Computational Tools
for
Biological Data Analysis and Clinical Trial Design

by

David Merrell

A preliminary report submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy

(Computer Sciences)

at the

UNIVERSITY OF WISCONSIN–MADISON

2020

Contents

Introduction

Modern biological research generates a flood of data. However, extracting useful scientific insights from that data remains an open challenge. There are many kinds of data, and many questions biologists hope to answer with it.

The scenario creates a vast landscape of opportunity for computer science and statistics. My research aims to produce practical computational tools at certain points in this landscape. I have a particular interest in Bayesian methods that bring prior knowledge into data analysis. I have also stumbled into some opportunities related to the design of clinical trials. This report outlines my efforts in each of these areas.

Overview of contents. Chapters 1 and 2 have a shared theme. Both chapters present Bayesian inference tasks, applied to biological questions.

Chapter 1 falls under the category of *completed work*. It describes a Markov Chain Monte Carlo method I developed to infer signaling pathways (i.e., network structures) from phosphorylation time series data. My work in that area will appear as a proceedings paper in ECCB 2020 (Merrell and Gitter, 2020). The chapter gives a distilled presentation of that paper.

Chapter 2 describes research that is still at an early stage. It promises to be my most ambitious project. The idea is to infer patient-specific *signaling pathway activities* from *multi-omic data*, exploiting prior knowledge

in the form of *pathway structures* (the chapter defines each of these terms). Most of this work is still ahead of me—I am still exploring different probabilistic formulations. This project constitutes the bulk of my remaining dissertation work.

Chapter 3 has a distinct theme from Chapters 1 and 2. It describes an algorithm for designing adaptive randomized controlled trials. This research is partially finished. I have implemented a prototype, but still need to run some experiments and prepare a manuscript for publication. I include it in this report for completeness. It may be one of my more significant works as a PhD student.

Lastly, I include a coarse-grained **timeline** that schedules certain milestone tasks. These milestones include the completion of prototypes, the submission of manuscripts, etc. The timeline aims at a final defense and graduation in December 2021.

Chapter 1

Inferring Signaling Pathways with Probabilistic Programming

This chapter presents some of my completed work. It describes a Bayesian inference method called Sparse Signaling Pathway Sampling (SSPS). SSPS uses Markov Chain Monte Carlo to infer the structure of a signaling pathway—i.e., a network structure—from time series proteomic data.

This chapter gives a condensed explanation of SSPS. For a more thorough presentation I refer you to the full manuscript on arXiv (Merrell and Gitter, 2020). My code is also available on GitHub (https://github.com/gitter-lab/ssps). I implemented SSPS in Julia, using the Gen.jl probabilistic programming language (Cusumano-Towner et al., 2019).

1.1 Background and motivation

Signaling pathways. Cells regulate themselves via complicated biochemical processes called *signaling pathways*. You can think of a signaling pathway as a directed graph, where nodes represent proteins and edges represent the regulatory relationships between them. The graph generally contains cycles, giving rise to the complicated feedback loops that make

life possible. Signaling pathways are a crucial tool for understanding the mechanisms of disease and therapy at a cellular level.

Signaling pathway inference. Over decades of experimentation, biologists have assembled databases full of commonly-observed pathways. These pathway databases can be useful for understanding biological processes under "typical" conditions. However, the structure of a signaling pathway can change extensively under abnormal conditions—disease or pharmaceutical exposure, for example. It follows that the ability to infer customized pathway structures from condition- or patient-specific data has high clinical value.

A variety of techniques exist for inferring signaling pathways from data. They take diverse approaches including Granger causality (Shojaie and Michailidis, 2010; Carlin et al., 2017), information theory (Cheong et al., 2011; Krishnaswamy et al., 2014), logic models (Eker et al., 2002; Guziolowski et al., 2013; Gjerga et al., 2020), differential equations (Schoeberl et al., 2002; Molinelli et al., 2013; Henriques et al., 2017), non-parametric statistical tests (Zhang and Song, 2013), and probabilistic graphical models (Sachs et al., 2005) among others.

1.2 Original contributions

I present a signaling pathway inference technique called Sparse Signaling Pathway Sampling (SSPS). SSPS builds on past works that formulate signaling pathway inference as a Dynamic Bayesian Network (DBN) structure estimation problem on phosphoproteomic time course data (Werhli and Husmeier, 2007; Gregorczyk, 2010; Hill et al., 2012; Oates et al., 2014; Spencer et al., 2015).

SSPS takes a Bayesian approach, using Markov Chain Monte Carlo to estimate a posterior distribution over possible DBN structures. My pri-

mary contributions are (i) the relaxation of common restrictive modeling assumptions and (ii) a novel proposal distribution that efficiently samples sparse graphs.

The method

Probabilistic model. We first define some notation for convenience. Let G denote a directed graph with vertices V. We can equivalently represent G as a matrix Z of edge-indicator variables $z_{i,j}$. Let X denote a time series data set—a $T \times |V|$ matrix where the jth column corresponds to the jth graph vertex. As a convenient shorthand, let X_+ denote the *latest* $T-1$ timepoints in X and let X_- denote the *earliest* $T-1$ timepoints. Lastly, define $B_j \equiv X_{-,\mathrm{pa}_G(j)}$, the value of vertex j's parents at the $T-1$ earliest timepoints.

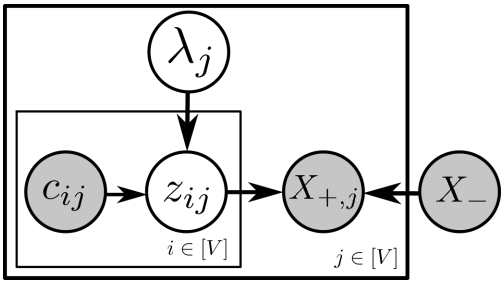
SSPS aims to infer G from X. More precisely, it seeks a posterior distribution $P(G|X) \propto P(X|G) \cdot P(G)$. SSPS uses a Gaussian DBN as its *likelihood function* $P(X|G)$. Intuitively, it assumes linear relationships between variables and their parents:

$$X_{+,j} \sim \mathcal{N}(B_j \beta_j, \sigma_j^2) \qquad \forall j \in \{1 \dots |V|\}.$$

An appropriate choice of priors over the DBN parameters β_j and σ_j^2 allows us to marginalize them away, yielding the following likelihood:

$$P(X|G) \propto \prod_{j=1}^{|V|} T^{-\frac{|\mathrm{pa}_G(j)|}{2}} \left(X_{+,j}^\top X_{+,j} - \frac{T-1}{T} X_{+,j}^\top (B_j \hat{\beta}_{ols}) \right)^{-\frac{T-1}{2}} \tag{1.1}$$

where $\hat{\beta}_{ols} = (B_j^\top B_j)^{-1} B_j^\top X_{+,j}$ is the ordinary least squares estimate of β_j.



$$\lambda_j \sim \text{Uniform}(\lambda_{\min}, \lambda_{\max}) \qquad \forall j \in \{1 \dots |V|\}$$

$$z_{ij} \mid c_{ij}, \lambda_j \sim \text{Bernoulli}\left(\frac{e^{-\lambda_j}}{e^{-c_{ij}\lambda_j} + e^{-\lambda_j}}\right) \qquad \forall i, j \in \{1 \dots |V|\}$$

$$\sigma_j^2 \propto \frac{1}{\sigma_j^2} \qquad \forall j \in \{1 \dots |V|\}$$

$$\beta_j \mid \sigma_j^2 \sim \mathcal{N}\left(0, T\sigma_j^2 (B_j^\top B_j)^{-1}\right) \qquad \forall j \in \{1 \dots |V|\}$$

$$X_{+,j} \mid B_j, \beta_j, \sigma_j^2 \sim \mathcal{N}\left(B_j \beta_j, \sigma_j^2 I\right) \qquad \forall j \in \{1 \dots |V|\}$$

Figure 1.1: The SSPS generative model. (top) Plate notation. DBN parameters β_j and σ_j^2 have been marginalized out. (bottom) Full probabilistic specification. We usually set $\lambda_{\min} \simeq 3$ and $\lambda_{\max}$=15. If $\lambda_{\min}$>0 is too small, Markov chains will occasionally be initialized with very large numbers of edges, causing computational issues. The method is insensitive to $\lambda_{\max}$ as long as it's sufficiently large. Notice the improper prior $1/\sigma_j^2$. In this specification B_j denotes $X_{-,\text{pa}_Z(j)}$; that is, the parents of vertex j depend on edge existence variables Z.

