MULTITASK FEATURE SELECTION WITH TASK DESCRIPTORS

VÍCTOR BELLÓN*, VÉRONIQUE STOVEN AND CHLOÉ-AGATHE AZENCOTT

MINES ParisTech, PSL-Research University, CBIO-Centre for Computational Biology,
35 rue St Honoré 77300 Fontainebleau, France
Institut Curie, 75248 Paris Cedex 05,France
INSERM U900, 75248 Paris Cedex 05, France
*victor.bellon@mines-paristech.fr**

Machine learning applications in precision medicine are severly limited by the scarcity of data to learn from. Indeed, training data often contains many more features than samples. To alleviate the resulting statistical issues, the multitask learning framework proposes to learn different but related tasks joinlty, rather than independently, by sharing information between these tasks. Within this framework, the joint regularization of model parameters results in models with few non-zero coefficients and that share similar sparsity patterns. We propose a new regularized multitask approach that incorporates task descriptors, hence modulating the amount of information shared between tasks according to their similarity. We show on simulated data that this method outperforms other multitask feature selection approaches, particularly in the case of scarce data. In addition, we demonstrate on peptide MHC-I binding data the ability of the proposed approach to make predictions for new tasks for which no training data is available.

1. Introduction

A substantial limiting factor for many machine learning applications in bioinformatics is the scarcity of training data. This issue is particularly critical in precision medicine applications, which revolve around the analysis of considerable amounts of high-throughput data, aiming at identifying the similarities between the genomes of patients who exhibit similar disease susceptibilities, prognoses, responses to treatment, or immune responses to vaccines. In such applications, collecting large numbers of samples is often costly. It is therefore frequent for the number of samples (n) to be orders of magnitudes smaller than the number of features (p) describing the data. Model estimation in such $n \ll p$ settings is a major challenge of modern statistics, and the risk of overfitting the training data is high.

Fortunately, it is often the case that data is available for several related but different problems (or tasks). While such data cannot be pooled together to form a single, large data set, the *multitask* framework makes it possible to leverage all the available information to learn related but separate models for each of these problems. For example, genetic data may be available for patients who were included and followed under different but related conditions. If each condition is considered separately, we may not have enough data to detect the relevant genetic variations associated to the trait under study. Multitask learning approaches where each condition corresponds to a task can be used to circumvent this issue by increasing the number of learning examples while keeping the specificity of each dataset[1,2]. Another prevalent strategy to avoid overfitting the training data is to apply *regularization*, that is to say, to impose a penalization on the complexity of the model. One of the most common penalizations takes the form of an l_1-norm over the weights assigned to the features. In the context of least-squares regression, this is known as the Lasso[3]. This approach drives many of the regression weights to

0, resulting in sparse models, that is to say, models that involve a small number of predictors. This makes them particularly suitable for biological applications, where it is often desirable for models to not only exhibit good predictive abilities, but also to be interpretable. For example, if samples are patients encoded by genetic features, if only a small number of features are selected by the model (i.e. are assigned non-zero weights), it may be possible to relate these features to the biological pathways involved in the predicted trait. Further down the line, these features can be used to aid diagnosis or design companion tests. However, l_1-regularized methods are sensitive to small perturbations of the data, and it is therefore necessary to pay attention to their stability.

The Multitask Lasso[4] was the first approach to apply regularization in the multitask setting. It employs a block regularization over the weights that imposes to select the same features for all tasks. The coefficients assigned to these features are allowed to vary smoothly, resulting in separate models for the separate tasks. However, many problems require more flexibility. Indeed, since the tasks considered in multitasks approaches are related, but not identical, we can expect some sharper variation in the degree to which the selected features are relevant for the different tasks.

In line with this idea, the Multi-level Multitask Lasso[5] expresses each regression coefficient as the product of two factors. One factor controls the overall sparsity and captures features common to all tasks; the second factor modulates the weights of the selected features, reflecting the task specificities.

These approaches have two limitations. First, they cannot be directly applied to make predictions for new tasks for which no training data is available. This could be relevant to predict the cytotoxicity of a new drug on cells or patients, or to evaluate the prognosis of a previously unseen cancer subtype. Second, the degree of similarity between tasks is not explicitly taken into account. However, intuitively, we would like to explicitly enforce that more information should be shared between more similar tasks.

These two limitations can be addressed by defining an explicit representation of the tasks. This provides a convenient way to relate tasks and to share information between them, as is done in kernel methods[6,7]. Based on the intuition that the second factor of the MML[5] should be similar for similar tasks, we propose to characterize each task by a set of descriptor variables and re-write this factor as a linear combination of these descriptor variables.

In this paper, we start by formulating the multitask least-squares regression problem and by presenting the state of the art. We then introduce our model, give a result on the asymptotic convergence of the estimator, and present an algorithm for solving the optimization problem. Experimental results on simulated data show our approach to be competitive both in terms of prediction error and in terms of the quality of the selected features. Finally, we illustrate the validity of the proposed method for the prediction of new tasks by applying it to MHC-I binding prediction, a problem relevant to the design of personalized vaccines.

2. State of the art

In this section we present existing approaches to the problem of multitask feature selection.

2.1. *Problem formulation*

Let us assume that we want to learn K different tasks, corresponding to K datasets $(X^k, Y^k)_{k=1,\ldots,K}$. Let $X^k \in \mathbb{R}^{n_k \times p}$ be the data matrix containing n_k instances of dimension p, and $Y^k \in \mathbb{R}^{n_k}$ the corresponding real-valued output data. Our objective is to find, for every $k = 1, \ldots, K$ and for every $i = 1, \ldots, n_k$, $\beta \in \mathbb{R}^{K \times p}$ such that

$$y_i^k = f\left(x_i^k\right) + \epsilon_i^k = \sum_{j=1}^{p} \beta_j^k x_{ij}^k + \epsilon_i^k,$$

where ϵ_i^k is the noise for the i-th instance of task k. For each feature j, β_j is a K-dimensional vector of weights assigned to this feature for each task. Direct minimization of the loss between Y and f would be equivalent to fitting K different linear regressions in a single step. Therefore, this formulation does not allow to share information across tasks.

2.2. *Multitask Lasso and Sparse Multitask Lasso*

One of the first formulations for the joint selection of features across related tasks, commonly referred to as Multitask Lasso[4] (ML), uses a method related to the Group Lasso[8]. Information is shared between tasks through a regularization term: An l_2-norm forces the weights β_j of each feature to shrink across tasks, and an l_1-norm over these l_2-norms produces a sparsity pattern common to all tasks. These penalties produce patterns where every task is explained by the same features. This results in the following optimization problem:

$$\min_{\beta \in \mathbb{R}^{K \times p}} \frac{1}{2} \sum_{k=1}^{K} \frac{1}{n_k} \sum_{i=1}^{n_k} \left(y_i^k - \sum_{j=1}^{p} \beta_j^k x_{ij}^k \right)^2 + \lambda \sum_{j=1}^{p} \|\beta_j\|_2, \tag{1}$$

A common extension of this problem is the Sparse Multitask Lasso (SL), based on the Sparse Group Lasso[9]. It consists in adding the regularization term $\lambda_s \|\beta\|_1$ to Equation 1, which generates a sparse structure both on the features as well as between tasks. These sparse optimization problems have been well studied and can be solved using proximal optimization[10].

2.3. *Multi-level Multitask Lasso*

To allow for more flexibility in the sparsity patterns of the different tasks, the authors of the Multi-level Lasso[5] (MML) propose to decompose the regression parameter β into a product of two components $\theta \in \mathbb{R}^p$ and $\gamma \in \mathbb{R}^{K \times p}$. The intuition here is to capture the global effect of the features across all the tasks with θ, while γ provides some modulation according to the specific sensitivity of each task to each feature. This results in the following optimization problem:

$$\min_{\theta \in \mathbb{R}^p, \gamma \in \mathbb{R}^{K \times p}} \frac{1}{2} \sum_{k=1}^{K} \frac{1}{n_k} \sum_{i=1}^{n_k} \left(y_i^k - \sum_{j=1}^{p} \theta_j \gamma_j^k x_{ij}^k \right)^2 + \lambda_1 \sum_{j=1}^{p} |\theta_j| + \lambda_2 \sum_{k=1}^{K} \sum_{j=1}^{p} |\gamma_j^k| \tag{2}$$

with the constraint that $\theta > 0$.

The authors prove that this approach generates sparser patterns than the so-called Dirty model[11], where the β parameter is decomposed into the sum (rather than product) of two parameters. In practice, this model also gives sparser representations than the ML, and has the advantage not to impose to select the exact same features across all tasks.

The optimization of the parameters is a non-convex problem that can be decomposed in two alternate convex optimizations. Furthermore, the optimal θ can be calculated exactly given γ.[12] This optimization, however, is much slower than that of the ML. Finally, note that in this approach, the multitask character is explicitly provided by the parameter θ, which is shared across all tasks, rather than implicitly enforced by a penalization term.

3. Multiplicative Multitask Lasso with Task Descriptors

The approaches presented above do not explicitly model relations between tasks. However, an explicit representation of the task space might be available. Inspired by kernel approaches, where task similarities are encoded in the model[6,7], we introduce a new model called Multiplicative Multitask Lasso with Task Descriptors (MMLD), where we use a vector of task descriptor variables to encode each task, and to explain the specific effect modulating each feature for each task.

Following the MML formulation[5], we decompose the parameter β into a product of two components. We keep the notation θ for the first component, which corresponds to the global feature importance common to all tasks. The second component is now a linear combination of the L-dimensional task descriptors $D \in \mathbb{R}^{L\times K}$. The L task descriptors have to be defined beforehand and depend on the application. For example, if the different tasks are sensitivity to different drugs to which cell lines are exposed, one could use molecular fingerprints[13] to describe the drugs, i.e. the tasks. The regression parameter $\alpha \in \mathbb{R}^{p\times L}$ indicates the importance of each descriptor for each feature, and controls the specificity of each task. Hence we formulate the following optimization problem:

$$\min_{\theta\geq 0, \alpha\in\mathbb{R}^{p\times L}} \frac{1}{2}\sum_{k=1}^{K}\frac{1}{n_k}\sum_{i=1}^{n_k}\left(y_i^k - \sum_{j=1}^{p}\theta_j\left(\sum_{l=1}^{L}\alpha_{jl}d_l^k\right)x_{ij}^k\right)^2 + \lambda_1\sum_{j=1}^{p}|\theta_j| + \lambda_2\sum_{j=1}^{p}\sum_{l=1}^{L}|\alpha_{jl}|, \qquad (3)$$

where λ_1 and λ_2 are the regularization parameters for each component of β.

Importantly, because predictions for a new data point x are made as $\sum_{j=1}^{p}\theta_j(\sum_{l=1}^{L}\alpha_{jl}d_l^k)x_{ij}$, this formulation allows to make predictions for tasks for which no training data is available: the only task-dependent parameters are the descriptors d_l^k. This ability to extrapolate to new tasks is not shared by the existing multitask Lasso methods.

