Relabeling metabolic pathway data with groups to improve prediction outcomes *

Abdur Rahman M. A. Basher[1[0000−0002−3407−1187]] and Steven J. Hallam[1,2[0000−0002−4889−6876]]

[1] Graduate Program in Bioinformatics, University of British Columbia, Vancouver, BC V5Z 4S6, Canada
arbasher@student.ubc.ca

[2] Department of Microbiology & Immunology, University of British Columbia, Vancouver, BC V6T 1Z3, Canada
shallam@mail.ubc.ca

Abstract. Metabolic pathway inference from genomic sequence information is an integral scientific problem with wide ranging applications in the life sciences. As sequencing throughput increases, scalable and performative methods for pathway prediction at different levels of genome complexity and completion become compulsory. In this paper, we present reMap (relabeling metabolic pathway data with groups) a simple, and yet, generic framework, that performs relabeling examples to a different set of labels, characterized as groups. A pathway group is comprised of a subset of statistically correlated pathways that can be further distributed between multiple pathway groups. This has important implications for pathway prediction, where a learning algorithm can revisit a pathway multiple times across groups to improve sensitivity. The relabeling process in reMap is achieved through an alternating feedback process. In the first feed-forward phase, a minimal subset of pathway groups is picked to label each example. In the second feed-backward phase, reMap's internal parameters are updated to increase the accuracy of mapping examples to pathway groups. The resulting pathway group dataset is then be used to train a multi-label learning algorithm. reMap's effectiveness was evaluated on metabolic pathway prediction where resulting performance metrics equaled or exceeded other prediction methods on organismal genomes with improved predictive performance.

Keywords: pathway group · relabeling · data augmentation · correlated models · metabolic pathway prediction · MetaCyc

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

1 Introduction

Biological systems operate on the basis of information flow between genomic DNA, RNA and proteins. Proteins catalyze most reactions producing metabolites. Reaction sequences are called pathways when they contribute to a coherent set of interactions driving metabolic flux within or between cells. Inferring metabolic pathways from genomic sequence information is a fundamental problem in studying biological systems with far-reaching implications for our capacity to perceive, evaluate and engineer cells at the individual, population, and community levels of biological organization [4, 8]. Over the past decade, the rise of next generation sequencing platforms has created a veritable tidal wave of organismal and multi-organismal genomes that must be assembled and annotated at scale without intensive manual curation. In response to this need, gene-centric and pathway-centric methods have been developed to reconstruct metabolic pathways from genomic sequence information at different levels of complexity and completion. The most common methods are gene-centric and involve mapping predicted protein coding sequences onto known pathways using a reference database (e.g. the Kyoto Encyclopedia of Genes and Genomes (KEGG) [6]. Alternative pathway-centric methods including PathoLogic [7] and MinPath [20] predict the presence of a given metabolic pathway based on heuristic or rule-based algorithms. While gene-centric methods are effective at producing parts list, they are unable to infer pathway presence or absence given a set of predicted protein coding sequences. Conversely, while pathway-centric methods infer pathway presence or absence given a set of predicted protein coding sequences, the development of reliable and flexible rule sets is both difficult and time consuming [19].

Machine learning methods aim to improve on heuristic or rule-based pathway inference through features engineering and algorithmic solutions to overcome noise and class imbalance. Basher and colleagues developed mlLGPR [11], a multi-label classification method that uses logistic regression and feature vectors inspired by the work of Dale and colleagues [3] to predict metabolic pathways from genomic sequence information at different levels of complexity and completion [11]. Recently, triUMPF ([12, 13]) was proposed to reconstruct metabolic pathways from organismal and mutli-organismal genomes. This method uses meta-level interactions among pathways and enzymes within a network to improve the accuracy of pathway predictions in terms of communities represented by a cluster of nodes (pathways and enzymes). Despite triUMPF's predictive gains, its sensitivity scores on pathway datasets left extensive room for improvement. Here, we present reMap that relabels each example with a new label set called "pathway group" or "group" forming a pathway group dataset which then can be employed by a suitable pathway prediction algorithm (e.g. leADS [14]) to improve prediction results.

A subset of pathways in multiple organisms may be statistically correlated and this subset constitutes a group. Thus, the presence of a pathway entails the presence of a set of other correlated pathways. reMap performs an iterative procedure to group statistically related pathways into a set of "pathway groups"

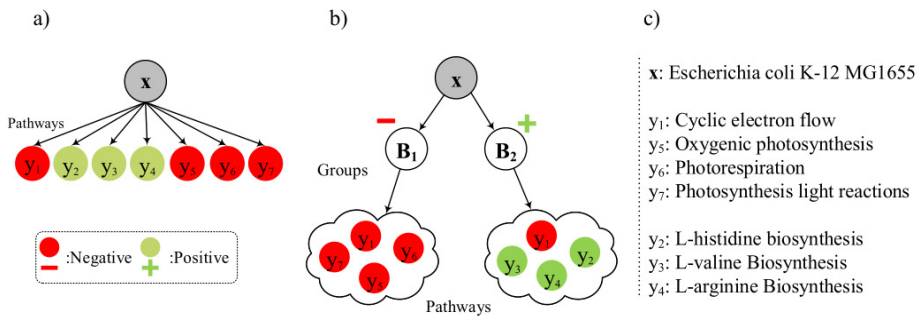


Fig. 1: Traditional vs proposed group-based pathway prediction methods. In the traditional method (a) pathways (i.e., y^{1-7}) are predicted for Escherichia coli K-12 MG1655, denoted by $\mathbf{x}$, without considering any grouping of pathways. In contrast, the group-based pathway prediction method (b) uses a two step process. First, it predicts a set of positive groups (i.e., $\mathbf{B}_2$), then the pathways within these groups are predicted (depicted as a cloud glyph and true pathways are green colored). The description of symbols is provided in subfigure (c).

using a correlation model (CTM, SOAP, and SPREAT see Appx. Section B). reMap then annotates organismal genomes with relevant groups. Pathways in these groups are correlated and allowed to be inter-mixed across groups with different proportions, resulting in an overlapping subset of groups over a subset of pathways (i.e., non-disjoint). This has important implications for pathway prediction, where a learning algorithm can revisit a pathway multiple times across groups to improve sensitivity. Unlike mlLGPR [11] and triUMPF (Fig. 1a), group based pathway prediction requires two consecutive parts. First, a set of pathway groups are inferred. In the second, pathways in these groups are predicted (Fig. 1b).

reMap's pathway grouping performance was compared with other methods including MinPath, PathoLogic, and mlLGPR on a set of Tier 1 (T1) pathway genome databases (PGDBs), low complexity microbial communities including symbiont genomes encoding distributed metabolic pathways for amino acid biosynthesis [15], genomes used in the Critical Assessment of Metagenome Interpretation (CAMI) initiative [16], and whole-genome shotgun sequences from the Hawaii Ocean Time Series (HOTS) [17] following the genomic information hierarchy benchmarks initially developed for mlLGPR enabling more robust comparison between pathway prediction methods [11].

2 Method

In this section, we provide a general description of the reMap method, presented in Fig. 2. reMap is trained in two phases using an alternating feedback process: i)- feed-forward in Figs 2(b-d), consisting of three components: 1)- constructing pathway group, 2)- building group centroid, 3)- mapping examples to groups;

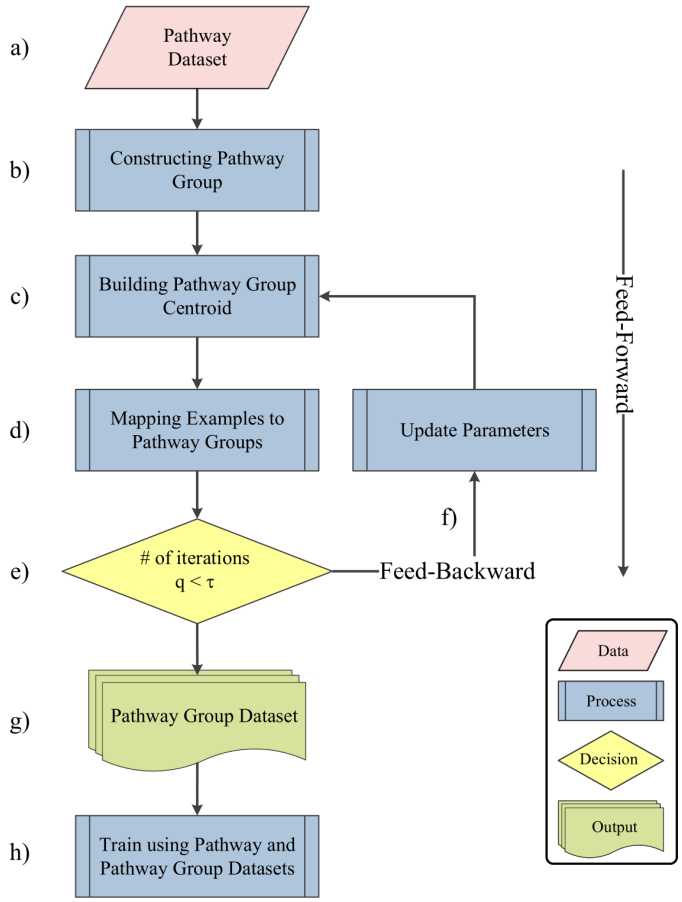


Fig. 2: A workflow diagram for reMAP. The relabeling process in reMap is achieved through an alternating feedback process. The feed-forward phase is composed of three components: (b) pathway group construction to build correlated pathway groups from pathway data (a), (c) building group centroid to estimate centroids of groups, and (d) mapping examples to groups. The feed-backward phase (f) optimizes reMap's parameters to increase accuracy of mapping examples to groups. The process is repeated τ ($\in \mathbb{Z}_{>1}$) times. If the current iteration q ($\in \mathbb{Z}_{>1}$) reaches the desired number of rounds τ, the training is terminated (e) and the pathway group dataset is produced (g) which can be used as inputs to a pathway inference algorithm (e.g. leADS [14]) to predict a set of pathways from a newly sequenced genome (h).

and ii)- feed-backward to update reMap's parameters in Fig. 2(f). After training is accomplished, a pathway group dataset is produced that can be used to predict metabolic pathways from a newly sequenced genome in Figs 2(g-h). Below, we

discuss these two phases while the analytical expressions of reMap are explained in Appx. Section A.

2.1 Feed-Forward Phase

During this stage, each example in a given pathway data (Appx. Def. 1) is annotated with a subset of pathway groups in three consecutive steps:

Constructing Pathway Group. In this step, pathways are partitioned into non-disjoint b ($\in \mathbb{Z}_{\geq 1}$) groups using any correlation models defined in Appx. Section B. These models are equipped to provide us with a group correlation matrix and a pathway distribution over groups, denoted by $\Phi \in \mathbb{R}^{b \times t}$, where t corresponds to the total number of distinct pathways. Each entry $\Phi_{i,j}$ corresponds to the probability of assigning a pathway j to the group i. For each group in Φ, we retain the top k ($\in \mathbb{Z}_{\geq 1}$) pathways based on the probability scores. The trimmed Φ serves as an input to constructing centroids in the next step.

Modeling pathway distribution and group correlation in this way are motivated by two key intuitions. First, organisms encoding similar pathways may share similar groups resulting in shared statistical properties for those organisms. Second, frequently occurring pathways in multiple organisms imply a similar relative contribution to a group.

Building Group Centroid. Having obtained a set of groups, reMap determined the relative contribution of each pathway to its associated group's centroid in the Euclidean space. Estimating centroids requires representing pathways and groups as vectors of real numbers. For this, we apply pathway2vec [10] to obtain pathway features. Then, the centroid of a group, say s, is computed as:

$$\mathbf{c}_s = \frac{\alpha}{n_s} \sum_{j \in \mathbf{B}_{s,j}=+1} \frac{\mathbf{P}_j}{||\mathbf{P}_j||} \tag{2.1}$$

where $\mathbf{B}_s \in \{-1,+1\}^t$ is the group s obtained from the trimmed Φ_s after transforming it to $\{-1,+1\}^t$. $\mathbf{c}_s$ corresponds the centroid of the group s, $\mathbf{P} \in \mathbb{R}^{t \times m}$ is a pathway representation matrix obtained from pathway2vec, n_s is the number of pathways ($|\{\mathbf{B}_{s,j} = +1, \forall j \in t\}|$) in group s, $||.||$ is the length of a feature vector, and α ($\in \mathbb{R}_{>0}$) is a hyper-parameter determined by empirical analysis (16 in this work). The proposed Eq. 2.1 is based on the intuition that pathways associated with a group are semantically "close enough" to the center of the corresponding group, and the overlapping pathways among groups exhibit similar semantics with their associated groups. In addition to determining centroids, reMap also estimates a maximum number of expected groups to be annotated for a given example, indexed by i, using the cosine similarity metric [9]:

$$\begin{aligned} \widehat{\mathbf{D}_i} &= \left(\left\{ \mathbb{I}\left(\frac{\mathbf{c}_s^\top \tilde{\mathbf{c}}_s^{(i)}}{||\mathbf{c}_s|| \cdot ||\tilde{\mathbf{c}}_s^{(i)}||} \geq \upsilon \right) : 1 \leq s \leq b \right\} \right. \\ \tilde{\mathbf{c}}_s^{(i)} &= \frac{\alpha}{n_s} \sum_{j \in (\mathbf{Y}_{i,j}=+1 \wedge \mathbf{B}_{s,j}=+1)} \frac{\mathbf{P}_j}{||\mathbf{P}_j||} \end{aligned} \tag{2.2}$$

where $\mathbb{I}(.)$ is an indicator function that results in either $+1$ or -1 depending on a user-defined threshold υ ($\in \mathbb{R}_{>0}$). $\mathbf{Y}_i \in \{-1,+1\}^t$ corresponds to pathways either present or absent for the ith example, indicated by $+1$ and -1, respectively. $\tilde{\mathbf{c}}_s^{(i)}$ represents the centroid of the group s calculated based on pathways that are associated with the group s and are present in ith example. $\tilde{n_s}$ is the number of pathways ($|\{\mathbf{Y}_{i,j} = +1 \wedge \mathbf{B}_{s,j} = +1, \forall j \in t\}|$) in group s. $\widehat{\mathbf{D}_i} \in \{+1,-1\}^b$ is a pre-optimized set of groups labelled for the ith example that will be used in the mapping step.

Mapping Pathways to Pathway Groups. This step maps an example to pathway groups, resulting in an optimized pathway group dataset $\widehat{\mathbf{D^{opt}}}$ ($\in \{+1,-1\}^{n\times b}$). Formally, let us denote a set of groups that are picked to label an example by $\mathcal{B}_{\mathbf{P}}^{(i)} \subseteq \arg\{\widehat{\mathbf{D}_{i,j}} = +1 : \forall j\}$ while the remaining unpicked groups is denoted by $\mathcal{B}_{\mathbf{U}}^{(i)}$, where $\widehat{\mathbf{D}_i}$ is obtained using Eq. 2.2. Both sets of groups are stored in $\mathcal{L}^{(i)} = \{\mathcal{B}_{\mathbf{P}}^{(i)} \cup \mathcal{B}_{\mathbf{U}}^{(i)}\}$. Then, reMap performs mapping in an iterative way, mirroring sequential learning and prediction strategy [18], where for each ith example, a group $\mathbf{B}_j$ at round q is either: i)-added to $\mathcal{L}^{(i)}$, indicated by $\mathcal{L}_q^{(i)} = \mathcal{L}_{q-1}^{(i)} \oplus \{\mathbf{B}_j : 1 < j \leq |\mathcal{B}_{\mathbf{U}}^{(i)}|\}$; or ii)- removed from the set of selected groups, represented by $\mathcal{L}_q^{(i)} = \mathcal{L}_{q-1}^{(i)} \ominus \{\mathbf{B}_j : 1 < j \leq |\mathcal{B}_{\mathbf{P}}^{(i)}|\}$. More specifically, at each iteration q, reMap estimates the probability of an example, given the selected groups that are obtained from the previous round $q-1$, using the threshold closeness (TC) metric [2] as:

$$p\Big(\mathbf{x}^{(i)}|\mathbf{H}_{q-1}^{(i)}, \mathcal{L}_{q-1}^{(i)}, \widehat{\mathbf{D}_{i,j}} = +1\Big) = \frac{\bar{p}_{\mathbf{H}_{q-1}^{(i)}}\Big(\widehat{\mathbf{D}_{i,j}}|\mathcal{L}_{q-1}^{(i)}, \mathbf{x}^{(i)}\Big)G + \zeta}{Z} \tag{2.3}$$

where $\mathbf{x}^{(i)} \in \mathbb{R}^r$ and r is the total number of enzymes, $G = 1 - \bar{p}_{\mathbf{H}_{q-1}^{(i)}}(\widehat{\mathbf{D}_{i,j}}|\mathcal{L}_{q-1}^{(i)}, \mathbf{x}^{(i)})$ and $\widehat{\mathbf{D}_{i,j}} = +1$ if the group $\mathbf{B}_j$ is tagged with the ith example. $\mathbf{H}_{q-1}^{(i)}$ represents the history of prediction probability storing all $p(\widehat{\mathbf{D}_{i,j}}|\mathcal{L}_{q-1}^{(i)}, \mathbf{x}^{(i)})$ before the current iteration q while $\bar{p}_{\mathbf{H}_{q-1}^{(i)}}(\widehat{\mathbf{D}_{i,j}}|\mathcal{L}_{q-1}^{(i)}, \mathbf{x}^{(i)})$ is the average probability of classifying $\mathbf{x}^{(i)}$ to the group $\mathbf{B}_j$ over values in $\mathbf{H}_{q-1}^{(i)}$. The term ζ ($\in \mathbb{R}_{>0}$) is a smoothness constant and Z is a normalization constant. Note that TC is a class conditional probability density function that encourages correct class probability to be close to the true unknown decision boundary. Hence, this step will ensure the correct latent group to be assigned to the ith example. The parameter $p(\widehat{\mathbf{D}_{i,j}}|\mathcal{L}_{q-1}^{(i)}, \mathbf{x}^{(i)})$ can be estimated using Appx. Eq. A.4. Afterwards, $\mathcal{L}^{(i)}$ will be updated either by adding or removing groups from a previous iteration. More details about this step is provided in Appx. Section A.1.

2.2 Feed-Backward Phase

During this phase, reMap updates its internal parameters by enforcing four constraints: i)- similarity between groups and associated pathways; ii)- weights of

pathways, in a group, should be close to each other; iii)- examples sharing similar pathways should share similar representations; and iv)- all reMap's parameters should not be too large or too small. These four constraints are important to allow smooth updates and mapping operations. More details are provided in Appx. Section A.2.

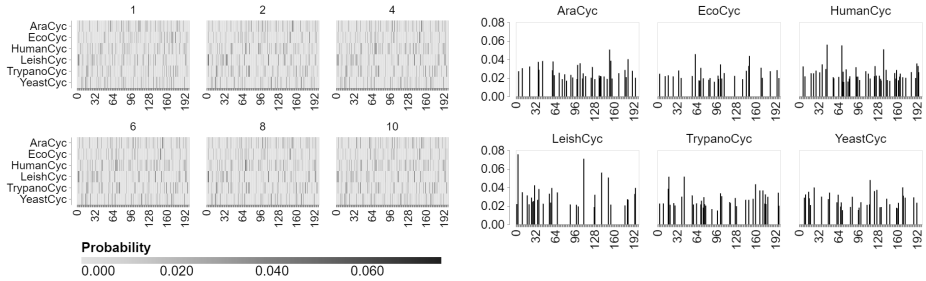
2.3 Closing the loop

The two phases are repeated for all examples in a given pathway data, until a predefined number of rounds τ ($\in \mathbb{Z}_{>1}$) is reached. At the end, a pathway group dataset is produced which consists of n examples with the assigned groups, i.e., $\widehat{\mathbf{D^{opt}}}$. After training is accomplished, a pathway group dataset is produced that can be used to predict metabolic pathways from a newly sequenced genome using an ML prediction method such as leADS [14].

3 Experiments

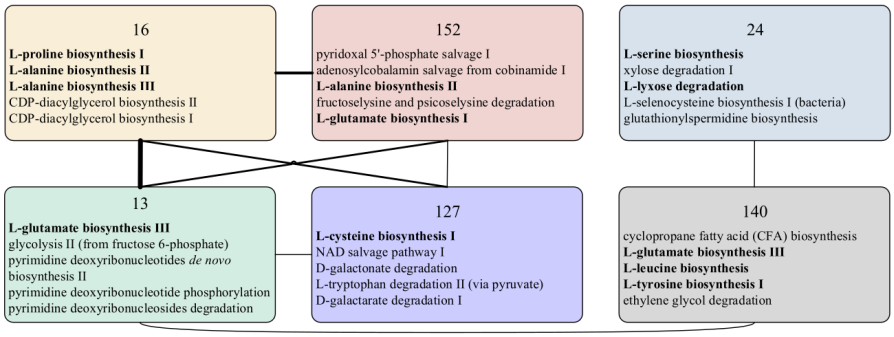
We evaluated reMap's performance on diverse pathway datasets traversing the genomic information hierarchy [11]: i)- T1 golden consisting of EcoCyc, HumanCyc, AraCyc, YeastCyc, LeishCyc, and TrypanoCyc; ii)- BioCyc (v20.5 T2 & 3) [1]; iii)- *Symbionts* genomes of *Moranella* (GenBank NC-015735) and *Tremblaya* (GenBank NC-015736) encoding distributed metabolic pathways for 9 amino acid biosynthesis [15]; iv)- Critical Assessment of Metagenome Interpretation (CAMI) dataset composed of 40 genomes [16]; and v)- whole genome shotgun sequences from the Hawaii Ocean Time Series (HOTS) at 25m, 75m, 110m (sunlit) and 500m (dark) ocean depth intervals [17]. Information about these datasets is presented in Appx. Section C.1.

Two experiments were conducted: i)- assessing the history probability and ii)- metabolic pathway prediction. The goal of the former test is to analyze the accumulated probability stored in $\mathbf{H}$ during the mapping process in the feed-forward phase for golden T1 datasets. We expect that few groups containing statistically related pathways will be annotated for T1 golden data. The metabolic pathway prediction test is followed to verify the quality of pathway groups for T1 golden, symbionts, CAMI, and HOTS data. For comparative analysis, reMap's performance on T1 golden datasets was compared to four pathway prediction methods: i)- MinPath version 1.2 [20], an integer programming based algorithm; ii)- PathoLogic version 21 [7], a symbolic approach that uses a set of manually curated rules to predict pathways; iii)- mlLGPR [11], a supervised multi-label classification and rich feature information algorithm, and iv)- triUMPF [12, 13], a non-negative matrix factorization and community detection based algorithm. Four metrics were used to report the performance of all pathway predictors for golden T1 and CAMI data: *average precision*, *average recall*, *average F1 score (F1)*, and *Hamming loss* as described in [11]. In addition, reMap's performance was compared to PathoLogic, mlLGPR, and triUMPF on mealybug symbionts



(a) During 10 successive rounds

(b) After 10 successive rounds



(c) Seven pathway groups that contain at least one amino acid biosynthesis pathways in these groups for Escherichia coli K-12 MG1655

Fig. 3: Fig. 3a illustrates the history probability **H** during annotation of T1 golden data over 10 successive rounds while Fig. 3b shows the results after 10th round. Darker colors indicate higher probabilities of assigning groups to the corresponding data. Fig. 3c shows six pathway groups and their correlations for Escherichia coli K-12 MG1655. Numbers at top boxes correspond to group indices. Edge thickness reflects the degree of associations between groups. Boldface text represent amino acid biosynthesis pathways.

and HOTS multi-organismal datasets. To construct pathway groups, we employed the correlated model SOAP using $b = 200$ groups.

reMap was written in Python v3 and is available under the GNU license at https://github.com/hallamlab/reMap. Unless otherwise specified all tests were conducted on a Linux server using 10 cores of Intel Xeon CPU E5-2650. For full experimental settings and additional tests, see Appx. Sections C and D.

