Feature Selection in the Presence of Batch Effect

Peng Dai[1], Sina Baharlouei[1], Taojian Tu[2], Handan Hong[2] , Sze-chuan Suen[1], Meisam Razaviyayn[1] , Bangyan Stiles[2]

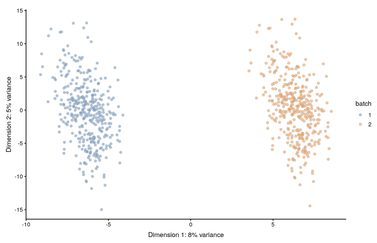
[1] Industrial and Systems Engineering, University of Southern California, Los Angeles, CA 90089, USA

[2] Pharmacology and Pharmaceutical Science, University of Southern California, Los Angeles, CA 90089, USA

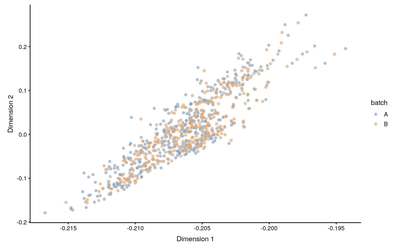
SCAN ME

What Is the Batch Effect

A dataset with batch effect (2 batches)



Dataset after removing batch effect



❖ **Example of Batch Effects in Genomic Datasets**

	GSE154892		GSE130528
ID	GSM4681905	ID	GSM3741792
1	8.829551	1	1177.7
2	8.357929	2	2846.8
3	9.097497	3	4091.1
4	9.165594	4	141.7
5	9.837312	5	1379
6	11.12557	6	626.8

- **Aggregation** of **related datasets** allows for **higher statistical performance**, particularly in the biological and genomic datasets.
- **Batch Effect**: **variability** in **distribution of datasets** due to different **data gathering procedures**, different **environmental effects**, different **data normalization/standardizations**
- **Adverse effect** on the **statistical performance** of inference tasks performed on the merged datasets

❖ **Research Questions**

- ❑ How can we correct for batch effects and perform feature selection for aggregated datasets?
- ❑ More generally, how can we 'learn' from the heterogeneous datasets simultaneously?

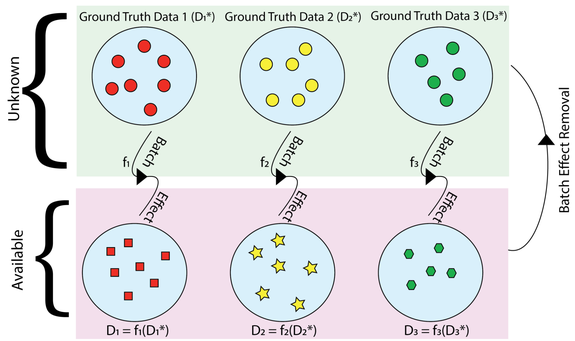
Problem Formulation and Methodology

❖ **Assumptions** (in a biological context):

➢ Datasets affected by **different batch effects**, but all are **monotonic transformations**

➢ The distribution of underlying ground-truth data is **multivariate Gaussian**

➢ A small number of features are relevant for the prediction of the target variable (**sparsity**)



❖ Mathematical Formulation:

➢ Let $Z \sim N(0,1)$, is a normally distributed vector with zero mean and unit variance..

$$\min_{\boldsymbol{\Phi},\boldsymbol{\theta},\boldsymbol{\Sigma}} \quad \frac{1}{n}\sum_{i=1}^{n}\sum_{(\boldsymbol{x},y)\in\mathcal{D}_i} \ell\Big(h_{\boldsymbol{\theta}}\big(\boldsymbol{\Phi}_i(\boldsymbol{x})\big),\boldsymbol{\Phi}_i(y)\Big) + \mu\sum_{i=1}^{m}\mathrm{MMD}\Big(\Phi_i(\mathcal{D}_i),\boldsymbol{\Sigma}Z\Big) + \lambda\|\boldsymbol{\theta}\|_1.$$

Lasso

How far the distribution of each dataset is to the reference distribution?

$$\mathrm{MMD}_u^2[x,y] = \frac{1}{m(m-1)}\sum_{i=1}^{m}\sum_{j\neq i}^{m} k(x_i,x_j) + \frac{1}{n(n-1)}\sum_{i=1}^{n}\sum_{j\neq i}^{n} k(y_i,y_j) - \frac{2}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} k(x_i,y_j)$$

Φ: f^{-1} fully connected neural networks with non-negative weights (monocity).
P^*: $\sim N(\mu,\Sigma)$, where $\Sigma = B^T B$ where B is a low rank matrix.

Implementation and Validation

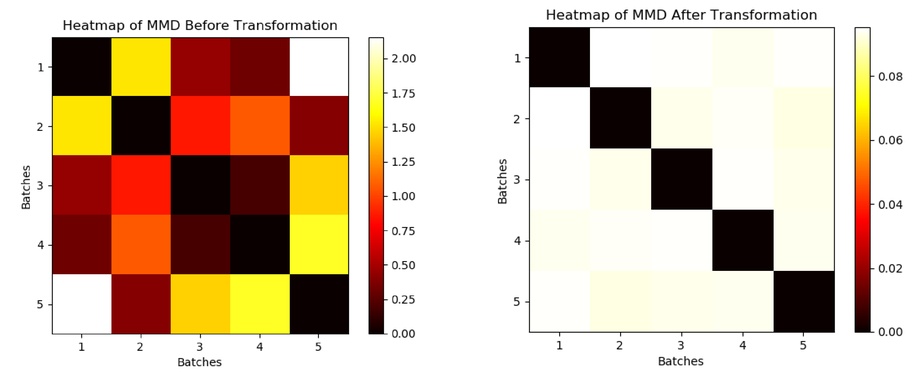
❖ Experimental Setup:

➢ Generate **m** synthetic datasets (**batches**) with **n** samples per dataset where all datasets come from the same multivariate Gaussian distribution and follow $Y = X\beta$ where β is sparse.

➢ The goal to recover $\{i|\beta_i \neq 0\}$. The result is evaluated using **F1 Score**.

(m, n)	Our Method	No Transformation	Combat-seq	Limma	Shaham	Normalization	PCA
(5,10)	**0.537**	0.381	0.424	0.077	0.326	0.061	0.143
(50,10)	**0.727**	0.145	0.313	0.109	0.143	0.204	0.089
(5,100)	**0.88**	0.289	0.445	0.217	0.289	0.231	0.228
(50,100)	**0.909**	0.289	0.759	0.238	0.297	0.16	0.238

F1 Score for Different State-of-the-art batch effect removal methods and different scenarios



Conclusions

- ❑ We propose a novel optimization framework to perform feature selection while removing batch effect via joint optimization of Lasso and MMD of different dataset distributions.
- ❑ Our experiments on synthetic datasets imply that our method outperforms existing state-of-the-art batch effect removal packages such as COMBAT-Seq and Limma drastically.

pengdai@usc.edu