}(\mathbf{V}^\top \mathbf{E}^\top \mathbf{E}\mathbf{V})\Big) \\ &+ \lambda_3 \Big(-2\mathbf{tr}(\mathbf{U}^\top \mathbf{V}) + \mathbf{tr}(\mathbf{V}^\top \mathbf{V})\Big) + \lambda_4 \mathbf{tr}(\mathbf{V}^\top \mathbf{V}) + \mathbf{tr}(\varrho \mathbf{V})\end{aligned} \tag{28}$$

Taking the derivative of this error with respect to $\mathbf{V}$ to 0 and after some manipulations, we have:

$$\varrho = 2\lambda_2\mathbf{E}^\top\mathbf{H} - 2(\lambda_2\mathbf{E}^\top\mathbf{E} + D)\mathbf{V} + 2\lambda_3\mathbf{U} \tag{29}$$

where $D = (\lambda_3 + \lambda_4)$. Following the Karush–Kuhn–Tucker conditions for the nonnegativity of $\mathbf{V}$, we have:

$$2\Big(\lambda_2\mathbf{E}^\top\mathbf{H} - (\lambda_2\mathbf{E}^\top\mathbf{E} + D)\mathbf{V} + \lambda_3\mathbf{U}\Big)_{j,k}\mathbf{V}_{j,k} = \varrho_{j,k}\mathbf{V}_{j,k} = 0 \tag{30}$$

As usual, the parameter $\mathbf{V}$ is updated according:

$$\mathbf{V} \leftarrow \mathbf{V} \circ \frac{\lambda_2\mathbf{E}^\top\mathbf{H} + \lambda_3\mathbf{U}}{(\lambda_2\mathbf{E}^\top\mathbf{E} + D)\mathbf{V}} \tag{31}$$

5. **Update the community indicator matrix $\mathbf{C}$ for pathways.** In a similar process, we fix $\mathbf{K}$, and update $\mathbf{C}$. The matrix $\mathbf{C}$ is updated such that the error is minimized:

$$\begin{aligned}\mathcal{J}(\mathbf{C}) =& -\mathbf{tr}(\mathbf{A}^{\mathbf{prox}\top}\mathbf{P}\mathbf{T}\mathbf{C}^\top) - \mathbf{tr}(\mathbf{C}\mathbf{T}^\top\mathbf{P}^\top\mathbf{A}^{\mathbf{prox}}) + \mathbf{tr}(\mathbf{C}\mathbf{T}^\top\mathbf{P}^\top\mathbf{P}\mathbf{T}\mathbf{C}^\top)\\ &+ \alpha\Big(\mathbf{tr}(\mathbf{C}^\top\mathbf{C}\mathbf{C}^\top\mathbf{C}) - 2\mathbf{tr}(\mathbf{C}^\top\mathbf{C})\Big) + \lambda_5\mathbf{tr}(\mathbf{C}^\top\mathbf{C}) + \mathbf{tr}(\varpi\mathbf{C})\\ &- \mathbf{tr}(\mathbf{Y}^\top\mathbf{L}\mathbf{T}\mathbf{C}^\top) - \mathbf{tr}(\mathbf{C}\mathbf{T}^\top\mathbf{L}^\top\mathbf{Y}) + \mathbf{tr}(\mathbf{C}\mathbf{T}^\top\mathbf{L}^\top\mathbf{L}\mathbf{T}\mathbf{C}^\top)\end{aligned} \tag{32}$$

Taking the derivative of this error with respect to $\mathbf{C}$ to 0, we have:

$$\varpi = 2\mathbf{A}^{\mathbf{prox}\top}\mathbf{P}\mathbf{T} + 2\mathbf{Y}^\top\mathbf{L}\mathbf{T} + 4\alpha\mathbf{C} - 2\mathbf{C}(\mathbf{T}^\top\mathbf{P}^\top\mathbf{P}\mathbf{T} + \mathbf{T}^\top\mathbf{L}^\top\mathbf{L}\mathbf{T} + 2\alpha\mathbf{C}^\top\mathbf{C} + \lambda_5) \tag{33}$$

Again, we follow the Karush–Kuhn–Tucker conditions for the nonnegativity of $\mathbf{C}$

$$2\Big(\mathbf{A}^{\mathbf{prox}\top}\mathbf{P}\mathbf{T} + \mathbf{Y}^\top\mathbf{L}\mathbf{T} + 2\alpha\mathbf{C} - \mathbf{C}(\mathbf{T}^\top\mathbf{P}^\top\mathbf{P}\mathbf{T} + \mathbf{T}^\top\mathbf{L}^\top\mathbf{L}\mathbf{T} + 2\alpha\mathbf{C}^\top\mathbf{C} + \lambda_5)\Big)_{j,k}\mathbf{C}_{j,k} = \varpi_{j,k}\mathbf{C}_{j,k} = 0 \tag{34}$$

The parameter $\mathbf{C}$ is updated according:

$$\mathbf{C} \leftarrow \mathbf{C} \circ \frac{\mathbf{A}^{\mathbf{prox}\top}\mathbf{P}\mathbf{T} + \mathbf{Y}^\top\mathbf{L}\mathbf{T} + 2\alpha\mathbf{C}}{\mathbf{C}(\mathbf{T}^\top\mathbf{P}^\top\mathbf{P}\mathbf{T} + \mathbf{T}^\top\mathbf{L}^\top\mathbf{L}\mathbf{T} + 2\alpha\mathbf{C}^\top\mathbf{C} + \lambda_5)} \tag{35}$$

6. **Update the community indicator matrix $\mathbf{K}$ for ECs.** Once the parameter $\mathbf{C}$ is updated, we use it to update $\mathbf{K}$. The matrix $\mathbf{K}$ is updated such that the error is minimized:

$$\begin{aligned}\mathcal{J}(\mathbf{K}) =& -\mathbf{tr}(\mathbf{B}^{\mathbf{prox}\top}\mathbf{E}\mathbf{R}\mathbf{K}^\top) - \mathbf{tr}(\mathbf{K}\mathbf{R}^\top\mathbf{E}^\top\mathbf{B}^{\mathbf{prox}}) + \mathbf{tr}(\mathbf{K}\mathbf{R}^\top\mathbf{E}^\top\mathbf{E}\mathbf{R}\mathbf{K}^\top)\\ &+ \beta\Big(\mathbf{tr}(\mathbf{K}^\top\mathbf{K}\mathbf{K}^\top\mathbf{K}) - 2\mathbf{tr}(\mathbf{K}^\top\mathbf{K})\Big) + \lambda_5\mathbf{tr}(\mathbf{K}^\top\mathbf{K}) + \mathbf{tr}(\xi\mathbf{K})\\ &- \mathbf{tr}(\mathbf{X}^\top\mathbf{L}\mathbf{R}\mathbf{K}^\top) - \mathbf{tr}(\mathbf{K}\mathbf{R}^\top\mathbf{L}^\top\mathbf{X}) + \mathbf{tr}(\mathbf{K}\mathbf{R}^\top\mathbf{L}^\top\mathbf{L}\mathbf{R}\mathbf{K}^\top)\end{aligned} \tag{36}$$

Taking the derivative of this error with respect to $\mathbf{K}$ to 0, we have:

$$\xi = 2\mathbf{B}^{\mathbf{prox}\top}\mathbf{E}\mathbf{R} + 2\mathbf{X}^\top\mathbf{L}\mathbf{R} + 4\beta\mathbf{K} - 2\mathbf{K}(\mathbf{R}^\top\mathbf{E}^\top\mathbf{E}\mathbf{R} + \mathbf{R}^\top\mathbf{L}^\top\mathbf{L}\mathbf{R} + 2\beta\mathbf{K}^\top\mathbf{K} + \lambda_5) \tag{37}$$

Using the Karush–Kuhn–Tucker conditions for the nonnegativity of $\mathbf{K}$, we obtain:

$$2\Big(\mathbf{B}^{\mathbf{prox}\top}\mathbf{E}\mathbf{R} + \mathbf{X}^\top\mathbf{L}\mathbf{R} + 2\beta\mathbf{K} - \mathbf{K}(\mathbf{R}^\top\mathbf{E}^\top\mathbf{E}\mathbf{R} + \mathbf{R}^\top\mathbf{L}^\top\mathbf{L}\mathbf{R} + 2\beta\mathbf{K}^\top\mathbf{K} + \lambda_5)\Big)_{j,k}\mathbf{K}_{j,k} = \xi_{j,k}\mathbf{K}_{j,k} = 0 \tag{38}$$

The parameter $\mathbf{K}$ is updated according:

$$\mathbf{K} \leftarrow \mathbf{K} \circ \frac{\mathbf{B}^{\mathbf{prox}\top}\mathbf{E}\mathbf{R} + \mathbf{X}^\top\mathbf{L}\mathbf{R} + 2\beta\mathbf{K}}{\mathbf{K}(\mathbf{R}^\top\mathbf{E}^\top\mathbf{E}\mathbf{R} + \mathbf{R}^\top\mathbf{L}^\top\mathbf{L}\mathbf{R} + 2\beta\mathbf{K}^\top\mathbf{K} + \lambda_5)} \tag{39}$$

7. **Update the community representation matrix T for pathways.** By fixing the parameters $\mathbf{C}$, $\mathbf{R}$, and $\mathbf{K}$, we update $\mathbf{T}$. The matrix $\mathbf{T}$ is updated such that the error is minimized:

$$\begin{aligned}\mathcal{J}(\mathbf{T}) = &- \mathbf{tr}(\mathbf{A}^{\mathbf{prox}\top}\mathbf{PTC}^\top) - \mathbf{tr}(\mathbf{CT}^\top\mathbf{P}^\top\mathbf{A}^{\mathbf{prox}}) \\ &+ \mathbf{tr}(\mathbf{CT}^\top\mathbf{P}^\top\mathbf{PTC}^\top) - \mathbf{tr}(\mathbf{Y}^\top\mathbf{LTC}^\top) \\ &- \mathbf{tr}(\mathbf{CT}^\top\mathbf{L}^\top\mathbf{Y}) + \mathbf{tr}(\mathbf{CT}^\top\mathbf{L}^\top\mathbf{LTC}^\top) \\ &+ \lambda_5\mathbf{tr}(\mathbf{T}^\top\mathbf{T}) + \mathbf{tr}(\zeta\mathbf{T})\end{aligned} \tag{40}$$