3.1 Accumulated History Probability Analysis

Fig. 3a shows **H** during the annotation process for the T1 golden data over 10 iterations. In the beginning, reMap attempts to select the maximum number

Table 1: Average F1 score of each comparing algorithm on 6 golden T1 data. Bold text suggests the best performance in each column.

Methods	Average F1 Score					
	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
reMap+SOAP	**0.8336**	**0.8285**	0.4764	0.4914	0.4144	**0.7305**

of groups that may exist for each example. However, with progressive updates and calibration of parameters, reMap rectifies groups assignments where it picks fewer relevant groups for each example. As an example, after the 10th round, Escherichia coli K-12 MG1655 was tagged with only 33 groups (Fig. 3b) and 18 of these groups contain amino acid biosynthesis pathways. Fig. 3c shows 7 of these 18 pathway groups (Appx. Table 5). Pathways in these 7 groups are statistically related (Appx. Table 6), and are observed to be distributed across groups reflected by the thickness of edges in Fig. 3c. For example, *L-alanine biosynthesis II* pathway is present in groups indexed by 16 and 152. Similarly, for the pathway *L-glutamate biosynthesis III* which is represented in groups indexed by 13 and 140. This mixture of pathway representation over groups increases the chance of a pathway inference algorithm (e.g. leADS [14]) to revisit a true positive pathway multiple times across groups which may result in improved predictions as reported in the next section. This experiment shows that reMap is able to capture statistically relevant pathways and map related groups to each example with a high degree of correlation.

3.2 Metabolic Pathway Prediction

T1 Golden data. Table 1 shows that reMap+SOAP achieved competitive performance against the other methods in terms of average F1 score with optimal performance on EcoCyc (0.8336). However, it under-performed on AraCyc, YeastCyc, and LeishCyc, yielding average F1 scores of 0.4764, 0.4914, and 0.4144, respectively. Since reMap+SOAP was trained using BioCyc containing less than 1460 trainable pathways, pathways outside the training set will be neglected.

Symbionts data. The goal of this test is to evaluate reMap+SOAP performance on distributed metabolic pathways that emerge as a result of interactions between two or more organisms. We used the reduced genomes of *Moranella* and *Tremblaya* [15] as an established model for benchmarking. The two symbiont genomes in combination encode 9 intact amino acids biosynthesis pathways. All four pathway predictors were used to predict pathways on individual symbiont genomes and a composite genome consisting of both. While reMap+SOAP, triUMPF and PathoLogic predicted 6 of the expected amino acid biosynthesis pathways on the composite genome, mlLGPR was able to predict 8 pathways

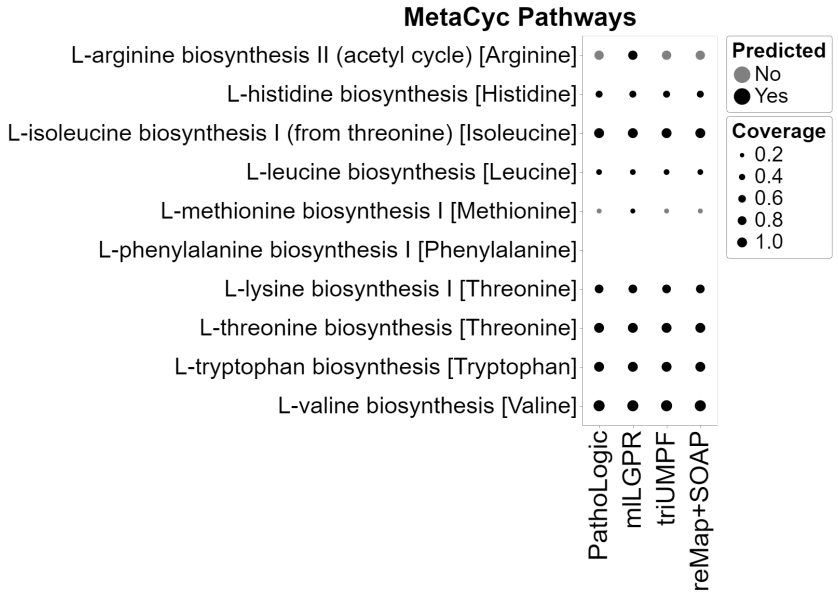


Fig. 4: Comparative study of predicted pathways for the composite genome between PathoLogic, mlLGPR, triUMPF, and reMap+SOAP. Black circles indicate predicted pathways by the associated models while grey circles indicate pathways that were not recovered by models. The size of circles corresponds the pathway coverage information.

(Fig. 4). We excluded phenylalanine biosynthesis (*L-phenylalanine biosynthesis I*) pathway from analysis because the associated genes were reported to be missing after initial gene prediction. Four predictors identified false positives for individual symbiont genomes in *Moranella* and *Tremblaya* although the pathway coverage information for both genomes was reduced in relation to the composite genome (Appx. Fig. 9).

CAMI and HOTS data. For CAMI low complexity data [16], reMap+SOAP exceeded mlLGPR and triUMPF, achieving an average F1 score of 0.6125 in compare to 0.4866 for mlLGPR and 0.5864 for triUMPF (Table 2). For HOTS data [17], triUMPF, mlLGPR, and PathoLogic predicted a total of 58, 62, and 54 pathways, respectively, while reMap+SOAP inferred 67 pathways (see Appx. Section D.3) from a subset of 180 selected water column pathways [5]. None of the algorithms were able to predict pathways for *photosynthesis light reaction* and *pyruvate fermentation to (S)-acetoin* despite the abundance of these pathways in the water column. Absence of specific EC numbers associated with each pathway likely contributed to their absence using rule-based or ML prediction algorithms. Results from this experiment indicates that the proposed pathway group based approach, in particular reMap+SOAP increases pathway prediction performance relative to other methods used in isolation.

Table 2: Predictive performance of mlLGPR, triUMPF, and reMap+SOAP 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	reMap+SOAP
Hamming Loss (↓)	0.0975	0.0436	**0.0407**
Average Precision Score (↑)	0.3570	0.7027	**0.7419**
Average Recall Score (↑)	**0.7827**	0.5101	0.5283
Average F1 Score (↑)	0.4866	0.5864	**0.6125**

4 Conclusion

In this paper, we demonstrated that iteratively mapping examples to groups e.g. relabeling, using reMap increased pathway prediction performance. The reMAP method is based on the intuition that organisms sharing a similar set of metabolic pathways may exhibit similar higher-level structures or groups. The relabeling process in reMap is achieved through an alternating feedback process. In the first feed-forward phase, a minimal subset of pathway groups is picked to label each example. In the second feed-backward phase, reMap's internal parameters are updated to increase the accuracy of mapping examples to pathway groups. After training reMap, a pathway group dataset is produced that can be used to predict metabolic pathways for a newly sequenced genome.

We evaluated reMap's performance for the pathway prediction task using a corpus of experimental datasets and compared results to other prediction methods including PathoLogic, MinPath, mlLGPR, and triUMPF. Overall, reMap showed promising results in boosting prediction performance over ML-based algorithms, such as mlLGPR and triUMPF. During benchmarking, we realized that reMap brings more frequent and sometimes irrelevant pathways, resulting in a significant performance loss on some T1 golden data, such as AraCyc. A possible treatment would be adding constraints in the form of associations among enzymes and pathways as applied in triUMPF. However, this may lead to sensitivity loss [14]. Another approach is to combine both graph-based and group-based strategies to predict pathways. Future development efforts will explore this dual approach to improve pathway prediction performance with emphasis on multi-organismal genomes encoding distributed metabolic processes.

References

1. Caspi, R., Billington, R., Foerster, H., et al.: Biocyc: Online resource for genome and metabolic pathway analysis. The FASEB Journal **30**(1 Supplement), lb192–lb192 (2016)
2. Chang, H.S., Learned-Miller, E., McCallum, A.: Active bias: Training more accurate neural networks by emphasizing high variance samples. In: Advances in Neural Information Processing Systems. pp. 1002–1012 (2017)
3. Dale, J.M., Popescu, L., Karp, P.D.: Machine learning methods for metabolic pathway prediction. BMC bioinformatics **11**(1), 1 (2010)

4. Hahn, A.S., Konwar, K.M., Louca, S., et al.: The information science of microbial ecology. Current opinion in microbiology **31**, 209–216 (2016)
5. Hanson, N.W., Konwar, K.M., Hawley, A.K., et al.: Metabolic pathways for the whole community. BMC genomics **15**(1), 1 (2014)
6. Kanehisa, M., Furumichi, M., Tanabe, M., et al.: Kegg: new perspectives on genomes, pathways, diseases and drugs. Nucleic Acids Research **45**(D1), D353–D361 (2017)
7. Karp, P.D., Latendresse, M., Paley, S.M., et al.: Pathway tools version 19.0 update: software for pathway/genome informatics and systems biology. Briefings in bioinformatics **17**(5), 877–890 (2016)
8. Lawson, C.E., Harcombe, W.R., Hatzenpichler, R., et al.: Common principles and best practices for engineering microbiomes. Nature Reviews Microbiology pp. 1–17 (2019)
9. Luo, C., Zhan, J., Xue, X., et al.: Cosine normalization: Using cosine similarity instead of dot product in neural networks. In: International Conference on Artificial Neural Networks. pp. 382–391. Springer (2018)
10. M. A. Basher, A.R., Hallam, S.J.: Leveraging heterogeneous network embedding for metabolic pathway prediction. Bioinformatics (10 2020). https://doi.org/10.1093/bioinformatics/btaa906
11. M. A. Basher, A.R., McLaughlin, R.J., Hallam, S.J.: Metabolic pathway inference using multi-label classification with rich pathway features. PLOS Computational Biology **16**(10), 1–22 (10 2020)
12. M. A. Basher, A.R., McLaughlin, R.J., Hallam, S.J.: Metabolic pathway prediction using non-negative matrix factorization with improved precision. Journal of Computational Biology (2021)
13. M. A. Basher, A.R., McLaughlin, R.J., Hallam, S.J.: Metabolic pathway prediction using non-negative matrix factorization with improved precision. In: Computational Advances in Bio and Medical Sciences. pp. 33–44. Springer International Publishing, Cham (2021)
14. M. A. Basher, A.R., Nallan, A.N., McLaughlin, R.J., et al.: leads: improved metabolic pathway inference based on active dataset subsampling. bioRxiv (2020). https://doi.org/10.1101/2020.09.14.297424
15. McCutcheon, J.P., Von Dohlen, C.D.: An interdependent metabolic patchwork in the nested symbiosis of mealybugs. Current Biology **21**(16), 1366–1372 (2011)
16. Sczyrba, A., Hofmann, P., Belmann, P., et al.: Critical assessment of metagenome interpretation—a benchmark of metagenomics software. Nature methods **14**(11), 1063 (2017)
17. Stewart, F.J., Sharma, A.K., Bryant, J.A., et al.: Community transcriptomics reveals universal patterns of protein sequence conservation in natural microbial communities. Genome biology **12**(3), R26 (2011)
18. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Advances in neural information processing systems. pp. 3104–3112 (2014)
19. Toubiana, D., Puzis, R., Wen, L., et al.: Combined network analysis and machine learning allows the prediction of metabolic pathways from tomato metabolomics data. Communications Biology **2**(1), 214 (2019)
20. Ye, Y., Doak, T.G.: A parsimony approach to biological pathway reconstruction/inference for genomes and metagenomes. PLoS Comput Biol **5**(8), e1000465 (2009)

Appendix

The appendix is divided into four parts: i)- the reMap framework (Section A), ii)- descriptions about correlated models (Section B), iii)- experimental settings (Section C), and iv)- empirical analysis (parameter sensitivity, history probability analysis, and metabolic pathway prediction) (Section D).

A The reMap Method

In this section, we provide important notations and definitions that will be used throughout the paper followed by a formal description of the research problem. All vectors are assumed to be column vectors and are represented by boldface lowercase letters (e.g., $\mathbf{x}$) while matrices are encoded 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 cell entry (i, j) of $\mathbf{X}$. A subscript character to a vector, $\mathbf{x}_i$, denotes an i-th cell of $\mathbf{x}$. Occasional superscript, $\mathbf{X}^{(i)}$, suggests an index to a example or current epoch during the learning period. The sets are characterized by calligraphic letters (e.g., $\mathcal{E}$) while we use the notation $|.|$ to denote the cardinality of a given set. With these notations, we introduce the problem examined in this paper, starting with a multi-label pathway dataset definition.

Definition 1. ***Multi-label Pathway Dataset*** *[11]. 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 the enzymatic reactions. 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, hence, the vector size* $\mathbf{x}^{(i)}$ *is* r*. The abundance of an enzymatic reaction for an example* i*, say* $e_l^{(i)}$*, is defined as* $a_l^{(i)} (\in \mathbb{R}_{\geq 0})$*. The class labels* $\mathbf{y}^{(i)} = [y_1^{(i)}, ..., y_t^{(i)}] \subseteq \{-1, +1\}^t$ *is a pathway label vector of size* t *that represents the total number of pathways, which themselves are derived from a set of universal metabolic pathway* $\mathcal{Y}$*. The entry* $+1$ *(or* -1*) indicates presence (or absence) of a pathway corresponding the example* i*. The matrix form of* $\mathbf{x}$ *and* $\mathbf{y}$ *are symbolized as* $\mathbf{X}$ *and* $\mathbf{Y}$*, respectively.* ■

Both $\mathcal{E}$ and $\mathcal{Y}$ are extracted from reliable knowledge-bases (e.g. KEGG [6] and MetaCyc [26]). In this paper, we adopt MetaCyc. Moving on, we define the term *pathway group set*.

Definition 2. ***Pathway Group Set.*** *Denote* $\mathcal{B} = \{\mathbf{B}_1, \mathbf{B}_2, ..., \mathbf{B}_b\}$ *a set with* b *pathway groups, where each group* $\mathbf{B}_c \in \{-1, +1\}^t$ *is presumed to contain a subset of correlated pathways, i.e.,* $\mathcal{Y}_c \subseteq \mathcal{Y}$*, and* t *is the number of pathways in Def. 1. The presence or absence of a pathway in a group* c *is indicated by* $+1$ *or* -1*, respectively. The matrix representation of* $\mathcal{B}$ *is* $\mathbf{B} \in \{-1, +1\}^{b \times t}$*.* ■

Pathway groups are also assumed to be correlated, i.e, non-disjoint, and can be modeled by a Gaussian covariance matrix, denoted by $\Sigma \in \mathbb{R}^{b \times b}$. Each entry $s_{i,j}$ in Σ characterizes the i-th group association with j-th group, where a larger

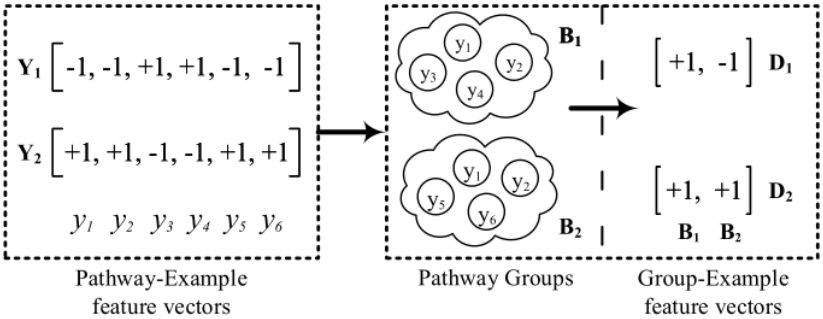


Fig. 5: An example of feature vectors for groups. The subfigure in the left represents the feature vector for six pathways corresponding to two examples. The right subfigure indicates two groups, $\mathbf{B}_1$ and $\mathbf{B}_2$, and their features for the same two examples. The first example, $\mathbf{D}_1$, suggests that only $\mathbf{B}_1$ is present because the corresponding pathways y_3 and y_4 are present, while the pathway group feature vector for the second example, $\mathbf{D}_2$, suggests that both groups are present.

score indicates both groups are highly correlated. As a result of correlation, we define the following two terminologies: *pathway group feature vector* and *pathway group's neighbor*.

Definition 3. ***Group-Example Feature Vector.*** *The pathway group feature vector for the ithe example is indicated by* $\mathbf{d}^{(i)} \in \{-1, +1\}^b$*, where* $\mathbf{d}_j^{(i)} = +1$*, iff the group* j *is observed for the example* i *and* $\mathbf{d}_j = -1$ *otherwise. The matrix form in represented as* $\mathbf{D} \in \{-1, +1\}^{n \times b}$*.* ■

An example of feature vectors for groups is illustrated in Fig. 5, where 2-dimensional feature vectors for groups encode presence or absence of two groups $\mathbf{B}_1$ and $\mathbf{B}_2$, given a set of 6 pathways and pathway-group association information, depicted as a cloud glyph.

Definition 4. ***Pathway Groups Neighbors.*** *A group* $\mathbf{B}_c \in \mathcal{B}$ *is said to be a neighbor to another pathway group* $\mathbf{B}_j \in \mathcal{B}$ *s.t.* $c \neq j$*, if there exits an intersected pathway* l *in both groups, i.e.,* $\mathbf{B}_{c,l} \wedge \mathbf{B}_{j,l} = +1$*.* ■

With the above definitions, we formulate the problem in this work.

Problem 1. Given a set of groups $\mathcal{B}$ and a multi-label pathway dataset $\mathcal{S}$, the goal is to learn an optimum relabeling function $h^{\mathbf{g}} : \mathcal{X} \rightarrow \{+1, -1\}^b$, such that leveraging groups to $\mathbf{X}$ incurs a high predictive score for the downstream pathway prediction task.

Inputs :
P: pathway features matrix ($\mathbf{P} \in \mathbb{R}^{t \times m}$)
$\mathcal{B}$: a set of b pathway groups ($\mathcal{B} = \{\mathbf{B}_1, ..., \mathbf{B}_b\}$)
α: a hyper-parameter for pathway groups' centroids construction ($\alpha \in \mathbb{R}_{>0}$)

Outputs:
C: the centroids of groups ($\mathbf{C} \in \mathbb{R}^{b \times m}$)

Process :
C: 0 ($\in \mathbb{R}^{b \times m}$);
for $s \leftarrow 1$ **to** b **do**
  $n_s \leftarrow \sum_j \mathbb{I}(\mathbf{B}_{s,j} = +1)$;
  $\mathbf{c}_s \leftarrow$ apply Eq. A.1;
  $\mathbf{C}_{s,:} = \mathbf{c}_s$;
Return **C**

Algorithm 1: GroupCentroid($\mathbf{P}$, $\mathcal{B}$, α)

Fig. 1 illustrates the benefit of incorporating groups for multi-label pathway classification (right panel). Here, a dataset consists of two groups, each consists of a set of 4 correlated pathways. To determine positive pathways (y_2, y_3, and y_4) given $\mathbf{X}_i$, we first predict the relevant group, indicated by +, then classify pathways within that pathway group. In contrast, the traditional multi-label classification approaches (left figure), mostly based on *binary relevance* technique, proceeds on predicting multiple pathway labels for $\mathbf{X}_i$. Hence, the proposed method will reduce computational complexity for pathway prediction.

Mapping a multi-label pathway dataset $\mathcal{S}$ to groups will result in another dataset, i.e., $\mathcal{S}_{group}$.

Definition 5. ***Multi-label Pathway Group Dataset***. *A group dataset is represented by* $\mathcal{S}_{group} = \{(\mathbf{x}^{(i)}, \mathbf{d}^{(i)}) : 1 < i \leqslant n\}$ *consisting of* n *examples.* $\mathbf{d}^{(i)} = [d_1^{(i)}, ..., d_t^{(i)}] \in \{-1, +1\}^b$ *is a pathway group label vector of size* b*. Each element of* $\mathbf{d}^{(i)}$ *indicates the presence/absence of the associated pathway group that is inherited from the set* $\mathcal{B}$ *in Def. 2.* ■

Now, we outline the reMap method (depicted in Fig. 2), which alternates between the following two phases: i)- feed-forward in Figs 2(b-d), consisting of three components: 1)- constructing pathway group, 2)- building group centroid, 3)- mapping examples to groups; and ii)- feed-backward to update reMap's parameters in Fig. 2(f). After training is accomplished, a pathway group dataset is produced that can used to predict metabolic pathways from a newly sequenced genome in Figs 2(g-h).

A.1 Feed-Forward Phase

During this phase, a minimal subset of groups is picked to annotate each example in a given pathway data (Def. 1) in three consecutive steps:

Inputs :
n: number of examples ($n \in \mathbb{N}_{>2}$)
$\mathbf{X}$: input space training set ($\mathbf{X} \in \mathbb{R}^{n \times r}$)
$\mathbf{Y}$: pathway space training set ($\mathbf{Y} \in \mathbb{Z}_{\geq 0}^{n \times t}$)
$\mathbf{P}$: pathway features matrix ($\mathbf{P} \in \mathbb{R}^{t \times m}$)
$\mathcal{B}$: a set of b groups ($\mathcal{B} = \{\mathbf{B}_1, ..., \mathbf{B}_b\}$)
$\mathbf{C}$: the centroids of groups ($\mathbf{C} \in \mathbb{R}^{b \times m}$)
v: a cutoff hyper-parameter ($v \in \mathbb{R}_{\geq 0}$)

Outputs:
$\widehat{\mathbf{D}}$: the expected maximum number of groups ($\widehat{\mathbf{D}} \in \mathbb{Z}_{\geq 0}^{n \times b}$)

Process :

```
// an empty matrix that will contain maximum
// number of groups for each example
D̂ ← 0;
// n is the number of examples
for i ← 1 to n do
    // b is the number of groups
    for k ← 1 to b do
        for j ← 1 to Y_{i,:} do
            if j ∈ Y_{i,j} ∧ B_{k,j} then
                D̂_{i,k} ← [apply Eq A.2];
Return D̂
```

Algorithm 2: MaxGroups(n, $\mathbf{X}$, $\mathbf{Y}$, $\mathbf{P}$,$\mathcal{B}$, $\mathbf{C}$, v)

Constructing Pathway Group. In this step, pathways in $\mathcal{S}$ are partitioned into non-disjoint b groups using any correlated models in Section B. These models are equipped to provide us with a group covariance matrix denoted by $\Sigma \in \mathbb{R}^{b \times b}$ that is transformed to a correlation matrix $\rho = C^{-1} \Sigma C^{-1}$ where $C = \sqrt{\text{diag}(\Sigma)}$, and a pathway distribution over groups denoted by $\Phi \in \mathbb{R}^{b \times t}$. Each entry $\Phi_{i,j}$ corresponds to the probability of assigning a pathway j to the group i. For each group in Φ, we retain the top k ($\in \mathbb{Z}_{\geq 1}$) pathways based on the probability scores. The trimmed $\Phi' \in \mathbb{R}^{b \times k} (\subseteq \Phi)$ serves as an input to constructing centroids in the next step. Modeling pathway distribution and group correlation in this way are motivated by two key intuitions. First, organisms encoding similar pathways may share similar groups, thus, encouraging to have near-identical statistical properties for those organisms. Second, frequently occurring pathways in multiple organisms imply a similar relative contribution to a group.

