Supplementary materials for "Drug susceptibility prediction against a panel of drugs using kernelized Bayesian multitask learning"

Mehmet Gönen and Adam A. Margolin

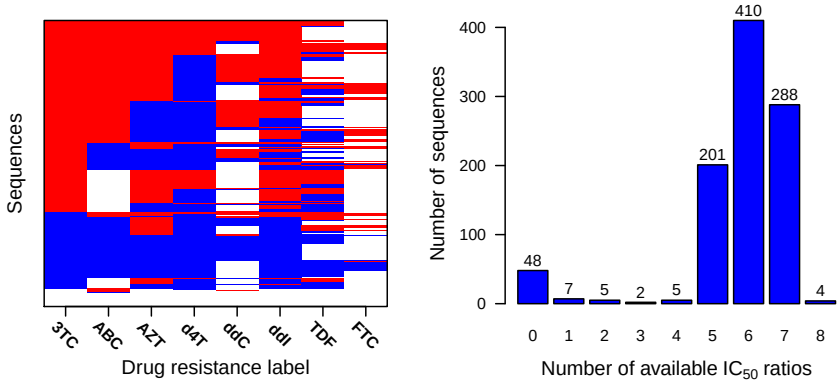

Figure 1: Drug resistance labels and histogram of available IC_{50} ratios for 970 reverse transcriptase sequences. For drug resistance labels, red: resistant; blue: susceptible; white: missing

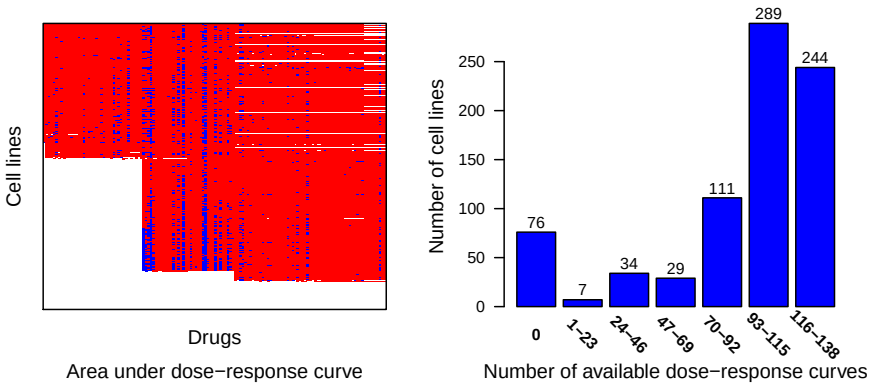

Figure 2: AUC values and histogram of available dose-response curves for 790 cell lines. For AUC values, red: higher; blue: lower; white: missing

