Heterogeneous Domain Adaptation for IHC Classification of Breast Cancer Subtypes

Heterogeneous Domain Adaptation for IHC Classification of Breast Cancer Subtypes

Firat Ismailoglu, Rachel Cavill, Evgueni Smirnov, Shuang Zhou, Pieter Collins, and Ralf Peeters

Abstract—Increasingly, multiple parallel omics datasets are collected from biological samples. Integrating these datasets for classification is an open area of research. Additionally, whilst multiple datasets may be available for the training samples, future samples may only be measured by a single technology requiring methods which do not rely on the presence of all datasets for sample prediction. This enables us to directly compare the protein and the gene profiles. New samples with just one set of measurements (e.g., just protein) can then be mapped to this latent common space where classification is performed. Using this approach, we achieved an improvement of up to 12 percent in accuracy when classifying samples based on their protein measurements compared with baseline methods which were trained on the protein data alone. We illustrate that the additional inclusion of the gene expression or protein expression in the training process enabled the separation between the classes to become clearer.

Index Terms—Breast cancer, classification, heterogeneous domain adaptation, transfer learning, data integration

1 Introduction

Data integration attempts to integrate multiple datatypes that are generated using different omics technology [4]. As datasets with matched samples and measurements using more than one omics technology are becoming much more common, data integration of omics data is a huge current research topic in bioinformatics [19]. In theory, one should be able to improve the classification accuracy using a data integration method, as data integration leads to increase the amount of training data.

Often before applying machine learning methods one would ideally wish to augment the data collected in one situation with data gathered in a different context, the process to make these different datasets usable together is transfer learning [20]. In this paper, we explore the application of *transfer learning* as a data integration method to enhance the classification of breast cancer samples according to their immunohistochemical (IHC) subtypes [7]. Breast cancer was chosen for this study as it is by far the most common cancer type and the leading cause of cancer death in women worldwide with over 1.6 million new cases diagnosed each year [27]. However, depending on the active mutations in the tumour, breast cancers differ considerably in molecular alterations, cellular composition, and clinical outcomes. This diversity is reflected by its subtypes. Thus in turn disclosing its subtypes is very important in terms of diagnosing and treating breast cancer.

Our samples are taken from the breast cancer dataset of the Cancer Gene Atlas [32]. The data we use consist of protein measurements and gene measurements. Here the protein and the gene measurements are represented with very different features. Thus we are specifically interested in heterogeneous domain adaption (HDA) that is one of the main branches of transfer learning [20]. We wish to investigate the improvement in learning from integrating both protein measurements and gene measurements in the learning phase. We specify a *target domain*, the domain of interest from which future samples will come, and a *source domain*, i.e., the auxiliary domain, from which we have only data for the current samples. That is, when proteins are the target domain, then test instances are formed of just the protein measurements.

HDA is a generalization of homogeneous domain adaption where the domains to be adapted share the same feature space but differ in their marginal distributions. Homogeneous domain adaptation is a relatively easy task and a wealth of transfer learning papers address this task [11] including some in the field of bioinformatics [6], [24]. Compared to homoegeneous domain adaptation, HDA is a much more challenging task as the domains lie in different feature spaces and thus more sophisticated methods are needed to identify the commonality between the domains. We emphasize that this paper is one of the first HDA works in the context of bioinformatics.

We use the Class Code Alignment (CCA) [12] method to identify the commonality between the protein and gene data. Although we shall mention the CCA method briefly in this paper, the more in depth details of this method such as its convergence and its working principle can be found in [12]. Rather, in this paper we utilize this method in the context of bionformatics.

CCA is chosen as it makes no assumptions regarding the structure of the datasets that it is adapting, and thus can be used to adapt protein and gene data that lie in different feature spaces. Second, CCA does not only project target and source domains to a common feature space, but also projects the common output space, i.e., the classes itself, into the common feature space. Therefore, in addition to being able to compute the similarity between target and source instances, i.e., protein and gene data, in the common space, CCA also enables one to easily classify test examples finding the nearest class image to them. More precisely, test examples of the target domain are mapped to the common space then assigned to the classes whose images in the common space are the nearest to them. On the other hand, CCA shares the common drawback of HDA methods in that the features in the common space are not interpretable.

In the next section, we reveal the potential of transfer learning for bioinformatics and review the prominent transfer learning applications in this scope. In Section 3, we provide insight about the subtypes of breast cancer based on the immunochemistry biomarkers: ER, PR and HER2. Next, in Section 4, we explain the derivation of the protein and the gene data, then give the preprocessing steps. In Section 5 we first construct the objective function that reflects our goals, then present the CCA algorithm that can handle the objective function. In Section 6, we report the experimental results obtained using the variants of CCA algorithm together with using the baselines. Finally, in Section 7, we sum up the present work.

2 Data Integration and Transfer Learning in Bioinformatics

Transfer learning has enormous potential to be useful in Bioinformatics. There are many current bioinformatics problems where the same elements have been measured by different technologies, for instance, gene expression measured by microarrays and RNAseq. The use of transfer learning to map these different datasets to a common domain, accounting for the different background distributions of measurements and noise with the different technologies might, for instance, allowing the reuse of archived microarray data with the more recently acquired RNAseq. Transfer learning may also be applicable to the widespread problem of batch effects in omics data [13],

- *F. Ismailoglu is with the Department of Computer Engineering, Sivas Cumhuriyet University, Sivas 58000, Turkey. E-mail: fismailoglu@cumhuriyet.edu.tr.*
- *R. Cavill, E. Smirnov, S. Zhou, P. Collins, and R. Peeters are with the Department of Data Science and Knowledge Engineering, Maastricht University, Maastricht 6211, LK, The Netherlands. E-mail: {rachel.cavill, smirnov, shuang.zhou, pieter.collins, ralf.peeters}@maastrichtuniversity.nl.*

Manuscript received 22 July 2016; revised 10 May 2018; accepted 27 Sept. 2018. Date of publication 24 Oct. 2018; date of current version 4 Feb. 2020.
(Corresponding author: Firat Ismailoglu.)
Digital Object Identifier no. 10.1109/TCBB.2018.2877755

where data acquired in one setting is not immediately relatable to data acquired from a similar population in another setting (for instance data processed in different laboratories, at different times or samples collected in different hospitals). Additionally, transfer learning may prove useful when the same experiments are performed on different species, to relate the measurements between the two studies or in toxicogenomics between omics measurements taken after dosing with similar compounds.

