reMap: Relabeling Multi-label Pathway Data with Bags to Enhance Predictive Performance

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

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

Abstract

We present reMap (relabeling multi-label pathway data based on bag approach), a simple, and yet, generic framework, that performs relabeling examples to a different set of labels, characterized as bags. A bag is comprised of a subset of correlated pathways, and a pathway is allowed to be mixed over bags, constituting an overlapping pathway over a subset of bags. Bag based approach was followed to overcome low sensitivity scores of triUMPF for the pathway prediction task. The relabeling process in reMap is achieved by alternating between 1) assigning bags to each sample and 2) updating reMap's parameters. reMap's effectiveness was evaluated on metabolic pathway prediction where resulting performance metrics equaled or exceeded other prediction methods on organismal genomes with improved sensitivity score.
Availability and implementation: The software package is published on github.com/reMap.
Contact: shallam@mail.ubc.ca

1 Introduction

Recently, triUMPF [3] 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 an alternative approach called reMap that relabels each example with a new label set called "bag" which can employed by an appropriate pathway prediction algorithm to improve prediction performance. A bag is defined as a subset of correlated pathways (from a pathway label set) and a pathway may be inter-mixed across bags with different proportions, resulting in an overlapping pathway over a subset of bags, i.e., non-disjoint bags.

2 Methods

The reMap pipeline (the top panel in Fig. 1) iteratively alternates between the following two phases:
Phase 1. Feed-Forward. During this stage, a minimal subset of bags is picked to tag each example in a three-step process:

1. **Constructing Bag Pathway.** In this step, pathways are partitioned into non-disjoint bags using any correlated models defined in Supp. Section 2. Note that this step is applied once during the initial round.
2. **Building Bag Centroid.** Having obtained a set of bags, reMap computes centroids for each bag to harness the relative association of each pathway to each bag's centroid. This ensure that pathways within a bag are semantically "close enough" to the center of that bag while also ensuring similar semantics among bags with overlapping pathways. To extract semantical information represented by pathways, the pathway2vec framework [1] is applied to automatically generate features for pathway inference. At the end of this step, reMap produces a maximum number of expected hypothetical bags for each example.
3. **Re-assigning Pathway Labels to Bags.** In this step, each example in a given dataset is re-annotated with a set of optimum multi-label bags using centroids and a history profile that stores bags probability is obtained from previous iterations.

Phase 2. Feed-Backward. After the feed-forward phase, reMap's internal parameters are updated by enforcing four constraints: i)- similarities between bags and the associated pathways weights, ii)- pathways enclosed in a bag should share similar weights, iii) the input space (i.e., enzyme) and the pathway space should share similar statistical properties, and iv) all reMap' parameters should not be too large or too small.

The two phases are repeated to all examples, until a predefined number of rounds ($\tau \in \mathbb{Z}_{>1}$) is reached. At the end, a new dataset is constructed which can used to train a mutli-label algorithm (e.g. leADS [2]) for pathway prediction. See Supp. Section 1 for mathematical derivations and optimization of the reMap algorithm.

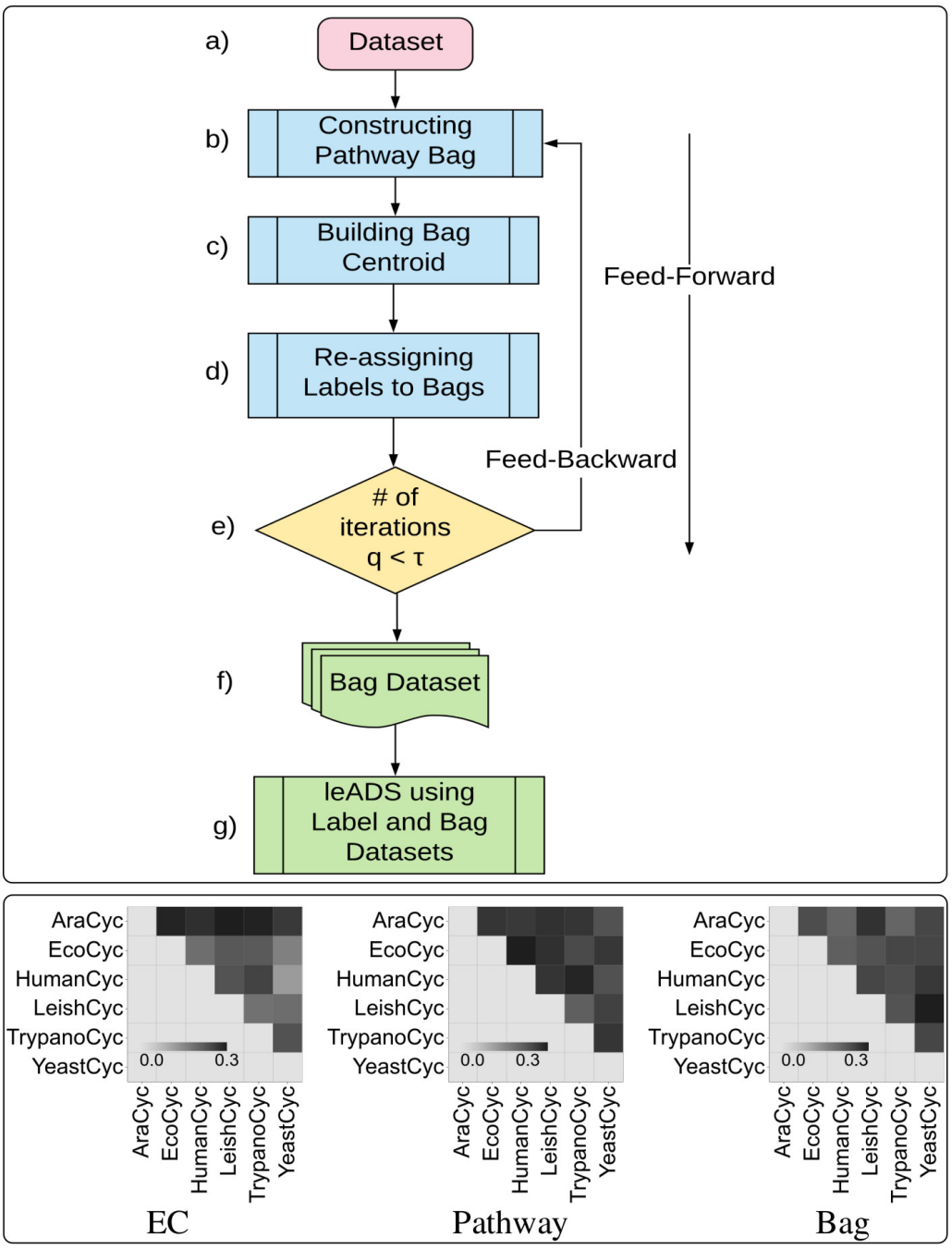


Figure 1: The top figure demonstrates the two-phased reMap workflow: feed-forward and feed-backward. The forward phase is consists of three components: (b) pathway bag construction to build correlated bags given multi-label data (a), (c) bag centroid determination that retrieves centroids of bags, and (d) pathway label re-assignment for each sample to bags. The feed-backward phase (e) optimizes reMap's parameters to maximize accuracy of mapping examples to bags. The process is repeated τ times. If the current iteration q reaches a desired number of rounds τ, the training is terminated while producing the multi-label bag data (f). The bag dataset then can be used to train a multi-label learning algorithm (e.g. leADS) (g). The bottom figure shows pairwise similarity scores (measured by cosine distance) among golden T1 data according to EC, pathway, and bag spaces for reMap+SOAP.

3 Results

We applied 6 golden tier 1 (T1) datasets described previously in [4] to demonstrate the effectiveness of relabeling pathways with bags for the purpose of pathway prediction following the configurations outlined in Supp. Section 3. Here, we focus on SOAP (sparse correlated bag pathway) based bag pathway construction which will be referred to as **reMap+SOAP**. For discussions about other correlated models and additional experiments, please see Supp. Section 4. The pathway prediction results are reported using four evaluation metrics after 10 trials: *Hamming loss*, *average precision*, *average recall*, and *average F1 score*. As shown in Supp. Table 4, reMap+SOAP attained high average F1 scores on HumanCyc (0.8285) and TrypanoCyc (0.7305), achieving $\sim$ 10% performance gain over mlLGPR's predictions (0.7468 and 0.6768 on the same datasets respectively [4]). Moreover, reMap+SOAP outperforms triUMPF on all golden T1 data in terms of sensitivity scores, hence, demonstrating the smooth labeling of pathways to bags by reMap. Relabeling smoothness is indicated as a heat-map figures in Fig. 1 (bottom panel) showing the pairwise distance (using the cosine distance metric) among golden T1 data according to EC, pathway, and bag spaces where a higher score (or higher intensity) between a pair of data entails high similarity, and vice versa. Similarity scores among golden T1 data in the resulted bag space is effected by a combination of EC and pathway spaces due to the four constraints (feed-backward phase) explained in Section 2. As an example, the similarity score for EcoCyc and HumanCyc in the EC space seems to be far distant than in the pathway space. Therefore, during the bag annotation process, information about ECs and pathways is combined and propagated to the bag space which can be seen in the bottom panel of Fig. 1. This observation has important implications for maximizing the bag annotation quality beyond even current performance benefits of reMap.

4 Conclusion

reMap offers an alternative approach to inferring metabolic pathways from organismal or multi-organismal genomes with high sensitivity and precision scores by relabeling pathways to bags and then using an appropriate multi-label algorithm (e.g. leADS) to predict pathways given bags.

Funding

This work was performed under the auspices of Genome Canada, Genome British Columbia, the Natural Science and Engineering Research Council (NSERC) of Canada, and Compute/Calcul Canada. ARMAB was supported by UBC four-year doctoral fellowships (4YF) administered through the UBC Graduate Program in Bioinformatics.
Conflict of Interest: SJH is a co-founder of Koonkie Inc., a bioinformatics consulting company that designs and provides scalable algorithmic and data analytics solutions in the cloud.

References

[1] Abdur Rahman MA Basher and Steven J Hallam. Leveraging heterogeneous network embedding for metabolic pathway prediction. *bioRxiv*, feb 2020.

[2] Abdur Rahman MA Basher and Steven J Hallam. `https://github.com/hallamlab/leADS`, 2020.

[3] Abdur Rahman MA Basher, Ryan J McLaughlin, and Steven J Hallam. Incorporating triple nmf with community detection to metabolic pathway inference. *bioRxiv*, 2020.

[4] Abdur Rahman MA Basher, Ryan J McLaughlin, and Steven J Hallam. Metabolic pathway inference using multi-label classification with rich pathway features. *bioRxiv*, February 2020.

Supplementary - reMap: Relabeling Multi-label Pathway Data with Bags to Enhance Predictive Performance

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

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

We introduce reMap pipeline that annotates each example with a new label set, called "bag" set, inspired by the multi-graph classification (MGC) technique [32], to strike the balance between precision and recall for the pathway prediction model. Each bag is comprised of correlated pathways from a pathway label set, which are allowed to be inter-mixed across bags with different proportions, resulting in overlapping subset of pathways over a subset of bags (non-disjoint). Hence, this approach is fundamentally different than triUMPF [4] that follows the clustering strategy (on pathway and enzyme graphs). Moreover, the bag-based technique has an important implication for the pathway prediction task (in Fig. 1) . That is, unlike mlLGPR [5] and triUMPF, the bag based pathway prediction applies two consecutive steps, where a set of bags are inferred at first then pathways within these bags are recovered. Since pathways are distributed over bags, it is conceivable that a pathway may be revisited multiple times, thereby, increasing the chance of boosting sensitivity scores while improving precision. This potential benefit was not leveraged in mlLGPR and neither triUMPF.

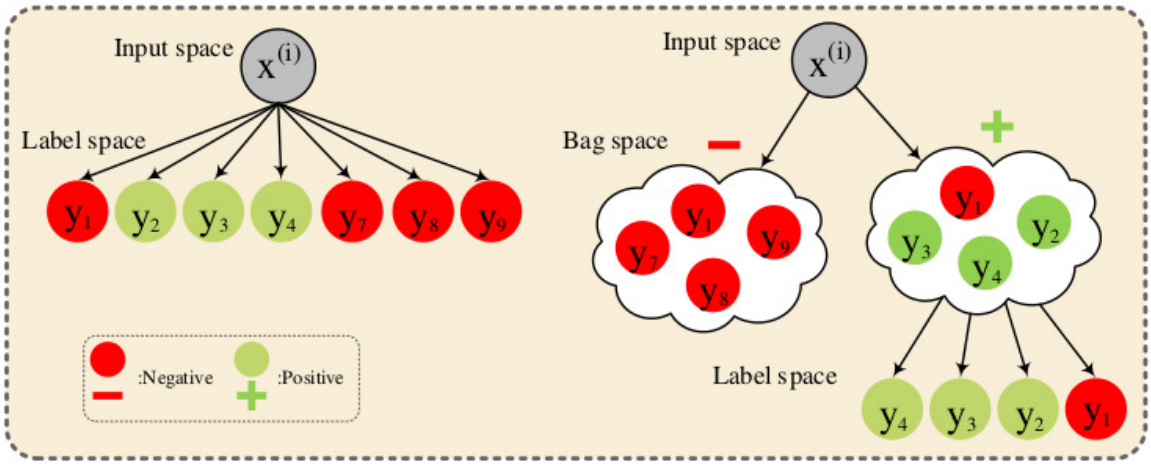


Figure 1: **The traditional vs the proposed bag based multi-label classification approaches**. The traditional supervised multi-label classification is displayed on the left panel, where labels (i.e., red or green colors) are associated with an input instance $\mathbf{x}^{(i)}$. This approach sought to predict a set of labels for $\mathbf{x}^{(i)}$ without considering any compartmentalization of labels. On contrary, bag based multi-label classification approach, on the right, applies two steps, where it predicts a set of positive bags (depicted as a cloud glyph), at first, then the labels within these bags are predicted (green colored labels).

This material is divided into three parts: i)- the reMap framework (Section 1), ii)- descriptions about correlated models (Section 2), iii)- experimental settings (Section 3), and iv)- empirical analysis (parameter sensitivity and visualization for correlated models, assessing prediction probability history during relabeling process of reMap, and metabolic pathway prediction effectiveness on golden T1 dataset) (Section 4).

1 The reMap Method

In this section, we provide important notations and definitions that used throughout this paper and then state our 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 sample 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 dataset definition.

Definition 1.1. Multi-label Pathway Dataset ([5]). 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. 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 $\mathbf{x}^{(i)}$ and $\mathbf{y}^{(i)}$ are symbolized as $\mathbf{X}$ and $\mathbf{Y}$, respectively. ■

Both $\mathcal{E}$ and $\mathcal{Y}$ are extracted from reliable knowledge-bases (e.g. KEGG [18] and MetaCyc [10]). In this paper, we adopt MetaCyc. Moving on, We define the term *bag*, which is borrowed from multi-graph classification (MGC) technique [32]. In MGC, the goal is to learn a model from a set of labeled bags each containing several graphs. A bag is tagged positive, if at least one graph in the bag is positive, and negative otherwise. Here, we slightly abuse the term and reserve it to describe a composition of several correlated pathways.

Definition 1.2. Pathway Bag. Denote $\mathcal{B} = \{\mathbf{B}_1, \mathbf{B}_2, ..., \mathbf{B}_b\}$ a set with b bags, where each bag $\mathbf{B}_c \in \{-1, +1\}^t$ 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. 1.1. The presence or absence of a pathway in bag c is indicated by $+1$ or -1, respectively. The matrix representation of $\mathcal{B}$ is $\mathbf{B} \in \{-1, +1\}^{t \times t}$. ■

Bags 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 bag association with j-th bag, where a larger score indicates both bags are highly correlated. This correlation can be discovered using either SOAP or SPREAT given pathway datasets, $(\mathbf{Y})$ introduced in Def. 1.1, incorporating pathway abundance information which can be obtained by mapping enzyme with abundances onto pathways. Both models are a form of mixed-membership hierarchical Bayesian network, where each example is encoded as a vector of bag probabilities and each bag, in turn, is comprised of a set of correlated pathways. The pathways are permitted to be inter-mixed across bags with different proportion, resulting in overlapping pathways over bags.

These two models extend the functions of CTM (correlated topic model) [7] by incorporating *dual sparseness* and *supplementary pathways* in modeling bag proportions. That is, SOAP and SPREAT encodes missing probable pathways as "background" without compromising the original pathway labels, i.e., $\mathbf{Y}$. The supplementary pathways are stored in a matrix $\mathbf{M} \in \mathbb{Z}_{\geq 0}^{n \times t}$, where each entry is an integer value indicating the abundance of the pathway associated with a specific sample and n corresponds the number of samples. These background pathways can be modeled as in the case of SPREAT or employed directly as in SOAP. With regard to dual sparseness, it is applied to encourage selection of: i)- few focused bags for each individual example and ii)- few focused pathways for each bag. For thorough discussions about these models, see Section 2.

As a result of correlation, we define the following two terminologies: *bag feature vector* and *bag's neighbor*.

Definition 1.3. Bag Feature Vector. Given an example i, the associated bag feature vector is indicated by $\mathbf{d}^{(i)} \in \{-1, +1\}^b$, where $\mathbf{d}_j^{(i)} = +1$, iff the bag j is observed for the sample i and $\mathbf{d}_j = -1$ otherwise. The matrix form in represented as $\mathbf{D} \in \{-1, +1\}^{n \times b}$. ■

***Example* 1.3.1.** An example of feature vectors for bags is illustrated in Fig. 2, where 2-dimensional feature vectors for bags encode presence or absence of two bags $\mathbf{B}_1$ and $\mathbf{B}_2$, given a set of 6 pathways and bag-pathway association information, depicted as a cloud glyph.

