UNIVERZITET U NOVOM SADU
FAKULTET TEHNIČKIH NAUKA
NOVI SAD

Sanja Brdar

ALGORITMI INTEGRATIVNOG KLASTEROVANJA PODATAKA PRIMENOM NENEGATIVNE FAKTORIZACIJE MATRICE

doktorska disertacija

Novi Sad, 2016

УНИВЕРЗИТЕТ У НОВОМ САДУ ● **ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА**
21000 НОВИ САД, Трг Доситеја Обрадовића 6

КЉУЧНА ДОКУМЕНТАЦИЈСКА ИНФОРМАЦИЈА

Редни број, **РБР**:		
Идентификациони број, **ИБР**:		
Тип документације, **ТД**:		Монографска документација
Тип записа, **ТЗ**:		Текстуални штампани материјал
Врста рада, **ВР**:		Докторска дисертација
Аутор, **АУ**:		Сања Брдар
Ментор, **МН**:		Проф. др Дејан Вукобратовић
Наслов рада, **НР**:		Алгоритми интегративног кластеровања података применом ненегативне факторизације матрице
Језик публикације, **ЈП**:		Енглески
Језик извода, **ЈИ**:		енглески / српски
Земља публиковања, **ЗП**:		Србија
Уже географско подручје, **УГП**:		Аутономна Покрајина Војводина
Година, **ГО**:		2016.
Издавач, **ИЗ**:		Ауторски репринт
Место и адреса, **МА**:		Факултет техничких наука, Трг Доситеја Обрадовића 6, Нови Сад
Физички опис рада, **ФО**: (поглавља/страна/ цитата/табела/слика/графика/прилога)		8 поглавља/ 115 страна/ 32 слике/18 табела/130 референци/ 3 прилога.
Научна област, **НО**:		Електротехничко и рачунарско инжењерство
Научна дисциплина, **НД**:		Машинско учење
Предметна одредница/Кqучне речи, **ПО**:		кластеровање података, интегративно кластеровање, ненегативна факторизација матрице, фузија података, биоинформатика
УДК		Монографска документација
Чува се, **ЧУ**:		Библиотека Факултета техничких наука, Универзитет у Новом Саду
Важна напомена, **ВН**:		
Извод, **ИЗ**:		Предмет истраживања докторске дисертације су алгоритми кластеровања, односно груписања података, и могућности њиховог унапређења интегративним приступом у циљу повећања поузданости, робустности на присуство шума и екстремних вредности у подацима, омогућавања фузије података. У дисертацији су предложене методе засноване на ненегативној факторизацији матрице. Методе су успешно имплементиране и детаљно анализиране на разноврсним подацима са *UCI* репозиторијума и синтетичким подацима које се типично користе за евалуацију нових алгоритама и поређење са већ постојећим методама. Већи део дисертације посвећен је примени у домену биоинформатике која обилује хетерогеним подацима и бројним изазовним задацима. Евалуација је извршена на подацима из домена функционалне геномике, геномике рака и метагеномике.
Датум прихватања теме, **ДП**:		12.02.2015.
Датум одбране, **ДО**:		
Чланови комисије, **КО**:	Председник:	Проф. др Вељко Милутиновић
	Члан:	Проф. др Војин Шенк
	Члан:	Проф. др Владимир Црнојевић
	Члан:	Проф. др Срђан Шкрбић
	Члан:	Проф. др Татјана Лончар-Турукало
	Члан, ментор:	Проф. др Дејан Вукобратовић

Потпис ментора

UNIVERSITY OF NOVI SAD ● **FACULTY OF TECHNICAL SCIENCES**
21000 NOVI SAD, Trg Dositeja Obradovića 6

KEY WORDS DOCUMENTATION

Образац **Q2.HA.06-05**- Издање 1

Accession number, **ANO**:			
Identification number, **INO**:			
Document type, **DT**:		Monograph documentation	
Type of record, **TR**:		Textual printed material	
Contents code, **CC**:		PhD thesis	
Author, **AU**:		Sanja Brdar	
Mentor, **MN**:		Dr Dejan Vukobratović, associate professor	
Title, **TI**:		Non-negative matrix factorization for integrative clustering	
Language of text, **LT**:		English	
Language of abstract, **LA**:		English/Serbian	
Country of publication, **CP**:		Serbia	
Locality of publication, **LP**:		Autonomous Province of Vojvodina	
Publication year, **PY**:		2016	
Publisher, **PB**:		Author's reprint	
Publication place, **PP**:		Faculty of Technical Sciences, Trg Dositeja Obradovića 6, Novi Sad	
Physical description, **PD**: (chapters/pages/ref./tables/pictures/graphs/appendixes)		8 chapters/115 pages/130 references/18 tables/32 figures/3 appendixes	
Scientific field, **SF**:		Electrical and Computer Engeneering	
Scientific discipline, **SD**:		Machine learning	
Subject/Key words, **S/KW**:		clustering, ensemble clustering, non-negative matrix factorization, data fusion, bioinformatics	
UC			
Holding data, **HD**:		Library of the Faculty of Technical Sciences, University of Novi Sad	
Note, **N**:			
Abstract, **AB**:		Integrative approaches are motivated by the desired improvement of robustness, stability and accuracy. Clustering, the prevailing technique for preliminary and exploratory analysis of experimental data, may benefit from integration across multiple partitions. In this thesis we have proposed integration methods based on non-negative matrix factorization that can fuse clusterings stemming from different data sets, different data preprocessing steps or different sub-samples of objects or features. Proposed methods are evaluated from several points of view on typical machine learning data sets, synthetics data, and above all, on data coming form bioinformatics realm, which rise is fuelled by technological revolutions in molecular biology. For a vast amounts of 'omics' data that are nowadays available sophisticated computational methods are necessary. We evaluated methods on problem from cancer genomics, functional genomics and metagenomics.	
Accepted by the Scientific Board on, **ASB**:		12.02.2015.	
Defended on, **DE**:			
Defended Board, **DB**:	President:	Veljko Milutinović, PhD, full professor	
	Member:	Vojin Šenk, PhD, full professor	
	Member:	Vladimir Crnojević, PhD, full professor	Menthor's sign
	Member:	Srđan Škrbić, PhD, associate professor	
	Member:	Tatjana Lončar-Turukalo, PhD, assistant professor	
	Member, Mentor:	Dejan Vukobratović, PhD associate professor	

UNIVERSITY OF NOVI SAD ● **FACULTY OF TECHNICAL SCIENCES**
21000 NOVI SAD, Trg Dositeja Obradovića 6

KEY WORDS DOCUMENTATION

Obrazac **Q2.HA.06-05**- Izdanje 1

Nonnegative matrix factorization for integrative clustering

Sanja Brdar

Faculty of Technical Sciences

University of Novi Sad

A dissertation submitted in partial fulfilment of the requirements for the degree of

Doctor of Philosophy (PhD)

Novi Sad, 2016

Abstract

In machine learning integrative approaches are motivated by the desired improvement of robustness, stability and accuracy. Clustering, the prevailing technique for preliminary and exploratory analysis of experimental data, may benefit from integration across multiple partitions. Different partitions can be inferred from different initialization, algorithms, parameters, features subsamples, items subsamples, similarity/distance functions or heterogeneous data sources. To overcome users' dilemma of selecting data partition among many possible, we developed a technique that infers separate clusters from diverse inputs and then fuses them by means of non-negative matrix factorization (NMF).

The proposed fusion technique is evaluated within the scope of bioinformatics and obtained results are of interest for functional genomics, cancer genomics and metagenomics. In functional genomics NMF based integrative clustering contributes to an increase of the quality of clusters with respect to enrichment of their associated gene functions. On high-dimensional cancer genomics microarrays, experiments unveiled how large are uncertainties in the recoveries of cancer types or subtypes by clustering. The best outcome of the integration was to be at the level of the best partition in the ensemble, while the worst was at average level. Thus integration helps in avoiding the risk of choosing a poor individual partition. In metagenomics we examined the stability of clusters in human microbiome samples in a context of various beta diversity measures and evinced that microbial diversity assessment may also benefit from ensemble clustering. We explore the effects of 24 different diversity metrics on clustering outcomes and their impact on the accuracy of the clustering of microbial samples. To overcome obscure results coming from individual clusterings that rely on distinct beta diversity metrics we integrated results of individual clustering by NMF ensemble approach. Obtained results on human microbiome samples imply that ensemble approach produces stable results in reconstructing clusters corresponding to the different host and body habitat.

The landscape of integrative clustering algorithms is further explored by comprehensive comparison of the partitions generated by NMF and 5 alternative ensemble algorithms on the typical machine learning and synthetic data sets. NMF compares favorably to other approaches on a range of validation criteria. Finally, the research on a graph regularized NMF integrative clustering that allows including additional information is presented, as well as distributed ensemble clustering. Promising results in both open the avenues of possibilities for future research directed towards semi-supervised and large scale clustering.

Acknowledgements

Firstly, as my PhD journey started at the University of Ljubljana, I would like to express my sincere gratitude to professor Blaž Zupan for accepting me into his Bioinformatics Laboratory at Faculty of Computer and Information Science. One year in such welcoming and great place immensely enriched my life in many ways - friendships, knowledge, experience.

Next, I would like to thank my supervisor, professor Vladimir Crnojević, for continuous support throughout my PhD studies, for giving me freedom to pursue my ideas and above all for his enthusiasm and vision to create high quality research place in Novi Sad where we can do exciting research.

I also wish to thank professor Dejan Vukobratović, for his immense help in the final stages of this PhD and all members of the Chair for Communications and Signal Processing at The Faculty of Technical Sciences for their support.

Furthermore, I thank to my colleagues from BioSense institute: Peđa, Marko, Oskar, Vladan, Olivera, Sanja, Branko and all others for joint work, scientific discussions, good atmosphere, shared lunches...

Finally, I am grateful to my parents, mother Nada and father Stevo for encouraging me to chase my dreams. A special thanks goes to my dear fellowship, that followed me through all ups and downs of this PhD journey: my friend Jelena, sister Jelena and my Zoran for their love, support, critique, understanding, inspiration.

Sanja Brdar,
in Novi Sad, September 2016.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

An abundance of data sources and their increasing volume impose challenges in front of data science community. To analyse and discover underlying structure in such data, novel algorithms are necessary. The algorithms need to provide us with robustness against noise in the data and high dimensional feature-spaces and to enable scalability for large datasets. The most prevailing data exploration technique is clustering that in unsupervised manner, i.e. without use of label information, divides data into groups of similar objects. Clustering simplifies data by representing it with fewer prototypes corresponding to discovered clusters. We can also denote clusters as hidden patterns and clustering then corresponds to search for hidden patterns [1].

To deal with fast growing and heterogeneous data sources clustering algorithms are evolving through new directions: ensemble, semi-supervised and large-scale. Ensemble approaches in clustering leverage different outcomes of clusterings obtained across many potential inputs induced by various parameters, metrics, features, algorithms [2]. Semi-supervised clusterings utilize available labels to constrain the search for clusters [3]. These approaches are especially useful when small subset of the data is labelled. For massive data clusterings where the numbers of objects are large, we need to scale up and speed up algorithms [4] with parallel and distributed approaches. Algorithm itself determines possible ways to parallelize and distribute computations. Solutions may rely on approximate calculations of nearest neighbours [5] or on offloading work to GPU (graphical processing units) [6]. Memory issues that arise when algorithms need to calculate and store complete similarity/distance matrix can be solved by random sub-sampling [7] or low-dimensional embedding [8]. Major part of the thesis presented here covers ensemble approaches in the clustering, but other two directions are also tackled and discussed.

When it comes to clustering of the data, a question whether it is a science or an art is ever present [9]. To qualify it as a science we need measures and scores to evaluate and compare the results of clustering. Measuring the quality of a clustering or quantifying its utility is crucial here, but the problem arises from different validation measures. For validating the results of clustering, one direction proposes considering just data without any external information. However, clustering is not domain independent and it should serve the end goals of application. Therefore, clustering must be placed in the context of a specific problem where it will be applied. Furthermore, large scale benchmarking across diverse sets and different validations can provide general perspectives of the usefulness of algorithms. Although we do not expect that one clustering algorithm will surpass all others in this large scale setting, interesting patterns can emerge and point out on robust algorithms or recognize suitable for specific problems. The same validation issues are present in the integrative clustering, where they additionally serve to judge whether the integration helps or not.

In this thesis real-world applications of integrative clustering are explored, evaluated and discussed. Different problems from bioinformatics were selected to evaluate algorithms as the demand for computational tools in this domain increases with growing accumulation of biological data. Applying clustering on biological data is faced with many challenges that arise from following complications: high level of noise in data, cluster shapes may be irregular or non-convex, objects of interest may naturally belong to more than one category and prior knowledge of the underlying distributions is missing. To overcome these challenges efforts in this thesis are directed towards integrative approaches that rely on combined analyses od data sources, subsets of features, samples, etc.

1.1 Applications in bioinformatics

Current biology produces a wealth of data, especially at molecular level. Over the last decade its growth rate surpassed Moor's laws of doubling size every 18 months with recently estimated [10] doubling size every 7 months. Those diverse high-throughput data are often refereed as 'omics' to collectively characterize different sources - genomics, transcriptomics, proteomics, metabolomics, etc. A plethora of data types and their quantity induced fundamental shift in molecular biology research. A new field, bioinformatics, arose from the ambition to develop powerful tools for the analysis and making sense of genomics data. Development of such tools will have important implications on all life sciences.

The growth of bioinformatics started at the end of Human Genome Project [11] that revealed the complexity hidden in huge number of DNA fragments. It was apparent that produced data could not be handled without appropriate solutions for storing, distributing, analysing and extracting knowledge. With further

advancement of molecular technology that moreover produced information overload, bioinformatics reached its acceleration point and will certainly increase the role it plays in biology, medicine, agriculture and other bio-related fields. Bioinformatics strives to develop computational solutions for omics data [12]. To meet the computational and analytical challenges it relies on data mining, machine learning, graph algorithms, data compression and data/knowledge integration. Machine learning algorithms became indispensable tool in extracting biological knowledge [13] and they are further enhanced to handle data on larger scale [14].

Bioinformatics' cross-disciplinary nature requires understanding of data sources and specificity of molecular domains before developing tools that will serve for data analysis and knowledge discovery. Here we shortly introduce 'Omics' data and 'Omics' domains that are further explored in the applications within the thesis.

1.1.1 'Omics' data representations

'Omics' data are highly heterogeneous and come in diverse forms: sequences, expressions, interactions, pathways, ontologies. Challenge is now shifted from creation toward analysis of such data. Each representation fosters development of specific methods that are necessary for processing, analysis and extracting information. Here we enumerate and briefly explain data used in the experiments.

- **Sequences** are basic form of information in molecular biology and may come from different molecules: DNA, RNA or proteins. They are ordered array of basis - nucleotides (A, C, G, T) in case of DNA, (A, C, G, U) for RNA, or amino acids (21 symbols). Methods typical for the analysis of sequence data include alignments, assembling, comparisons, identification of features, mutations, measuring genetic diversity [15]. Within the thesis protein sequence data were used in the Chapter 6, and RNA tag sequences in the Chapter 7.

- **Data Matrices** are two dimensional arrays with numeric values. Matrices encompass multiple variables and each row or column can be seen as profile of data values. Representative examples of data in a matrix form are gene expressions, where dimensions can correspond to $patents \times genes$, $timepoints \times genes$, $states \times genes$. Another example is biological observation matrix that contains counts of observations on a per-sample basis, where observation can be taxonomic unit. Broad range of methods coming from machine learning, optimization theory, algebra, etc. are offered for the analysis of data represented as matrices [16], [17]. In the thesis matrix data were used in the Chapter 4, 5, 6 and 7.

- **Networks** are mathematically represented as the graph structures where objects of interest are modelled as nodes, and their connections as edges. This structure allows to quantify associations between nodes, analyse local and overall composition of the network. Typical examples of biological networks are protein-protein interactions (PPI), where proteins are nodes and their interactions are edges. Beside this raw network data, networks can be sparse variants of initial data like gene co-expression networks that are derived form expressions data matrices. PPI and co-expression networks are intensely analysed in bioinformatics. Graph based algorithms can characterize local interconnectivity, compare given network topologies, detect given sub-graphs, decompose network into clusters, etc. [18]. Chapter 6 includes protein-protein interaction data. In Chapter 5 and 6 matrix data were processed to obtain gene networks.

- **Ontologies** are knowledge representation schemes. They define formal framework and controlled vocabulary for organising existing knowledge. Ontologies help to search, annotate and integrate data. Biomedical domains extensively use ontologies to describe objects and the associations or relationships between them [19]. Well known example of ontology in bioinformatics is Gene Ontology GO [20] that captures information about genes/proteins and their functions. The results presented in the chapters 5 and 6 were evaluated against existing knowledge in Gene Ontology.

Figure 1.1 illustrates previously described data representations of omics data. Six genes of Yeast (ALG5, ALG8, ALG12, DIE2, OST3, OST6) are presented in the form of sequences, measured expressions under different conditions, corresponding proteins interactions and part of their functional annotations from Gene Ontology.

1.1.2 'Omics' domains

The rise of genomics brought the flood of the '-omics' suffix. Many fields now add it to better denote and differentiate studies on 'omics' data [21]. Some of the terms used in omics research are: Genomics, Transcriptomics, Proteomics, Metabolomics, Functional Genomics, Metagenomics, Epigenomics, Comparative, Evolutionary and Population genomics, Pharmacogenomics. Thesis encompasses studies and presents results that are of interest to cancer genomics, functional genomics and metagenomics. Here we provide general background of these domains.

- **Cancer genomics** aims to uncover molecular basis of cancer. Among different layers of genomic information used in cancer studies gene expression profiles (transcriptome) are the most common. High-throughput gene expression profiling offers a global view on genes activity under different

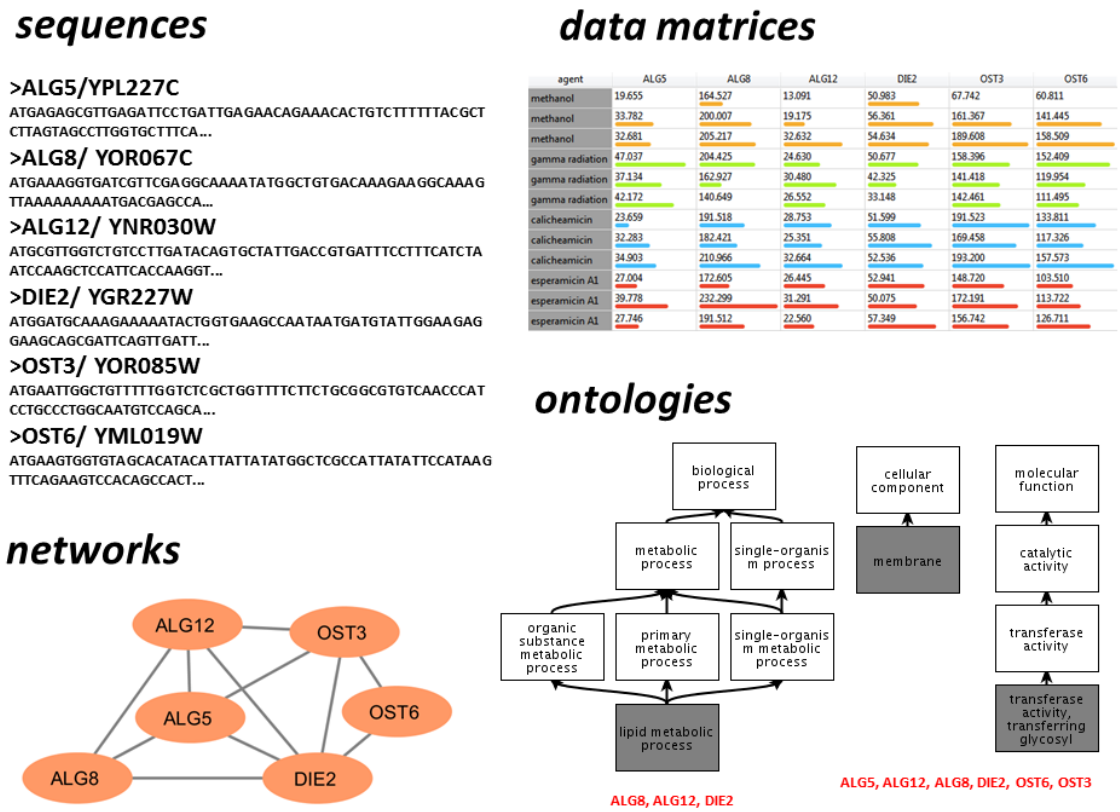


Figure 1.1: Heterogeneous representations of 'omics' data. Example includes six genes of Yeast.

conditions. Large amounts of genome-wide gene expression data are collected and stored in public archives [22]. Harnessing these available data sets can provide important insights into signatures of cancer, that can later serve for the precision medicine.

- **Functional genomics** attempts to answer what are the functional roles of genes/proteins: their molecular functions, locations where they evince functions and biological process to which they contribute [23]. To understand and describe gene/protein functions, researchers integrate information from various omics sources. Computational approaches for predicting gene function provide an opportunity to direct and facilitate experiments that are necessary for function verification. Predicted functions can serve as a starting point for further low throughput analysis. In a systematic fashion CAFA challenge (*The Critical Assessment of protein Function Annotation algorithms*) [24] benchmarks proposed approaches in assessing gene func-

tions on a set of newly experimentally verified functions and continuously tracks progress in the field [25].

- **Metagenomics** encompasses studies of microbial communities that are present in an environmental samples [26]. NGS technology allowed sequencing genetic material directly from the environment in a way not possible before, and thus shedding light on diverse and complex microbial communities that live in/on human, air, water, soil [27]. Sequencing of specific marker genes enables studying the community compositions (i.e. answering the question "Who is there?"), while sequencing total DNA in a samples facilitates studies on functional role and capacity of microbes present in the samples.

'Omics' domains mentioned here are not isolated from each other, rather they are intertwined. For example, functional genomics helps in understanding the functions behind mutations found in cancer genomics. Part of the metagenomics is dedicated to functional annotations of genes discovered in microbiome samples, and new cancer studies include metagenomics into analysis to better characterize different cancers.

1.2 Thesis contributions and structure

Clustering has important role in the analysis of data in all 'omics' domains. For instance, in cancer genomics, the hypothesis is that clusters discovered in expression data correspond to different types or subtypes of cancer. Identification of cancer subtypes could pave the way for more targeted therapies and thus improve patient response. But high-dimensionality, i.e. number of genes, leads to high degree of variability of the results of clustering. Therefore, ensemble approaches in clustering are necessary.

The main contributions of our work include the proposed clustering fusion framework, an algorithm for extracting final clusters after NMF, and evaluation of proposed data fusion technique within the scope of functional genomics, cancer genomics and metagenomics. By combining results of different runs of clustering algorithm we have enhanced the quality and stability of the final clusters. The landscape of integrative clustering algorithms is further explored by in-depth benchmark of the partitions generated by state-of-the-art algorithms on large number of data sets.

Chapter 2 provides brief review on related work: clustering, integrative approaches in clustering, and non-negative matrix factorization. Chapter 3 presents the integrative clustering algorithms based on non-negative matrix factorization. The results of comprehensive benchmarking of the proposed and other related algorithms on 70 data sets are included in Chapter 4. Chapter 5 encompasses

experiments and results obtained within functional genomic application, while chapter 6 extends on the regularized approach. Chapter 7 is dedicated to the application on microbial data sets. Finally, thesis conclusion is provided in the Chapter 8.

The Python library *iclust* developed in the thesis is available at git-hub repository (https://github.com/brdars). Code for baseline clusterings was utilized from well known Python libraries Orange [28] and Scikit-learn [29], while for biological computations we used Biopython [30], bioinformatics add-on of Orange and QIIME package [31].

Chapter 2

Related work

2.1 Clustering

The domain of clustering has amazing variety of proposed algorithms [32]. Clustering aims to group, find structure and summarize data, to compress information, to find representatives or to detect unusual objects - outliers. It belongs to unsupervised learning i.e exploratory data analysis without guidelines from target outputs. A lot of applications benefited from the clustering. For example, grouping pixels based on similar features partitions an image into segments useful for the analysis and further interpretation of the image [33]. Detection of similar purchase behaviour leads to marketing segmentation [34] that enables decision makers to reach customers more effectively. In power systems, identifying typical load curves based on clustering can optimize distribution and supply services [35].

Clustering starts from input data set $X = \{x_1, x_2, ..., x_N\}$ where each observation (object, item, sample) is a d-dimensional real vector–feature vector. Through clustering similar object are grouped into clusters, but meaning of a cluster is under open discussion. Generally, clustering should fulfil properties of homogeneity of objects within cluster and heterogeneity of objects between clusters. The problem of finding clusters arises from dilemmas that user face when choosing algorithm and corresponding parameters. For the most of clustering algorithms defining similarity/distance is an essential part. Another issue in clustering is selecting the criterion functions and appropriate number of clusters. Different choices lead to different clusters.

Many algorithms emerged due to fact that none of the algorithms is suitable for all data sets. In general, we can categorize clustering algorithms into partitioning and hierarchical [36]. Also we can differentiate clusterings by membership functions, and divide them into exclusive, overlapping and fuzzy. In exclusive

(crisp, hard) clustering each object belongs to one cluster, while in soft (fuzzy) [37] it belongs to all clusters with a degree of membership. Overlapping type that limits overlap among clusters can be defined as a trade off between crisp and fuzzy clustering.

2.1.1 Partitioning clustering

Partitioning clustering searches for the K-partition $C = C_1, ...C_K$ of data X by optimizing defined clustering criterion. For hard partitioning, clusters fulfil conditions:

1. $C_i \neq \emptyset, i = 1..K$,
2. $\bigcup_{i=1}^{K} C_i = X$,
3. $C_i \bigcap C_i = \emptyset, i, j = 1..K$ and $i \neq j$.

The most popular among partitioning clustering algorithms is certainly K-means algorithm. Even 50 years beyond [38] of its proposal, it is still widely used due to its simplicity and linear computational complexity. In K-means each cluster is represented by its centre i.e. its prototype that characterizes all objects in cluster. It uses Euclidean distance to assign objects to clusters by optimizing sum of the squared error (SSE):

$$\arg\min_{\mathbf{C}} \sum_{i=1}^{K} \sum_{\mathbf{x} \in C_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|^2 \tag{2.1}$$

where i denotes index of cluster, and μ_i is corresponding cluster centre.

Initial centres of clusters are set randomly or based on some prior knowledge. In the iterative procedure K-means alters centres to minimize objective function and after reaching defined number of the iterations or other stopping criterion returns as a result a local optimum. Example of partitioning 2-d points by K-means is presented in Fig. 2.1. Clusters are labelled with different colours and corresponding centres are denoted with circle symbols. Example illustrates how data points assigned to clusters form Voronoi cells around their centroids.

Another approach in the partition clustering is probabilistic. Probabilistic methods rely on assumption that objects belonging to a cluster are drawn from a specific probability distribution and the overall distribution of the data is a mixture of several distributions. The aim of such algorithms is to estimate the parameters of distributions. Commonly, it is assumed that densities are multivariate Gaussian and then clusters are identified by selecting the density parameters that maximize the likelihood of the data samples.

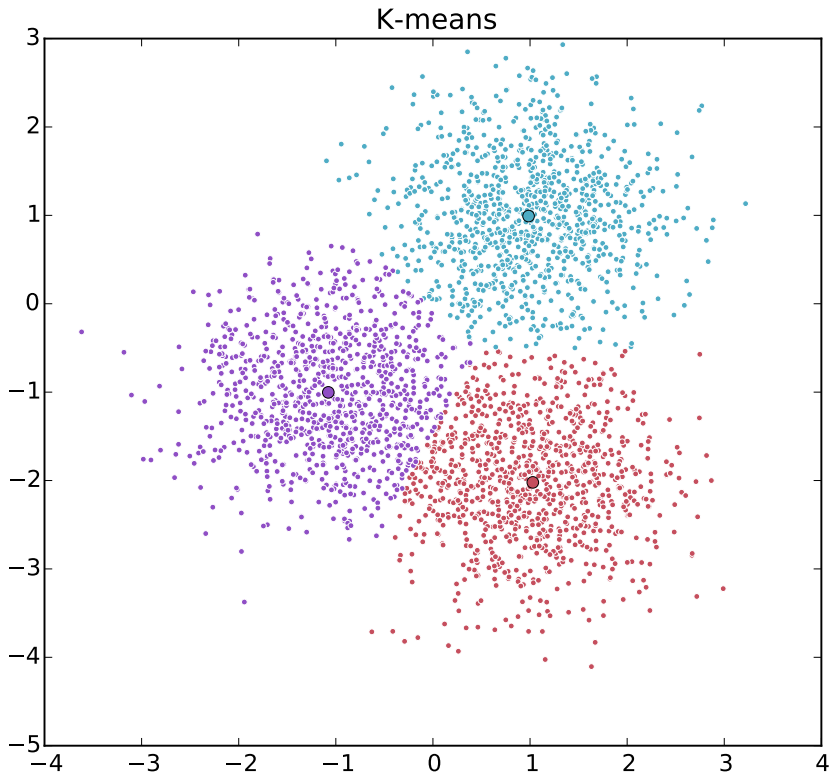


Figure 2.1: Result of K-means clustering. Clusters are labelled with different colours and circle symbols denote corresponding centres.

Partition based approaches also include graph clustering methods [39]. Formally, we can represent undirected graph structure as $G = (V, E, w)$, where V denotes the set of N vertices and E the set of edges with corresponding edge weights w. In a graph based clustering vertices represent objects from data set X and edges of the graph connect objects and have weights that reflect their similarity (proximity, strength of interaction). If initial data are not in the graph structure, there exist several methods to transform them into graph representation. Graph partitioning algorithms identify clusters based on properties of high intra-cluster density and inter-cluster sparsity. Thus, clusters in the graph are vertex subsets with many internal and few external edges. Fig. 2.2 illustrates graph with nodes coloured according to the cluster memberships obtained after applying a graph based clustering algorithm.

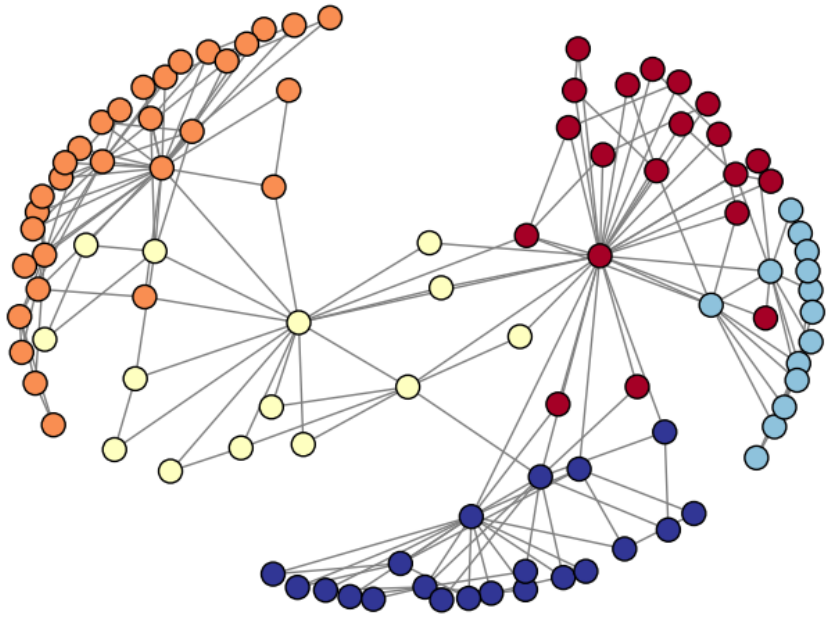


Figure 2.2: A graph structure with nodes coloured according to the result of graph based clustering algorithm

2.1.2 Hierarchical clustering

The results of hierarchical clustering is a sequence of nested partitions of data X, $H = \{H_1, H_2 ... H_Q\}$. If there exist two clusters H_i and H_j such that $H_i \bigcap H_j \neq \emptyset$ then either $H_i \subseteq H_j$ or $H_j \subseteq H_i$. The output in a form of clusters and sub-clusters allows visualization of clustering structure in the form of tree (dendrogram). Leaf nodes are individual objects, each inner node in the tree is the union of its sub-clusters and root is the cluster containing all objects. The final clustering can be obtained by cutting the tree.

Hierarchical structure can be obtained in two ways: agglomerative and divisive. Agglomerative type of hierarchical clustering initially assigns each object to its own cluster and then repeatedly merges pairs of clusters until all reaches the root–all objects in one cluster. Divisive clustering works in an opposite way. It starts from initial cluster that encompass all object and then repeats dividing procedure until all clusters become singleton clusters. Divisive clustering is not commonly used in practice due to very expensive computation ($2^{N-1} - 1$ possible

two-subset divisions for a set of N objects).

Example of agglomerative tree is presented in Fig. 2.3 and it is an informative way to represent data structure. Distance is plotted on horizontal axis and objects are aligned on vertical axis according to the distances between objects and their successive merge into clusters. Vertical lines in the tree correspond to merging step. Tree structure can assist in making decision on the number of clusters in data; large gap in merging clusters unveils possible cut of the three.

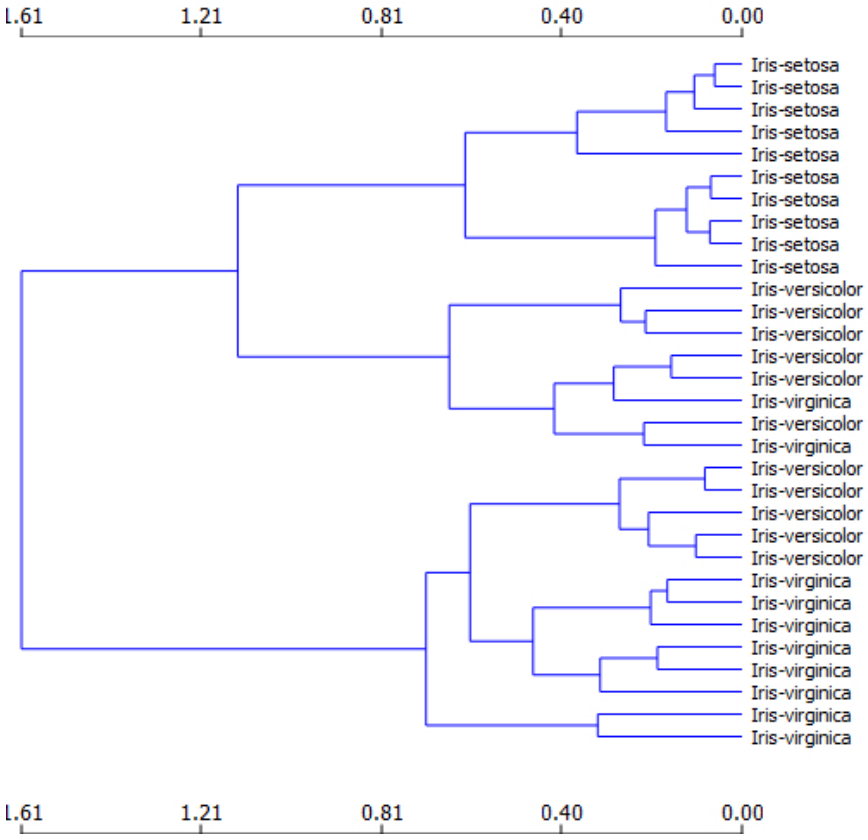


Figure 2.3: Hierarchical clustering dendogram for *iris* data set with corresponding labels.

