MoSBI: Automated signature mining for molecular stratification and subtyping

Tim Daniel Rose¹, Thibault Bechtler¹, Octavia-Andreea Ciora¹, Kim Anh Lilian Le¹, Florian Molnar¹, Nikolai Koehler¹, Jan Baumbach³, Richard Röttger², Josch Konstantin Pauling¹

¹ LipiTUM, Chair of Experimental Bioinformatics, TUM School of Life Sciences, Technical University of Munich, Germany

² Department for Mathematics and Computer Science, University of Southern Denmark, Odense

³ Chair of Computational Systems Biology, University of Hamburg, Germany

Abstract

The improving access to increasing amounts of biomedical data provides completely new chances for advanced patient stratification and disease subtyping strategies. This requires computational tools that produce uniformly robust results across highly heterogeneous molecular data. Unsupervised machine learning methodologies are able to discover de-novo patterns in such data. Biclustering is especially suited by simultaneously identifying sample groups and corresponding feature sets across heterogeneous omics data. The performance of available biclustering algorithms heavily depends on individual parameterization and varies with their application. Here, we developed MoSBI (Molecular Signature identification using Biclustering), an automated multi-algorithm ensemble approach that integrates results utilizing an error model-supported similarity network. We evaluated the performance of MoSBI on transcriptomics, proteomics and metabolomics data, as well as synthetic datasets covering various data properties. Profiting from multi-algorithm integration, MoSBI identified robust group and disease specific signatures across all scenarios overcoming single algorithm specificities. Furthermore, we developed a scalable network-based visualization of bicluster communities that support biological hypothesis generation. MoSBI is available as an R package and web-service to make automated biclustering analysis accessible for application in molecular sample stratification.

Introduction

Optimizing treatments and improving patients' health is the goal of precision medicine. In contrast to canonical medicine, where treatments are prescribed empirically¹, precision medicine aims to identify individually adapted treatments. Nowadays, diseases are commonly diagnosed based on the international classification of diseases (ICD). This assumes that a disease shows similar symptoms in every individual, hence treatments are meant to act on the majority of symptoms. Patient stratification for precision medicine builds on the idea that a cohort of patients with varying or similar symptoms might have different molecular causes. They can then be stratified on the molecular level and divided into subgroups². Therefore, precision medicine wants to move away from classical disease definitions to characteristic signatures of molecular alterations which enable individualized treatments.

Achieving this requires understanding of molecular disease mechanisms. Unsupervised machine learning methods are best-suited, since they uncover the inherent structure of the given data and do not require labeled data, which might be biased towards classical disease understandings³. Unsupervised clustering methods seek to identify distinct subgroups over the entire features set; but it is unrealistic to assume that diseases manifest in all features. Instead they are limited to a subtype specific subset. Biclustering algorithms can meet this requirement.

Molecular data is usually available in data matrices with patient samples as columns and biomolecular features as rows. Biclustering algorithms cluster samples and biomolecules of a data matrix simultaneously. This results in sample groups with a molecular subset which characterizes the group. Numerous algorithms have been published, which try to tackle the problem from different angles. An overview about important concepts was published by Madeira and Oliveira⁴.

Similar to clustering⁵, evaluations of biclustering algorithms have shown differences in the performance under various real-life and synthetic conditions^{6,7}. A common way to improve the results of machine learning techniques are ensemble approaches. The goal is to improve robustness, consistency, novelty and stability over what single algorithms could achieve⁸. Also for biclustering problems ensemble algorithms have been proposed^{9–15}. Most of these ideas are adaptations of approaches for ensemble clustering. Some of the proposed methods are not implemented^{9,12} and therefore not easily accessible, while others are single algorithm ensemble approaches which cannot overcome limitations of one algorithm.

The analysis and interpretation of biclustering results can profit from visualizations, which show the content or relations between biclusters. Many approaches have been developed^{16–22}, which are often bound to specific algorithms or do not scale well for many biclusters¹⁷.

Here we propose a multi-algorithm biclustering ensemble approach for the stratification of molecular samples. In the manuscript, we 1) Introduce the methodology and network

visualization. 2) Evaluate the performance on multiple experimental metabolomics, proteomics and transcriptomics datasets. 3) With a framework for synthetic data generation, evaluate the approach on synthetic data. 4) Apply our approach in a multi-omics context and 5) Present open source software to make bicustering more accessible for research.

Results

A multi-algorithm ensemble bicustering approach

The steps of our ensemble approach (MoSBi - Molecular signature identification using bicustering) are described in **Figure 1A**, for full details, please refer to the methods section. At first, we selected a set of established or recently developed bicustering algorithms (**Table 1**), which are executed independently. Next, similarities between all biclusters are calculated. The similarity is described by the degree of overlap, meaning, the more samples and features shared between biclusters, results in higher similarity. Highly similar biclusters point towards the same pattern in the data. Similarities are filtered for random overlaps and a bicluster network is generated with biclusters as nodes and connections between them if they exceed a higher than random similarity. The example network shown in Figure 1A reveals several highly connected communities in the network, which are not as strongly connected between each other. By using the Louvain modularity, such communities can be extracted and converted into ensemble biclusters. Two thresholds control the size of resulting ensemble biclusters. We previously successfully utilized the principle of MoSBi to identify de-novo subtypes of non-alcoholic liver disease based on clinical lipidomics data²³.

