July 3, 2018

BACHELOR THESIS – APPLIED MATHEMATICS

INCREASING CANCER CELL RECOGNITION WITH RAMAN MICROSCOPIC DATA USING SPARSE CODING

Pascal Loohuis

Faculty of Electrical Engineering, Mathematics and Computer Science (EEMCS)
Applied Analysis

Exam committee:
L.L. Zeune MSc
A.E. Martinez MSc
Dr.Ir. G.F. Post

UNIVERSITY OF TWENTE.

Increasing cancer cell recognition with Raman microscopic data using sparse coding

P. Loohuis*

3rd July 2018

Abstract

Traditional methods of research on cancer cells are done via tissue biopsy. Due to the fact that these biopsies are poorly able to predict the treatment response, other research methods are investigated to eventually replace tissue biopsies. One method is performing research on circulating tumor cells from the blood stream, whereas Raman microscopic techniques are used to distinguish different sorts of cancer. This data is used to obtain a fingerprint per sort cancer by classifying the data. Principle component analysis (PCA) is used in order to make this hyperspectral data insightful. Data often contains nonlinear statistical dependencies, so it is questionable if PCA is the right method to use. This report introduces two other methods, based on sparse coding, that tackles this shortcoming of PCA. In sparse coding a signal is decomposed in a multiplication between a set of basis vectors and a sparse matrix, whereas each pixel of the hyperspectral data will be described with only a few of these basis vectors. The introduced methods proved to give good classifications and were noise resilient.

Keywords — sparse coding, PCA, cancer cells, hyperspectral imaging, Raman ISTA, CoD

*Student Applied Mathematics, S1725866, University of Twente, Enschede, the Netherlands. This report is supervised by L.L. Zeune MSc, PhD student within the departments Medical Cell BioPhysics (MCBP) and Applied Mathematics (AM) at the University of Twente and A. Enciso Martinez, PhD student within the department Medical Cell BioPhysics (MCBP) at the University of Twente.

1 Introduction

Almost 1 out of 6 deaths are caused by cancer [1]. Research on the composition of cancer cells is therefore an important topic within the health-care community these days. Current cancer diagnostics is primarily done via tissue biopsy. A single biopsy has shown its limitations due to the heterogeneity of the tumor. Although multiple biopsies sounds like a clear continuation, the implementation is rather impractical because of its invasive nature and risks[15]. Acquisition of cancer tissue is necessary for cancer research so researchers have been searching for other ways to gain cancer cells. *Circulating tumor cells* (CTCs) in the blood vessels have been chosen as an alternative because the isolation of these cells is safe and less expensive[25]. Identification of the composition of these cells can be done via *Raman microscopy*. Raman microspectroscopy is an imaging technique that uses hyperspectral cameras to measure the electromagnetic energy scattered from a sample using laser excitation. These energy characteristics are measured in thousands of spectral bands and are then used to obtain a fingerprint from cells based on their scattered light[11, 17]. The fingerprint will serve as an input for unsupervised statistical classification methods like *hierarchical cluster analysis* (HCA) and *principal component analysis* (PCA). Use of the latter clusters the different cell components such that distinction between them can be visualized and different cell types can be classified.

Processing the hyperspectral data, compared to classical fluorescent images, leads to a great increase in the processing complexity and time. Therefore, effectively reducing the amount of data is an essential task for hyperspectral data analysis. One common approach for this dimensionality reduction is PCA. It is a type of dimensionality reduction where high dimensional data will be expressed in a lower dimensional dataset of active components. PCA tries to select a few mutually independent principle components which describes the data set best and uses all components to represent each pixel of the observed data [11, 13, 26].

Although PCA is frequently used, there are some drawbacks to this method. Due to the fact that the principal components have to be orthogonal, the method is less flexible[24]. Furthermore, PCA is a good method for data where linear pairwise correlations are predominate, but data often contains important higher-order statistical dependencies. If so, then it is questionable if PCA is the right way to go[18].

The goal of this paper is to introduce robust and noise resilient methods that can serve as a dimensionality reduction tool without the mentioned drawbacks. The methods that are used are based on sparse coding. The PCA method will be replaced by a *Iterative Shrinkage Thresholding Algorithm* (ISTA) and a *Coordinate Descent method* (CoD). These methods will be compared with PCA based on their classification of cancer cells and based on clustering validation indices. A requirement for using these sparse coding methods is a

learned dictionary. A dictionary learning algorithm from [24] will be used in this paper because of its fast convergence.

This report is organized as follows: in section 2 a brief overview of the biomedical process is given. Section 3 discusses the current method of classifying cancer cells using PCA. In section 4 a method based on sparse coding is presented that will replace the discussed method from section 3. This section will also contain a brief overview of the used dictionary learning algorithm. The results of the method presented will be given in section 5 and section 6 will contain the conclusion and recommendations of this report.

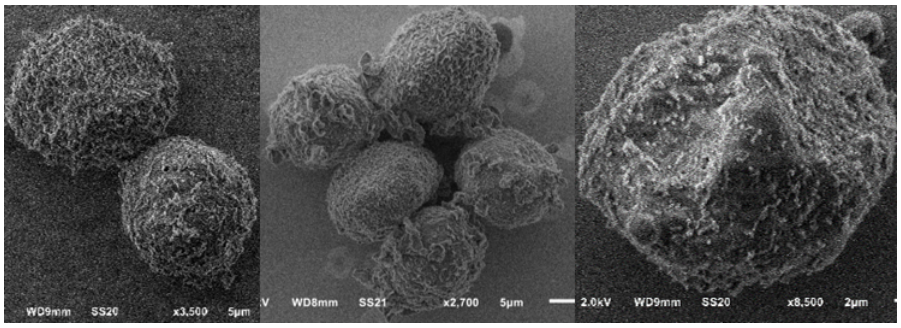
2 Background information

Cancer tissue has been important for cancer diagnosis, the cancer's fingerprint and the prediction of matching cancer therapy. Until now, there has been no evidence that increasing the understanding of the tumor via tissue biopsies has led to an increase in treatment response or even survival. The heterogeneity of cancer cells and its dynamic characteristics over time are responsible for this poor prognosis. Research has shown that one biopsy is inadequate to map the full diversity of the cancer and even multiple biopsies are not appropriate for this task, because multiple biopsies do not provide enough information and are impractical due to clinical risks for the patient[16].

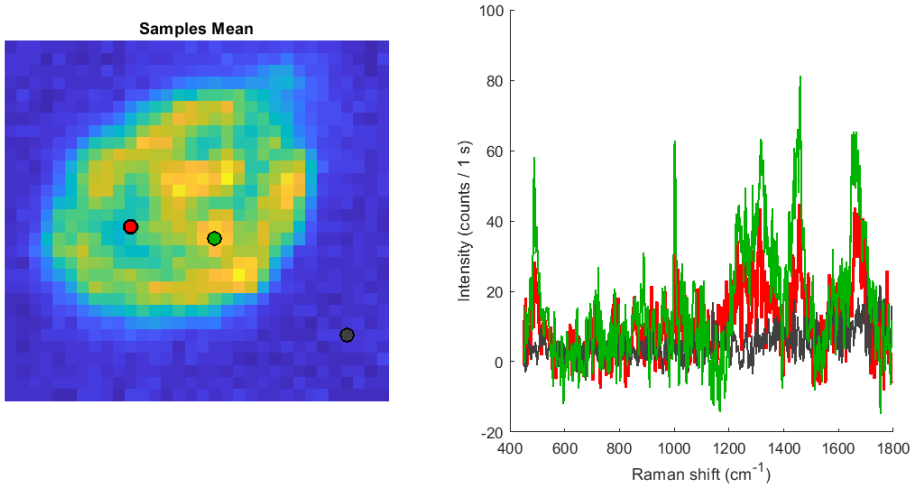
Because a good identification of the tumor is essential for an optimal treatment [9, 21], researchers have been searching for other methods to gain cancer cells for research. CTCs have become an attractive alternative for obtaining tumor tissue because the described disadvantages do not occur. So-called "liquid biopsy" can be used for the accession of cell-free DNA from cancer cells and the CTCs. These cells are released in the blood vessels during the spread of the cancer. This method of obtaining cancer cells has the advantage that taking multiple samples do not harm the patient and the segregating of pure cancer tissue and other material is not expensive and difficult[21].

