UNICAMP

Saullo Haniell Galvão de Oliveira

On bicluster aggregation and its benefits for enumerative solutions. Aglomeração de biclusters e seus benefícios para soluções enumerativas.

Campinas

2015

UNIVERSIDADE DE CAMPINAS
Faculdade de Engenharia Elétrica e de Computação

Saullo Haniell Galvão de Oliveira

On bicluster aggregation and its benefits for enumerative solutions.

Aglomeração de biclusters e seus benefícios para soluções enumerativas.

Master dissertation presented to the Electrical Engineering Graduate Program of the School of Electrical and Computer Engineering of the University of Campinas to obtain the M.Sc. grade in Electrical Engineering, in the field of Computer Engineering.
Dissertação de mestrado apresentada a Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas para obtenção do título de Mestre em Engenharia Elétrica, na área de Engenharia de Computação.

Orientador: Prof. Dr. Fernando José Von Zuben

Este exemplar corresponde à versão final da dissertação defendida pelo aluno Saullo Haniell Galvão de Oliveira, e orientada pelo Prof. Dr. Fernando José Von Zuben

Campinas

2015

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca da Área de Engenharia e Arquitetura
Elizangela Aparecida dos Santos Souza - CRB 8/8098

OL4o

Oliveira, Saullo Haniell Galvão de, 1988-
On biclusters aggregation and its benefits for enumerative solutions / Saullo Haniell Galvão de Oliveira. – Campinas, SP : [s.n.], 2015.

Orientador: Fernando José Von Zuben.
Dissertação (mestrado) – Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Aprendizado de máquina. 2. Análise de Cluster. 3. Mineração de dados (Computação). 4. Outliers (Estatística). 5. Problemas de enumeração combinatória. I. Von Zuben, Fernando José. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.

Informações para Biblioteca Digital

Título em outro idioma: Agregação de biclusters e seus benefícios para soluções enumerativas
Palavras-chave em inglês:
Machine learning
Cluster analysis
Data mining and knowledge discovery
Outliers (statistics)
Combinatorial enumeration problems
Área de concentração: Engenharia de Computação
Titulação: Mestre em Engenharia Elétrica
Banca examinadora:
Fernando José Von Zuben [Orientador]
Ricardo José Gabrielli Barreto Campelo
Guilherme Palermo Coelho
Fabrício Olivetti de França
Romis Ribeiro de Faissol Attux
Data de defesa: 27-02-2015
Programa de Pós-Graduação: Engenharia Elétrica

COMISSÃO JULGADORA - TESE DE MESTRADO

Candidato: Saullo Haniell Galvão de Oliveira

Data da Defesa: 27 de fevereiro de 2015

Título da Tese: "On Bicluster Aggregation and its Benefits for Enumerative Solutions (Aglomeração de Biclusters e Seus Benefícios para Soluções Enumerativas)"

Prof. Dr. Fernando José Von Zuben (Presidente): Fernando José Von Zuben
Prof. Dr. Ricardo José Gabrielli Barreto Campello: Ricardo Campello
Prof. Dr. Guilherme Palermo Coelho:

Abstract

Biclustering involves the simultaneous clustering of objects and their attributes, thus defining local models for the two-way relationship of objects and attributes. Just like clustering, biclustering has a broad set of applications, ranging from an advanced support for recommender systems of practical relevance to a decisive role in data mining techniques devoted to gene expression data analysis. Initially, heuristics have been proposed to find biclusters, and their main drawbacks are the possibility of losing some existing biclusters and the incapability of maximizing the volume of the obtained biclusters. Recently efficient algorithms were conceived to enumerate all the biclusters, particularly in numerical datasets, so that they compose a complete set of maximal and non-redundant biclusters. However, the ability to enumerate biclusters revealed a challenging scenario: in noisy datasets, each true bicluster becomes highly fragmented and with a high degree of overlapping, thus preventing a direct analysis of the obtained results. Fragmentation will happen no matter the boundary condition adopted to specify the internal coherence of the valid biclusters, though the degree of fragmentation will be associated with the noise level. Aiming at reverting the fragmentation, we propose here two approaches for properly aggregating a set of biclusters exhibiting a high degree of overlapping: one based on single linkage and the other directly exploring the rate of overlapping. A pruning step is then employed to filter intruder objects and/or attributes that were added as a side effect of aggregation. Both proposals were compared with each other and also with the actual state-of-the-art in several experiments, including real and artificial datasets. The two newly-conceived aggregation mechanisms not only significantly reduced the number of biclusters, essentially defragmenting true biclusters, but also consistently increased the quality of the whole solution, measured in terms of *Precision* and *Recall* when the composition of the dataset is known a priori.

Keywords: Biclustering; bicluster enumeration, bicluster aggregation, outlier removal, metrics for biclusters.

Resumo

Biclusterização envolve a clusterização simultânea de objetos e seus atributos, definindo modelos locais de relacionamento entre os objetos e seus atributos. Assim como a clusterização, a biclusterização tem uma vasta gama de aplicações, desde suporte a sistemas de recomendação, até análise de dados de expressão gênica. Inicialmente, diversas heurísticas foram propostas para encontrar biclusters numa base de dados numérica. No entanto, tais heurísticas apresentam alguns inconvenientes, como não encontrar biclusters relevantes na base de dados e não maximizar o volume dos biclusters encontrados. Algoritmos enumerativos são uma proposta recente, especialmente no caso de bases numéricas, cuja solução é um conjunto de biclusters maximais e não redundantes. Contudo, a habilidade de enumerar biclusters trouxe mais um cenário desafiador: em bases de dados ruidosas, cada bicluster original se fragmenta em vários outros biclusters com alto nível de sobreposição, o que impede uma análise direta dos resultados obtidos. Essa fragmentação irá ocorrer independente da definição escolhida de coerência interna no bicluster, sendo mais relacionada com o próprio nível de ruído. Buscando reverter essa fragmentação, nesse trabalho propomos duas formas de agregação de biclusters a partir de resultados que apresentem alto grau de sobreposição: uma baseada na clusterização hierárquica com *single linkage*, e outra explorando diretamente a taxa de sobreposição dos biclusters. Em seguida, um passo de poda é executado para remover objetos ou atributos indesejados que podem ter sido incluídos como resultado da agregação. As duas propostas foram comparadas entre si e com o estado da arte, em diversos experimentos, incluindo bases de dados artificiais e reais. Essas duas novas formas de agregação não só reduziram significativamente a quantidade de biclusters, essencialmente defragmentando os biclusters originais, mas também aumentaram consistentemente a qualidade da solução, medida em termos de precisão e recuperação, quando os biclusters são conhecidos previamente.

Palavras-chaves: Biclusterização, enumeração de biclusters, agregação de biclusters, remoção de *outliers*, métricas para biclusters.

Contents

I Main Concepts 5

II Proposals of this dissertation 27

Dedicado a Irglis, Shayra, Junior, Carlos e Nelcy (em memória).

Agradecimentos pessoais

Antes de tudo sou grato a minha mãe, Irglis, minha irmã, Shayra, meu irmão, Junior, e ao meu padastro, Carlos. Foi com eles que aprendi a ter motivação e coragem para enfrentar qualquer tipo de desafio, mesmo os que me parecem além da minha capacidade. São meus grandes companheiros de vida. Ao meu cunhado Vanderlei e minha sobrinha Julia, que apesar de tão nova, inspira muito meus dias! Aprenda logo a falar titio e deixe de ser "brava"!

Minha avó, Nelcy (*em memória*), também tem um papel fundamental no que sou hoje. Também incluo meu tio, Irgledson, tia Edelaine, tia Irglênia, tio Clóvis, meus primos e primas: Marcos, Igor, Elen e Matheus.

Agradeço ao meu pai, Claudemir, sua esposa, Madalena, e meus irmãos: Yago e Yasmin. Apesar da distância, vocês também têm parte nessa conquista.

Aos meu grandes amigos do ensino médio: Renan, Josemir, Igor, Lee Marwin, Fellipe, Alessandra. Aprendo muito com vocês!

A Graça Tomazella, minha professora da graduação que me estimulou e desafiou bastante. A Dilermando Piva Jr. que me orientou no trabalho de conclusão de curso. Também agradeço meus grandes amigos da graduação: Luis Felipe e Wilson Santos.

Agradeço a um cara com quem trabalhei, que me incentivou muito a começar a carreira acadêmica e me ensinou muito sobre a dinâmica de poderes e influências no ambiente corporativo. Aí Salgado (caveira!), o aspira terminou o mestrado!

Aos amigos de Indaiatuba que me ensinaram uma lição ou outra em vários momentos distintos: Cris, Felipe, Viviane, Gi Guio, Marcinha, Eriton, Giselle, Murillo, Michele, Robson, Fabio, Neia. Talvez vocês nem saibam, mas foram e são muito importantes para mim.

Aos amigos que conheci no LBiC e na pós: Rosana, Alan, Andrea, André, Carlos, Flávia, Kamila, Salomão (*em memória*), Hamilton, Wilfredo, Kurka, Marcos, Thalita, Righetto, Micael, Raul, Eliezer, André Vergílio, Alan Caio e Salim. A vivência com vocês é fantástica! Obrigado pelas trilhas, churrasco, conversas, discussões, GRUDEs, reuniões do capítulo, reuniões da Apogeeu, etc. Obrigado por essa grande experiência! Kurka e Alan Caio, mais um obrigado pelas horas de estudo para as disciplinas. Alan, valeu pelas corridas e cervejas!

Rosana, não tenho palavras para agradecer por tudo que passamos e por sua influência. Foram muitas histórias, muitos contextos, com muita cumplicidade. Estendo os agradecimentos a sua família, seu Tico, dona Irene, ao grande Junior e a Renata. Todos vocês me ajudaram muito!

Agradeço aos amigos aqui de Barão Geraldo: Fran, Aldo e Danilo; e a amiga Rosana

Rogeri. Vocês me ajudaram muito na transição após minha mudança para Barão Geraldo, e me deram um apoio fundamental na fase final do meu mestrado!

Sou imensamente grato ao meu orientador, Fernando, por ter me oferecido a oportunidade de participar de sua equipe, e pela grande inspiração nas aulas e nos momentos de orientação. Sem dúvidas eu tive uma excelente inserção no ambiente acadêmico.

Em todo o meu trajeto, muitas pessoas incríveis passaram pelo meu caminho e me ajudaram de forma direta ou indireta. Infelizmente tenho que me contentar com uma memória imperfeita, que não fará justiça a todas elas. Peço desde já perdão.

O autor desse trabalho foi financiado pelo Centro Nacional de Desenvolvimento Científico e Tecnológico (CNPq).

"Essa realização teria sido impossível se eu tivesse querido me apegar com teimosia à minha origem e às lembranças de juventude."

(Um relatório para uma academia - Franz Kafka)

List of Figures

List of Tables

1 Introduction

Data mining is a popular research field that aims to detect hidden patterns in datasets and to extract knowledge of these analyses. Clustering is a data mining task that finds groups of highly correlated objects in a dataset. If two objects are in the same group, they are highly correlated; but if they are in distinct groups, they should not be correlated. The applications of clustering techniques are disseminated, varying from marketing purposes to outlier detection, going through protein identification, among others. The reader may refer to Jain *et al.* (1999) for a survey of clustering.

To be part of a cluster, an assumption is made: all attributes of the clustered objects must show certain correlation. If the process do not find a global correlation, the objects will not be part of the same cluster. This assumption is problematic for some applications and specially for objects with many attributes. For example, in microarray gene expression data analysis, it is very hard to find a global correlation between all the attributes. Usually, the objects of this kind of dataset are genes, and the attributes are samples of experiments. A cluster would represent a set of genes that exhibit a similar expression considering all the samples. But the samples can refer to distinct subjects. For example, one sample can come from a healthy tissue, another from a cancer tissue. If a gene is related to the manifestation of that cancer, it will certainly be expressed differently in the two samples when compared to a non-related gene. In this case, the clustering methods will not be able to find proper groups in these datasets.

Biclustering is a set of clustering algorithms capable of finding groups of objects in a subset of attributes. In this case, the group of objects only makes sense considering that specific subset of attributes. This class of algorithms rapidly found application in gene expression data analysis and several other fields. As finding all biclusters in a dataset is an NP-hard problem, most of the algorithm proposals are heuristics, that may miss important biclusters.

The enumeration of biclusters is something recent. Veroneze *et al.* (2014) proposed a family of bicluster enumeration algorithms for real datasets, that we will explore in this dissertation. In the literature, it is known that the presence of noise when enumerating a dataset leads to a result with too many biclusters with high overlapping (ZHAO; ZAKI, 2005). Even in small datasets, the quantity of enumerated biclusters can be enormous, leading to a complex and timing consuming analysis, or even impracticable. In this case, the aggregation plays a fundamental role in removing the unnecessary overlapping and simplifying the final biclusters of the solution.

An area of research similar to bicluster aggregation is the biclustering ensemble, which is a combination of different results into a more robust final result. Ensemble is a common practice in classification and regression tasks, and is gaining attention in clustering tasks. The literature presents a variety of biclustering ensemble algorithms, but the aggregation is a subtly different task, to be better explained along the text. Despite that difference, we compared our proposals with the results produced by an ensemble algorithm, as well as with the most similar approach to an aggregation algorithm that we were able to find in the literature. After aggregating fragmented biclusters, an outlier removal step is conceived to filter out intruder objects and / or attributes that supposedly were incorporated in the aggregated solution.

We will show that the aggregation of biclusters based on the high overlapping of the enumerative solution, can lead to better results, severely reducing the quantity of biclusters. This conclusion was obtained considering three artificial datasets and two real datasets from different backgrounds.

1.1 Main goal of the research

The goal of this research is to remove the redundancy and improve the quality of a bicluster solution that presents a high degree of overlapping among its components. This characteristic is easily found in enumerative solutions of biclustering.

1.2 Structure of the dissertation

This dissertation is divided into five parts.

Part I - Main Concepts In this part we will explain the necessary background to support our contribution. This part contains three chapters. The first chapter will discuss clustering and biclustering methods. The second chapter will discuss the ensemble methods for biclustering, and we will highlight the difference between bicluster ensemble and bicluster aggregation. The third chapter will discuss the metrics for biclustering evaluation that best fit our needs.

Part II - Proposals In this part we will explain our contributions. We will start by explaining our first contribution: an aggregation based on hierarchical clustering with single linkage. After that, we will explain a second proposal: an aggregation based on the overlapping between the biclusters. We will continue by explaining an outlier removal step that proved to be important on our contributions, and we will end this chapter by giving an example of the entire aggregation procedure.

Part III - Discussion In this part we will explain how the experiments were planned and delimited. We will also explain how the artificial datasets were generated, discuss the results of the experiments, and show how our proposal can be positioned in comparison with the existent methods.

Part IV - Final considerations This chapter concludes this dissertation and outlines the future work that can be done starting from the achieved contributions.

Part I

Main Concepts

2 Clustering and biclustering

In this chapter we will explain the limitations of clustering that motivated the development of biclustering algorithms and how these algorithms look for patterns in a dataset.

2.1 Clustering

Clustering is a well-known data mining task, that groups objects from a dataset into clusters. Ideally the similarity between objects on the same cluster is high, and the similarity between objects from distinct clusters is low. The clustering usefulness led this problem to be studied in many contexts and by researchers in many disciplines. Some important applications for clustering are market research, astronomy, psychiatry, weather classification, archeology, bioinformatics, among others (EVERITT *et al.*, 2011). For a broad view of the area, we recommend the surbvey by Jain *et al.* (1999).

2.2 Clustering with Single Linkage

Single linkage is a method of agglomerative hierarchical clustering. In this class of methods, each object starts in its own cluster. The clusters are then sequentially combined with the closest cluster specified by a pre-defined distance, up to the point where all objects belong to the same cluster.

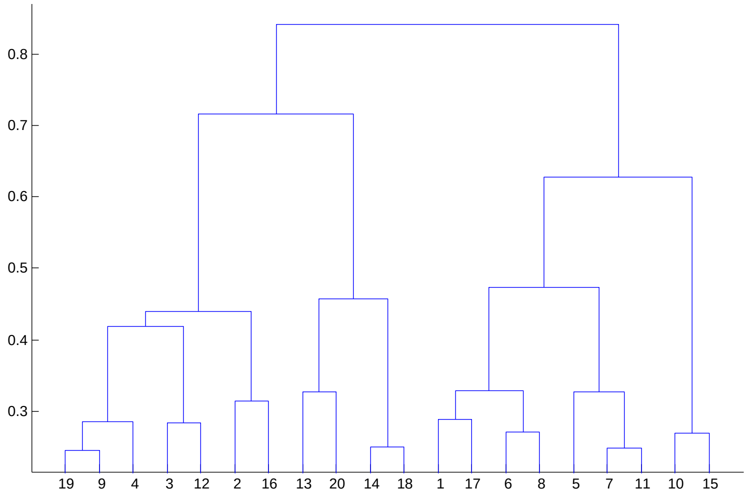


Figure 1 – Example of a dendrogram.

It is common to represent the process of hierarchical clustering in a visual form using a dendrogram, as in Figure 1. The leafs of the dendrogram represent the objects, each junction represents a cluster and the height of a junction represents the distance between the two combined clusters.

The term "single linkage" refers to how the distance between the clusters will be calculated (JAIN *et al.*, 1999). In this case, the distance is measured by the closest elements of the two groups in comparison.

A drawback of the hierarchical clustering is that the user must choose where to cut the dendrogram. The cut determines how many clusters the solution will have. Usually the height of the junctions is used to indicate a good cut: when the height of a junction is much bigger than the height of the junctions below, it may be a good place to cut the dendrogram. Another drawback of single linkage is what is known as the chaining effect. If the clusters are connected by some elements, the process may not find useful clusters (JAIN *et al.*, 1999).

2.3 Biclustering Motivation

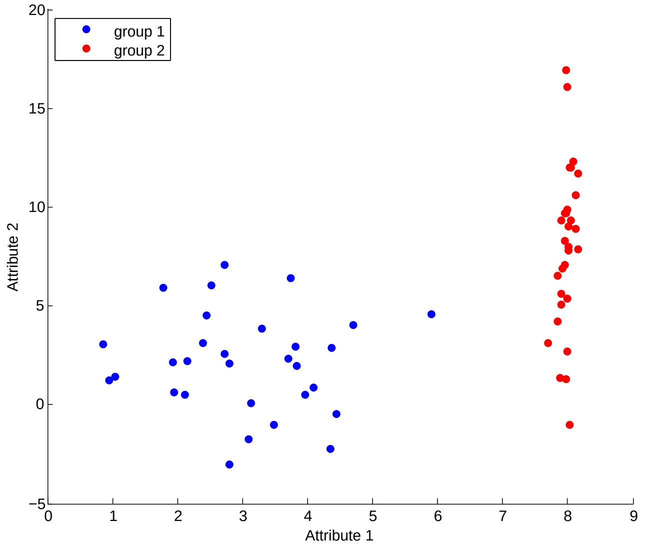


Figure 2 – Example of clustering considering objects composed of two attributes.

Figure 2 shows an example of clustering. We can see that the two groups are well-defined in distinct areas of the feature space. However, even in this limited dimensionality, we can verify one major limitation of clustering: the assumption of correlation in all attributes. Note that *Attribute 2* is not relevant for clustering group 2. We can see this problem in a practical example. Table 1 shows a toy dataset, where the rows represent movies, the columns represent customers and the table fields are evaluations that these customers gave to the movies. The first four movies are from the category "documentary" and the last four are from the category "action". If we consider all attributes, we will not be able to find any relevant group. But considering subsets of the attributes, we can easily find two distinct groups. This example leads us to biclustering.

Table 1 – Example of biclusters in a movie evaluation dataset.

	Maria	João	José	Sofia
Cosmos: A Space Time Odissey	5	5	2	5
Fed Up	5	4	1	5
Myth Busters	4	4	5	2
Catfish	5	5	5	1
X-men	2	5	4	5
Godzilla	4	1	5	4
Lone Survivor	2	1	5	5
Non-Stop	5	3	5	4

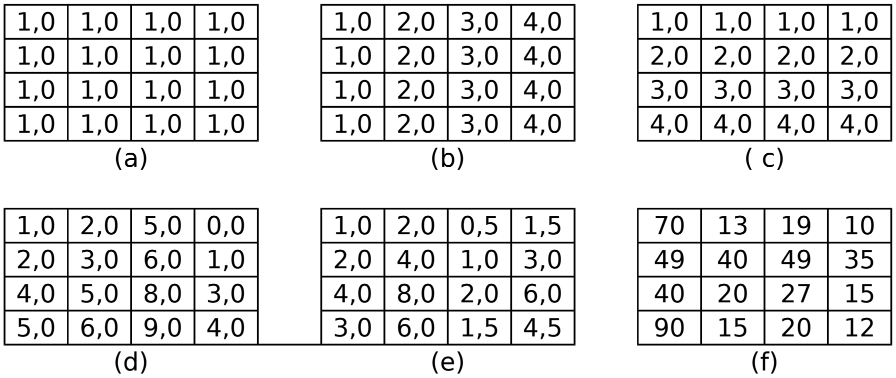
Biclustering is a set of clustering algorithms that simultaneously cluster the rows (objects) and columns (attributes) of a dataset. In this case, biclustering does not require correlation between all attributes, given that it finds the most related set of attributes for each set of objects. Hartigan (1972) proposed the first biclustering algorithm, and Cheng & Church (2000) coined the term, applying their proposal to gene expression data. In fact, biclustering is already considered a common technique to analyze gene expression data.

2.4 Problem formulation and definitions

Consider a dataset $\mathbf{A} \in \mathbb{R}^{n \times m}$, with rows $X = \{x_1, x_2, \ldots, x_n\}$ and columns $Y = \{y_1, y_2, \ldots, y_m\}$. We define a bicluster B by $B = (B^r, B^c)$, where $B^r \subseteq X$ and $B^c \subseteq Y$, such that the elements in the bicluster show a coherence pattern. A bicluster solution is a set of biclusters represented by $\bar{B} = \{B_i\}_{i=1}^{q}$, where q is the quantity of biclusters on the solution set. A bicluster is maximal if and only if we can not include any other object / attribute without violating the coherence threshold. The overlapping function between two biclusters B and C is defined as

$$ov(B, C) = \frac{|B^r \cap C^r \times B^c \cap C^c|}{min(|B^r \times B^c|, |C^r \times C^c|)}. \tag{2.1}$$

Madeira & Oliveira (2004) categorized the types of biclusters according to their coherence patterns. In Figure 3 we show only the most important types of biclusters, that are explained below.



1,0	1,0	1,0	1,0
1,0	1,0	1,0	1,0
1,0	1,0	1,0	1,0
1,0	1,0	1,0	1,0

(a)

1,0	2,0	3,0	4,0
1,0	2,0	3,0	4,0
1,0	2,0	3,0	4,0
1,0	2,0	3,0	4,0

(b)

1,0	1,0	1,0	1,0
2,0	2,0	2,0	2,0
3,0	3,0	3,0	3,0
4,0	4,0	4,0	4,0

(c)

1,0	2,0	5,0	0,0
2,0	3,0	6,0	1,0
4,0	5,0	8,0	3,0
5,0	6,0	9,0	4,0

(d)

1,0	2,0	0,5	1,5
2,0	4,0	1,0	3,0
4,0	8,0	2,0	6,0
3,0	6,0	1,5	4,5

(e)

70	13	19	10
49	40	49	35
40	20	27	15
90	15	20	12

(f)

Figure 3 – Types of biclusters. (a) Constant bicluster, (b) Constant columns, (c) Constant rows, (d) Coherent values (additive model) (e) Coherent values (multiplicative model) (f) Coherent evolution.

Constant bicluster: This type of bicluster is represented by $a_{ij} = \mu + \omega_{ij}$, where μ is a constant value, and ω_{ij} is a level of noise associated with the entry a_{ij}. Figure 3a shows an example where $\omega_{ij} = 0, \forall i, j$.

Constant columns: This type of bicluster is represented by $a_{ij} = \mu + \beta_j + w_{ij}$ or $a_{ij} = \mu \times \beta_j + w_{ij}$, where w_{ij} is a level of noise associated with the entry a_{ij}. For the perfect case, Figure 3b shows an example. When the bicluster is not perfect, $\exists i, j$ such that $w_{ij} \neq 0$.

Constant rows: This type of bicluster is represented by $a_{ij} = \mu{+}\alpha_i{+}w_{ij}$ or $a_{ij} = \mu{\times}\alpha_i{+}w_{ij}$, where w_{ij} is a level of noise associated with the entry a_{ij}. Figure 3c shows an example.

Coherent values: This type of bicluster is represented by $a_{ij} = \mu + \alpha_i + \beta_j + w_{ij}$, on the additive case; and $a_{ij} = \mu{\times}\alpha_i{\times}\beta_j{+}w_{ij}$ on the multiplicative case; where w_{ij} is a level of noise associated with the entry a_{ij}. Figures 3d and 3e represent these cases, respectively. We can see that one row (column) plus (times) some constant value produces another row (column).

Coherent evolution: This type of bicluster is defined according to the order of the values, not depending explicitly on the values themselves. We can see an example in Figure 3f, where the columns show the following pattern: $a_{i4} < a_{i2} < a_{i3} < a_{i1}$, for $i = \{1, 2, 3, 4\}$. Coherent evolution is then a generalization of coherent values.

We highlight that biclusters with constant values, constant values on rows or constant values on columns, are special cases of biclusters with coherent values –with $\alpha = 0$ or $\beta = 0$ –and we will focus our attention on the latter.

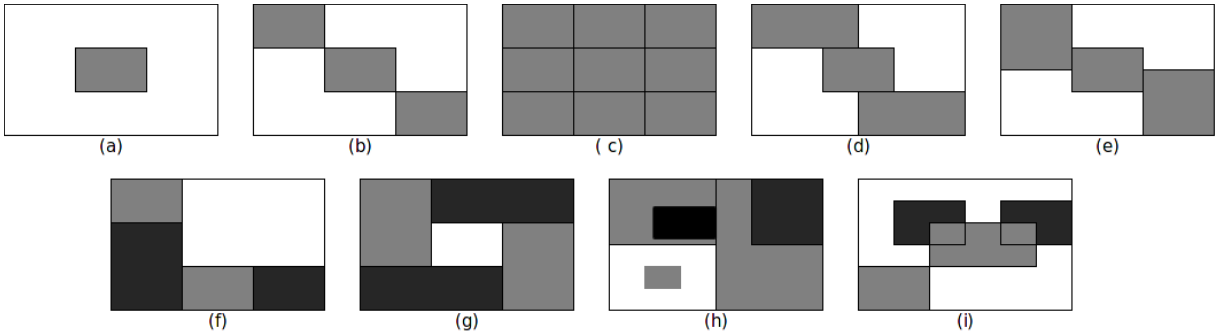


Figure 4 – Bicluster structure. (a) single bicluster, (b) exclusive row and column biclusters, (c) checkerboard structure, (d) exclusive rows biclusters, (e) exclusive columns biclusters, (f) non-overlapping biclusters with tree structure, (g) non-overlapping non-exclusive biclusters, (h) overlapping biclusters with hierarchical structure, and (i) arbitrarily positioned overlapping biclusters.