Taking the derivative of this error with respect to $\mathbf{T}$ to 0, we have:

$$\zeta = 2\mathbf{P}^\top\mathbf{A}^{\mathbf{prox}}\mathbf{C} + 2\mathbf{L}^\top\mathbf{YC} - 2(\mathbf{P}^\top\mathbf{CC}^\top\mathbf{P} + \lambda_5)\mathbf{T} - 2\mathbf{L}^\top\mathbf{LTC}^\top\mathbf{C} \tag{41}$$

Using the Karush–Kuhn–Tucker conditions for the nonnegativity of $\mathbf{T}$, we obtain:

$$2\Big(\mathbf{P}^\top\mathbf{A}^{\mathbf{prox}}\mathbf{C} + \mathbf{L}^\top\mathbf{YC} - (\mathbf{P}^\top\mathbf{CC}^\top\mathbf{P} + \lambda_5)\mathbf{T} - \mathbf{L}^\top\mathbf{LTC}^\top\mathbf{C}\Big)_{j,k}\mathbf{T}_{j,k} = \zeta_{j,k}\mathbf{T}_{j,k} = 0 \tag{42}$$

The parameter $\mathbf{T}$ is updated according:

$$\mathbf{T} \leftarrow \mathbf{T} \circ \frac{\mathbf{P}^\top\mathbf{A}^{\mathbf{prox}}\mathbf{C} + \mathbf{L}^\top\mathbf{YC}}{(\mathbf{P}^\top\mathbf{CC}^\top\mathbf{P} + \lambda_5)\mathbf{T} + \mathbf{L}^\top\mathbf{LTC}^\top\mathbf{C}} \tag{43}$$

8. **Update the community representation matrix R for EC features.** By fixing the parameters $\mathbf{C}$, $\mathbf{T}$, and $\mathbf{K}$, we update $\mathbf{R}$. The matrix $\mathbf{R}$ is updated such that the error is minimized:

$$\begin{aligned}\mathcal{J}(\mathbf{R}) = &- \mathbf{tr}(\mathbf{B}^{\mathbf{prox}\top}\mathbf{ERK}^\top) - \mathbf{tr}(\mathbf{KR}^\top\mathbf{E}^\top\mathbf{B}^{\mathbf{prox}}) \\ &+ \mathbf{tr}(\mathbf{KR}^\top\mathbf{E}^\top\mathbf{ERK}^\top) - \mathbf{tr}(\mathbf{X}^\top\mathbf{LRK}^\top) \\ &- \mathbf{tr}(\mathbf{KR}^\top\mathbf{L}^\top\mathbf{X}) + \mathbf{tr}(\mathbf{KR}^\top\mathbf{L}^\top\mathbf{LRK}^\top) \\ &+ \lambda_5\mathbf{tr}(\mathbf{R}^\top\mathbf{R}) + \mathbf{tr}(\kappa\mathbf{R})\end{aligned} \tag{44}$$

Taking the derivative of this error with respect to $\mathbf{R}$ to 0, we have:

$$\kappa = 2\mathbf{E}^\top\mathbf{B}^{\mathbf{prox}}\mathbf{K} + 2\mathbf{L}^\top\mathbf{XK} - 2(\mathbf{E}^\top\mathbf{KK}^\top\mathbf{E} + \lambda_5)\mathbf{R} - 2\mathbf{L}^\top\mathbf{LRK}^\top\mathbf{K} \tag{45}$$

Using the Karush–Kuhn–Tucker conditions for the nonnegativity of $\mathbf{R}$, we obtain:

$$2\Big(\mathbf{E}^\top\mathbf{B}^{\mathbf{prox}}\mathbf{K} + \mathbf{L}^\top\mathbf{XK} - (\mathbf{E}^\top\mathbf{KK}^\top\mathbf{E} + \lambda_5)\mathbf{R} - \mathbf{L}^\top\mathbf{LRK}^\top\mathbf{K}\Big)_{j,k}\mathbf{R}_{j,k} = \kappa_{j,k}\mathbf{R}_{j,k} = 0 \tag{46}$$

The parameter $\mathbf{R}$ is updated according:

$$\mathbf{R} \leftarrow \mathbf{R} \circ \frac{\mathbf{E}^\top\mathbf{B}^{\mathbf{prox}}\mathbf{K} + \mathbf{L}^\top\mathbf{XK}}{(\mathbf{E}^\top\mathbf{KK}^\top\mathbf{E} + \lambda_5)\mathbf{R} + \mathbf{L}^\top\mathbf{LRK}^\top\mathbf{K}} \tag{47}$$

9. **Update the weight matrix Θ.** By fixing the other parameters, we update Θ. The matrix Θ is updated such that the error is minimized:

$$\begin{aligned}\mathcal{J}^{\mathbf{path}}(\Theta) = &\sum_{i\in n}\sum_{k\in t}\log\left(1 + e^{-\mathbf{y}_k^{(i)}\Theta_k^\intercal\mathbf{x}^{(i)}}\right) + \rho\Big(\mathbf{tr}(\Theta^\top\Theta) - \mathbf{tr}(\Theta^\top\mathbf{ZHW}^\top) \\ &- \mathbf{tr}(\mathbf{WH}^\top\mathbf{Z}^\top\Theta)\Big) + \lambda_6||\Theta||_{2,1}\end{aligned} \tag{48}$$

where $f(.)$ is a non-lniear sigmoid function, i.e., $f(x) = \sigma(x) = \frac{1}{1+e^{-x}}$. This choice can be generalized to any non-linear functions. By transforming $\mathbf{X}$ with $\sigma(.)$ and Θ, our method enables pathway prediction. Taking the derivative of this error with respect to Θ to 0, we have:

$$\nabla_{\Theta}\mathcal{J}^{\mathbf{path}}(\Theta) = \frac{1}{n}\sum_{i\in n}\sum_{k\in t}\left(\frac{-\mathbf{y}_k^{(i)}\mathbf{x}^{(i)}}{1+e^{\mathbf{y}_k^{(i)}\Theta_k^\top\mathbf{x}^{(i)}}}\right) + 2\rho(\Theta - \mathbf{ZHW}^\top) + \lambda_6\mathbf{tr}(\frac{\Theta}{2||\Theta||_2}) \tag{49}$$

Due to non-closed form of the above equation, we use iterative gradient descent approach with a defined learning rate η. Hence, the general update rule for Θ becomes:

$$\Theta^{i+1} \leftarrow \Theta^i - \eta \circ \nabla_{\Theta}\mathcal{J}^{\mathbf{path}}(\Theta^i) \tag{50}$$

10. **Update the auxiliary matrix L.** By fixing the rest of parameters in $\mathcal{J}^{\mathbf{path}}$, the matrix $\mathbf{L}$ is updated such that the error is minimized:

$$\begin{aligned}\mathcal{J}^{\mathbf{path}}(\mathbf{L}) = &-\mathbf{tr}(\mathbf{X}^\top\mathbf{LRK}^\top) - \mathbf{tr}(\mathbf{KR}^\top\mathbf{L}^\top\mathbf{X}) + \mathbf{tr}(\mathbf{KR}^\top\mathbf{L}^\top\mathbf{LRK}^\top)\\ &-\mathbf{tr}(\mathbf{Y}^\top\mathbf{LTC}^\top) - \mathbf{tr}(\mathbf{CT}^\top\mathbf{L}^\top\mathbf{Y})\\ &+\mathbf{tr}(\mathbf{CT}^\top\mathbf{L}^\top\mathbf{LTC}^\top) + \lambda_6\mathbf{tr}(\mathbf{L}^\top\mathbf{L})\end{aligned} \tag{51}$$

Taking the derivative of this error with respect to $\mathbf{L}$ to 0, we have:

$$\nabla_{\mathbf{L}}\mathcal{J}^{\mathbf{path}}(\mathbf{L}) = 2(\mathbf{LTC}^\top\mathbf{CT}^\top + \mathbf{LRK}^\top\mathbf{KR}^\top - \mathbf{YCT}^\top - \mathbf{XKR}^\top + \lambda_6\mathbf{L}) \tag{52}$$

The parameter $\mathbf{L}$ is updated according:

$$\mathbf{L}^{i+1} \leftarrow \mathbf{L}^i - \eta \circ \nabla_{\mathbf{L}}\mathcal{J}^{\mathbf{path}}(\mathbf{L}^i) \tag{53}$$

11. **Update the auxiliary matrix Z.** By fixing the rest of parameters in $\mathcal{J}^{\mathbf{path}}$, the matrix $\mathbf{Z}$ is updated such that the error is minimized:

$$\begin{aligned}\mathcal{J}^{\mathbf{path}}(\mathbf{Z}) = &-\rho\mathbf{tr}(\Theta^\top\mathbf{ZHW}^\top) - \rho\mathbf{tr}(\mathbf{WH}^\top\mathbf{Z}^\top\Theta)\\ &+\rho\mathbf{tr}(\mathbf{WH}^\top\mathbf{Z}^\top\mathbf{ZHW}^\top) + \lambda_6\mathbf{tr}(\mathbf{Z}^\top\mathbf{Z})\end{aligned} \tag{54}$$

Taking the derivative of this error with respect to $\mathbf{Z}$ to 0, we have:

$$\nabla_{\mathbf{Z}}\mathcal{J}^{\mathbf{path}}(\mathbf{Z}) = 2(\rho\mathbf{ZHW}^\top\mathbf{WH}^\top - \rho\Theta\mathbf{WH}^\top + \lambda_6\mathbf{Z}) \tag{55}$$

The parameter $\mathbf{Z}$ is updated according to gradient descent approach as:

$$\mathbf{Z}^{i+1} \leftarrow \mathbf{Z}^i - \eta \circ \nabla_{\mathbf{Z}}\mathcal{J}^{\mathbf{path}}(\mathbf{Z}^i) \tag{56}$$

5.4 Appendix A4: Experimental Setup

In this section, we describe the experimental framework used to demonstrate triUMPF pathway prediction performance across multiple datasets spanning the genomic information hierarchy [25]. All experimental tests were conducted on a Linux server using 10 cores of Intel Xeon CPU E5-2650.

