A Network Component Analysis based Divide and Conquer Method for Transcriptional Regulatory Network Analysis

Thesis by
Sachin Prabhu Haladi Ramanatha

Submitted for the degree of
Doctor of Philosophy

The University of Sheffield
Faculty of Engineering
Department of Automatic Control and Systems Engineering
Sheffield, England, United Kingdom

September, 2018

Acknowledgements

I would like to thank my supervisor Dr. Hua-Liang Wei for trusting my abilities and giving me freedom to shape my research. I also thank Prof. Dave Kelly, Prof. Jeff Green, Prof. Mike Holcombe and Dr. Hao Bai for their support and guidance during the initial months of my PhD.

Let us sacrifice our today so that our children can have a better tomorrow.
— A P J Abdul Kalam

I cannot thank my parents enough for their support and encouragement without which this would not have been possible.

I thank Prof. Koshy George for encouraging me to conduct independent research and empowering me through a well-tailored training.

I thank my friends Bernardo, Cristian, Matei, Roberto, Jinny, Sanket, Veerendra, Richard, Mike, Barun, Sina, Vel, Dinesh, Gopi and Ravi for making my years in Sheffield memorable.

I thank Pinaki Bhattacharya for his valuable advice and support, and my colleagues at Insigneo for providing me with a friendly and supportive environment which made thesis writing less stressful.

I thank University of Sheffield International College and Nottingham Trent International College for providing me with opportunities to teach various modules to international students.

I thank my wife, Swathi, for bringing joy in my life and supporting me through thick and thin over the past few stressful months.

Abstract

A Network Component Analysis based Divide and Conquer Method for Transcriptional Regulatory Network Analysis

Understanding gene regulation has played a major role in several biomedical applications ranging from cancer studies to genetic engineering. Transcriptional regulatory networks (TRN) have been studied extensively to understand rules of interactions between transcription factor (TF) and genes that constitute gene regulation. In a particular type of gene expression data modelling problem, only gene expression profiles and regulatory patterns are available. Regulatory patterns or networks are binary matrices that indicate connections between genes and TFs. A TRN modelling method must simultaneously estimate regulatory strengths and concentrations of TFs. Therefore, TRN modelling problem is a structure-constrained matrix factorisation (SCMF) problem.

In this thesis, among various available TRN modelling algorithms, Network Component Analysis (NCA) is chosen for further investigation. Several methods that extend NCA are proposed in literature. However, the following fundamental issues in NCA theory have remained unresolved

1. a method to test feasibility of NCA problem does not exist
2. a method to solve NCA problem with an infeasible start does not exist

One of the major contributions in this thesis is a method to test NCA feasibility. It is made possible for the first time in relevant literature to test a dataset-network pair for NCA feasibility before applying NCA. This is done by translating NCA rank conditions on posteriori variables to rank conditions on a priori available dataset-network pair. In this process, it is shown that binary rank of a regulatory pattern is important to define NCA feasibility.

Another major contribution in thesis is a divide and conquer method that computes unique solutions corresponding to NCA infeasible dataset-network pairs. Techniques that extend NCA to solve SCMF problems with an infeasible start proposed

in literature are shown to be inaccurate and limited in applicability. In this thesis, a bipartite matching based approach is developed to decompose an infeasible NCA network into a set of full-rank factorisable sub-networks. A solution corresponding to the original network is obtained as a convex combination of matrix factors corresponding to identified sub-networks. It is shown that the resulting matrix factors for the whole network are unique up to two scaling factors if all data subset-sub-network pairs are NCA feasible.

Contents

List of Figures

List of Algorithms

List of Tables

Nomenclature

3DNCA	Three-step Divide and Conquer Network Component Analysis
ChIP	Chromatin Immuno-Precipitation
ICA	Independent Component Analysis
ISNCA	Iterative Sub-Network Component Analysis
MLE	Maximum Likelihood Estimation
NCA	Network Component Analysis
NMF	Non-negative Matrix Factorisation
ODE	Ordinary Differential Equation
PCA	Principal Component Analysis
SCA	Sparse Component Analysis
SCMF	Structure Constrained Matrix Factorisation
SVD	Singular Value Decomposition
TF	Transcription Factor
TMM	Trimmed Mean of M-values
TRN	Transcriptional Regulatory Network

Chapter 1

Introduction

1.1 Background

Cells are one of the complex systems that have been studied extensively [4]. Several thousands of datasets related to cell biology, collectively known as multi-omics data, can be found in research areas dedicated to understanding different mechanisms of a cell. A sub-topic that focuses on genes and associated research questions is called genomics. Genes are segments of DNA that encode a specific function of biological significance. Gene expression is a process where a genes produce messenger RNAs (mRNA) which in turn produce protein compounds [5]. Gene expression data contains a measure of concentrations of genes in a population of living cells in a controlled experiment. Modelling gene expression data is a topic that has gained a great amount of interest over the past decades. Gene network modelling provides valuable insight into disease mechanisms and aids drug discovery [6]. However, extracting meaningful models from gene expression profiles is challenging.

Genomics is a popular research area. More than a thousand articles were published between years 2005 and 2012 reporting studies solely dedicated to identifying biological pathways related diseases in humans [7]. A wide variety of mathematical problems arises from different aspects of genomics [8]. A particular area of research that combines knowledge from genomics and proteomics (are of study related to protein networks) is called as transcriptomics [4]. Transcriptional regulatory network (TRN) models are an integral part of transcriptomics studies. Large scale experimental data analysis has shown that transcriptional regulation plays an important role in understanding cellular mechanisms [9].

Transcriptional regulation is a process where the expression of a gene is controlled by protein compounds facilitated by RNA polymerase [5]. Structures of compounds that constitute transcriptional regulation are well studied [10]. Tran-

scription factors (TF) are protein compounds that regulate gene expression. An ordered series of gene expression is crucial for adaptation of a cell to varying environmental conditions. Disrupting the order of gene expression can result in several undesirable events such as cancer – uncontrolled cell multiplication. As a consequence, understanding transcriptional regulatory mechanism is of great importance in drug discovery, cancer treatment, and many more areas of study [11].

1.2 Transcriptional regulation

A brief discussion on transcriptional regulatory mechanism is presented here, a detailed description can be found in [12]. Structure of a regulated gene is illustrated in Fig. 1.1 and 1.2. The strip running from left to right is a strand of DNA. All genes related to a particular biological function are generally found next to each other on this strand. The number of genes and regulatory mechanisms vary for different biological functions. Two key regions involved in regulating expression of a gene are promoter and operator regions found right before genes on a DNA strand. Each gene produces a protien as shown in Fig. 1.1. Some of these proteins regulate expressions of other genes.

RNA polymerase binds to the promoter region of the gene(s) of interest. If the operator region of gene(s) is free, i.e., no protein compound is binding to that region, then RNA polymerase slides along the DNA to make RNA which then produces corresponding protein products, Fig. 1.1. These proteins are then used for various processes in the cell. Proteins that facilitate gene expression by helping RNA polymerase dock onto the promoter region are called promotors. On the other

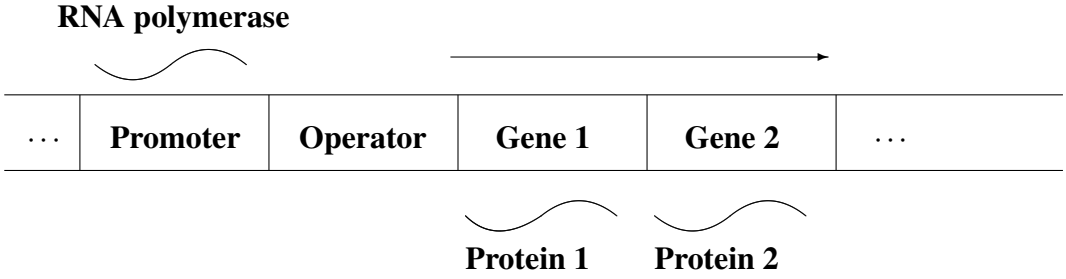

Figure 1.1: Gene expression – operator is free

hand, RNA polymerase is restricted by a protein compound called repressor or an inhibitor (black solid circle, Fig. 1.2) when it binds to the operator region of gene(s) and prevents polymerase from sliding. There are other mechanisms of repression,

but they are not important in the context of this thesis and hence, not discussed here. A gene can be positively regulated in which case one or more transcription

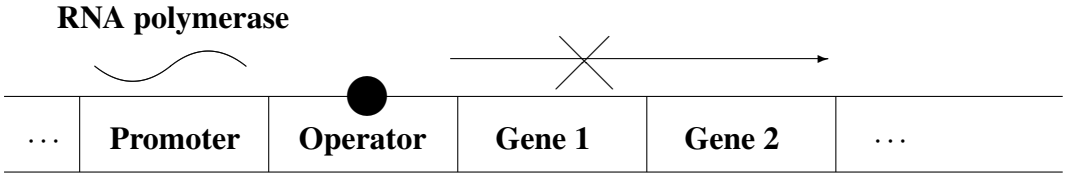

Figure 1.2: No gene expression – repressor binds to operator region

factors promote its expression. Negative regulation of gene occurs when one or more transcription factors act as repressors. Promotors are represented by a positive variable in mathematical models discussed in the next chapter, whereas repressors are represented by negative variables.

1.3 Transcriptonal regulatory networks

Chromatin Immuno-Precipitation (ChIP) sequencing technique [13] is used along with Trimmed Mean of M values (TMM) [14] normalisation technique to record gene expression data in biological experiments. The nature of recorded data directs the choice of two representations of TRNs introduced in this section – graphs and system of linear equations. A wide variety of biological systems are represented as bipartite graphs [15]. In a graph representation, interacting compounds such as genes and TFs are represented by vertices whereas the edges represent regulatory connections. Weights of the edges encode regulatory strength. A bipartite graph is shown in Fig. 1.3 (edited Fig. 3.6, [16]). In Fig. 1.3, $X = \{u_1,\ u_2,\ u_3\}$ represents a set of source signal vertices, TFs in the case of TRNs. $Y = \{v_1,\ v_2\}$ represents a set of data vertices, genes in TRNs. The edge weights can be collected in a matrix as

$$W = \begin{pmatrix} w_{11} & 0 & w_{13} \\ 0 & w_{22} & w_{23} \end{pmatrix} \tag{1.1}$$

Edge set $B(G)$ corresponding to the graph in Fig. 1.3 can be represented as

$$B(G) = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix} \tag{1.2}$$

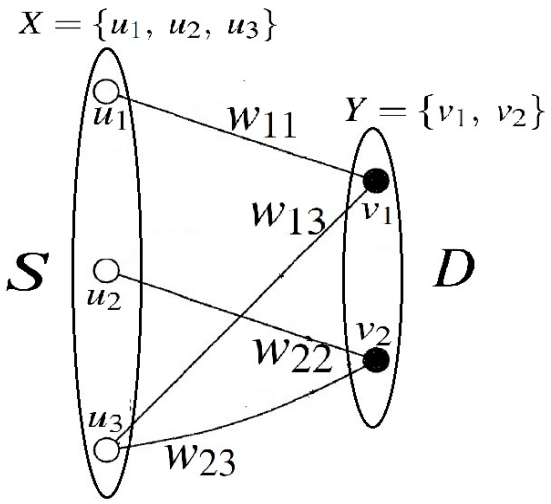

Figure 1.3: Undirected bipartite graph

In the context of graph theory, neighbours of a vertex v_i is a set of all vertices that are connected to v_i. For the example graph in Fig. 1.3, set of neighbours of v_1 is given as

$$\mathcal{N}(v_1) = \{u_1, u_3\}$$

TRNs can be represented using bipartite graphs similar to the one in Fig. 1.3 as $G = (V, B(G))$, where set of nodes are partitioned in to two disjoint sets as $V = X \cup Y, X \cap Y = \{\}$. X represents a set of TFs and Y represents a set of genes in the TRN. Edges $(i, j) \in B(G)$ connect nodes $u_i \in X$ and $v_j \in Y$. Gene expression data D corresponding to TRN in Fig. 1.3 can be represented as

$$D = WS \tag{1.3}$$

where, $D \in \mathbb{R}^{M \times N}$ and $S \in \mathbb{R}^{L \times N}$ represent N samples of M genes and L TFs, respectively. $W \in \mathbb{R}^{M \times L}$ similar to that in (1.1) contains regulatory strengths corresponding to the edges in $B(G)$ described in (1.2). It is easy to move from graph representation of TRN in Fig. 1.3 to a system of linear equations representation in (1.3). These two representations are used throughout this thesis. Generally, gene expression data D and regulatory pattern $B(G)$ are available whereas the regulatory strengths or weights in W and source signals or TF activities in S must be simultaneously estimated. Therefore, obtaining a TRN description of the form (1.3) requires factorising data matrix D while enforcing the structure described by $B(G)$ on W. This is problem is referred to as a structure-constrained matrix factorisation

(SCMF) problem [17] in the context of this thesis.

1.4 Network component analysis - a popular method

Several algorithms are proposed in literature to solve SCMF problem related to a TRN. Among such methods, Network Component Analysis (NCA) [2] has gained popuarity as evidenced by a citation report generated from web of science repository [3]. It is found that NCA [2] proposed in year 2003 has been cited 370 times between years 2004 and 2017 according to web of science [3]. A yearly publication trend is tabulated in table 1.1.

Year	no. of publications	Year	no. of publications
2004	9	2011	22
2005	25	2012	31
2006	34	2013	28
2007	36	2014	19
2008	33	2015	17
2009	39	2016	22
2010	42	2017	13

Table 1.1: number of publications citing NCA [2] by year according to we of science citation tool [3]

It looks like there has been a recent decline in interest. However, coverage of journals is limited with web of science [18]. Therefore, the numbers in table 1.1 are only representative of a continued interest in NCA, and not the actual level of its reception among whole research community. Nevertheless, NCA has been used in several studies to model underlying TRNs. In a recent article [19] published in current opinions in biotechnology, NCA has been pointed out as one of the data-integrative approaches that provides meaningful biological insights.

1.5 Overview of thesis

NCA is a popular choice for modelling TRNs as evidenced by literature. In addition to being a popular choice, NCA MATLAB toolbox [20] contains experimental datasets that can be used for testing TRN modelling methods. In this thesis, unresolved theoretical issues related to NCA are identified and fixed. The thesis is organised as follows

- **Chapter 2** – a review of different models of TRN and associated parameter estimation methods is presented. A particular model is chosen based on the data acquisition technology used in biological experiments. A particular class of SCMF methods applicable to TRNs are reviewed. One of these methods, NCA, is discussed in greater detail. Methods that describe NCA identifiability of a TRN and methods that extend NCA are discussed. Open theoretical problems in NCA are pointed out. A case is made for use of graph based methods to exploit better the sparse structure of TRNs. An experimental data available in literature is introduced. This dataset is used in other chapters to test the methods developed in this thesis.

- **Chapter 3** – numerical examples are used to test the validity of NCA rank conditions. It is found that the fundamental definition of NCA compatible networks is partially inaccurate. This inaccuracy stems from the fact that regulatory pattern of a TRN is a binary matrix whereas NCA rank conditions are defined over real numerical field. Importance of binary rank of a network is demonstrated and a fundamental relationship between real rank and binary rank is developed. It is demonstrated that size of a maximal matching in a graph is not necessarily equal to its rank. Possibility of estimating graph rank by applying three different graph operations is explored. A method to test uniqueness of NCA solutions is developed. This method is used to demonstrate the fact that full-ranked NCA solution are not necessarily uniqueness. Findings in this chapter are used as motivation for the next three chapters.

- **Chapter 4** – NCA rank conditions are amended to include binary rank of a network in order to accurately characterise NCA feasibility. A method to calculate a trivial solution when a network fails to satisfy binary rank based NCA feasibility conditions but satisfies original NCA rank conditions is presented. A method to characterise a subspace that can be avoided while looking for NCA solutions for this type of infeasibility is proposed. NCA rank condition on source signal matrix is translated to rank conditions on an augmented matrix containing dataset and structural constraints matrix. As dataset and network structure are known a priori, theorems developed in this chapter can be used to test a given dataset-network pair for NCA feasibility ahead of NCA implementation.

- **Chapter 5** – a method is developed to calculate binary rank of a bipartite graph using breadth-first search, graph reduction and minimum degree vertex reordering based matching. Graph theoretical conditions developed in this

chapter are shown to be equivalent to NCA feasibility conditions developed in chapter 4. Using these conditions, an algorithm is proposed to decompose a network into NCA feasible sub-networks. This algorithm is extended to limit the size of identified sub-networks based on number of available data points. This ensures that the resulting data subset-sub-network pairs are full-rank factorisable. It is demonstrated that full-rank factorisability of all sub-networks can be guaranteed even though their NCA feasibility is not guaranteed.

- **Chapter 6** – a possibility of obtaining unique matrix factors corresponding to NCA infeasible dataset-network pairs by combining NCA estimates of data subset-sub-network pairs is pointed out. Data subset-sub-networks are assumed to be identified using algorithms developed in chapter 5. A need to introduce more parameters whilst combining sub-network estimates is pointed out. Scaling issues related to adding more parameters are identified. These issues are addressed by reducing a multi-part mixing problem to a chain of two-part mixing problems. It is shown that matrix factors obtained by using such a divide and conquer method are unique up to two scaling factors. Strengths and weaknesses of the developed method are demonstrated with the help of a randomly chosen multi-part network and an experimental dataset-network pair available in literature.

- **Chapter 7** – contributions of this thesis are summarised. Scope and limitations of methods developed in this thesis are discussed. Few problems identified, but not addressed, throughout this thesis are discussed and objectives for future work are identified.

1.5.1 Contributions

- **Chapter 4** – an accurate method is proposed to test a priori available dataset-network pair for NCA feasibility before estimating corresponding matrix factors.

    1. theorem 2 accurately defines an NCA feasible network using its binary rank
    2. proposition 1 presents a linear algebra based method to tackle a particular type of NCA infeasibility
    3. theorem 3 defines NCA feasibility conditions based on rank conditions on a priori available dataset-network pair

- **Chapter 5** – a graph based method is developed to decompose NCA infeasible networks into a set of NCA feasible sub-networks.

    1. lemma 2 establishes a relationship between a particular type of graph matching and rank of a binary matrix
    2. lemma 3 establishes a relationship between a particular type of graph matching and binary rank of a binary matrix after removing all linearly dependent columns using algorithm 3
    3. theorem 4 defines graph theoretical NCA feasibility conditions for a given network
    4. algorithm 4 decomposes a given NCA infeasible network into multiple NCA feasible sub-networks
    5. algorithm 5 ensures full-rank factorisability by limiting sizes of sub-networks identified using algorithm 4 based on the number of available data points

- **Chapter 6** – results from chapters 4 and 5 are combined along with a mixing problem to uniquely solve a general SCMF problem.

    1. algorithm 6 combines results from previous chapters to estimate a unique convex combination of NCA solutions corresponding to identified full-rank factorisable sub-networks of a given network

1.5.2 Publications

- Conference papers based on results from chapter 5 –
  Prabhu, H. R. S., Wei, H. L. "Preprocessing Graphs for Network Inference Applications." 14^{th} *International Conference on Informatics in Control, Automation, and Robotics*, vol. 1, pp. 406 – 413, 2017.

- Conference paper based on a brief version of results from chapter 6 –
  Prabhu, H. R. S., Wei, H. L. "Incremental modelling of transcriptional regulatory networks." *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pp. 740 – 744, 2017.

- Book chapter based on results from section 4.2, chapter 5, and chapter 6 –
  Prabhu, H. R. S., Wei, H. L. "BNCA: full-rank fatorisable subgraphs based unique structure-constrained matrix factorisation", *Lecture Notes in Electrical Engineering*, In press.

Chapter 2

Literature review

2.1 Models of transcriptional regulation

Modelling principles and methods for a wide range of biological systems are presented in [21], where importance of recruiting simple models is explained. In this section, various models of TRN available in literature are considered followed by a discussion on two simple modelling methods.

As mRNA is a single strand copy of a gene, each gene maps to a unique mRNA. Expression of a gene is quantified by measuring concentration of corresponding mRNA in post-transcriptional phase [22]. Gene expression recorded in lab based experiments using Chromatin Immuno-Precipitation (ChIP) sequencing technique is useful in predicting biological pathways [13]. This is referred to as high-throughput data in relevant area of research called genomics. Various methods used to model different aspects of biological systems using ChIP sequences are reported in [23]. Among those methods, rule-based networks are used to model transcriptional regulation. Such rule-based networks are called Transcriptional Regulatory Networks (TRN) where genes are assumed to be regulated by proteins or transcription factors. In [24], different modelling approaches are compared in terms of how close the estimates are to biological reality, amount of data required, the degree of insight provided, and others. There, it is pointed out that differential equations based models are well-balanced with respect to different dimensions of comparison.

Ordinary differential equations [25] and log-linear models [26] are popular among biologists who study gene regulation. Stochastic differential equations with delays are used in [27] in order to capture the effects of noise and bistability in regulatory networks. However, in general, stable states in biological systems are well separated and hence, ordinary differential equations based models are sufficient to capture dynamics of a regulatory network [28].

2.1.1 Ordinary differential equations based model

Gene regulation can be interpreted as a nonlinear dynamical stable system described by a pair of differential equations as described in [29] as

$$\begin{aligned} \tfrac{d}{dt}T(t) &= AR(t) - BT(t) \\ \tfrac{d}{dt}R(t) &= f(T(t)) - CR(t) \end{aligned} \tag{2.1}$$

where, $f : \mathbb{R}^L \mapsto \mathbb{R}^M$ is a vector of M polynomials, $R(t) \in \mathbb{R}^M$ is a vector of mRNA concentrations, and $T(t) \in \mathbb{R}^L$ is a vector of protein concentration at time t. A, B and C are matrices of appropriate dimensions representing degradation factors of associated transcription factors and mRNAs. $T(t)$ is generally referred to as transcription factor activity. The model in (2.1) can be linearised at $T(0)$ to obtain a first-order linear autonomous state-space model as

$$\frac{d}{dt}X(t) = \Phi X(t) \tag{2.2}$$

where, $X(t)^T = \begin{pmatrix} T(t)^T & R(t)^T \end{pmatrix}$ are the states of the system and

$$\Phi = \begin{pmatrix} -B & A \\ E & -C \end{pmatrix}$$

is the state transition matrix, where $E = \frac{d}{dt}f(T(t))$ is evaluated at $T(0)$. As mentioned in the previous subsection, linearised state-space model in (2.2) provides a biologically sensible description of transcriptonal regulation [29]. A state-space based approach for simultaneous estimation of inputs and states is presented in [30]. This method works well with the model in (2.2), but it assumes availability of state transition matrix Φ. However, $T(t)$ is not measured, but inferred from gene expression data [31]. As a result, A and B are unknown and hence, the Φ is only partially known. In order to overcome this issue, a specific type of ODE model is considered in the next subsection.

2.1.2 Log-linear model

As mentioned in the previous subsection, as $T(t)$ is generally not measured. Therefore, an estimate of Φ in (2.2) cannot be obtained. However, as discussed in the previously, ODE based models offer a simple and realistic representation of transcriptional regulatory networks. Log-linear models introduced in [32] use a particular type of ODE to address this issue by imposing quasi-steady state assumption.

In [33], log-linear models are recommended for ease of parameter estimation among different modelling techniques that are reviewed therein.

A brief discussion on log-linear models is presented next. Log-linear models boil down the regulatory network modelling problem to a simple set of linear equations as shown in [34]. The following first-order dynamical model, generally referred to as power-law model is used in [34]

$$\frac{d}{dt}R_i(t) = \alpha_i \prod_j T_j(t)^{w_{ij}} - \beta_i R_i(t), \;\; 1 \leq i \leq M, \; 1 \leq j \leq L \tag{2.3}$$

where, $R_i(t)$ represents mRNA concentration at time t and β_i represents degradation factor corresponding to i-th gene. $T_j(t)$ represents concentration of j-th transcription factor at time t and α_i some constant based on underlying biochemistry. Exponent w_{ij} is the strength with which j-th transcription factor regulates i-th gene. The regulatory strengths $w_{ij} \in \mathbb{R}$ are such that

$$\begin{aligned} &w_{ij} > 0, \quad \text{if } T_j \text{ promotes } R_i \\ &w_{ij} < 0, \quad \text{if } T_j \text{ represses } R_i \\ &w_{ij} = 0, \quad \text{if } T_j \text{ has no effect on } R_i \end{aligned} \tag{2.4}$$

Generally, mRNA concentrations $R_i(t)$ are measured at discrete time instants k where the expression levels $R_i(k)$s are assumed to have achieved (quasi) steady-state [35]. With that assumption, the equation in (2.3) reduces to a difference equation given by

$$\beta_i R_i(k) = \alpha_i \prod_j T_j(k)^{w_{ij}} \tag{2.5}$$

Rearranging the equation and dividing throughout by $R_i(0)$ gives

$$\begin{aligned} &\frac{\beta_i R_i(k)}{\beta_i R_i(0)} = \frac{\alpha_i \prod_j T_j(k)^{w_{ij}}}{\alpha_i \prod_j T_j(0)^{w_{ij}}} \\ \implies \quad &\frac{\cancel{\beta_i} R_i(k)}{\cancel{\beta_i} R_i(0)} = \frac{\cancel{\alpha_i} \prod_j T_j(k)^{w_{ij}}}{\cancel{\alpha_i} \prod_j T_j(0)^{w_{ij}}} \end{aligned}$$

Thus, with quasi-steady state assumption in place, the equation in (2.3) can be rewritten as

$$\frac{R_i(k)}{R_i(0)} = \prod_j \left(\frac{T_j(k)}{T_j(0)} \right)^{w_{ij}} \tag{2.6}$$

For a regulatory network with L transcription factors T_js regulating M genes R_is, a log-linear relationship between R_is and T_js can be obtained by applying logarithm on both sides in (2.6) as

$$D(k) = WS(k) \tag{2.7}$$

where,

$$D(k) = \begin{pmatrix} \log \frac{R_1(k)}{R_1(0)} \\ \log \frac{R_2(k)}{R_2(0)} \\ \vdots \\ \log \frac{R_M(k)}{R_M(0)} \end{pmatrix}, \; S(k) = \begin{pmatrix} \log \frac{T_1(k)}{T_1(0)} \\ \log \frac{T_2(k)}{T_2(0)} \\ \vdots \\ \log \frac{T_L(k)}{T_L(0)} \end{pmatrix}$$

and

$$W = \begin{pmatrix} w_{11} & w_{12} & \cdots & w_{1L} \\ w_{21} & w_{22} & \cdots & w_{2L} \\ \vdots & \vdots & \ddots & \vdots \\ w_{M1} & w_{M2} & \cdots & w_{ML} \end{pmatrix}$$

In this thesis, matrix notations of the following form are used for all matrices

$$W(i,j) = w_{ij}, \; W(i,:) = \begin{pmatrix} w_{i,1} & \cdots & w_{iM} \end{pmatrix}, \; W(:,j) = \begin{pmatrix} w_{1j} \\ \vdots \\ w_{Lj} \end{pmatrix}$$

For an experiment where N samples of gene expression $D(k)$ are recorded, the system of equations in (2.7) can be written as a set of linear equations in (1.3) with

$$D = \begin{pmatrix} D_1(1) & D_1(2) & \cdots & D_1(k) & \cdots & D_1(N) \\ D_2(1) & D_2(2) & \cdots & D_2(k) & \cdots & D_2(N) \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ D_M(1) & D_M(2) & \cdots & D_M(k) & \cdots & D_M(N) \end{pmatrix}$$

and

$$S = \begin{pmatrix} S_1(1) & S_1(2) & \cdots & S_1(k) & \cdots & S_1(N) \\ S_2(1) & S_2(2) & \cdots & S_2(k) & \cdots & S_2(N) \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ S_L(1) & S_L(2) & \cdots & S_L(k) & \cdots & S_L(N) \end{pmatrix}$$

In the context of this thesis, variables D, W and S respectively represent experimental dataset, weights of the regulatory network and source signals or transcription factors. Equation (1.3) represents a hypothetical experiment with no noise. However, in reality, experimentally recorded mRNA concentrations are noisy in nature [27]. Noise can be factored into the model in (1.3) by adding a matrix $\Gamma \in \mathbb{R}^{M \times N}$ as

$$D = WS + \Gamma \tag{2.8}$$

where, Γ is generally assumed to be Gaussian in nature [36] and hence, maximum

likelihood estimates W and S can be obtained. With this assumption, system equation in (1.3) is used throughtout the rest of this thesis.

As mentioned earlier, transcription factor activities in S are not available from biological experiments. Therefore, both W and S in (1.3) must be estimated and hence, this is a matrix factorisation problem. ChIP sequencing technology can only reveal the pattern of regulation [31] – a binary connectivity pattern indicating regulatory relationship between transcription factors and genes. ChIP data cannot be used to quantify the strengths of these connections. Therefore, in practice, S is unknown and only the structure, a binary matrix $B(G)$, of W is known as

$$B(G)(i,j) = \begin{cases} 1 \text{ (on)}, & \text{if } T_j \text{ regulates } R_i \\ 0 \text{ (off)}, & \text{if } T_j \text{ has no effect on } R_i \end{cases} \tag{2.9}$$

Therefore, the problem of estimating parameters of the model in (1.3) is a structure-constrained matrix factorisation (SCMF) problem. A formal definition of SCMF problem is presented in the next section where a binary valued matrix $B(G)$ is used to impose structural constraints in (2.9).

2.2 Structure-Constrained Matrix Factorisation

The problem of Structure-Constrained Matrix Factorisation (SCMF) arises when one of the factors is sparse. Characterisation of sparsity depends on the target application. For example, intra-sample correlations are used in [37] to construct constraint matrices for image processing applications, graph-theoretical metrics are used in [38] for applications in neuroscience, mass-spectrometry and machine-learning techniques are used in [39] for protein interaction network mapping, correlation based clustering is used to identify data sub-matrices in [40]. Regardless of how the structural constraints are identified, the objective that follows is to extract useful information from the recorded data by solving a SCMF problem. In the context of TRNs, structural constraints are imposed by regulatory patterns. It is assumed in this thesis that regulatory patterns are fixed and corresponding TRNs strictly follow specified regulatory pattern.

In this section, mathematical formulation of SCMF is presented followed by a review of methods available in literature to solve a general SCMF problem and those used in the context of TRNs in particular. Consider a porblem where a data matrix $D \in \mathbb{R}^{M\times N}$ is to be factorised into a product of two full-rank matrices $W \in \mathbb{R}^{M\times L}$ and $S \in \mathbb{R}^{L\times N}$ as described in (1.3). In a general SCMF [17], one of the two

matrix factors is expected to follow a specific structure defined by a binary matrix $B(G) \in GF_2^{M\times L}$. Here, $GF_2 = [0,1]$ is the binary field over which $B(G)$ is defined. In other words, entries of $B(G)$ are either 0 or 1. Without loss of generality, structural constraints can be imposed on matrix factor W. Notation $B(G)$ is used to make room for a graph theoretical interpretation of structural constraints where G is a bipartite graph with rows and columns of $B(G)$ representing two disjoint set of vertices as described in section 1.3. Further discussion on graph theoretical interpretation is presented later in this chapter.

Consider a bijection $\phi : \mathbb{R}^{M\times L} \mapsto GF_2^{M\times L}$ defined as

$$\phi(W)(i,j) = \begin{cases} 1, \text{ if } W(i,j) \neq 0 \\ 0, \text{ otherwise} \end{cases} \tag{2.10}$$

Thus, the structural constraints on W can be written as

$$\phi(W) = B(G) \tag{2.11}$$

A SCMF problem can be defined as

$$\begin{aligned} \min_{W,S} \quad & J = \|D - WS\|_F^2 \\ \text{sub. to} \quad & \phi(W) = B(G) \end{aligned} \tag{2.12}$$

where, $\|.\|_F$ represents Frobenius norm. Frobenius norm [41] of a matrix $W \in \mathbb{R}^{M\times L}$ is defined as

$$\|W\|_F^2 = \sum_{i=1}^{M}\sum_{j=1}^{L} W(i,j)^2$$

Square of Frobenius norm is considered in SCMF problem (2.12) as a quadratic function has a global minimum.

2.2.1 General sparse matrix factorisation methods

Several techniques have been proposed to address the problem of biological network inference. Techniques applicable to gene expression data are reviewed in [42], [43], [44] and more recently in [45]. Dimensionality reduction methods such as Principal Component Analysis (PCA) and Singular Value Decomposition (SVD) are used to extract regulatory patterns from gene expression data [46]. However, these methods are not considered in this thesis as a regulatory pattern $B(G)$ is assumed to be known in SCMF problem (2.12). A Partial Least Squares (PLS) based method is

presented in [47] where S in (1.3) is decomposed into latent components. However, no structural constraints are explicitly imposed on W or S. Therefore, PLS based methods are also not considered. Methods in [17] and [48] impose different forms of sparsity constraints on W in the norm sense in order to minimise the number of non-zero entries in W. These methods do not respect structural constraints in $B(G)$ arising from underlying biology. These methods are not considered in this thesis.

Boolean logic based method to identify TRNs is presented in [49]. Primary goal of this and related methods available in literature is to identify the connectivity pattern $B(G)$, not to solve an SCMF problem of the form (2.12). Tensor based methods are proposed in [50, 51]. Tensor based TRN modelling problem is similar to the SCMF problem in (2.12), but additional information such as DNA methylation profiles are required. Such information cannot be made available by ChIP sequencing technology alone.

SCMF problem defined in (2.12) can be interpreted as a blind source separation problem [52] or that of generalised component analysis [53]. Independent Component Analysis (ICA), Non-negative Matrix Factorisation (NMF), Sparse Component Analysis (SCA), and their extensions are used to solve such problems [53]. Data integrative methods reviewed in [45] are frequently used in the context of gene expression data analysis. These techniques impose certain assumptions on the underlying system of equations in (1.3) as

- ICA based methods assume S_js in (1.3) are statistically independent and W is completely unknown [54]
- NMF based methods assume that the entries of all the matrices in (1.3) are non-negative [55]
- SCA based methods estimate W based on correlations in the recorded data D [56]
- Data integrative methods assume availability of $B(G)$ described in (2.11)

Practical considerations – each of the techniques identified above have been successfully applied in the context of TRN modelling. However, choice of a method to analyse data at hand must be based on the following factors identified in [45]

1. objective of the study
2. type of prior information being used
3. measurement techniques used in procuring data

As mentioned in section 1.5, methods used to model TRNs in bacterial cells based on defined regulatory patterns are being considered in this thesis. Going by this objective and the type of prior knowledge, ICA and SCA based methods can be discarded as those methods do not impose specified constraints on the structure of W. From (2.4), entries of W can be positive or negative. From (2.6), $T(k)$ is normalised with respect to its initial value $T(0)$. This implies that $S = \log \frac{T(k)}{T(0)}$ in (2.7) can be negative if $T(k) < T(0)$. Therefore, NMF cannot be directly used to solve SCMF in (2.12). Furthermore, NMF based methods might result in solutions stuck in multiple local-extrema [57]. NMF involves normalisation of W whereas solutions with normalised S are easier to interpret in the context of TRNs [58]. In addition to that, Trimmed Mean of M-values (TMM) [14] is a commonly used sampling schemes along with ChIP sequencing technique. TMM is favoured over other sampling procedures as it is good at avoiding false positives [59]. TMM normalises recorded gene expression data with respect to a chosen initial data-point and generates log-fold changes in gene expression relative to their initial value. This directly corresponds to the type of model chosen in section 2.1.2. Therefore, ICA, NMF or SCA are not directly applicable to ChIP sequence data based log-linear models. Based on these practical considerations, data integrative TRN modelling methods identified in [45] are considered for further discussions.

2.2.2 Data integrative methods

Data integrative methods in the context of TRN modelling are those that use a priori available information to solve SCMF problem in (2.12). As mentioned earlier, $B(G)$ is assumed to be known. Data integrative parameter estimation methods in [2], [60], [20], [61], [47] solve the original SCMF problem (2.12) whereas [62], [63], [64] employ modified version of log-linear model in (2.7). A new variable $es(i, j)$ is used to encode edge/connection strength between a given gene-TF pair such that entries of W in (1.3) are given as

$$W(i, j) = es(i, j)\tilde{w}(i, j) \tag{2.13}$$

where, $\tilde{w}(i, j)$ is a parameter to be estimated. This calls for modification of SCMF problem in (2.12). Equation in (2.13) replaces the constraint $\phi(W) = B(G)$ in (2.12). In [64] $es(i, j)$ is a real-valued variable used to quantify the degree of confidence in a gene-TF connection based on the number of times such a connection is evidenced in literature. On the other hand, $es(i, j)$ is binary-valued in [61], [62] and [63]. Data integrative TRN modelling methods can be broadly categorised as

1. optimisation based methods – [2], [60], [20], [64]
2. probabilistic methods – [61], [62], [63]
3. singular value based method – [65], [66]

Optimisation based methods – Network Component Analysis (NCA) introduced in [2] solves the original SCMF problem by exploiting the sparsity of W. NCA starts with a random choice of W and S and estimates a solution of the form (1.3). An iterative procedure is proposed in [2] where W is estimated assuming S is constant and vice versa at each iteration. Solutions are guaranteed to be unique up to a scaling factor. A method to identify largest NCA identifiable sub-network of a TRN is presented in [60]. NCA is extended in [20] to model TRNs in cases where the number of recorded data points is smaller than the number of source signals in S. Different mathematical functions are used to define S in [64] and a switching based method is proposed to solve a modified SCMF problem. All the optimisation based TRN modelling methods use mixed integer linear programming to solve SCMF problem. However, any other optimisation method discussed in [67] such as linear or quadratic programming can be used. A regression based approach is presented in [68] where log-linear model similar to that in (2.6), but without normalising with respect to initial measurement is used. However, this can be translated to an SCMF problem of the form (2.12).

Probabilistic methods – it was mentioned in section 2.1.1 that a state-space model cannot be used to model TRNs directly as protein concentrations are not measured. However, in [61] and [62], $S(1)$ is chosen from a set of prior distributions computed from D and the dynamics of S is assumed to form a first-order Markov chain. This allows use of a state-space model similar to that in (2.2). Separate SCMF problem of the form (2.12) is set up for each row of D in [61]. This allows for different set of matrix factors (W_i, S_i) corresponding to D_i. This idea is extended in [62] where a modified SCMF problem formulation is used. Both methods resort to Maximum Likelihood Estimation (MLE) to compute W and S. An open source MATLAB software package that implements the method in [62] is presented in [63]. A dynamic Bayesian network and MLE based approach is presented in [69]. Applicability of this model is limited as it assumes that values of degradation factors β_is in (2.3) are available.

Singular value based method – a semi-blind source separation technique is presented in [65] where for every column of $B(G)(:, j)$ in (2.11), two sub-matrices of

D in (1.3) are identified corresponding to 1s and 0s in the chosen column of $B(G)$. A generalised eigen value problem is solved to simultaneously diagonalise sub-matrices of D. Column $W(:, j)$ in (1.3) is set equal to eigen vector corresponding to the largest generalised eigen value. S in (1.3) is estimated as $W^{\dagger}D$, where † represents pseudo-inverse. This is a fast and direct method. However, as shown in [65], for some i, j, $W(i, j) \neq 0$ when $B(G)(i, j) = 0$. As a result, underlying biological constraints are not strictly respected. Therefore, this method is not considered in this thesis. A non-iterative technique is presented in [66] where SVD of correlation of noisy data D in (2.8) is used to obtain direct estimates of W and S. However, the method in [66] assumes that NCA rank conditions in [2] hold. As a result, NCA identifiability of a TRN is necessary for the method in [66].