2.1.3 Clustering validation

Validation of clustering allows us to compare clustering algorithms, decide on the number of clusters, avoid finding patterns in noise. There are three categories of validation criteria: external, internal, and relative. External compare ground-truth labels with those assigned by clustering while internal use underlying dataset alone to measure how well obtained clusters satisfy compactness,

connectedness and/or separation criterion [40]. Relative approaches in validation compare the results of the same algorithm but realized under different values of parameters. Numerous variants of those criteria or their combination exist. Here we describe measures used in the thesis.

Adjusted Rand Index (ARI) [41] compares labels obtained by clustering $C = \{C_1, C_2, \ldots, C_k\}$ against external ground-truth labels $L = \{L_1, L_2, \ldots, L_s\}$ and it is advanced version of Rand Index (RI) [42]. RI quantifies agreement between partitions by counting number of pairs of objects that are clustered together or placed in different clusters in both partitions, and disagreement between partitions by counting number of pairs that are clustered together in one partition but not in the other. ARI corrects RI for a chance that random partitions agree; it ensures that value is then close to 0. Maximum value of 1 is reached when external labels and those assigned by clustering are identical up to a permutation.

To calculate ARI we need a contingency table:

C L	L_1	L_2	$\ldots$	L_s	Sums
C_1	n_{11}	n_{12}	$\ldots$	n_{1s}	a_1
C_2	n_{21}	n_{22}	$\ldots$	n_{2s}	a_2
$\vdots$	$\vdots$	$\vdots$	n_{ij}	$\vdots$	$\vdots$
C_k	n_{k1}	n_{k2}	$\ldots$	n_{ks}	a_k
Sums	b_1	b_2	$\ldots$	b_s	

where n_{ij} denotes the number of overlapping objects between C_i and L_j. Then ARI is expensed with following equation:

$$ARI = \frac{\sum_{ij}\binom{n_{ij}}{2} - [\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}]/\binom{n}{2}}{\frac{1}{2}[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}] - [\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}]/\binom{n}{2}} \tag{2.2}$$

where n_{ij}, a_i, b_j are values from the contingency table.

Normalized Mutual Information (NMI) [43] measures information shared by ground-truth labels and result of clustering. Obtained value is further normalized by product of their entropies.

$$NMI = \frac{I(C, L)}{\sqrt{H(C)H(L)}}\,. \tag{2.3}$$

where $I(C, L)$ denotes mutual information and $H(C)$ and $H(L)$ are entropies associated with clustering:

$$I(C, L) = \sum_{C_i \in C} \sum_{L_j \in L} P(C_i, L_j) \log\left(\frac{P(C_i, L_j)}{P(C_i)\,P(L_j)}\right), \tag{2.4}$$

$$H(C) = -\sum_{C_i \in C} P(C_i) log P(C_i) \tag{2.5}$$

Final NMI score is between 0 (no shared information) and 1 (perfect agreement).

Silhouette index [44] takes into account compactness and separation of clusters. Silhouette is calculated for each object i in the data set and is expressed through ratio (Eq 2.6) defined by two measures $a(i)$ - average dissimilarity to all objects in its cluster and $b(i)$ - minimum average dissimilarity to objects in other clusters.

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \tag{2.6}$$

Finally, average silhouette across all objects in data set is used for validation of clustering. Larger values indicate better overall quality of the clustering result.

Isolation index, firstly introduced for image segmentation [45] and further explored for shape-invariant validation [46], measures a percentage of k-neighbour objects, $v_k(i)$, that are clustered together. In a good clustering neighbouring points should have the same label, hence, for object i in assigned cluster we can expect $v_k(i) \approx 1$, while for a random clustering, $v_k(i) \approx \frac{1}{K}$, where K is the number of clusters.

Isolation index evaluates local homogeneity and it is independent of the cluster-topology. Isolation index for overall clustering is obtained by averaging over all objects in the data set:

$$\frac{1}{N} \sum_i v_k(i) \tag{2.7}$$

In a lack of unique score of quality, we argue that multiple scores should be taken into account in the assessment of clustering results and their comparisons.

2.2 Integrative clustering

To overcome previously described problems of individual clustering, research on integrative techniques gained attention. Integration implies some sort of diversity on its input [47]. Diversity may come from different initialization, algorithms' parameters, feature subsamples, object subsamples, similarity/distance functions and/or different clustering algorithms. When integration includes heterogeneous

data sources, we can think of it as data fusion framework. This framework can be realized through different strategies: early and late [48]. In early integration data is fused before the application of a clustering algorithm by simple concatenation of data or by aggregation of similarity matrices. Algorithms based on multiple runs of individual clusterings and followed by procedure of merging obtained clusters, are denoted as late integration techniques. Here we introduce five algorithms that are used in the thesis along with the proposed algorithm.

Consensus clustering CONS [49] is a well-known late integration approach. Originally, it was proposed for integration of different clusterings obtained from samples of the same data sets and later was broadly explored in other set-ups. Consensus clustering integrates cluster memberships, in a pairwise manner, into a consensus matrix, where its elements refer to the proportion that two objects were clustered together out of the number of times they were present in the input clusterings. Consensus matrix can be viewed as a similarity matrix and post-processed through additional methods to obtain final clusters.

Two other extensively used ensemble clustering algorithms are **Hypergraph partitioning algorithm HGPA**, and **Meta clustering MCLA** [50]. Both algorithms represent clusters as hyperedges. To produce output clustering, HGPA partitions hypergraph into k unconnected components by cutting a minimal number of hyperedges. MCLA firstly creates meta-graph based on similarities between hyperedges determined by binary Jaccard measure, than groups related hyperedges into k meta-clusters by graph based clustering algorithm, collapses related hyperedges and finally assigns each object to the most associated meta-cluster.

Divisive Clustering Ensemble DICLENS algorithm [51] calculates inter cluster similarity among clusters in the ensemble based on a coexistence information and then creates minimum spanning tree. Every possible cluster is a subtree of minimum spanning tree. Algorithm seeks for the best final clustering by iteratively removing minimum weighted edge in the tree. At each iteration connected components are evaluated as potential clusters. By majority voting objects are assigned to clusters and then intra-cluster and inter-cluster similarity are measured and combined to form one quality function. The final clustering is the one that maximizes the quality function. DICLENS algorithm does not require setting the number of clusters in the final clustering.

Optimized kernel k-means clustering OKKC [52] objective is to optimize the kernel combination. The algorithm starts from normalized centered kernel matrices $G_1; ...; G_p$ that could be inferred from heterogeneous data sets, different kernel functions and/or feature subsets. The algorithm combines kernel matrices in parametric linear additive manner. The parameter determines the strength of constraints imposed on the coefficients that multiply kernel matrices in an optimization process. The optimization is bilevel - it iteratively optimizes cluster membership matrix and coefficients of kernel matrices. The final crisp

cluster assignments are obtained with k-means applied on multi-cluster membership matrix.

In this thesis, we further propose an alternative technique for cluster integration [53] that relies on non-negative matrix factorization (NMF) [54]. In the algorithm inferences of clusters are made separately from each data set, similarity measure or other source of input diversity and then combined to obtain final clusters. The algorithm is described in the next chapter and extended with regularization part in the Chapter 6.

2.3 Non-negative matrix factorization

Non-negative matrix factorization (NMF) is a decomposition method that approximate matrix $R \in \mathbb{R}^{m \times n}$ with matrix factors $W \in \mathbb{R}^{m \times k}$ and $H \in \mathbb{R}^{k \times n}$ ($R \approx WH$), by imposing non-negativity constraints on W and H. Choice of inner dimension k of the matrix product WH is the problem dependent and is usually chosen as $k < min(m, n)$ thus causing low-rank approximation of initial matrix with aim to discover latent structures in the data.

NMF allows only additive combinations of basis components. In this way it obtains the parts-based representation of initial matrix. Objective of NMF is the minimization of the reconstruction error in representing R with WH product. Different loss functions have been proposed, but commonly optimization methods minimize the square error:

$$\frac{1}{2}\|R - WH\|_F^2 = \frac{1}{2}\sum_{ij}(R_{ij} - [WH]_{ij})^2 \tag{2.8}$$

where F denotes the Frobenius norm, or Kullback-Leibler divergence:

$$D_{KL}(R\|WH) = \sum_{ij}(R_{ij} \ln \frac{R_{ij}}{[WH]_{ij}} - R_{ij} + [WH]_{ij}) \tag{2.9}$$

Solution of NMF is not unique and for some NMF algorithms convergence is not guaranteed. If algorithm converge, then it is usually only to local minimum. For many applications even local minimum can provide desirable results. To minimize Eq. 2.8 or Eq. 2.9 different strategies are proposed: multiplicative update algorithms, gradient descent and alternating least squares, projected gradient bound-constrained optimization [55].

Multiplicative update algorithms keep one factor fixed and updates other. In the next step factors are interchanged. Algorithms repeat those update steps until maximum number of iterations is reached or error is below defined threshold. W

(a) **(b)**

Figure 2.4: Example of NMF: (a) Part of face database (b) 64 basis components obtained after NMF decomposition.

and H remain positive throughout the iterations. This optimization approach is parameter free, but converges slowly.

Important step before the optimization starts is the initialization of W and H. In a standard approach W and H are initialized with random nonnegative values. More advanced approaches are NNDSVD [56] and the initialization based on clustering [57]. NNDSVD involves two SVD processes; one approximates the data matrix and the other positive section (nonnegative elements and 0 elsewhere) of the obtained SVD factors. This initialization contains no randomization and thus provides unique solution. Experiments indicate that NMF converge faster when initialized with NNDSVD. A cluster based approach takes centroids of clusters obtained by k-means algorithm on the input matrix R to initialize the basis vectors of factor matrix.

The concept of NMF was introduced in factor analysis of environmental data [58]. After applied on learning parts of faces and semantic features of text [54], method received great attention and broadly outspread into other application areas [59]. Fig. 2.4 presents intuitive interpretation of combining parts of face to form a whole. Here, images of faces (Fig. 2.4a) are columns of the input matrix R, columns of W are basis vectors and the rows of H contain encoding coefficients. Basis vectors (Fig. 2.4b) manifest part based representation of the data as they correspond to parts of the faces mouth, nose, eyes, cheeks, etc.

Another intuitive example of part based representation discovered by NMF

are text mining applications. After factorization of initial matrix of words counts in articles, basis vectors contained semantic features that correspond to different topics–groups of semantically related words. For example, discovered basis vectors were (government, council, culture, supreme, constitutional, rights justice), (disease, behavior, glands, contact, symptoms, skin, pain, infection), etc. Factor matrix with encoding coefficients indicate which topics are co-activated in the articles.

Approaches based on NMF have become widely accepted for the analysis of bioinformatics data [60] and useful tools have emerged [61], [62]. NMF has been applied to reduce dimensions in microarray data and infer reduced features–metagenes–that were then later for clustering and visualization [63]. In another example, Wang *et al.* [64] reduced data dimensions by least squares NMF. The authors observed improved results when uncertainty measurements of gene expression data were incorporated in the algorithm. Zheng et al. used NMF for clustering cancer gene expression data [65]. A Specific NMF application was reported by Greene *et al.* [66], where the authors proposed to ensemble non-negative matrix factorizations of proteins pairwise similarity matrices, each obtained with different random initialization of the method. In a text mining study, Chagoyen [67] developed a corpus of gene-relevant documents and relied on NMF to transform the initial high dimensional vocabulary space into reduced semantic representation. Hierarchical clustering was then used to group genes in the new feature space. Discovered groups were functionally coherent, but the authors limited the evaluation to only eight functional terms.

Major part of this thesis, as well, covers bioinfotmatics applications. Several problems from functional genomics, cancer genomics and metagenomics were examined through the lenses of proposed technique for integrative clustering based on NMF.

Chapter 3

NMF for Integrative Clustering

Previous chapter introduced all relevant ingredients for NMF based ensemble clustering: baseline clusterings, assembling concepts and non-negative matrix factorization. Combining results of baseline clusterings together into an ensemble leverages evidence accumulation in order to improve the results of clustering. Method for integrative clustering proposed here builds upon ideas from work presented by Greene and Cunningham [68] that proposed late integration approach. Their work is here further elaborated and extended.

3.1 Ensemble creation

The result of individual clustering from different data set/similarity measure combinations can be presented as a matrix of cluster memberships [68], where one dimension represents items (objects) and the other clusters. Cluster memberships by baseline methods are all crisp and the values in membership matrix are either 1 or 0, indicating whether a item was assigned to a specific cluster. Clustering information from individual clusterings were merged by concatenating membership matrices in the cluster dimension to obtain the joint cluster membership matrix $R = \{0, 1\}^{m \times n}$, where m is the total number of clusters from all clusterings and n is the number of objects considered. A small example of the matrix of cluster memberships R can be seen on Fig. 3.1.

NMF finds an approximation $R \approx WH$, where W and H are two non-negative factors such that $W \in \mathbb{R}^{m \times k}$ and $H \in \mathbb{R}^{k \times n}$. Parameter k is a factorization rank and equals to the desired (target) number of clusters. In the resulting factorization the matrix W contains encoding coefficients while rows of H are the basis vectors that can be interpreted as (continuous) memberships to target clusters discovered by factorization.

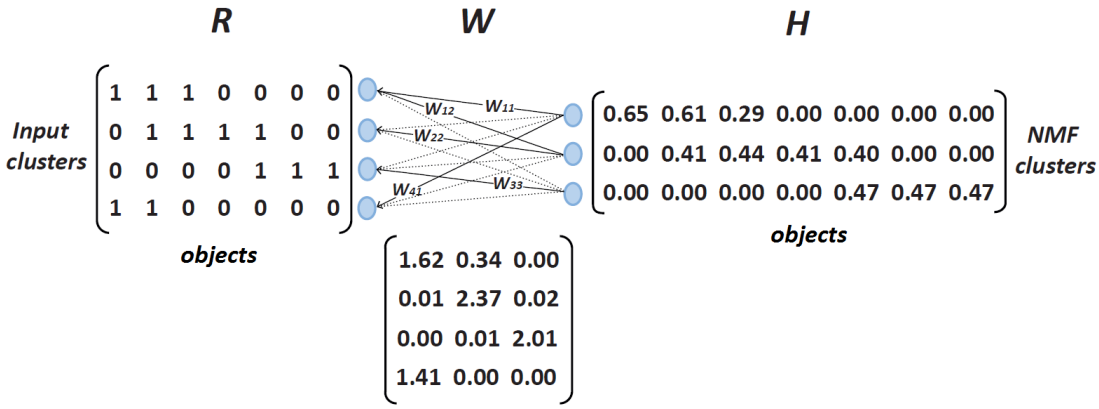


Figure 3.1: Example of NMF decomposition. The original matrix with crisp memberships to four clusters R is transformed to new membership matrix H with three clusters and fuzzy memberships. Encoding matrix W contains weights that indicate how input clusters are intertwined.

3.2 Matrix decomposition

NMF used an algorithm with multiplicative updates [69]. Since our input matrix is sparse, multiplicative updates also provide sparse solutions and there is no need to include regularization into the process of factorization. Values of H and W are iteratively updated (Eq. 3.1 and Eq. 3.2) by multiplying the current values with the factors that depend on the quality of the approximation $R \approx WH$:

$$H \leftarrow H. * ((W^T R)./(W^T W H)), \tag{3.1}$$

$$W \leftarrow W. * ((RH^T)./(WHH^T)). \tag{3.2}$$

Under the multiplicative updates, approximation of R improves monotonically in the Frobenius norm of reconstruction error:

$$\|R - WH\|_F^2 = \sum_i \sum_j [R_{ij} - (WH)_{ij}]^2 \tag{3.3}$$

The optimization starts with matrices W and H computed by non-negative double singular value decomposition (NNDSVD) [56], used for speeding up the convergence of the optimization and supporting the reproducibility of the results.

3.3 Clusters reconstruction

The cluster reconstruction process involves setting the threshold on object cluster memberships. Fig. 3.1 illustrates NMF decomposition of an example cluster membership matrix. For thresholding, we implement a scaling procedure described below. Namely, the results of non-negative matrix factorization are not necessary unique. There may exist nonsingular matrices $D \in \mathbb{R}^{k\times k}$ that satisfy $WD \geq 0$ and $D^{-1}H \geq 0$, and we can rewrite factorization as:

$$WH = WDD^{-1}H = W^*H^* \tag{3.4}$$

Matrix D can perform transformations such as scaling or permutation. Difficulty in determination of new clusters comes from a scale variance. Instead of factorization presented in Fig. 3.1 which results in a pair of coefficients $w_{3,3} = 2.10$ and $h_{3,5} = 0.47$, NMF can also result in $w_{3,3} = 1.82$, $h_{3,5} = 0.54$ (other values in W and H are also changed). Therefore, it would not be appropriate to assign an absolute threshold value for creation of new clusters. In order to eliminate encoding variations we rescaled the columns of encoding matrix W and rows of basis matrix H, and use the following two diagonal matrices D_W and D_H:

$$D_W = diag([max(w_{:,1}), max(w_{:,2})...max(w_{:,k})]) \tag{3.5}$$

$$D_H = diag([max(h_{1,:}), max(h_{2,:})...max(h_{k,:})]) \tag{3.6}$$

Part of the procedure used in binary matrix factorization [70] was suitable for rescaling obtained W and H. For matrices D_W and D_H, the following relations hold:

$$D_W = D_W^{1/2}D_W^{1/2} \qquad D_H = D_H^{1/2}D_H^{1/2} \tag{3.7}$$

$$D_W^{-1} = D_W^{-1/2}D_W^{-1/2} \qquad D_H^{-1} = D_H^{-1/2}D_H^{-1/2} \tag{3.8}$$

$$\begin{aligned}\tilde{R} = WH &= (WD_W^{-1})(D_W D_H)(D_H^{-1}H) \\ &= (WD_W^{-1/2}D_H^{1/2})(D_H^{-1/2}D_W^{1/2}H)\end{aligned} \tag{3.9}$$

In equation (5) rescaling matrix D can be expressed as $D = D_W^{-1/2}D_H^{1/2}$:

$$W^* = WD_W^{-1/2}D_H^{1/2} \quad H^* = D_H^{-1/2}D_W^{1/2}H \tag{3.10}$$

Transformations of W and H into W^* and H^* keep product WH unchanged, but ensure that values in the encoding and basis matrices are comparable and can be interpreted. Each element of W and H is rescaled in the following manner:

$$w^*_{i,k} = w_{i,k}\sqrt{\frac{max(h_{k,:})}{max(w_{:,k})}} = \frac{w_{i,k}}{max(w_{:,k})}\sqrt{max(w_{:,k})max(h_{k,:})} \tag{3.11}$$

$$h^*_{k,j} = h_{k,j}\sqrt{\frac{max(w_{:,k})}{max(h_{k,:})}} = \frac{h_{k,j}}{max(h_{k,:})}\sqrt{max(h_{k,:})max(w_{:,k})} \tag{3.12}$$

We infer the membership to k new clusters from coefficients in W^* and H^* in either overlapping or exclusive manner. In overlapping clustering, objects may belong to more than one cluster, while in exclusive clustering, each object is assigned only to one, most likely cluster. Overlapping clustering assigns objects to clusters according to their membership coefficients in H^*, but only if the membership exceeds the threshold of 0.5. For exclusive clustering, additional ranking is used that takes into account the importance of a object within cluster and strength of cluster. Importance is derived from H^* and strength from W^*. The ranking algorithm can be summarized by the pseudo code given in Algorithm 1.

Algorithm 1: Extraction of clusters

Inputs: $W^* \in \mathbb{R}^{M\times K}$, $H^* \in \mathbb{R}^{K\times N}$,
objects [o_1, o_2, ... o_N], $T_r = 0.5$
Outputs: clusters C = [c_1, c_2, ... c_K]
$WSUM^* \leftarrow$ sum over columns W^*
for $k \leftarrow 1 : K$ **do**
  for $j \leftarrow 1 : N$ **do**
    if clustering = overlapping **then**
      if $h^*_{k,j} \geq T_r$ **then**
        append cluster c_k with object o_j
      end if
    else
      if $(h^*_{k,j} \geq T_r)$ and $(h^*_{k,j} * wsum^*_k = \max(h^*_{k',j} * wsum^*_{k'}$, for $k' \leftarrow 1 : K))$ **then**
        append cluster c_k with object o_j
      end if
    end if
  end for
end for

Factorization of the input matrix R is iterative and runs for defined number iterations or stops earlier if reconstruction error is below specified threshold.

In what follows we demonstrate the utility of proposed technique. In each chapter we focus on a specific field of study: cancer genomics (Chapter 4), functional genomics (Chapter 5 and 6) and metagenomics (Chapter 7). Chapter 4 also provides results of extensive benchmarking. Regularized version of the algorithm that allows including side information into factorization is presented within Chapter 6.

Chapter 4

Comprehensive benchmarking of integrative clusterings

Thoroughly comparison of integrative clustering algorithms would provide general assessment of existing algorithms and valuable guidance on selecting appropriate one. However, large benchmarking studies on integrative clustering are missing due to fact that such comparative studies are non-trivial tasks. Large number of experiments on real and artificial datasets with different characteristics are necessary. While few studies managed to benchmark different clustering algorithms at larger scale [71], [72], comparing the performance of integrative clusterings calls for further efforts since it involves not just running baseline clusterings, but also creating ensemble and final extraction of clusters.

In our study we benchmarked 6 integrative clustering algorithms on 70 synthetic and real word datasets. To assess the quality of clusterings we used 4 validation measures: 2 external and 2 internal. Externals are Adjusted rand index and Normalized mutual information, while Silhouette and Isolation index are internal measures. All used measures are explained in cluster validation subsection of the Chapter 2.

The comparative study encompassed NMF integrative clustering algorithm, that was proposed and described in the previous chapter, along with CONS, HGPA, MCLA, DICLENS, OKKC, all briefly described in integrative clustering subsection of the Chapter 2 that is devoted to the related work. All algorithms, except OKKC, employ late integration ensemble approach. OKKC performs early integration at the level of kernel matrices.

As baseline clustering method we used kernel k-means [73] with random initialization. Kernel k-means was selected to enable comparisons of late integration

algorithms with OKKC that fuses different kernel matrices. We used either radial basis function (RBF) or linear kernel function. Diversity of RBF kernels is achieved by different kernel width σ and additionally increased, if necessary, by different feature subsets. Diversity of linear kernels is achieved by different feature subsets. In procedure of building the kernels ensemble we followed the rule of moderate level of diversity [74] implying that in the case of low diversity we can not benefit from ensemble and too much diversity can be harmful to the final clustering.

4.1 Data sources

To systematically evaluate the performance of integrative clustering algorithms we used diverse collection of data sets:

1. Data from UCI machine learning repository [75]: 20 data sets. The collection includes *iris*, *wine*, *seed*, *wdbc*, *breast-cancer-wisconsin-cont*, *satimage*, *pendigit*, *image-segmentation*, *zoo*, *letter-recognition*, *soybean*, *yeast*, *shuttle*, *optdigits*, *parkinsons*, *ecoli*, *movement-libras*, *semeion-handwritten*, *dermatology*, and *led7digit* data set.

2. Synthetic data: 20 data sets suitable for clustering with kernel K-means. The collection includes: *aggregation*, *compound*, *pathbased*, *d31*, *flame*, *r15* from cs.joensuu.fi/sipu/datasets; *hepta*, *lsun*, *tetra* from Fundamental Clustering Problems Suite (FCPS) [76]; and *2dnormals*, *cassini*, *cuboids*, *hypercube*, *shapes*, *simplex*, *smiley*, *waveform*, *twonorm*, *xor* that are available in mlbench package [77]. Data from mlbench package were generated with number of samples set to 500.

3. Cancer genomic data: 30 data sets, obtained from Affymetrix or cDNA microarrays and selected to benchmark performance of clustering algorithms in recovery of cancer types [78]. Originally this collection contains 35 data sets, but in our experiments with kernel K-means as baseline algorithm, in five data sets we could not detect clusters that have any agreement with true labels. Thus, we excluded those from our study.

Properties of data sets are provided in the Appendix A - Benchmarking datasets, tables A1, A2 and A3. Tables include information on the data set names, sample sizes, numbers of clusters and types of kernel that were used in the experiments.

4.2 Illustrative example

Before reporting on overall benchmarking results, we here present an example that sheds light on the integrative clustering. In the illustrative example we used simple 2-dimensional data set, *smiley*, that is a part of synthetic data sets collection (Table A2). The dataset consists of 500 instances and contains 4 natural clusters: two eyes, nose and mouth (Fig. 4.1a). The clusters have different shapes. While eyes can represent Gaussian clusters, nose is triangular group of point and month has elongated moon like shape. Ground truth of clustering is coloured according to natural clusters in the data set. Example allows us to visually explore the different results of clusterings obtained by kernel k-means (see Fig. 4.1b to 4.1e). Fig. 4.1b and 4.1c present results of kernel k-means across both features $x1$ and $x2$. We can observe that colouring, corresponding to the results of clusterings, is different. This disparity comes from random initializations that further caused kernel k-means to converge to the different local optimums. Fig. 4.1d and 4.1e show results of kernel k-means across one of the features, $x1$ or $x2$. In this way, clustering can uncover dense regions in lower dimensional spaces. Clustering results (Fig. 4.1b to 4.1e) along with 6 other similar realizations were assembled into NMF framework and final clusters are presented in Fig. 4.1f. We can notice that NMF most closely reaches to the ground truth labels with only few points not coloured according to the natural cluster they belong to. This illustrative example thus shows benefits of integrating diverse results.

4.3 Benchmarking results

We conduct a comprehensive experimental analysis to test performance of ensemble clustering algorithms. In our experiments final output of ensemble clustering algorithms is crisp partition that assigns object exactly to one cluster. For each data set we ran 100 repetitions, where in each repetition we created random ensemble of 10 kernel matrices, applied kernel k-means to produce ensemble of clusterings for all examined algorithms, except for OKKC that works directly on kernel matrices, and finally ran ensemble algorithms. The same randomly created ensemble of kernel matrices used in all algorithms allowed us to fairly rank them on 4 validation measures.

Enough number of repetitions allowed us to reliably estimate average ranking. Based on the average ranking we produced final ranking for each data set. To compare multiple algorithms over multiple data sets we evaluated the differences between average ranks across data sets with method [79] based on Nemenyi test. Sample size in the test refers to the number of data sets used, and along with the number of benchmarked algorithms, determines the value of critical difference (CD) used for identifying significantly different performance among methods.

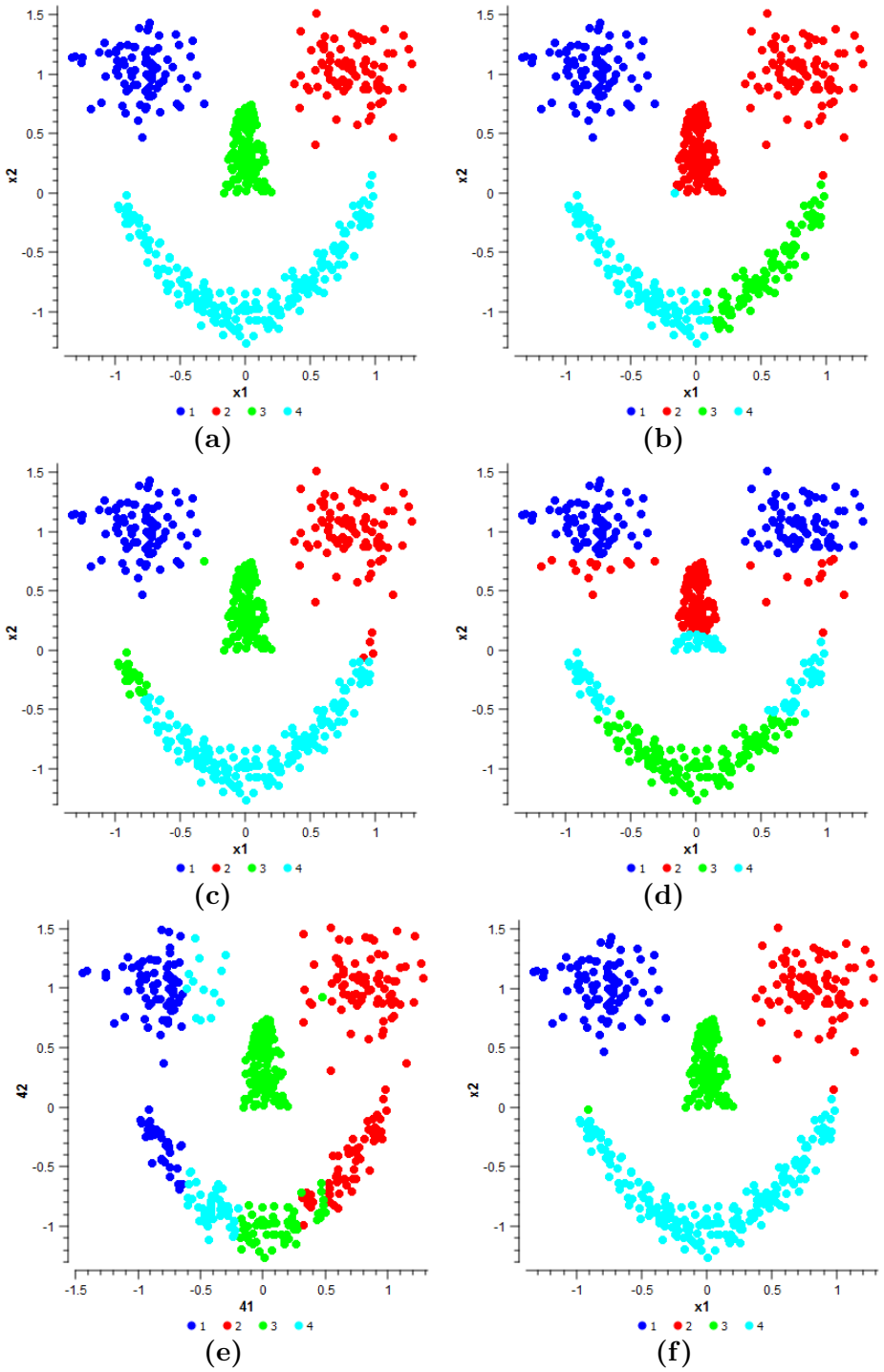


Figure 4.1: Illustrative example of integrative clustering: (a) ground truth, (b) - (e) results of diverse individual clusterings obtained by kernel k-means and (f) NMF ensemble result.

We first report on the results obtained on each of three used data collections: UCI, synthetic and genomics, and then merge results to report overall performance. Results are presented on graphs with average ranks of evaluated algorithms; smaller rank indicates better performance. Lines on the graph connect the groups of algorithms that are not significantly different at $p - value < 0.05$ cut-off.

The results for UCI data sets are presented in Fig. 4.2, for synthetic data sets in Fig. 4.3 and for genomic data sets in Fig. 4.4. NMF integrative clustering ranked first, closely followed by CONS when evaluated on external labels and silhouette index on all three data collections. The third ranked algorithm according to the adjusted rand index is MCLA, but based on normalised mutual information DICLENS is on the third place. For isolation index, rankings of the best performing algorithms lack consistency across data collections. On UCI and cancer genomics data the best three ranked algorithms are DICLENS, CONS and NMF, while on synthetic collection CONS, NMF, OKKC are the best. Possible explanation is that silhouette index favours hyper-spherical clusters, type of clusters that baseline linear kernel k-means detects, while Isolation can detect arbitrary shaped clusters. However the results of the ranking have to be analysed in the context of significant difference. Methods with insignificant differences in ranking are connected with line. For example, in Fig. 4.2a, NMF significantly outranked OKKC and HGPA, but the result does not imply significant difference to CONS, MCLA and DICLENS.

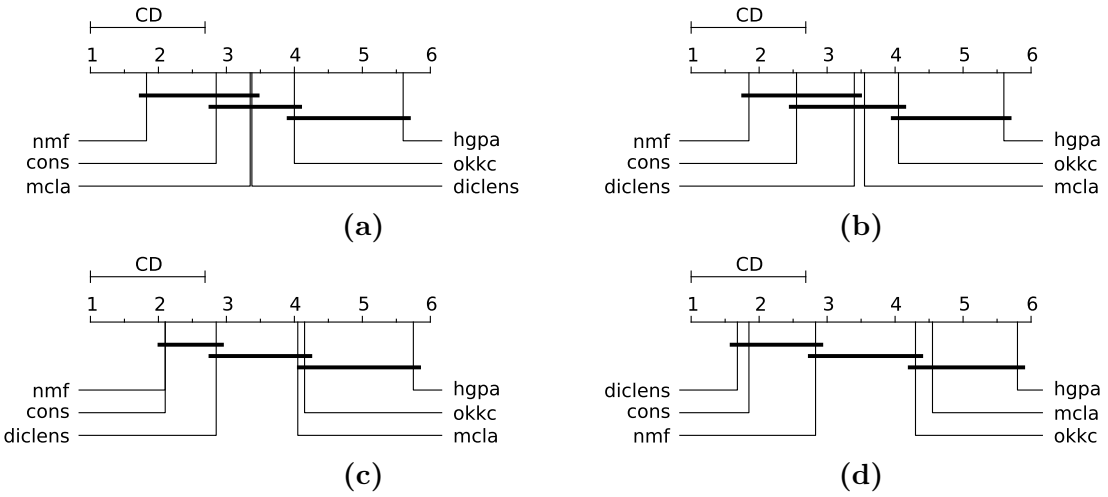


Figure 4.2: Average ranks from integrative clustering results validated by: (a) Adjusted Rand Index, (b) Normalized Mutual Information, (c) Silhouette Index and (d) Isolation Index. Comparison on 20 UCI datasets. Smaller rank indicates better performance. CD denotes critical difference in ranks necessary to evince significantly different performance.

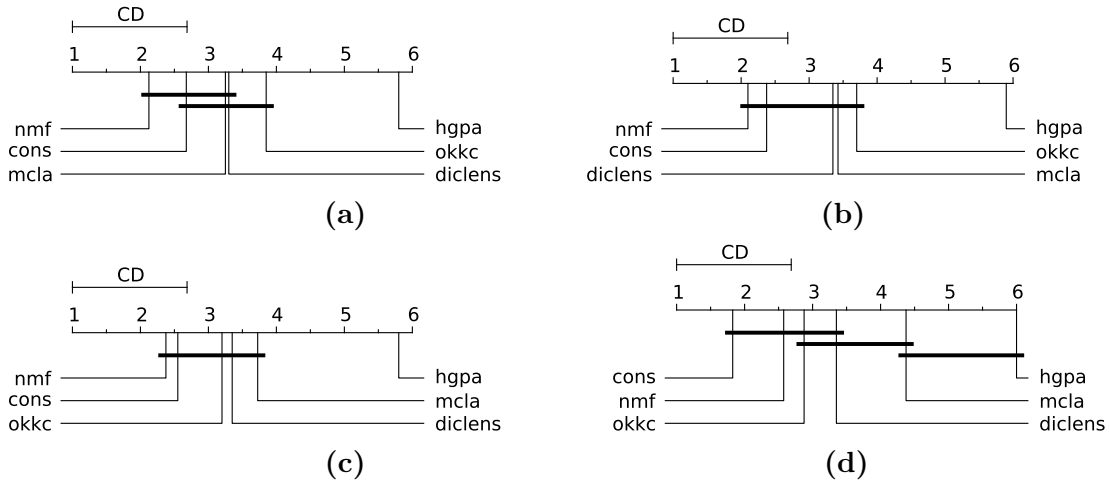


Figure 4.3: Average ranks from integrative clustering results validated by: (a) Adjusted Rand Index, (b) Normalized Mutual Information, (c) Silhouette Index and (d) Isolation Index. Comparison on 20 synthetic datasets. Smaller rank indicates better performance. CD denotes critical difference in ranks necessary to evince significantly different performance.