5.4.1 Association Matrices

MetaCyc v21 ([4]) was used to obtain the three association matrices, P2E ($\mathbf{M}$), P2P, ($\mathbf{A}$), and E2E ($\mathbf{B}$). Some of the properties for each matrix are summarized in Table 3. All three matrices are extremely sparse. For example, $\mathbf{M}$ contains 2526 pathways, having an average of four EC associations per pathway, leaving more than 3600 columns with zero values. These matrices will be utilized to obtain higher-order proximity (Section 5.5.1) and to analyze triUMPF's robustness (Section 5.5.2).

Table 3: Characteristics of MetaCyc database and the three association matrices. MetaCyc (uec) denotes enzymatic reactions where links among enzymatic reactions are removed. The "–" indicates non applicable operation.

	#EC	#Compound	#Pathway	$\|\mathcal{V}\|$	$\|\mathcal{E}\|$
MetaCyc (uec)	6378	13689	2526	22593	33353
M	3650	–	2526	–	8576
A	–	–	2526	–	9938
B	3650	–	–	–	35629

5.4.2 Description of Datasets

We report the performance of triUMPF using the following data: i)- T1 golden consisting of six PGDBs from the BioCyc collection (biocyc): *EcoCyc (v21)*, *HumanCyc (v19.5)*, *AraCyc (v18.5)*, *YeastCyc (v19.5)*, *LeishCyc (v19.5)*, and *TrypanoCyc (v18.5)*; ii)- three *E.coli* genomes consisting of E. coli K-12 substr. MG1655 (TAX-511145), E. coli str. CFT073 (TAX-199310), and E. coli O157:H7 str. EDL933 (TAX-155864) [36]; iii)- BioCyc (v20.5 T2 & 3) [5] consisting of 9255 Pathway/Genome Databases (PGDBs) with 1463 distinct pathways; iv)- reduced complexity of mealybug symbiont genomes from *Moranella* (GenBank NC-015735) and *Tremblaya* (GenBank NC-015736) encoding distributed metabolic pathways for amino acid biosynthesis [26]; v)- the Critical Assessment of Metagenome Interpretation (CAMI) initiative low complexity dataset (edwards.sdsu.edu/research/cami-challenge-datasets/), consisting of 40 genomes [31], and vi)- whole genome shotgun sequences from the Hawaii Ocean Time Series (HOTS) at 25m, 75m, 110m (sunlit) and 500m (dark) ocean depth intervals downloaded from the NCBI Sequence Read Archive under accession numbers SRX007372, SRX007369, SRX007370, SRX007371 [33]. T1 PGDBs were refined to include only those pathways that cross-intersect with the *MetaCyc* database (v21) [4]. The detailed characteristics of the datasets are summarized in Table 4. For each dataset $\mathcal{S}$, we use $|\mathcal{S}|$ and L($\mathcal{S}$) to represent the number of instances and pathway labels, respectively. In addition, we also present some characteristics of the multi-label datasets, which are denoted as:

1. Label cardinality (LCard($\mathcal{S}$) $= \frac{1}{n}\sum_{i=1}^{i=n}\sum_{j=1}^{j=t}\mathbb{I}[\mathbf{Y}_{i,j} \neq -1]$), where $\mathbb{I}$ is an indicator function. It denotes the average number of pathways in $\mathcal{S}$.
2. Label density (LDen($\mathcal{S}$) $= \frac{LCard(\mathcal{S})}{L(\mathcal{S})}$). This is simply obtained through normalizing LCard($\mathcal{S}$) by the number of total pathways in $\mathcal{S}$.
3. Distinct labels (DL($\mathcal{S}$)). This notation indicates the number of distinct pathways in $\mathcal{S}$.
4. Proportion of distinct labels (PDL($\mathcal{S}$) $= \frac{DL(\mathcal{S})}{|\mathcal{S}|}$). It represents the normalized version of DL($\mathcal{S}$), and is obtained by dividing DL(.) with the number of instances in $\mathcal{S}$.

The notations R($\mathcal{S}$), RCard($\mathcal{S}$), RDen($\mathcal{S}$), DR($\mathcal{S}$), and PDR($\mathcal{S}$) have similar meanings for the enzymatic reactions $\mathcal{E}$ in $\mathcal{S}$. Finally, PLR($\mathcal{S}$) represents a ratio of L($\mathcal{S}$) to R($\mathcal{S}$).

5.4.3 Pathway and Enzymatic Reaction Features

triUMPF was trained using BioCyc v20.5 which contains less than 1460 trainable pathways. To offset this limit, we applied pathway2vec [24] using RUST-norm (or "crt") module to obtain pathway and EC features, indicated by **P** and **E**, respectively, with the following settings: the number of memorized domain is 3, the explore and the in-out hyperparameters are 0.55 and 0.84, respectively, the number of sampled path instances was 100, the walk length is 100, the embedding dimension size was $m = 128$, the neighborhood size was 5, the size of negative samples was 5, and the used configuration of MetaCyc was "uec", indicating links among ECs are being trimmed.

After generating node features, we only apply EC features to concatenate each example i according to:

$$\tilde{\mathbf{x}}^{(i)} = \mathbf{x}^{(i)} \oplus \frac{1}{r}\mathbf{x}^{(i)}\mathbf{E} \tag{57}$$

Table 4: Experimental data set properties. The notations $|\mathcal{S}|$, L($\mathcal{S}$), LCard($\mathcal{S}$), LDen($\mathcal{S}$), DL($\mathcal{S}$), and PDL($\mathcal{S}$) represent: number of instances, number of pathway labels, pathway labels cardinality, pathway labels density, distinct pathway labels, and proportion of distinct pathway labels for $\mathcal{S}$, respectively. The notations R($\mathcal{S}$), RCard($\mathcal{S}$), RDen($\mathcal{S}$), DR($\mathcal{S}$), and PDR($\mathcal{S}$) have similar meanings for the enzymatic reactions $\mathcal{E}$ in $\mathcal{S}$. PLR($\mathcal{S}$) represents a ratio of L($\mathcal{S}$) to R($\mathcal{S}$). The last column denotes the domain of $\mathcal{S}$.

Dataset	$\lvert\mathcal{S}\rvert$	L($\mathcal{S}$)	LCard($\mathcal{S}$)	LDen($\mathcal{S}$)	DL($\mathcal{S}$)	PDL($\mathcal{S}$)	R($\mathcal{S}$)	RCard($\mathcal{S}$)	RDen($\mathcal{S}$)	DR($\mathcal{S}$)	PDR($\mathcal{S}$)	PLR($\mathcal{S}$)	Domain
AraCyc	1	510	510	1	510	510	2182	2182	1	1034	1034	0.2337	Arabidopsis thaliana
EcoCyc	1	307	307	1	307	307	1134	1134	1	719	719	0.2707	Escherichia coli K-12 substr.MG1655
HumanCyc	1	279	279	1	279	279	1177	1177	1	693	693	0.2370	Homo sapiens
LeishCyc	1	87	87	1	87	87	363	363	1	292	292	0.2397	Leishmania major Friedlin
TrypanoCyc	1	175	175	1	175	175	743	743	1	512	512	0.2355	Trypanosoma brucei
YeastCyc	1	229	229	1	229	229	966	966	1	544	544	0.2371	Saccharomyces cerevisiae
Three E.coli	3	–	–	–	–	–	2353	784.3333	0.3333	634	211.3333	–	E. coli K-12 substr. MG1655 (TAX-511145), E. coli str. CFT073 (TAX-199310), and E. coli O157:H7 str. EDL933 (TAX-155864)
BioCyc	9255	1804003	194.9220	0.0001	1463	0.1581	8848714	956.1009	0.0001	2705	0.2923	0.2039	BioCyc version 20.5 (tier 2 & 3)
Symbiont	3	–	–	–	–	–	304	101.3333	0.3333	130	43.3333	–	Composed of Moranella and Tremblaya
CAMI	40	6261	156.5250	0.0250	674	16.8500	14269	356.7250	0.0250	1083	27.0750	0.4388	Simulated microbiomes of low complexity
HOTS	4	–	–	–	–	–	182675	26096.4286	0.1429	1442	206.0000	–	Metagenomic Hawaii Ocean Time-series (10m, 75m, 110m, and 500m)

where $\oplus$ indicates the vector concatenation operation, $\mathbf{E} \in \mathbb{R}^{r \times m}$ corresponds the feature matrix of ECs and $m = 128$. The addition of features results in a dimension of size $r + m$, where $r = 3650$. We expect by incorporating enzymatic reactions features into the original r dimensional example $\mathbf{x}^{(i)}$, the modified $\tilde{\mathbf{x}}^{(i)}$ summarizes informative characteristics, which are expected to be useful in the prediction task.

5.4.4 Parameter Settings

For training, unless otherwise indicated, the learning rate was set to 0.0001, batch size to 50, number of epochs to 10, number of components $k = 100$, number of pathway and EC communities to $p = 90$ and $v = 100$, respectively. The higher-order proximity for $\mathbf{A}^{\mathbf{prox}}$ and $\mathbf{B}^{\mathbf{prox}}$ (corresponding P2P and E2E matrices, respectively, in Section 5.4.1) were set to $l^p = 3$ and $l^e = 1$ and their associated weights fixed as $\omega = 0.1$ and $\gamma = 0.3$, respectively. The α and β were fixed to 10^9. For the regularized hyperparameters λ_*, we performed 10-fold cross-validation on MetaCyc and a subsample of BioCyc T2 &3 data and found the settings $\lambda_{1:5} = 0.01$, $\lambda_6 = 10$, and $\rho = 0.001$ to be optimum on golden T1 data.

5.5 Appendix A5: Experimental Results

Four tests were performed to benchmark the performance of triUMPF including parameter sensitivity, network reconstruction, impact of ρ, and metabolic pathway prediction.