There is no particular reason to choose or discard one of the methods identified in this subsection over the other. Strengths and weaknesses of all these methods are discussed in the articles in which they are proposed. However, NCA in [2] is chosen for further investigation in this thesis for two reasons

1. as pointed out in section 1.4, NCA is a popular and tested TRN modelling method

2. a few fundamental issues pertaining to NCA theory have gone unnoticed in relevant literature

The second reason is of higher priority and this thesis is dedicated to addressing issues identified in chapter 3.

2.3 Network Component Analysis

In this section, a brief overview of NCA [2] is presented. Various methods that extend NCA are discussed in section 2.3.2. A review of theoretical results in literature that characterise NCA identifiability is presented in section 2.3.3.

The cost function J in (2.12) must be minimised over W and S. NCA [2] is one of the optimisation based algorithms that solve the original problem (2.12). NCA is a two-step iterative approach where at each iteration J is minimised with respect to S with W constant and then with respect to W with S constant. It is shown in [2] that a solution obtained using NCA is unique up to a scaling factor if the solution satisfies the following assumptions

Assumption 1 *W has full column rank, i.e., $rk(W) = L$*

Assumption 2 *All reduced sub-matrices W_j obtained by removing every i-th row with $W(i,j) \neq 0$ have full column rank, i.e., $rk(W_j) = L-1$*

Assumption 3 *S has full row rank, i.e., $rk(S) = L$*

$M \geq L$ and $N \geq L$ are implied respectively by assumption 1 and 3. Therefore, the system of linear equations in (1.3) is assumed to be over-determined.

NCA algorithm proposed in [2] implements a two-step iterative procedure described as

Algorithm 1 NCA [2]

input: $D, B(G)$
Randomly choose W_0, S_0 which satisfy assumptions 1-3, set $k = 0$
while $\hat{\Gamma} = \|D - WS\|_F^2 > \bar{\Gamma}$ **do**
  step 1 – solve for S in (2.12) with W constant
  step 2 – solve for W in (2.12) with S constant
end while
outputs: W, S

for some arbitrarily small $\bar{\Gamma} > 0$. In algorithm 1, the original SCMF problem in (2.12) is solved iteratively in an alternating fashion. In this sense, this is a bilinear optimisation problem. A discussion on bilinear optimisation algorithm and its application to biological systems can be found in [70].

NCA implementation – optimisation problem in step 2 can be divided into M individual problems corresponding to M rows in W. This is a simple way to address the constraint imposed by $B(G)$ on structure of W as described in (2.10). In doing so, it is easy to consider the non-zero parameters in W alone, along with corresponding rows in D and S. This also reduces the number of optimisation variables in each of the M sub-problems. Another reason to divide the problem into M sub-problems is to facilitate the use of any optimisation algorithm or interior point method in [67] to solve the NCA problem. This extra step is necessary as the NCA problem looks at Frobenius norm whereas interior-point algorithms consider 2-norm. Equivalence between Frobenius and 2 norms can also be achieved by vectorisation approach [71].

2.3.1 Uniqueness and robustness of NCA solutions

In the context of this thesis, a network $B(G)$ is NCA identifiable if for a given dataset D corresponding NCA solutions (W,S) satisfy assumptions 1–3. For some

NCA identifiable network $B(G)$ and a dataset D, let two pairs of NCA solutions be given as

$$D = ({}^1W)({}^1S) \text{ and } D = ({}^2W)({}^2S)$$

it is shown in [2] that a diagonal matrix $\chi \in \mathbb{R}^{L\times L}$ can be computed such that

$$({}^2W) = ({}^1W)\chi, \;\; ({}^2S) = \chi^{-1}({}^1S) \tag{2.14}$$

The original NCA theorem [2] combines assumptions 1–3 with (2.14) as

Theorem 1 *Given matrices $D \in \mathbb{R}^{M\times N}$ and $B(G) \in GF_2^{M\times L}$ such that $L \leq N \leq M$, if there exist matrices $({}^1W), ({}^2W) \in \mathbb{R}^{M\times L}$ and $({}^1S), ({}^2S) \in \mathbb{R}^{L\times N}$ such that*

$$\phi(({}^1W)) = \phi(({}^2W)) = B(G),$$

$$D = ({}^1W)({}^1S) = ({}^2W)({}^2S)$$

and assumptions 1–3 are satisfied, then there exists a non-singular diagonal matrix χ as given in (2.14)

Proof of uniqueness is presented in [2]. However, existence of a solution or identifiability of a given network has not been characterised therein. These can only be checked after applying NCA.

Robustness of NCA has been addressed in [2]. Following application of NCA in algorithm 1 to obtain a solution (W,S), bootstrap technique [72] is used in [2] to generate replacement samples Γ^* of resulting error

$$\hat{\Gamma} = D - WS \tag{2.15}$$

A population of resampled data is then generated as $D^* = WS + \hat{\Gamma}^*$. Algorithm 1 is applied to each D^* in resampled population to obtain new estimates of W, S, and $\hat{\Gamma^*}$ in (2.15). The authors show that 95% confidence intervals achieved with bootstrap resampling technique are tight.This stands to validate the robustness in obtained estimates in the sense that the obtained estimates are consistent across multiple trials. However, accuracy of estimates of W and S can be validated only by experiments. As mentioned earlier S is not measured and hence, such a validation might be hard in practice. Experimental validation of NCA estimate of one of the transcription factor activities is presented in [1].

2.3.2 Extensions of NCA

An in-depth review of NCA and related factorisation techniques designed to solve the problem in (2.12) is available in [73]. NCA has continued to peak the interest of researchers, most recent extensions of NCA can be found in [40, 74, 75]. NCA based algorithms can be categorised as iterative and non-iterative approaches. Iterative approaches start from randomly chosen W and S as described in algorithm 1 whereas non-iterative techniques utilise linear algebraic tools to reduce computational complexity. Six different methods that extend NCA are briefly discussed here.

Generalized NCA [34]:
Generalized NCA (gNCA) extends NCA to resolve the issue of ill-conditioned solutions by adding a regularisation parameter λ that penalises S. In addition to that, the structure of S is constrained by requiring it to follow the pattern imposed by $B(S)$. Structural constraints on S are introduced to represent variability in experiments where one of the source signals might be turned off. This does not improve robustness or uniqueness of NCA solutions. The problem is redefined as:

$$\begin{aligned} \min_{W,S} \quad & \|D - WS\|_F^2 + \lambda \|S\|_F^2 \\ \text{subject to} \quad & \phi(W) = B(G) \\ & \phi(S) = B(S) \end{aligned}$$

Some matrix manipulation techniques are presented in [34] which allow simultaneous factorisation of different epochs or experimental data records of same TRN.

Fast NCA [76]:
Fast NCA provides a linear algebra based non-iterative solution to the NCA problem. Gene expression profiles in D are rearranged for every individual transcription factor $S(j,:)$ as

$$D = \begin{pmatrix} D_j \\ \hline D'_j \end{pmatrix}$$

where, D_j consists rows $D(i,:)$ for which $B(G)(i,j) = 1$ whereas D'_j is comprised of remaining rows of D regulated by transcription factor S_j. SVD decomposition of size m_j is applied to rearranged D, where m_j is equal to the number of rows in D_j. Resulting left eigen vectors are used to estimate W by imposing a sparsity constraint in terms of l_1 norm. S is estimated in a similar way without a sparsity constraint. This method improves speed of convergence. However, the original structural constraints $B(G)$ are not necessarily enforced directly, but via rearranging the dataset.

This means that the estimates of W may not strictly follow the constraints in $B(G)$

Robust NCA [77]:
Robust NCA(ROBNCA) address the problems created by statistical outliers in the observed data D. This is done by modifying the objective function J_k in (2.12)as

$$J_k = \|D - WS - O\|_F^2 + \lambda_0 \|O\|_0$$

where, O represents unknown outliers, $\|\cdot\|_0$ represents a l_0 norm that counts the number of non-zero entries in O, and λ_0 is a regularisation parameter. ROBNCA penalises the outliers and thereby reducing uncertainties in the obtained solutions.

Iterative Sub-NCA [74, 78]:
Many biological entities work in synergy to achieve a common objective. This gives rise to redundancy in processes which appears as linearly dependent rows in W or S in the case of TRNs. Iterative Sub-NCA (ISNCA) factors in such redundancies by isolating regulatory pattern in $B(G)$ into three parts – two independent parts that are regulated by two disjoint sets of transcription factors and a coregulated part. System of linear equations in (1.3) after such a decomposition is given as

$$\left(\begin{array}{c} D_U^{(1)} \\ \hline D_U^{(2)} \\ \hline D_C \end{array}\right) = \left(\begin{array}{c|c} W_U^{(1)} & 0 \\ \hline 0 & W_U^{(2)} \\ \hline W_C^{(1)} & W_C^{(2)} \end{array}\right) \left(\begin{array}{c} S^{(1)} \\ S^{(2)} \end{array}\right) \tag{2.16}$$

where, $D_U^{(1)}$ and $D_U^{(2)}$ are independently regulated respectively by $S^{(1)}$ and $S^{(2)}$ through $W_U^{(1)}$ and $W_U^{(2)}$. Common data D_C is regulated by both $S^{(1)}$ and $S^{(2)}$ through $W_C^{(1)}$ and $W^{(2)}$. Independent parts of W are estimated by applying conventional NCA. Common parts of W are randomly initialised and updated at every step such that error in data reconstruction is minimised. This approach breaks the problem down into three parts out of which two parts abide by the rank conditions of NCA and are solved using NCA. A random search approach is used to construct two independent parts shown in (2.16).

Local NCA [40]:
Global expression profile recorded in a dataset D is decomposed into multiple data subsets. Each data subset consists rows D that are highly correlated as obtained by k nearest neighbours method. These are called local data profiles. Data subset in each neighbourhood along with the corresponding sub-network of $B(G)$ is fac-

torised using NCA. A weighted sum of source signals corresponding to different profiles is obtained to estimate global source signal profile S. As the estimates of S are updated to estimate a global profile, estimates of W must also be updated. Therefore, a correction factor is introduced to reduce the errors in reconstructing local profiles. LNCA solution generates estimates of global profile of S, local connectivity patterns W, and correction factors. LNCA is used in [40] to combine data from different experiments in order to identify general regulatory mechanisms.

Sparse NCA [75]:
This method extends Fast NCA proposed in [76] by changing the way in which W is estimated. A change in computational method is introduced as the singular values computed in [76] are dependent on the number of samples available in a dataset D. The new iterative method to compute W introduced in [75] is shown to eliminate this problem. However, as in the case of Fast NCA, estimated W may not follow structural constraints in $B(G)$.

The goal in this thesis is not to improve computational efficiency of NCA as is the case with the methods identified here. Regardless of which method among those identified here is used, the issue of NCA identifiability persists as all these methods assume NCA identifiability of a given network. Developing accurate NCA identifiability conditions is one of the objectives in this thesis.

2.3.3 NCA identifiability - state-of-art

Characterising identifiability of complex systems related to network inference problems of the form in (2.12) is still a problem of interest for researchers [79]. NCA rank conditions in assumption 1–3 require a over-determined set of linear equations. However, number of data points available on different databases corresponding to individual experiments is lesser than the number of source signals or TFs [8,80,81]. As a consequence, SCMF problems related to TRNs are generally under-determined. Uniqueness of NCA solution corresponding to such datasets cannot be guaranteed. Therefore, there is a need to develop accurate NCA identifiability conditions that can be exploited to develop methods that solve under-determined SCMF problems.

As mentioned in section 2.3.3, a proof of uniqueness of NCA solutions is presented in [2], but no discussion is presented on existence of a solution or, a method to test uniqueness of NCA solutions. A simple method can be developed to test the uniqueness (2.14) of NCA-solutions whenever they are available. However, it is

desirable to check in advance if NCA is applicable to SCMF problem (2.12). This is not trivial as only D and $B(G)$ are known a priori whereas W and S are unknown. Therefore, assumptions 1–3 must be translated to equivalent assumptions on $B(G)$ and/or D.

Translations of assumptions 1 and 2 in terms of $B(G)$ are given in [60](lemma 1, page 292). The conditions therein are based on a graph theoretical interpretation of the underlying structural constraints matrix $B(G)$. These conditions essentially translate to

Assumption 4 *$B(G)$ has full column rank, i.e., $rk(B(G)) = L$*

Assumption 5 *All reduced sub-matrices $B(H_j)$ obtained by removing every i-th row with $B(G)(i, j) \neq 0$ have full column rank, i.e., $rk(B(H_j)) = L - 1$*

In assumptions 4 and 5, $H_j \subset G$ is the sub-graph corresponding to reduced sub-matrices W_j

$$\phi(W_j) = B(H_j)$$

A network $B(G)$ satisfying these assumptions is said to be NCA compatible/identifiable. An attempt is made in [60](theorem 1, page 293) to outline generalised identfiability criteria where NCA solutions to SCMF problem in (2.12) are argued to be unique up to a scaling factor X (2.14) if the corresponding graph $B(G)$ is NCA compatible and S has full row rank. Two major drawbacks of NCA identifiability conditions in [60] are

1. graph theoretical interpretation of NCA rank conditions on W are not used in any method proposed in the same article
2. S is argued to have full row rank as it is a part of assumptions, but no mathematical proof is provided to show if and when this argument is valid

A specific case of NCA incompatibility is addressed in [20] where $L > N$, i.e., number of source signals is more than the number of samples available. In this case, $rk(S) < L$ and hence, assumption 3 cannot be satisfied. A modification to assumption 3 is proposed in [20] as

Assumption 6 *Every sub-matrix $S_i, 1 \leq i \leq M$ of S formed by rows $S(j, 1 : N)$ such that $B(G)(i, j) = 1$ has full row rank*

In other words, every row $D(i, 1 : N)$ of data matrix is obtained by mixing linearly independent signals. This also implies that number of 1s on every row of $B(G)$ is less than N. A method to enforce this assumption in NCA is presented in [20]. A

proof is also provided in [20] in which it is shown that if χ (2.14) is a diagonal matrix then assumption 6 holds. However, converse is not proven. Therefore, there is no guarantee that enforcing this assumption ensures uniqueness of NCA solutions. Authors of [20] have made available a MATLAB toolbox for NCA along with several datasets to test its applicability. It is shown in next chapter that NCA solutions of one of those datasets are not unique. Therefore, NCA uniqueness conditions proposed in [20] are incomplete.

Another attempt to characterise NCA identifiability is made in [82] where graph theoretical property of matching is used. A network, in the context of NCA, is interpreted as a bipartite graph $G = (X \cup Y, B(G))$, where $X = \{u_j\}$ is the set of source vertices u_js, $Y = \{v_i\}$ the set of output vertices v_is, and $B(G)$ the adjacency matrix. A matching M is a set of edges representing unique pairs of vertices (u_i, v_j). No vertex is repeated in a matching. For example, consider the graph in Fig. 1.3. Two unique pairs (u_1, v_1) and (u_2, v_2) can be identified as shown in Fig. 2.1. Edges corresponding to these two pairs of vertices form a matching $M = \{(u_1, v_1),\ (u_2, v_2)\}$. A maximal matching is a matching with maximum possi-

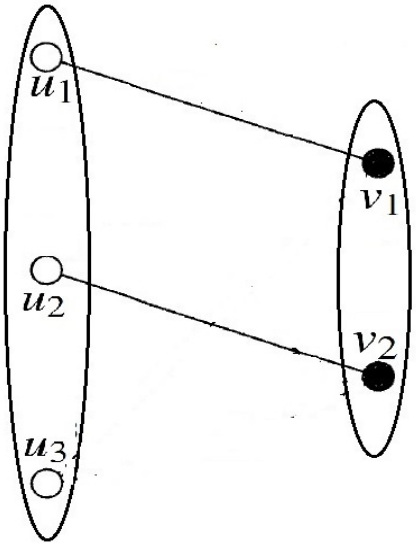

Figure 2.1: Maximal matching in bipartite graph presented in Fig. 1.3

ble number of edges, $\max(|M|)$. As there are no more pairings possible in the graph in Fig. 1.3, matching shown in Fig. 2.1 is maximal. Phrase 'maximal matching' is used as two more matchings of size 2 can be identified in the same graph.

An identifiable graph is argued to be the one that has a maximal matching of size L and every sub-graph $H_j \subset G$ induced by anti-neighbourhood of source vertex $j \in X$ has a matching of size $L-1$. This result is shown to be applicable in testing

if a network is NCA compatible or not. However, the fundamental assumption that rank of any bipartite graph is equal to the size of a maximal matching is inaccurate as shown in the next chapter. In fact, size of a maximal matching is equal to the structural rank of a matrix [83] which is the maximum rank that a matrix of a particular structure can achieve. Accurate description of NCA identifiability is developed in chapter 4 where assumptions 1–3 are translated to assumptions on $B(G)$ and D.

Methods to identify largest NCA identfiable sub-networks of a given network are presented in [60], [82] and [74]. Despite the fact that these methods are based on inaccurate description of NCA identifiability, they carry other limitations. Methods developed in [60] and [82] identify one largest NCA identifiable sub-network in a given network $B(G)$. This implies that a part of the system of linear equations in (1.3) remains unsolved. On the other hand, methods presented in [60] and [74] identify two sub-networks and can be extended to idenify more NCA identifiable sub-networks. The method employs a random search based method to group columns of $B(G)$ that satisfy assumption 5 individually and assumption 4 collectively. However, such a method might take a long time to converge and might end up in a sub-optimal solution. Method proposed in [82] is defines largest sub-network as the one that is made up of the largest separable set of vertices that is NCA identifiable. A separable set of vertices is the one whose neighbours are not connected to any other vertex outside the identified set. It is shown in chapter 5 that the whole network can be decomposed into NCA identifiable parts without ignoring any part of the original network or requiring them to be completely separable.

In summary, rank conditions imposed in [2] on W and S are sufficient to characterise uniqueness of NCA solutions. However, uniqueness of NCA solutions can be tested only after application of NCA. One of the goals in this thesis is to develop equivalent rank conditions on $B(G)$ and D that facilitate a priori NCA identifiability test of a given network. NCA identifiability of networks apt for such a priori feasibility test has not been accurately described in any relevant work.

2.4 Exploiting sparsity and graph based approaches

Graph representations play a major role in solving sparse systems of linear equations [84]. Use of graph theoretical interpretations in analysing data from biological experiments is not new. A review of graph based methods applied to general data based modelling problems in the context of cell biology is presented in [85]. Use of graph matching is not new and is shown to be comparable to matrix inversion in

terms of computational complexity [86]. A review on graph matching techniques is available in [87]. Methods that improve the speed of computing maximal matching in bipartite graphs are presented in [88] and [89].

In the context of NCA, [60] and [82] have explicitly use graph notations to achieve two goals – characterising NCA identifiability and identifying a largest NCA identifiable sub-network of a given network. Though these methods attempt to exploit the structure of $B(G)$ to a certain extent, they are limited in applicability. Method in [60] attempts to find a largest possible NCA identifiable sub-network in a random manner. A computational complexity analysis is not presented in [60], but this problem can be compared to the optimal graph pattern mining problem discussed in [90]. It is pointed there that a brute-force method involves listing all possible patterns and then maximising a criterion. Such an approach can prove to be a significant computational hurdle. For a network $B(G)$ with M rows and L columns, total number of possible permutations are

$$n_P = \sum_{m=1}^{M} {}^{M}P_m \times \sum_{l=1}^{L} {}^{L}P_l \tag{2.17}$$

where, m and l respectively represent the number of rows and columns being moved in $B(G)$ to obtain a new permutation. First part the equation in (2.17) represents number of possible ways in which rows of $B(G)$ can be reordered whereas the second part represents the number of possible ways in which the columns of $B(G)$ can be reordered. In Fig. 2.2, growth of n_P (2.17) as a function of M and L is shown on a log scale. It can be seen in Fig. 2.2 that the number of possible permutations grow

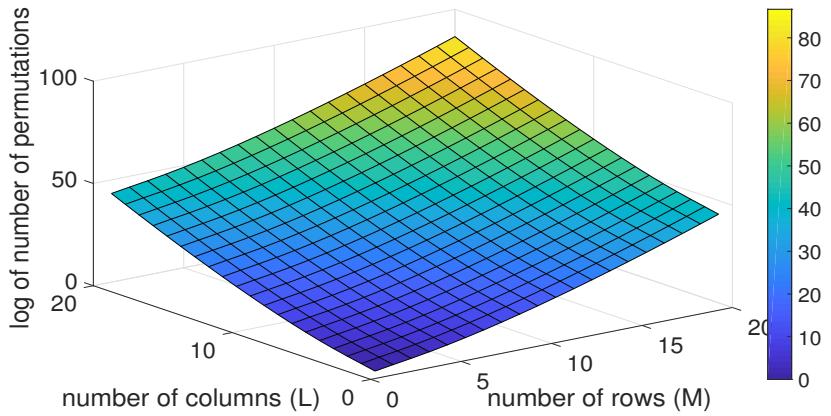

Figure 2.2: $log(n_P)$ as a function of M and L

rapidly for a networks with maximum of 20 rows and 20 columns. Method in [60] is

not computationally as complex as an brute-force based optimal graph pattern mining problem as not all possible combinations are considered. However, this implies that there is no guarantee that a sub-network identified in [60] is the largest possible sub-network. Therefore, this method is not considered in this thesis. Matching based method in [82] looks for a single largest separable set of columns that are NCA identifiable. Though this method is computationally not as intensive as the previous one, but the definition of the largest NCA identifiable graph is quite limiting. This is because modularity in transcriptional regulatory networks does not imply functional separability [91] and hence, NCA identifiable separable sets might be either too small in size to be meaningful or simply might not exist in a given biological network. Definition of largest NCA identifiable sub-network in [82] is based on inaccurate matching based asumption and limited in its applicability. Therefore, it is not used for decomposing a network into NCA identifiable sub-networks or for comparative analysis in this thesis.

In biological systems, sparsity is an emergent property [92]. Graph based methods and sparse matrix based techniques are frequently used together for big-data analysis [93]. General interest in problems involving sparse matrices is growing steadily and a wide variety of techniques are applicable to biological systems [94]. Biological systems are modular in nature [95, 96]. Modularity and sparseness of a biological network are assumed to be inter-dependent in some cases and assessed separately in others [97]. Modularity can be considered by treating the constraint matrix $B(G)$ in (2.12) as semi-separable along the lines discussed in [98]. Minimum degree reordering has been used for pivoting to achieve less fill-in for Gaussian elimination thereby reducing computation complexity [84]. In a recent study, it is shown that minimum degree reordering improves speed of convergence in interior point based iterative methods for solving systems of linear equations [99]. Reordering rows and columns of $B(G)$ based on the number of zeros leads to modular structures in SCMF problem as shown in [74, 78]. Apart from those two methods, modularity has not been an important aspect of NCA discussed in section 2.3.1 and its extensions in section 2.3.2.

As discusssd in this subsection, graph based techniques that exploit sparsity of a given network not only offer computationally efficient solutions, but also provide a way to exploit modularity that is generally observed in biological networks. However, previous attempts in NCA literature to develop such methods are only partially successful. Interesting ideas have been proposed in [82] and [78]. The former method is theoretically inaccurate and limited in its applicability whereas the latter lacks in theoretical rigour. Moreover, it is shown in this thesis that it is not

necessary to find a largest NCA identifiable sub-network when given network $B(G)$ is NCA incompatible. It is shown in this thesis that decomposing $B(G)$ into NCA feasible parts is sufficient to solve SCMF problem in (2.12).

2.5 Carbon source switch in Escherichia coli - experimental data

Escherichia coli (E. coli) is a model organism biological mechanisms of which are well understood [100]. Thousands of functional genes and hundreds of transcriptional factors in E. coli have been discovered and studied. Large number of multi-omics experimental datasets related to E. coli studies are available in [101] and [102]. However, a dataset presented in [1] related to a carbon source switch experiment that focuses on a relatively small regulatory network with 100 genes and 16 transcription factors is chosen for comparative analysis in this thesis. Reasons to choose this dataset are

1. indirect measure of activity of a protein called CRP is available, this can be used to validate methods developed in this thesis

2. this dataset can be used to validate new methods developed in this thesis with minimal understanding of underlying biology

This dataset provides a neat platform to test applicability and validate the outcomes of methods developed in this thesis. A brief discussion on the experimental data is presented next.

The experiment in [1] is designed to record gene expression D of 100 genes of E. coli bacteria when carbon source changes in the culture's environment. 25 data points are collected using ChIP sequencing technology. However, only 9 data points are made available along with NCA toolbox [20]. As mentioned in section 2.3.3, gene expression data is generally recorded at few time points. Authors of NCA-toolbox [20] have not made any more data points available. Data points recorded in columns 2 to 10 of D correspond to 5, 15, 30, 60, 120, 180, 240, 300, and 360 minutes, respectively. Available data matrix D is illustrated in Fig. 2.3. Goal in [1] is to estimate activities S of 16 transcription factors. The underlying regulatory network $B(G)$ represents regulatory connections between different genes and proteins. The network $B(G)$ with 100 rows and 16 columns is illustrated in Fig. 2.4 where black colour corresponds to 0 or no connection whereas white represents 1 or a connection. As mentioned earlier, protein activities are not generally

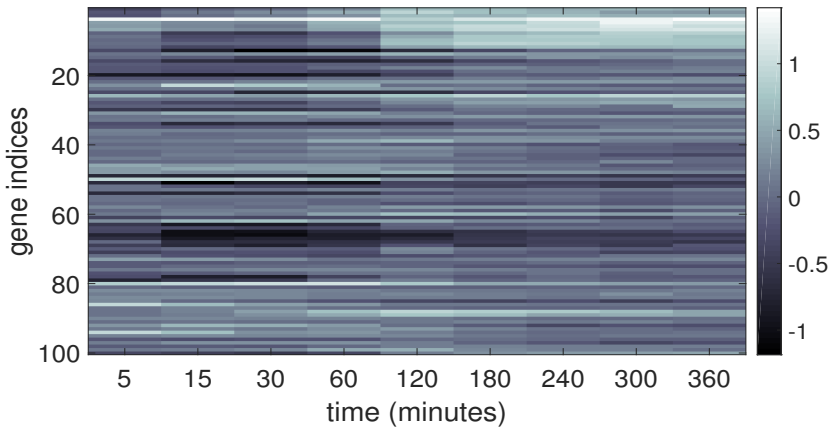

Figure 2.3: Colour map of gene expression data – carbon source switching experiment [1]

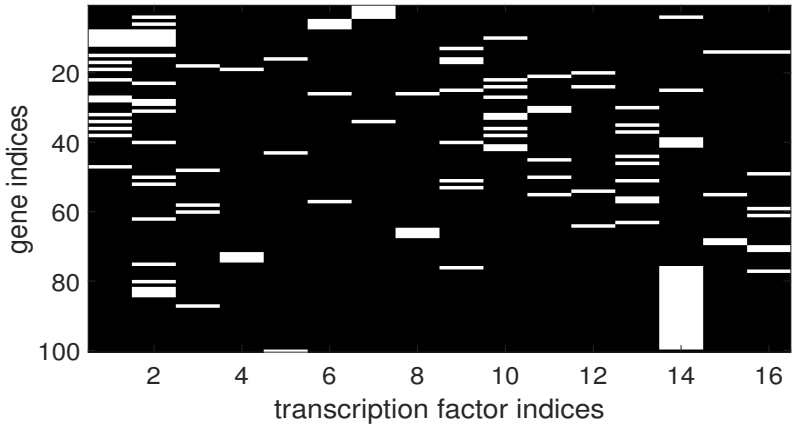

Figure 2.4: Colour map of $B(G)$ (0 – black, 1 – white): transcriptional regulatory network, carbon source switching experiment [1]

measured in ChIP sequencing based experimental setups. However, it is mentioned in [1] that one of the transcription factors called CRP must bind to a particular metabolite called cyclic AMP. Concentrations of cyclic AMP are measured and are used to validate NCA estimate of CRP activity in [1]. Most important features deduced from cyclic AMP concentrations that are used to validate CRP activity are

1. CRP activity is expected to peak within 60 minutes and monotonically reduce thereafter

2. a time difference of approximately 4 hours is expected between peak and valley of CRP activity

These features are used to validate estimates of CRP activity obtained by new meth-

ods developed in this thesis.

Some initial assessment on NCA identifiability of the chosen dataset D and network $B(G)$ is done. It is found that $B(G)$ (Fig. 2.4) satisfies assumptions 4 and 5. However, number of transcription factors $L = 16$ is greater than the number of data points $N = 9$. Therefore, rank of source signal matrix S cannot exceed 9. This violates the condition in theorem 1 that S must have full row rank. Thus, NCA solutions are not expected to be unique for this experimental data. It cannot be tested if S satisfies assumption 6 or not unless NCA is applied to the given dataset-network pair $(D, B(G))$. It is shown in the next chapter that NCA solutions are not unique as uniqueness criteria in (2.14) is not satisfied. This dataset is used throughout the chapter to demonstrate theoretical gaps in state-of-art and to validate the novel contributions made in this thesis.

2.6 Summary

ChIP sequencing technology for recording gene expression combined with TMM data normalisation technique limits the choice of models and modelling methods. Among different models of TRN that are available in literature ODE and log-linear models are best suited to model data recorded in a ChIP-TMM setup. As ChIP sequencing method is generally not used to record protein concentrations in lab based experiments, the problem of modelling TRNs using log-linear models is, mathematically, a matrix factorisation problem. TRNs generally have well defined structure stemming from biology. Therefore, gene expression data modelling is a structure-constrained matrix factorisation problem. In this thesis, it is assumed that a regulatory pattern for a TRN of interest is available and the TRN strictly follows this pattern. This further limits the type of parameter estimation methods that can be used. NCA is a popular optimisation based method designed to obtain unique matrix factors for a given dataset-network pair. However, NCA imposes several rank conditions on these matrix factors which cannot be tested beforehand. Different methods based on NCA available in literature do not accurately define NCA identifiability. Furthermore, NCA based methods which address a case where a given network is not NCA identifiable only partially solve the original SCMF problem. Therefore, there are three open issues with respect to NCA – method to test NCA problem feasibility, accurate definition of NCA identifiability, and method to uniquely and completely solve problems involving non-NCA identifiable TRNs. Exploiting sparsity of TRNs with the help of graph based methods might be useful in resolving these open issues. However, sparsity of TRNs has not been exploited

well in the context of NCA. Goals related to the three open issues are pursued in this thesis. First goal in this thesis is to develop accurate NCA identifiability conditions that can be tested before applying NCA to a dataset. Second goal is to develop a graph based method to decompose a given network into NCA identifiable sub-parts and the third is to develop a method to combine NCA solutions of the identified sub-networks to obtain unique solutions to a general SCMF problem.

Chapter 3

Preliminary analysis

The objective of this chapter is to test NCA with a set of numerical examples to identify open issues in NCA. NCA proposed in [2] and [20], among other SCMF techniques, factorise gene expression data obtained using ChIP sequencing. As pointed out in section 2.2.2, this thesis focuses only on NCA as there are ample opportunities to carry out fundamental theoretical research. Two key aspects related to NCA studied in this chapter are

1. rank conditions imposed on matrices W and S – assumptions 1, 2 and 3
2. uniqueness of solutions

In this chapter, these two aspects of NCA are probed with the help of numerical examples to expose gaps in theory and implementation of NCA.

As pointed out in section 2.3.3, only handful number of articles in NCA literature focus on theoretical aspects of NCA identifiability. It is argued in the same section that all such theoretical results related to NCA are only partially accurate. Counter examples are used in this chapter to demonstrated that such a claim is valid and an accurate description of NCA identifiability is not yet available. In section 3.1, it is shown that binary rank of $B(G)$ might affect its NCA compatibility. A relationship between rank of W and binary rank of $B(G)$ is developed in section 3.1.3. Observations made in section 3.1 serve as motivation for chapter 4 where accurate NCA feasibilty conditions are developed.

Use of graph theoretical methods to identify NCA feasible sub-networks of a given network is reported in section 2.4. Size of a maximal matching in a bipartite graph G is claimed to be equal to the rank of its adjacency matrix $B(G)$ in [82]. This relationship is shown to be inaccurate in section 3.2 with the help of a counter example. It is shown in section 3.2.2 that rank of an example matrix $B(G)$ is equal to size of a matching after removing duplicate and zero rows in $B(G)$. This serves

as motivation for chapter 5 where a bipartite matching based method is developed to identify NCA feasible sub-networks of a given network.

As mentioned in section 2.3.1, uniqueness of NCA solutions can be guaranteed by testing a pair of NCA solutions $(({}^1W),({}^1S))$ and $(({}^2W),({}^2S))$ against (2.14). However, no method is explicitly made available in NCA literature. Moreover, testing uniqueness of NCA solutions is generally ignored in most of the associated literature. A method to compute a scaling matrix that connects two NCA solutions as given by (2.14) is developed in section 3.3. This method is used in section 3.3.2 to demonstrate the fact that NCA solutions corresponding to the experimental dataset presented in section 2.5 are not unique. Results from chapters 4 and 5 are combined in chapter 6 where estimating unique solutions corresponding to NCA incompatible networks is of prime importance.

3.1 Binary rank of a network

In this section, it is shown with the help of a numerical example that binary rank of $B(G)$ can affect NCA compatibility of a network. This serves as a counter example that disproves NCA identifiability theorem [60]. Consider a binary matrix $B(G) \in GF_2^{6\times 4}$ with set of rows $Y = \{v_i : 1 \leq i \leq 6\}$ and set of columns $X = \{j : 1 \leq j \leq 4\}$

$B(G)$	u_1	u_2	u_3	u_4
v_1	0	0	1	1
v_2	0	1	0	1
v_3	0	1	1	0
v_4	1	0	0	1
v_5	1	0	1	0
v_6	1	1	0	0

(3.1)

$B(G)$ is the adjacency matrix of a bipartite graph $G = (X \cup Y, B(G))$ similar to the one in Fig. 1.3, where ij-th element of $B(G)$ is given as

$$B(G)(i,j) = \begin{cases} 1, \text{ if } (i,j) \text{ is an edge} \\ 0, \text{ otherwise} \end{cases}$$

If the network in (3.1) is NCA identifiable, then $B(G)$ is expected to satisfy assumptions 4 and 5. In this section, it is shown that these assumptions are inaccurate.

3.1.1 NCA identifiability - example network

In section 2.3.3, assumption 5 is introduced where reduced sub-matrices $B(H_j)$s of $B(G)$ are discussed. It is important to note the difference in reduction of $B(G)$ in the sense of NCA and graph theory. In the context of NCA, reduction refers to deletion of j-th column and all rows with non-zero entries on j-th position in $B(G)$ whereas in graph theory, reduction of a graph G means removal of duplicate and isolated vertices in G. These two operations are not equivalent. A reduced sub-matrix in the sense of NCA is equal to the adjacency matrix of a sub-graph obtained by deleting vertex u_j and all its neighbours $N(u_j)$

$$B(H_j) = B(G \setminus \{u_j, N(u_j)\}) \tag{3.2}$$

$B(G)$ in (3.1) has 4 linearly-independent rows in $\mathbb{R}$: v_1, v_2, v_3 and v_4, and hence, $rk(B(G)) = 4$. The other rows can be expressed as

$$\begin{aligned} v_5 &= v_2 - v_3 + v_4 \\ v_6 &= -v_1 + v_3 + v_4 \end{aligned} \tag{3.3}$$

The sub-matrices of $B(G)$, $B(H_j), j = 1,\ 2,\ 3$ and 4, are obtained by removing j-th column and all rows with 1 on j-th position as

$B(H_1)$	u_1	u_2	u_3	u_4
v_1	0	0	1	1
v_2	0	1	0	1
v_3	0	1	1	0
~~v_4~~	~~1~~	~~0~~	~~0~~	~~1~~
~~v_5~~	~~1~~	~~0~~	~~1~~	~~0~~
~~v_6~~	~~1~~	~~1~~	~~0~~	~~0~~

$B(H_2)$	u_1	u_2	u_3	u_4
v_1	0	0	1	1
~~v_2~~	~~0~~	~~1~~	~~0~~	~~1~~
~~v_3~~	~~0~~	~~1~~	~~1~~	~~0~~
v_4	1	0	0	1
v_5	1	0	1	0
~~v_6~~	~~1~~	~~1~~	~~0~~	~~0~~

(3.4)

$B(H_3)$	u_1	u_2	u_3	u_4
~~v_1~~	~~0~~	~~0~~	~~1~~	~~1~~
v_2	0	1	0	1
~~v_3~~	~~0~~	~~1~~	~~1~~	~~0~~
v_4	1	0	0	1
~~v_5~~	~~1~~	~~0~~	~~1~~	~~0~~
v_6	1	1	0	0

$B(H_4)$	u_1	u_2	u_3	u_4
~~v_1~~	~~0~~	~~0~~	~~1~~	~~1~~
~~v_2~~	~~0~~	~~1~~	~~0~~	~~1~~
v_3	0	1	1	0
~~v_4~~	~~1~~	~~0~~	~~0~~	~~1~~
v_5	1	0	1	0
v_6	1	1	0	0

(3.5)

It can be verified that all sub-matrices $B(H_j)$s (3.4) and (3.5) are permutations of

$B(H)$	c_1	c_2	c_3
r_1	0	1	1
r_2	1	0	1
r_3	1	1	0

(3.6)

It is easy to verify that $B(H)$ (3.6) is full-ranked. Therefore, all sub-matrices $B(H_j)$s (3.4) and (3.5) of $B(G)$ (3.1) are full-ranked. Thus, B(G) is NCA-compatible according to assumptions 4 and 5 (chapter 2, section 2.3).

Consider $W \in \mathbb{R}^{6\times 4}$

W	u_1	u_2	u_3	u_4
v_1	0	0	2	-2
v_2	0	2	0	-2
v_3	0	-2	2	0
v_4	-2	0	0	2
v_5	2	0	-2	0
v_6	-2	2	0	0

(3.7)

W is such that $\phi(W) = B(G)$, i.e., W follows the structure imposed by $B(G)$. In theory, it is possible to construct a dataset D such that NCA-estimate of the corresponding weight matrix is as given in (3.7). It is shown in the next chapter that W (3.7) can be derived from $B(G)$. It can be verified that in W (3.7)

$$\begin{aligned} -v_5 &= v_2 + v_3 + v_4 \\ v_6 &= v_1 - v_3 + v_4 \\ v_3 &= v_1 - v_2 \end{aligned} \tag{3.8}$$

and hence,

$$rk(W) = 3 \tag{3.9}$$

It is important to note that not all choices of $W, \phi(W) = B(G)$ will lose rank. It can be verified W (3.7) will achieve full column rank when one of its entries is changed to some real number other than 2 or -2. Therefore, only those matrices chosen from a particular subspace of matrices characterised by the linear dependencies in (3.8) will lose rank. Existence of such matrices is explored in chapter 4.