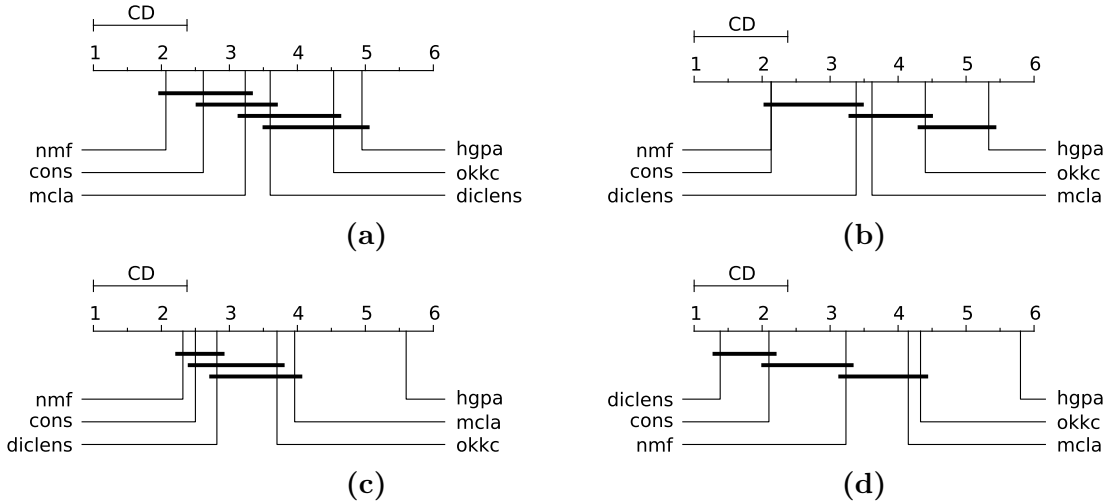


Figure 4.4: Average ranks from integrative clustering results validated by: (a) Adjusted Rand Index, (b) Normalized Mutual Information, (c) Silhouette Index and (d) Isolation Index. Comparison on 30 cancer genomics synthetic datasets. Smaller rank indicates better performance. CD denotes critical difference in ranks necessary to evince significantly different performance.

The number of data sets used to compare the performance of the algorithms affects the length of critical difference. Therefore, we further investigated the differences between average rank for integrative clustering algorithm across all 70 data sets. Comparison results are presented in Fig. 4.5. According to the adjusted rand index and normalised mutual information, the ranking for NMF is here significantly different from MCLA, DICLENS, OKKC and HGPA, while the ranking of CONS is below NMF, but not significantly. When evaluated with silhouette index, NMF significantly outperformed OKKC, MCLA, and HGPA, while its ranking is close to the second and third ranked algorithm CONS and DICLENS, respectively. Isolation index evaluation indicate CONS, DICLENS and NMF as the group with significantly better performance than other three algorithms. Performance of HGPA was consistently poor.

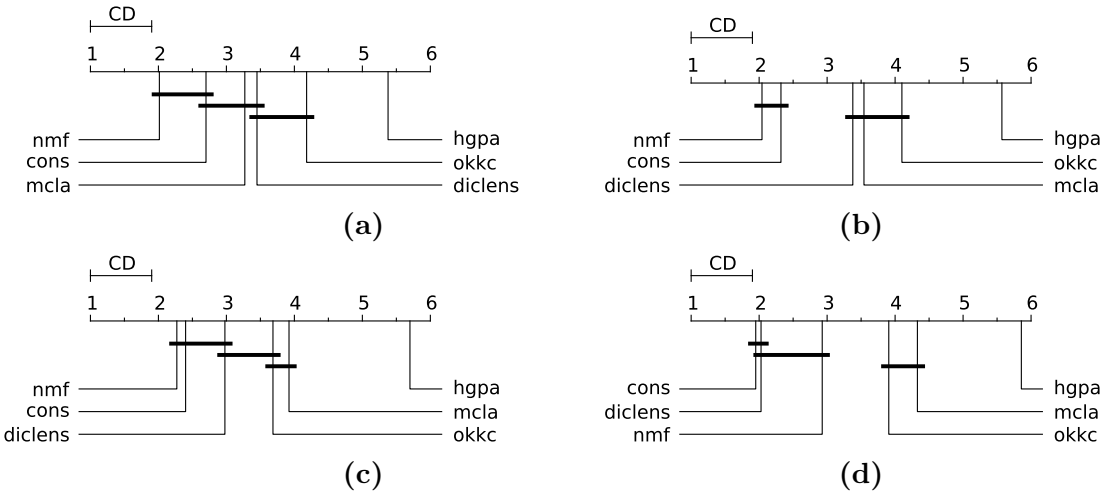


Figure 4.5: Average ranks from integrative clustering results validated by: (a) Adjusted Rand Index, (b) Normalized Mutual Information, (c) Silhouette Index and (d) Isolation Index. Comparison on 70 datasets (UCI + synthetic + cancer genomics). Smaller rank indicates better performance. CD denotes critical difference in ranks necessary to evince significantly different performance.

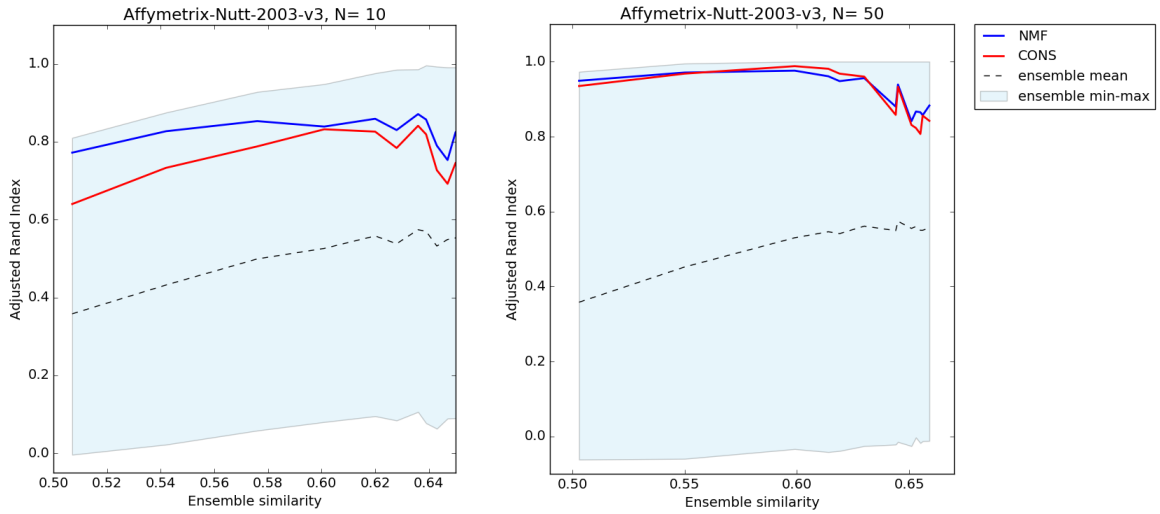
Detailed results of benchmarking are available in the Appendix B, for UCI data sets Tables: B1, B2, B3 and B4, for synthetic data set Tables: B5, B6, B7 and B8, and for genomics data Tables: B9, B10 B11 B12.

4.4 Further insights from cancer genomics data sets

Among three data collections used in this comparative study, particularly interesting is the one related to the cancer genomics domain. This collection is challenging for the clustering due to high dimensionality of the feature space (the number of genes) and high level of noise in the measured values of genes expressions. Here, we selected a few data sets to highlight relationship between diversity of the ensemble and achieved adjusted rand index results. The best two performing algorithms, NMF and CONS, are further evaluated under different diversities of input ensembles and the number of partitions that constitute ensemble. DICLENS was not examined due to its large complexity coming from the calculation of inter-cluster similarity between all clusters (number of partitions x number of clusters) that is followed by construction of a minimum spanning tree and its partition.

The first analysed data set *Affymetrix-Nutt-2003-v3* contains gene expressions of 2 brain cancers. Data set has 22 samples and 1152 features. Results of 100 ensemble clusterings iterations on *Affymetrix-Nutt-2003-v3*, run as a part of overall benchmarking of algorithms, were extended with 100 additional experiments in order to further examine the impact of a diversity/similarity between partitions in the ensemble. Larger diversity was induced by selecting 10-100% of the features on random. Also, number of partition was increased at 50, to allow comparing those larger ensembles with smaller sized ensembles containing 10 results of clustering. Fig. 4.6 presents results of experiments on *Affymetrix-Nutt-2003-v3* data set with (a) ensemble size 10 and (b) ensemble size 50. Similarity range was divided into 10 intervals and obtained results were averaged across intervals. On *Affymetrix-Nutt-2003-v3* similarity of the input ensemble varies from 0.50 to 0.65. An ensemble composed of the same partitions would have similarity of 1. From the graphs we can observe how slightly higher diversity (smaller similarity) among partitions in the ensemble helps integrative algorithms, but enforcing too much diversity eventually degrades performance of the integrative clustering. If we compare the results of assembling 10 to 50 individual clusterings performance of NMF and CONS raised at th level of the best component in the ensemble. Both algorithms benefited from the larger evidence accumulation. Under higher diversity scores for NMF and CONS remained stable.

Another data set that we analysed is *cDNA-Bredel-2005*, containing 50 samples of 1739 gene expressions in human gliomas, with labels corresponding to the 3 subclasses. Fig. 4.7 summarizes results of experiments, run as those in the previous example with ensemble sizes of 10 and 50, diversity further induced by random feature subsampling, and through 200 experimental iterations. Average similarity between partitions in the ensemble ranged from 0.36 to 0.47 for the

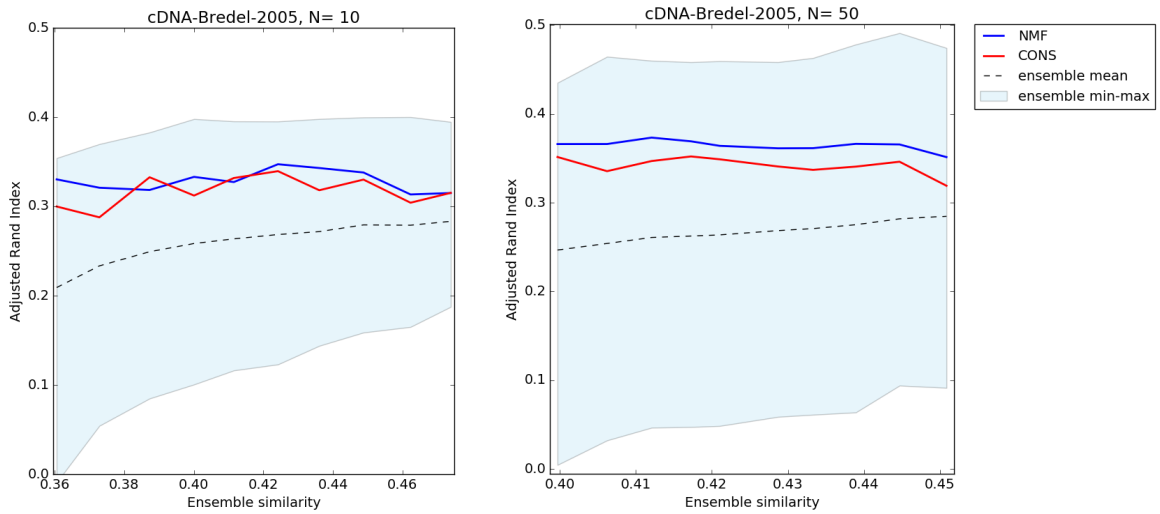


(a) **(b)**

Figure 4.6: Max, min and average adjusted rand index scores from ensemble of (a) 10 (b) 50 kernel k-means clusterings and corresponding NMF and CONS results as a function of ensemble similarity on Affymetrix-Nutt-2003-v3 date set. NMF and CONS results are denoted with blue and red lines, respectively.

ensemble size of 10 and for the larger ensembles similarity varied from 0.4 to 0.45. In this example scores of NMF and CONS were not so close to the best partition in the ensemble, bur still outperformed average score of the ensemble. Smaller agreement between clustering and true labels along with larger diversity of partitions across ensemble prevented integrative algorithms to be at the level of the best as in the previous example.

The last example shows the worst case scenario of integrative clustering that we observed in our experiments. Selected example comes from the analysis of *cDNA-Khan-2001* set, encompassing gene expression signatures of round blue-cell tumors. Data set has 1069 features (genes) and 83 samples that belong to four distinct diagnostic categories. Fig. 4.8 presents obtained results from experiments designed as in the previous two examples. Ensemble similarity ranged from 0.34 to 0.54. Adjusted rand index scores of the kernel k-means partitions varied highly from poor as 0.05 to high as 0.85. However, the results of both examined integrative clusterings, NMF and CONS, achieved scores near 0.35. Larger sized ensembles did not improved results. Integrative results were at the average level of partitions in the ensemble. Integration of clusterings from previous example overpassed the average, but here failed in accomplishing that. A likely explanation is that the mean score of the ensemble, which is closer to its minimum

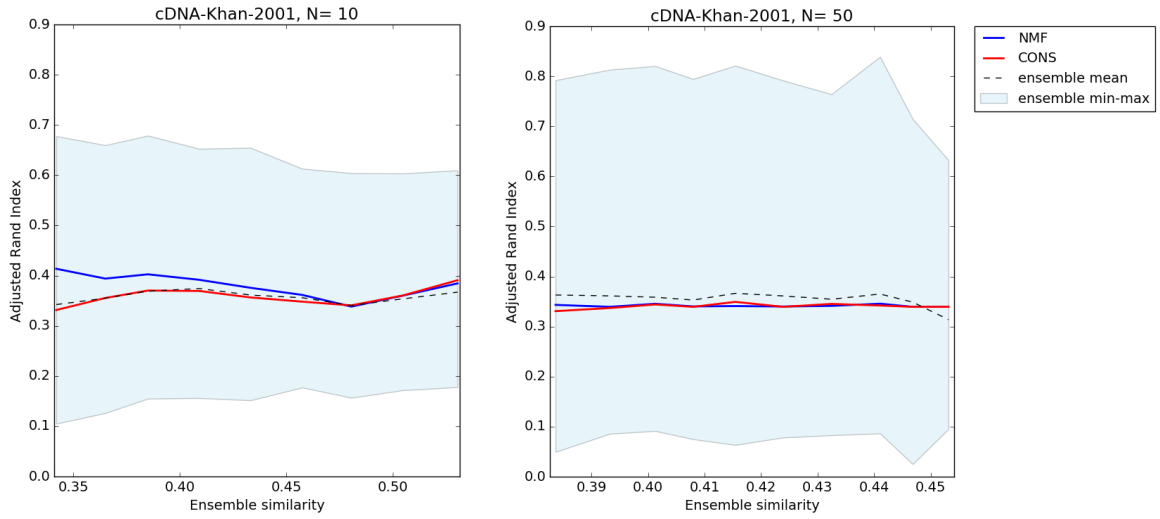


(a) **(b)**

Figure 4.7: Max, min and average adjusted rand index scores from ensemble of (a) 10 (b) 50 kernel k-means clusterings and corresponding NMF and CONS results as a function of ensemble similarity on cDNA-Bredel-2005 date set. NMF and CONS results are denoted with blue and red lines, respectively.

than maximum, along with large disagreement between partitions produced such result.

The three examples revealed possible outcomes of the integrative clusterings: good, where final result is at the level of the best in the ensemble, moderate that scores between average and the best, and the lowest where integration produces average result. Those outcomes alternate across examined collection of 70 data sets. It would take too much place to present here results on each of analysed data set, therefore, we summarize results on the overall collection. Proportionally, outcome of integration was good in 37%, moderate in 42% and average in 21% of the data sets. On genomic data (30 data sets) percentages were 26%, 43% and 30% for good, moderate and average, respectively. Higher percentage of least desired outcome evince intrinsic complexity of such data. On the other hand, synthetic data collection, created with particular aim to evaluate clustering algorithms, integration performances were qualified as good, moderate and average in 50%, 25% and 25% cases. In these experiments we also observed examples (3 out of 20) where the results of the integration surpassed all individuals in the ensemble. This outcome also occasionally happened on other data sets, but averaging across 100 experimental realization of integration reported only the cases where this effect was considerable.



(a) **(b)**

Figure 4.8: Max, min and average adjusted rand index scores from ensemble of (a) 10 (b) 50 kernel k-means clusterings and corresponding NMF and CONS results as a function of ensemble similarity on cDNA-Bredel-2005 date set. NMF and CONS results are denoted with blue and red lines, respectively.

4.5 Discussion

The study presented here is the largest and most comprehensive benchmarking of integrative clustering algorithms in terms of the number of data sets and the number of experimental repetitions on each data set. The study attempts to set some guidelines for a potential user of a integrative clustering algorithms. Other studies included smaller number of data sets and repetitions on each data set [52] or even report on results obtained on a just one input ensemble per data set [51]. Beside commonly used external validations that measure how clustered data align with actual labels, we also evaluated algorithms without reference to the external information. Since there is no general measure of cluster validation we explored two different internal indexes: one suitable to evaluate spherical clusters other more general, that highly scores also elongated, path-based or arbitrarily shaped clusters.

In the context of diverse collection of data we did not detect absolute winner among integrative clustering, but useful conclusions can be made. Our analysis revealed advantage of NMF and CONS integrative clustering methods compared to other methods due to their ability to manage variety of real and artificial datasets on a range of validation criteria. DICLENS is also among better ranked

algorithms, especially when clusters were validated by Isolation index. Its additional advantage is possibility to automatically detect the number of clusters, however its complexity is the highest and thus we limited ensemble size at 10 individual clusterings in our benchmarking experiments. MCLA is fast algorithm, but ranks from the third to fifth place and NMF and CONS significantly outperformed it (Fig. 4.5). OKKC performed with large variance. As an early integration method, OKKC can weight differently kernel matrices, and thus have more flexible strategy in search for final clustering, but experiments revealed instability of its underling procedure based on optimization and usage of k-means for final partition. HGPA algorithm performed worst regardless on the validity measure used. Our experiments indicate that it is not appropriate algorithm for assembling smaller number of baseline clusterings all partitioned into the same number of target clusters. Its performance improves when diversity is achieved by overproducing clusters, i.e. setting the number of clusters k in input clusterings to the smaller value than targeted number of clusters in the final partitioning of data, or by varying k across input clusterings.

Further analysis on cancer genomics data set collection provided us with some insights on interplay between diversity, number of partitions and the final result. As revealed by obtained experimental results for examined exclusive type of clustering, the best outcome of the integration was to be at the level of the best in the ensemble, while the worst was at average level. The quality of the final results depends on the overall quality of assembling components and furthermore on synergy between them. Nevertheless, prominent ensemble approaches (NMF and CONS) in the worst case scenario are at least good as average of their assembling components and they still could be used to deal with dilemmas and uncertainties in clustering. To improve the integrative approaches for complex genomic data, different ways of constructing ensemble and relevant data fusions should be further explored. In the next two chapters perspectives of data fusion are examined .

Chapter 5

NMF for integrative discovery of functionally related genes

A common task in molecular biology is gene function prediction. We can exploit currently available functional annotations in model organisms in combination with various source of experimental data to infer functions of yet uncharacterized genes. A popular approach for this task is gene clustering [80]. Clustering infers groups of similarly-profiled genes. The experimental data that characterizes genes is considered for the assessment of gene similarity and the function of uncharacterized genes is inferred from the prevailing function of the genes in the cluster. This "guilt by association" principle assumes that gene clusters are also functionally enriched, that is, genes with similar functions will cluster together, making the clusters coherent in terms of functions carried out by genes in the cluster.

Large-scale molecular biology experiments may provide the data for profiling thousands of genes. These profiles may include condition- or development stage-specific gene expressions, mutant-based phenotypes such as growth rates or measurements of fitness, and gene interactions. Profiles that stem from different types of experiments may result in gene clusters of different coherence and hence different utility for gene function prediction. An open question is how to integrate the results of clustering coming from different types of gene profiles to increase the quality of clusters with respect to enrichment of their associated gene functions.

In bioinformatics, integrative approaches are motivated by the desired improvement of robustness, stability and accuracy. Troyanskaya *et al.* introduced a Bayesian integrative framework [81], [82], [83] that examines information from various data sources. Each data source provides information to independently

estimate the likelihood that a pair of genes is functionally related. These likelihoods are then merged across data sources via the Bayesian approach. The structure of the Bayesian network and conditional probability tables are often obtained from domain experts or inferred from Gene Ontology (GO) [20]. A related, but methodologically different unsupervised approach to data integration was proposed by Tanay *et al.* [84], where biclustering of genes and their characteristics led to identification of groups of genes with correlated behavior across diverse data sources.

The approach proposed in this thesis is motivated by consensus clustering [49], a method that originally incorporates resampling to yield diverse data sets of which clustering is a subject to consensus analysis to find groups of genes that consistently co-cluster across data samples. Consensus clustering increases the stability of discovered clusters. Instead of resampling employed in consensus clustering, we propose to examine gene clusters that are developed from different data sources and different similarity measures.

In our study, gene clusters are inferred from gene networks [85] [86] [87], where these are constructed from gene profile data applying some profile similarity measure. We considered different data sources and also diversify input data by considering various estimates of gene profile similarity. For clustering, we use a state-of-the-art network-based algorithm SPICi (Speed and Performance In Clustering) [88] and two well-known Markov Cluster [89] and Affinity Propagation [90] algorithms. Different clustering algorithms provided us an opportunity to study their effects on quality of data fusion.

5.1 Data sources

We considered three different data sets on budding yeast (*Saccharomyces cerevisiae*) that include a collection of gene expressions measured at 36 different time points of the metabolic cycle [91] (YMC), gene interaction data from SGA experiments [92], and gene expression data sets from the Saccharomyces Genome Database - Expression Connection (SGD) [93]. SGA interaction data profiles 3,475 query genes by recording a fitness of a double mutant, where each of the query genes was knocked-out together with another gene chosen from the set of 1,712 genes. In gene expression data from SGD we have merged various SGD data subsets to derive profiles of genes whose expression was observed under 740 different conditions. The selected data collections include different sets of genes; we focused on the subset of 1,799 genes that were present in all three data sources.

5.2 Inference of gene networks

We inferred gene networks from gene profile similarities and considered three alternative measures: mutual information, Pearson correlation coefficient and Euclidean distance. Each inferred network is an undirected weighted graph $G = (V, E, w)$, where V is the set of nodes (genes), $E \subseteq V \mathrm{x} V$ is the set of edges and w are edge weights that refer to estimated similarity. In the case of mutual information and Pearson correlation, two nodes are connected if the profile similarity between their corresponding genes is above the 99^{th} percentile of similarities from ten thousand arbitrarily chosen gene pairs from randomly perturbed data (c.f., [85]). For Euclidean distance, significant edge weights are those below 25^{th} percentile of estimated null-hypothesis distribution. Initial threshold that selects edges below the 1^{st} percentile was too restrictive and would result in a loss of more than half of networks nodes that became singletons after thresholding.

After the thresholding described above the resulting gene networks still include about half a million edges and are too dense for identification of groups by graph-based clustering. Hence, we have additionally removed the edges by retaining at most 100 highest-scored edges for each gene. The choice of this threshold was inspired from results of the studies of yeast's co-expression networks in [94], [95] which exhibit small-world and scale-free typologies with high modularity. The degrees of our resulting metabolic, expression and SGA networks along with the other main properties of inferred graphs are reported in the Table 5.1. Analysis was carried out with the Network Analyzer [96] plug-in for the Cytoscape [97]. These properties are similar to those of the co-expression networks from [95] where clustering coefficient was 0.2 and diameter was 3, and are similar to properties of the networks from [94], where the average node degree was 73.4.

Table 5.1: Statistical Properties of Inferred Networks

Data Set	**Similarity Score**	Number of Nodes	Average Degree	Clustering Coefficient	Network Diameter
YMC	Mutual Inf.	1798	42.32	0.23	6
	Pearson	1797	41.38	0.32	7
	Euclidean	1788	62.74	0.53	14
SGA	Mutual Inf.	1799	76.85	0.07	3
	Pearson	1799	73.36	0.09	3
	Euclidean	1799	67.19	0.17	5
SGD	Mutual Inf.	1799	35.20	0.21	6
	Pearson	1797	31.36	0.24	7
	Euclidean	1428	33.82	0.33	14

5.3 Clustering algorithms

The SPICi [88] network clustering algorithm searches for highly connected regions in the network and uses a greedy heuristic approach. It calculates the density of sub-network $S \subset G$ as the sum of the weights of all edges in S divided by the total number of possible edges that would be present in a complete sub-graph. Another measure used in SPICi is node support provided by a sub-network S, which is defined as the sum of the weights of edges that are incident to nodes in S. The algorithm starts with nodes of the highest-weighted edge and grows the cluster based on two parameters: Ts - the support threshold and Td - the density threshold. The number of clusters is determined by the algorithm. After clustering, some nodes remain as singleton clusters due to their relatively low similarity with adjacent nodes and they are discarded at the end of the process. Our networks were clustered with parameter Ts set to 0.5 and Td adapted to the network properties. The starting value of Td was set to 0.5 and was decreased until coverage, expressed as the ratio between the number of genes included in the clusters and the total number of genes, reached at least 50 % of genes.

The Markov Clustering (MCL) [89] algorithm uses random walks and assumes that longer network paths are more likely to occur for a pair of associated nodes. The algorithm starts with an adjacency matrix that represents a weighted graph, where the diagonal elements are added to include self-loops. The matrix is transformed to a stochastic transition matrix where each column sums to one. After this, expansion and inflation operators are applied in iterative steps. Expansion corresponds to the power of a matrix and provides higher step transition probabilities. The inflation operator takes entry-wise powers with coefficient r and it is followed by re-scaling to keep the matrix stochastic. This operator emphasizes strong connections and further weakens already weak ones. Inflation parameter r affects clustering granularity. In our experiments, we start clustering with r set to 2.0. If the algorithm produced oversized clusters with more than 300 genes, inflation parameter r was increased. For SGA/Mutual information, SGA/Euclidean and YMC/Euclidean networks this parameter was set to 2.0, 2.5 and 4.0, respectively. For all others networks, $r = 2.2$ fulfilled this condition and provided good quality and coverage of clusters. In the initialization step, self-loops were assigned to the graph with weights that equal the maximum weight of incident edges for each node [98]. Compared to the case where the self-loop is left at zero or equal to the sum of incident weights, this setting produced better results in terms of the higher gene function enrichment scores.

The third algorithm, Affinity Propagation [90] (AP), searches for representative nodes (so-called exemplars) that provide seeds for clusters. Seeds are chosen to maximize within-cluster similarities. Nodes exchange messages on availability and responsibility. Responsibility $r(i, k)$ is sent from non-representative nodes to exemplars and inform on the suitability of exemplar k for node i, considering

other potential exemplars. Availability $a(i,k)$ is sent from exemplar k to data point i to inform it on how appropriate it would be for point i to choose k as its exemplar. Messages trigger actions on choice of cluster membership, and are exchanged until reaching convergence. The number of exemplars (clusters) emerges through the use of a clustering algorithm.

5.4 Integration by nonnegative matrix factorization

The result of network clustering from different data set/similarity measure combinations can be merged into a matrix of cluster memberships R. Cluster memberships by SPICi, AP and MCL are all crisp and the values in membership matrix are either 1 or 0, indicating whether a gene was assigned to a specific cluster. After merging results of baseline clusterings, matrix R is factorised by the procedure explained in Chapter 3, and final clusters are reconstructed from matrix H in case of overlapping clustering, and from both matrices, W and H, for exclusive types of clusters.

Factorization of the input matrix R is iterative and runs for 500 iterations. This is also the number of iterations that is required for to reach a stable results in terms of a clustering structure in number of clusters and involved genes.

5.5 Cluster scoring

Any useful clustering should infer gene groups that are coherent in terms of gene function or any other observed gene properties. To test this aspect of the method, we use gene annotations from Gene Ontology [20] (GO) and focus on its 92 yeast slim terms that represent the major branches of the GO. We assume that the quality of the cluster is associated with the enrichment of a subset of slim terms in the annotations of genes from the clusters. Term enrichment, expressed through a p-value, was computed with a hypergeometric test that assesses the probability that, for a particular GO term, the abundance of term-annotated genes in the cluster is not the result of chance. Intuitively, the clusters with no enriched terms are not useful for function prediction and hence are of poor quality. In general, good clusters may have several slim terms that are enriched. For instance, Fig. 5.1 presents sub-network of one of the discovered clusters in the network inferred by Euclidean metric on YMC data. The cluster is enriched with several slim terms (*mitochondrion*, *structural molecule activity*, *cellular biosynthesis* and *transferease activity*). Enriched slim terms are denoted with different colours.

Another example of a cluster discovered in network that was inferred on SGD data by using Euclidean network is presented in Fig. 5.2. This large cluster is

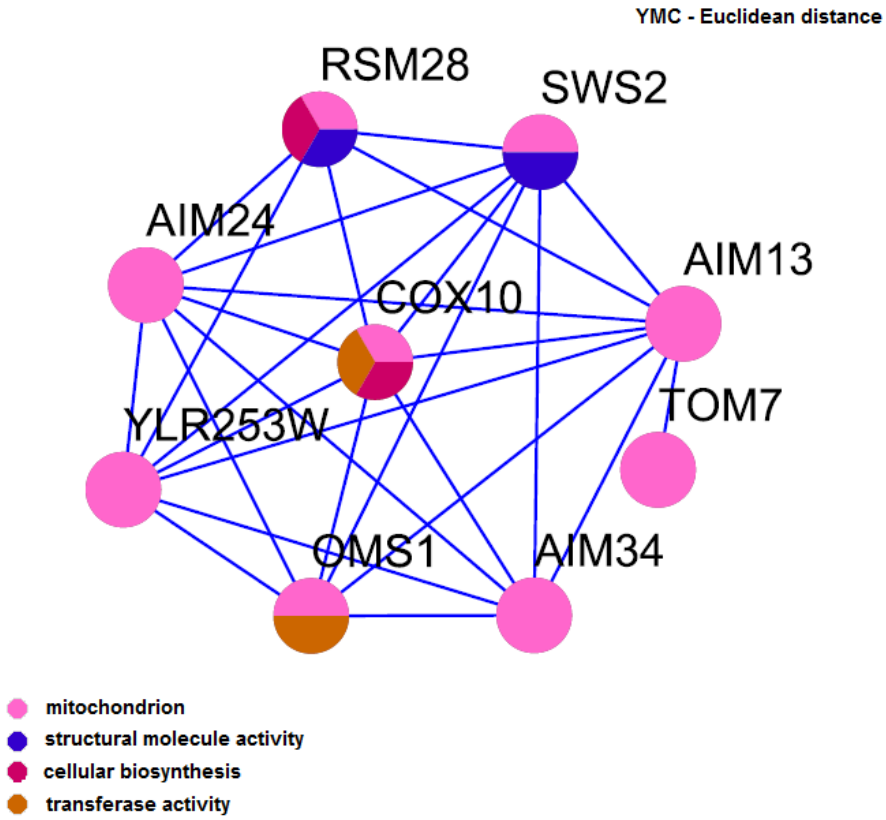


Figure 5.1: A cluster discovered in the network inferred by Euclidean metric on YMC data. Enriched slim terms: mitochondrion, structural molecule activity, cellular biosynthesis and transferase activity are denoted with different colours.

enriched with *ribosome*, *structural molecule activity*, *protein biosynthesis*, *cytoplasm* and *RNA binding*. Enriched slim terms are denoted with different colours. Two genes without colour (white nodes) are still uncategorised. Those genes are placed according to their connections to other genes. Based on 'guilty by association' principle, those genes may also have similar functional roles as other genes from the cluster.

Discovered clusters in Fig. 5.1 and 5.2 were coloured by GOlorize - a Cytoscape plug-in for network visualization [99]. The tool places the nodes based on both their connection structure and GO terms structure.

To account for more functional enrichments we score the clusters by averaging $-log$(enrichment p-value) of the three most-enriched slim terms. Improvements in clustering algorithm should yield clusters with increased proportion of genes that share common function, and thus exhibit higher function enrichment scores [100].

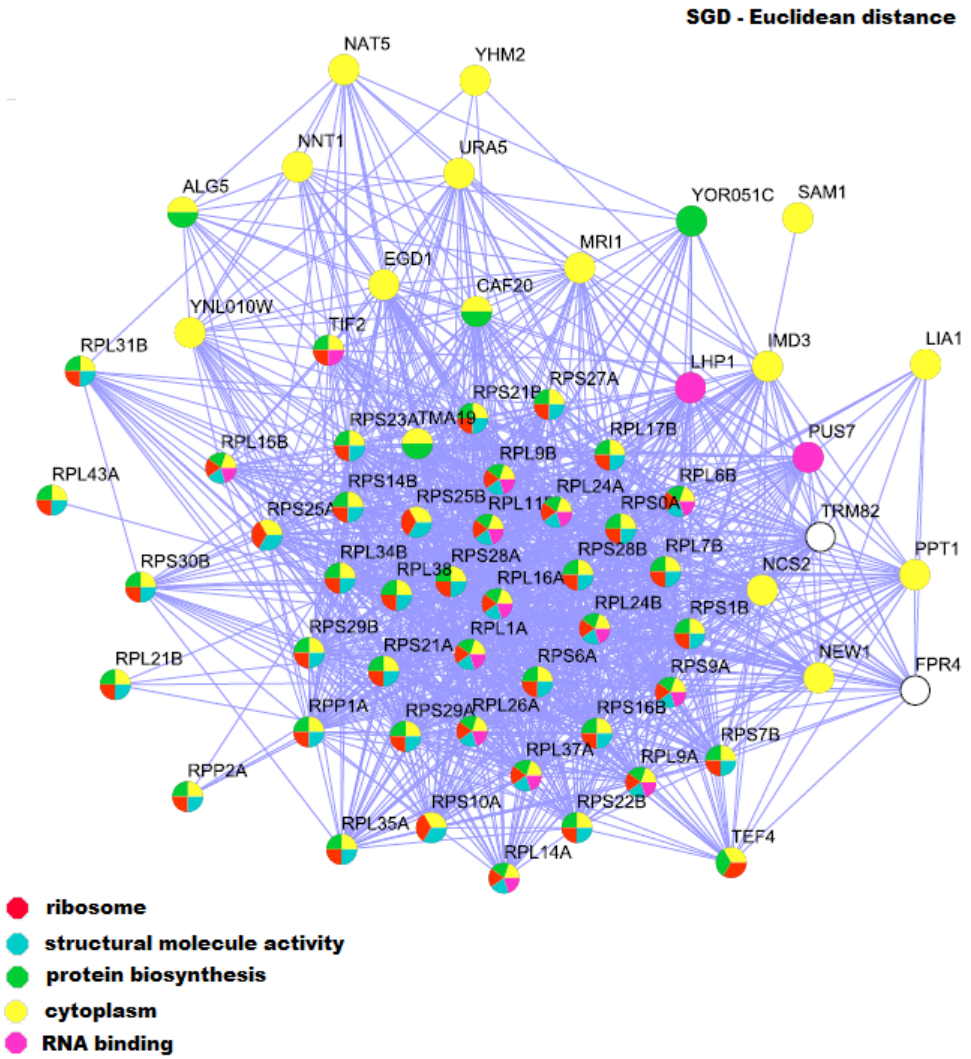


Figure 5.2: A cluster discovered in the network inferred by Euclidean metric on SGD data. Enriched slim terms: ribosome, structural molecule activity, protein biosynthesis, cytoplasm and RNA binding

5.6 Results

This section provides in-depth view on different integration scenarios. The properties of individual clustering used in integrations are outlined in Table 5.2 and include number of clusters and coverage - the ratio between clustered and total number of genes. We first describe experiments with this set of input clusterings. Later, we evaluate method on larger set created by altering the parameters that affect clustering properties. In the experiments we have varied the factorization

rank k according to the average number of clusters inferred by individual clusterings that participate in the integration (bottom row of Table 5.2). We then used $k \in \{150, 200, 250, 300, 350\}$ for SPICi and $k \in \{100, 150, 200, 250, 300\}$ for the other two methods. In this way we could test the effectiveness of representing new clusters by virtue of merging, splitting and combining input clusters.