Building Group Centroid. Having obtained a set of groups, reMap computes centroids for each group to capture the relative contribution of each pathway to its associated group's centroid in the Euclidean space. Estimating centroids requires representing pathways and groups as vectors of real numbers. For this, we apply pathway2vec [10] to obtain pathway features. Then, the centroid of a group, say s, is computed according to:

$$\mathbf{c}_s = \frac{\alpha}{n_s} \sum_{j \in \mathbf{B}_{s,j}=+1} \frac{\mathbf{P}_j}{||\mathbf{P}_j||} \tag{A.1}$$

where $\mathbf{B}_s \in \{-1,+1\}^t$ is the group s obtained from the trimmed Φ_s after transforming it to $\{-1,+1\}^t$. $\mathbf{c}_s$ corresponds the centroid of the group s, $\mathbf{P} \in \mathbb{R}^{t\times m}$ is a pathway representation matrix obtained from pathway2vec, n_s is the number of pathways ($|\{\mathbf{B}_{s,j} = +1, \forall j \in t\}|$) in group s, $||.||$ is the length of a feature vector, and α ($\in \mathbb{R}_{>0}$) is a hyper-parameter determined by empirical analysis (16 in this work). The proposed Eq. A.1 is based on the intuition that pathways associated with a group are semantically "close enough" to the center of the corresponding group, and the overlapping pathways among groups exhibit similar semantics with their associated groups. This procedure is described in Algorithm 1. In addition to the centroid computation, reMap also estimates a maximum number of expected groups to be annotated for a given example, indexed by i, using cosine similarity metric [9]:

$$\begin{aligned}\widehat{\mathbf{D}_i} =& \left(\left\{\mathbb{I}\left(\frac{\mathbf{c}_s^\top \tilde{\mathbf{c}}_s^{(i)}}{||\mathbf{c}_s|| \cdot ||\tilde{\mathbf{c}}_s^{(i)}||} \geq \upsilon\right) : 1 \leq s \leq b\right\} \\ \tilde{\mathbf{c}}_s^{(i)} =& \frac{\alpha}{n_s} \sum_{j\in(\mathbf{Y}_{i,j}=+1 \wedge \mathbf{B}_{s,j}=+1)} \frac{\mathbf{P}_j}{||\mathbf{P}_j||}\end{aligned} \tag{A.2}$$

where $\mathbb{I}(.)$ is an indicator function that results in either $+1$ or -1 depending on a user-defined threshold υ ($\in \mathbb{R}_{>0}$). $\mathbf{Y}_i \in \{-1,+1\}^t$ corresponds to pathways either present or absent for the ith example, indicated by $+1$ and -1, respectively. $\tilde{\mathbf{c}}_s^{(i)}$ represents the centroid of the group s calculated based on pathways that are associated with the group s and are present in ith example. $\tilde{n_s}$ is the number of pathways ($|\{\mathbf{Y}_{i,j} = +1 \wedge \mathbf{B}_{s,j} = +1, \forall j \in t\}|$) in group s. $\widehat{\mathbf{D}_i} \in \{+1,-1\}^b$ is a pre-optimized set of groups tagged for the ith example that will be used in the mapping step. Algorithm 2 describes the pseudocode for Eq. A.2.

Mapping Pathways to Pathway Groups. The goal of this step is to map an example to pathway groups, resulting in an optimized pathway group dataset $\widehat{\mathbf{D^{opt}}} \in \{+1,-1\}^{n\times b}$. Formally, let us denote a set of groups that are picked to tag an example by $\mathcal{B}_{\mathbf{P}}^{(i)} \subseteq \arg\{\widehat{\mathbf{D}_{i,j}} = +1 : \forall j\}$ while the remaining unpicked groups is denoted by $\mathcal{B}_{\mathbf{U}}^{(i)}$, where $\widehat{\mathbf{D}_i}$ is obtained using Eq. 2.2. Both sets of groups are stored in $\mathcal{L}^{(i)} = \{\mathcal{B}_{\mathbf{P}}^{(i)} \cup \mathcal{B}_{\mathbf{U}}^{(i)}\}$. Then, reMap performs mapping in an iterative way, mirroring sequential learning and prediction strategy [18], where for each ith example, a group $\mathbf{B}_j$ at round q is either: i)-added to $\mathcal{L}^{(i)}$, indicated by $\mathcal{L}_q^{(i)} = \mathcal{L}_{q-1}^{(i)} \oplus \{\mathbf{B}_j : 1 < j \leq |\mathcal{B}_{\mathbf{U}}^{(i)}|\}$; or ii)- removed from the set of selected groups, represented by $\mathcal{L}_q^{(i)} = \mathcal{L}_{q-1}^{(i)} \ominus \{\mathbf{B}_j : 1 < j \leq |\mathcal{B}_{\mathbf{P}}^{(i)}|\}$. More specifically, at each iteration q, reMap estimates the probability of an example, given the selected groups that are obtained from the previous round $q-1$, using threshold closeness (TC) metric [2] as:

$$p\left(\mathbf{x}^{(i)}|\mathbf{H}_{q-1}^{(i)}, \mathcal{L}_{q-1}^{(i)}, \widehat{\mathbf{D}_{i,j}} = +1\right) = \frac{\bar{p}_{\mathbf{H}_{q-1}^{(i)}}\left(\widehat{\mathbf{D}_{i,j}}|\mathcal{L}_{q-1}^{(i)}, \mathbf{x}^{(i)}\right)G + \zeta}{Z} \tag{A.3}$$

where $\mathbf{x}^{(i)} \in \mathbb{R}^r$ and r is the total number of enzymes, $G = 1 - \bar{p}_{\mathbf{H}^{(i)}_{q-1}}(\widehat{\mathbf{D}_{i,j}}|\mathcal{L}^{(i)}_{q-1}, \mathbf{x}^{(i)})$ and $\widehat{\mathbf{D}_{i,j}} = +1$ if the group $\mathbf{B}_j$ is tagged with the ith example. $\mathbf{H}^{(i)}_{q-1}$ represents the history of prediction probability storing all $p(\widehat{\mathbf{D}_{i,j}}|\mathcal{L}^{(i)}_{q-1}, \mathbf{x}^{(i)})$ before the current iteration q while $\bar{p}_{\mathbf{H}^{(i)}_{q-1}}(\widehat{\mathbf{D}_{i,j}}|\mathcal{L}^{(i)}_{q-1}, \mathbf{x}^{(i)})$ is the average probability of classifying $\mathbf{x}^{(i)}$ to the group $\mathbf{B}_j$ over values in $\mathbf{H}^{(i)}_{q-1}$. The term ζ ($\in \mathbb{R}_{>0}$) is a smoothness constant and Z is a normalization constant. Note that TC is a class conditional probability density function that encourages correct class probability to be close to the true unknown decision boundary. Hence, this step will ensure the correct latent group to be assigned to the ith example. To estimate $p(\widehat{\mathbf{D}_{i,j}}|\mathcal{L}^{(i)}_{q-1}, \mathbf{x}^{(i)})$, we jointly compute the probability of groups and pathways that are associated with $\widehat{\mathbf{D}_{i,j}}$ at round $q-1$ as:

$$
\begin{aligned}
p(\widehat{\mathbf{D}_{i,j}}|\mathcal{L}^{(i)}_{q-1}, \mathbf{x}^{(i)}) \propto \mathbf{H}^{(i)}_{q-1} &\left(\sum_{e \in \mathcal{L}^{(i)}_{q-1}} z_{j,e} \left(\sum_{s \in \mathbf{B}_{j,s}=+1} p(\widehat{\mathbf{D}_{i,j}}|l_s = +1, \Theta^{\mathbf{g}}_j) p(y^{(i)}_s|\mathbf{x}^{(i)}, \Theta^{\mathbf{p}}_s) \right) \right) \\
z_{j,e} &= \frac{\rho_{j,e} - \min(\rho)}{\max(\rho) - \min(\rho)} \\
p(\widehat{\mathbf{D}_{i,j}}|l_s = +1, \Theta^{\mathbf{g}}_j) &= \frac{1}{1 + e^{-\Theta^{\mathbf{g},\mathrm{T}}_j \left| \bar{\mathbf{c}}^{(i)}_j - \mathbf{P}_s \right|}} \\
p(y^{(i)}_s|\mathbf{x}^{(i)}, \Theta^{\mathbf{p}}_s) &= \frac{1}{1 + e^{-\Theta^{\mathbf{p},\mathrm{T}}_s \mathbf{x}^{(i)}}}
\end{aligned}
\tag{A.4}
$$

where $y^{(i)}_s = +1$ if the pathway index s is found to be present in both group j and in example $\mathbf{x}^{(i)}$ and 0 otherwise, and $l_s = 1$ if the pathway index s is associated with group j and 0 otherwise. $z_{j,e}$ is a normalized correlation between groups j and e, respectively, obtained from ρ and $\bar{\mathbf{c}}^{(i)}_j$ is presented in Eq A.2. $\Theta^{\mathbf{g}}_j \in \mathbb{R}^m$ and $\Theta^{\mathbf{p}}_s \in \mathbb{R}^r$ denote parameters for the group j and the pathway s model's, respectively, and are learned during the feed-backward stage.

To reduce computational latency, instead of applying the above procedure to all groups for each example at every round, we randomly sub-sample groups of size γ ($\in \mathbb{Z}_{>1}$). Also, the estimate is still in the probability realm, therefore, we utilize a cut-off decision threshold (β) to retrieve a subset of groups having less overlapping pathways. Afterwards, $\mathcal{L}^{(i)}$ will be updated either by adding or removing groups from a previous iteration. Algorithm 3 presents the pseudocode for relabeling multi-label dataset with groups.

Inputs :

n: number of examples ($n \in \mathbb{N}_{>2}$)
$\mathbf{X}$: input space training set ($\mathbf{X} \in \mathbb{R}^{n \times r}$)
$\mathbf{Y}$: pathway space training set ($\mathbf{Y} \in \mathbb{Z}_{\geq 0}^{n \times t}$)
$\widehat{\mathbf{D}}$: the expected maximum number of groups ($\widehat{\mathbf{D}} \in \mathbb{Z}_{\geq 0}^{n \times b}$)
$\mathcal{B}$: a set of b groups ($\mathcal{B} = \{\mathbf{B}_1, ..., \mathbf{B}_b\}$)
$\mathbf{P}$: pathway features matrix ($\mathbf{P} \in \mathbb{R}^{t \times m}$)
$\Theta^{\mathbf{g}}$: groups' parameters ($\Theta^{\mathbf{g}} \in \mathbb{R}^{b \times m}$)
$\Theta^{\mathbf{p}}$: pathways' parameters ($\Theta^{\mathbf{p}} \in \mathbb{R}^{t \times r}$)
$\mathbf{C}$: the centroids of groups ($\mathbf{C} \in \mathbb{R}^{b \times m}$)
z: normalized groups's correlation ($z \in \mathbb{R}^{b \times b}$)
α: a hyper-parameter for groups' centroids construction ($\alpha \in \mathbb{R}_{>0}$)
d: a subexample size hyper-parameter ($d \in \mathbb{N}_{>1}$)
ϵ: a smoothness constant ($\epsilon \in \mathbb{R}_{>0}$)
v: a cutoff hyper-parameter for the maximum number of groups ($v \in \mathbb{R}_{\geq 0}$)
ζ: a decision threshold for selecting groups ($\zeta \in \mathbb{R}_{\geq 0}$)
τ: number of rounds ($\tau \in \mathbb{N}_{>1}$)

Outputs:

$\widehat{\mathbf{D}^{\mathbf{opt}}}$: an optimum multi-label group set ($\widehat{\mathbf{D}^{\mathbf{opt}}} \in \mathbb{Z}_{\geq 0}^{n \times b}$)

Process :

```
// an empty matrix that will contain maximum
// number of groups for each example
```
$\widehat{\mathbf{D}^{\mathbf{opt}}} \leftarrow 0$;
for $i \leftarrow 1$ **to** n **do**
  `// an initial set of groups for an example i`
  $\mathcal{L}^{(i)} = \arg\{\widehat{\mathbf{D}_{i,j}} = +1 : \forall j\}$;
  `// sub example d groups from` $|\mathcal{L}^{(i)}|$ `groups`
  $b_{sub} \leftarrow$ randomly select d groups from $|\mathcal{L}^{(i)}|$;
  $\mathbf{H}^{(i)} = 0 (\in \mathbb{R}^{\tau \times b})$;
  for $q \leftarrow 1$ **to** τ **do**
    for $j \leftarrow 1$ **to** b_{sub} **do**
      if $j \in \mathcal{L}^{(i)}$ **then**
        continue;
      $\mathtt{tmp}_1 = 0$;
      for $e \leftarrow 1$ **to** $|\mathcal{L}^{(i)}|$ **do**
        $\mathtt{tmp}_2 = 0$;
        for $k \leftarrow 1$ **to** $|B_j|$ **do**
          $p_j = \frac{1}{1+e^{-\Theta_j^{\mathbf{g},\mathbf{T}} \left| \tilde{\mathbf{c}}_j^{(i)} - \mathbf{P}_k \right|}}$;
          $p_k = \frac{1}{1+e^{-\Theta_k^{\mathbf{p},\mathbf{T}} \mathbf{x}^{(i)}}}$;
          $\mathtt{tmp}_2 = \mathtt{tmp}_2 + p_j p_k$;
        $\mathtt{tmp}_1 = \mathtt{tmp}_1 + z_{j,e} \times \mathtt{tmp}_2$;
      $\mathbf{H}_{q,j}^{(i)} = z_{j,e} \times \mathtt{tmp}_1$;
      if $q - 1 > 0$ **then**
        $\mathbf{H}_{q,j}^{(i)} = \mathbf{H}_{q-1,j}^{(i)} \times \mathbf{H}_{q,j}^{(i)}$;
    $A = \textsc{Avg}(\mathbf{H}^{(i)})$;
    $G = 1 - A$;
    $\mathbf{Q}^{(i)} = \frac{A.G+\epsilon}{Z}$;
    for $j \leftarrow 1$ **to** b_{sub} **do**
      if $\mathbf{Q}_j^{(i)} \geq \zeta$ **then**
        $\mathcal{L}^{(i)} = \mathcal{L}^{(i)} \oplus j$;
      else
        $\mathcal{L}^{(i)} = \mathcal{L}^{(i)} \ominus j$;
  `// the` *`transform`* `function is a simple operation to translate groups`
  `// from` $\mathcal{L}^{(i)}$ `groups into +1, -1 indicating presence/absence of groups`
  $\widehat{\mathbf{D}_i^{\mathbf{opt}}} \leftarrow \mathtt{transform}(\mathcal{L}^{(i)})$;
Return $\widehat{\mathbf{D}^{\mathbf{opt}}}$

Algorithm 3: Relabel2Group(n, $\mathbf{X}$, $\mathbf{Y}$, $\widehat{\mathbf{D}}$, $\mathcal{B}$, $\mathbf{P}$, $\Theta^{\mathbf{g}}$, $\Theta^{\mathbf{p}}$, $\mathbf{C}$, z, α, d, ϵ, v, ζ, τ)

A.2 Feed-Backward Phase

Here, we set up the learning framework for computing reMap's group and pathway parameters, jointly denoted as $\Theta = \{\Theta^{\mathbf{g}}, \Theta^{\mathbf{p}}\}$. From Eq. A.3, three learning components can be identified: i)- a hyper-plane in the group space to absorb group correlation, ii)- a hyper-plane in the pathway space to encode semantic information about pathways, and iii)- a joint learning between groups and pathways to exploit pathway-group relationship. Let us define three empirical loss functions, corresponding the three components: $\epsilon^{\mathbf{g}} : \{0,1\}^b \rightarrow \mathbb{R}_{\geq 0}$, $\epsilon^{\mathbf{p}} : \{0,1\}^t \rightarrow \mathbb{R}_{\geq 0}$, and $\epsilon^{\mathbf{gp}} : \{0,1\}^b \rightarrow \mathbb{R}_{\geq 0}$ of margin $\mathbf{d}h^{\mathbf{g}}(\mathbf{x})$, $\mathbf{y}h^{\mathbf{p}}(\mathbf{x})$, and $\mathbf{d}h^{\mathbf{gp}}(y)$, respectively, where $h^{(\cdot)}$ are decision functions. The last two loss functions are based on the logistic loss while the first loss is a sum of the two other losses. Now, to compute Θ, we maximize the posterior probability of Eq. A.4:

$$\begin{aligned} \widehat{\Theta} = \underset{\Theta}{\operatorname{argmax}} \prod_{q=1}^{q=\tau} \prod_{i=1}^{i=n} \mathbf{H}_{q-1}^{(i)} \prod_{j=1}^{j=b} p(\widehat{\mathbf{D}_{i,j}} | \mathcal{L}_{q-1}^{(i)}, \mathbf{x}^{(i)}) \\ \times \left(\sum_{s \in \mathbf{B}_{j,s}=+1} p(\widehat{\mathbf{D}_{i,j}} | l_s = +1, \Theta_j^{\mathbf{g}}) p(y_s^{(i)} | \mathbf{x}^{(i)}, \Theta_s^{\mathbf{p}}) \right) \Bigg) \end{aligned} \tag{A.5}$$

Estimation of parameters in Eq. A.5 is intractable due to the chain of probabilities $\mathbf{H}_{q-1}^{(i)}$ and the two marginalizations over $\mathcal{L}_{q-1}$ and s. Hence, we propose the following two diagnoses: i)- conditional independence assumptions where the previous history values are independent given the most recent estimates and ii)- collapse the marginalization over $\mathcal{L}_{q-1}$ by choosing only the maximum correlation z, irrelevant to which groups were considered. These simplified treatments provide an efficient way to optimize the parameters, where we adopt the "one-vs-all" scheme learning for each group and pathway [52].

In addition, we apply four constraints to retrieve a good set of parameters: i)- similarity between groups and associated pathways; ii)- weights of pathways, in a group, should be close to each other; iii)- examples sharing similar pathways should share similar representations; and iv)- all reMap's parameters should not be too large or too small. These four constraints are important to allow smooth updates and mapping operations. Using these four constraints, the obtained pathway group dataset ($\widehat{\mathbf{D}^{\mathbf{opt}}}$), and the pathway data ($\mathbf{Y}$), our objective function is formulated according to:

$$\begin{aligned} \min_{\Theta^{\mathbf{g}}, \mathbf{W}, \mathbf{U}, \mathbf{S}} & - \sum_{q \in \tau} \sum_{i \in n} \sum_{j \in b} \sum_{k \in \mathbf{B}_{j,k}=+1} v_i \log \left(p(\widehat{\mathbf{D}_{i,j}} | l_k = +1, \Theta_j^{\mathbf{g}}) \right) \\ & + \sum_{q \in \tau} \sum_{i \in n} ||\mathbf{y}^{(i)} - \widehat{\mathbf{D}_i} \mathbf{W}||_2^2 + \sum_{j \in b} C(\Theta_j^{\mathbf{g}}) \\ \min_{\Theta^{\mathbf{p}}, \mathbf{W}, \mathbf{S}} & - \sum_{q \in \tau} \sum_{i \in n} \sum_{k \in t} v_i \log \left(p(y_k^{(i)} | \mathbf{x}^{(i)}, \Theta_k^{\mathbf{p}}) \right) \\ & + \sum_{q \in \tau} \sum_{i \in n} ||\mathbf{y}^{(i)} - \widehat{\mathbf{D}_i} \mathbf{W}||_2^2 + \sum_{k \in t} C(\Theta_k^{\mathbf{p}}) \end{aligned} \tag{A.6}$$

where,

$$
\begin{aligned}
v^{(i)} &= p(\mathbf{x}^{(i)}|\mathbf{H}^{(i)}_{q-1}, \mathcal{L}^{(i)}_{q-1}, \widehat{\mathbf{D}_{i,j}} = +1) \\
-\log\left(p(\widehat{\mathbf{D}_{i,j}}|l_k = +1, \Theta^{\mathbf{g}}_j)\right) &= \log\left(1 + e^{-\mathbf{d}^{(i)}_j \Theta^{\mathbf{g}\intercal}_j |\tilde{\mathbf{c}}^{(i)}_j - \mathbf{P}_k|}\right) \\
&\triangleq \epsilon^{\mathbf{gp}}(\mathbf{d}^{(i)}_j h^{\mathbf{gp}}_j(l^{(i)}_k), \Theta^{\mathbf{g}}_j) \\
-\log p(\mathbf{y}^{(i)}_k|\mathbf{x}^{(i)}, \Theta^{\mathbf{p}}_k) &= \log\left(1 + e^{-\mathbf{y}^{(i)}_k \Theta^{\mathbf{p}\intercal}_k \mathbf{x}^{(i)}}\right) \\
&\triangleq \epsilon^{\mathbf{p}}(\mathbf{y}^{(i)}_k h^{\mathbf{p}}_k(\mathbf{x}^{(i)}), \Theta^{\mathbf{p}}_k) \\
l^{\mathbf{g}}(\mathbf{d}^{(i)}_j h^{\mathbf{g}}_j(\mathbf{x}^{(i)}), \Theta^{\mathbf{g}}_j) &\triangleq \epsilon^{\mathbf{gp}}(\mathbf{d}^{(i)}_j h^{\mathbf{gp}}_j(l^{(i)}_k), \Theta^{\mathbf{g}}_j) \\
&\quad + \sum_{k \in \mathbf{B}_{j,k}=+1} \epsilon^{\mathbf{p}}(\mathbf{y}^{(i)}_k h^{\mathbf{p}}_k(\mathbf{x}^{(i)}), \Theta^{\mathbf{p}}_k)
\end{aligned}
\tag{A.7}
$$

$$
\begin{aligned}
C(\Theta^{\mathbf{g}}_j) = &\underbrace{\sum_{k \in \mathbf{B}_{j,k}=+1} ||\mathbf{U}^{\intercal}\Theta^{\mathbf{p}}_k - \Theta^{\mathbf{g}}_j||^2_2}_{\text{pathways within a group}} + \underbrace{\frac{\lambda_1}{2} \sum_{q,l \in n} \mathbf{S}_{q,l} ||\widehat{\mathbf{d}^{(q)}} - \widehat{\mathbf{d}^{(l)}}||^2_2}_{\text{correlated groups}} \\
&+ \lambda_2 ||\Theta^{\mathbf{g}}_j||_{2,1} + \lambda_3 ||\mathbf{U}||_{2,1} + \frac{1}{2}\kappa ||\mathbf{S}1 - 1||^2_2
\end{aligned}
\tag{A.8}
$$

$$
\begin{aligned}
C(\Theta^{\mathbf{p}}_k) = &\underbrace{\sum_{q \in \mathbf{B}_{j,q}=+1} ||\Theta^{\mathbf{p}}_q - \Theta^{\mathbf{p}}_k||^2_2}_{\text{pathways closeness}} + \underbrace{\frac{\lambda_4}{2} \sum_{q,l \in n} \mathbf{S}_{q,l} ||\mathbf{y}^{(q)} - \mathbf{y}^{(l)}||^2_2}_{\text{correlated pathways}} \\
&+ \underbrace{\frac{1}{2} \sum_{q,l \in n} \mathbf{S}_{q,l} ||\Theta^{\mathbf{p}\intercal}_k \mathbf{x}^{(q)} - \Theta^{\mathbf{p}\intercal}_k \mathbf{x}^{(l)}||^2_2}_{\text{correlated examples of a pathway}} \\
&+ \lambda_5 ||\Theta^{\mathbf{p}}_k||_{2,1} + \frac{1}{2}\kappa ||\mathbf{S}1 - 1||^2_2
\end{aligned}
\tag{A.9}
$$

where $||.||^2_2$ represents the squared L_2 norm, $||.||^2_{2,1}$ is the sum of the Euclidean norms of columns of a matrix, $v^{(i)}$ is the weight of a example $\mathbf{x}^{(i)}$ to emphasize selection of informative examples, and $\lambda_{[1,2,3,4,5]} \in \mathbb{R}$ are hyper-parameters controlling the relative contributions of the associated constraint terms. Let us explain all the terms involved in Eqs A.7-A.9. The function $||\mathbf{U}^{\intercal}\Theta^{\mathbf{p}}_k - \Theta^{\mathbf{g}}_j||^2_2$ reflects the first constraint, where it enforces similarities between pathways, associated to a group j, and the pathway group j itself. $\mathbf{U} \in \mathbb{R}^{r \times m}$ is the linear transformation matrix from r onto m dimensional space. For the second constraint, the term $||\Theta^{\mathbf{p}}_q - \Theta^{\mathbf{p}}_k||^2_2$ considers the similarities among pathways, grouped under a specific pathway group. To adopt the third constraint, we used four terms: $||\mathbf{y}^{(i)} - \widehat{\mathbf{D}_i}\mathbf{W}||^2_2$, $||\widehat{\mathbf{d}^{(q)}} - \widehat{\mathbf{d}^{(l)}}||^2_2$, $||\mathbf{y}^{(q)} - \mathbf{y}^{(l)}||^2_2$, and $||\Theta^{\mathbf{p}\intercal}_k \mathbf{x}^{(q)} - \Theta^{\mathbf{p}\intercal}_k \mathbf{x}^{(l)}||^2_2$.