Mapping the characteristics of these cells can be done in different ways. Popular methods make use of microscopic equipment, e.g. CellSearch®. *Scanning Electron Micropscopy* (SEM) and Raman micro-spectroscopy are microscopic techniques for revealing the characteristics of cancer tissue. SEM uses beams of electrons to gain information about the sample's surface morphology. The fact that it makes use of electron beams ensures high resolution data from the sample's surface. Raman microscopy uses laser excitation on the sample and collects the scattered light from the tissue. This scattered light can be used for the classification of cell composition but is less accurate than SEM because laser beams are substantially bigger than electrons. In figure 1a and 1b below the results of cell measurements from both methods are shown. For the rest of the report the focus will lie on Raman data.

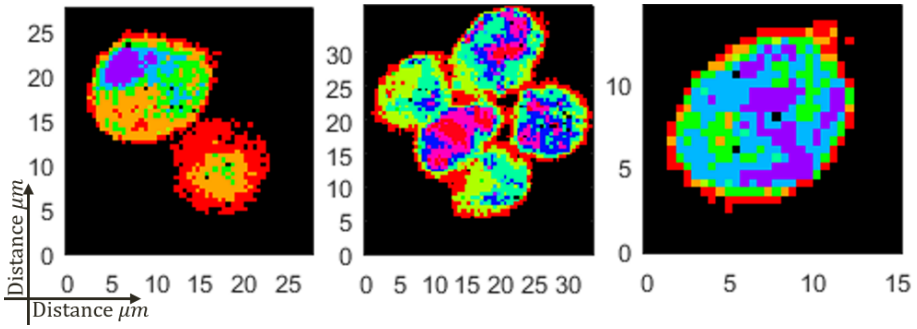
Performing analysis on Raman data requires the data to be more insightful. Before clusters based on their chemical components can be formed, certain preprocessing steps have to be made. The first task is to refine the data by, among other things, removing cosmic rays and outliers. This refined data is still inappropriate for cluster algorithms to be efficient and precise. This has a lot to do with the large amounts of data coming from these hyperspectral cameras. Reducing the amount of data while preserving the utmost of the variation within the data, is the crucial next step. Methods like PCA and sparse coding will help out with that problem. Afterwards clusters can be made and are based on their chemical composition (see Raman spectrum in figure 1b). HCA clusters parts of the data from PCA or sparse coding and maps it into a 2D figure. In figure 1c the result of this mapping is shown.



(a) SEM images of several cancer cells.



(b) A Raman image of a cell with corresponding pixel characteristics.



(c) Results of HCA performed on preprocessed Raman data of the cells in figure 1a.

Figure 1: In figure 1a a SEM images of several cancer cells are shown. Due to the use of electron beams the morphology is clearly visible. In figure 1b a Raman image and several pixel characteristics are displayed. The Raman image shows the mean intensity captured by the sensor. From that image 3 different samples are used for the calculation of the characteristics. It states that the different colours have different intensities and different peaks. The Raman shift represents difference in wavelength from the captured scattered light and the laser beam. In this stage the data is still very raw. After preprocessing, dimensionality reduction and HCA the Raman data will look like the images in figure 1c.

3 PCA analysis for hyperspectral images

In this section an overview is given of the current method for cancer cell recognition. This section contains a short description of all preprocessing steps, an overview of the PCA algorithm and a brief description of the clustering method.

3.1 Preprocessing

Before PCA is performed, preprocessing techniques are applied on the Raman data. Below, all the preprocessing steps performed on the raw data are given in order[17]:

(i) cosmic rays and outliers are removed;

(ii) the region of interest is divided in cell or background. The area outside this region will not be included;

(iii) the solvent residue of the sample is subtracted by using a linear least squares fit; and

(iv) furthermore, a baseline correction an denoising is executed.

3.2 PCA

3.2.1 Principle components

When the preprocessing is done, the PCA algorithm can be used as a dimensionality reduction tool. The main reasons why this algorithm is used is because of its power to make data insightful and its low computational costs. Suppose this method is used on data containing k pixels with each containing n spectral bands, we have the following data matrix of dimension n x k:

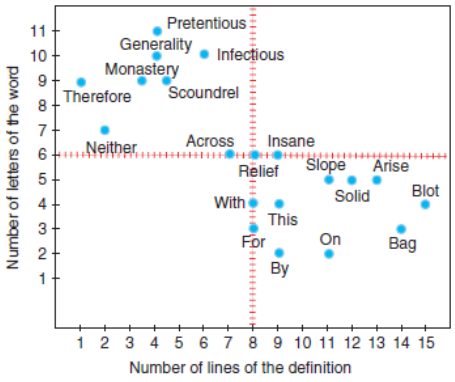
$$X = \begin{bmatrix} X_{11} & X_{12} & \cdots & X_{1k} \\ X_{21} & X_{22} & \cdots & X_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ X_{n1} & X_{n2} & \cdots & X_{nk} \end{bmatrix}$$

The goal of PCA is to find a $m-$dimensional subspace (i.e. $m < k$), while maintaining the utmost of the variation in the data. The new measurements $W_1, \ldots, W_k$ are linear combinations of the column of X, so each W_p $(p = 1, \ldots, k)$ can be expressed as follows.

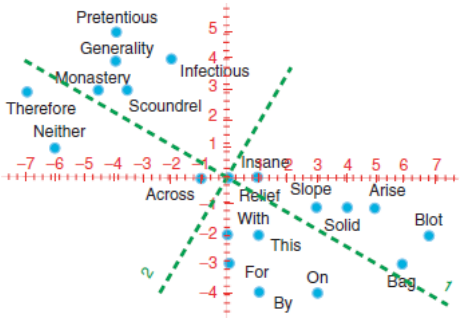
$$\begin{aligned} \mathbf{W}_p &= c_{1p}\mathbf{X}_1 + c_{2p}\mathbf{X}_2 + \cdots + c_{1p}\mathbf{X}_p \\ &= \mathbf{c}_p^T\mathbf{X} \end{aligned}$$

where $\mathbf{c}_p^T = (c_{1p}, c_{2p}, \ldots, c_{kp})$ are constants and $\text{cov}(W_i, W_j) = 0$ for $i \neq j$. The last constraint ensures the new measurements to be orthogonal[2].

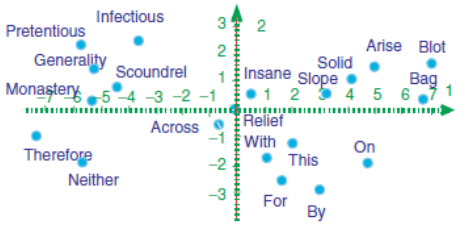
The calculated $\mathbf{W}_p$ $(p = 1, \ldots, k)$ are called *the principle components* (PCs). There is no standard number of PC that need to be calculated but at least a 80% coverage of the variation is suggested[2]. The first PC represents the direction with the biggest variation. The next PC will be orthogonal (because the covariance with the first PC is equal to zero) and represents the direction with the second largest variation. The remainder components are calculated in a similar way.
The calculated PCs helps to make the data insightful. The process needed is drawn in figure 2. In figure 2a the two dimensional data set is drawn in a regular scatterplot. The first task is to find the PCs and this result is drawn in figure 2b (indicated with the green dotted lines). The last step is a replacement of the original axis by the calculated PC (indicated in figure2c). When data needs to be plotted in 3D then the next PC is simply added to the figure[4].



(a) Orginal data.



(b) Calculation the principle components.



(c) Switch the axis.

Figure 2: Schematisation construction principle components[4].

3.2.2 Scores

In the first paragraph the directions with the biggest variations are calculated. These direction are of great importance for the eventual classification of cancer cells. Generally, the directions with highest variation are more important for the classification[10]. Before this classification can be made, the data needs to be transformed to another coordinate system because of the change of axis (as shown in figure 2c). Below this transformation is given algebraically and geometrically.