5.5.1 Parameter Sensitivity

The impact of seven hyperparameters (k, p, v, l_p, l_e, ω and γ) was evaluated in relation to matrix reconstruction costs for ($\mathbf{M}$, $\mathbf{A}^{\mathbf{prox}}$, and $\mathbf{B}^{\mathbf{prox}}$). The reconstruction cost (or error) defines the

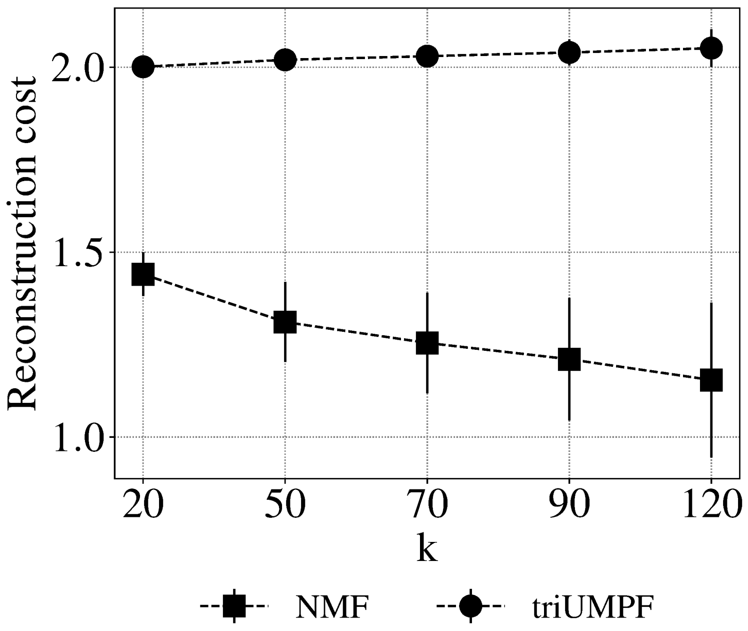


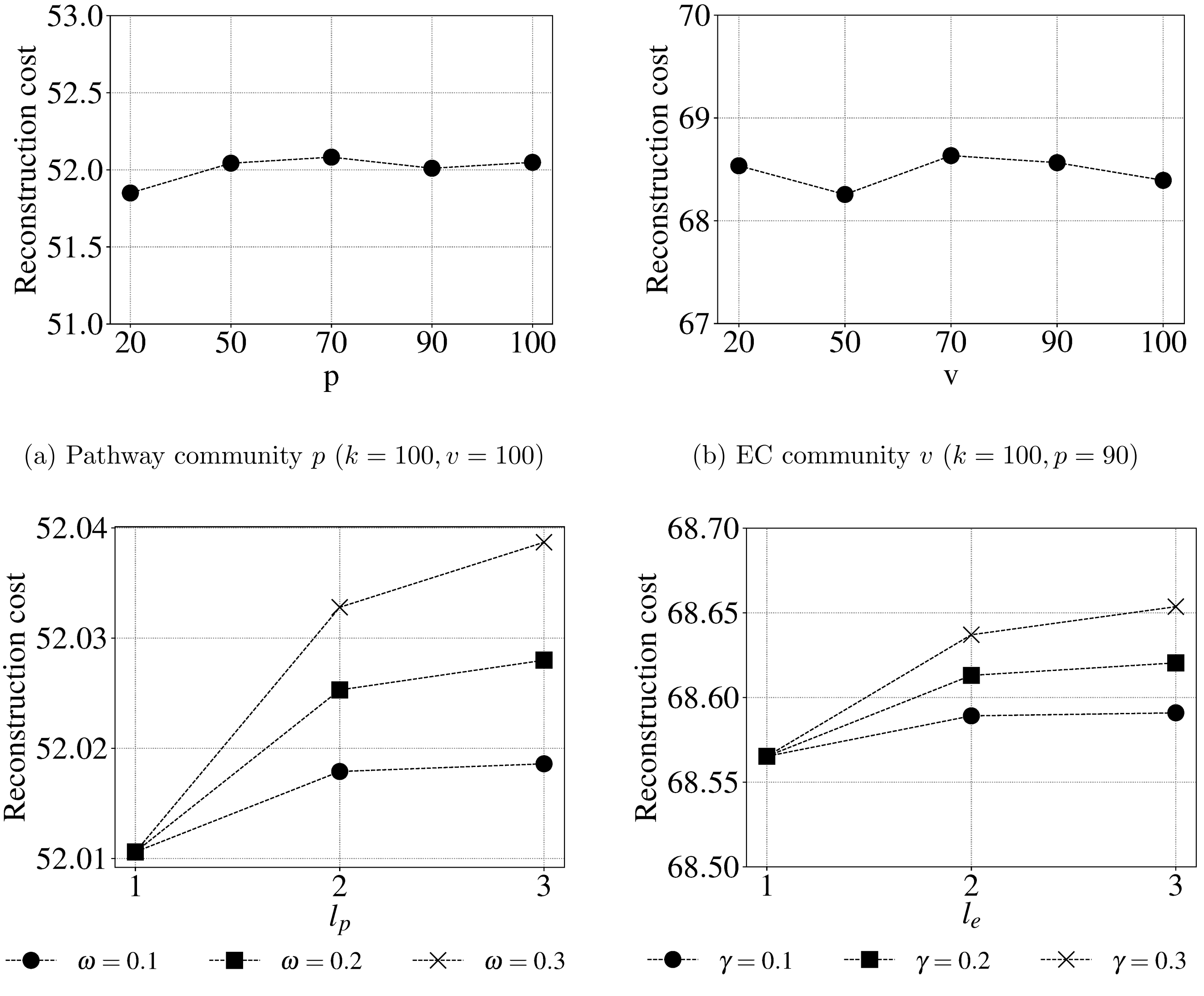
Figure 4: Sensitivity of components k based on reconstruction cost.

sum of mean squared errors accounted in the process of transforming the decomposed matrices into its original form where lower cost entails the decomposed low dimensional matrices were able to better capture the representations of the original matrix. We specifically evaluated the effects of varying the following parameters: i)- the number of components $k \in \{20, 50, 70, 90, 120\}$, ii)- the community size of pathway $p \in \{20, 50, 70, 90, 100\}$ and EC $v \in \{20, 50, 70, 90, 100\}$, iii)- the higher-order proximity l_p and $l_e \in \{1, 2, 3\}$, and iv)- weights of the polynomial order ω and $\gamma \in \{0.1, 0.2, 0.3\}$. We used the full matrix $\mathbf{M}$, for each test, however, for community detection, we used BioCyc T2 &3 data that is divided into training (80%), validation (5%) and test sets (15%). The final costs for community detection are reported based on the test set after 10 successive trials. In addition, we contrast triUMPF with the standard NMF for monitoring the reconstruction costs of $\mathbf{M}$ by varying k values. We emphasize that $\mathbf{M}$, $\mathbf{A}^{\mathbf{prox}}$, and $\mathbf{B}^{\mathbf{prox}}$ were collected from MetaCyc (Section 5.4.1) and not from BioCyc T2 &3 (Section 5.4.2).

Fig. 4 shows the effect of rank k on triUMPF performance. In general, we observe steady performance with increasing k. Although this contrasts standard NMF, where reconstruction cost decreases as the number of features increases it is expected because, unlike standard NMF, triUMPF exploits two types of correlations to recover $\mathbf{M}$: i)- within ECs or pathways and ii)- betweenness interactions that serve as additional regularizers. As observed in Fig. 4, higher k values result in improved outcomes. Consequently, we selected $k = 100$ for downstream testing.

For community detection, we observed optimal results with respect to pathway community size at $p = 20$ under parameter settings $k = 100$ and $v = 100$, as shown in Fig. 5a. However, because $\mathbf{A}^{\mathbf{prox}}$ is so sparse, we suggest that this low rank may not correspond to the optimum community size. As with all methods of community detection triUMPF is sensitive to community size and requires empirical testing. Therefore, we tested settings between $p = 20$ and $p = 100$ and observed a decrease in performance under parameter settings $k = 100$ and $v = 100$ with $p = 90$ providing a balance between cost and increased community size. A similar result was observed for EC community size at $v = 100$ under parameter settings $p = 90$ and $k = 100$ in Fig. 5b.

Finally, we show the effect of changing polynomial orders, and their weights on triUMPF performance. From Fig. 5c, we see that reconstruction cost progressively increases with varying higher orders for l_p for all the three weights ω. However, for the same reasons described above, we prefer more long distances with less weight to preserve community structure, and remarkably, when $\omega = 0.1$ triUMPF performance was relatively stable after the second order. The same



(a) Pathway community p $(k = 100, v = 100)$

(b) EC community v $(k = 100, p = 90)$

(c) Effect of l_p

(d) Effect of l_e

Figure 5: Sensitivity of community size and higher order proximity with weights based on reconstruction cost.

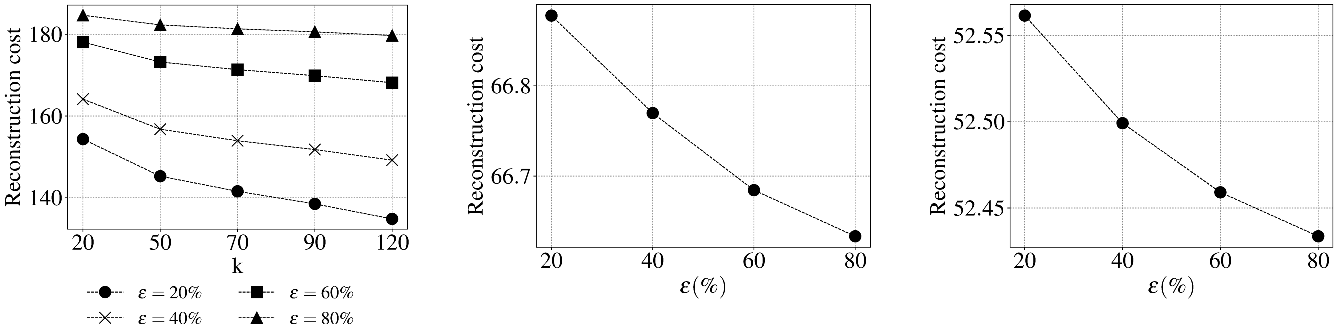
conclusion can be drawn for l_e and its associated weights γ in Fig. 5d.