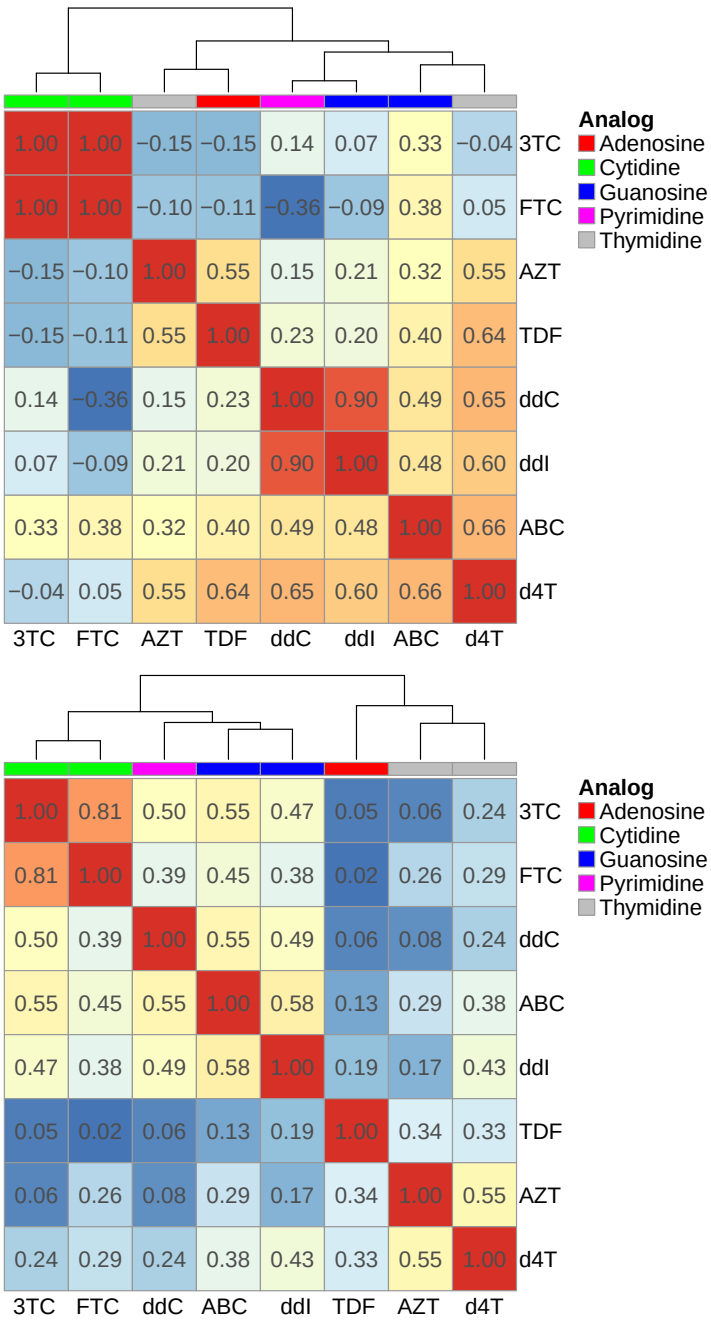

Figure 3: Hierarchical clustering of HIV-1 drugs based on correlation matrices. Top: Using the correlation matrix calculated over the measured IC_{50} ratios. Bottom: Using the correlation matrix calculated over the task-specific classification parameters found by kernelized Bayesian multitask learning

1 Inference details of kernelized Bayesian multitask learning

The approximate posterior distributions of the precision priors can be updated as

$$\alpha(\lambda_s^i) = \alpha_\lambda + 1/2$$
$$\beta(\lambda_s^i) = \big(1/\beta_\lambda + \big\langle (a_s^i)^2 \big\rangle / 2\big)^{-1},$$

where $\langle g(\cdot) \rangle$ denotes the posterior expectation as usual, i.e. $\mathrm{E}_{q(\cdot)}[g(\cdot)]$.

The approximate posterior distribution of the projection matrix can be updated as

$$\Sigma(\boldsymbol{a}_s) = \big(\mathrm{diag}(\langle \boldsymbol{\lambda}_s \rangle) + \mathbf{K}\mathbf{K}^\top / \sigma_h^2\big)^{-1}$$
$$\mu(\boldsymbol{a}_s) = \Sigma(\boldsymbol{a}_s)\big(\mathbf{K}\big\langle (\boldsymbol{h}^s)^\top \big\rangle / \sigma_h^2\big),$$

and the approximate posterior distribution of the hidden representation for each data point can be updated as

$$\Sigma(\boldsymbol{h}_i) = \left(\mathbf{I}/\sigma_h^2 + \sum_{t \in \mathcal{T}_i} \big\langle \boldsymbol{w}_t \boldsymbol{w}_t^\top \big\rangle\right)^{-1}$$
$$\mu(\boldsymbol{h}_i) = \Sigma(\boldsymbol{h}_i)\left(\big\langle \mathbf{A}^\top \big\rangle \boldsymbol{k}_i / \sigma_h^2 + \sum_{t \in \mathcal{T}_i} \Big[\langle f_{t,i} \rangle \langle \boldsymbol{w}_t \rangle\Big]\right),$$

where $\mathcal{T}_i$ gives the indices of tasks with given class labels for data point i. Note that, for each data point, we use the tasks with given class labels only to update the shared hidden representation.