Besides all these types, a dataset can dispose their biclusters in many structures. Figure 4 shows several examples of structures of biclusters, as in Madeira & Oliveira (2004). The checkerboard structure and the arbitrarily positioned overlapping are the most explored in the literature, as they are the most common in real scenarios. Figure 4 shows only contiguous biclusters, but this is not a necessary condition, as $B^r \subseteq X$ and $B^c \subseteq Y$, and non-contiguous biclusters are very common. This work focuses on maximal biclusters of the structure depicted in Figure 4i, also considering contiguous or non-contiguous biclusters. A bicluster is considered maximal when no row or column can be included without violating the chosen coherence pattern.

2.5 Some heuristics

As finding all biclusters in a dataset is an NP-hard problem (MADEIRA; OLIVEIRA, 2004), several methods resort to heuristics. Cheng & Church (2000) proposed the CC algorithm, one of the most famous methods for biclustering. They look for a bicluster per execution and the volume of the bicluster directly influences the search, as long as the internal values do not trespass a residue value δ. Eq. 2.2 defines the residue of a bicluster B:

$$H(B) = \frac{1}{|B^r||B^c|} \sum_{i \in B^r, j \in B^c} (a_{ij} - a_{iJ} - a_{Ij} + a_{IJ})^2, \tag{2.2}$$

where a_{iJ} is the row mean, a_{Ij} is the column mean, a_{IJ} is the bicluster mean. When a bicluster is perfect (constant bicluster, constant on rows / columns, or coherent value), its residue is zero. Also, when the algorithm finds a bicluster, it replaces their values with random numbers.

Jiong *et al.* (2003) proposed the FLOC (FLexible Overlapped biClustering), another important heuristic. They based their algorithm on CC but avoiding the step of replacing the values of the found bicluster with random numbers. Moreover, FLOC also identifies more than one bicluster simultaneously.

The main limitation of these heuristics is that they are not able to guarantee finding all the biclusters. Also, the biclusters that are found may not be of maximal volume, in the sense that some rows and columns may be missed.

For a comprehensive survey of bicluster algorithms, the reader may refer to Madeira & Oliveira (2004) and Tanay *et al.* (2005).

2.6 Bicluster enumeration

The bicluster enumeration is accomplished by a class of biclustering algorithms that performs an exhaustive search for all maximal biclusters in a dataset, given a desired coherence pattern.

In the case of binary datasets, there are plenty of algorithms for enumerating all maximal biclusters. Some examples are Makino & Uno (MAKINO; UNO, 2004) and In-Close2 (SIMON, 2009). The enumeration of all maximal biclusters in an integer or real-valued dataset is a much more challenging scenario, but we already have some proposals, such as RIn-Close (VERONEZE *et al.*, 2014), and RAP (PANDEY *et al.*, 2009). In *subspace clustering*, where biclustering is called *clustering by pattern similarity*, some algorithms have an enumerative approach to mine coherent values biclusters (VERONEZE *et al.*, 2014). As an example we have pCluster, proposed by Wang *et al.* (2002). Some shortcomings of this algorithm pointed by Veroneze *et al.* (2014) are: *a*) it does not find all biclusters; *b*) it finds biclusters that do not meet the user-defined measure of similarity.

An extension of pCluster is MaPle, proposed by the same authors (PEI *et al.*, 2003). In order to return just maximal biclusters, for each newly found bicluster MaPle looks at all previously found biclusters, increasing the computational cost of this algorithm. Veroneze *et al.* (2014) also pointed some scenarios where MaPle is not able to properly enumerate all biclusters of a dataset. In other words, pCluster and MaPle are not enumerative, although it was suggested by the authors.

Zhao & Zaki (2005) proposed MicroCluster. The algorithm starts by building a multigraph, then mines the maximal biclusters from this multigraph. Veroneze *et al.* (2014) pointed that MicroCluster still miss some biclusters and thus is not enumerative. This algorithm has an agglomerative step that will be used in this work and will be explained in more details later. The first proposal able to achieve the enumeration of coherent values biclusters in a numerical dataset is the work of Veroneze *et al.* (2014). They proposed RIn-Close, a family of algorithms that perform a complete, correct and non-redundant enumeration of biclusters. By complete we mean that it returns all maximal biclusters present on the dataset; by correct we expect that it returns only biclusters that attend the informed coherence pattern; and by non-redundant we mean that it does not return the same bicluster more than once.

The parametrization of RIn-Close is also very important, as it is directly correlated with the number of enumerated biclusters, and runtime. The authors draw attention to the fact that even in a small dataset, depending on the parametrization, the algorithm may return a large amount of biclusters. For more details about the actual state of research on bicluster enumeration of several different types, the reader may refer to Veroneze *et al.* (2014).

In this work, we propose a way of aggregating biclusters from a biclustering result. It is important that the obtained biclusters present high overlapping, as it is the case when enumerating biclusters in noisy datasets. For this reason, in this work we will focus on enumerative results. We will use the RIn-Close family, as the other options present drawbacks pointed here and by Veroneze *et al.* (2014).

3 Ensemble and Aggregation

Ensemble is a common practice in supervised learning, which consists of combining the results from several components into a single result of better performance and more robust to noise (SHARKEY, 1996). According to Lima (2004), "it is about getting a result, for a classification / regression problem, from several results of multiple alternative solutions for the same problem, called components of the ensemble" [1]. It is important for the components to show a good individual performance, and also diverge in the error (PERRONE; COOPER, 1993). In other words, the components should fail in distinct aspects of the problem, and hopefully for distinct samples of the dataset. In this way, where a component has a bad performance, other ones can perform better.

Recently the ensemble setting started to be extended to unsupervised learning, such as clustering. The problem of cluster ensemble is more difficult than classification ensemble, as the labels of the clusters are hypothetical and we have a correspondence problem. Besides that, the quantity and the shape of the clustering solutions may vary, according to the assumptions of the algorithms, and to the view that each component has of the dataset (STREHL; GHOSH, 2002). From the motivations of cluster ensemble (GHOSH; ACHARYA, 2011; STREHL; GHOSH, 2002; STREHL *et al.*, 2000; WANG *et al.*, 2011), we highlight:

Knowledge reuse: It is possible that some clustering solutions are already available from the dataset. Thus, we can use that information to influence a new solution. Also, discarding the previous knowledge can be wasteful.

Distributed computing: Some restrictions may rule the data, such as content privacy, geographic, technical, or even contractual restrictions. Thus, we can deal with each component independently and use the results to reach a consensus.

Content privacy: Some data may be protected by privacy restrictions or may belong to different companies or government organs. Move that data may not be possible. Thus, again, we can obtain each component independently and use the results to reach a consensus.

Robustness: Combining solutions from different algorithms or data views contributes to get a more robust result, as it does a better exploration of the hypothesis space (Note:

[1] "Trata-se da obtenção de uma saída, para um problema de classificação, ou de regressão, a partir das saídas individuais propostas por múltiplas soluções alternativas para o mesmo problema, denominadas componentes do ensemble".

by hypothesis space we mean the space of all models that can be proposed to represent the data).

For a comprehensive survey of cluster ensemble, the reader may refer to Vega-Pons & Ruiz-Shulcloper (2011). In this survey, the authors categorized cluster ensemble algorithms based on their consensus function, i.e. the function that combines the distinct results.

3.1 Bicluster ensemble

Since ensemble methods may promote the improvement of the performance of supervised classification and clustering, it is reasonable to think that they can also be extended to tackle the biclustering problem (HANCZAR; NADIF, 2011b). In fact, many approaches for biclustering ensemble already proposed in the literature, are based on the methods for clustering ensemble.

Usually the process of getting an ensemble is the same: initially, we generate some biclustering solutions, looking for diversity; then we use a consensus function to combine the previous solutions into a single one. A selection step may also be included before combining the candidate components. We will briefly describe the status of research on this topic.

Gullo *et al.* (2012) proposed an ensemble method in which they used distinct biclustering algorithms to generate diverse solutions, and modeled the consensus as a multiobjective optimization problem. They also proposed a method to choose the most promising solution on the Pareto front. The authors commented on the need to tune the parameters of the solutions to get a good result, as simpler approaches led to comparable results.

Hanczar & Nadif (2011b) used the bagging technique to get the components and combined the results using a “metacluster of biclusters”, based on Strehl & Ghosh (2002). Although promising, one needs to pay attention to the runtime of this proposal.

Huang *et al.* (2012) proposed a scalable biclustering ensemble method. They used the ITCC (Information-Theoretic Co-Clustering) algorithm to generate the components. The authors implemented the algorithm in a distributed way using the Hadoop MapReduce framework. They based the consensus on evidence accumulation (FRED; JAIN, 2005), and analyzed their algorithm on: text mining datasets, a comments authorship discovery task, and a Brazilian Sign Language (LIBRAS[2]) dataset.

Hanczar & Nadif (2011a) published promising results of biclustering ensemble on microarray gene expression data, improving the biological significance of the final result.

[2] Linguagem Brasileira de Sinais

They represented the biclusters in a binary matrix and then used a triclustering (HANCZAR; NADIF, 2012) technique to get the consensus. We will use this algorithm in this dissertation, as it was already applied to gene expression data analysis.

Aggarwal & Gupta (2013) divided the ensemble in two steps. The first is a label correspondence problem, where each bicluster receives a label and they search for a correspondence between the biclusters. The second is the consensus via optimization. The cost function considers the dissimilarity between objects and attributes. A drawback of this method is the assumption that all objects / attributes must be part of a bicluster, which may not be true and can lead to poor results (PIO *et al.*, 2013).

One major point in ensemble is that we want to combine the results reinforcing the biclusters that seem to be important for several components, and discarding the ones that may come from noise. If an area (set of objects and attributes) of the dataset is covered just by one bicluster while the other covered areas have much more biclusters, the ensemble should discard this bicluster, as it may be considered of low importance.

3.2 Bicluster aggregation

A major drawback of enumeration, particularly in the context of noisy datasets, is the existence of a large number of biclusters, due to fragmentation of a much smaller number of original biclusters, exemplified in Fig. 5, and verified in one of our experiments. The noise is responsible for fragmenting the original biclusters into many with high overlapping, so that the aggregation of these biclusters is recommended (LIU *et al.*, 2004; ZHAO; ZAKI, 2005). This fragmentation leads to a challenging scenario for the analysis of the results, that can become impractical even in small datasets.

The aggregation aims at recovering the true bicluster from its fragmented counterparts. Although this combination seems similar to bicluster ensemble, the problem is different. While on ensemble tasks we discard biclusters that seem unimportant and combine the ones that contribute most for the solution, in bicluster aggregation we never discard any bicluster.

We will explain some methods of bicluster aggregation already published in the literature.

3.2.1 MicroCluster aggregation

After the enumeration, Zhao & Zaki (2005) added two steps to their algorithm. These steps have the task of deleting or merging biclusters that are not covering an area much different from other biclusters. Consider two biclusters $B = (B^r, B^c)$ and $C = (C^r, C^c)$. We

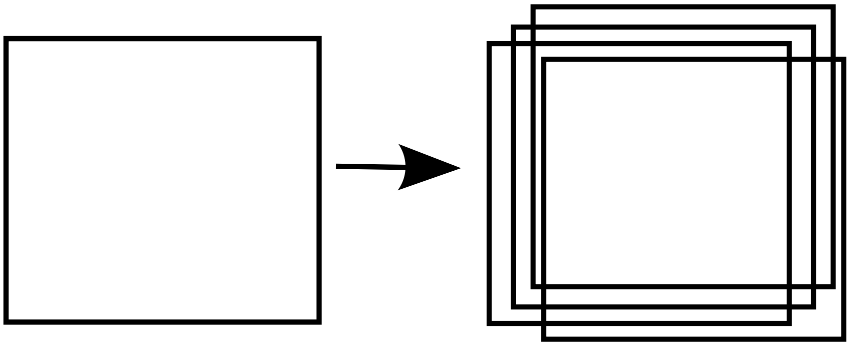


Figure 5 – Illustration of the fragmentation of a bicluster in a noisy dataset.

define the span of a bicluster B as the set of object-attribute pairs belonging to the bicluster, given as $L_B = \{(g, s) \mid g \in B^r, s \in B^c\}$, and $L_C = \{(g, s) \mid g \in C^r, s \in C^c\}$. Then we can define these derived spans:

$$L_{B \cup C} = L_B \cup L_C \tag{3.1}$$

$$L_{B-C} = L_B - L_C \tag{3.2}$$

$$L_{B+C} = L_{(B^r \cup C^r) \times (B^c \cup C^c)}. \tag{3.3}$$

With these definitions, MicroCluster deletes the bicluster B if a set of biclusters $\{C_i\}$ exists such that

$$\frac{L_B - L_{\cup_i C_i}}{L_B} < \eta. \tag{3.4}$$

Notice that the set $\{C_i\}$ can have just one bicluster.

This step can be better understood looking at the left side of Figure 6. We can see that two black biclusters are overlapping a red bicluster. If the ratio of the gray area by the area of the red bicluster is less than the parameter η, we remove the red bicluster.

If

$$\frac{|L_{B+C-B-C}|}{|L_{B+C}|} < \gamma \tag{3.5}$$

holds, we merge B and C into one bicluster $D = (B^r \cup C^r, B^c \cup C^c)$. At the right side of Figure 6, if the ratio of the gray area by the area covered by the two black biclusters is less than the parameter γ, we merge the two biclusters by the union of their objects / attributes.

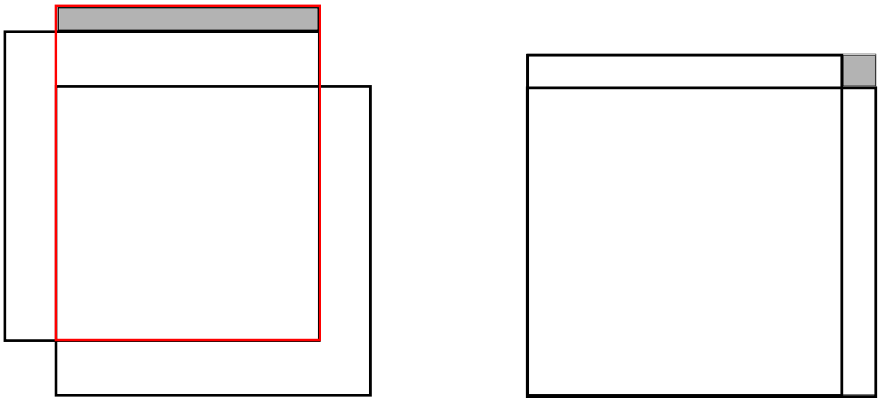


Figure 6 – Examples of the delete / merge of the MicroCluster algorithm.

3.2.2 CoClusLSH Aggregation

Gao & Akoglu (2014) used the principle of Minimum Description Length to propose CoClusLSH, an algorithm that returns a hierarchical set of biclusters. The hierarchical part can be seen as an aggregation step. This step is done based on the LSH technique as a hash function. Candidates hashed to the same bucket are then aggregated until no merge improves the final solution. Their work is focused on finding biclusters in a checkerboard structure, that does not allow overlapping, thus being not suitable for the focus of our research.

3.2.3 OPC-Tree Aggregation

Liu *et al.* (2004) proposed OPC-Tree, a deterministic algorithm to mine Order Preserving Clusters (OP-Clusters), a general case of OPSM type of biclusters. They also have an additional step for creating a hierarchical aggregation of the encountered OP-Clusters. The Kendall coefficient is used to determine which clusters should be merged and in which order the objects should participate in the resultant OP-Cluster. The highest the Rank Correlation using the Kendall coefficient, the highest the similarity between two OP-Clusters. The merging is allowed according to a threshold that is reduced in a level-wise way. It is important to highlight that this work considers the order of the rows in the cluster. In this case, a perfect coherent values bicluster keeps the order of its rows and the hierarchical step of OPC-Tree would be able to be used in this case as well. But we are considering noisy datasets, in which this assumption probably will not hold, and thus, the hierarchical step of OPC-Tree is not suitable for the problem we are dealing with.

To the best of our knowledge, these are the proposals in the literature that are most similar to the problem we are dealing. These algorithms will be better explored in Chapter 6, where they will be compared with our contributions.

To evaluate our results we will need metrics for biclusters evaluation. This is the subject of the next chapter.

4 Evaluation of Bicluster Results

The comparison of clustering solutions is well established in the clustering literature, which comprehends several studies on the analysis of the properties of similarity measures for comparing partitions. However, we cannot directly use those metrics to compare biclusters, since a bicluster comprehends a tuple of two sets (rows and columns) (HORTA; CAMPELLO, 2014). Besides that, in this work we have two additional restrictions:

Overlapping: as two biclusters may overlap, the metric must consider this scenario.

Quantity of biclusters: as we will verify in the experiments, the enumeration usually returns a quantity of biclusters that are far from the real quantity of biclusters. In this case, we need a metric that does not consider the quantity of biclusters, but evaluates how the biclusters include the proper elements (rows and columns). On the other hand, to evaluate the results from aggregation we need a metric that considers the quantity of biclusters, as we expect to achieve the right amount.

4.1 External metrics

Some metrics for evaluating a result consider only the data itself, without using external information. This is the characteristic of an internal metric. External metrics compare the results with a reference solution. In this work we will use only external metrics, except for the Gene Ontology Enrichment Analysis. For an extensive comparison of external metrics for biclustering solutions, the reader may refer to (HORTA; CAMPELLO, 2014).

4.1.1 Pairwise Precision, Recall and F-score

Precision, *Recall* and *F-score* are often used on information retrieval for measuring binary classification (SALTON, 1971; RIGSBERGEN, 1979). If we take pairs of elements, we can extend these metrics to evaluate clustering / biclustering solutions. For each pair of points that share at least one bicluster in the overlapping biclustering results, these measures try to estimate whether the prediction of this pair as being in the same bicluster was correct with respect to the underlying true categories in the dataset (BANERJEE *et al.*, 2005). It is important to highlight that these metrics do not consider the quantity of biclusters.

Lets define

$$pairs(\bar{B}) = \bigcup_{i=1}^{q} \{((g_1, s_1), (g_2, s_2)) \mid g_1, g_2 \in B_i^r; s_1, s_2 \in B_i^c; (g_1, s_1) \neq (g_2, s_2)\}, \tag{4.1}$$

as a function that returns a set with all pairs of elements of the biclusters of a solution $\bar{B} = \{B_i\}_{i=1}^{q}$, where q is the quantity of biclusters on the solution set.

Lets consider $\bar{B}$ as the proposed solution and $\bar{C}$ as the reference solution. We will define the metrics *Pairwise Precision* - or just *Precision*, *Pairwise Recall* - or just *Recall*, and *F-score* as follows:

$$PairwisePrecision(\bar{B}, \bar{C}) = \frac{|pairs(\bar{B}) \cap pairs(\bar{C})|}{|pairs(\bar{B})|}. \tag{4.2}$$

$$PairwiseRecall(\bar{B}, \bar{C}) = \frac{|pairs(\bar{B}) \cap pairs(\bar{C})|}{|pairs(\bar{C})|}. \tag{4.3}$$

$$F - score(precision, recall) = \frac{2 \times precision \times recall}{precision + recall}. \tag{4.4}$$

It is important to discuss the behavior of these metrics. We can interpret the *Pairwise Precision*, or just *Precision* for simplicity, as a percentage of true indications in a solution. For example, if 30% of the elements that are part of a bicluster in a solution are not part of any biclusters on the reference solution, the *Precision* will be impacted. An extreme case is if a solution returns only one bicluster with one element (pair object / attribute). If this object in fact belongs to a bicluster on the reference solution, then the *Precision* will be 1. This solution is very precise, as all elements that it said to be part of a bicluster in fact are.

We can interpret the *Pairwise Recall*, or just *Recall* for simplicity, as the percentage of elements that are truly part of a bicluster and the solution included in a bicluster. For example, if a solution includes in their biclusters only 70% of the elements that are part of a bicluster in the reference solution, then the *Recall* will be affected. An extreme case happens if a solution includes all the dataset into one bicluster. In this case the *Recall* is 1, the maximum value. This solution was able to include in a bicluster every element that should be part of a bicluster.

Precision is the fraction of retrieved pairs that are relevant; while *Recall* is the fraction of relevant pairs that are retrieved. The F-score is the harmonic mean of the *Precision* and the *Recall*. For more details about these metrics, the reader may refer to Menestrina *et al.* (2009).

4.1.2 Clustering error

Horta & Campello (2014) made an extensive analysis of several external metrics for biclustering evaluation. One of these metrics was Clustering Error (CE), that considers the quantity of biclusters in its evaluation.

Supposing that $\bar{B}$ represents the proposed solution; $|\bar{B}| = k$; and $\bar{C}$ represents the reference solution; $|\bar{C}| = q$. We will define the CE metric as follows:

$$CE(\bar{B}, \bar{C}) = \frac{d_{max}}{|U|}, \tag{4.5}$$

where $d_{max} = \sum_{i=1}^{min\{k,q\}} |B^r \times B^c \cap C^r \times C^c|$, where i represents a map between the biclusters of the proposed solution with the reference solution having maximum overlap; and $|U| = |\bigcup_{i=1}^{k} B_i^r \times B_i^c \cup \bigcup_{i=1}^{q} C_i^r \times C_i^c|$ is the number of elements in the area covered by biclusters of both the reference and the proposed solution. This metric severely penalizes a solution with more biclusters than the reference, thus not recommended for evaluating enumerative results.

4.1.3 Difference in Coverage

We propose the difference in coverage, that measures what the reference biclustering solution covers and the found biclustering solution does not cover, and vice versa. This metric gives an intuitive idea of how two solutions cover distinct areas of the dataset. Let $\cup_{\bar{B}} = \bigcup B_i^r \times B_i^c$ be the usual union set of a biclustering solution $\bar{B}$. Let $\bar{B}$ and $\bar{C}$ be the found and the reference biclustering solution, respectively. Then the difference in coverage is given by:

$$dif\_cov(\bar{B}, \bar{C}) = \frac{|\cup_{\bar{B}} - \cup_{\bar{C}}| + |\cup_{\bar{C}} - \cup_{\bar{B}}|}{m \times n}. \tag{4.6}$$

Figure 7 shows an example. Consider the red biclusters as the reference solution, and the black biclusters as the proposed solution. The difference in coverage is the gray area. We will use this measure to verify how different an aggregated solution is from the enumeration.

This metric is very similar to the pairwise definitions of Precision and Recall, but gives a more direct and intuitive idea of how the proposed solution differs from the original solution.

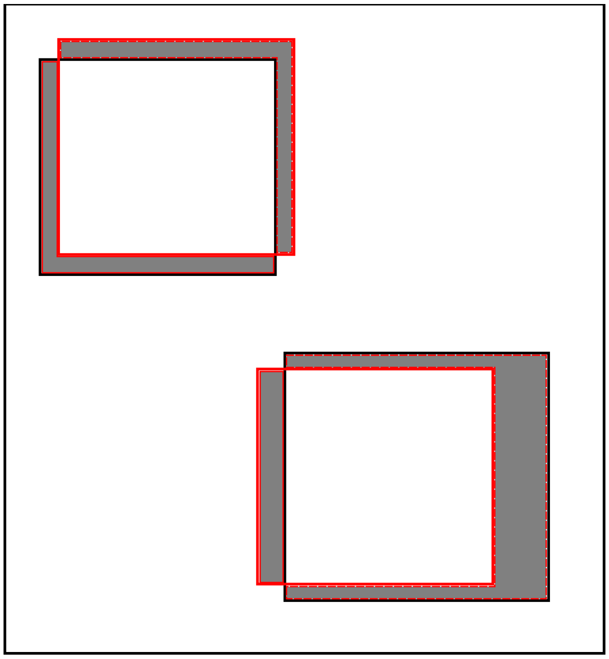


Figure 7 – Illustration of the difference in coverage.

4.2 Other forms of evaluation

4.2.1 Gene Ontology Enrichment Analysis

The Gene Ontology Project [1] (GO) is an initiative to develop a computational representation of the knowledge of how genes encode biological functions at the molecular, cellular and tissue system levels. Groups around the world collaborate developing gene function annotations and keeping it up to date on the gene ontology website.

One of the main uses of GO is to perform enrichment analysis on gene sets. For example, given a set of genes that are up-regulated under certain conditions, an enrichment analysis will find which GO terms are over-represented (or under-represented) using annotations for that gene set[2].

It is important to notice that this is not an external metric, or a metric at all. It will not compare two bicluster solutions, but just verify how relevant is a set of genes when compared to the annotations in Gene Ontology Project datasets. This method is commonly used to analyze results from biclustering techniques on microarray gene expression datasets, because it may indicate the relevance of the gene sets. Just to cite some: (HARTIGAN, 1972; CHENG; CHURCH, 2000; LAZZERONI; OWEN, 2000; JIONG *et al.*, 2003; ZHAO;

[1] http://geneontology.org

[2] http://geneontology.org/page/about Acessed on 2014, November, 26

ZAKI, 2005; HANCZAR; NADIF, 2011b). Another drawback is that this analysis discard the information of the attributes, using just the set of genes in its calculation.

We will use the Gene Ontolgoy Enrichment Analysis (GOEA) to verify the relevance of the results in a gene expression dataset. Table 2 shows an example of a result from GOEA. In this table, the GO term column represents an ontology term that the set of genes is related to in the annotations. The column p-val "is the probability or chance of seeing at least x number of genes out of the total n genes in the list annotated to a particular GO term, given the proportion of genes in the whole genome that are annotated to that GO Term. That is, the GO terms shared by the genes in the user's list are compared to the background distribution of annotation. The closer the p-value is to zero, the more significant the particular GO term associated with the group of genes is (i.e. the less likely the observed annotation of the particular GO term to a group of genes occurs by chance)"[3]. The column counts shows how many times the gene set was related to that specific annotation versus how many times the gene set was related to other annotations. The column definition is a brief description of the GO term.

We have interest in the p-val column, that is somehow an indication of the biological relevance of the gene set to the related GO Term, and may indicate the importance of the bicluster. If this value is less than 0.05, the bicluster may be considered enriched and we have a good indication that a further analysis of the gene set should be conducted.

[3] http://geneontology.org/page/go-enrichment-analysis

Table 2 – Enrichment analysis of the first bicluster from the aggregation by overlapping with rate of 70%.

GO Term	p-val	counts	definition
GO:0044464	0.00000000	39 / 774	Any constituent part of a cell, the basic structural and functional unit of all organisms. [GOC:jl]...
GO:0044444	0.00000011	19 / 608	Any constituent part of the cytoplasm, all of the contents of a cell excluding the plasma membrane and nucleus, but including other subcellular structures. [GOC:jl]...
GO:0044424	0.00000350	19 / 578	Any constituent part of the living contents of a cell; the matter contained within (but not including) the plasma membrane, usually taken to exclude large vacuoles and masses of secretory or ingested material. In eukaryotes it includes the nucleus and cytoplasm. [GOC:jl]...
GO:0098593	0.00010607	16 / 492	A cup shaped specialization of the cytoskeleton that forms a thin layer located just below the apical mass of mature mucin secretory granules in the cytoplasm of goblet cells of the intestinal epithelium. It consists of an orderly network of intermediate filaments and microtubules. Microtubules are arranged vertically, like barrel staves, along the inner aspect of the theta. Intermediate filaments form two networks: an inner, basketlike network and an outer series of circumferential bundles resembling the hoops of a barrel. [PMID:6541604]...

Part II

Proposals of this dissertation

5 Proposals for aggregation

In this chapter, we will introduce our proposals for aggregating fragmented biclusters. We already briefly introduced the clustering with single linkage method in section 2.2, and the problem of bicluster aggregation in section 3.2. These concepts will be used now.

