

# Manifold Mixing for Stacked Regularization

João Pereira<sup>1,3</sup>, Erik S. G. Stroes<sup>1</sup>, Albert K. Groen<sup>1</sup>, Aeilko H. Zwinderman<sup>2</sup>,  
and Evgeni Levin<sup>1,3</sup>

<sup>1</sup> Department of Cardiovascular Medicine,

University Medical Center Amsterdam, The Netherlands

<sup>2</sup> Department of Clinical Epidemiology, Biostatistics and Bioinformatics,

University Medical Center Amsterdam, The Netherlands

<sup>3</sup> Horaizon BV, Delft, The Netherlands

{j.p.belopereira, e.s.stroes, a.k.groen, a.h.zwinderman,  
e.levin}@amsterdamumc.nl

**Abstract.** In many real-world learning tasks, one has access to datasets consisting of multiple modalities, for example, various omics profiles of the patients coupled with medical records and other unstructured data sources. Often, the "core mechanism" (e.g. health or disease state) is reflected in all of these modalities and so this commonality can become more evident when the source domain (e.g. proteins) can accordingly transform the local geometry of the target (e.g. lipids). In this paper, we propose a novel algorithm that takes multiple data sources, constructs corresponding manifolds, and "mixes" information across them to find the common denominators in the observable outcomes. By leveraging manifold information from these different sources we obtain more robust and accurate results in comparison to standard methods. In the empirical evaluation on a clinical cohort related to ischaemia in patients with coronary artery disease, we demonstrate the applicability and efficacy of the proposed algorithm.

**Keywords:** Manifold Mixing · Multi-view · Stacked Regularization

## 1 Introduction

Exponential increase in multi-modal data, stemming from different instruments and measurements presents both an opportunity and a challenge. With a larger sheer volume of information, there is potentially more we can learn for a given process, but coherently combining different data sources with the goal of improving analysis remains a challenging and underdeveloped task. In the medical field for instance, multiple omics data such as proteins or lipids encode somewhat related biological information. Therefore, one might expect that health or disease state is reflected in both of these modalities, despite their different format. Learning frameworks such as manifold alignment [10, 5] and domain adaptation [6, 8] may not be directly applicable as they try to find a common latent manifold and learn to transfer knowledge from a source to a target domain, respectively. Orthogonally to existing methods, we present a way of "mixing" information

from multiple domains, without imposing hard similarity between them. The motivation is that for a given outcome, the "core mechanism" (e.g. health or disease state) is reflected in all of these modalities and so this commonality can become more evident when the source domain (e.g. proteins) can accordingly transform the local geometry of the target (e.g. lipids).

## 2 Approach

We use a stacked regularization setting [11] where each level-one model is trained using "mixed manifolds" of various data modalities. In the next subsections we briefly discuss classical stacked regularization, domain alignment, adaption, and finally propose our mixing algorithm. Regarding notation, we will use capital bold, bold and no formatting for matrices, vectors and scalars or functions, respectively (e.g.  $\mathbf{X}$ ,  $\mathbf{x}$ ,  $f/W$ ). We will also use calligraphic font to denote spaces (e.g.  $\mathcal{X}$ ).

### 2.1 Stacking

Let  $\mathbf{X}$  be a dataset of  $N$  samples whose values are sampled from an input space  $\mathcal{X} = \{\mathcal{X}^1, \mathcal{X}^2, \dots, \mathcal{X}^M\}$  where  $\mathcal{X}^1$  to  $\mathcal{X}^M$  are subspaces corresponding to different "sources" or "views"  $1 \rightarrow M$  which we will refer to as "domains". Denote by  $y$  the output sampled from an output space  $\mathcal{Y}$ . In a supervised setting, the goal is to compute  $p(y|\mathbf{x}^1, \dots, \mathbf{x}^M)$ , where  $\mathbf{x}^i$  are the coordinates of an instance from  $\mathbf{X}$  in the domain  $\mathcal{X}^i$ . In stacked regularization, or stacking, the input is passed to a first layer of  $W_0$  predictors  $g_1^0(\mathbf{x}), \dots, g_{W_0}^0(\mathbf{x})$ , with:

$$g_i^0(\mathbf{x}) = p(y|\mathbf{x}^1, \dots, \mathbf{x}^M, \boldsymbol{\theta}_i^0) , \quad (1)$$