Definition 1.4. Bag's Neighbors. A bag $\mathbf{B}_c \in \mathcal{B}$ is said to be a neighbor to another bag $\mathbf{B}_j \in \mathcal{B}$ s.t. $c \neq j$, iff there exits an intersected pathway l in both bags, i.e., $\mathbf{B}_{c,l} \wedge \mathbf{B}_{j,l} = 1$. ■

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

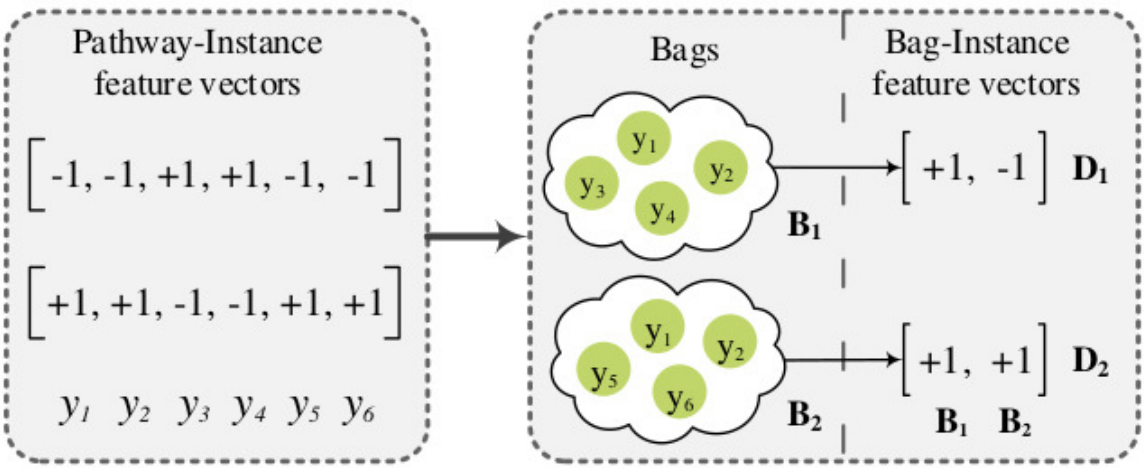


Figure 2: **An example of feature vectors for bags.** The subfigure in the left represents the feature vector for six pathways corresponding to two instances. The right subfigure indicates two bags, $\mathbf{B}_1$ and $\mathbf{B}_2$, and their features for the same two instances, where the first sample, $\mathbf{D}_1$, suggests that $\mathbf{B}_1$ is positive because the corresponding pathways y_3 and y_4 are present, while the bag feature vector for the second example, $\mathbf{D}_2$, suggests that both bags are present.

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

To better crystallize the idea of incorporating bags in the pathway prediction, consider the following example.

***Example* 1.4.1.** Fig. 1 illustrates the benefit of incorporating bags for multi-label pathway classification (right panel). Here, a dataset consists of two bags, each of which groups 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 bag, indicated by $+$, then classify pathways within that bag. In contrast, the traditional multi-label classification approaches (left figure), mostly based on *binary relevance* technique, proceeds on predicting multiple pathway labels for $\mathbf{X}_i$.

Relabeling a multi-label dataset $\mathcal{S}$ will result in another dataset of $\mathcal{S}_{bag}$.

Definition 1.5. Multi-label Bag Dataset. A bag dataset is represented by $\mathcal{S}_{bag} = \{(\mathbf{x}^{(i)}, \mathbf{d}^{(i)}) : 1 < i \leqslant n\}$ consisting of n examples. $\mathbf{d}^{(i)} = [d_1^{(i)}, ..., d_t^{(i)}] \in \{-1, +1\}^b$ is a bag label vector of size b. Each element of $\mathbf{d}^{(i)}$ indicates the presence/absence of the associated bag that is inherited from a set $\mathcal{B} = \{\mathbf{B}_1, \mathbf{B}_2, ..., \mathbf{B}_b\}$. Each bag $\mathbf{B}_c \in \{-1, +1\}^t$ is presumed to contain a subset of correlated pathways, i.e., $\mathcal{Y}_c \subseteq \mathcal{Y}$. The presence or absence of a pathway in bag c is indicated by $+1$ or -1, respectively. The matrix representation of $\mathcal{B}$ is $\mathbf{B} \in \{-1, +1\}^{t \times t}$. In addition, bags are correlated through overlapping pathways, i.e., for two correlated bags $\mathbf{B}_c$ and $\mathbf{B}_j$ s.t. $c \neq j$, there exits an intersected pathway l in both bags, i.e., $\mathbf{B}_{c,l} \wedge \mathbf{B}_{j,l} = +1$. The matrix form of bag label vector with n instances is denoted by $\mathbf{D}$ ($\in \mathbb{Z}_{\geq 0}^{n \times b}$). ■

Now, we outline the reMap framework (depicted in Fig. 3), which iteratively alternates between the following two phases: i)- feed-forward, where it consists of three components: 1)- constructing pathway bag, 2)- building bag centroid, 3)- re-assigning labels to each example; and ii)- feed-backward to update reMap's parameters.

1.1 Feed-Forward Phase

During this phase, a minimal subset of bags is picked to tag each example according to the following three steps.

1.1.1 Constructing Pathway Bag

In this step, the pathways in $\mathcal{S}$ are partitioned into non-disjoint b bags using any correlated bag models (CTM, SOAP or SPREAT). In addition to bags, the covariance matrix ($\Sigma \in \mathbb{R}^{b \times b}$) can be obtained which is transformed to a correlation matrix $\rho = C^{-1} \Sigma C^{-1}$, where $C = \sqrt{\text{diag}(\Sigma)}$. The correlated bag models also provide us with pathway distribution over bags, denoted by $\Phi \in \mathbb{R}^{b \times t}$, where we trim the pathway distribution $\Phi' \in \mathbb{R}^{b \times k} (\subseteq \Phi)$ by keeping top k pathways for each bag, provided that aggregation of k pathways from all bags is equal to t.

Modeling pathway distribution and bag's correlations have two important implications: First, organisms encoding similar functions may share similar bags, thus, encouraging to have near-identical statistical strength (as in triUMPF); and Second, pathways that are observed to be frequently occurring together may suggest a similar relative contribution to a bag.

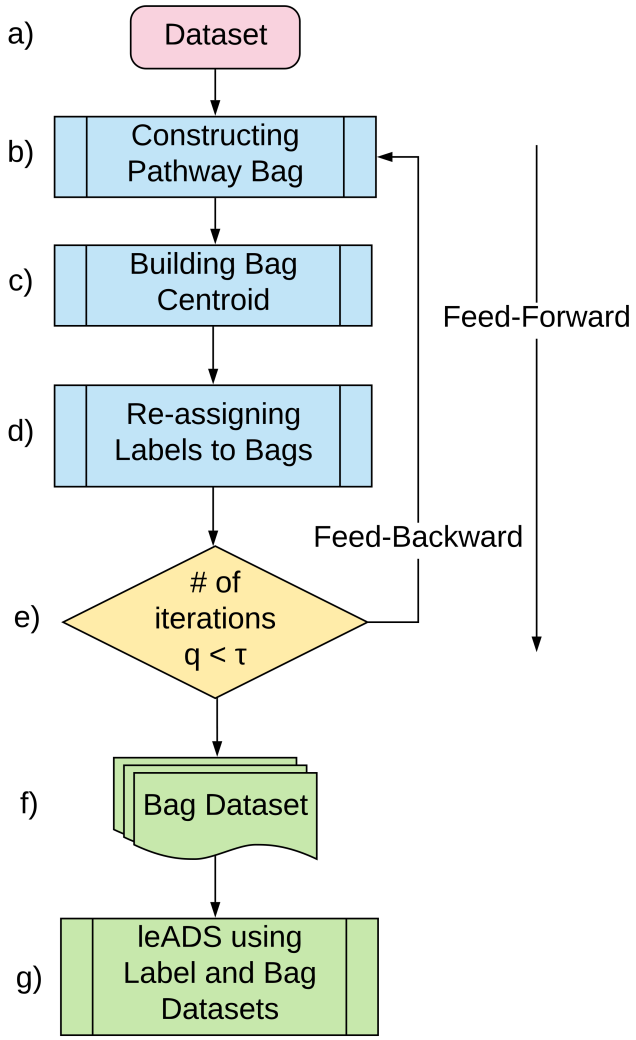


Figure 3: **A workflow diagram showing the proposed reMap pipeline to relabel a multi-label data**. The method consists of two phases: feed-forward and feed-backward. The forward phase is composed of three components: (b) pathway bag construction that aims to build correlated bags given multi-label data (a), (c) building bag centroid that retrieves centroids of bags, and (d) re-assigning pathway labels, for each sample, to bags. The feed-backward phase (e) optimizes reMap's parameters to maximize accuracy of mapping examples to bags. The process is repeated τ times. If the current iteration q reaches a desired number of rounds τ, the training is terminated while producing the final $\mathcal{S}_{bag}$ data points (f). The bag dataset then can be used to train a multi-label learning algorithm (e.g. leADS ([3])) (g).

1.1.2 Building Bag Centroid

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

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

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. 1.1;
  $\mathbf{C}_{s,:} = \mathbf{c}_s$;
Return **C**

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

Having obtained a set of bags, we compute centroids of each bag to harness the relative association of each pathway to each bag's centroid. This procedure is described in Algorithm 1. Specifically, We assume that pathways in a bag are semantically "close enough" to the center of that bag and, hence, pathways in that bag should share similar low-level representation while also ensuring similar semantics among bags with overlapping pathways. Using this simple assumption, reMap computes the centroid $\mathbf{c}_s$ of a bag s according to:

$$\mathbf{c}_s = \frac{\alpha}{n_s} \sum_{j \in \mathbf{B}_{s,j}=+1} \frac{\mathbf{P}_j}{||\mathbf{P}_j||} \tag{1.1}$$

where $\mathbf{P} \in \mathbb{R}^{t \times m}$ is a matrix storing pathway representation obtained from pathway2vec framework [2], n_s is the number of associated pathways to bag s, $||.||$ is the length of a feature vector, and α is a hyper-parameter, which is determined by empirical evaluation (16 in this work). It is important to note that bags centroids enable us to obtain a maximum number of expected bags for a given example by applying a suitable metric, such as cosine similarity [23] as:

$$\begin{aligned} \widehat{\mathbf{D}^{(i)}} =& \texttt{vec}\left(\left\{\mathbb{I}\left(\frac{\mathbf{c}_s^\top \tilde{\mathbf{c}}_s^{(i)}}{||\mathbf{c}_s|| \cdot ||\tilde{\mathbf{c}}_s^{(i)}||} \geq \upsilon\right) : 1 \leq s \leq b\right\}\right) \\ \tilde{\mathbf{c}}_s^{(i)} =& \frac{\alpha}{n_s} \sum_{j \in \mathbf{Y}_{i,j} \wedge \mathbf{B}_{s,j}=+1} \frac{\mathbf{P}_j}{||\mathbf{P}_j||} \end{aligned} \tag{1.2}$$

where $\mathbb{I}(.)$ is an indicator function that results in either $+1$ or -1 depending on a user-defined threshold $\upsilon \in \mathbb{R}_{>0}$, $\tilde{\mathbf{c}}_s^{(i)}$ is the centroid for bag s, and $\widehat{\mathbf{D}^{(i)}}$ is an aggregated hypothetical bags for an example i of size b after the results are vectorized into a series of $+1$ and -1 using `vec` operation. Algorithm 2 describes the pseudocode for Eq. 1.2.

1.1.3 Re-assigning Labels to Bags

The goal of this step is to map an example's input space onto a bag space using a decision function $h^{\mathbf{bag}}$ which would produce an optimum multi-label bag set ($\widehat{\mathbf{D}^{\mathbf{opt}}} \in \mathbb{Z}_{\geq 0}^{n \times b}$). Formally, let us denote a set of bags that are picked to relabel an example as $\mathcal{B}_{\mathbf{P}}^{(i)} \subseteq \arg\{\widehat{\mathbf{D}_{i,j}} = +1 : \forall j\}$ while we denote $\mathcal{B}_{\mathbf{U}}^{(i)}$ the set of remaining bags, where $\widehat{\mathbf{D}^{(i)}}$ obtained using Eq. 1.2. Collectively, both sets of bags are stored in $\mathcal{L}^{(i)} = \{\mathcal{B}_{\mathbf{P}}^{(i)} \cup \mathcal{B}_{\mathbf{U}}^{(i)}\}$. Then, the process of re-annotation is achieved iteratively, mirroring sequential learning and prediction strategy [14], where for each example, a bag $\mathbf{B}_j$ 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 bags, represented by $\mathcal{L}_q^{(i)} = \mathcal{L}_{q-1}^{(i)} \ominus \{\mathbf{B}_j : 1 < j \leq |\mathcal{B}_{\mathbf{P}}^{(i)}|\}$.

More concretely, at each iteration q, we estimate the probability of an example i given the selected bags at $q-1$, using threshold closeness (TC) metric [13]:

$$p\Big(\mathbf{x}^{(i)} | \mathbf{H}_{q-1}^{(i)}, \mathcal{L}_{q-1}^{(i)}, \widehat{\mathbf{D}_{i,j}} = +1\Big) = \frac{\bar{p}_{\mathbf{H}_{q-1}^{(i)}}\Big(\widehat{\mathbf{D}_{i,j}} | \mathcal{L}_{q-1}^{(i)}, \mathbf{x}^{(i)}\Big) G + \zeta}{Z} \tag{1.3}$$

where $G = 1 - \bar{p}_{\mathbf{H}_{q-1}^{(i)}}(\widehat{\mathbf{D}_{i,j}} | \mathcal{L}_{q-1}^{(i)}, \mathbf{x}^{(i)})$, $\widehat{\mathbf{D}_{i,j}} = +1$ if bag $\mathbf{B}_j$ is in example i, and $\bar{p}_{\mathbf{H}_{q-1}^{(i)}}(\widehat{\mathbf{D}_{i,j}} | \mathcal{L}_{q-1}^{(i)}, \mathbf{x}^{(i)})$ is the average probability of classifying $\mathbf{x}^{(i)}$ into the bag $\mathbf{B}_j$ over values collected in $\mathbf{H}_{q-1}^{(i)}$ that 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. The term ζ is a smoothness constant while 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.

To estimate $p(\widehat{\mathbf{D}_{i,j}} | \mathcal{L}_{q-1}^{(i)}, \mathbf{x}^{(i)})$, we jointly compute the probability of bags and pathways that are associated with $\widehat{\mathbf{D}_{i,j}}$ at round $q-1$ as:

$$\begin{aligned} p(\widehat{\mathbf{D}_{i,j}} | \mathcal{L}_{q-1}^{(i)}, \mathbf{x}^{(i)}) \propto& \mathbf{H}_{q-1}^{(i)} \left(\sum_{e \in \mathcal{L}_{q-1}^{(i)}} z_{j,e} \left(\sum_{s \in \mathbf{B}_{j,s}=+1} p(\widehat{\mathbf{D}_{i,j}} | l_s = +1, \Theta_j^{\mathbf{bag}}) p(y_s^{(i)} | \mathbf{x}^{(i)}, \Theta_s^{\mathbf{path}}) \right) \right) \\ z_{j,e} =& \frac{\rho_{j,e} - \min(\rho)}{\max(\rho) - \min(\rho)} \\ p(\widehat{\mathbf{D}_{i,j}} | l_s = +1, \Theta_j^{\mathbf{bag}}) =& \frac{1}{1 + e^{-\Theta_j^{\mathbf{bag},\mathrm{T}} \left| \tilde{\mathbf{c}}_j^{(i)} - \mathbf{P}_s \right|}} \\ p(y_s^{(i)} | \mathbf{x}^{(i)}, \Theta_s^{\mathbf{path}}) =& \frac{1}{1 + e^{-\Theta_s^{\mathbf{path},\mathrm{T}} \mathbf{x}^{(i)}}} \end{aligned} \tag{1.4}$$

where $y_s^{(i)} = 1$ if the pathway index s is found to be present in both bag j and in sample $\mathbf{x}^{(i)}$ and 0 otherwise, and $l_s = 1$ if the pathway index s is associated to bag j and 0 otherwise. $z_{j,e}$ is a normalized correlation between bags j and e, respectively, obtained from ρ (see Section 1.1.1) and $\bar{\mathbf{c}}_j^{(i)}$ is presented in Eq 1.2. $\Theta_j^{\mathbf{bag}} \in \mathbb{R}^m$ and $\Theta_s^{\mathbf{path}} \in \mathbb{R}^r$ denote parameters for the bag j and the pathway s model's, respectively, and are learned during the feed-backward stage in Section 1.2.

