Incorporation of Biological Pathway Knowledge in the Construction of Priors for Optimal Bayesian Classification

Mohammad Shahrokh Esfahani and Edward R. Dougherty

Abstract—Small samples are commonplace in genomic/proteomic classification, the result being inadequate classifier design and poor error estimation. The problem has recently been addressed by utilizing prior knowledge in the form of a prior distribution on an uncertainty class of feature-label distributions. A critical issue remains: how to incorporate biological knowledge into the prior distribution. For genomics/proteomics, the most common kind of knowledge is in the form of signaling pathways. Thus, it behooves us to find methods of transforming pathway knowledge into knowledge of the feature-label distribution governing the classification problem. In this paper, we address the problem of prior probability construction by proposing a series of optimization paradigms that utilize the incomplete prior information contained in pathways (both topological and regulatory). The optimization paradigms employ the marginal log-likelihood, established using a small number of feature-label realizations (sample points) regularized with the prior pathway information about the variables. In the special case of a Normal-Wishart prior distribution on the mean and inverse covariance matrix (precision matrix) of a Gaussian distribution, these optimization problems become convex. Companion website: gsp.tamu.edu/Publications-supplementary/shahrokh13a.

Index Terms—Phenotype classification, biological pathway knowledge, optimal Bayesian classifier (OBC), prior probability construction, regularization, convex optimization, synthetic pathway generation

1 INTRODUCTION

Phenotypic classification based on genomic data is probably the most important current issue in translational genomics. It is problematic because there are often tens of thousands of potential features (gene expressions) with very small samples, typically under 100 and often less than 50. In such circumstances, model-free classification is virtually impossible [1] and the use of prior knowledge is critical. To be more specific, once the features, sampling procedure, and classification rule are decided upon, the typical classification rule proceeds without operational knowledge concerning the features. In particular, no assumptions are made regarding the feature-label distribution (population) from which the sample data are drawn, despite the availability of a large amount of information contained in signaling pathways that specify underlying interactions between entities (e.g., genes or proteins), contributed either in normal functioning or malfunctioning (e.g., diseases) cellular states. These pathways are mostly available in the relevant literature or in public databases (e.g., KEGG, BioCarta, Reactome).

If knowledge concerning the feature-label distribution is available, then it can be used in classifier design. For instance, in [2], prior information in the form of a finite uncertainty class of feature-label distributions is incorporated to design a discrete steady-state classifier. One can employ minimum-mean-square-error (MMSE) error estimation based on a prior distribution over an uncertainty class of feature-label distributions [3], [4]. An optimal Bayesian classifier (OBC) is introduced in [5], [6] by minimizing the corresponding MMSE error estimator. Together, the MMSE error estimator and the Bayesian classifier significantly improve the two-fold goal of pattern classification, classifier design, and error estimation.

The application we have in mind is phenotype classification based on gene (or protein) expression measurements. Rather than depend only on expression data, one can use genetic pathway information to provide prior knowledge and augment classifier design. The procedure involves the following chain:

\{pathways\} → \{prior probability\} → \{optimal Bayesian classifier\}.

Prior knowledge in the form of a set of pathways is employed to constrain the space of all the measures on the feature-label distribution in accordance with the assumption that the constructed prior probability should be consistent with the pathway information. For instance a simplified illustration of the pathways that are highly influential in colon cancer is shown in Fig. 1.

Given the prior distribution governing the uncertainty class of feature-label distributions, a classifier can be constructed that performs optimally relative to the prior distribution and new data [5].

For about 200 years after the Bayes-Laplace uniform prior, Bayesian statistics was based on non-informative...
priors [8]. Jeffreys’ non-informative prior was constructed using the Fisher information [9]. There followed a series of information-theoretic and statistical methods: maximal data information priors [10], non-informative priors for integers [11], entropic priors [12], reference (non-informative) priors obtained through maximization of the missing information [8], and least-informative priors [13]. The principle of maximum entropy can be seen as a method of constructing least-informative priors [14], [15]. Except for Jeffreys’ prior, almost all methods are based on optimization: maximizing or minimizing an objective function. In [16], several non-informative and informative priors for different problems are found. All of these methods emphasize the separation of prior knowledge and observed sample data. Although these methods are appropriate tools for generating prior probabilities, they are quite general methodologies, i.e., they do not target specific scientific prior information.

Here we aim to construct a prior distribution on an uncertainty class of covariance matrices utilizing a framework consisting of three steps: 1) Pathway information quantification: information in the biological pathways is quantified via an information-theoretic perspective and translated into a set of “testable” quantities [14], [15]. Except for Jeffreys’ prior, almost all methods are based on optimization: maximizing or minimizing an objective function. In [16], several non-informative and informative priors for different problems are found. All of these methods emphasize the separation of prior knowledge and observed sample data. Although these methods are appropriate tools for generating prior probabilities, they are quite general methodologies, i.e., they do not target specific scientific prior information.

Fig. 1. A simplified wiring diagram showing the key components of the colon cancer pathways used in [7] and in Section 6. Dashed boxes are used to simplify the representation indicating identical components of their counterparts in the solid boxes.

Because the current paper makes use of incomplete regulatory knowledge to form a prior distribution to be utilized in conjunction with data to form a classifier that is optimal relative to the prior distribution and the data, it fits into the general paradigm of using incomplete regulatory knowledge to infer networks and feature-label distributions. For instance, in [18], a procedure, which resolves pathway inconsistencies by relaxing pathway timings, is proposed to infer deterministic dynamical models from Boolean pathway knowledge. In the case of incomplete knowledge the procedure outputs an uncertainty class of deterministic models. In [19], inconsistencies and incompleteness are incorporated into a single stochastic dynamical model that can cope with underlying pathway inconsistencies stemming from timing overlaps, different cellular contexts, and incomplete knowledge regarding pathways. In [2], the regularized maximum likelihood (RML) classification rule utilizes uncertainty classes constructed via an intermediate step in which pathway information is transformed to a finite number of dynamical systems, each possessing a steady-state distribution. Two class-conditional distributions are estimated using a regularization between the likelihood function and a distance to the uncertainty classes. The RML classifier is then built from these. A basic problem in all uncertainty-based methods is to quantify the uncertainty in the knowledge relative to achieving the objective, which in our case would be classification. In [20], an objective cost of uncertainty is proposed that provides such a measure based on the performance difference between the actual optimal operator (which one would know without the need for data if there were no uncertainty) and the optimal operator relative to the prior distribution and the data.

In a different approach to prior knowledge, gene-gene relationships (pathway-based or protein-protein interaction (PPI) networks) are used to improve classification accuracy [21], [22], [23], [24], [25], [30], consistency of biomarker discovery [26], [27] and targeted therapeutic strategies [28], [29]. The majority of these studies utilize gene expressions corresponding to sub-networks in PPI networks, for instance: mean or median of gene expression values in gene ontology network modules [21], probabilistic inference of pathway activity [24], and producing candidate sub-networks via a Markov clustering algorithm applied to high quality PPI networks [26], [31]. None of these methods incorporate the regulating mechanisms (activating or suppressing) into classification or feature-selection. In our case, the fundamental difference is that we develop machinery to transform knowledge contained in biological signaling pathways to prior probabilities. No single model is selected; instead, we consider all possible models that can be representative of the available prior information and assign probabilities to each model via the constructed prior. The proposed framework is unique to the extent that for the first time gene expression data can be integrated with the signaling pathways through an optimization framework, whereby the final product is an OBC. Hence, we remove a key obstacle confronting Bayesian analysis. In this vein, E. T. Jaynes remarks, “Bayesian methods, for all their advantages, will not be entirely satisfactory until we face the problem of finding the prior probability squarely [15]”.

This paper is organized as follows. Section 2 is devoted to a review of the OBC and then a methodology for quantifying the information in biological pathways. The proposed
sample-based prior constructing framework is introduced in Section 3. In Section 4, the optimization framework is developed for the Gaussian distribution with unknown mean and precision matrix governed by a Normal-Wishart prior. Simulation results on the synthetically generated pathways are provided in Section 5. We test the proposed framework on real pathways containing genes associated with colon cancer in Section 6. Finally, Section 7 contains concluding remarks.

We summarize some notation used in the paper. Boldface lower case letters denote column vectors. Concatenation of several vectors is also denoted by a boldface lower case letter. The $k$th element of the vector $\pi$ is denoted by $\pi(k)$. Boldface upper case letters are used to denote matrices. $\text{tr}(), (.)^T$, and $|.|$ denote the trace, transpose, and determinant operators, respectively; however, when the argument is not a matrix, the notation $|.|$ stands for the cardinality of a set. For a matrix $W$, if $A$ and $B$ consist of a set of rows and a set of columns in $W$, respectively, then the sub-matrix corresponding to the rows in $A$ and columns in $B$ is denoted by $W_{A,B}$. If $A = B$, then we simply write $W_A$. $\text{Pr}(E)$ denotes the probability of event $E$. $E(x|y)$ denotes taking the expectation of $g(x)$ with respect to $x$. Finally, $\log(.)$ denotes the natural logarithm. We use the terms: feature, variable, and entity interchangeably.

