Gene Expression Guided Distance Metric Learning for Breast Cancer Whole Slide Image Analysis

Kajsa Ledesma Eriksson

Master's thesis
2022:E36

Lund University

Faculty of Engineering
Centre for Mathematical Sciences
Mathematics

Centrum Scientiarum Mathematicarum

Master's Theses in Mathematical Sciences 2022:E36
ISSN 1404-6342

LUTFMA-3483-2022

Mathematics
Centre for Mathematical Sciences
Lund University
Box 118, SE-221 00 Lund, Sweden

http://www.maths.lth.se/

Gene Expression Guided Distance Metric Learning for Breast Cancer Whole Slide Image Analysis

Kajsa Ledesma Eriksson

Spring 2022

LUND
UNIVERSITY

Supervisors

Anders Heyden, LTH
Mattias Rantalainen, KI

Examiner

Mikael Nilsson, LTH

Abstract

Female breast cancer is a complex and heterogeneous disease that accounts for most of the deaths caused by cancer in women worldwide. To stratify breast cancer patients into treatment groups is a challenging task, and in recent years, analysis of the genes active in the tumour has been used in the decision of cancer therapy. Although gene expression analysis is expensive and not available for most breast cancer patients, calling for a more cost-effective and reproducible alternative.

In the following thesis, a gene expression guided embedding extractor network is trained that maps whole slide images of female breast cancer tumours into embeddings in a metric space in which relative distances should be similar to the distances in the corresponding gene expression data. In the thesis, the embedding extractor network is the convolutional-based neural network ResNet-50. The metrics studied for distance measurements were the L1-distance d_{L1}, cosine distance d_{CL}, L2-distance d_{L2} and an average L1-distance d_{MAD}. In the thesis, each whole slide image consisted of smaller tiles. Examining the model's performance basing the distance measurement on one or multiple tiles from each slide, it was seen that the best performing metric was d_{MAD} with the multi-tile calculation. The final model gave a Pearson correlation coefficient between predicted- and ground truth distances of $\rho = 0.631$ on the test data. The statistical significance of the correlation between predicted- and ground truth distances was evaluated with a Mantel test, resulting in a p-value $< 10^{-15}$.

The thesis suggests that an image-based approach could serve as a potential alternative to gene expression profiling, with the possibility of further research and evaluation.

Keywords: Whole Slide Image, Breast Cancer, Deep Metric Learning, Histopathology, Deep Learning, Image Analysis

Acknowledgements

I would like to begin by thanking my supervisor Mattias Rantalainen at Karolinska Institutet, Department of Medical Epidemiology and Biostatistics, for all the input, ideas, knowledge and support throughout the thesis. I also want to express a great thanks to everyone in the research group at Karolinska Institutet for all the interesting discussions, contributions, and weekly journal clubs over the past months. I am so grateful to have been given the opportunity to do my Master's Thesis in such a talented and experienced team within this interesting and important area of research.

Furthermore, I would like to thank my supervisor at Lund University, Anders Heyden for the valuable feedback he has given me, in addition to making this thesis possible.

Contents

Chapter 1

Introduction

1.1 Background

Female breast cancer is the most prevalent form of cancer and accounts for most deaths caused by cancer in women worldwide [1]. The complexity of the biological phenotypes and morphological subtypes of breast cancer tumours gives rise to the challenging task of identifying the most beneficial treatment for each individual patient without the risk of overtreatment or undertreatment. The current strategy to decide cancer therapy is to take into consideration both factors about the patient, such as age and menopausal status, as well as key features of the cancer tumour. Some of the features looked at are the tumour size and histological grade based on morphological studies of tissue samples from surgery. It is also taken into account if there is an invasion of tumour cells in the lymphovascular area or if the cancer has spread to distant organs or lymph nodes [2]. However, a morphologic study alone is insufficient both when stratifying breast cancer patients into treatment groups with precise outcome predictions and when describing the complexity in the biological behaviour of the breast tumour. Therefore, an additional analysis on a molecular level is often considered, capturing the molecular alterations and -patterns that underlie the biology and pathophysiology of breast cancer [2, 3].

Beginning with the pioneering work by Therese Sørlie et al., it was shown that variations in which genes that were expressed in the tumour were giving rise to five molecular subtypes of breast cancer (Basal-Like, HER2-enriched, Normal Breast-Like, Luminal A, Luminal B) reflecting both the clinical outcome and the phenotype [4]. The study demonstrates the prognostic and clinical value in the gene expression profiles

with the possibility of stratification of patients with respect to their predicted clinical outcome from therapy. Although the clustering of breast cancer into five molecular subtypes is a simplification of a complex biological system. After classification there still remains a great diversity in the discriminative subtypes both with respect to the histopathological features but also in the response to chemotherapy and survival rates [5].

1.2 Project Motivation

Gene expression analysis has not only given a better understanding of the biological diversity in breast cancer, it also acts as an aid in cancer therapy with prognostic information for individual patients. However, gene expression profiling is expensive and not available for most breast cancer patients [6]. Hence, a more cost-effective and reproducible approach is sought after. In this study, it will be evaluated if an image-based approach can be used to extract the information given by gene expression profiling from histological whole slide images (WSI).

The aim of the thesis is to train a deep neural network (referred to as the embedding extraction network) that maps WSIs of breast cancer tumours into a metric space in which images with similar gene expression profiles should be close to each other and images with dissimilar gene expression profiles should be far apart. In the thesis, different distance measurements, or metrics, will be considered to evaluate which metric space that is best suited for incorporating the similarities in the gene expressions. In addition to serving with a more reproducible and cost-effective alternative to molecular subtyping via gene expression analysis, the study aims at capturing a more complex, continuous mapping of the diversity in breast cancers in the metric space guided towards the multi-dimensional space spanned by the gene expression data. The aspiration for the study is for the continuous mapping of breast cancer in the metric space to describe both the relative similarities between the different subtypes and to incorporate the diversity inside the subtypes.

1.3 Research Objective

The research question to be studied in the following thesis will be the following:

Can a deep neural network be trained that maps histopathological whole slide images of breast cancer tumours to a metric space describing the pairwise distances between the gene profiles of the tumour?

1.4 Outline of Report

The report will begin with an introduction to the theory used in the thesis. First a brief introduction of the breast cancer subtypes, and gene expression analysis will be presented followed by an overview of the area of histopathology. The technical theory underlying the embedding extraction network will then be explained, followed by some previous work applying distance metric learning in histopathology.

In Chapter 7 the implementation of the proposed method will be explained, beginning with an overview of the method and an introduction to the data used in the study. The development of the model will then be explained, where the proposed strategies for guiding the embedding extraction towards describing the distances between gene expressions will be put forward. In the method, different metric spaces are examined with the aim of finding a suitable metric in which to relate extracted embeddings.

In the last three chapters of the thesis, the results are presented, followed by a discussion and conclusion of the work.

1.5 Delimitations

Due to the long training time of the models examined in the following thesis, the depth of the study was delimited in relation to the time frame at hand. As will be discussed throughout the report, only a subset of tiles from each WSI will be considered in the distance measurements. It is also assumed that every tile from the same WSI has the same gene expression profile. Concerning the tuning of the network, not all combinations of hyperparameters were tuned for each model examined. Instead, the time was more focused on examining the different metric spaces in which to map the WSIs.

The study is not handling subclassification of breast cancer and the aim of the thesis is not to perform classification based on the gene expressions from each tumour. The usage of the annotated molecular subtypes of the data is not used in the training of networks, but is only used to illustrate the relative locations of the embeddings in the metric spaces.

Chapter 2

Breast Cancer

The risk for breast cancer is affected by several components, ranging from genetic predisposition, where approximately 10% of breast cancers are hereditary [7], to environmental- and lifestyle factors such as maternal age of first pregnancy and physical activity. The mechanisms that initiate breast cancer are unknown; however, studies show that the disease appears to evolve along two molecular pathways of progression [7]. One of which is associated with lower grade breast cancers and the other with higher grade, and more aggressive, breast cancers. In the two pathways of progression, different hormone receptors are active, which determines the biological process of the cell. A majority of the genes active on the low-grade path are connected to the oestrogen receptor (ER), while the higher-grade path shows expression of the human epidermal growth factor receptor 2 (HER2), which, among other receptors, plays a key role in cancer progression [7].

2.1 Breast Cancer Subtypes

Breast cancer is a heterogeneous disease that differs both between patients and within the tumour itself [8]. Over the past decade, cancer biology has been the focus of decision-making for therapy that takes into account the heterogeneity of the disease. As mentioned above, studying gene expression has given rise to five subtypes of breast cancers: Luminal A, Luminal B, HER2-Enriched, Basal-Like and Normal Breast-Like [4, 9]. These subtypes are based on the expression of gene products in which Luminal A and B expresses, for example, the oestrogen receptor (ER), while HER2-Enriched expresses the human epidermal growth factor receptor 2 (HER2) without ER expression [7]. In current clinical practice, molecular and histological characteristics are used as the ba-

sis for stratification of cancer cases. Tumours that express the ER and / or progesterone receptor (PR) are considered hormone receptor positive breast cancers, while tumours that do not express the ER, PR or HER2 receptors are considered triplet negative breast cancer (TNBC) [7]. To define molecular subtypes, variations in gene expression profiles are studied with gene expression analysis.

2.1.1 Gene Expression Analysis

Gene expression is the complex process where information from the genome of a cell is transcribed via messenger RNA (mRNA) and translated into the production of proteins giving rise to traits and functions of a cell. This transcriptome is of importance for gaining knowledge about the phenotype of a cell or tissue, as well as for understanding its functionality and the development of disease [10]. Each cell is expressing only a part of its genes from the genome giving rise to that different cells can have different functionality depending on the level of expression of genes [9]. To measure the level of gene transcripts, that is, the level of mRNA gene products in a tissue sample, RNA sequencing can be used [10]. In RNA sequencing, RNA is extracted from the tissue sample and, by the steps of process and analysis, it can be determined which genes are active and to what extent the genes are transcribed in the sample [10]. This information can be used to study the difference between healthy and cancerous cells to better understand the genetic origin of defective functionality and to provide treatment targets [9].

Using gene expression analysis, a gene expression profile can be defined for a tissue sample or a cell. This profiling has given a better understanding of breast cancer biology and serves as a tool for the prediction of response to therapy, disease prognosis, and breast cancer subclassification [9].

2.1.2 PAM50 Gene Signature

To classify a breast cancer tumour as one of the five intrinsic molecular subtypes, the measured expression levels of a set of 50 genes in the tumour have been shown to be sufficient [11]. The set of genes is called the PAM50 gene signature and in addition to identifying the subtype of the breast cancer sample, the set of genes serves as a source of information about the probability of recurrence of the disease [11].

2.1.3 Clinical Relevance of Breast Cancer Subtypes

The prognostic information gained by studying the gene expression profile of a breast cancer tumour facilitates the therapeutic decision-making, where information of survival rate and probability of relapse can be taken into consideration [12]. In terms of survival rate, Basal-Like breast cancers are observed to be associated with a lower survival rate followed by HER2-enriched, Luminal A and Luminal B [13, 14].

The correlation between molecular subtypes of breast cancer and relapse was studied in [12] where Luminal A tumours were found to be associated with a low rate of local- and regional relapse while Luminal B, HER2 enriched, and Basal-like showed a higher rate of relapse. When studying the rate of local relapse at 10 years after mastectomy, 8% of the studied patients with Luminal A tumours had local relapse, compared to 22% for patients with Luminal B [12]. This study is in line with several studies, [2, 4, 15, 16], showing that Luminal A tumours have the best prognosis of the subtypes, while HER2-Enriched and Basal-Like tumours have worse prognosis regarding survival rate.

These studies illuminate the impact of the gene expression profile on the pathophysiology of breast cancer, where cancer functionality, clinical outcome, and aggressiveness of the cancer are dependent on the underlying gene expressions.

Chapter 3

Histopathology

Histopathology refers to the study of disease in tissue on the basis of microscopical examination [17]. Pathological analysis of histological images is the centre of cancer diagnosis and decision-making for caner treatment in current clinical practise [18].

3.1 Image Analysis in Histopathology

In the past decade, the area of histopathology has undergone a transformation toward a digital workflow, where pathologists can study histological images digitally instead of through microscopes. This facilitates both the examination of the specimen, where the software not only enables comparison, measurement and reproducibility, but also allows pathologists to assess images from remote sites and work in collaboration with other pathologists [19].

The histopathological images are produced via high-resolution digital slide scanners that produce whole slide images (WSI). A tissue sample is placed on a glass slide and stained to enhance the structures in the sample as cells, fat, collagen fibres, etc. The most widely used stain is the hematoxylin-eosin (H&E) stain [20]. The sample is then scanned, where the resolution depends on the slide scanner, but is in the size of 0.5 μm/pixel for a $\times 20$ magnification and 0.25 μm/pixel for a $\times 40$ magnification [19].

3.1.1 Preprocessing of Histology WSI

The high resolution of the scanned images poses a practical challenge for the application of deep learning-based models to whole slide histology images [21, 22]. Their large file size prevents an image from being loaded to the graphics processing units (GPUs), making preprocessing a requirement for analysis of the slides [21]. The preprocessing consists of patch extraction (or tiling) where the WSI is divided into smaller subimages [21, 22] with either overlapping pixels between tiles or with no overlap in the division (figure 3.1d). To reduce the variation in colour between slides due to differences in staining between scanning cites, colour normalization is usually applied followed by selection of the area of interest (for example, the area with invasive cancer). To get the deep learning model to generalize well to new, unseen images, the tiles often undergo image augmentation of rotation and flipping (figure 3.1c,e) [21].

The tiling of WSIs adds additional challenges for deep learning applications. Since data annotation is time-consuming, where ground truth labels are manually defined by pathologists, whole slide histological images are often annotated on slide level [22]. This turns the learning objective on tile level into a weakly supervised setting where ground truth annotation is based on the whole slide. On the other hand, the tiling can enable learning objectives on patient-level data, where different clinically relevant regions in the data can be annotated by tile. This facilitates clinical learning outcomes such as detecting crucial areas in the tissue.

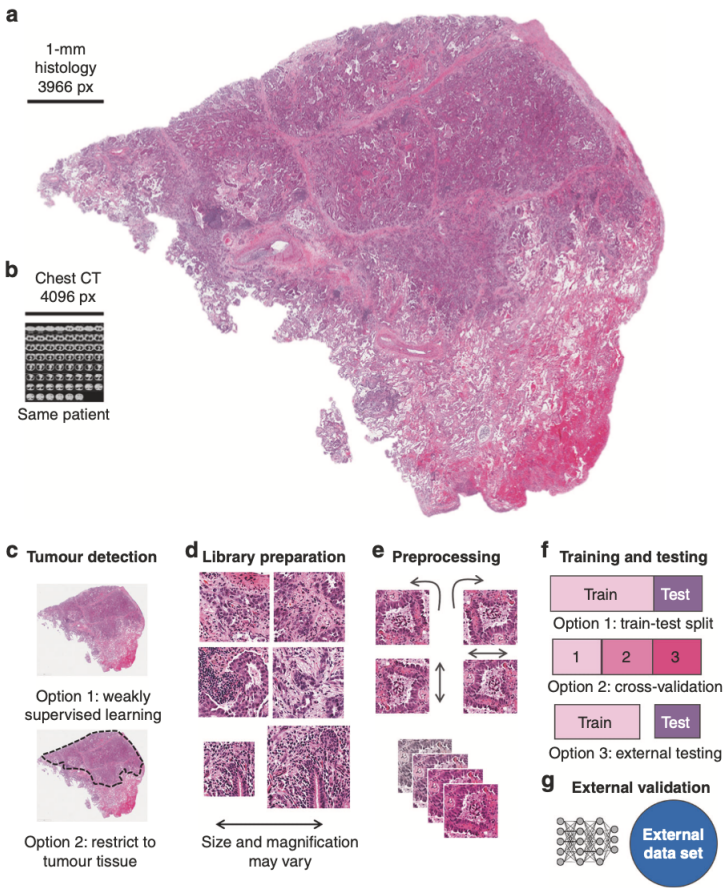


Figure 3.1: Pipeline of deep learning in pathology. **a** Histology image of lung cancer from The Cancer Genome Atlas (TCGA) and The Cancer Imaging Archive (TCIA). **b** Size comparison in pixels for a computed tomography scan of the chest of the same patient. **c** Image processing, either the tissue or tumour is set as the area of interest. **d** Tiling of the image. **e** Preprocessing of tiles, augmentation of the images. **f** Separation of data into training and test set, alternatively using cross validation. **g** An additional external dataset is ideal to evaluate the result of the work. Image from [21], CC BY 4.0.

Chapter 4

Artificial Neural Networks

Artificial Neural Networks, or Neural Networks, are a class of models inspired by the neural network of the human brain to learn patterns inherent in observations given to the system. To process information, neural networks usually use a feedforward pass through the network nodes, which are connected with weights and biases. The most simple learnable neural network is the perceptron, which is a single-layer network consisting of one output unit. Given an observation or input, $\mathbf{v} \in \mathbb{R}^D$ the network gives an output $y(\mathbf{v}; \boldsymbol{\theta})$ from an activation function f of the weighted sum of the inputs as [23]

$$y(\mathbf{v}; \boldsymbol{\theta}) = f(\mathbf{w}^\top \mathbf{v} + w_0). \tag{4.1}$$

In (4.1) $\boldsymbol{\theta} = \{\mathbf{w}, w_0\}$ is the set of parameters, where $\mathbf{w} = [w_i]_{i=1}^D \in \mathbb{R}^D$ is the weight vector between the nodes and w_0 is the bias [23]. The activation function is a non-linear function that adds non-linearity to the model [24]. Two examples of activation functions are the rectified linear unit (ReLU) $f(z) = \max\{0, z\}$, which sets all negative outputs to zero, and the sigmoid function $f(z) = \frac{1}{1+e^z}$. In Figure 4.1 a visualization of a one-layer feedforward neural network with sigmoid activation is presented.

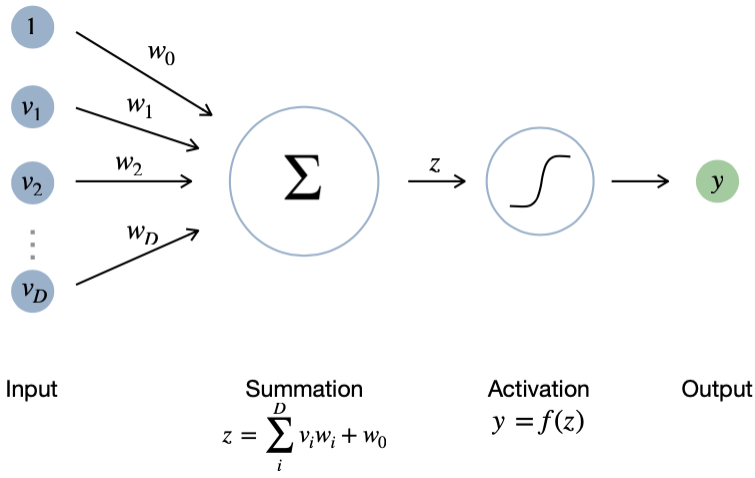


Figure 4.1: Visualization of a single layer feedforward neural network. A weighted summation, with weights $\mathbf{w} = [w_i]_{i=1}^D \in \mathbb{R}^D$ and bias w_0, of the input vector $\mathbf{v} \in \mathbb{R}^D$ is computed as $z = \mathbf{w}^\top \mathbf{v} + w_0$. A sigmoid activation f is then applied to z giving the output $y = f(z)$

This one-layer network is the building block for deeper neural networks with multiple layers and multiple outputs, as visualized in Figure 4.2.

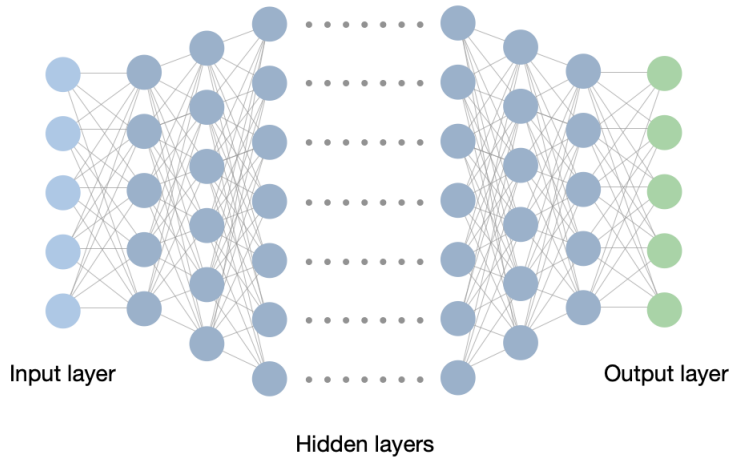


Figure 4.2: Visualization of multiple layer feedforward neural network.

Learning the network parameters is an optimization problem, where the objective is to minimize the error or loss of the model. The loss function is different for different learning tasks, but is a measurement of the difference between the output of the network and the ground-truth output, given the input. For the minimization of the loss function, the gradient of the loss is central. To determine the gradient, back propagation is used, where the idea is to pass the gradient of the loss $\mathcal{L}$, with respect to the weights, from the output layer to the input layer using the chain rule. For a network with L layers, the gradient of the loss function $\mathcal{L}$ with respect to the weights for the l:th layer is given by

$$\frac{\partial \mathcal{L}}{\partial \mathbf{W}^{(l)}} = \frac{\partial \mathcal{L}}{\partial \mathbf{a}^{(L)}} \frac{\partial \mathbf{a}^{(L)}}{\partial \mathbf{a}^{(L-1)}} \cdots \frac{\partial \mathbf{a}^{(l+1)}}{\partial \mathbf{a}^{(l)}} \frac{\partial \mathbf{a}^{(l)}}{\partial \mathbf{z}^{(l)}} \frac{\partial \mathbf{z}^{(l)}}{\partial \mathbf{W}^{(l)}} \tag{4.2}$$

where $\mathbf{z}^{(l)}$ is the preactivation vector of layer l and $\mathbf{a}^{(l)} = f(\mathbf{z}^{(l)})$ is the vector after activation by the activation function f [23]. The optimization algorithm for minimizing the loss function, using back propagation, is handled in Subsection 4.1.4.

