**Project in Bioinformatics**
**Bioinformatics Research Centre**

# Prediction and Investigation of Cancerous Metastasis Gene Expression Data through Machine Learning Methods

By Mathias Byskov Nielsen

December 13, 2019

https://github.com/mathiasbyskov/PiB

# Abstract

Cancer of unknown primary (CUP) is defined as the situation where a cancerous tumor is confirmed histologically, but regular diagnostic tools fails to identity the primary site of the tumor [3]. This type of cancer accounts for $3-5\%$ of all tumors and the median overall survival rate is only 6 months [5]. Deciding the site of origin is crucial in order to provide the patients with the best treatment possible. The present primary strategy to decide on the treatment is right now based on light microscopy and immunohistochemistry (IHC) staining [4]. Although, a new era by using molecular profiling in the investigation of primary site has gained a lot of interest the last couple of years and is thought to have the potential to replace the current methods [7]. This is the approach used in this project.

The Human Cancer Metastatis Databse was the source of data in this project. This is the first public database providing published cancer metastasis expression profiles, although it still lacks the ability of providing large-samples datasets derived from metastasis tumors [9]. The dataset used was samples from the GPL570 platform, that resulted in a dataset with the dimensions $192 \times 54680$. It was cleaned to the dimensions $192 \times 44142$. An attempt to increase the sample-size by merging different platforms was tried without any luck.

The dimensionality of the dataset was reduced by a mix of six different feature selection and extraction methods including: A simple extraction based on the Coefficient of Variance score, principal component analysis, multidimensional scaling, isomap, locally linear embedding and t-SNE. This resulted in 6 different datasets with the dimensions $192 \times 100$. All datasets were divided into a training and validation set with a 80/20 split in a stratified manner. Locally linear embedding and t-SNE turned out to visually distinguish the primary sites of the samples best in 2 dimensions (Figure 5). An anaylysis of the first two principal components from PCA showed that 97.1% of the variance was captured in the 100 embedded dimensions.

Classification was performed with 4 different methods including: logistic regression, random forest, xgboost and support vector machine on all the 6 different datasets. They were optimized with k-fold crossvalidation with $k = 5$ and the best training CV accruacy turned out to be classifying with xgboost on the PCA dataset, which yielded an accuracy of 94.1%. Predicting on the validation sets proved that the methods logistic regression, random forest and xgboost yielded best results on the PCA and locally linear embedding datasets. The training and validation accuracies were between $91\% - 94\%$.

Furthermore, inference was investigated using a decision tree on the feature selection dataset. This approach identified 2 genes important for distinguishing breast-cancer and skin-cancer samples from the other that also previously has been mentioned in literature and is known to be important in identifying tumorous cells [16, 17].

# Contents

# 1 Introduction

The main focus in this project was to extract gene expression profiles from samples derived from a metastatic tumor and try to correctly classify the primary site of those samples. The theory, extraction, wrangling and workflow of this project is described in this section. Next section focus on the methodology of the statistical approaches used in the project. The third section present the results obtained from the study. Lastly, a final conclusion based on the results is provided. It is assumed throughout the report, that the reader has some basic biological knowledge and is familiar with general statistical and machine learning terminology. All code to extract the data or produce any results can be found in the Github repository provided on the front-page.

## 1.1 Cancer of Unknown Primary

Cancer is the well-known disease in which cells divide without control and are capable of invading nearby tissue [1]. These cancerous cells can spread throughout the body via the blood or lymph and create a metastasis. A metastasis is defined as the process, where cancer cells break away from the original primary tumor and form a new tumor in different tissue or at another organ. After this process has happened the metastatic tumor belongs the same category of cancer as the primary tumor did [2]. Hence, when a primary tumor arises in breast tissue, breaks through and metastasize in the lung-tissue, it should still be considered as breast-cancer.

Cancer of Unknown primary (CUP) is defined as the situation where a cancerous tumor is confirmed histologically, but the regular diagnostic tools fails to identify the primary origin site of the tumor. Therefore, the identification of CUP is based on a two-step approach: (1) First it must be histologically confirmed, that the tumor presumably originated somewhere else in the body (that it is distinct from the surrounding tissue), and (2) the tissue where it originated remains unknown. Step 1 is carried out through light microscopy performed by a pathologist [3].

The standard approach is to classify the CUP into 5 distinct morphological subtypes based on light microscopy and immunohistochemistry (IHC) staining. These five subtypes each provides some potential occult primary sites to help deciding which treatment the patient should have [4]. The 5 subtypes are present in the table below, where the potential primary sites also are given. The table is from the F. Losa et. al. article [3].

| Tumor Type | % | Potential occult primary (site/types) |
|---|---|---|
| Well or moderately differentiated adenocarcinomas | 60 | Lung, pancreas, hepatobiliary tree, kidney, colon, ovary, breast |
| Squamous-cell carcinomas | 5 | Head and neck, lung, cervix, penis, vulva, bladder |
| Carcinomas with neuroendocrine differentiation | 1 | Pancreas, GI tract, lung |
| Poorly differentiated carcinomas | 25-30 | Adenocarcinoma, melanoma, sarcoma, lymphoma |
| Undifferentiated neoplasm | 5 | Carcinoma, lymphoma, germ-cell tumors, melanoma, sarcoma, embryonal carcinoma |

Table 1: Tumor type and potential occult primary site [3].

CUP accounts for 3-5% of all tumors and the median overall survival rate is 6 months, which makes the prognosis for cancer patients suffering from CUP very bad [5]. This is primarily due to the fact, that patients with CUP often does not belong to a specific subgroup of cancer patients with a standard treatment schedule [6]. A new era regarding using molecular profiling in the investigation of the primary tumor has been of great interest the last decade. Here models based on DNA-sequences, gene-expression and epigenetics has been used to evaluate the primary site. Although, this is still an immature and developing field, since the vast majority of the models are based on short series, retrospective studies, or phase II studies [7].

Although the field is still immature and developing there has been confirmed some promising results within the field and models have been useful to determine the primary tumor with a degree of accuracy of 82%-97% [7]. In Handorf et al they compared gene expression profiling tests with immunochemistry (IHC) tests. Here gene expression profiling proved to be superior when compared to IHC-tests. Haghighi et al proved that multi-omics data can improve the classification of cancer types, which also might be very interesting in the future to come [8].

## 1.2 Human Cancer Metastasis Database

The human cancer metastasis database (HMCDB) was the source of data used in this project. HMCDB is the first public database providing published cancer metastasis expression profiles [9]. It aims to collect all the microarray and RNA sequencing profiles of cancerous metastasis tissue in one repository. This ensures to make the analyzing of metastasis related gene-profiling easier and more efficient. They expect the database to facilitate the identification of metastasis related genes and their impact on metastasis development. Although, general investigation of gene-inference in regard to metastasizes was not the purpose of this project, the database was still useful in order to extract gene-profiling data with belonging information regarding the primary and metastasis site of the tumor.

The sources that contributed to the HMCDB were NCBI Gene Expression Omnibus (GEO), Sequence Read Archive (SRA) and The Cancer Genome Atlas (TCGA). Raw data was extraced from those datasources compiled in the HMCDB. The workflow of constructing the database can be seen in figure 1. The figure shows that the largest contribution of databases was derived from GEO and SRA. The entire database contains 11.425 gene-expression samples derived from 60 distinct platforms.

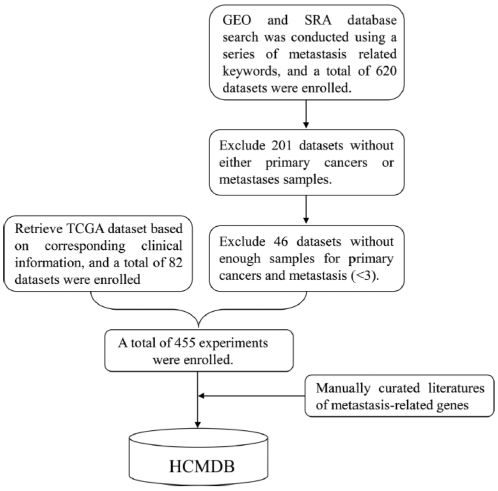

GEO and SRA database search was conducted using a series of metastasis related keywords, and a total of 620 datasets were enrolled.
Exclude 201 datasets without either primary cancers or metastases samples.
Retrieve TCGA dataset based on corresponding clinical information, and a total of 82 datasets were enrolled
Exclude 46 datasets without enough samples for primary cancers and metastasis (<3).
A total of 455 experiments were enrolled.
Manually curated literatures of metastasis-related genes
HCMDB

Figure 1: Flow-chart of constructing HCMDB [9].