Sub-matrices W_j of S corresponding to $B(H_j),\ j = 1,\ 2,\ 3$ and 4 in (3.4) and

(3.5) are permutations of

$\bar{W}$	c_1	c_2	c_3
r_1	0	2	-2
r_2	2	0	-2
r_3	-2	2	0

(3.10)

It is evident in (3.10) that $r_3 = r_1 - r_2$ and hence,

$$rk(\bar{W}) = 2 \tag{3.11}$$

From (3.9) and (3.11), any NCA solution with W (3.7) as the weight matrix is not unique according to the original NCA theorem 1 (chapter 2, section 2.3). This also implies by contradiction that $B(G) = \phi(W)$ is not NCA compatible. This is a case of inaccurate classification of given network as NCA compatible.

3.1.2 Binary rank - example network

Ranks of $B(G)$ (3.1) and $B(H_j)$s (3.4) and (3.5) are computed over real-field $\mathbb{R}$ in section 3.1.1, but all matrices are binary valued. Computing binary ranks of these matrices may lead to accurate classification of networks as NCA compatible. $B(G)$ (3.1) is used here to demonstrate the importance of considering binary rank to test a network for NCA compatibility.

Definition of elementary operations such as addition, subtraction, and multiplication depend on the numerical field to which the variables of interest belong [41]. In binary field, denoted by GF_2 in this thesis, both addition and subtraction are defined as

$$0 \oplus 0 = 0,\ 1 \oplus 0 = 1 \text{ and } 1 \oplus 1 = 0 \tag{3.12}$$

where $\oplus$ denotes logical-XOR operation. Multiplication in GF_2 is defined as

$$0 \,.\, 0 = 0,\ 1 \,.\, 0 = 0 \text{ and } 1 \,.\, 1 = 1 \tag{3.13}$$

where $.$ denotes logical-AND operation. Binary rank of a binary matrix $B(G)$, denoted by $rk_2(B(G))$, can be computed using any conventional technique in [103] whilst considering the definitions in (3.12) and (3.13).

It is shown in section 3.1.1 that $rk(B(G)) = 4$ and $rk(B(H)) = 3$ and hence, $B(G)$ is NCA-compatible. However, these matrices lose rank when their ranks are computed over GF_2. Upon examining the rows of $B(G)$, it can be seen that the

linearly-dependent rows in $\mathbb{R}$ (3.3) are linearly-dependent in GF_2

$$v_5 = v_2 \oplus v_3 \oplus v_4$$
$$v_6 = v_1 \oplus v_3 \oplus v_4$$

In addition to that,

$$v_3 = v_1 \oplus v_2 \tag{3.14}$$

Thus, there are 3 linearly-independent rows in GF_2: v_1, v_2 and v_4 and hence,

$$rk_2(B(G)) = 3 \tag{3.15}$$

Similarly, upon examining $B(H)$ (3.6), it is evident that $r_3 = r_1 \oplus r_2$ and hence,

$$rk_2(B(H)) = 2 \tag{3.16}$$

Therefore, from (3.15) and (3.16), $B(G)$ does not satisfy extended NCA rank-conditions if rk_2 is considered.

Thus, considering binary rank of any structural-constraints matrix is crucial for ensuring uniqueness of NCA solutions. These observations serve as motivation in the next chapter to develop an accurate definition of NCA compatible structural constraints matrix.

3.1.3 Binary rank and real rank

It is shown in the previous section that binary rank of a network plays an important role in testing NCA compatibility. In this section, a fundamental result that relates binary rank of $B(G)$ $rk_2(B(G))$ to the real rank of W $rk(W)$ is developed as

Lemma 1 *Matrices $W \in \mathbb{R}^{M\times L}$ and $B(G) \in GF_2^{M\times L}$ satisfy*

$$rk(W) \geq rk_2(B(G))$$

if $\phi(W) = B(G)$, where $\phi(.)$ is a bijection defined in (2.10).

Proof If $\phi(W) = B(G)$, then

$$B(G)(i,j) \neq 0 \implies W(i,j) \neq 0,\ 1 \leq i \leq M,\ 1 \leq j \leq L$$

Using appropriate row and column reordering followed by row reduction, it can be shown that $B(G)$ and W will have same number of pivots

$$rk(W) = rk(B(G))$$

A binary matrix $B(G)$ satisfies the condition $rk(B(G)) \geq rk_2(B(G))$ [104]. Thus,

$$rk(W) \geq rk_2(B(G))$$

□

This result is used in the next chapter to develop binary rank based NCA compatibility conditions.

3.2 Matching based NCA identifiability

It is shown in the previous section that binary rank of a network is important in characterising a network $B(G)$ as NCA identifiable or otherwise. In section 2.4, it is pointed out that accurate graph theoretical interpretation of NCA rank conditions is useful in identifying NCA identifiable sub-networks. Graph theoretical interpretation of NCA conditions in [82] is based on the fact that size of a maximal matching in a graph G is equal to $rk(B(G))$. In section 3.2.1, it is shown that a matching based characterisation of NCA compatible networks in [82] is inaccurate. Also, it is shown that for $B(G)$ in (3.1) that size of a maximal matching in G is not equal to $rk_2(B(G))$. This further reduces the applicability of results in [82] as binary rank is shown to be important in the context of NCA. It is shown in sections 3.2.3 and 3.2.2 that matrix manipulation such as reordering and deleting some of the rows and columns might lead to a better graph theoretical charcaterisation of NCA identifiable graphs.

3.2.1 Matching and rank of a network

Consider a sub-matrix $B(H)$ of $B(G)$ in (3.1) given by

$$B(H) = \begin{array}{c|cccc} & u_1 & u_2 & u_3 & u_4 \\ \hline v_1 & 0 & 0 & 1 & 1 \\ v_3 & 0 & 1 & 1 & 0 \\ v_4 & 1 & 0 & 0 & 1 \\ v_6 & 1 & 1 & 0 & 0 \end{array} \tag{3.17}$$

It can be seen that $B(H)$ in (3.17) has linearly dependent rows identified in (3.3) and hence

$$rk(B(H)) = 3 \tag{3.18}$$

A maximal matching in $B(H)$ is given by

$$M_{(H)} = (\{u_1,\ u_2,\ u_3,\ u_4\}, \{v_4,\ v_6,\ v_3,\ v_1\}) \tag{3.19}$$

Entries of $B(H)$ that correspond to $M_{(H)}$ in (3.19) are encircled as

$B(H)$	u_1	u_2	u_3	u_4
v_1	0	0	1	①
v_3	0	1	①	0
v_4	①	0	0	1
v_6	1	①	0	0

From (3.18) and (3.19)

$$\begin{gathered} rk(B(H)) = 3,\ \ |M_{(H)}| = 4 \\ rk(B(H)) < |M_{(H)}| \end{gathered} \tag{3.20}$$

This shows that it is possible to construct a network $B(H)$ such that its rank is smaller than size of a maximal matching $|M_{(H)}|$. Therefore, maximal matching based NCA identifiability conditions presented in [82] are inaccurate.

Importance of binary rank in the context of NCA is demonstrated in the previous section. It is shown here that size of a maximal matching in a graph is not equal to the binary rank of its adjacency matrix. Consider $B(G)$ in (3.1). A maximal matching in $B(G)$ is given by

$$M_{(G)} = (X^*, Y^*), X^* \subset X = \{u_1,\ u_2,\ u_3,\ u_4\}, Y^* \subset Y = \{v_4,\ v_2,\ v_3,\ v_1\} \tag{3.21}$$

Entries of $B(G)$ that correspond to $M_{(G)}$ in (3.21) are encircled as

$B(G)$	u_1	u_2	u_3	u_4
v_1	0	0	1	①
v_2	0	①	0	1
v_3	0	1	①	0
v_4	①	0	0	1
v_5	1	0	1	0
v_6	1	1	0	0

As claimed in [82], from (3.21) and (3.3)

$$rk(B(G)) = |M_{(G)}|$$

However, from (3.15) and (3.21)

$$rk_2(B(G)) < |M_{(G)}| \tag{3.22}$$

It can be seen from the two examples in this subsection that

$$rk_2(B(G)) \leq rk(B(G)) \leq |M_{(G)}| \tag{3.23}$$

As mentioned in section 2.4, finding a maximal matching is equivalent to computing structural rank of a matrix. In section 3.1.1, it is shown that W in (3.7) characterised by linear dependencies in (3.8) lose rank despite corresponding structural constraints matrix $B(G)$ being full ranked in $\mathbb{R}$. Thus, identifying a maximal matching (X_1^*, Y_1^*) is not necessarily equivalent to calculating binary rank of the original graph G

$$rk_2(B(G)) \neq |M_{(G)}|$$

For $B(H)$ in (3.17), a simple breadth-first based matching is employed to check if $rk_2(B(H))$ can be accurately estimated. This is done by scanning rows from left to right until a 1 is found. The found 1 is encircled and the corresponding row and column are removed from search and the process is repeated till all rows are exhausted. A breadth-first search based matching $M_{(H,bfs)}$ in $B(H)$ is identified as

$B(H)$	u_1	u_2	u_3	u_4
v_1	0	0	①	1
v_3	0	①	1	0
v_4	①	0	0	1
v_6	1	1	0	0

(3.24)

Note that this is not a maximal matching. However, from (3.24), size of the matching $M_{(H,bfs)}$ with breadth-first approach is such that

$$|M_{(H,bfs)}| = rk_2(B(H)) = 3 \tag{3.25}$$

However, $B(H)$ in (3.17) has equal number of non-zero entries across rows and columns. It is shown in the next subsection that a breadth-first matching fails to

accurately estimate binary-rank of a binary matrix if its rows and columns differ in number of non-zero entries.

3.2.2 Eliminating duplicate and zero rows

Consider another example $B(G_1)$ with breadth-first search based matching $M_{(G_1,bfs)}$ given as

$B(G_1)$	u_1	u_2	u_3	u_4
v_1	①	0	1	1
v_2	0	①	1	0
v_3	0	1	①	0
v_4	0	1	1	①

(3.26)

In this case $M_{(G_1,bfs)}$ is maximal and is such that

$$|M_{(G_1,bfs)}| > rk_2(B(G_1))$$

It can be seen in $B(G_1)$ (3.26) that row v_3 is a duplicate of v_2. A reduced bipartite graph is defined as follows

Definition 1 *A reduced bipartite graph $G_r = (X_r \cup Y_r)$ is a sub-graph of G such that*

1. *$X_r \subseteq X$ and $Y_r \subseteq Y$*
2. *there are no isolated vertices, $d(v) > 0, \forall v \in X_r \cup Y_r$*
3. *there are no duplicate vertices, $\mathcal{N}(v_1) \neq \mathcal{N}(v_2), \forall v_1, v_2 \in X_r \cup Y_r$*

Consider a reduced version of $B(G_1)$ in (3.26) given by

$$B(G_{1,r}) = \begin{array}{c|cccc} & u_1 & u_2 & u_3 & u_4 \\ \hline v_1 & 1 & 0 & 1 & 1 \\ v_2 & 0 & 1 & 1 & 0 \\ v_4 & 0 & 1 & 1 & 1 \end{array} \tag{3.27}$$

where, r in subscript represents graph reduction. A breadth first matching in $B(G_{1,r})$ can be identified as

$$B(G_{1,r}) = \begin{array}{c|cccc} & u_1 & u_2 & u_3 & u_4 \\ \hline v_1 & ① & 0 & 1 & 1 \\ v_2 & 0 & ① & 1 & 0 \\ v_4 & 0 & 1 & ① & 1 \end{array} \tag{3.28}$$

It can be verified that $rk_2(B(G_1)) = 3$. From (3.28)

$$|M_{(G_{1,r},bfs)}| = rk_2(B(G_1)) \tag{3.29}$$

It is shown in chapter 5 that reducing graph is important in the context of NCA feasibility.

3.2.3 Reordering rows and columns

Importance of identifying largest full-ranked sub-network in the context of NCA is highlighted in section 2.3.3. It is shown in the previous subsection that reducing graphs combined with breadth-first matching can lead to an accurate estimate of binary rank of a binary matrix. However, the pivots identified in $B(G_{1,r})$ in (3.28) may not necessarily correspond to largest full-ranked sub-matrix of $B(G_1)$. In order to demonstrate this fact, consider the following sub-matrix of $B(G_{1,r})$ in (3.28) with a breadth-first matching given as

$$B(H_1) = \begin{array}{c|ccc} & u_1 & u_2 & u_3 \\ \hline v_1 & \textcircled{1} & 0 & 1 \\ v_2 & 0 & \textcircled{1} & 1 \\ v_4 & 0 & 1 & \textcircled{1} \end{array} \tag{3.30}$$

It is easy to see that $rk_2(B(H_1)) = 2$ whereas $|M_{(H_{1,bfs})}| = 3 > rk_2(B(G_1))$. This is because v_4 is a copy of v_2. Reducing H_1 can lead to accurate estimate of binary rank of $B(H_1)$, but it does not resolve the issue of finding the largest full-ranked sub-matrix of the original matrix $B(G_1)$ in (3.26).

A minimum degree reordering on G is defined as

Definition 2 *A minimum degree reordered bipartite graph $G_< = \{X_< \cup Y_<\}$ is obtained by rearranging vertices in X and Y in the increasing order of their degrees.*

Consider a minimum degree reordering as described in definition 2 of $B(G_{1,r})$ in (3.27) given as

$$B(G_{1,r<}) = \begin{array}{c|cccc} & u_1 & u_2 & u_4 & u_3 \\ \hline v_2 & 0 & 1 & 0 & 1 \\ v_1 & 1 & 0 & 1 & 1 \\ v_4 & 0 & 1 & 1 & 1 \end{array} \tag{3.31}$$

Note that the row and column indices in (3.27) have been reordered in (3.31) such

that rows and columns appear in the increasing order of the number of 1s in them. Notation $G_{1,r<}$ is used in (3.31) where $<$ represents minimum degree ordering. A breadth-first matching $M_{(G_{1,r<})}$ can be identified in (3.31) as

$$B(G_{1,r<}) = \begin{array}{c|cccc} & u_1 & u_2 & u_4 & u_3 \\ \hline v_2 & 0 & \textcircled{1} & 0 & 1 \\ v_1 & \textcircled{1} & 0 & 1 & 1 \\ v_4 & 0 & 1 & \textcircled{1} & 1 \end{array} \tag{3.32}$$

From (3.32), a full-ranked sub-matrix of $B(G_1)$ in (3.26) as

$$B(H_{full-ranked}) = \begin{array}{c|ccc} & u_1 & u_2 & u_4 \\ \hline v_2 & 0 & 1 & 0 \\ v_1 & 1 & 0 & 1 \\ v_4 & 0 & 1 & 1 \end{array}$$

In the context of NCA, discussion in this section demonstrates the need to reduce and order adjacency matrix of a given network before looking for a maximal matching as defined in [82]. A formal definition of NCA feasible sub-networks is developed in chapter 5 based on the findings of this section.

3.3 NCA uniqueness test

As pointed out in section 2.3.1, uniqueness of NCA solutions can be guaranteed if χ (2.14) is diagonal [2]. However, there is no method available in literature to compute χ. In this section, a method based on vectorisation of (2.14) is developed. This method is used to test uniqueness of NCA solutions corresponding to the experimental dataset presented in section 2.5. An open issue with respect to NCA uniqueness is pointed out in this section.

3.3.1 Vectorised NCA uniqueness relationship

Vectorised version $vec.X$ of a matrix $X \in \mathbb{R}^{N\times P}$ is obtained by stacking all columns of X in a single column. Kronecker-product $A \otimes B \in \mathbb{R}^{MP\times NQ}$ of arbitrary matrices

$A \in \mathbb{R}^{M\times N}$ and $B \in \mathbb{R}^{P\times Q}$ is defined as

$$A \otimes B = \begin{pmatrix} A(1,1)B & \cdots & A(1,N)B \\ \vdots & \ddots & \vdots \\ A(M,1)B & \cdots & A(M,N)B \end{pmatrix}$$

Consider a matrix relationship

$$AXB = C \tag{3.33}$$

where, C is of appropriate dimensions. Vectorising (3.33) results in

$$(B^T \otimes A)vec.X = vec.C \tag{3.34}$$

Using the relationship in (3.34), (2.14) can be rewritten as

$$(I_L \otimes ({}^1W))vec.\chi = vec.({}^2W) \tag{3.35}$$

Given any two pairs of NCA solutions $(({}^1W),({}^1S))$ and $(({}^2W),({}^2S))$, $vec.\chi$ can be calculated by solving (3.35). NCA solutions are unique if χ obtained by rearranging $vec.\chi$ is a diagonal matrix.

3.3.2 Carbon source switch - uniqueness

NCA for short data records [20] is applied twice to dataset D (Fig. 2.3) obtained from carbon source switch experiment [1]. Solutions $(({}^1W),({}^1S))$ and $(({}^2W),({}^2S))$ are illustrated in Fig. 3.2 – 3.5. It can be seen from the scales in Fig. 3.2 and 3.3 that all the entries in $({}^1W)$ and $({}^2W)$ are within the range $(-2.5,2.5)$. However, the two matrices are not identical. On visual inspection, it can be seen that patterns in columns 2 and 14 are noticably different. From Fig. 3.4 and 3.5, colour maps of $({}^1S)$ and $({}^2S)$ are visibly different. The maximum and minimum values on scales next to the colour maps are also different in the two figures. It is found in computations that both $({}^1S)$ and $({}^2S)$ satisfy assumption 6.

NCA uniqueness can be tested by solving for $vec.\chi$ in (3.35). It was observed that matrix χ corresponding to $({}^1W)$ (Fig. 3.2) and $({}^2W)$ (Fig. 3.3) is not diagonal. Several off-diagonal entries are non-zero as shown in Fig. 3.1, where dashed line runs along the diagonal entries. Diagonal entries of χ are larger in magnitude compared to non-zero off-diagonal entries. Therefore diagonal entries of χ in Fig. 3.1 are set to zero to improve readability of the figure. From Fig. 3.1, it is evident that NCA solutions are not unique as (2.14) is not satisfied. Furthermore, though

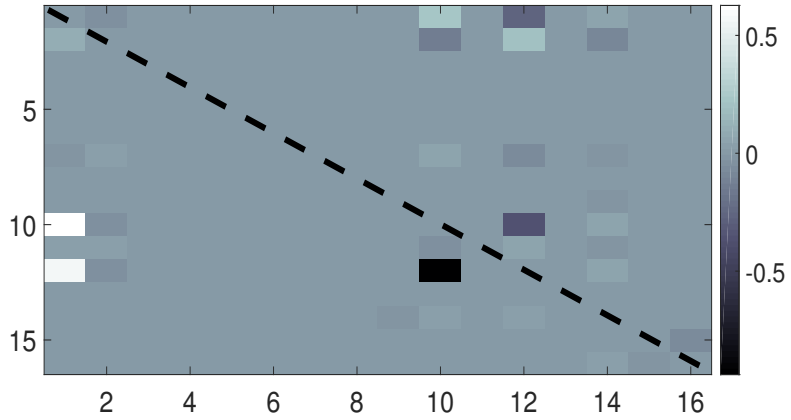

Figure 3.1: Colour map of χ after setting diagonal entries to zero

assumption 6 is satisfied,

$$({}^2S) \neq \chi^{-1}({}^1S) \tag{3.36}$$

Findings in this section demonstrates the fact mentioned in section 2.3.3 that sat-

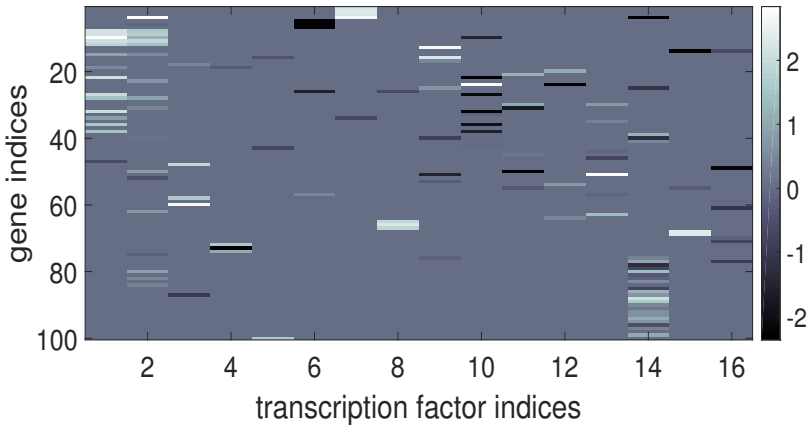

Figure 3.2: Colour map of $({}^1W)$ – carbon source switching experiment

isfying assumption 6 is not sufficient to show that NCA solutions are unique. All of the methods that extend NCA presented in section 2.3.2 are based on the assumption that NCA algorithms in [2] and [20] generate unique solutions. However, this is not true as demonstrated in this section. In chapter 6, a new divide and conquer method is proposed to obtain unique solutions to the SCMF problem in (2.12).

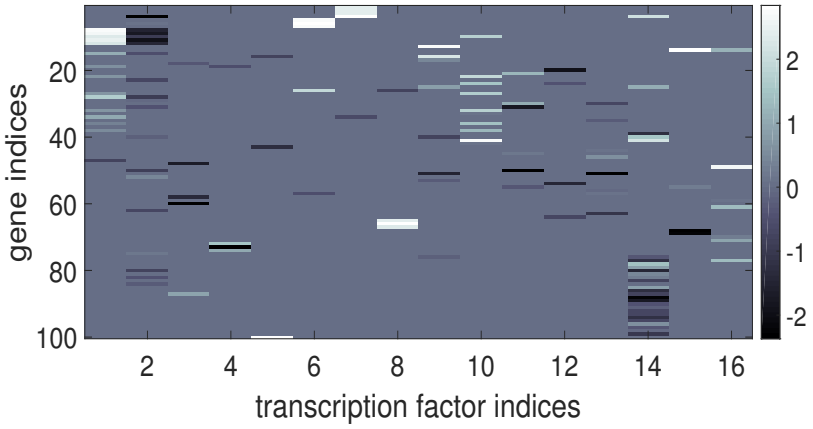

Figure 3.3: Colour map of $({}^{2}W)$ – carbon source switching experiment

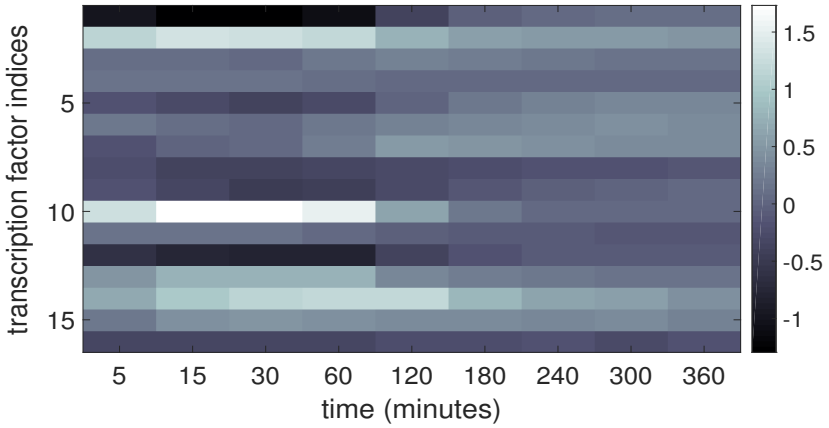

Figure 3.4: Colour map of $({}^{1}S)$ – carbon source switching experiment

3.4 Summary

In this chapter, two theoretical aspects of NCA – identifiability and uniqueness of solutions – are probed. Numerical examples are used to demonstrate the shortcomings of NCA. It is shown that the rank conditions imposed on the matrix factors in [2] are only partially accurate with the help of a counter example. In doing so, importance of binary rank of a network is demonstrated. A possible way to test existence of a solution is introduced. These open problems are solved in chapter 4. A graph based method to identify largest NCA sub-network available in literature assumes that size of a maximal bipartite matching is equal to graph rank. It is shown in this chapter that such an assumption is inaccurate. Graph operations such as breadth-first search, reduction and ordering are shown to be useful in identifying the largest full-ranked sub-matrix of a given binary matrix. These findings are used

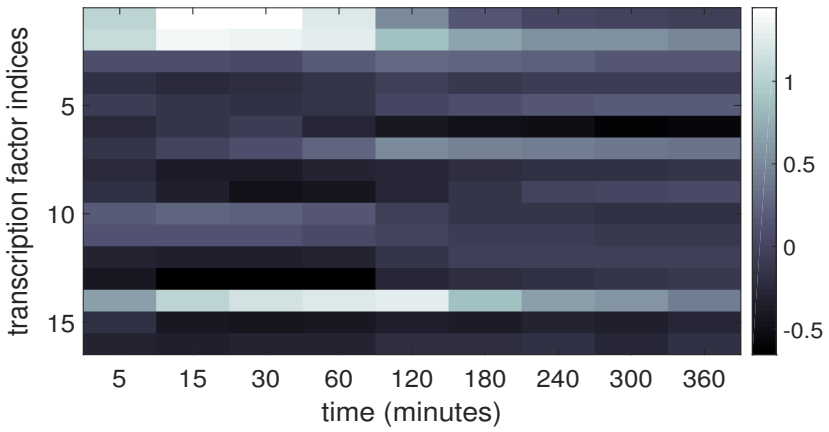

Figure 3.5: Colour map of $({}^2S)$ – carbon source switching experiment

in chapter 5 to identify the largest NCA feasible sub-network and decompose the whole network into multiple NCA feasible parts. It is shown in this chapter that rank conditions imposed on source signals matrix in [20] are insufficient to guarantee uniqueness of NCA solutions. This issue is resolved in chapter 6 where a method is developed to compute unique solutions to a general SCMF problem.

Chapter 4

NCA feasibility

NCA compatibility of a network, $D = WS$, is defined based on rank properties of W and S whereas only D and $B(G)$ are available in an application. In the context of this thesis, NCA compatibility refers to pair (W,S) that satisfies assumptions 1–3, or $B(G)$ that satisfies assumptions 4 and 5 (section 2.3). The definition of NCA compatible network can be used to test compatibility of a given network $B(G)$, but uniqueness of NCA solutions can be tested only after W and S are available. NCA compatible networks are those with a full-ranked S. It is impossible to use the original NCA compatibility definition [2] to test beforehand if a network satisfies this condition or not as S is a priori unknown. An alternate version of full rank condition is enforced on S in a modified NCA algorithm [20]. This algorithm has been used in its original form by several algorithms reviewed in [73], and those reported more recently in [40] and [74]. It is shown in section 3.3 that these conditions are inaccurate. No other work in literature has attempted developing a priori feasibility test. In addition to that, it is shown in section 3.1 that ignoring binary rank of $B(G)$ can lead to misclassification of $B(G)$ as NCA compatible. Therefore, there is a need to incorporate binary rank in the definition of NCA compatibility. Any priori NCA feasibility test should translate the rank conditions on the posteriori pair (W,S) to equivalent conditions on priori known pair $(D,B(G))$. This chapter addresses these challenges. Novel contributions in this chapter are as follows:

- Results from linear algebra, matrix theory and optimisation theory are used to compute a trivial solution in section 4.1.1. A proposition to characterise NCA feasible region is presented.
- In section 4.1.2, inaccuracies of assumptions 4 and 5 (chapter 2, section 2.3.3) are amended to accurately test NCA feasibility of $B(G)$.
- Bounds on $rk(S)$ based on pair $(D,B(G))$ are developed in section 4.2.

- NCA feasibility theorem is developed in section 4.3. This theorem can be used to test NCA feasibility of dataset-network pair $(D, B(G))$ before application of NCA.

Applicability of these novel results are demonstrated in section 4.4 with the help of dataset from a biological application described in section 2.5. In the context of this thesis, NCA feasibility refers to a pair $(D, B(G))$ that satisfies set of conditions developed in this chapter. Keyword *feasibility* is used to signify that all corresponding computations can be executed independent of and before applying NCA.

4.1 Accurate classification of structural constraints matrix

It is shown in section 3.1 that assumptions 4 and 5 (section 2.3.3) are invalid. In section 4.1.1, a linear algebra and optimisation based method is proposed to compute a trivial solution that not only respects the structural constraints imposed by underlying network, but also mimics the linear dependencies seen therein. Such a trivial solution is used to define a region in solution space that must be avoided and conditions that must be met to achieve uniqueness in NCA. Relationship between $rk_2(B(G))$ and $rk(W)$ developed in section 3.1.3 is used to modify assumptions 4 and 5.

4.1.1 A trivial solution

The goal in this section is to compute a trivial solution W_0, $\phi(W_0) = B(G)$ the rows and columns of which mimic in $\mathbb{R}$ linear dependencies exhibited in GF_2 by rows and columns of $B(G)$. Motivation to do so is to develop a method to compute a W_0 that is not full ranked whereas underlying $B(G)$ satisfies assumptions 4 and 5, but loses rank in GF_2. Such a pursuit is in the interest of NCA. If it is possible to compute such a W_0, then a region $I(B(G))$ defined by W_0 in NCA solution space $\mathcal{R}(B(G))$ can be avoided to obtain unique NCA solutions. Three regions of interest – $\mathcal{R}(B(G))$, $I(B(G))$, and $\mathcal{R}(B(G)) \setminus I(B(G))$ – are depicted in Fig. 4.1.

Binary matrices that record linear dependencies in $B(G)$ are defined in this section. These matrices are used to define network parameter subspace $I(B(G))$ and solve for W_0. Only those $B(G)$s are of interest in this section that satisfy one of the following two conditions:

1. $M > L$

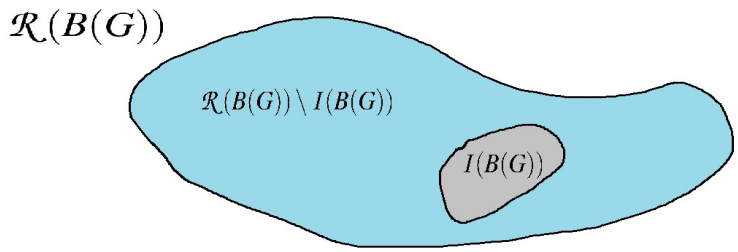

Figure 4.1: $\mathcal{R}(B(G))$ – NCA weights matrix solution space, $I(B(G))$ – subspace where W loses rank, $\mathcal{R}(B(G)) \setminus I(B(G))$ – subspace corresponding to unique NCA solutions

2. $M = L$ and $rk_2(B(G)) < L$

Consider a structural-constraints matrix $B(G) \in GF_2^{M \times L}$ with set of rows $Y = \{v_i : 1 \leq i \leq M\}$ and set of columns $X = \{u_j : 1 \leq j \leq L\}$. Let n_R and n_C respectively be the number of rows and columns in $B(G)$ that are linearly dependent in GF_2. Using rank-nullity theorem [105],

$$\begin{aligned} n_R &= M - rk_2(B(G)) \\ n_C &= L - rk_2(B(G)) \end{aligned} \tag{4.1}$$

Let $\psi_R \in GF_2^{n_R \times M}$ be such that each row $\psi_R(.,1:M)$ represents an equation in GF_2 connecting a linearly dependent row in $B(G)$ to its linearly independent rows. For example,

$$v_3 = v_1 \oplus v_2$$

maps to a row

$$\psi_R(.,1:M) = \begin{pmatrix} 1 & 1 & 1 & \mathbf{0}^{1 \times M-3} \end{pmatrix}$$

where, $\mathbf{0}^{1 \times M-3}$ is a vector of $M-3$ zeros. Similarly, let $\psi_C \in GF_2^{L \times n_C}$ be such that each column $\psi_C(1:L,.)$ represents an equation connecting a linearly dependent column to linearly independent columns in $B(G)$. $B(G)$ satisfies the following equation

$$\psi_R B(G) = \mathbf{0}^{n_R \times L} \tag{4.2}$$

and

$$B(G)\psi_C = \mathbf{0}^{M \times n_C} \tag{4.3}$$

In other words, ψ_C lies in the null-space of $B(G)$ evaluated over GF_2

$$\psi_C \in N_2(B(G)) \tag{4.4}$$

Similarly,

$$\psi_R^T \in N_2(B(G)^T) \tag{4.5}$$

Two definitions of a NCA network parameter space are presented next.

Definition 3 *Given a network $B(G)$, a network parameter subspace $I(B(G)) \subset \mathcal{R}(B(G))$ is defined as a space of real valued matrices such that any $W_0 \in I(B(G))$ satisfies*

1. $\phi(W_0) = B(G)$
2. $\psi_R W_0 = \mathbf{0}_{n_R \times L}$
3. $W_0 \psi_C = \mathbf{0}_{M \times n_C}$

where, ψ_R and ψ_C are respectively computed as given in (4.5) and (4.4). An alternate definition is provided here

Definition 4 *Given a network $B(G)$, a network parameter subspace $I(B(G)) \subset \mathcal{R}(B(G))$ is defined as*

$$I(B(G)) = \{W_0 \in N(\psi_R) \cap N^T(\psi_C^T) : \phi(W_0) = B(G)\}$$

Note that definitions 3 and 4 are different in that the former defines network parameter space as a space of matrices whereas the latter defines the same as a constrained vector space. Either of the two definitions can be used to devise a computational method to compute a trivial solution. In this section, a linear algebra and optimisation based method is proposed in line with definition 4 by solving for W_0 in

$$\begin{gathered} \psi_R W_0 \psi_C = \mathbf{0}_{n_R \times n_C} \\ \text{sub. to } \phi(W_0) = B(G) \end{gathered} \tag{4.6}$$

On vectorising equation (4.6), we have

$$\Psi vec.W_0 = \mathbf{0}^{ML \times 1} \tag{4.7}$$

where, $\Psi \in GF_2^{(n_R L + n_C M) \times ML}$ is determined as

$$\Psi = \begin{pmatrix} I_L \otimes \psi_R \\ \psi_C^T \otimes I_M \end{pmatrix} \tag{4.8}$$

In (4.8), $\otimes$ represents Kronecker product, I_M and I_L are identity matrices of size M and L. $n_R > 0$ follows from the assumption made at the beginning of this section and

$$rk_2(\Psi_R) = n_R$$
$$rk_2(\Psi_C) = n_C$$

However, Ψ is not always full-ranked and a solution to (4.7) might not exist. Assuming that a solution exists, solving for $vec.W_0$ (4.7) does not guarantee that $\phi(W_0) = B(G)$. This is because Ψ captures all linear-dependencies in $B(G)$, but not the structural information. Therefore, a pruned version $\bar{\Psi}$ of Ψ is obtained by removing every $(j-1)M+i$ th column, where $B(G)(i,j) = 0$. Let Z be the number of zeros in $B(G)$. $ML - Z$ non-zero entries of $vec.W_0$, denoted by $vec.\bar{W}_0$, can be computed by solving for null-space of $\bar{\Psi}$ as

$$\bar{\Psi} vec.\bar{W}_0 = \mathbf{0}^{(ML-Z)\times 1} \tag{4.9}$$

$\bar{\Psi}$ may not be full-ranked and hence, solution to (4.9) might not exist. Even if a solution exists, it may not be unique as $\bar{\Psi}$ may be non-square/fat. $vec.W_0$ can be obtained by inserting Z zeros in $vec.\bar{W}_0$ in appropriate locations and matrix W_0 can be obtained by rearranging $vec.W_0$. A straightforward way to solve the problem in (4.9) is

$$\bar{W}_0 = \bar{\Psi}^{-1}\mathbf{0}^{(ML-Z)\times 1} \tag{4.10}$$

However, it cannot be guaranteed that all the entries in $\bar{W}_0$ are non zero. From lemma 1, $rk(W_0) = rk(B(G))$ is true only if all entries of $vec.\bar{W}_0$ are non-zero. This can be achieved by reformulating problem in (4.9) as an optimisation problem with element-wise absolute value constraints

$$\begin{aligned} \min \quad & \|vec.\bar{W}_0\|_2^2 \\ \text{sub. to} \quad & \bar{\Psi} vec.\bar{W}_0 = 0 \\ & |vec.\bar{W}_{0,i}| \geq w \end{aligned} \tag{4.11}$$

where, $w > 0$ is an arbitrary constant dependent on application of interest and $\|.\|$ is the standard two-norm. Two-norm [41] of a vector $\mathbf{a} \in \mathbb{R}^m$ is defined as

$$\|\mathbf{a}\|_2^2 = \sum_{i=1}^{m} \mathbf{a}(i)^2$$

Optimisation method [106] with nonlinear inequality constraints can be used to solve this problem. Ideally, a W_0 that solves (4.11) is such that $W_0(i,j) \neq 0$ for any $i,\ j$ such that $B(G)(i,j) = 1$ and hence, $rk(W_0) = rk_2(B(G))$. Such a W_0 is not full

ranked and hence, is not NCA compatible. However, a solution may not always exist as the feasible region in (4.11) is not convex, i.e., there is a discontinuity at 0.

The steps to compute a trivial solution proposed in this section are summarised in the form of an algorithm in 2.

Algorithm 2 Trivial NCA solution

input: $B(G) \in GF_2^{M\times L}$
STEP 1: compute ψ_C (4.3), ψ_R (4.2), and Ψ (4.8)
STEP 2: determine $\bar{\Psi}$ by deleting row $(j-1)M+i$ from Ψ whenever $B(i,j) = 0,\ 1 \leq i \leq M,\ 1 \leq j \leq L$
STEP 3: solve (4.11) to obtain $\bar{W}_0$
STEP 4: insert 0 in $vec.\bar{W}_0$ wherever appropriate and rearrange to obtain W_0
output: W_0

It is important to develop a meaningful interpretation of results in this section. As mentioned at the beginning of this section, the goal in this section is not to fix the issues raised by using rk_2 in the context of NCA. Instead, the objective is to identify a trivial solution so as to identify that region which corresponds to a low rank NCA incompatible network. Excluding $I(B(G))$ defined by W_0 from NCA solution space $\mathcal{R}(B(G))$ might allow computation of unique NCA solutions despite $rk_2(B(G)) < rk(B(G))$. This interpretation is summarised in the form of a proposition

Proposition 1 *Let $B(G) \in GF_2^{M\times L}$ be such that it satisfies assumptions 4 and 5, and*

$$rk_2(B(G)) < L$$

or

$$rk_2(B(H_j)) < L-1$$

for some sub-matrix $B(H_j)$, $H_j \subset G, 1 \leq j \leq L$. If there exists a solution W_0 to (4.11), NCA solutions that are unique up to a diagonal matrix can be computed by solving for W in the subspace characterised by

$$\mathcal{R}(B(G)) \setminus I(B(G))$$

□

Results in this section are not included in NCA framework in this thesis as they need further refinement. This is considered as a candidate for future work.

4.1.2 NCA feasible networks

Replacing W with $B(G)$ in original NCA conditions leads to inaccuracies as shown in section 3.1.2. This is because $B(G)$ is binary valued matrix. NCA feasibility conditions for $B(G)$ can be designed by considering rk_2 instead of rk. However, such a choice must be theoretically justified. This can be done by relating $rk(B(G))$ to $rk(W)$. From lemma 1

$$rk(W) \geq rk_2(B(G))$$

This is true only when $|W(i,j)| > 0$ for any $i,\ j$ such that $B(G)(i,j) = 1$. In this section, it is assumed that this is true even though NCA does not explicitly impose such a constraint on W. This partially justifies the decision to replace real valued matrices in assumptions 1 and 2 with corresponding binary valued structural constraints matrices. However, rk may overestimate ranks of binary matrices as shown in section 3.1.2. Lemma 1 can be used to bridge this gap by replacing rk with rk_2.