SSPS uses the following *prior distribution* $P(G) \equiv P(Z)$:

$$P(Z|C, \Lambda) = \prod_{(i,j)} \left(\frac{e^{-\lambda_j}}{e^{-c_{ij}\lambda_j} + e^{-\lambda_j}}\right)^{z_{ij}} \left(\frac{e^{-c_{ij}\lambda_j}}{e^{-c_{ij}\lambda_j} + e^{-\lambda_j}}\right)^{1-z_{ij}} \tag{1.2}$$

where $\Lambda = \{\lambda_j \mid j = 1, \dots, |V|\}$ is a set of *inverse temperature variables* and C is a matrix of real-valued *edge confidences* c_{ij}. The inverse temperature variables control the strength of the prior knowledge and are, themselves, random variables. See Figure 1.1 for the full probabilistic specification.

Inference procedure. SSPS samples from the posterior distribution using MCMC with Metropolis-Hastings updates. In each iteration, the procedure loops through the graph vertices. For each graph vertex j, the procedure (i) updates the corresponding inverse temperature variable λ_j; and then (ii) updates the vertex's *parent set* with a specially designed *parent set proposal distribution*.

The parent set proposal distribution is my most significant contribution in this work. It attains efficient sampling by biasing the Markov Chain toward *sparse* networks. The proposal modifies a vertex's parent set by choosing one of three actions: (i) "add parent", (ii) "remove parent", or (iii) "swap parent" (which simultaneously adds one parent and removes another). SSPS assigns probabilities to these actions in a way that prefers sparse networks. Intuitively, when a parent set is "too large," there is a high probability of `remove-parent`; when a parent set is "too small," then there is a high probability of `add-parent`; and when the parent set is "about right," then each action has similar probabilities. See Figure 1.2 for illustration.

After the Markov chain has run sufficiently long, SSPS discards the first half of the samples as burn-in and computes expected values from the second half. I used two diagnostics to identify convergence failure: the *Potential Scale Reduction Factor* (PSRF) and *effective number of samples* (N_{eff}). See Gelman et al. (2014) for a detailed explanation.

Evaluating the method

Baselines of comparison. I compared SSPS against a suite of established pathway inference techniques. These included the exact DBN inference technique of Hill et al. (2012); a nonparametric statistical test called FunChisq (Zhang and Song, 2013); and a simple LASSO technique.

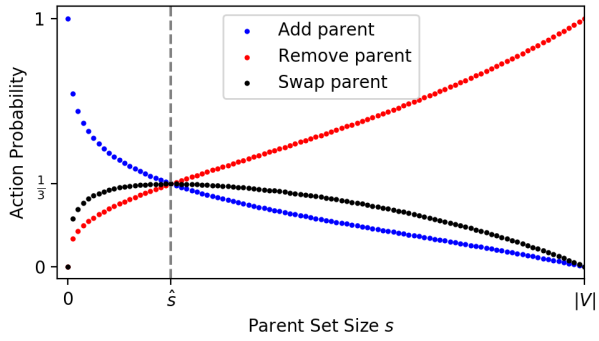


Figure 1.2: Action probabilities as a function of parent set size. The reference size $\hat{s}$ is determined from prior knowledge. It approximates the size of a "typical" parent set. When $s<\hat{s}$, `add-parent` is most probable; when $s>\hat{s}$, `remove-parent` is most probable; and when $s=\hat{s}$, all actions have equal probability.

Simulation study. Simulated data allowed me to evaluate SSPS's network inference abilities, in a setting with access to "ground truth." I generated the data as follows:

1. Generate a random directed graph G with $|V|$ vertices and $|E|$ edges.
2. Let G define the structure of a DBN. Generate random parameters for the DBN, and use that DBN to generate multiple time series, X.
3. *Corrupt* the graph G by randomly adding $\alpha \cdot |E|$ new edges and removing $r \cdot |E|$ original edges. This yields a *corrupted prior graph*, $\tilde{G}$.

I swept over a grid of values for $|V|$, r, and α, generating multiple datasets at each point in the grid.

For each of these datasets, the task was to infer G from X and $\tilde{G}$. I framed it as *edge classification*, using AUCPR (average precision) to score SSPS and the baseline techniques. Figure 1.3 shows the results. Figure 1.4 shows some representative PR and ROC curves.

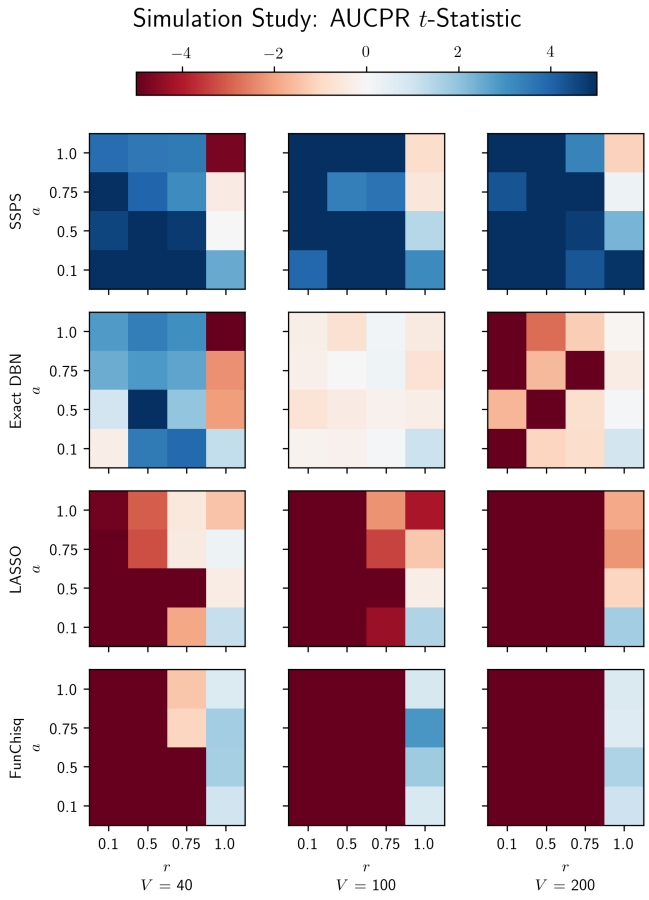


Figure 1.3: Heatmap of differential performance against the (corrupt) prior knowledge, measured by AUCPR paired t-statistics. SSPS consistently outperforms the prior knowledge across problem sizes and shows robustness to errors in the prior knowledge.

SSPS attained superior AUCPR across differing problem sizes and amounts of corruption. It successfully scaled to larger problems than the exact DBN method of Hill et al. (2012). Results on simulated data come with many caveats. However, we can safely say SSPS dominates the exact DBN method since they share very similar modeling assumptions.

