BICLUSTERING FMRI TIME SERIES

Eduardo Jorge do Nascimento Rolha Castanho

Mestrado em Ciência de Dados

Dissertação orientada por:

Professora Doutora Sara Alexandra Cordeiro

2020
“Nature uses only the longest threads to weave her patterns, so each small piece of her fabric reveals the organization of the entire tapestry.”

— Richard Feynman, *The Character of Physical Law*
Acknowledgements

This work would not have been possible without the FCT funding to NEUROCLINOMICS2 (PTDC/EEI-SII/1937/2014) research project.

First, I must thank my advisor, Prof. Sara Madeira, for all trust deposited in me this year. The relation between an advisor and a student is somewhat complex, and I believe it must go beyond the scientific aspects of the work. Without her trust in me, this thesis would never be finished in time. It gave me the strength to keep working for long late hours during the adversities. I can only say that I am really excited about the idea of extending this relation for my next step in my academic path.

Generating scientific work is a creative process, and inspiration, ideas, solutions, problems can come from anywhere. Simple talks can cause some unexpected insights that guide research for a long time. In that sense, I must thank my colleagues João Lobo and Miguel Silva for all the support during the thesis’s execution. Not only I had the luxury of having assistance in practical issues that would take many hours to solve, but for the ideas I had when discussing their work. Additionally, I must thank Professor Raquel Barreira for reviewing this work and providing her insights about it.

Last but not least, there is my family. It’s tough to describe contributions from people who raise me into the person I am. I can only expect to be able to retribute adequately during all my life. However, I must provide extraordinary gratitude to my grandfather João for teaching me the most unexpected things.
Resumo

**Biclustering** é um método de análise que procura gerar clusters tendo em conta simultaneamente as linhas e as colunas de uma matriz de dados. Este método tem sido vastamente explorado em análise de dados genéticos. Apesar de diversos estudos reconhecerem as capacidades deste método de análise em outras áreas de investigação, as últimas duas décadas tem sido marcadas por um número elevado de estudos aplicados em dados genéticos e pela ausência de uma linha de investigação que explore as capacidades de biclustering fora desta área tradicional.

Esta tese segue pistas que sugerem potencial no uso de biclustering em dados de natureza espaço-temporal. Considerando o contexto particular das neurociências, esta tese explora as capacidades dos algoritmos de biclustering em extrair conhecimento das séries temporais geradas por técnicas de imagem por ressonância magnética funcional (fMRI).

Esta tese propõe uma metodologia para avaliar a capacidade de algoritmos de biclustering em estudar dados fMRI, considerando tanto dados sintéticos como dados reais. Para avaliar estes algoritmos, usamos métricas de avaliação interna. Os nossos resultados discutem o uso de diversas estratégias de busca, revelando a superioridade de estratégias exaustivas para obter os biclusters mais homogéneos. No entanto, o elevado custo computacional de estratégias exaustivas ainda são um desafio e é necessário pesquisa adicional para a busca eficiente de biclusters no contexto de análise de dados fMRI.

Propomos adicionalmente uma nova metodologia de análise de biclusters baseada em algoritmos de descoberta de padrões para determinar os padrões mais frequentes presentes nas soluções de biclustering geradas. Um bicluster não é mais que um hipervértice num hipergrafo. Extrair padrões frequentes numa solução de biclustering implica extrair os hipervértices mais significativos. Numa primeira abordagem, isto permite entender relações entre regiões do cérebro e traçar perfis temporais que métodos tradicionais de estudos de correlação não são capazes de detetar. Adicionalmente, o processo de gerar os biclusters permite filtrar ligações pouco interessantes, permitindo potencialmente gerar hipervgrafos de forma eficiente.

A questão final é o que podemos fazer com este conhecimento. Conhecer a relação entre regiões do cérebro é o objetivo central das neurociências. Entender as ligações entre regiões do cérebro para vários sujeitos permitem traçar perfis. Nesse caso, propomos uma metodologia para extrapolar biclusters para dados tridimensionais e efetuar triclustering. Adicionalmente, entender a ligação entre zonas cerebrais permite identificar doenças como a esquizofrenia, demência ou o Alzheimer.

Este trabalho aponta caminhos para o uso de biclustering na análise de dados espaço-temporais, em particular em neurociências. A metodologia de avaliação proposta mostra evidências da eficácia do bi-
clustering para encontrar padrões locais em dados de fMRI, embora mais trabalhos sejam necessários em relação à escalabilidade para promover a aplicação em cenários reais.

**Palavras Chave:** Biclustering, fMRI, Neurociência, Análise de séries temporais.
The effectiveness of biclustering, simultaneous clustering of both rows and columns in a data matrix, has been primarily shown in gene expression data analysis. Furthermore, several researchers recognize its potentialities in other research areas. Nevertheless, the last two decades witnessed many biclustering algorithms targeting gene expression data analysis and a lack of consistent studies exploring the capacities of biclustering outside this traditional application domain.

Following hints that suggest potentialities for biclustering on Spatio-temporal data, particularly in neurosciences, this thesis explores biclustering’s capacity to extract knowledge from fMRI time series.

This thesis proposes a methodology to evaluate biclustering algorithms’ feasibility to study the fMRI signal, considering both synthetic and real-world fMRI datasets. In the absence of ground truth to compare bicluster solutions with a reference one, we used internal valuation metrics. Results discussing the use of different search strategies showed the superiority of exhaustive approaches, obtaining the most homogeneous biclusters. However, their high computational cost is still a challenge, and further work is needed for the efficient use of biclustering in fMRI data analysis.

We propose a new methodology for analyzing biclusters based on performing pattern mining algorithms to determine the most frequent patterns present in the generated biclustering solutions. A bicluster is nothing more than a hyperlink in a hypergraph. Extracting frequent patterns in a biclustering solution implies extracting the most significant hyperlinks. In a first approach, this allows to understand relationships between regions of the brain and draw temporal profiles that traditional methods of correlation studies cannot detect. Additionally, the process of generating biclusters allows filtering uninteresting links, potentially allowing to generate hypergraphs efficiently.

The final question is, what can we do with this knowledge. Knowing the relationship between brain regions is the central objective of neurosciences. Understanding the connections between regions of the brain for various subjects allows one to draw profiles. In this case, we propose a methodology to extrapolate biclusters to three-dimensional data and perform triclustering. Additionally, understanding the link between brain zones allows identifying diseases like schizophrenia, dementia, or Alzheimer’s.

This work pinpoints avenues for the use of biclustering in Spatio-temporal data analysis, in particular neurosciences applications. The proposed evaluation methodology showed evidence of biclustering’s effectiveness in finding local fMRI data patterns, although further work is needed regarding scalability to promote the application in real scenarios.

**Keywords:** Biclustering, fMRI, Neurosciences, Time Series Analysis.
Resumo Alargado

No contexto de aprendizagem automática, detetar padrões é responsabilidade dos algoritmos de aprendizagem não-supervisionada, em particular algoritmos de clustering. Estes algoritmos usam a semelhança de variáveis para agrupar objectos em grupos. Esta abordagem é considerada demasiada rígida, uma vez que exige que os grupos sejam parecidos em todas as variáveis em estudo. Para abordar este problema foram desenvolvidos algoritmos capazes de efetuar agrupamento de dados em várias dimensões. Estes algoritmos de agrupamento são conhecidos por biclustering.

Ao longo das últimas duas décadas, dezenas de algoritmos de biclustering têm sido desenvolvidos e aplicados maioritariamente no contexto de dados de expressão genética. Apesar de existirem indícios que a capacidade destes algoritmos tem potencial em contextos para além deste contexto tradicional, não existe atualmente uma linha de investigação consistente neste contexto. A insistência no contexto de expressão genética está a gerar um viés na investigação com duas grandes consequências. Em primeiro lugar, os algoritmos são desenvolvidos, testados e comparados sempre neste contexto, o que impede determinar o seu potencial geral. Em segundo lugar, a investigação em dados genéticos usa métricas de avaliação específica para dados genéticos que não podem ser usados noutros contextos. Assim sendo, o caminho para extrair padrões usando biclustering ainda é confuso.

O cérebro é o órgão central do sistema nervoso humano. Dentro dele existe uma densa rede de células interligadas entre si. Esta rede de células é responsável pela vasta maioria dos cálculos necessários para o funcionamento do corpo. Esta rede está dividida em regiões altamente especializadas, e é particularmente interessante entender como é que estas regiões interagem entre si, e como é que elas respondem a um estímulo. Duas regiões cerebrais afastadas podem interagir juntas, enquanto duas regiões próximas podem exigir um sinal muito diverso.

Para detetar atividade cerebral, diversas tecnologias foram desenvolvidas com diferentes objetivos. Nos últimos anos, técnicas de imagem por ressonância magnética funcional (fMRI) revelam-se particularmente promissoras devido à sua resolução espacial e temporal. Portanto, esta técnica é considerada ideal para estudar padrões e relações no cérebro humano, sendo capaz de detetar atividade cerebral em questão de segundos e a milímetros da sua origem. Os conceitos de espaço e de tempo não são novos na ciência, e as neurociências partilham propriedades fundamentais com domínios como redes de transporte, ciências da terra e do clima. Nós acreditamos que o contexto específico de dados espaço-temporais é particularmente promissor para estudar a aplicação de algoritmos de biclustering. Aplicações bem sucedidas de biclustering em dados espaço-temporais, bem como a presença de estruturas semelhantes a biclusters
em dados gerados por levam-nos a crer que esta é uma área particularmente promissora para a avaliação de algoritmos de biclustering.

Em primeiro lugar, o nosso objetivo passa por avaliar quantitativamente a capacidade de diferentes algoritmos de biclustering em extrair biclusters homogêneos. As abordagens atuais de comparação de algoritmos de biclustering passam pelo uso de métricas específicas do contexto de dados de expressão genética, portanto nós desenvolvemos uma nova metodologia de análise. Em primeiro lugar, começamos por escolher os sete algoritmos de biclustering para serem comparados. Escolhemos os algoritmos **BicPAM, CCC, ISA, FABIA, Spectral, XMotifs e Bimax**. Estes algoritmos foram escolhidos de forma a refletir o atual estado da arte em desenvolvimento de algoritmos de biclustering, cobrindo diversas estratégias de pesquisa. Adicionalmente procuramos algoritmos que fossem capazes de lidar com padrões temporais de natureza aditiva ou multiplicativa e robustos à existência de ruído. A estes algoritmos de biclustering adicionamos três algoritmos de clustering tradicionalmente usados em dados fMRI: **k-means, método hierárquico de ward e spectral**. Foram escolhidas três coleções de dados, tendo em conta dados reais e sintéticos. Usaram-me quatro métricas de avaliação interna para avaliar e comparar os biclusters gerados (variância, msr, msre e erro virtual). Destas quatro medidas, três delas são específicas para detetar padrões de biclustering específicos (constante, aditivos e multiplicativos) e a quarta métrica é genérica, capaz de capturar os três tipos de padrões. As correções entre as quatro medidas permitem tirar informações sobre os padrões intrínsecos que esperamos encontrar neste conjunto de dados.

Os nossos resultados demonstram que, apesar de biclustering gerar estruturas com uma homogeneidade superior à de tradicionais métodos de clustering, a sua performance depende do tipo de busca. Algoritmos como FABIA e Spectral não foram capazes de detetar biclusters particularmente homogêneos. As razões para isso são diferentes. Enquanto o FABIA está naturalmente limitado pelo reduzido número de biclusters que pode gerar (portanto, serão menos homogêneos), o Spectral é limitado pela sua busca de um tipo específico de biclusters (estruturas em tabuleiro de xadrez). Os algoritmos estocásticos, ISA e XMotifs conseguem encontrar alguns biclusters com homogeneidade interessante, em particular o ISA. Este tipo de algoritmos têm a vantagem de serem rápidos. No entanto, os melhores resultados ocorrem para os algoritmos exaustivos (BicPAM e CCC). Em relação ao CCC, é um método interessante, não apenas por permitir obter os melhores resultados em termos de homogeneidade, mas também por gerar biclusters temporalmente contínuos, que têm uma interpretação trivial. Apesar deste algoritmo ser exaustivo, a restrição de contiguidade temporal leva a que o algoritmo seja computacionalmente menos exigente do que outras abordagens exaustivas.

Estes algoritmos geram estruturas com uma homogeneidade superior à de tradicionais métodos de clustering, motivando o uso de biclustering para analisar dados fMRI, mas não chega. É preciso ainda extrair conhecimento destas estruturas. Um problema na investigação de biclustering é o fosso entre o desenvolvimento de algoritmos de biclustering e a aplicação prática destes algoritmos para extrair conhecimento. Apesar de ser fácil analisar um bicluster individual, analisar individualmente biclusters não é viável, principalmente tendo em conta que os recentes algoritmos de biclustering geram milhares de biclusters.

Por isso, propomos uma nova metodologia de análise de biclusters baseada em algoritmos de descoberta de padrões para determinar os padrões mais frequentes presentes nas soluções de biclustering.
geradas. Um bicluster não é mais que um hipervértice num hipergrafa. Extraírem padrões frequentes numa solução de biclustering implica extraírem os hipervértices mais significativos. Numa primeira abordagem, isto permite entender relações entre regiões do cérebro e traçar perfis temporais que métodos tradicionais de estudos de correlação não são capazes de detetar. Adicionalmente, o processo de gerar os biclusters permite filtrar ligações pouco interessantes, permitindo potencialmente gerar hipergrafos de forma eficiente (em comparações com métodos tradicionais combinatórios). A questão final é o que podemos fazer com este conhecimento. Conhecer a relação entre regiões do cérebro é o objetivo central das neuroneciências. Entender as ligações entre regiões do cérebro para vários sujeitos permitem traçar perfis. Nesse caso, propomos uma metodologia para extrapolar biclusters para dados tridimensionais e efetuar triclustering. Adicionalmente, entender a ligação entre zonas cerebrais permite identificar doenças como a esquizofrenia, demência ou o Alzheimer.

Com o desenvolvimento de diversas técnicas de posicionamento como Global Position System (GPS), a quantidade, qualidade e disponibilidade dos dados espaço-temporais é cada vez melhor. A extração de conhecimentos a partir de dados espaço-temporais é extremamente importante para diversas aplicações no mundo real, incluindo a compreensão da mobilidade humana, planeamento urbano, segurança pública, saúde e gestão ambiental. A complexidade dos dados espaço-temporais e das relações intrínsecas limita a utilidade das técnicas convencionais de ciência de dados para extrair padrões espaço-temporais.

Esta tese expande as capacidades do biclustering, demonstrando que é um método igualmente promissor para dados fMRI. Estes conjuntos de dados partilham propriedades fundamentalmente com áreas como ciências do clima, epidemiologia e sociologia, áreas que lidam regularmente com dados com natureza espaço-temporal e que nós esperamos ser capazes de beneficiar com este estudo.
Contents

1 Introduction ................................................. 1
   1.1 Thesis Outline ........................................... 2

2 Background .................................................. 3
   2.1 Capturing neuronal activity: fMRI ............................ 3
       2.1.1 fMRI as part of the Spatio-temporal data family .......... 4
       2.1.2 Temporal response to stimulus .......................... 6
   2.2 Clustering ............................................... 7
   2.3 Biclustering ............................................. 9
       2.3.1 Bicluster Patterns ................................... 10
       2.3.2 Search strategies and biclustering algorithms ............ 11
       2.3.3 Evaluating Biclusters .............................. 16
   2.4 Triclustering ........................................... 18
   2.5 Pattern mining .......................................... 19
   2.6 Graph and Hypergraph theory ............................ 19
   2.7 Supervised Learning: Classification ....................... 21

3 Related Work ............................................... 23
   3.1 Biclustering reviews .................................... 23
   3.2 fMRI data Analysis ....................................... 25
       3.2.1 Simulating fMRI data ................................. 25
       3.2.2 Preprocessing fMRI data ............................. 26
       3.2.3 Clustering: Parceling the brain ....................... 26
       3.2.4 Graph Approaches: Measures of brain connectivity .... 27
       3.2.5 Brain connectivity diseases and the case of schizophrenia 28
   3.3 Biclustering, Triclustering, and Spatio-temporal data ....... 29
   3.4 Bicluster-based classification ............................. 30
   3.5 Motivating Biclustering .................................. 31

4 Biclustering fMRI Time Series: A Comparative Approach .... 33
   4.1 Methodology ............................................. 34
       4.1.1 Types of signal ....................................... 34
List of Figures

2.1 Two different application cases using the same base data type. On the left, a spatial map representing Dutch meteorological stations. These stations are spatially fixed while doing their measurement periodically (Figure adapted from [141]). On the right, the MRI machine obtains a regular grid of observations of the (ideally) immobile brain (Figure adapted from [103]). Despite the difference in the application domain, the observations are done under the same observational principles. .................................................. 5

2.2 The Standard canonical model used to model the Hemodynamic response function (HRF). Neuronal activity will cause an increase in the inflow of oxygenated blood to active regions of the brain (Figure adapted from [67]). .............................................. 6

2.3 The BOLD response is modeled using a convolution between the stimulus function and the standard hemodynamic function. This diversity translates into additional layers of complexity for analyzing fMRI temporal data (Figure adapted from [81]). ...................... 7

2.4 In clustering, similarities in data features can be used to generate clusters. These clusters generate strips in the data matrix. ................................................. 8

2.5 A bicluster is a subset of rows and columns that exhibit similar behavior. This allows to find a more diverse set of structures in comparison to traditional clustering (Figures adapted from [84]). ......................................................... 10

2.6 The same row and column parameters can generate different coherent biclusters depending on their relation being additive (shifting) or multiplicative (scaling). ................. 11

2.7 Illustration of the bimax algorithm. The original matrix is fragmented into three submatrices. One has only zeros and was removed. The algorithm then repeats the division for the two remaining matrices (Figure adapted from [107]). ......................... 13

2.8 The outer product of the two vectors results in a matrix with a bicluster. The sum of the multiple biclusters allow to retrieve the original data matrix, plus some noise (Figure adapted from [64]). ......................................................... 13

2.9 Objective of spectral biclustering. (Figure adapted from[79]). ......................... 14

2.10 BicPAM methodology. BicPAM improved PM-based biclustering with several strategies to improve the general biclustering quality (Figure adapted from [60]). ................. 15
2.11 An alphabet transformation is applied to the data by adding the column index to the discretized dataset. Then, a suffix tree is constructed from the data. The biclusters are identified in specific nodes in the tree and found by doing a depth-first search in the tree. In the figure, biclusters are represented by the ball icon (Figure adapted from [28]).