4.1 Learning in Deep Neural Networks

The usage of multiple-layer neural networks to learn patterns in the data falls within the category of deep learning. As introduced in Section 3.1, the standard learning procedure is to split the data into training-, validation- and test data. The network is given the training data as input to learn the parameters of the network. The central challenge in training the network is that the model should generalize well to new, unseen data. Hence, the network is evaluated on a separate dataset that is not seen during training: the validation set. This dataset is often part of the training in the way that it is a measure of the model's performance, where the best model-parameters are selected depending on the performance on the validation set. To perform a final evaluation of the trained model, it is applied to the test set. The test set can be an unseen subset of the original dataset, or an external dataset gathered from another data source. [24]

4.1.1 Supervised Learning

When learning is performed on a dataset consisting of examples that are associated with a label or target, it is considered supervised learning [24]. An example is the classification problem where the network is learnt to, given an image of an object, identify which class the object belongs to. Here, the annotation of data is performed by a human, who for each image sets its corresponding label. During training, the loss function is, in this example, based on how well the model predicts the labels for the images. The performance of the model is then evaluated in the validation set, where the validation loss can determine which trained model performed the best.

4.1.2 Unsupervised Learning

In the unsupervised setting, the network is given data without their corresponding target label. From the data, the model is usually sought to learn the probability distribution that generated the data and to find patterns and latent features in the data [24]. The aim can be, for example, data selection, denoising of data or clustering data into groups which share similar features.

4.1.3 The Problem of Overfitting

Overfitting is the problem that occurs when the model is learnt in the way that the optimized parameters overfit toward the training data but do not generalize well to the validation data [24]. Overfitting can be seen in the loss during training, where the training loss, that is, the calculated loss for the model on the training data, is decreasing, while the loss for the validation loss remains high or even increases. The gap between the loss for the training and testing then indicates that the model is learning parameters, which overfits towards the selected training data while it does not generalize well to new, unseen data.

4.1.4 Parameter Optimization

The minimization of the loss function is an optimization problem in which the gradients of the loss function are central [24]. As presented in the beginning of the chapter, the gradient of the loss function $\mathcal{L}$ with respect to the weights $\mathbf{W}$ is calculated as in (4.2) and back propagated through the layers of the network. Being an effective optimization method, nearly all deep learning uses stochastic gradient descent (SGD)

for the optimization objective [24]. The main source of effectiveness of SGD is that it uses an estimated expectation of the gradient based on only a small set of samples, called a mini batch $\mathbb{B} = \{\boldsymbol{x}^{(1)}, \cdots, \boldsymbol{x}^{(b)}\}$. The batch is drawn uniformly from the training dataset $\boldsymbol{x}$, where b is called the batch size. The estimate of the gradient $\boldsymbol{g}$ is based on the gradient of the loss for the mini batch, which is given by the back propagation. The stochastic gradient descent algorithm iteratively follows the gradient downhill to find the parameters $\boldsymbol{\theta}$ which minimizes the loss function. The update of the parameters is given by

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \alpha \boldsymbol{g}$$

where α is called the learning rate [24].

Adam Optimiser

An efficient stochastic optimization method: adaptive moment estimation (Adam) was proposed in [25]. The method, as its name reflects, adapts the learning rate by taking into consideration estimates of the first- and second moments (that is, the mean and uncentered variance) of the gradient. The update of the parameters $\boldsymbol{\theta}_t$ at time step t is carried out as follows:

$$\begin{aligned}
\boldsymbol{m}_t &\leftarrow \beta_1 \cdot \boldsymbol{m}_{t-1} + (1 - \beta_1) \cdot \boldsymbol{g}_t \\
\boldsymbol{v}_t &\leftarrow \beta_2 \cdot \boldsymbol{v}_{t-1} + (1 - \beta_2) \cdot \boldsymbol{g}_t^2 \\
\widehat{\boldsymbol{m}}_t &\leftarrow \boldsymbol{m}_t / (1 - \beta_1^t) \\
\widehat{\boldsymbol{v}}_t &\leftarrow \boldsymbol{v}_t / (1 - \beta_2^t) \\
\boldsymbol{\theta}_t &\leftarrow \boldsymbol{\theta}_{t-1} - \alpha \cdot \widehat{\boldsymbol{m}}_t / (\sqrt{\widehat{\boldsymbol{v}}_t} + \epsilon).
\end{aligned}$$

In the update, $\boldsymbol{m}_t$ is the estimate of the mean and $\boldsymbol{v}_t$ is the estimated variance of the gradients. In the estimates, $\beta_1, \beta_2 \in [0, 1)$ are hyperparameters controlling the decay rate of the moving averages. The moment estimates $\boldsymbol{m}_t$, $\boldsymbol{v}_t$ are initialized with zero-vectors, resulting in a bias towards zero for the moment estimates [25]. This is counteracted by bias-corrected estimates $\widehat{\boldsymbol{m}}_t$ and $\widehat{\boldsymbol{v}}_t$. In the final update of the parameters, α denotes the learning rate and ϵ is a small value added to the denominator to avoid division by zero.

4.2 Convolutional Neural Networks

For data with grid-like topology, as images, convolutional neural networks (CNN) are often used. CNNs are a special neural network that applies the mathematical operation convolution to the data. A kernel, usually a square matrix, with weights is used, describing the transformation of the input to the output. The kernel is typically smaller than the input size, giving the attribute of sparse interactions or spare connectivity where detection of features on parts of the image is enabled. This leads to a reduction of both the memory requirements for the model and, since the parameters are shared for the kernel, also to a reduction of the storage requirements for optimized parameters [24]. The kernel slides over the input data with a stride that describes the translation length to the next step. For each step, a weighted summation is calculated for the input pixels in the location of the kernel. An example of convolution on a 2D input (can be seen as a greyscale image) with a kernel of size 2×2 and with stride 1 is illustrated in Figure 4.3. As can be seen in the illustration, the output size of the convolution will be reduced, depending on the size of the kernel, where the image will shrink with one pixel less than the kernel size. This artefact can be controlled by zero padding the input, where a boarder of zeros is added around the image.

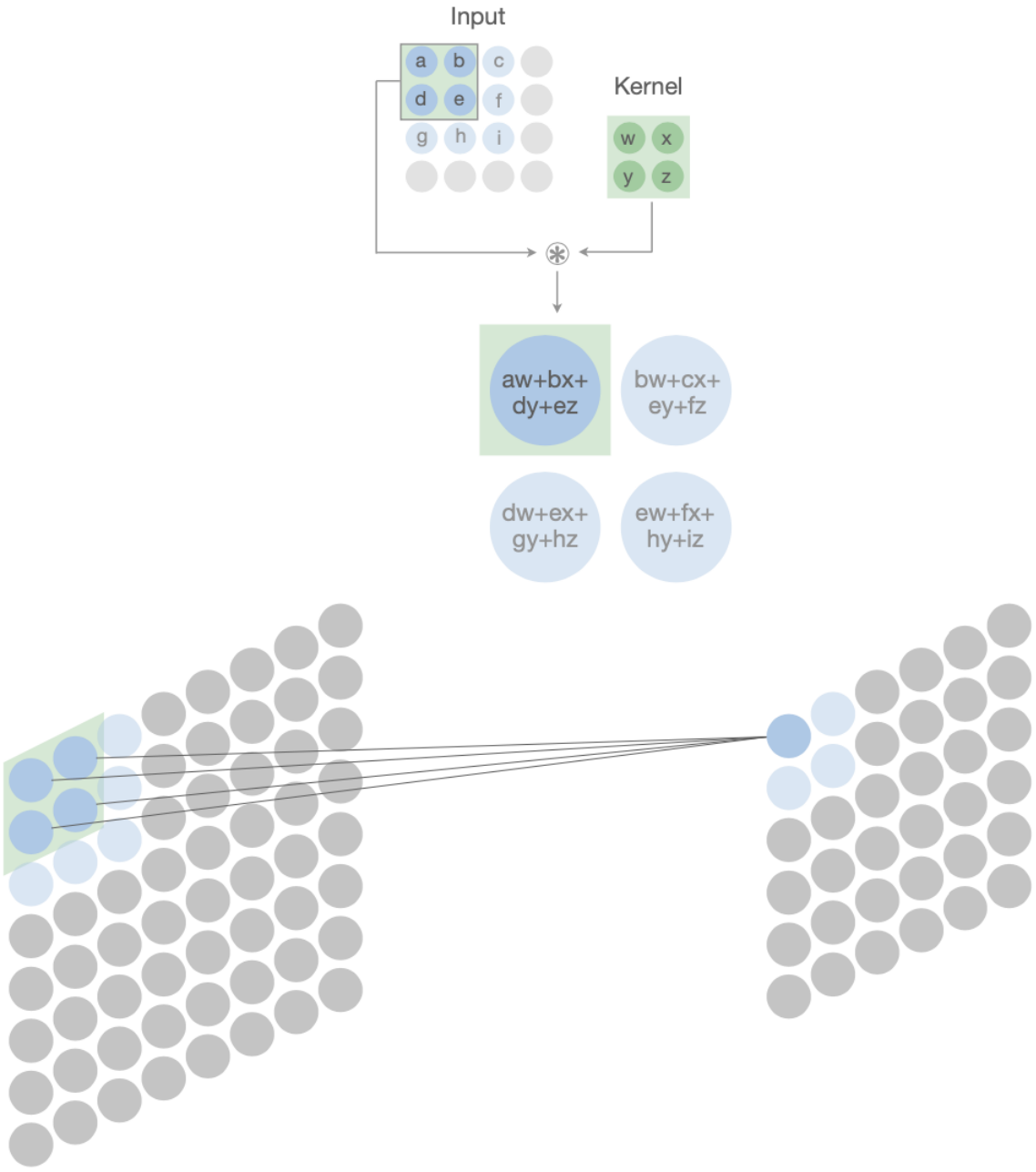


Figure 4.3: An example of 2D convolution with kernel of size 2×2 (marked in green) and stride 1. The formation of the top-left element of the output is marked in blue and computed at the top of the figure.

4.2.1 Pooling

The layers of a CNN typically consist of three steps [24], where in the first stage a set of linear activations is produced by several parallel convolutions. A non-linear activation is then applied on each linear activation in the second stage, such as ReLU or sigmoid activation. In the third and final stage, a further modification of the output is applied by a pooling function.

The pooling function is a form of downsampling in which the nearby outputs are summarized. Some pooling functions are the max pooling operation which takes the maximum output in a rectangular neighbourhood and the average pooling operation which averages over the rectangular neighbourhood. An example of 2D max pooling and 2D average pooling is visualized in Figure 4.4.

Figure 4.4: An example of 2D max pooling, to the left, and 2D average pooling, to the right, on an 4×4 output.

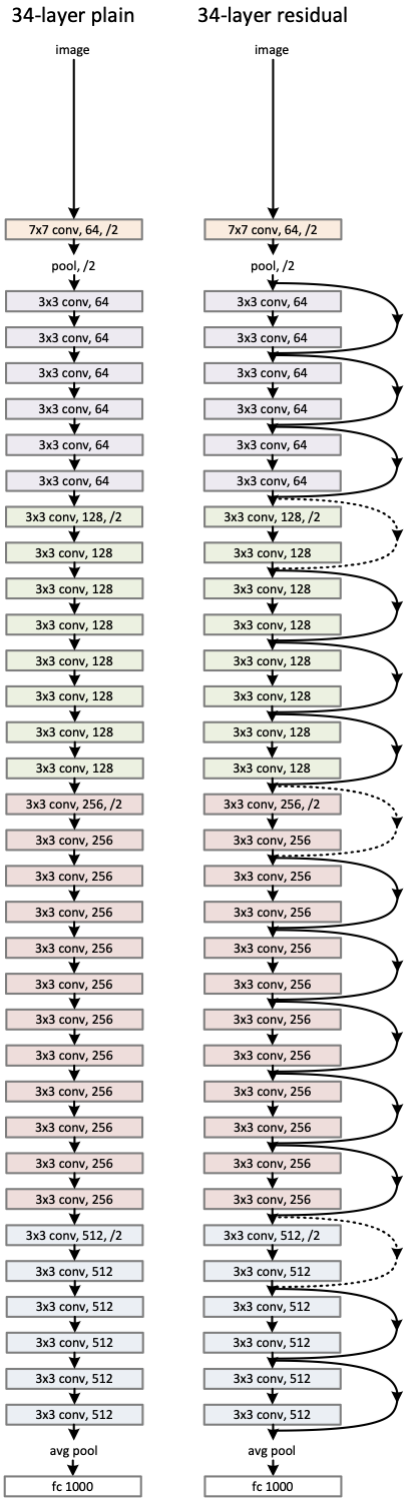


Figure 4.5: Left: 34 layer neural network. Right: 34 layer ResNet. The shortcuts are marked with arrows, and the dotted shortcuts mark an increased dimension. This due to the skipping between convolutional blocks with different number of kernels. Image from [26]. © [2016] IEEE appear prominently with each reprinted figure and/or table.

4.2.2 Residual Neural Networks

Deep CNNs have enabled breakthroughs in areas such as image classification, where the network predicts the class-belonging of an image. The depth of the network, i.e. the number of layers in the network, is of crucial importance for the precision of the model [26]. Although with deeper networks comes the problem of vanishing gradients where the gradients of the loss function goes towards zero during the back propagation through the layers [26]. The vanishing gradients hamper the network to converge towards a minimization of the loss function. The depth of the network also causes a degradation problem in which the accuracy of the model becomes saturated, not due to overfitting, but due to the indication that deeper networks are more complex to optimize [26].

In [26], the degradation problem is addressed by introducing a type of Neural Network called a Residual Neural Network (ResNet). In ResNet, shortcut connections are added to the network so that outputs from the activation in one layer can skip over a number of layers and be added to the weighted sum from a layer deeper in the network. This is illustrated in Figure 4.5.

Chapter 5

Distance Metric Learning

Many areas in computer vision aim at capturing similarities between images, both when it comes to face recognition [27, 28], image retrieval [29, 30] and motion analysis [31]. Distance Metric Learning (DML) is a type of learning approach that relates objects, in this case images, with respect to similarity or dissimilarity based directly on a distance metric. The metric on a set X can be explained by the metric space defined by [32]

Definition 1 (Metric space)**.** A metric space (X, d) consists of a set of data points X and a distance function d which maps $d : X \times X \to \mathbb{R}$. The distance function satisfies the following properties:

1. Non-negativity: $d(x, y) \geq 0$ for every $x, y \in X$
2. Identity of indiscernible: $d(x, y) = 0 \iff$ x=y for every $x, y \in X$
3. Symmetry: $d(x, y) = d(y, x)$ for every $x, y \in X$
4. Triangle inequality: $d(x, y) + d(y, z) \geq d(x, z)$ for every $x, y, z \in X$

Based on the distance function in the metric space, the similarity of images can be learnt. As a branch of machine learning, DML has the purpose of learning relevant distances from a dataset $\mathcal{X} := \{x_1, x_2, \cdots, x_N\}$ on which similarities between, for example, pairs or triplets of data are collected. The similarities can be determined by the sets [33]

$$\begin{aligned} S &= \{(x_i, x_j) \in \mathcal{X} \times \mathcal{X} : x_i \text{ and } x_j \text{ are similar}\} \\ D &= \{(x_i, x_j) \in \mathcal{X} \times \mathcal{X} : x_i \text{ and } x_j \text{ are not similar}\} \\ R &= \{(x_i, x_j, x_l) \in \mathcal{X} \times \mathcal{X} \times \mathcal{X} : x_i \text{ is more similar to } x_j \text{ than to } x_l\}. \end{aligned} \tag{5.1}$$

To measure the similarity between images, the image can be represented in a lower-dimensional metric space. This is often done with a deep convolutional network where the CNN maps, or embeds, an image x_i into a metric space $\Phi = \{\phi(x_i), d\}$ where the embedding is represented by $\phi(x_i)$ on which distances can be measured by d.

Consider the mapping ϕ with $\phi_i := \phi(x_i)$, which maps an image of size $N \times N \times 3$ from the dataset $\mathcal{X} \in \mathbb{R}^{N \times N \times 3}$ to the metric space $\Phi = (X \in \mathbb{R}^M, d)$ as $\phi : \mathbb{R}^{N \times N \times 3} \to \Phi : \mathbb{R}^M$. The learning goal in DML is to find ϕ so that the constraints in (5.1) are fulfilled in the metric space Φ, that is, capturing the similarity between samples in the metric space Φ. Technically, the goal can be seen as a minimization of a loss function $\mathcal{L}$ depending on the distance function d in the metric space as well as the similarity constraints in (5.1) as [33]

$$\min \mathcal{L}(d, S, D, R). \tag{5.2}$$

In (5.2), d is the distance function on the embeddings
$X \in \{\phi(x_1), \phi(x_2), \cdots, \phi(x_k)\}$ of images x_i in the original dataset $\mathcal{X}$.

In the supervised setting, the similarity of the images can be determined by the corresponding labels $y_1, y_2, \cdots, y_k$ for each sample in $\mathcal{X}$. The sets S, D and R can then be defined, depending on the labels as

$$\begin{aligned} S &= \{(x_i, x_j) \in \mathcal{X} \times \mathcal{X} : y_i = y_j\} \\ D &= \{(x_i, x_j) \in \mathcal{X} \times \mathcal{X} : y_i \neq y_j\} \\ R &= \{(x_i, x_j, x_l) \in \mathcal{X} \times \mathcal{X} \times \mathcal{X} : y_i = y_j \neq y_l\}. \end{aligned}$$

The set of images (x_i, x_j, x_l) in R is called a triplet and has been used frequently in the DML area for computer vision [34, 27].

5.1 Triplet Loss

The triplet loss describes the similarity constraints by enforcing pairs of samples with the same label ($(x_i, x_j) \in S$) to have a smaller distance in the metric space than those with different labels ($(x_i, x_j) \in D$). This constraint is fulfilled by looking at triplets of data (x^a, x^p, x^n) where x^a is the anchor image, x^p is a positive pair to the anchor image, i.e. they have the same label ($(x^a, x^p) \in S$) while x^n is a negative image and has a different label from the anchor ($(x^a, x^n) \in D$). The loss to be minimized in (5.2) is then described by

$$\mathcal{L}_{triplet} = \sum_{i}^{k} \left[d(x_i^a, x_i^p) - d(x_i^a, x_i^n) + \alpha \right]_+ \tag{5.3}$$

where α is the enforced margin between positive- and negative pairs. Hence, the wanted relative distances should be learned to satisfy

$$d(x_i^a, x_i^p) + \alpha < d(x_i^a, x_i^n). \tag{5.4}$$

A visualization of DML using the triplet loss is presented in Figure 5.1.

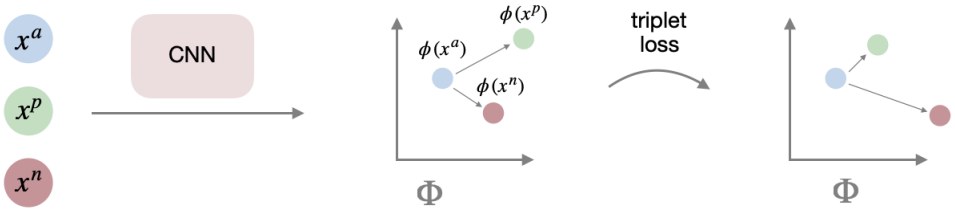


Figure 5.1: The triplet loss minimizes the distance in metric space Φ between anchor and positive samples ($(x^a, x^p) \in S$) and maximizes the distance between anchor and negative ($(x^a, x^n) \in D$). In the figure, the anchor is marked as a blue dot, the positive sample as a green dot and the negative sample as a red dot.

A problem with the triplet loss is that it is discriminative in the way that it does not generalize well to novel classes [34]. This since the network learns to separate only the classes occurring in the training data. A solution to the problem is suggested in [34] where the authors incorporate features that are shared between various classes. The approach is to introduce a negative triplet with every example being from different classes so that the network is forced to additionally take into consideration similarity between samples of different classes. The network is constructed by two embeddings ϕ and ϕ^*, where ϕ maps the images to the metric space Φ and ϕ^* to Φ^*. The mappings are trained so that the metric space Φ explains the discriminative class-separated characteristics of the images, similar to the regular triplet loss. In contrary, Φ^* should capture the shared features between samples of different classes which is learnt by forcing the network to evaluate the similarity between images of different classes [34].

5.2 Multi-Similarity Loss

For pair-based DML methods, where the similarity of pairs, triplets or quadruplets are considered, another key issue is that random sampling of training data can lead to a majority of redundant pairs that does not give additional information to the network [35]. This leads to a reduced convergence rate and model degeneration, which several studies have tried to avoid by improving the sampling strategy of data pairs.