where  $\boldsymbol{\theta}_i^0$  are the hyperparameters of the  $i$ th model. For our task, we suggest to pass one data source per model:  $g_i^0(\mathbf{x}) = p(y|\mathbf{x}^i, \boldsymbol{\theta}_i^0)$ , so that the width of the first layer  $W_0$  is equal to the number of domains  $M$ . The output from this layer is then passed to one or more layers of  $W_k$  models  $g_1^k, \dots, g_{W_k}^k$  which blend the outputs of the previous ones:

$$g_i^k(\mathbf{x}) = p(y|g_1^{k-1}(\cdot), \dots, g_{W_k}^{k-1}(\cdot), \boldsymbol{\theta}_i^k) , \quad k \in [1, L] , \quad (2)$$

where  $L$  is the total number of blending layers and  $\boldsymbol{\theta}_i^k$  the hyperparameters of  $i$ th model from the  $k$ th layer. The last blending layer is then passed to a final model  $f$  that produces the output  $f(\mathbf{x}) = p(y|g_1^L(\mathbf{x}), \dots, g_{W_L}^L(\mathbf{x}), \boldsymbol{\theta}^{L+1})$ , where  $\boldsymbol{\theta}^{L+1}$  are the hyperparameters of  $f$ . You can visualize the stacked model general architecture in figure 2.1. From a frequentist point of view, the goal of stacking is then to find:

$$\underset{\boldsymbol{\theta}}{\operatorname{argmin}} \mathcal{L}(\mathbf{y}; f(\mathbf{X}), \boldsymbol{\theta}) , \quad (3)$$

where  $\boldsymbol{\theta}$  is the set of hyperparameter values from all of the stack models,  $\mathbf{y}$  is the output for all of the data, and  $\mathcal{L}$  is the loss function when using  $f(\mathbf{x})$  to predict

$\mathbf{y}$ . For a fully Bayesian approach, one should compute the posterior probability of each model by integrating out the hyperparameter values:

$$\begin{aligned}
 p(g_i^k|Z) &\propto p(Z|g_i^k) p(g_i^k) \propto \\
 p(g_i^k) &\int p(Z|\theta_i^k, g_i^k) p(\theta_i^k|g_i^k) d\theta_i^k,
 \end{aligned} \tag{4}$$

where  $Z$  is the complete dataset  $(\mathbf{X}, \mathbf{y})$ .

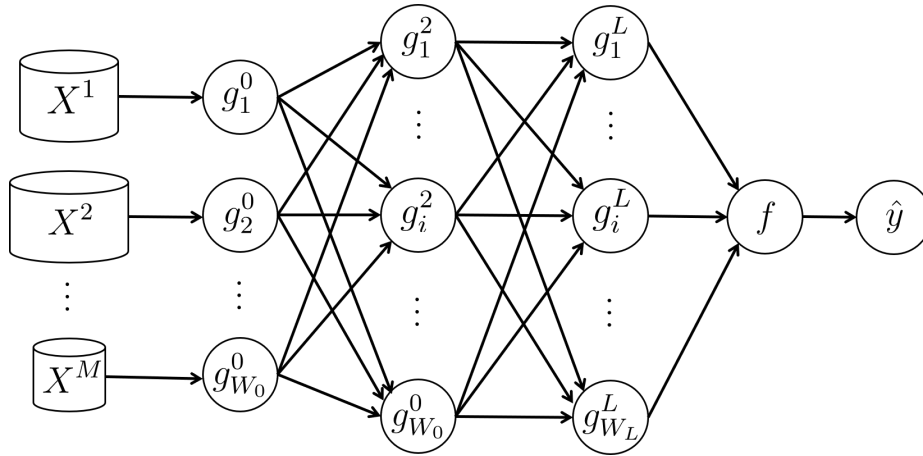


Fig. 1. Our proposed stacked setting

Although this approach is attractive because it considers the uncertainty of the model, it also incurs high computational cost for large  $\theta_i^k$ . Two important aspects are: a) optimizing each model independently does not guarantee finding the global optimal stacked model and b) there is an implicit assumption that each model  $g_i^k$  can learn/handle data from different sources (possibly with different formats) effectively.