The mutual relationships between the PC and the data are called *scores*. Scores represent the new positions in a coordinate system where the PC form the new axis. The score of the mth sample on the pth PC can be written as[20].

$$\mathbf{W}_{pm} = c_{p1}\mathbf{Y}_{p1} + c_{p2}\mathbf{Y}_{p2} + \cdots + c_{pk}\mathbf{Y}_{pk}$$

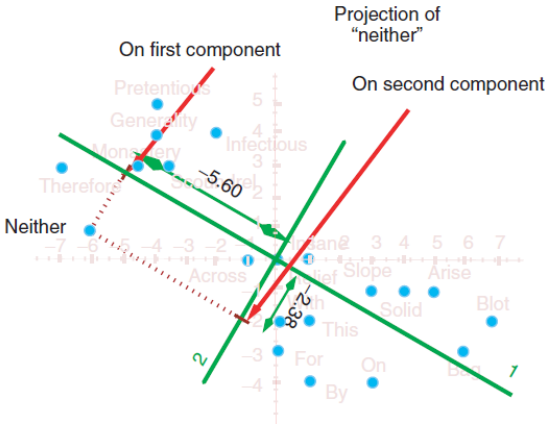


Figure 3: Geometrically interpretation of the scores[4].

These scores can also be interpreted geometrically and is illustrated in figure 3. The figure clarifies that the element 'Neither', compared to the new axis, respectively has a new horizontal and vertical displacement of -5.6 and -2.38.

Now these scores are known, the clustering part can be executed. The scores are used in a *hierarchical cluster algorithm* (HCA) using the *'Ward'* method. In a HCA distances between clusters are the only measurements used. Most HCAs measure distance between elements from different clusters. Ward's method handles distance differently. It states that the distance between clusters A and B is how much the sum of squares grows when the clusters are merged. This means:

$$\Delta(A,B) = \frac{n_A n_B}{n_A + n_B} \|\mathbf{m}_A - \mathbf{m}_B\|^2$$

where $\mathbf{m}_i$ is the center of cluster i and n_i is the number of elements in cluster i. Starting with zero, the merging costs $\Delta(A,B)$ will be added every time clusters are merged. This algorithm tries to keep it as low as possible[27]. During the clustering a fixed number of clusters is used and is set to 9.

4 Sparse coding

In this chapter two dimensionality reduction algorithms are introduced whereby both algorithms are based on sparse coding. First a general introduction of sparse coding will be given. In the second part a description of dictionary learning is presented and afterwards the ISTA and CoD algorithm are presented.

4.1 Sparse coding idea

In sparse coding a signal X will be decomposed in a dictionary D and a sparse matrix Z (i.e., $\underset{n\times k}{X} = \underset{n\times m}{D} \cdot \underset{m\times k}{Z}$). The signal of a pixel $\mathbf{x} \in \mathbb{R}^n$ is a linear combination of basis vectors $\mathbf{d}_i$ $i = [1, \ldots, m]$ plus additive noise ε i.e.,

$$\underset{n\times 1}{\mathbf{x}} = \underset{n\times m}{D} \times \underset{m\times 1}{\mathbf{z}} + \underset{n\times 1}{\varepsilon}. \tag{1}$$

If $\mathbf{z}$ is sparse, this model describes a signal $\mathbf{x}$ with only a few elements from the dictionary D. The Raman data used in this report contains 13228 pixels with 943 spectral bands each and the dictionary is composed with 1000 basis vector, i.e. $D = [\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_{1000}]$. That means that the signal X can be written as:

$$\underset{943\times 13228}{X} = \underset{943\times 1000}{D} \times \underset{1000\times 13228}{Z} \tag{2}$$

where each pixel is expressed as $\underset{(943\times 1)}{\mathbf{x}} = \underset{(943\times 1000)}{D} \underset{(1000\times 1)}{\mathbf{z}} + \underset{(943\times 1)}{\varepsilon}$

and

$$\mathbf{x} = \sum_{i=1}^{m} D \cdot \mathbf{z}_i + \varepsilon \tag{3}$$

where $\{\mathbf{z}_i\}$ are the decomposition coefficients.

This method has recently seen a lot of attention in the fields of machine learning, neuroscience and image processing[13, 24]. Just like PCA, sparse coding can serve as a dimensionality reduction tool. PCA tries to find a set of principle components that represents each pixel in the data, while sparse coding tries to train a dictionary whereby only a few elements will be used for the representation of a pixel[13]. This difference in data representation is illustrated in figure 4. The goal of sparse representations is to find a set of vectors that serves the data while using a minimal number of nonzero elements. In order to find an optimal sparse code, the following minimization problem can be solved:

$$\min_{Z,D} E(X, Z) = \min_{Z,D} \|X - DZ\|_2^2 + \alpha \|Z\|_1 \tag{4}$$

where X is the data, D is the learned dictionary, Z is the sparse matrix and α is a parameter controlling the influence of both terms. The part $\|X - DZ\|_2^2$ directly comes from the goal of the decomposition of signal X in D and Z (see equation (4.1)). Besides that decomposition, another goal is to use a minimal number of nonzero elements from the sparse matrix Z. Counting the number of nonzero elements of Z will generally be done by using $\|Z\|_0$. Since this will make equation (4) a non-convex minimization problem and more sensitive for outliers, the ℓ^1-norm will be used instead[8]. The α in (4) determines how much nonzero elements there are in Z. A large α causes the $\|Z\|_1$ to be small. That means that Z can only contain a small number of nonzero elements. When α is small, the opposite result occurs.

There are cases where a fixed dictionary is suitable for finding a good sparse representation of the data, but in most cases learning a dictionary will drastically improve the results of this method[6]. Although learned dictionaries improve the results, learning them is a computationally expensive procedure [22]. Fixed dictionaries typically lead to a fast transform but are limited in sparsifying the signals and can only be used for specific types of signals.

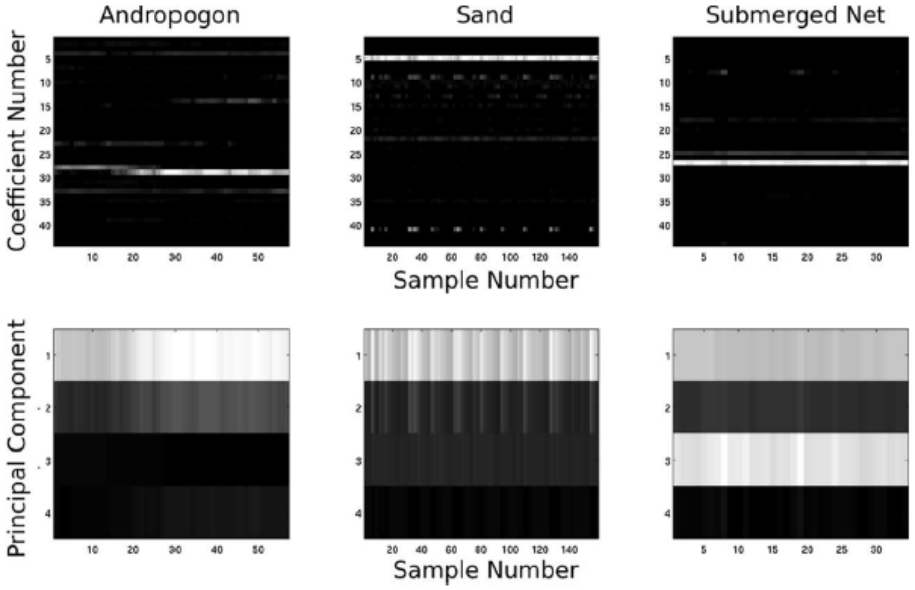


Figure 4: Three classes are represented by using a sparse representation and PCA as dimensionality reduction tool. The brighter the pixel, the higher the intensity of each coefficient. The three images in the second row show that PCA uses all the components to describe the classes, while the sparse representation only uses several coefficients to describe the classes[13].

Sparse coding has a couple of benefits in comparison to methods like PCA. In chapter 3 the property of orthogonal principle components is described. According to [24] this restricts the method to be flexible. Besides that, PCA can only work with pairwise linear statistical dependencies while data often contains crucial higher order statistical dependencies. A method that tackles both problems of PCA, is sparse coding[13, 24].