Assumption 7 *$B(G)$ has full column rank, i.e., $rk_2(B(G)) = L$*

Assumption 8 *All reduced sub-matrices $B(H_j)$ obtained by removing every i-th row with $B(G)(i,j) \neq 0$ have full column rank, i.e., $rk_2(B(H_j)) = L-1$*

NCA feasibility conditions for $B(G)$ can now be developed as

Theorem 2 *A network $B(G) \in GF_2^{M\times L}$ is NCA feasible if $B(G)$ satisfies assumptions 7 and 8*

Proof Assume $B(G)$ satisfies assumptions 7 and 8. Consider an arbitrary matrix $W \in \mathbb{R}^{M\times L}, \phi(W) = B(G)$. From lemma 1, $rk(W) \geq rk_2(B(G))$. Since, $M \geq L$, $rk(W) \ngeq L$ and hence,

$$rk(W) = rk_2(B(G))$$

Similarly, sub-matrices $W_j, 1 \leq j \leq L, \phi(W_j) = B(H_j)$ corresponding to sub-graphs $H_j \subset G$ are such that

$$rk(W_j) = rk_2(B(H_j))$$

Therefore, W is NCA compatible as it satisfies assumptions 1 and 2. Hence, $B(G)$ is NCA feasible. □

Assumptions 7 and 8 can be tested before applying NCA. Therefore, these can be used to test NCA feasibility of a given network $B(G)$. Results in this section are combined with those in the next section to develop a NCA feasibility test for a dataset-network pair $(D, B(G))$ arising from an application.

4.2 Bounds on rank of input matrix

Uniqueness of NCA solutions can be guaranteed only if $rk(S) = L$ for $S \in \mathbb{R}^{L \times N}$. Though an attempt to enforce uniqueness has been made in [20] by introducing assumption 6 (section 2.3.3), an example where it fails is presented in section 3.3. It is impossible to compute $rk(S)$ prior to applying NCA. The only way to guarantee uniqueness of NCA-solutions in state-of-art is to obtain estimates of W and S, and test if assumptions 1 –3 are satisfied.

In previous section, a method to test NCA feasibility of $B(G)$ was presented. In this section, it is shown that NCA solution may not be unique despite $B(G)$ being NCA feasible. An obvious case is one where $N < L$ which is common in most of the TRN applications [81]. It is trivial to see that NCA solutions are not unique in this case as S will not have full row rank. There can be cases in which S losing rank is not obvious. In this section, theoretical bounds on $rk(S)$ are developed based on priori available information – D and $B(G)$ so as to avoid such cases.

Consider an augmented matrix $[B(G) : D]$. Using the result in [107], $rank([B(G) : D])$ can be computed as

$$rk([B(G) : D]) = rk(B(G)) + rk((I_M - B(G)B(G)^{-})D) \tag{4.12}$$

where, I_M is an identity matrix of size M and $B(G)^{-}$ is a generalised inverse of $B(G)$.

$$rk(D) = rk(WS) \leq min(rk(W), rk(S)) \tag{4.13}$$

From (4.12) and (4.13)

$$rk((I_M - B(G)B(G)^{-})D) \leq \min(rk((I_M - B(G)B(G)^{-})W), rk(S))$$

$$\implies rk((I_M - B(G)B(G)^{-})D) \leq rk(S) \tag{4.14}$$

Original NCA assumes $L \leq N$ and hence,

$$rk(S) \leq L$$

However, in a general application this may not be true. Therefore, rank of S is bounded above as

$$rk(S) \leq \min(L, N) \tag{4.15}$$

From (4.14) and (4.15),

$$rk([B(G):D]) - rk(B(G)) \leq rk(S) \leq \min(L,N) \tag{4.16}$$

The upper bound in (4.16) is used in [20] to show that NCA uniqueness can be achieved despite $L > N$. Note that the objective here is only to test whether NCA can be applied to $(D, B(G))$ or not. Guaranteeing uniqueness in cases where $L > N$ requires additional set of theoretical results developed in chapter 5 and 6.

Theoretical bounds in (4.16) can be computed before applying NCA as both D and $B(G)$ are available. There are two ways of interpreting (4.16)

1. Uniqueness of NCA solutions can be tested apriori whenever L is fixed
2. size of the largest NCA feasible sub-network can be determined based on number of data points N

First of these interpretations is used in the next section to develop two NCA feasibility theorem whereas the latter is used to motivate discussions in the next chapter.

It was demonstrated in section 3.1.2 that binary rank rk_2 is important in NCA setting. This raises an important question on usage of $rk(B(G))$ in (4.16). Answering this question requires further analysis which is not necessary as only the upper bound on $rk(S)$ is of importance in NCA setting. As a result, $rk(B(G))$ is retained in (4.16).

4.3 NCA feasibility theorem

NCA feasible networks are characterised in section 4.1.2. Theoretical bounds on $rk(S)$ based on priori available information are presented in section 4.2. In this section, results from previous two sections are combined to characterise NCA feasible dataset-network pair $(D, B(G))$.

Theorem 3 *A dataset-network pair* $(D \in \mathbb{R}^{M\times N},\ B(G) \in GF_2^{M\times L})$ *is NCA feasible if*

1. $B(G)$ *satisfies assumption 7 and 8*
2. $rank([B(G):D]) = 2L$

Proof From theorem 2, $B(G)$ is NCA feasible if assumptions 7 and 8 are satisfied. Furthermore,

$$rk(B(G)) = L$$

and there exists a $W \in \mathbb{R}^{M\times L}, \phi(W) = B(G)$ that satisfies assumptions 1 and 2. If $rank([B(G) : D]) = 2L$, then from (4.16)

$$L \leq rank(P) \leq L$$

$$\implies rank(P) = L$$

Thus, assumptions 1–3 of the original NCA theorem [2] are satisfied. Hence, the dataset-network pair is NCA compatible. As all the conditions in this theorem can be tested before applying NCA, pair $(D, B(G))$ is NCA feasible. □

Applicability of this theorem is demonstrated with the help of two examples in next section.

4.4 Simulation results

In this section, applicability and limitations of the results developed in this chapter are demonstrated. Discussions in this section are organised in two subsections. Synthetic numerical examples, systems 1 and 2, are presented in section 4.4.1 to explore applicability of algorithm 2 and proposition 1. In section 4.4.2, dataset from a biological experiment [1] is used to demonstrate applicability of algorithm 2 and theorem 3.

4.4.1 Synthetic numerical examples

Two networks are studied in this section – Network 1 – NCA infeasible with a trivial solution and Network 2 – NCA infeasible with no trivial solution. In the context of structural constraints, no trivial solution means some of the parameters $W_0(i, j) = 0$ whereas $B(G)(i, j) = 1$. In such cases, structural constraints are not respected as $\phi(W_0) \neq B(G)$.

Network 1 – Numerical example in (3.1) is shown to be NCA infeasible in section 3.1.2 with the help of carefully chosen W (3.7). When algorithm 2 is applied to a

$B(G)$, a trivial solution W_0 is obtained as

$$W_0 = \begin{pmatrix} 0 & 0 & w & -w \\ 0 & w & 0 & -w \\ 0 & -w & w & 0 \\ -w & 0 & 0 & w \\ w & 0 & -w & 0 \\ -w & w & 0 & 0 \end{pmatrix} \tag{4.17}$$

with $w = 0.0031$. W_0 obtained by solving for $\bar{W}_0$ as in (4.10) is equal to the matrix in (4.17) with $w = 0.2887$. W in (3.7) can be obtained by setting $w = 2$ in (4.17). W_0 (4.17) seems to represent a general solution to this system that can be used to define $I(B(G))$.

Network 2 – Consider a network $B(G)$ given by

$$B(G) = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \end{pmatrix} \tag{4.18}$$

It can be verified that $B(G)$ in (4.18) does not satisfy assumptions 7 and 8 and is NCA infeasible. W_0 determined from $\bar{W}_0$ (4.10) is given by

$$W_0 = \begin{pmatrix} 0 & w & 0 & -w \\ 0 & 0 & 0 & 0 \\ 0 & -w & 0 & w \\ -w & 0 & 0 & w \\ w & 0 & 0 & -w \\ -w & w & 0 & 0 \\ w & -w & 0 & 0 \end{pmatrix} \tag{4.19}$$

with $w = 0.2887$. Third column of W_0 (4.19) is zero and $rk(W_0) = 3$. This means to say that some of the linear dependencies in $B(G)$ (4.18) cannot be captured by W_0, i.e. $\phi(W_0) \neq B(G)$. This observation is consistent with the fact that any solution to the optimisation problem in (4.11) will converge to an infeasible point.

Remarks – networks 1 and 2 show that the results in section 4.1.1 have some potential in going a step beyond NCA. Both networks are NCA infeasible, but the former might allow application of NCA if resulting W lies outside a space $I(B(G))$ defined by general-trivial- solution W_0. This is possible as network 1 satisfies 4 and 5. In the latter case, neither does a trivial solution exist nor does the network satisfy assumption 5. These observations show that there is an opportunity to explore further differences between rk and rk_2 based NCA rank conditions. This is not dealt with in this thesis, but is a potential candidate for future work.

4.4.2 Carbon source switch - NCA feasibility

Consider the transcriptional regulatory network $B(G)$ (Fig. 2.4, section 2.5). The following facts are mentioned in section 2.5

- $B(G)$ has $M = 100$ rows and $L = 16$
- $B(G)$ satisfies assumptions 4 and 5
- dataset D made available by the authors of [1] has only $N = 9$ data points
- pair $(D, B(G))$ is NCA incompatible as $rk(S) \leq N < L$

It is shown in section 3.3 that applying NCA to $(D, B(G))$ results in solutions that satisfy assumptions 1, 2, and 6. However, in the same section, the solutions are shown to be non unique as χ that scales both matrix factors W and S could not be found. In this section, results from this chapter are used to identify a data subset $D^{(1)}$ and a sub-network $B(H^{(1)})$ of $B(G)$ such that pair $(D^{(1)}, B(H^{(1)}))$ is NCA feasible.

It can be verified that $B(G)$ satisfies assumptions 7 and 8. From theorem 2, $B(G)$ is NCA feasible. However, pair $(D, B(G))$ is NCA infeasible as $L > N$. However, (4.16) can be used to identify lower and upper bounds on $rk(S)$. In this case, $\min(L, N) = N$, $rk(B(G)) = 9$ and

$$rk([B(G) : D]) = 25$$

From (4.16),

$$\begin{aligned} & 9 \leq rk(S) \leq 9 \\ \implies & rk(S) = 9 \end{aligned} \tag{4.20}$$

This implies that applying NCA to $(D, B(G))$ will result in a matrix factor S with rank 9. This agrees with outcome of simulations in section 3.3 where it is also shown that NCA solutions are not unique up to a diagonal scaling matrix.

Consider a sub-network $B(H^{(1)})$ that consists of first 9 columns of $B(G)$ and corresponding non zero rows as depicted in Fig. 4.2. Neither the result in (4.16)

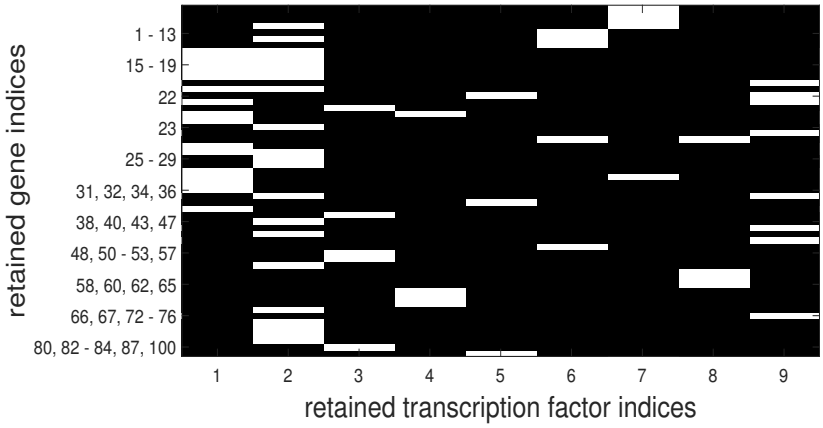

Figure 4.2: Colour map of $B(H^{(1)})$ (0 – black, 1 – white): NCA feasible sub-network of transcriptional regulatory network in [1]

nor the conditions in theorem 3 can be used to identify $B(H^{(1)})$. A method to identify NCA feasible sub-networks of a NCA infeasible network is presented in next chapter.

From theorem 2, $B(H^{(1)})$ is NCA feasible as it satisfies assumptions 7 and 8. Pair $(D^{(1)}, B(H^{(1)}))$ is NCA feasible as

$$rk([B(H^{(1)}) : D^{(1)}]) = 18$$

and

$$rk(B(H^{(1)})) = 9$$

$$\implies 9 \leq rk(S^{(1)}) \leq 9$$

In order to verify that NCA solutions are indeed unique up to a diagonal matrix, NCA is applied twice to the pair $(D^{(1)}, B(H^{(1)}))$ to obtain two pairs of estimates of matrix factors – $(({}^1W^{(1)}), ({}^1P^{(1)}))$ and $(({}^2W^{(1)}), ({}^2P^{(1)}))$. $({}^1W^{(1)})$ and $({}^2W^{(1)})$ are as illustrated respectively in Fig. 4.3 and 4.4. It can be seen in Fig. 4.3 that the maximum and minimum values on scale to the right of the colour map are respectively > 2.5 and < -1.5. Maximum and minimum values on scale in Fig. 4.4 respectively are < 2.5 and < -2. Columns 2, 3, and 7 in Fig. 4.3 and 4.4 are visibly different.

Similar observations can be made in colour maps of $({}^1S^{(1)})$ and $({}^2S^{(1)})$ depicted in Fig. 4.5 and 4.6.

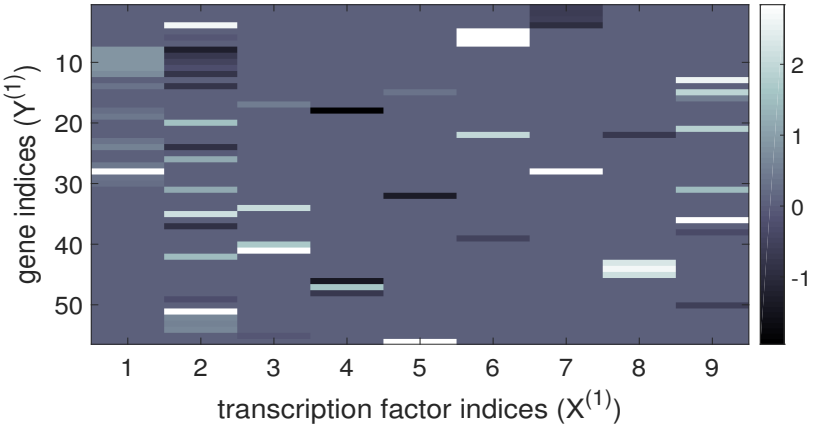

Figure 4.3: Colour map of $({}^1W^{(1)})$: NCA feasible sub-network of transcriptional regulatory network in [1]

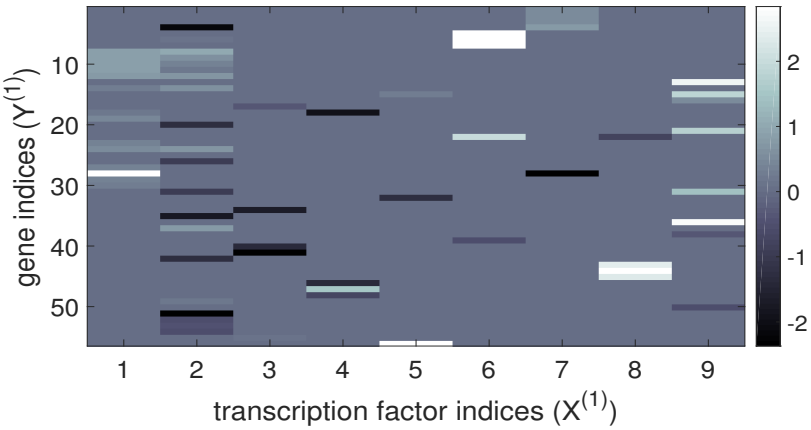

Figure 4.4: Colour map of $({}^2W^{(1)})$: NCA feasible sub-network of transcriptional regulatory network in [1]

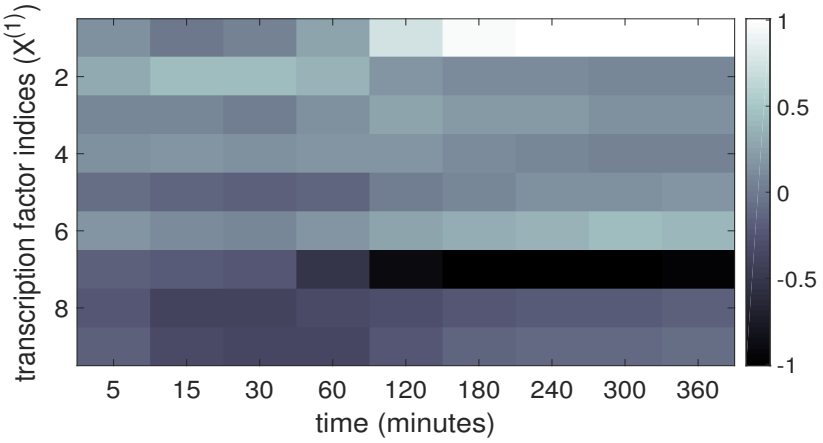

Figure 4.5: Colour map of $({}^{1}S^{(1)})$: NCA feasible sub-network of transcriptional regulatory network in [1]

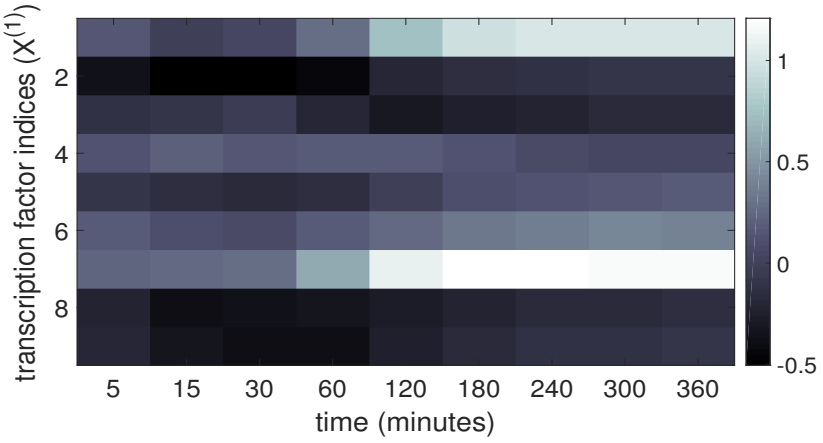

Figure 4.6: Colour map of $({}^{2}S^{(1)})$: NCA feasible sub-network of transcriptional regulatory network in [1]

Using the method presented in section 3.3, chapter3, $\chi^{(1)}$ is determined as

$$\chi^{(1)} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0.84 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.83 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.97 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.99 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -0.83 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.06 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.97 \end{pmatrix} \tag{4.21}$$

$\chi^{(1)}$ satisfies (2.14) (section 2.3.3, chapter 2). Thus, NCA solutions of $(D^{(1)}, B(H^{(1)}))$ are indeed unique. It is important to note that χ is not unique with respect to a given dataset-network pair. Value of χ depends on the values of NCA solutions being tested for uniqueness. In this section, value of $\chi^{(1)}$ in (4.21) is dependent on $(({}^1W^{(1)}), ({}^1S^{(1)}))$ and $(({}^2W^{(1)}), ({}^2S^{(1)}))$. Value of $\chi^{(1)}$ changes if uniqueness of $(({}^1W^{(1)}), ({}^1S^{(1)}))$ is tested with respect to an arbitrary third NCA solution $(({}^3W^{(1)}), ({}^3S^{(1)}))$. However, from theorem 3, $(D^{(1)}, B(H^{(1)})$ is NCA feasible and hence, $\chi^{(1)}$ is guaranteed to be diagonal in nature.

Remarks – applicability of theorem 3 is demonstrated with the help of a real application data in this section. It is shown that the bounds on $rk(S)$ developed in (4.16) can be used to address cases of NCA infeasibility. However, the choice of sub-network $B(H^{(1)})$ (Fig. 4.2) is not explained in this section. Challenge of developing a method to identify NCA feasible sub-networks of NCA infeasible networks is taken up in the next chapter.

4.5 Summary

In this chapter, conditions that a dataset-network pair must satisfy for NCA to be applicable is developed. One of the major challenges in developing a feasibility test for NCA is that of developing accurate rank conditions for binary valued matrix representing the underlying network. In order to address this issue, NCA feasibility of a network is defined based on its binary rank. The other major hurdle in developing NCA feasibility test is that source matrix cannot be tested for original NCA rank condition. In this chapter bounds on rank of source matrix are developed based on priori available information which can be tested before applying NCA. In addition to successfully achieving the goal of developing a priori NCA feasibility test, mathematical description of a region in NCA solution space is developed with the help of linear algebra and optimistaion methods. Though this is applicable only to a particular type of networks, it provides a better characterisation of NCA feasible region. The case where such a region cannot be described is not addressed in this chapter. A proposition to extend definition of NCA compatibility is presented where the idea is to avoid this region while computing NCA solutions. Furthermore, it is shown with the help of simulation that NCA feasible sub-network of a NCA infeasible network can be uniquely factorised. This finding serves as a motivation for the next chapter.

Chapter 5

Full-rank factorisable sub-networks

According to theorem 3, a network $B(G)) \in GF_2^{M\times L}$ and a dataset $D \in \mathbb{R}^{M\times N}$ are NCA feasible if $B(G)$ satisfies the conditions of theorem 2 and $rk([B(G)\ :\ D]) = 2L$. Unique solutions to an associated structure constrained matrix factorisation problem (2.12) cannot be sought if either of the two conditions are not satisfied. The goal in this chapter is to identify full-rank factorisable sub-networks of a given NCA infeasible dataset-network pair. Theoretical results and algorithms developed in this chapter decompose the given structure constrained matrix factorisation problem into a set of sub-problems each working with a data-subset-sub-network pair $(D^{(i)}, B(H^{(i)}))$. In this chapter, it is shown that each identified sub-network is NCA feasible and the associated source signal matrix is full-ranked. Hence the name *full-rank factorisable sub-networks*. Solving the sub-problems, uniqueness of the obtained solutions and finding a solution to the original structure constrained matrix factorisation problem (2.12) are addressed in the next chapter.

As pointed out in section 2.4, bipartite matching based method can be used to develop graph theoretical conditions equivalent to the NCA rank conditions in theorem 1. Graph theoretical methods developed in [82] are based on the assumption that rank $rk(B(G))$ of a bipartite graph G and size of a maximal matching in G are equal. However, it was shown in section 3.2 that such an assumption is inaccurate. In sections 3.2.2 and 3.2.3 importance of a breadth-first matching in G after removing duplicate vertices and reordering the remaining vertices in calculating $rk_2(B(G))$ is demonstrated. This chapter builds on the preliminary results in sections 3.2.2 and 3.2.3.

Three novel contributions in this chapter are

1. a method to calculate $rk_2(B(G))$ based on graph reduction, vertex reordering and breadth-first matching is developed in section 5.2

2. accurate graph theoretical conditions are developed in section 5.2 equivalent to the original NCA rank condition (theorem 1)

3. Two algorithms are presented in section 5.3 to identify full-rank factorisable sub-networks of $B(G)$ when the given dataset-network pair $(D, B(G))$ is NCA infeasible

Note that in this chapter reduction refers to elimination of duplicate and isolated vertices. sub-graphs corresponding to sub-matrices $B(H_j)$ (3.2), referred to as reduced sub-matrices in the context of NCA, are induced sub-graphs given by

$$H_j = (X_j \cup Y_j, B(H_j)) \equiv G \setminus \{u_j, \mathcal{N}(u_j)\} \tag{5.1}$$

where, X_j is defined as

$$X_j = X \setminus u_j$$

and Y_j as

$$Y_j = Y \setminus \mathcal{N}(u_j)$$

5.1 Bipartite matching and rank

It is mentioned in passing in [82] that $rk(B(G))$ is equal to the size of a maximal matching. As pointed out in section 2.3.3, structural rank, not the rank, of a sparse matrix is equal to the size of a maximal matching. This is not in the interest of NCA and hence, structural rank of $B(G)$ is not discussed in this thesis. In section 3.2, a counter example is used to demonstrate that removing duplicates and isolated nodes followed by reordering rows and columns can potentially lead to accurate estimation of $rk(B(G))$. In this section, a relationship between a particular type of matching and $rk_2(B(G))$ is developed. This relationship is used later in this chapter to develop graph theoretical NCA feasibility conditions.

5.1.1 Reduced ordered matching

Consider a bipartite graph $G = (X \cup Y, B(G), W)$, where $B(G)$ and W are as defined in section 2.4 with $\phi(W) = B(G)$. $X = \{u_j, 1 \leq j \leq L\}$ and $Y = \{v_i, 1 \leq i \leq M\}$ are set of vertices that respectively represent columns and rows of $B(G)$. In order to develop accurate graph theoretical interpretation of assumptions 4 and 5, three graph operations are considered

1. *Reduce*: determine a set of vertices $X_r \subseteq X$ and $Y_r \subseteq Y$ by eliminating duplicate and isolated vertices in G

2. *Order*: determine posets $X_{r<}$ and $Y_{r<}$ by sorting X_r and Y_r in increasing order of degrees of their elements

3. *Match*: find the largest possible matching $M_{(G_{r<})}$ in the ordered-reduced graph $G_{r<} = (X_{r<} \cup Y_{r<}, \cdots)$ using a breadth-first search strategy

A novel concept of ordered matching is introduced as

Definition 5 *An ordered matching $M_{(G_<)}$ is a breadth-first matching in an ordered graph $G_<$*

Definition 6 *A reduced ordered matching (ROM) $M_{(G_{r<})}$ is a breadth-first matching in a reduced and ordered graph $G_{r<}$*

Ordering a graph corresponds to permutations of rows and columns of $B(G)$. Permutations are edge preserving bijections and hence, a graph G is isomorphic [108] to its ordered version $G_<$

$$G \equiv G_<$$

Reducing a graph removes duplicate rows and columns of $B(G)$. As G is isomorphic to $G_<$, same vertices get deleted from G and $G_<$ when reduced. Thus, sequentially ordering and reducing G is equivalent to sequentially reducing and ordering G

$$G_{r<} \equiv G_{<r}$$

However, breadth-first matching can be different in $G_<$ and G_r. Therefore, breadth-first matching is obtained only after reduction and ordering.

Remark 1 *$M_{(G_{r<})}$ can be uniquely determined using the following rules:*

- $B(G)$ is ordered starting from the top most row and left most column. When two vertices $u_i,\ u_j \in X,\ i, j \in \mathbb{R},\ i < j$ are such that $|\mathcal{N}(u_i)| = |\mathcal{N}(u_i)|$, then column corresponding to u_i precedes that corresponding to u_j in $B(G)$. Similarly, row corresponding to v_i precedes that corresponding to v_j if $|\mathcal{N}(u_i)| = |\mathcal{N}(u_i)|$ for some vertices $v_i,\ v_j \in Y,\ i, j \in \mathbb{R},\ i < j$

- A graph is reduced as follows: if for some $u_i,\ u_j \in X,\ i, j \in \mathbb{R},\ i < j$ are such that $\mathcal{N}(u_i) \equiv \mathcal{N}(u_i)$, then u_i is retained and u_j is deleted

These rules for ordering and reducing graphs makes it possible to uniquely determine $M_{(G_{r<})}$ as same nodes will be picked by a matching algorithm every time.

5.1.2 Determining rank based on reduced ordered matching

The relationship between reduced ordered matching (definition 6) and rank of a graph is established as follows

Lemma 2 *Rank over $\mathbb{R}$ of a bipartite graph $G = (X \cup Y, B(G))$ can be determined as*

$$rk(G) = |M_{(G_{r<})}|$$

where, $M_{(G_{r<})}$ is as defined in definition 6

Proof Rank of a graph is equal to the rank of reduced graph G_r [109]. Defining a partial order $<$ on vertices of G_r does not affect its rank as it corresponds to permutations of $B(G)$. Every linearly independent column in $B(G_{r<})$ corresponds to a pivot in the row-reduced echelon form of $B(G_{r<})$. A simple breadth-first search based reduced ordered matching $M_{(G_{r<})}$ will contain all the pivots. Therefore, it is sufficient to show that $M_{(G_{r<})}$ corresponds to a set of linearly independent columns in $B(G_{r<})$, and hence, in G_r and G.

Assume that there exists a vertex $u_i \in X_{r<}$ such that the corresponding column $B(G_{r<})(:,i)$ can be expressed as

$$B(G_{r<})(:,i) = \sum_{j \neq i} \alpha_j B(G_{r<})(:,j)$$

where, $B(G_{r<})(:,j)$s are linearly independent in $B(G_{r<})$. As a result of partial order imposed in $G_{r<}$, vertices corresponding to $B(G_{r<})(:,j)$s, u_js, are such that

$$|\mathcal{N}(u_i)| \geq |\mathcal{N}(u_j)|$$

Thus, for every neighbour of u_i, $v \in \mathcal{N}(u_i)$, there exists some $u_j \in X_{r<}, j \neq i$ such that $v \in \mathcal{N}(u_j)$. As the columns corresponding to u_js precede the column corresponding to u_i in $B(G_{r<})$, every $v \in \mathcal{N}(u_i)$ matches with some preceding vertex u_j. Therefore, u_i will not be a part of $M_{(G_{r<})}$. Thus, columns $B(G_{r<})(:,j)$s corresponding $u_j \in M_{(G_{r<})}$ are linearly independent and hence,

$$rk(B(G_{r<})) = |M_{(G_{r<})}|$$

$$\implies rk(B(G)) = |M_{(G_{r<})}|$$

A similar argument with respect to the rows of $B(G_{r<})$ by replacing u with v completes the proof. □

Lemma 2 can be used to identify linear dependencies in $\mathbb{R}$, but not in GF_2. No

graph theoretical method is presented in this thesis to identify linearly dependent rows or columns of $B(G)$ in GF_2. An algorithm that combines graph theoretical and linear algebraic methods to determine binary rank of $B(G)$, $rk_2(B(G))$ is presented next.

Algorithm 3 Removing columns linearly dependent in GF_2

input: G and corresponding $B(G) \in GF_2^{M\times L}$
set $G' = G_{r<}$ and $\psi_C = N_2(B(G'))$
while $\psi_C \neq \{\}$ **do**
  step 1 – for every column i of ψ_C, determine highest j such that $\psi_C(i,j) = 1$
  step 2 – update G' by deleting from $B(G')$ columns corresponding to js identified in step 1
  set $\psi_C = N_2(B(G'))$
end while
output: G'

Lemma 3 *Rank over GF_2 of a bipartite graph $G = (X \cup Y, B(G))$ can be determined as*

$$rk_2(G) = |M_{(G')}|$$

where, G' is determined using algorithm 3

Proof G' has no columns that are linearly dependent in GF_2. Therefore,

$$rk(B(G')) = rk_2(B(G'))$$

From lemma 2,

$$rk(B(G')) = |M_{(G')}|$$
$$\implies rk_2(B(G')) = |M_{(G')}|$$

□

5.2 Bipartite matching based NCA feasibility

In this section, novel graph theoretical interpretation of NCA rank conditions on weight matrix is developed. Rank of a given graph can be determined using reduce order matching as shown in lemma 2. This novel result is used to devise graph theoretical conditions that $B(G)$ must satisfy for G to be NCA feasible as

Theorem 4 *A bipartite graph $G = (X \cup Y, B(G), W),\ |X| \leq |Y|$ is NCA feasible if and only if for each $u_j \in X$*

$$|M_{(H'_j)}| = L - 1$$

where, graphs H'_j are determined using algorithm 3

Proof Condition $|X| \leq |Y|$ is imposed to make sure that the system of equations $D = WS$ is over determined as required by NCA (theorem 1, section 2.3.1). Assume every vertex $u_j \in X$ satisfies the condition

$$|M_{(H'_j)}| = L - 1$$

Then, from lemma 3,for all $u_j \in X$,

$$rk_2(B(H_j)) = L - 1$$

Adding u_j and any one of its neighbours $v \in \mathcal{N}(u_j)$ will add a pivot to H'_j. Therefore, upon adding all vertices $v \in \mathcal{N}(u_j)$ that constitute G'

$$|M_{G')}| = L$$

Thus, assumptions 7 and 8 (section 4.1.2) are satisfied and hence, G is NCA feasible.
Conversely, assume G is NCA feasible. $B(G)$ satisfies assumptions 7 and 8 by definition. From lemma 3 and assumption 7,

$$rk_2(B(G)) = L \implies |M_{G')}| = L$$

Similarly, from lemma 3 and assumption 8,

$$rk_2(B(H_j)) = L - 1 \implies |M_{(H'_j)}| = L - 1$$

Thus,

$$|M_{(H'_j)}| = L - 1$$

for all $u_j \in X$ if G is NCA feasible. □

Theorem 4 is equivalent to the rank conditions developed for NCA feasible networks in theorem 2, section 4.1.2.

5.3 NCA feasible sub-networks

Graph theoretical approach to assess NCA feasibility of a given network $B(G)$ is presented in theorem 4. As discussed in section 2.4, methods available in [82] and [60] produce the largest possible NCA feasible sub-network of an NCA infeasible

network. In this section, a reduced ordered matching based method is presented to decompose a given NCA infeasible graph $G = (X \cup Y, B(G))$ into a set of NCA feasible sub-graphs $H^{(g)},\ 1 \leq g \leq L$. The sub-graphs are defined as

$$H^{(g)} = (X^{(g)} \cup Y^{(g)}, \cdots) \tag{5.2}$$

where, $X^{(g)} \subseteq X$ are such that

$$X^{(a)} \cap X^{(b)} = \{\},\ a \neq b$$

and $Y^{(g)} = \mathcal{N}(X^{(g)}) \subseteq Y$. Identified sub-graphs are such that

$$G = \underset{g}{\cup} H^{(g)} \tag{5.3}$$

Thus, no part of the original graph is ignored as in the case of methods available in literature.

Relationship between $rk_2(G)$ and a maximal reduced ordered matching $M_{(G')}$ presented in lemma 3 is used in algorithm 4 to identify subsets $X^{(g)}$ of X. All

Algorithm 4 NCA feasible sub-graph identification

input: $B(G) \in GF_2^{M \times L}$
STEP 1: for every $u_j \in X$, identify $M_{(H'_j)}$
STEP 2: set $X_j^{(M)} = \{u_i : u_i \in M_{(H'_j)}\} \cup u_j$
STEP 3: set $g = 1$, $X^{(g)} = \{u_1\}$, and $I = \{1,\ 2,\ \cdots,\ L\}$
while $I \neq \{\}$ **do**
  for $i = 2$ to $|I|$ **do**
    if $X_{I(1)}^{(M)} \equiv X_{I(i)}^{(M)}$ **then**
      set $X^{(g)} = X^{(g)} \cup u_{I(i)}$
    end if
  end for
  set $I = I \setminus X^{(g)}$, $g = g + 1$
end while
outputs: $H^{(i)} = (X^{(i)} \cup \mathcal{N}(X^{(i)}), \cdots)$, for $i = 1, 2, \cdots, g$

vertices in a set $X^{(g)}$ form maximal ordered matching of the same size with other vertices in the same set.

In theorem 1, a dataset-network pair is assumed to satisfy the condition that $|X| \leq N$, i.e the number of source signals are lesser than the number of available data points. This assumption is not satisfied in several applications such as transcriptional regulatory networks [20, 81]. Algorithm 4 and other methods in litera-

ture to identify NCA feasible sub-networks do not address the case where $L > N$ and hence, $rank(S) < L$. As a result, a given dataset D cannot be factorised into product of matrices with full rank. In order to address this problem, algorithm 4 is extended by adding a step to limit size of groups $X^{(i)}$s as

$$|X^{(i)}| \leq n_{max} = rank([B(G) : D]) - rank(B(G)) \tag{5.4}$$

The upper bound on cardinality of sets $X^{(g)}$ in (5.4) is derived from the upper bound

Algorithm 5 Identifying full-rank factorisable NCA feasible sub-graphs

input: sets $X^{(i)}$ from algorithm 4 and sample size N
regrouping subroutine
set $i = 1, X^{(g+1)} = \{\}, f = 0$
while $i \leq g$ **do**
  if $|X^{(i)}| > n_{max}$ **then**
    set $X^{(g+1)} = X^{(g+1)} \cup X^{(i)}(n_{max} + 1 \text{ to } |X^{(i)}|)$
    set $X^{(i)} = X^{(i)}(1 \text{ to } n_{max})$
    $f = 1$
  end if
  $i = i + 1$
end while
repeat
  use algorithm 4 to decompose $H = (X^{(g+1)} \cup \mathcal{N}(X^{(g+1)}))$
  call **regrouping subroutine** for H
until $f = 0$
outputs: $H^{(i)} = (X^{(i)} \cup \mathcal{N}(X^{(i)}), \cdots)$, for $i = 1, 2, \cdots, g$

on rank of S given in section 4.2, (4.16). Limiting the size of groups in algorithm 4 ensures that

$$rank(B(H^{(i)})) = rank(S^{(i)}) = n_{max},\ i = 1,\ 2,\ \cdots,\ g$$

Extended graph decomposition algorithm 5 can be used to solve a general SCMF problem (2.12) as long as dataset D satisfies the condition $M \geq L$. The newly formed sub-graphs are full-rank factorisable as the sub-networks $B(H^{(i)})$s satisfy the conditions in theorem 2 and corresponding source signal matrices $S^{(i)}$s are guaranteed to be full-ranked. However, it cannot be guaranteed that all data-subset-sub-network pairs $(D^{(i)}, B(H^{(i)}))$ satisfy the following condition in theorem 3

$$rk([B(H^{(i)})\ :\ D^{(i)}]) = 2L^{(i)}$$

where, $L^{(i)} = |X^{(i)}|$ is the number of source signals in $H^{(i)}$. This fact is demonstrated

in section 5.4.2 with the help of an NCA infeasible experimental dataset. Therefore, algorithm 5 is titled *Identifying full-rank factorisable NCA feasible sub-graphs* and not *Identifying NCA feasible data subset-sub-graph pair*.

5.4 Simulation results

In this section, applicability of lemma 3 and theorem 4 are demonstrated with the help of

- example network $B(G)$ (3.1) presented in section 3.1
- carbon source switching network $B(G)$ (Fig. 2.4) presented in section 2.5)

It is shown that the example network is NCA infeasible whereas the carbon source switching network is NCA feasible. Therefore, algorithm 4 is applied to the example network to obtain NCA feasible sub-networks. On the other hand, it can be verified that the carbon source switching network is NCA feasible. However, the corresponding dataset D contains 9 data points as pointed out in section 2.5 whereas the number of source signals is 16. Hence, dataset-network pair $(D, B(G))$ corresponding to the carbon source switching network is NCA infeasible. Therefore, algorithm 5 is applied to the carbon source switching network to identify NCA feasible sub-networks.