5.1 Aggregation with single linkage

Our proposal receives as input a bicluster solution $\bar{S}$, from enumeration or from a result presenting high overlapping among its components. With this solution, we transform each bicluster into a binary vector representation as follows: Given the dimensions of the dataset $\mathbf{A} \in \mathbb{R}^{n \times m}$, each bicluster will be a binary vector $\mathbf{x}$ of length $n + m$. For a bicluster B transformed into the binary vector $\mathbf{x}$, the first n positions represent the rows of the dataset $\mathbf{A}$ and their values are given by the function $\mathbf{1}_R \to \{0, 1\}$ defined as:

$$\mathbf{1}_R(i) := \begin{cases} 1 \; if \; i \in B^r, \\ 0 \; if \; i \notin B^r, \end{cases} \tag{5.1}$$

where i is the index of the vector $\mathbf{x}$. In other words, if the bicluster contains the ith row, $\mathbf{x}_i = 1$, otherwise, $\mathbf{x}_i = 0$. The last m positions represent the columns of the dataset $\mathbf{A}$ and their values are given by the function $\mathbf{1}_C \to \{0, 1\}$ defined as:

$$\mathbf{1}_C(i) := \begin{cases} 1 \; if \; i \in B^c, \\ 0 \; if \; i \notin B^c. \end{cases} \tag{5.2}$$

In other words, if the bicluster contains the ith column, $\mathbf{x}_{n+i} = 1$, otherwise, $\mathbf{x}_{n+i} = 0$.

After this transformation, we use the Hamming distance, defined on the Eq. 5.3, to apply the single linkage clustering on the existing biclusters.

$$dist(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n+m} |x_i - y_i|. \tag{5.3}$$

Notice that the Hamming distance on this transformation will just count how many rows and columns are different on the two bicluster. In this case, a non-maximal bicluster may be distant from the bicluster that covers its maximal area, thus impacting the quality of the results of this method of aggregation. So, it is important to ensure the maximality of the biclusters in the solution that will be aggregated by this method.

We still have the task of cutting the dendrogram. This is a well studied task in the clustering community and some guides are given in (SOKAL; ROHLF, 1962; RAND, 1971; LANGFELDER *et al.*, 2008). In this work we will choose the cut visually. Basically, when the height of a junction is more pronounced than the heights of the junctions below it, we consider it as a good cut, since the distance of the clusters being joined is higher.

After choosing a cut on the dendrogram, we aggregate all biclusters that belong to a junction using the function *aggreg*, defined as:

$$aggreg(B, C) = (B^r \cup C^r, B^c \cup C^c), \tag{5.4}$$

that is simply the union of rows / columns of the biclusters. Note that the *aggreg* function is associative, as demonstrated below:

$$\begin{aligned} agreg(B, agreg(C, D)) = & \quad agreg(B, (C^r \cup D^r, C^c \cup D^c)) \\ = & \quad (B^r \cup (C^r \cup D^r), B^c \cup (C^c \cup D^c)) \\ = & \quad ((B^r \cup C^r) \cup D^r, (B^c \cup C^c) \cup D^c) \\ \equiv & \quad agreg(agreg(B, C), D) \end{aligned}$$

Moreover, we want to highlight that the direct union of rows / columns may include elements that shouldn't be part of a bicluster. In Section 5.3 we will present a way to remove rows / columns that are interpreted as outliers.

5.2 Aggregation by overlapping

Considering that the biclusters we want to aggregate have high overlapping, it is natural to aggregate x biclusters with an overlapping rate above a defined threshold. This proposal is based on the aggregation by pairs: while having two biclusters with an overlapping rate higher than a pre-determined threshold th, we remove them from the set of biclusters, and include the result of the function *aggreg*, defined on Eq. 5.4, taking these two biclusters as the arguments.

Figure 8 shows the aggregation of two biclusters. If the gray area is higher than the threshold, we aggregate the two biclusters.

Let B, C, D, and E be biclusters. Note that $ov(D, E) \geq ov(B, E)$ and $ov(D, E) \geq ov(C, E)$ for $D = aggreg(B, C)$. So, for all biclusters E where $ov(B, E) \geq th$ or $ov(C, E) \geq th$, $ov(D, E) \geq th$. For this reason, the order of the aggregation does not interfere on the final result. It is also important to note that the new bicluster D can have $ov(D, E) \geq th$, for some bicluster E where $ov(B, E) < th$ and $ov(C, E) < th$.

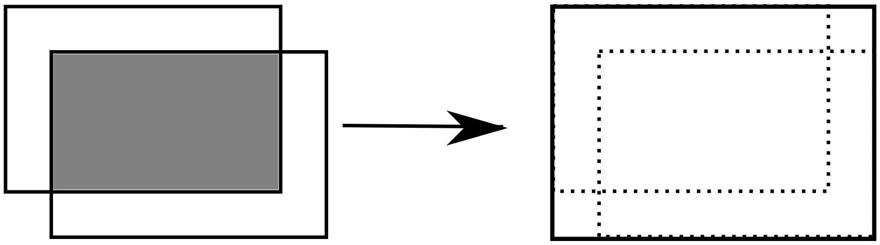


Figure 8 – Example of aggregation of two biclusters.

Again, as the *aggreg* function unites the rows (columns), we need an additional step for outlier removal.

5.3 Outlier removal

As explained in Sections 5.1 and 5.2, and as we will see on the experiments in Chapter 6, in our proposals of aggregation it will be necessary a step of outlier removal. Ideally, this step is going to remove elements that should not be part of a bicluster, and do nothing when these elements do not exist.

Let $B = (B^r, B^c)$ be an aggregated bicluster, with $|B^r| = o, |B^c| = p$. We define a participation matrix $\mathbf{P} \in \mathbb{R}^{o \times p}$, where each element p_{ij} indicates the quantity of biclusters in which this element takes part on B. For example, if an element is part of 15 biclusters that compose B, then its value on the $\mathbf{P}$ matrix will be 15.

So, we will explain the process of outlier removal with the help of Figure 9. The first step is to compute the matrix $\mathbf{P}$ for the aggregated bicluster. Then we have two steps of outlier removal: one for the objects, and other for the attributes.

To remove possible outlier objects, we take the mean and the standard deviation of all columns on the bicluster of matrix $\mathbf{P}$. The left side of Figure 9 illustrates this step. After that, we check the values of each element of the columns. If the value is less than the mean minus one standard deviation, then we check this element as a potential outlier. In the right side of Figure 9, we can see that the entire first row was checked as potential outlier because $1 < 7.75 - 4$. If we mark the entire row as a potential outlier, it is removed from the bicluster. In our example, that is the case.

We execute the same process for the columns, calculating the mean, standard deviation and checking for potential outliers on the rows. We remove the column if it is entirely marked as a potential outlier.

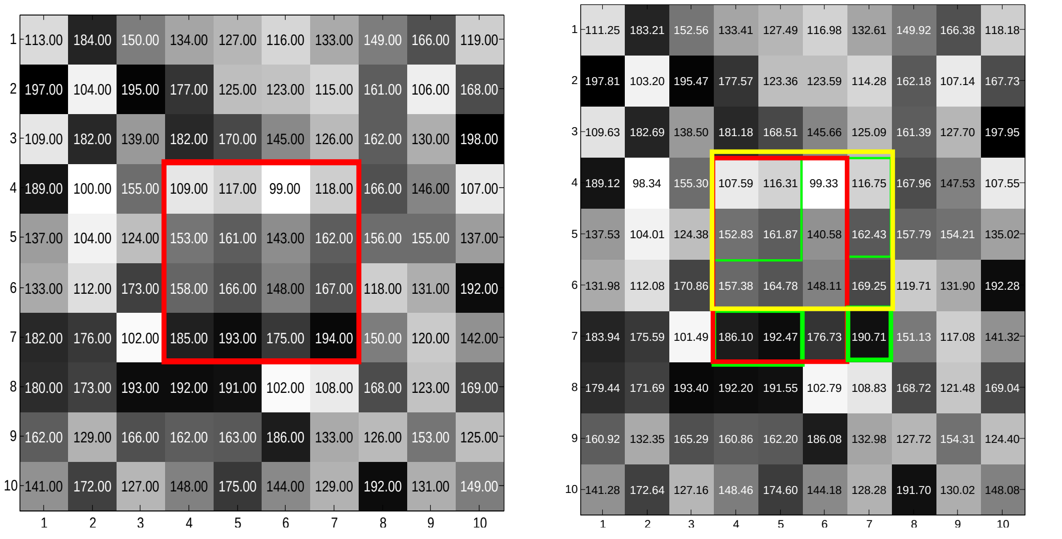
	1	1	1	1		1	1	1	1
	10	10	9	9		0	0	0	0
	10	10	9	9	→	0	0	0	0
	10	10	9	9		0	0	0	0
Mean	7.75	7.75	7	7					
Std	4.5	4.5	4	4					

Figure 9 – Example of outlier removal.

The next section shows a toy example of the entire process: aggregation with single linkage and outlier removal.

5.4 Practical Example

We will now supply an example of the process of aggregation. Figure 10a shows a toy dataset with a bicluster of coherent values. After adding a Gaussian noise ($\mu = 0, \sigma = 1$), we can see in Figure 10b that now we have three biclusters, one depicted in yellow (B), one depicted in green (C) and one depicted in red (D). The noise fragmented the first bicluster into those three.



	1	2	3	4	5	6	7	8	9	10
1	113.00	184.00	150.00	134.00	127.00	116.00	133.00	149.00	166.00	119.00
2	197.00	104.00	195.00	177.00	125.00	123.00	115.00	161.00	106.00	168.00
3	109.00	182.00	139.00	182.00	170.00	145.00	126.00	162.00	130.00	198.00
4	189.00	100.00	155.00	109.00	117.00	99.00	118.00	166.00	146.00	107.00
5	137.00	104.00	124.00	153.00	161.00	143.00	162.00	156.00	155.00	137.00
6	133.00	112.00	173.00	158.00	166.00	148.00	167.00	118.00	131.00	192.00
7	182.00	176.00	102.00	185.00	193.00	175.00	194.00	150.00	120.00	142.00
8	180.00	173.00	193.00	192.00	191.00	102.00	108.00	168.00	123.00	169.00
9	162.00	129.00	166.00	162.00	163.00	186.00	133.00	126.00	153.00	125.00
10	141.00	172.00	127.00	148.00	175.00	144.00	129.00	192.00	131.00	149.00

(a) *Example of a bicluster.*

	1	2	3	4	5	6	7	8	9	10
1	111.25	183.21	152.56	133.41	127.49	116.98	132.61	149.92	166.38	118.18
2	197.81	103.20	195.47	177.57	123.36	123.59	114.28	162.18	107.14	167.73
3	109.63	182.69	138.50	181.18	168.51	145.66	125.09	161.39	127.70	197.95
4	189.12	98.34	155.30	107.59	116.31	99.33	116.75	167.96	147.53	107.55
5	137.53	104.01	124.38	152.83	161.87	140.58	162.43	157.79	154.21	135.02
6	131.98	112.08	170.86	157.38	164.78	148.11	169.25	119.71	131.90	192.28
7	183.94	175.59	101.49	186.10	192.47	176.73	190.71	151.13	117.08	141.32
8	179.44	171.69	193.40	192.20	191.55	102.79	108.83	168.72	121.48	169.04
9	160.92	132.35	165.29	160.86	162.20	186.08	132.98	127.72	154.31	124.40
10	141.28	172.64	127.16	148.46	174.60	144.18	128.28	191.70	130.02	148.08

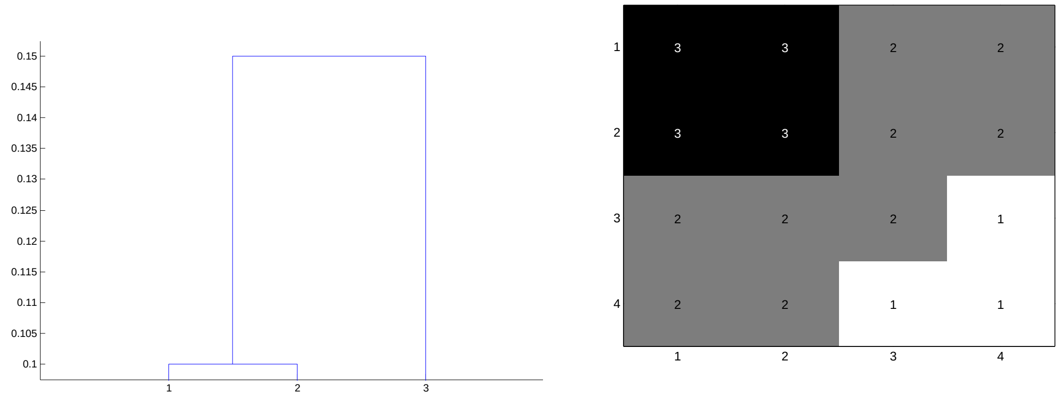
(b) *Example of the fragmentation of the bicluster.*

Figure 10 – Example of the impact of noise fragmenting an original single bicluster.

Let us transform those three biclusters in three binary vectors, using the process described in Section 5.1.

$$\begin{array}{ll}
B =[0,0,0,1,1,1,0,0,0,0, & 0,0,0,1,1,1,1,0,0,0] \\
C =[0,0,0,1,1,1,1,0,0,0, & 0,0,0,1,1,1,0,0,0,0] \\
D =[0,0,0,1,1,0,1,0,0,0, & 0,0,0,1,1,0,1,0,0,0] \\
\quad\quad\quad \mathit{rows} & \quad\quad\quad\quad\quad\quad \mathit{columns}
\end{array}$$

With the proper representation of the biclusters, we can run the hierarchical clustering with single linkage using these vectors as objects.



	1	2	3	4
1	3	3	2	2
2	3	3	2	2
3	2	2	2	1
4	2	2	1	1

(a) *Example of the dendrogram of an aggregation with single linkage.*

(b) *Example of a* ***P*** *matrix.*

Figure 11a shows the dendrogram of this process. As this example is too small, the height of the whole dendrogram represents a very short distance between the groups, and we can cut it on the top, having just one cluster. The Figure 11b shows the **P** matrix for outlier removal, that in this example will not filter any element. Notice that with one cluster, we have recovered the original bicluster.

The next part of this dissertation will explain the experiments that we run to test our proposals of bicluster aggregation.

Part III

Results and Discussion

6 Results and Discussion

This chapter describes the methods and implementations used in the experiments performed along the development of the research. Except when explicitly denoted, the author implemented all the programs and scripts mentioned in this text. We will also present the results and discussion.

6.1 Datasets

In our experiments we employed three artificial datasets: *art1*, *art2*, and *art3*; and two real datasets: GDS2587 and *Food*. We designed the artificial datasets to present different scenarios with increasing difficulty. We will use them to verify the impacts of noise, and to compare the performance of several aggregation methods and outlier removal of elements in the aggregation. The two real datasets are from different backgrounds. The first real dataset is from microarray gene expression data, as it is a well known application of biclustering methods. The second real dataset is about nutritional information, and it was used to evaluate a well-known biclustering algorithm called *Plaid Models* (LAZZERONI; OWEN, 2000). We will describe the details of each dataset in what follows.

6.1.1 *Art1*

This dataset has 1000 objects and 15 attributes. Each entry is a random integer, drawn from a discrete uniform distribution on the set {1, 2, ..., 100}. Then we inserted 5 bicluster of coherent values, arbitrarily positioned and without overlapping. For each bicluster, the quantity of objects was randomly drawn from the set $\{50, \ldots, 60\}$, and the quantity of attributes was randomly drawn from the set $\{4, 5, 6, 7\}$. To insert a bicluster, we fixed the value of the first attribute and obtained the values of the other attributes by adding a constant value to the first column. This constant value was randomly drawn from the set $\{-10, -9, \ldots, -1, 1, \ldots, 9, 10\}$.

6.1.2 *Art2*

We generated this dataset by the same process of *art1*. The only difference is that 4 of 5 biclusters have some overlapping. Two biclusters have approximately 36% of overlapping, other two biclusters have approximately 11% and the last bicluster has no overlap with others. These percentages of overlapping were decided to have a controlled difficulty of the task.

6.1.3 *Art3*

We generated this dataset by the same process of *art1*. The difference now is that it has 15 biclusters, with different levels of overlapping and some biclusters overlap with more than one peer. The overlapping setup in this dataset is: 3 pairs of biclusters with approximately 15% of overlapping, a pair with 30%, a pair with 34%, a pair with 39%, a pair with 48% and a pair with 60%. Again, these percentages of overlapping were decided to have a controlled difficulty of the task.

6.1.4 Microarray Gene Expression *GDS2587*

GDS2587[1] is a microarray gene expression dataset. Each entry in the matrix is the $\log_2$ ratio of the expression. The $\log_2$ ratio is defined as $\log_2(T/R)$, where T is the gene expression level in the testing sample and R is the gene expression level in the reference sample. The data was collected from the organism *E. coli*. We removed every gene with missing data in any sample, and the data was normalized by mean centralization, as common in gene expression data analysis (PRELIć *et al.*, 2006). After this pre-processing step, the dataset contains 2792 genes and 7 samples. In this dataset we aim to validate our contribution when devoted to the analysis of microarray gene expression data, as it is considered a relevant application of biclustering methods.

6.1.5 *Food*

Food[2] is a dataset with 961 objects, which represent different foods, and 7 attributes, which represent nutritional information (grams of fat, calories of food energy, grams of carbohydrate, grams of protein, milligrams of cholesterol, grams of saturated fat, and the weight in gram of the food). As the values of each attribute are in different ranges, we used the same pre-processing as (VERONEZE *et al.*, 2014), rescaling the attributes to the range [0, 1000]. In this dataset our goal is to illustrate the usefulness of bicluster aggregation in a different scenario and to verify if the aggregation leaves uncovered areas that the enumeration has covered at first.

6.2 Experiment 1: The impact of noise

In this experiment we will only use the artificial datasets. Our goal is to verify the impact of noise in the enumeration of biclusters. To this end, we will add a Gaussian noise

[1] http://www.ncbi.nlm.nih.gov/sites/GDSbrowser?acc=GDS2587
[2] http://www.ntwrks.com/chart1a.htm

with $\mu = 0$ and $\sigma \in \{0, 0.01, \dots, 1\}$, to each dataset, and run the RIn-Close algorithm. This procedure will be repeated for 30 times and all reported values will be the average of these executions. We set the RIn-Close parameters as follows:

Type of bicluster: as we know the type of the biclusters on the datasets, we set RIn-Close to mine coherent values biclusters.

minRow: as we know the minimum possible size of the biclusters, we set minRow to 4.

minCol: as we know the minimum possible size of the biclusters, we set minCol to 50.

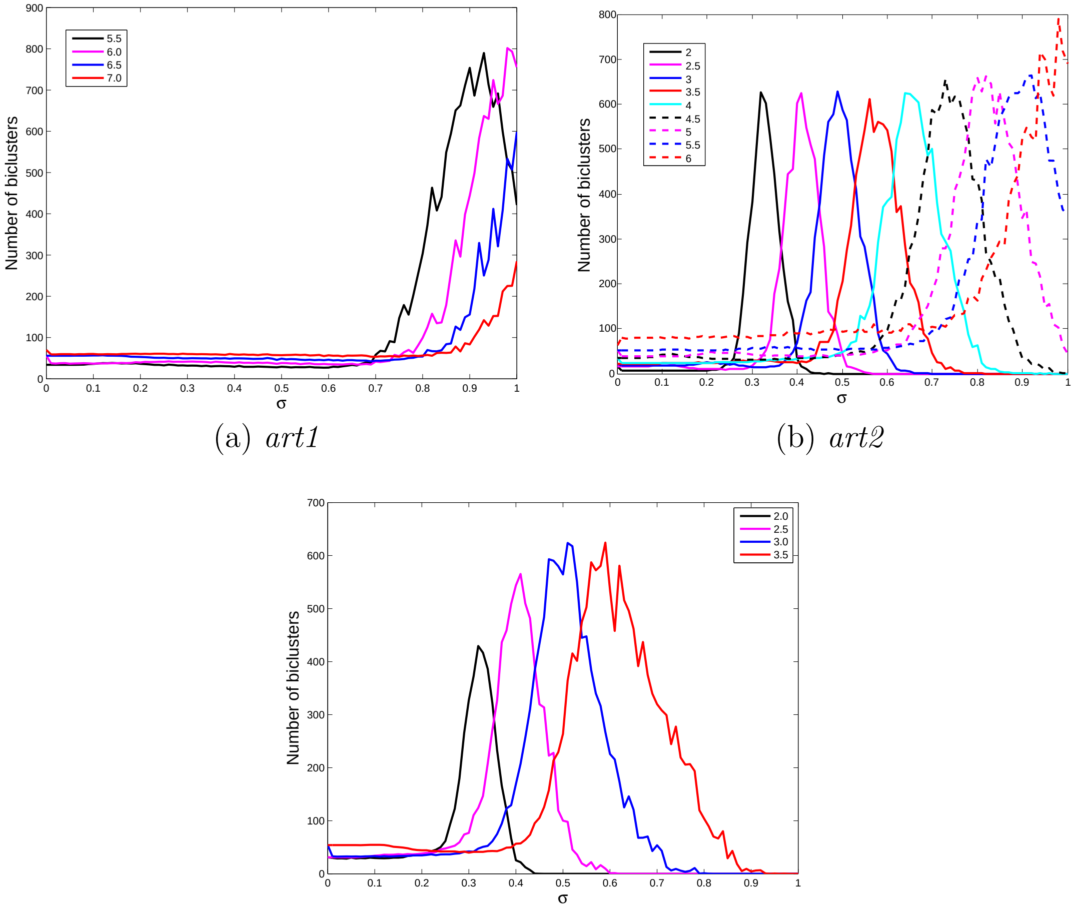
ϵ: We will test a sequence of crescent values for ϵ due to its importance for a good result. If ϵ is too small, we may miss important biclusters expressing more internal variance. If ϵ is too high, the biclusters may include unexpected objects and attributes.

Figure 11 shows the evolution of the quantity of enumerated biclusters, by the variance of noise, for each value of ϵ, for all artificial datasets.

In all datasets, for each value of ϵ, the behavior is the same: as the noise increases the quantity of enumerated biclusters eventually starts to increase. In Figures 11a and 11b, we know that the real quantity of biclusters is 5, but when the noise increases, the enumerated quantity reaches approximately 800 biclusters, depending on the value of ϵ. Notice that the biclusters are distinct and no bicluster is contained in another one. In Figure 11c, we can see that the quantity of biclusters reaches high values too. At some level of noise, the number of biclusters starts to decrease to a point that the algorithm is not able to find any bicluster, as there is no way to satisfy the coherence threshold. But, as we can see in Figure 11a, the added noise was not enough to reach this point for dataset *art1*, which is the easiest one.

In Figure 12, we can see the quality of the enumeration without considering the quantity of biclusters. Due to the effect of noise on the enumeration, if we use a metric that considers the quantity of biclusters, we will not be able to verify the quality of the solution. In this case, for this experiment we will report only the metrics *Precision*, *Recall* and *F-score*.

As we can see in Figure 12b, the noise has almost no interference in the *Recall*. It means that this dataset has biclusters very well defined. Even with high degrees of noise they are not missed. On the other hand, when the variance of the noise is too low, Figure 12a shows that the enumerated biclusters contains more elements than expected. It is happening because the parameter ϵ is high, allowing some elements to be part of the biclusters, even without being part of the original solution. As the noise increases, less of these initial elements are going to satisfy the ϵ restriction to be included in some bicluster. In this dataset, the



(a) *art1*

(b) *art2*

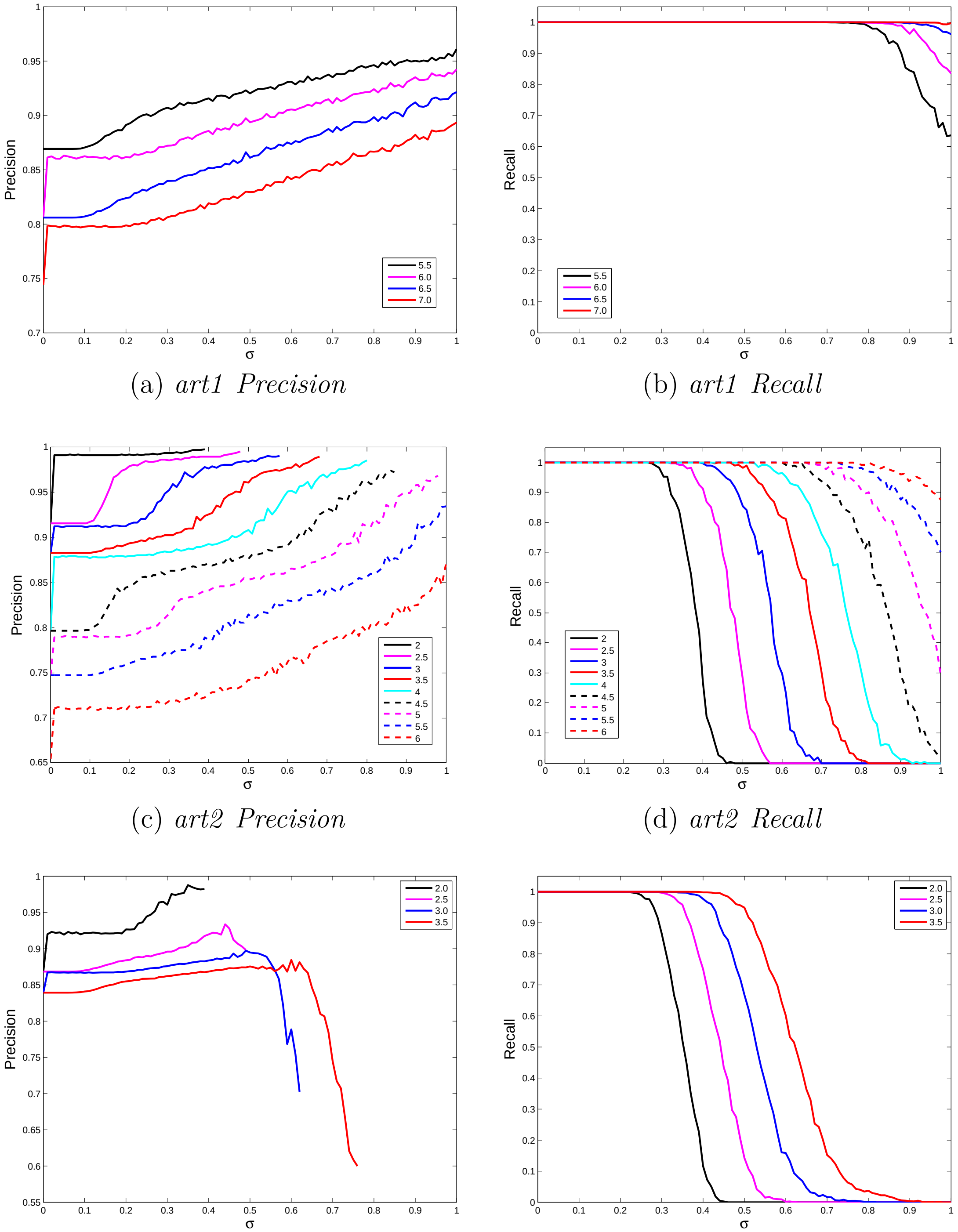
(c) *art3*

Figure 11 – Quantity of biclusters by the variance of the Gaussian noise in the artificial datasets. Each curve is parameterized by ϵ.

effect of the noise was not so severe on the *Recall*, given that it only started to decrease when the variance of the noise was close to 1.