2.12 Traditional machine learning datasets and datasets used in the triclustering task (adapted from [61]).

2.13 Examples of undirected and directed graphs. While undirected graphs have simetrical relations among variables, directed graphs have asymmetric relations leading to the existence of two different edges in two nodes (figures adapted from [1]).

2.14 An hypergraph. In contrast to traditional graph, hypergraphs establish connections with more than 2 nodes (Figure adapted from [131]).

2.15 On the left an illustrative example of the classification task. Given information about what we know are dogs and cats, predict if a new animal is a dog or a cat. On the right, a visualization of decision trees, an example of a classifier. This type of classifier uses rules learnt from data to distinguish between labels.

3.1 Networks are commonly represented by their connectivity matrices, with rows and columns representing nodes and matrix entries representing links (Figure adapted from [113]).

3.2 Connectivity matrices for a schizophrenic patient an a control. While the densities are equivalent, the minimum and mean weight for the patient is lower than for the control. This points that patient’s connectivity matrix will contain more low-value when compared to a healthy patient (Figure adapted from [46]).

3.3 In the data matrix, a colored element indicates that the feature is present for a given subject. A, B, C, and D are submatrices in X that allow to distinguish between healthy and schizophrenic patients (Figure adapted from [14]).

3.4 After discovering the biclusters for a group of subjects, a data matrix can be obtained locating the presence of some bicluster in a subject, and then used for classification tasks. The biclusters (sets of features and corresponding representative values) are used as features (Figure adapted from [90]).

3.5 The relation among brain regions isn’t constant over time. Different regions will react together under some specific temporal stamps. Traditional clustering will not be able to detect the visible biclusters-like structures (Figures adapted from [17]).

4.1 Comparing the general capacity of biclustering algorithms versus the two clustering variants in synthetic and real data. This is done by aggregating the results from all algorithms and comparing the median value of the metric. The results motivate the capacity of biclustering to obtain promising results for analysing the data.

4.2 Virtual Error measure for every tested algorithm in our synthetic data collection. Despite having great oscillations, the median performance of the exhaustive approaches (CCC and BicPAM) show promising results in comparison with the remaining biclustering approaches.
4.3 Virtual Error measure for every tested algorithm in our real data collection. Despite the biclustering algorithms not being indisputable better than the traditional clustering, the use of exhaustive biclustering approaches such as CCC and BicPAM show a good capacity of generating coherent biclusters.

4.4 Comparison between the best generated biclusters for each biclustering algorithm. From the previous biclustering solution, the top-k biclusters (filtering by the virtual error and removing biclusters with virtual error smaller than 0.001). The results follow the previous conclusions pointing for a high capacity of the exhaustive algorithms to generate good biclusters. Additionally the ISA results suggest that, while is general performance is bad, it does have the capability of generating some good biclusters.

4.5 Illustrative biclusters found in synthetic and real data. Biclustering is able to discover a subset of brain regions that behave similarly during a temporal interval. This type of structures go beyond traditional clustering.

5.1 Alternative views of the previous biclusters.

5.2 Filtered hypergraph considering only the top 5 more frequent itemsets. In this particular case, most of hyperedges are actually traditional graph edges. If we force the itemset search to find only links of size two or smaller, this technique will define traditional graphs.

5.3 On the left, pairwise correlation between temporal stamps. On the right, the frequencies for each temporal stamp. Both plots agree in some particular activity near stamps 100 and 125.

5.4 Measure of correlation among brain regions compared to state of the art Pearson correlation coefficient. Despite the differences in the way the correlation is computed, the plots are similar.

5.5 Comparing the original dataset with the length 2 region-time itemset allows to observe what regions and what time stamps were most used in the biclustering. Similar structures can be found.

5.6 Effective connectivity between pairs of regions. As expected, the confidence measure is not symmetrical. Additionally, the confidence plot highlights correlations that are not present in the support case. Notice however that the plots are similar.

5.7 Number of hypernodes generated by the biclustering algorithms. With the increase of possible hypernodes, the quantity of generated hyperedges by the biclustering algorithms decreases when comparing to the combinatory limit.

5.8 Pairwise region correlation in two algorithms. The similarities between the two plots give us some confidence that the two algorithms are discovering some significant pattern, at least pairwise ones.

5.9 Correlation between biclustering results. As visible before, the two exhaustive algorithms, CCC and BicPAM seem to locate the same type of relation between items. For the remaining biclustering algorithms this correlation is not that strong.

5.10 Activity for every available patient according the two considered dimensions: Brain regions and temporal stamps. Some activity only happens for some patients.
List of Tables

2.1 General spatio-temporal properties ................................................................. 4
2.2 Clustering categories (Table adapted from [57]).................................................. 8
2.3 Biclustering algorithms search strategy categories.......................................... 12

4.1 Biclustering algorithms considered for this study. Additionally, they are going to be compared to the most used algorithms for clustering fMRI data: k-means, spectral and ward’s hierarchical methods. For clustering, we are using the scikit-learn implementations. ........................................ 35
4.2 Median values for the four selected measures for the first synthetic dataset, with uncertainties given by the standard deviation (except for the case of smsr were the standard deviation is orders of magnitude higher than the median value). From this results it is visible that A) The high values of uncertainty discourage focus on optimizing the biclustering method parameters and B) Choosing the right evaluation metric is important, however in most of the biclusters cases they seem to agree for the same ”best” configuration ............................................... 39
4.3 Median values (and associated standard deviation) for the typical bicluster dimension parameters in both data collections: number of regions in each bicluster, number of temporal stamps and bicluster area. When comparing this results to the virtual error values, a apparent relation comes between the bicluster size and the associated virtual error, which make sense. ............................................................. 44
4.4 Correlation between the virtual error and the three specific coherence measures: Variance (constant biclusters), MSR (shifting biclusters) and SMSR (scaling biclusters). Most of the algorithms agree that the expected patterns are of shifting nature. .............. 45

5.1 Top five more frequent itemsets (there are other itemsets with support 1/3 that for simplicity where excluded) ................................................................. 52
5.2 Traditional biclustering-based classification uses the raw found biclusters for classification (or even a clusters of biclusters). Our approach uses not the biclusters as features, but the frequent patterns found in biclustering solutions. The rule interest metrics can be used to define the weigh of the features. ....................................................... 63
5.3 Results from our biclustering classification, showing that our approach have a simillar performance than a pearson-based discriminator. ........................................ 64
## Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DM</td>
<td>Data Mining</td>
</tr>
<tr>
<td>ML</td>
<td>Machine Learning</td>
</tr>
<tr>
<td>MRI</td>
<td>Magnetic Resonance Imaging</td>
</tr>
<tr>
<td>fMRI</td>
<td>functional Magnetic Resonance Imaging</td>
</tr>
<tr>
<td>EEG</td>
<td>Electroencephalography</td>
</tr>
<tr>
<td>MEG</td>
<td>Magnetoencephalography</td>
</tr>
<tr>
<td>PET</td>
<td>Positron Eletron Topography</td>
</tr>
<tr>
<td>ST</td>
<td>Spatio-temporal</td>
</tr>
<tr>
<td>BOLD</td>
<td>Blood Oxygenation Level Dependent</td>
</tr>
<tr>
<td>ISA</td>
<td>Iterative Signature Algorithm</td>
</tr>
<tr>
<td>CCC</td>
<td>Contiguous column coherent biclustering</td>
</tr>
<tr>
<td>FABIA</td>
<td>Factor Analysis for Bicluster Acquisition</td>
</tr>
<tr>
<td>MSR</td>
<td>Mean Squared Residue</td>
</tr>
<tr>
<td>SMSR</td>
<td>Scaling Mean Squared Residue</td>
</tr>
<tr>
<td>GO</td>
<td>Gene Ontology</td>
</tr>
<tr>
<td>KEGG</td>
<td>Kyoto Encyclopedia of Genes and Genomes</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

Patterns are everywhere and human nature seeks to detect them. A millennia ago, finding and understanding the seasonal development of crops led to the development of agriculture. The discovery of patterns in the winds, ocean currents, and skies lead to trans-oceanic navigation and a global world.

In the machine learning domain, the task of obtaining and understanding patterns falls into the context of unsupervised learning, in particular into clustering, which uses the similarity between instances to partition a set of objects into groups [144]. This approach is somewhat too rigid since it looks for similarities considering all subjects’ features and does not allow a subject to belong to multiple groups [126]. Algorithms that perform clustering in both row and column dimensions allow overcoming this limitation. This simultaneous clustering is commonly known as Biclustering [84].

The Cheng and Church algorithm [31], applied to gene expression data, leads to dozens of biclustering algorithms and several computational frameworks [143]. However, we do believe biclustering is still far from reaching its full potential.

Biclustering was tested in the past in biomedicine, text mining, and marketing analysis. However, its main application field is still the traditional gene expression context [26, 84]. This gene expression context generates a bias in the development and application of biclustering with two main consequences. First, biclustering algorithms are developed considering the gene expression context, which acts as a benchmark during development and the independent comparative studies. Typically, the algorithms are not compared in a context other than this application domain. Second, results on real datasets are usually compared using measures of biological relevance such as the Gene Ontology (GO) annotations [88], which are specific to the gene expression data context, and therefore not useful for any other context. We do believe that for biclustering to achieve its full potential, it must expand beyond its gene expression context cocoon.

In 1854 the English physician John Snow conducted one of the earliest records of rudimentary clustering. During a cholera outbreak in London, Snow mapped the cases in a bi-dimensional spatial grid, discovering a relationship between cholera cases’ spatial location and the presence of a public water pump. The discovery of these relationships in a bi-dimensional spatial context influenced public health in the
biggest city on earth at the time. Today the number, volume, and resolution of Spatio-temporal datasets require specific data mining methods. The Spatio-temporal characteristics exist in several domains such as epidemiology, transportation, climate science, social sciences, criminology, and neurosciences [18, 133].

Simultaneous clustering was used before in Spatio-temporal domains such as transportation networks, neurosciences, and earth sciences [11, 25, 45, 55, 72, 89, 119, 140, 141]. The case of neurosciences is particularly interesting since multiple distant regions can fire together, while two nearby regions can exhibit different signals. Additionally, these signals and correlations are not static, and the brain exhibits a dynamic structure [17, 106]. These successful applications in Spatio-temporal data and the presence of bicluster-like structures in fMRI time series motivate the use of biclustering on this type of data. The objective of this thesis is to explore biclustering’s capacity to extract knowledge from fMRI time series.

First, this thesis evaluates the feasibility of biclustering algorithms to study the fMRI signal, considering both synthetic and real-world fMRI datasets. To our knowledge, biclustering algorithms were never compared in a real-world biomedical context other than gene expression data. In the absence of ground truth to compare bicluster solutions with a reference one, we opted for internal valuation metrics. The proposed methodology allows to evaluate biclustering in a group of application areas other than the traditional gene expression context. Sharing fundamental properties with neurosciences, areas such as climate science, epidemiology, and sociology are scientific fields that are expected to benefit from this study.

Second, we focus on the interpretation of biclustering solutions. The task of extracting knowledge from biclustering solutions is still fuzzy since most research is focused on the development of new algorithms and less on their application [143]. To improve biclustering interpretability, we propose a new biclustering evaluation approach. We explore its capacities in modeling the brain network, comparing biclustering algorithms, exploring triclustering, and conducting biclustering based classification.

1.1 Thesis Outline

This thesis is organized as follows: Chapter 2 Background introduces theoretical used concepts. In Chapter 3 Related work we introduce the related work that motivates this thesis. Chapter 4 Biclustering fMRI Time Series: A Comparative Approach introduces a methodology to compare the performance of biclustering algorithms outside the traditional gene expression context. Chapter 5 Improving Biclustering Interpretability: A Pattern Mining Approach focus on the task of extracting useful knowledge from biclustering results. For this, we suggest a new approach to analyze biclustering and provide practical applications. Finally, Chapter 6 Conclusions and Future Work highlights the main contributions of this thesis and highlights the principal research paths derived from it.
Chapter 2

Background

The main objective of this chapter introduce crucial theoretical background directly used later on. Section 2.1 begins by introducing the application domain, the time series generated during fMRI, a popular neuroimaging technique. For this work, we are calling these time series fMRI data. This section explains their general properties and how do they fit in the Spatio-temporal data family. Then we enter the data mining domain. Most of the thesis work is related to unsupervised learning techniques. Section 2.2 introduces a popular form of it: clustering. The clustering section has a double objective. First, it should explain the basic concepts of the task of grouping similar objects. Second, it establishes concepts for later chapters. Bicustering, the main focus of the thesis, is explained in section 2.3. In particular, we focus on current approaches to compute these structures, possible interesting patterns present in data, and bicluster evaluation. Additionally, this section covers the seven state-of-the-art biclustering algorithms used in this work.

Despite not being the main focus of this thesis, triclustering is explained in section 2.4 to generalize the previous concepts. Section 2.5 focus on pattern mining, a particular form of unsupervised analysis that focus on finding associations and patterns. This section is introduced, not only due to his close relation with Biclustering, but also because later it will be used to offer visual representations for patterns found in biclusterings. For the same reasons, we introduce in section 2.6 the basics of graph and hypergraph theory, which is a useful mathematical structures to study similarities among objects. Finally, section 2.7 covers the basic concepts of supervised learning, briefly explored in the later section of this thesis.

2.1 Capturing neuronal activity: fMRI

Located inside the skull, the brain acts as the central organ of the human nervous system. This organ contains a net consisted of around 86.1 billion highly interconnected cells (neurons), responsible for most of the computations needed for the human body to function appropriately [12, 19]. Consisting of
highly specialized regions, it is of particular scientific interest to understand how different brain regions interact and how they react to external stimulus [12].

Addressing the human brain’s general temporal behavior and the nature of brain areas’ connections is not a straightforward task. Popular neuroimaging techniques are Electroencephalography (EEG), Magnetoencephalography (MEG), Positron Electron Topography (PET) and Functional Magnetic Resonance Imaging (fMRI) [12]. fMRI data have a great spatial resolution while maintaining a reasonable temporal resolution, so they have emerged as an ideal technology to study patterns and relationships in the whole-brain activity. Additionally, in contrast to PET, fMRI does not rely on the use of any ionizing radiation (is a non-invasive method), allowing to repeat studies as many times as needed [17, 20, 80, 81].

2.1.1 fMRI as part of the Spatio-temporal data family

fMRI, as a neuroimaging method, allows to locate brain activity on second-by-second bases and within millimeters of its origin [17, 81]. These temporal and spatial properties place this type of data into the Spatio-temporal data family, sharing properties with areas as diverse as earth sciences, epidemiology, and sociology, which regularly deal with space and time concepts and their properties [17, 18].

The study of Spatio-temporal data (ST data) is not a new domain in science, and its fundamental differences over other types of data lead to multiple surveys written over the years [18, 118, 121]. Considering the diversity of ST data, a well-defined group of properties is not well defined. Therefore, different authors specify different general properties [18, 75]. Table 2.1 highlight some generic Spatio-temporal properties that are also true for the specific case of fMRI data.