Table 1: List of evaluated bicustering algorithms in alphabetical order.

The results of algorithms can be imported and accessed with our MoSBi R package or executed using the webtool.

Algorithm	Publication
BicARE	Gestraud et al. (2020) ²⁴
Bimax	Prelić et al. (2006) ²⁵
CC	Cheng & Church (2000) ²⁶
Fabia	Hochreiter et al. (2010) ²⁷
Isa	Bergmann et al. (2003) ²⁸
Plaid	Lazzeroni and Owen (2002) ²⁹
QUBIC	Zhang et al. (2010) ³⁰
Quest	Murali et al. (2002) ³¹
Spectral	Kluger et al. (2003) ³²
UniBic	Wang et al. (2016) ³³

Xmotifs	Murali et al. (2002) ³¹
---------	------------------------------------

Before evaluating the performance of MoSBI on multiple omics datasets, we selected a public thymic epithelial tumor dataset³⁴ to show the application and potential of our approach. Ku et al.³⁴ measured the proteome of 134 tumor, tumor adjacent and normal thymus samples and revealed significant differences in the proteome signatures of thymoma subtypes. In **Figure 1A** the similarity network of predicted biclusters colored by sample groups can be seen. Additionally, node sizes were scaled according to the number of samples. It provides an overview of the match of predicted biclusters with known information about samples, in this case cancer subtypes/tissues. While being a central part of the workflow, to compute ensemble biclusters, networks also serve as a visualization of biclustering predictions.

It is immediately obvious that clusters of nodes can be found in the network, indicating biclusters with a similar set of samples and features. This can be observed by similar color distributions of biclusters clustering together. After applying the Louvain modularity, these clusters result in network communities.

Some communities, in particular 2, 4 & 8 predominantly consist of type A, B and AB thymoma. We performed KEGG pathway enrichment of protein sets from the ensemble biclusters (**Figure 1B**). All selected communities showed significant repair mechanism pathways, which is well known for tumors to influence those pathways. Additionally community 2, which includes samples of all thymoma subtypes, indicating a common signature on the proteomic level, showed two cancer related terms.

We investigated the occurrence of proteins and samples in biclusters belonging to one community (**Figure 1C**). A difference in the distributions of samples and features can be observed. The distribution of features is strongly positively skewed with a very low mode. In contrast, the sample distribution e.g. for communities 5 and 8 has a mode close to one. This shows that biclusters inside the same community (after filtering edges for random overlaps) can carry very different features and samples. By setting thresholds, ensemble biclusters can be restricted to point to consistent patterns in the data or to allow for variability.

In **Figure 1D**, we visualize the affiliation of biclusters to algorithms, which predicted them. This reveals that biclustering algorithms tend to identify overlapping regions in the data, resulting in highly connected communities consisting only of one algorithm. This shows the necessity of taking the results of multiple biclustering algorithms into account and not relying on one but many different algorithms to capture patterns in the data beyond the specificities of a single algorithm.

The results above demonstrate the power and utility of the workflow to establish a sophisticated biclustering analysis, to generate biological hypotheses. In the following we will examine MoSBI in a multi omics context.

Figure 1: Workflow of MoSBI with exemplary network visualizations.

(A) Steps of the MoSBI approach. First, biclusters are predicted by multiple algorithms and a similarity matrix is computed, which is then filtered for larger than random overlaps using an error model. The matrix is then converted to a network which can be visualized with meta-information about samples or features. Louvain communities are then extracted and converted into ensemble biclusters. As an example the bicluster network of proteomics data from Ku et al.³⁴ is shown. Nodes represent biclusters, with edges between them if their overlap exceeds the error threshold. (B) KEGG pathway enrichment for features of selected communities 2, 4 & 8. (C) Frequency of features (upper) and samples (lower) in biclusters that belong to one community. (D) Bicluster network of proteomics data from Ku et al.³⁴. Node colors represent algorithms, by which they were predicted.

Individual biclustering algorithms vs. MoSBI

Next, we compared the individual performances of available biclustering algorithms and contrasted them with the performance of MoSBI. For that, we selected 6 published and publicly available datasets from the metabolomics, transcriptomics and proteomics disciplines (details in **Supplementary Table 1**). All datasets were analyzing cancer tissues or investigated cancer subtypes. As a gold standard, we used the condition match score to quantify the overlap between predicted biclusters and sample labels (see methods), where the relevance describes how well predicted biclusters correspond to known labels and recovery how well the labels were recovered by predictions. Additionally GO and KEGG pathway enrichment was performed to evaluate the gene sets in predicted biclusters.