### 2 BACKGROUND

#### 2.1 Optimal Bayesian Classifier

Given a binary classification problem with classes $y \in \{0, 1\}$, we observe a collection of $n$ sample points, $S_n$, in a sample space $X$, with $n_x$ i.i.d. points from each class. Call $c$ the a priori probability that an individual sample point $x \in \mathbb{R}^n$ is from class 0 and let the class-conditional distribution for class $y_i$ be parameterized by $\theta_i$. The feature-label distribution is completely specified by the modeling parameters $\theta = [c, \theta_0, \theta_1]$. In [3], it is assumed that $c, \theta_0, \theta_1$ are all independent prior to observing the data. Denoting the prior for $\theta_i$ by $p(\theta_i)$, we have $p(\theta) = p(c)p(\theta_0)p(\theta_1)$. The posterior preserves independence. Denoting it by $\pi^*(\theta_i)$ and letting $x_i$ be the $i$th sample point in class $y_i$ [5],

$$
\pi^*(\theta_i) \propto \pi(\theta_i) \prod_{i=1}^{n_y} f_{\theta_i}(x_i|y_i).
$$

Priors quantify the known information about the distribution before observing data. We have the option of using diffuse (non-informative) priors, as long as the posterior (conditioned on the sample) is a valid density function. Alternatively, informative priors can supplement the classification problem with additional information. In the Bayesian framework, we characterize the initial uncertainty in the “actual distribution” through the prior. As data are observed, this uncertainty should converge to a certainty on the true distribution. The Bayesian framework for the problem of pattern classification has been widely studied in the Bayesian network view [32], [33], [34], [35]. The Bayesian framework for the pattern classification in which two prior distributions on the feature-label distributions are assumed was developed more recently with the introduction of the Bayesian MMSE error estimator [3]. The OBC is obtained by minimizing this error estimator. This classifier is given by [5]

$$
\psi_{\text{OBC}} = \begin{cases} 0, & \text{if } E_{\pi^*}[c]f(x|0) \geq (1 - E_{\pi^*}[c])f(x|1) \\ 1, & \text{otherwise}, \end{cases}
$$

where $f(x|y), y \in \{0, 1\}$, called the “effective class-conditional densities” (ECCD), are defined by

$$
f(x|y) = \int f_{\theta_y}(x|y)\pi^*(\theta_y)d\theta_y.
$$

Henceforth, we drop the sub (sup)-script denoting the dependency on the label, $y_i$, but one should recognize that the prior knowledge is assumed to be available for both classes, separately.

#### 2.2 Biological Prior Knowledge

In this section, we give a formal definition of prior knowledge in the form of pathways. We adapt some terms from [18]. For the sake of integrity, we denote the entities (e.g., gene or protein) contributed in a given set of pathways by $x(i)$ (as the $i$th element of the feature vector $x$).

Define the term “activating pathway segment” (APS) $x(i) \rightarrow x(j)$ to mean that, if $x(i)$ is “up-regulated” (UR), then $x(j)$ becomes UR (in some time steps). Similarly, the term “repressing pathway segment” (RPS) $x(i) \rightarrow \neg x(j)$ means that, if $x(i)$ is UR, then $x(j)$ becomes “down-regulated” (DR). A pathway is defined to be an APS/RPS sequence, for instance, $x(1) \rightarrow x(2) \rightarrow \neg x(3)$. In this pathway, there are two pathway segments, one APS $x(1) \rightarrow x(2)$ and one RPS $x(2) \rightarrow \neg x(3)$. A set of pathways used as the prior knowledge is denoted by $G$. We define $G_A$ and $G_R$ to include all the APS and RPS segments in $G$, respectively. We refer to regulations of the form $x(i) \rightarrow x(j)$ and $x(i) \rightarrow \neg x(j)$ as “pairwise regulations.” We denote the set of genes involved in $G$ by $g$ and, without loss of generality, we fix an order to the genes in $G$ and denote this vector of genes by $g$.

In addition to pairwise regulations, one can consider a subset of pathways. The regulatory set for gene $x$ is the set of genes affected by $x$, i.e., regulated by $x$ through some APS/RPS. We denote this set by $R_x(i)$ for gene $x(i)$. We denote the union of a gene, $x(i)$, with its regulatory set, $R_x(i)$, by $\overline{R}_x(i)$. As an example, for the pathways shown in Fig. 2,

$$
R_x(i) = \{x(3), x(4)\}, \quad R_x(2) = \{x(4), x(5)\}, \quad R_x(3) = \{x(1)\},
$$

$$
R_x(4) = \{x(5)\}, \quad R_x(5) = \{x(6)\}, \quad R_x(6) = \emptyset.
$$

$R_x(i)$ and $\overline{R}_x(i)$ denote the vectors of genes in $R_x(i)$ and $\overline{R}_x(i)$ given the order induced from vector $g$.

Pathway information is not regulatory (in a functional sense) and is understood to be marginal and incomplete.
Moreover, these pathways provide no “testable piece of information” [14]. Nevertheless, we can introduce a way of quantifying them objectively. For the moment, assume that these pathways convey “complete information,” that is, they are not affected by unspecified crosstalk or conflicting interaction therein. Under this assumption and recognizing the way in which the pathways are built from different experimental settings (conditions) in different cell lines, in a manner analogous to [36], we quantify the pairwise regulations in a conditional probabilistic manner

\[
\text{APS} : P(r(x) = \text{UR}|x(i) = \text{UR}) \\
\geq 1 - \varepsilon_{ij}; \text{ for some small } \varepsilon_{ij} > 0
\]

\[
\text{RPS} : P(r(x) = \text{DR}|x(i) = \text{UR}) \\
\geq 1 - \varepsilon_{ij}; \text{ for some small } \varepsilon_{ij} > 0.
\]

For Gaussian joint distributions, we change the inequalities in equation (3) as correlations in flow of influence is preserved; on the other hand, the definition in equation (4) are symmetric but not directional. Moreover, the interpretation of equation (3) as correlations in equation (4) is not always appropriate. Specifically, in case of a cycle (directed loop regardless of type of regulation), this two-way interpretation is inapplicable. As an example, see Fig. 2, there is an APS from x(1) to x(3) while an RPS from x(3) to x(1). Hence, when using equation (4) for the Gaussian case, we only apply it for acyclic pathways.

We also employ the conditional Shannon entropy of a gene given its regulatory set via the constraint

\[
\mathbb{H}_\theta(x(i)|R_{x(i)}) \leq \xi_i; \forall x(i) \in \mathcal{C}, \text{ for some small } \xi_i > 0,
\]

where \(\mathbb{H}_\theta(y|v_1,v_2)\) is the conditional Shannon entropy, obtained by a \(\theta\)-parameterized distribution and computed with respect to the uniform measure. \(\mathbb{H}_\theta(x(i)|R_{x(i)})\) is the amount of information needed to describe the outcome of \(x(i)\) given \(R_{x(i)}\). In (5), \(\mathcal{C}\) is the set of all elements whose regulatory set is non-empty. Note that the regulatory set information does not take regulation type (activation, inhibition) into account. Hence, we consider them as two separate pieces of information.

The assumption of having complete pathways is unrealistic and there are many sources of uncertainty impeding us from constructing a single distribution on the features. Nonetheless, the available information can be utilized to impose a probability measure (prior probability) on an uncertainty class of distributions—that is, a prior distribution, \(\pi(\theta)\), over the \(\theta\)-parameterized feature-distribution. We extend the quantification in (3) and (4) to this prior probability by

\[
\text{APS} : E_\theta[\text{Pr}(x = \text{UR}|x(i) = \text{UR})] \\
\geq 1 - \varepsilon_{ij}; \text{ for some small } \varepsilon_{ij} > 0
\]

\[
\text{RPS} : E_\theta[\text{Pr}(x = \text{DR}|x(i) = \text{UR})] \\
\geq 1 - \varepsilon_{ij}; \text{ for some small } \varepsilon_{ij} > 0.
\]

Upon relaxation to the correlation coefficients, we have

\[
\text{APS} : E_\theta[\rho_{x(i),x(j)}] \geq 1 - \varepsilon_{ij}; \text{ for some small } \varepsilon_{ij} > 0
\]

\[
\text{RPS} : E_\theta[\rho_{x(i),x(j)}] \leq -1 + \varepsilon_{ij}; \text{ for some small } \varepsilon_{ij} > 0.
\]

Furthermore, for the conditional entropy,

\[
E_\theta[H_\theta(x(i)|R_{x(i)})] \leq \xi_i; \forall x(i) \in \mathcal{C}, \text{ for some small } \xi_i > 0.
\]

### 3 Regularized Expected Mean Log-Likelihood Prior

Using prior knowledge in the form of signaling pathways, we propose a regularized expected mean-log-likelihood (REML) framework in which the expectation is taken to marginalize the dependency of the mean-log-likelihood to the actual feature-label distribution parameters (e.g., mean and covariance matrix in a Gaussian setting). The regularization is performed to apply prior information as soft constraints. The final objective function is a function of the hyperparameters of interest to determine the prior distribution.

To this end, we first split the given sample, \(S_{\text{tr}}\), into two parts for each class \(y \in \{0, 1\}\): \(S_{\text{tr}}^{y=0}\) and \(S_{\text{tr}}^{y=1}\), with \(n_y = n_0^y + n_1^y\) and \(n = n_0 + n_1\). Assume that the sample set (consisting of \(n_p = n_0^p + n_1^p\) sample points) used for prior construction is denoted by \(S_{\text{tr}}^{\text{prior}}\). Henceforth, for notational ease, we drop the index \(y\).

We state the proposed optimization framework with multiple constraints in \(\mathcal{C}\):

\[
\pi_{\text{REML}}: = \arg\min_{\pi(\theta)\in\mathcal{C}, \xi_i \geq 0} -(1 - \lambda_1 - \lambda_2)E_\theta[\ell_n(\theta)] \\
+ \lambda_1 \sum_{i=1}^{\mathcal{A}} \xi_i + \lambda_2 \left[ \sum_{(i,a) \in \mathcal{G}_A} \varepsilon_{i,a} + \sum_{(i,r) \in \mathcal{G}_R} \varepsilon_{i,r} \right]
\]

subject to the following constraints:

\[
E_\theta[H_\theta(x(i)|R_{x(i)})] \leq \xi_i, x(i) \in \mathcal{C}
\]

\[
E_\theta[\text{Pr}(x(i) = \text{UR}|x(i) = \text{UR})] \geq 1 - \varepsilon_{i,a}, (i,a) \in \mathcal{G}_A
\]

\[
E_\theta[\text{Pr}(x(i) = \text{DR}|x(i) = \text{UR})] \geq 1 - \varepsilon_{i,r}, (i,r) \in \mathcal{G}_R
\]

where \(\Pi\) is the feasible region to which the prior distribution belongs and \(\ell_n(\theta) := -1/2 \ell(\theta; S_{\text{tr}}^{\text{prior}})\), in which \(\ell(\theta; S_{\text{tr}}^{\text{prior}})\) is the log-likelihood function. \(\ell_n(\theta)\) can be interpreted as an estimator of [37], [38], [39]

\[
\int_{x \in \mathcal{X}} f(x|\theta_{\text{true}}) \log f(x|\theta) dx.
\]

which is a measure of “similarity” between the true model, governed by \(\theta_{\text{true}}\), and the one governed by the parameter \(\theta\). This estimate is also employed in the Akaike’s information
criterion for model selection [40]. In (9), the parameters $\lambda_1$ and $\lambda_2$, for which we have $\lambda_1, \lambda_2 \geq 0$ and $\lambda_1 + \lambda_2 \leq 1$, are the regularization parameters (or the design parameter) depending on the relative importance of the prior sources and likelihood. The Shannon entropy, $H_\theta(\cdot)$, is computed with respect to the uniform measure.