3.1. *Theoretical guaranties*

Let us define, for all $k = 1,\dots,K$, $i = 1,\dots,n_k$, $j = 1,\dots,p$, $l = 1,\dots,L$, $\xi_{ijl}^k = d_l^k x_{ij}^k$ and $\mu_{jl} = \theta_j\alpha_{jl}$. Problem 3 can be reformulated as

$$\min_{\theta\geq 0, \mu\in\mathbb{R}^{p\times L}} \frac{1}{2}\sum_{k=1}^{K}\frac{1}{n_k}\sum_{i=1}^{n_k}\left(y_i^k - \sum_{j=1}^{p}\sum_{l=1}^{L}\mu_{jl}\xi_{ijl}^k\right)^2 + \lambda_1\|\theta\|_1 + \lambda_2\sum_{j=1}^{p}\theta_j^{-1}\|\mu_j\|_1 \qquad (4)$$

Following Lemma 1 in Ref. 12, it is immediate to prove that, when $\omega = 2\sqrt{\lambda_1\lambda_2}$, Problem 4 is equivalent to

$$\min_{\mu\in\mathbb{R}^{p\times L}} \frac{1}{2}\sum_{k=1}^{K}\frac{1}{n_k}\sum_{i=1}^{n_k}\left(y_i^k - \sum_{j=1}^{p}\sum_{l=1}^{L}\mu_{jl}\xi_{ijl}^k\right)^2 + \omega\sum_{j=1}^{p}\sqrt{\|\mu_j\|_1}, \tag{5}$$

with $\hat{\theta}_j = \sqrt{\frac{\lambda_1}{\lambda_2}\|\mu_j\|_1}$. Problem 5 has a convex loss function and a non-convex regularization term. The characterization of the asymptotic distribution of the estimator for this problem, as well as its $\sqrt{n}$-consistency, have been previously given by Lozano and Swirszcz[5], based on a more general result[14].

3.2. *Algorithm*

Problem 3 is non-convex. We therefore propose to adapt the algorithm of Ref. 5 and separate it in alternate convex optimization steps: the optimization of θ for a fixed α, corresponding to a nonnegative Garrote problem[15], and the optimization of α for a gixed θ, corresponding to a Lasso optimization[3]. Details can be found in Algorithm 3.1. Python code is available at: `https://github.com/vmolina/MultitaskDescriptor`

Algorithm 3.1.

```
Input $\{X^k, Y^k, D^k\}_{k=1,\dots,K}$, $\lambda_1, \lambda_2, \epsilon, m_{max}$.
Define $n = \sum_{k=1}^K n^k$, $\tilde{X} = \left\{x_1^1, \dots, x_{n^1}^1, x_1^2, \dots, x_{n^k}^k\right\}$ and $\tilde{Y} = \left\{y_1^1, \dots, y_{n^1}^1, y_1^2, \dots, y_{n^k}^k\right\}$
Initialize $\theta_j(0) = 1$ and $\alpha_j(0)$ according to an initial estimate, for $j$ = 1,...,p.
For $m = 1, \dots m_{max}$:
    Solve for $\alpha$:
        $w_{ijl}(m) = \theta_j(m-1)d_{il}\tilde{x}_{ij}$.
        $\alpha(m) = \arg\min_\alpha \frac{1}{2}\sum_{i=1}^n \left(\tilde{y}_i - \sum_{j=1}^p\sum_{l=1}^L \alpha_{jl}w_{ijl}(m)\right)^2 + \lambda_2\sum_{j=1}^p\sum_{l=1}^L|\alpha_{jl}|$
    Solve for $\theta$:
        $z_{*j}(m) = \left[\sum_{l=1}^L \alpha_{jl}(m)d_l^1x_{1j}^1, \dots, \sum_{l=1}^L \alpha_{jl}(m)d_l^k x_{n^kj}^k\right]$, for $j = 1, \dots p$
        $\theta(m) = \arg\min_{\theta\geq 0} \frac{1}{2}\sum_{i=1}^n \left(\tilde{y}_i - \sum_{j=1}^p \theta_j(m-1)z_{ij}(m)\right)^2 + \lambda_1\sum_{j=1}^p|\theta_j(m-1)|$
    $\beta_j^k(m) = \theta_j(m)\sum_{l=1}^L \alpha_{jl}(m)d_l^k$
    If $R(\beta(m-1)) - R(\beta(m)) \leq \epsilon$ (where $R(\beta)$ denote the squared loss over all tasks)
        Break
Return $\beta(m)$
```

4. Experiments on simulated data

In this section, we compare our method to the ML, the SL and the MML based on two different criteria. First, we compare them in terms of the *quality* of the selected features. By quality, we mean the ability to recover the true support of β (that is to say, its non-zero entries), as well as the stability of the selection upon data perturbation. Second, we evaluate the methods in terms of prediction performance.

4.1. *Simulated data*

We simulate K design matrices $X_k \in \mathbb{R}^{n_k \times p}$ according to a Gaussian distribution with mean 0 and a precision matrix $\Sigma \sim Wishart(p+20, I_p)$, where I_p is the identity matrix of dimension p. In our simulations $n_1 = n_2 = \ldots = n_K$. For each task k, we sample L descriptors d_l^k from a normal distribution with mean $\mu_{d_l} \sim \mathcal{N}(0,5)$ and variance $\sigma^2_{d_l} \sim \mathcal{G}amma(0.2,1)$. We build θ by randomly selecting $p_s < p$ indices for non-zero coefficients, which we sample from a Gamma distribution $\mathcal{G}amma(1,2)$. All other entries of θ are set to 0. We build α in the following manner: For each of the non-zero θ_j, we randomly select $L_s < L$ entries of α_j to be non-zero, and sample them from a Gaussian distribution $\mathcal{N}(0,2)$. All other α_{jl} are set to 0.

We then compute $\beta_j^k = \theta_j \left(\sum_{l=1}^{L} \alpha_{jl} D_l^k\right)$ and normalize it by dividing by $\beta^* = \max_{j,k} |\beta_j^k|$. Finally, we randomly chose with replacement S_s entries of β_k. If the chosen entry is different from 0, we set it to 0; conversely, if it was equal to 0, we set it to a new value sampled from a Gaussian distribution $\mathcal{N}(0,0.5)$. This last randomization step is performed to relax the structure of β. Finally, we simulate $Y = \beta X + \epsilon$ where ϵ is Gaussian noise with $\sigma^2 = 0.1$.

Each of our experiments consists in evaluating the different models in a 10-fold cross-validation. We create a first set of experiments containing 5 datasets generated with the parameters $K = 4$, $n_k = 100$, $p = 100$, $L = 10$, $p_s = 20$, $L_s = 4$, and $S_s = 100$. We generate a second set of experiments using $n_k = 20$ to simulate a scarce setting. We report the results of additional experiments in a scarcer setting ($p = 8000$, $n_k = 20$) in the Supplementary Materials.

In each experiment we train 4 different models: the ML[4], the SL[9], the MML[5], and the MMLD we propose here. In order to better understand the role of the task descriptor space, we use 3 variants of the MMLD: one that uses the same task descriptors as those from which the data was generated; one that uses these descriptors, perturbed with Gaussian noise ($\sigma = 0.1$); and one with a random set of task descriptors, sampled from a uniform distribution over $[0,1]$. Perturbing the task descriptors with more noise should give results in between those obtained in those last two scenarios.

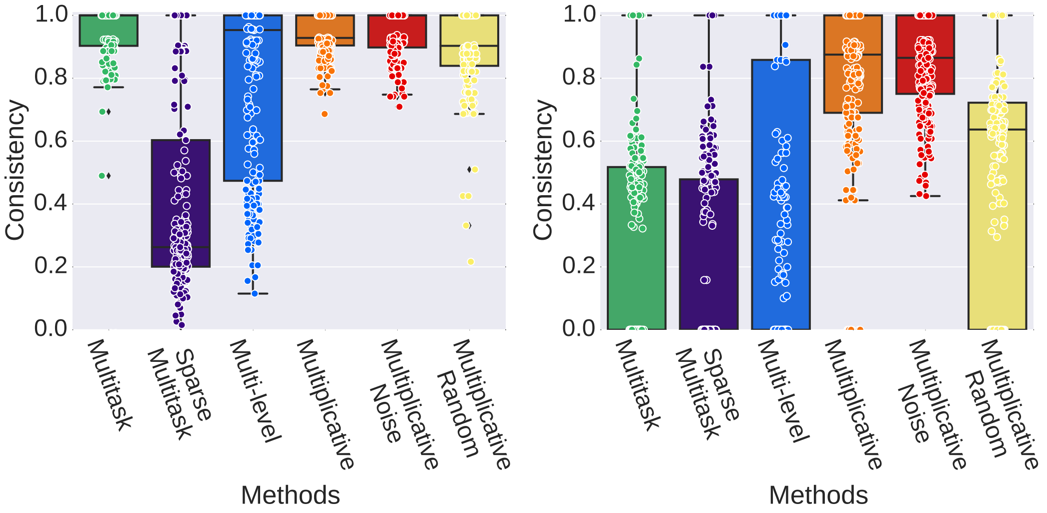
Each of these 6 methods estimates a real-valued matrix $\hat{\beta} \in \mathbb{R}^{K \times p}$. We then consider as selected, for a given task k, the features j for which $\hat{\beta}_j^k$ is different from 0. For all methods, λ is set by cross-validation: Let $\lambda_{\min}$ be the value of λ that yields the lowest cross-validated RMSE $E_{\min}$. Then, we pick, amond all $\lambda > \lambda_{\min}$ resulting in a cross-validated RMSE less than one standard deviation away from E_min, the λ that yields the median cross-validated RMSE. This heuristic compromises between optimizing for RMSE and imposing more regularization.

4.2. *Feature selection and stability*

In this section, we evaluate the ability of the feature selection procedure to select the correct features, as well as the stability of the procedure, on two sets of experiments.

Stability of the feature selection In precision medicine applications, it is often critical that feature selection methods be stable: If a method selects different features under small perturbations, we cannot rely on it to identify biologically meaningful features. To evaluate the stability of the feature selection procedures, we calculate the consistency index[16] between the sets of features selected over each fold.

Figure 1(a) shows the consistencies of the different methods for the first set of experiments. We observe that the consistency of the feature selection for the proposed method is much higher than the consistency of SL and MML. By contrast, ML presents a very high consistency index, that decays when the data is scarcer. (Fig. 1(b)). The addition of small noise to the task descriptors does not have a strong effect on the stability of the selection, using random task descriptors negatively affects it, especially when data is scarce. In an even scarcer scenario the consistency presents high variation for all methods (Supp. Mat.).