In [35] a general pair-weighting strategy is proposed that generalizes the sampling problem in DML to a weighting of data pairs so that informative pairs are considered during training of metric learning models. In the batch, only hard positive and hard negative samples are considered. This is done by only looking at negative pairs that are more similar than the most dissimilar positive to the anchor point

$$S_{ij}^{-} > \min_{k \in \mathcal{P}_i} S_{ik} - \epsilon \tag{5.5}$$

and the positives that are more dissimilar than the most similar negative pair to the anchor point as

$$S_{ij}^{+} < \max_{k \in \mathcal{N}_i} S_{ik} + \epsilon. \tag{5.6}$$

Here S_{ij} denotes the similarity between the i:th and j:th data points and ϵ is a hyperparameter which sets a margin for the hard mining. The sets $\mathcal{P}_i$ and $\mathcal{N}_i$ represent the positive- and negative pairs to the i:th anchor data point. Together with pair mining, the authors in [35] define the multi-similarity loss as

$$\mathcal{L}_{MS} = \frac{1}{b}\sum_{i=1}^{b}\Big\{\frac{1}{\alpha}\log\big[1 + \sum_{k \in \mathcal{P}_i} e^{-\alpha(S_{ik}-\lambda)}\big] \\ \frac{1}{\beta}\log\big[1 + \sum_{k \in \mathcal{N}_i} e^{\beta(S_{ik}-\lambda)}\big]\Big\} \tag{5.7}$$

where b denotes the batch size, S represents the similarity matrices and α, β, λ are hyperparameters for weighting the negative- and positive parts of the loss as well as marginalize the similarity measurement.

Chapter 6

Previous Work in the Field

The application of DML has been applied to several areas of computer vision, such as face recognition [27, 28], image retrieval [29, 30], and motion analysis [31]. In the following section, the usage of deep metric learning in digital histopathology will be considered.

6.1 Histopathological Image Retrieval

In [36] DML was used for histopathological image retrieval on WSIs of liver cancer. To facilitate pathologist workflow, the proposed method aims to return images similar to a query image given to the network. In the method, the dataset was annotated at the tile level for five biological structures: central vein (CV), intrahepatic bile duct (IBD), interlobular vein (IV), interlobular artery (IA), and sinusoids (Sinus) (see Figure 6.1 for examples of data). The mapping ϕ of the images to the embedding space was performed using a CNN model with a mixed attention mechanism, where attention to the spatial structure was taken into account. To train the model, the multi-similarity loss presented in Section 5.2 was used where the similarity metric between the embeddings $S_{ij} = S(\phi(x_i), \phi(x_j))$ was the cosine similarity as

$$S_{ij} = \frac{\phi(x_i) \cdot \phi(x_j)}{\|\phi(x_i)\|_2 \|\phi(x_j)\|_2}. \tag{6.1}$$

Some successful and unsuccessful retrieval results along with the cosine similarity to the query images are presented in Figure 6.1.

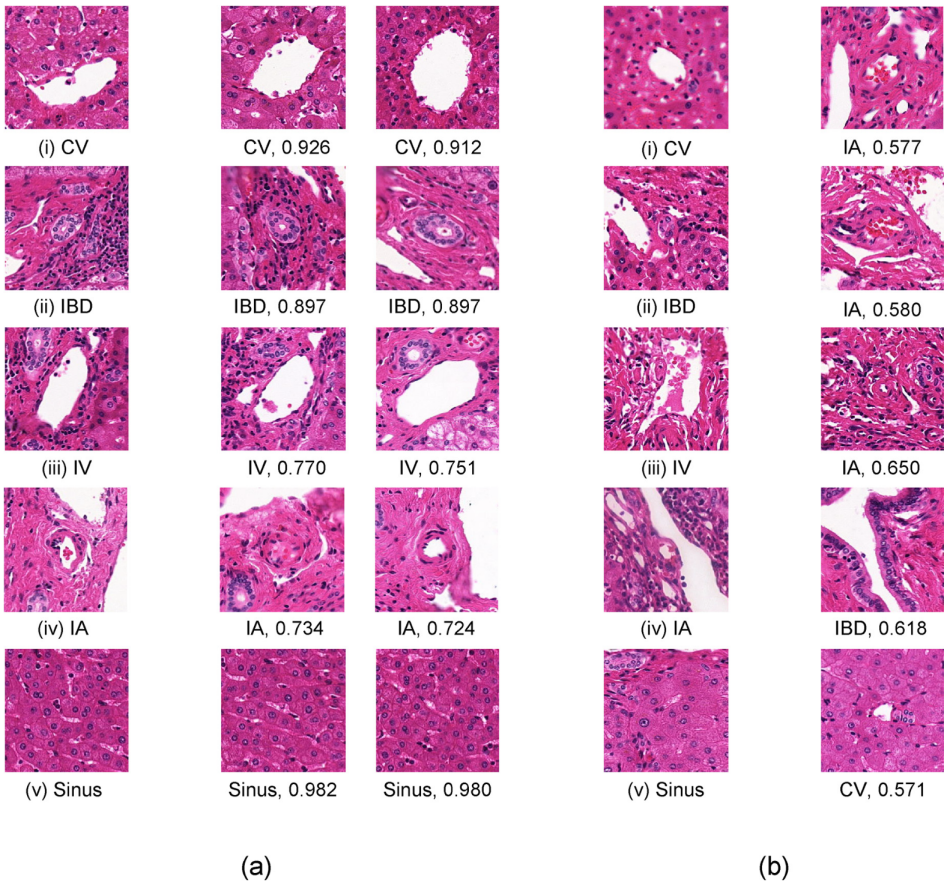


Figure 6.1: Visualization of retrieval results from the proposed method by [36]. (a) Successful retrieval results based on the query image in the first column. The second and third column is the top two most similar images to the query image where the cosine similarity to the query is presented, along with the label, below each image. (b) Failed retrieval result. The second to last column is presenting the query image, and the last column is the correspondingly most similar image. Image from [36], *Copyright 2022 by Elsevier. Reproduced with permission of Elsevier in the format for reuse in a thesis/dissertation via Copyright Clearance Center.*

6.2 Classification of Histopathological Images

Distance metric learning has also been used in digital histopathology for image classification. In [37] a predefined 2D-embedding was used to guide the mapping of human bone marrow microscopy images to a metric space where cell classification was performed. The idea was to leverage expert knowledge of how bone marrow cells are related to each other when measuring the similarity between the images of the cells. Referring back to the loss function (5.2) in DML, the similarity S, dissimilarity D and relative similarity R between images are determined by the embedding guide. A visualization of the 15 different cell types handled in the study, as well as the embedding guide, is shown in Figure 6.2.

The architecture for mapping the images into the embedding space was DenseNet-121, which is a 121 layer convolutional neural network where every layer is connected to every second layer in a feedforward fashion. To train the network, the Adam optimizer was used and the authors investigated different loss functions in the training of the embedding mapping, one of which being the triplet loss presented in Section 5.1. In the investigation of loss functions, the best performing loss was a distance-based loss proposed by the authors:

$$\mathcal{L}_{\text{dist}} = \frac{2}{b(b-1)} \sum_{i=0}^{b-1} \sum_{j=0}^{i-1} L_1(L_1(p_i, p_j), L_1(e_{\text{2D},i}, e_{\text{2D},j})) \tag{6.2}$$

where pairs of samples are learned to have the same relative distance in the embedding guide and in the learned embedding space.

Inspired by the work in [37], the thesis will implement a similar guide embedding to the one described in the study for guiding the mapping of whole slide images of breast cancer. In order for the guided mapping in this thesis to describe relative similarities in the gene expression data, the genes expressed in breast cancer tumours will be the basis of the embedding guide. This, instead of using a manually defined embedding guide as in [37]. Additionally, the distance-based loss proposed in (6.2) will be the ground for network training in the thesis, as described in the following chapter.

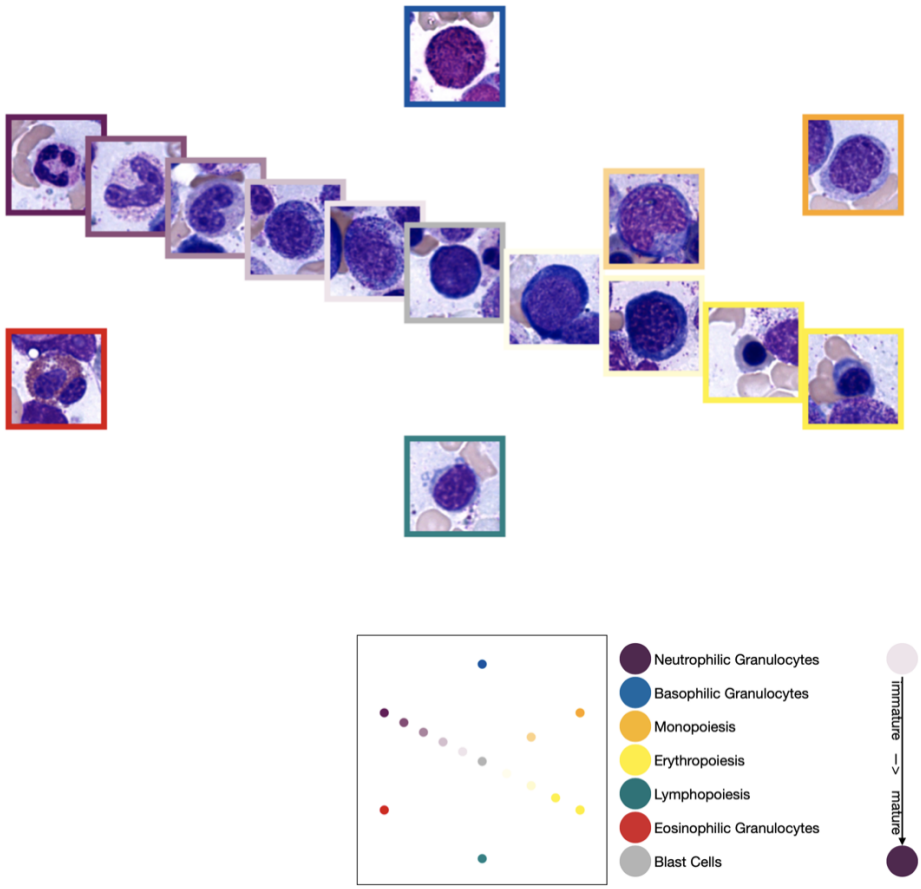


Figure 6.2: Visualization of embedding guide, where examples of the 15 cell classes are depicted to the left. The images are related to one another in accordance to the manually defined embedding guide at the bottom of the figure. The colours of the boarder around the images as well as of the points in the embedding guide are related to the cell types presented to the right of the guide. The maturity level is defined by the colour saturation in the visualization and by distance to the centre of the guide in the metric space. Image adapted from [37], CC BY 4.0

Chapter 7

Materials and Methods

In the following section, the proposed approach to learn the gene expression guided embedding extractor network ϕ will be explained. The embedding extractor aims to map a WSI of a breast cancer tumour into a metric space Φ that captures relative similarities between the gene expression profiles from tumour samples. An overview of the method is presented in Figure 7.1 where the WSI, consisting of tiles, is the input to the embedding extraction network (see a more detailed description of the architecture in Section 7.2). The embedding extractor maps the tiles into embedding vectors in the metric space Φ_{pred}, which should be related in concordance with the corresponding gene expression vectors for each WSI which lies in the metric space Φ_{gene}. In Figure 7.1 the predicted embedding vectors are marked as dots while the gene expressions are marked as squares, the colour of the vectors represents the corresponding WSI. All experiments in the thesis are implemented in Python, and the training of the neural networks is performed in PyTorch.

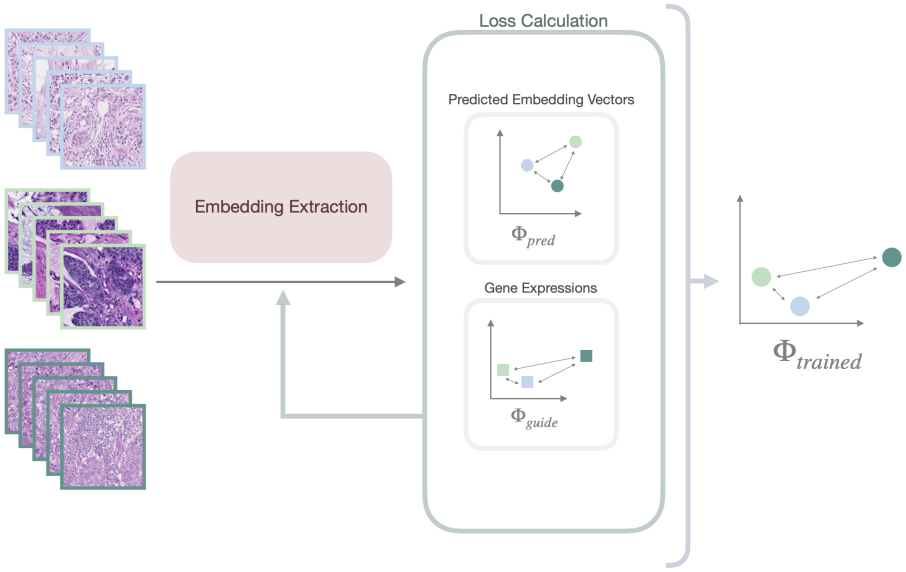


Figure 7.1: Overview of method for training the embedding extraction network. The WSIs, consisting of tiles, are sent as inputs to the embedding extraction network. The mapped embedding vectors are presented as dots in the metric space Φ_{pred} and the corresponding gene expression as squared in Φ_{guide}. The colours represent each WSI.

To guide the network to map embedding vectors that have pairwise distances as in Φ_{guide}, the loss is calculated depending on the difference between the pairwise distances between WSIs in the predicted metric space Φ_{pred} and the gene expression guided embedding space Φ_{guide}. The loss function will be explained in more detail in Section 7.3. After training the network, the goal is that the embedding extraction maps images into embedding vectors in the metric space Φ_{pred} where the embeddings are related as in Φ_{guide}, as visualized to the right in Figure 7.1.

Before giving a more thorough description of the technical implementations of the gene expression guided metric learning method, the data used in the study will be introduced.

7.1 Datasets

Data used for the study were preprocessed and provided by Karolinska Institutet. The study consists of female breast cancer patients from two data sources: Cliseq-BC (272 patients) and The Cancer Genome Atlas (TCGA-BC) (721 patients). The Clinseq-BC data was scanned in-house by Karolinska Institutet with a Hamamatsu Nanozoomer XR at 40X magnification (0.226 μm/pixel) while whole slide images in TCGA-BC data had been downloaded from `https://portal.gdc.cancer.gov` where WSIs with a magnification of 20X were excluded from the study to ensure equal image quality. One WSI was included from each patient with the corresponding RNA-sequence data available. All WSIs were stained with H&E-staining.

The WSIs were preprocessed by Karolinska Institutet where the images underwent quality control and colour normalization. Tissue segmentation was performed for the WSIs and regions of invasive cancer were annotated by pathologists. The WSIs were split into tiles, where only regions representing invasive cancer were considered in the study. The tiles were of size 598×598 pixels and had 50% overlap between neighbouring tiles.

For the gene expression data from RNA-sequencing, the Clinseq-BC data were normalized in the preprocessing so that the gene expression data had the median value of each gene equal to the gene expressions from TCGA-BC. For the study in this thesis, the genes considered were restricted to the PAM50 set, which was the base for the guiding gene expression vector for each WSI.

The data was randomly split into training- and test sets, where the data was split at patient level. The number of WSIs N in each set was $N = 697$ (85.10%, 4.81 million tiles) for the training set and $N = 122$ (14.90%, 0.86 million tiles) for the test set. During the development and training of the model, the test set remained untouched and was only used once at the end of the project for the final evaluation of the model performance. Only the training data were used for hyperparameter tuning and development of the model. During the tuning of the model, the training data was additionally split for 5-fold cross validation where the training data were split into 5 validation folds, of 20% each. The models were trained on 4 of the folds and validated on the last, for the model unseen, fold. During training of the model, the tiles were augmented with random 90° rotations followed by random horizontal- and vertical

flipping of the images. In Figure 7.2 an example of a WSI in the Clinseq training data is presented along with 9 of the tiles from the WSI.

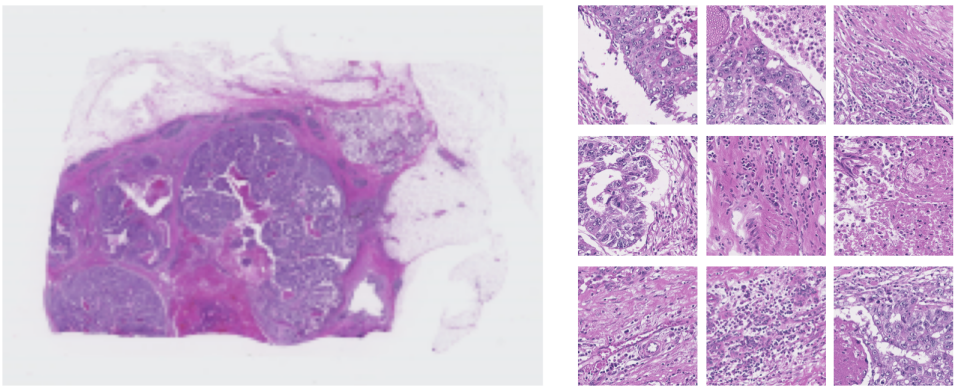


Figure 7.2: Example of an WSI from the Clinseq training data. To the right in the figure 9, examples of tiles in the WSI are presented.

In Figure 7.3 the gene expressions in the training data are presented. In the plot, a linear dimensionality reduction has been performed on the 50-dimensional vectors via principal component analysis so that the gene expression data is plotted in the two most principal components. Each point, representing the gene expressions of the WSIs in the training data, is marked with the sample's intrinsic molecular subtype (Luminal A, Luminal B, Basal-Like, HER2-Enriched, Normal Breast-Like). In the cases where the samples are marked with NaN and coloured light green, there is no annotation of the subtype for the slide. Below the plot of the gene expression embeddings, there are examples of three WSIs from the Clinseq training data, along with 9 tiles from the corresponding slide. Each example is marked with a boarder with colours representing the intrinsic molecular subtypes.

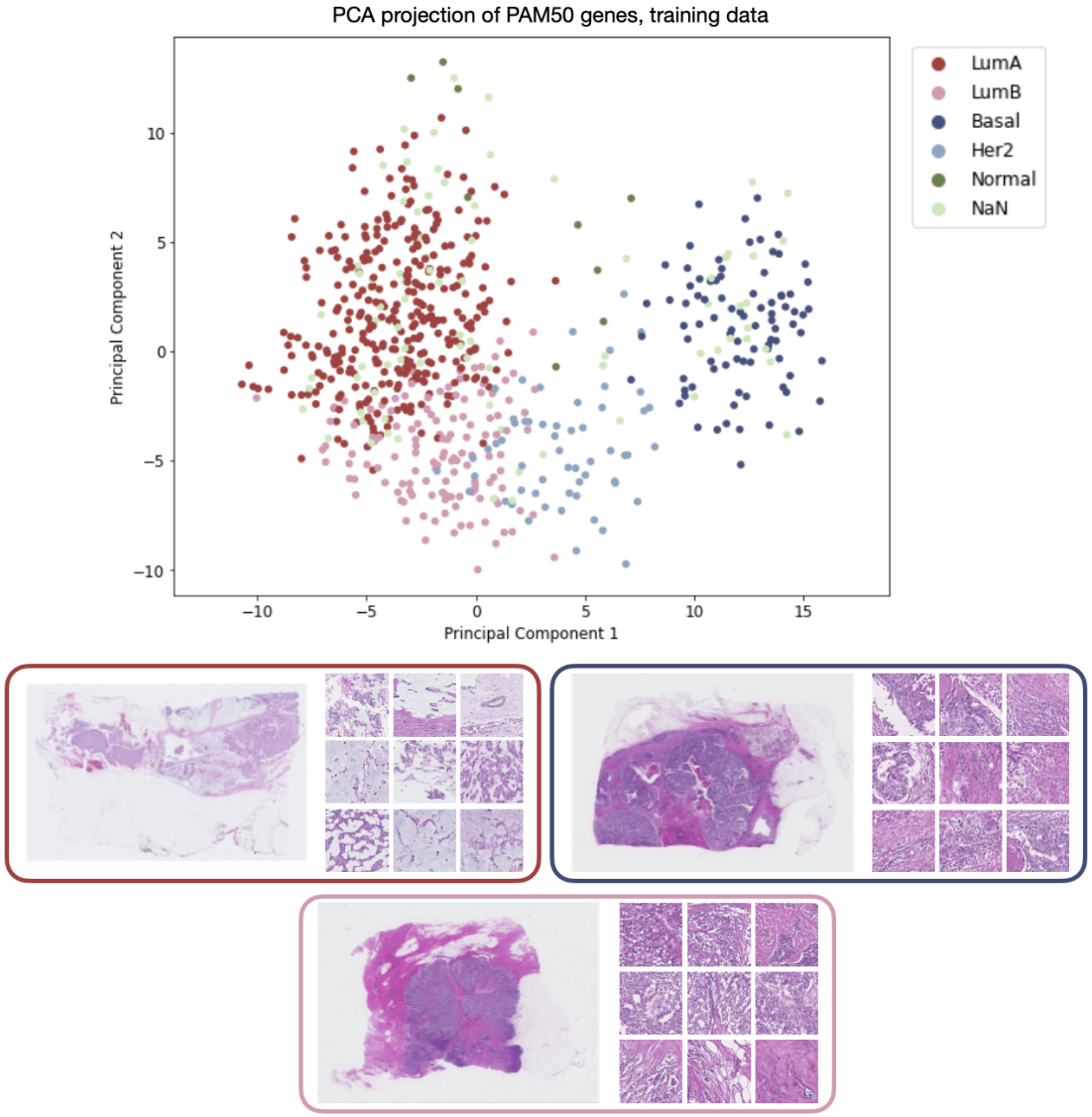


Figure 7.3: Gene expressions from the WSIs in the training data, plotted in the two most principal components of the data. Each point represents an WSI where the colour marks the molecular subtype of the tumour. For the points marked with NaN, the molecular subtype is not annotated.

7.2 Embedding Extraction

Figure 7.4: Overview of embedding extraction in the gene expression guided model. For each WSI x_i there exists a vector of gene expression e_i. The embedding extractor network ResNet50 generates an embedding $\phi(x_i)$ in the metric space Φ_{pred} which, ideally, should incorporate the relative distances between gene expressions in Φ_{guide}.