5.4.1 Example network

Consider $B(G)$ described in (3.1). $B(G)$ represents a bipartite graph $G = (X \cup Y,\ B(G))$, where

$$X = \{u_1,\ u_2,\ u_3,\ u_4\} \tag{5.5}$$

and

$$Y = \{v_1,\ v_2,\ v_3,\ v_4,\ v_5,\ v_6\} \tag{5.6}$$

It can be verified that $B(G)$ has no duplicate rows or columns. In addition to that, $|v_i| = 2, 1 \leq i \leq 6$ and $|u_j| = 3, 1 \leq j \leq 4$. Hence, $B(G_{r<}) = B(G)$. However,

$$\psi_C = \mathcal{N}(B(G_{r<})) = \begin{pmatrix} 1 & 1 & 1 & 1 \end{pmatrix}^T \tag{5.7}$$

From (5.7), column u_4 of $B(G)$ is linearly dependent in GF_2 on its first three columns. $B(G')$ obtained by applying algorithm 3 is given by

$B(G')$	u_1	u_2	u_3
v_1	0	0	1
v_2	0	1	0
v_3	0	1	1
v_4	1	0	0
v_5	1	0	1
v_6	1	1	0

(5.8)

Reduced ordered matching in G' is determined as

$$M_{(G')} = \{\ (v_4,u_1),\quad (v_2,u_2),\quad (v_1,u_3)\ \}$$

and

$$|M_{(G')}| = 3 \tag{5.9}$$

From lemma 3 and (5.9), $rk_2(B(G))$ can be accurately determined as 3. Similarly, it can be verified that

$$|M_{(H'_j)}| = 2, 1 \leq j \leq 4 \tag{5.10}$$

From (5.10) and theorem 4, the example network $B(G)$ is NCA infeasible. It is shown in section 3.1.2 that example network $B(G)$ is indeed NCA infeasible as a $W, \phi(W) = B(G)$ can be determined such that $rk(W) = 3$.

On applying algorithm 4 to $B(G)$ (3.1) two sub-graphs of G, $H^{(1)}$ and $H^{(2)}$, are obtained. The two sub-graphs are defined as

$$H^{(1)} = (X^{(1)} \cup Y^{(1)},\ B(H^{(1)})) \tag{5.11}$$

and

$$H^{(2)} = (X^{(2)} \cup Y^{(2)},\ B(H^{(2)})) \tag{5.12}$$

The sets $X^{(1)}$ in (5.11) and $X^{(2)}$ in (5.12) are given by

$$\begin{gathered} X^{(1)} = \{u_1,\ u_2,\ u_3\} \\ X^{(2)} = \{u_4\} \end{gathered} \tag{5.13}$$

and sets $Y^{(1)}$ and $Y^{(2)}$ are given by

$$\begin{aligned} Y^{(1)} &= \{v_1,\ v_2,\ v_3,\ v_4,\ v_5,\ v_6\} \\ Y^{(2)} &= \{v_1,\ v_2,\ v_4\} \end{aligned} \tag{5.14}$$

It can be seen from (5.5) and (5.13) that

$$X^{(1)} \cap X^{(2)} = \{\}$$

and

$$X = X^{(1)} \cup X^{(2)}$$

$B(H^{(1)})$ in (5.11) is a sub-matrix of $B(G)$ (3.1) formed by choosing from $B(G)$ rows indexed by $Y^{(1)}$ (5.14) and columns indexed by $X^{(1)}$ (5.13). Similarly, $B(H^{(2)})$ in (5.12) is a sub-matrix of $B(G)$ (3.1) formed by choosing from $B(G)$ rows indexed by $Y^{(2)}$ (5.14) and columns indexed by $X^{(2)}$ (5.13). The two sub-matrices of $B(G)$, $B(H^{(1)})$ and $B(H^{(2)})$, are such that

$$B(G) = (\ B(H^{(1)})\ :\ B(H^{(2)})\ )$$

G	$H^{(1)}$			$H^{(2)}$
	u_1	u_2	u_3	u_4
v_1	0	0	1	1
v_2	0	1	0	1
v_3	0	1	1	0
v_4	1	0	0	1
v_5	1	0	1	0
v_6	1	1	0	0

(5.15)

From (5.8) and (5.15)

$$H^{(1)} \equiv G'$$

and hence, from (5.9) and lemma 3,

$$rk_2(H^{(1)}) = 3 \tag{5.16}$$

In the sense of NCA, the reduced matrices $B(H_1^{(1)})$, $B(H_2^{(1)})$ and $B(H_3^{(1)})$ obtained by respectively removing columns u_1, u_2 and u_3 from $H^{(1)}$ and their neighbour rows

are equivalent to a bipartite graph $\bar{H}$ as

$$B(H_1^{(1)}) \equiv B(H_2^{(1)}) \equiv B(H_3^{(1)}) \equiv B(\bar{H}) = \begin{pmatrix} 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{pmatrix} \tag{5.17}$$

Note that the relationship $\equiv$ in (5.17) implies that the three sub-networks are homomorphic to $\bar{H}$. This distinction is important as distinct set of vertices constitute each sub-network. It can be verified from (5.17) that first two rows and two columns of $B(\bar{H})$ form a reduced ordered matching of size 2. Therefore,

$$|M_{(\bar{H}')}| = 2 \tag{5.18}$$

Thus, from (5.17), (5.18) and theorem 4, sub-network $H^{(1)}$ (5.15) is NCA feasible. On the other hand, sub-network $H^{(2)}$ (5.15) has only one column and $H^{(2)}$ is NCA feasible as

$$rk_2(B(H^{(2)})) = 1$$

Thus, NCA infeasible example network $B(G)$ (3.1) is decomposed into a set of NCA feasible sub-networks $H^{(1)}$ and $H^{(2)}$.

5.4.2 Carbon source switching network decomposition

In this section, results from this chapter are applied to the experimental dataset-network pair $(D, B(G))$ (section 2.5). Unlike in the case of example network in section 5.4.1, the transcriptional regulatory network $B(G)$ (Fig. 2.4) is associated with the experimental dataset D. Therefore, algorithm 5 is applied in this section to the pair $(D, B(G))$ to obtain NCA feasible sub-networks.

It was shown in section 4.4.2 that though the underlying network $B(G)$ is NCA feasible, the dataset-network pair $(D, B(G))$ is NCA infeasible as the number of data points $N = 9$ in D are lesser than the number of source signals $L = 16$ in S. It was shown in (4.20) that unique NCA solutions can be obtained if the number of source signals comprising matrix factor S is restricted to 9. A sub-network $B(G)^{(1)}$ (Fig. 4.2) and the corresponding data-subset $D^{(1)}$ were chosen to illustrate this fact. However, the choice of such a sub-network was not explained there. Such a choice is not random as shown in this section.

Network decomposition – Carbon source switching network $B(G)$ (fig. 2.4) corresponds to a bipartite graph $G = (X \cup Y, B(G))$, where the set of column vertices X

is given as

$$X = \{u_1, \cdots, u_{16}\} \tag{5.19}$$

and the set of row vertices Y is given as

$$Y = \{v_1, \cdots, v_{100}\} \tag{5.20}$$

$B(G)$ has full column rank in GF_2

$$rk_2(B(G)) = 16$$

and hence, it has no zero or duplicate columns. There are no zero rows in $B(G)$. Symmetric difference [16] between sets of neighbours of row vertices v_i and v_j are calculated as

$$(\mathcal{N}(v_i) - \mathcal{N}(v_j)) - (\mathcal{N}(v_j) - \mathcal{N}(v_i)) = \{\}, \;\; 1 \leq i, j \leq M$$

Not that the operation '$-$' represents set difference. Rows v_i and v_j are duplicates if the corresponding symmetric difference is an empty set. Using this method, 61 out of 100 rows of $B(G)$ are identified as duplicates of the other 39 rows $v_i, i \in \{1, 4, 5, 6, 8, 10, 13, 14, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 30, 31, 33, 34, 35, 39, 40, 41, 43, 47, 49, 51, 55, 57, 65, 68, 72, 77, 80, 87\}$. Therefore, reduced version of $B(G)$ $B(G_r)$ has 39 rows and 16 columns. Reduced and ordered version of $B(G)$ $B(G_{r<})$ also has 39 rows and 16 columns. As there are no linearly dependent columns in $B(G_{r<})$,

$$B(G') = B(G_{r<})$$

A maximal reduced ordered matching in G' is given by

$$M_{(G')} = \{(X', Y')\} \tag{5.21}$$

where, the set Y' is given as

$$Y' = \{v_i\}, \; i = 4, \; 16, \; 11, \; 14, \; 5, \; 1, \; 15, \; 13, \; 2, \; 6, \; 9, \; 8, \; 3, \; 12, \; 7, \; 10$$

and the set X' is given as

$$X' = \{u_j\}, \; 1 \leq j \leq 16$$

From (5.21),

$$|M_{(G')}| = 16 \tag{5.22}$$

From (5.22) and lemma 3,

$$rk_2(B(G)) = 16 \tag{5.23}$$

It can be verified that $B(G)$ satisfies the conditions of theorem 4 and hence, it is NCA feasible. However, as pointed out earlier, D has 9 data points which makes the dataset-network pair $(D, B(G))$ NCA infeasible according to theorem 3, section 4.3. From (4.20), section 4.4.2 and (5.4),

$$n_{max} = 9 \tag{5.24}$$

On applying algorithm 5 to $B(G)$ (Fig. 2.4) with n_{max} as given in (5.24), two sub-graphs of G, $H^{(1)}$ and $H^{(2)}$, are obtained. The two sub-graphs are defined as

$$H^{(1)} = (X^{(1)} \cup Y^{(1)},\ B(H^{(1)})) \tag{5.25}$$

and

$$H^{(2)} = (X^{(2)} \cup Y^{(2)},\ B(H^{(2)})) \tag{5.26}$$

The sets $X^{(1)}$ and $Y^{(1)}$ in (5.25) are given by

$$\begin{gathered} X^{(1)} = \{u_1, \cdots, u_9\},\ Y^{(1)} = \{v_i\} \\ i = 1,\ 2,\ 3,\ 4,\ 5,\ 6,\ 7,\ 8,\ 9,\ 10,\ 11,\ 12,\ 13,\ 15,\ 16,\ 17,\ 18,\ 19, \\ 22,\ 23,\ 25,\ 26,\ 27,\ 28,\ 29,\ 31,\ 32,\ 34,\ 36,\ 38,\ 40,\ 43,\ 47,\ 48,\ 50, \\ 51,\ 52,\ 53,\ 57,\ 58,\ 60,\ 62,\ 65,\ 66,\ 67,\ 72,\ 73,\ 74,\ 75,\ 76,\ 80,\ 82,\ 83, \\ 84,\ 87,\ 100 \end{gathered} \tag{5.27}$$

and sets $X^{(2)}$ and $Y^{(2)}$ in (5.26) are given by

$$\begin{gathered} X^{(2)} = \{u_{10}, \cdots, u_{16}\},\ Y^{(2)} = \{v_i\} \\ i = 4,\ 10,\ 14,\ 20,\ 21,\ 22,\ 24,\ 25,\ 27,\ 30,\ 31,\ 32,\ 33,\ 35,\ 36,\ 37,\ 38, \\ 39,\ 40,\ 41,\ 42,\ 44,\ 45,\ 46,\ 49,\ 50,\ 51,\ 54,\ 55,\ 56,\ 57,\ 59,\ 61,\ 63,\ 64, \\ 68,\ 69,\ 70,\ 71,\ 76,\ 77,\ 78,\ 79,\ 80,\ 81,\ 82,\ 83,\ 84,\ 85,\ 86,\ 87,\ 88,\ 89, \\ 90,\ 91,\ 92,\ 93,\ 94,\ 95,\ 96,\ 97,\ 98,\ 99 \end{gathered} \tag{5.28}$$

It can be seen from (5.19), (5.27) and (5.28) that

$$X^{(1)} \cap X^{(2)} = \{\}$$

and

$$X = X^{(1)} \cup X^{(2)}$$

Let the sub-matrix $B(H^{(1)})$ of $B(G)$ be formed by choosing rows and columns of $B(G)$ indexed respectively by $X^{(1)}$ and $Y^{(1)}$ (5.27). On comparing the subscripts of u and values of i in (5.27) respectively with the indices on horizontal and vertical axes in Fig. 4.2, it can be seen that Fig. 4.2 is a colour map representing $H^{(1)}$ described in (5.25).

NCA feasibility of sub-networks – It is demonstrated in section 4.4.2 that $B(H^{(1)})$ (Fig. 4.2) is NCA feasible. It can be verified that $H^{(1)}$ satisfies the conditions in theorem 4 and hence, is NCA feasible. NCA solutions corresponding to $H^{(1)}$ are unique up to a diagonal matrix as the number of source signals is $L^{(1)} = 9$ and

$$rk([B(H^{(1)}) : D^{(1)}]) = 18 = 2L^{(1)}$$

On the other hand, consider the sub-network $B(H^{(2)})$ defined by the underlying NCA feasible sub-graph $H^{(2)}$ (5.26). $B(H^{(2)})$ consists of last 7 columns of $B(G)$ and corresponding non zero rows as depicted in Fig. 5.1. It can be verified that $B(H^{(2)})$

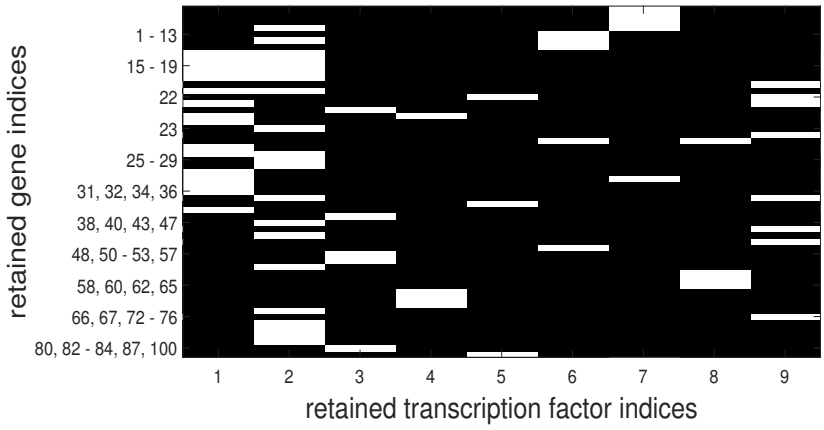

Figure 5.1: Colour map of $B(H^{(2)})$ (0 – black, 1 – white): NCA feasible sub-network of transcriptional regulatory network in [1]

is NCA feasible as it satisfies conditions of both theorems 2 and 4. However, the pair $(D^{(2)}, B(H^{(2)}))$ is NCA infeasible as it partially satisfies the conditions of theorem 3. The data-subset $D^{(2)}$ does not satisfy second condition of theorem 3 as the number of source signals in $H^{(2)}$ is $L^{(2)} = 7$ whereas the number of source signal that are required to ensure NCA feasibility is given as

$$rk([B(H^{(2)}) \ : \ D^{(2)}]) = 16 > 2L^{(2)} \tag{5.29}$$

As pointed out at the end of section 5.3, algorithm 4 ensures that the matrix factors $W^{(2)}$ and $S^{(2)}$ are full-ranked. However, it does not guarantee that the corresponding NCA solutions are unique. It is shown later in this section that though $H^{(2)}$ is full-rank factorisable, corresponding NCA solutions are not unique up to a scaling factor as described in theorem 1.

As a direct result of application of algorithm 5, matrix factors $W^{(2)}$ corresponding to $H^{(2)}$ is full-ranked. Here, it is shown that $S^{(2)}$ corresponding to $H^{(2)}$ is also full-ranked. The bounds on rank of source signal matrix $S^{(2)}$ can be computed using (4.16) as

$$rk([B(H^{(2)}) : D^{(2)}]) - rk(B(H^{(2)})) \leq rk(S^{(2)}) \leq \min(L^{(2)}, N)$$

where, $L^{(2)} = 7$. The lower limit on $rk(S^{(2)})$ evaluates to

$$rk([B(H^{(2)}) : D^{(2)}]) - rk(B(H^{(2)})) = 16 - 7 = 9$$

This is meaningless in this case as there are only 7 columns in $B(H^{(2)})$. However, an appropriate upper limit on $rk(S^{(2)})$ can be calculated as

$$\min(L^{(2)}, N) = \min(7, 9) = 7$$

As a result,

$$rk(S^{(2)}) \leq 7 \tag{5.30}$$

It remains to see if $S^{(2)}$ obtained by applying NCA to the pair $(D^{(2)}, B(H^{(2)}))$ is full-ranked. An estimate of $S^{(2)}$ obtained by applying NCA to the pair $(D^{(2)}, B(H^{(2)}))$ is given as

$$S^{(2)} = \begin{pmatrix} -0.02 & 0.02 & -0.03 & -0.12 & -0.28 & -0.33 & -0.32 & -0.33 & -0.30 \\ 0.3 & 0.33 & 0.35 & 0.25 & 0.10 & -0.01 & -0.02 & -0.05 & -0.07 \\ 0.37 & 0.38 & 0.41 & 0.36 & 0.18 & 0.05 & 0.04 & 0.07 & 0.08 \\ -0.27 & -0.41 & -0.35 & -0.34 & -0.12 & -0.12 & -0.13 & -0.11 & -0.12 \\ -0.53 & -0.59 & -0.60 & -0.64 & -0.50 & -0.46 & -0.43 & -0.43 & -0.41 \\ -0.09 & -0.22 & -0.24 & -0.22 & -0.19 & -0.21 & -0.17 & -0.19 & -0.15 \\ 0.42 & 0.43 & 0.40 & 0.41 & 0.29 & 0.30 & 0.25 & 0.33 & 0.26 \end{pmatrix} \tag{5.31}$$

It can be verified from (5.31) that

$$rk(S^{(2)}) = 7$$

Uniqueness of NCA solutions – It is shown in section 4.4.2 that the solutions obtained by applying NCA to data-subset corresponding to $B(H^{(1)})$ (Fig. 4.2) are unique up to a diagonal matrix $\chi^{(1)}$ (4.21). These results are applicable to the sub-network $H^{(1)}$ (5.25). A similar analysis is conducted here in order to test the uniqueness of NCA solutions corresponding to $H^{(2)}$, two pairs of NCA solutions $(({}^1W^{(2)}),({}^1S^{(2)}))$ and $(({}^2W^{(2)}),({}^2S^{(2)}))$ are estimated.

Weight matrices $({}^1W^{(2)})$ and $({}^2W^{(2)})$ are as illustrated respectively in Fig. 5.2 and 5.3. It can be seen in Fig. 5.2 and Fig. 5.3 that the maximum and minimum values on scale to the right of the colour map are the same for both weight matrices $-> 2$ and < -2. However, columns 1, 3, and 5 in 5.2 and 5.3 are visibly different. On the other hand, source signal matrices $({}^1S^{(2)})$ and $({}^2S^{(2)})$ depicted in Fig. 5.4 and 5.5 have different minimum values as seen on scales to the right of respective colour maps. In addition to that, the two colour maps in Fig. 5.4 and 5.5 are visibly different. These observations indicate that the two solutions $(({}^1W^{(2)}),({}^1S^{(2)}))$ and $(({}^2W^{(2)}),({}^2S^{(2)}))$ are not identical.

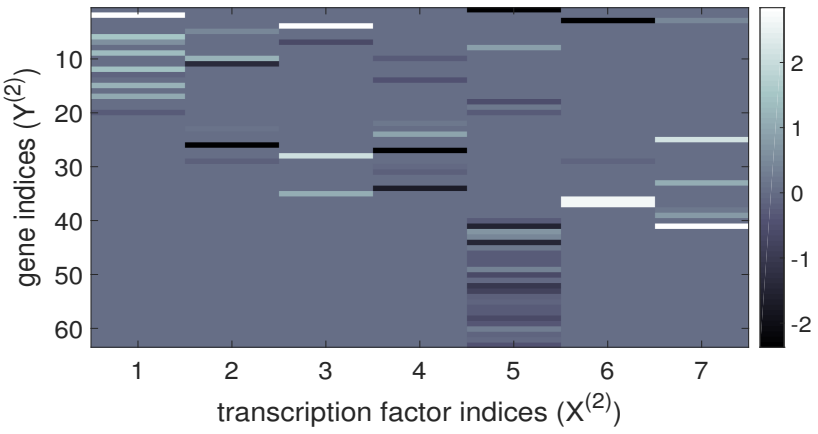

Figure 5.2: Colour map of $({}^1W^{(2)})$: NCA infeasible sub-network $H^{(2)}$ of transcriptional regulatory network in [1]

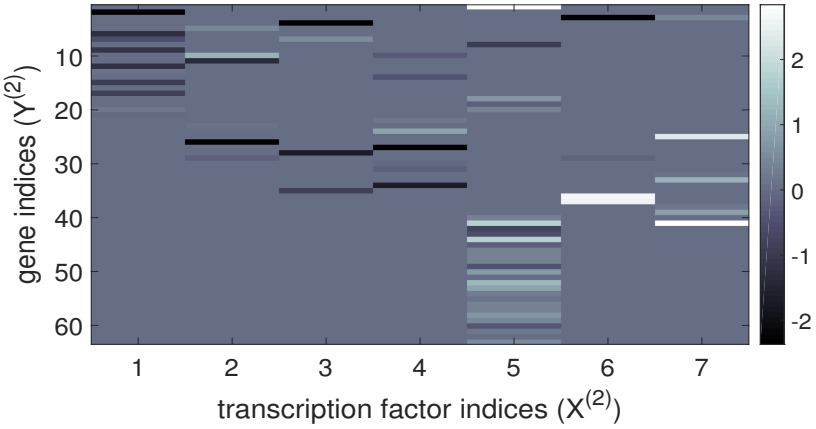

Figure 5.3: Colour map of $({}^2W^{(2)})$: NCA infeasible sub-network $H^{(2)}$ of transcriptional regulatory network in [1]

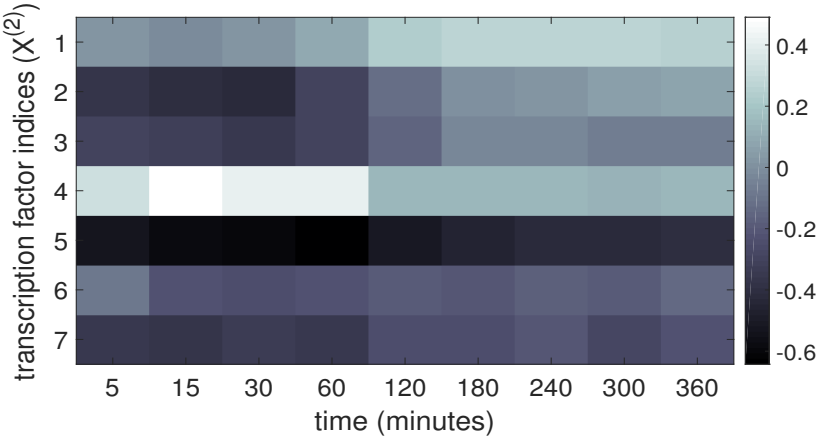

Figure 5.4: Colour map of $({}^1S^{(2)})$: NCA infeasible sub-network $H^{(2)}$ of transcriptional regulatory network in [1]

Using the method presented in section 3.3, chapter3, $\chi^{(2)}$ is determined as

$$\chi^{(2)} = \begin{pmatrix} -0.84 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.84 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1.19 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & -0.01 & 0 & 1.02 \end{pmatrix} \tag{5.32}$$

It can be seen in $\chi^{(2)}$ (5.32) that one of the off-diagonal entries in the last row is

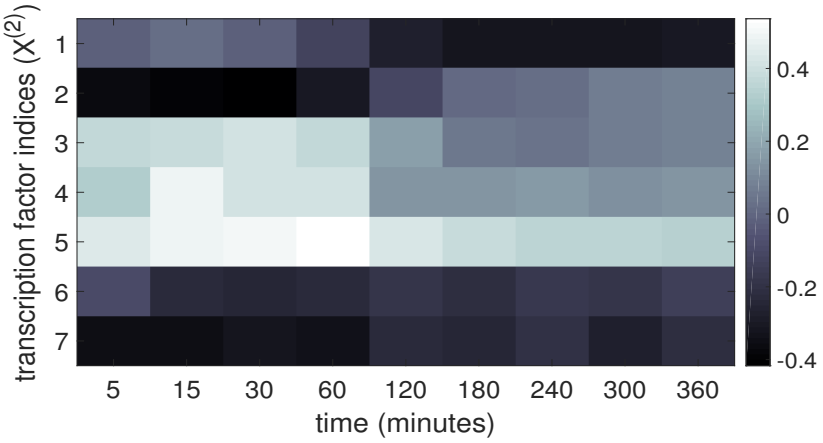

Figure 5.5: Colour map of $({}^2S^{(2)})$: NCA infeasible sub-network $H^{(2)}$ of transcriptional regulatory network in [1]

non-zero. This is because, as pointed out in (5.29), the number of source signals is lesser than what is necessary to ensure NCA feasibility. However, all hopes are not lost as the error between $({}^1S^{(2)})$ and $\chi^{(2)}({}^2S^{(2)})$ is negligible

$$\|({}^1S^{(2)}) - \chi^{(2)}({}^2S^{(2)})\|_F = 6 \times 10^{-4} \tag{5.33}$$

This implies that the two NCA solutions $(({}^1W^{(2)}), ({}^1S^{(2)}))$ and $(({}^2W^{(2)}), ({}^2S^{(2)}))$ are almost unique up to a scaling factor $\chi^{(2)}$.

Theoretically, an optimal number of data points that can ensure NCA feasibility of all sub-networks can be sought if it exists. This number can be used in algorithm 5 to limit the size of sub-networks. This problem is considered as a part of future work and is not pursued in this thesis.

5.5 Summary

In this chapter, a matching based method to calculate binary rank of a graph is developed. For the first time in NCA literature, accurate graph theoretical conditions for NCA feasibility of a given network are developed in this chapter. Building on these conditions, two algorithms are developed – one decomposes a given network into a set of NCA feasible sub-networks when a dataset is not available whereas the other identifies NCA feasible sub-networks that can be used to compute full-rank factors for a given dataset. Applicability of these algorithms is demonstrated with the help of a numerical example and an experimental dataset. Algorithms developed in this chapter along with NCA are shown to generate full-ranked factors of a subset

of an experimental dataset. However, uniqueness of obtained solutions cannot be guaranteed. It is pointed out that an optimal size of sub-networks can be computed to ensure NCA feasibility of sub-networks and hence, uniqueness of NCA solutions. Such a problem is not addressed in this thesis. Results from this chapter are used in the next chapter to develop a NCA based divide and conquer approach to solve a general structure-constrained matrix factorisation problem.

Chapter 6

Divide and conquer NCA – unique matrix factors

NCA feasibility is accurately defined in chapter 4. Method to decompose NCA infeasible networks into full-rank factorisable sub-networks is developed in chapter 5. Goal in this chapter is to develop a method to solve a general SCMF problem in (2.12) where the underlying dataset-network pair $(D, B(G))$ is not assumed to be NCA feasible. This objective is achieved by combining the results developed in the previous chapters.

A brief discussion on iterative sub-network network component analysis (IS-NCA) [74] is presented in section 2.3.2. This chapter borrows the idea of partitioning the system of linear equations in (1.3) in a way similar to the one described in (2.16).

The idea that is pursued in this chapter is – given a dataset-network pair $(D, B(G))$ does not satisfy one or more conditions of theorem 3 (section 4.3), algorithm 5 can be used to identify full-rank factorisable sub-graphs as given in (5.3). NCA can then be used to independently estimate $W^{(i)}$s and $S^{(i)}$s corresponding to data-subset-sub-network pairs $(D^{(i)}, B(H^{(i)})$s such that

$$D^{(i)} = W^{(i)} S^{(i)},\ 1 \leq i \leq g \tag{6.1}$$

A convex combination of the NCA solutions of all sub-networks given as

$$D = \begin{pmatrix} \lambda^{(1)}W^{(1)} & \cdots & \lambda^{(i)}W(i) & \cdots & \lambda^{(g)}W^{(g)} \end{pmatrix} \begin{pmatrix} S^{(1)} \\ \vdots \\ S^{(i)} \\ \vdots \\ S^{(j)} \end{pmatrix} \tag{6.2}$$

solves the SCMF problem in (2.12). The parameters $\lambda^{(i)} > 0, 1 \leq i \leq g$ in (6.2) are such that

$$\sum_{i=1}^{g} \lambda^{(i)} = 1 \tag{6.3}$$

The parameters $\lambda^{(i)}$s quantify percentages of D factorised by the respective sub-networks $H^{(i)}$s. The divide and conquer solution proposed in (6.2) can be summarised in words as

1. identify full-rank factorisable data-subset-sub-network pairs $(D^{(i)}, B(H^{(i)}))$s
2. use NCA to estimate independently matrix factors $W^{(i)}$ and $S^{(i)}$ corresponding to pairs $(D^{(i)}, B(H^{(i)}))$s
3. combine estimated sub-network weight matrices $W^{(i)}$s in a convex fashion so as to minimise the error in reconstructing D

In the context of this thesis, the divide and conquer approach proposed in (6.2) is referred to as 3DNCA where 3 indicates three steps involved and D stands for divide and conquer. Uniqueness of solutions obtained by such a divide and conquer method is guaranteed up to two scaling factors, both of which are diagonal matrices. It is shown in the previous chapter that algorithm 5 guarantees full-rank factorisability, not NCA feasibility. Therefore, uniqueness results developed in this chapter assume that all data subset-sub-network pairs are NCA feasible. Theoretical results with respect to the uniqueness of solutions developed in this chapter are in coherence with that for NCA.

A formal definition of 3DNCA problem is presented in section 6.1. Need for adding more parameters to achieve better solutions is pointed out in section 6.1. Scalability issues related to adding more parameters in terms of book-keeping and size of parameter space are brought forward in section 6.2. An iterative 3DNCA algorithm is developed in section 6.3 that achieves the objective of adding more parameters while avoiding the issues of scalability. A theorem is presented in section 6.4 that guarantees uniqueness of 3DNCA solutions assuming all underlying data

subset-sub-network pairs are NCA feasible. Strengths and limitations of 3DNCA are demonstrated in section 6.5 with the help of a random numerical example. Applicability of 3DNCA is demonstrated with the help a practical dataset introduced in section 2.5.

6.1 Divide and conquer NCA problem

In this section, a framework is developed to set up a structure constrained matrix factorisation problem that formalises the idea presented in (6.2). There are 3 steps involved in obtaining a solution of the form (6.2)

1. identifying all sub-networks $H^{(i)}$s
2. NCA factorisation of individual datasets $D^{(i)}$s
3. estimating mixing coefficients $\lambda^{(i)}$s

Hence the name 3-part divdie and conquer NCA (3DNCA) In this section, the objective is to identify and address the difficulties in setting up a divide and conquer problem, the solution to which is of the form given in (6.2).

Initially, it is assumed that the underlying dataset-network pair $(D, B(G))$ is NCA infeasible. It is shown later in this section that a feasible NCA problem is a special case of 3DNCA. Assume that algorithm 5 is used to identify sub-networks $H^{(i)}$s of a given NCA infeasible network G. Data-subsets $D^{(i)}$s can be easily identified by collecting all rows of D corresponding to row subsets $Y^{(i)}$s in $H^{(i)}$s. Algorithm 5 divides the network such that the column subsets $X^{(i)}$s are disjoint. As a result, dividing source signals matrix S as shown (6.2) is straightforward. However, the row subsets $Y^{(i)}$s are not necessarily disjoint. Therefore, $D^{(i)}$s might have rows that are to be factorised by multiple sub-networks. As a result, dividing D and W is not as easy as in the case of S in (6.2). In order to overcome this difficulty, the dataset D is divided into non-overlapping part D_U and overlapping part D_C as

$$D = \left(\frac{D_U}{D_C} \right) \tag{6.4}$$

where, D_U represents data subsets that are unique to the identified sub-networks $H^{(i)}$s and D_C represents data subsets common to multiple sub-networks. The rows in $D_U^{(i)}$ are such that the corresponding rows in all other sub-networks $B(H^{(j)}), 1 \leq j \leq, g, j \neq i$ are zero rows. On the other hand, rows in $D_C^{(i)}$ are such that there is

at least one sub-network $H^{(j)}, 1 \leq j \leq g, j \neq i$ with corresponding rows in $B(H^{(j)})$ non-zero. The weights matrix W will have a structure similar to that of D

$$W = \begin{pmatrix} W_U \\ \hline W_C \end{pmatrix} \tag{6.5}$$

where, W_U and W_C correspond respectively to unique and common parts of D. For every sub-network $H^{(i)}, 1 \leq i \leq g$, the weights sub-matrix $W^{(i)}$ estimated using NCA can be rewritten as

$$W^{(i)} = \begin{pmatrix} W_U^{(i)} \\ \hline W_C^{(i)} \end{pmatrix} \tag{6.6}$$

Structural details of W_U and W_C are explored next. From (6.2)–(6.5), the overlapping part of D is given by

$$D_C = W_C S \tag{6.7}$$

Each row in D_C can be shared by any number of sub-networks between $n = 2$ and g. As a result, the maximum number of possible combinations is

$$\sum_{n=2}^{g} {}^{g}C_n \tag{6.8}$$

where, ${}^{g}C_n$ represents number of combinations possible while choosing n out of g identified sub-networks. Thus, D_C and W_C cannot be characterised in a simple way by separating rows or columns as in the case of S in (6.2). One of the ways to address this issue is by defining D_C as a convex combination of $W_C^{(i)}$s, where $W_C^{(i)}$ comprises of rows of sub-matrix W_C corresponding to the sub-network $H^{(i)}$. In other words, $W_C^{(i)}$ has those rows of $W^{(i)}$ for which the corresponding rows in some $W^{(j)}, j \neq i$ is non-zero. The equation in (6.7) can be rewritten as

$$D_C = W_C S = \begin{pmatrix} \lambda^{(1)} W_C^{(1)} & \cdots & \lambda^{(i)} W_C^{(i)} & \cdots & \lambda^{(g)} W_C^{(g)} \end{pmatrix} \begin{pmatrix} S^{(1)} \\ \vdots \\ S^{(i)} \\ \vdots \\ S^{(g)} \end{pmatrix} \tag{6.9}$$

where, $\lambda^{(i)}$s are as given in (6.3). Unlike in the case of D_C, there is only one way in which D_U can be divided – g unique data subsets corresponding to g individual

sub-networks. As a consequence, D_U in (6.11) is defined as

$$D_U = \begin{pmatrix} D_U^{(1)} \\ \vdots \\ D^{(i)} \\ \vdots \\ D_U^{(g)} \end{pmatrix} \tag{6.10}$$

where, $D_U^{(i)}, 1 \leq i \leq g$ correspond to those rows in D that are unique to individual sub-networks $H^{(i)}$s. From (6.2)–(6.5), non-overlapping part of D is given as

$$D_U = W_U S \tag{6.11}$$

In order to maintain consistency in notations between (6.9) and (6.11), the relationship in (6.11) must be of the form

$$D_U = \begin{pmatrix} \lambda^{(1)} W_U^{(1)} & \cdots & \lambda^{(i)} W_U^{(i)} & \cdots & \lambda^{(g)} W_U^{(g)} \end{pmatrix} S \tag{6.12}$$

such that the whole system of equations resemble that in (6.2). However, for every $i, 1 \leq i \leq g$,

$$D_U^{(i)} = W_U^{(i)} S^{(i)}$$

must be satisfied indicating that 100% of $D_U^{(i)}$ is factorised by sub-network $H^{(i)}$. As from (6.3) $\lambda^{(i)} > 0, \forall i$, W_U can be defined as

$$W_U = \begin{pmatrix} \frac{1}{\lambda^{(1)}} W_U^{(1)} & 0 & \cdots & 0 \\ 0 & \frac{1}{\lambda^{(2)}} W_U^{(2)} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{1}{\lambda^{(g)}} W_U^{(g)} \end{pmatrix} \tag{6.13}$$

such that

$$D_U^{(i)} = \cancel{\lambda^{(i)}} \frac{1}{\cancel{\lambda^{(i)}}} W_U^{(i)} S^{(i)}$$

and hence, (6.10) is satisfied. From (6.2) – (6.13), the whole system of equations can be represented as

$$D = WS$$

$$\left(\frac{D_U}{D_C}\right) = \left(\begin{array}{cccc} W_U^{(1)} & 0 & \cdots & 0 \\ 0 & W_U^{(2)} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & W_U^{(g)} \\ \hline \lambda^{(1)}W_C^{(1)} & \lambda^{(2)}W_C^{(2)} & \cdots & \lambda^{(g)}W_C^{(g)} \end{array}\right) \left(\begin{array}{c} S^{(1)} \\ S^{(2)} \\ \vdots \\ S^{(g)} \end{array}\right) \tag{6.14}$$

It can be verified that (6.2) and (6.14) are equivalent. Note that, from (6.6) and (6.14),

$$W \neq \left(\begin{array}{cccccc} W^{(1)} & \cdots & W^{(i)} & \cdots & W^{(g)} \end{array} \right)$$

as $W_U^{(i)}$s sit in a block diagonal form in W whereas $W_C^{(i)}$s are multiplied by $\lambda^{(i)}$s. Also,

$$W \neq \left(\begin{array}{cccccc} \lambda^{(1)}W^{(1)} & \cdots & \lambda^{(i)}W^{(i)} & \cdots & \lambda^{(g)}W^{(g)} \end{array} \right)$$

The system representation in (6.14) implies that NCA solutions of the individual sub-networks are sufficient to minimise error in reconstructing D_U as $W_U^{(i)}$s are not affected by the *mixing coefficients* $\lambda^{(i)}$s. Therefore, the problem statement involving $\lambda^{(i)}$s must consider only D_C. Accordingly, a *mixing problem* is formulated as

$$\begin{array}{ll} \min\limits_{\lambda^{(i)}} & \|D_C - \sum_{i=1}^{g} \lambda^{(i)} W_C^{(i)} S^{(i)}\|_F^2 \\ \text{subject to} & \lambda^{(i)} > 0, 1 \leq i \leq g \\ & \sum_{i=1}^{g} \lambda^{(i)} = 1 \end{array} \tag{6.15}$$

Mixing problem in (6.15) formally completes description of three-part problem verbally described at the beginning of this chapter. This presents an initial description of 3DNCA problem. A few hurdles in implementing a solution to this problem are identified in the following sections. Description of 3DNCA problem is modified wherever such issues are addressed.

Step 2 of 3DNCA involves independent application of NCA to the identified sub-networks. An underlying assumption is that the estimation of matrix factors corresponding to one sub-network does not affect or depend on that of the others. However, when the overlapping or shared part of data subsets $D_C^{(i)}, 1 \leq i \leq g$ are non-empty, NCA factorisation of the sub-networks are dependent on that of the others. Mixing coefficients $\lambda^{(i)}$s are used in the third step of 3DNCA to capture the effect of these dependencies. This allows for parallel/independent NCA factorisation of all identified data subsets. Note that classical NCA is a special case of 3DNCA. If a dataset-network pair $(D, B(G))$ is NCA feasible, then there is only one sub-network in step 1, $H^{(1)} = G$. In this case, a solution to the divide and conquer problem is the same as that to the classical NCA problem with $W^{(1)} = W$, $S^{(1)} = S$ and $\lambda^{(1)} = 1$.