In dataset *art2*, the effects of the noise can be better observed. Figure 12d shows that the noise starts to affect the solutions very early. When $\epsilon = 2$, the *Recall* starts to decrease very soon, when $\sigma \approx 0.3$. However, for more relaxed values of ϵ, we can still see the decrease on the *Recall*. Being the most difficult, dataset *art3* is the most affected by noise. Independently of the value of ϵ, the RIn-Close can not find any bicluster after some levels of variance in the noise. For example, when $\epsilon = 2$, after $\sigma \approx 0.4$ the *Precision* gets undefined. It happens because the denominator of Eq. 4.2 is not defined when the quantity of biclusters is zero. In Figure 12f, we can see that the decline of the *Recall* starts when $\sigma \approx 0.3$ for $\epsilon = 2$.

In this experiment we may conclude that the noise fragments the true biclusters into many with high overlapping. This was observed in Liu *et al.* (2004), in Zhao & Zaki (2005), and in Gao & Akoglu (2014). Intuitively, it seems to be advantageous to explore this high overlapping aiming at aggregating the enumerated biclusters, getting a result closer to the ground truth. Now we will verify the effects of the aggregation on the artificial datasets.



(a) *art1 Precision*

(b) *art1 Recall*

(c) *art2 Precision*

(d) *art2 Recall*

(e) *art3 Precision*

(f) *art3 Recall*

Figure 12 – *Precision* and *Recall* for the solutions of RIn-Close, by the variance of the Gaussian noise in the artificial datasets. Each curve is parameterized by ϵ.

6.3 Experiment 2: The impact of aggregation

As we could see on the previous experiment, the enumeration is affected by the noise. Not only the quantity of biclusters increases significantly, but also the quality of the solution decreases.

In this experiment, our goal is to verify the impact of aggregation on the previous results of RIn-Close. We will choose the results with the ϵ that led to a initial *Precision* closest to 0.85. That gives us: *art1* with $\epsilon = 6$, *art2* with $\epsilon = 4$ and *art3* with $\epsilon = 3$. This value was chosen because if the Precision is too low, it means that the ϵ value is allowing too many undesired objects or attributes in the enumerated biclusters. In this case, the aggregation may not improve the quality of the final results because their input is not of good quality. If the Precision is too high, we will only be able to see improvements in the reduced quantity of biclusters, but the aggregation may increase the Precision too.

This value was chosen because if the *Precision* is too low, the input of the aggregation may be of poor quality. If the *Precision* is too high, we may not be able to see the improvements on the quality on the final aggregated results.

We will consider the following algorithms (explained in Sections 3.1, 3.2.1, 5.1 and 5.2) here:

Triclustering We will set k to the true number of biclusters. The authors supplied the code for this algorithm.

MicroCluster To parameterize this algorithm, we will run a grid search with the values in the set $0.15, 0.1, 0.05$ for each of the two parameters, getting 9 results for each run. Also, as the aggregation step of the algorithm is composed of two steps, merging and deleting, we will run each experiment twice: with the merging step first (MD) and with the deleting step first (DM). Unless we want to draw attention to some particular fact, we will report only the best result. The authors supplied the code for this algorithm.

Single linkage We will cut the dendrogram with the proper quantity of biclusters: for *art1* and *art2*, 5 biclusters; for *art3*, 15 biclusters. Again, there are several ways to choose the cut, please see Section 5.1.

By overlapping We will test several values for the rate of overlapping.

After getting the results for all executions of the listed algorithms, we will choose the best result from each one and compare them using the *CE* metric. We will also run a two-sided

Wilcoxon Rank Sum Test at 5% significance level on the mean with the *CE* metric, to verify if the results have significant difference between each other.

As we have a large quantity of results for this experiment, we will organize them by dataset.

6.3.1 Results on *art1*

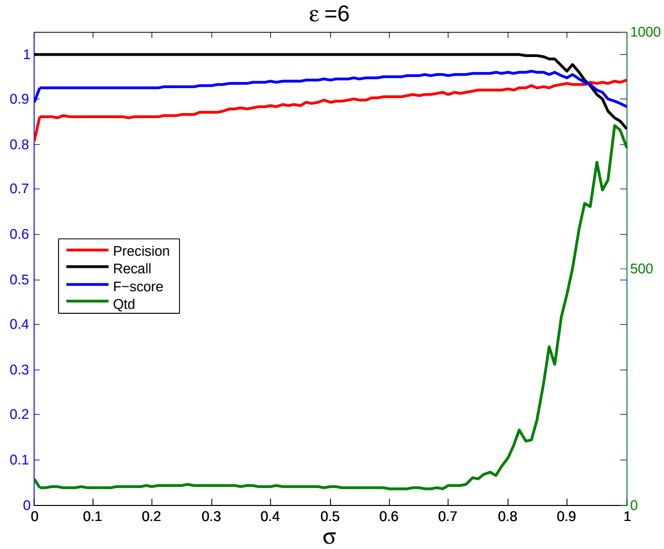


Figure 13 – Results produced by RIn-Close on art1. The scale on the right refers to the quantity of biclusters on the solution.

Figure 13 shows the results of RIn-Close when $\epsilon = 6.0$. This is the result that we will try to improve. We expect that the aggregation increases the *Precision* without decreasing the *Recall*, while reaching the true number of biclusters.

Figure 14 shows the quality of the results of the aggregation with single linkage. Despite the decrease in *Precision*, this is a good result because now we have only 5 biclusters. So, the aggregation with single linkage was able to reduce the quantity of biclusters to the right number, at the cost of the *Precision*. We may also pay attention to the fact that the *Recall* remained in the maximum value until $\sigma \approx 0.9$. It means that the biclusters of the aggregated solution are including more objects (attributes) than expected, which indicates decrease in *Precision*; but are not missing any elements, thus keeping the *Recall* at high values. In other words, every element that should be in a bicluster, in fact are, but elements that should not be included, are also part of the biclusters. So, this method of aggregation was not able to reach all of our goals, as the *Precision* is decreasing. However, the results suggested that we can add a step of outlier removal to try to improve the *Precision* of the final solution.

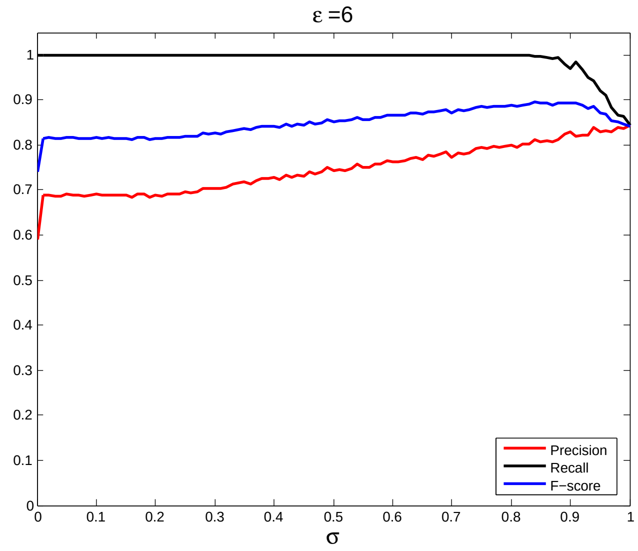
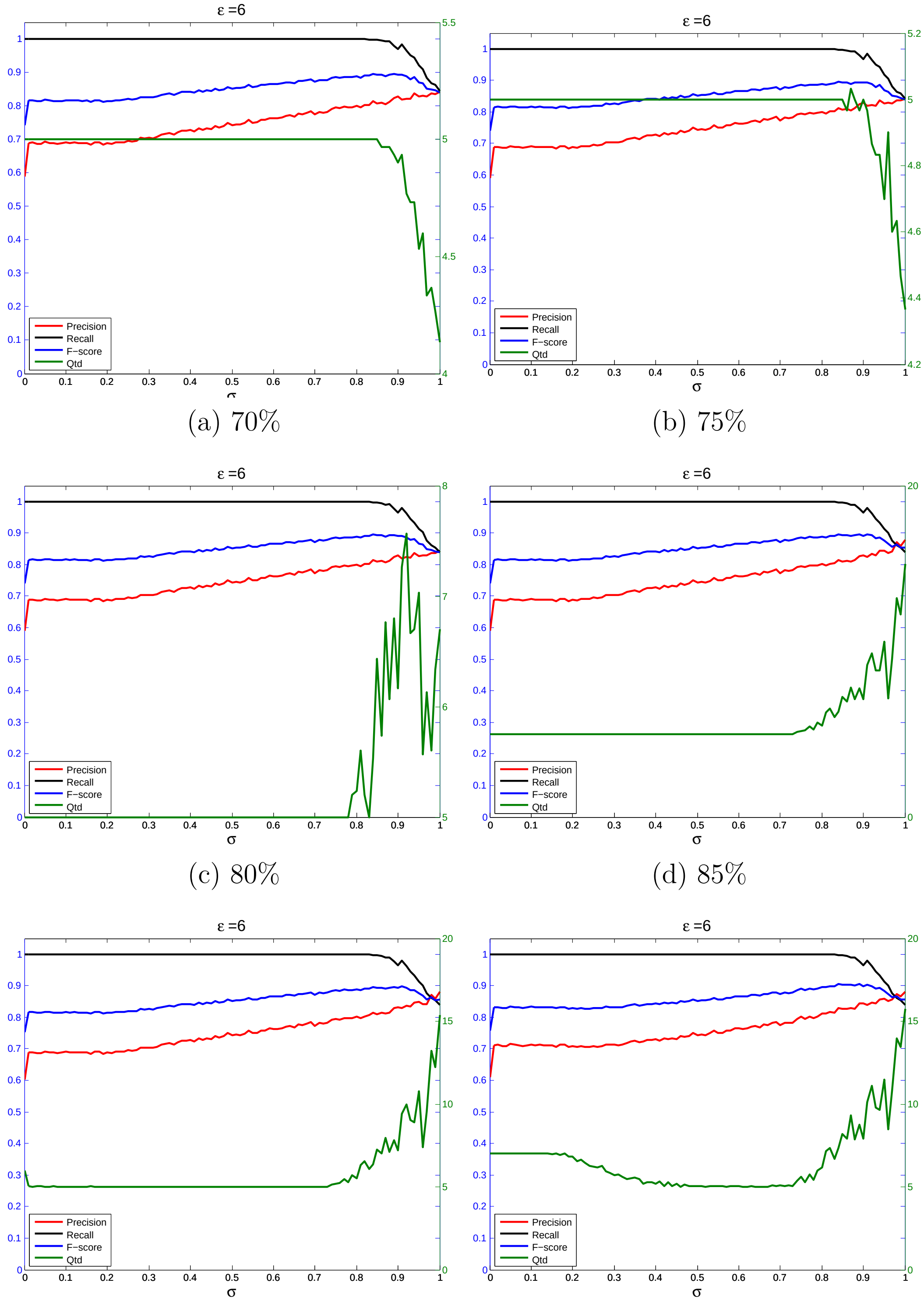


Figure 14 – Aggregation with single linkage using Hamming distance on *art1*.

In Figure 15 we can see the results from the aggregation by overlapping with several rates. The rates 60%, 65%, and 70% led to the same results and we will report just the latter result. We can see that even when the rate of overlapping was 95%, the aggregation reached the correct quantity of biclusters. But again the *Precision* decreased a little and the *Recall* remained high. This is the same scenario of the aggregation with single linkage and we can draw the same conclusions.

Figure 16 shows the results from the aggregation of MicroCluster algorithm when $\eta = \gamma$. The other results were omitted because they were not so different from the ones we are showing. We start by focusing on the stability of the results. For example, comparing Figures 16a and 16b, we can see that the latter showed more stability on the quantity of biclusters, on the *Precision* and on the *Recall*. The difference on the *Precision* and *Recall* metrics were not significant when changing the order of execution. But, as the quantity of biclusters when the deleting step was executed first is closest to the real one, from now on, for this dataset we will only consider the result when $\eta = 0.15$, $\gamma = 0.15$ in that order of operation. This aggregation decreased the *Recall* and increased the *Precision*, when compared to the results of the enumeration. Different from the results from the aggregation with single linkage or by overlapping, the results of MicroCluster are not including all objects (attributes) that should be part of a bicluster.

In Figure 17 we see the *Precision* and *Recall* of the solution obtained by the triclustering algorithm. We must remember that the parameter k is set to 5, which is the correct number of biclusters in the dataset. We can see that the *Precision* is very high, independently of the variance of noise, but the *Recall* decreased a lot. This fact indicates that the



(a) 70%

(b) 75%

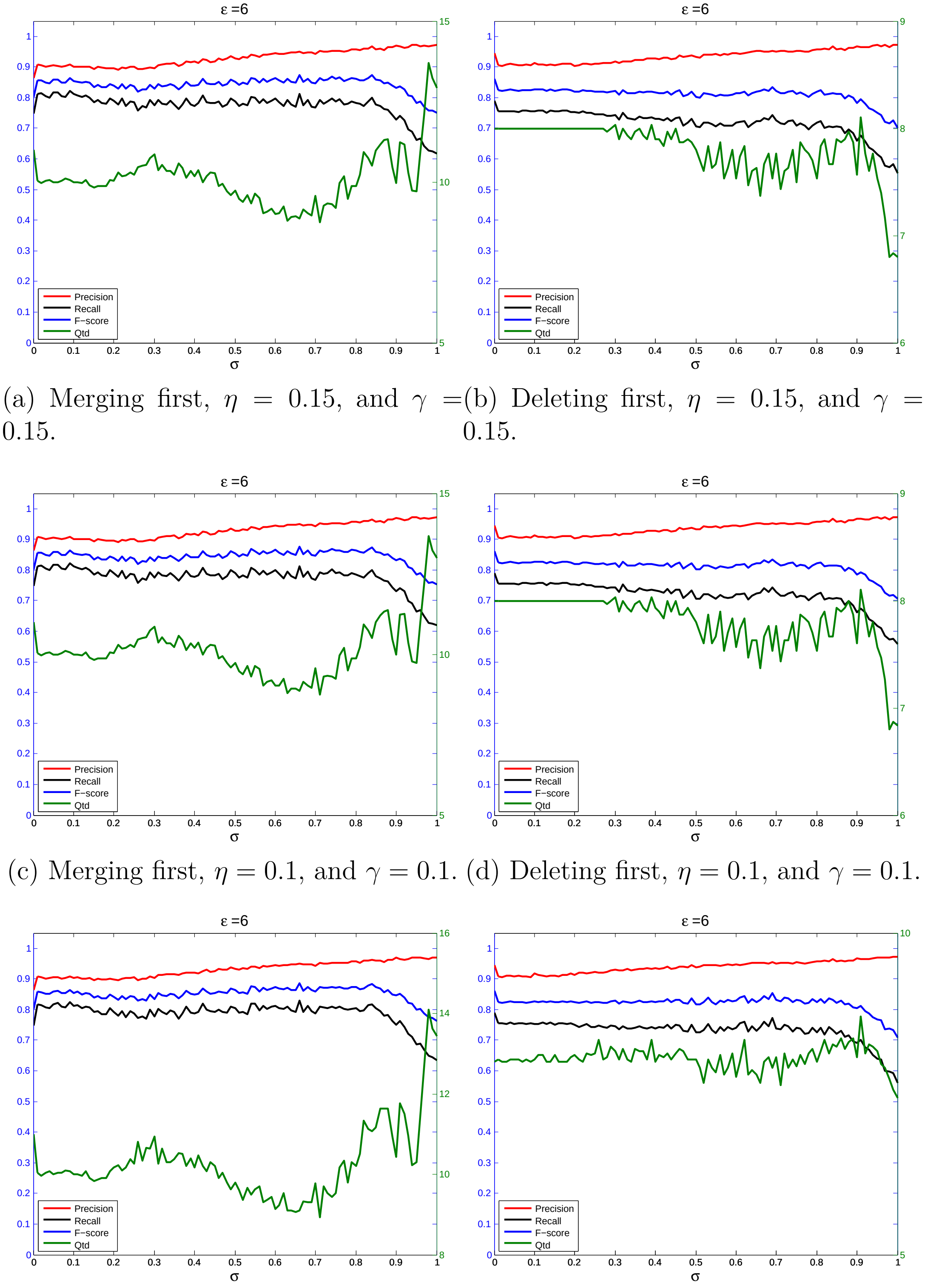
(c) 80%

(d) 85%

(e) 90%

(f) 95%

Figure 15 – Results from the aggregation by overlapping on *art1*. The scale on the right refers to the quantity of biclusters on the solution.



(a) Merging first, $\eta = 0.15$, and $\gamma = 0.15$.

(b) Deleting first, $\eta = 0.15$, and $\gamma = 0.15$.

(c) Merging first, $\eta = 0.1$, and $\gamma = 0.1$.

(d) Deleting first, $\eta = 0.1$, and $\gamma = 0.1$.

(e) Merging first, $\eta = 0.05$, and $\gamma = 0.05$.

(f) Deleting first, $\eta = 0.05$, and $\gamma = 0.05$.

Figure 16 – Results from the aggregation using MicroCluster on *art1*. The scale on the right refers to the quantity of biclusters on the solution.

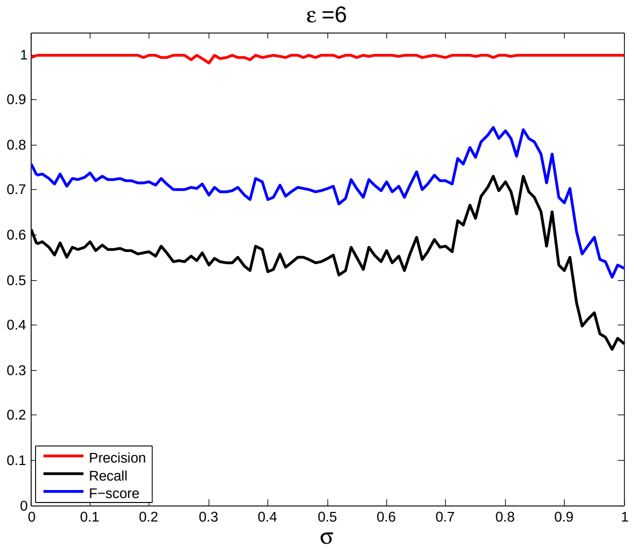


Figure 17 – Aggregation with the triclustering algorithm on *art1*.

biclusters of this solution only contain elements that should be in a bicluster. However, it is not including all objects (attributes) that it should, thus decreasing the *Recall*.

We will then compare the best results from each agglomeration, using the *CE* metric, and verify if the results have significant difference between each other. They are: *a*) single linkage; *b*) aggregation by overlapping with 70%; *c*) MicroCluster with $\eta = \gamma = 0.15$, deleting operation first; and *d*) triclustering.

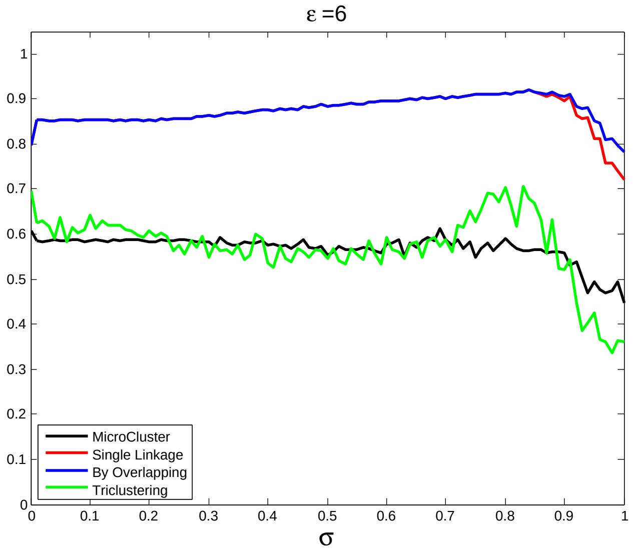


Figure 18 – *CE* of the best results of aggregation on *art1*.

Figure 18 shows the *CE* metric for the best solutions of each algorithm. We can see that the aggregation with single linkage and by overlapping had the best results on this metric, and they exhibit a similar pattern.

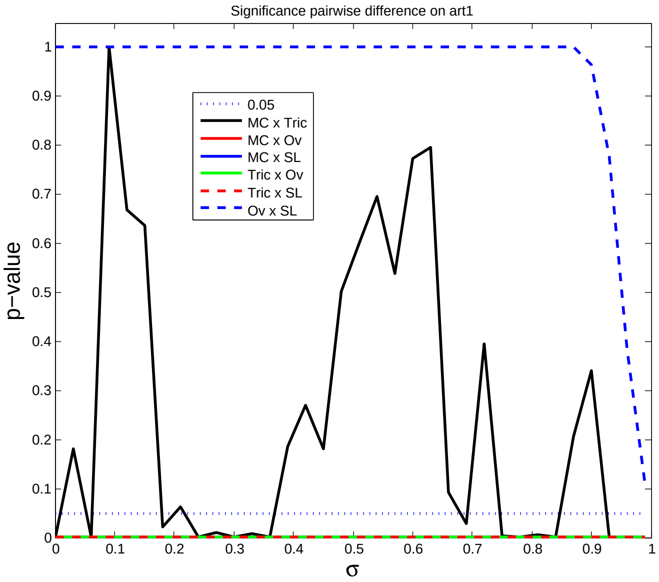


Figure 19 – Two-sided Wilcoxon Rank Sum Test at 5% significance level on CE metric, on *art1*.

Fig. 19 shows the statistical significance of the pairwise difference between the solutions, using a Two-sided Wilcoxon Rank Sum Test. When a curve is below the 0.05 threshold, it means that the two compared solutions do differ significantly. We can see that for all levels of noise, just the solutions of single linkage and aggregation by overlapping did not show statistical difference. For most levels of noise, the solutions of Triclustering and MicroCluster also did not exhibit a significant difference.

6.3.2 Results on *art2*

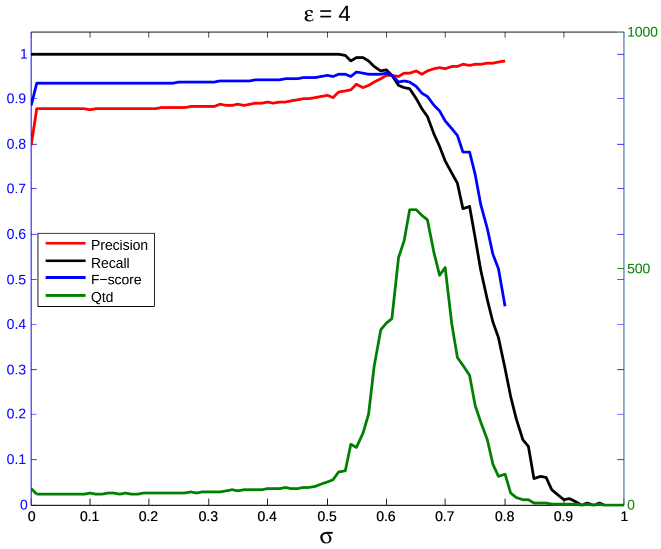


Figure 20 – Results produced by RIn-Close on *art2*. The scale on the right refers to the quantity of biclusters on the solution.

In Figure 20, we can see the results of RIn-Close when $\epsilon = 4$. In this dataset, the impact of noise starts earlier than on dataset *art1*. We can see that there is a high decrease on the *Recall*, simultaneously with an increase on the quantity of biclusters, when $\sigma \approx 0.55$. This may seem a little contradictory, as we have more biclusters, with a penalized *Recall*.

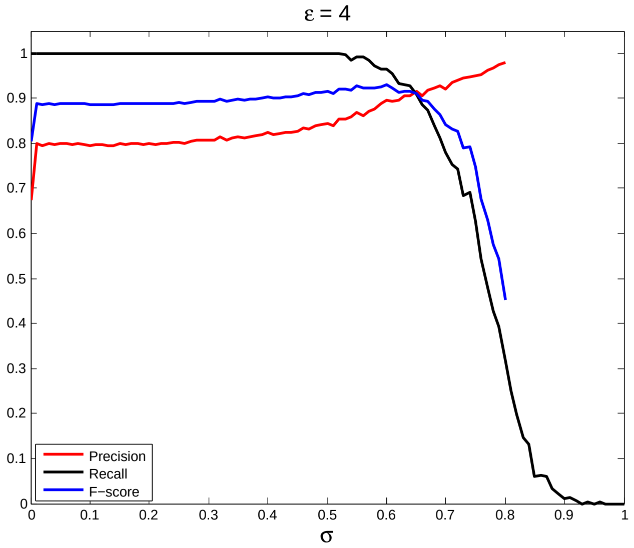


Figure 21 – Aggregation with single linkage using Hamming distance on *art2*.

Figure 21 shows the quality of the results of the aggregation with single linkage. Despite the impacts of noise happening earlier in this dataset, the scenario here is pretty much the same of *art1*: *Precision* decreases a little and *Recall* remains high. It indicates that some intruder elements may be filtered from the biclusters of this solution. This will be verified on experiment 3.

In Figure 22, we can see the results of aggregation by overlapping. When the rate of overlapping was 60% and 65% the results were identical, and very similar to these from 70%. So, we will show only the results when the rate of overlapping is greater or equal to 70%. But again, the scenario is very similar to the one obtained with single linkage. We can see that the aggregation was able to get the true number of biclusters when the noise was not so high, independently of the rate of aggregation. This also indicates the high degree of overlapping of the enumerated biclusters, even when the rate was 95% the aggregation was able to get the correct quantity. In Figures 22c, 22d, 22e, and 22f, we can see that when $\sigma \approx 0.6$ the quantity of biclusters starts to increase. If we compare this result with the one presented in Figure 21, we can see that we have practically the same *Precision* and *Recall*, but the aggregation with single linkage has the correct number of biclusters for all variance of the noise, which is an advantage.