<table>
<thead>
<tr>
<th>Data characteristic</th>
<th>Description</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>Autocorrelation effect</td>
<td>Dependencies among temporal and spatial dimensions</td>
<td>[18, 118]</td>
</tr>
<tr>
<td>Data Heterogeneity</td>
<td>Different regions and temporal periods may have different distributions in varying levels</td>
<td></td>
</tr>
<tr>
<td>Continuous data</td>
<td>Challenge since many classical data mining techniques assume discrete data</td>
<td></td>
</tr>
</tbody>
</table>

Table 2.1: General spatio-temporal properties

Different works on the topic provide different categorizations for these data. In this work, we follow the approach given by Atluri et al. [18], where definitions of Data Type and Data Instance are provided. Differences between these definitions establish the link between the fundamental phenomena under study and the data mining approach applied to extract knowledge from collected data. While the first is a classification that measures how data is collected and represented, the second is the basic unit that a data mining algorithm operates upon.
An fMRI scan measures a sequence of 3D images over time. These scans measure the mean value inside volume elements (voxel) that partition the brain into equally sized boxes [81]. Adding the temporal dimension, we obtain measurements over a 4D grid. This type of measurement falls into the Spatio-temporal raster data type. A fixed measuring device collects the observations in this type of data. Diverse scientific domains use this data type. Figure 2.1 illustrates this concept considering both an MRI brain scan and weather stations’ measurements.

![Figure 2.1](image)

(a) Dutch meteorological stations  
(b) Brain slice obtained by a MRI machine

Figure 2.1: Two different application cases using the same base data type. On the left, a spatial map representing Dutch meteorological stations. These stations are spatially fixed while doing their measurement periodically (Figure adapted from [141]). On the right, the MRI machine obtains a regular grid of observations of the (ideally) immobile brain (Figure adapted from [103]). Despite the difference in the application domain, the observations are done under the same observational principles.

After collecting data, they must then be converted into a format to be analyzed. The ST raster type is traditionally converted into three possible analysis scenarios: spatial maps, ST raster and time series. The differences between each possible instance are related to the relative importance of the spatial and temporal context.

The **spatial map** data format interprets data as a collection of spatial figures under temporal stamps. Tasks such as detecting brain activity and diagnose diseases apply deep learning on this data representation [134]. The **ST raster** data format uses the whole measurements overall dimensions and all temporal points under analysis. For fMRI data, this means representing data using 4-way arrays and then extracting features from it. Finally, we got the **time series** analysis that discards the data’s spatial information and treats each spatial point as an independent time-series. This simplification is strongly motivated...
by computational and conceptual challenges associated with the formulation of Spatio-temporal models. Additionally, it opens up the literature related to time series analysis [17, 106].

2.1.2 Temporal response to stimulus

The temporal dynamics of human brain activity are captured by fMRI using an indirect measure, based on blood oxygenation. This measure is called the blood oxygenation level-dependent (BOLD) signal. It will have different formats considering both the nature of the associated stimulus and the intrinsic dynamics of blood flow (hemodynamics) [81].

Figure 2.2 represents the standard model to measure this signal, referred to as the hemodynamic response function (HRF). Following an initial stimulus, the BOLD signal does not change for 1-2 seconds. This initial dip is observed empirically and can indicate some oxygen extraction before a later overcompensatory response. The BOLD activity peak happens 5-8 seconds after the stimulus, and the whole event happens with a total duration of 10-20 seconds [9, 67, 81].

This response function assumes a single instantaneous stimulus. This is unrealistic, not only a stimulus will last some time but also multiple stimulus can happen in a sequence [81]. Therefore, the BOLD signal $b(t)$ is modeled using a convolution between the stimulus function $s(t)$ and the hemodynamic function $h(t)$: $b(t) = (s * h)(t)$. Figure 2.3 illustrates the BOLD signal corresponding to two different types of stimulus patterns leading to different features.

![Figure 2.2: The Standard canonical model used to model the Hemodynamic response function (HRF).](image) Neuronal activity will cause an increase in the inflow of oxygenated blood to active regions of the brain (Figure adapted from [67]).

![Figure 2.3](image)
Chapter 2 Background

2.2 Clustering

Clustering is a family of unsupervised learning algorithms whose objective is to group data into a discrete set of hidden data structures called clusters [57, 144]. More formally, given a dataset consisted of $n$ observations (rows) $X = \{x_1, ..., x_n\}$ and $m$ attributes (columns) $Y = \{y_1, ..., y_m\}$, the objective of clustering is to find a subsets of the observations (clusters) that follow some similarity criteria over the attributes [61]. This approach generates what are called a clusters of rows. With minor adaptations in the data format, it is also possible to use clustering to obtain clusters of columns [84].
Given the large quantity of clustering algorithms in the literature, it is usual to classify them under some criteria. Jiawei and Micheline [57] pointed four non-exclusive categories for clustering algorithms: **Partitioning Methods**, **Hierarchical Methods**, **Density-based Method** and **Grid-based Methods**, which are resumed in Table 2.2.

<table>
<thead>
<tr>
<th>Category</th>
<th>General Characteristics</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Partitioning methods</td>
<td>Uses metrics of distance in the attributes to partition the observations into a predefined number groups</td>
<td>K-means</td>
</tr>
<tr>
<td>Hierarchical methods</td>
<td>Successively merges or breaks clusters until an ideal number of clusters is found</td>
<td>Ward’s method</td>
</tr>
<tr>
<td>Density-based methods</td>
<td>Model clusters as dense regions in the data space allowing to obtain arbitrary shape clusters</td>
<td>Spectral</td>
</tr>
<tr>
<td>Grid-based methods</td>
<td>Partition the data space as a grid, independently of the observations</td>
<td>STING</td>
</tr>
</tbody>
</table>

Table 2.2: Clustering categories (Table adapted from [57]).

Typically, a clustering algorithm generates a set of cluster structures. Evaluating a set of clusters is not a straightforward task. There are two categories of quality evaluation methods: **extrinsic** and **intrinsic**. While the first category uses an ideal clustering (ground truth) that must be known, the second evaluates the clusters by measuring how well they clusters are separated and how compact they are [57].
Limitations

One of the most critical limitations regarding clustering is related to the number of generated clusters, which is often pre-defined. While there are approaches that try to evaluate the correct number of clusters, the lack of ground truth means that no answer is entirely correct. While a high number of clusters means that each one of them will be more homogeneous, they will have fewer points in each one of them, increasing the chances of modeling noise. A low number of clusters will mean heterogeneous ones. Additionally, clustering algorithms force all points to belong to a cluster, not recognizing the meaning of noise. One of the most important limitations is that, despite some exceptions, most clustering algorithms organize the observations into disjoint groups.

Additionally, clustering is a rigid approach since they force the cluster to be homogeneous under all attributes. Some subjects could relate to each other on a subset of attributes, requiring less rigid approaches to detect these similarities.

2.3 Biclustering

Traditional clustering generates groups of data observations considering all its attributes. While this is great in a plenitude of contexts, it assumes a restricted set of conditions that are not generally true. In the context of gene expression data, the need to find genes with similar activity under a specific group of experimental conditions leads to the development of biclustering algorithms that perform simultaneous clustering of rows and columns [84].

Biclustering is, therefore, a generalization of the traditional clustering since it identifies sub-matrices of the original data matrix rather than disjoint stripes of rows or columns. In the limit, a cluster can be seen as a particular case of a bicluster. An algorithm able to find this type of sub-matrices is called a biclustering algorithm.

While some biclustering algorithms generate only one bicluster at a time, others allow discovering a set of biclusters. When this is the case, the algorithms find a biclustering solution, or merely a biclustering. While a singular bicluster should illustrate a well distinct pattern, the collection of biclusters generated by an algorithm can be rich depending on the associated search strategy.


2.3.1 Bicluster Patterns

Biclustering is able to find more flexible structures than traditional clustering. While the nomenclature for this possible structures is not standardized in the literature, the definitions provided by Madeira and Oliveira [84] and Aguilar-Ruiz [8] are commonly used. The most simple structure is the constant bicluster which is a submatrix (I,J) where all values $b_{ij}$ are equal: $b_{ij} = \pi$.

In the context of a real-valued matrix, it is of particular interest to discover coherent biclusters. A coherent bicluster can be described done using row parameter $\pi_i$ and a column one $\beta_j$. Their interaction between the two parameters can be either additive (shifting) or multiplicative (scaling). A perfect shifting bicluster follows the additive relation among the variable: $b_{ij} = \pi_i + \beta_j$. Similiarly, a scaling bicluster follows a multiplicative relation: $b_{ij} = \pi_i \times \beta_j$. Figure 2.6 illustrates the visual differences between the two patterns.
2.3.2 Search strategies and biclustering algorithms

When categorizing biclustering algorithms, a particular concern is related to the search strategies. This section aims to explain four main search strategies: greedy, divide and conquer, exhaustive, and distribution parameter identification. These categories, defined by Madeira and Oliveira [84], are used in recent comparison studies, such as Padiha et al. [100] and Henriques et al. [62] and are explained in Table 2.3.
A high number of biclustering methods were developed during the past two decades [26, 62, 84, 105, 126, 143]. The objective of this section is not to review every single algorithm. Instead, seven state-of-the-art biclusters were selected for our analysis, whose general description follows.

**Bimax**

Bimax works with binary datasets. Its objective is to find sequences of *ones* across features and conditions. To achieve it, it recursively divides the data matrix in order to locate these submatrices of ones. Figure 2.7 illustrates the splitting process.

<table>
<thead>
<tr>
<th>Category</th>
<th>General Characteristics</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Greedy</td>
<td>Biclusters are generated by adding or removing columns to a initial random bicluster in order to improve some gain function. The final objective is that, after some iterations the algorithm leads to a global minimum solution. Despite making wrong decisions, and loosing good biclusters due to being stuck in local minima, they have the potential of being fast algorithms.</td>
<td>Isa</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Xmotifs</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[21, 98]</td>
</tr>
<tr>
<td>Distribution</td>
<td>Assume some statistical model behind the data, and then apply some iterative procedure in order to obtain its parameters by minimizing some criterion.</td>
<td>Fabia</td>
</tr>
<tr>
<td>Identification</td>
<td></td>
<td>Spectral</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[64, 79]</td>
</tr>
<tr>
<td>Divide</td>
<td>Divide the original data matrix into smaller instances. With the potential of being very fast, they could fail to find good biclusters, splitted before identified.</td>
<td>Bimax</td>
</tr>
<tr>
<td>and conquer</td>
<td></td>
<td>[107]</td>
</tr>
<tr>
<td>Exhaustive</td>
<td>Based on the premise that finding the best biclusters can only be done by using an exhaustive enumeration of all possible biclusters in the data matrix. Despite being able to find the best biclusters they do it by imposing restrictions to the biclusters size (since these algorithms are typically very slow).</td>
<td>Bicpam</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CCC</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[60, 85]</td>
</tr>
</tbody>
</table>

Table 2.3: Biclustering algorithms search strategy categories.
Figure 2.7: Illustration of the bimax algorithm. The original matrix is fragmented into three sub-matrices. One has only zeros and was removed. The algorithm then repeats the division for the two remaining matrices (Figure adapted from [107]).

Since the algorithm works only using binary data, its capacity to locate complex patterns will always be limited. Additionally, the data binarization method will significantly influence the algorithm performance [42]. Despite its theoretical limitations, it was shown to achieve similar performance scores to more complex algorithms, with the advantage of being fast [107].

**FABIA**

As a distribution parameter identification algorithm, FABIA assumes a multiplicative generative model. A single bicluster is resultant from the product of two space vectors $\lambda$ and $z$. Figure 2.8 illustrates how one bicluster is obtained from this product.

![Diagram of FABIA](image)

Figure 2.8: The outer product of the two vectors results in a matrix with a bicluster. The sum of the multiple biclusters allow to retrieve the original data matrix, plus some noise (Figure adapted from [64]).
The original dataset $B$ is then assumed to be the sum of $p$ biclusters, plus some noise $\Upsilon$.

$$B = \sum_{i=1}^{p} \lambda_i z_i^T + \Upsilon = \Lambda Z + \Upsilon$$  \hspace{1cm} (2.1)

To fit this model to the dataset, FABIA relies on conducting factor analysis on the dataset columns. To turn the factor analysis results into biclusters, the model relies on thresholds.

**Spectral**

Spectral biclustering relies on singular value decomposition (SVD) to decompose the original data matrix into a checkerboard pattern in the data. Figure 2.9 illustrates the idea.

![Spectral Biclustering](image)

**Figure 2.9: Objective of spectral biclustering. (Figure adapted from[79]).**

After a normalization step, the spectral biclustering describes the data matrix as a row and column eigenvectors product. The biclustering task is turned into an eigenproblem. After obtaining the pairs or row and column eigenvectors, the algorithm filters the relevant cells considering a threshold.

**ISA**

The Iterative Signature Algorithm describes the dataset as a product between row and column eigenvectors (similar to spectral biclustering). However, it does not apply SVD to the data. ISA is a greedy algorithm, meaning that it starts with a seed bicluster consisted of random rows. Then it iteratively adds and removes rows and columns of the bicluster to reduce its standard deviation [21].

**XMotifs**

XMotifs is also a greedy algorithm, seeking biclusters with conserved rows in discretized datasets. The objective of XMotifs is to find the largest bicluster in the dataset. The algorithm begins by randomly selecting some columns. For each one, XMotifs will find rows with the same symbols over the columns. It does it iteratively to find the largest bicluster. After detecting a bicluster, they remove its rows from the dataset, meaning that the algorithm cannot identify row overlapping biclusters [98].
**BicPAM**

BicPAM is an exhaustive algorithm with the flexibility of detecting multiple types of patterns. It is based on pattern mining, and its approach consists of three steps: **mapping, mining** and **closing**.

1. **Mapping** conducts several preprocessing steps. This step is particularly important to handle real-valued data.

2. **Mining** or pattern discovery is the core of BicPAM, where several pattern miners will search for the biclusters given some particular type of pattern.

3. **Closing** uses several techniques to improve the quality of the found solutions.

These main steps are state of the art in PM-based biclustering. However, BicPAM contributed to this area with new strategies. Figure 2.10 highlights the general methodology, as well as the new proposed BicPAM’s contributions.

![BicPAM methodology](image)

**Figure 2.10**: BicPAM methodology. BicPAM improved PM-based biclustering with several strategies to improve the general biclustering quality (Figure adapted from [60]).

**CCC**

CCC, contiguous column coherent biclustering, is an exhaustive algorithm that considers temporal contiguity restraints. It can only find biclusters in contiguous columns. Therefore, this algorithm is well suited to work with temporal data. The objective is to find all maximal contiguous column coherent biclusters using a discretized version of the original matrix. The algorithm introduces an alphabet transformation to the data and constructs a suffix tree from which the biclusters are extracted [85]. Figure 2.11 illustrates the alphabet transformation and the constructed suffix tree.
2.3.3 Evaluating Biclusters

There are multiple ways to evaluate a biclustering solution and Horta and Campello [65] state four approaches commonly used to evaluate biclusters. Biological interpretation, based on visualization methods and previous knowledge about the topic. Other approaches must accompany this topic due to its subjective nature. It is impractical when several algorithms are compared. Biological significance is used in order to compare solutions with their biological significance. This possibility is valid when considering the gene expression application context. External evaluation can be used when ground truth is known, allowing to compare biclustering solutions with reference. This evaluation method is a popular choice to evaluate biclustering algorithms on synthetic datasets. Finally, internal evaluation, considers only information inherent to the data to evaluate the solutions.

For our study, of particular interest are the approaches based on internal evaluation indexes that translate the bicluster’s internal quality into a single number. However, since there are multiple types of patterns that fall into the definition of bicluster, there is no single metric that can capture all types of patterns. As stated by Hartigan [58], one simple hypothesis to evaluate a bicluster is to calculate its Variance:

\[
VAR(B) = \frac{1}{|I| \times |J|} \sum_{i=1}^{\mid I \mid} \sum_{j=1}^{\mid J \mid} (b_{ij} - b_{IJ})^2. \tag{2.2}
\]

In this equation, \(b_{ij}\) refers to the element \((i,j)\) of the data matrix \(B\), \(b_{IJ}\) refers to the global mean of
the elements belonging to the data matrix, and $|I|, |J|$ represent the total number of rows and columns. A disadvantage of using the variance is that it can only measure constant biclusters. Therefore, it will fail to detect most of the more complex patterns we want to detect.

Introduced by Cheng and Church [31], the Mean Squared Residue (MSR) aims to find the coherence of the bicluster over rows and columns:

$$MSR(B) = \frac{1}{|I| \times |J|} \sum_{i=1}^{|I|} \sum_{j=1}^{|J|} (b_{ij} - b_{iJ} - b_{IJ} + b_{IJ})^2,$$

(2.3)

Where $B_{iJ}$ and $B_{IJ}$ point to the row $i$ mean and the column $j$ mean. This approach is an improvement comparing to the variance since it can capture shifting tendencies. However, as pointed by Bozdag et al. [23], it is unable to capture scaling patterns.

