Reconstructing Biological Systems Incorporating Multi-Source Biological Data via Data Assimilation Techniques

データ同化手法を用いた多種生体内データの統合による生体内システム再構築の研究

Takanori Hasegawa
長谷川 嵩矩

Dissertation submitted to Graduate School of Informatics
Kyoto University
Thesis Supervisor: Tatsuya Akutsu

Abstract

More than twenty thousands genes are coded on human's 3 billion base pairs of deoxyribonucleic acid (DNA) sequences. The functions of these genes are mutually regulated by their products, *e.g.*, proteins and ribonucleic acid (RNA), and other factors, and the elucidation of the entire structure of these biological systems has broad utility in applications of drug development, medical treatment and preventive medicine. Therefore, this has been one of the major challenges in the field of systems biology. Especially, in recent technological advances in biotechnology, microarray technologies have been greatly contributed to systems biology researches since we can infer the regulatory relationships and biological processes by comparing the expression levels of RNAs and proteins in several conditions. In the field of systems biology, there exist two types of data obtained by microarrays; thus, time-course data and static (non time-course) data. The time-course data is a set of expression data measured at designed time points after drug stimulations, heat shocks and so on. The static data is measured at the steady state condition in knock-down cells, drug stimulated cells and so on. In this thesis, we handle the time-course microarray data for the elucidation of biological systems.

In using such time-course observation data, there are two major approaches, *i.e.*, simulation-based and statistical approaches. The simulation-based approach uses biologically validated mathematical equations, *e.g.*, ordinary differential equation (ODE) and stochastic differential equation (SDE), to represent the complex dynamics of biological systems and has constructed several biological simulation models including gene regulatory networks and metabolic pathways by combining the accumulated knowledge of biomolecular reactions. Here, the purposes of these constructions are to understand the dynamic behavior of biological systems, to infer the relationships among genes and to predict the changes of their dynamics when adding or removing some biomolecules. However, we have two major problems in this approach; (i) we cannot apply this approach when most part of target systems are unknown due to computational difficulties in evaluating a lot of candidate models and (ii) we cannot handle systems consisting of ten or more genes due to a computational problem known as the curse of dimensionality. In contrast, the statistical approach uses highly abstract models with assumptions such as linearity and simply describes biological systems to infer relationships among several hundreds of genes from the data. In this approach, many computational methods, *e.g.*, dynamic Bayesian networks, state space models (SSM) and vector auto-regressive (VAR) models, have been proposed. Recently, these approaches have further incorporated several biological findings, *e.g.*, transcription factor (TF)

information and protein-protein interaction, to infer biologically validated results. However, the high abstraction generates false regulations that are not permitted biologically. Thus, there is a trade-off relationship between accuracy and computational ease.

In this thesis, we propose a set of analysis procedures for biological systems using time-course observation data in the context of genomic data assimilation, which tries to collaborate simulation-based and statistical approaches to reveal biological systems. Depending on the accumulation of biological knowledge and the required accuracy of inference results, we attempt to infer biological systems that can best predict observation data based on (i) a linear SSM that covers basic processes of gene regulatory systems as represented in complex nonlinear differential equations, (ii) a simple nonlinear SSM that is constructed by extending the linear SSM, and (iii) a complex nonlinear differential equations. For each case, we propose a novel method to estimate the values of the parameters under several biological constraints and infer regulatory relationships among genes and biomolecules to be consistent with the data.

At first, we consider the case of dealing with several tens of genes of which regulatory relationships are partially known. Then, we propose a novel method for the inference of gene regulatory networks (GRNs) using a newly established state space representation of a vector auto-regressive model with $L1$ regularization. In contrast to the previous linear VAR models and SSMs, the proposed model can represent basic components of gene regulatory systems and the proposed method can infer the regulatory structure with a sparse constraint. Furthermore, the method is capable of incorporating various types of existing biological knowledge, *e.g.*, drug kinetics and literature-recorded pathways. For an application example, we infer corticosteroid pharmacogenomic pathways consisting of 40 genes in rat skeletal muscle using time-course microarray data, TF information, corticosteroid pharmacodynamics and literature-derived pathways.

Next, we consider the case that we have GRNs that are derived from literature or inferred by some computational methods. In this case, we try to improve and extend the networks in which parts of regulations can be reliable based on a state space representation of a simple nonlinear model termed the combinatorial transcription model. In contrast to the previous approaches, *e.g.*, nonlinear VAR models, the proposed model can handle non-equally spaced time-course data and separately deal with system and observation noises. For the inference of GRNs and the estimation of the parameter values in the nonlinear SSM, we propose an algorithm to efficiently explore candidate networks utilizing the unscented Kalman filter (UKF). Under this algorithm, UKF calculates approximate conditional distributions of the hidden state variables to efficiently estimate the parameter values maximizing prediction ability for observational data by the EM-algorithm. Although UKF can efficiently estimate the parameter values and explore model space, it does not fully satisfy the requirements of estimating the optimal parameter values; thus, the first four moments are required to obtain the optimal ones. Therefore, we further develop a novel method termed a higher moment ensemble particle filter that can retain the first two and the third and the fourth central moments of the conditional distributions of the hidden state variables through prediction, filtering and smoothing steps. Starting from the original model, which is derived from literature or inferred by some computational methods,

the proposed algorithm can sequentially evaluate candidate models, which are generated by partially changing the current best model, to find the model that can best predict the data. For an application example, we also use corticosteroid pharmacogenomic pathways in rat skeletal muscle.

Finally, we consider the case of handling relatively small pathways that are described by differential equations. Utilizing corticosteroid pharmacogenomic pathways in rat liver cells as an application example, we first propose a computational approach to comprehensively screen candidate pathways for gene expression profiles. In this approach, a systematic model generation strategy is developed; candidate pharmacogenomic pathways are automatically generated from some prototype pathways constructed from existing literature. The parameters values in the nonlinear differential equations within a state space model are estimated based on time-course gene expression data by the particle filter. The candidate pathways are also ranked based on their prediction power measured by Bayesian information criterion. However, this procedure is computationally costly and can not handle a large number of candidates that are required to find models whose simulation results are highly consistent with the data. To overcome the problem, we focus on the fact that the qualitative dynamics of candidate pathways are highly similar if they share a certain amount of regulatory structures. This indicates that better fitting candidates tend to share basic regulatory structures of the best fitting candidate, which can best predict the data among candidates. Thus, instead of evaluating all candidates, we propose an efficient exploration method that can selectively and sequentially evaluate candidates based on the similarity of their regulatory structures.

論文概要

ヒトの細胞内に存在する三十億塩基対もの DeoxyriboNucleic Acid (DNA) には, 二万を超える遺伝子が記述されている. これらの遺伝子の機能はそれぞれの遺伝子の生産物, 即ちタンパク質や RiboNucleic Acid (RNA), 若しくはその他の要因によって相互に制御されており, この生体内システムの全体像を明らかにすることは創薬や医学療法, 予防医学への応用に於ける幅広い実用可能性を秘めている為, システム生物学に於ける最も中心的な目標の一つとなっている. この目標に対し, 計算機的若しくは統計的手法を用いて幾つもの状況下に於ける RNA やタンパク質の発現値を比較することで遺伝子の制御関係や生体内プロセスを推定することが出来る為, マイクロアレイ技術の発達は近年の生命工学の進展の中でも, 特にシステム生物学に貢献する技術の一つだと考えられる. ここで, システム生物学領域に於いて使われるこれらマイクロアレイによって得られるデータは, 時系列データと非時系列データの二つに大別することが出来る. 時系列データは, 例えば薬物投与や熱ショックを与えた細胞内の遺伝子の発現値を, ある適当な時間間隔で測定した一連のデータセットのことを示し, 非時系列データは, 例えばノックアウト遺伝子細胞や薬物刺激を与えた細胞の定常状態に於ける遺伝子の発現値を測定したデータセットなどのことを示す. 本論文では, 時系列データを用いた細胞内システムの解析を取り扱う.

このような時系列データを解析する手法は大きく, シミュレーションモデルベース手法と統計学的手法の二つに分けることが出来る. シミュレーションモデルベースの手法に於いて, 生体内システムの複雑な動的振る舞いは, 生物学的に妥当であると検証が成されている常微分方程式や確率微分方程式によって再現することが出来, 文献化されている生体内分子の反応を組み合わせることで, これまでに多くの遺伝子制御ネットワークや代謝ネットワークなどが構築されてきた. ここで, これらのモデルは生体内システムの動的な振る舞いの理解や遺伝子の制御関係の推定, 生体内分子を付加若しくは取り除いたときの生体内システムの振る舞いの変化を予測するために構築される. しかしながら, この手法は二つの大きな問題を抱えている. 一つ目として, この手法は計算負荷が非常に大きい為, 生体内システムの大部分が未知で候補となる多数の生体内システムを評価しなくてはならない場合に適用不可能であることが挙げられる. 二つ目としては, 次元の呪いとして知られる数学的制約によって, 十を超える生体内分子を含むようなシステムの評価を行うことが著しく困難であることが挙げられる. 一方で, 統計学的手法に於いては, 例えば線形性などの仮定を置くことで複雑な生体内システムを抽象化した統計モデルなどを用いることにより, 観測データから百を超える遺伝子間の制御関係を推定することが可能である. このような手法としては, 動的ベイジアンネットワークや状態空間モデル, ベクトル回帰モデルなどの手法が提案されてる. 近年では, 転写因子やタンパク質相互作用などの生物学的知見を組み合わせることで, 推定精度の向上を達成する手法も提案されている. しかしながら, 抽象化されたモデルを用いた推定結果はしばし

ば生物学的な知見と相反する結果を多分に含むことが分かっている. 即ち, 精度と計算上の取り扱いやすさの間にはトレードオフの関係性が成立しているのである.

本論文では, シミュレーションベース手法と統計学的手法の融合によって生体内システムの解明を目指す, ゲノムデータ同化という手法を用いて, 時系列データから生体内システムを解析する一連の手法を提案する. ここでは, 既存のデータ同化手法が取り扱っていない, 対象とする生体内システムに関する知見の多さと必要な精度に関する幾つかの条件に対応するモデルを提案し, そのモデルに於いて生体内観測データを最も良く予測し得る生体内システムを推定する. 提案したモデルは, (i) 生体内システムを叙述する非線形微分方程式に組み込まれている遺伝子制御システムの基本要素を含む線形モデルを用いた状態空間モデル, (ii) (i) のモデルを拡張した単純な非線形状態空間モデル, (iii) 複雑なシステムを表現可能な非線形の微分方程式を組み込んだ状態空間モデルの三つのモデルである. それぞれのケースに対して, 幾つかの生物学的な制約の元で, モデルのパラメータを推定し, 生体内観測データに一致するような制御関係を持つモデルを推定する手法を提案する.

まず初めに, 部分的に制御関係が判明している数十の遺伝子を対象とするケースを考える. このようなケースに対して, 生体内システムの基本的な要素を表し得る新しいベクトル回帰モデルの状態空間表現を用い, $L1$ 制約下で遺伝子の制御構造を推定する手法を提案する. このモデルは一般的なベクトル回帰モデルや状態空間モデルと異なり, 遺伝子制御システムの基本的な要素を網羅した上で, 疎構造の条件下に於いて遺伝子間の制御関係を推定することが可能である. 加えて, 薬物動態や文献由来制御構造などを取り組むことが出来るように拡張を施する. 適用例として, ラット骨筋細胞に対してコルチコステロイド刺激を加えたときの mRNA の時系列観測データと転写因子の情報, コルチコステロイド薬物動態, 文献由来制御関係の情報を組み合わせ, 四十のコルチコステロイド関連遺伝子とコルチコステロイドの間の制御構造を推定する.

次に, 文献由来や, 何らかの遺伝子制御構造の推定手法を用いて得られた遺伝子制御構造を有しているケースを考える. このケースに於いて, これら一部に信頼性の高い制御関係を含むような遺伝子制御構造を, combinatorial transcription model という非線形システムの状態空間表現を用いて, 修正若しくは拡張する. 非線形ベクトル回帰モデルなどの従来手法と異なり, 提案手法は非等間隔時点に於いて観測された時系列データを扱うことが出来, 更にシステムノイズと観測ノイズを別々に扱うことが出来る. 遺伝子間制御構造とパラメータの推定に関しては unscented Kalman filter という手法を適用することで, 候補モデルを効率的に探索するアルゴリズムを開発した. このアルゴリズムでは, ガウス分布として近似された隠れ変数の確率分布を逐次的に計算し, 観測データを最大限予測し得るようなパラメータ値を EM アルゴリズムを用いて推定する. unscented Kalman filter を用いたアルゴリズムでは効率的にパラメータ値を推定し, モデル空間を探索することが可能であるが, 提案モデルの観測データに対する最適なパラメータ値を推定する為には隠れ変数の条件付き分布の一次から四次までのモーメントが用いられる為, この近似を用いて選ばれたモデルと真のモデルに差異が発生する可能性がある. そこで我々は, 予測と濾波, 平滑化を通して隠れ変数の条件付き分布の一次と二次のモーメント, 更に三次と四次の中心モーメントを維持することが可能である, higher moment ensemble particle filter という手法を開発した. 文献情報や他の手法によって推定されたモデルを元にし, 提案手法は現在の最適モデルを部分的に修正することで作成した候補モデルを逐次的に評価していくことによって, 観測データを最も良く予測し得るモデルを探索する. 適用例として, 同様のラット骨筋細胞に対するコルチコステロイド刺激経路の

データを用いる.

最後に, 微分方程式で叙述される比較的小さい生体内システムを取り扱うことを考える. ここではラット肝細胞に於けるコルチコステロイド薬理遺伝学パスウェイを適用例として取り扱い, ラット肝細胞に於けるコルチコステロイド刺激に対応する遺伝子発現データを予測し得るような候補パスウェイを網羅的にスクリーニングする方法論を提案する. この手法に於いてはまず, 文献由来のプロトタイプパスウェイから自動的に候補モデルを作成するモデル構築戦略を考える. 次に, 非線形微分方程式を持つ状態空間モデルのパラメータ推定には粒子フィルタアルゴリズムを適用し, 候補モデルの観測データに対する予測性能の評価にはベイズ情報量規準を利用する. しかしながら候補モデルの網羅的評価は計算機コストが非常に大きく, 複雑な観測データを高い精度で予測し得るようなモデルを含むような多数の候補モデルセットを扱うことは出来ない. この問題に対処するため, 我々は, 制御構造の一部若しくは大部分を共有しているようなモデル間に於いて, それらのパスウェイモデルの動的振る舞いが非常に似通っていという事実に注目する. 即ち, 観測データに対する高い予測精度を持つモデルの制御構造は, それを最も良く予測し得るモデルの制御構造の一部を共有し得るという傾向を用いる. このような, 観測データの予測に関する制御構造の類似関係を利用することで, 全ての候補モデルを評価すること無しに, 効率的かつ選択的に候補モデルを評価可能な手法を提案する.

Acknowledgments

First of all, I am deeply grateful to my supervisor, Professor Dr. Tatsuya Akutsu. He kindly gave me an opportunity to work and environments to study this thesis, a lot of guidance and advices to my research direction. I would also like to thank other referees of this thesis for valuable discussion and comments: Professor Dr. Hisashi Kashima and Professor Dr. Shin Ishii.

I would also like to thank Associate Professor Dr. Seiya Imoto, Lecturer Dr. Rui Yamaguchi and Professor Dr. Satoru Miyano at The University of Tokyo for giving me a lot of valuable comments, advices and research directions to the developments of methodologies and algorithms for gene regulatory networks inference. My special thanks goes to Professor Dr. Masao Nagasaki at Tohoku University for invaluable discussion and comments to the construction of biological models. My special thanks also goes to Assistant Professor Dr. Atsushi Niida at The University of Tokyo, Associate Professor Dr. Teppei Shimamura at Nagoya University and Professor M.D. Dr. Koshi Mimori at Kyushu University Beppu Hospital who gave me a lot of advices about genome analysis, and Professor M.D. Dr. Masato Inoue at Waseda University who introduced me to research design and statistics.

I am grateful for giving me a lot of advices not only about study but also about my daily life from Assistant Professor Dr. Morihiro Hayashida, Tomoya Mori, Jaewook Hwang, Assistant Professor Dr. Mayumi Kamada, Dr. Yang Zhao, Peiying Ruan and Qiu Yushan. I would also like to give a huge thanks to each and every one of staff members of Akutsu laboratory, Assistant Professor Dr. Takeyuki Tamura, Noriko Kishida, Tamami Fukushiro and all of members including former members as well. I would also like to express my thanks to all of the people in Bioinformatics Center.

Finally, I express deep gratitude to my family and all of my friends to their support and encouragement. I could not have finished and compiled this thesis without all the people around me.

Publication Notes

Chapter 3 is based on the paper "Inference of Gene Regulatory Networks Incorporating Multi-Source Biological Knowledge via a State Space Model with L1 Regularization" that is published in *PLoS ONE*.

Chapter 4 is based on the paper "An Efficient Data Assimilation Schema for Restoration and Extension of Gene Regulatory Networks Using Time-course Observation Data" that is published in *Journal of Computational Biology*.

Chapter 6 is based on the paper "Comprehensive pharmacogenomic pathway screening by data assimilation" that is published in *Bioinformatics Research and Applications*, volume 6674 of *Lecture Notes in Computer Science*.

Chapter 7 is based on the paper "An efficient method of exploring simulation models by assimilating literature and biological observational data" that is published in *BioSystems*.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and Motivation

The cells in living things have the complete set of huge DNA sequences, termed genome, that sustains, controls and differentiates complex biological systems through the expression of genes. For example, human beings have almost 3.1 billion base-pairs of DNA sequences and almost 20 thousands genes are on the DNA sequences. The expression of these genes are controlled by several factors, *e.g.*, mutual regulations among genes, chromatin structures, activities of microRNAs and geographic positions among genes. In the field of systems biology, which focuses on researches of analyzing such intracellular biological systems including regulatory relationships among DNAs, RNAs, proteins and chemical compounds, one of the main goals is the comprehensive understanding of the biological systems since it can contribute to developments of innovative drugs with fewer side effects and medical treatments for new and unknown diseases. In order to achieve this purpose, a great deal of computational methodologies, *e.g.*, Bayesian statistics and time-series modeling, have been developed using biological data of DNAs, RNAs, proteins and chemical compounds.

Due to technological developments of biology, several types of biological data such as microarrays, Chromatin immunoprecipitation Sequencing (Chip-Seq), Chip-Chip and whole genome sequence data have been accumulated. Especially, developments of microarray technologies, which can measure the expression levels of, *e.g.*, mRNA and microRNA, have been greatly contributed to systems biology researches since we can infer regulatory relationships among genes by comparing their expression levels on many different times or conditions. Then, several computational approaches using these data have been developed from the late 20th century according to developments of hardware and software in computer science. For example, methodologies in Bayesian statistics, information theory and time-series analysis, have been developed [9, 30, 45, 70]. Furthermore, recent progresses of biological researches have elucidated parts of biological systems through biological experiments [53]. Thus, although the goal is still far from being understood, many findings related to intracellular biological systems have been published, for example, gene regulatory networks (GRNs), protein-protein interactions (PPI) and transcription factor (TF)

information. Recently, systems biology researches have tried to reveal biological systems incorporating these accumulated findings and observation data from biological experiments.

In regard to these attempts analyzing gene expression data, there exist roughly two types of microarray data, *i.e.*, static and time-course data. The static data is obtained by measuring the expression levels of RNAs at the steady state in, *e.g.*, knockdown cells, which are silenced the expression of some specific genes or are removed such genes. For example, in Bayesian statistics, conditional decencies among genes are represented as probabilistic graphical models and their causal relationships are inferred [30,45]. The time-course data is obtained by measuring the time-dependent expression levels of RNAs after heat shocks, drug stimulations and so on at designed time points. For example, dynamic Bayesian networks [56] and vector auto-regressive (VAR) models [33,34], have been applied to infer the regulatory relationships among genes by assessing the variations of these expressions. In this thesis, we handle the time-course microarray data for the elucidation of biological systems.

For the analysis using these data, there are two major approaches, *i.e.*, simulation-based and statistical approaches. It is well known that the dynamic behavior of biomolecules can be represented by mathematical equations such as differential equations [18,24], *e.g.*, the Michaelis-Menten model [91] and S-system [92]. Thus, in the simulation-based approach, chemical reaction networks (CRNs) and GRNs have been analyzed through the evaluation of these mathematical models that are constructed by combining biomolecular reactions in literature. Following the direction, several methods have been proposed to infer regulatory structures [42,81], to reproduce the dynamic behavior of biological systems recorded in the literature [59, 73, 79, 83, 85, 117], and to improve published pathways so that they are consistent with the data [39, 40]. Although these models can represent detailed dynamics of biological systems, the estimation of their parameter values is computationally heavy task and we cannot evaluate a lot of candidate models. Besides the case, we cannot estimate the parameter values of complex models due to the curse of dimensionality. Therefore, when most part of target systems are unknown, we cannot apply this approach.

In contrast, a statistical approach using highly abstracted models, *e.g.*, Bayesian networks [29, 43, 54, 107, 118, 120], information theory [70, 123], regression-based methods [31–33, 98, 99] and state space models (SSM) [9, 12, 43, 86, 112, 114], has been successfully applied to infer the structures of transcriptional regulation and chemical reactions from biological observational data. Because these methods simply describe biological systems, more than a hundred genes can be handled computationally with ease. Whereas methods relying purely on data need to consider all possibilities of transcriptional regulation, some studies have further incorporated other information including protein-protein interaction networks (PINs), literature-recorded pathways and TF information [7, 20, 22, 88, 109]. Although these methods can infer relationships among more than a hundred genes simultaneously, high levels of abstraction can also generate false regulations that are difficult to interpret biologically.

In this thesis, we propose a set of genomic data assimilation techniques, which tries to collaborate simulation-based and statistical approaches for the inference of biological systems,

using time-course observation data. In order to extend the previous genomic data assimilation, which considers to infer regulatory relationships among more than a hundred genes or evaluates well-known systems consisting of less than ten genes, we establish new models that can be selected depending on the accumulation of biological knowledge and the required accuracy of inference results for target systems, and propose novel methods for the estimation of the parameter values and the inference of regulatory relationships. Thus, incorporating several biological findings, *e.g.*, TF information and literature-based regulatory relationships, we try to explore the biological systems that can best predict observation data based on (i) a linear state space representation of a vector auto-regressive model (VAR-SSM) that covers basic processes of gene regulatory systems as represented in complex nonlinear differential equations, (ii) a simple nonlinear SSM that is constructed by extending the linear VAR-SSM and (iii) a complex nonlinear differential equations within a SSM. For each case, we propose a novel method to estimate the values of the parameters under several biological constraints and infer regulatory relationships among genes to be consistent with the data. The conceptual view is illustrated in Fig 1.1.

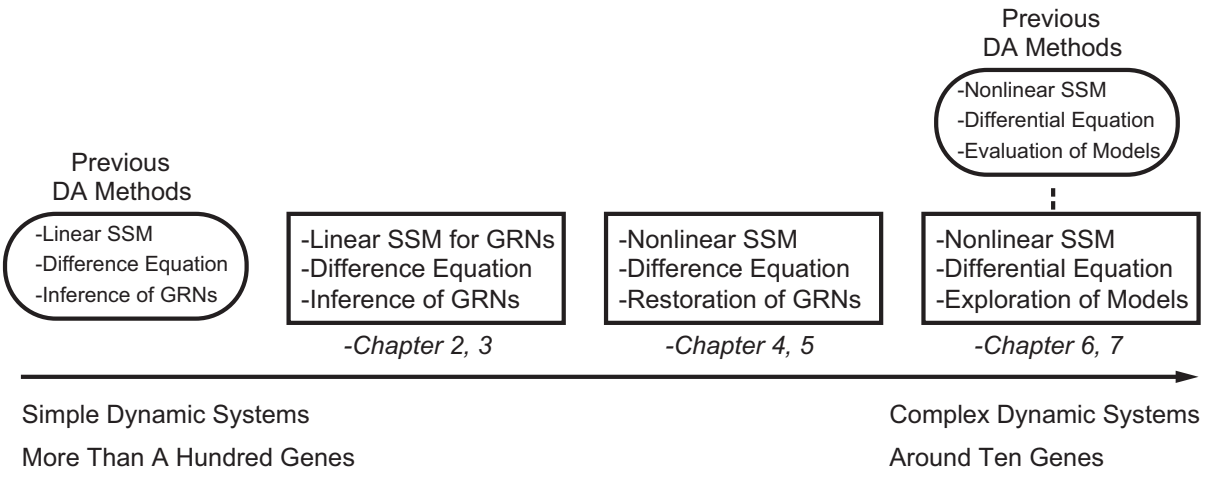

Figure 1.1: A conceptual view explaining the proposed methods in this thesis.

1.2 Data Assimilation

In order to predict response to input data using a set of training data, in the field of computer science, there are roughly two major approaches, *i.e.*, inductive and deductive approaches. The inductive approach that is recently explained as, *e.g.*, data mining, machine learning and statistical science, has great potential for many application fields to achieve the purpose. In contrast, the weakest point of the inductive approach is that it cannot predict incident that has not been learned in the training data. Thus, in the context of a long-term time-series forecasting, it can be difficult to apply the inductive approach to predict the data since it is not realistic to obtain a set of the comprehensive data that covers whole incident.

In contract, theoretically, the deductive approach can overcome the problem. Thus, in this

approach, we first derive the dominant equations describing target phenomena and obtain the dynamic behavior using these equations through simulations because they are often analytically intractable. However, due to the limitation of computational power and the lack of information with respect to the initial conditions, the parameter values and often the dominant equations, we cannot obtain exact results.

The data assimilation approach attempts to collaborate these two approaches by recursively executing the simulation, analyzing the simulated results, estimating the parameter values and reflecting the analyzed results to the simulation. Especially, in the sequential data assimilation schema, SSMs have been utilized using observation data in many application fields, *e.g.*, aerospace, biology and earth science. The basic theories of the state space model is explained in Chapter 2. Then, the proposed extensions for genomic data assimilation are introduced in Chapters 3-7.

1.3 Contribution of Thesis

In this thesis, we deal with four topics in the field of systems biology and genomic data assimilation.

First, in Chapter 2, we introduce basic theories of linear/nonlinear SSMs and their parameter estimation procedures in the context of systems biology. In addition, methodologies of statistics and linear algebra that are used for deriving these theoretical results and the proposed methods in Chapters 3-7 are introduced.

Second, in Chapter 3, we consider the case that the number of target genes is several tens in which regulatory structures are partially known and the purpose is to infer the regulatory relationships among genes incorporating several biological findings, *e.g.*, TF information. Then, we establish a new VAR-SSM that is constructed to cover basic processes of gene regulatory systems as represented in hill function-based differential equations. In contrast to the previous methods utilizing linear models [9, 12, 33, 43, 86, 98, 114], the proposed method can handle the observation data with non-equally spaced time-point data, and system and observation noises separately. Since GRNs are known to have sparse structures, the previous approaches using linear SSMs applied statistical tests to obtain significant regulations. In contrast, we impose $L1$-regularization to the VAR-SSM and propose a method to infer the regulatory structure with updating the parameter values to maximally predict the data. Furthermore, the proposed method can combine several biological findings in inferring the regulatory relationships. When handling several tens of genes and time-course gene expression profiles with drug stimulation, the proposed method shows better performance with respect to the accuracy for inferring the regulatory structure through several computational experiments compared to the previous GRNs inference methods.

Third, in Chapters 4 and 5, we consider the case that there are candidate regulatory networks for target genes and the purpose is to restore the networks to better predict the data based on less abstract models rather than linear models. Then, we employ a state space representation

of a simple nonlinear model termed the combinatorial transcription model [81, 110]. The main problem here is to calculate non-Gaussian conditional distributions of the hidden state variables. In Chapter 4, we apply the unscented Kalman filter (UKF) [16, 49, 51], which efficiently calculates approximate conditional distributions of the hidden state variables, and propose a novel algorithm to improve given GRNs, which are derived from literature or inferred by other GRNs inference methods, utilizing the EM-algorithm. In Chapter 5, to overcome the drawback of UKF for inferring the combinatorial transcription model in which approximate distributions do not fully satisfy the requirements of estimating the optimal parameter values, we further propose a novel method termed a higher moment ensemble particle filter (HMEnPF) that can retain the first two moments and the third and the fourth central moments without reducing the number of the survived particles. Through the simulation experiments, the proposed algorithms successfully restore the given GRNs that are constructed by other GRNs inference methods.

Finally, in Chapters 6 and 7, we consider a SSM in which the system function is described by differential equations to accurately represent the dynamic behavior of biological systems. The purpose is to evaluate the validity of biological pathways, derived from literature, for biological observation data and suggest better pathways that can predict the data. In this scenario, in contrast to the GRNs inference methods in the previous chapters, which attempt to infer only regulatory relationships among genes, we deal with detailed regulatory functions. For example, even when we have a simple pathway that gene A is activated by gene B, we further evaluate cases, *e.g.*, A is linearly activated or not, and a synthesis process of A is activated or a degradation process of A is repressed. In Chapter 6, we propose an approach to systematically creates candidate models from some prototype pathways and comprehensively evaluates these candidate models. However, since we are required to use Monte Carlo approaches to calculate the conditional distributions of the hidden states and the parameter values, it is computationally intensive to evaluate many candidate models, *e.g.*, more than a thousand. In Chapter 7, to overcome the problem, we develop an efficient explorative method that sequentially creates plausible candidates and evaluates them. Through the studies using real data of rat liver cells [47], we show that the proposed method can find the best candidate that is selected by the comprehensive method instead of evaluating all candidate models.

1.4 Organization of Thesis

The rest of this thesis is organized as follows.
In Chapter 2, we explain theories of linear and nonlinear SSMs in context of GRNs inference and introduce solutions for the estimation of the parameter values. Furthermore, complementary theories that are used for deriving these results and the proposed methodologies in Chapters 3-7 are also explained.
In Chapter 3, we establish a VAR-SSM representing gene regulatory systems and propose a method for the inference of regulatory relationships with estimating the parameter values by the EM-algorithm.

In Chapters 4 and 5, we deal with a simple nonlinear SSM in which the system function is the combinatorial transcription model. In Chapter 4, for the estimation of the parameter values, we propose a novel algorithm to improve given GRNs utilizing UKF. In Chapter 5, we further propose a novel method termed HMEnPF that can retain the first two moments and the third and the fourth central moments through the prediction, filtering and smoothing steps.
In Chapters 6 and 7, we deal with a differential equation-based SSM. In Chapter 6, we propose a procedure to explore candidate models that can better predict the data for expression profiles of rat pharmacogenomics. In Chapter 7, we further propose an efficient model exploration method for the same case.

Chapter 2

Preliminarily

2.1 Inference of Gene Regulatory Networks Using Linear State Space Model

Theoretical results explained in this section are based on several literatures [43, 52, 101, 102, 112, 114] and almost all intermediate expression are given by the author (detailed solutions are missing in these literatures).

2.1.1 Linear State Space Model

Let $\boldsymbol{x}_t$ and $\boldsymbol{y}_t$ $(t = 1, \ldots, T)$ be the p-dimensional hidden state variables and the q-dimensional observation variables at time t. We call a time-series modeling representing the following equations as the state space model;

$$\boldsymbol{x}_t = \boldsymbol{f}(\boldsymbol{x}_{t-1}, \boldsymbol{v}_t), \tag{2.1}$$

$$\boldsymbol{y}_t = \boldsymbol{h}(\boldsymbol{x}_t, \boldsymbol{w}_t), \tag{2.2}$$

where $\boldsymbol{f}$, $\boldsymbol{h}$, $\boldsymbol{v}_t$ and $\boldsymbol{w}_t$ are a parametric $R^p \to R^p$ system function generating the next state $\boldsymbol{x}_t$ from the current state $\boldsymbol{x}_{t-1}$, a parametric $R^p \to R^q$ observation function $\boldsymbol{h}$ mapping the state variable $\boldsymbol{x}_t$ to the observation variables $\boldsymbol{y}_t$, p-dimensional system and q-dimensional observation noises, respectively. Here, $\boldsymbol{f}$ and $\boldsymbol{h}$ are possibly nonlinear functions. The basic concept of SSM is shown in Figure 2.1.

Especially, when $\boldsymbol{f}$ and $\boldsymbol{h}$ are linear functions and $\boldsymbol{v}_t$ and $\boldsymbol{w}_t$ are according to Gaussian distributions $N(0, Q)$ where $Q = \mathrm{diag}(q_1, ..., q_p)$ and $N(0, R)$ where $R = \mathrm{diag}(r_1, ..., r_q)$, respectively, a linear SSM can be formulated as

$$\boldsymbol{x}_t = F\boldsymbol{x}_{t-1} + \boldsymbol{v}_t, \tag{2.3}$$

$$\boldsymbol{y}_t = H\boldsymbol{x}_t + \boldsymbol{w}_t, \tag{2.4}$$

where $F \in \mathcal{R}^{p \times p}$ and $H \in \mathcal{R}^{p \times q}$ are termed system and observation matrices, respectively. In

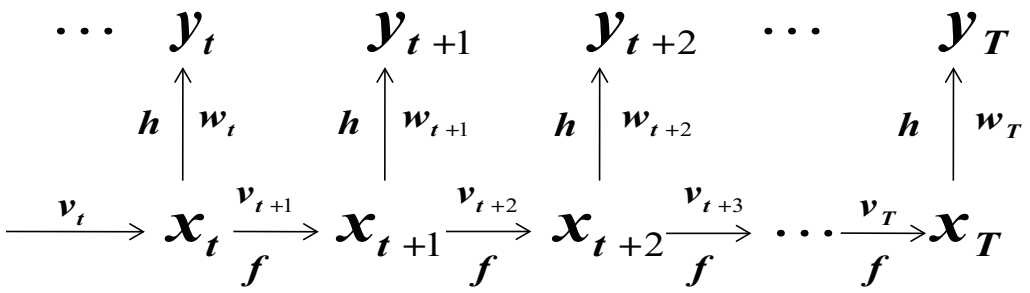

Figure 2.1: A conceptual view of the state space model.

linear SSMs, when $\boldsymbol{x}_0$ is according to a Gaussian distribution, $\boldsymbol{x}_t$ and $\boldsymbol{y}_t$ $(t = 1, \ldots, T)$ also belong to Gaussian distributions.

2.1.2 Estimation of Hidden State Variables Using Kalman Filter

The Kalman filter (KF) is a sequential estimation algorithm for the conditional probability distributions of the hidden state variables given observation data in linear SSMs. Let $Y_s = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_s\}$ be a set of observation data. Then, we describe the expectation and the variance-covariance matrix of the hidden state $\boldsymbol{x}_t$ given Y_s as

$$\boldsymbol{x}_{t|s} = E[\boldsymbol{x}_t|Y_s], \tag{2.5}$$

$$\Sigma_{t|s} = E[(\boldsymbol{x}_t - \boldsymbol{x}_{t|s})(\boldsymbol{x}_t - \boldsymbol{x}_{t|s})'|Y_s]. \tag{2.6}$$

KF formulates the procedure to calculate the optimal states of $p(\boldsymbol{x}_{t+1}|Y_t)$, $p(\boldsymbol{x}_t|Y_t)$ and $p(\boldsymbol{x}_t|Y_T)$ as followings;

1. Prediction: The conditional probability distribution $p(\boldsymbol{x}_t|Y_{t-1})$ is calculated by using $p(\boldsymbol{x}_{t-1}|Y_{t-1})$ as follows.

$$\boldsymbol{x}_{t|t-1} = F\boldsymbol{x}_{t-1|t-1}, \tag{2.7}$$

$$\Sigma_{t|t-1} = F\Sigma_{t-1|t-1}F' + Q. \tag{2.8}$$

(Proof)

$$\begin{aligned}
\boldsymbol{x}_{t|t-1} &= E[\boldsymbol{x}_t|Y_{t-1}], && (2.9)\\
&= E[F\boldsymbol{x}_{t-1} + \boldsymbol{v}_t|Y_{t-1}], && (2.10)\\
&= FE[\boldsymbol{x}_{t-1}|Y_{t-1}], && (2.11)\\
&= F\boldsymbol{x}_{t-1|t-1}, && (2.12)\\
\Sigma_{t|t-1} &= Var[\boldsymbol{x}_t|Y_{t-1}], && (2.13)\\
&= E[\{F(\boldsymbol{x}_{t-1} - \boldsymbol{x}_{t-1|t-1}) + \boldsymbol{v}_t\}\{F(\boldsymbol{x}_{t-1} - \boldsymbol{x}_{t-1|t-1}) + \boldsymbol{v}_t\}'], && (2.14)\\
&= FE[\{\boldsymbol{x}_{t-1} - \boldsymbol{x}_{t-1|t-1}\}\{\boldsymbol{x}_{t-1} - \boldsymbol{x}_{t-1|t-1}\}']F' + E[\boldsymbol{v}_t\boldsymbol{v}_t']', && (2.15)\\
&= F\Sigma_{t-1|t-1}F' + Q'. && (2.16)
\end{aligned}$$

2. Filtering: The conditional probability distribution $p(\boldsymbol{x}_t|Y_t)$ is calculated by using $p(\boldsymbol{x}_t|Y_{t-1})$ and $\boldsymbol{y}_t$ as follows.

$$\begin{aligned}
\boldsymbol{x}_{t|t} &= \boldsymbol{x}_{t|t-1} + K_t(\boldsymbol{y}_t - H\boldsymbol{x}_{t|t-1}), && (2.17)\\
\Sigma_{t|t} &= (I - K_tH)\Sigma_{t|t-1}, && (2.18)\\
K_t &= \Sigma_{t|t-1}H'(H\Sigma_{t|t-1}H' + R)^{-1}. && (2.19)
\end{aligned}$$

For computational ease of calculating inverse matrices, we often apply the Woodbury identity and consider

$$K_t = \Sigma_{t|t-1}H'(R^{-1} - R^{-1}H(\Sigma_{t|t-1}^{-1} + H'R^{-1}H)^{-1}H'R^{-1}). \quad (2.20)$$

(Proof) Let $\boldsymbol{\epsilon}_t$ be

$$\begin{aligned}
\boldsymbol{\epsilon}_t &= \boldsymbol{y}_t - E[\boldsymbol{y}_t|Y_{t-1}], && (2.21)\\
&= \boldsymbol{y}_t - E[H\boldsymbol{x}_t + \boldsymbol{w}_t|Y_{t-1}], && (2.22)\\
&= H(\boldsymbol{x}_t - \boldsymbol{x}_{t|t-1}) + \boldsymbol{w}_t. && (2.23)
\end{aligned}$$

Then, $\boldsymbol{\epsilon}_t$ satisfies the following equations;

$$\begin{aligned}
E[\boldsymbol{\epsilon}_t] &= 0, && (2.24)\\
Var[\boldsymbol{\epsilon}_t] &= V[H(\boldsymbol{x}_t - \boldsymbol{x}_{t|t-1}) + \boldsymbol{w}_t], && (2.25)\\
&= H\Sigma_{t|t-1}H' + R, && (2.26)\\
Cov[\boldsymbol{x}_t, \boldsymbol{\epsilon}_t|Y_{t-1}] &= E[\{\boldsymbol{x}_t - \boldsymbol{x}_{t|t-1}\}\{H(\boldsymbol{x}_t - \boldsymbol{x}_{t|t-1}) + \boldsymbol{w}_t - E[\boldsymbol{\epsilon}_t]\}'], && (2.27)\\
&= \Sigma_{t|t-1}H'. && (2.28)
\end{aligned}$$

Thus, the joint distribution of $\boldsymbol{x}_t$ and $\boldsymbol{\epsilon}_t$ can be described as

$$\begin{pmatrix} \boldsymbol{x}_t \\ \boldsymbol{\epsilon}_t \end{pmatrix} \sim N\left(\begin{pmatrix} \boldsymbol{x}_{t|t-1} \\ 0 \end{pmatrix}, \begin{pmatrix} \Sigma_{t|t-1} & \Sigma_{t|t-1}H' \\ H\Sigma_{t|t-1} & H\Sigma_{t|t-1}H' + R \end{pmatrix} \right). \tag{2.29}$$

By applying Appendices A.1 and A.2 to Eq. (2.29), we can derive

$$\boldsymbol{x}_{t|t} = \boldsymbol{x}_{t|t-1} + K_t(\boldsymbol{y}_t - H\boldsymbol{x}_{t|t-1}), \tag{2.30}$$
$$\Sigma_{t|t} = (I - K_t H)\Sigma_{t|t-1}, \tag{2.31}$$
$$K_t = \Sigma_{t|t-1}H'(H\Sigma_{t|t-1}H' + R)^{-1}. \tag{2.32}$$

3. Smoothing: The conditional probability distribution $p(\boldsymbol{x}_t|Y_T)$ is calculated by using $p(\boldsymbol{x}_t|Y_t)$, $p(\boldsymbol{x}_{t+1}|Y_t)$ and $p(\boldsymbol{x}_{t+1}|Y_T)$.

$$\boldsymbol{x}_{t|T} = \boldsymbol{x}_{t|t} + J_t(\boldsymbol{x}_{t+1|T} - \boldsymbol{x}_{t+1|t}), \tag{2.33}$$
$$\Sigma_{t|T} = \Sigma_{t|t} + J_t(\Sigma_{t+1|T} - \Sigma_{t+1|t})J_t', \tag{2.34}$$
$$J_t = \Sigma_{t|t}F'\Sigma_{t+1|t}^{-1}. \tag{2.35}$$

(Proof) At first, we define the prediction error $\boldsymbol{\delta}_{t+1}$ of $\boldsymbol{x}_{t+1}$ as

$$\boldsymbol{\delta}_{t+1} = \boldsymbol{x}_{t+1} - \boldsymbol{x}_{t+1|t}. \tag{2.36}$$

Then, $\boldsymbol{\delta}_{t+1}$ satisfies the following equations;

$$E[\boldsymbol{\delta}_{t+1}] = 0, \tag{2.37}$$
$$Var[\boldsymbol{\delta}_{t+1}] = \Sigma_{t+1|t}, \tag{2.38}$$
$$Cov[\boldsymbol{x}_t, \boldsymbol{\delta}_{t+1}|Y_t] = Cov[\boldsymbol{x}_t, F(\boldsymbol{x}_t - \boldsymbol{x}_{t|t}) + \boldsymbol{v}_{t+1}], \tag{2.39}$$
$$= E[\{\boldsymbol{x}_t - \boldsymbol{x}_{t|t}\}\{F(\boldsymbol{x}_t - \boldsymbol{x}_{t|t}) + \boldsymbol{v}_{t+1}\}'], \tag{2.40}$$
$$= E[\boldsymbol{x}_t - \boldsymbol{x}_{t|t}]^2 F', \tag{2.41}$$
$$= \Sigma_{t|t}F'. \tag{2.42}$$

Furthermore, we define Z_t as

$$Z_t = Y_t \oplus \boldsymbol{\delta}_{t+1} \oplus \{\boldsymbol{v}_{t+1}, \cdots, \boldsymbol{v}_T\} \oplus \{\boldsymbol{w}_{t+1}, \cdots, \boldsymbol{w}_T\}, \tag{2.43}$$

where $\oplus$ is a direct sum.

Let $\boldsymbol{z}_t$ be the projection to Z_t from $\boldsymbol{x}_t$. Considering $Proj(x|y) = E[x|y]$, where $Proj(x|y)$

is a projection from x to y as explained in Appendix A.3, we can obtain

$$\boldsymbol{z}_t \equiv Proj(\boldsymbol{x}_t|Z_t), \tag{2.44}$$

$$= Proj(\boldsymbol{x}_t|Y_t) + Proj(\boldsymbol{x}_t|\delta_{t+1}) + Proj(\boldsymbol{x}_t|\boldsymbol{v}_{t+1}, \cdots, \boldsymbol{v}_T, \boldsymbol{w}_{t+1}, \cdots, \boldsymbol{w}_T), \tag{2.45}$$

because $\{\boldsymbol{v}_{t+1}, \cdots, \boldsymbol{v}_T\}$, $\{\boldsymbol{w}_{t+1}, \cdots, \boldsymbol{w}_T\}$, $\boldsymbol{\delta}_{t+1}$ and Y_t are orthogonal each other since $\boldsymbol{\delta}_{t+1}$ and Y_t cross at right angles as illustrated in Fig. 2.2 and $\boldsymbol{v}_t$ and $\boldsymbol{w}_t$ are independent from other variables.

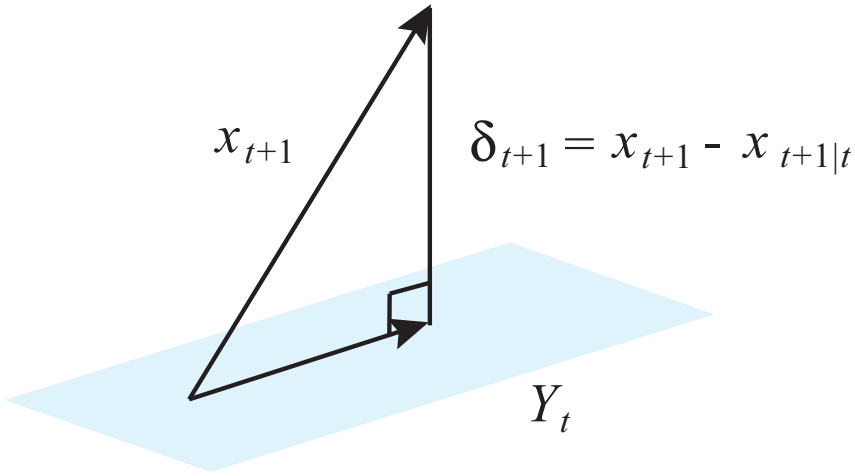

Figure 2.2: δ_{t+1} and Y_t in the probability space.

Utilizing Appendices A.2 and A.3 to Eq. (2.45), we can obtain

$$Proj(\boldsymbol{x}_t|Y_t) = E[\boldsymbol{x}_t|Y_t], \tag{2.46}$$

$$= \boldsymbol{x}_{t|t}, \tag{2.47}$$

$$Proj(\boldsymbol{x}_t|\boldsymbol{\delta}_{t+1}) = E[\boldsymbol{x}_t|\boldsymbol{\delta}_{t+1}], \tag{2.48}$$

$$= Cov[\boldsymbol{x}_t, \boldsymbol{\delta}_{t+1}] Var[\boldsymbol{\delta}_{t+1}]^{-1} \{\boldsymbol{\delta}_{t+1} - E[\boldsymbol{\delta}_{t+1}]\}, \tag{2.49}$$

$$Proj(\boldsymbol{x}_t|\boldsymbol{v}_{t+1}, \cdots, \boldsymbol{v}_T, \boldsymbol{w}_{t+1}, \cdots, \boldsymbol{w}_T) = 0. \tag{2.50}$$

Then, we can derive

$$\boldsymbol{z}_t = \boldsymbol{x}_{t|t} + J_t(\boldsymbol{x}_{t+1} - \boldsymbol{x}_{t+1|t}), \tag{2.51}$$

$$J_t = Cov[\boldsymbol{x}_t, \boldsymbol{\delta}_{t+1}] Var[\boldsymbol{\delta}_{t+1}]^{-1}, \tag{2.52}$$

$$= \Sigma_{t|t} F'_{t+1} \Sigma^{-1}_{t+1}. \tag{2.53}$$

The relationship between Z_t and Y_T can be shown in Fig. 2.3.

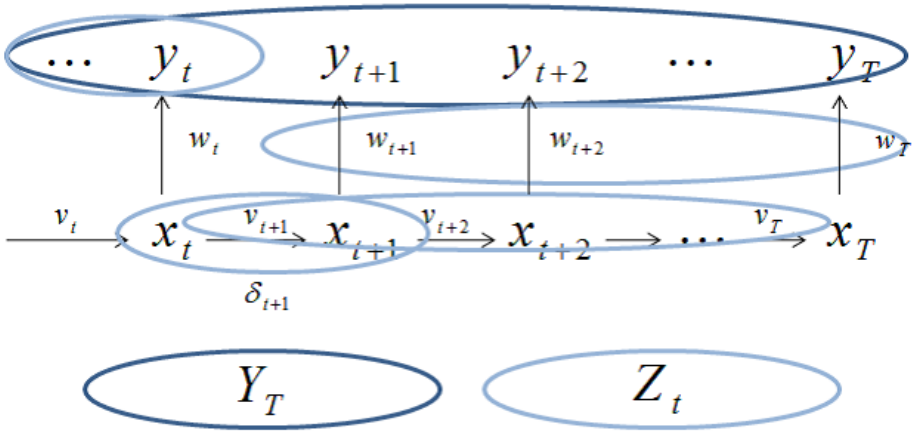

Figure 2.3: The relationship between Z_t and Y_T.