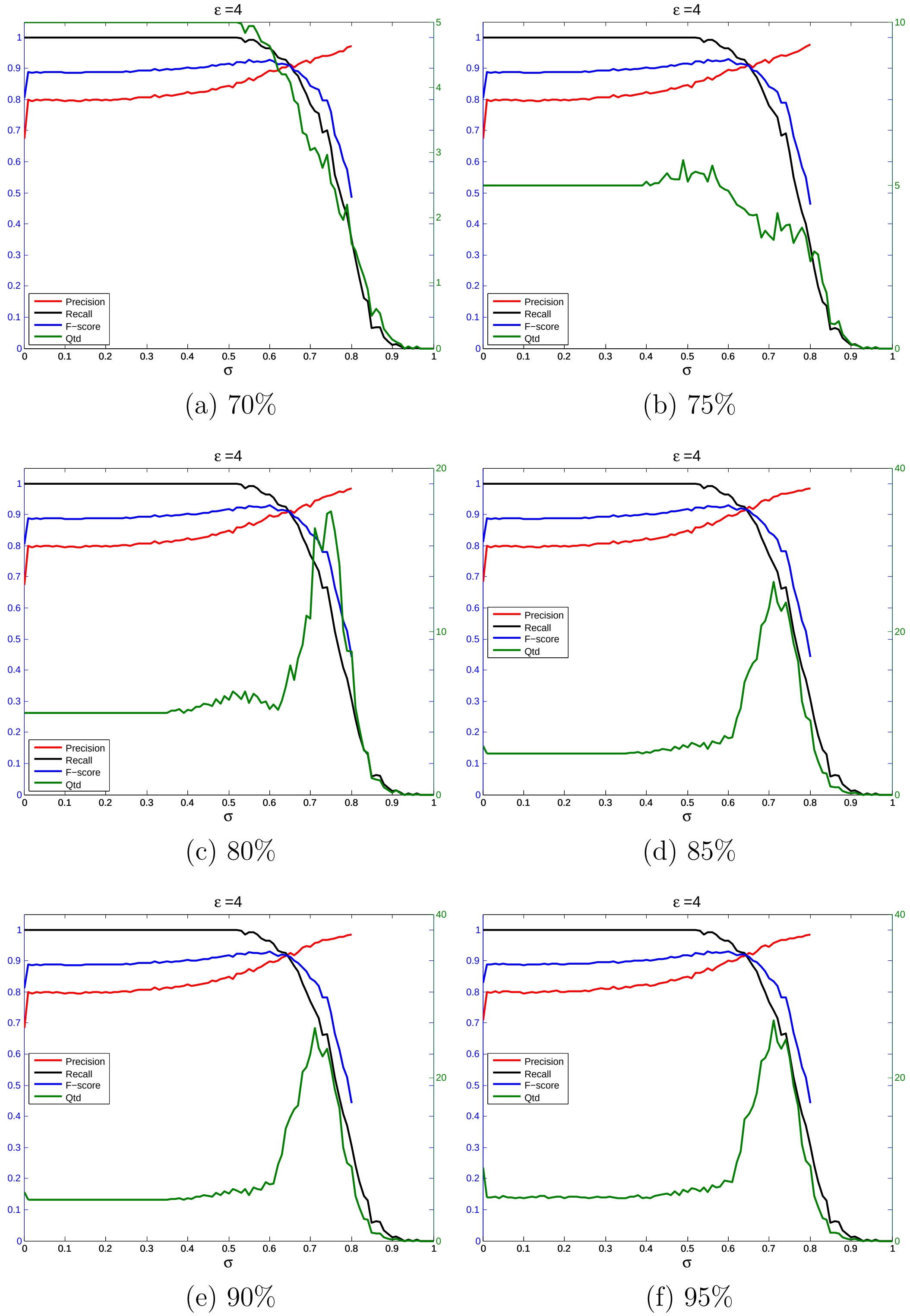
Figure 23 shows the results after the aggregation with MicroCluster. In this dataset, the MicroCluster aggregation got better results when compared to the ones obtained in *art1*. In all configurations, we can see that the *Precision* is higher than the one of RIn-Close, but the *Recall* decreased a little. Considering the quantity of biclusters, we can see that when the deleting operation came first, the aggregation reached a quantity closer to the true number of biclusters.

Figure 24 shows the quality of the results of the aggregation using the triclustering algorithm. Again, the *Precision* is high and the *Recall* decreased considerably, what indicates that the biclusters of this solution are very conservative with their elements. In other words, the objects (attributes) that are in any bicluster of this solution should really be part of that bicluster. But this result is too conservative, not including a good percentage of the elements that should be part of a bicluster.

Now we will compare the best results from each agglomeration, using the *CE* metric and verify if the results have a significant difference among each other. They are: *a*) single linkage; *b*) aggregation by overlapping with 75%; *c*) MicroCluster with $\eta = \gamma = 0.15$, deleting operation first; and *d*) triclustering.

Figure 25 shows the *CE* metric for the best solutions of each algorithm. We can see that the aggregation with single linkage and by overlapping had the best results on this metric, and they exhibit a similar pattern again.

Fig. 26 shows the significance pairwise difference of the results on *art2*. We can see that for most levels of noise, only the comparisons between aggregation by overlapping versus



(a) 70%

(b) 75%

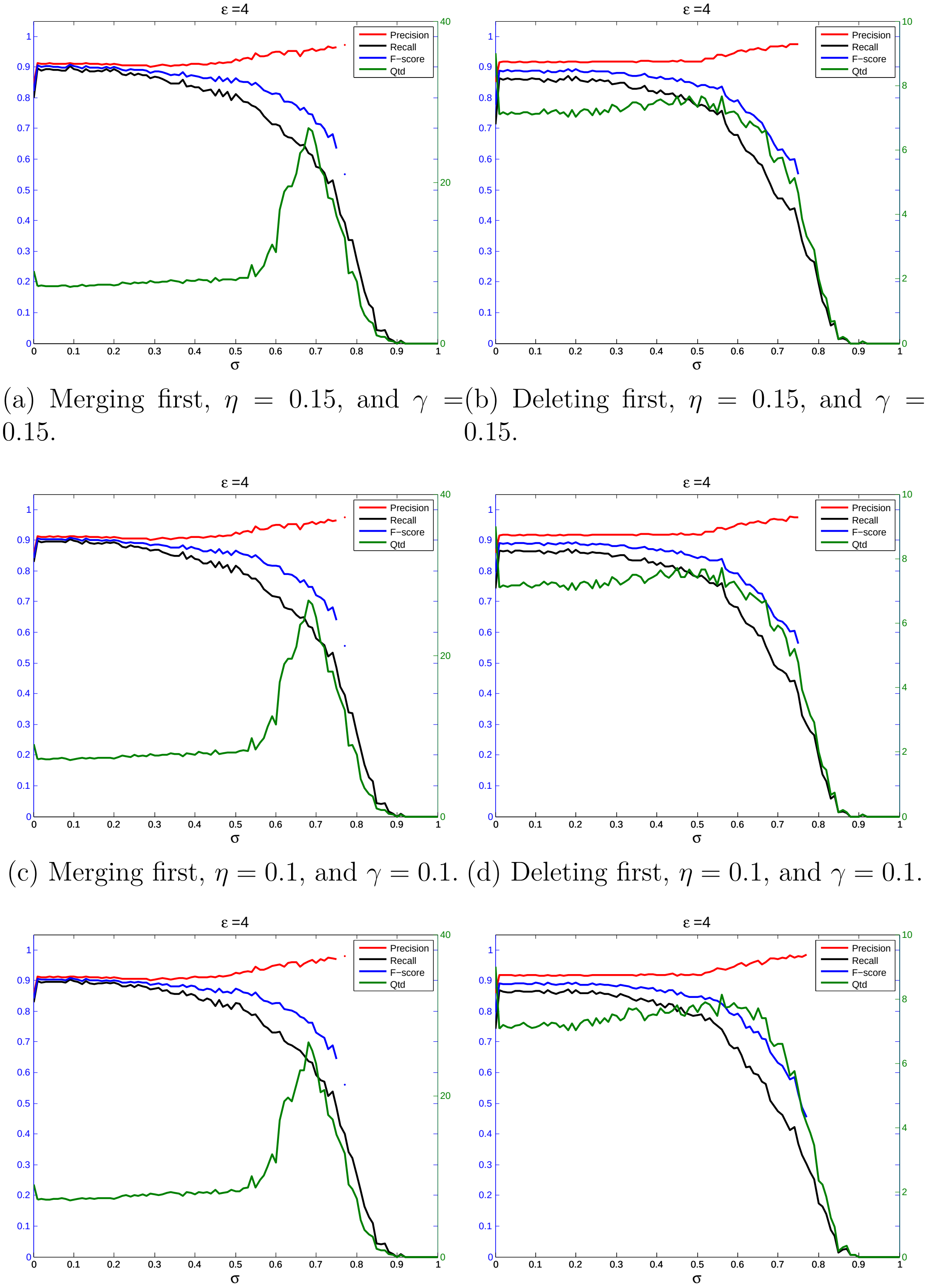
(c) 80%

(d) 85%

(e) 90%

(f) 95%

Figure 22 – Results from the aggregation by overlapping on *art2*. The scale on the right refers to the quantity of biclusters on the solution.



(a) Merging first, $\eta = 0.15$, and $\gamma = 0.15$.

(b) Deleting first, $\eta = 0.15$, and $\gamma = 0.15$.

(c) Merging first, $\eta = 0.1$, and $\gamma = 0.1$.

(d) Deleting first, $\eta = 0.1$, and $\gamma = 0.1$.

(e) Merging first, $\eta = 0.05$, and $\gamma = 0.05$.

(f) Deleting first, $\eta = 0.05$, and $\gamma = 0.05$.

Figure 23 – Results from the aggregation using MicroCluster on *art2*. The scale on the right refers to the quantity of biclusters on the solution.

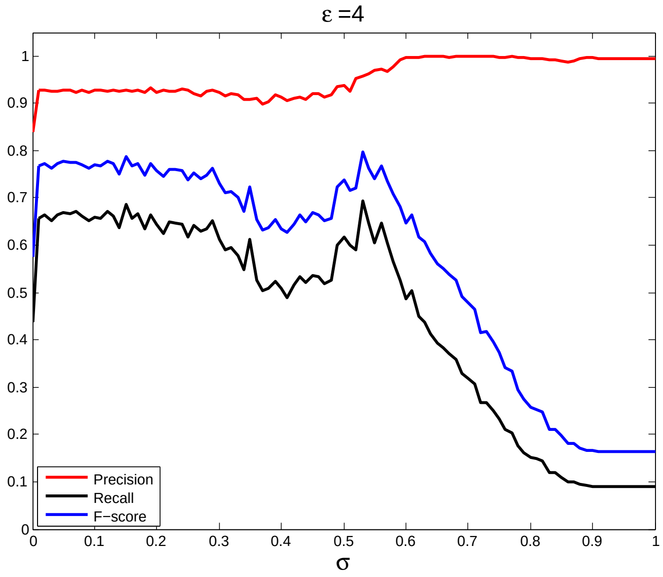


Figure 24 – Aggregation with the triclustering algorithm on *art2*.

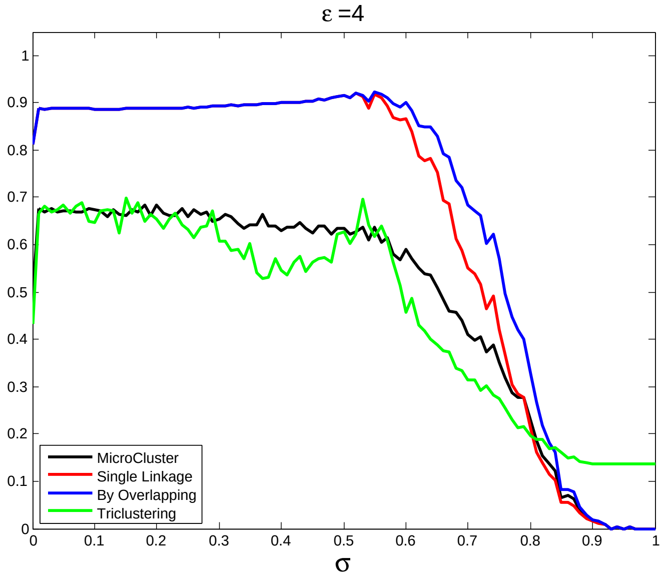


Figure 25 – *CE* of the best results of aggregation on *art2*.

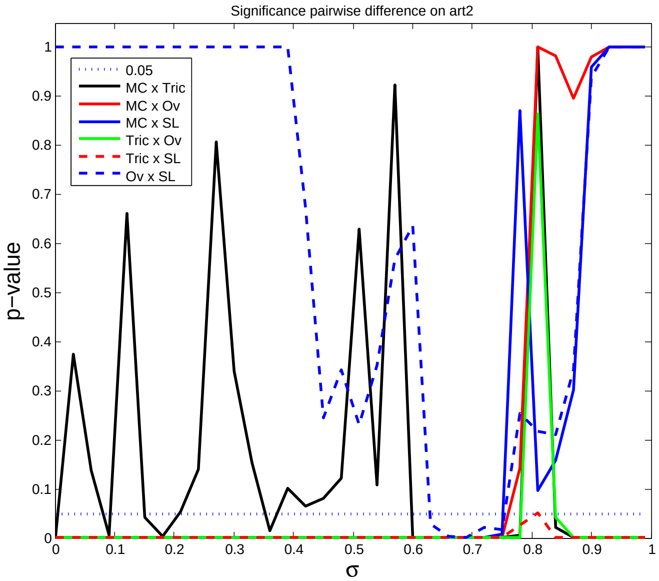


Figure 26 – Two-sided Wilcoxon Rank Sum Test at 5% significance level on CE metric, on *art2*.

single linkage, and MicroCluster versus Triclustering, exhibit significant difference.

6.3.3 Results on *art3*

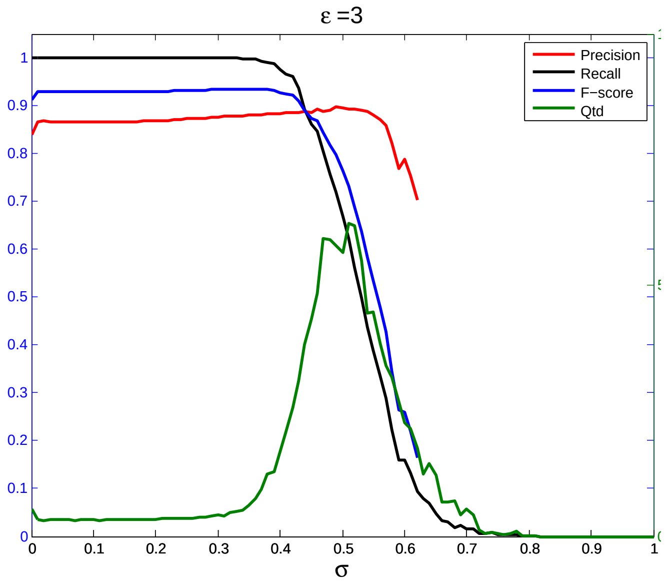


Figure 27 – Results produced by RIn-Close on *art3*. The scale on the right refers to the quantity of biclusters on the solution.

Figure 27 shows the results of RIn-Close when $\epsilon = 3$. The effects of the noise here are more severe than on the other datasets. When $\sigma \approx 0.4$, the *Precision* already starts to decrease. This is the result that we will try to improve and we must remember that this is the most challenging among the artificial datasets.

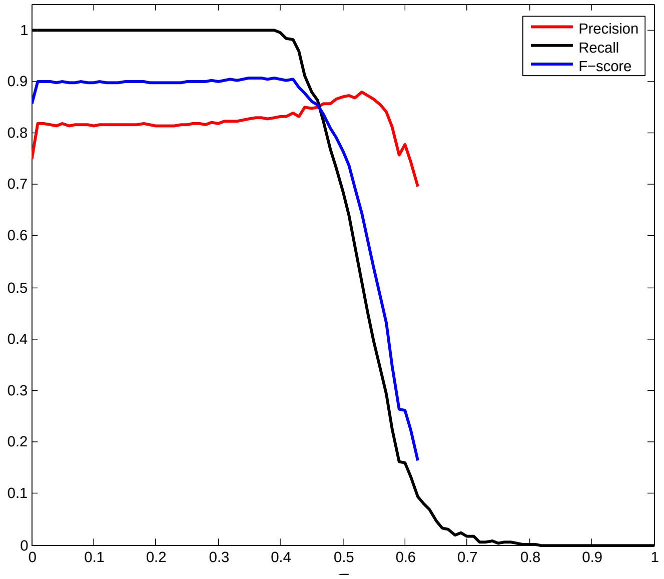


Figure 28 – Aggregation with single linkage using Hamming distance on *art3*.

Figure 28 is showing the quality of the aggregation with single linkage. We can see a little decrease on the *Precision* but the *Recall* still high at the beginning. The scenario seems to be very similar to the one provided by the other datasets, and if we consider that this dataset is the most challenging, we can realize the positive effect promoted by aggregation.

Figure 29 shows the results obtained on the aggregation by overlapping. On the other datasets, we reported only the results when the rate of aggregation was greater than or equal to 70%. Thus we will follow the same procedure for this dataset. When the rate of overlapping was less than or equal to 90%, and the noise was not high enough ($\sigma < 0.3$), the aggregation returned on average 13 biclusters, that is less than the true number of biclusters. However, as this dataset has a pair of biclusters that have 60% of overlapping, probably these biclusters are being merged on the final solution. In Figure 29f, we can see the details of the solution when the rate of overlapping was 95%. We can see that the *Precision* of this result is a little higher than the one for the other rates. But up to $\sigma \approx 0.3$, the mean quantity of bicluster is 17, which is higher than the true number of biclusters. We can compare the *Precision* of these results with the *Precision* of the aggregation with single linkage, presented in Figure 28. The latter has the proper number of biclusters in its solution and has very similar *Precision* and *Recall* when compared to the former.

Figure 30 shows the results when the aggregation was performed by the MicroCluster algorithm. We can see that when the deleting operation was executed first, the final number of biclusters were more stable and the *Precision* was a little higher. We can also see that these results did not show a visual difference in both *Precision* and *Recall*.

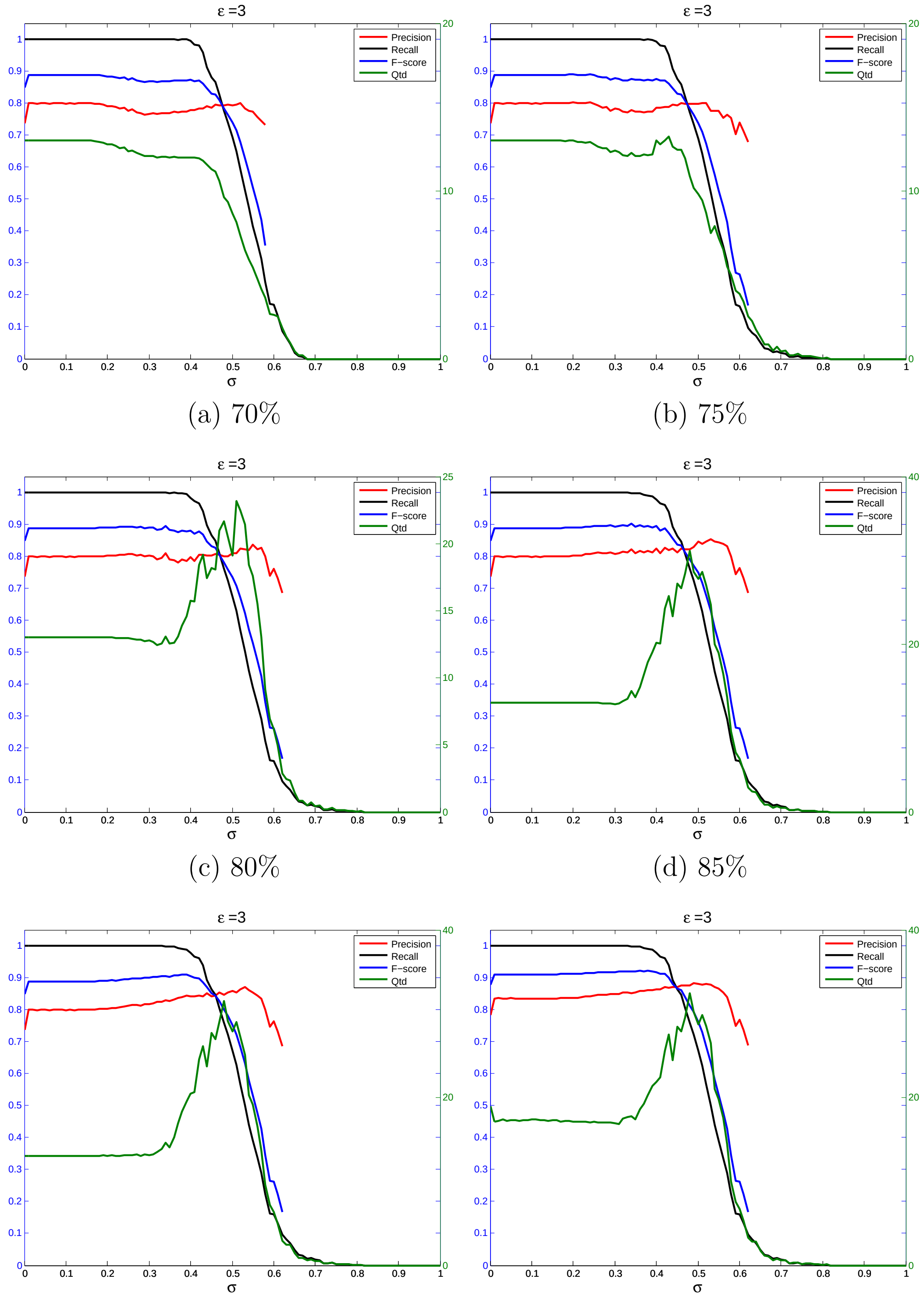
As presented in Figure 31, the aggregation with the triclustering algorithm could not get a good result when compared to the alternatives previously exposed, even with k being set to the true number of biclusters in the dataset.

We will then compare the best results from each agglomeration using the *CE* metric, and verify if the results have significant difference among each other. They are: *a*) single linkage; *b*) aggregation by overlapping with 75%; *c*) MicroCluster with $\eta = \gamma = 0.15$, deleting operation first; and *d*) triclustering.

Figure 32 shows the *CE* metric for the best solutions of each algorithm. We can see that, except for triclustering, all the other results are similar, and the aggregation by overlapping seems to be the most stable of the comparison.

Fig. 33 shows the significance comparison for each algorithm. As we can see, most of the results did not show significant difference between them, except when the level of noise assumed high values. But in this case, where the noise is high and the quality of the solutions is poor, this indication of difference is not relevant.

In this experiment, we could see that the aggregation in fact was able to get much less biclusters with a comparable or better quality. We could notice that our proposals can



(a) 70%

(b) 75%

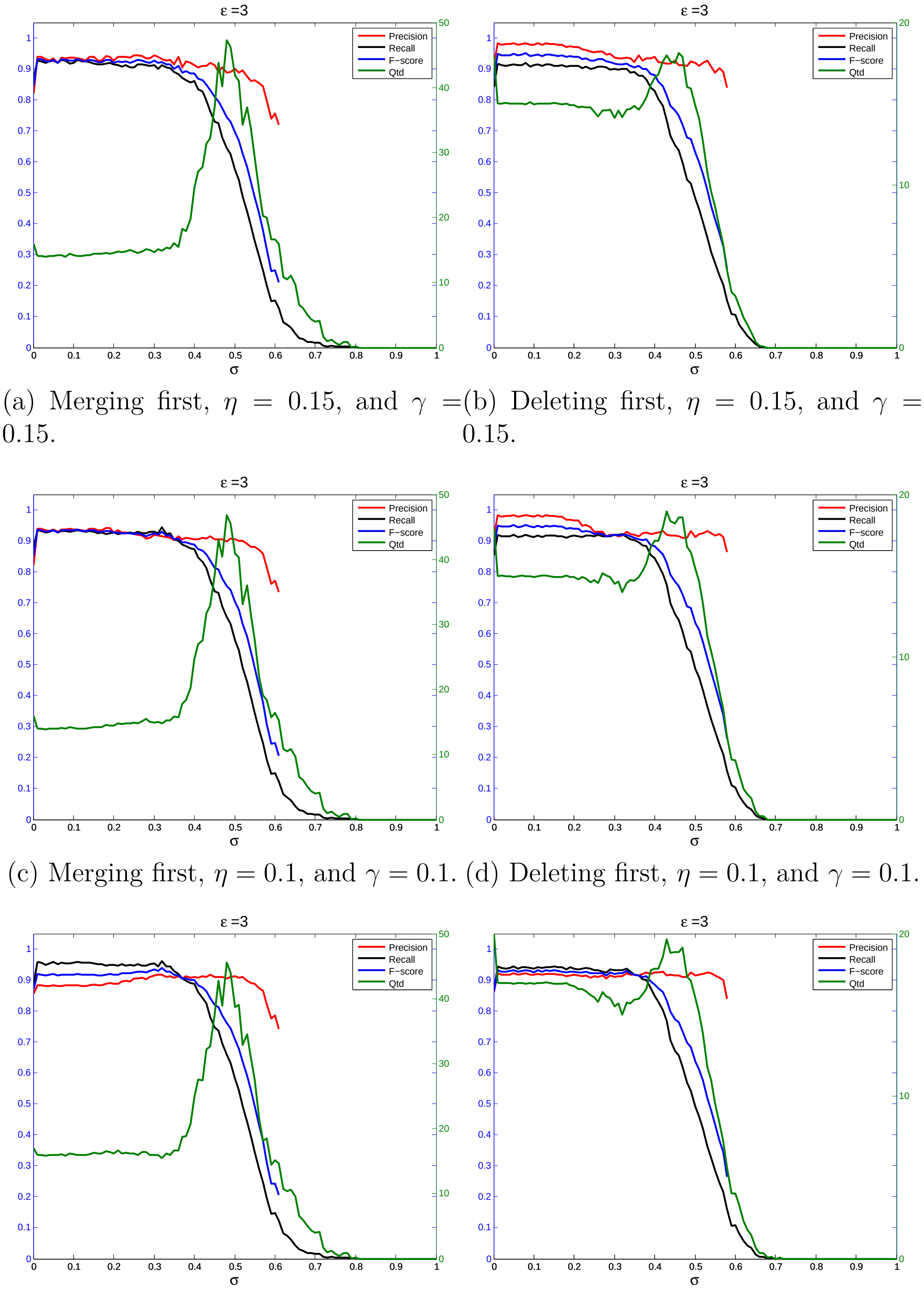
(c) 80%

(d) 85%

(e) 90%

(f) 95%

Figure 29 – Results from the aggregation by overlapping on *art3*. The scale on the right refers to the quantity of biclusters on the solution.



(a) Merging first, $\eta = 0.15$, and $\gamma = 0.15$.

(b) Deleting first, $\eta = 0.15$, and $\gamma = 0.15$.

(c) Merging first, $\eta = 0.1$, and $\gamma = 0.1$.

(d) Deleting first, $\eta = 0.1$, and $\gamma = 0.1$.

(e) Merging first, $\eta = 0.05$, and $\gamma = 0.05$.

(f) Deleting first, $\eta = 0.05$, and $\gamma = 0.05$.

Figure 30 – Results from the aggregation using MicroCluster on *art3*. The scale on the right refers to the quantity of biclusters on the solution.

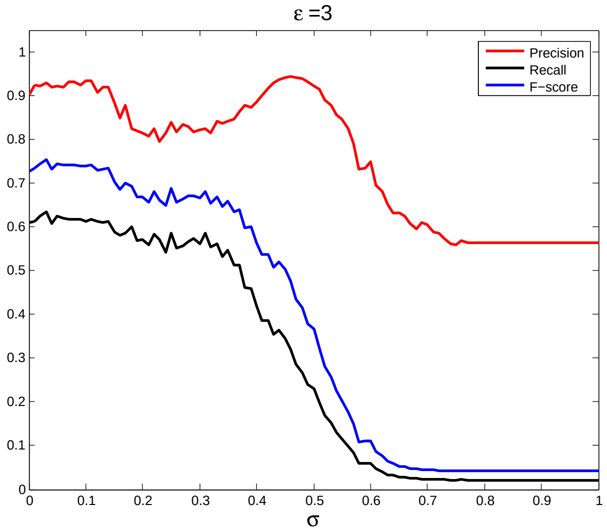


Figure 31 – Aggregation with the triclustering algorithm on *art3*.