Since the samples comes from different platforms, the first main-objective was to identify the platforms that would be interesting to extract the samples from. Not all platforms can be merged, and even if different platforms has merge-able columns (using the same probes for measuring the expression of different genes) it is not always suitable to do so (elaborated in next section). The platforms with > 100 samples were extracted and are summarized in Table 2. Only metastasis samples was of interest in this project and from Table 2 it is obvious that 6 platforms has > 100 metastasis samples. Therefore, these were chosen as candidate-platforms of interest and were further investigated due to their distribution of primary sites of the metastases.

| Platform | Metastasis | | Primary | | |
|---|---|---|---|---|---|
| ID | Normal | Tumor | Normal | Tumor | Total |
| TCGA | 0 | 0 | 2 | 198 | 200 |
| GPL96 | 26 | 167 | 29 | 71 | 293 |
| GPL570 | 6 | 192 | 18 | 390 | 606 |
| GPL8432 | 0 | 104 | 0 | 0 | 104 |
| GPL10379 | 0 | 207 | 0 | 0 | 207 |
| GPL10558 | 0 | 263 | 0 | 0 | 263 |
| GPL14951 | 0 | 42 | 0 | 71 | 113 |
| GPL15659 | 0 | 145 | 0 | 0 | 145 |
| GPL16744 | 0 | 58 | 0 | 46 | 104 |

Table 2: Categories by different platforms

Table 3 shows the distribution of the primary sites for all of the candiate-platforms, where all the tumorous metastasis samples were extracted. Unfortunately, it turned out that the distributions of primary sites were highly skewed, and from this the 6 candidates was narrowed down to only 2. Namely, GPL96 and GPL570. This was done due to the fact, that these were the only 2 platforms with multiple different Primary sites. One very important point here, is that the distribution of primary sites are very skewed between the two platforms. Actually only two sites are shared between the platforms: breast & colorectum. This is elaborated further in the next section 1.3.

| Platform | Primary Site | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| ID | Breast | Colorectum | Kidney | Liver | Oral Cavity | Pancreas | Prostate | Skin | Thyroid | Total |
| GPL96 | 42 | 86 | 0 | 0 | 5 | 5 | 29 | 0 | 0 | 167 |
| GPL570 | 66 | 9 | 60 | 19 | 0 | 0 | 0 | 14 | 24 | 192 |
| GPL8432 | 104 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 104 |
| GPL10558 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 263 | 0 | 263 |
| GPL10379 | 207 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 207 |
| GPL15659 | 0 | 0 | 0 | 0 | 0 | 0 | 145 | 0 | 0 | 145 |

Table 3: Primary site for each platform

## 1.3 Data Preparation

Data preparation and wrangling proved to be the largest part and the most time-consuming part of this project. The data came in a very raw and non-tidy format and for that reason, this required much of the main attention in the project. Figure 2 shows a flow-chart of this process. The overall process consists of 4 individual steps: (1) Extracting data from HCMDB, (2) investigate this data and clean it, (3) using dimensionality reduction methods to reduce the dimensions of the datasets and (4) make predictions on the primary site and inference of the features influence on the response. Step 1-3 is now elaborated in details, along with the considerations, choices and the reasoning behind those made in each of the steps. Step 4 is further described in the results-section.

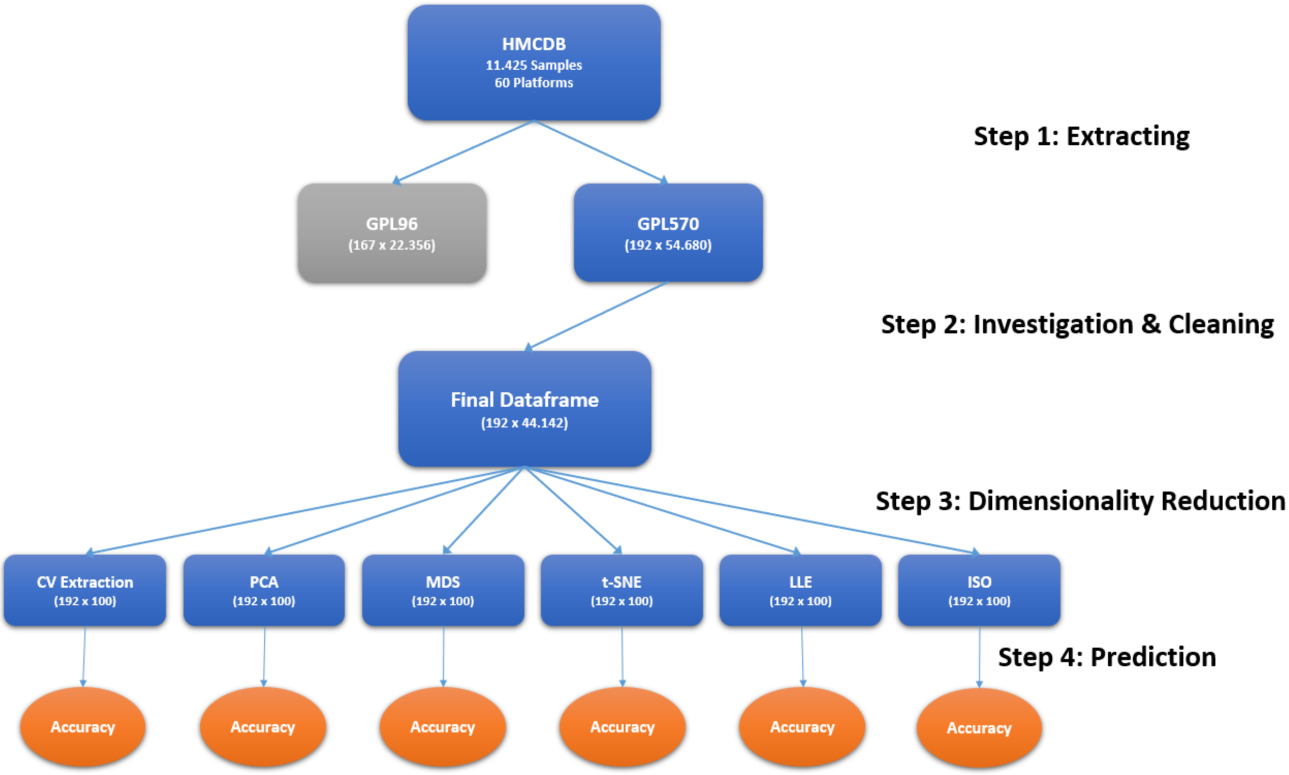


Figure 2: Flow-chart of the work-flow.

**Step 1**
Section 1.2 described the reasoning of choosing the two platforms GPL96 and GPL570 as candidate platforms (based on table 2 & 3). Therefore, a list of all the "Metastasis Tumor" samples from the two platforms were constructed, and the gene expression profiles of the samples were extracted/downloaded from their original source (NCBI Gene Expression Omnibus in this case). This was done using the GEOparse library, and the samples from the same platform were constructed into a tidy dataset, $n \times m$, where each row, $n$, represented a sample/observation and each column, $m$, represented a variable/gene/probe. The latter was done using the pandas library. Step 1 resulted in two dataframes named according to their platforms GPL96 ($167 \times 22356$) and GPL570 ($192 \times 54680$).

**Step 2**
Next, the datasets were further investigated and cleaned. One important property of the two datasets is that they are both extremely high dimensional ($p \gg n$), which is often the case for gene expression data. Since machine learning models often performs well for a large amount of training data, each sample was considered very precious. For that reason, the datasets were cleaned in the manner, that if any column contained one missing value, the column was simply removed - which also helped to reduce the dimensions of each of them. This was done for both of the datasets and the dimensions was reduced to $167 \times 18259$ and $192 \times 44142$ for the GPL96 and GPL570, respectively.

Since the number of samples were quite low, and the dimension of the datasets were so high it was of great interest to increase the number of samples in the final dataset. Therefore, all the original 6 platform candidates was actually downloaded and tidyed, to investigate if any of the platforms allowed for them to be merged. The conclusion was that the two most interesting platforms (2 final candidate-platforms) actually were merge-able, and the other 4 platforms were not. Therefore the merging of those two were performed and a new merged dataset was created with the dimensions $359 \times 16662$ - so the two platforms shared 16662 probes with no missing values. This merging should be done with carefulness, since it might give rise to bias in the prediction-outcome. Therefore, the differences and similarities was thoroughly investigated. The Coefficient of Variance (CoV) was determined for each of the shared features in both of the datasets. CoV is the standard deviation normalized by the mean and is given by the quantity $\frac{\sigma}{\bar{Y}}$. This provides a good measure for the variablity in the features. Figure 3 shows the distributions of CV in each of the platforms using two distinct histograms. It is obvious that the two distributions are very distinct, and the GPL570 shows a much higher variability in many of the features than GPL96. This raises the concern, that they somehow have distinct values and therefore merging the two platform datasets might not be a convenient approach.

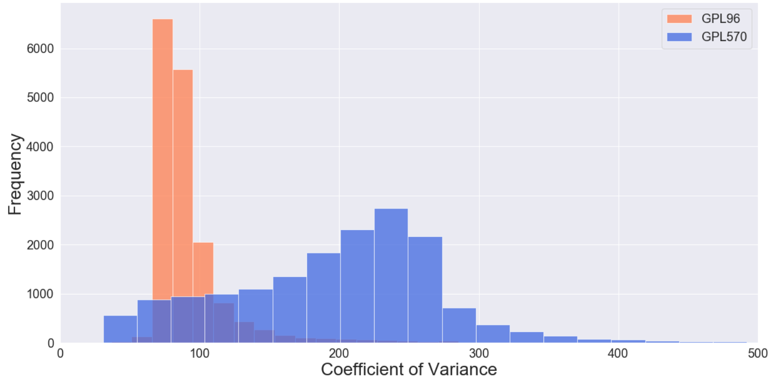
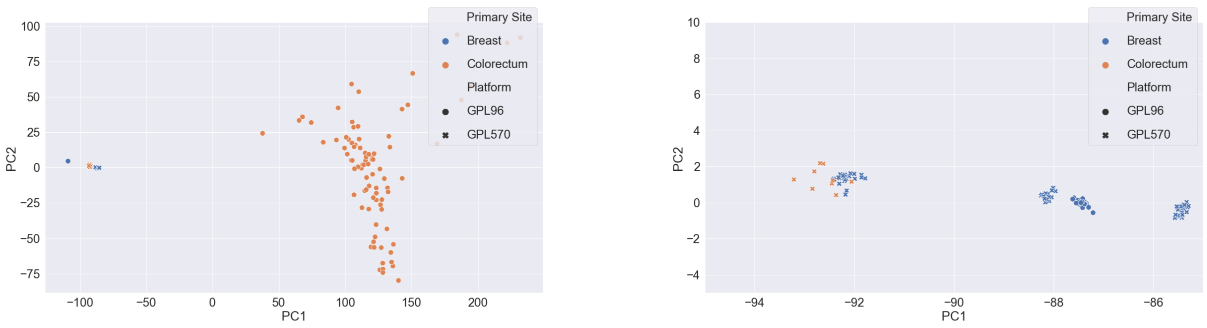


Figure 3: Distribution of the CoV in the two platforms.