## 2.2 Stacking Optimization

Finding an optimal stacked model can be done by optimizing each sub-model individually or by jointly optimizing all the sub-models. Optimizing each model individually has an important complexity advantage because the number of possible  $\theta$  combinations increases exponentially with the number of parameters:  $k^{|\theta|}$ , where  $k$  is the number of values considered for each parameter.

**Lemma 1.** For a given dataset  $\mathbf{X}, \mathbf{y}$ , stacked model  $f(x)$  and parameters  $\theta$ , the following relation is true:  $\mathcal{L}(\mathbf{y}, f(x), \theta^*) \leq \mathcal{L}(\mathbf{y}, f(x), \theta^l)$ , where

$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \mathcal{L}(\mathbf{y}, f(x), \boldsymbol{\theta})$ ,  $\boldsymbol{\theta}' = \varphi_{k=1}^L \left( \varphi_{i=0}^{W_k} \left( \underset{\boldsymbol{\theta}_i^k}{\operatorname{argmin}} \mathcal{L}(\mathbf{y}, g_i^k(x), \boldsymbol{\theta}_i^k) \right) \right)$ ,  
 and  $\varphi_{j=1}^L \left( \underset{\boldsymbol{\theta}_j}{\operatorname{argmin}} f(\boldsymbol{\theta}_j) \right)$  is a sequential composition of minimizations with  
 respect to index  $j$ :  $\underset{\boldsymbol{\theta}_{j=L}}{\operatorname{argmin}} \left( \underset{\boldsymbol{\theta}_{j=L-1}}{\operatorname{argmin}} \left( \dots \underset{\boldsymbol{\theta}_{j=1}}{\operatorname{argmin}} (f(\boldsymbol{\theta}_j)) \right) \right)$ .

*Proof.* Let  $\mu$  be a measure on the measurable space  $(\Theta, \boldsymbol{\theta})$ . Since  $\boldsymbol{\theta}$  is a disjoint set, its measure is just:

$$\mu(\boldsymbol{\theta}) = \sum_{k=0}^{L+1} \sum_{i=1}^{W_k} \mu(\boldsymbol{\theta}_i^k) \quad (5)$$

Denote by  $\{\boldsymbol{\theta}_i^k\}^*$  the set of values that satisfy  $\underset{\boldsymbol{\theta}_i^k}{\operatorname{argmin}} \mathcal{L}(\mathbf{y}, g_i^k(x), \boldsymbol{\theta}_i^k)$ , and  $\{\boldsymbol{\theta}_i^k\}$  the set of values  $\boldsymbol{\theta}_i^k$  can take. Since  $\{\boldsymbol{\theta}_i^k\}^* \subset \{\boldsymbol{\theta}_i^k\}$ , then  $\mu(\{\boldsymbol{\theta}_i^k\}^*) \leq \mu(\{\boldsymbol{\theta}_i^k\})$ ,  $\forall i, k$ , and so:

$$\mu(\boldsymbol{\theta}') = \sum_{k=0}^{L+1} \sum_{i=1}^{W_k} \mu(\{\boldsymbol{\theta}_i^k\}^*) \leq \sum_{k=0}^{L+1} \sum_{i=1}^{W_k} \mu(\{\boldsymbol{\theta}_i^k\}) \quad (6)$$

$\mathcal{L}(\mathbf{y}, f(x), \boldsymbol{\theta}^*)$  is thus optimizing over a larger domain than  $\mathcal{L}(\mathbf{y}, f(x), \boldsymbol{\theta}')$  is, yielding  $\mathcal{L}(\mathbf{y}, f(x), \boldsymbol{\theta}^*) \leq \mathcal{L}(\mathbf{y}, f(x), \boldsymbol{\theta}')$

There is a trade-off between complexity and performance when it comes to optimizing the model. If performance is the goal, then the next step is to decide what form is the optimization going to take. A grid-search would quickly become unfeasible for models with multiple hyperparameters, so an attractive solution is to instead use Bayesian optimization [1].