As illustrated in Fig. 2.3, we can obtain Y_T by Z_t; thus, all of the basis vectors in Y_T can be represented by parts of the basis vectors of Z_t. This concludes that the projection to Y_T from the projection to Z_t from $\boldsymbol{x}_t$ corresponds to the projection to Y_T from $\boldsymbol{x}_t$. These relationships are illustrated in Fig. 2.4 and described by

$$\begin{aligned}
\boldsymbol{x}_{t|T} &= Proj(\boldsymbol{x}_t|Y_T), && (2.54)\\
&= Proj\{Proj(\boldsymbol{x}_t|Z_t)|Y_T\}, && (2.55)\\
&= Proj(\boldsymbol{z}_t|Y_T), && (2.56)\\
&= \boldsymbol{x}_{t|t} + J_t(\boldsymbol{x}_{t+1|T} - \boldsymbol{x}_{t+1|t}). && (2.57)
\end{aligned}$$

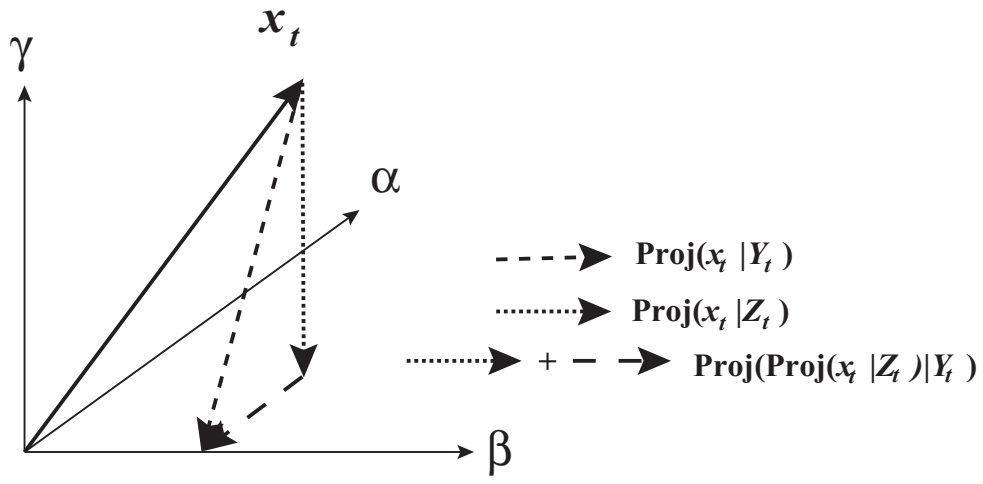

Figure 2.4: The relationships between Z_t and Y_T. $\alpha \oplus \beta \oplus \gamma$:the entire probability space and $\alpha \oplus \beta$: the probability space of Z_t, β : the probability space of Y_t

Next, we derive the variance-covariance matrix $\Sigma_{t|T}$. Considering the second moment of $\boldsymbol{x}_t$ given Y_T, Eq. (2.57) is transformed to

$$\boldsymbol{x}_{t|T} = \boldsymbol{x}_{t|t} + J_t(\boldsymbol{x}_{t+1|T} - \boldsymbol{x}_{t+1|t}), \tag{2.58}$$

$$\Leftrightarrow \boldsymbol{x}_t - \boldsymbol{x}_{t|T} + J_t\boldsymbol{x}_{t+1|T} = \boldsymbol{x}_t - \boldsymbol{x}_{t|t} + J_t\boldsymbol{x}_{t+1|t}, \tag{2.59}$$

$$\Rightarrow \Sigma_{t|T} + J_t E[(\boldsymbol{x}_t - \boldsymbol{x}_{t|T})\boldsymbol{x}'_{t+1|T}|Y_T] + J_t E[\boldsymbol{x}_{t+1|T}\boldsymbol{x}'_{t+1|T}]J'_t, \tag{2.60}$$

$$= \Sigma_{t|t} + J_t E[(\boldsymbol{x}_t - \boldsymbol{x}_{t|t})\boldsymbol{x}'_{t+1|t}|Y_T] + J_t E[\boldsymbol{x}_{t+1|t}\boldsymbol{x}'_{t+1|t}]J'_t. \tag{2.61}$$

$$\Leftrightarrow \Sigma_{t|T} + J_t E[\boldsymbol{x}_{t+1|T}\boldsymbol{x}'_{t+1|T}]J'_t = \Sigma_{t|t} + J_t E[\boldsymbol{x}_{t+1|t}\boldsymbol{x}'_{t+1|t}]J'_t, \tag{2.62}$$

since

$$E[(\boldsymbol{x}_t - \boldsymbol{x}_{t|T})\boldsymbol{x}'_{t+1|T}|Y_T] = E[(\boldsymbol{x}_t - \boldsymbol{x}_{t|t})\boldsymbol{x}'_{t+1|t}|Y_T] = 0. \tag{2.63}$$

Furthermore, we can derive

$$E[\boldsymbol{x}_{t+1|T}\boldsymbol{x}'_{t+1|T}] = E[(\boldsymbol{x}_{t+1|T} - \boldsymbol{x}_{t+1} + \boldsymbol{x}_{t+1})((\boldsymbol{x}_{t+1|T} - \boldsymbol{x}_{t+1} + \boldsymbol{x}_{t+1}))'], \tag{2.64}$$

$$= \Sigma_{t+1|T} + 2E[(\boldsymbol{x}_{t+1|T} - \boldsymbol{x}_{t+1})\boldsymbol{x}'_{t+1}] + E[\boldsymbol{x}_{t+1}\boldsymbol{x}'_{t+1}], \tag{2.65}$$

$$= \Sigma_{t+1|T} - 2E[(\boldsymbol{x}_{t+1|T} - \boldsymbol{x}_{t+1})(\boldsymbol{x}'_{t+1} - \boldsymbol{x}_{t+1})] + 2E[(\boldsymbol{x}_{t+1|T} - \boldsymbol{x}_{t+1})\boldsymbol{x}'_{t+1}] + E[\boldsymbol{x}_{t+1}\boldsymbol{x}'_{t+1}], \tag{2.66}$$

$$= E[\boldsymbol{x}_{t+1}\boldsymbol{x}'_{t+1}] - \Sigma_{t+1|T}, \tag{2.67}$$

$$= E[\boldsymbol{x}_{t+1|t}\boldsymbol{x}'_{t+1|t}] + \Sigma_{t+1|t} - \Sigma_{t+1|T}. \tag{2.68}$$

Finally, by substituting Eq. (2.68) to Eq. (2.62), we can derive

$$\Sigma_{t|T} = \Sigma_{t|t} + J_t(\Sigma_{t+1|T} - \Sigma_{t+1|t})J'_t. \tag{2.69}$$

4. The lag-covariance matrix: The lag-covariance matrix between $\boldsymbol{x}_{t-1}$ and $\boldsymbol{x}_{t-2}$ is obtained as folows.

$$\Sigma_{t-1,t-2|T} = \Sigma_{t-1|t-1}J'_{t-2} + J_{t-1}(\Sigma_{t,t-1|T} - F\Sigma_{t-1|t-1})J'_{t-2}, \tag{2.70}$$

$$\Sigma_{T,T-1|T} = (I - K_T H)F\Sigma_{T-1|T-1}. \tag{2.71}$$

(Proof) Define $\tilde{\boldsymbol{x}}_{t|s} = \boldsymbol{x}_t - \boldsymbol{x}_{t|s}$. Then,

$$\Sigma_{t,t-1|t} = E[\tilde{\boldsymbol{x}}_{t|t}\tilde{\boldsymbol{x}}'_{t-1|t}], \tag{2.72}$$

$$= E[\{(\tilde{\boldsymbol{x}}_{t|t-1} - K_t(\boldsymbol{y}_t - H\boldsymbol{x}_{t|t-1})\}\{\tilde{\boldsymbol{x}}_{t-1|t-1} - J_{t-1}K_t(\boldsymbol{y}_t - H\boldsymbol{x}_{t|t-1})\}'], \tag{2.73}$$

$$= E[\{(\boldsymbol{x}_t - \boldsymbol{x}_{t|t-1}) - K_t(H\tilde{\boldsymbol{x}}_{t|t-1} + \boldsymbol{w}_t)\}\{\tilde{\boldsymbol{x}}_{t-1|t-1} - J_{t-1}K_t(H\tilde{\boldsymbol{x}}_{t|t-1} + \boldsymbol{w}_t)\}'], \tag{2.74}$$

$$= \Sigma_{t,t-1|t-1} - \Sigma_{t|t-1}H'K'_tJ'_{t-1} - K_tH\Sigma'_{t,t-1|t-1} + K_t(H\Sigma_{t|t-1}H' + R)K'_tJ'_{t-1}. \tag{2.75}$$

At $t = T$, we can obtain

$$\Sigma_{T,T-1|T} = (I - K_TH)F\Sigma_{T-1|T-1}, \tag{2.76}$$

since

$$K_t(H\Sigma_{t|t-1}H' + R) = \Sigma_{t|t-1}H', \tag{2.77}$$

$$\Sigma_{t,t-1|t-1} = F\Sigma_{t-1|t-1}. \tag{2.78}$$

In order to derive Eq. (2.70), Eq. (2.33) is transformed to

$$\tilde{\boldsymbol{x}}_{t-1|T} + J_{t-1}\boldsymbol{x}_{t|T} = \boldsymbol{x}_{t-1} - \boldsymbol{x}_{t-1|T} + J_{t-1}\boldsymbol{x}_{t|T}, \tag{2.79}$$

$$= \boldsymbol{x}_{t-1} - \boldsymbol{x}_{t-1|t-1} - J_{t-1}(\boldsymbol{x}_{t|T} - \boldsymbol{x}_{t|t-1}) + J_{t-1}\boldsymbol{x}_{t|T}, \tag{2.80}$$

$$= \tilde{\boldsymbol{x}}_{t-1|t-1} + J_{t-1}F\boldsymbol{x}_{t-1|t-1}. \tag{2.81}$$

Similarly, we can obtain

$$\tilde{\boldsymbol{x}}_{t-2|T} + J_{t-2}\boldsymbol{x}_{t-1|T} = \tilde{\boldsymbol{x}}_{t-2|t-2} + J_{t-2}F\boldsymbol{x}_{t-2|t-2}. \tag{2.82}$$

In regard to the left equation of Eqs. (2.78) and (2.82), we consider the following transformations.

$$E[(\tilde{\boldsymbol{x}}_{t-1|T} + J_{t-1}\boldsymbol{x}_{t|T})(\tilde{\boldsymbol{x}}_{t-2|T} + J_{t-2}\boldsymbol{x}_{t-1|T})'|Y_T], \tag{2.83}$$

$$= E[(\boldsymbol{x}_{t-1} - \boldsymbol{x}_{t-1|T})(\boldsymbol{x}_{t-2} - \boldsymbol{x}_{t-2|T})' + (\boldsymbol{x}_{t-1} - \boldsymbol{x}_{t-1|T})\boldsymbol{x}_{t-1|T}J'_{t-2} \tag{2.84}$$

$$+ J_{t-1}\boldsymbol{x}_{t|T}(\boldsymbol{x}_{t-2} - \boldsymbol{x}_{t-2|T})' + J_{t-1}\boldsymbol{x}_{t|T}\boldsymbol{x}'_{t-1|T}J'_{t-2}|Y_T], \tag{2.85}$$

$$= \Sigma_{t-1,t-2} + J_{t-2}E[\boldsymbol{x}_{t|T}\boldsymbol{x}'_{t-1|T}]J_{t-2}, \tag{2.86}$$

where

$$
\begin{aligned}
E[\boldsymbol{x}_{t|T}\boldsymbol{x}'_{t-1|T}] =& E[\{\boldsymbol{x}_t - (\boldsymbol{x}_t - \boldsymbol{x}_{t|T}\}\{\boldsymbol{x}_{t-1} - (\boldsymbol{x}_{t-1} - \boldsymbol{x}_{t-1|T}\}'|Y_T], && (2.87)\\
=& E[\boldsymbol{x}_t\boldsymbol{x}'_{t-1}] - E[(\boldsymbol{x}_t - \boldsymbol{x}_{t|T})\boldsymbol{x}'_{t-1}] - E[\boldsymbol{x}_t(\boldsymbol{x}_{t-1} - \boldsymbol{x}_{t-1|T})'] + \Sigma_{t,t-1|T}, && (2.88)\\
=& E[\boldsymbol{x}_t\boldsymbol{x}'_{t-1}] - E[(\boldsymbol{x}_t - \boldsymbol{x}_{t|T})\{(\boldsymbol{x}_{t-1} - \boldsymbol{x}_{t-1|T}) + \boldsymbol{x}_{t-1|T}\}'] \\
& - E[\{(\boldsymbol{x}_t - \boldsymbol{x}_{t|T}) + \boldsymbol{x}_{t|T}\}(\boldsymbol{x}_{t-1} - \boldsymbol{x}_{t-1|T})'] + \Sigma_{t,t-1|T}, && (2.89)\\
=& E[\boldsymbol{x}_t\boldsymbol{x}'_{t-1}] - \Sigma_{t,t-1|T}, && (2.90)\\
=& E[(F\boldsymbol{x}_{t-1} + \boldsymbol{v}_t)(F\boldsymbol{x}_{t-2} + \boldsymbol{v}_{t-1})'] - \Sigma_{t,t-1|T}, && (2.91)\\
=& E[F\boldsymbol{x}_{t-1}\boldsymbol{x}'_{t-2}F' + F\boldsymbol{x}_{t-1}\boldsymbol{v}'_{t-1}]\Sigma_{t,t-1|T}, && (2.92)\\
=& E[F\boldsymbol{x}_{t-1}\boldsymbol{x}'_{t-2}F' + F(F\boldsymbol{x}_{t-2} + \boldsymbol{v}_{t-1})\boldsymbol{v}'_{t-1}] - \Sigma_{t,t-1|T}, && (2.93)\\
=& FE[\boldsymbol{x}_{t-1}\boldsymbol{x}'_{t-2}]F' + FQ - \Sigma_{t,t-1|T}. && (2.94)
\end{aligned}
$$

In regard to the right equation of Eq. (2.81), we consider the following transformations.

$$
\begin{aligned}
& \tilde{\boldsymbol{x}}_{t-1|t-1} + J_{t-1}F\boldsymbol{x}_{t-1|t-1}, && (2.95)\\
=& \boldsymbol{x}_{t-1}\{\boldsymbol{x}_{t-1|t-2} + K_{t-1}(\boldsymbol{y}_{t-1} - H\boldsymbol{x}_{t-1|t-2})\} + J_{t-1}F\boldsymbol{x}_{t-1|t-1}, && (2.96)\\
=& (\boldsymbol{x}_{t-1} - \boldsymbol{x}_{t-1|t-2}) - K_{t-1}(H\{\boldsymbol{x}_{t-1} - \boldsymbol{x}_{t-1|t-2}) + \boldsymbol{w}_{t-1}\} + J_{t-1}F\boldsymbol{x}_{t-1|t-1}. && (2.97)
\end{aligned}
$$

Then, we can obtain

$$
\begin{aligned}
& E[\{\tilde{\boldsymbol{x}}_{t-1|t-1} + J_{t-1}F\boldsymbol{x}_{t-1|t-1}\}\{\tilde{\boldsymbol{x}}_{t-2|t-2} + J_{t-2}F\boldsymbol{x}_{t-2|t-2}\}'], && (2.98)\\
=& \Sigma_{t-1,t-2|t-2} - K_{t-1}H\Sigma_{t-1,t-2|t-2} + J_{t-1}FK_tH\Sigma_{t-1,t-2|t-2} && (2.99)\\
& + J_{t-1}FE[\boldsymbol{x}_{t-1|t-2}\boldsymbol{x}'_{t-2|t-2}]F'J'_{t-2}, && (2.100)
\end{aligned}
$$

where

$$
E[\boldsymbol{x}_{t-1|t-2}\boldsymbol{x}_{t-2|t-2}] = E[\boldsymbol{x}_{t-2|t-1}\boldsymbol{x}_{t-2}\boldsymbol{x}'_{t-2}] = E[\boldsymbol{x}_{t-1}\boldsymbol{x}'_{t-2}] - \Sigma_{t-1,t-2|t-2}. \quad (2.101)
$$

By solving Eq. (2.94) = Eq. (2.100), we obtain

$$\begin{aligned}\Sigma_{t-1,t-2|T} =& J_{t-1}\Sigma_{t,t-1|T}J'_{t-2} + \Sigma_{t-1,t-2|t-2} - K_{t-1}H\Sigma_{t-1,t-2|t-2} \\ & - J_{t-1}FQJ'_{t-2} - J_{t-1}F\Sigma_{t-1,t-2|t-2}F'J'_{t-2} + J_{t-1}FK_{t-1}H\Sigma_{t-1,t-2|t-2}, \qquad (2.102)\\ =& J_{t-1}\Sigma_{t,t-1|T}J'_{t-2} - J_{t-1}F(\Sigma_{t-1,t-2|t-2}F' + Q)J'_{t-2} \\ & - J_{t-1}FQJ'_{t-2} - J_{t-1}F\Sigma_{t-1,t-2|t-2}F'J'_{t-2} + J_{t-1}FK_{t-1}H\Sigma_{t-1,t-2|t-2}, \qquad (2.103)\\ =& J_{t-1}\Sigma_{t,t-1|T}J'_{t-2} - J_{t-1}F\Sigma_{t-2|t-1}J'_{t-2} \\ & - J_{t-1}FQJ'_{t-2} - J_{t-1}F\Sigma_{t-1,t-2|t-2}F'J'_{t-2} + J_{t-1}FK_{t-1}H\Sigma_{t-1,t-2|t-2}, \qquad (2.104)\end{aligned}$$

since

$$\begin{aligned}\Sigma_{t-1,t-2|t-2} - K_{t-1}H\Sigma_{t-1,t-2|t-2} &= F\Sigma_{t-2|t-2} - K_{t-1}HF\Sigma_{t-2|t-2}, && (2.105)\\ &= \Sigma_{t-1|t-2}J'_{t-2} - K_{t-1}H\Sigma_{t-1|t-2}J'_{t-2}, && (2.106)\\ &= \Sigma_{t-1|t-1}J'_{t-2}, -J_{t-1}FQJ'_{t-2} - J_{t-1}F\Sigma_{t-1,t-2|t-2}F'J'_{t-2}, && (2.107)\\ &= -J_{t-1}F(\Sigma_{t-1,t-2|t-2}F' + Q)J'_{t-2}, && (2.108)\end{aligned}$$

and

$$\begin{aligned}\Sigma_{t-1,t-2|t-2}F' + Q &= E[(\boldsymbol{x}_{t-1} - \boldsymbol{x}_{t-1|t-2})(F\boldsymbol{x}_{t-2} - Fx_{t-2|t-2})'] + Q, && (2.109)\\ &= E[(\boldsymbol{x}_{t-1} - \boldsymbol{x}_{t-1|t-2})(\boldsymbol{x}_{t-1} - \boldsymbol{x}_{t-1|t-2})'] \\ &\quad - E[(\boldsymbol{x}_{t-1} - \boldsymbol{x}_{t-1|t-2})\boldsymbol{w}'_{t-1}] + Q, && (2.110)\\ &= \Sigma_{t-2|t-1}. && (2.111)\end{aligned}$$

Using Eq. (2.111) for the third and fourth terms in Eq. (2.104), we have

$$-J_{t-1}FQJ'_{t-2} - J_{t-1}F\Sigma_{t-1,t-2|t-2}F'J'_{t-2} = -J_{t-1}F\Sigma_{t-2|t-1}J'_{t-2}. \qquad (2.112)$$

In addition, using Eq. (2.78) for the fifth term in Eq. (2.104), we have

$$J_{t-1}FK_{t-1}H\Sigma_{t-1,t-2|t-2} = J_{t-1}FK_{t-1}HF\Sigma_{t-2|t-2}. \qquad (2.113)$$

Concluding Eqs. (2.112) and (2.113), we obtain

$$J_{t-1}FK_{t-1}HF\Sigma_{t-2|t-2} - J_{t-1}F\Sigma_{t-2|t-1}J'_{t-2} = -J_{t-1}F(I - K_{t-1}H)\Sigma_{t-1|t-2}J_{t-2}, \tag{2.114}$$

$$= -J_{t-1}\Sigma_{t-1|t-1}J_{t-2}. \tag{2.115}$$

Finally, summarizing Eq. (2.104), we can derive

$$\Sigma_{t-1,t-2|T} = \Sigma_{t-1|t-1}J'_{t-2} + J_{t-1}(\Sigma_{t,t-1|T} - F\Sigma_{t-1|t-1})J'_{t-2}, \tag{2.116}$$

$$= \{\Sigma_{t-1|t-1} + J_{t-1}(\Sigma_{t,t-1|T} - F\Sigma_{t-1|t-1})\}J'_{t-2}, \tag{2.117}$$

$$= \{\Sigma_{t-1|t-1} + J_{t-1}(\Sigma_{t,t-1|T} - \Sigma_{t|t-1})\}J'_{t-2}, \tag{2.118}$$

$$= \Sigma_{t-1|T}J'_{t-2}. \tag{2.119}$$

2.1.3 EM-algorithm for Linear Gaussian State Space Model

Let $\{Y_T, X_T\}$ be the complete data, where $X_T = \{\boldsymbol{x}_0, ..., \boldsymbol{x}_T\}$ is the set of hidden state variables. Here, the likelihood function for the complete data is described as

$$P(Y_T, X_T; \boldsymbol{\theta}) = P(\boldsymbol{x}_0)\prod_{t=1}^{T} P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})P(\boldsymbol{y}_t|\boldsymbol{x}_t), \tag{2.120}$$

where $\boldsymbol{\theta} = \{H, F, R, Q, \boldsymbol{\mu}_0\}$ and $\boldsymbol{\mu}_0 = E[\boldsymbol{x}_0]$. We consider to estimate the parameter values $\boldsymbol{\theta}$ maximizing Eq. (2.120) by the EM-algorithm.

In linear SSMs, $P(\boldsymbol{x}_0)$, $P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$ and $P(\boldsymbol{y}_t|\boldsymbol{x}_t)$ are according to Gaussian distributions, $N_p(\boldsymbol{\mu}_0, \Sigma_0)$, $N_p(F\boldsymbol{x}_{t-1}, Q)$ and $N_q(H\boldsymbol{x}_t, R)$, respectively. Then, the logarithm of Eq. (2.120) is described as

$$\log P(\boldsymbol{Y}_T, \boldsymbol{X}_T; \theta) = -\frac{1}{2}\log \mid \Sigma_0 \mid -\frac{1}{2}(\boldsymbol{x}_0 - \boldsymbol{\mu}_0)'\Sigma_0^{-1}(\boldsymbol{x}_0 - \boldsymbol{\mu}_0) \tag{2.121}$$

$$-\frac{T}{2}\log \mid Q \mid -\frac{1}{2}\sum_{t=1}^{T}(\boldsymbol{x}_t - F\boldsymbol{x}_{t-1})'Q^{-1}(\boldsymbol{x}_t - F\boldsymbol{x}_{t-1}) \tag{2.122}$$

$$-\frac{T}{2}\log \mid R \mid -\frac{1}{2}\sum_{t=1}^{T}(\boldsymbol{y}_t - H\boldsymbol{x}_t)'R^{-1}(\boldsymbol{y}_t - H\boldsymbol{x}_t) \tag{2.123}$$

$$-\frac{p + T(p+q)}{2}\log 2\pi. \tag{2.124}$$

In the EM algorithm, the conditional expectation of the joint log-likelihood of the complete data set

$$q(\boldsymbol{\theta}|\boldsymbol{\theta}_i) = E[\log P(\boldsymbol{Y}_T, \boldsymbol{X}_T|\boldsymbol{\theta})|\boldsymbol{Y}_T, \boldsymbol{\theta}_i)], \tag{2.125}$$

is iteratively maximized with respect to $\boldsymbol{\theta}$ until convergence, where $\boldsymbol{\theta}_i$ is the parameter vector

obtained at the ith iteration.

Considering the transformations

$$\begin{aligned}
E[(\boldsymbol{x}_0-\boldsymbol{\mu}_0)'\Sigma_0^{-1}(\boldsymbol{x}_0-\boldsymbol{\mu}_0)] =& E[\mathrm{tr}\{\Sigma_0^{-1}(\boldsymbol{x}_0-\boldsymbol{\mu}_0)(\boldsymbol{x}_0-\boldsymbol{\mu}_0)'\}], && (2.126)\\
=& \mathrm{tr}E[\{\Sigma_0^{-1}(\boldsymbol{x}_0-\boldsymbol{\mu}_0)(\boldsymbol{x}_0-\boldsymbol{\mu}_0)'\}], && (2.127)\\
=& \mathrm{tr}\{\Sigma_0^{-1}E[(\boldsymbol{x}_0-\boldsymbol{\mu}_0)(\boldsymbol{x}_0-\boldsymbol{\mu}_0)']\}, && (2.128)\\
=& \mathrm{tr}\{\Sigma_0^{-1}E[(\boldsymbol{x}_0-\boldsymbol{x}_{0|T}+\boldsymbol{x}_{0|T}-\boldsymbol{\mu}_0)\\
& \cdot(\boldsymbol{x}_0-\boldsymbol{x}_{0|T}+\boldsymbol{x}_{0|T}-\boldsymbol{\mu}_0)']\}, && (2.129)\\
=& \mathrm{tr}\{\Sigma_0^{-1}(\Sigma_{0|T}+(\boldsymbol{x}_{0|T}-\boldsymbol{\mu}_0)(\boldsymbol{x}_{0|T}-\boldsymbol{\mu}_0)')\}, && (2.130)
\end{aligned}$$

we can obtain

$$\begin{aligned}
q(\boldsymbol{\theta}|\boldsymbol{\theta}_i) =& -\frac{1}{2}\log\mid\Sigma_0\mid -\frac{1}{2}\mathrm{tr}\{\Sigma_0^{-1}(\Sigma_{0|T}+(\boldsymbol{x}_{0|T}-\boldsymbol{\mu}_0)(\boldsymbol{x}_{0|T}-\boldsymbol{\mu}_0)')\} && (2.131)\\
& -\frac{T}{2}\log\mid Q\mid -\frac{1}{2}\mathrm{tr}\{Q^{-1}(C-BF'-FB'+FAF')\} && (2.132)\\
& -\frac{T}{2}\log\mid R\mid -\frac{1}{2}\mathrm{tr}[R^{-1}\sum_{t=1}^{T}\{(\boldsymbol{y}_t-H\boldsymbol{x}_{t|T})(\boldsymbol{y}_t-H\boldsymbol{x}_{t|T})' && (2.133)\\
& +H\Sigma_{t|T}H'\}] - \frac{p+T(p+q)}{2}\log 2\pi, && (2.134)
\end{aligned}$$

and

$$A=\sum_{t=1}^{T}(\Sigma_{t-1|T}+\boldsymbol{x}_{t-1|T}\boldsymbol{x}'_{t-1|T}), \tag{2.135}$$

$$B=\sum_{t=1}^{T}(\Sigma_{t,t-1|T}+\boldsymbol{x}_{t|T}\boldsymbol{x}'_{t-1|T}), \tag{2.136}$$

$$C=\sum_{t=1}^{T}(\Sigma_{t|T}+\boldsymbol{x}_{t|T}\boldsymbol{x}'_{t|T}), \tag{2.137}$$

where $\boldsymbol{x}_{t|T}$, $\boldsymbol{y}_{t|T}$, $\Sigma_{t|T}$ and $\Sigma_{t,t-1}$ $(t=1,\cdots,T)$ are calculated by KF.

In the M-step, we differentiate

$$\frac{\partial q(\boldsymbol{\theta}|\boldsymbol{\theta}_i)}{\partial\boldsymbol{\theta}}=0, \tag{2.138}$$

to update the parameter values. Note that we apply the procedure of the differentiation explained in Appendix A.4.

For H, we solve

$$\frac{\partial q(\boldsymbol{\theta}|\boldsymbol{\theta}_i)}{\partial H}=0. \tag{2.139}$$

Then, we can obtain

$$\frac{\partial q(\boldsymbol{\theta}|\boldsymbol{\theta}_i)}{\partial H} = \frac{\partial \mathrm{tr}\{R^{-1}\sum_{t=1}^{T}[(\boldsymbol{y}_t - H\boldsymbol{x}_{t|T})(\boldsymbol{y}_t - H\boldsymbol{x}_{t|T})' + H\Sigma_{t|T}H']\}}{\partial H}, \tag{2.140}$$

$$= R^{-1}\sum_{t=1}^{T}\{-2\boldsymbol{y}_t\boldsymbol{x}'_{t|T} + 2H\boldsymbol{x}_{t|T}\boldsymbol{x}'_{t|T} + 2H\Sigma_{t|T}\}, \tag{2.141}$$

$$= 0, \tag{2.142}$$

$$\Longleftrightarrow H = \sum_{t=1}^{T}(\boldsymbol{y}_t\boldsymbol{x}'_{t|T})\sum_{t=1}^{T}(\Sigma_{t|T} + \boldsymbol{x}_{t|T}\boldsymbol{x}'_{t|T})^{-1}, \tag{2.143}$$

$$\Longleftrightarrow H = \{\sum_{t=1}^{T}E(\boldsymbol{y}_t\boldsymbol{x}'_t|Y_T)\}\cdot C^{-1}. \tag{2.144}$$

For F, we solve

$$\frac{\partial q(\boldsymbol{\theta}|\boldsymbol{\theta}_i)}{\partial F} = 0. \tag{2.145}$$

Then, we can obtain

$$\frac{\partial q(\boldsymbol{\theta}|\boldsymbol{\theta}_i)}{\partial F} = (Q^{-1})'\{-B' - B + F(A + A')\}, \tag{2.146}$$

$$= Q^{-1}\{-2B + 2FA\}, \tag{2.147}$$

$$= 0, \tag{2.148}$$

$$\Longleftrightarrow F = BA^{-1}. \tag{2.149}$$

For $\boldsymbol{\mu}_0$, we solve

$$\frac{\partial q(\boldsymbol{\theta}|\boldsymbol{\theta}_i)}{\partial \boldsymbol{\mu}_0} = 0. \tag{2.150}$$

Then, we can obtain

$$\frac{\partial q(\boldsymbol{\theta}|\boldsymbol{\theta}_i)}{\partial \boldsymbol{\mu}_0} = -\frac{1}{2}\Sigma_0^{-1}(\boldsymbol{x}_{0|T} - \boldsymbol{\mu}_0), \tag{2.151}$$

$$= 0, \tag{2.152}$$

$$\Longleftrightarrow \boldsymbol{\mu}_0 = \boldsymbol{x}_{0|T}. \tag{2.153}$$

For R, considering $R^{-1} = P$, we solve

$$\frac{\partial q(\boldsymbol{\theta}|\boldsymbol{\theta}_i)}{\partial P} = 0. \tag{2.154}$$

Then, we can obtain

$$\frac{\partial q(\boldsymbol{\theta}|\boldsymbol{\theta}_i)}{\partial P} = \frac{T}{2}R - \frac{1}{2}\sum_{t=1}^{T}[(\boldsymbol{y}_t - H\boldsymbol{x}_{t|T})(\boldsymbol{y}_t - H\boldsymbol{x}_{t|T})' + H\Sigma_{t|T}H'], \tag{2.155}$$

$$= 0, \tag{2.156}$$

$$\Longleftrightarrow R = \frac{1}{T}\sum_{t=1}^{T}[(\boldsymbol{y}_t - H\boldsymbol{x}_{t|T})(\boldsymbol{y}_t - H\boldsymbol{x}_{t|T})' + H\Sigma_{t|T}H']. \tag{2.157}$$

For Q, considering $Q^{-1} = K$, we solve

$$\frac{\partial q(\boldsymbol{\theta}|\boldsymbol{\theta}_i)}{\partial K} = 0. \tag{2.158}$$

Then, we can obtain

$$\frac{\partial q(\boldsymbol{\theta}|\boldsymbol{\theta}_i)}{\partial K} = TQ - \{C - BF' - FB' + FAF'\}, \tag{2.159}$$

$$= TQ - \{C - B(BA^{-1})' - BA^{-1}B' + B(A^{-1})'B'\}, \tag{2.160}$$

$$= TQ - \{C - BA^{-1}B'\}, \tag{2.161}$$

$$= 0, \tag{2.162}$$

$$\Longleftrightarrow Q = \frac{1}{T}(C - BA^{-1}B'). \tag{2.163}$$

The EM-algorithm monotonically increases the log-likelihood of the observation data

$$\log P(Y_T) = -\frac{1}{2}\sum_{t=1}^{T}\{\log|2\pi V_{t|t-1}| + (\boldsymbol{y}_t - H\boldsymbol{x}_{t|t-1})'V_{t|t-1}^{-1}(\boldsymbol{y}_t - H\boldsymbol{x}_{t|t-1})\}, \tag{2.164}$$

$$V_{t|t-1} = H\Sigma_{t|t-1}H' + R. \tag{2.165}$$

2.2 Nonlinear State Space Model

Consider the system and observation equations

$$\boldsymbol{x}_{t+1} = \boldsymbol{f}(\boldsymbol{x}_t, \boldsymbol{v}_{t+1}), \tag{2.166}$$

$$\boldsymbol{y}_t = \boldsymbol{h}(\boldsymbol{x}_t, \boldsymbol{w}_t), \tag{2.167}$$

in the context of nonlinear SSMs.

We can calculate the conditional probability distributions $p(\boldsymbol{x}_{t+1}|Y_t)$ and $p(\boldsymbol{x}_{t+1}|Y_{t+1})$ as follows.

Prediction

$$
\begin{aligned}
p(\boldsymbol{x}_{t+1}|Y_t) &= \int p(\boldsymbol{x}_{t+1}, \boldsymbol{x}_t|Y_t)d\boldsymbol{x}_t, && (2.168)\\
&= \int p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, Y_t)p(\boldsymbol{x}_t|Y_t)d\boldsymbol{x}_t, && (2.169)\\
&= \int p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t)p(\boldsymbol{x}_t|Y_t)d\boldsymbol{x}_t. && (2.170)
\end{aligned}
$$

Filtering

$$
\begin{aligned}
p(\boldsymbol{x}_t|Y_t) &= p(\boldsymbol{x}_t|\boldsymbol{y}_t, Y_{t-1}), && (2.171)\\
&= \frac{p(\boldsymbol{y}_t|\boldsymbol{x}_t, Y_{t-1})p(\boldsymbol{x}_t|Y_{t-1})}{p(\boldsymbol{y}_t|Y_{t-1})}, && (2.172)\\
&= \frac{p(\boldsymbol{y}_t|\boldsymbol{x}_t)p(\boldsymbol{x}_t|Y_{t-1})}{p(\boldsymbol{y}_t|Y_{t-1})}, && (2.173)\\
&= \frac{p(\boldsymbol{y}_t|\boldsymbol{x}_t)p(\boldsymbol{x}_t|Y_{t-1})}{\int p(\boldsymbol{y}_t|\boldsymbol{x}_t)p(\boldsymbol{x}_t|Y_{t-1})d\boldsymbol{x}_t}. && (2.174)
\end{aligned}
$$

We can calculate $p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t)$ by the system function $\boldsymbol{f}$ and $p(\boldsymbol{y}_t|\boldsymbol{x}_t)$ by the observation function $\boldsymbol{h}$. Therefore, we can recursively obtain $p(\boldsymbol{x}_{t+1}|Y_t)$ and $p(\boldsymbol{x}_t|Y_t)$ from $\boldsymbol{p}(\boldsymbol{x}_0)$.

Smoothing

$$
\begin{aligned}
p(\boldsymbol{x}_t, \boldsymbol{x}_{t-1}|Y_t) &= p(\boldsymbol{x}_t|Y_N)p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, Y_{t-1}), && (2.175)\\
&= p(\boldsymbol{x}_t|Y_N)p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, Y_{t-1}), && (2.176)\\
&= p(\boldsymbol{x}_t|Y_N)\frac{p(\boldsymbol{x}_{t-1}|Y_{t-1})p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, Y_{t-1})}{p(\boldsymbol{x}_t|Y_{t-1})}, && (2.177)\\
&= p(\boldsymbol{x}_t|Y_N)\frac{p(\boldsymbol{x}_{t-1}|Y_{t-1})p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})}{p(\boldsymbol{x}_t|Y_{t-1})}, && (2.178)\\
p(\boldsymbol{x}_{t-1}|Y_t) &= \int_{-\infty}^{\infty} p(\boldsymbol{x}_{t-1}, \boldsymbol{x}_t|Y_{t-1})d\boldsymbol{x}_t. && (2.179)
\end{aligned}
$$

In contrast to linear SSMs, actual calculations of these distributions in nonlinear SSMs are difficult in many cases. Therefore, we introduce the particle filter (PF) [36, 57, 58] for actual calculations.

2.2.1 Particle Filter

PF was developed by Gordon *et al.* [36] and Kitagawa *et al.* [58] at the same time but in another place. Although the particle filter had been called 'Monte Carlo filter' by Kitagawa and 'bootstrap filter' by Gordon, now we generally call 'particle filter' as the name for the algorithm.

The basic idea of PF is to approximate a probability distribution by using a set of particles termed ensemble as shown in Fig. 2.5.

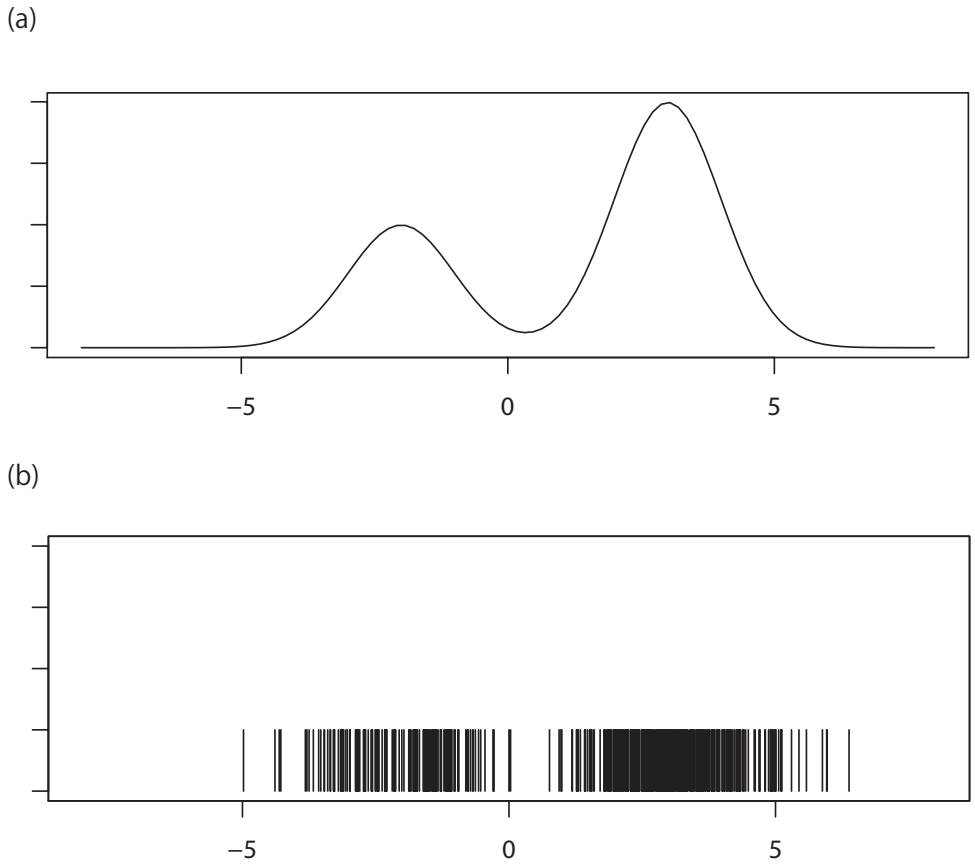

Figure 2.5: An example of the approximation of a probability distribution using PF. The probability distribution illustrated in (a) is approximated by a set of particles as illustrated in (b). The number of particles $N = 500$.

This algorithm is easy to be implemented computationally, and can be adopted to problems handling non-Gaussian distributions but requires huge memory space.

Let sets of particles $\{\boldsymbol{p}_t^1, \ldots, \boldsymbol{p}_t^n, \ldots, \boldsymbol{p}_t^N\}$, $\{\boldsymbol{f}_t^1, \ldots, \boldsymbol{f}_t^n, \ldots, \boldsymbol{f}_t^N\}$, $\{\boldsymbol{s}_t^1, \ldots, \boldsymbol{s}_t^n, \ldots, \boldsymbol{s}_t^N\}$ and $\{\boldsymbol{v}_t^1, \ldots, \boldsymbol{v}_t^n, \ldots, \boldsymbol{v}_t^N\}$ be samples from $p(\boldsymbol{x}_t|Y_{t-1})$, $p(\boldsymbol{x}_t|Y_t)$, $p(\boldsymbol{x}_t|Y_T)$ and $p(\boldsymbol{v}_t)$, respectively. Using these sets of particles we approximate $p(\boldsymbol{x}_t|Y_t)$ as

$$p(\boldsymbol{x}_t|Y_t) \simeq \frac{1}{N}\sum_{n=1}^{N}\delta(\boldsymbol{x}_t - \boldsymbol{f}_t^n), \tag{2.180}$$

where $\delta(\cdot)$ and N are the Dirac delta function and the amount of particles, respectively. Theoretically, we can represent any complex distribution by ensemble if there exists enough amount of particles. Similar to section 2.2, we derive the conditional distributions through 'Prediction' and 'Filtering' and 'Smoothing' in the context of PF.

Prediction

We have the ensemble $\{\boldsymbol{f}_{t-1}^n\}$ and $\{\boldsymbol{v}_t^n\}$. Then, the conditional probability distribution of $\boldsymbol{x}_t$ given Y_{t-1} is calculated by

$$p(\boldsymbol{x}_t|Y_{t-1}) = \int\int p(\boldsymbol{x}_t, \boldsymbol{x}_{t-1}, \boldsymbol{v}_t|Y_{t-1})d\boldsymbol{x}_{t-1}d\boldsymbol{v}_t, \tag{2.181}$$

$$= \int\left[\int p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{v}_t, Y_{t-1})p(\boldsymbol{v}_t|\boldsymbol{x}_{t-1}, Y_{t-1})d\boldsymbol{v}_t\right] \cdot p(\boldsymbol{x}_{t-1}|Y_{t-1})d\boldsymbol{x}_{t-1}. \tag{2.182}$$

Hence $\boldsymbol{x}_t$ depends only on $\boldsymbol{x}_{t-1}$ and $\boldsymbol{v}_t$, and the system noise $\boldsymbol{v}_t$ is independent from all parameters, Eq. (2.182) can be written as

$$p(\boldsymbol{x}_t|Y_{t-1}) = \int\int p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{v}_t)p(\boldsymbol{v}_t)p(\boldsymbol{x}_{t-1}|Y_{t-1})d\boldsymbol{x}_{t-1}d\boldsymbol{v}_t. \tag{2.183}$$

The probability distribution of the system noise $\boldsymbol{v}_t$ is described as

$$p(\boldsymbol{v}_t) \simeq \frac{1}{N}\sum_{n=1}^{N}\delta(\boldsymbol{v}_t - \boldsymbol{v}_t^n), \tag{2.184}$$

then,

$$p(\boldsymbol{x}_t|Y_{t-1}) \simeq \int\int\left[\frac{1}{N}\sum_{n=1}^{N}p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{v}_t)\delta(\boldsymbol{x}_{t-1} - \boldsymbol{f}_{t-1}^n)\delta(\boldsymbol{v}_t - \boldsymbol{v}_t^n)\right]d\boldsymbol{x}_{t-1}d\boldsymbol{v}_t, \tag{2.185}$$

$$= \frac{1}{N}\sum_{n=1}^{N}p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}^n, \boldsymbol{v}_t^n), \tag{2.186}$$

$$= \frac{1}{N}\sum_{n=1}^{N}\delta(\boldsymbol{x}_t - \boldsymbol{f}(\boldsymbol{f}_{t-1}^n, \boldsymbol{v}_t^n)). \tag{2.187}$$

Consequently, we can obtain

$$\boldsymbol{p}_t^n = \boldsymbol{f}(\boldsymbol{f}_{t-1}^n, \boldsymbol{v}_t^n). \tag{2.188}$$

Filtering

Assume that we have the ensemble $\{\boldsymbol{p}_t^n\}$. In the filtering step, we first calculate the likelihoods of $\boldsymbol{p}_t^n$ based on the observed value $\boldsymbol{y}_t$. Let α_t^n be the likelihood of $\boldsymbol{p}_t^n$. Then, α_t^n is calculated by

$$\alpha_t^n = p(\boldsymbol{y}_t|\boldsymbol{p}_t^n). \tag{2.189}$$

From Eq. (2.174), we can approximate the conditional distribution $p(\boldsymbol{x}_t|Y_t)$ as

$$p(\boldsymbol{x}_t|Y_t) = \frac{p(\boldsymbol{y}_t|\boldsymbol{x}_t)p(\boldsymbol{x}_t|Y_{t-1})}{\int p(\boldsymbol{y}_t|\boldsymbol{x}_t)p(\boldsymbol{x}_t|Y_{t-1})d\boldsymbol{x}_t}, \tag{2.190}$$

$$\simeq \frac{1}{\sum_{n=1}^{N} p(\boldsymbol{y}_t|\boldsymbol{p}_t^n)} \sum_{n=1}^{N} p(\boldsymbol{y}_t|\boldsymbol{p}_t^n)\delta(\boldsymbol{x}_t - \boldsymbol{p}_t^n), \tag{2.191}$$

$$= \sum_{n=1}^{N} \frac{\alpha_t^n}{\sum_{m=1}^{N} \alpha_t^m} \delta(\boldsymbol{x}_t - \boldsymbol{p}_t^n). \tag{2.192}$$

Since we can consider α_t^n as the importance of $\boldsymbol{p}_t^n$ for constructing $p(\boldsymbol{x}_t|Y_t)$, we sample $\boldsymbol{p}_t^n$ in proportion to α_t^n. Thus, the ensemble $\{\boldsymbol{f}_t^n\}$ is obtained by sampling with replacement as

$$\boldsymbol{f}_t^n = \begin{cases} \boldsymbol{p}_t^1 & \text{with } \frac{\alpha_t^1}{\sum_{n=1}^{N} \alpha_t^n}, \\ \vdots & \vdots \\ \boldsymbol{p}_t^N, & \text{with } \frac{\alpha_t^N}{\sum_{n=1}^{N} \alpha_t^n}. \end{cases} \tag{2.193}$$

Particle Filter

Consequently, we can obtain the conditional distributions of the hidden state variables as follows.

1. Generate $\{\boldsymbol{p}_0^n\} = \{\boldsymbol{f}_0^n\}$ according to the prior distribution $p(\boldsymbol{x}_0)$.
2. Calculate $\{\boldsymbol{p}_t^n\}$ by using $\{\boldsymbol{f}_{t-1}^n\}$ and $\{\boldsymbol{v}_t^n\}$.
3. Calculate $\{\alpha_t^n\}$ based on $\boldsymbol{y}_t$ and $\{\boldsymbol{w}_t^n\}$.
4. Obtain $\{\boldsymbol{f}_t^n\}$ by resampling $\{\boldsymbol{p}_t^n\}$.
5. Repeat (2)-(4) until t becomes T.

Smoothing

From Eq. (2.179), we have

$$p(\boldsymbol{x}_{t-1}|Y_t) = \int p(\boldsymbol{x}_t|Y_N) \frac{p(\boldsymbol{x}_{t-1}|Y_{t-1})p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})}{p(\boldsymbol{x}_t|Y_{t-1})} d\boldsymbol{x}_t. \tag{2.194}$$

Based on the Bayse theorem, we obtain

$$\frac{p(\boldsymbol{x}_t|Y_t)}{p(\boldsymbol{x}_t|Y_{t-1})} = \frac{p(\boldsymbol{x}_t|Y_{t-1}, Y_{t:T})}{p(\boldsymbol{x}_t|Y_{t-1})}, \tag{2.195}$$

$$= \frac{\frac{p(\boldsymbol{x}_t, Y_{t:T}|Y_{t-1})}{p(Y_{t:T}|Y_{t-1})}}{p(\boldsymbol{x}_t|Y_{t-1})}, \tag{2.196}$$

$$= \frac{p(Y_{t:T}|\boldsymbol{x}_t, Y_{t-1})}{p(Y_{t:T}|Y_{t-1})}, \tag{2.197}$$

where $Y_{t:T} = \{\boldsymbol{y}_t, \ldots, \boldsymbol{y}_T\}$.

Then, Eq. (2.194) is transformed to

$$
\begin{aligned}
p(\boldsymbol{x}_{t-1}|Y_t) &= \int \frac{p(Y_{t:T}|\boldsymbol{x}_t, Y_{t-1})}{p(Y_{t:T}|Y_{t-1})} p(\boldsymbol{x}_{t-1}|Y_{t-1}) p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) d\boldsymbol{x}_t, && (2.198)\\
&= \int\int \frac{p(Y_{t:T}|\boldsymbol{x}_t, Y_{t-1})}{p(Y_{t:T}|Y_{t-1})} \cdot p(\boldsymbol{x}_{t-1}|Y_{t-1}) p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{v}_t) p(\boldsymbol{v}_t) d\boldsymbol{x}_t d\boldsymbol{v}_t, && (2.199)\\
&\simeq \int \frac{p(Y_{t:T}|\boldsymbol{x}_t, Y_{t-1})}{p(Y_{t:T}|Y_{t-1})} \cdot \frac{1}{N} \sum_{n=1}^{N} \delta(\boldsymbol{x}_t - \boldsymbol{p}_t^n), \delta(\boldsymbol{x}_{t-1} - \boldsymbol{f}_{t-1}^n) d\boldsymbol{x}_t, && (2.200)\\
&\simeq \sum_{n=1}^{N} \frac{p(Y_{t:T}|\boldsymbol{p}_t^n, Y_{t-1})}{\sum_{m=1}^{N} p(Y_{t:T}|\boldsymbol{p}_t^m, Y_{t-1})} \delta(\boldsymbol{x}_{t-1} - f_t^n). && (2.201)
\end{aligned}
$$

As a result, we can calculate the smoothed distributions as same as the filtering distributions. $\{\boldsymbol{s}_t^n\}$ is called 'particle smoother'.

Likelihood

In PF, the likelihood is approximated to be calculated by

$$
\begin{aligned}
p(\boldsymbol{y}_t|Y_{t-1}; \boldsymbol{\theta}) &= \int p(\boldsymbol{y}_t|\boldsymbol{x}_t; \boldsymbol{\theta}) p(\boldsymbol{x}_t|Y_{t-1}; \boldsymbol{\theta}) d\boldsymbol{x}_t, && (2.202)\\
&\simeq \frac{1}{N} \sum_{t=1}^{T} p(\boldsymbol{y}_t|\boldsymbol{p}_t^n), && (2.203)\\
&= \frac{1}{N} \sum_{n=1}^{N} \alpha_t^n. && (2.204)
\end{aligned}
$$

Then, we have

$$
\begin{aligned}
l(\boldsymbol{\theta}) = \log p(Y_T; \boldsymbol{\theta}) &= \sum_{t=1}^{T} \log p(\boldsymbol{y}_t|Y_{t-1}; \boldsymbol{\theta}), && (2.205)\\
&\simeq \sum_{t=1}^{T} \log(\sum_{n=1}^{N} \alpha_t^n) - T \log N. && (2.206)
\end{aligned}
$$

In principle, we prompt to maximize the likelihood by estimating the optimal parameter vector $\hat{\boldsymbol{\theta}}$. However, due to the approximation errors and often the difficulties of applying optimization methods such as Newton method, Kitagawa *et al.* [58] developed the self-organized state space model to obtain the posterior distributions of the parameter values.

2.3 Self-organizing State Space Model

To obtain the posterior distributions of the parameter vector $\boldsymbol{\theta}$, Kitagawa *et al.* [58] developed the self-organizing state space model. Set the hidden state vector $\boldsymbol{z}_t$ as

$$\boldsymbol{z}_t = \begin{bmatrix} \boldsymbol{x}_t \\ \boldsymbol{\theta} \end{bmatrix}, \boldsymbol{\theta} = \begin{bmatrix} \boldsymbol{\theta}_s \\ \boldsymbol{\theta}_o \end{bmatrix}, \tag{2.207}$$

where $\boldsymbol{\theta}_s$ is the parameter vector for the system model, $\boldsymbol{\theta}_o$ is the parameter vector for the observation model. Therefore, Eqs. (2.166) and (2.167) are written by

$$\begin{aligned} \boldsymbol{z}_{t+1} &= \boldsymbol{f}(\boldsymbol{z}_t, \boldsymbol{v}_{t+1}, \boldsymbol{\theta}_s), & (2.208) \\ \boldsymbol{y}_t &= \boldsymbol{h}(\boldsymbol{z}_t, \boldsymbol{w}_t, \boldsymbol{\theta}_o), & (2.209) \end{aligned}$$

where

$$\begin{aligned} \boldsymbol{f}(\boldsymbol{z}_t, \boldsymbol{v}_{t+1}, \boldsymbol{\theta}_s) &= \begin{bmatrix} \boldsymbol{f}(\boldsymbol{x}_t, \boldsymbol{v}_{t+1}, \boldsymbol{\theta}_s) \\ \boldsymbol{\theta} \end{bmatrix}, & (2.210) \\ \boldsymbol{h}(\boldsymbol{z}_t, \boldsymbol{w}_t, \boldsymbol{\theta}_o) &= \boldsymbol{h}(\boldsymbol{x}_t, \boldsymbol{w}_t, \boldsymbol{\theta}_o). & (2.211) \end{aligned}$$

After obtaining the posterior distribution $p(\boldsymbol{z}_t|Y_T)$ by PF, we can get the maximum likelihood estimator $\hat{\boldsymbol{\theta}}$ by marginalizing $p(\boldsymbol{z}_t|Y_T)$.

2.4 Bayesian Information Criterion

To evaluate a model fittingness to the observed data, Akaike *et al.* [2] and Schwarz*et al.* [96] developed the Bayesian information criterion (BIC) based on the posterior probabilities of observed data given models. Here, we have a set of candidate models $\{M_1, \ldots, M_k, \ldots, M_K\}$, parametric model $\boldsymbol{f}_k(\boldsymbol{x}_t, \boldsymbol{\theta}_k)$ $(= \boldsymbol{f}_k)$ according to the model M_k, the observed data D, the parameter vector $\boldsymbol{\theta}_k \in R^{\nu_k}$ and the prior distribution $\pi_k(\boldsymbol{\theta}_k)$. Our interest is a marginal likelihood $L(M_k)$ or log-likelihood $l(M_k)$ of the model M_k, *i.e.*,

$$\begin{aligned} L(M_k) = P(Y_t|M_k) &= \int p(\boldsymbol{f}_k|\boldsymbol{\theta}_k)\pi_k(\boldsymbol{\theta}_k)d\boldsymbol{\theta}r, & (2.212) \\ &= \int \exp\{\log p(\boldsymbol{f}_k|\boldsymbol{\theta}_k)\}\pi_k(\boldsymbol{\theta}_k)d\boldsymbol{\theta}r, & (2.213) \\ &= \exp\{l'(\boldsymbol{\theta}_k)\}\pi_k(\boldsymbol{\theta}_k), & (2.214) \\ l'(\boldsymbol{\theta}_k) &= \log p(\boldsymbol{f}_k|\boldsymbol{\theta}_k), & (2.215) \end{aligned}$$

where $P(Y_t|M_k)$ is the probability of the data given the model M_k. In the notation below, we write M_k, $\boldsymbol{f}_k$, $\boldsymbol{\theta}_k$, ν_k and $\pi_k(\cdot)$ as M, $\boldsymbol{f}$, $\boldsymbol{\theta}$, ν and $\pi(\cdot)$ for brevity. Using the Taylor expansion,

$l'(\boldsymbol{\theta})$ is extended around maximum likelihood estimator $\hat{\boldsymbol{\theta}}$ as follows.

$$l'(\boldsymbol{\theta}) = l'(\hat{\boldsymbol{\theta}}) - \frac{N}{2}(\boldsymbol{\theta}-\hat{\boldsymbol{\theta}})' J(\hat{\boldsymbol{\theta}})(\boldsymbol{\theta}-\hat{\boldsymbol{\theta}}) + \cdots, \tag{2.216}$$

$$J(\hat{\boldsymbol{\theta}}) = -\frac{1}{N}\frac{\partial^2 \log p(\boldsymbol{f}_k|\boldsymbol{\theta}_k)}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}'}|_{\hat{\boldsymbol{\theta}}}, \tag{2.217}$$

where N is the amount of data. As same as the above equations, $\pi(\boldsymbol{\theta})$ is approximated by the Taylor expansion as

$$\pi(\boldsymbol{\theta}) = \pi(\hat{\boldsymbol{\theta}}) + (\boldsymbol{\theta}-\hat{\boldsymbol{\theta}})'\frac{\partial\pi(\boldsymbol{\theta})}{\partial\theta}|_{\hat{\boldsymbol{\theta}}}\cdots. \tag{2.218}$$

Furthermore, by substituting Eq. (2.216) and (2.218) to (2.214), we can derive the following equations,

$$\begin{aligned} P(Y_t|M) &= \int \exp\{l'(\hat{\boldsymbol{\theta}}) - \frac{N}{2}(\boldsymbol{\theta}-\hat{\boldsymbol{\theta}})' J(\hat{\boldsymbol{\theta}})(\boldsymbol{\theta}-\hat{\boldsymbol{\theta}}) + \cdots\}\{\pi(\hat{\boldsymbol{\theta}}) + (\boldsymbol{\theta}-\hat{\boldsymbol{\theta}})'\frac{\partial\pi(\boldsymbol{\theta})}{\partial\theta}|_{\hat{\boldsymbol{\theta}}}\cdots\}, \\ &\simeq \exp\{l'(\hat{\boldsymbol{\theta}})\}\pi(\hat{\boldsymbol{\theta}})\int \exp\{-\frac{N}{2}(\boldsymbol{\theta}-\hat{\boldsymbol{\theta}})^T J(\hat{\boldsymbol{\theta}})(\boldsymbol{\theta}-\hat{\boldsymbol{\theta}})\}d\boldsymbol{\theta}, \qquad (2.219) \\ &\simeq \exp\{l'(\hat{\boldsymbol{\theta}})\}\pi(\hat{\boldsymbol{\theta}})(2\pi)^{\frac{\nu}{2}} n^{-\frac{\nu}{2}}|J(\hat{\boldsymbol{\theta}})|^{-\frac{1}{2}} \quad (N\to\infty), \qquad (2.220) \end{aligned}$$

by using

$$\int(\boldsymbol{\theta}-\hat{\boldsymbol{\theta}})\exp\{-\frac{N}{2}(\boldsymbol{\theta}-\hat{\boldsymbol{\theta}})^T J(\hat{\boldsymbol{\theta}})(\boldsymbol{\theta}-\hat{\boldsymbol{\theta}})\}d\boldsymbol{\theta} = 0, \tag{2.221}$$

and $(\hat{\boldsymbol{\theta}}-\boldsymbol{\theta})$ converges to 0 when N becomes infinity. The order of the convergence is $\mathcal{O}_\nu(N^{-\frac{1}{2}})$. Taking the logarithm of Eq. (2.220), we can get

$$\begin{aligned} -2\log P(Y_t|M) &= -2\log\exp\{l'(\boldsymbol{\theta})\}\pi(\boldsymbol{\theta}), \qquad (2.222) \\ &= -2l'(\hat{\boldsymbol{\theta}}) + \nu\log N + \log|J(\hat{\boldsymbol{\theta}})| - \nu\log(2\pi) - 2\log\pi(\hat{\boldsymbol{\theta}}), \qquad (2.223) \\ &\simeq -2l'(\hat{\boldsymbol{\theta}}) + \nu\log N. \qquad (2.224) \end{aligned}$$

Eq. (2.224) equals to BIC.

2.5 Appendix

2.5.1 A.1 Minimum Variance Estimator

Let X and Y be random variables, y be a real value of Y. Then, we define the minimum variance estimator $\hat{\zeta}$ as

$$\hat{\zeta} = \arg\min_{\zeta} E_X[\| X - \hat{\zeta} \|^2 | Y = y]. \tag{2.225}$$

$\hat{\zeta}$ is obtained as

$$\hat{\zeta} = E_X[X|Y=y] = \int_{-\infty}^{\infty} XP(X|Y=y)dX, \tag{2.226}$$

and

$$E_X[\| X - \hat{\zeta} \|^2 |Y=y] = \mathrm{tr}\{E_X[(X-\hat{\zeta})(X-\hat{\zeta})'|Y=y]\}, \tag{2.227}$$

$$= E_X[\| X \|^2 |Y=y] - \| \hat{\zeta} \|^2 . \tag{2.228}$$

(Proof)

$$E_X[\| X - \hat{\zeta} \|^2 |Y=y] = \int_{-\infty}^{\infty} (X-\hat{\zeta})'(X-\hat{\zeta})P(X|Y=y)dX, \tag{2.229}$$

$$= \int_{-\infty}^{\infty} X'XP(X|Y=y)dX - 2\zeta' \int_{-\infty}^{\infty} XP(X|Y=y)dX + \zeta'\zeta, \tag{2.230}$$

$$= \int_{-\infty}^{\infty} X'XP(X|Y=y)dX,$$
$$+ \{\zeta - \int_{-\infty}^{\infty} XP(X|Y=y)dX\}'\{\zeta - \int_{-\infty}^{\infty} XP(X|Y=y)dX\}$$
$$- \| \int_{-\infty}^{\infty} XP(X|Y=y)dX \|^2 \tag{2.231}$$

$$= E_X[\| X \|^2 |Y=y] + \{\zeta - \int_{-\infty}^{\infty} XP(X|Y=y)dX\}^2 - \| E_X[X|Y=y] \|^2 . \tag{2.232}$$

In the last equation, since only the second term includes ζ, we obtain $\hat{\zeta} = E_X[X|Y=y]$ that minimizes $\{\zeta - \int_{-\infty}^{\infty} XP(X|Y=y)dX\}^2 = 0$.

2.5.2 A.2 Conditional Distribution Minimizing Mean Square Errors

Lest X and Y be random variables according to Gaussian distributions Then, set $E_X[X] = \bar{X}$, $E_Y[Y] = \bar{Y}$, $V_X[X] = \Sigma_X$, $V_Y[Y] = \Sigma_Y$ and $Cov[X,Y] = \Sigma_{XY}$. We have

$$\hat{\zeta} = E_X[X|Y=y] = \bar{X} + \Sigma_{XY}\Sigma_Y^{-1}(y - \bar{Y}), \tag{2.233}$$

$$E_X[(X-\hat{\zeta})(X-\hat{\zeta})'|Y=y] = \Sigma_X - \Sigma_{XY}\Sigma_Y^{-1}\Sigma_{YX}. \tag{2.234}$$

(Proof) Set

$$Z = \begin{pmatrix} X \\ Y \end{pmatrix}, \quad (X \in R^p, Y \in R^q). \tag{2.235}$$

Then, $\bar{Z} = E_Z[Z]$ and $V_Z[Z] = \Sigma_Z$ can be described as

$$\bar{Z} = \begin{pmatrix} \bar{X} \\ \bar{Y} \end{pmatrix}, \tag{2.236}$$

$$\Sigma_Z = \begin{pmatrix} \Sigma_X & \Sigma_{XY} \\ \Sigma_{YX} & \Sigma_Y \end{pmatrix}. \tag{2.237}$$

Assume that Σ is a positive definite and X and Y are multivariate normal distributions. Then, we can obtain

$$P(X|Y) = \frac{P(X,Y)}{P(Y)} = \frac{\parallel \Sigma_Y \parallel^{\frac{1}{2}} \exp\{-\frac{1}{2}(X-\bar{X}, Y-\bar{Y})'\Sigma_Z^{-1}(X-\bar{X}, Y-\bar{Y})\}}{(2\pi)^{\frac{p}{2}} \parallel \Sigma_Z \parallel^{\frac{1}{2}} \exp\{-\frac{1}{2}(Y-\bar{Y})'\Sigma_Y^{-1}(Y-\bar{Y})\}}. \tag{2.238}$$

We have

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix} = \begin{pmatrix} A & 0 \\ C & I \end{pmatrix} \begin{pmatrix} I & A^{-1}B \\ 0 & D - CA^{-1}B \end{pmatrix}, \tag{2.239}$$

$$= \begin{pmatrix} I & B \\ 0 & D \end{pmatrix} \begin{pmatrix} A - BD^{-1}C & 0 \\ D^{-1}C & I \end{pmatrix}. \tag{2.240}$$

From Eq. (2.240), we can obtain

$$\parallel \Sigma_Z \parallel = \parallel \Sigma_X - \Sigma_{XY}\Sigma_Y^{-1}\Sigma_{YX} \parallel \cdot \parallel \Sigma_Y \parallel . \tag{2.241}$$

Next, Σ_Z^{-1} can be obtained as follows.

$$\begin{pmatrix} I & -\Sigma_{YX}\Sigma_Y^{-1} \\ 0 & I \end{pmatrix} \Sigma_Z \begin{pmatrix} I & 0 \\ -\Sigma_Y^{-1}\Sigma_{XY}' & I \end{pmatrix} = \begin{pmatrix} \Sigma_X - \Sigma_{XY}\Sigma_Y^{-1}\Sigma_{YX} & 0 \\ 0 & \Sigma_Y \end{pmatrix}, \tag{2.242}$$

$$\Longleftrightarrow \Sigma_Z = \begin{pmatrix} I & -\Sigma_{XY}\Sigma_Y^{-1} \\ 0 & I \end{pmatrix}^{-1} \begin{pmatrix} \Sigma_X - \Sigma_{XY}\Sigma_Y^{-1}\Sigma_{YX} & 0 \\ 0 & \Sigma_Y \end{pmatrix} \begin{pmatrix} I & 0 \\ -\Sigma_Y^{-1}\Sigma_{XY}' & I \end{pmatrix}^{-1}, \tag{2.243}$$

$$\Longleftrightarrow \Sigma_Z^{-1} = \begin{pmatrix} I & 0 \\ -\Sigma_Y^{-1}\Sigma_{XY}' & I \end{pmatrix} \begin{pmatrix} (\Sigma_X - \Sigma_{XY}\Sigma_Y^{-1}\Sigma_{YX})^{-1} & 0 \\ 0 & \Sigma_Y^{-1} \end{pmatrix} \begin{pmatrix} I & -\Sigma_{XY}\Sigma_Y^{-1} \\ 0 & I \end{pmatrix}. \tag{2.244}$$

Finally, by substituting Eqs. (2.241) and (2.244) to Eq. (2.238), we can derive

$$P(X|Y) = \frac{1}{(2\pi)^{\frac{p}{2}} \parallel \Sigma_X - \Sigma_{XY}\Sigma_Y^{-1}\Sigma_{YX} \parallel^{\frac{1}{2}}} \exp\{-\frac{1}{2}(X-\bar{X})'(\Sigma_X - \Sigma_{XY}\Sigma_Y^{-1}\Sigma_{YX})^{-1}(X-\bar{X})\}. \tag{2.245}$$

2.5.3 A.3 Projection

Define the probability space Ω, its subspace A and B ($A \supset B$) and finitely additive measures ω on Ω. We consider the optimal approximation on B of the random variable $X(\omega)$ on A as $\tilde{X}(\omega)$.