The term $||\mathbf{y}^{(i)} - \widehat{\mathbf{D}_i}\mathbf{W}||^2_2$ maintains the integrity of both pathway group and pathway vectors on example i, thus, encouraging groups to have similar contents as pathways, and $\mathbf{W} \in \mathbb{R}^{b \times t}$ captures the correlation between groups

and pathways. Both $||\widehat{\mathbf{d}^{(q)}} - \widehat{\mathbf{d}^{(l)}}||_2^2$ and $||\mathbf{y}^{(q)} - \mathbf{y}^{(l)}||_2^2$ describes the resemblances between the two pathway group vectors, $\widehat{\mathbf{d}^{(q)}}$ and $\widehat{\mathbf{d}^{(l)}}$, and the two pathway vectors, $\mathbf{y}^{(q)}$ and $\mathbf{y}^{(l)}$, suggesting the similarity between input instances $\mathbf{x}^{(q)}$ and $\mathbf{x}^{(k)}$. The similarity scores of the aforementioned instances are captured by $\mathbf{S}_{q,k} \in \mathbb{R}_{\geq 0}^{n \times n}$, where a high score, indicates both examples have near identical pathways and, hence, should have similar groups, and vice-versa hold as well.

The formula $||\Theta_k^{\mathbf{p}\intercal}\mathbf{x}^{(q)} - \Theta_k^{\mathbf{p}\intercal}\mathbf{x}^{(l)}||_2^2$ addresses the neighborhood relationship in the example feature space between $\mathbf{x}^{(q)}$ and $\mathbf{x}^{(k)}$, as characterized by $\mathbf{S}_{q,l}$ score [50]. As discussed before, if two instances are close to each other then they may possess relevant labels, which leads to relabeling a dataset with a proper subset of groups, hence, mitigating from the negative influences of imperfectly labeling groups. The terms $||\Theta_j^{\mathbf{g}}||_{2,1}$, $||\Theta_j^{\mathbf{p}}||_{2,1}$, and $||\mathbf{U}||_{2,1}$, constitute the fourth constraint that aim to shrink weights and perform feature selection. Finally, $\kappa||\mathbf{S}1 - 1||_2^2$ enforces equality constraint such that $\forall q, \sum_{k \in n} \mathbf{S}_{q,l} = 1$, where κ is a Lagrange multiplier, and 1 denotes a column vector with all of it's elements are equal to 1.

Taken together, the trainable parameters of reMap are: 1)- group-projection weight matrix $\mathbf{W}$, 2)- pathway-projection weight matrix $\mathbf{U}$, 3)- group-specific weight matrix $\Theta^{\mathbf{g}}$, 4)- pathway-specific weight matrix $\Theta^{\mathbf{p}}$, 5)- example-similarity specific weight matrix $\mathbf{S}$, and 6)- group-specific updating matrix $\widehat{\mathbf{D}}$. The last parameter is a binary matrix indicating the presence/absence of groups in the training dataset, which is gradually updated based on the gradient score strategy.

Unfortunately, the objective function in Eq. A.6 involves $L_{2,1}$-norm that is non-smooth and difficult to be solved, instead we perform iterative gradient descent method for reMap which alternatively optimizes over one of six classes of variables ($\mathbf{W}, \mathbf{U}, \Theta^{\mathbf{g}}, \Theta^{\mathbf{p}}, \mathbf{S}$, and $\widehat{\mathbf{D}}$) at a time while the others are held constant. The partial derivative of each term in Eq. A.6 is a positive semi-definitive, hence, the whole term is jointly convex, which leads to the following independent optimization problems for all pathways and groups classifiers according to the multi-label 1-vs-All approach [52].

- **Update W**. The gradient of Eq. A.6 w.r.t. $\mathbf{W}$ has the following formula:

$$\nabla \mathbf{W} = \frac{2}{nb}\left(\widehat{\mathbf{D}}^{\intercal}\widehat{\mathbf{D}}\mathbf{W} - \widehat{\mathbf{D}}^{\intercal}\mathbf{y}\right) + \lambda_3 \mathbf{K}_{\mathbf{W}}\mathbf{W} \tag{A.10}$$

where $\mathbf{K}_{\mathbf{W}} = \begin{pmatrix} \frac{1}{2||\mathbf{W}^1||_2} & & \\ & \ddots & \\ & & \frac{1}{2||\mathbf{W}^b||_2} \end{pmatrix}$

- **Update U**. The gradient of Eq. A.6 w.r.t. $\mathbf{U}$ becomes:

$$\begin{aligned} \nabla \mathbf{U} = & \frac{1}{b}\sum_{j \in b} \frac{2}{\sum_k \mathbb{I}(\mathbf{B}_{j,k} = +1)} \sum_{k \in \mathbf{B}_{j,k}=+1} \left(\Theta_k^{\mathbf{p}}\Theta_k^{\mathbf{p}\intercal}\mathbf{U} - \Theta_k^{\mathbf{p}}\Theta_j^{\mathbf{g}\intercal}\right) \\ & + \lambda_3 \mathbf{K}_{\mathbf{U}}\mathbf{U} \end{aligned} \tag{A.11}$$

where $\mathbf{K_U} = \begin{pmatrix} \frac{1}{2||\mathbf{U}^1||_2} & & \\ & \ddots & \\ & & \frac{1}{2||\mathbf{U}^r||_2} \end{pmatrix}$

– **Update** $\Theta^{\mathbf{g}}$. The partial derivative for each pathway group, say $\Theta^{\mathbf{g}}_j$, is:

$$\begin{aligned}\nabla\Theta^{\mathbf{g}}_j =& \frac{1}{n}\sum_{i\in n} v_i \left(\frac{1}{\sum_k \mathbb{I}(\mathbf{B}_{j,k}=+1)} \sum_{k\in\mathbf{B}_{j,k}=+1} \frac{-\widehat{\mathbf{D}_{i,j}}|\tilde{\mathbf{c}}^{(i)}_j - \mathbf{P}_k|}{1+e^{\widehat{\mathbf{D}_{i,j}}\Theta^{\mathbf{g}\intercal}_j|\tilde{\mathbf{c}}^{(i)}_j-\mathbf{P}_k|}} \right) \\ &+ \frac{1}{\sum_k \mathbb{I}(\mathbf{B}_{j,k}=+1)} \sum_{k\in\mathbf{B}_{j,k}=+1} \Big(-2\widehat{\mathbf{U}}^{\intercal}\Theta^{\mathbf{p}}_k + 2\Theta^{\mathbf{g}}_j \Big) + \lambda_2 \frac{\Theta^{\mathbf{g}}_j}{2||\Theta^{\mathbf{g}}_j||_2}\end{aligned} \tag{A.12}$$

where $\widehat{\mathbf{U}}$ obtained from Eq. A.11.

– **Update** $\Theta^{\mathbf{p}}$. The partial derivative w.r.t one pathway k of $\Theta^{\mathbf{p}}$ with the new $\widehat{\mathbf{U}}$ and $\widehat{\Theta^{\mathbf{g}}}$ updates has the following form:

$$\begin{aligned}\nabla\Theta^{\mathbf{p}}_k =& \frac{1}{n}\sum_{i=1}^{i=n} v_i \left(\frac{-\mathbf{y}^{(i)}_k \mathbf{x}^{(i)}}{1+e^{\mathbf{y}^{(i)}_k \Theta^{\mathbf{p}\intercal}_k \mathbf{x}^{(i)}}} \right) + 2\widehat{\mathbf{U}}\widehat{\mathbf{U}}^{\intercal}\Theta^{\mathbf{p}}_k - \frac{2}{b}\sum_{j\in b}\widehat{\mathbf{U}}\widehat{\Theta}^{\mathbf{g}}_j \\ &+ \frac{1}{b}\sum_{j\in b} \frac{2}{\sum_k \mathbb{I}(\mathbf{B}_{j,k}=+1)} \sum_{q\in\mathbf{B}_{j,q}=+1} \Big(\Theta^{\mathbf{p}}_k - \Theta^{\mathbf{p}}_q\Big) \\ &+ \mathbf{X}^{\intercal}\mathbf{L}\mathbf{X}\Theta^{\mathbf{p}}_k + \lambda_5 \frac{\Theta^{\mathbf{p}}_k}{2||\Theta^{\mathbf{p}}_k||_2}\end{aligned} \tag{A.13}$$

where $\mathbf{L} \triangleq \mathbf{M} - \mathbf{S}$ is the graph Laplacian matrix and $\mathbf{M}$ is a diagonal matrix with $\mathbf{M}_{j,j} = \sum_{k=1} \mathbf{S}_{j,k}$. Note that $\frac{1}{2}\sum_{q,l\in n} \mathbf{S}_{q,l}||\Theta^{\mathbf{p}\intercal}_k(\mathbf{x}^{(q)} - \mathbf{x}^{(l)})||^2_2 = \mathbf{tr}\big(\Theta^{\mathbf{p}\intercal}_k \mathbf{X}^{\intercal}\mathbf{L}\mathbf{X}\Theta^{\mathbf{p}}_k\big)$. Following the work of [44], it is important for practical purpose to normalize the graph Laplacian, to account for the fact that some examples are more similar than others [48]: $\bar{\mathbf{L}} \triangleq \mathbf{M}^{-1/2}\mathbf{L}\mathbf{M}^{-1/2} = \mathbf{I} - \mathbf{M}^{-1/2}\mathbf{S}\mathbf{M}^{-1/2}$. Adhering to this property, we consider the following formula: $\frac{1}{2}\sum_{q,l\in n} \mathbf{S}_{q,l}||\Theta^{\mathbf{p}\intercal}_k\big(\frac{\mathbf{x}^{(q)}}{\sqrt{\mathbf{M}_{q,q}}} - \frac{\mathbf{x}^{(l)}}{\sqrt{\mathbf{M}_{l,l}}}\big)||^2_2 = \mathbf{tr}\big(\Theta^{\mathbf{p}\intercal}_k \mathbf{X}^{\intercal}\bar{\mathbf{L}}\mathbf{X}\Theta^{\mathbf{p}}_k\big)$.

– **Update** $\mathbf{S}$. Given the updated values of $\widehat{\mathbf{U}}$ and $\widehat{\Theta}^{\mathbf{p}}$, we obtain the equivalent objective function of Eq. A.6 with the terms only related to $\mathbf{S}$ as:

$$\min_{0\le\mathbf{S}_{q,j}\le 1} \lambda_1 \mathrm{tr}(\widehat{\mathbf{D}}^{\intercal}\mathbf{L}\widehat{\mathbf{D}}) + \lambda_4 \mathrm{tr}(\mathbf{Y}^{\intercal}\mathbf{L}\mathbf{Y}) + \mathrm{tr}(\widehat{\Theta^{\mathbf{p}}}\mathbf{X}^{\intercal}\mathbf{L}\mathbf{X}\widehat{\Theta^{\mathbf{p}\intercal}}) + \kappa||\mathbf{S}1 - 1||^2_2 \tag{A.14}$$

For the inequality constraint, during iterative updates we force values of $\mathbf{S}$ to be within the range of $[0, 1]$. Then, the gradient update can be written as:

$$\nabla\mathbf{S} = \lambda_1\widehat{\mathbf{D}}\widehat{\mathbf{D}}^{\intercal} + \lambda_4\mathbf{Y}\mathbf{Y}^{\intercal} + \mathbf{X}\widehat{\Theta^{\mathbf{p}\intercal}}\widehat{\Theta^{\mathbf{p}}}\mathbf{X}^{\intercal} + 2\kappa(\mathbf{S} - 1) \tag{A.15}$$

As we have mentioned, the similarity matrix $\mathbf{S}$ captures reliable and discriminative locality information in the projected example feature space, and

this information is utilized to optimize correlations in the predicted pathway space, which ensures example-pathway space consistency. Consequently, a set of groups can be inferred with high fidelity, for each example, if features with labels correlation information is disseminated to features with groups correlations, thus, alleviating the effects of imperfectly detecting negative groups.

– **Update $\widehat{\mathbf{D}}$**. We iteratively update groups, where a set of groups are added or removed at each round. In particular, a positive subset of groups is selected $\mathcal{B}_{\mathbf{P}}^{(i)} \subseteq \arg\{\widehat{\mathbf{D}_{i,j}} = +1 : \forall j\}$, for each example $\mathbf{x}^{(i)}$, and the remaining groups are considered to be negative to that example. While it is relatively easy to compile a set of positive groups for an example, however, groups not belonging to that example are too diverse to be considered as negative. Thus, it is better to consider the remaining groups as unassigned $\mathcal{B}_{\mathbf{U}}^{(i)}$. We use the gradient score strategy, where the values of $\widehat{\mathbf{D}}$ is updated based on the gradient score according to:

$$\begin{aligned}\nabla\widehat{\mathbf{D}} =& \frac{1}{n}\sum_{i\in n} v_i \sum_{j\in b}\left(\frac{1}{\sum_k \mathbb{I}(\mathbf{B}_{j,k}=+1)}\sum_{k\in\mathbf{B}_{j,k}=+1}\frac{-\Theta_j^{\mathbf{g}\intercal}|\tilde{\mathbf{c}}_j^{(i)}-\mathbf{P}_k|}{1+e^{\widehat{\mathbf{D}_{i,j}}\Theta_j^{\mathbf{g}\intercal}|\tilde{\mathbf{c}}_j^{(i)}-\mathbf{P}_k|}}\right)\\ &+\lambda_1\mathbf{L}^\intercal\widehat{\mathbf{D}}+\frac{2}{nb}\left(\widehat{\mathbf{D}}\mathbf{W}\mathbf{W}^\intercal-\lambda_3\mathbf{y}\mathbf{W}^\intercal\right)\end{aligned} \tag{A.16}$$

After getting the gradient score, we assign groups to examples based on:

$$\widehat{\mathbf{D}_{i,j}} = \begin{cases} +1 & \text{if } \nabla\widehat{\mathbf{D}_{i,j}} \geq 1 \\ 0 & \text{if } 0 < \nabla\widehat{\mathbf{D}_{i,j}} < 1 \\ -1 & \text{if } \nabla\widehat{\mathbf{D}_{i,j}} \leq 1 \end{cases} \tag{A.17}$$

where $\widehat{\mathbf{D}_{i,j}} = +1$ (resp. -1 and 0) means the group is selected to be positive (resp. negative and unknown) given a training example i. Having acquired a new selected set of groups for each instance, we update $\mathcal{L}^{(i)}$ accordingly.

The pseudocode for this phase in presented in Algorithm 4

A.3 Closing the loop

The two phases are repeated for all examples in a given pathway data, until a predefined number of rounds τ ($\in \mathbb{Z}_{>1}$) is reached. At the end, a pathway group dataset is produced which consists of n examples with the assigned groups, i.e., $\widehat{\mathbf{D}^{\mathbf{opt}}}$. This data can be used as inputs to a pathway predictor to perform pathway prediction for a newly sequenced genome.

Inputs :
n: number of examples ($n \in \mathbb{N}_{>2}$)
$\mathbf{X}$: input space training set ($\mathbf{X} \in \mathbb{R}^{n \times r}$)
$\mathbf{Y}$: pathway space training set ($\mathbf{Y} \in \mathbb{Z}_{\geq 0}^{n \times t}$)
$\widehat{\mathbf{D}}$: pathway group space training set ($\widehat{\mathbf{D}} \in \mathbb{Z}_{\geq 0}^{n \times b}$)
$\mathbf{P}$: pathway features matrix ($\mathbf{P} \in \mathbb{R}^{t \times m}$)
$\mathbf{C}$: the centroids of groups ($\mathbf{C} \in \mathbb{R}^{b \times m}$)
z: normalized groups's correlation ($z \in \mathbb{R}^{b \times b}$)
d: a subexample size hyper-parameter ($d \in \mathbb{N}_{>1}$)
ξ: number of epochs ($\xi \in \mathbb{N}$)
`// for brevity, the collection of all hyperparamters`
`// is represented as` λ
λ: a set of all hyperparameters, including cut-off thresholds
γ: learning rate ($lr \in \mathbb{R}_{>0}$)

Outputs:
$\Theta^{\mathbf{g}}$: groups' parameters ($\Theta^{\mathbf{g}} \in \mathbb{R}^{b \times m}$)
$\Theta^{\mathbf{p}}$: pathways' parameters ($\Theta^{\mathbf{p}} \in \mathbb{R}^{t \times r}$)
$\mathbf{W}$: group-projection parameters ($\mathbf{W} \in \mathbb{R}^{b \times t}$)
$\mathbf{U}$: pathway-projection parameters ($\mathbf{W} \in \mathbb{R}^{r \times m}$)
$\mathbf{S}$: instance-similarity specific parameters ($\mathbf{S} \in \mathbb{R}^{n \times n}$)

Process :
`// groups' parameters (`$\Theta^{\mathbf{g}} \in \mathbb{R}^{b \times m}$`)`
$\Theta^{\mathbf{g}} \leftarrow 0$;
`// pathways' parameters (`$\Theta^{\mathbf{p}} \in \mathbb{R}^{t \times r}$`)`
$\Theta^{\mathbf{p}} \leftarrow 0$;
for $q \leftarrow 1$ **to** ξ **do**
  $\widehat{\mathbf{D}^q} \leftarrow$ Relabel2Group(n, $\mathbf{X}$, $\mathbf{Y}$, $\widehat{\mathbf{D}^{q-1}}$, $\mathcal{B}$, $\mathbf{P}$, $\widehat{\Theta^{\mathbf{g},q-1}}$, $\widehat{\Theta^{\mathbf{p},q-1}}$, $\mathbf{C}$, z, λ);
  `// update W's parameters using Eq. A.10`
  $\mathbf{W}^q \leftarrow \mathbf{W}^{q-1} - \gamma \nabla \mathbf{W}^q$;
  `// update U's parameters using Eq. A.11`
  $\mathbf{U}^q \leftarrow \mathbf{U}^{q-1} - \gamma \nabla \mathbf{U}^q$;
  `// update` $\Theta^{\mathbf{g}}$`'s parameters using Eq. A.12`
  $\Theta^{\mathbf{g},q} \leftarrow \Theta^{\mathbf{g},q-1} - \gamma \nabla \Theta^{\mathbf{g},q}$;
  `// update` $\Theta^{\mathbf{p}}$`'s parameters using Eq. A.13`
  $\Theta^{\mathbf{p},q} \leftarrow \Theta^{\mathbf{p},q-1} - \gamma \nabla \Theta^{\mathbf{p},q}$;
  `// update S's parameters using Eq. A.15`
  $\mathbf{S}^q \leftarrow \mathbf{S}^{q-1} - \gamma \nabla \mathbf{S}^q$;
  `// update` $\widehat{\mathbf{D}}$ `using gradient score strategy`
  $\widehat{\mathbf{D}^q} \leftarrow$ apply Eq. A.17;
Return $\Theta^{\mathbf{g}}$, $\Theta^{\mathbf{p}}$, $\mathbf{W}$, $\mathbf{U}$, $\mathbf{S}$

Algorithm 4: Backward(n, $\mathbf{X}$, $\mathbf{Y}$, $\widehat{\mathbf{D}}$, $\mathbf{P}$, $\mathbf{C}$, z, d, ξ, λ, γ)

B Correlated Models

We present three correlated pathway models that can be applied during pathway group construction step in the feed-forward phase of reMap: i)-CTM (correlated

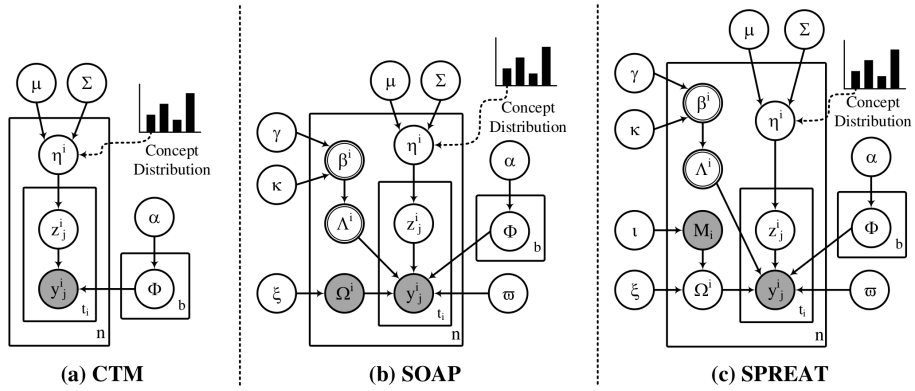


Fig. 6: Graphical model representation of the correlated concept models. The boxes are "plates" representing replicates. The outer plate represents instances, while the inner plate represents the repeated choice of features within an example. The logistic normal distribution, used to model the latent concept proportions of an example, captures correlations among concepts that are impossible to capture using a single Dirichlet. The observed data for each example $\mathbf{x}_i$ are a set of annotated features $\mathbf{y}_i$ and a set of hypothetical features $\mathbf{M}_i$. The hidden variables are: per-example concept proportions η_i, per-example concept selection parameters Λ_i, per-example hypothetical feature distributions Ω_i, per-feature concept assignment $z_{i,j}$, per-concept distribution over features Φ_a, and per-feature indicator parameter $d_{i,j}$.

topic model) [23], ii)- SOAP (sparse correlated pathway group) and iii)- SPREAT (distributed sparse correlated pathway group). These models incorporate pathway abundance information to encode each example as a mixture distribution of groups, and each pathway group, in turn, is a mixture of pathways with different mixing proportions. The pathway abundance information can be obtained by mapping enzyme –with abundances– onto the reference pathway database (e.g. MetaCyc). Before we discuss these three models, first let us provide some background information and notations. We note that each mathematical symbol is only related in the context of this section.

Definition 6. ***Pathway Collection***. *Let $\mathcal{P} = \{\mathbf{y}^{(i)} : 1 < i \leqslant n\}$ be a collection of n examples, where each example $\mathbf{y}^{(i)} = (y_1^{(i)}, y_1^{(i)}, \ldots, y_t^{(i)})$ is a vector encoding the unnormalized abundance information of pathways and t is the pathway size. Let $\mathcal{Y} = \{h_1, h_2, \ldots, h_t\}$ be a set of all known metabolic pathways obtained from a trusted source (e.g., MetaCyc [28]), and $\mathcal{Y}_i \subseteq \mathcal{Y}$ corresponds to a subset of true pathways associated with the example i.* ■

Recovering latent distributions of $\mathcal{P}$ mirrors the concept modeling paradigm, which aims to reconstruct the thematic structure, called "topics", from a corpus [25].

Definition 7. ***Concept Modeling***. *Given a collection of n examples, a concept distribution for i-th example is a multinomial distribution vector, denoted by $\eta^{(i)}$ of size b concepts, i.e., $\{p(\Phi_a|\eta^{(i)})\}_{a=1}^{a=b}$, where Φ_j in a multinomial feature distribution over the concept j, i.e., $\{p(y_k|\Phi_j)\}_{k=1}^{k=t}$. The overall goal of concept modeling is to recover the b salient concepts of each example.* ■

In this paper, the term concept is referred to as "pathway group" or "group". For brevity purposes, the following terms: *concept*, *topic*, or *pathway group*, are used interchangeably. Also, *features* correspond to *pathways*.

The classical studies in concept modeling attempt to discover concepts from a collection of examples that are composed of features, as in the case of latent Dirichlet allocation (LDA) [25]. However, this approach neglects dependencies among concepts. We take advantage of the inherent thematic structure of examples and model the concept dependencies to extract the concept distributions of examples.