In equation (9), the term $E_\theta[\ell_{n_p}(\theta)]$ reflects the “expected similarity between the observed data and the true model.” Prior averaging performs marginalization with respect to the model parametrization making us depend only on the hyperparameters.

Assuming Gaussian distributions, equations (11)-(12) become

$$E_\theta[\rho_x(x_i, x(j_i))] \geq 1 - \varepsilon_{i_a, j_a}, (i_a, j_a) \in \mathcal{G}_a$$

$$E_\theta[\rho_x(x_i, x(j_i))] \leq -1 + \varepsilon_{i_v, j_v}, (i_v, j_v) \in \mathcal{G}_v. \quad (14)$$

### 4 Multivariate Gaussian with Normal-Wishart Prior

For the multivariate Gaussian distribution, $x \sim \mathcal{N}(\mu, \Lambda^{-1})$, we have $\theta = [\mu, \Lambda]$. Define the feasible region, $\Pi$, for given $\nu$ and $\kappa$, for the prior probability as $\Pi = \{\mathcal{N}(m, v, W, \kappa) : m \in \mathbb{R}^p, W > 0\}$, the set of all Normal-Wishart distributions (hence, an inverse Wishart distribution for the covariance matrix). The Normal-Wishart distribution is determined fully by four parameters, $[m_{p \times 1}, v, W_{p \times p}, \kappa]$, via

$$\mu|\Lambda \sim \mathcal{N}(m, (\nu\Lambda)^{-1})$$

$$\Lambda = \Sigma^{-1} \sim W(\nu, W, k)$$

$$= B(W, \kappa)|\Lambda|^{(\nu-p-1)/2} \exp\left\{-\frac{1}{2} tr(W^{-1} \Lambda)\right\}, \quad (15)$$

where $B(W, \kappa) \propto |W|^{-k/2}$ [41]. In order to have a proper prior, we should have $W > 0$ and $\kappa > p-1$. As $v \to 0$, the prior probability for the mean vector tends to be more non-informative (flatter).

The general optimization framework proposed in (9)-(12) does not yield a convex programming for which a guaranteed converging algorithm exists. Therefore, to facilitate convergence to the global optimum, we decompose the full procedure into two optimization problems. The main advantage of this decomposition is tractability of existing algorithms for solving convex problems. In particular, although the final solution is different from that of the initial problem, the effect of prior knowledge can be assessed by deriving analytical expressions for the gradient and the hessian of the cost functions. First, we assume $\lambda_2 = 0$ by utilizing only the regulatory set constraints: $\lambda_2 = 0 \to$ solve optimization in equations (9)-(10). Then, the second optimization problem treats the regulation types according to the constraints simplified to correlations in (13)-(14). This procedure is outlined in Fig. 3. The second optimization will be discussed in detail in Section 4.2.

#### 4.1 Regulatory Set Constraints: $\lambda_2 = 0$

Setting the regularization parameter $\lambda_2 = 0$, the general REML optimization reduces to

$$\min_{m, W > 0, \xi \geq 0} -(1 - \lambda_1) \int \ell_{n_p}(\theta) \tau(\theta)d\theta + \lambda_1 \xi, \quad (16)$$

subject to constraints of the form

$$\int H_\theta(x|R_x) \tau(\theta)d\theta \leq \xi.$$
From this distribution we obtain $E[\boldsymbol{A}] = \boldsymbol{V}^{-1}$ (see [42] for the moments of the Wishart distribution).

We will consider two cases for the covariance matrix, and consequently for $\boldsymbol{W}$. Throughout, we assume $x \notin R_x$ (no self-regulation) and have $x \sim \mathcal{N}(\mu_x, \Sigma_x)$, $r_x \sim \mathcal{N}(\mu_{r_x}, \Sigma_{r_x})$, and $\tau_x \sim \mathcal{N}(\mu_{\tau_x}, \Sigma_{\tau_x})$. $\Sigma$ denotes the precision matrix. We use $\Sigma_x$ to denote the variance of the single variable $x$.

4.1.1 Covariance Matrix Containing $\Pi_x(i)$

Suppose the precision matrix contains only those entities contributed in the constraint: the constrained entity and the elements of its regulatory set. Omitting the gene index and simply denoting a gene by $x$, if we write the precision matrix $\Lambda_{\Pi_x} = \Sigma^{-1}$ in blocks as

$$
\Lambda_{\Pi_x} = \begin{bmatrix}
\Lambda_{R_x} & \Lambda_{12} & \Lambda_{13} \\
\Lambda_{21} & \Lambda_x & \Lambda_{23} \\
\Lambda_{31} & \Lambda_{32} & \Lambda_{33}
\end{bmatrix},
$$

knowing that $\Lambda_{\Pi_x} \sim \mathcal{W}(\boldsymbol{W}_{\Pi_x}, \kappa)$, we have $\Lambda_{R_x} \sim \mathcal{W}(\boldsymbol{W}_{R_x}, \kappa)$, where

$$
\boldsymbol{W}_{\Pi_x} = \begin{bmatrix}
\boldsymbol{W}_{R_x} & \boldsymbol{W}_{12} \\
\boldsymbol{W}_{21} & \boldsymbol{W}_{x}
\end{bmatrix}.
$$

Before restating the optimization problem for this special case, we first find the constraint reflected from the Wishart prior distribution. Using existing formulas for the mutual information and the entropy of Gaussian distributions, we obtain (see Appendix A.1 on the companion website)

$$
H_x(R_x) \propto \log \left( 2\pi e - \log|\Lambda_{\Pi_x}| - \log\left(\Lambda_{\Pi_x}^{-1}\right)_{R_x} \right)
= \log 2\pi e - \log|\Lambda_x|.
$$

(19)

From the properties of the Wishart distribution, $\Lambda_x \sim \mathcal{W}(\boldsymbol{W}_x, \kappa)$. Hence, the constraint can be written as

$$
E_{\Lambda}[H_x(R_x)] \propto \log \pi e - \log|\boldsymbol{W}_x| - \psi\left(\frac{k}{2}\right).
$$

(20)

Plugging the preceding results into the optimization framework yields

$$
\text{CP}_1(\kappa) : \min_{\boldsymbol{W} > 0, \xi \geq 0} -\frac{1}{2}(1 - \lambda_1)[\log|\boldsymbol{W}_{\Pi_x}| - \kappa \text{tr}(\boldsymbol{W}_{\Pi_x} \boldsymbol{V})] + \lambda_1 \xi
$$

Subject to $-\log|\boldsymbol{W}_x| - \psi\left(\frac{k}{2}\right) \leq \xi; \xi \geq \xi_i$.

(21)

where $\xi = -\log(\pi e)$. From the inequalities in [43], one can see that the parts containing $\log|\boldsymbol{W}_{\Pi_x}|$ are concave, thereby making the optimization problem (i.e., the objective function and constraints) in (21) convex in the matrix $\boldsymbol{W}_{\Pi_x}$.

4.1.2 Covariance Matrix Containing $\Pi_x$ along with Other Entities in $\mathcal{G}$

In this section, we assume that the covariance matrix which needs to be estimated has more genes than that of $\Pi_x$. We denote the covariance matrix and its inverse by $\Sigma$ and $\Lambda$, respectively, and the parameters of the Wishart distribution governing the precision matrix by $\boldsymbol{W}$ and $\kappa$.

The precision matrix and its prior are represented in a block format by

$$
\Lambda = \begin{bmatrix}
\Lambda_{R_x} & \Lambda_{12} & \Lambda_{13} \\
\Lambda_{21} & \Lambda_x & \Lambda_{23} \\
\Lambda_{31} & \Lambda_{32} & \Lambda_{33}
\end{bmatrix},
$$

$$
\boldsymbol{W} = \begin{bmatrix}
\boldsymbol{W}_{R_x} & \boldsymbol{W}_{12} & \boldsymbol{W}_{13} \\
\boldsymbol{W}_{21} & \boldsymbol{W}_x & \boldsymbol{W}_{23} \\
\boldsymbol{W}_{31} & \boldsymbol{W}_{32} & \boldsymbol{W}_{33}
\end{bmatrix}.
$$

(22)

Then, knowing that $\Lambda_x - \Lambda_{23}\Lambda_{33}^{-1}\Lambda_{32} \sim \mathcal{W}(\boldsymbol{W}_x - \boldsymbol{W}_{23}\boldsymbol{W}_{33}^{-1}\boldsymbol{W}_{32}, \kappa - \text{dim}(\boldsymbol{W}_{33}))$, where $\text{dim}(\cdot)$ returns the dimension of a matrix, the optimization problem in (16) can now be restated as (for the conditional entropy constraints please refer to Appendix A.2 in the supplementary materials)

$$
\text{CP}_2(\kappa) : \min_{\boldsymbol{W} > 0, \xi \geq 0} -\frac{1}{2}(1 - \lambda_1)[\log|\boldsymbol{W}_x| - \kappa \text{tr}(\boldsymbol{W}V)] + \lambda_1 \xi
$$

Subject to $-\log|\boldsymbol{W}_x - \boldsymbol{W}_{23}\boldsymbol{W}_{33}^{-1}\boldsymbol{W}_{32}|$

$$
- \psi\left(\frac{k - (p - |R_x| - 1)}{2}\right) \leq \xi; \xi \geq \xi_i,
$$

(23)

Lemma 1. The programming, $\text{CP}_2(\kappa)$ is a convex programming.