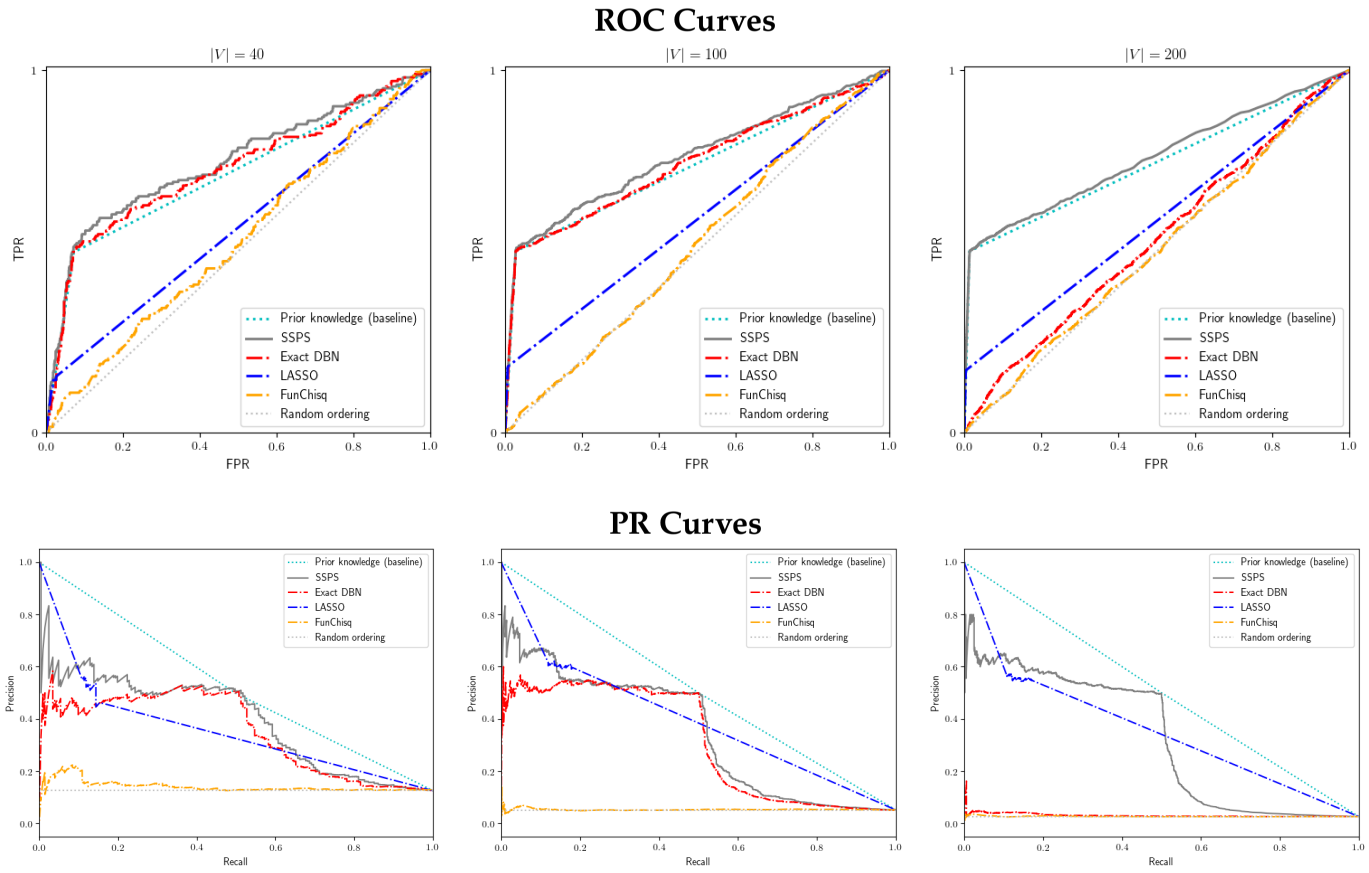


Figure 1.4: Representative ROC curves (top) and PR curves (bottom) from the simulation study. We show curves for three different simulations: $|V| = 40, 100$, and 200 (left, middle, right respectively). Each of these simulations used corruption parameters $r = a = 0.5$.

HPN-DREAM challenge. I measured SSPS's performance on real data, using the evaluation outlined in the HPN-DREAM Breast Cancer Network Inference Challenge (Hill et al., 2016). The challenge dataset consists of time series measured from 32 distinct biological "contexts" (combination of cell line and chemical stimulus). The task is to infer 32 signaling pathway structures (i.e., networks)—one for each of these contexts.

The challenge organizers have provided a set of prior edge confidences, derived from pathway databases. I use this as the matrix C in SSPS's prior distribution.

The challenge aims to score methods by their ability to capture causal relationships between pairs of variables. It estimates this by comparing predicted *descendant sets* against experimentally generated descendant sets. More specifically, the challenge organizers experimentally determined the phosphosites downstream of the protein mTOR. Comparing predicted descendants of mTOR against the experimentally validated descendants gives a notion of *false positives* and *false negatives*. As we vary a threshold on edge probabilities, the predicted mTOR descendants change, which allows us to construct an ROC curve. The resulting area under the ROC curve (AUCROC) is used to score the inferred networks.

I applied SSPS and the baseline methods to this task. The results are shown in Figure 1.5. In short, SSPS attains superior scores to the exact DBN method and comparable scores to FunChisq—a strong competitor in the original challenge. However, closer inspection reveals a more complicated picture: *prior knowledge outperforms any of the inference techniques*. This suggests quality issues in the time series data.

Additionally, the challenge is "weak" since the descendant set of one vertex provides little basis for scoring an entire network. Such an evaluation may be useful for discarding poor methods, but has little ability to discern between reasonably "good" methods. Bearing these caveats in mind, SSPS's performance on the challenge appears to be consistent with

that of a viable method. It's hard to say anything much stronger than that, though.

1.3 Conclusions

SSPS uses MCMC to infer a posterior distribution over DBN structures. Its novel *parent set proposal distribution* allows it to efficiently sample sparse networks.

The results from the simulation study and HPN-DREAM challenge show that SSPS is a viable method for signaling pathway inference. It clearly dominates the exact DBN technique of Hill et al. (2012) in its predictive power and scalability.

SSPS could be extended in several ways. On the algorithmic side, one could improve sampling efficiency by adaptively tuning the parameters in the proposal distribution during burn-in, as described by Gelman et al. (2014). The model could be modified to accommodate causal assumptions via Pearl's intervention operators (Spencer et al., 2015). Alternatively, SSPS could be extended to jointly model related pathways in a hierarchical fashion similar to Oates et al. (2014) or Hill et al. (2017). However, I don't plan to pursue these during my PhD.

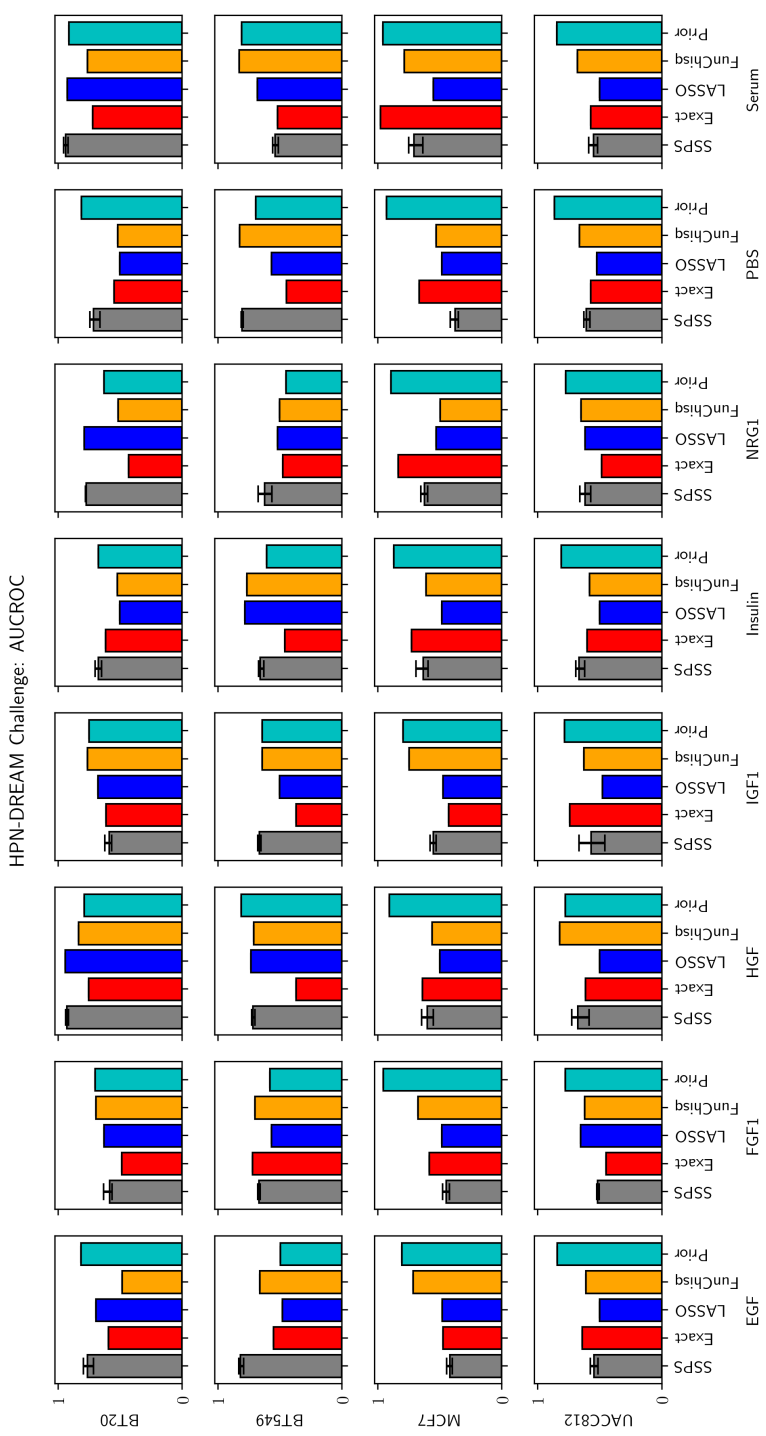


Figure 1.5: Methods' performances across contexts in the HPN-DREAM Challenge. MCMC is stochastic, so we run SSPS 5 times; the error bars show the range of AUCROC scores. The other methods are all deterministic and require no error bars.