Table 5.2: Properties of Individual Network-Based Clusterings (Inputs to Integration)

Data Set	Similarity Score	SPICi		MCL		AP	
		Clusters	Coverage	Clusters	Coverage	Clusters	Coverage
YMC	Pearson	221	0.77	197	0.94	185	0.99
	Mutual Inf.	183	0.70	214	0.92	252	0.99
	Euclidean	141	0.81	179	0.95	136	0.99
SGA	Pearson	385	0.76	155	0.73	245	1.00
	Mutual Inf.	307	0.86	174	0.91	280	1.00
	Euclidean	285	0.89	118	0.76	162	1.00
SGD	Pearson	256	0.84	232	0.84	195	0.99
	Mutual Inf.	279	0.73	205	0.85	213	1.00
	Euclidean	170	0.61	176	0.76	175	0.78
Average		247	0.77	183	0.85	205	0.97

5.6.1 Partial integration across data sets or across different similarity scores

We integrated either a single input data set where the clustering was inferred from similarity networks obtained with application of three different similarity measures, or integrated three different data sets where a single similarity measure was considered. Experimental results of these six integration scenarios are summarized in Fig. 5.3 and Fig. 5.4. The bar charts present the average enrichment scores of SPICi clusters (before the integration), and the line graphs present the enrichment scores after the NMF integration with both overlapping and exclusive clusters at five granularity levels (k). Each panel shows result for specific integration scenario:

- YMC data x 3 measures, Fig. 5.3(a)
- SGA data x 3 measures, Fig. 5.3(b)
- SGD data x 3 measures, Fig. 5.3(c)
- 3 data sets x Pearson Correlation, Fig. 5.4(a)

- 3 data sets x Mutual Information, Fig. 5.4(b)
- 3 data sets x Euclidean distance, Fig. 5.4(c)

Corresponding coverages of integrative clusterings can be followed in Fig. 5.5. From the graphs we can observe that reducing the number of clusters after integrations k reduces the coverage of genes.

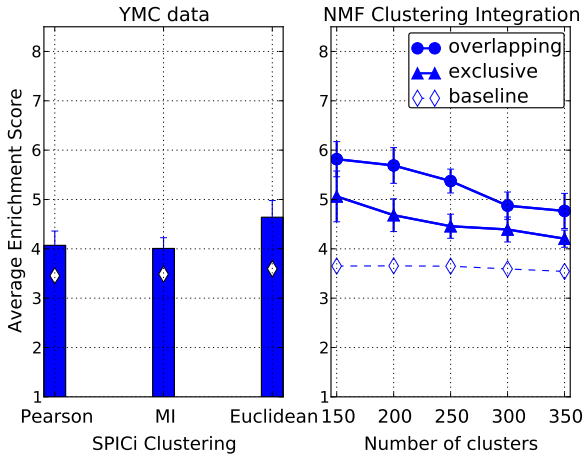
In all cases the NMF integration results in increased enrichment scores and with this improved quality of clusters. The enrichment scores are compared to the baseline scores (diamond symbol on bars and dashed lines) inferred from clustering with random assignment of genes to the clusters. The graphs provide baseline scores for clustering before integration (bar charts) and for overlapping NMF clustering (line charts); the baseline scores for exclusive clustering were slightly lower and are not shown.

The results demonstrate that integration improves enrichment, as we always observe higher scores for the clusterings after integration. The results also suggest that the efficiency of integrative clustering can be boosted not only by considering the integration of different sources of data, but also by considering different measures of similarity. Comparison with baseline enrichment derived from clustering with the same structure of clusters but arbitrary association of gene cluster membership demonstrates that improvement from initial clustering is truly due to integration and appropriate assignment of genes to the clusters, and is not obtained just by changing the size and number of clusters.

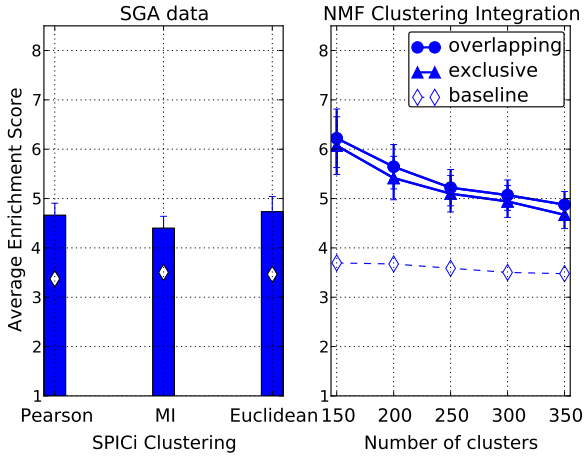
5.6.2 Integration of complete set of input clusterings

In the next experiment we tested the effectiveness of integrating the entire set of nine clusterings where all data sets and all similarity measures were involved. This integration (see Fig. 5.6(a)) improves the results over previous models of integration. In graph on the left we report on average enrichment scores for clusterings that participate in the integration, and the right part presents average enrichment scores after NMF integration produced at five different granularity levels. Higher enrichment scores indicate better functional coherence of clusters. The enrichment scores after integration are also consistently above the baseline obtained by evaluating random clusterings of the the same clustering structure.

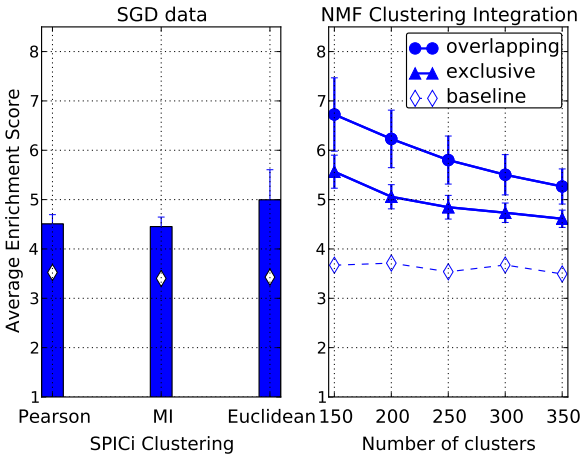
NMF grouped genes into clusters with an average enrichment score from 6.15 to 8.11 for overlapping clustering, and from 4.91 to 6.19 for exclusive clustering. That is significantly higher than the coherence in original clusters since the best clustering that was involved in this integration (SGD data set, Euclidean measure) has an enrichment score of 4.99. Integrated clusters have higher gene function coherence than clusters that served as an input to the integration.



(a) Integration scenario: YMC data x 3 measures

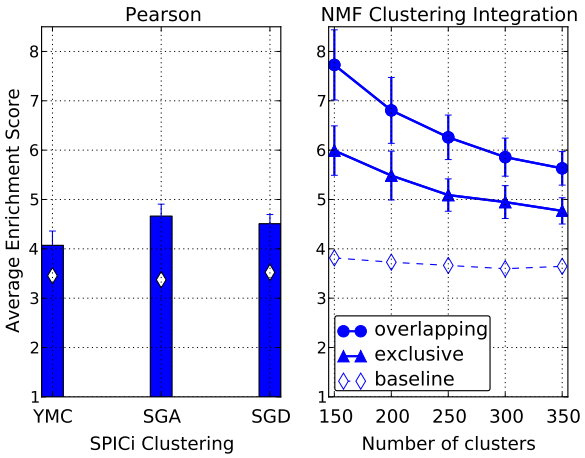


(b) Integration scenario: SGA data x 3 measures

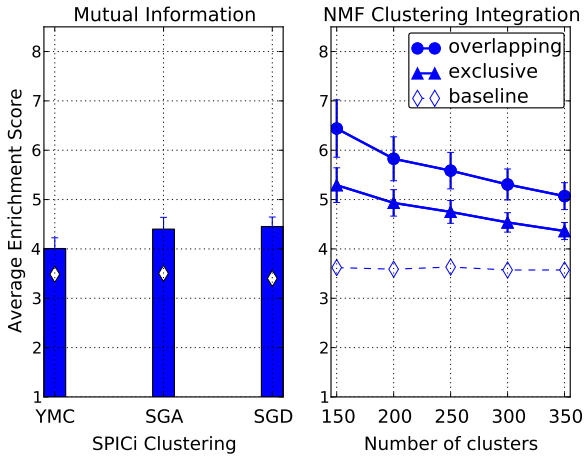


(c) Integration scenario: SGD data x 3 measures

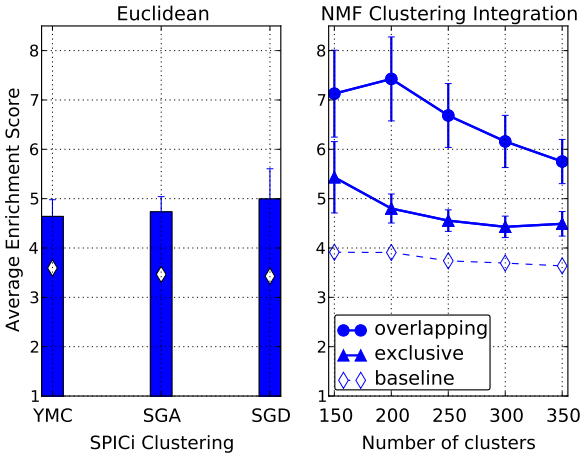
Figure 5.3: Comparison of clustering results before and after the integration.



(a) Integration scenario: 3 data sets x Pearson Correlation



(b) Integration scenario: 3 data sets x Mutual Information



(c) Integration scenario: 3 data sets x Euclidean distance

Figure 5.4: Comparison of clustering results before and after the integration.

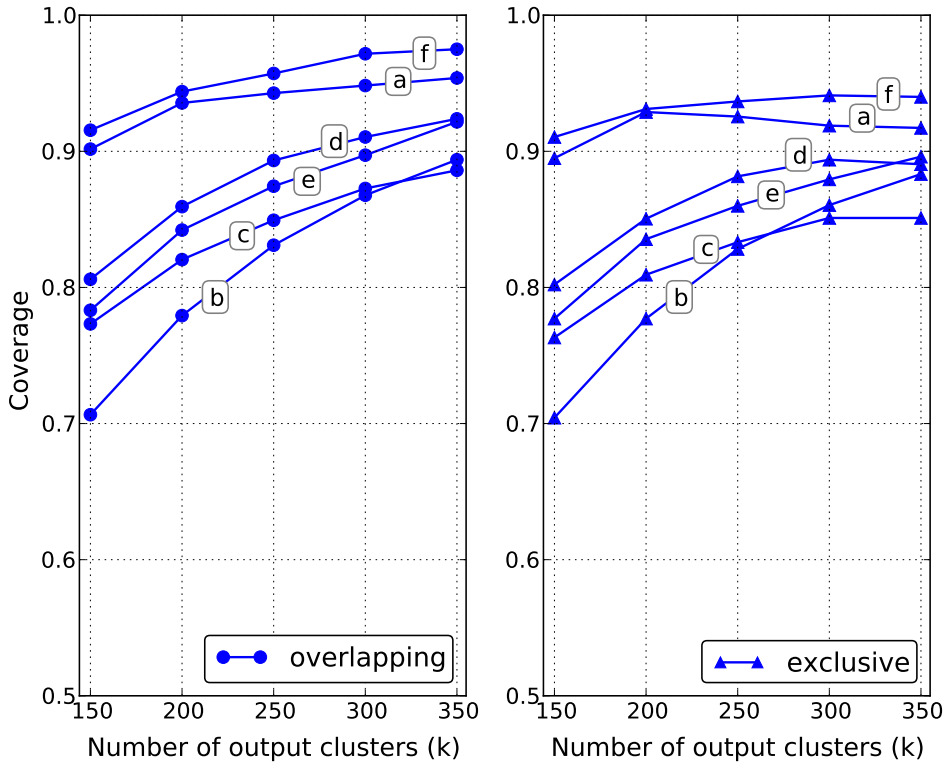


Figure 5.5: Coverage of genes as a function of the number of output clusters k. The figure reports on the coverage of overlapping (left) and exclusive NMF clusters (right) from six experiments presented in Fig. 5.4. Letters on the lines in the graph (from a to f) refer to panels with different integrations scenarios from Fig. 5.4.

We further tested the behavior of the proposed data fusion with two other clustering algorithms, MCL and AP. Again, clustering was carried out on networks inferred from all three data sets, where we used each of the three similarity measures. The results (Fig. 5.6b and 5.6c) demonstrate better performance of overlapping representative clusters compared to all individual clusterings for both MCL an AP. In the case of MCL, the quality of exclusive representative clusters outperforms all individual clusterings when k is set to 100 and 150 and it is at the level of the best used in integration when k is 200. When we increase granularity (250 and 300 clusters), the integrative approach performs slightly worse, with enrichment scores that are still higher than in seven out of nine individual clusterings. In the case of AP, our method is able to successfully transform input clusters in 100 and 150 exclusive representative clusters. If we addition-

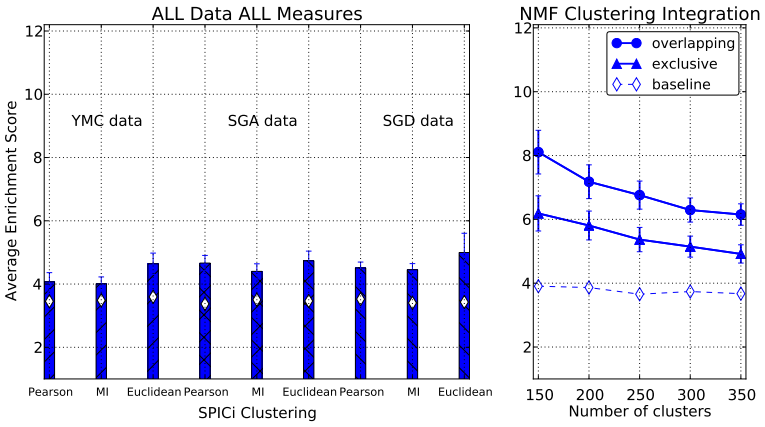
ally increase granularity when creating representative clusters, the quality of the resulting system declines.

5.6.3 Choice of the number of clusters with respect to its effect on average accuracy and gene coverage

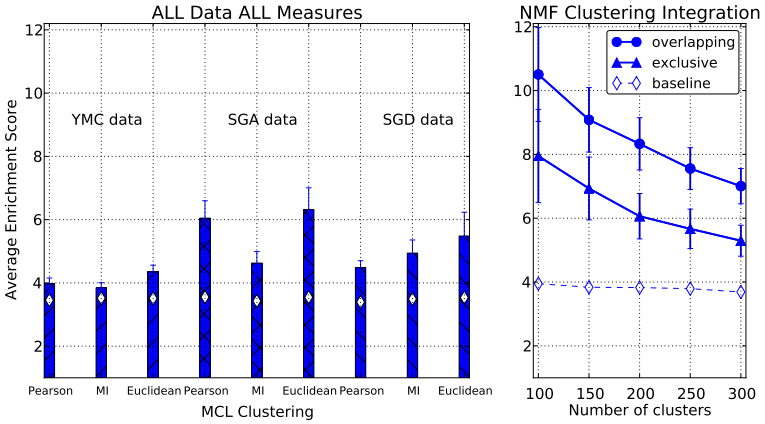
Both average enrichment and gene coverage depend on the choice of the number of output clusters k. Results suggest that both scores improve after integration. For instance, the average number of input SPICi clusters was 247 with gene coverage of 0.77 (Table 5.2, bottom row). At similar number of clusters ($k = 250$), the integration — especially the one with overlapping clusters — improves the average enrichment score (Fig. 5.3 and Fig. 5.4) but has also higher coverage (Fig. 5.5).

To further study this two-fold benefit of integration, and isolate its dependency on number of clusters, we altered the parameters of our network clustering methods that provide for initial clustering. Our aim was to infer a cluster sets with specific number of input clusters, and then output the same number of clusters after the integration. SPICi ($k = 150$) and MCL ($k = 100$) clustering were considered, as AP clustering is parameter-free. Shrinking the number of clusters when compared to our previous experiments (Table 5.2) slightly improved enrichment for MCL clusters, but had a mixed effect on SPICi-based clusters. Average enrichment score in a set of SPICi-inferred clusters was 4.56 with best individual clustering scoring 5.08, at 0.95 coverage. Integration increased both the coverage to 0.97 and average enrichment score to 5.56 for exclusive, and to coverage of 0.99 and enrichment of 7.93 for overlapping clustering. Average score in a set of clusters by MCL was 5.43 with best individual clustering scoring 6.77 at 0.96 coverage, while NMF again increased the coverage and enrichment to 0.99 and 7.97 for exclusive and to 1.00 and 9.65 for overlapping clustering, respectively. This set of experiments further confirms the utility of integration by increasing both average enrichment and coverage. We have obtained qualitatively similar results with cluster reduction by pruning of the smallest clusters in the input clusterings (results not presented for brevity).

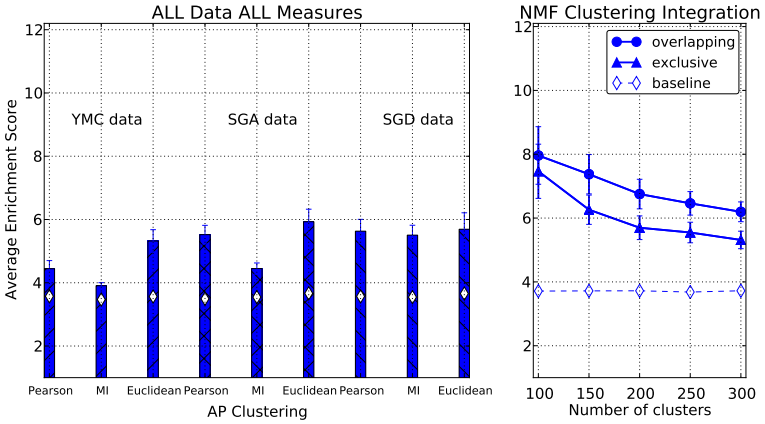
The number of clusters k after the integration is a user-specified parameter. When k is small, the effect of integration is stronger, while for higher values of k the initial clusters may be split to smaller ones. The choice of parameter k involves considering the trade-off between enrichment scores and coverage, and may depend on the goals of particular application. For an appropriate starting choice we recommend setting the number of clusters to the average number of clusters in the input set of clusterings. Our experiments suggest that under such setup the clustering integration already has a positive effect by increasing both enrichment scores and coverage.



(a) Integration of SPICi clusterings



(b) Integration of MCL clusterings



(c) Integration of AP clusterings

Figure 5.6: NMF integration complete set of input clusterings (3 data sets x 3 measures).

5.6.4 Further insight into the effects of cluster integration

To further demonstrate the inner workings of the proposed approach, we provide an illustration obtained from our experiment with integration of nine clusterings (3 data sets x 3 similarity measures). Fig. 5.7 shows part of the input matrix R considered by NMF. Matrix columns correspond to genes and rows to clusters. Information on the data source and corresponding similarity scoring is provided in the last column of the matrix. In the figure, we provide details on two initial clusters c_1 and c_{21} (the first and the last row) that are the best among the 21 presented and compare them with the output clusters after NMF transformation. For each of the clusters we have analyzed we report on the most enriched GO terms. Since only a subset of genes is shown in the figure, we print in black the cluster memberships that comprise only the genes present in the displayed matrix, and in gray those that also comprise some genes outside the displayed matrix. Notice how NMF reorganizes clusters. Based on the supported evidence, NMF prunes initial clusters and creates functionally more consistent groups. For 33 genes in Fig. 5.7 assigned to 21 input clusters, NMF identified two clusters that are related to this particular set of genes. Genes CAT2, TCB3, YML131W, YNR014W, HXK2, MTO1, SIS2 and YIR024C were excluded from these clusters due to obvious lack of supporting evidence. CAT2 shares label peroxisome - prevailing function assigned to $c1$, but except that cluster none of the other input clusters uphold its connection to genes that remained clustered together after NMF. We have further examined other clusters that included CAT2. Interestingly, this gene was assigned to another group also enriched in peroxisome, but additionally associated with cellular amino acid and derivative metabolic process. Through other NMF clusters, YML131W was additionally associated with membrane, HXK2 and MTO1 with cytoplasm and mitochondrion, SIS2 with enzyme regulator activity and YIR024C with mitochondrion. TCB3 was not assign to any NMF cluster due to small support, only YNR014W was in cluster were did not contribute to the enrichment score. Output clusters with assigned functional labels indicate that not only is the NMF approach able to identify representatives among input clusters, but also succeeds in further improving them.

5.6.5 On initialization of matrix factorization procedure

Although there is no guarantee that NMF with multiplicative updates converges to global optimum, obtained solutions proved useful and improved clustering results. Through the use of deterministic initialization by NNDSVD [56], our procedure always converges to the same solution. Alternatively, we could use a random initialization of matrices W and H. To examine the differences with deterministic initialization in terms of quality of resulting clusters, we ran 50 experiments with random initialization for 6 integration scenarios from Fig. 5.3

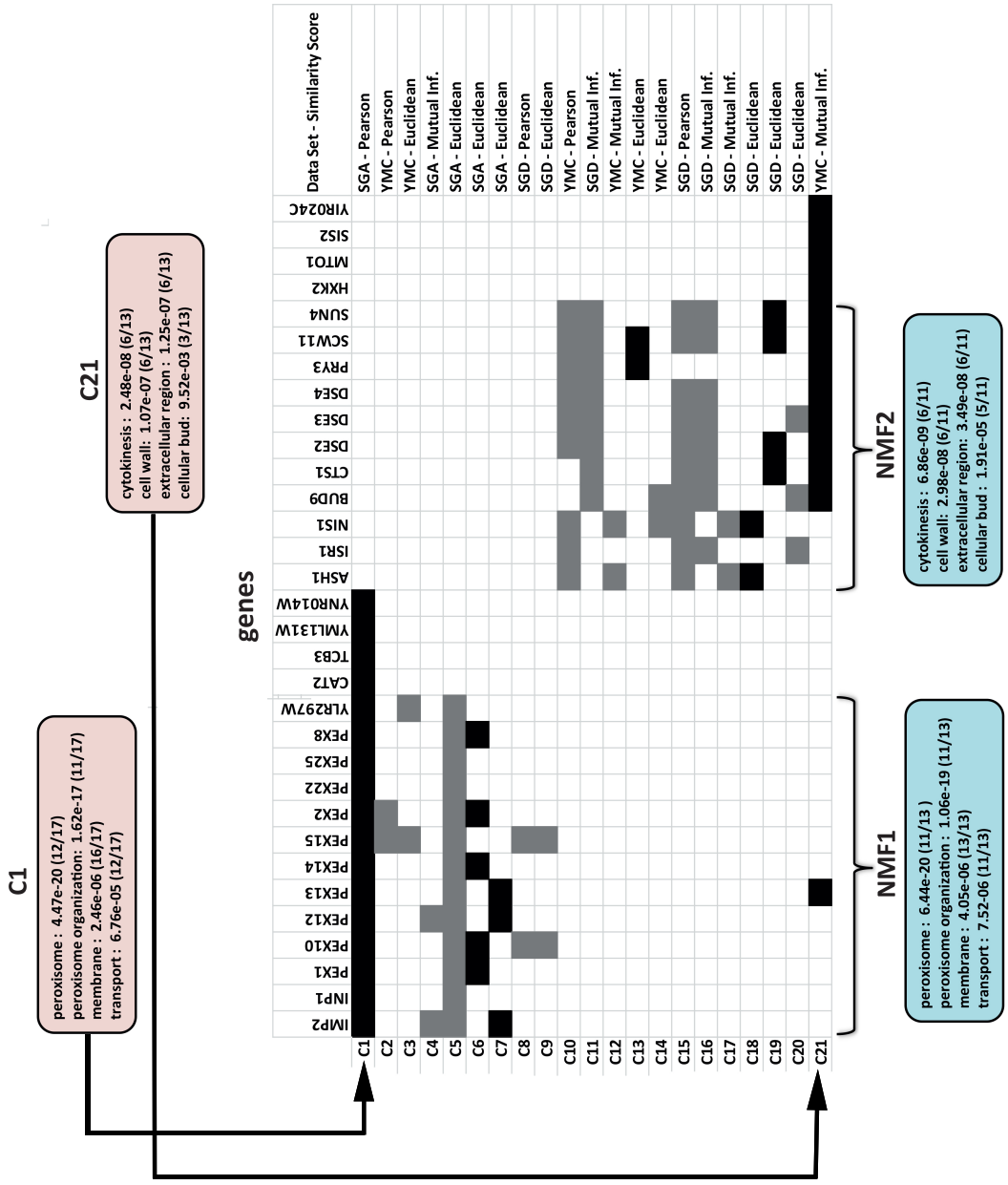


Figure 5.7: Integration of information through NMF discovers more meaningful clusters. The figure shows a fragment of integrated cluster membership matrix. The black colour indicates that the fragment of matrix encompasses all members of the cluster, and the grey colour indicates that cluster includes other genes besides those presented. To compare the results we assigned corresponding enriched functional terms to two input clusters (the best in this example) and to output clusters (obtained through NMF framework). Improved enrichment values demonstrate the benefits of the integrative approach.

and Fig. 5.4. Results (Fig. 5.8) indicate that both initialization techniques lead to data integration of similar quality.

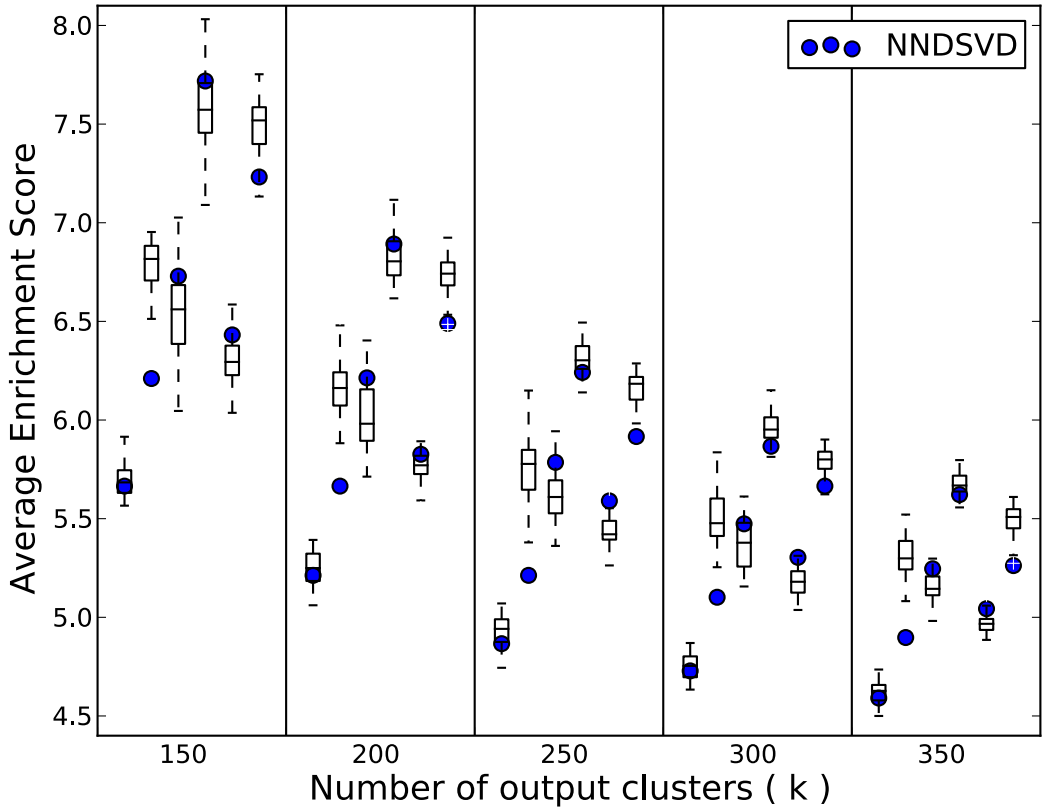


Figure 5.8: Comparison of matrix factorization initialization by NNDSVD and random initialization across six different integration scenarios from Fig. 5.3 and 5.4 and using five different factorization ranks (k). Initialization by NNDSVD is deterministic and using it our data integration procedure converges to a unique solution (blue dots). Results of 50 runs of data integration by random initialization are summarized with box-plots.

In some cases random initialization may yield better results and hint at potential utility of assembling of randomly-initialized models. However, considering substantially increased computational requirements of such procedure, we therefore prefer a faster, deterministic, and, as shown in our study, useful initialization by NNDSVD.

5.6.6 On overlapping vs. non-overlapping cluster integration

Our proposed integrative method consistently performs better in terms of average enrichment scores when inferring overlapping clusters. This was in part expected as gene annotation terms in general overlap in coverage of the genes, that is, a particular gene may be annotated with more then one term. The problem considered in this paper, that is, finding gene groups with enriched annotations, is therefore biased and benefits from overlapping clustering. We believe that this is with no loss of generality, as many problems from natural sciences deal with objects that are annotated with a set of labels, rather than classified to a single specific class. Being able to infer overlapping clusters should thus be considered a major strength of NMF-based integration. Other studies also indicate that overlapping clustering better address problems in various fields of molecular biology, such as those investigating protein complexes [101], [102] and biological processes [103].

5.7 Comparison with other data integration techniques

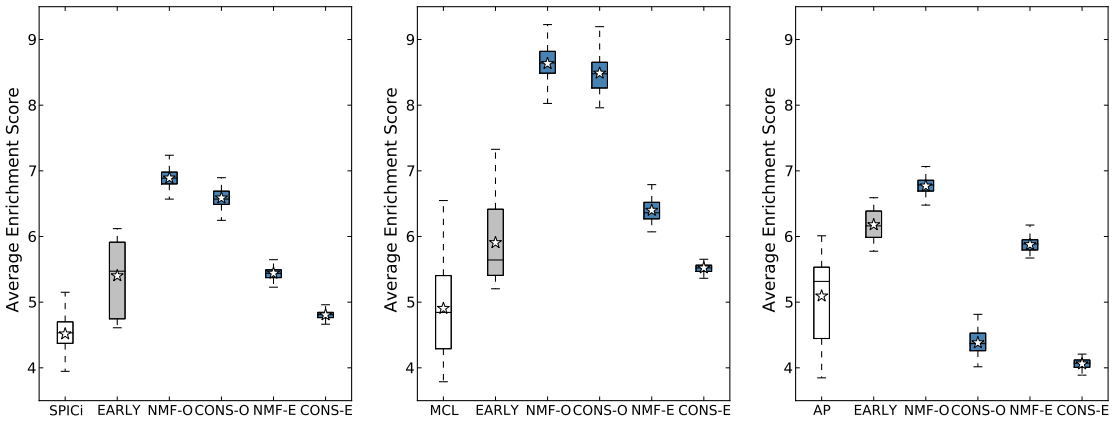
Our proposed approach belongs to the late integration type of ensemble techniques, where aggregation is performed after individual clusterings have already been formed. We have compared our method to well-known late integration approach of consensus clustering [49]. Originally proposed for integration of different clusterings obtained from samples of the same data sets, consensus clustering may also be used when different cluster models stem from different data sets or from different preprocessing steps, as in our case. Consensus clustering integrates cluster memberships into a consensus matrix that can be viewed as a similarity matrix and post-processed through additional methods to obtain final clusters. We used kernel k-means to create exclusive consensus clusters and its soft version to detect overlapping clusters [104]. Soft kernel k-means assigns genes to clusters based on distances to cluster centers. The number of clusters was set to the same level as in the proposed NMF-based integration. Evaluation score for consensus integration in each experiment is averaged across 10 runs due to random initialization of kernel k-means.

A different type of data fusion is an early aggregation, where data is fused before the application of a clustering algorithm by merging gene profiles or by aggregation of similarity matrices [48]. To compare our approach to this technique, we merged gene profiles before clustering and then independently inferred gene similarity networks with all three measures and finally ran individual clustering.

To compare various integration approaches we have first established a collection of different gene networks. We have considered all nine combinations of three data sets and three similarity measures. To additionally diversify the networks, these were pruned so that each node included a maximum of t edges, where $t \in \{80, 85, \ldots 125\}$. Notice that in the previous experiments this parameter was fixed to 100. In this way we have obtained 90 different networks. For the case of early integration, where the data set where first merged, the number of considered networks was 30 (3 similarity measures, 10 choices of t).

Just like in experiments from Fig. 5.6, we have considered three different clustering methods (SPICi, MCL and AP) to obtain the initial clusters from each of the networks. Fig. 5.9 reports on the resulting average enrichment scores for the baseline approach (no data integration), early integration (EARLY), and late integration approaches by overlapping and non-overlapping NMF-based integration (NMF-O and NMF-E) and overlapping and non-overlapping consensus integration (CONS-O and CONS-E). Box plots in the figure summarize the average enrichment scores obtained from each of 90 networks for baseline approaches (no data integration, box plots labeled SPICi, MCL, and AP) and scores from clusters from each of 30 networks for early integration. Late integration techniques were run 50 times, each time on a random sample of 9 networks from our collection of 90 networks. For the late integration approaches, box plots in Fig. 5.9 thus summarize 50 different average enrichment scores. The number of output clusters for each run of late integration methods was set to the average number of clusters in 9 sampled networks.

ANOVA test indicate that significant difference exists among different methods ($p < 10^{-70}$ for all experiments within initial clustering by SPICi, MCL and AP). Post-hoc Tukey test with 99% confidence reveals groups that are significantly different. For integration of clusters proposed by SPICi (Fig. 5.9a) the ranking order is (NMF-O, CONS-O, NMF-E, EARLY, CONS-E, SPICi) with corresponding grouping (A, B, C, C, D, E). Groups that do not share the same letter are significantly different. Thus, in results from Fig. 5.9a, the score distribution for NMF-O is significantly different than those of other methods, while score distributions of NMF-E and EARLY are different to score distributions of the CONS-E and SPICi but are, between themselves, not significantly different. For integration of clusters proposed by MCL (Fig. 5.9b), the ranking is (NMF-O, CONS-O, NMF-E, EARLY, CONS-E, MCL) with corresponding grouping of (A, A, B, C, D, E), and for the integration of AP clusters (Fig. 5.9c) the ranking is (NMF-O, EARLY, NMF-E, AP, CONS-O, CONS-E) with grouping of (A, B, C, D, E, F). Notice that all types of integration surpasses the clustering where no integration took place, except in experiments with AP where both type of CONS lose in performance. For all three types of initial clustering the best results are achieved by overlapping type of NMF integrative clustering. Scores for NMF-E are higher to those for CONS-E. EARLY integration performs comparatively



(a) **(b)** **(c)**

Figure 5.9: Comparison of clustering integration approaches for initial clustering by SPICi (a), MLC (b) and AP (c). Box plots refer to the baseline approach (no integration, the first box plot in each panel), early integration (EARLY), late integration by NMF (NMF-O for overlapping and NMF-E for exclusive clustering), and consensus clustering (CONS-O for overlapping and CONS-E for exclusive clustering). The length of a box is the interquartile range of the enrichment score distribution, the line across the box represents the median, and the mean is denoted with a star symbol.

well, but its score depends on an appropriate choice of similarity measure that, in our experience, is the parameter causing high variance in performance of this approach.