As pointed out in (6.8), in a general case with multiple sub-networks, there can be a large number of possible ways in which data D_C is shared among the sub-networks. Number of non-zero entries increases in rows of W_C with an increase in the number of sub-networks sharing D_C. As pointed out in section 2.2.1, denser sub-matrices have a more significant impact on parameter estimation. As a consequence, the values of $\lambda^{(i)}$s in (6.15) depend heavily on the denser parts of W_C. This might result in larger errors in reconstructing other rows of D_C shared by relatively smaller number of sub-networks. This problem can be solved by setting up separate mixing problems of the form (6.15) for each of the $\sum_{n=2}^{g} {}^{g}C_n$ (6.8) possible combination of sub-networks. This ensures smaller errors in reconstructing all rows of D_C. However, it is shown in the next section that setting up multiple mixing problems is challenging. The notations provided in this section are limited to a single mixing problem based 3DNCA. In the following section, notations used in this section are modified to make way for multiple mixing problems. Changes in notations are explained wherever they are introduced.

6.2 Scalability of multiple mixing problems

Setting up multiple mixing problems is recommended in the previous section to improve the quality of data reconstruction. However, extending the problem in (6.15) to formalise the idea of multiple mixing problems gives rise to scalability issues. In this section, the challenges in setting up multiple mixing problems are identified. An iterative approach is proposed in the next section to address these challenges.

Scalability of mixing problems with respect to D_C is challenging as there is a large number (6.8) of ways in which data can be shared among sub-networks. This requires introduction of addition parameters as shown in this section. To get a flavour of this scalability issue, a two-part network is considered followed by a three-part network. It is shown that the number of additional parameters required grow significantly with an increase in the number of sub-networks added.

6.2.1 Two-part mixing problem

Consider a simple case with two sub-networks $H^{(1)}$ and $H^{(2)}$. Assume that for some dataset-network pair $(D, B(G))$ algorithm 5 results in a two-part decomposition of the whole network $B(G)$ given as

$$G = H^{(1)} \cup H^{(2)} \tag{6.16}$$

A solution of the form (6.2) to a related 3DNCA problem is of the form

$$D = \begin{pmatrix} \lambda^{(1)}W^{(1)} & \lambda^{(2)}W^{(2)} \end{pmatrix} \begin{pmatrix} S^{(1)} \\ S^{(2)} \end{pmatrix} \tag{6.17}$$

As mentioned in the previous section, data subsets $D^{(1)}$ and $D^{(2)}$ are identified by collecting all rows of D corresponding to row subsets $Y^{(1)}$ and $Y^{(2)}$. As there are only two sub-networks, from (6.8), there is only one possible way in which the two sub-networks can share data – all rows in D_C are common to both sub-networks. Let $D_C^{(1,2)} \subseteq D^{(1)}$ represent the rows in $D^{(1)}$ that are shared by $H^{(2)}$, which implies the corresponding rows in $B(H^{(1)})$ and $B(H^{(2)})$ are non-zero. Similarly, let $D_C^{(2,1)} \subseteq D^{(2)}$ represent the rows in $D^{(2)}$ that are shared by $H^{(1)}$. The common parts of data subsets $D^{(1)}$ and $D^{(2)}$ are denoted in (6.4) respectively by $D_C^{(1)}$ and $D_C^{(2)}$. At this point, a change in notation from $D_C^{(1)}$ and $D_C^{(2)}$ to $D_C^{(1,2)}$ an $D_C^{(2,1)}$ seems unnecessary. The reason is that there is no other way in which data can be shared among the two sub-networks. Hence, $D_C^{(1,2)}$ and $D_C^{(2,1)}$ are identical

$$D_C^{(1,2)} = D_C^{(2,1)} \tag{6.18}$$

Though a change in notation does not seem warranted when a two-part network (6.16) is considered, it is necessary when a third sub-network $H^{(3)}$ is to be added. When a third sub-network is added, from (6.8) there are

$$\sum_{i=2}^{3} {}^3C_i = 4$$

possible ways in which common data can be shared among the three sub-networks. Shared data subsets can be identified as

1. $D_C^{(1,2)}$, data common to $H^{(1)}$ and $H^{(2)}$ only

2. $D_C^{(1,3)}$, data common to $H^{(1)}$ and $H^{(3)}$ only

3. $D_C^{(2,3)}$, data common to $H^{(2)}$ and $H^{(3)}$ only

4. $D_C^{(1,2,3)}$, data common to all 3 sub-networks

Though the following relationships hold

$$\begin{gathered} D_C^{(1,2)} = D_C^{(2,1)} \\ D_C^{(1,3)} = D_C^{(3,1)} \\ D_C^{(2,3)} = D_C^{(3,2)} \\ D_C^{(1,3,2)} = D_C^{(2,1,3)} = D_C^{(2,3,1)} = D_C^{(3,1,2)} = D_C^{(3,2,1)} \end{gathered} \tag{6.19}$$

Each of the four data subsets $D_C^{(1,2)}$, $D_C^{(1,3)}$, $D_C^{(2,3)}$, and $D_C^{(1,2,3)}$ are distinct. As mentioned in the previous section, rows of W_C corresponding to $D_C^{(1,2,3)}$ are denser and hence, might increase errors in reconstructing other three data subsets. It is necessary to distinguish between the four data subsets in (6.19). Therefore, notation different to that used in (6.4) is used. Similar notations are used for W_C and λ in this section. A two-part mixing problem is formulated as discussed next.

In the case of a network with two-part decomposition (6.16), $D_C^{(1,2)}$ must be factorised by both the sub-networks. $D_C^{(1,2)}$ is not always an empty set and hence, dataset D cannot necessarily be clearly divided into two disjoint parts. Instead, the dataset D is divided into unique and common parts as described in (6.4) and (6.10) as

$$D = \begin{pmatrix} D_U^{(1)} \\ D_U^{(2)} \\ \hline D_C^{(1,2)} \end{pmatrix} \tag{6.20}$$

where, $D_U^{(1)}$ and $D_U^{(2)}$ are data subsets corresponding to rows in D unique to $H^{(1)}$ and $H^{(2)}$, respectively. The two data subsets corresponding to $H^{(1)}$ and $H^{(2)}$ are given by

$$D^{(1)} = \begin{pmatrix} D_U^{(1)} \\ D_C^{(1,2)} \end{pmatrix}, \; D^{(2)} = \begin{pmatrix} D_U^{(2)} \\ D_C^{(1,2)} \end{pmatrix} \tag{6.21}$$

The source signal matrix factor S is divided into two disjoint parts as described in (6.2) as

$$S = \begin{pmatrix} S^{(1)} \\ S^{(2)} \end{pmatrix} \tag{6.22}$$

The network parameter matrix factor W is divided into two parts as described in (6.6)

$$W^{(1)} = \begin{pmatrix} W_U^{(1)} \\ W_C^{(1,2)} \end{pmatrix}, \; W^{(2)} = \begin{pmatrix} W_U^{(2)} \\ W_C^{(2,1)} \end{pmatrix} \tag{6.23}$$

where, $W_U^{(1)}$ and $W_U^{(2)}$ respectively correspond to $D_U^{(1)}$ and $D_U^{(2)}$ in (6.20) such that

$$D^{(i)} = W_U^{(i)} S^{(i)},\ i = 1,\ 2 \tag{6.24}$$

From (6.14), (6.20), (6.22), and (6.23), the whole network can be rewritten as

$$D = WS$$

$$\left(\begin{array}{c} D_U^{(1)} \\ D_U^{(2)} \\ \hline D_C^{(1,2)} \end{array}\right) = \left(\begin{array}{c|c} W_U^{(1)} & 0 \\ 0 & W_U^{(2)} \\ \hline \lambda^{(1,2)} W_C^{(1,2)} & \lambda^{(2,1)} W_C^{(2,1)} \end{array}\right) \left(\begin{array}{c} S^{(1)} \\ S^{(2)} \end{array}\right) \tag{6.25}$$

Note that the decomposition in (6.25) is similar to the one in (2.16). However, the decomposition in (6.25) is achieved with the help of algorithm 5 whereas the one in (2.16) is not well-defined in [74]. As established in (6.3), parameters $\lambda^{(1,2)}$ and $\lambda^{(2,1)}$ in (6.23) are expected to satisfy the following relationships

$$\lambda^{(1,2)}, \lambda^{(2,1)} > 0,\ \lambda^{(1,2)} + \lambda^{(2,1)} = 1 \tag{6.26}$$

The parameters $\lambda^{(1,2)}$ and $\lambda^{(2,1)}$ quantify percentages of $D_C^{(1,2)}$ factorised respectively by the two sub-networks $H^{(1)}$ and $H^{(2)}$. $W_C^{(1,2)}$ and $W_C^{(2,1)}$ in (6.23) are such that

$$D^{(1,2)} = \lambda^{(1,2)} W_C^{(1,2)} S^{(1)} + \lambda^{(2,1)} W_C^{(2,1)} S^{(2)} \tag{6.27}$$

In order to compute a complete solution (6.25), the two solutions in (6.24) are combined by solving for $\lambda^{(1,2)}$ and $\lambda^{(2,1)}$ in

$$\begin{array}{ll} \min\limits_{\lambda^{(1,2)}, \lambda^{(2,1)}} & \| D_C^{(1,2)} - \lambda^{(1,2)} W_C^{(1,2)} S^{(1)} - \lambda^{(2,1)} W_C^{(2,1)} S^{(2)} \|_F^2 \\ \text{subject to} & \lambda^{(1,2)}, \lambda^{(2,1)} > 0 \\ & \lambda^{(1,2)} + \lambda^{(2,1)} = 1 \end{array} \tag{6.28}$$

The problem presented in (6.28) is referred to as the two-part mixing problem in the context of this chapter. Though, from (6.18), $D_C^{(1,2)} = D_C^{(2,1)}$, in most of the cases

$$W_C^{(1,2)} \neq W_C^{(2,1)},\ \lambda^{(1,2)} \neq \lambda^{(2,1)}$$

This is because of the fact that $H^{(1)}$ and $H^{(2)}$ are not necessarily identical. The two sub-networks factorise $D_C^{(1,2)}$ in different ways as the sparsity patterns of $B(H^{(1)})$

and $B(H^{(2)})$ are distinct. Only in those cases where $B(H^{(1)}) \equiv B(H^{(2)})$,

$$W_C^{(1,2)} = W_C^{(2,1)},\ \lambda^{(1,2)} = \lambda^{(2,1)}$$

Apart from changes in notations, the mixing problem in (6.28) is similar to the one in (6.15). This is because of the fact that from (6.8) there is only one combination of sub-networks possible. Hence, one mixing problem is sufficient to address the case of two-part network (6.16).

6.2.2 Three-part mixing problem

Here, a case with three sub-networks is considered. The whole network G is assumed to be decomposed using algorithm 5 as

$$G = H^{(1)} \cup H^{(2)} \cup H^{(3)} \tag{6.29}$$

Let $D_U^{(i)},\ i = 1,\ 2,\ 3$ be the data subsets unique to corresponding sub-networks $H^{(i)}$s. Common part D_C of data D can be further divided into four subsets as given in (6.19). As some of the data subsets are identical, only four distinct data subsets are used to divide D as

$$D = \begin{pmatrix} D_U \\ \hline D_C \end{pmatrix} = \begin{pmatrix} D_U^{(1)} \\ D_U^{(2)} \\ D_U^{(3)} \\ \hline D_C^{(1,2)} \\ D_C^{(1,3)} \\ D_C^{(2,3)} \\ D_C^{(1,2,3)} \end{pmatrix} \tag{6.30}$$

where, $D_U^{(i)}$s are data unique to the three sub-networks. The data subsets $D^{(i)},\ i = 1,\ 2,\ 3$ are of the form

$$D^{(1)} = \begin{pmatrix} D_U^{(1)} \\ \hline D_C^{(1,2)} \\ D_C^{(1,3)} \\ D_C^{(1,2,3)} \end{pmatrix},\ D^{(2)} = \begin{pmatrix} D_U^{(2)} \\ \hline D_C^{(1,2)} \\ D_C^{(2,3)} \\ D_C^{(1,2,3)} \end{pmatrix},\ D^{(3)} = \begin{pmatrix} D_U^{(3)} \\ \hline D_C^{(1,3)} \\ D_C^{(2,3)} \\ D_C^{(1,2,3)} \end{pmatrix} \tag{6.31}$$

Since $X^{(i)},\ i=1,\ 2,\ 3$ are disjoint, source signal matrix factor S can be divided as describved in (6.2) as

$$S = \begin{pmatrix} S^{(1)} \\ S^{(2)} \\ S^{(3)} \end{pmatrix} \tag{6.32}$$

The network parameter matrix factor W can be divided as described in (6.6) as

$$W^{(1)} = \begin{pmatrix} \frac{W_U^{(1)}}{W_C^{(1,2)}} \\ W_C^{(1,3)} \\ W_C^{(1,2,3)} \end{pmatrix},\ W^{(2)} = \begin{pmatrix} \frac{W_U^{(2)}}{W_C^{(2,1)}} \\ W_C^{(2,3)} \\ W_C^{(2,3,1)} \end{pmatrix},\ W^{(3)} = \begin{pmatrix} \frac{W_U^{(3)}}{W_C^{(3,1)}} \\ W_C^{(3,2)} \\ W_C^{(3,1,2)} \end{pmatrix} \tag{6.33}$$

As pointed out earlier, $W_C^{(i,j)}$s in (6.33) are not identical. However, the following holds

$$W_C^{(i,j,k)} = W_C^{(i,k,j)},\ i,j=1,2,3, i\neq j\neq k$$

This implies that as long as the first sub-network of interest $H^{(i)}$ is fixed, the weights $W_C^{(i,\cdot,\cdot)}$ corresponding to the data common to all sub-networks is the same regardless of what order the other sub-network indices appear in the notation. From (6.30), (6.32), and (6.33), the whole network can be rewritten as

$$D = \left(\begin{array}{ccc} W_U^{(1)} & 0 & 0 \\ 0 & W_U^{(2)} & 0 \\ 0 & 0 & W_U^{(3)} \\ \hline \lambda^{(1,2)}W_C^{(1,2)} & \lambda^{(2,1)}W_C^{(2,1)} & 0 \\ \lambda^{(1,3)}W_C^{(1,3)} & 0 & \lambda^{(3,1)}W_C^{(3,1)} \\ 0 & \lambda^{(2,3)}W_C^{(2,3)} & \lambda^{(3,2)}W_C^{(3,2)} \\ \lambda^{(1,2,3)}W_C^{(1,2,3)} & \lambda^{(2,3,1)}W_C^{(2,3,1)} & \lambda^{(3,1,2)}W_C^{(3,1,2)} \end{array}\right) \begin{pmatrix} S^{(1)} \\ S^{(2)} \\ S^{(3)} \end{pmatrix} \tag{6.34}$$

where, $\lambda^{(i,j)}, i,j=1,2,3, i\neq j$ quantify the percentages of corresponding $D_C^{(i,j)}$s factorised respectively by the two sub-networks $H^{(i)}$ and $H^{(j)}$, and $\lambda^{(i,j,k)}$s quantify the percentages of $D_C^{(i,j,k)}$ factorised respectively by all the three sub-networks. All mixing coefficients $\lambda^{(\cdot)}$s in (6.34) satisfy relationships similar to that in (6.3) given

by

$$\begin{aligned} \lambda^{(i,j)}, \lambda^{(i,j,k)} > 0, 1 \leq i,j,k \leq 3, i \neq j \neq k \\ \lambda^{(1,2)} + \lambda^{(2,1)} = 1 \\ \lambda^{(1,3)} + \lambda^{(3,1)} = 1 \\ \lambda^{(2,3)} + \lambda^{(3,2)} = 1 \\ \lambda^{(1,2,3)} + \lambda^{(2,3,1)} + \lambda^{(3,1,2)} = 1 \end{aligned} \tag{6.35}$$

In this case, there are 4 mixing problems corresponding to 4 distinct data subsets in D_C (6.30). Mixing problems in this case are given as

$$\begin{aligned} \min_{\lambda^{(1,2)},\lambda^{(2,1)}} \quad & D_C^{(1,2)} - \lambda^{(1,2)} W_C^{(1,2)} S^{(1)} - \lambda^{(2,1)} W_C^{(2,1)} S^{(2)} \\ \min_{\lambda^{(1,3)},\lambda^{(3,1)}} \quad & D_C^{(1,3)} - \lambda^{(1,3)} W_C^{(1,3)} S^{(1)} - \lambda^{(3,1)} W_C^{(3,1)} S^{(3)} \\ \min_{\lambda^{(2,3)},\lambda^{(3,2)}} \quad & D_C^{(2,3)} - \lambda^{(2,3)} W_C^{(2,3)} S^{(2)} - \lambda^{(3,2)} W_C^{(3,2)} S^{(3)} \\ \min_{\lambda^{(1,2,3)},\lambda^{(2,3,1)},\lambda^{(3,2,1)}} \quad & D_C^{(1,2,3)} - \lambda^{(1,2,3)} W_C^{(1,2,3)} S^{(1)} - \lambda^{(2,3,1)} W_C^{(2,3,1)} S^{(2)} \\ & -\lambda^{(3,2,1)} W_C^{(3,2,1)} S^{(3)} \\ \text{subject to} \quad & (6.35) \end{aligned} \tag{6.36}$$

On comparing the problems in (6.28) and (6.36), it can be seen that the notations get relatively complicated upon adding a sub-network. Addition of another sub-network worsens the readability of notations. A combination with n out of g sub-networks will require $n\ {}^gC_n$ mixing coefficients $\lambda^{(\cdot)}$s. Therefore, the number of parameters in a general case is

$$\sum_{n=2}^{g} n\ {}^gC_n$$

Fig. 6.1 shows the growth in number of mixing parameters on a log scale with an increase in the number of parts constituting the whole network. Furthermore, it is not necessary that for a given pair $(D, B(G))$ all $\sum_{n=2}^{g} {}^gC_n$ combinations of sub-networks will be realised in resulting decomposition. Therefore, in a general case with large number of sub-networks, identifying combination of sub-networks sharing data and corresponding data subsets D_C corresponding, and solving related mixing problems requires a great amount of book-keeping and computational efforts. In order to overcome this hurdle, an iterative approach is developed in the next section.

In summary, the challenges identified in this section in setting up 3DNCA with multiple mixing problems are

1. identifying combinations of sub-networks for a given pair $(D, B(G))$

2. rapid growth in number of mixing coefficients with increase in number of

sub-networks

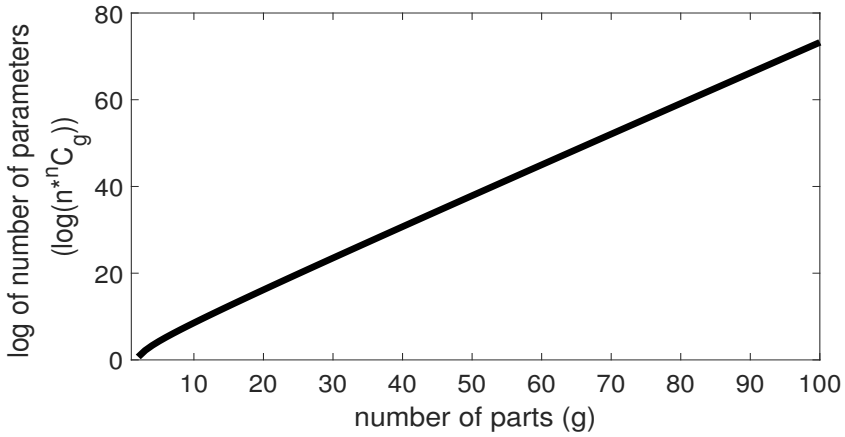

Figure 6.1: Log growth in number of parameters with increasing number of parts

6.3 Iterative approach to multiple mixing problems

In this section, two specific cases – two (6.25) and three (6.34) part networks – presented in the previous section are used to demonstrate how three-part mixing problem (6.36) can be reduced to a series of two two-part mixing problems (6.28). This idea is then extended to a general case. It is also shown that a simple set of notations can be used to solve a general g-part mixing problem thereby reducing the book-keeping effort.

Consier the three-part network G described in (6.29). Let $H^{(A)}$ and $H^{(B)}$ such that

$$H^{(A)} = H^{(1)} \cup H^{(2)},\ H^{(B)} = H^{(3)} \tag{6.37}$$

From (6.29) and (6.37), three-part network G can be rewritten as a two-part network as

$$G = H^{(A)} \cup H^{(B)} \tag{6.38}$$

From (6.37), $H^{(A)}$ is a two-part network. Therefore, at first, a two-part solution $(W^{(A)}, S^{(A)})$ of the form in (6.25) to a 3DNCA problem corresponding to $H^{(A)}$ is sought. The obtained matrix factors are then used to compute a two-part solution corresponding to the network G in (6.38). Thus, a three-part mixing problem can be reduced to a pair of two-part mixing problems. By induction, g-part mixing problem can be solved by sequentially solving $n-1$ two-part problems. Estimates $W^{(i)}$s and $S^{(i)}$s computed in step 2 of 3DNCA do not change in the third step where

mixing problem is to be solved. Therefore, it is not necessary to solve all the mixing problems in one shot. Instead, the identified sub-networks can be sequentially processed in pairs as shown in this section.

6.3.1 Three-part network as a pair of two-part networks

For the three-part network in (6.38), it is shown in the remainder of this section that a solution obtained by iterative application of two-part 3DNCA is equivalent to that obtained by solving three-part 3DNCA.

Assume NCA estimates $(W^{(1)},S^{(1)})$, $(W^{(2)},S^{(2)})$ and $(W^{(3)},S^{(3)})$ are available. Assuming the pair $W^{(A)},S^{(A)}$ corresponding to $H^{(A)}$ (6.37) is available, a complete solution corresponding to G (6.38) – (W,S) as described in (6.34) – can be computed. System equation corresponding to the two-part network G (6.38) is given by

$$D = WS$$

$$\left(\begin{array}{c} D_U^{(A)} \\ \hline D_U^{(B)} \\ \hline D_C^{(A,B)} \end{array}\right) = \left(\begin{array}{c|c} W_U^{(A)} & 0 \\ \hline 0 & W_U^{(B)} \\ \hline \lambda^{(A)}W_C^{(A)} & \lambda^{(B)}W_C^{(B)} \end{array}\right) \left(\begin{array}{c} S^{(A)} \\ \hline S^{(B)} \end{array}\right) \tag{6.39}$$

Note that the quantities in W in (6.39) should have been denoted by the following symbols – $\lambda^{(A,B)}W_C^{(A,B)}$ and $\lambda^{(B,A)}W_C^{(B,A)}$. However, the notation is changed from that used in section 6.2 back to the one presented in section 6.1. This is because a relatively complicated set of notations in section 6.2 is necessary only if there are more than two sub-networks. Discussions in this section follow notations presented in section 6.1. sub-matrices in (6.39) are identified from three-part 3DNCA solution in (6.34) as shown next.

W in (6.34) is reordered as follows

$$W = \left(\begin{array}{cc|c} W_U^{(1)} & 0 & 0 \\ 0 & W_U^{(2)} & 0 \\ \lambda^{(1,2)}W_C^{(1,2)} & \lambda^{(2,1)}W_C^{(2,1)} & 0 \\ 0 & 0 & W_U^{(3)} \\ \hline \lambda^{(1,3)}W_C^{(1,3)} & 0 & \lambda^{(3,1)}W_C^{(3,1)} \\ 0 & \lambda^{(2,3)}W_C^{(2,3)} & \lambda^{(3,2)}W_C^{(3,2)} \\ \lambda^{(1,2,3)}W_C^{(1,2,3)} & \lambda^{(2,3,1)}W_C^{(2,3,1)} & \lambda^{(3,1,2)}W_C^{(3,1,2)} \end{array}\right) \tag{6.40}$$

In order to further simplify the notations, let the mixing coefficients λs in (6.40) be

defined as

$$\begin{gathered}\lambda^{(1,3)} = \lambda^{(2,3)} = \lambda^{(A)} \\ \lambda^{(1,2)} = \lambda^{(1)} \\ \lambda^{(2,1)} = \lambda^{(2)} \\ \lambda^{(1,2,3)} = \lambda^{(B)}\lambda^{(1)} \\ \lambda^{(2,3,1)} = \lambda^{(B)}\lambda^{(2)} \\ \lambda^{(3,1)} = \lambda^{(3,2)} = \lambda^{(3,1,2)} = \lambda^{(B)}\end{gathered} \tag{6.41}$$

From (6.40) and (6.41), W can be rewritten as

$$W = \left(\begin{array}{cc|c} W_U^{(1)} & 0 & 0 \\ 0 & W_U^{(2)} & 0 \\ \lambda^{(1)}W_C^{(2,1)} & \lambda^{(2)}W_C^{(1,2)} & 0 \\ 0 & 0 & W_U^{(3)} \\ \hline \lambda^{(A)}W_C^{(1,3)} & 0 & \lambda^{(B)}W_C^{(3,1)} \\ 0 & \lambda^{(A)}W_C^{(2,3)} & \lambda^{(B)}W_C^{(3,2)} \\ \lambda^{(A)}\lambda^{(1)}W_C^{(1,2,3)} & \lambda^{(A)}\lambda^{(2)}W_C^{(2,3,1)} & \lambda^{(B)}W_C^{(3,1,2)} \end{array}\right) \tag{6.42}$$

On comparing (6.42) to W in(6.39)

$$\begin{gathered} W_U^{(A)} = \begin{pmatrix} W_U^{(1)} & 0 \\ 0 & W_U^{(2)} \\ \lambda^{(1)}W_C^{(2,1)} & \lambda^{(2)}W_C^{(1,2)} \end{pmatrix}, \qquad W_U^{(B)} = W_U^{(3)}, \\ W_C^{(A)} = \begin{pmatrix} W_C^{(1,3)} & 0 \\ 0 & W_C^{(2,3)} \\ \lambda^{(1)}W_C^{(1,2,3)} & \lambda^{(2)}W_C^{(2,3,1)} \end{pmatrix}, \quad W_C^{(B)} = \begin{pmatrix} W_C^{(3,1)} \\ W_C^{(3,2)} \\ W_C^{(3,1,2)} \end{pmatrix} \end{gathered} \tag{6.43}$$

Similarly, dataset D in (6.34) can be used to define D in (6.39) as

$$D = \left(\begin{array}{c} D_U^{(A)} \\ D_U^{(B)} \\ \hline D_C^{(A,B)} \end{array}\right)$$

with

$$D_U^{(A)} = \begin{pmatrix} D_U^{(1)} \\ D_U^{(2)} \\ D_C^{(1,2)} \end{pmatrix}, \; D_U^{(B)} = D_U^{(3)}, \; D_C^{(A,B)} = \begin{pmatrix} D_C^{(1,3)} \\ D_C^{(2,3)} \\ D_C^{(1,2,3)} \end{pmatrix} \tag{6.44}$$

and S in (6.34) can be used to define S in (6.39) as

$$S = \begin{pmatrix} S^{(A)} \\ \hline S^{(B)} \end{pmatrix} = \begin{pmatrix} S^{(1)} \\ S^{(2)} \\ \hline S^{(3)} \end{pmatrix} \tag{6.45}$$

Discussion is briefly diverted to consider $H^{(A)}$ alone. On comparing W in (6.39) to that in (6.42), $W^{(A)}$ can be isolated as

$$W^{(A)} = \left(\begin{array}{c|c} W_U^{(1)} & 0 \\ W_C^{(1,3)} & 0 \\ 0 & W_U^{(2)} \\ 0 & W_C^{(2,3)} \\ \hline \lambda^{(1)} W_C^{(1,2)} & \lambda^{(2)} W_C^{(2,1)} \\ \lambda^{(1)} W_C^{(1,2,3)} & \lambda^{(2)} W_C^{(2,3,1)} \end{array}\right) \tag{6.46}$$

Mixing coefficients $\lambda^{(A)}$ and $\lambda^{(B)}$ are dropped while defining $W^{(A)}$ as they are relevant only when the whole network G is considered. $W^{(A)}$ in (6.46) can also be represented in a form suitable for a two-part mixing problem as

$$W^{(A)} = \begin{pmatrix} W_U^{(A)} \\ \hline W_C^{(A)} \end{pmatrix} = \left(\begin{array}{c|c} W_U^{(A_1)} & 0 \\ 0 & W_U^{(A_2)} \\ \hline \lambda^{(1)} W_C^{(A_1)} & \lambda^{(2)} W_C^{(A_2)} \end{array}\right) \tag{6.47}$$

Symbols A_1 and A_2 are introduced in (6.47) as $W_U^{(A_1)} \neq W_U^{(1)}$, $W_U^{(A_2)} \neq W_U^{(2)}$, $W_C^{(A_1)} \neq W_C^{(1,2)}$, $W_C^{(A_2)} \neq W_C^{(2,1)}$, but

$$\begin{aligned} W_U^{(A_1)} &= \begin{pmatrix} W_U^{(1)} \\ \lambda^{(1,3)} W_C^{(1,3)} \end{pmatrix}, \quad W_U^{(A_2)} = \begin{pmatrix} W_U^{(2)} \\ \lambda^{(2,3)} W_C^{(2,3)} \end{pmatrix} \\ W_C^{(A_1)} &= \begin{pmatrix} W_C^{(1,2)} \\ W_C^{(1,2,3)} \end{pmatrix}, \qquad W_C^{(A_2)} = \begin{pmatrix} W_C^{(2,1)} \\ W_C^{(2,3,1)} \end{pmatrix} \end{aligned} \tag{6.48}$$

dataset $D^{(A)}$ can be identified as

$$\begin{aligned} D^{(A)} &= \begin{pmatrix} D_U^{(A_1)} \\ D_U^{(A_2)} \\ \hline D_C^{(A_1,A_2)} \end{pmatrix}, \\ D_U^{(A_1)} = \begin{pmatrix} D_U^{(1)} \\ D_C^{(1,3)} \end{pmatrix}, \; D_U^{(A_2)} &= \begin{pmatrix} D_U^{(2)} \\ D_C^{(2,3)} \end{pmatrix}, \; D_C^{(A_1,A_2)} = \begin{pmatrix} D_C^{(1,2)} \\ D_C^{(1,2,3)} \end{pmatrix} \end{aligned} \tag{6.49}$$

It can be seen that data subset $D_U^{(3)}$ in (6.30) is not copied to $D^{(A)}$ (6.49). Similarly, the row with $W_U^{(3)}$ in (6.33) is not copied to $W^{(A)}$ (6.46). This is because $H^{(B)} = H^{(3)}$ is not being considered at the moment and hence, is assumed to be non-existent. For the sake of completeness, adding $D_U^{(3)}$ to $D^{(A)}$ would not affect computations as the corresponding row, 4-th row in W (6.40), is a zero-row. Matrix factor $S^{(A)}$ is given by

$$S^{(A)} = \begin{pmatrix} S^{(A_1)} \\ \hline S^{(A_2)} \end{pmatrix} = \begin{pmatrix} S^{(1)} \\ \hline S^{(2)} \end{pmatrix} \tag{6.50}$$

As $H^{(A)}$ is a two-part network, there is only one mixing problem of the form (6.28) to be solved. The mixing problem is given as

$$\begin{aligned} \min_{\lambda^{(1)},\lambda^{(2)}} \quad & \|D_C^{(A_1,A_2)} - \lambda^{(1)} W_C^{(A_1)} S^{(1)} - \lambda^{(2)} W_C^{(A_2)} S^{(2)}\|_F^2 \\ \text{subject to} \quad & \lambda^{(1)}, \lambda^{(2)} > 0 \\ & \lambda^{(1)} + \lambda^{(2)} = 1 \end{aligned} \tag{6.51}$$

A two-part solution of the form (6.25) can be obtained by putting together the quantities in (6.47), (6.49) and (6.50) as

$$D^{(A)} = W^{(A)} S^{(A)} \tag{6.52}$$

Bringing the focus of discussion back to the whole network G (6.38), matrix factors W and S in (6.39) can be computed by solving for $\lambda^{(A)}$ and $\lambda^{(B)}$ in another two-part mixing problem with A and B in place of A_1 and A_2 in (6.51). A two-part solution to a 3DNCA problem corresponding to G in (6.29) can be obtained as described in (6.39) by putting together all the required quantities in (6.42).

Discussion in this section up to this point shows that a three-part mixing problem in (6.36) can be translated to a series of two two-part mixing problems of the form (6.51). A solution $D = WS$ can be obtained by aggregating all computations as described in (6.39). However, it remains to verify if products of λs in the last row of W in (6.42) satisfy convexity constraints similar to those in (6.3). This ensures that all common data subsets in D_C are factorised to an extent of 100%. A quick check can confirm that all mixing coefficients satisfy convexity constraints imposed in (6.3).

As claimed at the beginning of this section, it is shown in this section that a three-part mixing problem can indeed be reduced to a pair of two-part mixing problems. By extension, a general g-part mixing problem can be reduced to a series of $g-1$ two-part mixing problems that are to be solved sequentially.

6.3.2 Divide and conquer algorithm

Following the discussions in the previous subsection, an algorithm is developed in this subsection that processes a given network G with g parts in $g-1$ steps. Assuming $g > 1$, the algorithm starts with $H^{(A)} = H^{(1)}$ and $H^{(B)} = H^{(2)}$ to compute a solution of the form (6.39). For every additional sub-network $H^{(i)}, 3 \leq i \leq g$, $H^{(A)}$ and $H^{(B)}$ from the previous step are used to update the definitions of sub-networks considered in (6.39) as

$$H^{(A)} = H^{(A)} \cup H^{(B)},\ H^{(B)} = H^{(i)}$$

Definitions of unique and common parts of sub-networks $H^{(A)}$ and $H^{(B)}$, and data are also updated at each step. While updating the definitions of all variables, mixing coefficients $\lambda^{(A)}$ and $\lambda^{(B)}$ are considered as integral parts of $W^{(A)}$ and $W^{(B)}$, respectively. Mixing coefficients computed at the current step are multiplied to the common parts of $W^{(A)}$ and $W^{(B)}$ regardless of whether some of the rows in there were multiplied with mixing coefficients in the previous step. In this sense, each step is independent of the previous step. Steps described up to this point are summarised in the form of an algorithm as It can be seen in step 3 of algorithm 6 that if

Algorithm 6 3DNCA

inputs: D, $B(G)$
STEP 1: Identify sub-graphs $H^{(i)}$s using algorithm 5 with $B(G)$ as input
STEP 2: Estimate $W^{(i)}$s and $S^{(i)}$s using NCA
STEP 3: set $H^{(A)} = H^{(1)}$
if $g > 1$ **then**
  for $i = 2$ to g **do**
    set $H^{(B)} = H^{(i)}$
    update definitions of $D_U^{(A)}$,$D_U^{(B)}$, $D_C^{(A,B)}$, $W_U^{(A)}$, $W_C^{(A)}$, $W_U^{(B)}$ and $W_C^{(B)}$
    estimate W in (6.39)
    set $H^{(A)} = H^{(A)} \cup H^{(B)}$
  end for
end if
outputs: $W = W^{(A)}$ and $S = S^{(A)}$

there is only one sub-network $H^{(A)}$, then that is the solution to the SCMF problem in (2.12). Therefore, as pointed out earlier, NCA [2] is a special case of 3DNCA.

As mentioned at the beginning of this chapter, the most important assumption in 3DNCA is that a given dataset-network pair $(D, B(G))$ follows some mixing rule. In other words, λs are defined. For some pair $(D, B(G))$, if λs are not defined, then it is not guaranteed that 3DNCA solutions will result in small errors in reconstructing

D. An example is provided later in this chapter where, 3DNCA results in unique matrix factors, but large errors are observed in reconstructing corresponding D.

6.4 Uniqueness of solutions

Existence of a diagonal matrix χ as described in (2.14) establishes the uniqueness of NCA solutions up to a scaling factor for an NCA feasible network. Uniqueness of 3DNCA solutions obtained by applying algorithm 6 is established along similar lines. A key assumption in establishing uniqueness of 3DNCA solutions is that all data subset-sub-network pairs $(D^{(i)}, B(H^{(i)}), 1 \leq i \leq g$ are NCA feasible. It is shown in section (5.4.2) that algorithm 5 guarantees full-rank factorisability of all sub-networks. However, not all pairs $(D^{(i)}, B(H^{(i)})$s are necessarily NCA feasible. This is because of the fact that the number of source signals $|X^{(i)}|$ in some of the sub-networks are lesser than $rk([B(H^{(i)} : D^{(i)}])$ as shown in (5.29). Assuming NCA feasibility of all data subset-sub-network pairs, uniqueness of 3DNCA solutions can be established as described in this section.

In section (6.3.1), it is shown that a three-part mixing problem can be reduced to a series of two two-part mixing problems. Algorithm 6 implements this idea in a way such that g sub-networks are iteratively processed in $g-1$ pairs of sub-networks in a decoupled manner. It is shown in (**??**) that convexity constraints imposed on λs are satisfied in the iterative process in 3DNCA. Therefore, it s sufficient to establish uniqueness of solutions (6.39) corresponding to a two-part network $G = H^{(A)} \cup H^{(B)}$ (6.37). However, uniqueness of 3DNCA solutions cannot be established as with NCA solutions as described in [2]. This is because $\lambda^{(A)}$ and $\lambda^{(B)}$ appear inside W in (6.39). Role of these mixing factors must be carefully analysed while establishing uniqueness of 3DNCA solutions. It is demonstrated next that 3DNCA solutions of the form (6.39) are not unique up to a single scaling matrix χ as in the case of NCA.