Proof. Please refer to Appendix B.

Corollary 1. The optimization problems $\text{CP}_1(\kappa)$ and $\text{CP}_2(\kappa)$ satisfy Slater’s condition.

Proof. It can be readily seen from the constraints and considering the relative interior of the feasible region of the problem, by choosing $\boldsymbol{W}_x$ a scaled identity matrix.

The optimization problem in (23) can be readily extended to multiple constraints, i.e., the situation where we incorporate all the entities’ information simultaneously, by considering the corresponding submatrix for a gene and its regulatory set. This is given, for any $\xi \geq 0$ by

$$
\text{CP}_2(\kappa) : \min_{\boldsymbol{W} > 0, \xi \geq 0} -\frac{1}{2}(1 - \lambda_1)[\log|\boldsymbol{W}_x| - \kappa \text{tr}(\boldsymbol{W}V)] + \lambda_1 \sum_{i=1}^{\xi} \xi_i
$$

Subject to $-\log|\boldsymbol{W}_x| - \psi\left(\frac{k - (p - |R_x(i)| - 1)}{2}\right) \leq \xi_i$,

$$
\forall x(i) \in \mathcal{G}
$$

(24)

where

$$
\boldsymbol{W}_x(i) := \boldsymbol{W}_x(i) - \boldsymbol{W}_{x(i)}\pi_{x(i)}^{-1}\pi_{x(i)}\pi_{x(i)}, \boldsymbol{W}_{x(i)g}\pi_{x(i)}\pi_{x(i)}\pi_{x(i)}\pi_{x(i)}.
$$

(25)

4.2 Incorporating Regulation Types

Biological signaling pathways not only contain dependency information between variables, they also illustrate the type of regulation between entities. This can help decrease the uncertainty and modify our estimation of the matrix $\boldsymbol{W}$. Since, we are assuming that the underlying feature distribution is a joint Gaussian, we incorporate the APS and RPS effects using equations (13)-(14). Therefore, similar to our interpretation indicated in $\text{CP}_1(\kappa)$ or $\text{CP}_2(\kappa)$, we try to manipulate the expected correlation coefficients. However, instead of taking the expectation of the correlation coefficient, which ends up with a non-convex function, we fix the variances according to what we get from $\text{CP}_2(\kappa)$.

From the properties of the Wishart distribution, $\Sigma \sim \mathcal{W}^{-1}(\Psi, \kappa)$, where $\Psi = \boldsymbol{W}^{-1}$. Define $\Psi' = \boldsymbol{W}^{-1}$, where
$W^*$ is the optimal solution of $CP_2(k)$. The first moments of the elements of the covariance matrix distributed according to an inverse Wishart distribution, i.e., $\Sigma = [\sigma_{ij}]_{p \times p} \sim W^{-1}(\Psi, \kappa)$, are $E[\sigma_{ij}] = \frac{1}{k-1} \psi_{ij}, i, j \in \{1, \ldots, p\}$ [42], from which we approximately write

$$E[\rho_{ij} = \rho_{x(i),x(j)}] = E\left[\frac{\sigma_{ij}}{\sqrt{\lambda_{ii} \lambda_{jj}}} \right] \approx \frac{E[\sigma_{ij}]}{\sqrt{\psi_{ii} \psi_{jj}}} = \frac{\psi_{ij}}{\sqrt{\psi_{ii} \psi_{jj}}}.$$

The goal of the second optimization paradigm is twofold: while satisfying the correlation-coefficient constraints according to the regulation types, we wish to be as close as possible to the $CP_2(k)$ solution as possible. Thus, we introduce a penalty term based on the distance from the solution of $CP_2(k)$ and aim to find the closest, in the sense of the Frobenius norm, positive definite matrix to the matrix $\Psi^*$. Hence, we introduce the following optimization problem, with optimization parameter $\Psi = [\psi_{ij}]_{p \times p}$:

$$CP_3: \min_{\Psi > 0, \xi_{i,j}, \eta_{i,j} \geq 0} (1 - \lambda_2) \| \Psi - \Psi^* \|^2_F + \lambda_2 \left[ \sum_{(i,j) \in G_A} \xi_{i,j} + \sum_{(i,j) \in G_E} \eta_{i,j} \right],$$

subject to the constraints

$$\begin{align*}
1 - \xi_{i,j} & \leq \frac{\psi_{i,j}}{\sqrt{\psi_{ii} \psi_{jj}}} \leq 1; (i, j) \in G_A, \\
1 - \eta_{i,j} & \leq \frac{-\psi_{i,j}}{\sqrt{\psi_{ii} \psi_{jj}}} \leq 1; (i, j) \in G_E, \\
\psi_{ij} & = \psi_{ji}, \forall i, j \in \{1, \ldots, p\}.
\end{align*}$$

(26)

(27)

The parameter $\lambda_2 \in (0, 1)$ is again the regularization factor making the balance between two functions. It can be readily shown that the optimization problem in equations (26)-(27) is convex.

In sum, we break the general REML problem in equation (9)-(12) into two sequential problems: (1) the optimization in equations (24)-(25) ($CP_2(k)$), and then (2) the optimization in equations (26)-(27) ($CP_3$).

### 4.3 Algorithm for Solving $CP_2(k)$

For the sake of simplicity in the formula below, we only consider the one-constraint problem. The multiple constraint problem can be treated similarly. Being a nonlinear inequality constrained programming, we choose the log-barrier interior point method for solving the optimization problem $CP_2(k)$. The basic idea of the log-barrier interior point method is to replace an inequality constrained nonlinear optimization with a sequential equality constrained problems whose total number of iterations depends on the barrier parameter, some tolerance parameter, number of constraints, and the convergence criterion for the centering problem solved via Newton’s method [44].

From Corollary 1, a local optimum for $CP_2(k)$, which also satisfies the KKT (Karush-Kuhn-Tucker) conditions, corresponds to the global optimum of the optimization problem [44] (refer to Section 5.5 in [44] for more details). Hence, the solution to the KKT system of equations will provide the optimal solution to the problem of interest. Our proposed strategy is a mixture of existing strategies to solve nonlinear convex and log-determinant problems [45], [46]. The core of the algorithm is the log-barrier type of interior point method. Writing the first-order conditions for optimality (KKT system of equations), we approximate these equations by their quadratic approximation [44]. We change some notation to use existing results for log-determinant problems. We write (owing to the symmetry property)

$$W = \begin{bmatrix}
w_1 & w_2 & w_3 & \ldots & w_p \\
w_p & w_{p+1} & w_{p+2} & \ldots & w_{2p-1} \\
& \ddots & \ddots & \ddots & \ddots \\
& & \ddots & \ddots & \ddots \\
& & & \ddots & w_{(p+1)/2}
\end{bmatrix}.$$ 

(28)

This matrix can be written as $\sum_{i=1}^{(p+1)/2} w_i E_i$, where depending on the column (in the above representation) at which the variable $w_i$ is located (e.g., $j$), the matrix $E_i$ is either $E_i = e_i e_i^T + e_i e_i^T, j > i$, or $E_i = e_i e_i^T$, where the vector $e_i$ is the column-vector with 1 in its $i$ th position. Then, instead of the matrix $W$, which in general is the parameter needing to be optimized, we need only find $p(p+1)/2$ positions (due to symmetry), these being denoted by $w = [w_1, w_2, \ldots, w_{(p+1)/2}]^T$.

Hence, denoting the optimization parameter by $z = [w, \xi]$, we may write the objective function as follows, where to avoid confusion with the probability density $f$ and feature vector $x$, we use $g$ and $z$ to denote the objective function and its argument, respectively,

$$g(z) = -\frac{1}{2} \left( 1 - \lambda_1 \right) \| \log|W| - \kappa \text{tr}(WV) \| + \lambda_1 \xi.$$

(29)

Denoting the log-barrier parameter by $\mu$, the optimization problem $CP_2(k)$ may be replaced by

$$\min_{z, u} g(z) - \mu \sum_{i=1}^2 \log u(i),$$

subject to the new constraints

$$\begin{align*}
\xi - \xi - u(1) & = 0, \\
\xi + \log \begin{bmatrix} W_x & W_{23} \\
W_{32} & W_{33} \end{bmatrix} - \log|W_{33}| - u(2) & = 0.
\end{align*}$$

(30)

The new optimization problem is convex and therefore the solution to the KKT conditions provides the global optimum. In order to solve the KKT conditions, we employ the Newton method [45]. Define the vector $y$ containing the Lagrangian multipliers. Following [45], we form the dual normal matrix

$$N(z, y, u) = Hess(z, y) + A^T(z) U^{-1} YA(z),$$

(31)

where $Hess(z, y)$ is the Hessian matrix and the matrices $U$ and $Y$ are diagonal matrices whose elements correspond to vectors $u$ and $y$, respectively. In equation (31), the matrix $A(z)$ is the Jacobian matrix of the constraints (please refer to Appendix C for the Hessian and Jacobian calculus). Hence, the direction is found based on the Newton’s method solver for the KKT conditions.

Considering the $k$th iteration, once the direction, $\Delta z(k)$, is determined, a line-search is employed to find an
appropriate length of each step. As a standard approach for constrained problems for determining the step length, \( \alpha_{(k)} \), we use the “merit function” similar to that of \([45]\).

The line-search used in the \( k \)th step of the procedure (\( z_{k+1} = z_k + \alpha_k \Delta z_k \)) is described in detail in Algorithm 1.