Then, we have

$$\tilde{X}(\omega) = Proj_B(X(\omega)) = E[X(\omega)|B]. \tag{2.246}$$

(Proof) In the space of square-summable sequences $L^2(\Omega, A, \omega) \supset L^2(\Omega, B, \omega)$, which are also known as a Hilbert space, the optimal basis functions on $L^2(\Omega, B, \omega)$ of $X(\omega)$ are represented by orthographic projection as

$$\tilde{X}(\omega) = Proj_B(X(\omega)). \tag{2.247}$$

Since $X(\omega) - \tilde{X}(\omega)$ crosses $L^2(\Omega, B, \omega)$ at right angles, we have

$$\int_B (X(\omega) - \tilde{X}(\omega))dP(\omega) = \left\langle X(\omega) - \tilde{X}(\omega), \chi(B) \right\rangle, \tag{2.248}$$

$$= 0, \tag{2.249}$$

$$\iff \int_B X(\omega)dP(\omega) = \int_B \tilde{X}(\omega)dP(\omega), \tag{2.250}$$

$$\iff E[X(\omega)|B] = \tilde{X}(\omega), \tag{2.251}$$

where $\langle X, Y \rangle$ is the inner product between X and Y, and $\chi(B)$ is the measurable set of B.

2.5.4 A.4 Matrix Transformation and Differentiation

Let $X = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_p)'$, $\boldsymbol{x}_t = (x_{t1}, \ldots, x_{tk})'$, $R = diag(r_1, \ldots, r_p)$ and $A \in R^{k \times k}$ where $A = A'$. Then, we have

$$\frac{\partial X'AX}{\partial X} = \frac{\partial \mathrm{tr}(AXX')}{\partial X} = (A + A')X = 2AX, \tag{2.252}$$

and

$$\frac{\partial \mathrm{tr}(RXAX')}{\partial X} = \frac{\partial \sum_{i=1}^{p} r_i x_i' A x_i}{\partial X}, \tag{2.253}$$

$$= \begin{pmatrix} \frac{\partial \sum_{i=1}^{p} r_i x_i' A x_i}{\partial x_{11}} & \cdots & \frac{\partial \sum_{i=1}^{p} r_i x_i' A x_i}{\partial x_{1k}}, \\ \vdots & \ddots & \vdots \\ \frac{\partial \sum_{i=1}^{p} r_i x_i' A x_i}{\partial x_{p1}} & \cdots & \frac{\partial \sum_{i=1}^{p} r_i x_i' A x_i}{\partial x_{pk}} \end{pmatrix} \tag{2.254}$$

$$= \begin{pmatrix} \frac{\partial r_1 x_1' A x_1}{\partial x_{11}} & \cdots & \frac{\partial r_1 x_1' A x_1}{\partial x_{1k}} \\ \vdots & \ddots & \vdots \\ \frac{\partial r_p x_p' A x_p}{\partial x_{p1}} & \cdots & \frac{\partial r_p x_p' A x_p}{\partial x_{pk}} \end{pmatrix}$$

$$= R \begin{pmatrix} \frac{\partial x_1' A x_1}{\partial x_{11}} & \cdots & \frac{\partial x_1' A x_1}{\partial x_{1k}}, \\ \vdots & \ddots & \vdots \\ \frac{\partial x_p' A x_p}{\partial x_{p1}} & \cdots & \frac{\partial x_p' A x_p}{\partial x_{pk}} \end{pmatrix} \tag{2.255}$$

$$= R \frac{\partial \mathrm{tr}(XAX')}{\partial X}.$$

Chapter 3

Inference of Gene Regulatory Networks Incorporating Multi-Source Biological Knowledge via a State Space Model with L1 Regularization

3.1 Background

Transcriptional regulation, which is controlled by several factors, plays essential roles to sustain complex biological systems in cells. Thus, identifying the structure and dynamics of such regulation can facilitate recognition of and control over systems for many practical purposes, *e.g.*, treatment of diseases. To accomplish this, many mathematical methods have been developed for the analysis of high-throughput biological data, *e.g.*, time-course microarray data [30, 45, 70]. In addition, recent technological advances have facilitated experimental discoveries, *e.g.*, DNA-protein interactions and the pharmacogenomics of chemical compounds. These contributions have allowed the knowledge of GRNs to accumulate.

For elucidation of GRN dynamics, time-course observational data have been generally used. Currently, one strategy to elucidate transcriptional regulation using observational data is to apply ODE (or SDE)-based approach, which can represent the dynamic behavior of biomolecular reactions based on biologically reliable models, *e.g.*, the Michaelis-Menten equation [91] or the S-system [92], which are described by differential equations. Thus, this approach can recapitulate the complex dynamic behavior of biological systems [62, 87]. In this approach, several methods have been proposed to infer regulatory structures [42, 81], to reproduce the dynamic behavior of biological systems recorded in the literature [77,79,83,85] and also to improve literature-recorded pathways so as to be consistent with the data [40]. However, nonlinearity of the system results

in an analytically intractable problem of estimating the parameter values that minimize loss function with updating simulated results. Thus, under this statistically efficient paradigm [8], this approach cannot be applied to ten or more genes to infer regulatory structures if the missing information is extensive [79].

In contrast, a statistical model-based approach using highly abstracted models, *e.g.*, Bayesian networks [29, 54, 120] and the state space model [9, 12, 43, 86], has been successfully applied to infer the structure of transcriptional regulation from biological observational data. Because these methods simply describe biological systems, hundreds of genes can be handled computationally with ease. Whereas methods relying purely on data need to consider all possibilities of transcriptional regulation, some studies have further incorporated other information, *e.g.*, PINs, literature-recorded pathways and TF information [7, 20, 22, 37, 88]. Although these methods can infer relationships among hundreds of genes simultaneously, high levels of abstraction can also generate false regulations that are difficult to interpret biologically. Thus, when several tens of genes are handled with partially understood relationships, highly abstract models can be insufficient to represent biological systems. In this case, there is an urgent need for a method that can infer system dynamics and the structure of GRNs based on a model with a low abstraction that can emulate the dynamics of ODE-based gene regulatory models incorporating existing biological knowledge.

We propose a novel method for inference of GRNs based on a newly developed model that uses a VAR-SSM [21,43,60]. The model is a type of state space models constructed from a typical gene regulatory system consisting of a synthesis process, a degradation process and regulatory effects by other genes within a linear Gaussian model. The method can infer the dynamic behavior of gene expression profiles and the regulatory structure for several tens of genes by assimilating time-course observational data. Furthermore, the method is capable of integrating the existing biological knowledge, *e.g.*, literature-recorded pathways and intracellular kinetics/dynamics of chemical compounds, and can deal with even non-equally spaced time-course observational data. A regulatory structure is inferred by maximization of the $L1$ regularized likelihood. To this end, we developed a new algorithm to obtain active sets of parameters and estimate a maximizer of the $L1$ regularized likelihood using the EM algorithm.

To demonstrate its effectiveness, we compared this method to a SSM [43], a general VAR model using LARS-LASSO algorithm [23], GeneNet [80, 94] based on an empirical graphical Gaussian model (GGM), dynamic Bayesian networks using first order conditional dependencies [63], GLASSO [29] based on sparse GGM and the mutual information-based network inference algorithms: ARACNE [70], CLR [26] and MRNET [75] by implementing artificial simulation models. The first two observational datasets are generated by two simulation models representing pharmacogenomic pathways [5, 115], including drug kinetics/dynamics, described by difference and differential equations, respectively. These pathways are initiated by the drug stimulation and observational data are obtained as non-equally spaced time-course data. The next observational dataset is generated by GeneNetWaver [69, 95] using a yeast network that is a part of Dialogue for Reverse Engineering Assessments and Methods (DREAM) 4 challenge.

As an application example, we applied the proposed method to corticosteroid pharmacogenomics in rat skeletal muscle [5, 100, 115]. Because this system has been investigated previously through biological experiments, corticosteroid kinetics/dynamics and the related genes are already partly elucidated. Therefore, we incorporated time-course mRNA expression data (observational data), candidate genes/pathways related to corticosteroids, intracellular corticosteroid kinetics/dynamics and, additionally, TF information from ITFP (Integrated Transcription Factor Platform) [122]. As in the simulation experiment, the observational data were obtained as non-equally spaced time-course data (GSE490) after stimulating rat skeletal muscle with corticosteroid. Consequently, we propose candidate pathways for extensions of corticosteroid-related pathways and their simulation dynamics in the presence of corticosteroid.

3.2 Methods

3.2.1 Linear Description of Biological Systems

For gene regulatory systems, we consider a general hill function-based model of transcriptional control, in which each gene has a synthesis process (regulated by other factors) and a degradation process, described by a differential equation [18, 24]. Let $x_n(t)$ be a time-dependent function representing the abundance of the nth (n =1, ...,N) mRNA in a cell, where t means time. Further, we consider subsets of $\{1, \ldots, N\}$, $\mathcal{N}_1$ and $\mathcal{N}_2$ ($\mathcal{N}_1 \oplus \mathcal{N}_2 = \{1, \ldots, N\}$), whose regulatory functions are described by two different forms [41, 47, 115]. Then, the time-evolution of $x_n(t)$ is represented by

$$\frac{d}{dt}x_n(t) = \prod_{k=1}^{N}\{1 + \phi_{n,k}(x_k(t))\} \cdot u_n - x_n(t) \cdot d_n, \quad n \in \mathcal{N}_1, \tag{3.1}$$

$$\frac{d}{dt}x_n(t) = \{1 + \sum_{k=1}^{N}\phi_{n,k}(x_k(t))\} \cdot u_n - x_n(t) \cdot d_n, \quad n \in \mathcal{N}_2, \tag{3.2}$$

where $\phi_{n,k}$ represents the regulatory effect of the kth gene on the nth gene as a hill-function, $u_n > 0$ and $d_n > 0$ are the synthesis and degradation rates of the mRNA, respectively. For example, in a previous pharmacogenomic study [115] that is analyzed in the results section, $\phi_{n,k}(x_k(t))$ was represented by

$$\phi_{n,k}(x_k(t)) = \frac{\alpha_{n,k} \cdot x_k(t)^{\gamma_{n,k}}}{\beta_{n,k}^{\gamma_{n,k}} + x_k(t)^{\gamma_{n,k}}}, \tag{3.3}$$

where $\alpha_{n,k}$, $\beta_{n,k}$ and $\gamma_{n,k}$ are tuning parameters.

In inferring the regulatory structure of GRNs consisting of several tens of genes, hill-function based differential equations, *e.g.*, Eqs. (3.1) and (3.2), become intractable. Therefore, as same as the previous researches [9, 12, 18, 37, 43, 60, 81, 86, 119], we postulate discretized and linearized gene regulatory systems instead of using these equations. Thus, linear functions are utilized

for representing the regulatory effects. The influence of these simplifications was discussed in the previous researches [18, 24] and we also assess the performance of inferring the regulatory structure described by Eqs. (3.1) and (3.2) based on simplified models in the results section. Furthermore, we assume that biological processes should include the effects by noise [14]. Let $\boldsymbol{x}_t = (x_{t,1}, \ldots, x_{t,n}, \ldots, x_{t,N})'$ be a series of N dimensional vectors containing expression levels of N genes at the tth time point. Then, we consider a gene regulatory system represented by

$$x_{t+\Delta t,n} - x_{t,n} = \{(1 + \boldsymbol{a}_n' \boldsymbol{x}_t)u_n - x_{t,n} \cdot d_n + v_{t,n}\}\Delta t, \tag{3.4}$$

where $\boldsymbol{a}_n = (a_{n,1}, \ldots, a_{n,N})'$ is an N-dimensional vector including regulatory effects on the nth gene by other genes, $v_{t,n}$ is the effects by noise at the tth time point, and Δt indicates a minute displacement. Then, a VAR model for GRNs simulation can be constructed.

In constructing gene regulatory models, we make an assumption that observational data are measured with observational noise. Under this assumption, to separately handle a system model (*i.e.*, Eq. (3.4)) and biological observational data, we utilize a state space representation [7, 43, 60, 66, 83]. Here, a minimum observational time step and Δt are usually handled as 1 for reducing computational cost, however, we can set any value for Δt less than a minimum observational time step. Therefore, we evaluated the influence of changing Δt in the results section and describe the case of $\Delta t = 1$ in the following for simplicity. Consequently, we consider a model described by

$$\boldsymbol{x}_{t+1} = A\boldsymbol{x}_t + \boldsymbol{u} + \boldsymbol{d}_t + \boldsymbol{v}_{t+1}, \tag{3.5}$$

$$\boldsymbol{y}_t = \boldsymbol{x}_t + \boldsymbol{w}_t, \tag{3.6}$$

$$\boldsymbol{d}_t = ((1 - d_1) \cdot x_{t,1}, \ldots, (1 - d_N) \cdot x_{t,N})', \tag{3.7}$$

where $A = (\boldsymbol{a}_1, \ldots, \boldsymbol{a}_N)'$ is an $N \times N$ matrix representing regulation among genes, $\boldsymbol{x}_t$ is an N-dimensional hidden state variable, $\boldsymbol{u} = (u_1, \ldots, u_N)'$ is an N-dimensional vector including synthesis rates, $\boldsymbol{y}_t \in R^N$ is a series of vectors containing observed expression levels of N genes at the tth time point and $\boldsymbol{w}_t \in R^N$ is observational noise. Here, we define a set of all points of time $\mathcal{T}$ ($t \in \mathcal{T}$), consisting of the observed time set $\mathcal{T}_{obs}$ ($\mathcal{T}_{obs} \subset \mathcal{T}$). We set system noise $\boldsymbol{v}_t \sim N_N(\boldsymbol{0}_N, Q)$ and observation noise $\boldsymbol{w}_t \sim N_N(\boldsymbol{0}_N, R)$, where Q and R are $N \times N$ diagonal matrices. The initial state vector $\boldsymbol{x}_0$ is assumed to be a Gaussian random vector with mean vector $\boldsymbol{\mu}_0$ and covariance matrix Σ_0, *i.e.*, $\boldsymbol{x}_0 \sim N_N(\boldsymbol{\mu}_0, \Sigma_0)$. Note that $\boldsymbol{u}$ and $\boldsymbol{d}$ must be dense vectors; nevertheless, A should be a sparse matrix, and activation and repression correspond to positive and negative values of $a_{n,k}$, respectively.

Contrary to the derivation of Eq. (3.5), in previous linear state space models for GRN analysis [43, 60], a simulation model was constructed as

$$\boldsymbol{x}_{t+1} = F\boldsymbol{x}_t + \boldsymbol{v}_{t+1}, \tag{3.8}$$

where F is an $N \times N$ matrix in which the nth row and kth column element is represented by

$$f_{n,k} = \begin{cases} 1 - d_n + a_{n,k} & (n = k) \\ a_{n,k} & (n \neq k) \end{cases}. \tag{3.9}$$

In this model, $\boldsymbol{u}$ is removed by shifting the average of the observed time-course data for each element to 0, *i.e.*, $\sum_{t \in \mathcal{T}_{obs}} y_{t,n} = 0$ for $n = 1, \ldots, N$, where $y_{t,n}$ is the nth row element of $\boldsymbol{y}_t$, as a normalization procedure. However, this model may cause marked difficulty in estimating gene regulatory relationships if the observed time-course includes a steady state. Fig. 3.1 exemplifies such a situation.

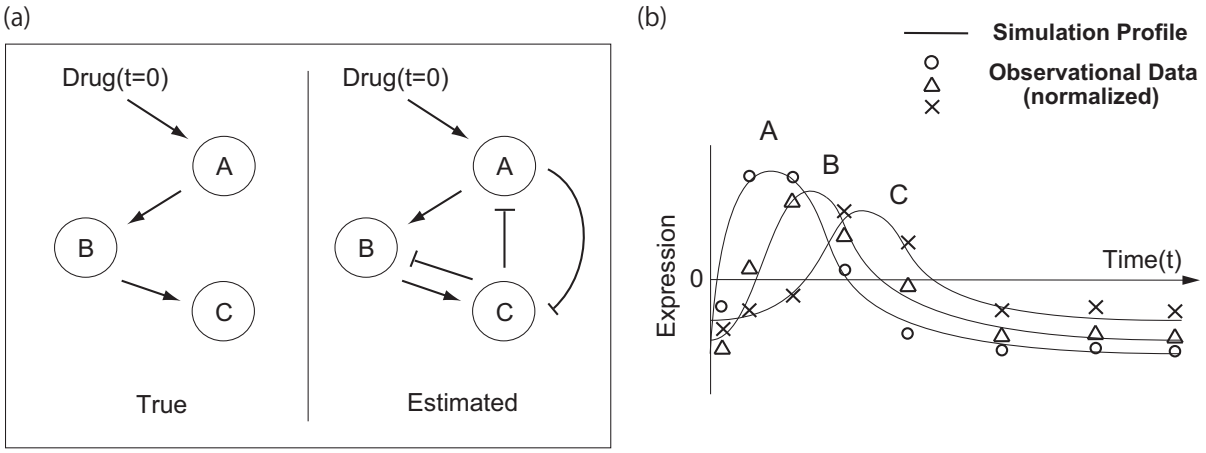

Figure 3.1: The problem of deleting a term representing a synthesis rate. A toy model indicating the problem of deleting a synthesis rate $\boldsymbol{u}$ by shifting an average of observed time-course data for each element to 0, *i.e.*, $\sum_{t \in \mathcal{T}_{obs}} y_{t,n} = 0$ for $n = 1, \ldots, N$ as a normalization procedure. The true network and the adjusted data are illustrated in the left panel in (a) and (b), respectively. As shown in the right panel in (a), some false positive edges are possibly estimated in comparison to the true relationships.

Fig. 3.1 shows a small pathway consisting of three genes (left panel in Fig. 3.1 (a)) and the averages of the observed time-course data for each element are shifted to 0 (Fig. 3.1(b)). By applying Eq. (3.8) to the observed data, we expect to obtain three false edges added to the true pathway (right panel in Fig. 3.1(a)) because nodes must retain a constant steady state regardless of their negative steady state values and positive regulation from negative nodes. In some cases, such additional false regulation possibly hide true regulation. The above result encourages us to use a model explicitly implementing terms to represent a steady state of gene expressions to estimate gene regulatory relationships precisely. Furthermore, in using Eq. (3.8), when elements of F are regularized to be selected non-zero elements, even $1 - d_n$ is regularized and $f_{n,n}$ can be zero. To penalize the regulatory effect $a_{n,k}$ only, A and $\boldsymbol{d}_t$ are separately described in our proposed model.

3.2.2 Incorporation of Biomolecules Affecting Biological Systems

When simulating the dynamic behavior of GRNs including biomolecules that cannot be represented by $\boldsymbol{x}_t$ and can affect biological systems, *e.g.*, corticosteroids in corticosteroid-stimulated GRNs, we should consider the concentration of such biomolecules. For these cases, we remodel Eq. (3.5) to add a term representing the concentration of such biomolecules as

$$\boldsymbol{x}_{t+1} = A\boldsymbol{x}_t + \boldsymbol{u} + \boldsymbol{d}_t + G\boldsymbol{z}_t + \boldsymbol{v}_{t+1}, \tag{3.10}$$

where $\boldsymbol{z}_t$ is an M-dimensional vector containing the concentration of the biomolecules at the tth time point, $G = (\boldsymbol{g}_1, \ldots, \boldsymbol{g}_N)'$ is an $N \times M$ matrix and $\boldsymbol{g}_n = (g_{n,1}, \ldots, g_{n,M})'$ is an M-dimensional vector representing their regulatory effects on the nth gene. We consider the case that the concentration is known or can be simulated. In the results section, for an application example, we deal with corticosteroid drug pathways that have been well studied previously [5, 100, 115]; $\boldsymbol{z}_t$ is given the concentration of the intra-nuclear corticosteroid-receptor complex employed in Yao *et al.* [115].

3.2.3 State Space Model and Kalman Filter for Estimating the Hidden State

Recently, many types of state space models have been proposed and applied in the context of systems biology [9, 12, 43, 66, 83, 88, 105]. They are roughly divided into two major classes, *i.e.*, linear and nonlinear models. In using linear state space models, posterior probability densities of the hidden state can be obtained as Gaussian distributions and the optimal mean and covariance matrices can be analytically calculated by the Kalman filter algorithm [52, 101]. In contrast, for nonlinear state space models, because the analytical form can be intractable, several extensions of the Kalman filter algorithm, *e.g.*, extended Kalman filter [74], unscented Kalman filter [49, 51] and particle filter [57], which utilize approximation techniques, have been applied to obtain posterior probability densities of hidden state and parameters [7, 66, 67, 83, 105]. In using linear state space models [9,12,43,86], the main concern is to infer causal relationships among genes, for which regulatory structure is assumed to be sparse, *i.e.*, genes are regulated by only a few specific regulators. Imposing such a sparse constraint to regression approaches is a general problem, but for state space models to simultaneously estimate optimal hidden state and parameter values (including penalization parameters), it is not a trivial problem [21, 23, 37, 60, 99]. Then, for example, a sparse regulatory structure was extracted by statistical tests after estimating parameter values [43]. In this article, under the framework of a state space representation of a VAR model, we intend to infer the parameter values and the hidden state maximizing prediction ability for observational data with a sparse regulatory structure. For this purpose, we apply the regularized EM algorithm [19,46,64,104] in the next subsection and the conditional expectations of hidden state are given by using the Kalman filter algorithm.

3.2.3.1 Kalman Filter Algorithm for VAR-SSM

Let $\boldsymbol{U}_t$ be the sum of $\boldsymbol{u}$ and $G\boldsymbol{z}_t$. For simplicity, we here use F in Eq. (3.8) rather than A. The prediction, filtering, and smoothing of the Kalman filter are calculated by the following formulas:

- Prediction:

$$\boldsymbol{x}_{t|t-1} = F\boldsymbol{x}_{t-1|t-1} + \boldsymbol{U}_{t-1}, \tag{3.11}$$
$$\Sigma_{t|t-1} = F\Sigma_{t-1|t-1}F' + Q, \tag{3.12}$$

- Filtering:

$$\boldsymbol{x}_{t|t} = \boldsymbol{x}_{t|t-1} + \Sigma_{t|t}R^{-1}(\boldsymbol{y}_t - \boldsymbol{x}_{t|t-1}), \tag{3.13}$$
$$\Sigma_{t|t} = (R^{-1} + \Sigma_{t|t-1}^{-1})^{-1}, \tag{3.14}$$

- Smoothing

$$\boldsymbol{x}_{t|T} = \boldsymbol{x}_{t|t} + J_t(\boldsymbol{x}_{t+1|T} - \boldsymbol{x}_{t+1|t}), \tag{3.15}$$
$$\Sigma_{t|T} = \Sigma_{t|t} + J_t(\Sigma_{t+1|T} - \Sigma_{t+1|t})J_t', \tag{3.16}$$
$$\Sigma_{t,t-1|T} = \Sigma_{t|t}J_{t-1}' + J_t(\Sigma_{t+1,t|T} - F\Sigma_{t|t})J_{t-1}', \tag{3.17}$$
$$J_t = \Sigma_{t|t}F'\Sigma_{t+1|t}^{-1}, \tag{3.18}$$

$$\Sigma_{T,T-1|T} = (I - \Sigma_{T|T}R^{-1})F\Sigma_{T-1|T-1}, \tag{3.19}$$

where $\mathrm{E}[\boldsymbol{x}_t]$ given $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_s$ is represented by $\boldsymbol{x}_{t|s}$ and $\mathrm{Var}[\boldsymbol{x}_t]$ given $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_s$ is represented by $\boldsymbol{\Sigma}_{t|s}$. To calculate an inverse of the $N \times N$ matrix, we use a matrix inversion theorem [60]. The derivations of KF are introduced in Chapter 2.

3.2.4 Maximum Likelihood Estimation Using the Regularized EM Algorithm with L1 Regularization

In biological systems, most genes are regulated by a few specific genes, *i.e.*, A and G can be sparse matrices. Thus, we applied $L1$ regularization to select effective sets of elements for A and G. Let $\{Y_T, X_T\}$ be the complete data set, where $Y_T = \{\boldsymbol{y}_1, \cdots, \boldsymbol{y}_T\}$ is the set of observed data and $X_T = \{\boldsymbol{x}_0, \cdots, \boldsymbol{x}_T\}$ is the set of state variables. Furthermore, let the probability densities $P(\boldsymbol{x}_0)$, $P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$ and $P(\boldsymbol{y}_t|\boldsymbol{x}_t)$ be the N-dimensional Gaussian distributions $N(\boldsymbol{\mu}_0, \Sigma_0)$, $N(F_{t-1}\boldsymbol{x}_{t-1} + U_{t-1}, Q)$ and $N(\boldsymbol{x}_t, R)$, respectively. Then joint likelihood for the complete data set is given by

$$P(Y_T, X_T; \boldsymbol{\theta}) = P(\boldsymbol{x}_0) \prod_{t \in \mathcal{T}} P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) \prod_{t \in \mathcal{T}_{obs}} P(\boldsymbol{y}_t|\boldsymbol{x}_t), \tag{3.20}$$

where $\boldsymbol{\theta} = \{A, \boldsymbol{u}, \boldsymbol{d}, G, Q, R, \boldsymbol{\mu}_0\}$. In this study, we used the regularized EM algorithm [19,46, 64,104] to search for the parameter vector $\boldsymbol{\theta}$ that maximizes $P(Y_T;\boldsymbol{\theta})$ under $L1$ regularization. The $L1$ regularized log-likelihood is given by

$$\log \int P(\boldsymbol{x}_0) \prod_{t\in\mathcal{T}} P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) \prod_{t\in\mathcal{T}_{obs}} P(\boldsymbol{y}_t|\boldsymbol{x}_t) d\boldsymbol{x}_0 \dots d\boldsymbol{x}_T - \sum_{n=1}^{N}\sum_{k=1}^{N} \lambda_n |A_{n,k}| - \sum_{n=1}^{N}\sum_{k=1}^{M} \lambda_n |G_{n,k}|, \tag{3.21}$$

where λ_n is the $L1$ regularization term for the nth row. In the EM algorithm, the conditional expectation of the joint log-likelihood of the complete data set

$$q(\boldsymbol{\theta}|\boldsymbol{\theta}_i) = E[\log P(Y_T, X_T|\boldsymbol{\theta})|Y_T, \boldsymbol{\theta}_i] - \sum_{n=1}^{N}\sum_{k=1}^{N} \lambda_n |A_{n,k}| - \sum_{n=1}^{N}\sum_{k=1}^{M} \lambda_n |G_{n,k}|, \tag{3.22}$$

is iteratively maximized with respect to $\boldsymbol{\theta}$ until convergence, where $\boldsymbol{\theta}_i$ is the parameter vector obtained at the ith (previous) iteration. Note that the convergence of the $L1$ regularized log-likelihood using the EM algorithm was guaranteed [46, 64, 104].

In the Expectation-step, $q(\boldsymbol{\theta}|\boldsymbol{\theta}_i)$ is calculated by

$$\begin{aligned}
q(\boldsymbol{\theta}|\boldsymbol{\theta}_i) = &-\frac{1}{2}\log \mid \Sigma_0 \mid -\frac{1}{2}\mathrm{tr}\{\Sigma_0^{-1}(\Sigma_{0|T} + (\boldsymbol{x}_{0|T} - \boldsymbol{\mu}_0)(\boldsymbol{x}_{0|T} - \boldsymbol{\mu}_0)')\} \\
&-\frac{T}{2}\log \mid Q \mid -\frac{1}{2}\mathrm{tr}\{Q^{-1}(V_t - V_{lag}F'FV_{lag}' + FV_{t-1}F' + FE_{t-1}G' + GE_{t-1}'F' \\
&- E_{lag}G' - GE_{lag} + GZG' + G\boldsymbol{z}\boldsymbol{u}' + \boldsymbol{u}\boldsymbol{z}'G' + F\boldsymbol{s}_{t-1}\boldsymbol{u}' + \boldsymbol{u}\boldsymbol{s}_{t-1}'F' - \boldsymbol{s}_t\boldsymbol{u}' - \boldsymbol{u}\boldsymbol{s}_t' + T\boldsymbol{u}\boldsymbol{u}')\} \\
&-\frac{T}{2}\log \mid R \mid -\frac{1}{2}\mathrm{tr}[R^{-1}\sum_{t=1}^{T}\{(\boldsymbol{y}_t - \boldsymbol{x}_{t|T})(\boldsymbol{y}_t - \boldsymbol{x}_{t|T})' + \Sigma_{t|T}'\}] \\
&- N(T+\frac{1}{2})\log 2\pi - \sum_{n=1}^{N}\sum_{k=1}^{N} \lambda_n |A_{n,k}| - \sum_{n=1}^{N}\sum_{k=1}^{M} \lambda_n |G_{n,k}|,
\end{aligned} \tag{3.23}$$

where

$$V_t = \sum_{t\in\mathcal{T}}^{T} (\Sigma_{t|T} + \boldsymbol{x}_{t|T}\boldsymbol{x}_{t|T}'), \tag{3.24}$$

$$V_{lag} = \sum_{t\in\mathcal{T}}^{T} (\Sigma_{t,t-1|T} + \boldsymbol{x}_{t|T}\boldsymbol{x}_{t-1|T}'), \tag{3.25}$$

$$V_{t-1} = \sum_{t\in\mathcal{T}}^{T} (\Sigma_{t-1|T} + \boldsymbol{x}_{t-1|T}\boldsymbol{x}_{t-1|T}'), \tag{3.26}$$

$$\boldsymbol{s}_t = \sum_{t\in\mathcal{T}}^{T} \boldsymbol{x}_{t|T}, \tag{3.27}$$

$$\boldsymbol{s}_{t-1} = \sum_{t\in\mathcal{T}}^{T} \boldsymbol{x}_{t-1|T}, \tag{3.28}$$

$$E_{lag} = \sum_{t\in\mathcal{T}}^{T} \boldsymbol{x}_{t|T}\boldsymbol{z}'_{t-1|T}, \tag{3.29}$$

$$E_{t-1} = \sum_{t\in\mathcal{T}}^{T} \boldsymbol{x}_{t-1|T}\boldsymbol{z}'_{t-1|T}, \tag{3.30}$$

$$\boldsymbol{z} = \sum_{t\in\mathcal{T}}^{T} \boldsymbol{z}_{t-1|T}, \tag{3.31}$$

$$Z = \sum_{t\in\mathcal{T}}^{T} \boldsymbol{z}_{t-1|T}\boldsymbol{z}'_{t-1|T}. \tag{3.32}$$

In the Maximization-step, $\boldsymbol{\theta}_i$ is updated to $\boldsymbol{\theta}_{i+1}$ to be $\boldsymbol{\theta}_{i+1} = \arg\max_{\boldsymbol{\theta}} q(\boldsymbol{\theta}|\boldsymbol{\theta}_i)$. Let $\boldsymbol{v}_{t,n}$, $\boldsymbol{v}_{lag,n}$, $\boldsymbol{v}_{t-1,n}$, $\boldsymbol{e}_{t,n}$ and $\boldsymbol{e}_{lag,n}$ set a transpose of nth row vector of V_t, V_{lag}, V_{t-1}, E_{lag} and E_{t-1}, respectively. Further, set $s_{t,n}$ and $s_{t-1,n}$ as an nth element of $\boldsymbol{s}_t$ and $\boldsymbol{s}_{t-1}$, and $v_{t,n,k}$ and $v_{t-1,n,k}$ as an nth row kth column element of V_t and V_{t-1}, respectively. Then, $\boldsymbol{\theta}$ is updated as

$$\begin{aligned}\boldsymbol{a}_n = \arg\min_{\boldsymbol{a}_n}\ &\{\boldsymbol{a}'_n V_{t-1}\boldsymbol{a}_n + 2(1-d_n)\boldsymbol{v}'_{t-1,n}\boldsymbol{a}_n - 2\boldsymbol{v}'_{lag,n}\boldsymbol{a}_n + 2u_n\boldsymbol{s}'_{t-1}\boldsymbol{a}_n \\ &+ 2\boldsymbol{g}'_n E'_{t-1}\boldsymbol{a}_n + 2q_n\sum_{k=1}^{N}\lambda_n|a_{n,k}|\},\end{aligned} \tag{3.33}$$

$$\boldsymbol{g}_n = \arg\min_{\boldsymbol{g}_n}\ \{\boldsymbol{g}'_n Z\boldsymbol{g}_n + 2\boldsymbol{f}'_n E_{t-1}\boldsymbol{g}_n - 2\boldsymbol{e}'_{lag,n}\boldsymbol{g}_n + 2u_n\boldsymbol{z}'\boldsymbol{g}_n + 2q_n\sum_{k=1}^{M}\lambda_n|g_{n,k}|\}, \tag{3.34}$$

$$d_n = 1 - \frac{v_{t,n,n} - u_n s_{t-1,n} - \boldsymbol{v}'_{t-1,n}\boldsymbol{a}_n - \boldsymbol{g}'_n\boldsymbol{e}_{lag,n}}{v_{t-1,n,n}}, \tag{3.35}$$

$$\boldsymbol{u} = \frac{\boldsymbol{s}_t - F\boldsymbol{s}_{t-1} - G\boldsymbol{z}}{T}, \tag{3.36}$$

$$\begin{aligned}Q = \frac{1}{T}(&V_t - V_{lag}F' - FV'_{lag} + FV_tF' + FE_{lag}G' + GE'_{lag}F' - E_tG' - GE_t + GZG' \\ &+ G\boldsymbol{z}\boldsymbol{u}' + \boldsymbol{u}\boldsymbol{z}'G' + F\boldsymbol{s}_{t-1}\boldsymbol{u}' + \boldsymbol{u}\boldsymbol{s}'_{t-1}F' - \boldsymbol{s}_t\boldsymbol{u}' - \boldsymbol{u}\boldsymbol{s}'_t + T\boldsymbol{u}\boldsymbol{u}'),\end{aligned} \tag{3.37}$$

$$R = \frac{1}{T}\sum_{t\in\mathcal{T}_{obs}}^{T}\{(\boldsymbol{y}_t - \boldsymbol{x}_{t|T})(\boldsymbol{y}_t - \boldsymbol{x}_{t|T})' + \Sigma_{t|T}\}, \tag{3.38}$$

where d_n is set 0 if $d_n < 0$.

In solving Eqs. (3.33) and (3.34), we further consider the non-zero/zero elements in A and G. Let $\mathcal{A} = \{\mathcal{A}_1, \ldots, \mathcal{A}_N\}$ and $\mathcal{G} = \{\mathcal{G}_1, \ldots, \mathcal{G}_N\}$, where $\mathcal{A}_n$ and $\mathcal{G}_n$ be active sets of elements for $\boldsymbol{a}_n$ and $\boldsymbol{g}_n$, *i.e.*, $\forall\{a_{n,k} \neq 0\} \in \mathcal{A}_n$ and $\forall\{g_{n,k} \neq 0\} \in \mathcal{G}_n$ for $k = 1, \ldots, N$, respectively. The descriptions of ${}^{\mathcal{A}_n}$ and ${}^{\mathcal{G}_n}$ stand for an $|\mathcal{A}_n| \times |\mathcal{A}_n|$ matrix or an $|\mathcal{A}_n|$ dimensional vector and a $|\mathcal{G}_n| \times |\mathcal{G}_n|$ matrix or a $|\mathcal{G}_n|$ dimensional vector, respectively. Then, Eqs. (3.33) and (3.34) are differentiated to satisfy

$$(3.33) \Leftrightarrow \boldsymbol{a}_n^{\mathcal{A}_n} = {V_{t-1}^{\mathcal{A}_n}}^{-1}(\boldsymbol{v}_{lag,n}^{\mathcal{A}_n} - (1-d_n)\boldsymbol{v}_{t-1,n}^{\mathcal{A}_n} - u_n \boldsymbol{s}_{t-1}^{\mathcal{A}_n} - E_{t-1}^{\mathcal{A}_n}\boldsymbol{g}_n^{\mathcal{A}_n} - q_n\lambda_n sign(\boldsymbol{a}_n^{\mathcal{A}_n})), \quad (3.39)$$

$$(3.34) \Leftrightarrow \boldsymbol{g}_n^{\mathcal{G}_n} = -{Z^{\mathcal{G}_n}}^{-1}(\boldsymbol{e}_{lag,n}^{\mathcal{G}_n} - E_{t-1}^{\mathcal{G}'_n}\boldsymbol{f}_n^{\mathcal{G}_n} - u_n \boldsymbol{z}^{\mathcal{G}_n} - q_n\lambda_n sign(\boldsymbol{g}_n^{\mathcal{G}_n})), \quad (3.40)$$

where *sign* means a sign vector consisting positive (+1) or negative (-1) values. These equations are derived as the same way explained in Chapter 2.

3.2.5 Parameter Optimization Algorithm with $L1$ Regularization

Because of the combination of the regularization terms and a state space representation, updating an element of $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_N)'$ influences the other active sets. Thus, it is difficult to select the optimal active sets $\mathcal{A}$ and $\mathcal{G}$, the values of $\boldsymbol{\theta}$ and $\boldsymbol{\lambda}$ at the same time. Therefore, we proposed a novel algorithm to separately update $\boldsymbol{\theta}$ and $\boldsymbol{\lambda}$ in each row as follows. In this algorithm, we consider candidates of active sets for $\mathcal{A}_n$ and $\mathcal{G}_n$ as $\tilde{\mathcal{A}}_n$ and $\tilde{\mathcal{G}}_n$, respectively. In the EM algorithm, we constraint that the active sets $\mathcal{A}_n$ and $\mathcal{G}_n$ can be selected from $\tilde{\mathcal{A}}_n$ and $\tilde{\mathcal{G}}_n$, respectively, *i.e.*, $\mathcal{A}_n \subseteqq \tilde{\mathcal{A}}_n$ and $\mathcal{G}_n \subseteqq \tilde{\mathcal{G}}_n$. In contrast to other algorithms using the $L1$ regularization such as least absolute shrinkage and selection operator (LASSO), we cannot obtain the global maximum/minimum of the objective function depending on initial parameter values. Then, we performed several trials starting from different initial values.

3.2.5.1 Algorithm

-Initial Settings

1. Set $\boldsymbol{\lambda} = \boldsymbol{0}$ and recursively update $\boldsymbol{\theta}$ to obtain $\boldsymbol{x}_{t|T}$ $(t = 1, \ldots, T)$ using the EM algorithm until convergence is attained. In this step, active sets $\mathcal{A}_n$ and $\mathcal{G}_n$ $(n = 1, \ldots, N)$ consist of all elements, *i.e.*, A and G become dense matrices, since the regularization terms can be neglected. Thus, the solution of the EM algorithm is directly obtained from Eqs. (3.33)-(3.38).

2. Set the maximum number of iterations to be i_{max}, the maximum number of regulatory edges for each gene to be k_{max} and $\boldsymbol{\lambda}$ to be sufficiently high to allow all elements of A and G to become 0, and $\tilde{\mathcal{A}}_n$ and $\tilde{\mathcal{G}}_n$ to be full. Alternatively, i_{max} can be set as a value when BIC [96, 112, 124], which are used to select the best model in this algorithms, is not updated through iterations and k_{max} can be set a sufficiently high value, *e.g.*, $\frac{N}{2}$. The BIC

score in this algorithm is defined as

$$BIC_{VARSSM} = -2\log L(Y_N|\boldsymbol{\theta}) + \mathrm{df}(\boldsymbol{\lambda}, \boldsymbol{\theta})\log\nu, \tag{3.41}$$

$$L(Y_N|\boldsymbol{\theta}) = \int P(\boldsymbol{x}_0) \prod_{t\in\mathcal{T}} P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) \prod_{t\in\mathcal{T}_{obs}} P(\boldsymbol{y}_t|\boldsymbol{x}_t) d\boldsymbol{x}_0 \ldots d\boldsymbol{x}_T, \tag{3.42}$$

where $\mathrm{df}(\boldsymbol{\lambda}, \boldsymbol{\theta})$ is the degree of freedom, *i.e.*, the number of active parameters [124], and ν is the number of samples. The derivation of BIC is briefly introduced in Chapter 2. Approaches for selecting appropriate model complexity and fit when maximizing regularized log-likelihood at some regularization parameter values were discussed previously [98, 104, 124].

3. Set $i = 1$ and recursively update $\{\boldsymbol{\lambda}, \mathcal{A}, \mathcal{G}, \tilde{\mathcal{A}} = \{\tilde{\mathcal{A}}_1, \ldots, \tilde{\mathcal{A}}_N\}, \tilde{\mathcal{G}} = \{\tilde{\mathcal{G}}_1, \ldots, \tilde{\mathcal{G}}_N\}, \boldsymbol{\theta}\}$ as follows. Note that, at $i = 1$, we fix $\boldsymbol{x}_{t|T}$ as the values obtained at Step 1, except for the updating elements indicated as n_{upd} in the next step. Thus, we only update the values of the parameters for the n_{upd}th row at $i = 1$.

-Main Routine

4. For $n_{upd} = 1, \ldots, N$

   a). Set $\tilde{\mathcal{A}}_{n_{upd}}$ and $\tilde{\mathcal{G}}_{n_{upd}}$ full and $\lambda_{n_{upd}}$ sufficiently high to allow all elements of $\boldsymbol{a}_{n_{upd}}$ and $\boldsymbol{g}_{n_{upd}}$ become $\mathbf{0}$. Through the following steps, fixing λ_n $(n \neq n_{upd})$, $\lambda_{n_{upd}}$ is gradually decreased to find an optimum $\lambda_{n_{upd}}$ for which the BIC score is minimized.

   b). Calculate conditional expectations using the Kalman filter.

   c). Update $\mathcal{A}$, $\mathcal{G}$, and $\boldsymbol{\theta}$ by Eqs. (3.33)-(3.40). Here, $\mathcal{A}_n$ and $\mathcal{G}_n$ of Eqs. (3.39)-(3.40) can be constructed from $\tilde{\mathcal{A}}_n$ and $\tilde{\mathcal{G}}_n$, respectively.

   d). Calculate the BIC score and decrease $\lambda_{n_{upd}}$ if the regularized log-likelihood of Eq. (3.21) is converged. Then, repeat from step (b) until the sum total of $\mathcal{A}_{n_{upd}}$ and $\mathcal{G}_{n_{upd}}$ becomes k_{max}.

   e). Set $\{\boldsymbol{\lambda}, \mathcal{A}, \mathcal{G}, \boldsymbol{\theta}\}$ as the value with the lowest BIC score obtained through the above described steps. Furthermore, set $\tilde{\mathcal{A}} \leftarrow \mathcal{A}$ and $\tilde{\mathcal{G}} \leftarrow \mathcal{G}$.

   f). Consider the set of all subsets of $\mathcal{A}_{n_{upd}}$ and $\mathcal{G}_{n_{upd}}$ as sub_A and sub_G, respectively. For all $s_A \in sub_A$ and $s_G \in sub_G$, setting $\tilde{\mathcal{A}}_{n_{upd}} \leftarrow s_A$ and $\tilde{\mathcal{G}}_{n_{upd}} \leftarrow s_G$, repeat steps 4(b) and (c), and then obtain the BIC scores of converged log-likelihood.

   g). Set $\{\mathcal{A}, \mathcal{G}, \boldsymbol{\theta}\}$ as the value with the lowest BIC score. Furthermore, $\tilde{\mathcal{A}} \leftarrow \mathcal{A}$ and $\tilde{\mathcal{G}} \leftarrow \mathcal{G}$.

5. Set $i \rightarrow i + 1$ and repeat from step 4 until i becomes i_{max}.

A conceptual view and a pseudo code of the algorithm are shown in Figure 3.2 and Algorithm 1, respectively. We should note that, since the active sets $\mathcal{A}$ and $\mathcal{G}$ obtained at step 4(e) may

not be the optimal ones for the selected $\boldsymbol{\lambda}$, *i.e.*, there can exist better ones having lower BIC scores for the selected $\boldsymbol{\lambda}$, the proposed algorithm further explores such better ones by evaluating subsets of the obtained active sets at step 4(e) through steps 4(f) and (g).

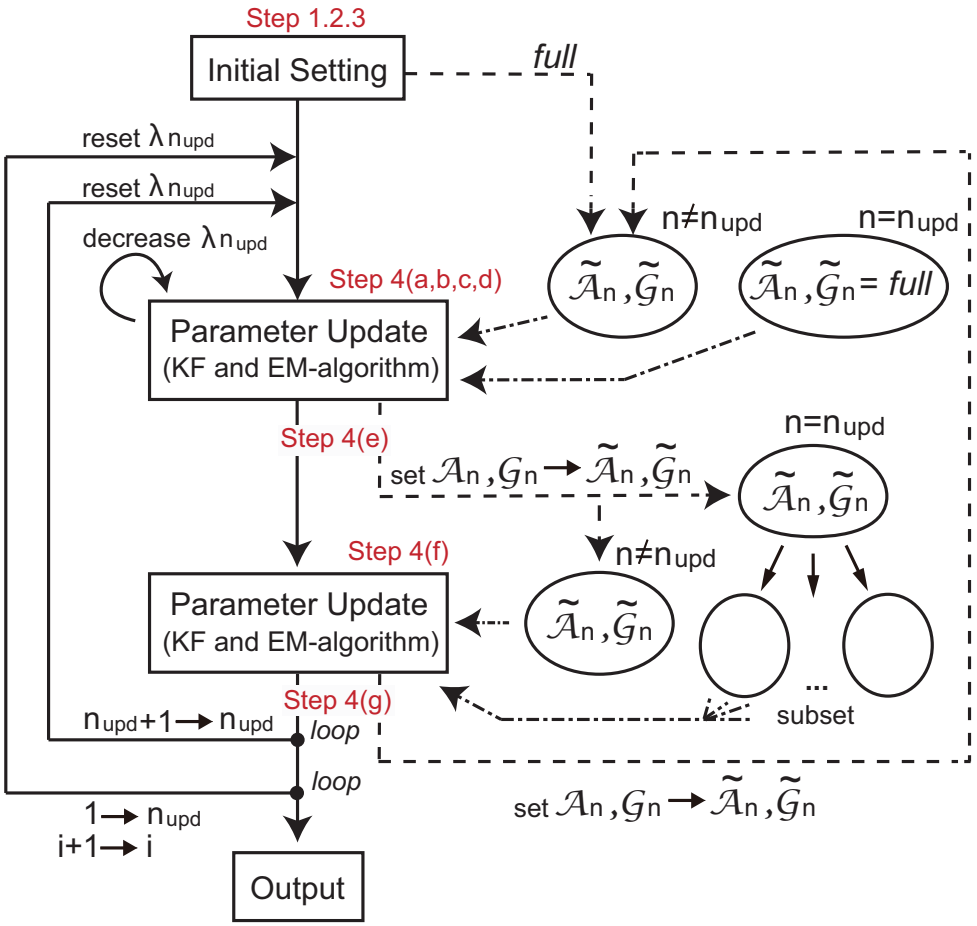

Figure 3.2: The conceptual view of the proposed algorithm. This figure illustrates a conceptual view of the proposed algorithm. The notations 'Step' correspond to those of the proposed algorithm. Solid, dashed and chain lines represent flowchart of the algorithm, setting the parameter values and active sets, and setting candidates of active sets used for selecting active sets.

Algorithm 1 A pseudo code of Main Routine (step 4 and 5) in the proposed algorithm.

$BIC_{min} \leftarrow +\infty$;
for $i = 1$ to i_{max} **do**
  for $n_{upd} = 1$ to N **do**
    $\tilde{\mathcal{A}}_{n_{upd}} \leftarrow full; \tilde{\mathcal{G}}_{n_{upd}} \leftarrow full; \lambda_{n_{upd}} \leftarrow$ a sufficiently high value;
    while $|\mathcal{A}_{n_{upd}}| + |\mathcal{G}_{n_{upd}}| \leq k_{max}$ **do**
      while convergence criterion is not satisfied **do**
        Update X_T and parameter values using the Kalman filter and the EM algorithm;
      end while
      if $BIC_{min} > BIC_{current}$; **then**
        $BIC_{min} \leftarrow BIC_{current}$; Store the current parameter values;
        where $BIC_{current}$ is the BIC score of the current parameter values
      end if
      Decrease $\lambda_{n_{upd}}$;
    end while
    Set the stored parameter values as the current parameter values;
    $sub_A \leftarrow$ the set of all subsets of the current $\mathcal{A}_{n_{upd}}$;
    $sub_G \leftarrow$ the set of all subsets of the current $\mathcal{G}_{n_{upd}}$;
    for all $s_A \in sub_A$ **do**
      $\tilde{\mathcal{A}}_{n_{upd}} \leftarrow s_A$;
      for all $s_G \in sub_G$ **do**
        $\tilde{\mathcal{G}}_{n_{upd}} \leftarrow s_G$;
        while convergence criterion is not satisfied **do**
          Update X_T and parameter values using the Kalman filter and the EM algorithm;
        end while
        if $BIC_{min} > BIC_{current}$ **then**
          $BIC_{min} \leftarrow BIC_{current}$; Store the current parameter values;
        end if
      end for
    end for
    Set the stored parameter values as the current parameter values;
  end for
end for

3.2.6 Weighting Known Regulations

To weight parameters of known regulations, *e.g.*, as recorded in the literature, we derive the weighted regularization [98]. For the nth row, we define the weight vectors $\boldsymbol{\omega}_n^a = (\omega_{n,1}^a, \ldots, \omega_{n,N}^a)'$ and $\boldsymbol{\omega}_n^g = (\omega_{n,1}^g, \ldots, \omega_{n,M}^g)'$. The elements of these vectors for known regulations are set to less than 1 or, otherwise set to 1. Then, in the M step of the EM algorithm and the regularized log-likelihood, regularization terms are handled as

$$\sum_{k=1}^{N} \lambda_n |a_{n,k}| \rightarrow \sum_{k=1}^{N} \omega_{n,k}^a \lambda_n |a_{n,k}|, \tag{3.43}$$

$$\sum_{k=1}^{M} \lambda_n |g_{n,k}| \rightarrow \sum_{k=1}^{M} \omega_{n,k}^g \lambda_n |g_{n,k}|. \tag{3.44}$$

In practice, the purpose of the weight is to select known regulation in the instance where multiple candidates are highly correlated with the same gene. Thus, when the correlation of a known regulation is still a low value, the regulation should not be selected as an active regulation. For example, weights for literature-recorded pathways and regulations by TFs are set as $\frac{1}{20}$ and $\frac{1}{10}$ in the real data experiment, respectively. The effectiveness of the weighted regularization is demonstrated in the results section.

3.3 Results

3.3.1 Comparison Results

To show the effectiveness of the proposed method, we compared it with other GRN inference methods, *i.e.*, a SSM [43,107], a general VAR model using the LARS-LASSO algorithm [23,124], GeneNet [80, 94] based on an empirical graphical Gaussian model (GGM), dynamic Bayesian networks using first order conditional dependencies (G1DBN) [63], GLASSO [29] based on sparse GGM and the mutual information-based network inference algorithms: ARACNE [70], CLR [26] and MRNET [75]. We applied these inference methods by using R-package ('GeneNet', 'G1DBN', 'glasso' and 'parmigene') and implementing the others. The comparison analysis was performed using three artificial data, which were generated based on pharmacogenomic pathways that we assumed and a yeast network that was produced as a part of the DREAM4 (Dialogue for Reverse Engineering Assessments and Methods) challenge. We should note that, because ARACNE, CLR and MRNET are intended to infer static relationships between genes, we considered time-course observational data as static data utilizing a time-lag matrix, in which the tth row vector consists of $\boldsymbol{y}_{t+1} - \boldsymbol{y}_t$, according to Shimamura *et al.* [99]. Note that the Jar file of the proposed method is available at: `http://sunflower.kuicr.kyoto-u.ac.jp`.

3.3.1.1 Comparison Using Pharmacogenomic Pathways

For the comparison, we first generated two time-courses from (i) linear difference equations as Eq. (3.4) and (ii) nonlinear differential equations as Eqs. (3.1)-(3.3) representing pharmacogenomic pathways (*e.g.*, Yao *et al.* [115]) using Cell Illustrator 5.0 (http://www.cellillustrator.com/home). The details of the artificial simulation models are as follows.

-Dataset (i)

1. The number of genes is 18.

2. Each gene undergoes synthesis and degradation processes, and genes are mutually regulated as shown in Fig 3.3 (The details of the figure are explained below).

3. A drug is added at $t = 0$ and its concentration gradually decreases according to one compartment model, *i.e.*, $\frac{d}{dt}z(t) = \zeta z(t)$, where $z(t)$ is the concentration of the drug as a function of time t and ζ is the degradation rate. The simulated expression profiles of the genes are initiated by the drug at $t = 0$ and gradually converge to their steady states as illustrated in Fig. 3.4.

4. The expression data is observed at $\mathcal{T}_{obs}$=(0, 1, 2, 3, 4, 6, 8, 12, 16, 24, 32, 48, 52, 96, 128, 160, 192, 224, 256 and 288) with Gaussian observation noise of mean 0 and a variance that is proportional to the intensity.

5. The number of replicated observations with different observational noise for each time point is three.

6. The simulated expression is updated according to the linear difference equations represented by Eq. (3.4) at $\Delta t = \frac{1}{5}$.

-Dataset (ii)

1 to 5 of dataset (i) are also satisfied in dataset (ii).

6. The simulated expression is updated according to the differential equations. Regulatory relationships are the same as in (i) but the regulatory effects are represented by hill functions, such as Eqs. (3.1)-(3.3), or linear functions, as illustrated in Fig 3.3. In this figure, $h(c)$ indicates that the regulation is described by Eq. (3.3) when $\gamma_{n,k} = c$.