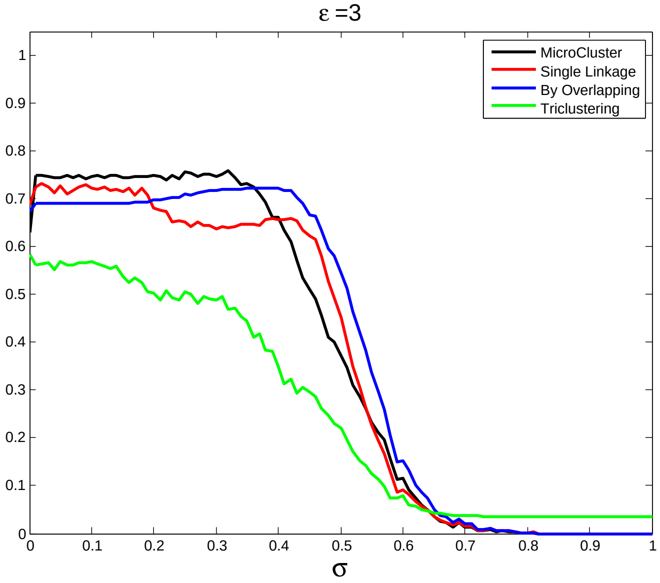


Figure 32 – *CE* of the best results of aggregation on *art3*.

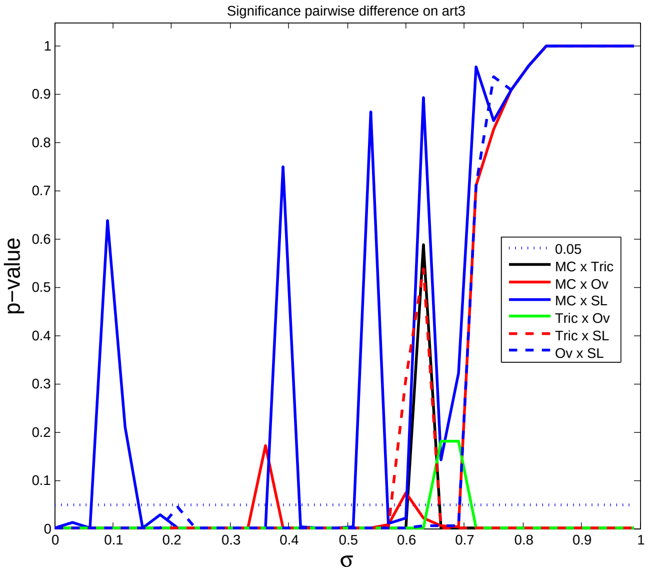


Figure 33 – Two-sided Wilcoxon Rank Sum Test at 5% significance level on CE metric, on *art3*.

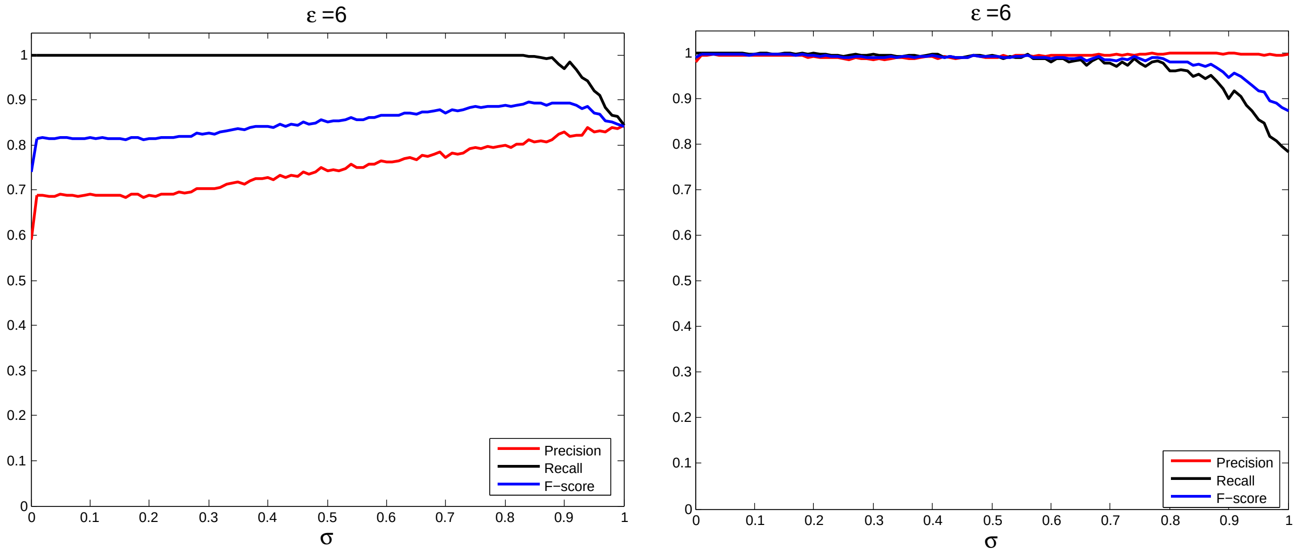
get better results if we add a step of outlier removal on the final results. This will be tested on the next experiment.

6.4 Experiment 3: The impact of outlier removal on aggregation

In the previous experiment, both the aggregation with single linkage and the aggregation by overlapping presented a high rate of *Recall* and the *Precision* was not as high. It indicates that the biclusters were bigger than they should be. In other words, the aggregation was forcing the biclusters to include more objects and / or attributes, because its unique operation was the union of objects and attributes. The quality of the result could be improved if these elements, that should not be included in any bicluster, were removed in the final result.

This third experiment intends to verify the impact of outlier removal after the aggregation of the enumerated results considering our proposals. The method of outlier removal was explained in Section 5.3. After getting the results of the outlier removal step on our proposals, we will again compare them with the results from the other algorithms of the experiment 2, using the *CE* metric. We will also run a two-sided Wilcoxon Rank Sum Test at 5% significance level on the mean of the *CE* metric, to verify if the results have significant difference among each other. We will again group the results by dataset.

6.4.1 Results on *art1*



(a) Before outlier removal

(b) After outlier removal

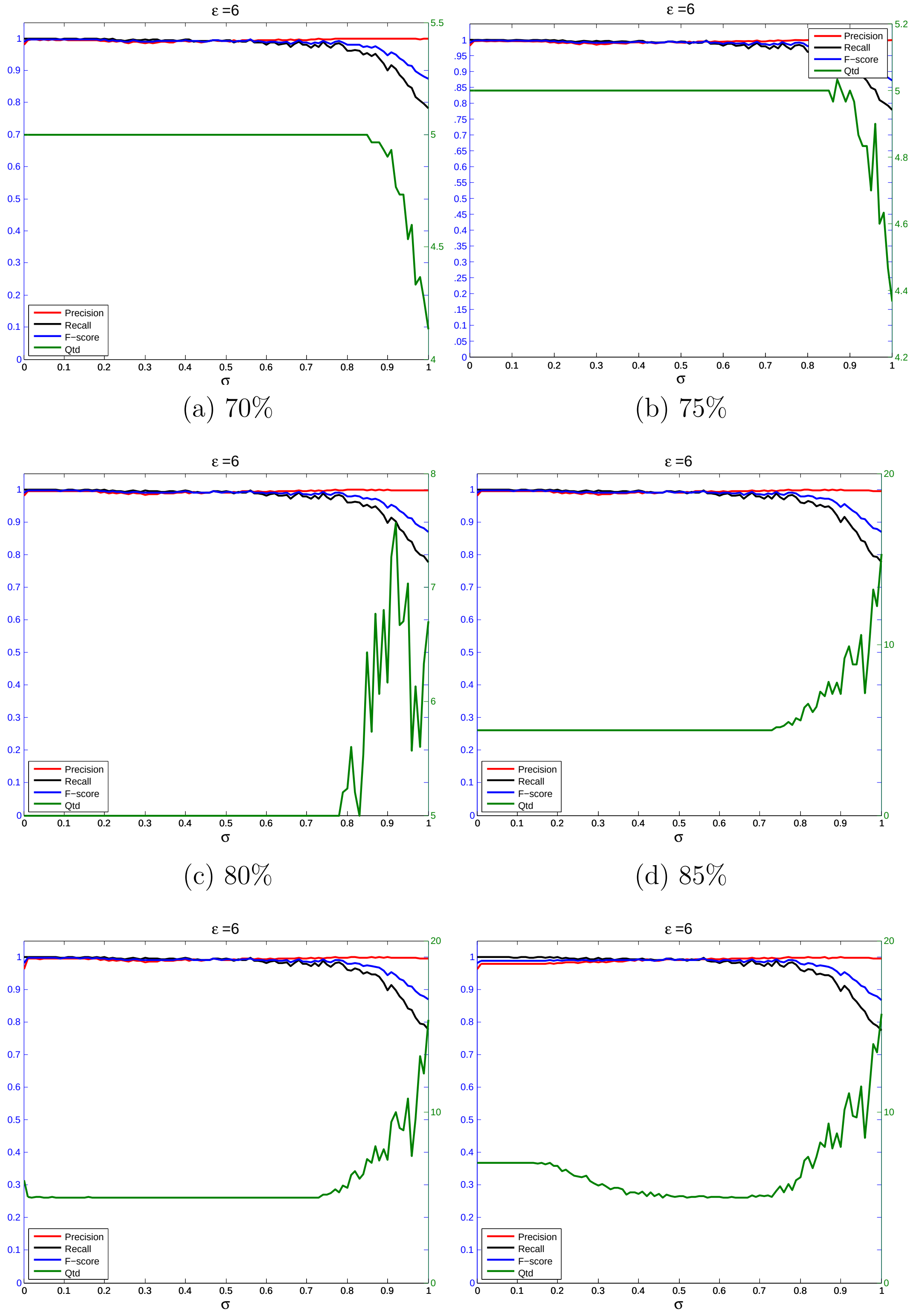
Figure 34 – Aggregation with single linkage using Hamming distance on *art1*, before and after outlier removal and as a function of the noise standard deviation.

Figure 34b shows the quality of the aggregation with single linkage after outlier removal, while Figure 34a is a repetition of Figure 14, for the ease of the comparison. We can see that for this dataset, the result is close to the maximum achievable performance. The noise only starts to impact when $\sigma \geq 0.8$. The method of outlier removal was able to remove only the objects and / or attributes that should in fact be removed from the biclusters.

Figure 35 shows the quality of the aggregation by overlapping after outlier removal. We can see that, regardless of the chosen rate of overlapping, all results were again close to the maximum achievable performance.

We will then compare the best results from each aggregation procedure using the CE metric, and verify if the results have significant pairwise difference. As for this dataset our proposals exhibit high performance, we will choose the same results that we have chosen on the previous experiment, except that now we will use the results after outlier removal. The comparison will include the following contenders: *a*) single linkage after outlier removal; *b*) aggregation by overlapping with rate of 70% after outlier removal; *c*) MicroCluster with $\eta = \gamma = 0.15$, deleting operation first; and *d*) triclustering.

In Figure 36 we can see that our proposals exhibit a performance even better than that produced without outlier removal, being both again very similar among each other. In fact, they did not show statistical difference in any level of noise, as we can see on Fig. 37.



(a) 70%

(b) 75%

(c) 80%

(d) 85%

(e) 90%

(f) 95%

Figure 35 – Results from the aggregation by overlapping on *art1*, after outlier removal, and as a function of the noise standard deviation. The scale on the right refers to the quantity of biclusters on the solution.

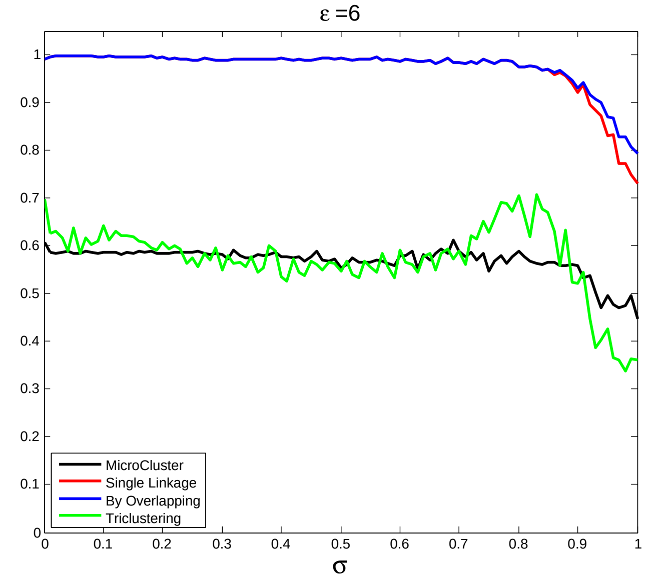


Figure 36 – *CE* of the best results of aggregation on *art1*.

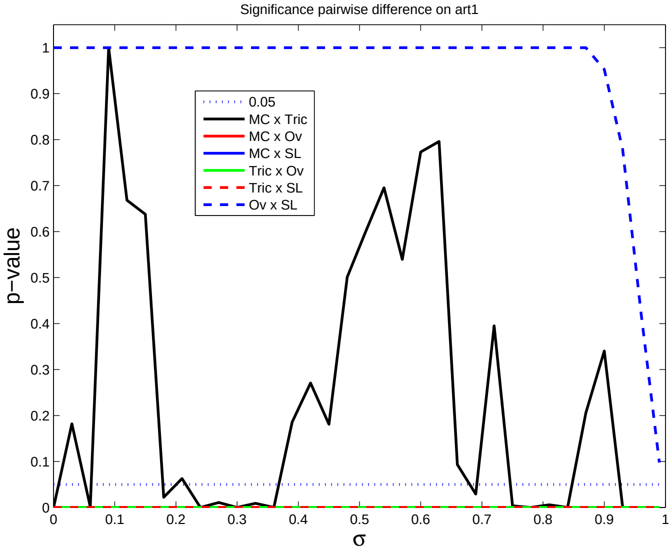
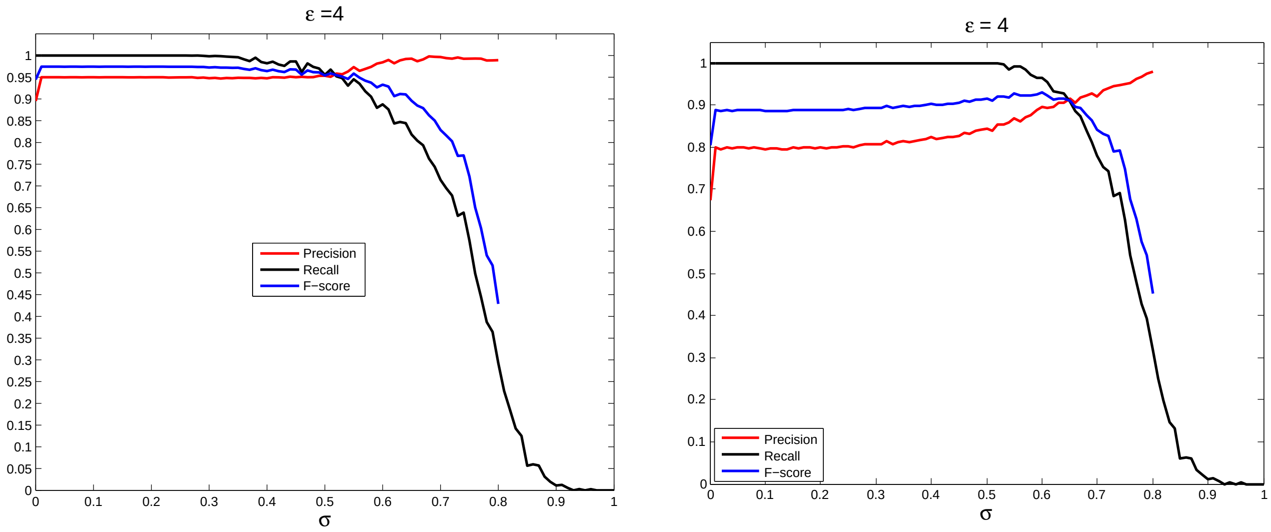


Figure 37 – Two-sided Wilcoxon Rank Sum Test at 5% significance level on CE metric, on *art1*.

6.4.2 Results on *art2*

In the previous subsection, we could realize that the outlier removal was very efficient for dataset *art1*. For dataset *art2* we will follow the same protocol.



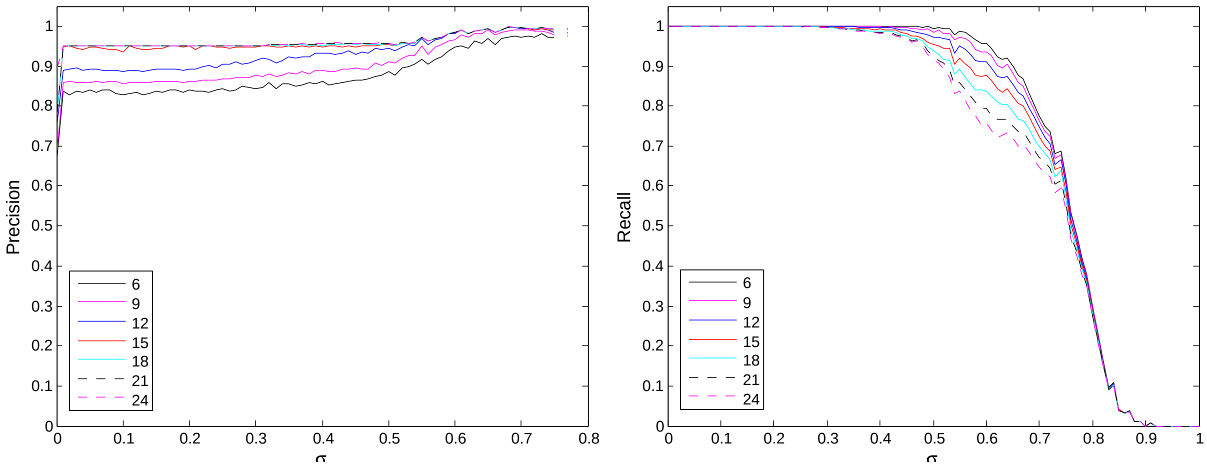
(a) After outlier removal

(b) Before outlier removal

Figure 38 – Aggregation with single linkage using Hamming distance on *art2*, before and after outlier removal, and as a function of the noise standard deviation.

Figure 38a shows the quality of the aggregation with single linkage after outlier removal, while Figure 38b is a repetition of Figure 21, for the ease of the comparison. We can see that the step of removing outliers was able to increase the *Precision* of the result without decreasing the *Recall*. It makes this result better and concordant with our goal: increasing the *Precision* without decreasing the *Recall*, with the proper number of biclusters.

The *Precision* is also close to 1, but not so close to the maximum achievable performance as it was for *art1*. We interpreted this result as an indication that the outlier removal could be more aggressive, removing more elements than it is currently removing. It would increase the *Precision* even more. So we modified the outlier removal method as follows: instead of marking for removal the elements that were below the mean minus on standard deviation, we supplied a percentile and all elements below that percentile should be marked for removal. Assuming a normal distribution, the mean minus one standard deviation is equivalent to -1 *z-score*, that in percentile is approximately to 15.8655. Figure 39 shows the quality of the aggregation with single linkage and several values for outlier removal based on percentile. Comparing the *Precision* (Figure 39a) and *Recall* (Figure 39b), we can see a trade-off. For example, when the value of the percentile was 6 we got the worst *Precision* and the best *Recall*. It indicates that when the value of the percentile is lower than the equivalent



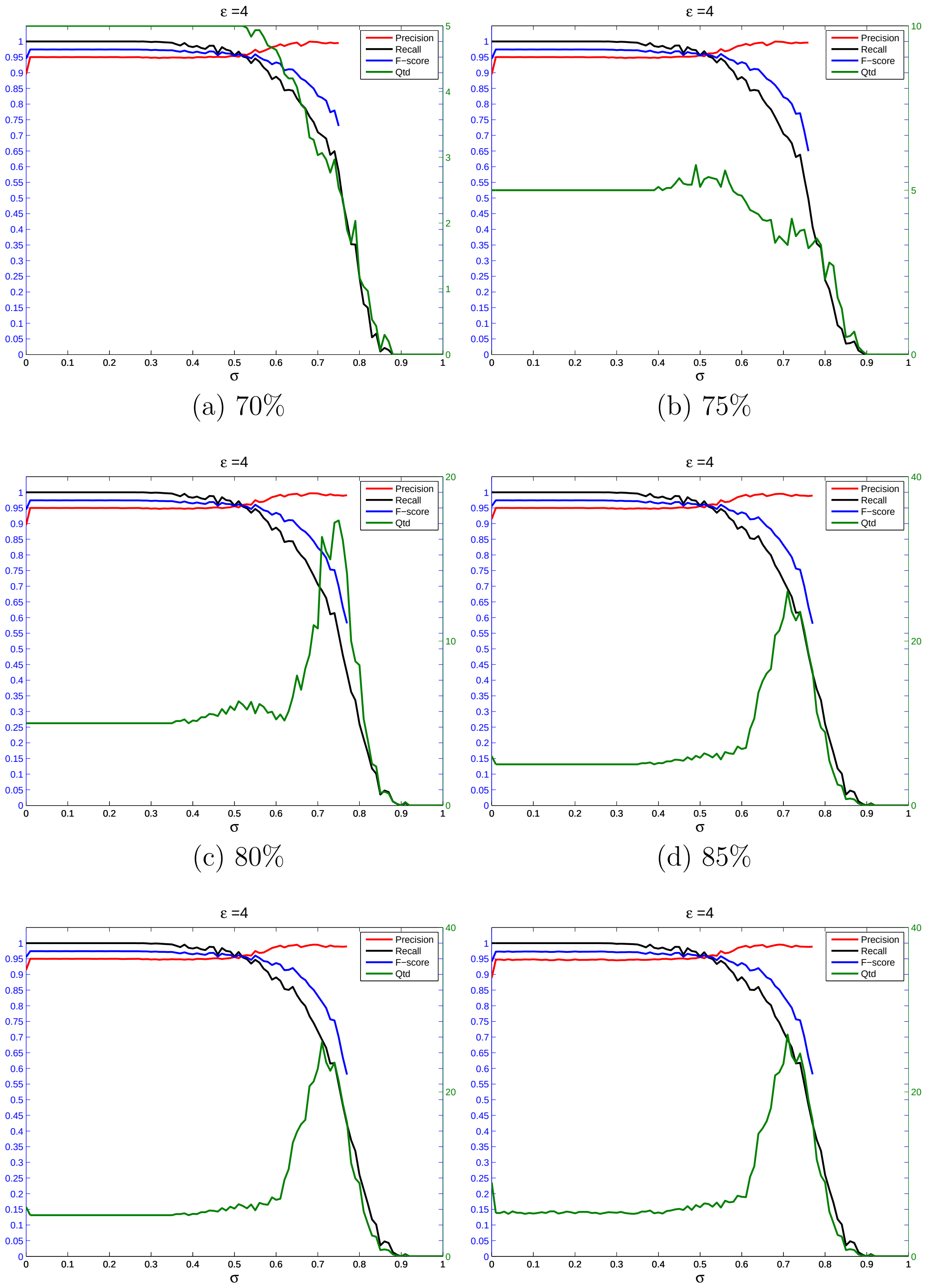
(a) *Precision* (b) *Recall*

Figure 39 – Aggregation with single linkage and outlier removal, and as a function of the noise standard deviation. Each curve is parameterized by the percentile.

to one standard deviation (thus reducing the action of outlier removal), we are removing fewer elements, decreasing the *Precision*. On the other hand, when the value of the percentile was 24 we got the best *Precision* and the worst *Recall*. It indicates that when the value of the percentile is higher than the equivalent to one standard deviation (thus increasing the action of outlier removal), we are removing more elements, including the ones that should be removed (increasing the *Precision*) and the ones that should not (decreasing the *Recall*). We can also see in Figure 39a that the *Precision* increases a little more between the values 12 and 15. The decrease on *Recall* for the same values is not so great. This may indicate that 15 is a good choice for the percentile, which is very close to the equivalent to one standard deviation, that we were using before.

Figure 40 presents the results of the aggregation by overlapping after outlier removal. We can see that the different rates did not change too much the number of final biclusters, and did not lead to significant differences between the rates of *Precision* and *Recall*. However, when the rate was 70% (Figure 40a) the quantity of biclusters were a little more stable than for the other rates. We were also able to improve the *Precision* without decreasing the *Recall*, reaching our goal with the aggregation of this dataset.

We will then compare the best results from each aggregation procedure using the *CE* metric, and verify if the results have a significant pairwise difference. We will choose the same results that we have chosen on the previous experiment, except that now we will use the results after outlier removal. The comparison will include the following contenders: *a*) single linkage after outlier removal; *b*) aggregation by overlapping with rate of 75% after outlier



(a) 70%

(b) 75%

(c) 80%

(d) 85%

(e) 90%

(f) 95%

Figure 40 – Results from the aggregation by overlapping on *art2*, after outlier removal, and as a function of the noise standard deviation. The scale on the right refers to the quantity of biclusters on the solution.

removal; *c*) MicroCluster with $\eta = \gamma = 0.15$, deleting operation first; and *d*) triclustering.

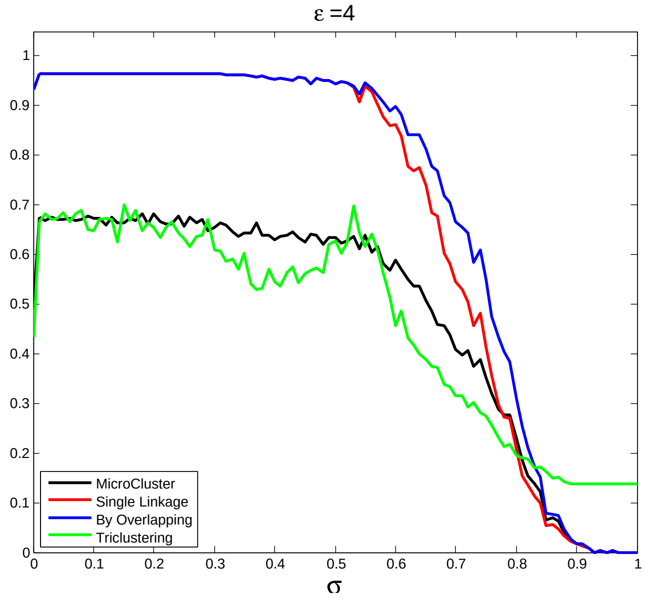


Figure 41 – *CE* of the best results of aggregation on *art2*.

Figure 41 shows behavior of the *CE* metric for the best solutions of each algorithm. We can see that the aggregation with single linkage and by overlapping produced the best results on this metric, and they seem very related again.