Definition 8. ***Concept Correlation***. *Given $\mathcal{P}$, the pairwise concept-correlation is defined by a Gaussian covariance matrix, denoted by Σ. Each entry $s_{i,j}$ in Σ characterizes the i-th pathway group association with the pathway group j, where a larger score indicates both concepts are highly correlated.* ■

However, there exist situations where a set of pathways may not be included in $\mathcal{P}$ because $\mathcal{P}$ has high noise. An alternative way to incorporate missing pathways is to store these pathway in a separate list while keeping the original pathway collection intact for further investigation. Lets us denote $\mathbf{M} \in \mathbb{Z}_{\geq 0}^{n \times t}$ a matrix holding a set of missing pathways where each entry is an integer value indicating the abundance of a pathway in an example. Here, this matrix is referred to as a "background" or "supplementary" matrix, analogous to studies in [32, 53]. With these definitions, we describe the correlated models.

B.1 Correlated Topic Model

The correlated topic model (CTM) is a probabilistic graphical model that extends the generative story of LDA [25] to incorporate correlation among concepts. Fig. 6a shows the Bayesian graphical model for CTM using plate notation. Like latent Dirichlet allocation [25], the CTM is comprised of a hierarchical Bayesian mixture model, where features (words as described in the original paper) are mixed to constitute concepts. And, the concepts are assumed to be correlated to each other by a Gaussian covariance matrix.

Formally, let n be the total number of a collection, where each example i consists of features, i.e., $\mathbf{y}^{(i)}$. Then, the generative process for CTM is described as follows. First, we draw a multinomial feature distribution Φ_a from a Dirichlet prior $\alpha > \mathbb{R}_{>0}$ for each concept $a \in \{1, \ldots, b\}$. Then, for each example i, a Gaussian random variable is drawn $\eta^{(i)} \sim \mathcal{N}(\mu, \Sigma)$, where μ is a b dimensional mean and $\Sigma \in \mathbb{R}^{b \times b}$ is the covariance matrix. The random variable $\eta^{(i)}$ is projected onto the probability simplex to obtain the concept distributions $\theta^{(i)} = \text{softmax}(\eta^{(i)})$, corresponding the logistic-normal distribution, from which

a concept indicator $z_j^{(i)} \in \{1, \ldots, b\}$ is sampled. Finally, each observed feature $j \in \{1, \ldots, t^i\}$ is drawn from the associated feature distribution, indicated by it's concept assignment, i.e., $y_j^{(i)} \sim \Phi_{z_j^{(i)}}$. This generative process is outlined in Algorithm 5, which can be observed that the process is identical to LDA except the concept distributions is sampled from the logistic normal rather than a Dirichlet prior.

1. For each concept $a \in \{1, \ldots, b\}$:
   (a) example a distribution over features $\Phi_a \sim \mathrm{Dir}(.|\alpha)$;
2. For each example $i \in \{1, \ldots, n\}$:
   (a) Draw the example concept weight $\eta^{(i)} \sim \mathcal{N}(.|\mu, \Sigma)$;
   (b) Draw concept proportions $\theta^{(i)} = \mathrm{softmax}(\eta^{(i)})$;
   (c) For each feature $j \in \{1, \ldots, t^{(i)}\}$:
      i. example a concept assignment $z_j^{(i)} \sim \mathrm{Mult}(.|\theta^{(i)})$;
      ii. example a feature $y_j^{(i)} \sim \mathrm{Mult}(.|\Phi_{z_j^{(i)}})$;

Algorithm 5: The generative process for CTM given a collection

B.2 Correlated Pathway-Group Model

Correlated pathway group models are extension to CTM (Figs 6b and c): i)- SOAP and ii)- SPREAT. Both models incorporate dual sparseness and supplementary pathways in modeling group proportions. These important properties are not adopted in CTM. Let us discuss these two models.

Analogous to CTM, given n number of examples and a matrix encoding the missing features $\mathbf{M}$, the generative process for SOAP and SPREAT can be described as follows. First, we draw a multinomial feature distribution Φ_a from asymmetric Dirichlet prior $\alpha \in \mathbb{R}_{>0}$ for each concept $a \in \{1, ..., b\}$, where b is assumed to be known and fixed in advance. The symmetric assumption is appropriate, in such a scenario, because our prior knowledge, associated with these features, is inaccessible. For each example i, a concept proportion is drawn $\theta^{(i)} = \mathrm{softmax}(\eta^{(i)})$, where $\eta^{(i)}$ is a Gaussian random variable with mean and covariance are denoted by μ and Σ, respectability.

To sample a concept, it is reasonable to expect that each example is usually explained with a handful set of a mixed proportion of concepts. Besides, a concept should cover a few focused features, instead of absorbing all features. Thus, we borrow the idea from [21,22,30,36,43] to enforce dual sparsity to retain those relevant focused concepts and features by: i)- introducing an auxiliary Bernoulli variable $\Lambda^{(i)}$ of size b to determine whether a concept is selected for an example i or ignored, and ii)- applying a cutoff threshold to retain top $k \ll t$ features for each concept. Instead of sampling each entry in $\Lambda^{(i)}$ directly from a Bernoulli coin toss, we assume that each entry is sampled from a Beta distribution $\beta^{(i)}$, parameterized by two hyperparameters $\gamma \in \mathbb{R}_{>0}$ and $\kappa \in \mathbb{R}_{>0}$. Applying this dual sparsity, we aim to enhance the interpretability of the learned concepts while reducing the negative correlation among concepts on Σ.

Next, a concept indicator $z_j^{(i)} \in \{1, ..., b\}$ is drawn according to the example-specific mixture proportion $\Lambda^{(i)} \odot \theta^{(i)}$, where $\odot$ represents the Hadamard product. Now each feature $y_j^{(i)}$ in example i is generated from a weighted distribution $\Omega^{(i)}_{z_j^{(i)}} \odot \Phi_{z_j^{(i)}}$, as indicated by it's concept assignment, using a smoothing prior $\varpi \in \mathbb{R}_{>0}$. The parameter $\Omega^{(i)} \in \mathbb{R}^t$, derived from $\mathbf{M}_i$, represents a normalized supplementary feature of size t, which is assumed to be drawn from a symmetric Dirichlet prior $\xi \in \mathbb{R}_{>0}$. For SPREAT, this parameter encodes distribution, where each element of $\Omega^{(i)}_j$ corresponds to the example's probability of using feature $y_j \in \mathbf{M}_i$. Here, the background feature is assumed to be drawn from a sparse binary vector prior $\iota \in \mathbb{R}_{>0}$ that is included for completeness because each example's feature $\mathbf{M}_i$ is already observed.

1. For each concept $a \in \{1, ..., b\}$:
   (a) example a distribution over features $\Phi_a \sim \text{Dir}(.|\alpha)$;
2. For each example $i \in \{1, ..., n\}$:
   (a) Draw the example concept weight $\eta^{(i)} \sim \mathcal{N}(.|\mu, \Sigma)$;
   (b) Draw concept proportions $\theta^{(i)} = \text{softmax}(\eta^{(i)})$;
   (c) Draw beta distribution $\beta^{(i)} \sim \text{Beta}(.|\gamma, \kappa)$;
   (d) Draw a sparsity indicator vector $\Lambda^{(i)} \sim \text{Bernoulli}(.|\beta^{(i)})$;
   if `SPREAT`:
      i. example a vector $\mathbf{M}_i \sim \text{Prior}(.|\iota)$;
      ii. example background distribution $\Omega^{(i)}|\mathbf{M}_i \sim \text{Dir}(.|\xi)$;
   else:
      i. Draw background feature proportions $\Omega^{(i)} \sim \text{Dir}(.|\xi)$;
   (e) For each feature $j \in \{1, \ldots, t^{(i)}\}$:
      i. example a concept assignment $z_j^{(i)} \sim \text{Mult}(.|\Lambda^{(i)} \odot \theta^{(i)})$;
      ii. example a feature $y_j^{(i)} \sim \text{Mult}(.|(1 - \Omega^{(i)}_{z_j^{(i)}}) \odot \Phi_{z_j^{(i)}})$;

Algorithm 6: The generative process for SOAP and SPREAT

Representing SOAP and SPREAT as layer-wise mixing components supports the hierarchical modularity of metabolic pathway generation, where the components of one level (e.g., features) permit to contribute to other structures with different degrees of granularity. The generative process of SOAP and SPREAT models is summarized in Algorithm 6. Note that by setting all entries in Ω, Λ, and ϖ to 1, SOAP and SPREAT are reduced to CTM ("collapse2ctm" or c2m), which is an additional benefit to the these models.

B.3 Evidence Lower Bound (ELBO) for SPREAT

Here, we discuss the inference for the SPREAT model. Similar expression is straightforward to derive for SOAP. Given $\mathcal{P}$, the goal of inference is to compute the posterior distribution of the per-example concept proportions $\eta^{(i)}$, the per-example concept selection parameters $\Lambda^{(i)}$ and the associated beta distributions $\beta^{(i)}$, the per-example background feature distributions $\Omega^{(i)}$, the per-feature concept assignment $z_j^{(i)}$, and the per-concept distribution over features Φ_a.

Table 3: Correspondence between variational and original parameters.

Original parameter	Φ	μ	Σ	Λ	Ω	z
Variational parameter	ϕ	ν	ζ^2	λ	ω	ς

Looking at the topology of the Bayesian network, we can specify the complete-data likelihood, i.e., the joint distribution of all observed and latent variables given the hyperparameters and sparse supplementary feature matrix following the model's independence assumptions:

$$p(z, y, \eta, \Phi, \Lambda, \beta, \Omega | \mathbf{M}, \gamma, \kappa, \alpha, \iota, \xi, \beta) = \left[\prod_{a=1}^{b} p(\Phi_a | \alpha) \right] \left[\prod_{i=1}^{n} p(\eta | \mu, \Sigma) p(\Lambda^{(i)} | \beta^i) p(\beta^i | \gamma, \kappa) \right. \\ \times p(\Omega^{(i)} | \mathbf{M}^{(i)}, \xi) \left[\prod_{j=1}^{t_i} p(y_j^{(i)} | z_j^{(i)}, \Omega_j^{(i)}, \Lambda^{(i)}, \Phi, \varpi) \right. \\ \left. \left. \times p(z_j^{(i)} | \eta) \right] \right] \tag{B.1}$$

By denoting all parameters as Θ and variables as $\mathbf{V}$ while omitting hyperparameters, we obtain the following posterior expression:

$$p(\Theta, \mathbf{V} | \mathbf{Y}, \mathbf{M}) = \frac{p(\mathbf{Y}, \mathbf{M}, \Theta, \mathbf{V})}{p(\mathbf{Y}, \mathbf{M})} \tag{B.2}$$

Unfortunately, the exact posterior distribution of the latent variables is computationally intractable. The numerator is easy to compute for any configuration of the hidden variables and parameters. The problem is the denominator, which is the marginal probability of the data:

$$p(\mathbf{Y}, \mathbf{M}) = \int_{\Theta} \int_{\mathbf{V}} p(\mathbf{Y}, \mathbf{M}, \Theta, \mathbf{V}) \tag{B.3}$$

Computing the marginal requires a complicated integral over n examples of $|\Theta|$ parameters and another integral over the $|\mathbf{V}|^n$ configurations multiplied by the size of each variable in $\mathbf{V}$. As such, we appeal to the variational inference algorithm [25]. The main intuition behind variational methods is to first posit a family of distributions over the hidden parameters and variables that are indexed by a set of free parameters, and then fitting the parameters to find the member of the family that is closest to the true posterior of interest in Eq. B.2. The closeness is commonly measured using Kullback–Leibler (KL) divergence [35]. The resulting variational distribution is simpler than the true posterior so that the solution can be approximated. However, directly minimizing the KL divergence is infeasible due to the same reason that the posterior is difficult to compute, but, we can optimize an objective function that is equal to the negative KL divergence up to a constant. This is known as the evidence lower bound (ELBO), a lower bound on the logarithm of the marginal probability in Eq. B.3, i.e., $\log p(\mathbf{Y}, \mathbf{M})$.

This ELBO can be defined using Jensen's inequality on a variational distribution over the hidden variables $q(\Theta, \mathbf{V})$ as:

$$\begin{aligned}
\log p(\mathbf{Y}, \mathbf{M}) &= \log \int_{\Theta} \int_{V} p(\mathbf{Y}, \mathbf{M}, \Theta, \mathbf{V}) \\
&= \log \int_{\Theta} \int_{\mathbf{V}} p(\mathbf{Y}, \mathbf{M}, \Theta, V) \frac{q(\Theta, \mathbf{V})}{q(\Theta, \mathbf{V})} \\
&= \log(\mathbb{E}_q[\frac{p(\mathbf{Y}, \mathbf{M}, \Theta, \mathbf{V})}{q(\Theta, \mathbf{V})}]) \\
&\geq \mathbb{E}_q[\log p(\mathbf{Y}, \mathbf{M}, \Theta, \mathbf{V})] + \mathbb{H}(q) \\
&\triangleq \mathcal{L}(q)
\end{aligned} \tag{B.4}$$

The ELBO contains two terms. The first term, $\mathbb{E}_q[\log p(\mathbf{Y}, \mathbf{M}, \Theta, \mathbf{V})]$, captures how well $q(\Theta, \mathbf{V})$ describes a distribution of the model. The second term is the entropy of the variational distribution, $\mathbb{E}_q[-\log q(\Theta, \mathbf{V})]$, which protects the variational distribution from "overfitting" [24]. Both of these terms depend on $q(\Theta, \mathbf{V})$, the variational distribution of the hidden variables.

The simplest variational family of distributions is the mean-field family where each hidden variable/parameter is fully-factorized and governed by its own parameter. This allows us to tractably optimize the parameters to find a local minimum of the KL divergence. For SPREAT, the mean-field variational distribution is expressed as:

$$\begin{aligned}
q(\eta, \Lambda, z, d, \beta, \Phi, \Omega) = \prod_{a=1}^{b} q(\Phi_a|\phi_a) \Bigg[\prod_{i=1}^{n} q(\eta^{(i)}|\nu, \zeta^2) q(\Lambda^{(i)}|\lambda^{(i)}) q(\Omega^{(i)}|\omega^{(i)}) \\
\prod_{j=1}^{j=t_i} q(z_j^{(i)}|\varsigma_j^{(i)}) \Bigg]
\end{aligned} \tag{B.5}$$

where $\phi, \nu, \zeta^2, \lambda, \omega$ and ς are variational free parameters. Table 3 shows the correspondence between variational and the original parameters.

Taking together, the first term in Eq. B.4, $\mathbb{E}_q[\log p(\mathbf{Y}, \mathbf{M}, \Theta, V)]$, can be decomposed into:

$$\begin{aligned}
\mathbb{E}_q[\log p(\mathbf{Y}, \mathbf{M}, \Theta, V)] = \sum_{a=1}^{a=b} \mathbb{E}_q[\log p(\Phi_a|\alpha)] + \sum_{i=1}^{i=n} \Bigg(\mathbb{E}_q[\log p(\eta|\mu, \Sigma)] \\
+ \mathbb{E}_q[\log p(\Lambda^{(i)}|\beta^i)] + \mathbb{E}_q[\log p(\beta^i|\gamma, \kappa)] \\
+ \mathbb{E}_q[\log p(\Omega^{(i)}|\mathbf{M}^{(i)}, \xi)] \\
+ \sum_{j=1}^{j=t_i} \Big(\mathbb{E}_q[\log p(y_j^{(i)}|z_j^{(i)}, \Omega_j^{(i)}, \Lambda^{(i)}, \Phi, \varpi)] + \mathbb{E}_q[p(z_j^{(i)}|\eta)] \Big) \Bigg)
\end{aligned} \tag{B.6}$$

The second term $\mathbb{H}(q)$ in Eq. B.4 can be expressed as:

$$\begin{aligned}\mathbb{H}(q) = &-\sum_{a=1}^{a=b} \mathbb{E}_q[\log q(\Phi_a|\phi_a)] - \sum_{i=1}^{i=n} \Bigg(\mathbb{E}_q[\log q(\eta^{(i)}|\nu, \zeta^2)] + \mathbb{E}_q[\log q(\Lambda^{(i)}|\lambda^{(i)})] \\ &+ \mathbb{E}_q[\log q(\Omega^{(i)}|\omega^{(i)})] + \sum_{j=1}^{j=t_i} \mathbb{E}_q[\log q(z_j^{(i)}|\varsigma_j^{(i)})] \Bigg)\end{aligned} \tag{B.7}$$

Variational Lower Bound. Given Eq. B.6, we derive expressions for each term:

1. For the concept distribution over features, which are Dirichlet-distributed,

$$\begin{aligned}\mathbb{E}_q[\log p(\Phi_a|\alpha)] &= \mathbb{E}_q[\log \mathrm{Dir}(\Phi_a|\alpha)] \\ &= \mathbb{E}_q\Bigg[\log\Bigg(\frac{\Gamma(\sum_{j=1}^{j=t}\alpha_j)}{\prod_{j=1}^{j=t}\Gamma(\alpha_j)} \prod_{j=1}^{j=t} \Phi_{a,j}^{\alpha_j-1}\Bigg)\Bigg] \\ &= \mathbb{E}_q\Bigg[\log\Bigg(\frac{\Gamma(\sum_{j=1}^{j=t}\alpha_j)}{\prod_{j=1}^{j=t}\Gamma(\alpha_j)}\Bigg) + \sum_{j=1}^{j=t} \log \Phi_{a,j}^{\alpha_j-1}\Bigg] \\ &= \log\Gamma\Big(\sum_{j=1}^{j=t}\alpha_j\Big) - \sum_{j=1}^{j=t}\log\Gamma(\alpha_j) + \sum_{j=1}^{j=t}(\alpha_j-1)\mathbb{E}_q[\log\Phi_{a,j}]\end{aligned} \tag{B.8}$$

2. For the concepts probabilities for each example, which are Gaussian distributed,

$$\begin{aligned}\mathbb{E}_q[\log p(\eta|\mu,\Sigma)] &= \mathbb{E}_q\Big[\log\Big(\mathcal{N}(\eta|\mu,\Sigma)\Big)\Big] \\ &= \mathbb{E}_q\Big[\Big(\frac{1}{2}\log|\Sigma^{-1}| - \frac{b}{2}\log 2\pi - \frac{1}{2}(\eta-\mu)^\top\Sigma^{-1}(\eta-\mu)\Big)\Big] \\ &= \frac{1}{2}\log|\Sigma^{-1}| - \frac{b}{2}\log 2\pi \\ &\quad - \frac{1}{2}\Big(\mathrm{tr}(\mathrm{diag}(\zeta^2)\Sigma^{-1}) + (\nu-\mu)^\top\Sigma^{-1}(\nu-\mu)\Big)\end{aligned} \tag{B.9}$$

3. For the focused concept distributions for each example, which are Bernoulli distributed,

$$\begin{aligned}\mathbb{E}_q[\log p(\Lambda^{(i)}|\beta^{(i)})] &= \mathbb{E}_q\Big[\log \mathrm{Bernoulli}(\Lambda^{(i)}|\beta^{(i)})\Big] \\ &= \mathbb{E}_q\Bigg[\log\Bigg(\prod_{a=1}^{a=b}\beta_a^{(i),\Lambda_a^{(i)}}(1-\beta_a)^{(i),1-\Lambda_a^{(i)}}\Bigg)\Bigg] \\ &= \sum_{a=1}^{a=b}\Big(\lambda_a^{(i)}\log\beta_a^{(i)} + (1-\lambda_a^{(i)})\log(1-\beta_a^{(i)})\Big)\end{aligned} \tag{B.10}$$

4. For selecting a set of focused concepts for each example, which are beta distributed,

$$\begin{aligned}\mathbb{E}_q[\log p(\beta^{(i)}|\gamma,\kappa))] &= \mathbb{E}_q\Big[\log \mathrm{Beta}(\beta^{(i)}|\gamma,\kappa)\Big] \\ &= \sum_{a=1}^{a=b}\Big((\gamma-1)\log(\beta_a^{(i)}) + (\kappa-1)\log(1-\beta_a^{(i)}) - \log(B(\gamma,\kappa)\Big)\end{aligned} \tag{B.11}$$

5. For the hypothetical feature distributions for each example, which are Dirichlet distributed,

$$\begin{aligned}\mathbb{E}_q[\log p(\Omega_i|\mathbf{M}^{(i)},\xi)] &= \mathbb{E}_q\Bigg[\log\Bigg(\frac{\Gamma(\sum_{j=1}^{j=t}\xi_j+\mathbf{M}_j^{(i)})}{\prod_{j=1}^{j=t}\Gamma(\xi_j+\mathbf{M}_j^{(i)})}\prod_{j=1}^{j=t}\Omega_j^{(i),\xi_j+\mathbf{M}_j^{(i)}-1}\Bigg)\Bigg]\\ &= \log\Gamma\Big(\sum_{j=1}^{j=t}\xi_j+\mathbf{M}_j^{(i)}\Big)-\sum_{j=1}^{j=t}\log\Gamma(\xi_j+\mathbf{M}_j^{(i)})\\ &\quad+\sum_{j=1}^{j=t}(\xi_j+\mathbf{M}_j^{(i)}-1)\mathbb{E}_q[\log\Omega_j^{(i)}]\end{aligned} \tag{B.12}$$

6. For the feature assignments from both concept-feature and hypothetical feature distributions,

$$\begin{aligned}\mathbb{E}_q[\log p(y_j^{(i)}|z_j^{(i)},\Omega_j^{(i)},\Lambda^{(i)},\Phi,\varpi)] &= \mathbb{E}_q\Bigg[\log\Bigg(\prod_{c=1}^{c=t}\prod_{a=1}^{a=b}\varpi\Phi_{a,c}^{\mathbb{I}(a=z_{a,j}^{(i)}\wedge\Lambda_a^{(i)},c=y_j^{(i)}),(1-\Omega_c^{(i)})}\Bigg)\Bigg]\\ &= \log\varpi+\sum_{c=1}^{c=t}\sum_{a=1}^{a=b}\Big(y_{j,c}^{(i)}\varsigma_{a,j}^{(i)}\lambda_a^{(i)}\mathbb{E}_q[(1-\Omega_c^{(i)})]\mathbb{E}_q[\log\Phi_{a,j}]\Big)\end{aligned} \tag{B.13}$$

7. For the concept assignments over features, the expectation of the log probability of the latent concepts is given by:

$$\begin{aligned}\mathbb{E}_q[\log p(z_j^{(i)}|\eta)] &= \mathbb{E}_q\Bigg[\log\Bigg(\frac{\exp(\eta^\top(\mathrm{diag}(z_j^{(i)}))}{\sum_{k=1}^{k=b}\exp(\eta_k)}\Bigg)\Bigg]\\ &= \mathbb{E}_q\Big[\eta^\top(\mathrm{diag}(z_j^{(i)}))\Big]-\mathbb{E}_q\Big[\log\Big(\sum_{k=1}^{k=b}\exp(\eta_k)\Big)\Big]\\ &= \sum_{a=1}^{a=b}\nu_a\varsigma_{a,j}^{(i)}-\mathbb{E}_q\Big[\log\Big(\sum_{k=1}^{k=b}\exp(\eta_k)\Big)\Big]\end{aligned} \tag{B.14}$$

The second term is hard to compute, hence, we use the solution suggested by [23] in order to obtain the tightest lower bound on $-\mathbb{E}_q\Big[\log\Big(\sum_{k=1}^{k=b}\exp(\eta_k)\Big)\Big]$ using a first-order Taylor expansion. Because the function $-\log$ is convex, a first-order Taylor expansion about the point ϱ, a variational parameter, produces the following inequality:

$$\begin{aligned}-\mathbb{E}_q\Bigg[\log\Bigg(\sum_{k=1}^{k=b}\exp(\eta_k)\Bigg)\Bigg] &\geq -\log\varrho-\frac{\Big(\sum_{k=1}^{k=b}\mathbb{E}_q[\exp(\eta_k)]\Big)-\varrho}{\varrho}\\ &= 1-\log\varrho-\Bigg(\sum_{k=1}^{k=b}\mathbb{E}_q[\exp(\eta_k)]\Bigg)\varrho^{-1}\end{aligned} \tag{B.15}$$