**Algorithm 1 Line Search for \( \alpha_{(k)} \) for the centering program**

**Input:** \( \alpha_{(k)} \), \( \rho = 0.5 \) (Default Value)

**Output:** \( \alpha_{(k)} \)

**Initialize:** \( \alpha_{\text{max}} = \frac{1}{\rho} \) (\([45]\)-Section 2.1)

\[ \text{while } W \leq 0 \text{ do} \]

\[ \alpha_{\text{max}} \leftarrow \alpha_{\text{max}} \text{ (similar to \([46]\)}) \]

\[ W \leftarrow W + \alpha_{\text{max}} \Delta W(k) \]

end while

if \( \alpha_{\text{max}} \leq \alpha_{(k),1} \) then

\[ \alpha_{(k),1} \leftarrow 0.6 \alpha_{\text{max}} \]

end if

Form the merit function \( \phi_{(k)}(\alpha) \) (similar to \([45]\))

Implement back-tracking Algorithm for \( \alpha \in [\alpha_{(k),1}, \alpha_{\text{max}}] \)

return \( \alpha_{(k)} \)

The augmented parameter vector \( u \) and the Lagrangian multiplier vector \( y \) must be element-wise non-negative. Hence, as the initialization, we find an upper bound denoted by \( \Xi \) \([45]\). To assure the positive definiteness of the matrix \( W \), we decrease this maximum until the resulting matrix at the current iteration becomes positive definite. From \([46]\), if at the previous iteration the matrix \( W \) satisfies positive definiteness, then for a symmetric \( \Delta W \), there exists an \( \alpha_{\text{max}} \) for which decreasing \( \alpha \) will preserve positive definiteness. The parameter \( \rho \) is set to 0.5 as a default value. We provide the algorithm with the input

\[ \alpha_{(k),1} = -2 \frac{\phi_{(k)}(0) - \phi_{(k-1)}(0)}{\phi_{(k)}(0)}. \]

The “back-tracking algorithm” \([47]\) implementing the Wolfe first condition searches for the best reduction in the merit function.

### 4.4 Solving CP_3

The optimization problem CP_3 is a linearly-constrained quadratic programming problem. The quadratic programming without the positive definiteness constraint could be easily solved. However, since we seek to find a proper prior distribution, a positive definite matrix is of interest, making the quadratic programming more challenging. To cope with this constraint, we add the logarithm of the determinant of the matrix to the objective function. The added term can be considered as a log-barrier function used to satisfy the constraint of having a matrix with positive determinant. Because the latter condition is only a necessary condition for positive definiteness, we still check the step size to be sure the search remains in the feasible region of positive definite matrices. Overall, the search space is more restricted, thereby leading to faster convergence. Hence, we simply add the term \( \mu \log(|\Psi|) \) to the objective function, the parameter \( \mu \) being the barrier parameter.

### 4.5 Regularization Parameter

The parameter \( \kappa \) represents the spread of the prior, larger \( \kappa \) meaning that the prior is more centered about the scale matrix. Thus, \( \kappa \) can be viewed as the total amount of information in the prior. The regularization parameter aims at making a balance between two sources of information; 1) data through expected likelihood, and 2) slackness variables controlling the conditional entropy. \( \lambda_1 \) governs the relative importance of the slackness variables (information in the pathways) to the total information. We can view the total information, as represented by \( \kappa \), as being a “sum” of the amount of data used to form the prior and a proportion of \( \kappa \) relating to the importance of the slackness variables. Under this heuristic \( \kappa = n_p + \lambda_1 \kappa \), so that \( \lambda_1 = \frac{\kappa - n_p}{n_p} \).

We can also view \( \kappa \) as a sum of the data used to form the prior and the amount of data, \( n_{pw} \), that is “equivalent” to the pathway knowledge (recognizing that this “equivalence” is purely a heuristic notion). This leads to \( \kappa = n_p + n_{pw} \). Inserting this expression into the expression for \( \lambda_1 \) yields

\[ \lambda_1 = \frac{n_{pw}}{n_p + n_{pw}}. \]

We are left with defining \( n_{pw} \). In the simulations we let \( n_{pw} = m p \) for different values of \( m \geq 2 \) and see that the performance is not very sensitive to \( m \) so that a default value could simply be \( n_{pw} = 2p \).

Reflecting on the preceding heuristics we see that we are confronted with a standard problem in pattern recognition, how to regularize two conflicting factors. One thinks of the problem of adding a complexity term when dealing with model selection. We take the usual approach of applying some heuristics and then demonstrating the benefit of the regularization via simulation.

### 4.6 Differences between RML and REML Methods

Although both the RML classifier method of \([2]\) and the current REML prior construction method involve uncertainty classes, they are very different. The RML classifier is built using two estimates of the class-conditional distributions that are improved using the uncertainty classes. Moreover, there is no prior probability involved and prior knowledge in the form of finite uncertainty classes of distributions is utilized to improve classification accuracy. There are two key differences between the RML and REML methods: 1) the REML method is used to construct prior probabilities to be utilized by a Bayesian framework, e.g., optimal Bayesian classification, and the designed classifier is optimal with respect to the assumed model; 2) the RML classification rule needs a knowledge transformation, i.e., from biological pathways to a set of models, whereas the REML prior construction approach performs this knowledge transformation via the proposed optimization framework while automatically assigning probabilities to the models.

### 5 Simulations on Synthetic Examples

Our aim is to compute the true error associated with the OBC using the REML prior to examine the performance of the proposed prior construction approach. To perform the simulations, we need to fix the ground-truth model from which sample data are taken or pathways built up. We propose a method of generating synthetic pathways for a fixed model to serve as the true model governing the stochastic
5.1 Generating Synthetic Pathways Inspired by Real Experiments

We propose a method to generate synthetic pathways with different amounts of incompleteness and uncertainty, controlled by the number of experiments and number of sample points in each experiment. Details for the Gaussian case are described in Algorithm 2, in which we assume an underlying ground-truth stochastic system governed by a Gaussian distribution $\mathcal{N}(\mu_{\text{true}}, \Sigma_{\text{true}})$. An experiment takes observations, $X$, from this distribution. These observations, denoted by $S_{\mathcal{E}}$ in Algorithm 2, are used for pathway construction. Each experiment generates a set, $G$, of signaling pathways.

Pathway construction is based on Coefficient of Determination (CoD) [48]. The CoD for a random variable $x$, considering the vector $y$ as its predictor set, is defined by $\text{CoD}_y(x) = (\varepsilon - \varepsilon_*)/\varepsilon$, where $\varepsilon$ is the error of predicting $x$ without observations, that is, based on its own statistics, and $\varepsilon_*$ is the error of the optimal predictor of $x$ based on $y$. The CoD has been used since the early days of microarrays to analyze gene interaction [49]. Based on the observations $X$, the model covariance matrix is estimated and the CoD is computed using the least minimum-mean-squared error (LMMSE) estimator. For entity $x(i)$, we denote all subsets of size $k_i$ excluding $x(i)$, by $G_{k_i}(x(i))$. The best CoD-based set of size $k$ is given by

$$R_{x(i)} = \arg\max_{y \in G_{k_i}(x(i))} \text{CoD}_y(x(i)).$$

To choose APS or RPS, denote the corresponding LMMSE estimate of $x(i)$ as $\hat{x}_{\text{LMMSE}}(i)$. If the coefficient associated with a variable in this estimate is positive, then we assume APS; if it is negative, then we assume RPS. For example, if $\hat{x}_{\text{LMMSE}}(1) = 0.3x(2) - 0.7x(3)$, then $x(1) \rightarrow x(2)$ and $x(1) \rightarrow x(3)$.

Algorithm 2 Synthetic Pathways Generation

```
Input: $\mu_{\text{true}}$, $\Sigma_{\text{true}}$, $\mathcal{E}$ \in \{0,1\}^p, k \geq 1
Output: $G$
for $i = 1$ to $|\mathcal{E}|$ do
  $S_{\mathcal{E}}$ <- take $\mathcal{E}$ Sample Points $x_i \sim \mathcal{N}(\mu_{\text{true}}, \Sigma_{\text{true}})$; $i = 1, \ldots, |\mathcal{E}|$
  CoD-based pathways construction using $S_{\mathcal{E}}$
for $d = 1$ to $p$ do
  $R_{x(d)} = \arg\max_{y \in G_{k_i}(x(d))} \text{CoD}_y(x(d))$
  $x_{\text{LMMSE}(d)} = \text{LMMSE}_\mu(x(d))$ based on $R_{x(d)}$ using $S_{\mathcal{E}}$
  use $x_{\text{LMMSE}(d)}$: positive/negative coefficient: APS/RPS started from $x(d)$
end for
build $G$
end for
combine $G_i$’s to build a consensus $G$
for $d = 1$ to $p$ do
  $A \leftarrow \emptyset$
  for $i = 1$ to $|\mathcal{E}|$ do
    $A \leftarrow$ collect all the entities $x$ in $G_i$ for which we have an APS, $x(d) \rightarrow x$,
    or RPS $x(d) \rightarrow x$
  end for
  $k' \leftarrow$ average of the counts in $A$
  $k = \lceil \max(k', |\mathcal{E}|)/100 \rceil$
  $R_{x(d)} \leftarrow$ select elements of $A$ whose repetitions are greater than or equal to $k$
end for
build the consensus $\hat{G}$ using $R_{x(d)}; d \in \{1, \ldots, p\}$
return $\hat{G}$
```

Referring to Algorithm 2, each set, $G_i$, of pathways is constructed via CoD maximization after observing sample points. Having $|\mathcal{E}|$ sample sets, we have $|\mathcal{E}|$ sets of pathways $G_1, G_2, \ldots, G_{|\mathcal{E}|}$. These need to be combined to find a single consensus (similar to Fig. 1). To build this consensus, for each entity its regulatory set is the union of regulatory sets obtained in $G_1, G_2, \ldots, G_{|\mathcal{E}|}$. This union is denoted by $\hat{A}$ in Algorithm 2. Then we find the most frequent entries (controlled by $k$) in $\hat{A}$, as observed in the regulatory sets. Moreover, for a link to exist in the final consensus, it must be present in a certain percentage, $r\%$, of the $G_i$’s. Knowing all the regulatory sets, a single consensus is constructed.