5.8 Discussion

Clustering that infers gene groups from their profiles that can be gathered from any of the current genome-wide experimental techniques is currently one of the most common computational tools in functional genomics. While other more focused and specialized computational approaches exist that could manifest better accuracy by learning from class-labeled data [105], clustering is still prevent method for discovering functionally related genes.

We proposed a technique that develops separate gene clusters and fuses them by means of non-negative matrix factorization. Gene clusters are inferred from gene networks that are built from each of available data sources by applying various estimates of gene profile similarity. Integrative approach allows us to better handle noise and other uncertainties by generalizing across multiple data sources.

We show that proposed integration increases cluster coherence estimated through gene function enrichment [100]. That was confirmed through various integration scenarios, and on three different baseline clustering approaches. Experiments unveiled that overlapping clusterings were particularly useful for discovering enriched gene sets and surpassed exclusive type. We used gene profile data on the budding yeast *S. cerevisiae* to demonstrate that this approach can successfully integrate heterogeneous data sets and yield high-quality clusters that could otherwise not be inferred by simply merging the gene profiles prior to clustering. The clusters discovered through integration are more representative as they include higher proportion of genes that share common function.

Chapter 6

Regularized NMF for integrative discovery of functionally related genes

In the context of machine learning regularization means imposing penalty for complexity to avoid over-fitting, setting prior distributions on model parameters or introducing additional information. Regularizations in nonnegative matrix factorization (NMF) incorporate extra constraints into factorization process. Constraints may affect one of the factor matrices W or H, or both. In bioinformatics, recent studies demonstrated usefulness of regularizing NMF. General framework that can uphold qualities like sparseness and smoothness, or introduce specific relationships between components was presented in the work of Taslaman and Nilsson [106]. The authors applied a wide range of regularization terms on high-dimensional data from gene expression studies that helped them to identify cell type-specific markers. Another interesting example is an orthogonality-regularized nonnegative matrix factorization [107] that imposes orthogonality on the basis vectors. Through that framework authors integrated multiple data sources in modelling protein–RNA interactions and discovered non-overlapping, class-specific RNA binding patterns.

Prominent example among regularized versions of NMF is the Graph Regularized NMF (GNMF) [108]. GNMF extends NMF with affinity graph that encodes a geometrical information. Affinity graph further serves as regularization term. GNMF builds upon manifold assumption, implying that if two data points x_i, x_j are close in the intrinsic geometry of the data, then their representations in the space spanned by new basis vectors should also be close to each other. In other words, GNMF tends to preserve the local manifold structure.

We utilized GNMF to extend our integrative clustering procedure. While ensemble creation and final clusters reconstruction steps described in the Chapter

3 remain the same, the middle step (factorization) changes. Instead of using geometric information, affinity graph in our settings serves to incorporate additional information. We performed extensive experiments on data used in previous chapter, regularized with additional sources that are relevant for functional genomics: protein sequences and interactions.

6.1 Graph regularized ensemble

To incorporate the graph structure into matrix factorization process, its objective function changes. Affinity graph that carries problem-specific prior knowledge is encoded into matrix A and additional derived matrices need to be defined: a diagonal matrix D whose entries are column sums of A, $D_{ii} = \sum_j A_{ij}$, and a matrix L defined as $L = D - W$ that is called graph Laplacian. Regularization term have to be added to the loss function defined by Eq. 3.3. Now, Frobenius norm of the reconstruction error in regularized case becomes:

$$\|R - WH\|_F^2 = \sum_i \sum_j [R_{ij} - (WH)_{ij}]^2 + \lambda Tr(HLH^T) \tag{6.1}$$

where $Tr(\cdot)$ denotes the trace of a matrix, and $\lambda \geq 0$ is the regularization parameter.

Optimization procedure based on iterative updates of the two factor matrices also changes. Objective function defined by Eq. 6.1 is nonincreasing and iteratively converges to a local optimum under following update rules:

$$H \leftarrow H. * ((W^T R + \lambda HA)./(W^T WH + \lambda HD)), \tag{6.2}$$

$$W \leftarrow W. * ((RH^T)./(WHH^T)). \tag{6.3}$$

Compared to the Eq. 3.1 and 3.2, only update of matrix H changed in a way that now includes matrices A, D and the parameter λ, while update rule for W remained the same.

6.2 Data sources for affinity graph

Affinity graph can be directly represented by original data (for instance, interactions between genes or between proteins) or can be constructed from profile or sequence data by applying some similarity measure. Two important data sources for hypothesizing protein functions are protein sequences and protein-protein interactions. We explored both options.

To measure sequence similarities, we rely on BLAST (*Basic Local Alignment Search Tool*) [109] that is the one of the most commonly used sequence analysis tool. By using *blastp* (Protein-protein BLAST) from Biopython package [30] we locally aligned protein sequences (found the most similar regions between two sequences) and quantified similarity between them. Procedure ran in pairwise manner where each pair of protein sequences was forwarded to the *blastp* in a form of a query and subject sequence. As a result BLAST returns the similarity bit score, query coverage, E-value and max identity percentage. For obtained bit score, E-value (*Expectation value*) captures the number of hits we can expect to see by chance given the length of the query sequence and the size of searching database. The lower the E-value, the more significant the score is. Blast enabled us to find possible homologous proteins and thus incorporate homology-based function inference as additional information into our integrative clustering.

Information on the protein–protein interactions was accessed from STRING (*Search Tool for the Retrieval of Interacting Genes/Proteins*) database [110] that provides interaction confidence scoring and comprehensive coverage of proteins. Database combines multiple sources: known experimental interactions, pathway knowledge, automated text-mining on large collection of Medline abstracts and full-text articles, genomic information, results of co-expression analysis, transferred knowledge from other organisms. All protein interactions are weighted by confidence score. Scores from underlying data sources are integrated to compute a final combined score.

From both sources we created affinity graphs having 1799 proteins that correspond to genes used in data fusion framework in Chapter 5. Links in the graphs refer either to protein sequence similarities or their interactions.

6.3 Results

In our experiments GNMF aims to find a new representation space in which two genes are close to each other if their corresponding protein sequences are similar or proteins are connected in PPI graph. To evaluate GNMF approach we used fusion scenarios described in previous chapter, section 5.7. Input diversity comes form a collection of 90 different gene networks, 10 from each of the 9 sources. As suggested by study [108], affinity graph shroud be sparse. To enforce sparsity in affinity matrix with sequence similarities we kept only pairs with $E-value$ ¡ 0.001, that is generally used threshold to eliminate matches of lower quality. Further we kept only 5 closest neighbours. In PPI affinity matrix we applied only the second rule, in a lack of a notion of statistical significance of interaction scores. In both cases, affinity matrices have only up to 0.5% non-zero elements. Sparsity is highly forced to incorporate only strong signals from additional sources. Finally, values in the affinity matrix were normalized into zero-one interval.

The value of the regularization parameter λ balances information coming from the ensemble of nine individual clusterings and additional source used in the experiments, either protein sequences or interactions. We set set to 10, as in the study [108]. Smaller values of λ typically produced results as those without regularization while large values tended to deform clustering by shrinking overall coverage of genes.

Fig. 6.1 summarizes average enrichment scores from experiments where sequence similarities were used to regularize NMF procedure. Here we compare the performances of NMF and GNMF based integrative clusterings, both types overlapping (NMF-O, GNMF-O) and exclusive (NMF-E, GNMF-E). As in previous chapter, the results are presented for ensembles created by three different clustering methods (SPICi, MCL and AP). Since experiments from previous chapter confirmed that overlapping type highly surpasses exclusive, we compare only the scores between pairs of overlapping approaches NMF-O and GNMF-O and then between exclusive NMF-E and GNMF-E. We hypothesized that GNMF, by introducing additional information through regularization, would improve result. To evaluate it we used one-tailed paired t-test. Condition of matched pairs was fulfilled through experimental set-up by ensuring that the same ensemble was passed to the examined pairs of integrative methods.

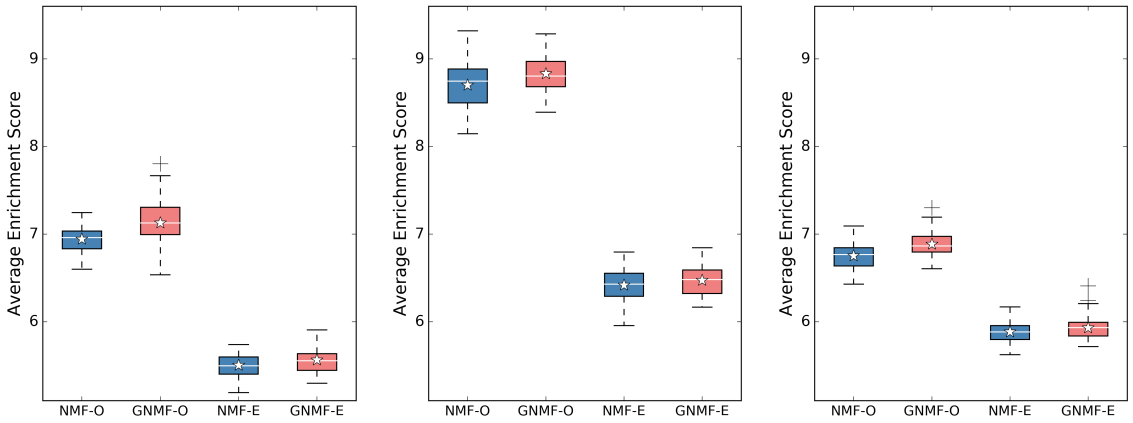


Figure 6.1: Comparison of clustering integration approaches for initial clustering by SPICi (a), MLC (b) and AP (c). Box plots refer to the integration by NMF (NMF-O for overlapping and NMF-E for exclusive clustering), by GNMF clustering (GNMF-O for overlapping and GNMF-E for exclusive clustering). Affinity graph was inferred from protein sequence similarities. The length of a box is the interquartile range of the enrichment score distribution, the line across the box represents the median, and the mean is denoted with a star symbol.

Results suggest that GNMF improves results. Statistical significance for hypothesis tests varied across tested pairs, but overall we can claim that results were significant at 5% cut-off (p-value<0.05). Obtained p-values in testing whether GNMF-O integration performs better than NMF-0 for different baseline clusterings were SPICi: 2.19e-08, MCL: 5.97e-08 and AP: 1.27e-06. Among exclusive types, obtained p-values were SPICi: 4.58e-04, MCL: 2.66e-02 and AP: 4.51e-02. Significance was higher for overlapping type and among baseline clusterings used for ensemble creation the most significant improvement was observed for SPICi.

Results of experiments where protein-protein interactions were used as side information for factorization process (Fig. 6.2) also imply that regularization improves enrichment scores. All comparisons were significant at p-value<0.05. In testing whether GNMF-O integration exceeds NMF-0, the level of significance for different baseline clusterings was SPICi: 4.61e-06, MCL: 3.68e-03, AP: 1.81e-02. Among exclusive types, obtained $p - values$ were SPICi: 4.46e-07, MCL: 1.78-e03 and AP: 2.27-e02. Again, improvement was most significant for experiment where SPICi was used as baseline for creating ensemble.

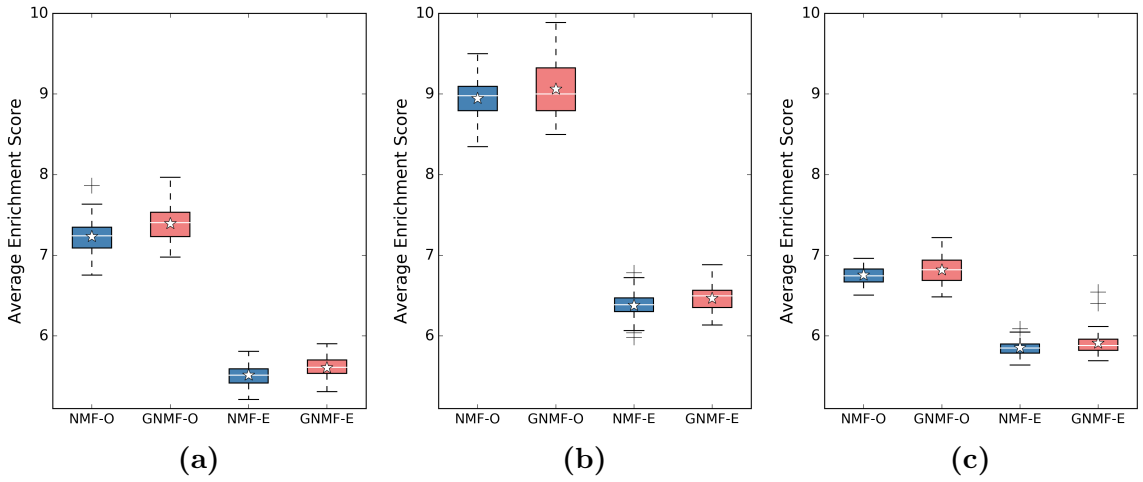


Figure 6.2: Comparison of clustering integration approaches for initial clustering by SPICi (a), MLC (b) and AP (c). Box plots refer to the integration by NMF (NMF-O for overlapping and NMF-E for exclusive clustering), by GNMF clustering (GNMF-O for overlapping and GNMF-E for exclusive clustering). Affinity graph was inferred from protein-protein interactions. The length of a box is the interquartile range of the enrichment score distribution, the line across the box represents the median, and the mean is denoted with a star symbol.

6.4 Discussion

Advancement of NMF procedure by graph-based regularization was beneficial for integrative clustering. It improved the quality of obtained clusters–their functional coherence–for both sources of side information: protein-protein sequence similarities and interactions. Paired t-test confirmed the statistical significance ($p < 0.05$) of the observed differences in mean performance. Only the strongest parts of graphs were included into factorization procedure since protein sequence similarity and interactions are not guarantee for the same functional roles, rather in case of strong connection they serve as a solid hypothesis for functional inference. When used in this way, progress in the results was observed. The next research step is to find a way for combining both sources in factorization.

Our initial experiments with GNMF produced promising results, but moreover opened the avenue of new possibilities. Future work will address problems of setting the parameter for balancing the sources and adding more graphs at once into procedure; evaluate other forms of regularizations and explore the ways to add information on functional annotations from GO in a semi-supervised manner.

Chapter 7

NMF for Stable Assessment of Clusters in Microbiome Samples

High-throughput experiments revolutionize microbial ecology by increasing the speed of research and discoveries related to the diversity of microorganisms and their roles in ecological processes [111]. Fundamental endeavour to understand microbiome and its functions starts with detecting which microbes are present in the samples and continues with comparing different samples and finding similar based on their community compositions. Current studies investigate microbial communities extracted directly from the environment and sequenced with NGS technology. They aim at understanding microorganisms that exists in different environments: human (gut, skin, oral...), water, soil. The question: "Who is there?" comes first in studying microbiome sample. Identifying which microbes are present and quantifying their abundances provides insights into the diversity of the examined sample. Currently prevailing technique in studying microbial diversity is sequencing of marker genes 16S (prokaryotic) or 18S (eukaryotic) rRNA, that are highly conserved between different species and thus suitable for phylogenetic taxonomy. Such approaches are denoted as DNA metabarcoding [112] and characterized as economic way of taxonomic identification that enables monitoring diversity and comparisons of taxonomic compositions among various environmental samples.

Taxonomy relies on clustering analysis i.e. grouping similar species into clusters. Groups of microbial species that show a certain level of similarity represent operational taxonomic units (OTUs). After identifying OTUs in a multiple samples under the analysis, the next step includes between-sample comparisons based on some distance measure, termed as beta diversity analysis and then

again applying clustering to identify communities among samples. But clustering brings numerous users' dilemmas such as selecting algorithm, parameters, similarity/distance metrics, thresholds, etc.

Although an importance of studying complex microbial communities in a natural environments is recognized, studies that address reliability of derived conclusions are just recently increasingly appreciated. Inconsistent results may be implication of unstable OTUs obtained by *de novo* clustering [113], different diversity measures [114] or as examined in the detection of enterotypes, results may be affected by OTU taxonomic level, OTU-picking method, 16S rRNA variable region and most substantially by distance metric and the clustering score method [115].

Ensemble clustering approaches hold potential for improving the robustness, stability and accuracy of discovered clusters. In this light, microbial diversity analysis may also benefit from an integration across multiple partitions. Benefits of the integration across different clusterings algorithms and parameters have been recently evidenced [116]. Here, in our study, integration scenario covers different beta diversity measures [117]. We used 24 beta diversity measures to quantify pairwise differences among samples and then ran spectral clustering [118] on the similarity matrices obtained by transforming pairwise distances. Finally, for the assessment of communities in microbiome samples we integrated the results of individual clusterings and applied two ensemble approaches–one proposed in this thesis that utilizes non-negative matrix factorization - NMF [53] and another well known consensus clustering - CONS [49]. As presented in the Chapter 4, those two algorithms were the best according to the benchmarking results.

7.1 Data

We used data from "Moving pictures of the human microbiome" study [119]. Data set encompasses approximately 69 million sequences obtained form NGS (*next-generation sequencing*) experiment that included 1967 microbiome samples extracted from oral, skin and gut sites on the human body of two individuals, female and male, sampled over 396 timepoints. Differences in microbial compositions between body sites and individuals were relatively stable over time what makes data set suitable for evaluating clustering algorithms. Data were accessed through MG-RAST API [120] after quality filtering step. Overall size of the set is $\approx$ 12 GB. Sample labels that indicate microbiome host and body site were extracted from corresponding metadata. Number of samples across gender/body site labels are presented in Table 7.1.

Table 7.1: Microbiome experimental data.

gender	place	number of samples
female	oral	135
female	skin	268
female	gut	131
male	oral	373
male	skin	724
male	gut	336

7.2 Methods

Diversity studies can be reference-based, i.e. rely on sequence similarity against reference database or reference-free where sequences are clustered based on the similarities to one another. In the first approach clustering can be performed largely in parallel, but only sequences that match a sequence in a reference database with high similarity are clustered and those below defined threshold are discard. In reference-free clustering, refereed as *de-novo*, all reads are clustered, but the process is not easily parallelized. Recently proposed subsampled open-reference OTU picking method [121] provides trade-of between these two. Here, we used reference-based approach to produce OTUs. To perform microbial community analysis we used QIIME package [31] extended with the ensemble clustering algorithms. QIIME package includes a large number of tools for processing and analysing microbial sequence data. Its workflow is presented in Fig. 7.1. Various pipelines can be performed starting from the raw sequence data to the final diversity analysis and visualizations. The steps that were conducted in our experiment are marked and numbered at workflow scheme.

Our experimental pipeline includes following steps:

1. Preprocessing of raw sequence data
2. OTU picking
3. Making OTU biom table
4. Measuring β–diversity among samples

In the preprocessing step, sequences undergo demultiplexing, removing primers and quality filtering. The next step, OTU picking, performs clustering of sequences. Sequences were clustered into OTUs by default taxonomy assigner - UCLUST [122] with a sequence similarity threshold of 97% or 99% against

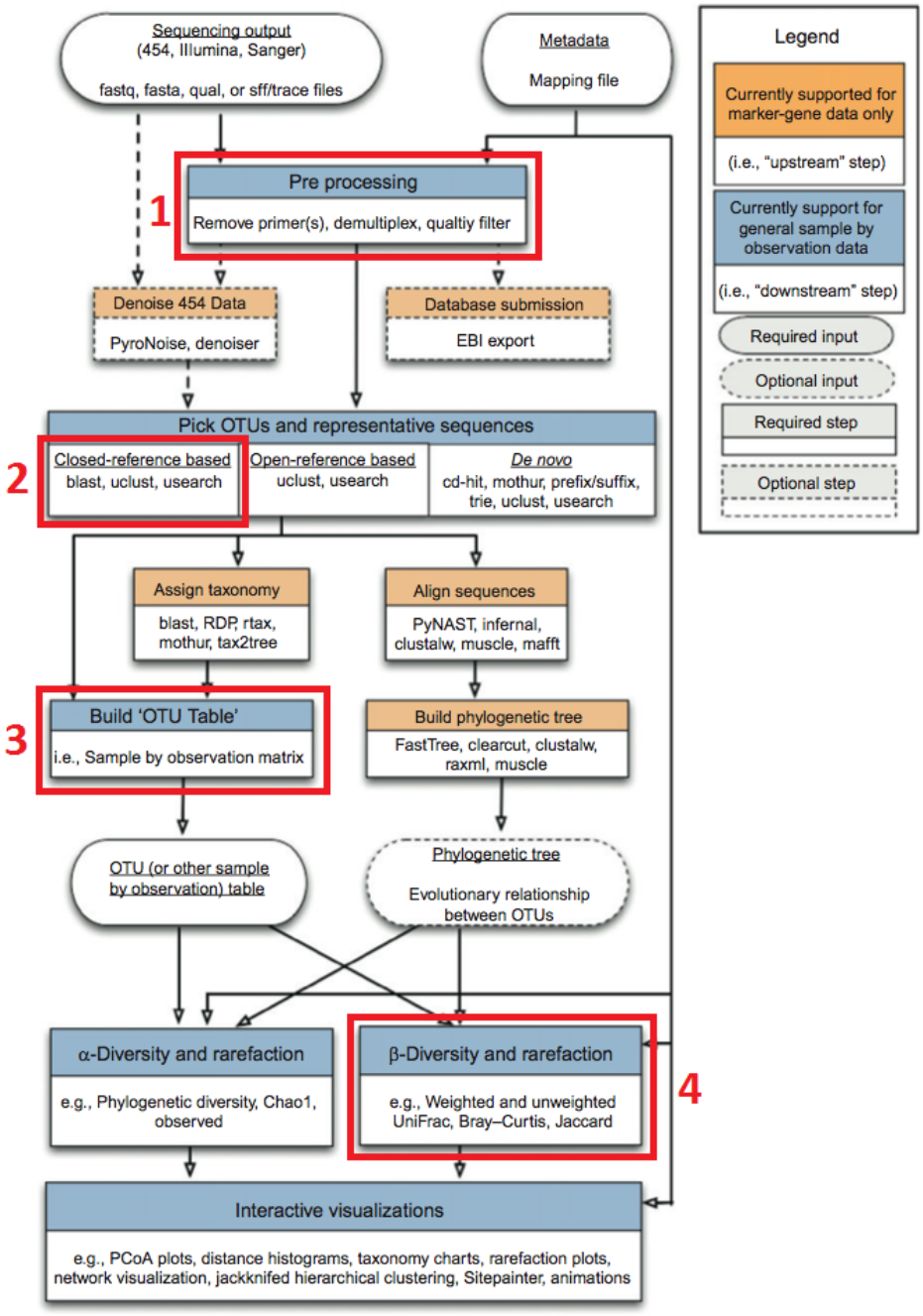


Figure 7.1: QIIME workflow shows available processing pipelines. Scheme from Knights Lab Wiki (https://sites.google.com/site/knightslabwiki/qiime-workflow) was further edited to denote steps used in our experiments.

Greengenes reference database [123]. Threshold of 97% is a commonly used rule of thumb to define species, but also tighter threshold of 99% have been proposed. Therefore, we explored both. Clustering algorithm, UCLUST is a greedy algorithm. Given the query sequence, it searches database of reference sequences. If UCLUST finds a sequence in the reference collection with similarity greater than or equal to defined threshold, it creates OTU defined by the reference sequence and assigns query sequence to it, otherwise query sequence is discarded. After clustering sequences in OTUs, results were processed in the third step - making OTUs biom table. OTU table summarizes taxonomy of samples in a form of observations counts per-sample. One dimension of OTU table denotes identified OTUs, and the other samples. Observations refer to the sequences assigned to OTUs, and the values in the table are their counts or frequencies. For studies where the number of samples is very large, many OTUs remain empty, without observations, for a given sample. Final OTU table in such case can be represented as sparse matrix. In terms of machine learning, OTUs denote features and we can further measure distances and similarities among samples in that feature space. This is actually, related to the biological concept of biodiversity that measures varieties at different scales. If it measures between-sample diversity, it is denoted as β diversity. To quantify beta diversity, we explored 24 non-phylogenetic beta diversity measures: (1) abundance weighted Jaccard distance, (2) binary Chi-square, (3) binary Chord, (4) binary Euclidean distance, (5) binary Hamming (6) binary Jaccard (7) binary Lennon (8) binary Ochiai (9) binary Pearson, (10) binary Sörensen-Dice (11) Bray-Curtis (12) Canberra, (13) Chi-square, (14) Chord, (15) Euclidean, (16) Gower, (17) Hellinger, (18) Kulczynski, (19) Manhattan distance, (20) Morisita-Horn, (21) Pearson, (22) Soergel, (23) Spearman rank and (24) Species profile distance.

Previously described steps form QIIME workflow were extended with two additional: one that runs spectral clustering and the other that integrates clustering results and creates final clusters. Spectral clustering was selected due to its property that can work directly with pair-wise distances/similarities. We also considered kernel k-means, but it produced clusters of lower quality compared to the spectral. Pairwise differences among samples were transformed into similarities by using element-wise transformation $S = e^{-D/(2\mu^2)}$, where D is a pair-wise β–diversity matrix, μ is mean value of that matrix, and S is the final similarity matrix. Spectral clustering then uses S as input.

The results of individual clustering on different pairwise distance matrices are combined to perform ensemble clustering. In NMF approach, as described in the Chapter 3, ensemble is represented as a matrix of cluster memberships $R = \{0, 1\}^{m \times n}$, where one dimension represents clusters (m is the total number of clusters produced by individual clusterings) and the other samples (n is the total number of examined samples). Factorization rank equals to the target number of clusters. Other approach, consensus clustering, integrates cluster memberships

into a consensus matrix $\mathbb{R}^{n \times n}$, where indexes correspond to samples. In a pairwise manner, matrix sums number of times each pair of samples was clustered together and divides it by number of times they were both present in the clustering output. Final values range between 0 and 1. Consensus matrix can be viewed as a similarity matrix and post-processed through additional clustering methods to obtain final clusters. Here, we used agglomerative hierarchical clustering on consensus matrix.

Although, our workflows include clusterings in two steps: OTU picking and clustering samples according to calculated β–diversities, here, fusion is done only at samples level. Creating clusters of sequences, OTU picking, is fixed with selected close reference approach. Variability at clustering sequences can be elicited by different algorithms for OTU picking, or by different approaches: open and *de-novo*. However, those extensive experiments remain for future work.

7.3 Results

7.3.1 Ensemble creation

To evaluate effectiveness of the clusterings on microbiome samples we employed v-measure [124] and adjusted rand index [125], two commonly used measures for evaluating clusters against true labels. V-measure calculates a harmonic mean between homogeneity and completeness. If all of clusters contain only samples which are members of a single class, homogeneity is the highest and equals 1. Completeness is satisfied if samples that are members of a given class belong the same cluster. The good clustering result should highly score in both, homogeneity and completeness. Therefore v-measure takes them both into account. Here we measured how clustering results align with 6 labels corresponding to different gender/body sites. Examined microbiome samples belong to a time series study. Although temporal variation exists, stable patterns among body habitats and individuals emerge, thus making the data suitable for benchmarking of clustering algorithms.

We evaluate all individual clusterings obtained with spectral clustering algorithm on different beta diversity matrices and two ensemble approaches. The results evaluated by adjusted rand index and v-measure scores for experiments where similarity cut-off was set to 97% are summarized in Fig 7.2 and 7.3, respectively. Results of spectral clusterings on different β–diversity measures and integrative clusterings by NMF and CONS were presented with horizontal bars. Vertical blue line denotes average performance of assembling partitions, and red line highlights the score of better ensemble approach. Figures unveil that ensemble clusterings outperform an average performance of individual clusterings,

NMF reached better result than CONS, and it was slightly below the best individual clustering score in the ensemble. We can observe variability of the obtained results elicited by chosen distance measure. If we compare results by the used evaluation measures, adjusted rand index or v-measure, the results differ to some extent only in the rankings of individual clustering results, but general conclusions are the same. NMF, as well as CONS, ensemble approaches provided result that overcomes dependencies on underlying diversity measures. This results is confirmed by both evaluation measures.

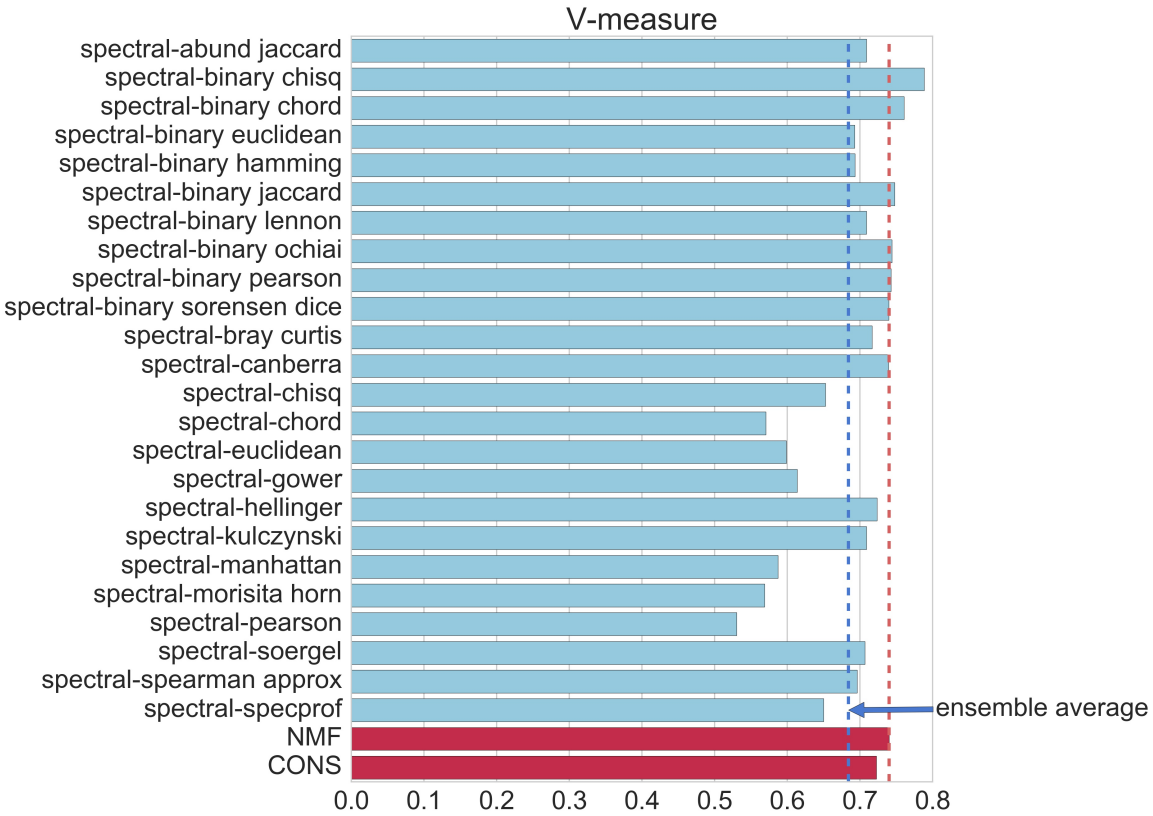


Figure 7.2: V-measure between true labels and cluster labels from spectral clustering applied on different pairwise diversity matrices, NMF and CONS algorithms. Prior to clustering samples, cut-off threshold in OTU-picking was set to 97%. Blue vertical line denotes average v-measure of the ensemble's ingredients and red indicates better ensemble approach.

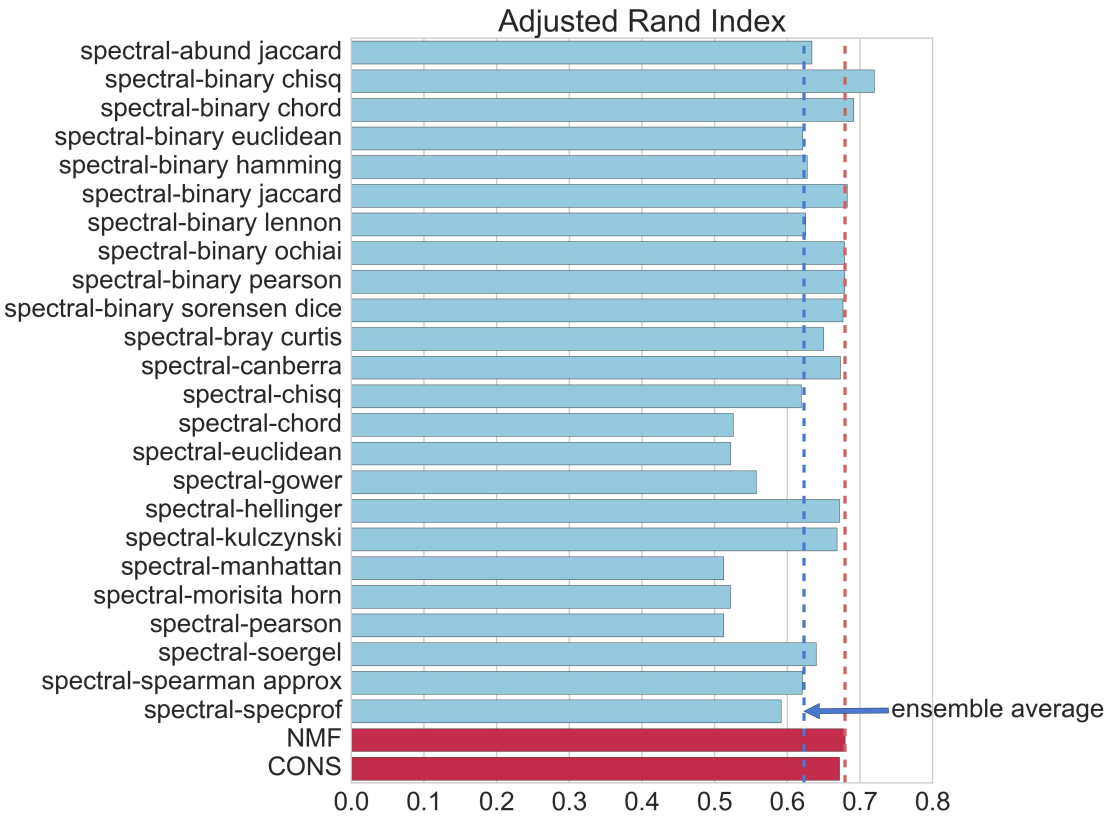


Figure 7.3: Adjusted rand index between true labels and cluster labels from spectral clustering applied on different pairwise diversity matrices, NMF and CONS algorithms. Prior to clustering samples, cut-off threshold in OTU-picking was set to 97%. Blue vertical line denotes average adjusted rand index of the ensemble's ingredients and red indicates better ensemble approach.