Plugging back the results into Eq. B.14, we obtain:

$$\mathbb{E}_q[\log p(z_j^{(i)}|\eta)]\approx 1-\log\varrho+\sum_{a=1}^{a=b}\nu_a\varsigma_{a,j}^{(i)}-\Big(\sum_{k=1}^{k=b}\mathbb{E}_q[\exp(\eta_k)]\Big)\varrho^{-1} \tag{B.16}$$

Now, for the entropy $\mathbb{H}(q)$ in Eq. B.7, we decompose their expectations as:

1. For the concept-feature distributions, which are Dirichlet distributed,

$$\begin{aligned}\mathbb{E}_q[\log q(\Phi_a|\phi_a)] &= \mathbb{E}_q[\log \mathrm{Dir}(\Phi_a|\phi_a)] \\ &= \mathbb{E}_q\Big[\log\Big(\frac{\Gamma(\sum_{j=1}^{j=t}\phi_{a,j})}{\prod_{j=1}^{j=t}\Gamma(\phi_{a,j})}\prod_{j=1}^{j=t}\Phi_{a,j}^{\phi_{a,j}-1}\Big)\Big] \\ &= \log\Gamma\Big(\sum_{j=1}^{j=t}\phi_{a,j}\Big) - \sum_{j=1}^{j=t}\log\Gamma(\phi_{a,j}) + \sum_{j=1}^{j=t}(\phi_{a,j}-1)\mathbb{E}_q[\log\Phi_{a,j}]\end{aligned} \tag{B.17}$$

2. For the concept distributions, which are Gaussian distributed,

$$\begin{aligned}\mathbb{E}_q[\log q(\eta^{(i)}|\nu,\zeta^2)] =& \mathbb{E}_q\Big[\log\prod_{a=1}^{a=b}\mathcal{N}(\eta_a^{(i)}|\nu_a,\zeta_a^2)\Big] \\ =& -\sum_{a=1}^{a=b}\frac{1}{2}\Big(\log\zeta_a^2+\log(2\pi)+1\Big)\end{aligned} \tag{B.18}$$

3. For the concept choice parameter, which are Bernoulli distributed,

$$\begin{aligned}\mathbb{E}_q[\log q(\Lambda^{(i)}|\lambda^{(i)})] &= \mathbb{E}_q[\log\prod_{a=1}^{a=b}\mathrm{Bernoulli}(\Lambda_a^{(i)}|\lambda_a^{(i)})] \\ &= \sum_{a=1}^{a=b}\Big(\lambda_a^{(i)}\log\lambda_a^{(i)}+(1-\lambda_a^{(i)})\log(1-\lambda_a^{(i)})\Big)\end{aligned} \tag{B.19}$$

4. For the supplementary feature distributions over examples, which are Dirichlet distributed,

$$\begin{aligned}\mathbb{E}_q[\log q(\Omega^{(i)}|\omega^{(i)})] &= \mathbb{E}_q[\log \mathrm{Dir}(\Omega^{(i)}|\omega^{(i)})] \\ &= \mathbb{E}_q\Big[\log\Big(\frac{\Gamma(\sum_{j=1}^{j=t}\omega_j^{(i)})}{\prod_{j=1}^{j=t}\Gamma(\omega_j^{(i)})}\prod_{j=1}^{j=t}\Omega_j^{(i),\omega_j^{(i)}-1}\Big)\Big] \\ &= \log\Gamma\Big(\sum_{j=1}^{j=t}\omega_j^{(i)}\Big) - \sum_{j=1}^{j=t}\log\Gamma(\omega_j^{(i)}) + \sum_{j=1}^{j=t}(\omega_j^{(i)}-1)\mathbb{E}_q[\log\Omega_j^{(i)}]\end{aligned} \tag{B.20}$$

5. For the feature assignments over examples, which are multinomially distributed,

$$\mathbb{E}_q[\log q(z_j^{(i)}|\varsigma_j^{(i)})] = \mathbb{E}_q\Big[\log\prod_{a=1}^{a=b}(\varsigma_{a,j}^{(i)})^{z_{a,j}^{(i)}}\Big] = \sum_{a=1}^{a=b}\varsigma_{a,j}^{(i)}\log\varsigma_{a,j}^{(i)} \tag{B.21}$$

where the exceptions of all the above equations can be derived using:

$$\mathbb{E}_q[\log \Phi_{a,j}] = \Big(\Psi(\phi_{a,j}) - \Psi(\sum_{k=1}^{k=t} \phi_{a,k})\Big)$$
$$\mathbb{E}_q[\log \Omega_j^{(i)}] = \Big(\Psi(\omega_j^{(i)}) - \Psi(\sum_{k=1}^{k=t} \omega_k^{(i)})\Big)$$
$$\mathbb{E}_q[(1-\Omega_c^{(i)})] = \frac{1-\omega_c^{(i)}}{\sum_{k=1}^{k=t}(1-\omega_k^{(i)})}$$
$$\mathbb{E}_q[\exp(\eta_k)] = \exp(\nu_a + \frac{1}{2}\zeta_a^2)$$
$$B(\gamma,\kappa) = \frac{\Gamma(\gamma)\Gamma(\kappa)}{\Gamma(\gamma+\kappa)}$$

Not that Γ denotes the Gamma function while Ψ is the logarithmic derivative of the Gamma function.

Merging All the Expectations of the ELBO Terms Now, by joining all the terms, the full ELBO can be defined as:

$$\mathcal{L}(q) = \mathcal{L}^1(q) + \mathcal{L}^2(q) \tag{B.22}$$

where,

$$\begin{aligned}
\mathcal{L}^1(q) = &\sum_{a=1}^{a=b}\Bigg(\log\Gamma\Big(\sum_{j=1}^{j=t}\alpha_j\Big) - \sum_{j=1}^{j=t}\log\Gamma(\alpha_j) + \sum_{j=1}^{j=t}(\alpha_j-1)\Big(\Psi(\phi_{a,j}) - \Psi(\sum_{k=1}^{k=t}\phi_{a,k})\Big)\Bigg)\\
&+\sum_{i=1}^{i=n}\Bigg(\frac{1}{2}\log|\Sigma^{-1}| - \frac{b}{2}\log 2\pi - \frac{1}{2}\Big(\mathrm{tr}(\mathrm{diag}(\zeta^2)\Sigma^{-1}) + (\nu-\mu)^\top\Sigma^{-1}(\nu-\mu)\Big)\Bigg)\\
&+\sum_{i=1}^{i=n}\sum_{a=1}^{a=b}\Bigg(\lambda_a^{(i)}\log(\beta_a^{(i)}) + (1-\lambda_a^{(i)})\log(1-\beta_a^{(i)})\Bigg)\\
&+\sum_{i=1}^{i=n}\sum_{a=1}^{a=b}\Bigg((\gamma-1)\log(\beta_a^{(i)}) + (\kappa-1)\log(1-\beta_a^{(i)}) - \log(B(\gamma,\kappa)\Bigg)\\
&+\sum_{i=1}^{i=n}\Bigg(\log\Gamma\Big(\sum_{j=1}^{j=t}\xi_j + \mathbf{M}_j^{(i)}\Big) - \sum_{j=1}^{j=t}\log\Gamma(\xi_j + \mathbf{M}_j^{(i)}) + \sum_{j=1}^{j=t}(\xi_j + \mathbf{M}_j^{(i)} - 1)\Big(\Psi(\omega_j^{(i)}) - \Psi(\sum_{k=1}^{k=t}\omega_k^{(i)})\Big)\Bigg)\\
&+\sum_{i=1}^{i=n}\sum_{j=1}^{j=t_i}\Bigg(\log\varpi + \sum_{c=1}^{c=t}\sum_{a=1}^{a=b}\Bigg(y_{j,c}^{(i)}\varsigma_{a,j}^{(i)}\lambda_a^{(i)}\frac{1-\omega_c^{(i)}}{\sum_{k=1}^{k=t}(1-\omega_k^{(i)})}\Big(\Psi(\phi_{a,j}) - \Psi(\sum_{k=1}^{k=t}\phi_{a,k})\Big)\Bigg)\Bigg)\\
&+\sum_{i=1}^{i=n}\sum_{j=1}^{j=t_i}\Bigg(1 - \log\varrho + \sum_{a=1}^{a=b}\nu_a\varsigma_{a,j}^{(i)} - \Big(\sum_{k=1}^{k=b}\exp(\nu_k + \frac{1}{2}\zeta_k^2)\Big)\varrho^{-1}\Bigg)
\end{aligned} \tag{B.23}$$

$$\begin{aligned}\mathcal{L}^2(q) = &-\sum_{a=1}^{a=b}\left(\log\Gamma\Big(\sum_{j=1}^{j=t}\phi_{a,j}\Big) - \sum_{j=1}^{j=t}\log\Gamma(\phi_{a,j}) + \sum_{j=1}^{j=t}(\phi_{a,j}-1)\Big(\Psi(\phi_{a,j}) - \Psi(\sum_{k=1}^{k=t}\phi_{a,k})\Big)\right)\\
&+\sum_{i=1}^{i=n}\sum_{a=1}^{a=b}\left(\frac{1}{2}\Big(\log\zeta_a^2 + \log(2\pi) + 1\Big)\right)\\
&-\sum_{i=1}^{i=n}\sum_{a=1}^{a=b}\left(\lambda_a^{(i)}\log\lambda_a^{(i)} + (1-\lambda_a^{(i)})\log(1-\lambda_a^{(i)})\right)\\
&-\sum_{i=1}^{i=n}\left(\log\Gamma\Big(\sum_{j=1}^{j=t}\omega_j^{(i)}\Big) - \sum_{j=1}^{j=t}\log\Gamma(\omega_j^{(i)}) + \sum_{j=1}^{j=t}(\omega_j^{(i)}-1)\Big(\Psi(\omega_j^{(i)}) - \Psi(\sum_{k=1}^{k=t}\omega_k^{(i)})\Big)\right)\\
&-\sum_{i=1}^{i=n}\sum_{j=1}^{j=t_i}\sum_{a=1}^{a=b}\varsigma_{a,j}^{(i)}\log\varsigma_{a,j}^{(i)}
\end{aligned} \tag{B.24}$$

B.4 Optimizing the ELBO Terms

In this section, we maximize the bound in Eq. B.22 with respect to each variational parameters using coordinate ascent updates. Using this approach, each variational parameter is optimized individually while holding the remaining variables fixed. Practically, a more convenient way is to apply the mini-batch gradient approach that alternates between subsampling a batch of examples and updating each variational parameter, after being scaled by a learning rate [31]. This structure of learning assists us to approximate the posterior with massive examples, making the complete problem computationally scalable.

1. **Optimizing w.r.t. ς.** Gathering only the terms in the bound that contain ς, we obtain:

$$\begin{aligned}\mathcal{L}(q)_{[\varsigma]} = &\sum_{i=1}^{i=n}\sum_{j=1}^{j=t_i}\sum_{c=1}^{c=t}\sum_{a=1}^{a=b} y_{j,c}^{(i)}\varsigma_{a,j}^{(i)}\lambda_a^{(i)}\frac{1-\omega_c^{(i)}}{\sum_{k=1}^{k=t}(1-\omega_k^{(i)})}\Big(\Psi(\phi_{a,j}) - \Psi(\sum_{k=1}^{k=t}\phi_{a,k})\Big)\\
&+\sum_{i=1}^{i=n}\sum_{j=1}^{j=t_i}\sum_{a=1}^{a=b}\nu_a\varsigma_{a,j}^{(i)} - \sum_{i=1}^{i=n}\sum_{j=1}^{j=t_i}\sum_{a=1}^{a=b}\varsigma_{a,j}^{(i)}\log\varsigma_{a,j}^{(i)}
\end{aligned} \tag{B.25}$$

Taking derivatives w.r.t. $\varsigma_{a,j}^{(i)}$, we obtain:

$$\begin{aligned}\frac{\partial\mathcal{L}(q)_{[\varsigma]}}{\partial\varsigma_{a,j}^{(i)}} = &\sum_{c=1}^{c=t} y_{j,c}^{(i)}\lambda_a^{(i)}\frac{1-\omega_c^{(i)}}{\sum_{k=1}^{k=t}(1-\omega_k^{(i)})}\Big(\Psi(\phi_{a,j}) - \Psi(\sum_{k=1}^{k=t}\phi_{a,k})\Big)\\
&+\nu_a - \log\varsigma_{a,j}^{(i)} - 1
\end{aligned} \tag{B.26}$$

The analytical expression of the variational concept assignment $q(\varsigma)$ for each feature y_j and concept a is not amenable due to the non-conjugacy of logistic-normal with latent variables. Instead, we approximate the solution according to:

$$\varsigma_{a,j}^{(i)} \propto \exp\left(\sum_{c=1}^{c=t} y_{j,c}^{(i)} \lambda_a^{(i)} \frac{1-\omega_c^{(i)}}{\sum_{k=1}^{k=t}(1-\omega_k^{(i)})}\Big(\Psi(\phi_{a,j}) - \Psi(\sum_{k=1}^{k=t}\phi_{a,k})\Big) + \nu_a - 1\right) \tag{B.27}$$

where $\Psi(.)$ is the digamma function. Observe how the variational parameter $\omega_*^{(i)}$ serves as the smoothing term in selecting concepts for each feature, either from $\mathbf{M}_i$ or from $\mathcal{P}$, when $\omega_c^{(i)} > 0$. However, if $\omega_c^{(i)} = 0$, then $\varsigma_{a,j}^{(i)}$ is updated based on $\phi_{a,j}$.

2. **Optimizing w.r.t. ν.** Collecting only the terms in the bound that contain ν gives,

$$\begin{aligned}\mathcal{L}(q)_{[\nu]} = &\sum_{i=1}^{i=n}\Bigg(-\frac{1}{2}(\nu-\mu)^\top \Sigma^{-1}(\nu-\mu) + \sum_{j=1}^{j=t_i}\sum_{a=1}^{a=b}\nu_a \varsigma_{a,j}^{(i)} \\ &- \sum_{j=1}^{j=t_i}\Big(\sum_{k=1}^{k=b}\exp(\nu_k + \frac{1}{2}\zeta_k^2)\Big)\varrho^{-1}\Bigg)\end{aligned} \tag{B.28}$$

Taking derivatives w.r.t. ν_a for each concept a, we obtain:

$$\frac{\partial \mathcal{L}(q)_{[\nu]}}{\partial \nu_a} = -\Sigma^{-1}(\nu-\mu) + \sum_{j=1}^{j=t_i}\varsigma_{a,j}^{(i)} - \Big(\exp(\nu_a + \frac{1}{2}\zeta_a^2)\Big)t_i\varrho^{-1} \tag{B.29}$$

where ϱ is another variational parameter, as in CTM [23]. However, the above equation in hard to optimize, instead, we use a conjugate gradient algorithm.

3. **Optimizing w.r.t. ζ^2.** By symmetry, we gather all the terms that has ζ^2 from Eq. B.22:

$$\begin{aligned}\mathcal{L}(q)_{[\zeta^2]} = &-\frac{1}{2}\sum_{i=1}^{i=n}\mathrm{tr}\Big(\mathrm{diag}(\zeta^2)\Sigma^{-1}\Big) - \sum_{i=1}^{i=n}\sum_{j=1}^{j=t_i}\Big(\sum_{k=1}^{k=b}\exp\Big(\nu_k + \frac{1}{2}\zeta_k^2\Big)\Big)\varrho^{-1} \\ &+ \frac{1}{2}\sum_{i=1}^{i=n}\sum_{a=1}^{a=b}\log\zeta_a^2\end{aligned} \tag{B.30}$$

Taking derivatives w.r.t. ζ_a^2 for each concept a, we obtain:

$$\frac{\partial \mathcal{L}(q)_{[\zeta^2]}}{\partial \zeta_a^2} = -\frac{1}{2}\Big(\Sigma_{a,a}^{-1} + t_i\varrho^{-1}\exp\Big(\nu_a + \frac{1}{2}\zeta_a^2\Big) - \frac{1}{\zeta_a^2}\Big) \tag{B.31}$$

Again, there is no analytical solution to the above formula. Instead, we use the Newton's method for each coordinate such that $\zeta_a \in \mathbb{R}_{>0}$.

4. **Optimizing w.r.t. ϱ.** Extracting the terms involving ϱ in the bound gives,

$$\mathcal{L}(q)_{[\varrho]} = -\sum_{i=1}^{i=n}\sum_{j=1}^{j=t_i}\log\varrho - \sum_{i=1}^{i=n}\sum_{j=1}^{j=t_i}\Big(\sum_{k=1}^{k=b}\exp(\nu_k + \frac{1}{2}\zeta_k^2)\Big)\varrho^{-1} \tag{B.32}$$

Taking derivatives w.r.t. ϱ, we obtain:

$$\frac{\partial \mathcal{L}(q)_{[\varrho]}}{\partial \varrho} = -t_i n \varrho^{-1} + t_i n \Big(\sum_{k=1}^{k=b} \exp(\nu_k + \frac{1}{2}\zeta_k^2) \Big) \varrho^{-2} \tag{B.33}$$

Equating the above formula to zero to obtain a maximum, we get:

$$\varrho = \sum_{k=1}^{k=b} \exp(\nu_k + \frac{1}{2}\zeta_k^2) \tag{B.34}$$

5. **Optimizing w.r.t. ω.** Isolating only the terms in the bound that contain variational background feature distributions $q(\omega)$, we obtain:

$$\begin{aligned}\mathcal{L}(q)_{[\omega]} =& \sum_{i=1}^{i=n}\sum_{j=1}^{j=t} \Psi(\omega_j^{(i)})(\xi_j + \mathbf{M}_j^{(i)} - \omega_j^{(i)}) - \sum_{i=1}^{i=n}\sum_{j=1}^{j=t} \Psi(\sum_{k=1}^{k=t} \omega_k^{(i)})(\xi_j + \mathbf{M}_j^{(i)} - \omega_j^{(i)}) \\ &+ \sum_{i=1}^{i=n}\sum_{j=1}^{j=t_i}\sum_{c=1}^{c=t}\sum_{a=1}^{a=b} \Big(y_{j,c}^{(i)} \varsigma_{a,j}^{(i)} \lambda_a^{(i)} \frac{1-\omega_c^{(i)}}{\sum_{k=1}^{k=t}(1-\omega_k^{(i)})} \Big(\Psi(\phi_{a,j}) - \Psi(\sum_{k=1}^{k=t} \phi_{a,k}) \Big) \Big) \\ &+ \sum_{i=1}^{i=n}\sum_{j=1}^{j=t} \log \Gamma(\omega_j^{(i)}) - \sum_{i=1}^{i=n} \log \Gamma \Big(\sum_{j=1}^{j=t} \omega_j^{(i)} \Big)\end{aligned} \tag{B.35}$$

Taking derivatives w.r.t. $\omega_c^{(i)}$ gives

$$\begin{aligned}\frac{\partial \mathcal{L}(q)_{[\omega]}}{\partial \omega_c^{(i)}} =& \Big(\Psi'(\omega_c^{(i)}) - \Psi'(\sum_{k=1}^{k=t} \omega_k^{(i)}) \Big)(\xi_c + \mathbf{M}_c^{(i)} - \omega_c^{(i)}) \\ &- \Big(\frac{1-\omega_c^{(i)} - \sum_{k=1}^{k=t}(1-\omega_k^{(i)})}{(\sum_{k=1}^{k=t}(1-\omega_k^{(i)}))^2} \Big) \sum_{j=1}^{j=t_i}\sum_{a=1}^{a=b} y_{j,c}^{(i)} \varsigma_{a,j}^{(i)} \lambda_a^{(i)} \Big(\Psi(\phi_{a,j}) - \Psi(\sum_{k=1}^{k=t} \phi_{a,k}) \Big)\end{aligned} \tag{B.36}$$

Setting it's derivatives to zero does not lead to a closed-form solution, instead, we approximate $\omega_c^{(i)}$ for each example i according to:

$$\begin{aligned}\omega_c^{(i)} \propto& \xi_c + \mathbf{M}_c^{(i)} - \Big(\frac{1-\omega_c^{(i)} - \sum_{k=1}^{k=t}(1-\omega_k^{(i)})}{(\sum_{k=1}^{k=t}(1-\omega_k^{(i)}))^2} \Big) \sum_{j=1}^{j=t_i}\sum_{a=1}^{a=b} y_{j,c}^{(i)} \varsigma_{a,j}^{(i)} \lambda_a^{(i)} \\ &\times \Big(\Psi(\phi_{a,j}) - \Psi(\sum_{k=1}^{k=t} \phi_{a,k}) \Big)\end{aligned} \tag{B.37}$$

6. **Optimizing w.r.t. λ.** Collecting the terms that contain λ, we obtain:

$$\begin{aligned}\mathcal{L}(q)_{[\lambda]} =& \sum_{i=1}^{i=n}\sum_{a=1}^{a=b} \lambda_a^{(i)} (\log(\beta_a^{(i)}) - \log \lambda_a^{(i)}) \\ &+ \sum_{i=1}^{i=n}\sum_{a=1}^{a=b} (1-\lambda_a^{(i)}) \Big(\log(1-\beta_a^{(i)}) - \log(1-\lambda_a^{(i)}) \Big)\end{aligned} \tag{B.38}$$

Taking derivatives w.r.t. $\lambda_a^{(i)}$, we obtain:

$$\frac{\partial \mathcal{L}(q)_{[\lambda]}}{\partial \lambda_a^{(i)}} = \log(1-\lambda_a^{(i)}) - \log \lambda_a^{(i)} + \log(\beta_a^{(i)}) - \log(1-\beta_a^{(i)}) \tag{B.39}$$

Equating the above formula to zero to obtain a maximum, we get the canonical parameterisation of the Bernoulli distribution:

$$\theta = \log\left(\frac{\lambda_a^{(i)}}{1-\lambda_a^{(i)}}\right) = \log(\beta_a^{(i)}) - \log(1-\beta_a^{(i)}) \tag{B.40}$$

Therefore, we get the following updates:

$$\lambda_a^{(i)} = \frac{1}{1+\exp^{-\theta}} \tag{B.41}$$

7. **Optimizing w.r.t. ϕ.** Finally, the optimal solution of the variational concept feature distribution $q(\Phi_a|\phi_a)$ for each concept a is obtained by isolating terms involved in the bound Eq. B.4:

$$\begin{aligned}\mathcal{L}(q)_{[\phi]} =& \sum_{a=1}^{a=b}\sum_{c=1}^{c=t} \Psi(\phi_{a,c})\left(\alpha_c - \phi_{a,c} + \sum_{i=1}^{i=n}\sum_{j=1}^{j=t_i} y_{j,c}^{(i)}\varsigma_{a,j}^{(i)}\lambda_a^{(i)}\frac{1-\omega_c^{(i)}}{\sum_{k=1}^{k=t}(1-\omega_k^{(i)})}\right)\\ &- \sum_{a=1}^{a=b}\sum_{j=1}^{c=t}\Psi(\sum_{k=1}^{k=t}\phi_{a,k})\left(\alpha_c - \phi_{a,c} + \sum_{i=1}^{i=n}\sum_{j=1}^{j=t_i} y_{j,c}^{(i)}\varsigma_{a,j}^{(i)}\lambda_a^{(i)}\frac{1-\omega_c^{(i)}}{\sum_{k=1}^{k=t}(1-\omega_k^{(i)})}\right)\\ &- \sum_{a=1}^{a=b}\log\Gamma\Big(\sum_{j=1}^{j=t}\phi_{a,j}\Big) + \sum_{a=1}^{a=b}\sum_{j=1}^{j=t}\log\Gamma(\phi_{a,j})\end{aligned} \tag{B.42}$$