Current data integration methods for omics data in bioinformatics tend to come from one of four classes of methods [4]: concatenation methods, which make a combined feature space by concatenating the datasets, possibly with block weighting to prevent the over dominance of the larger dataset and then apply standard methods on the concatenated data. An example of this would be the integration of transcriptomic and metabolomic data by Shen et al. [25] for the classification of glioblastomas. Correlation methods, which look for correlative relationships between items in different datasets, but as these tend to be descriptive of the datasets, they are rarely used when classification is the aim. Pathway methods, which map the data onto biological pathways or networks and analyse these e.g., [3] and finally multi-variate model methods, which generally utilise the PLS family of methods, for instance O2PLS [29], to relate the datasets to each other in a linear model. O2PLS is conceptually the most similar to transfer learning. It is a symmetric model, which represents two datasets in terms of five matrices, the matrix modelling the joint variation, two matrices modelling the unique variation in each dataset and two residual matrices containing the unmodelled variation for each dataset.

Due to all these potential benefits, research on applying homogeneous domain adaptation in bioinformatics is a burgeoning new field and some prominent examples are now described.

The authors in [9] perform transfer rule learning. More precisely, the classification rules derived in a different clinic to predict leukemia or lung cancer are transferred to the target dataset so that the transferred rules are to be used as prior rules. However, in order for the proposed transfer rule learning to be successful, target and source datasets should consists of the same discrete attributes or in the case where these datasets involve the same real valued attributes then these attributes should lie in the same range in order to discretise them harmoniously.

In [6], the authors perform semantic role labeling (SRL) on biomedical articles using the newswire domain. Here the need for domain adaptation methods is self-evident because of the lack of a large biomedical corpus consisting of labeled semantic roles such as agent, cause, experiencer [10]. On the other hand the newswire domain is rich in texts with semantic roles. Another homogenoeous domain adaptation implementation in biology is [24], in which the task is to recognize acceptor splice sites in the less studied model organisms such as C. remanei, P. pacificus and D. melanogaster as the target domains, while exploiting the well studied organism C. elegans as the source domain. This is another example of a homoegeneous domain adaptation task, as the target domains diverged from the source domain taken from C. elegans, thus the gene sequence in each domains show similarity.

To our knowledge, the only work regarding heterogeneous domain adaptation in bioinformatics has been recently presented in [1]. The task here is to predict subcellular localisation of protein sequences. However, in addition to the GO terms, a wide variety of auxiliary sources such as Human Protein Atlas and immunocytochemistry data are exploited. For this purpose, the authors adapt two methods: k-NN TL and SVM TL which were originally proposed in [33]. These two methods also require the associated auxiliary part (the information in terms of the auxiliary data) of a test protein sequence during prediction. This differs significantly from our method which only requires the target domain information to classify an instance.

3 IHC Subtypes of Breast Cancer

There are four common subtypes of breast cancer, these correlate with groups defined by three proteins ER, PR and HER2. [17]: HER2 overexpressing (HER2 positive), triple negative (basal-like), luminal A and luminal B. HER2 overexpressing breast cancer is characterized by the presence of HER2. In triple negative breast cancer, however, all three proteins are absent. The shortest survival times are observed in patients who have HER2 positive or triple negative subtypes. In fact it was reported that triple negative cancers are more aggressive and more resistant to treatment [7]. In contrast to the other two subtypes, luminal A and luminal B were first identified using whole genome gene expression data from microarrays [21] and do not directly relate to these three proteins. However, luminal A subtype tends to have either a high expression of oestrogen receptor (ER) or progesterone receptor (PR) and (HER2) is negative. Luminal B is similar to luminal A in terms of the existence of ER and PR markers, yet it differs with respect to HER2 which tends to be positive in this case. When it comes to the clinical outcomes luminal A is more favourable than luminal B, but patients with luminal A or B tumours respond similarly to endocrine therapies [23].

The three proteins which characterise these subtypes can be measured through immunohistochemistry and thus these classes are assigned in this way in a clinical setting. However, we know that the protein and gene measurements taken from these tissues and measured through high-throughput technologies should allow the prediction of these immunohistochemical markers, and therefore also the subtype classes identified above.

The classification of breast cancer samples, particularly from gene expression data, is a well studied problem. There have been many proposed gene signatures [22], [28], [30]. However the overlap between proposed gene signatures, particularly those predicting prognosis, has often been very small [31]. However, many of these gene signatures have focussed on prognosis rather than classification of subtypes as we do. But, as prognosis strongly relates to these subtypes, the genes/proteins distinguishing these subtypes feature will heavily in these signatures of prognosis.

4 Data and Pre-processing

4.1 Derivation

The data used in this study are taken from The Cancer Genome Atlas (TCGA) [32]. This is a large repository of datasets from many types of tumours. In total there are over 11,000 samples in the database, and breast cancer is the most represented cancer type with over 1000 samples stored. Having removed protein and gene data from samples which were missing data on the IHC biomarker(s), we had 578 samples with protein data available and 419 with gene expression data measured through microarrays.

4.2 Pre-Processing

Originally, there were 284 protein measurements in the protein data. However some proteins in the data are isoforms and in this dataset we see only a single non-zero entry per patient amongst the isoforms of a single protein. This led us to merge the isoform and to construct a single column for a set of isoforms. As a result we end up with 211 distinct protein measurements as listed in Supplementary Table S1, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TCBB.2018.2877755.