Inputs :
n: number of samples ($n \in \mathbb{N}_{>2}$)
$\mathbf{X}$: 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 bags ($\mathcal{B} = \{\mathbf{B}_1, ..., \mathbf{B}_b\}$)
$\mathbf{C}$: the centroids of bags ($\mathbf{C} \in \mathbb{R}^{b \times m}$)
v: a cutoff hyper-parameter ($v \in \mathbb{R}_{\geq 0}$)

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

Process :

```
// an empty matrix that will contain maximum
// number of bags for each example
```
$\widehat{\mathbf{D}} \leftarrow 0$;
```
// n is the number of samples
```
for $i \leftarrow 1$ **to** n **do**
  `// b is the number of bags`
  for $k \leftarrow 1$ **to** b **do**
    for $j \leftarrow 1$ **to** $\mathbf{Y}_{i,:}$ **do**
      if $j \in \mathbf{Y}_{i,j} \wedge \mathbf{B}_{k,j}$ **then**
        $\widehat{\mathbf{D}}_{i,k} \leftarrow$ [apply Eq 1.2];

Return $\widehat{\mathbf{D}}$

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

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

Inputs :

n: number of samples ($n \in \mathbb{N}_{>2}$)
$\mathbf{X}$: 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 bags ($\widehat{\mathbf{D}} \in \mathbb{Z}_{\geq 0}^{n \times b}$)
$\mathcal{B}$: a set of b bags ($\mathcal{B} = \{\mathbf{B}_1, ..., \mathbf{B}_b\}$)
$\mathbf{P}$: pathway features matrix ($\mathbf{P} \in \mathbb{R}^{t \times m}$)
$\Theta^{\mathbf{bag}}$: bags' parameters ($\Theta^{\mathbf{bag}} \in \mathbb{R}^{b \times m}$)
$\Theta^{\mathbf{path}}$: pathways' parameters ($\Theta^{\mathbf{path}} \in \mathbb{R}^{t \times r}$)
$\mathbf{C}$: the centroids of bags ($\mathbf{C} \in \mathbb{R}^{b \times m}$)
z: normalized bags's correlation ($z \in \mathbb{R}^{b \times b}$)
α: a hyper-parameter for bags' centroids construction ($\alpha \in \mathbb{R}_{>0}$)
d: a subsample 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 bags ($v \in \mathbb{R}_{\geq 0}$)
ζ: a decision threshold for selecting bags ($\zeta \in \mathbb{R}_{\geq 0}$)
τ: number of rounds ($\tau \in \mathbb{N}_{>1}$)

Outputs:

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

Process :

```
// an empty matrix that will contain maximum
// number of bags for each example
```

$\widehat{\mathbf{D}^{\mathbf{opt}}} \leftarrow 0$;
for $i \leftarrow 1$ **to** n **do**
  `// an initial set of bags for an example i`
  $\mathcal{L}^{(i)} = \arg\{\widehat{\mathbf{D}_{i,j}} = +1 : \forall j\}$;
  `// sub sample d bags from` $|\mathcal{L}^{(i)}|$ `bags`
  $b_{sub} \leftarrow$ randomly select d bags 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;
      $\texttt{tmp}_1 = 0$;
      for $e \leftarrow 1$ **to** $|\mathcal{L}^{(i)}|$ **do**
        $\texttt{tmp}_2 = 0$;
        for $k \leftarrow 1$ **to** $|B_j|$ **do**
          $p_j = \frac{1}{1+e^{-\Theta_j^{\mathbf{bag},\mathrm{T}} \left| \bar{\mathbf{c}}_j^{(i)} - \mathbf{P}_k \right|}}$;
          $p_k = \frac{1}{1+e^{-\Theta_k^{\mathbf{path},\mathrm{T}} \mathbf{x}^{(i)}}}$;
          $\texttt{tmp}_2 = \texttt{tmp}_2 + p_j p_k$;
        $\texttt{tmp}_1 = \texttt{tmp}_1 + z_{j,e} \times \texttt{tmp}_2$;
      $\mathbf{H}_{q,j}^{(i)} = z_{j,e} \times \texttt{tmp}_1$;
      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 bags`
  `// from` $\mathcal{L}^{(i)}$ `bags into +1, -1 indicating presence/absence of bags`
  $\widehat{\mathbf{D}_i^{\mathbf{opt}}} \leftarrow \texttt{transform}(\mathcal{L}^{(i)})$;
Return $\widehat{\mathbf{D}^{\mathbf{opt}}}$

Algorithm 3: $\textsc{Relabel2Bag}(n, \mathbf{X}, \mathbf{Y}, \widehat{\mathbf{D}}, \mathcal{B}, \mathbf{P}, \Theta^{\mathbf{bag}}, \Theta^{\mathbf{path}}, \mathbf{C}, z, \alpha, d, \epsilon, v, \zeta, \tau)$

1.2 Feed-Backward Phase

Here, we set up a learning framework for computing reMap's bags and pathways parameters, jointly denoted as $\Theta = \{\Theta^{\mathbf{bag}}, \Theta^{\mathbf{path}}\}$. From Eq. 1.3, it is clear that our algorithm has three learning components: i)- a hyper-plane in the bag feature space to absorb bag correlation, ii)- a hyper-plane in the pathway feature space to encode semantic information about pathways, and iii)- a joint learning between bags and pathways to exploit bag-pathway relationship.

Let us define three empirical loss functions, corresponding the three components: $\epsilon^{\mathbf{bag}} : \{0,1\}^b \rightarrow \mathbb{R}_{\geq 0}$, $\epsilon^{\mathbf{path}} : \{0,1\}^t \rightarrow \mathbb{R}_{\geq 0}$, and $\epsilon^{\mathbf{bag\text{-}path}} : \{0,1\}^b \rightarrow \mathbb{R}_{\geq 0}$ of margin $\mathbf{d}h^{\mathbf{bag}}(\mathbf{x})$, $\mathbf{y}h^{\mathbf{path}}(\mathbf{x})$, and $\mathbf{d}h^{\mathbf{bag\text{-}path}}(y)$, respectively, where $h^{(.)}$ 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. 1.4:

$$\widehat{\Theta} = \underset{\Theta}{\operatorname{argmax}} \prod_{q=1}^{q=\tau} \prod_{i=1}^{i=n} \mathbf{H}_{q-1}^{(i)} \prod_{j=1}^{j=b} p(\widehat{\mathbf{D}_{i,j}} | \mathcal{L}_{q-1}^{(i)}, \mathbf{x}^{(i)}) \times \left(\sum_{s \in \mathbf{B}_{j,s}=+1} p(\widehat{\mathbf{D}_{i,j}} | l_s = +1, \Theta_j^{\mathbf{bag}}) p(y_s^{(i)} | \mathbf{x}^{(i)}, \Theta_s^{\mathbf{path}}) \right) \right) \quad (1.5)$$

Estimation of parameters in Eq. 1.5 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 bags were considered. These simplified treatments provide an efficient way to optimize the parameters, where we adopt the "one-vs-all" scheme learning for each bag and pathway [35].

In addition, we apply four constraints to retrieve a good set of parameters: i)- enforcing similarity between bags' and the associated pathway labels' parameters; ii)- weights of pathways, assembled in a bag, should be close to each other; iii) the input space and the label space should share similar local topological structure, which entails that if two instances in feature space exhibit strong association then they may share the same label set in the label space; and iv) the reMap' parameters should not be too large or too small.

Using the four constraints, the obtained an optimum bag set ($\widehat{\mathbf{D}^{\mathbf{opt}}}$) in Section 1.1.3 and the given pathway label set ($\mathbf{Y}$), the objective function is formulated according to:

$$\begin{aligned}
\min_{\Theta^{\mathbf{bag}}, \mathbf{W}, \mathbf{U}, \mathbf{S}} & - \sum_{q \in \tau} \sum_{i \in n} \sum_{j \in b} \sum_{k \in \mathbf{B}_{j,k}=+1} v_i \log \left(p(\widehat{\mathbf{D}_{i,j}} | l_k = +1, \Theta_j^{\mathbf{bag}}) \right) \\
& + \sum_{q \in \tau} \sum_{i \in n} ||\mathbf{y}^{(i)} - \widehat{\mathbf{d}^{(i)}} \mathbf{W}||_2^2 + \sum_{j \in b} C(\Theta_j^{\mathbf{bag}}) \\
\min_{\Theta^{\mathbf{path}}, \mathbf{W}, \mathbf{S}} & - \sum_{q \in \tau} \sum_{i \in n} \sum_{k \in t} v_i \log \left(p(y_k^{(i)} | \mathbf{x}^{(i)}, \Theta_k^{\mathbf{path}}) \right) \\
& + \sum_{q \in \tau} \sum_{i \in n} ||\mathbf{y}^{(i)} - \widehat{\mathbf{d}^{(i)}} \mathbf{W}||_2^2 + \sum_{k \in t} C(\Theta_k^{\mathbf{path}})
\end{aligned} \quad (1.6)$$

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\Big(p(\widehat{\mathbf{D}_{i,j}}|l_k = +1, \Theta^{\mathbf{bag}}_j)\Big) =& \log\Big(1 + e^{-\mathbf{d}^{(i)}_j \Theta^{\mathbf{bag}\intercal}_j |\bar{\mathbf{c}}^{(i)}_j - \mathbf{P}_k|}\Big) \\
\triangleq& \ \epsilon^{\mathbf{bag\text{-}path}}(\mathbf{d}^{(i)}_j h^{\mathbf{bag\text{-}path}}_j(l^{(i)}_k), \Theta^{\mathbf{bag}}_j) \\
-\log p(\mathbf{y}^{(i)}_k|\mathbf{x}^{(i)}, \Theta^{\mathbf{path}}_k) =& \log\Big(1 + e^{-\mathbf{y}^{(i)}_k \Theta^{\mathbf{path}\intercal}_k \mathbf{x}^{(i)}}\Big) \\
\triangleq& \ \epsilon^{\mathbf{path}}(\mathbf{y}^{(i)}_k h^{\mathbf{path}}_k(\mathbf{x}^{(i)}), \Theta^{\mathbf{path}}_k) \\
l^{\mathbf{bag}}(\mathbf{d}^{(i)}_j h^{\mathbf{bag}}_j(\mathbf{x}^{(i)}), \Theta^{\mathbf{bag}}_j) \triangleq& \ \epsilon^{\mathbf{bag\text{-}path}}(\mathbf{d}^{(i)}_j h^{\mathbf{bag\text{-}path}}_j(l^{(i)}_k), \Theta^{\mathbf{bag}}_j) \\
&+ \sum_{k \in \mathbf{B}_{j,k}=+1} \epsilon^{\mathbf{path}}(\mathbf{y}^{(i)}_k h^{\mathbf{path}}_k(\mathbf{x}^{(i)}), \Theta^{\mathbf{path}}_k) \\
C(\Theta^{\mathbf{bag}}_j) =& \underbrace{\sum_{k \in \mathbf{B}_{j,k}=+1} ||\mathbf{U}^\intercal \Theta^{\mathbf{path}}_k - \Theta^{\mathbf{bag}}_j||^2_2}_{\text{pathways within a bag}} + \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 bags}} \\
&+ \lambda_2 ||\Theta^{\mathbf{bag}}_j||_{2,1} + \lambda_3 ||\mathbf{U}||_{2,1} + \frac{1}{2}\kappa ||\mathbf{S}1 - 1||^2_2 \\
C(\Theta^{\mathbf{path}}_k) =& \underbrace{\sum_{q \in \mathbf{B}_{j,q}=+1} ||\Theta^{\mathbf{path}}_q - \Theta^{\mathbf{path}}_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{path}\intercal}_k \mathbf{x}^{(q)} - \Theta^{\mathbf{path}\intercal}_k \mathbf{x}^{(l)}||^2_2}_{\text{correlated instances of a pathway}} \\
&+ \lambda_5 ||\Theta^{\mathbf{path}}_k||_{2,1} + \frac{1}{2}\kappa ||\mathbf{S}1 - 1||^2_2
\end{aligned}
\tag{1.7}
$$

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 sample $\mathbf{x}^{(i)}$ to emphasize selection of informative samples, and $\lambda_{[1,2,3,4,5]}$ are hyper-parameters controlling the relative contributions of the associated constraint terms. Let us explain all the terms involved in Eq. 1.7. The function $||\mathbf{U}^\intercal \Theta^{\mathbf{path}}_k - \Theta^{\mathbf{bag}}_j||^2_2$ reflects the first constraint, where it enforces similarities between pathways, associated to a bag j, and the bag 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 $||\Theta^{\mathbf{path}}_q - \Theta^{\mathbf{path}}_k||^2_2$ considers the similarities among pathway labels, grouped under a specific bag. 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 $||\Theta^{\mathbf{path}\intercal}_k \mathbf{x}^{(q)} - \Theta^{\mathbf{path}\intercal}_k \mathbf{x}^{(l)}||^2_2$.

The term $||\mathbf{y}^{(i)} - \widehat{\mathbf{d}^{(i)}}\mathbf{W}||^2_2$ maintains the integrity of both bag and pathway label vectors on example i, thus, encouraging bags to have similar contents as pathway labels, and $\mathbf{W} \in \mathbb{R}^{b \times t}$ captures the correlation between bags and pathway labels. 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 bag label vectors, $\widehat{\mathbf{d}^{(q)}}$ and $\widehat{\mathbf{d}^{(l)}}$, and the two pathway label 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 $\mathbf{S}_{q,k} \in \mathbb{R}^{n \times n}_{\geq 0}$, where a high score, indicates both examples have near identical pathway labels and, hence, should have similar bags, and vice-versa hold as well.

The formula $||\Theta^{\mathbf{path}\intercal}_k \mathbf{x}^{(q)} - \Theta^{\mathbf{path}\intercal}_k \mathbf{x}^{(l)}||^2_2$ addresses the neighborhood relationship in the feature space between $\mathbf{x}^{(q)}$ and $\mathbf{x}^{(k)}$, as characterized by $\mathbf{S}_{q,l}$ score [33]. 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 bags, hence, mitigating from the negative influences of imperfectly labeling bags. The terms $||\Theta^{\mathbf{bag}}_j||_{2,1}$, $||\Theta^{\mathbf{path}}_j||_{2,1}$, and $||\mathbf{U}||_{2,1}$, constitute the fourth constraint in the form of $L_{2,1}$-norm on all weights, which reduces the negative impact of label correlations in the model while at the same time 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)- bag-projection weight matrix $\mathbf{W}$, 2)- pathway-projection weight matrix $\mathbf{U}$, 3)- bag-specific weight matrix $\Theta^{\mathbf{bag}}$, 4)- pathway-specific weight matrix $\Theta^{\mathbf{path}}$, 5)- instance-similarity specific weight matrix $\mathbf{S}$, and 6)- bag-specific updating matrix $\widehat{\mathbf{D}}$. The last parameter is a binary matrix indicating the presence/absence of bags in the training dataset, which is gradually updated based on the gradient score strategy.

Unfortunately, the objective function in Eq. 1.6 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 ($\mathbf{W}, \mathbf{U}, \Theta^{\mathbf{bag}}, \Theta^{\mathbf{path}}, \mathbf{S}$, and $\widehat{\mathbf{D}}$) at a time while the others are held constant. The partial derivative of each term in Eq. 1.6 is a positive semi-definitive, hence, the whole term is jointly convex, which leads to the following independent optimization problems for all pathways and bags classifiers according to the multi-label 1-vs-All approach [35].

- **Update W**. The gradient of Eq. 1.6 w.r.t. $\mathbf{W}$ has the following formula:

$$\nabla \mathbf{W} = \frac{2}{nb}\Big(\widehat{\mathbf{D}}^{\intercal}\widehat{\mathbf{D}}\mathbf{W} - \widehat{\mathbf{D}}^{\intercal}\mathbf{y}\Big) + \lambda_3 \mathbf{K}_{\mathbf{W}}\mathbf{W} \tag{1.8}$$

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. 1.6 w.r.t. $\mathbf{U}$ becomes:

$$\begin{aligned} \nabla \mathbf{U} =& \frac{1}{b}\sum_{j\in b} \frac{2}{\sum_k \mathbb{I}(\mathbf{B}_{j,k}=+1)} \sum_{k\in \mathbf{B}_{j,k}=+1} \Big(\Theta_k^{\mathbf{path}}\Theta_k^{\mathbf{path}\intercal}\mathbf{U} - \Theta_k^{\mathbf{path}}\Theta_j^{\mathbf{bag}\intercal}\Big) \\ &+ \lambda_3 \mathbf{K}_{\mathbf{U}}\mathbf{U} \end{aligned} \tag{1.9}$$

where,

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

- **Update $\Theta^{\mathbf{bag}}$**. The partial derivative for each bag, say $\Theta_j^{\mathbf{bag}}$, is:

$$\begin{aligned} \nabla \Theta_j^{\mathbf{bag}} =& \frac{1}{n}\sum_{i\in n} v_i \left(\frac{1}{\sum_k \mathbb{I}(\mathbf{B}_{j,k}=+1)} \sum_{k\in \mathbf{B}_{j,k}=+1} \frac{-\widehat{\mathbf{D}_{i,j}}|\tilde{\mathbf{c}}_j^{(i)} - \mathbf{P}_k|}{1+e^{\widehat{\mathbf{D}_{i,j}}\Theta_j^{\mathbf{bag}\intercal}|\tilde{\mathbf{c}}_j^{(i)}-\mathbf{P}_k|}} \right) \\ &+ \frac{1}{\sum_k \mathbb{I}(\mathbf{B}_{j,k}=+1)} \sum_{k\in \mathbf{B}_{j,k}=+1} \Big(-2\widehat{\mathbf{U}}^{\intercal}\Theta_k^{\mathbf{path}} + 2\Theta_j^{\mathbf{bag}} \Big) + \lambda_2 \frac{\Theta_j^{\mathbf{bag}}}{2||\Theta_j^{\mathbf{bag}}||_2} \end{aligned} \tag{1.10}$$

where $\widehat{\mathbf{U}}$ obtained from Eq. 1.9.

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

$$\begin{aligned} \nabla \Theta_k^{\mathbf{path}} =& \frac{1}{n}\sum_{i=1}^{i=n} v_i \left(\frac{-\mathbf{y}_k^{(i)}\mathbf{x}^{(i)}}{1+e^{\mathbf{y}_k^{(i)}\Theta_k^{\mathbf{path}\intercal}\mathbf{x}^{(i)}}} \right) + 2\widehat{\mathbf{U}}\widehat{\mathbf{U}}^{\intercal}\Theta_k^{\mathbf{path}} - \frac{2}{b}\sum_{j\in b}\widehat{\mathbf{U}}\widehat{\Theta}_j^{\mathbf{bag}} \\ &+ \frac{1}{b}\sum_{j\in b} \frac{2}{\sum_k \mathbb{I}(\mathbf{B}_{j,k}=+1)} \sum_{q\in \mathbf{B}_{j,q}=+1} \Big(\Theta_k^{\mathbf{path}} - \Theta_q^{\mathbf{path}}\Big) \\ &+ \mathbf{X}^{\intercal}\mathbf{L}\mathbf{X}\Theta_k^{\mathbf{path}} + \lambda_5 \frac{\Theta_k^{\mathbf{path}}}{2||\Theta_k^{\mathbf{path}}||_2} \end{aligned} \tag{1.11}$$

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 $\frac{1}{2}\sum_{q,l\in n} \mathbf{S}_{q,l}||\Theta_k^{\mathbf{path}\intercal}(\mathbf{x}^{(q)} - \mathbf{x}^{(l)})||_2^2 = \mathbf{tr}\big(\Theta_k^{\mathbf{path}\intercal}\mathbf{X}^{\intercal}\mathbf{L}\mathbf{X}\Theta_k^{\mathbf{path}}\big)$. Following the work of [27], it is important for practical purpose to normalize the graph Laplacian, to account for the fact that some examples are more similar than others [30]: $\bar{\mathbf{L}} \triangleq \mathbf{M}^{-1/2}\mathbf{L}\mathbf{M}^{-1/2} = \mathbf{I} - \mathbf{M}^{-1/2}\mathbf{S}\mathbf{M}^{-1/2}$. Adhering to this property, we consider the following formula: $\frac{1}{2}\sum_{q,l\in n} \mathbf{S}_{q,l}||\Theta_k^{\mathbf{path}\intercal}\big(\frac{\mathbf{x}^{(q)}}{\sqrt{\mathbf{M}_{q,q}}} - \frac{\mathbf{x}^{(l)}}{\sqrt{\mathbf{M}_{l,l}}}\big)||_2^2 = \mathbf{tr}\big(\Theta_k^{\mathbf{path}\intercal}\mathbf{X}^{\intercal}\bar{\mathbf{L}}\mathbf{X}\Theta_k^{\mathbf{path}}\big)$.