4.2 Dictionary learning

A dictionary based on training data can be learned in different ways. In this report an overcomplete (more columns than rows) dictionary is chosen because these dictionaries are more flexible and more resilient to noise[5]. Traditional learning methods for overcomplete dictionaries are based on iterative batch methods, whereby in each iteration a cost function is minimized while addressing the training data. The most popular batch method is the *K-means Singular Value Decomposition* (K-SVD)[14, 28]. This method consists of two stages: calculation of a sparse matrix solving a version of equation 4 using a *matching pursuit* algorithm and a stage where the dictionary is updated column-by-column. Disadvantages of this method are its slow convergence[28] and the large memory requirement with large training data[24].
In [24] a method is introduced that solves these problems of K-SVD and is therefore used as the dictionary learning algorithm for this thesis. Below the two parts of the dictionary learning algorithm are listed. The result is a dictionary which is slightly overcomplete with 1000 columns (compared to the 943 spectral bands). A bigger dictionary was not an option due to computational costs.

Algorithm 1 Dictionary learning

function DICTIONARY LEARNING(X, Z, D_0, α)
Require: $\mathbf{x} \in \mathbb{R}^n \sim p(\mathbf{x})$ (random variable that randomly selects a column), $D_0 \in \mathbb{R}^{n \times m}$, T (number of iterations)
Initialize: $A_0 = \mathbf{0}$, $B_0 = \mathbf{0}$
for $t = 1$ to T **do**
    Calculate sparse matrix using ISTA:

$$Z_t = \min_Z \frac{1}{2} \|X - D_{t-1}Z\|_2^2 + \alpha \|Z\|_1 \tag{5}$$

    $A_t = A_{t-1} + Z_t Z_t^T$
    $B_t = B_{t-1} + \mathbf{x}_t Z_t^T$
    Computer D_t using algorithm 2 with D_{t-1} as input.
end for
Return $\mathbf{D}_T$ (the learned dictionary)
end function

For the proof of convergence of algorithm 1 the following assumptions were needed[24]:
Assumption 1. *The data meets a bounded probability density (i.e. the error in the data is bounded)*

Assumption 2. *The smallest eigenvalue of $\boldsymbol{A}_t$ is greater than or equal to a non-zero constant (i.e. $\boldsymbol{A}_t$ invertible).*

Assumption 3. *The smallest eigenvalue of $\boldsymbol{D}_t^T \boldsymbol{D}_t$ $(t = 1, \ldots, T)$ is greater than or equal to a non-zero constant $\forall \boldsymbol{D}_t$ and $\forall \boldsymbol{x} \in \mathbb{R}^n$. Therefore the solution is unique.*

Algorithm 2 Dictionary update

function DICTIONARY UPDATE(D, A, B)
Require: $D \in \mathbb{R}^{n \times m}$ dictionary from algorithm 1,
        $A \in \mathbb{R}^{m \times m}$ and $B \in \mathbb{R}^{n \times m}$
Repeat:
    for $j = 1$ to k **do**
        Update the $j-$th column of D

$$\begin{aligned} \mathbf{u}_j &= \frac{1}{A_{jj}}(\mathbf{b}_j - D\mathbf{A}_j) + \mathbf{D}_j \\ \mathbf{D}_j &= \frac{1}{\max(\|\mathbf{u}_j\|_2, 1)}\mathbf{u}_j \end{aligned} \tag{6}$$

    end for
Until convergence
Return D (the updated dictionary)

4.3 Algorithms to compute sparse codes

The result of the presented method in paragraph 4.2 serves as input for the computation of the sparse matrix. Below two different algorithms are presented that compute the algorithms. In the next paragraphs a short description is given and their pseudo codes.

4.3.1 ISTA

ISTA minimizes equation (4) over Z while fixing dictionary D. The strength of this algorithm is its simplicity. The algorithm of ISTA is listed below[19]:

Algorithm 3 ISTA

function ISTA(X, Z, D, α, L)
    Require: L $>$ largest eigenvalue of $D^T D$
    Initialize: $Z = \mathbf{0}$
    repeat
        $Z = h_{(\alpha/L)}(Z - \frac{1}{L}D^T(DZ - X))$
    until change in Z under a threshold
end function

whereby $h_{\frac{\alpha}{L}}$ is the so-called shrinkage function. This shrinkage function can be expressed as:

$$[h_\theta(V)]_i = sign(\mathbf{V}_i)(|\mathbf{V})_i| - \boldsymbol{\theta}_i)_+ \tag{7}$$

and is used to update Z iteratively with

$$Z^{[k+1]} = h_{(\frac{\alpha}{L})}((I - \frac{1}{L}D^T D)Z^{[k]} + \frac{1}{L}D^T X). \tag{8}$$

This method has proven to converge, even with dense a data matrix[8].

4.3.2 CoD

Besides ISTA, the more efficient Coordinate Descent method (CoD) is introduced. Just like ISTA, a CoD method minimizes equation (4) over Z while fixing D. The difference lies in the selection of components that will be changed per iteration. A CoD method selects one component to modify while ISTA modifies all components, causing a CoD method to converge faster. In algorithm 1 D has the size $n \times m$ and X $n \times k$. That means that the Z is of the size $m \times k$ and therefore the computational complexity is $\mathcal{O}(mn)$, $\mathcal{O}(m^2)$ or $\mathcal{O}(ml)$ with l the average sparsity across samples and iterations. In CoD one component at a time is changed, which takes $\mathcal{O}(n)$ operations. This will be repeated for $\mathcal{O}(n)$ or $\mathcal{O}(m)$ times and thus it is faster than ISTA. The algorithm of CoD is listed below[19]:

Algorithm 4 Coordinate Descent

function CoD(X, Z, D, α, S)
  Require: $S = I - D^T D$
  Initialize: $Z = \mathbf{0}$, $B = D^T X$
  repeat
    $\bar{Z} = h_{(\alpha)}(B)$
    $k =$ index of largest component of $|Z - \bar{Z}|$
    $\forall j \in [1, n] : \mathbf{B}_j = \mathbf{B}_j + S_{jk}(\bar{\mathbf{Z}}_k - \mathbf{Z}_k)$
    $\mathbf{Z}_k = \bar{\mathbf{Z}}_k$
  until change in Z under a threshold
$Z = h_\alpha(B)$
end function

Since CoD updates only one component at a time, this algorithm is not performing a multivariate minimization but a scalar minimization subproblem instead. That means that every subproblem improves the estimation of the solution by minimizing along one direction while fixing others. This principle can easily be shown in 1D. The following minimization problem will then be solved[23]:

$$\min_{\mathbf{z} \in \mathbb{R}^m} E(z) = |\mathbf{z}|_1 + \lambda \left\| D\mathbf{z} - \mathbf{x} \right\|_2^2 \tag{9}$$

where $x \in \mathbb{R}^n$, D an $n \times m$ matrix with $n < m$ and n are the number of spectral band and m the number of basis vectors in the dictionary.

Minimizing this problem will deliver the same result as minimizing equation (4). The solution of this problem can be written as a shrinkage function where

$$\text{shrinkage}(f,\mu) \quad = \quad \begin{cases} f-\mu, \text{ if } f > \mu; \\ 0, \text{ if } -\mu \leq f \leq \mu; \\ f+\mu, \text{ if } f < -\mu; \end{cases} \tag{10}$$

and can be visualized as follows.

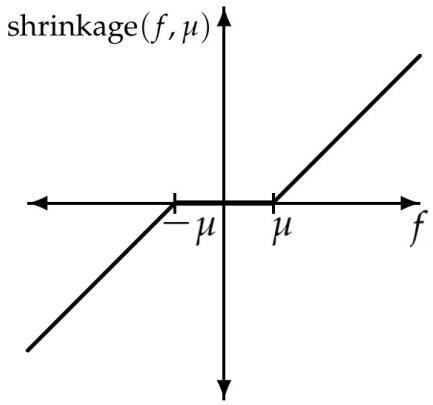


Figure 5: 1D shrinkage displayed in 2D.