Similar results were obtained on different sequence similarity cut-off of 99% (Fig 7.4 and 7.5). NMF and CONS, outperformed average score of individual clustering that entered ensemble. Scores among individual clustering change, as well as their rankings. The best score comes from other diversity measure, while ensemble algorithms remain stable. The best v-measure score among individual clusterings in experiments with cut-off of 97% was produced on a β–diversity matrix measured by binary Chi-square (Fig 7.2) and on 99% cut-off (Fig 7.4) the best score comes from Kulczynski measure. For adjusted rand index score, the best result among individual clusterings for 97% cut-of was obtained with binary

Chi-square and for 99% cut-off with Chi-square. Related to Chi-square measure, we can observe how small change in cut-off threshold highly impacts outcome from the best to below average of the ensemble. Interestingly, Gower and Canberra distances, recommended as the well performing for detecting clusters [114], here produced divergent results. While Canberra distance was among better metrics, but still below NMF ensemble, Gower distance failed to detect clusters that align with labels of human microbiome data set.

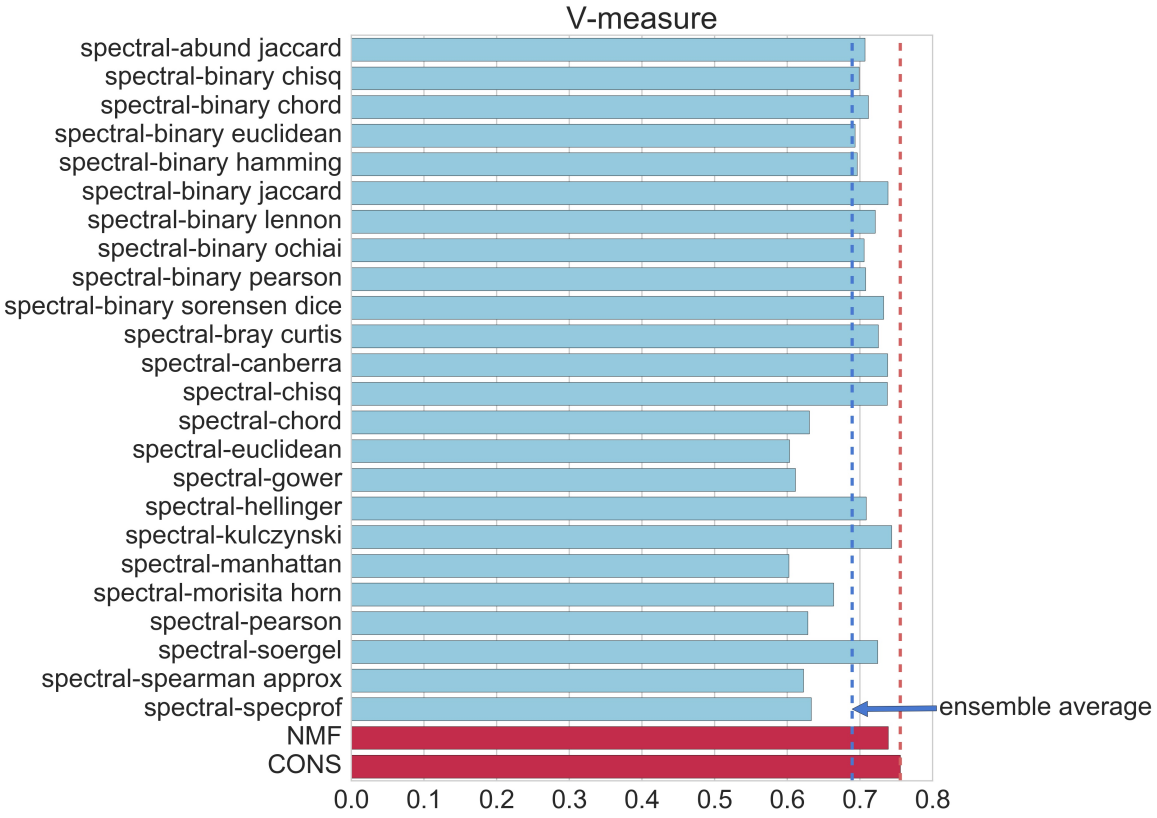


Figure 7.4: V-measure between true labels and cluster labels from spectral clustering applied on different pairwise diversity matrices, NMF and CONS algorithms. Prior to clustering samples, cut-off threshold in OTU-picking was set to 99%. Blue vertical line denotes average v-measure of the ensemble's ingredients and red indicates better ensemble approach

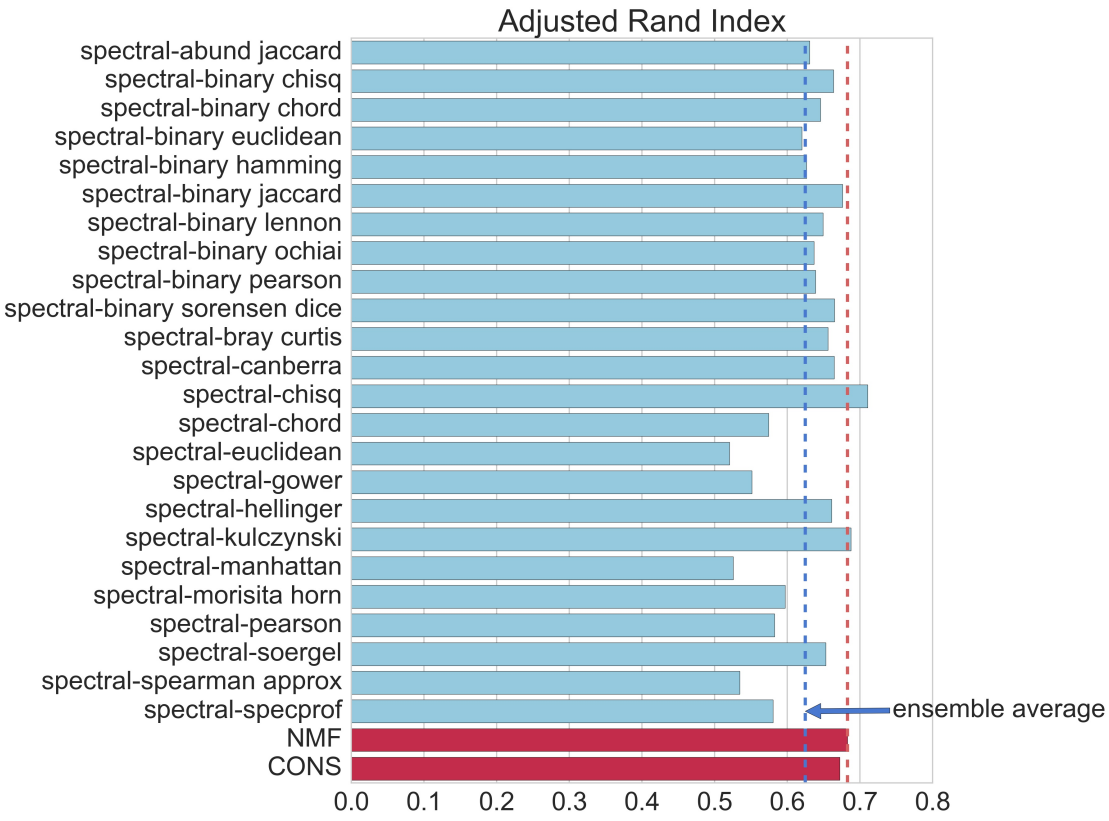


Figure 7.5: Adjusted rand index between true labels and cluster labels from spectral clustering applied on different pairwise diversity matrices, NMF and CONS algorithms. Prior to clustering samples, cut-off threshold in OTU-picking was set to 99%. Blue vertical line denotes average adjusted rand index of the ensemble's ingredients and red indicates better ensemble approach.

7.3.2 Stability analysis

To further examine stability of results, we performed random selection of 1000 out of 1967 samples. This subsampling and overall process of OTU picking, forming OTU tables, calculating β–diversities, clustering and assembling was repeated 50 times. Those experiments allowed us to make more general conclusions on the usefulness of the integration in the clustering microbiome samples.

The results of subsampling experiments, evaluated by adjusted rand index and v-measure, are summarized by box plots (Fig 7.6), one for each of the ap-

proaches - spectral clustering combined with different diversity measures, NMF and CONS ensemble algorithms. We can observe high variability of individual clusterings and stability of the ensemble algorithms. Running experiments on subsamples allowed us to measure statistical significance. ANOVA tests indicate that significant difference exists among methods ($p < 10^{-13}$ and $p < 10^{-12}$ for v-masure and adjusted rand index, respectively). Post-hoc Tukey test with 99% confidence reveals that both, NMF and CONS, significantly outperformed results of individual clusterings, while difference in the mean performance of the ensemble approaches was not significant. Overall results imply that while spectral clustering coupled with some β–diversity measure can provide better result than ensemble, chances that we will select wining measure are small. This puts ensemble approaches into favourable position.

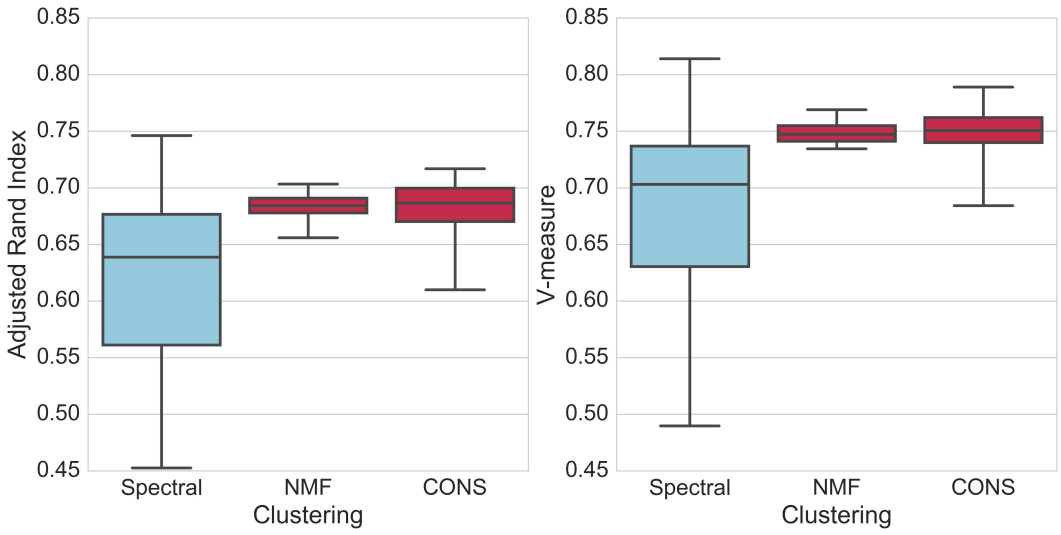


Figure 7.6: Adjusted rand index and v-measure scores on 50 random subsampling experiments. For each subsampling experiment we randomly selected 1000 samples from microbiome data set.

7.3.3 Distributed clustering using the cluster ensembles

In the preceding section subsampling was used to evaluate stability of the results – experiments on randomly selected subsamples were just evaluated on the corresponding labels. Here, we explore subsampling in the context of ensemble

generation [126]. We can think of this framework as object distributed clustering that utilizes ensemble to merge partially clustered data [127]. Potentially, this approach is of great importance for scaling up the experiments through distribution and parallelization. In distributed set-up, data are initially split into fragments and individual clusterings only look at fractions of the data. To enable a meaningful assembling of partitions from different fragments two conditions in splitting the data have to be fulfilled: each object have to be assigned to at least one fragment, and fragments have to be overlapping.

Object distributed ensemble clustering can achieve lower computational complexity depending on the number of data fragments and their sizes that are necessary to reconstruct or surpass results on whole data sets. Spectral clustering, which is a baseline algorithm in our experiments, highly impacts overall complexity. In general, its computational complexity is $O(N^3)$, where N denotes number of samples. Constructing the pairwise distance/affinity matrices between samples is an another bottleneck of experimental pipelines with a computational cost of $O(N^2d)$, where d represents dimensionality of data (here, number of OTUs). Hence running more instances of spectral clusterings, but on smaller set of samples, could potentially reduce computational time.

To evaluate possibility of assembling partitions coming from data fragments we used subsamples of size 500 ($\approx \frac{1}{4}$ of overall number of samples) and created pool of 40 random data fragments of that size (by performing 10 independent splits of data set into 4 chunks). Fig. 7.7 reports on the results of object distributed clustering by NMF and CONS ensemble algorithm depending on the number of data fragments used in the integration. Results present average values across 10 runs with corresponding standard deviation estimates. Each experimental run started with 5 fragments (minimum number that can fulfil aforementioned splitting criterion), followed by subsequent adding of new fragments. On every fragment β–diversity matrices were calculated and then passed to spectral clustering. For defined number of fragments result of spectral clusterings were assembled for NMF and CONS to evaluate performance of integration. Horizontal lines in Fig. 7.7 represent results of NMF, CONS and average spectral clustering performance on the whole data set (see Fig. 7.3).

Results suggest that less than 10 fragments are enough to surpass average performance of spectral clustering on the full data set. Performance of CONS progressed better in the initial accumulation of the evidences, while NMF outperformed CONS for ensemble sizes of 15 and higher. We can notice interesting result when 20 partitions are integrated. At that point both, NMF and CONS, exceeded the results obtained on the full set of samples clustered at once, suggesting the additional benefits of assembling the clusterings from fragmented data.

On this point we can elaborate computational pros and cons. Here, overall size of the data is increased 5 times (20 data fragments * 1/4 of overall data size), but fragments are processed independently. Within fragments we worked on $N/4$

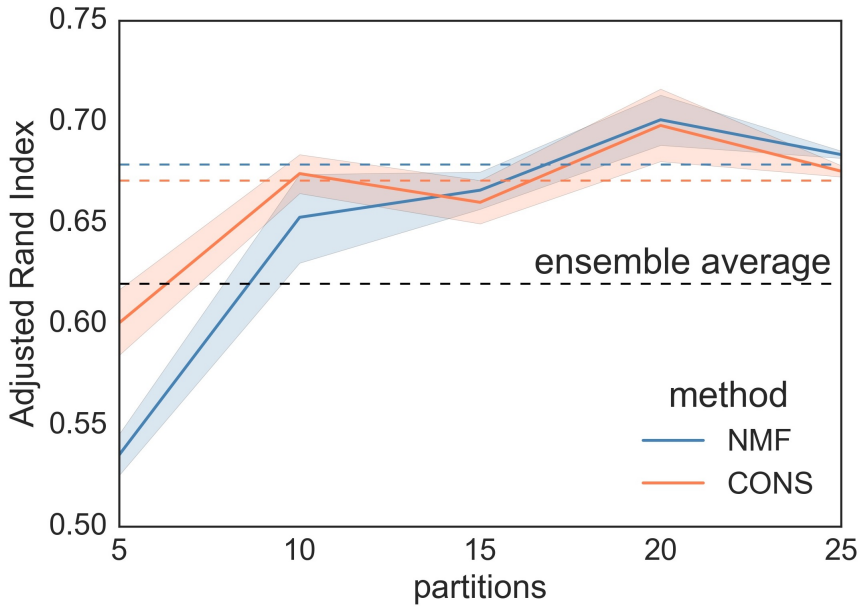


Figure 7.7: Adjusted rand index scores of NMF and CONS in object distributed set-up as the functions of the number of partitions (fragments) that participate in the ensemble.

samples what highly reduces time for running spectral clustering ($\approx 1/64$). Calculation of pair-wise β–diversity matrices is reduced 16 times due to quadratic complexity over N. This part also linearly depends on d - the number of features (OTUs) In clustering microbiome samples we have a dimensionality reduction (smaller d) that is not so obvious. When working with smaller number of microbiome samples, size of feature vector–number of discovered OTUs–decreases. Number of OTUs on full data set was 35257, while on fragments of 500 samples that number was reduced at $\approx$ 26000. If we take into account that there are 20 fragments for processing then execution time reduces $\approx 20/64$ times due to spectral clustering. Execution times of measuring β–diversities on 20 fragments vs full data set are similar (ratio is $\approx 20 * 0.7/16$).

On the other side we have reference based OTU-picking that scales linearly with N for analysed data set. As already noted we are using 5 times more samples. Since OTU picking is time consuming this would increase overall execution time on fragmented data. However, when we utilised multicore/multiprocessor environment supported by QIIME we managed to reduce execution time of OTU picking and thus balance overall cons and pros of object distributed ensemble clustering. In summary, on ensemble size of 20 fragments we obtained increased quality in approximately same execution time. From Fig. 7.7 we can observe that

for achieving the same quality of results as those on the whole data, less then 20 fragments are needed, i.e. we can accomplish the same quality of results in reduced execution time.

Detailed analysis would require examination on other microbiome data sets. Further investigation on how small fragments could be to still enable reconstructing overall result will be interesting. Also there exists approximate versions of spectral clustering that should be analysed and evaluated from both aspects, execution time and quality. Another alternative for speeding up experiments arise from the nature of late integration techniques that allows overall framework to span across distributed computers. Once data fragments are created, all individual clusterings can run in parallel and their results are assembled at the end. This enables further reducing of time in running computationally demanding ensemble clustering workflows on microbiome data.

7.4 Discusion

The study presented here underscores the sensitivity of clustering results on choosing beta diversity measure that further leads to the uncertainties in the results interpretation. To avoid risk of selecting the less appropriate metrics and obtaining misleading or vague conclusions, we propose using ensemble approaches in clustering. Ensemble clusterings produced stable results that highly surpassed average of the ensemble and were on the level of the best in the ensemble. These results were further confirmed by running experiments on random subsamples. NMF approach performed slightly better in terms of the lower variance compared to the CONS. Improved stability of the ensemble approaches comes at the price of larger computations. Ensemble clustering requires multiple runs of clustering algorithms under different input settings and that pose additional challenges in a large scale studies. Initial study on distributed version of ensemble clustering provide us promising results. Our future work will be extended on the environmental and soil metagenomics where data exceed TB size and good reference databases are missing. We need distributed computing solutions and highly parallel workflows for running such experiments.

Chapter 8

Conclusion

Clustering is of a great importance for preliminary and explorative data analysis, however as demonstrated on the numerous examples, different methods can deliver distinct partitions of the data. Integrative or ensemble clustering algorithms combine multiple partitions with aim to strengthen the quality and stability of the clustering. Further gains in the quality of discovered clusters may stem from data integration, as different data sources may provide different but complementary insight into the observed system.

In this thesis we have proposed integration methods based on nonnegative matrix factorization that can fuse clusterings stemming from different data sets, different data preprocessing steps or different subsamples of features or objects. Proposed methods are evaluated from several points of view on typical machine learning data sets, synthetics data, and above all, on data coming form bioinformatics realm, which rise is fuelled by technological revolutions in molecular biology. For a vast amounts of 'omics' data that are nowadays available sophisticated computational methods are necessary. We evaluated methods on problems from cancer genomics, functional genomics and metagenomics. Experiments on a large collection of cancer genomics data sets shed light on interplay between diversity, number of partitions and the final result of integrative clustering. The results that are of interest for functional genomics suggest that proposed approach based on nonnegative matrix factorization can fuse diverse data sources and infer gene groups with high functional enrichment and improved gene coverage. Regularised version of the algorithm brought additional improvement. On metagenomics problem of clustering microbime samples, we showed that the proposed NMF method was able to produce stable clusters, that are robust on the change in parameters and subsamplng. Also, we presented perspective of distributed

ensemble clusterings in speeding metagenomics workflows. Our proposed method is general and compares favourably to alternative integration approaches.

Integrative clustering is an area of active research and still remains a lot of exciting research for the future. Our future work will include a rethinking of ensemble creation [128], exploration of semi-supervised approaches, and scaling issues [129], [130] engendered by data of growing volume, dimensionality and complexity.

Appendix A: Benchmarking datasets

Table A1: UCI datasets

Data set	Samples	K	Kernel
iris	150	3	RBF
wine	178	3	RBF
seed	210	3	RBF
wdbc	569	2	RBF
beast-cancer-wisconsin-cont	683	2	RBF
satimage	480	6	RBF
pendigit	800	10	RBF
image-segmentation	560	7	RBF
zoo	101	7	RBF
letter-recognition,	520	26	RBF
soybean	47	4	RBF
yeast	505	10	RBF
shuttle	374	7	RBF
optdigits	800	10	RBF
parkinsons	195	2	RBF
ecoli	336	8	Linear
movement-libras	360	15	Linear
semeion-handwritten	500	10	Linear
dermatology	358	6	Linear
led7digit	500	10	Linear

Table A2: Synthetic datasets

Data set	Samples	K	Kernel
aggregation	433	7	RBF
compound	399	6	RBF
pathbased	300	3	RBF
d31	620	31	RBF
jain	373	2	RBF
flame	240	2	RBF
r15	600	15	Linear
hepta	212	7	Linear
lsun	400	3	Linear
tetra	400	4	Linear
2dnormals	500	2	Linear
cassini	500	3	Linear
cuboids	500	4	Linear
hypercube	496	8	Linear
shapes	500	4	Linear
simplex	500	5	Linear
smiley	500	4	Linear
waveform	500	3	Linear
twonorm	500	2	Linear
xor	500	4	Linear

Table A3: Genomic datasets

Data set	Samples	K	Kernel
Affymetrix-Armstrong-2002-v1	72	2	RBF
Affymetrix-Armstrong-2002-v2	72	3	RBF
Affymetrix-Bhattacharjee-2001	203	5	RBF
Affymetrix-Chowdary-2006	104	2	RBF
Affymetrix-Dyrskjot-2003	40	3	RBF
Affymetrix-Golub-1999-v1	72	2	RBF
Affymetrix-Golub-1999-v2	72	3	RBF
Affymetrix-Laiho-2007	37	2	RBF
Affymetrix-Nutt-2003-v1	50	4	RBF
Affymetrix-Nutt-2003-v2	28	2	RBF
Affymetrix-Nutt-2003-v3	22	2	RBF
Affymetrix-Pomeroy-2002-v2	42	5	RBF
Affymetrix-Ramaswamy-2001	190	14	RBF
Affymetrix-Singh-2002	102	2	RBF
Affymetrix-Su-2001	174	10	RBF
Affymetrix-West-2001	49	2	RBF
Affymetrix-Yeoh-2002-v1	248	2	RBF
Affymetrix-Yeoh-2002-v2	248	6	RBF
cDNA-Alizadeh-2000-v1	42	2	Linear
cDNA-Alizadeh-2000-v2	62	3	Linear
cDNA-Alizadeh-2000-v3	62	3	Linear
cDNA-Bredel-2005	50	3	Linear
cDNA-Garber-2001	66	4	Linear
cDNA-Khan-2001	83	4	Linear
cDNA-Lapointe-2004-v1	69	3	Linear
cDNA-Lapointe-2004-v2	110	4	Linear
cDNA-Liang-2005	37	3	Linear
cDNA-Risinger-2003	42	4	Linear
cDNA-Tomlins-2006-v1	104	5	Linear
cDNA-Tomlins-2006-v2	92	4	Linear

Appendix B: Benchmarking results

Table B1: Adjusted rand index results on UCI datasets

Data set	Average	NMF	CONS	HGPA	MCLA	DICLENS	OKKC
iris	0.701	0.727	0.724	0.315	0.729	0.724	0.745
wine	0.324	0.363	0.362	0.109	0.363	0.364	0.290
seed	0.660	0.712	0.704	0.121	0.712	0.709	0.621
wdbc	0.167	0.482	0.487	-0.001	0.403	0.156	0.440
beast-cancer-w.	0.375	0.713	0.732	-0.001	0.604	0.740	0.368
satimage	0.431	0.478	0.430	0.280	0.500	0.420	0.428
pendigit	0.505	0.588	0.549	0.339	0.586	0.538	0.521
image-segment.	0.264	0.367	0.332	0.204	0.251	0.296	0.353
zoo	0.372	0.498	0.613	0.433	0.461	0.609	0.232
letter-recognition	0.105	0.154	0.151	0.104	0.148	0.087	0.078
soybean	0.397	0.617	0.609	0.607	0.598	0.602	0.365
yeast	0.105	0.177	0.199	0.125	0.175	0.139	0.109
shuttle	0.169	0.310	0.141	0.133	0.077	0.156	0.267
optdigits	0.511	0.690	0.677	0.349	0.681	0.629	0.612
parkinsons	0.081	0.183	0.095	0.001	0.179	0.097	0.148
ecoli	0.376	0.384	0.447	0.256	0.355	0.486	0.423
movement-libras	0.14	0.141	0.141	0.129	0.137	0.133	0.138
semeion-handwr.	0.31	0.342	0.348	0.236	0.34	0.324	0.311
dermatology	0.579	0.672	0.646	0.397	0.586	0.644	0.617
led7digit	0.346	0.456	0.383	0.254	0.425	0.334	0.380
final average rank	-	**1.825**	2.850	5.600	3.350	3.375	4.000

Table B2: Normalized mutual information results on UCI datasets

Data set	Average	NMF	CONS	HGPA	MCLA	DICLENS	OKKC
iris	0.724	0.750	0.750	0.367	0.749	0.750	0.772
wine	0.370	0.418	0.417	0.131	0.417	0.419	0.331
seed	0.654	0.702	0.694	0.153	0.700	0.700	0.618
wdbc	0.176	0.459	0.462	0.001	0.382	0.193	0.415
beast-cancer-w.	0.394	0.674	0.683	0.000	0.585	0.692	0.419
satimage	0.55	0.583	0.545	0.374	0.585	0.532	0.553
pendigit	0.657	0.701	0.697	0.491	0.695	0.681	0.678
image-segment.	0.437	0.523	0.529	0.317	0.425	0.491	0.515
zoo	0.547	0.685	0.738	0.653	0.665	0.722	0.445
letter-recognition	0.396	0.465	0.466	0.395	0.451	0.345	0.362
soybean	0.529	0.756	0.760	0.742	0.740	0.749	0.495
yeast	0.216	0.310	0.330	0.239	0.298	0.263	0.229
shuttle	0.370	0.479	0.375	0.248	0.223	0.379	0.431
optdigits	0.610	0.775	0.777	0.489	0.766	0.755	0.730
parkinsons	0.051	0.103	0.056	0.006	0.099	0.057	0.088
ecoli	0.548	0.582	0.612	0.443	0.565	0.613	0.603
movement-libras	0.387	0.391	0.391	0.367	0.383	0.375	0.375
semeion-handwr.	0.486	0.520	0.534	0.401	0.507	0.507	0.491
dermatology	0.722	0.760	0.757	0.512	0.721	0.752	0.739
led7digit	0.485	0.559	0.518	0.398	0.539	0.497	0.513
final average rank	-	**1.850**	2.550	5.600	3.550	3.400	4.050

Table B3: Silhouette index results on UCI datasets

Data set	Average	NMF	CONS	HGPA	MCLA	DICLENS	OKKC
iris	0.566	0.612	0.608	0.176	0.606	0.613	0.577
wine	0.429	0.534	0.530	0.043	0.528	0.531	0.305
seed	0.473	0.511	0.508	0.010	0.509	0.511	0.439
wdbc	0.239	0.687	0.686	0.000	0.553	0.666	0.607
beast-cancer-w..	0.344	0.633	0.647	-0.002	0.566	0.649	0.363
satimage	0.376	0.376	0.389	0.142	0.356	0.387	0.255
pendigit	0.299	0.323	0.336	0.075	0.308	0.322	0.279
image-segment.	0.191	0.186	0.286	0.013	0.134	0.229	0.252
zoo	0.220	0.469	0.528	0.352	0.412	0.505	0.035
letter-recognition	0.011	0.075	0.058	-0.049	0.052	-0.084	-0.072
soybean	0.208	0.38	0.387	0.375	0.368	0.384	0.177
yeast	-0.006	0.061	0.079	-0.059	0.049	-0.009	0.003
shuttle	0.214	0.258	0.446	-0.135	-0.113	0.316	0.308
optdigits	0.186	0.258	0.251	0.054	0.243	0.239	0.217
parkinsons	0.382	0.518	0.620	0.017	0.504	0.616	0.458
ecoli	0.174	0.183	0.225	0.005	0.177	0.205	0.225
movement-libras	0.203	0.214	0.211	0.083	0.197	0.12	0.206
semeion-handw.	0.096	0.105	0.097	0.036	0.095	0.061	0.099
dermatology	0.387	0.464	0.421	0.08	0.371	0.437	0.402
led7digit	0.309	0.413	0.311	0.151	0.350	0.202	0.331
final average rank	-	**2.100**	**2.100**	5.750	4.050	3.400	4.150

Table B4: Isolation Index results on UCI datasets

Data set	Average	NMF	CONS	HGPA	MCLA	DICLENS	OKKC
iris	0.942	0.963	0.965	0.579	0.957	0.967	0.949
wine	0.881	0.973	0.969	0.563	0.967	0.973	0.767
seed	0.934	0.945	0.945	0.470	0.944	0.945	0.925
wdbc	0.993	0.993	0.992	0.504	0.924	0.970	0.936
beast-cancer-w.	0.833	0.955	0.956	0.504	0.914	0.958	0.881
satimage	0.898	0.907	0.911	0.637	0.901	0.914	0.883
pendigit	0.898	0.922	0.941	0.603	0.910	0.921	0.903
image-segment.	0.886	0.882	0.934	0.562	0.847	0.904	0.897
zoo	0.666	0.839	0.885	0.770	0.811	0.900	0.529
letter-recognition	0.457	0.617	0.658	0.445	0.609	0.725	0.353
soybean	0.615	0.798	0.814	0.789	0.784	0.806	0.582
yeast	0.508	0.694	0.746	0.453	0.663	0.794	0.541
shuttle	0.908	0.890	0.949	0.456	0.599	0.911	0.863
optdigits	0.739	0.889	0.906	0.533	0.884	0.903	0.852
parkinsons	0.968	0.961	0.978	0.526	0.944	0.969	0.915
ecoli	0.775	0.796	0.838	0.493	0.778	0.844	0.825
movement-libras	0.712	0.721	0.742	0.6	0.697	0.758	0.736
semeion-handw.	0.686	0.718	0.75	0.539	0.695	0.755	0.692
dermatology	0.944	0.959	0.963	0.644	0.934	0.952	0.947
led7digit	0.862	0.876	0.873	0.639	0.855	0.885	0.865
final average rank	-	2.825	1.850	5.800	4.550	**1.675**	4.300

Table B5: Adjusted rand index results on synthetic datasets

Data set	Average	NMF	CONS	HGPA	MCLA	DICLENS	OKKC
aggregation	0.807	0.872	0.869	0.536	0.841	0.880	0.816
compound	0.557	0.535	0.538	0.314	0.503	0.539	0.618
pathbased	0.444	0.518	0.487	0.096	0.544	0.474	0.773
d31	0.302	0.303	0.301	-0.002	0.300	0.303	0.286
jain	0.281	0.545	0.473	-0.003	0.491	0.479	0.411
flame	0.721	0.926	0.858	0.634	0.936	0.782	0.813
r15	0.674	0.989	0.987	0.512	0.911	0.939	0.917
hepta	0.500	0.655	0.551	0.119	0.536	0.472	0.405
lsun	0.644	0.881	0.844	-0.008	0.850	0.843	0.994
tetra	0.598	0.725	0.760	0.040	0.770	0.697	0.609
2dnormals	0.723	0.955	0.966	-0.006	0.880	0.928	0.884
cassini	0.653	0.911	0.988	0.231	0.910	0.865	0.781
cuboids	0.726	0.975	0.989	-0.006	0.980	0.971	0.801
hypercube	0.807	0.936	0.928	0.003	0.934	0.933	0.929
shapes	0.592	0.801	0.716	-0.005	0.613	0.709	0.671
simplex	0.249	0.250	0.250	0.082	0.249	0.250	0.250
smiley	0.852	0.890	0.888	0.000	0.891	0.890	0.150
waveform	0.210	0.178	0.228	0.007	0.168	0.177	0.145
final average rank	-	**2.125**	2.675	5.800	3.250	3.300	3.850

Table B6: Normalized mutual information results on synthetic datasets

Data set	Average	NMF	CONS	HGPA	MCLA	DICLENS	OKKC
aggregation	0.877	0.906	0.908	0.683	0.890	0.910	0.893
compound	0.718	0.717	0.721	0.472	0.683	0.719	0.746
pathbased	0.493	0.590	0.568	0.127	0.607	0.557	0.784
d31	0.354	0.507	0.550	0.429	0.434	0.190	0.422
jain	0.358	0.359	0.357	0.000	0.357	0.359	0.347
flame	0.273	0.488	0.437	0.001	0.445	0.441	0.385
r15	0.863	0.970	0.950	0.812	0.968	0.923	0.932
hepta	0.796	0.993	0.993	0.688	0.941	0.963	0.968
lsun	0.550	0.694	0.589	0.166	0.578	0.523	0.526
tetra	0.688	0.877	0.852	0.000	0.843	0.848	0.997
2dnormals	0.535	0.603	0.602	0.000	0.604	0.602	0.583
cassini	0.611	0.727	0.763	0.063	0.755	0.699	0.624
cuboids	0.795	0.966	0.976	0.000	0.908	0.943	0.912
hypercube	0.807	0.959	0.994	0.413	0.947	0.928	0.916
shapes	0.803	0.984	0.994	0.000	0.984	0.981	0.893
simplex	0.822	0.921	0.917	0.013	0.919	0.920	0.921
smiley	0.713	0.854	0.819	0.000	0.739	0.794	0.793
waveform	0.350	0.359	0.360	0.110	0.357	0.360	0.359
twonorm	0.769	0.816	0.813	0.002	0.817	0.816	0.814
xor	0.275	0.219	0.308	0.014	0.197	0.217	0.163
final average rank	-	**2.100**	2.375	5.900	3.425	3.350	3.700

Table B7: Silhouette Index results on synthetic datasets

Data set	Average	NMF	CONS	HGPA	MCLA	DICLENS	OKKC
aggregation	0.500	0.525	0.504	0.200	0.502	0.526	0.434
compound	0.437	0.462	0.460	0.096	0.394	0.465	0.445
pathbased	0.315	0.498	0.515	-0.001	0.480	0.519	0.393
d31	-0.232	-0.290	-0.298	-0.283	-0.306	-0.407	-0.214
jain	0.484	0.484	0.484	-0.005	0.484	0.484	0.480
flame	0.260	0.344	0.350	-0.005	0.337	0.349	0.314
r15	0.463	0.671	0.619	0.272	0.674	0.542	0.591
hepta	0.058	0.092	0.063	0.117	0.095	0.12	0.045
lsun	0.437	0.727	0.724	0.209	0.646	0.679	0.663
tetra	0.323	0.370	0.329	0.083	0.332	0.297	0.451
2dnormals	0.321	0.439	0.428	-0.020	0.425	0.421	0.482
cassini	0.369	0.406	0.411	-0.022	0.402	0.358	0.408
cuboids	0.432	0.536	0.541	-0.016	0.498	0.511	0.505
hypercube	0.364	0.650	0.734	-0.039	0.645	0.597	0.553
shapes	0.480	0.664	0.676	-0.021	0.663	0.657	0.612
simplex	0.369	0.428	0.426	-0.019	0.428	0.428	0.423
smiley	0.420	0.527	0.483	-0.017	0.437	0.476	0.493
waveform	0.380	0.388	0.388	0.070	0.388	0.388	0.387
twonorm	0.296	0.303	0.303	0.000	0.303	0.303	0.303
xor	0.216	0.223	0.211	-0.021	0.226	0.220	0.237
final average rank	-	**2.375**	2.550	5.800	3.725	3.350	3.200