(a) $n_k = 100$ instances per task

(b) $n_k = 20$ instances per task

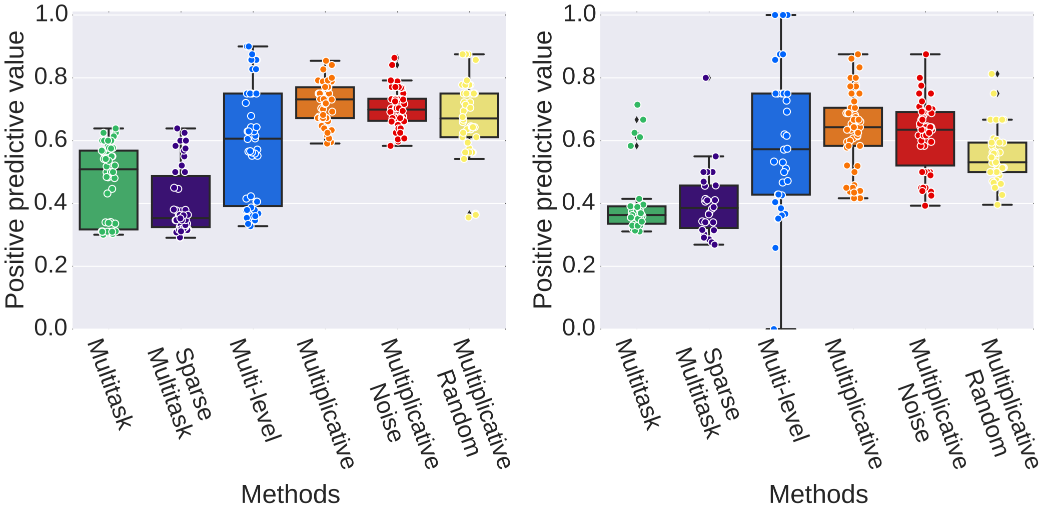
Fig. 1. Boxplot depiction of the consistency index of the different methods for simulated data.

Number of selected features We report in Table 1 the mean number of non-zero coefficients assigned by each method in each scenario. We evaluate sparsity at the level of the β coefficients, hence the total number of coefficients is $n_k \times K$. The ML and the SL both recover more features than all other methods. The MML chooses more features than the MMLD when $n_k = 100$, but selects fewer parameters when the number of instances is reduced. Finally, the MMLD presents a much lower variation in the number of selected features than all other methods.

Table 1. Mean number of non-zero coefficients assigned by each method.

	True	ML	SL	MML	MMLD	Noisy MMLD	Random MMLD
$n_k = 100$	126.8 ± 6.8	169.28 ± 163.4	231.62 ± 121.3	83.9 ± 80.7	54.88 ± 9.8	56.88 ± 11.2	49.12 ± 56.5
$n_k = 20$	126.8 ± 3.2	80.88 ± 79.8	43.96 ± 48.8	17.82 ± 21.7	46.24 ± 15.9	48.56 ± 18	46.72 ± 34.6

Ability to select the correct features We report the Positive Predictive Value (PPV, Fig. 2) and the sensitivity (Fig. 3) of the feature selection for the different methods. The PPV is the proportion of selected features that are correct. The sensitivity is the proportion of correct



(a) $n_k = 100$ instances per task

(b) $n_k = 20$ instances per task

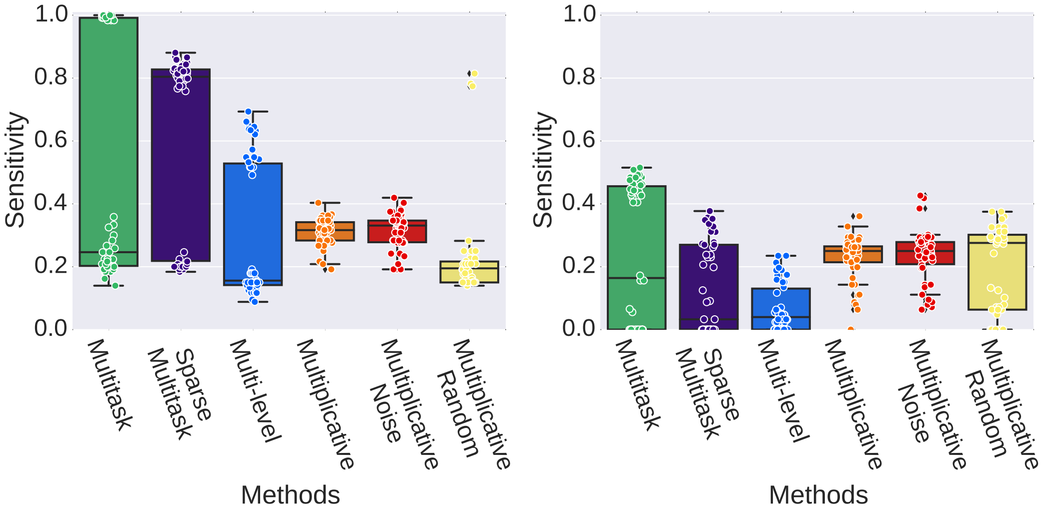
Fig. 2. Boxplot depiction of the positive predictive value of the different methods for simulated data.

While the MML outperforms the ML and the SL in terms of PPV (Fig. 2), its sensitivity is worse (Fig. 3). Indeed, the ML and the SL select many more features: this higher sensitivity comes to the price of a large number of false positives. By contrast, the proposed MMLD performs well according to both criteria. It clearly outperforms all other methods in terms of PPV (Fig. 2), even when using noisy descriptors. In the case of random descriptors, the performance is close to that of the MML, and more degraded when the data is scarce. In terms of sensitivity (Fig. 3), the MMLD also outperforms its competitors. We observe a higher variability in performance for these other methods, due to the higher variability in the number of features they select. The ML, SL and MML suffer greater losses in sensitivity than the proposed method when data is scarce. Using task descriptors hence seems to increase the robustness of the feature selection procedure. As would be expected, using random task descriptors negatively affects the ability of the MMLD to recover the correct features. Small perturbations of the task descriptors appear to have little effect on the quality of the selected features. We report similar results for the setting where $p = 8000$ (Supp. Mat.).

4.3. *Prediction error*

The other important criterion on which to evaluate the model we propose is the quality of the predictions it makes. Figure 4 presents the 10-fold cross-validated Root Mean Squared Error (RMSE) of the different methods, for both $n_k = 100$ and $n_k = 20$. We observe that the proposed method performs better than its competitors, even with perturbed task descriptors. According to a paired Wilcoxon signed rank test (Supp. Mat.), these differences in RMSE on scarce data are significant. Interestingly, this is true even in comparison with the ML and the SL, which select more features and could hence be expected to yield lower RMSEs.

This improvement in predictive performance is particularly visible in the scarce setting(Fig. 4). In addition, the variance of the RMSE of the MMLD remains stable when the

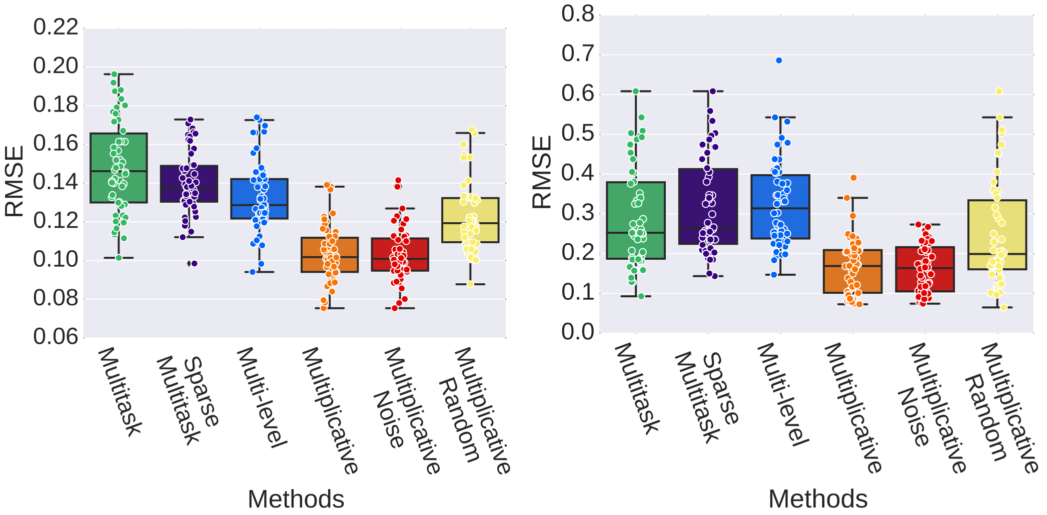


(a) $n_k = 100$ instances per task

(b) $n_k = 20$ instances per task

Fig. 3. Boxplot depiction of the sensitivity of the different methods for simulated data.

number of samples decreases, while it clearly increases for the other approaches. Once again, we report similar results for the setting where $n_k = 20$ and $p = 8000$ (Supp. Mat.)



(a) $n_k = 100$ instances per task

(b) $n_k = 20$ instances per task

Fig. 4. Boxplot of the 10-fold cross-validated Root Mean Squared Error (RMSE) of the different methods for simulated data. For readability, (a) and (b) are plotted on different scales.

5. Peptide-MHC-I binding prediction

The prediction of whether a peptide can bind to a given MHC-I (major histocompatibility complex class I) protein is an important tool for the development of peptide vaccines. MHC-I genes are highly polymorphic, and hence express proteins with diverse physico-chemical properties across individuals. The binding affinity of a peptide is thus going to depend on the MHC-I allele expressed by the patient. It is therefore important that predictions are allele-specific. This in turns opens the door to administering patient-specific vaccines.

While some MHC-I alleles have been well studied, others have few if any known binders. Sharing information across different alleles has the potential to improve the predictive accuracy of models. Indeed, the multitask framework, where different tasks correspond to different MHC-I proteins, has been previously shown to be beneficial for this problem[17,18]. In addition, it can be necessary in this context to make predictions for tasks (i.e. alleles) for which no training data is available.

5.1. *Data*

Following previous work[17], we test our model on three freely available benchmark datasets[19,20]. The data consists of pairs of peptide sequences and MHC-I alleles, labeled as binding or non-binding. Ref. 19 provides two datasets for the same 54 alleles, containing 1363 (resp. 282) positive and 1361 and (resp. 141784) negative examples. The dataset from Ref. 20 has 35 different alleles, 1548 positive examples and 4331 negative examples. As an example of an allele with few training data, allele B*57:01 in Ref. 20 only has 11 known binders.