The approximate posterior distribution of the classification parameters for each task can be updated as

$$\Sigma(\boldsymbol{w}_t) = \left(\mathbf{I}/\sigma_w^2 + \sum_{i \in \mathcal{I}_t} \Big\langle \boldsymbol{h}_i \boldsymbol{h}_i^\top \Big\rangle\right)^{-1}$$
$$\mu(\boldsymbol{w}_t) = \Sigma(\boldsymbol{w}_t) \sum_{i \in \mathcal{I}_t} \Big[\langle f_{t,i} \rangle \langle \boldsymbol{h}_i \rangle\Big],$$

where it can be seen that the inference mechanism transfers information between the tasks because the task-specific classification parameters are updated using the shared hidden representations.

The approximate posterior distribution of the predicted outputs can be updated as

$$\Sigma(f_{t,i}) = 1$$
$$\mu(f_{t,i}) = \big\langle \boldsymbol{w}_t^\top \big\rangle \langle \boldsymbol{h}_i \rangle$$
$$\rho(f_{t,i}) \triangleq f_{t,i} y_{t,i} > \nu,$$

where we can fortunately calculate the expectation of the truncated normal distribution in closed-form.

The inference procedure summarized in Algorithm **??** sequentially updates the approximate posterior distributions of the priors, latent variables and model parameters until convergence, which can be checked by monitoring the lower bound. The first term of the lower bound corresponds to the sum of exponential form expectations of the distributions in the joint likelihood. The second term is the sum of negative entropies of the approximate posteriors in the ensemble. The only nonstandard distribution in these terms is the truncated normal distribution used for the predicted outputs; nevertheless, the truncated normal distribution has a closed-form formula also for its entropy.

Algorithm 1 Kernelized Bayesian Multitask Learning (KBMTL)

Require: $\mathbf{K}$, $\{\boldsymbol{y}_t\}_{t=1}^T$, R, α_λ, β_λ, σ_h, σ_w and ν
1: Initialize $q(\mathbf{A})$, $q(\mathbf{H})$ and $\{q(\boldsymbol{w}_t), q(\boldsymbol{f}_t)\}_{t=1}^T$ randomly
2: **repeat**
3: Update $q(\mathbf{\Lambda})$, $q(\mathbf{A})$ and $q(\mathbf{H})$
4: Update $\{q(\boldsymbol{w}_t), q(\boldsymbol{f}_t)\}_{t=1}^T$
5: **until** convergence
6: **return** $q(\mathbf{A})$ and $\{q(\boldsymbol{w}_t)\}_{t=1}^T$

2 Extension to regression problems

In regression problems, for each task, we are given an output vector $\boldsymbol{y}_t = \{y_{t,i} \in \mathbb{R}\}_{i \in \mathcal{I}_t}$ instead of a label vector. The binary classification part of our method is replaced with the following distributional assumptions:

$$\begin{aligned} w_{t,s} &\sim \mathcal{N}(w_{t,s}; 0, \sigma_w^2) && \forall (t,s) \\ \epsilon_t &\sim \mathcal{G}(\epsilon_t; \alpha_\epsilon, \beta_\epsilon) && \forall t \\ y_{t,i} | \boldsymbol{h}_i, \boldsymbol{w}_t, \epsilon_t &\sim \mathcal{N}(y_{t,i}; \boldsymbol{w}_t^\top \boldsymbol{h}_i, \epsilon_t^{-1}) && \forall (t, i \in \mathcal{I}_t), \end{aligned}$$

where we do not need to use a bias term by centering the output vector to zero mean. We again used a variational approximation for inference, and the factorable ensemble approximation of the required posterior becomes

$$p(\mathbf{\Theta} | \mathbf{K}, \{\boldsymbol{y}_t\}_{t=1}^T) \approx q(\mathbf{\Theta}) = q(\mathbf{\Lambda}) q(\mathbf{A}) q(\mathbf{H}) \prod_{t=1}^T \big[q(\boldsymbol{w}_t) q(\epsilon_t) \big],$$

where $q(\mathbf{\Lambda})$, $q(\mathbf{A})$, $q(\mathbf{H})$ and $q(\boldsymbol{w}_t)$ remain intact. One additional factor in the ensemble is defined as

$$q(\epsilon_t) = \mathcal{G}(\epsilon_t; \alpha(\epsilon_t), \beta(\epsilon_t)).$$