In our simulations, we build the ground-truth covariance matrix using a blocked structure proposed in [50] to model the covariance matrix of gene expression microarrays. Here, however, we place a small correlation between blocks. A three-block covariance matrix with block size 3 has the structure

$$\Sigma = \begin{bmatrix} B_1 & C & C \\ C & B_2 & C \\ C & C & B_3 \end{bmatrix},$$

where

$$B_i = \begin{bmatrix} \sigma^2 & \rho_{1,2} \sigma^2 & \rho_{1,3} \sigma^2 \\ \rho_{2,1} \sigma^2 & \sigma^2 & \rho_{2,3} \sigma^2 \\ \rho_{3,1} \sigma^2 & \rho_{3,2} \sigma^2 & \sigma^2 \end{bmatrix}, \quad C = \begin{bmatrix} \rho_{1,1} \sigma^2 & \rho_{1,2} \sigma^2 & \rho_{1,3} \sigma^2 \\ \rho_{2,1} \sigma^2 & \rho_{2,2} \sigma^2 & \rho_{2,3} \sigma^2 \\ \rho_{3,1} \sigma^2 & \rho_{3,2} \sigma^2 & \rho_{3,3} \sigma^2 \end{bmatrix},$$

$\sigma^2$ is the variance of each variable, $\rho_i, i = 1, 2, 3$, are the correlation coefficients inside blocks, and $\rho_{ij}$ is the correlation coefficient between elements of different blocks.

5.2 Simulation Setup

The more concentrated the prior distribution is around the value of $\theta = [\theta_0, \theta_1]$ corresponding to the true feature-label
distribution, the better should be the performance of the OBC. Since our aim herein is prior construction, we analyze the simulations in that light. Let the misclassification error of a designed classifier, $\psi : \mathbb{R}^n \rightarrow \{0, 1\}$, designed via feature-label distributions parameterized by $\theta$ be denoted by $\epsilon(\psi, \theta) = Pr(\psi(x) \neq y | \theta)$.

Assume that we observe sample points $S_n = \{\mathbf{s}_{n1}, \ldots, \mathbf{s}_{ni}, \ldots, \mathbf{s}_{nM}\}$. Denote the OBC designed according to REML priors constructed using $\mathbf{s}_{n1}$ and training points $\mathbf{s}_{n2}$ to $\mathbf{s}_{nM}$ by $\mathbf{OBC}_{n1}$. We are concerned with $\epsilon_n(\mathbf{OBC}_{n1}, \theta_{true})$. If the solutions to the optimization paradigms stated in CP2 and CP3, shown by $\pi_{REML} = \pi(\theta_y, y) = 0, 1$, produce good priors, that is, priors that have strong concentration around $\theta_{true}$, then we should have $\epsilon_n(\mathbf{OBC}_{n1}, \theta_{true}) \leq \epsilon_n(\psi(\theta_{true})$, where $\psi$ is some other classifier, the exact relation depending on the feature-label distribution, classification rule, and sample size. On the other hand, if $\pi_{REML} : y = 0, 1$, are not concentrated around $\theta_{true}$, then it may be that $\epsilon_n(\mathbf{OBC}_{n1}, \theta_{true}) > \epsilon_n(\psi, \theta_{true})$.

Fixing the true feature-label distribution, we generate $n$ points, composing $S_{n,i}$ in the $i$th iteration, where, $i = 1, \ldots, M$. These points are split (randomly) into two parts, denoted by $S_{n,ip}$ and $S_{n,im}$, where $n_i = n_i$. Denote the given pathways by $\mathcal{G}$. Using $\mathcal{G}$ and $S_{n,ip}$, we construct prior distributions $\pi_{REML,i} : y \in \{0, 1\}$. These are updated using the remaining points from which $\mathbf{OBC}_{n,i}$ is trained. The expected true error, $\epsilon_n(\mathbf{OBC}_{n,i}, \theta_{true})$, is evaluated via Monte-Carlo simulations:

$$e_n(\mathbf{OBC}_{n,i}, \theta_{true}) \approx \frac{1}{M} \sum_{i=1}^{M} \epsilon_n(\mathbf{OBC}_{n,i}, \theta_{true}),$$

where the error term, $\epsilon_n(\mathbf{OBC}_{n,i}, \theta_{true})$ is also computed via Monte-Carlo simulations with 10,000 repetitions. The overall strategy, repeated through Monte-Carlo simulations, is partly shown in Fig. 4, and implemented step-wise as follows:

1. Fix true parameterization for two classes: $[\mu_{y, true}, \Sigma_{y, true}], y \in \{0, 1\}$.
2. Use Algorithm 2 to generate two sets of pathways, $\mathcal{G}_y, y \in \{0, 1\}$.
3. Take observations from $\mathcal{N}(\mu_{y, true}, \Sigma_{y, true})$ to generate $S_n$.
4. Randomly choose $n_{ip}$ points from $S_n$ for prior construction, i.e., $\pi_{n,ip}$ and the rest $S_{n,im}$ for training.
5. Use $\pi_{n,ip}$ and $\mathcal{G}_y$ to construct the prior $\pi_{REML,i} : y \in \{0, 1\}$, by REML (CP2 and CP3).
6. Use (2) to optimally combine the priors, $\pi_{n,ip}$ and $\pi_{REML,i}, y \in \{0, 1\}$, and $S_{n,im}$ to build the OBC, $\mathbf{OBC}_{n,i}$.

The parameters used in our simulations are summarized in Table 1. We considered a setting with $p = 8$ entities. The covariance matrix in the form of (31) is used with block sizes 3, 3, 2 for the first, second, and the third blocks, respectively. We compute the Monte-Carlo approximation of the expected true error of the designed OBC using the priors from CP2 and CP3 (shown by REML) and Jeffreys’ prior.

We also train both quadratic discriminant analysis (QDA) and linear discriminant analysis (LDA) for the purpose of comparison. QDA is the plug-in classifier for the Gaussian model with different covariance matrices, meaning that it is obtained from the Bayes (optimal) classifier for the true model by estimating the means and covariance matrices by the sample mean and sample covariance matrices, respectively. LDA is the plug-in classifier for the Gaussian model with common covariance matrix. With small samples, LDA often performs better than QDA in the different-covariance model on account of better estimation using the pooled sample covariance matrix for LDA. In the simulations, we fixed the true underlying model for two classes according to Table 1.

### 5.3 Results

We set $\lambda_1$ according to (30), $\lambda_2 = 0.5$, and consider three sample sizes, $n \in \{30, 50, 70\}$, and two class prior probabilities $c \in \{0.5, 0.6\}$. The sample sizes $n_0$ and $n_1$ are determined according to the class prior probability as $n_0 = cn$, and $n_1 = n - n_0$. We consider $\kappa_y = m + n_y, m = 2, 3, 4$. We change the ratio of the number of sample points used for prior construction to the total sample size, $r_y = \frac{n_y}{n}$, from 0.1 to 0.9. We consider at most 90% to keep points for prior update and finding the posterior. The sample sizes allocated for prior construction are determined as $n_{y} = \lceil r_y n_y \rceil$, $y = 0.1$. For example, for $c = 0.6$ and $n = 30$, when 50% of the points are used for prior construction, $n_0 = 9$ and $n_1 = 6$.

The results for the settings in Table 1 for $m = 2$ are shown in Fig. 5. Since we split the data, $n_0$ for REML prior construction and $n - n_y$ to design the OBC from the REML prior, we need to examine the effect of $n_y$. Therefore, we plot the expected true error as a function of the percentage of the data points used for prior construction, $100 \times \frac{n_0 + n_y}{n},$ for the OBC using the REML prior. The work-flow is depicted in Fig. 6, in which there are two general possibilities: 1) use all data points for prior construction, shown in the hypotenuse of the figure, or 2) use part of the data for prior construction and the rest for constructing the posterior.

"We compare these results to both QDA and LDA." We also consider the OBC with Jeffreys’ non-informative prior. Since there is no data splitting for QDA, LDA, and the OBC with a non-informative prior, all sample points are used for classifier construction so that the plots in Fig. 5 are constant.

In Figs. 5a, 5d, 5g, and 5j, $n = 30$, by increasing number of sample points used for the prior construction, the true error decreases. Thus, one should use at least 90% of the sample points for prior construction. However, when the total number of sample points increases, from 30 to 70, there is an
Fig. 5. The expected true error as a function of the percentage of the sample points used for prior construction, $n_p^0 + n_p^1$, shown in the $x$-axis. Sample points for each class are stratified according to $c = \Pr(y = 0)$. 
Sample size is changed from 30 to 70 for four scenarios C by increasing k from 2 to 70. We only consider configurations C1 and C3 for which c = 0.5. Tables 2 and 3 demonstrate n_p as a function of total sample size n for four scenarios m = 2, 3, 4. Sample size n is changed from 30 to 70.

The key point is that increasing n does not necessarily lead to a larger n_p; on the contrary, there is a saturation point after which increasing n does not significantly influence the optimal sample size for prior construction. For both tables (reflecting ε_Bayes = 0.167 and ε_Bayes = 0.091), the optimal value is approximately n_p ≈ 30, with small variations around 30 having negligible effect on classifier performance. This means that 30 points for prior construction provides close to optimal performance when n ≥ 30, so that we can view the REML prior as taking up to 30 sample points for its construction, after which further sample points, however many there be, are used for posterior construction. Should we have n < 30, using prior information is still superior to a completely data-driven classifier; however, classifier design is strictly from the prior without using a posterior to design the OBC. We would like a closed form for the true error of the OBC designed using the REML prior, but this problem appears difficult given the nature of the prior information and the optimization problems involved in prior construction.