5 Results

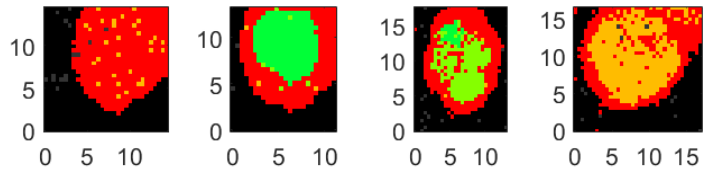
The method discussed in the last chapter is performed on a dataset containing 4 lymphocytes, 4 neutrophils, 4 breast cancer cells (SKBR3), 4 prostate cancer cells (PC3) and 4 LNCaP cells. PCA struggles with finding the distinction between LNCaP and SKBR3 and are therefore part of his dataset. Important is that only PCA is replaced and that the remainder of chapter 3 stays the same. The dictionary is learned in 25 iterations and based on this dictionary, ISTA and CoD are used for the calculation of the sparse matrix Z (see equation (4). This matrix is then used by HCA for making the clusters visible in a classification. Each simulation of ISTA is considered to be converted when

$$\left\| Z^{[k+1]} - Z^{[k]} \right\|_2 < 1$$

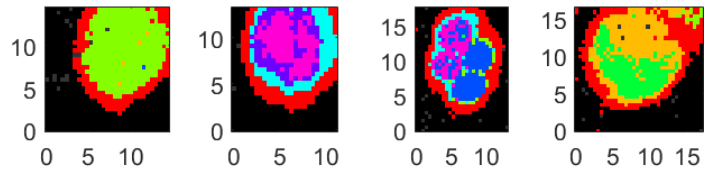
and CoD is considered to be converted when 3500 iterations are reached or

$$\left\| Z^{[k+1]} - Z^{[k]} \right\|_2 < 2.$$

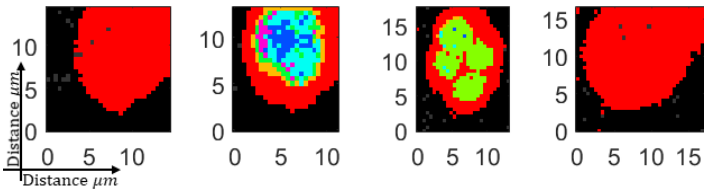
Below results of minimizing equation (4) for different ISTA and COD, for different values of α and 9 clusters are shown.



(a) Cancer cell classification with $\alpha = 1$.

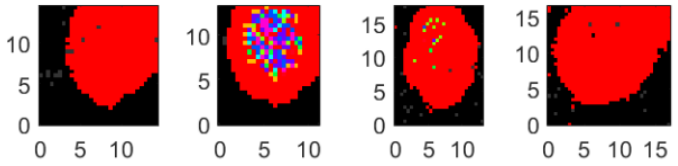


(b) Cancer cell classification with $\alpha = 200$.

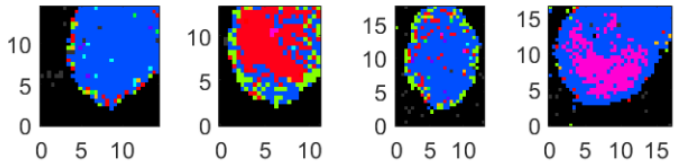


(c) Cancer cell classification with $\alpha = 500$.

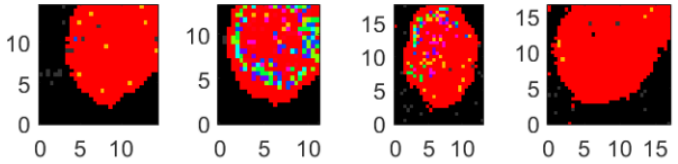
Figure 6: Cancer cell classification for different values of α using ISTA. The cells used in this figure (from left to right): lymphocyte, breast cancer, prostate cancer and LNCaP.



(a) Cancer cell classification with $\alpha = 0.2$.



(b) Cancer cell classification with $\alpha = 50$.



(c) Cancer cell classification with $\alpha = 160$.

Figure 7: Cancer cell classification for different values of α using CoD. The cells used in this figure (from left to right): lymphocyte, breast cancer, prostate cancer and LNCaP.

Figure 6 shows that there is a value of α between 1 and 500 that describes the data best. The value of α determines the extent to which the ℓ^1-norm of the sparse matrix Z is dominant. For example, a high value of α ensures the ℓ^1-norm of Z to be small and consequently give a large value of θ in equation (5). Because most components of D and Z are close to zero, this θ is then a hard threshold and the shrinkage function $[h_\theta(V)]_i$ tends to go to zero for the utmost of the components. Therefore, as stated in figure 6c, a lot of pixels belong to the same red cluster at which it does not contribute anything to the classification. This is illustrated in figure 8 whereby it illustrates that the red cluster has a very low intensity. For the residual components the HCA tries to fit 9 clusters and is shown in the second (from left to right) illustration of figure 6c. Besides visual interpretation, unveiling the properties of the corresponding sparse matrix of $\alpha = 500$, compared to $\alpha = 200$, gives a good insight in the effect of a high value of α for the decomposition of signal Y in DZ. For $\alpha = 500$ the matrix Z is nonzero for 0.54% of the elements and it uses on average 5.44 elements to describe a pixel of Y (see equation (4.1)). That in comparison to 3.3% and 33.02 respectively for $\alpha = 200$.
Figure 7 also presents the fact that there is an optimal value for α for CoD. Besides that, the difference in active components per pixel is also occurring for the CoD algorithm. In the case of $\alpha = 160$ for CoD, only 0.45% of the elements are nonzero and on average 4.53 elements are used to describe one pixel. On the other hand, 1.51% of the elements are nonzero for $\alpha = 50$ and uses on average 15.1 elements to describe one pixel.

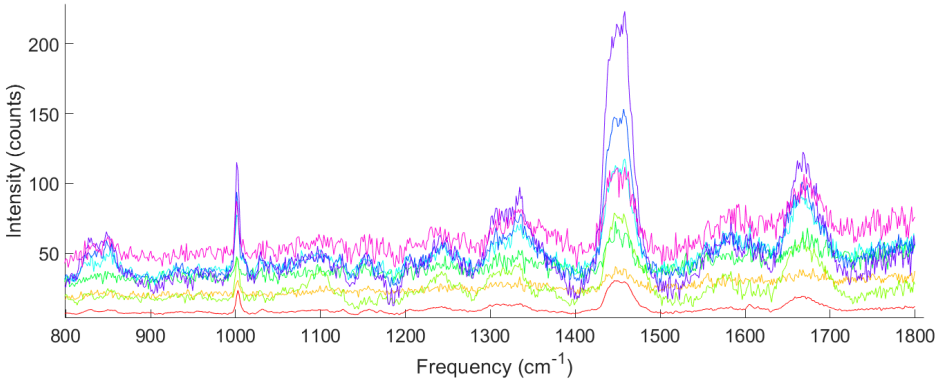


Figure 8: The measured intensity for the coloured clusters per frequency.

5.1 Optimizing α

The figures above have shown that there is an optimal value α for this classification which describes the data best. For this optimization cluster validation tool *Silhouette Validity Index* (SVI) is used because this index can be used to test the input (Z) for the clustering for different values of α and can also be used for the determination of the number of clusters[7, 29]. SVI is an internal cluster validation index that is used in situations when no ground truth is known. Ground truth is data where the classification preferable has a high accuracy. The SVI for the i^{th} data point is defined as:

$$S_i = \frac{b_i - a_i}{max(a_i, b_i))} \quad , -1 \leq S_i \leq 1 \tag{11}$$

where

(i) a_i is the average Euclidean distance of the i^{th} data point to all other points in the same cluster;

(ii) b_i is the average Euclidean distance of the i^{th} data point to all other points in the next nearest cluster.

A SVI between -1 and 0 states that the clustering is insufficient and a SVI close to 1 states that the clustering is sufficient.

Below a table is shown with the SVIs from ISTA and CoD for different values of α. For these calculations the number of clusters is kept constant at 9.