### 2.3 Domain Alignment

Note that at this point there is no information sharing between the first layer's models. However, in many situations it may be desirable that some information *is* shared across these models since they are build using different modalities of the same sample set. The motivation is that even though the samples come from different distributions, the generating processes should be similar and thus they should lie in a similar low-dimensional manifold. This is the central problem of *Manifold Alignment* [7]. Our *Manifold Mixing* is based on similar motivation, with crucial difference - we consider that each domain has a contribution of its own, and therefore we will not enforce an exact match between the manifolds but merely a transformation of the local inter-sample geometry using all the domains, indirectly linking the stacked first layer models together.

## 2.4 Domain Adaptation

Given two domains  $\mathcal{X}^s$  and  $\mathcal{X}^t$ , that are different but related, the goal of *Domain Adaptation* is that of learning to transfer knowledge acquired from the source  $\mathcal{X}^s$  to the target  $\mathcal{X}^t$ . The most common setting is when there are many labeled examples in the source, but not in the target, and therefore one tries to learn an estimator  $h$  such that it minimizes the error on both the source and target distribution prediction [2]. In our setting, the source and target domains represent different modalities of the same sample. For clarity, we will use source and target domain definitions as well, and use the former to transform the latter.

## 2.5 Manifold Mixing

We would like to address the question: how to combine data from different domains with a similar relation to the output? Our approach consists in creating a map between each pair of domains  $\mathcal{X}^s \rightarrow \mathcal{X}^t$ , while deforming the local geometry of the two to become more similar. We drew inspiration from LLE [9] in that we will also use the neighbours of a point to predict its position. In our case, we will use the neighbours of this point in the other domains to predict its position in the original domain. Consider a set of points  $S$  and two mappings taking the points in  $S$  to two coordinate systems of domains  $\mathcal{X}^t$  and  $\mathcal{X}^s$ :  $\varphi : S \rightarrow \mathbb{R}^{|t|}$ ,  $\psi : S \rightarrow \mathbb{R}^{|s|}$ , and suppose the subsets  $\mathbf{X}^t$ ,  $\mathbf{X}^s$  of the dataset  $\mathbf{X}$  are measured in these coordinate systems. Let us introduce an approximation  $\mathbf{L}_s^t$  to the mapping  $\varphi \circ \psi^{-1} : \mathbb{R}^{|s|} \rightarrow \mathbb{R}^{|t|}$  from the coordinates of domain  $\mathcal{X}^s$  to the coordinates of domain  $\mathcal{X}^t$ :  $\min_{\mathbf{L}_s^t} \sum_{i=1}^N \|\mathbf{x}_i^t - \mathbf{L}_s^t \mathbf{x}_i^s\|^2$ , with  $\mathbf{x}_i^t$ ,  $\mathbf{x}_i^s$  corresponding to the  $i$ th entry of  $\mathbf{X}^t$  and  $\mathbf{X}^s$ , respectively. The optimal solution is then given by:

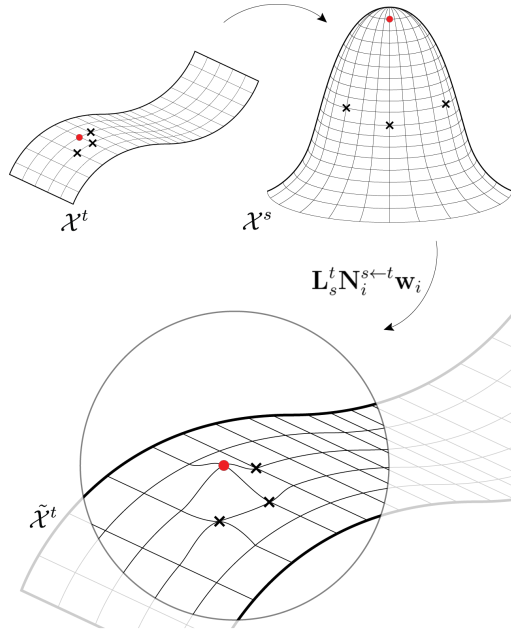
$$\begin{aligned} \frac{\partial}{\partial \mathbf{L}_s^t} \sum_{i=1}^N \|\mathbf{x}_i^t - \mathbf{L}_s^t \mathbf{x}_i^s\|^2 = 0 &\Leftrightarrow \sum_{i=1}^N \mathbf{L}_s^t (\mathbf{x}_i^s \mathbf{x}_i^{s\top}) = \sum_{i=1}^N \mathbf{x}_i^t \mathbf{x}_i^{s\top} \\ \mathbf{L}_s^t (\mathbf{X}^s \mathbf{X}^{s\top}) &= \mathbf{X}^t \mathbf{X}^{s\top} \Leftrightarrow \mathbf{L}_s^t = \mathbf{X}^t \mathbf{X}^{s\top} (\mathbf{X}^s \mathbf{X}^{s\top})^{-1} . \end{aligned} \quad (7)$$