- **Update S**. Given the updated values of $\widehat{\mathbf{U}}$ and $\widehat{\Theta}^{\mathbf{path}}$, we obtain the equivalent objective function of Eq. 1.6 with the terms only related to $\mathbf{S}$ as:

$$\min_{0 \leq \mathbf{S}_{q,j} \leq 1} \lambda_1 \text{tr}(\widehat{\mathbf{D}}^\intercal \mathbf{L} \widehat{\mathbf{D}}) + \lambda_4 \text{tr}(\mathbf{Y}^\intercal \mathbf{L} \mathbf{Y}) + \text{tr}(\widehat{\Theta^{\mathbf{path}}} \mathbf{X}^\intercal \mathbf{L} \mathbf{X} \widehat{\Theta^{\mathbf{path}}}^\intercal) + \kappa ||\mathbf{S}1 - 1||_2^2 \tag{1.12}$$

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:

$$\nabla \mathbf{S} = \lambda_1 \widehat{\mathbf{D}} \widehat{\mathbf{D}}^\intercal + \lambda_4 \mathbf{Y} \mathbf{Y}^\intercal + \mathbf{X} \widehat{\Theta^{\mathbf{path}}}^\intercal \widehat{\Theta^{\mathbf{path}}} \mathbf{X}^\intercal + 2\kappa(\mathbf{S} - 1) \tag{1.13}$$

As we have mentioned, the proposed similarity matrix $\mathbf{S}$ captures reliable and discriminative locality information in the projected feature space, and this information is utilized to optimize correlations in the predicted label space, which ensures feature-label space consistency. Consequently, a set of bags can be inferred with high fidelity, for each example, if features with labels correlation information is disseminated to features with bags correlations, thus, alleviating the effects of imperfectly detecting negative bags. That is what exactly $\mathbf{S}$ is expected to behave.

- **Update $\widehat{\mathbf{D}}$**. We iteratively update bags, where a set of bags are added or removed at each round. In particular, a positive subset of bags 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 bags are considered to be negative to that example. While it is relatively easy to compile a set of positive bags for an example, however, bags not belonging to that example are too diverse to be considered as negative. Thus, it is better to consider the remaining bags 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:

$$\begin{aligned} \nabla \widehat{\mathbf{D}} = & \frac{1}{n} \sum_{i \in n} v_i \sum_{j \in b} \left(\frac{1}{\sum_k \mathbb{I}(\mathbf{B}_{j,k} = +1)} \sum_{k \in \mathbf{B}_{j,k} = +1} \frac{-\Theta_j^{\mathbf{bag}\intercal} |\tilde{\mathbf{c}}_j^{(i)} - \mathbf{P}_k|}{1 + e^{\widehat{\mathbf{D}_{i,j}} \Theta_j^{\mathbf{bag}\intercal} |\tilde{\mathbf{c}}_j^{(i)} - \mathbf{P}_k|}} \right) \\ & + \lambda_1 \mathbf{L}^\intercal \widehat{\mathbf{D}} + \frac{2}{nb} \Big(\widehat{\mathbf{D}} \mathbf{W} \mathbf{W}^\intercal - \lambda_3 \mathbf{y} \mathbf{W}^\intercal \Big) \end{aligned} \tag{1.14}$$

After getting the gradient score, we ascertain the labels of examples based on:

$$\widehat{\mathbf{D}_{i,j}} = \begin{cases} +1 & \text{if } \nabla \widehat{\mathbf{D}_{i,j}} \geq 1 \\ 0 & \text{if } 0 < \nabla \widehat{\mathbf{D}_{i,j}} < 1 \\ -1 & \text{if } \nabla \widehat{\mathbf{D}_{i,j}} \leq 1 \end{cases} \tag{1.15}$$

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

The pseudocode for this phase in presented in Algorithm 4

Inputs :

n: number of samples ($n \in \mathbb{N}_{>2}$)
$\mathbf{X}$: input space training set ($\mathbf{X} \in \mathbb{R}^{n \times r}$)
$\mathbf{Y}$: pathway space training set ($\mathbf{Y} \in \mathbb{Z}_{\geq 0}^{n \times t}$)
$\widehat{\mathbf{D}}$: bag 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 bags ($\mathbf{C} \in \mathbb{R}^{b \times m}$)
z: normalized bags's correlation ($z \in \mathbb{R}^{b \times b}$)
d: a subsample 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:

$\Theta^{\mathbf{bag}}$: bags' parameters ($\Theta^{\mathbf{bag}} \in \mathbb{R}^{b \times m}$)
$\Theta^{\mathbf{path}}$: pathways' parameters ($\Theta^{\mathbf{path}} \in \mathbb{R}^{t \times r}$)
$\mathbf{W}$: bag-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 :

`// bags' parameters (`$\Theta^{\text{bag}} \in \mathbb{R}^{b \times m}$`)`
$\Theta^{\mathbf{bag}} \leftarrow 0$;
`// pathways' parameters (`$\Theta^{\text{path}} \in \mathbb{R}^{t \times r}$`)`
$\Theta^{\mathbf{path}} \leftarrow 0$;
for $q \leftarrow 1$ **to** ξ **do**
  $\widehat{\mathbf{D}^q} \leftarrow$ Relabel2Bag(n, $\mathbf{X}$, $\mathbf{Y}$, $\widehat{\mathbf{D}^{q-1}}$, $\mathcal{B}$, $\mathbf{P}$, $\widehat{\Theta^{\mathbf{bag},q-1}}$, $\widehat{\Theta^{\mathbf{path},q-1}}$, $\mathbf{C}$, z, λ);
  `// update W's parameters using Eq. 1.8`
  $\mathbf{W}^q \leftarrow \mathbf{W}^{q-1} - \gamma \nabla \mathbf{W}^q$;
  `// update U's parameters using Eq. 1.9`
  $\mathbf{U}^q \leftarrow \mathbf{U}^{q-1} - \gamma \nabla \mathbf{U}^q$;
  `// update` Θ^{bag}`'s parameters using Eq. 1.10`
  $\Theta^{\mathbf{bag},q} \leftarrow \Theta^{\mathbf{bag},q-1} - \gamma \nabla \Theta^{\mathbf{bag},q}$;
  `// update` Θ^{path}`'s parameters using Eq. 1.11`
  $\Theta^{\mathbf{path},q} \leftarrow \Theta^{\mathbf{path},q-1} - \gamma \nabla \Theta^{\mathbf{path},q}$;
  `// update S's parameters using Eq. 1.13`
  $\mathbf{S}^q \leftarrow \mathbf{S}^{q-1} - \gamma \nabla \mathbf{S}^q$;
  `// update` $\widehat{\mathbf{D}}$ `using gradient score strategy`
  $\widehat{\mathbf{D}^q} \leftarrow$ apply Eq. 1.15;
Return $\Theta^{\mathbf{bag}}$, $\Theta^{\mathbf{path}}$, $\mathbf{W}$, $\mathbf{U}$, $\mathbf{S}$

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

1.3 Closing the loop

The two phases are repeated to all samples, until a predefined number of rounds (τ) is reached. At the end, a new dataset is constructed, $\mathcal{S}_{bag} = \{(\mathbf{x}^{(i)}, \widehat{\mathbf{D}^{(i),\mathbf{opt}}}) : 1 < i \leqslant n\}$ consisting of n examples and $\widehat{\mathbf{D}^{\mathbf{opt}}}$ is an optimum set of bags that contain a small number of b bags annotated to each example.

2 Correlated Models

In this section, we present the CBT (correlated bag-pathway) package that consists of three hierarchical Bayesian (parametric) mixture models: i)- CTM (correlated topic model), ii)- SOAP (sparse correlated bag pathway), and iii)- SPREAT (distributed sparse correlated bag pathway) that incorporate pathway abundance information to encode each example as a mixture distribution of bags, and each bag, in turn, is a mixture of pathways with different mixing proportions.

But first let us provide some background and notations. We note that each mathematical symbol is only related in the context of this section.

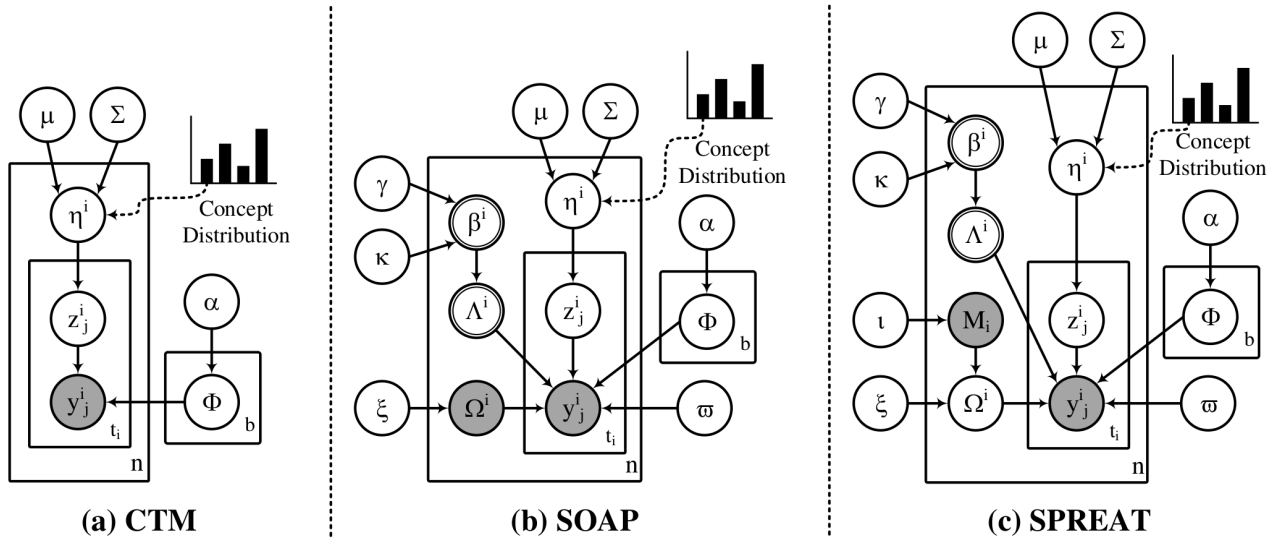


Figure 4: Graphical model representation of the correlated concept models. The boxes are "plates" representing replicates. The outer plate represents instances, while the inner plate represents the repeated choice of features within an example. The logistic normal distribution, used to model the latent concept proportions of an example, can represent capture 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 $\mathbf{M}_i$ while 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 $d_{i,j}$ are hidden variables.

Definition 2.1. 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 t known metabolic pathways, as defined from a trusted source, such as MetaCyc ([12]), and $\mathcal{Y}_i \subseteq \mathcal{Y}$ corresponds to a subset of true pathways associated with the sample i. ■

Recovering latent distributions of $\mathcal{P}$ mirrors the concept modeling paradigm, which was introduced to reconstruct the thematic structure, called "topics", from a corpus [9].

Definition 2.2. Concept Modeling. Given a collection of n samples, 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 report, the term concept is referred to as "bag" which is coined from the multi-graph classification (MGC) technique [32], where each bag is comprised of several correlated pathways. For brevity purposes, the following terms: *concept*, *topic*, or *bag*, 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) [9]. 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 2.3. 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 bag association with the bag j, where a larger score indicates both concepts are highly correlated. ■

However, due to the inherent noise in curating $\mathcal{P}$ there are situations where a set of pathways may not included in $\mathcal{P}$ for a set of examples. An alternative way to incorporate missing pathways is to record them in a separate list while keeping the original pathway collection intact for further investigation. Lets us denote a matrix $\mathbf{M} \in \mathbb{Z}_{\geq 0}^{n \times t}$ which stores a set of missing pathways where each entry is an integer value indicating the abundance of the pathway associated with a specific sample. The matrix $\mathbf{M}$ is referred to as a "background" or "supplementary" set, analogous to studies in [36, 17]. With these definitions, we describe the correlated models.

2.1 Correlated Topic Model

The correlated topic model (CTM) is a probabilistic graphical model that extends the generative story of LDA [9] to incorporate correlation among concepts. Fig. 4 (a) shows the Bayesian graphical model for CTM using

plate notation. Like latent Dirichlet allocation [9], 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, in contrast to LDA, are assumed to be correlated to each other, as reflected by a Gaussian covariance matrix.

1. For each concept $a \in \{1, \ldots, b\}$:
   (a) Sample a distribution over features $\Phi_a \sim \text{Dir}(.|\alpha)$;
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 $\theta^{(i)} = \text{softmax}(\eta^{(i)})$;
   (c) For each feature $j \in \{1, \ldots, t^{(i)}\}$:
      i. Sample a concept assignment $z_j^{(i)} \sim \text{Mult}(.|\theta^{(i)})$;
      ii. Sample a feature $y_j^{(i)} \sim \text{Mult}(.|\Phi_{z_j^{(i)}})$;

Algorithm 5: The generative process for CTM given a collection

Formally, let n be the total number of a collection, where each example i consists of feature indices as $\mathbf{y}^{(i)}$. 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, as indicated by it's concept assignment, i.e., $y_j^{(i)} \sim \Phi_{z_j^{(i)}}$. The CTM's 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.

2.2 Correlated Bag-Pathway Model

The correlated bag pathway is an extension to CTM and comes in two flavors, as depicted in Fig. 4 (b) and (c): i)- sparse correlated bag pathway (SOAP) and ii)- distributed sparse correlated bag pathway (SPREAT). Both models incorporate dual sparseness and supplementary pathways in modeling bag proportions. These important properties are not adopted in CTM. In contrast to SOAP, SPREAT is equipped to reconstruct the latent distribution of supplementary pathways. Let us formally explain both models in detail. For the purpose of coherence, we use features to define pathways as before.

1. For each concept $a \in \{1, ..., b\}$:
   (a) Sample 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. Sample a vector $\mathbf{M}_i \sim \text{Prior}(.|\iota)$;
      ii. Sample 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. Sample a concept assignment $z_j^{(i)} \sim \text{Mult}(.|\Lambda^{(i)} \odot \theta^{(i)})$;
      ii. Sample 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

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. Similar to CTM, for each example i, a concept proportion is drawn $\theta^{(i)} = \text{softmax}(\eta^{(i)})$, 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, which may include interactions among them. Besides, a concept should cover only a few focused features, instead of absorbing all features. Thus, we borrow the idea from [22, 1, 26, 6, 15] 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 keep only the 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 minimizing 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 $\Omega_j^{(i)}$ 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.

By 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 (e.g., examples). The generative process of the proposed 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 former models.

2.3 Evidence Lower Bound (ELBO) for SPREAT

Here, we discuss inference for 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.

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 using the model's independence assumptions:

$$\begin{aligned} p(z, y, \eta, \Phi, \Lambda, \beta, \Omega | \mathbf{M}, \gamma, \kappa, \alpha, \iota, \xi, \beta) = & \left[\prod_{a=1}^{b} p(\Phi_a | \alpha) \right] \left[\prod_{i=1}^{n} p(\eta | \mu, \Sigma) p(\Lambda^{(i)} | \beta^i) p(\beta^i | \gamma, \kappa) \right. \\ & \times p(\Omega^{(i)} | \mathbf{M}^{(i)}, \xi) \left[\prod_{j=1}^{t_i} p(y_j^{(i)} | z_j^{(i)}, \Omega_j^{(i)}, \Lambda^{(i)}, \Phi, \varpi) \right. \quad (2.1) \\ & \left. \left. \times p(z_j^{(i)} | \eta) \right] \right] \end{aligned}$$

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

$$p(\Theta, \mathbf{V} | \mathbf{Y}, \mathbf{M}) = \frac{p(\mathbf{Y}, \mathbf{M}, \Theta, \mathbf{V})}{p(\mathbf{Y}, \mathbf{M})} \quad (2.2)$$

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:

$$p(\mathbf{Y}, \mathbf{M}) = \int_{\Theta} \int_{\mathbf{V}} p(\mathbf{Y}, \mathbf{M}, \Theta, \mathbf{V}) \quad (2.3)$$

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

Table 1: **Correspondence between variational and original parameters.**

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 variational inference, which has been extensively employed in many complex probabilistic models. Examples include latent Dirichlet allocation (LDA) [9], sparse LDA [26], supervised topic models [24], mixed membership stochastic blockmodels [1], nested hierarchical Dirichlet process [28], and many others. 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. 2.2. The closeness is commonly measured using Kullback–Leibler (KL) divergence [21]. 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. 2.3, 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:

$$\begin{aligned}
\log p(\mathbf{Y}, \mathbf{M}) &= \log \int_{\Theta} \int_{V} p(\mathbf{Y}, \mathbf{M}, \Theta, \mathbf{V}) \\
&= \log \int_{\Theta} \int_{\mathbf{V}} p(\mathbf{Y}, \mathbf{M}, \Theta, V) \frac{q(\Theta, \mathbf{V})}{q(\Theta, \mathbf{V})} \\
&= \log(\mathbb{E}_q[\frac{p(\mathbf{Y}, \mathbf{M}, \Theta, \mathbf{V})}{q(\Theta, \mathbf{V})}]) \\
&\geq \mathbb{E}_q[\log p(\mathbf{Y}, \mathbf{M}, \Theta, \mathbf{V})] + \mathbb{H}(q) \\
&\triangleq \mathcal{L}(q)
\end{aligned} \tag{2.4}$$

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 that is likely under the model, keeping both the priors and data in mind through the joint distribution. 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" [8]. 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:

$$\begin{aligned}
q(\eta, \Lambda, z, d, \beta, \Phi, \Omega) = \prod_{a=1}^{b} q(\Phi_a|\phi_a) \Bigg[\prod_{i=1}^{n} q(\eta^{(i)}|\nu, \zeta^2) q(\Lambda^{(i)}|\lambda^{(i)}) q(\Omega^{(i)}|\omega^{(i)}) \\
\prod_{j=1}^{j=t_i} q(z_j^{(i)}|\varsigma_j^{(i)}) \Bigg]
\end{aligned} \tag{2.5}$$

where $\phi, \nu, \zeta^2, \lambda, \omega$ and ς are variational free parameters. Table 1 shows the correspondence between variational and the original parameters.