The approximate posterior distribution of the regression parameters for each task can be updated as

$$\begin{aligned} \Sigma(\boldsymbol{w}_t) &= \left(\mathbf{I}/\sigma_w^2 + \sum_{i \in \mathcal{I}_t} \langle \epsilon_t \rangle \left\langle \boldsymbol{h}_i \boldsymbol{h}_i^\top \right\rangle \right)^{-1} \\ \mu(\boldsymbol{w}_t) &= \Sigma(\boldsymbol{w}_t) \sum_{i \in \mathcal{I}_t} \Big[\langle \epsilon_t \rangle \langle f_{t,i} \rangle \langle \boldsymbol{h}_i \rangle \Big], \end{aligned}$$

and the approximate posterior distribution of the additional precision priors can be updated as

$$\alpha(\epsilon_t) = \alpha_\epsilon + N_t/2$$
$$\beta(\epsilon_t) = \left(1/\beta_\epsilon + \sum_{i \in \mathcal{I}_t} \langle (y_{t,i} - \boldsymbol{w}_t \boldsymbol{h}_i)^2 \rangle / 2\right)^{-1},$$

where other update equations for $q(\boldsymbol{\Lambda})$, $q(\mathbf{A})$ and $q(\mathbf{H})$ are not changed.

3 Baseline algorithms

3.1 Bayesian probit classifier

Its distributional assumptions are defined as

$$\begin{aligned}
\gamma &\sim \mathcal{G}(\gamma; \alpha_\gamma, \beta_\gamma) \\
b|\gamma &\sim \mathcal{N}(b; 0, \gamma^{-1}) \\
\eta_f &\sim \mathcal{G}(\eta_f; \alpha_\eta, \beta_\eta) && \forall f \\
w_f|\eta_f &\sim \mathcal{N}(w_f; 0, \eta_f^{-1}) && \forall f \\
f_i|b, \boldsymbol{w}, \boldsymbol{x}_i &\sim \mathcal{N}(f_i; \boldsymbol{w}^\top \boldsymbol{x}_i + b, 1) && \forall i \\
y_i|f_i &\sim \delta(f_i y_i > \nu) && \forall i,
\end{aligned}$$

where the predicted outputs of data points are modeled as a linear function of input features (i.e. $\boldsymbol{w}^\top \boldsymbol{x}_i + b$). We learn the priors $\{\gamma, \boldsymbol{\eta}\}$, latent variables $\boldsymbol{f}$ and model parameters $\{b, \boldsymbol{w}\}$ using a deterministic variational approximation as we do for our method.

3.2 Bayesian relevance vector machine

Its distributional assumptions are defined as

$$\begin{aligned}
\gamma &\sim \mathcal{G}(\gamma; \alpha_\gamma, \beta_\gamma) \\
b|\gamma &\sim \mathcal{N}(b; 0, \gamma^{-1}) \\
\lambda_i &\sim \mathcal{G}(\lambda_i; \alpha_\lambda, \beta_\lambda) && \forall i \\
a_i|\lambda_i &\sim \mathcal{N}(a_i; 0, \lambda_i^{-1}) && \forall i \\
f_i|\boldsymbol{a}, b, \boldsymbol{k}_i &\sim \mathcal{N}(f_i; \boldsymbol{a}^\top \boldsymbol{k}_i + b, 1) && \forall i \\
y_i|f_i &\sim \delta(f_i y_i > \nu) && \forall i,
\end{aligned}$$

where the predicted outputs of data points are modeled as a linear function of their kernel representations (i.e. $\boldsymbol{a}^\top \boldsymbol{k}_i + b$). We again learn the priors $\{\gamma, \boldsymbol{\lambda}\}$, latent variables $\boldsymbol{f}$ and model parameters $\{\boldsymbol{a}, b\}$ using a deterministic variational approximation as we do for our method.