Table B8: Isolation Index results on synthetic datasets

Data set	Average	NMF	CONS	HGPA	MCLA	DICLENS	OKKC
aggregation	0.969	0.980	0.986	0.672	0.972	0.981	0.977
compound	0.938	0.939	0.941	0.638	0.933	0.939	0.954
pathbased	0.857	0.964	0.978	0.474	0.948	0.978	0.963
d31	0.179	0.377	0.467	0.279	0.323	0.800	0.275
jain	0.983	0.982	0.983	0.443	0.983	0.982	0.980
flame	0.949	0.950	0.958	0.455	0.925	0.954	0.910
r15	0.904	0.980	0.977	0.742	0.971	0.962	0.975
hepta	0.863	0.994	0.995	0.654	0.944	0.969	0.993
lsun	0.954	0.968	0.959	0.647	0.929	0.931	0.971
tetra	0.887	0.947	0.942	0.251	0.930	0.938	0.989
2dnormals	0.967	0.969	0.969	0.504	0.968	0.969	0.970
cassini	0.960	0.967	0.971	0.460	0.959	0.962	0.962
cuboids	0.977	0.996	0.998	0.258	0.985	0.980	0.989
hypercube	0.904	0.982	0.997	0.374	0.966	0.965	0.990
shapes	0.973	0.998	1.000	0.241	0.996	0.996	0.996
simplex	0.924	0.962	0.962	0.222	0.961	0.962	0.964
smiley	0.971	0.982	0.984	0.239	0.969	0.969	0.982
waveform	0.894	0.903	0.903	0.591	0.902	0.903	0.901
twonorm	0.920	0.929	0.929	0.505	0.929	0.929	0.929
xor	0.862	0.857	0.863	0.254	0.850	0.854	0.865
final average rank	-	2.575	**1.825**	6.000	4.375	3.350	2.875

Table B9: Adjusted rand index results on cancer genomics datasets

Data set	Average	NMF	CONS	HGPA	MCLA	DICLENS	OKKC
Armstrong-2002-v1	0.201	0.254	0.256	0.016	0.252	0.240	0.213
Armstrong-2002-v2	0.541	0.711	0.675	0.544	0.683	0.691	0.527
Bhattacharjee-2001	0.200	0.193	0.212	0.074	0.137	0.208	0.192
Chowdary-2006	0.144	0.121	0.066	-0.008	0.066	0.066	0.535
Dyrskjot-2003	0.483	0.583	0.571	0.425	0.560	0.566	0.470
Golub-1999-v1	0.620	0.637	0.637	0.003	0.637	0.635	0.595
Golub-1999-v2	0.503	0.689	0.625	0.371	0.646	0.654	0.471
Laiho-2007	0.202	0.229	0.222	0.072	0.227	0.214	0.162
Nutt-2003-v1	0.303	0.371	0.360	0.316	0.329	0.350	0.274
Nutt-2003-v2	0.121	0.096	0.053	0.141	0.095	0.038	0.321
Nutt-2003-v3	0.489	0.797	0.776	0.541	0.765	0.785	0.015
Pomeroy-2002-v2	0.416	0.520	0.515	0.414	0.491	0.471	0.417
Ramaswamy-2001	0.117	0.169	0.127	0.352	0.203	0.115	0.059
Singh-2002	0.023	0.026	0.026	0.002	0.027	0.024	0.024
Su-2001	0.423	0.511	0.467	0.347	0.468	0.464	0.423
West-2001	0.332	0.375	0.374	0.057	0.380	0.368	0.305
Yeoh-2002-v1	0.495	0.790	0.849	-0.002	0.774	0.839	0.017
Yeoh-2002-v2	0.160	0.205	0.199	0.175	0.228	0.199	0.033
Alizadeh-2000-v1	0.116	0.144	0.171	0.035	0.135	0.144	0.130
Alizadeh-2000-v2	0.573	0.591	0.760	0.413	0.442	0.703	0.510
Alizadeh-2000-v3	0.378	0.387	0.400	0.426	0.336	0.374	0.380
Bredel-2005	0.270	0.325	0.337	0.275	0.304	0.364	0.274
Garber-2001	0.171	0.213	0.199	0.089	0.097	0.182	0.169
Khan-2001	0.390	0.380	0.364	0.426	0.385	0.376	0.398
Lapointe-2004-v1	0.101	0.152	0.159	0.057	0.120	0.094	0.116
Lapointe-2004-v2	0.088	0.089	0.100	0.039	0.080	0.073	0.082
Liang-2005	0.161	0.157	0.157	0.105	0.157	0.157	0.146
Risinger-2003	0.100	0.132	0.105	0.183	0.124	0.102	0.099
Tomlins-2006-v1	0.289	0.315	0.317	0.264	0.304	0.312	0.284
Tomlins-2006-v2	0.187	0.202	0.200	0.168	0.207	0.195	0.183
final average rank	-	**2.066**	2.616	4.950	3.233	3.600	4.533

Table B10: Normalized mutual information results on cancer genomics datasets

Data set	Average	NMF	CONS	HGPA	MCLA	DICLENS	OKKC
Armstrong-2002-v1	0.333	0.371	0.372	0.026	0.370	0.363	0.328
Armstrong-2002-v2	0.570	0.688	0.674	0.551	0.668	0.678	0.552
Bhattacharjee-2001	0.386	0.408	0.426	0.285	0.329	0.403	0.387
Chowdary-2006	0.202	0.183	0.142	0.001	0.142	0.142	0.485
Dyrskjot-2003	0.463	0.548	0.537	0.482	0.519	0.537	0.469
Golub-1999-v1	0.568	0.591	0.591	0.013	0.591	0.590	0.529
Golub-1999-v2	0.535	0.660	0.621	0.454	0.637	0.634	0.481
Laiho-2007	0.163	0.191	0.192	0.122	0.191	0.167	0.158
Nutt-2003-v1	0.437	0.501	0.503	0.444	0.471	0.491	0.402
Nutt-2003-v2	0.155	0.139	0.095	0.145	0.135	0.102	0.309
Nutt-2003-v3	0.465	0.743	0.727	0.570	0.715	0.725	0.054
Pomeroy-2002-v2	0.560	0.627	0.623	0.552	0.596	0.591	0.555
Ramaswamy-2001	0.477	0.538	0.541	0.536	0.456	0.532	0.292
Singh-2002	0.044	0.047	0.047	0.008	0.048	0.044	0.045
Su-2001	0.604	0.649	0.645	0.554	0.614	0.643	0.594
West-2001	0.283	0.309	0.309	0.059	0.312	0.307	0.257
Yeoh-2002-v1	0.468	0.707	0.769	0.001	0.689	0.741	0.146
Yeoh-2002-v2	0.293	0.353	0.371	0.270	0.353	0.391	0.088
Alizadeh-2000-v1	0.110	0.147	0.161	0.044	0.122	0.142	0.120
Alizadeh-2000-v2	0.675	0.668	0.748	0.602	0.613	0.734	0.639
Alizadeh-2000-v3	0.547	0.566	0.583	0.614	0.507	0.556	0.549
Bredel-2005	0.317	0.349	0.359	0.311	0.351	0.366	0.323
Garber-2001	0.184	0.196	0.165	0.165	0.157	0.157	0.177
Khan-2001	0.560	0.586	0.576	0.540	0.574	0.584	0.571
Lapointe-2004-v1	0.108	0.158	0.174	0.081	0.131	0.113	0.122
Lapointe-2004-v2	0.121	0.125	0.140	0.079	0.115	0.109	0.110
Liang-2005	0.363	0.355	0.355	0.274	0.355	0.355	0.351
Risinger-2003	0.280	0.300	0.289	0.300	0.282	0.277	0.284
Tomlins-2006-v1	0.444	0.463	0.476	0.414	0.447	0.467	0.444
Tomlins-2006-v2	0.283	0.299	0.301	0.233	0.285	0.282	0.278
final average rank	-	**2.133**	**2.133**	5.333	3.616	3.383	4.400

Table B11: Silhouette index results on cancer genomics datasets

Data set	Average	NMF	CONS	HGPA	MCLA	DICLENS	OKKC
Armstrong-2002-v1	0.295	0.302	0.301	0.010	0.302	0.305	0.277
Armstrong-2002-v2	0.223	0.233	0.227	0.179	0.239	0.234	0.203
Bhattacharjee-2001	0.221	0.246	0.247	0.050	0.176	0.223	0.214
Chowdary-2006	0.823	0.816	0.915	0.036	0.915	0.915	0.231
Dyrskjot-2003	0.254	0.274	0.263	0.154	0.236	0.272	0.222
Golub-1999-v1	0.361	0.375	0.375	0.002	0.375	0.375	0.291
Golub-1999-v2	0.165	0.157	0.152	0.084	0.154	0.155	0.137
Laiho-2007	0.294	0.313	0.312	0.157	0.313	0.313	0.253
Nutt-2003-v1	0.191	0.181	0.297	0.080	0.134	0.286	0.129
Nutt-2003-v2	0.339	0.380	0.405	0.124	0.397	0.410	0.229
Nutt-2003-v3	0.227	0.325	0.320	0.242	0.314	0.335	0.019
Pomeroy-2002-v2	0.192	0.219	0.208	0.181	0.196	0.211	0.176
Ramaswamy-2001	0.082	0.126	0.139	-0.162	-0.169	0.113	-0.106
Singh-2002	0.500	0.624	0.624	0.059	0.620	0.625	0.592
Su-2001	0.144	0.188	0.192	-0.005	0.134	0.160	0.145
West-2001	0.210	0.212	0.212	0.030	0.207	0.217	0.195
Yeoh-2002-v1	0.333	0.392	0.386	0.000	0.394	0.395	0.228
Yeoh-2002-v2	0.028	0.058	0.062	-0.099	-0.043	0.039	0.024
Alizadeh-2000-v1	0.044	0.049	0.048	0.010	0.045	0.047	0.041
Alizadeh-2000-v2	0.184	0.168	0.158	0.127	0.151	0.159	0.170
Alizadeh-2000-v3	0.135	0.141	0.126	0.150	0.125	0.134	0.138
Bredel-2005	0.071	0.058	0.049	0.059	0.056	0.035	0.070
Garber-2001	0.085	0.071	0.160	0.007	0.030	0.082	0.103
Khan-2001	0.143	0.195	0.194	0.066	0.178	0.188	0.148
Lapointe-2004-v1	0.386	0.420	0.454	0.194	0.369	0.460	0.402
Lapointe-2004-v2	0.219	0.251	0.271	-0.003	0.254	0.257	0.246
Liang-2005	0.487	0.532	0.532	0.347	0.532	0.532	0.485
Risinger-2003	0.101	0.124	0.138	0.038	0.104	0.108	0.107
Tomlins-2006-v1	0.081	0.072	0.090	0.012	0.059	0.065	0.080
Tomlins-2006-v2	0.087	0.073	0.078	0.002	0.056	0.036	0.087
final average rank	-	**2.316**	2.500	5.600	3.960	2.816	3.700

Table B12: Isolation index results on cancer genomics datasets

Data set	Average	NMF	CONS	HGPA	MCLA	DICLENS	OKKC
Armstrong-2002-v1	0.884	0.890	0.891	0.509	0.890	0.891	0.858
Armstrong-2002-v2	0.786	0.813	0.830	0.731	0.814	0.820	0.759
Bhattacharjee-2001	0.772	0.803	0.817	0.606	0.739	0.815	0.762
Chowdary-2006	0.954	0.954	0.958	0.474	0.958	0.958	0.908
Dyrskjot-2003	0.706	0.721	0.731	0.650	0.699	0.743	0.674
Golub-1999-v1	0.889	0.901	0.901	0.496	0.901	0.901	0.842
Golub-1999-v2	0.735	0.751	0.763	0.626	0.734	0.780	0.689
Laiho-2007	0.800	0.802	0.806	0.654	0.803	0.808	0.763
Nutt-2003-v1	0.623	0.624	0.696	0.501	0.576	0.694	0.541
Nutt-2003-v2	0.773	0.799	0.821	0.636	0.797	0.847	0.639
Nutt-2003-v3	0.679	0.740	0.735	0.694	0.729	0.760	0.508
Pomeroy-2002-v2	0.523	0.561	0.578	0.514	0.536	0.596	0.512
Ramaswamy-2001	0.602	0.608	0.686	0.396	0.357	0.700	0.314
Singh-2002	0.945	0.968	0.968	0.553	0.965	0.967	0.944
Su-2001	0.722	0.747	0.801	0.544	0.702	0.809	0.696
West-2001	0.790	0.790	0.799	0.542	0.791	0.794	0.752
Yeoh-2002-v1	0.777	0.877	0.892	0.488	0.873	0.905	0.530
Yeoh-2002-v2	0.545	0.601	0.700	0.369	0.533	0.774	0.302
Alizadeh-2000-v1	0.633	0.674	0.659	0.524	0.636	0.669	0.629
Alizadeh-2000-v2	0.772	0.781	0.859	0.658	0.732	0.842	0.748
Alizadeh-2000-v3	0.668	0.687	0.738	0.672	0.651	0.750	0.672
Bredel-2005	0.640	0.687	0.701	0.603	0.667	0.742	0.638
Garber-2001	0.543	0.527	0.627	0.384	0.449	0.678	0.555
Khan-2001	0.771	0.809	0.810	0.663	0.793	0.810	0.773
Lapointe-2004-v1	0.896	0.892	0.920	0.649	0.861	0.947	0.890
Lapointe-2004-v2	0.812	0.777	0.806	0.489	0.787	0.824	0.802
Liang-2005	0.925	0.935	0.935	0.752	0.935	0.935	0.925
Risinger-2003	0.543	0.537	0.584	0.437	0.508	0.589	0.540
Tomlins-2006-v1	0.636	0.635	0.683	0.560	0.614	0.688	0.635
Tomlins-2006-v2	0.643	0.648	0.673	0.500	0.624	0.692	0.644
final average rank	-	3.233	2.100	5.800	4.150	**1.383**	4.330

Appendix C: Produženi apstrakt na srpskom jeziku

Za analizu ogromnih količina složenih i heterogenih podataka koje su nam danas dostupne i koje se konstantno generišu potrebno je osmisliti nove algoritme za njihovu obradu i analizu. Mašinsko učenje ima centralnu ulogu u analizi podataka, pružajući mogućnost da se podaci grupišu, otkriju skrivene relacije i obuče modeli za predikciju. U eri velikih i složenih podataka integrativni pristupi u mašinskom učenju su posebno značajni. Integrativni pristupi se oslanjaju na više algoritama, ulaznih parametara, koriste različite izvore podataka, a motivisani su željom za povećanjem tačnosti, robustnosti i stabilnosti.

C.1 Predmet i ciljevi istraživanja

Predmet istraživanja doktorske disertacije su algoritmi klasterovanja – grupisanja podataka i mogućnosti njihovog unapređenja integrativnim pristupom u cilju povećanja pouzdanosti, robustnosti na prisustvo šuma i ekstremnih vrednosti u podacima, omogućavanja fuzije podataka. Algoritmi klasterovanja [1] grupišu posmatrane objekte u klastere prema sličnosti u definisanom skupu obeležja. U najvažnije primene klasterovanja podataka ubrajamo: pružanje uvida u strukturu podataka, detekciju anomalija, generisanje hipoteza, otkrivanje znanja, kompresiju podataka predstavljanjem svakog klastera prototipom, filogenetsku analiza. Jedan od najjednostavnijih i najčešće korišćenih algoritama je K-means (metoda K srednjih vrednosti). Nakon 50 godina od njegovog nastanka, predloženo je mnoštvo novih algoritama ali je tema klasterovanja i dalje aktuelna. Novi istraživački pravci su: integrativno klasterovanje [2], polu-nadgledano klasterovanje [3], klasterovanje velikih podataka [4].

Klasterovanje podataka je nenadgledani tip mašinskog učenja čiji rezultat u velikoj meri zavisi od definisanih parametara, odabrane mere sličnosti ili rastojanja, inicijalizacije. Tako isti algoritam može na svom izlazu dati različite klastere. Iz toga proizilazi pitanje ponovljivosti rezultata, procene kvaliteta dobijenih klastera i odabira konačnog razultata. Integrativnim pristupom se navedeni

problemi rešavaju formiranjem ansambla rezultata klasterovanja i primenom algoritma koji na osnovu združenih informacija u ansamblu određuje konačan rezultat. Ansambl se kreira višestrukim izvršavanjem pojedinačnih klasterovanja. Pri tome se osnovni algoritam pokreće sa različitim parametrima, incijalizacijama, podskupovima obeležja ili se koristi više različitih osnovnih algoritama. Predmet istraživanja doktorske disertacije je mogućnost primene metode nenegativne faktorizacije matrice u analizi dobijenog anasambla rezultata pojedinačnih klasterovanja.

Primena nenagativne faktorizacije za integrativno klasterovanje podataka inicijalno predložena radom [68] nije detaljno istražena, a naročito njene mogućnosti u integraciji podataka u bioinformatici. Dalje unapređenje je moguće ostvariti regularizacijom postupka faktorizacije koja omogućava definisanje ograničenja i uvođenje predznanja i tako razvoj polu-nadgledanih algoritama mašinskog učenja.

Osnovni ciljevi istraživanja obuhvaćeni disertacijom su:

1. Razvoj algoritama za integrativno klasterovanje koji se zasnivaju na primeni nenegativne faktorizacije matrice.

2. Implementacija softverskog modula koji obuhvata predložene algoritme i ostale najznačajnije algoritme integrativnog klasterovanja.

3. Evaluacija na podacima tipičnim za mašinsko učenje i sintetičkim podacima.

4. Primena na probleme sa područja bioinformatike.

Različite interne i eksterne mere kvaliteta klasterovanja su korišćene u disertaciji. Odabrane mere kvaliteta merene su pre i posle integracije kako bi se detaljno ispitao doprinos integracije. Istraživanjem su takođe obuhvaćeni koncepti fuzije različitih izvora podataka kao i ugrađivanja domenskog predznanja uvođenjem regularizacije u postupak integracije.

C.2 Primene u bioinformatici

Razvijeni algoritmi primenjeni su u rešavanju konkretnih problema sa područja bioinformatike, odabrane zbog eksponencijalnog rasta količine genomskih podataka i njihove velike heterogenosti. U bioinformatici na raspolaganju su nam podaci u obliku sekvenci, ekspresija, interakcija, ontologija. Heterogenost bioinformatičkih podataka ilustrovana je na slici 1.1, gde je 6 gena predstavljeno u obliku DNK sekvenci, merenih ekspresija, odgovarajućih proteinskih interakcija, kao i delom genske ontologije kojem izdvojeni geni pripadaju.

Za analizu podataka u bioinformatici često se koriste algoritmi klasterovanja. Zbog raznolikosti podataka i problema većeg broja obeležja od broja uzoraka

značajno se smanjuje pouzdanost klasterovanja i dalja mogućnost otkrivanja znanja. Rezultati se mogu poboljšati integrativnim pristupima u klasterovanju. Postupci integrativnog klasterovanja razvijeni u okviru disertacije primenjeni su u domenima genomike raka [22], funkcionalne genomike [23] i metagenomike [26]. Genomika raka ima za cilj da rasvetli molekularne osnove bolesti raka. U javnim biomedicinskim bazama podataka, danas su dostupni podaci o ekspresijama hiljade gena pacijenata obolelih od različitih podtipova raka. Grupisanje pacijenata na osnovu ekspresija gena i traganje za obeležjima koja opisuju podtipove bolesti su važni koraci koji vode ka preciznoj medicini. Funkcionalna genomika teži da utvrdi funkcije gena, odnosno, proteina. Računarski pristupi u predikciji funkcije gena pružaju priliku da se postave hipoteze, usmere istraživanja i ubrzaju postupci otkrivanja funkcionalnih uloga gena. Domen metagenomike obuhvata istraživanja mikroba na genetskom nivou. Mikrobi se izučavaju kao zajednice, a prvi korak u analizi je utvrđivanje vrsta koje su prisutne u uzorku, dok se dalje radi na određivanju njihovih funkcija. Nove tehnologije sekvenciranja omogućile su dobijanje detaljnih genomskih informacija uzoraka uzetih direktno iz prirodnih okruženja i time donele revolucionarne promene u analizi mikroba koji žive u ljudskom organizmu, vodi, zemljištu. Milioni sekvenci dobijeni iz mikrobioloških uzoraka moraju se procesirati naprednim tehnikama analize podataka uz korišćenje značajnih računarskih resursa. Klasterovanje podataka se koristi u svim navedenim oblastima i stoga je od velike važnosti da se algoritmi dalje razvijaju i unapređuju kako bi mogli da se koriste na sve većim količinama izrazito heterogenih podataka.

C.3 Integrativno klasterovanje primenom nenegativne faktorizacije matrice

Integrativnim pristupima u klasterovanju podataka teži se rešavanju problema koji su karakteristični za individulno klasterovanje: određivanje broja klastera, inicijalizacija, dilema oko izbora parametara, mera sličnosti. Na području integrativnog klasterovanja među prvim predloženim algoritmima bili su konsenzus klasterovanje [49] i algoritmi zasnovani na hipergrafovima HGPA (*HyperGraph Partitioning Algorithm*) i MCLA (*Meta-Clustering Algorithm*) [50]. Noviji algoritmi integrativnog klasterovanja su DICLENS (*Divisive Clustering Ensemble*) [51] i OKKC (*Optimized Kernel K-means*) [52]. DICLENS koristi minimalno razapinjuće stablo za reprezentaciju veza između klastera, a kod OKKC se integracija vrši na nivou kernel matrica. Metode koje su predložene i ispitane u ovoj disertaciji zasnivaju se na nenegativnoj faktorizacije matrice (eng. *Nonnegative matrix factorization NMF*) [54].

Nenegativnom faktorizacijom matrice se ulazna matrica $R \in \mathbb{R}^{m \times n}$ aproksimira proizvodom matrica $W \in \mathbb{R}^{m \times k}$ i $H \in \mathbb{R}^{k \times n}$ ($R \approx WH$) postavljanjem dodatnog

uslova o nenegativnosti elemenata u W i H. Izbor unutrašnje dimenzije k, matričnog proizvoda WH zavisi od konkretnog problema koji se rešava. Obično se bira da je $k < min(m, n)$ čime se radi aproksimacija inicijalne matrice faktorima manjeg ranga što omogućava otkrivanje latentnih (skrivenih) struktura u podacima. U zavisnosti kako su podaci predstavljeni u inicijalnoj matrici (po kolonama ili vrstama), faktori matrice sadrže vektore baze i koeficijente za njihovo kombinovanje.

Nenegativna faktorizacija matrice je značajna za oblast mašinskog učenja zbog osobine da pruža uvid u delove i njihovu povezanost unutar celine. Intuitivan prikaz ove osobine dat je na slici 2.4, gde je kolekcija slika lica faktorisana NMF postupkom. Dobijeni vektori baze predstavljaju delove lica, a matrica koeficijenata sadrži informacije kako se delovi kombinuju za potrebe rekonstrukcije inicijalnih slika lica. Primenom NMF postignuti su dobri rezultati u bioinformatici u rešavanju problema redukcije skupa obeležja [63], smanjenja nesigurnosti u podacima [64], klasterovanja uzoraka tumora i proteinskih interakcija [65], analize biomedicinskih dokumenata i genskih ekspresija [67].

NMF dozvoljava samo aditivne kombinacije vektora baze i na ovaj način dekompozicijom se stvara reprezentacija inicijalnih podataka u obliku delovicelina što je značajno za analizu podataka. Cilj NMF je da minimizuje grešku reprezentacije matrice R proizvodom W i H. Predložene su različite funkcije cilja, a među njima se najviše koriste kvadratna greška:

$$\frac{1}{2}\|R - WH\|_F^2 = \frac{1}{2}\sum_{ij}(R_{ij} - [WH]_{ij})^2, \tag{1}$$

gde F označava Frobenijusovu normu, ili *Kullback-Leibler* divergencija:

$$D_{KL}(R\|WH) = \sum_{ij}(R_{ij}\ln\frac{R_{ij}}{[WH]_{ij}} - R_{ij} + [WH]_{ij}). \tag{2}$$

Nenegativna faktorizacija se može izvršiti pomoću postupka multiplikativnog ažuriranja matrica W i H [69]. Vrednosti u W i H, se prvo inicijalizuju nekom od predloženih metoda, a potom se iterativno ažuriraju množenjem trenutnih vrednosti sa faktorima koji zavise od tačnosti aproksimacije $R \approx WH$:

$$H \leftarrow H. * ((W^T R)./(W^T WH)), \tag{3}$$

$$W \leftarrow W. * ((RH^T)./(WHH^T)). \tag{4}$$

Navedenim multiplikativnim ažuriranjem aproksimacija R se monotono poboljšava

po Frobenijusovoj normi. Faktorizacija se izvršava u iterativnim koracima sve dok greška aproksimacije ne bude ispod predefinisanog praga ili se dostigne maksimalni broj iteracija.

Algoritmi za integrativno klasterovanje predloženi u okviru disertacije zasnivaju se na opisanom NMF postupku i obuhvataju tri koraka: formiranje ansambla, faktorizaciju matrice i izdvajanje klastera pomoću informacija sadržanih u faktorisanoj matrici ansambla. Za potrebe formiranja ansambla, osnovni algoritam se izvršava više puta. U zavisnosti od karakteristika ulaznih podataka, u disertaciji su kao osnovni algoritmi klasterovanja korišćeni: kernel K-means, algoritmi zasnovani na particionisanju grafova i spektralno klasterovanje. Pojedinačni rezultati se integrišu u binarnu matricu $R = \{0,1\}^{m\times n}$, gde vrsta označava klaster, a kolona objekat (m je ukupan broj klastera nastao višestrukim izvršavanjem osnovnog algoritma, a n je broj objekata koji se klasteruje). Matrica R se potom faktoriše na dve matice $W \in \mathbb{R}^{m\times k}$ and $H \in \mathbb{R}^{k\times n}$ koje daju novu reprezentaciju pripadnosti objekata klasterima. W sadrži koeficijente, a H vektore baze koji se mogu interpretirati kao (kontinualna) pripadnost objekata klasterima nakon faktorizacije. Parametar k je rang faktorizacije i jednak je broju klastera na izlazu.

Ulazna ansambl matrica je retka i dobijeni faktori nakon postupka faktorizacije su retke matrice, stoga nije bilo potrebno uvoditi dodatne regularizatore retkosti matrica. Inicijalizacija početnih faktora vršena je postupkom nenegativne dvostruke singularne dekompozicije (eng. *Non-negative Double Singular Value Decomposition* (NNDSVD)) [56], koja ubrzava konvergenciju i daje jedinstveno rešenje. Na slici 3.1 ilustrovana je faktorizacija matrice na malom primeru. Ulazna binarna matrica R sa informacijama o pripadnosti objekata klasterima, dimenzija 4×7, faktoriše se na matrice dimenzija 4×3 i 3×7. Matrica W je predstavljena i u obliku povezanosti inicijalnih klastera sa vektorima baza iz kojih će se kreirati novi klasteri.

Postupak izdvajanja finalnih klastera obuhvata postavljanje praga na vrednosti pripadnosti objekata klasteru u vektorima baze. Pre postavljanja praga potrebno je izvršiti skaliranje vrednosti u matricama. Naime, rezultat nenegativne faktorizacije nije jedinstven. Može postojati nesingularna matrica $D \in \mathbb{R}^{k\times k}$ koja zadovoljava $WD \geq 0$ i $D^{-1}H \geq 0$, tako da se faktorizacija može predstaviti i kao:

$$WH = WDD^{-1}H = W^*H^* \tag{5}$$

Matrica D može izvršiti skaliranja i permutacije vrednosti. Iz tog razloga je potrebno adekvatno preskalirati vrednosti u kolonama težinskih koeficijenata matrice W i redovima matrice H (vektorima baza) pre definisanja praga. Za taj

postupak koristili smo dijagonalne matrice D_W i D_H:

$$D_W = diag([max(w_{:,1}), max(w_{:,2})...max(w_{:,k})]) \tag{6}$$

$$D_H = diag([max(h_{1,:}), max(h_{2,:})...max(h_{k,:})]) \tag{7}$$

Deo procedure iz binarne faktorizacije matrice [70] pogodan je za preskaliranje W i H. Za matrice D_W i D_H važi:

$$D_W = D_W^{1/2} D_W^{1/2} \qquad D_H = D_H^{1/2} D_H^{1/2} \tag{8}$$

$$D_W^{-1} = D_W^{-1/2} D_W^{-1/2} \qquad D_H^{-1} = D_H^{-1/2} D_H^{-1/2} \tag{9}$$

$$\begin{aligned} \tilde{R} &= WH = (WD_W^{-1})(D_W D_H)(D_H^{-1} H) \\ &= (WD_W^{-1/2} D_H^{1/2})(D_H^{-1/2} D_W^{1/2} H) \end{aligned} \tag{10}$$

U jednačini (5) matrica D se može izraziti kao $D = D_W^{-1/2} D_H^{1/2}$:

$$W^* = WD_W^{-1/2} D_H^{1/2} \quad H^* = D_H^{-1/2} D_W^{1/2} H \tag{11}$$

Transformacije $W \to W^*$ i $H \to H^*$ ne menjaju proizvod WH, a obezbeđuju uporedivost i adekvatnu interpretaciju vrednosti težinskih koeficijenata, kao i vrednosti u vektorima baze. Elementi W i H se preskaliraju na sledeći način:

$$w_{i,k}^* = w_{i,k} \sqrt{\frac{max(h_{k,:})}{max(w_{:,k})}} = \frac{w_{i,k}}{max(w_{:,k})} \sqrt{max(w_{:,k}) max(h_{k,:})} \tag{12}$$

$$h_{k,j}^* = h_{k,j} \sqrt{\frac{max(w_{:,k})}{max(h_{k,:})}} = \frac{h_{k,j}}{max(h_{k,:})} \sqrt{max(h_{k,:}) max(w_{:,k})} \tag{13}$$

Pripadnost objekata novim klasterima se određuje pomoću odgovarajućih vrednosti u W^* and H^*. Izlazni klasteri mogu biti preklapajući (fazi) ili nepreklapajući (ekskluzivni). Kod preklapajućih klastera objekat može pripadati u više klastera, dok kod nepreklapajućih svaki objekat je pridružen samo jednom klasteru za koji mu je pripadnost bila najveća. Algoritam za izdvajanje klastera opisan je pseudo kodom:

Algoritam: Ekstrakcija klastera

Ulaz: $W^* \in \mathbb{R}^{M \times K}$, $H^* \in \mathbb{R}^{K \times N}$,
objekti $[o_1, o_2, \ldots o_N]$, $T_r = 0.5$
Izlaz: klasteri C = $[c_1, c_2, \ldots c_K]$
$WSUM^* \leftarrow$ suma po kolonama W^*
for $k \leftarrow 1 : K$ **do**
  for $j \leftarrow 1 : N$ **do**
    if tip klasterovanja = preklapajući **then**
      if $h^*_{k,j} \geq T_r$ **then**
        proširi klaster c_k objektom o_j
      end if
    else
      if $(h^*_{k,j} \geq T_r)$ & $(h^*_{k,j} * wsum^*_k = \max(h^*_{k',j} * wsum^*_{k'}$, za $k' \leftarrow 1 : K))$ **then**
        proširi klaster c_k objektom o_j
      end if
    end if
  end for
end for

Nepreklapajući klasteri se popunjavaju na osnovu koeficijenata pripadnosti iz vektora u matrici H^*. Kod ekskluzivnog klasterovanja vrši se dodatno rangiranje koje uzima u obzir važnost objekta u klasteru (koristi se informacija iz H^*) i važnost celukupnog klastera (koristi se informacija iz W^*).

Integrativno klasterovanje zasnovano na NMF može se dopuniti postupcima regularizacije. Cilj regularizacije je nametanje dodatnih ograničenja na matrice W i H od kojih se može dalje zahtevati retka, glatka reprezentacija ili zadovoljavanje nekih unapred definisanih relacija između elemenata u matricama. Takođe regularizacijom se mogu uvesti domenska predznanja značajna za problem koji se rešava. Ako se regularizacija vrši strukturom grafa, postupak se naziva GNMF (eng. *Graph Regularized NMF*). U strukturi grafa čvorovi označavaju objekte koji se klasteruju, a ivice veze između objekata, odnosno, dodatne informacije koje ulaze u postupak faktorizacije. Menja se algoritam za faktorizaciju matrice, dok deo za izdvajanje klastera nakon faktorizacije ostaje isti. U optimizacioni postupak za pronalaženje faktora dodatno se uvode sledeći elementi: matrica povezanosti grafa A, dijagonalna matrica D koja sumira A po kolonama, $D_{ii} = \sum_j A_{ij}$, Laplasijan L koji se dobija kao $L = D - W$, i parametar λ koji određuje jačinu regularizacije.

Kod GNMF postupka greška rekonstrukcije izražena Frobenijusovom normom ima dodatni član - trag proizvoda matrica u kojem učestvuje definisani Laplasijan, i izražena je formulom:

$$\|R - WH\|_F^2 = \sum_i \sum_j [R_{ij} - (WH)_{ij}]^2 + \lambda Tr(HLH^T) \tag{14}$$

Aproksimacija R faktorima W i H monotono se poboljšava po Frobenijusovoj normi korišćenjem postupka multiplikativnog ažuriranja. Ažuriranje matrica W i H kod GNMF algoritma se izvršava na sledeći način:

$$H \leftarrow H. * ((W^T R + \lambda HA)./(W^T WH + \lambda HD)), \tag{15}$$

$$W \leftarrow W. * ((RH^T)./(WHH^T)). \tag{16}$$

U poređenju sa postupkom ažuriranja matrica kod NMF algoritma, možemo primetiti da se kod GNMF menja samo formula za ažuriranje matrice H, dok izračunavanje matrice W ostaje isto.

C.4 Rezultati

Predloženi algoritmi integrativnog klasterovanja evaluirani su na raznovrsnim skupovima podataka i upoređeni sa relevantnim algoritmima sa tog područja. Podaci na kojima je vršena evaluacija obuhvataju:

1. Standardne skupove podataka sa „UCI Machine Learning“ repozitorijuma (20 setova podataka)
2. Sintetičke podacke (20 setova podataka)
3. Bioinformatičke podatke:
   - Genske ekspresije sa područja genomike raka [78] (30 setova podataka, gde svaki set sadrži $\approx$ hiljade obeležja–gena i labele–podtipove raka)
   - Heterogeni podaci model organizma *Saccharomyces cerevisiae* (kvasca): ekspresije [91], [93], proteinske sekvence, proteinske interakcije [110], fenotipski podaci sintetičkih genomskih nizova [92] (5 setova podataka iz kojih je analiziran zajednički podskup od $\approx$ 2000 gena/proteina)
   - 16S rRNK sekvence iz mikrobioloških uzoraka (1 set podataka uz odgovarajuće metapodatke [119])

U poglavljima 4, 5, 6 i 7 detaljno su opisani eksperimenti i predstavljeni dobijeni rezultati.