Based on these results, triUMPF performance is stable while minimizing cost under the following parameter settings: $k = 100$, $p > 90$, $e > 90$, $l_p = 3$, $\omega = 0.1$, $l_e = 1$, and $\gamma = 0.3$. Therefor, we recommend these settings for both MetaCyc and BioCyc T2 &3.

5.5.2 Network Reconstruction

In this section, we explore the robustness of triUMPF when exposed to noise. Links were randomly removed from **M**, **A**, and **B** according to $\varepsilon \in \{20\%, 40\%, 60\%, 80\%\}$. We used the partially linked matrices to refine parameters while comparing the reconstruction cost against the full association matrices **M**, **A** and **B**. Specifically for **M**, we varied components of **M** according to $k \in \{20, 50, 70, 90, 120\}$ along with ϵ. For all experiments, both MetaCyc and BioCyc T2 &3 were applied for training using hyperparameters described in Section 3.4 of the primary text.

Fig. 6a indicate that by progressively increasing noise ε to **M**, the reconstruction cost increases when k is low. As more features are incorporated the cost at all noise levels steadily decreases up to $k = 100$. This tendency indicates that both pathway and EC features (**P** and **E** contain useful correlations that contribute to the resilience of triUMPF's performance when



(b) EC links recovery

(c) Pathway links recovery

(a) Effect of k

Figure 6: Link prediction results by varying noise levels $\varepsilon \in \{20\%, 40\%, 60\%, 80\%\}$ based on reconstruction cost.

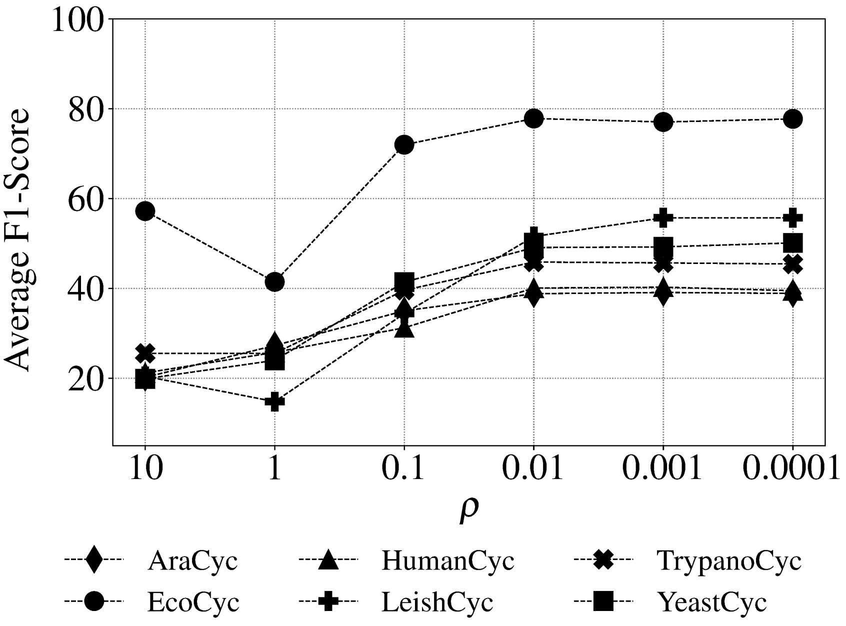


Figure 7: Effect of ρ based on average F1 score using golden datasets.

$\mathbf{M}$ is perturbed.

For $\mathbf{A}^{\mathbf{prox}}$ and $\mathbf{B}^{\mathbf{prox}}$, as shown in Figs 6b and 6b, the costs are reduced in the presence of noise, which is not surprising as the reconstruction of associated communities are constrained on both data and $\mathbf{A}^{\mathbf{prox}}$ and $\mathbf{B}^{\mathbf{prox}}$. These results are directly linked to the sparseness of both matrices, as previously described in [8]. The pathway graph network, depicted in Fig. 1 of the primary text, indicates that many pathways constitute islands with no direct links, while some pathways are densely connected. For community detection, it is sufficient to group nodes that are densely connected, while links between communities can remain sparse. The same line of reasoning follows for the EC network.

5.5.3 Impact of ρ

Fig. 7 shows the inverse effect in predictive performance on T1 golden datasets when decreasing ρ before reaching a performance plateau at $\rho = 0.001$. The hyperparameter ρ in Eq. 5 controls the amount of information propagation from $\mathbf{M}$ to pathway label coefficients Θ. This suggests, in practice, lesser constraints should be emphasized on Θ, while not neglecting associations between EC numbers and pathways indicated in $\mathbf{M}$.

Table 5: Predictive performance of each comparing algorithm on 6 golden T1 data. For each performance metric, '↓' indicates the smaller score is better while '↑' indicates the higher score is better.

Methods	Hamming Loss ↓					
	EcoCyc	HumanCyc	AraCyc	YeastCyc	LeishCyc	TrypanoCyc
PathoLogic	0.0610	**0.0633**	0.1188	**0.0424**	**0.0368**	**0.0424**
MinPath	0.2257	0.2530	0.3266	0.2482	0.1615	0.2561
mlLGPR	0.0804	**0.0633**	**0.1069**	0.0550	0.0380	0.0590
triUMPF	**0.0435**	0.0954	0.1560	0.0649	0.0443	0.0776

Methods	Average Precision ↑					
	EcoCyc	HumanCyc	AraCyc	YeastCyc	LeishCyc	TrypanoCyc
PathoLogic	0.7230	**0.6695**	0.7011	0.7194	**0.4803**	**0.5480**
MinPath	0.3490	0.3004	0.3806	0.2675	0.1758	0.2129
mlLGPR	0.6187	0.6686	0.7372	0.6480	0.4731	0.5455
triUMPF	**0.8662**	0.6080	**0.7377**	**0.7273**	0.4161	0.4561

Methods	Average Recall ↑					
	EcoCyc	HumanCyc	AraCyc	YeastCyc	LeishCyc	TrypanoCyc
PathoLogic	0.8078	0.8423	0.7176	0.8734	0.8391	0.7829
MinPath	**0.9902**	**0.9713**	**0.9843**	**1.0000**	**1.0000**	**1.0000**
mlLGPR	0.8827	0.8459	0.7314	0.8603	0.9080	0.8914
triUMPF	0.7590	0.3835	0.3529	0.3319	0.7126	0.6229

Methods	Average F1 ↑					
	EcoCyc	HumanCyc	AraCyc	YeastCyc	LeishCyc	TrypanoCyc
PathoLogic	0.7631	0.7460	0.7093	**0.7890**	0.6109	0.6447
MinPath	0.5161	0.4589	0.5489	0.4221	0.2990	0.3511
mlLGPR	0.7275	**0.7468**	**0.7343**	0.7392	**0.6220**	**0.6768**
triUMPF	**0.8090**	0.4703	0.4775	0.4735	0.5254	0.5266

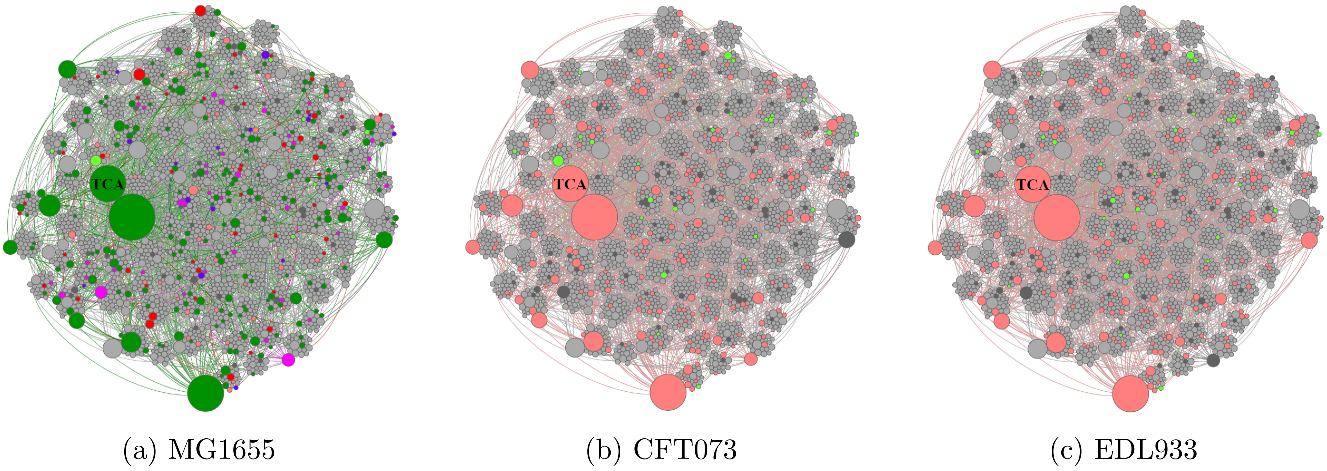
5.5.4 Metabolic Pathway Prediction

Here, we investigate the effectiveness of triUMPF for the pathway prediction task on i)- T1 golden data, ii)- three *E. coli* data, and iii)- HOTS.

T1 Golden Data. We compare the performance of triUMPF on 6 benchmark datasets, as described in Section 5.4.2, against the other pathway prediction algorithms using four evaluation metrics: *Hamming loss*, *average precision*, *average recall*, and *average F1 score*. As shown in Table 5, triUMPF achieved competitive performance against the other methods in terms of average precision.

Three E.coli Data. Fig. 8 shows pathway communities observed for MG1655, CFT073 and EDL933 using BioCyc T2 &3 including MetaCyc in training. Fig. 9 shows that PathoLogic was able to infer over 90 additional pathways when taxonomic pruning is disabled. Table 7 summarizes GapMind [29] results for MG1655, CFT073 and EDL933. Fig. 10 shows the results for both PathoLogic with taxonomic pruning enabled and triUMPF. Without taxonomic pruning, PathoLogic predicted 56 pathways across the three strains encompassing 15 amino acid biosynthesis pathways and 20 pathway variants, including *L-proline biosynthesis II (from arginine)* pathway that is known only for eukaryotes (Fig. 11), consequently, increasing false-positive pathway prediction.