Let sub-matrices $({}^1W_U^{(A)})$, $({}^1W_C^{(A)})$ and $({}^1S^{(A)})$ corresponding to $H^{(A)}$ be such that

$$D^{(A)} = ({}^1W^{(A)})({}^1S^{(A)}) = \begin{pmatrix} D_U^{(A)} \\ \hline D_C^{(A,B)} \end{pmatrix} = \begin{pmatrix} ({}^1W_U^{(A)}) \\ \hline ({}^1W_C^{(A,B)}) \end{pmatrix} ({}^1S^{(A)})$$

Similarly, sub-matrices $({}^2W_U^{(A)})$, $({}^2W_C^{(A)})$ and $({}^2S^{(A)})$ corresponding to $H^{(A)}$ and sub-matrices $({}^1W_U^{(B)})$, $({}^1W_C^{(B)})$, $({}^1S^{(B)})$, $({}^2W_U^{(B)})$, $({}^2W_C^{(B)})$ and $({}^2S^{(B)})$ corresponding to $H^{(B)}$ can be defined. It is assumed that NCA is used to compute these sub-

matrices such that

$$D^{(A)} = ({}^1W^{(A)})({}^1S^{(A)}) = ({}^2W^{(A)})({}^2S^{(A)})$$

and

$$D^{(B)} = ({}^1W^{(B)})({}^1S^{(B)}) = ({}^2W^{(B)})({}^2S^{(B)})$$

As the two sub-networks $H^{(A)}$ and $H^{(B)}$ are NCA feasible, their corresponding NCA solutions are unique according to theorem 1. In other words, the corresponding NCA solutions are related by diagonal matrices $\chi^{(A)}$ and $\chi^{(B)}$ as

$$\begin{aligned} ({}^1W^{(A)})\chi^{(A)} = ({}^2W^{(A)}),\ ({}^1S^{(A)}) = \chi^{(A)}({}^2S^{(A)}), \\ ({}^1W^{(B)})\chi^{(B)} = ({}^2W^{(B)}),\ ({}^1S^{(B)}) = \chi^{(B)}({}^2S^{(B)}) \end{aligned} \tag{6.53}$$

Using these sub-matrices, two 3DNCA solutions of the form (6.39) corresponding to $G = H^{(A)} \cup H^{(B)}$ can be defined as

$$D = ({}^1W)({}^1S)$$

$$\left(\begin{array}{c} D_U^{(A)} \\ \hline D_U^{(B)} \\ \hline D_C^{(A,B)} \end{array}\right) = \left(\begin{array}{c|c} ({}^1W_U^{(A)}) & 0 \\ \hline 0 & ({}^1W_U^{(B)}) \\ \hline ({}^1\lambda^{(A)})({}^1W_C^{(A)}) & ({}^1\lambda^{(B)})({}^1W_C^{(B)}) \end{array}\right) \left(\begin{array}{c} ({}^1S^{(A)}) \\ ({}^1S^{(B)}) \end{array}\right) \tag{6.54}$$

and

$$D = ({}^2W)({}^2S)$$

$$\left(\begin{array}{c} D_U^{(A)} \\ \hline D_U^{(B)} \\ \hline D_C^{(A,B)} \end{array}\right) = \left(\begin{array}{c|c} ({}^2W_U^{(A)}) & 0 \\ \hline 0 & ({}^2W_U^{(B)}) \\ \hline ({}^2\lambda^{(A)})({}^2W_C^{(A)}) & ({}^2\lambda^{(B)})({}^2W_C^{(B)}) \end{array}\right) \left(\begin{array}{c} ({}^2S^{(A)}) \\ ({}^2S^{(B)}) \end{array}\right) \tag{6.55}$$

Values of mixing factors $({}^1\lambda^{(A)})$ and $({}^1\lambda^{(B)})$ in (6.54) depend on the values of NCA solutions $(({}^1W^{(A)}), ({}^1S^{(A)}))$ and $(({}^1W^{(B)}), ({}^1S^{(B)}))$. Similarly, values of $({}^2\lambda^{(A)})$ and $({}^2\lambda^{(B)})$ in (6.55) depend on the values $(({}^2W^{(A)}), ({}^2S^{(A)}))$ and $(({}^2W^{(B)}),\ ({}^2S^{(B)}))$. As NCA solutions corresponding $H^{(A)}$ and $H^{(B)}$ are not identical, corresponding mixing factors are also not identical

$$({}^1\lambda^{(A)}) \neq ({}^2\lambda^{(A)}),\ ({}^1\lambda^{(B)}) \neq ({}^2\lambda^{(B)})$$

Using the relationships in (6.53), consider rearranging right hand side of the equa-

tion in (6.55) as

$$\left(\begin{array}{c|c} \frac{({}^1\lambda^{(A)})}{({}^2\lambda^{(A)})}({}^1W_U^{(A)})\chi_U^{(A)}\frac{({}^2\lambda^{(A)})}{({}^1\lambda^{(A)})} & 0 \\ 0 & \frac{({}^1\lambda^{(B)})}{({}^2\lambda^{(B)})}({}^1W_U^{(B)})\chi_U^{(B)}\frac{({}^2\lambda^{(B)})}{({}^1\lambda^{(B)})} \\ \hline \frac{({}^2\lambda^{(A)})}{({}^1\lambda^{(A)})}({}^1\lambda^{(A)})({}^1W_C^{(A)})\chi_C^{(A)} & \frac{({}^2\lambda^{(B)})}{({}^1\lambda^{(B)})}({}^1\lambda^{(B)})({}^1W_C^{(B)})\chi_C^{(B)} \end{array}\right)\left(\begin{array}{c} \chi^{(A)^{-1}}({}^1S^{(A)}) \\ \chi^{(B)^{-1}}({}^1S^{(B)}) \end{array}\right) \tag{6.56}$$

It can be verified that right hand side of the equation in (6.55) is equivalent to the representation in (6.56). Consider two diagonal matrices Λ_L and Λ_R whose principal diagonal $pd(\cdot)$s are given as

$$\begin{aligned} pd(\Lambda_R) &= \left(\begin{array}{ccc|ccc|ccc} \frac{({}^1\lambda^{(A)})}{({}^2\lambda^{(A)})} & \cdots & \frac{({}^1\lambda^{(A)})}{({}^2\lambda^{(A)})} & \frac{({}^1\lambda^{(B)})}{({}^2\lambda^{(B)})} & \cdots & \frac{({}^1\lambda^{(B)})}{({}^2\lambda^{(B)})} & 1 & \cdots & 1 \end{array}\right) \\ pd(\Lambda_L) &= \left(\begin{array}{ccc|ccc} \frac{({}^2\lambda^{(A)})}{({}^1\lambda^{(A)})} & \cdots & \frac{({}^2\lambda^{(A)})}{({}^1\lambda^{(A)})} & \frac{({}^2\lambda^{(B)})}{({}^1\lambda^{(B)})} & \cdots & \frac{({}^2\lambda^{(B)})}{({}^1\lambda^{(B)})} \end{array}\right) \end{aligned} \tag{6.57}$$

where, the three parts of $pd(\Lambda_L)$ correspond respectively to the rows in $W_U^{(A)}$, $W^{(B)}$, and W_C whereas the two parts of $pd(\Lambda_R)$ correspond to the columns of W representing respectively $H^{(A)}$ and $H^{(B)}$. From (6.56) and (6.57), right hand side of the equation in (6.55) can be rewritten as

$$\Lambda_L\left(\begin{array}{c|c} ({}^1W_U^{(A)}) & 0 \\ 0 & ({}^1W_U^{(B)}) \\ \hline ({}^1\lambda^{(A)})({}^1W_C^{(A)}) & ({}^1\lambda^{(B)})({}^1W_C^{(B)}) \end{array}\right)\chi\chi^{-1}\left(\begin{array}{c} ({}^1S^{(A)}) \\ ({}^1S^{(B)}) \end{array}\right) \tag{6.58}$$

where, χ is given as

$$\chi = \left(\begin{array}{cc} \chi^{(A)} & 0 \\ 0 & \chi^{(B)} \end{array}\right)\Lambda_R^{\frac{1}{2}} \tag{6.59}$$

As $\chi^{(A)}$, $\chi^{(B)}$ and Λ_R in (6.59) are diagonal matrices, χ is also diagonal in nature. From (6.58), the two 3DNCA solutions in (6.54) and (6.55) can be related to each other as

$$({}^2W)({}^2S) = \Lambda_L({}^1W)\chi\chi^{-1}({}^1S) \tag{6.60}$$

A theorem on uniqueness of 3DNCA solutions is stated as

Theorem 5 *Given a g-part G described by*

$$G = \bigcup_{i=1}^{g} H^{(i)}$$

3DNCA solutions (W,S) obtained by using algorithm 6 are unique up to two scaling factors Λ_L (6.57) and χ (6.60)

Proof For a general g-part network G, two pairs of solutions $(({}^1W),({}^1S))$ and $(({}^2W),({}^2S))$ can be obtained by using 3DNCA algorithm (algorithm 6). Λ_L and Λ_R (6.57), and χ (6.59) corresponding to the two solutions being computed can be calculated at every iteration of 3DNCA algorithm. Therefore, Λ_L and χ calculated in the last iteration of 3DNCA captures scaling factors corresponding to each iteration. Λ_L and χ from the last iteration characterise the uniqueness of solutions corresponding to the whole network G as described in (6.60). □

6.5 A multi-part network example

In section 5.4.1, example network in (3.1) is used to test the applicability of algorithm 4. It is not possible to test the validity of algorithm 5 as an associated dataset D is not available. In this section, a relatively larger example network $B(G) \in GF_2^{24\times 12}$ is introduced. Using randomly chosen matrices $W, \phi(W) = B(G)$ and S, a dataset $D \in \mathbb{R}^{24\times 4}$ is constructed as $D = WS$. The newly introduced dataset-network pair $(D, B(G))$ is then used to test the validity of algorithm 5. Note that the number of source signals in S is $L = 12$ whereas the number of data points available is $N = 4$. Therefore, application of algorithm 5 is warranted. Once the full-rank factorisable sub-networks of G are identified, assuming W and S are unknown, the goal in this section is to estimate (W,S) given $(D,B(G))$ using 3DNCA (algorithm 6).

Note that, as mentioned in section 6.3.2, algorithm 6 (3DNCA) assumes that the underlying dataset is obtained from a system with well defined mixing rules. In other words, 3DNCA results in minimal error in reconstructing a dataset D if the corresponding W is not a random choice. In this section, W is chosen randomly and hence, it is not expected that any mixing rule holds. Therefore, 3DNCA is not expected to perform better than NCA in terms of reconstructing D, but it is shown to generate unique solutions as described in section 6.4. This example serves to demonstrate both strengths and weaknesses of the divide and conquer algorithm developed in chapter.

Example network description – Consider a network $B(G)$ shown in Fig. 6.2. It can be verified that $B(G)$ has full-column rank in both $\mathbb{R}$ and GF_2

$$rk(B(G)) = rk_2(B(G)) = 12$$

Matrices $W \in \mathbb{R}^{24\times 12}$ and $S \in \mathbb{R}^{12\times 4}$ are such that

$$\phi(W) = B(G),\ rk(S) = \min(L,N) = 4$$

Matrix entries $W(i,j)$ and $S(j,k)$ with $1 \leq i \leq 24$, $1 \leq j \leq 12$ and $1 \leq k \leq 4$ are chosen randomly as

$$W(i,j) = w_{min} + w_{ij} \times (w_{max} - w_{min})$$
$$S(j,k) = s_{min} + s_{jk} \times (s_{max} - s_{min})$$

where, w_{ij} and s_{jk} are random numbers chosen from a uniform distribution $\mathcal{U} \sim [0,1]$, $w_{max}, s_{max} = 10$ and $w_{min}, s_{min} = -10$. W and S are as shown respectively in Figs. 6.3 and 6.4. A dataset D corresponding to $B(G)$ in Fig. (6.2) is constructed as

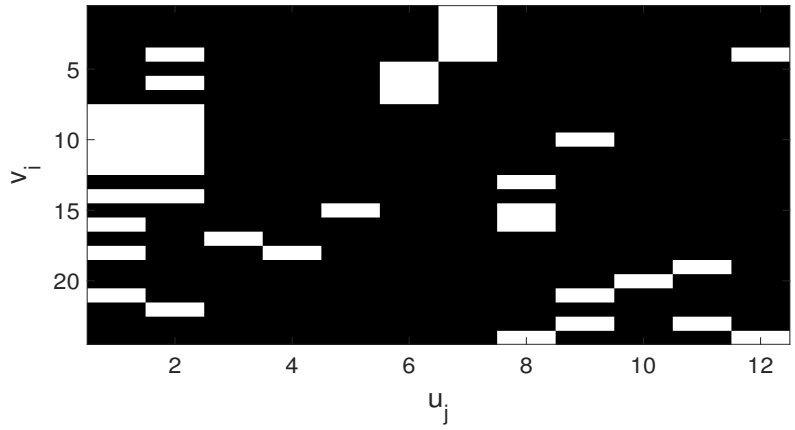

Figure 6.2: Colour map of $B(H)$ (0 – black, 1 – white): NCA infeasible example network

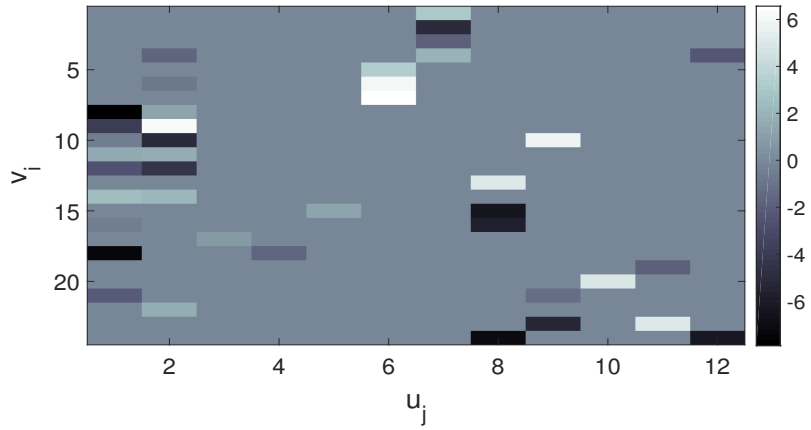

Figure 6.3: Colour map of W corresponding to $B(G)$ in Fig. 6.2

$$D = WS$$

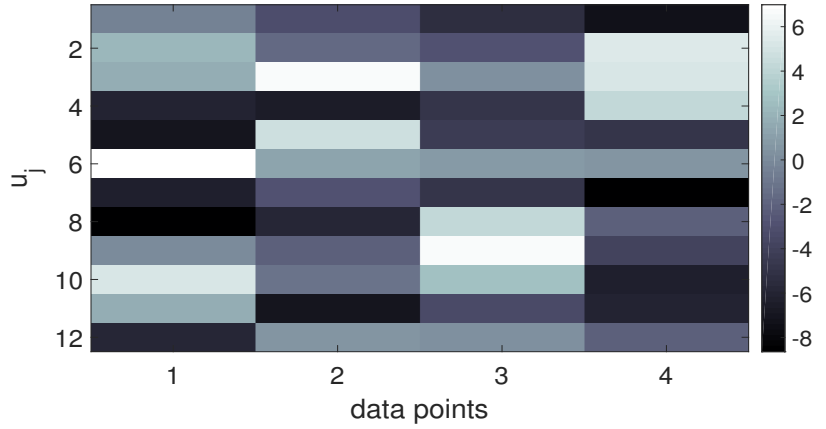

Figure 6.4: Colour map of S corresponding to $B(G)$ in Fig. 6.2

D, W and S are as shown respectively in Figs. 6.5,

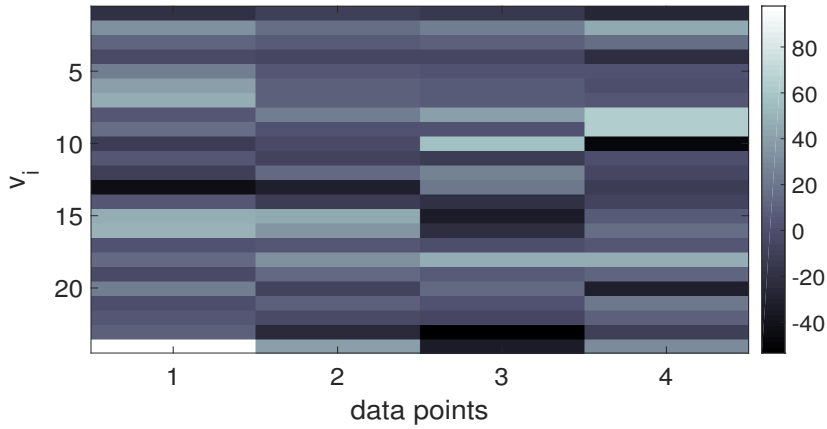

Figure 6.5: Colour map of D corresponding to $B(G)$ in Fig. 6.2

Network decomposition – $B(G)$ is NCA infeasible according to theorem 4 as induced sub-graphs H_js of $B(G)$ are such that

$$rk_2(H_1) < L-1,\ rk_2(H_8) < L-1,\ rk_2(H_j) = L-1,\ 1 \leq j \leq 12,\ j \neq 1,8 \quad (6.61)$$

On applying algorithm 4, two NCA feasible sub-networks of $B(G)$ can be identified as

$$H^{(1)} = (X^{(1)} \cup Y^{(1)}, B(H^{(1)})),\ H^{(2)} = (X^{(2)} \cup Y^{(2)}, B(H^{(2)})) \quad (6.62)$$

where, $X^{(1)}$ and $X^{(2)}$ are given as

$$X^{(1)} = \{u_1,\ u_8\},\ X^{(2)} = \{u_j,\ 1 \leq j \leq 12,\ j \neq 1,8\} \quad (6.63)$$

This is not surprising as H_1 and H_8 are not full-ranked as seen in (6.61). However, sub-networks identified in (6.63) cannot be used in the context of NCA as D has only 4 data points. Therefore, pair $(D, B(G))$ is NCA infeasible as well. Full-rank factorisable sub-networks of $B(G)$ are identified by applying algorithm 5 as

$$\begin{aligned} H^{(i)} &= (X^{(i)} \cup Y^{(i)}, B(H^{(i)})), 1 \leq i \leq 4, \\ X^{(1)} &= \{u_1,\ u_8\},\ X^{(2)} = \{u_2,\ u_3,\ u_4,\ u_5\}, \\ X^{(3)} &= \{u_6,\ u_7,\ u_9,\ u_{10}\},\ X^{(4)} = \{u_{11},\ u_{12}\} \end{aligned} \tag{6.64}$$

The whole network G can be represented as

$$G = \bigcup_{i=1}^{4} H^{(i)} \tag{6.65}$$

where, sub-networks $H(i)$s are as described in (6.64). For $1 \leq i \leq 4$, number of source signals need in each $S^{(i)}$ to ensure NCA feasibility of corresponding data subset-network pair $(D^{(i)}, B(H^{(i)})$ are given as

$$\begin{aligned} rk([B(H^{(1)})\ :\ D^{(1)}]) &= 6 \\ rk([B(H^{(2)})\ :\ D^{(2)}]) &= 8 \\ rk([B(H^{(3)})\ :\ D^{(3)}]) &= 8 \\ rk([B(H^{(4)})\ :\ D^{(4)}]) &= 4 \end{aligned} \tag{6.66}$$

From (6.64) and (6.66), $(D^{(1)}, B(H^{(1)}))$ is NCA infeasible according to theorem 3 whereas rest of the data subset-sub-network pairs are NCA feasible. However, all sub-networks are guaranteed to be full-rank factorisable as a result of application of algorithm 5.

3**DNCA solutions** – Two pairs of NCA estimates $(({}^1W^{(i)}), ({}^1S^{(i)}))$ and $(({}^2W^{(i)}), ({}^2S^{(i)}))$ are obtained from pairs $(D, B(H^{(i)}))$s corresponding to the sub-networks $H^{(i)}$s in (6.64). Diagonal matrices $\chi^{(i)}$s relating two pairs of solution of each sub-network are calculated as described in (3.35). Principal diagonal of $\chi^{(i)}$s are determined as

$$\begin{aligned} pd(\chi^{(1)}) &= \begin{pmatrix} -2.03 & 1 \end{pmatrix} \\ pd(\chi^{(2)}) &= \begin{pmatrix} -1.44 & -0.13 & 1.62 & -0.005 \end{pmatrix} \\ pd(\chi^{(3)}) &= \begin{pmatrix} 1.26 & 1 & 1 & 2.27 \end{pmatrix} \\ pd(\chi^{(4)}) &= \begin{pmatrix} -1.6 & 0.49 \end{pmatrix} \end{aligned} \tag{6.67}$$

From theorem 1, $\chi^{(2)}$, $\chi^{(3)}$ and $\chi^{(4)}$ are guaranteed to be diagonal in nature as the corresponding data subset-sub-network pairs are NCA feasible. Though $(D^{(1)}, B(H^{(1)}))$

$H^{(A)}$	$H^{(B)}$	$({}^1\lambda^{(A)})$	$({}^1\lambda^{(B)})$	$({}^2\lambda^{(A)})$	$({}^2\lambda^{(B)})$
$H^{(1)}$	$H^{(2)}$	0.2429	0.7571	0.2365	0.7635
$H^{(1)} \cup H^{(2)}$	$H^{(3)}$	0.1989	0.8011	0.1978	0.8022
$H^{(1)} \cup H^{(2)} \cup H^{(3)}$	$H^{(4)}$	0.2066	0.7934	0.2061	0.7939

Table 6.1: sub-network descriptions and values of mixing factors at each iteration of 3DNCA

is NCA infeasible, corresponding $\chi^{(1)}$ is diagonal. However, this does not imply NCA feasibility. It can be verified that for all $1 \leq i \leq 4$

$$({}^2W^{(i)}) = ({}^1W^{(i)})\chi^{(i)},\ ({}^2S^{(i)}) = \chi^{(i)^{-1}}({}^1S^{(i)})$$

Now, two pairs of solutions $(({}^1W), ({}^1S))$ and $(({}^2W), ({}^2S))$ are obtained by applying 3DNCA (algorithm 6) to the multi-part network G in (6.65). Values of mixing factors at each iteration is tabulated in table 6.1. Each row of table 6.1 represents one iteration in step 3 of algorithm 6. sub-networks that make up $H^{(A)}$ and $H^{(B)}$ at each iteration of 3DNCA are tabulated respectively in first two column of table 6.1 whereas next two pairs of columns respectively tabulate the values of mixing factors $\lambda^{(A)}$ and $\lambda^{(B)}$ corresponding to the two 3DNCA solutions being computed.

In order to characterise the uniqueness of 3DNCA solutions, diagonal matrices Λ_L and Λ_R must be calculated. This can be done by calculating diagonal matrices described in (6.57) at each iteration of algorithm 6. Λ_L and Λ_R obtained in the last iteration can be used to characterise uniqueness of G as described in (6.58). However, from table 6.1, for every iteration

$$({}^1\lambda^{(i)}) \approx ({}^2\lambda^{(i)}),\ 1 \leq i \leq 4$$

Therefore, final values of Λ_L and Λ_R can be calculated directly as

$$\Lambda_L = I_{24}, \quad \Lambda_R = I_4 \tag{6.68}$$

where, I_{24} and I_4 are identity matrices of size 24 and 4, respectively. From (6.59)

and (6.68), theoretical value of matrix χ can be evaluated as

$$\chi = \begin{pmatrix} \chi^{(1)} & 0 & 0 & 0 \\ 0 & \chi^{(2)} & 0 & 0 \\ 0 & 0 & \chi^{(3)} & 0 \\ 0 & 0 & 0 & \chi^{(4)} \end{pmatrix} \Lambda_R^{\frac{1}{2}}$$
$$\Lambda_R^{\frac{1}{2}} = I_4, \implies \chi = \begin{pmatrix} \chi^{(1)} & 0 & 0 & 0 \\ 0 & \chi^{(2)} & 0 & 0 \\ 0 & 0 & \chi^{(3)} & 0 \\ 0 & 0 & 0 & \chi^{(4)} \end{pmatrix} \tag{6.69}$$

where, $\chi^{(i)},\ 1 \leq i \leq 4$ are as defined in (6.67). If algorithm 6 works as expected, then $\hat{\chi}$ computed as described in (3.35) from two 3DNCA solutions $(({}^1W),({}^1S))$ and $(({}^2W),({}^2S))$ must be equal to χ in (6.69). Principal diagonal entries of $\hat{\chi}$ evaluated as described in (3.35) as

$$pd(\hat{\chi}) = \left(\begin{array}{cc|cccc|} -2.04 & 1 & -1.44 & -1.13 & 1.61 & -0.005 \\ & & & & & \\ & & 1.26 & 1 & 1 & 2.27 \end{array} \right.\left. \begin{array}{cc} & \\ & \\ -1.6 & 0.49 \end{array} \right) \tag{6.70}$$

where, vertical bars are used to separate entries corresponding to the four sub-networks $H^{(i)}$s. From (6.67), (6.69) and (6.70), theoretical and estimated values of χ are in agreement

$$pd(\chi) = pd(\hat{\chi}) \tag{6.71}$$

However, some of the off-diagonal entries in $\hat{\chi}$ evaluate to

$$\begin{gathered} \hat{\chi}(1,3) = 0.02,\ \hat{\chi}(3,1) = -0.002, \\ \hat{\chi}(5,1) = 0.003,\ \hat{\chi}(5,3) = 0.007,\ \hat{\chi}(6,2) = -0.004 \end{gathered} \tag{6.72}$$

An interesting observation is that all the non-zero off-diagonal entries identified in (6.72) have either row or column indices equal to 1 or 2. This implies that uniqueness of 3DNCA solutions is affected by NCA estimates corresponding to $H^{(1)}$. This behaviour is expected as $(D^{(1)}, B(H^{(1)}))$ is found to be NCA infeasible as shown in (6.66).

Numbers in (6.72) will not be visible on a colour map of $\hat{\chi}$ as they are smaller by two orders compared to those in (6.70). Therefore, for the purpose of better illustration, Fig. 6.6 shows $100 \times \hat{\chi}$ after setting diagonal entries to zero. White dashed line

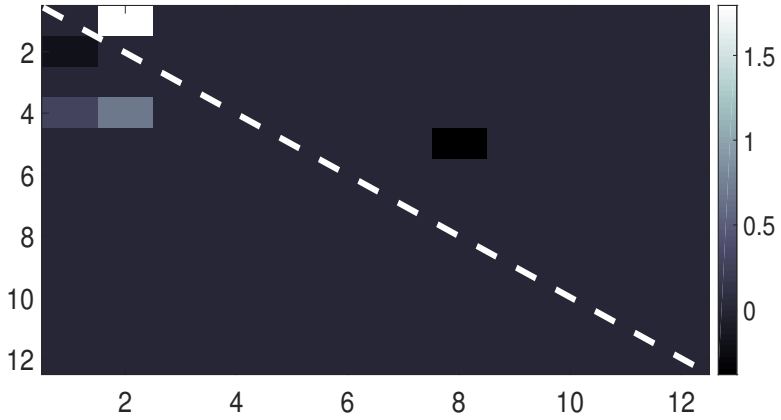

Figure 6.6: Colour map of $100 \times \hat{\chi}$ corresponding to example network after setting diagonal entries to zero

in Fig. 6.6 runs along the principal diagonal. In Fig. 6.6, lighter colours represent positive non-zero values. It can be seen in Fig. 6.6 that there are a handful number of off-diagonal entries that are non-zero as identified in (6.72).

Compring 3DNCA with NCA [2] – Applying NCA to $B(G)$ in Fig. 6.2 and D in Fig. 6.5 is futile as neither $B(G)$ nor the pair $(D, B(G))$ is NCA feasible. However, NCA is applied to the pair $(D, B(G))$ in this section to obtain two pairs of solutions $(({}^1W_N), ({}^1S_N))$ and $(({}^2W_N), ({}^2S_N))$, where subscript N represents NCA. A matrix χ_N is sought by solving the equation in (3.35). As in the case of $\hat{\chi}$ in Fig. 6.6, χ_N is shown in Fig. 6.7 after setting its diagonal entries to zero.

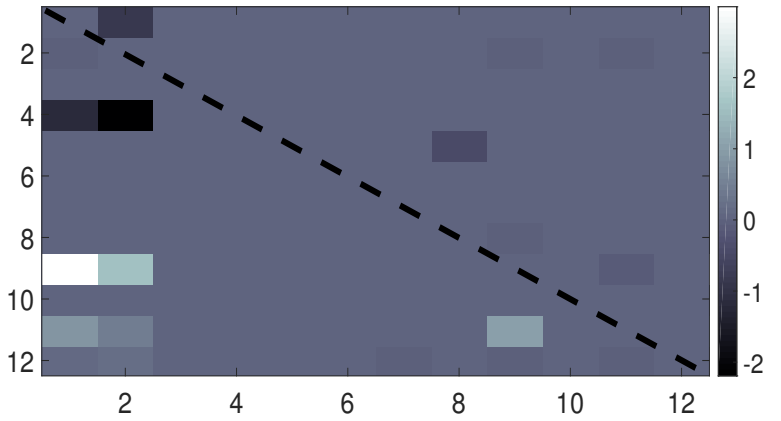

Figure 6.7: Colour map of χ_N corresponding to example network after setting diagonal entries to zero

In Fig. 6.7, black dashed line runs along the principal diagonal and several off-diagonal entries are non-zero. On comparing Fig. 6.7 and Fig. 6.6, it can be seen that number of off-diagonal entries that are non-zero in χ_N is bigger than that in $\hat{\chi}$. Furthermore, $\hat{\chi}$ in Fig. 6.6 is scaled by a factor of 100 in order to improve the contrast between zero and non-zero values whilst illustrating. On the other hand, χ_N in Fig. 6.7 has not been scaled up or down. Despite that, non-zero entries can be clearly identified in Fig. 6.7. This implies that the non-zero off-diagonal entries in χ_N are comparable in value to those on its principal diagonal. This clearly indicates that NCA solutions are not any close to being unique for the pair $(D, B(G))$ considered in this section.

Data reconstruction – First step of 3DNCA involves reordering columns to separate out sub-networks $H^{(i)}$s. Third step of 3DNCA involves reordering rows to identify unique and common parts of the two sub-networks considered in that iteration. Therefore, it is important to ensure that the rows of $({}^1W)$ and $({}^2W)$ are aligned with the original order as dictated by $B(G)$ in Fig. 6.2. Once this is done, error $\hat{\Gamma}$ in reconstructing D can be calculated as described in (2.15).

Absolute values of error in reconstructing D using NCA is shown in Fig. 6.8 while that with one of the two 3DNCA solutions is shown in Fig. 6.9. The two figures are plotted on same scale – 0 to 30 – as seen on the colour bars to their left. On comparing the two figures, NCA results in almost zero error overall as seen in in Fig. 6.8. However, uniqueness of NCA solutions cannot be guaranteed as discussed earlier. On the other hand, from Fig. 6.9, 3DNCA results in significant errors in reconstructing D. This shows that 3DNCA computes unique solutions corresponding to a dataset D resulting from a random choice of W and S, but low error in reconstructing D cannot be expected.

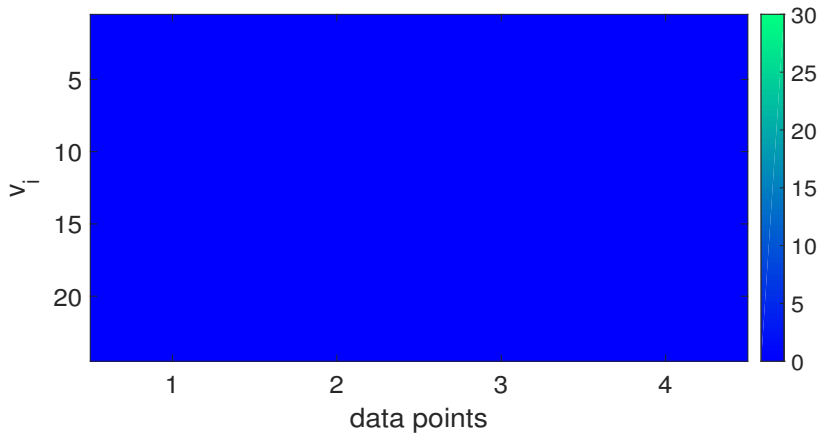

Figure 6.8: Colour map of $|\hat{\Gamma}|$ with NCA corresponding to example network

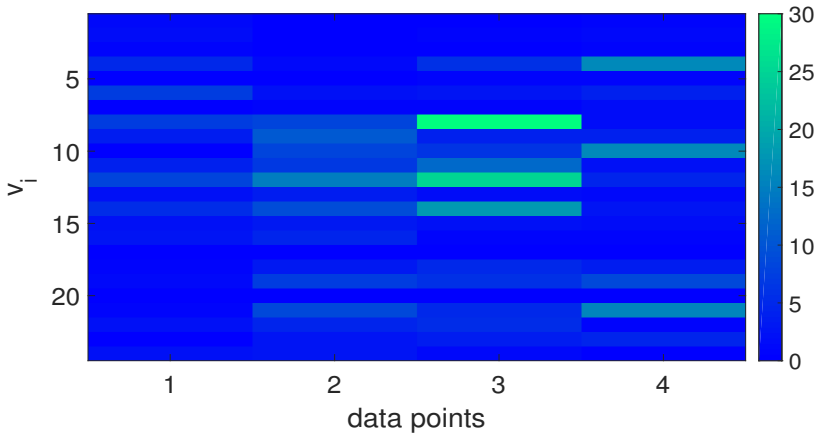

Figure 6.9: Colour map of $|\hat{\Gamma}|$ with 3DNCA corresponding to example network

6.6 Experimental data based algorithm validation

In the previous section, it is shown that 3DNCA generates unique solutions, but it does not necessarily guarantee low errors in reconstructing a dataset D. This is because the matrix factors W and S in the previous section were chosen randomly. In this section, the dataset D of interest is generated from an experiment involving a biological system in [1]. As mentioned in at the beginning of chapter 1, biological systems are modular in nature. As pointed out in section 1.2, transcriptional regulation is a stable process with well defined pathways. Therefore, $B(G)$ in (2.4) can be interpreted as a multi-part network and hence, W is unlikely to be random. In section 2.1, it is shown that protein activity can be modelled using first-order differential equations. Therefore, S is unlikely to be random.

Goals in this section are to show that 3DNCA generates unique solutions that reconstruct an experimental dataset with low error and to show that 3DNCA estimates agree with experimentally observed facts. It is shown that the error in reconstructing experimental data D in [1] using 3DNCA is comparable to that with NCA. As mentioned at the beginning of this chapter, 3DNCA is not expected to perform significantly better than NCA in terms of data reconstruction as 3DNCA builds on NCA sub-networks estimates. However, an important improvement over NCA is that 3DNCA solutions are shown to be unique.

It is shown in section 3.3.2 that estimates of matrix factors obtained by applying NCA to the experimental dataset D corresponding to the carbon source switching network (section 2.5) are not unique. Two pairs of NCA solutions are estimated

such that

$$D = (({}^1W), ({}^1S)) = (({}^2W), ({}^2S))$$

In (**??**), diagonal matrix χ is calculated that relates the two weight matrix estimates $({}^1W)$ and $({}^2W)$ as

$$({}^2W) = ({}^1W)\chi$$

However, it is shown in (3.36) that

$$({}^2S) \neq \chi^{-1}(({}^1S)$$

It is shown in section 4.4.2 that the dataset-network pair $(D, B(G))$ corresponding to the carbon source switching experiment in [1] is NCA infeasible. Therefore, classical NCA in [2] cannot be applied to $(D, B(G))$. In this section, 3DNCA (algorithm 6) is applied to this experimental dataset-network pair.

Sub-network matrix factors – First step in 3DNCA is to identify full-rank factorisable sub-networks $H^{(i)}$s of G such that $G = \underset{i}{\cup} H^{(i)}$. This problem is solved in section 5.4.2 where two sub-networks $H^{(1)}$ and $H^{(2)}$ of G are identified by applying algorithm 5 to $(D, B(G))$ in [1]. It is shown in section 4.4.2 that $H^{(1)}$ is NCA feasible and corresponding NCA solutions are unique up to a scaling factor $\chi^{(1)}$ (4.21). As mentioned there, $\chi^{(1)}$ is guaranteed to be diagonal in nature. However, its value is dependent on the two pairs of NCA solutions being tested for uniqueness. In this section, values of $(({}^1W^{(1)}), ({}^1S^{(1)}))$, $(({}^2W^{(1)}), ({}^2S^{(1)}))$ and $\chi^{(1)}$ are borrowed from section 4.4.2.

In section 5.4.2, it is shown that the second sub-network $H^{(2)}$ of $B(G)$ in [1] is full-rank factorisable. In the same section, it is shown that $(D^{(2)}, B(H^{(2)}))$ is NCA infeasible. As a result, corresponding NCA solutions $(({}^1W^{(2)}), ({}^1S^{(2)}))$ and $(({}^2W^{(2)}), ({}^2S^{(2)}))$ are found to be non-unique. However, in the same section, it is shown with the help of (5.33) that the two solutions are almost unique up to $\chi^{(2)}$ (5.32). In this section, values of $(({}^1W^{(2)}), ({}^1S^{(2)}))$, $(({}^2W^{(2)}), ({}^2S^{(2)}))$ and $\chi^{(2)}$ are borrowed form section 5.4.2.

On comparing $B(G)$ in Fig. 2.4 with $B(H^{(1)}$ in Fig. 4.2 and $B(H^{(2)}$ in Fig. 5.1, there are 37 rows of D unique to $H^{(1)}$, 44 rows unique to $H^{(2)}$ and 19 rows common to both the sub-networks. Subsets of rows of D, $Y_U^{(1)}$, $Y_U^{(2)}$ and $Y_C^{(1,2)}$ corresponding

respectively to $D_U^{(1)}$, $D_U^{(2)}$ and $D_C^{(1,2)}$ are given as

$$
\begin{aligned}
Y_U^{(1)} = \quad & (1,\ 2,\ 3,\ 5,\ 6,\ 7,\ 8,\ 9,\ 11,\ 12,\ 13,\ 15,16,\ 17,\ 18,\\
& 19,\ 23,\ 26,\ 28,\ 29,\ 34,\ 43,\ 47,\ 48,52,\ 53,\ 58,\ 60,\\
& 62,\ 65,\ 66,\ 67,\ 72,\ 73,\ 74,\ 75,\ 100)\\
\\
Y_U^{(2)} = \quad & (14,\ 20,\ 21,\ 24,\ 30,\ 33,\ 35,\ 37,\ 39,\ 41,\ 42,\ 44,\ 45,\\
& 46,\ 49,\ 54,\ 55,\ 56,\ 59,\ 61,\ 63,\ 64,\ 68,\ 69,\ 70,\ 71,\\
& 77,\ 78,\ 79,\ 81,\ 85,\ 86,\ 88,\ 89,\ 90,\ 91,\ 92,\ 93,\ 94,\\
& 95,\ 96,\ 97,\ 99,\ 99)\\
\\
Y_C^{(1,2)} = \quad & (4,\ 10,\ 22,\ 25,\ 27,\ 31,\ 32,\ 36,\ 38,\ 40,\ 50,\ 51,\ 57,\\
& 76,\ 80,\ 82,\ 83,\ 84,\ 87)
\end{aligned}
\tag{6.73}
$$

3**DNCA solutions** – Given two sets of solutions

$$
\begin{aligned}
&((^1W^{(1)}),(^1S^{(1)})),\ \ ((^1W^{(2)}),(^1S^{(2)}))\\
&((^2W^{(1)}),(^2S^{(1)})),\ \ ((^2W^{(2)}),(^2S^{(2)}))
\end{aligned}
\tag{6.74}
$$

a two-part mixing problem in (6.51) is solved with

$$H^{(A)} = H^{(1)},\ H^{(B)} = H^{(2)}$$

Mixing coefficients corresponding to first set of solutions $((^1W^{(A)}),(^1S^{(A)}))$ and $((^1W^{(B)}),(^1S^{(B)}))$ are found to be

$$(^1\lambda^{(A)}) = 0.4147,\ (^1\lambda^{(B)}) = 0.5853 \tag{6.75}$$

Similarly, mixing coefficients corresponding to second set of solutions $((^2W^{(A)}),$ $(^2S^{(A)}))$ and $((^2W^{(B)}),(^2S^{(B)}))$ are found to be

$$(^2\lambda^{(A)}) = 0.4148,\ (^2\lambda^{(B)}) = 0.5852 \tag{6.76}$$

Diagonal matrices Λ_L and Λ_R can be defined as described in (6.57) where, lengths of the parts of $pd(\Lambda_L)$ and $pd(\Lambda_R)$ are set in accordance with the lengths or row subsets in (6.73). From (6.75) and (6.76)

$$
\begin{aligned}
& (^1\lambda^{(A)}) \approx (^2\lambda^{(A)}), \quad (^1\lambda^{(B)}) \approx (^2\lambda^{(B)})\\
\implies \qquad & \Lambda_L = I_{100}, \qquad\qquad \Lambda_R = I_{16}
\end{aligned}
\tag{6.77}
$$

where, I_{100} and I_{16} are identity matrices of size 100 and 16, respectively. From (6.59) and (6.77), matrix χ can be evaluated as

$$\begin{aligned}\chi &= \begin{pmatrix} \chi^{(A)} & 0 \\ 0 & \chi^{(B)} \end{pmatrix} \Lambda_R^{\frac{1}{2}} \\ &= \begin{pmatrix} \chi^{(A)} & 0 \\ 0 & \chi^{(B)} \end{pmatrix} I_{16} \\ &= \begin{pmatrix} \chi^{(A)} & 0 \\ 0 & \chi^{(B)} \end{pmatrix}\end{aligned}$$

where, $\chi^{(A)} = \chi^{(1)}$ as given in (4.21) and $\chi^{(B)} = \chi^{(2)}$ as given in (5.32). An exception is made by setting the non-zero off-diagonal element in the last row of $\chi^{(2)}$ in (5.32) to zero while evaluating χ in (6.78). This enforces the assumption that $H^{(2)}$ is NCA feasible. This is done for the sake of testing the validity of results developed in this chapter. Principal diagonal of χ is given as

$$\begin{aligned} pd(\chi) &= \begin{pmatrix} pd(\chi^{(1)}) & | & pd(\chi^{(2)}) \end{pmatrix} \\ &= \begin{pmatrix} 1 & -0.84 & -0.83 & 0.97 & 0.99 & 1 & -0.83 & 1.06 & 0.97 & | \\ & -0.84 & 1 & -0.84 & 1 & -1.19 & 1 & 1.02) & & \end{pmatrix} \end{aligned} \tag{6.78}$$

The two sets of solutions in (6.74) can be tested against 3DNCA uniqueness relationship in (6.60). On the other hand, as mentioned at the beginning of this section, NCA solutions are not unique as the corresponding χ in Fig. 3.1 does not satisfy the relationship between two pairs of solutions as described by theorem 1.