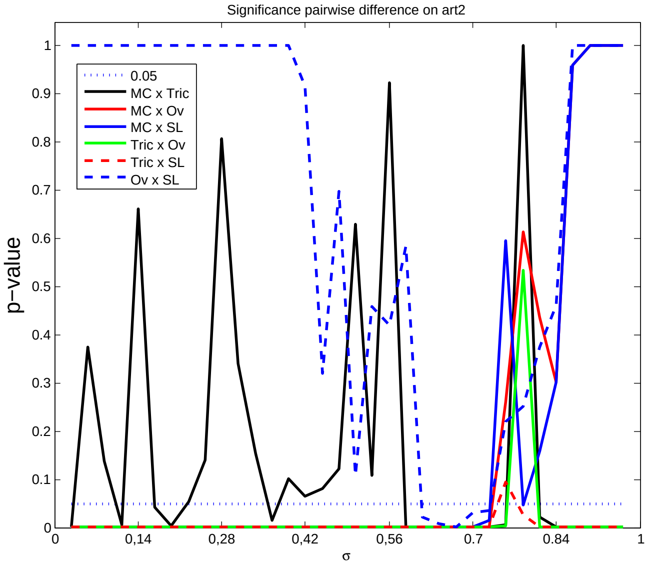
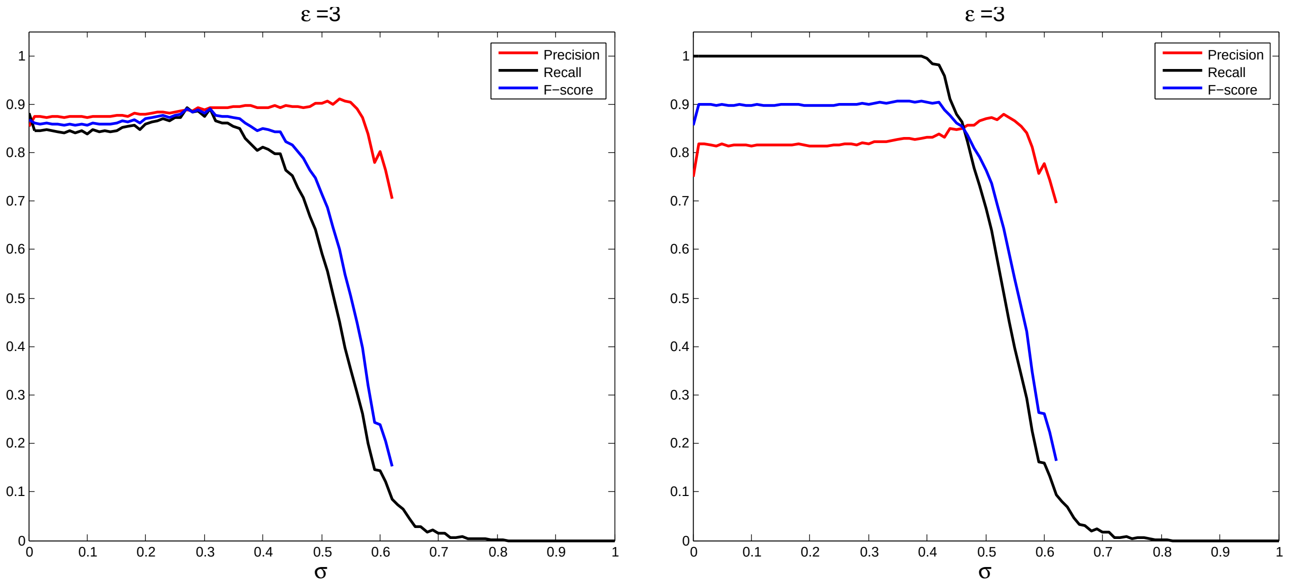


Figure 42 – Two-sided Wilcoxon Rank Sum Test at 5% significance level on CE metric, on *art2*.

Aggregation by single linkage and by overlapping did not show statistical difference for most levels of noise, as we can see on Table 42. As the results from MicroCluster and triclustering are the same of the experiment 2, they also did not show significant pairwise difference for most levels of noise.

6.4.3 Results on *art3*



(a) After outlier removal

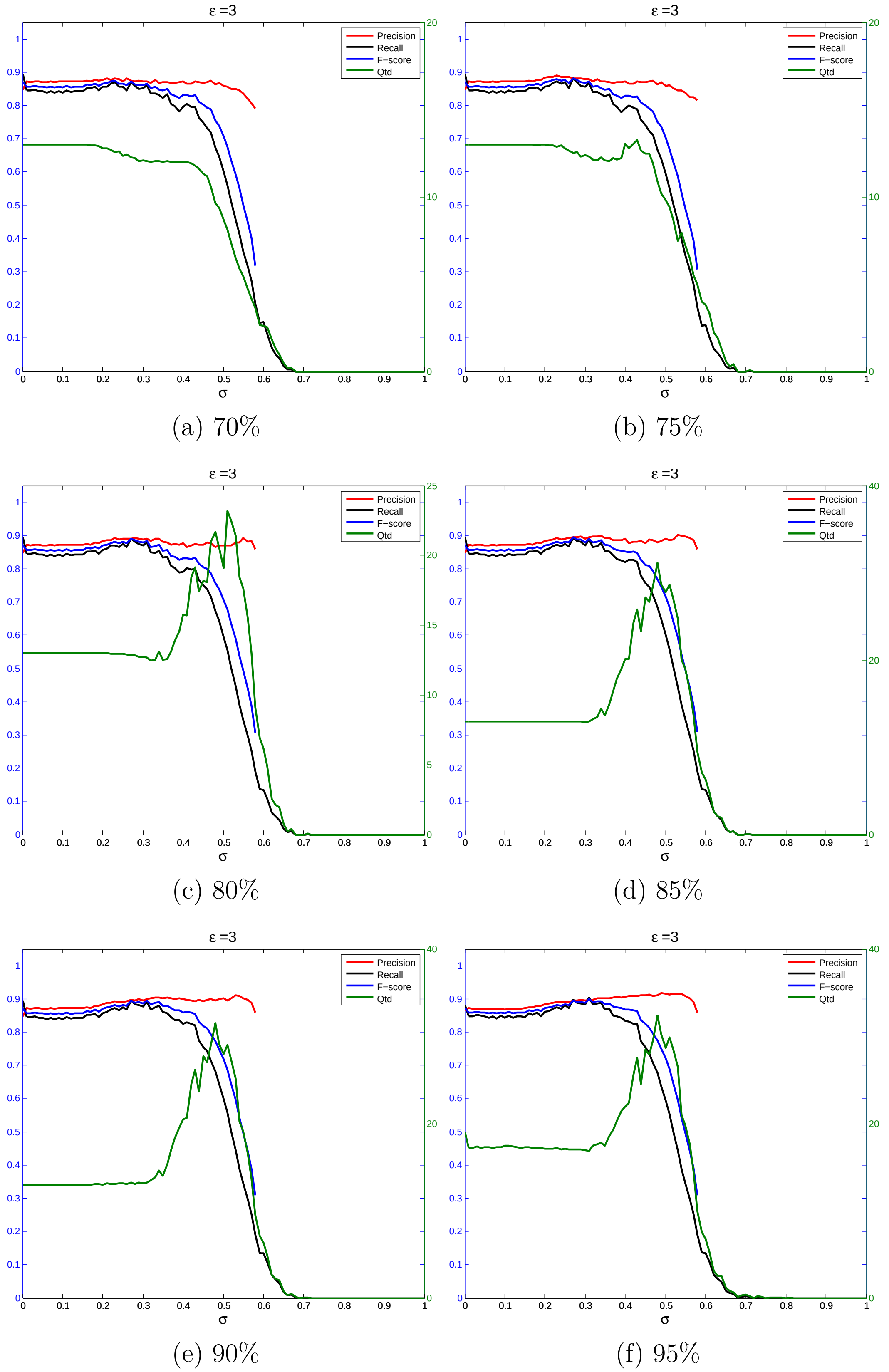
(b) Before outlier removal

Figure 43 – Aggregation with single linkage using Hamming distance on *art3*, before and after outlier removal, and as a function of the noise standard deviation.

Figure 43a shows the quality of the aggregation with single linkage after outlier removal, while Figure 43b is a repetition of Figure 28, for the ease of the comparison. We can see that the *Recall* decreased a little and the *Precision* did not increase. This indicates that the outlier removal step in this solution did not remove elements that should be removed (*Precision* did not increase), but instead it removed a few elements that should not be removed (decrease on *Recall*). So, for this dataset the outlier removal step did not improve the quality of the solution, promoting instead a small decrease in the *Recall*.

Figure 44 presents results of the aggregation by overlapping, after the outlier removal step. When comparing with Figure 29, we can see that the *Precision* increased a little, and the *Recall* decreased.

For this dataset, the step of outlier removal was not able to improve the final result. We believe that this is due to the design of the dataset, where a single bicluster overlaps more than 60% of its area with another two. However, this step is still important, as it has the potential to remove objects and / or attributes that should not be included on the solution,



(a) 70%

(b) 75%

(c) 80%

(d) 85%

(e) 90%

(f) 95%

Figure 44 – Results from the aggregation by overlapping on *art3*, after outlier removal, and as a function of the noise standard deviation. The scale on the right refers to the quantity of biclusters on the solution.

and when the method was not able to do so, it did not impair *Precision* or *Recall* in a relevant way.

We will then compare the best results from each aggregation procedure, using the *CE* metric, and verify if the results have significant pairwise difference. They are: *a*) single linkage after outlier removal; *b*) aggregation by overlapping with rate of 75% after outlier removal; *c*) MicroCluster with $\eta = \gamma = 0.15$, deleting operation first; and *d*) triclustering.

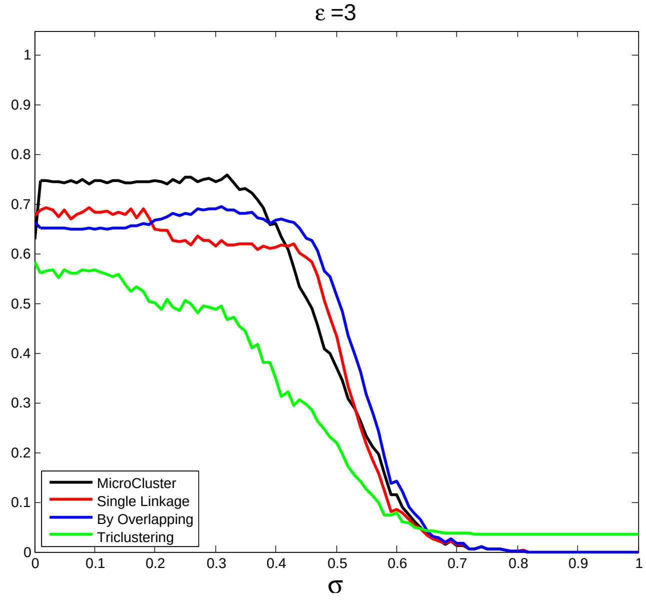


Figure 45 – *CE* of the best results of aggregation on *art3*.

Figure 45 shows the *CE* metric for the best solutions of each algorithm. We can see that, except for triclustering, all the other results showed similar results, with MicroCluster having the highest CE when $\sigma \leq 0.4$, but aggregation with single linkage and by overlapping having the highest CE when $0.4 < \sigma \leq 0.55$.

Fig. 46 shows the pairwise significance comparison. As we can see, except from the comparisons between Triclustering versus Single Linkage; and Triclustering versus aggregation by overlapping; none of the results showed a significant pairwise difference pattern for most of the levels of noise. In other words, the difference is not stable. As all methods are somehow based on the overlapping between the biclusters, this similarity reinforces the belief that this dataset is the most challenging among the artificial datasets considered here.

The first three experiments already presented were useful to show that the bicluster aggregation not just reduces the quantity of biclusters, but may also increase the quality of the final result. Now we will verify the benefits of the aggregation on real datasets.

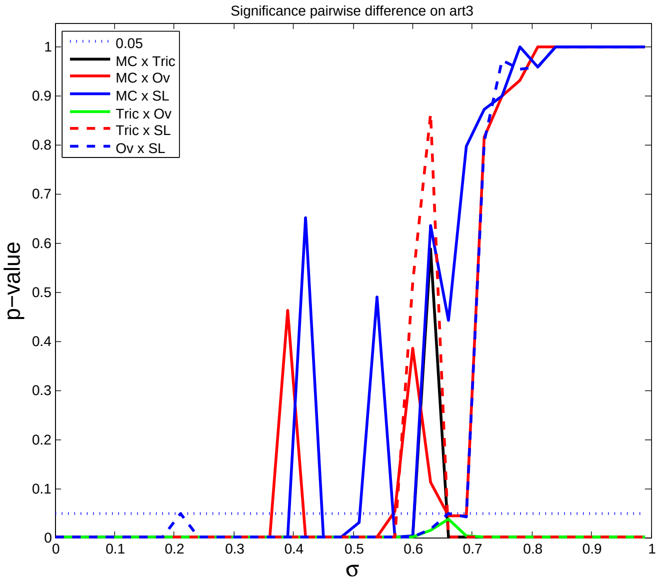


Figure 46 – Two-sided Wilcoxon Rank Sum Test at 5% significance level on CE metric, on *art3*.

6.5 Experiment 4: Application to gene expression data

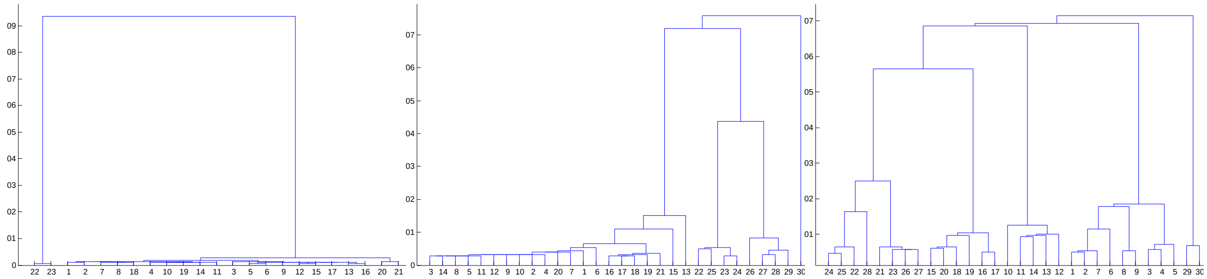
This experiment compares the solutions of the aggregation algorithms on the *GDS2587* dataset, which comes from gene expression data. We will run the RIn-Close algorithm with several values for ϵ and compare the final solutions of each aggregation algorithm.

In this experiment we will run the RIn-Close to enumerate the biclusters of the *GDS2587* dataset. After that, we will compare the agglomerative algorithms. The parametrization of the agglomeration algorithms will follow the same methods of the Experiment 2, and for the triclustering algorithm, we will use the results from the aggregation by overlapping to choose k. We will compare the results with the *gene ontology enrichment analysis*. Moreover, we will describe some of the enriched biclusters. As explained in Section 4.2.1, this analysis is common for gene expression data.

Table 3 – Quantity of biclusters enumerated with the RIn-Close algorithm on the *GDS2587* dataset.

ϵ	Qtd. of biclusters
2.8	23
2.9	2825
3.0	19649

Table 3 shows the quantity of biclusters enumerated for 3 values of ϵ. When $\epsilon < 2.8$, no biclusters were found, and due to memory limits, we decided that the quantity of biclusters when $\epsilon = 3$ is sufficiently large for aggregation.



(a) $\epsilon = 2.8$

(b) $\epsilon = 2.9$

(c) $\epsilon = 3$

Figure 47 – Dendrograms of the aggregation with single linkage, on the *GDS2587* dataset.

Figure 47 shows the dendrograms of the aggregation with single linkage, for each value of ϵ. We can see in Figure 47a that the cut is very straightforward, having 2 very distinct groups. In Figure 47b the cut is also easy, having 4 clear groups of objects. In Figure 47c, we may cut the dendrogram in 5 groups.

Table 4 – Quantity of biclusters for the aggregation by overlapping on the GDS dataset.

Rate	Qtd ($\epsilon = 2.8$)	Qtd ($\epsilon = 2.9$)	Qtd ($\epsilon = 3$)
60 %	2	4	5
65 %	2	4	5
70 %	2	4	5
75 %	2	4	5
80 %	2	4	5
85 %	2	4	8
90 %	2	5	10
95 %	2	9	22

Table 4 shows the quantity of biclusters after the aggregation by overlapping. The results here seem to agree with the solution obtained by the aggregation with single linkage. When $\epsilon = 2.8$, all rates led to 2 biclusters. When $\epsilon = 2.9$ the majority of the rates indicated 4 biclusters, the same reasonable cut of the dendrogram. And when $\epsilon = 3$, the majority of the rates indicated 5 biclusters, that also agrees with the cut of the dendrogram of the aggregation with single linkage. As both solutions are based on the level of overlapping, it seems intuitive that they would reach similar solutions.

Table 5 shows the quantity of biclusters that had objects and / or attributes removed after the outlier removal step. We can see that, when $\epsilon = 3$, the aggregation did not include outliers when the rate was less then or equal to 80%, which indicates that the fragmentation of the biclusters in these settings did not include outliers, only fragmenting inside the bicluster.

Table 5 – Number of biclusters that changed after the outlier removal step.

Rate	Qtd ($\epsilon = 2.8$)	Qtd ($\epsilon = 2.9$)	Qtd ($\epsilon = 3$)
60 %	1 of 2	1 of 4	0 of 5
65 %	1 of 2	1 of 4	0 of 5
70 %	1 of 2	1 of 4	0 of 5
75 %	1 of 2	1 of 4	0 of 5
80 %	1 of 2	1 of 4	0 of 5
85 %	1 of 2	1 of 4	2 of 8
90 %	1 of 2	2 of 5	2 of 10
95 %	1 of 2	4 of 9	3 of 22

We also compared the coverage of each rate of the aggregation by overlapping, and they were always the same, independently of the quantity of final biclusters. For example, when $\epsilon = 2.9$, both the rates 60% and 95% covered the same area of the dataset.

Table 6 – Quantity of biclusters for the aggregation with MicroCluster on the GDS dataset.

Order	η	γ	Qtd $\epsilon = 2.8$	Qtd $\epsilon = 2.9$	Qtd $\epsilon = 3$
DM	0.15	0.15	2	3	7
DM	0.15	0.1	2	5	8
DM	0.15	0.05	2	5	11
DM	0.1	0.15	2	3	7
DM	0.1	0.1	2	5	8
DM	0.1	0.05	2	5	11
DM	0.05	0.15	2	3	7
DM	0.05	0.1	2	5	8
DM	0.05	0.05	2	5	11
MD	0.15	0.15	2	3	7
MD	0.15	0.1	2	5	8
MD	0.15	0.05	2	5	11
MD	0.1	0.15	2	3	7
MD	0.1	0.1	2	5	8
MD	0.1	0.05	2	5	11
MD	0.05	0.15	2	3	7
MD	0.05	0.1	2	5	8
MD	0.05	0.05	2	5	11

Table 6 shows the quantity of biclusters obtained by the aggregation with MicroCluster. When $\epsilon = 2.8$, the solution of MicroCluster agrees with the aggregation with single linkage and by overlapping. The 2 found biclusters were exactly the same, independently of the parametrization. When $\epsilon = 2.9$, we can see that only the parameter γ interfered on the quantity of biclusters. Despite that, the final biclusters were again the same. When $\epsilon = 3$, for each value of γ we have a quantity of biclusters that are identical again.

The triclustering algorithm was configured to find 2 biclusters when $\epsilon = 2.8$, 4 when $\epsilon = 2.9$ and 5 when $\epsilon = 3$.

As all different methods agreed on the final quantity of biclusters, we are left with the GOEA to see the quality of the results.

6.5.1 Gene Ontology Enrichment Analysis

As explained in Section 4.2.1, GOEA is a common analysis for groups of genes obtained by clustering or biclustering in gene expression data.

When $\epsilon = 2.8$, except from triclustering, all the algorithms returned only enriched biclusters. In fact, the four main enriched terms were always the same, sometimes on different orders but with very close p-values. Only the first bicluster from the triclustering algorithm was enriched.

Table 7 shows the main enriched terms of the first bicluster from the aggregation by overlapping with a rate of 70%, after outlier removal, when $\epsilon = 2.8$.

Table 8 shows the main enriched terms of the first bicluster from the aggregation with MicroCluster, when $\epsilon = 2.8$.

Table 9 shows the main enriched terms of the first bicluster from the aggregation with triclustering when $\epsilon = 2.8$.

When $\epsilon = 2.9$, all algorithms returned only enriched biclusters, including triclustering. When $\epsilon = 3$, all algorithms, except for triclustering, returned only enriched biclusters. triclustering returned 4 from 5 enriched biclusters.

In this experiment, we could see that the aggregation was able to significantly reduce the quantity of biclusters, and recovered enriched biclusters.

6.6 Experiment 5: Application to Food dataset

As we have seen in the previous experiment, our proposals for aggregation got only enriched biclusters for the gene expression dataset. In this experiment we will verify how the aggregation changes the coverage of the dataset when compared to the enumeration, considering another real dataset. As the aggregation will severely reduce the quantity of final biclusters, it is important to see if it will leave uncovered areas that were previously covered.

Table 7 – Enrichment analysis of the first bicluster from the aggregation by overlapping with rate of 70%.

GO Term	p-val	counts	definition
GO:0044464	0.00000000	39 / 774	Any constituent part of a cell, the basic structural and functional unit of all organisms. [GOC:jl]...
GO:0044444	0.00000011	19 / 608	Any constituent part of the cytoplasm, all of the contents of a cell excluding the plasma membrane and nucleus, but including other subcellular structures. [GOC:jl]...
GO:0044424	0.00000350	19 / 578	Any constituent part of the living contents of a cell; the matter contained within (but not including) the plasma membrane, usually taken to exclude large vacuoles and masses of secretory or ingested material. In eukaryotes it includes the nucleus and cytoplasm. [GOC:jl]...
GO:0098593	0.00010607	16 / 492	A cup shaped specialization of the cytoskeleton that forms a thin layer located just below the apical mass of mature mucin secretory granules in the cytoplasm of goblet cells of the intestinal epithelium. It consists of an orderly network of intermediate filaments and microtubules. Microtubules are arranged vertically, like barrel staves, along the inner aspect of the theta. Intermediate filaments form two networks: an inner, basketlike network and an outer series of circumferential bundles resembling the hoops of a barrel. [PMID:6541604]...

If this happens, it means that the aggregation may be eliminating objects and / or attributes that can be important for some applications.

In this experiment we will run the RIn-Close algorithm and aggregate the enumerated biclusters of the *Food* dataset. We will then compare the final solutions to see the differences between the results of each agglomerative algorithm. As the true biclusters of this dataset are unknown, we will compare how the aggregation differs from the enumeration in terms of coverage of the dataset. In the comparison, we will report our proposals after the outlier removal step.

However, this concern can be more or less relevant depending on the application. Our proposals may increase the coverage, as their basic operator is the union of sets. Even after the outlier removal step, we can end covering more area than the enumerative solution.

We replicated the experiment from Veroneze *et al.* (2014) on this dataset and we will use $\epsilon = 1.25$ as recommended on that work. In Table 10 we can see the quantity of

Table 8 – Enrichment analysis of the first bicluster from the aggregation with MicroCluster.

GO Term	p-val	counts	definition
GO:0044464	0.00000000	39 / 774	Any constituent part of a cell, the basic structural and functional unit of all organisms. [GOC:jl]...
GO:0044444	0.00000011	18 / 608	Any constituent part of the cytoplasm, all of the contents of a cell excluding the plasma membrane and nucleus, but including other subcellular structures. [GOC:jl]...
GO:0044424	0.00000350	18 / 578	Any constituent part of the living contents of a cell; the matter contained within (but not including) the plasma membrane, usually taken to exclude large vacuoles and masses of secretory or ingested material. In eukaryotes it includes the nucleus and cytoplasm. [GOC:jl]...
GO:0098593	0.00010607	15 / 492	A cup shaped specialization of the cytoskeleton that forms a thin layer located just below the apical mass of mature mucin secretory granules in the cytoplasm of goblet cells of the intestinal epithelium. It consists of an orderly network of intermediate filaments and microtubules. Microtubules are arranged vertically, like barrel staves, along the inner aspect of the theta. Intermediate filaments form two networks: an inner, basketlike network and an outer series of circumferential bundles resembling the hoops of a barrel. [PMID:6541604]...

enumerated biclusters for several values of ϵ, and for $\epsilon = 1.25$ we have 8676.

Figure 48 shows the dendrogram of the aggregation with single linkage for the *FOOD* dataset, when $\epsilon = 1.25$. We can see that the cuts between 2 and 7 are quite acceptable. In fact, cutting in two groups seems the best option, but 2 may be considered a small quantity of biclusters. As from 4 to 5 the height is more pronounced, for the comparison it seems acceptable to cut the dendrogram on 4 biclusters.

Table 11 shows the quantity of final biclusters for the aggregation by overlapping, for several rates. When the rate of overlapping was $\leq$ 80%, we found always the same 4 biclusters. As for the next rate the quantity of final biclusters is much bigger, we will use the solution from the rate 80% for the comparison.

Table 12 shows the quantity of biclusters from the aggregation with MicroCluster. We can see that when the deleting operation came first, the procedure was not able to properly aggregate the biclusters. It is important to highlight that this behavior is the opposite of

Table 9 – Enrichment analysis of the first bicluster from the aggregation with triclustering.

GO Term	p-val	counts	definition
GO:0044464	0.00000000	38 / 774	Any constituent part of a cell, the basic structural and functional unit of all organisms. [GOC:jl]...
GO:0044444	0.00000017	18 / 608	Any constituent part of the cytoplasm, all of the contents of a cell excluding the plasma membrane and nucleus, but including other subcellular structures. [GOC:jl]...
GO:0044424	0.00000488	18 / 578	Any constituent part of the living contents of a cell; the matter contained within (but not including) the plasma membrane, usually taken to exclude large vacuoles and masses of secretory or ingested material. In eukaryotes it includes the nucleus and cytoplasm. [GOC:jl]...
GO:0098593	0.00011049	15 / 492	A cup shaped specialization of the cytoskeleton that forms a thin layer located just below the apical mass of mature mucin secretory granules in the cytoplasm of goblet cells of the intestinal epithelium. It consists of an orderly network of intermediate filaments and microtubules. Microtubules are arranged vertically, like barrel staves, along the inner aspect of the theta. Intermediate filaments form two networks: an inner, basketlike network and an outer series of circumferential bundles resembling the hoops of a barrel. [PMID:6541604]...

what happened with the artificial datasets. There, when the deleting operation came first the results were more effective. When the merging operation came first, the aggregation was able to reach 13 to 27 biclusters, depending on the γ parameter. As on the artificial datasets the best parameters were $\eta = \gamma = 0.15$, for the comparison we will use this parameterization with the merging operation occurring first.

For the triclustering algorithm we set $k = 4$, using insider information from the aggregation by overlapping.

6.6.1 Comparison of the coverage

Table 13 shows the pairwise comparison of coverage of the chosen solutions and the enumerated solution from RIn-Close. We can see that the triclustering algorithm produces the most distinct solution when compared with the enumerated solution obtained with RIn-Close. The difference in coverage of the solutions was $\approx 61.33\%$. The solutions from the aggregation by overlapping and with single linkage were not so close as on the artificial

Table 10 – Quantity of biclusters enumerated with the RIn-Close algorithm on the *FOOD* dataset.

ϵ	Qtd
0	29
0.25	390
0.5	1752
0.75	2946
1	6603
1.25	8676
1.5	13915
1.75	15767
2	23906

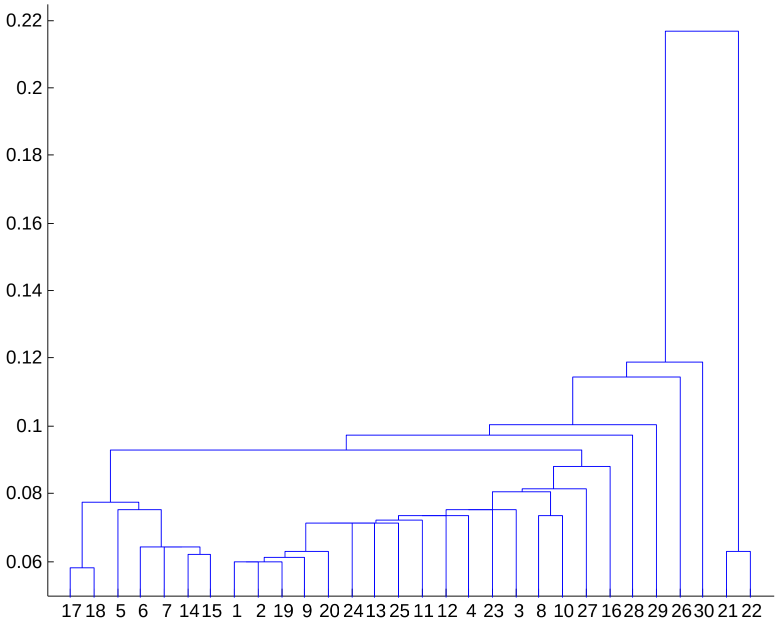


Figure 48 – Dendrogram for the aggregation with single linkage when $\epsilon = 1.25$, on the *Food* dataset.

datasets, showing a difference in coverage of $\approx 12.50\%$.