The match between predictions and sample groups can be seen in **Figure 3A**. It reveals a heterogeneous performance of the individual biclustering algorithms. Spectral only predicted biclusters in two out of the six scenarios. Isa has the highest recovery on the Tang et al.³⁵ metabolomics and Ku et al.³⁴ proteomics data and both transcriptomics datasets, but has a poor performance on Yang et al.³⁶ metabolomics and Wiśniewski et al.³⁷ proteomics data. While having a good recovery, Isa never scores best on relevance. A similar behavior can be observed for Plaid, which on average performs very well for relevance, but shows low recoveries. It can also be observed that only plaid has in one proteomics dataset a relevance and recovery higher than 0.5. We then applied our ensemble approach on the predictions of all algorithms per dataset (**Figure 2A**, black marker). The ensemble approach is one of the two best performing tools in either recovery or relevance in all other datasets, except for the Tang et al.³⁵ metabolomics data, where we could observe high overlaps with other clinical confounders (**Supplementary Figure 1**). On metabolomics data, with fewer features compared to sequencing data, the communities can additionally be visualized as co-occurrence networks (**Supplementary Figure 2**). Over all six datasets, MoSBI performed second best on average by Relevance and second best by recovery after plaid and Isa, which both have poorer performance on the other scale.

To investigate the performance of the algorithms on the gene level, we performed KEGG pathway and GO biological process enrichment for the proteomics and transcriptomics datasets. In KEGG enrichment (**Figure 2B**), BicARE predicted the most biclusters with at least one significantly enriched term in three datasets. Interestingly, it did not stand out when investigating sample group labels. The ensemble method again showed a better performance than the average of biclustering algorithms. The same holds for enrichment of biological processes with GO terms (**Figure 2C**). Since all investigated proteomic and transcriptomic datasets were cancer related, we searched specifically for enriched KEGG pathways including the word “cancer”, “carcinoma” or “tumor” (**Figure 2D**). On the Wiśniewski et al.³⁷ proteomics data, only Isa and BicARE found significant terms for biclusters, but only at very low frequencies. In the Ku et al.³⁴ proteomics data, MoSBI found the most significant terms, on the two transcriptomics datasets, the second most after fabia and BicARE.

This reveals that individual biclustering algorithms peak in one or another measure or data set, but in an unpredictable manner. However, the MoSBI ensemble approach is more consistent and therefore more reliable for biclustering analysis.

Figure 2: Performance of MoSBI and individual biclustering algorithms on cancer related omics data.

For further information about the data see **Supplementary Table 1**. (A) Recovery and relevance for the condition match score of biclustering tools based on samples for cancer (subtypes). (B) Frequency of predicted biclusters per algorithm, with one or more significant KEGG terms (Adjusted p-value cut-off < 0.05). (C) Frequency of biclusters with one or more significant GO terms from the “biological process” category. (D) Frequency of predicted biclusters per algorithm, with one or more cancer related KEGG terms.

Performance on synthetic data

Evaluation on experimental data is preferable, since it accurately resembles real-life application of biclustering and stratification. Unfortunately, 2D gold standards are usually not available since many factors are influencing the molecular state of samples. Synthetic data can overcome this problem. This is frequently done to evaluate biclustering algorithms^{6,25,38}.

Based on the synthetic data generation of Prelic et al.²⁵, we developed a workflow to create synthetic scenarios, where one or multiple properties can be investigated (See **Supplementary Information**: Synthetic evaluation scenarios). We repeated previous scenarios from Prelic et al.²⁵ and created novel scenarios, covering sparsity, overlaps and mixed sizes (**Supplementary Table 2**) and evaluated them on biclustering algorithms (See methods, **Supplementary Figures 5-9**). Since molecular omics data can include missing values, we investigated the effect of sparsity on the performance of biclustering algorithms (**Figure 3A**). While the overall performance of all algorithms decreases with increased sparsity, Fabia and Isa showed a higher resilience until a sparsity of 20% (percentage of missing values in the matrix, see **Supplementary Information**) after which the results deteriorated. The relevance was more robust against sparsity and did not decrease as strongly as the recovery.

So far, synthetic evaluation focussed on the assessment of individual characteristics of the data (e.g. noise or size). Using our workflow and knowledge from previous synthetic scenarios, we defined a complex scenario, incorporating all previously mentioned manipulations to the data (**Figure 3B**). We evaluated all approaches in this scenario and added a negative binomial background to simulate unique molecular identifier RNA sequencing data (**Figure 3B left**). Performance analysis separated the tools in two groups, clearly higher performing tools consisting of Fabia, Isa and MoSBI, and the rest performing significantly inferior. Fabia shows the best recovery and the ensemble approach the best relevance, however only marginally above Fabia and Isa. Even with a poor performance of many algorithms, it can still achieve high recovery and relevance. Algorithm selection has an influence on every ensemble approach, therefore, excluding the worst performing algorithms from the ensemble approach, yields a high increase of the relevance of the ensemble approach, while the recovery remains similar (**Supplementary Figure 10A**).

Being an average, the relevance does not characterize every distribution correctly, but is widely used in biclustering evaluation studies. We therefore investigated the relevance distribution of all algorithms independently (**Figure 3C**) and combined (**Figure 3D**). Some distributions are skewed. The combined distribution is positively skewed, showing that the

majority of biclusters have very low overlap with the gold standard. Predictions by the ensemble approach show a different distribution (**Figure 3E**), where the majority of biclusters have a score above 0.5. Since an ensemble approach is sensitive to the performance of the underlying biclustering algorithms, we selected the best performing algorithms and repeated the analysis (**Supplementary Figure 10C&D**). As can be seen, the performance of MoSBi is even more evident, stating the importance of the utilized algorithms. By combining highly overlapping biclusters, MoSBi can reduce the number of mismatched biclusters. This also shows that the relevance distribution can give more detailed insights into algorithm performance. MoSBi additionally reduces the number of biclusters drastically, making an investigation of all predictions more manageable.