A true positive (TP), false positive (FP), false negative (FN), precision rate (PR=$\frac{\mathrm{TP}}{\mathrm{TP+FP}}$), and recall rate (RR=$\frac{\mathrm{TP}}{\mathrm{TP+FN}}$) were used to measure the performance. At first, in applying the proposed method to the data, we changed the simulation time interval of Eq. (3.4) to $\frac{1}{\Delta t} = (1, 2, \ldots, 15)$, and estimated active sets of regulation ($\mathcal{A}$ and $\mathcal{G}$) and the values of the parameters for each $\frac{1}{\Delta t}$ for each dataset. The results for datasets (i) and (ii) are illustrated in Figs. 3.5 and 3.6, respectively. The precision and recall rates in Figs. 3.5 and 3.6 show that the performance of the structure inference gradually increases from $\frac{1}{\Delta t} \in 1$ and is optimal at $\frac{1}{\Delta t} = 10$ for (i) and $\frac{1}{\Delta t} = 9$ for (ii). This indicates that the simulation time interval Δt can

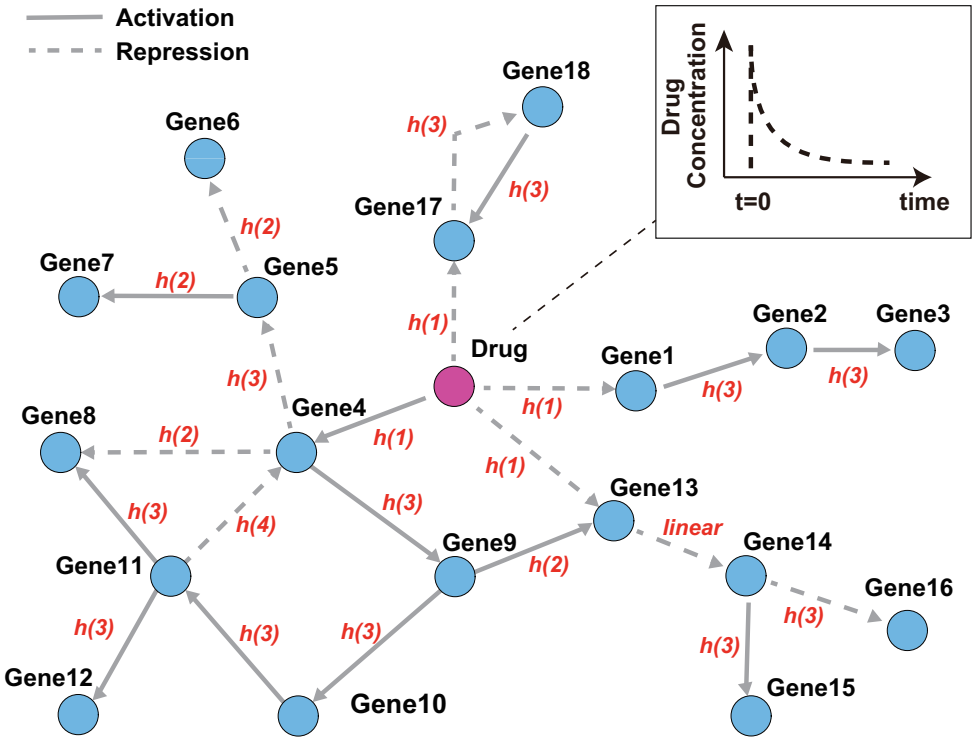

Figure 3.3: A pharmacogenomic pathway of the artificial simulation model. The figure illustrates the pathway of the artificial simulation model used for datasets (i) and (ii). Each regulation is represented by (i) a linear and (ii) a nonlinear function, such as Eq. (3.3). For dataset (ii), descriptions on edges as *linear* or $h(c = 1, 2, 3 \text{ or } 4)$ means a linear function and a hill function, described in Eq. (3.3) when $\gamma_{n,k} = c$, respectively. The system is stable at first ($t < 0$) and undergoes stimulation by a drug at $t = 0$. The concentration of the drug is gradually decreased according to the drug kinetics. A solid arrow and a dotted arrow mean activation and repression, respectively.

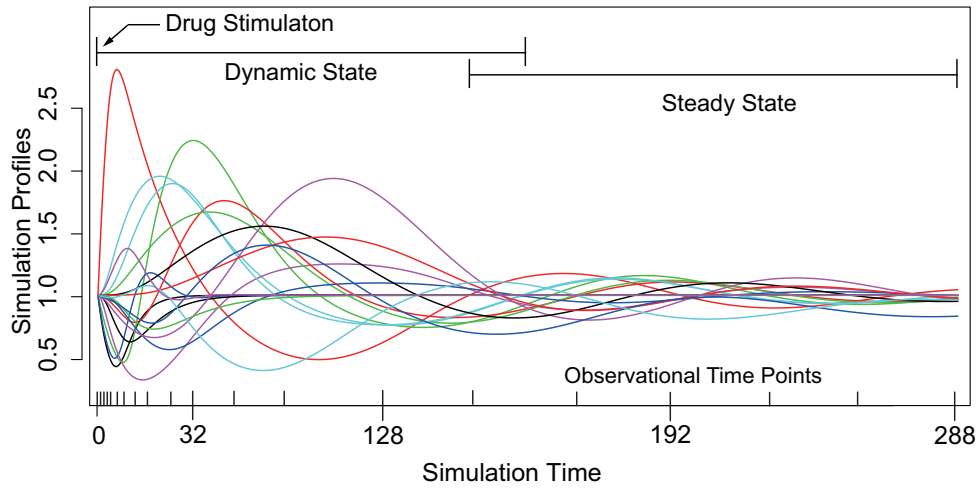

Figure 3.4: The simulation expression profiles of genes of the artificial simulation model. This illustrates the simulation expression profiles of genes of the artificial simulation model used for dataset (ii). The simulated data for datasets (i) and (ii) have both dynamic and steady state, and stimulated by the drug at $t = 0$. Observational time-course data is obtained with Gaussian noise from the simulation expression at the time points that are indicated on the bottom axis.

influence the performance of structure inference and we should carefully design Δt for biological simulations. In order to determine Δt, we measured the BIC scores and the sum of squared prediction errors (SPE) at three time points ($t = 6, 8$ and 12) for each $\frac{1}{\Delta t}$ using (i) and (ii), as represented in Fig. 3.7 and Fig. 3.8, respectively. Here, we measured the prediction errors for each time point by optimizing the values of the estimated parameters without using the observational data at the corresponding time point ($t = 6, 8, 12$).

For dataset (i), although the PR and RR values peak at $\frac{1}{\Delta t} = 10$, the BIC scores become lowest at $\frac{1}{\Delta t} = 2$. Similarly, the BIC score becomes lowest at $\frac{1}{\Delta t} = 7$ but peaks at $\frac{1}{\Delta t} = 9$ for dataset (ii). SPE gradually converges when $\frac{1}{\Delta t}$ becomes large and has the lowest value at $\frac{1}{\Delta t} = 11$ and $\frac{1}{\Delta t} = 9$ for datasets (i) and (ii), respectively. Therefore, SPE can be an indicator for determining the best time interval for this hill function-based system of pharmacogenomics. Note that the measured time points for the prediction errors should be the points that are not steady state values.

Next, we compared the results of (a) the proposed VAR-SSM with the lowest BIC and (b) the proposed VAR-SSM with the lowest SPE to (c) SSM [43,107] (permutation tests were utilized to select regulations), (d) VAR model with $L1$ regularization using the LARS-LASSO algorithm [23, 124] (the BIC score is used to determine the value of the regularization parameters), GeneNet [80,94], G1DBN [63], GLASSO [29], ARACNE [70], CLR [26] and MRNET [75]. The comparison results for datasets (i) and (ii) are listed in Tables 3.1 and 3.2, respectively. In these comparisons, we added the drug profiles to the observational data and did not count regulations in response to drugs and self-regulation. For the methods inferring undirected regulations, *i.e.*, GeneNet, GLASSO, ARACNE, CLR and MRNET, we considered the true network (directed network)

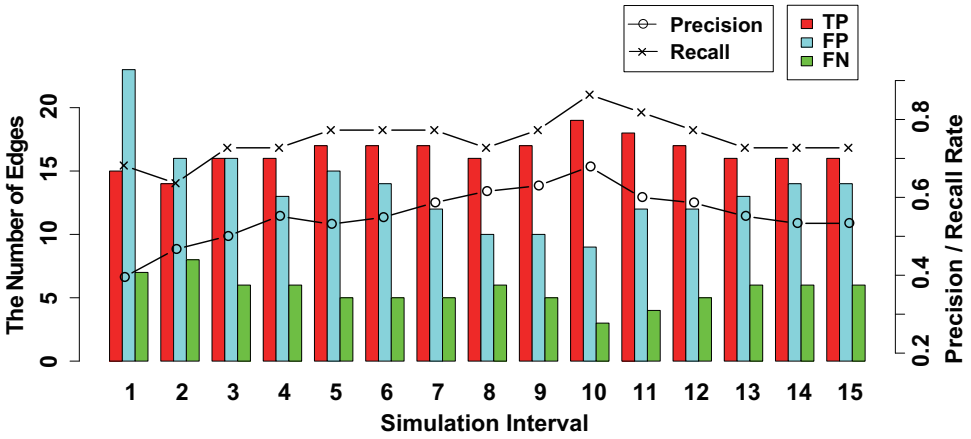

Figure 3.5: The results of the structure inference using dataset (i) of a pharmacogenomic pathway by the proposed method. This figure illustrates the results of the structure inference after applying the proposed method to dataset (i) for each simulation time interval Δt. The histogram represents the number of true positive (TP), false positive (FP), and false negative (FN) findings for each $\frac{1}{\Delta t} = (1, 2, \cdots, 15)$ as red, blue, and green bars, respectively. Black lines with circles and crosses represent 'precision rate (PR=$\frac{\mathrm{TP}}{\mathrm{TP+FP}}$)' and 'recall rate (RR=$\frac{\mathrm{TP}}{\mathrm{TP+FN}}$)', respectively. The values of the histogram and lines correspond to the left and right axes, respectively.

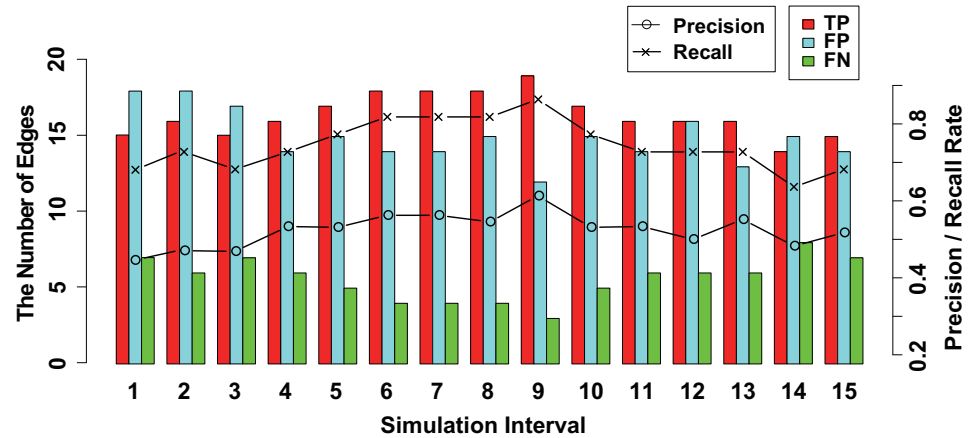

Figure 3.6: The results of the structure inference using dataset (ii) of a pharmacogenomic pathway by the proposed method. This figure illustrates the results of the structure inference after applying the proposed method to dataset (ii) for each simulation time interval Δt. The histogram represents the number of true positive (TP), false positive (FP), and false negative (FN) findings for each $\frac{1}{\Delta t} = (1, 2, \cdots, 15)$ as red, blue, and green bars, respectively. Black lines with circles and crosses represent 'precision rate (PR=$\frac{\mathrm{TP}}{\mathrm{TP+FP}}$)' and 'recall rate (RR=$\frac{\mathrm{TP}}{\mathrm{TP+FN}}$)', respectively. The values of the histogram and lines correspond to the left and right axes, respectively.

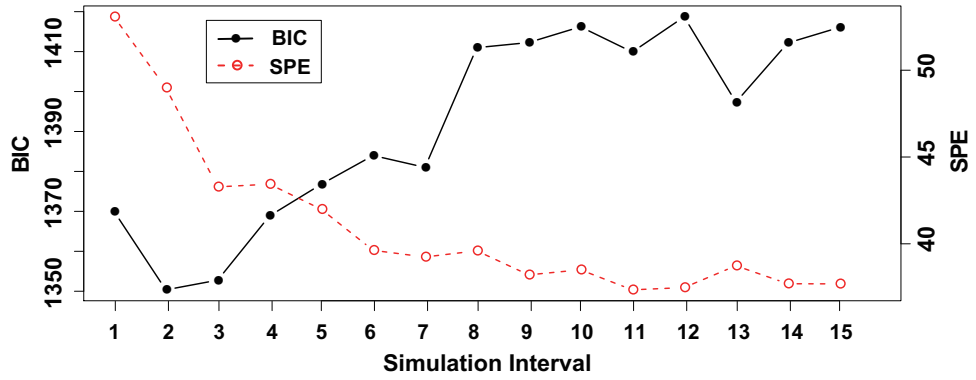

Figure 3.7: The result of the BIC scores and SPE for each simulation time interval using dataset (i). This illustrates the BIC scores and SPE ($t = 6, 8, 12$) for $\frac{1}{\Delta t} = = (1, 2, \cdots, 15)$ for dataset (i). The values of the BIC scores and SPE correspond to the left and right axes, respectively.

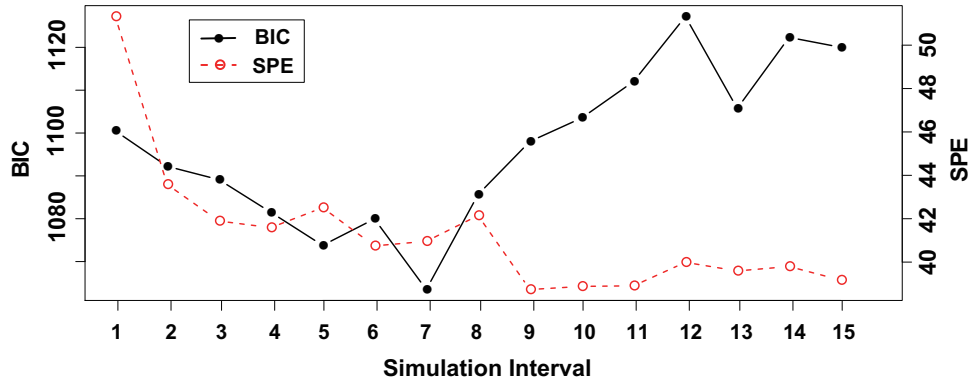

Figure 3.8: The result of the BIC scores and SPE for each simulation time interval using dataset (ii). This illustrates the BIC scores and SPE ($t = 6, 8, 12$) for $\frac{1}{\Delta t} = = (1, 2, \cdots, 15)$ for dataset (ii). The values of the BIC scores and SPE correspond to the left and right axes, respectively.

as an undirected network and then measured the performance by comparing this undirected network to the inferred networks. Additionally, for GeneNet, G1DBN and mutual information-based methods (ARACNE, CLR and MRNET), which are required to set a threshold value to determine the existence of regulation, we checked the results of setting the threshold q-value (GeneNet) and posterior probability (G1DBN) to $(0.01, 0.05, 0.1, 0.2, \ldots,$ and $0.5)$ and a cut-off value (ARACNE, CLR and MRNET) to $(0, 0.01, 0.05, 0.1, 0.2, \ldots,$ and $0.8)$, and adopted the best thresholds with respect to $F\text{-}measure = \frac{2 \cdot PR \cdot RR}{PR+RR}$. We should note that the simulation time interval of SSM is set $\Delta t = 1$ (no other choice is available) due to the implementation of Tamada *et al.* [107]. It is hard to make the simulation time interval short; hence, the simulated expression profiles often oscillated in such situations.

Table 3.1: Comparison of the proposed method and the existing methods using dataset (i).

	PR	**RR**	**TP**	**FP**	**TN**	**FN**
(a)VARSSM(BIC)	0.467	0.634	14	16	286	8
(b)VARSSM(SPE)	0.600	0.818	18	12	290	4
(c)SSM	0.308	0.182	4	9	293	18
(d)VAR	0.150	0.773	17	97	205	5
(e)Genenet	0.280	0.667	14	36	114	7
(f)G1DBN	0.314	0.500	11	24	278	11
(g)GLASSO	0.094	0.286	6	58	92	15
(h)ARACNE	0.131	0.524	11	71	79	10
(i)CLR	0.135	0.619	13	83	67	8
(j)MRNET	0.121	0.571	12	87	63	9

Table 3.2: Comparison of the proposed method and the existing methods using dataset (ii).

	PR	**RR**	**TP**	**FP**	**TN**	**FN**
(a)VARSSM(BIC)	0.563	0.818	18	14	288	4
(b)VARSSM(SPE)	0.613	0.864	19	12	290	3
(c)SSM	0.234	0.318	7	23	279	15
(d)VAR	0.206	1.000	22	84	236	0
(e)Genenet	0.278	0.714	15	39	111	6
(f)G1DBN	0.647	0.500	11	6	296	11
(g)GLASSO	0.052	0.143	3	55	95	18
(h)ARACNE	0.191	0.429	9	38	112	12
(i)CLR	0.156	0.667	14	76	74	7
(j)MRNET	0.156	0.667	14	76	74	7

Consequently, the proposed method achieved a low false positive rate while maintaining a high true positive rate. These results may be acceptable because the system model of the proposed method is the same as or similar to the artificial simulation models. Thus, it is conceivable that the proposed method is highly capable of inferring the regulatory structure of the assumed hill-function based model. Furthermore, we demonstrated the effectiveness of the weighted regularization for known prior information using dataset (ii). To evaluate the performance, we adapted a simulation time interval of $\frac{1}{\Delta t} = 9$. Setting weights for true regulations

as $\frac{1}{w_{n,k}} = (1.5, 2, 3, \cdots, 20)$, PR and RR were evaluated as illustrated in Fig. 3.9. The correct weights reduced the FP and FN edges, and the performance was gradually improved according to the increase in the weight coefficient. In contrast, several FP edges still exist even when the weight coefficients take on high values. It can be considered that the simplification of the true regulatory system using the proposed model generates these false edges to effectively predict the data.

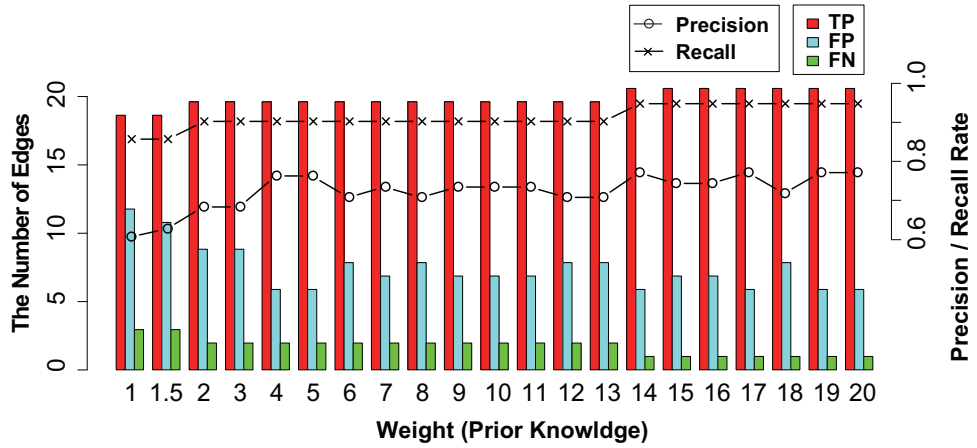

Figure 3.9: The performance of using prior knowledge as the weighted regularization. This figure illustrates the effectiveness of the weighted regularization (prior knowledge) at simulation time interval $\frac{1}{\Delta t} = 9$ using dataset (ii). The histogram represents the number of true positive (TP), false positive (FP), and false negative (FN) findings for each $\frac{1}{w_{n,k}} = (1, 1.5, 2, 3,, \cdots, 20)$ as red, blue, and green bars, respectively. Black lines with circles and crosses represent 'precision rate (PR= $\frac{\mathrm{TP}}{\mathrm{TP+FP}}$)' and 'recall rate (RR= $\frac{\mathrm{TP}}{\mathrm{TP+FN}}$)', respectively. The values of the histogram and lines correspond to the left and right axes, respectively.

3.3.1.2 Comparison Using Yeast Network of a Part of the DREAM4 Challenge

In contrast to the previous comparisons, for which the data were based on the assumed models as Eqs. (3.1)-(3.4), we next prepared data generated by GeneNetWaver [69, 95] using a 10-node yeast network (*yeast 1*) of a part of the DREAM4 challenge (in silico network challenge). To measure the performance of the proposed method, in this comparison, we generated dataset (iii), which was a set of 100 time-course observational data, in which the measured time points were $t = (0, 1, \ldots, 30)$.

According to the original setting, three genes, which were randomly selected for each time-course, were perturbed among $t = 0$ to 15. Here, since we intended to consider the case that observational data have a steady state, the number of time points was to be set larger than those of the original setting $t = (0, 1, \ldots, 20)$.

We applied the methods (a)-(j) to dataset (iii); however, since SSM [43, 107] requires large computational costs to perform permutation tests for each time-course, we neglected SSM for this comparison. The time points to calculate SPE for the proposed method are $t = (16, 17, 18)$,

which are the time points shortly after removal of perturbations. For each method, we summed the existence of the estimated regulation on the ith gene by jth gene as $est_{i,j}$ and considered the values $\frac{est_{i,j}}{100}$ as the confidence level for the regulation. Then, TP rate (TPR=$\frac{\mathrm{TP}}{\mathrm{TP+FN}}$), FP rate (FPR=$\frac{\mathrm{FP}}{\mathrm{FP+TN}}$), precision rate (PR=$\frac{\mathrm{TP}}{\mathrm{TP+FP}}$) and recall rate (RR=$\frac{\mathrm{TP}}{\mathrm{TP+FN}}$) were calculated to draw ROC and PR curves. Using these curves, we measured the performance with respect to the AUROC (area under the ROC curve) and AUPR (area under the PR curve). These comparison results are illustrated in Fig. 3.10. Note that, similarly to the previous experiments, we selected the best threshold values with respect to AUROC for the methods (e), (f) and (h)-(j).

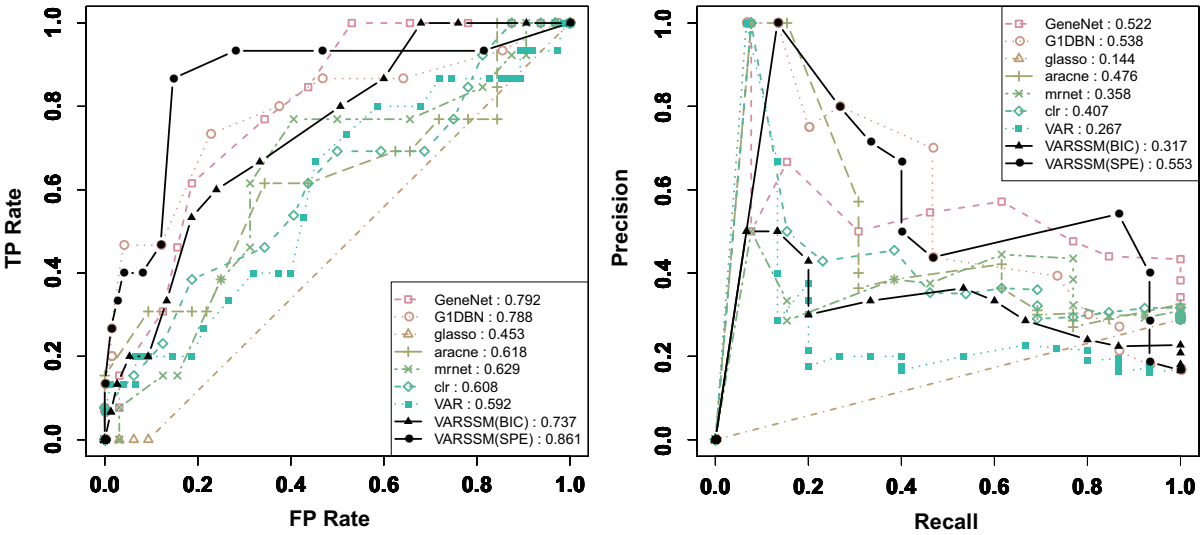

Figure 3.10: The ROC and PR curves using dataset (iii). The left and the right figures illustrate the ROC and PR curves for dataset (iii), respectively. In the left figure, the vertical axis and horizontal axis correspond to TP rate and FP rate, respectively. In the right figure, the vertical axis and horizontal axis correspond to PR and RR, respectively. AUROC and AUPR are represented at the right side of the inference methods.

As a result, although the simulation model for dataset (iii) is different from the models that we assumed, the proposed method using SPE outperformed the other methods in terms of both AUROC and AUPR. The number of selected simulation time intervals Δt is shown in Table 3.3. These results indicate that the proposed method has good ability for inferring the regulatory relationships using time-course observational data for which regulations are not based on the model that we assumed. Furthermore, we can consider the SPE as a good indicator for determining the simulation time interval.

Table 3.3: The number of selected simulation time intervals for dataset (iii).

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
BIC	99	0	0	0	0	1	0	0	0	0	0	0	0	0	0
SPE	3	13	9	1	4	0	2	6	5	6	3	11	6	12	19

3.3.2 Application to Corticosteroid Pathways in Rats

As an application example, we analyzed microarray time-course gene expression data from rat skeletal muscle [5, 115], which is assumed to have the same system used in simulation studies. The microarray data were downloaded from the GEO database (GSE490). The time-course gene expression was measured at 0, 0.25, 0.5, 0.75, 1, 2, 4, 5, 5.5, 7, 8, 12, 18, 30, 48, and 72 [h] (16 time points) after the glucocorticoid was applied. The data at time 0 represent controls (untreated). There were two, three, or four replicated observations for each time point.

Because corticosteroid pharmacokinetics/dynamics in skeletal muscle have been modeled based on differential equations [115] as shown in Fig. 3.11, the time-dependent concentration of corticosteroid in nucleus in rat skeletal muscle $\boldsymbol{z}_t$ can be obtained as followings;

$$\frac{d\mathrm{mRNA_R}(t)}{dt} = k_{s\_Rm} \cdot \left\{1 - \frac{\mathrm{DR_N}(t)}{\mathrm{IC_{50\_Rm}} + \mathrm{DR_N}(t)}\right\} - k_{d\_Rm} \cdot \mathrm{mRNA_R}(t), \tag{3.45}$$

$$\frac{d\mathrm{R}(t)}{dt} = k_{s\_R} \cdot \mathrm{mRNA_R}(t) + R_f \cdot k_{re} \cdot \mathrm{DR_N}(t) - k_{on} \cdot \mathrm{D}(t) \cdot \mathrm{R}(t) - k_{d\_R} \cdot \mathrm{R}(t), \tag{3.46}$$

$$\frac{d\mathrm{DR}(t)}{dt} = k_{on} \cdot \mathrm{D}(t) \cdot \mathrm{R}(t) - k_T \cdot \mathrm{DR}(t), \tag{3.47}$$

$$\frac{d\mathrm{DR_N}(t)}{dt} = k_T \cdot \mathrm{DR}(t) - k_{re} \cdot \mathrm{DR_N}(t), \tag{3.48}$$

where $\mathrm{mRNA_R}(t)$ is the concentration of mRNA of the receptor protein, $\mathrm{R}(t)$ is the concentration of the receptor protein, $\mathrm{DR}(t)$ is the concentration of the drug-receptor complex, $\mathrm{DR_N}(t)$ is the concentration of the drug-receptor complex in nucleus, and *Synthesis* and *Degradation* mean synthesis and degradation processes, respectively. $\mathrm{DR_N}(t)$ was used for $\boldsymbol{z}_t$. These parameter values, $k_{s\_Rm}, \mathrm{IC_{50\_Rm}}, k_{d\_Rm}, k_{s\_R}, k_{d\_R}, R_f, k_{re}, k_{on}, k_{d\_R}, k_T$, are shown in Table 3.4. According to the previous research [106], the time-evolution of the plasma concentration of corticosteroid is given as

$$\mathrm{D}(t) = 39,130 \cdot e^{-7.54t} + 12,670 \cdot e^{-1.20t}. \tag{3.49}$$

Table 3.4: The values of the parameters for corticosteroid pharmacodynamics.

parameter	value
k_{s_Rm}(fmol/g/h)	0.416
k_{d_Rm}(1/h)	0.139
k_{s_R}(fmol/g/h)	0.777
k_{d_R}(1/h)	0.0356
k_{on}(1/nmol/h)	0.00269
k_T(1/T)	90
k_{re}(1/h)	0.618
R_f	0.720
$IC_{\_50Rm}$(fmol/mg)	0.911
$mRNA_R^0$(fmol/g)	2.99
R^0(fmol/mg)	65.3

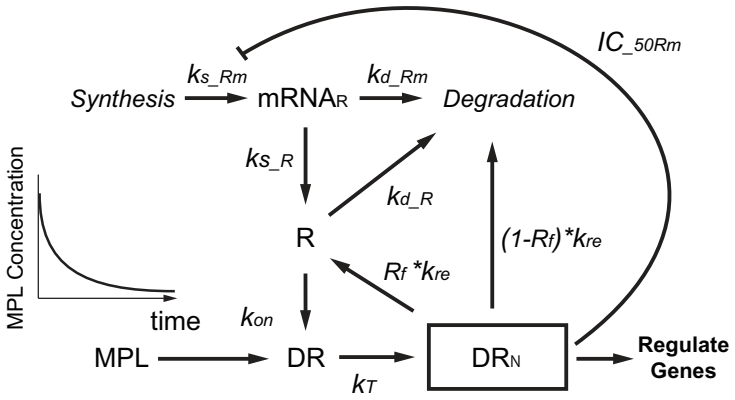

Figure 3.11: The pharmacokinetics/dynamics developed previously [115]. The dynamics behavior of the concentration of biomolecules is described by differential equations. $\mathrm{mRNA_R}$ is the concentration of mRNA of the receptor protein, R is the concentration of the receptor protein, DR is the concentration of the drug-receptor complex, $\mathrm{DR_N}$ is the concentration of the drug-receptor complex in nucleus, and *Synthesis* and *Degradation* mean synthesis and degradation processes, respectively.

Furthermore, corticosteroid catabolic/anabolic processes in rat skeletal muscle have been partly established [100]; thus, these regulatory relationships can also be used. Given this information, we included *Mtor*, *Anxa3*, *Bnip3*, *Bcat2*, *Foxo1*, *Trim63*, *Akt1*, *Akt2*, *Akt3*, *Rheb*, *Igf1*, *Igf1r*, *Pik3c3*, *Pik3cd*, *Pik3cb*, *Pik3c2g*, *Slc2a4*, and *Mstn*. Note that the microarray (GSE490) does not include three genes in the original pathway [100], *Redd1*, *Bcaa* and *Klf15*. In addition, we employed the genes, *Irs1*, *Srebf1*, *Rxrg*, *Scarb1*, *Gpam*, *Scd*, *Gpd2*, *Mapk6*, *Ace*, *Ptpn1*, *Ptprf*, *Edn1*, *Agtr1a*, *Ppard*, *Hmgcs2*, *Serpine1*, *Cebpb*, *Cebpd*, *Il6r*, *Mapk14*, *Ucp3*, and *Pdk4*, which have been suggested to be corticosteroid-induced genes [5]. In summary, we applied the method to these 40 genes with weights for the established pathway and the concentration of corticosteroid.

First, to determine the simulation time interval from $\frac{1}{\Delta t} = \{1, 2, \ldots, 9\}$, we evaluated the BIC scores and SPE ($t = 1, 2, 4$). The results are shown in Fig. 3.12. Interestingly, even for the observational data, we obtained the same tendency for both indicators. Therefore, we obtained $\frac{1}{\Delta t} = 4$ for the lowest SPE. Next, we analyzed the result of $\frac{1}{\Delta t} = 4$. The inferred structure with some simulated expression profiles are illustrated in Fig. 3.13. From the figure, we can capture the propagation of gene expression stimulated by *corticosteroid* and hub genes regulating other genes. However, these results may be difficult to biologically interpret because some mRNAs are not considered to regulate other genes. Therefore, to exploit biological meaning correctly and demonstrate the effectiveness of incorporating prior information in the case of real biological data, we finally performed an experiment using TF information from ITFP [122]. Then, weights for regulations by TFs, *Trim63*, *Akt1*, *Akt2*, *Mstn*, *Irs1*, *Srebf1*, *Gpam*, *Cebpb*, and *Cebpd*, were set $\frac{1}{w_{n,k}} = 10$. The inferred structure at $\frac{1}{\Delta t} = 4$ using the TF information is illustrated in Fig. 3.14.

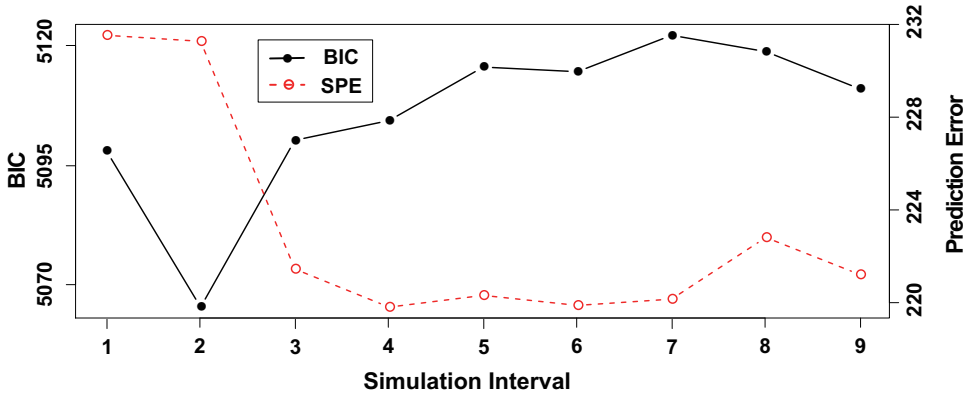

Figure 3.12: The result of the BIC scores and SPE for each simulation time interval using the real data. This illustrates the BIC scores and SPE ($t = 1, 2, 4$) for each time interval $\frac{1}{\Delta t} = \{1, 2, \ldots, 9\}$. The values of these indicators corresponds to the left and right axes, respectively.

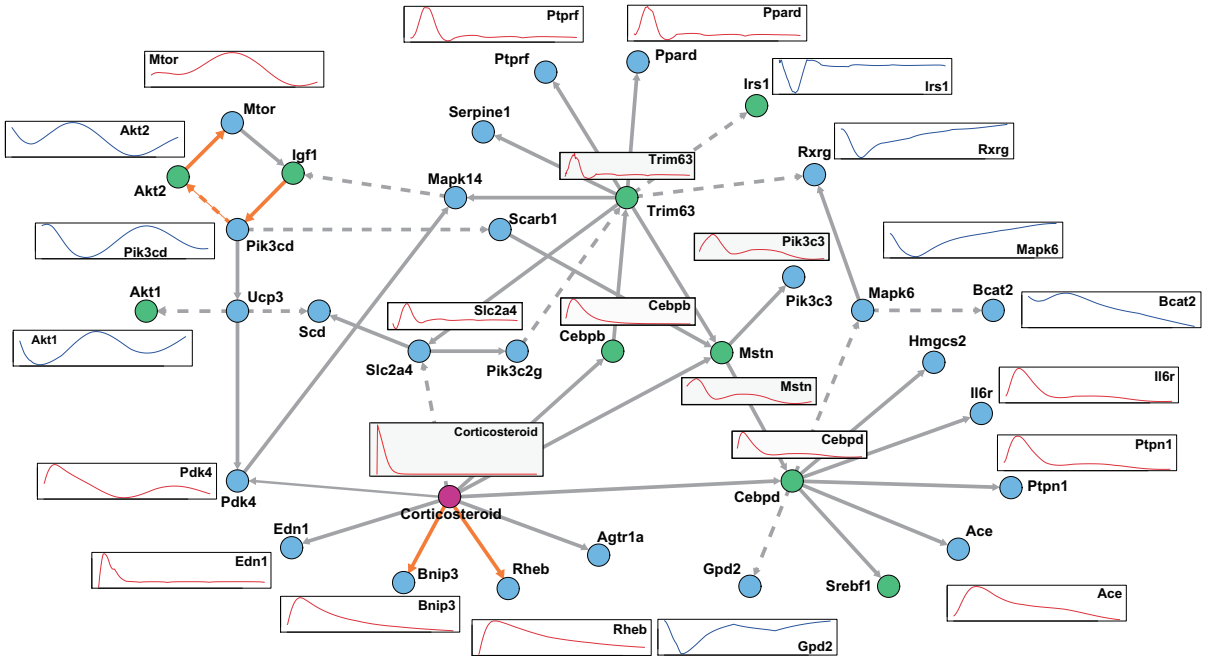

Figure 3.13: The estimated network with weighting literature-recorded pathways. This figure illustrates the inferred gene regulatory network with weights for literature-recorded pathways. *Corticosteroid* and genes of TFs are drawn as a red circle and green circles, respectively. Estimated edges with weights are illustrated as orange. Further, on some genes, simulation expression profiles are attached as examples. Red and blue profiles are roughly distinguished to up-regulated and down-regulated genes, respectively.

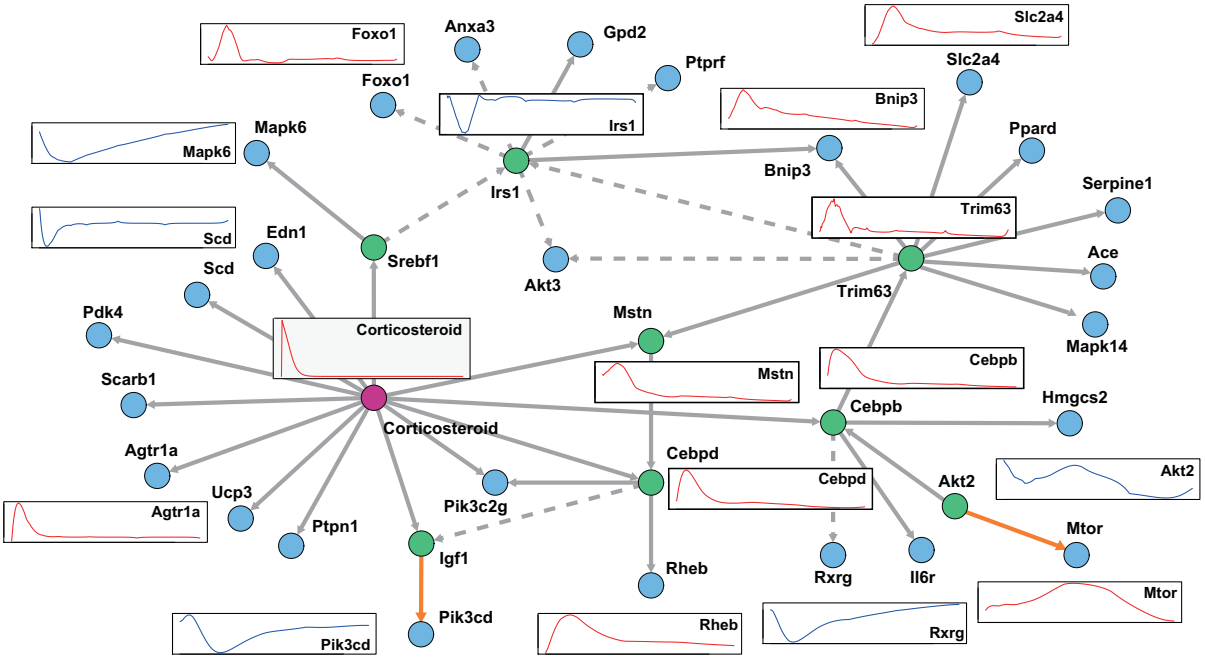

Figure 3.14: The estimated network with weighting literature-recorded pathways and regulations by TFs. This figure illustrates the inferred gene regulatory network with weights for literature-recorded pathways and regulations by TFs. *Corticosteroid* and genes of TFs are drawn as a red circle and green circles, respectively. Estimated edges with weights for literature derived regulations are illustrated as orange. Red and blue simulation profiles are roughly distinguished to up-regulated and down-regulated genes, respectively.

In Figs. 3.13 and 3.14, there are some interesting observations. At first, some genes are directly regulated by corticosteroids, which are included in the model as $\boldsymbol{z}_t$. Thus, other models that do not include the drug terms cannot estimate such regulation. Second, only weighted regulations, *i.e.*, literature-recorded pathways and regulation by TFs, were inferred in contrast to the non-weighted network in Fig. 3.13. Thus, we could successfully incorporate prior knowledge, and further candidates may extend our understanding of regulation not yet reported in literature. Additionally, some weighted genes, *Cebpb*, *Mstn*, *Cebpd*, and *Trim63*, were also selected as hub genes with no weight in Fig. 3.13. Third, *Cebpb*, which is known as a transcription factor related to immune and inflammatory responses, is indicated as a hub gene (illustrated as a green circle). *Cebpd* and *Cebpb* are assumed to be candidate genes for insulin-related transcription factors [28]. This finding may confirm the findings of previous studies [5, 115] indicating that corticosteroid stimulation of skeletal muscle can induce the expression of insulin.

Finally, we applied the other methods, *i.e.*, GeneNet and G1DBN, to the pharmacogenomic data and attached significance levels (q-val and *posterior probability* for GeneNet and G1DBN, respectively) for the regulations inferred by the proposed method. The results are presented in Table 3.5. Interestingly, some regulations have very high significance levels but others do not. For example, regulations of *Srebf1, Agtr1a, Cebpb* and *Cepbd* by a *corticosteroid* are quite probable. In contrast, some regulations were not significant when using these methods. We can suppose, for example, that differences between the models, the prior weights for TF candidates and literature derived pathways, steady state gene expression profiles and corticosteroid drug dynamics in the proposed model may have caused the results. Although some inferred regulations had low significance levels in other approaches, we believe that these regulations can be candidates for true regulation in corticosteroid pharmacogenomic pathways because the proposed method outperformed the other methods through the comparison using synthetic pharmacogenomic pathways.

Although we actually used 40 genes, only 35 genes were found to be regulated because the expression of residual genes did not vary through the time-course. Hence the expression of these genes can represent only synthesis and degradation processes, for which regulation was not estimated.

Table 3.5: The confidence levels of estimated pharmacogenomic regulations using GeneNet and G1DBN.

Regulator	**Target**	***q*–val**	**post–prob.**
Corticosteroid	Srebf1	0.101	0.000
Corticosteroid	Agtr1a	0.864	0.002
Corticosteroid	Cebpd	0.021	0.003
Corticosteroid	Cebpb	0.747	0.003
Trim63	Serpine1	0.375	0.005
Corticosteroid	Mstn	0.198	0.012
Trim63	Irs1	0.385	0.065
Corticosteroid	Scd	0.905	0.068
Akt2	Mtor	0.881	0.069
Cebpb	Il6r	0.836	0.102
Trim63	Ppard	0.395	0.105
Trim63	Slc2a4	0.915	0.189
Corticosteroid	Ucp3	0.663	0.195
Trim63	Bnip3	0.629	0.217
Trim63	Mstn	0.935	0.273
Mstn	Cebpd	0.413	0.280
Irs1	Ptprf	0.928	0.452
Igf1	Pik3cd	0.897	0.457
Trim63	Mapk14	0.909	0.503
Irs1	Anxa3	0.107	0.632
Irs1	Gpd2	0.853	0.749
Corticosteroid	Edn1	0.833	0.799
Corticosteroid	Pik3c2g	0.929	0.821
Cebpb	Trim63	0.864	0.991
Irs1	Akt3	0.396	1.000
Srebf1	Mapk6	0.453	1.000
Corticosteroid	Scarb1	0.651	1.000
Cebpb	Rxrg	0.734	1.000
Corticosteroid	Ptpn1	0.827	1.000
Srebf1	Irs1	0.832	1.000
Akt2	Cebpb	0.863	1.000
Corticosteroid	Pdk4	0.871	1.000
Cebpd	Pik3c2g	0.888	1.000
Irs1	Foxo1	0.894	1.000
Cebpb	Hmgcs2	0.897	1.000
Corticosteroid	Igf1	0.908	1.000
Trim63	Akt3	0.913	1.000
Cebpd	Igf1	0.924	1.000
Irs1	Bnip3	0.925	1.000
Cebpd	Rheb	0.935	1.000
Trim63	Ace	0.936	1.000

3.4 Discussion

In this study, we proposed a novel method for inference of gene regulatory networks incorporating existing biological knowledge and time-course observation data. The properties of the method are as follows; (i) the dynamics of the gene expression profiles can be estimated based on the proposed linear model with a hidden state, (ii) $L1$ regularized log-likelihood is maximized to infer the active sets of regulation, (iii) the dynamics of other biomolecules can be included in the model, (iv) existing biological knowledge, *e.g.*, literature-recorded pathways and TF information, can be integrated. Furthermore, we proposed an indicator for selecting a simulation time interval for the inference.

To show the effectiveness of the proposed method, we compared it to the previously reported GRNs inference methods using hill function-based pharmacogenomic pathways [115] and a yeast network that is a part of the DREAM4 challenge [69, 95]. Since the artificial simulation models were described by differential equations or difference equations, in which the time intervals were smaller than the measurement interval, to reproduce a realistic biological system, the simulated expressions was updated in detail. In this situation, we assumed that the simulation time interval for the method is crucial for inference. As we expected, the results demonstrated that inference of the regulatory structure depends greatly on the simulation time interval. This indicates that we should carefully design the simulation time interval even for analysis of real observational data. For this purpose, we introduced indicators to determine the simulation time interval and measured their validity. Here, since the tendency of the indicator for the simulation time interval depends on the analyzed biological system, it is recommended to check the tendency by using simulation models. Upon comparison of the inferred structures, the proposed method using the indicator showed the highest performance in terms of precision and recall rates for all three data types. Although the first two synthetic data include the time-evolution of the drug profiles as same as the real data of rat skeletal muscle that was focused in this study, the previous methods can only handle the concentration of the drug at the observed non-equally spaced time points. In addition, the previous methods cannot deal with data including both of the dynamic and the steady states. This could contribute to the higher performance of the proposed method. The fact that the proposed method outperformed the other methods in using synthetic datasets, which includes the model we do not assume, indicates the adaptability of our proposed method.

For an application example, we applied the proposed method to a corticosteroid-stimulated pathway in rat skeletal muscle. Because pathways and genes related to corticosteroids have been widely investigated, we were able to obtain the concentration of the drug as a function of time from the corticosteroid kinetics/dynamics and the literature-recorded pathways. By incorporating time-course mRNA expression data, corticosteroid kinetics/dynamics, literature-recorded pathways and TF information, we inferred the regulatory relationships among 40 genes that are candidate or known corticosteroid-related genes. The tendency of the BIC scores and the SPE for the simulated time intervals were the same as in the simulation studies, in which the regulatory systems were based on the previous corticosteroid pharmacogenomic studies, and

interesting findings for corticosteroid regulation were obtained. For example, genes that are suggested to be significant factors in corticosteroid pharmacogenomics were predicted to be hub genes regulating other genes in the results both with and without prior information. Furthermore, we found that the properties of the proposed method, *i.e.*, the weighted regularization and inclusion of a term for other biomolecules, influenced the results of selecting potential regulators and introducing drug effects to genes, respectively. Finally, these inferred regulations were evaluated by GeneNet and G1DBN, and some of the regulations had high significance. Since our approach imposed prior weights for reliable regulations and included drug terms to explicitly represent their dynamics, not only these regulations but also regulations that are evaluated as non-significant could be candidate regulations for corticosteroid pharmacogenomics. These results indicate that the proposed method can help to elucidate candidates that will allow extension of GRNs in which the regulation among genes is partly understood by incorporating multi-source biological knowledge.

Chapter 4

An Efficient Data Assimilation Schema for Restoration and Extension of Gene Regulatory Networks Using Time-course Observation Data

4.1 Background

Intracellular systems in cells consist of many genetic and chemical interactions and GRNs play a crucial role in sustaining such systems. Although comprehensive understanding of GRNs is still lacking, much data have been recorded in the literature following recent advances in biotechnology, *e.g.*, microarray and Chip-Seq. Thus, by integrating these findings, we may be able to reconstruct GRNs and understand the dynamic behavior of gene expression through mathematical simulation models. However, since some unverified interactions are present in the literature, simulation results may not match the observed data, *e.g.*, microarray expression data. In this respect, a method for finding candidate networks that are consistent with the data by improving and extending literature-based models is needed to elucidate GRNs [39, 40, 78].

In order to construct simulation models of GRNs, interactions between biomolecules, *e.g.*, mRNA and proteins, are firstly collected from the literature and are integrated to construct the networks. Then, mathematical differential or difference equations are given to the constructed networks to simulate the dynamic behavior of these biomolecules. Thus, biologically reliable models, *e.g.*, the Michaelis-Menten model [91] and S-system [92], described by differential equations, have been applied in dealing with the limited number of genes [40, 67, 76, 83, 87]. In these approaches, a simulation-based methodology, called data assimilation, was employed for estimating parameter values and evaluating such simulation models [77, 79]. However, although

simulation results generated from these models can be biologically reasonable, evaluation of even one simulation model with estimating optimal parameter values is computationally demanding since parameter estimation must rely on a type of Monte Carlo methodologies [51, 58, 59, 76]. Therefore, it is computationally implausible to find appropriate models from a large number of candidate models.

In contrast to such approaches, in order to cope with the computational burden known as the curse of dimensionality in applying mathematical models to elucidate GRNs, there exists the other approach to use linear models for dealing with more than a hundred genes. In this approach, many effective methods have been developed, *e.g.*, state space models [12, 43, 86] and Bayesian inference [29, 68, 111]. For restoring literature-based GRNs, a concept, called network completion, has also been developed [3, 78]. However, these methods could fail in some cases, *e.g.*, handling non-equally spaced time-point data, because of simplified abstractions of biological systems. Thus, since these models cannot adequately represent the dynamics of gene expression due to simplified abstractions of biological systems, biologically invalid results might often be obtained. For improving and extending literature-based GRNs, these models are not sufficient because the number of genes is limited and their regulatory relationships are mostly reliable.

Here, applied simulation models should maximally emulate reliable biological dynamics under the constraint that their parameter values can be efficiently estimated. To satisfy the requirements, we developed a new data assimilation schema that applies a simple nonlinear simulation model, termed the combinatorial transcription model [81, 110]. As a part of this schema, we applied the unscented Kalman filter (UKF) [16, 49, 51] to obtain approximate posterior probability distributions of the hidden state and estimated parameter values maximizing prediction ability for observational data by means of the EM-algorithm. Then, a novel algorithm was developed to efficiently select and evaluate a candidate network to obtain a network that can best predict the data within a framework of the nonlinear state space model.

To show the effectiveness of the proposed method, we performed a comparison using artificial data in regard to a previously proposed network completion method [78]. For the comparison, synthetic data with equally and non-equally spaced time-points were generated from WNT5A [55] and a yeast cell cycle network [53]. Next, as real data experiments, a yeast cell-cycle network from KEGG database [53] and candidate genes from The Saccharomyces Genome Database (SGD) [15], which can have functions related to this network, were integrated to extend the network using real mRNA expression data [103].

4.2 Methods

4.2.1 A State Space Representation of Combinatorial Transcription Model

Let $x_i(t)$ be the abundance of the ith ($i = 1, \ldots, p$) gene as a function of time t. As a gene regulatory model, we assume a system in which each gene undergoes synthesis and degradation processes, and its expression value can be controlled through regulations of its synthesis process

by other genes. Thus, $x_i(t)$ is determined by

$$\frac{dx_i(t)}{dt} = f_i(\boldsymbol{x}(t), \boldsymbol{\theta}) \cdot u_i - x_i(t) \cdot d_i + v_{i,t}, \tag{4.1}$$

where f_i is a function of the regulatory effect on the ith gene, $\boldsymbol{x}(t) = (x_1(t), \ldots, x_p(t))'$, $\boldsymbol{\theta}$ is a tuning parameter, u_i is synthesis coefficient, d_i is a degradation coefficient and $v_{i,t}$ is a system noise at time t. Typically, f_i is represented by a hill function, such as the Michaelis-Menten model [91].

Due to its heavy computational cost to estimate parameter values maximizing prediction ability for the data, Eq. (4.1) is often approximated as a difference equation. Then, we apply the combinatorial transcription model [81, 110] as

$$x_{i,t+\Delta t} = x_{i,t} + (\sum_{j \in \mathcal{A}_i} a_{i,j} \cdot x_{j,t} + \sum_{j \in \mathcal{A}_i} \sum_{k \in \mathcal{A}_i \setminus j} b_{i,(j,k)} \cdot x_{j,t} \cdot x_{k,t} + u_i - x_{i,t} \cdot d_i + v_{i,t}) \cdot \Delta t, \tag{4.2}$$

where $x_{i,t}$ is the amount of the ith gene at time t, $a_{i,j}$ is an individual effect of the jth gene on the ith gene, $b_{i,(j,k)}$ is a combinatorial effect from the jth and the kth genes to the ith gene, $\mathcal{A}_i$ is an active set of genes regulating the ith gene and Δt is a minute displacement. Here, we set $\Delta t = 1$ (:a minimum observational interval) for simplicity. Fig. 4.1 exemplifies this model.

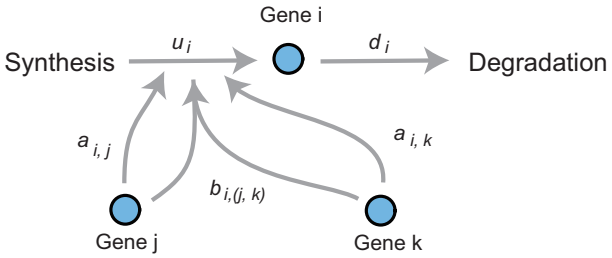

Figure 4.1: An example of the combinatorial transcription model regarding the ith gene. A gene undergoes synthesis and degradation processes, and its synthesis process is regulated through individual effects $a_{i,j}, a_{i,k}$ and a combinatorial effect $b_{i,(j,k)}$.

In order to assimilate a simulation model and observational data, we apply a nonlinear state space model [7, 43, 60, 66, 83]. Let $\boldsymbol{x}_t = (x_{1,t}, \ldots, x_{p,t})'$ be the vector of hidden variables and $\boldsymbol{y}_t$ be the observational data at time t. A state space representation of Eq. (4.2) is given by

$$\boldsymbol{x}_{t+1} = A\boldsymbol{x}_t + B\mathrm{vec}(\boldsymbol{x}_t\boldsymbol{x}_t') + \boldsymbol{u} + \boldsymbol{v}_t, \tag{4.3}$$

$$\boldsymbol{y}_t = \boldsymbol{x}_t + \boldsymbol{w}_t, \tag{4.4}$$

where $A = (\boldsymbol{a}_1, \ldots, \boldsymbol{a}_p)' \in R^{p \times p}$ is a linear effect matrix, $\boldsymbol{a}_i = (a_{i,1}, \ldots, a_{i,p})'$ $(i = 1, \ldots, p)$, $B = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_p)' \in R^{p \times p^2}$ is a combinatorial effect matrix, $\boldsymbol{b}_i = (b_{i,(1,1)}, \ldots, b_{i,(1,p)}, b_{i,(2,1)}, \ldots, b_{i,(p,p)})$ $(i = 1, \ldots, p)$, vec is a transformation function $(R^{p \times p} \rightarrow R^{p^2})$, $\boldsymbol{u} = (u_1, \ldots, u_p)'$, and $\boldsymbol{v}_t \sim N(0, Q)$ and $\boldsymbol{w}_t \sim N(0, R)$ are system and observational noises with diagonal covariance matri-

ces, respectively. We define an entire set of time points $\mathcal{T} = \{1, \ldots, T\}$and the observed time set $\mathcal{T}_{obs}$ ($\mathcal{T}_{obs} \subset \mathcal{T}$).

4.2.2 Unscented Kalman Filter

In Eqs. (4.3) and (4.4), conditional probability densities $P(\boldsymbol{x}_t|Y_{t-1})$, $P(\boldsymbol{x}_t|Y_t)$ and $P(\boldsymbol{x}_t|Y_T)$ can be non-Gaussian form, where $Y_t = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_t)$. Therefore, we apply UKF [16, 49, 51] to approximately obtain these conditional probability densities. The procedure is explained below.

Prediction and Filtering Steps

Let $\boldsymbol{x}_{t|s}$ and $V_{t|s}$ be the expectation and the covariance matrix, given observational data Y_s, at time t. For $t = 0, \ldots, T-1$,

1. Select sigma points $\boldsymbol{x}_{t|t}^{(n)}$ $(n = 0, \ldots, 2p)$ as

$$\boldsymbol{x}_{t|t}^{(n)} = \boldsymbol{x}_{t|t}, \qquad (n = 0), \tag{4.5}$$

$$\boldsymbol{x}_{t|t}^{(n)} = \boldsymbol{x}_{t|t} + \sqrt{(p+\lambda)\Sigma_{t|t}^{(n)}}, \qquad (n = 1, \ldots, p), \tag{4.6}$$

$$\boldsymbol{x}_{t|t}^{(n)} = \boldsymbol{x}_{t|t} - \sqrt{(p+\lambda)\Sigma_{t|t}^{(n-p)}}, \qquad (n = p+1, \ldots, 2p), \tag{4.7}$$

   where $\Sigma_{t|t}^{(n)}$ is the nth column vector of $\Sigma_{t|t}$ and $\lambda = \alpha^2(p+\kappa) - p$. Here, $\alpha^2 = 3/10$ and $\kappa = 0$ were applied to set $p + \lambda = 3$ [50].

2. Predict the next state of the generated sigma points $\boldsymbol{x}_{t|t}^{(n)}$ as $\boldsymbol{x}_{t+1|t}^{(n)}$ using the system equation of Eq. (4.3) without adding the system noise.