The gene dataset contains 17815 measurements per sample. Although the employed domain adaptation algorithm CCA can handle such big data in theory, from the practical point of view, execution times restrict using this amount of data. Therefore, we reduced the number of genes using MSVM-RFE [34] which is an extension of the state-of-the-art gene subset selection method Support Vector

TABLE 1
Class Frequencies in the Protein and the Gene Datasets

Dataset	luminal A	luminal B	Triple Neg.	HER2 Over.
Protein	349	98	95	36
Gene	257	78	60	24

Machine–Recursive Feature Elimination (SVM-RFE) for multiclass problems. MSVM-RFE is an iterative method in which K, (K is equal to 4 in our case) SVM classifiers are built in one-vs-all manner at each iteration. This results in a coefficient matrix W with the entries w_{ij} corresponding the jth coefficient for the ith SVM classifier (ith class), where $i \in \{1, \ldots, K\}$ and $j \in \{1, \ldots, d\}$. Here d shows the dimension at the current iteration. The current features j are then ranked with respect to the ranking criterion $\sum_{r=1}^{K} w_{rj}^2$. Finally the feature, i.e., the gene, with the smallest ranking criterion is eliminated. With MSVM-RFE, we obtained a subset of the gene data consisting of 1069 genes as we observed that the coefficients are relatively larger after this point.

In the protein and the gene data, the patients are labeled according to their immunohistochemically measured ER, PR and HER2 status resulting in 8 potential groups. We used the correlation between these markers and the four breast cancer subtypes: luminal A, luminal B, HER2 overexpressing and triple negative as described in the introduction, to aggregate these groups into clinically meaningful classes. This resulted in the class distributions for the protein and the gene data shown in Table 1.

5 The Method

5.1 Preliminary

In this paper, $\mathbf{X}^T \subseteq \mathbb{R}^{d_T}$ will denote the feature (input) space of the target domain described with d_T features and $\mathbf{X}^S \subseteq \mathbb{R}^{d_S}$ will denote the feature space of the source domain described with d_S features. The target instances $\mathbf{x}^T \in \mathbf{X}^T$ are generated by the target distribution, and the source instances $\mathbf{x}^S \in \mathbf{X}^S$ are generated by the source distribution which is different from the target distribution. Specifically, when the target domain is the protein (resp. gene) then $\mathbf{X}^T$ stands for the feature space of the protein (resp. gene) and d_T, which shows the dimension of the target domain, is equal to 211 (resp. 1069). $\mathbf{X}^S$ in this case stands for the feature space of the source domain: gene (resp. protein) and d_S, which shows the dimension of the source domain, is equal to 1069 (resp. 211). We observe the same classes (labels) in both domains, namely the 4 breast cancer subtypes defined in Section 3. The label of the kth class ($k \in \{1, \ldots, 4\}$) is given by an unique standard unit vector $e_k \in \mathbb{R}^4$ whose kth bit equals 1. The output label space $\mathcal{Y} \in \mathbb{R}^4$ consists of the labels of all the four classes and is common for both domains.

The target training data D_T is formed in the labeled feature space $\mathbf{X}^T \times \mathcal{Y}$ and it consists of n_T training instances $(\mathbf{x}_i^T, y_i)$. Analogously, the source training data D_S is formed in the labeled feature space $\mathbf{X}^S \times \mathcal{Y}$ and it consists of n_S training instances $(\mathbf{x}_i^S, y_i)$. To refer the jth feature of a data instance $\mathbf{x}$, we shall use $\mathbf{x}(j)$.

With the *classification* problem, that we have, we aim to classify unseen test instances from the target domain. This is generally handled by constructing a *classifier* that has been trained on the target training instances. Differently, transfer learning methods also allow for making use of the source instances as training instances.

5.2 Class Code Alignment

In this work, the heterogeneous domain adaptation method being used to adapt the gene and the protein data is Class Code Alignment (CCA)[1] which was previously proposed in [12]. CCA takes its root in the bunching algorithm [18] which was originally derived to handle conventional classification problems in which it is assumed that the training instances and the test instances are generated by the same distribution. In the case where this assumption is violated, then the generalization performance of the classifier is expected to reduce considerably. However, for CCA, this does not hold, as it can make use of training instances that lie in a different feature space.

1. The code can be found at https://github.com/FiratIsmailoglu/CCA

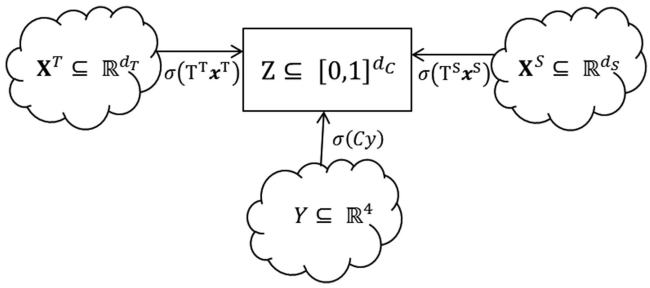

Fig. 1. The projections of the target domain, source domain, and the output label space.

CCA projects the instances of the target domain, those of the source domain and the classes of the output label space into a latent common space Z so that the instances from the same classes will be grouped around the image of their classes. This is illustrated in Fig. 1.