Data reconstruction – As mentioned in the previous section, it is made sure that the rows and columns of W and S are restored to their original order. Error $\hat{\Gamma}$ in reconstructing D is calculated as described in (2.15) for one of the NCA and one of the 3DNCA solutions. Absolute values of error in reconstructing D using NCA is shown in Fig. 6.10 while that with 3DNCA is shown in Fig. 6.11.

The two figures are plotted on same scale – 0 to 0.7 – as seen on the colour bars to their left. On comparing the two figures, it can be seen that error with 3DNCA is comparable to that with NCA. It is mentioned at the beginning of this section that 3DNCA is expected to perform equally well as NCA in this case as the underlying dataset D corresponds to a biological system. It is evident from Figs. 6.10 and 6.11 that the set expectation is met with.

As mentioned in section 2.5, the second source signal $S(2,:)$ corresponds to a

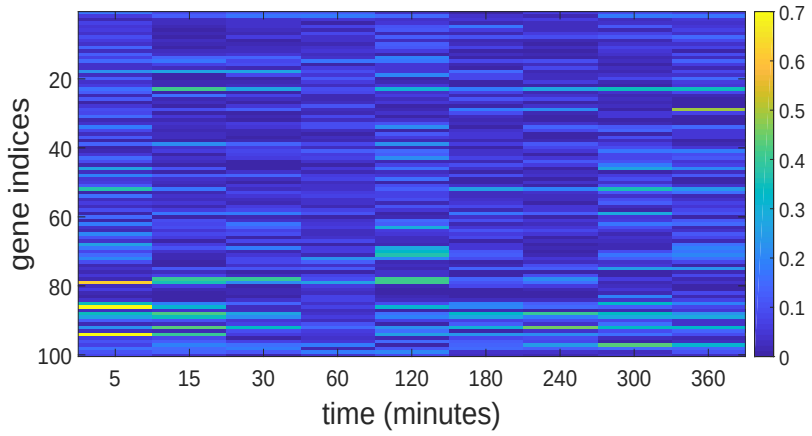

Figure 6.10: Colour map of $|\hat{\Gamma}|$ with NCA corresponding to experimental data D in [1]

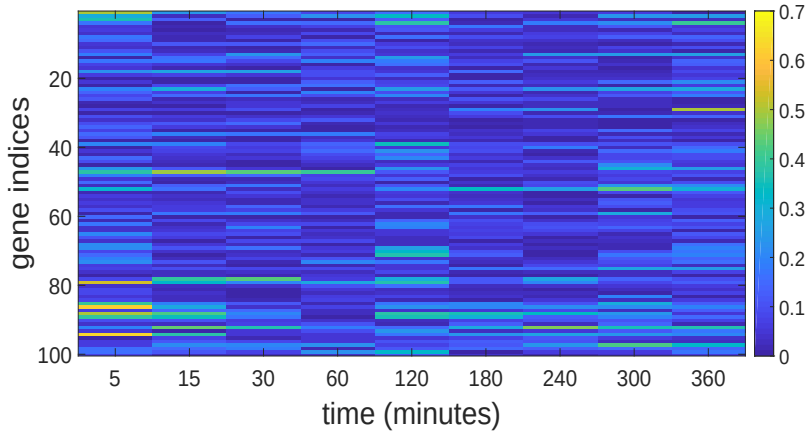

Figure 6.11: Colour map of $|\hat{\Gamma}|$ with 3DNCA corresponding to experimental data D in [1]

protein named CRP in E. coli. Activity of CRP is used in [1] to validate NCA algorithm. CRP activity as estimated by 3DNCA is shown in Fig. 6.12. In [1], CRP activity is expected peak around 5 minutes followed by a monotonous decline in activity over next 4 hours. A key validating point in [1] is that there is a 4 hour time difference between peak and valley of CRP activity. Howvever, the word valley is not defined in the discussions in [1]. In this section, time taken for the value to drop by more than 25% is considered to define valley in CRP activity. 3DNCA estimates an activity of 0.31 units at 5 m nutes and a peak of 0.42 units in CRP activity at 30 minutes. More than 25% drop in activity is estimated over next 210 minutes. CRP activity at 30 and 240 minutes are marked respectively with red and green cross in Fig. 6.12. This means that 3DNCA estimates more than 3.5 hour time difference

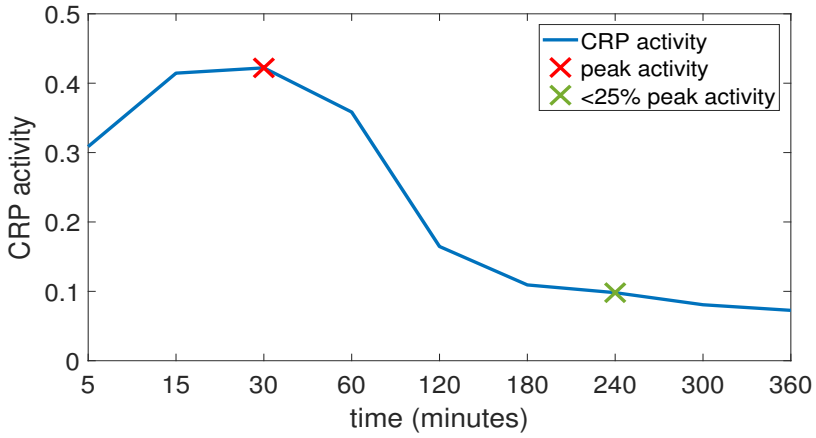

Figure 6.12: 3DNCA estimate of CRP $(S(2,:))$ activity

between peak and valley of CRP activity. As shown in [1], NCA results in similar estimates as well. Thus, for a given experimental dataset, 3DNCA can be used to estimate unique solutions that are in agreement with underlying biology.

Remarks – on comparing the results in Fig. 6.12 with CRP activity estimates presented in [1], 3DNCA seems to be less accurate in terms of time to peak activity and time difference between peak and valley of CRP activity. It is important to note that carbon source switch network in [1] has been studied by the different researchers in the same research group in [34] and [20] under different experimental conditions. Only 9 data points are made available available in [20] whereas CRP estimates in [1] are based on 25 data points. It is not clear which experiment the dataset made available in [20] corresponds to. NCA estimates of CRP activities under two different experimental conditions are compared in Fig. 6, page 138 of [34]. Let solid line in that figure represent experiment– 1 and dashed line represent experiment–2. NCA estimates of CRP in [1] appear to be closer to those corresponding to experiment–1 in [34] whereas 3DNCA estimates of CRP in Fig. 6.12 appear to be closer to CRP estimates corresponding to experiment–2. There is no way to resolve this confusion other than asking for clarification from the authors. As no explanation has not been sought from the authors of [20] yet, it is assumed that the dataset made availble by the authors correspond to experiment–2. Nevertheless, it is shown in this section that 3DNCA achieves the primary goal of this chapter – obtaining unique estimates of W and S.

6.7 Summary

In this chapter, an iterative divide and conquer method is developed to solve a general structure-constrained matrix factorisation problem. This method combines the results developed in previous chapters thereby extending the applicability of a popular bioinformatics tool, NCA. In the process of developing such a method, challenges in scaling the problem to larger systems of equations are identified. Identified issues are addressed by reducing a mixing problem corresponding to a multi-part network to a series of two-part mixing problems. Uniqueness of solutions so obtained are shown to be unique up to two scaling factors. One of the two scaling factors characterises uniqueness of NCA solutions corresponding to the sub-networks whereas the other characterises uniqueness corresponding to the mixing coefficients. Different features 3DNCA solutions are demonstrated using two examples – a random numerical example and an experimental dataset. It is shown that 3DNCA is capable of generating unique estimates of mixing coefficients. Two limitations of 3DNCA are identified – high error in factorising randomly generated data and dependence on NCA feasibility of data subset-sub-network pairs. These two issues are not addressed in this thesis and are counted as potential candidates for future work.

Chapter 7

Conclusions

7.1 Summary of contributions

Novel theoretical results related to NCA [2] have been developed throughout this thesis. A thorough review of NCA has been presented initially where every aspect of NCA theorem is inspected. A minor detail related to binary rank of a network has been ignored for the past 14 years. In this thesis, it is shown that binary rank of a network affects NCA identifiability. A graph theoretical interpretation of TRNs is discussed where three graph operations – reduction, minimum degree ordering and breadth-first matching – are found to be of value in estimating rank of binary matrices. Uniqueness of NCA solutions has been characterised in [2], but a method to test the same is not provided. This aspect of NCA is again ignored in most of the related work. In this thesis, a method based on vectorisation of NCA uniqueness relationship is developed. This method is used to test uniqueness of NCA like solutions computed using novel methods proposed in this thesis. Three major unaddressed challenges related to NCA theory are identified as

- developing accurate NCA feasibility conditions that can be used to check if NCA is applicable to a given dataset-network pair
- developing a method to identify sub-networks that are NCA feasible whenever a given network is NCA infeasible
- developing a method to compute unique solutions regardless of whether a network is NCA feasible or not

Each of these identified research questions have been answered with sufficient technical depth in this thesis. In the process of developing a method to address each of

the identified challenges, several associated hurdles are identified and overcome in this thesis.

The problem of developing NCA feasibility conditions required finding conditions on a priori known dataset and network. A straightforward extension of NCA rank conditions is not possible. Therefore, results from linear algebra and augmented matrix theory are used to define NCA feasibility in terms of given dataset-network pair. Theorem 2 and 3 are developed to separately characterise NCA feasibility of a network alone and that of a dataset-network pair. Binary rank of a network is employed in doing so. This is the first instance in relevant literature where NCA identifiability of a network and a dataset have been separately characterised.

Developing a technique to decompose a NCA infeasible network into multiple NCA feasible sub-networks required exploiting structural properties of a given network. As NCA feasibility conditions are based on binary rank of the given network, bipartite matching based method to find pivots is used. Graph reduction and vertex reordering are used to ensure that all pivots correspond to columns that are linearly independent over binary field. Relationships between size of breadth-first search based matching in reduced ordered graph and its rank over real and binary fields are established. These relationships are used to define NCA feasibility of a network using graph theoretical metrics. The reason to redefine NCA feasibility in the language of graphs is justified with the help of algorithm 5 designed to decompose NCA infeasible networks into NCA feasible sub-networks. As NCA feasibility conditions are different when a dataset is brought into the picture, a method to limit the size of sub-networks is devised. Algorithm 5 takes in a dataset-network pair and returns a set of data subset-sub-network pairs that are full-rank factorisable.

Unique solutions corresponding to a general dataset-network pair is obtained in three steps. First step is to decompose a given dataset-network pair into full-rank factorisable data subset-sub-network pairs. Second step involves computation of NCA solutions corresponding to each sub-network. Last step is to combine sub-network NCA solutions in a convex manner. It is shown that if all data subset-sub-network pairs are NCA feasible, then a convex combination of sub-network NCA solutions is unique up to two scaling factors. This method is shown to generate close to unique and biologically meaningful estimates of TF activities for a particular experimental dataset.

Limitations –

- Chapter 4 – though proposition 1 offers a better characterisation of NCA solution space, cases where there is a need to use the same might be rare. Moti-

vation to look at such special cases is that NCA identifiability conditions are inaccurate. Existence of a trivial solution or lack thereof does not seem to affect 3DNCA. Therefore, pursuing the idea presented in proposition 1 might not be necessary.

- Chapter 5 – it is desirable to identify NCA feasible data subset-sub-network pairs, but algorithm 5 generates only full rank factorisable data subset-sub-network pairs.

- Chapter 6 – uniqueness of 3DNCA solution depends on NCA feasibility of all data subset-sub-network pairs. Therefore, limitations of algorithm 5 limits applicability of 3DNCA. Furthermore, error in reconstructing dataset using 3DNCA is not any better than that with NCA. This issue can possibly be resolved by using some other TRN modelling method to factorise individual sub-networks instead of NCA.

7.2 Future work

In the process of developing the methods summarised in the previous sections, a set of problems are identified as potential candidates for future work.

- A particular kind of NCA infeasibility arising because of differences in real and binary ranks of a network is characterised as a subspace in the associated solution space. A proposition is made to tackle this kind of NCA infeasibility by avoiding the defined subspace. This result is preliminary in nature, future work in this direction will lead to a method that realises the proposition.

- Method to decompose NCA infeasible dataset-network pairs can only guarantee full-rank factorisability of resulting data subset-sub-network pairs, but not their NCA feasibility. An idea to overcome this issue is discussed. NCA feasibility of data subset-sub-network pairs if the limit on network size is chosen in an optimal way. A formal definition of optimal sub-network size and a method to calculate the same can be pursued as a part of the future work.

- Both for a randomly chosen example and an experimental dataset, divide and conquer method generated close to unique solutions despite one of the data subset-sub-network pairs being NCA infeasible. Furthermore, values of mixing factors did not seem to change significantly when two different solutions corresponding to the same dataset were compared. This was true despite one of the sub-network solutions being different compared to the other. This

might be indicative of uniqueness of mixing factors. Uniqueness of mixing factors can be explicitly studied as a part of the future work.

Of the three candidates identified, last one offers a quickly reachable goal. This can be a first choice for further exploration. Apart from the challenges spinning out of the methods developed in this thesis, several other possibilities exists. Different extensions of NCA discussed in the literature review chapter of this thesis can be used to compute sub-network solutions. This will improve the speed of convergence and data reconstruction quality as those are the goals that methods extending NCA are designed to achieve. Divide and conquer method developed in this thesis can be applied to different datasets available in different databses and its performance can be validated. Different dataset correspond to experiments in which different transcription factors were forced to be inactive. Divide and conquer method can be simultaneously applied to multiple datasets. It will be interesting to see if mixing factors so estimated can be used to quantify relative importance of different transcription factors.

Bibliography

[1] K. C. Kao, Y. L. Yang, R. Boscolo, C. Sabatti, V. Rowchowdhury, and J. C. Liao, "Transcriptome-based determination of multiple transcription regulator activities in Escherichia coli by using network component analysis," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101, no. 2, pp. 641–646, January 2004.

[2] J. C. Liao, R. Boscolo, Y. L. Yang, L. M. Tran, C. Sabatti, and V. P. Roychowdhury, "Network component analysis: Reconstruction of regulatory signals in biological systems," *Proceedings of the National Academy of Sciences*, vol. 100, no. 26, pp. 15522–15527, December 2003.

[3] Clarivate Analytics, "Web of Science Databases," *Web of Science*, vol. 20, 2018.

[4] V. Gligorijević and N. Pržulj, "Methods for biological data integration: perspectives and challenges," *Journal of The Royal Society Interface*, vol. 12, no. 112, pp. 1–19, 2015.

[5] A. J. Courey, *Mechanisms in Transcriptional Regulation*, John Wiley & Sons, April 2008.

[6] E. P. van Someren, L. F. A. Wessels, E. Backer, and M. J. T. Reinders, "Genetic network modeling," *Pharmacogenomics*, vol. 3, no. 4, pp. 507–525, 2002.

[7] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science*, vol. 349, no. 6245, pp. 255–260, 2015.

[8] R. De Smet and K. Marchal, "Advantages and limitations of current network inference methods," *Nature Reviews Microbiology*, vol. 8, no. 10, pp. 717–729, August 2010.

[9] J. J. Li, P. J. Bickel, and M. D. Biggin, "System wide analyses have underestimated protein abundances and the importance of transcription in mammals," *PeerJ*, vol. 2, pp. e270, February 2014.

[10] A. Liljas, L. Liljas, J. Piskur, P. Nissen, M. Kjeldgaard, et al., *Textbook of Structural Biology*, World Scientific Publishing Company, 2009.

[11] B. He and K. Tan, "Understanding transcriptional regulatory networks using computational models," *Current Opinion in Genetics & Development*, vol. 37, pp. 101–108, April 2016, Genome architecture and expression.

[12] V. Singh and P.K. Dhar, *Systems and Synthetic Biology*, Springer Netherlands, 2016.

[13] A. Visel, M. J. Blow, Z. Li, T. Zhang, J. A. Akiyama, A. Holt, I. Plajzer-Frick, M. Shoukry, C. Wright, F. Chen, V. Afzal, B. Ren, E. M. Rubin, and L. A. Pennacchio, "ChIP-seq accurately predicts tissue-specific activity of enhancers," *Nature*, vol. 457, no. 7231, pp. 854–858, February 2009.

[14] M. D. Robinson and A. Oshlack, "A scaling normalization method for differential expression analysis of RNA-seq data," *Genome Biology*, vol. 11, no. 3, pp. R25, March 2010.

[15] G. A. Pavlopoulos, P. I. Kontou, A. Pavlopoulou, C. Bouyioukos, E. Markou, and P. G. Bagos, "Bipartite graphs in systems biology and medicine: a survey of methods and applications," *GigaScience*, vol. 7, no. 4, pp. giy014, 2018.

[16] R.J. Wilson, *Introduction to Graph Theory*, Longman, 5th edition, 2010.

[17] B. Haeffele, E. Young, and R. Vidal, "Structured Low-Rank Matrix Factorization: Optimality, Algorithm, and Applications to Image Processing," *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, vol. 5, pp. 4108–4117, 2014.

[18] P. Mongeon and A. Paul-Hus, "The journal coverage of Web of Science and Scopus: a comparative analysis," *Scientometrics*, vol. 106, no. 1, pp. 213–228, January 2016.

[19] M. Kim, B. G. Park, J. Kim, J. Y. Kim, and B. G. Kim, "Exploiting transcriptomic data for metabolic engineering: toward a systematic strain design," *Current Opinion in Biotechnology*, vol. 54, pp. 26–32, December 2018.

[20] S. J. Galbraith, L. M. Tran, and J. C. Liao, "Transcriptome network component analysis with limited microarray data," *Bioinformatics*, vol. 22, no. 15, pp. 1886–1894, June 2006.

[21] B. Hannon and M. Ruth, *Modeling Dynamic Biological Systems*, Springer International Publishing, 2nd edition, 2014.

[22] R. Brockmann, A. Beyer, J. J. Heinisch, and T. Wilhelm, "Posttranscriptional Expression Regulation: What Determines Translation Rates?," *PLOS Computational Biology*, vol. 3, no. 3, pp. 1–9, March 2007.

[23] A. Bordbar, J. M. Monk, Z. A. King, and B. O. Palsson, "Constraint-based models predict metabolic and associated cellular functions," *Nature Reviews Genetics*, vol. 15, no. 2, pp. 107–120, January 2014.

[24] G. Karlebach and R. Shamir, "Modelling and analysis of gene regulatory networks," *Nature Reviews Molecular Cell Biology*, vol. 9, no. 10, pp. 770–780, September 2008.

[25] Q. Zhang, Y. Yu, J. Zhang, and H. Liang, "Using single-index ODEs to study dynamic gene regulatory network," *PLOS ONE*, vol. 13, no. 2, pp. 1–20, February 2018.

[26] M. Enea and G. Lovison, "A penalized approach for the bivariate ordered logistic model with applications to social and medical data," *Statistical Modelling*, p. 1471082X18782063, 2018.

[27] M. Jansen and P. Pfaffelhuber, "Stochastic gene expression with delay," *Journal of Theoretical Biology*, vol. 364, pp. 355–363, January 2015.

[28] S. K. Hahl and A. Kremling, "A Comparison of Deterministic and Stochastic Modeling Approaches for Biochemical Reaction Systems: On Fixed Points, Means, and Modes," *Frontiers in Genetics*, vol. 7, pp. 157, August 2016.

[29] T. Chen, H. L. He, and G. M. Church, "Modeling Gene Expression with Differential Equations," in *Pacific Symposium of Biocomputing*, 1999, pp. 29–40.

[30] Q. P. Ha and H. Trinh, "State and input simultaneous estimation for a class of nonlinear systems," *Automatica*, vol. 40, no. 10, pp. 1779–1785, October 2004.

[31] J. Qin, M. J. Li, P. Wang, M. Q. Zhang, and J. Wang, "ChIP-Array: combinatory analysis of ChIP-seq/chip and microarray gene expression data to discover direct/indirect targets of a transcription factor," *Nucleic Acids Research*, vol. 39, no. 2, pp. W430–W436, May 2011.

[32] V. Hatzimanikatis and J. E. Bailey, "Mca Has More to Say," *Journal of Theoretical Biology*, vol. 182, no. 3, pp. 233–242, October 1996.

[33] E. O. Voit and I. C. Chou, "Parameter Estimation in Canonical Biological Systems Models," *International Journal of Systems and Synthetic Biology*, vol. 1, pp. 1–19, June 2010.

[34] L. M. Tran, M. P. Brynildsen, K. C. Kao, J. K. Suen, and J. C. Liao, "gNCA: A framework for determining transcription factor activity based on transcriptome: identifiability and numerical implementation," *Metabolic Engineering*, vol. 7, no. 2, pp. 128–141, 2005.

[35] E. O. Voit, "Biochemical Systems Theory: A Review," *ISRN Biomathematics*, vol. 2013, 2013.

[36] M. Heinonen, O. Guipaud, F. Milliat, V. Buard, B. Micheau, G. Tarlet, M. Benderitter, F. Zehraoui, and F. dAlcheBuc, "Detecting time periods of differential gene expression using Gaussian processes: an application to endothelial cells exposed to radiotherapy dose fraction," *Bioinformatics*, vol. 31, no. 5, pp. 728–735, 2015.

[37] N. Lu and H. Miao, "Structure constrained nonnegative matrix factorization for pattern clustering and classification," *Neurocomputing*, vol. 171, pp. 400–411, January 2016.

[38] C.J. Stam and J.C. Reijneveld, "Graph theoretical analysis of complex networks in the brain," *Nonlinear Biomedical Physics*, vol. 1, no. 3, pp. 1–19, July 2007.

[39] P. Cloutier, R. Al-Khoury, M. Lavallée-Adam, D. Faubert, H. Jiang, C. Poitras, A. Bouchard, D. Forget, M. Blanchette, and B. Coulombe, "High-resolution mapping of the protein interaction network for the human transcription machinery and affinity purification of RNA polymerase II-associated complexes," *Elsevier Methods*, vol. 48, no. 4, pp. 381–386, August 2009.

[40] Q. Shi, C. Zhang, W. Guo, T. Zeng, L. Lu, Z. Jiang, Z. Wang, J. Liu, and L. Chen, "Local network component analysis for quantifying transcription factor activities," *Elsevier Methods*, vol. 124, no. C, pp. 25–35, July 2017.

[41] J. N. Franklin, *Matrix Theory*, Dover, 2012.

[42] R. Albert, "Network Inference, Analysis, and Modeling in Systems Biology," *The Plant Cell*, vol. 19, pp. 3327–3338, 2007.

[43] F. Markowetz and R. Spang, "Inferring cellular networks - a review," *BMC Bioinformatics*, vol. 8, pp. 1–17, 2007.

[44] W. Lee and W. Tzou, "Computational methods for discovering gene networks from expression data," *Briefings in Bioinformatics*, vol. 10, pp. 408–423, 2009.

[45] T. Äijö and R. Bonneau, "Biophysically Motivated Regulatory Network Inference: Progress and Prospects," *Human Heredity*, vol. 81, no. 2, pp. 62–77, 2016.

[46] C. Meng, O. A. Zeleznik, G. G. Thallinger, B. Kuster, A. M. Gholami, and A. C. Culhane, "Dimension reduction techniques for the integrative analysis of multi-omics data," *Briefings in Bioinformatics*, vol. 17, no. 4, pp. 628–641, July 2016.

[47] A. L. Boulesteix and K. Strimmer, "Partial least squares: a versatile tool for the analysis of high-dimensional genomic data," *Briefings in Bioinformatics*, vol. 8, no. 1, pp. 32–44, January 2006.

[48] Q. Yan, J. Ye, and X. Shen, "Simultaneous Pursuit of Sparseness and Rank Structures for Matrix Decomposition," *The Journal of Machine Learning Research*, vol. 16, no. 1, pp. 47–75, 2015.

[49] R. Silva-Rocha and V. de Lorenzo, "Mining logic gates in prokaryotic transcriptional regulation networks," *FEBS Letters*, vol. 582, no. 8, pp. 1237–1244, 2008.

[50] Y. Ye, L. Gao, and S. Zhang, "Integrative analysis of transcription factor combinatorial interactions using a bayesian tensor factorization approach," *Frontiers in Genetics*, vol. 8, pp. 140, 2017.

[51] L. Zhu, W. Guo, S. Deng, and D. Huang, "ChIP-PIT: Enhancing the Analysis of ChIP-Seq Data Using Convex-Relaxed Pair-Wise Interaction Tensor

Decomposition," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 13, no. 1, pp. 55–63, January 2016.

[52] X. Yu, D. Hu, and J. Xu, *Blind Source Separation: Theory and Applications*, John Wiley & Sons, January 2014.

[53] A. Cichocki, R. Zdunek, A. H. Phan, and S. Amari, *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-Way Data Analysis and Blind Source Separation*, John Wiley & Sons, October 2009.

[54] A. Hyvärinen, "Independent component analysis: recent advances," *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 371, no. 1984, pp. 1–19, 2013.

[55] Y. X. Wang and Y. J. Zhang, "Nonnegative Matrix Factorization: A Comprehensive Review," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 6, pp. 1336–1353, 2013.

[56] K. Yu, K. Yang, and Y. Bai, "Estimation of modal parameters using the sparse component analysis based underdetermined blind source separation," *Mechanical Systems and Signal Processing*, vol. 45, no. 2, pp. 302–316, April 2014.

[57] K. Devarajan, "Nonnegative Matrix Factorization: An Analytical and Interpretive Tool in Computational Biology," *PLOS Computational Biology*, vol. 4, no. 7, pp. 1–12, July 2008.

[58] M. F. Ochs and E. J. Fertig, "Matrix Factorization for Transcriptional Regulatory Network Inference," in *Computational Intelligence in Bioinformatics and Computational Biology (CIBCB), 2012 IEEE Symposium on*. IEEE, May 2012, pp. 387–396.

[59] S. Weiss, Z. Z. Xu, S. Peddada, A. Amir, K. Bittinger, A. Gonzalez, C. Lozupone, J. R. Zaneveld, Y. Vázquez-Baeza, A. Birmingham, E. R. Hyde, and R. Knight, "Normalization and microbial differential abundance strategies depend upon data characteristics," *BMC Microbiome*, vol. 5, no. 1, pp. 27, March 2017.

[60] R. Boscolo, C. Sabatti, J. C. Liao, and V. P. Roychowdhury, "A generalized framework for network component analysis," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 2, no. 4, pp. 289–301, 2005.

[61] G. Sanguinetti, M. Rattray, and N. D. Lawrence, "A probabilistic dynamical model for quantitative inference of the regulatory mechanism of transcription," *Bioinformatics*, vol. 22, no. 14, pp. 1753–1759, April 2006.

[62] G. Sanguinetti, N. D. Lawrence, and M. Rattray, "Probabilistic inference of transcription factor concentrations and gene-specific regulatory activities," *Bioinformatics*, vol. 22, no. 22, pp. 2775–2781, November 2006.

[63] H. M. S. Asif, M. D. Rolfe, J. Green, N. D. Lawrence, M. Rattray, and G. Sanguinetti, "TFInfer: a tool for probabilistic inference of transcription factor activities," *Bioinformatics*, vol. 26, no. 20, pp. 2635–2636, October 2010.

[64] T. Schacht, M. Oswald, R. Eils, S. B. Eichmüller, and R. König, "Estimating the activity of transcription factors by the effect on their target genes," *Bioinformatics*, vol. 30, no. 17, pp. i401–i407, September 2014.

[65] C. Wang, J. Xuan, I. Shih, R. Clarke, and Y. Wang, "Regulatory component analysis: A semi-blind extraction approach to infer gene regulatory networks with imperfect biological knowledge," *Signal Processing*, vol. 92, no. 8, pp. 1902–1915, August 2012.

[66] N. Jacklin, Z. Ding, W. Chen, and C. Chang, "Noniterative Convex Optimization Methods for Network Component Analysis," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 9, no. 5, pp. 1472–1481, 2012.

[67] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.

[68] F. Gao, B. C. Foat, and H. J. Bussemaker, "Defining transcriptional networks through integrative modeling of mRNA expression and transcription factor binding data," *BMC Bioinformatics*, vol. 5, no. 1, pp. 31, March 2004.

[69] I. Nachman, A. Regev, and N. Friedman, "Inferring quantitative models of regulatory networks from expression data," *Bioinformatics*, vol. 20, no. suppl_1, pp. i248–i256, August 2004.

[70] H.M. Lodhi and S.H. Muggleton, *Elements of Computational Systems Biology*, Wiley, 2010.

[71] D.A. Turkington, *Generalized Vectorization, Cross-Products, and Matrix Calculus*, Cambridge University Press, 2013.

[72] B. Efron, *The Jackknife, the Bootstrap, and Other Resampling Plans*, vol. 38, SIAM CBMS-NSF Regional Conference Series in Applied Mathematics, 1982.

[73] X. Wang, M. Alshawaqfeh, X. Dang, B. Wajid, A. Noor, M. Qaraqe, and E. Serpedin, "An Overview of NCA-Based Algorithms for Transcriptional Regulatory Network Inference," *Microarrays*, vol. 4, no. 4, pp. 596–617, November 2015.

[74] N. D. Jayavelu and N. Bar, "ISNCA: A new iterative approach for constrained matrix factorization methods," *Journal of Process Control*, vol. 60, pp. 24–33, December 2017.

[75] A. Noor, A. Ahmad, and E. Serpedin, "Sparse NCA: Sparse Network Component Analysis for Recovering Transcription Factor Activities with Incomplete Prior Information," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 15, no. 2, pp. 387–395, March 2018.

[76] C. Chang, Z. Ding, Y. S. Hung, and C. W. Fung, "Fast network component analysis (fastnca) for gene regulatory network reconstruction from microarray data," *Bioinformatics*, vol. 24, pp. 1349–1358, 2008.

[77] A. Noor, A. Ahmad, E. Serpedin, M. Nounou, and H. Nounou, "Robnca: robust network component analysis for recovering transcription factor activities," *Bioinformatics*, vol. 29, pp. 2410–2418, 2013.

[78] N. D. Jayavelu, L. S. Aasgaard, and N. Bar, "Iterative sub-network component analysis enables reconstruction of large scale genetic networks," *BMC Bioinformatics*, vol. 16, no. 1, pp. 1–13, November 2015.

[79] H. T. Wai, A. Scaglione, B. Barzel, and A. Leshem, "NETWORK INFERENCE FROM COMPLEX SYSTEMS STEADY STATES OBSERVATIONS: THEORY AND METHODS," in *2018 IEEE Data Science Workshop (DSW)*. IEEE, June 2018, pp. 155–159.

[80] J. Ernst, G. J. Nau, and Z. Bar-Joseph, "Clustering short time series gene expression data," *Bioinformatics*, vol. 21, pp. i159–i168, 2005.

[81] E. Sefer, M. Kleyman, and Z. Bar-Joseph, "Tradeoffs between Dense and Replicate Sampling Strategies for High-Throughput Time Series Experiments," *Cell systems*, vol. 3, no. 1, pp. 35–42, 2016.

[82] E. Fritzilas, M. Milanič, J. me Monnot, and Y. A. Rios-Solis, "Resilience and optimization of identifiable bipartite graphs," *Discrete Applied Mathematics*, vol. 161, no. 4–5, pp. 593–603, March 2013.

[83] S. Salin, M. Manguoğlu, and H. M. Aktulga, "Learning the Domain of Sparse Matrices," in *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, December 2016, pp. 800–805.

[84] Y. Saad, *Iterative Methods for Sparse Linear Systems*, vol. 82 of *Other Titles in Applied Mathematics*, Society for Industrial and Applied Mathematics (SIAM), 2003.

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

[86] K. Mulmuley, U. V. Vazirani, and V. V. Vazirani, "Matching is as easy as matrix inversion," *Combinatorica*, vol. 7, no. 1, pp. 105–113, March 1987.

[87] P. Foggia, G. Percannella, and M. Vento, "GRAPH MATCHING AND LEARNING IN PATTERN RECOGNITION IN THE LAST 10 YEARS," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 28, no. 01, pp. 1450001, 2014.

[88] F. Serratosa, "Fast computation of Bipartite graph matching," *Pattern Recognition Letters*, vol. 45, pp. 244–250, August 2014.

[89] F. Serratosa, "Speeding up Fast Bipartite Graph Matching Through a New Cost Matrix," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 29, no. 02, pp. 1550010, 2015.

[90] X. Yan, H. Cheng, J. Han, and P. S. Yu, "Mining significant graph patterns by leap search," in *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*. ACM, December 2008, pp. 433–444.

[91] M. M. Babu, N. M. Luscombe, L. Aravind, M. Gerstein, and S. A. Teichmann, "Structure and evolution of transcriptional regulatory networks," *Current Opinion in Structural Biology*, vol. 14, no. 3, pp. 283–291, June 2004.

[92] D. M. Busiello, S. Suweis, J. Hidalgo, and A. Maritan, "Explorability and the origin of network sparsity in living systems," *Scientific Reports*, vol. 7, no. 1, pp. 12323, September 2017.

[93] A. Sandryhaila and J. M. F. Moura, "Big Data Analysis with Signal Processing on Graphs: Representation and processing of massive data sets with irregular structure," *IEEE Signal Processing Magazine*, vol. 31, no. 5, pp. 80–90, September 2014.

[94] T. A. Davis and Y. Hu, "The University of Florida sparse matrix collection," *ACM Transactions on Mathematical Software (TOMS)*, vol. 38, no. 1, pp. 1, 2011.

[95] D. E. Cameron, C. J. Bashor, and J. J. Collins, "A brief history of synthetic biology," *Nature Reviews Microbiology*, vol. 12, no. 5, pp. 381–390, May 2014.

[96] P. Sah, L. O. Singh, A. Clauset, and S. Bansal, "Exploring community structure in biological networks with random graphs," *BMC Bioinformatics*, vol. 15, no. 1, pp. 220, June 2014.

[97] C. Espinosa-Soto, "On the role of sparseness in the evolution of modularity in gene regulatory networks," *PLOS Computational Biology*, vol. 14, no. 5, pp. 1–24, May 2018.

[98] J. Ballani and D. Kressner, "Matrices with Hierarchical Low-Rank Structures," in *Exploiting Hidden Structure in Matrix Computations: Algorithms and Applications*, pp. 161–209. Springer, February 2016.

[99] D. Silva, M. Velazco, and A. Oliveira, "Influence of matrix reordering on the performance of iterative methods for solving linear systems arising from interior point methods for linear programming," *Mathematical Methods of Operations Research*, vol. 85, no. 1, pp. 97–112, 2017.

[100] I. M. Keseler, A. Mackie, A. Santos-Zavaleta, R. Billington, C. Bonavides-Martínez, R. Caspi, C. Fulcher, S. Gama-Castro, A. Kothari, M. Krummenacker, M. Latendresse, L. Muñiz-Rascado, Q. Ong, S. Paley, M. Peralta-Gil, P. Subhraveti, D. A. Veláquez-Ramírez, D. Weaver, J. Collado-Vides, I. Paulsen, and P. D. Karp, "The EcoCyc database: reflecting new knowledge about Escherichia coli K-12," *Nucleic Acids Research*, vol. 45, no. D1, pp. D543–D550, November 2016.

[101] A. M. Huerta, H. Salgado, D. Thieffry, and J. Collado-Vides, "Regulondb: a database on transcriptional regulation in Escherichia coli," *Nucleic Acids Research*, vol. 26, no. 1, pp. 55–59, January 1998.

[102] J. J. Faith, B. Hayete, J. T. Thaden, I. Mogno, J. Wierzbowski, G. Cottarel, S. Kasif, J. J. Collins, and T. S. Gardner, "Large-Scale Mapping and Validation of Escherichia coli Transcriptional Regulation from a Compendium of Expression Profiles," *PLOS Biology*, vol. 5, no. 1, pp. 1–13, January 2007.

[103] R. Lidl and H. Niederreiter, *Introduction to Finite Fields and their Applications*, Cambridge University Press, 1994.

[104] E. Ben-Sasson, S. Lovett, and N. Ron-Zewi, "An Additive Combinatorics Approach Relating Rank to Communication Complexity," in *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*, October 2012, pp. 177–186.

[105] S. Lang, *Introduction to Linear Algebra*, Springer Science & Business Media, 2012.

[106] R. H. Byrd, M. E. Hribar, and J. Nocedal, "An Interior Point Algorithm for Large-Scale Nonlinear Programming," *SIAM Journal on Optimization*, vol. 9, no. 4, pp. 877–900, 1999.

[107] S. Puntanen, G. P. H. Styan, and J. Isotalo, *Rank of the Partitioned Matrix and the Matrix Product*, pp. 121–144, Springer Berlin Heidelberg, 2011.

[108] D. Jungnickel, *Graphs, Networks and Algorithms*, vol. 5, Springer Science & Business Media, 2007.

[109] H. Li, L. Su, and H. Sun, "On bipartite graphs which attain minimum rank among bipartite graphs with given diameter," *Electronic Journal of Linear Algebra*, vol. 23, pp. 1–14, 2012.