From table 3 it was clear, that the two platforms contained very different distributions of primary sites. This could be one explanation for the very different distributions of CoVs in Figure 3. To further investigate this matter a PCA plot was made for the merged dataframe, where only the breast and colorectum samples were included (since these are the two overlapping primary sites). Figure 4 shows two PCA plots where figure 4a shows all the datapoints and 4b shows only a subset of these (a limit were set for the x-axis). The pattern is clear. The separation of the points are highly determined by the platform. From figure 4a it is clear, that the colorectum samples (orange colored) are separated due to the platform, and figure 4b shows breast samples are clumped due the platform of interest.



(a) Plot with all data points included.

(b) PCA zoomed into the region xlim(-95, -80).

Figure 4: Two PCA plots over the filtered merged dataframe.

It is behaving almost like the two platforms have different units of measure. One possible way to overcome this problem could be to normalize the features beforehand. Therefore a similar approach to the upper one was performed, but instead of merging and then normalizing the variables in the datasets were normalized individually before merging them. Meaning, each feature is set to have a mean of 0 and standard deviation of 1. Unfortunately, the resulting PCA plot shows the exact same pattern as Figure 4a and is overall very identical to that (the plot is given in Appendix §1).

The problem with this is that when predictions are made, one cannot conclude that the you are able to correctly classify or separate the kidney and prostate samples (that are classes unique to each of the platform) based on their genetic profiles. It could be because of the reason that they originated from each platform, and this issue would highly bias the results and conclusions drawn from them. For this reason platform GPL570 was chosen to proceed with and do the predictions on. The reasoning for choosing this is the higher number of samples ($n = 192$) and also the samples are more equally distributed between the classes compared to GPL96.

**Step 3**
The GPL570 dataframe is extremely high-dimensional. So despite the small sample size, it would still be computationally heavy to do the calculations when $p = 44.142$ on the cleaned dataset. Also, some machine learning approaches are known to heavily overfit, when the dimensions are too high (*curse of dimensionality*). For those reasons the dimensions of the dataset were reduced significantly with different methods. Another important aspect of this is, that it would be very interesting to investigate which dimensionality reducing approaches that would capture the variance and information best on lower dimensions. Therefore, in step 3 six different methods to reduce dimensions was performed in order to overcome the computational limitations and to compare each method. One feature selection approach and 5 feature extraction approaches were done. All methods are further elaborated in the Method-section.

All methods were performed so the new dimensions were equal to 192 × 100, which resulted in 6 equally sized dataframes. The two first features - referred to as principal components (PCs) - are shown in figure 5 for each method. The CV_EXT method is simply an extraction of the features with highest CoV-values. Since these genes are the ones showing the most variability, they are thought to be important in differentiating between the different primary sites. Looking at the figure in the top left corner, there seems to be some patterns in regard to separating the different primary sites - especially the kidney seems to some extend to be distinguishable between the two features with highest CoV-values. The same pattern is present in the two next methods PCA and MDS (top-center and top-right corner). Here the kidney-samples are even more separated from the other samples, and two skin samples are also very distinct. The other samples are quite clumped together. ISOMAP in the lower left corner distinguishes the kidney, skin and a group of breast samples from the rest. LLE and t-SNE creates more diverse plots, where the differentiation of the different primary tumors are even more present. In the LLE plot there has been added some jitter, beacuse the ”groups” present on the plot proved to have very small deviations from each other. LLE seems to differentiate the kidney, breast and skin quite well, and the rest is clumped together. The plot with most differentiation between the multiple classes was the t-SNE plot, where all of the different sites are put into different groups. Although, the method also to a grater extend than the other approaches separates the same sites into different subgroups.

Overall, the different techniques seems to quite clearly differentiate the kidney-samples from the other. In only 2 dimensions LLE and t-SNE seems to be separating the primary sites best, although this is no guarantee that these are the ones that are best to predict on, due to the simple fact that only 2 dimensions are present here, and all the datasets includes 100 dimensions. The 3 lower methods (ISO, LLE and TSNE) all has parameter-settings included in their method. ISOMAP and LLE requires to set the number of neighbors to consider, and t-SNE requires a learning-rate and a perplexity. Different values of these were investigated and the ones that seemed to separate the dataset best in 2 dimensions is the ones present in Figure 5 and also were chosen to be the ones to do the prediction on. The plots with different parameters is provided in the Appendix (§2, §3 and §4).

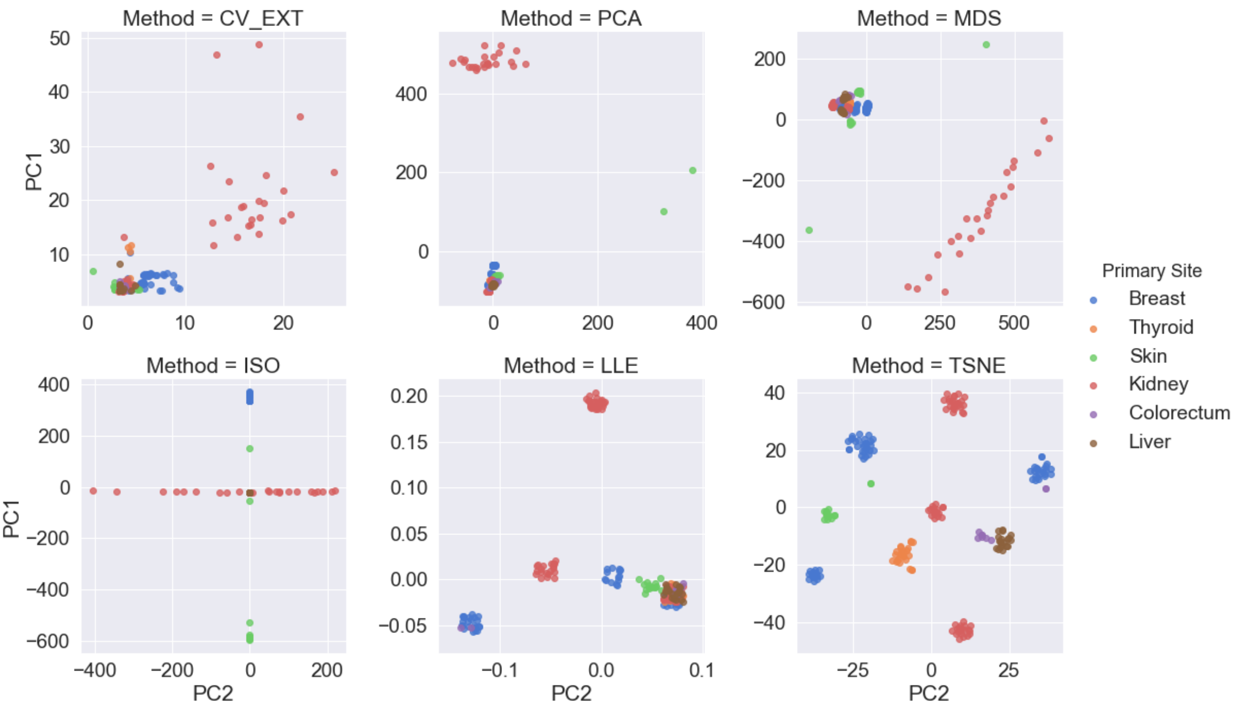


Figure 5: Two first PCs of each dimensionality reducing method.

The main reasoning for performing these methods are to capture as much valuable information/variance from the original 44.142 features on the new embedded 100 features. PCA provides a very easily interpretation of this, since the methodology behind it seeks to find as much variance as possible on the first principal component and then as much variance as possible on the next principal components, that are perpendicular to the previous one. This allows one to calculate the proportion of variance explained (PVE) by one of the dimensions (see methodology section 2.1.1). Figure 6 shows the cumulative variance explained by the first 50 principal components of the total dataset (included the 44.142 features). The figure shows that, that half of the PCs approximately accounts for 91% of the total variance, and the first PC accounts for around 60%. All of the 100 PCs accounts for 97.1% of the total variance. This explains the reasoning of why these techniques work, and it is quite impressive that 97.1% of the variance in 44.142 features can be explained by only 100 PCs.

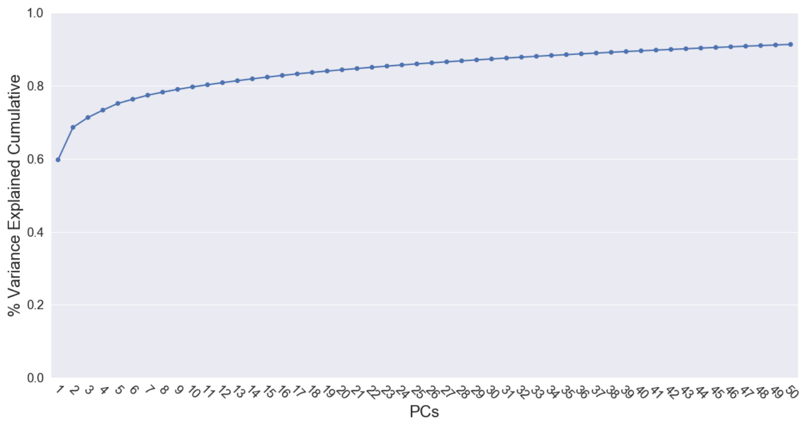


Figure 6: Cumulative proportion of variance explained by PCA.

# 2 Methodology

In this project overall two types of methods were used: (1) Dimensionality reduction methods and (2) Classification methods. Each of them representing a step in the 4-step approach used in this project, namely step 3 and 4 (Figure 2). Step 3 consist of purely unsupervised approaches, meaning it does not take the classes into account and only looks at the set of features $X_1, X_2, \dots X_p$ and step 4 is based on purely supervised approaches, meaning the classes are taken into account.

## 2.1 Dimensionality Reduction

Dimensionality reduction methods are the process of reducing the number of features/dimensions. The methods can be divided into two overall categories: (1) Feature selection and (2) feature extraction methods. Feature selection includes a variety of methods for selecting important variables, but the method used in this project, was simply to select the features with most extreme Coefficient of Variance estimates. Furthermore, 5 methods for feature extraction were used - Principal Component Analysis (PCA), Multidimensional Scaling (MDS) and t-Distributed Stochastic Neighbor Embedding (t-SNE), Isometric Feature Mapping (ISOMAP) and Locally Linear Embedding (LLE). The two latter approaches are extensions of MDS. Common for all of the methods are, that they all seek a low-dimensional embedding of the data, that still contains as much of the variance and information as possible.