To arrive at the latent common space Z, CCA learns three linear mappings, $T^T : \mathbb{R}^{d_T} \to \mathbb{R}^{d_C}$, $T^S : \mathbb{R}^{d_S} \to \mathbb{R}^{d_C}$, and $C : \mathbb{R}^4 \to \mathbb{R}^{d_C}$, which correspond to the projections of the target domain, the source domain and the output label space respectively, given that d_C is the dimension of the common space. These mappings are applied together with the logistic transformation σ (see Fig. 1). The logistic transformation σ is a function from $\mathbb{R}^{d_C}$ to $(0,1)^{d_C}$ defined by:

$$\sigma(\mathbf{w}) = (\sigma(\mathbf{w}(1)), \ldots, \sigma(\mathbf{w}(d_C)))^T,$$

where $\sigma(\mathbf{w}(k)) = (1 + e^{-\mathbf{w}(k)})^{-1}, \forall k \in \{1, \ldots, d_C\}$.

Since the mappings T^T, T^S, and C are applied together with the logistic transformation σ, the images in the latent common space Z are represented by multivariate [0,1]-valued random variables. Thus, in turn, the instances, i.e., the protein and the gene data, all become *comparable* in Z and the *KL divergence* can be used as the similarity measure in this space.

As mentioned above and illustrated in Fig. 1, we aim that the instances, i.e., the gene and protein measurements, from the same breast cancer subtypes will be nearby in the latent common space Z, while those from the different classes will be far apart. This is desirable to build an accurate classifier in Z. We therefore minimize the distance between the projections of instances and the projections of their classes in terms of the KL divergence measure. This requires us to calculate the loss between each instance label pair, which is reflected by the loss function l defined as: $l((\mathbf{x}, y) \,|\, C, T) = KL[\sigma(Cy) \| \sigma(T\mathbf{x})]$. Here the KL divergence between the images of $\mathbf{x}$ and y under the projection matrices T and C is given as:

$$KL[\sigma(Cy) \| \sigma(T\mathbf{x})] = \sum_{k=1}^{d_C} \left[\sigma(C(k,\cdot)y) \log \left(\frac{\sigma(C(k,\cdot)y)}{\sigma(T(k,\cdot)\mathbf{x})} \right) + (1 - \sigma(C(k,\cdot),y)) \log \left(\frac{1 - \sigma(C(k,\cdot)y)}{1 - \sigma(T(k,\cdot)\mathbf{x})} \right) \right],$$

where $C(k,\cdot)$ and $T(k,\cdot)$ show the kth row of the projection matrices C and T respectively. Recall that we map both the target and the source data, thus each mapping contributes to the total loss which is given as:

$$\mathcal{L}(D_T \,|\, C, T^T) + \mathcal{L}(D_S \,|\, C, T^S) = \sum_{i=1}^{n_T} l(\mathbf{x}_i^T, y_i \,|\, C, T^T) + \sum_{i=1}^{n_S} l(\mathbf{x}_i^S, y_i \,|\, C, T^S). \tag{1}$$

By using projection matrices T^T, T^S and C that minimise the loss function (1), the projection of the target domain, the source domain and the output label space to Z will guarantee that instances from the same class are projected to similar locations in Z. However, we also need a mechanism to prevent instances of different classes from getting too close to each other in Z after the projection. This is important to achieve a better classification in the common space. In this respect we introduce a regularization term over C, since the location of class images in Z are determined by the projection matrix C. In fact $\sigma(Cy)$, which shows the location of the class with label $y \in \mathcal{Y}$ in Z, is actually the image of the column of C that corresponds to that class, as the classes are encoded as the standard unit vectors in $\mathbb{R}^4$. Thus we want C to be near to a reference matrix C_{ref} whose columns are set in advance to be distant to one another. This leads us to add the following regularization term to the loss function (1):

$$\mathcal{L}(D_T \,|\, C_{ref}, C) + \bar{\mathcal{L}}(D_S \,|\, C_{ref}, C), \tag{2}$$

with $\mathcal{L}(D \,|\, C, C_{ref}) = \sum_{(\mathbf{x},y)\in D} KL\big[\sigma(Cy) \| \sigma\big(C_{ref}\ y\big)\big]$.

Tying the above regularization term (2) to the loss function (1) with a paramter α that controls the relative importance of the regularization term, we arrive at the final objective function:

$$\mathcal{O}(D_T, D_S \,|\, T^T, T^S, C) = \mathcal{L}(D_T \,|\, C, T^T) + \mathcal{L}(D_S \,|\, C, T^S) + \alpha\big(\mathcal{L}(D_T \,|\, C_{ref}, C) + \mathcal{L}(D_S \,|\, C_{ref}, C)\big). \tag{3}$$

CCA is an alternating minimization type algorithm. Alternating between T^T, T^S, and C CCA iteratively improves the projection of the target domain, the source domain and the output label space in the IMPROVE-T and the IMPROVE-C steps respectively, where the iteration number is controlled by the parameter $maxIterOut > 1$. The IMPROVE-T steps are of of gradient descent type with backtracking line search for determining step size at each iteration. The backtracking line search is parameterized by β and the number of iteration in the gradient descents is controlled by $maxIterIn > 1$. Unlike the IMPROVE-T steps, the IMPROVE-C step solves the objective function analytically, since for a fixed choice of T^T and T^S the objective function (3) has a unique stationary point ([12]).

5.2.1 Clustering and Classification and with CCA

CCA outputs three projection matrices T^T, T^S and C. T^T and T^S enable one to transform the target and the source instances to the common space where they are all comparable, despite their representations being very different in their original spaces. Specifically, we can project the protein and the gene data represented with different features to a common space by means of the aforementioned projection matrices, so the protein and the gene data are made comparable. Hence, any clustering algorithm can be applied to *cluster protein and the gene data* in the common space after the application of CCA.

CCA is a classifier independent approach. This means that any classifier can be built after having performed CCA, since source and target instances now lie in the same common space Z. Nevertheless, we will define three classification methods as showcases. The first of the three is the classical *nearest neighbor* classifier that compares test target instances with target and source training instances using the KL divergence. This classifier will be called as CCA.IDS, where IDS stands for *instance decomposition schemes*. The second method builds *random forest* classifier again using both target and source instance. This will be called as CCA.RF. The final classification method of this type is CCA.NB, where NB stands for *naive Bayes*.