U četvrtom poglavlju predložena NMF metoda integrativnog klasterovanja za nepreklapajuće klastere je opsežno poređena sa 5 alternativnih algoritama. Evaluacija je izvršena pomoću dve interne i dve eksterne mere kvaliteta klasterovanja. Eksperimenti su obuhvatili tri grupe podataka: „UCI Machine Learning“ podatke, sintetičke podatke i genske ekspresije iz studija genomike raka. Integracija je rađena na različitim podskupovima obeležja i parametrima kernel funkcije u „kernel K-means“ metodi klasterovanja. Dodatno su analizirani setovi genskih ekspresija pacijenata obolelih od raka u kontekstu diverziteta/slaganja rezultata pojedinačnih klasterovanja. Rezultati poređenja metoda na svakom od skupova podataka kao i združeno na svim podacima prikazani su na slikama: 4.2, 4.3, 4.4 i 4.5. Detaljni rezultati dati su tabelama B1, B2, B3, B4, B5, B6, B7, B8, B9, B10, B11 i B12. Rezultati poređenja pokazali su da predloženo integrativno klasterovanje spada u grupu boljih algoritama. U objedinjenim rezulatatima na 70 skupova podataka NMF i CONS su se izdvojili po značajnosti od ostalih algoritama. Evaluacija NMF i dalje poređenje sa CONS algoritmom je u narednim poglavljima detaljno ispitano na podacima sa područja funkcionalne genomike i metagenomike.

U petom poglavlju metoda integrativnog klasterovanja podataka primenjena je na problem otkrivanja funkcionalno koherentnih grupa gena/proteina. Eksperimenti su vršeni na tri skupa podataka koji su značajni za otkrivanje funkcionalne povezanosti. Integracija je rađena po različitim izvorima podataka i merama sličnosti/rastojanja (Pirsonova korelacija, međusobna informacija i Euklidsko rastojanje). Evaluacija je izvršena nad genskom ontologijom koja predstavlja hijerarhijski organizovanu kategorizaciju funkcija gena. Intuitivnim grafičkim prikazima 5.3, 5.4, 5.6 predstavljeni su rezultati klasterovanja individualnim metodama i dobijeni rezultat nakon integracije. Rezultati jasno pokazuju da se integracijom dobijaju funkcionalno koherentnije grupe gena. U rešavanju ovog problema pokazalo se da je preklapajći tip klasterizacije, gde objekat može pripadati u više klastera, od posebne važnosti za biološke podatke. S obzirom na to da geni imaju više funkcija, ovaj tip klasterovanja je dao značajno bolje razultate. Poređenjem NMF i CONS metode utvrđeno je da NMF daje bolje rezultate.

Šesto poglavlje predstavlja rezultate dodatnih istraživanja na podacima iz petog poglavlja. Integracija je ovde obuhvatila različite izvore podataka i mere sličnosti, ali uz uvođenje regularizacije u postupak integracije, realizovane primenom GNMF metode. Graf struktura kojom se regularizuje postupak faktorizacije formirana je na osnovu dodatnih podataka: sličnosti između proteinskih sekvenci ili postojanja interakcije između proteina. Analizirana su oba tipa klasterovanja, preklapajući i nepreklapajući. Rezultati su upoređeni sa NMF pristupom i predstavljeni grafički, slike 6.2 i 6.1. Uvođenje regularizacije doprinelo je poboljšanju rezultata.

U sedmom poglavlju analiziran je skup podataka sa područja metagenomike. Podaci obuhvataju približno 69 miliona marker sekvenci - 16S rRNK dobijenih sekvenciranjem 1967 uzoraka. Važan korak u analizi ovih podataka je merenje diveriteta. Sekvence se prvo prema sličnosti grupišu u mikrobiološke zajednice, takozvane OTU (eng. Operational taxonomic units), a zatim se mere sličnosti ili razlike između uzoraka u prostoru detektovanih OTU. Diverzitet uzoraka po stukturi mikrobioloških zajednica može se izmeriti raznovrsnim merama ali postoji dilema koju meru izabrati. U disertaciji je predložena integracija rezultata klasterovanja uzoraka dobijenih primenom 24 različite mere diverziteta. Integracijom se postiže stabilan rezultat klasterovanja, na nivou najboljeg rezultata u ansamblu i rezultat je robustan na promene parametara. Rezultati su predstavljeni na slikama: 7.2, 7.3, 7.4, 7.5. Ovo je dodatno potvđeno eksperimentima na slučajno generisanim skupovima sačinjenih od 1000 poduzoraka inicijalnog skupa (slika 7.6). Za obrada podataka sa područja metagenomike potrebni su značajniji računarski resursi i uvođenje distribucije i paralelizacije u izvršavanje algoritama. Na kraju sedmog poglavlja razmotrene su mogućnosti izvršavanja algoritma na distribuiran način, gde se podaci prvo podele, zatim se na svakom delu vrši klasterovanje, a potom se rezultati spajaju za ceo skup podataka. Da bi moglo da se izvrši spajanje rezultata svaki objekat iz skupa mora biti bar u jednom podskupu i mora postojati preklapanje među podskupovima objekata. Eksperimenti su rađeni sa podskupovima od 500 uzoraka, $\approx \frac{1}{4}$ podataka, sa 5 do 25 različitih podskupova poduzoraka čiji rezultati će biti deo ansambla. Fuzija po merama diverziteta rađena je nivou podskupova. Rezultati su predstavljeni grafički na slici 7.7 i idu u prilog distribuiranog integrativnog klasterovanja. Mali broj podskupova je dovoljan da se nadmaši prosečan rezultat pojedinačnih klasterovanja. Broj podskupova potreban da se dostigne rezultat integrativnog klasterovanja na celom skupu podataka donosi uštede u vremenu izvršavanja.

C.5 Zaključak

Područje istraživanja doktorske disertacije spojilo je dva koncepta: integrativno klasterovanje i nenegativnu faktorizaciju matrice. Predložene metode zasnovane na nenegativnoj faktorizaciji matrice pružaju mogućnost fuzije podataka, mera sličnosti i/ili podskupova obeležja, kao i mogućnost uvođenja domenskog predznanja u formi regularizacije algoritma. Metode su uspešno implementirane i detaljno analizirane na raznovrsnim podacima sa UCI repozitorijuma i sintetičkim podacima koje se tipično koriste za evaluaciju novih algoritama i poređenje sa već postojećim metodama. Veći deo disertacije posvećen je primeni u domenu bioinformatike koja obiluje heterogenim podacima i brojnim izazovnim zadacima. Evaluacija je izvršena na podacima iz domena funkcionalne genomike, genomike raka i metagenomike.

Rezultati istraživanja potvrdili su hipotezu da integracija (podataka, parametara, objekata, obeležja i/ili uzoraka) može doprineti poboljšanju rezultata klasterovanja i umanjiti rizik od pogrešnog odabira jedne realizacije. Pokazano je da se integrativnim klasterovanjem postiže veća stabilnost konačnih klastera u poređenju sa individualnim klasterovanjem.

Dobijeni rezultati mogu pružiti i širu primenu jer se predloženi postupci integracije zasnivaju na matematičkoj metodi koja se lako može primeniti na novim problemima sa područja bioinformatike, ali na problemima iz drugih domena kao što su segmentacija slike u daljinskoj detekciji ili fuzija podataka u senzorskim mrežama. Buduća istraživanja biće posvećena naprednijim postupcima za kreiranje ansambla, novim vidovima regularizacije sa ciljem razvoja algoritama za polu-nadgledano klasterovanje, a posebna pažnja biće posvećena skaliranju algoritama kako bi se omogućila njihova primena na velikim podacima.

References

[1] P. Berkhin, "A survey of clustering data mining techniques," in *Grouping multidimensional data*. Springer, 2006, pp. 25–71. 1, 94

[2] J. Ghosh and A. Acharya, "Cluster ensembles," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 1, no. 4, pp. 305–315, 2011. 1, 94

[3] E. Bair, "Semi-supervised clustering methods," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 5, no. 5, pp. 349–361, 2013. 1, 94

[4] A. S. Shirkhorshidi, S. Aghabozorgi, T. Y. Wah, and T. Herawan, "Big data clustering: a review," in *International Conference on Computational Science and Its Applications*. Springer, 2014, pp. 707–720. 1, 94

[5] M. Shindler, A. Wong, and A. W. Meyerson, "Fast and accurate k-means for large datasets," in *Advances in neural information processing systems*, 2011, pp. 2375–2383. 1

[6] B. Hong-Tao, H. Li-Li, O. Dan-Tong, L. Zhan-shan, and L. He, "K-means on commodity GPUs with CUDA," in *Computer Science and Information Engineering, 2009 WRI World Congress on*, vol. 3. IEEE, 2009, pp. 651–655. 1

[7] R. Chitta, R. Jin, T. C. Havens, and A. K. Jain, "Approximate kernel k-means: Solution to large scale kernel clustering," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2011, pp. 895–903. 1

[8] A. Elgohary, A. K. Farahat, M. S. Kamel, and F. Karray, "Embed and conquer: Scalable embeddings for kernel k-means on mapreduce." in *SDM*. SIAM, 2014, pp. 425–433. 1

[9] U. Von Luxburg, R. C. Williamson, and I. Guyon, "Clustering: Science or art?" in *ICML Unsupervised and Transfer Learning*, 2012, pp. 65–80. 2

[10] Z. D. Stephens *et al.*, "Big data: astronomical or genomical?" *PLoS Biol*, vol. 13, no. 7, p. e1002195, 2015. 2

[11] F. S. Collins, M. Morgan, and A. Patrinos, "The human genome project: lessons from large-scale biology," *Science*, vol. 300, no. 5617, pp. 286–290, 2003. 2

[12] B. Berger, J. Peng, and M. Singh, "Computational solutions for omics data," *Nature Reviews Genetics*, vol. 14, no. 5, pp. 333–346, 2013. 3

[13] I. Inza, B. Calvo, R. Armañanzas, E. Bengoetxea, P. Larrañaga, and J. A. Lozano, "Machine learning: an indispensable tool in bioinformatics," in *Bioinformatics methods in clinical research*. Springer, 2010, pp. 25–48. 3

[14] H. Kashyap, H. A. Ahmed, N. Hoque, S. Roy, and D. K. Bhattacharyya, "Big data analytics in bioinformatics: A machine learning perspective," *arXiv preprint arXiv:1506.05101*, 2015. 3

[15] S. Pabinger *et al.*, "A survey of tools for variant analysis of next-generation genome sequencing data," *Briefings in bioinformatics*, vol. 15, no. 2, pp. 256–278, 2014. 3

[16] A. Brazma and J. Vilo, "Gene expression data analysis," *FEBS letters*, vol. 480, no. 1, pp. 17–24, 2000. 3

[17] A. V. Kossenkov and M. F. Ochs, "Matrix factorisation methods applied in microarray data analysis," *International journal of data mining and bioinformatics*, vol. 4, no. 1, pp. 72–90, 2010. 3

[18] T. Aittokallio and B. Schwikowski, "Graph-based methods for analysing networks in cell biology," *Briefings in bioinformatics*, vol. 7, no. 3, pp. 243–255, 2006. 4

[19] O. Bodenreider and R. Stevens, "Bio-ontologies: current trends and future directions," *Briefings in bioinformatics*, vol. 7, no. 3, pp. 256–274, 2006. 4

[20] M. Ashburner *et al.*, "Gene ontology: tool for the unification of biology," *Nature genetics*, vol. 25, no. 1, pp. 25–29, 2000. 4, 37, 40

[21] E. Gubb and R. Matthiesen, "Introduction to omics," *Bioinformatics Methods in Clinical Research*, pp. 1–23, 2010. 4

[22] J. Rung and A. Brazma, "Reuse of public genome-wide gene expression data," *Nature Reviews Genetics*, vol. 14, no. 2, pp. 89–99, 2013. 5, 96

[23] P. Radivojac *et al.*, "A large-scale evaluation of computational protein function prediction," *Nature methods*, vol. 10, no. 3, pp. 221–227, 2013. 5, 96

[24] P. Radivojac, "A (not so) quick introduction to protein function prediction," 2013. 5

[25] Y. Jiang *et al.*, "An expanded evaluation of protein function prediction methods shows an improvement in accuracy," *arXiv preprint arXiv:1601.00891*, 2016. 6

[26] A. Oulas *et al.*, "Metagenomics: tools and insights for analyzing next-generation sequencing data derived from biodiversity studies," *Bioinformatics and biology insights*, vol. 9, p. 75, 2015. 6, 96

[27] M. B. Scholz, C.-C. Lo, and P. S. Chain, "Next generation sequencing and bioinformatic bottlenecks: the current state of metagenomic data analysis," *Current opinion in biotechnology*, vol. 23, no. 1, pp. 9–15, 2012. 6

[28] J. Demšar, T. Curk, A. Erjavec, Č. Gorup, T. Hočevar, M. Milutinovič, M. Možina, M. Polajnar, M. Toplak, A. Starič *et al.*, "Orange: data mining toolbox in python," *The Journal of Machine Learning Research*, vol. 14, no. 1, pp. 2349–2353, 2013. 7

[29] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011. 7

[30] P. J. Cock *et al.*, "Biopython: freely available python tools for computational molecular biology and bioinformatics," *Bioinformatics*, vol. 25, no. 11, pp. 1422–1423, 2009. 7, 59

[31] J. G. Caporaso, J. Kuczynski, J. Stombaugh, K. Bittinger, F. D. Bushman, E. K. Costello, N. Fierer, A. G. Pena, J. K. Goodrich, J. I. Gordon *et al.*, "Qiime allows analysis of high-throughput community sequencing data," *Nature methods*, vol. 7, no. 5, pp. 335–336, 2010. 7, 65

[32] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM computing surveys (CSUR)*, vol. 31, no. 3, pp. 264–323, 1999. 8

[33] B. Sathya and R. Manavalan, "Image segmentation by clustering methods: performance analysis," *International Journal of Computer Applications*, vol. 29, no. 11, 2011. 8

[34] J.-J. Huang, G.-H. Tzeng, and C.-S. Ong, "Marketing segmentation using support vector clustering," *Expert systems with applications*, vol. 32, no. 2, pp. 313–317, 2007. 8

[35] G. J. Tsekouras, N. D. Hatziargyriou, and E. N. Dialynas, "Two-stage pattern recognition of load curves for classification of electricity customers," *Power Systems, IEEE Transactions on*, vol. 22, no. 3, pp. 1120–1128, 2007. 8

[36] R. Xu, D. Wunsch *et al.*, "Survey of clustering algorithms," *Neural Networks, IEEE Transactions on*, vol. 16, no. 3, pp. 645–678, 2005. 8

[37] F. Höppner, *Fuzzy cluster analysis: methods for classification, data analysis and image recognition*. John Wiley & Sons, 1999. 9

[38] A. K. Jain, "Data clustering: 50 years beyond k-means," *Pattern Recognition Letters*, vol. 14, no. 4, pp. 327–344, 2010. 9

[39] S. E. Schaeffer, "Graph clustering," *Computer science review*, vol. 1, no. 1, pp. 27–64, 2007. 10

[40] E. Rendón, I. M. Abundez, C. Gutierrez, S. D. Zagal, A. Arizmendi, E. M. Quiroz, and H. E. Arzate, "A comparison of internal and external cluster validation indexes," in *Proceedings of the 2011 American Conference, San Francisco, CA, USA*, vol. 29, 2011. 13

[41] L. Hubert and P. Arabie, "Comparing partitions," *Journal of classification*, vol. 2, no. 1, pp. 193–218, 1985. 13

[42] W. M. Rand, "Objective criteria for the evaluation of clustering methods," *Journal of the American Statistical association*, vol. 66, no. 336, pp. 846–850, 1971. 13

[43] B. E. Dom, "An information-theoretic external cluster-validity measure," in *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 2002, pp. 137–145. 13

[44] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of computational and applied mathematics*, vol. 20, pp. 53–65, 1987. 14

[45] E. J. Pauwels and G. Frederix, "Finding salient regions in images: nonparametric clustering for image segmentation and grouping," *Computer Vision and Image Understanding*, vol. 75, no. 1, pp. 73–85, 1999. 14

[46] G. Frederix and E. J. Pauwels, "Shape-invariant cluster validity indices," in *Advances in Data Mining*. Springer, 2005, pp. 96–105. 14

[47] L. I. Kuncheva and S. T. Hadjitodorov, "Using diversity in cluster ensembles," in *Proceedings of IEEE international conference on Systems, man and cybernetics*, vol. 2, 2004, pp. 1214–1219. 14

[48] S. Yu, B. De Moor, and Y. Moreau, "Clustering by heterogeneous data fusion: framework and applications," in *Proceedings of NIPS workshop on Learning with Multiple Sources*, 2009. 15, 53

[49] S. Monti, P. Tamayo, J. Mesirov, and T. Golub, "Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data," *Machine learning*, vol. 52, no. 1-2, pp. 91–118, 2003. 15, 37, 53, 64, 96

[50] A. Strehl and J. Ghosh, "Cluster ensembles – a knowledge reuse framework for combining multiple partitions," *The Journal of Machine Learning Research*, vol. 3, no. 1–2, pp. 583–617, 2003. 15, 96

[51] S. Mimaroglu and E. Aksehirli, "Divisive clustering ensemble with automatic cluster number," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 9, no. 2, pp. 408–420, 2012. 15, 34, 96

[52] S. Yu, L.-C. Tranchevent, X. Liu, W. Glanzel, J. A. Suykens, B. De Moor, and Y. Moreau, "Optimized data fusion for kernel k-means clustering," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 5, pp. 1031–1039, 2012. 15, 34, 96

[53] S. Brdar, V. Crnojevic, and B. Zupan, "Integrative clustering by nonnegative matrix factorization can reveal coherent functional groups from gene profile data," *Biomedical and Health Informatics, IEEE Journal of*, vol. 19, no. 2, pp. 698–708, 2015. 16, 64

[54] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999. 16, 17, 96

[55] C.-J. Lin, "Projected gradient methods for nonnegative matrix factorization," *Neural computation*, vol. 19, no. 10, pp. 2756–2779, 2007. 16

[56] C. Boutsidis and E. Gallopoulos, "Svd based initialization: A head start for nonnegative matrix factorization," *Pattern Recognition*, vol. 41, no. 4, pp. 1350–1362, 2008. 17, 20, 50, 98

[57] Y. Xue, C. S. Tong, Y. Chen, and W.-S. Chen, "Clustering-based initialization for non-negative matrix factorization," *Applied Mathematics and Computation*, vol. 205, no. 2, pp. 525–536, 2008. 17

[58] P. Paatero and U. Tapper, "Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values," *Environmetrics*, vol. 5, no. 2, pp. 111–126, 1994. 17

[59] Y.-X. Wang and Y.-J. Zhang, "Nonnegative matrix factorization: A comprehensive review," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 25, no. 6, pp. 1336–1353, 2013. 17

[60] K. Devarajan, "Nonnegative matrix factorization: an analytical and interpretive tool in computational biology," *PLoS computational biology*, vol. 4, no. 7, p. e1000029, 2008. 18

[61] A. Pascual-Montano, P. Carmona-Saez, M. Chagoyen, F. Tirado, J. M. Carazo, and R. D. Pascual-Marqui, "bionmf: a versatile tool for non-negative matrix factorization in biology," *BMC bioinformatics*, vol. 7, no. 1, p. 366, 2006. 18

[62] M. Žitnik and B. Zupan, "Nimfa: A python library for nonnegative matrix factorization," *The Journal of Machine Learning Research*, vol. 13, no. 1, pp. 849–853, 2012. 18

[63] W. Liu, K. Yuan, and D. Ye, "Reducing microarray data via nonnegative matrix factorization for visualization and clustering analysis," *Journal of biomedical informatics*, vol. 41, no. 4, pp. 602–606, 2008. 18, 97

[64] G. Wang, A. V. Kossenkov, and M. F. Ochs, "Ls-nmf: a modified non-negative matrix factorization algorithm utilizing uncertainty estimates," *BMC bioinformatics*, vol. 7, no. 1, p. 175, 2006. 18, 97

[65] C.-H. Zheng, D.-S. Huang, D. Zhang, and X.-Z. Kong, "Tumor clustering using nonnegative matrix factorization with gene selection," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 13, no. 4, pp. 599–607, 2009. 18, 97

[66] D. Greene, G. Cagney, N. Krogan, and P. Cunningham, "Ensemble non-negative matrix factorization methods for clustering protein–protein interactions," *Bioinformatics*, vol. 24, no. 15, pp. 1722–1728, 2008. 18

[67] M. Chagoyen, P. Carmona-Saez, H. Shatkay, J. M. Carazo, and A. Pascual-Montano, "Discovering semantic features in the literature: a foundation for building functional associations," *BMC bioinformatics*, vol. 7, no. 1, p. 41, 2006. 18, 97

[68] D. Greene and P. Cunningham, "A matrix factorization approach for integrating multiple data views," in *Machine Learning and Knowledge Discovery in Databases*. Springer, 2009, pp. 423–438. 19, 95

[69] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Advances in neural information processing systems*, 2001, pp. 556–562. 20, 97

[70] Z.-Y. Zhang, T. Li, C. Ding, X.-W. Ren, and X.-S. Zhang, "Binary matrix factorization for analyzing gene expression data," *Data Mining and Knowledge Discovery*, vol. 20, no. 1, pp. 28–52, 2010. 21, 99

[71] C. Wiwie, J. Baumbach, and R. Röttger, "Comparing the performance of biomedical clustering methods," *Nature methods*, vol. 12, no. 11, pp. 1033–1038, 2015. 24

[72] O. Arbelaitz, I. Gurrutxaga, J. Muguerza, J. M. PéRez, and I. Perona, "An extensive comparative study of cluster validity indices," *Pattern Recognition*, vol. 46, no. 1, pp. 243–256, 2013. 24

[73] M. Girolami, "Mercer kernel-based clustering in feature space," *Neural Networks, IEEE Transactions on*, vol. 13, no. 3, pp. 780–784, 2002. 24

[74] S. T. Hadjitodorov, L. I. Kuncheva, and L. P. Todorova, "Moderate diversity for better cluster ensembles," *Information Fusion*, vol. 7, no. 3, pp. 264–275, 2006. 25

[75] M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: http://archive.ics.uci.edu/ml 25

[76] A. Ultsch, "Clustering wih som: U* c," in *Proceedings of the 5th Workshop on Self-Organizing Maps*, vol. 2, 2005, pp. 75–82. 25

[77] F. Leisch and E. Dimitriadou, *mlbench: Machine Learning Benchmark Problems*, 2010, r package version 2.1-1. 25

[78] M. C. de Souto, I. G. Costa, D. S. de Araujo, T. B. Ludermir, and A. Schliep, "Clustering cancer gene expression data: a comparative study," *BMC bioinformatics*, vol. 9, no. 1, p. 497, 2008. 25, 101

[79] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *The Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006. 26

[80] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein, "Cluster analysis and display of genome-wide expression patterns," *Proceedings of the National Academy of Sciences*, vol. 95, no. 25, pp. 14 863–14 868, 1998. 36

[81] O. G. Troyanskaya, K. Dolinski, A. B. Owen, R. B. Altman, and D. Botstein, "A bayesian framework for combining heterogeneous data sources for gene function prediction (in saccharomyces cerevisiae)," *Proceedings of the National Academy of Sciences*, vol. 100, no. 14, pp. 8348–8353, 2003. 36

[82] O. G. Troyanskaya, "Putting microarrays in a context: integrated analysis of diverse biological data," *Briefings in bioinformatics*, vol. 6, no. 1, pp. 34–43, 2005. 36

[83] C. L. Myers, D. Robson, A. Wible, M. A. Hibbs, C. Chiriac, C. L. Theesfeld, K. Dolinski, and O. G. Troyanskaya, "Discovery of biological networks from diverse functional genomic data," *Genome biology*, vol. 6, no. 13, p. R114, 2005. 36

[84] A. Tanay, R. Sharan, M. Kupiec, and R. Shamir, "Revealing modularity and organization in the yeast molecular network by integrated analysis of highly heterogeneous genomewide data," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101, no. 9, pp. 2981–2986, 2004. 37

[85] A. J. Butte and I. S. Kohane, "Mutual information relevance networks: functional genomic clustering using pairwise entropy measurements," in *Pac Symp Biocomput*, vol. 5, 2000, pp. 418–429. 37, 38

[86] J. Ruan, A. K. Dean, and W. Zhang, "A general co-expression network-based approach to gene expression analysis: comparison and applications," *BMC systems biology*, vol. 4, no. 1, p. 8, 2010. 37

[87] A. A. Margolin, I. Nemenman, K. Basso, C. Wiggins, G. Stolovitzky, R. D. Favera, and A. Califano, "Aracne: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context," *BMC bioinformatics*, vol. 7, no. Suppl 1, p. S7, 2006. 37

[88] P. Jiang and M. Singh, "Spici: a fast clustering algorithm for large biological networks," *Bioinformatics*, vol. 26, no. 8, pp. 1105–1111, 2010. 37, 39

[89] A. J. Enright, S. Van Dongen, and C. A. Ouzounis, "An efficient algorithm for large-scale detection of protein families," *Nucleic acids research*, vol. 30, no. 7, pp. 1575–1584, 2002. 37, 39

[90] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *science*, vol. 315, no. 5814, pp. 972–976, 2007. 37, 39

[91] B. P. Tu, A. Kudlicki, M. Rowicka, and S. L. McKnight, "Logic of the yeast metabolic cycle: temporal compartmentalization of cellular processes," *Science*, vol. 310, no. 5751, pp. 1152–1158, 2005. 37, 101

[92] M. Costanzo, A. Baryshnikova, J. Bellay, Y. Kim, E. D. Spear, C. S. Sevier, H. Ding, J. L. Koh, K. Toufighi, S. Mostafavi *et al.*, "The genetic landscape of a cell," *Science*, vol. 327, no. 5964, pp. 425–431, 2010. 37, 101

[93] "The saccharomyces genome database (sgd)," September 2012. [Online]. Available: http://www.yeastgenome.org 37, 101

[94] L. Tari, C. Baral, and P. Dasgupta, "Understanding the global properties of functionally-related gene networks using the gene ontology." in *Pacific Symposium on Biocomputing*, vol. 10, 2005, pp. 209–220. 38

[95] V. Van Noort, B. Snel, and M. A. Huynen, "The yeast coexpression network has a small-world, scale-free architecture and can be explained by a simple model," *EMBO reports*, vol. 5, no. 3, pp. 280–284, 2004. 38

[96] Y. Assenov, F. Ramírez, S.-E. Schelhorn, T. Lengauer, and M. Albrecht, "Computing topological parameters of biological networks," *Bioinformatics*, vol. 24, no. 2, pp. 282–284, 2008. 38

[97] M. E. Smoot, K. Ono, J. Ruscheinski, P.-L. Wang, and T. Ideker, "Cytoscape 2.8: new features for data integration and network visualization," *Bioinformatics*, vol. 27, no. 3, pp. 431–432, 2011. 38

[98] J. Vlasblom and S. J. Wodak, "Markov clustering versus affinity propagation for the partitioning of protein interaction graphs," *BMC bioinformatics*, vol. 10, no. 1, p. 99, 2009. 39

[99] O. Garcia, C. Saveanu, M. Cline, M. Fromont-Racine, A. Jacquier, B. Schwikowski, and T. Aittokallio, "Golorize: a cytoscape plug-in for network visualization with gene ontology-based layout and coloring," *Bioinformatics*, vol. 23, no. 3, pp. 394–396, 2007. 41

[100] J.-H. Hung, T.-H. Yang, Z. Hu, Z. Weng, and C. DeLisi, "Gene set enrichment analysis: performance evaluation and usage guidelines," *Briefings in bioinformatics*, p. bbr049, 2011. 41, 56

[101] X.-F. Zhang, D.-Q. Dai, L. Ou-Yang, and M.-Y. Wu, "Exploring overlapping functional units with various structure in protein interaction networks," *PloS one*, vol. 7, no. 8, p. e43092, 2012. 53

[102] T. Nepusz, H. Yu, and A. Paccanaro, "Detecting overlapping protein complexes in protein-protein interaction networks," *Nature methods*, vol. 9, no. 5, pp. 471–472, 2012. 53

[103] A. Battle, E. Segal, and D. Koller, "Probabilistic discovery of overlapping cellular processes and their regulation," *Journal of Computational Biology*, vol. 12, no. 7, pp. 909–927, 2005. 53

[104] M. Deodhar and J. Ghosh, "Consensus clustering for detection of overlapping clusters in microarray data." in *ICDM Workshops*, 2006. 53

[105] P. Larrañaga, B. Calvo, R. Santana, C. Bielza, J. Galdiano, I. Inza, J. A. Lozano, R. Armañanzas, G. Santafé, A. Pérez *et al.*, "Machine learning in bioinformatics," *Briefings in bioinformatics*, vol. 7, no. 1, pp. 86–112, 2006. 55

[106] L. Taslaman and B. Nilsson, "A framework for regularized non-negative matrix factorization, with application to the analysis of gene expression data," *PLoS ONE*, vol. 7, p. 46331, 2012. 57

[107] M. Stražar, M. Žitnik, B. Zupan, J. Ule, and T. Curk, "Orthogonal matrix factorization enables integrative analysis of multiple rna binding proteins," *Bioinformatics*, vol. 32, no. 10, pp. 1527–1535, 2016. 57

[108] D. Cai, X. He, J. Han, and T. S. Huang, "Graph regularized nonnegative matrix factorization for data representation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 8, pp. 1548–1560, 2011. 57, 59, 60

[109] S. McGinnis and T. L. Madden, "Blast: at the core of a powerful and diverse set of sequence analysis tools," *Nucleic acids research*, vol. 32, no. suppl 2, pp. W20–W25, 2004. 59

[110] D. Szklarczyk, A. Franceschini, S. Wyder, K. Forslund, D. Heller, J. Huerta-Cepas, M. Simonovic, A. Roth, A. Santos, K. P. Tsafou *et al.*, "String v10: protein–protein interaction networks, integrated over the tree of life," *Nucleic acids research*, p. gku1003, 2014. 59, 101

[111] J. Zhou, Z. He, Y. Yang, Y. Deng, S. G. Tringe, and L. Alvarez-Cohen, "High-throughput metagenomic technologies for complex microbial community analysis: open and closed formats," *MBio*, vol. 6, no. 1, pp. e02 288–14, 2015. 63

[112] M. L. Z. Mendoza, T. Sicheritz-Pontén, and M. T. P. Gilbert, "Environmental genes and genomes: understanding the differences and challenges in the approaches and software for their analyses," *Briefings in bioinformatics*, p. bbv001, 2015. 63

[113] Y. He, J. G. Caporaso, X.-T. Jiang, H.-F. Sheng, S. M. Huse, J. R. Rideout, R. C. Edgar, E. Kopylova, W. A. Walters, R. Knight *et al.*, "Stability of operational taxonomic units: an important but neglected property for analyzing microbial diversity," *Microbiome*, vol. 3, no. 1, p. 1, 2015. 64

[114] J. Kuczynski, Z. Liu, C. Lozupone, D. McDonald, N. Fierer, and R. Knight, "Microbial community resemblance methods differ in their ability to detect biologically relevant patterns," *Nature methods*, vol. 7, no. 10, pp. 813–819, 2010. 64, 71

[115] O. Koren, D. Knights, A. Gonzalez, L. Waldron, N. Segata, R. Knight, C. Huttenhower, and R. E. Ley, "A guide to enterotypes across the human body: meta-analysis of microbial community structures in human microbiome datasets," *PLoS Comput Biol*, vol. 9, no. 1, p. e1002863, 2013. 64

[116] P. Yang, X. Su, L. Ou-Yang, H.-N. Chua, X.-L. Li, and K. Ning, "Microbial community pattern detection in human body habitats via ensemble clustering framework," *BMC systems biology*, vol. 8, no. Suppl 4, p. S7, 2014. 64

[117] P. Legendre and M. Cáceres, "Beta diversity as the variance of community data: dissimilarity coefficients and partitioning," *Ecology Letters*, vol. 16, no. 8, pp. 951–963, 2013. 64

[118] A. Y. Ng, M. I. Jordan, Y. Weiss *et al.*, "On spectral clustering: Analysis and an algorithm," *Advances in neural information processing systems*, vol. 2, pp. 849–856, 2002. 64

[119] J. G. Caporaso, C. L. Lauber, E. K. Costello, D. Berg-Lyons, A. Gonzalez, J. Stombaugh, D. Knights, P. Gajer, J. Ravel, N. Fierer *et al.*, "Moving pictures of the human microbiome," *Genome Biol*, vol. 12, no. 5, p. R50, 2011. 64, 101

[120] A. Wilke, J. Bischof, T. Harrison, T. Brettin, M. D'Souza, W. Gerlach, H. Matthews, T. Paczian, J. Wilkening, E. M. Glass *et al.*, "A restful api for accessing microbial community data for mg-rast," *PLoS Comput Biol*, vol. 11, no. 1, p. e1004008, 2015. 64

[121] J. R. Rideout, Y. He, J. A. Navas-Molina, W. A. Walters, L. K. Ursell, S. M. Gibbons, J. Chase, D. McDonald, A. Gonzalez, A. Robbins-Pianka *et al.*, "Subsampled open-reference clustering creates consistent, comprehensive otu definitions and scales to billions of sequences," *PeerJ*, vol. 2, p. e545, 2014. 65

[122] R. C. Edgar, "Search and clustering orders of magnitude faster than blast," *Bioinformatics*, vol. 26, no. 19, pp. 2460–2461, 2010. 65

[123] T. Z. DeSantis, P. Hugenholtz, N. Larsen, M. Rojas, E. L. Brodie, K. Keller, T. Huber, D. Dalevi, P. Hu, and G. L. Andersen, "Greengenes, a chimera-checked 16s rrna gene database and workbench compatible with arb," *Applied and environmental microbiology*, vol. 72, no. 7, pp. 5069–5072, 2006. 67

[124] A. Rosenberg and J. Hirschberg, "V-measure: A conditional entropy-based external cluster evaluation measure." in *EMNLP-CoNLL*, vol. 7, 2007, pp. 410–420. 68

[125] S. Wagner and D. Wagner, *Comparing clusterings: an overview*. Universität Karlsruhe, Fakultät für Informatik Karlsruhe, 2007. 68

[126] B. Minaei-Bidgoli, H. Parvin, H. Alinejad-Rokny, H. Alizadeh, and W. F. Punch, "Effects of resampling method and adaptation on clustering ensemble efficacy," *Artificial Intelligence Review*, vol. 41, no. 1, pp. 27–48, 2014. 74

[127] J. Ghosh, A. Strehl, and S. Merugu, "A consensus framework for integrating distributed clusterings under limited knowledge sharing," in *Proc. NSF Workshop on Next Generation Data Mining*, 2002, pp. 99–108. 74

[128] A. Zimek and J. Vreeken, "The blind men and the elephant: On meeting the problem of multiple truths in data from clustering and pattern mining perspectives," *Machine Learning*, vol. 98, no. 1-2, pp. 121–155, 2015. 78

[129] R. Liao, Y. Zhang, J. Guan, and S. Zhou, "Cloudnmf: a mapreduce implementation of nonnegative matrix factorization for large-scale biological datasets," *Genomics, proteomics & bioinformatics*, vol. 12, no. 1, pp. 48–51, 2014. 78

[130] C. P. Chen and C.-Y. Zhang, "Data-intensive applications, challenges, techniques and technologies: A survey on big data," *Information Sciences*, vol. 275, pp. 314–347, 2014. 78