Consider the WSIs $\mathcal{X} = \{x_1, x_2, \ldots, x_b\}$, in a mini-batch of size b, where each WSI $x_i \in \mathcal{X}$ consists of the n_i image tiles $\boldsymbol{t}^i = \begin{bmatrix} t_1^i & t_2^i & \cdots & t_{n_i}^i \end{bmatrix}^\top \in \mathcal{T}$. The production of the embedding vectors is done by the embedding extractor ϕ, which maps the tiles in the mini-batch to vector representations of the images. In Figure 7.4 an overview of the embedding extraction for the model is presented where the batch size is set to $b = 3$. The embedding extractor ϕ is a deep neural network mapping WSIs $x_i \in \mathcal{X}$ to the M-dimensional embedding vectors X in the metric space $\Phi_{pred}(X \in \mathbb{R}^M, d)$ as $\phi : x \longrightarrow \Phi_{pred} : \mathbb{R}^M$ where

$$X = \{\phi(x_1), \phi(x_2), \ldots, \phi(x_b)\} = \{\phi(\boldsymbol{t}^1), \phi(\boldsymbol{t}^2), \ldots, \phi(\boldsymbol{t}^b)\} = \\ = \left\{ \begin{bmatrix} \phi(t_1^1) \\ \phi(t_2^1) \\ \vdots \\ \phi(t_{n_1}^1) \end{bmatrix}, \begin{bmatrix} \phi(t_1^2) \\ \phi(t_2^2) \\ \vdots \\ \phi(t_{n_2}^2) \end{bmatrix}, \ldots, \begin{bmatrix} \phi(t_1^b) \\ \phi(t_2^b) \\ \vdots \\ \phi(t_{n_b}^b) \end{bmatrix} \right\}.$$

In the thesis, the metric spaces $\Phi_{pred}(X \in \mathbb{R}^M, d)$ and $\Phi_{guide}(E \in \mathbb{R}^{50}, d)$ were considered. The distance function d was the same for the two metric spaces and had the purpose, alike mentioned in Chapter 5, to measure the similarity between the embeddings. The different distance functions, as well as the different dimensionalities, M, of Φ_{pred} studied in the thesis, are presented in Section 7.4. The dimension of the guide metric space Φ_{guide} was 50, since the gene expression vectors were restricted to the PAM50 genes. The data points E in Φ_{guide} are the gene expressions for each WSI as $E = \{e_1, e_2, \ldots, e_b\}$. In Figure 7.4 the guiding gene expression for WSIs in the batch is presented in Φ_{guide} together with an example of the predicted embeddings from the embedding extraction network in Φ_{pred}. In the figure, only one embedding vector, i.e. one mapped tile from the WSI, is plotted in Φ_{pred}.

7.2.1 Network Architecture

The embedding network, extracting the embedding vectors from the tiles, was the ResNet50 model, which is the 50-layer Residual Neural Network introduced in Subsection 4.2.2. The parameters of the network were initially set to the parameters pre-trained on the public dataset ImageNet, which is a dataset consists of more than 14 million images containing more than 20 thousand categories of images. The last fully connected layer, after the average pooling in Figure 4.5, was replaced

with a dropout layer that randomly sets the elements of the input tensor to the layer to zero with a probability p that was set to 0.5 in all experiments. The dropout layer was added for regularization purposes and reduces the overfitting of the model towards the training data. The dropout layer was followed by a fully connected layer with the output size set to the embedding size of the image embeddings, that is, the dimension of the metric space M.

7.2.2 Optimization Method

For training the embedding extraction network, the Adam optimizer, introduced in Subsection 4.1.4, was used. The hyperparameters $\beta_1, \beta_2, \epsilon$ were set to the default values: $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$ as suggested in [25]. The learning rate α was set, depending on the distance function used in the models. In the early stages of the thesis, a grid search over the values $\alpha = \begin{bmatrix} 10^{-3} & 10^{-4} & 10^{-5} \end{bmatrix}$ was performed, resulting in that $\alpha = 10^{-3}$ was excluded due to unstable validation loss in the training. The learning rates $\alpha = 10^{-4}, 10^{-5}$ did not show great differences in the validation loss in the early stages, based on this, the models were first trained with the learning rate $\alpha = 10^{-4}$. In the case where the validation loss was unstable or did not decrease with learning $\alpha = 10^{-4}$, a grid search was performed on $\alpha = \begin{bmatrix} 10^{-4} & 10^{-5} \end{bmatrix}$ and the learning rate with the lowest- and smoothest validation loss curves was chosen. This was the case for the distance functions d_{L1} (Figure A.3 in Appendix A) and d_{L2} (Figure A.4 in Appendix A) where the learning rate was chosen as $\alpha = 10^{-5}$.

7.2.3 Weakly Supervised Training

Since the embedding network extracts image embeddings from the tiles, and the gene expression data is given on WSI-level, the training of the network is weakly supervised. By relating each tile to the gene expression of the full WSI we assume that the gene expression is homogeneous in the sample, so that each part of the tissue is expressing the same amount of the genes. This is a simplification, especially due to the known heterogeneity in breast cancer tumours. Some attempts to reduce this simplification were done in the thesis, where, for example, several tiles from the same WSI were considered in the calculation of the loss.

7.3 Loss function

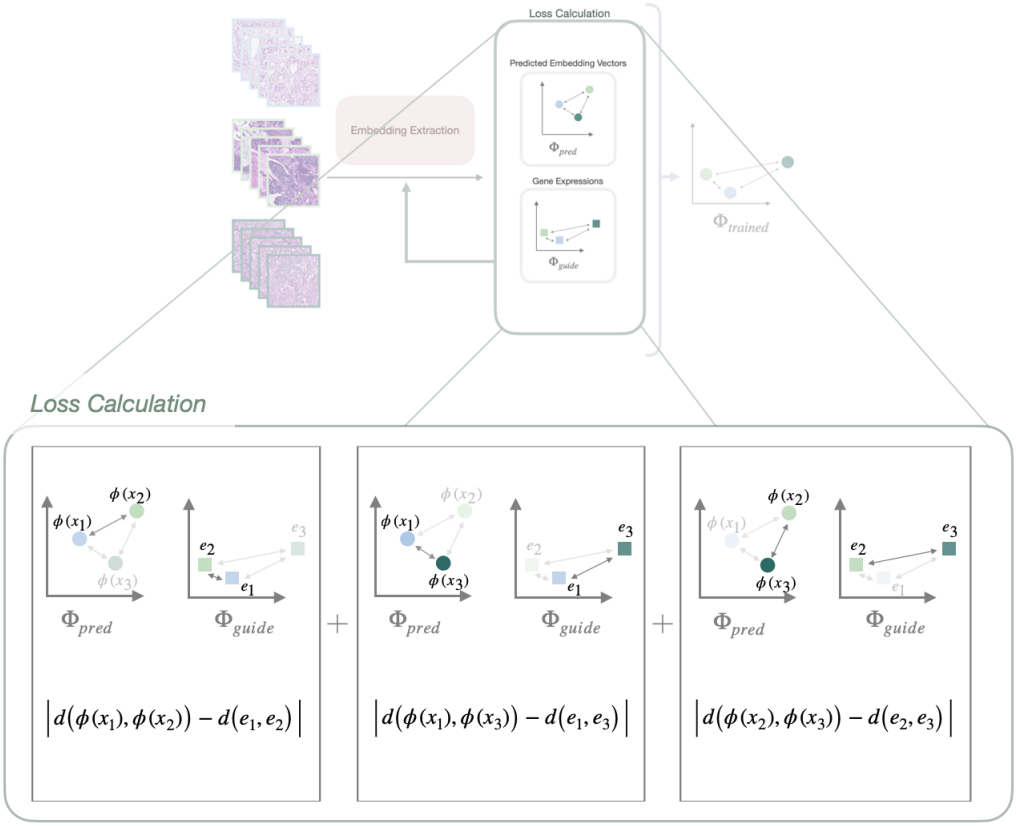


Figure 7.5: Overview of loss calculation in the gene expression guided model. In the loss calculation, the distance between each pair of predicted embeddings in the batch is calculated and compared to the distance between the two corresponding gene expressions from the slides. In the figure, the batch size is set to 3 for visualization purposes.

To measure the difference between the relative similarities in the predicted embeddings and in the guiding gene expression vectors, the distance loss proposed by [37] was implemented as

$$\mathcal{L} = \frac{2}{b(b-1)} \sum_{i=1}^{b} \sum_{j=1}^{i-1} L_1(d(\phi(x_i), \phi(x_j)), d(e_i, e_j)) = \tag{7.1}$$

$$= \frac{2}{b(b-1)} \sum_{i=1}^{b} \sum_{j=1}^{i-1} \Big| d(\phi(x_i), \phi(x_j)) - d(e_i, e_j) \Big|. \tag{7.2}$$

In the loss function, each pair of predicted- and guiding embeddings (i.e. the gene expression vectors) in the batch of size b are considered. For each pair in the batch, the loss is calculated as the mean absolute difference between the relative pairwise distances between the predicted embeddings of two slides in the batch and the distance between the corresponding gene expressions for the slides. This is done with the idea of minimizing the difference between pairwise distances in the predicted metric space Φ_{pred} and the guiding metric space Φ_{guide}. A visualization of the loss calculation in a batch with batch size $b = 3$ is presented in Figure 7.5 (the batch size is larger in the network implementations, see Section 7.4, but is set to 3 in the figure for visualization purposes). In the figure, it can be seen that the distance between each pair of predictions in the batch is calculated with the distance function d and compared to the distance between the two gene expressions that belong to the corresponding slide.

To measure the distance between image embeddings from two slides $d(\phi(x_i), \phi(x_j))$, two approaches were considered:

(i) Each distance computation $d(\phi(x_i), \phi(x_j))$ was based on one tile, randomly drawn, from each slide: $d(\phi(x_i), \phi(x_j)) = d(\phi(t^i), \phi(t^j))$. This approach will be referred to as Single Tile (ST) distance computation.

(ii) Each distance computation $d(\phi(x_i), \phi(x_j))$ was based on four tiles, randomly drawn from the number of tiles n_i and n_j of each slide. The distances between the slides were then calculated as the average distance between the tiles of the slides as:

$$d(\phi(x_i), \phi(x_j)) = \frac{1}{16} \sum_{k \in K} \sum_{l \in L} d(t_k^i, t_l^j); \quad \begin{cases} K \sim \mathcal{U}_{[1,n_i]}, & |K| = 4 \\ L \sim \mathcal{U}_{[1,n_j]}, & |L| = 4 \end{cases}$$

where K and L are sets of 4 randomly drawn indices from the uniform distributions $\mathcal{U}_{[1,n_i]}$, $\mathcal{U}_{[1,n_j]}$ respectively. This approach will be referred to as Multi Tile (MT) distance computation.

7.4 Experimental Setup

In the experiments of the thesis, the dimension M of the predicted metric space Φ_{pred} was set to $M = 128$. The batch size b, determining the number of tiles sent to the network for each embedding mapping and distance calculation, was set depending on the choice of distance calculation between slides:

- Single Tile **ST:** The batch size was set to $b = 32$ where each tile came from different slides.
- Multi Tile **MT:** The batch size was set to $b = 14 * 4 = 56$ where four different tiles were randomly selected from 14 different slides.

The choice of hyperparameters M and b was based on a grid search on each hyperparameter, where the best parameter was set as the one that gave the lowest validation loss during training. For the dimension of Φ_{pred}, the parameters looked at were $M = [128, 256, 512, 1024]$ where the resulting train- and validation loss can be seen in Figure A.1 in Appendix A. In the figure it can be seen that $M = 128$ and $M = 256$ gave similar validation loss curves. The choice of dimension $M = 128$ was based on that a less complex representation of the extracted embeddings was less prone to overfit towards the training data. For the batch size for single tile distance calculation, the sizes looked at were $b = [16, 32, 64]$. In Figure A.2 in Appendix A, the training- and validation loss for the grid search are presented, where the batch size $b = 32$ was chosen based on that it had the smallest validation loss. The batch size for MT (14×4) was set so that as many slides as possible would fit the memory of the GPU. The grid searches presented above were done in an early stage of the L1 model, where the single tile L1 distance measurement was considered together with a log-transformation of the gene expression data. The logarithmic transformation was not performed for the models presented in the thesis, since it did not improve the L1 model. Therefore, it will not be further considered nor studied in the thesis.

For each epoch in the training of the network, the number of slides looked at was set to 10,000 where the slides in the batches were randomly selected so that the same slide never appeared more than once in

each batch. The reason for looking at the same slides several times in the epoch was so that the network would handle several combinations of pairs of the training slides in each epoch. For validation after each epoch, the number of slides was set to 4,000. During training with 5-fold cross validation, using the training data, the maximum number of epochs was set to 250 and early stopping of the training occurred if the validation loss had not been improved in 50 epochs. For the training of the final model (the Multi Tile Mean Absolute Distance (MT_MAD) model), trained on the full training data, the number of epochs was set to 150. This was based on that the validation loss for the final model had reached stationarity in 150 epochs in the loss curves for the 5-fold cross validation (plotted in A.9 Appendix A).

In the thesis the distance functions d considered for comparison between ST- and MT distance computation were the L1-distance (referred to as d_{L1}), based on the work by [37] in Section 6.2, and the cosine distance (referred to as d_{CL} for Cosine Loss), based on the work by [36] in Section 6.1. The distances are defined as

$$d_{L_1}(\boldsymbol{p}, \boldsymbol{q}) = \sum_{i=1}^{M} |p_i - q_i| \tag{7.3}$$

$$d_{CL}(\boldsymbol{p}, \boldsymbol{q}) = 1 - \frac{\boldsymbol{p} \cdot \boldsymbol{q}}{\|\boldsymbol{p}\|_2 \|\boldsymbol{q}\|_2} \tag{7.4}$$

where M is the length of the vectors $\boldsymbol{p}, \boldsymbol{q}$. One thing that can be noted is that, since the loss is determined based on the absolute difference between distances in Φ_{pred} and Φ_{guide}. The loss function in (7.2), when considering the cosine distance, will be the same as measuring the cosine similarity S in (6.1) between the embeddings since

$$\begin{aligned}
\mathcal{L}_{CL} &= \frac{2}{b(b-1)} \sum_{i=1}^{b} \sum_{j=1}^{i-1} \Big| d_{CL}(\phi(x_i), \phi(x_j)) - d_{CL}(e_i, e_j) \Big| = \\
&= \frac{2}{b(b-1)} \sum_{i=1}^{b} \sum_{j=1}^{i-1} \Big| \big(1 - \frac{\phi(x_i) \cdot \phi(x_j)}{\|\phi(x_i)\|_2 \|\phi(x_j)\|_2}\big) - \big(1 - \frac{e_i \cdot e_j}{\|e_i\|_2 \|e_j\|_2}\big) \Big| = \\
&= \frac{2}{b(b-1)} \sum_{i=1}^{b} \sum_{j=1}^{i-1} \Big| \frac{\phi(x_i) \cdot \phi(x_j)}{\|\phi(x_i)\|_2 \|\phi(x_j)\|_2} - \frac{e_i \cdot e_j}{\|e_i\|_2 \|e_j\|_2} \Big| = \\
&= \frac{2}{b(b-1)} \sum_{i=1}^{b} \sum_{j=1}^{i-1} \Big| S(\phi(x_i), \phi(x_j)) - S(e_i, e_j) \Big|.
\end{aligned}$$

A second aspect to keep in mind is that the distance function d_{CL} does not satisfy the triangle inequality and hence is not a proper distance metric. Therefore, the loss function, based on d_{CL}, should be considered a way to evaluate the relative similarities between the pairwise predicted embeddings and the guiding embeddings rather than a proper distance measurement.

7.4.1 Additional Distance Functions for MT

After comparison between the Single Tile and Multi Tile distance computations, the Multi Tile case was further studied and two additional distance measurements were examined: the Euclidean distance d_{L_2} and the mean absolute distance d_{MAD}, which is the average d_{L_1} over the embedding size.

$$d_{L_2}(\boldsymbol{p}, \boldsymbol{q}) = \sqrt{\sum_{i=1}^{M} (p_i - q_i)^2} \tag{7.5}$$

$$d_{MAD}(\boldsymbol{p}, \boldsymbol{q}) = \frac{1}{M} \sum_{i=1}^{M} |p_i - q_i| \tag{7.6}$$

7.5 Evaluation

To evaluate the model performance, the distances between slides in Φ_{pred} and Φ_{guide} were compared. Distances in Φ_{pred} are the predicted distances between image-embeddings mapped by the embedding extraction network ϕ. Since, again, each slide consists of smaller tiles, the calculation of distances between slides in Φ_{pred} was calculated as the mean distance between s randomly selected tiles from each slide. Since the high number of tiles from each slide contributes to a high computational time, the number of tiles that was taken into consideration in each distance calculation was set to $s = 500$.

7.5.1 Distance Matrix Evaluation

To compare the relative distances between slides in Φ_{pred} and Φ_{guide}, the distance matrices were studied. The distance matrices D_{pred} and D_{guide} were calculated so that

$$D_{pred}[i][j] = d(\phi(x_i), \phi(x_j))$$
$$D_{guide}[i][j] = d(e_i, e_j).$$

Since it holds that $d(x_i, x_j) = d(x_j, x_i)$ for all distance functions considered, the distance matrices will be symmetrical.

For a qualitative evaluation of the distance between predicted embeddings, the distance matrices were sorted based on hierarchical clustering of the distances. This was done with the Python library *fast cluster*, which clusters the data based on the distances between the data given by the distance matrix [38].

To obtain a quantitative measurement, the root mean squared error (RMSE) was calculated between each distance in D_{pred} and D_{guide} as

$$\text{RMSE}(D_{pred}, D_{guide}) = \sqrt{\frac{1}{N}\sum_{i}^{N}\Big(L_{pred}(i) - L_{guide}(i)\Big)^2} \tag{7.7}$$

where L_{pred} and L_{guide} represents the N number of lower triangular elements in the respective distance matrices.

To allow comparison between RMSEs from models with different distance measurements, the distance matrices in the computation of the RMSE were normalized so that the mean value was 1 and standard deviation 1. This is achieved by normalizing D_{pred} and D_{guide} as

$$D_{\text{normalilzed}} = \frac{D - \text{Mean}[D]}{\sqrt{\text{Var}[D]}} + 1$$

where $\text{Var}[D]$ is the variance- and $\text{Mean}[D]$ is the mean of the distances in each distance matrix.

7.5.2 Correlation Evaluation

The correlation between pairwise distances was evaluated by studying the Pearson correlation coefficient, which is a measurement of the linear correlation between data. The correlation coefficient between two variables X, Y is defined as [39]

$$\rho[X, Y] = \frac{\text{Cov}[X, Y]}{\sqrt{\text{Var}[X]\text{Var}[Y]}}, \tag{7.8}$$

where $\text{Cov}[X, Y]$ is the covariance between the variables and $\text{Var}[\cdot]$ denotes the variance. To evaluate if the correlation between the predicted distances and target distances were statistically significant, a statistical test called the Mantel test was conducted. The idea is to randomly permute distances from the distance matrices and compute the distribution of covariance of the shuffled data. The permutation was done in 100,000 iterations, and the mean and standard deviation of the distribution from the Mantel test were used to evaluate the significance of the correlation.

Chapter 8

Results

In the following section, the results from the experiments conducted in the thesis will be presented. Beginning with Section 8.1, the results from the experiments with the different distance functions will be presented along with the study of Single- or Multi Tile distance computation. These results are presented for the 5 folds in the cross validation and are underlying the selection of the final model (Multi Tile with Mean-Absolute-Distance (MT_MAD)) which was evaluated on the test data. The results of the final model's performance on the test data are presented in Section 8.2.

8.1 Model Selection

Beginning with the selection of which metric to be used for the distance measurement in the embedding extractor network, both the different distance functions d_{L1} in equation (7.3) and d_{CL} in (7.4) were studied as well as the different distance calculations based on one Single Tile (ST) from each slide or four different tiles from each slide: Multi Tile (MT). Each of the four models were trained with 5-fold cross validation, and the resulting root mean squared error (RMSE) of the difference in pairwise normalized distances between predicted embeddings for the slides and the corresponding normalized distance in the gene expression data is presented in the box plot in Figure 8.1. In the figure the RMSE is plotted for each fold for the models: distance measured in d_{L1} with ST (L1), distance measured in d_{L1} with MT (MT_L1), distance measured in d_{CL} with ST (CL) and distance measured in d_{CL} with MT (MT_CL). In the box plot, the median is represented by an orange horizontal line, the box extends from the lower- (Q1) to upper quartile values (Q3) of the data from the folds and the whiskers represents the upper- and lower

quartiles $\pm 1.5 \cdot \text{IQR}$ where IQR stands for the interquartile range (Q3-Q1). The mean of the RMSE for each model, along with the standard deviation, is also presented in Table 8.1.

From Table 8.1 and the box plot in Figure 8.1 it can be seen that MT_L1 is the model with the smallest mean RMSE (0.8656) over the five folds and is also the model which gives the smallest maximal and minimal RMSE for the folds. The biggest RMSE was given by CL, both concerning mean (0.9732), median (0.9540) and maximum value (1.0766). It can be seen in Figure 8.1 that the RMSE over the folds are generally higher in the ST models than in the MT models, where the latter has the lowest maximal- and minimal RMSE between the respective distance measurements. For both distance functions, the distance computations using MT instead of ST also results in lower mean RMSE. The root mean squared error should be considered in relation to the range of the distance distribution between slides in the gene expression data plotted for distances, measured with d_{L1} and d_{CL}, in Figure 8.3. In the figure the predicted distances, from 5-fold cross validation of the training data, are presented. By comparing the distribution of predicted distances and guiding gene expression distances in the histograms, it can be seen that the predictions are more similar to the distance distribution in the gene expression data when the multi tile models are used. This suggests that the multi tile models are performing better than the single tile calculations, which is in accordance with the result from the RMSE.