### 2.1.1 Principal Component Analysis

In PCA, the principal components of the dataset are computed and analyzed. The components are based on the loadings, $\phi_{pk}$, and are used to determine the new datapoints, $z_{ij}$, which are referred to as the scores. The first principal component loading vector can be solved by the optimization problem [10]:

$$\max_{\phi_{11},\dots,\phi_{p1}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \left( \sum_{j=1}^{p} \phi_{j1} x_{ij} \right)^2 \right\} \quad \texttt{subject to} \quad \sum_{j=1}^{p} \phi_{j1}^2 = 1$$

The upper equation shows, that the first principal component is the normalized linear combination of the features that has the maximum variance. The reason behind the normalization is that otherwise the loadings, $\phi$, could be set to any arbitrary large value to increase the variance. The problem can be solved via an egiendecomposition, where the loading-vector, $\phi_1 = (\phi_{11},\ \phi_{21},\ \dots\ ,\ \phi_{p1})^T$ for a principal component is referred to as the egienvector, and the variance of a principal component is the egienvalue.

Next, the second principal component can be found. This is done by the same procedure as in the upper equation, although with this extra constraint, that the 2nd principal component must be orthogonal to the 1st principal component. This also have the advantage that is makes the 1st and 2nd principal component uncorrelated.

Since PCA is a matter of capturing as much variance in all of the features, it is important to scale the variables before doing the embedding. This is due to the fact, that the variables might be measured in different units and thereby this will highly affect the variance contributed by each. Furthermore, one important aspect of this technique is that we can measure the proportion of variance explained (PVE) by each principal component. This is an extremely useful tool, that can be used to determine to what extend the lower embedded dimensions captures the important aspects of the data but also how many principal components that should be used to capture enough of the information and variance. This is illustrated on the scree plot on figure 6, and the mathematical expression to calculate this is given by (assuming the data has been centered) [10]:

$$\frac{\texttt{Variance of m'th PC}}{\texttt{Total Variance}} = \frac{\sum_{i=1}^{n} (\sum_{j=1}^{p} \phi_{jm} x_{ij})^2}{\sum_{j=1}^{p} \sum_{i=1}^{n} x_{ij}^2}$$

### 2.1.2 Multidimensional Scaling

Multidimensional Scaling (MDS) is an alternative approach to map the different observations to a lower embedded space. MDS does so by taking a set of observations $x_1, x_2, \dots, x_N \in \mathbb{R}^p$ and transforms them into a new set of observations $z_1, z_2, \dots, z_N \in \mathbb{R}^k$ represented in a lower dimension $(p > k)$ [11]. The key idea is to create the lower dimensional space, such that the original distances, $d_{ij}$, between the observations are preserved. Therefore, it minimizes the following expression [11]:

$$\min_{z_n} \sum_{i \neq i'} (d_{ii'} - ||z_i - z_{i'}||)^2$$

In this project, the `sklearn.manifold.MDS` object was used to perform the MDS. It uses a type of gradient descent algorithm (SMACOF) to minimize the upper equation, which runs until a stopping criterion is met. The original pairwise distances, $d_{ij}$, is set to be Euclidean by default, which was used in this case.

Isometric feature mapping (ISOMAP) is an extension of MDS, that seeks to find a lower-dimensional representation of the observations that maintains the geodisic distances between the points (in contrast to classical MDS, that - by default - tries to maintain the Euclidean distances) [12]. The sklearn.manifold.ISOMAP object was used to do the calculations. This includes to either fix an integer, $K$, or $\epsilon > 0$ and let $N(i)$ be the the K-nearest neighbors of $x_i$ or the points within a ball-radius $\epsilon$ (centered at $x_i$). Then a K-Nearest-Neighbor graph must be computed and the new distance, $d_{ij}$, is the shortest fully-connected path between the two points. This means, that the new distance is an approximation of the geodisic distance, as it is the sum of all the edges on the KNN-graph, and those are measured in Euclidean distances. After obtaining these distances, regular MDS is applied to them [12]. The low dimensional embedding of ISOMAP was performed on different $K$-values. The one that looked as having the best representation in a lower dimension was chosen to perform classification on. The results of the $K$-values from $4 - 15$ is given in Appendix §2.

Locally Linear Embedding (LLE) is an approach, that tries to solve the same overall problem as ISOMAP but in a different way. The idea behind the approach is to identify weights, such that each data point is approximated by a linear combination of neighboring points. After these weights have been identified, the new embedded datapoints $(z_1, z_2, \dots, z_n)$, are created such that these are approximated by the same linear combination of neighbors. Again, a KNN-graph is constructed, and each point is then approximated by the points in the neighborhood [13]:

$$\min_{w_{ij}} ||x_i - \sum_{j \in N(i)} w_{ij} x_j||^2 \quad \texttt{subject to} \quad \sum_{j=1}^{N} w_{ij} = 1 \quad \texttt{and} \quad w_{ij} = 0 \texttt{ if } j \notin N$$

When the weights are found through the upper equation, they can be used to find the new embedded coordinates through:

$$\sum_{i=1}^{N} ||y_i - \sum_{k=1}^{N} w_{ij} y_j||^2$$

In this project multiple values of K Nearest Neighbors was again tried, and they can be seen in Appendix §3. The package sklearn.manifold.LocallyLinearEmbedding was used to do the calculations.

### 2.1.3 t-SNE

t-Distributed Stochastic Neighbor Embedding is a dimensionality reduction technique that requires a couple of calculations. One of the main features of the method is, that it converts distances to probabilities. This work by defining the probability that the datapoint $i$ chooses $j$ as its neighbor for any point [14]:

$$p_{j|i} = \frac{exp(-||x_i - x_j||^2/2\sigma_i^2)}{\sum_{k \neq i} exp(-||x_i - x_k||^2/2\sigma_i^2)} \quad \texttt{,where:} \quad p_{i|i} = 0$$

The upper equation is not symmetric. Practically, this means that the probability that $x_i$ chooses $x_j$ as its neighbor is not necessarily the same as $x_j$ choosing $x_i$, since $\sigma$ is dependent on the specific datapoint. Therefore, the symmetric value is defined as:

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2N}$$

The upper equation defines the probability in the high-dimensional space. Now the exact same is done for the low-dimensional space. Although, here one important change is done: Instead of assuming a Gaussian distribution the t-distribution is used with one degree of freedom.

$$q_{ij} = \frac{(1 + ||y_i - y_j||^2)^{-1})}{\sum_{k \neq l}(1 + ||y_k - y_t||^2)^{-1}}$$

This is done due to the *crowding problem*, which to some extend is overcome by this, since the t-distribution is more heavily tailed [14]. Therefore, a small $p_{ij}$ for the points $x_i$ and $x_j$, the lower embedded projections of $y_i$ and $y_j$ must be even further away from each other. This makes t-SNE a better tool for separating the clusters of points even more distinct than other methods. The measure of distance between the densities $p_{ij}$ and $q_{ij}$ is the Kullbeck-Leibler divergence, and this quantity is minimized to find the new datapoints:

$$KL(p||q) = \sum_{i,j} p_{ij} \texttt{log} \frac{p_{ij}}{q_{ij}}$$

The upper equation can be minimized iteratively, and by looking at the function, it is clear that if points with high probability of choosing each other in high-dimensionsal space $p_{ij}$, gets a low probability of choosing each other in the lower embedded space $q_{ij}$, the cost rises. Therefore, t-SNE is encouraged to preserve local structure [14]. The calculations were done using the sklearn.manifold.TSNE object, which has two tuning-parameters. Perplexity is correspondent to the number of nearest neighbors in other algorithms and controls the $\sigma$ in the high-dimensional formula for calculating $p_{ij}$. The learning-rate is another which is used when solving the minimization problem. Different settings for the two parameters can be seen in Appendix §4.

## 2.2 Classification Methods

After creating multiple different embedded datasets 4 different supervised classification methods were used to create a model that could classify the labels. Each of those methods has its advantages and disadvantages. Common for all of them is that they are trying to identify patterns in the set of features $X_1, X_2, \ldots, X_p$, that are crucial for determine the class, $y$. They do so in a parametric or non-parametric manner.

### 2.2.1 Logistic Regression

Logistic regression is a parametric linear classification method that in its basic form uses the logistic function to model a binary outcome. Given a set of features, the formulation of multiple logistic regression is given below, where $p(X)$ is the probability of the observation to be equal to the class set to 1 given the set of features [10]:

$$p(X) = \frac{e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}{1+e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}$$

The $\beta$-coefficients are estimated maximizing the likelihood function, $\ell(\beta_0, \beta_1, \ldots, \beta_p)$, given by:

$$\ell(\beta_0, \beta_1, \ldots, \beta_p) = \prod_{i:y_i=1} p(x_i) \prod_{i':y_{i'}=0} (1 - p(x_{i'}))$$

It is obvious from the upper formula, that if $y_i = 1$ the $\beta$-coefficients must be estimated in a manner such that the probability yield a result that is as high as possible and the opposite if $y_i = 0$. The idea of regularization can also be applied to logistic regression. This is done by subtracting a quantity from the likelihood-function in such a manner, the $\beta$-coefficients are shrunken towards zero when the regularization is stronger. The formula for this is given below, where $q$ determines whether to apply lasso- or ridge-regularization [15]:

$$\max_{\beta_0,\beta_1,\ldots,\beta_p} \left\{ \ell(\beta_0, \beta_1, \ldots, \beta_p) - \lambda \sum_{j=1}^{p} |\beta_j|^q \right\}$$

The classification with this approach was done using the sklearn.linear_model.LogisticRegression object. In this project it was a multi-class problem (and not a binary, which the basic logistic regression handles). This can be taken into account in the model by using a multinomial distribution, although here the one-vs-rest approach was taken. This means, that the approach is fitting $K$ models, each where a specific class is fit against the rest of the classes. The predicted outcome is the one of the $K$ models, that yields the highest probability.