In order to capture the scaling tendencies that the MSR cannot recognize, Mukhopadhyay et al. [95] developed a new measure called Scaling Mean Squared Residue (SMSR). This comes with the price of being unable to capture the shifting patterns.

$$SMSR(B) = \frac{1}{|I| \times |J|} \sum_{i=1}^{|I|} \sum_{j=1}^{|J|} \frac{(b_{ij} \times b_{iJ} - b_{ij} \times b_{IJ})^2}{b_{iJ}^2 \times b_{IJ}^2}.$$

(2.4)

An important property of biclusters is the possibility to have a different range of values between each other, suggesting the use of standardisation processes. As pointed by Pontes et al. [104], an advantage of this approach is to characterise their tendency. One way of doing it is to standardize data by row. In the context of fMRI, this means normalizing the behaviour or every brain region to have a similar amplitude as follows:

$$\hat{b}_{ij} = \frac{b_{ij} - \mu_{gi}}{\sigma_{gi}}, 1 \leq i \leq |I|, 1 \leq j \leq |J|,$$

(2.5)

where $\mu_{gi}$ and $\sigma_{gi}$ represent the mean and standard deviation over the rows of the data matrix (regions of the brain). Based on this approach, a quality measure was proposed under the name of Virtual Error (VE) [8, 37], considering the concept of a virtual pattern, which can be defined over rows or columns [104]. The pattern over rows is given by:

$$\hat{\rho}_j = \frac{1}{|I|} \sum_{i=1}^{|I|} \hat{b}_{ij}.$$

(2.6)

This pattern is an average behavior over time, and the virtual error is thus defined as a difference between the row values and the virtual row:

$$VE(B) = \frac{1}{|I| \times |J|} \sum_{i=1}^{|I|} \sum_{j=1}^{|J|} |\hat{b}_{ij} - \hat{\rho}_j|.$$

(2.7)
A particular advantage of the Virtual Error measure is its capacity to detect both shifting and scaling patterns [104]. It is thus a suitable internal measure to compare different possible patterns obtained by different biclustering methods. When considering the presence of noise, Pontes et al. [104] pointed out that the virtual error can capture both scaling and shifting patterns since its value varies linearly with the induced error.

### 2.4 Triclustering

While clusters are strips on the original dataset, and biclusters are sub-matrices of the original matrix, triclustering generalizes this concepts, considering the third dimension of analysis. In triclustering, instead of dealing with a traditional matrix, we work with a tensorial structure, as seen in Figure 2.12. One possibility of generating this tensorial structure from a bidimensional dataset is to use one variable as a stratifying variable and re-organize it based on its possible categories.

![Figure 2.12: Traditional machine learning datasets and datasets used in the triclustering task (adapted from [61]).](image)

Just as biclustering’s objective is to obtain sub-matrices of the original dataset, the objective of triclustering is to obtain subspaces (triclusters) of the original tensor, such that each tricluster satisfies some specific criteria of homogeneity and statistical significance [61].
Chapter 2 Background

2.5 Pattern mining

The objective of pattern mining as an unsupervised learning family of algorithms, is to extract substructures from a dataset. It is traditional to work with transactional datasets and search for statistically relevant relations among items [57]. In pattern mining, the basic unit is an item $I_i$, and algorithms operate on sets of items, $T_k$, also called transactions. The general objective of pattern mining is to detect relations among transactions based on the items inside them. These relations among transactions must be statistically significant. It is usual to use rule interest metrics such as the support. The support of an item $I_i$ can be defined as the probability of these items appear in a transaction:

$$\text{Support}(I_i) = P(I_i). \quad (2.8)$$

The support definition can also be applied, not to a single item but also a transaction. In this case, it is usually defined as the probability of all items in the transaction to be together. For example, given a transaction made of two items: $R_k = I_i, I_j$, its support is given by:

$$\text{Support}(I_i, I_j) = P(I_i \cup I_j). \quad (2.9)$$

This definition allows to quantify the relation among the itemsets $I_i$ and $I_j$, and is symmetric, which means that $\text{Support}(I_i, I_j) = \text{Support}(I_j, I_i)$. Not every metric verifies this condition.

Another metric is the confidence, that establishes a relation between the two items, and computes the probability of the consequent given the presence of the antecedent:

$$\text{Confidence}(I_i, I_j) = \frac{P(I_i \cup I_j)}{P(I_i)}. \quad (2.10)$$

In the general case, $\text{Confidence}(I_i, I_j) \neq \text{Confidence}(I_j, I_i)$ since the probability of both items is not equal in the general case. Therefore, this measure establishes a directed relation between the two items compared to the support measure.

2.6 Graph and Hypergraph theory

In its most simplistic definition, a graph is a set of vertices connected by edges and is frequently used to model relations between objects [131]. Examples of graphs are shown in figure 2.13.

When modeling, it is usual to represent entities as the graphs’ vertices while the edges represent a measure of connection between them. Notice that these relations are not necessarily symmetric. In contrast to simple graphs (figure 2.13a). Directed graphs establish a direction to the vertices (figure 2.13b). This means that, in general $(a, b) \neq (b, a)$. 
Figure 2.13: Examples of undirected and directed graphs. While undirected graphs have simmetrical relations among variables, directed graphs have assymetric relations leading to the existence of two different edges in two nodes (figures adapted from [1]).

Graphs are frequently used to model relations and are frequently used in scientific domains such as physics, biology, and computer sciences [24, 44, 56, 78, 112, 116, 131]. However, graph theory is limited to model only connections between two entities.

Hypergraph theory acts as a generalization of traditional graph theory, allowing to model connections between more than two entities. In it, entities are still represented as vertices. However, the connections now can link more than two vertices (hyperedges). Figure 2.14 represents a hypergraph. Traditional hypergraph theory considers only undirected hypergraphs. Outside traditional hypergraph theory, dirhypergraphs act as a generalization to directed graphs [24].

Figure 2.14: An hypergraph. In contrast to traditional graph, hypergraphs establish connections with more than 2 nodes (Figure adapted from [131]).
2.7 Supervised Learning: Classification

Unsupervised Learning has the name from the lack of labels on data. The objective of unsupervised learning is to find unknown groups. In supervised learning, a label is present for the data, and the objective is, given new unlabeled data, to predict its label. In this thesis, classification is used, where the label has a categoric symbol \([57]\). Classifiers are the responsible methods for learning from the data and predicting classes. Figure 2.15 presents an illustrative view of the classification task and a popular classifier.

![Illustrative classification](a) Illustrative classification

![Decision tree classifier](b) Decision tree classifier

Figure 2.15: On the left an illustrative example of the classification task. Given information about what we know are dogs and cats, predict if a new animal is a dog or a cat. On the right, a visualization of decision trees, an example of a classifier. This type of classifier uses rules learnt from data to distinguish between labels.
Chapter 3

Related Work

This chapter has two major objectives:

1. Introduce the related work that fundament our approaches.

2. Mix previous concepts to motivate our analysis.

The purpose of this chapter is to provide practical topics that fundament the approach developed during this thesis. This section begins by categorizing previous biclustering reviews, which are crucial documents to study for any use of biclustering. Then, it explains traditional steps for fMRI time series analysis. After that, we focus on previous uses of Biclustering and Triclustering on Spatio-temporal data family, including neurosciences. Then, we review the use of techniques based on biclustering for classification. Finally, we justifying why we hypothesize that fMRI data is an adequate application domain for biclustering.

3.1 Biclustering reviews

During the last two decades several authors reviewed biclustering methods [23, 26, 42, 62, 65, 84, 100, 104, 105, 107, 115, 126, 143]. These reviews can be fitted into three main categories, and all focus gene expression data analysis: 1) General Surveys [26, 62, 84, 105, 126, 143], 2) Comparison Surveys [23, 42, 62, 100, 107] and 3) Measure Reviews [65, 104, 115].

On the general survey category, we have papers that reunite possible multiple views on biclustering into a single perspective. They do it by providing abstract categorizations for the algorithms and associated solutions and reviewing state-of-the-art approaches and typical application cases. These papers are beneficial for the first look on biclustering to understand its concepts and guide research. However, they usually lack a guideline for selecting one method over the other by reviewing multiple cases.

Madeira et al. [84] presented a structured view of biclustering, providing a mathematical formulation on the problem, classifying the biclustering algorithms along four dimensions, and listing diverse
application cases. Tanay et al. [126] reviewed six state-of-the-art biclustering methods. Busygin et al. [26] provided a new mathematical definition of biclustering, as well as a review of 15 biclustering algorithms and application cases. Henriques et al. [62] proposed a structured view on a class of biclustering algorithms based on pattern mining providing a categorization specific to these algorithms. Pontes et al. [105] analyzed 47 biclustering algorithms considering the patterns they can find. Xie et al. [143] revised the possible biclustering applications in biological and biomedical data analysis. They further provided a workflow to illustrate the research path from original data to biclustering methods and final results.

**Comparison surveys**, approach the task of finding the “best” biclustering algorithm. Since dozens of biclustering methods were developed, evaluating them all under appropriate circumstances is not practicable; thus, the traditional approach is to select some state-of-the-art methods and compare results. As the development of better biclustering algorithms follows, old comparisons tend to lose importance. To our knowledge, Prelić et al. [107] was the first to compare multiple biclustering algorithms by adapting methodologies used in clustering. They chose five prominent biclustering methods, proposed a new algorithm, and compared their performance using synthetic and real gene expression data. The validation was performed using external indices on synthetic data (ground truth was known) and the biological relevance (based on Gene Ontology (GO) annotations and metabolic and protein interaction networks). Bozdağ et al. [23] compared 5 biclustering algorithms following an approach similar to Prelić et al. [107], but considering the effects of noise, size and bicluster overlap. Eren et al. [42] expanded the research by comparing 12 algorithms, using synthetic datasets following six different biclustering models and eight gene expression datasets. An external measure was used with synthetic data, while biological relevance was used for real-world data. Henriques et al. [62] compared the performance of pattern-based biclustering solutions to the traditional approaches, evaluating a total of 15 state-of-the-art approaches. Their evaluation was performed in synthetic and real datasets, considering two external measures and computational efficiency and biological relevance of results. Padilha et al. [100] used three synthetic data collections and two real data collections to analyze the performance of 17 algorithms. Two new external validation metrics were used on synthetic data, and real-world data was compared using two biological significance metrics (GO and KEGG).

Finally, **measure reviews** discuss possible metrics to evaluate biclustering methods. Of particular interest is the work developed by Padilha et al. [65], where the task of evaluating biclustering is discussed, considering properties of an external biclustering measure and listing possible measures to use. Santamaría et al. [115] reviewed some internal and external indices. Pontes et al. [104] focused on internal measures, placing them into three possible categories, and discussing their capability of detecting different types of biclusters.


3.2 fMRI data Analysis

To understand the dynamics of the brain, diverse neuroimaging techniques were developed to understand how different brain regions interact with each other and how they react to some external stimulus [17]. fMRI measures changes in oxygenation of the blood flow associated with the neuronal activity (BOLD) [67] and generates a 4-dimensional figure with a temporal resolution of a few seconds. This is an adequate resolution since typical events have a temporal amplitude of 10-20 seconds in terms of hemodynamic response [9, 67, 81].

The focus of this section is in explaining the processing of fMRI data.

- First, we explain the process of simulating fMRI data and their importance,
- Second, the traditional preprocessing steps for raw data is pointed, as well as the use of brain atlas to parcellate the brain, reducing the dimensionality of data,
- Finally, we review techniques to analyze fMRI time series.

3.2.1 Simulating fMRI data

Due to their complicated structure, simulating fMRI data is often used as a vital tool to evaluate processing and analyzing methods [43]. There are dozens of simulators developed having a single study in mind. These types of simulators are based on different models, with different assumptions and therefore have very different behavior and performance, with the possible consequence of generating a bias considering the study in mind [43]. Therefore, it is useful to use generic simulators to generate artificial fMRI datasets. In literature, two main categories of computational simulations can be distinguished: 1) Simulating the time series using an experimental design and 2) simulating the magnetic signal by solving the Bloch equations.

Both approaches face step-backs. There is no dominant approach for the first one, meaning that each simulator has a different model. Usual methods model the activation in the time series using a convolution between the hemodynamic response function and a stimulus vector. Different sources of noise can then be added. There are computational drawbacks for the second type of simulators since solving the Bloch equations is computationally expensive [136]. Four simulators are worth commenting.

- In order to solve the Bloch equation, POSSUM [38, 39] was developed as part of the of FSL (FMRIB’s software library)[124];
- To provide a fast tool for simulating fMRI data (allowing its use in large simulation studies), neuRosim [136] was developed as a package for the R computing language;
- For the python language, fmrisim was developed for advanced neuroimaging analyses [40];
Finally, SimTB [43] (developed for the MATLAB environment) was developed by using a full parametrized models and a model of spatio-temporal separability.

3.2.2 Preprocessing fMRI data

Traditional fMRI datasets are available in a specific format and must be preprocessed to be analyzed [41]. In order to preprocess fMRI data there are numerous software available, such as FSL [71, 123, 138], AFNI [33], SPM [13] and BrainIAK [2]. Each software has an associated preprocessing pipeline. Typical preprocessing steps are:

- Skullstripping (removing non-brain elements from figures such as the skull)
- Correcting head motions
- Smoothing the figure
- Normalizing: transforms low-resolution functional images to a high-resolution scan.
- Registering: aligns the high-resolution scan to a standard

After preprocessing, data are ready to be analyzed. However, at this point, data are still very high dimensional time series. One way to reduce the dimension is to parcellate the brain into regions of interest using brain atlas [17, 34, 35]. Atlases can be used to define full-brain parcellations. Multiple brain atlases are available in the literature. Popular choices for brain atlas to be used are the AAL atlas or the Harvard-Oxford one [34]. It is then possible to compute the mean time series of these regions using the voxels time series and find patterns over these reduced data [17].

3.2.3 Clustering: Parceling the brain

The number of fMRI possible experiments are diverse, and with it, the number of possible objectives. Analyzing fMRI consists of understanding how some stimulus could lead to changes in neuronal activity [81]. Doing it involves analyzing a complex signal while dealing with multiple sources of noise. Thus, naturally, the number of tools to analyze fMRI data had grown over the last decades [87, 106].

Considering clustering, the traditional task is to partition the brain regions into groups. Since clustering is a family of techniques, multiple ones have been used in this task. The most popular algorithms are the partitional clustering algorithm k-means, hierarchical clustering approaches, and spectral clustering [52, 87, 128]. Due to computational limitations, performing clustering at the voxel level is not usually considered a good practice, and a traditional approach is to reduce the quantity of data either by using data from a limited number of slices [32] or by using a parcellation scheme [114, 127].
3.2.4 Graph Approaches: Measures of brain connectivity

The human brain consists of a group of spatially distributed regions sharing information. Figure 3.1 represents two possible representations of a brain network.

To understand brain connectivity there are two fundamental aspects that need to be distinguished: functional and effective connectivity [47]. Functional connectivity is defined as the temporal correlations between spatially distant brain regions [5, 48], and is commonly extracted from fMRI, EEG or MEG data. Effective connectivity is defined based on a model applied to data and refers to the direct influence of one region on another [66].

![Connectivity matrix and Equivalent network](image)

Figure 3.1: Networks are commonly represented by their connectivity matrices, with rows and columns representing nodes and matrix entries representing links (Figure adapted from [113]).

A traditional tool to study these networks uses graph theory [87, 130]. The graph typically uses brain regions as nodes and compares the relationship between them as a connection. Binary links between brain regions register connections, while weighted links can store more complex information about connection strengths [113].

The use of structures such as hypergraphs is scarce. Wang and Abugharbieh [132] proposed an approach based on hypergraphs to face the challenge of pronounced noise in neuroimaging data. Additionally, hypergraphs have been used to identify high order brain connectome biomarkers for disease prediction [146], and study relationships between functional and structural connectome data [97].

Agrawal et al. [6] introduced the concept of tripole in the concept of time series analysis. This structure results from a non-symmetric combination of three nodes. These structures simplify a directed hypergraph and were applied to real-world datasets from climate science and neurosciences.
3.2.5 Brain connectivity diseases and the case of schizophrenia

A group of studies has been exploring fMRI techniques to study neurologic and psychiatric brain disorders, suggesting that neurodegenerative diseases target cortical networks rather than single regions [117]. This approaches in diseases such as Alzheimer’s [53, 110], depression [54], dementia [111], multiple sclerosis [83], amyotrophic lateral sclerosis [94] and schizophrenia [16, 22, 50, 82, 137].

In general, the goal of biomedical research is to establish clinical biomarkers. These biomarkers are a set of characteristics (features) that allows early disease detection and prognostic prediction [3, 14, 30, 99, 135]. They are usually complex, computed from the raw features obtained from neuroimaging methods.

Schizophrenia is a psychiatric disease characterized by delusions and hallucinations, loss of emotion, and disrupted thinking. Schizophrenia is a particular case since it was stated to be a disconnection disease since it was defined at the beginning of the twentieth-century [4]. In recent years, the efforts to map the neural elements and connections provided significant benefits to the study of connectivity abnormalities in schizophrenia [46]. Figure 3.2 illustrates this concept. Despite the progress, the regions showing disconnectivity in this disorder are inconsistent across studies [102].