To investigate the performance of all algorithms under best conditions, we optimized their parameters to achieve the best possible performance (**Supplementary Figure 11**). This showed that algorithms can produce markedly better results given correct parameters compared to their standard parameters in **Figure 3B**. However this is time consuming and only possible for data with an existing gold standard. The differences between the two complex synthetic scenarios showed that parameters and performances vary widely between datasets. Therefore, an ensemble method offers an easier method to achieve good performance independently of parameter optimization.

Figure 3: Evaluation of biclustering algorithms on synthetic data.

(A) Recovery and relevance of biclustering algorithms with increasing sparsity, for one hidden shift bicluster. (B) Performance of biclustering algorithms and ensemble approach on a synthetic scenario including different bicluster types, sizes, sparsity and noise with a negative binomial distributed background (left) and normal distributed background (right). (C) Relevance distribution of biclustering algorithms for the scenario shown in (B, right). (D) Relevance distribution of all algorithms summed up from (C). (E) Relevance distribution of the predictions of the ensemble approach using the biclusters from (D).

Biclustering in a multi-omics context

Since biclustering requires a data matrix as input, it can naturally be applied to multi-omics data, when combined to one data matrix. To investigate the performance of MoSbi in a

multi-omics context, we used the TCGA breast cancer cohort from the Xena Platform³⁹, which provides omics data for multiple breast cancer subtypes. RNAseq, miRNA and protein data was run independently and combined for all biclustering algorithms. All resulting bicluster networks (**Figure 4A**) appear similar, with big Basal communities and multiple communities consisting mainly of the LumA or LumB subtype, often highly interconnected. The protein data network shows a less distinct Basal community, whereas the miRNA data network shows Her2 samples mixed with LumB samples.

In the next step, we evaluated the performance of the biclustering algorithms on the different data types. A consistent performance of most algorithms can be observed (**Figure 4B**), with only Plaid showing a high increase in relevance on the multi-omics data compared to the other data sets and did not identify any biclusters on the protein data. Only with MoSBi, a relevance and recovery higher than 0.25 could be observed in all four data sets. This shows that the multi-omics data did not yield a big performance increase for most algorithms, but rather that all data types carry the information to identify subtypes, with the ensemble approach being most robust throughout all data types.

While we did not find big differences on the overall performance, we next looked at the recovery of the subtypes individually (**Figure 4C**). Most algorithms did not recover all subtypes equally well. Isa has the highest recovery for Basal (above 0.75 for RNAseq and Multi-omics) and worst for Normal (All below 0.25). Fabia exhibits a more equal distribution, except for Normal, which has a low recovery throughout all algorithms. It can again be observed that all data types are similarly able to identify subtypes. In MoSBi, the Basal subtype has a better recovery in RNAseq and multi-omics data. Another interesting observation is that BicARE consistently recovers the LumA subtype through all data types.

In this analysis we can show that multi-omics biclustering is possible and can add value to the results. However, an individual biclustering analysis on all data types is also possible and yields similar performance. However, a combined analysis might be beneficial for a biological interpretation of biclusters, which consists of features from different omics types.

Figure 4: Biclustering on breast cancer multi-omics data.

(A) Bicluster similarity networks on TCGA breast cancer miRNA, Protein expression, RNASeq, and combined data. Biclusters are colored by subtype and node size is proportional to sample size. (B) Relevance and recovery for the condition match score on the data sets from (A) for each algorithm individually and combined with MoSbi. All algorithms were executed 10 times. (C) Recovery for subtypes on the data sets from (A) for each algorithm individually and combined with MoSbi. All algorithms were executed 10 times.

The MoSbi software suite

To make our ensemble approach and biclustering algorithms in general, accessible for scientists and provide an easy to use interface, we developed the MoSbi suite for the Identification of *Molecular Signatures using Biclustering*. MoSbi is available as a R package (<https://gitlab.lrz.de/lipitum-projects/mosbi>) and web-app (<https://exbio.wzw.tum.de/mosbi>).

Many biclustering algorithms, such as Isa²⁸ and Fabia²⁷ or the biclust package use different result formats for returning biclusters. Therefore, we developed a unified framework, which is able to import predictions from various biclustering algorithms to simplify analysis of biclustering algorithms and apply our ensemble approach. Our network-based visualizations are also available in MoSBI which can be used with our ensemble approach or single biclustering algorithms. The framework can be extended to offer support for new biclustering algorithms and integrate them into the workflow. Networks can be exported as graphML for compatibility with tools such as Cytoscape⁴⁰.

The web-app allows users without programming knowledge to stratify samples with our ensemble approach and profit from visualizations. Additionally, all biclustering algorithms can be accessed and executed with all parameters independently, if users are interested in specific algorithms. We also provide a docker image of the webtool, which allows the webtool to be deployed locally.