HOTS water column. Here, we use triUMPF to infer a set of pathways from the HOTS water column spanning sunlit and dark ocean depth intervals comparing results to other prediction methods including PathoLogic and mlLGPR. The results are presented in Fig. 12.



(a) MG1655 (b) CFT073 (c) EDL933

Figure 8: Pathway community networks for related T1 and T3 organismal genomes. Pathway communities for (a) E. coli K-12 substr. MG1655 (TAX-511145), (b) E. coli str. CFT073 (TAX-199310), and (c) E. coli O157:H7 str. EDL933 (TAX-155864) based on community detection. Nodes colored in *dark grey* indicate pathways predicted by PathoLogic; *lime* pathways predicted by triUMPF; *salmon* pathways predicted by both PathoLogic and triUMPF; *red* expected pathways not predicted by both PathoLogic and triUMPF; *magenta* expected pathways predicted only by PathoLogic; *purple* expected pathways predicted solely by triUMPF; and *green* expected pathways predicted by both PathoLogic and triUMPF. *light-grey* indicates pathways not expected to be encoded in either organismal genome. The node sizes reflect the degree of associations between pathways.

Availability of Data and Materials

The triUMPF source code is available under the MIT License on GitHub (hallamlab/triUMPF) with detailed descriptions on how to install and execute all commands run to generate results in our GitHub repository. The MetaCyc database can be obtained from metacyc.org. The T1 golden datasets can be downloaded from biocyc.org. For the symbiotic *Candidatus Moranella endobia* and *Candidatus Tremblaya princeps* genomes, they can be downloaded from GenBank under accession numbers NC-015735 and NC-015736 while the simulated CAMI low complexity dataset can be obtained from edwards.sdsu.edu/research/cami-challenge-datasets. Unassembled whole genome shotgun DNA pyrosequences from HOTS (10m, 75m, 110m, and 500m) can be obtained from the NCBI Sequence Read Archive under accession numbers SRX007372, SRX007369, SRX007370, SRX007371. The preprocessed datasets used in this paper can be downloaded from zenodo.org/YLIBG_lKhPZ. The same zenodo repo contains a pre-trained triUMPF (triUMPF_Xe.pkl) using configurations stated in Section 5.4. We also included the three preprocessed E.coli data in the github repo under the "sample" directory.

Acknowledgments

We would like to thank Connor Morgan-Lang, Julia Anstett, Kishori Konwar and Aria Hahn for lucid discussions on the function of the triUMPF model and all members of the Hallam Lab for helpful comments along the way.

Author Disclosure Statement

SJH is a co-founder of Koonkie Inc., a bioinformatics consulting company that designs and provides scalable algorithmic and data analytics solutions in the cloud.

Table 6: Top 5 communities with pathways predicted by triUMPF for E. coli K-12 substr. MG1655 (TAX-511145). The last column asserts whether a pathway is present in or absent (a false-positive pathway) from EcoCyc reference data.

Community Index	MetaCyc Pathway ID	MetaCyc Pathway Name	Status
67	PWY0-1182	trehalose degradation II (trehalase)	true
	PWY-6910	hydroxymethylpyrimidine salvage	true
	HOMOSER-THRESYN-PWY	L-threonine biosynthesis	true
	PUTDEG-PWY	putrescine degradation I	true
	PWY-6611	adenine and adenosine salvage V	true
	FERMENTATION-PWY	mixed acid fermentation	true
	ENTNER-DOUDOROFF-PWY	Entner-Doudoroff pathway I	true
34	ASPARAGINESYN-PWY	L-asparagine biosynthesis II	true
	PWY-5340	sulfate activation for sulfonation	true
	PWY-6618	guanine and guanosine salvage III	true
	PWY0-1314	fructose degradation	true
	PWY-7181	pyrimidine deoxyribonucleosides degradation	true
	PWY0-1299	arginine dependent acid resistance	true
	PWY0-42	2-methylcitrate cycle I	true
9	NAGLIPASYN-PWY	lipid-A-precursor biosynthesis (E. coli)	true
	PWY-7221	guanosine ribonucleotides de novo biosynthesis	true
	KDOSYN-PWY	Kdo transfer to lipid IV$_A$ I (E. coli)	true
	PWY0-1309	chitobiose degradation	true
	PPGPPMET-PWY	ppGpp biosynthesis	true
	PWY-6608	guanosine nucleotides degradation III	true
	PWY-5656	mannosylglycerate biosynthesis I	false
47	PLPSAL-PWY	pyridoxal 5'-phosphate salvage I	true
	PWY0-1313	acetate conversion to acetyl-CoA	true
	PYRUVDEHYD-PWY	pyruvate decarboxylation to acetyl CoA	true
	PWY-4381	fatty acid biosynthesis initiation (bacteria and plants)	true
	PWY0-662	PRPP biosynthesis	true
81	HISTSYN-PWY	L-histidine biosynthesis	true
	PWY-6147	6-hydroxymethyl-dihydropterin diphosphate biosynthesis I	true
	PWY-7176	UTP and CTP de novo biosynthesis	true
	PWY-6932	selenate reduction	false

Funding Information

This work was performed under the auspices of Genome Canada, Genome British Columbia, the Natural Science and Engineering Research Council (NSERC) of Canada, and Compute/Calcul Canada). ARMAB and RJM were supported by a UBC four-year doctoral fellowship (4YF) administered through the UBC Graduate Program in Bioinformatics.

References

[1] Amos Bairoch. The enzyme database in 2000. *Nucleic acids research*, 28(1):304–305, 2000.

[2] Rasmus Bro and Age K Smilde. Principal component analysis. *Analytical Methods*, 6(9):2812–2831, 2014.

[3] Pablo Carbonell, Jerry Wong, Neil Swainston, Eriko Takano, Nicholas J Turner, Nigel S Scrutton, Douglas B Kell, Rainer Breitling, and Jean-Loup Faulon. Selenzyme: Enzyme selection tool for pathway design. *Bioinformatics*, 34(12):2153–2154, 2018.

[4] Ron Caspi, Richard Billington, Luciana Ferrer, Hartmut Foerster, Carol A. Fulcher, Ingrid M. Keseler, Anamika Kothari, Markus Krummenacker, Mario Latendresse, Lukas A. Mueller, Quang Ong, Suzanne Paley, Pallavi Subhraveti, Daniel S. Weaver, and Peter D. Karp. The metacyc database of metabolic pathways and enzymes and the biocyc collection of pathway/genome databases. *Nucleic Acids Research*, 44(D1):D471–D480, 2016.

[5] Ron Caspi, Richard Billington, Hartmut Foerster, Carol A Fulcher, Ingrid Keseler, Anamika Kothari, Markus Krummenacker, Mario Latendresse, Lukas A Mueller, Quang Ong, et al. Biocyc: Online resource for genome and metabolic pathway analysis. *The FASEB Journal*, 30(1 Supplement):lb192–lb192, 2016.

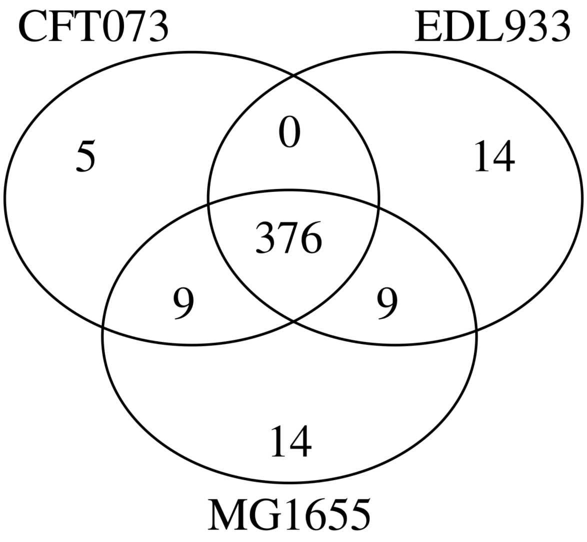


Figure 9: A three way set analysis of predicted pathways for E. coli K-12 substr. MG1655 (TAX-511145), E. coli str. CFT073 (TAX-199310), and E. coli O157:H7 str. EDL933 (TAX-155864) using PathoLogic (without taxonomic pruning).

[6] Andrzej Cichocki, Rafal Zdunek, and Shun-ichi Amari. Hierarchical als algorithms for non-negative matrix and 3d tensor factorization. In *International Conference on Independent Component Analysis and Signal Separation*, pp. 169–176. Springer, 2007.

[7] Joseph M Dale, Liviu Popescu, and Peter D Karp. Machine learning methods for metabolic pathway prediction. *BMC bioinformatics*, 11(1):1, 2010.

[8] Santo Fortunato and Darko Hric. Community detection in networks: A user guide. *Physics reports*, 659:1–44, 2016.

[9] Xiao Fu, Kejun Huang, Nicholas D. Sidiropoulos, and Wing-Kin Ma. Nonnegative matrix factorization for signal and data analytics: Identifiability, algorithms, and applications. *IEEE Signal Processing Magazine*, 36(2):59–80, 2019.

[10] Nicolas Gillis. The why and how of nonnegative matrix factorization. *Regularization, Optimization, Kernels, and Support Vector Machines*, 12(257), 2014.

[11] Aria S Hahn, Kishori M Konwar, Stilianos Louca, Niels W Hanson, and Steven J Hallam. The information science of microbial ecology. *Current opinion in microbiology*, 31:209–216, 2016.

[12] Niels W Hanson, Kishori M Konwar, Alyse K Hawley, Tomer Altman, Peter D Karp, and Steven J Hallam. Metabolic pathways for the whole community. *BMC genomics*, 15(1):1, 2014.

[13] Haibo He and Edwardo A Garcia. Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, 21(9):1263–1284, 2009.

[14] Dazhi Jiao, Yuzhen Ye, and Haixu Tang. Probabilistic inference of biochemical reactions in microbial communities from metagenomic sequences. *PLoS Comput Biol*, 9(3):e1002981, 2013.