Chapter 2

Multi-omic Reasoning

Similar to Chapter 1, this chapter describes a Bayesian inference task on biological data. I aim to infer *activation levels of signaling pathways* in biological samples, given (*i*) multi-omic data from those samples and (*ii*) prior knowledge about signaling pathways.

Unlike SSPS in Chapter 1, this project is far from complete. As such, this chapter raises many questions and offers few answers. Answering those questions constitutes the bulk of my remaining PhD research.

2.1 Background and motivation

Multi-omics and the central dogma. The central dogma of molecular biology describes a flow of information from DNA to mRNA, and from mRNA to protein. There exist technologies for collecting data at each level of this process; see Figure 2.1 for illustration.

Each kind of "-omics" data offers a complementary view of a biological system. An increasing number of techniques integrate multiple kinds of -omics data to derive useful scientific insights. We use the adjective *multi-omic* to describe these datasets and methods. See [this repository] for a list of examples.

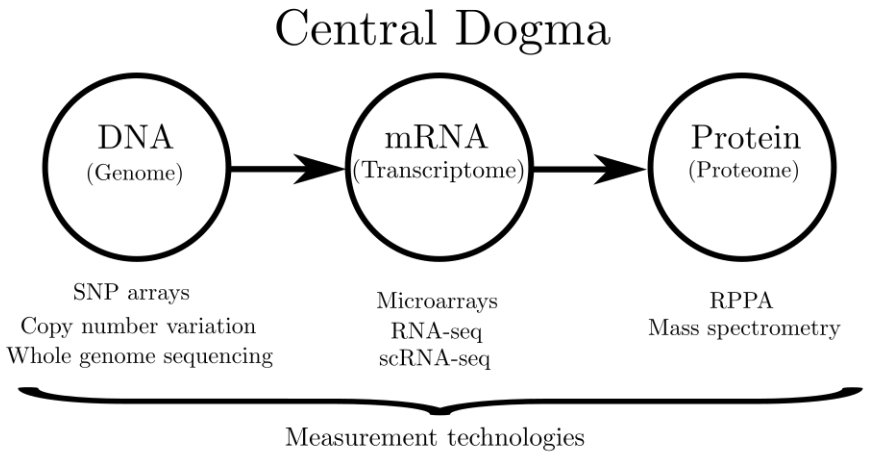


Figure 2.1: The central dogma of molecular biology. There are inexpensive technologies for collecting genomic, transcriptomic, and proteomic data.

Pathways and pathway activation. Cells regulate themselves via immensely complicated networks of interacting biochemicals. Despite the staggering complexity, biologists have identified subnetworks of biochemicals that consistently cooperate with each other to produce distinct biological functions. These "subnetworks" are called *pathways.*

Biologists have created databases to store their accumulated knowledge about pathways. These usually represent a pathway as a directed graph, with vertex and edge attribute data. Several pathway databases are freely available online.

Pathways provide a useful abstraction for understanding cellular processes. A biologist derives useful insights about a biological system by knowing which of its pathways are at work. Biologists use the somewhat abstract, intuitive term *pathway activation* to describe the extent to which a pathway explains the behavior of a system. A pathway may be more or less active in a given system.

In Chapter 1 I aimed to infer pathway structures, and used pathway database information as prior knowledge. In contrast, the present chapter

holds pathway structures fixed and uses them to impose strong bias in a high-dimensional inference task.

Clinical motivation. I collaborate with the Wisconsin Head & Neck Cancer Specialized Program of Research Excellence (HNC SPORE). HNC SPORE conducts scientific and clinical research for Head and Neck Squamous Cell Carcinoma (HNSCC).

Part of HNC SPORE's mission is to develop computational tools that extract clinically useful insights from multi-omic data. They are particularly interested in estimating pathway activations for tumor samples. The idea is to (*i*) take a tumor sample from a patient, (*ii*) collect multi-omic data from the sample, (*iii*) estimate pathway activation levels in the tumor, and (*iv*) use the pathway activations to tailor therapies for the patient. Such a tool would have scientific and clinical value. I aim to build a multi-omic probabilistic model that infers pathway activations from tumor samples.

2.2 Original contributions

Some aspects of this project are settled. Others are still subject to change.

Settled aspects of the project

Datasets. I decided to use an extensive multi-omic cancer dataset from The Cancer Genome Atlas (TCGA). TCGA contains multi-omic measurements for 11,368 cancer patients divided into 38 cancer types (The Cancer Genome Atlas Research Network et al., 2013). Not all types of data were collected for each patient—i.e., patients generally have missing data. You can think of the dataset as a hierarchy; see Figure 2.2 for illustration.

I decided to use a pathway database called the National Cancer Institute Pathway Interaction Database (NCI PID). I'm starting with a set of 138 pathways used by Vaske et al. (2010).

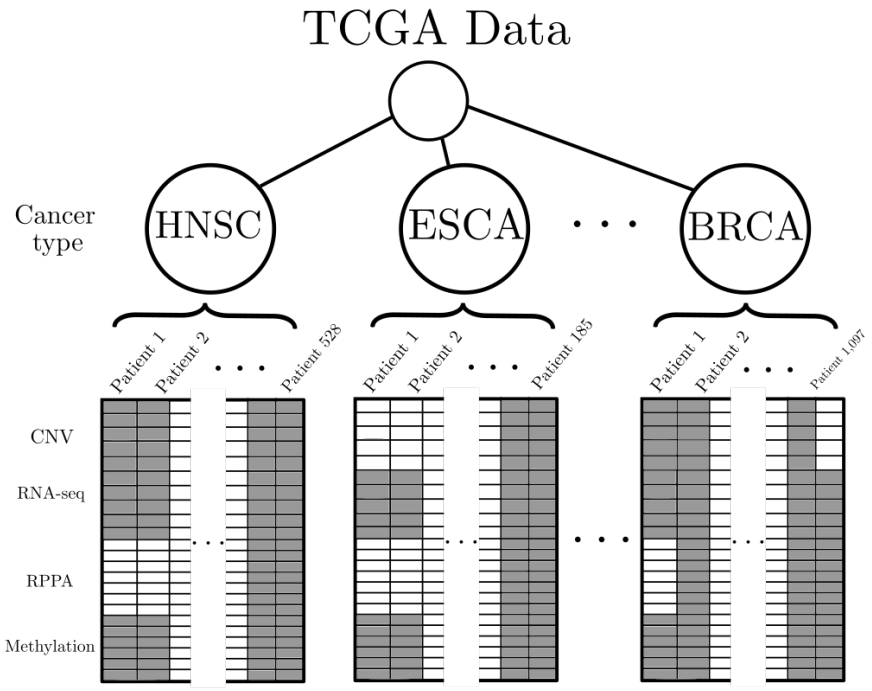


Figure 2.2: The Cancer Genome Atlas (TCGA) dataset has a hierarchical structure. Each row of the data is a feature, each column is a patient. Gray fields indicate observed data; not all measurements were taken for all patients.

Basic shape of the probabilistic model. At a high level, I have decided to model pathway activations and multi-omic data with a hierarchical Bayesian network. The idea is to allow information-sharing between patients and cancer types. See Figure 2.3 for illustration.