### 6. AN EXAMPLE INSPIRED BY THE COLON CANCER PATHWAY

#### 6.1 Pathway Description

In this section, we evaluate the performance of the proposed method on real pathways. These pathways, associated with colon cancer, are depicted in Fig. 7. This is a diagram that includes three basic pathways: the Ras/Raf/Mek pathway at the left and middle in red, the PI3K pathway in the middle in blue, and the JAK/STAT pathway on the right in green. On the top are the ligands/stimulation factors, HGF, EGF, HGF, NRG1, IL6, etc. They carry the external signals generated by neighboring cells (sometimes themselves). Immediately under the factors are the ligand receptors, which are anchored at the membrane. Once the ligand binds to its receptor, it will initiate the downstream process, usually to form a dimer or similar complex so that the kinase in one unit can activate the other unit. If two nodes are drawn as attached together, they normally closely bind together to form a dimer. For example, EGFR-ERBB2 is a heterodimer. MET-MET is an homodimer. Or it means the interaction is attached together, they normally closely bind together to form a dimer. If two nodes are drawn as attached together, they normally closely bind together to form a dimer. For example, EGFR-ERBB2 is a heterodimer.

![Diagram](image-url)  
**Fig. 6.** A schematic view of two possibilities starting from a partially known prior probability, i.e., in the Normal-Wishart prior in this paper, we assume known v and k. First, using some part of the data for prior construction, and then using the rest for finding the posterior probability. Second, utilizing all the data points with the pathways to find a prior knowledge, or precisely the posterior probability.

The optimal number of points which should be utilized for the prior construction. For instance, as illustrated in Figs. 5c, 5f, 5i, 5l, after about n_p = 30, the true error of the designed OBC increases. Note that in Fig. 5k, the LDA classifier outperforms the OBC. Here we must remind ourselves that the OBC is optimal on average relative to the prior distribution, but may be outperformed for individual distributions. Even in this case, however, the REML-based OBC still significantly outperforms the OBC with Jeffreys’ prior. Results for m = 3, 4 are provided on the companion website.

The demonstrations illustrate that splitting the data provides better performance – that is, using n_p points (n_p < n) to design the prior and the remaining n - n_p points to train the OBC provides the minimum expected error. To be precise, for a given n, we are interested in the number of sample points for which the minimum expected true error is achieved using the OBC designed via the REML prior, namely,

\[ n_p^*(n) = \arg \min_{n_p \in \{2,...,n\}} \epsilon_n(\psi_{OBC,n-n_p}, \theta_{true}). \]

n_p^*(n) represents the REML-optimal investment of data size in the prior construction process. After this point, the remaining points should be employed to update the constructed prior. Since there is no closed form for the true error of the OBC designed using the REML prior, the exact value of n_p cannot be determined. Thus, we search for n_p via Monte-Carlo simulations: For fixed n, we exhaustively search for n_p as increasing n_p from 2 to n. We only consider configurations C1 and C3 for which c = 0.5. Tables 2 and 3 demonstrate n_p as a function of total sample size n for four scenarios m = 2, 3, 4. Sample size n is changed from 30 to 70.

<table>
<thead>
<tr>
<th>n</th>
<th>30</th>
<th>34</th>
<th>38</th>
<th>42</th>
<th>46</th>
<th>50</th>
<th>54</th>
<th>58</th>
<th>62</th>
<th>66</th>
<th>70</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\kappa_0 = 2p + n_p)</td>
<td>30</td>
<td>26</td>
<td>28</td>
<td>28</td>
<td>32</td>
<td>28</td>
<td>28</td>
<td>26</td>
<td>28</td>
<td>28</td>
<td>30</td>
</tr>
<tr>
<td>(\kappa_0 = 3p + n_p)</td>
<td>26</td>
<td>28</td>
<td>26</td>
<td>30</td>
<td>30</td>
<td>26</td>
<td>28</td>
<td>28</td>
<td>28</td>
<td>28</td>
<td>28</td>
</tr>
<tr>
<td>(\kappa_0 = 4p + n_p)</td>
<td>30</td>
<td>28</td>
<td>28</td>
<td>30</td>
<td>24</td>
<td>28</td>
<td>28</td>
<td>32</td>
<td>28</td>
<td>28</td>
<td>30</td>
</tr>
</tbody>
</table>

This means that 30 points for prior construction provides close to optimal performance when n ≥ 30, so that we can view the REML prior as taking up to 30 sample points for its construction, after which further sample points, however many there be, are used for posterior construction. Should we have n < 30, using prior information is still superior to a completely data-driven classifier; however, classifier design is strictly from the prior without using a posterior to design the OBC. We would like a closed form for the true error of the OBC designed using the REML prior, but this problem appears difficult given the nature of the prior information and the optimization problems involved in prior construction.

#### Table 2

The Optimal Prior Constructing Sample Size, n_p as a Function of Total Sample Size for the Configuration C1

![Diagram](image-url)
IS6ST-IS6R is located. These reactions happen very fast once the ligand binds to the receptor. A long arrow means that the protein will move into the cytoplasm and activate/inhibit/modify the target protein. As indicated in the legend, “+P” means phosphorylation and “T” means transcription. “GAP” means GTPase-activating protein, because RHEB is a GTPase. So this is an activating process. Overall, phosphorylation and GTPas-activating are both protein modification procedures that happen very fast since they involve no transcription/translation. In particular, such a process cannot be observed in a transcription assay, such as a microarray or RNAseq.

From the wiring diagram in Fig. 7, we concentrate on 11 entities: EGF, Ras, MEK1/2, PIK3CA, STAT3, mTORC1, HGF, IL6, PKC, SPYR4, and TSC1/TSC2. Thus, the feature vector is

\[
x = [\text{EGF}, \text{HGF}, \text{IL6}, \text{Ras}, \text{PIK3CA}, \text{STAT3}, \text{TSC1}/\text{TSC2}, \text{mTORC1}, \text{SPYR4}, \text{PKC}, \text{MEK1}/2],
\]

where the TSC1/TSC2 tumor suppressor complex is considered as a single entity.

Since we do not exactly know what type of functioning exists for each of these genes, we simply assign some dependency between these genes according to the pathways in Fig. 7. We do a similar procedure for both classes, assuming some mutations or changes for the malfunctioning label. From these assumed dependencies, we construct covariance matrices for two classes.

### 6.2 Pathway-Consistent True Model Construction

As discussed in Section 5.1, the basic requirements for the numerical experiments are pathways for two classes and

| Table 3: The Optimal Prior Constructing Sample Size, \(n^*_p\) as a Function of Total Sample Size for the Configuration C3 |
|-----------------|------|------|------|------|------|------|------|------|------|
| \(\kappa_x\) = \(2p + n_g\) | 30   | 34   | 38   | 42   | 46   | 50   | 54   | 58   | 62   | 66   | 70   |
| \(\kappa_y\) = \(3p + n_g\) | 28   | 30   | 30   | 34   | 28   | 28   | 30   | 26   | 30   | 28   | 28   |
| \(\kappa_z\) = \(4p + n_g\) | 30   | 28   | 32   | 38   | 38   | 24   | 34   | 34   | 32   | 30   | 28   |

Fig. 7. A wiring diagram showing proliferation and survival pathway elements whose transcriptional states could be altered in a cell exposed to the drug lapatinib [7], [51]. Nodes marked in yellow are ones for which a reporter would be used to assess transcription for that gene. The places where the drug of interest and other drugs that act at other points on these pathways are indicated by red labels.
sample data generated using a fixed true model. In the Gaussian case, we need to fix the underlying true mean vector and covariance matrix, \( \mu_y^{true}, \Sigma_y^{true} \), for \( y \in \{0, 1\} \). Owing to the structure of the colon cancer pathways in Fig. 7, we first set the covariance matrix restricted to the top genes \( \text{EGF, HGF, IL6} \), i.e., \( [x(1) x(2) x(3)] \). We denote the mean vector and covariance matrix restricted to these three genes by \( [\mu_y^{true}]_{x(1)2(2)3(3)} \) and \( [\Sigma_y^{true}]_{x(1)2(2)3(3)} \) respectively. Thus, for class \( y = 0 \),

\[
\text{EGF, HGF, IL6} \sim \mathcal{N}
\left( [\mu_y^{true}]_{x(1)2(2)3(3)}, [\Sigma_y^{true}]_{x(1)2(2)3(3)} \right).
\]

We assume \( [\mu_y^{true}]_{x(1)2(2)3(3)} = -[\mu_0^{true}]_{x(1)2(2)3(3)} \) and \( [\Sigma_y^{true}]_{x(1)2(2)3(3)} = 2[\Sigma_0^{true}]_{x(1)2(2)3(3)} \). Then, to keep the Gaussianity, for both classes we assume linear dependencies in the form of \( x(i) = a_i^T x_{-i} + z; i = 4, 5, \ldots, 11 \), where the vectors \( a_i \), \( i = 4, \ldots, 11 \), are coefficients determining the influence of each gene in \( x_{-i} = [x(1), x(i-1)]^T \) on the target gene \( x(i) \). The \( z \)'s are additive zero-mean Gaussian noise, \( z_i \sim \mathcal{N}(0, \sigma^2_z) \), considered to model the effects of latent variables outside the model [52], [53].