### 2.2.2 Random Forest

Random forest is a non-parametric decision-tree based method, that creates an ensemble of fitted decision trees and hereafter combines all to predict an output. A decision tree is based on principle that it segments the predictor-space into non-overlapping rectangular regions, trough a top-down greedy approach called recursive binary splitting [10]. The cost function is often given by the Gini-index, which is a measure of node-purity, and are minimized in each split of the tree. The index is given by:

$$G = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk})$$

Here, $\hat{p}_{mk}$ is the proportion of training observations in the mth region from the kth class. The index has the property, whenever $\hat{p}_{mk}$ becomes close to 0 or 1 the index becomes small ($G \to 0$). In each split all features are normally considered, and the one causing the largest minimization of the Gini index is chosen as the split. This often results in trees suffering from high variance, and the trees heavily overfits the training data. Random forest seeks to avoid this problem by creating an ensemble of $B$ decision trees made from bootstrapped samples of the training data (bagging). This will lead to a decrease in variance and therefore often a gain in prediction accuracy. Furthermore, the method includes picking a subset of the predictors, $m$, to consider in each split. This leads to a decorrelation of the decision trees made (otherwise many of them would look very much alike), and decreases the variance even further [10]. The final prediction of random forest be be described mathematically by:

$$\hat{f}_{RF}(x) = \frac{1}{B} \sum_{b=1}^{B} f^{*\hat{b}}(x)$$

.

The sklearn.ensemble.RandomForestClassifier object was used to make the predictions, where $B$ was set to 5000 and the number of features to consider in each split, $m$, and the maximum depth of the tree, $k$ was the two parameters optimized using k-fold cross-validation.

### 2.2.3 Boosting

Boosting is a general machine learning technique, that builds an ensemble model typically of decision trees. It it distinct from random forest in the manner, that the trees are created sequentially. Where the bagging method is a random sample of the original dataset, boosting is not a random sample, instead each new tree is created on a modified version of the original dataset. The idea behind the sequentially format is that the trees are learning slowly, and each newly created tree is using information from the previously grown tree to improve its mistakes [10]. The mathematically expression is given by:

$$\hat{f}_{boost}(x) = \sum_{b=1}^{B} \lambda \hat{f}^{b}(x)$$

So the final model, $\hat{f}_{boost}(x)$, is the average between many constructed decision trees, $\hat{f}^{b}(x)$. The parameter $\lambda$ is the essential part of the boosting model and are called the shrinkage parameter (also, learning-rate). This is the extend that each newly constructed tree learns from the previous model: $\hat{f}(x) \leftarrow \hat{f}(x) + \lambda \hat{f}^{b}(x)$. In this project the xgboost.XGBClassifier object was used. The learning-rate, subsample (conceptually similar to the $m$ in random forest), max-depth of each constructed tree and the number of estimators, $B$ was optimized using k-fold cross-validation.

### 2.2.4 Support Vector Machines

Support Vector Machines (SVMs) is a non-parametric classifier, that is based on the idea, that the model is separating the training observations into different groups using hyperplanes. SVM is an extension of the support vector classifier (SVC). SVC corrects the problem with the maximal margin classifier, namely if the training observations cannot be perfectly separated it has no solution (non-separable case). SVC creates a hyperplane even though the data points are not perfectly separable. It can be described as follows [10]:

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}) \geqslant M(1 - \epsilon_i), \qquad \epsilon_i \geqslant 0, \qquad \sum_{i=1}^{n} \epsilon_i \leqslant C$$

The upper equation states, that each point must be equal or bigger to the quantity $M(1-\epsilon_i)$. M is the margin, and $\epsilon_i$ indicates where the training observation is located relative to the margin and hyperplane. If $\epsilon_i > 0$ then the $\epsilon_i th$ observation is on the "wrong" side of the margin and if $e_i > 1$ then the $e_i th$ observation is on the wrong side of the hyperplane. $C$ is a therefore a non-negative tuning parameter that has to be chosen. This parameter decides the amount for which the margin can be violated, for example if $C$ is high many training observations are allowed to violate the margin/hyperplane and if $C = 0$, then the support vector classifier equation corresponds to the maximal margin classifier, since no violation is allowed to occur. This means that when $C$ is high the margin is wide, and therefore would include many support vectors (data points that lies exactly on the margin-line or on the wrong side of it) that will affect the separating hyperplane. The goal is to calculate the maximum value for M [10].

SVM is an extension of the SVC. SVC produces linear boundaries, although it can be proven that the solution to the support vector classifier involves only the inner products of the observations. Therefore, the "kernel-trick" can be used to manipulate these kernels to make them non-linear. In this project the sklearn.svm.SVC object was used with 3 different kernels: (1) Linear, so basically the SVC, (2) radial and (3) sigmoid. These 3 was optimized along with a specified range for the C-value. One important note is, that SVMs, like logistic regression, were using the one-vs-rest principle. Meaning, that also in this case $K$ models was trained on the training data.

# 3 Results

In this section the classification results on all the six dataframes are presented. It is important to note, that the predictions were made with the 100 embedded features, $X_1, X_2, \ldots, X_{100}$ with the response variable as the class (in total 6 different classes). The dataset did include more meta-information like the site of the metastasis tumor, which could be argued to always be known. Although, that information was left out here, since the distribution of these sites turned out to be skewed, which could lead to a biased result (namely, the classification was purely based on the skewed distributions between primary and metastasis sites). An example of this is that all 60 kidney primary tumor samples has lung-tissue as primary site - which is also the most extreme case.

In order to perform the classification the first thing was to divide the datasets into two distinct sets of samples; training and validation set. Because of the very small sample size, the split between these were chosen to be 80/20, so the training was done on as many samples as possible while the validation still has a size, that is large enough to make trustworthy results. Although, the small size of the validation has its implications, that are commented on in section 3.1.2, where the results of validation are presented. The split was performed in a stratified manner, due to the skewed distribution of primary sites in the dataset. So 20% of the samples in each class were randomly assigned to the validation set. Table 4 shows the number of samples in the training and validation set.

| | Training | Validation |
|---|---|---|
| Breast | 53 | 13 |
| Kidney | 48 | 12 |
| Thyroid | 19 | 5 |
| Liver | 15 | 4 |
| Skin | 11 | 3 |
| Colorectum | 7 | 2 |

Table 4: Number of classes in the training and validation set.

## 3.1 Training Accuracies (K-fold cross-validation)

Each of the 4 classification methods (logistic regression, random forest, xgboost and SVM) was optimized using a stratified k-fold cross-validation. The number of folds was set to $K = 5$, since the number of folds could not exceed the number of samples in the smallest group (colorectum with 7 samples), due to the settings of the sklearn library. The training accuracy for each method with a specific set of parameters was obtained by taking the mean of the accuracies on the folds. Figure 7 shows a boxplot of the predictions made on each dataset for the different methods. It is clear, that on an overall level random forest and XGboost often outperforms logistic regression and support vector machines. Although, with the correct parameters, these methods are able to compete with the tree-based methods. t-SNE and Isomap are the two dataframes that struggles the most with high training-accuracies, and the highest score of the other 4 dataframes are very similar. The highest CV-score was obtained on the PCA dataset by XGboost which was estimated to be 94.1%.

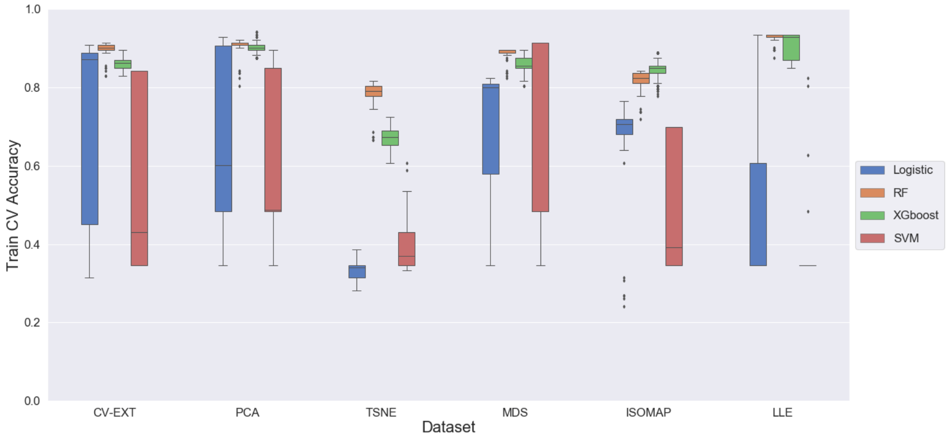


Figure 7: Boxplot plot on the predictions made on each dataset for the different methods.

### 3.2 Validation Accuracies

The validation accuracies were obtained by taking each method with the specific parameter-settings that had the highest estimated train accuracy from each dataset, and using those to predict on the validation set. In figure 8 each point represents a prediction of each method (colored) with highest estimated CV-score on the specific dataset (point-style). The plot shows that most of the points are in the upper triangle, which is an indication of the methods are better than estimated based on their CV-estimate (not overfitted).

Figure 8 shows a great deal of interesting things, although three main-objectives are to be concluded from the results: (1) It is clear that especially the PCA and LLE dataset are well-suited for capturing the amount of variance on only 100 dimensions. Here 5 predictions are the are the ones closest to the upper right corner. Both logistic regression, XGboost and random forest are present in that group. (2) One prediction actually got a 100% validation accuracy (SVM on MDS). This is a great result, but it must be considered that it comes with a great deal of uncertainty. Since the sample-size is very limited, the amount of stocasticity involved in choosing the training and validation set is quite high. Meaning, that all of these methods might perform significantly better or badder at a different distribution of samples within the training and validation-set. This uncertainty could be overcome by running the results multiple times on different sample-assignments to the training- and validation-set, and would yield a more clear picture of which results that performs the best. Although, this was not done here. (3) The CV-EXT dataset with a subset of the original features performs OK. The accuracy is around 90% for the training and validation set. Although, it seems like it is overfitting a bit with 3 clear and distinct points below the line. This is very useful for inference purposes and for that reason the inference of the genes' influence on the primary site is investigated in the next section.