[15] Minoru Kanehisa, Miho Furumichi, Mao Tanabe, Yoko Sato, and Kanae Morishima. Kegg: new perspectives on genomes, pathways, diseases and drugs. *Nucleic Acids Research*, 45(D1):D353–D361, 2017.

[16] Peter D Karp, Mario Latendresse, Suzanne M Paley, Markus Krummenacker, Quang D Ong, Richard Billington, Anamika Kothari, Daniel Weaver, Thomas Lee, Pallavi Subhraveti, et al. Pathway tools version 19.0 update: software for pathway/genome informatics and systems biology. *Briefings in bioinformatics*, 17(5):877–890, 2016.

Table 7: 18 amino acid biosynthesis pathways and 27 pathway variants.

Amino Acid	MetaCyc Pathway ID	MetaCyc Pathway Name
Arginine	ARGSYNBSUB-PWY PWY-5154 PWY-7400	L-arginine biosynthesis II (acetyl cycle) L-arginine biosynthesis III (via N-acetyl-L-citrulline) L-arginine biosynthesis IV (archaebacteria)
Asparagine	ASPARAGINE-BIOSYNTHESIS ASPARAGINESYN-PWY	L-asparagine biosynthesis I L-asparagine biosynthesis II
Chorismate	PWY-6163	chorismate biosynthesis from 3-dehydroquinate
Cysteine	CYSTSYN-PWY PWY-6308	L-cysteine biosynthesis I L-cysteine biosynthesis II (tRNA-dependent)
Glutamine	GLNSYN-PWY	L-glutamine biosynthesis I
Glycine	GLYSYN-PWY GLYSYN-THR-PWY	glycine biosynthesis I glycine biosynthesis IV
Histidine	HISTSYN-PWY	L-histidine biosynthesis
Isoleucine	ILEUSYN-PWY PWY-5104	L-isoleucine biosynthesis I (from threonine) L-isoleucine biosynthesis IV
Leucine	LEUSYN-PWY	L-leucine biosynthesis
Lysine	DAPLYSINESYN-PWY PWY-2941 PWY-2942	L-lysine biosynthesis I L-lysine biosynthesis II L-lysine biosynthesis III
Methionine	HOMOSER-METSYN-PWY PWY-702	L-methionine biosynthesis I L-methionine biosynthesis II
Phenylalanine	PHESYN	L-phenylalanine biosynthesis I
Proline	PROSYN-PWY	L-proline biosynthesis I
Serine	SERSYN-PWY	L-serine biosynthesis
Threonine	HOMOSER-THRESYN-PWY	L-threonine biosynthesis
Tryptophan	TRPSYN-PWY	L-tryptophan biosynthesis
Tyrosine	TYRSYN	L-tyrosine biosynthesis I
Valine	VALSYN-PWY	L-valine biosynthesis

[17] Hyunsoo Kim and Haesun Park. Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis. *Bioinformatics*, 23(12):1495–1502, 2007.

[18] Kishori M Konwar, Niels W Hanson, Antoine P Pagé, and Steven J Hallam. Metapathways: a modular pipeline for constructing pathway/genome databases from environmental sequence information. *BMC bioinformatics*, 14(1):202, 2013.

[19] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, pp. 6402–6413, 2017.

[20] Christopher E Lawson, William R Harcombe, Roland Hatzenpichler, Stephen R Lindemann, Frank E Löffler, Michelle A O'Malley, Héctor García Martín, Brian F Pfleger, Lutgarde Raskin, Ophelia S Venturelli, et al. Common principles and best practices for engineering microbiomes. *Nature Reviews Microbiology*, pp. 1–17, 2019.

[21] Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788, 1999.

[22] Daniel D Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*, pp. 556–562, 2001.

[23] Yu Li, Ying Wang, Tingting Zhang, Jiawei Zhang, and Yi Chang. Learning network embedding with community structural information. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pp. 2937–2943. International Joint Conferences on Artificial Intelligence Organization, 7 2019.

[24] Abdur Rahman M. A. Basher and Steven J. Hallam. Leveraging heterogeneous network embedding for metabolic pathway prediction. *Bioinformatics*, 10 2020.

[25] Abdur Rahman M. A. Basher, Ryan J. McLaughlin, and Steven J. Hallam. Metabolic pathway inference using multi-label classification with rich pathway features. *PLOS Computational Biology*, 16(10):1–22, 10 2020.

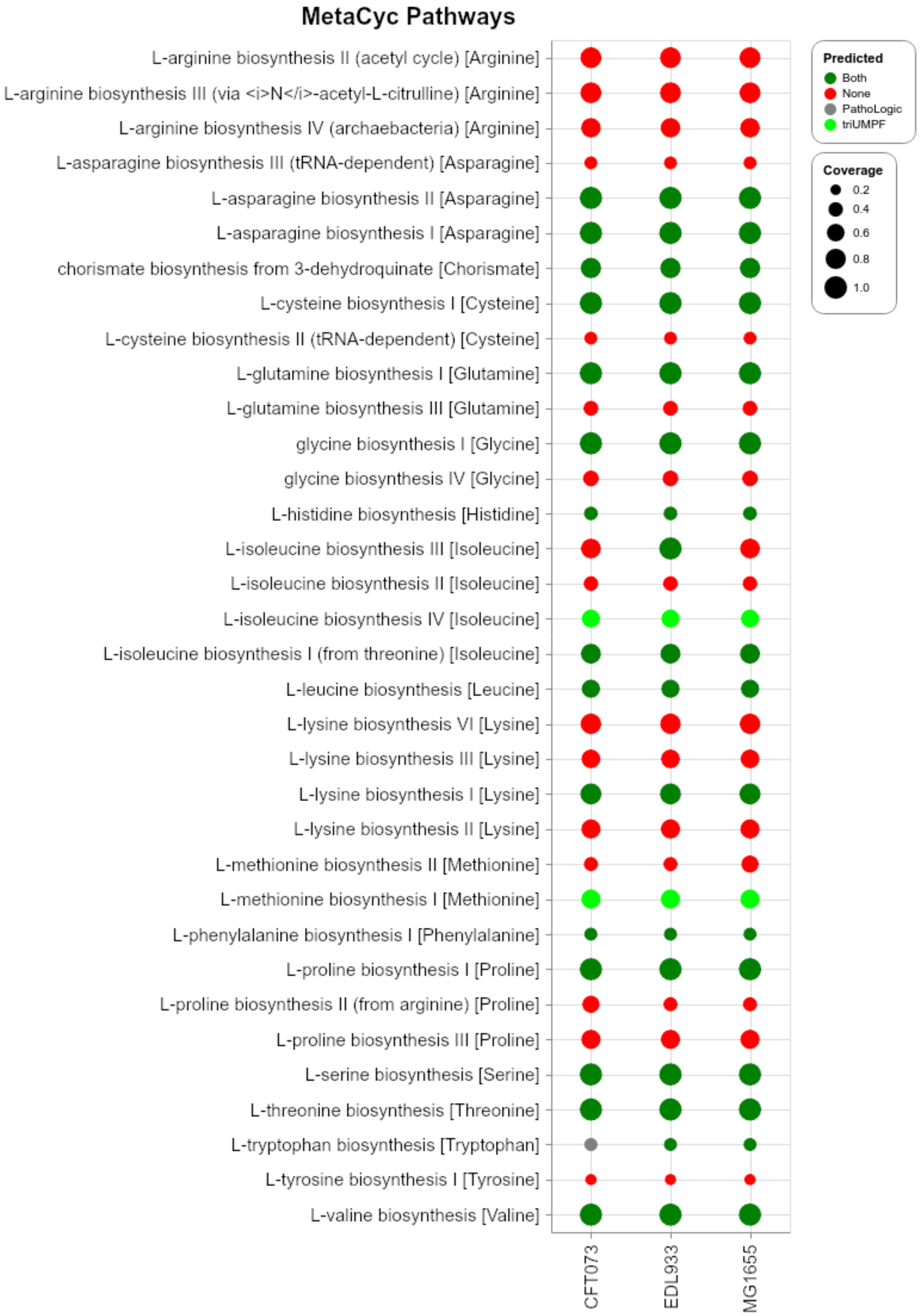


Figure 10: Comparison of predicted pathways for E. coli K-12 substr. MG1655 (TAX-511145), E. coli str. CFT073 (TAX-199310), and E. coli O157:H7 str. EDL933 (TAX-155864) datasets between PathoLogic (taxonomic pruning) and triUMPF. Red circles indicate that neither method predicted a specific pathway while green circles indicate that both methods predicted a specific pathway. Lime circles indicate pathways predicted solely by mlLGPR and gray circles indicate pathways solely predicted by PathoLogic.The size of circles corresponds to associated pathway coverage information.

[26] John P McCutcheon and Carol D Von Dohlen. An interdependent metabolic patchwork in the nested symbiosis of mealybugs. *Current Biology*, 21(16):1366–1372, 2011.

[27] Andrew G McDonald, Sinead Boyce, and Keith F Tipton. Explorenz: the primary source of the iubmb enzyme list. *Nucleic acids research*, 37(suppl 1):D593–D597, 2009.