α	SVI ISTA	α	SVI CoD
1	0.0118	0.2	0.6271
30	0.0833	40	0.4705
60	0.1827	75	0.4328
200	0.2597	100	0.4872
235	0.2663	105	0.5038
285	0.4395	110	0.4755
500	0.8354	150	0.1052

Table 1: Parameter optimization for ISTA and CoD using the SVI and 9 clusters.

The table shows that an increase in α ensures an increase in SVI for the outputs of ISTA. The SVIs from CoD give a peak around the value $\alpha = 105$ but further analyis is needed to indicate the optimal value of α. So based on this validation tool alone an optimal value for α can not be chosen and therefore another internal validation index, *Dunn's validity index* (DVI), is used for the determination of an optimal value of α. Just like the SVI, DVI also test the input of this index (Z) and can be calculated as follows[7]:

$$D = \min_{1 \leq i \leq k} \left(\min_{i+1 \leq j \leq k} \left(\frac{\text{dist}(c_i, c_j)}{\max\limits_{1 \leq i \leq k} \text{diam}(c_l)} \right) \right) \tag{12}$$

where

$\text{dist}(c_i, c_j)$ is the distance between cluster c_i and c_j.
$\text{dist}(c_i, c_j) = \min\limits_{x_i \in c_i, x_j \in c_j} d(x_i, x_j)$
$d(x_i, x_j)$ is the distance between data points x_i and x_j.
$\text{diam}(c_l)$ is the diameter of cluster c_l where
$\text{diam}(c_l) = \max\limits_{x_l, x_2 \in c_l} d(x_1, x_2)$.

According to this tool, the higher the index the better the classification. Below in figure 9 the DVIs are calculated for several values of α. Each calculated DVI value is based on the average of 50 calculations because the DVI has not one specific value per α per iteration. The figure shows a clear peak at $\alpha = 200$ for ISTA and at $\alpha = 103$ for CoD. Therefore, these values are used as the optimal value for α.

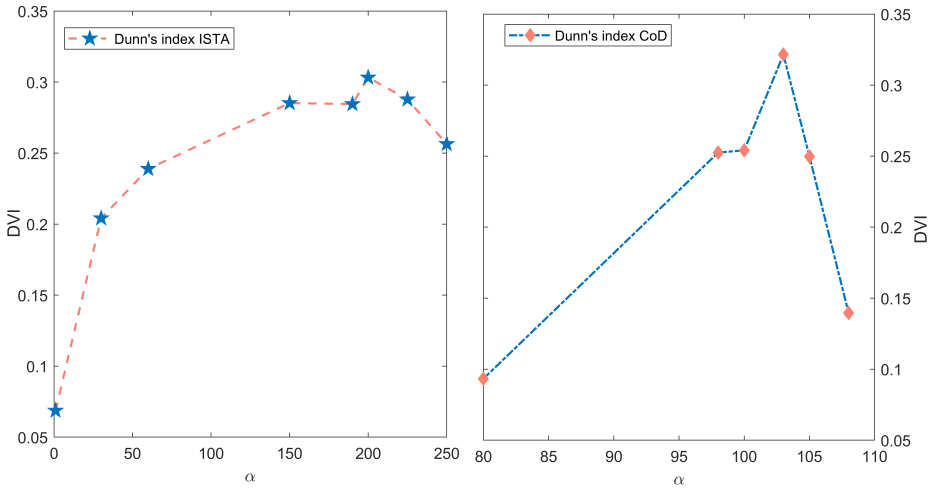


Figure 9: DVI for different values of α using ISTA and CoD.

5.2 Optimizing number of clusters

In the previous paragraph $\alpha = 200$ is set as the optimal value for ISTA and $\alpha = 103$ for CoD. These values are then used for the determination of the optimal value for the number of clusters. Both SVI and DVI are used to determine this value and the results are shown in the table below.

# clusters	SVI	DVI
9	0.2597	0.3027
10	0.2594	0.2937
11	0.2594	0.2924
12	0.2602	0.3001

Table 2: Determination of the optimal number of clusters using the SVI and DVI on ISTA.

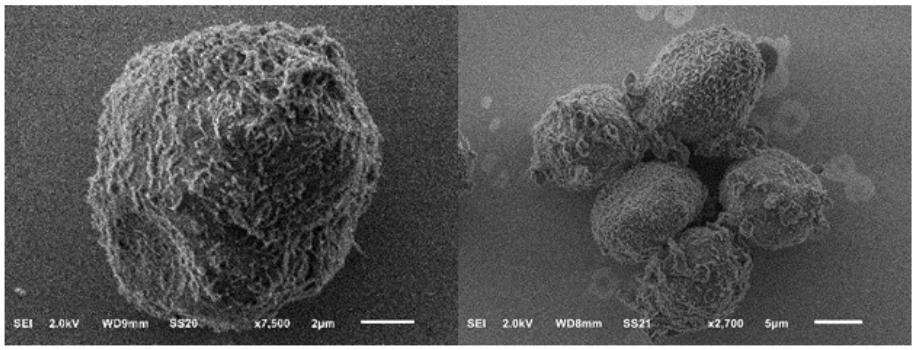
# clusters	SVI	DVI
9	0.4901	0.3216
10	0.4914	0.1864
11	0.4819	0.2685
12	0.4832	0.1334

Table 3: Determination of the optimal number of clusters using the SVI and DVI on CoD.

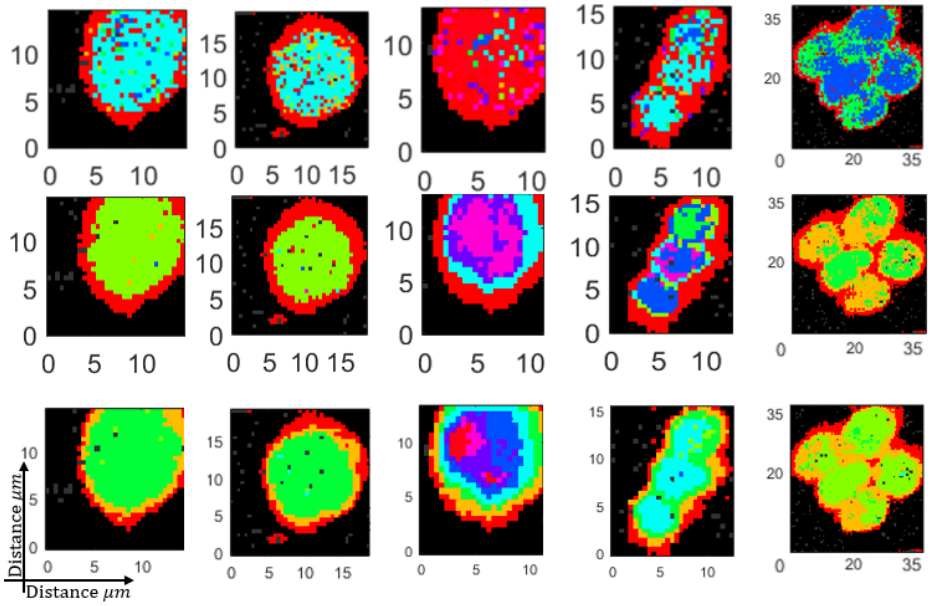
Based on these results 9 clusters are used for the classification of the cancer cells for both ISTA and CoD.

5.3 Comparison with PCA

For the comparison, the same validation tools are used for the classification of the method discussed in section 3. First, the classifications of the 5 different cancer cells are given below, then in table 4 the results of the validation tools are presented.



(a) SEM images from PC3 cell and the SKBR3.



(b) Difference in classification between CoD, ISTA and PCA (from top to down). From left to right: LNCaP, PC3, neutrophil, lymphocyte and SKBR3.

Figure 10: Comparison between classification via PCA and sparse coding, with in figure 10a the corresponding SEM images and in 10b the difference in classification.

	SVI	DVI
ISTA	0.2597	0.3027
CoD	0.4901	0.3216
PCA	0.6166	0.0921

Table 4: Overview of the results of the validation tools SVI and DVI, using PCA and sparse coding.