Discussion

Stratification of patients based on molecular omics data is a challenging task and requires modern computational tools. Unsupervised approaches are suited to identify novel subgroups in the data. Biclustering is able to find meaningful patterns in modern omics data. In contrast to traditional clustering, algorithms output not only sample subgroups, but additionally feature subsets which characterize this similarity and can be further analyzed e.g. for functional associations to find disease mechanisms. We developed a biclustering ensemble approach, which takes the results of multiple biclustering algorithms and computes ensemble biclusters using a network-based approach. This is based on the assumption that biclustering algorithms predict highly overlapping biclusters, which we could show in our work. Various biclusters pointing to the same underlying data structures can indicate robust biclusters, which are then identified with MoSBI. We showed this on thymic epithelial tumor data³⁴, where we were able to retrieve known cancer subtypes.

We demonstrated the application of MoSBI on cancer-related data sets and showed the possibility to perform a multi-omics analysis using biclustering. On various synthetic and experimental datasets, we assessed the performance of different biclustering algorithms and compared them to our ensemble approach. While Fabia and Isa on average performed best of all considered biclustering algorithms, no algorithm performed best in all scenarios and can be universally recommended. MoSBI did not always stand out but achieved a robust good performance in most scenarios. While the optimization of algorithm parameters on synthetic data could significantly improve the results, it leads to extensive run times and requires gold standard annotations, which are usually not available in real data, indicating that MoSBI is a preferable choice for biclustering. Additionally, it markedly reduces the number of biclusters. The network visualization gives an overview of the results and compared to other methods¹⁷ scales well with an increased number of biclusters.

The advantage of our ensemble approach over other biclustering ensemble approaches is that it is not algorithm-specific and via the MoSBI suite accessible as an API and graphical user interface. Unfortunately, some proposed approaches lack implementation^{9,12}.

Calculation of similarities between biclusters was proposed by Hanczar and Nadif¹¹, where the authors calculated overlaps based on sums of overlaps of rows and columns, which can result in non-zero similarities for biclusters which share rows but no columns and therefore are in fact not overlapping (**Supplementary Figure 12**). They proposed their method as a single algorithm ensemble approach, did not filter for random overlaps and applied hierarchical clustering on the similarity matrix. This introduces another parameter for the number of consensus biclusters and assigns each bicluster to an ensemble bicluster, even with low overlap. Our approach avoids this by using the Louvain modularity. Additionally, MoSBi makes further analysis more accessible, since it markedly reduces the number of predictions, while not missing important biclusters. However, as an ensemble approach, MoSBi relies on the performance of multiple biclustering algorithms. We showed how the selection of biclustering algorithms can influence the results of MoSBi (**Supplementary Figure 10C&D**). While MoSBi is robust against a few badly performing algorithms, the majority of algorithms need to identify reasonable biclusters for MoSBi in order to work correctly. With new developments on available algorithms, MoSBi can be extended to improve performance in the future.

Our methodology offers a new perspective on biclustering and can visualize detailed properties of predictions. We demonstrated how a bicluster network analysis provides additional biological and structural insights into data. Clinical or experimental conditions can be associated with biological features. Using our approach, biclustering has the potential to play a significant role in disease subtyping and understanding.

Methods

The biclustering ensemble algorithm consists of 4 major steps. These are execution of multiple biclustering algorithms, followed by a similarity computation for all returned biclusters, filtering of the similarity matrix for random overlaps and community detection on the similarity network. In the following all steps are described in detail:

1. Algorithms

Given an input matrix $M \in \mathbb{R}^{R \times C}$, we utilize different biclustering algorithms (**Table 1**) and collect their results in one combined list of biclusters $B = [B_1, B_2, \dots, B_n]$, where $B_i = (B_i^r, B_i^c)$ and $B_i^r \subset [1, \dots, R]$, $B_i^c \subset [1, \dots, C]$ is a set of row and column indices of matrix M that belongs to bicluster B_i . We implemented interfaces for all algorithms in our R package to generate this list using one unified API.

2. Similarity metrics

In the next step, pairwise similarities between all biclusters in B are computed. This is done using common similarity metrics, where the similarity is expressed as a two-dimensional overlap between biclusters. To do so, we treated a bicluster matrix as a two-dimensional area and computed their similarity in terms of overlapping areas. This is different to the additive similarity as proposed by Hanczar and Nadif¹¹. One implemented metric is the Jaccard index. Our adaption resulted in the following formula:

$$J(B_1, B_2) = \frac{|B_1 \cap B_2|}{|B_1 \cup B_2|} = \frac{|B_1^c \cap B_2^c| \times |B_1^r \cap B_2^r|}{(|B_1^r| \times |B_1^c|) + (|B_2^r| \times |B_2^c|) - (|B_1^c \cap B_2^c| \times |B_1^r \cap B_2^r|)}$$

Besides the widely used Jaccard index, also the Bray-Curtis similarity, overlap coefficient and Fowlkes–Mallows index were implemented in a similar two-dimensional fashion. This results in a similarity Matrix S with $S_{i,j} = J(B_i, B_j)$. Note that MoSBi can use any other definition of similarity as well.

3. Cut-off estimation

Since biclusters can have random overlaps that do not represent meaningful interactions, we estimate a cut-off to filter for such overlaps in the similarity matrix. This is done by randomly generating a list of biclusters B' such that $B' = [B_1', B_2', \dots, B_n']$, $|B'| = |B|$ and $B_i' = (B_i^{r'}, B_i^{c'})$ where $B_i^{r'}$, $B_i^{c'}$ are randomly drawn without replacement from $[1, \dots, R]$ and $[1, \dots, C]$ correspondingly such that $|B_i^{r'}| = |B_i^r|$ and $|B_i^{c'}| = |B_i^c|$.