After taking derivatives w.r.t. $\phi_{a,c}$, we obtain:

$$\begin{aligned}\frac{\partial \mathcal{L}(q)_{[\phi]}}{\partial \phi_{a,c}} =& \Psi'(\phi_{a,c})\left(\alpha_c - \phi_{a,c} + \sum_{i=1}^{i=n}\sum_{j=1}^{j=t_i} y_{j,c}^{(i)}\varsigma_{a,j}^{(i)}\lambda_a^{(i)}\frac{1-\omega_c^{(i)}}{\sum_{k=1}^{k=t}(1-\omega_k^{(i)})}\right)\\ &- \Psi'(\sum_{k=1}^{k=t}\phi_{a,k})\left(\alpha_c - \phi_{a,c} + \sum_{i=1}^{i=n}\sum_{j=1}^{j=t_i} y_{j,c}^{(i)}\varsigma_{a,j}^{(i)}\lambda_a^{(i)}\frac{1-\omega_c^{(i)}}{\sum_{k=1}^{k=t}(1-\omega_k^{(i)})}\right)\end{aligned} \tag{B.43}$$

Equating the above formula to zero to obtain a maximum, we get:

$$\phi_{a,c} = \alpha_c + \sum_{i=1}^{i=n}\sum_{j=1}^{j=t_i} y_{j,c}^{(i)}\varsigma_{a,j}^{(i)}\lambda_a^{(i)}\frac{1-\omega_c^{(i)}}{\sum_{k=1}^{k=t}(1-\omega_k^{(i)})} \tag{B.44}$$

The variational inference algorithm samples a mini-batch from a collection, and use it to compute the local latent parameters in Eqs B.27, B.29, B.31, B.34, B.37, and B.41 until the evidence lower bound in Eq. B.4 converges. Then, the global variational parameter ϕ is updated using the posteriors (β, Λ, η, z, Ω) collected from the previous step in Eq. B.44, after being scaled according to

the learning rate $\tau = (s + l)^{-g}$, where s is the current step, $l \geq 0$ is the delay factor, and $g \in (0.5, 1]$ is the forgetting rate. The variational inference process for SPREAT is summerized in Algorithm 7.

```
Initialize ϕ, ν, ζ², λ, ω, ς, γ, κ, ξ, α, ϖ, ι, s = 0, l ≥ 0, g ∈ (0.5, 1]
repeat
    s = s + 1;
    example a minibatch randomly B ⊂ P;
    for i ∈ B do
        repeat
            Update ς^(i) with Eq. B.27;
            Update ν^(i) with Eq. B.29 using conjugate gradient algorithm;
            Update ζ^(2,(i)) with Eq. B.31 using Newton's method;
            Update ϱ^(i) with Eq. B.34;
            Update ω^(i) with Eq. B.37;
            Update λ^(i) with Eq. B.41;
        until local variational parameters converge;
        Compute optimal values μ = ν/|B|, Σ = diag(ζ²/|B|) + μμ^⊤;
        Compute global optimal values ϕ with Eq. B.44;
        Update the current estimate of the global variational paramters,
          x = (1 − τ)x + τx, where x ∈ {ϕ, μ, Σ};
    Update the learning rate τ = (s + l)^(−g);
until global convergence criterion is satisfied;
```

Algorithm 7: Stochastic variational inference for SPREAT

B.5 Posterior Predictive Distribution for SPREAT

The posterior predictive distribution is a useful and practical method to evaluate model's fitness and to compare models without requiring to compute bounds of those models. This metric estimates the distribution of an unobserved value ($\tilde{\mathbf{y}}$) given the observed values ($\mathbf{Y}_{obs}$) and parameters (Θ and $\mathbf{V}$) that are trained on a held-out training set [31]. The predictive distribution for SPREAT is:

$$\begin{aligned} p(\tilde{\mathbf{y}}|\mathbf{Y}_{obs}, \tilde{\mathbf{M}}, \mathbf{M}_{obs}) &= \int p(\tilde{\mathbf{y}}|\Theta, \tilde{\mathbf{M}}) p(\Theta|\mathbf{Y}_{obs}, \mathbf{M}_{obs}) d\Theta \\ &\approx \sum_{a=1}^{a=b} \left(\eta_a^{(i)} \times \sum_{j=1}^{j=t} \left(\Phi_{a,j} \times \tilde{\mathbf{y}}_j^{(i)} \right) \right) q(\Theta, \mathbf{V}) \end{aligned} \tag{B.45}$$

where $q(\Theta, \mathbf{V})$ corresponds to Eq. B.5 and trained on $\mathbf{Y}_{obs}$ and $\mathbf{M}_{obs}$.

C Experimental Setup

In this section, we describe the experimental settings and outline the materials used to evaluate the performance of reMap. The reMap and correlated models were written in Python v3 and depend on third party libraries (e.g. Numpy [49]). Unless otherwise specified all tests were conducted on a Linux server using 10 cores of Intel Xeon CPU E5-2650.

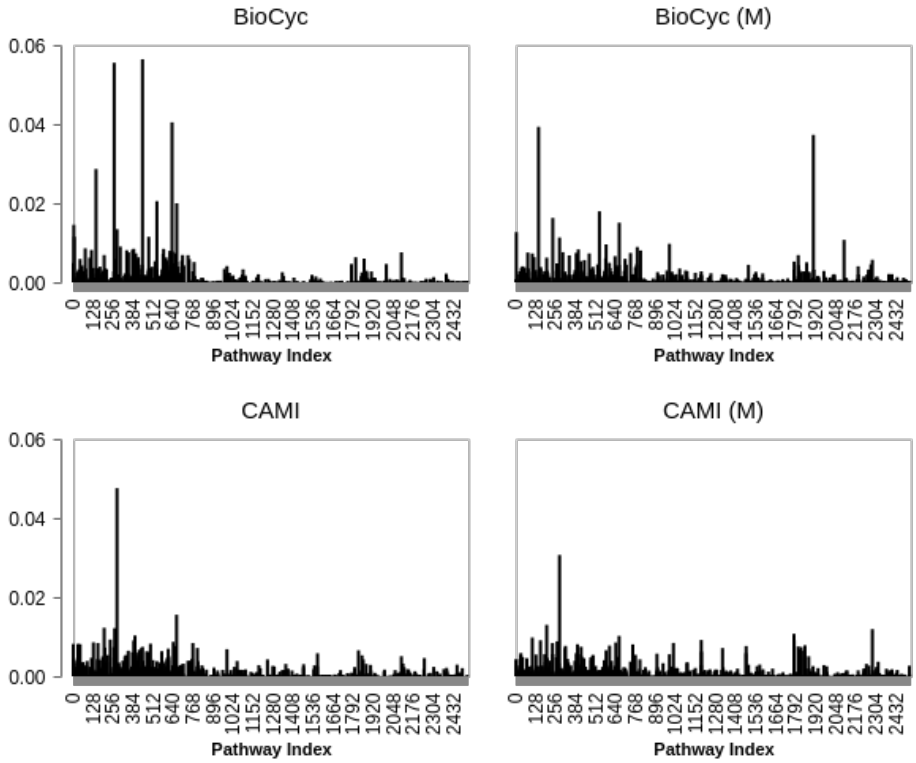


Fig. 7: Pathway frequency (averaged on all examples) in BioCyc (v20.5 T2 &3) and CAMI data, and their background pathways, indicated by **M**.

C.1 Description of Datasets

We used 11 simulated, organismal, and multi-organismal datasets to evaluate reMap's grouping performance: i)- BioCyc v20.5 T2 & 3 [1], ii)- 6 T1 golden data that are composed of six databases (*EcoCyc (v21)*, *HumanCyc (v19.5)*, *AraCyc (v18.5)*, *YeastCyc (v19.5)*, *LeishCyc (v19.5)*, and *TrypanoCyc (v18.5)*), iii)- Symbiont genomes describing distributed metabolic pathways between *Moranella* (GenBank NC-015735) and *Tremblaya* (GenBank NC-015736) [15], iv)- Critical Assessment of Metagenome Interpretation (CAMI) initiative low complexity dataset [16], consisting of 40 genomes and is obtained from edwards.sdsu.edu/research/cami-challenge-datasets/; v)- whole genome shotgun sequences from HOTS at 25m, 75m, 110m (sunlit) and 500m (dark) ocean depth intervals [17], and vi)- Synset-2, a noisy corrupted training set [11]. The detailed characteristics of the datasets are summarized in Table 4. For each dataset $\mathcal{S}$, we use $|\mathcal{S}|$ and $\mathrm{L}(\mathcal{S})$ 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:

Table 4: Characteristics of the experimental datasets. The notations $|\mathcal{S}|$, $\mathrm{L}(\mathcal{S})$, $\mathrm{LCard}(\mathcal{S})$, $\mathrm{LDen}(\mathcal{S})$, $\mathrm{DL}(\mathcal{S})$, and $\mathrm{PDL}(\mathcal{S})$ 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 $\mathrm{R}(\mathcal{S})$, $\mathrm{RCard}(\mathcal{S})$, $\mathrm{RDen}(\mathcal{S})$, $\mathrm{DR}(\mathcal{S})$, and $\mathrm{PDR}(\mathcal{S})$ have similar meanings as before but for the enzymatic reactions $\mathcal{E}$ in $\mathcal{S}$. $\mathrm{PLR}(\mathcal{S})$ represents a ratio of $\mathrm{L}(\mathcal{S})$ to $\mathrm{R}(\mathcal{S})$. The last column denotes the domain of $\mathcal{S}$.

Dataset	$\lvert\mathcal{S}\rvert$	$\mathrm{L}(\mathcal{S})$	$\mathrm{LCard}(\mathcal{S})$	$\mathrm{LDen}(\mathcal{S})$	$\mathrm{DL}(\mathcal{S})$	$\mathrm{PDL}(\mathcal{S})$	$\mathrm{R}(\mathcal{S})$	$\mathrm{RCard}(\mathcal{S})$	$\mathrm{RDen}(\mathcal{S})$	$\mathrm{DR}(\mathcal{S})$	$\mathrm{PDR}(\mathcal{S})$	$\mathrm{PLR}(\mathcal{S})$	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
BioCyc	9255	1804003	194.9220	0.0001	1463	0.1581	848714	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
HOT	4	–	–	–	–	–	182675	26096.4286	0.1429	1442	206.0000	–	Metagenomic Hawaii Ocean Time-series (10m, 75m, 110m, and 500m)
Synset-2	15000	6806262	453.7508	0.00007	2526	0.1684	34006386	2267.0924	0.00007	3650	0.2433	0.2001	Synthetically generated (corrupted)

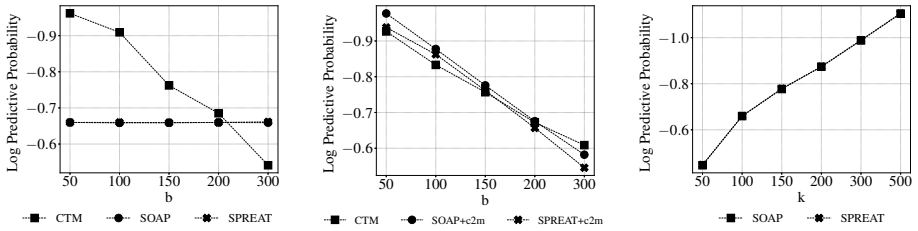
1. Label cardinality (LCard($\mathcal{S}$) $= \frac{1}{n}\sum_{i=1}^{i=n}\sum_{j=1}^{j=t}\mathbb{I}[\mathbf{Y}_{i,j} \neq -1]$), where $\mathbb{I}$ is an indicator function. It denotes the average number of pathways in $\mathcal{S}$.
2. Label density (LDen($\mathcal{S}$) $= \frac{LCard(\mathcal{S})}{L(\mathcal{S})}$). This is simply obtained through normalizing LCard($\mathcal{S}$) by the number of total pathways in $\mathcal{S}$.
3. Distinct pathway labels (DL($\mathcal{S}$)). This notation indicates the number of distinct pathways in $\mathcal{S}$.
4. Proportion of distinct pathway 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 as before but for the enzymatic reactions $\mathcal{E}$ in $\mathcal{S}$. Finally, PLR($\mathcal{S}$) represents a ratio of L($\mathcal{S}$) to R($\mathcal{S}$). The preprocessed experimental datasets can be obtained from https://zenodo.org/record/3971534#.YX9dpWDMK3A.

C.2 Parameter Settings

We applied the following default configurations:
1. reMap's parameters. The parameters for the reMap model are configured as: the learning rate is $\eta = 0.0001$, the batch size is 30, the number of epochs is $\tau = 10$, the group centroid hyperparameter is $\alpha = 16$, the cutoff threshold for cosine similarity is $v = 0.2$, the cutoff decision threshold for groups is $\beta = 0.3$, the number of groups is $b = 200$, and the subsampled group size is $\gamma = 50$. For regularized hyperparameters $\lambda_{1:5}$ and κ, we performed 10-fold cross-validation on a sample of BioCyc data (v20.5 T2 &3) and found the settings $\lambda_{1:5} = 0.01$ and $\kappa = 0.01$ to be the optimum.
2. Correlated models parameters. The parameters for the three correlated models are configured as: the pathway distribution over concepts Φ are initialized using gamma distribution (with shape and scale parameters are fixed to 100 and 1/100, respectively), the forgetting rate is $g = 0.9$, the delay rate is $l = 1$, the batch size is 100, the number of epochs is 3, the number of concepts is $b = 200$, top k pathways is 100 (only for SOAP and SPREAT), the Dirichlet hyperparameters α and ξ are 0.0001, and the beta hyperparameters γ and κ are 2 and 3, respectively. The supplementary pathways $\mathbf{M}$ for BioCyc, CAMI, and golden T1 datasets are obtained using mlLGPR [11] trained on Synset-2. A schematic view of pathway frequency across datasets for BioCyc T2 & 3 and CAMI, along with their augmented pathways is depicted in Fig. 7.
3. pathway2vec's parameters. The parameters for the pathway2vec framework [10] are configured as: "crt" as the embedding method, 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 is 100, the walk length is 100, the embedding dimension size is $m = 128$, the neighborhood size is 5, the size of negative examples is 5, and the used configuration of MetaCyc is "uec", indicating trimmed links among ECs.

Both reMap and correlated models are trained using BioCyc (v20.5 T2 &3) collection. After obtaining groups $\mathcal{S}_{group}$, we train leADS [14] using built-in



(a) Effect of b (b) Effect of b by collapsing (c) Effect of k with $b = 200$

Fig. 8: Log predictive distribution on CAMI data.

"factorization" option that enables training pathway groups and pathways, simultaneously, for 10 epochs using "nPSP" as the acquisition function and "pref-voting" as the prediction strategy with cutoff threshold 0.5. For all the remaining hyperparameters in pathway2vec, correlated models, leADS, and mlLGPR [11], they are fixed to their default values.

D Experimental Results

Two tests were performed to benchmark the performance of reMap including parameter sensitivity for correlated models and metabolic pathway prediction.

D.1 Sensitivity Analysis of Correlated Models

A fundamental challenge for the reMap pipeline is to acquire a good distribution of groups and pathways from correlated models for the purpose of relabeling. Following the common practice, here we examined various hyperparameters associated with correlated models. First, we compared the sensitivity of SOAP and SPREAT against CTM by incorporating the background pathways $\mathbf{M}$ while varying the number of groups according to $b \in \{50, 100, 150, 200, 300\}$. Next, we examined the "c2m" option for SOAP and SPREAT to show that these two models exhibit similar performances as CTM. Finally, we conducted sparsity analysis of group distribution by varying the cutoff threshold value according to $k \in \{50, 100, 150, 200, 300, 500\}$. For the comparative analysis, we applied CAMI as a test data to report the log predictive distribution (Section B.5), where a lower score entails higher generalization capability for the associated models.

While the log predictive scores for SOAP and SPREAT in Fig. 8a appears to be flat across group size, the CTM model projects a more realistic view where it's performances are seen to be gaining by including more groups. For the former models, this phenomena is expected due to the effects of supplementary pathways. That is, both models are encouraged to learn more pathways from $\mathbf{M}$ because the average pathway size for an example in $\mathbf{M}$ is ≈ 500 whereas

Table 5: 22 amino acid biosynthesis pathways and 28 pathway variants.

Amino Acid	MetaCyc Pathway ID	MetaCyc Pathway Name
Glycine	GLYSYN-PWY	glycine biosynthesis I
Alanine	ALANINE-VALINESYN-PWY	L-alanine biosynthesis I
	ALANINE-SYN2-PWY	L-alanine biosynthesis II
	PWY0-1021	L-alanine biosynthesis III
Arginine	ARGSYN-PWY	L-arginine biosynthesis I (via L-ornithine)
Asparagine	ASPARAGINE-BIOSYNTHESIS	L-asparagine biosynthesis I
	ASPARAGINESYN-PWY	L-asparagine biosynthesis II
Aspartate	ASPARTATESYN-PWY	L-aspartate biosynthesis
Chorismate	PWY-6163	chorismate biosynthesis from 3-dehydroquinate
	ARO-PWY	chorismate biosynthesis I
Cysteine	CYSTSYN-PWY	L-cysteine biosynthesis I
	PWY-7870	L-cysteine biosynthesis VII (from S-sulfo-L-cysteine)
Glutamate	GLUTSYN-PWY	L-glutamate biosynthesis I
	GLUTSYNIII-PWY	L-glutamate biosynthesis III
Glutamine	GLNSYN-PWY	L-glutamine biosynthesis I
Histidine	HISTSYN-PWY	L-histidine biosynthesis
Isoleucine	ILEUSYN-PWY	L-isoleucine biosynthesis I (from threonine)
Leucine	LEUSYN-PWY	L-leucine biosynthesis
Lysine	DAPLYSINESYN-PWY	L-lysine biosynthesis I
Methionine	HOMOSER-METSYN-PWY	L-methionine biosynthesis I
Phenylalanine	PHESYN	L-phenylalanine biosynthesis I
Proline	PROSYN-PWY	L-proline biosynthesis I
Selenocysteine	PWY0-901	L-selenocysteine biosynthesis I (bacteria)
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

in BioCyc v20.5 T2 & 3 is ≈ 195. By excluding $\mathbf{M}$ ("c2m"), we observe that the log predictive distribution of SOAP and SPREAT are similar with that of CTM, as shown in Fig. 8b, which supports our previous discussion. From Figs 8a and 8b, it is evident that $b = 200$ represents the optimum group size with the average number of distinct pathways is ≈ 15. By fixing $b = 200$, we search for an optimum k value. As illustrated in Fig. 8c, both SOAP and SPREAT deteriorate their performances (< -0.6) when $k > 100$. Taken together, we suggest the settings $b \in \mathbb{Z}_{[150,300]}$ and $k \in \mathbb{Z}_{[50,100]}$ to recover good pathway group and pathway distributions.

D.2 Accumulated History Probability Analysis

Table 5 shows 22 amino acid biosynthesis pathways with their 28 variants. Table 6 represents the selected 7 pathway groups that contain these amino acid pathways in their top 5 pathways for Escherichia coli K-12 MG1655 organism.

D.3 Metabolic Pathway Prediction

Here, groups obtained from all correlated modules are used for the pathway prediction task. For this, we trained leADS using the configuration discussed

Table 6: 7 pathway groups containing 22 amino acids and their top 5 pathways for Escherichia coli K-12 MG1655.

Group Index	MetaCyc Pathway ID	MetaCyc Pathway Name
13	GLUTSYNIII-PWY	L-glutamate biosynthesis III
	PWY-5484	glycolysis II (from fructose 6-phosphate)
	PWY-7187	pyrimidine deoxyribonucleotides ¡i¿de novo¡/i¿ biosynthesis II
	PWY-7197	pyrimidine deoxyribonucleotide phosphorylation
	PWY-7181	pyrimidine deoxyribonucleosides degradation
16	PROSYN-PWY	L-proline biosynthesis I
	ALANINE-SYN2-PWY	L-alanine biosynthesis II
	PWY0-1021	L-alanine biosynthesis III
	PWY0-1319	CDP-diacylglycerol biosynthesis II
	PWY-5667	CDP-diacylglycerol biosynthesis I
22	PWY0-541	cyclopropane fatty acid (CFA) biosynthesis
	PWY-7206	pyrimidine deoxyribonucleotides dephosphorylation
	HOMOSER-THRESYN-PWY	L-threonine biosynthesis
	PWY0-1585	formate to nitrite electron transfer
	PWY0-1352	nitrate reduction VIII (dissimilatory)
24	SERSYN-PWY	L-serine biosynthesis
	XYLCAT-PWY	xylose degradation I
	LYXMET-PWY	L-lyxose degradation
	PWY0-901	L-selenocysteine biosynthesis I (bacteria)
	PWY-4121	glutathionylspermidine biosynthesis
127	CYSTSYN-PWY	L-cysteine biosynthesis I
	PYRIDNUCSAL-PWY	NAD salvage pathway I
	GALACTCAT-PWY	D-galactonate degradation
	TRYPDEG-PWY	L-tryptophan degradation II (via pyruvate)
	GALACTARDEG-PWY	D-galactarate degradation I
140	PWY0-541	cyclopropane fatty acid (CFA) biosynthesis
	GLUTSYNIII-PWY	L-glutamate biosynthesis III
	LEUSYN-PWY	L-leucine biosynthesis
	TYRSYN	L-tyrosine biosynthesis I
	PWY0-1280	ethylene glycol degradation
152	PLPSAL-PWY	pyridoxal 5'-phosphate salvage I
	COBALSYN-PWY	adenosylcobalamin salvage from cobinamide I
	ALANINE-SYN2-PWY	L-alanine biosynthesis II
	PWY0-521	fructoselysine and psicoselysine degradation
	GLUTSYN-PWY	L-glutamate biosynthesis I

in Section C.2. The results are reported on golden T1 and CAMI data using four evaluation metrics: *Hamming loss*, *average precision*, *average recall*, and *average F1 score*. We also studied reMap's performance on Symbiont and HOTS data. For comparative analysis, four pathway prediction algorithms are used: i)- MinPath v1.2 [20], ii)- PathoLogic v21 [7], iii)- mlLGPR (elastic net with enzymatic reaction and pathway evidence features) [11], and iv)- triUMPF [13].

Table 7 shows that reMap+SOAP outperforms triUMPF on five T1 golden data (excluding LeishCyc) with regard to average recall and average F1 scores where numbers in boldface represent the best performance score in each column while the underlined text indicates the best performance among correlated models. For the remaining correlated models, their sensitivity scores are higher than triUMPF with the exception to EcoCyc and AraCyc. Similar results are observed for Symbiont, CAMI, and HOTS (Figs 10, 11, 12, and 13) data. In summary, this experiment demonstrates that pathway group based approach, in particular reMap+SOAP, improves pathway predictions.

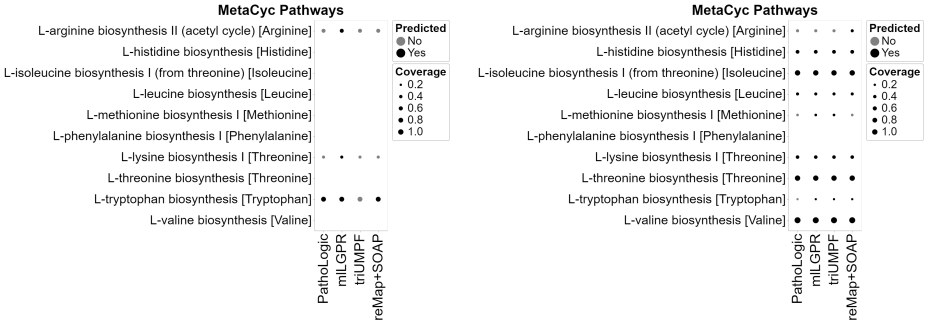
Table 7: Predictive performance of each comparing algorithm on 6 benchmark datasets. For each performance metric, '↓' indicates the smaller score is better while '↑' indicates the higher score is better. Bold text suggests the best performance in each column.