5.4 Noise resilient

For testing the noise resilience of the algorithms PCA, ISTA and CoD, Gaussian noise is added to the data. Afterwards, the sparse matrix Z is calculated with the algorithms and the results are visually tested and validated with SVI and DVI. The corresponding probability density function is given as follows:

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} \cdot e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

were μ and σ are the noise parameters. In this model the μ is set to 0 and σ to 1. Gaussian noise is a good method for testing the noise resilience of algorithms because it resembles real world cases[12]. In table 5 the resulting SVI and DVI are presented and below the table the visual interpretation is shown.

	SVI	DVI
ISTA with noise	0.2534	0.2978
CoD with noise	0.4894	0.2705
PCA with noise	0.6103	0.0814

Table 5: Results of validation with ISTA using SVI and DVI.

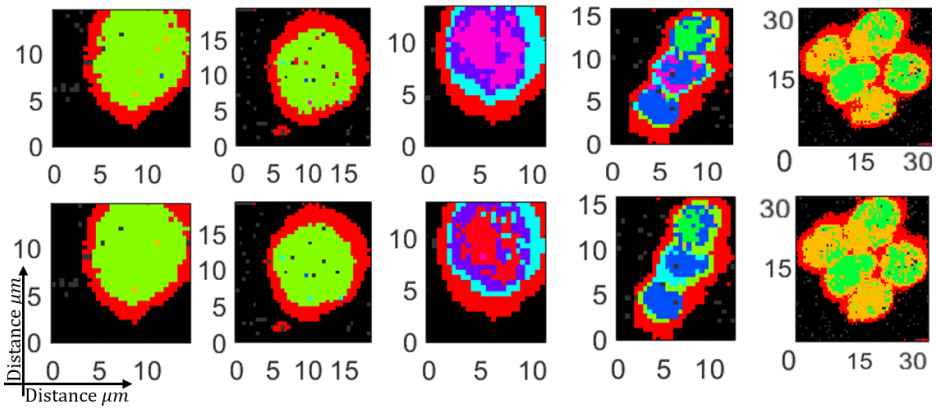


Figure 11: Normal classification ISTA (top) versus a classification with added Gaussian noise.

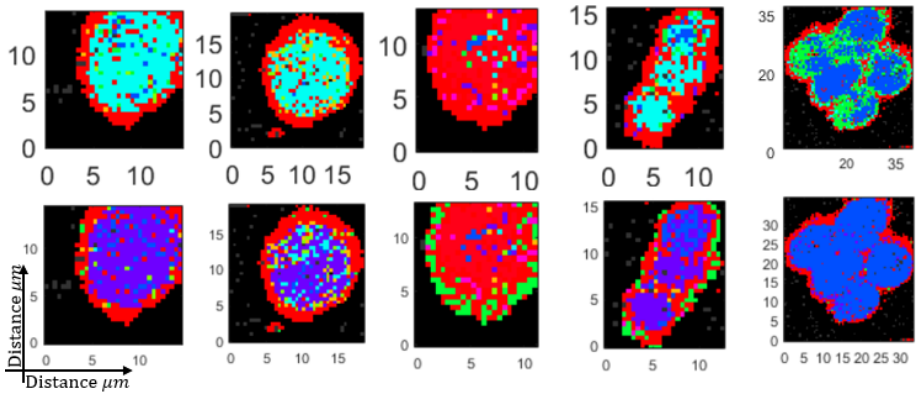


Figure 12: Normal classification CoD (top) versus a classification with added Gaussian noise.

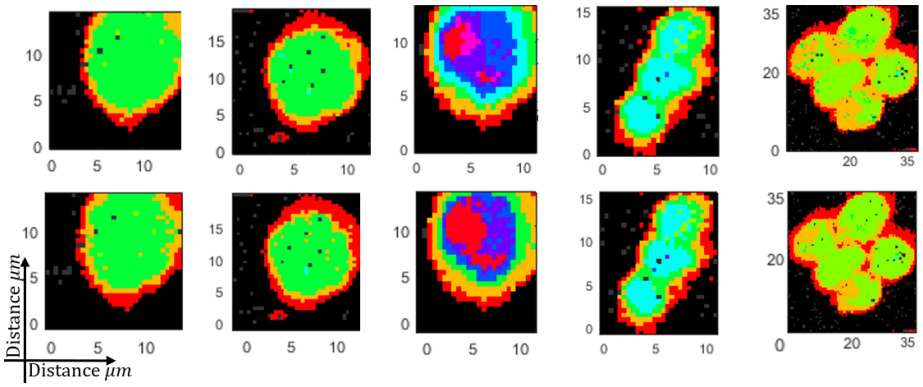


Figure 13: Normal classification PCA (top) versus a classification with added Gaussian noise.

5.5 Summary of the results

In this chapter the results of two methods for replacement of PCA are presented. All methods were able to classify cancer cells but were not able to distinguish LNCaP with SKBR3 cells. Besides that, ISTA contained the more clear classifications in comparison with CoD (see figure 10b). Furthermore, the proposed methods and PCA all turned out to be noise resilient. This conclusion is based on a small changes in classifications, SVI and DVI.

6 Conclusion & Recommendations

This report presented two methods based on sparse coding that were used to make hyperspectral camera data insightful. These dimensionality reduction algorithms, ISTA and CoD, made use of a pre-learned dictionary. The α of equation (4) is optimized via two internal clustering validation tools. The proposed methods were applied to a data set containing 4 lymphocytes, 4 neutrophils, 4 breast cancer cells, 4 prostate cancer cells and 4 LNCaP cells.

Sparse coding has led to acceptable classifications of cancer cells and the corresponding algorithms turned out to be resilient to noise. ISTA gave more clear distinctions between cancer cells than CoD. Based on the two internal validation tools it can not be said which algorithm, PCA or ISTA, is better at classifying cancer cells. This is because PCA had a higher SVI and ISTA scored better on the DVI. Besides that, both method were able to distinct neutrophils, lymphocytes and breast cancer, but were both not able to distinct LNCaP with PC3.

The results show that the resulting sparse model contains outliers and the model struggles to distinct the orange and red cluster (see the leftmost figure in figure 10b). If further research on this subject is conducted, I recommend to change a few parameters. In this research convergence for ISTA was reached when $\left\| Z^{[k+1]} - Z^{[k]} \right\|_2 < 1$. Putting the norm difference closer to zero will lead to a more converted solution of ISTA. This also applies for CoD, whereas there was not enough time to run the code longer. Besides that, the learned dictionary formed an input for ISTA and CoD and was learned within 25 iterations. In [13] was stated that most dictionaries were well-converted after 1000 iterations, but they recommend to upscale that number of iterations even more. Due to lack of a strong computer and time, I was not able to do these number of iterations. The 25 iteration and the large norm difference were therefore insufficient.
Besides this recommendation for longer computation time, I recommend to obtain the results based on more data. In this report 20 cancer cells in total are used for the input of the algorithms. More data and longer computations would improve the results of this research.

During this research one dictionary was learned based on the Raman microscopic data. Because a cell contains multiple materials, a clear classification per pixel is therefore hard. In [3] a method is proposed which trains multiple dictionaries based on labeled data with high accuracy (ground truth). In this case, this ground truth can be assessed by using a microtome. A mircotome cuts very thin slices of a sample (cancer) cell and can be examined with a microscope in a much easier and precise way. After collecting data from several sorts of cancer cells, the dictionaries can be learned such that each material has its footprint in the shape of a Raman spectrum (see figure 1b). The method in [3] then selects per pixel which dictionary describes the data best and based on that information a classification is made.

Besides more dictionaries, better understanding of the size of the dictionary would probably improve the classification. Expected is that there is an optimal size for this application but due to lack of time, this research has not been executed.

Furthermore, two internal classification validation tools are used for the optimization of α and the number of clusters. For a better optimization it would be better to perform a research on which validation tools suit this problem best or on usage of more validation tools (internal and/or external), whereas external validation tools are based on ground truth. This ground truth can for instance be accessed by the use of a microtome. One of the validation tools, Dunn's index, proved to be inconsistent. If this validation is used again, it would be recommended to calculate the average over more iteration.