To estimate the best cut-off c^* for the values in the similarity matrix S , we treat the S as an adjacency matrix and optimized for the biggest ratio between remaining edges in S and S' , where $S'_{i,j} = J(B_i', B_j')$ (To increase robustness, multiple randomizations K of B are used):

$$c^* = \underset{c}{\operatorname{argmax}} \frac{\sum_{i,j} \Theta_c(S_{i,j})}{\sum_k (\sum_{i,j} \Theta_c(S'_{i,j}^k)) / K} \quad \text{with} \quad \Theta_c: \mathbb{R} \rightarrow \{0, 1\}$$

$$x \mapsto \begin{cases} 0 & : x < c \\ 1 & : x \geq c \end{cases} \quad \text{This results in}$$

$$S_{i,j}^{c^*} \mapsto \begin{cases} 0 & : S_{i,j} < c^* \\ S_{i,j} & : S_{i,j} \geq c^* \end{cases}$$

the final and filtered similarity matrix S^{c^*} where

4. Community detection

Finally, S^{c^*} is used as an adjacency matrix with biclusters as nodes and edges representing similarities. We compute the weighted Louvain modularity⁴¹, with similarities as weights, to find bicluster communities in the network. These highly similar bicluster communities can then be converted into ensemble biclusters using 3 parameters: `min_size` which defines the minimum number of biclusters in a community to convert a community into a bicluster, smaller communities are not considered. `row_threshold` & `col_threshold`, the minimum frequency of occurrence of a row/column element in a bicluster community to be taken over into an ensemble bicluster. E.g. with values of 0.5 only genes and samples will be part of the new ensemble bicluster if they occur in at least 50% of all biclusters in the corresponding community.

Implementation

The workflow was implemented in the R programming language (version ≥ 3.6) and C++17. The web interface was realized with the Shiny web framework for R (version 1.4.0.2). The implementation for the R package is available here:

<https://gitlab.lrz.de/lipitum-projects/mosbi> and for the web application here: <https://gitlab.lrz.de/lipitum-projects/mosbi-webapp>. The workflow can be executed from our web app on our servers or on a local machine using a public Docker image. For higher throughput or for the integration of our approach into a bioinformatics pipeline, the R package can be used directly.

Visualizations

Network visualizations of the MoSbi package are implemented in R using the “igraph” package. Interactive plots in the webtool use the “visNetwork” library. All other visualizations use the “ggplot2” library in R.

Co-occurrence Networks

For co-occurrence networks, biclusters from one community were selected. From this a new network is computed with samples and features as nodes. Edges can occur between samples and samples, samples and features and features and features. An edge is drawn between two nodes, if they occur together in at least one bicluster of the community. Edges are weighted by the number of biclusters, where two nodes co-occur. For the visualization, a network layout is computed, which takes the edge weights into account.

Match score

The performance of biclustering algorithms and MoSbi was evaluated by comparing their overlap to labeled gold standard data. We used the commonly applied gene match score:

$$MS_G(M_1, M_2) = \frac{1}{|M_1|} \sum_{(G_1, C_1) \in M_1} \max_{(G_2, C_2) \in M_2} \frac{|G_1 \cap G_2|}{|G_1 \cup G_2|}$$

where M_1 and M_2 are two sets of biclusters, with each bicluster consisting of a set of genes G_i and conditions C_i (rows and columns)²⁵. To investigate sample/condition overlaps, we define the according condition match score:

$$MS_C(M_1, M_2) = \frac{1}{|M_1|} \sum_{(G_1, C_1) \in M_1} \max_{(G_2, C_2) \in M_2} \frac{|C_1 \cap C_2|}{|C_1 \cup C_2|}$$

On synthetic data, where a two dimensional (2D) gold standard is available, we define the 2D match score as the multiplicative score of both dimensions:

$$MS_{2D}(M_1, M_2) = \frac{1}{|M_1|} \sum_{(G_1, C_1) \in M_1} \max_{(G_2, C_2) \in M_2} \frac{|C_1 \cap C_2|}{|C_1 \cup C_2|} \times \frac{|G_1 \cap G_2|}{|G_1 \cup G_2|}$$

The scores can be used to compute relevance and recovery. Let M_{opt} be a set of implanted biclusters or a gold standard and M the output of a biclustering algorithm. Then, the average bicluster relevance is defined as $MS(M, M_{opt})$ and describes to what extent the biclusters found by the algorithm correspond to the true hidden biclusters in the gene, condition or both dimensions. Similarly, the average bicluster recovery is defined as $MS(M_{opt}, M)$ and describes how well each of the true biclusters is recovered by the

algorithm. The recovery and relevance score both have an optimal value of 1 indicating a perfect overlap and 0 indicating no overlap.