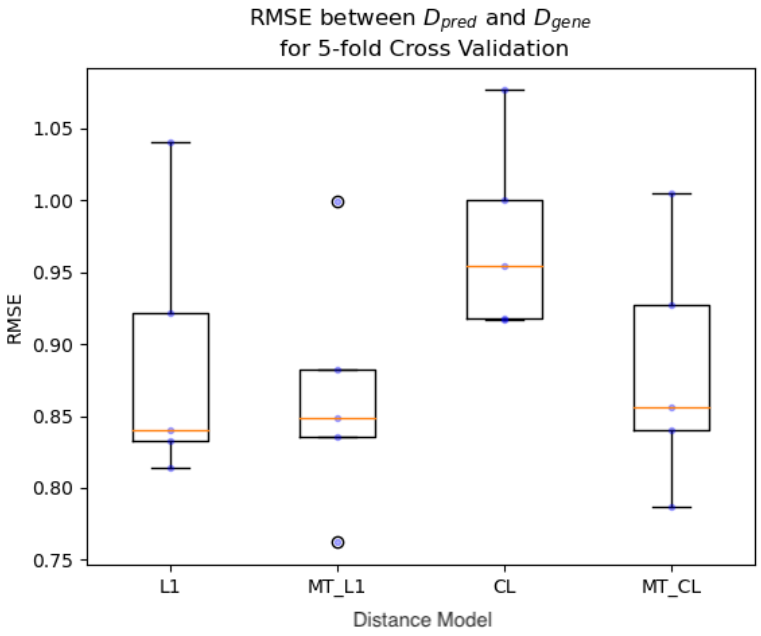


Figure 8.1: Box plot of Root Mean Squared Error (RMSE) for pairwise distances between slide embeddings extracted from the WSIs in D_{pred} and the corresponding distances in the gene expression data D_{gene}. The error is presented for the four models: Single Tile computation with d_{L1} (L1), Multi Tile computation with d_{L1} (MT_L1), Single Tile computation with d_{CL} (CL), Multi Tile computation with d_{CL} (MT_CL) where each model is evaluated with 5-fold cross validation.

For each of the models, the correlation coefficient ρ was calculated between the distances in D_{pred} and D_{gene}. The result is presented in Figure 8.2 for each of the 5 folds in the data. In the box plot, it can be seen that the highest median ρ over the folds is given by the L1-model. The highest minimal- and maximal ρ over the folds is achieved when using the Multi Tile distance measurement, where MT_L1 has the highest both minimal- and maximal correlation coefficient. The mean and standard deviation of the correlation coefficient for the folds are presented in Table 8.1 where it can be seen that the highest mean of ρ is given by MT_L1 (0.6135) followed by MT_CL (0.6018). The smallest mean of ρ is given by CL (0.5157) and the second-smallest mean is given by L1 (0.5903). In Table 8.1 it can be seen that the biggest standard deviation is given by the models using the distance measurement d_{L1} while d_{CL} give lower standard deviation both in ρ and RMSE.

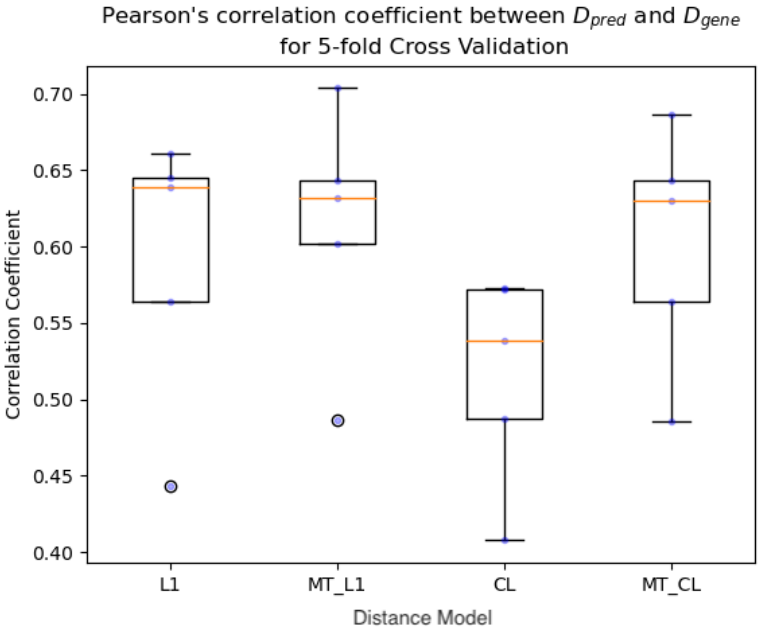


Figure 8.2: Box plot of correlation coefficient ρ for pairwise distances between slide embeddings extracted from the WSIs in D_{pred} and the corresponding distances in the gene expression data D_{gene}. ρ is plotted for the four models: Single Tile computation with d_{L1} (L1), Multi Tile computation with d_{L1} (MT_L1), Single Tile computation with d_{CL} (CL), Multi Tile computation with d_{CL} (MT_CL) where each model is evaluated with 5-fold cross validation.

Table 8.1: Mean (Standard Deviation) of the Root Mean Squared Error (RMSE) and correlation coefficient ρ for the four models: Single Tile computation with d_{L1} (L1), Multi Tile computation with d_{L1} (MT_L1), Single Tile computation with d_{CL} (CL), Multi Tile computation with d_{CL} (MT_CL), evaluated on 5-fold cross validation.

Model	**RMSE**	$\boldsymbol{\rho}$
L1	0.8897(0.08387)	0.59035 (0.08086)
MT_L1	**0.8656 (0.07757)**	**0.6135 (0.07167)**
CL	0.9732 (0.06001)	0.5157 (0.06239)
MT_CL	0.8830 (0.07587)	0.6018 (0.07004)

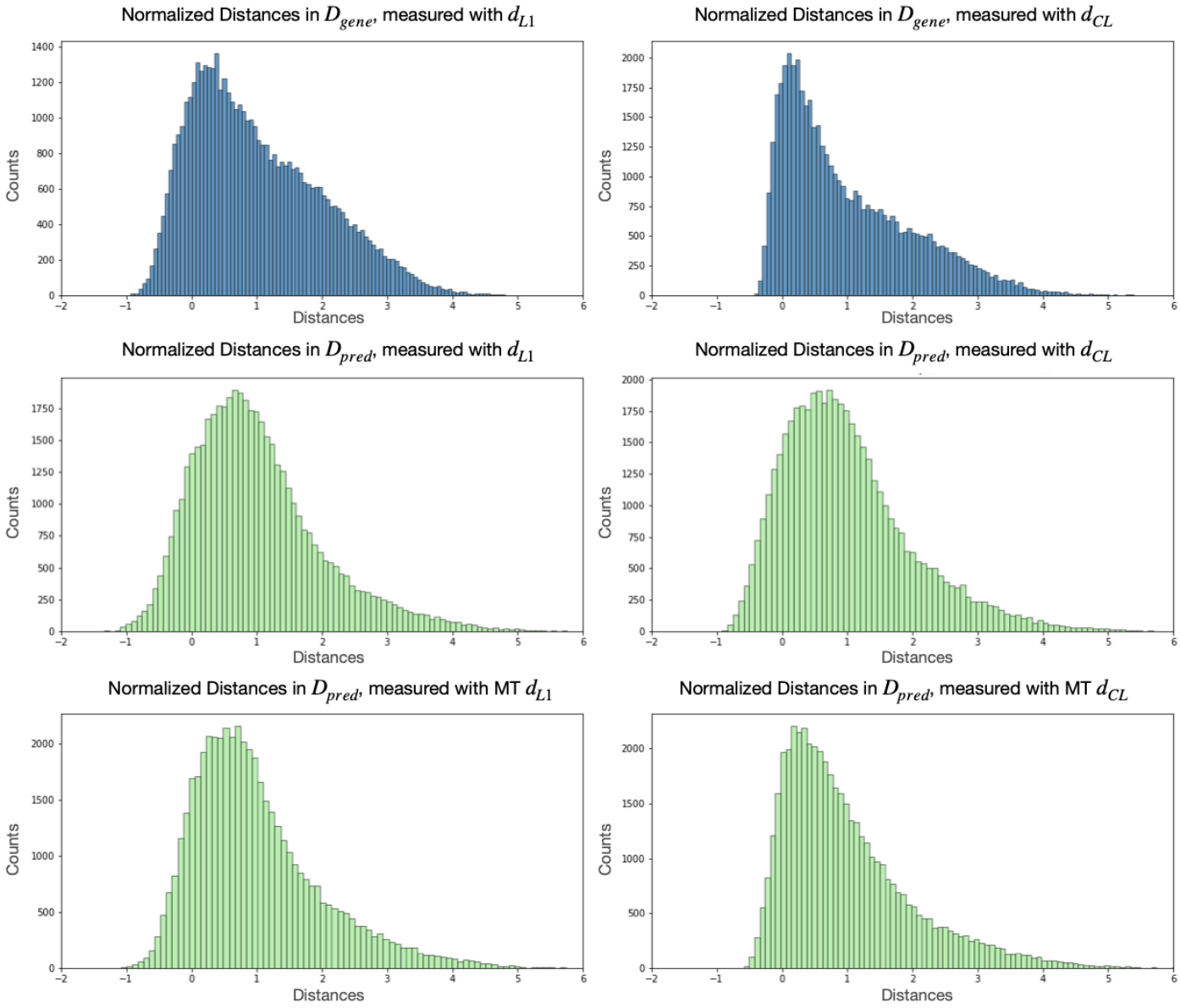


Figure 8.3: Distribution of normalized distances in the training data between the gene expressions, plotted in blue in the top row, and for the distances between predicted embeddings in green in the two bottom rows. The distances are measured with the distance function d_{L1} to the left and d_{CL} to the right, and the predicted distances between WSIs are given by the Single Tile models in the middle row and the Multi Tile models in the bottom row.

8.1.1 Study of Additional Distance Functions for MT

For the further experiments with Multi Tile distance computations, the additional distance functions studied for the metrics in the embedding extraction network were the euclidean distance d_{L_2} in (7.5) and the mean absolute distance d_{MAD} in (7.6). The models will be referred to as MT_L2 and MT_MAD respectively.

The resulting RMSE from 5-fold cross validation for the different distance functions using the Multi Tile distance calculation is presented in the box plot in Figure 8.4. The mean and standard deviations are also presented in Table 8.2. From the box plot it can be seen that the MT_MAD model gave the lowest median RMSE (0.8154) as well as the lowest maximal- (0.9447) and minimal value (0.7041). In Table 8.2 it can be seen that MT_MAD also gave the lowest mean RMSE over the folds (0.8173). The model with the highest median RMSE in the box plot in Figure 8.4 was the MT_L2 model (0.8630) followed by the MT_CL (0.8561) and MT_L1 (0.8482). The same trend can be seen in Table 8.2 where the highest mean RMSE was given by MT_L2 (0.8843) followed by MT_CL (0.8830) and MT_L1 (0.8656). The magnitude of the RMSE should be compared to the relative distances between pairs in the gene expressions plotted for calculations with distance functions d_{L1} and d_{CL} in Figure 8.3 and d_{MAD} and d_{L2} below in Figure 8.6.

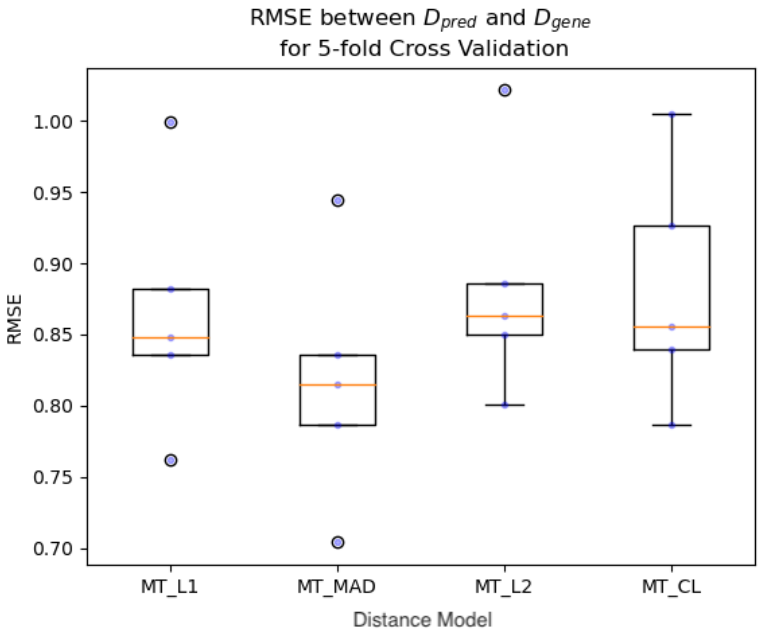


Figure 8.4: Box plot of Root Mean Squared Error (RMSE) for pairwise distances between slide embeddings extracted from the WSIs in D_{pred} and the corresponding distances in the gene expression data D_{gene}. The error is presented for the four Multi Tile models: MT_L1, MT_MAD, MT_L2, MT_CL where each model is evaluated with 5-fold cross validation.

The resulting correlation coefficients from the 5-fold cross validation on the MT models, plotted as box plots in Figure 8.5 and the mean and standard deviation are presented in Table 8.2, also showed that the MT_MAD was the best performing model. MT_MAD resulted in the highest mean- (0.6389), median- (0.6439), min- (0.5188) and max correlation coefficient (0.7361) while MT_L2 had the lowest corresponding measures of ρ where the median was calculated to 0.6158 and mean 0.5941. The second-lowest mean and median of ρ were given by model MT_CL, with mean 0.6018 and median 0.6297.

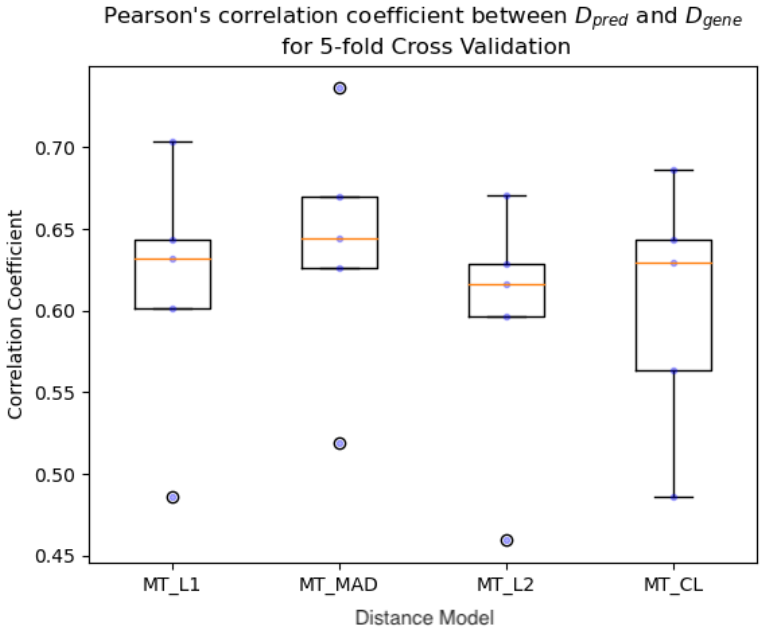


Figure 8.5: Box plot of correlation coefficient ρ for pairwise distances between slide embeddings extracted from the WSIs in D_{pred} and the corresponding distances in the gene expression data D_{gene}. ρ is plotted for the four models: MT_L1, MT_MAD, MT_L2, MT_CL where each model is evaluated with 5-fold cross validation.

Table 8.2: Mean (Standard Deviation) of the Root Mean Squared Error (RMSE) and correlation coefficient ρ for the four models: Single Tile computation with d_{L1} (L1), Multi Tile computation with d_{L1} (MT_L1), Single Tile computation with d_{CL} (CL), Multi Tile computation with d_{CL} (MT_CL), evaluated on 5-fold cross validation.

Model	**RMSE**	$\boldsymbol{\rho}$
MT_L1	0.8656 (0.07757)	0.6135 (0.07167)
MT_MAD	**0.8173 (0.07791)**	**0.6389 (0.07073)**
MT_L2	0.8843 (0.07413)	0.5941 (0.07163)
MT_CL	0.8830 (0.07587)	0.6018 (0.07004)

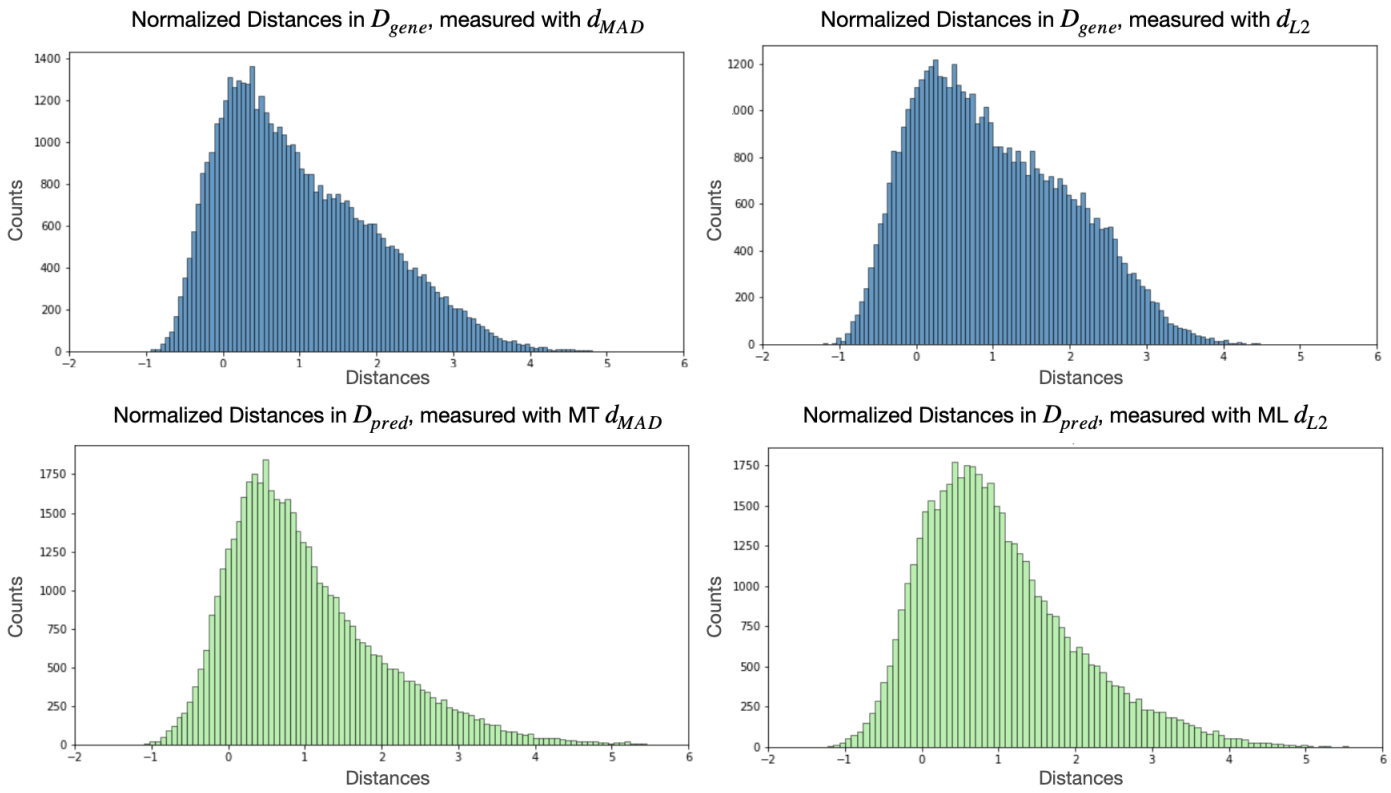
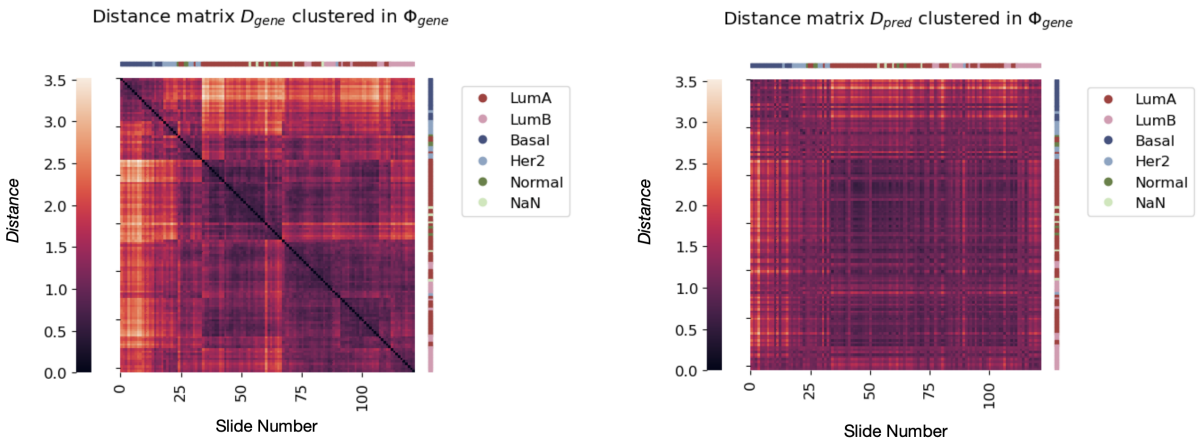


Figure 8.6: Distribution of normalized distances in the training data between the gene expressions, plotted in blue in the top row, and for the distances between predicted embeddings in green in the bottom row. The distances are measured with the distance function d_{MAD} to the left and d_{L2} to the right, and the predicted distances between WSIs are given by the Multi Tile models in the bottom row.