One challenge in these studies is constructing a brain network, usually relying on Brain Atlas to define the nodes. Other is the large number of possible connections, even for a small number of regions (a graph with n nodes has $\binom{n}{2}$ connections). This large number of nodes, added to low signal-to-noise ratio, can prevent the discovery of true associations [16].

![Connectivity matrix for schizophrenic patient](image1.png)

![Connectivity matrix for control patient](image2.png)

Figure 3.2: Connectivity matrices for a schizophrenic patient and a control. While the densities are equivalent, the minimum and mean weight for the patient is lower than for the control. This points that patient’s connectivity matrix will contain more low-value when compared to a healthy patient (Figure adapted from [46]).

A popular assumption is that the discovered biomarkers are valid across all the study subjects. This
assumption is not always valid. **Different patients tend to have different factors due to disease and population heterogeneity.** Biomarkers based on pattern mining can capture these subspace scenarios by exploring differences between subsets of subjects instead of all subjects. Figure 3.3 illustrates this concept [14].

![Diagram](https://via.placeholder.com/150)

Figure 3.3: In the data matrix, a colored element indicates that the feature is present for a given subject. A, B, C, and D are submatrices in X that allow to distinguish between healthy and schizophrenic patients (Figure adapted from [14]).

### 3.3 Biclustering, Triclustering, and Spatio-temporal data

One of the traditional types of analysis conducted on ST data is related to clustering with two main objectives: 1) Finding clusters with an unusually high density of points (*hot-spots*) and 2) detecting clusters of ST points with similar non-ST attributes [18]. In more recent years, more general approaches than clustering have been successfully used in distinct Spatio-temporal application domains. This section focus on the uses of biclustering and triclustering (generalizations to clustering) to analyze data with Spatio-temporal properties.

Considering **biclustering**, Wu et al. [142] analysed Chinese meteorological Spatio-temporal data. Shen et al. [119] used biclustering to analyze the global distribution of natural disasters. Kaban et al. [72] used biclustering to identify spatial of social vulnerability in Indonesia and was able to distinguish profiles of social vulnerability. Borgnat et al. [49] to identify spatial and temporal profiles in a bike-sharing system in Lyon. Izenman et al. [68] used biclustering to group juvenile-offense data.

Considering triclustering, Martínez-Álvarez et al. [89] analyzed parameters to seismogenic zoning using triclustering in the Iberian Peninsula. Guigourès et al. [55] proposed a technique to analyze time-varying graphs, illustrating it in the London bike-sharing system. Wu et al. [140, 141] used triclustering to analyze meteorological data from Duch weather stations. Melgar-García et al. [91] applied a triclustering-based algorithm to discover patterns over time in maize crops in Portugal to help farmers improve their harvests.
Additionally, we must point out that the biclustering problem is known under the name of multidimensional time series motif discovery in time series literature [7, 15, 125, 145].

**Uses in neurosciences**

Biclustering and triclustering have been used in different Spatio-temporal application domains. Applications on neuroimaging data are not typical, but some approaches have been used in the past. Busygín et al. [25, 45] used a form of biclustering known as consistent biclustering to analyze EEG data and identify brain regions activated with a stimulus. Amar et al. [11] used a variation of a biclustering algorithm to analyze three-way fMRI data (Region X Time X Subject) in order to identify coherent regions over multiple subjects. Their results suggest the capacity of this family of algorithms to identify pertinent brain regions.

### 3.4 Bicluster-based classification

Biclustering has been used for supervised learning tasks [27, 28, 90]. Since a bicluster is representative of a certain characteristic of a subject, the presence of a bicluster can be used to discriminate characteristics of the population. Figure 3.4 illustrates this approach.

<table>
<thead>
<tr>
<th>Patient</th>
<th>B1</th>
<th>B2</th>
<th>B3</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure 3.4: After discovering the biclusters for a group of subjects, a data matrix can be obtained locating the presence of some bicluster in a subject, and then used for classification tasks. The biclusters (sets of features and corresponding representative values) are used as features (Figure adapted from [90]).

Bicluster based classification have been used previously for both temporal and non-temporal data. Considering temporal data, Carreiro et al.[27] considers biclusters of *Genes × Time* to identify biological processes related to the progression of Multiple Sclerosis patients. This approach was extended, considering not biclusters, but meta-biclusters (clusters of biclusters) [28]. In these approaches, the biclustering algorithm CCC was used as a base for the classifiers.

Matos et al. [90] used the *BicPAM* biclustering algorithm as a base to analyze non-temporal data, together with the concept of meta-biclusters to characterize amyotrophic lateral sclerosis patients.
3.5 Motivating Biclustering

fMRI time series data is not a traditional area of application of biclustering. Therefore, the number of previous direct work on the field is limited. Despite the lack of a research line combining biclustering and fMRI data, there are hints in the literature that suggest opportunities for these algorithms to be applied in this type of data.

When analyzing the methodology conducted by clustering, it is usual to cluster brain regions using the temporal series as features. However, since the brain structure is dynamic, different brain regions will fire together based on different stimulus [17, 106].

This dynamic behavior of the human brain motivates the need for a more flexible approach, identifying brain regions that fire together under different temporal stamps, exactly the critical goal of biclustering. In figure 3.5a, the time-series signal from two non-adjacent brain regions demonstrate that the correlation between the two regions is not static. Biclustering promises to detect precisely the time-specific correlation patterns while ignoring the non-correlated regions and time stamps, obtaining more flexible structures than traditional clustering. Figure 3.5b shows a whole-brain time series heatmap. Some row and column clusters are visible in the heatmap, and some events happen only for some regions for some specific temporal stamps. Traditional clustering tasks do not detect this type of structure.

![Figure 3.5a](image1.png)
(a) Correlation time series between two non-adjacent brain regions.

![Figure 3.5b](image2.png)
(b) Whole brain time series.

Figure 3.5: The relation among brain regions isn’t constant over time. Different regions will react together under some specific temporal stamps. Traditional clustering will not be able to detect the visible biclusters-like structures (Figures adapted from [17]).
Another hint that biclustering could be of particular interest for analyzing fMRI data comes from graph theory. Heuvel et al. [130] and Rubinov et al. [113] discuss the use of graph theory to analyze brain network, in particular, the use of fMRI time series to construct the brain network. This approach to analyze fMRI data has been used in the past, not only to cluster the brain [96, 120], but also in tasks such as detecting schizophrenia [16].

According to Madeira and Oliveira [84], and Henriques et al. [62], there is a connection between data matrices and graph theory since a data matrix can be seen as a weighted bipartite graph. Biclustering can thus be used to find maximal cliques or other structures from graphs obtained from binary or real-valued matrices [29, 86], where edge values identify connection strength.
Chapter 4

Biclustering fMRI Time Series: A Comparative Approach

Dozens of biclustering algorithms have been developed during the last two decades [143]. In an application, context is not viable to test every single available biclustering algorithm. Therefore, decisions regarding the best algorithm to use must be made.

Multiple biclustering reviews were created to guide the research. General surveys provide some qualitative answers by providing abstract categorizations for the algorithms and reviewing state-of-the-art approaches and typical application cases. Comparison surveys provide quantitative answers about the performance of the biclustering algorithms. In these comparison studies, biclustering algorithms are typically compared in gene expression benchmark datasets, generating a bias in research. Results in real datasets are usually compared using biological relevance measures, such as the Gene Ontology (GO) annotations [100], specific to the gene expression data, and therefore not useful for any other context. To our knowledge, biclustering algorithms were never compared in a real-world biomedical context, besides gene expression data.

This work aims to evaluate the effectiveness of biclustering algorithms outside the gene expression context. fMRI is used as an application domain with the object of finding coherent and relevant brain activity patterns. Seven state-of-the-art biclustering algorithms were selected (to cover different search strategies) and applied to synthetic and real-world fMRI datasets. Additionally, we added to this comparison two variations of three clustering algorithms, generally used to analyze fMRI data. In the absence of ground truth to evaluate biclustering solutions, we opted for internal valuation metrics and used them to determine the type of patterns expected in fMRI data.

Additionally, this work proposes a methodology to evaluate biclustering in application areas beyond the traditional gene expression context by using independent internal evaluation metrics. Besides neurosciences, we hypothesize that climate science, epidemiology, and sociology, sharing fundamental properties with the former, are scientific fields expected to benefit from this study.

This chapter is organized as follows. Section 4.1 describes the methodology followed and the used
Chapter 4  Biclustering fMRI Time Series: A Comparative Approach

algorithms and evaluation metrics. Section 4.2 expands on the goals behind each data collection. Section 4.3 presents and discusses the results, while section 4.4 highlights some of the biclusters found in order to illustrate the potentialities of the proposed biclustering approaches in fMRI data analysis. Finally, section 4.5 draws conclusions and discusses further implications.

4.1 Methodology

4.1.1 Types of signal

In real-valued matrices there are three main signal categories. It is possible to identify constant, shifting or scaling patterns. This patterns were introduced in section 2.3, and different biclustering algorithms will find different patterns. This motivates the use of an internal comparison metric such as the virtual error, to compare them. However, this type of comparison does not add knowledge in terms of the types of signals found in fMRI data by biclustering algorithms. Knowing the signal is critical to select a biclustering algorithm best suited for these types of patterns.

The virtual error can detect any of the three previously mentioned patterns (constant, shift, and scaling). Additionally, Padilha and Carvalho [101], in a study consisting of 9 biclustering algorithms and a benchmark of 19 different real gene expression datasets, showed that the Virtual Error is not correlated with any of these three measures. In this context, since the virtual error promises to be a stable measure for different types of biclusters, we expect it to be correlated with finding the most common pattern. Therefore, a practical approach to detect the type of patterns found in the data is to study the correlation between the three mentioned quality measures (Variance, MSR, and SMSR) and the independent virtual error. For that, we used the square of the Pearson coefficient.

4.1.2 Selecting biclustering algorithms

Two possible approaches can be considered for selecting the algorithms to be tested. A first possible approach is to review the general comparisons over the biclustering algorithms and select the ones that obtain better results (such as [42, 62, 74, 100]). A problem with this approach is that it is blind to the context and leads us to use a generic metric to make decisions on the “best algorithm”. A second approach is based on real-world knowledge to guide the investigation, identify the type of signal present in fMRI data, and properly select the biclustering algorithms.

In this work, we considered both approaches when selecting the biclustering candidates. In particular, we considered it necessary that the selected biclustering algorithms could detect shifting and scaling patterns and deal with noisy datasets. There is also the possibility of dealing with the temporal dimension. We also considered algorithms that cover different search strategies. In this context, we selected seven
state-of-the-art biclustering algorithms to cover four search strategies. Table 4.1 highlights the general characteristics of these algorithms, together with the reasons for their selection.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Type of Search</th>
<th>Available at</th>
<th>References</th>
<th>Reason to choose it</th>
</tr>
</thead>
<tbody>
<tr>
<td>BicPAM</td>
<td>Exhaustive</td>
<td>BicPAMS</td>
<td>[60, 63]</td>
<td>It is a state-of-the-art pattern mining based biclustering method</td>
</tr>
<tr>
<td>CCC</td>
<td>Exhaustive</td>
<td>BiGGEsTS</td>
<td>[51, 85]</td>
<td>Allows to obtain temporal contiguous biclusters</td>
</tr>
<tr>
<td>ISA</td>
<td>Greedy</td>
<td>isa2</td>
<td>[21]</td>
<td>It is a state-of-the-art greedy algorithm able to deal with real data</td>
</tr>
<tr>
<td>XMotifs</td>
<td>Greedy</td>
<td>biclust</td>
<td>[73, 98]</td>
<td>It is a state-of-the-art greedy algorithm based on a strategy of discretizing data</td>
</tr>
<tr>
<td>Bimax</td>
<td>Divide and Conquer</td>
<td>biclust</td>
<td>[73, 107]</td>
<td>It is a very fast algorithm able to detect simple structures</td>
</tr>
<tr>
<td>FABIA</td>
<td>Distribution Parameter Identification</td>
<td>FABIA</td>
<td>[64]</td>
<td>State-of-the-art algorithm</td>
</tr>
<tr>
<td>Spectral</td>
<td>Distribution Parameter Identification</td>
<td>biclust</td>
<td>[73, 79]</td>
<td>State-of-the-art algorithm able to detect a specific type of bicluster structures</td>
</tr>
</tbody>
</table>

Table 4.1: Biclustering algorithms considered for this study. Additionally, they are going to be compared to the most used algorithms for clustering fMRI data: k-means, spectral and ward’s hierarchical methods. For clustering, we are using the scikit-learn implementations.

We further compare the biclustering approaches to traditional clustering approaches. Clustering approaches are usually used to group brain-regions with similar activity over time [52, 87, 128, 139]. Among the most popular clustering approaches are the K-means algorithm, Spectral Clustering, and Hierarchical methods (in particular using the Ward’s algorithm) [128]. However, both row and column clustering make sense and can be considered. In what follows, and considering data format, traditional row clustering and column clustering will be referred to as region clustering and temporal clustering, respectively.

Seeing clustering as a particular case of biclustering enables the use of biclustering quality metrics to evaluate both and obtain a comparison between the capacity of the two families of methods to obtain patterns.
4.2 Data and Experimental Setup

In this work, we used three data collections. The first consisted of a single synthetic dataset (one subject time series) and was used to test biclustering parameters. Then, we compared the algorithm on two data collections. One has synthetic nature, and the other is a real dataset. In both collections, 20 subjects are used to compare the capacity of algorithms to detect structures and obtain some insight into the structures present. In the datasets where more than one subjects is present, Biclusters are calculated for each subject and then our evaluation metrics are calculated on this collection of generated biclusters.

The synthetic datasets were obtained using the SimTB simulator [43] to generate data consistent with an auditory oddball experiment [77]. This approach was validated in previous studies and is part of the documentation of the SimTB software [10, 43]. Each dataset consists of a single session with 150 temporal stamps (with a time between each temporal stamp of 2 seconds) and around 30 brain regions.

The real data collection was collected by Vaisvaser et al. [129] to analyze the stress response. This data was preprocessed by Amar et al. [11]. The final collection contains 20 datasets, consisting of 463 brain regions and 94-time stamps (with a time between each temporal stamp of 2 seconds).

In terms of parameter choice, a popular decision is to test the algorithms under the default conditions. However, as pointed by Eren et al. [42], the selection of the right parameters is of particular concern to optimize results. While it is not viable to test every single parametrization for every single algorithm, we decided to test some different configurations of the algorithms on the first single synthetic dataset, but avoiding a grid search for the optimal parameter choice that, not only is not computationally viable but that could lead to potentially overfitted results.

Based on this initial test, algorithm configurations were based on minimizing the median virtual error for the obtained biclusters. After selecting the set of configurations to be tested, the algorithms were then compared under the two remaining data collections. For these data collections, the point was to verify the viability of biclustering algorithms under the following conditions: 1) biclustering versus region and temporal clustering; 2) different biclustering algorithms; 3) top-K biclusters; 4) bicluster size; 5) type of bicluster pattern found.

4.2.1 First data collection: testing configurations

This dataset consists of a single synthetic fMRI scan (obtained using the simTB simulator) consisting of 26 spatial sources and 150-time points separated by 2 seconds. For this data collection, it is of interest to understand how good variations of each algorithm are.

In terms of input, the format Regions × Time was selected to run most of the algorithms. This approach was chosen since the algorithms are in general implemented to run in this configuration since they were designed in gene expression context, where genes are usually used in the rows and the conditions, such as time, fit in the columns [84]. An exception was made for BicPAM since this algorithm was reported to be more efficient for matrices with a larger number of rows than columns (which for this dataset
Chapter 4  Biclustering fMRI Time Series: A Comparative Approach

consists of the time × Region format). Since BicPAM allows to select the dimension where to search for patterns, the temporal dimension was chosen in order for the algorithm to find patterns similar to the remaining algorithms. Due to its flexibility in finding different types of patterns, BicPAM was also run in three different configurations to search for constant, shifting, and scaling patterns. The minimum number of biclusters before merging parameter was selected as high as possible to guarantee a good exploration of the dataset (while forcing it to run in a reasonable time). Additionally, BicPAM has a discretization step, and we used the default that uses five symbols.

In FABIA the number of biclusters is always limited: it cannot be higher than the number of rows and the number of columns. In this case, we asked for 26 biclusters (number of columns of the dataset). However, asking for this number of biclusters does not mean that FABIA will find them since the algorithm uses a set of thresholds to filter the results. Two hypotheses were considered: 1) default thresholds and 2) relaxed values to find more biclusters.

In Binax the default values were used for the number of biclusters, changed as follows: 10 biclusters, 100 biclusters, 1 000 biclusters, 10 000 biclusters, and 100 000 biclusters.