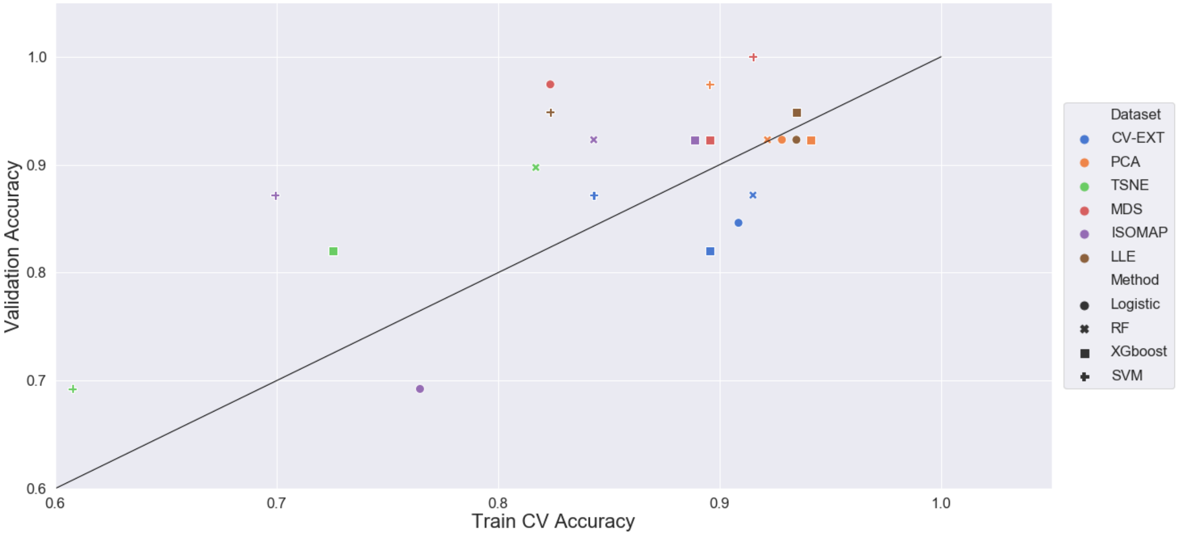


Figure 8: The validation accuracy vs. training accuracy for each method and dataset.

Finally, it should be mentioned that the exact same approach for obtaining the results was first made on the merged dataset (between GPL96 and GPL570). What is surprising about this is, that a great deal of the predictions yielded higher accuracies even though the classes were extended to 9. Although, as section 1.3 step 2 described, the approach comes with a lot of bias, and therefore they are not considered as main-findings in this project. The training accuracy plot (box-plot) and validation accuracy plot is provided in Appendix §5.

### 3.3 Inference of Important Genes

Section 3.1 and 3.2 was purely focused on prediction and the accuracy of that. This was the main-purpose of this project, but also another important aspect is the reasoning behind being able to distinguish between different primary sites. This of course boils down to the fact, that it is thought that different expressions in specific genes can distinguish between multiple possible sites of the origin. Therefore, inference of what genes was important due to separate the different primary tumor sites are investigated in this section. Many methods are suited for inference, although random forest, XGboost and SVM are not the best ones. Logistic regression could be useful, but here a single decision tree was made based on the traning samples, since they are extremely easy to interpret and are very suitable for this case (since the splits are based on differences in expression).

The tree was made with a maximum depth of 4, and it is presented in Figure 9. The train-accuracy was 79% and the validation accuracy was 66%. So really not the best, although the tree still provided some very interesting insights (marked from 1 - 3 in red circles). The first split (1) separates more than half of the primary breast-samples from the rest. This was due to a high expression of the 206378_at-probe. This probe detects the SCGB2A2-gene (which encodes for the Secretoglobin, Family 2A, Member 2 protein), and are in fact used to identify and detect disseminated breast cancer cells [16], which is exactly the case of this study. (2) Separates skin from breast and liver samples, which is due to a high expression of a Tyrosinase-encoding gene, that helps control the expression of melanin. A protein known to be mostly expressed in skin-tisue. (3) is the expression of SCGB2A1-gene (same family as the protein from (1)). This has also shown to be related to identification of cancerous cellls [17] These findings helps to confirm, that some of the patterns present in the dataset were truly related to the expression-profiles of different primary sites.

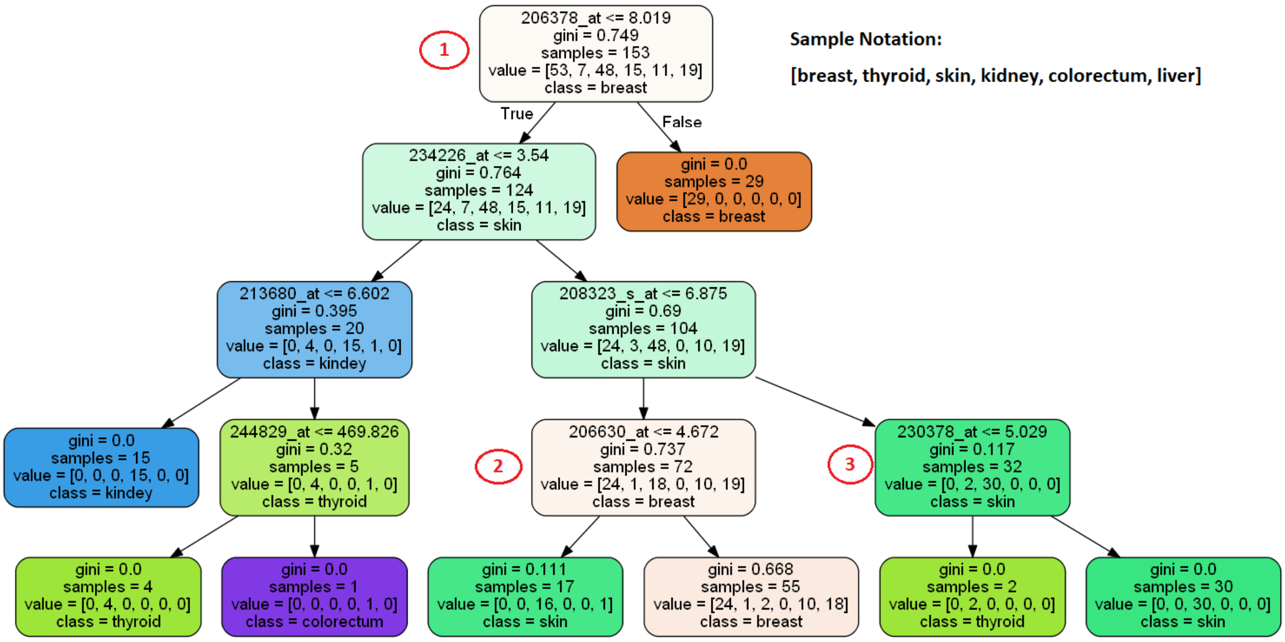


Figure 9: Simple decision tree for the CV-EXT dataset.

# 4 Conclusion

The Human Cancer Metastasis Database was the source of data in this project. Even though the database included 11.425 samples, it turned out to be quite messy and difficult to obtain a large useful dataset from it due to complication of merging samples from different platforms. It resulted in 192 samples derived from the GPL570 platform that included 6 different types of primary sites (breast, colorectum, kidney, liver, skin and thyroid) and 42.142 features. Despite the small sample size, the very high dimensions and the skewed distributions of classes, it was workable using different approaches to overcome and handle this dataset.

Six different dimensionality reduction approaches were performed in order to overcome the high-dimensional problem in relation the dataset ($p \gg n$) and hereby trying the capture the variance from the 42.142 dimensions into 100 embedded dimensions. This resulted in 6 new datasets all $192 \times 100$. Figure 5 summarized them all in 2 dimensions. Different patterns were present on all of them, where LLE and t-SNE was superior compared to the other techniques in order to distinguish between the samples visually in 2 dimensions. Figure 6 exemplify the idea behind using these techniques. In the PCA dataset 97.1% of the total variance in the original dataset, were captured in only 100 features.

Four different classification methods were performed on the 6 different embedded datasets. Figure 7 shows the performance of the different methods on the training set. The two tree-based methods (random forest and xgboost) were identified to be the - on average - best approaches. Furthermore, the t-SNE and ISOMAP datasets shows the worst ability of yielding high training accuracies. The best CV-accuracy turned out to be XGboost on the PCA dataset (94.1%). Figure 8 summarized all the validation accuracies on the different datasets using the optimized methods. LLE and PCA turned out to perform best regarding the datasets, and the methods XGboost, logistic regression and random forest all performed very similarly with train- and validation-accuracies around 93%. The feature selected dataset CV-EXT yielded decent results, meaning the genes with highest variability contains useful information in regards to distinguish between the primary sites.

Since CV-EXT yielded decent results the dataset were used to make inferences about the important genes in order to distinguish between the primary sites. This was done using a decision tree and it provided very interesting and useful insights. Two genes, SCGB2A1 and SCGB2A2, that previously have been mentioned in articles were identified in the tree separating breast-cancer and skin-cancer primary sites from others. Those finding helped to confirm the overall goal, that there truly were important genes that helped identify the primary tumor origin site.

Overall the project is concluded to have worked out successfully. Previous studies have shown to correctly identify primary sites with a degree of accuracy of $82\% - 97\%$. The results provided here are within the upper part of this range. Although, one thing that would help the trustworthy of the results presented here would be to run the analysis multiple times and yield a more exact validation-accuracy.

# References

[1] National Cancer Institute: Cancer. https://www.cancer.gov/publications/dictionaries/cancer-terms/def/cancer. Date: 7 December 2019.

[2] National Cancer Institute: Metastasis. https://www.cancer.gov/publications/dictionaries/cancer-terms/def/metastasis. Date: 7 December 2019.

[3] Losa, F et. al. (2017), *"SEOM Clinical Guideline on Unknown Primary Cancer"*, Clinical Guides in Oncology.