Denote by  $\mathbf{n}_i^t[j]$  the  $j$ th neighbour of instance  $\mathbf{x}_i$  in the domain  $\mathcal{X}^t$ . Let the array of the points in  $\mathcal{X}^s$  which are the neighbours of instance  $\mathbf{x}_i^t$  in the domain  $\mathcal{X}^t$  be:  $\mathbf{N}_i^{s \leftarrow t} = [\mathbf{x}_{\mathbf{n}_i^t[1]}^s, \mathbf{x}_{\mathbf{n}_i^t[2]}^s, \dots, \mathbf{x}_{\mathbf{n}_i^t[k]}^s]$ . Our goal is to 'mix' information from different manifolds. This is accomplished by projecting the neighbours of  $\mathbf{x}_i^t$  from the source to the target domain and then finding the linear combination of the points that best reconstructs  $\mathbf{x}_i^t$  in the original domain:

$$\min_{\mathbf{w}_i} \sum_i \|\mathbf{x}_i^t - \tilde{\mathbf{x}}_i^{t \leftarrow s}\|^2 = \min_{\mathbf{w}_i} \sum_i \|\mathbf{x}_i^t - L_s^t \mathbf{N}_i^{s \leftarrow t} \mathbf{w}_i\|^2, \quad (8)$$

where  $\tilde{\mathbf{x}}_i^{t \leftarrow s}$  is the reconstruction of  $\mathbf{x}_i^t$  using domain  $\mathcal{X}^s$ . We visualize how substituting  $\mathbf{x}_i$  by  $\tilde{\mathbf{x}}_i^{t \leftarrow s}$  might affect the target manifold in figure 2. After setting the derivative w.r.t.  $\mathbf{w}_i$  to zero, the optimal solution corresponds to:

$$\mathbf{w}_i = \left( (\tilde{\mathbf{N}}_i^{t \leftarrow s})^\top \tilde{\mathbf{N}}_i^{t \leftarrow s} \right)^{-1} (\tilde{\mathbf{N}}_i^{t \leftarrow s})^\top \mathbf{x}_i^t, \quad (9)$$



**Fig. 2.** Target manifold being deformed by the source manifold using the manifold mixing method. The crosses are the neighbours of point  $\mathbf{x}_i$  (point in red) in the target domain. These neighbours are mapped from the source to the target domain and then used to locate  $\mathbf{x}_i$ . This causes the target manifold to be locally deformed by the source manifold.

where  $\tilde{\mathbf{N}}_i^{t \leftarrow s} = \mathbf{L}_s^t \mathbf{N}_i^{s \leftarrow t}$ , the neighbors of  $\mathbf{x}_i$  in  $\mathcal{X}_t$  projected from their coordinates in  $\mathcal{X}_s$  back to the coordinates in  $\mathcal{X}_t$ . We can now transform the original space  $\mathcal{X}^t$  into the space reconstructed from the other domains  $\tilde{\mathcal{X}}^t$  by computing for each instance the weighted mean of its reconstructions:

$$\tilde{\mathbf{x}}_i^t = \beta_t \mathbf{x}_i^t + \sum_{s \neq t} \beta_s \tilde{\mathbf{x}}_i^{t \leftarrow s} \quad , \quad (10)$$

where  $\beta_j$  can be seen as the prior of domain  $\mathcal{X}^j$ 's relevance, and  $\sum_j \beta_j = 1$ . When evaluating a new point  $\mathbf{x}_{new}$ , first the nearest neighbours from the training set are found, and then the reconstruction is given by  $\tilde{\mathbf{N}}_i^{s \leftarrow t} \mathbf{w}_{new}$ . The complexity of the algorithm is bounded by the matrix inversion of the coordinate mapping in equation 7, and therefore the algorithm complexity is  $\mathcal{O}(d^3)$ , where  $d$  is the maximum number of features among all the domains.