Algorithm 1. The CCA Algorithm

CCA
Input: target data D_T, source data D_S,
initial projection matrices T_0^T and T_0^S, reference projection matrix C_{ref},
step parameter $\beta \in (0,1)$, regularization parameter $\alpha > 0$, and iteration numbers $maxIterOut > 1$ and $maxIterIn > 1$.
Output: projection matrices T_t^T, T_t^S and C_t for $t = maxIterOut$.
$C_0 := C_{ref}$;
for $t := 1$ to $maxIterOut$ **do**
  $T_t^T :=$ IMPROVE-T$(D_T, C_{t-1}, T_{t-1}^T, maxIterIn)$;
  $T_t^S :=$ IMPROVE-T$(D_S, C_{t-1}, T_{t-1}^S, maxIterIn)$;
  $C_t :=$ IMPROVE-C$(D_T, D_S, \alpha, C_{ref}, T_{t-1}^T, T_{t-1}^S)$;
end for
return T_t^T, T_t^S, and C_t.

IMPROVE-T

Input: data D,
projection matrices C and T, and
iteration number $maxIterIn > 1$.
Output: projection matrix T.
let d_C and d be the sizes of T;
for $t := 1$ to $maxIterIn$ **do**
  Set η equal to 1;
  for $i := 1$ to d_C and $j := 1$ to d **do**
    $G(i,j) = \sum_{(\mathbf{x},y)\in D} \mathbf{x}(j)\Big(\sigma\big(-C(i,\cdot)y\big)\sigma\big(T(i,\cdot)\mathbf{x}\big) - \sigma\big(C(i,\cdot)y\big)\sigma\big(-T(i,\cdot)\mathbf{x}\big)\Big)$;
  end for
  while $\mathcal{L}(D_T|C, T - \eta G) > \mathcal{L}(D_T|C,T) - \frac{\eta}{4}\|G\|^2$ **do**
    $\eta := \beta\eta$;
  end while
  $T := T - \eta \times G$;
end for
return T.

IMPROVE-C

Input: target data D_T, source data D_S,
regularization parameter $\alpha > 0$,
reference projection matrix C_{ref}, projection matrices T^T and T^S.
Output: projection matrix C.
let d_C and d be the sizes of C_{ref};
for $i := 1$ to d_C and $j := 1$ to d
  $D_{Tj} := \big\{\big(\mathbf{x}^T, y\big) \in D_T \,|\, y = e_j \wedge e_j \in \mathcal{Y}\big\}$;
  $D_{Sj} := \big\{\big(\mathbf{x}^S, y\big) \in D_S \,|\, y = e_j \wedge e_j \in \mathcal{Y}\big\}$;
  $C(i,j) := \frac{1}{1+\alpha}\left(\alpha C_{ref}(i,j) + \frac{\sum_{(\mathbf{x}^T,y)\in D_{Tj}} T^T\mathbf{x}^T + \sum_{(\mathbf{x}^S,y)\in D_{Sj}} T^S\mathbf{x}^S}{|D_{Tj}| + |D_{Sj}|}\right)$;
end for
return C.

In addition to the aforementioned methods that make use of T^T and T^S, we propose another classification method which considers the mapping of the label space achieved through the matrix C. The second classification method that CCA offers relies on the use of C and T^T. Due to C, one can obtain the images of the classes, i.e., the class prototypes, in Z. To classify a test instance, the instance can be simply compared with the class prototypes in terms of the KL divergence after having been mapped to Z through T^T. This gives CCA. CDS illustrated in Fig. 2 where the suffix CDS stands for *class decomposition schemes*. Note also that C, which in our case determines the location of the classes of the breast cancer subtypes, carries

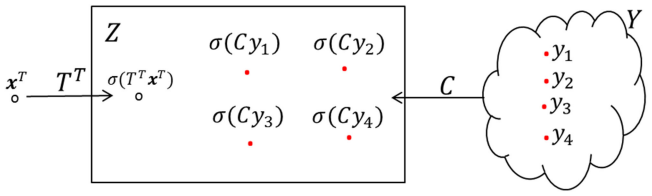

Fig. 2. Classification with CCA.CDS

information from both target and source data; i.e., the gene and protein measurements. See the IMPROVE-C step of CCA in Algorithm 1.

6 Experiments

We used the gene and the protein datasets in both target and source domain roles. That is, each was used as the target and the source data in turn. We compared the CCA variants, i.e., CCA.IDS, CCA.RF, CCA.NB and CCA.CDS, that use information from both protein and gene data against their counterparts that use information only from the target data. These counterparts will be later explained in the following section and will be called as baseline methods. The reported results of these comparisons are based on 5-fold cross-validation. It is organized as follows: for each $k \in \{1, \ldots, 5\}$ of the inner cycle of the cross validation the training data consists of 4 randomly chosen folds of the target data and the same source data (both projected in the common space Z). The remaining fifth fold of the target data (also in Z) is used for testing. In what follows, we give the set-up which is necessary to reproduce the results, as well as the baseline methods with their settings.

6.1 Experimental Setup and the Baselines

One needs to determine the reference matrix C_{ref} and to tune the regularization parameter α when using CCA. For C_{ref}, we opted for the exhaustive code matrix that is explained in [8]. Exhaustive code matrices consist of 0-1 bits with the size of $2^{K-1} - 1 \times K$, where K is the number of classes. K is equal to 4 in our case, as there are 4 breast cancer subtypes. Thus C_{ref} has 7 rows which is equal to the dimension of the common space. The columns of the exhaustive code matrices are distant from one another, in fact each column is 4 bits away from the other columns in our case. This ensures that instances of different breast cancer subtypes will be projected to different locations in the common space.