[4] Economopoulou P & Mountzios G et. al. (2015), *"Cancer of Unknown Primary origin in the genomic era: elucidating the dark box of cancer"*, Cancer Treat Rev.

[5] Gauri R. Varadhachary & Martin N. Raber. (2014), *"Cancer of Unknown Primary Site"*, New England Journal of Medicine.

[6] Bugat, R & Bataillar, D et. al. (2003), *"Summary of the standards, options and recommendations for the management of patients with carcinoma of unknown primary site."*, Br J Cancer.

[7] Hainsworth JD, Greco FA. (2014), *"Gene Expression Profiling in patients with carcinoma of unknown primary site: from translational research standard of care."*, Virchows.

[8] Haghighi, Elham & Knudsen, Michael et. al. (2019), *"Hierarchical Classification of Cancers of Unknown Primary Using Multi-Omics Data"*, Cancer Informatics.

[9] Guantao, Zheng et. al. (2018), *"HCMDB: The Human Cancer Metastatis Database"*, Nucleic Acids Research.

[10] James G. & Witten D. & Hastie, T. & Tibshirani, R (2017), *"Introduction to Statistical Learning with Applications in R"*, Springer, New York.

[11] J. B. Kruskal. (1964), *"Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis"*, Psychometrika.

[12] Tenenbaum, J.B. et. al. (2000), *"A global geometric framework for nonlinear dimensionality reduction"*, Science.

[13] Roweis, S. & Saul, L. (2000), *"Nonlinear dimensionality reduction by locally linear embedding"*, Science.

[14] Van der Maaten, Laurens et. al. (2008), *"Visualizing Data using t-SNE"*, Journal of Machine Learning Research.

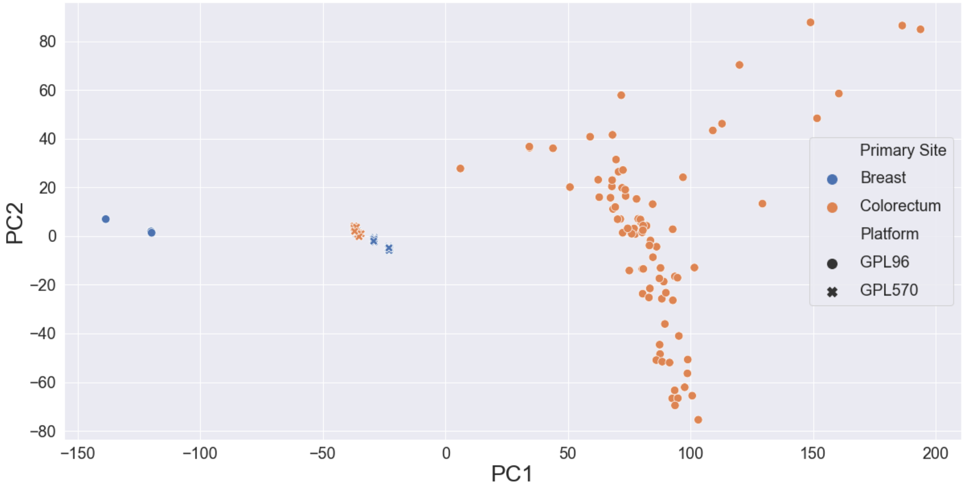
[15] Hastie, T. & Tibshirani, R. & Friedman, J (2017), *"The Elements of Statistical Learning: Data Mining, Inference, and Prediction"*, Springer, New York.

[16] Lacroix, Marc (2006), *"Significance, detection and markers of disseminated breast cancer cells"*, Endocrine-Related Cancer.

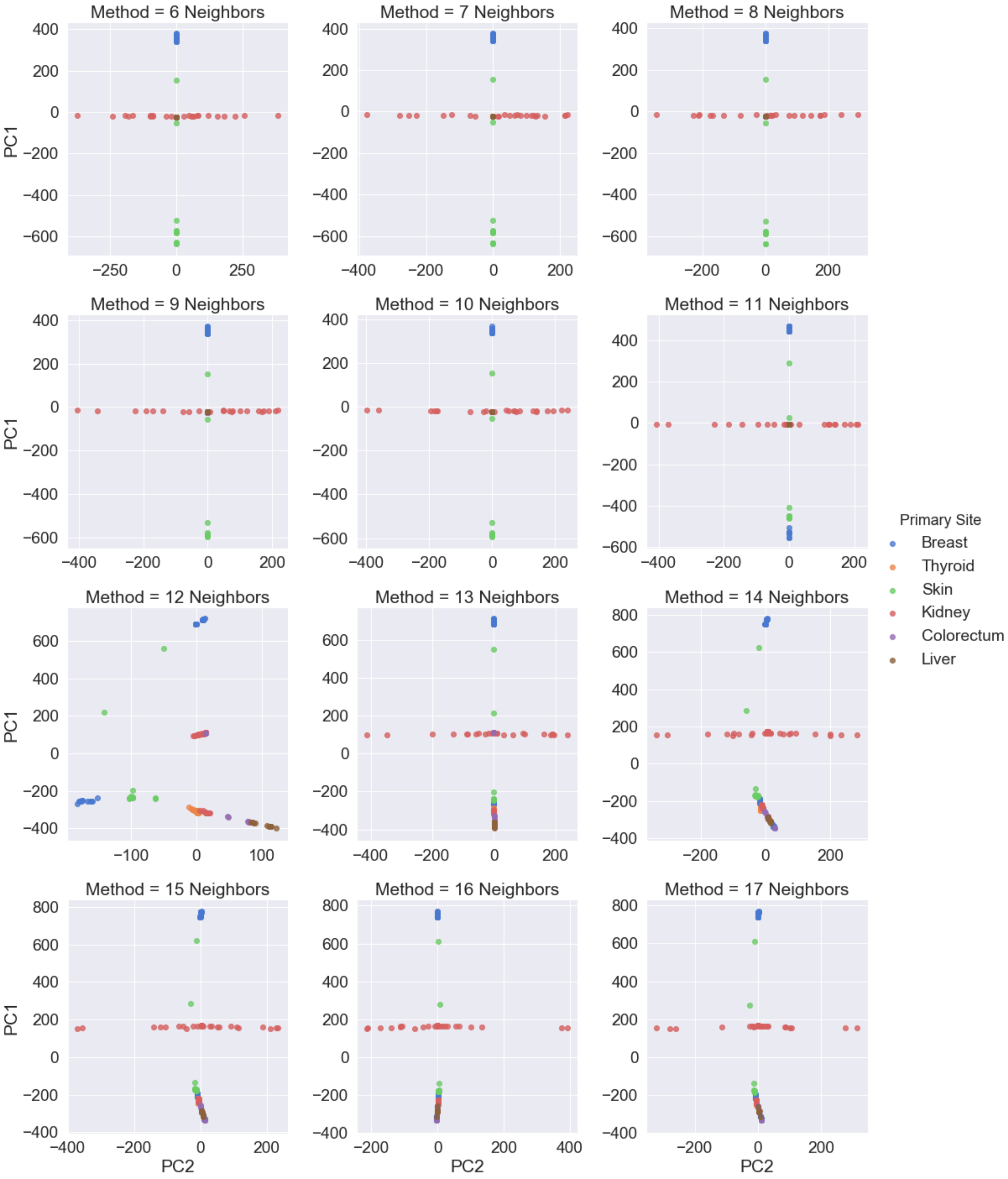
[17] E. Krop, Ian et. al. (2001), *"HIN-1, a putative cytokine highly expressed in normal but not cancerous mammary epithelial cells"*, Proceedings of the National Academy of Sciences (PNAS).

# 5 Appendix

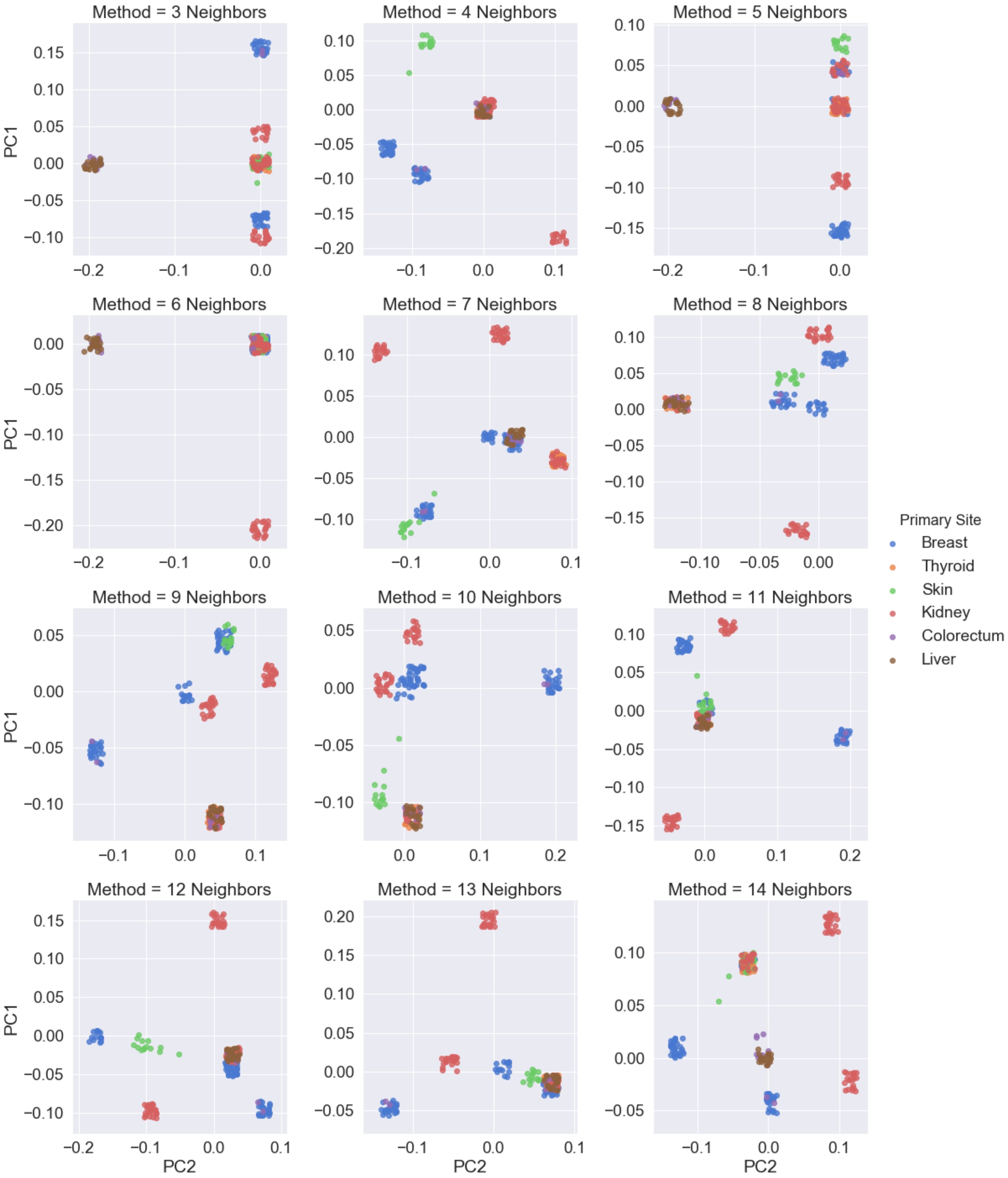
### §1 - PCA on beforehand normalized dataframes