The peptides are of length 9 and are classically represented by a 20-dimensional binary vector indicating which amino acid is present. While in this case $p < n$, this example allows us to evaluate the proposed method on real data, relevant for personalized medicine applications. Because the MHC-I alleles are much longer than that, we do not adopt the same representation and define task descriptors as follows: Using sequences extracted from the IMGT/HLA database[21], we keep only the amino acids located at positions involved in the binding sites of all three HLA superfamilies[17,22]. Inspired by the Linseq kernel[17], we then compute a similarity matrix between all alleles (tasks), based on the proportion of coincident amino acids at each position. We then perform a Principal Component Analysis on this matrix and keep the first 4 principal components, having observed that the structure of this matrix is not much perturbed by this dimensionality reduction. In the end, each task is represented by the 4-dimensional vector of its projections on each of these 4 components.

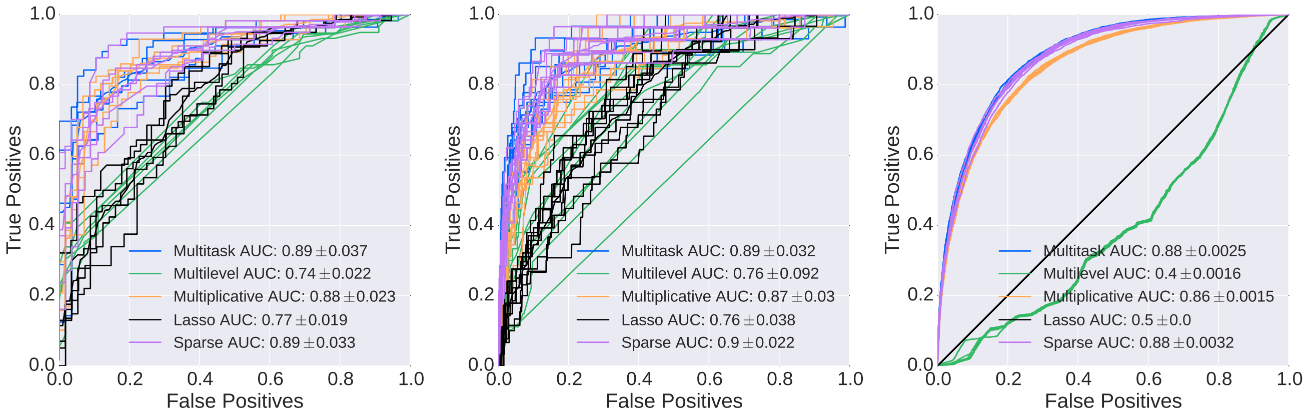
5.2. *Experiments*

We predict whether a peptide binds to a certain allele using the ML, the SL, the MML and the MMLD. Additionally, we compare these approaches to single task Lasso regressions.

We run cross-validation using the same folds as in the original publications[19,20]. The first Heckerman dataset[19] is divided in 5 folds and the second one in 10. Because this second dataset is highly unbalanced, we randomly keep only one negative example for each of the positive examples. The Peters dataset[20] is divided in 5 folds. We run an inner cross-validation to set the regularization parameters.

We show in Fig. 5.2 the Receiver Operator Curves (ROC) for the three datastets. Each curve corresponds to one fold. We additionally report the mean and standard deviation of the area under the ROC curve (ROC-AUC) for each approach. We observe that the ML, the SL and the MMLD perform comparatively, and consistently outperform the two other methods.

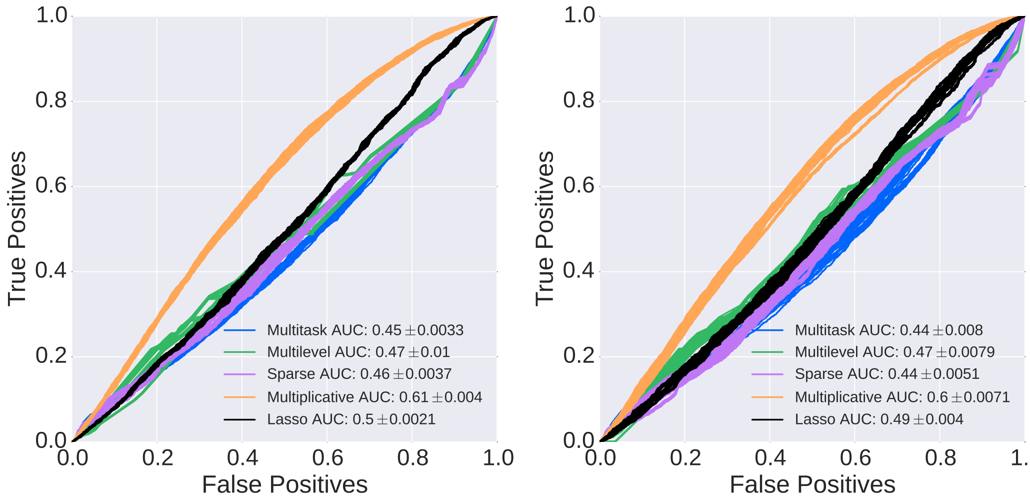
Furthermore, we evaluate the ability of the different methods to predict binding for alleles



(a) First Heckerman dataset (b) Second Heckerman dataset (c) Peters dataset

Fig. 5. Cross-validated ROC curves for the prediction of MHC-I binding.

for which no training data is available. For this purpose, we use the models previously trained on the folds of the two first datasets to predict on the folds of the third dataset. When predicting for a new task with the ML, the SL and the MML, we use the mean of the predictions made by all trained models. As can be seen in Fig. 6, the proposed method is the only one that outperforms the trivial baseline (ROC-AUC=0.5), hence illustrating its ability to make predictions for previously unseen tasks, by contrast with all other methods.



(a) Models trained on the first Heckerman dataset, evaluated on the Peters dataset

(b) Models trained on the second Heckerman dataset, evaluated on the Peters dataset

Fig. 6. ROC curves for the prediction of MHC-I binding, cross-dataset.

6. Conclusion

We have presented a novel approach for multitask least-squares regression. Our method extends the MML framework[5] to leverage task descriptors. This allows to tune how much information is shared between tasks according to their similarity, as well as to make predictions for new tasks. Multitask kernel methods[6,17,18] also allow to model relations between tasks, but do not offer the advantages of the Lasso framework in terms of sparsity and interpretability, which are key for biomedical applications.

Our experiments on simulated data show that the proposed method is more stable than other Lasso approaches. The features it selects are hence more reliable, and the resulting models more easily interpreted. In addition, true support recovery suffers less in scarce settings. Finally, the predictivity of the resulting models is competitive with that of other Lasso approaches. Unsurprisingly, performance deteriorates when task descriptors are inappropriate. However, neither the quality of the selected features nor the model predictivity suffer much from the addition of small noise to these descriptors. These results suggest that the MMLD approach we propose is well adapted to precision medicine applications, which require building stable, intepretable models from $n \ll p$ data.

Finally, our experiments on MHC-I peptide binding prediction illustrate that the method we propose is well-suited to making predictions for tasks for which no training data is available.

7. Supplementary Materials

`http://cazencott.info/dotclear/public/publications/Bellon_PSB2016_SuppMat.pdf`

Acknowledgments

This project has received funding from the European Unions Seventh Framework Programme for research, technological development and demonstration under grant agreement no. 316861.

References

1. K. Puniyani, S. Kim and E. P. Xing, *Bioinformatics* **26**, i208 (2010).
2. L. Chen, C. Li, S. Miller and F. Schenkel, *BMC Genetics* **15**, p. 53 (2014).
3. R. Tibshirani, *J Roy Stat Soc B Stat Meth* , 267 (1996).
4. G. Obozinski, B. Taskar and M. Jordan, *Statistics Department, UC Berkeley, Tech. Rep* (2006).
5. A. C. Lozano and G. Swirszcz, Multi-level lasso for sparse multi-task regression (2012).
6. T. Evgeniou, C. A. Micchelli and M. Pontil, *J Mach Learn Res* , 615 (2005).
7. E. V. Bonilla, K. M. Chai and C. Williams, *NIPS* **20**, 153 (2007).
8. M. Yuan and Y. Lin, *J Roy Stat Soc B Stat Meth* **68**, 49 (2006).
9. N. Simon, J. Friedman, T. Hastie and R. Tibshirani, *J Comput Graph Stat* **22**, 231 (2013).
10. Y. Nesterov, *Introductory Lectures on Convex Optimization* (Springer, 2004).
11. A. Jalali, S. Sanghavi, C. Ruan and P. K. Ravikumar, *NIPS* **23**, 964 (2010).
12. X. Wang, J. Bi, S. Yu and J. Sun, On multiplicative multitask feature learning (2014).
13. D. R. Flower, *J Chem Inform Comput Sci* **38**, 379 (1998).
14. G. V. Rocha, X. Wang and B. Yu, *arXiv preprint arXiv:0908.1940* (2009).
15. L. Breiman, *Technometrics* **37**, 373 (1995).
16. L. I. Kuncheva, *AIA* **25**, 421 (2007).
17. L. Jacob and J.-P. Vert, *Bioinformatics* **24**, 358 (2008).
18. C. Widmer, N. C. Toussaint, Y. Altun and G. Rätsch, *BMC Bioinformatics* **11**, p. S5 (2010).
19. D. Heckerman, C. Kadie and J. Listgarten, *J Comput Biol* **14**, 736 (2007).
20. B. Peters, H.-H. Bui, S. Frankild, M. Nielson *et al.*, *PLoS Comput Biol* **2**, p. e65 (2006).
21. J. Robinson, A. Malik, P. Parham, J. Bodmer and S. Marsh, *Tissue Antigens* **55**, 280 (2000).
22. I. A. Doytchinova, P. Guan and D. R. Flower, *J Immunol* **172**, 4314 (2004).

PERSONALIZED HYPOTHESIS TESTS FOR DETECTING MEDICATION RESPONSE IN PARKINSON DISEASE PATIENTS USING iPHONE SENSOR DATA

ELIAS CHAIBUB NETO*, BRIAN M. BOT, THANNEER PERUMAL, LARSSON OMBERG, JUSTIN GUINNEY, MIKE KELLEN, ARNO KLEIN, STEPHEN H. FRIEND, ANDREW D. TRISTER

Sage Bionetworks, 1100 Fairview Avenue North, Seattle, Washington 98109, USA
**corresponding author e-mail: elias.chaibub.neto@sagebase.org*

We propose hypothesis tests for detecting dopaminergic medication response in Parkinson disease patients, using longitudinal sensor data collected by smartphones. The processed data is composed of multiple features extracted from active tapping tasks performed by the participant on a daily basis, before and after medication, over several months. Each extracted feature corresponds to a time series of measurements annotated according to whether the measurement was taken before or after the patient has taken his/her medication. Even though the data is longitudinal in nature, we show that simple hypothesis tests for detecting medication response, which ignore the serial correlation structure of the data, are still statistically valid, showing type I error rates at the nominal level. We propose two distinct personalized testing approaches. In the first, we combine multiple feature-specific tests into a single union-intersection test. In the second, we construct personalized classifiers of the before/after medication labels using all the extracted features of a given participant, and test the null hypothesis that the area under the receiver operating characteristic curve of the classifier is equal to 1/2. We compare the statistical power of the personalized classifier tests and personalized union-intersection tests in a simulation study, and illustrate the performance of the proposed tests using data from mPower Parkinsons disease study, recently launched as part of Apples ResearchKit mobile platform. Our results suggest that the personalized tests, which ignore the longitudinal aspect of the data, can perform well in real data analyses, suggesting they might be used as a sound baseline approach, to which more sophisticated methods can be compared to.