8.2 Final Model - Performance on Test Data

The MT_MAD model was selected as the final model and was evaluated on the test data. Beginning with the qualitative evaluation of the model, the distance matrices were plotted as heatmaps for comparison. The distance matrix for D_{gene} is presented in Figure 8.7a where the matrix is sorted based on hierarchical clustering of the distances in the metric space Φ_{gene}. In Figure 8.7b D_{pred} is sorted in the same order as D_{gene} for comparison purposes. In the figure, the molecular subtypes of the tumours are presented to see the correlation between subtype classification and the hierarchical clustering based on distances in the metric space.



(a) D_{gene} sorted in Φ_{gene}.

(b) D_{pred} sorted in Φ_{gene}.

Figure 8.7: Heatmap of distance matrices D_{gene} and D_{pred} sorted based on hierarchical clustering in Φ_{gene}. The subtype of the tumour in the slides are plotted as separate colour axes on top and to the right of each heatmap. For the slides labelled with the subtype NaN, the molecular subtype of the tumour was not available.

The RMSE between the distance matrices D_{gene} and D_{pred} was calculated at 0.8475. To evaluate how good the resulting distance prediction between slides was in the test data, the difference between the measured distances in D_{pred} and D_{gene} was compared to the distances in the gene expression data between slides. That is, the error in the distance prediction from the network between slides is compared to the inter-slide distance in Φ_{gene}. The histograms of the difference in distances D_{pred} and D_{gene} as well as the distance between slides in the gene expression D_{gene} is presented in Figure 8.8. Ideally, the difference between predicted pairwise distances and the ground truth distances (i.e., items in $|D_{pred} - D_{gene}|$) should be less than the inter-slide distance between gene expressions from different slides (i.e. items in D_{gene}). In the figure it can be seen that the majority of the differences between D_{pred} and D_{gene} are less than the distance between different slides in D_{gene}, but that some slide-pairs have larger differences in distance $|D_{gene} - D_{pred}|$ than the slides with the smallest distances in Φ_{gene}. However, this overlap in distributions consisted of the error between the predicted distance and the ground truth distance, where the slides were far apart in Φ_{gene}, and only 15 distances in the test data, out of 7,381, had a greater absolute difference between the predicted distance and the ground truth distance than the actual distance in Φ_{gene}. That is, for 99.8% of the

distances in the test data, the inequality $|D_{pred} - D_{gene}| > D_{gene}$ was fulfilled. It was also seen that 94.2% of the differences were smaller than half the distance in D_{gene} ($\frac{|D_{pred}-D_{gene}|}{D_{gene}} < 0.5$) and that 56.9% fulfilled $\frac{|D_{pred}-D_{gene}|}{D_{gene}} < 0.25$.

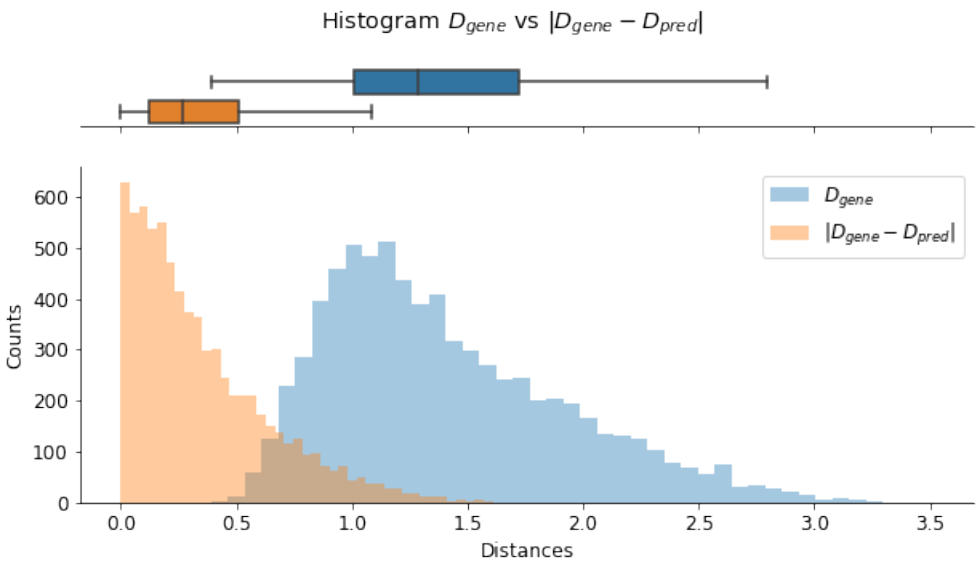


Figure 8.8: Comparison of differences in distance measurement in D_{gene} and D_{pred} (as $|D_{gene} - D_{pred}|$), and the distances in D_{gene}. This to relate the difference in distance between each pair of slides in the metric spaces Φ_{gene} and Φ_{pred}, to the inter-slide distances between gene expressions from different tumours.

The distance matrix D_{pred} was also sorted based on hierarchical clustering in Φ_{pred}. The resulting heatmap, along with the molecular subtypes for the tumour in each slide, is plotted in Figure 8.9.

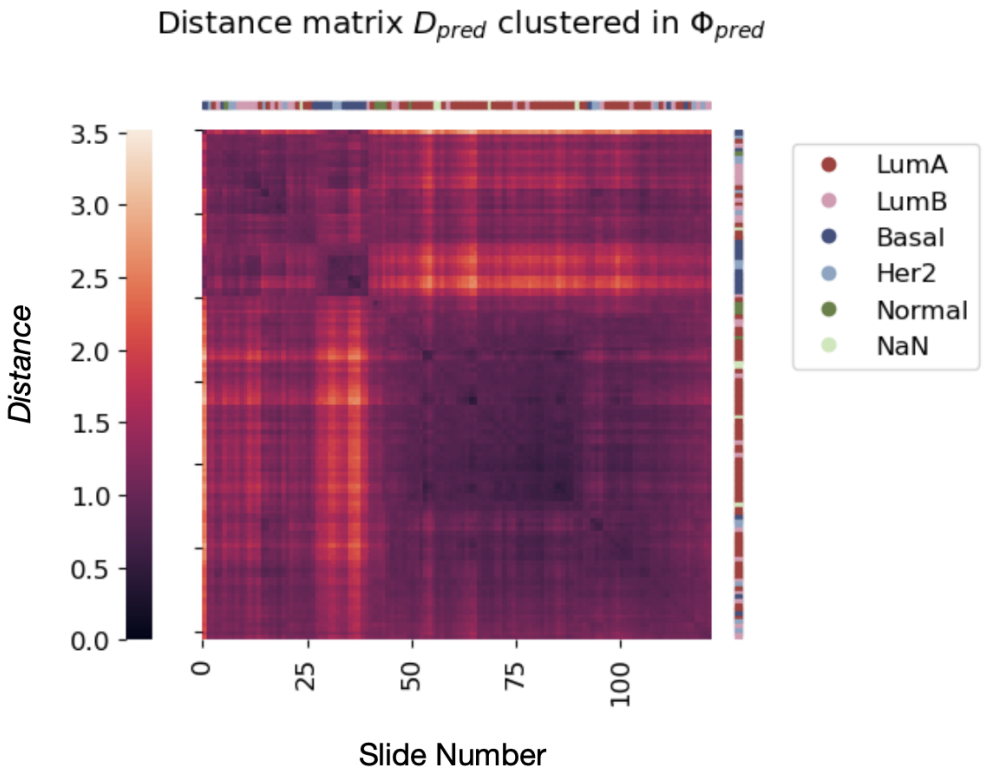


Figure 8.9: Heatmap of distance matrix D_{pred} sorted based on hierarchical clustering in the metric space Φ_{pred}. For each slide, the molecular subtype is presented along the axes at the top and to the left of the heatmap. For the slides labelled with the subtype NaN, the molecular subtype was not annotated for the tumour.

For a more quantitative evaluation of the model's performance on the test data, the correlation between the distances between slides in Φ_{pred} and Φ_{gene} was studied. A scatter plot of the predicted distances and the target distances are plotted in Figure 8.10 along with a linear regression of the data in blue. The distribution of the data points are visualized both with a heatmap of the density of the data points and with a histogram of the distribution of data points over the distances in both Φ_{pred} and Φ_{guide}. In the plot the correlation coefficient ρ is presented which was calculated as $\rho = 0.6315$.

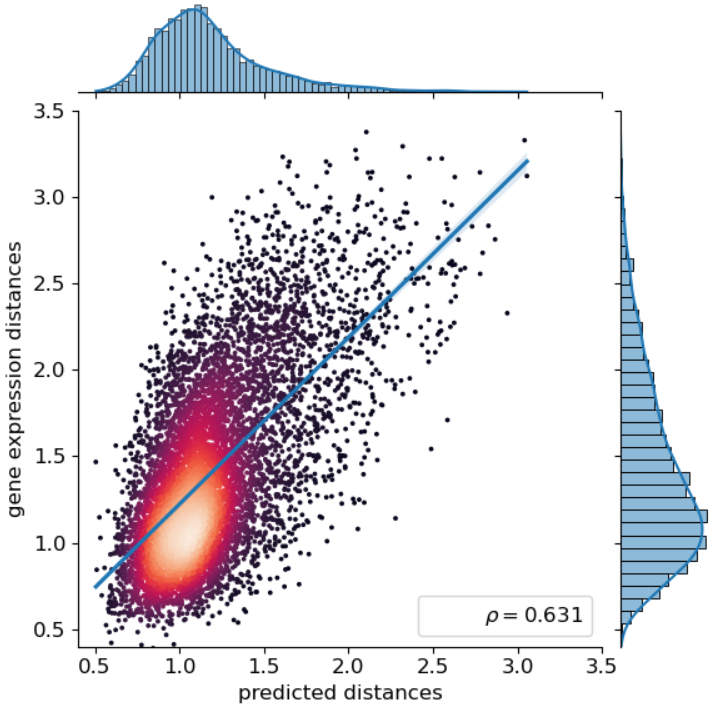


Figure 8.10: Scatter plot of distances between slides, measured in Φ_{pred} on the x-axis and Φ_{gene} on the y-axis. A linear regression of the data is presented as a blue line and the correlation coefficient is presented as ρ in the bottom right of the figure. The distribution of distances is presented both as a density heatmap and in the histograms at the top and to the right of the scatter plot.

8.2.1 Evaluation of Statistical Significance of the Distance Correlation

The distribution of the correlations in the test data measured from the 100,000 permutations in the Mantel test is plotted in blue in Figure 8.11 along with a dotted line at the measured correlation $\rho = 0.6315$. The mean of the sampled correlations was measured to $\bar{\rho} = -1.4706 \cdot 10^{-5}$ and the standard deviation of the sampled distribution was $S = 0.01284$. The standard score, measuring the number of standard deviations by which the measured correlation is above the mean of the sampled distribution of correlation coefficients, was calculated to

$$Z = \frac{\rho - \bar{\rho}}{S} = 49.1718. \tag{8.1}$$

From the Mantel test, a two-tailed hypothesis test can be performed with the null hypothesis:

H_0 : *There is no linear correlation between the predicted distances in* D_{pred} *and the ground truth distances in the gene expression data* D_{gene}.

From the resulting correlation distribution from the permutations in the Mantel test, the null hypothesis can be rejected since the correlation coefficient is statistically significant with a p-value of $< 10^{-15}$. That is, the probability of measuring a correlation $\rho \geq 0.631$, given that there is no correlation between D_{gene} and D_{pred} is $p < 10^{-15}$.

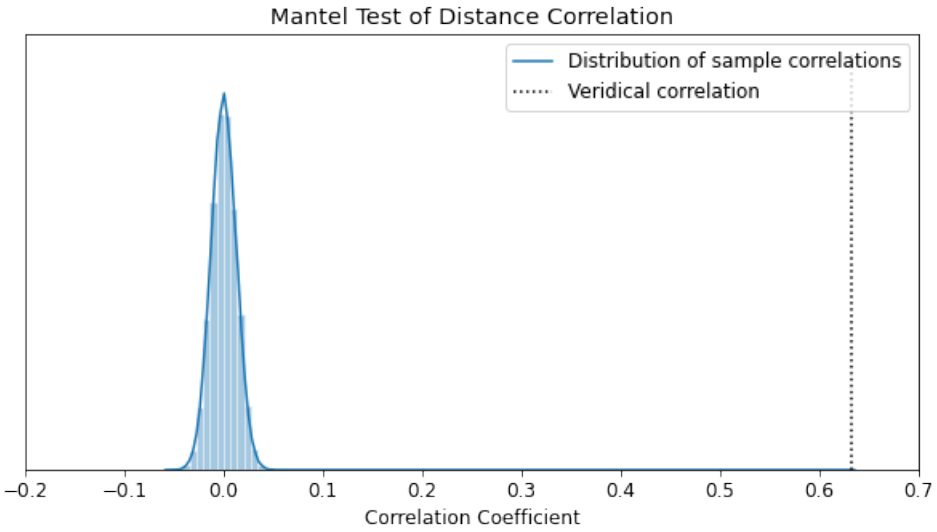


Figure 8.11: Distribution of sample correlation coefficient based on 100,000 permutation Mantel test of the distances in the test data in the metric spaces Φ_{pred} and Φ_{gene}. The true measured correlation coefficient between the distance matrices D_{pred} and D_{gene} is marked as the vertical correlation in the dotted line.

8.2.2 Visualization of Metric Space

To visualize the embedding spaces Φ_{gene} and Φ_{pred} for the final model MT_MAD a non-linear dimensionality reduction of the embeddings in the metric spaces was performed via Uniform Manifold Approximation and Projection (UMAP). The UMAP transformation of the data was achieved with the umap-learn package in Python, which implements the learning technique based on the Riemannian geometry and algebraic topology, see [40].

The two first components from the UMAP projection of the test data in the metric spaces Φ_{gene} and Φ_{pred} are presented in Figure 8.12. In the figure, each embedding is coloured based on the molecular subtype of the tumour. In case there was no annotation of the molecular subtype, the embedding is marked with NaN.

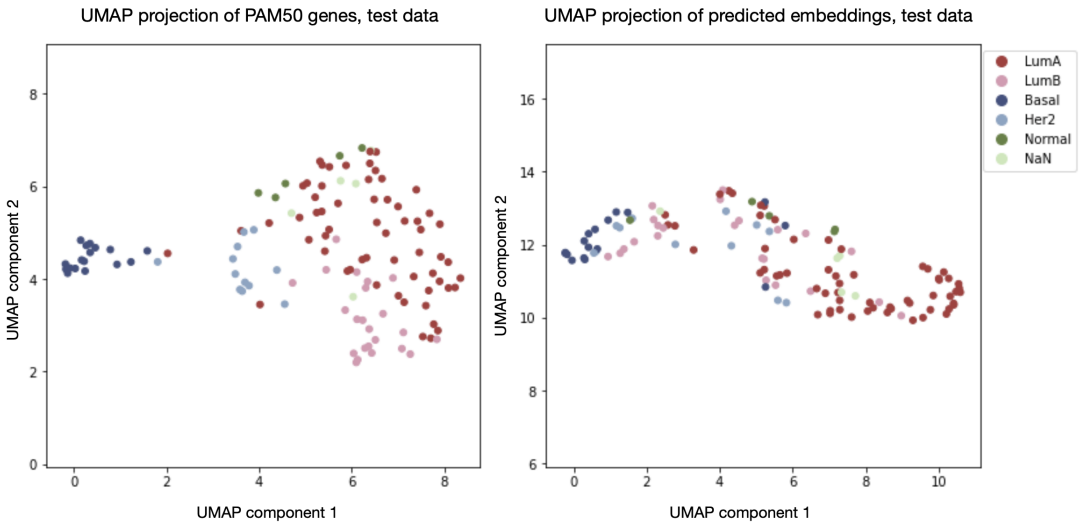


Figure 8.12: UMAP projection of test data in Φ_{gene} (gene expression data) and Φ_{pred} (embeddings extracted from the final embedding extraction network using metric MT_MAD). The data is plotted in the two first UMAP components, and each data point is coloured based on the molecular subtype of the tumour. In case there were no annotated subtypes, the embeddings are marked with NaN.

Chapter 9

Discussion

The aim of the thesis was to examine whether a CNN-based deep neural network could be used to map the WSIs of female breast cancer tumours into an embedding space in which information from the relative distances in the gene expressions was captured. In the following section, the results presented in chapter 8 will be analysed to establish if the aim of the thesis has been achieved.

9.1 Analysis of Model Selection

From the results of the first study of the models with distance functions d_{L1} and d_{CL}, it was shown that the L1-distance, both measured with ST and MT, had better correlation coefficient (closer to 1) and lower RMSE. This could be due to that the cosine similarity is based on the distance measurement on the angle between points in the metric space. Looking at the PCA projection of the gene expression embeddings in Figure 7.3, there are several embeddings that have the same angle to another point, but are located at different distances. Although the PCA projection is a simplification of the distance relation between the 50-dimensional vectors, one can get an indication that the cosine distance measurement may result in less separation of the gene expression embeddings in Φ_{gene}. This can be seen in the distance distribution in 8.3 where a majority of the distances are centred close to 0 when measured in d_{CL}, while the distances between slides are more spread out, between values around 0 to 3, when measured with d_{L1}. Concerning the evaluation of the models, one needs to consider how well the models are based on the measured RMSE and correlation coefficient. Since this gene expression guide is a novel approach for the field of metric learning in histopathol-

ogy, with, to the author's knowledge, no previous research, it is hard to relate how good this RMSE is compared to other models studying the specific task. To get an indication of how well the RMSE between the normalized distances in D_{pred} and D_{gene} was, it was compared to the distribution of normalized distances in gene expression between slides (Figure 8.3). One can see that the normalized distances between slides have an approximate range of 5, which would indicate that if the RMSE was close to 5 we would have a root mean squared difference that is as large as the maximal difference between pairwise distances between patients. That is, we would have larger error between the predicted- and gene expression distance than the range of the ground truth data. Although this comparison is not enough to decide how good an RMSE of around 1 is, since it can be seen that most of the distances between slides in Φ_{gene} are around 0 to 1. Therefore, additional evaluation of the difference between distances in Φ_{pred} and Φ_{gene} were considered for the evaluation of the model's performance on the test set, which will be discussed below. Additional analysis of the statistical significance of the correlation between distances in Φ_{pred} and Φ_{gene} will also be discussed below for the test data.

From the experiments comparing the models L1, MT_L1, CL, and MT_CL it was shown that the multi tile calculations were performing better both in consideration of the RMSE between distances and for the correlation coefficient. This could be since the network is getting a more comprehensive and generalized view of the slides since it is given four tiles from each slide in the distance calculation instead of only one. Technically speaking, the network will base its calculations of distances between embeddings on an average over four tiles from each slide and will hence be less prone to tune the network parameters based on outliers, or tiles that are dissimilar to other tiles in the slide. Since, as introduced earlier in the thesis, cancer tumours are heterogeneous, tiles will have different morphological features depending on the location in the slide, giving that an average over several tiles (MT calculations) was shown to be preferred in this application.

In the thesis, the MT models were considered superior to the ST models, and two additional distance functions: d_{MAD} and d_{L2} were studied for the multi tile calculations. While the Euclidean distance d_{L2} did not show any improvement in the RMSE nor correlation coefficient, it was seen that taking an average over the embedding size for the L1-distande in d_{MAD} improved both RMSE and correlation coefficient. This was not

expected, as it was thought that scaling the distance measurement with a constant would result in a similar performing model. In the following, the possible reason why taking the average over the absolute distance in d_{MAD} gives better results, compared to taking the sum over the absolute distance in d_{L1}, will be discussed.

Initially, it was thought that since the distance measurements results in smaller distance values, it could affect the distribution of normalized distances and therefore also have an impact on the computed RMSE as to tend towards smaller values. Although, looking at the normalized distance distributions, both concerning distances between predictions and gene expressions, using d_{MAD} in Figure 8.6 and comparing to the distribution using d_{L1} in Figure 8.3 it can be seen that the distributions share the same range, median and shape of the gene distance distribution, which contradicts this initial hypothesis. Concerning the correlation coefficient, it is not dependent on the magnitude of the data, and multiplying the distances by a constat $\frac{1}{M}$, will not change the correlation coefficient. Hence, the average over the absolute distance should not affect the correlation coefficients either. Instead, it seems as if the learning of the network is improved by the averaging in the metric. This can be noted in comparing the distance distributions from the models, since the distance distribution between predicted image embeddings are more similar to the true distribution of distances between gene expressions when using MT_MAD, whereas the MT_L1 results in a more smooth distribution. This could be due to that, for the d_{L1} distance measurement, the summation over absolute difference for each component of the embedding will result in big distances, and especially big difference in distances between slide-pairs (this can be seen in the distance distribution in Appendix B where the L1-metrics has much higher distances than the other metrics). This may result in a big variety in the loss function depending on which slides are considered in the batch, and hence a large gradient estimate. This, in turn, will lead to large changes in the weights of the network, which will affect the training of the network. Since the same optimization and hyperparameters (except for the learning rate) are used for the models, it could be further evaluated if another set of hyperparameters would be more optimal for the L1 models. The hypothesis that the training of the network is affected by the distance distribution could also be tested for the Euclidean metric d_{L2} (which also measures higher distances compared to d_{MAD} and d_{CL}) to see if an average over the dimension of the embeddings could improve the MT_L2 model. A normalization of the distances in D_{gene} could also have been

performed prior to the training of the network to reduce this effect. A proposition is to force the metric spaces to be a unit hypersphere where the maximal distance between points is set to 1.