The match scores describe a normalized sum of values. To investigate how well all individual biclusters predicted by one algorithm match the gold standard, we investigated the relevance distribution $RD = [rd_1, rd_2, \dots, rd_n]$ with n as the number of biclusters in set of biclusters M and

$$rd_i = \max_{(G_{opt}, C_{opt}) \in M_{opt}} \frac{|C_i \cap C_{opt}|}{|C_i \cup C_{opt}|} \times \frac{|G_i \cap G_{opt}|}{|G_i \cup G_{opt}|},$$

where C_i and G_i the columns and rows of bicluster M_i .

Experimental omics data

We evaluated the biclustering algorithms and MoSBI on six publicly available metabolomics (Tang et al.³⁵, Yang et al.³⁶), proteomics (Ku et al.³⁴, Wiśniewski et al.³⁷) and transcriptomics (Berger et al.⁴², Curtis et al.⁴³) datasets (See **Supplementary Table 1**). Feature wise z-score were computed for all datasets, and prior to that log2 transformed (except for Ku et al.³⁴ and Curtis et al.⁴³, which already showed a normal distribution). Transcriptomics data was filtered for genes with 80% coverage in all samples and filtered 5000 most variant genes, to reduce algorithm runtime. Gene set/pathway enrichment was performed using the “clusterProfiler” R package using the “enrichGO” (biological process enrichment) and “enrichKEGG” functions.

TCGA Breast Cancer data was downloaded from the Xena Platform³⁹. RNAseq transcriptomics data was processed as described above, miRNA and protein data was filtered for 80% coverage in all samples and z-score transformed. Only samples occurring in all three datasets were considered for the individual and multi-omics analysis, which resulted in 484 samples with measurements for all three data types.

Synthetic data generation

To investigate the performance of tools in a controlled environment with a fully known gold standard, we developed a pipeline to generate synthetic datasets with implanted biclusters and additional properties such as noise and sparsity. The pipeline is shown in **Supplementary Figure 1**.

A detailed description of all synthetic scenarios is available in the supplementary information.

Acknowledgments

TDR, NK, and JKP are funded by the Bavarian State Ministry of Science and the Arts in the framework of the Bavarian Research Institute for Digital Transformation (bidt, grant LipiTUM).

Author contributions

TDR, JB, RR and JKP conceived and designed the study. TDR implemented the approach as a package. TDR and TB developed the webtool. TDR, KALL, TB, OC, FM, NK performed the evaluation of the algorithms and provided feedback about the implementation. All authors provided critical feedback about the study and helped in writing the manuscript.

References

1. Trusheim, M. R., Berndt, E. R. & Douglas, F. L. Stratified medicine: strategic and economic implications of combining drugs and clinical biomarkers. *Nat. Rev. Drug Discov.* **6**, 287–293 (2007).
2. Khakabimamaghani, S. & Ester, M. BAYESIAN BICLUSTERING FOR PATIENT STRATIFICATION. in *Pacific Symposium on Biocomputing* 345–356 (World Scientific Publishing Co. Pte Ltd, 2016).
3. Lazareva, O. *et al.* BiCoN: Network-constrained biclustering of patients and omics data. *Bioinformatics* (2020) doi:10.1093/bioinformatics/btaa1076.
4. Madeira, S. C. & Oliveira, A. L. Biclustering algorithms for biological data analysis: a survey. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* vol. 1 24–45 (2004).
5. Wiwie, C., Baumbach, J. & Röttger, R. Comparing the performance of biomedical clustering methods. *Nat. Methods* **12**, 1033–1038 (2015).
6. Padilha, V. A. & Ricardo J G. A systematic comparative evaluation of biclustering techniques. *BMC Bioinformatics* vol. 18 (2017).
7. Pontes, B., Giráldez, R. & Aguilar-Ruiz, J. S. Biclustering on expression data: A review. *J. Biomed. Inform.* **57**, 163–180 (2015).
8. Vega-Pons, S. & Ruiz-Shulcloper, J. A SURVEY OF CLUSTERING ENSEMBLE ALGORITHMS. *International Journal of Pattern Recognition and Artificial Intelligence*

- vol. 25 337–372 (2011).
9. Hanczar, B. & Nadif, M. Ensemble methods for biclustering tasks. *Pattern Recognition* vol. 45 3938–3949 (2012).
10. Aggarwal, G. & Gupta, N. BiETopti-BiClustering Ensemble Using Optimization Techniques. *Advances in Data Mining. Applications and Theoretical Aspects* 181–192 (2013) doi:10.1007/978-3-642-39736-3_14.
11. Hanczar, B. & Nadif, M. Using the bagging approach for biclustering of gene expression data. *Neurocomputing* vol. 74 1595–1605 (2011).
12. Hanczar, B. & Nadif, M. Unsupervised Consensus Functions Applied to Ensemble Biclustering. *Proceedings of the 3rd International Conference on Pattern Recognition Applications and Methods* 30–39 (2014).
13. Yin, L. & Liu, Y. Ensemble biclustering gene expression data based on the spectral clustering. *Neural Computing and Applications* vol. 30 2403–2416 (2018).
14. Aggarwal, G. & Gupta, N. BEMI Bicluster Ensemble Using Mutual Information. *2013 12th International Conference on Machine Learning and Applications* (2013) doi:10.1109/icmla.2013.65.
15. Kasim, A. Applied Biclustering Methods for Big and High-Dimensional Data Using R. (2016) doi:10.1201/9781315373966.
16. Heinrich, J., Seifert, R., Burch, M. & Weiskopf, D. BiCluster Viewer: A Visualization Tool for Analyzing Gene Expression Data. *Advances in Visual Computing* 641–652 (2011) doi:10.1007/978-3-642-24028-7_59.
17. Aouabed, H., Santamaría, R. & Elloumi, M. VisBicluster: A Matrix-Based Bicluster Visualization of Expression Data. *Journal of Computational Biology* (2020) doi:10.1089/cmb.2019.0385.
18. Grothaus, G. A., Mufti, A. & Murali, T. M. Automatic layout and visualization of biclusters. *Algorithms Mol. Biol.* **1**, 1–11 (2006).