Keywords: personalized medicine, hypothesis tests, sensor data, remote monitoring, Parkinson

1. Introduction

Parkinson disease is a severe neurodegenerative disorder of the central nervous system caused by the death of dopamine-generating cells in the midbrain. The disease has considerable worldwide morbidity and is associated with substantial decrease in the quality of life of the patients (and their caregivers), decreased life expectancy, and high costs related to care. Early symptoms in the motor domain include shaking, rigidity, slowness of movement and difficulty for walking. Later symptoms include issues with sleeping, thinking and behavioral problems, depression, and finally dementia in the more advanced stages of the disease. Treatments are usually based on levodopa and dopamine agonist medications. Nonetheless, as the disease progresses, these drugs often become less effective, while still causing side effects, including involuntary twisting movements (dyskinesias). Statistical approaches aiming to determine if a given patient responds to medication have key practical importance as they can help the physician in making more informed treatment recommendations for a particular patient.

In this paper we propose personalized hypothesis tests for detecting medication response in Parkinson patients, using longitudinal sensor data collected by iPhones. Remote monitoring of Parkinson patients, based on active tasks delivered by smartphone applications, is an active research field.[1] Here we illustrate the application of our personalized tests using sensor data collected by the mPower study, recently launched as part of Apple's ResearchKit[2,3] mobile platform. The active tests implemented in the mPower app include tapping, voice, memory, posture and gait tests, although in this paper we focus on the tapping data only. During a tapping test the patient is asked to tap two buttons on the iPhone screen alternating between two fingers on the same hand for 20 seconds. Raw sensor data collected during a single test is given by a time series of the screen x-y coordinates on each tap. Processed data corresponds to multiple features extracted from the tapping task, such as the number of taps and the mean inter-tapping interval. Since the active tests are performed by the patient on a daily basis, before and after medication, over several months, the processed data corresponds to time series of feature measurements annotated according to whether the measurement was taken before or after the patient has taken his/her medication. Though others have investigated the feasibility of monitoring medication response in Parkinson patients using smartphone sensor data, this previous work did not focus on the individual effects that medications have, but rather focused on the classification on a population level.[4]

The first step in analyzing these data is to show that simple feature-specific tests, which ignore the serial correlation in the extracted features, are statistically valid (the distribution of the p-values for tests applied to data generated under the null hypothesis is uniform). This condition guarantees that the tests are exact, that is, the type I error rates match the nominal levels, so that our inferences are neither conservative nor liberal. In other words, if we adopt a significance level cutoff of α, the probability that our tests will incorrectly reject the null when it is actually true is given by α.

Even though the simple feature-specific tests are valid procedures for testing for medication response, in practice, we have multiple features and need to combine them into a single decision procedure. The second main contribution of this paper is to propose two distinct approaches to combine all the extracted features into a single hypothesis test. In the first, and most standard approach, we combine simple tests, applied to each one of our extracted features, into a single union-intersection test. Although simple to implement, scalable, and computationally efficient, this approach requires multiple testing correction, which might become burdensome when the number of extracted features is large. In order to circumvent this potential issue, our second approach is to construct personalized classifiers of the before/after medication labels using all the extracted features of a given patient, and test the null hypothesis that the area under the receiver operating characteristic curve (AUROC) of the classifier is equal to 1/2 (in which case the patient's extracted features are unable to predict the before/after medication labels, implying that the patient does not respond to the medication). A slight disadvantage of the classifier approach, compared to the union-intersection tests, is the larger computational cost (especially for classifiers that require tuning parameter optimization by cross-validation) involved in the classifier training. In any case, the increased computational demand is by no means a limiting factor for the application of the approach.

The rest of this paper is organized as follows. In Section 2 we present our personalized tests, discuss their statistical validity, and perform a power study comparison. In Section 3 we illustrate the application of our tests to the tapping data of the mPower study. Finally, in Section 4 we discuss our results.

2. Methods

2.1. *Notation and a few preliminary comments on the data*

Throughout this paper, we let x_{kt}, $k = 1, \dots, p$, $t = 1, \dots, n$, represent the measurement of feature k at time point t, and let $y_t = \{b, a\}$, represent the binary outcome variable, corresponding to the before/after medication label, where b and a stand for "before" and "after" medication, respectively.

Even though the participants where asked to perform the active tasks 3 times per day, one before the medication, one after, and one at any other time of their choice, participants did not always follow the instructions correctly. As a result, the data is non-standard, with variable number of daily tasks (sometimes fewer, sometimes greater than 3 tasks per day), and variable timing relative to medication patterns (e.g., bbabba..., aaabbb..., instead of bababa...). Furthermore, the data also contains missing medication labels, as sometimes, a participant performed the active task but did not report whether the task was taken before or after medication. In our analysis we restrict our attention to data collected before and after medication only. Hence, for each participant, the number of data points used in our tests is given by $n = n_b + n_a$, where n_b and n_a correspond, respectively, to the number of before/after medication labels.

2.2. *On the statistical validity of personalized tests which ignore the autocorrelation structure of the data*

It is common knowledge that the t-test, the Wilcoxon rank-sum test, and other two-sample problem tests, suffer from inflated type I error rates in the presence of dependency. We point out, however, that this can happen when the data within each group is dependent, but the two groups are themselves statistically independent. When the data from both groups is sampled jointly from the same multivariate distribution, the dependency of the data might no longer be an issue. Figure 1 provides an illustrative example with t-tests applied to simulated data.

The t-test's assumption of independence (within and between the groups' data) is required in order to make the analytical derivation of the null distribution feasible. It doesn't mean the test will always generate inflated type I error rates in the presence of dependency (as illustrated in Figure 1f). As a matter of fact, a permutation test based on the t-test statistic is valid if the group labels are exchangeable under the null,[5] even when the data is statistically dependent. Exchangeability[6] captures a notion of symmetry/similarity in the data, without requiring independence. On the examples presented in Figure 1, the group labels are exchangeable on panels a and c as illustrated by the symmetry/similarity of the data between groups 1 and 2 at each row of the heatmaps. For panel b, on the other hand, the lack of symmetry between the groups on each row illustrates that the group labels are not exchangeable.

In the context of our personalized tests, the before/after medication labels are exchangeable under the null of no medication response, even though the measurements of any extracted feature are usually serially correlated. Note that the exchangeability is required for the medication labels, and not for the feature measurements, which are not exchangeable due to their serial correlation. Figure 2 illustrates this point, showing the symmetry/similarity of the separate time series for the before and after medication data.

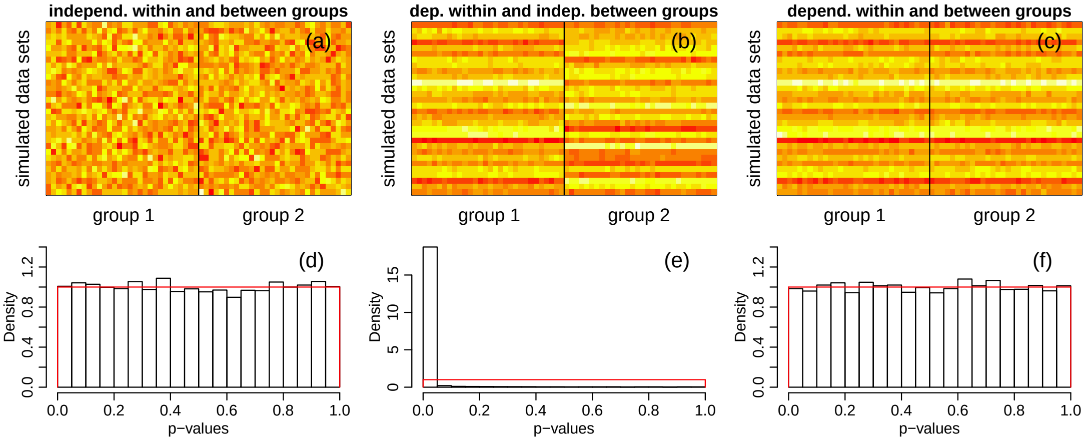


Fig. 1. The effect of data dependency on the t-test. Panels a, b, and c show heatmaps of the simulated data. Columns are split between group 1 and 2, and each row corresponds to one simulated null data set (we show the top 30 simulations only). Bottom panels show the p-value distributions for 10,000 tests applied to null data simulated according to: (i) $\boldsymbol{x}_1 \sim \mathrm{N}_{30}(\boldsymbol{\mu}_1, \boldsymbol{I})$ and $\boldsymbol{x}_2 \sim \mathrm{N}_{30}(\boldsymbol{\mu}_2, \boldsymbol{I})$ with $\boldsymbol{\mu}_1 = \boldsymbol{\mu}_2 = \boldsymbol{0}$ (panel d); (ii) $\boldsymbol{x}_1 \sim \mathrm{N}_{30}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma})$ and $\boldsymbol{x}_2 \sim \mathrm{N}_{30}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma})$, where $\boldsymbol{\mu}_1 = \boldsymbol{\mu}_2 = \boldsymbol{0}$ and $\boldsymbol{\Sigma}$ is a correlation matrix with off-diagonal elements equal to $\rho = 0.95$ (panel e); and (iii) $(\boldsymbol{x}_1, \boldsymbol{x}_2)^t \sim \mathrm{N}_{60}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, with $\boldsymbol{\mu} = (\boldsymbol{\mu}_1, \boldsymbol{\mu}_2)^t = \boldsymbol{0}$ and $\boldsymbol{\Sigma}$ as before (panel f). The density of the Uniform$[0, 1]$ distribution is shown in red. Panel d shows that under the standard assumptions of the t-test, the p-value distribution under the null is (as expected) uniform. Panel e shows the p-value distribution for strongly dependent data, showing highly inflated type I error rates, even though the data was simulated according to t-test's null hypothesis that $\boldsymbol{\mu}_1 = \boldsymbol{\mu}_2$. Panel b clarifies why that is the case. For each row (i.e., simulated data set), the data tends to be quite homogeneous inside each group, but quite distinct between the groups. Because on each simulation we sample the data vectors $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ from a multivariate normal distribution with a very strong correlation structure, all elements in the $\boldsymbol{x}_1$ vector tend to be close to each other, and all elements in $\boldsymbol{x}_2$ tend to be similar to each other. However, because $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are sampled independently from each other, their values tend to be distinct. In combination, the small variability in each group vector together with the difference in their means leads to high test statistic values and small p-values. Panel f shows the p-value distribution for strongly dependent data, when sampled jointly. In this case, the distribution is uniform. Panel c clarifies why. Now, each row tends to be entirely homogeneous (within and between groups), since the joint sampling of $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ makes all elements in both vectors quite similar to each other, so that the difference in their means tends to be small.