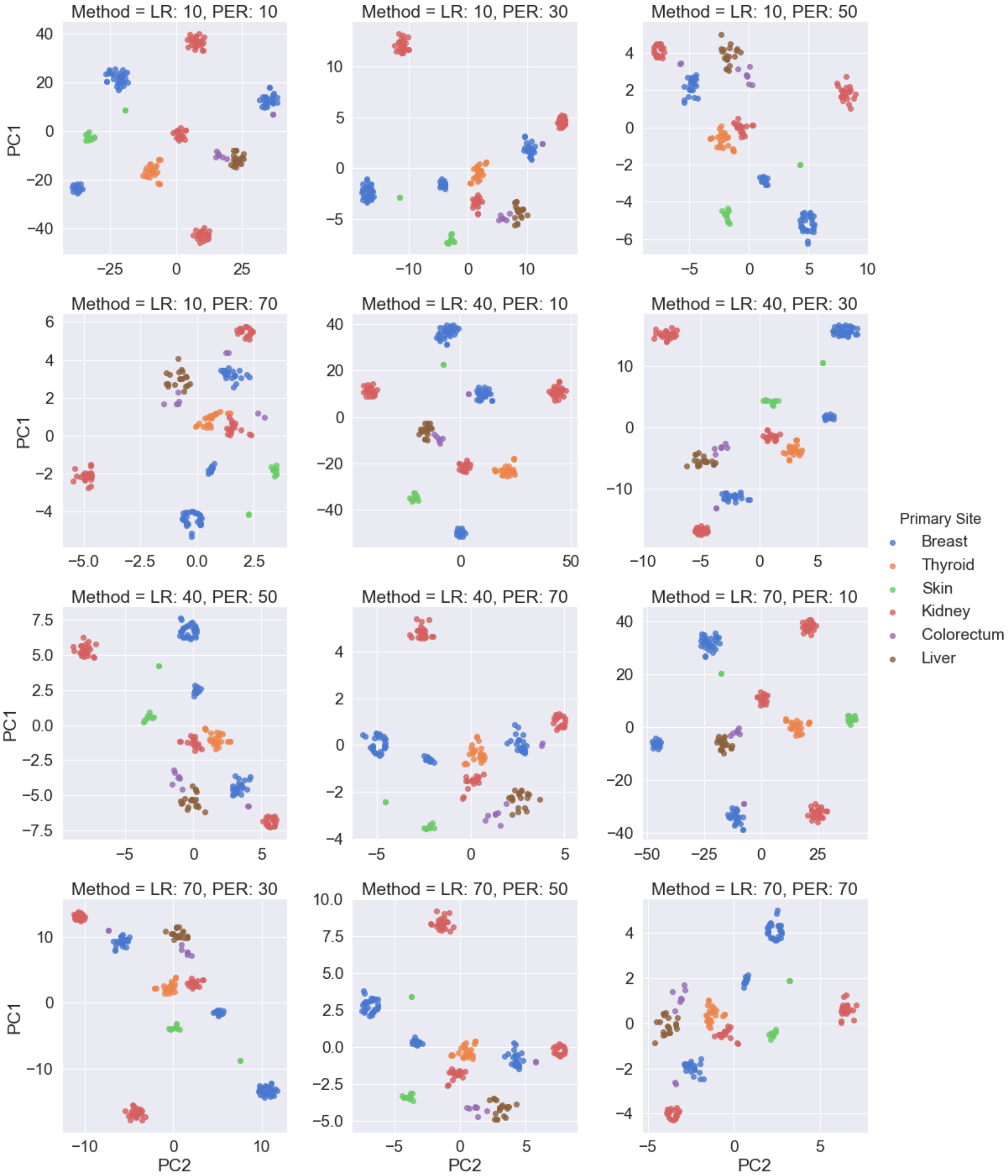
## §2 - ISOMAP (different neighbor values)



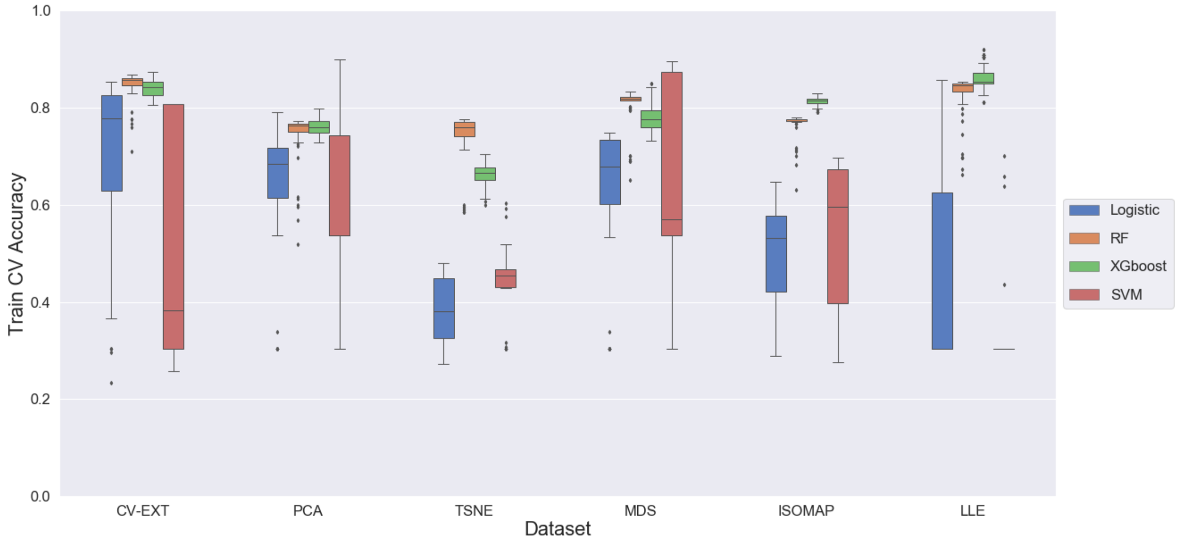
**§3 - Locally Linear Embedding (different neighbor values). OBS: Jitter is present in all plots.**

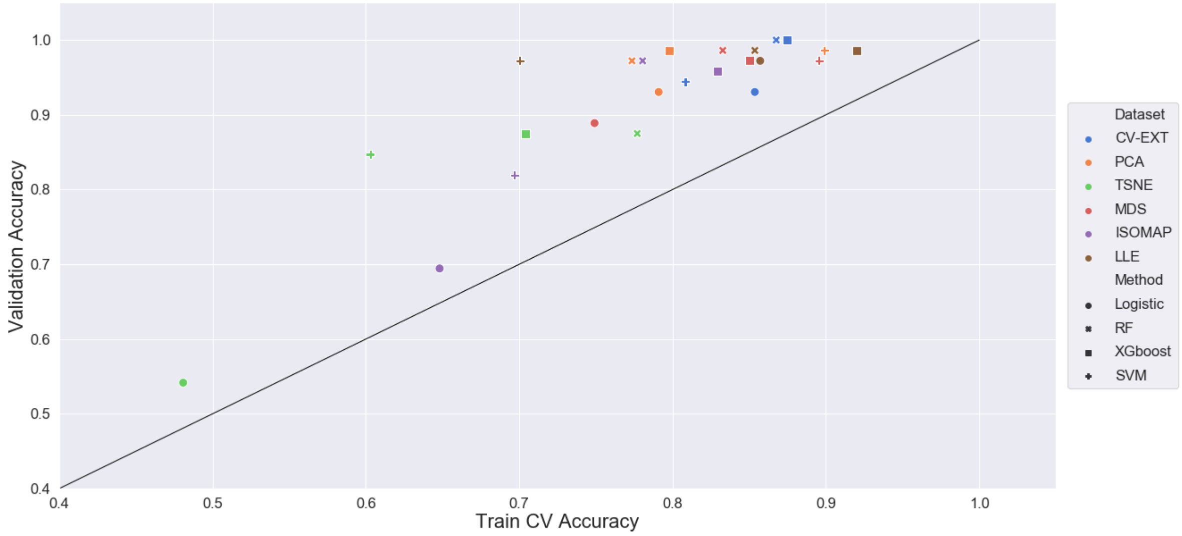


## §4 - t-SNE (Different learning-rate and Perplexity values)



## §5 - Results for the merged dataframe (Train CV Accuracy and Validation Accuracy)


Train CV Accuracy
CV-EXT
PCA
TSNE
MDS
ISOMAP
LLE
Dataset
Logistic
RF
XGboost
SVM


Validation Accuracy
Train CV Accuracy
Dataset
CV-EXT
PCA
TSNE
MDS
ISOMAP
LLE
Method
Logistic
RF
XGboost
SVM