In CCC, most of the parameters are associated with the discretization step. We used the two significant possibilities: use a traditional discretization per row (for this, five symbols were used) or use variation between time points (2 or 3 symbols) as described in Madeira et al. [85].

The greedy algorithm XMotifs was executed 30 times (to avoid interference from the starting seeds and following the approach in [100]). Additionally, since it requires a discretization step, the number of used symbols are of particular interest. Five symbols were tested (to be similar to discretizations used by CCC and BicPAM) and two symbols (to be similar to Binax).

Spectral biclustering uses a normalization method. We used the three normalization methods made available by the authors: Logarithmic (log), Independent Recalling (Irrc), and “bistochastization” [79].

An additional concern was forcing the biclusters to have at least two rows and two columns. It was ensured either in the parameterization or as a postprocessing step (by removing this type of biclusters).

4.2.2 Second and third data collections: Verifying biclustering viability

In order to compare the biclustering algorithms, we used synthetic data (using the simtb generator) and real data provided by Amar et al. [11].

In this case, the parameters were selected based on the previous results to minimize the median virtual error. Therefore, for BicPAM the additive configuration was used, for Binax 100 000 biclusters were asked, for CCC a traditional 5 symbols discretization was used, standard configurations were used for FABIA, for XMotifs a discretization step with 5 symbols was used and for Spectral the recommended “log” normalization was used. Additionally, the ISA biclustering algorithm was used. Being a greedy algorithm like Xmotifs, running the algorithm 30 times was used to avoid interference from the starting seeds.
To compare the biclustering approaches with the traditional clustering (\textit{Kmeans}, \textit{Spectral} Clustering and \textit{Ward} Hierarchical methods), both region and temporal variants were tested. In terms of the number of clusters, half of the number of available dimensions was used. This approach was made empirically to avoid clusters with too many elements and clusters with only one element.

4.3 Results and Discussion

4.3.1 Testing configurations

In this scenario, the objective was to test the behavior of different configurations on six bicluster algorithms. Table 4.2 shows the results of this testing.

When focusing on methods, for FABIA, the relaxed configuration achieves far worse median results than the standard configuration. It suggests that the strategy of relaxing the thresholds does not provide more meaningful results. For XMotifs, the strategy of using a discretization with five symbols provided biclusters with smaller errors than using a binary discretization. This conclusion is similar to the one obtained by Kemal et al. [42]; however, it must be pointed that the improvement by increasing the number of discretization levels comes at the price of reducing the number of found biclusters. For Spectral, the three configurations generate similar results in all metrics (except SMSR). The “log” configurations achieved the best performance in all four metrics and is recommended by the authors. So it was selected for the next stages of research.

For Bimax, it is to be noticed that the algorithm’s performance becomes worse with the increasing of the number of asked biclusters. However, we observed that generating more biclusters means adding more noise to the previously generated ones. Additionally, we observed an effect of saturation when generating a large number of biclusters since asking the algorithm to generate “100 000” biclusters does not produce any more biclusters than the “10 000” option.

In contrast to other biclustering algorithms, BicPAM does not have a configuration with the best results in all metrics. Since the criteria to choose the configuration were the virtual error’s value, the ”additive” version was selected. For CCC, another surprising result is the traditional discretization to generate best biclusters than the variations options, contradicting results obtained by Madeira et al. [85]. It could be due to two possible reasons: different application contexts or a different evaluation metric.

4.3.2 Comparing Biclustering and Clustering

In this scenario, the general objective was to compare the capacity of biclustering algorithms with traditional clustering algorithms (in both variants). Results are summarized in figure 4.1.

The general conclusion anticipates biclustering to obtain more homogeneous structures than the traditional clustering structures since it achieves better values in all four quality evaluation criteria.
<table>
<thead>
<tr>
<th>Method</th>
<th>Configuration</th>
<th>Variance</th>
<th>msr</th>
<th>Smsr</th>
<th>Virtual Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>BicPAM</td>
<td>Additive Version</td>
<td>0.024 ± 0.024</td>
<td>0.001 ± 0.003</td>
<td>2.49</td>
<td><strong>0.262 ± 0.232</strong></td>
</tr>
<tr>
<td></td>
<td>Constant Version</td>
<td>0.014 ± 0.026</td>
<td>0.001 ± 0.002</td>
<td>1.96</td>
<td>0.434 ± 0.185</td>
</tr>
<tr>
<td></td>
<td>Multiplicative</td>
<td>0.025 ± 0.024</td>
<td>0.002 ± 0.005</td>
<td>3.07</td>
<td>0.461 ± 0.263</td>
</tr>
<tr>
<td></td>
<td>10 Biclusters</td>
<td>0.004 ± 0.002</td>
<td>0.001 ± 0.001</td>
<td>0.03</td>
<td>0.881 ± 0.148</td>
</tr>
<tr>
<td>Bimax</td>
<td>100 Biclusters</td>
<td>0.006 ± 0.004</td>
<td>0.003 ± 0.002</td>
<td>0.05</td>
<td>0.729 ± 0.128</td>
</tr>
<tr>
<td></td>
<td>1 000 Biclusters</td>
<td>0.008 ± 0.008</td>
<td>0.004 ± 0.004</td>
<td>0.06</td>
<td>0.706 ± 0.168</td>
</tr>
<tr>
<td></td>
<td>10 000 Biclusters</td>
<td>0.008 ± 0.007</td>
<td>0.004 ± 0.004</td>
<td>0.06</td>
<td><strong>0.695 ± 0.169</strong></td>
</tr>
<tr>
<td></td>
<td>100 000 Biclusters</td>
<td>0.008 ± 0.007</td>
<td>0.004 ± 0.004</td>
<td>0.06</td>
<td>0.695 ± 0.169</td>
</tr>
<tr>
<td>CCC</td>
<td>Traditional Discretization (5 symbols)</td>
<td>0.189 ± 0.359</td>
<td>0.017 ± 0.021</td>
<td>0.51</td>
<td><strong>0.288 ± 0.345</strong></td>
</tr>
<tr>
<td></td>
<td>Variation between time points (2 Symbols)</td>
<td>0.596 ± 0.491</td>
<td>0.067 ± 0.108</td>
<td>0.93</td>
<td>0.370 ± 0.315</td>
</tr>
<tr>
<td></td>
<td>Variation between time points (3 Symbols)</td>
<td>0.596 ± 0.491</td>
<td>0.067 ± 0.108</td>
<td>0.93</td>
<td>0.370 ± 0.315</td>
</tr>
<tr>
<td>FABIA</td>
<td>Standard</td>
<td>0.579 ± 0.502</td>
<td>0.006 ± 0.081</td>
<td>0.51</td>
<td><strong>0.079 ± 0.167</strong></td>
</tr>
<tr>
<td></td>
<td>Relaxed</td>
<td>0.946 ± 0.053</td>
<td>0.881 ± 0.095</td>
<td>3485.60</td>
<td>0.787 ± 0.036</td>
</tr>
<tr>
<td>Spectral</td>
<td>log</td>
<td>0.039 ± 0.053</td>
<td>0.033 ± 0.004</td>
<td>670.30</td>
<td><strong>0.726 ± 0.012</strong></td>
</tr>
<tr>
<td></td>
<td>bistochastization</td>
<td>0.040 ± 0.005</td>
<td>0.035 ± 0.004</td>
<td>3232.21</td>
<td>0.731 ± 0.015</td>
</tr>
<tr>
<td></td>
<td>irrc</td>
<td>0.040 ± 0.004</td>
<td>0.035 ± 0.004</td>
<td>1271.71</td>
<td>0.726 ± 0.015</td>
</tr>
<tr>
<td>XMotifs</td>
<td>Discretization with 2 symbols</td>
<td>0.017 ± 0.004</td>
<td>0.012 ± 0.001</td>
<td>1168.76</td>
<td>0.662 ± 0.048</td>
</tr>
<tr>
<td></td>
<td>Discretization with 5 symbols</td>
<td>0.007 ± 0.009</td>
<td>0.003 ± 0.001</td>
<td>6.59</td>
<td><strong>0.585 ± 0.035</strong></td>
</tr>
</tbody>
</table>

Table 4.2: Median values for the four selected measures for the first synthetic dataset, with uncertainties given by the standard deviation (except for the case of smsr were the standard deviation is orders of magnitude higher than the median value). From this results it is visible that A) The high values of uncertainty discourage focus on optimizing the biclustering method parameters and B) Choosing the right evaluation metric is important, however in most of the biclusters cases they seem to agree for the same "best" configuration.
4.3.3 Comparing algorithms

While previous results motivate the use of biclustering, they do not explain each algorithm’s capacity to generate homogeneous algorithms. In this scenario, we tested the performance of each biclustering (and clustering) algorithm. Figures 4.2 and 4.3 illustrate our results in both synthetic and real data.

The first immediate conclusion is related to the lack of clustering algorithms’ capacity to generate good results under the temporal clustering option. It makes sense since these clustering scenarios generate groups of temporal stamps where all regions behave similarly. Since the brain is heterogeneous, different regions will behave differently under the same temporal stamp.

Much more comparable are the tasks of region clustering and biclustering algorithms. In this scenario, most biclustering algorithms achieve worse results than the traditional clustering tasks, except for exhaustive biclustering algorithms. However, the capacity of the exhaustive approaches (BicPAM and CCC) to find homogeneous biclusters shows that while the clustering methods are fundamentally limited (due to the cluster structure), the biclustering task can achieve good results.

Focusing on the individual performance, it is not too surprising that the Bimax algorithm would achieve bad results since it discretizes data in a particular way (it searches for biclusters of ones in a binary dataset). This step of turning the dataset into a binary one strongly limits finding good biclusters, but it can find biclusters very fast. About Spectral, the results are not surprising since it does search for a particular type of bicluster. FABIA has similar problems when compared to clustering approaches since its factor-analysis approach strongly limits the algorithm’s capacity to generate biclusters. About ISA and XMotifs, both produce bad results.

4.3.4 Comparing top-K biclusters

Biclustering solutions are composed of a different number of generated biclusters. While some algorithms generate the best bicluster, others allow selecting the number of desirable biclusters, and others generate
Figure 4.2: Virtual Error measure for every tested algorithm in our synthetic data collection. Despite having great oscillations, the median performance of the exhaustive approaches (CCC and BicPAM) show promising results in comparison with the remaining biclustering approaches.

In this chapter, we selected the $K$ value empirically considering the number of already generated biclusters. For the synthetic data, the $K$ number was selected as 50, and for the real dataset, the selected number was 500. Our results are shown in figure 4.4 showing the capacity of exhaustive approaches to generate homogeneous biclusters. Additionally, ISA had a significant performance improvement, indicating that it could be an exciting choice for dealing with this type of data. XMotifs, while also a greedy algorithm, had a worse performance but still better than the FABIA and Spectral algorithms (that are based on distribution parameter identification).

4.3.5 Size remarks

One possible approach to understanding why some algorithms achieve better results than others is to compare the general properties of the obtained biclusters. Since the algorithms generate biclusters with
Figure 4.3: Virtual Error measure for every tested algorithm in our real data collection. Despite the biclustering algorithms not being indisputable better than the traditional clustering, the use of exhaustive biclustering approaches such as CCC and BicPAM show a good capacity of generating coherent biclusters.
different properties, now we focus on bicluster size parameters. Our results are shown in table 4.3.

A first view of the result shows that the biclustering solutions tend to generate biclusters with sizes that can vary significantly. It makes sense since the search strategies of each algorithm are different.

In the past sections, we observed exhaustive algorithms’ capacity to generate better biclusters than the remaining algorithms. These results show some possible hints of why they do it since CCC and BicPAM generate the smallest biclusters in terms of area. For BicPAM, these results show a lack of subspace exploration. BicPAM achieves excellent results by allowing us to generate very small biclusters. Additionally, while the algorithm promised to generate biclusters of all sizes (due to his exhaustive nature), this comes not only at a price of execution speed (which we observed to be far greater than the remaining algorithms) but also at a running memory price, making this promise to be potentially unfeasible. Despite being also exhaustive, CCC’s temporal contiguity constraints lead to a faster exploration of the datasets.

From the remaining algorithms, FABIA produces larger biclusters. It is related to the search strategy that uses factor analysis as a base. For Spectral, the use of singular value decomposition and the generated checkerboard structures help explain the generated biclusters’ size. Bimax generates biclusters of different sizes due to his simplistic approach. The greedy algorithms ISA and XMotif generate relatively small biclusters.

Finally, clustering approaches generate the largest solutions. It is strongly related to their restriction of including all rows (or columns) in the clustering.
### Table 4.3: Median values (and associated standard deviation) for the typical bicluster dimension parameters in both data collections: number of regions in each bicluster, number of temporal stamps and bicluster area. When comparing this results to the virtual error values, a apparent relation comes between the bicluster size and the associated virtual error, which make sense.

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Synthetic Data</th>
<th>Real Data</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Time Points</td>
<td>Region Points</td>
</tr>
<tr>
<td>Biclustering</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bimax</td>
<td>6 ± 7</td>
<td>4 ± 2</td>
</tr>
<tr>
<td>BicPAM</td>
<td>7 ± 3</td>
<td>2 ± 1</td>
</tr>
<tr>
<td>CCC</td>
<td>3 ± 4</td>
<td>3 ± 2</td>
</tr>
<tr>
<td>FABIA</td>
<td>43 ± 5</td>
<td>4 ± 2</td>
</tr>
<tr>
<td>Spectral</td>
<td>23 ± 11</td>
<td>7 ± 3</td>
</tr>
<tr>
<td>ISA</td>
<td>14 ± 9</td>
<td>3 ± 1</td>
</tr>
<tr>
<td>XMotifs</td>
<td>49 ± 13</td>
<td>2 ± 1</td>
</tr>
<tr>
<td>Region Clustering</td>
<td></td>
<td></td>
</tr>
<tr>
<td>kmeans</td>
<td>150</td>
<td>3 ± 2</td>
</tr>
<tr>
<td>spectral</td>
<td>150</td>
<td>4 ± 4</td>
</tr>
<tr>
<td>ward</td>
<td>150</td>
<td>6 ± 5</td>
</tr>
<tr>
<td>Temporal Clustering</td>
<td></td>
<td></td>
</tr>
<tr>
<td>kmeans</td>
<td>3 ± 5</td>
<td>26</td>
</tr>
<tr>
<td>spectral</td>
<td>3 ± 6</td>
<td>26</td>
</tr>
<tr>
<td>ward</td>
<td>3 ± 9</td>
<td>26</td>
</tr>
</tbody>
</table>
### Table 4.4: Correlation between the virtual error and the three specific coherence measures: Variance (constant biclusters), MSR (shifting biclusters) and SMSR (scaling biclusters). Most of the algorithms agree that the expected patterns are of shifting nature.

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Synthetic Data</th>
<th>Real Data</th>
<th>Type of Pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Variance</td>
<td>MSR</td>
<td>SMSR</td>
</tr>
<tr>
<td>BicPAM</td>
<td>0.009</td>
<td>0.133</td>
<td>0.000</td>
</tr>
<tr>
<td>Bimax</td>
<td>0.003</td>
<td>0.037</td>
<td>0.087</td>
</tr>
<tr>
<td>CCC</td>
<td>0.112</td>
<td>0.007</td>
<td>0.000</td>
</tr>
<tr>
<td>FABIA</td>
<td>0.051</td>
<td>0.627</td>
<td>0.061</td>
</tr>
<tr>
<td>ISA</td>
<td>0.050</td>
<td>0.453</td>
<td>0.000</td>
</tr>
<tr>
<td>Spectral</td>
<td>0.009</td>
<td>0.125</td>
<td>0.000</td>
</tr>
<tr>
<td>XMotifs</td>
<td>0.115</td>
<td>0.378</td>
<td>0.016</td>
</tr>
</tbody>
</table>

#### 4.3.6 Types of bicluster patterns

To detect the pattern structures found by the biclustering algorithms, we used the square of the Pearson coefficient between the variance, MSR and SMSR and the virtual error. A high correlation could be indicative of the expected type of pattern. Our results are shown in Table 4.4.

Most biclusters agree that the expected patterns have a shifting nature. Bimax is one of the exceptions, supporting constant and scaling patterns. However, it must be pointed out that Bimax does not recognize any particular type of structure since it works only with binary data. Another exception is CCC, supporting the hypothesis of constant patterns. It could be a consequence of the temporal contiguity constraint that generates biclusters that are fundamentally different from those obtained by other algorithms.

#### 4.4 Illustrative Results

With the objective of highlighting the potentialities of biclustering of detecting structures, some illustrative biclusters were selected from the CCC biclustering algorithm outputs. This algorithm was chosen not only because it achieves good results in the previous analysis but generates contiguous temporal biclusters with a more straightforward interpretation. Illustrative results are shown in figure 4.5.

Biclustering can find regions with similar activity over time (in synthetic data, the bicluster found four regions programmed to have a high correlation). When comparing both results, biclustering found temporal events with temporal amplitudes consistent with the hemodynamic model. Other biclustering algorithms will still find highly correlated regions, but instead of finding temporal intervals, they will find temporal stamps.
Figure 4.5: Illustrative biclusters found in synthetic and real data. Biclustering is able to discover a subset of brain regions that behave similarly during a temporal interval. This type of structures go beyond traditional clustering.