3. Calculate $\boldsymbol{x}_{t+1|t}$ and $V_{t+1|t}$ as

$$\boldsymbol{x}_{t+1|t} = \sum_{n=0}^{2p} \mathcal{W}_1^{(n)} \boldsymbol{x}_{t+1|t}^{(n)}, \tag{4.8}$$

$$\Sigma_{t+1|t} = \sum_{n=0}^{2p} \mathcal{W}_2^{(n)} (\boldsymbol{x}_{t+1|t}^{(n)} - \boldsymbol{x}_{t+1|t})(\boldsymbol{x}_{t+1|t}^{(n)} - \boldsymbol{x}_{t+1|t})' + Q, \tag{4.9}$$

$$\mathcal{W}_1^{(0)} = \frac{\lambda}{p+\lambda}, \tag{4.10}$$

$$\mathcal{W}_2^{(0)} = \frac{\lambda}{p+\lambda} + 1 - \alpha^2 + \beta, \tag{4.11}$$

$$\mathcal{W}_1^{(n)} = \mathcal{W}_2^{(n)} = \frac{1}{2(p+\lambda)}, \quad (n = 1, \ldots, 2p), \tag{4.12}$$

   where β is set 2 [48].

4. In the combinatorial model, the observational equation of Eq. (4.4) is a linear function. Then, we can apply the general Kalman filter algorithm [52, 60] to obtain the optimal conditional expectation and covariance matrix as follows

$$\boldsymbol{x}_{t+1|t+1} = \boldsymbol{x}_{t+1|t} + \Sigma_{t+1|t+1} R^{-1}(\boldsymbol{y}_{t+1} - \boldsymbol{x}_{t+1|t}), \tag{4.13}$$
$$\Sigma_{t+1|t+1} = (R^{-1} + \Sigma_{t+1|t}^{-1})^{-1}. \tag{4.14}$$

More details can be referred to [49, 51].

Smoothing Step

In order to obtain the conditional expectation and covariance matrix of the hidden state given full observational data Y_T, we apply the Rauch-Tung-Striebel (RTS) smoother for UKF [90]. The formulation of the RTS smoother is described as follows:

$$\boldsymbol{x}_{t|T} = \boldsymbol{x}_{t|t} + K_t(\boldsymbol{x}_{t+1|T} - \boldsymbol{x}_{t+1|t-1}), \tag{4.15}$$
$$\Sigma_{t|T} = \Sigma_{t|t} + K_t(\Sigma_{t+1|T} - \Sigma_{t+1|t-1})K_t', \tag{4.16}$$
$$K_t = C_t \Sigma_{t+1|t}^{-1}, \tag{4.17}$$
$$C_t = \sum_{n=0}^{2p} \mathcal{W}_2^{(n)} (\boldsymbol{x}_{t|t-1}^{(n)} - \boldsymbol{x}_{t|t-1})(\boldsymbol{x}_{t+1|t}^{(n)} - \boldsymbol{x}_{t+1|t})'. \tag{4.18}$$

Since we have $\boldsymbol{x}_{T|T}$ and $\Sigma_{T|T}$ after prediction and filtering steps, the above equations are recursively applied for $t = T-1, \ldots, 0$.

4.2.3 Parameter Estimation Using EM-algorithm

Let $X_T = \{\boldsymbol{x}_0, \ldots, \boldsymbol{x}_T\}$ be the set of state variables, and $\boldsymbol{\theta} = \{A, B, \boldsymbol{u}, Q, R, \boldsymbol{\mu}_0\}$ be the parameter vector. The log-likelihood of observational data is given by

$$\log L = \log \int P(\boldsymbol{x}_0) \prod_{t \in \mathcal{T}} P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) \prod_{t \in \mathcal{T}_{obs}} P(\boldsymbol{y}_t|\boldsymbol{x}_t) d\boldsymbol{x}_1 \ldots d\boldsymbol{x}_T, \tag{4.19}$$

where $P(\boldsymbol{x}_0)$ is a probability density of N-dimensional Gaussian distributions $N(\boldsymbol{\mu}_0, \Sigma_0)$, $P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$ and $P(\boldsymbol{y}_t|\boldsymbol{x}_t)$ can be probability densities of N-dimensional non-Gaussian distributions approximated by Eqs. (4.3) and (4.4) in Section 4.2.1 and the unscented transformation in Section 4.2.2.

In this chapter, we attempted to estimate the parameter vector $\boldsymbol{\theta}$ by maximizing Eq. (4.19) using the EM-algorithm [19]. Thus, the conditional expectation of the joint log-likelihood of the complete data (X_T, Y_T) at the lth iteration

$$q(\boldsymbol{\theta}|\boldsymbol{\theta}_l) = E[\log P(Y_T, X_T|\boldsymbol{\theta})|Y_T, \boldsymbol{\theta}_l], \tag{4.20}$$

is iteratively maximized with respect to $\boldsymbol{\theta}$ until convergence.

In the Expectation step, set conditional expectations of $\boldsymbol{x}_t$ as

$$V_t = \sum_{t\in\mathcal{T}}\sum_{n=0}^{2p} \mathcal{W}_2^{(n)} \boldsymbol{x}_{t|T}^{(n)} \boldsymbol{x}_{t|T}^{(n)\prime}, \tag{4.21}$$

$$V_{lag} = \sum_{t\in\mathcal{T}}\sum_{n=0}^{2p} \mathcal{W}_2^{(n)} \boldsymbol{x}_{t|T}^{(n)} \boldsymbol{x}_{t-1|T}^{(n)\prime}, \tag{4.22}$$

$$V_{t-1} = \sum_{t\in\mathcal{T}}\sum_{n=0}^{2p} \mathcal{W}_2^{(n)} \boldsymbol{x}_{t-1|T}^{(n)} \boldsymbol{x}_{t-1|T}^{(n)\prime}, \tag{4.23}$$

$$\Phi_{lag} = \sum_{t\in\mathcal{T}}\sum_{n=0}^{2p} \mathcal{W}_2^{(n)} \boldsymbol{x}_{t|T}^{(n)} \mathrm{vec}(\boldsymbol{x}_{t-1|T}^{(n)} \boldsymbol{x}_{t-1|T}^{(n)\prime})', \tag{4.24}$$

$$\Phi_{t-1} = \sum_{t\in\mathcal{T}}\sum_{n=0}^{2p} \mathcal{W}_2^{(n)} \boldsymbol{x}_{t-1|T}^{(n)} \mathrm{vec}(\boldsymbol{x}_{t-1|T}^{(n)} \boldsymbol{x}_{t-1|T}^{(n)\prime})', \tag{4.25}$$

$$\Psi_{t-1} = \sum_{t\in\mathcal{T}}\sum_{n=0}^{2p} \mathcal{W}_2^{(n)} \mathrm{vec}(\boldsymbol{x}_{t-1|T}^{(n)} \boldsymbol{x}_{t-1|T}^{(n)\prime}) \mathrm{vec}(\boldsymbol{x}_{t-1|T}^{(n)} \boldsymbol{x}_{t-1|T}^{(n)\prime})', \tag{4.26}$$

$$\boldsymbol{s}_t = \sum_{t\in\mathcal{T}}\sum_{n=0}^{2p} \mathcal{W}_1^{(n)} \boldsymbol{x}_{t|T}^{(n)}, \tag{4.27}$$

$$\boldsymbol{s}_{t-1} = \sum_{t\in\mathcal{T}}\sum_{n=0}^{2p} \mathcal{W}_1^{(n)} \boldsymbol{x}_{t-1|T}^{(n)}, \tag{4.28}$$

$$\boldsymbol{s}_{t-1}^2 = \sum_{t\in\mathcal{T}}\sum_{n=0}^{2p} \mathcal{W}_1^{(n)} \mathrm{vec}(\boldsymbol{x}_{t-1|T}^{(n)} \boldsymbol{x}_{t-1|T}^{(n)\prime})'. \tag{4.29}$$

In the Maximization-step, $\boldsymbol{\theta}_l$ is updated to $\boldsymbol{\theta}_{l+1} = \arg\max_{\boldsymbol{\theta}} q(\boldsymbol{\theta}|\boldsymbol{\theta}_l)$. Let $\boldsymbol{v}_{lag,i}$, $\boldsymbol{\phi}_{lag,i}$ and $\boldsymbol{\phi}_{t-1,i}$ be a transpose of the ith row vector of V_{lag}, Φ_{lag} and Φ_{t-1}, respectively. Then, $\boldsymbol{\theta}$ is updated as

$$\boldsymbol{a}_i^{\mathcal{A}} = V_{t-1}^{\mathcal{A}^{-1}} (\boldsymbol{v}_{lag,i}^{\mathcal{A}} - \phi_{t-1}^{\mathcal{A}\times\mathcal{B}} \boldsymbol{b}_i^{\mathcal{B}} - u_i \boldsymbol{s}_{t-1}^{\mathcal{A}}), \tag{4.30}$$

$$\boldsymbol{b}_i^{\mathcal{B}} = \Psi_{t-1}^{\mathcal{B}^{-1}} (\boldsymbol{\phi}_{lag,i}^{\mathcal{B}} - \phi_{t-1}^{\mathcal{A}\times\mathcal{B}'} \boldsymbol{a}_i^{\mathcal{A}} - u_i \boldsymbol{s}^{2^{\mathcal{B}}}), \tag{4.31}$$

$$\boldsymbol{u} = \frac{\boldsymbol{s}_t - A\boldsymbol{s}_{t-1} - B\boldsymbol{s}^2}{T}, \tag{4.32}$$

$$Q = \frac{1}{T}\sum_{t=1}^{T} E[(\boldsymbol{x}_t - A\boldsymbol{x}_{t-1} - B\mathrm{vec}(\boldsymbol{x}_{t-1}\boldsymbol{x}'_{t-1}) - \boldsymbol{u}) \cdot (\boldsymbol{x}_t - A\boldsymbol{x}_{t-1} - B\mathrm{vec}(\boldsymbol{x}_{t-1}\boldsymbol{x}'_{t-1}) - \boldsymbol{u})'|Y_T], \tag{4.33}$$

$$\boldsymbol{\mu}_0 = \boldsymbol{x}_{0|T}, \tag{4.34}$$

$$R = \frac{1}{T}\sum_{t\in\mathcal{T}_{obs}} \{(\boldsymbol{y}_t - \boldsymbol{x}_{t|T})(\boldsymbol{y}_t - \boldsymbol{x}_{t|T})' + \Sigma_{t|T}\}, \tag{4.35}$$

where $\mathcal{A}$ and $\mathcal{B}$ are active sets of elements for A and B, respectively. For example, $\boldsymbol{a}_i^{\mathcal{A}}$ is an $|\mathcal{A}|$-dimensional vector consisting of elements regulating the ith gene.

4.2.4 Network Restoration Algorithm

When an original gene regulatory network $\mathcal{M}_{original}$ is given, the purpose is to find the model $\mathcal{M}_{best}$ that can best predict observational data. Here, the prediction ability of a model $\mathcal{M}$ is evaluated using BIC [96] described by

$$BIC = -2\log L + D\log\nu, \tag{4.36}$$

where D and ν are the number of samples and the non-zero parameters, respectively. The derivation of BIC is briefly introduced in Chapter 2. Due to the high computational cost involved in estimating the values of the parameters $\boldsymbol{\theta}$ for $\mathcal{M}$, we can not evaluate all candidate models. Therefore, starting from $\mathcal{M}_{original}$, one strategy is to sequentially evaluate candidate models that are constructed by changing a part of the regulatory structure of the current model $\mathcal{M}_{current}$ of which prediction ability is the best among evaluated ones. In this paradigm, we consider three operations, *i.e.*, adding, deleting and replacing a regulation, which are shown in Fig. 4.2, and the constraints add_{max} and del_{max}, which restrict the number of additional and deleted regulations from $\mathcal{M}_{original}$. Then, we propose a novel algorithm, which can efficiently evaluate only highly possible candidates, for improving and extending GRNs to obtain $\mathcal{M}_{best}$ as concluded in Algorithms 2-4. In these algorithm, we consider a function for measuring the possibility of the Model $\mathcal{M}$ that is added or deleted a regulation to the ith gene from $\mathcal{M}_{current}$ as

$$e(\mathcal{M}, i) = \boldsymbol{a}_i' V_{t-1}\boldsymbol{a}_i - 2\boldsymbol{v}_{lag,i}\boldsymbol{a}_i + 2\boldsymbol{b}_i'\phi'_{t-1}\boldsymbol{a}_i + 2u_i\boldsymbol{s}'_{t-1}\boldsymbol{a}_i. \tag{4.37}$$

To measure the effectiveness of the candidate models when changing active sets, Eq. (4.37) of which active sets are changed as those of the next candidate is calculated. Then, only for r top models with respect to $-e(\mathcal{M}, i)$ for each i, the BIC scores are evaluated by estimating the parameter values maximizing prediction ability for observational data using UKF and the EM-algorithm. This procedure is shown in Fig. 4.3. Note that $e(\mathcal{M}, i)$ can be derived when calculating $\arg\max_{\boldsymbol{a}_i} q(\boldsymbol{\theta}|\boldsymbol{\theta}_l)$.

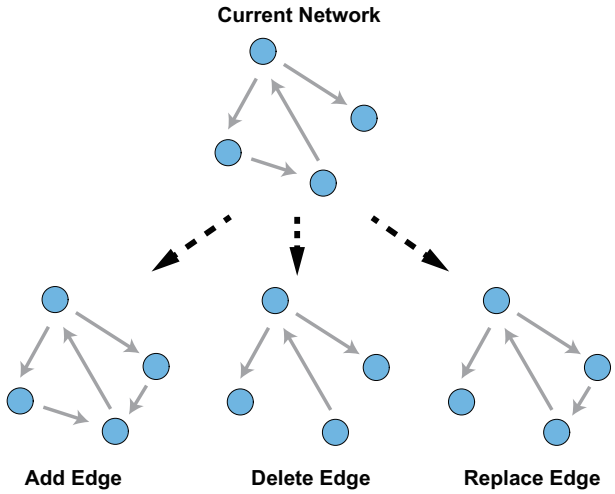

Figure 4.2: The operations of changing the current network. We consider the three types of operations for an improvement of GRN, *i.e.*, 'Add Edge' (adding), 'Delete Edge' (deleting) and 'Replace Edge' (replacing). Under the constraints of add_{max} and del_{max}, these operations are recursively executed until the network cannot be changed through these operations to decrease the BIC score.

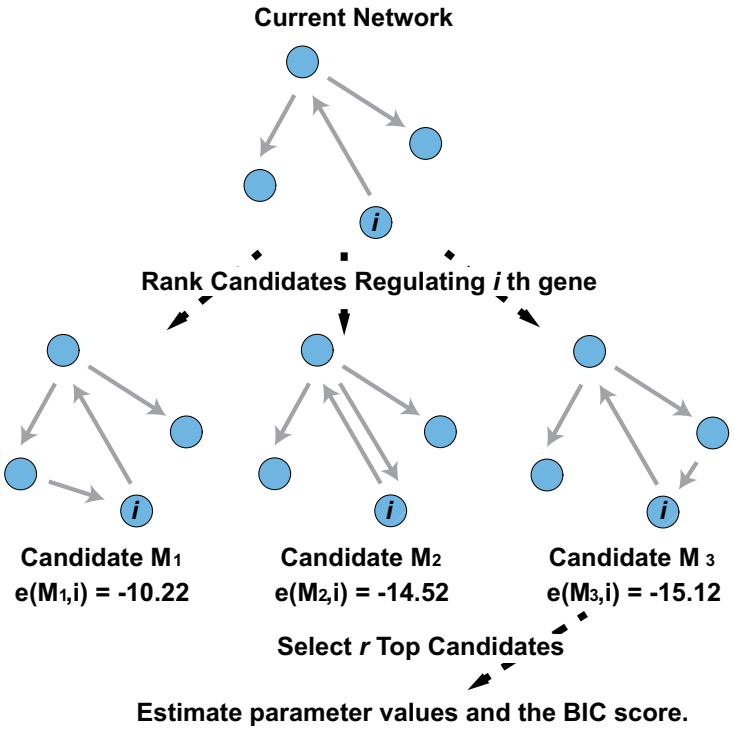

Figure 4.3: A cartoon figure of the proposed algorithm. For the current network, the proposed algorithm constructs candidates networks by adding, deleting and replacing edges and ranks them using $e(\mathcal{M}, i)$. Then, only r top networks with respect to $-e(\mathcal{M}, i)$ are evaluated the BIC score by estimating the parameter values.

Algorithm 2 The proposed algorithm for improving GRNs based on the approximate posterior probability.

Set r;
$add \leftarrow 0$; $del \leftarrow 0$; $flag \leftarrow 0$
$BIC_{current} \leftarrow$ the BIC score of the original model;
$\mathcal{M}_{current} \leftarrow \mathcal{M}_{original}$;
while $flag < N^2$ **do**
  for $i = 1$ to N **do**
    for $j = 1$ to N **do**
      if $A_{i,j}$ of $\mathcal{M}_{current} = 0$ **then**
        $changed \leftarrow$ Execute sub-algorithm 1;
      else
        $changed \leftarrow$ Execute sub-algorithm 2;
      end if
      if *changed* **then**
        $flag \leftarrow 0$;
      else
        $flag \leftarrow flag + 1$;
      end if
      if $flag >= N^2$ **then**
        break;
      end if
    end for
    if $flag >= N^2$ **then**
      break;
    end if
  end for
end while

Algorithm 3 Sub-algorithm 1

$changed \leftarrow FALSE$;
Consider $\mathcal{M}_{candidate}$ that is constructed from $\mathcal{M}_{candidate}$ by setting a regulation to the ith gene by the jth gene as an active element;
Estimate the parameter values and obtain the BIC score $BIC_{candidate}$ by the UKF and the EM-algorithm;
if $BIC_{current} > BIC_{candidate}$ and $add_{max} > add$ **then**
  Set $\mathcal{M}_{candidate}$ as $\mathcal{M}_{current}$; $BIC_{candidate} \leftarrow BIC_{current}$;
  $changed \leftarrow TRUE$;
else
  for $i = 1$ to N **do**
    for $k = 1$ to r **do**
      $j \leftarrow$ the kth minimum element with respect to $e(i, j_{col})$ $(j_{col} = 1, \ldots, N)$ of $\mathcal{M}_{candidate}$;
      if $A_{i,j}$ of $\mathcal{M}_{candidate}$ is 0 **then**
        continue;
      end if
      if $A_{i,j}$ of $\mathcal{M}_{original}$ is 1 or $add_{max} > add$ **then**
        continue;
      end if
      Consider $\mathcal{M}_{candidate2}$ that is constructed from $\mathcal{M}_{candidate}$ by setting a regulation to the ith gene by the jth gene as a non-active set;
      Estimate the parameter values and obtain the BIC score by the UKF and the EM-algorithm;
    end for
  end for
  if $BIC_{current} >$ the minimum BIC score among models calculated above **then**
    Set $\mathcal{M}_{current}$ and $BIC_{current}$ as those of the minimum one;
    $changed \leftarrow TRUE$;
  end if
end if
Set add and del as those of the $\mathcal{M}_{current}$;
return $changed$;

Algorithm 4 Sub-algorithm 2

1: $changed \leftarrow FALSE$;
2: Consider $\mathcal{M}_{candidate}$ that is constructed from $\mathcal{M}_{candidate}$ by setting a regulation to the ith gene by the jth gene as a non-active element;
3: Estimate the parameter values and obtain the BIC score $BIC_{candidate}$ by the UKF and the EM-algorithm;
4: **if** $BIC_{current} > BIC_{candidate}$ and $del_{max} > del$ **then**
5: Set $\mathcal{M}_{candidate}$ as $\mathcal{M}_{current}$; $BIC_{candidate} \leftarrow BIC_{current}$;
6: $changed \leftarrow TRUE$;
7: **else**
8: **for** $i = 1$ to N **do**
9: **for** $k = 1$ to r **do**
10: $j \leftarrow$ the kth minimum element with respect to $e(i, j_{col})$ $(j_{col} = 1, \ldots, N)$ of $\mathcal{M}_{candidate}$;
11: **if** $A_{i,j}$ of $\mathcal{M}_{candidate}$ is 1 **then**
12: continue;
13: **end if**
14: **if** $A_{i,j}$ of $\mathcal{M}_{original}$ is 0 or $add_{del} > del$ **then**
15: continue;
16: **end if**
17: Consider $\mathcal{M}_{candidate2}$ that is constructed from $\mathcal{M}_{candidate}$ by setting a regulation to the ith gene by the jth gene as an active set;
18: Estimate the parameter values and obtain the BIC score by the UKF and the EM-algorithm;
19: **end for**
20: **end for**
21: **if** $BIC_{current} >$ the minimum BIC score among models calculated above **then**
22: Set $\mathcal{M}_{current}$ and $BIC_{current}$ as those of the minimum one;
23: $changed \leftarrow TRUE$;
24: **end if**
25: **end if**
26: Set add and del as those of the $\mathcal{M}_{current}$;
27: **return** $changed$;

4.3 Results

4.3.1 Comparison Analysis Using Synthetic Data of WNT5A and Yeast Network

To show the effectiveness of the proposed algorithm, we used artificial time-course gene expression data from two synthetic networks of WNT5A [55] and a yeast cell cycle network [53] as illustrated in Figs. 4.4 and 4.5, respectively. For each network, we generated two time-courses consisting of $\mathcal{T} = \{1, 2, \ldots, 30\}$ and $\{1, 2, \ldots, 10, 12, \ldots, 30\}$ by using Eqs. (4.3) and (4.4). For Eq. (4.3), the values of the parameters were determined between 0 and 1, and the system noise was according to Gaussian distribution with a mean 0 and a variance 0.1. In Eq. (4.4), Gaussian observational noise with a mean of 0 and a variance of 0.3 were added to these artificial data. Note that the networks were used for the performance comparison in the previous study [78].

For this comparison, we applied (a) the proposed method, (b) a regression-based method (DPLSQ) [78], (c) DPLSQ with the BIC [96], (d) Akaike information criterion (AIC) [1] to the data sets. Here, since DPLSQ is based only on the least-square errors, it may infer many false positives. Then, we modified the algorithm to use BIC and AIC. r in the proposed algorithm is set 3.

For each data set, 10 trials were executed, for each of which the true network of Figs. 4.4 and 4.5 is randomly modified and given as an original network. Thus, a network obtained by adding and deleting 5 edges from the true network was given as an original network and then the (a)-(d) were applied to obtain the true network. We evaluated the average performance of true positive (TP), false positive (FP), true negative (TN), false negative (FN), precision rate ($PR = \frac{\mathrm{TP}}{\mathrm{TP+FP}}$), recall rate ($RR = \frac{\mathrm{TP}}{\mathrm{TP+FN}}$) and F-measure ($=\frac{2\mathrm{PR}\cdot\mathrm{RR}}{\mathrm{PR+RR}}$) over 10 trials for each data. In contrast to a usual way, we counted TP when an altered edge was successfully improved as the true model, FN when an altered edge was not improved, and FP when an edge in the true model was changed in the improved model. The results of using the 4 time-courses are summarized in Tables 4.1-4.4 (the proposed method is noted as 'UKF-Completion'), respectively. These results clearly show that the proposed algorithm has the highest performance as compared to the other methods for all data sets. In particular, for non-equally spaced time-point data, the proposed method could better infer true regulations than the previous methods since our approach utilizes the hidden state and can handle non-observational time point.

Table 4.1: Comparison of the proposed method and DPLSQ using equally spaced artificial data from WNT5A network.

	PR	RR	F-measure	TP	FP	TN	FN
DPLSQ	0.580	0.290	0.386	2.9	2.1	87.9	7.1
DPLSQ (BIC)	0.677	0.670	0.673	6.7	3.2	86.8	3.3
DPLSQ (AIC)	0.700	0.650	0.673	6.5	2.8	87.2	3.5
UKF-Completion	0.760	0.760	0.760	7.6	2.4	87.6	2.4

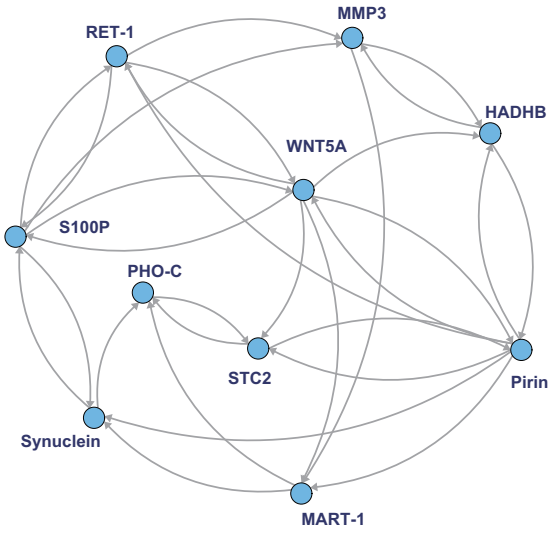

Figure 4.4: A real biological network termed WNT5A network [55], used for the comparison analysis. Based on the network, the original networks are generated by randomly adding and deleting 5 edges.

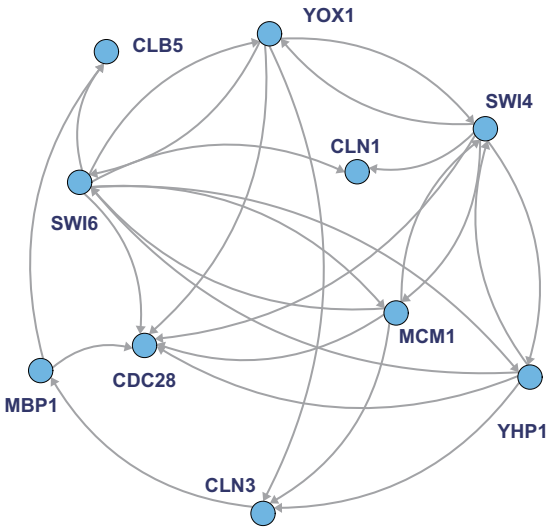

Figure 4.5: A real biological network of yeast cell cycle from the KEGG database [53, 78] used for the comparison analysis. Based on the network, the original networks are generated by randomly adding and deleting 5 edges.

Table 4.2: Comparison of the proposed method and DPLSQ using non-equally spaced artificial data from WNT5A network.

	PR	RR	F-measure	TP	FP	TN	FN
DPLSQ	0.540	0.270	0.360	2.7	2.3	87.7	7.3
DPLSQ (BIC)	0.520	0.520	0.520	5.2	4.8	85.2	4.8
DPLSQ (AIC)	0.523	0.49	0.506	4.9	4.5	85.5	5.1
UKF-Completion	0.720	0.720	0.720	7.2	2.8	87.2	2.8

Table 4.3: Comparison of the proposed method and DPLSQ using equally spaced artificial data from a yeast cell cycle network.

	PR	RR	F-measure	TP	FP	TN	FN
DPLSQ	0.600	0.300	0.400	3.0	2.0	88.0	7.0
DPLSQ (BIC)	0.597	0.590	0.593	5.9	4.0	86.0	4.1
DPLSQ (AIC)	0.600	0.600	0.600	6.0	4.0	86.0	4.0
UKF-Completion	0.650	0.650	0.650	6.5	3.5	86.5	3.5

4.3.2 Real Data Analysis Using Yeast Cell Cycle Network

As an application example of improving and extending literature-based networks, we dealt with a yeast cell-cycle network from KEGG [53] and used the corresponding observational data [103]. By using time-course data including 25 genes of which regulatory relationships are represented as red arrows in Fig. 4.6, and considering this as an original network, we attempted to improve the network. However, since the network is classical and highly reliable in KEGG database, we focused on the extension of the network using additional genes. Thus, we considered the network consisting of these 25 genes and 38 additional candidate genes, which can have functions related to a yeast cell cycle pathway, from SGD [15]. We did not set prior regulatory structure to these 38 genes and extended the KEGG-based network consisting of 25 genes by adding regulations to these 38 genes ($del_{max} = 0$).

Consequently, 38 candidate genes were integrated in the KEGG-based yeast cell cycle network as illustrated in Fig. 4.6. In this figure, the KEGG-based regulatory network consisting of 25 genes was drawn as rectangles (gene) and red arrows (regulation), and newly estimated relationships were drawn as circles (gene) and black chained arrows (regulation). Interestingly, there exist many combinatorial regulations of which regulated genes have more than two regulations. Since these regulations can have non-zero values of the combinatorial effect $b_{i,(j,k)}$, the results may not be obtained by linear models. Furthermore, some genes such as $YOX1$ and $Cdc6$, becomes hub gene regulating many other genes and they are known as upper stream genes regulating down stream genes on the KEGG database. These results show the possibility of the causal relationships between them.

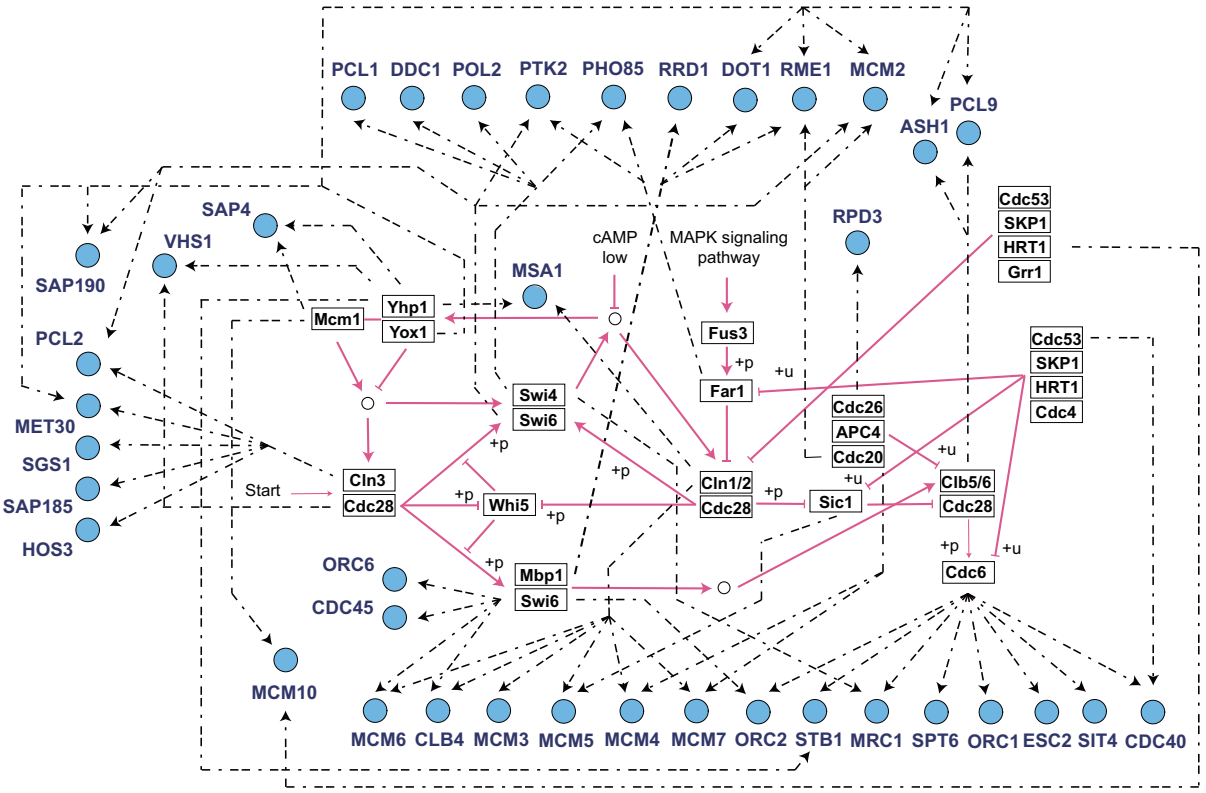

Figure 4.6: A part of a yeast cell cycle network and candidate genes for extending the network. The KEGG-based regulatory network consisting of 25 genes is drawn as rectangles (gene) and red arrows (regulation), and newly estimated relationships are drawn as circles (gene) and black chained arrows (regulation).

Table 4.4: Comparison of the proposed method and DPLSQ using non-equally spaced artificial data from a yeast cell cycle network.

	PR	RR	F-measure	TP	FP	TN	FN
DPLSQ	0.440	0.220	0.293	2.2	2.8	87.2	7.8
DPLSQ (BIC)	0.400	0.400	0.400	4.0	6.0	84.0	6.0
DPLSQ (AIC)	0.400	0.400	0.400	4.0	6.0	84.0	6.0
UKF-Completion	0.550	0.550	0.550	5.5	4.5	85.5	4.5

4.4 Discussion

We proposed a genomic data assimilation schema using a nonlinear simulation model for improving and extending literature-based networks. The method can efficiently estimate parameter values of a simulation model by using the EM-algorithm with UKF. Furthermore, the proposed algorithm avoids to evaluate all possible candidates that are constructed by modifying the original network and selects only plausible ones through measuring the effectiveness when modifying the regulation of the current network for the data. Therefore, this schema makes it possible to deal with many candidate networks and finds better networks for the data.

The performance of this approach was demonstrated by implementing artificial simulation data from real biological networks termed WNT5A and a yeast cell cycle network. Consequently, our proposed method can evaluate GRNs more accurately than could a previously developed method (DPLSQ). In particular, since our method is based on the state space representation using the hidden state for representing gene regulatory dynamics, the flexibility for the observational data, *i.e.*, which can handle observational data with non-equally spaced time points, can be ensured. These results indicated the high performance and adaptability of the proposed method to improve and extend the original network using time-course observational data. As an application example, using a part of a well-investigated yeast cell-cycle network from KEGG, we applied the proposed method to extend the network by integrating additional candidate genes from SGD [15]. Interestingly, we found hub genes regulating candidate genes that are indicated as upstream genes in KEGG database. Since these are biologically related candidates of the original networks, these extensions might be true regulations and thus should be confirmed by biological experiments.

Chapter 5

Genomic Data Assimilation Using a Higher Moment Filtering Technique for Restoration of Gene Regulatory Networks

5.1 Background

GRNs are fundamental for sustaining complex biological systems in cells. Although a comprehensive understanding of intracellular systems is still far from complete, many findings regarding intracellular systems have been published as a result of recent technological advances in biotechnology, *e.g.*, microarray and Chip-Seq. By combining these findings, we can construct biological simulation models in which the dynamics of biomolecules are described by mathematical equations, *e.g.*, the Michaelis-Menten model [91] and S-system [92]. However, simulation results may not match results from biological observations due to inaccurate or missing information about intracellular systems.

In oder to infer unknown parts of biological systems, there exist roughly two major approaches, *i.e.*, simulation model-based and statistical approaches. In constructing biological simulation models, regulatory relationships among biomolecules are collected from the literature. To represent the regulatory systems, mathematical equations, often differential equations [18, 24, 91, 92], are given to describe the dynamic behavior of the involved biomolecules. The parameter values of these simulation models have been estimated to be consistent with the data by some computational methodologies. Several methods have been proposed to infer regulatory structures [42, 81], to reproduce the dynamic behavior of biological systems recorded in the literature [59, 73, 79, 83, 85], and to improve published pathways so that they are consistent with the data [39, 40]. However, since differential equation-based approaches are computationally intensive, when updating parameter values and simulation results simultaneously, they cannot be applied to more than several genes when much of the regulatory structure is unknown.

A statistical approach using more abstracted models, *e.g.*, Bayesian networks [29,54,118,120] and SSMs [9, 12, 43, 86], have been successfully applied to infer the structure of transcriptional regulation using data from biological observations. Whereas purely data-driven methods need to explore a large model space, some studies have further incorporated other information, *e.g.*, literature-recorded pathways and TF information [7, 20, 22, 88, 109]. In contrast, these approximations can generate false regulations; there is a trade-off relationship between accuracy and computational ease. To overcome the problem, methods to improve and deconvolve networks, which are inferred by some computational approaches, utilizing less abstract models to better predict the data have been also proposed recently [10, 27, 78]. In following the direction, we should apply models that can emulate the nonlinear dynamics of gene regulatory networks and establish a method for estimating the parameter values that maximize the ability to predict the data.

For this purpose, we proposed a novel data assimilation algorithm utilizing a simple nonlinear model, termed the combinatorial transcription model [81, 110], and a state space representation [52, 101], to infer GRNs by restoring networks inferred by some GRNs inference methods or literature-derived networks. Since the nonlinearity results in generating non-Gaussian posterior distributions of the hidden state variables, we applied the unscented Kalman filter (UKF) [16, 49, 51] that can efficiently calculate the approximated posterior distributions as Gaussian distributions. However, UKF cannot satisfy the requirements for estimating accurate parameter values of the model; thus the first four moments of the posterior distributions of the hidden states should be retained. To address this problem, we developed a novel method, termed a higher-moment ensemble particle filter (HMEnPF), which can retain the first two moments and the third and fourth central moments throughout the prediction, filtering, and smoothing steps. Combining UKF and HMEnPF, the proposed algorithm improves and extends the original model, which are derived from the literature and some GRNs inference algorithms, based on the nonlinear state space model. A criterion that can rank candidate models, which are generated by partially changing the current best model before evaluating them, enables us to evaluate only plausible candidates within large model space to explore the best model. Furthermore, the combinatorial transcription model was extended so that the model can handle additional biomolecules such as drugs.

To show the effectiveness of the proposed algorithm, we first used synthetic data and compared the proposed algorithm to previous methods, GeneNet [80, 94] based on an empirical graphical Gaussian model (GGM) and G1DBN [63] based on dynamic Bayesian networks using first order conditional dependencies, and the proposed algorithm using UKF only. For the comparison, synthetic data with 30 time-points were generated for a WNT5A [55] and a yeast-cell-cycle network [53]. As an application example, we prepared the time-course microarray data after stimulating rat skeletal muscle with corticosteroid, which were downloaded from the GEO database (GSE490). For this experiment, we utilized corticosteroid pharmacogenomics [5, 115], a previously defined regulatory structure for rat skeletal muscle [100], TF information from ITFP [122] and the original network inferred by G1DBN. Consequently, we proposed candidate

pathways for the extension of corticosteroid-related pathways.

5.2 Methods

5.2.1 A State Space Representation of Combinatorial Transcription Model

Under the framework of data assimilation, in order to combine the simulation results with the observed experimental data, we apply a state space representation of the combinatorial transcription model given by

$$\boldsymbol{x}_{t+1} = A\boldsymbol{x}_t + B\mathrm{vec}(\boldsymbol{x}_t\boldsymbol{x}_t') + \boldsymbol{u} + \boldsymbol{v}_t, \tag{5.1}$$

$$\boldsymbol{y}_t = \boldsymbol{x}_t + \boldsymbol{w}_t, \tag{5.2}$$

that are introduced in Chapter 4. Note that A and B should be sparse matrices, and we also consider an active set of elements $\mathcal{B}_i$ $(i = 1, \ldots, p)$, which are sets of non-zero columns in the ith row of B.

5.2.2 Incorporation of Biomolecules Affecting Biological Systems

Although the regulatory system of Eqs. (5.1) and (5.2) can only represent dynamic regulation among genes, other biomolecules, such as drugs, can affect the regulatory system. To address these cases, we further consider a term representing the concentration of other biomolecules, as represented by

$$\boldsymbol{x}_t = A\boldsymbol{x}_{t-1} + B\mathrm{vec}(\boldsymbol{x}_t\boldsymbol{x}_t') + G\boldsymbol{d}_{t-1} + \boldsymbol{u} + \boldsymbol{v}_t, \tag{5.3}$$

where $\boldsymbol{d}_t$ is an M-dimensional vector containing the concentration of the biomolecules at the tth time point, $G = (\boldsymbol{g}_1, \ldots, \boldsymbol{g}_p)'$ is an $p \times M$ matrix and $\boldsymbol{g}_i = (g_{i,1}, \ldots, g_{i,M})'$ $(i = 1, \ldots, p)$ is an M-dimensional vector representing their regulatory effects on the ith gene. As with $\mathcal{A}_i$ and $\mathcal{B}_i$, we consider the active sets of elements $\mathcal{G}_i$ for the ith row of the drug effect G. This model of Eq. (5.3) is exemplified in Fig. 5.1.

5.2.3 A Higher-Moment Ensemble Particle Filter

In Eqs. (5.1) and (5.2), conditional probability densities $P(\boldsymbol{x}_t|Y_{t-1})$, $P(\boldsymbol{x}_t|Y_t)$ and $P(\boldsymbol{x}_t|Y_T)$ can be non-Gaussian form, where $Y_t = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_t)$. Therefore, we applied an ensemble approximation, which is a type of Monte Carlo approach, to approximate these densities. In this approach, for example, $p(\boldsymbol{x}_t)$ is approximated by

$$p(\boldsymbol{x}_t) = \frac{1}{N}\sum_{n=1}^{N}\delta(\boldsymbol{x}_t - \boldsymbol{x}_t^{(n)}), \tag{5.4}$$

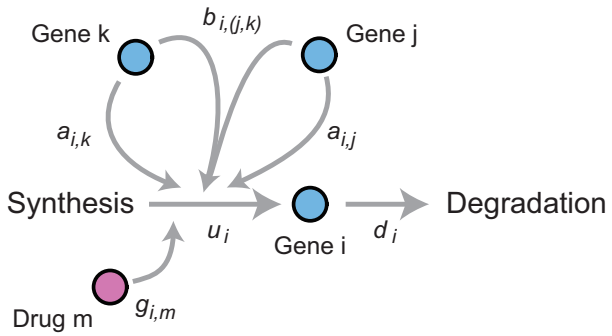

Figure 5.1: A cartoon figure of the combinatorial transcription model regarding the ith gene. A gene undergoes synthesis and degradation processes, and its synthesis process is regulated through individual effects $a_{i,j}, a_{i,k}$ and a combinatorial effect $b_{i,(j,k)}$.

where $\boldsymbol{x}_t^{(n)}$ is the nth sample from $p(\boldsymbol{x}_t)$, N is the number of samples and δ is a Dirac delta function. A sample $\boldsymbol{x}_t^{(n)}$ and a set of samples $\{\boldsymbol{x}_t^{(n)}\}$ are called particle and ensemble, respectively. Previously, many types of ensemble approximation methods have been developed to obtain conditional distributions in a state space model, *e.g.*, the ensemble Kalman filter (EnKF) [25] and the particle filter (PF) [36, 57]. Here, our requirements for this study are the following; (i) particles should survive through filtering steps even for the high-dimensional hidden variables $\boldsymbol{x}_t$ and the parameter vector $\boldsymbol{\theta}$, (ii) the third and fourth moments of probability densities should be kept in order to optimize $\boldsymbol{\theta}$ as explained in the next sub-section. To meet these requirements, we extended a method termed the ensemble particle filter (EnPF) [6, 82], which can keep the first two moments through filtering steps, and developed a novel method termed a HM-EnPF that can additionally retain third and fourth central moments without reducing particles. The procedure of the proposed method is explained below.

5.2.3.1 Prediction Step

Let $\boldsymbol{x}_{t|t}^{(n)}$ be a sample from a conditional probability density $P(\boldsymbol{x}_t|Y_t)$. Initially, generate particles $\boldsymbol{x}_{0|0}^{(n)} \sim p(\boldsymbol{x}_0)$ for $n = 1, \ldots, N$. Then, for $t = 1, \ldots, T$,

1. Generate particles $\boldsymbol{v}_t^{(n)} \sim N(0, Q)$ for $n = 1, \ldots, N$.
2. Calculate $\boldsymbol{x}_{t+1|t}^{(n)}$ by applying $\boldsymbol{x}_{t|t}^{(n)}$ and $\boldsymbol{v}_t^{(n)}$ to Eq. (5.1) for $n = 1, \ldots, N$.

5.2.3.2 Filtering Step

It consists of the following three sub-steps. At tth ($t \in \mathcal{T}_{obs}$) time step,

1. Particle Filter Step

(a) Resample $\hat{\boldsymbol{x}}_{t|t}^{(n)}$ according to

$$p(\boldsymbol{x}_t|\boldsymbol{Y}_t) = \frac{1}{\sum_{\dot{n}} p(\boldsymbol{y}_t|\boldsymbol{x}_{t|t-1}^{(\dot{n})})} \sum_{n=1}^{N} p(\boldsymbol{y}_t|\boldsymbol{x}_{t|t-1}^{(n)}) \delta(\boldsymbol{x}_t - \boldsymbol{x}_{t|t-1}^{(n)}). \tag{5.5}$$

(b) Calculate first and second moments $\mu_{t|t} = E[\{\hat{\boldsymbol{x}}_{t|t}^{(n)}\}]$ and $V_{t|t} = Var[\{\hat{\boldsymbol{x}}_{t|t}^{(n)}\}]$, respectively.

(c) Standardize $\hat{\boldsymbol{x}}_{t|t}^{(n)}$ as

$$\hat{\boldsymbol{z}}_{t|t}^{(n)} = V_{t|t}^{-\frac{1}{2}} \cdot (\hat{\boldsymbol{x}}_{t|t}^{(n)} - \boldsymbol{\mu}_{t|t}). \tag{5.6}$$

(d) Calculate third and fourth central moments $\hat{\boldsymbol{m}}_{t|t}^{(3)} = E[\{\hat{\boldsymbol{z}}_{t|t}^{(n)}\}^3]$ and $\hat{\boldsymbol{m}}_{t|t}^{(4)} = E[\{\hat{\boldsymbol{z}}_{t|t}^{(n)}\}^4]$, respectively.

2. Ensemble Kalman Filter Step

(a) Generate particles $\boldsymbol{w}_t^{(n)} \sim N(0, R)$ for $n = 1, \ldots, N$.

(b) Calculate Kalman gain

$$K_t = V_{t|t-1}(V_{t|t-1} + R_t)^{-1}, \tag{5.7}$$

where $V_{t|t-1} = Var[\{\boldsymbol{x}_{t|t-1}^{(n)}\}]$ and $R_t = Var[\{\boldsymbol{w}_t^{(n)}\}]$.

(c) Calculate $\tilde{\boldsymbol{x}}_{t|t}^{(n)}$ as

$$\tilde{\boldsymbol{x}}_{t|t}^{(n)} = \boldsymbol{x}_{t|t-1}^{(n)} + K_t(\boldsymbol{y}_t - \boldsymbol{x}_{t|t-1}^{(n)} + \boldsymbol{w}_t^{(n)}). \tag{5.8}$$

(d) Calculate first and second moments $\tilde{\mu}_{t|t} = E[\{\tilde{\boldsymbol{x}}_{t|t}^{(n)}\}]$ and $\tilde{V}_{t|t} = Var[\{\tilde{\boldsymbol{x}}_{t|t}^{(n)}\}]$, respectively.

(e) Standardize $\tilde{\boldsymbol{x}}_{t|t}^{(n)}$ as

$$\tilde{\boldsymbol{z}}_{t|t}^{(n)} = \tilde{V}_{t|t}^{-\frac{1}{2}} \cdot (\tilde{\boldsymbol{x}}_{t|t}^{(n)} - \tilde{\boldsymbol{\mu}}_{t|t}). \tag{5.9}$$

(f) Calculate third and fourth central moments $\tilde{\boldsymbol{m}}_{t|t}^{(3)} = E[\{\tilde{\boldsymbol{z}}_{t|t}^{(n)}\}^3]$ and $\tilde{\boldsymbol{m}}_{t|t}^{(4)} = E[\{\tilde{\boldsymbol{z}}_{t|t}^{(n)}\}^4]$, respectively.

3. Merging Step

Here, we needed to use a standardization function $S(\boldsymbol{\gamma}, \boldsymbol{\alpha}, \boldsymbol{\beta})$ that transforms a normal random vector $\boldsymbol{\gamma}$ into a normalized random vector $\boldsymbol{x}$ whose the third and fourth central moments are $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, respectively. Since a previous study [121] had proposed a standardization function satisfying the requirements, we applied this function as $S(\boldsymbol{\gamma}, \boldsymbol{\alpha}, \boldsymbol{\beta})$.

Then, we obtained $\boldsymbol{x}_{t|t}^{(n)}$ as

$$\boldsymbol{x}_{t|t}^{(n)} = \hat{V}_{t|t}^{\frac{1}{2}} S(\boldsymbol{z}_t^{(n)}, \hat{\boldsymbol{m}}_{t|t}^{(3)}, \hat{\boldsymbol{m}}_{t|t}^{(4)}) + \hat{\boldsymbol{\mu}}_{t|t}, \tag{5.10}$$

$$\boldsymbol{z}_t^{(n)} = S(\tilde{\boldsymbol{z}}_{t|t}^{(n)}, \tilde{\boldsymbol{m}}_{t|t}^{(3)}, \tilde{\boldsymbol{m}}_{t|t}^{(4)})^{-1}. \tag{5.11}$$

5.2.3.3 Smoothing Step

The smoothing step used for calculating $\boldsymbol{x}_{t|s}$ $(s > t)$ was essentially the same as the filtering step. The smoothing step also consists of the following three sub-steps. At tth ($t \in \mathcal{T}_{obs}$) time step, for $s = t+1, \ldots, T$,

1. Particle Filter Step

   (a) Resample $\hat{\boldsymbol{x}}_{t|s}^{(n)}$ according to

   $$p(\boldsymbol{x}_t|\boldsymbol{Y}_s) = \frac{1}{\sum_{\dot{n}} p(\boldsymbol{y}_s|\boldsymbol{x}_{s|s-1}^{(\dot{n})})} \sum_{n=1}^{N} p(\boldsymbol{y}_s|\boldsymbol{x}_{s|s-1}^{(n)}) \delta(\boldsymbol{x}_t - \boldsymbol{x}_{t|s-1}^{(n)}), \tag{5.12}$$

   where $\delta(\cdot)$ is a Dirac delta function.

   (b) Calculate first and second moments $\mu_{t|s} = E[\{\hat{\boldsymbol{x}}_{t|s}^{(n)}\}]$ and $V_{t|s} = Var[\{\hat{\boldsymbol{x}}_{t|s}^{(n)}\}]$, respectively.

   (c) Standardize $\hat{\boldsymbol{x}}_{t|s}^{(n)}$ as

   $$\hat{\boldsymbol{z}}_{t|s}^{(n)} = V_{t|s}^{-\frac{1}{2}} \cdot (\hat{\boldsymbol{x}}_{t|s}^{(n)} - \boldsymbol{\mu}_{t|s}). \tag{5.13}$$

   (d) Calculate third and fourth central moments $\hat{\boldsymbol{m}}_{t|s}^{(3)} = E[\{\hat{\boldsymbol{z}}_{t|s}^{(n)}\}^3]$ and $\hat{\boldsymbol{m}}_{t|s}^{(4)} = E[\{\hat{\boldsymbol{z}}_{t|s}^{(n)}\}^4]$, respectively.

2. Ensemble Kalman Filter Step

   (a) Calculate Kalman gain

   $$K_s = \frac{1}{N-1} \{\sum_{n=1}^{N} (\boldsymbol{x}_{t|s-1}^{(n)} - E[\{\boldsymbol{x}_{t|s-1}^{(n)}\}])(\boldsymbol{x}_{s|s-1}^{(n)} - E[\{\boldsymbol{x}_{s|s-1}^{(n)}\}])'\}(V_{s|s-1} + R_s)^{-1}. \tag{5.14}$$

   (b) Calculate $\tilde{\boldsymbol{x}}_{t|s}^{(n)}$ as

   $$\tilde{\boldsymbol{x}}_{t|s}^{(n)} = \boldsymbol{x}_{t|s-1}^{(n)} + K_s(\boldsymbol{y}_s - \boldsymbol{x}_{s|s-1}^{(n)} + \boldsymbol{w}_s^{(n)}). \tag{5.15}$$

(c) Calculate first and second moments $\tilde{\mu}_{t|s} = E[\{\tilde{\boldsymbol{x}}_{t|s}^{(n)}\}]$ and $\tilde{V}_{t|s} = Var[\{\tilde{\boldsymbol{x}}_{t|s}^{(n)}\}]$, respectively.

(d) Standardize $\tilde{\boldsymbol{x}}_{t|s}^{(n)}$ as

$$\tilde{\boldsymbol{z}}_{t|s}^{(n)} = \tilde{V}_{t|s}^{-\frac{1}{2}} \cdot (\tilde{\boldsymbol{x}}_{t|s}^{(n)} - \tilde{\boldsymbol{\mu}}_{t|s}). \tag{5.16}$$

(e) Calculate third and fourth central moments $\tilde{\boldsymbol{m}}_{t|s}^{(3)} = E[\{\tilde{\boldsymbol{z}}_{t|s}^{(n)}\}^3]$ and $\tilde{\boldsymbol{m}}_{t|s}^{(4)} = E[\{\tilde{\boldsymbol{z}}_{t|s}^{(n)}\}^4]$, respectively.

3. Merging Step
Calculate $\boldsymbol{x}_{t|s}^{(n)}$ as

$$\boldsymbol{x}_{t|s}^{(n)} = \hat{V}_{t|s}^{\frac{1}{2}} S(\boldsymbol{z}_t^{(n)}, \hat{\boldsymbol{m}}_{t|s}^{(3)}, \hat{\boldsymbol{m}}_{t|s}^{(4)}) + \hat{\boldsymbol{\mu}}_{t|s}, \tag{5.17}$$

$$\boldsymbol{z}_t^{(n)} = S(\tilde{\boldsymbol{z}}_{t|s}^{(n)}, \tilde{\boldsymbol{m}}_{t|s}^{(3)}, \tilde{\boldsymbol{m}}_{t|s}^{(4)})^{-1}. \tag{5.18}$$

5.2.4 Parameter Estimation Using EM-algorithm

Let $X_T = \{\boldsymbol{x}_0, \ldots, \boldsymbol{x}_T\}$ be the set of state variables and $\boldsymbol{\theta} = \{A, B, G, \boldsymbol{u}, Q, R, \boldsymbol{\mu}_0\}$ be the parameter vector. The log-likelihood of observation data is given by

$$\log L = \log \int P(\boldsymbol{x}_0) \prod_{t \in \mathcal{T}} P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) \prod_{t \in \mathcal{T}_{obs}} P(\boldsymbol{y}_t|\boldsymbol{x}_t) d\boldsymbol{x}_1 \ldots d\boldsymbol{x}_T, \tag{5.19}$$

where $P(\boldsymbol{x}_0)$ is a probability density of N-dimensional Gaussian distributions $N(\boldsymbol{\mu}_0, \Sigma_0)$, $P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$ and $P(\boldsymbol{y}_t|\boldsymbol{x}_t)$ can be probability densities of N-dimensional non-Gaussian distributions obtained by ensemble approximation.

In this study, we estimate the parameter values $\boldsymbol{\theta}$ by maximizing Eq. (5.19) using the EM-algorithm. Thus, the conditional expectation of the joint log-likelihood of the complete data (X_T, Y_T) at lth iteration

$$q(\boldsymbol{\theta}|\boldsymbol{\theta}_l) = E[\log P(Y_T, X_T|\boldsymbol{\theta})|Y_T, \boldsymbol{\theta}_l], \tag{5.20}$$

is iteratively maximized with respect to $\boldsymbol{\theta}$ until convergence.