In the experiments, we tuned the regularization parameter α in CCA testing different values from the set $\{0.01, 0.1, 1, 10, 50, 100\}$. For each test, we performed five fold cross-validation using a different α. Tables 2 and 3 show accuracies corresponding to each α for each variant of CCA. These tables suggest that the ideal choice of α depends on the classification method built on top of CCA, as the highest accuracy is achieved with different values of alpha. Also note that the confusion matrices regarding to Tables 2 and 3 can be found in Supplementary Table S2, available online.

In addition to the conventional accuracies shown in Tables 2 and 3, we also evaluated the performances of all the classifiers in terms of *balanced accuracy*, as both protein and gene data suffer from imbalanced classes (See Table 1). Concretely, balanced accuracy is the arithmetic average of the class specific accuracies thus is invariant to class distributions and is often preferable to the conventional accuracy in the presence of imbalanced data [2]. Formally, given a confusion matrix M, the test statistic λ of balanced accuracy is given by

$$\hat{\lambda} = \frac{1}{K} \sum_{j=1}^{K} \frac{M(j,j)}{\sum_{i=1}^{K} M(i,j)}, \tag{4}$$

where K is the number of classes. Here the rows (resp. columns) of M correspond to the actual (resp. predicted) classes. The equivalent measures with balanced accuracies are given in Tables 4 and 5. Again the results are obtained using 5-fold cross-validation runs.

The CCA variants were compared against the following baseline classifiers, where each of which is trained only using target training instances.

- *KNN.T* is the classical k-nearest neighbour classifier that finds the nearest k (target) training instances to the test instance using the euclidean distance. Here k is set to 1.
- *Bunching.T* is the original Bunching algorithm given in [18]. Bunching.T projects only the target data and the output label space to a common space. The classification rule in Bunching.T is comparable to that in CCA.CDS, as both assign the test instance to the class whose image is nearest to the projection of the test instance.
- *RF.T* is a random forest algorithm with the following parameters. The number of the trees generated is 30 and the number of features that each tree uses is 15 or 32, which are the approximately square roots of 211 and 1069 respectively, when the target domain is the protein data (given with 211 features) and the gene data (given with 1069 features) respectively.
- *NB.T* is a Naive Bayes classifier, where the features are assumed to follow normal distribution for each class.

TABLE 2
Accuracy for Different Values of α

Method	0.01	0.1	1	10	50	100
CCA.IDS	66.96	68.14	69.03	**69.26**	69.19	68.89
CCA.CDS	73.20	**74.01**	**74.01**	73.53	73.65	73.58
CCA.RF	**74.38**	73.01	70.58	73.52	73.52	73.86
CCA.NB	74.73	74.57	74.74	**75.43**	74.56	74.90

Target data: protein data, Source data: gene data. The best results are given in bold.

TABLE 3
Accuracy for Different Values of α

Method	0.01	0.1	1	10	50	100
CCA.IDS	67.51	70.40	74.70	75.41	73.51	**75.65**
CCA.CDS	72.55	74.69	**75.89**	74.28	74.38	74.27
CCA.RF	71.60	78.04	74.46	76.37	76.37	**78.99**
CCA.NB	73.50	72.78	77.33	77.33	77.79	**78.51**

Target data: gene data, Source data: protein data. The best results are given in bold.

TABLE 4
Balanced Accuracy for Different Values of α

Method	0.01	0.1	1	10	50	100
CCA.IDS	58.29	58.46	**61.99**	61.30	60.61	61.06
CCA.CDS	56.5	**64.31**	57.63	55.94	57.41	55.27
CCA.RF	**64.58**	64.23	60.43	63.64	63.32	62.96
CCA.NB	63.73	67.70	**68.20**	68.14	67.57	67.62

Target data: protein data, Source data: gene data. The best results are given in bold.

TABLE 5
Balanced Accuracy for Different Values of α

Method	0.01	0.1	1	10	50	100
CCA.IDS	56.96	59.80	61.51	**67.66**	64.29	63.60
CCA.CDS	61.42	62.41	**63.12**	62.01	61.49	62.70
CCA.RF	55.95	64.85	62.17	62.38	62.91	**66.14**
CCA.NB	60.36	54.39	58.2	57.63	57.81	**63.61**

Target data: gene data, Source data: protein data. The best results are given in bold.

TABLE 6
A Comparison between Our Proposed Method and the Baseline ML Methods Shown with the Classification Accuracies in Percent

Target	Source	KNN.T	CCA.IDS	Bunching.T	CCA.CDS	RF.T	CCA.RF	NB.T	CCA.NB
Protein	Gene	62.39	69.26	62.98	74.01	71.42	74.38	63.25	75.43
Gene	Protein	72.16	75.65	73.87	75.89	76.42	78.99	77.01	78.51

TABLE 7
A Comparison between Our Proposed Method and the Baseline ML Methods Shown with the Balanced Classification Accuracies in Percent

Target	Source	KNN.T	CCA.IDS	Bunching.T	CCA.CDS	RF.T	CCA.RF	NB.T	CCA.NB
Protein	Gene	55.26	61.99	61.87	64.31	56.23	64.58	61.49	68.20
Gene	Protein	65.14	67.66	64.49	63.12	60.63	66.14	69.85	63.61