---

**Algorithm 1** Manifold Mixing Algorithm
 

---

**Input:** data  $\mathbf{X} = [\mathbf{X}^1, \dots, \mathbf{X}^M]$ , domain weights  $\beta$   
**Output:** transformed data  $\tilde{\mathbf{X}}$   
**for**  $t = 1$  **to**  $M$  **do**  
      $\mathbf{n}^t \leftarrow \text{NearestNeighbours}(\mathbf{X}^t, k)$   
      $\tilde{\mathbf{X}}^t \leftarrow \beta_t \mathbf{X}^t$   
     **for**  $s$  **in**  $m \in [1, M] \setminus t$  **do**  
          $\mathbf{L}_s^t \leftarrow \mathbf{X}^t \mathbf{X}^{s\top} (\mathbf{X}^s \mathbf{X}^{s\top})^{-1}$   
         **for**  $\mathbf{x}_i = 1$  **to**  $N$  **do**  
              $\mathbf{N}_i^{s \leftarrow t} \leftarrow \mathbf{X}^s[\mathbf{n}_i^t], \tilde{\mathbf{N}}_i^{t \leftarrow s} \leftarrow \mathbf{L}_s^t \mathbf{N}_i^{s \leftarrow t}$   
              $\mathbf{w}_i \leftarrow \left( (\tilde{\mathbf{N}}_i^{t \leftarrow s})^\top \tilde{\mathbf{N}}_i^{t \leftarrow s} \right)^{-1} (\tilde{\mathbf{N}}_i^{t \leftarrow s})^\top \mathbf{x}_i^t$   
              $\tilde{\mathbf{X}}_i^t += \beta_s \tilde{\mathbf{N}}_i^{t \leftarrow s} \mathbf{w}_i$   
         **end for**  
     **end for**  
**end for**  
 $\tilde{\mathbf{X}} \leftarrow [\tilde{\mathbf{X}}_1, \dots, \tilde{\mathbf{X}}_M]$   
**return:**  $\tilde{\mathbf{X}}$

---

### 3 Experimental Section

#### 3.1 Methods

To test our method we used a recent clinical cohort [3] containing data on patients with cardiovascular disease. There are 440 subjects in the dataset of which 56 suffered from an early cardiovascular event. For each patient, 359 protein levels and 9 clinical parameters are measured. We evaluate performance our method (stacked regularization with manifold mixing) for predicting a cardiovascular event. We compare proposed approach to that of using a standard stacked model with joint Bayesian optimization of the hyper-parameters, as well as with using random forest on the merged/ feature concatenated datasets (protein levels and clinical parameters). For both our method and the standard stacking, the architecture consisted of two larger random forest models in the first layer and a smaller one in the output.

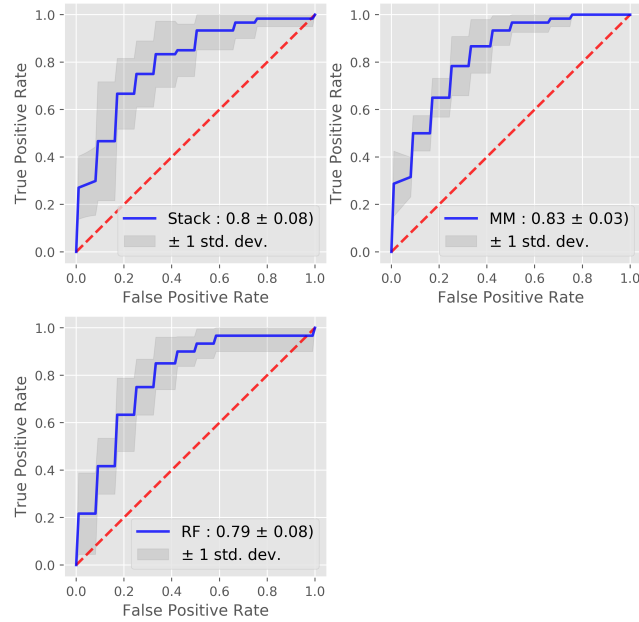
#### 3.2 Data selection and preprocessing