4.5 Conclusions and Future Prospects

This chapter evaluated the use of biclustering in the context of fMRI data. Seven state-of-art biclustering algorithms were selected, compared among with other and with three traditional clustering algorithms. Our results show that most of the biclustering methods cannot surpass the traditional clustering when using the virtual error metric. However, the tested exhaustive biclustering methods (BicPAM and CCC) can achieve the best coherency values of all tested algorithms. Additionally, and independently of the measured homogeneity, it must be pointed that biclustering can be useful and an improvement compared to clustering only by its more flexible nature of considering both spatial and temporal dimensions to discover the groups. Additionally, we observed that the bicluster structures found in this type of data have a shifting nature.

Focusing on individual performance, BicPAM achieved high homogeneity levels by generating many small biclusters. It is not consistent with the promise of an exhaustive search. The issue is related to the algorithm’s performance since, to achieve bigger biclusters, it needs a prohibitive quantity of time and resources. It could be solved considering two approaches: the first one is related to the algorithm that can be optimized to generate results faster. Another is related to the algorithm’s parameterization that could be used if the size of the desirable biclusters is known apriori. It must be noticed that these observations are not inconsistent with the obtained results: even if this is considered, BicPAM will still be able to achieve adequate levels of homogeneity in its best biclusters.

Another impressive result comes from greedy biclustering solutions, ISA, and XMotifs. While not being able to achieve results as impressive as the exhaustive ones, they can still find some good ones using a fraction of the time BicPAM needs to operate. Additionally, while ISA works by using the real data directly, XMotifs requires a discretization step, allowing different approaches.

Considering the trade-off between the number of generated biclusters, their quality, and execution
time, Bimax is an interesting choice since it achieves a vast number of biclusters very fast. It is achieved by doing a binarization of the data. While not detecting the best biclusters, he is very fast, so it means that the algorithm could get some insights on the expected biclustering structures before running other algorithms.

FABIA and Spectral are not able to achieve good results. Despite being both based on the same search type (Distribution Parameter identification), we hypothesize that the reasons for this are different. FABIA's search strategy means that the number of solutions will be limited to the number of columns of the dataset, which will strongly limit the capacity to generate solutions with many biclusters. Spectral, on the other hand, is limited due to the generated particular checkerboard bicluster structure.

The last algorithm considered is CCC, which uses temporal contiguity notions during the exhaustive search for biclusters. It means that CCC can find multidimensional time series motifs. It leads to the generation of many biclusters with a straightforward interpretation that could be ideal for these analyses. Since time is contiguous in these biclusters, a new set of quality measures based on statistical significance can be used to improve when filtering results. Furthermore, the high coherency levels observed in this study motivate specific temporal biclustering methods to study fMRI data.

Biclustering is a tool to search for patterns in data. This chapter suggests that biclustering is equally promising in fMRI data. While comparative studies guide the selection of the methods, the biclustering algorithm’s choice to analyze must be guided by the study objectives.
Chapter 5

Improving Biclustering Interpretability: A Pattern Mining Approach

The biclustering task became popular over the last two decades and has been successfully applied in domains such as biomedicine, text mining, and marketing analysis [26, 84]. Despite that, extracting knowledge from biclustering solutions is still fuzzy since most research is focused on the development of new algorithms and less on their evaluation and interpretation [143].

There are essentially two ways to evaluate a bicluster. In quantitative terms, there are multiple measures, either measures of biological significance, internal evaluations or external evaluations. Considering a qualitative evaluation, one bicluster can be analyzed considering the rows and columns included, relying on visualization methods and previous domain knowledge; This approach is called biological interpretation.

We do hypothesize that the current methodology of extracting knowledge from biclusters has three significant flaws:

1. Most options for biclustering visualization are based on line plots and heatmaps. These options are not reliable for a high number of biclusters (and even less to visualize overlapping ones).

2. The remaining biclustering evaluation research is based on analyzing individually biclusters generated by the algorithms. This description of a biclustering solution is equivalent to describing a tapestry as a collection of nice threads while ignoring their entanglement.

3. Most biclustering algorithm development is affected by a lack of diversity associated with the traditional gene expression application context. In this application context it is usual too use measures of biological significance, not usefull for any other context.

In this chapter, we propose a new biclustering evaluation approach not based on the bicluster element but on the biclustering solution as a whole. For that, pattern mining algorithms operate on the biclustering solution to extract frequent itemsets. This chapter focus on explaining:
Chapter 5

Improving Biclustering Interpretability: A Pattern Mining Approach

- The theoretical base of our approach,
- Why makes sense,
- Measures that can be extracted,
- Practical applications for extracting additional knowledge.

Rather than providing a well-defined methodology, this chapter digresses on the general aspects that we hypothesize can be achieved using this analysis approach. fMRI data is used as an application scenario. However, this approach can be used in any other scientific domain and whenever using any biclustering algorithm to generate many biclusters.

This chapter is organized as follows. Section 5.1 motivates and explains our approach. Section 5.2 develops on the datasets used for this chapter and the generally used methodology. Section 5.3 validates the approach using correlation analysis, exploring a biclustering solution in terms of both dimensions. Section 5.4 continues to expand this idea by introducing non-symmetric measures of connectivity. Section 5.5 explores the capacity of this approach to model the generation of hypergraphs considering diverse biclustering algorithms. Section 5.6 compares the biclustering solutions obtained by the biclustering algorithms. The later two sections are exploratory only and are focused on extending fMRI pattern-bases analysis. Section 5.7 expand the dimension of analysis by motivating and proposing a methodology for triclustering. Section 5.8 presents the hypothesis of using this approach to conduct biclustering-based classification. Finally, Section 5.9 draws conclusions and discusses further implications.

5.1 Our approach

A bicluster $B^k$ can be formally defined as a subset of $I$ rows and $J$ columns $B^k = \{x_1, ..., x_I, y_1, ..., y_J\}$ from an original data matrix [84]. A biclustering is a collection of $K$ biclusters $B \triangleq \{B^1, ..., B^K\}$, usually associated to the solutions obtained by executing an biclustering algorithm [65].

Consider a biclustering solution consisting of three biclusters:

- $B^1 = \{x_1, x_2, y_1\}$
- $B^2 = \{x_1, x_3, y_2, y_3, y_4\}$
- $B^3 = \{x_1, x_3, y_5, y_6\}$

Hints of a relation between biclustering and hypergraph theory are present in the biclustering literature. However, this formal connection was not established as far as we know, and is done considering a bicluster as a biclique in a bipartite graph [84]. Every hypergraph has a corresponding bipartite graph
[24]. This means that a single bicluster could be seen as a hyperedge in a hypergraph (see figure 5.1a). A bicluster’s presence defines a hyperedge in a hypergraph (see figure 5.1b).

Current state-of-the-art biclustering algorithms can generate thousands of biclusters. As a consequence, using biclusters to define hyperedges is not practical. The use of pattern mining on the set of biclusters allows to obtain the most statistical significant hyperedges (metrics such as support can be used to weight the hyperedges). Additionally, metrics such as confidence can be used to define directed hypergraphs. From the previous set of biclusters, it is possible to extract the most frequent patterns. Table 5.1 represents this step of obtaining the most frequent patterns from the biclustering solution.

These results allow simplifying the previous hypergraph considering only the most frequent patterns. Additionally, using a pattern mining metric allows us to weigh this hypergraph (for the simplified hypergraph, check figure 5.2).
Chapter 5  
Improving Biclustering Interpretability: A Pattern Mining Approach

<table>
<thead>
<tr>
<th>Itemset</th>
<th>Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>((x_1))</td>
<td>1</td>
</tr>
<tr>
<td>((x_1,x_3))</td>
<td>(2/3)</td>
</tr>
<tr>
<td>((y_2,y_3,y_4))</td>
<td>(1/3)</td>
</tr>
<tr>
<td>((y_5,y_6))</td>
<td>(1/3)</td>
</tr>
<tr>
<td>((y_1))</td>
<td>(1/3)</td>
</tr>
</tbody>
</table>

Table 5.1: Top five more frequent itemsets (there are other itemsets with support \(1/3\) that for simplicity were excluded).

Figure 5.2: Filtered hypergraph considering only the top 5 more frequent itemsets. In this particular case, most of hyperedges are actually traditional graph edges. If we force the itemset search to find only links of size two or smaller, this technique will define traditional graphs.
5.2 Data and Experimental Setup

5.2.1 Datasets

We selected two real datasets corresponding to well studied real datasets freely available.

The first is called “Flanker task” and comprises data collected from 26 healthy adults while they performed a slow event-related Eriksen Flanker task [76, 92, 93]. This dataset is used during most of the chapter 1.

The second dataset is called “Working memory in healthy and schizophrenic individuals”. This dataset consists of 20 healthy and 23 schizophrenic subjects performing and n-back task [108, 109]. This second dataset is used exclusively in section 5.8 2.

Traditional fMRI data pre-processing was conducted using the FSL software [71, 123, 138] and its tools. Firstly, BET (Brain Extraction Tool) was used to delete non-brain tissues [122]. Pre-defined parameters were used except the Fractional intensity threshold, which we empirically set to 0.6 to obtain smaller brain outline estimates (the default was not enough to filter the non-brain tissues).

FMRI data processing was carried out using FEAT (FMRI Expert Analysis Tool) Version 6.00, part of FSL (FMRIB’s Software Library, www.fmrib.ox.ac.uk/fsl). Registration to high resolution structural and/or standard space images was carried out using FLIRT [69, 70]. The following pre-statistics processing was applied:

- motion correction using MCFLIRT [70],
- non-brain removal using BET [122],
- spatial smoothing using a Gaussian kernel of FWHM 5mm,
- grand-mean intensity normalization of the entire 4D dataset by a single multiplicative factor,
- highpass temporal filtering (Gaussian-weighted least-squares straight-line fitting, with sigma=50.0s).

Finally, our dataset must be downsampled. For that, we used the Harvard-Oxford cortical atlas to downsample the data into 48 Regions of interest. This atlas is a popular choice and is part of the FSL neuroimaging software [34, 36, 71].

5.2.2 Experimental Setup

The objective of this chapter is only to illustrate our approach to extract knowledge from biclustering. Therefore, we focus only on the general approach since it can be adapted to obtain better results.

---

1 Original data available at the openneuro portal https://www.openfmri.org/dataset/ds000102/
2 Original data available at the openneuro portal https://www.openfmri.org/dataset/ds000115/
we use the same biclustering algorithms used in chapter 3. In most of the sections, the CCC biclustering algorithm was chosen for two reasons: first, it achieved interesting scores in the previous chapter, and second, it forces the biclusters to be temporally contiguous, which is easier to understand. We used the original Java jar implementation\(^1\), considering the standard parameters. In some sections, the algorithms BicPAM, Bimax, ISA, FABIA, XMotifs and Spectral were used. The previously mentioned implementations were used, and the standard parameters were selected.

Then, our approach requires the application of a pattern mining algorithm and the selection of metrics. For our approach, we used the FP-Growth algorithms to generate frequent itemsets. We used the support measure to weigh the itemsets during most of the sections. To better analyze the results, the support results were rescaled to have a maximum value of 1. In section 5.4, a rescaled version of the confidence measure is used alongside the support one.

## 5.3 Correlation analysis

The simplest analysis that can be conducted is to consider itemsets of length 2. This effectively groups pairwise correlations among items. With itemsets of length 2, there are three options:

- Time-Time
- Region-Region
- Region-Time

Our measure of connectivity will be the symmetric pattern mining measure support, rescaled for visualization purposes.

### 5.3.1 Time-Time analysis

Searching only for relations in the context time-time is interesting since it gives us clues of the general temporal activity.

Figure 5.3 shows us results. On the left, pairwise correlation between temporal stamps. CCC obtains temporally contiguous biclusters. Therefore, one temporal stamp will be strongly correlated only to the near temporal stamps (other biclustering algorithms will not observe that). Additionally, in this figure there are “clusters” of activity near temporal stamps 100 and 125. These results can be compared with the frequencies observed for each temporal stamp (right plot). The same two peaks of activity near stamps 100 and 125 are visible. These two plots show some hints about our dataset and the registered activity.

\(^1\)Implementation available at http://kdbio.inesc-id.pt/software/ccc-biclustering/
5.3.2 Region-Region analysis

After analyzing the temporal profile of our temporal solution, we can try to understand if the relations found by biclustering between regions make sense. This type of correlations measures is not new, and metrics such as the Pearson coefficient (using the raw time series as input) are standard in the literature [59]. Since the Pearson correlation coefficient admits both positive and negative values, there is a need to change this measure. First, we considered only the absolute of these values. Second, it can be rescaled to be between 0 and 1. We can then compare the correlations detected between the two correlation metrics.

Figure 5.4 shows the results. Visually comparing the heatmaps, similar structures between the two correlation maps are visible, suggesting that the biclustering solution could obtain expected correlations among the data.
5.3.3 Region-Time analysis

Finally, we can consider itemsets of length two considering region and time. Since these are the dimensions of the dataset (Region X Time), we expect the result to resemble the original dataset.

Figure 5.5 illustrates the results. At first glance, it is possible to visualize similar structures in both heatmaps, such as column-like events near stamps 100 and 125. However, our results show more than that. Since CCC searches for temporally contiguous biclusters, randomly peaks of activity in a single region will be ignored by the algorithm (in the heatmap, this is highlighted). Thus this allows us to filter the noise to observe the data.
Figure 5.5: Comparing the original dataset with the length 2 region-time itemset allows to observe what regions and what time stamps were most used in the biclustering. Similar structures can be found.

5.4 Effective connectivity

In the previous section, a rescaled version of support was used to extract knowledge from biclusterings. Our approach uses metrics of rule interest, therefore, inherits their limitations. One problem with using support as measure is that regions with high activity will be more present in this search for frequent itemsets. Correlated regions with low activity will thus not be found using the support measure. Pattern mining solved this problem by introducing other measures such as confidence, that measures the probability of the consequent given the presence of antecedents.

As an additional consequence, this type of measure has an interesting property. Unlike support, the confidence measure is not symmetrical, e.g. $\text{confidence}(A, B) \neq \text{confidence}(B, A)$. In practical terms, this means that this type of measure defines a directed graph (in the case of B and A being a single item) or a dihypergraph in the general case.

To illustrate this, we consider the simplest task: comparing the results of support and confidence for pairwise relations. Figure 5.6 highlights this for the case of relations Region-Region. The heatmap associated with the confidence measure is a little asymmetrical, which makes sense. However, this asymmetry is weak, which is fundamented by traditional studies of effective connectivity, suggesting a symetrization of these measures [113, 130].
Figure 5.6: Effective connectivity between pairs of regions. As expected, the confidence measure is not symmetrical. Additionally, the confidence plot highlights correlations that are not present in the support case. Notice however that the plots are similar.

5.5 Using biclustering to define hypergraphs

The previous examples used pairwise relations to illustrate the potentialities of this approach on graphs. However, these capacities go beyond that, since biclusters have usually more than two lines or two columns.

Pattern mining algorithms generate arbitrary sized itemsets. It means that this approach allows generating hyperedges of arbitrary length.

Traditional hypergraph hyperedge weight calculation is usually a hard combinatorial task. If we consider a not-directed hypergraph consisted of $n$ hypernodes and admit a maximum size $k$ for the hyperedges, the total number of edges $N$ will be given by:

$$N(n, k) = \sum_{p=1}^{k} \binom{n}{p}.$$  \hspace{1cm} (5.1)

Admitting even a small data matrix of ten rows and ten columns and a maximum size $k = 5$ for the hyperedges, the number of possible hyperedges will be $N \approx 21700$. For the dirhyperedge case we use
permutations instead of combinations, and in that simple case, the number of possible directed hyperedges is a hundred times higher.

Using biclustering has two advantages against a combinatory approach. First, non-interesting links are automatically cropped out during the execution of the algorithm. Additionally, when executing the search for frequent itemsets, minimum support can be defined, cropping the search space.

Different algorithms will crop the number of total links differently. For this section, let’s consider the biclustering algorithms we used in the previous chapter: BicPAM, CCC, Bimax, FABIA, ISA, XMotifs and Spectral. Let’s admit, once again for simplicity, we are only interested in correlations among brain regions.

Figure 5.7 shows how successful each algorithm is to crop itemsets. While the number of hyperedges increases with the increase in the number of hypernodes, the relative number to the combinatory limit decreases fast, showing that this is a fast promising way to compute hyperedges in a hypergraph, avoiding the high combinatory cost. Additionally, different algorithms will be more or less effective for cutting hyperedges. Typically, exhaustive algorithms look less effective in reducing the number of hyperedges. However, Bimax becomes worse than the exhaustive ones. Due to its particular simplification of the data (Bimax binarizes the data before dividing it into biclusters), it finds a high number of huge biclusters.