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
reMap+SOAP	**0.0392**	**0.0400**	0.1714	0.0934	0.0772	0.0479
reMap+SPREATE	0.0519	0.0827	0.1489	0.0748	0.0629	0.0503
reMap+CTM	0.0558	0.0835	0.1425	0.0804	0.0622	0.0503
reMap+SOAP+c2m	0.0590	0.0780	0.1457	0.0772	0.0614	0.0534
reMap+SPREATE+c2m	0.0542	0.0796	0.1520	0.0772	0.0598	0.0558

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
reMap+SOAP	0.8611	**0.7871**	0.6215	0.4851	0.2805	0.5985
reMap+SPREATE	**0.9400**	0.6750	0.8350	0.6000	0.3200	**0.6200**
reMap+CTM	0.9150	0.6700	**0.8750**	0.5650	0.3250	**0.6200**
reMap+SOAP+c2m	0.8950	0.7050	0.8550	0.5850	0.3300	0.6000
reMap+SPREATE+c2m	0.9250	0.6950	0.8150	0.5850	0.3400	0.5850

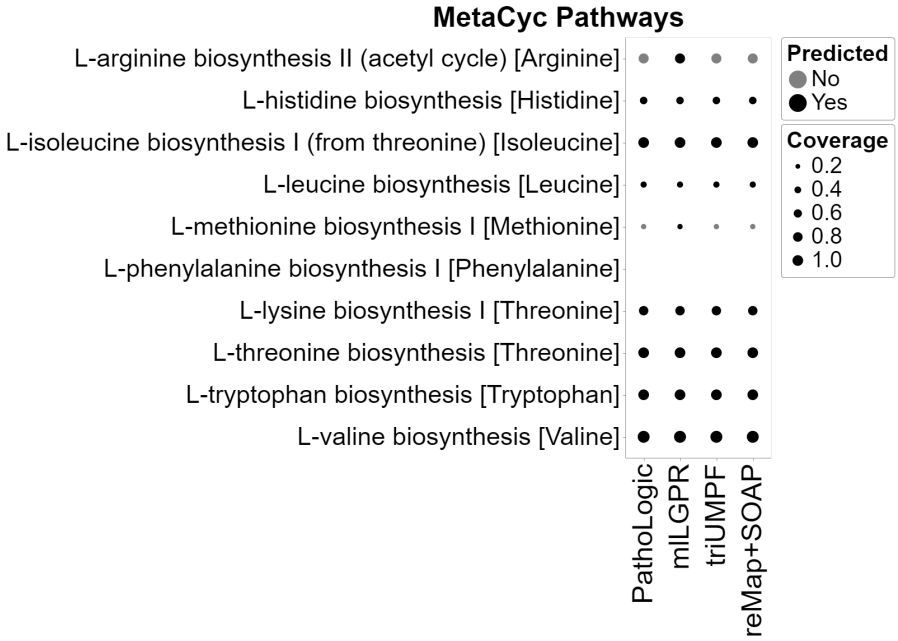
Methods	Average Recall Score ↑					
	EcoCyc	HumanCyc	AraCyc	YeastCyc	LeishCyc	TrypanoCyc
PathoLogic	0.8078	0.8423	0.7176	0.8734	0.8391	0.7829
MinPath	**0.9902**	**0.9713**	**0.9843**	**1.0000**	**1.0000**	**1.0000**
mlLGPR	0.8827	0.8459	0.7314	0.8603	0.9080	0.8914
triUMPF	0.7590	0.3835	0.3529	0.3319	0.7126	0.6229
reMap+SOAP	0.8078	0.8746	0.3863	0.4978	0.7931	0.9371
reMap+SPREATE	0.6124	0.4839	0.3275	0.5240	0.7356	0.7086
reMap+CTM	0.5961	0.4803	0.3431	0.4934	0.7471	0.7086
reMap+SOAP+c2m	0.5831	0.5054	0.3353	0.5109	0.7586	0.6857
reMap+SPREATE+c2m	0.6026	0.4982	0.3196	0.5109	0.7816	0.6686

Methods	Average F1 Score ↑					
	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
reMap+SOAP	**0.8336**	**0.8285**	0.4764	0.4914	0.4144	**0.7305**
reMap+SPREATE	0.7416	0.5637	0.4704	0.5594	0.4460	0.6613
reMap+CTM	0.7219	0.5595	0.4930	0.5268	0.4530	0.6613
reMap+SOAP+c2m	0.7061	0.5887	0.4817	0.5455	0.4599	0.6400
reMap+SPREATE+c2m	0.7298	0.5804	0.4592	0.5455	0.4739	0.6240



(a) Moranella

(b) Tremblaya



(c) Composite genomes

Fig. 9: Comparative study of predicted pathways for symbiont data between PathoLogic, mlLGPR, triUMPF, and reMap+SOAP. Black circles indicate predicted pathways by the associated models while grey circles indicate pathways that were not recovered by models. The size of circles corresponds the pathway coverage information.

References

21. Airoldi, E.M., Blei, D.M., Fienberg, S.E., Xing, E.P.: Mixed membership stochastic blockmodels. Journal of machine learning research **9**(Sep), 1981–2014 (2008)

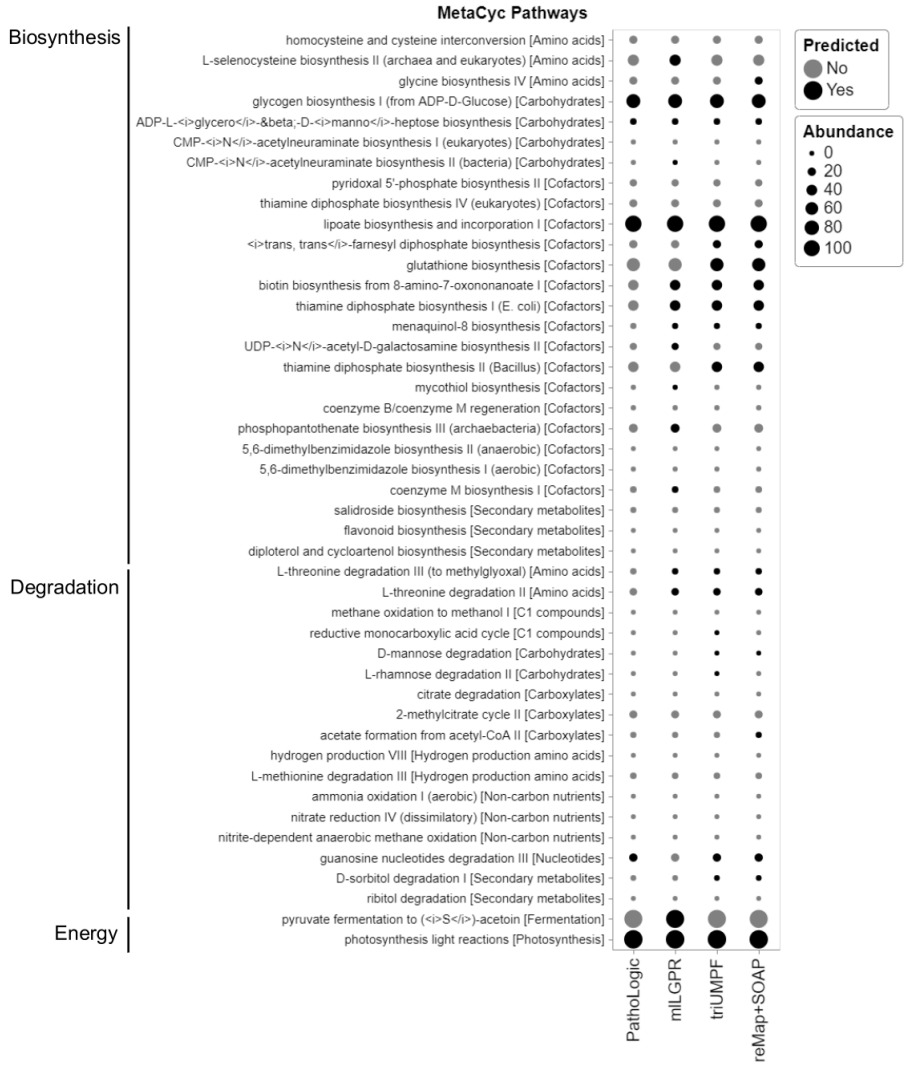


Fig. 10: Comparative study of predicted pathways for HOTS 25m dataset between PathoLogic, mlLGPR, triUMPF, and reMap+SOAP. Black circles indicate predicted pathways by the associated models while grey circles indicate pathways that were not recovered by models. The size of circles corresponds the pathway abundance information.

22. Bien, J., Tibshirani, R.J.: Sparse estimation of a covariance matrix. Biometrika **98**(4), 807–820 (2011)

23. Blei, D., Lafferty, J.: Correlated topic models. Advances in neural information processing systems **18**, 147 (2006)

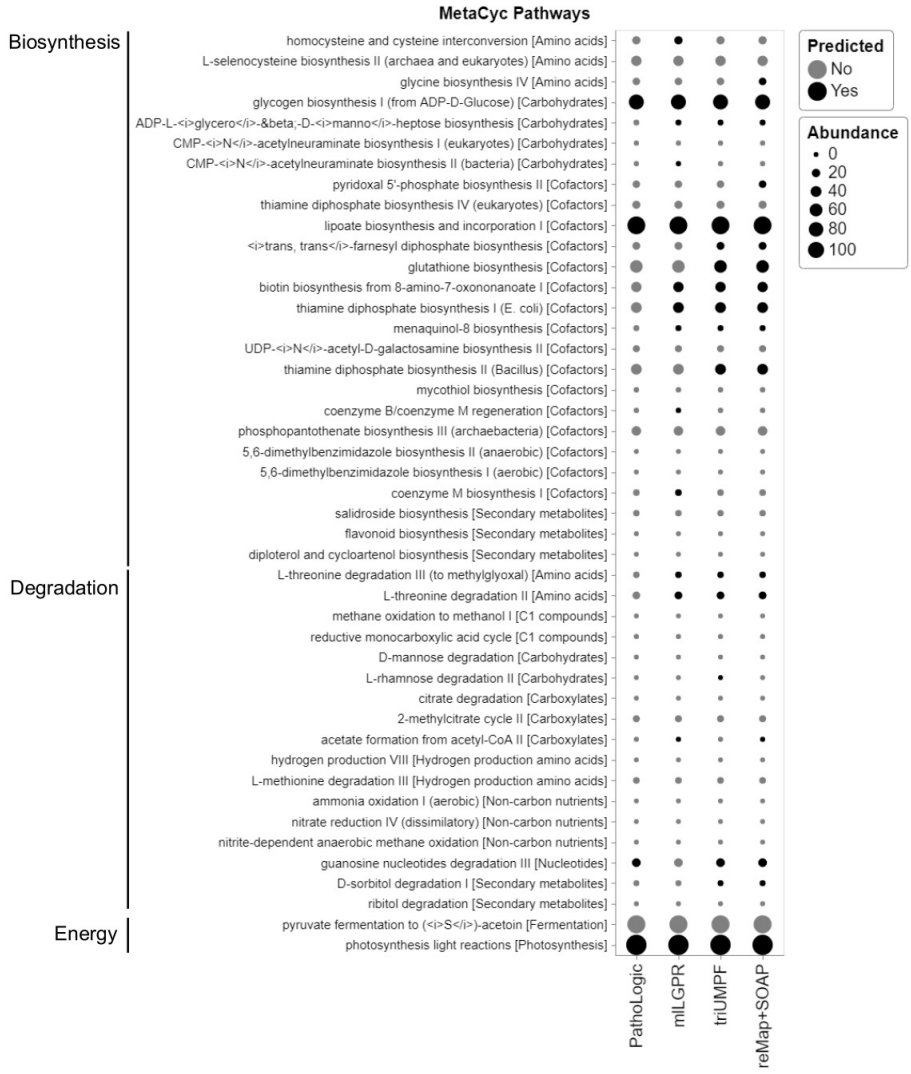


Fig. 11: Comparative study of predicted pathways for HOTS 75m dataset between PathoLogic, mlLGPR, triUMPF, and reMap+SOAP. Black circles indicate predicted pathways by the associated models while grey circles indicate pathways that were not recovered by models. The size of circles corresponds the pathway abundance information.

24. Blei, D.M., Kucukelbir, A., McAuliffe, J.D.: Variational inference: A review for statisticians. arXiv preprint arXiv:1601.00670 (2016)
25. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. Journal of machine Learning research **3**(Jan), 993–1022 (2003)

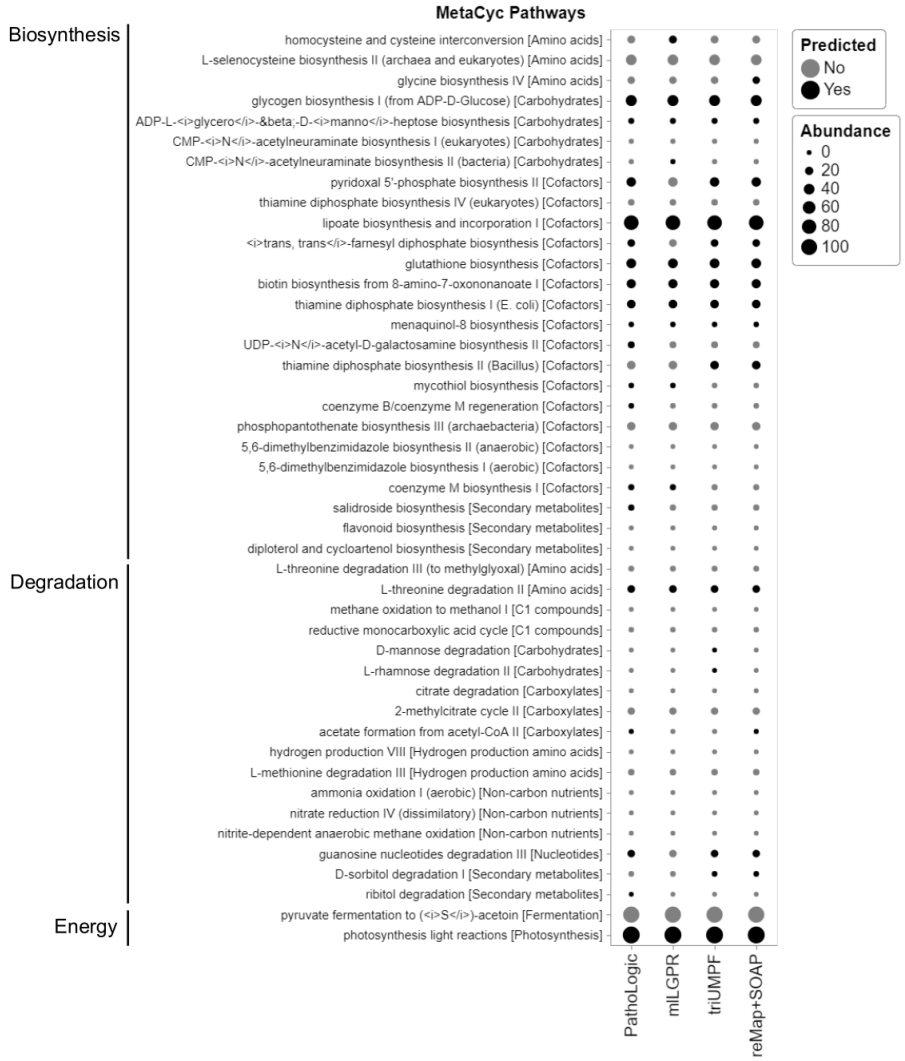


Fig. 12: Comparative study of predicted pathways for HOTS 110m dataset between PathoLogic, mlLGPR, triUMPF, and reMap+SOAP. Black circles indicate predicted pathways by the associated models while grey circles indicate pathways that were not recovered by models. The size of circles corresponds the pathway abundance information.

26. Caspi, R., Billington, R., Ferrer, L., et al.: The metacyc database of metabolic pathways and enzymes and the biocyc collection of pathway/genome databases. Nucleic Acids Research **44**(D1), D471–D480 (2016)

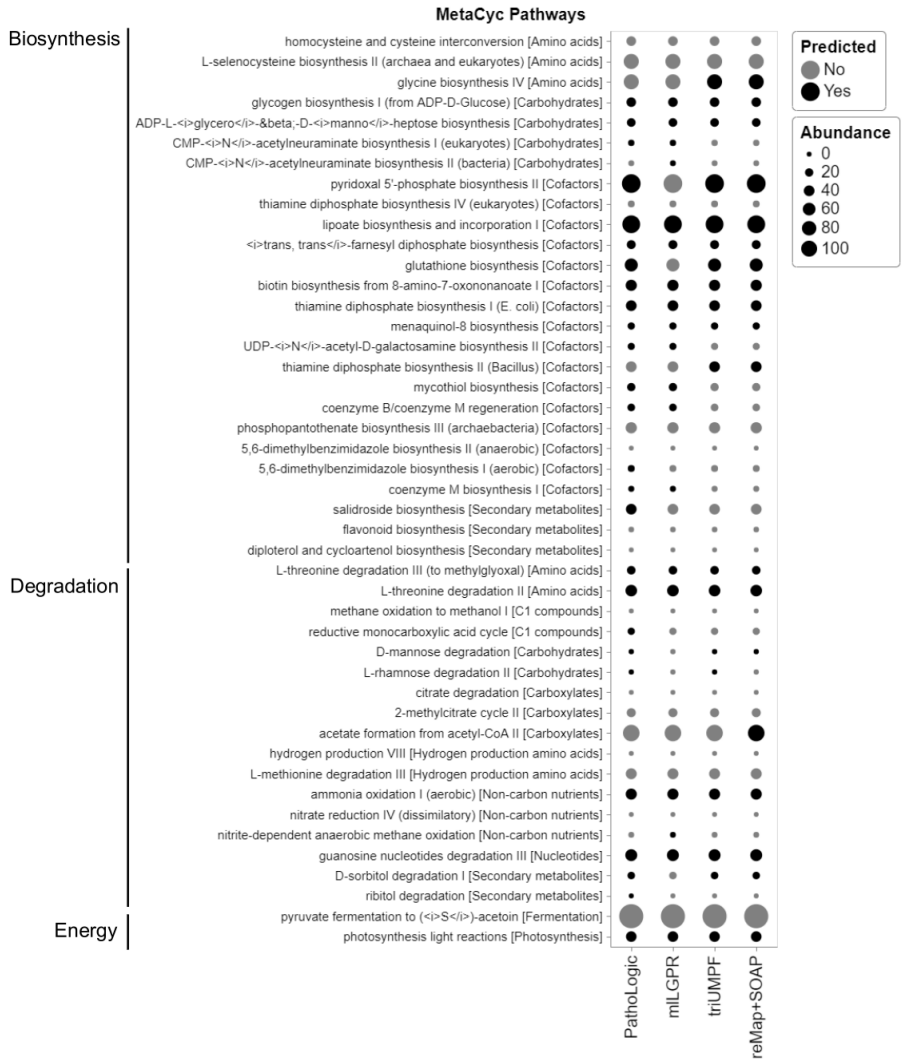


Fig. 13: Comparative study of predicted pathways for HOTS 500m dataset between PathoLogic, mlLGPR, triUMPF, and reMap+SOAP. Black circles indicate predicted pathways by the associated models while grey circles indicate pathways that were not recovered by models. The size of circles corresponds the pathway abundance information.

27. Caspi, R., Billington, R., Foerster, H., et al.: Biocyc: Online resource for genome and metabolic pathway analysis. The FASEB Journal **30**(1 Supplement), lb192–lb192 (2016)

28. Caspi, R., Billington, R., Keseler, I.M., Kothari, A., Krummenacker, M., Midford, P.E., Ong, W.K., Paley, S., Subhraveti, P., Karp, P.D.: The metacyc database of metabolic pathways and enzymes-a 2019 update. Nucleic acids research (2019)
29. Chang, H.S., Learned-Miller, E., McCallum, A.: Active bias: Training more accurate neural networks by emphasizing high variance samples. In: Advances in Neural Information Processing Systems. pp. 1002–1012 (2017)
30. He, J., Hu, Z., Berg-Kirkpatrick, T., et al.: Efficient correlated topic modeling with topic embedding. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 225–233. ACM (2017)
31. Hoffman, M.D., Blei, D.M., Wang, C., Paisley, J.W.: Stochastic variational inference. Journal of Machine Learning Research **14**(1), 1303–1347 (2013)
32. Hong, L., Davison, B.D.: Empirical study of topic modeling in twitter. In: Proceedings of the first workshop on social media analytics. pp. 80–88. acm (2010)
33. Kanehisa, M., Furumichi, M., Tanabe, M., et al.: Kegg: new perspectives on genomes, pathways, diseases and drugs. Nucleic Acids Research **45**(D1), D353–D361 (2017)
34. Karp, P.D., Latendresse, M., Paley, S.M., et al.: Pathway tools version 19.0 update: software for pathway/genome informatics and systems biology. Briefings in bioinformatics **17**(5), 877–890 (2016)
35. Kullback, S., Leibler, R.A.: On information and sufficiency. The annals of mathematical statistics **22**(1), 79–86 (1951)
36. Lin, T., Tian, W., Mei, Q., Cheng, H.: The dual-sparse topic model: mining focused topics and focused terms in short text. In: Proceedings of the 23rd international conference on World wide web. pp. 539–550. ACM (2014)
37. Luo, C., Zhan, J., Xue, X., et al.: Cosine normalization: Using cosine similarity instead of dot product in neural networks. In: International Conference on Artificial Neural Networks. pp. 382–391. Springer (2018)
38. M. A. Basher, A.R., Hallam, S.J.: Leveraging heterogeneous network embedding for metabolic pathway prediction. Bioinformatics (10 2020). https://doi.org/10.1093/bioinformatics/btaa906
39. M. A. Basher, A.R., McLaughlin, R.J., Hallam, S.J.: Metabolic pathway inference using multi-label classification with rich pathway features. PLOS Computational Biology **16**(10), 1–22 (10 2020)
40. M. A. Basher, A.R., McLaughlin, R.J., Hallam, S.J.: Metabolic pathway prediction using non-negative matrix factorization with improved precision. In: Computational Advances in Bio and Medical Sciences. pp. 33–44. Springer International Publishing, Cham (2021)
41. M. A. Basher, A.R., Nallan, A.N., McLaughlin, R.J., et al.: leads: improved metabolic pathway inference based on active dataset subsampling. bioRxiv (2020). https://doi.org/10.1101/2020.09.14.297424
42. McCutcheon, J.P., Von Dohlen, C.D.: An interdependent metabolic patchwork in the nested symbiosis of mealybugs. Current Biology **21**(16), 1366–1372 (2011)
43. Mimno, D.M., Hoffman, M.D., Blei, D.M.: Sparse stochastic inference for latent dirichlet allocation. In: Proceedings of the 29th International Conference on Machine Learning, ICML 2012, Edinburgh, Scotland, UK, June 26 - July 1, 2012 (2012)
44. Ng, A.Y., Jordan, M.I., Weiss, Y.: On spectral clustering: Analysis and an algorithm. In: Advances in neural information processing systems. pp. 849–856 (2002)
45. Sczyrba, A., Hofmann, P., Belmann, P., et al.: Critical assessment of metagenome interpretation—a benchmark of metagenomics software. Nature methods **14**(11), 1063 (2017)

46. Stewart, F.J., Sharma, A.K., Bryant, J.A., et al.: Community transcriptomics reveals universal patterns of protein sequence conservation in natural microbial communities. Genome biology **12**(3), R26 (2011)
47. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Advances in neural information processing systems. pp. 3104–3112 (2014)
48. Von Luxburg, U.: A tutorial on spectral clustering. Statistics and computing **17**(4), 395–416 (2007)
49. Walt, S.v.d., Colbert, S.C., Varoquaux, G.: The numpy array: a structure for efficient numerical computation. Computing in Science & Engineering **13**(2), 22–30 (2011)
50. Xu, Y., Wang, J., An, S., et al.: Semi-supervised multi-label feature selection by preserving feature-label space consistency. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management. pp. 783–792. ACM (2018)
51. Ye, Y., Doak, T.G.: A parsimony approach to biological pathway reconstruction/inference for genomes and metagenomes. PLoS Comput Biol **5**(8), e1000465 (2009)
52. Zhang, M.L., Zhou, Z.H.: A review on multi-label learning algorithms. IEEE transactions on knowledge and data engineering **26**(8), 1819–1837 (2014)
53. Zhao, W.X., Jiang, J., Weng, J., et al.: Comparing twitter and traditional media using topic models. In: European conference on information retrieval. pp. 338–349. Springer (2011)