In the Expectation-step, $q(\boldsymbol{\theta}|\boldsymbol{\theta}_l)$ is calculated by

$$\begin{aligned} q(\boldsymbol{\theta}|\boldsymbol{\theta}_l) = &-\frac{1}{2}\mathrm{tr}\{V_{0|0}^{-1}(V_{0|t} + (\boldsymbol{x}_{0|t} - \boldsymbol{\mu}_0)(\boldsymbol{x}_{0|t} - \boldsymbol{\mu}_0)')\} - \frac{1}{2}\log|\Sigma_{0|t}| \\ &- \frac{1}{2}\sum_{t=1}^{T}\mathrm{tr}\{Q^{-1}E[(\boldsymbol{x}_t - F\boldsymbol{x}_{t-1} - B\mathrm{vec}(\boldsymbol{x}_{t-1}\boldsymbol{x}_{t-1}') - G\boldsymbol{d}_{t-1} - \boldsymbol{u}) \\ &\cdot (\boldsymbol{x}_t - F\boldsymbol{x}_{t-1} - B\mathrm{vec}(\boldsymbol{x}_{t-1}\boldsymbol{x}_{t-1}') - G\boldsymbol{d}_{t-1} - \boldsymbol{u})'|Y_T]\} \\ &- \frac{T}{2}\log|Q| - \frac{1}{2}\mathrm{tr}\{R^{-1}\sum_{t=1}^{T}\{(\boldsymbol{y}_t - \boldsymbol{x}_{t|t})(\boldsymbol{y}_t - \boldsymbol{x}_{t|t})' + V_{t|t}')\} - \frac{T}{2}\log|R| - N(T + \frac{1}{2})\log 2\pi. \end{aligned} \tag{5.21}$$

In the Maximization-step, $\boldsymbol{\theta}_l$ is updated to $\boldsymbol{\theta}_{l+1} = \arg\max_{\boldsymbol{\theta}} q(\boldsymbol{\theta}|\boldsymbol{\theta}_l)$. At first, set conditional expectations of $\boldsymbol{x}_t$ as

$$V_t = \frac{1}{N}\sum_{t\in\mathcal{T}}\sum_{n=1}^{N}\boldsymbol{x}_{t|T}^{(n)}\boldsymbol{x}_{t|T}^{(n)'}, \tag{5.22}$$

$$V_{lag} = \frac{1}{N}\sum_{t\in\mathcal{T}}\sum_{n=1}^{N}\boldsymbol{x}_{t|T}^{(n)}\boldsymbol{x}_{t-1|T}^{(n)'}, \tag{5.23}$$

$$V_{t-1} = \frac{1}{N}\sum_{t\in\mathcal{T}}\sum_{n=1}^{N}\boldsymbol{x}_{t-1|T}^{(n)}\boldsymbol{x}_{t-1|T}^{(n)'}, \tag{5.24}$$

$$\Phi_{lag} = \frac{1}{N}\sum_{t\in\mathcal{T}}\sum_{n=1}^{N}\boldsymbol{x}_{t|T}^{(n)}\mathrm{vec}(\boldsymbol{x}_{t-1|T}^{(n)}\boldsymbol{x}_{t-1|T}^{(n)'})', \tag{5.25}$$

$$\Phi_{t-1} = \frac{1}{N}\sum_{t\in\mathcal{T}}\sum_{n=1}^{N}\boldsymbol{x}_{t-1|T}^{(n)}\mathrm{vec}(\boldsymbol{x}_{t-1|T}^{(n)}\boldsymbol{x}_{t-1|T}^{(n)'})', \tag{5.26}$$

$$\Psi_{t-1} = \frac{1}{N}\sum_{t\in\mathcal{T}}\sum_{n=1}^{N}\mathrm{vec}(\boldsymbol{x}_{t-1|T}^{(n)}\boldsymbol{x}_{t-1|T}^{(n)'})\mathrm{vec}(\boldsymbol{x}_{t-1|T}^{(n)}\boldsymbol{x}_{t-1|T}^{(n)'})', \tag{5.27}$$

$$E_{lag} = \frac{1}{N}\sum_{t\in\mathcal{T}}\sum_{n=1}^{N}\boldsymbol{x}_{t|T}^{(n)}\boldsymbol{d}_{t-1}', \tag{5.28}$$

$$E_{t-1} = \frac{1}{N}\sum_{t\in\mathcal{T}}\sum_{n=1}^{N}\boldsymbol{x}_{t-1|T}^{(n)}\boldsymbol{d}_{t-1}', \tag{5.29}$$

$$E_{t-1}^2 = \frac{1}{N}\sum_{t\in\mathcal{T}}\sum_{n=1}^{N}\mathrm{vec}(\boldsymbol{x}_{t-1|T}^{(n)}\boldsymbol{x}_{t-1|T}^{(n)'})\boldsymbol{d}_{t-1}', \tag{5.30}$$

$$\boldsymbol{z} = \sum_{t\in\mathcal{T}}^{T} \boldsymbol{d}_{t-1}, \tag{5.31}$$

$$Z = \sum_{t\in\mathcal{T}}^{T} \boldsymbol{d}_{t-1}\boldsymbol{d}'_{t-1}. \tag{5.32}$$

$$\boldsymbol{s}_t = \frac{1}{N}\sum_{t\in\mathcal{T}}\sum_{n=1}^{N} \boldsymbol{x}_{t|T}^{(n)}, \tag{5.33}$$

$$\boldsymbol{s}_{t-1} = \frac{1}{N}\sum_{t\in\mathcal{T}}\sum_{n=1}^{N} \boldsymbol{x}_{t-1|T}^{(n)}, \tag{5.34}$$

$$\boldsymbol{s}_{t-1}^2 = \frac{1}{N}\sum_{t\in\mathcal{T}}\sum_{n=1}^{N} \mathrm{vec}(\boldsymbol{x}_{t-1|T}^{(n)}\boldsymbol{x}_{t-1|T}^{(n)\prime}). \tag{5.35}$$

Let $\boldsymbol{v}_{lag,i}$, $\boldsymbol{\phi}_{lag,i}$ and $\boldsymbol{\phi}_{t-1,i}$ be a transpose of the ith row vector of V_{lag}, Φ_{lag} and Φ_{t-1}, respectively. Then, $\boldsymbol{\theta}$ is updated as

$$\boldsymbol{a}_i^{\mathcal{A}_i} = {V_{t-1}^{\mathcal{A}_i}}^{-1}(\boldsymbol{v}_{lag,i}^{\mathcal{A}_i} - \phi_{t-1}^{\mathcal{A}_i\times\mathcal{B}_i}\boldsymbol{b}_i^{\mathcal{B}_i} - E_{t-1}^{\mathcal{A}_i\times\mathcal{G}_i}\boldsymbol{g}_n^{\mathcal{G}_i} - u_i\boldsymbol{s}_{t-1}^{\mathcal{A}_i}), \tag{5.36}$$

$$\boldsymbol{b}_i^{\mathcal{B}_i} = \Psi_{t-1}^{\mathcal{B}_i^{-1}}(\boldsymbol{\phi}_{lag,i}^{\mathcal{B}_i} - \phi_{t-1}^{\mathcal{A}_i\times\mathcal{B}_i{}'}\boldsymbol{a}_i^{\mathcal{A}_i} - E_{t-1}^{2^{\mathcal{B}_i}\times\mathcal{G}_i}\boldsymbol{g}_i^{\mathcal{G}_i} - u_i\boldsymbol{s}^{2^{\mathcal{B}_i}}), \tag{5.37}$$

$$\boldsymbol{g}_i^{\mathcal{G}_i} = {Z^{\mathcal{G}_i}}^{-1}(\boldsymbol{e}_{lag,n}^{\mathcal{G}_i} - E_{t-1}^{\mathcal{A}_i\times\mathcal{G}_i'}\boldsymbol{a}_n^{\mathcal{A}_i} - E_{t-1}^{2^{\mathcal{B}_i}\times\mathcal{G}_i}{}'\boldsymbol{b}_i^{\mathcal{B}_i} - u_n\boldsymbol{z}^{\mathcal{G}_i}) \tag{5.38}$$

$$\boldsymbol{u} = \frac{\boldsymbol{s}_t - A\boldsymbol{s}_{t-1} - B\boldsymbol{s}^2 - G\boldsymbol{z}}{T}, \tag{5.39}$$

$$\begin{aligned} Q = \frac{1}{T}\sum_{t=1}^{T} E[&(\boldsymbol{x}_t - A\boldsymbol{x}_{t-1} - B\mathrm{vec}(\boldsymbol{x}_{t-1}\boldsymbol{x}'_{t-1}) - G\boldsymbol{d}_{t-1} - \boldsymbol{u}) \\ &\cdot(\boldsymbol{x}_t - A\boldsymbol{x}_{t-1} - B\mathrm{vec}(\boldsymbol{x}_{t-1}\boldsymbol{x}'_{t-1}) - G\boldsymbol{d}_{t-1} - \boldsymbol{u})'|Y_T], \end{aligned} \tag{5.40}$$

$$\boldsymbol{\mu}_0 = \boldsymbol{x}_{0|T}, \tag{5.41}$$

$$R = \frac{1}{T}\sum_{t\in\mathcal{T}_{obs}}\{(\boldsymbol{y}_t - \boldsymbol{x}_{t|t})(\boldsymbol{y}_t - \boldsymbol{x}_{t|t})' + V_{t|t}\}, \tag{5.42}$$

where $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{G}$ are active sets of elements for A, B and G, respectively. For example, $\boldsymbol{a}_i^{\mathcal{A}}$ is an $|\mathcal{A}|$-dimensional vector consisting of elements regulating the ith gene.

5.2.5 Network Restoration Algorithm

We consider an algorithm to explore the best model by sequentially evaluating candidate models generated from the current best model $\mathcal{M}_{current}$ by partially modifying the regulation. Briefly, given the original model $\mathcal{M}_{original}$, we attempt to sequentially create and evaluate candidates that are generated by adding, deleting and replacing regulatory components of $\mathcal{M}_{current}$ until the best model is no longer updated. The conceptual view is illustrated in Fig. 5.2.

Due to the heavy computational cost to evaluate the model by HMEnPF, we proposed a novel algorithm for reconstructing GRNs with combining UKF and HMEnPF as described in

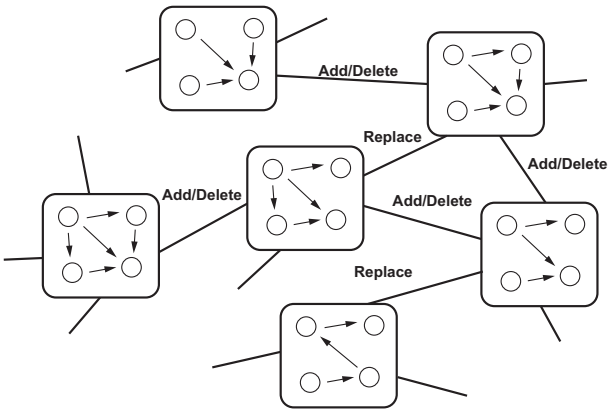

Figure 5.2: A conceptual view of the proposed algorithm. The proposed algorithm performs three ways to explore model space, thus, adding, deleting and replacing a regulation from the current best model. Starting from the original model, the proposed algorithm tries to find the best model with respect to the BIC score.

A

Algorithms 5-9 and illustrated in Fig 5.3. In these algorithms, we first evaluate the candidates comprehensively by UKF and then evaluate the $r_{1,2}$ top candidates with respect to e_a, e_b and e_g, which are the effectiveness of individual, combinatorial and drug effects on to the ith gene under the model $\mathcal{M}$, described as

$$e_a(\mathcal{M}, i) = \boldsymbol{a}_i' V_{t-1} \boldsymbol{a}_i - 2\boldsymbol{v}_{lag,i} \boldsymbol{a}_i + 2\boldsymbol{b}_i' \phi_{t-1}' \boldsymbol{a}_i + 2\boldsymbol{g}_i' E_{t-1}' \boldsymbol{a}_i + 2u_i \boldsymbol{s}_{t-1}' \boldsymbol{a}_i, \tag{5.43}$$

$$e_b(\mathcal{M}, i) = \boldsymbol{b}_i' \Psi_{t-1} \boldsymbol{b}_i - 2\boldsymbol{\phi}_{lag,i}' \boldsymbol{b}_i + 2\boldsymbol{a}_i' \phi_{t-1} \boldsymbol{b}_i + 2\boldsymbol{g}_i' E_{t-1}^{2'} \boldsymbol{b}_i + 2u_i \boldsymbol{s}_{t-1}^{2'} \boldsymbol{b}_i, \tag{5.44}$$

$$e_g(\mathcal{M}, i) = \boldsymbol{g}_i' Z \boldsymbol{g}_i - 2\boldsymbol{e}_{lag,i}' \boldsymbol{g}_i + 2\boldsymbol{a}_i' E_{t-1} \boldsymbol{g}_i + 2\boldsymbol{b}_i' E_{t-1}^2 \boldsymbol{g}_i + 2u_i \boldsymbol{z}' \boldsymbol{g}_i, \tag{5.45}$$

respectively. It should be noted that the functions are derived when calculating $\arg\max_{\boldsymbol{a}_i} q(\boldsymbol{\theta}|\boldsymbol{\theta}_l)$, $\arg\max_{\boldsymbol{b}_i} q(\boldsymbol{\theta}|\boldsymbol{\theta}_l)$ and $\arg\max_{\boldsymbol{g}_i} q(\boldsymbol{\theta}|\boldsymbol{\theta}_l)$, respectively.

Note that, when the systems include G, regulations by the drugs are inferred in the same way as A in Algorithms 5-9. In Results section, we set $\{r_1, r_2, add_{max}, del_{max}\} = \{5, 5, +\infty, +\infty\}$.

Algorithm 5 The proposed algorithm for improving GRNs utilizing UKF and HMEnPF.

1: Set add_{max}, del_{max}, r_1 and r_2;
2: Define that add and del are the number of added and deleted regulations from $\mathcal{M}_{original}$, respectively;
3: $flag \leftarrow 0$; $c \leftarrow 0$; $\mathcal{M}_{current} \leftarrow \mathcal{M}_{original}$;
4: $BIC_{current} \leftarrow$ the BIC score of the original model;
5: Execute the first phase of the proposed algorithm (Algorithm 2)
6: Execute the second phase of the proposed algorithm (Algorithm 3)
7: Output $\mathcal{M}_{current}$

Algorithm 6 The first phase of the proposed algorithm.

$flag \leftarrow 0$;
$c \leftarrow 0$;
while $flag < 2$ **do**
  for $i = 1$ to p **do**
    for $k = 1$ to r_1 **do**
      if c (mod2) $= 0$ and $A_{i,j}$ of $\mathcal{M}_{current} = 0$ and $add < add_{max}$ **then**
        $j \leftarrow$ the kth minimum element with respect to $e(i, j_{can})$ ($j_{can} \notin \mathcal{A}_i$) of $\mathcal{M}_{current}$;
        Consider $\mathcal{M}$ that is constructed from $\mathcal{M}_{current}$ by setting a regulation to the ith gene by the jth gene as included in the active set;
      else if c (mod2) $= 1$ and $A_{i,j}$ of $\mathcal{M}_{current} = 1$ and $del < del_{max}$ **then**
        $j \leftarrow$ the kth minimum element with respect to $e(i, j_{can})$ ($j_{can} \in \mathcal{A}_i$) of $\mathcal{M}_{current}$;
        Consider $\mathcal{M}$ that is constructed from $\mathcal{M}_{current}$ by setting a regulation to the ith gene by the jth gene as not included in the active set;
      end if
      Estimate the parameter values and obtain the BIC score of $\mathcal{M}$ by UKF;
    end for
  end for
  Estimate the parameter values and obtain the BIC score of the top r_2 candidates by HMEnPF;
  if $BIC_{current} >$ the minimum BIC score among models calculated above **then**
    Set $\mathcal{M}_{current}$ and $BIC_{current}$ as those of the minimum one;
    $flag \leftarrow 0$;
  else
    $flag \leftarrow flag + 1$;
  end if
  $c \leftarrow c + 1$;
end while

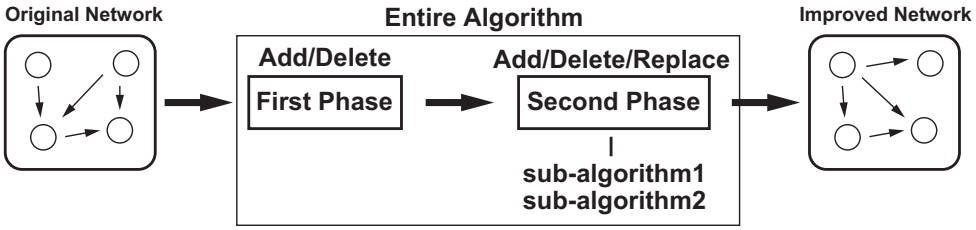

Figure 5.3: A procedure of exploring the best model using the proposed algorithm. The proposed algorithm (Algorithm 5) consists of two phases (Algorithms 6 and 7) and the second phase consists of two sub-algorithms (Algorithms 8 and 9). Starting from the original model, the proposed algorithm tries to explore the best model.

Algorithm 7 The second phase of the proposed algorithm.

```
1: flag ← 0;
2: while flag < p^2 do
3:   for i = 1 to p do
4:     for j = 1 to p do
5:       if A_{i,j} of M_current = 1 then
6:         changed ← Execute sub-algorithm 1(i, j);
7:       end if
8:       if changed then
9:         flag ← 0;
10:        Execute sub-algorithm 2;
11:      else
12:        flag ← flag + 1;
13:      end if
14:      changed ← FALSE;
15:      if flag >= p^2 then
16:        break;
17:      end if
18:    end for
19:    if flag >= p^2 then
20:      break;
21:    end if
22:  end for
23: end while
```

Algorithm 8 Sub-algorithm 1(i_{orig}, j_{orig}).

$changed \leftarrow FALSE$;
Set $\mathcal{M}_{candidate}$ as $\mathcal{M}_{current}$ with deleting a regulation to the i_{orig}th gene by the j_{orig}th gene;
Estimate the parameter values and obtain the BIC score $BIC_{candidate}$ by HMEnPF;
if $BIC_{current} > BIC_{candidate}$ and $del_{max} > del$ **then**
  Set $\mathcal{M}_{candidate}$ as $\mathcal{M}_{current}$; $BIC_{candidate} \leftarrow BIC_{current}$;
  $changed \leftarrow TRUE$;
else
  for $i = 1$ to p **do**
    for $k = 1$ to r_1 **do**
      $j \leftarrow$ the kth minimum element with respect to $e(i, j_{can})$ ($j_{can} \notin \mathcal{A}_i$) of $\mathcal{M}_{candidate}$;
      Consider $\mathcal{M}_{candidate}$ that is constructed from $\mathcal{M}_{candidate}$ by setting a regulation to the ith gene by the jth gene as included in the active set;
      if $add_{max} < add$ or $del_{max} < del$ of $\mathcal{M}_{candidate}$ **then**
        continue;
      end if
      Estimate the parameter values and obtain the BIC score by UKF;
    end for
  end for
  Estimate the parameter values and obtain the BIC score of the top r_2 candidates by HMEnPF;
  if $BIC_{current} >$ the minimum BIC score among candidate models calculated above **then**
    Set $\mathcal{M}_{current}$ and $BIC_{current}$ as those of the minimum one;
    $changed \leftarrow TRUE$;
  end if
end if
return *changed*;

Algorithm 9 Sub-algorithm 2.

while $TRUE$ **do**
  for $i = 1$ to p **do**
    for $k = 1$ to r **do**
      $j \leftarrow$ the kth minimum element with respect to $e(i, j_{can})$ ($j_{can} \notin \mathcal{A}_i$) of $\mathcal{M}_{current}$;
      if $A_{i,j}$ of $\mathcal{M}_{current} = 0$ and $add < add_{max}$ **then**
        Consider $\mathcal{M}$ that is constructed from $\mathcal{M}_{current}$ by setting a regulation to the ith gene by the jth gene as included in the active set;
        Estimate the parameter values and obtain the BIC score of $\mathcal{M}$ by UKF;
      end if
    end for
  end for
  Estimate the parameter values and obtain the BIC score of the top r_2 candidates by HMEnPF;
  if $BIC_{current} >$ the minimum BIC score among models calculated above **then**
    Set $\mathcal{M}_{current}$ and $BIC_{current}$ as those of the minimum one;
  else
    break;
  end if
end while

5.3 Results

5.3.1 Comparison Using Synthetic Data

To show the effectiveness of the proposed method, we prepared artificial time-course gene expression data based on the synthetic networks, WNT5A [55] and the yeast cell cycle [53], as illustrated in Figs. 4.4 and 4.5, respectively. For each network, we generated a time-course ($\mathcal{T} = \{1, 2, \ldots, 30\}$) by using Eqs. (5.1) and (5.2). The values of the parameters A and B in Eq. (5.1) were determined between 0 and 1, and the system noise $\boldsymbol{v}_t$ was generated according to a Gaussian distribution with a mean 0 and a variance 0.1. For Eq. (5.2), we generated Gaussian observational noise with a mean of 0 and a variance of 0.3 and added to synthetic simulation data. For the original networks to be improved by the proposed algorithm, we utilized GeneNet [80,94] based on an empirical graphical Gaussian model (GGM) and G1DBN [63] based on dynamic Bayesian networks using first order conditional dependencies. Then, the original and improved networks were evaluated by true positive (TP), false positive (FP), true negative (TN), false negative (FN), precision rate ($PR = \frac{\mathrm{TP}}{\mathrm{TP+FP}}$), recall rate ($RR = \frac{\mathrm{TP}}{\mathrm{TP+FN}}$) and F-measure ($=\frac{2\mathrm{PR}\cdot\mathrm{RR}}{\mathrm{PR+RR}}$). Note that, since GeneNet infers undirected regulations between genes, we evaluated the results using undirected true networks. In addition, to apply the network inferred by GeneNet as an original network for the proposed algorithm, we generated networks in which regulations was prepared as follows; (i) a true directed regulation was set when an inferred undirected regulation was correct and (ii) a false directed regulation of which direction was randomly selected was set when an inferred undirected regulation was incorrect. Here, to clarify the significance of HMEnPF, we also showed the results of the proposed algorithm using

UKF only. The results are summarized in Tables 5.1-5.4.

Table 5.1: Comparison of the proposed method, that of using UKF, and G1DBN from WNTA5A network, where networks inferred by G1DBN were used as the original networks for the former two methods.

	PR	RR	F-measure	TP	FP	TN	FN
G1DBN	0.730	0.633	0.679	19	7	53	11
UKF	0.778	0.700	0.737	21	6	54	9
HMEnPF	0.786	0.733	0.759	22	6	54	9

Table 5.2: Comparison of the proposed method, that of using UKF, and GeneNet from WNTA5A network, where networks inferred by GeneNet were used as the original networks for the former two methods.

	PR	RR	F-measure	TP	FP	TN	FN
GeneNet	0.500	0.433	0.464	13	13	47	17
UKF	0.656	0.700	0.677	21	11	49	9
HMEnPF	0.710	0.733	0.721	22	9	51	8

Table 5.3: Comparison of the proposed method, that of using UKF, and G1DBN from a yeast cell-cycle network, where networks inferred by G1DBN were used as the original networks for the former two methods.

	PR	RR	F-measure	TP	FP	TN	FN
G1DBN	0.556	0.577	0.567	15	12	52	11
UKF	0.750	0.692	0.720	18	6	58	8
HMEnPF	0.730	0.730	0.730	19	7	57	7

The results indicate that the proposed methods using HMEnPF and only UKF could outperform G1DBN and GeneNet, and the proposed algorithm showed better performance than that of using UKF only. This concludes that retaining higher moment information can improve the accuracy of approximation and estimate correct parameter values. Additionally, we recognized that the performance of the proposed algorithm strongly depends on the accuracy of the original network. Thus, to obtain better results, we should carefully construct original networks or select inference methods for creating the original network.

5.3.2 Inference Using Real Data

As an application example, we analyzed microarray time-course gene expression data from rat skeletal muscle [5,115]. The microarray data were downloaded from the GEO database (GSE490). The time-course gene expression data was measured at 0, 0.25, 0.5, 0.75, 1, 2, 4, 5, 5.5, 7, 8, 12, 18, 30, 48, and 72 [h] (16 time points) after stimulation of corticosteroid, but we removed data at 48 and 72[h] (steady state profiles) for computational efficiency. The data at time 0 represent controls (untreated). There were two, three, or four replicated observations for each time point.

Since corticosteroid pharmacokinetics/dynamics in skeletal muscle have been established based on differential equations [115] as shown in Fig. 3.11, the time-dependent concentration of

Table 5.4: Comparison of the proposed method, that of using UKF, and GeneNet from a yeast cell-cycle network, where networks inferred by GeneNet were used as the original networks for the former two methods.

	PR	RR	F-measure	TP	FP	TN	FN
GeneNet	0.375	0.230	0.286	6	10	54	20
UKF	0.680	0.654	0.667	17	8	56	9
HMEnPF	0.730	0.730	0.730	19	7	57	7

corticosteroid in nucleus in rat skeletal muscle $\boldsymbol{d}_t$ can be obtained. The details are explained in Chapter 3.3.2.

Furthermore, corticosteroid catabolic/anabolic processes in rat skeletal muscle have been partially established [100]; thus, we handled gene (i) TFs, *Trim63*, *Akt1*, *Akt2*, *Mstn*, *Mtor*, *Irs1*, and (ii) non-TFs, *Akt3*, *Anxa3*, *Bcat2*, *Bnip3*, *Foxo1*, *Igf1*, *Igf1r*, *Pik3c3*, *Pik3cb*, *Pik3cd*, *Pik3c2g*, *Rheb*, *Slc2a4* with their regulatory relationships. Additionally, we handled genes (iii) TFs, *Cebpb*, *Cebpd*, *Gpam*, *Srebf1* and (iv) non-TFs, *Rxrg*, *Scarb1*, *Scd*, *Gpd2*, *Mapk6*, *Ace*, *Ptpn1*, *Ptprf*, *Edn1*, *Agtr1a*, *Ppard*, *Hmgcs2*, *Serpine1*, *Il6r*, *Mapk14*, *Ucp3* and *Pdk4* that are suggested as corticosteroid related genes [5]. Note that the microarray (GSE490) does not include three genes in the original pathway [100], *Redd1*, *Bcaa* and *Klf15*. In summary, we handled the time-dependent concentration of corticosteroid in nucleus, these 40 genes (shown in Table 5.5) and an original network that was inferred by G1DBN with regulatory relationships among (i) and (ii). Note that TF information was derived from ITFP [122].

Table 5.5: Sets of pharmacogenomic genes handled in the real data experiment.

	Gene Set	Literature [115]/ [5]	TF candidate
(i)	*Trim63*, *Akt1*, *Akt2*, *Mstn*, *Irs1*	∘/-	∘
(ii)	*Akt3*, *Anxa3*, *Bcat2*, *Bnip3*, *Foxo1*, *Igf1*, *Igf1r*, *Mtor* *Pik3c3*, *Pik3cb*, *Pik3cd*, *Pik3c2g*, *Rheb*, *Slc2a4*	∘/-	-
(iii)	*Cebpb*, *Cebpd*, *Gpam*, *Srebf1*	-/∘	∘
(iv)	*Rxrg*, *Scarb1*, *Scd*, *Gpd2*, *Mapk6*, *Ace*, *Ptpn1* *Ptprf*, *Edn1*, *Agtr1a*, *Ppard*, *Hmgcs2*, *Serpine1* *Il6r*, *Mapk14*, *Ucp3*, *Pdk4*	-/∘	-

Consequently, we obtained the improved network as illustrated in Fig. 5.4. A purple circle, blues circles, and green circles represent corticosterid, TF candidates and non-TF candidates, respectively. In the center of this figure, there exist corticosteroid regulations to several TF and nonTF genes and regulatory effects transmit to down stream genes of TF candidates genes. Since some combinatorial regulations were inferred, it is conceivable that higher moment approximation can affect the estimation results beyond linear models.

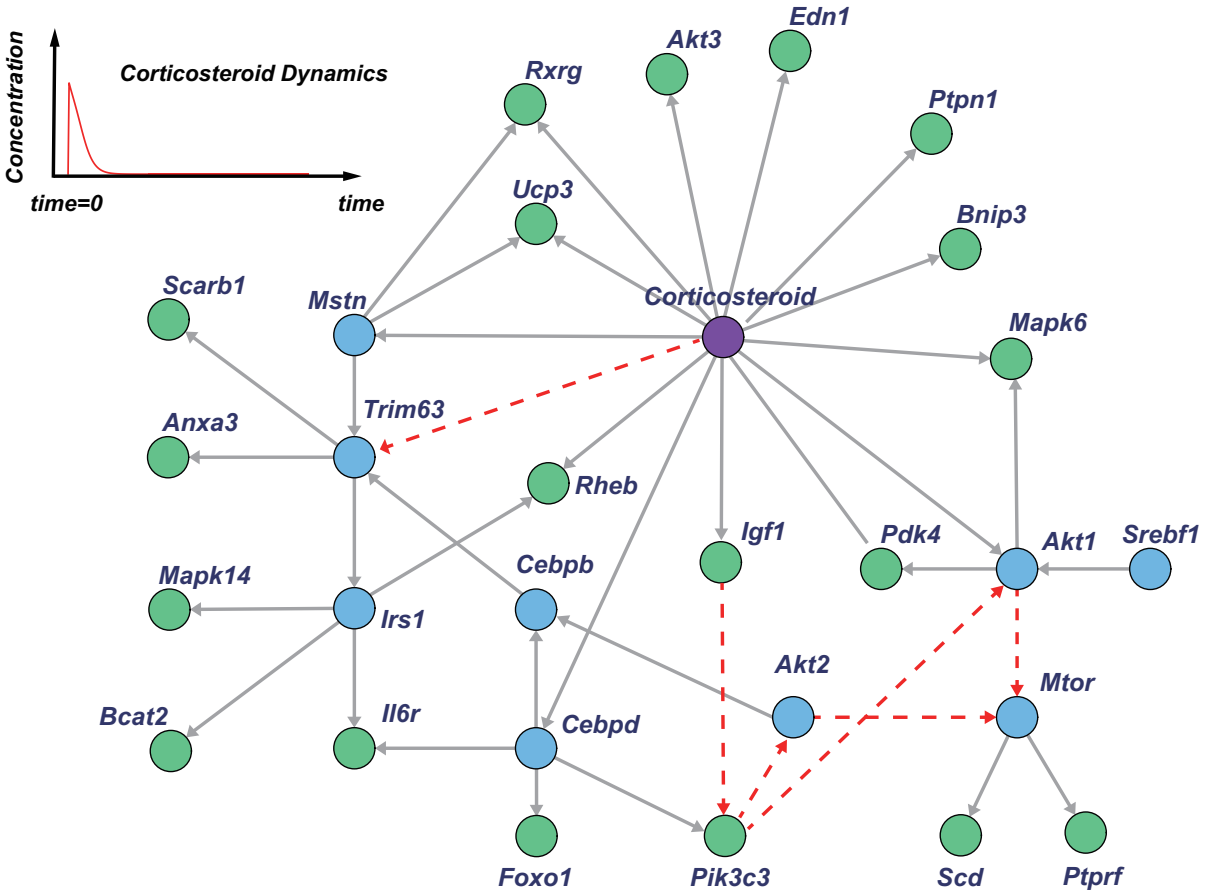

Figure 5.4: An inferred network of corticosteroid pharmacogenomics in rat skeletal muscle by the proposed algorithm. Since a part of the pharmacogenomic system has been investigated previously, we inferred the relationships incorporating known pathways (red dotted arrows) and related genes [100,115], where a purple circle, blues circles and green circles represent corticosterid, TF candidates and non-TF candidates, respectively.

5.4 Discussion

In this chapter, we developed a novel approach to restore original GRNs to be consistent with time-course mRNA expression data based on the combinatorial transcription model. Since we applied a state space representation with the nonlinear system equation in the context of data assimilation, the posterior distributions of the hidden variables can be non-Gaussian distributions. In contrast to the previous approaches using particle filter algorithm, Gaussian approximation and regression-based solutions, our proposed approach, HMEnPF, can retain the first, second, third and fourth central moments through filtering steps to estimate near optimal parameter values by the EM-algorithm.

According to the comparison results using two synthetic data based on the real biological pathways, the proposed method successfully explored better models than previous methods, G1DBN and GeneNet, considering linear relevance. Moreover, the proposed algorithm using HMEnPF outperformed that of using UKF. This concludes that HMEnPF retaining parts of higher moment information can improve the accuracy of the estimation of the parameter values beyond unscented approximation (that cannot retain any moment through filtering steps based on Gaussian approximation). Through the experimental results, we also observed that the performance of the restoration algorithm strongly depends on the original network, which was prepared by literature information or some GRNs inference methods. Thus, one of significant points is to select methods to infer the original network.

As an application example, we prepared corticosteroid pharmacogenomic pathways in rat muscle that have been investigated and established a part of regulatory relationships and related genes. Additionally, the intracellular concentration of corticosteroid that directly/indirectly affects gene expression can be obtained by the previously developed differential equations and TF information for rat genes can also be utilized. In summary, we added the time-depending concentration of corticosteroid to the model and inferred the regulatory relationships among 40 genes and corticosteroid with fixing the established pathways and restricting that only TF candidates can regulate other genes. G1DBN was employed to construct the original model for the proposed method. Consequently, several combinatorial regulations and regulations by corticosteroid were inferred by extending the original network. Since previous linear models may not be able to infer these regulations, the proposed method can be valuable to restore inferred and literature-based networks to be consistent with the data.

Chapter 6

Comprehensive Pharmacogenomic Pathway Screening by Data Assimilation

6.1 Background

Construction and simulation of biological pathways are crucial steps in understanding complex networks of biological elements in cells [59, 72, 77, 79, 108, 113, 117]. To construct simulatable models, structures of networks and chemical reactions are collected from existing literature and the values of parameters in the model are set based on the results of biological experiments or estimated based on observed data by some computational method [79]. However, it is possible that there are some missing relationships or elements in the literature-based networks. Therefore, we need to develop a computational strategy to improve a prototype model and create better ones that can predict biological phenomena.

To propose novel networks of genes, statistical graphical models including Bayesian networks [54] and vector autoregressive models [33, 98] have been applied to gene expression data. An advantage of these methods is that we can find networks with a large number of genes and analyze them by a viewpoint of systems. However, due to the noise and the limited amount of the data, some parts of the networks estimated by these methods are not biologically reasonable and cannot be validated. In this chapter, we focus on another strategy. Unlike the statistical methods, our method can create a set of extended simulatable models from prototype literature-based models.

There are two key points in our proposed strategy: One is that various structures of candidate simulation models are systematically generated from the prototypes. The other is that, for each created model, the values of parameters are automatically estimated by data assimilation technique [79, 117]; the values of parameters will be determined by maximizing the prediction capability of the model. For each of simulation models, by using data assimilation technique, we can discover that which genes are appropriately predicted their temporal expression patterns by

Table 6.1: Parameter Setting for the core model and for the constructed pharmacogenomic models.

Fixed Parameter	Value	Unit
k_{s_Rm}	2.90	fmol/g/h
k_{d_Rm}	0.1124	fmol/g/h
IC_{50_Rm}	26.2	fmol/mg
k_{on}	0.00329	l/nmol/h
k_T	0.63	h^{-1}
k_{re}	0.0572	h^{-1}
R_f	0.49	
k_{s_R}	1.2	h^{-1}
k_{d_R}	0.0572	h^{-1}
$mRNA_R^0$	25.8	fmol/g
R^0	540.7	fmol/mg

the candidate model. Since we consider pharmacogenomic pathways, these genes are possibly placed on the mode-of-action of target chemical compound. The results obtained by our proposed strategy could be essential to create a larger and more comprehensive simulation model and systems biology driven pharmacology.

To show the effectiveness of the proposed strategy, we analyze time-course microarray data of rat liver cells treated with corticosteroid [47]. In the previous study, differential equation-based simulation models, named fifth generation model [106], were used and predictable expression patterns by this model were discussed for 197 genes selected by clustering analysis [47]. In this chapter, we systematically generated 58 simulatable models from five prototypes and determined which 63 models suitably predict expression pattern of each gene. Finally, we show a comprehensive pharmacogenomics pathway screening that elucidates associations between genes and simulation models.

6.2 Methods

6.2.1 Corticosteroid Pharmacokinetic and Pharmacogenomics Models

In this chapter, we also handle the corticosteroid pharmacogenomic pathways [47] as illustrated in Fig. 6.1 explained in Chapter 3.2.2. However, in rat liver cell, the reaction parameters were set according to Sun et al. [106] and summarized in Table 6.1.

Based on the fundamental model represented in Fig. 6.1, we want to know how DR and DR(N) affect other genes in transcriptional level. As a basic pharmacogenomic model for finding relationship between drug-receptor complex and other genes, we consider extending five pharmacogenomic models [47] shown in Fig. 6.1 (right). The original five pharmacogenomic pathways [47] have the same elements as the core pharmacokinetic pathway, DR and DR(N), and represent relationships between corticosteroid and its downstream genes. However, more variations can be considered as candidates of pharmacogenomic pathway of corticoid. Therefore,

Table 6.2: Parameter settings for constructed pharmacogenomic model.

Estimated Parameter	Model	Unit
k_sm	All	l/nmol/h
k_dm	All	l/nmol/h
S or IC_50	All	l/nmol/h or fmol/mg
k_sBSm	C	l/nmol/h
k_dBSm	C	l/nmol/h
k_sBS	C, DE	l/nmol/h
k_dBS	C	l/nmol/h
S_bs or IC_50bs	C, DE	l/nmol/h or fmol/mg
S_dr	C	l/nmol/h
$mRNA\_BS^0$	C	fmol/mg
BS^0	DE	fmol/mg

from these five models, we automatically constructed 58 models with the following three rules.

(i) If a regulator, DR(N), DR or BS, activates (represses) the synthesis (degradation) of mRNA, a revised model tests to repress (activate) the degradation (synthesis) of mRNA. However, we do not consider combination effects of them.

(ii) If two regulators regulate the same element, we also consider either two regulator model or one regulator model that is defined by removing one of two edges.

(iii) If two regulators regulate the same element, we consider either independent regulation model that employs additive form or cooperative regulation model with the product of the regulators.

We create these rules for generating simulation models that covers all patterns of regulations when we do not change the number of elements such that mRNAs and proteins in each simulation model.

From Model A: One model with three parameters ("k_sm", "k_dm" and "S or IC_50") was generated by applying the rule (i). These models include only mRNA and can simply represent activation of mRNA expression.

From Model B: One model with three parameters ("k_sm", "k_dm" and "S or IC_50") was generated by applying the rule (i). These models include only mRNA and can simply represent repression of mRNA expression.

From Model C: First, 15 models with 11 or 10 parameters ("k_sm", "k_dm", "S or IC_50", "k_sBSm", "k_dBSm", "k_sBS", "k_dBS", "S_bs or IC_50bs", "S_dr", and "initial values of mRNA_BS" and "BS") were generated by applying the rule (i) and (ii). These models include mRNA, BS, and mRNA_BS. Since DR is included only in Model C, we evaluate the necessity

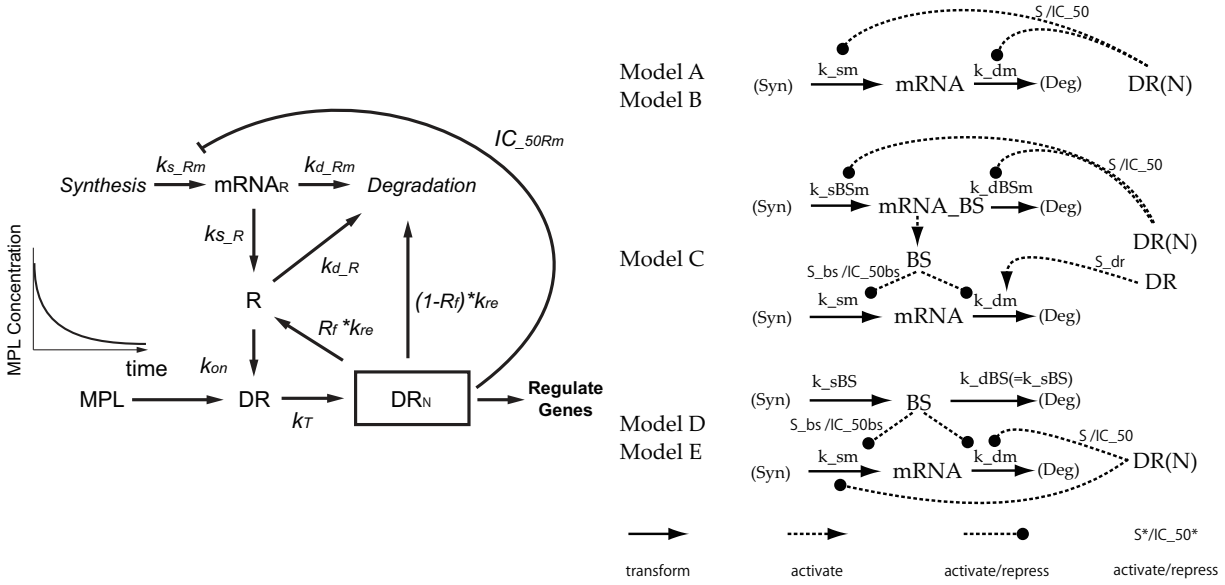

Figure 6.1: Core model for corticosteroid pharmacokinetics and prototype pharmacogenomic models. The left and right figures shows core model for corticosteroid pharmacokinetics and prototype pharmacogenomic models with extensions respectively. In the right figure, the dashed lines with circle are the candidate relations to be extended and BS is the intermediate biosignal.

of the presence of DR by creating models without DR (rule (ii)). Therefore, 16 models that do not have DR were additionally created and finally we have 31 models from Model C.

From Model DE: 24 models with 5 or 6 parameters ("k_sm", "k_dm", "S or IC_50", "k_sBS", "S_bs" or "IC50_bs", and "initial value of BS") were generated by applying the rules (i), (ii) and (iii). These models include mRNA and BS. We unified the notation of Model D and E, because these two models are similar and the extended models are hard to be separated. We constructed 16 models, 4 models and 4 models according to rule (i), (ii) and (iii) respectively. In these simulation models, the parameters, "k_sBSm", "k_sBS", "k_sm", "BS^0 (initial concentration of BS)" and "$mRNA^0_{BS}$ (initial concentration of $mRNA_{BS}$)" were fixed in the original work [47], but we estimate these five parameters together with the other parameters. Note that we focused on the dynamics behavior of the most down stream gene of each model and compared it to observation data of each gene according to the previous work.

Trash Model: Trash Model: To identify genes whose expression data do not significantly changes over time, we created a trash model. Since the trash model is the simplest model that has only two parameters ("k_sm" and "k_dm"), genes with unchanged expression patterns most fit to the trash model in terms of BIC described in the next section.

For these 58 and original 5 pharmacogenomic models, we estimate the values of parameters by using time-course microarray gene expression data from liver cells of rats received glucocorticoid. We also evaluate which models can predict the expression profiles of each gene; it enables us to find better pharmacogenomic models for each gene. For this purpose, a mathematical technique called data assimilation for parameter estimation and model selection is described in the next section.

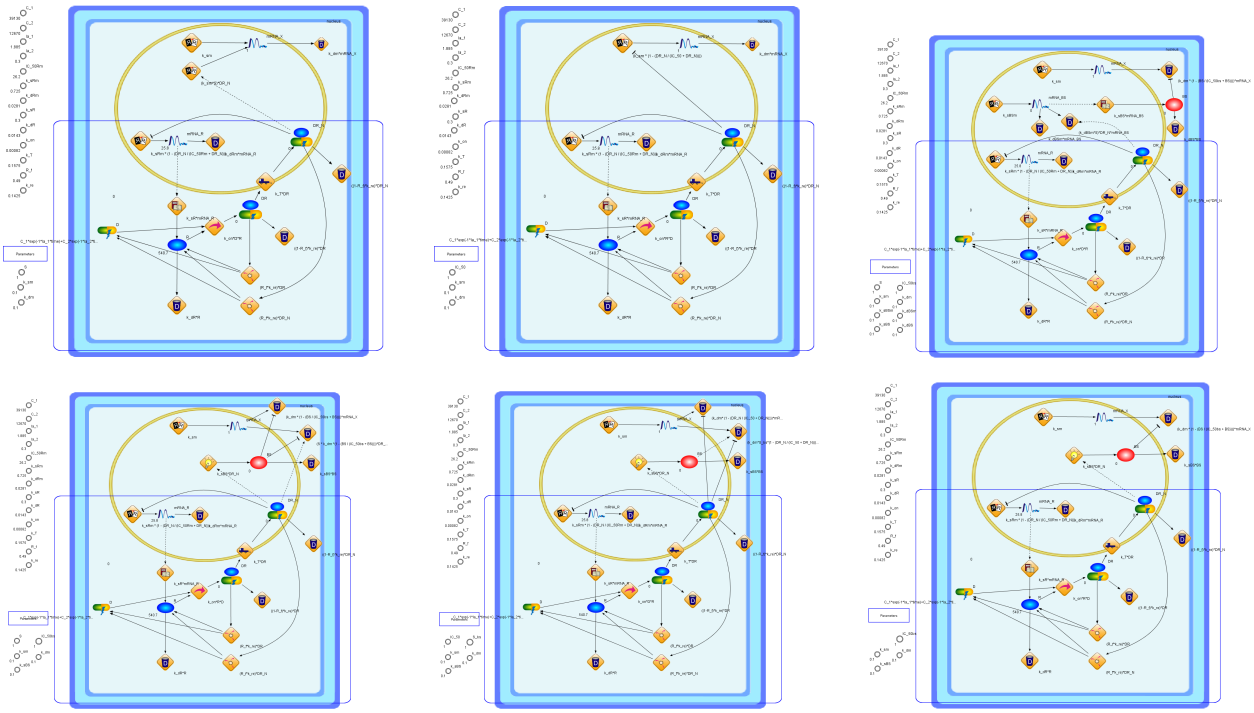

Figure 6.2: Six representative pharmacogenomic simulation models. Six representative pharmacogenomic simulation models (From top left to right, Model A, B, C12, DE10, DE12 and DE20). These models have high predictive power for many of 8,799 rat liver genes. These models are described by Cell Illustrator 5.0.

6.2.2 Data Assimilation for Parameter Estimation and Model Selection

To perform simulations by the pharmacogenomic models described in the previous section, we implemented them using Cell Illustrator [77], a software for biological pathway simulation based on hybrid functional Petri net with extensions. Six representative models in Cell Illustrator are shown in Fig. 6.2.

The differential equations of a candidate simulation model give the time evolution $\boldsymbol{x}$, which is a vector including simulation values of nodes, *e.g.*, a concentration of the drug-receptor complex 'DR'. In obtaining $\boldsymbol{x}$ in the candidate simulation models, Cell Illustrator uses DA 1.0 [59] in which simulation models described by differential equations are discretized to be simulated utilizing the hybrid functional Petri net with extension (HFPNe) [77]. In this framework, the mth candidate simulation model is represented by a function $\boldsymbol{f}_m$ of which $\boldsymbol{x}$ is obtained at evenly spaced time-points $\{\ldots, \boldsymbol{x}_{t-1}, \ldots, \boldsymbol{x}_{t-\Delta t}, \boldsymbol{x}_t, \boldsymbol{x}_{t+\Delta t}, \ldots, \boldsymbol{x}_{t+1}, \ldots\}$ through the procedure of HFPNe [77], where $\boldsymbol{x}_t$ is the vector of values of nodes at time t and Δt is a minute displacement. For simplicity, we represent, $\boldsymbol{f}_m$, as a function that maps $\boldsymbol{x}_{t-1}$ to $\boldsymbol{x}_t$, although $\boldsymbol{f}_m$ minutely updates simulation values toward $\boldsymbol{x}_t$. Then, we consider a stochastic simulation model of the form:

$$\boldsymbol{x}_t = \boldsymbol{f}_m(\boldsymbol{x}_{t-1}, \boldsymbol{\theta}_m, \boldsymbol{v}_t), \quad t \in \mathcal{T}, \tag{6.1}$$

where $\boldsymbol{\theta}_m$ is the parameter vector, $\boldsymbol{v}_t$ represents system noise according to a log-normal distribution $p(\boldsymbol{v})$, with the location parameter 0 and the scale parameter 0.1 (derived from the previous study [77]), and $\mathcal{T} = \{1, ..., T\}$ is the set of simulation time points.

Let $y_j[t]$ be the expression value of jth gene at time t. We consider the following model to connect the simulation model with the observed data:

$$y_j[t] = h(\boldsymbol{x}_t) + w_t, \quad t \in \mathcal{T}_{obs}, \tag{6.2}$$

where h is a function that maps simulation variables to the observation and w_t is a Gaussian observation noise, with mean 0 and variance σ^2. Here, $\mathcal{T}_{obs} \subset \mathcal{T}$ is the set of time points of time-course gene expression data. The model constructed by combining Eqs. (6.1) and (6.2) is called a nonlinear state space model. For simplicity, we assume $\mathcal{T}_{obs} = \mathcal{T}$, since it is easy to extend the general case of $\mathcal{T}_{obs} \subset \mathcal{T}$.

The parameter vector, $\boldsymbol{\theta}_m$, is estimated by the maximum likelihood method that chooses the values of $\boldsymbol{\theta}_m$ as the maximizer of the likelihood

$$L(\boldsymbol{\theta}_m, m|Y_{jT}) = \int p(\boldsymbol{x}_0) \prod_{t=1}^{T} p(y_j[t]|\boldsymbol{x}_t) p(\boldsymbol{x}_t|Y_{jt-1}, \boldsymbol{\theta}_m, m) d\boldsymbol{x}_1 \cdots d\boldsymbol{x}_T, \tag{6.3}$$

where $Y_{jT} = (y_j[1], ..., y_j[T])$ and $p(y_j[t]|\boldsymbol{x}_t)$ is a Gaussian distribution with mean $h(\boldsymbol{x}_t)$ and variance σ^2. In order to calculate the integral in Eq. (6.3), we use the particle filter (PF) [36,57]. The procedure to calculate the conditional distributions in PF is introduced in Chapter 2.

To find the best simulation model among all M candidate simulation models $\boldsymbol{f}_1, ..., \boldsymbol{f}_M$, we employ the BIC [96]. For the mth model $\boldsymbol{f}_m$ and the jth gene, BIC is defined by

$$\mathrm{BIC}(m, j) = -2 \log L(\hat{\boldsymbol{\theta}}_m, m|Y_{jT}) + \nu_m \log T, \tag{6.4}$$

where ν_m is the dimension of $\hat{\boldsymbol{\theta}}_m$. Therefore, the best simulation model for jth gene $\boldsymbol{f}^j_{best}$ can be obtained by

$$\boldsymbol{f}^j_{best} = \boldsymbol{f}_{\arg\min_m \mathrm{BIC}(m,j)}. \tag{6.5}$$

The derivation of BIC is briefly introduced in Chapter 2

6.3 Results

6.3.1 Time-course Gene Expressions

We analyze microarray time-course gene expression data of rat liver cells [47]. The microarray data were downloaded from GEO database (GSE487). The time-course gene expressions were measured at 0, 0.25, 0.5, 0.75, 1, 2, 4, 5, 5.5, 7, 8, 12, 18, 30, 48 and 72 hours (16 time-points) after receiving glucocorticoid. The data at time 0 hour are control (non-treated). The number

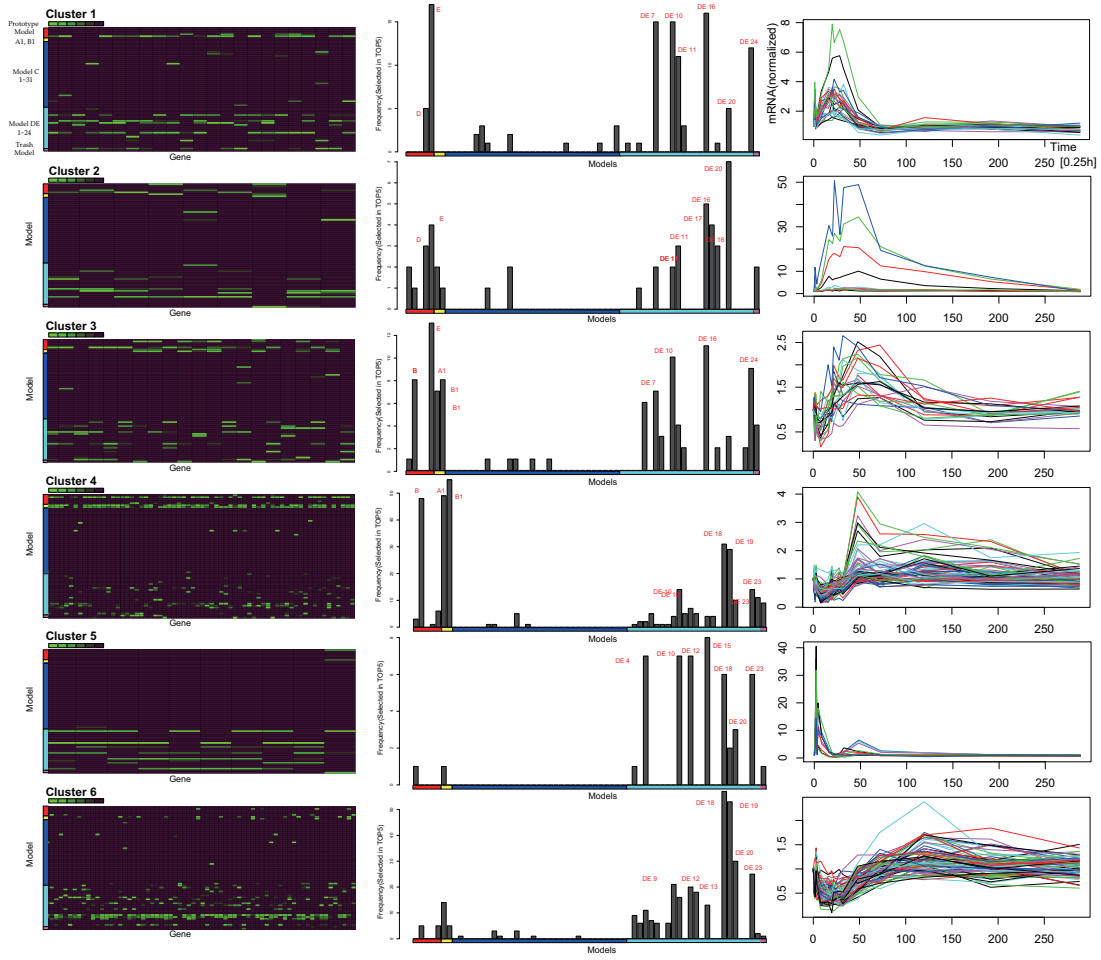

Figure 6.3: Top 5 simulation models for 197 gene. Top 5 simulation models for each gene in a cluster defined by Jin et al. [47] are represented by a heat map. The green elements means that the model well fits to the gene expression profiles. The histograms of the frequencies of the models selected as top 5 are shown in the middle panels, and gene expression profiles are also shown in the right panels.

of replicated observations is 2, 3 or 4 at a time point.

6.3.2 Results for Selected 197 Genes

First, we focused on 197 genes that were identified by the previous work [47] as the drug-affected genes by the clustering analysis. For the genes in each cluster, we explored which simulation models have better prediction power and the results are summarized in Fig. 6.3. According to the results obtained previously [47,106], the genes in the clusters 1, 2, 3, 4, 5 and 6 were reported to be well predicted by the Models "A", "A", "C", "D or E", "cell-cell interaction model"and "B, D or E", respectively. This result indicated that the genes in the cluster 1, 2 have almost same expression profiles. We should note that the cell-cell interaction model is not included in the five prototype models.

Fig. 6.3 shows the results for each cluster and the gene expression profiles. We can summarize the results as follows:

Cluster 1: The previous research [47] suggested that these genes are well predicted by Model A. However, interestingly, in our results, Model A was not selected. On the other hand, Models D and E and their extended models were selected many times. We presume the reason is that, particularly in the early observed time points, the profiles of these genes are not so simple.

Cluster 2: These genes are also suggested to be suitably predicted with Model A. However like cluster 1, similar results were obtained; for these genes, Model A was not selected in many times, and unlike cluster 1, the trash model was sometimes selected.

Cluster 3: The previous research [47] suggested that these genes fitted to Model C. However, in our results, not so many genes in cluster 3 are well predicted by Model C, but they fit to Models D and E and their extended models. We guess the reason is that Model C has more parameters than necessary. Therefore, in BIC, the second term, i.e., penalty for the number of parameters, takes large value and BIC cannot be small, so Model C and its extended versions were not selected. The same things can be said from the other works [41, 115].

Cluster 4: These genes were suggested to be fit with Models D or E. In our results, Model B and its extension and extension of Model A fit well, and Model E is especially fit, but Model D is not selected much. Instead, some extended versions of Models D and E fit well. The genes in cluster 4, we can see that some expression profiles do not vary widely. Such genes are well fit to Models A, B and its extensions, because of these simplicity. On the other hand, Models D and E and their extended models can follow complex behaviors and were selected in many times for other genes.

Cluster 5: Since these genes were judged to be fitted with the cell-cell interaction model that is not included in the five prototypes, these genes are not covered by our prepared models. However, in practice, the extended models of Model DE showed high predictive power for these genes. The expression profiles of these genes show sudden increasing patterns. Actually, our models can represent such dynamic patterns of gene expression profiles.

Cluster 6: These genes were suggested to be fit with Models B, D and E, but most genes were selected as the extended models of Models D and E. We presume the reason is that Models D and E are flexible and can follow various types of complex expression patterns.

6.3.3 Comprehensive Pathway Screening for 8,799 Genes

We next illustrate the results of pharmacogenomic pathway screening for whole 8,799 rat liver genes. Fig. 6.4 shows the results with heatmap of the selected top 5 models for each gene and time-course expression profiles of genes that are specific for Models C6, C12, DE10 and DE12. For each gene, we test the significance of the top ranked simulation model by using Smirnov–Grubbs test. If the expression profile of a gene was predicted very well by several simulation models, we cannot find pharmacogenomic mechanism specific for the gene. However, if only one model could predict the behavior of a gene, the model is a strong candidate that represents corticosteroid's mode-of-action for the gene. In such a case, we say the gene is specific for the above model.