Taking together, the first term in Eq. 2.4, $\mathbb{E}_q[\log p(\mathbf{Y}, \mathbf{M}, \Theta, V)]$, can be decomposed into:

$$\begin{aligned}
\mathbb{E}_q[\log p(\mathbf{Y}, \mathbf{M}, \Theta, V)] = \sum_{a=1}^{a=b} \mathbb{E}_q[\log p(\Phi_a|\alpha)] + \sum_{i=1}^{i=n} \Bigg(\mathbb{E}_q[\log p(\eta|\mu, \Sigma)] \\
+ \mathbb{E}_q[\log p(\Lambda^{(i)}|\beta^i)] + \mathbb{E}_q[\log p(\beta^i|\gamma, \kappa)] \\
+ \mathbb{E}_q[\log p(\Omega^{(i)}|\mathbf{M}^{(i)}, \xi)] \\
+ \sum_{j=1}^{j=t_i} \Big(\mathbb{E}_q[\log p(y_j^{(i)}|z_j^{(i)}, \Omega_j^{(i)}, \Lambda^{(i)}, \Phi, \varpi)] + \mathbb{E}_q[p(z_j^{(i)}|\eta)] \Big) \Bigg)
\end{aligned} \tag{2.6}$$

And, the second term $\mathbb{H}(q)$ in Eq. 2.4 can be defined as:

$$\begin{aligned}\mathbb{H}(q) = &-\sum_{a=1}^{a=b} \mathbb{E}_q[\log q(\Phi_a|\phi_a)] - \sum_{i=1}^{i=n} \Bigg(\mathbb{E}_q[\log q(\eta^{(i)}|\nu, \zeta^2)] + \mathbb{E}_q[\log q(\Lambda^{(i)}|\lambda^{(i)})] \\ &+ \mathbb{E}_q[\log q(\Omega^{(i)}|\omega^{(i)})] + \sum_{j=1}^{j=t_i} \mathbb{E}_q[\log q(z_j^{(i)}|\varsigma_j^{(i)})] \Bigg)\end{aligned} \tag{2.7}$$

2.3.1 Variational Lower Bound

Given Eq. 2.6, we derive expressions for each term:

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

$$\begin{aligned}\mathbb{E}_q[\log p(\Phi_a|\alpha)] &= \mathbb{E}_q[\log \mathrm{Dir}(\Phi_a|\alpha)] \\ &= \mathbb{E}_q\left[\log\left(\frac{\Gamma(\sum_{j=1}^{j=t}\alpha_j)}{\prod_{j=1}^{j=t}\Gamma(\alpha_j)}\prod_{j=1}^{j=t}\Phi_{a,j}^{\alpha_j-1}\right)\right] \\ &= \mathbb{E}_q\left[\log\left(\frac{\Gamma(\sum_{j=1}^{j=t}\alpha_j)}{\prod_{j=1}^{j=t}\Gamma(\alpha_j)}\right) + \sum_{j=1}^{j=t}\log\Phi_{a,j}^{\alpha_j-1}\right] \\ &= \log\Gamma\Big(\sum_{j=1}^{j=t}\alpha_j\Big) - \sum_{j=1}^{j=t}\log\Gamma(\alpha_j) + \sum_{j=1}^{j=t}(\alpha_j-1)\mathbb{E}_q[\log\Phi_{a,j}]\end{aligned} \tag{2.8}$$

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

$$\begin{aligned}\mathbb{E}_q[\log p(\eta|\mu,\Sigma)] &= \mathbb{E}_q\Big[\log\Big(\mathcal{N}(\eta|\mu,\Sigma)\Big)\Big] \\ &= \mathbb{E}_q\Big[\Big(\frac{1}{2}\log|\Sigma^{-1}| - \frac{b}{2}\log 2\pi - \frac{1}{2}(\eta-\mu)^\top\Sigma^{-1}(\eta-\mu)\Big)\Big] \\ &= \frac{1}{2}\log|\Sigma^{-1}| - \frac{b}{2}\log 2\pi \\ &\quad - \frac{1}{2}\Big(\mathrm{tr}(\mathrm{diag}(\zeta^2)\Sigma^{-1}) + (\nu-\mu)^\top\Sigma^{-1}(\nu-\mu)\Big)\end{aligned} \tag{2.9}$$

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

$$\begin{aligned}\mathbb{E}_q[\log p(\Lambda^{(i)}|\beta^{(i)})] &= \mathbb{E}_q\Big[\log \mathrm{Bernoulli}(\Lambda^{(i)}|\beta^{(i)})\Big] \\ &= \mathbb{E}_q\left[\log\left(\prod_{a=1}^{a=b}\beta_a^{(i),\Lambda_a^{(i)}}(1-\beta_a)^{(i),1-\Lambda_a^{(i)}}\right)\right] \\ &= \sum_{a=1}^{a=b}\Big(\lambda_a^{(i)}\log\beta_a^{(i)} + (1-\lambda_a^{(i)})\log(1-\beta_a^{(i)})\Big)\end{aligned} \tag{2.10}$$

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

$$\begin{aligned}\mathbb{E}_q[\log p(\beta^{(i)}|\gamma,\kappa))] &= \mathbb{E}_q\Big[\log \mathrm{Beta}(\beta^{(i)}|\gamma,\kappa)\Big] \\ &= \sum_{a=1}^{a=b}\Big((\gamma-1)\log(\beta_a^{(i)}) + (\kappa-1)\log(1-\beta_a^{(i)}) - \log(B(\gamma,\kappa)\Big)\end{aligned} \tag{2.11}$$

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

$$\begin{aligned}\mathbb{E}_q[\log p(\Omega_i|\mathbf{M}^{(i)},\xi)] &= \mathbb{E}_q\left[\log\left(\frac{\Gamma(\sum_{j=1}^{j=t}\xi_j+\mathbf{M}_j^{(i)})}{\prod_{j=1}^{j=t}\Gamma(\xi_j+\mathbf{M}_j^{(i)})}\prod_{j=1}^{j=t}\Omega_j^{(i),\xi_j+\mathbf{M}_j^{(i)}-1}\right)\right] \\ &= \log\Gamma\Big(\sum_{j=1}^{j=t}\xi_j+\mathbf{M}_j^{(i)}\Big) - \sum_{j=1}^{j=t}\log\Gamma(\xi_j+\mathbf{M}_j^{(i)}) \\ &\quad + \sum_{j=1}^{j=t}(\xi_j+\mathbf{M}_j^{(i)}-1)\mathbb{E}_q[\log\Omega_j^{(i)}]\end{aligned} \tag{2.12}$$

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

$$\begin{aligned}\mathbb{E}_q[\log p(y_j^{(i)}|z_j^{(i)},\Omega_j^{(i)},\Lambda^{(i)},\Phi,\varpi)] &= \mathbb{E}_q\Big[\log\Big(\prod_{c=1}^{c=t}\prod_{a=1}^{a=b}\varpi\Phi_{a,c}^{\mathbb{I}(a=z_{a,j}^{(i)}\wedge\Lambda_a^{(i)},c=y_j^{(i)}),(1-\Omega_c^{(i)})}\Big)\Big] \\ &= \log\varpi + \sum_{c=1}^{c=t}\sum_{a=1}^{a=b}\Big(y_{j,c}^{(i)}\varsigma_{a,j}^{(i)}\lambda_a^{(i)}\mathbb{E}_q[(1-\Omega_c^{(i)})]\mathbb{E}_q[\log\Phi_{a,j}]\Big)\end{aligned} \tag{2.13}$$

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

$$\begin{aligned}\mathbb{E}_q[\log p(z_j^{(i)}|\eta)] &= \mathbb{E}_q\Big[\log\Big(\frac{\exp(\eta^\top(\mathrm{diag}(z_j^{(i)}))}{\sum_{k=1}^{k=b}\exp(\eta_k)}\Big)\Big] \\ &= \mathbb{E}_q\Big[\eta^\top(\mathrm{diag}(z_j^{(i)}))\Big] - \mathbb{E}_q\Big[\log\Big(\sum_{k=1}^{k=b}\exp(\eta_k)\Big)\Big] \\ &= \sum_{a=1}^{a=b}\nu_a\varsigma_{a,j}^{(i)} - \mathbb{E}_q\Big[\log\Big(\sum_{k=1}^{k=b}\exp(\eta_k)\Big)\Big]\end{aligned} \tag{2.14}$$

The second term is hard to compute, hence, we appeal to the solution suggested by [7] in order to pertain 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:

$$\begin{aligned}-\mathbb{E}_q\Big[\log\Big(\sum_{k=1}^{k=b}\exp(\eta_k)\Big)\Big] &\geq -\log\varrho - \frac{\Big(\sum_{k=1}^{k=b}\mathbb{E}_q[\exp(\eta_k)]\Big)-\varrho}{\varrho} \\ &= 1-\log\varrho - \Big(\sum_{k=1}^{k=b}\mathbb{E}_q[\exp(\eta_k)]\Big)\varrho^{-1}\end{aligned} \tag{2.15}$$

Plugging back the results into Eq. 2.14, we obtain:

$$\mathbb{E}_q[\log p(z_j^{(i)}|\eta)] \approx 1-\log\varrho + \sum_{a=1}^{a=b}\nu_a\varsigma_{a,j}^{(i)} - \Big(\sum_{k=1}^{k=b}\mathbb{E}_q[\exp(\eta_k)]\Big)\varrho^{-1} \tag{2.16}$$

Now, for the entropy $\mathbb{H}(q)$ in Eq. 2.7, we decompose their expectations as:

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

$$\begin{aligned}\mathbb{E}_q[\log q(\Phi_a|\phi_a)] &= \mathbb{E}_q[\log\mathrm{Dir}(\Phi_a|\phi_a)] \\ &= \mathbb{E}_q\Big[\log\Big(\frac{\Gamma(\sum_{j=1}^{j=t}\phi_{a,j})}{\prod_{j=1}^{j=t}\Gamma(\phi_{a,j})}\prod_{j=1}^{j=t}\Phi_{a,j}^{\phi_{a,j}-1}\Big)\Big] \\ &= \log\Gamma\Big(\sum_{j=1}^{j=t}\phi_{a,j}\Big) - \sum_{j=1}^{j=t}\log\Gamma(\phi_{a,j}) + \sum_{j=1}^{j=t}(\phi_{a,j}-1)\mathbb{E}_q[\log\Phi_{a,j}]\end{aligned} \tag{2.17}$$

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

$$\begin{aligned}\mathbb{E}_q[\log q(\eta^{(i)}|\nu,\zeta^2)] =& \mathbb{E}_q\Big[\log\prod_{a=1}^{a=b}\mathcal{N}(\eta_a^{(i)}|\nu_a,\zeta_a^2)\Big] \\ &= -\sum_{a=1}^{a=b}\frac{1}{2}\Big(\log\zeta_a^2 + \log(2\pi)+1\Big)\end{aligned} \tag{2.18}$$

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

$$\begin{aligned}\mathbb{E}_q[\log q(\Lambda^{(i)}|\lambda^{(i)})] &= \mathbb{E}_q[\log\prod_{a=1}^{a=b}\mathrm{Bernoulli}(\Lambda_a^{(i)}|\lambda_a^{(i)})] \\ &= \sum_{a=1}^{a=b}\Big(\lambda_a^{(i)}\log\lambda_a^{(i)} + (1-\lambda_a^{(i)})\log(1-\lambda_a^{(i)})\Big)\end{aligned} \tag{2.19}$$

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

$$\begin{aligned}\mathbb{E}_q[\log q(\Omega^{(i)}|\omega^{(i)})] &= \mathbb{E}_q[\log \mathrm{Dir}(\Omega^{(i)}|\omega^{(i)})] \\ &= \mathbb{E}_q\Big[\log\Big(\frac{\Gamma(\sum_{j=1}^{j=t}\omega_j^{(i)})}{\prod_{j=1}^{j=t}\Gamma(\omega_j^{(i)})}\prod_{j=1}^{j=t}\Omega_j^{(i),\omega_j^{(i)}-1}\Big)\Big] \\ &= \log\Gamma\Big(\sum_{j=1}^{j=t}\omega_j^{(i)}\Big) - \sum_{j=1}^{j=t}\log\Gamma(\omega_j^{(i)}) + \sum_{j=1}^{j=t}(\omega_j^{(i)}-1)\mathbb{E}_q[\log\Omega_j^{(i)}]\end{aligned} \tag{2.20}$$

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

$$\mathbb{E}_q[\log q(z_j^{(i)}|\varsigma_j^{(i)})] = \mathbb{E}_q\Big[\log\prod_{a=1}^{a=b}(\varsigma_{a,j}^{(i)})^{z_{a,j}^{(i)}}\Big] = \sum_{a=1}^{a=b}\varsigma_{a,j}^{(i)}\log\varsigma_{a,j}^{(i)} \tag{2.21}$$

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

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

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

2.3.2 Merging All the Expectations of the ELBO Terms

Now, by joining all the terms in Section 2.3.1, the full ELBO can be defined as:

$$
\begin{aligned}
\mathcal{L}(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) \\
&- \sum_{a=1}^{a=b}\Bigg(\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)\Bigg) \\
&+ \sum_{i=1}^{i=n}\sum_{a=1}^{a=b}\Bigg(\frac{1}{2}\Big(\log\zeta_a^2 + \log(2\pi) + 1\Big)\Bigg) \\
&- \sum_{i=1}^{i=n}\sum_{a=1}^{a=b}\Bigg(\lambda_a^{(i)}\log\lambda_a^{(i)} + (1-\lambda_a^{(i)})\log(1-\lambda_a^{(i)})\Bigg) \\
&- \sum_{i=1}^{i=n}\Bigg(\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)\Bigg) \\
&- \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{2.22}
$$

2.4 Optimizing the ELBO Terms

In this section, we maximize the bound in Eqs 2.6 & 2.7 with respect to each variational parameters using coordinate ascent updates, which optimizes each variational parameter while holding the remaining variables fixed. Practically, a more convenient way is to apply the mini-batch gradient approach, as outlined in [16] that alternates between subsampling a batch of examples and updating each variational parameter, after being scaled by a learning rate. 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:

$$
\begin{aligned}
\mathcal{L}(q)_{[\varsigma]} = &\sum_{i=1}^{i=n}\sum_{j=1}^{j=t_i}\sum_{c=1}^{c=t}\sum_{a=1}^{a=b} y_{j,c}^{(i)}\varsigma_{a,j}^{(i)}\lambda_a^{(i)}\frac{1-\omega_c^{(i)}}{\sum_{k=1}^{k=t}(1-\omega_k^{(i)})}\Big(\Psi(\phi_{a,j}) - \Psi(\sum_{k=1}^{k=t}\phi_{a,k})\Big) \\
&+ \sum_{i=1}^{i=n}\sum_{j=1}^{j=t_i}\sum_{a=1}^{a=b}\nu_a\varsigma_{a,j}^{(i)} - \sum_{i=1}^{i=n}\sum_{j=1}^{j=t_i}\sum_{a=1}^{a=b}\varsigma_{a,j}^{(i)}\log\varsigma_{a,j}^{(i)}
\end{aligned}
\tag{2.23}
$$

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

$$\begin{aligned}\frac{\partial \mathcal{L}(q)_{[\varsigma]}}{\partial \varsigma_{a,j}^{(i)}} =& \sum_{c=1}^{c=t} y_{j,c}^{(i)} \lambda_a^{(i)} \frac{1-\omega_c^{(i)}}{\sum_{k=1}^{k=t}(1-\omega_k^{(i)})} \Big(\Psi(\phi_{a,j}) - \Psi(\sum_{k=1}^{k=t}\phi_{a,k})\Big) \\ &+ \nu_a - \log \varsigma_{a,j}^{(i)} - 1\end{aligned} \tag{2.24}$$

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:

$$\varsigma_{a,j}^{(i)} \propto \exp\left(\sum_{c=1}^{c=t} y_{j,c}^{(i)} \lambda_a^{(i)} \frac{1-\omega_c^{(i)}}{\sum_{k=1}^{k=t}(1-\omega_k^{(i)})} \Big(\Psi(\phi_{a,j}) - \Psi(\sum_{k=1}^{k=t}\phi_{a,k})\Big) + \nu_a - 1\right) \tag{2.25}$$

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,

$$\begin{aligned}\mathcal{L}(q)_{[\nu]} =& \sum_{i=1}^{i=n}\Bigg(-\frac{1}{2}(\nu-\mu)^\top \Sigma^{-1}(\nu-\mu) + \sum_{j=1}^{j=t_i}\sum_{a=1}^{a=b} \nu_a \varsigma_{a,j}^{(i)} \\ &- \sum_{j=1}^{j=t_i}\Big(\sum_{k=1}^{k=b}\exp(\nu_k + \frac{1}{2}\zeta_k^2)\Big)\varrho^{-1}\Bigg)\end{aligned} \tag{2.26}$$

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

$$\frac{\partial \mathcal{L}(q)_{[\nu]}}{\partial \nu_a} = -\Sigma^{-1}(\nu-\mu) + \sum_{j=1}^{j=t_i}\varsigma_{a,j}^{(i)} - \Big(\exp(\nu_a + \frac{1}{2}\zeta_a^2)\Big) t_i \varrho^{-1} \tag{2.27}$$

where ϱ is another variational parameter, as in CTM [7]. 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. 2.22:

$$\begin{aligned}\mathcal{L}(q)_{[\zeta^2]} =& -\frac{1}{2}\sum_{i=1}^{i=n}\mathrm{tr}\Big(\mathrm{diag}(\zeta^2)\Sigma^{-1}\Big) - \sum_{i=1}^{i=n}\sum_{j=1}^{j=t_i}\Bigg(\sum_{k=1}^{k=b}\exp\Big(\nu_k + \frac{1}{2}\zeta_k^2\Big)\Bigg)\varrho^{-1} \\ &+ \frac{1}{2}\sum_{i=1}^{i=n}\sum_{a=1}^{a=b}\log\zeta_a^2\end{aligned} \tag{2.28}$$

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

$$\frac{\partial \mathcal{L}(q)_{[\zeta^2]}}{\partial \zeta_a^2} = -\frac{1}{2}\Bigg(\Sigma_{a,a}^{-1} + t_i\varrho^{-1}\exp\Big(\nu_a + \frac{1}{2}\zeta_a^2\Big) - \frac{1}{\zeta_a^2}\Bigg) \tag{2.29}$$

Again, there is no analytic solution. We use Newton's method for each coordinate, constrained such that $\zeta_a \in \mathbb{R}_{>0}$.