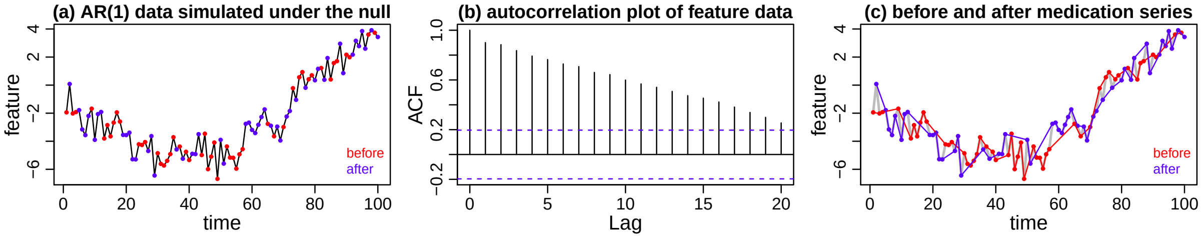


Fig. 2. Exchangeability of the before/after medication labels in time series data. Panel a shows a feature, simulated from an AR(1) process, under the null hypothesis that the patient does not respond to medication. In this case, the before medication (red dots) and after medication (blue dots) labels are randomly assigned to the feature measurements. Panel b shows an autocorrelation plot of the feature data. Panel c shows the separate "before medication" (red) and "after medication" (blue) series. Note the symmetry/similarity of the two series. Clearly, under the null hypothesis that the patient does not respond to medication, the medication labels are exchangeable, since shuffling of the before/after medication labels would not destroy the serial correlation structure or the trend of the series.

Hence, even though our longitudinal data violates the independence assumption of the t-test, the permutation test based on the t-test statistic is still valid. Of course the same argument is valid for permutation tests based on other test statistics. Figure 3 illustrates this point, with permutation tests based on the t-test and Wilcoxon rank-sum test. Panel a shows the original data. Red and blue dots represent measurements before and after medication, respectively. The grey dots represent data collected at another time or where the medication label is missing. Panel b shows one realization of a random permutation of the before/after medication labels. In order to generate a permutation null distribution, we perform a large number of random label shuffles, and for each one, we evaluate the adopted test statistic in the permuted data. Panels c and d show the permutation null distributions generated from 10,000 random permutations of the medication labels based, respectively, on the t-test and on the Wilcoxon rank-sum test statistics. The red curve on panel c shows the analytical density of a

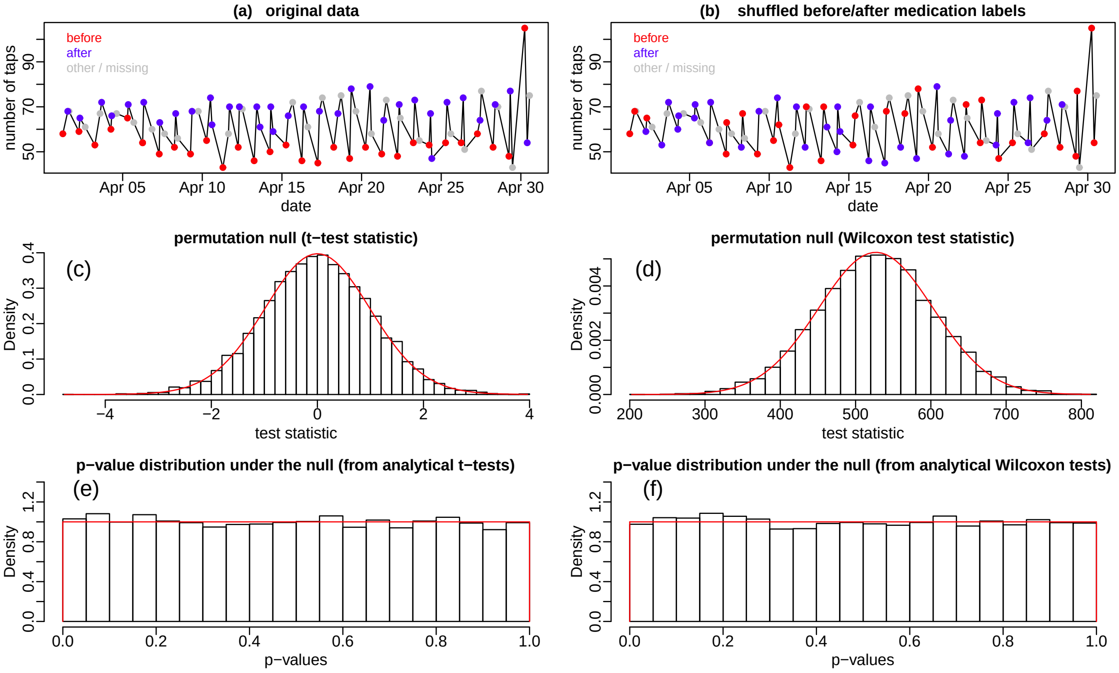


Fig. 3. Personalized tests for the null hypothesis that the patient does not respond to medication, according to a single feature (number of taps), and based on t-test and Wilcoxon rank-sum test statistics. Panel a shows the data on the number of taps for one of mPower's study participants during April 2015. Panel b shows one realization of a random permutation of the before/after medication labels. Panel c shows the permutation null distribution based on the t-test statistic. The red curve represents the analytical density of a t-test, namely, a t-distribution with $n_b + n_a - 2$ degrees of freedom. Panel d shows the permutation null distribution based on Wilcoxon rank-sum test statistic. The red curve shows the density of the normal asymptotic approximation for Wilcoxon's test, namely, a Gaussian density with mean, $n_b\, n_a/2$, and variance, $n_b\, n_a(n_b + n_a + 1)/12$. In this example $n_b = 31$ and $n_a = 34$. Panels e and f show the analytical p-value distributions under the null. Results are based on 10,000 null data sets. Each null data set was generated as follows: (i) randomly sample the number of "before" medication labels, n_b, from the set $\{10, 11, \ldots, n - 10\}$, where n is the total number of measurements; (ii) compute the number of "after" medication labels as $n_a = n - n_b$; and (iii) randomly assign the "before medication" and "after medication" labels to the number of taps measurements. The plots show the histograms of the p-values derived from the application of t-tests (panel e) and Wilcoxon's tests (panel f) to each of the null data sets. The density of the Uniform$[0, 1]$ distribution is shown in red.

t-test for the data on panel a, while the red curve on panel d shows the density of the normal asymptotic approximation for the Wilcoxon test represented in panel b (we show the normal approximation density, since the exact null distribution is discrete). The close similarity of the permutation and analytical distributions suggests that, in practice, we can use the analytical p-values of t-tests or Wilcoxon rank-sum tests instead of the permutation p-values. Panels e and f further corroborate this point, showing uniform distributions for the analytical p-values of t-tests and Wilcoxon tests respectively, derived from 10,000 distinct null data sets.

2.3. *Personalized union-intersection tests*

In order to combine the feature-specific tests, H_{0k} : the patient does not respond to the medication, according to feature k, versus H_{1k} : the patient responds to the medication, across all extracted features, we construct the union-intersection test,

$$H_0 : \cap_{k=1}^{p} H_{0k} \quad \text{versus} \quad H_1 : \cup_{k=1}^{p} H_{1k} \ , \tag{1}$$

where, in words, we test the null hypothesis that the patient does not respond to medication, for all p features, versus the alternative hypothesis that he/she responds to medication according to at least one of the features. Under this test, we reject H_0 if the p-value of at least one of the feature-specific tests is small. Hence, the p-value for the union-intersection test corresponds to the smallest p-value (across all p tests) after multiple testing correction.

We implement union-intersection tests based on the t-test and Wilcoxon rank-sum test statistics. We adopt the Benjamini-Hochberg approach[7] for multiple testing correction. As detailed in Figure 4, the union-intersection test tends to be slightly conservative when applied to correlated features.

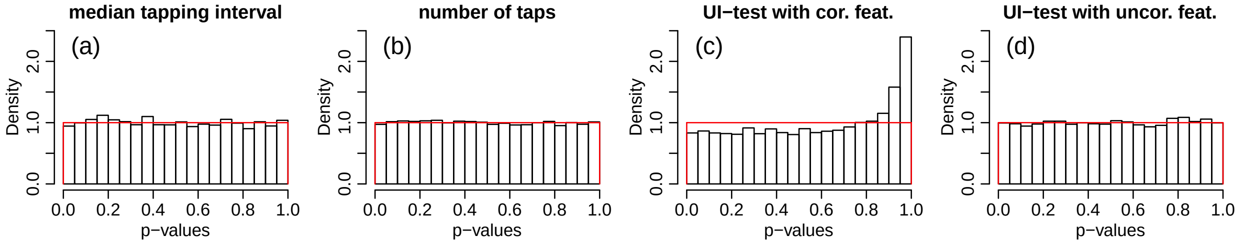


Fig. 4. P-value distributions for the union-intersection test under the null. Results are based on 10,000 null data sets generated by randomly permuting the before/after medication labels. Panels a and b show the p-value distributions from, H_{0k}, for 2 out of the 24 features combined in the union-intersection test. We observed uniform distributions for all features, but report just 2 here due to space constraints. Panel c shows the p-value distribution of the union-intersection test using Benjamini-Hochberg correction. The skewness of the distribution towards larger p-values indicates that the test is conservative, meaning that at a nominal significance level α the probability of rejecting the null when it is actually true is smaller than α. Since the p-value distributions of the features are uniform (panels a and b), the skewness of the union-intersection p-value distribution is clearly due to the multiple testing correction. We experimented with other procedures, but the Benjamini-Hochberg correction was the least conservative one. Panel d reports the p-value distribution for the union-intersection test using a shuffled version of the feature data. Note that by shuffling the data, we destroy the correlation among the features, so that the now uniform distribution suggests that the violation of the independence assumption (required by the Benjamini-Hochberg approach) seems to be the reason for the slightly conservative behavior of the union-intersection test. Results were generated using Wilcoxon's test.