I define the notation appearing in Figure 2.3. Each leaf variable $x_{i,j} \in \mathbb{R}^d$ denotes the vector of multi-omic data for patient j from cancer type i. Each latent variable $\alpha \in \mathbb{R}^k$ denotes *pathway activations* for different levels of the hierarchy. My task is to infer these latent variables—particularly the patient-level ones at the bottom of the hierarchy.

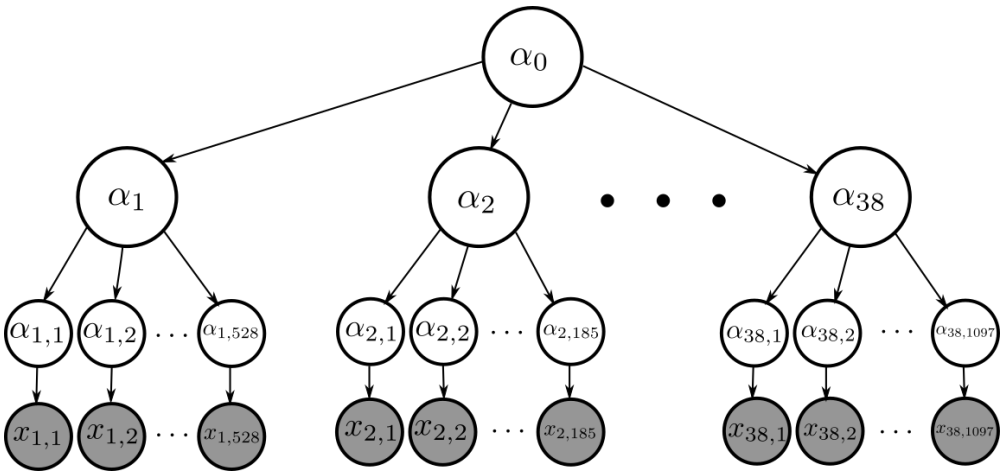


Figure 2.3: The probabilistic model will have a hierarchical structure mirroring that of the TCGA data. The bottom level is for individual patients. The middle level is for the 38 cancer types.

Method evaluations. I plan to evaluate the method's performance using (i) simulated data and (ii) inference tasks on real data. Simulations serve as a sanity check. They test the method's ability to recover latent variables in an ideal setting where data are consistent with model assumptions.

Pathway activation is an abstract, unmeasurable quantity. This makes it difficult to evaluate our method's performance on real data. However, I have other ways to determine whether the method produces useful inferences. For example, I can treat the method as a dimensionality reduction technique for supervised or unsupervised learning tasks. If the inferred pathway activations are biologically meaningful, then they should yield improved performance on those tasks. I can also perform posterior predictive tasks. This entails holding out some of the multi-omic measurements and scoring the method's ability to infer them. Lastly, I have access to experts in the HNC SPORE who can inspect the method's inferred pathway activations and judge whether they make biological sense.

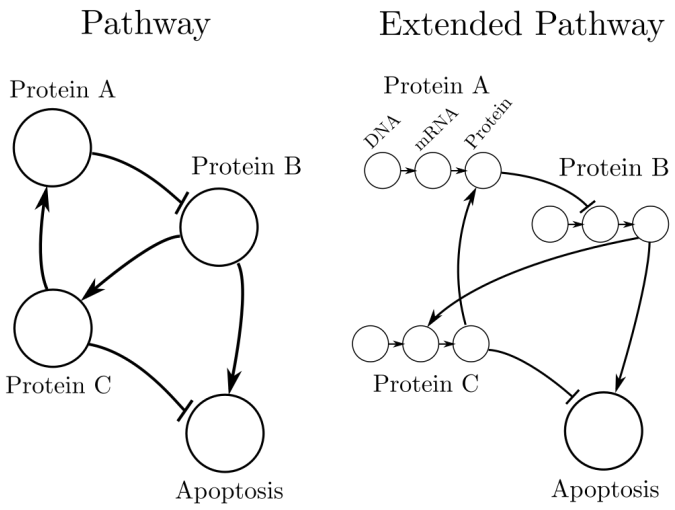


Figure 2.4: Illustration of a pathway and its corresponding *extended pathway*. New DNA and mRNA nodes are added for each protein node in the original pathway, consistent with the central dogma.

Undecided aspects of the project

Probabilistic model details. Our previous discussion glossed over many important details of the probabilistic model. The most important detail is our choice of $P(x_{i,j}|\alpha_{i,j})$. I.e., the probabilistic connection between multi-omic data and pathway activations. Once I make this decision, the other conditional distributions will have natural choices.

I have a couple of concepts that I plan to explore. Both rely on an insight introduced by Vaske et al. (2010): that pathways can define dependencies within multi-omics data. Figure 2.4 shows how pathways can be *extended* to integrate multi-omics data across the central dogma.

My first concept is based on Gaussian Graphical Models (GGMs). The idea is to construct a multivariate normal distribution that encodes some of the dependencies implied by the pathway structure. See Figure 2.5 for an illustration. Specifically, the normal distribution's *precision matrix*

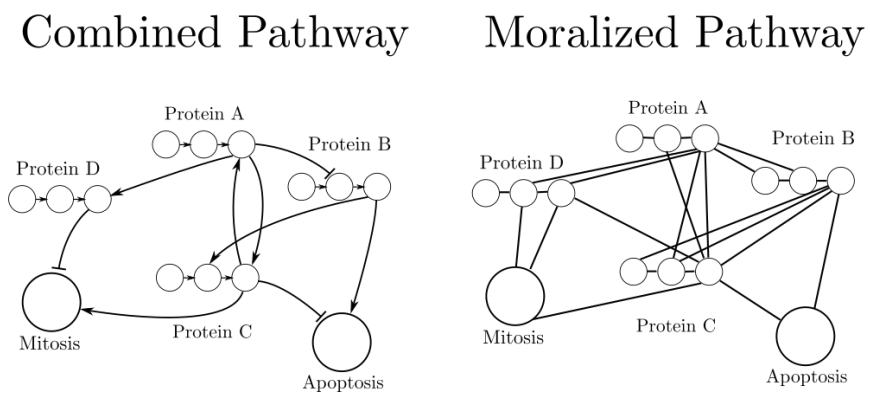


Figure 2.5: *Moralizing* the directed graph yields an undirected graph with equivalent local Markov structure. This undirected graph defines the structure of a Gaussian Graphical Model (i.e., multivariate normal distribution) for our multi-omic data.

encodes these dependencies. In short, I have the following conditional distribution in mind:

$$x_{i,j} \mid \alpha_{i,j} \sim \mathcal{N}\left(0\,,\ \epsilon I + \sum_{l=1}^{k} [\alpha_{i,j}]_l \Omega_l\right)$$

where Ω_l is a precision matrix constructed from the lth pathway. I assume each feature has been standardized to zero mean, unit variance.

Another concept I have in mind is based on linear factor analysis. The idea is to assume each multi-omic vector $x_{i,j}$ is a linear combination of vectors representing the pathways:

$$x_{i,j} = B\alpha_{i,j} + \epsilon$$

where the columns of B are the "pathway factors" and ϵ is a noise term. This is mathematically simple, but raises a challenging question: *how do we "correctly" represent pathways as vectors?*

It is well known that PCA yields orthogonal B that attains minimal *reconstruction error* (Shalev-Shwartz and Ben-David, 2014). PCA has many variants (Udell, 2015). I'm interested in exploring certain graph-regularized variants. For example, Paradkar and Udell (2017) use the following objective:

$$\min_{B} \ \left\|X - BB^\top X\right\|_F^2 - \lambda \cdot \mathrm{tr}(X^\top BLB^\top X) - \lambda' \cdot \mathrm{tr}(B^\top L'B)$$

where L and L' are *graph Laplacians*. This formulation balances reconstruction error against *similarities* defined by graphs.

Inference procedure details. The choice of inference procedure depends heavily on the model. However, I plan to derive an MLE procedure for the model I choose. Given the model's hierarchical structure, this will be a straightforward instance of Expectation-Maximization.

I also plan to implement a Bayesian procedure for my chosen model. Alternating Direction Variational Inference (ADVI) (Blei et al., 2017) seems promising, since the model is hierarchical and will most likely have differentiable conditional distributions with closed-form updates. I'm also open to other approaches such as Gibbs sampling.