At the end, the closest solution to the RIn-Close results was the aggregation by overlapping filtered, with a difference in coverage of 9.1%. If we consider that this solution reduced the quantity of biclusters from 8676 to 4 biclusters, the difference in coverage of only 9.1% seems very promising.

Table 11 – Quantity of biclusters for the aggregation by overlapping, for several rates.

Rate	Qtd
60 %	4
65 %	4
70 %	4
75 %	4
80 %	4
85 %	40
90 %	65
95 %	368

Table 12 – Quantity of biclusters for the aggregation with MicroCluster.

Order	η	γ	Qtd
DM	0.15	0.15	809
DM	0.15	0.1	810
DM	0.15	0.05	818
DM	0.1	0.15	545
DM	0.1	0.1	550
DM	0.1	0.05	555
DM	0.05	0.15	553
DM	0.05	0.1	557
DM	0.05	0.05	564
MD	0.15	0.15	13
MD	0.15	0.1	17
MD	0.15	0.05	27
MD	0.1	0.15	14
MD	0.1	0.1	17
MD	0.1	0.05	27
MD	0.05	0.15	14
MD	0.05	0.1	17
MD	0.05	0.05	27

6.7 Further analysis

The experiments presented in this chapter indicate that the aggregation can improve the quality while removing redundancy caused by the noise on enumerative algorithms. However, some aspects of the analysis drew our attention while performing the experiments. Although these details are not part of any experiment, we draw some conclusions of practical interest from this analysis.

The first aspect is associated with the following question: how does the quantity of

Table 13 – Pairwise comparison of difference in coverage among the solutions of aggregation and RIn-Close, on *FOOD* dataset.

	Single Linkage	MicroCluster	Triclustering	RIn-Close
By Ov.	12.50%	35.50%	70.31%	9.1%
Single Linkage	-	46.60%	81.51%	20.17%
MicroCluster	-	-	45.73%	27.38%
Triclustering	-	-	-	61.33%

biclusters increase on the aggregation, while they explode on the enumeration? We decided to verify this behavior on the real datasets.

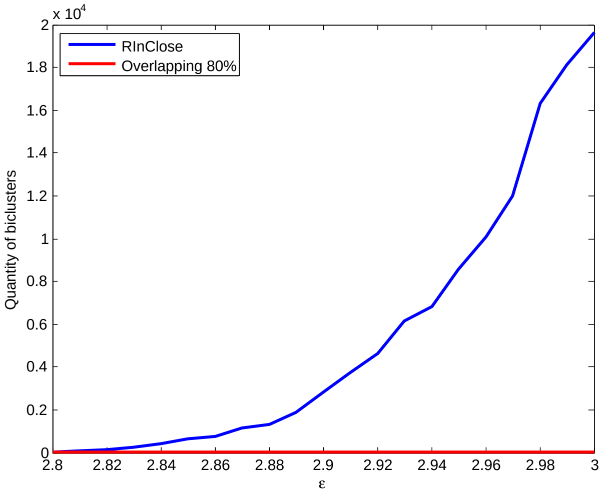


Figure 49 – Comparison of quantity of biclusters as a function of ϵ, of the aggregation by overlapping with rate = 80% and of RIn-Close, on the *Food* dataset.

Figure 49 shows the comparison of the quantity of biclusters, varying the ϵ parameter on the *GDS2587* dataset. The comparison is just on the RIn-Close results versus the aggregation by overlapping with rate = 80%. We can see that while the quantity of biclusters found by RIn-Close increases exponentially, the aggregation seems to produce a stable quantity of biclusters, that we expect to be close to the true value.

Figure 50 shows the same analysis, but for the *FOOD* dataset. We decided to show several rates of overlapping now, and we can see that the behavior of the aggregation is the same, independently of the rate used. All rates of aggregation showed a stable quantity of biclusters that do not exhibit an apparent increase.

Another aspect that drew our attention was: is the aggregation by overlapping robust to the values of ϵ? Obviously we expected that, independently of ϵ, the bigger the rate, the

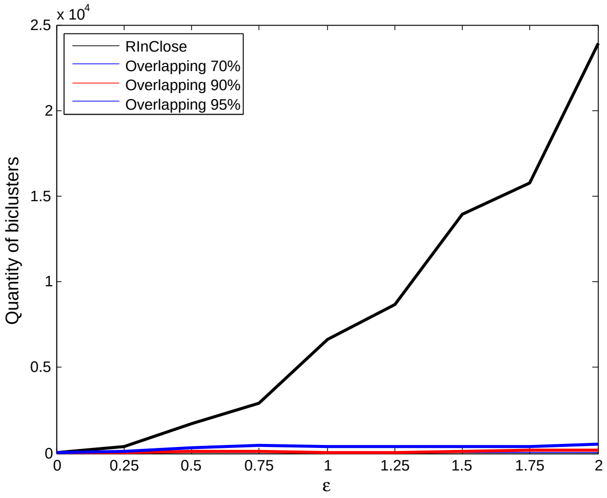


Figure 50 – Comparison of quantity of biclusters as a function of ϵ, of the aggregation by overlapping with several rates and of RIn-Close, on the *Food* dataset.

more biclusters we would have, as we are making it more difficult to aggregate when we expect 95% of overlapping, for example. But does ϵ change that behavior? We run for more values of ϵ on the *FOOD* dataset, and the obtained results are presented in Figure 51.

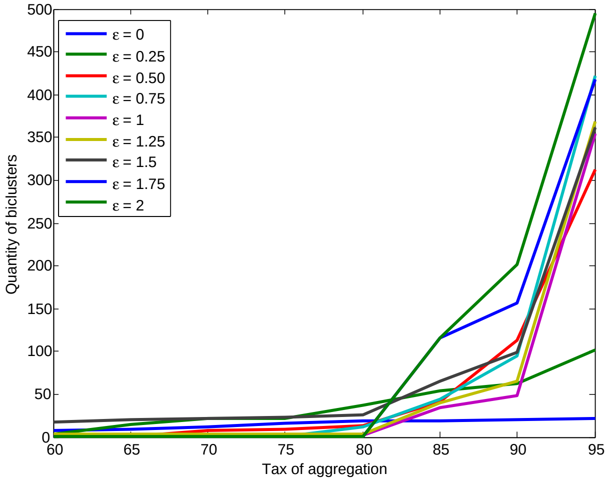


Figure 51 – Quantity of biclusters as a function of the rate of aggregation by overlapping, on the *Food* dataset. Each curve is parameterized by the value of ϵ.

Figure 51 shows the quantity of bicluster by the rate of aggregation, for several values of ϵ. We can see that, when $\epsilon = 0$, the quantity of biclusters remains stable, while when $\epsilon \geq 0.5$

the quantity of biclusters increases exponentially when we increase the rate of aggregation. It shows that the rate of overlapping and the value of ϵ are somehow related to the final quantity of biclusters, when the rate is greater than 80%. If ϵ assumes high values and the rate is equal to or greater then 80%, the aggregation may end up with too much biclusters. But even with high values of ϵ, if the rate of overlapping is low (between 60% and 80%), we are able to get fewer biclusters.

Part IV

Final Considerations

7 Conclusions and future work

In this chapter, we present our concluding remarks. We review the problem of bicluster aggregation and the motivation for this work and also review the settings of the experiments. After that we highlight the main conclusions that can be derived from the results of the experiments. Finally, we indicate potential next steps of the research.

7.1 Concluding remarks

Hartigan (1972) proposed one of the earliest biclustering algorithms, the block clustering. From there, the area drew the attention of many communities, becoming an important non-supervised analysis. The term “biclustering” was first used by Cheng & Church (2000) in their seminal work on gene expression data analysis, one of the major applications of biclustering techniques.

Since finding all biclusters in a data set is an NP-hard problem –it is equivalent to enumerating all bicliques in a bipartite graph –, the majority of the biclustering algorithms are heuristics, that usually miss important biclusters (VERONEZE *et al.*, 2014). Even with this drawback, the usefulness of this task is unquestionable, given the amount of applications and proposed heuristics. In the literature, we can find applications in gene expression data analysis, recommendation systems and marketing (MADEIRA; OLIVEIRA, 2004).

With the development of several algorithms, it was noticed that the inherent noise of the data fragments the original biclusters into many with high overlapping. This fragmentation heavily influenced the outcome of the recent enumerative algorithms, leading to a large quantity of highly overlapped biclusters. Also, the high overlapping of the enumerated biclusters leads to a redundant result, which also increases the complexity of the analysis.

At first, the problem of aggregation is very similar to bicluster ensemble. Ensemble is a common practice in supervised learning (and is gaining attention in unsupervised learning), where we combine the outcome of several results into a single one that is more robust to noise interferences. Ideally, we obtain different results from different algorithms, or by distinct views of the dataset, which increases the diversity of the results. This diversity is directly related to the robustness of the final combination of the distinct results. Bicluster ensemble is very similar. First we obtain several different results, then we combine them to get a single one. But the aggregation poses different challenges. The differences between bicluster ensemble and bicluster aggregation are:

Source of results In the ensemble setup we need diversity of the input results. In aggregation we can work just with results that show a high degree of overlapping among their components, which are generally of low diversity. This is common in bicluster enumeration.

Importance of a single bicluster If an area of the dataset is covered just by one bicluster, the ensemble should consider this bicluster insignificant, as it was not encountered by any other solutions. The aggregation does not eliminate biclusters.

We focused on the aggregation of biclusters from enumeration, and we proposed two approaches. The first one transforms each bicluster into a binary vector. After that we apply the single linkage hierarchical clustering using the Hamming distance. After cutting the dendrogram, we aggregate the biclusters of the same partitions by uniting the rows and columns of each one. This approach has the drawback of having one parameter that is the number of final biclusters. However, the dendrogram can be helpful in this task and there already are several methods for properly defining the cut location.

The other approach is based on the overlapping between two biclusters. The method can be summarized as: while having two biclusters with an overlapping area larger than a pre-defined threshold, aggregate them. Again, the aggregation is the union of rows and columns of the involved biclusters.

The aggregation not only severely reduced the final quantity of biclusters, but ended up impacting a little the *Precision* of the final biclusters. The way that we perform the aggregation explains this behavior. By just uniting the rows and columns of the involved biclusters, we may end up including intruder rows or columns that should not be included. This requires a step of outlier removal. To this end, we provided a method to remove possible outlier elements from the biclusters of the final result. This method showed to be robust, improving the quality of the solution.

We compared the performance of our proposals against the most similar proposal in the literature, which is the last step of the MicroCluster algorithm. We also included in the comparison an algorithm of bicluster ensemble, which is the triclustering algorithm. We executed 5 experiments to verify distinct hypotheses.

The first experiment aimed at viewing the effects of the noise fragmenting the original biclusters. We only used artificial datasets. Using the RIn-Close algorithm, we could verify the fragmentation of the original biclusters into many with high overlapping. As the variance of the noise increases, we could see that: the quantity of biclusters increases reaching a high value and then goes down to zero; the *Precision* starts low and also increases; and when we have enough noise, the *Recall* decreases.

On the second experiment we tested several approaches of aggregation, including our proposals. We were able to reach the true number of biclusters without decreasing the *Recall*, but decreasing a little the *Precision*. Despite that, our proposals had a nice performance, getting the best results when compared with the MicroCluster and the triclustering algorithm on the datasets *art1* and *art2*. The *art3* dataset showed to be the most challenging one. Our two proposals had very similar results, and when we run the Wilcoxon Rank Sum Test, they did not show significant difference. On this experiment we could also see that the aggregation in fact is able to get much less biclusters with a comparable or even better quality results when compared with the enumeration. The main challenge is to parameterize the algorithms. The deleting step of the MicroCluster algorithm acts as an outlier removal, and our proposals did not exclude bad objects and attributes from the biclusters. Our proposals could benefit from a step of outlier removal, which is the subject of experiment 3.

On the third experiment, we added the step of outlier removal to the aggregation with single linkage and by overlapping. These proposals achieved a high performance score on the *art1* dataset, and were able to increase the *Precision* on the *art2* dataset. On the *art3* dataset, the outlier removal was not able to increase the *Precision* and in fact decreased a little the *Recall*. But we could conclude that this step is very important to avoid biclusters with objects and attributes that should not be part of any bicluster.

The purpose of the fourth experiment was to verify if the aggregation could get enriched biclusters of a gene expression dataset. For different values of ϵ on the RIn-Close algorithm, we could see that the different methods of aggregation reached very similar results. The main challenge of the aggregation with single linkage is to decide where to cut the dendrogram, but as we could see, on this dataset this task was very easy. Likewise, it was easy to identify a good rate of overlapping, as the results of several rates were identical. Except for the triclustering, all aggregations returned only enriched biclusters.

And finally, we applied the aggregation methods to the *FOOD* dataset and analyzed how the aggregation changed the covered area when compared to the enumeration. Triclustering led to the most different result, and the aggregation by overlapping covered an area very similar to the area covered by the enumeration.

We could see that while the quantity of biclusters increases exponentially on the enumeration, the aggregation can keep a stable quantity of biclusters, independently of the value of ϵ. We could also see that the value of ϵ changes the behavior of the aggregation by overlapping when the rate is high. For values greater than 80%, some values of ϵ led to a high quantity of biclusters. It indicates that the rate of overlapping must be reasonable, given that high values cannot guide to an effective aggregation.

We can conclude that the aggregation is suitable and can be indicated when enumerating all biclusters from a dataset. The aggregation will not only significantly reduce the quantity of biclusters, but tends to improve the quality of the final result. A post-processing step for outlier removal brings additional robustness to the methodology.

7.2 Future Work

As a further step of the research, we need to compare the time / memory complexity of the proposals, and explore recommendations for the parameterization (cut on the dendrogram and rate of overlapping). The chaining effect on single linkage hierarchical clustering using Hamming distances should be more explored to verify its impacts in the aggregation. Another further step is to test our proposals in biclustering heuristics results.

We can also adapt our proposals to work on an ensemble configuration, and extend this work to deal with time series biclusters, which require contiguous attributes.

Bibliography

AGGARWAL, G.; GUPTA, N. BiETopti-BiClustering Ensemble Using Optimization Techniques. In: PERNER, P. (Ed.). *Advances in Data Mining. Applications and Theoretical Aspects*. Springer, 2013, (Lecture Notes in Computer Science, v. 7987). p. 181–192. ISBN 978-3-642-39735-6. Disponível em: <http://dx.doi.org/10.1007/978-3-642-39736-3_14>. Citado na página 17.

BANERJEE, A.; KRUMPELMAN, C.; GHOSH, J.; BASU, S.; MOONEY, R. J. Model-based overlapping clustering. In: *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*. New York, NY, USA: ACM, 2005. (KDD '05), p. 532–537. ISBN 1-59593-135-X. Disponível em: <http://doi.acm.org/10.1145/1081870.1081932>. Citado na página 21.

CHENG, Y.; CHURCH, G. M. Biclustering of expression data. *Proceedings of International Conference on Intelligent Systems for Molecular Biology; ISMB. International Conference on Intelligent Systems for Molecular Biology*, Department of Genetics, Harvard Medical School, Boston, MA 02115, USA. yizong.cheng@uc.edu, v. 8, p. 93–103, 2000. ISSN 1553-0833. Disponível em: <http://view.ncbi.nlm.nih.gov/pubmed/10977070>. Citado 5 vezes nas páginas 9, 11, 24, 25, and 87.

EVERITT, B. S.; LANDAU, S.; LEESE, M.; STAHL, D. An introduction to classification and clustering. In: _____. *Cluster Analysis*. John Wiley and Sons, Ltd, 2011. p. 321–330. ISBN 9780470977811. Disponível em: <http://dx.doi.org/10.1002/9780470977811.index>. Citado na página 7.

FRED, A. L. N.; JAIN, A. K. Combining multiple clusterings using evidence accumulation. *IEEE transactions on pattern analysis and machine intelligence*, v. 27, n. 6, p. 835–50, jun. 2005. ISSN 0162-8828. Disponível em: <http://www.ncbi.nlm.nih.gov/pubmed/15943417>. Citado na página 16.

GAO, T.; AKOGLU, L. Fast information-theoretic agglomerative co-clustering. In: WANG, H.; SHARAF, M. A. (Ed.). *Databases Theory and Applications*. Springer International Publishing, 2014, (Lecture Notes in Computer Science, v. 8506). p. 147–159. ISBN 978-3-319-08607-1. Disponível em: <http://dx.doi.org/10.1007/978-3-319-08608-8_13>. Citado 2 vezes nas páginas 19 and 41.

GHOSH, J.; ACHARYA, A. Cluster ensembles. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, v. 1, n. 4, p. 305–315, jul. 2011. ISSN 19424787. Disponível em: <http://doi.wiley.com/10.1002/widm.32>. Citado na página 15.

GULLO, F.; TALUKDER, A. K. M. K.; LUKE, S.; DOMENICONI, C.; TAGARELLI, A. Multiobjective optimization of co-clustering ensembles. In: SOULE, T.; MOORE, J. H. (Ed.). *GECCO (Companion)*. ACM, 2012. p. 1495–1496. ISBN 978-1-4503-1178-6. Disponível em: <http://dblp.uni-trier.de/db/conf/gecco/gecco2012c.html#GulloTLDT12>. Citado na página 16.

HANCZAR, B.; NADIF, M. Improving the Biological Relevance of Biclustering for Microarray Data in Using Ensemble Methods. *2011 22nd International Workshop on Database and Expert Systems Applications*, Ieee, p. 413–417, ago. 2011. Disponível em: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6059852>. Citado na página 16.

HANCZAR, B.; NADIF, M. Using the bagging approach for biclustering of gene expression data. *Neurocomputing*, Elsevier, v. 74, n. 10, p. 1595–1605, maio 2011. ISSN 09252312. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/S0925231211001196>. Citado 3 vezes nas páginas 16, 24, and 25.

HANCZAR, B.; NADIF, M. Ensemble methods for biclustering tasks. *Pattern Recognition*, v. 45, n. 11, p. 3938 – 3949, 2012. ISSN 0031-3203. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0031320312001677>. Citado na página 17.

HARTIGAN, J. A. Direct Clustering of a Data Matrix. *Journal of the American Statistical Association*, American Statistical Association, v. 67, n. 337, p. 123–129, 1972. ISSN 01621459. Disponível em: <http://dx.doi.org/10.2307/2284710>. Citado 4 vezes nas páginas 9, 24, 25, and 87.

HORTA, D.; CAMPELLO, R. J. G. B. Similarity measures for comparing biclusterings. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, v. 11, n. 5, p. 942–954, Sept 2014. ISSN 1545-5963. Citado 2 vezes nas páginas 21 and 23.

HUANG, Q.; CHEN, X.; HUANG, J. Z.; FENG, S.; FAN, J. Scalable Ensemble Information-Theoretic Co-clustering for Massive Data. In: *Proceedings of the International MultiConference of Engineers nad Computer Scientists 2012 Vol I*. [S.l.: s.n.], 2012. I. ISBN 9789881925114. Citado na página 16.

JAIN, A. K.; MURTY, M. N.; FLYNN, P. J. Data clustering: A review. *ACM Comput. Surv.*, ACM, New York, NY, USA, v. 31, n. 3, p. 264–323, set. 1999. ISSN 0360-0300. Disponível em: <http://doi.acm.org/10.1145/331499.331504>. Citado 3 vezes nas páginas 1, 7, and 8.

JIONG, Y.; WANG, H.; WANG, W.; YU, P. Enhanced biclustering on expression data. In: *Bioinformatics and Bioengineering, 2003. Proceedings. Third IEEE Symposium on*. [S.l.: s.n.], 2003. p. 321–327. Citado 3 vezes nas páginas 12, 24, and 25.

LANGFELDER, P.; 0002, B. Z.; HORVATH, S. Defining clusters from a hierarchical cluster tree: the dynamic tree cut package for r. *Bioinformatics*, v. 24, n. 5, p. 719–720, 2008. Disponível em: <http://dblp.uni-trier.de/db/journals/bioinformatics/bioinformatics24.html#LangfelderZH08>. Citado na página 30.

LAZZERONI, L.; OWEN, A. Plaid models for gene expression data. *Statistica Sinica*, v. 12, p. 61–86, 2000. Citado 3 vezes nas páginas 24, 25, and 37.

LIMA, C. A. M. *Comitê de Máquinas : Uma Abordagem Unificada Empregando Máquinas de Vetores-Suporte, Tese de Doutorado, FEEC / Unicamp*. 2004. Citado na página 15.

LIU, J.; WANG, J.; WANG, W. Biclustering in gene expression data by tendency. In: *CSB*. IEEE Computer Society, 2004. p. 182–193. ISBN 0-7695-2194-0. Disponível em: <http://dblp.uni-trier.de/db/conf/csb/csb2004.html#LiuYW04>. Citado 3 vezes nas páginas 17, 19, and 40.

MADEIRA, S. C.; OLIVEIRA, A. L. Biclustering algorithms for biological data analysis: A survey. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, IEEE Computer Society Press, Los Alamitos, CA, USA, v. 1, n. 1, p. 24–45, jan. 2004. ISSN 1545-5963. Disponível em: <http://dx.doi.org/10.1109/TCBB.2004.2>. Citado 4 vezes nas páginas 9, 11, 12, and 87.

MAKINO, K.; UNO, T. New algorithms for enumerating all maximal cliques. In: HAGERUP, T.; KATAJAINEN, J. (Ed.). *SWAT*. Springer, 2004. (Lecture Notes in Computer Science, v. 3111), p. 260–272. ISBN 3-540-22339-8. Disponível em: <http://dblp.uni-trier.de/db/conf/swat/swat2004.html#MakinoU04>. Citado na página 12.

MENESTRINA, D.; WHANG, S. E.; GARCIA-MOLINA, H. *Evaluating Entity Resolution Results (Extended version)*. [S.l.], 2009. Disponível em: <http://ilpubs.stanford.edu:8090/930/>. Citado na página 22.

PANDEY, G.; ATLURI, G.; STEINBACH, M.; MYERS, C. L.; KUMAR, V. An association analysis approach to biclustering. In: *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. [s.n.], 2009. (KDD '09), p. 677–686. ISBN 978-1-60558-495-9. Disponível em: <http://doi.acm.org/10.1145/1557019.1557095>. Citado na página 12.

PEI, J.; ZHANG, X.; CHO, M.; WANG, H.; YU, P. Maple: a fast algorithm for maximal pattern-based clustering. In: *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*. [S.l.: s.n.], 2003. p. 259–266. Citado na página 12.

PERRONE, M. P.; COOPER, L. N. When networks disagree: Ensemble methods for hybrid neural networks. In: MAMMONE, R. J. (Ed.). *Artificial Neural Networks for Speech and Vision*. [S.l.]: Chapman and Hall, 1993. p. 126–142. Citado na página 15.

PIO, G.; CECI, M.; D'ELIA, D.; LOGLISCI, C.; MALERBA, D. A Novel Biclustering Algorithm for the Discovery of Meaningful Biological Correlations between microRNAs and their Target Genes. *BMC Bioinformatics*, v. 14, n. Suppl 7, p. S8, 2013. ISSN 1471-2105. Disponível em: <http://www.biomedcentral.com/1471-2105/14/S7/S8>. Citado na página 17.

PRELIć, A.; BLEULER, S.; ZIMMERMANN, P.; WILLE, A.; BüHLMANN, P.; GRUISSEM, W.; HENNIG, L.; THIELE, L.; ZITZLER, E. A systematic comparison and evaluation of biclustering methods for gene expression data. *Bioinformatics*, Oxford University Press, Oxford, UK, v. 22, n. 9, p. 1122–1129, maio 2006. ISSN 1367-4803. Disponível em: <http://dx.doi.org/10.1093/bioinformatics/btl060>. Citado na página 38.

RAND, W. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, v. 66, n. 336, p. 846–850, 1971. Citado na página 30.

RIGSBERGEN, C. J. van. *Information Retrieval*. Englewood Cliffs: Prentice Hall, 1979. 112–123 p. Citado na página 21.

SALTON, G. Evaluation parameters. *The SMART Retrieval System, Experiments in Automatic Document Processing*, p. 55–112, 1971. Citado na página 21.

SHARKEY, A. J. C. On combining artificial neural nets. *Connection Science*, v. 8, p. 299–313, 1996. Citado na página 15.

SIMON, A. In-close, a fast algorithm for computing formal concepts. In: *the Seventeenth International Conference on Conceptual Structures*. [S.l.: s.n.], 2009. Citado na página 12.

SOKAL, R. R.; ROHLF, F. J. The comparison of dendrograms by objective methods. *Taxon*, v. 11, n. 2, p. 33–40, 1962. Citado na página 30.

STREHL, A.; GHOSH, J. Cluster Ensembles – A Knowledge Reuse Framework for Combining Multiple Partitions. *Journal of Machine Learning Research*, v. 3, p. 583–617, 2002. Citado 2 vezes nas páginas 15 and 16.

STREHL, A.; STREHL, E.; GHOSH, J. A scalable approach to balanced, high-dimensional clustering of market-baskets. In: *In Proceedings of HiPC*. [S.l.]: Springer, 2000. p. 525–536. Citado na página 15.

TANAY, A.; SHARAN, R.; SHAMIR, R. Biclustering algorithms: A survey. In: *In Handbook of Computational Molecular Biology Edited by: Chapman & Hall/CRC Computer and Information Science Series*. [S.l.: s.n.], 2005. Citado na página 12.

VEGA-PONS, S.; RUIZ-SHULCLOPER, J. A survey of clustering ensemble algorithms. *International Journal of Pattern Recognition and Artificial Intelligence*, v. 25, n. 03, p. 337–372, maio 2011. ISSN 0218-0014. Disponível em: <http://dx.doi.org/10.1142/s0218001411008683>. Citado na página 16.

VERONEZE, R.; BANERJEE, A.; ZUBEN, F. J. V. Enumerating all maximal biclusters in real-valued datasets. *arXiv:1403.3562v3*, abs/1403.3562, 2014. Citado 6 vezes nas páginas 1, 12, 13, 38, 77, and 87.

WANG, H.; SHAN, H.; BANERJEE, A. Bayesian cluster ensembles. *Statistical Analysis and Data Mining*, v. 4, n. 1, p. 54–70, fev. 2011. ISSN 19321864. Disponível em: <http://doi.wiley.com/10.1002/sam.10098>. Citado na página 15.

WANG, H.; WANG, W.; YANG, J.; YU, P. S. Clustering by pattern similarity in large data sets. In: *Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data*. New York, NY, USA: ACM, 2002. (SIGMOD '02), p. 394–405. ISBN 1-58113-497-5. Disponível em: <http://doi.acm.org/10.1145/564691.564737>. Citado na página 12.

ZHAO, L.; ZAKI, M. J. Microcluster: Efficient deterministic biclustering of microarray data. *IEEE Intelligent Systems*, v. 20, n. 6, p. 40–49, 2005. Disponível em: <http://doi.ieeecomputersociety.org/10.1109/MIS.2005.112>. Citado 6 vezes nas páginas 1, 12, 17, 24, 25, and 40.