19. Santamaria, R., Theron, R. & Quintales, L. BicOverlapper 2.0: visual analysis for gene expression. *Bioinformatics* vol. 30 1785–1786 (2014).
20. Barkow, S., Bleuler, S., Prelic, A., Zimmermann, P. & Zitzler, E. BicAT: a biclustering analysis toolbox. *Bioinformatics* vol. 22 1282–1283 (2006).
21. Streit, M. *et al.* Furby: fuzzy force-directed bicluster visualization. *BMC Bioinformatics* vol. 15 (2014).
22. Santamaria, R., Theron, R. & Quintales, L. BicOverlapper: A tool for bicluster visualization. *Bioinformatics* vol. 24 1212–1213 (2008).
23. Vvedenskaya, O. *et al.* Non-alcoholic fatty liver disease Stratification by Liver Lipidomics. *J. Lipid Res.* 100104 (2021).
24. Gestraud, P., Brito, I. & Barillot, E. *BicARE : Biclustering Analysis and ResultsExploration*.
<https://www.bioconductor.org/packages/release/bioc/html/BicARE.html> (2020).
25. Prelić, A. *et al.* A systematic comparison and evaluation of biclustering methods for gene expression data. *Bioinformatics* vol. 22 1122–1129 (2006).
26. Cheng, Y. & Church, G. M. Biclustering of expression data. *Ismb* 8, 93–103 (2000).
27. Hochreiter, S. *et al.* FABIA: factor analysis for bicluster acquisition. *Bioinformatics* vol. 26 1520–1527 (2010).
28. Bergmann, S., Ihmels, J. & Barkai, N. Iterative signature algorithm for the analysis of large-scale gene expression data. *Physical Review E* vol. 67 (2003).
29. Lazzeroni, L. & Owen, A. PLAID MODELS FOR GENE EXPRESSION DATA. *Stat. Sin.* 12, 61–86 (2002).
30. Zhang, Y. *et al.* QUBIC: a bioconductor package for qualitative biclustering analysis of gene co-expression data. *Bioinformatics* btw635 (2016)
doi:10.1093/bioinformatics/btw635.
31. Murali, T. M. & Kasif, S. EXTRACTING CONSERVED GENE EXPRESSION MOTIFS

FROM GENE EXPRESSION DATA. *Biocomputing 2003* (2002)

doi:10.1142/9789812776303_0008.

32. Kluger, Y. Spectral Biclustering of Microarray Data: Cocustering Genes and Conditions. *Genome Research* vol. 13 703–716 (2003).
33. Wang, Z., Li, G., Robinson, R. W. & Huang, X. UniBic: Sequential row-based biclustering algorithm for analysis of gene expression data. *Scientific Reports* vol. 6 (2016).
34. Ku, X. *et al.* Deciphering tissue-based proteome signatures revealed novel subtyping and prognostic markers for thymic epithelial tumors. *Mol. Oncol.* **14**, 721–741 (2020).
35. Tang, X. *et al.* A joint analysis of metabolomics and genetics of breast cancer. *Breast Cancer Res.* **16**, 415 (2014).
36. Yang, Y. *et al.* Integrated microbiome and metabolome analysis reveals a novel interplay between commensal bacteria and metabolites in colorectal cancer. *Theranostics* **9**, 4101–4114 (2019).
37. Wiśniewski, J. R. *et al.* Absolute Proteome Analysis of Colorectal Mucosa, Adenoma, and Cancer Reveals Drastic Changes in Fatty Acid Metabolism and Plasma Membrane Transporters. *J. Proteome Res.* **14**, 4005–4018 (2015).
38. Eren, K., Deveci, M., Küçüktunç, O. & Çatalyürek, Ü. V. A comparative analysis of biclustering algorithms for gene expression data. *Brief. Bioinform.* **14**, 279–292 (2013).
39. Goldman, M. J. *et al.* Visualizing and interpreting cancer genomics data via the Xena platform. *Nat. Biotechnol.* **38**, 675–678 (2020).
40. Shannon, P. *et al.* Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Res.* **13**, 2498–2504 (2003).
41. Blondel, V. D., Guillaume, J.-L., Lambiotte, R. & Lefebvre, E. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment* vol. 2008 P10008 (2008).

42. Berger, A. C. *et al.* A Comprehensive Pan-Cancer Molecular Study of Gynecologic and Breast Cancers. *Cancer Cell* **33**, 690–705.e9 (2018).
43. Curtis, C. *et al.* The genomic and transcriptomic architecture of 2,000 breast tumours reveals novel subgroups. *Nature* **486**, 346–352 (2012).