We perform random shuffles with 90% train size and even class distribution in the train/test set. We use remaining 10% to test the model. Since the dataset is unbalanced (much larger number of negative than positive subjects), we took a random sample from the negative class of size equal to the total number of positive class subjects, prior to the split at each shuffle. The protein were measured using a technology that uses standard panels for different proteins, meaning some of the proteins might have no relation to the outcome at all. For this reason, for each run we pre-selected 50 proteins using Univariate Feature Selection on the training set. Then, we normalized the train and test data independently, and measured

the average ROC for each of the methods. We perform 5-fold cross validation for optimal hyper-parameter estimation on the train set for the random forests, and bayesian optimization for the stacked models. Once that is accomplished, we retrain the model with the optimal parameters on the complete training set and test on the remaining 10%. We repeat the this procedure multiple times and report the average ROC-AUC as well as the standard deviation. Described strategy is frequently referred to as stability selection procedure [4] The proteins were measured using OLINK technology that records expression levels of proteins via targeted and customised analysis [3].

### 3.3 Results

The results are presented in figure 3. Proposed approach (MM stacked) outperformed both the regular stacked model and the random forests (RF) using the merged data. Both stacked regularized techniques outperformed standard RF.



**Fig. 3.** Average AUC for the three methods compared. The highest performance is that of the Manifold Mixing stacked model (MM), and both stacked models outperformed using Random Forests (RF) on the merged data



## 4 Conclusions and Future work

In this paper we propose the manifold mixing framework to improve the analysis of multi-modal data stemming from different sources. In our preliminary experiments, the obtained results support efficacy of our method. We outperform both standard stacked regularization and the model built on feature concatenated data. In the near future, we plan on performing further tests with larger number of shuffles, and testing on different datasets and heterogeneous domains. One pitfall of the current algorithm is the linearity of the map between manifolds which might fail in highly curved regions. A possible solution is to kernelize the method using graph kernels. Another interesting direction is to subdivide the manifold into multiple subregions based on the local curvature and create a mapping per subregion.

## Acknowledgments

Supported by a European Research Area Network on Cardiovascular Diseases (ERA-CVD) grant (ERA CVD JTC2017, OPERATION). Thanks to Cláudia Pinhão for supporting with the design of the figures.

## References

1. Acerbi, L., Ma, W.J.: Practical bayesian optimization for model fitting with bayesian adaptive direct search. In: 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA (2017)
2. Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., Vaughan, J.W.: A theory of learning from different domains. *Machine Learning Journal* **79** (1–2), 151–175 (2010)
3. Bom, M.J., Levin, E., Driessen, R., Danad, I., Kuijk, C.V., van Rossum, A.C., ..., Knaapen, P.: Predictive value of targeted proteomics for coronary plaque morphology in patients with suspected coronary artery disease. *EBioMedicine*. (2018). <https://doi.org/10.1016/j.ebiom.2018.12.033>
4. Bühlmann, N.M.P.: Stability selection. *Journal of the Royal Statistical Society* **72**(4), 417–473 (2010)
5. Cui, Z., Chang, H., Shan, S., Chen, X.: Generalized unsupervised manifold alignment. In: *Advances in Neural Information Processing Systems 27* (NIPS 2014) (2014)
6. Hajiramezanali, E., Dadaneh, S.Z., Karbalayghareh, A., Zhou, M., Qian, X.: Bayesian multi-domain learning for cancer subtype discovery from next-generation sequencing count data. In: 32nd Conference on Neural Information Processing Systems (NIPS 2018) (2018)
7. Ham, J.H., Lee, D.D., Saul, L.K.: Learning high dimensional correspondences from low dimensional manifolds. In: *Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003)* (2003)
8. Kumar, A., Sattigeri, P., Wadhawan, K., Karlinsky, L., Feris, R., Freeman, B., Wornell, G.: Co-regularized alignment for unsupervised domain adaptation. In: *Advances in Neural Information Processing Systems 31* (NIPS 2018) (2018)

9. Roweis, S., Saul, L.: Nonlinear dimensionality reduction by locally linear embedding. *Science* **290**(5500), 2323–2326 (2000)
10. Wang, C., Mahadevan, S.: Heterogeneous domain adaptation using manifold alignment. In: *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence* (2011)
11. Wolpert, D.H.: Stacked generalization. *Neural Networks* **5**(241-259) (1992)