Unlike the genes from the clustering analysis, the trash model, Model A, Model B and their

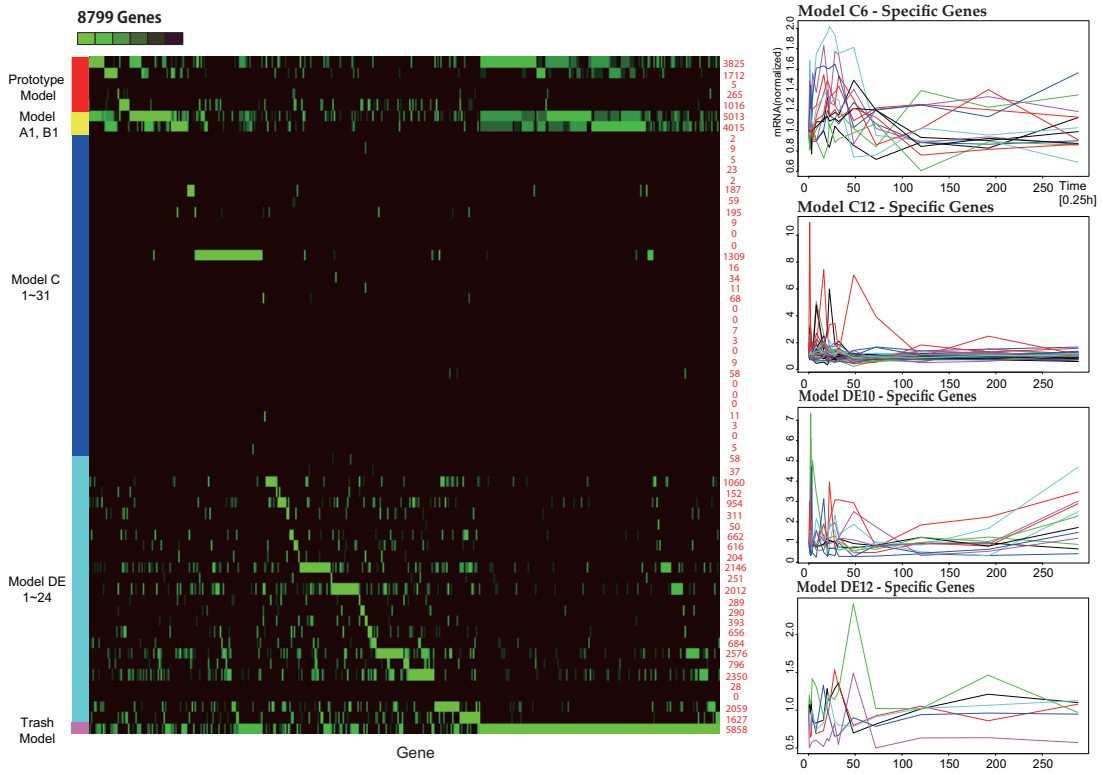

Figure 6.4: The result of comprehensive pharmacogenomic pathway simulation model screening. The result of comprehensive pharmacogenomic pathway simulation model screening. Heat map for top 5 models is shown from 63 simulation models for 8,799 rat liver genes. Time-course expression profiles are shown for genes that are specific for Models C6, C12, DE10 and DE12.

extensions were selected as top 5 in many times. We presume the reason is that, in the whole gene, there are many genes whose expression patterns are almost flat (not show clear dynamic patterns) and these models can follow them with a small number of parameters. Although the prototype D and E models were not selected many times, their extended models were frequently selected as top 5. This suggests that Models D and E can work well as the seed models for generating other simulation models with higher predictive power. The amount of genes obtained by this test varied widely depending on the models. From ModelA1, B1, C6, C12, C16, DE2, DE10, DE12 and DE20, we can obtained some specific genes. Interestingly, the number of genes fitting to Model C is relatively low, but many specific genes are obtained by Model C. It suggest that there are some expression profiles that can be represented by only the one of Model C. We then perform a functional analysis in order to reveal enriched gene functions for each set of Model-specific genes. For the functional analysis, we used Ingenuity and the results can be summarized as follows:

ModelC_6: These genes have function of “Cellular Assembly and Organization” and “RNA Post–Transcriptional Modification” and relate to “Protein Ubiquitination Pathway”. **Mod-**

elC_12: These genes are most interesting genes. These have "Amino acid Metabolism", "Nucleic Acid Metabolizm", "Cell Death", "Cellular Grows and Proliferation", "Drug Metabolism" and "Lipid Metabolism" and so on. Additionally, these genes relate to "Aldosterone Signaling Epithelial Cells" and "Glucocorticoid Recepter Signaling". Beneficial effects of Corticosteroid is inhibition of immune system and adverse effect is numerous metabolic side effects, including osteoporosis, muscle wasting, steroid diabetes, and others. Therefore, these result in ModelC_12 is biologically significant because these genes may have a function concerning metabolic side effects. **ModelDE_10**: These genes are also interesting. The functions are "Neurological Disease", "Organismal Injury and Abnormalities" and "Immunological Disease", and are affected by "Graft–versus–Host Disease", "Autoimmune Thyroid Disease, "T Helper Cell Differentiation" and so on. Because of the above therapeutic and adverse effects of CS, the function of these genes are also significant concerning immune system function. **ModelDE_12**: The functions of these genes are "Cellular Development", "Cardiovascular Disease", and "Hematological Disease". These are also affected by "EIF2 signaling".

We consider that such genes are important among 8,799 genes, because these were estimated to have a similar pathway and it may be difficult to collect these genes by clustering analysis simply using the gene expression profile.

6.4 Discussion

In this chapter, we proposed a computational strategy for automatic generation of pharmacogenomic pathway simulation models from the prototype simulation models that are built based on literature information. The parameters in the constructed simulation models were estimated based on the observed time-course gene expression data measured by dosing some chemical compound to the target cells. We constructed totally 63 pharmacogenomic simulation models on a pathway simulation software, Cell Illustrator, and used data assimilation technique for parameter estimation. For pathway screening, we introduce Bayesian information criterion for pathway model selection in the framework of data assimilation. We performed comprehensive pathway screening for constructed 63 pharmacogenoimc simulation models with gene expression data of rat liver cells treated with glucocorticoid.

The prototype five models fit to somewhat large number of genes well. However, there are more extended models that can predict the dynamic patterns of gene expressions better than the prototypes. This suggests that, from the prototype simulation models, we can automatically construct various extended simulation models and some of them could have higher prediction ability than the originals. Also, we performed a functional analysis to the sets of Model-specific genes identified by the Smirnov-Grubbs test. As shown above, some meaningful functions were found.

Chapter 7

An Efficient Method of Exploring Simulation Models by Assimilating Literature and Biological Observational Data

7.1 Background

Given the remarkable developments in biotechnology, many biomolecular reactions, *e.g.*, gene-protein and protein-protein interactions, have been discovered experimentally, and when combined, represent several parts of intracellular systems. In order to understand the dynamic behavior and control of these systems, simulation models have been constructed and evaluated by using biological observational data, *e.g.*, time-course RNA expression data [13, 61, 79, 108]. However, the results of such simulations can be incompatible with the observational data if molecules or reactions are omitted, or suspect molecules or reactions are included. Thus, such models should be improved on in order to ensure that simulation results are consistent with the observational data.

Schematic representations of regulatory structures are collected from the literature as pathway models, to which are the ascribed mathematical formulas, which represent the dynamic behavior of biomolecules, as simulation models based on biologically reliable models, *e.g.*, Michaelis-Menten equation [91] or S-system [92], which are described by differential equations. Their parameters, *e.g.*, initial concentrations and synthesis rates, are estimated to predict observational data maximally by some computational methods [11, 62, 67, 79, 89]. Then, the ability of these simulation models to predict the data is measured by some model criterion, *e.g.*, BIC [96]. Through such a procedure, *i.e.*, modeling and evaluation, in order to obtain better models, which can better predict observational data than can be done by the literature-recorded model, several computational approaches, *e.g.*, data assimilation [59, 77, 79, 108] and ensemble modeling [61, 93], have been developed. In our previous study, many candidate pathway mod-

els were automatically generated by partially changing the literature-recorded pathway models (template pathway models), which are tried to be improved, and their simulation models (candidate simulation models) were comprehensively evaluated to obtain better models using the data assimilation technique [40]. However, since evaluation of even a single simulation model is computationally costly, a large number of candidates cannot be handled. Furthermore, to evaluate one model that includes high-dimensional parameters is also computationally intensive, due to dimensionality.

In order to overcome the problem, we propose an efficient method for selectively and sequentially exploring candidates by exploiting the similarity of regulatory structures among candidate models (structure similarity). The proposed method uses an algorithm resembling the simulated tempering algorithm [35, 65, 71], which was developed to efficiently obtain probability distributions of parameter values with controlling the temperature parameter, to search a candidate model space. Thus, instead of evaluating candidate models comprehensively, the method employs an evaluated candidate as the current model according to the selection criterion and sequentially evaluates the next candidate with a regulatory structure similar to that of the current model. Furthermore, estimated parameter values for the current model are used to estimate parameter values of the next model, in order to reduce computational cost. Additionally, in order to assimilate simulated results to the observation data, we applied the nonlinear state space model [36, 57], which is a type of time-series modeling consisting of system and observational models, describing regulatory relationships and connecting simulated values to the data, respectively.

We have exemplified the method by applying it to pharmacogenomic pathways of corticosteroids [47]. In the Method Section, we have first introduced the pharmacogenomic pathways, including pharmacokinetics/dynamics [17, 38, 47, 84, 106], as template pathway models and explained the process of generating candidate pathway models from them. As a method for estimating the parameter values of their simulation models and ranking them according to their prediction ability, the data assimilation technique is explained briefly. In the results section, we have provided three experimental results using corticosteroid pharmacogenomics. First, we confirmed whether simulation modes sharing a certain amount of regulatory structure have approximately the same potential for predicting the data. Second, to show the effectiveness of the proposed method, we compared the performance of the proposed method with the comprehensive analysis [40] by using alternatively smaller number of candidates (195 models) and target genes (200 genes). Third, we applied the proposed method to an increased number of candidates (1,170 models), the relevance of 142 genes to corticosteroid validated biologically [47].

7.2 Methods

7.2.1 Corticosteroid Pharmacokinetics/dynamics and Pharmacogenomics

In this chapter, we also use the corticosteroid pharmacogenomic pathways [47] as illustrated in Fig. 7.1 used in Chapter 6.2.1.

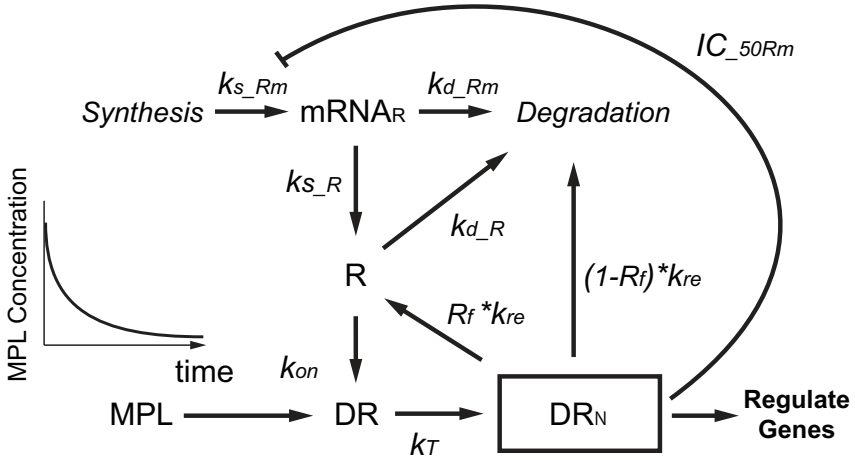

Figure 7.1: Corticosteroid pharmacokinetic/dynamic model. The corticosteroid pharmacodynamic model described here was developed by [84], where 'k_{s_Rm}', 'k_{s_R}', 'k_{on}' and 'k_T' are synthesis quantities, 'k_{d_Rm}', 'k_{d_R}' and 'k_{re}' are degradation quantities, and 'R_f' and 'IC_{50Rm}' are tuning parameters. 'Syn' and 'Deg' mean synthesis and degradation processes, respectively, 'D' is a concentration of corticosteroid modeled by corticosteroid pharmacokinetics [106], 'R' is a receptor of 'D', 'DR' is a complex of 'D' and 'R', '$\mathrm{DR_N}$' is 'DR' in nucleus, and '$\mathrm{mRNA_R}$' is a mRNA of 'R'.

Based on this pharmacokinetic/dynamic model (PK/PD model), [47] and other groups [41, 115] developed several types of intracellular pharmacogenomic pathway models (PG models), including corticosteroid ('$\mathrm{DR_N}$' or 'DR' in PK/PD model) and corticosteroid-induced genes. As an example, two pathways of PG models are illustrated in Fig. 7.2. PG models are used as an application example of the proposed method to explore improved simulation models for observational data, in the results section. Thus, PG models are handled as template pathway models.

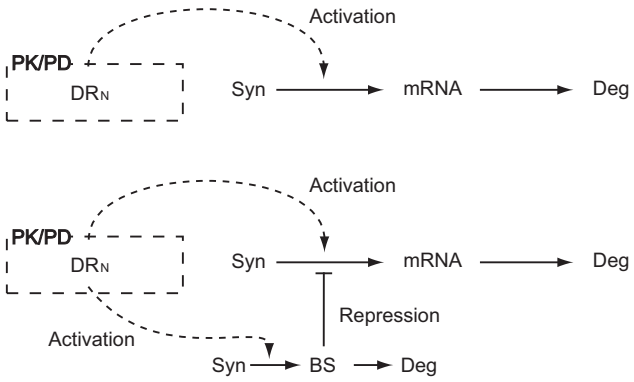

Figure 7.2: Examples of pharmacogenomic pathway models. These are examples of pharmacogenomic pathway models [47]. 'mRNA' is a mRNA regulated by the corticosteroid and 'BS' is an intermediate biosignal. A dotted rectangle 'PK/PD' means the pharmacokinetic/dynamic model represented in Fig. 7.1. *Syn* and *Deg* mean synthesis and degradation processes, respectively.

7.2.2 Create Candidate Pathway Models from Template Pathway Models

To explore improved simulation models, we require candidate pathway models that have been modified partially from the template ones. In our proposed method, we first set a regulator and a target, which are the up-stream and the down-stream biomolecules on a part of the template pathways that is tried to be improved. For example, in corticosteroid pharmacogenomics of Fig. 7.2, the regulator is drug ('DR_N') and the target is a corticosteroid-induced gene ('mRNA'). Then, the target is assumed to be directly or indirectly influenced by the regulator. This relationship can be illustrated in Fig. 7.3(a) as a type of feed-forward loop (FFL) [97]. Considering the existence and the type of regulation, we can obtain 15 types of regulatory structures, as illustrated in Fig. 7.3(b). Additionally, 15 structures are extended, as illustrated in Fig. 7.3(c), by increasing the number of intermediate nodes to delay the effect and adding a node, 'Product', which represents a product of the target, to promote self-regulation. We apply these regulatory structures (basic structures) to the relationship between the regulator and the target using the other regulation and regulatory functions in the template pathways. Since the components of the basic structures have been found commonly in biological systems [44, 97, 116], they can conceivably extend the template pathways.

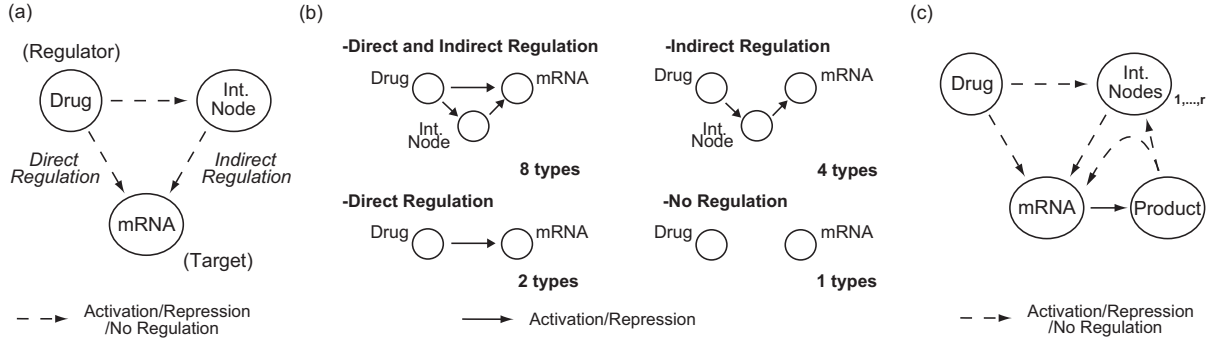

Figure 7.3: Basic regulatory structures as a form of feed-forward loop. These figures illustrate the basic regulatory structures as a form of feed-forward loop [97]. In order to generate candidate pathway models, we consider a regulatory structure as illustrated in (a), where 'Int. Node' represents an intermediate node. From (a), we construct four types of regulatory structures illustrated in (b), and 15 basic structures are captured by considering their types of regulation, *i.e.*, activation and repression. Then, as an extension, we increase the number of intermediate nodes to delay the effect and self-regulation through their product 'Product' to promote self-regulation as summarized in (c). For example, when selecting activation by 'Regulator' and removing regulations by both of 'Product' and 'Int. Node', the extracted structure corresponds to 'Direct Regulation' in (b). Furthermore, when adding regulation by 'Product' extensions of this basic structure can be obtained.

In order to generate candidates covering these basic structures in combination with the template pathway models, we first established a large pathway model, termed an integrated model, by integrating them. An integrated model for corticosteroid pharmacogenomics is illustrated in Fig. 7.4. The procedure for generating the integrated model is summarized as follows:

1. Integrate components of the template pathway models, *i.e.*, direct activation/repression of 'mRNA' by 'DR_N', indirect activation of 'mRNA' by 'DR_N' and direct repression of

'mRNA' by 'DR'. Here, all intermediate nodes in the template pathways, *e.g.*, proteins affecting 'mRNA', are represented as an intermediate biosignal 'BS' for ease of viewing, *i.e.*, BS_1, ..., BS_r.

2. The basic structure is combined to the relationships between 'DR_N' and 'mRNA' in the obtained model. Here, their regulatory functions are given from those of the template pathway models.

3. The number of 'BS' (intermediate biosignal), r, is set at 3 (= 2 in the template pathways).

In Fig. 7.4, 'mRNA' represents a corticosteroid-target mRNA, the dotted block corresponds to the corticosteroid pharmacokinetics/dynamics in Fig. 7.1, *Syn* is a synthesis process, and *Deg* is a degradation process. Dotted and solid arrows mean regulation and fixed processes (synthesis, degradation, and activation between $BS_{1,2,3}$), respectively. Nevertheless, we can delay the regulatory effect by setting the values of threshold parameters sufficiently high; this also shortens an effective term. This means that, since the regulatory effect can be valid when the abundance of regulator is larger than the threshold, a high threshold value causes not only a delay but also the abbreviation of operating time by the regulator. Then, the number of 'BS' is increased. The effect of the transcriptional cascade through intermediate biosignals was discussed in [44]. These intermediate biosignals are summarized as '$BS_{1,2,3}$' ('BS_3' is upstream and 'BS_1' is downstream in the regulatory order). We should mention that, although the production process of each intermediate biosignal is described as a one-step model (a set of synthesis and degradation processes), in some cases, two-step models, *e.g.*, consisting of production processes of mRNA and protein, may be better to represent the dynamics of the biosignals, for example, in order to introduce additional delays. In addition, to avoid the curse of dimensionality and the combination explosion, we focused on the dynamic behavior of the most downstream gene in candidate models and then estimated genes corresponding to intermediate biosignals after obtaining their simulation dynamics in the results section.

Finally, we generate candidate pathway models from the integrated model by selecting the type and the existence of regulation. In order to create candidate models covering all basic structures, we consider the following rules (i)-(v):

(i) 'DR_N' can regulate either the synthesis or the degradation process of 'mRNA' through either activation or repression,

(ii) 'BS_1' and 'Product' can regulate only the synthesis process of 'mRNA' by either activation or repression ('$BS_{2,3}$' cannot regulate 'mRNA' directly),

(iii) 'DR_N' can also activate or repress only a single node of '$BS_{1,2,3}$'.

(iv) 'Product' can activate or repress only a single node of '$BS_{1,2,3}$'.

(v) 'DR' can repress the degradation process of 'mRNA'.

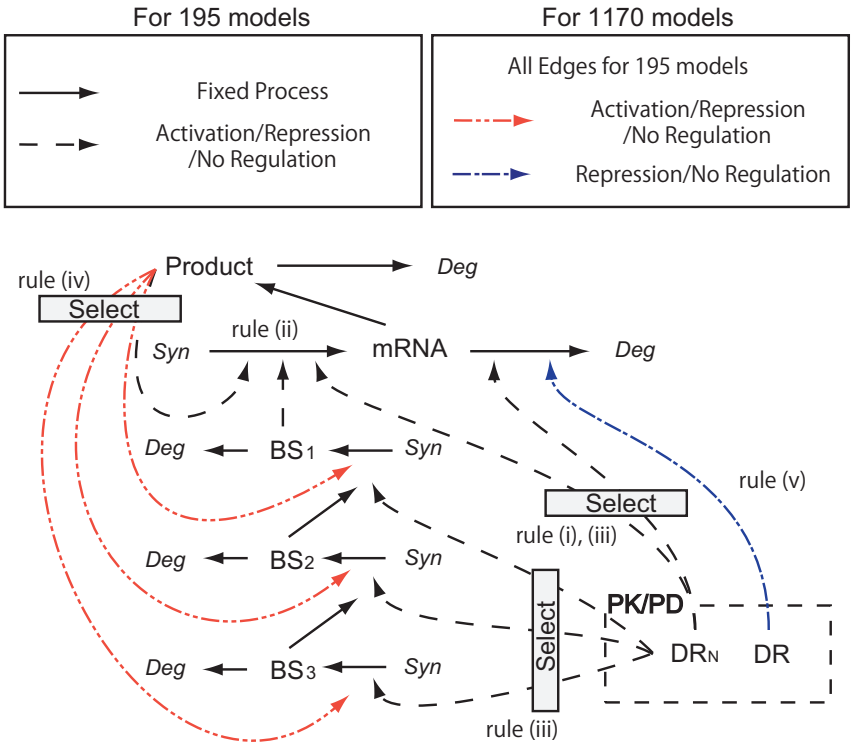

Figure 7.4: Integrated model for constructing candidate pathway models. This pathway model, termed the integrated model, is constructed to create candidate pathway models. 'BS_1', 'BS_2' and 'BS_3' are intermediate biosignals and 'Product' is a product of 'mRNA'. By selecting an edge from edges annotated 'select' and the type of regulation (Activation/Repression/No Regulation) of dotted edges, we can generate 1,170 (All-Models) and 195 (Core-Models) candidate pathway models. For Core-Models, edges and the type of regulation in the left top block are considered and the remainder are deleted.

To compare the proposed method with the existing method [40], we constructed 195 pathway models (Core-Models) using rules (i)-(iii), because these three rules were adopted in previous studies [41,115] and Core-Models are feasible as an application of the existing one. Using rule (i)-(v), 1,170 candidate pathway models (All-Models) are constructed to find improved simulation models for corticosteroid-induced genes. Note that Core-Models are included in All-Models.

7.2.3 Data Assimilation

In this chapter, the estimation of the parameter values and the evaluation of candidate models are according to the same way with introduced in Chapter 6.2.2.

7.2.4 Model Transition Rule for Efficient Model Exploration

Since a numerical optimization of the non-differentiable parameters is involved in the process of choosing the best simulation model, the model evaluation is computationally costly [40]. However, if we know pairs of candidate simulation models that express similar qualitative dynamics, their ability to predict the data can be highly correlated. In this case, starting from a simpler candidate, to sequentially and selectively evaluate candidates for which possible dynamics are

similar to those of previously evaluated models having higher prediction ability, can be an efficient way to find the best model. One approach to obtaining such correlated pairs of candidates involves the following procedure. At first, for each candidate simulation model, the BIC scores of Eq. (6.4) are calculated for several genes and the sequence of the BIC scores is regarded as the feature vector of the model. Next, Spearman's rank correlations in the model are obtained to select highly correlated pairs of candidates setting a threshold. If the feature vector is constructed using a sufficiently large data set, these selected pairs can be considered to express similar qualitative dynamics. As an example, the relationships network among Core-Models, showing their connections, is illustrated in Fig. 7.5. The relevant details are described in the results section.

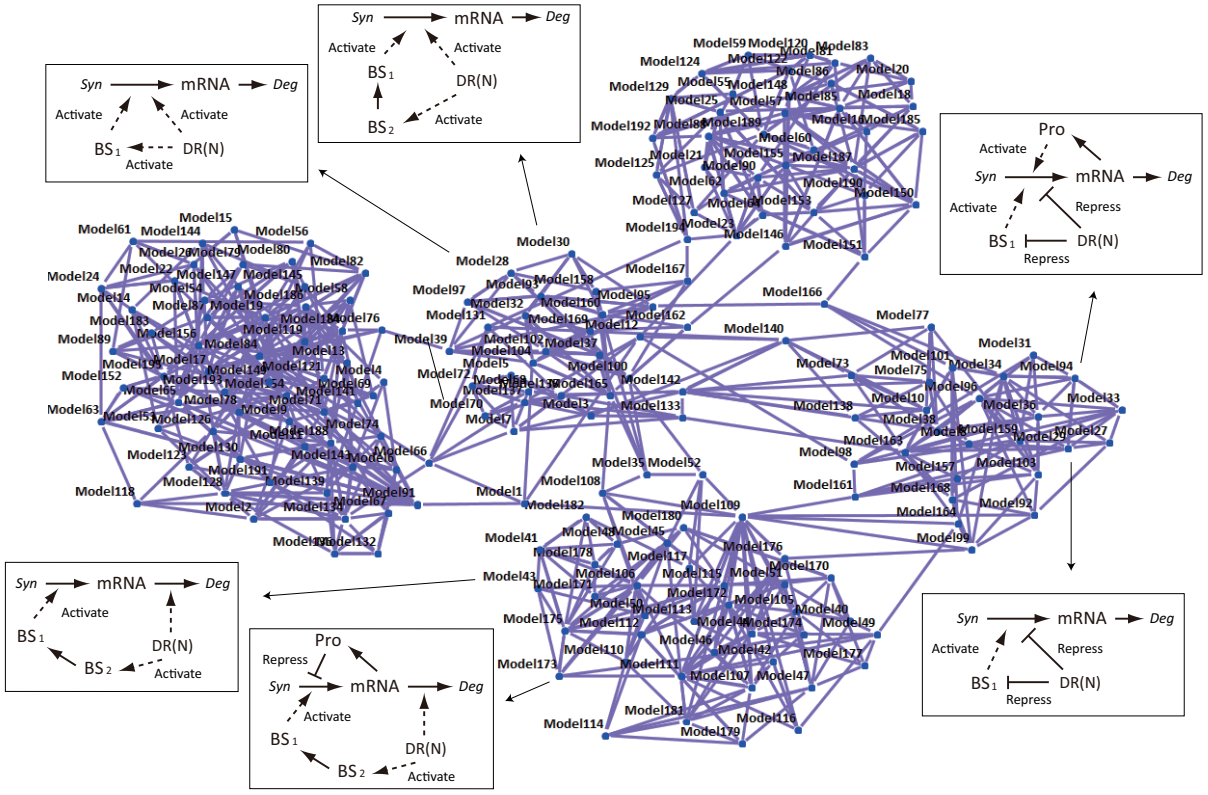

Figure 7.5: Relationships network of candidate simulation models. This figure shows relationships among Core-Models (each node represents a candidate simulation model) by drawing edges between correlating candidate simulation models. These edges are drawn according to the following criteria. For each candidate simulation model, the BIC scores of Eq. (6.4) were calculated for Test-Genes in Core-Models and we regarded the sequence of the BIC scores as the feature vector of the model. For a candidate simulation model, we drew edges between the model and the five models that have five highest Spearman's rank correlations to the model. As an example, pathway models of six candidate simulation models are illustrated.

However, in this approach, we must calculate the BIC scores in Eq. (6.4) of all candidate simulation models for a sufficient amount of gene expression data in order construct the relationships network. On the other hand, we observed that, when candidate pathway models include the same or a similar basic structure, their simulation models can express similar qualitative

dynamics. Therefore, we can connect candidates according to the similarity of their dynamics based on these basic structures. Then, we propose the model transition rule described in Fig. 7.6 instead of the relationships network as: (a) add or reduce an intermediate biosignals to delay or shorten the effect; (b) change the type of regulation (activation or repression) of the direct regulation from 'DR_N'; (c) change the indirect regulation while retaining the effect; (d) add or remove either a direct or an indirect regulation; (e) add, remove, or change self-regulation through 'Product'. These rules give the transition between the candidate simulation models that can express similar simulation dynamics, except for (b). Thus, rule (b) was prepared to enable the algorithm to move to slightly similar models for escaping from local minimum. We should note that an additional rule, (f) add or remove repression of 'mRNA' by 'DR', was applied to All-Models.

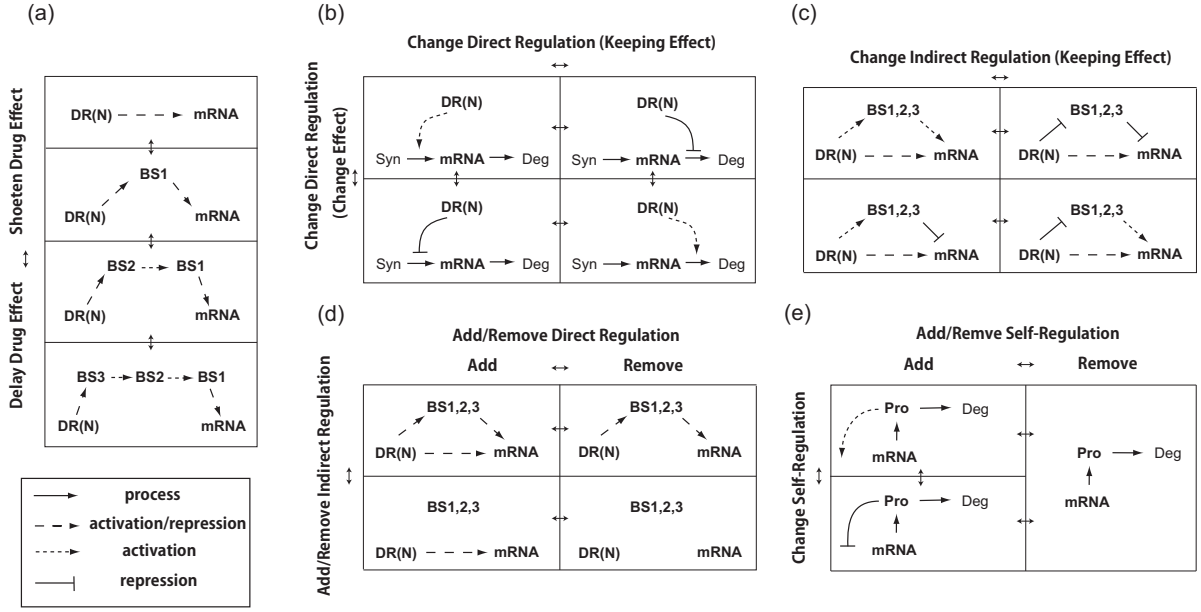

Figure 7.6: The model transition rule. Instead of using the relationships network, the model transition rule is used to select the next candidate simulation model from the current model. One of rules (a)-(e) is applied in selecting the next model according to the model transition probability $q(m_{current}, m_{next})$. The rule is constructed by completely or partially retaining the basic structures.

7.2.5 Simulated Tempering Like Exploration Algorithm

For an exploration algorithm using the model transition rule, we consider following three requirements. Firstly, all candidate simulation models should be explored if the algorithm is executed for a sufficiently long time. Secondly, the algorithm should escape from a local minimum quickly. Thirdly, simulation models should be recursively evaluated since it is difficult to estimate parameter values maximizing the prediction ability. Then, we propose a simulated tempering like exploration algorithm (STE algorithm). The simulated tempering (ST) algorithm [35, 65, 71] was proposed to obtain distributions of the parameter vector $\boldsymbol{\theta}_{ST}$ for the Boltzmann distribu-

tion $b_T(\boldsymbol{\theta}_{ST}) \propto \exp\{-H(\boldsymbol{\theta}_{ST})/T\}$, where $H(\boldsymbol{\theta}_{ST})$ and T are an objective function, often a log-likelihood function, and a temperature, respectively. ST algorithm treats the temperature T as a dynamic variable and adds a one-dimensional temperature ladder T_k ($T_1 < \ldots < T_k < \ldots < T_K$) to change T. We use the algorithm by setting m=$\{1, \ldots, M\}$ as $\boldsymbol{\theta}_{ST}$ and apply to the problem.

Let $q(m, m')$ be the model transition probability from $\boldsymbol{f}_m$ to $\boldsymbol{f}_{m'}$.

1. Set the initial temperature $T = T_0$ and evaluate a simple simulation model $\boldsymbol{f}_{m_{simple}}$ to obtain $P(Y_{jN}|m_{simple})$ by data assimilation, and set $\boldsymbol{f}_{m_{simple}}$ as $\boldsymbol{f}_{m_{current}}$.

2. Decide upon the next candidate simulation model $\boldsymbol{f}_{m_{next}}$ from the current simulation model $\boldsymbol{f}_{m_{current}}$ according to $q(m_{current}, m_{next})$ and calculate $P(Y_{jN}|m_{next})$.

3. If $P(Y_{jN}|m_{next})$ is higher than $P(Y_{jN}|m_{next})_{highest}$, store $P(Y_{jN}|m_{next})$ as $P(Y_{jN}|m_{next})_{highest}$. Initially, $P(Y_{jN}|m_{next})_{highest}$ is stored zero.

4. Set the next temperature T_{next} according to the temperature radder. Generally, the probability $P_{a,b}$ from temperature T_a to T_b is set as $P_{0,1} = P_{K,K-1} = 1$ and $P_{a,a+1} = P_{a,a-1} = 0.5$.

5. Accept $\boldsymbol{f}_{m_{next}}$ and T_{next} as the current model and the current temperature with probability

$$\beta_j(m_{current}, m_{next}) = \min\{1, \frac{b_{T_{next},j}(m_{next})q(m_{next}, m_{current})}{b_{T_{current},j}(m_{current})q(m_{current}, m_{next})}\}, \tag{7.1}$$

$$b_{T,j}(m) = \exp\{\frac{-H_j(m)}{T} + g_{T,j}\}, \tag{7.2}$$

$$g_{T_{k+1},j} - g_{T_k,j} = (\frac{1}{T_{k+1}} - \frac{1}{T_k})\frac{E_j(T_k) + E_j(T_{k+1})}{2}, \tag{7.3}$$

$$H_j(m) = -\log\{P(Y_{jN}|m)_{highest}\}, \tag{7.4}$$

   where $E_j(T_k)$ is an expectation value of $H_j(m)$ at temperature T_k. Alternatively, if $\boldsymbol{f}_{m_{next}}$ and T_{next} were not accepted, retain the current model and temperature, and then return to Step (2) to seek another candidate.

6. Repeat steps 2-5 until the iteration is maximized.

We used the results of the first several steps to approximate $E_j(T_k)$. An example of STE algorithm is illustrated in Fig. 7.7. Note that we can evaluate some simple candidates and set one of them as $\boldsymbol{f}_{current}$ in step 1 of STE algorithm.

7.2.6 Efficient Parameter Estimation in STE algorithm

PF is used to calculate the marginal probability distribution of parameter $\boldsymbol{\theta}_m$ for a simulation model $\boldsymbol{f}_m$ in step 2 of STE algorithm. While using PF, we set the ranges of prior distributions of parameter values $\boldsymbol{\theta}_m$ by referring to the literature, if available, or widely set the ranges. Since setting an appropriate prior distribution is highly effective, the prior distribution of $\boldsymbol{f}_{next}$ should be determined by referring to estimated parameter values of $\boldsymbol{f}_{current}$. Hence, $\boldsymbol{f}_{current}$

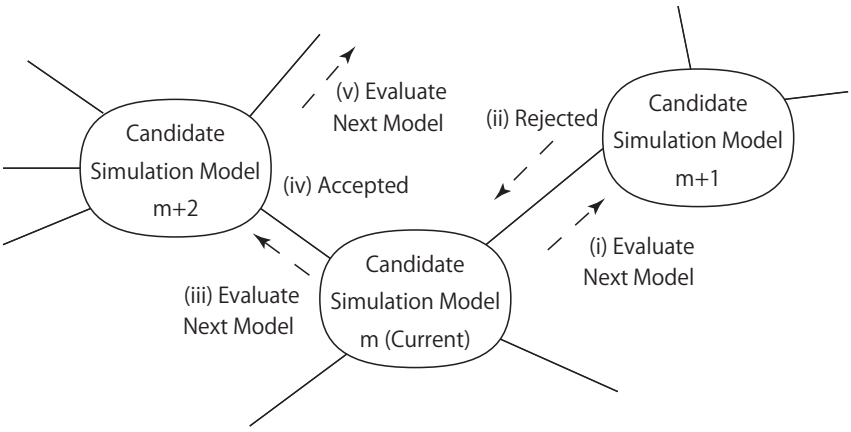

Figure 7.7: Simulated Tempering Like Exploration. This illustrates an example of the simulated tempering-like exploration algorithm. The algorithm accepts or rejects the next candidate as the current model according to the criterion.

and $\boldsymbol{f}_{next}$ share a certain amount of regulatory structure; some common parameter values are directly utilized to set prior distributions of corresponding parameters. Nevertheless, in some cases, parameter values in $\boldsymbol{f}_{current}$ cannot be directly utilized for setting the prior distribution of $\boldsymbol{f}_{next}$. In this case, we set the ranges of prior distributions widely. Additionally, we here derived a constraint for Syn/Deg by focusing on simulation expression profiles of 'mRNA' in a steady state condition in the rest of this section.

7.2.6.1 Case: Direct Regulation

We consider the model transition from the left top model in Fig. 7.6(b) to the right top model in Fig. 7.6(b). The left top model in Fig. 7.6(b) is described by

$$\frac{d}{dt}\mathrm{mRNA} = Syn \cdot (1 + F_{\mathrm{DR_N}}) - \mathrm{mRNA} \cdot Deg, \tag{7.5}$$

$$F_{\mathrm{DR_N}} = \frac{S_{\mathrm{DR_N}} \cdot \mathrm{DR_N}}{IC_{\mathrm{DR_N}} + \mathrm{DR_N}}, \tag{7.6}$$

and the right top model in Fig. 7.6(b) is described by

$$\frac{d}{dt}\mathrm{mRNA} = Syn - (1 - G_{\mathrm{DR_N}}) \cdot \mathrm{mRNA} \cdot Deg, \tag{7.7}$$

$$G_{\mathrm{DR_N}} = \frac{S'_{\mathrm{DR_N}} \cdot \mathrm{DR_N}}{IC'_{\mathrm{DR_N}} + \mathrm{DR_N}}, \tag{7.8}$$

where Syn and Deg are synthesis and degradation quantities of 'mRNA', respectively, $S_{\mathrm{DR_N}}$ is an amplification parameter for the mRNA synthesis process from the drug, $IC_{\mathrm{DR_N}}$ is a tuning parameter for the mRNA synthesis process from the drug, $S'_{\mathrm{DR_N}}$ is a reduction rate for mRNA degradation process from the drug and $IC'_{\mathrm{DR_N}}$ is a tuning parameter for the mRNA degradation

process from the drug.

Let mRNA_{ss} be the expression level of mRNA in the steady state. Eq. (7.5) in the steady state is

$$\mathrm{mRNA}_{ss} = \frac{Syn \cdot (1 + F_{\mathrm{DR_N}})}{Deg}. \tag{7.9}$$

Because $\mathrm{DR_N}$ becomes zero in the steady state, $F_{\mathrm{DR_N}}$ also becomes zero, and then mRNA_{ss} becomes $\frac{Syn}{Deg}$. Through a similar procedure, mRNA_{ss} from Eq. (7.7) yields the same value as that from Eq. (7.5), *i.e.*, $\frac{Syn}{Deg}$. Then, $\frac{Syn}{Deg}$ in the left top model of Fig. 7.6(b) can be applied to the parameter estimation of the right top model in Fig. 7.6(b) as a restriction in distribution of particles. This transformation can be applied to the adding/removing/changing transition of the direct regulation in Figs. 7.6(b) and (d).

7.2.6.2 Case: Indirect Regulation

We can also apply the same steady state solution for '$\mathrm{BS}_{1,2,3}$' of Fig. 7.6(a) (which updates the model from the bottom one to top one or from the top to the bottom in this figure). We exemplify using Eq. (7.5) and a complicated simulation model, including direct activation, indirect repression, and self-activation, given by

$$\begin{aligned}\frac{d}{dt}\mathrm{mRNA}^{comp} &= Syn^{comp} \cdot (1 + F_{\mathrm{DR_N}}) \cdot (1 - G_{\mathrm{BS}_1}) \cdot (1 + F_{\mathrm{Pro}}) \\ &\quad - \mathrm{mRNA} \cdot Deg^{comp}, \end{aligned} \tag{7.10}$$

$$G_{BS}(\mathrm{BS}_1) = \frac{S'_{\mathrm{BS}_1} \cdot \mathrm{BS}_1^{\gamma}}{IC'^{\gamma}_{\mathrm{BS}_1} + \mathrm{BS}_1^{\gamma}}, \tag{7.11}$$

$$F_{\mathrm{Pro}}(\mathrm{Pro}) = \frac{S_{\mathrm{Pro}} \cdot \mathrm{Pro}}{IC_{\mathrm{Pro}} + \mathrm{Pro}}, \tag{7.12}$$

where S'_{BS1} is the reduction rate of 'BS_1', IC'_{BS1} is a tuning parameter of 'BS_1', γ is a tuning parameter for the sensitivity of 'BS_1', S_{Pro} is an amplification parameter of 'Product', IC_{Pro} is a tuning parameter of 'Product', and the notation $\cdot^{comp}$ stands for the case of this model. BS_1 and Pro represent the level of 'BS_1' and 'Product', respectively.

mRNA_{ss} for Eq. (7.10) is given by

$$\mathrm{mRNA}^{comp}_{ss} = (1 - G_{\mathrm{BS}_1}(\mathrm{BS}_{1,ss}))(1 + F_{\mathrm{Pro}}(\mathrm{Pro}_{ss})) \cdot \frac{Syn^{comp}}{Deg^{comp}}, \tag{7.13}$$

where $\mathrm{BS}_{1,ss}$ and Pro_{ss} are the level of 'BS_1' and 'Product' in the steady state, respectively. By solving the equation $\mathrm{mRNA}^{comp}_{ss} = \mathrm{mRNA}_{ss}$, we have

$$\frac{Syn^{comp}}{Deg^{comp}} = \frac{Syn}{(1 - G_{\mathrm{BS_{BS}}}(BS_{1,ss}))(1 + F_{\mathrm{Pro}}(\mathrm{Pro}_{ss})) \cdot Deg}. \tag{7.14}$$

This can be applied as a restriction for setting prior distributions. Each term representing regulatory effects in Eq. (7.13) can be removed and applied to another cases in Fig. 7.6, if necessary.

7.3 Results

7.3.1 Time-course Gene Expression

We have analyzed microarray time-course gene expression data from rat liver cells [47]. The microarray data were downloaded from the GEO database (GSE487). The time-course of gene expression was measured at 0, 0.25, 0.5, 0.75, 1, 2, 4, 5, 5.5, 7, 8, 12, 18, 30, 48, and 72 h (16 time-points) after corticosteroid stimulation. The data at time 0 h are considered to be controls (untreated). There are two, three, or four replicated observations for each time point.

7.3.2 Relationships among Candidate Models Represented by a Comprehensive Search

First, we investigated the assumption that the simulation models that share a certain amount of regulatory structure (the basic structures) can show equal prediction ability. For each simulation model in Core-Models, the BIC scores of Eq. (6.4) were calculated for 200 genes (Test-Genes) and we regarded the sequence of the BIC scores as the feature vector of the model. Test-Genes were selected as centers by k-means ($k = 200$) clustering of the 8,799 gene expression profiles. Thus, totally 39,000 (200×195) simulation models were evaluated. In Fig. 7.5, five clusters of models that are highly correlated (Spearman's rank correlation) are observed, where an edge was drawn if and only if a correlation coefficient was higher than the fifth highest one for each simulation model. In this figure, we find that simulation models sharing the basic structures among the candidates are connected by edges. Therefore, we confirmed that candidates sharing the same or similar basic structures can have a similar tendency in prediction ability.

7.3.3 Comparison of Comprehensive Search and Proposed Method

To demonstrate the validity of the proposed method, we compared it to the previously reported comprehensive search (C-Search) [40]. Core-Models and Test-Genes were used in this comparison because using All-Models is not computationally feasible for C-Search.

For C-Search, we obtained the complete results of Core-Models for 200 Test-Genes. Next, we applied the proposed method and obtained the ranking of Core-Models for each of the Target-Genes. Finally, we compared the performance of the proposed method in terms of the correct rate, which represents the percentage of the best models (ranked as the top model by the proposed method) satisfying the following two conditions: it is the same as the correct model (ranked as the top model by C-Search) and its BIC score is lower than that of the second top model by C-Search.

Fig. 7.8 summarizes the comparison of calculation speed (blue bar) and the correct rate (other color bars). The colors in Fig. 7.8 are assigned as follows: (i) The green bar represents the correct rate. It also includes the case of finding a candidate model that has a lower BIC score than that of the correct model. (ii) The light blue bar represents the incorrect rate caused by failure of the parameter estimation. Thus, the method searched for the correct model from the initial candidate using STE algorithm, while the BIC score is higher than that of the second top model by C-Search. (iii) The red bar also represents the incorrect rate caused by failure of the model transition. This indicates that the method could not even evaluate the correct model. (iv) The blue bar represents the calculation time for the proposed method for each gene. C-Search requires approximately 400 h on average for a gene in Test-Genes. The horizontal axis represents the measured time points of the proposed method. In order to compare the performance in terms of execution time, the proposed method was executed for a given time. The left and right vertical axes represent the percentages of the number of candidates and calculation time relative to C-Search, respectively. For instance, 10 [%] on the left vertical axis means 20 [genes] in this case. Note that all calculations were executed on an HGC super computer (`http://www.hgc.jp/english/`) and its executing time was limited to 48 h.

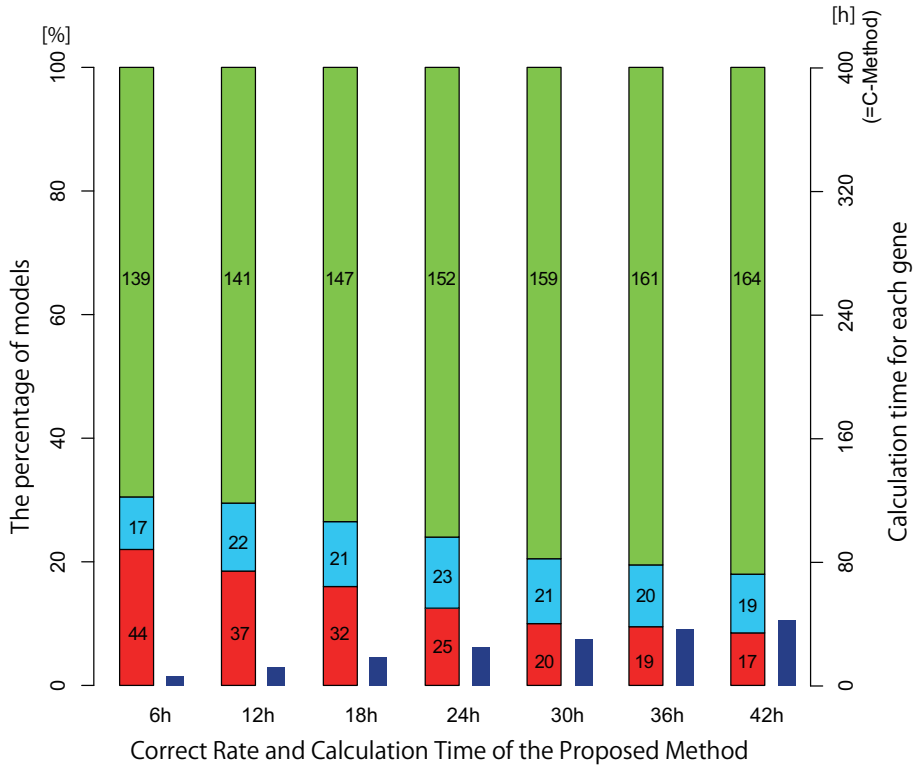

Figure 7.8: Comparison results for the proposed method and C-Search. This figure illustrates the comparison results of the proposed method with the C-search in terms of the correct rate and execution time. The green bar represents the correct rate. It means that the rate at which the best simulation model that is consistent with the correct model is obtained. The light blue bar represents the incorrect rate caused by failure of parameter estimation. The red bar also represents the incorrect rate caused by failure of model transition. The blue bar represents the calculation time for the proposed method for each gene. The results of the proposed method are evaluated at 6, 12, 18, 24, 30, 36, and 42 h.

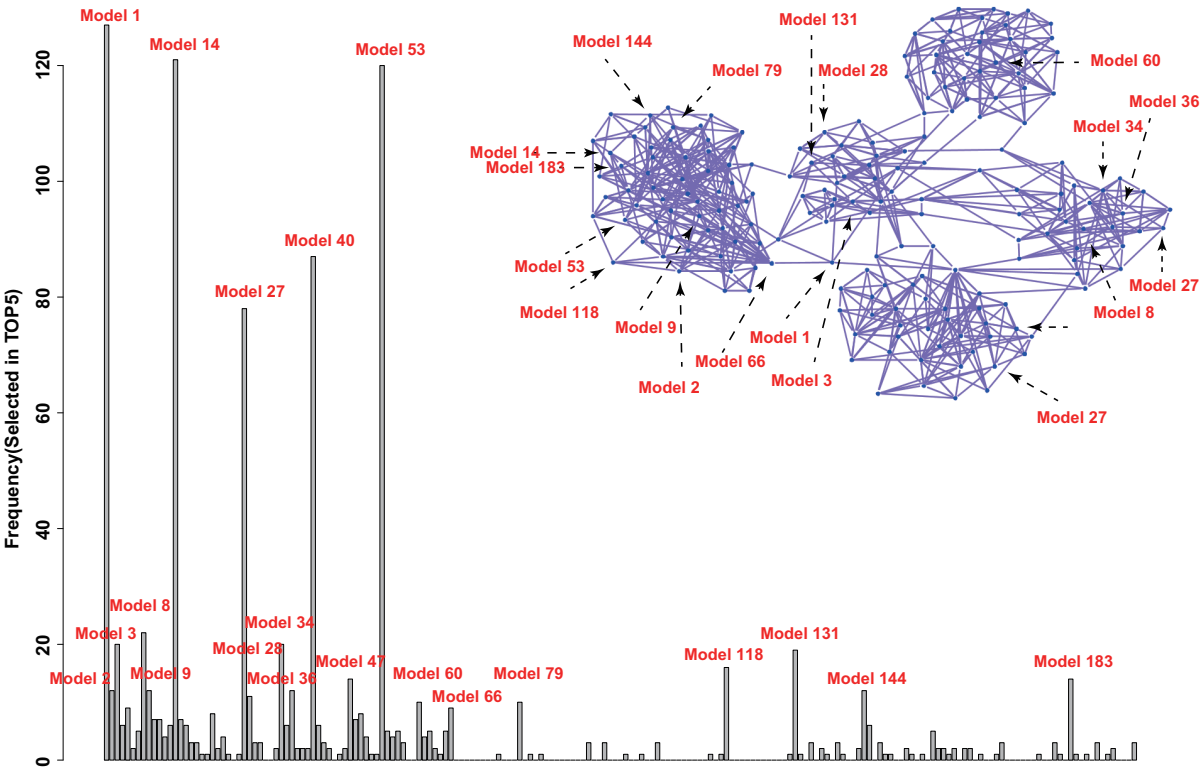

Figure 7.9: Top five candidate simulation models selected by C-Search. This illustrates a histogram of frequencies of candidate simulation models selected as top five for the Test-Genes by C-Search. The right top graph is the same as in Fig. 7.5. Highly selected candidates are indicated their positions.

Fig. 7.8 indicates that the proposed method attained more than 80% of the correct answers within 10[%] to 15 [%] of the calculation time required by C-Search. In addition, we have also shown histograms for the top five candidate simulation models for each gene selected by C-Search and the proposed method in Figs. 8 and 9, respectively. In these figures, 10 of the top 10 and 16 of the top 20 selected candidates are the same. Thus, the high degree of similarity suggests that the proposed method selectively explored candidates that have high prediction ability, and did so within a short span of time.

7.3.4 Exploring Better Corticosteroid Pharmacogenomics

We applied the proposed method using the corticosteroid pharmacogenomic pathways and explored simulation models having higher prediction ability for the observational data than those reported in the literature. For this purpose, we focused on 142 unique genes (PG-Genes), which have been biologically validated in terms of their relevance to corticosteroid responses [47], and applied these to All-Models. The template pathways of PG-Genes can be found in [47] and examples are illustrated in Fig. 7.2.

We obtained 142 best simulation models, which had the lowest BIC score of Eq. (6.4) among All-Models for each gene. In these models, only the three simulation models of 'Odc1', 'Map1lc3b', and 'Ccng1' could absolutely match the template simulation models of these genes,

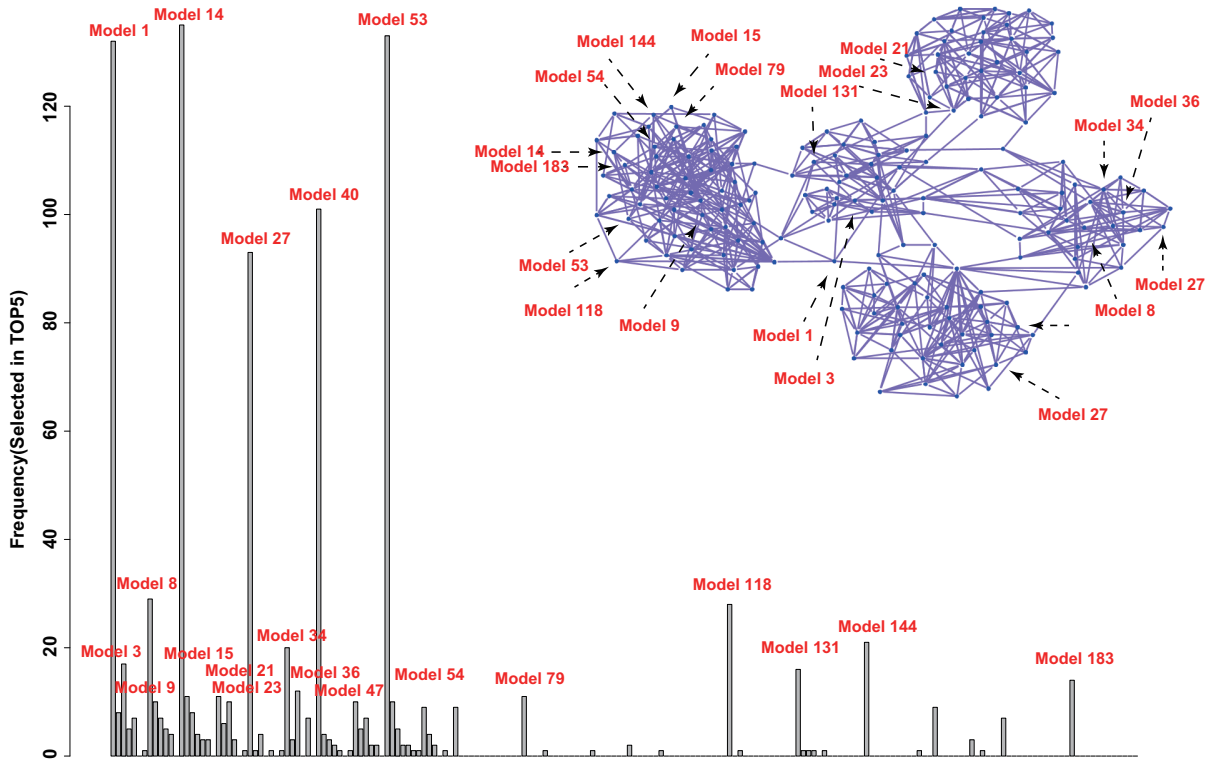

Figure 7.10: Top five candidate simulation models selected by the proposed method. This illustrates a histogram of frequencies of candidate simulation models selected as top five for the Test-Genes by the proposed method. The right top graph is the same as in Fig. 7.5. Highly selected candidates are indicated their positions. This result is obtained from the right-most case (42 h) in Fig. 7.8.

and then the residual 134 simulation models (five genes, 'Il1a', 'Il1b', 'Ccl2', 'Ccl4', and 'Vcam1', did not have templates) were selected from the others. For example, 'A2m' was previously assigned to Model 14 (one of the template pathways), in which expression is only activated by 'DR_N'. However, since the observational data of 'A2m' has a profile representing both the activated and repressed forms, our algorithm selected Model 870, in which mRNA is directly activated by 'DR_N', directly repressed by 'DR', and indirectly repressed by 'BS'. The formulation processes of the best candidate models for these 142 genes are concluded in Tables 7.2-7.4. Additionally, the simulation expression profiles of '$BS_{1,2,3}$' for each best simulation model were used to identify their candidate genes using the Smirnov-Grubbs test (outliers test) if intermediate biosignals were present. The outliers test was performed by calculating least-square errors between the simulation expression profiles of $BS_{1,2,3}$ and expression profiles of the observed 8,799 genes (5,162 unique genes), where the p-value for deciding outliers is 0.01. These genes, selected from among 5,162 genes, are named intermediate genes. After the procedure, we obtained the regulatory pathways and orders of intermediate genes (upstream and downstream genes) of the best simulation models for PG-Genes.