Relevant works

Many techniques integrate multi-omic data for supervised and unsupervised learning tasks. See [this repository] for examples. However, very few use pathway structure to inform their analysis. PARADIGM is an exception (Vaske et al., 2010), and is the primary inspiration for this project. PLIER (Mao et al., 2019) relates closely to the linear factor analysis concept from Section 2.2, but only treats pathways as bags of genes.

There are several classic techniques for estimating pathway activations: GSEA (Subramanian et al., 2005), SPIA (Tarca et al., 2009), and HotNet2

(Leiserson et al., 2015) are widely used. However, none of them incorporate multiple kinds of omic data or share information between samples.

GNAT (Pierson et al., 2015) is an instructive example of information-sharing across a hierarchy of samples, though it solves a different inference task and doesn't integrate multi-omic data.

2.3 Conclusions

I aim to infer patient-specific pathway activations with the help of (*i*) multi-omic data, (*ii*) knowledge of pathway structures, and (*iii*) information-sharing over a hierarchy of samples. This is an ambitious project and is far from complete. However, I have the resources to produce a viable method and publishable research within the coming year.

Chapter 3

A Markov Decision Process for Adaptive Trial Design

This chapter is distinct from the others. It proposes an algorithm for designing Block Response-Adaptive Randomized (Block RAR) clinical trials. I am collaborating on this with Thevaa Chandereng[1] and Yeonhee Park[2].

The work in this chapter is partially complete. I have already formulated the problem and implemented a prototype. However, the algorithm is subject to change and I still have many experiments to run.

3.1 Background and motivation

Block Response-Adaptive Randomization (Block RAR). Suppose we have a new therapy and want to evaluate its efficacy with a Randomized Controlled Trial (RCT). For the purposes of this chapter I will always assume a two-treatment, binary-outcome trial.

[1]Department of Biostatistics, Columbia University
[2]Department of Biostatistics and Medical Informatics, University of Wisconsin–Madison

The most straightforward RCT design would assign patients to each treatment in a 1:1 ratio.This maximizes the trial's *statistical power*—the probability of correctly rejecting the null hypothesis that both treatments have equal efficacy. However, this can be undesirable since it may condemn half the patients to receive an inferior treatment. It would be better if we somehow balanced the trial's statistical power *and* successful patient outcomes.

One strategy for striking this balance is called Block Response-Adaptive Randomization (Block RAR). The idea is to break the trial into *blocks*, which are run sequentially. If a treatment performs well on earlier blocks, then the next block assigns more patients to it.

Block RAR is a simple idea, but it raises questions:

1. How many blocks should we have?
2. How big should each block be?
3. How should we allocate patients to treatments, given the results of previous blocks?

This chapter proposes an algorithm for answering these questions. I frame a block RAR trial as a Markov Decision Process (MDP) and solve it via dynamic programming. The algorithm yields an *optimal policy*; at any point in the trial, the policy tells (*i*) how large the next block should be and (*ii*) the treatment allocation ratio for that block.

Past approaches. Many approaches exist for designing adaptive RCTs. See Chapter 10 of Rosenberger and Lachin (2015) for a useful overview. Some formulate adaptive RCTs as a multi-armed bandit problem (Villar et al., 2015). Other authors find closed-form optimal allocation ratios under restrictive assumptions (Rosenberger et al., 2001; Chandereng and Chappell, 2019). Another set of approaches use dynamic programming to compute optimal designs (Hardwick and Stout, 1995, 2002).

Our method most closely resembles the dynamic programming methods of the past. However, it does not explicitly constrain the number of blocks in our trials. Unlike the closed-form analytical approaches, our method optimizes the sizes of blocks in addition to the treatment allocation ratios. Lastly, our formulation does not fit the bandit setting because it has a meaningful notion of *state*, and the set of actions depends on the current state. We assume the trial has access to a finite number of patients; as the trial progresses, the number of patients decreases and therefore fewer actions are possible.

3.2 Original contributions

Our method treats a block RAR trial as a Markov Decision Process and solves it via dynamic programming. I plan to evaluate the method according to best practices for RCT designs—i.e., simulation studies. I aim to distribute an R package containing an efficient implementation.

Markov Decision Process (MDP) formulation. A MDP is defined by four components: (*i*) a *state space*, (*ii*) *actions*, (*iii*) *transition distributions*, and (*iv*) a *reward function*.

Suppose we have a two-armed, binary outcome block RAR trial with N patients. The state space S consists of all 2×2 contingency tables of size 0 through N. Four parameters define each state: $s = (N_A, n_A, N_B, n_B)$ where N_A is the total number of patients assigned to treatment A and n_A is the number of successful outcomes for treatment A (with analogous definitions for N_B, n_B). The MDP always begins at the *empty* contingency table. The MDP has *terminal* states S_T consisting of all tables with size N. We can decrease the size of the state space by introducing a parameter κ, the *block size increment*. The idea is to constrain the size of each block to be a multiple of κ. See Figure 3.1 for an illustration of S.

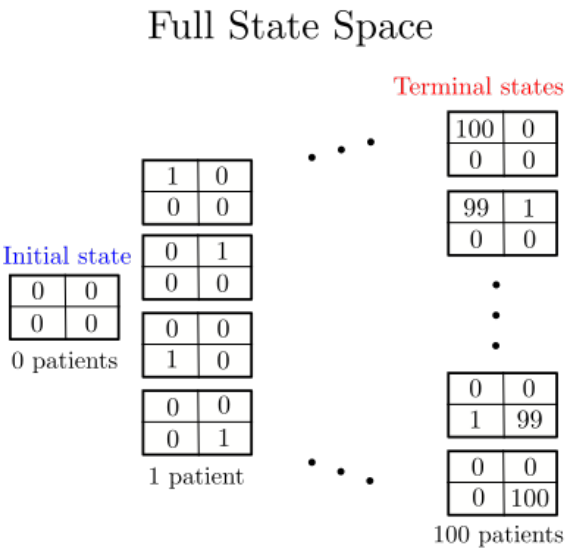


Figure 3.1: An illustration of the state space for N=100 patients and block increment κ=1. In this example, the state space contains all 2×2 contingency tables of size 0 through 100. All of the tables of size 100 are terminal states—the trial ends when all patients have participated.

Actions in the MDP correspond to possible designs for the next block of the trial. Each action has two components—the *block size* β and the *fraction allocated to treatment* A, ϕ. We assume ϕ is chosen from a discrete set Φ; for example $\Phi=\{0.2, 0.3, \ldots, 0.7, 0.8\}$.

Given a state s and action $a=(\beta, \phi)$, the MDP draws the next state s' at random from a *transition distribution*: $s' \sim t(s, a)$. In particular, we assume the patient outcomes are governed by binomial distributions, whose parameters are MAP estimates from the outcomes thus far.

$$n'_A \sim \text{Binomial}\left(\texttt{round}(\beta\cdot\phi), \frac{n_A + M\cdot\gamma}{N_A + M}\right)$$

$$n'_B \sim \text{Binomial}\left(\beta - \texttt{round}(\beta\cdot\phi), \frac{n_B + M\cdot\gamma}{N_B + M}\right)$$

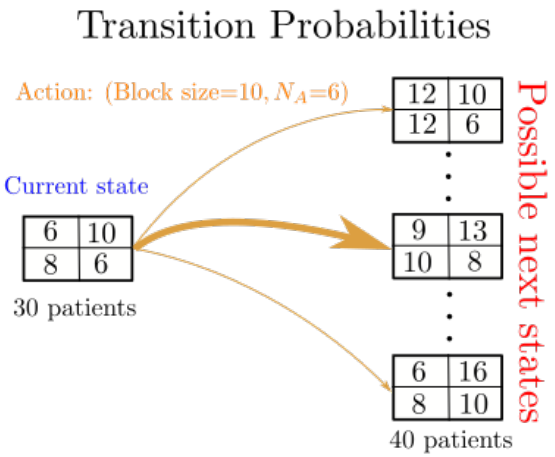


Figure 3.2: For a given state s and action a, the MDP transitions to a new state in a random fashion. The randomness is governed by a transition distribution, $t(s, a)$. In this illustration, $s = (16, 10, 14, 6)$ and $a = (10, 6)$. The MDP must transition to a state with 40 patients, but some of those states are more probable than others.

where M shrinks the estimate toward γ.