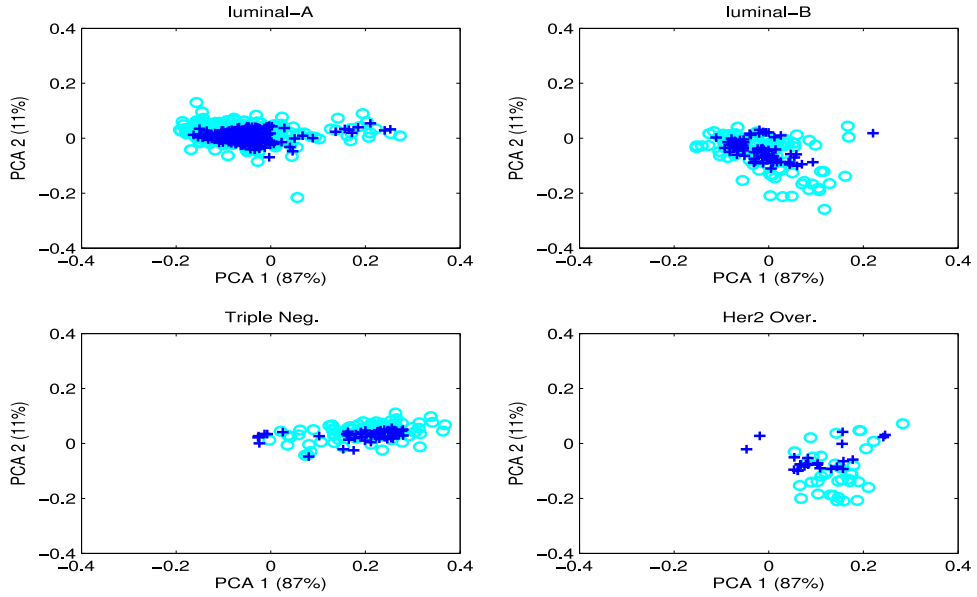

Fig. 3. The protein and the gene data in the common space. In each subplot, the protein samples are marked with circles and the gene samples are marked with plus.

6.2 Results

Table 6 shows the best classification accuracies of B.HDA.NN, CCA.CDS, CCA.RF and CCA.NB together with those of the baselines explained above. The accuracies are obtained using 5-fold cross-validation. The results reported in Table 6 show that using the protein data and the gene data together in the training stage results in better classification of the breast cancer subtypes. In fact, the CCA variants that use both the protein data and the gene data in training always outperform their counterparts, i.e., the baselines. The improvement in the breast cancer classification ranges from 2 to 12 percent. Importantly, this improvement is achieved even when the gene data, which has rich representation, is added to the much smaller protein data.

The balanced accuracy results are parallel to the conventional accuracy results, as shown in Table 7. Indeed, it is fair to claim that using protein and gene data together in training results in obtaining a classifier that is more accurate for each breast cancer subtype, especially when the protein data is the target domain. In this case, the balanced accuracy is improved by up to 8 percent over the respective baseline.

Fig. 3 shows four projections of the common space where in each case instances of one breast cancer subtype are shown using the PCA dimensions. Clearly, for all breast cancer subtypes, the instances of the protein data and that of the gene data are mapped nearby in the common space, as desired. Additionally, this may suggest that the proposed method can be used to find out similar gene expression patterns for a given protein expression pattern or vice versa. Considering the centres of the subtypes, Fig. 3 reveals that for both the protein and the gene data breast cancers of type luminal A and luminal B are near to each other which is to be expected given the ambiguity between the original definitions of these subtypes based on the three immunohistochemical markers. Also, we observe that the triple negative instances are far apart from the luminal A and luminal B instances, as expected.

7 Conclusion

In the present paper we applied a heterogeneous domain adaptation method CCA [12] on the protein and the gene data from the cancer genome atlas with the goal of improving classification of breast cancer subtypes. The projection of both the protein and the gene data to the same (latent) common space, through CCA allows for the visualisation and comparison between the protein and the gene samples in this space despite of their different feature sets in their respective original domains. We also demonstrate benefits of using transfer learning methods in the context of bioinformatics, and have discussed other potential areas for exploration as bioinformatics has not yet fully benefited from transfer learning methods. This paper presents one of the first heterogeneous domain adaptation works in bioinformatics. In the experiments, our results show that using the protein and the gene data together in the training phase improves the classification of the breast cancer subtypes by up to 12 percent when compared to the models that use either the gene data or the protein data. In addition to that,

relying on the balanced accuracy results, we have shown that mapping the protein data and the gene data to a common space through CCA provides better discrimination of the breast cancer subtypes. Therefore, this paper shows the promise of using heterogeneous domain adaptation methods in bioinformatics, in particular as a technique to improve classification accuracy when multiple omics datasets are available for training.

REFERENCES

[1] L. M. Breckels, S. B. Holden, D. Wojnar, C. M. Mulvey, A. Christoforou, A. Groen, M. W. B. Trotter, O. Kohlbacher, K. S. Lilley, and L. Gatto, "Learning from heterogeneous data sources: An application in spatial proteomics," *PLoS Comput. Biol.*, vol. 12, no. 5, 2016, Art. no. e1004920.

[2] K. H. Brodersen, C. S. Ong, K. E. Stephan, and J. M. Buhmann, "The balanced accuracy and its posterior distribution," in *Proc 20th Int. Conf. Pattern Recognit.*, 2010, pp. 3121–3124.

[3] R. Cavill, A. Kamburov, J. K. Ellis, et al., "Consensus-phenotype integration of transcriptomic and metabolomic data implies a role for metabolism in the chemosensitivity of tumour cells," *PLoS Comput. Biol.*, vol. 7, 2011, Art. no. 12.

[4] R. Cavill, D. Jennen, J. Kleinjans, and J. J. Briedé, "Transcriptomic and metabolomic data integration," *Briefings Bioinf.*, vol. 17, pp. 891—901, 2016.

[5] C. C. Chang, C. J Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, pp. 27:1–27:27, 2011.

[6] D. Dahlmeier and H. T. Ng, "Domain adaptation for semantic role labeling in the biomedical domain," *J. Bioinf.*, vol. 26 no. 8, pp. 1098–1104, 2010.

[7] R. Dent, M. Trudeau, K. I. Pritchard, W. M. Hanna, H. K. Kahn, C. A. Sawka, L. A.Lickley, E. Rawlinson,P. Sun, and S. A. Narod, "Triple-negative breast cancer: Clinical features and patterns of recurrence," *Clinical Cancer Res.*, vol. 13, pp 4429–4434, 2007.