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

$$\mathcal{L}(q)_{[\varrho]} = -\sum_{i=1}^{i=n}\sum_{j=1}^{j=t_i}\log\varrho - \sum_{i=1}^{i=n}\sum_{j=1}^{j=t_i}\Big(\sum_{k=1}^{k=b}\exp(\nu_k + \frac{1}{2}\zeta_k^2)\Big)\varrho^{-1} \tag{2.30}$$

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

$$\frac{\partial \mathcal{L}(q)_{[\varrho]}}{\partial \varrho} = -t_i n \varrho^{-1} + t_i n\Big(\sum_{k=1}^{k=b}\exp(\nu_k + \frac{1}{2}\zeta_k^2)\Big)\varrho^{-2} \tag{2.31}$$

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

$$\varrho = \sum_{k=1}^{k=b}\exp(\nu_k + \frac{1}{2}\zeta_k^2) \tag{2.32}$$

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

$$\begin{aligned}\mathcal{L}(q)_{[\omega]} =& \sum_{i=1}^{i=n}\sum_{j=1}^{j=t}\Psi(\omega_j^{(i)})(\xi_j + \mathbf{M}_j^{(i)} - \omega_j^{(i)}) - \sum_{i=1}^{i=n}\sum_{j=1}^{j=t}\Psi(\sum_{k=1}^{k=t}\omega_k^{(i)})(\xi_j + \mathbf{M}_j^{(i)} - \omega_j^{(i)}) \\ &+ \sum_{i=1}^{i=n}\sum_{j=1}^{j=t_i}\sum_{c=1}^{c=t}\sum_{a=1}^{a=b}\left(y_{j,c}^{(i)}\varsigma_{a,j}^{(i)}\lambda_a^{(i)}\frac{1-\omega_c^{(i)}}{\sum_{k=1}^{k=t}(1-\omega_k^{(i)})}\left(\Psi(\phi_{a,j}) - \Psi(\sum_{k=1}^{k=t}\phi_{a,k})\right)\right) \\ &+ \sum_{i=1}^{i=n}\sum_{j=1}^{j=t}\log\Gamma(\omega_j^{(i)}) - \sum_{i=1}^{i=n}\log\Gamma\Big(\sum_{j=1}^{j=t}\omega_j^{(i)}\Big)\end{aligned} \tag{2.33}$$

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

$$\begin{aligned}\frac{\partial\mathcal{L}(q)_{[\omega]}}{\partial\omega_c^{(i)}} =& \Big(\Psi'(\omega_c^{(i)}) - \Psi'(\sum_{k=1}^{k=t}\omega_k^{(i)})\Big)(\xi_c + \mathbf{M}_c^{(i)} - \omega_c^{(i)}) \\ &- \left(\frac{1-\omega_c^{(i)} - \sum_{k=1}^{k=t}(1-\omega_k^{(i)})}{(\sum_{k=1}^{k=t}(1-\omega_k^{(i)}))^2}\right)\sum_{j=1}^{j=t_i}\sum_{a=1}^{a=b}y_{j,c}^{(i)}\varsigma_{a,j}^{(i)}\lambda_a^{(i)}\Big(\Psi(\phi_{a,j}) - \Psi(\sum_{k=1}^{k=t}\phi_{a,k})\Big)\end{aligned} \tag{2.34}$$

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

$$\begin{aligned}\omega_c^{(i)} \propto& \xi_c + \mathbf{M}_c^{(i)} - \left(\frac{1-\omega_c^{(i)} - \sum_{k=1}^{k=t}(1-\omega_k^{(i)})}{(\sum_{k=1}^{k=t}(1-\omega_k^{(i)}))^2}\right)\sum_{j=1}^{j=t_i}\sum_{a=1}^{a=b}y_{j,c}^{(i)}\varsigma_{a,j}^{(i)}\lambda_a^{(i)} \\ &\times\Big(\Psi(\phi_{a,j}) - \Psi(\sum_{k=1}^{k=t}\phi_{a,k})\Big)\end{aligned} \tag{2.35}$$

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

$$\begin{aligned}\mathcal{L}(q)_{[\lambda]} =& \sum_{i=1}^{i=n}\sum_{a=1}^{a=b}\lambda_a^{(i)}(\log(\beta_a^{(i)}) - \log\lambda_a^{(i)}) \\ &+ \sum_{i=1}^{i=n}\sum_{a=1}^{a=b}(1-\lambda_a^{(i)})\Big(\log(1-\beta_a^{(i)}) - \log(1-\lambda_a^{(i)})\Big)\end{aligned} \tag{2.36}$$

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

$$\frac{\partial\mathcal{L}(q)_{[\lambda]}}{\partial\lambda_a^{(i)}} = \log(1-\lambda_a^{(i)}) - \log\lambda_a^{(i)} + \log(\beta_a^{(i)}) - \log(1-\beta_a^{(i)}) \tag{2.37}$$

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

$$\theta = \log\left(\frac{\lambda_a^{(i)}}{1-\lambda_a^{(i)}}\right) = \log(\beta_a^{(i)}) - \log(1-\beta_a^{(i)}) \tag{2.38}$$

Therefore, we get the following updates:

$$\lambda_a^{(i)} = \frac{1}{1+\exp^{-\theta}} \tag{2.39}$$

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. 2.4:

$$\begin{aligned}\mathcal{L}(q)_{[\phi]} =& \sum_{a=1}^{a=b}\sum_{c=1}^{c=t}\Psi(\phi_{a,c})\left(\alpha_c - \phi_{a,c} + \sum_{i=1}^{i=n}\sum_{j=1}^{j=t_i}y_{j,c}^{(i)}\varsigma_{a,j}^{(i)}\lambda_a^{(i)}\frac{1-\omega_c^{(i)}}{\sum_{k=1}^{k=t}(1-\omega_k^{(i)})}\right) \\ &- \sum_{a=1}^{a=b}\sum_{j=1}^{c=t}\Psi(\sum_{k=1}^{k=t}\phi_{a,k})\left(\alpha_c - \phi_{a,c} + \sum_{i=1}^{i=n}\sum_{j=1}^{j=t_i}y_{j,c}^{(i)}\varsigma_{a,j}^{(i)}\lambda_a^{(i)}\frac{1-\omega_c^{(i)}}{\sum_{k=1}^{k=t}(1-\omega_k^{(i)})}\right) \\ &- \sum_{a=1}^{a=b}\log\Gamma\Big(\sum_{j=1}^{j=t}\phi_{a,j}\Big) + \sum_{a=1}^{a=b}\sum_{j=1}^{j=t}\log\Gamma(\phi_{a,j})\end{aligned} \tag{2.40}$$

```
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;
    Sample a minibatch randomly B ⊂ P;
    for i ∈ B do
        repeat
            Update ς^(i) with Eq. 2.25;
            Update ν^(i) with Eq. 2.27 using conjugate gradient algorithm;
            Update ζ^(2,(i)) with Eq. 2.29 using Newton's method;
            Update ϱ^(i) with Eq. 2.32;
            Update ω^(i) with Eq. 2.35;
            Update λ^(i) with Eq. 2.39;
        until local variational parameters converge;
        Compute optimal values μ = ν/|B|, Σ = diag(ζ²/|B|) + μμ^⊤;
        Compute global optimal values ϕ with Eq. 2.42;
        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

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

$$\begin{aligned}\frac{\partial \mathcal{L}(q)_{[\phi]}}{\partial \phi_{a,c}} =& \Psi'(\phi_{a,c})\Bigg(\alpha_c - \phi_{a,c} + \sum_{i=1}^{i=n}\sum_{j=1}^{j=t_i} y_{j,c}^{(i)}\varsigma_{a,j}^{(i)}\lambda_a^{(i)}\frac{1-\omega_c^{(i)}}{\sum_{k=1}^{k=t}(1-\omega_k^{(i)})}\Bigg) \\ &- \Psi'(\sum_{k=1}^{k=t}\phi_{a,k})\Bigg(\alpha_c - \phi_{a,c} + \sum_{i=1}^{i=n}\sum_{j=1}^{j=t_i} y_{j,c}^{(i)}\varsigma_{a,j}^{(i)}\lambda_a^{(i)}\frac{1-\omega_c^{(i)}}{\sum_{k=1}^{k=t}(1-\omega_k^{(i)})}\Bigg)\end{aligned} \tag{2.41}$$

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

$$\phi_{a,c} = \alpha_c + \sum_{i=1}^{i=n}\sum_{j=1}^{j=t_i} y_{j,c}^{(i)}\varsigma_{a,j}^{(i)}\lambda_a^{(i)}\frac{1-\omega_c^{(i)}}{\sum_{k=1}^{k=t}(1-\omega_k^{(i)})} \tag{2.42}$$

The variational inference algorithm samples a mini-batch from a collection, and use it to compute the local latent parameters in Eqs 2.25, 2.27, 2.29, 2.32, 2.35, and 2.39 until the evidence lower bound (in Eq. 2.4) converges. Then, the global variational parameter ϕ is updated using the posteriors (β, Λ, η, z, Ω) collected from the previous step in Eq. 2.42, 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. We summarize our variational inference in Algorithm 7.

2.5 Posterior Predictive Distribution for SPREAT

We apply posterior predictive distribution to evaluate model's fitness, which estimates the distribution of unobserved values given the observed values and parameters trained on a held-out training set [16]. This metric is useful in evaluating models as it avoids comparing bounds of those models. The predictive distribution is formulated as follows:

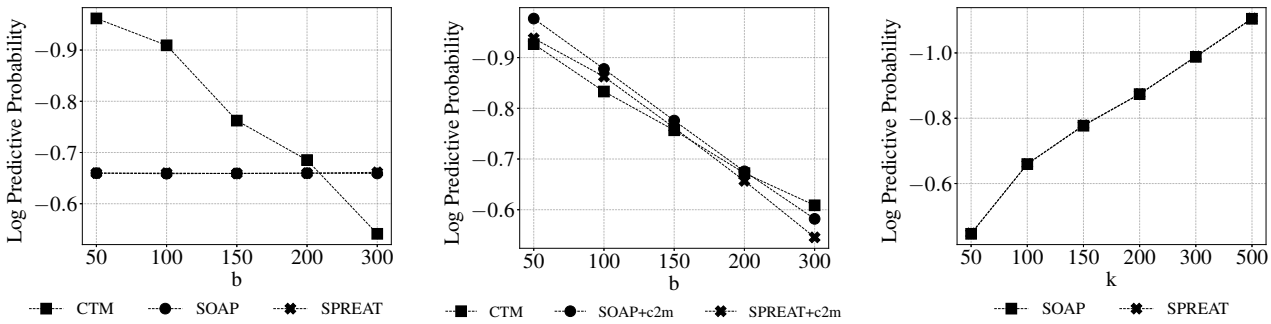
$$
\begin{aligned}
p(\tilde{\mathbf{Y}}|\mathbf{Y}_{obs},\mathbf{M}) = \int & p(\tilde{\mathbf{Y}}|\Theta,\mathbf{M})p(\Theta|\mathbf{Y}_{obs},\mathbf{M})d\Theta \\
& + \sum_{i=1}^{i=n}\left(\frac{1}{2}\log|\Sigma^{-1}| - \frac{b}{2}\log 2\pi - \frac{1}{2}\Big(\mathrm{tr}(\mathrm{diag}(\zeta^2)\Sigma^{-1}) + (\nu-\mu)^\top\Sigma^{-1}(\nu-\mu)\Big)\right) \\
& + \sum_{i=1}^{i=n}\sum_{a=1}^{a=b}\left(\lambda_a^{(i)}\log(\beta_a^{(i)}) + (1-\lambda_a^{(i)})\log(1-\beta_a^{(i)})\right) \\
& + \sum_{i=1}^{i=n}\sum_{a=1}^{a=b}\left((\gamma-1)\log(\beta_a^{(i)}) + (\kappa-1)\log(1-\beta_a^{(i)}) - \log(B(\gamma,\kappa)\right) \\
& + \sum_{i=1}^{i=n}\left(\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)\right) \\
& + \sum_{i=1}^{i=n}\sum_{j=1}^{j=t_i}\left(\log\varpi + \sum_{c=1}^{c=t}\sum_{a=1}^{a=b}\left(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)\right)\right) \\
& + \sum_{i=1}^{i=n}\sum_{j=1}^{j=t_i}\left(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}\right) \\
& - \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{2.43}
$$

3 Experimental Setup

In this section, we describe the experimental settings and outline the materials used to evaluate the performance of reMap. The reMap and CBT were written in Python v3 and depends on third party libraries (e.g. Numpy [31]). Unless otherwise specified all tests were conducted on a Linux server using 10 cores of Intel Xeon CPU E5-2650.

3.1 Description of Datasets

We evaluated reMap's performance using a subset of datasets used in triUMPF [4]: i)- 6 T1 golden data that are composed of six databases, retrieved from biocyc website: *EcoCyc (v21)*, *HumanCyc (v19.5)*, *AraCyc (v18.5)*, *YeastCyc (v19.5)*, *LeishCyc (v19.5)*, and *TrypanoCyc (v18.5)*, and are refined to include only those pathways that cross-intersect with the *MetaCyc* database (v21) [10]; ii)- Critical Assessment of Metagenome Interpretation (CAMI) initiative low complexity dataset consisting of 40 genomes [29] that can be obtained from edwards.sdsu.edu/research/cami-challenge-datasets/; and iii)- BioCyc (v20.5 T2 & 3) [11], which consists of 9255 PGDBs (Pathway/Genome Databases) with 1463 distinct pathway labels and is constructed using the Pathway Tools v21 [20]. We also employed Synset-2, a noisy corrupted training set, introduced in [5].



(a) Effect of b

(b) Effect of b by collapsing

(c) Effect of k with $b = 200$

Figure 5: **Log predictive distribution on CAMI data.**

The detailed characteristics of the datasets are summarized in Table 2. For each dataset $\mathcal{S}$, we use $|\mathcal{S}|$ and L($\mathcal{S}$) to represent the number of instances and pathway labels, respectively. In addition, we also present some characteristics of the multi-label datasets, which are denoted as:

1. Label cardinality (LCard($\mathcal{S}$) $= \frac{1}{n}\sum_{i=1}^{i=n}\sum_{j=1}^{j=t}\mathbb{I}[\mathbf{Y}_{i,j} \neq -1]$), where $\mathbb{I}$ is an indicator function. It denotes the average number of pathways in $\mathcal{S}$.

2. Label density (LDen($\mathcal{S}$) $= \frac{LCard(\mathcal{S})}{L(\mathcal{S})}$). This is simply obtained through normalizing LCard($\mathcal{S}$) by the number of total pathways in $\mathcal{S}$.

3. Distinct label sets (DL($\mathcal{S}$)). This notation indicates the number of distinct pathways in $\mathcal{S}$.

4. Proportion of distinct label sets (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($\mathcal{S}$).

3.2 Parameter Settings

reMap depends on both correlated bag pathway models (Section 2) and pathway2vec [2], therefore, we applied the following default configurations for each component:

- The parameters for reMap are configured as: the learning rate $\eta = 0.0001$, the batch size to 30, the number of epochs to $\tau = 10$, the bag centroid hyperparameter $\alpha = 16$, the cutoff threshold for cosine similarity $v = 0.2$, the cutoff decision threshold for bags $\beta = 0.3$, the number of bags $b = 200$, and the subsampled bag size $\gamma = 50$. For the regularized hyperparameters $\lambda_{1:5}$ and κ, we performed 10-fold cross-validation on a subsampled of (v20.5 T2 &3) collection and found the settings $\lambda_{1:5} = 0.01$ and $\kappa = 0.01$ to be optimum on golden T1 datasets.

- The parameters for three correlated models are configured as: the pathway distribution over concepts Φ were initialized using gamma distribution (with shape and scale parameters were fixed to 100 and 1/100, respectively), the forgetting rate to $g = 0.9$, the delay rate to $l = 1$, the batch size to 100, the number of epochs to 3, the number of concepts $b = 200$, top k pathways to 100 (only for SOAP and SPREAT), the Dirichlet hyperparameters α and ξ to 0.0001, and the beta hyperparameters γ and κ to 2 and 3, respectively. The supplementary pathways $\mathbf{M}$ for BioCyc, CAMI, and golden T1 datasets were obtained using mlLGPR [5]. We train mlLGPR (elastic-net) using enzymatic reaction and pathway evidence features where hyperparameters were fixed to their default values. A schematic view of pathway frequency across datasets for BioCyc T2 & 3 and CAMI, along with their augmented pathways is depicted in Fig. 6.

- The parameters for pathway2vec 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 was 100, the walk length is 100, the embedding dimension size was $m = 128$, the neighborhood size was 5, the size of negative samples was 5, and the used configuration of MetaCyc was "uec", indicating links among ECs are being trimmed.

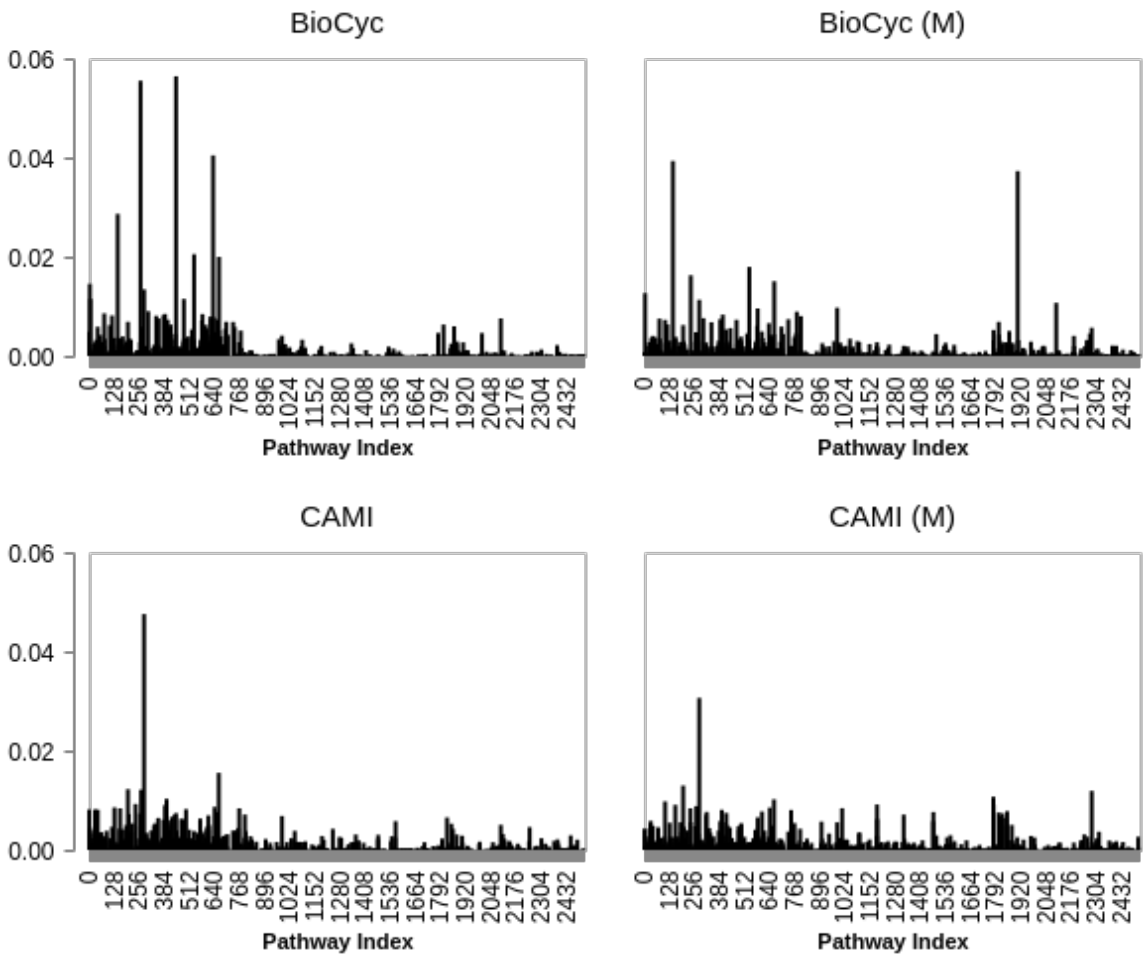


Figure 6: **Illustration of pathway frequency (averaged on all examples) in BioCyc (v20.5 T2 &3) and CAMI data, and their background pathways, indicated by M.**

Both reMap and correlated models were trained using BioCyc (v20.5 T2 &3) collection. After obtaining bags $\mathcal{S}_{bag}$, we train leADS [3] using built-in "factorization" option that enables training bags and pathways, simultaneously, for 10 epochs while the prediction strategy was set to "pref-vrank" with the ranking hyperparameter was fixed to 300. For all the remaining hyperparameters in pathway2vec, CBT, leADS, and mlLGPR [5], they are fixed to their default values.

4 Experimental Results and Discussion

We conducted four experimental studies: parameter sensitivity and visualization for correlated models, assessing prediction probability history during relabeling process of reMap, and metabolic pathway prediction effectiveness after obtaining $\mathcal{S}_{bag}$ using leADS.

4.1 Sensitivity Analysis of Correlated Models

Experimental setup. A fundamental challenge for the reMap pipeline is to acquire a good distribution of bags 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 **M** while varying the number of bags 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 bag distribution by varying the cutoff threshold value according to $k \in \{50, 100, 150, 200, 300, 500\}$. For the comparative analysis, we used CAMI as a test data to report the log predictive distribution (Section 2.5), where a lower score entails higher generalization capability for the associated models.

Experimental results. While the log predictive scores for SOAP and SPREAT in Fig. 5a appears to be horizontal across bag size, the CTM model projects a more realistic view where it's performances are seen to

Dataset	$\|\mathcal{S}\|$	L($\mathcal{S}$)	LCard($\mathcal{S}$)	LDen($\mathcal{S}$)	DL($\mathcal{S}$)	PDL($\mathcal{S}$)	R($\mathcal{S}$)	RCard($\mathcal{S}$)	RDen($\mathcal{S}$)	DR($\mathcal{S}$)	PDR($\mathcal{S}$)	PLR($\mathcal{S}$)	Domain
AraCyc	1	510	510	1	510	510	2182	2182	1	1034	1034	0.2337	Arabidopsis thaliana
EcoCyc	1	307	307	1	307	307	1134	1134	1	719	719	0.2707	Escherichia coli K-12 substr.MG1655
HumanCyc	1	279	279	1	279	279	1177	1177	1	693	693	0.2370	Homo sapiens
LeishCyc	1	87	87	1	87	87	363	363	1	292	292	0.2397	Leishmania major Friedlin
TrypanoCyc	1	175	175	1	175	175	743	743	1	512	512	0.2355	Trypanosoma brucei
YeastCyc	1	229	229	1	229	229	966	966	1	544	544	0.2371	Saccharomyces cerevisiae
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
BioCyc	9255	1804003	194.9220	0.0001	1463	0.1581	8848714	956.1009	0.0001	2705	0.2923	0.2039	BioCyc version 20.5 (tier 2 & 3)
Synset-2	15000	6806262	453.7508	0.00007	2526	0.1684	34006386	2267.0924	0.00007	3650	0.2433	0.2001	Synthetically generated (corrupted)

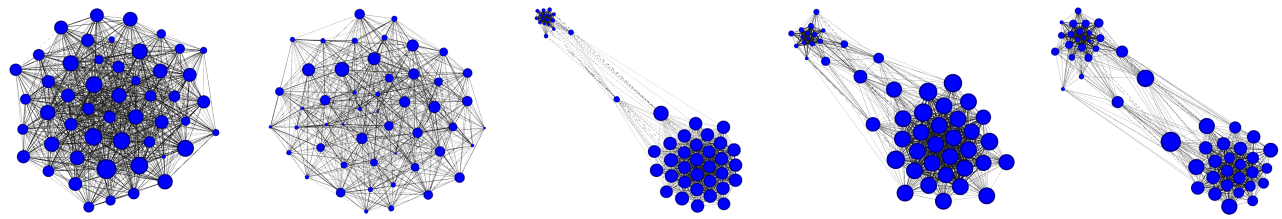
Table 2: **Characteristics of the experimental datasets.** The notations $|\mathcal{S}|$, L($\mathcal{S}$), LCard($\mathcal{S}$), LDen($\mathcal{S}$), DL($\mathcal{S}$), and PDL($\mathcal{S}$) represent number of instances, number of pathway labels, pathway labels cardinality, pathway labels density, distinct pathway labels set, and proportion of distinct pathway labels set for $\mathcal{S}$, respectively. The notations R($\mathcal{S}$), RCard($\mathcal{S}$), RDen($\mathcal{S}$), DR($\mathcal{S}$), and PDR($\mathcal{S}$) have similar meanings as before but for the enzymatic reactions $\mathcal{E}$ in $\mathcal{S}$. PLR($\mathcal{S}$) represents a ratio of L($\mathcal{S}$) to R($\mathcal{S}$). The last column denotes the domain of $\mathcal{S}$.

be gaining by including more bags. For the former models, this phenomena is not a consequence of design flaw, instead, it 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 in BioCyc T2 & 3 is ~ 195 while only retaining 100 pathways for each bag. By excluding $\mathbf{M}$ (through enabling c2m option), we observe that the log predictive distribution of SOAP and SPREAT are similar with that of CTM, as shown in Fig. 5b, which supports our previous discussion. We found that $b = 200$ gives a good set of overlapping pathways while having on average ~ 15 distinct pathways for each bag from 2526 pathways. By fixing $b = 200$, we search for an optimum k value. As illustrated in Fig. 5c, 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 bag and pathway distributions.

4.2 Bag Visualization

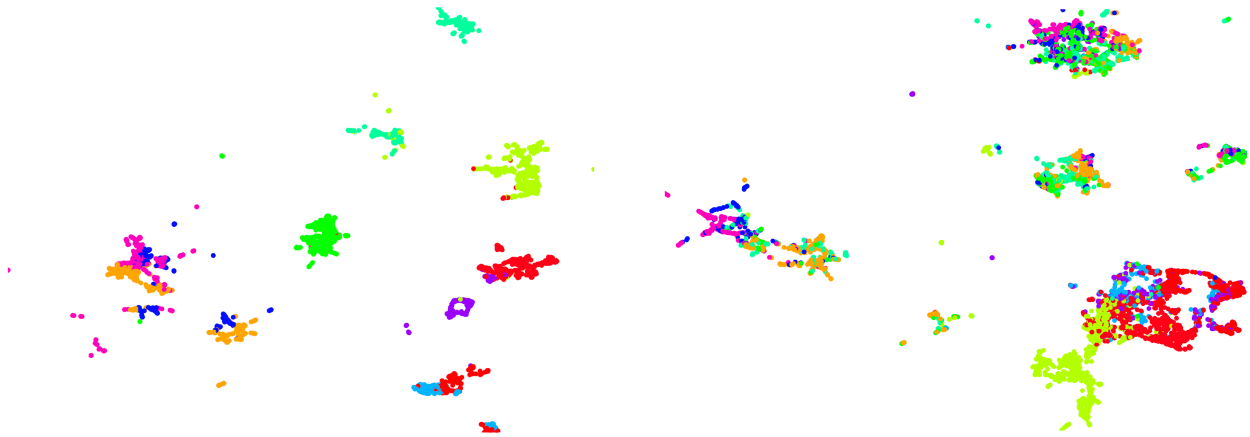
Experimental setup. Here, we visualized the discovered bags using models in Section 4.1 with the goal to assess the quality of bags. First, we examined the influence of augmented pathways on bag correlation patterns, i.e., Σ, in SOAP and SPREAT and contrast the outputs with CTM, SOAP+c2m, and SPREAT+c2m. Then, we performed an in-depth comparison between the sparse models (SOAP+c2m and SPREAT+c2m) with CTM to ensure that our modeling assumptions are aligned with the observed data, where a bag containing more focused and fewer pathways is preferred. For all experiments here, we applied the settings described in Section 3.2.