Lastly, the MDP has a reward function $R(s, s')$. The goal of a MDP is to maximize expected reward. Recall that we want to design trials that balance (i) statistical power and (ii) successful patient outcomes. With that in mind, consider the following utility function:

$$U = \mathbb{E}\{W(s_T) - \lambda_F F(s_T) - \lambda_B B\}$$

where $W(s_T)$ is the trial's *test statistic*, evaluated at the end of the trial; $F(s_T)$ is the total number of failures in the trial; and B is the total number of blocks in the trial. The expectation is with respect to *random histories*.

We translate this utility into the "expected reward maximization" frame-

work by crafting the following reward function:

$$R(s, s') = \begin{cases} W(s') - \lambda_F F(s') - \lambda_B & s' \in S_T \\ -\lambda_B & \text{otherwise} \end{cases} \tag{3.1}$$

For now we let $W(s_T)$ be the Wald statistic of a hypothesis test, with null hypothesis $\mathcal{H}_0 : p_A = p_B$. That is, the null hypothesis gives equal probability of success to both treatments.

Solving the MDP via dynamic programming. The algorithm is a straightforward dynamic program. We solve the following recurrence

$$U(s) = W(s) - \lambda_F F(s) \qquad \forall s \in S_T$$

$$U(s) = \max_a \mathbb{E}_{s' \sim t(s,a)} \{R(s, s')) + U(s')\} \qquad \forall s \notin S_T,$$

iterating through the states in order of decreasing size. The algorithm computes expected values exactly, by enumeration. It memoizes results along the way. The algorithm has this time complexity:

$$O\left(\frac{|\Phi| \cdot N^7}{\kappa^2}\right)$$

which looks prohibitively expensive. However, if we let κ increase with N (e.g., $\kappa = N/10$) then the complexity decreases to $O\left(|\Phi| \cdot N^5\right)$. In preliminary runs, the prototype solves realistic problem sizes ($N = 100$, a phase II clinical trial) in minutes.

Evaluating performance. A simulation study is the standard way to evaluate RCT designs. At a bare minimum, our algorithm must attain satisfactory power ($\geqslant 80\%$) and type I error ($\leqslant 5\%$).

I also plan to compare the algorithm against relevant baselines—e.g., the past works mentioned above. Since the algorithm is designed to maxi-

mize a specific utility function, it ought to attain higher utilities than other methods.

Implementation and code availability. I aim to write an efficient implementation in C++ and distribute it in an R package. My prototype is currently single-threaded and unoptimized. The dynamic program is embarrassingly parallelizable, so I would be remiss not to implement multi-threading.

This by itself seems to be a worthwhile contribution. None of the related dynamic programming works provide any code at all.

Other questions I plan to answer.

- How sensitive is the algorithm to "prior" parameters M and γ? Is there a reasonable heuristic for setting them?
- The utility function contains two free parameters: λ_F and λ_B. What does the Pareto frontier of optimal designs look like as those parameters vary? Does it reveal anything surprising about the tradeoffs between statistical power, patient outcomes, and number of blocks? Is there a sensible way to set these parameters?
- How does the block increment κ affect the utility of a trial design?
- Can a design for a small trial be used to guide a larger trial? This may be useful given the algorithm's computational expense.

3.3 Conclusions

This project differs from the others described in this report. I am confident in its value for multiple reasons: (i) it improves on past dynamic programming approaches in substantive ways; (ii) as a multi-objective

optimization problem, it may reveal something interesting about tradeoffs between statistical power and patient outcomes; and (*iii*) I'm delivering an efficient and accessible software package, filling a gap that exists in the area of adaptive trial designs.

Timeline

- **Fall 2020**
  - Multi-omic Reasoning
    - ☐ Acquire data
      - ☐ Head and Neck Cancer data (TCGA)
      - ☐ Other kinds of cancer data (TCGA)
    - ☐ Build the evaluation pipeline
      - ☐ Simulation study
      - ☐ Test on held-out data
      - ☐ Clustering/identifying clinically significant groups
  - Block RAR Optimization
    - ☐ Finish prototype
    - ☐ Build the evaluation pipeline
  - Dissertation
    - ☐ Write preliminary report
    - ☐ Complete preliminary exam
- **Spring 2021**
  - Multi-omic Reasoning
    - ☐ Implement prototypes, continually make improvements

    - ☐ "Linear regression" model class
    - ☐ GGM model class
  - ☐ Obtain code for baselines of comparison; incorporate into evaluation
- Block RAR Optimization
  - ☐ Finish evaluations
  - ☐ Write journal article (*Biostatistics*)
  - ☐ Submit article for review

- **Summer 2021**
  - Multi-omic Reasoning
    - ☐ Finish evaluations
    - ☐ Write journal article (*Bioinformatics*)
    - ☐ submit article for review
  - Block RAR Optimization
    - ☐ Do any additional work necessary for review and publication
  - Dissertation
    - ☐ Begin writing dissertation

- **Fall 2021**
  - Multi-omic Reasoning
    - ☐ Do any additional work necessary for review and publication
  - Block RAR Optimization
    - ☐ Do any additional work necessary for review and publication

- Dissertation
  - ☐ Finish writing dissertation
  - ☐ Thesis defense
  - ☐ Submit dissertation

References

Blei, David M., Alp Kucukelbir, and Jon D. McAuliffe. 2017. Variational Inference: A Review for Statisticians. *Journal of the American Statistical Association* 112(518):859–877.

Carlin, Daniel E., Evan O. Paull, Kiley Graim, Christopher K. Wong, Adrian Bivol, Peter Ryabinin, Kyle Ellrott, Artem Sokolov, and Joshua M. Stuart. 2017. Prophetic granger causality to infer gene regulatory networks. *PLOS ONE* 12(12):e0170340.

Chandereng, Thevaa, and Rick Chappell. 2019. Robust Blocked Response-Adaptive Randomization Designs. *arXiv:1904.07758* [*stat*]. ArXiv: 1904.07758.

Cheong, Raymond, Alex Rhee, Chiaochun Joanne Wang, Ilya Nemenman, and Andre Levchenko. 2011. Information Transduction Capacity of Noisy Biochemical Signaling Networks. *Science* 334(6054):354–358.

Cusumano-Towner, Marco F., Feras A. Saad, Alexander K. Lew, and Vikash K. Mansinghka. 2019. Gen: A general-purpose probabilistic programming system with programmable inference. In *Proceedings of the 40th acm sigplan conference on programming language design and implementation*. PLDI 2019, New York, NY, USA: ACM.

Eker, Steven, Merrill Knapp, Keith Laderoute, Patrick Lincoln, Jose Meseguer, and Kemal Sonmez. 2002. Pathway logic: symbolic analysis of biological signaling. *Pacific Symposium on Biocomputing* 400–412.

Gelman, Andrew, John B. Carlin, Hal S. Stern, David B. Dunson, Aki Vehtari, and Donald B. Rubin. 2014. *Bayesian data analysis*. 3rd ed. CRC Press.

Gjerga, Enio, Panuwat Trairatphisan, Attila Gabor, Hermann Koch, Celine Chevalier, Francesco Ceccarelli, Aurelien Dugourd, Alexander Mitsos, and Julio Saez-Rodriguez. 2020. Converting networks to predictive logic models from perturbation signalling data with CellNOpt. *bioRxiv* 2020.03.04.976852.

Gregorczyk, Marco. 2010. An introduction to Gaussian Bayesian networks. In *Systems biology in drug discovery and development: Methods and protocols*, ed. Qing Yan, chap. 6, 121–147. Springer.

Guziolowski, Carito, Santiago Videla, Federica Eduati, Sven Thiele, Thomas Cokelaer, Anne Siegel, and Julio Saez-Rodriguez. 2013. Exhaustively characterizing feasible logic models of a signaling network using Answer Set Programming. *Bioinformatics* 29(18):2320–2326.

Hardwick, Janis, and Quentin F. Stout. 2002. Optimal few-stage designs. *Journal of Statistical Planning and Inference* 104(1):121–145.

Hardwick, Janis P., and Quentin F. Stout. 1995. Exact Computational Analyses for Adaptive Designs. *Lecture Notes-Monograph Series* 25:223–237. Publisher: Institute of Mathematical Statistics.