Having Fig. 7 as the foundation for the pathways, if two genes are not connected we simply assume that the corresponding coefficient in the vector \( a_i \) is zero. If there is an APS/RPS, we assume a positive/negative coefficient, respectively. Hence, considering the normal functioning of the cell, we consider the following linear relationships among the variables conditioned on being in class \( y = 0 \):

\[
\text{Ras} = a_4(1) \text{EGF} + a_4(2) \text{HGF} + a_4(3) \text{IL6} + z_4 \quad (34a)
\]

\[
\text{PIK3CA} = a_5(2) \text{HGF} + a_5(4) \text{Ras} + z_5 \quad (34b)
\]

\[
\text{STAT3} = a_6(1) \text{EGF} + a_6(3) \text{IL6} + a_6(5) \text{PIK3CA} + z_6 \quad (34c)
\]

\[
\text{TSC1/TSC2} = a_7(5) \text{PIK3CA} + z_7 \quad (34d)
\]

\[
\text{mTORC1} = a_8(7) \text{TSC1/TSC2} + z_8 \quad (34e)
\]

### Table 4

Regulatory Sets of the Genes Considered in our Classification Scenario Using Pathways in Fig. 7

<table>
<thead>
<tr>
<th>Gene</th>
<th>Regulatory set ( (y = 0) )</th>
<th>Regulatory set ( (y = 1) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>EGF</td>
<td>{Ras, STAT3}</td>
<td>{Ras, STAT3}</td>
</tr>
<tr>
<td>HGF</td>
<td>{Ras, PIK3CA}</td>
<td>{Ras, PIK3CA}</td>
</tr>
<tr>
<td>IL6</td>
<td>{Ras, STAT3, PKC}</td>
<td>{Ras, STAT3, PKC}</td>
</tr>
<tr>
<td>Ras</td>
<td>{MEK1/2, PIK3CA}</td>
<td>{MEK1/2, PIK3CA}</td>
</tr>
<tr>
<td>PIK3CA</td>
<td>{STAT3, TSC1/TSC2}</td>
<td>{STAT3}</td>
</tr>
<tr>
<td>STAT3</td>
<td>{STAT3, IL6, SPRY4}</td>
<td>{STAT3, IL6, SPRY4}</td>
</tr>
<tr>
<td>TSC1/TSC2</td>
<td>{mTORC1}</td>
<td>{mTORC1}</td>
</tr>
<tr>
<td>mTORC1</td>
<td>{SPRY4}</td>
<td>{SPRY4}</td>
</tr>
<tr>
<td>SPRY4</td>
<td>{PKC}</td>
<td>{PKC}</td>
</tr>
<tr>
<td>PKC</td>
<td>{MEK1/2}</td>
<td>{MEK1/2}</td>
</tr>
<tr>
<td>MEK1/2</td>
<td>\emptyset</td>
<td>\emptyset</td>
</tr>
</tbody>
</table>

The second and the third columns correspond to two classes \( y = 0 \) and \( y = 1 \), respectively. The only mutation considered to distinguish two classes is in TSC1/TSC2 complex which is stuck at zero.

<table>
<thead>
<tr>
<th>Gene</th>
<th>Regulatory set ( (y = 0) )</th>
<th>Regulatory set ( (y = 1) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>EGF</td>
<td>{Ras, STAT3}</td>
<td>{Ras, STAT3}</td>
</tr>
<tr>
<td>HGF</td>
<td>{Ras, PIK3CA}</td>
<td>{Ras, PIK3CA}</td>
</tr>
<tr>
<td>IL6</td>
<td>{Ras, STAT3, PKC}</td>
<td>{Ras, STAT3, PKC}</td>
</tr>
<tr>
<td>Ras</td>
<td>{MEK1/2, PIK3CA}</td>
<td>{MEK1/2, PIK3CA}</td>
</tr>
<tr>
<td>PIK3CA</td>
<td>{STAT3, TSC1/TSC2}</td>
<td>{STAT3}</td>
</tr>
<tr>
<td>STAT3</td>
<td>{STAT3, IL6, SPRY4}</td>
<td>{STAT3, IL6, SPRY4}</td>
</tr>
<tr>
<td>TSC1/TSC2</td>
<td>{mTORC1}</td>
<td>{mTORC1}</td>
</tr>
<tr>
<td>mTORC1</td>
<td>{SPRY4}</td>
<td>{SPRY4}</td>
</tr>
<tr>
<td>SPRY4</td>
<td>{PKC}</td>
<td>{PKC}</td>
</tr>
<tr>
<td>PKC</td>
<td>{MEK1/2}</td>
<td>{MEK1/2}</td>
</tr>
<tr>
<td>MEK1/2</td>
<td>\emptyset</td>
<td>\emptyset</td>
</tr>
</tbody>
</table>

### Table 5

Table of Parameters Used for Simulations

<table>
<thead>
<tr>
<th>Class ( y )</th>
<th>( \mu_y^{true} )</th>
<th>( \Sigma_y^{true} )</th>
<th>Noise variance</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.31p</td>
</tr>
<tr>
<td>0.2 1 0.2</td>
<td>0.2 0.2 1</td>
<td>( \sigma^2_z = 0.2 )</td>
<td>( i = 1, \ldots, 8 )</td>
</tr>
<tr>
<td>1 0.4 0.4</td>
<td>0.4 0.4 2</td>
<td>-0.31p</td>
<td>( \sigma^2_z = 0.05, i \neq 7 )</td>
</tr>
<tr>
<td>1 0.4 0.4</td>
<td>0.4 0.4 2</td>
<td>-0.31p</td>
<td>( \sigma^2_z = 0.05, i \neq 7 )</td>
</tr>
</tbody>
</table>

The Bayes error is \( \epsilon_{\text{Bayes}} = 0.132 \).

In which, those \( a_i(j) \)'s, not contributing in equations (34a)-(34h), are set to zero. Moreover, for nonzero coefficients, except \( a_5(5) \), \( a_7(10) \), and \( a_9(9) \), all other coefficients are positive. The other difference we assume for distinguishing two classes is a mutation for the TSC1/TSC2 tumor suppressor complex [54], [55]. Precisely, for \( y = 1 \), we change equation (34d) to \( x(7) = TSC1/TSC2 = z_7 \), meaning that this gene is stuck at 0 with a small probability of being changed. Considering the conditional entropy constraints, we extract the regulatory set connections used for the REML prior construction in Table 4 for the two classes. We set \( a_i(j) = \frac{1}{\sqrt{N}} \), where \( N_i \) is the number of nonzero elements of \( a_i \). The sign is determined based on whether the influence is through an APS or an RPS. For example, for \( \text{STAT3} : a_6(1) = a_6(2) = \frac{1}{2} \) and \( a_6(3) = -\frac{1}{2} \).

The coefficients for the true model used for simulations are given in Table 5, \( 1_m \) denotes an all-one column-vector with dimension \( m \). Then, according to these coefficients, using equations (34a), (34b), (34c), (34d), (34e), (34f), (34g), (34h), we build the underlying true mean vectors and covariance matrices for both classes. These moments will be used to generate data points during our simulations, i.e., \( x \sim cN(\mu^{true}_0, \Sigma^{true}_0) + (1-c)N(\mu^{true}, \Sigma^{true}) \), where \( c \) is fixed in our simulations to 0.5. We also have \( p = 11 \).

### 6.3 Results

Similar to Section 5.3, we show the results for different sample sizes, but for a single Bayes error \( \epsilon_{\text{Bayes}} = 0.108 \) in Fig. 8, which shows the comparisons for sample sizes \( n = 30, 50, \) and 70, with \( d_0 = n_1 = n/2 \) and \( m = 2 \) (\( m = 3.4 \) on the companion website). We have removed the line for LDA, since the LDA error is so large that the differences between the other methods could not be easily seen. One can see that the superiority of the OBC designed using the constructed prior diminishes as number of sample points increases, with only small improvement over QDA. Nonetheless, we see a significant improvement in the small sample settings \( (n \leq 50) \), which is our ultimate goal.

### 7 Conclusion

Purely data-driven approaches to classifier design with small samples tend to produce poor classifiers whose errors
cannot be reliably estimated. The importance of small-sample classification is highlighted by its prevalence in genomic/proteomic applications. In general, prior (probability) selection is one of the main challenges when one is dealing with any Bayesian framework. Conjugate priors are of great interest because of their convenient properties for deriving the posterior probabilities; however, there is no general rigorous mathematical machinery from which to estimate the hyperparameters. The proposed optimization framework is different from its predecessors in the sense that the REML prior relies on sample data and incorporates these data with “pure prior knowledge” to obtain a prior probability. The objective function is based on the notion of a model selection criterion, where the criterion is marginalized using the prior probability. The performance of the designed prior is examined by evaluating the true error of the OBC designed via the posterior.

As a final comment, let us note that the overarching goal is to use prior knowledge, in the form of biological pathways, to assist in the design of genomic classifiers. Since we use some initial data in prior construction and thereafter use new data to construct a posterior distribution in the Bayesian framework, one might consider this a “hybrid” approach. But from the perspective of our goal, integration of pathway knowledge and data, this characterization is semantic. The fundamental conclusion is that pathway knowledge and data, this characterization is semantic. The fundamental conclusion is that pathway knowledge and data are used in a Bayesian framework to produce classifiers that are superior to those based on data alone, and this is done via an optimization procedure that transforms the pathway knowledge into constraints on the feature-label distribution.

REFERENCES


Mohammad Shahrokh Esfahani received the BS degree from the University of Tehran, Iran, in 2007 and the MS degree from Sharif University of Technology, Iran, in 2009, all in electrical engineering. Since 2009, he has been working toward the PhD degree in Genomic Signal Processing Laboratory, Department of Electrical and Computer Engineering, Texas A&M University, College Station, Texas. His current research interests include phenotype classification under uncertainty.

Edward R. Dougherty received the PhD degree in mathematics from Rutgers University and the MS degree in computer science from Stevens Institute of Technology, and has been awarded the Doctor Honoris Causa by the Tampere University of Technology in Finland. He is a professor in the Department of Electrical and Computer Engineering, Texas A&M University in College Station, Texas, where he holds the Robert M. Kennedy 26 Chair in electrical engineering and is the director of the Genomic Signal Processing Laboratory. He is also the codirector of the Computational Biology Division of the Translational Genomics Research Institute in Phoenix, Arizona. He is a fellow of both the IEEE and SPIE, has received the SPIE Presidents Award, and served as the Editor of the SPIE/IS&T Journal of Electronic Imaging. At Texas A&M University he has received the Association of Former Students Distinguished Achievement Award in Research, been named Fellow of the Texas Engineering Experiment Station, and been named Halliburton Professor of the Dwight Look College of Engineering. He is the author of 16 books, editor of 5 others, and author of 300 journal papers.

For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.