[8] T. G. Dietterich and G. Bakiri, "Solving multiclass learning problems via error-correcting output codes," *J. Artif. Intell. Res.*, vol. 2, pp 263–286, 1995.

[9] P. Ganchev, D. Malehorn, W. L. Bigbee, and V. Gopalakrishnan, "Transfer learning of classification rules for biomarker discovery and from molecular profiling studies," *J. Biomed. Inform.*, vol. 44, pp. S17–S23, 2011.

[10] D. Gildea and D. Jurafsky, "Automatic labeling of semantic roles," *Comput. Linguistic*, vol. 28, pp 245–288, 2002.

[11] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 2066–2073.

[12] F. Ismailoglu, E. Smirnov, R. Peeters, S. Zhou, and P. Collins, "Heterogeneous domain adaptation based on class decomposition schemes," in *Proc Pacific-Asia Conf. Knowl. Discovery Data Mining*, 2018, pp. 169–182.

[13] J. T. Leek, R. B. Scharpf, H. C. Bravo, et al., "Tackling the widespread and critical impact of batch effects in high-throughput data," *Nature Rev. Genetics*, vol. 11, pp. 733–739, 2010

[14] S. Mei, W. Fei, and S. Zhou, "Gene ontology based transfer learning for protein subcellular localization," *BMC Bioinf.*, vol. 12, pp: 44–56, 2011.

[15] S. Mei, "Multi-kernel transfer learning based on Chou's PseAAC formulation for protein submitochondria localization," *J. Theoretical Biol.*, vol. 293, pp. 121–130, 2012.

[16] S. Mei, "Probability weighted ensemble transfer learning for predicting interactions between HIV-1 and human proteins," *PLoS One*, vol. 8, no. 11, 2013, Art. no. e79606.

[17] A. A. Onitilo, J. M. Engel, R. T. Greenlee, and B. N. Mukesh, "Cancer subtypes based on ER/PR and Her2 expression: Comparison of clinicopathologic features and survival," *Clinical Med. Res.*, vol 7, pp 4–13, 2009.

[18] O. Dekel and Y. Singer, "Multiclass learning by probabilistic embeddings," in *Proc 15th Int. Conf. Neural Inf. Process. Syst.*, 2002, pp. 969–1000.

[19] B. O. Palsson and K. Zengler, "The challenges of integrating multi-omics data sets," *Nature Chemical Biol.*, vol. 6, pp 787–798, 2010

[20] S. Pan and Q. Yang "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp 1345–1359, Oct. 2010.

[21] C. M. Perou, T. SÃçrlie, and M. B. Eisen, "Molecular portraits of human breast tumours," *Nature*, vol. 406, pp 747–752, 2000.

[22] A. Prat, M. J. Ellis, and C. M. Perou, "Practical implications of gene-expression-based assays for breast oncologists," *Nature Rev. Clinical Oncology*, vol. 9, pp. 48–57, 2011.

[23] A. Prat, M. C. Cheang, M. Martin, J. S. Parker, E. Carrasco, R. Caballero, S. Tyldesley, K. Gelmon, P. S. Bernard, T. O Nielsen, and C. M. Perou, "Prognostic significance of progesterone receptor-positive tumor cells within immunohistochemically defined luminal a breast cancer," *J. Clinical Oncology*, vol. 31, pp. 203–209, 2012.

[24] G. Schweikert, C. Widmer, G. Rätsch, and B. Scholköpf, "An empirical analysis of domain adaptation algorithms for genomic sequence analysis," in *Proc. Advances Neural Inf. Process. Syst. 21*, 2009, pp. 1433–1440.

[25] R. Shen, Q. Mo, and N. Schultz, et al., "Integrative subtype discovery in glioblastoma using iCluster," *PLoS One*, vol. 7, no. 4, 2012, Art. no. e35236.

[26] X. L. Shi, Q. Liu, W. Fan, P. S. Yu, and R. Zhu, "Transfer learning on heterogeneous feature spaces via spectral transformation," in *Proc. IEEE Int. Conf. Data Mining*, 2010, pp. 1049–11054.

[27] R. Siegel, L. Miller, and J. Ahmedin, "Cancer statistics," *A Cancer J. Clinicians*, vol. 66, pp 7–30, 2016.

[28] C. Sotiriou and L. Pusztai, "Gene-expression signatures in breast cancer," *New England J. Medicine*, vol. 360, pp. 790–800, 2009.

[29] J. Trygg, S. Wold, "O2-PLS, a two-block (X-Y) latent variable regression (LVR) method with an integral OSC filter," *J. Chemometrics*, vol. 17, pp. 53–64, 2003.

[30] B. Weigelt, F. L. Baehner, and J. S. Reis-Filho, "The contribution of gene expression profiling to breast cancer classification, prognostication and prediction: A retrospective of the last decade," *J. Pathol.*, vol. 220, pp. 263–280, 2010.

[31] B. Weigelt, L. Pusztai, A. Ashworth, et al.,"Challenges translating breast cancer gene signatures into the clinic," *Nature Rev. Clinical Oncology*, vol. 9, pp. 58–64, 2011.

[32] J. N. Weinstein, E. A. Collisson, G. B. Mills, et al., "The cancer genome atlas pan-cancer analysis project," *Nature Genetics*, vol. 45, pp. 1113–1120. 2013

[33] P. Wu and T. G. Dietterich, "Improving SVM accuracy by training on auxiliary data sources," in *Proc. 21st Int. Conf. Mach. Learn.*, 2004, Art. no. 110.

[34] X. Zhou and D. P. Tuck, "Gene expression MSVM-RFE: Extensions of SVM-RFE for multiclass gene selection on DNA microarray data," *Bioinf.*, vol. 23, no. 9, pp 1106–1114, 2007.