Henriques, David, Alejandro F. Villaverde, Miguel Rocha, Julio Saez-Rodriguez, and Julio R. Banga. 2017. Data-driven reverse engineering of signaling pathways using ensembles of dynamic models. *PLOS Computational Biology* 13(2):e1005379.

Hill, Steven M., Laura M. Heiser, Thomas Cokelaer, Michael Unger, Nicole K. Nesser, Daniel E. Carlin, Yang Zhang, Artem Sokolov, Even O. Paull, Chris K. Wong, Kiley Graim, Adrian Bivol, Haizhou Wang, Fan Zhu, Bahman Afsari, Ludmila V. Danilova, Alexander V. Favorov, Wai Shing Lee, Dane Taylor, Chenyue W. Hu, Byron L. Long, David P. Noren, Alexander J. Bisberg, HPN-DREAM Consortium, Joe W. Gray, Michael Kellen, Thea Norman, Stephen Friend, Amina A. Qutub, Elana J. Fertig, Yuanfang Guan, Mingzhou Song, Joshua M. Stuart, Paul T. Spellman, Heinz Koeppl, Gustavo Stovolitzky, Julio Saez-Rodriguez, and Sach Mukherjee. 2016. Inferring causal molecular networks: empirical assessment through a community-based effort. *Nature Methods*.

Hill, Steven M., Yiling Lu, Jennifer Molina, Laura M. Heiser, Paul T. Spellman, Terence P. Speed, Joe W. Gray, Gordon B. Mills, and Sach Mukherjee. 2012. Bayesian inference of signaling network topology in a cancer cell line. *Bioinformatics* 28:2804–2810.

Hill, Steven M., Nicole K. Nesser, Katie Johnson-Camacho, Mara Jeffress, Aimee Johnson, Chris Boniface, Simon E.F. Spencer, Yiling Lu, Laura M. Heiser, Yancey Lawrence, Nupur T. Pande, James E. Korkola, Joe W. Gray, Gordon B. Mills, Sach Mukherjee, and Paul T. Spellman. 2017. Context specificity in causal signaling networks revealed by phosphoprotein profiling. *Cell Systems* 73–83.

Krishnaswamy, Smita, Matthew H. Spitzer, Michael Mingueneau, Sean C. Bendall, Oren Litvin, Erica Stone, Dana Pe'er, and Garry P. Nolan. 2014. Conditional density-based analysis of T cell signaling in single-cell data. *Science* 346(6213):1250689.

Leiserson, Mark D M, Fabio Vandin, Hsin-Ta Wu, Jason R Dobson, Jonathan V Eldridge, Jacob L Thomas, Alexandra Papoutsaki, Younhun Kim, Beifang Niu, Michael McLellan, Michael S Lawrence, Abel

Gonzalez-Perez, David Tamborero, Yuwei Cheng, Gregory A Ryslik, Nuria Lopez-Bigas, Gad Getz, Li Ding, and Benjamin J Raphael. 2015. Pan-cancer network analysis identifies combinations of rare somatic mutations across pathways and protein complexes. *Nature Genetics* 47(2): 106–114.

Mao, Weiguang, Elena Zaslavsky, Boris M. Hartmann, Stuart C. Sealfon, and Maria Chikina. 2019. Pathway-level information extractor (PLIER) for gene expression data. *Nature Methods* 16(7):607–610.

Merrell, David, and Anthony Gitter. 2020. Inferring Signaling Pathways with Probabilistic Programming. *arXiv:2005.14062* [*cs, q-bio, stat*]. ArXiv: 2005.14062.

Molinelli, Evan J., Anil Korkut, Weiqing Wang, Martin L. Miller, Nicholas P. Gauthier, Xiaohong Jing, Poorvi Kaushik, Qin He, Gordon Mills, David B. Solit, Christine A. Pratilas, Martin Weigt, Alfredo Braunstein, Andrea Pagnani, Riccardo Zecchina, and Chris Sander. 2013. Perturbation biology: inferring signaling networks in cellular systems. *PLOS Computational Biology* 9.

Oates, Chris J., Jim Korkola, Joe W. Gray, and Sach Mukherjee. 2014. Joint estimation of multiple related biological networks. *The Annals of Applied Statistics* 8:1892–1919.

Paradkar, Mihir, and Madeleine Udell. 2017. Graph-regularized generalized low rank models. In *Cvpr workshop on tensor methods in computer vision*.

Pierson, Emma, the GTEx Consortium, Daphne Koller, Alexis Battle, and Sara Mostafavi. 2015. Sharing and Specificity of Co-expression Networks across 35 Human Tissues. *PLOS Computational Biology* 11(5):e1004220.

Rosenberger, William F., and John M. Lachin. 2015. *Randomization in Clinical Trials: Theory and Practice*. John Wiley & Sons. Google-Books-ID: ZJEvCgAAQBAJ.

Rosenberger, William F., Nigel Stallard, Anastasia Ivanova, Cherice N. Harper, and Michelle L. Ricks. 2001. Optimal Adaptive Designs for Binary Response Trials. *Biometrics* 57(3):909–913. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.0006-341X.2001.00909.x.

Sachs, Karen, Omar Perez, Dana Pe'er, Douglas A. Lauffenburger, and Garry P. Nolan. 2005. Causal Protein-Signaling Networks Derived from Multiparameter Single-Cell Data. *Science* 308(5721):523–529.

Schoeberl, Birgit, Claudia Eichler-Jonsson, Ernst Dieter Gilles, and Gertraud Müller. 2002. Computational modeling of the dynamics of the MAP kinase cascade activated by surface and internalized EGF receptors. *Nature Biotechnology* 20(4):370–375.

Shalev-Shwartz, S., and S. Ben-David. 2014. *Understanding machine learning; from theory to algorithms*. Cambridge University Press.

Shojaie, Ali, and George Michailidis. 2010. Discovering graphical Granger causality using the truncating lasso penalty. *Bioinformatics* 26(18): i517–i523. `https://academic.oup.com/bioinformatics/article-pdf/26/18/i517/536841/btq377.pdf`.

Spencer, Simon E.F., Steven M. Hill, and Sach Mukherjee. 2015. Inferring network structure from interventional time-course experiments. *The Annals of Applied Statistics* 9:507–524.

Subramanian, A., P. Tamayo, V. K. Mootha, S. Mukherjee, B. L. Ebert, M. A. Gillette, A. Paulovich, S. L. Pomeroy, T. R. Golub, E. S. Lander, and J. P. Mesirov. 2005. Gene set enrichment analysis: A knowledge-based

approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences* 102(43):15545–15550.

Tarca, Adi Laurentiu, Sorin Draghici, Purvesh Khatri, Sonia S. Hassan, Pooja Mittal, Jung-sun Kim, Chong Jai Kim, Juan Pedro Kusanovic, and Roberto Romero. 2009. A novel signaling pathway impact analysis. *Bioinformatics* 25(1):75–82.

The Cancer Genome Atlas Research Network, John N Weinstein, Eric A Collisson, Gordon B Mills, Kenna R Mills Shaw, Brad A Ozenberger, Kyle Ellrott, Ilya Shmulevich, Chris Sander, and Joshua M Stuart. 2013. The Cancer Genome Atlas Pan-Cancer analysis project. *Nature Genetics* 45(10): 1113–1120.

Udell, Madeleine. 2015. Generalized Low Rank Models. Ph.D. thesis, Stanford University.

Vaske, Charles J., Stephen C. Benz, J. Zachary Sanborn, Dent Earl, Szeto Christopher, Jingchun Zhu, David Haussler, and Joshua M. Stuart. 2010. Inference of patient-specific pathway activities from multi-dimensional cancer genomics data using paradigm. *Bioinformatics* 26:i237–i245.

Villar, Sofía S., Jack Bowden, and James Wason. 2015. Multi-armed Bandit Models for the Optimal Design of Clinical Trials: Benefits and Challenges. *Statistical Science* 30(2):199–215.

Werhli, Adriano V., and Dirk Husmeier. 2007. Reconstructing gene regulatory networks with Bayesian networks by combining expression data with multiple sources of prior knowledge. *Statistical Applications in Genetics and Molecular Biology* 6.

Zhang, Yang, and Mingzhou Song. 2013. Deciphering interactions in causal networks without parametric assumptions. *arXiv* arXiv:1311.2707. 1311.2707.