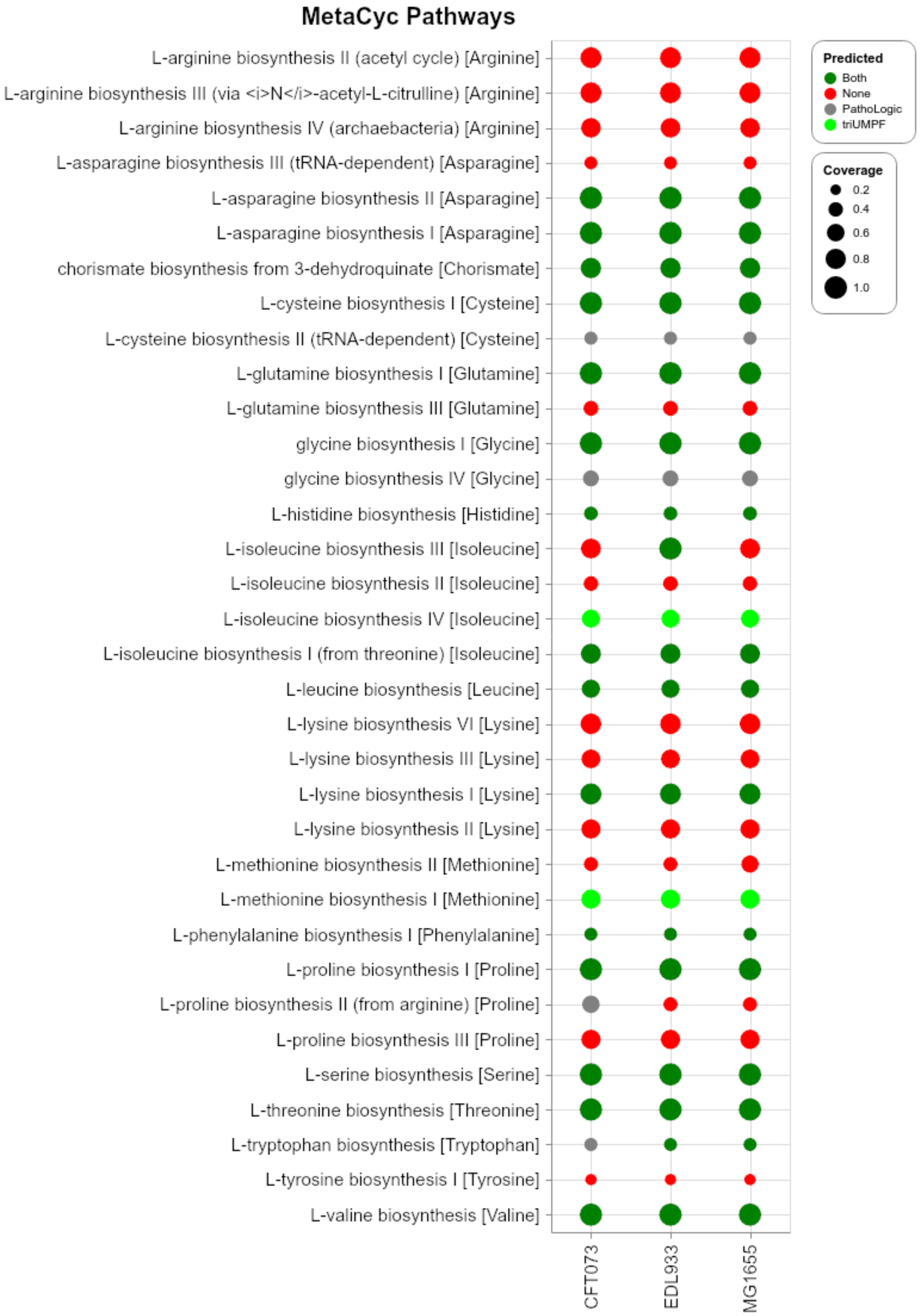


Figure 11: Comparison of predicted pathways for E. coli K-12 substr. MG1655 (TAX-511145), E. coli str. CFT073 (TAX-199310), and E. coli O157:H7 str. EDL933 (TAX-155864) datasets between PathoLogic (without taxonomic pruning) and triUMPF. Red circles indicate that neither method predicted a specific pathway while green circles indicate that both methods predicted a specific pathway. Lime circles indicate pathways predicted solely by mlLGPR and gray circles indicate pathways solely predicted by PathoLogic. The size of circles corresponds the associated coverage information.

[28] Nagarajan Natarajan and Inderjit S Dhillon. Inductive matrix completion for predicting gene–disease associations. *Bioinformatics*, 30(12):i60–i68, 2014.

[29] Morgan N Price, Grant M Zane, Jennifer V Kuehl, Ryan A Melnyk, Judy D Wall, Adam M

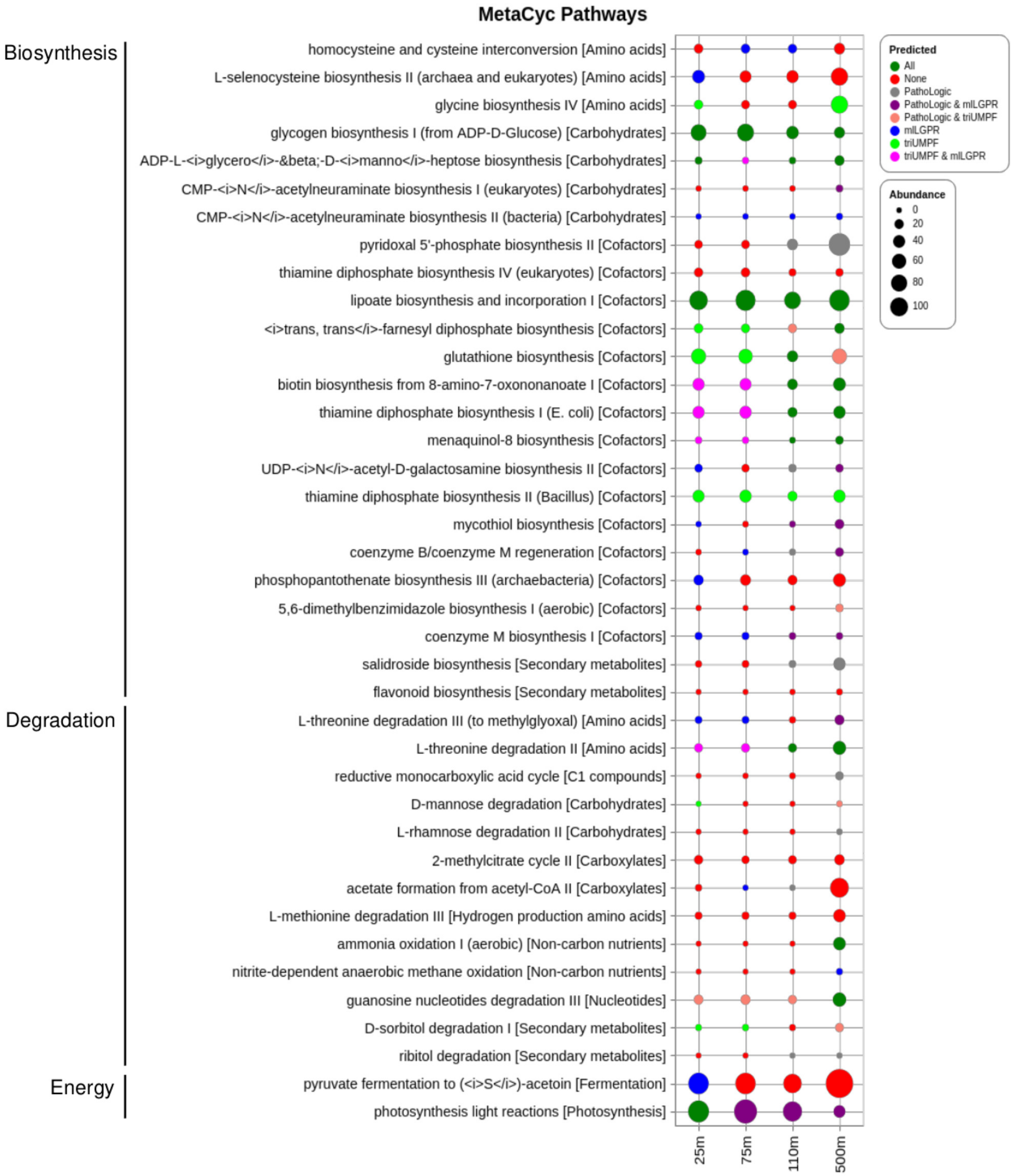


Figure 12: Comparative study of predicted pathways for HOT DNA samples. The size of circles corresponds the associated coverage information.

Deutschbauer, and Adam P Arkin. Filling gaps in bacterial amino acid biosynthesis pathways with high-throughput genetics. *PLoS genetics*, 14(1), 2018.

[30] Ryan A Rossi, Di Jin, Sungchul Kim, Nesreen K Ahmed, Danai Koutra, and John Boaz Lee. From community to role-based graph embeddings. *arXiv preprint arXiv:1908.08572*, 2019.

[31] Alexander Sczyrba, Peter Hofmann, Peter Belmann, David Koslicki, Stefan Janssen, Johannes Dröge, Ivan Gregor, Stephan Majda, Jessika Fiedler, Eik Dahms, et al. Critical

assessment of metagenome interpretation—a benchmark of metagenomics software. *Nature methods*, 14(11):1063, 2017.

[32] Mahdi Shafiei, Katherine A Dunn, Hugh Chipman, Hong Gu, and Joseph P Bielawski. Biomenet: A bayesian model for inference of metabolic divergence among microbial communities. *PLoS Comput Biol*, 10(11):e1003918, 2014.

[33] Frank J Stewart, Adrian K Sharma, Jessica A Bryant, John M Eppley, and Edward F DeLong. Community transcriptomics reveals universal patterns of protein sequence conservation in natural microbial communities. *Genome biology*, 12(3):R26, 2011.

[34] David Toubiana, Rami Puzis, Lingling Wen, Noga Sikron, Assylay Kurmanbayeva, Aigerim Soltabayeva, Maria del Mar Rubio Wilhelmi, Nir Sade, Aaron Fait, Moshe Sagi, et al. Combined network analysis and machine learning allows the prediction of metabolic pathways from tomato metabolomics data. *Communications Biology*, 2(1):214, 2019.

[35] Xiao Wang, Peng Cui, Jing Wang, Jian Pei, Wenwu Zhu, and Shiqiang Yang. Community preserving network embedding. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[36] Rodney A Welch, V Burland, GIII Plunkett, P Redford, P Roesch, D Rasko, EL Buckles, S-R Liou, A Boutin, Jeremiah Hackett, et al. Extensive mosaic structure revealed by the complete genome sequence of uropathogenic escherichia coli. *Proceedings of the National Academy of Sciences*, 99(26):17020–17024, 2002.

[37] Zi Yang and George Michailidis. A non-negative matrix factorization method for detecting modules in heterogeneous omics multi-modal data. *Bioinformatics*, 32(1):1–8, 2015.

[38] Yuzhen Ye and Thomas G Doak. A parsimony approach to biological pathway reconstruction/inference for genomes and metagenomes. *PLoS Comput Biol*, 5(8):e1000465, 2009.