Experimental results. From Fig. 7, we notice two core findings. First, in contrast to CTM, SOAP+c2m, SPREAT, and SPREAT+c2m, bags in SOAP are densely connected (~ 74 bags) where the width of edges indicates the strength of correlations. Second, leveraging background pathways in SOAP and SPREAT resulted in gathering more pathways for each bag. These observations demonstrate the influence of $\mathbf{M}$ (obtained from mlLGPR) to pathway distribution over bags. As an example, Fig. 8 shows samples from BioCyc T2 & 3 pathways and corresponding background pathways after projecting them onto 2D space using UMAP [25]. The colors encode samples from BioCyc T2 & 3 pathways that are clustered using the K-means algorithm [19] with 10 groups. While examples of the same color in BioCyc T2 & 3 pathways form a clear distinct group, the same examples are seen to be intermixed for $\mathbf{M}$, possibly comprising of many false-positive pathways as depicted in Fig. 6 where many pathways (represented as column bars) are differentially distributed between BioCyc T2 & 3 pathways and $\mathbf{M}$. For the collapsed models, they are observed to share similar behaviors as CTM (Fig. 5b). However, their bag distribution consists of fewer pathways than CTM (Fig. 7). For example, Fig. 9 shows 50



(a) SOAP (#bags: ∼ 74; #pathways: ∼ 138) (b) SPREAT (#bags: ∼ 55; #pathways: ∼ 141) (c) CTM (#bags: ∼ 50; #pathways: ∼ 69) (d) SOAP+c2m (#bags: ∼ 52; #pathways: ∼ 8) (e) SPREAT+c2m (#bags: ∼ 51; #pathways: ∼ 8)

Figure 7: **Visualizing** 50 **randomly picked bags for each model, trained with** $b = 200$**.** The first term within the bracket, i.e., #bags, corresponds to the average number of correlated bags while the second term, i.e., #pathways, represents the average number of pathway size per bag. The circles represent bags, and their sizes reflect the correlation strength with other bags. Two clusters of bags can be seen for the last three models indicating the two clusters contain distinct pathways.



(a) BioCyc (v20.5 T2 & 3) (b) BioCyc (v20.5 T2 & 3) (M)

Figure 8: **2D UMAP projections of BioCyc (v20.5 T2 & 3) pathways and the corresponding background pathways.** Fig. 8a serves as a basis for color-coding where examples of one color in BioCyc are clustered together while the same examples are seen to be spread across the augmented BioCyc pathways (**M**) in Fig. 8b. Better viewed in color.

randomly selected bags with associated 100 pathways, where CTM is shown to encapsulate more pathways per bag (encoded by gradient darker colors) while SOAP+c2m and SPREAT+c2m exhibit a sparse distribution.

The results from these experiments link with our previous remarks, entailing that SOAP and SPREAT are equipped to reduce irrelevant pathways by applying dual sparseness. In particular, SPREAT is observed to generate fewer correlations than SOAP. With regard to incorporating supplementary pathways, SOAP and SPREAT are both sensitive to false-positive pathways, therefore, including accurate augmented pathways may recover better pathway distribution over bags.

4.3 Assessing the History Probability

Experimental setup. After analyzing the correlated models in Sections 4.1 and 4.2, here we investigated the accumulated history probability **H** on relabeling golden T1 datasets, during the feed-forward phase, using the settings in Section 3.2. Recall that reMap employs four constraints in the feed-backward phase to select bags with high probability to annotate data. The hope is to obtain an optimum $h^{\mathbf{bag}}$ to remapping pathways to bags. For the demonstration purposes, we only focused on SOAP to collect bags although any models can be used to conduct this experiment.

Experimental results. Fig. 10 shows **H** during the annotation process of the six datasets. In the beginning, reMap attempts to select a set of bags from $\widehat{\mathbf{D}}$, corresponding to the maximum number of bags that may exist for each example. However, with progressive updates and calibration of parameters, reMap rectifies bags assignments where it picks fewer bags, containing more informative contents associated with each sample. For instance, at $q = 1$, the HumanCyc is tagged with multiple bags, represented by darker colors where higher

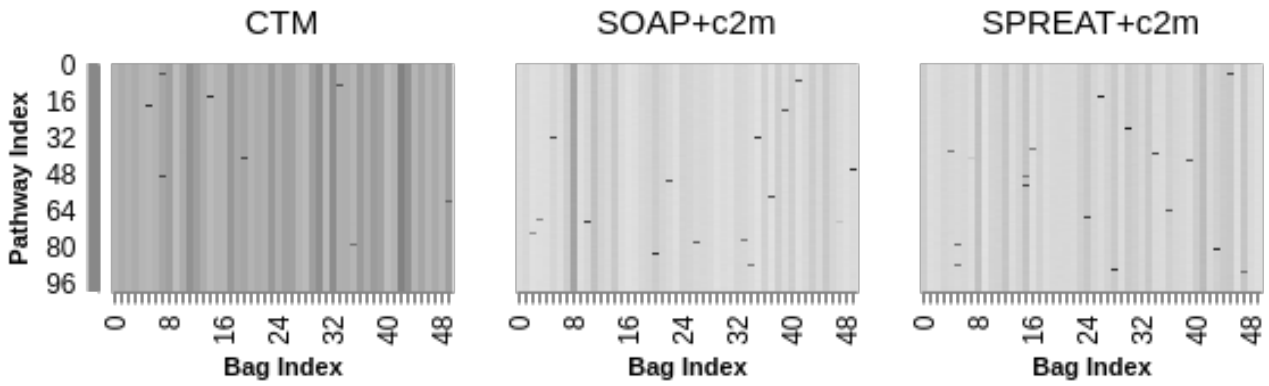


Figure 9: **Heatmap representing bag distribution of CTM, SOAP+c2m, SPREAT+c2m for randomly picked** 50 **bags with their associated** 100 **pathways.** The entries is color-coded on a gradient scale ranging from light-gray to dark-gray, where higher intensity entails higher probability.

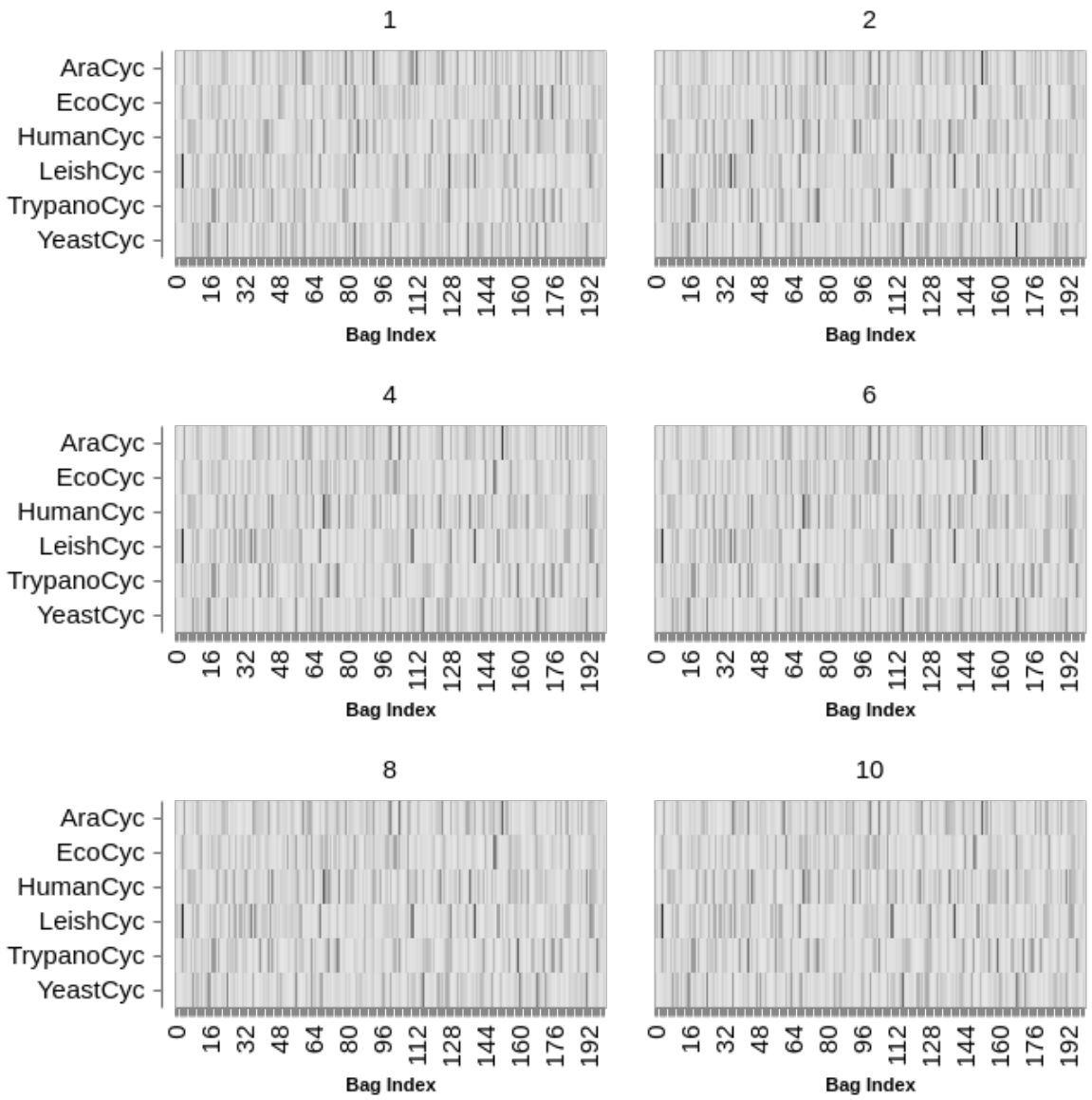


Figure 10: **Snapshot of the history probability H during the relabeling process of golden T1 data for** 10 **successive rounds.** The x-axis shows 200 bags while the y-axis corresponds data. Darker colors indicate high probability to assigning bags to the corresponding data.

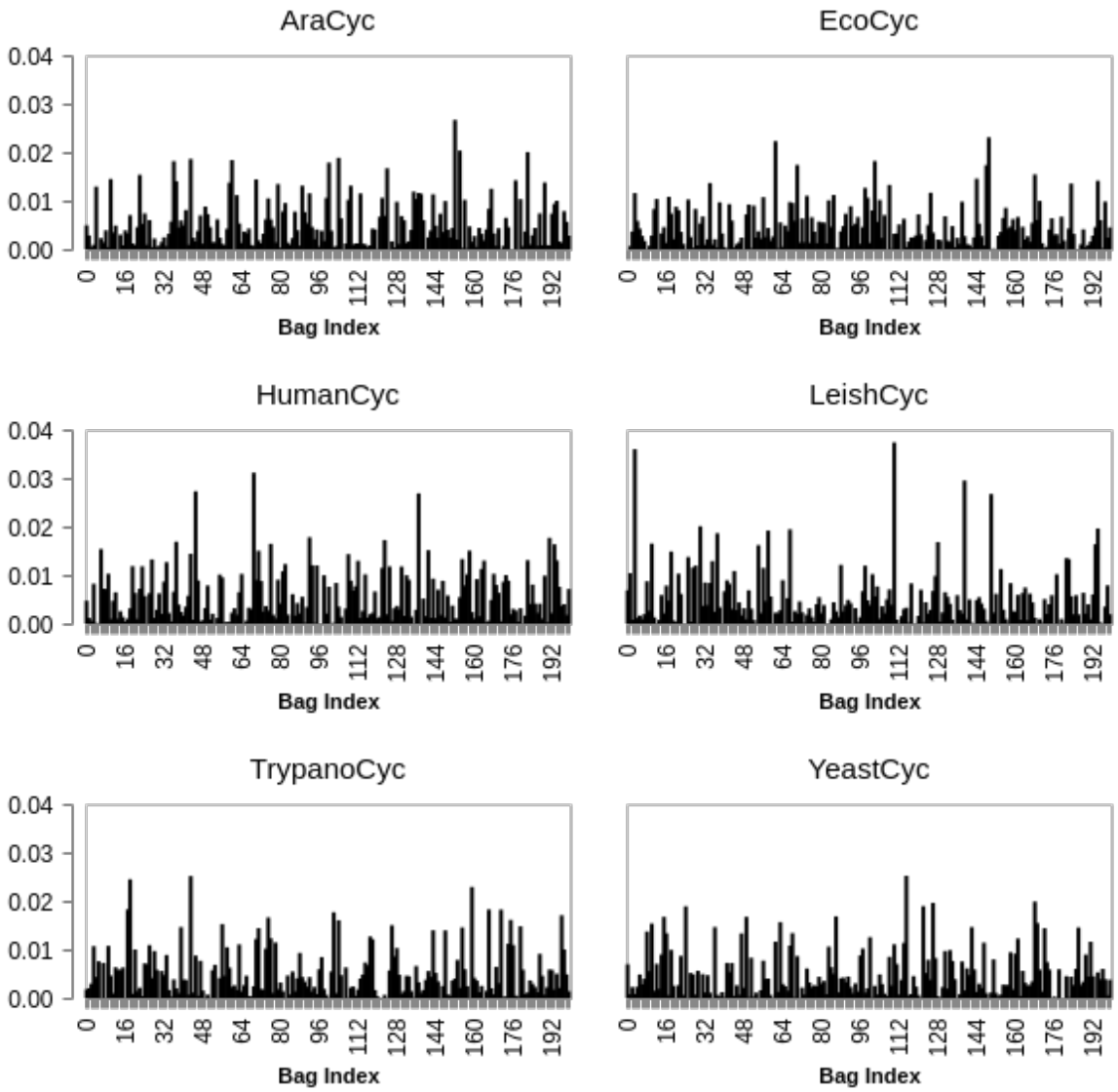
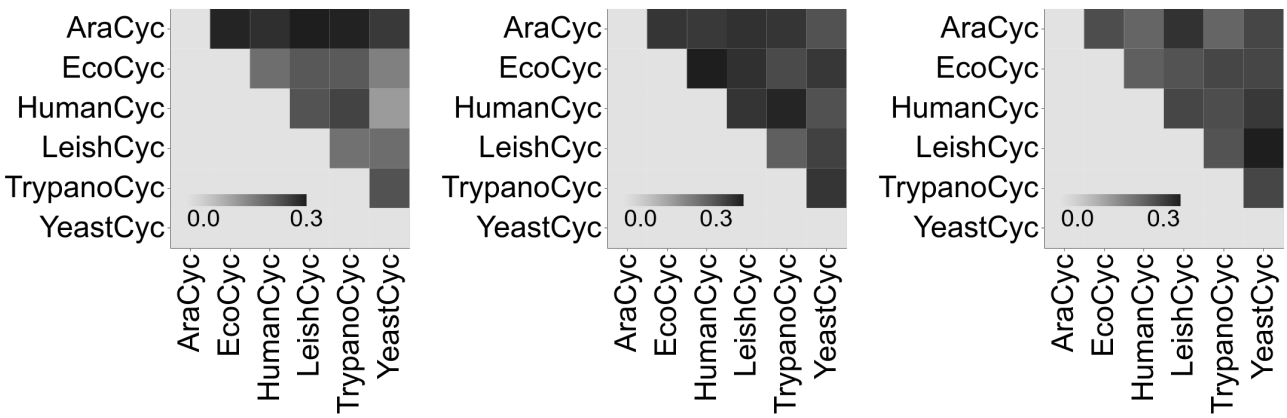


Figure 11: **The probability history H during annotation of T1 golden data after** 10 **successive rounds.** The x-axis shows 200 bags while the y-axis corresponds the associated probability. Darker colors indicate high probability to assigning bags to the corresponding data.



(a) EC (b) Pathway (c) Bag

Figure 12: **Similarity among golden datasets, measured by cosine distance.**

#	MataCyc Pathway ID	MataCyc Pathway Name
1	PROSYN-PWY	L-proline biosynthesis I
2	PWY-5137	fatty acid β;-oxidation III (unsaturated, odd number)
3	PWY-3982	uracil degradation I (reductive)
4	TRNA-CHARGING-PWY	tRNA charging
5	MANNOSYL-CHITO-DOLICHOL-BIOSYNTHESIS	protein N-glycosylation initial phase (eukaryotic)
6	PWY-5667	CDP-diacylglycerol biosynthesis I
7	PWY0-662	PRPP biosynthesis I
8	PWY-46	putrescine biosynthesis III
9	PWY-6261	thyroid hormone metabolism II (via conjugation and/or degradation)
10	PWY-6061	bile acid biosynthesis, neutral pathway
11	PWY66-388	fatty acid α;-oxidation III

Table 3: **The** 11 **HumanCyc pathways corresponding the bag index** 16**.**

intensity indicates a high probability of assigning the corresponding bag to HumanCyc, but at $q = 10$ less than 35 bags were retained out of 200 possible bags. Table 3 shows the 11 HumanCyc pathways corresponding the assigned bag index 16, which includes *L-proline biosynthesis I*, *thyroid hormone metabolism II (via conjugation and/or degradation)*, and *bile acid biosynthesis, neutral pathway*. The pathway set in bag 16 for HumanCyc is an indication that reMap is robust to a certain degree and able to capture relevant pathways for a bag.

Bags preferences by reMap can be further examined by computing similarity (measured in cosine distance metric) among datasets according to EC, pathway, and bag spaces, depicted in Fig. 12. As elucidated in Section 1.2, reMap exploits four constraints to strike balance among these spaces. This is particularly sighted for AraCyc and EcoCyc, where ECs (Fig. 12a) have similar density with the remaining golden data, however, the pathways are differentially represented (Fig. 12b). Hence, this information is down propagated to the bag annotation process depicted in Fig. 12c. Overall, reMap exhibits a steady, albeit slow, annotation progression while preserving consistency across different spaces.

4.4 Metabolic Pathway Prediction

Experimental setup. In this section, we quantified the quality of annotation by leveraging the obtained bags from CBT modules for the pathway prediction task. We train leADS using the configuration discussed in Section 3.2. The results are reported on golden T1 data using four evaluation metrics: *Hamming loss*, *average precision*, *average recall*, and *average F1 score*. For comparative analysis, we use the following pathway prediction algorithms: i)- MinPath v1.2 [34], which infers a conserved set of biological pathways from a list of enzymatic reactions; ii)- PathoLogic v21 [20], which is a rule based approach that incorporates a set of manually specified rules to predicting pathways; iii)- mlLGPR (elastic net with enzymatic reaction and pathway evidence features) [5], which adopts multi-label classification approach to reconstructing pathways from enzymatic reactions; and iv)- triUMPF [4] that incorporates three association matrices, pathway to EC, pathway to pathway, and EC to EC, to precisely reconstruct pathways by applying triple non-negative matrix factorization technique.
Experimental results. From Table 4, we observe that reMap+SOAP outperforms triUMPF on EcoCyc, HumanCyc, AraCyc, YeastCyc, and TrypanoCyc 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. In summary, this experiment demonstrates that bag based approach, in particular reMap+SOAP, improves pathway prediction performance.

References

[1] Edoardo M Airoldi, David M Blei, Stephen E Fienberg, and Eric P Xing. Mixed membership stochastic blockmodels. *Journal of machine learning research*, 9(Sep):1981–2014, 2008.

[2] Abdur Rahman MA Basher and Steven J Hallam. Leveraging heterogeneous network embedding for metabolic pathway prediction. *bioRxiv*, feb 2020.

[3] Abdur Rahman MA Basher and Steven J Hallam. `https://github.com/hallamlab/leADS`, 2020.

[4] Abdur Rahman MA Basher, Ryan J McLaughlin, and Steven J Hallam. Incorporating triple nmf with community detection to metabolic pathway inference. *bioRxiv*, 2020.

[5] Abdur Rahman MA Basher, Ryan J McLaughlin, and Steven J Hallam. Metabolic pathway inference using multi-label classification with rich pathway features. *bioRxiv*, February 2020.

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
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

Table 4: **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.

[6] Jacob Bien and Robert J Tibshirani. Sparse estimation of a covariance matrix. *Biometrika*, 98(4):807–820, 2011.

[7] David Blei and John Lafferty. Correlated topic models. *Advances in neural information processing systems*, 18:147, 2006.

[8] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *arXiv preprint arXiv:1601.00670*, 2016.

[9] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

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

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

[12] Ron Caspi, Richard Billington, Ingrid M Keseler, Anamika Kothari, Markus Krummenacker, Peter E Midford, Wai Kit Ong, Suzanne Paley, Pallavi Subhraveti, and Peter D Karp. The metacyc database of metabolic pathways and enzymes-a 2019 update. *Nucleic acids research*, 2019.

[13] Haw-Shiuan Chang, Erik Learned-Miller, and Andrew McCallum. Active bias: Training more accurate neural networks by emphasizing high variance samples. In *Advances in Neural Information Processing Systems*, pp. 1002–1012, 2017.

[14] Alex Graves. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, 2013.

[15] Junxian He, Zhiting Hu, Taylor Berg-Kirkpatrick, Ying Huang, and Eric P Xing. 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.

[16] Matthew D Hoffman, David M Blei, Chong Wang, and John William Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

[17] Liangjie Hong and Brian D Davison. Empirical study of topic modeling in twitter. In *Proceedings of the first workshop on social media analytics*, pp. 80–88. acm, 2010.

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

[19] Tapas Kanungo, David M Mount, Nathan S Netanyahu, Christine D Piatko, Ruth Silverman, and Angela Y Wu. An efficient k-means clustering algorithm: Analysis and implementation. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (7):881–892, 2002.

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

[21] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

[22] Tianyi Lin, Wentao Tian, Qiaozhu Mei, and Hong Cheng. 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.

[23] Chunjie Luo, Jianfeng Zhan, Lei Wang, and Qiang Yang. Cosine normalization: Using cosine similarity instead of dot product in neural networks. *arXiv preprint arXiv:1702.05870*, 2017.

[24] Jon D Mcauliffe and David M Blei. Supervised topic models. In *Advances in neural information processing systems*, pp. 121–128, 2008.

[25] Leland McInnes, John Healy, Nathaniel Saul, and Lukas Grossberger. Umap: Uniform manifold approximation and projection. *The Journal of Open Source Software*, 3(29):861, 2018.

[26] David Mimno, Matt Hoffman, and David Blei. Sparse stochastic inference for latent dirichlet allocation. *arXiv preprint arXiv:1206.6425*, 2012.

[27] Andrew Y Ng, Michael I Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in neural information processing systems*, pp. 849–856, 2002.

[28] John Paisley, Chong Wang, David M Blei, and Michael I Jordan. Nested hierarchical dirichlet processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(2):256–270, 2015.

[29] Alexander Sczyrba, Peter Hofmann, Peter Belmann, David Koslicki, Stefan Janssen, Johannes Dröge, Ivan Gregor, Stephan Majda, Jessika Fiedler, Eik Dahms, et al. Critical assessment of metagenome interpretation—a benchmark of metagenomics software. *Nature methods*, 14(11):1063, 2017.

[30] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

[31] Stéfan van der Walt, S Chris Colbert, and Gael Varoquaux. The numpy array: a structure for efficient numerical computation. *Computing in Science & Engineering*, 13(2):22–30, 2011.

[32] Jia Wu, Xingquan Zhu, Chengqi Zhang, and S Yu Philip. Bag constrained structure pattern mining for multi-graph classification. *Ieee transactions on knowledge and data engineering*, 26(10):2382–2396, 2014.

[33] Yuanyuan Xu, Jun Wang, Shuai An, Jinmao Wei, and Jianhua Ruan. 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.

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

[35] Min-Ling Zhang and Zhi-Hua Zhou. A review on multi-label learning algorithms. *IEEE transactions on knowledge and data engineering*, 26(8):1819–1837, 2014.

[36] Wayne Xin Zhao, Jing Jiang, Jianshu Weng, Jing He, Ee-Peng Lim, Hongfei Yan, and Xiaoming Li. Comparing twitter and traditional media using topic models. In *European conference on information retrieval*, pp. 338–349. Springer, 2011.