Interestingly, only 20 genes (0.39%) were selected as intermediate genes for BS_1, 21 genes (0.41%) for BS_2, and 20 genes (0.39%) for BS_3. The details of each gene are summarized in

Table 7.1: The Best Model For Each Time-Point (i).

GeneSymbol	6h	12h	18h	24h	30h	36h	42h	Selected
A2m	53	53	1167	1037	1037	1000	870	870
Abat	47	38	38	244	246	246	246	36
Abhd14b	8	215	215	215	215	215	215	215
Acsl1	255	255	228	228	228	293	293	363
Adk	27	27	27	27	1047	666	666	666
Adra1b	27	34	294	294	231	231	231	231
Ahcy	40	320	320	320	645	621	621	621
Akr1c12	40	38	293	293	293	506	1058	1058
Akr1d1	40	60	153	493	493	493	493	493
Akr7a2	307	307	307	683	683	683	683	683
Aldh3a2	40	40	40	40	40	40	40	1113
Anpep	40	856	856	856	856	856	856	856
Aqp9	632	311	439	439	790	790	790	790
Arg1	632	896	896	948	623	623	623	595
Asnsd1	1	899	318	903	903	903	903	903
Atic	1	1	679	679	679	679	679	679
Cacna1d	315	315	315	315	315	315	57	57
Cald1	38	233	707	707	707	707	707	707
Camk1	40	8	8	8	8	8	8	34
Ccdc56	10	319	319	320	320	320	320	320
Ccl2	27	27	256	256	256	256	1122	1122
Ccl4	14	14	14	14	14	14	14	14
Ccng1	53	53	53	53	53	53	53	53
Cox6a1	27	55	229	229	229	229	229	229
Cps1	40	731	731	731	731	738	738	738
Csda	273	990	990	990	990	990	990	990
Ctsh	27	34	34	34	34	34	34	34
Cxxc5	38	660	660	660	660	660	660	660
Cyb5b	27	27	844	909	909	909	909	909
Cyb5r3	739	739	739	36	766	766	766	766
Cyc1	40	633	893	685	685	685	685	308
Cyp2a1	27	27	27	27	27	27	27	27
Cyp2a2	268	268	268	268	36	426	426	426
Cyp2b3	60	259	387	322	322	322	322	1026
Cyp4f4	64	12	12	12	1123	1123	1123	1123
Dio1	263	530	530	530	530	530	530	530
Dpys	40	309	47	47	47	1074	948	621
Enpp3	27	785	785	229	229	426	426	426
Ephx2	40	40	292	292	71	71	71	71
Epn2	53	53	53	801	801	801	801	801
Fbp2	1	1	1	1	1	161	98	98
Fdft1	27	27	1057	997	1127	1127	602	604
Fkbp4	34	34	34	34	34	746	744	744
Fn1	27	27	307	307	307	307	307	307
Fn3k	27	23	23	616	616	696	696	1011
Gcgr	27	27	27	33	293	298	298	816
Gchfr	47	203	203	8	8	8	8	8
Gclm	27	294	233	233	608	608	608	608
Grhpr	40	645	645	645	645	645	645	645
Gstk1	27	27	27	27	27	27	113	113

Table 7.5. The expression profiles of these genes are relatively simple and smooth, as shown in Fig. 7.11. Their biological functions were investigated by FatiGo [4], for example, biological

Table 7.2: The Best Model For Each Time-Point (ii).

GeneSymbol	6h	12h	18h	24h	30h	36h	42h	Selected
Haao	40	25	278	278	278	38	181	51
Hagh	1	649	649	649	1060	774	774	774
Hmbs	27	27	27	27	27	27	27	27
Homer2	27	27	27	27	965	662	610	610
Hpn	229	229	595	603	1115	1115	1115	1115
Hsd17b12	27	27	857	857	1130	1130	1130	1130
Hsd17b13	27	618	839	839	138	138	138	138
Hsd3b7	27	27	27	27	27	27	27	27
Idh1	27	27	617	617	617	1081	769	769
Ifrd1	54	258	258	258	318	184	184	184
Igfals	40	40	40	40	40	40	40	40
Il1a	1	1	36	10	10	10	10	10
Il1b	14	14	14	14	14	14	14	14
Itga7	100	100	100	100	100	100	623	38
Kmo	8	49	49	49	967	967	967	967
Kras	53	53	53	53	67	67	67	67
Lgr4	60	60	114	94	94	94	484	484
Lipa	27	27	787	792	592	592	592	592
LOC100360011	27	294	296	296	296	296	296	296
LOC689574	645	593	593	894	894	894	894	894
Lyve1	14	14	14	14	14	14	1058	1058
Maob	255	203	203	203	8	296	293	293
Map1lc3b	14	14	14	14	14	14	14	14
Mapk9	8	8	8	8	528	528	608	606
Marc2	27	27	27	858	650	650	650	650
Mgat1	27	27	957	957	315	315	315	605
Mgp	53	638	639	906	769	641	641	641
Ncl	53	53	53	53	53	53	1160	1160
Ndufc1	27	27	27	52	52	52	52	52
Ndufv3	27	27	27	27	27	27	166	166
Nfia	27	8	853	853	853	853	853	853
Nfyb	27	27	27	910	908	1013	1013	1013
Nme1	53	316	316	316	316	316	316	316
Nolc1	14	41	305	305	773	780	780	1055
Npm1	53	53	53	53	53	671	608	608
Nr1h3	40	40	40	40	40	440	440	440
Nr1h4	34	216	216	933	933	933	933	933
Nudt4	53	15	15	15	193	193	193	193
Odc1	1	14	14	14	14	14	14	14
Oplah	27	203	203	203	203	203	203	203
Otc	40	40	40	40	40	40	40	40
Pcyt2	40	40	40	40	609	609	667	674
Pde3b	40	40	21	21	21	21	21	21
Pigr	27	250	250	250	250	250	250	164
Pla2g16	27	229	21	855	926	796	605	605
Pold3	164	164	164	892	892	644	644	644
Ppdpf	29	31	31	31	31	161	567	179
Ppil3	60	60	125	125	125	125	125	125
Ppox	40	309	309	874	874	874	874	874
Prkaca	40	8	8	892	879	879	619	619

functions of response to corticosteroid stimulus (GO:0031960) ($p = 1.63 \times 10^{-4}$), inflammatory response (GO:0006954) ($p = 1.98 \times 10^{-4}$), regulation of cell proliferation (GO:0042127) ($p =$

Table 7.3: The Best Model For Each Time-Point (iii).

GeneSymbol	6h	12h	18h	24h	30h	36h	42h	Selected
Prmt7	54	314	314	145	186	123	123	123
Qdpr	27	778	1116	1116	1116	1116	1116	1116
Rab8a	40	27	27	267	267	267	267	267
Raf1	40	40	40	421	421	421	421	493
Rb1	40	40	608	629	844	844	844	844
Rdh10	27	27	27	27	425	425	425	362
RGD1565496	14	14	14	14	14	14	14	14
RGD620382	27	42	42	42	309	309	309	309
Rgn	571	571	571	571	606	606	606	606
Rpp21	59	57	57	57	905	863	863	863
Scly	64	179	904	974	961	961	961	961
Scp2	27	216	216	216	216	216	216	216
Sdc1	40	40	40	8	8	8	8	597
Slc10a1	317	408	1126	621	621	1162	1162	1089
Slc12a7	27	27	27	27	27	27	27	360
Slc25a10	27	27	27	27	27	27	27	27
Slc25a11	27	27	27	27	27	27	27	27
Slc25a23	8	296	296	34	597	597	597	597
Slc30a1	27	10	10	541	541	541	541	541
Slc37a4	27	35	35	35	35	35	35	35
Slc6a13	610	974	517	517	517	803	803	803
Slco2a1	1	29	38	298	233	233	233	233
Smn1	53	54	249	279	409	409	409	409
Sord	228	488	34	34	51	34	634	634
Sri	27	259	127	153	226	226	226	857
Sult1a1	48	626	626	626	626	637	615	615
Sult1b1	27	27	27	27	27	27	373	373
Sult1e1	27	27	320	320	320	16	12	12
Sult2a1	7	7	7	7	616	616	616	616
Sult2a2	47	51	51	246	246	34	34	34
Sult2al1	27	23	23	23	23	23	23	23
Tmem53	47	294	294	294	506	493	493	493
Tnk2	1	53	53	53	53	320	60	60
Tomm20	53	210	210	210	210	210	210	210
Trmt6	14	316	286	286	286	286	249	249
Tyro3	47	826	822	822	970	970	970	42
Ugt2b	613	1153	102	102	102	102	102	102
Ugt2b1	40	285	298	298	298	298	298	298
Uox	1	1	269	269	269	6	6	6
Vars	14	380	380	380	380	380	380	380
Vcam1	14	14	14	14	14	14	14	14
Zadh2	198	601	289	597	727	727	727	727

1.85×10^{-5}), and negative regulation of metabolic process (GO:0009892) ($p = 1.28 \times 10^{-5}$) were ascribed to these intermediate genes. The functions of response to corticosteroid stimulus and inflammatory response are clearly related to the corticosteroid. Furthermore, in a previous study [41], the relationships between corticosteroid and metabolic pathways were investigated, and intermediate genes with functions related to metabolic process were also obtained. Their regulating genes can be candidate genes related to the same biological functions.

Using the procedure shown in Fig. 7.12, the simulation models obtained were integrated to construct a biological pathway model, starting from 'DR_N' and 'DR', as illustrated in Fig. 7.13.

Table 7.4: The list of genes selected as intermediate genes.

Gene Symbol	$BS_1/BS_2/BS_3$	Entrez Gene Name
A2M	○/○/○	alpha-2-macroglobulin
ATP1B1	○/○/-	ATPase, Na+/K+ transporting, beta 1 polypeptide
ATP2B2	-/○/○	ATPase, Ca++ transporting, plasma membrane 2
Bmyc	○/○/-	brain expressed myelocytomatosis oncogene
Cebpd	○/○/○	CCAAT/enhancer binding protein (C/EBP), delta
CITED2	○/○/○	Cbp/p300-interacting transactivator, with Glu/Asp-rich carboxy-terminal domain, 2
Cxcl2	○/○/○	chemokine (C-X-C motif) ligand 2
DIO3	○/○/○	deiodinase, iodothyronine, type III
FABP4	○/○/○	fatty acid binding protein 4, adipocyte
GUCY2C	○/○/○	guanylate cyclase 2C (heat stable enterotoxin receptor)
IL10	○/○/○	interleukin 10
INPP4A	○/○/○	inositol polyphosphate-4-phosphatase, type I, 107kDa
ITPR1	○/-/-	inositol 1,4,5-trisphosphate receptor, type 1
Mmp8	-/-/○	matrix metallopeptidase 8
MYC	○/○/○	v-myc myelocytomatosis viral oncogene homolog
NOLC1	○/○/-	nucleolar and coiled-body phosphoprotein 1
PPP1R15A	○/○/○	protein phosphatase 1, regulatory subunit 15A
Pvr	○/○/○	poliovirus receptor
RGS1	○/○/○	regulator of G-protein signaling 1
Scd4	○/○/-	stearoyl-coenzyme A desaturase 4
SEC22A	-/○/○	SEC22 vesicle trafficking protein homolog A
Tgm1	○/○/○	transglutaminase 1 (K polypeptide epidermal type I protein-glutamine-gamma-glutamyltransferase)
VHL	○/○/○	von Hippel-Lindau tumor suppressor

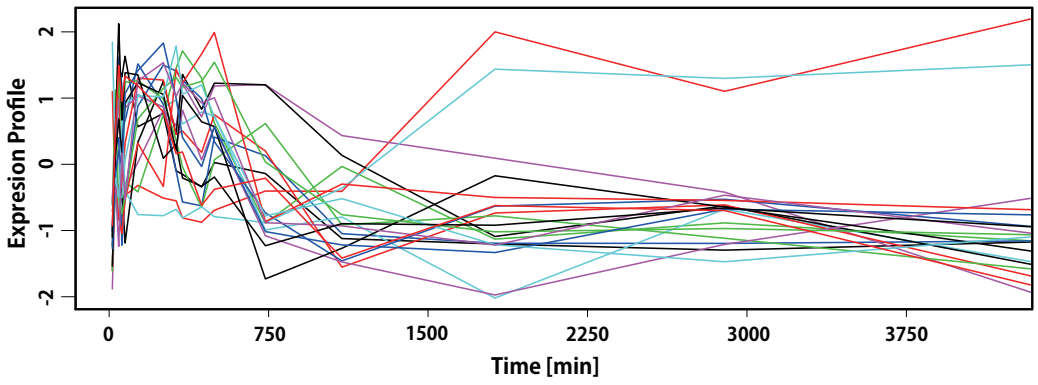

Figure 7.11: Expression profiles of genes selected as intermediate genes.

Let green and purple edges mean activation and repression, respectively. Genes that have virtually the same regulatory structures, (*i.e.*, that are regulated by the same gene) are summarized in 'Group1-15' for ease of viewing. These are illustrated as circles with red outlines and are summarized in Table 7.6. Intermediate genes are illustrated as circles with filled in red. Intermediate biosignals that are not associated with particular genes by the outliers test are described as 'BS1', 'BS2', 'BS3', 'BS12', 'BS23', and 'BS123'. The notations '+' and '-' mean activation and repression by 'DR_N', respectively. For example, 'BS12+' represents the intermediate biosignals

'BS_1' and 'BS_2', that are activated by 'DR_N'.

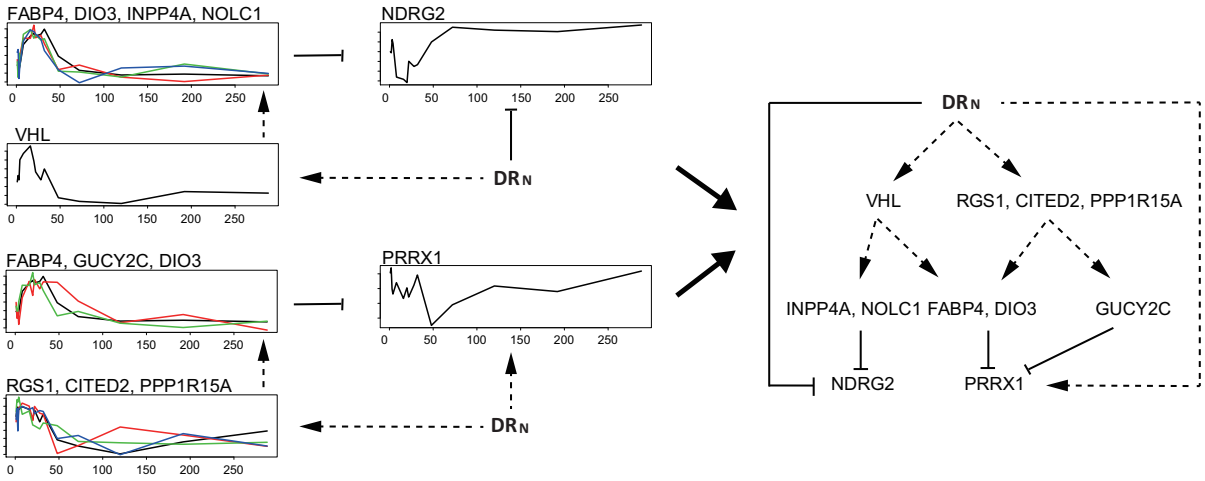

Figure 7.12: An example for integrating obtained pathways. Two pathways in the left are integrated to the gene network in the right. Arrows in this figure are the same as those in Fig. 7.6.

Table 7.5: Grouped Genes in Fig. 7.13.

Group	Genes
Group1	Grhpr,Npm1,Epn2,Hsd17b12
Group2	Ndufv3,Raf1,Fn1,Il1a
Group3	Cebpd,Cited2,Il10,Cxcl2,Rgs1,Myc,Ppp1r15a,Tgm1
Group4	Nolc1,Dio3,Fabp4,Inpp4a,A2m,Gucy2c
Group5	Adra1b,Haao,Itga7,LOC100360011,Tmem53,LOC689574,Acsl1 Abat,Slco2a1,Enpp3,Akr1d1,Slc25a23
Group6	Ifrd1,Tomm20,Trmt6,Prmt7,Smn1
Group7	Sdc1,Zadh2,Cxxc5,Aqp9,Sord,Rgn,Ppdpf,Scly,Cyb5r3,Dpys Ahcy,Cyp2b3,Sri,Hpn,Gcgr
Group8	Slc6a13,Gclm,Mapk9
Group9	Aldh3a2,Nfia,Nfyb,Prkaca,Arg1,Fn3k
Group10	Ugt2b1,Oplah,Sult2a2,Ctsh,Cox6a1,Gchfr,Dio1,Hsd17b13 RGD620382,Sult1e1
Group11	Slc30a1,Sult2al1,Scp2,Pde3b,Tnk2,Ppil3,Ccdc56
Group12	Otc,Cyp2a1,Igfals,Hsd3b7,Hmbs,Slc25a11,Slc25a10
Group13	Ccng1,Il1b,Ccl4,Vcam1,Map1lc3b,Odc1,Nme1,RGD1565496
Group14	Cyp4f4,Cald1,Akr1c12,Kmo,Lyve1,Rpp21
Group15	Mgp,Homer2,Pla2g16

7.4 Discussion

We have developed a computational method to explore simulation models that can predict observational data better than those reported in the literature. For generating candidate models, we applied the regulatory structures [97] to the template models. Instead of evaluating all of the candidates, the proposed method can selectively evaluate high-potential candidates and obtain the best model within a short span of time.

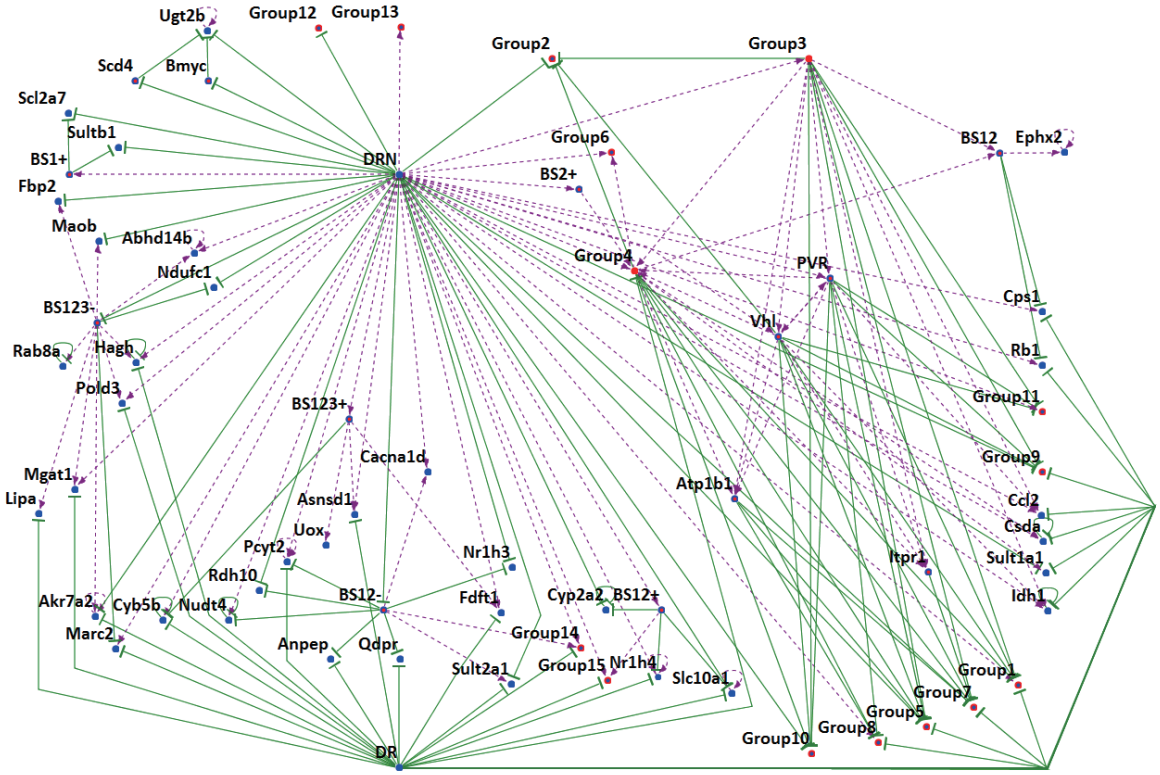

Figure 7.13: A pathway model of the best pathways for 142 focused genes. Pathways of simulation models selected as the best among All-Models are integrated into a pathway model. Genes regulated by the same genes are summarized to 'Group1-15' and indicated as circles with red outlines. These genes are summarized in Table 7.6. Intermediate genes are illustrated as circles filled in red. Thus, the red circles represent both grouped and intermediate genes. Dotted purple arrows and green arrows mean activation and repression, respectively.

The effectiveness of the proposed method was demonstrated through numerical experiments using corticosteroid pharmacogenomics reported for the rat. In comparison with the previously reported method, the proposed method achieved a performance of 80% accuracy rate while consuming less than 15% computational time than a comprehensive search [40]. The method is expected to correctly identify candidates that have higher prediction ability if the model transition rule appropriately captures the relationships among candidate models. In contrast, it requires a long time to search for the best candidate if pairs of connected simulation models do not represent almost the same simulation results. In fact, after 42 h of computation, 8.5% of the correct candidates could not even be investigated. In order to reduce computational cost, in the proposed method, the estimated parameter values of simple simulation models are used for parameter estimation of complicated simulation models. From the comparison of results, it appears that even the complicated models could be evaluated correctly within a short span of time.

In the final experiment, for 134 of 142 corticosteroid-induced genes, simulation models differ-

ent from the template models of these genes were selected as the best model for predicting the data. Furthermore, 23 genes were selected as candidates of intermediate biosignals regulating corticosteroid-induced genes. Certainly, intermediate biosignals can be other biomolecules, *e.g.*, proteins and chemical compounds, and mRNA expression profiles may not correspond to their protein expression profiles, and some of their simulation expression profiles in the selected 142 simulation models could not be represented by the 5,162 gene expression profiles. However, we focused on the fact that some of the simulation expression profiles of intermediate biosignals could match the observed gene expression profiles and that some of them have biological functions related to the corticosteroid. Thus, biological functions and related genes can potentially facilitate the investigation of corticosteroid pharmacogenomics. In addition, since some simulation models including three intermediate biosignals were selected as the best, more extended models with more than three biosignals might be selected as the best. These results demonstrate that there is room for improvement in literature-reported pathways, and that improved pathways and their details can be proposed systematically by simulation-based approaches, by assimilating biological observational data. Furthermore, although we utilized the software DA1.0 for biological simulation and the corticosteroid pharmacogenomic pathways, we believe that the procedure in this study, (i) extracting the template, (ii) creating candidates, (iii) estimating parameter values and (iv) finding the best model, is capable of applying to other models with different simulation methodologies.

Chapter 8

Conclusion

In this thesis, we studied mainly three topics in the field of systems biology, (i) the inference of GRNs using a VAR-SSM, (ii) the restoration of GRNs based on a nonlinear SSM and (iii) the exploration of candidate pathways based on a differential equation-based SSM.

The main contributions of the first topic are to establish a VAR-SSM describing GRNs and develop a network inference method with $L1$ regularization utilizing the EM-algorithm. In contrast to the previous methods, the proposed linear VAR-SSM was constructed to cover basic processes of gene regulatory systems, *i.e.*, a synthesis process, a degradation process and mutual regulations among genes. This results in enabling us to handle expression data with dynamic and steady state profiles. Since GRNs are known to have sparse structures, we developed an algorithm to infer GRNs under the constraint of $L1$ regularization to the regulatory matrix maximizing the regularized log-likelihood. For this model, we further considered two extensions; (i) adding a term representing the existence of other biomolecules, *e.g.*, drugs, that can affect gene expressions, and (ii) giving weighted regularization terms for plausible genes, *e.g.*, known to be TF candidates and established pathways. Including these extensions, the proposed method can infer the regulatory relationships among genes and other biomolecules with weighting plausible regulations. The effectiveness of the proposed method was shown using three synthetic data that were generated from the networks described by linear difference equations and hill function-based differential equations, and also the network used in a part of DREAM 4 challenge. As a real application example, we handled the microarray data in rat skeletal muscle with a stimulation of corticosteroid. Since a part of rat pharmacogenomic pathways, related genes, TF candidate genes and corticosteroid pharmacodynamics have been available, we incorporated these information to infer the GRNs with regard to rat pharmacogenomics.

In the second topic, we established a state space representation of the combinatorial transcription model, which is a simple nonlinear model representing combinatorial effects by sets of two genes, and proposed novel algorithms to infer GRNs that can best predict the data. At first, since the conditional distributions of the hidden state variables in this nonlinear SSM can be non-Gaussian forms, we applied UKF to efficiently approximate these distributions as Gaussian distributions. Then, we also developed an algorithm to restore given GRNs to be consistent with

the data based on the model by modifying the original model. As a result using two synthetic data generated from WNT5A and a yeast cell-cycle networks, the proposed method could outperform the previous regression-based method. Although the proposed method outperformed the previous method, there exists a drawback in the utilization of UKF to the nonlinear SSM. Thus, the parameter estimation procedure for the model requires to retain the first four moments of the conditional distributions of the hidden state variables. Therefore, we further proposed a novel method termed HMEnPF, which can retain the first two moments and the third and the fourth central moments through the prediction, filtering and smoothing steps, and developed an algorithm to explore the best model incorporating UKF and HMEnPF. Through the simulation studies to restore the original GRNs inferred by other well-known GRNs inference methods, the proposed method could show better performance. Moreover, the significance of HMEnPF, to retain higher moment information, could also be shown by comparing the results of the proposed algorithm to that of using UKF only.

In the last topic, we considered relatively small pathways described by differential equations within a state space representation. Utilizing pharmacogenomic pathways in rat liver cells as an application example, we first proposed a systematic way to create candidate models based on the original pathways and a procedure to evaluate their validity. By comprehensively evaluating 63 created candidate pathways for 8,799 genes, we could suggest better candidate pathways that can predict the expression profile of each gene. However, because even the evaluation of one simulation model needs a high computational cost, it is computationally intensive to handle more than several hundreds candidate models, which should be more complicated than 63 candidate models used for the previous study. To address the problem, we proposed an efficient explorative method that can find better candidates by sequentially creating plausible candidates, estimating the parameter values with prior information and evaluating the validity of the model. The proposed method imitates the way employed in the simulated tempering algorithm. Through the simulation studies, the proposed method could successfully find better candidate models that were selected by the comprehensive procedure within short span of time. Finally, for 142 corticosteroid pharmacogenomic genes that were suggested their regulatory systems by corticosteroid, we proposed the alternative candidate pathways that can better predict the observation data.

Future Direction

For further developments of data assimilation techniques, we here introduce some directions. At first, for a set of time-course observation data that are affected by some different drugs or silenced some different genes in the same type of cells, we should develop methods to infer GRNs incorporating the whole dataset in different conditions. These types of time-course data have been recently established, *e.g.*, real data in the DREAM challenges. Second, it is useful to develop methods that can further combine other pathway information such as metabolic pathways. Thus, although we dealt with pharmacogenomic pathways in which transcriptional sequences can be mainly represented by gene regulations in this thesis, there are many types of

biological systems that should be represented by the combination of GRNs, PPIs, CRNs and so on. Third, if we could efficiently estimate the parameter values and infer the regulatory relationships based on more complex nonlinear SSMs, it can contribute to the comprehensive understanding of biological systems in cells.

In this thesis, we tried the tasks in the filed of system biology mainly for gene regulatory networks and biological systems. We believe that these works can contribute to the developments of systems biology and understanding of biological systems, and further to create innovative drugs and medical treatments. Moreover, the hope is that, along with the developments of computer science and statistical theories, these studies could contribute to other scientific fields.

List of Publications by the Author

Journal Paper

1. Takanori Hasegawa, Rui Yamaguchi, Masao Nagasaki, Satoru Miyano, Seiya Imoto: Inference of Gene Regulatory Networks Incorporating Multi-Source Biological Knowledge via a State Space Model with L1 Regularization, *PLoS ONE*, vol.9(8), e105942, 2014.

2. Takanori Hasegawa, Tomoya Mori, Rui Yamaguchi, Seiya Imoto, Satoru Miyano, Tatsuya Akutsu: An Efficient Data Assimilation Schema for Restoration and Extension of Gene Regulatory Networks Using Time-course Observation Data, *Journal of Computational Biology*, vol.21(11), pp.1–14, 2014.

3. Takanori Hasegawa, Masao Nagasaki, Rui Yamaguchi, Seiya Imoto, Satoru Miyano: An efficient method of exploring simulation models by assimilating literature and biological observational data, *BioSystems*, vol.121, pp.54–66, 2014.

Conference Paper

1. Takanori Hasegawa, Rui Yamaguchi, Masao Nagasaki, Seiya Imoto, Satoru Miyano: Comprehensive pharmacogenomic pathway screening by data assimilation, In *Bioinformatics Research and Applications*, volume 6674 of *Lecture Notes in Computer Science*, pages 160-171, Springer Berlin Heidelberg, 2011.

Bibliography

[1] H. Akaike. A New Look at the Statistical Model Identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, 1974.

[2] H. Akaike. On entropy maximization principle. In *Applications of Statistics (Dayton, OH, 1976)*, pages 27–41. North-Holland, 1977.

[3] T. Akutsu, T. Tamura, and K. Horimoto. Completing networks using observed data. In *Algorithmic Learning Theory*, volume 5809 of *Lecture Notes in Computer Science*, pages 126–140. Springer Berlin Heidelberg, 2009.

[4] F. Al-Shahrour, R. Díaz-Uriarte, and J. Dopazo. FatiGO: a web tool for finding significant associations of Gene Ontology terms with groups of genes. *Bioinformatics*, 20(4):578–580, 2004.

[5] R. R. Almon, D. C. DuBois, J. Y. Jin, and W. J. Jusko. Temporal profiling of the transcriptional basis for the development of corticosteroid-induced insulin resistance in rat muscle. *Journal of Endocrinology*, 184(1):219–232, 2005.

[6] L. J. Anderson and L. S. Anderson. A monte carlo implementation of the nonlinear filtering problem to produce ensemble assimilations and forecasts. *Monthly Weather Review*, 127(12):2741–2758, 1999.

[7] H. M. S. Asif and G. Sanguinetti. Large-scale learning of combinatorial transcriptional dynamics from gene expression. *Bioinformatics*, 27(9):1277–1283, 2011.

[8] Y. Bard. *Nonlinear parameter estimation*. New York: Academic Press, 1974.

[9] M. Barenco, D. Tomescu, D. Brewer, R. Callard, J. Stark, and M. Hubank. Ranked prediction of p53 targets using hidden variable dynamic modeling. *Genome Biology*, 7(3):R25+, 2006.

[10] B. Barzel and A.-L. L. Barabási. Network link prediction by global silencing of indirect correlations. *Nature biotechnology*, 31(8):720–725, 2013.

[11] G. Batt, M. Page, I. Cantone, G. Goessler, P. Monteiro, and H. de Jong. Efficient parameter search for qualitative models of regulatory networks using symbolic model checking. *Bioinformatics*, 26(18):i603–i610, 2010.

[12] M. J. Beal, F. Falciani, Z. Ghahramani, C. Rangel, and D. L. Wild. A bayesian approach to reconstructing genetic regulatory networks with hidden factors. *Bioinformatics*, 21:349–356, 2005.

[13] J. Cao and H. Zhao. Estimating dynamic models for gene regulation networks. *Bioinformatics*, 24(14):1619–1624, 2008.

[14] K.-C. Chen, T.-Y. Wang, H.-H. Tseng, C.-Y. F. Huang, and C.-Y. Kao. A stochastic differential equation model for quantifying transcriptional regulatory network in saccharomyces cerevisiae. *Bioinformatics*, 21(12):2883–2890, 2005.

[15] J. M. Cherry, E. L. Hong, C. Amundsen, R. Balakrishnan, G. Binkley, E. T. Chan, K. R. Christie, M. C. Costanzo, S. S. Dwight, S. R. Engel, D. G. Fisk, J. E. Hirschman, B. C. Hitz, K. Karra, C. J. Krieger, S. R. Miyasato, R. S. Nash, J. Park, M. S. Skrzypek, M. Simison, S. Weng, and E. D. Wong. Saccharomyces genome database: the genomics resource of budding yeast. *Nucleic Acids Research*, 40(Database-Issue):700–705, 2012.

[16] S.-M. Chow, E. Ferrer, and J. R. Nesselroade. An unscented kalman filter approach to the estimation of nonlinear dynamical systems models. *Multivariate Behavioral Research*, 42(2):283–321, 2007.

[17] A. P. Davis, T. C. Wiegers, R. J. Johnson, J. M. Lay, K. Lennon-Hopkins, C. Saraceni-Richards, D. Sciaky, C. G. Murphy, and C. J. Mattingly. Text mining effectively scores and ranks the literature for improving chemical-gene-disease curation at the comparative toxicogenomics database. *PLoS ONE*, 8(4):e58201, 2013.

[18] H. de Jong. Modeling and simulation of genetic regulatory systems: A literature review. *Journal of Computational Biology*, 9(1):67–103, 2002.

[19] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B*, 39:1–38, 1977.

[20] T. G. do Rego, H. G. Roider, F. A. T. de Carvalho, and I. G. Costa. Inferring epigenetic and transcriptional regulation during blood cell development with a mixture of sparse linear models. *Bioinformatics*, 28(18):2297–2303, 2012.

[21] C.-Y. Dong, D. Shin, S. Joo, Y. Nam, and K.-H. Cho. Identification of feedback loops in neural networks based on multi-step granger causality. *Bioinformatics*, 28(16):2146–2153, 2012.

[22] F. Eduati, J. De Las Rivas, B. Di Camillo, G. Toffolo, and J. Saez-Rodriguez. Integrating literature-constrained and data-driven inference of signalling networks. *Bioinformatics*, 28(18):2311–2317, 2012.

[23] B. Efron, T. Hastie, L. Johnstone, and R. Tibshirani. Least angle regression. *Annals of Statistics*, 32:407–499, 2004.

[24] M. B. Elowitz and S. Leibler. A synthetic oscillatory network of transcriptional regulators. *Nature*, 403(6767):335–338, 2000.

[25] G. Evensen. Sequential data assimilation with a nonlinear quasi-geostrophic model using monte carlo methods to forecast error statistics. *Journal of Geophysical Research*, 99:10143–10162, 1994.

[26] 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*, 5(1):e8, 2007.

[27] S. Feizi, D. Marbach, M. Medard, and M. Kellis. Network deconvolution as a general method to distinguish direct dependencies in networks. *Nature Biotechnology*, 31(8):726–733, 2013.

[28] D. Foti, R. Iuliano, E. Chiefari, and A. Brunetti. A nucleoprotein complex containing sp1, c/ebpb, and hmgi-y controls human insulin receptor gene transcription. *Molecular and Cellular Biology*, 23(8):2720–2732, 2003.

[29] J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

[30] N. Friedman, M. Linial, I. Nachman, and D. Pe'er. Using Bayesian networks to analyze expression data. *Journal of Computational Biology*, 7(3):601–620, 2000.

[31] A. Fujita, J. R. Sato, H. M. Garay-Malpartida, P. A. Morettin, M. C. Sogayar, and C. E. Ferreira. Time-varying modeling of gene expression regulatory networks using the wavelet dynamic vector autoregressive method. *Bioinformatics*, 23(13):1623–1630, 2007.

[32] A. Fujita, J. R. Sato, H. M. Garay-Malpartida, M. C. Sogayar, C. E. Ferreira, and S. Miyano. Modeling nonlinear gene regulatory networks from time series gene expression data. *Journal of Bioinformatics and Computational Biology*, 6(5):961–979, 2008.

[33] A. Fujita, J. R. Sato, H. M. Garay-Malpartida, R. Yamaguchi, S. Miyano, M. C. Sogayar, and C. E. Ferreira. Modeling gene expression regulatory networks with the sparse vector autoregressive model. *BMC systems biology*, 1:39, 2007.

[34] A. Fujita, P. Severino, J. R. Sato, and S. Miyano. Granger causality in systems biology: Modeling gene networks in time series microarray data using vector autoregressive models. In *Advances in Bioinformatics and Computational Biology*, volume 6268 of *Lecture Notes in Computer Science*, pages 13–24. Springer Berlin Heidelberg, 2010.

[35] C. J. Geyer and E. A. Thompson. Annealing Markov Chain Monte Carlo with Applications to Ancestral Inference. *Journal of the American Statistical Association*, 90(431):909–920, 1995.

[36] N. Gordon, D. Salmond, and A. Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. *IEEE Proceedings F, Radar and Signal Processing*, 140(2):107–113, 1993.

[37] A. Greenfield, C. Hafemeister, and R. Bonneau. Robust data-driven incorporation of prior knowledge into the inference of dynamic regulatory networks. *Bioinformatics*, 29(8):1060–1067, 2013.

[38] U. Hahn, K. B. Cohen, Y. Garten, and N. H. Shah. Mining the pharmacogenomics literature-a survey of the state of the art. *Briefings in Bioinformatics*, 13(4):460–494, 2012.

[39] T. Hasegawa, M. Nagasaki, R. Yamaguchi, S. Imoto, and S. Miyano. An efficient method of exploring simulation models by assimilating literature and biological observational data. *Biosystems*, 121:54–66, 2014.

[40] T. Hasegawa, R. Yamaguchi, M. Nagasaki, S. Imoto, and S. Miyano. Comprehensive pharmacogenomic pathway screening by data assimilation. In *Bioinformatics Research and Applications*, volume 6674 of *Lecture Notes in Computer Science*, pages 160–171. Springer Berlin Heidelberg, 2011.

[41] A. Hazra, D. C. DuBois, R. R. Almon, G. H. Snyder, and W. J. Jusko. Pharmacodynamic modeling of acute and chronic effects of methylprednisolone on hepatic urea cycle genes in rats. *Gene Regulation and Systems Biology*, 2:1–19, 2008.

[42] J. Henderson and G. Michailidis. Network reconstruction using nonparametric additive ode models. *PLoS ONE*, 9(4):e94003, 2014.

[43] O. Hirose, R. Yoshida, S. Imoto, R. Yamaguchi, T. Higuchi, D. S. Charnock-Jones, C. Print, and S. Miyano. Statistical inference of transcriptional module-based gene networks from time course gene expression profiles by using state space models. *Bioinformatics*, 24:932–942, 2008.

[44] S. Hooshangi, S. Thiberge, and R. Weiss. Ultrasensitivity and noise propagation in a synthetic transcriptional cascade. *Proceedings of the National Academy of Sciences of the United States of America*, 102(10):3581–3586, 2005.

[45] S. Imoto, T. Goto, and S. Miyano. Estimation of genetic networks and functional structures between genes by using bayesian networks and nonparametric regression. In *Pacific Symposium on Biocomputing*, pages 175–186, 2002.

[46] B. Jia and X. Wang. Regularized em algorithm for sparse parameter estimation in nonlinear dynamic systems with application to gene regulatory network inference. *EURASIP Journal on Bioinformatics and Systems Biology*, 2014(1):5, 2014.

[47] J. Y. Jin, R. R. Almon, D. C. DuBois, and W. J. Jusko. Modeling of corticosteroid pharmacogenomics in rat liver using gene microarrays. *Journal of Pharmacology and Experimental Therapeutics*, 307(1):93–109, 2003.

[48] S. Julier. The scaled unscented transformation. In *Proceedings of American Control Conference 2002*, volume 6, pages 4555–4559, 2002.

[49] S. Julier and J. Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422, 2004.

[50] S. Julier, J. Uhlmann, and H. Durrant-Whyte. A new method for the nonlinear transformation of means and covariances in filters and estimators. *Automatic Control, IEEE Transactions on*, 45(3):477–482, 2000.

[51] S. J. Julier and J. K. Uhlmann. A new extension of the kalman filter to nonlinear systems. In *Proceedings of AeroSense: The 11th International Symposium on Aerospace/Defense Sensing, Simulations and Controls*, pages 182–193, 1997.

[52] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[53] M. Kanehisa, S. Goto, Y. Sato, M. Furumichi, and M. Tanabe. Kegg for integration and interpretation of large-scale molecular data sets. *Nucleic Acids Research*, 40(D1):D109–D114, 2012.

[54] S. Kim, S. Imoto, and S. Miyano. Dynamic bayesian network and nonparametric regression for nonlinear modeling of gene networks from time series gene expression data. *Biosystems*, 75(1-3):57–65, 2004.

[55] S. Kim, H. Li, E. R. Dougherty, N. Cao, Y. Chen, M. Bittner, and E. B. Suh. Can markov chain models mimic biological regulation? *Journal of Biological Systems*, 10(4):337–357, 2002.

[56] S. Y. Kim, S. Imoto, and S. Miyano. Inferring gene networks from time series microarray data using dynamic bayesian networks. *Briefings in Bioinformatics*, 4(3):228–235, 2003.

[57] G. Kitagawa. Monte Carlo Filter and Smoother for Non-Gaussian Nonlinear State Space Models. *Journal of Computational and Graphical Statistics*, 5(1):1–25, 1996.

[58] G. Kitagawa. A self-organizing state-space model. *Journal of the American Statistical Association*, 93(443):1203–1215, 1998.

[59] C. H. H. Koh, M. Nagasaki, A. Saito, L. Wong, and S. Miyano. DA 1.0: parameter estimation of biological pathways using data assimilation approach. *Bioinformatics*, 26(14):1794–1796, 2010.

[60] K. Kojima, R. Yamaguchi, S. Imoto, M. Yamauchi, M. Nagasaki, R. Yoshida, T. Shimamura, K. Ueno, T. Higuchi, N. Gotoh, and et al. A state space representation of var models with sparse learning for dynamic gene networks. *Genome Informatics*, 22:56–68, 2009.

[61] L. Kuepfer, M. Peter, U. Sauer, and J. Stelling. Ensemble modeling for analysis of cell signaling dynamics. *Nature Biotechnology*, 25(9):1001–1006, 2007.

[62] N. D. Lawrence, G. Sanguinetti, and M. Rattray. Modelling transcriptional regulation using gaussian processes. In *Advances in Neural Information Processing Systems 19*, pages 785–792. MIT Press, 2006.

[63] S. Lébre. Inferring dynamic genetic networks with low order independencies. *Statistical Applications in Genetics and Molecular Biology*, 8(1):1–38, 2009.

[64] H. Li, K. Zhang, and T. Jiang. The regularized em algorithm. In *Proceedings of the 20th National Conference on Artificial Intelligence - Volume 2*, AAAI'05, pages 807–812. AAAI Press, 2005.

[65] Y. Li, V. A. Protopopescu, and A. Gorin. Accelerated simulated tempering. *Physics Letters A*, 328(4-5):274–283, 2004.

[66] G. Lillacci and M. Khammash. Parameter estimation and model selection in computational biology. *PLoS Computational Biology*, 6(3):e1000696, 2010.

[67] X. Liu and M. Niranjan. State and parameter estimation of the heat shock response system using kalman and particle filters. *Bioinformatics*, 28(11):1501–1507, 2012.

[68] R. Mahdi, A. S. Madduri, G. Wang, Y. Strulovici-Barel, J. Salit, N. R. Hackett, R. G. Crystal, and J. G. Mezey. Empirical bayes conditional independence graphs for regulatory network recovery. *Bioinformatics*, 28(15):2029–2036, 2012.

[69] D. Marbach, T. Schaffter, C. Mattiussi, and D. Floreano. Generating Realistic In Silico Gene Networks for Performance Assessment of Reverse Engineering Methods. *Journal of Computational Biology*, 16(2):229–239, 2009.

[70] A. Margolin, I. Nemenman, K. Basso, C. Wiggins, G. Stolovitzky, R. Favera, and A. Califano. ARACNE: An Algorithm for the Reconstruction of Gene Regulatory Networks in a Mammalian Cellular Context. *BMC Bioinformatics*, 7(Suppl 1):S7+, 2006.

[71] E. Marinari and G. Parisi. Simulated tempering: A new monte carlo scheme. *Europhysics Letters*, 19(6):451, 1992.

[72] H. Matsuno, S. T. Inouye, Y. Okitsu, and Y. Fujii. A new regulatory interaction suggested by simulations for circadian genetic control mechanism in mammals. *Journal of Bioinformatics and Computational Biology*, 4(1):139–153, 2006.

[73] H. Matsuno, M. Nagasaki, and S. Miyano. Hybrid petri net based modeling for biological pathway simulation. *Natural Computing*, 10:1099–1120, 2011.

[74] P. S. Maybeck. *Stochastic models, estimation and control. Volume I.* Academic Press, 1979.

[75] P. Meyer, K. Kontos, F. Lafitte, and G. Bontempi. Information-theoretic inference of large transcriptional regulatory networks. *EURASIP Journal on Bioinformatics and Systems Biology*, 2007(1):79879, 2007.

[76] S. Murtuza Baker, C. H. Poskar, F. Schreiber, and B. H. Junker. An improved constraint filtering technique for inferring hidden states and parameters of a biological model. *Bioinformatics*, 29(8):1052–1059, 2013.

[77] M. Nagasaki, R. Yamaguchi, R. Yoshida, S. Imoto, A. Doi, Y. Tamada, H. Matsuno, S. Miyano, and T. Higuchi. Genomic data assimilation for estimating hybrid functional petri net from time-course gene expression data. *Genome Informatics*, 17(1):46–61, 2006.

[78] N. Nakajima, T. Tamura, Y. Yamanishi, K. Horimoto, and T. Akutsu. Network completion using dynamic programming and least-squares fitting. *ScientificWorldJournal*, 2012, 2012.

[79] K. Nakamura, R. Yoshida, M. Nagasaki, S. Miyano, and T. Higuchi. Parameter estimation of *in silico* biological pathways with particle filtering toward a petascale computing. In *Proceedings of Pacific Symposium on Biocomputing 2009*, volume 14, pages 227–238, 2009.

[80] R. Opgen-Rhein and K. Strimmer. From correlation to causation networks: a simple approximate learning algorithm and its application to high-dimensional plant gene expression data. *BMC Systems Biology*, 1(1):37, 2007.

[81] M. Opper and G. Sanguinetti. Learning combinatorial transcriptional dynamics from gene expression data. *Bioinformatics*, 26(13):1623–1629, 2010.

[82] D. T. Pham. Stochastic methods for sequential data assimilation in strongly nonlinear systems. *Monthly Weather Review*, 129(5):1194–1207, 2001.

[83] M. Quach, N. Brunel, and F. d'Alche Buc. Estimating parameters and hidden variables in non-linear state-space models based on odes for biological networks inference. *Bioinformatics*, 23(23):3209–3216, 2007.

[84] R. Ramakrishnan, D. DuBois, R. Almon, N. Pyszczynski, and W. Jusko. Fifth-generation model for corticosteroid pharmacodynamics: Application to steady-state receptor down-regulation and enzyme induction patterns during seven-day continuous infusion of methyl-

prednisolone in rats. *Journal of Pharmacokinetics and Pharmacodynamics*, 29(1):1–24, 2002.

[85] J. O. Ramsay, G. Hooker, D. Campbell, and J. Cao. Parameter estimation for differential equations: a generalized smoothing approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(5):741–796, 2007.

[86] C. Rangel, J. Angus, Z. Ghahramani, M. Lioumi, E. Sotheran, A. Gaiba, D. L. Wild, and F. Falciani. Modeling t-cell activation using gene expression profiling and state-space models. *Bioinformatics*, 20:1361–1372, 2004.

[87] S. Rogers, R. Khanin, and M. Girolami. Bayesian model-based inference of transcription factor activity. *BMC Bioinformatics*, 8(Suppl), 2007.

[88] C. Sabatti and G. M. James. Bayesian sparse hidden components analysis for transcription regulation networks. *Bioinformatics*, 22(6):739–746, 2006.

[89] G. Sanguinetti, A. Ruttor, M. Opper, and C. Archambeau. Switching regulatory models of cellular stress response. *Bioinformatics*, 25(10):1280–1286, 2009.

[90] S. Sarkka. Unscented rauch–tung–striebel smoother. *Automatic Control, IEEE Transactions on*, 53(3):845–849, 2008.

[91] M. A. Savageau. Biochemical systems analysis: II. The steady-state solutions for an n-pool system using a power-law approximation. *Journal of Theoretical Biology*, 25(3):370–379, 1969.

[92] M. A. Savageau and E. O. Voit. Recasting nonlinear differential equations as s-systems: a canonical nonlinear form. *Mathematical Biosciences*, 87(1):83–115, 1987.

[93] J. Schaber, M. Flöttmann, J. Li, C.-F. Tiger, S. Hohmann, and E. Klipp. Automated ensemble modeling with modelmage: Analyzing feedback mechanisms in the sho1 branch of the hog pathway. *PLoS ONE*, 6(3):e14791, 2011.

[94] J. Schäfer and K. Strimmer. An empirical bayes approach to inferring large-scale gene association networks. *Bioinformatics*, 21(6):754–764, 2005.

[95] T. Schaffter, D. Marbach, and D. Floreano. Genenetweaver: in silico benchmark generation and performance profiling of network inference methods. *Bioinformatics*, 27(16):2263–2270, 2011.

[96] G. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464, 1978.

[97] S. S. Shen-Orr, R. Milo, S. Mangan, and U. Alon. Network motifs in the transcriptional regulation network of escherichia coli. *Nature Genetics*, 31:1061–4036, 2002.

[98] T. Shimamura, S. Imoto, R. Yamaguchi, A. Fujita, M. Nagasaki, and S. Miyano. Recursive regularization for inferring gene networks from time-course gene expression profiles. *BMC systems biology*, 3:41, 2009.

[99] T. Shimamura, S. Imoto, R. Yamaguchi, M. Nagasaki, and S. Miyano. Inferring dynamic gene networks under varying conditions for transcriptomic network comparison. *Bioinformatics*, 26:1064–1072, 2010.

[100] N. Shimizu, N. Yoshikawa, N. Ito, T. Maruyama, Y. Suzuki, S.-I. Takeda, J. Nakae, Y. Tagata, S. Nishitani, K. Takehana, M. Sano, K. Fukuda, M. Suematsu, C. Morimoto, and H. Tanaka. Crosstalk between Glucocorticoid Receptor and Nutritional Sensor mTOR in Skeletal Muscle. *Cell metabolism*, 13(2):170–182, 2011.

[101] R. H. Shumway and D. S. Stoffer. An approach to time series smoothing and forecasting using the em algorithm. *Journal of Time Series Analysis*, 3(4):253–264, 1982.

[102] R. H. Shumway and D. S. Stoffer. *Time Series Analysis and Its Applications: With R Examples (Springer Texts in Statistics)*. Springer, 2nd edition, 2006.

[103] P. T. Spellman, G. Sherlock, M. Q. Zhang, V. R. Iyer, K. Anders, M. B. Eisen, P. O. Brown, D. Botstein, and B. Futcher. Comprehensive identification of cell cycle-regulated genes of the yeast saccharomyces cerevisiae by microarray hybridization. *Molecular Biology of the Cell*, 9(12):3273–3297, 1998.

[104] N. Städler and P. Bühlmann. Missing values: sparse inverse covariance estimation and an extension to sparse regression. *Statistics and Computing*, 22(1):219–235, 2012.

[105] X. Sun, L. Jin, and M. Xiong. Extended Kalman Filter for Estimation of Parameters in Nonlinear State-Space Models of Biochemical Networks. *PLoS ONE*, 3(11):e3758+, 2008.

[106] Y.-N. Sun, D. DuBois, R. Almon, and W. Jusko. Fourth-generation model for corticosteroid pharmacodynamics: A model for methylprednisolone effects on receptor/gene-mediated glucocorticoid receptor down-regulation and tyrosine aminotransferase induction in rat liver. *Journal of Pharmacokinetics and Biopharmaceutics*, 26(3):289–317, 1998.

[107] Y. Tamada, R. Yamaguchi, S. Imoto, O. Hirose, R. Yoshida, M. Nagasaki, and S. Miyano. Sign-ssm: open source parallel software for estimating gene networks with state space models. *Bioinformatics*, 27(8):1172–1173, 2011.

[108] S. Tasaki, M. Nagasaki, M. Oyama, H. Hata, R. Ueno, Kazuko Yoshida, T. Higuchi, S. Sugano, and S. Miyano. Modeling and estimation of dynamic egfr pathway by data assimilation approach using time series proteomic data. *Genome Informatics*, 17(2):226–238, 2006.

[109] Y. Tian, B. Zhang, E. Hoffman, R. Clarke, Z. Zhang, I.-M. Shih, J. Xuan, D. Herrington, and Y. Wang. Knowledge-fused differential dependency network models for detecting significant rewiring in biological networks. *BMC Systems Biology*, 8(1), 2014.

[110] W. Wang, J. M. Cherry, Y. Nochomovitz, E. Jolly, D. Botstein, and H. Li. Inference of combinatorial regulation in yeast transcriptional networks: A case study of sporulation. *Proceedings of the National Academy of Sciences of the United States of America*, 102(6):1998–2003, 2005.

[111] Y. Watanabe, S. Seno, Y. Takenaka, and H. Matsuda. An estimation method for inference of gene regulatory net-work using bayesian network with uniting of partial problems. *BMC Genomics*, 13(S-1):S12, 2012.

[112] R. Yamaguchi and T. Higuchi. State space approach with the maximum likelihood principle to identify the system generating time-course gene expression data of yeast. *International Journal of Data Mining and Bioinformatics*, 1:77–87, 2006.

[113] R. Yamaguchi, S. Imoto, M. Yamauchi, M. Nagasaki, R. Yoshida, T. Shimamura, Y. Hatanaka, K. Ueno, T. Higuchi, N. Gotoh, and S. Miyano. Predicting difference in gene regulatory systems by state space models. *Genome Informatics*, 21:101–113, 2008.

[114] R. Yamaguchi, R. Yoshida, S. Imoto, T. Higuchi, and S. Miyano. Finding module-based gene networks with state-space models - Mining high-dimensional and short time-course gene expression data. *IEEE Signal Processing Magazine*, 24(1):37–46, 2007.

[115] Z. Yao, E. P. Hoffman, S. Ghimbovschi, D. C. DuBois, R. R. Almon, and W. J. Jusko. Mathematical modeling of corticosteroid pharmacogenomics in rat muscle following acute and chronic methylprednisolone dosing. *Molecular Pharmaceutics*, 5(2):328–339, 2008.

[116] E. Yeger-Lotem, S. Sattath, N. Kashtan, S. Itzkovitz, R. Milo, R. Y. Pinter, U. Alon, and H. Margalit. Network motifs in integrated cellular networks of transcription-regulation and protein-protein interaction. *Proceedings of the National Academy of Sciences of the United States of America*, 101(16):5934–5939, 2004.

[117] R. Yoshida, M. Nagasaki, R. Yamaguchi, S. Imoto, S. Miyano, and T. Higuchi. Bayesian learning of biological pathways on genomic data assimilation. *Bioinformatics*, 24(22):2592–2601, 2008.

[118] W. Young, A. Raftery, and K. Yeung. Fast Bayesian inference for gene regulatory networks using ScanBMA. *BMC Systems Biology*, 8(1):47+, 2014.

[119] N. Yukinawa, J. Yoshimoto, S. Oba, and S. Ishii. Modeling gene expression dynamics based on a linear dynamical system model. In *Proceedings of 2004 International Symposium on Nonlinear Theory and its Applications*, pages 577–580, 2004.

[120] B. Zacher, K. Abnaof, S. Gade, E. Younesi, A. Tresch, and H. Fröhlich. Joint Bayesian inference of condition-specific miRNA and transcription factor activities from combined gene and microRNA expression data. *Bioinformatics*, 28(13):1714–1720, 2012.

[121] Y. Zhao and Z. Lu. Fourth-moment standardization for structural reliability assessment. *Journal of Structural Engineering*, 133(7):916–924, 2007.

[122] G. Zheng, K. Tu, Q. Yang, Y. Xiong, C. Wei, L. Xie, Y. Zhu, and Y. Li. Itfp: an integrated platform of mammalian transcription factors. *Bioinformatics*, 24(20):2416–2417, 2008.

[123] P. Zoppoli, S. Morganella, and M. Ceccarelli. Timedelay-aracne: Reverse engineering of gene networks from time-course data by an information theoretic approach. *BMC Bioinformatics*, 11(1):154, 2010.

[124] H. Zou, T. Hastie, and R. Tibshirani. On the "degrees of freedom" of the lasso. *The Annals of Statistics*, 35(5):2173–2192, 2007.