References

[1] Cancer, key facts, howpublished = http://www.who.int/en/news-room/fact-sheets/detail/cancer, note = Accessed: 2018-05-30.

[2] Pca notes. http://www.maths.manchester.ac.uk/~peterf/MATH38062/. Accessed: 2018-05-06.

[3] A. Castrodad, Z. Xing, J. G. E. B. L. C. and Sapiro, G. (2010). Discriminative sparse representations in hyperspectral imagery. *Image Processing (ICIP)*, pages 1313–1316.

[4] Abdil, H. and Williams, L. J. (2010). Principal component analysis. *WIREs Comp Stat*, 2:433 – 459.

[5] Agarwal, A., Anandkumar, A., Jain, P., Netrapalli, P., and Tandon, R. (2014). Learning sparsely used overcomplete dictionaries. *JMLR: Workshop and Conference Proceeding*, pages 1–15.

[6] Aharon, M., Elad, M., and Bruckstein, A. (2006). K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE TRANSACTIONS ON SIGNAL PROCESSING*, 54:4311–4322.

[7] Ansari, Z., Azeem, M., Ahmed, W., and Babu, A. (2011). Quantitative evaluation of performance and validity indices for clustering the web navigational sessions. *World of Computer Science and Information Technology Journal(WCSIT)*, 1:217–226.

[8] Beck, A. and Teboulle, M. (2009). A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging sciences*, 2:183–187.

[9] Behrmannl, J., Etmann, C., Boskamp, T., Casadonte, R., Kriegsmann, J., and Maass, P. (2018). Deep learning for tumor classification in imaging mass spectrometry. *Bioinformatics*, 34:1215–1223.

[10] Ben-Hur, A. and Guyon, I. (2003). Detecting stable clusters using principal component analysis. In Brownstein, M. and Kohodursky, A., editors, *Functional Genomics: Methods and Protocols*, pages 159–182. Humana press.

[11] Bioucas-Dias, J. M., Chanussot, J., an Qian Du, N. D., Gader, P., Parente, M., and Plaza, A. (2012). Hyperspectral unmixing overview: geometrical, statistical, and sparse regression-based approaches. *IEEE journal of slected topics in applied earth observations and remote sensing*, 5:356–366.

[12] Boyat, A. K. and Joshi, B. K. (2015). A review paper: Noise models in digital image processing. *Signal Image Processing : An International Journal (SIPIJ)*, 6:64–75.

[13] Charles, A. S., Olshausen, B. A., and Rozell, C. J. (2011). Learning sparse codes for hyperspectral imagery. *IEEE journal of selected topics in signal processing*, 5:963–965.

[14] Chenglong, B., Ji, H., Quan, Y., and Shen, Z. (2016). Dictionary learning for sparse coding: algorithms and convergence analysis. *IEEE transactions on pattern analysis and machine intelligence*, 38:1356–1358.

[15] Crowleya, E., Nicolantonio, F. D., Loupakis, F., and Bardelli, A. (2013). Liquid biopsy: monitoring cancer-genetics in the blood. *Nature reviews, clinical oncology*, 10:472–484.

[16] Cruz, M. R., Costa, R., and Cristofanilli, M. (2006). The truth is in the blood: The evolution of liquid biopsies in breast cancer management. `https://am.asco.org/truth-blood-evolution-liquid-biopsies-breast-cancer-management`. Accessed: 2018-06-17.

[17] Enciso-Martinez, A., Timmermans, F. J., Nanou, A., Terstappen, L., and Otto, C. Sem-raman image cytometry of cells. *The Royal Society of Chemistry*.

[18] Field, D. J. and Olshausen, B. A. (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381:607–609.

[19] Gregor, K. and LeCun, Y. (2010). Learning fast approximations of sparse coding. *Proceedings of the 27th international conference on machine learning*.

[20] Holland, S. M. Principal components analysis (pca). `https://strata.uga.edu/software/`. Accessed: 2018-05-06.

[21] Ilie, M. and Hofman, P. (2016). Pros: Can tissue biopsy be replaced by liquid biopsy? *Transl. Lung Cancer Res.*, 5:420–423.

[22] Julazadeh, M. (2012). Medical image segmentation and classification based on sparse representation and dictionary learning algorithms. *Theses and dissertations*, pages 13–17, 29–39.

[23] Li, Y. and Osher, S. (2009). Coordinate descent optimization for l1 minimization with application to compressed sensing; a greedy algorithm. *Inverse Problems and Imaging*, 3:487–503.

[24] Mairal, J. and Bach, F. (2009). Online dictionary learning for sparse coding julien. *International Conference on Machine Learning*, 26.

[25] Neugebauer, U., Clement, J. H., Bocklitz, T., Krafft, C., and Popp, J. (2010). Identification and differentiation of single cells from peripheral blood by raman spectroscopic imaging. *Journal of Biophotonics*, 3:579–582.

[26] Rodarmel, C. and Shan, J. (2002). Principal component analysis for hyperspectral image classification. *Surveying and land information systems*, 62:115–118.

[27] Shalizi, C. Distances between clustering, hierarchical clustering. `http://www.stat.cmu.edu/~cshalizi/350/lectures/08/lecture-08.pdf`. Accessed: 2018-05-08.

[28] Tariyal, S., Majumdar, A., Singh, R., and Vatsa, M. Greedy deep dictionary learning. *CoRR*, 54abs/1602.00203.

[29] Wang1, K., Wang, B., and Peng, L. (2009). Cvap: Validation for cluster analyses. *Data Science Journal*, 8:88–93.

A MATLAB implementations

A.1 The dictionary learning algorithm

```
function [X,beta]=dictionary(X,Y,Z,epsilon,alpha,L,theta,W)
[a b]=size(X);
[c d]=size(Y);
A=zeros(b);
B=zeros(a,b);
for t=1:50;
    beta=ISTA(Y,Z,X,epsilon,alpha,L,theta,W);


    A=A+beta*transpose(beta); %update A
    B=B+Y*transpose(beta); %update B

    X=dictionaryupdate(A,B,X); %upload dictionary
end
end
```

A.2 The dictionary update

```
function [D_old]= dictionaryupdate(A,B,D)
D_old=D;
[q,r]=size(D);
D_new=zeros(q,r);
norm_D=10;

eta=1;
j=1;
while norm_D>eta;
    for j=1:r;
        if A(j,j)==0; %special treatment for singularities
            u_j=(10^-14)*(B(:,j)-D_old*A(:,j))+D_old(:,j);
        else
            u_j=(1/A(j,j))*(B(:,j)-D_old*A(:,j))+D_old(:,j);
        end

        D_old(:,j)= (1/max(norm(u_j),1)).*u_j;

    end
    norm_D=norm(D_old-D_new) %check norm
    D_new=D_old;

end
end
```

A.3 ISTA

```
function [W]=ISTArobust(input,Z,D,epsilon,alpha,L,theta)
norm_ista=10;
W=Z-(1/L)*transpose(D)*(D*Z-input);
while norm_ista>epsilon;
    Z=W;
    W=sign(Z-(1/L)*transpose(D)*(D*Z-input))
       .*max(abs(Z-(1/L)*transpose(D)*(D*Z-input))-theta,0);
end
end
```

A.4 CoD

```
function [Z]=CODrobust(X,Z,D,S,alpha,epsilon)
B=transpose(D)*X;
norm_cod=10;
Z_new=Z;
[a,b]=size(B);
Z=zeros(a,b);
k = 0;
while norm_cod(end)>epsilon && k < 3000
    Z_bar=sign(B).*max(abs(B)-alpha,0);
    absolute=abs(Z-Z_bar);
    [~, index] = max(absolute);

    for i = 1:b
        for j=1:a
            B(j,i)=B(j,i)+S(j,index(i))
               *Z_bar(index(i),i)-Z(index(i),i));
        end
        Z_new(index(i),i)=Z_bar(index(i),i);
    end

    norm_cod(end+1) = norm(Z_new-Z);
    Z=Z_new;
    k = k + 1;
end
Z=sign(B).*max(abs(B)-alpha,0);
plot(norm_cod)
end
```