![Figure 5.7: Number of hypernodes generated by the biclustering algorithms. With the increase of possible hypernodes, the quantity of generated hyperedges by the biclustering algorithms decreases when comparing to the combinatory limit.](image)
Comparing biclustering approaches

After obtaining the biclusterings, an important question is what did the biclustering find?. This question falls in the context of the biological interpretation of biclustering, using the evaluation system defined by Horta and Campello [65]. So far, we defined a new way to conduct this analysis, considering not the bicluster properties but rather the properties of the biclustering.

Besides analyzing the results of a biclustering using this tool visually, it is also possible to evaluate how good they are. One possible way to know this is to use more than one bicluster algorithm and then compare the results. If the results are similar, we expect then to be good and are not a statistical fluctuation.

Previously we compared the Region-Region results between the CCC biclustering algorithm and the Pearson coefficient, computed considering the time series. Figure 5.8 does the equivalent comparison between the exaustive biclustering algorithms, CCC and Bicpam.

![Figure 5.8: Pairwise region correlation in two algorithms. The similarities between the two plots give us some confidence that the two algorithms are discovering some significant pattern, at least pairwise ones.](image)

Additionally, a correlation measure can be used to investigate how similar the results of each biclustering algorithm are. Figure 5.9 show us the similarities between the previously used algorithms considering the region dimension and relations of size 2 and 3. Our results show that considering the region dimension, the results from CCC and BicPAM are similar, which is impressive considering that CCC searches only for temporally contiguous biclusters. For the remaining algorithms, the relations are
not that strong (and are uniform), suggesting that these algorithms do not find good biclusters.

![Figure 5.9: Correlation between biclustering results. As visible before, the two exhaustive algorithms, CCC and BicPAM seem to locate the same type of relation between items. For the remaining biclustering algorithms this correlation is not that strong.](image)

### 5.7 A roadmap for triclustering

The previous sections considered the biclustering task of analyzing a subject scan to obtain strong correlations considering both regions and temporal stamps. However, an essential task in fMRI research is to distinguish groups of patients with similar brain activity. In terms of biclustering, this means that similar patients will have similar interactions between brain regions. Additionally, if the external stimulus is equivalent between patients (which can be experimentally imposed), the temporal activity should be similar among similar patients. Figure 5.10 illustrates this concept, showing the (rescaled) support of individual brain regions or temporal stamps for every available subject.
Chapter 5  Improving Biclustering Interpretability: A Pattern Mining Approach

Figure 5.10: Activity for every available patient according the two considered dimensions: Brain regions and temporal stamps. Some activity only happens for some patients.

Since we can consider the patient dimension, we can turn this into a triclustering problem and look for tensors in the format \( \text{Region} \times \text{Time} \times \text{Subject} \). Most triclustering algorithms are based on greedy searches, and concepts such as time contiguity are not currently well explored in the literature. This section aims to explore the potentialities of using a biclustering-based approach to obtain flexible triclusters.

Triclustering algorithms based on biclustering are not new in the literature, and it is usual to used 2D merit functions to discover biclusters for each slide of a 3D data space [61]. The most interesting case is the TWIGS triclustering algorithm, which was tested on fMRI data using the Bimax biclustering algorithm as base [11]. Our approach is simple.

1. First, use a biclustering algorithm on two dimensions of the data and separately for each stratum of the third dimension. For example, use biclustering in \( \text{Region} \times \text{Time} \) independently for each subject. It has two great advantages. **First**, it opens up the biclustering literature and all its algorithms. **Second**, it allows distributing the computations.

2. Then, do pattern mining over the formed biclusterings to find frequent itemsets. Searching for these patterns can be computationally expensive. However, it can be controlled, setting minimum support and a maximum itemset size. Additional options are to search for patterns other than the frequent ones. This task can still be distributed.

3. Our result will be a set of patterns associated with a patient, and each pattern will be defined with a metric such as its support.
4. At this point, we can search for frequent patterns among the multiple subjects. Our final results will be a set of patients with matching patterns. Since these patterns have both temporal and regions, our result will be a Tricluster.

The need for unsupervised analysis of three-way data motivated the development of triclustering algorithms, with an application beyond the biclustering traditional gene expression context in a diversity of contexts. This roadmap can lead to the development of triclustering algorithms with the flexibility of using any biclustering algorithm as a base (or even more than one) and have the freedom to select the variable to stratify the data.

### 5.8 Biclustering based classification

Biclustering can be used for classification. The biclusters can be used directly, meaning that if a biclustering method generates hundreds of biclusters, the classification dataset will have hundreds of columns. An improvement is to consider not the raw biclusters but meta-biclusters, i.e., clusters of biclusters \[28, 90\].

In this chapter, instead of considering biclusters, we analyze patterns found inside biclustering solutions, and we hypothesize that these patterns can be used for classification. For example, if we consider only itemsets of length two between regions, we used biclustering to estimate our patients’ connectivity matrix and then use the differences to perform classification. Table 5.2 illustrates the difference between a raw approach and our approach.

<table>
<thead>
<tr>
<th>Patient</th>
<th>B1</th>
<th>B2</th>
<th>B3</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

(a) Traditional Biclustering based classification: Biclusters as features

<table>
<thead>
<tr>
<th>Patient</th>
<th>T1</th>
<th>T2</th>
<th>T3</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.6</td>
<td>0.4</td>
<td>0.8</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0.7</td>
<td>0.2</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>0.2</td>
<td>0.8</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0.4</td>
<td>0.7</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

(b) Our proposal: Using frequent itemsets as features

Table 5.2: Traditional biclustering-based classification uses the raw found biclusters for classification (or even a clusters of biclusters). Our approach uses not the biclusters as features, but the frequent patterns found in biclustering solutions. The rule interest metrics can be used to define the weigh of the features.

For this section, our objective is to illustrate the general capacities of this approach in the task of detecting schizophrenic patients. We used the biclustering algorithm CCC under traditional parameters as base. Then, for simplicity, we ignored the temporal links, and calculated frequent itemsets between regions. In this section we compared the performance of the following set of features:
1. Pearson correlation between regions,

2. Itemsets of size two considering regions (rescaled support),

3. Itemsets of size three considering regions (rescaled support),

We use the Random Forest classifier, since it is a classification approach, not only fast to train but known to be robust against overfitting [90]. This is a particular concern since we have a dataset with far more features than users. We set the number of trees empirically to 500. Classifier evaluation metrics were calculated using 10-fold Cross-Validation. As it is standard approach, the recall and precision were selected as evaluation metrics. Additionally, we selected the f1 measure as an alternative to evaluating the results. Our results are compared in figure 5.3, showing that our approach can be used to discriminate schizophrenic patients.

<table>
<thead>
<tr>
<th></th>
<th>Precision</th>
<th>Recall</th>
<th>f1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pearson</td>
<td>0.821 ± 0.042</td>
<td>0.868 ± 0.652</td>
<td>0.840 ± 0.026</td>
</tr>
<tr>
<td>Size 2 itemsets</td>
<td>0.936 ± 0.054</td>
<td>0.960 ± 0.031</td>
<td>0.947 ± 0.028</td>
</tr>
<tr>
<td>Size 3 itemsets</td>
<td>0.956 ± 0.037</td>
<td>0.950 ± 0.040</td>
<td>0.950 ± 0.018</td>
</tr>
</tbody>
</table>

(a) Train set results

<table>
<thead>
<tr>
<th></th>
<th>Precision</th>
<th>Recall</th>
<th>f1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pearson</td>
<td>0.588 ± 0.322</td>
<td>0.652 ± 0.215</td>
<td>0.530 ± 0.158</td>
</tr>
<tr>
<td>Size 2 itemsets</td>
<td>0.637 ± 0.323</td>
<td>0.767 ± 0.263</td>
<td>0.588 ± 0.195</td>
</tr>
<tr>
<td>Size 3 itemsets</td>
<td>0.620 ± 0.390</td>
<td>0.625 ± 0.310</td>
<td>0.520 ± 0.236</td>
</tr>
</tbody>
</table>

(b) Test set results

Table 5.3: Results from our biclustering classification, showing that our approach have a similar performance than a pearson-based discriminator.

First, it must be said that our results show a clear overfitting. This is natural, and it is a direct consequence of the dataset size. These approaches generate a high number of features (around a few thousands), which is this too much when compared to the number of rows (that was around a few dozens). Results from size 2 itemsets have better performance when compared with pearson results. However, the performance for size 3 itemsets becomes worse. This could be a consequence of the even higher number of features. This is a very importance concern, since the number of possible features will grow greatly with the size of the itemsets (despite slower than a combinatorial approach). In neuroscience, collecting a dataset with a high number of subjects is expensive, so it could mean that this approach is not adequate for data originating from fMRI data. Therefore this approach should be tested on other application domains.

To deal with the high number of generated itemsets, there are multiple possibilities. Initial filtering on the bicluster level can be conducted using a relevance criteria. A simple approach is to reduce the
minimum support for an item to be considered frequent. Another is to search for maximal or closed item-
sets. Feature selection can also be done, even though Random Forests perform faster, their performance 
suffers from a huge number of features.

In a final consideration, this section is simply illustrative of the concept, and further research must be 
conducted to evaluate the results.

5.9 Conclusions and Future Prospects

Evaluating the solution generated by a biclustering algorithm considering only the individual quality of 
the generated biclusters is equivalent to describing a tapestry as a set of threads while ignoring how they 
interconnect. While some information can be retrieved, it fails to describe the tapestry patterns, which is 
a beautiful part.

This chapter considers a bicluster not as independent solutions from a biclustering algorithm, but as 
a group of objects from which relations can be extracted. To extract these relations, pattern mining was 
used to filter the most frequent relations between items. The objective of this chapter was to provide an 
overview of the potentialities of this type of approach. Therefore, any of the sections of this chapter can 
be expanded and adapted to obtain improved results.

This chapter approaches an answer to a question rarely present in the literature: “What to do with the 
biclustering results?” Because most of the biclustering literature consists of proposing new algorithms 
(instead of analyzing results in a practical approach), this question is rarely explored.

The correlation analysis results suggest that biclustering is a valid approach to analyze fMRI data. 
The similarity between the pairwise correlation coefficient obtained considering the Pearson correlation 
coefficient, and our approach is high, suggesting that biclustering captures some interesting activity be-
tween brain regions. Additionally, the mixed results partially reproduced the original dataset, suggesting 
that the biclustering algorithm effectively captures brain activity.

The results from CCC and BicPAM are highly correlated. In the previous chapter, these were the two 
biclustering algorithms that achieved better results considering the virtual error, and now we conclude 
that the results are somewhat similar. This is an additional hint that it is possible to extract neuronal 
activity using biclustering algorithms.

These correlation results consider only correlations among pairs of items. Increasing the number of 
items means that we are modeling not a graph but a hypergraph. Here, the biclustering correlations allow 
reducing the number of nodes in comparison to a hypergraph algorithm. Even CCC with a high diversity 
of generated hypernodes has a decrease of about 40% considering hypernodes of size 7.

This chapter’s first great challenge is related to the computational cost of executing pattern mining 
searches, leading to some simplifications to reduce the computational time of obtaining frequent itemsets. 
These simplifications were done in three aspects: first, focusing only on the regions, reducing the num-
ber of nodes of the biclustering solutions significantly; second, increasing the minimum support; third,
considering a low maximum length of itemsets.

These approaches worked in this chapter since its main objective was to motivate the possible approaches that can be used. However, this aspect must be considered in future research, particularly in larger datasets, where the pattern mining cost could be high. This is of particular interest since, when trying to model hypergraphs using biclustering, it is key to prevent that the cost of executing biclustering and pattern mining is higher than the cost of executing a traditional hypergraph algorithm.

In neurosciences, it is of particular interest to study the brain patterns for most subjects (and not results from a single patient). For this, triclustering must be used. Therefore a roadmap is presented to adapt a biclustering algorithm into a triclustering one. This is useful since it allows us to use a good biclustering algorithm as a basis for a triclustering one.

This approach has several challenges. The first one is related to the pattern mining step. There is no guarantee that the patterns will include stamps from both original dimensions. For example, it could be a pattern of three regions, and we have little control over that approach. However, this is a challenge that even biclustering algorithms have (for example, FABIA can generate biclusters consisted of zero columns as part of his search method). Then there is the efficiency step, due to the computational cost of conducting the first biclustering, then pattern mining over the results, and finally joining the results to obtain triclusters. In the end, a triclustering method must be compared versus others, and only then a quantitative conclusion can be achieved.

The final section of this chapter was regarded as a classification. Traditional biclustering classification used the biclusters directly as input. Typically, having a bicluster pattern or similarity to the bicluster was used as feature. This method of generating biomarkers can be adapted to use not the biclustering presence but for a metric of overlap. Our approach used, not the biclusters but the frequent itemsets associated with them. However, our classification study has several problems, that need further investigation.

Extracting knowledge from biclustering solutions is a fuzzy task. Most of the research is focused on providing new efficient biclustering algorithms than contributing to methodologies to extract knowledge from the solutions. This chapter discussed a different approach to analyze, not biclusters, but biclustering solutions. Our results were in line with what was expected from other traditional methods, showing that fMRI time series data is a valid research domain considering biclustering.
Chapter 6

Conclusions and Future Work

Biclustering is an unsupervised learning technique, introduced by Cheng and Church, applied to the particular context of gene expression data. Since then, it was applied in diverse tasks from diverse application contexts. Since then, dozens of new algorithms were proposed, and the methodology for proposing a new biclustering algorithm is well defined in the literature. An algorithm is proposed, and a comparative study, including other biclustering algorithms, follows. However, the current methodology for applying biclustering is still fuzzy, in particular, when the study explores new scientific application domains.

The Ising model was formulated in 1925 to model the magnetic dipole moments of atomic spins and grew far beyond the embryonic context in physics. Today it is used in totally different fields such as neurosciences, econometrics, and machine learning, illustrating the capacity of an approach to generalize. Biclustering should be no different, and this thesis explores a full methodology to explore a biclustering analysis beyond its traditional data context.

Previous applications of simultaneous clustering on Spatio-temporal data, as well with the presence of bicluster-like structures in fMRI times series, motivate the use of biclustering on this type of data, thus we chosen it as an application context.

First question, **How to select a biclustering algorithm to use.** With dozens of biclustering algorithms available, choices must be made. One option is to use real-world knowledge to guide the selection. Another is to use the research focused on comparing the performance of a biclustering algorithm. However, this research is biased by the use of both gene expression benchmark datasets and the use of measures of biological relevance, such as the Gene Ontology annotations. This thesis begins by proposing a methodology to evaluate biclustering, concluding that exhaustive biclustering methods tested are able to achieve the best values of coherency of all tested algorithms. In particular, CCC is able to discover temporally contiguous biclusters. These biclusters have an easy interpretation since they are essentially multidimensional time series motifs.

Second question, **How to analyze the biclustering results.** Understanding one bicluster can be done considering line plots or heatmap. The problem is for biclustering methods able to generate hundreds of biclusters, often overlapping ones. Quantitative measures allow to compare one bicluster to another.
and obtain that the measure considers being the “best bicluster”. However, we do believe that these approaches lack the capacity to interpret a biclustering solution. For that, pattern mining algorithms operate on the biclustering solution to extract frequent relations among region and temporal dimensions. While single biclusters can be seen as a hypernode in a hypergraph, extracting frequent patterns allows to not only reduce the number of possible hypernodes but also allows to weight the nodes using both symmetric and asymmetric measures such as support and confidence.

Final question and strongly related to the previous one. **What to do with the biclustering solutions**, which is maybe the essential question in any application study, and one that is sparsely discussed since biclustering is typically connected with other statistical methods, which depend on an application context. To tackle this question, this thesis discussed the capacity of bicluster to conduct analyse on multiple subjects. This can be done considering unsupervised data, and is known as triclustering. For this, we present a roadmap to generate triclustering algorithms based on the frequent patterns extracted from the biclustering solutions. Additionally, biclustering can be used for supervised tasks, and this thesis discussed its use on discriminating schizophrenic patients.

This thesis expands the capacities of biclustering, showing that it is equally promising in fMRI data. fMRI belong to the Spatio-temporal data domain and share fundamental properties with areas such as climate science, epidemiology, and sociology. This are scientific fields expected to benefit from this study.
References


74


75


77


81


[143] J. Xie, A. Ma, A. Fennell, Q. Ma, and J. Zhao. It is time to apply biclustering: a comprehensive review of biclustering applications in biological and biomedical data. *Briefings in bioinformatics*, 20, 02 2018. doi: 10.1093/bib/bby014. 1, 2, 12, 23, 24, 33, 49


Appendix A

Appendix GitHub Repository

GitHub repositories were create to store data, code and results produced in the context of this thesis.

A.1 Biclustering fMRI Time Series: A Comparative Approach

- GitHub Repository Link

A.2 Improving Biclustering Interpretability: A Pattern Mining Approach

- GitHub Repository Link