9.2 Analysis of Final Model's Performance on Test Data

The final model MT_MAD was evaluated on the test data and the results were presented in Section 8.2. The distance matrices D_{gene} and D_{pred}, both sorted by the hierarchical clustering of D_{gene}, are shown in Figure 8.7. In the distance matrix D_{gene}, it can be seen that hierarchical clustering of slides in Φ_{gene} showed correlation with the molecular subtypes of the tumours, where slides showing Basal-Like breast cancers were clustered to the far left and Luminal-B cancers to the right. This is expected since the PAM-50 genes are the ground to the molecular subtype classification, but it is a good way to check that the guiding metric space is separating the subtypes with the chosen metric d_{MAD}. When comparing the predicted distances in D_{pred} with the same order of slides as in D_{gene}, it can be seen that the overall characteristics of the distance matrix D_{gene} are captured by the predicted distances, but that some distances are under- or overestimated. Something that can be noted is that the distances between slides with the same molecular subtypes are slightly lower than those between slides with different subtypes. This can be seen as the darker squares close to the diagonal. Although the trained model does not capture all the relative distances between slides as in D_{gene} and further development of the method is needed to better predict the exact distances in Φ_{gene}. Something interesting can be seen in the lighter cross in D_{pred} in figure 8.7b at approximately slide number 80. Here, a small cluster of Luminal-B cancers, which does not appear in D_{gene}, is picked up by the network. This could be that the embedding extractor network more easily detects interclass distances to Luminal-B type cancers, especially distances between Luminal-A and Luminal-B. From a qualitative evaluation, it seems as if the predicted distance matrix captures the general differences between the clustering in Φ_{gene} but that the more detailed and local differences in distance are not captured by the image-based network.

To obtain a more quantitative evaluation of the performance of the model in estimating the relative distances between slides in Φ_{gene}, the absolute difference in the distances D_{gene} and D_{pred} was plotted along with the distances in D_{gene} in Figure 8.8. Alike mentioned in Section

8.2, the difference in distance $|D_{gene} - D_{pred}|$ should be smaller than the distance between slides in Φ_{gene}, however the overlap in the distributions shows that some distance predictions had higher absolute error to the true distance than the smallest measured distances between slides in Φ_{gene}. Although, as presented with the results, the greatest errors were given by the slides far apart in Φ_{guide}, and the majority of the errors were small in relation to the actual gene distance between the WSIs. More than half of the distances in the test data had errors smaller than a fourth of the actual distance in Φ_{gene}, and 94.2% of the errors were smaller than half the actual distance measured in Φ_{gene}. Although it should still be noted that the precision in distance prediction is worse than the smallest distances in D_{gene}.

When the distance matrix D_{pred} was clustered in Φ_{pred}, shown in Figure 8.9, the slides of different subtypes were not as separated. It can be seen that a cluster of Basal-Like breast cancers appeared at slide number 25 to around 40, where the distances to other slides were higher (approximately 2.5 - 3) than the intra-class distance (approximately 1), as can be seen in the small dark square close to the diagonal. The purpose of the thesis was not to cluster the breast cancer subtypes; rather, plotting the subtypes along with the distance matrix was a way to examine whether the relative distances in Φ_{pred} maintained some separation between the molecular subtypes of the tumours. Lastly, it can be seen that the diagonal in the distance matrix D_{pred} is slightly darker than the rest of the distances, but is not 0 as in D_{gene}. The distances along the diagonal are calculated as the mean distance between the same slide, where each slide is represented by 500 randomly drawn tiles from the WSI. The value of a diagonal element $D[i][i] \neq 0$ indicates that tiles from the same slide may be mapped to different locations in Φ_{pred}. How much the distance measurements depends on which tiles that are selected needs to be further investigated as discussed in Section 9.3.

Continuing with the examination of the correlation between the distances in D_{pred} and D_{gene}, it can be seen in Figure 8.10 that the correlation coefficient was measured to $\rho = 0.631$, compared to the perfectly linear correlation coefficient $\rho_{max} = 1$. In the scatter plot, the density of the distances are presented in the heatmap, and it can be seen that the majority of distances in D_{gene} are between 0.5 to 2.0. The predicted distances are generally underestimated, which can be seen firstly in the scatter plot, where there are slides with distances over 1.5 that are estimated to between 1.0 to 1.5 by the network. Secondly, it can be seen in

the predicted distance distribution at the top of Figure 8.10, where there are less predicted distances over 1.5 in comparison to the distribution of gene expression distances to the right of the figure. Thirdly, it can be seen in the distance matrices in figure 8.7 where the distances are lower (i.e. darker colours in the heatmap) in D_{pred} compared to D_{gene}. This could be due to that a majority of the distances in the training data in Φ_{gene} are less than 2.0, looking at the distribution of distances in Figure B.3 in Appendix B. Due to the distance distribution in the training data, the network will be less likely to get an input where the distance between slides are above 2.0 and hence will most likely be worse at estimating the bigger dissimilarity between slides in the data. This can be seen in the distribution of predicted distances at the top of the scatter plot in Figure 8.10 where the tail of higher distances is much shorter and thinner than in the gene expression distances, where there are only a few pairs of slides with predicted distances greater than 2.0.

To study how statistically significant the correlation between the predicted- and guiding distances was, the mantel test was performed. This statistical test gives an indication of how probable it is, given the two distance measurements D_{gene} and D_{pred}, that the correlation coefficient measured ($\rho = 0.6135$) was due to a random correlation in the distances. The correlation coefficients between D_{pred} and D_{gene} from 100,000 random permutations were plotted in 8.11. It was calculated that the probability of measuring a correlation of $\rho \geq 0.6315$, given that there was no correlation in the distance matrices, was less than 10^{-15}. This small probability can also be seen in the figure where the measured, veridical correlation is far away from the sampled distribution from the Mantel test. The Z-score was calculated at 49.2, showing that the veridical correlation coefficient was more than 49 standard deviations from the sampled mean. In conclusion, we can reject that there is no correlation in the distance matrices D_{pred} and D_{gene}.

The locations of the predicted embeddings in Φ_{pred} were also visualized by doing a dimensionality reduction of the embedding space, using UMAP, and plotting the embeddings projected onto the two most principal UMAP components. The UMAP projection of the predicted embeddings was compared to the UMAP projection of the gene embeddings in Figure 8.12. In the visualization of the embeddings, it can be seen that the predicted metric space Φ_{pred} captures that the Basal-Like breast cancers and Luminal-A breast cancers are the furthest apart in the metric space Φ_{gene}. In the predicted metric space, it can be seen

that Luminal-B breast cancers and the HER2 breast cancers are more spread between the Basal-Like to the far left and the Luminal-A to the far right. If this is due to that the network has not learnt to determine the similarity between these types of slides, or if the network is recognizing patterns in the slides which shows inter-class similarities between the subtypes, needs to be further analysed. As an example, there is a cluster of Luminal-B type breast cancers, spotted to the left in Figure 8.12, which the network identifies as more similar to the Basal-Like group than what the gene expressions indicate.

9.3 Sources of Error and Discussion of Delimitations

As noted throughout the report, the training of the embedding extraction network is weakly supervised, and the mapping of tiles to embeddings in the metric space Φ_{pred} is guided by the gene expression of the entire slide. Due to the known heterogeneity of cancer tumours, tiles from the same WSI will have different appearance, and the weights in the network will be trained differently depending on which tiles that are randomly selected from the slide. Since 10,000 tiles are looked at for each epoch, and there are 697 WSIs in the training data (total of 4.81 million tiles), multiple tiles from the same slide will be handled in the training. Since the network will learn its weights based on the input and, most importantly, the frequency of the input, the network will hopefully base the update of the weights on an average phenotype of the slide. This problem was briefly addressed by basing the distance in the loss calculation on four tiles from the same slide instead of letting only one tile from each slide determine the distance for the slide pair. This contribution improved both the RMSE and correlation coefficient for the L1- and CL models, seen in Table 8.1. The standard deviation in the results over the folds were lowered in the L1 model using the multi tile calculation, although the standard deviation was higher when using the MT calculation in the CL model in comparison to using only one tile in the measurement. This needs to be further studied, since the idea of the MT-measurement was to lower the model's dependence on which tiles were sent to the network by averaging the distance calculation over several tiles. A reason could be that, when calculating the distances in the predicted metric space, 500 tiles from each slide were used in the average distance between slides. The distance measurement will then be dependent on which tiles were randomly selected in the computation

and can therefore affect the result. Additional propositions to study this error will be presented in Section 10.1.

Another source of error, or rather simplification, is that only the PAM50 genes were used in the embedding guide. This was a conscious choice based on studies proposing that this set of genes are sufficient for molecular subtype classification of female breast cancer. Although considering the expression of additional genes may improve the model.

Lastly, the delimitation of which hyperparameters that were considered in the tuning of the network may affect the performance of the model. The tuning of the embedding dimension and batch size was done in an early stage of the model, where a log-transformation was applied to the gene expression data. After the selection of the parameters, based on this grid search, the parameters were not additionally tuned for the studied models due to lack of time. It was found that, with the set of hyperparameters used in this study, MT_MAD was the best performing metric. Although, tuning the hyperparameters for each metric studied in the thesis could lead to another model outperforming MT_MAD. To further develop the model, additional hyperparameter tuning needs to be done, for at least the batch size and embedding dimension, but also for the hyperparameters in the Adam optimizer.

9.4 Ethical Consideration

During the thesis, no sensitive personal information or data that could have compromised the integrity of any concerned parties was available. All data was handled with a secure data infrastructure and the project has ethical approval.

Chapter 10

Conclusion and Future Work

In summary, the best performing metric to guide the embedding extraction network in mapping breast cancer WSIs to embeddings that incorporated the information given by the genes expressed in the tumour was the mean absolute distance d_{MAD}. It was also seen that letting the network handle multiple tiles from the same slide when comparing the distance between slides in Φ_{pred} and Φ_{gene}, improved the model's ability to capture the relative distances between slides in Φ_{gene}. Evaluating the correlation between the distances between embeddings, predicted by the MT_MAD model, and the ground truth distances between gene expressions, it was seen that the correlation coefficient was measured to $\rho = 0.631$ on the test data. The statistical significance of the correlation was evaluated using the Mantel test, resulting in a standard score of $Z = 49.2$ and a p-value of $< 10^{-15}$.

In conclusion, we can answer the question of research. A deep neural network can be trained that maps the histopathological WSI of breast cancer tumours so that the information given by genes expressed in the tumour is incorporated. This has been shown by comparing the predicted- and ground truth distances in the metric spaces, as well as serving with statistical evaluation of the results. A visualization of the UMAP projection of the embeddings in the respective metric spaces showed that the overall distance relation between slides in Φ_{gene} could be incorporated in the predicted embeddings in Φ_{pred}. Although the difference in the location of the slides in Φ_{pred} and Φ_{gene} must be further, and more quantitatively, examined to determine whether the network performs poorly in measuring the similarity between certain WSI phe-

notypes, if only looking at two dimensions of the UMAP is not capturing all distance-relation of the data, or whether it could be that the network detects patterns in the data that are not described by the PAM50 genes. Lastly, the thesis shows that an image based deep metric learning model may have the potential of serving as a cost-effective, reproducible alternative to gene expression profiling, where there is a possibility of further research for the application.

10.1 Future Work

To develop the proposed gene expression guided distance metric learning in the area of histopathology, the effect of the weakly supervised setting that comes from having the guide on slide level should be addressed. Firstly, the variance of the model's performance, depending on which tiles are considered in the distance computation, both in the training, and in the final evaluation, is proposed to be studied. This gives an indication of how sensitive the network is to the choice of tiles in the distance measurements. Secondly, the weakly supervised model could be compared to a model where the gene expression is measured on smaller parts of the tumour, so that the effect of assuming that all tiles have the same gene profile can be addressed.

It could also be studied whether the use of a CNN model with a mixed attention mechanism for the embedding extractor network, proposed in [36], has the potential to improve the model. The embedding network would then take into consideration the spatial structure in the images, which may result in that redundant tiles from the slides are less weighted in the training of the network. Another proposition to address the problem of redundant tiles in the slides is to do hard mining and weighting of the slide pairs in the training of the embedding extraction network. As proposed in the Multi-Similarity Loss [35], the weighting of positive pairs that are more dissimilar than the most similar negative pair, and vice versa, could improve the model since the network is forced to handle "hard" positive and negative pairs. The approach was briefly looked at in the final stages of the thesis, and the Multi-Similarity Loss was implemented so that tiles were separated based on that tiles from the same slide should be close in the metric space, and tiles from different slides should be far apart. Although, since we do not only want to separate the slides in the metric space, but to also guide the metric space towards Φ_{gene}, one needs to address how to incorporate the embedding guide to the Multi-Similarity Loss. One way could be to first train an

embedding extractor that maps WSIs to embeddings in a metric space, which separates the slides based on similarities between the images and keeps tiles from the same slide close (this, using the triplet-based hard mining in regular deep metric learning). The embedding space, separating the slides based on image similarity, could then be guided towards describing the gene expressions in Φ_{gene}.

Additional improvements of the network should also be considered, where tuning of the optimization method, network architecture and hyperparameters is needed. The effect of embedding dimensionality on the model's performance should also be evaluated, as well as how many slides that should be compared in the proposed loss calculation. It is also proposed to investigate how the sampling of tiles affects the performance of the model. In this thesis, only four tiles from the same slide were considered in the multi tile distance computation. Additional experiments could be performed, determining how many tiles are optimal in the average distance measurement for the multi tile calculation.

To further improve the results, it is necessary, as in all machine learning approaches, to train the model on more data, in this case more whole slide images. Although the networks are trained on nearly 5 million tiles, the slides come from 697 different patients from two different cohorts. The model should also be evaluated on data from another hospital to investigate how well the model generalizes to differently scanned slides.

Lastly, the predicted metric space's ability to capture the similarity in Φ_{gene} could be further studied by looking at the clustering of predicted embeddings in Φ_{pred}, in comparison to the clustering of gene expression in Φ_{gene}.

Bibliography

[1] Jacques Ferlay, Murielle Colombet, Isabelle Soerjomataram, Donald M Parkin, Marion Piñeros, Ariana Znaor, and Freddie Bray. Cancer statistics for the year 2020: An overview. *International Journal of Cancer*, 149(4):778–789, 2021.

[2] Dongfeng Tan and Lynch Henry T. *Principles of Molecular Diagnostics and Personalized Cancer Medicine*, chapter 27. Lippincott Williams & Wilkins, Philadelphia, 2013.

[3] Renan Gomes do Nascimento and Kaléu Mormino Otoni. Histological and molecular classification of breast cancer: what do we know. *Mastology*, 30:1–8, 2020.

[4] Therese Sørlie, Charles M Perou, Robert Tibshirani, Turid Aas, Stephanie Geisler, Hilde Johnsen, Trevor Hastie, Michael B Eisen, Matt Van De Rijn, Stefanie S Jeffrey, et al. Gene expression patterns of breast carcinomas distinguish tumor subclasses with clinical implications. *Proceedings of the National Academy of Sciences*, 98(19):10869–10874, 2001.

[5] Darina Vuong, Peter T Simpson, Benjamin Green, Margaret C Cummings, and Sunil R Lakhani. Molecular classification of breast cancer. *Virchows Archiv*, 465(1):1–14, 2014.

[6] Yifei Chen, Yi Li, Rajiv Narayan, Aravind Subramanian, and Xiaohui Xie. Gene expression inference with deep learning. *Bioinformatics*, 32(12):1832–1839, 2016.

[7] Nadia Harbeck, Frédérique Penault-Llorca, Javier Cortes, Michael Gnant, Nehmat Houssami, Philip Poortmans, Kathryn Ruddy, Janice Tsang, and Fatima Cardoso. Breast cancer. *Nature Reviews Disease Primers*, 5(1), September 2019.

[8] Gulisa Turashvili and Edi Brogi. Tumor heterogeneity in breast cancer. *Frontiers in medicine*, 4:227, 2017.

[9] Dimitrios Vlachakis. *Gene Expression Profiling in Cancer*, chapter 1. IntechOpen, London, England, 2019.

[10] Nalini Raghavachari and Natalia Garcia-Reyero. *Gene expression analysis*, chapter 1-2. Humana Press, New York, NY, 2019.

[11] Torsten Nielsen, Brett Wallden, Carl Schaper, Sean Ferree, Shuzhen Liu, Dongxia Gao, Garrett Barry, Naeem Dowidar, Malini Maysuria, and James Storhoff. Analytical validation of the pam50-based prosigna breast cancer prognostic gene signature assay and ncounter analysis system using formalin-fixed paraffin-embedded breast tumor specimens. *BMC cancer*, 14(1):1–14, 2014.

[12] K David Voduc, Maggie CU Cheang, Scott Tyldesley, Karen Gelmon, Torsten O Nielsen, and Hagen Kennecke. Breast cancer subtypes and the risk of local and regional relapse. *Journal of clinical oncology*, 28(10):1684–1691, 2010.

[13] Jin Hu, Huiqiong Zhang, Fang Dong, Ximeng Zhang, Shuntao Wang, Jie Ming, and Tao Huang. Metaplastic breast cancer: Treatment and prognosis by molecular subtype. *Translational oncology*, 14(5):101054, 2021.

[14] Saber Fallahpour, Tanya Navaneelan, Prithwish De, and Alessia Borgo. Breast cancer survival by molecular subtype: a population-based analysis of cancer registry data. *Canadian Medical Association Open Access Journal*, 5(3):E734–E739, 2017.

[15] Laura J Van't Veer, Hongyue Dai, Marc J Van De Vijver, Yudong D He, Augustinus AM Hart, Mao Mao, Hans L Peterse, Karin Van Der Kooy, Matthew J Marton, Anke T Witteveen, et al. Gene expression profiling predicts clinical outcome of breast cancer. *nature*, 415(6871):530–536, 2002.

[16] Xiaofeng Dai, Ting Li, Zhonghu Bai, Yankun Yang, Xiuxia Liu, Jinling Zhan, and Bozhi Shi. Breast cancer intrinsic subtype classification, clinical use and future trends. *American journal of cancer research*, 5(10):2929, 2015.

[17] Guy Orchard and Brian Nation. *Histopathology*. Oxford University Press, 2011.

[18] Balázs Acs, Mattias Rantalainen, and Johan Hartman. Artificial intelligence as the next step towards precision pathology. *Journal of internal medicine*, 288(1):62–81, 2020.

[19] André Huisman, Arnoud Looijen, Steven M van den Brink, and Paul J van Diest. Creation of a fully digital pathology slide archive by high-volume tissue slide scanning. *Human pathology*, 41(5):751–757, 2010.

[20] John KC Chan. The wonderful colors of the hematoxylin–eosin stain in diagnostic surgical pathology. *International journal of surgical pathology*, 22(1):12–32, 2014.

[21] Amelie Echle, Niklas Timon Rindtorff, Titus Josef Brinker, Tom Luedde, Alexander Thomas Pearson, and Jakob Nikolas Kather. Deep learning in cancer pathology: a new generation of clinical biomarkers. *British journal of cancer*, 124(4):686–696, 2021.

[22] Neofytos Dimitriou, Ognjen Arandjelović, and Peter D Caie. Deep learning for whole slide image analysis: an overview. *Frontiers in medicine*, page 264, 2019.

[23] S Kevin Zhou, Hayit Greenspan, and Dinggang Shen, editors. *Deep learning for medical image analysis*. Academic Press, San Diego, CA, January 2017.

[24] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[25] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[26] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[27] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[28] Junlin Hu, Jiwen Lu, and Yap-Peng Tan. Discriminative deep metric learning for face verification in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1875–1882, 2014.

[29] Zechao Li and Jinhui Tang. Weakly supervised deep metric learning for community-contributed image retrieval. *IEEE Transactions on Multimedia*, 17(11):1989–1999, 2015.

[30] Xudong Lin, Yueqi Duan, Qiyuan Dong, Jiwen Lu, and Jie Zhou. Deep variational metric learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 689–704, 2018.

[31] Huseyin Coskun, David Joseph Tan, Sailesh Conjeti, Nassir Navab, and Federico Tombari. Human motion analysis with deep metric learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 667–683, 2018.

[32] Moses Charikar. Lecture 1: Metric spaces, embeddings, and distortion. "`https://web.stanford.edu/class/cs369m/cs369mlecture1.pdf`", September 2018.

[33] Juan Luis Suárez, Salvador García, and Francisco Herrera. A tutorial on distance metric learning: Mathematical foundations, algorithms, experimental analysis, prospects and challenges. *Neurocomputing*, 425:300–322, 2021.

[34] Timo Milbich, Karsten Roth, Biagio Brattoli, and Björn Ommer. Sharing matters for generalization in deep metric learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(1):416–427, 2020.

[35] Xun Wang, Xintong Han, Weilin Huang, Dengke Dong, and Matthew R Scott. Multi-similarity loss with general pair weighting for deep metric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5022–5030, 2019.

[36] Pengshuai Yang, Yupeng Zhai, Lin Li, Hairong Lv, Jigang Wang, Chengzhan Zhu, and Rui Jiang. A deep metric learning approach for histopathological image retrieval. *Methods*, 179:14–25, 2020.

[37] Philipp Gräbel, Martina Crysandt, Barbara M Klinkhammer, Peter Boor, Tim H Brümmendorf, and Dorit Merhof. Guided representation learning for the classification of hematopoietic cells. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 545–551, 2021.

[38] Daniel Müllner. fastcluster: Fast hierarchical, agglomerative clustering routines for r and python. *Journal of Statistical Software*, 53:1–18, 2013.

[39] Georg Lindgren, Holger Rootzen, and Maria Sandsten. *Stationary stochastic processes for scientists and engineers*. CRC Press, Boca Raton, FL, October 2013.