2.4. *Personalized classifier tests*

An alternative approach to combine the information across multiple features into a single decision procedure is to test whether a classifier trained using all the features is able to predict the before/after medication labels better than a random guess. Adopting the AUROC as a classification performance metric, we have that a prediction equivalent to a random guess would have an AUROC equal to 0.5, whereas a perfect prediction would lead to an AUROC equal to 1. Furthermore, if a classifier generates an AUROC smaller than 0.5, we only need to switch the labels in order to make the AUROC larger than 0.5. Therefore, we can test if a classifier's prediction is better than random using the one-sided test,

$$H_0 : \text{AUROC} = 1/2 \quad \text{versus} \quad H_1 : \text{AUROC} > 1/2 \ . \tag{2}$$

It has been shown[8] that, when there are no ties in the predicted class probabilities used for the computation of the AUROC, the test statistic of the Wilcoxon rank-sum test (also known as the Mann-Whitney U test), U, is related to the AUROC statistic by $U = n_b \, n_a(1 - \text{AUROC})$ (see section 2 of reference[9] for details). Hence, under the assumption of independence (required by Wilcoxon's test) the analytical p-value for the hypothesis test in (2) is given by the left tail probability, $P(U \leq n_b \, n_a(1 - \text{AUROC}))$, of Wilcoxon's null distribution. In the presence of ties, the p-value is given by the left tail of the asymptotic approximate null,

$$U \approx \text{N}\left(\frac{n_b \, n_a}{2} \, , \, \frac{n_b \, n_a(n+1)}{12} - \frac{n_b \, n_a}{12 \, n \, (n-1)} \sum_{j=1}^{\tau} t_j(t_j - 1)(t_j + 1) \right) \ , \tag{3}$$

where τ is the number of groups of ties, and t_j is the number of ties in group j.[9] Alternatively, we can get the p-value as the right tail probability of the corresponding AUROC null,

$$\text{AUROC} \approx \text{N}\left(\frac{1}{2} \, , \, \frac{n+1}{12 \, n_b \, n_a} - \frac{1}{12 \, n_b \, n_a \, n \, (n-1)} \sum_{j=1}^{\tau} t_j(t_j - 1)(t_j + 1) \right) \ . \tag{4}$$

As before, even though the test described above assumes independence, the exchangeability of the before/after medication labels guarantees the validity of the permutation test based on the AUROC statistic. Figure 5 illustrates this point.

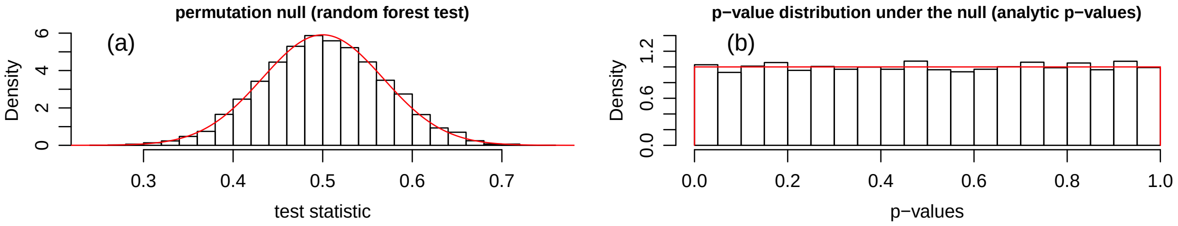


Fig. 5. Panel a shows the permutation null distribution for the personalized classifier test (based on the random forest algorithm). The red curve represents the density of the AUROC (approximate) null distribution in eq. (4). Panel b shows the p-value distribution for the (analytical) classifier test, based on 10,000 null data sets, generated by randomly permuting the before/after medication labels as described in Figure 3. The density of the Uniform[0, 1] distribution is shown in red.

2.5. *Statistical power comparison*

In this section we compare the statistical power of the personalized classifier test (based on the random forest[10] and extra trees[11] classifiers) against the personalized union-intersection tests (based on t-tests and Wilcoxon rank-sum tests). We simulated feature data, x_{kt}, according to the model,

$$x_{kt} \,=\, A\,\cos(\pi\, t) \,+\, \epsilon_{kt}\ , \tag{5}$$

where $\epsilon_{kt} \sim \text{N}(0, \sigma_k^2)$ represents i.i.d. error terms, and the function $A \cos(\pi t)$ describes the periodic signal, with A representing the peak amplitude of the signal. Figure 6 describes the additional steps involved in the generation of the before/after medication labels.

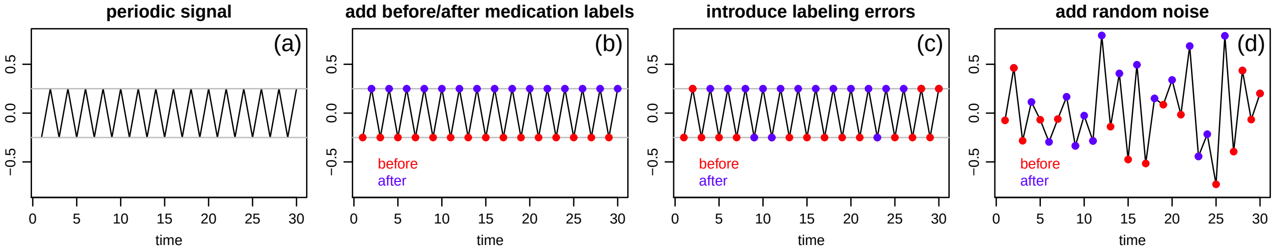


Fig. 6. Data simulation steps. First, we generate the periodic signal, $A \cos(\pi t)$, shown in panel a for $t = 1, \ldots, 30$, and $A = 0.25$. Second, we assign the "before medication" label (red dots) to the negative values, and the "after medication" label (blue dots) for the positive values (panel b). Third, we introduce some labeling errors (panel c). Finally, at the fourth step (panel d), we add $\epsilon_{kt} \sim \text{N}(0, \sigma_k^2)$ error terms to the measurements.

We ran six simulation experiments, covering all combinations of sample size, $n = \{50, 150\}$, and number of features, $p = \{10, 50, 200\}$. In all simulations we adopted $A = 0.25$, mislabeling error rate of 10%, and increasing $\sigma_k = \{1.00, 1.22, 1.44, 1.67, 1.89, 2.11, 2.33, 2.56, 2.78\}$ for the first 9 features, and $\sigma_k = 3$, for $10 \leq k \leq p$. In each experiment, we generated 1,000 data sets and computed the personalized tests p-values. Figure 7 presents a comparison of the empirical power of the personalized tests as a function of the significance threshold α. For each test, the empirical power curve was estimated as the fraction of the p-values smaller than α, for a dense grid of α values varying between 0 and 1.

The results showed some clear patterns. First, the comparison of the top and bottom panels showed that for all 4 tests an increase in sample size leads to an increase in statistical power (as one would have expected). Second, the empirical power of the union-intersection tests based on Wilcoxon and t-tests was very similar in all simulation experiments, with the t-test being slightly better powered than the Wilcoxon test. This result was expected since the simulated features were generated using Gaussian errors, and the t-test is known to be slightly better powered than the Wilcoxon rank-sum test under normality. Similarly, the empirical power of the personalized classifier tests was also similar, with the extra trees algorithm tending to be slightly better powered. Third, this study shows that neither the personalized classifier nor the union-intersection approaches dominate each other. Rather, we see that for this particular data generation mechanism and simulation parameters choice, the union-intersection tests tended to be better powered than the classifier tests for smaller values of p, whereas the converse was

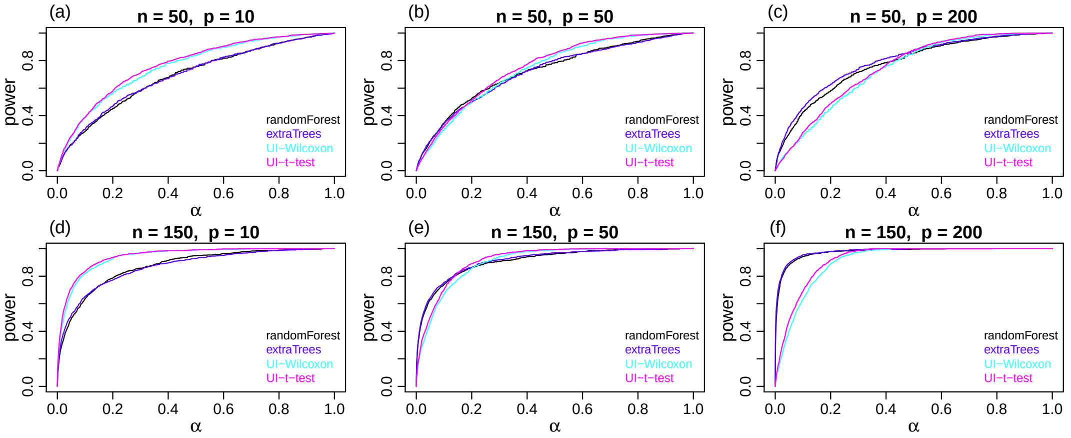


Fig. 7. Comparison of the personalized test's empirical power as a function of the significance cutoff, α.

true for larger p. Furthermore, observe that the power of the union-intersection tests tended to decrease as the number of features increased (note the slight decrease in the slope of the cyan and pink curves as we go from the panels on the left to the panels on the right). The tests based on the classifiers, on the other hand, tended to get better powered as the number of features increased (note the respective increase in the slope of the blue and black curves).

The observed decrease in power of the union-intersection tests might be explained by the increased burden caused by the multiple testing correction required by these tests. On the other hand, the personalized classifier tests are not plagued by the multiple testing issue since all features are simultaneously accounted for by the classifiers. Furthermore, in situations where none of the features is particularly informative, the classifiers might still be able to better aggregate information across the multiple noisy features.

3. Real data illustrations

In this section we illustrate the performance of our personalized hypothesis tests, based on tapping data collected by the mPower study between 03/09/2015 (date the study opened) and 06/25/2015. We restrict our analyzes to 57 patients, who performed at least 30 tapping tasks before medication, as well as 30 or more tasks after medication.

Figure 8 presents the results. Panel a shows the number of tapping tasks (before and after medication) performed by each participant. Panel b reports the AUROC scores for 4 classifiers (random forest, logistic regression with elastic-net penalty,[12] logistic regression, and extra trees). Panel c presents the p-values for the respective personalized classifier tests[a], as well as for 2 personalized union-intersection tests (based on t- and Wilcoxon rank-sum tests).

Panel b shows that the tree-based classifier tests (random forest and extra trees) showed comparable performance across all participants, whereas the regression-based approaches (elastic net and logistic regression) were sometimes comparable but sometimes strongly out-

[a] The results for the personalized classifier tests were based on 100 random splits of the data into training and test sets, using roughly half of the data for training and the other half for testing. The AUROC and p-values reported on Figure 8 b and c correspond to the median of the AUROCs and p-values across the 100 data splits.

performed by the tree-based tests. Panel c shows that the union-intersection tests, nonetheless, produced at times much smaller p-values than the classifier tests. At a significance level of 0.001 (grey horizontal line), about one quarter of the patients (leftmost quarter) respond to dopaminergic medication, according to most tests.