[40] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

Appendix A

Presentation of Loss Curves

A.1 Grid Search over Hyper Parameters

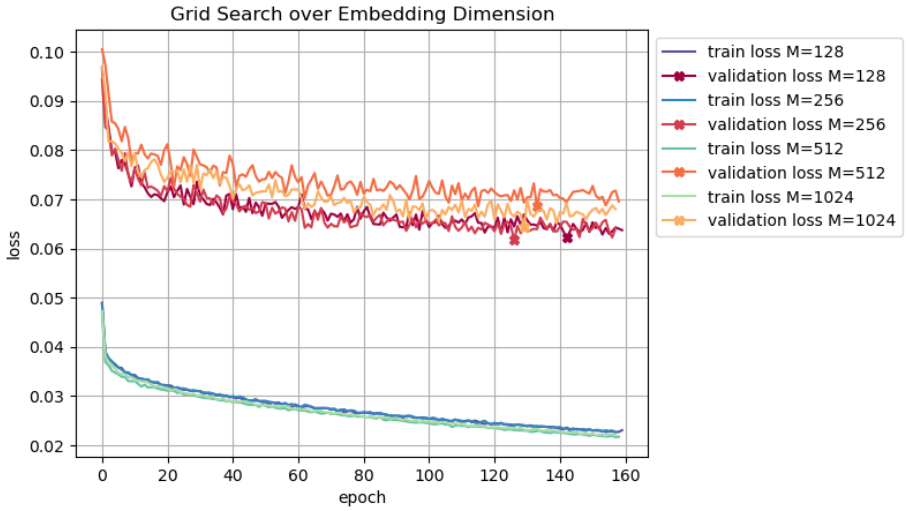


Figure A.1: Training- and validation loss for grid search over embedding dimensions $M = \begin{bmatrix}128 & 256 & 512 & 1024\end{bmatrix}$. Each training is based on a random set of 80% of the training data and all other hyperparameters were fixed (batch size 32, learning rate 10^{-5}).

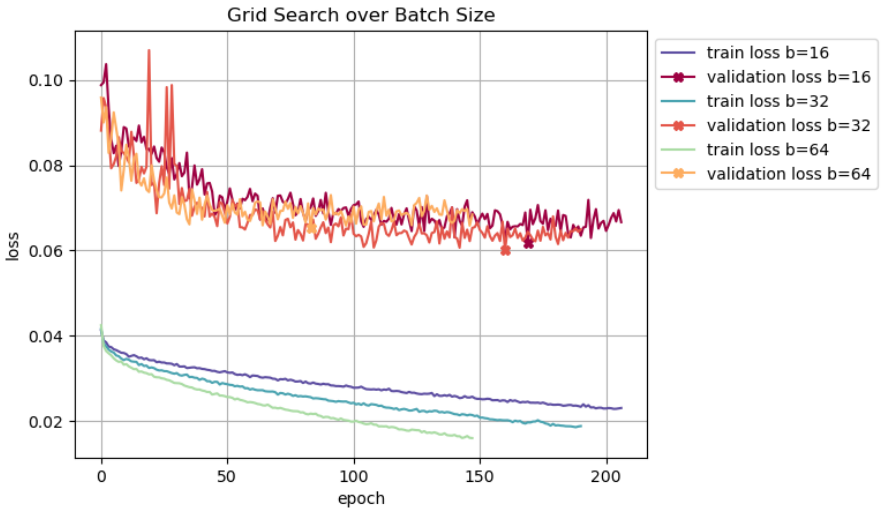


Figure A.2: Training- and validation loss for grid search over batch size $b = \begin{bmatrix}16 & 32 & 64\end{bmatrix}$. Each training is based on a random set of 80% of the training data and all other hyperparameters were fixed (embedding dimension 128, learning rate 10^{-4}).

A.1.1 Learning Rate

L1 distance

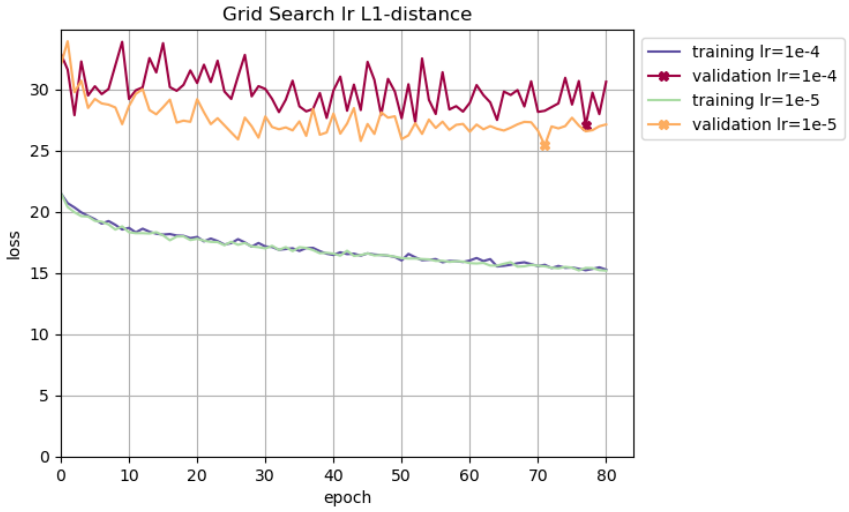


Figure A.3: Training- and validation loss for one fold from 5-fold cross validation when using the distance function d_{L1} with single tile calculation and doing a grid search over the learning rates $\alpha = \begin{bmatrix} 10^{-4} & 10^{-5} \end{bmatrix}$.

L2 distance

Figure A.4: Training- and validation loss for one fold from 5-fold cross validation when using the distance function d_{L2} with multi tile calculation and doing a grid search over the learning rates $\alpha = \begin{bmatrix} 10^{-4} & 10^{-5} \end{bmatrix}$.

A.2 Results from 5-fold Cross Validation

A.2.1 Single Tile Calculations

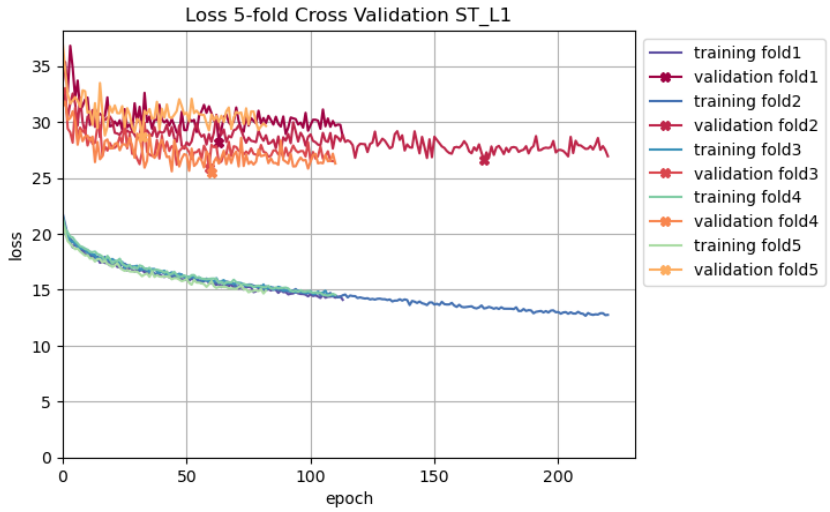


Figure A.5: Training- and validation loss for 5-fold cross validation when using the distance function d_{L1} with single tile calculation.

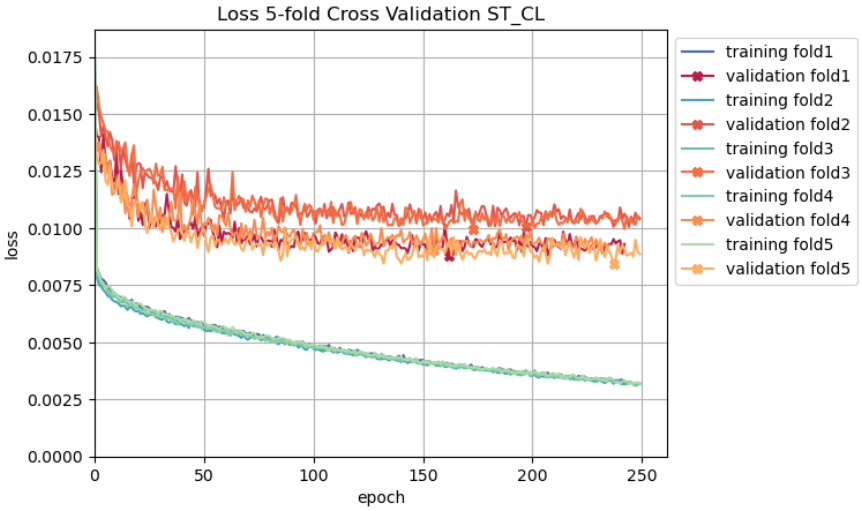


Figure A.6: Training- and validation loss for 5-fold cross validation when using the distance function d_{CL} with single tile calculation.

A.2.2 Multi Tile Calculations

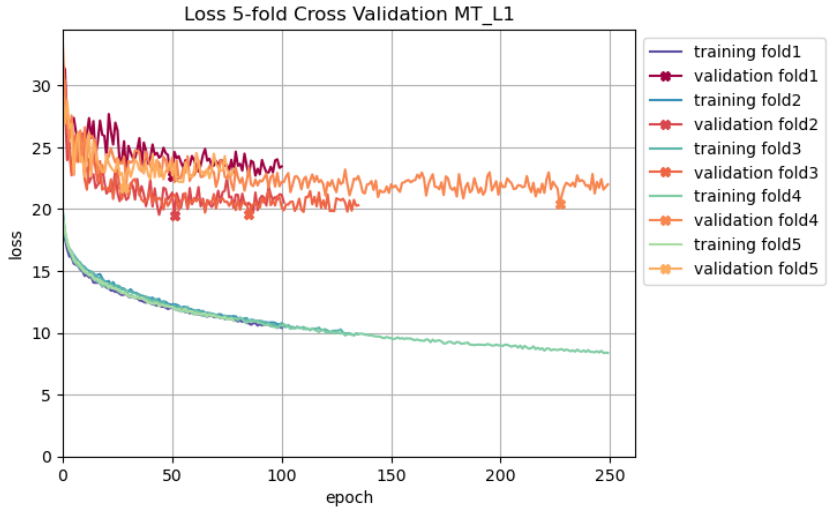


Figure A.7: Training- and validation loss for 5-fold cross validation when using the distance function d_{L1} with multi tile calculation.

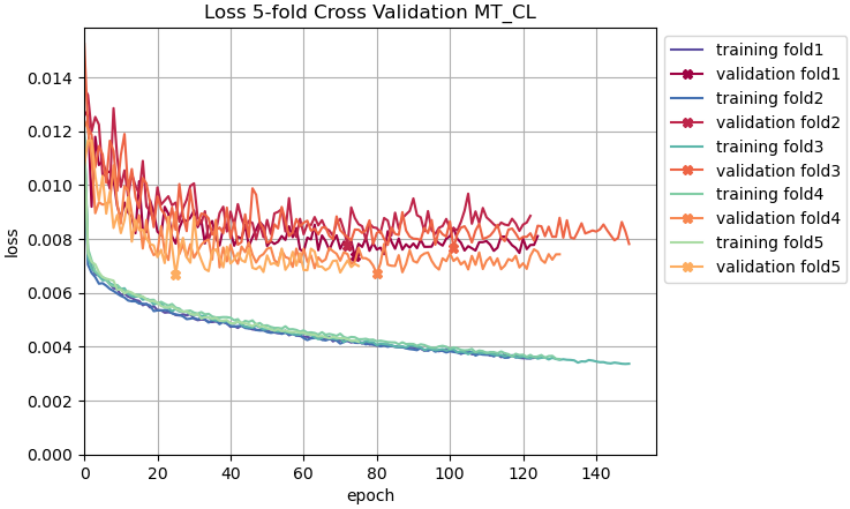


Figure A.8: Training- and validation loss for 5-fold cross validation when using the distance function d_{CL} with multi tile calculation.

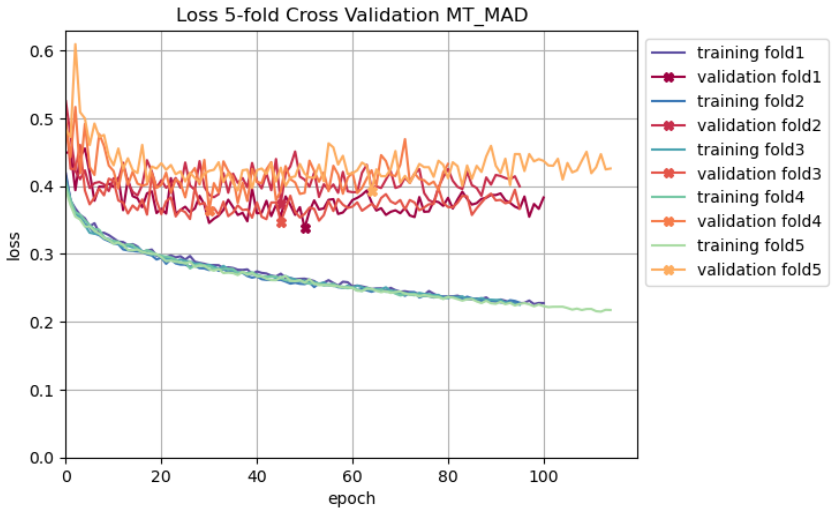


Figure A.9: Training- and validation loss for 5-fold cross validation when using the distance function d_{MAD} with multi tile calculation.

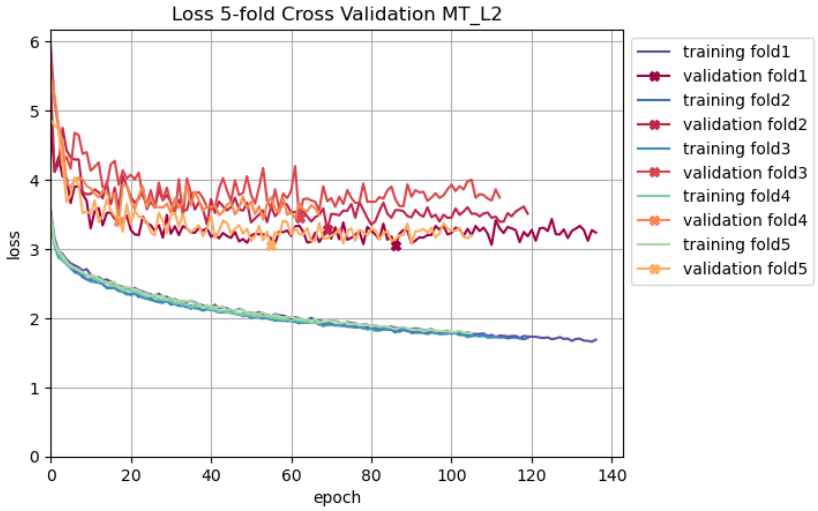


Figure A.10: Training- and validation loss for 5-fold cross validation when using the distance function d_{L2} with multi tile calculation.

Appendix B

Presentation of Distance Distributions

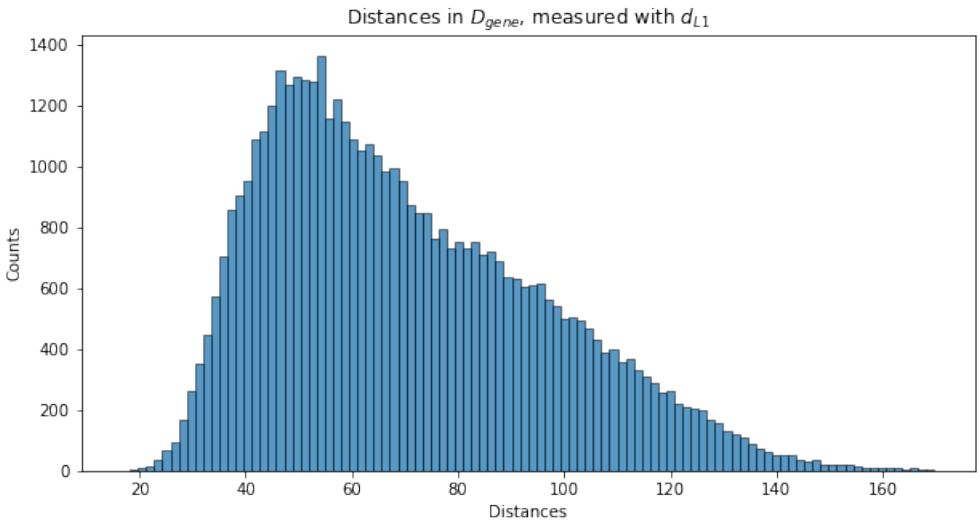


Figure B.1: Distribution of distances measured with d_{L1} in Φ_{gene} between WSIs in the training data.

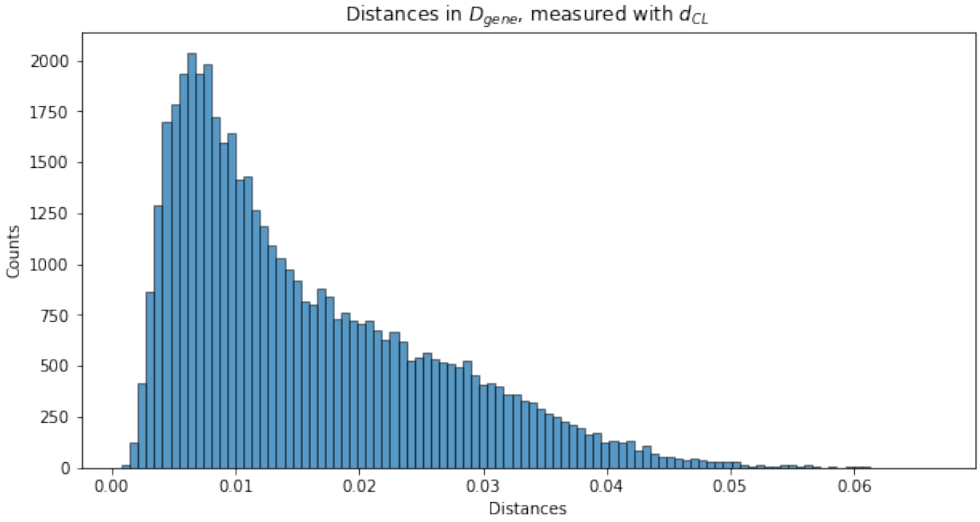


Figure B.2: Distribution of distances measured with d_{CL} in Φ_{gene} between WSIs in the training data.

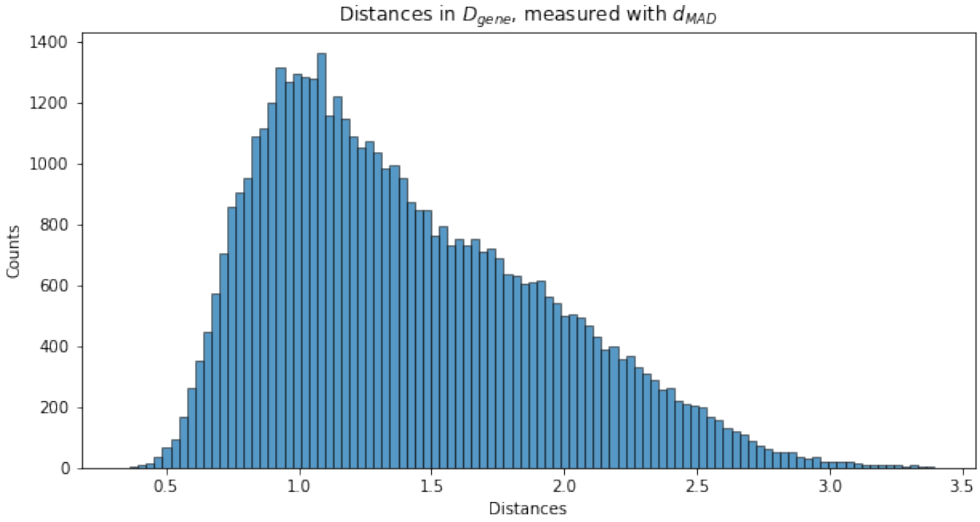


Figure B.3: Distribution of distances measured with d_{MAD} in Φ_{gene} between WSIs in the training data.

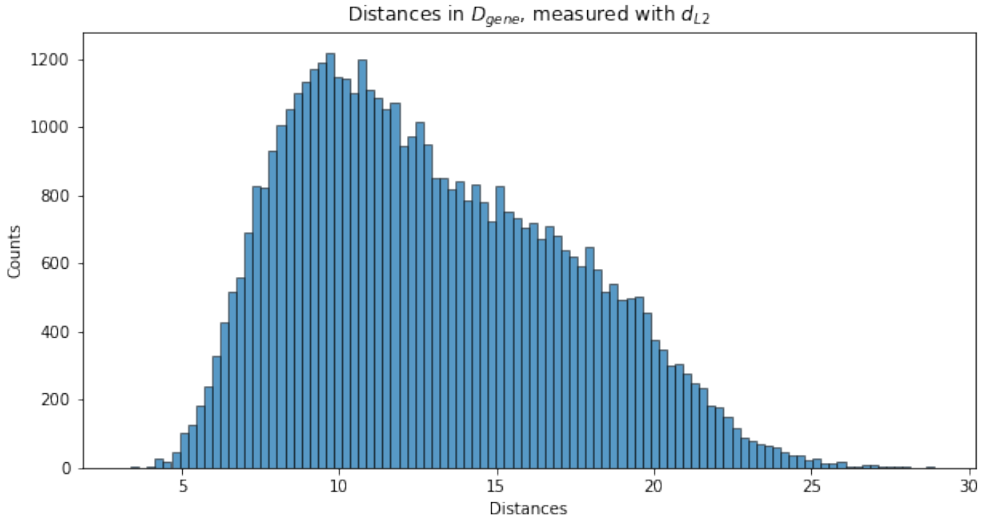


Figure B.4: Distribution of distances measured with d_{L2} in Φ_{gene} between WSIs in the training data.