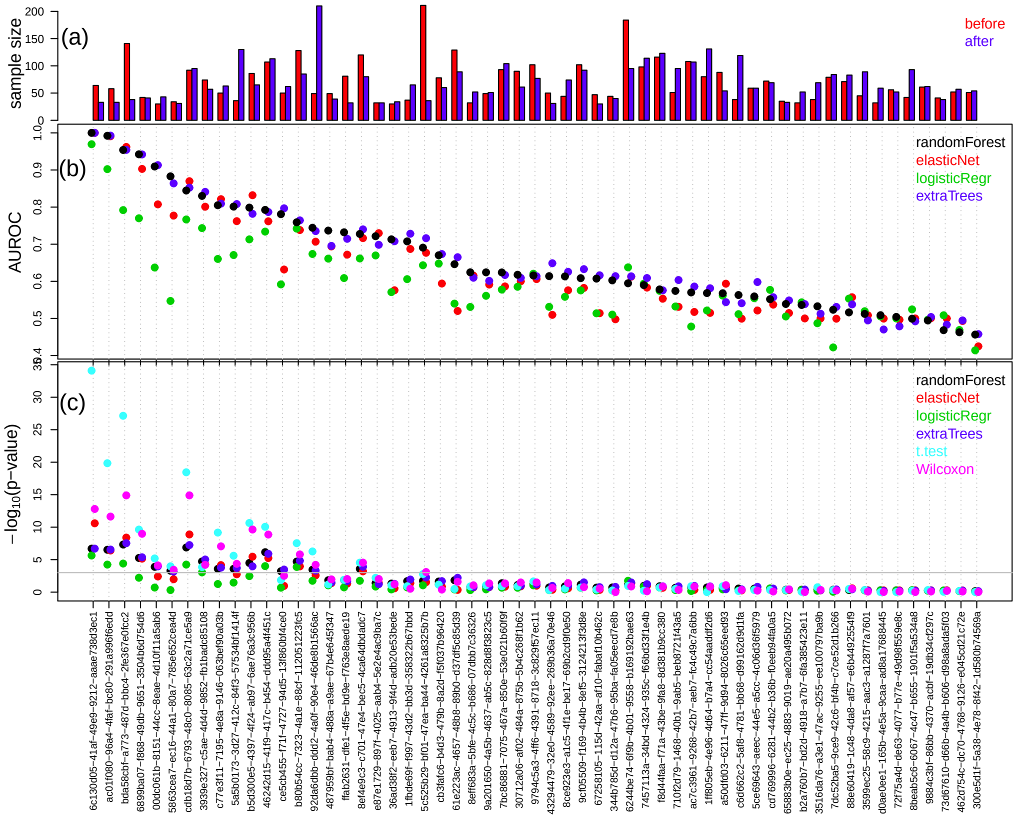


Fig. 8. Results of personalized hypothesis tests applied to the tapping data from the mPower study. Panel a shows the number of tapping tasks performed before (red) and after (blue) medication, per participant. Panel b shows the AUROC scores (across 100 random splits of the data into training and test sets) for four classifiers. Panel c shows the adjusted p-values (in negative log base 10 scale) of 4 classification tests and 2 union-intersection tests. The p-values were adjusted using Benjamini-Hochberg multiple testing correction. The grey horizontal line represents an adjusted p-value cutoff of 0.001. The participants were sorted according to the AUROC of the random forest algorithm (black dots in panel b).

In order to illustrate how our personalized tests are able to pick up meaningful signal from the extracted features, we present on Figure 9, time series plots of 2 features (number of taps and mean tapping interval) which were consistently ranked among the top 3 features for the top 3 patients on the left of Figure 8c, and compared it to the data from the bottom 3 patients on the right of Figure 8c. (For the random forest classifier test, we ranked the features according to the importance measure provided by the random forest algorithm. For the union-intersection tests, we ranked the features according to the p-values of the feature-specific tests.) Comparison of panels a to f, which show the data from patients that respond to medication (according to our tests), against panels g to l, which report the data from patients that do not respond to medication, shows that our tests can clearly pick up meaningful signal from these two extracted features.

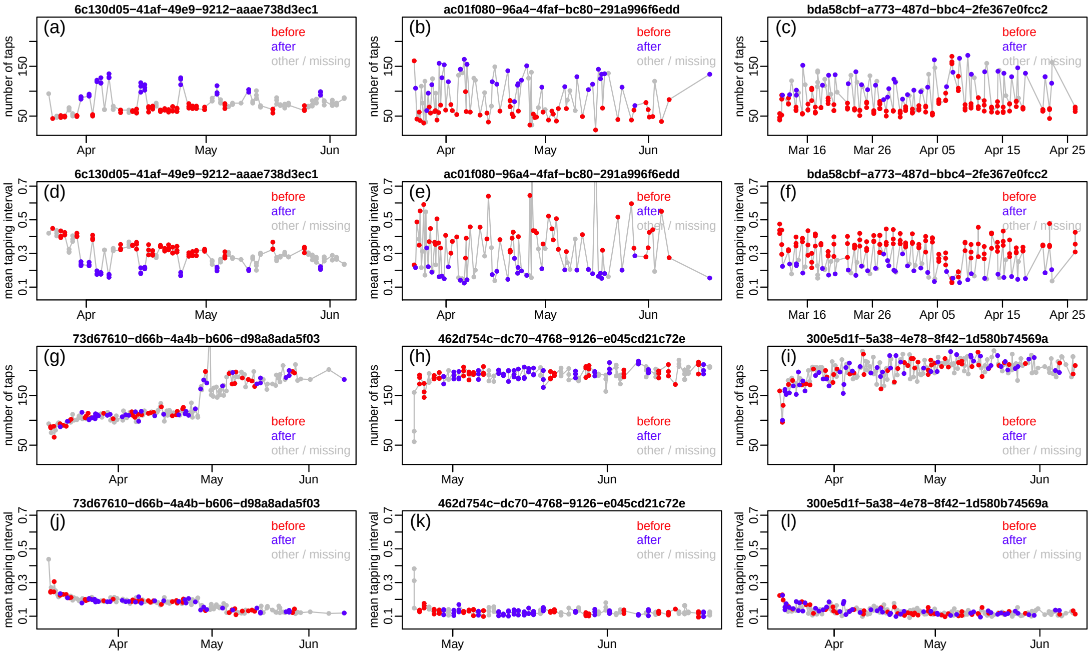


Fig. 9. Comparison of the leftmost 3 patients against the rightmost 3 patients in Figure 8c, according to 2 tapping features (number of taps and mean tapping interval). Panels a to f show the results for the top 3 patients (which respond to medication according to our tests), while panels g to l show the results for the bottom 3 patients (which don't respond to medication according to our tests).

4. Discussion

In this paper we describe personalized hypothesis tests for detecting dopaminergic medication response in Parkinson patients. We propose and compare two distinct strategies for combining the information from multiple extracted features into a single decision procedure, namely: (i) union-intersection tests; and (ii) hypothesis tests based on classifiers of the medication labels. We carefully evaluated the statistical properties of our tests, and illustrated their performance with tapping data from the mPower study. We have also successfully applied our tests to features extracted from the voice and accelerometer data collected using the mPower app, but cannot present these results here due to space limitations.

About one quarter of the patients analyzed in this study showed strong statistical evidence for medication response. For these patients, we observed that the union-intersection tests tended to generate smaller p-values than the classifier based tests. This result corroborates our observations in the empirical power simulation studies, where union-intersection tests tended to out-perform classifier tests when the number of features is small (recall that we employed only 24 features in our analyses).

Although our tests can detect medication responses, they do not explicitly determine the direction of the response, that is, they cannot differentiate a response in the expected direction from a paradoxical one. We point out, however, that their main utility is to help out the physician pinpoint patients in need of a change on their drug treatment. For instance, our tests are able to detect patients for which the drug has an effect in the expected direction but is not well calibrated (so that its effect is wore off by the time the patient takes the medication), or patients showing paradoxical responses. Note that both cases flag a situation which requires an action by the physician (calibrating the medication dosage in the first case, and stopping/changing the medication in the second one). Therefore, even though our tests cannot distinguish between these cases they are still able to detect patients which can potentially benefit from a dose calibration or a change in medication.

Even though we restricted our attention to 4 classifiers, other classification algorithms could also be easily used for generating additional personalized classifier tests. In our experience, however, tree based approaches such as the random forest and extra trees classifiers tend to perform remarkably well in practice, providing a robust default choice. Similarly, we focused on t- and Wilcoxon rank-sum tests in the derivation of our personalized union-intersection tests, since these tests show robust practical performance for detecting group differences.

Although others have investigated the feasibility of monitoring medication response in Parkinson patients using smartphone sensor data,[4] their study focused on the classification of the before/after medication response at the population level, with the classifier applied to data across all patients, and not at a personalized level, as done here.

In this paper we show that simple hypothesis tests, which ignore the serial correlation in the feature data, are statistically valid and well powered to detect medication response in the mPower study data. We point out, however, that, in theory, more sophisticated approaches able to leverage the longitudinal nature of the data could in principle improve the statistical power to detect medication response. In any case, the good practical performance of our simple personalized tests suggests that they can be used as sound baseline approaches, against which more sophisticated methods can be compared.

Code and data availability. All the R[13] code implementing the personalized tests, and used to generate the results and figures in this paper is available at `https://github.com/Sage-Bionetworks/personalized_hypothesis_tests`. The processed tapping data is available at `doi:10.7303/syn4649804`.

Acknowledgements. This work was funded by the Robert Wood Johnson Foundation.

References

1. S. Arora, et al, *Parkinsonism and related disorders* **21**, 650-653 (2015).
2. Editorial, *Nature Biotechnology* **33**, 567 (2015).
3. S. H. Friend, *Science Translational Medicine* **7**, 297ed10 (2015).
4. A. Zhan, et al, *High frequency remote monitoring of Parkinson's disease via smartphone: platform overview and medication response detection* (manuscript under review) (2015).
5. B. Efron, R. J. Tibshirani, *An introduction to the bootstrap*, Chapter 15 (Chapman and Hall/CRC, 1993).
6. J. Galambos, *Encyclopedia of Statistical Sciences* **7**, 573-577 (1996).
7. Y. Benjamini, Y. Hochberg, *Journal of the Royal Statistical Society, Series B* **57**, 289-300 (1995).
8. D. Bamber, *Journal of Mathematical Psychology* **12**, 387-415 (1975).
9. S. L. Mason, N. E. Graham, *Quarterly Journal of the Royal Meteorological Society* **128**, 2145-2166 (2002).
10. L. Breiman, *Machine Learning* **45**, 5-32 (2001).
11. P. Geurts, D Ernst, L. Wehenkel, *Machine Learning* **63**, 3-42 (2006).
12. J. H. Friedman, T. Hastie, R. Tibshirani, *Journal of Statistical Software* **33 (1)**, (2010).
13. R Core Team, *R Foundation for Statistical Computing* (Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org/ 2015).